

地方中小企業 新卒エンジニア向け

新卒研修シラバス

Syllabus for New-Graduate Engineers

技術の前に、人としての軸を。

バージョン v10.05

公開日 2026年6月30日

公開先 無償公開版（無保証・自己責任利用）

研修期間 全44日（約2ヶ月半）

対象 新卒ITエンジニア



<https://quality-arts.com/>

© 2026～ クオリティアーツ

本資料の取り扱いと注意事項

本資料は、クオリティアーツが地方の中小IT企業における新卒エンジニア研修のために作成したものです。このたび、地域の人材育成への社会貢献を目的として、非営利での利用に限り、無保証・無償で公開いたします。本資料をご利用になる場合、以下のすべてに同意したものとみなします。ご利用は、すべて利用者ご自身の責任において行ってください。

- ① 【利用範囲】 本資料は、非営利の利用（企業内の新卒研修・個人の学習・教育目的での使用等）に限り、無償で利用できます。営利目的または有償での利用（販売、有料の研修・セミナー・講座等での使用、商用サービスへの組み込み等を含む）は禁止します。
- ② 【著作権・二次利用】 本資料の著作権はクオリティアーツ（以下、当社）に帰属します（© クオリティアーツ 2026）。内容の転載・複製・改変・翻案・流用・再配布等は、目的・媒体を問わず原則として認めません。ただし、①の非営利の利用範囲に限り、本資料をもとに自社の新卒研修教材・カリキュラム等を作成し、社内で使用・配布することができます。その場合は、出典として当社の著作権クレジット（例：© クオリティアーツ 2026）を表記してください。この特例は社内での利用に限られ、営利目的・有償での利用、第三者への販売・公開・再配布、および本資料そのもの（全部または一部の原文）の転載・再配布は、引き続き認めません。
- ③ 【生成AIの利用】 本資料の全部または一部を、生成AI・機械学習サービス等へ入力・アップロード・学習データとして提供することを禁止します。なお、本資料の作成・編集の一部には生成AIを利用しています。
- ④ 【商標】 本文中に記載の会社名・製品名・サービス名等は、各社の商標または登録商標です。本文では®・TM等の記号を省略している場合があります。
- ⑤ 【記載時点】 本資料は2026年時点の情報・法令・制度に基づいて作成しています。法改正や制度変更により、記載が古くなる場合があります。
- ⑥ 【事例の扱い】 本文中の事例のうち、企業名・人名を挙げていないものは、要点を伝えるための学習用の例（創作・合成）です。実在の事件・出来事に触れる場合も、関係者が特定されないよう、社名や年などを伏せて一般化していることがあります。
- ⑦ 【試算について】 複利・手取り等の試算は一定の前提に基づく概算であり、結果を保証するものではありません。投資には元本割れ等のリスクがあります。
- ⑧ 【自己責任】 本資料の利用は、その全てを利用者自身の判断と責任において行うものとします。給与・税・社会保険・NISA/iDeCo・労働法・各種規格（ISO/OWASP等）の数値や要件は、実務では必ず官公庁・公式サイト等の最新の一次情報を確認してください。
- ⑨ 【無保証・ノーサポート】 本資料は「現状有姿（AS IS）」で提供されます。当社は、本資料の正確性・完全性・最新性・有用性・特定目的への適合性、および第三者の権利を侵害しないことについて、明示・黙示を問わず一切保証しません。また、本資料に関する更新・保守・サポート、内容に関するお問い合わせ・ご質問・誤りのご指摘への対応等は、一切行いません。
- ⑩ 【免責】 本資料の利用または利用できないことに起因して、利用者または第三者に生じたいかなる損害（直接・間接・付随的・特別・結果的損害、逸失利益、データの喪失、業務の中断、第三者からの請求等を含みますが、これらに限りません）についても、当社は理由のいかんを問わず一切責任を負いません。
- ⑪ 【その他・包括的免責】 本注意事項に明示されていない事項を含め、本資料および本資料の利用に関連して生じた一切の事項について、当社は何らの責任も負いません。本資料は、利用者ご自身の責任においてのみ利用できるものとします。

はじめに

人の一生には、土台が据えられる時期があります。新卒社員として迎える最初の季節、とりわけ入社して間もない研修のひとつは、その人がこれから歩む数十年、すなわち職業人としての道のりも、一人の人間としての生き方も、その根を静かに張っていく時間です。この研修資料（シラバス）は、その大切な時期のために、私がこれまで重ねてきた新卒研修のなかで得た考え方や資料を、一つのかたちにとまとめたものです。

若い人にまず手渡すべきは、道具の使い方ではないと、私は考えています。プログラミング言語も、手順も、いずれ時とともに身につくものです。それよりも先に渡したいのは、道具を手にする前に立つ、一人の人間としての構えそのものです。けれども現実には、「即戦力」という言葉がいつのまにか、目の前の技術や手順を覚えさせることへと読み替えられ、そこに重きを置いた新卒研修が、少なからず行われています。私は、そうした光景を、これまで幾度も目にしていまいりました。

そこで本シラバスでは、目の前の技術や手順といった、いずれ学べるものは、実践講座やOJTに譲ることといたしました。そのかわりに据えたのは、一つひとつの仕事の意義と目的、その奥にある背景、そして新卒社員として最低限必ず知っておかねばならない知識、すなわち「なぜそう考えるのか」という、目には見えない芯の部分です。

本シラバスが終始一貫して大切にしているのは、ただ一つの問いです。「正しく、誠実に、人のために働くとは、どういうことか」。技術の巧拙よりも先に、この問いに自分なりの答えを持てるかどうか。そこにこそ、長く信頼され、長く立ち続けられるエンジニアの、そして一人の人間の、揺らがぬ礎があると、私は確信しています。

本シラバスは、この問いを四つの層からとらえています。一人の人として、ビジネスパーソンとして、エンジニアとして、そして組織の一員（社員）として。この四つの層のそれぞれに必要な事柄を、できるかぎり広く、ていねいに記しました。これを手にする新卒社員には、研修を終えたとき、この四つの層を自らの知識として携え、現場で、そしてこれからの人生で、実践してほしいと思います。

そして、新卒社員にそう望むとき、私は迎える側のことを思わずにはられません。なぜなら、新卒社員を迎えるということは、その人の今後の人生を、決定的に方向づけてしまうことでもあるからです。それは、一人の若者を労働力として雇うことではなく、その人のこれからの人生ごと、預かることにほかなりません。とりわけ最初の新卒研修は、まっさらな心に最初の輪郭を刻む、いわば刷り込みです。配属の前に最初に映る景色、最初に出会う人（講師）が何を語るのか。その一つひとつが、その人の価値観の芯となって、長く残り続けます。だからこそ、それは育てる側に立つ企業や管理職こそが、もっとも真剣に考えねばならないことだと、私は思っています。私自身、一人の人間の出発点を、そしてその後の人生をも左右しかねない重さを忘れぬよう、覚悟をもってこの仕事に向き合っていました。

もし本シラバスが、単なる作業者ではなく、広く社会に、そして力強く企業で活躍する人を育てる研修づくりに、ささやかでも寄り添えたなら、これにまさる喜びはありません。これを

無償で公開するのには、二つの願いがあります。一つは、これまで多くの企業と社会に育てられ、一人のエンジニアとして歩んでこられた、その恩に、ささやかでも報いたいということ。そしてもう一つは、一人でも多くの新卒社員に、ただ作業をこなすだけの人生ではなく、自らの手で未来を掴んでいくエンジニアとしての人生を、歩んでいってほしいということです。

新卒社員がこれから歩む道には、楽しいことばかりでなく、困難も待ち受けています。けれども、ここで手にした軸を信じて歩んでいけば、その困難は必ず越えられる。そしてその先には、豊かなエンジニアとしての歩みと、豊かな人としての人生が待っている。私はそう信じています。本シラバスが、巣立っていく一人ひとりの確かな第一歩を、そっと支える灯となることを、心から願っています。

二〇二六年 六月 三十日

クオリティアーツ 池田 暁

本シラバスの使い方

「シラバス」とは、もともと大学などで用いられる授業計画書のことで、何を・どんな順序で・どこまで学ぶのかを示した「学びの設計図」です。本書もその考え方になっています。完成した研修パッケージ（スライドや手順書）ではなく、研修をかたちにするための土台であり、同時に一冊で読み通せる「読み物」でもあります。次の二通りの使い方ができます。

■ 二つの使い方

(1) **研修をつくる土台として**——本書を設計図に、自社に合わせた教材・演習・スケジュールを組み立てる。

(2) **一冊の読み物として**——研修の有無にかかわらず、最初から通読して社会人の基礎を身につける。

■ 立場ごとの使い方

研修担当者・講師の方へ：本書を出発点に、自社の事業・技術・文化に合わせて実際の教材や演習を作ってください。各章の「ゴール」「まず自己診断」「やってみよう」「考えてみよう」は、そのまま研修の骨組みに使えます。順序や分量、日数（44日は標準例です）は、受講者の理解度や現場の事情に合わせて調整して構いません。変更したときは、その理由を受講者に伝えると効果的です。

新卒・受講者の方へ：予習・復習に使ってください。各章の冒頭で「読む前の自己診断」をつけ、読んだあとに「理解度チェック」で前後を比べると、自分の伸びが見えます。研修がなくても、通読すれば社会人としての軸が身につくように書いてあります。現場で迷ったときに開き直す「辞書」としても使えます。

迎える側（管理職・経営層）の方へ：新人が何を学ぶのかの全体像をつかみ、受け入れる側の期待値をそろえるために使ってください。新卒を一人のビジネスパーソンとして育てるという本書の姿勢は、迎える側の心構えにも通じます。

■ 各章の読み方

各章は「ゴール → まず自己診断 → 本文 → 事例から学ぶ → 考えてみよう → やってみよう → 理解度チェック」という同じ流れで進みます。各要素は専用のアイコンで示しているので、アイコンを手がかりに拾い読みもできます。巻末には、さらに深めたい人のための付録・自習の手引きを置いています。

■ 使うときの前提

本書は「正しく、誠実に、人のために働くとは何か」という一つの問いを軸にし、技術や手順そのものは実践講座やOJTに委ねています。記載は2026年時点の情報・法令に基づくため、実際に使うときは最新の制度を確認してください（詳しくは「本資料の取り扱いと注意事項」）。

■ ご注意

本シラバスは、作成にあたってできる限り事実確認を行っていますが、誤りが含まれている可能性があります。もし誤りや古くなった記載を見つけた場合は、研修資料を作成する際に、訂正した内容へ差し替えるなどの対応をお願いします。本資料は現状有姿（AS IS）・無保証で提供されます（詳しくは「本資料の取り扱いと注意事項」）。

■ 出典の表記について

本シラバスをもとに研修資料を作成・配布する場合は、出典としてクオリティアーツの著作権クレジット（例：© クオリティアーツ 2026）を表記してください。著作権・転載等の詳細は「本資料の取り扱いと注意事項」に従ってください。

目次

※ページ番号は自動生成です。表示されない場合は、目次の上で右クリック → 「フィールド更新」を選んでください。

研修全体スケジュール	14
序章 なぜあなたはここにいるのか（1日目）	15
0-1 今日という日について	15
0-2 入社は、入学ではありません—あなたは今日から「ビジネスパーソン」です	15
0-3 最初の問い—あなたは今、何をしに来ましたか	17
0-4 あなたの仕事は社会につながっている	17
0-5 今日から変わること—学生とビジネスパーソンの違い	18
0-6 「正しさ」には4つの層がある	18
0-7 なぜこの研修が存在するのか—44日間の意味	20
0-8 この研修の全体像—44日間の旅	20
0-9 「使用者と労働者」—入社初日に知っておくべき関係性	21
0-10 プロフェッショナルリズムとは何か—入社初日から始まる問い	22
0-11 44日後の自分へ—この研修に向き合う姿勢	23
第1部 土台をつくる（2～6日目・5日間）	25
第1章 社会人・ビジネスパーソンとしての出発点（2～3日目・1.5日）	26
1-1 学生と社会人の本質的な違い	26
1-2 あなたはひとりのビジネスパーソンである	28
1-3 「指示待ち」から「自分で考えて動く」への転換	29
1-4 入社後「最初の90日間」を成功させる—新卒社員のスタートダッシュ	30
1-5 プロとしての自覚—「アマチュア」と「プロ」を分けるもの	31
1-6 職場で「絶対に避けるべき」言動—信頼を瞬時に壊すもの	32
1-7 理解度チェック（読む前の自分とくらべよう）	34
第2章 会社という組織の中で働くとはどういうことか（3～4日目・1.5日）	35
2-1 会社はなぜ存在するのか	35
2-2 「使用者」と「労働者」—会社の中での自分の立場を知る	36
2-3 自社・業界を知る	36
2-4 ステークホルダーを理解する	37
2-5 上司・同僚・部下とはなにか—人間関係の本質を理解する	39
2-6 組織のマネジメント—分割統治・権限委譲・レポートラインを理解する	40
2-7 役職の地図—「役職は偉い順ではない」	42
2-8 トップと現場のすれ違いは、なぜ起きるのか—視座を上げる	46
2-9 理解度チェック（読む前の自分とくらべよう）	47
第3章 ビジネスマナーの基本（4～5日目・1.5日）	48
3-1 ビジネスマナーの本質—なぜマナーが必要か	48
3-2 身だしなみと職場環境	49
3-3 対面のマナー	50
3-4 テレワーク・リモート環境でのマナー	51
3-5 敬語の使い方—正しく・自然に・臆せず	52
3-6 社内コミュニケーションの作法—上司・先輩との日常の関わり方	54
3-7 社外コミュニケーション—顧客・取引先との関わり方	55
3-8 理解度チェック（読む前の自分とくらべよう）	56
第1部 総合ワークショップ（6日目・1日）	57
第2部 動き方を身につける（7～15日目・9日間）	59
第4章 報告・連絡・相談—情報を正しく流す（7日目・1日）	60
4-1 報連相の本質	60
4-2 報連相の実践	61

4-3	報告の場面別実践—状況に応じた使い分け	63
4-4	連絡の技術—情報を「届ける」から「使われる形で届ける」へ	65
4-5	相談の技術—「迷惑にならないか」という不安を超えて	66
4-6	報連相の失敗パターンと対処法	68
4-7	理解度チェック（読む前の自分とくらべよう）	70
第5章	仕事と作業の違い—仕事の進め方（8～9日目・1.5日）	71
5-1	「仕事」と「作業」はなにが違うのか	71
5-2	上長から「仕事」をもらう	72
5-3	仕事の進め方—段取りと品質の基本	72
5-4	プロジェクト管理の基本—チームで仕事を進める技術	73
5-5	時間管理の実践—「忙しい」から「効果的」へ	74
5-6	問題解決の実践プロセス—「詰まったとき」に動ける力を持つ	75
5-7	プロの視座の核心—後工程はお客様・自工程完結・カイゼン	77
5-8	「業務」とは何か—改善も育成も、立派な業務である	79
5-9	理解度チェック（読む前の自分とくらべよう）	82
第6章	人と働く—コミュニケーションと信頼の築き方（9日日後半～11日目・2日）	84
6-1	伝える・聴く・確認する	84
6-2	コミュニケーションの実践技術	85
6-3	チームワークと協働	86
6-4	社内外の関係構築	87
6-5	難しい会話を乗り越える—コンフリクトと向き合う力	87
6-6	感情と仕事—EQ（感情知性）を育てる	88
6-7	1on1・会議・オンラインコミュニケーション—場面別の実践	90
6-8	生産性の高いチームで働く—プロとして生産性を意識する	91
6-9	理解度チェック（読む前の自分とくらべよう）	92
第7章	文書作成・プレゼンテーション・会議（11日日後半～13日目・2日）	94
7-1	ビジネス文書の基本	94
7-2	ビジネスメールの実践—伝わるメールを書く技術	95
7-3	報告書・議事録・技術文書の書き方	96
7-4	伝わるプレゼンテーション	97
7-5	プレゼンテーションの実践技術—「話す力」を磨く	98
7-6	会議の進め方	99
7-7	会議をムダにしない—ファシリテーションと参加者の技術	100
7-8	理解度チェック（読む前の自分とくらべよう）	102
第2部	総合ワークショップ（14～15日目・2日）	103
第3部	考える力・数字の力（16～20日目・5日間）	105
第8章	ロジカルシンキングと問題解決（16～17日目・2日）	106
8-1	ロジカルシンキングの基本	106
8-2	問題解決の手法	109
8-3	ピラミッドストラクチャー—「伝わる論理」の設計図	111
8-4	プレゼンテーションの論理構造—「話す論理」と「書く論理」	112
8-5	数字と論理で「説得力」を上げる—根拠のある主張	114
8-6	情報収集と整理—正しい情報を効率よく手に入れる	114
8-7	理解度チェック（読む前の自分とくらべよう）	115
第9章	数字・財務の基本（18～19日目・2日）	116
9-1	なぜ数字を読めることが大切か	116
9-2	財務三表の基本	117
9-3	数字の使い方—ビジネスでの実践	118
9-4	身近な数字—給与・税金・社会保険の実務	119

9-5 数字で考える・数字で話す—ビジネスパーソンとしての基礎	120
9-6 理解度チェック（読む前の自分とくらべよう）	122
第3部 総合ワークショップ（20日目・1日）	123
中間振り返り（20日日後半・0.5日）	125
第4部 正しさを知り、身につける（21～30日目・10日間）	126
中間振り返り—20日間の学びを統合する	126
第10章 「正しさ」の4層を解剖する（21日目・1日）	128
10-1 人としての正しさ—普遍的な誠実さ・品格	128
10-2 社会人としての正しさ—法令・契約・守秘義務	129
10-3 専門家としての正しさ—専門知識を持つ者の責任	130
10-4 社員としての正しさ—組織への貢献と正直な発言	131
10-5 4層が衝突したとき、どう判断するか	132
10-6 倫理的ジレンマの解決フレームワーク—迷ったときの判断軸	133
10-7 内部通報制度—組織の不正と向き合う勇気	134
10-8 理解度チェック（読む前の自分とくらべよう）	135
第11章 社会人が知っておくべき法律の基本（22～23日目・2日）	136
11-1 法律を知ることの意義	136
11-2 使用者と労働者—労働法の大前提	138
11-3 会社の仕組みを定めた法律（会社法）	139
11-4 契約・取引を定めた法律	139
11-5 情報・不正行為を定めた法律	140
11-6 法律の全体マップ—自分の仕事とどう関わるか	141
11-7 「知らなかった」では済まない—法リスクに気づく日常習慣	142
11-8 理解度チェック（読む前の自分とくらべよう）	144
第12章 人を傷つけないために—ハラスメント・感情・誠実・公正（24日目・1日）	146
12-1 この2章で目指す3つの立場—加害者にならない／身を守る／巻き込まれない	146
12-2 ハラスメントの全体像—種類・3要件・360度	147
12-3 「悪意がない」が一番危ない—気づかない加害	149
12-4 ハラスメントと法律—自分と仲間を守る知識	150
12-5 感情とつきあう—怒りを制御し、他人の怒りに飲まれない	151
12-6 言葉は凶器になる—陰口・呼び捨て・SNS・記録	153
12-7 誠実さ—嘘をつかない・隠さない・ごまかさない	154
12-8 公正であること—機会と評価は公平に開かれているか	155
12-9 理解度チェック（読む前の自分とくらべよう）	155
第13章 健全な組織をつくるために—知っておくこと、できること（25日目・1日）	157
13-1 組織の健全さを支えるもの—そして、人が流される仕組みを知る	157
13-2 健全な職場・リーダーの「ものさし」—違和感に気づく力	159
13-3 健全さを守り、自分も守る—染まらないための構え	160
13-4 困ったときに—相談先・証拠・周りの一歩	162
13-5 心理的安全性と、あなたがリーダーになるとき	163
13-6 理解度チェック（読む前の自分とくらべよう）	164
第14章 情報を扱う責任（26日目・1日）	165
14-1 情報へのアクセスと責任—組織人として知っておくべきこと	165
14-2 情報セキュリティの基本	166
14-3 SNS・情報発信のルール	167
14-4 SNS・インターネット上の情報発信と法的リスク	168
14-5 情報を「漏らさない」ために—自分とキャリアを守る実践	169
14-6 理解度チェック（読む前の自分とくらべよう）	172
第15章 会社と社会を守るルール（27日目・1日）	173

15-1	コーポレートガバナンスとコンプライアンス	173
15-2	ISMS・QMS—組織を守る「マネジメントシステム」を理解する	174
15-3	インサイダー取引	178
15-4	独占禁止法・カルテル	179
15-5	公益通報制度	179
15-6	理解度チェック（読む前の自分とくらべよう）	180
第16章	働く権利・環境・社会への責任（28日目・1日）	181
16-1	労働法の整理と補強	181
16-2	文書管理と証跡—記録を残す意義	182
16-3	社会・環境への責任	182
16-4	労働時間の法律—「何時間まで働かせてよいか」を理解する	182
16-5	有給休暇—「取れる権利」を正しく理解する	184
16-6	育児・介護・その他の制度—ライフイベントに備える	184
16-7	メンタルヘルス—自分の心を守ること仕事のうち	185
16-8	副業・兼業—複数の仕事を持つ時代への理解	186
16-9	テレワーク・多様な働き方と法律	187
16-10	理解度チェック（読む前の自分とくらべよう）	188
第4部	総合ワークショップ（29～30日目・2日）	189
第5部	自分を育てる（31～34日目・4日間）	191
第17章	成長し続けることの重要性と阻害要素への対応（31～32日目・2日）	192
17-1	成長し続けることで広がる4つの人生の可能性	192
17-2	どんな環境でも成長を続ける方法	193
17-3	成長の習慣をつくる	195
17-4	成長を阻む「内側の壁」—自分で自分を止めないために	195
17-5	学習の技術—効果的に、長く、楽しく学ぶために	197
17-6	フィードバックを受け取る力—成長の最大の加速装置	198
17-7	理解度チェック（読む前の自分とくらべよう）	200
第18章	キャリアプラン—自分の成長を設計する（33～34日目・2日）	201
18-1	キャリアは「自分がつくるもの」です	201
18-2	スキルの三層構造と成長の方向	202
18-3	新卒3年間のステージ設計	204
18-4	リスクリングと自己投資	206
18-5	1on1・評価面談の活かし方—成長の機会として使い倒す	207
18-6	理解度チェック（読む前の自分とくらべよう）	209
第6部	人生という視点で、働くことを設計する（35～37日目・3日間）	210
第19章	ライフプラン—人生とお金を設計する（35～36日目・2日）	211
19-1	人生の全体地図を描く	211
19-2	お金のリアル—手取り・固定費・3年で貯められる額	212
19-3	人生の四大支出を知る	215
19-4	社会保険と資産形成の基本	216
19-5	働き方と制度の変化を見通す	216
19-6	NISAとiDeCo—今すぐ始めるべき資産形成の実践	217
19-7	人生の各ステージでの備え—保険と緊急予備資金	218
19-8	税金の基本—社会人として知っておくべき最低限の知識	218
19-9	理解度チェック（読む前の自分とくらべよう）	220
第20章	仕事と人生を統合する—「なぜ働くのか」を問い直す（37日目・1日）	222
20-1	「生活のため」を超えた働く動機	222
20-2	変化に適応し続ける自分をつくる	224
20-3	序章で書いた「大切にしたいこと」と照らし合わせる	225

20-4	ウェルビーイング—豊かな仕事人生のために	225
20-5	人生の意味と仕事—「何のために働くか」の問いと向き合う	226
20-6	「正しく・誠実に・よく」生きるということ—44日間の学びの統合	227
20-7	理解度チェック（読む前の自分とくらべよう）	230
第7部	「特論：新卒社員向けITエンジニア入門 —エンジニアとしての入口に立つために知っておくべきこと」（38～43日目・6日間）	231
第21章	本当のエンジニアとは—あるべき姿・倫理・必須スキル（38日目）	232
21-1	概要—「作業」ではなく「仕事」をする／この章で目指すエンジニア像	232
21-2	歴史と背景—「プログラマ→SE→ソフトウェアエンジニア」の変遷と技術者倫理の成立	233
21-3	あるべきエンジニア像とプロフェッショナリズム	233
21-4	技術者倫理の基礎—綱領と誠実さ	234
21-5	迷ったときの判断—説明責任と倫理的意思決定	235
21-6	必須スキルの全体像—テクニカル／ヒューマン／コンセプチュアル	236
21-7	主要BOKの見取り図—机に常備する3大BOK	236
21-8	コンピュータサイエンス基礎の全体像	237
21-9	自走する力—調べる・読む・学ぶ	237
21-10	問題解決とデバッグの思考	237
21-11	段取る力—分解・見積もり・優先順位	238
	〔コラム〕受託のビジネス構造（人月・稼働率・原価率）／客先データと倫理綱領	238
21-12	考えてみよう（グループ討議）	239
21-13	理解度チェック（読む前の自分とくらべよう）	240
第22章	IT産業の現在地と未来（39日目）	241
22-1	概要—なぜ業界の現在地を知るのか	241
22-2	歴史と背景—情報処理の加速	241
22-3	産業の構造—SIerピラミッドとバリューチェーン	242
22-4	働き方とつくり方の現在	244
22-5	人材市場と学びの変化	244
22-6	未来の見通し—技術トレンドと産業の行方	245
	〔コラム〕受託の位置取りと伸びる4つの道	246
22-7	考えてみよう（グループ討議）	247
22-8	理解度チェック（読む前の自分とくらべよう）	248
第23章	ソフトウェアエンジニアリングの体系（SWEBOK V4）（39日目）	249
23-1	概要—なぜ体系を学ぶのか／「書ける」と「働ける」は別物	249
23-2	歴史と背景—ソフトウェア危機とソフトウェア工学の誕生	250
23-3	SWEBOK V4 全体地図—18の知識領域	251
23-4	要求と要件定義—何を作るべきかを定める	253
23-5	アーキテクチャと設計—設計原則とデータ設計	254
23-6	モデルで考える—UMLで設計を伝える	256
23-7	構築と構成管理—バージョン管理とコードレビュー	256
23-8	運用と保守—動かし続ける責任	257
23-9	マネジメント・プロセス・経済性	258
23-10	エンジニアのロールはこんなに広い	258
	〔コラム〕仕様の握りと検収／保守は受託の収益源	259
23-11	考えてみよう（グループ討議）	259
23-12	理解度チェック（読む前の自分とくらべよう）	260
第24章	開発プロセスとアジャイル（40日目）	261
24-1	概要—なぜプロセスを学ぶのか	261
24-2	歴史と背景—開発プロセスの系譜	261

24-3	受託開発の流れ全体—一枚の絵で押さえる	262
24-4	アジャイルの価値観	262
24-5	スクラムとカンバンの基礎	263
24-6	DevOpsと継続的デリバリ—開発と運用をつなぐ	264
24-7	構成管理をプロセスに組み込む—ブランチ戦略・プルリクエスト・GitOps	264
24-8	ウォーターフォールとアジャイルの使い分け・ハイブリッド	265
24-9	ふりかえりとカイゼン—レトロスペクティブ	266
24-10	受託でアジャイルをやる難しさ—契約・準委任	266
	〔コラム〕受託アジャイルの落とし穴／客先指定のブランチ運用・CI	267
24-11	考えてみよう（グループ討議）	268
24-12	理解度チェック（読む前の自分とくらべよう）	269
第25章	品質管理とソフトウェアテスト（40日目）	270
25-1	概要—なぜ品質が最優先か／QA（作り込む）とテスト（検出）の違い	270
25-2	歴史と背景—デミングからTQMへ／テストの目的の変遷	271
25-3	品質管理の体系—SQuBOKと品質特性（ISO/IEC 25010:2023）	271
25-4	品質を作り込む—コスト・メトリクス・文化	272
25-5	ソフトウェアテストの体系—JSTQB（CTFL）とISO/IEC/IEEE 29119	273
25-6	レビューと静的解析—動かす前に見つける	276
25-7	品質・テストは、全エンジニアの「当たり前技術」	277
	〔コラム〕品質=契約不適合責任と検収	277
25-8	考えてみよう（グループ討議）	278
25-9	理解度チェック（読む前の自分とくらべよう）	279
第26章	セキュリティの基礎（41日目）	280
26-1	概要—なぜ全エンジニアの責任か	280
26-2	歴史と背景—性善説から「バイ・デザイン」へ	281
26-3	基礎概念—守りの共通言語	281
26-4	主要な脅威を知る—OWASP Top 10	283
26-5	依存・サプライチェーンの基礎	284
26-6	日常のセキュリティ習慣	285
26-7	インシデント対応の初動	285
	〔コラム〕客先の本番データ・多重下請けの責任分界	286
26-8	考えてみよう（グループ討議）	286
26-9	理解度チェック（読む前の自分とくらべよう）	287
第27章	ドキュメントとプレゼンテーション（41日目）	288
27-1	概要—なぜ「書く・伝える」が技術なのか	288
27-2	歴史と背景—テクニカルライティングとドキュメント文化の発展	289
27-3	テクニカルライティングの基礎	289
27-4	エンジニアの文書とDocs as Code	290
27-5	図で伝える	291
27-6	プレゼンテーション	292
27-7	レビューと非同期コミュニケーション	293
	〔コラム〕客先フォーマット・議事録が身を守る	293
27-8	考えてみよう（グループ討議）	294
27-9	理解度チェック（読む前の自分とくらべよう）	295
第28章	最低限知っておくべき法律（42日目）	296
28-1	概要—なぜ法を知るか／法の全体像	296
28-2	歴史と背景—情報化と法整備の歩み	297
28-3	知的財産—成果物の権利は誰のものか	297
28-4	情報セキュリティ関連法と営業秘密	298

28-5	個人情報保護法	298
28-6	受託の取引法①—契約形態と成果物責任	299
28-7	受託の取引法②—派遣法と取適法	300
28-8	労働法の基礎と、これからの法	300
	〔コラム〕受託成果物の権利帰属／検収トラブルと仕様書／客先常駐で指揮命令を受けたら	301
28-9	考えてみよう（グループ討議）	302
28-10	理解度チェック（読む前の自分とくらべよう）	302
第29章	生成AIによる開発の現在（42日目）	304
29-1	概要—AIはエンジニアに何をもたらすか	304
29-2	歴史と背景—推論から学習、そして生成へ	305
29-3	LLMの仕組み超入門—なぜ間違えるのか	305
29-4	AI開発の進化と新しいスタイル	306
29-5	AIを使いこなすスキル	307
29-6	AIの使いどころと注意点	308
29-7	AI時代にこそ体系が効く	309
	〔コラム〕受託でのAI活用と顧客機密・成果物責任	309
29-8	考えてみよう（グループ討議）	310
29-9	理解度チェック（読む前の自分とくらべよう）	310
第30章	活躍できるエンジニアへの成長（43日目）	312
30-1	概要—なぜ学び続けるのか（一生もののOS／陳腐化しない力）	312
30-2	歴史と背景—OSS・コミュニティの系譜	313
30-3	学びの科学	313
30-4	自分の現在地を測る—成長段階と技術ロードマップ	314
30-5	キャリアをどう設計するか	315
30-6	メンターとロールモデルの見つけ方	316
30-7	外へ開く—発信とコミュニティ	316
30-8	燃え尽きずに続ける	317
	〔コラム〕受託から伸びる道	318
30-9	考えてみよう（グループ討議）	318
30-10	理解度チェック（読む前の自分とくらべよう）	319
終章	研修の総ざらい—正しいエンジニアとして、現場へ（44日目・1日）	320
終-1	序章から、全44日をたどり直す—部ごとのふり回り	320
終-2	4つの正しさで束ね直す—人・社会人・専門家・社員として	321
終-3	到達の自己点検—白紙に書き出して、たしかめる	322
終-4	困ったら、ここへ戻る—逆引きマップ	322
終-5	ワークショップの進め方	323
終-6	発表① わたしの行動指針—4層で自分の判断を語る	323
終-7	発表② グループ・ケーススタディ—4層で優先順位を論じる	323
終-8	ここからは現場—特別章（配属後OJT）への橋	324
終-9	正直に言う—現場には、困難が待っている	324
終-10	それでも、越えていける—正しいエンジニア像を、守り、極める	324
終-11	これから数十年の、エンジニア人生と、人としての人生のために	325
特別章	配属後OJTガイド—研修の学びを、現場で育てる	326
特-0	これは、あなたがOJTを「運転する」章	326
特-1	最初の2週間、観察して「運転席」に座る	327
特-2	指導役を「使い倒す」	328
特-3	フィードバックを「取りに行く」	329
特-4	振り返りを「自走」させる	329
特-5	90日から1年の歩き方	330

特-6 停滞も、つまずきも、通過点	330
特-7 現場で使う「正しさ」を、自分事として生きる	331
特-8 OJTは、ずっと続く	332
特-付録 OJT行動チェックリスト	333

研修全体スケジュール

週	日程	主な内容
第1週	1～5日目	序章・第1部（土台：社会人の自覚・会社理解・ビジネスマナー）
第2週	6～10日目	第1部総合WS・第2部前半（報連相・仕事と作業）
第3週	11～15日目	第2部後半（コミュニケーション・文書作成・プレゼン・会議）・第2部総合WS
第4週	16～20日目	第3部（ロジカルシンキング・財務・数字）・中間振り返り
第5週	21～25日目	第4部前半（正しさの4層・法律・ハラスメント）
第6週	26～30日目	第4部後半（情報・遵法・権利・総合ワークショップ）
第7週	31～35日目	第5部（成長・キャリア）・第6部前半（ライフプラン）
第8週	36～40日目	第6部（仕事と人生の統合）・第7部 特論：新卒社員向け ITエンジニア入門（前半）
第9週	41～44日目	第7部 特論：新卒社員向け ITエンジニア入門（後半）・終章（最終発表）

序章 なぜあなたはここにいるのか（1日目）

🎯 この章のゴール

この章では、これからの44日間で「何のためにあるのか」と、社会人としての出発点を学びます。終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ①【違いがわかる】入社は入学と違い、「働いて価値を生み、その対価を得る」スタートだとわかる。
- ②【イメージできる】自分が「報酬を得るビジネスパーソン」になった、という立場の変化がわかる。
- ③【見通しがもてる】これからの44日間で何のための時間か、ざっくり見通せる。

0-1 今日という日について

今日から、あなたの職業人生が始まります。学生として過ごした時間は終わり、これからは自らの働きで価値を生み出し、その対価を得て生きていきます。

不安を感じるのは当然です。ただし、不安より先に自覚すべき事実があります。今日のあなたが持つ「新鮮な視点」「組織に染まっていない目」「学んだばかりの最新知識」は、**この先二度と手に入らない、今この瞬間だけの強みです**。これらは時間とともに必ず失われます。その価値を理解し、意識して使うこと。それが最初の課題です。

この44日間は、その強みを土台に、自律的に働くための基礎を体系的に固める期間です。

0-2 入社は、入学ではありません—あなたは今日から「ビジネスパーソン」です

最初に、いちばん強く刻んでほしいことがあります。**今日のこの日は、「入学」ではありません。「入社」です**。この2つは、似ているようで根本から違います。

学校では、あなたは**授業料を払って、教えてもらう側**でした。先生はあなたが学ぶために存在し、評価されるのは努力や成長そのもの。わからなければ教えてもらえ、失敗しても次に取り返せばよかった。

会社は違います。**会社は、あなたに報酬（毎月の給与や賞与）を払います**。あなたは、自分のスキル・時間・成果という価値を提供し、その対価としてお金を受け取る側になりました。つまり、**あなたは今日から、ひとりのビジネスパーソンです**。

言い換えれば、**あなたの今回の入社は、一人のビジネスパーソンとして、営利企業と「雇用契約」というビジネス契約を結んだ、ということです**。会社は、利益を生み出すために存在する営利組織です。その会社が対価を払ってまであなたと契約したのは、あなたが生み出す価値に期待しているからにほかなりません。

この会社は、あなたにとって最初のクライアント（取引相手）でもあり、立場の上下なく向き合う関係です。（労働契約の法的な意味は0-9・2-2で扱います。ここで掴んでほしいのは「意識」です。）

この違いを、頭ではなく体で理解できた人から、伸びていきます。**「教えてもらう」のを待つ人ではなく、「自分は何を提供できるか」を考えて動く人に、仕事も信頼も集まります**。今日、ここで

切り替えてください。**あなたはもう、学生ではありません。一人のプロフェッショナルとして、ビジネスの場に立っています。**

■ 「入社」という言葉の呪いを解く

「入社」という言葉に、惑わされてはいけません。この言葉は「入学」とよく似た音を持つために、知らないうちに「学校に入る」ような感覚をあなたに植えつけます。守ってもらえる、手取り足取り教えてもらえる、お客様のように扱ってもらえる。そんな受け身の構えを、言葉そのものが刷り込んでくるのです。**これを「入社という言葉の呪い」と呼びます。**今日、この呪いを解いてください。

はっきり言います。あなたは「会社に入れてもらった生徒」ではなく、会社と契約を結んだ一人のビジネスパーソンです。会社は、あなたが価値を提供する最初のビジネスパートナー。守られる側でも教わるだけの側でもなく、価値を生み出す当事者として、この場に立っています。

そして、ともに働くパートナーは、会社だけではありません。社長も、役員も、部長も、課長も、先輩も、同僚も、同期も、あなたが一緒に働くすべての人が、同じ目的に向かう仕事の仲間です。役割や責任の大きさは、もちろん人によって違います。指示を出す人と、指示を受ける人という役割の違いもあります（そのしくみは0-9・2-2で学びます）。しかし、一人の人としての立場は、誰が相手でも対等です。だからこそ、相手が誰であっても敬意をもって接し、自分から価値を提供する。

「お客様」として待つのではなく、「パートナー」として動く。これが、呪いを解いた人の姿です。

そして、「入社」と対になるもう一つの言葉に気をつけてください。「新卒」です。「新卒だから、できなくて当然」「新卒だから、まだ無理」。こうした言い方は、いつのまにか甘えや萎縮の言い訳になり、自分の意識を下げてしまいます。けれど**「新卒」は、あなたの実力や価値ではなく、ただの時期（タイミング）にすぎません。**あなたは今日から、一人のプロです。

この自覚こそが、これから学ぶマナーも、報連相も、チームづくりも、すべての土台になります。

■ 企業の側も、覚悟と敬意をもってあなたに向き合う

対等とは、片方だけが頑張ることではありません。**あなたを迎える企業の側にも、重い覚悟が求められます。**

新卒を迎えるということは、ただ労働力を一人雇うことではありません。その人のこれからの数十年、職業人としての歩みも、一人の人間としての生き方も、その出発点を預かるということです。だからこそ企業は、あなたを育てる責任を負います。仕事の意味を教え、必要な機会を用意し、つまづいたときに支え、あなたの成長に本気で投資する。この研修にこれだけの時間と人をかけているのも、その覚悟の表れです。

そして企業は、ともに価値を生むあなたに、敬意をもって向き合います。あなたの心と体の安全を守り、働きを正当に評価し、対話の場を持つ。これらは「してあげること」ではなく、人を預かる側の当然の責任です。法律（賃金の支払い、労働契約法第5条の安全配慮義務、労働時間の管理、ハラスメント防止、不当な差別の禁止など。第16章で学びます）は、その責任の最低ラインにすぎません。

同じことを、会社の側も戒めなければなりません。「新卒だから」は、任せない・侮る・雑に扱うことの言い訳になりがちで、相手の意識も、自分たちの育てる責任も下げてしまいます。「新卒」は時期にすぎず、あなたの価値を決めるものではありません。**会社は、あなたを初日から一人のプロとして、半人前ではなく一人前の相手として扱います。**

（迎える側へ）この資料は、新卒だけでなく、役員や管理職の方にも読んでいただきたいものです。最初の研修で何を語り、どんな背中を見せるかは、一人の若者の価値観の芯となって、長く残ります。新卒を一人前のビジネスパートナーとして敬意をもって扱い、その人生を預かる重みを忘れないこと。それは、育てる側に立つ人こそが、もっとも真剣に考えるべきことです。

あなたが価値を提供し、企業が覚悟と敬意をもって成長を支える。その両方がそろってはじめて、ほんとうのパートナーシップが成り立ちます。

0-3 最初の問い—あなたは今、何をしに来ましたか

「仕事をするために来た」。間違いではありませんが、解像度が足りません。同じ「働く」でも、自分をどの立場と捉えるかで、数年後の成長に決定的な差が生まれます。

あなたは今日から、ひとりのビジネスパーソンとして、この会社が推進するビジネスに参加しています。

「入社して社員になった」という受け身の表現と、「ビジネスパーソンとしてビジネスに参加した」という表現は、根本的に異なる意味を持っています。

会社の事業に貢献すること自体が、あなた自身の事業の始まりでもあります。

今日ここで**「自分はビジネスパーソンだ」という感覚を持って踏み出すことが**、この44日間のすべての土台になります。

0-4 あなたの仕事は社会につながっている

品質を重んじる組織では、こうした考え方が、新人にまず共有されます。たとえば、こう表現できます。「あなたが作ったものは社会に届く。その品質・安全・誠実さが、社会への責任である」。ITエンジニアも例外ではありません。ソフトウェアは多数の利用者へ同時に影響するぶん、責任は一層重いと考えるべきです。

あなたが書くコード・設計する仕組みは……	最終的に何に影響するか
Webサービスのログイン機能	何百万人もの人が毎日使うシステムの入口になります
金融システムの計算ロジック	人々の資産・給与・年金の計算に使われます
医療・介護システムの機能	医療現場での判断・患者の安全に関わります
ECサイトの決済処理	顧客の購買体験・企業の売上に直結します
業務管理システムの設計	会社のスタッフが毎日8時間使うインフラになります
セキュリティの実装	何万人もの個人情報・プライバシーを守ります

「動けばいい」「バグがなければいい」ではありません。求められるのは、必要な品質・安全・誠実さを満たしたうえで、価値を正しく社会へ届けることです。これがエンジニアの仕事の本質です。責任は重い一方、自分の仕事が社会を動かすという事実は、他の職業にはない誇りでもあります。

- ・ 「後工程（次の工程・最終ユーザー）はお客様」、自分の出力が次の人・最終的なユーザーへの入力になることを常に意識します
- ・ 品質・コスト・納期のすべてに当事者意識を持ちます
- ・ 「自分が作ったものが社会に届く」という誇りと責任感が、仕事の質を根本から変えます

💡 ポイント

プロフェッショナルな視座とは、一流のプロが最初に身につけるのは「自分が作るものに責任を持つ」という姿勢です。
技術力があるだけでなく、品質・安全・納期・コストを自分ごととして考え、改善を提案し続ける。
この研修を通じて、その視座を身につけることが目標です。

0-5 今日から変わること—学生とビジネスパーソンの違い

「学生から社会人になる」と言葉では言いますが、具体的に何が変わるのかを整理しておきましょう。

観点	学生時代	ビジネスパーソンとして
責任の範囲	自分の課題・提出物	チーム・顧客・社会への影響
時間の使い方	自分のスケジュールで管理	会社・顧客・チームとの約束で動く
失敗したとき	次に頑張ればいい	再発防止策を考えて提案する
お金の感覚	自分のお金・奨学金・バイト代	会社のお金・顧客が払ったお金
情報の扱い	自由に話してよい	機密・個人情報は厳重に管理する
評価される軸	努力・姿勢・成長	生み出した価値・成果・影響
「わからない」とき	先生・友人に聞く	上司に報告・相談し、自分で考えて提案する
完了の基準	「やった」「提出した」	「価値が届いた」「後工程が受け取れる状態になった」

この変化は一夜にして身につくものではありません。しかし「何が変わるのか」を理解した上で入社した人と、そうでない人とでは、6ヶ月後・1年後の姿が大きく変わります。

0-6 「正しさ」には4つの層がある

「正しく、よく働くとはどういうことか」、この問いへの答えを持つ人は強いです。どんな状況でも自分の軸を持って判断できるからです。

正しさには4つの層があります。この4層が、この研修全体を貫く軸になります。

「正しさ」の4層 — 判断が衝突したら、下の層（より根本）を優先する

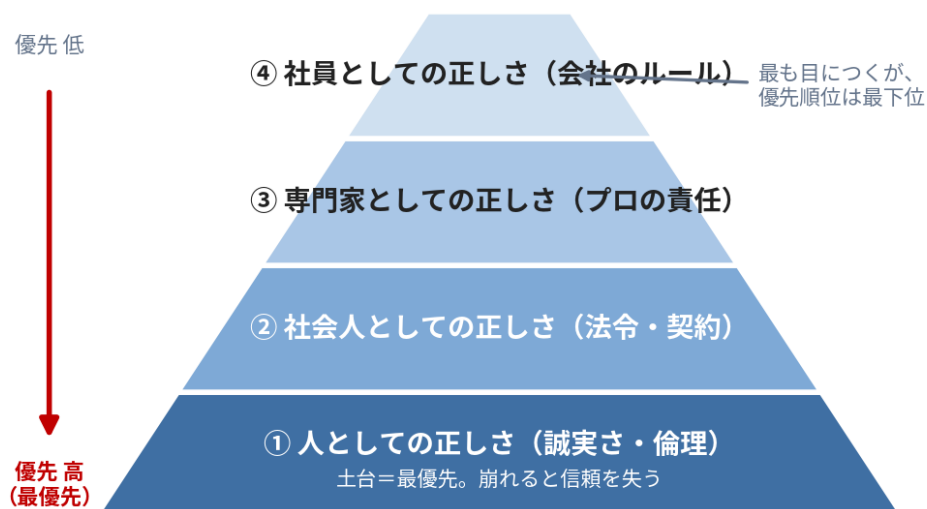


図 「正しさ」の4層（判断が衝突したら、下の層を優先する）

層	正しさの内容	崩れたとき何が起きるか
① 人としての正しさ	誠実さ・品格・倫理観、時代・文化・組織を超えた普遍的な価値	人として信頼されなくなります
② 社会人としての正しさ	法令・契約・社会規範を理解し、守ること	法的責任・社会的信頼の喪失につながります
③ 専門家としての正しさ	技術・知識を持つ者の責任と誇り	システム・社会・ユーザーへの被害が生まれます
④ 社員としての正しさ	組織のルールと社会の常識を両立させること	組織の秩序・外部からの信頼が失われます

この4層は互いに矛盾することがあります。上司の指示と自分の倫理観が衝突するとき、会社のルールと法律が食い違うとき、そのときにどう判断するかが、プロとしての真価を問われる場面です。

判断の優先順位は①→②→③→④の順です。「会社のルール（④・社員）」が最も目につく存在ですが、実は最も下位にあります。**「人としての正しさ（①）」が常に最優先です。**この軸があれば、どんな場面でも迷わず行動できます。

ここで、本テキスト全体を貫く一つの思想を示します。**技術力がどれほど高くても、倫理・ガバナンス・コンプライアンスを軽視するビジネスパーソンは、必ず問題を起こします。**優れたビジネスパーソンや技術者ほど、目先の技術よりも倫理・ガバナンス・コンプライアンスを重視し、そのための学習や教育を決して軽視しません。**ビジネスパーソンや技術者である前に、人として正しくあるとする、その姿勢こそが、すべての土台です。**

💡 この研修の読み方

この44日間は「知識を覚える」期間ではなく、「入社後の判断と行動を変える」ための準備です。

そしてこの研修自体が、あなたの最初の業務です。
 学生のように受け身で「教わる」のではなく、ビジネスパーソンとして主体的に「取りに行く」姿勢で臨んでください。
 各章には“勘違いの反証”“分岐ケース”“章末の自己採点”を置いています。
 読んで終わりにせず、「自分ならどうするか」を毎回考えながら進めること。
 それが、知識を現場で使える力に変えます。
 なお演習の多くには“評価の観点・達成基準”を添えています。
 基準が無い演習でも、「第三者が読んで同じ行動を再現できるか」を自分の合格ラインにして自己採点してください。
 また本書の「事例から学ぶ」のうち、企業名・人名のない事例は、要点を伝えるための学習用の例（創作・合成）です。
 実在の事件・出来事に触れる場合も、関係者が特定されないよう、社名や年などを伏せて一般化していることがあります。

0-7 なぜこの研修が存在するのか—44日間の意味

新卒研修は「会社のルールを覚える時間」ではありません。社会に出た瞬間から「判断を求められる場面」が始まります。この研修はその判断の質を高めるための投資です。そしてもう一つ、本テキストを貫く原則があります。**会社は、成長し続けなければなりません。停滞は、後退です。なぜなら、競争相手は常に成長しているからです。**会社にいる以上、あなたにも、自分の立場で会社の成長に貢献する義務があります。

■ 技術スキルだけでは足りない理由

ITエンジニアの仕事は「コードを書くこと」だけではありません。チームで働き、顧客と関わり、社会に影響するシステムを作る以上、技術力と同等かそれ以上に重要なことがあります。

- ・ コミュニケーション力：最高の技術も、チームに伝わらなければ価値を生みません。第4・6・7章で学びます
- ・ 倫理観と判断力：「技術的にできること」と「すべきこと」を区別する力、第10・12章で学びます
- ・ 法律知識：知らずに違反することを防ぐ最低限の知識、第11・14～15章で学びます
- ・ 長期的な視点：キャリア・ライフプラン・働く意味を問い続ける力、第18～20章で学びます

■ 「知っている」と「できる」の間にある深い溝

この研修を通じて何度も問われることになる最も重要な問いは、「知識が行動に変わっているか」です。

- ・ 知識レベル：「報連相が大切だとわかった」
- ・ 理解レベル：「なぜ報連相が大切か、その理由を人に説明できる」
- ・ 行動レベル：「実際に問題が起きたとき、すぐに上司に報告できた」
- ・ 習慣レベル：「考えなくても自然に報連相ができていく」

この研修が目指すのは「習慣レベル」への到達です。**知識を行動に変えるのは、最終的にあなた自身の意志と実践しかありません。**

0-8 この研修の全体像—44日間の旅

44日間の旅 — 今どこにいて、次に何が来るか

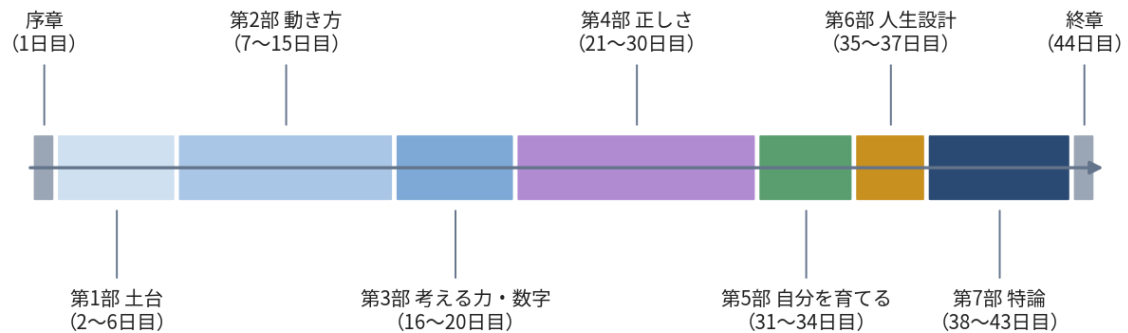


図 44日間のロードマップ (今どこにいて、次に何が来るか)

この研修は7つの部と序章・終章で構成されています。**それぞれが独立しているようで、すべて「正しく・よく働く」という一つの問いでつながっています。**

フェーズ	期間	内容	到達目標
序章	1日目	ビジネスパーソンとしての出発	「自分はビジネスに参加している」という自覚を持つ
第1部	2～6日目	社会人の自覚・組織理解・マナー	働く土台と組織の仕組みを理解する
第2部	7～15日目	報連相・仕事の進め方・コミュニケーション	日々の仕事の動き方を身につける
第3部	16～20日目	ロジカルシンキング・財務・数字	考える力と数字の感覚を養う
中間振り返り	20日目後半	前半の学びを統合	「知っていた」と「できた」の差を確認する
第4部	21～30日目	法律・倫理・ハラスメント・情報	正しく働くための知識と判断力を得る
第5部	31～34日目	成長・キャリア	成長し続ける姿勢とキャリアの方向性を定める
第6部	35～37日目	ライフプラン・仕事と人生の統合	長期的な視点で人生を設計する
第7部	38～43日目	特論：新卒社員向け ITエンジニア入門（あるべき姿・倫理・体系・品質・セキュリティ・伝える力・法律・生成AI・成長）	専門家としての誇りと実力の土台をつくる
終章	44日目	最終発表・行動指針の宣言	研修の学びを自分の言葉で語れるようになる

0-9 「使用者と労働者」—入社初日に知っておくべき関係性

「今日からあなたは社員です」、その言葉は、法的には「労働契約を結んだ」ことを意味します。
雇用関係の本質を理解することで、自分の権利と責任が明確になります。

■ 労働契約とは何か―対等な取引の始まり

- ・ 労働契約の本質：「労働者が労働を提供し、使用者がその対価として賃金を支払う」という対等な合意、一方的な支配関係ではありません
- ・ 書面での確認：入社時に受け取る「労働条件通知書」は、自分の労働条件を確認する重要な書類です。給与・労働時間・休日・業務内容を必ず確認します
- ・ 労働契約法の保護：使用者が一方的に労働条件を不利に変更することは原則として違法です
- ・ 就業規則：会社のルールブックです。入社後に必ず通読し、「知らなかった」という状況を防ぎましょう

■ 使用者と労働者の関係―「指揮命令」の正しい理解

会社員として働くとは、業務上の指揮命令に従うことを意味します。しかし、その範囲には限界があります。

- ・ 指揮命令に従う義務：業務上の指示・命令には原則として従う義務があります。しかし「違法な指示」には従う義務はありません
- ・ 人格の従属ではない：使用者は「業務の遂行」について指揮命令できますが、社員の人格・プライバシー・人権を侵害することは許されません
- ・ パワーハラスメントとの境界：適切な業務指導と違法なパワハラとの境界線は「業務上の必要性和相当性があるか」です。第12章で詳しく学びます
- ・ 労働者と使用者は、どちらも義務と権利を持つ、対等な関係です。整理すると、こうなります。

項目	労働者（あなた）	使用者（会社）
主な義務	誠実に労務を提供する ／適法な範囲で指揮命令に従う	賃金を支払う／安全に配慮する（労働契約法第5条）／労働時間を適正に管理する
主な権利	賃金・休日・有給・安全な環境を求める	適法な範囲で業務を指揮命令する
越えてはならない線	違法・非倫理的な指示には従わない	人格・人権・プライバシーは侵害できない

0-10 プロフェッショナリズムとは何か―入社初日から始まる問い

「プロフェッショナル」という言葉には様々な定義がありますが、この研修では一つのシンプルな定義を提示します。

プロフェッショナルとは：「自分の仕事が誰かの生活・社会に影響することを理解し、その責任を誠実に果たし続ける人」

■ 新卒1年目からプロである理由

- ・ 給与を受け取る日から：給与は「会社への価値提供の対価」です。プロとしての責任は入社初日から始まります
- ・ 顧客への影響：あなたが書いたコードの1行が、何万人ものユーザーに影響することがあります。規模は小さくても責任は本物です

- ・ チームへの影響：あなたの報告・コード品質・コミットメントが、チーム全体の生産性と信頼に影響します

そして会社の側も、あなたをそのプロとして遇します。お客様扱いで遠ざけるのではなく、責任のある仕事を任せ、必要な助けを用意する。任せることと支えること、その両方が、プロを育てます。

■ 「プロとして働く」ことの喜び

責任の大きさはばかり強調しましたが、プロとして働くことには深い喜びもあります。

- ・ 「自分の仕事が誰かの役に立った」という実感、これは職業人生を豊かにする最も基本的な喜びです
- ・ 「難しい問題を解決できた」という達成感、成長と挑戦の連続が、仕事を「ただの労働」から「生きがい」に変えます
- ・ 「信頼される」喜び、上司・同僚・顧客から「あなたなら任せられる」と言われる瞬間が、キャリアを前に進める原動力になります

📦 事例から学ぶ：新卒1年目のコードが100万人に影響した事例から学べること

大手IT企業の新卒エンジニアが担当したAPIの一部に軽微なバグがあり、リリース後に特定の条件下で100万人以上のユーザーに影響が出た事例があります。

本人は「自分のコードがそんなに大きな影響を持つとは思っていなかった」と語りました。

この教訓から学べること：規模の大小にかかわらず、ソフトウェアエンジニアの仕事は社会インフラの一部です。

「自分の仕事が誰かに影響する」という意識を持って働くことが、品質と責任感の源になります。

▶ やってみようーワーク・演習

序章 追加ワーク：「なぜエンジニアになったか」を言語化する

以下の問いに、正直に・深く向き合って書いてみましょう。

（正解はありません。自分の言葉で書くことが目的です）

- ①なぜITエンジニアになろうと思ったか、きっかけ・動機を書きましょう
 - ②エンジニアとして「作りたいもの・解決したい問題」を1つ書きましょう
 - ③「良いエンジニア」だと思える人は誰か（リアルでも架空でも）、その人のどこを尊敬するか
 - ④10年後、どんなエンジニアでありたいか、技術・人間性・社会への影響の3つの観点から
- この回答は、研修の各章で学ぶ内容と照らし合わせながら更新していきましょう。

0-11 44日後の自分へーこの研修に向き合う姿勢

最後に、この研修に向き合うための3つのお願いがあります。

お願い	意味
「知っている」と「できる」を区別する	「それは知っています」と思った瞬間が、最も危ない瞬間です。 知識が行動に変わって初めて「できる」になります

お願い	意味
疑問をその場で声に出す	「こんな質問をしたら恥ずかしい」は今日で終わりにしましょう。疑問は、理解が深まっている証拠です
自分のこととして受け取る	「自分には関係ない」「うちの会社には当てはまらない」と思った瞬間、学びは止まります。 すべてを自分ごととして受け取ってください

44日後、研修が終わったとき、今日書いた「大切にしたいこと」を読み返して、どう感じるかを楽しみにしてください。きっと、今日の自分には見えていなかったものが見えるようになっていくはずです。今日のあなたと、44日後のあなた。その差こそが、あなたの成長の証です。

▶ やってみようワーク・演習

今日の宣言：「自分が社会人・ビジネスパーソンとして大切にしたいこと」を書き出しましょう。

この紙は研修の最後（終章・44日目）に再び開きます。

今日の自分と44日後の自分を比べるための、この研修で最も大切な記録です。

書くときに考えてほしい3つの問い：

- ①自分がこれから社会に届けたいもの（価値・仕事・成果）は何か
- ②社会人・ビジネスパーソンとして、自分が大切にしたい「正しさ」は何か
- ③10年後、どんなビジネスパーソン・エンジニアになりたいか

第1部 土台をつくる（2～6日目・5日間）

第1部では、社会人・ビジネスパーソンとしての自覚、会社という組織の仕組み、ビジネスの現場での振る舞いという、最も基本的な土台を築きます。あなたは今、一人のビジネスパーソンとして、業務としてこの研修に臨んでいます。ここで扱うのは、どの会社・どの職種でも通用する、新卒ビジネスパーソンの必須教養です。土台が固まれば、以降のどの課題にも軸を持って臨めます。

最初の五日間は、目に見える成果を急ぐ時間ではありません。社会人としての構え、会社というしくみの見方、人と接する作法という、これから何年も使い続ける「型」を、ここで身につけます。一つひとつの意味を急いで覚えるより、腹に落とすことを優先してください。**土台は地味ですが、あとから必ず効いてきます。**

第1章 社会人・ビジネスパーソンとしての出発点（2～3日目・1.5日）

🎯 この章のゴール

この章では、学生から社会人への切りかえ、「学生と社会人の違い」、そして「指示待ち」から「自分から動く」へを学びます。

終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ①【**違いがわかる**】学生と社会人の違いがわかり、自分の言葉で一言と言える。
- ②【**イメージできる**】「指示を待つ」と「自分から動く」の違いがわかる。
- ③【**理由がわかる**】最初の30日を大事にする理由がわかる。

💡 まず自己診断—読む前に、いまの自分を知らう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「学生と社会人の違いを自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
- ② 「学生と社会人のいちばん大きな違いは何か」を、いまの考えで一言だけ書きます（正解は気にしません）。

読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

会社の仕組みを学ぶ前に、「自分の立場がどう変わったのか」「どのような意識で働くのか」を定義します。この基礎認識が定まらなければ、以降の知識は断片にとどまります。ここが固まれば、第2章以降の学びは一本の軸でつながります。

✦ よくある勘違いと、その反証

【勘違い】「社会人になる＝責任が重くなって自由が減るだけ」→

【反証】最大の違いは“評価のされ方”です。

学生は努力・過程が評価されますが、社会人は生み出した価値・結果で評価されます。

だから「やった」で終わらず、「価値が相手に届いたか」までを完了の基準にする、これが学生気分との分かれ目です。

1-1 学生と社会人の本質的な違い

■ 「やってみた」で終わらない—結果責任とは何か

学生時代は、努力や挑戦それ自体が評価されました。**社会人では、努力の結果として何を生み出したかが評価されます。**これを「結果責任」といいます。努力は前提にすぎず、評価の対象は相手に届いた成果です。この一点が、学生と社会人を分ける最も本質的な違いです。

同時に、会社や上司の側には、新人が成果を出せるよう適切な仕事と機会を与え、過程の努力も含めて公正に評価する責任があります。結果責任は、本人と、機会を整える組織の、双方で成り立ちます。

- ・ 学生：過程が評価されます

- ・ 社員：成果が評価されます。「なぜそれが必要か」を理解して行動することが求められます

「成果が評価される」とは、「成果が出なければ価値がない」という意味ではありません。新人のうちは、すぐに大きな成果は出ません。それでも、「成果につなげようと考えて動く」姿勢そのものが評価されます。結果と、結果に向かう姿勢、その両方を大事にしてください。

- ・ ミスをしたとき、「再発を防ぐために何をするか」が問われます。失敗を次の成長につなげるのがプロの姿勢です

ただし、誤解しないでください。「結果がすべて」と聞くと冷たく感じますが、努力が無駄になるわけではありません。努力は、結果につながって初めて報われる、それだけのこと。だから、がむしやに頑張るのではなく、「この頑張りは、何を生むか」を意識する。努力の“向き”を考えることが、社会人の第一歩です。

■ アルバイトと社員の本質的な違い

観点	アルバイト	社員
期待されること	決められた作業を正確にこなすこと	価値を生み出し、成長し続けること
問題が起きたとき	上司に報告して指示を待つ	原因を考えて再発防止策を提案する
指示のない時間	休む・待つ	次にやるべきことを自分で考えて動く
評価基準	時間あたりの正確な作業量	生み出した価値・成長の度合い

■ 時間・お金・情報の扱い方が根本的に変わります

- ・ 時間、業務時間は会社のために使う時間です。「会社の時間を預かっている」意識が仕事の質を高めます
- ・ お金、会社のお金は会社の資産です。経費は「必要な業務のため」に適切に使います
- ・ 情報、会社の情報は大切な資産です。顧客情報・技術情報・経営情報は適切に管理します

学生時代の「自分の時間・自分のお金」と、社会人の「会社から預かる時間・お金・情報」は、扱い方が根本的に変わります。会社の時間は、給料という形で“買われている”時間。だから、その時間で何を生むかに責任が生まれます。難しく考えず、「預かっているものは、丁寧に扱う」と思えば十分です。

■ ミスをしたとき・困ったときの正しい動き方

事例から学ぶ：「自分で直せば気づかれない」が招いた失敗（ありがちな失敗シーン）

本番リリース後、ある新人は自分の修正で一部ユーザーがログインできなくなっていることに気づきました。

「自分で直せば気づかれないかも」と一人で原因を探すうち、問い合わせは30件積み上がりました。最初に先輩へ伝えたのは1時間後。

先輩は「気づいた瞬間に一言くれれば、5分で切り戻せたのに」と言いました。

ミスそのものより、報告の遅れが信頼を削った典型です。

- ・ まず被害の拡大を止めます
- ・ 上司・関係者に事実を正確に報告します。正直に、早く伝えることが信頼につながります

- ・ 原因を特定します
- ・ 再発防止策を考えて提案します

問題は、時間の経過とともに対応の選択肢を失います。早期報告は被害を最小化する最善手であり、同時に「都合の悪い事実を隠さない人物」という信頼を築きます。

学生と社会人の違いは、ひとことで言えば「お金を払う側から、お金をもらう側になる」ことです。学生は授業料を払って、サービスを受ける立場でした。社会人は、給料をもらって、価値を提供する立場になります。この向きが180度変わる、ここが、すべての違いの出発点です。

✦ 考えてみよう：あなたは何を“提供”する？

あなたがこれから受け取る給料は、誰かが「その価値がある」と認めて払うお金です。あなたは、会社や社会に、どんな価値を提供できるでしょうか。

今すぐ立派な答えはいりません。「提供する側になった」という視点を、まず持ってみてください。

1-2 あなたはひとりのビジネスパーソンである

■ 入社とは「ビジネスの取引を始めること」です

序章で示した「ビジネスパーソンとしての参加」という意識を、具体的な構造として捉え直します。

この「参加する」という意識は、決して気負うものではありません。むしろ、「言われたからやる」より、「自分も組織の一員として、価値を出す側にいる」と思えたほうが、仕事は面白くなります。受け身でいるより、参加しているほうが、ずっと楽しいのです。

この「雇われた社員」と「ビジネスパーソン」の対比は、序章で示した対比表と同じ考え方です（繰り返しになるため表は割愛します）。要点は一つ、会社に所属して指示に従うのではなく、自分のリソース（スキル・時間・専門性）を提供して会社のビジネスに貢献する一人のプレイヤーである、ということ。この意識の差が、5年後・10年後の姿を大きく変えます。

- ・ 経営者のビジョン・戦略を理解すると、日々の仕事の「なぜ」が見えてきます
- ・ **会社はあなたという「リソース」に投資しています。給与・研修・設備・機会がそのコストです**
- ・ 会社があなたに期待しているのは「コストに見合った価値の提供」、さらには「コストを超える価値の創出」です

同時に、会社の側にも、その投資に見合うよう、新人が価値を発揮できる仕事・指導・環境を整える責任があります。価値の提供は、本人の努力と、それを引き出す組織の責任が噛み合って実現します。

■ 新卒の「強み」を正確に知る

新卒の強み	具体的な価値
先入観のなさ	「なぜそうするのか」を素直に問えます。組織の慣習を外の目で見られます
学習速度の速さ	新しい知識・スキルを吸収するコストが低いです。最新の教育を受けていま

新卒の強み	具体的な価値
	す
エネルギーとフレッシュな発想	既存の制約に縛られないアイデアを提案できます
デジタルネイティブの感覚	新しいツール・技術への適応が速いです。AIなどの活用感覚があります

これらの強みは、入社直後の今が最大値です。組織に順応するほど「会社のやり方」が当たり前になり、外部の視点は失われます。違和感や素朴な疑問は、その場で記録すること。半年後には、同じ違和感を持てなくなります。

💡 ポイント

「社員として雇われた」ではなく「ビジネスパーソンとして参加した」という意識の違いは、5年後・10年後の姿を大きく変えます。

1-3 「指示待ち」から「自分で考えて動く」への転換

- ・ 1年目後半からは：指示を待たず、次にやることを自分で考えて動きます
- ・ 2～3年目には：問題を自分で発見し、解決策を提案します
- ・ その先には：誰も気づいていない課題を発見し、チームを動かします
- ・ **「これが終わりました。次は〇〇をやろうと思いますが、いかがでしょうか」への転換が、信頼とより良い仕事へのチャンスを生み出します**

主体性とは、勝手に進めることではありません。「自分で考える」と「確認する・相談する」は、セットです。考えた上で相談すると、相手も答えやすく、あなたの考える力も伝わります。考えずに聞くのでも、聞かずに突っ走るのでもなく、その間（あいだ）を狙いましょう。

- ・ エンジニアにとっての主体性、「この仕様で本当にユーザーの課題が解決されるか」を考える姿勢が、エンジニアとしての価値を高めます

📖 事例から学ぶ：ある地方銀行の不正融資事件から学べること

「上司の指示に従い続けた」という弁明が通じなかった事例です。この教訓から学べること：組織の中でも個人として考え、判断する習慣を持つことが大切です。自分で考える力が、組織への真の貢献につながります。

✦ 考えてみよう—倫理の視点

「会社の常識」が世間の非常識になるとき、定期的に「これは社会から見て正しいか」と問い直す習慣を持つことが、長期的に信頼される社会人としての基盤になります。

📖 知っておこう—法律の紐づけ

労働契約法、入社時に結んだ雇用契約の意味を正確に理解してください。自分の労働条件・業務内容・給与を確認しておくことが、権利を守ることにつながります。

▶ やってみよう（所要20分） — 「自分の提供価値」を言語化する

目的：自分を「雇われた労働力」ではなく「価値を提供する当事者」として定義し直す。

①現時点で提供できるスキル・強みを3つ、行動レベルで具体的に書く（例：「コードが書ける」ではなく「Pythonで小規模なデータ集計スクリプトを一人で作れる」）。

②それぞれが自社のどの事業・どの工程で価値になりうるかを、1つずつ対応づける。

③「1年後、自分はどんな価値を会社と顧客に提供できる人材になっているか」を、具体的な成果物または数値の形で1文にする（例：「APIの設計から実装・テストまで一人で担当できる」）。

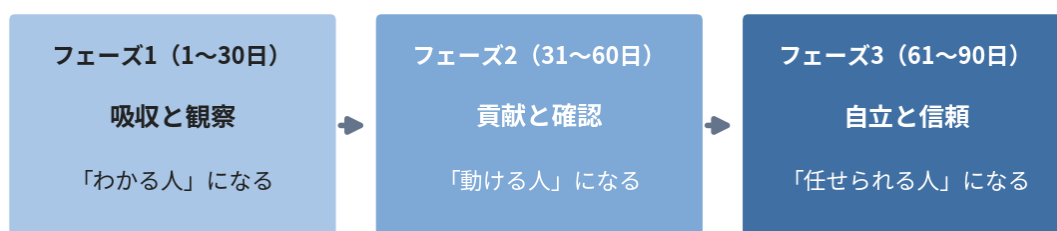
評価の観点：抽象語（がんばる・成長する）で終わっていないか／第三者が読んで具体的な行動を想像できるか。

「自分で考えて動く」と聞くと、ハードルが高く感じるかもしれません。でも、最初は小さくていいのです。「次はこれをやろうと思いますが、いいですか」と一言添える。指示を待つだけでなく、自分の考えを“添える”、その小さな積み重ねが、やがて「任せられる人」への道になります。

1-4 入社後「最初の90日間」を成功させる—新卒社員のスタートダッシュ

どんな優秀な人でも、入社後最初の90日（約3ヶ月）の過ごし方がその後数年間の評価を左右します。「最初の90日間」を意識して設計することが、スタートダッシュの鍵です。

入社後「最初の90日」— 焦らず、段階で力をつける



※ 最初の1ヶ月は「判断する前に観察する」。受け身に見えても、立派な仕事。

図 入社後「最初の90日」の3フェーズ

■ フェーズ1（1～30日目）：吸収と観察—「わかる人になる」

最初の1ヶ月は「判断する前に観察する」期間です。業界・会社・チームの文化・仕事の流れを正確に把握することに集中します。

- ・ 職場の人間関係を把握する：誰が何を担当しているか・誰がキーパーソンか・誰に相談すれば何が進むかを整理します

- ・ 会社・業界の基本情報を自分で調べる：自社のサービス・顧客・競合・業界トレンドを調べます。「そんなことも知らないの？」と言われる前に自分で学びます
- ・ 職場の「評価される行動」を観察する：積極的な発言が期待される職場と、静かに実行することが重視される職場があります。文化を読んでから行動します
- ・ 早めに「小さな成果」を出す：最初のアウトプットは「完璧」より「早い」が有利です。まず動かして見せることでフィードバックを得られます

観察というと受け身に聞こえますが、これは立派な“仕事”です。早とちりで動いて空回りするより、最初の1ヶ月で土地勘をつかむほうが、結果的に早く戦力になれます。「すぐに成果を」と焦らず、まずはよく見て、よく聞く時期だと割り切りましょう。

■ フェーズ2（31～60日目）：貢献と確認—「動ける人になる」

- ・ 自分から仕事を取りに行く：「何かお手伝いできることはありますか」の一声が信頼構築を加速させます
- ・ 「指示待ち」から「提案型」へ移行する：「〇〇をやるべきですか」より「〇〇をやろうと思っていますが、よいでしょうか」の形で確認します
- ・ 1on1を最大限活用する：「今の自分のパフォーマンスについてフィードバックをください」と積極的に求めます

あわせて、迎える側にも責任があります。新人が知らなくて当然のことを惜しまず共有して質問を歓迎し、求められなくても定期的にフィードバックを返すこと。立ち上がりの速さは、本人の主体性と、組織の支えの、両方で決まります。

- ・ 「わかったふり」をしない：不明点をそのままにすることが後から大きなミスにつながります

■ フェーズ3（61～90日目）：自立と信頼—「任せられる人になる」

- ・ 「終わったら報告」から「途中でも報告」へ：進捗を自発的に共有することで、上司の安心感が高まります
- ・ ミスの後の動きで評価が決まる：正直に報告・原因分析・再発防止策の提示を迅速に行うことで、むしろ信頼が上がります
- ・ 次の3ヶ月のビジョンを共有する：「次の3ヶ月でこんな貢献がしたい」という意欲の表明が、次の仕事の機会を引き寄せます

なぜ「最初の90日」がそれほど大切なのでしょう。それは、最初に作られた印象が、その後ずっと“その人らしさ”として扱われやすいからです。とはいえ、気負いすぎる必要はありません。完璧を見せるより、「素直に学び、誠実に動く」姿勢を見せること。それが、いちばん良い第一印象になります。

◆ 考えてみよう：最初の30日、何を観察する？

配属されたら、まず「観察」から始めます。誰が何を担当しているか、どう相談すれば物事が進むか、どんな行動が評価されているか。

あなたなら、最初の30日で“まず知っておきたいこと”を、何から挙げますか。

1-5 プロとしての自覚—「アマチュア」と「プロ」を分けるもの

「**プロフェッショナル**」とは役職ではなく**姿勢**です。どんな仕事でも「相手の期待を誠実に超え続ける人」がプロです。この意識を初日から持てるかどうか、長期的なキャリアを分けます。

プロの核心は、「相手目線」です。自分が「やった」かどうかではなく、相手が「助かった」かどうか。同じ作業でも、相手が何を求めているかを一步想像するだけで、仕事の質は変わります。新人のうちから、この“相手目線”を意識する人は、必ず伸びます。

■ 行動で示すプロとアマチュアの違い

アマチュアの思考・行動	プロの思考・行動
締め切りに間に合わせることを最終目標にする	締め切り3日前に「80%できた状態で確認する」を習慣にする
「できません」で終わらせる	「今の自分には難しいですが、〇〇なら対応できます」と代替案を出す
指示された以上のことはしない	指示の背景にある「本当のニーズ」を理解し、そこに応える
うまくいかないと環境・他者のせいにする	自分がコントロールできることに集中する
スキルが固まったら学ぶのをやめる	変化し続ける世界に合わせて学び続ける

■ 主体性と自走力を今日から育てる

- ・ 問いを持って出社する：「今日は〇〇を達成する」「〇〇について確認する」という意図を持って1日を始めます
- ・ 「なぜ」を問い続ける：「この仕事は何のためにあるのか」を常に考えます。目的がわかると成果の質が大きく変わります
- ・ 判断の精度を上げる：「この状況で自分ならどうするか」を考えてから上司に相談します。考えたプロセスが信頼につながります

完璧な判断ができなくても大丈夫です。大事なのは、「自分なりに考えてから相談する」という順番。考えた跡が見えれば、たとえ答えが違って、相手は「この人は伸びる」と感じます。

プロとアマチュアを分けるのは、才能ではなく「準備」と「姿勢」です。締め切りギリギリで滑り込むのではなく、余裕をもって仕上げる。「やりました」で終わらせず、「相手が本当に求めていたもの」まで考える。この差は、小さいようでいて、信頼の差として確実に積み上がっていきます。

1-6 職場で「絶対に避けるべき」言動—信頼を瞬時に壊すもの

良い評判は積み上げるのに時間がかかりますが、悪い評判は一瞬でつきます。入社後に「あの人は難しい」と思われると、挽回に大きなエネルギーが必要になります。

■ 入社後に気をつけるべき言動リスト

- ・ 時間を守らない：遅刻・締め切り破り・返信の遅れは、じわじわと信頼を削ります。「少し遅れる」という感覚は相手への時間の侵害です
- ・ 「聞いていません」という言い訳：確認不足・メモを取らなかった・共有資料を読まなかった、これらはすべて「自分の準備不足」です

もちろん、組織や伝える側にも、必要な情報を確実に届ける責任はあります。ただし、それを理由に自分の確認を怠ってよいわけではありません。「聞いていない」をなくすのは、受け手の確認と送り手の伝達の、両方で実現します。

- ・ できないことを「できます」と言う：後からできないことが判明する方が、最初から「難しいですが挑戦します。〇〇が不安なので確認させてください」と言うよりも大きなダメージになります
- ・ 悪口・陰口：職場での悪口は必ず当事者に伝わります。「あの人のことをこの人に言うなら、私のことも他の人に言うだろう」と思われます
- ・ 「まあいいか」で進める：「このバグはあとで直せばいい」「この確認は後回しでいい」という先送りが大きな問題の種になります
- ・ SNSへの軽率な投稿：職場の情報・愚痴・顧客が特定できる内容の投稿は、キャリアを壊す行動です。第15章で詳しく学びます

これらは、どれも「能力の問題」ではなく「姿勢の問題」です。だからこそ、誰でも今日から避けられます。難しいスキルは要りません。時間を守る、正直に言う、陰口を言わない、避けるべきことを避けるだけで、あなたの評価は守られます。

事例から学ぶ：入社1年目の姿勢が5年後を決めた事例から学べること

あるIT企業の新卒社員Aさんは入社後から「なぜこの仕事をするのか」を必ず確認し、進捗を自発的に共有し続けました。

ミスをしたときも正直に報告して再発防止策を出した結果、2年目から大型プロジェクトのメンバーに選ばれました。

一方、Bさんは技術的には優秀でしたが「言われたことだけやる」スタンスで、報告も最小限でした。

5年後、Aさんはチームリードを任せられ、Bさんはスキルは高いが「次のステップ」が見えない状況になりました。

この教訓から学べること：技術力とプロとしての行動の両方が長期的なキャリアを作ります。入社1年目の姿勢は数年後の評価の基盤になります。

ポイント

社会人としての第一歩は「できることを誠実にやり続けること」から始まります。最初から完璧な社会人である必要はありません。

わからないことを正直に言い、確認を怠らず、約束を守り、ミスしたら正直に報告する。この基本を守り続けることが、最速で信頼を積み上げる道です。

▶ やってみよう（所要30分） — 「最初の30日計画」を設計する

目的：配属直後の30日を、運任せにせず設計する。

【キーパーソン5名】名前・役割・「この人に確認すべきこと」を書き出す（誰に聞けば何が進むかの地図をつくる）。

【自力で調べる3項目】自社サービス・主要競合・業界動向のうち3つを、配属前に自分で調べて要点をまとめる。

【毎日の習慣を3つ宣言】例：日報を書く／翌日のタスクを前夜に整理する／質問を1つ用意して出社する。

達成基準：30日後、この計画のうち何を実行できたかを自己採点する（できなかった項目は、その理

由まで書く)。

やってみよう②、「アマチュア／プロ」の置き換え

学生時代やアルバイトでの自分の「アマチュア的だった行動」を1つ挙げ、同じ場面で「プロならどう動くか」を、具体的な行動として書き直す。

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・ 仕事や研修の課題を頼まれたら、手をつける前に「これは何のためか（目的）」を一度たしかめる。

💡 ポイント：AIの時代も、最後は“人柄”

技術やツールがどれだけ進歩しても、「この人と一緒に働きたい」と思われるかどうかは、人柄で決まります。

誠実さ、素直さ、約束を守ること、AIには代われない、あなたの土台です。スタートのこの時期に、その土台を丁寧に作っておきましょう。

✦ 考えてみよう：信頼を“貯金”する

信頼は、毎日の小さな行動で貯まる「貯金」のようなものです。今日からできる“信頼の貯金”を、一つ挙げてみましょう。

「返信を早くする」「約束の時間を守る」、小さなことで構いません。

1-7 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ 学生と社会人のいちばん大きな違いを、一言で。
- ・ 「指示を待つ人」と「自分から動く人」の違いを、一言で。
- ・ 最初の30日を大事にするのは、なぜか。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、これからの学びや仕事の中で、少しずつ埋めていきましょう。

第2章 会社という組織の中で働くとはどういうことか（3～4日目・1.5日）

🎯 この章のゴール

この章では、会社という組織の仕組み、会社は何で成り立ち、給料はどこから出るのかを学びます。終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ① **【仕組みがわかる】** 会社が利益・価値で成り立ち、給料の出どころがどこかわかる。
- ② **【考え方がわかる】** 役職は「偉い順」でなく役割の違い、とわかる。
- ③ **【知っている】** 自社の業態（SES／受託／自社開発）の違いを知っている。

💡 まず自己診断—読む前に、いまの自分を知ろう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「会社の仕組みを自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
- ② 「給料は、どこから出ていると思いますか？」を、いまの考えで一言だけ書きます（正解は気にしません）。

読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

「自分はビジネスパーソンとして参加している」という自覚ができれば、次は「参加している事業とはどういうものか」を理解します。会社・業界・組織の仕組みを知ることは、日々の仕事の意味を具体的に把握するための、ビジネスパーソンの必須教養です。

2-1 会社はなぜ存在するのか

■ 利益・価値・社会への貢献の三位一体

- ・ 利益、会社が継続的に存在するために必要な「酸素」。利益があるからこそ、社員を雇い続け、社会への貢献を続けられます
- ・ 価値、お客様・社会に提供する便益。価値のあるところに利益が生まれます
- ・ 社会への貢献、地域雇用・税収・インフラ・文化への貢献。会社は社会の中に存在します

■ 報酬はどこから来るのか

- ・ 売上（顧客が支払うお金）－コスト（材料費・人件費・経費等）＝利益
- ・ 報酬は「コスト」の一部、同時に、報酬を受け取る社員が「価値を生む」主体でもあります

報酬とは、あなたが生み出した価値に対して“得る”対価です。雇用契約では給与（毎月）や賞与として受け取りますが、本質は「価値の対価」。だから、まず価値を出すことが先で、報酬はその結果として決まります。

【報酬の順番を間違えない】「報酬の分しか働かない」と言う人がいますが、順番が逆です。先に価値を出すから報酬が決まる。新人のうちは、まず価値を出す側に立つことが、将来の報酬につながります。

- ・ 自分もたらす価値 > 自分にかかるコスト、**これが「会社貢献している」の意味です**

◆ 考えてみよう—倫理の視点

会社の利益と社会への責任が衝突したとき、利益と社会的責任は両立させることが持続可能な経営の条件です。

⚖️ 知っておこう—法律の紐づけ

会社法、株式会社の仕組み・株主総会・取締役会・従業員の位置づけ。自分の行動が会社の行為とみなされる場面を知っておくと、より責任ある判断ができるようになります。

2-2 「使用者」と「労働者」—会社の中での自分の立場を知る

会社という組織を理解する上で、真っ先に押さえておきたい法的な区分があります。「使用者」と「労働者」という2つの立場です。**同じ職場にしながら、立場によって担う義務と持つ権利がまったく異なります。**

立場	定義	主な義務・権利	今のあなたは
使用者側	会社・経営者・管理職（課長以上）	部下の権利を守る法的義務を負う	将来、役職が上がったとき
労働者	雇用契約のもとで働くすべての人	適切な労働条件・有給・安全な職場環境を求める権利を持つ	今日からここに属します

- ・ 新卒社員は今日から「権利を持つ労働者」です。残業代・有給取得・安全な職場環境を求める権利があります
- ・ 管理職になる瞬間、立場は「使用者側」に切り替わります。部下の権利を守る義務を担う側になります
- ・ この研修は両方の目線で読んでください。今日の権利を知ることと、将来の責任を知ることがつながっています

⚖️ 知っておこう—法律の紐づけ

労働基準法・労働契約法、使用者と労働者の関係を規律する法律体系です。第11章で詳しく学びますが、「自分がどちらの立場か」を知ることが、すべての出発点になります。

2-3 自社・業界を知る

■ 自社の歴史・事業・製品・サービスの概観

- ・ 自社の設立経緯と歴史、「なぜこの会社が存在するか」を知ること、仕事への使命感が生まれます
- ・ 現在の主要事業・製品・サービスのラインアップ
- ・ 自社の強みと差別化要因、競合と比べて何が違うのかを理解します

- ・ 現在の課題と経営が注力している方向性

■ 自らが属する業界の構造

- ・ 競合企業、直接競合と間接競合の違いを理解します
- ・ 取引先・パートナー企業、バリューチェーンの中での自社の位置を把握します
- ・ 顧客セグメント、誰のためにビジネスをしているかを理解します
- ・ 業界のトレンドと自社が向き合っている外部環境変化

▶ やってみようワーク・演習

やってみよう：自社のビジネスモデルをキャンバス（図）で描き、自分の言葉で説明してみましょう。

■ IT業界の「業態」と構造を、言葉にできるようにする

自らがIT業界のどこに立っているかを言葉にできると、自分のキャリアを自分で設計できます。最初に押さえておきたい3つの観点を紹介します。

- ・ 業態（SES／受託開発／自社開発）：SES（System Engineering Service）はエンジニアの労働力を契約期間で他社に提供する形で、客先常駐が多い働き方です。受託開発は顧客の依頼を受けて作り納品する形、自社開発は自社のサービス・プロダクトを作る形です。3つは働き方もキャリアパスも大きく異なります。
- ・ 多重下請け構造：日本のIT業界には、元請け→1次下請け→2次下請け…という構造が今も多く残っています。自社が「何次請け」かによって、技術選択の自由度・利益率・労働条件が変わります。
- ・ 開発手法（ウォーターフォール／アジャイル）：要件定義から順に進めるのがウォーターフォール、短いサイクルで作って試すのがアジャイル。自社がどちらを主に使うかで、日々の進め方が変わります。

これらを1年目で全部理解する必要はありません。「自社の業態を、3年後には自分の言葉で説明できる」を目標にしてください。それが、自分のキャリアを運転する第一歩です。

2-4 ステークホルダーを理解する

ステークホルダー — 会社は多くの関係者に支えられ、責任を負う

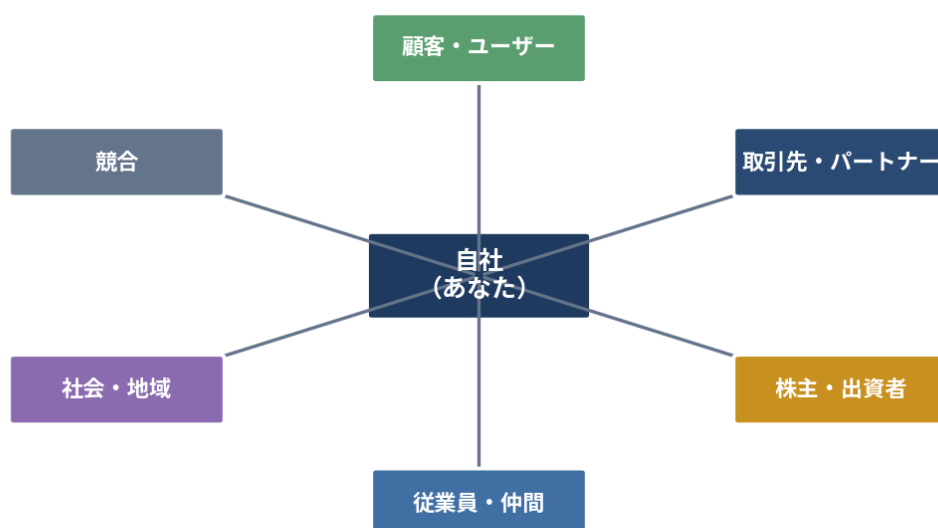


図 ステークホルダー・マップ

ステークホルダー	会社への期待	エンジニアとしての関わり方
顧客・ユーザー	高品質・使いやすい製品・サービス	ユーザーの課題を理解して設計・開発する
取引先・パートナー	信頼できる取引・情報の正確な共有	契約内容を守り、問題は早期に報告する
株主・投資家	持続的な成長・透明な情報開示	会社の価値向上につながる仕事をする
社員・同僚	成長できる環境・適切な評価	チームに貢献し、情報をオープンにする
地域社会	雇用・環境への配慮・地域への貢献	法令を守り、会社の評判を高める行動をする

- ・ エンジニアにとっての「顧客」、直接の発注者と最終ユーザーの両方を意識します
- ・ 「要件を出してきた人」の要求に応えることと「実際に使う人」に価値を届けることは、時に異なる視点が必要になります

◆ 考えてみよう—倫理の視点

組織の意思決定に自分がどう関わるか、「指示に従いながらも自分の判断を持つ」という姿勢が、健全な組織文化をつくれます。

2-5 上司・同僚・部下とはなにか―人間関係の本質を理解する

序章で確かめた「入社という言葉の呪い」を、ここでも思い出してください。役割は違っても、上司も、同僚も、後輩も、あなたと同じく価値を生むビジネスパートナーです。立場は対等です。役割の違いを理解したうえで、相手が誰であっても敬意をもって向き合う。それが、良い人間関係と良い仕事の土台になります。

■ 上司とはなにか

上司との関係	正しい理解	誤った理解
上司の役割	自分の仕事の方向性を示し、成長を支援し、権限の範囲を画定する人	命令に従わせる人（それだけではありません）
上司への報告	情報を届けることで、上司が適切な判断・支援ができるようにする行為	「怒られないようにする」ための報告
上司との関係	仕事上のパートナー。目標を共有し、互いに貢献し合う関係	上下の支配・服従関係

- ・ 上司も人間です。間違えることもあります。「上司が言ったから正しい」ではなく「なぜそう判断したか」を理解しようとする姿勢が大切です
- ・ 良い上司関係は「双方向の信頼」でつくられます。誠実に仕事をし、情報を正確に報告することが、上司との良い関係を育てます

■ 同僚とはなにか

同僚とは「同じ組織・チームで共通の目標に向けて働く仲間」です。友人とは違います。**友人は選べますが、同僚は選べません。**だからこそ、関係のつくり方が重要になります。

同僚の種類	特徴	関係のポイント
同期（同じ入社年次）	同じスタート地点から成長を共にする仲間	切磋琢磨しながらも協力し合える関係が理想です
先輩	経験・知識・文化を受け継ぐ立場の人	学ぶ姿勢を持ちつつ、自分の視点も大切にします
他部署の社員	異なる専門性・視点を持つ仲間	組織全体の目標のために協力し合う関係です
チームメンバー	日常業務を共に進める最も近い仲間	心理的安全性・情報共有・相互尊重が鍵です

■ 部下とはなにか

- ・ 部下を持つということは「相手の成長と成果に責任を持つ」ことです
- ・ 後輩・インターンと関わる時、今日の自分がいつか「上司」や「先輩」として誰かに振り返られます
- ・ 「教える」ことは「自分が最も深く理解する」機会でもあります

💡 ポイント

上司・同僚・部下はどれも「役割」であり「人格の優劣」ではありません。どの立場でも、人としての誠実さと敬意が関係の土台になります。

■ プロジェクトの中での役割（役職とは別の「仕事上の役割」）

プロジェクト内の役割	主な責任	新卒との関係
プロジェクトマネージャー (PM)	スコープ・コスト・スケジュール全体の管理	報連相の最終報告先の一つです
テックリード・アーキテクト	技術方針・設計の責任者・品質の全体感	技術判断の拠り所・相談相手になります
シニアエンジニア	実装・コードレビュー・後輩指導	最も近いメンター・レビューワーズ
ジュニアエンジニア（新卒）	タスクを預かり確実に・品質高く遂行	今のあなたの立場です

2-6 組織のマネジメント—分割統治・権限委譲・レポートラインを理解する

■ 組織の分割統治

分割の軸	例	特徴
機能別（職能別）	開発部・営業部・人事部・経理部	専門性を深めやすい。部門間の連携がボトルネックになりやすい
事業別（プロダクト別）	〇〇サービス事業部・△△プロダクト部	事業の独立性が高く、意思決定が速い
地域別	東日本事業部・西日本事業部	地域特性に対応しやすい
マトリクス（複合型）	事業部×機能部門が交差	柔軟性が高い。指示系統が複雑になりやすい

■ 権限委譲—「任せる」ことの本質と構造

権限委譲の要素	内容	なぜ重要か
権限 (Authority)	何を決めていいか・どこまでの決断を自分でできるか	範囲が不明確だと、すべてを上司にお伺いするか、逆に越権行為が起きます
責任 (Responsibility)	その決断・業務の結果に対して説明責任を持つこと	権限と責任はセットです
説明責任 (Accountability)	上位者・ステークホルダーに結果を報告・説明する義務	レポートラインを通じて組織全体の透明性を保ちます

- 権限委譲がうまくいくのは、上司が「目的・期待・制約」を明確に伝えて任せ、定期的な報告で進捗と問題を共有するとき。逆に「とりあえずやっておいて」「任せたら報告不要」という曖昧さは失敗のもとです。

■ レポートライン

レポートラインの形	特徴	注意点
シングルライン（一本線）	一人の上司にのみ報告する。指示系統が明確	上司が不在・異動のときに機能しにくくなる場合があります
マトリクス（二重ライン）	「機能上の上司」と「業務上の上司」が別に存在する	優先順位の競合が起きやすい
ドット（点線）ライン	直接の上司ではないが、一定の関係性を持つ上位者	アドバイス・情報共有の関係として整理されます

- 入社後まずやること：自分のレポートライン（直属の上司は誰か）を確認します
- 「誰に報告すればいいかわからない」は危険なシグナルです。早めを確認します**
- 一人のマネージャーが直接管理できる部下の人数には限界があります（一般的に5～8人程度）

■ 稟議・決裁・エスカレーション

- 稟議、重要な判断は所定の承認ルートを経ます。この流れを知ることで、組織の意思決定に正しく参加できます
- エスカレーション、**自分の判断を超える問題は早めに上位者に判断を仰ぎます**
- 「根回し」の本質、関係者全員が準備できた状態で意思決定を行うための合意形成文化です

✦ 考えてみよう—倫理の視点

「たぶん大丈夫」を超えて確認する習慣、根拠のある判断をすることが専門家としての誠実さです。確認・計測・検証の習慣が、自分とチームと顧客を守ります。

💡 ポイント

組織の仕組みは「制約」ではなく「インフラ」です。レポートラインや権限委譲の仕組みは、大勢の人が協力して価値を届けるための共通のインフラです。仕組みを理解して使いこなすことで、自分の仕事の速度と質が上がります。

▶ やってみよう—ワーク・演習

やってみよう①：自分のレポートライン（直属の上司・報告先）と、自分が判断している範囲（権限の境界）を確認し、書き出してみましょう
 やってみよう②：「自社のビジネスを5分で説明する」プレゼンを行い、相互フィードバックをしましょう

2-7 役職の地図―「役職は偉い順ではない」

役職は「偉い順」ではない― 見ている「先」（何年先を見て動くか）が違う

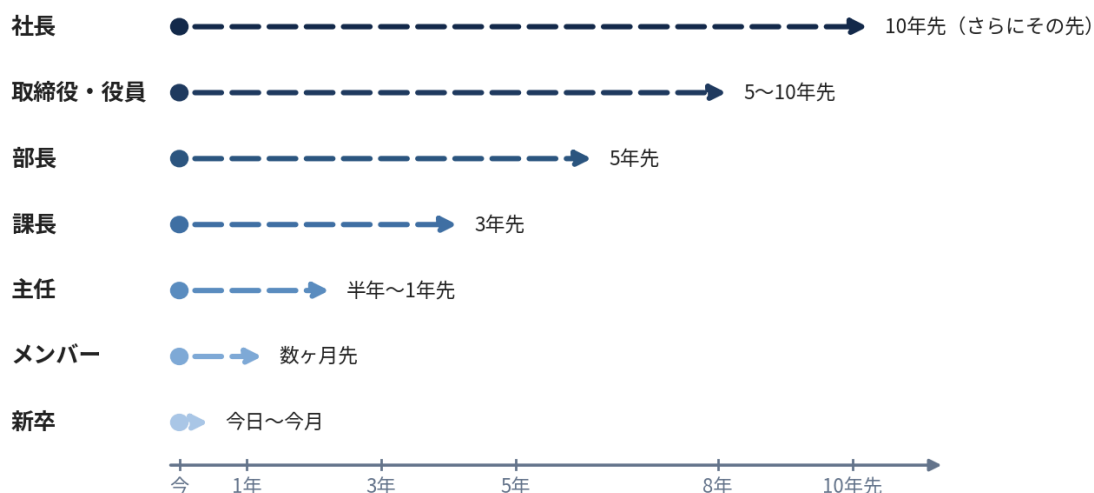


図 役職と「見ている先」（何年先を見て動くか）

配属されると、新卒研修期間とは違って、さまざまな役職を目にします。「メンバー」「主任」「課長」「部長」、「上に行くほど偉い人」というイメージを持ちがちです。しかし、最初に正しておきます。**役職は「偉い順」ではありません。**それぞれの役職は、見ているもの・背負っているもの・頭の使い方が違うだけです。（※呼称や役割は業態で異なります。ここでの呼び方は受託・SES型を基にしているため、自社開発・社内SE・組込みでは一致しないことがあります。本節は“考え方の地図”として読み、細部は配属後に自社の呼称で上書きしてください。役職名の暗記は不要で、“層ごとに視座が違う”という構造だけ掴めば十分です。）

なお、2-5で見た「PM・テックリード」などはプロジェクト上の“仕事の役割”であり、ここで扱う「主任・課長・部長」は会社の“役職”です。別々の軸であることを、頭の片隅に置いておきましょう。

そしてもうひとつ。上の役職の人は、下の役職を一通り通ってきています。現場の勘所もわかったうえで、より広い時間と責任で判断しているのです。「上はわかっていない」と決めつける前に、その視座の違いを思い出してみてください。

※以下は世の中の平均的な企業における一例です。職名や責任範囲は会社によって異なりますが、「概ねこういうイメージ」を持ったうえで、ひとつの物差しとして自分の会社を眺めてみてください。

■ 役職を読み解く「3つのモノサシ」

役職を理解するのに、便利な3つのモノサシがあります。

- ・ 見ている時間：どれくらい先の未来に責任を持っているか（今日のこと？ 3年後？ 10年後？）
- ・ 背負う責任の大きさ：自分の仕事だけか、チームか、事業全体か、会社全体か
- ・ 考えている「層」：手を動かす「実行」か、やり方を考える「戦術」か、力（人・お金・時間）の大きな配分を決める「作戦」か、そもそもどこで戦うかを定める「戦略」か

この3つを手がかりに、あなたの足元から順に、それぞれの役職が「何に責任を持ち、ふだん何をしているのか」を見ていきましょう。

役職	要は	時間軸	責任範囲	思考の層
新卒（あなた）	一人前になるために学ぶ見習い	半年～1年	自分の成長	実行
メンバー	自律的に動くプレイヤー	1年	自分の担当の成果	実行＋工夫
主任	アカウントを預かる現場監督（プレイングマネージャー）	2年	担当顧客の案件・継続受注・利益率	戦術を現場で実行
課長	新規ビジネスを開拓する事業経営者	3年	課の売上・利益・育成	戦術
部長	事業の今と未来を背負う事業経営総責任者	4年	事業全体の数字（赤字責任も）	作戦
役員	会社全体を担う会社経営者	5年	会社の経営（経営責任）	戦略と作戦の間
社長	未来を描く会社経営総責任者	10年以上	会社の最終責任	戦略

■ まず、あなた（新卒）―指示を受けて学ぶ見習い

入社したばかりのあなたの仕事は、ずばり「一人前になること」です。まだ会社の戦力として数えられる前の「準備運動」の期間。だからこそ、いちばん伸びる時期でもあります。

- ・ 責任：今はまだ会社の数字に直接の責任はありません。そのぶん、**学んで成長すること自体が、あなたの責任**です。
- ・ やること：指導役のもとで指示をこなし、わからないことは早めに聞く。報連相を習慣にする（相談はまず、いちばん近くの先輩へ）。
- ・ 見ている時間：半年～1年。考えている層：実行。
- ・ 新卒のいちばんの武器は、素直に聞けること、そして「これって何でこうなっているんだろう？」とまっさらな目で疑問を持てることです。

つまりポイント：新卒のいちばんの失敗は、わからないことを抱えたまま黙ってしまうこと。**聞くのは恥ではなく、新卒のうちは「聞くこと」そのものが仕事です。**

■ 一歩先：メンバー―自律的に動くプレイヤー

新卒の次のステップが「メンバー」です。あなたが一番近くで一緒に働く先輩の多くがここにいます。最大の違いは、**指示を待つのではなく、自律的に動くようになる**こと。教わる側から、自分でやり切る側へ変わります。

- ・ 責任：与えられた仕事を自分の力で最後までやりきる。さらに「自分の仕事が会社の利益にどうつながるか」を意識しはじめる。先輩ができたなら、その面倒を少し見るのもメンバーの役割です。
- ・ やること：与えられた仕事を工夫して改善し、「もっと少ない手間で良い成果を」と考える。先輩の面倒も少し見る。
- ・ 見ている時間：1年。考えている層：実行＋工夫。

つまりポイント：いつまでも指示待ちのまま、報告が「事後報告」になる人は、ここで足踏みします。**「自分で動く」と「勝手にやる」は違うので、迷ったら早めに相談を。**

■ その上：主任―アカウントを預かる現場監督

メンバーの次が「主任」です。課長が持つ事業経営の一部を、アカウント（顧客）単位で預かる「現場監督」です。

- ・ 責任：課長から任されたアカウント（担当顧客）の案件を確実にやり遂げる。これは課長の事業責任の一部を、権限ごと預かることを意味します。
- ・ やること：QCD（品質・費用・納期）を管理し、課長の方針をチームの動きに落とし込んでメンバーに指示する。
- ・ 見ている時間：2年。考えている層：戦術を現場の動きに落とす。

多くの主任は、自分でも手を動かしながらチームもまとめる「プレイングマネージャー」です。監督が選手を兼ねるようなもので、それゆえ「自分でやった方が早い」と抱え込み、人に任せられなくなる罠に陥りやすい立場でもあります。

つまりポイント：案件の遅れやトラブルでお客様の信頼を失うこと。そして、何でも自分で抱え込んでメンバーに任せられないと、ここでパンクします。「任せる」のも大事な技術です。

■ コラム：「労使」ってなに？

会社には大きく分けて2つの立場があります。

- ・ 労働者：会社に雇われて、指示を受けて働く側。新卒・メンバー・主任は、基本的にこちらです。
- ・ 使用者：会社の側に立って人を動かし、事業や数字に責任を持つ側（経営する側）。

この2つをまとめて「労使（ろうし）」と呼びます。

では、その境目はどこか。

ざっくり言うと、課長から上が「使用者＝会社側・経営する側」です。

主任までは現場で手を動かす側ですが、課長になると事業の数字に責任を持ち、人やお金の使い方を決める「会社側」の人間になります。

厳密には課長・部長も会社に雇われて報酬をもらう一人で、その意味では労働者でもあります。が、「立場の感覚」としては、課長から上は「会社を背負う側」に切り替わる、と覚えておくとうわかりやすいです（序章0-9・本章2-2の「使用者と労働者」もあわせて見てください）。

新卒のあなたが知っておくといのは、課長と話すときは「会社を背負っている人」と話している、ということ。

そう思って接すると、課長の言葉の重みや、ときに厳しい判断の理由も、ずっと理解できます。

■ さらに上：課長—新しいビジネスを開拓する事業経営者

主任の上が「課長」です。ここからはっきり「事業に責任を持つ人」＝事業経営者になります。主任の仕事を一通り通ったうえで、課の数字を背負います。

- ・ 責任：自分の課の数字（売上・利益）に責任を持つ。今年の達成も、3年後のチームの成長も担います。
- ・ やること：部長の計画を現場で実行できる「やり方（戦術）」に落とし込み、主任に任せて進める。同時に、部長の成長戦略を自分の課で形にする。
- ・ 見ている時間：3年。考えている層：戦術を組み立てる。

課長が席にいないのはサボりではなく、新しいビジネスの種を探して社外を回っているからです。継続受注は主任、新規の芽は課長が担います。

つまりポイント：目の前の案件に追われて、次の新しいビジネスの種まきが止まってしまうこと。「今」だけになると、事業はじわじわ細っていきます。

■ コラム：事業の「成長」って、何が伸びること？

会社は成長し続けなければなりません。

停滞は後退です。

競争相手が常に成長している以上、立ち止まることは相対的な後退を意味します。

この原則を深く理解しているマネージャーは、「成長」を最重要の価値観に置きます。

その「事業の成長」は、じつは中身が3つに分かれます。

- ・ ビジネスの成長：売上や利益が増えること（いちばんイメージしやすい「お金」の面）。
- ・ 組織の成長：人が増え、育ち、チームの仕組み（ルールや役割分担）が整っていくこと。
- ・ 技術の成長：会社が持つ技術の幅と深さが増すこと（新しい技術、特許、社外発信など）。

大事なのは、事業の成長はこの3つの**掛け算**だということ。

掛け算なので、どれか1つでもゼロに近いと事業全体が伸び悩みます。

だから良い課長・部長は、忙しくても「案件対応だけ」にならず、組織と技術の成長にも時間を割きます（この罫は第5章5-8で詳しく扱います）。

新卒のあなたも、自分の仕事がこの3つのどれに効いているかを意識すると視野が広がります。

案件を仕上げるのは「ビジネス」、後輩に教えたりやり方を改善するのは「組織」、新しい技術を学んで発信するのは「技術」への貢献です。

会社にいる以上、あなたには、自分の立場でこの3つの成長に貢献する義務があります。

■ 課長の上：部長—事業の今と未来に責任を持つ事業経営総責任者

課長が一つの「課」の事業経営者なら、部長は事業の今と未来のすべてに責任を持つ「事業経営総責任者」です。

- ・ 責任：事業全体の数字に責任を持つ。約束した成長を実現するため中長期のロードマップを描き、やり切る。赤字なら最終的な責任を負います。
- ・ やること：事業の目指す姿（ミッション・ビジョン）を示し、人・お金・時間をどこにどれだけ割くかの大きな計画（作戦）を描く。
- ・ 見ている時間：4年。考えている層：作戦を描く。

部長は後方の指示役ではなく、常に部の先頭に立ち、業界の集まりなど社外にも出て、評判や人脈を築きます。

つまずきポイント：事業の赤字。そして、数年先の手を打ち損ねて、事業がじわじわ細っていくこと。**「今の数字」と「未来の仕込み」を同時に回す難しさが、この立場の重さです。**

■ 経営の入口：取締役・執行役員—会社全体を担う会社経営者

部長のさらに上が、取締役や執行役員です。ここで責任の「種類」が変わります。部長・課長が「事業」に責任を持つ（事業責任）のに対して、役員や社長は「会社の経営そのもの」に責任を持ちます（経営責任）。

- ・ 責任：一つの事業だけでなく、会社全体の経営に責任を持つ。社長が描く長期ビジョンを、自分の担当領域（営業・開発・管理など）で実現させる。
- ・ やること：担当領域の大きな方針を決め、どこに会社の力を投資するか、どんなリスクに備えるかを判断する。経営会議で議論し、株主・取引先・社会といった社外に対して会社を代表して向き合う。
- ・ 見ている時間：5年。考えている層：戦略と作戦のあいだ。

なお「取締役」は株主から経営を任された人で、会社の重要事項を決める法的責任を負います（会社とは委任の関係）。執行役員は事業運営を担う役員です。

■ いちばん遠くを見る人：社長—未来を描く会社経営総責任者

そして、いちばん遠くを見ているのが社長です。役員までのすべてを通して来た人が、最後に会社全体の最終責任を引き受けます。

- ・ 責任：会社が10年後、その先もずっと続いて成長していくためのビジョンを描き、会社全体の経営に最終的な責任を持つ。うまくいかなかったとき、最後に責任を取るのも社長です。
- ・ やること：会社全体の方向を決め、投資やリスクへの備えを判断する。経営会議で議論し、社外に対して会社を代表する。
- ・ 見ている時間：10年以上。考えている層：戦略そのもの。

社長の決断には「正解の参考書」がありません。誰も答えを知らない問いに自分で答えを出し、結果を引き受ける、だから社長は、いちばん遠くを見ると同時に、いちばん重い孤独を引き受ける立場でもあります。

■ 全員に共通する3つのこと

- ・ 責任と権限はセット：役職が上がると、責任が重くなるぶん、決められること（権限）も増えます。今のあなたに大きな決定権がないのは、まだ大きな責任を負っていないから。一段ずつ、責任と権限と一緒に増やしていくのが健全な成長です。
- ・ 学び続ける（上ほど学ぶ量は増える）：役職が上がるほど扱う情報の幅とレベルが上がるので、学ぶ量はむしろ増えます。
- ・ ビジネスと数字に貢献する：会社は「売上－コスト＝利益」で動き、この利益から報酬も次の投資も出ています。役職に関係なく、全員が「自分が動かせる範囲で、どう会社の儲けに貢献できるか」を意識することが、強い会社をつくります。

そして、「上に行く＝管理職」だけが道ではありません。技術や専門性を磨く専門職としてステップアップする道も用意されています。

▶ やってみようワーク・演習

自社の役職を「3つのモノサシ」で分解してみましょう。

①自分の直属の上司は、どれくらい先の時間を見ていそうか（見ている時間）

②その人はどこまでの範囲に責任を持っていそうか（背負う責任）

③その人は「実行・戦術・作戦・戦略」のどの層で多くの時間を使っていそうか

わからなくて構いません。「一歩上の人は何を見て、何を考えているか」を想像すること自体が、視座を上げ、成長を早めます。

2-8 トップと現場のすれ違いは、なぜ起きるのか—視座を上げる

「うちの上司は現場をわかっていない」、どの会社でも聞きますが、「上司が悪い・現場が悪い」にすると解決しません。原因の多くは、立場による構造の違いです。

- ・ 時間軸の違い：現場は今日～今月を、経営層は3年・5年・10年先を見えています。同じ会議に同じ情報を持ち込んでも、そもそも別の問いに答えようとしています。

- ・ 役割の違い：トップの仕事の多くは「現場の外」（商談、業界の情報、将来への種まき）でしか起きません。サッカーの監督がピッチに立たず、全体を見て次に勝てる環境を整えるのと同じです。
- ・ 翻訳機能：トップと現場の間のミドル（部課長）の本来の役割は、伝言係ではなく双方向の翻訳者です。現場の問題を提案に変え、経営判断の背景を現場に伝えます。

新卒のあなたが今できるのは、上司や経営層の言動を、いったん立場の違いや構造として捉え直してみることです。次の3つの習慣が、視座（ものを見る高さ）を上げてくれます。

- ・ 一段上の会議に、聴衆として出させてもらう（「勉強のために聞いてもいいですか」）。報告を受けるのとはまったく違う景色が見えます。
- ・ トップの発言や行動の意図を、自分なりに解釈してみる。答えは出なくていい。問うこと自体が視座を上げる筋トレです。
- ・ 問題を上に持ち込む前に、一度構造化してみる。「こういう問題があります」ではなく「背景にはこういう構造があり、私はこう解決したいが、ここだけ相談したい」と整えてから持ち込む。これだけで対話の質が変わります。

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・ 自社が「何を売って」「だれの役に立っているか」を、一言で書いてみる。

2-9 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ 会社の利益はどこから生まれ、給料はどこから出るか。
- ・ 役職は「偉い順」か、それとも何の違いか。
- ・ 自社の業態を、ざっくり一言で。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、これからの学びや仕事の中で、少しずつ埋めていきましょう。

第3章 ビジネスマナーの基本（4～5日目・1.5日）

🎯 この章のゴール

この章では、ビジネスマナーの基本、マナーは何のためにあるのかを学びます。

終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ①【理由がわかる】マナーは「相手の負担と不安を減らす配慮」だとわかる。
- ②【知っている】対面・電話・オンラインの基本所作を知っている。
- ③【違いがわかる】敬語3種（尊敬・謙譲・丁寧）の違いがわかる。

💡 まず自己診断—読む前に、いまの自分を知ろう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「マナーの意味を自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
- ② 「マナーは、何のためにあると思いますか？」を、いまの考えで一言だけ書きます（正解は気にしません）。

読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

※ マナーや慣習は、時代や相手・場面によって変わります。本章の内容は一つの目安であり、鵜呑みにせず、実際の場面では職場や相手に合わせて、最新のあり方を常に確認するようにしてください。

✦ よくある勘違いと、その反証

【勘違い】「マナー＝細かい作法を暗記すること」→

【反証】マナーの本質は“相手の負担と不安を減らす配慮”です。

形を丸暗記するより、「これは相手にとってどうか」で判断できる人が、初対面でも信頼されます。形式は手段、目的は相手への敬意、この順番を押さえると、応用が利きます。

3-1 ビジネスマナーの本質—なぜマナーが必要か

マナーとは「相手への敬意と信頼を伝えるための手段」です。（就活やアルバイトで基本所作（お辞儀・5S・名刺の基礎）が身につけている人は、該当箇所は速習で構いません。“相手の負担と不安を減らす”という本質と、社外での名乗り・クレーム電話・オンライン会議など現場特有の場面に時間を使いましょう。）この意識を持つと、マナーは窮屈なものではなく自分を助けるものになります。

観点	ルール	マナー
性質	強制力のある規則	相手への配慮・文化的慣習
違反した場合	罰則・ペナルティがある	信頼を損なう・関係が悪化する

観点	ルール	マナー
目的	秩序の維持	関係性の構築・円滑な協力関係

ビジネスマナーを身につける本当の理由は、相手の時間・感情・立場を尊重することで信頼関係を素早く構築するためです。

◆ 考えてみようー倫理の視点

マナーを守るとは、相手の時間と感情を尊重することです。約束の時間を守ることは相手の時間を大切にすること。マナーの本質は、相手への思いやりです。

マナーを「細かい作法の暗記」だと思ってしまうと、苦しくなります。でも本質はシンプルで、「相手に敬意を伝え、相手を心地よくする」、それだけです。だから、所作を完璧に覚えていなくても、相手を思いやる気持ちがあれば、たいていは伝わります。形は後からついてきます。まず「相手を大切にしたい」という心が出発点です。

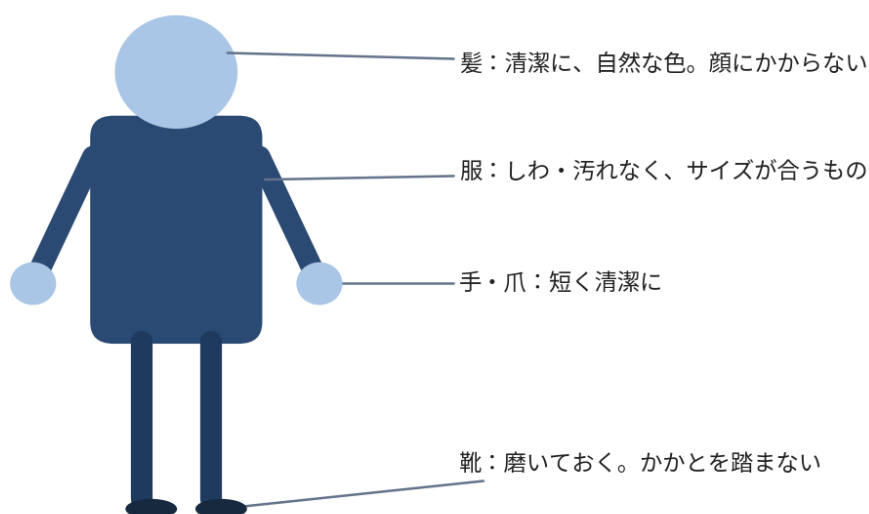
◆ 考えてみよう：されて嬉しかったマナー

これまでに、誰かの振る舞いで「丁寧だな」「気持ちいいな」と感じた経験はありませんか。挨拶、言葉遣い、ちょっとした気遣い、その“された側”の記憶が、あなたがどんなマナーを大切にしたいかのヒントになります。

3-2 身だしなみと職場環境

■ 身だしなみの基準

身だしなみのチェックポイント



「おしゃれ」は自分のため、「身だしなみ」は相手のため。基準は“清潔感”。

図 身だしなみのチェックポイント

- ・ 清潔感、汚れ・シワ・においがいいこと。これが最優先です
- ・ 機能性、仕事の動きを妨げないこと
- ・ 調和、職場の雰囲気・顧客層に合わせる

■ 自分のまわりを整える

自分の机やまわりを清潔・整然と保つことも、いっしょに働く人への基本的な配慮です。なお、整理整頓を体系化した「5S（整理・整頓・清掃・清潔・躰）」は、仕事の進め方・業務改善として第5章で学びます。

3-3 対面のマナー

■ 挨拶の基本

- ・ 積極的な挨拶が、職場全体のコミュニケーションを活性化します

挨拶は、マナーの中でいちばん簡単で、いちばん効果が大きいものです。特別なスキルは要りません。自分から、明るく、相手の目を見て、たったこれだけで、「感じのいい人」という印象が生まれます。迷ったら、まず挨拶。これが対人関係の、最強の一手です。

- ・ 挨拶は、気づいた人から先に（迷ったら自分から声をかけると好印象です）
- ・ 挨拶のポイント：相手の目を見る・笑顔・聞こえる声の大きさ

■ お辞儀の使い分け

種類	角度	使う場面
会釈	15度	廊下ですれ違う・エレベーターで乗り合わせる
敬礼	30度	一般的なビジネス挨拶・お礼・謝罪
最敬礼	45度	深いお礼・深い謝罪・重要な来客をお迎えするとき

■ 名刺交換

- ・ 渡すとき：両手で持ち、相手を読める向きに。「〇〇社の△△と申します」と名乗りながら渡します
- ・ 受け取るとき：両手で受け取り、相手の名前・会社名を一読してから「頂戴いたします」
- ・ テーブルでの扱い：名刺入れの上に置きます。会議が終わるまでテーブルに置いておきます

■ 来客対応・訪問マナー

- ・ 来客対応：笑顔・迅速な対応・適切な敬語・丁寧な案内
- ・ 席次：入口から遠い席・景色の良い席が「上座」です。上座にお客様を案内します

席次の早見図－①が上座、番号が大きいほど下座

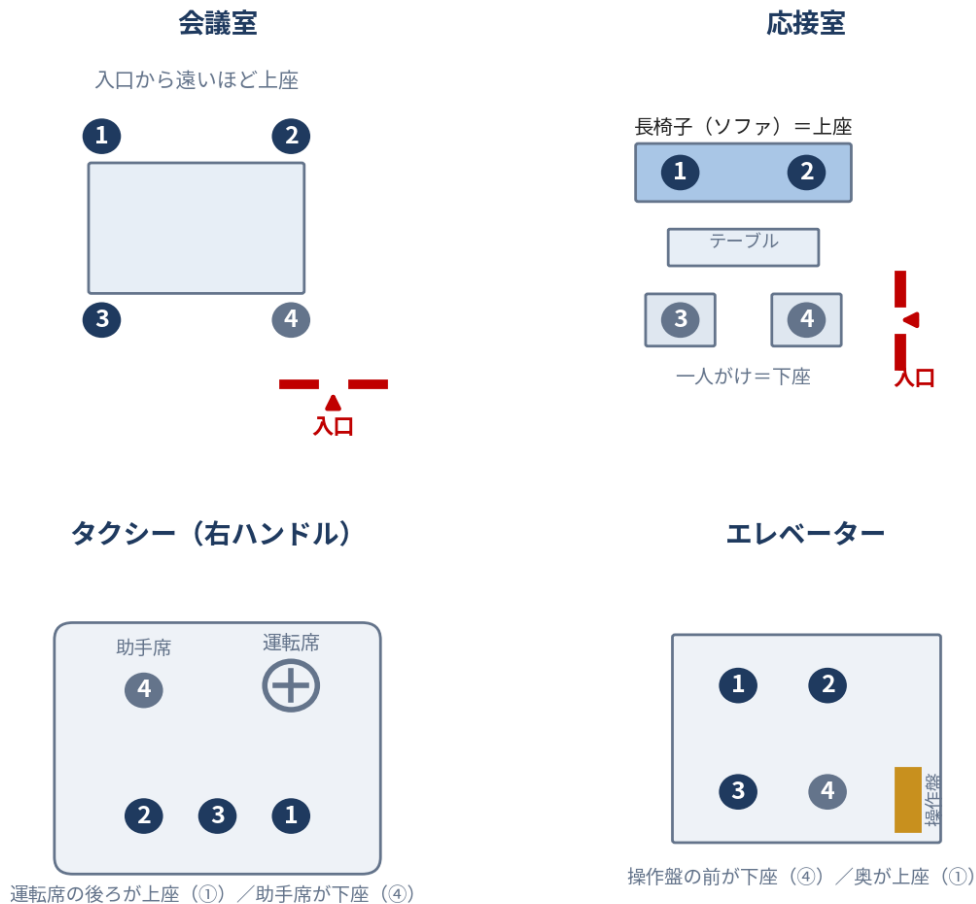


図 席次の早見図

- ・ 訪問時：約束の5分前到着・建物に入る前に身だしなみを確認します

■ 電話対応

電話の第一声で会社の印象が決まります。丁寧な対応の積み重ねが、会社全体の信頼を育てます。

- ・ 受けるとき：2コール以内に出ます。「お電話ありがとうございます。〇〇会社〇〇部の△△でございます」
- ・ 取り次ぎ：「少々お待ちください」→保留→担当者に取り次ぎます
- ・ クレーム電話：まず「ご不便をおかけしまして申し訳ございません」と謝罪→事実確認→対応策の提示

対面の所作は、最初は誰でも緊張します。名刺交換で手が震えても、お辞儀の角度を迷っても、大丈夫。完璧にこなすことより、「丁寧にやろうとしている」姿勢が、ちゃんと相手に伝わります。場数を踏めば自然にできるようになるので、最初から100点を目指さず、まずはやってみることが大切です。

3-4 テレワーク・リモート環境でのマナー

■ オンライン会議でのマナー

- ・ カメラ：基本的にはカメラをONにします。顔が見えることで信頼感・集中度が上がります

- ・ 発言のタイミング：「〇〇ですが、よいですか」と一言添えると自然に発言できます
- ・ チャット・リアクション機能：賛成・質問・挙手などを積極的に活用します

■ チャットツールの使い方

- ・ 件名を明確に、「〇〇の件で確認があります（3分で済みます）」と目的を添えることで相手が準備できます
- ・ 情報は記録に残す、口頭で伝えた重要な内容は、後からチャットでも共有します

■ 「見えない環境」での信頼のつくり方

- ・ 成果物と報告で存在感を示す、こまめな進捗共有が、チームへの安心感につながります
- ・ レスポンスの速さ、「**確認しました。〇時までに回答します**」という一言が信頼を積み上げます

▶ やってみようワーク・演習

やってみよう：以下の場面を実際に練習しましょう（複数シナリオ）

- ①名刺交換（一対一・複数人）
- ②電話対応（取り次ぎ・伝言・クレーム）
- ③来客対応（受付・案内・席次）
- ④オンライン会議（参加・発言・議事録共有）

オンラインでは、対面より情報が少なくなります。表情や空気が伝わりにくいぶん、ちょっとした気配りが効きます。カメラ越しでも笑顔を意識する、相手が話し終わってから話す、チャットで一言フォローする、画面の向こうにも人がいる、という当たり前を忘れないことが、リモート時代のいちばんのマナーです。

3-5 敬語の使い方—正しく・自然に・臆せず

敬語は「難しいから使えない」と諦める必要はありません。基本の3種類を正しく理解し、よく使う場面のパターンを覚えれば、自然に使えるようになります。

■ 敬語の3種類—整理すると難しくない

敬語の早見表 — 同じ動作でも相手/自分で言い換える

動作	尊敬語（相手を高める）	謙譲語（自分を低める）	丁寧語
言う	おっしゃる	申す／申し上げる	言います
行く・来る	いらっしゃる	伺う／参る	行きます
見る	ご覧になる	拝見する	見ます
する	なさる	いたす	します
食べる	召し上がる	いただく	食べます
いる	いらっしゃる	おる	います

図 敬語の早見表

- ・ 尊敬語：相手や相手に関係する行為・物を高めて表現する、「いらっしゃる」「おっしゃる」「ご覧になる」「召し上がる」「お持ちになる」
- ・ 謙譲語：自分や自分側の行為・物をへりくだって表現する、「おる・参る」「申す」「拝見する」「いただく」「伺う」「頂戴する」
- ・ 丁寧語：丁寧な言葉遣いで表現する、「です」「ます」「ございます」。日常的に最も使う

■ よくある敬語の間違いと正しい表現

× NG表現	✓ 正しい表現と理由
「了解しました」	「承知いたしました」「かしこまりました」、目上の人への「了解」は失礼にあたります
「ご苦労様です」	「お疲れ様です」、「ご苦労様」は目上から目下へ使う表現です
「なるほどですね」	「おっしゃる通りです」「ご指摘の通りです」、「なるほど」は評価的なニュアンスを含み、目上の方には不適切な場合があります
「よかったでしようか」	「よろしいでしょうか」、過去形の多用は不自然です
「お客様がお越しになりました」	「お客様がお越しになりました」、「なる」に「れる」を重ねる二重敬語は避けます
「部長の田中さん（社外で）」	「部長の田中」、社外では自社の人間の敬称（さん・様）は省きます

■ 場面別・敬語フレーズ集

以下は職場でよく使うフレーズの一覧です。声に出して練習することで、自然に口から出るようになります。

場面	フレーズ例
依頼するとき	「〇〇をお願いしますでしょうか」「ご確認いただけますと幸いです」「ご都合のよいときに〇〇をしていただけますか」
断るとき	「あいにく〇〇の都合がございまして」「せっかくお声がけいただきましたが」「今回は見送らせていただきます」
謝罪するとき	「大変ご迷惑をおかけしました」「誠に申し訳ございません」「以後、このようなことのないよう努めます」
確認するとき	「〇〇についてお伺いしてもよろしいでしょうか」「ご確認いただけましたでしょうか」「念のため確認させてください」
受け取るとき	「頂戴いたします」「ありがたく頂戴します」「おそれ入ります」

敬語は、「完璧に使えないと恥ずかしい」と気負わなくて大丈夫です。最初は、丁寧語（です・ます）がきちんと使えれば十分。尊敬語・謙譲語は、よく使うフレーズから少しずつ慣れていけば、自然に身につきます。多少間違えても、丁寧に話そうとする姿勢があれば、相手は悪い印象を持ちません。

3-6 社内コミュニケーションの作法—上司・先輩との日常の関わり方

入社後の毎日は「社内の人との関係構築」の連続です。**敬語や挨拶だけでなく、日常の働き方の中でどう振る舞うかが信頼関係を決めます。**

■ 上司・先輩との関わり方—基本的な姿勢

- ・ 話しかけるタイミング：集中しているとき・電話中は避けます。「今、少しよいですか（1分ほど）」と一言確認してから話しかけます
- ・ 指示を受けたとき：必ずメモを取り、「〇〇を〇日までに〇〇の方法でやればよいでしょうか」と復唱確認します。後から「聞いていません」という状況を防ぎます
- ・ わからないことを聞くとき：「自分なりに調べましたが、〇〇のところが理解できませんでした。〇〇という解釈で合っていますか」、「調べてから聞く」姿勢が信頼につながります
- ・ 指摘・叱られたとき：言い訳をせずにまず「申し訳ありませんでした」。その後で背景説明が必要なら「お時間があるときに経緯をご説明させてください」
- ・ 意見・アイデアを伝えるとき：「提案があるのですが、お時間よろしいでしょうか」と前置きしてから話します。自分の考えを「現状の問題→自分の提案→期待できる効果」の順で伝えます

■ 同僚・同期との関わり方

- ・ 困っているときに声をかける：「何か手伝えることがありますか」の一言が職場全体の雰囲気を変えます
- ・ 頼まれたことは誠実に：「やります」と言ったことは必ずやる、小さな約束の積み重ねが信頼です
- ・ 感謝を言葉にする：「ありがとうございました」「助かりました」を具体的に言います。「〇〇の件でフォローしてくれてありがとう、おかげで〇〇ができました」
- ・ 成功・失敗をチームで分かち合う：誰かの成功はチームの成功。誰かの失敗は一人で背負わない

日々の関わりで大切なのは、特別なことではなく、「報告・相談を早めにする」「お礼をきちんと言う」といった当たり前の積み重ねです。小さな誠実さが、毎日少しずつ信頼を育てます。

3-7 社外コミュニケーション—顧客・取引先との関わり方

社外の方との関わり方は、自分個人だけでなく「会社の代表」として振る舞うことを意識します。
第一印象は取り戻せません。丁寧・誠実・迅速の3点を常に意識しましょう。

■ お客様・取引先の方との初対面

場面	行動指針
訪問時の基本	約束の5～10分前に到着します。ビルに入る前に携帯電話をマナーモードにし、身だしなみを確認します
受付での対応	「〇〇社の△△と申します。〇時に〇〇部の××様とお約束をしております」と明確に名乗ります
待合室での振る舞い	スマートフォンを見ず、姿勢よく座ります。資料を見直す時間に使うのは良いです
名刺交換の順番	目下（訪問側・役職が下）から先に差し出します。両手で持ち、相手が読める向きに。受け取ったらすぐに相手の名前・会社名を確認します

■ ビジネスの場での言葉遣いのポイント

- ・ クッション言葉を使う：「お手数ですが」「恐れ入りますが」「差し支えなければ」、依頼・断り・質問の前に添えることで、柔らかい印象になります
- ・ 断るときは代替案を：「それはできません」で終わらず「今回は難しいのですが、〇〇という形ではいかがでしょうか」と代替案を提示します
- ・ Yes/Noを明確に：あいまいな返答は相手を混乱させます。できないことは「できません」と明確に伝えることが誠実さです

社外の人と接するとき、あなたは「あなた個人」とであると同時に「会社の代表」として見られます。これはプレッシャーに感じるかもしれませんが、裏を返せば、あなたの丁寧な対応一つが、会社全体の信頼を高めるということです。難しく考えず、「自分の振る舞いが会社の顔になる」と意識するだけで、自然と所作は整っていきます。

事例から学ぶ：第一印象が採用・契約を左右した事例から学べること

ある企業で、技術スキルが高い候補者が面接での言葉遣い（「了解です」「なるほどですね」の多用）と姿勢の悪さが理由で不採用になった事例があります。

また、取引先との打ち合わせで名刺をテーブルに忘れていったことをきっかけに関係が悪化したケースも報告されています。

この教訓から学べること：技術スキルと同様に、ビジネスマナーは「自分の価値を伝える技術」です。

マナーが良いだけで能力が高く見えることもありますが、逆にマナーのミスで能力が低く見られてしまうこともあります。

💡 ポイント

ビジネスマナーは「暗記するもの」ではなく「相手への思いやりを形にしたもの」です。形式を覚えることも大切ですが、「相手はどう感じるだろうか」という視点を持てば、知らない場面でも自然に正しい行動ができるようになります。

完璧でなくていいです。

誠実に・丁寧に・迅速に、この3点を軸に毎日実践しましょう。

▶ やってみようワーク・演習

やってみよう追加①：敬語変換クイズ

以下の言葉を正しい敬語に直しましょう。

- ① 「見ました」→ 目上の人に伝える場合
- ② 「言いました」→ 目上の方が言った場合に報告するとき
- ③ 「来ます」→ 自分が相手のところへ行く場合
- ④ 「食べましたか」→ 目上の方へ聞く場合
- ⑤ 「わかりました」→ 上司・顧客に対して

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・ あいさつ・お辞儀・名乗りを、声に出して一度やってみる。

3-8 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ マナーは、何のためにあるのか。
- ・ 対面・電話・オンラインの基本所作を、思い出せるだけ。
- ・ 敬語3種の違いを、ざっくり。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、これからの学びや仕事の中で、少しずつ埋めていきましょう。

第1部 総合ワークショップ（6日目・1日）

第1～3章の学びを統合する実践演習の日です。「知識として知っていること」と「実際にできること」の差を体感し、次の成長への手がかりを見つけることが目的です。

第1部（第1～3章）で学んだ「社会人の自覚・組織理解・ビジネスマナー」を統合する実践演習です。知識として「知っていること」と「実際に行動できること」の差を体感し、入社直後の職場での振る舞いに直接つながるスキルを磨きます。

■ 演習①：ロールプレイリアルな職場場면을体験する（午前：3時間）

グループに分かれて以下の場면을ロールプレイします。一人が複数の役割を演じ、終了後に全員でフィードバックします。

- ・ 場面A：初日の自己紹介と名刺交換、4人グループ（新入社員2人・先輩・上司役）で、挨拶・名刺交換・席次の確認を実践します
- ・ 場面B：電話対応（取り次ぎ・伝言・クレーム）、電話を受けて担当者に取り次ぐ・担当者が不在の場合の伝言・お客様からのクレーム電話の3種類をそれぞれ練習します
- ・ 場面C：上司への報告・相談、「タスクが期日に間に合わない可能性がある」という遅延報告と、「仕様書を理解できない部分がある」という相談をPREP法・四点セットを使って練習します
- ・ 場面D：会議への参加、ファシリテーター役・参加者役に分かれ、15分の模擬会議（アジェンダあり・議事録担当あり）を実施します

■ 演習②：ケーススタディ討議―「あなたならどうする？」（午後前半：2時間）

実際の職場で起きうるシナリオを提示し、グループで「正しい判断と行動」を議論します。正解を出すことより、「どう考えたか」のプロセスを言語化することが目的です。

- ・ ケース1：入社2週間で先輩から「このバグはリリース後に修正すればいい。報告しなくていい」と言われた、どう対応するか
- ・ ケース2：上司が会議中、顧客からの電話が入った。「急ぎではないから後で折り返す」と言われたが、顧客は「急ぎ」と言っている、どう判断するか
- ・ ケース3：同期の社員がSNSに「今日の打ち合わせで〇〇社が〇〇に困っているらしい」と投稿しているのを発見した、どう行動するか

■ 演習③：「自分の成長の証拠」を語る発表（午後後半：1時間）

個人ワークとして、以下の問いに答えるミニ発表（一人2分）を行います。

- ・ 第1部の学びの中で「最も自分に刺さったこと」を1つ選び、「なぜ刺さったか」と「明日からどう行動を変えるか」を話します
- ・ フィードバックの観点：「具体的か」「自分の言葉で語れているか」「行動が明確か」

▶ やってみようワーク・演習

▶ 第1部総合ワークショップ 振り返り

ワークショップ終了後、以下の問いに個人で答えましょう。

- ①今日のロールプレイで「思っていたよりできた」ことは何か
- ②今日のロールプレイで「もっと練習が必要」と感じたことは何か

- ③ケーススタディで「自分の判断が揺らいだ場面」はどこか、その理由は何か
- ④第1部全体を通じて「最も大切だと思ったこと」を1文で書く

▶ やってみようーワーク・演習

やってみよう①：自社のビジネスを新入社員同士でプレゼンし合いましょう（5分×人数）
やってみよう②：「社会人・ビジネスパーソンとして自分が大切にしたいこと」を宣言しましょう
ロールプレイ：入社初日の一連の場面を通してやってみましょう（挨拶→自己紹介→来客対応→電話対応→名刺交換）

第2部 動き方を身につける（7～15日目・9日間）

第2部では、報連相・仕事の進め方・コミュニケーション・文書作成という、ビジネスパーソンが毎日使う基本動作を体系的に習得します。これらは才能ではなく技術です。新卒のうちに身につけるかどうか、その後の信頼の積み上がり方を分けます。

報連相も、仕事の受け方も、人との関わりも、最初はぎこちなくて当たり前です。大切なのは、うまくやることなく、正しい型を知って、毎日くり返すこと。型が身についた人は、半年後には「安心して任せられる人」に変わっています。**これらは才能ではなく、誰でも身につけられる技術です。**

第4章 報告・連絡・相談—情報を正しく流す（7日目・1日）

🎯 この章のゴール

この章では、報告・連絡・相談（報連相）、情報を正しく流すことを学びます。

終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ①【違いがわかる】報告・連絡・相談の違いと「完了」がいつかわかる。
- ②【コツがわかる】結論を先に伝える（PREP）考え方がわかる。
- ③【理由がわかる】悪い情報ほど早く伝える理由がわかる。

💡 まず自己診断—読む前に、いまの自分を知ろう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「報連相のコツを自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
 - ② 「報告は、何をした時点で「完了」だと思いますか？」を、いまの考えで一言だけ書きます（正解は気にしません）。
- 読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

4-1 報連相の本質

「報連相」は社会人の常識として語られますが、それが「技術」であることを理解することで、本来の意味で使えるようになります。

✦ よくある勘違いと、その反証

【勘違い】「報連相＝報告すれば、自分の責任は果たした」

【反証】**報連相のゴールは“伝えること”ではなく、“相手が次に動ける状態をつくること”です。**

あなたが完璧に報告しても、相手が判断・行動できなければ、報連相は完了していません。だから「誰に・いつ・何を・どの粒度で」を相手起点で選ぶ、これが、報連相を“常識”から“技術”に変える分かれ目です。

■ 報連相の目的を理解する

- ・ 報告、仕事の結果・進捗を上司に伝えることで、組織としての確に動ける状態をつくります
- ・ 連絡、情報を共有することで、チーム全員が同じ状況を把握できます
- ・ 相談、判断や意見を求めることで、一人では解決できない問題を組織の力で解決します

■ 「悪い情報ほど早く」—早期報告が問題を小さくします

トラブルが発生したとき、早めに報告することで取れる対応策の選択肢が増えます。「もう少し自分で」と抱え込むほど打てる手は減り、気づいたときには上司にも手の打ちようがない状態になります。早い報告だけが、選択肢を残します。

- ・ 「まだ確認中」でも速報を入れることが大切です。「現在〇〇が発生。詳細は確認中」と伝えるだけでも価値があります
- ・ 早い一報だけが「任せられる人」をつくります。報告が遅い人は、能力ではなく姿勢を疑われます

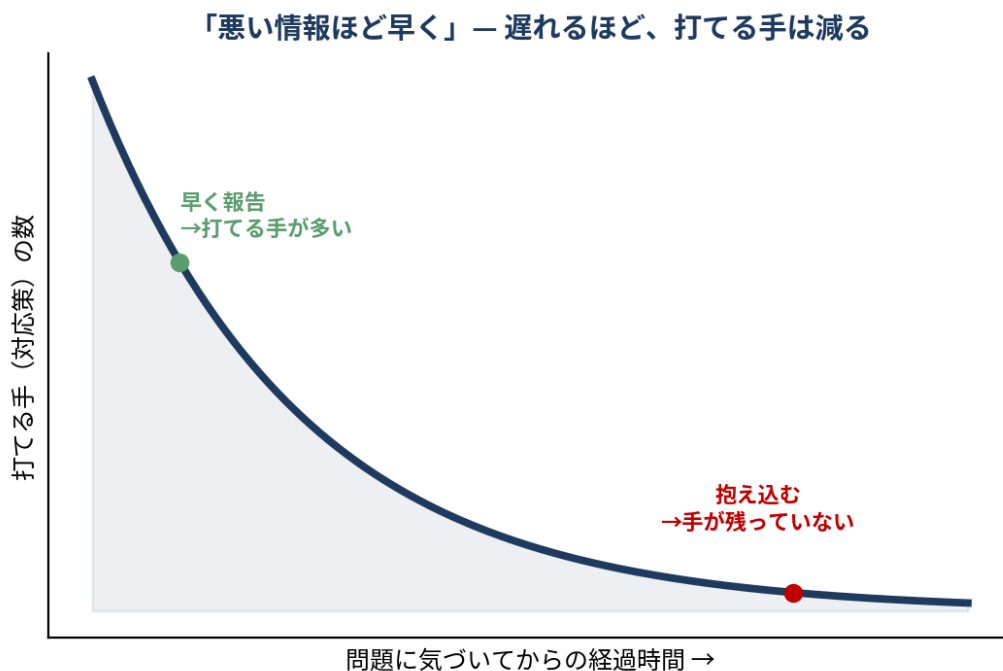


図 早く報告するほど、打てる手が多い

同時に、上司や組織の側には、悪い報告ほど歓迎し、報告したこと自体を評価して、報告者を責めない環境をつくる責任があります。早い報告は、報告する勇気と、それを安心して受け止める組織の、両方があって成り立ちます。

■ 上司が本当に求めている報告とは何か

- ・ 結論→理由→詳細の順番、「〇〇は完了しました。理由は〇〇です」
- ・ 事実と意見・推測を分ける、「〇〇が起きました（事実）。原因はおそらく〇〇だと思います（推測）。対応策として〇〇を提案します（意見）」
- ・ 次のアクションを提示する、「報告して終わり」ではなく「こう対応します」まで持っていく

4-2 報連相の実践

■ 報告の型—結論から話す「PREP法」

PREP	意味	例
P : Point (結論)	最初に結論を言う	〇〇プロジェクトの開発が本日完了しました
R : Reason (理由)	なぜそうなったかを説明	当初の予定通りに作業が進み、テストも通過しました
E : Example (具体例)	具体的な事実・データ	テスト件数120件、合格率100%です

PREP	意味	例
P：Point（再度結論）	まとめる	明日の予定通りリリースできます

■ 相談の型—四点セット

- ・ 問題：何が起きているか（事実）
- ・ 状況：現在どういう状態か（背景）
- ・ 自分の考え：自分はこう思う・こうしたいと考えています（自分の判断）
- ・ お願いしたいこと：何を決めてほしいか（依頼事項）

「**自分の考えを持った上で判断を仰ぐ**」ことが、相談の本質です。この形で相談することで、上司との信頼関係が自然と深まります。

■ エンジニアにとっての報連相—Slackを使った実践

💡 発展トピック：エンジニア特有の報連相の実践

エンジニアの現場では、メール・口頭だけでなく、Slackをはじめとするチャットツールがプロジェクト管理（PM）の中心になっています。ツールを正しく使うことで、報連相の質と速度が大きく上がります。

□ Slackを使った報告・連絡

- ・ チャンネル設計が報連相の質を決めます。#dev-general（全体共有）・#dev-incident（障害速報）・#dev-review（レビュー依頼）など目的別に使い分けます
- ・ 障害・トラブルの速報：まず事実だけを即投稿します
- ・ 例：「【速報】本番環境でAPIのレスポンスタイムが急増しています。現在調査中です。詳細は30分以内に報告します」
- ・ 「調査中だから報告しない」のではなく、まず「事実と状況」を共有することで、チーム全体が早期に動けます
- ・ 進捗報告のテンプレートを使うと書きやすくなります
- ・ 例：「【進捗】〇〇機能の実装 完了：ログイン画面のUI 進行中：APIとの結合テスト 
ブロッカー：認証APIの仕様が未確定（PMに確認中）」

□ Slackを使った相談

- ・ 相談する前に「自分の考え」を持ってから投稿します。「わからないので教えてください」より「〇〇と考えましたが、△△の場合が不明です。□□という解釈で合っていますか？」
- ・ スレッドを使って会話を整理します。メインチャンネルを散らかさず、関連する議論は同一スレッドにまとめます
- ・ 相談のテンプレート：
- ・ 例：「【相談】〇〇の実装方針について / 【状況】〇〇を実装中。△△という問題が発生しました / 【自分の考え】A案：〇〇（メリット△△、デメリット□□）B案：〇〇 / 【お願い】A案で進めてよいでしょうか」

□ デイリースタンドアップとSlack

- ・ 対面またはSlackのハドルで毎日15分以内で実施します。「昨日やったこと」「今日やること」「ブロッカー（詰まっていること）」の3点だけ共有します
- ・ ブロッカーは「詰まってから3時間以内」に共有するルールを持つと、チーム全体の生産性が上がります
- ・ Slackの絵文字リアクション (✅ 👁 🙏) で「確認した」「対応します」を素早く伝える文化をつくれます

□ 記録として残すことの重要性

- ・ Slackのやり取りは、口頭会話と違って「記録」として残ります。後から「言った・言わない」にならないための最大の防止策です
- ・ 口頭で決まったことはSlackに投稿して記録に残します。「先ほど〇〇さんとの会話で、××を△△方針で進めることになりました」

📖 事例から学ぶ：ある自動車メーカーの燃費不正事件（2016年）から学べること

「報告しない文化」が組織全体の不正につながった構造を示す事例です。
この教訓から学べること：情報を早期に共有することで、組織全体の問題解決能力が高まります。
悪い情報ほど早く正確に伝えることが、組織の健全性を守ります。

✦ 考えてみよう—倫理の視点

不都合な事実をどう伝えるか、正直な報告が長期的な信頼を生みます。早めに伝えることで、組織として最善の対応が取れるようになります。

⚖️ 知っておこう—法律の紐づけ

刑法（背任）、意図的な情報隠蔽は、場合によっては法的責任につながりうる場合があります。正直な情報共有が自分と組織の両方を守ります。

▶ やってみよう—ワーク・演習

やってみよう：トラブル発生時の報告・相談を練習しましょう（複数シナリオ） シナリオ①：本番環境でバグが発生した。

どう報告するか シナリオ②：作業が遅延しており、期日に間に合わない可能性があります。

どう報告・相談するか シナリオ③：上司の指示に疑問を感じます。

どう相談するか

4-3 報告の場面別実践—状況に応じた使い分け

報告は「何かが終わったら伝える」というものだけではありません。仕事の状況によって、報告の

内容・タイミング・方法を使い分けることが大切です。報告の場面を4つに整理します。

▶ 分岐ケース—あなたならどうする？

状況：金曜16時。来週月曜が締切のタスクが、7割は終わったが、残り3割で見たことのないエラーに当たった。土日は休み。

A：月曜の朝イチで上司に報告・相談する（今は自分で調べてみる）

B：金曜のうちに「7割完了／残りでこのエラー、月曜午前に相談したい」と一報入れておく
両方の帰結を考えましょう。

Aは、もし月曜に解決できなければ締切当日にすべてが露見し、上司は打ち手がありません。

Bは、上司が「土日の間に詳しい人に聞いておく」「締切を半日ずらす」など先回りできます。

判断軸は“自分が楽かどうか”ではなく、“相手の打ち手を最大化するのはどちらか”。

迷ったらB。

■ ①完了報告—仕事が終わったことを伝える

最も基本的な報告です。ただし「終わりました」だけでは不十分です。「何を・どのレベルで・何のために終わらせたか」を添えることで、上司が次の判断をしやすくなります。

- ・ 型：「〇〇が完了しました。品質確認済みです。次のステップとして〇〇を進めてよいでしょうか」
- ・ 注意：「完了」の基準を自分なりに持つことが重要です。「コードを書いた」ではなく「動作確認・テストまで完了した」が本当の完了です
- ・ エンジニアの実践：PRを出したとき、「〇〇機能のPRを作成しました。ローカルで動作確認済みです。レビューをお願いしますか」

■ ②中間報告—進行中でも定期的に状況を伝える

長いタスクや複数日にわたる仕事では、上司・チームが「今どこにいるか」を把握できるよう、完了を待たずに定期的に報告します。中間報告があることで、問題の早期発見と軌道修正ができます。

- ・ タイミング：作業の3分の1、半分、3分の2のタイミングで報告するのが目安です
- ・ 内容：「現在〇〇まで完了しました（全体の約50%）。予定通り〇日までに完了できる見込みです。現時点での懸念点は〇〇です」
- ・ 80%ルール：8割完成した段階で一度見せる、方向性のズレを早期に発見できます

■ ③遅延報告—期日に間に合わないと感じたとき

最も勇気がある報告ですが、最も価値のある報告でもあります。問題を抱えたまま期日を過ぎてしまうより、気づいた時点で早めに報告することで、組織として対応策が取れます。

- ・ 大原則：「遅れそう」と気づいた瞬間に報告する、気づいてから報告まで24時間以内を目安にします
- ・ 型：「〇〇の件ですが、現在〇〇の状況で、〇日の期日に間に合わない可能性があります。原因は〇〇です。対応策として①〇〇または②〇〇を考えています。どちらを進めるかご判断をいただけますか」
- ・ 絶対にやってはいけないこと：「なんとかなるかも」と黙って期日を過ぎること、取り返しのつかない信頼の損失につながります

▶ なぞって覚える—遅延報告メール（完成例→穴埋め→白紙）

【STEP1 完成例】「〇〇機能の件、ご報告です。

テスト工程で想定外の不具合が見つかり、水曜納期に半日ほど遅れる見込みです（原因：外部APIの仕様変更）。

対応案は①金曜朝まで延長して品質優先、

②水曜に既知の制約付きでリリース、の2案です。

①を推奨しますが、ご判断いただけますか。」

【STEP2 穴埋め】「____の件、ご報告です。

____で____が見つかり、____納期に____遅れる見込みです（原因：____）。

対応案は①____、

②____です。

____を推奨しますが、ご判断いただけますか。」

← 自分の担当業務に置き換えて埋めましょう。

【STEP3 白紙】テンプレを見ずに、自分のタスクで遅延報告を一から書いてみましょう。

型：「結論（遅れる事実）→いつ・どれくらい→原因→対応案2つ→推奨と依頼」。この順番が、“責められる報告”を“相談できる報告”に変えます。

■ ④異常・トラブル報告—想定外の問題が発生したとき

本番環境の障害・ミス・情報漏洩の疑いなど、緊急性の高い問題は、詳細が不明でも「速報」を最優先します。完璧な情報が揃うのを待つ必要はありません。

- ・ 速報の型：「【緊急報告】〇〇で問題が発生しています。現在調査中です。詳細は30分以内に報告します」
- ・ 続報の型：「先ほどの件の続報です。原因は〇〇と特定されました。影響範囲は〇〇です。現在〇〇で対応中です」
- ・ 復旧後の型：「〇〇が復旧しました。根本原因・対応内容・再発防止策については本日中に報告書を提出します」

4-4 連絡の技術—情報を「届ける」から「使われる形で届ける」へ

「連絡」は一見簡単に見えますが、実は最もスキルの差が出る報連相です。情報を「発信した」だけでは連絡ではありません。**相手が「受け取り、理解し、必要な行動を取れる状態」になって初めて連絡が完了します。**

■ 連絡の5原則

原則	内容
①必要な人全員に届ける	「知っているはずだ」「誰かが伝えるだろう」という思い込みが情報伝達の穴をつくりま
②適切な手段を選ぶ	緊急度・重要度・対象者によって、口頭・チャット・メール・会議の使い分けが必要です
③内容を簡潔に整理する	「何を・いつ・誰が・どうする」の4点を必ず含めます
④記録に残す	口頭で伝えた重要事項は、後からチャットやメールで記録します

原則	内容
⑤受け取りを確認する	大切な連絡は「確認しました」の返事を促します。特にSlackでは「確認の絵文字リアクション」をつける文化を育てましょう

緊急度 × 重要度で「連絡手段」を選ぶ



図 緊急度×重要度で連絡手段を選ぶ

■ 連絡ツールの使い分け

ツールの使い分けを誤ると、重要な情報が埋もれたり、相手の時間を不必要に奪ったりします。以下は一般的な基準です。

手段	適する場面・内容
口頭（対面・音声）	感情を含む内容、関係構築、複雑な話し合い、緊急度が非常に高い場合
チャット（Slack等）	素早い確認・連絡、進捗共有、非同期でよい情報。記録が残ることを活かします
メール	正式な依頼・報告、外部向けの連絡、後から参照が必要な内容
会議	複数人での意思決定、認識合わせが必要な場面、ブレインストーミング
注意	情報の重要度が高いほど、記録性の高いツールを選ぶ、重要な決定は口頭だけでなくチャット・メールで追記します

4-5 相談の技術—「迷惑にならないか」という不安を超えて

新卒社員に多いのが「こんなことを聞いて迷惑じゃないか」「自分で解決できないと思われないか」という心理的なブレーキです。しかし、**相談をためらって一人で詰まり続けた数時間は、あなただけでなくチーム全員の時間とプロジェクトの納期を削ります。**

あわせて、上司や先輩の側には、新人が相談しやすい時間と雰囲気を用意し、相談されたことを歓迎する責任があります。

NG→なぜ→OK—相談メッセージの作り方

【NG】「〇〇がうまくいきません。どうすればいいですか？」

【なぜNGか】相手は、状況・試したこと・何を決めてほしいかが分からず、まず逆質問から始めるしかありません。

あなたの“考えた形跡”も見えないため、丸投げに映ります。

【OK】「〇〇の実装で詰まっています。

【状況】△△をすると□□のエラー。

【試したこと】公式ドキュメントとログ確認、◇◇は切り分け済み。

【自分の考え】A案＝簡単だが拡張性低／B案＝堅いが工数大。

【お願い】15分で、A・Bどちらで進めるか相談させてください。」

違いは“情報量”ではなく、相手が即座に判断・行動できるかどうかです。

■ 相談のタイミグー「10～15分考えてわからなければ聞く」ルール

エンジニアの世界では「自分で10～15分調べ・考えてわからなければ相談する」という暗黙のルールが広く共有されています。このルールには3つの意図があります。

- ・ ①自分で考える力を育てるため、何でもすぐ聞く癖がつくと、自分で考える力が育たないまま年次だけが上がり、いつまでも判断を任せてもらえません
- ・ ②チームの時間を尊重するため、すぐ解決できる問題を自分で解決できるようになることが、チームへの貢献です
- ・ ③一方で、わからないまま何時間も詰まり続けられないため、「詰まったら声をかける」は世界共通の開発文化です

ブロッカーの共有は最大3時間以内を目安にします。自分だけで抱え込まず、チームの力を活かすことが、プロとして正しい判断です。

■ 相談の前に準備することー「丸投げ」ではなく「判断の仰ぎ方」

相談は「わかりません、教えてください」から始めるのではなく、自分なりの考えを持った上で「判断を仰ぐ」形が理想です。これが「四点セット」（4-2で学んだ内容）の本質です。

- ・ 相談前チェックリスト：
- ・ 状況を整理できているか（何が問題で、今どういう状態か）
- ・ 自分なりの解決策を考えたか（A案・B案の候補はあるか）
- ・ 相談相手への「お願い事項」は明確か（決めてほしいのか、アドバイスが欲しいのか、情報が欲しいのか）
- ・ 相手の時間をどれくらい使うか予告できるか（「3分でよいですか」「15分ほどよいですか」）

■ 上司が多忙なときの相談の仕方

上司が会議中・集中していそうなとき、どうすればよいかは新卒社員が悩みやすい場面です。チームの文化にもよりますが、基本的な判断軸を持っておきましょう。

緊急度・状況	対応方法
● 高（本番障害・期日当日）	チャット・メールで「緊急です」と明記して連絡します。遠慮は禁物です
● 中（今日中に判断が必要）	チャットで「今日中にご判断いただきたい件があります。〇時以降にお時間ありますか」と予告します
● 低（急ぎではない）	「お時間のあるときに」と添えてチャットやメールで送ります。「〇〇について相談があります（急ぎではありません）」
17 lonlを活用	定期的にlonlの時間があれば、緊急でない相談をそこでまとめて行うのが理想です

4-6 報連相の失敗パターンと対処法

報連相のスキルを身につける上で、失敗パターンを知っておくことは大切です。「なぜ報連相がうまくいかないのか」の原因を理解することで、自分のクセを直しやすくなります。

■ 失敗パターン①：完璧な情報が揃うまで報告しない

「まだ確認中だから」「もう少し詳細が分かってから」と待ちすぎて、問題が大きくなってから報告するパターンです。「速報」という概念を持つことが解決策になります。

- ・ 対処法：事実だけでもよいので早く伝える、「〇〇が発生しています。詳細は調査中です。30分以内に続報します」で十分です

■ 失敗パターン②：長い前置きから入る

「えーと、〇〇のことなんですが、先週から取り組んでいて、その中でですね…」という報告の仕方は、聞く側のストレスになります。結論を先に言う習慣がないとこうなります。

- ・ 対処法：最初の一文で結論を言う練習をします。話す前に「結論は何か」を自分の頭の中で整理してから話し始めます

■ 失敗パターン③：事実と意見・推測が混ざる

「〇〇があって、だからたぶんこうなっていて、とにかく大変な状態で…」という報告は、上司が何を信じてよいか分からなくなります。事実・推測・意見を分けて話す習慣がないとこうなります。

- ・ 対処法：「事実として確認したこと」と「自分の推測・意見」をはっきり分けて伝えます。
「ログを見ると〇〇でした（事実）。原因はおそらく〇〇だと思います（推測）。対応策として〇〇を提案します（意見）」

■ 失敗パターン④：相談を「丸投げ」にする

「どうすればよいですか」とだけ聞くのは相談ではなく丸投げです。「自分の考え」を持たずに相談すると、上司の答えを待つだけになり、自分の成長機会も失います。

- ・ 対処法：相談前に必ず「自分はこう考えています」を準備します。自分の考えがないときは、「A案・B案のどちらが良いと思いますか」という形にするだけでも大きく変わります

■ 失敗パターン⑤：「大丈夫です」で上司の心配を打ち切る

「大丈夫？」と聞かれたときの「大丈夫です」が、実は大丈夫ではないケースは多いです。**気を遣って「大丈夫です」と言ったために、問題が表面化するのは決まって手遅れになってから。**そのとき上司はもう助けられません。

- ・ 対処法：「大丈夫です」の代わりに「現在〇〇の状態です。今日中に完了できると思います」と具体的な状況を返す習慣をつけます

事例から学ぶ：ハウレンソウが機能しない組織の末路—ある食品会社の集団食中毒事件（2000年）から学べること

2000年に発生した同社の集団食中毒事件では、製造現場の衛生問題が適切に報告・連絡されていなかったことが被害拡大の一因となりました。

「問題を言い出しにくい組織文化」「悪い情報が上に伝わらない仕組み」が、約14,000人もの食中毒患者を生む結果につながりました。

この教訓から学べること：報連相が機能する組織文化は、製品・サービスの品質だけでなく、社会への責任と直結しています。

新卒社員のうちから「悪い情報ほど早く」を実践する姿勢が、組織全体の安全を守ります。

ポイント

報連相は「形式」ではなく「チームを機能させる技術」です。上司に怒られないための報告ではなく、組織として最善の判断・行動をするための情報共有として捉えましょう。

報連相の質が上がると、信頼が積み上がり、より重要な仕事を任せてもらえるようになります。

▶ やってみようーワーク・演習

やってみよう①：報告の型を体に覚えさせる練習

以下のシナリオをPREP法で30秒で報告してみましょう。

シナリオA：「ログイン機能の実装が完了しました。テスト10件すべて合格です。明日からUI改善に取り組みます」

シナリオB：「APIの応答速度テストを完了しました。

平均500ms、最大2000msでした。

SLAの1000ms以内を超えるケースがあり、原因はデータベースのN+1問題と特定しました。

今日中に修正方針をご相談したいです」

やってみよう②：「速報→続報→完結報告」の練習

シナリオ：午前10時に本番環境でエラーが多発していることを発見した。原因はまだ不明。

①速報（10時00分）：何を・どう報告するか

②続報（10時30分）：原因がデータベース接続エラーと特定された。

影響範囲は〇〇ユーザー。

③完結報告（12時）：復旧完了。根本原因と再発防止策を報告する。

それぞれのメッセージをSlack・口頭・メールのどれで行うかも含めて考えましょう。

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・ 「結論を先に」で、30秒の報告を一つ口に出してみる。

4-7 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ 報告・連絡・相談の違いと、「完了」はいつか。
- ・ 「結論を先に」とは、どういうことか。
- ・ なぜ悪い情報ほど早く伝えるのか。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、これからの学びや仕事の中で、少しずつ埋めていきましょう。

第5章 仕事と作業の違い—仕事の進め方（8～9日目・1.5日）

🎯 この章のゴール

この章では、「仕事」と「作業」の違い、そして仕事の進め方を学びます。

終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ①【違いがわかる】「仕事」と「作業」の違いがわかり、一言で言える。
- ②【知っている】指示を受けるときの3点確認（目的・背景・期待）を知っている。
- ③【考え方がわかる】優先順位を「重要×緊急」で考える、とわかる。

💡 まず自己診断—読む前に、いまの自分を知ろう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「仕事と作業の違いを自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
 - ② 「言われたことを全部やれば「仕事をした」ことになると思いますか？」を、いまの考えで一言だけ書きます（正解は気にしません）。
- 読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

✦ よくある勘違いと、その反証

【勘違い】「指示どおりにやる＝仕事」→

【反証】それは“作業”です。

仕事は「目的（何のため）」と「価値（誰がどう助かるか）」を満たして初めて完了します。同じ指示でも、目的を理解して動く人は、想定外のときに正しく判断できます（「業務」の範囲は5-8で深めます）。

5-1 「仕事」と「作業」はなにが違うのか

「仕事と作業の違い」を理解することで、自分の働き方が根本的に変わります。**同じ行動でも「考えているか」「考えていないか」で、生み出せる価値が大きく変わります。**

観点	作業 (task)	仕事 (work)
定義	やり方が決まっています。指示通りにこなせば完結する	目的・背景・影響を考えながら動きます。判断・工夫が伴う
思考	考えなくても誰でも同じ結果になる	「なぜやるのか」を理解した上で動く
完了	やり終えた時点	目的が達成された時点
評価	作業量・正確さ	生み出した価値・改善の提案

作業が悪いものではありません。「作業を仕事に変える視点を持つ」ことで、同じ業務からより多く

の価値を生み出せるようになります。

作業を仕事に変える3つの問い：

- ・ なぜこれをやるのか、理解していますか？
- ・ 完了したら、誰が・何のために使うのですか？
- ・ 自分なりの工夫・判断が入る余地はありますか？

■ 「業務」の正しい定義

- ・ 業務とは「企業活動としてやるべきことすべて」を指します
- ・ 日々の勤怠管理・バックオフィス業務・この研修への参加、すべて業務です
- ・ 「業務優先」という言葉は正しく理解して使うことが大切です

5-2 上長から「仕事」をもらう

📖 事例から学ぶ：「集計しといて」の一日（ありがちな失敗シーン）

「このデータ、集計しといて」とだけ言われた新人が、丸一日かけてグラフ付きの精緻な資料を作りました。

提出すると上司は困った顔。

欲しかったのは「明日の朝会で口頭共有する概算」で、グラフは不要でした。

一日が消えた原因は能力ではなく、「何のために・誰がいつ使うか」を最初の30秒で確認しなかったことです。

3点確認（目的・背景・期待）は手間ではなく、最大の時短です。

もっとも、あいまいさの責任は、受け取る側だけにあるわけではありません。指示を出す側にも、目的・背景・期待を明確に伝える責任があります。

仕事を「仕事として受け取る」習慣を持つことで、自分の成長速度が上がります。

■ 指示を受けるときの3点確認

- ・ 目的、なぜこれをやるのか。何のためになるのか
- ・ 背景、この仕事はどこにつながっているのか。誰が最終的に使うのか
- ・ 期待、どんなアウトプットを・いつまでに・どのレベルで求められているのか

この3点を確認してから動くことで、方向性のズレが減り、より価値のある成果物が生まれます。

■ 仕事を渡すときの3点セット（将来の自分のために）

- ・ 目的：なぜこれをやるのか
- ・ 背景：この仕事はどこにつながっているのか
- ・ 期待：どんなアウトプットをいつまでに求めているか

5-3 仕事の進め方—段取りと品質の基本

■ 仕事の三要素—QCDのバランス

品質（Quality）・コスト（Cost）・納期（Delivery）はトレードオフの関係にあります。「どれを最優先にするか」を顧客・上司と合意した上で進めることで、期待に応える成果物が生まれます。

- ・ 合意した上で進めることで、後から「こんなはずじゃなかった」が防げます
- ・ プロとしては「どれを最優先にするか」を明示して動きます

■ 仕事を受けるときに確認すべき5つのこと

確認項目	内容
目的と背景 (Why)	なぜこの仕事が必要か
成果物の定義 (What)	何を・どのレベルで作るか
期限 (When)	いつまでに・どの段階で報告が必要か
リソース (How)	どんな方法・ツール・人を使えるか
制約条件 (Constraint)	予算・使えない方法・特別な注意点

■ 優先順位のつけ方—緊急度と重要度のマトリクス

	重要度：高	重要度：低
緊急度：高	【第1優先】 今すぐやる	【第3優先】 委任・圧縮
緊急度：低	【第2優先】 計画して取り組む、自己成長・準備・関係構築	【第4優先】 やらない・後回し

意識的に「低緊急・高重要」（自己成長・準備・関係構築）に時間を作ることが、長期的な成長につながります。

■ 締め切りの管理—早めの確認が大切

- ・ 「完了日の3日前」には進捗を確認し、問題があれば早めに報告します
- ・ 「80%できた段階」で中間確認、早期に方向性を確認することで、修正コストが下がります
- ・ エンジニアとしての実践：スプリントのデイリースタンドアップで毎日進捗を共有し、ブロックを早期に解消します

5-4 プロジェクト管理の基本—チームで仕事を進める技術

エンジニアとして働く場合、多くの仕事はチームでのプロジェクト形式で進みます。**プロジェクト管理の基本を知ること、個人の仕事の進め方が根本的に変わります。**

■ プロジェクトとは何か—定義を正確に理解する

- ・ プロジェクトの定義：「明確な目標を持ち、期間・予算・リソースに制約のある一連の活動」、日常の繰り返し業務（オペレーション）とは違います
- ・ プロジェクトの三大制約：スコープ（何をするか）・タイム（いつまでに）・コスト（いくらで）、この3つはトレードオフの関係にあります。1つを変えれば他の2つに影響します
- ・ プロジェクト成功の定義：スコープ・タイム・コストを守るだけでなく「顧客・ステークホルダーが満足しているか」が本当の成功です

■ ウォーターフォールとアジャイル—2つの開発手法

ソフトウェア開発における代表的なプロジェクト管理手法を理解することで、自分のチームがどう動いているかが見えてきます。

- ・ ウォーターフォール：要件定義→設計→開発→テスト→リリースと順番に進む手法。各フェーズが完了してから次に進みます。仕様が最初から明確な場合に向いています
- ・ アジャイル（スクラム等）：短いサイクル（スプリント：1～4週間）で計画→開発→振り返りを繰り返す手法。変化への対応力が高く、現代のWebサービス・SaaS開発で広く採用されています
- ・ どちらが正解かではない：プロジェクトの性質・チームの規模・顧客の関わり方によって適切な手法が変わります。大切なのは「なぜその方法で進めているか」を理解することです

■ スクラムの基本—アジャイルの代表的な実践方法

用語	説明
スプリント	1～4週間の開発サイクル。各スプリントの終わりに動くソフトウェアを届けます
デイリースタンドアップ（朝会）	毎日15分、「昨日やったこと・今日やること・ブロッカー（詰まっていること）」を共有します
スプリントレビュー	スプリント終了時に成果物をステークホルダーに見せてフィードバックを得ます
スプリントレトロスペクティブ（振り返り）	「よかったこと・改善すべきこと・次のアクション」をチームで議論します。このサイクルがチームの継続的な成長を支えます
バックログ	実装すべき機能・タスクをリスト化したもの。優先度順に並べて管理します

5-5 時間管理の実践—「忙しい」から「効果的」へ

「仕事が多すぎて時間が足りない」という状態は、時間管理の技術で大幅に改善できます。ただし、時間管理は「効率化」だけが目的ではありません。**重要なことに時間を使えるようにする「選択と集中」が本質です。**

■ タイムブロッキング—時間を「予約」して守る

- ・ タイムブロッキングとは：1日の時間帯ごとにやることを事前に決め、カレンダーに「予約」として入れる手法
- ・ 深い集中が必要な仕事（コーディング・設計・文書作成）：午前中の集中力が高い時間帯を確保します。会議を午後にまとめることで「集中ブロック」を守れます
- ・ メール・Slack対応：1日に2～3回（例：9時・12時・17時）の「対応タイム」を設けます。常時対応しないことで、思考の中断を防ぎます
- ・ バッファ時間：1日の20～30%は「計画外の緊急対応」のために空けておきます。詰め込みすぎると計画が崩れます

■ マルチタスクの罠—「同時進行」が生産性を下げる

脳科学の研究によると、人間は真のマルチタスクができません。複数の作業を切り替えながら進めることは、実際には「タスクスイッチング」であり、集中の切り替えコスト（準備時間）が発生しま

す。

- ・ タスクスイッチングのコスト：1つのタスクから別のタスクに切り替えた後、元の集中状態に戻るまで平均23分かかる（カリフォルニア大学の研究。中断後に作業へ復帰するまでの目安として広く引用される数字で、定義や条件により幅があります）
- ・ シングルタスクの実践：1つのタスクに集中する時間を意図的に作ります。Slackやメールの通知をオフにする・作業中に別の業務の文書を開かない
- ・ ポモドーロ・テクニック：25分集中→5分休憩のサイクルを繰り返す手法。集中力の維持と定期的な休憩の両立が目的です

■ 「締め切り効果」を使いこなすーパーキンソンの法則

- ・ パーキンソンの法則：「仕事は与えられた時間をすべて使い切るまで膨張する」、締め切りが遠いほど、仕事はだらだら続きます
- ・ 自分で「仮の締め切り」を設定する：期日より早い「自分の締め切り」を意識することで、効率が上がります
- ・ 「完璧主義」との戦い：「もう少し良くできる」という気持ちで締め切りを破ることより、「80%の品質で期日に届けてフィードバックをもらう」方が価値が高いです

5-6 問題解決の実践プロセスー「詰まったとき」に動ける力を持つ

仕事では「詰まる」場面が必ず生じます。その時に「パニックにならず・効果的に解決できる」能力が、プロとしての実力を示します。

■ 問題解決の基本ステップ

ステップ	内容
① 問題の定義	「何が問題か」を明確にすることが最初のステップです。「なんかうまくいかない」ではなく「〇〇の条件でのみ〇〇が起きる」と具体化します
② 情報収集	エラーメッセージ・ログ・直前の変更点・発生条件を整理します。「何が起きているか」の事実を集めます
③ 仮説を立てる	「おそらく〇〇が原因ではないか」という仮説を複数立てます。最初の仮説に固執せず、複数の可能性を考えます
④ 仮説を検証	一度に一つの変数を変えてテストします。複数の変更を同時に行うと「どれが効いたか」がわかりません
⑤ 実装・文書化	解決したら「何が原因で・どう解決したか」を記録します。同じ問題で次も詰まらないために、また仲間を助けるために

■ 「助けを求めるタイミング」の判断

- ・ 10～15分ルール：自分で調べて・考えて10～15分経ってもわからなければ相談します。すぐ聞く習慣は成長を妨げますが、長時間一人で詰まることもチームの損失です（基本の考え方は第4章4-5を参照）
- ・ 相談のコスト意識：上司・先輩に相談することは「相手の時間を使う」行動です。「自分で調べたこと・試したこと・今考えている仮説」を整理してから相談することで、相手の時間を尊重します

- ・ 詰まったとき最初に確認すること：①公式ドキュメント②エラーメッセージの全文をそのまま検索③GitHubのIssues・Stack Overflow、この順番で調べることで解決率が大幅に上がります

📖 事例から学ぶ：タスク管理の失敗と成功—同じチームの2人のエンジニアから学べること

同じチームに入った2人の新卒エンジニア。

Aさんは毎朝「今日のタスクリスト」を作り、優先度順に並べ、完了したらチェックしていきました。

詰まったら15分で相談し、進捗を午前・午後に一度ずつ報告しました。

Bさんは「なんとなく頭の中でタスクを管理」し、完了も共有せず、詰まっても「もう少し自分でやってみよう」と1日詰まり続けました。

3ヶ月後、Aさんは「任せやすいエンジニア」と評価され、新しいプロジェクトを任されました。この教訓から学べること：タスク管理と報連相は「仕事の量」ではなく「仕事の見える化」です。

見える化されていない仕事は、チームには「存在しない」と同じです。

💡 ポイント

仕事の「技術」は経験で磨かれますが、仕事の「姿勢」は今日から変えられます、締め切りを守る・詰まったら相談する・進捗を共有する・完了を報告する。
これらの習慣は特別なスキルではなく、意識して続けることで誰でも身につけられます。

▶ やってみよう—ワーク・演習

やってみよう追加①：タスク管理ツールを使い始める

以下の手順で今日からタスク管理を始めましょう。

- ①今日やるべきことを全部書き出す（ノート・付箋・NotionでもOK）
- ②「緊急度×重要度マトリクス」に当てはめて優先順位をつける
- ③最優先のタスクから取り組む。1時間に1回「今何をしているか」を確認する
- ④1日の終わりに「できたこと・できなかったこと・なぜできなかったか」を3行書く

やってみよう追加②：問題解決ロールプレイ

以下のシナリオで問題解決の5ステップを実践してみましょう。

シナリオ：本番環境のAPIが特定のユーザーIDで500エラーを返している。

- ①問題の正確な定義（どのユーザーIDで・どのAPIで・いつから）
- ②情報収集（どんなログを・どこで確認するか）
- ③仮説を3つ立てる（データの問題・コードの問題・インフラの問題）
- ④検証方法（各仮説をどう確認するか）
- ⑤解決後の記録（何が原因で・どう解決したか・再発防止策は何か）

◆ 考えてみよう—倫理の視点

品質への誠実さ、「とりあえず出せばいい」という妥協は、将来そのコードを触る全員への影響になります。品質への誠実さが、長期的な信頼とチームの生産性を守ります。

知っておこう—法律の紐づけ

民法（契約）、納期・品質の約束は法的な契約です。PL法、製品の品質への責任は出荷後も続きます。これらを知ること、より責任感を持って仕事に臨めます。

5-7 プロの視座の核心—後工程はお客様・自工程完結・カイゼン

品質・安全・納期に高い意識を持つプロフェッショナルな組織で最初に叩き込まれる3つの概念があります。ITエンジニアとして働く皆さんにも、全く同じ考え方が当てはまります。**この視座を持つことが、「技術者」と「プロのエンジニア」を分かつ差のひとつです。**

■ ①後工程はお客様—自分の出力が誰の入力になるかを意識する

品質管理の世界では「後工程（自分の次の工程を担う人）はお客様」という言葉があります。前工程が品質の悪いものを渡すと、後工程は品質の悪いものを受け取ることになります。これはITエンジニアの仕事にそのまま当てはまります。

場面	「後工程はお客様」の実践
コードを書く	自分のコードを次にレビューする人、次に修正する人が「お客様」。読みやすく・意図が明確なコードを書きます
ドキュメントを書く	自分のドキュメントを読んで作業する後任者が「お客様」。「自分がわかる」ではなく「相手がわかる」を基準にします
テストを行う	自分のテスト結果を受け取って判断する人が「お客様」。曖昧な結果・未確認のまま渡しません
要件を定義する	要件を受け取って実装するエンジニアが「お客様」。解釈のブレが生じないよう明確に定義します
設計書を書く	設計書を読んで実装するエンジニアが「お客様」。設計の「なぜ」まで書きます

- ・ 「後工程に迷惑をかけない」を最低ラインに、「後工程を喜ばせる」を目標にします
- ・ **「自分の仕事が終わった」ではなく「後工程が受け取れる状態になった」が本当の完了です**

■ ②自工程完結—「完全なものを次に渡す」責任を持つ

「自工程完結」とは、「自分の担当する工程で、品質を完結させてから次に渡す」という考え方です。「あとでテストで見つかるからいい」「レビューで指摘してもらえばいい」という後ろ向きな姿勢の反対です。

- ・ 自分の仕事の「完了基準」を自分で設定し、その基準を満たしてから渡します
- ・ 「たぶん大丈夫」で渡すのではなく、「確認した・動作確認した・テストした」という根拠を持って渡します

- ・ 不明な点・不安な点があるときは、曖昧なまま渡すのではなく、確認してから渡すか、「ここは未確認です」と明示して渡します

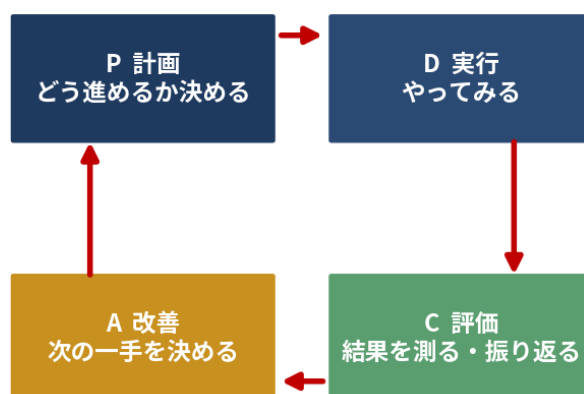
自工程完結できていない状態	自工程完結できている状態
「たぶん動く」と思いながらPRを出す	ローカルで動作確認し、テストが通ることを確認してからPRを出します
未解決のTODOコメントを残したままマージする	TODOはIssueに起票して、コード上には残しません
「あとでレビューで聞けばいい」と設計の疑問を放置する	疑問は実装前に解消します。不明なまま実装しません
バグを「誰かが気づくだろう」と思いながら放置する	気づいたバグは自分でIssueを起票し、または自分で修正します

■ ③カイゼン—改善提案ができるビジネスパーソンになる

「カイゼン（改善）」とは、日本の製造業で磨かれてきた品質管理の核心にある概念であり、世界中のプロフェッショナルな組織に広まっています。現状に満足せず、常により良い方法を考え、提案し、実行し続けることです。ITエンジニアとして、この姿勢は「技術的負債の解消」「プロセスの改善」「チームの生産性向上」として現れます。

カイゼンのサイクル（PDCA）：

PDCA — 小さく回して、改善を積み重ねる



1回で完璧を目指さず、回す速さと回数で差がつく

図 PDCAサイクル

フェーズ	内容
Plan (計画)	現状の問題を発見し、改善策を立案します

フェーズ	内容
Do (実行)	小さく・安全に試します
Check (確認)	改善前後を比較・数値で評価します
Action (改善)	さらに良くする点を次のサイクルに活かします

エンジニアとして今日からできるカイゼンの例：

- ・ 「毎回同じ手順をやっていて面倒だ」 → スクリプト化・自動化して提案します
- ・ 「このドキュメントは古くて使えない」 → 自分で更新して、チームに共有します
- ・ 「このフローは無駄が多い」 → 改善案をIssueやSlackで提案します
- ・ 「このコードは読みにくい」 → リファクタリングを提案してPRを出します

大切なのは「誰かがやるだろう」ではなく、「自分がやる」という当事者意識です。カイゼンは「担当者だけが行うもの」ではありません。気づいた人が、できる範囲で動く文化が、組織全体を強くします。

💡 ポイント

「後工程はお客様・自工程完結・カイゼン」、この3つの意識が、技術力を持った人を「プロのエンジニア」にします。
作業をこなす人ではなく、価値を届け続ける人になるために、今日からこの視座を持って働きましょう。

▶ やってみようーワーク・演習

やってみよう：「あいまいな指示を仕事として受け直す練習」

- ① 「顧客向けのレポートをまとめておいて」
- ② 「このバグを直しておいて」
- ③ 「新機能の仕様を考えておいて」 各指示について「目的・背景・期待」を確認するための質問を3つ書き出し、ロールプレイで実践しましょう。

さらに：自分が今の仕事の中で「後工程に渡している何か」を1つ特定し、「自工程完結できているか」を自己評価してみましょう。

5-8 「業務」とは何かー改善も育成も、立派な業務である

ここで、社会人としての土台になる、とても大事な考え方を押さえておきます。それは「業務とは何か」という問いです。**これを誤解したまま現場に出ると、本人も組織も知らないうちに衰退します。**だからこそ、研修のこの段階で確実に身につけてください。**会社は成長し続けなければ、競争相手に追い抜かれます。停滞は後退です。**改善・育成・学習を「業務」として回し続けることが、その後退を防ぎます。

「今は忙しいから」「まず目の前の業務を終わらせてから」。こうした「業務優先」という言葉に

は、落とし穴があります。多くの場合、「業務」という言葉が無意識に「案件対応＝売上に直結する仕事」だけを指すものとして使われているのです。**この認識は、間違っています。**

業務とは、組織の中で行われるすべての活動です。 顧客対応や開発はもちろん、社内手続き、教育・育成、採用、改善活動、自己研鑽、仕組みや文化づくり、これらすべてが業務です。

この認識が共有されていないとき、「業務優先」という言葉は恐ろしい凶器になります。教育が「余計なこと」になり、改善活動が「本業ではない」扱いを受け、自己研鑽が「業務時間外にやるもの」になり、採用が「手が空いたときに」になる。こうして、組織の成長に必要なものが次々と削られます。

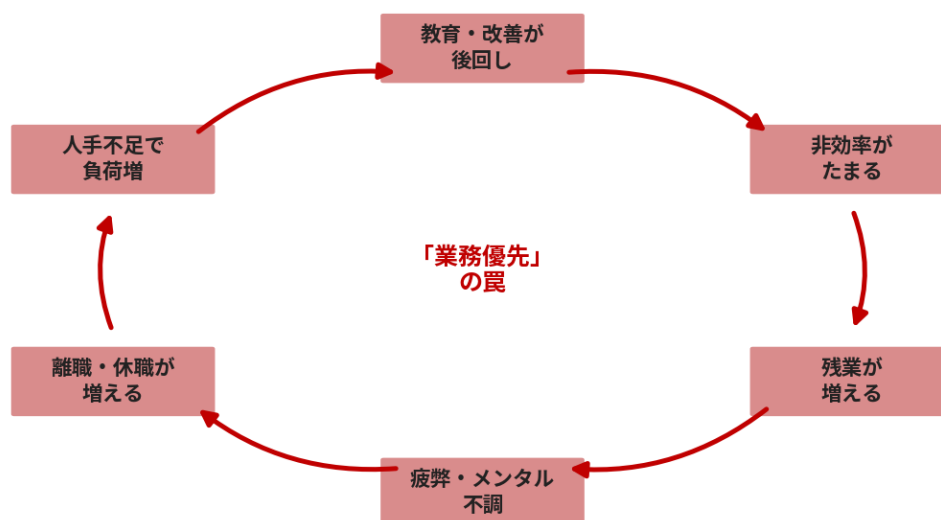
■ 「案件対応しかしない」と、なぜ組織は衰退するのか

最も皮肉なのは、それらを削れば削るほど、肝心の案件対応すら苦しくなることです。

- ・ 人が育たないから、品質が上がらない。
- ・ 非効率が蓄積するから、生産性が落ちる。
- ・ 疲弊が重なるから、ミスが増える。

「業務優先」という言葉が、本丸である案件対応そのものを蝕んでいきます。そして次のような悪循環が始まります。教育・改善が後回しになる → 非効率がたまり残業が増える → 人が疲弊し、メンタル不調や燃え尽きが起きる → 離職・休職が増える → 人手不足で残った人の負荷がさらに増える → 「改善する余裕がない」がますます強化される。改善提案は却下され続けて廃れ、声を上げていた人もやがて黙ります。その沈黙は「組織が安定している」のではなく、「誰も期待していない」状態です。そして最も深刻なことに、**改善思考を持つ優秀な人から辞めていきます。**

「業務優先」で改善を後回しにすると、組織は悪循環に陥る



※ 断ち切る鍵は「改善・育成・学習も“業務”として回す」こと

図 「業務優先」の悪循環ループ

△ 注意、「顧客優先」も同じ構造を持ちます

顧客を大切にすることは経営の大前提であり、わざわざ口にする必要のないことです（「水を飲むのは大事だから今は水優先で」とは言いません）。

それをあえて口にする人の言葉の直後に、何が来るかを観察してください。

「顧客優先なので、この改善は後回しに」「顧客優先なので、研修は別日で」。

顧客という言葉が盾になり、誰も反論できなくなる。

改善・育成を捨て続けた結果、サービスの質が落ち、疲弊した人が顧客対応をし、優秀な人が去る。

「顧客優先」が、最終的に顧客を裏切るのです。

■ 「業務改善」の組織は、複利で強くなる

衰退する組織には「業務優先」という言葉が、成長し続ける組織には「業務改善」という言葉が広がります。「業務改善」の組織では、社内のあらゆることが業務だと共有されているので、そもそも「優先」という概念が成立しません。次の2つのたとえが、その違いをよく表しています。

📖 事例から学ぶ：「きこりのジレンマ」と「茹で蛙」から学べること

【きこりのジレンマ】必死に木を切り続けるきこりに、通りかかった人が言います。

「斧が切れていませんよ。

一度休んで刃を研いだら？」

。

きこりは答えます。

「木を切るのに忙しくて、そんな時間はない」。

忙しいから改善できないのではありません。

改善しないから、永遠に忙しいのです。

【茹で蛙】熱湯に入れば飛び出す蛙も、ぬるま湯から徐々に温度を上げると、変化に気づかないまま茹で上がります。

「うちはこんなもの」「業界的に仕方ない」と外を見ない組織は、すでにぬるま湯の中。

基準が内側にしかない、問題が問題として認識されません。

その間も、外の世界（競合・採用市場）は着実に動いています。

改善の効果は、最初は小さくても複利で効いてきます。毎日少しでも刃を研ぐ人と、研がない人。最初の数か月は研がない人のほうが多く進みますが、やがて逆転し、1年後には追いつけない差がつき、その差は翌年さらに広がります。**「改善する時間がない」は、数字が証明する錯覚なのです。**

■ 改善の「地図」—3Mと5S

改善には、思想的な地図があります。代表的なものが3Mと5Sです。

- ・ 3M（ムリ・ムダ・ムラ）：排除すべき非効率の三類型です。問題に名前をつけられれば、議論も対処もできます。「ムリ」を「頑張れ」で解決しようとする、メンタル不調と燃え尽きを生みます。
- ・ 5S（整理・整頓・清掃・清潔・躰）：製造業の現場で磨かれてきた、改善（カイゼン）の手法です。ここを誤解してはいけません。**5Sは「環境美化（職場をきれいにすること）」の手法ではなく、ムダをなくして効率と品質を高めるための手法です。**だから、見た目がどれだけきれいでも、効率と品質が上がっていなければ、その5Sは「できていない」のです。整理・整頓・清掃は手段にすぎず、目的は成果（効率と品質の向上）にあります。中でも最後の「躰」＝改善を習慣・文化として定着させることが最も重要で最も難しく、その成果を一度きりで終わらせないための鍵になります。

- ・ この2つはセットで効きます。3Mが「なくすべき非効率（何を）」を名指す地図なら、5Sはそれを現場で実際に減らす型（どうやって）です。たとえば整理・整頓は「探す・運ぶ」のムダと、人によってやり方が違う「ムラ」を減らし、清掃は異常やミスに早く気づかせます。3Mで問題に名前をつけ、5Sで毎日それを潰す。この往復が、改善を日常に変えていきます。

そして、業務改善を積み重ねた組織にだけ、「そもそもこの仕事は必要か」と仕事の構造を問い直す**業務改革**が生まれ、その先ではじめて**DX**が意味を持ちます。業務優先の文化のままツールだけ導入しても、使われないか形骸化する、それがDX失敗の本当の理由です。技術や予算ではなく、文化の問題なのです。

■ 新卒のあなたにできること—この研修も「業務」です

新卒のあなたにできるのは、大きな提案でなくて構いません。「このメール、テンプレにしたら楽になりませんか」「この作業、来月の自分が楽になるよう整えておけないか」。その一言が改善文化の起点になります。違和感を覚えたら、「この仕事、もっと良くできないか？」と問いかけてみてください。

そして、ぜひ覚えておいてほしいことがあります。**いま受けているこの研修も、日々の学びも、振り返りも、立派な「業務」です。**「業務優先だから研修は後回し」「忙しいから勉強は自分の時間で」という組織が衰退していくのは、まさにここで見てきた理由からです。学ぶこと・改善すること・人を育てることを「本業の一部」として捉える感覚を、今のうちに身につけてください。それが、あなた自身と、あなたがこれから所属する組織の未来を守ります。

▶ やってみよう—ワーク・演習

自分の担当作業（または研修での作業）から、「ムダだな」「ムリがあるな」と感じるものを1つ見つけ、どう改善できるかを1案書き出してみましょう。改善の第一歩は、非効率に「名前をつける」ことです。

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・ 何か（研修の課題でも）頼まれたら「目的・期限・どこまでやるか」を3点メモする練習をする。

5-9 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ 「仕事」と「作業」の違いを、一言で。
- ・ 指示を受けるときの3点確認とは何か。
- ・ 優先順位は何で決めるか。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、これからの学びや仕事の中で、少しずつ埋めていきましょう。

第6章 人と働く—コミュニケーションと信頼の築き方（9日目後半～11日目・2日）

🎯 この章のゴール

この章では、人と働く、伝える・聴く・確認する、そして信頼の築き方を学びます。
終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ① **【コツがわかる】** 伝える・聴く・確認するで認識ズレを防ぐ、とわかる。
- ② **【知っている】** 自分も相手も尊重する言い方（アサーション）を知っている。
- ③ **【大切さがわかる】** 安心して言い合える場（心理的安全性）が大切な理由がわかる。

💡 まず自己診断—読む前に、いまの自分を知ろう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「伝え方・聴き方を自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
- ② 「「伝えた」と「伝わった」は、どこで見分けがつくと思いますか？」を、いまの考えで一言だけ書きます（正解は気にしません）。

読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

6-1 伝える・聴く・確認する

■ 信頼される3つのコミュニケーション力

能力	意味	信頼への影響
伝える力	自分の考えを相手に正確に届ける	「この人に任せると伝わる」という安心感
聴く力	相手の意図・感情・状況を正確に理解する	「この人は自分を理解してくれる」という信頼感
確認する力	理解のズレを防ぐために確認する	「この人とは認識のズレが起きない」という信頼感

✦ よくある勘違いと、その反証

【勘違い1】「言った＝伝わった」→

【反証】 **伝わったかどうかは、相手が再現・行動できて初めて確認できます。**

成否を決めるのは“発信”ではなく“受信”側です。

【勘違い2】「聴く＝黙って聞くこと」→

【反証】 聴く（傾聴）は受け身ではなく、相手の意図を“こちらから取りにいく”能動的な作業。

相槌・要約・確認がそろって初めて「聴いた」になります。

■ 伝えること—相手が理解できる言葉を選ぶ

- ・相手の知識レベルに合わせます。エンジニアが非エンジニアに話すとき、専門用語を使いません
- ・一文一意、一つの文章に一つのことだけ伝えます
- ・具体例を使います。抽象的な話は必ず具体例とセットで伝えます

■ 聴くこと—傾聴の実践

	「聞いたふり」	傾聴（アクティブリスニング）
目的	次に自分が話すことを考える	相手の話を理解すること
態度	適当に頷く・視線がそれる	体をむけて目を見て、相槌を打つ
確認	しない	「〇〇ということですね」と確認する
結果	話した相手が「わかってもらえた感」を持ってない	「理解してもらえた」という信頼が生まれる

■ 確認すること—認識のズレを防ぐ

- ・自分が伝えた後：「今の説明で、わからないところがありますか」と確認します
- ・自分が聞いた後：「〇〇と理解しましたが、合っていますか」と復唱確認します
- ・エンジニアの実践：タスクを受けるとき「〇〇を、〇日までに、〇〇の形式で作る、という理解で合っていますか」と確認してから動きます

6-2 コミュニケーションの実践技術

■ アサーション—自分も相手も尊重した自己表現

スタイル	特徴	長期的な影響
攻撃的（Aggressive）	自分の意見を押し付ける	関係が悪化しやすいです
非主張的（Non-assertive）	自分の意見を言わない	本当の問題が解決されにくいです
アサーティブ（Assertive）	自分の意見を正直に・相手を尊重しながら伝える	長期的な信頼関係を構築できます

- ・「NO」と言える力、丁寧に・理由とともに伝えることは信頼を壊しません。むしろ誠実さを示します
- ・アサーティブな断り方：「△△については対応が難しい状況です。理由は〇〇です。代わりに〇〇であれば対応できます」

▶ 分岐ケース—あなたならどう断る？

状況：手一杯のとき、先輩に「これ今日中にお願い」と追加で頼まれた。

A：「（無理だと思いつつ）……わかりました」と引き受ける

B：「いまAの締切が今日です。これも今日だと両方が中途半端になります。Aを優先し、こちらは明日午前でよいですか？ 難しければ優先順位を教えてください」

Aは一見“いい人”ですが、両方落とすと信頼を失います。Bは相手を尊重しつつ事実と代案を出すアサーティブな断り方。断る＝拒否ではなく、“約束を守る形に調整する”ことです。迷ったらB。

■ 非言語コミュニケーション

- ・ 表情・姿勢・声のトーンが相手に与える印象は大きいです
- ・ 「あなたの話を聴いています」という体勢が、相手の話しやすさを生みます

■ フィードバックの技術—「観察→影響→期待」の構造

- ・ 観察：「〇〇したとき/〇〇の状態のとき」（事実・観察を伝えます）
- ・ 影響：「〇〇という影響がある/〇〇と感じました」（影響・感情を伝えます）
- ・ 期待：「〇〇してほしい/〇〇すると良くなると思います」（期待・提案を伝えます）

例：「先ほどの報告（観察）で、数字の根拠が示されていなかったため、私には判断が難しかったです（影響）。次回は数字の根拠もセットで教えてもらえると助かります（期待）」

6-3 チームワークと協働

■ チームとグループの違い

観点	グループ	チーム
関係	ただ集まっている	共通の目標に向かって動いている
責任	個人の責任	共同の責任
コミュニケーション	必要なときだけ	積極的に情報を共有する

■ 心理的安全性の実践

- ・ 心理的安全性が高いチームは：エラーを早期に報告できます・新しいアイデアを提案できます・困ったことを相談できます
- ・ 心理的安全性はリーダーだけが作るものではなく、チームの全員が作るものです

特に、立場が上の人や組織には、まず安心して発言できる場を保証する、一次的な責任があります。

- ・ 一人ひとりが「言いにくいことを言える」行動をすることで、チーム全体の安全性が育まれます

■ 多様性を活かす

- ・ **異なる意見は「脅威」ではなく「情報」です**、自分と異なる視点は、自分が見えていないものを見せてくれます
- ・ 無意識の偏見（アンコンシャスバイアス）、定期的に自分の思い込みを見直す習慣が、公平な判断力を育てます

- ・ エンジニアチームにおける協働：コードレビューは「批判」ではなく「協働の実践」です。レビューの目的は品質を上げることであり、互いの成長につながります

6-4 社内外の関係構築

■ 社内の関係構築

- ・ 「誰に相談するか」マップを作ります。技術的な相談はテックリード・業務的な相談は直属の上司へ
- ・ 「自分で10～15分考えてわからなければ相談する」というルールを持ちます（第4章4-5の「10～15分ルール」を参照）
- ・ 「今少しお時間いただけますか」と一言確認してから話しかけると丁寧です

■ お客様・取引先との関係で求められること

- ・ 約束を守ります。「できることとできないこと」を正直に伝え、言ったことは必ず守ります
- ・ **問題が起きたとき、言い訳より解決策を先に示すことが信頼を維持します**

◆ 考えてみよう—倫理の視点

言いにくいことを言う勇気、「空気を読む」ことは大切ですが、組織の中で間違った方向に進んでいると感じたとき、それを適切に伝えることが組織の健全性を保ちます。
正直なコミュニケーションが、長期的な信頼の基盤になります。

⚖️ 知っておこう—法律の紐づけ

民法（不法行為）、言動が損害賠償の対象になるケースがあります。誠実なコミュニケーションが自分と相手の両方を守ります。

▶ やってみよう—ワーク・演習

やってみよう：フィードバックを与え・受け取る練習をしましょう

- ①「観察→影響→期待」の型でフィードバックを行います
- ②フィードバックを受け取った後の自分の反応を観察します（素直に受け取れるか） ロールプレイ：困難なコミュニケーション場面を演じてみましょう シナリオ①：上司の指示に疑問を感じます。アサーティブに伝えるには シナリオ②：同僚のコードにバグを発見した。どう伝えるか

6-5 難しい会話を乗り越える—コンフリクトと向き合う力

職場では意見の違い・価値観のぶつかり・誤解が生まれます。それ自体は問題ではありません。**コンフリクト（衝突）を避けるのではなく、建設的に解決する力が、チームをより強くします。**

■ コンフリクトを「悪」と思わない—適切な対立の価値

- ・ 心理的安全性の高いチームは「健全な対立」がある：異なる意見が出ることで、盲点が発見され、決断の質が上がります
- ・ 避けることのリスク：コンフリクトを避け続けると「本音と言えないチーム」になり、問題が水面下で積み重なります
- ・ 目標は「勝つこと」ではなく「良い解決策を見つけること」：対話の目的は相手を打ち負かすことではなく、チームにとって最良の結論を導くことです

■ 意見の相違を建設的に扱う—5つのステップ

- ・ ①感情をいったん置く：反論したい気持ちをいったん横に置き、「まず相手の言いたいことを正確に理解する」に集中します
- ・ ②相手の視点を確認する：「つまり〇〇ということですか」と言い換えて確認します。相手が「そうです」と言えるまで続けます
- ・ ③自分の立場を事実と理由で伝える：「私が懸念しているのは〇〇です。理由は〇〇だからです」と構造的に伝えます
- ・ ④共通点を見つける：多くの場合、両者の「最終的に達成したいこと」は同じです。「〇〇という目標は同じですね」と確認することで対話が建設的になります
- ・ ⑤解決策を一緒に考える：「どうすれば両方の懸念が解決されるか」をチームの問いとして共有します

■ 上司・先輩と意見が違うとき

組織の中で「上の人の意見に賛成できない」場面は必ず来ます。このとき、「従うか黙るか」の二択しかないように感じる場合があります。しかし第三の選択があります。

- ・ 「反論する」ではなく「懸念を伝える」：「私はこう思います」より「〇〇の点が気になっています。理由を聞いてもいいですか」の方が対話が生まれやすいです
- ・ 「Disagree and Commit」の精神：議論した上で最終決定に従うことと、異議を唱えないことは別物です。意見を述べた上で組織の決定に従うことが、プロとしての誠実さです
- ・ 「私が間違っているかもしれない」前提：自分の判断が常に正しいわけではありません。上司の判断には「自分には見えていない情報」が含まれていることがあります

6-6 感情と仕事—EQ（感情知性）を育てる

「感情はプロとして仕事に持ち込むべきでない」という考え方があります。しかし人間は感情なしには働けません。**大切なのは「感情を持たないこと」ではなく、「感情と上手に付き合うこと」です。**

■ EQ（感情知性）の4つの要素

- ・ 自己認識：自分の感情を正確に認識する力、「今自分はなぜイライラしているか」がわかると、感情に流される行動を防げます
- ・ 自己管理：感情を適切にコントロールする力、怒りや不安があっても、相手に対して建設的に行動できます
- ・ 社会的認識：相手の感情・状況を読み取る力、チームメンバーの「様子がおかしい」に気づき、適切な言葉をかけられます
- ・ 関係性管理：感情を活かして関係を構築・維持する力、褒める・励ます・感謝を伝えることで信頼関係が育まれます

■ 職場でのストレスと上手に付き合う

社会人生活ではストレスを感じる事が避けられません。ストレスと「上手に付き合う力」が長期的な健康と活躍を支えます。

- ・ ストレスの早期認識：「なんとなく調子が悪い」を放置しない、睡眠の質・食欲・集中力の変化はストレスのサインです
- ・ 「言葉にして話す」の力：信頼できる同期・先輩・上司に「最近こんなことが難しくて」と話すことで、ストレスが軽減され、アドバイスが得られます
- ・ セルフケアの習慣：適切な睡眠・食事・運動・休暇が、高いパフォーマンスの基盤です。「休むことも仕事のうち」は本当のことです
- ・ 相談窓口を知っておく：会社の産業医・EAP（従業員支援プログラム）・外部の相談機関、困ったとき一人で抱えない選択肢を持ちます

■ 「感謝を伝える」カーチームの文化を作る小さな習慣

- ・ 具体的な感謝を言葉にする：「ありがとうございます」に「〇〇してくれたおかげで〇〇できました」と具体性を加えることで、相手に真剣に伝わります
- ・ 感謝はその場で伝える：後回しにするほど「あのときの感謝を言えていない」状況が積み重なります。気づいたその場で伝える習慣を持ちます
- ・ 感謝の文化がチームを変える：感謝を伝え合うチームは心理的安全性が高まり、生産性も上がることが研究で示されています

事例から学ぶ：心理的安全性が成功チームを生んだ研究から学べること

ある世界的IT企業が社内で「最も成果を上げるチームの共通点」を調査したプロジェクトでは、メンバーのスキルや経歴より「心理的安全性（失敗や意見を安心して出せる雰囲気）」が最も重要な要因であることが判明しました。

この教訓から学べること：チームの成果は個人の能力の合計ではなく、全員が安心して「本音を言う・間違いを認められる」環境の質で決まります。

入社1年目から「チームの心理的安全性を高める行動」をとることが、チームへの最大の貢献の一つです。

ポイント

コミュニケーションは「生まれつきの才能」ではなく「練習で上手くなる技術」です。最初はぎこちなくても、毎日少しずつ実践すれば自然になります。

失敗を恐れず、誠実に、相手を尊重して話し続けましょう。

それが信頼の構築です。

▶ やってみようワーク・演習

やってみよう追加①：コンフリクト解決のロールプレイ

以下のシナリオで「建設的なコンフリクト解決」を実践します。

シナリオ：あなた（新卒エンジニア）はあるバグの修正方法について、先輩エンジニアと意見が異なる

ります。

先輩は「簡単なパッチを当てて今週リリースする」を主張。

あなたは「根本原因を修正しないと同じバグが再発する、来週まで待つべき」と考えています。

①5ステップのコンフリクト解決プロセスで議論を構成してください

②「技術的に正しいこと」と「チームの文化・関係性を守ること」を両立させるセリフを考えてください

やってみよう追加②：1on1 シミュレーション

来週の1on1（上司との面談）に向けて、以下を準備しましょう。

- ・現在詰まっていること・相談したいこと（1つ）
- ・フィードバックを求めたいこと（自分のコミュニケーションについて）
- ・キャリアについて確認したいこと（1つ）
- ・自分の強みとして上司に見せたい行動（1つ）

6-7 1on1・会議・オンラインコミュニケーション—場面別の実践

コミュニケーションの「場」によって、効果的な振る舞いは変わります。場面ごとのポイントを押さえることで、コミュニケーションの質が大きく上がります。

■ 1on1の活かし方—上司との個別面談を「宝」にする

- ・ 1on1は「評価の場」ではなく「自分の成長を設計する場」：上司に評価されるためではなく、自分の課題・方向性・サポートを確認するための時間です
- ・ 毎回「議題」を持っていく：「最近詰まっていること」「確認したいキャリアの方向性」「フィードバックのお願い」を1～2個準備します
- ・ フィードバックを求める：「今の自分の強みと改善点を教えてください」と積極的に聞きます。評価面談を待たず、普段の1on1で聞くことが重要です

面談は、年に数回しかない貴重な“節目”です。終わったら、話したことと次にやることを一言メモしておく、次の面談につながります。面談を点で終わらせず、線にしていく意識が、成長を加速させます。

- ・ 正直に話す：「うまくいっていないこと」を隠すより、「こんな問題があります、どうアプローチすれば良いですか」と共有する方が信頼を生みます

■ 会議の参加者として—「出るだけ」の参加者にならない

- ・ 事前準備：アジェンダを確認し「自分はこの会議で何を求められているか・何を言うべきか」を考えてから参加します
- ・ 発言する：「何か発言しなければ」というプレッシャーより「自分の視点が議論に役立つか」で判断します。「〇〇について確認してもよいですか」という質問も十分な貢献です
- ・ 議事録を取る：「誰かが取るだろう」ではなく、自分から申し出る、**議事録担当はその会議で最も多くを学べる役割です**
- ・ 会議後のフォロー：自分に割り当てられたアクションアイテムを即座にメモし、期日通りに実行します

■ リモート・オンラインコミュニケーションの実践技術

テレワークが普及した現代では、オンラインコミュニケーションの質がチームの生産性を大きく左

右します。

- ・ 非同期コミュニケーションの質：Slackやメールのメッセージには「背景・依頼内容・期日・優先度」を明記します。「ちょっと確認お願いします」だけでは相手が動きません
- ・ 「既読スルー」をしない：確認したが返答に時間がかかる場合「確認しました。○時までには返答します」という一言が相手の不安を解消します
- ・ オンライン会議のカメラ：基本的にカメラをONにします。顔が見えることで信頼感・参加意識が上がります
- ・ 文字で伝わらないことを意識する：テキストコミュニケーションは声のトーンが伝わりません。批判のつもりはなくても批判に見える文章は書き直す習慣を持ちます
- ・ ドキュメント化の習慣：重要な決定・背景はドキュメント（Notion・Confluence等）に残します。「あの話どこに書いてある？」という状態を防ぎます

6-8 生産性の高いチームで働く—プロとして生産性を意識する

■ 6-8-1 生産性を意識することがプロの条件です

「世の中・顧客に対して、1分1秒でも速く・高品質で・高アウトカムを届ける」を常に意識することが、プロとしての出発点になります。

- ・ 生産性は「頑張る量」ではなく「仕組みと意識」で決まります
- ・ チームとして最大の価値を届けるための働き方を、自分から考える習慣を持ちましょう

■ 6-8-2 「お客様優先」という言葉の本来の意味

- ・ 本来の意味：速く・高品質で・価値のあるものを届け続けることです
- ・ チームの生産性を最大化することが、真のお客様優先につながります

■ 6-8-3 コミュニケーションにはコストがあります—相手の時間を大切にする

コミュニケーションの形	発生するコスト
4人でのMTG1時間	4人分の工数（合計4時間相当）が消費されます
隣の人に5分聞く	自分5分＋相手5分＋相手の集中回復15～20分（合計25～30分相当）
テキストで言語化して非同期で共有	情報が残る・相手の都合の良いタイミングで対応できます

- ・ 非同期コミュニケーションを基本に、対面・音声は目的に合わせて使い分けます
- ・ パッションや感情を含めて伝える場面・関係構築の場面では対面が効果的です

■ 6-8-4 集中できる環境を、チームで守る

- ・ 集中できる環境はチーム全員で守るものです。互いの集中を尊重することが、チーム全体の成果を高めます
- ・ シングルタスク：MTGを特定の日・時間帯にまとめます。短時間タスクはAM、集中作業はPMに行います

■ 6-8-5 情報をオープンにする—知識をチームの財産にする

情報の状態	チームに起きること
情報がオープン	周りの動きを把握でき手伝える・属人化が起きにくくなります
情報がクローズ	問い合わせコストが膨大になります・担当者が休むと仕事が止まります

- ・ 新卒として今日からできること：自分の仕事の進捗・学んだことをチームに共有する習慣をつけます
- ・ エンジニアとしての実践：コードはGitで管理し、変更の意図をコミットメッセージに残します

■ 6-8-6 スペシャリストとして自律的に動く

- ・ スペシャリストとして成長し続けることが、チームへの最大の貢献につながります
- ・ 「自分のタスクが終わったとき、次に何をすべきかを自分で考えて動く」姿勢が、チームの生産性を高めます
- ・ 気になることがあれば改善を提案することもプロとしての大切な行動です

◆ 考えてみよう—倫理の視点

チームへの貢献を意識する、自分の働き方が「チームの生産性に貢献しているか」を定期的に問い直す習慣が、プロとしての成長につながります。

▶ やってみよう—ワーク・演習

やってみよう①：「自分がこれまでいたチームで、何が機能していて何が難しかったか」を振り返り、今日から変えられることを3つ書き出しましょう

やってみよう②：「チームのパフォーマンスに貢献するために今日からできること」を3つ書き出し、グループで共有しましょう

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・ 相手の話を聞いたあと「つまり～ですね」と一度言いかえてみる。

6-9 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ 認識ズレを防ぐには、何をすればよいか。
- ・ 自分も相手も尊重する言い方とは。

- ・ 心理的安全性は、なぜ大切か。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、これからの学びや仕事の中で、少しずつ埋めていきましょう。

第7章 文書作成・プレゼンテーション・会議（11日目後半～13日目・2日）

🎯 この章のゴール

この章では、文書・プレゼン・会議、伝わるように書く・話すコツを学びます。

終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ①【**コツがわかる**】わかりやすい文書は「結論を先に」だとわかる。
- ②【**考え方がわかる**】主張→根拠で組み立てる、とわかる。
- ③【**知っている**】会議の前・中・後でムダをなくす工夫を知っている。

💡 まず自己診断—読む前に、いまの自分を知ろう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「伝わる文書のコツを自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
- ② 「良い資料とは、どんな資料だと思いますか？」を、いまの考えで一言だけ書きます（正解は気にしません）。

読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

✦ よくある勘違いと、その反証

【勘違い】「良い資料＝情報を漏れなく盛り込んだ資料」→

【反証】**良い資料は“相手が判断・行動できる”資料です。**

情報が多いほど要点は埋もれます。

「結論は何か」「相手に何をしてほしいか」が一目でわかることが、網羅性より大事です。丁寧さの方向を、間違えないようにしましょう。

7-1 ビジネス文書の基本

■ ビジネス文書の三原則

原則	意味	実践のポイント
正確（Accurate）	事実・数字に誤りがない	送信前に必ず数字・固有名詞を確認します
簡潔（Concise）	必要な情報だけを・短く	1メール1要件の原則。長くなるなら対面で話します
迅速（Prompt）	必要なタイミングで届ける	返信は24時間以内が基本です

■ メールの書き方

- ・ 件名：「〇〇の件：対応依頼」、何の件か・何を求めているかを件名で伝えます
- ・ 宛先の設定：Toは「対応を求める人」・CCは「情報として知らせる人」
- ・ 本文の構造：挨拶（1行）→主旨（結論を先に）→詳細→お願い事項→署名
- ・ 返信の速度：「確認しました。〇日までに回答します」という速報だけでも送ることで信頼が積み上がります

■ 報告書・議事録の書き方

- ・ 事実と意見・解釈を明確に分けることが最も重要です
- ・ **議事録は「記録」ではなく「合意の証明」です**、決定事項・担当者・期限を必ず明記します
- ・ README、プロジェクトの入り口です。「自分以外の人が初めて見ても使える」を基準に書きます
- ・ ADR（アーキテクチャ決定記録）、「なぜそのアーキテクチャを選んだか」の意思決定の記録です。将来の変更時の判断材料になります
- ・ ポストモートム（事後検証レポート）、障害の事実・根本原因・再発防止策を記録します。「改善機会の記録」として書きます
- ・ これらすべてが「誰でも読めて理解できる」ことを目指します。良いドキュメントがチームの生産性を高めます

📁 事例から学ぶ：メールの誤送信から学べること

宛先ミス・添付ファイルの誤りは、顧客情報漏洩の原因として毎年多数報告されています。この教訓から学べること：送信前の宛先・添付ファイル確認を必ず習慣にすることで、信頼を守れます。

7-2 ビジネスメールの実践—伝わるメールを書く技術

メールは書いただけでは意味がありません。相手を読み、理解し、必要な行動を取れる状態をつくって初めて完了です。ビジネスメールの基本を体系的に身につけましょう。

■ 件名—「開封したくなる・内容がわかる」件名を書く

件名はメールの顔です。多忙なビジネスパーソンは件名だけで「今開くか・後回しにするか・無視するか」を判断します。件名で損をしないために、以下の原則を守りましょう。

- ・ ①何の件か + ②何をしてほしいか を入れる、「〇〇プロジェクト仕様書：確認依頼（期限〇月〇日）」
- ・ 緊急度を示す、「【至急】」「【本日中にご確認ください】」は本当に緊急なときだけ使います。乱用すると信頼を失います
- ・ 返信の件名、「Re: 〇〇の件」はそのまま使ってよいですが、話題が変わった場合は件名を変えましょう
- ・ NGな件名：「お世話になっております」「確認お願いします」「先日の件」、何が何の件かまったく伝わりません

■ 宛先—To・CC・BCCの正しい使い方

- ・ To（宛先）：対応・返信を求める人。複数のToがある場合は誰が主担当かを本文で明記します
- ・ CC（カーボンコピー）：情報として共有する人、対応は不要だが知っておいてほしい人。上司・関係部署など
- ・ BCC（ブラインドカーボンコピー）：他の受信者に知られずに送りたい人、大量一斉送信時のプライバシー保護に使います。社内ではほぼ使いません
- ・ 注意：「全員に返信」ボタンは慎重に、CCに入っている全員へ返信が必要か確認してから押しましょう
- ・ 外部への送信前チェック：宛先に間違いがないか・CCに余計な人が入っていないか・添付ファイルの内容は正しいか、この3点を毎回確認します

■ 本文の構造—「読みやすい順番」で書く

ビジネスメールの本文は、相手を読む時間を最小化しながら、必要な情報をすべて伝える構造が理想です。以下のテンプレートを体に覚えさせましょう。

- ・ ①挨拶（1～2行）：「お世話になっております。〇〇部の△△です。」
- ・ ②用件の核心（1～3行）：最初に「何のメールか・何をしてほしいか」を書きます。「〇〇についてご確認いただきたく、ご連絡しました」
- ・ ③詳細・背景（必要に応じて）：箇条書きを活用して読みやすくします
- ・ ④期限・アクション（あれば）：「〇月〇日（〇）までにご回答いただけますと幸いです」
- ・ ⑤締めの一言：「お忙しいところ恐れ入りますが、どうぞよろしくお願いいたします」
- ・ ⑥署名：氏名・部署・連絡先を必ず入れます

■ 返信のマナー—「速さ」と「丁寧さ」のバランス

- ・ 返信の基本：24時間以内が目安、確認に時間がかかる場合でも「確認しました。〇日までに回答します」と速報を送ります
- ・ 即答できないとき：「ご連絡ありがとうございます。〇〇について確認が必要なため、〇日（〇）中にご回答いたします」
- ・ 社外からのメール：業務時間内に届いた場合は当日中、業務時間外は翌営業日の午前中が目安です
- ・ 返信不要のメールへの対応：CCで届いた情報共有メールや「返信不要」と書かれたメールには、原則として返信不要です。ただし疑問・追記が必要な場合は返信してよいです

7-3 報告書・議事録・技術文書の書き方

エンジニアとして働く上で、コードを書くだけでなく「書く力」は不可欠です。**報告書・議事録・技術文書の質が、チームの生産性と信頼に直接影響します。**

■ 報告書—「読んだ人が次の判断をできる」文書

報告書は上司・関係者が意思決定をするための材料です。「何が起きたか・なぜか・どうすべきか」が明確に伝わる構成を意識します。

- ・ 基本構成：①件名（何の報告か）②概要（3行以内で要点）③詳細（事実・原因・対応）④今後の対応・提案⑤添付資料（あれば）
- ・ 事実と意見を必ず分ける、「〇〇が起きました（事実）」と「原因はおそらく〇〇と考えられます（推測）」と「対応策として〇〇を提案します（意見）」は必ず分けて書きます

- ・ 数字を使う、「遅延した」より「3日遅延した（予定7/10→実際7/13）」の方が明確です
- ・ 長さの目安：A4用紙1～2枚に収める、詳細資料は添付します

■ 議事録—「合意の証明」として書く

議事録は「会議で何が決まったか」を証明する文書です。後から「言った・言わない」のトラブルを防ぐ最も有効な手段でもあります。

- ・ 必須項目：①日時・場所・参加者②決定事項（明確に・番号で整理）③アクションアイテム（担当者・期限を必ず明記）④次回会議の日程（あれば）
- ・ 決定事項の書き方：「〇〇については〇〇とすることが決定された」、受け身形にすることで「誰かの意見」ではなく「会議の決定」として記録されます
- ・ アクションアイテムの書き方：「〇〇さんが〇月〇日までに〇〇を行う」、担当者・期限・内容の3点セットで必ず書きます
- ・ スピード：会議終了後24時間以内に共有、記憶が鮮明なうちに作成します。会議中にメモを取り、終了直後に仕上げる習慣をつけましょう
- ・ 確認：送付後に参加者から「相違なし」を得る、特に重要な決定事項は「内容に相違ないかご確認ください」と一言添えます

あわせて、会議を主催する側や参加者にも、決定事項をその場で明確に言語化する責任があります。

■ エンジニアが書く技術文書の実践

エンジニアにとって、コードと同様に重要なのが技術文書です。「誰でも読めて使える」文書が、チームの生産性を高め、引き継ぎを楽にします。

- ・ README、プロジェクトの入り口：含めるべき5項目：①このプロジェクトは何か（1段落）②動かすのに必要な環境・前提条件③セットアップ手順（コマンド付き）④使い方の例⑤ライセンス・連絡先
- ・ コミットメッセージ、「なぜ変えたか」を書く：「fix bug」「update」は最悪のコミットメッセージです。「feat: ログイン時のエラーメッセージを日本語化（Issue #123対応）」のように変更の意図を書きます
- ・ ADR（アーキテクチャ決定記録）、意思決定の記録：含める項目：①タイトル②ステータス（提案中/承認済/廃止）③背景（なぜこの決定が必要だったか）④決定内容⑤考慮した代替案⑥結果（この決定による影響）
- ・ ポストモーテム、障害の学習記録：含める項目：①発生日時・影響範囲②タイムライン（何時に何が起きたか）③根本原因④対応内容⑤再発防止策⑥ラッキーだったこと（より悪化を防いだ要因）

7-4 伝わるプレゼンテーション

■ 「話す」と「伝える」は違います

プレゼンテーションの目的は「伝えること」であり「話すこと」ではありません。受け手を起点に設計します。

- ・ 準備の問い：「この発表を聞いた後、聴衆にどんな行動・判断をしてほしいか」
- ・ **相手が知りたいことと、自分が言いたいことは必ずしも一致しません**

- ・ エンジニアが非エンジニアに技術的内容を説明するとき：「この機能でユーザーができるようになること」から始めます

■ プレゼンの構造—ピラミッド原則

層	内容	例
頂点（結論）	最初に言いたいことを1文で	〇〇システムの導入を提案します
第2層（理由）	なぜそう言えるか（3つ程度）	①コスト削減 ②効率化 ③セキュリティ向上
第3層（根拠）	各理由の具体的な根拠・データ	①現在の年間コストは〇〇円。導入後は〇〇円に削減できます

■ 資料作成の基本

- ・ 1スライド1メッセージ、1枚のスライドで伝えることは1つだけです
- ・ 文字を減らします。**スライドは読むものではなく見るものです**
- ・ 数字とグラフの使い方、「何を言いたいか」を先に決めてからグラフの種類を選びます

▶ やってみようワーク・演習

やってみよう：5分間プレゼンを実際にやってみましょう

テーマ：「自分が入社前に学んでいた技術・スキル、それが仕事にどう使えるか」

相互フィードバック：「結論が最初にあったか」「専門用語がわかりやすかったか」「聴衆の視点で設計されていたか」の3点で評価し合いましょう

7-5 プレゼンテーションの実践技術—「話す力」を磨く

資料を作ることと、実際に人前で話すことは別のスキルです。プレゼンテーションは練習によって確実に上達します。**技術エンジニアであっても、自分の仕事の価値を伝える力は不可欠です。**

■ 聴衆分析—「誰に話すか」で内容が変わる

同じ技術的内容でも、エンジニア向けと経営者向けでは全く異なる伝え方が必要です。発表前に必ず「誰に話すか」を明確にしましょう。

- ・ 技術者向け：専門用語を使ってよい・実装の詳細を説明する・パフォーマンスのデータを見せる
- ・ 非技術者（経営者・営業）向け：技術用語は使わない・ビジネスへの影響（コスト・時間・リスク）で話す・「この機能でユーザーが〇〇できるようになります」という形で伝える
- ・ 混在している場合：非技術者に合わせて話し、技術的詳細は「補足資料」に回す、「詳しくはスライドの補足をご覧ください」で対応します

■ スライド設計の実践—「見やすい」スライドのルール

- ・ 1スライド1メッセージ：スライドは「言いたいことを見せるもの」です。1枚に詰め込みすぎると、何を言いたいかが伝わりません

- ・ タイトルは結論を書く：「〇〇について」ではなく「〇〇により年間〇〇万円削減できます」、見出しを見るだけで内容が分かる状態が理想です
- ・ 文字は少なく：箇条書きは1項目15字以内が目安、スライドは「読む」ものではなく「見る」ものです。詳細は話し言葉で補います
- ・ グラフの使い方：棒グラフ（比較）・折れ線グラフ（推移）・円グラフ（構成比）、何を言いたいかを先に決めてからグラフを選びます
- ・ 色の使い方：2～3色に絞る。強調には赤・注意喚起には黄・ポジティブには青や緑、色の意味を統一して使います

■ 発表の実践技術—本番で緊張しないために

- ・ 準備、練習あるのみ：声に出して3回以上練習します。時間を計って「速すぎないか・遅すぎないか」を確認します
- ・ 話し始め：最初の30秒で「今日何を話すか・聴衆に何をもち帰ってほしいか」を明確に伝えます。聴衆の集中力が最も高いのは最初です
- ・ アイコンタクト：1人を2～3秒見てから次の人へ。全体をゆっくりスキャンするように視線を動かします。特定の人だけを見続けると、他の参加者が疎外感を感じます
- ・ 話す速度：緊張すると速くなります。意識的にゆっくり、聴衆が理解できる速度で話すことが最優先です
- ・ 質問への対応：「よい質問です」という決まり文句は避けます。わからないことは正直に「確認して後ほどご連絡します」、**知ったかぶりは信頼を大きく損ないます**

7-6 会議の進め方

■ 会議の種類と目的

会議の種類	目的	望ましい参加者数
意思決定会議	重要な判断をする	少数精鋭（3～5人）
情報共有会議	情報を伝える・認識を合わせる	関係者全員
問題解決会議	課題の原因を探り、解決策を考える	関係者（5～8人）

「今日の会議は意思決定会議か、情報共有会議か」を最初に明確にするだけで、会議の質が大きく上がります。

■ 参加者としての基本姿勢

- ・ 準備：アジェンダを事前に確認し、自分の意見・質問を準備して臨みます
- ・ 発言：必ず何か貢献します。発言することがチームへの貢献になります
- ・ フォローアップ：会議後に決定事項・自分のアクションアイテムを確認・実行します

■ ファシリテーションの基本

- ・ 時間管理：開始・終了時間を守ります。議論が脱線したら「〇〇の議題に戻しましょう」と引き戻します
- ・ 論点の整理：「今議論しているのは〇〇の問題ですね」と確認しながら進めます
- ・ 合意の確認：決まったことを「では〇〇は〇〇ということで合意ですね」と確認します

▶ やってみようワーク・演習

やってみよう：模擬会議を実施しましょう

- ①意思決定会議のシミュレーション：「次のプロジェクトに採用するフレームワークをAとBのどちらにするか」をテーマに、ファシリテーター役・参加者役に分かれて実施します
- ②振り返り：「目的は達成されたか」「時間管理はできたか」「全員が発言できたか」「合意事項が明確になったか」の4点で評価します

7-7 会議をムダにしないーファシリテーションと参加者の技術

「会議が多くて仕事が進まない」状況は多くの職場で生じています。**会議そのものの生産性を上げることが、チーム全体の生産性改善に直結します。**

■ 会議の「前」にすることー準備が会議の質を決める

- ・ アジェンダの事前配布：会議の少なくとも24時間前に、「目的・議題・決定したいこと・参加者への事前準備依頼」を送ります
- ・ 招待リストの精査：「この人が参加しないと決まらない」人だけを呼びます。傍聴のためだけの参加者は議事録で代替できます
- ・ 資料の事前共有：30分の会議で資料を初めて見せるのは時間の浪費です。事前に共有し「読んできてください」とお願いします
- ・ 参加者として：アジェンダを確認し、自分が貢献できることを1つ考えて参加します

■ 会議の「中」にすることーファシリテーションの基本

- ・ 時間を守る：開始・終了時間を厳守します。終了5分前に「残り5分です。決定事項を確認します」と声をかけます
- ・ 目的の宣言：会議の冒頭で「今日の会議のゴールは〇〇を決めることです」と宣言します。これだけで脱線が減ります
- ・ 発言を引き出す：「〇〇さん、この点についてはいかがですか」と指名することで、発言しにくい人の意見を拾います
- ・ 脱線を止める：「それは重要な話題ですが、今日のアジェンダ外です。別途話し合う機会を設けましょう」と言える勇気が必要です
- ・ 決定事項の確認：議題が終わるたびに「では〇〇については〇〇とすることで合意ですね」と確認します。**「たぶん合意した」で終わる会議が最もリスクが高いです**

■ 会議の「後」にすることーフォローアップで価値を最大化する

- ・ 議事録の送付：会議終了後24時間以内に送付します。「決定事項・アクションアイテム（担当者・期限）・次回日程」を含めます
- ・ アクションアイテムの実行：自分が担当になったことは必ず実行します。「言いっぱなし・決まりっぱなしの会議」を生み出さないことが信頼の源です
- ・ 振り返り：長い会議・重要な会議の後は「目的を達成できたか」「改善できる点はあったか」を簡単に振り返ります

■ 会議を減らすためにー非同期コミュニケーションの活用

すべての情報共有を会議でやる必要はありません。「何を会議でやるべきか・何をチャットやドキ

メントで代替できるか」を常に問いましょう。

- ・ 会議に適したもの：感情的な話し合い・複雑な問題の合意形成・関係構築・ブレインストーミング
- ・ 非同期で代替できるもの：一方向の情報共有・単純な確認・データの共有・意見収集（アンケートで可）
- ・ 会議を減らす実践：週次報告をドキュメント共有に変える・「相談がある」をSlackスレッドで完結させる・定例会議の頻度を毎週→隔週に見直す

事例から学ぶ：「議事録のない会議」が引き起こすトラブルから学べること

ある開発プロジェクトで、要件についての合意が口頭で行われ議事録に残らなかった結果、3ヶ月後に「そんな話はしていない」という認識の食い違いが発生。

追加開発に数百万円のコストと2ヶ月の遅延が生じた事例があります。

この教訓から学べること：会議での決定事項を文書化することは、面倒な作業ではなく、チームと顧客を守るためのプロとしての習慣です。

議事録を書く習慣が、組織全体の信頼と生産性を守ります。

ポイント

「書く力」と「話す力」は、技術力と同様にエンジニアの市場価値を決めます、自分の仕事の価値を言語化できる人は、上位の仕事を任せられ、信頼され、より大きなプロジェクトに関われるようになります。

今日から「書く・話す」を技術と同じように磨いていきましょう。

▶ やってみようーワーク・演習

やってみよう①：メールの添削練習

以下の「悪いメール」を「良いメール」に書き直してみましょう。

【悪いメール例】

件名：確認お願いします

本文：お世話になっております。先日お話しした件なのですが、どうなりましたでしょうか。もし可能であれば早めに教えていただけると助かります。

→ 問題点を3つ挙げた上で、良いメールに書き直してください。

やってみよう②：議事録を書く練習

ファシリテーター役・参加者役に分かれて15分の模擬会議を行い、参加者の一人が議事録を取ります。

テーマ：「来月のチームのオフサイトイベントの内容を決める」

会議後に議事録を全員で確認し、「決定事項が明確か」「アクションアイテムに担当者・期限があるか」を評価します。

やってみよう③：技術内容を非技術者に伝える練習

自分が理解している技術トピックを1つ選び、「技術を知らない親・祖父母」に伝えるつもりで1分間で説明してみましょう。

評価基準：専門用語を使っていないか・何が嬉しいのかが伝わるか・具体的な例を使っているか

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・ 短い連絡文を1つ書き、「結論が先頭か」「一文が長すぎないか」を見直す。

7-8 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ わかりやすい文書のコツを、一言で。
- ・ 話や資料は、どんな順で組み立てるとよいか。
- ・ 会議のムダをなくす工夫を、思い出せるだけ。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、これからの学びや仕事の中で、少しずつ埋めていきましょう。

第2部 総合ワークショップ（14～15日目・2日）

第2部（第4～7章）で学んだ「報連相・仕事の進め方・品質・コミュニケーション・文書作成・会議」を統合する大型実践演習です。2日間で「実際の仕事の流れ」を体験し、「知識として知っていること」が「実際の行動として出るか」を確認します。

■ 1日目：模擬プロジェクト演習（仮想プロジェクト参加）

4～5人のチームに分かれて、仮想プロジェクト「社内の勤怠管理システムの改善提案」に取り組みます。

- ・ 午前：プロジェクト発足、仕事の依頼を受ける・目的と背景を確認する・タスクを分解する・担当を決める・スケジュールを立てる（第5章の実践）
- ・ 午後：中間報告、途中で「仕様変更」という想定外の事態が発生します。進捗報告・問題の相談をPREP法・四点セットを使って実践します（第4章の実践）
- ・ 夕方：ドキュメント作成、現状分析・改善提案をまとめた報告書（A4・1枚）を作成します（第7章の実践）

■ 2日目：発表とフィードバック

- ・ 午前：プレゼンテーション準備、前日の報告書をもとに5分間のプレゼン資料を作成します。聴衆分析（社内の技術を知らない役員向け）を意識したスライド設計（第7章の実践）
- ・ 午後：グループ発表、各チームが5分間のプレゼンを実施します。発表後にフィードバック（よかった点2つ・改善点1つ）を相互に行います
- ・ 振り返り：チームワーク・コミュニケーション・ドキュメント・プレゼンについてKPT（Keep/Problem/Try）でチーム振り返りを実施します（第7章の実践）

▶ やってみようーワーク・演習

▶ 第2部総合ワークショップ 個人振り返り

2日間の演習を終えて、以下の問いに個人で答えましょう。

- ①「指示を受けるとき」で最も改善が必要だと感じたことは何か
- ②報連相のどの場面で「うまくできた」「うまくできなかった」と感じたか
- ③チームで仕事をする上で「自分の強み」として活かしたことは何か
- ④ドキュメント・プレゼンについて、今後重点的に練習したいことは何か
- ⑤第2部全体を通じて、明日から「1つだけ変える行動」を決めましょう

第3部 考える力・数字の力（16～20日目・5日間）

第3部では、論理的に考える力と、数字を読む力を学びます。ビジネスパーソンの判断は、最終的にこの2つの上に成り立ちます。技術力だけでは届かない「ビジネス上の判断」に踏み込むための、新卒に必須の土台です。

勢いや感覚で動くのではなく、筋道を立てて考え、数字で語れること。この二つは、職種や時代が変わっても古びない力です。むしろ、**AIがすぐに答えらしきものを出す時代だからこそ、その答えが正しいかを見抜き、根拠を数字で確かめられる人の価値が上がります。**ここで学ぶのは、一生使える「考える土台」です。

第8章 ロジカルシンキングと問題解決（16～17日目・2日）

🎯 この章のゴール

この章では、ロジカルシンキングと問題解決、筋道立てて考えるコツを学びます。

終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ①【考え方がわかる】「もしなくダブリなく」分ける考え方がわかる。
- ②【コツがわかる】「なぜ」を繰り返すと根本原因に近づく、とわかる。
- ③【違いがわかる】事実・意見・推測の違いがわかる。

💡 まず自己診断—読む前に、いまの自分を知ろう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「問題解決の考え方を自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
 - ② 「「問題を解決する」とは、まず何をすることだと思いますか？」を、いまの考えで一言だけ書きます（正解は気にしません）。
- 読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

8-1 ロジカルシンキングの基本

■ 論理的に考えるとはどういうことか

論理的思考とは「根拠に基づいて結論を導くこと」です。この習慣により、説明力・説得力・問題解決力が向上します。

- ・ 「そう思う」ではなく「なぜそう言えるか」を常に問う習慣を持ちます
- ・ 結論→根拠→根拠の根拠という「ピラミッド構造」で思考を整理します

論理的に考えるのは、難しい言葉を使うためでも、相手を言い負かすためでもありません。「自分の考えを、相手に分かってもらう」ための思いやりです。筋道が通っていれば、相手は安心して理解でき、納得もしやすい。ロジカルシンキングは、相手のための“伝える技術”でもあるのです。

✦ 考えてみよう：説明がうまい人

あなたの周りで「この人の説明は分かりやすい」と感じる人を思い浮かべてみましょう。その人は、何をどう話しているのでしょうか。

たいてい「結論が先」「理由が明確」、それが、ロジカルに伝える、ということです。

■ 「なぜ」を5回問う（なぜなぜ分析・5Why分析）—問題の根っこを掘る習慣

表面に見えている問題の下には、必ず根本原因があります。「なぜ」を繰り返して掘り下げること、で、本当の問題に近づけます。「なぜ」を5回ほど繰り返すことから「なぜなぜ分析（5Why分析）」と呼ばれ、製造業の品質管理から生まれた手法で、ソフトウェア開発の障害分析でも広く使われています。

- ・ 例：システムが遅い（問題）
- ・ →なぜ？：データベースのクエリが遅い
- ・ →なぜ？：インデックスが適切に設定されていない
- ・ →なぜ？：設計時にデータ量の増加を想定していなかった
- ・ →なぜ？：要件定義の段階で、将来のデータ量を見積もる工程がなかった
- ・ →なぜ？：性能・拡張性などの非機能要件をレビューする仕組みがチームになかった

「なぜ5回」の最後の答えが「本当に対処すべき問題（真因）」です。根本原因に対処すれば、同じ問題の再発を防げます。回数は5回が目安です。真因にたどり着けば3～4回でも、足りなければさらに繰り返します。

▶ なぞって覚える—なぜなぜの“浅い／深い”を見分ける

【浅い例（NG）】重い→なぜ？

「アクセスが多いから」→なぜ？

「人気だから」… これは“言い訳の連鎖”で、自分たちで打てる手に着地しません。

【深い例（OK）】上の本文のように、最後が「レビューする“仕組み”がなかった」のように“自分たちで変えられる対策”に着地します。

チェック：あなたのなぜなぜの最後の答えは、「明日から自分／チームで変えられること」になっていますか？ なっていないければ、まだ途中です。

■ 「なぜ」をもう一步深める—直接的原因と動機的原因（ある製造業で用いられてきた反省法）

「なぜ」を繰り返すと技術的な原因にはたどり着けますが、それだけでは再発防止が十分でないことがあります。ある大手メーカーの品質保証では、「なぜなぜ分析」をさらに一步進め、原因を「直接的原因」と「動機的原因」の2つに分けて考えます。

- ・ 直接的原因：不具合と物理的・論理的に直結する技術的な原因です。調査・解析で客観的に特定でき、対処すればその不具合は解決します。ただし効果はその製品・その事象に限定されがちです。
- ・ 動機的原因：その不具合を作り込んでしまった、あるいは見逃してしまった人の「心の働きの不足」（思考の不足・思い込み・確認不足・判断の誤り・他人任せなど）です。ここに対策を打つと、同じ不具合だけでなく、ほかの仕事でも似た失敗を防げます。

この動機的原因を引き出すために使うのが「反省法」です。次の3つを順番に問います。

- ・ ①何が悪かったのか？「これが悪かったので問題が起きた」
- ・ ②どうすれば良かったのか？「こうすれば良かった」
- ・ ③なぜ、そうしなかったのか？ここで出てくる「心の働きの不足」が動機的原因です

動機的原因は、(a) 再発防止に広く応用できる本質的なもの、(b) 他人や他部署に責任を求めないもの、(c) 現実的で実行できる対策に結びつくもの、を基準に選びます。大切なのは、自分に都合の悪い事実を避けず、正直に振り返る姿勢です。技術的な「なぜ」と、人の判断に踏み込む「反省法」を組み合わせることで、本当に効く再発防止につながります。

ただし「他部署に責任を求めない」は、自分の改善点を直視するための姿勢であって、組織の仕組みの問題を個人の心がけにすり替えてよい、という意味ではありません。動機的原因が見えたら、個人の反省で終わらず、同じ失敗を誰がやっても防げる仕組みへと、組織の側が落とし込む責任があります。

✦ よくある勘違いと、その反証

【勘違い1】「論理的＝難しい言葉で複雑に話すこと」→

【反証】論理的とは“誰でも筋道を追える”こと。

むしろ複雑さを削ることが論理力です。

【勘違い2】「問題解決＝すぐ良い解決策を出すこと」→

【反証】**問題解決は“問題の定義が9割”。**

原因が特定できていない段階で対策に飛びつくと、症状を叩いて真因を見逃します。

早いのは思考停止、速いのは正しい順番で進むことです。

■ 事実・意見・推測を分けて話す・書く

種類	定義	見分け方	例
事実	確認・計測できる客観的な情報	誰が確認しても同じ結果	応答時間が3秒以上かかっています
推測	確認されていないが可能性が高い	「おそらく」「～だと思われる」	おそらくDBのクエリが原因だと思います
意見	個人の判断・評価・提案	「～すべき」「～がよい」	インデックスを追加すべきだと思います

この3つを区別して話すことで、相手の理解が深まり、議論の質が上がります。

■ MECEの概念—「漏れなく・ダブリなく」整理する

MECE（Mutually Exclusive, Collectively Exhaustive）とは「互いに重複せず、全体として漏れない」状態のことです。問題の整理・分析に使う思考ツールです。

- ・ 漏れがあると：大事な問題・原因を見逃す可能性があります
- ・ ダブリがあると：同じことを二重に考えて効率が下がります
- ・ 実践例：バグの原因を「フロントエンド / バックエンド / インフラ / データ」に分類します

■ ロジックツリー—問題を構造的に分解する

ロジックツリー／なぜなぜ分析－問題を分解して「根っこ」を探す

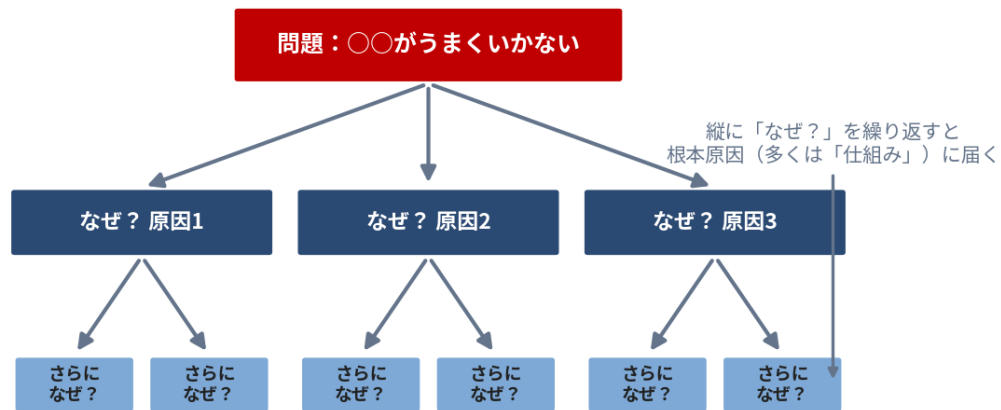


図 ロジックツリー／なぜなぜ分析（問題を分解する）

- ・ Whyツリー（なぜなぜ）：原因を掘り下げます
- ・ Howツリー（どうやって）：解決策を展開します
- ・ Whatツリー（何が）：要素を分解します

▶ やってみようワーク・演習

やってみよう：日常の問題をロジックツリーで分解してみましょう

テーマ：「自分のチームの開発速度を向上させるために、どこから取り組むべきか」 MECEで原因を分類→各原因についてなぜなぜを5回→優先順位をつけます

8-2 問題解決の手法

■ 問題解決の4ステップ

ステップ	内容	成功のポイント
①問題の定義	「何が問題か」を正確に捉える	症状と問題を混同しません
②原因の特定	なぜなぜ分析・特性要因図で根本原因を探る	表面的な原因で止まりません
③解決策の立案	仮説思考・アイデア発散と収束	複数の選択肢を比較します
④実行と検証	PDCAで回す・効果を測定する	実行後に必ず確認します

▶ 分岐ケース—どちらが「問題の定義」？

状況：「サイトが重い」とクレームが来た。

A：すぐサーバを増強する（重い＝サーバ不足だろう）

B：「いつ・どの画面・どの操作で・どれくらい遅いか」を先に特定する

Aは、もし原因が特定APIのN+1問題なら、サーバを増やしても改善せず費用だけ増えます。

Bは“真因に効く一手”にたどり着けます。

“症状（重い）”と“問題（どこで何が起きているか）”を分けることが、4ステップの①の正体です。

💡 つながる—なぜなぜ／反省法は、現場の「障害対応」で使います

ここで学ぶなぜなぜ分析（技術的な真因）と反省法の動機的原因（なぜ防げる仕組みにできなかったか）は、第23章23-8（運用と保守）や第26章26-7（インシデント対応）でそのまま使う道具です。研修の演習で“型”を作り、現場のポストモーテム（事後検証）で“本番投入”する、この往復が、知識を実力に変えます。

■ 仮説思考—「答えを出してから検証する」という発想

問題解決において、「仮説を立て、仮説を検証するために必要な情報を集める」アプローチが効率的です。

- ・ 仮説があれば「この情報があれば確認できる」という目的が生まれます
- ・ **仮説が間違っているても良いです。「検証によって否定された仮説」も重要な情報になります**

■ 「現場を見る」と「数字で考える」の両方を持つ

- ・ 現場の感覚：「データが正しいと思っていたが、実は測定方法が問題だった」という見落としを防ぎます
- ・ データの分析：感覚的なバイアスに引きずられることなく客観的に判断できます
- ・ エンジニアの実践：ログ・メトリクス・ユーザーフィードバックを組み合わせで判断します

📦 事例から学ぶ：なぜなぜ分析の価値から学べること

ソフトウェア開発・医療・航空など多くの分野で、原因分析を徹底した組織は問題の再発率が大幅に低下しています。

この教訓から学べること：症状を直すだけでなく根本原因まで掘り下げることが、長期的な品質向上と信頼につながります。

▶ やってみよう—ワーク・演習

やってみよう：与えられた問題を手順通りに解いてみましょう 問題：「ECサイトの購買転換率が先月比20%低下した。」

原因を特定し、対策を提案せよ」

①問題の定義→②原因の特定→③解決策の立案→④発表（ピラミッド構造で）

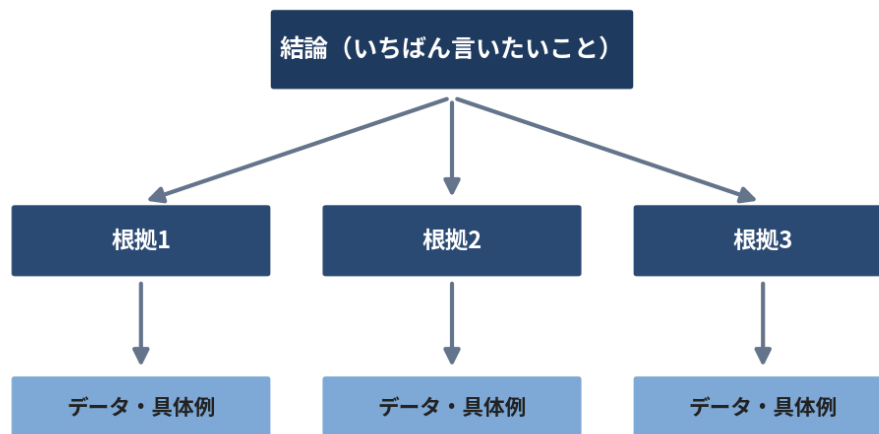
問題解決でいちばん大事なのは、実は「解く」ことより「何が問題かを正しく決める」ことです。

「症状」と「本当の問題」を混同せず、急がば回れで最初に問題をていねいに定義することが、解決の半分以上を占めます。

8-3 ピラミッドストラクチャー—「伝わる論理」の設計図

論理的に考えるだけでなく、「論理的に伝わる」ことが仕事では重要です。ピラミッドストラクチャーは、複雑な情報を相手が理解しやすい構造に整理するための強力な道具です。

ピラミッドストラクチャー—まず結論、次に根拠、最後にデータ



上から下へ「なぜ?」、下から上へ「だから?」でつながっているか確かめる

図 ピラミッドストラクチャー（結論→根拠→データ）

■ ピラミッドストラクチャーとは何か

ある外資系コンサルティング会社のコンサルタント、バーバラ・ミントが体系化した「考えを整理して伝えるための構造」です。頂点に「結論（メッセージ）」があり、その下に「根拠」「根拠の根拠」が積み上がる三角形の構造になっています。

- ・ 頂点：「結論・主張」、相手に一番伝えたいメッセージ。たとえ後の説明を聞かなくても、これだけは伝わるようにします
- ・ 第2層：「根拠（キーメッセージ）」、結論を支える主な理由。通常3～5点にまとめます
- ・ 第3層以下：「サポートデータ・事実・分析」、各根拠を裏付ける具体的な情報

■ 演繹法と帰納法—2種類の論理の組み立て方

ピラミッドを構成する論理の積み上げ方には、2つのパターンがあります。

- ・ 帰納法（下から上へ）：複数の事実・観察から共通点を見つけて結論を導く、「A・B・Cという事実があるから、〇〇という結論が導ける」という思考です
- ・ 演繹法（上から下へ）：一般的なルール・前提から特定の結論を導く、「AはBである。BはCである。したがってAはCである」という三段論法の形です
- ・ エンジニアの実践：バグ調査では「複数のログ（事実）から原因を推論する」帰納法と「このバグのパターンは〇〇のはずだから確認する」演繹法の両方を組み合わせます

■ PREP法—ビジネスでの「伝える型」

結論ファーストを型にしたのがPREP法です（P＝結論→R＝理由→E＝具体例→P＝結論。型と例

は第4章4-2で詳しく扱いました)。

PREP法 — 結論ファーストの型

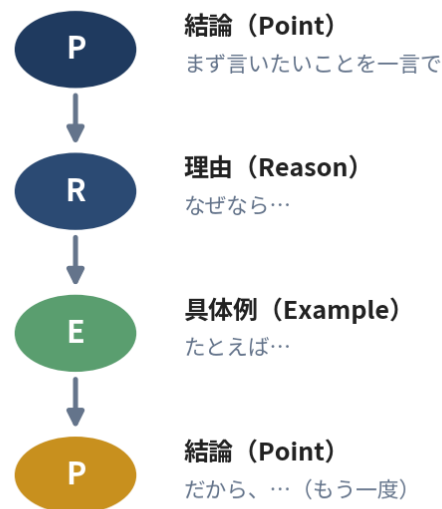


図 PREP法 (結論ファーストの型)

- ・ 使い方の例：「この機能のリリースを1週間延期すべきだと考えます (P)。なぜなら、テスト中に発見したバグが個人情報に影響するリスクがあるからです (R)。たとえば、特定の条件でパスワードが平文でログに記録されます (E)。したがって、修正完了後にリリースすることを強くお勧めします (P)」

伝わる話し方の基本は、「結論から言う」ことです。日本語はつい背景から話しがちですが、ビジネスでは、まず結論、次に理由、最後に具体例。PREP法やピラミッド構造は、要するに「結論ファースト」を形にしたものです。難しく考えず、まず結論から、を意識しましょう。

8-4 プレゼンテーションの論理構造—「話す論理」と「書く論理」

報告・提案・プレゼンテーションでは、「何を言うか」と同じくらい「どう構成するか」が重要です。**聴衆が迷子にならない論理の流れを設計することが、伝わるプレゼンの条件です。**

■ プレゼンの構成—「なぜ今これを話すのか」から始める

- ・ 問題提起から入る：「現在〇〇という状況があります (現状)」「これにより〇〇という問題が生じています (問題)」から始めると、聴衆が「だから聞く必要がある」と感じます
- ・ SCQA構造 (外資系コンサルティングで広まった型)：Situation (状況) → Complication (問題・変化) → Question (問い) → Answer (答え)、この流れで話すと、聴衆が自然に結論を「聞きたい」状態になります
- ・ 「だから何? (So What?)」を常に意識する：各スライド・各段落の後に「だからどうしてほしいのか」が明確になっているか確認します

■ 技術的な説明を非技術者に伝える—翻訳の技術

エンジニアに最も求められるコミュニケーションスキルの一つが、「技術内容をビジネス言語に翻訳する力」です。

- ・ 技術用語を使わない：「DBのインデックスが最適化されていない」→「いまの検索が遅い原因はこの仕組みです。直せば速くなり、ユーザーの離脱も減ります」
- ・ 「Why→What→How」の順番：まず「なぜこれが必要か」（ビジネスインパクト）を伝えてから「何をするか」「どうするか」の技術詳細に移ります。逆順で話すと聴衆が途中で「なぜ聞いているのか」を見失います
- ・ 数字でビジネスインパクトを示す：「パフォーマンスが向上する」より「ページロードが2秒速くなり、売上転換率が5%改善する」の方が判断者を動かします

事例から学ぶ：「根拠のない自信」vs「根拠のある提案」—エンジニアの提案力が結果を変えた事例

あるスタートアップの新卒エンジニアが「今のシステムアーキテクチャは将来スケールしない」という問題を発見しました。

A案は「なんとなくこれが問題だと思います」というSlackの一言投稿。

B案は「現在のアーキテクチャの月次成長率をグラフ化し、6ヶ月後に想定されるボトルネックをロジックツリーで特定、改善案3つの工数・リスク・効果を比較した1ページのドキュメント」として提案しました。

Bのエンジニアの提案は翌週の技術会議で採択されました。

この教訓から学べること：問題を発見する力と同等か、それ以上に「問題を論理的に整理して伝える力」が、エンジニアの影響力を決めます。

ポイント

ロジカルシンキングは生まれつきの才能ではなく、反復練習で身につく技術です。「なぜ？」を問い続ける・事実と意見を分ける・結論から話す・相手の反論を想定する。これらを毎日の仕事の中で意識し続けることが、論理的思考力を鍛える唯一の道です。

▶ やってみようワーク・演習

やってみよう追加①：ピラミッドストラクチャーで提案を作る

以下のシナリオに対して、ピラミッドストラクチャーを使って提案を作りましょう。

シナリオ：あなたの開発チームは毎週1〜2件の本番バグを発生させています。

チームリーダーに「この問題を改善したい」と提案したいです。

【ピラミッドを作ってください】

頂点（結論）：〇〇すべきだ

根拠1：（なぜなら〇〇だから）→ サポートデータ：

根拠2：（なぜなら〇〇だから）→ サポートデータ：

根拠3：（なぜなら〇〇だから）→ サポートデータ：

予想される反論と回答：

やってみよう追加②：「クリティカルシンキング」実践

以下の主張に対して「どこが怪しいか」を指摘してください。

主張：「先月から毎朝コーヒーを飲み始めた。先月から仕事のパフォーマンスが上がっている。つま

- りコーヒーがパフォーマンスを上げている」
- ①この主張の論理的な問題点を2つ指摘する
 - ②この主張を「正しく検証する」にはどんな方法があるか

8-5 数字と論理で「説得力」を上げる—根拠のある主張

ビジネスでは「感覚」より「根拠のある主張」の方が動きます。 データと論理を組み合わせることで、提案の説得力が上がります。

とはいえ、「根拠で固める」とことと「相手を論破する」とことは別です。目的は、勝つことではなく、相手と一緒に良い結論にたどり着くこと。根拠は、相手を打ち負かす武器ではなく、納得してもらうための“橋”だと考えましょう。

■ 定量的根拠と定性的根拠の組み合わせ

- ・ 定量的根拠（数字）：「エラー率が先月比30%増加しています」「このバグは1日平均50件のサポート問い合わせを引き起こしています」、数字は客観的で否定しにくい根拠になります
- ・ 定性的根拠（事例・証言）：「ユーザーから『〇〇の操作がわかりにくい』という声が多数届いています」、数字では表れない「質的な問題」を補完します
- ・ 組み合わせの力：「数字+事例」で主張することで、「頭では理解できる+感情的にも納得できる」の両方が実現します

■ 反論への備え—「相手が反論するとしたら何か」を考える

強い提案は「反論への準備」がされています。相手の視点に立って「このアイデアの弱点は何か」を自分で考えることが、思考の質を高めます。

- ・ Steelman（最強の反論）の思考：「相手の立場から考えられる最も強い反論」を想定します。弱い反論への備えより、強い反論を乗り越える準備の方が議論を強化します
- ・ 反論の先取り：「この提案に対して『コストがかかりすぎる』という懸念があるかもしれませんが、〇〇という理由でコスト回収が見込めます」と先に反論を認めて答えることで、誠実さと準備の良さを示せます
- ・ 「わからないこと」を正直に言う：すべての反論に答える必要はありません。「その点については確認できていません。確認した上でご回答します」という誠実な回答が信頼を生みます

8-6 情報収集と整理—正しい情報を効率よく手に入れる

現代は情報過多の時代です。「正しい情報を見分け、効率よく整理する力」がビジネスパーソンとして不可欠になっています。

■ 情報の質を見極める—信頼できる情報源とは何か

- ・ 一次情報と二次情報：一次情報は「原典（論文・公式発表・データ）」、二次情報は「まとめ・解説・記事」、重要な判断は一次情報で確認します
- ・ 誰が言っているか：情報の発信者の「専門性・利害関係・実績」を確認します。「〇〇の専門家が言っている」と「匿名のブログが言っている」では信頼性が異なります
- ・ いつの情報か：特に技術の世界では、3年前の情報が現在では無効になっていることがよくあります。情報の日付を必ず確認します

- ・ エンジニアの実践：公式ドキュメント・GitHubのソースコード・学术论文が信頼性の高い一次情報源です

■ クリティカルシンキング—「疑う力」で思考の質を上げる

ロジカルシンキングが「論理的に考える力」なら、クリティカルシンキングは「論理の正しさを疑う力」です。両方を持つことで、思考の精度が上がります。

- ・ 前提を疑う：「当たり前」とされていることの背後にある前提が正しいかを確認します。「なぜそうなのか」を繰り返すことで、見落としていた前提が見えてきます
- ・ 自分のバイアスに気づく：「自分が信じたいことを信じやすい（確証バイアス）」「最初に見た情報に引きずられる（アンカリングバイアス）」、これらの認知の歪みを知ることで、より公平な判断ができます
- ・ 「相関」と「因果」の違いを理解する：「AとBが同時に起きている（相関）」は「AがBを引き起こした（因果）」を意味しません。エンジニアとして、観測されたデータから安易に因果を導かない厳密さが求められます

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・ 身近な「困りごと」を1つ、「なぜ？」を3回ほど掘り下げてみる。

情報があふれる時代には、「考える力」と同じくらい「情報を見極める力」が大切です。誰が、いつ言ったのか。一次情報か、また聞きか。とくにAIが何でも答えてくれる時代だからこそ、「その答えは本当に正しいか」と一歩立ち止まれる人が強い。鵜呑みにせず、確かめる、この姿勢が、あなたの判断を守ります。

8-7 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ 「もしなくダブリなく」分けるとは、どういうことか。
- ・ 根本原因に近づくには、何を繰り返すか。
- ・ 事実・意見・推測の違いを、一言で。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、これからの学びや仕事の中で、少しずつ埋めていきましょう。

第9章 数字・財務の基本（18～19日目・2日）

🎯 この章のゴール

この章では、数字・財務の基本として、会社とお金の数字をこわがらずに読む力を学びます。
終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ①【違いがわかる】会社の3つの数字（PL/BS/CF）の違いがざっくりわかる。
- ②【考え方がわかる】感覚でなく数字で語る大切さがわかる。
- ③【イメージできる】額面と手取りの差がなぜ出るかわかる。

💡 まず自己診断—読む前に、いまの自分を知ろう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「数字・財務を自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
 - ② 「数字や財務は、誰の仕事だと思いますか？」を、いまの考えで一言だけ書きます（正解は気にしません）。
- 読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

✦ よくある勘違いと、その反証

【勘違い】「数字・財務は経理や経営層の仕事」→

【反証】あなたの工数・残業・使ったクラウド費用は、すべて会社の数字に直結します。

自分の仕事を「コストと価値」で語れる人は、提案が通りやすく、早く信頼されます（第8章8-5・第21章21-6にもつながります）。

9-1 なぜ数字を読めることが大切か

「財務・数字は経営者や経理が考えること」ではありません。**すべてのビジネスパーソンが数字を理解することで、仕事の意味が深まり、提案の説得力が増します。**

数字が読めると、ニュースや会社の発表も“自分ごと”として読めます。「増収増益」「赤字」「黒字化」、こうした言葉の意味がわかると、自分の会社が今どんな状況にあり、なぜその判断がされたのかが見えてきます。世の中の解像度が、ぐっと上がります。

- ・ 自分の仕事の意味が変わります。「この機能を追加するコストは〇万円、売上貢献は〇万円」という視点で考えられます
- ・ 提案の説得力が上がります。「〇〇をすれば、年間〇〇万円のコスト削減になります」の方が動きます
- ・ 組織の優先順位を理解できます。なぜこの機能より先にあの機能を作るのかが、財務の視点から理解できます

「数字や財務は苦手」と感じる人は多いものです。でも、ここで必要なのは、決算書を作る力ではなく、「数字をこわがらずに読む」力です。数字は、社内の人みんなが使う“共通言語”。読めるよう

になると、自分の仕事が会社にどう役立っているかが見え、提案も通りやすくなります。

◆ 考えてみよう：あなたの仕事は、いくら価値？

あなたが将来やる仕事は、会社にどんな形で「お金」として返っていくのでしょうか。コストを下げる、売上を生む、時間を節約する、ざっくりで構いません。

自分の仕事を“数字”で想像してみると、見え方が変わります。

「数字は苦手」と感じる人ほど、まずは“身近な数字”から始めるのがおすすめです。自分の給与明細、よく使うアプリの利用者数、ニュースの売上高、日常の数字に「これは何を意味するんだろう」と一言問いかけるだけで、数字との距離は縮まります。

9-2 財務三表の基本

■ 損益計算書（PL: Profit & Loss）—会社がいくら稼いでいくら使ったか

項目	意味	エンジニアとの関係
売上高	顧客から受け取るお金	自分が開発した製品・サービスの価格×販売数
売上原価	製品・サービスを提供するための直接コスト	開発コスト・インフラコスト
売上総利益（粗利）	売上高－売上原価	開発コストを回収できているか
営業利益	粗利－販管費（人件費・広告費等）	会社の本業での稼ぐ力
純利益	税引き後の最終利益	会社に残るお金

■ 貸借対照表（BS: Balance Sheet）—会社の財産と借金の一覧

区分	内容	例
資産（左側）	会社が持っているもの	現金・売掛金・設備・ソフトウェア
負債（右側上）	返さなければならないお金	借入金・買掛金・未払費用
純資産（右側下）	株主から集めたお金＋これまでの利益の蓄積	資本金＋利益剰余金

■ キャッシュフロー計算書（CF）—実際のお金の動き

PLで「利益が出ている」のに倒産することがあります、これが「黒字倒産」です。PLの利益と実際の現金の動きは一致しないからです。CFを理解することで、会社の健全性をより深く把握できます。

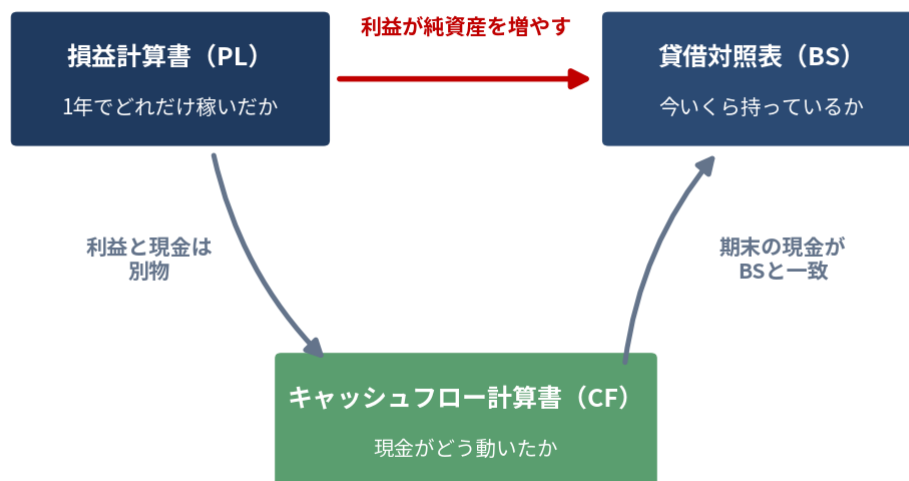
- ・ 売上は「商品を買った時点」で計上されますが、お金が入るのは後になります
- ・ CFは「実際に現金がいつ・いくら・どこから来て・どこへ出ていったか」を示します

📦 事例から学ぶ：黒字倒産の事例から学べること

売上好調のまま資金ショートで倒産した中小企業の事例があります。この教訓から学べること：PL（利益）とCF（現金）は別物であることを理解することで、会社の財務健全性を正しく判断できるようになります。

財務三表（PL・BS・CF）は、会社の“健康診断書”のようなものです。PLは「どれだけ稼いで使ったか（成績表）」、BSは「今どれだけ財産と借金があるか（体格）」、CFは「お金が実際にどう動いたか（血流）」。3つをセットで見ると、会社の元気が立体的に分かります。全部を暗記する必要はなく、「何を表すか」が分かれば十分です。

財務三表の関係



三表はつながっている — 同じ会社を3つの角度から見た姿

図 財務三表の関係

三表はバラバラではなく、つながっています。PLの「利益」はBSの「純資産」を増やし、その現金の動きはCFに表れる、3つは同じ会社を別の角度から写した写真のようなものです。

そして、数字には「なぜ？」と「何と比べて？」をセットにすると、はじめて意味が見えてきます。売上が増えても、コストがそれ以上に増えていれば利益は減ります。数字を単体で見えて一喜一憂せず、前月や目標と比べ、その差の理由を考える、その癖が、数字に振り回されない力になります。

9-3 数字の使い方—ビジネスでの実践

■ KPI（重要業績指標）の概念

- ・ KGI（Key Goal Indicator）：最終目標、「年間売上1億円を達成する」
- ・ KPI：KGIを達成するための中間指標、「月間新規ユーザー500人獲得」「解約率2%以下」
- ・ エンジニアのKPIの例：レスポンスタイム・エラー率・デプロイ頻度・インシデント復旧時間

■ コストの種類

コストの種類	定義	エンジニアの例
固定費	売上に関わらず発生するコスト	サーバ費・人件費・オフィス賃料
変動費	売上に比例して発生するコスト	API利用料・決済手数料
直接費	特定の製品・案件に直接紐づくコスト	特定プロジェクトのエンジニア工数
間接費	複数の製品・案件に共通して発生するコスト	共通インフラ・管理部門コスト

▶ やってみようワーク・演習

やってみよう：簡単なPLを読んで「この会社の強みと改善点はどこか」を考えましょう 架空の会社のPLを提示します。売上・利益率・各費用の推移を見て分析し、3分で発表します。

KPIと聞くと難しそうですが、要は「ゴールにたどり着くための“道しるべ”の数字」です。最終目標（KGI）だけ見ていても、日々何をすればいいかわかりません。そこで、途中の指標（KPI）を決めて、「今週はここまで進んだか」を確かめる。自分の仕事にも、「何を測れば前進が分かるか」という視点を持つと、動きが変わります。

数字でビジネスを語るうえで便利なのが、費用対効果（ROI）の考え方です。「この施策に100万円かけて200万円の効果が出るなら、やる価値がある」、この発想です。エンジニアの仕事も、「この改善にかかる時間（コスト）で、どれだけの価値（時間短縮・売上・品質）が生まれるか」で語れると、提案が通りやすくなります。完璧な計算でなくても、“ざっくりの見積もり”を持つだけで、説得力が変わります。

9-4 身近な数字—給与・税金・社会保険の実務

入社したら最初に手にするのが給与明細です。**自分の給与の仕組みを理解することは、権利と将来設計の基本になります。**

■ 給与明細を読む

控除項目	意味	およその割合
社会保険料（健康保険・厚生年金・雇用保険）	加入が義務付けられている保険	額面の約15～16%（本人負担分。年度・地域で変動）
所得税	国に納める税金（毎月仮計算）	額面・扶養等により異なります
住民税	都道府県・市区町村に納める税金（前年所得基準）	翌年6月から天引き開始

■ 社会保険の仕組みと意味

保険の種類	何のためにあるか	知っておくべきポイント
健康保険	病気・ケガの医療費補助（原則3割負担）	傷病手当金・出産手当金も活用できます
厚生年金	老後・障害・遺族の生活保障	会社が半額負担。老後だけでなく障害・死亡時の保障もあります
雇用保険	失業・育休・スキルアップ支援	育休給付金も雇用保険から出ます
労災保険	業務中・通勤中の事故の補償	全額会社負担。業務中のケガは必ず利用できます

- ・ 所得税・住民税の基本、年末調整で適切に申告することで、払いすぎた税金が還付されることがあります

知っておこう—法律の紐づけ

最低賃金法・労働基準法（賃金の基本）、自分の給与と権利を理解することが、適切な労働環境を守ることにつながります。

▶ やってみよう—ワーク・演習

やってみよう：自分の給与明細（モデルケース）を読み解きましょう 架空の給与明細を提示します。「額面から手取りまでの計算過程」を確認し、「各控除の意味と金額」を理解します。

給与明細は、社会人にとって最初の“身近な数字”です。額面と手取りの差、引かれている税金や社会保険料、最初は分かりにくくても、一度ちゃんと読むと、「自分のお金がどう動いているか」が見えてきます。会社の数字を学ぶ第一歩は、まず自分の給与明細から、です。

給与明細には、額面・控除・手取りのほかに、勤怠（残業時間など）や、会社が半分負担している社会保険料（労使折半）も表れます。自分が受け取る金額の“裏側”で、会社もコストを負担している、その全体像を知っておくと、働くことの仕組みが、より立体的に見えてきます。

9-5 数字で考える・数字で話す—ビジネスパーソンとしての基礎

エンジニアが「もっと評価されたい・もっと大きな仕事に関わりたい」と思うなら、技術の話を「ビジネスの数字」で語る力が不可欠です。

■ 「感覚」を「数字」に変換する習慣

- ・ 「遅い」→「平均レスポンスタイム2.3秒（目標値1.0秒に対し2.3倍）」
- ・ 「バグが多い」→「月間インシデント件数18件（前月から約4割増）、平均復旧時間4.2時間」
- ・ 「改善した」→「デプロイ頻度を週1回から毎日に改善。バグ検出時間が平均72時間から4時間に短縮」

- ・ 「コストがかかる」→「現状のインフラコスト月50万円。最適化により月30万円（40%削減）が見込める」

■ グラフの読み方・作り方—数字を「見える化」する

- ・ 折れ線グラフ：時間変化を見せるときに使います。「月次MRRの推移」「レスポンスタイムの改善履歴」
- ・ 棒グラフ：比較・ランキングに使います。「機能別エラー件数の比較」「チーム別デプロイ頻度」
- ・ 円グラフ：構成比を見せるときに使います。「コストの内訳」「インシデントの原因分類」。ただし5種類以上の比較には不向きです
- ・ グラフのよくある罫：Y軸を途中から始めて変化を大きく見せるのは印象操作です。数字の誠実な可視化を心がけましょう

▶ やってみよう—ワーク・演習

やってみよう追加：自分の仕事をROIで語る練習

自分が取り組んでいる（または取り組む予定の）技術的な改善を1つ選び、以下の形でROIを試算しましょう。

- ①改善内容：何を改善するか（例：ログイン処理の最適化）
- ②現状コスト：今の問題によって発生しているコストは何か（例：ユーザーの離脱率・インフラコスト）
- ③投資コスト：この改善に何人日・何円かかるか
- ④期待効果：改善後にどんな変化が期待できるか（数字で）
- ⑤ROI計算： $(\text{期待効果の金額換算} - \text{投資コスト}) \div \text{投資コスト} \times 100\%$

「数字で語れない改善提案」は、優先度をつける議論に参加できません。 小さな計算の習慣が、エンジニアとしての影響力を高めます。

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・ 初任給の明細が出たら（または本文のモデル例で）、額面と手取りの差がどこで出るかを見める。

同じ報告でも、「なんとなく速くなりました」より「処理時間が3秒から0.5秒に短縮しました」のほうが、何倍も伝わります。数字は、あなたの仕事を正しく評価してもらうための“翻訳ツール”です。感覚を数字に置き換えるクセをつけると、提案も評価も、ぐっと通りやすくなります。

◆ 考えてみよう：あなたの1時間は、いくら？

会社はあなたに、給与・設備・教育などのコストをかけています。ざっくりでよいので、「自分の1時間にいくらかかっているか」を想像してみましょう。

その視点を持つと、会議の時間や手戻りの“見えないコスト”にも、自然と意識が向くようになります。

9-6 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ 会社の3つの数字（PL／BS／CF）の違いを、ざっくり。
- ・ なぜ「数字で語る」ことが大切か。
- ・ 額面と手取りの差は、なぜ出るか。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、これからの学びや仕事の中で、少しずつ埋めていきましょう。

第3部 総合ワークショップ (20日目・1日)

第3部の学びを統合する演習です。「論理的思考」と「財務の数字」を組み合わせ問題进行を解きます。

第3部（第8～9章）で学んだ「ロジカルシンキング・問題解決・数字・財務」を統合するワークショップです。「考える力」と「数字の力」を組み合わせ、実際のビジネス課題を分析・解決する力を養います。

■ 演習①：ロジカルシンキング実践—問題の「本当の原因」を探る（午前：2時間）

以下のシナリオに対して、「なぜ5回」と「ロジックツリー」を使って根本原因を特定します。

- ・ シナリオ：あなたの会社のWebサービスの新規登録ユーザー数が先月比30%減少しました。社長から「原因を調べて対策を提案してほしい」と依頼されました
- ・ ステップ①：「なぜ30%減少したか」を5回繰り返して根本原因を探る（グループワーク15分）
- ・ ステップ②：「ロジックツリー（Whyツリー）」で原因の全体構造を可視化する（グループワーク20分）
- ・ ステップ③：特定した根本原因に対して「解決策ツリー（Howツリー）」で対策を列挙する（グループワーク20分）
- ・ ステップ④：対策の「実現可能性×効果」でマトリクスを作り、優先順位を決める（グループワーク15分）

■ 演習②：数字で考える・数字で話す（午後：3時間）

提示されたデータを読み解き、「経営者に30秒で伝える」プレゼンを作ります。

- ・ 与えられるデータ：月次売上・ユーザー数・解約率・インフラコストの推移グラフ（過去12ヶ月）
- ・ 課題①：グラフから「最も重要な問題」を1つ特定し、その根拠を数字で説明する
- ・ 課題②：問題に対して「数字ベースの改善提案」を1つ作る（ROIを含む）
- ・ 発表：各グループが「この会社の最大の課題と提案」を2分間でプレゼンします

▶ やってみようーワーク・演習

▶ 第3部総合ワークショップ 振り返り

- ①「なぜ5回」を実践して気づいたこと、自分が最初に思った「原因」は根本原因だったか
- ②数字でビジネスを考えることの「難しさ」と「面白さ」をそれぞれ書く
- ③第3部全体で最も印象に残った「思考の枠組み」を1つ選び、明日から使う場面を考える

▶ やってみようーワーク・演習

やってみよう①：ケーススタディ 問題：「あるSaaSプロダクトのユーザー解約率が増加しています。

数字を使って原因を分析し、エンジニアリングの観点から対策を提案せよ」 やってみよう②：グループ発表 各グループが分析結果と提案を5分でプレゼンします。

「数字の根拠があるか」「論理的に整合しているか」「エンジニアリングの視点が含まれているか」

の3点で相互評価します。

中間振り返り（20日目後半・0.5日）

研修の振り返り点です。「知識として学んだこと」が「自分の行動として出ているか」を正直に確認する時間です。

- ・ 第1～3部（第1～9章）の学びを振り返ります
- ・ 序章で書いた「大切にしたいこと」を読み返します。2週間前の自分と今の自分は何が変わったか
- ・ 「知っていた」と「できていた」の差を確認します。「これは知っていたが、実際の場面でできなかった」ことを1つ特定します
- ・ 後半の研修（第4～6部）に向けた個人目標を設定します

▶ やってみようーワーク・演習

やってみよう：「前半で最も気づいたこと」と「後半で最も習得したいこと」を書き出しましょう

前半の気づき：今まで無意識だったが、研修を通して意識するようになったこと

後半の目標：倫理観・法律・エンジニアリング・人生設計のどの領域で最も成長したいか

グループで共有し、研修終了後に見返すためのメモとして保存します

第4部 正しさを知り、身につける（21～30日目・10日間）

第4部では、法律・倫理・ハラスメント・情報管理・コンプライアンスという「正しく働くための知識」を体系的に学びます。これは道德の授業ではありません。ビジネスパーソンとして自分と組織を守り、信頼を失わないための実務知識です。**技術力がいくら高くても、この領域を軽視する人は、いずれ必ず問題を起こします。**優秀なビジネスパーソンほど、ここを最優先で学び続けます。

ここで扱う知識は、ふだんは表に出ません。けれど、ひとたび判断を誤れば、自分も、まわりも、会社も、深く傷つけてしまう領域です。だからこそ、「知らなかった」では済みません。怖がって萎縮するためではなく、**正しく怖がり、自分とまわりを守るために学びます。**

深掘り章（第10～14章）はケーススタディ・ロールプレイ中心で体験的に学びます。

概観章（第15～16章）は知識インプット+代表事例で理解します。

中間振り返り—20日間の学びを統合する

研修の折り返し地点に立っています。前半20日間で「知識として頭に入ったこと」と「実際に行動が変わったこと」の差を正直に確認します。この振り返りが、後半20日間の学びを深める出発点になります。

■ 個人振り返り（30分）—自分と正直に向き合う

以下の問いに、「正解を書こう」とせずに、自分の本音で向き合ってください。

- ・ ①序章で書いた「大切にしたいこと」を読み返す、今もそれが大切だと思うか。何か変化があったか
- ・ ②ここまで（序章～第9章）の中で「最も自分の行動を変えた内容」を1つ選ぶ、どんな状況で・どう変えた・その結果どうだったか
- ・ ③「知識としてはわかったが、実践できていないこと」を正直に2つ書く、なぜ実践できていないのか
- ・ ④前半20日間で「一番難しかった場面」、それはどんな場面で、どう対処したか（または対処できなかったか）
- ・ ⑤後半20日間で「特に深く学びたいこと・力をつけたいこと」を1～2章分書く

■ グループ共有（30分）—仲間の学びから学ぶ

4～5人のグループで「学びの共有」を行います。正解を競うのではなく、それぞれの気づきを共有することで学びを深め合います。

- ・ 一人2～3分：「前半で最も印象に残ったこと」と「正直にできていないと感じること」を話します
- ・ グループで議論：「入社後の実際の仕事の中で、研修の学びをどう活かそうか」を自由に意見交換します
- ・ 気づきのシェア：「他のメンバーの話を聞いて、新たに気づいたこと」を一人ひとりが発言します

■ 後半20日間へのコミットメント（15分）—「次の一步」を決める

後半（第4部～第6部）は「法律・倫理・情報・キャリア・人生」という、より長期的・社会的な視点の学びが続きます。

- ・ 第4部（21～30日目）：法律・倫理・ハラスメント・情報管理、「何が正しいか」を判断する軸を鍛えます。自分が加害者にも被害者にもならない知識と判断力を育てます
- ・ 第5部（31～34日目）：成長・キャリア、成長し続ける姿勢を持ち、専門家としての方向性を明確にします。「エンジニアとして10年後どうあるか」を考え始めます
- ・ 第6部（35～37日目）：ライフプラン・仕事と人生の統合、40年の社会人人生を豊かに設計する視野を持ちます。今から少しずつ準備を始める習慣が、将来の選択肢を大きく広げます
- ・ 第7部（38～43日目）：「特論：新卒社員向け ITエンジニア入門」、ソフトウェアエンジニアの土台（あるべき姿・倫理・体系・品質・セキュリティ・伝える力・法律・生成AI・成長）を体系的に学び、専門家としての土台を固めます

事例から学ぶ：「中間地点での正直な振り返り」が後半の成長を加速させた事例

ある企業の新卒研修で、前半終了時に「実践できていないことを正直に書く」振り返りをしたグループと、「学んだことをまとめる」振り返りをしたグループの後半の学習効果を比較した結果、前者の方が最終発表での行動変化の具体性が高かったという報告があります。

この教訓から学べること：「できていないことを正直に認めること」は失敗ではありません。それが次の学びへの最も正直なスタート地点です。

中間地点での正直な振り返りが、後半20日間を本物の学びにします。

▶ やってみようワーク・演習

中間振り返り 個人コミットメントシート

以下を記入して、後半20日間のスタートを切りましょう。

【前半で最も大切だと思ったこと（1文）】

【正直に「できていない」と感じること（1～2点）】

【後半で特に深く学びたい章・テーマ】

【後半の研修中に変える「具体的な行動」（1つだけ、明日からできること）】

例：「質問するとき必ず自分の考えを先に述べてから聞く」

例：「1日の終わりに3行の振り返りメモを書く」

【1年後の自分へ：今感じていることを1段落で書く】

（あなたの言葉で。正解はありません）

第10章 「正しさ」の4層を解剖する（21日目・1日）

この章のゴール

この章では、「正しさ」の4層（人として・社会人として・専門家として・社員として）を学びます。

終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ①【順番がわかる】「正しさの4層（人→社会人→専門家→社員）」の優先順がわかる。
- ②【考え方がわかる】迷ったら「セーフ/アウト/要確認」で考える、とわかる。
- ③【考え方がわかる】「技術的にできる≠やってよい」がわかる。

まず自己診断—読む前に、いまの自分を知らう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「正しさの判断を自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
 - ② 「会社のルールと自分の良心が食い違ったら、どちらを優先すべきだと思いますか？」を、いまの考えで一言だけ書きます（正解は気にしません）。
- 読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

10-1 人としての正しさ—普遍的な誠実さ・品格

「正しさ」の最も根本にあるのは、法律でも会社のルールでもなく、人として誰に対しても持つべき誠実さ・品格・良心です。

これは時代・文化・立場を超えて普遍的に問われるものです。

- ・ 誠実さ：思っていることと言っていることが一致している。嘘をつかない
- ・ 品格：自分の利益のためだけに動かない。他者の尊厳を傷つけない
- ・ 良心：「誰も見ていなくても」自分が恥じない行動を選べる
- ・ 責任感：自分の行動の結果を引き受ける。他者のせいにはしない

この4つ、誠実さ・品格・良心・責任感は、どれも特別な才能ではなく、日々の小さな選択の積み重ねで育ちます。約束の時間を守る、できないことはできないと言う、人の失敗を笑わない。そうした一つひとつが、あなたの“人としての土台”を少しずつ固めていきます。

ポイント：正しさは「減点」より「土台」

正しさを“減点されないためのルール”と考えると、窮屈になります。そうではなく、「信頼される自分をつくる土台」と考えてみてください。

正直で、約束を守り、誠実な人のところに、良い仕事と良い仲間が集まります。正しさは、あなたを守り、伸ばす力です。

社会人になると「業務上の指示だから」「会社の方針だから」という言い訳が生まれがちです。しかし、どの組織に属していても、個人の品格は問われ続けます。

「正しさの第1層」は、法律に書いていなくても、上司に言われなくても、あなた自身の内側から湧き出る誠実さです。技術や知識は、この土台の上であって初めて価値を持ち、土台が崩れれば、優れた技術ほど大きな害になります。

◆ よくある勘違いと、その反証

【勘違い1】「正しさ＝ルールを破らないこと」→

【反証】ルール遵守は4層のうちの2つ（社会人・社員）にすぎません。ルールに書いていなくても問われる“人としての誠実さ”が最上位です。

【勘違い2】「会社の指示なら従うのが正しい」→

【反証】第4層（社員）は第1層（人）・第2層（法令）を超えられません。

「上司に言われた」は、違法・非倫理の場面では免責になりません（第14章・情報の実践でも繰り返し出てくる原則です）。

「人としての正しさ」と聞くと、立派な道德の話に感じるかもしれません。でも中身はシンプルで、嘘をつかない、約束を守る、人をだまさない、困っている人を見て見ぬふりをしない、子どものころに教わったことと、根は同じです。社会人になっても、これが土台であることは変わりません。

■ なぜ「人としての正しさ」が土台なのか

法律も、会社のルールも、もとをたどれば人が作ったものです。だから、ときに不完全だったり、現実に合わなくなったりします。そんなとき、最後に頼れるのは「人として、これは正しいか」という良心です。字面だけを守って人を傷つけるより、ルールの背後にある「何のためか」を考えられる人、それが、第1層が土台であるという意味です。

新卒のうちは、「言われたことを正しくやる」だけで精一杯かもしれません。それで十分です。ただ、心の片隅に「これは人として大丈夫か」という問いを置いておく。その小さな習慣が、いざ判断を迫られたときに、あなたを助けます。

◆ 考えてみよう：誰も見ていないとき

もし、ズルをしても絶対に誰にもバレないとしたら、あなたはそれをするでしょうか。

「見られていないときの自分」が、その人の本当の品格です。正しさは、誰かに見せるためではなく、自分が自分を信じられるためにあります。

📖 事例から学ぶ：小さな正直が信頼を生む

ある新人は、自分のミスで生じた小さな不具合を、隠さずすぐ報告しました。その場では少し気まずかったものの、先輩からは「正直に言ってくれて助かった」と信頼されるように。逆に、ミスを隠す人は、たとえ優秀でも、だんだん信頼されなくなります。

→ あなたにできる一歩：迷ったら、正直なほうを選ぶ。正しさは、地味でも確実に積み上がる財産です。

10-2 社会人としての正しさ—法令・契約・守秘義務

第2層は、社会人として守らなければならない外部的な規範です。

「知らなかった」では済まない、社会が共通のルールとして定めたものです。

- ・ 法令遵守：労働法・個人情報保護法・著作権法など、社会人が守るべき法律

たとえば、ネットで見つけた画像を資料に無断で使う、音楽を勝手に使う、これらは著作権法に触れます。「みんなやっているから」「これくらい」は通用しません。社会人には、知らずに法を破らないよう、最低限のルールを“知っておく”責任があります（詳しくは第11章）。

- ・ 契約・約束：口頭でも書面でも、約束したことは守る。守れないなら事前に連絡する
- ・ 守秘義務：業務で知り得た情報（顧客情報・社内情報）を外部に漏らさない
- ・ 利益相反の回避：自分や特定の人物の利益のために会社・顧客を不利に扱わない

利益相反は、難しい話に聞こえますが、身近にもあります。発注先を選ぶときに「友人の会社だから」と選ぶ、自分の副業に有利になるよう会社の判断をゆがめる、こうした“自分の都合”と“会社・顧客の利益”がぶつかる場面です。迷ったら、「これを第三者が見たら、公正だと思うか」を基準にしてください。

これらは「最低限の正しさ」です。守って当然であり、破ると法的・社会的責任が生じます。

しかし、これを守るだけでは「良い社会人」とは言えません。

第1層（人としての誠実さ）があっても、第2層のルールが生きてきます。

守秘義務は、特別な機密だけの話ではありません。顧客の名前、案件の内容、システムの構成、これらを社外で話さない、SNSに書かない、というのも守秘義務です。「契約」も、書面のものだけではありません。口頭の約束も約束です。守れそうにないときは、黙って遅れるのではなく、早めに「ここまでしかできない」と伝える。それが、社会人としての正しさの基本です。

10-3 専門家としての正しさ—専門知識を持つ者の責任

第3層は、専門知識・技術を持つ者として果たすべき正しさです。エンジニアは、自分が作るものが社会やユーザーに与える影響に責任を持ちます。「技術的にできる」と「やってよい」は違います。品質・安全・セキュリティを軽視した実装、検証を省いたリリース、既知の脆弱性の放置は、たとえ上司の指示であっても、専門家としての正しさに反します。

この第3層は、第1層（人としての誠実さ）・第2層（法令）の上に立つことはできませんが、組織内のルール（第4層・社員）よりは優先されます。会社の都合よりも、社会とユーザーの安全が上からです。専門家としての正しさは、第21章（特論：新卒社員向け ITエンジニア入門）でさらに深めます。

私たちエンジニアが作るものは、人の生活、お金、ときに安全に関わります。決済システムが止まれば買い物ができず、医療や交通のシステムが誤れば命に関わることもあります。だからこそ、「動けばいい」ではなく「安全か」「誰かを傷つけないか」まで考えるのが、専門家の正しさです。

事例から学ぶ：「動くからOK」で済ませなかった

あるエンジニアは、急ぎの機能が一応動くものの、入力チェックが甘く、悪用される余地に気づきました。納期は迫っていましたが、リスクを上長に正直に伝え、最小限の対策を入れてからリリース。後日、それが大きな事故を防いだと分かりました。

→ あなたにできる一歩：「動く」だけで安心せず、「悪用されないか」「誰かが困らないか」を一度問う。

■ 「技術的にできる」と「やってよい」は違う

専門知識を持つ人は、知らない人にはできないことが「できて」しまいます。システムに入って他人のデータを見る、ログを書き換える、便利だが危ういコードを仕込む、技術的には可能でも、やってよいとは限りません。力を持つほど、「できる」と「やってよい」を区別する責任が重くなりま

す。

もう一つ、専門家には「説明する責任（説明責任）」があります。なぜその設計にしたのか、なぜその判断をしたのかを、専門でない人にも分かる言葉で説明できること。「自分にしか分からない」状態は、専門性ではなく、リスクです。

専門家の正しさには、「学び続ける義務」も含まれます。古い知識のまま判断すると、知らないうちに人に害を与えることがあります。最新の安全やルールを学び続けることも、専門家の誠実さの一部です。

事例から学ぶ：見られるけれど、見ない

ある若手は、運用作業の権限で、顧客の個人データをいくらでも閲覧できる立場にありました。好奇心がわく場面もありましたが、「業務に必要な範囲だけ」を徹底し、不要なデータには触れませんでした。技術的にできることと、やってよいことの線を、自分で引いていたのです。

→ あなたにできる一歩：アクセスできる＝見てよい、ではない。「これは今の仕事に必要なか」を一度問う。

10-4 社員としての正しさ—組織への貢献と正直な発言

第4層は、特定の組織に属する「社員」として求められる正しさです。

会社・チームへの貢献、組織内での誠実な行動がこれにあたります。

- ・ 組織への貢献：給与をもらっている以上、組織の目的に向けて誠実に働く義務がある
- ・ 正直な発言：上司に不都合な事実を隠さない。問題を早めに報告する勇気を持つ

「上司に不都合な事実を伝える」のは、勇気が要ります。でも、悪い知らせほど早く伝えるのが、結局はみんなを助けます。隠して大きくなるより、早く共有して一緒に対処するほうが、ずっと建設的です。正直な発言は、組織への最大の貢献の一つです。

- ・ 情報の適切な共有：「自分だけが知っている」という状態をなるべく作らない
- ・ 組織のリソースの適切な使用：会社の時間・設備・予算を私的に使わない
- ・ 改善への参加：問題を見つけたら指摘し、より良い組織作りに能動的に関わる

この「改善への参加」は、立派な正しさです。「おかしいな」と思ったことを、ただ我慢するのでも、陰で文句を言うのでもなく、「こうしたらどうでしょう」と前向きに提案する。小さな改善提案の積み重ねが、組織を少しずつ良くしていきます。

注意すべきは、「第4層（組織への忠誠）」が「第1層（人としての誠実さ）」や

「第2層（法令）」を超えることはないという点です。

上司の指示であっても、法律違反・倫理違反には従わない判断力を持ちましょう。

事例から学ぶ：忠誠と正しさがぶつかるとき

「会社のためだから」と、データを少しだけ良く見せる加工を頼まれたら。組織への貢献は大切ですが、それが事実をゆがめることなら、第1層・第2層が優先されます。本当の忠誠とは、会社が道を外しそうなときに「それはまずいのでは」と言えること。黙って従うことではありません。

→ あなたにできる一歩：「会社のため」と言われたときこそ、「それは人として・法的に大丈夫か」を一度立ち止まって考える。

「情報を抱え込まない」ことも、組織への正しさです。自分だけが知っている状態は、一見“頼ら

れている”ようで、実はチームのリスク。知っていることを共有し、誰かが休んでも回るようにしておくことが、健全な組織への貢献です。

10-5 4層が衝突したとき、どう判断するか

優先順位	層	なぜこの順番か
1位（最優先）	人としての正しさ	時代・法律・組織が変わっても変わらない普遍的な価値
2位	社会人としての正しさ	社会のルール、法律・契約を守ることは社会参加の前提
3位	専門家としての正しさ	技術者の責任、専門知識を持つ者の社会への責任
4位	社員としての正しさ	組織のルール、多くの場面で最も目につきますが、最も下位に位置します

「会社のルールに従うことが、人としての正しさに反する」、そのとき、4層の優先順位が判断の拠り所になります。この軸があれば、困難な判断の場面でも自分らしく行動できます。

◆ 考えてみよう：4層がぶつかる、身近な場面

あなたの想像で、「会社のルールと、人としての正しさがぶつかりそうな場面」を一つ思い浮かべてみましょう。そのとき、どの層を優先すべきでしょうか。

正解を出すことより、“迷ったときに立ち返る順番がある”と知っておくことが大切です。

▶ やってみようーワーク・演習

やってみよう：「あなたならどうするか」を議論します シナリオ①：上司から「このバグは今回は報告しなくていい」と言われた。

しかしそのバグはユーザーデータに影響する可能性がある。

どうするか。

シナリオ②：取引先との会議で、自社の製品の欠陥を隠すような説明をするよう指示された。

どうするか。

議論の観点：4層の正しさのどれが関係しているか / どんな行動が自分と組織と社会を守るか

⚖️ 迷ったときの3分類—「セーフ/アウト/要確認」

グレーに見える場面も、3つに仕分けると動けます。

【セーフ】4層のどれにも反しない→そのまま進める。

【アウト】人としての正しさ・法令に反する→指示でもやらない。

理由を添えて断り、記録を残す。

【要確認】判断がつかない→“やる前に”信頼できる人に相談する（やってしまったからでは戻せない）。

コツは「迷い＝赤信号ではなく黄信号」。黄信号は“止まって確認”の合図です。沈黙して進むのが一番危険です。

✦ 考えてみよう：あなたの“ゆずれない一線”

仕事で「これだけは、頼まれてもやらない」と思うことを、一つ思い浮かべてみましょう。
事前に自分の中で線を決めておくと、いざ迫られたときに迷いません。線は、平時にこそ引いておくものです。

判断に迷ったら、第21章で学ぶ手順、事実を確認 → 関係者を考える → 選択肢を出す → 原則に照らす → 後で堂々と説明できるか、も思い出してください。一人で抱えず、信頼できる人に相談するのも、立派な判断です。

10-6 倫理的ジレンマの解決フレームワーク—迷ったときの判断軸

「何が正しいか」が明確でない場面は、社会人生活の中で必ず訪れます。「4層の正しさ」に加えて、いくつかの倫理的推論のフレームワークを知っておくことで、ジレンマに直面したときの判断の質が高まります。

■ 倫理的推論の3つのアプローチ

倫理学では、正しい行動を判断するための考え方がいくつかあります。**どれか一つが「正解」ではなく、複数の視点から考えることで、より良い判断ができます。**

難しく考えすぎなくて大丈夫です。最初は「結果はどうか」「みんながやったらどうか」「尊敬する人ならどうするか」、この3つを順に自分に問うだけで十分。慣れてくると、自然と多面的に考えられるようになります。

- ・ ①結果主義（功利主義）：「この行動の結果、関係する人の幸福の総量が最大になるか」を問います。功利主義（ベンサム・ミル）の考え方です。「最大多数の最大幸福」が指針です
- ・ ②義務論（デオントロジー）：「この行動は義務・規則に従っているか」を問います。カントの「定言命法」：「この行動をすべての人が同じようにしたら、社会はどうか」が判断の基準です
- ・ ③徳倫理学：「この行動は、尊敬できる人（徳のある人）がとる行動か」を問います。「誠実な人ならどうするか」「信頼できる上司はどう判断するか」と問いかけます

■ 3つのアプローチで同じジレンマを考える—実例

例題：システムのバグを発見した。修正に1週間かかる。しかし明日がリリース予定日で、顧客に「明日リリース」と告知済みだ。バグの影響は軽微だが、可能性として個人情報に影響するかもしれない。どうすべきか。

- ・ 結果主義から考える：リリース延期の影響（顧客の信頼損失・スケジュール変更コスト）vs 情報漏洩リスクの影響（より多数・より深刻）→ 多数の顧客への深刻な影響を防ぐ方が最大幸福に合う
- ・ 義務論から考える：「バグを知らずにリリースする」という行動を全エンジニアが行ったとしたら社会はどうか？→ 社会のシステムへの信頼が崩壊する。したがって義務論的にも延期が正しい

- ・ 徳倫理学から考える：自分が尊敬する誠実なエンジニアならどうするか？→ 短期的な不便を顧客に与えても、誠実に問題を伝えて修正してからリリースするはずだ
- ・ 結論：3つのアプローチすべてが「リリースを延期し、顧客に誠実に説明する」を支持する、こうして複数の視点が一致するとき、判断に自信を持てます

大切なのは、どのアプローチが正解かではなく、複数の視点から考える習慣です。一つの見方だけだと、見落としが生まれます。結果（みんなにとってどうか）、義務（みんながやったらどうか）、人柄（尊敬する人ならどうか）、この3つの問いを順に当ててみると、判断が立体的になります。

10-7 内部通報制度—組織の不正と向き合う勇気

「上司の指示が明らかにおかしい」「会社が不正を行っていると感じた」、そのとき、あなたはどのようなべきか。内部通報制度はこのような状況に対処するための仕組みです。

■ 内部通報制度とは何か

- ・ 定義：従業員が会社の法令違反・不正行為・倫理違反を、社内外の通報窓口で匿名または実名で報告できる制度
- ・ 公益通報者保護法（2004年制定・2022年改正）：通報を理由とした解雇・降格・嫌がらせは違法。通報者を保護する国の法律です（制度の詳細・対象事業者は第15章15-5で扱います）
- ・ 通報できる内容：法令違反（労働基準法・個人情報保護法・不正競争防止法等違反）、財務不正、安全基準違反、ハラスメント、環境違反など

■ 「言いにくい」が組織の腐敗を招く

内部通報制度があっても、実際に使うには勇気が必要です。しかし歴史が示すのは、「言える人が一人いた」だけで組織が救われる事例が多いということです。

- ・ 告発を「裏切り」と捉えない：**組織への真の忠誠は、問題を隠すことではなく、組織が正しく機能するよう働きかけることです**
- ・ 段階的な対応：①直属の上司→②その上の上司→③社内のコンプライアンス担当→④外部の公的窓口（消費者庁・労働局等）という順で相談できます
- ・ 記録をとる：不正を発見したときは、日時・内容・関係者をメモに残します。通報の際に重要な証拠になります

ここで大切なのは、「いきなり大ごとにしなくていい」ということです。確信が持てないうちは、まず信頼できる人に「これって、どう思いますか？」と相談するだけでも立派な一歩。一人で判断を抱え込まず、段階を踏んで動けば、あなた自身も守られます。

事例から学ぶ：友だちの何気ない質問

仲のよい友だちから「ねえ、あなたの会社って今度どんな新しいサービスを出すの？」と聞かれた、よくある場面です。

人として（友だちに）は教えてあげたい。

でも社員としては、まだ公表していないことは話せません。

この「言いたいけど言えない」モヤモヤこそ、『正しさの4層』がぶつかる瞬間です。

どれか1つだけが正解なのではなく、層がぶつかったときに「いったん持ち帰って確かめる（要確認）」と動けることが、社会人の判断力です。

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・ 迷いそうな場면을1つ思い浮かべ、「セーフ/アウト/要確認」に分けてみる。

内部通報には勇気が要りますし、「自分が騒いでいいのか」と迷うのも自然です。だからこそ、一人で抱え込まないこと。まず信頼できる人に話し、事実を記録し、社内外に相談先があると知っておく。早く声を上げた人が、結果として組織と自分を守ります。

歴史を振り返ると、大きな事故や不正の多くは、「おかしい」と感じた人がいたのに、声を上げられなかったために広がっています。逆に、たった一人が勇気を出したことで、被害が最小限に食い止められた例もあります。あなたのその一言が、誰かを守るかもしれません。

ポイント：正しさは、技術と同じで“鍛えられる”

正しい判断は、生まれつきのセンスではありません。「4層で考える」「セーフ/アウト/要確認に分ける」「尊敬する人ならどうするか問う」、こうした“型”を、小さな場面で使うことで、誰でも少しずつ判断力が育ちます。

そして、これはAIの時代にこそ効きます。AIは「できること」を一気に増やしますが、「やってよいか」を決めるのは人間です。何が正しいかを考える力、その土台となる“正しさの体系”を持っておくことが、これからますます大切になります。

10-8 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ 「正しさの4層」を、優先順に。
- ・ 迷ったときの3分類（セーフ/アウト/要確認）とは。
- ・ 「できる」と「やってよい」は、なぜ違うのか。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、これからの学びや仕事の中で、少しずつ埋めていきましょう。

第11章 社会人が知っておくべき法律の基本（22～23日目・2日）

🎯 この章のゴール

この章では、社会人が知っておくべき法律の基本を、自分と仲間を守るために学びます。

終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ①【違いがわかる】民事・刑事・行政の責任の違いがざっくりわかる。
- ②【知っている】仕事に関わる主な法律があることを知っている。
- ③【構えがわかる】「迷ったらしない・相談する」という構えがわかる。

💡 まず自己診断—読む前に、いまの自分を知ろう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「法律と仕事を自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
- ② 「法律は、破らなければ問題ないと思いますか？」を、いまの考えで一言だけ書きます（正解は気にしません）。

読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

✦ よくある勘違いと、その反証

【勘違い】「法律＝専門家のもの。」

破らなければOK、グレーはセーフ →

【反証】大事なのは条文の暗記より“なぜその法律があるか（趣旨）”の理解です。

趣旨がわかれば、グレーな場面でも「これは趣旨に反するか」で判断できます。

「知らなかった」「グレーだから」は、責任を消しません（第10章の3分類とあわせて使います）。

11-1 法律を知ることの意義

法律を知るとは「縛られる」のではなく「自分を守る力を持つ」ことです。知識があれば、自信を持って仕事に臨めます。

- ・ 自分の権利、適切な労働条件・安全な職場環境を主張できます
- ・ 組織の信頼、コンプライアンスを守ることが、長期的な組織の信頼につながります
- ・ 社会への責任、プロとして社会のルールを理解し、守ることが専門家としての基盤になります

法律を知る最大のメリットは、「自信を持って動ける」ことです。「これは大丈夫」と分かれば迷わず進めるし、「これは危ない」と気づければ立ち止まれる。知らないで、不安なまま進むか、知らずに踏み外すかになりがちです。知識は、あなたの行動の“ブレーキとアクセル”の両方になります。

💡 ポイント：知らないことは、調べて・聞いていい

法律を全部覚えている社会人はいません。プロでも、迷えば調べ、専門家に聞きます。

新卒に求められるのは“全部知っていること”ではなく、“危なそうだと気づいて、確認できること”。

その姿勢こそが、いちばんの法的リスク対策です。

民事・刑事・行政のうち、新卒が日常でいちばん意識すべきは「民事」と「刑事」です。うっかりの情報漏洩や著作権侵害は、会社への賠償（民事）や、場合によっては処罰（刑事）につながります。「自分は新人だから関係ない」ではなく、「新人でも責任は生じる」と知っておくことが、最初の一步です。

三つの責任は、別々ではなく重なり合います。だからこそ、「これは何の責任か」を一つずつ覚えるより、「人に損害を与えない・隠さない・約束を守る」という基本を押さえるほうが、結果的に多くのリスクを防げます。

■ 民事責任・刑事責任・行政責任—三つの違いと重なり

責任の種類	内容	例
民事責任	相手（個人・会社）への損害賠償	情報漏洩で顧客に損害→損害賠償請求
刑事責任	国家による制裁（拘禁刑・罰金等）	個人情報を不正に持ち出し→刑事訴追
行政責任	行政機関による処分・制裁	個人情報保護法違反→行政指導・命令

民事・刑事・行政は別々ではなく重なり合う（例：情報漏えい）

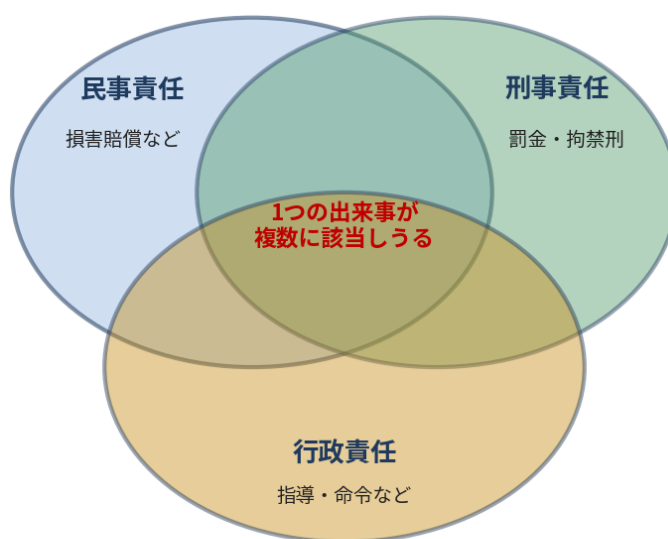


図 民事・刑事・行政の重なり（例：情報漏えい）

■ なぜ新卒が法律を知っておくのか

「法律は専門家やトラブルになった人のもの」と思うかもしれませんが。でも法律は本来、立場の弱い人を守るためにあります。労働法はあなたの働く条件を守り、著作権法はあなたの作ったものを守り、契約のルールはあなたの約束を支えます。知っていれば、不当な扱いに「それはおかしい」と言え、知らずに法を破ることも防げます。

「全部覚えるなんて無理」と感じるかもしれませんが。でも、暗記は要りません。「こういう法律が

ある」「迷ったら調べる・相談する」と知っておくだけで十分です。大切なのは、知識の量より、立ち止まれる感覚です。

✦ 考えてみよう：法律は誰の味方？

もし労働法や著作権法がなかったら、立場の弱い人や、何かを作った人は、どうなるでしょうか。法律は、強い者の横暴から弱い者を守るための“共通のルール”です。知ることは、自分を守る武器を持つことです。

11-2 使用者と労働者—労働法の大前提

労働法は「使用者と労働者の関係を規律する法律体系」の総称です。同じ法律でも、自分がどちらの立場にあるかによって読み方が変わります。

観点	使用者側（会社・経営者・管理職）	労働者（一般社員・新卒社員）
立場	義務を負う側	権利を持つ側
労働契約法	労働条件を書面で明示する義務	不利な条件を拒否できます。無期転換の権利があります
労働基準法	36協定なき残業は違法。有給取得拒否は違法	残業代は権利です。有給は取得できます
労働安全衛生法	部下の健康管理は法的義務	ストレスチェックを受ける権利があります。不調は早めに相談できます

「管理職になる」とは、部下の権利を守る責任を担う立場に変わることを意味します。昨日まで権利を持つ側だった人が、翌日には義務を負う側になります。

新卒のあなたは、今日から「権利を持つ労働者」です。残業代を受け取る、有給を取る、安全な環境で働く、これらは“お願い”ではなく“権利”です。遠慮しすぎる必要はありません。同時に、いつか管理職になれば、今度は部下のその権利を守る側に回ります。両方の視点で読んでおくと、将来も役立ちます。

労働法の中でも、新卒にまず関わるのが、労働時間や残業のルール（働きすぎを防ぐ）と、会社の安全配慮義務（心と体の安全を守る義務）です。残業には上限があり、会社はあなたが壊れるまで働かせてはいけません。詳しい権利（36協定・有給など）は第16章で扱いますが、まず「自分には守られる権利がある」と知っておいてください。

事例から学ぶ：ある会社で起きた過労自殺の事案から学べること

入社1年目の女性社員が過労・ハラスメントを原因に自殺した事件です。
この教訓から学べること：労働時間管理・ハラスメント防止は管理職の法的義務であり、組織全体で取り組むべき課題です。
この事件は長時間労働是正の社会的機運を高め、2018年成立の働き方改革関連法（残業の罰則付き上限規制）につながりました。

📖 事例から学ぶ：安全配慮義務と組織の責任から学べること

上司が部下のうつ症状を認識しながら見逃し、自殺に至ったケースで会社・上司に損害賠償が命じられた事例があります。

この教訓から学べること：安全配慮義務とは身体だけでなく精神的健康を守る義務も含みます（労働契約法5条）。

早めのサポートがチームメンバーの健康と組織の信頼を守ります。

11-3 会社の仕組みを定めた法律（会社法）

- ・ 株式会社の仕組み：株主が資金を出す→会社が運営する→利益を株主に還元します
- ・ 取締役の善管注意義務、取締役は会社のために誠実に職務を遂行する義務があります

新卒のうちは会社法を深く知る必要はありません。ただ、「会社は株主のお金で動き、その運営を任された取締役が責任を負っている」という大枠を知っておくと、なぜ承認や決裁の手続きがあるのか、なぜ勝手な判断が許されないのかが、腑に落ちます。

- ・ 従業員も「会社の行為者」になる場面、業務上の行為が会社の行為とみなされる場合があります

だからこそ、「自分の判断でいいのか」「これは上に確認すべきか」を見極める感覚が大切です。最初は分からなくて当然。迷ったら確認する、それを繰り返すうちに、線引きが身についていきます。

「従業員も会社の行為者になる」とは、たとえばあなたが業務で何かを発注したり、顧客に説明したりすると、それが“会社の行為”として扱われる、ということです。あなたの一言が、会社を法的に縛ることもあります。だからこそ、勝手に約束しない・あいまいなことは持ち帰る、が基本になります。

📖 事例から学ぶ：ある大手製紙会社の私的流用事件（2011年）から学べること

代表取締役が会社資金を私的に流用した特別背任罪の事例です。

この教訓から学べること：**会社のお金と個人のお金は明確に別物です。**

金額の大小にかかわらず、私的に使えば横領・背任として刑事責任を問われます。

この原則を理解することが、組織への信頼と自分自身の誠実さを守ります。

11-4 契約・取引を定めた法律**■ 民法（契約編）**

- ・ 契約は「申込み」と「承諾」の合致で成立します。口頭でも電子メールでも有効です
- ・ 業務上のやり取りは多くが「契約」、納期・品質・価格の約束は法的な契約行為です

約束（契約）を守れないとき、いちばんやってはいけないのは「黙って遅れる」ことです。守れないと分かった時点で、早めに正直に伝え、相手と対応を相談する。これは法律以前の信頼の問題でもあり、トラブルを小さく収める最善の方法です。

- ・ **「言った・言わない」を防ぐために書面・メール・チャットで記録を残す習慣が、自分と組織を守ります**

契約というと大げさですが、日々のメールの「では、その内容で進めます」も立派な契約です。受

託の仕事では、「どこまでやるか（範囲）」「いつまでに（納期）」「いくらで（金額）」の約束が中心。あいまいなまま進めると、後で「言った・言わない」のトラブルになります。だからこそ、口頭の合意も一言メモやメールで残す、これが自分を守る習慣です。

■ 製造物責任法（PL法）

- ・ 製品の欠陥によって損害が生じた場合、製造者は損害賠償責任を負います
- ・ 「設計上の欠陥」「製造上の欠陥」「警告上の欠陥」の三類型があります
- ・ 自分が関わった製品・システムが市場に出た後も品質への責任は続きます（賠償請求には期間制限があり、製品の引渡しから原則10年で権利が消滅します。このほか、損害と賠償義務者を知った時から3年などの期間制限もあります）

ソフトウェアやシステムも、考え方は同じです（※法律上はソフトウェア単体は無体物のため原則PL法の対象外で、機器に組み込まれた場合などが対象になりえます。法的責任の有無にかかわらず、品質への責任意識は同じです）。あなたが関わったものが世に出たあと、不具合で誰かが損害を受ければ、品質への責任が問われます。「リリースして終わり」ではなく、「出したあとも責任が続く」と意識しておきましょう。

事例から学ぶ：あるガス器具メーカーの湯沸器事故（2006年）から学べること

欠陥製品による死亡事故と企業の刑事・民事責任が問われた事例です。この教訓から学べること：エンジニアが品質に誠実であることが、社会への責任を果たし、組織と自分を守ることにつながります。

■ 取適法（旧・下請法）

- ・ 発注者の優越的地位を使った不当な扱いを禁止する法律です（代金の受領後60日以内の支払い義務、不当な減額・買いたたき・返品を禁止などを定めます）
 - ・ この法律を知ること、自社が取引先から不当な扱いを受けているかどうかを判断できます
- 受託の現場では、立場の弱い側が「ただで追加作業を」「一方的に値引きを」と迫られることがあります。取適法（旧・下請法）は、それを「おかしい」と言うための根拠になります。将来あなたが発注する側になったときは、同じことを取引先に強いない、その視点も、今から持っておきましょう（この法律は2026年1月施行の改正で「下請法」から「取適法（中小受託取引適正化法）」に名称が変わり、対象や規制も広がりました）。

11-5 情報・不正行為を定めた法律

■ 不正アクセス禁止法

- ・ **権限のないシステムへのアクセスは犯罪です**
- ・ 「同僚のIDを借りてアクセスした」、借りた側も貸した側も権限外のアクセスになりえます
- ・ 自分のアクセス権限を正しく理解することが、安全な業務遂行につながります

関連して「本番データの扱い」も要注意です。テストに本番の個人データを使う、必要もないのに顧客データを見る、これらは、権限や目的を外れると問題になります。「業務に必要な範囲だけ」を徹底するのが、自分を守る基本です。

■ ソフトウェアライセンスと著作権

- ・ OSSライセンスの種類と注意点：MIT・Apache・GPLなど、利用するOSSのライセンスを確認する習慣が組織を守ります

ライセンスを守るのは、堅苦しい決まりだからではありません。OSSは、世界中の人が「この条件なら自由に使っていい」と善意で公開してくれたもの。その条件を守ることは、作り手への敬意であり、コミュニティを支える行為です。いつかあなたが何かを公開する側になったとき、同じように尊重してもらえます。

- ・ 「無料で使える」と「なんでも使える」は異なります。ライセンスを理解することで安心して活用できます

著作権は、身近なところで関わります。ネットで見つけた画像・文章・コードを、確認せずに使うのは危険です。OSS（オープンソース）も「無料＝何でも自由」ではなく、ライセンスごとに条件（表示が必要、改変公開が必要など）があります。「使う前に、使ってよい条件かを確認する」、この一手間が、自分と会社を守ります。

◆ 考えてみよう：「ID貸して」と言われたら

忙しい先輩に「ちょっと君のIDでログインさせて」と頼まれたら、どうしますか。

権限のないアクセスは、貸した側も借りた側も責任を問われえます。親切のつもりが、二人とも危うくなる。やんわり断れる一言を、用意しておきましょう。

■ システム障害と法的責任

- ・ SLA（サービスレベルアグリーメント）、稼働率・応答時間・復旧時間の保証を契約書に明記します
- ・ 高い品質・可用性を設計から意識することが、顧客の信頼と組織の信頼を守ります

📖 事例から学ぶ：ある大手銀行のATM障害（2021年）から学べること

大規模なATM障害が社会・顧客・企業に与えた影響を示した事例です。

この教訓から学べること：金融インフラとしてのシステムの信頼性維持は社会的責任を伴います。設計段階から可用性・障害対応を考えることが重要です。

■ 刑法（業務関連）

- ・ 横領、会社の資産（お金・物・情報）を私的に使うこと。金額の大小を問いません
- ・ 背任、会社の利益より自己・第三者の利益を優先する行為です

横領や背任は、遠い世界の話に聞こえるかもしれませんが、身近な形もあります。経費を私的に使う、会社の備品を持ち帰る、業務で得た情報を自分のために使う、金額の大小にかかわらず、信頼を一瞬で失います。「会社のものと自分のものを、きっちり分ける」。シンプルですが、大切な線引きです。

- ・ 安全への配慮が法的義務であることを理解することで、安全な職場環境づくりに貢献できます

11-6 法律の全体マップー自分の仕事とどう関わるか

日常業務の場面	関連する法律	意識すべきこと
コードを書く	著作権法・不正アクセス禁止法	OSSライセンスの確認・権限の確認
情報を扱う	個人情報保護法・不正競争防止法	アクセス範囲・持ち出し禁止
契約・発注を行う	民法・取適法（旧・下請法）	書面での確認・合意内容の記録
製品をリリースする	PL法・著作権法	品質確認・OSSライセンス確認
退職・転職する	労働契約法・不正競争防止法	守秘義務の継続・情報の持ち出し禁止

- ・ 「これは法的にどうなのか？」と立ち止まる習慣を持つことが、自分と組織を守る力になります

◆ 考えてみよう：あなたの仕事に効く法律

あなたが将来やりそうな仕事を一つ思い浮かべ、「どんな法律が関わりそうか」を想像してみましょう。

コードを書くなら著作権、データを扱うなら個人情報、契約に関わるなら民法、仕事と法律は、思った以上に近くにあります。

- ・ 迷ったときの相談先：法務担当・顧問弁護士・外部の法律相談窓口

全部を覚える必要はありません。大事なのは、「これは法的にどうなんだろう？」と立ち止まれること、そして「迷ったら相談する場所がある」と知っていることです。自分で抱え込んで自己判断するのが、いちばん危険です。

▶ やってみようーワーク・演習

やってみよう①：「自分の業務に関わりそうな法律」を書き出す 入社後の具体的な業務を想定し、関連する法律を3つ以上特定します。

やってみよう②：「この行動は民事？
刑事？

それとも両方？」

複数のシナリオについて、グループで議論します。

11-7 「知らなかった」では済まないー法的リスクに気づく日常習慣

法律の違反は「意図的かどうか」にかかわらず責任が問われることがあります。日常業務で法的リスクに気づく習慣を持つことが、自分と組織を守ります。

グレーな場面で攻めて失敗すると、取り返しがつかないことがあります。攻めるべきは技術やアイデアであって、法令や倫理ではありません。ここは堂々と“守り”でいきましょう。

◆ 考えてみよう：「これ、大丈夫かな」と思った場面

最近の生活や仕事で、「これ、法的に大丈夫かな？」と一瞬でも思ったことはありませんか。その“ひっかかり”を無視せず、調べたり聞いたりする、その小さな習慣が、あなたを守ります。

■ 法的リスクが発生しやすい日常の場面チェックリスト

- ・ 新しいOSSを使うとき：ライセンスを確認しましたか（MIT/Apache/GPL等）
- ・ 外部APIを使うとき：利用規約を確認しましたか。個人情報を外部サービスに送信していませんか
- ・ 本番DBにアクセスするとき：業務上必要な範囲ですか。テストに本番データを使っていませんか
- ・ ユーザーの個人情報を処理するとき：同意を得た目的の範囲内の処理ですか。暗号化されていますか
- ・ 外部への情報発信をするとき：守秘義務・著作権の問題はありませんか
- ・ 外部の人とシステムの話をするとき：NDA（秘密保持契約）が締結されていますか

NDA（秘密保持契約）は、難しそうな名前ですが、「ここだけの話を、外に漏らさない約束」です。社外の人と仕事の中身を話す前に、その約束ができているかを確認する。これは相手を疑うことではなく、お互いを守るための、ビジネスの作法です。

■ グレーゾーンへの向き合い方—「迷ったらしない」の原則

- ・ まず相談する：「自分だけで判断しない」が基本、上司・法務担当・社外の弁護士に確認します
- ・ 保守的に判断する：**迷ったら「しない」を選びます。不法行為の責任は「やらかした側」にあります**
- ・ 記録を残す：「〇〇について法務に確認した」「〇〇という理由でこの判断をした」という記録が後の保護になります

記録は、あなたを守ります。「いつ・誰に・何を確認し、どう判断したか」を残しておけば、後で「なぜそうしたのか」を説明できます。正しく確認して進めたことなら、たとえ問題が起きても、それは個人の責めではなく、仕組みで対処すべきことになります。

- ・ 「みんなやっている」は根拠にならない：業界慣行として広く行われていても、法律違反は法律違反です

「迷ったらしない・相談する」は、臆病なのではなく、賢いのです。法的リスクは、やってしまってからでは取り返しがつきません。一方、やる前に相談すれば、たいていは安全な進め方が見つかります。新卒のうちは“石橋を叩く”くらいがちょうどよく、「念のため確認したいのですが」と言える人は、むしろ信頼されます。

事例から学ぶ：「うっかり」が重大事故になった事例—法律知識が守るもの

あるIT企業の新卒エンジニアが、テストのために「ちょっと確認しよう」と本番DBの顧客データをローカルPCにコピーしました。

そのPCを紛失したため数万件の個人情報が漏洩し、個人情報保護法違反による社内処分と顧客への損害賠償が発生しました。

「悪意はなかった」という事実は法的責任を消しませんでした。

この教訓から学べること：法律上の義務は「知っているかどうか」にかかわらず課されます。

入社初日から「個人情報は本番データを使わない」「不必要なデータを保持しない」という習慣を徹底することが、自分と会社を守ります。

💡 ポイント

法律知識は「問題が起きたときのため」だけのものではありません。法律を知っていることで日常の設計判断・開発判断に「法的な安全性」という軸が加わります。

技術的に正しいコードと法的に安全な設計の両方を意識できるエンジニアは、チームにとって非常に価値の高い存在です。

▶ やってみようワーク・演習

やってみよう追加：「法的に問題のある設計」を発見するレビュー

以下のシステム設計のシナリオを読み、法的・倫理的な問題点を指摘してください。

【シナリオ：ユーザー会員登録システムの設計】

- ・登録時にメールアドレス・氏名・生年月日・電話番号・位置情報（任意）を収集する
- ・パスワードはMD5でハッシュ化してDBに保存する
- ・ユーザーのアクセスログを無期限に保持する
- ・マーケティング目的で収集したメールアドレスを提携企業と共有する
- ・テスト環境でも本番と同じユーザーデータを使う
- ・ユーザーが退会しても個人情報を5年間保持する

①個人情報保護法の観点から問題点を3つ挙げる

②セキュリティの観点から問題点を2つ挙げる

③各問題点の修正案を提案する

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・ ネットで見つけた文章や画像を使うとき「使ってよい条件か」を確認するクセをつける。

AIが法律の文章を要約したり、契約のたたき台を作ったりできる時代になりました。それでも、「この場面でどの法律が関わり、何に気をつけるべきか」を判断し、AIの答えが本当に正しいかを確かめるのは、人間の仕事です。法律の“全体像（体系）”を知っておくことが、AIを安全に使いこなす土台になります。

💡 ポイント：法律は、あなたと仲間を守る“盾”

法律を「自分を縛るもの」と感じると、覚える気になれません。でも見方を変えれば、法律は、不当な扱いから自分を守り、トラブルから会社を守り、被害から社会を守る“盾”です。

盾の使い方を知っておくと、安心して前に進めます。

11-8 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ 民事・刑事・行政の責任の違いを、ざっくり。
- ・ 仕事に関わる法律を、思い出せるだけ。
- ・ 迷ったときは、どうするか。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、これからの学びや仕事の中で、少しずつ埋めていきましょう。

第12章 人を傷つけないために—ハラスメント・感情・誠実・公正（24日・1日）

この章のゴール

この章では、ハラスメント・感情・言葉・誠実さ・公正を通して、「自分も、まわりの人も傷つけないこと」を学びます（「自分を守る・健全な組織をつくる」は、続く第13章で学びます）。

終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ①【**考え方がわかる**】ハラスメントや不正は「悪意がなくても・普通の人でも」起こると、わかる。
- ②【**危機感がもてる**】軽い一言・嘘・隠しごとが、人やキャリアを大きく傷つけうると、実例でつかめる。
- ③【**考え方がわかる**】誠実であること（隠さない）と、公正であること（えこひいきしない）が、なぜ大切かがわかる。

まず自己診断（読む前に・2分）

- ① 現時点での「厳しい指導とハラスメントの違いを自分の言葉で説明できる自信」を1～5でメモ。
- ② 「冗談や"よかれと思って"の一言が、相手を傷つけることはあると思いますか？」を一言メモ（正解は気にしません）。

12-1 この2章で目指す3つの立場—加害者にならない／身を守る／巻き込まれない

この章と、続く第13章は、この研修の中でも、とりわけ大切なテーマを扱います。最初に、はっきりお伝えしておきたいことがあります。これは、**あなたをおどかすための章ではありません**。むしろ逆で、これから働くあなたが、つらい目に遭わず、誰かを傷つけてしまうこともなく、安心して長く働き続けるための「持ち物」を、まとめて渡しておくための章です。

働くとは、人と人とが毎日、近い距離で関わり続けることです。助け合いや喜びがある一方で、人が集まる場所では、人が人を傷つけること（ハラスメント）や、組織が道を外すこと（不正）も起こります。特別に悪い会社だけの話ではありません。だからこそ、配属前の今、知っておく価値があります。

この第12章と、続く第13章では、ひとつのテーマを、3つの立場から見ていきます。第12章ではおもに「加害者にならない」を、第13章では「自分を守る」「巻き込まれない（そして健全な組織をつくる）」を扱います。

第一に、「**加害者にならない**」。これは、知らないうちに、よかれと思って、あるいは何気ない一言で、誰かを傷つけてしまわないための視点です。あとで見るように、ハラスメントの多くは「悪い人」ではなく「ふつうの人」が起こします。だからこそ、自分も例外ではない、という前提で学びます。

第二に、「**被害者として身を守る**」。もし、あなた自身が傷つけられそうになったとき、自分を守るようにしておく視点です。「我慢するしかない」と思い込まないために、知識と相談先と、いくつかの具体的な動き方を持っておきます。

第三に、「**巻き込まれない**」。これが、この章のいちばん大事な裏のテーマです。世の中には、残念ながら、ルールや人への敬意を軽んじる人もいます。もしそういう人や場に出会っても、あなたが流されず、染まらず、共倒れにならないための構えを持っておきます。

💡 ポイント：この章の正しい使い方

この章は「暗記する」章ではありません。「**健全な職場やリーダーとは、こういうものだ**」という**"ものさし"**を、先に手に入れておくための章です。

人は、基準を知らないと、おかしいことが起きていても「こういうものなのかな」と受け入れてしまいます。逆に、ものさしを先に持っていれば、配属されてしばらくして「あれ？なんだか違うぞ」と、自分で気づくことができます。その"気づける力"こそが、何よりあなたを守ります。今は意味が分からない部分があっても大丈夫。配属後に「あ、これか」と思い出せれば、それで十分に役目を果たします。

なお、この章で扱う多くのテーマは、他の章ともつながっています。判断の土台となる「正しさの4層」は第10章で、安心して働けるチームの土台である「心理的安全性」は第6章で、すでに学びました。この章は、それらを「人を傷つけない・自分を守る」という角度から、もう一度束ね直すものだと考えてください。

もう一つ。この章には、つらい事例も出てきます。こわがるためではなく、「こういうことが起こりうる」と知り、「自分には備えがある」と思えるようになるためです。どの怖い話のあとにも、必ず「では、どうすればよいか」を置いています。だから、安心して読み進めてください。

12-2 ハラスメントの全体像—種類・3要件・360度

まずは、地図を持ちましょう。細かい分類を丸暗記する必要はまったくありません。大切なのは、「ハラスメントには、いろいろな形がある」「自分が思っているより広い」と知っておくことです。

問題になるハラスメントには、パワハラ（立場の強さを背景にした言動）、セクハラ（性的な言動）、マタハラ・パタハラ（妊娠・出産・育児を理由にした不利益）、モラハラ（無視・人格否定）、アルハラ（飲酒の強要）、カスハラ（顧客からの過剰な要求）などがあります。名前は増えていますが、根は一つ、「相手の尊厳を軽んじ、安心して過ごせる環境を壊す」ことです。

「種類が多くて覚えられない」と感じて大丈夫です。覚えるべきは名前ではなく、「自分の何気ない言動も、どれかに当てはまるかもしれない」という感覚のほうです。

なぜ、こんなに多くの「〇〇ハラ」があるのでしょうか。社会が「働きやすい環境とは何か」を年々まじめに考えてきた証でもあります。昔は「当たり前」と我慢を強いられたことに、一つずつ名前がつき、見えるようになってきたのです。大事なことは名前の暗記ではなく、「自分の“当たり前”も、誰かを傷つけているかもしれない」と立ち止まれることです。

■ パワハラの3要件と6類型

数あるハラスメントの中でも、職場でいちばん身近なのがパワハラです。これには、おおまかな成立の目安があります。

💡 ポイント：パワハラの3要件（ざっくり）

次の3つがそろえば、パワハラと判断されやすくなります。

- ① **優越的な関係を背景**にしている（上司と部下だけでなく、知識・経験・人数の差なども含みます）。
- ② **業務上必要な範囲を超えている**（指導の必要性を超えた、過剰なもの）。
- ③ **就業環境を害している**（相手が働きづらくなっている）。

逆に言えば、仕事に必要で、適切な範囲の指摘は、パワハラではありません。ここはあとで（12-2

末) もう一度ていねいに扱います。

厚生労働省は、パワハラを大きく6つの型で整理しています。これも「こんな形があるのか」と眺めるだけで十分です。

1. **身体的な攻撃**（叩く、物を投げるなど）
2. **精神的な攻撃**（人前での長時間の叱責、人格否定、暴言）
3. **人間関係からの切り離し**（無視する、一人だけ会議や情報から外す、仲間外れにする）
4. **過大な要求**（とても終わらない量や、無理な納期を押し付ける）
5. **過小な要求**（わざと簡単すぎる仕事しか与えない、あるいは仕事を与えない）
6. **個の侵害**（プライベートに過度に立ち入る、私的なことを言いふらす）

特に③や⑤は、「怒鳴る」ような分かりやすい形ではないため、見過ごされがちです。しかし、無視され続ける、自分にだけ仕事が回ってこない、というのも、立派なハラスメントになりえます。

■ ハラスメントは「360度」で起こる

「ハラスメント＝上司が部下にするもの」というイメージがあるかもしれませんが。これは半分しか当たっていません。実際には、**あらゆる方向**で起こります。

- ・ **上から下へ**：上司から部下へ（典型的なパワハラ）。
- ・ **下から上へ**：部下が集団で上司を無視する、SNSで攻撃する（逆パワハラ）。
- ・ **横へ**：同僚どうし、同期どうしの嫌がらせ。
- ・ **社内から社外へ**：取引先や下請けへの高圧的な態度。
- ・ **社外から社内へ**：顧客からの理不尽な要求（カスハラ）。

💡 ポイント：「自分は立場が下だから関係ない」は誤り

新人のうちは「自分は一番下だから、加害者にはならない」と思いがちです。でも、横の関係（同期どうし）でも、SNS上でも、加害は起こります。立場に関係なく、誰もが加害者にも被害者にもなるうる、これが「360度」の意味です。なお、上の立場の人が被害者になることもあります（逆パワハラ）。「立場が上なんだから、それくらい我慢すべき」というのも、思い込みです。

■ いちばん大事な区別：「正当な指導」と「ハラスメント」は違う

ここで、多くの新人が不安に思うことに、先に答えておきます。「厳しいことを言われたら、全部ハラスメントなの？」、いいえ、違います。ここを誤解すると、必要な指導まで「ハラスメントだ」と感じて萎縮してしまい、かえって成長の機会を失います。

🏛️ 知っておこう：指導とハラスメントの分かれ目

同じ「厳しさ」でも、次の点で分かります。

- ・ **向ける先**：「行動・結果」に向いているか（指導）、「人格・存在」を否定しているか（ハラスメント）。「この手順が抜けていた」は指導。「だからお前はダメなんだ」は人格否定。
- ・ **場所**：人前でさらしものにしていないか。
- ・ **程度・しつこさ**：必要な範囲か、執拗・長時間に及んでいないか。
- ・ **目的**：相手の成長のためか、自分の感情の発散か。

成長のためのフィードバックは、むしろありがたいものです（受け取り方は第17章）。「厳しさ＝悪」ではありません。萎縮しすぎず、でも人格を否定されたら「それは違う」と気づけること。その両方を持っておきましょう。

12-3 「悪意がない」が一番危ない—気づかない加害

ハラスメントと聞くと悪人を想像しがちですが、職場で多いのは「よかれと思って」「冗談のつもりで」やってしまうケースです。多くの加害者は、自分が加害しているとすら気づいていません。ここがいちばん怖いところです。

なぜ怖いのか。それは、「自分は悪い人間ではないから大丈夫」という思い込みが、いちばんの落とし穴になるからです。善良なあなたほど、「まさか自分が」と考えて、無自覚に踏み外してしまう危険があります。

もう少し仕組みを見ます。人は立場が上がったり、相手と親しくなると感じると、「これくらい大丈夫」という感覚が知らぬ間にゆるみます。「親しいから」「自分も言われて育った」「相手のため」、どれも本人の中では筋が通っているので、自分では止められません。歯止めは一つ、「相手は今どう感じているか」を想像する習慣です。立場や親しさは、相手の気持ちを勝手に決めてよい理由にはなりません。

💡 ポイント：成立するかどうかは「受け手」で決まる

これは、この章でいちばん覚えておいてほしい原則です。

ハラスメントが成立するかどうかは、する側の"意図"ではなく、受け手が感じた不利益や苦痛で判断されます。

「そんなつもりじゃなかった」「冗談だった」「愛情からだった」、どれも、残念ながら免責にはなりません。意図が良くても、相手が傷ついていれば、成立するのです。

だからこそ、予防はシンプルです。何かを言う前・する前に、「相手はこれをどう感じるだろう」と、一拍おいて想像する。この一手間が、あなたを「無自覚の加害者」にしないための、最強の盾になります。

■ 無意識の思い込み（アンコンシャス・バイアス）

人は誰でも、無意識のうちに「決めつけ」を持っています。「男性だから力仕事を」「女性だからお茶くみを」「若いから飲み会の幹事を」「地方出身だから～」。本人にはまったく悪気がなく、むしろ「気をきかせたつもり」のことすらあります。けれども、こうした決めつけは、相手の可能性を狭め、静かに傷つけます。これを**アンコンシャス・バイアス（無意識の偏見）**と呼びます。

また、一つひとつは小さな、しかし繰り返される刺さる一言を**マイクロアグレッション（小さなトゲ）**と言います。「君にしては良くできたね」「（外国にルーツのある人へ）日本語うまいね」。一回なら聞き流せても、毎日浴び続ければ、ボディブローのように効いてきます。

性的指向や性自認に関するからかい・決めつけ（SOGIハラ）も同じです。とくに、本人が望まないのにその人の性的指向などを別の人へ明かすこと（アウトティング）は、絶対にしてはいけません。人生を左右しかねない、重いことだからです。

セクハラも、体に触れることだけではありません。容姿を繰り返し言う、恋愛・結婚をしつこく尋ねる、性的な冗談を言う、性別で役割を決めつける、こうした言葉も該当します。

◆ 考えてみよう

あなたが今までに言われて「もやっとした」一言を、一つ思い出してみてください。その相手に、悪気はあったでしょうか。おそらく、なかったかもしれません。

「悪気のなさ」と「相手が傷つくこと」は、同時に成り立ちます。この事実こそが、この節の核心です。そして、あなた自身も、悪気なく誰かを「もやっ」とさせているかもしれない、と想像できることが、優しさの第一歩です。

■ 自分が「受けている側」のサインに気づく

加害を防ぐだけでなく、もう一つ、とても大切なことがあります。それは、**自分が傷つけられているときに、それに気づけること**です。

意外に思うかもしれませんが、被害を受けている人ほど、「自分が悪いのかも」「考えすぎかも」「これくらい普通かも」と、自分を責めたり、麻痺してしまったりします。そして、気づいたときには心や体が限界、ということが起こります。

知っておこう：これは「考えすぎ」ではないかもしれません

次のような状態が続くなら、それはあなたが弱いのではなく、環境のほうに問題があるサインかもしれません。

- ・ 特定の人の前に出ると、頭が真っ白になる、声が震える。
- ・ 朝、その職場に行くことを考えると、強い気の重さや体の不調がある。
- ・ 自分にだけ、必要な情報や仕事が回ってこない。
- ・ 人格や存在を否定する言葉を、繰り返し浴びている。

あなたの「つらい」という感覚は、それ自体が大切な情報です。「気のせい」と打ち消す前に、まずメモしておきましょう（記録の仕方は13-3で）。

なお、多様な人への配慮は、第10章「正しさの4層」や第16章「働く権利」とも深くつながっています。

12-4 ハラスメントと法律—自分と仲間を守る知識

法律と聞くと、難しく、自分には縁遠いものを感じるかもしれません。でも、ここで扱う法律は、**あなたと仲間を守るための道具**です。「守ってくれるものがある」と知っているだけで、いざというとき、ずいぶん心強くなります。要点だけ、やさしく押さえましょう。

■ 会社を縛り、あなたを守るルール

- ・ **パワハラ防止法**（正式には労働施策総合推進法）：会社に対して、ハラスメントを防ぐ措置（相談窓口の設置など）を**義務**として求めています。つまり、ハラスメント対策は「会社の善意」ではなく「会社の義務」です。
- ・ **男女雇用機会均等法／育児・介護休業法**：セクハラ、マタハラ、パタハラなどを禁止し、会社に防止を求めています。
- ・ **会社の安全配慮義務**（労働契約法）：会社には、働く人の心と体の安全に配慮する義務があります。ハラスメントを放置した会社は、この義務に違反したとして責任を問われます。

知っておいてほしいのは、これらの法律は「会社をしぼるもの」であると同時に、「あなたが声を上げたときの後ろ盾」だということです。相談を理由に会社が不利益な扱いをすることは、法律で禁

じられています。条文を覚える必要はありません。「自分には守ってくれる仕組みがある」と知っておくだけで、いざというときの一歩が軽くなります。

■ 「会社が払うから自分は大丈夫」は通用しない

ここは、加害者にならないために、はっきり知っておいてほしいところです。

💡 ポイント：加害した「個人」も責任を負う／会社も連帯する

ハラスメントをした**本人（個人）**は、民法上の損害賠償責任（不法行為責任）を負います。つまり、自分の財布から賠償することがあります。さらに会社も「使用者責任」として連帯して責任を負うことがあり、社内的には、悪質な場合、加害者は**懲戒処分や懲戒解雇**もありえます。

「組織の問題だから」「会社が対応するから」と他人事にはできません。**たった一度の言動が、自分の賠償・処分・信用・キャリアに、そのまま跳ね返ってくる**のです。

■ 言葉に関わる刑事の話（詳しくは12-6）

暴言や侮辱は**侮辱罪**、たとえ事実であっても公然と人の評価を下げれば**名誉毀損罪**に問われることがあります。これは飲み会の席でも、社内チャットでも例外ではありません。詳しくは12-6で扱います。

⚖️ 知っておこう：早く声を上げた人を守る仕組みがある

「不正やハラスメントを通報したら、自分が不利になるのでは」と不安になるのは自然なことです。しかし、通報した人を会社が不利益に扱うことを禁じる**公益通報者保護法**という仕組みがあります（詳しくは第15章）。新人がいきなり使う場面は少ないとしても、「正しいことを伝えた人を守る仕組みがある」と知っておくこと自体が、心の支えになります。一人で抱え込まなくて大丈夫です。

💡 ポイント：「カスハラ対策」は、会社の義務になります（2026年10月～）

お客さまや取引先からの、度を越した要求や暴言（カスタマーハラスメント）について、2025年の法改正で、企業に防止対策をとることが**義務づけられました**（労働施策総合推進法の改正・2026年10月1日施行）。あわせて、就職活動中の人に対するセクハラ対策も義務化されます。

つまり、「お客さまだから、何を言われても我慢するしかない」という時代は、終わりつつあります。客先で理不尽な言動を受けたとき、それはあなたが一人で耐えるものではなく、**会社を守るべきもの**です。「こんなことを言われて困っている」と、自社に上げて構いません。

なお、法律全体の見取り図は第11章で、コンプライアンス・公益通報・内部統制の詳細は第15章で扱います。

12-5 感情とつきあう—怒りを制御し、他人の怒りに飲まれない

職場のハラスメント、とくにパワハラやモラハラの多くは、ある共通の入り口から始まります。それは、**怒りの衝動**です。「カッとなって」「イライラして」、つい強い言葉が出てしまう。だからこそ、感情との上手なつきあい方を身につけることは、加害者にならないための、もっとも実践的な技術になります。

最初に、誤解を解いておきましょう。**怒りは、悪い感情ではありません**。怒りは、「自分の大切にしている価値観や期待が裏切られた」と感じたときに生まれる、ごく自然な心の反応です。問題な

のは「怒ること」ではなく、「怒りを、相手を傷つける形で表に出してしまうこと」です。怒りを感じるのと、それをどう表現するかの間には、実は**選ぶ余地**があります。その余地を、意識して少し広げる、それがアンガーマネジメントです。

なぜ怒りは人間関係を壊すのでしょうか。強い怒りの最中は、冷静に考える働きが弱まり、「相手をやり込めたい」衝動が前に出ます。その勢いで、言うつもりがなかった人格否定の一言が飛び出す。そして言われた側は、それを何年も覚えています。怒りは数分で消えても、言葉の傷は消えません。だから「怒りを感じた瞬間」と「言葉にする瞬間」の間に、数秒の間を差し込む、これが分かれ目です。練習で必ず上達します。

ポイント：怒りと上手につきあう、小さな技術

- ・ **6秒待つ**：怒りの衝動のピークは、数秒～6秒ほどと言われます（諸説あり・あくまで目安）。カッとしたら、すぐ口を開かず、一呼吸おく。それだけで、後悔する一言を、かなり減らせます。
- ・ **スケール化する**：「今の怒りは10段階でいくつだろう？」と頭の中で数字にしてみる。数値にするだけで、不思議と一歩引いて自分を見られます。
- ・ **「べき」をゆるめる**：「普通～すべきだ」「常識的に～のはずだ」という強い思い込みが、怒りを大きくします。「そう考えない人もいるよな」と視野を広げると、怒りは自然と和らぎます。
- ・ **その場を離れる**：どうしても収まらないときは、「少し席を外します」と物理的に距離をとるのも、立派な技術です。

伝え方そのもの、相手の話をしっかり聴く**傾聴**や、自分も相手も尊重して伝える**アサーティブ**な話し方、は、第6章でくわしく学びました。怒りを「あなたが悪い！」とぶつけるのではなく、「私はこう感じた」という形（アイメッセージ）で伝えると、相手も身構えずに受け取れます。

そして、この節でもう一つ、ぜひ渡しておきたい力があります。それは、自分が怒らないことだけでなく、**他人の怒りに飲み込まれない**力です。

知っておこう：他人の怒りに飲まれない（バウンダリー＝心の境界線）

配属後、理不尽な怒りをぶつけられることがあるかもしれません。そのとき、**あなたがそれを全部、まともに受け止めて、自分を責める必要はありません。**

「相手の怒りは、相手の課題」。心の中に、そっと一本の線を引いていいのです。それは冷たいことではなく、自分の心を守るための、健全な距離の取り方です。理不尽な感情を浴びたら、（1）少し物理的に距離をとる、（2）言われた事実を淡々とメモする、（3）信頼できる人に話す。この順番を覚えておきましょう。あなたが萎縮して小さくなる必要は、まったくありません。

同時に、理不尽な怒りを浴びせる言動そのものを防ぐのは、組織の責任です。心の境界線で身を守るのは大切ですが、それは会社が職場からそうした言動をなくす義務を免れる理由にはなりません。

事例から学ぶ：「怖くて聞けなかった」一言が招いた手戻り

ある新人は、先輩の口調がいつも強く、質問するのが怖くて、分からないまま自己判断で作業を進めてしまいました。結果、大きな手戻りが発生。新人は「自分の確認不足だ」と一人で抱え込みましたが、本当の問題は、「気軽に質問できない空気」のほうにありました。

→ **あなたにできる一歩**：怖くて聞けないとき用の「最初の一言」を、あらかじめ用意しておきましょう。「すみません、一分だけ確認させてください」。この短い一言が、事故も、あなた自身も守りま

す。質問は、能力が低い証拠ではなく、仕事を前に進めるプロの行動です。

12-6 言葉は凶器になる—陰口・呼び捨て・SNS・記録

「ちょっとした冗談だった」「飲み会の席のノリだった」「仲間内のグループチャットだった」。実際に裁判やトラブルに発展した出来事の多くは、まさにこういう「軽い気持ち」から始まっています。言葉は、便利な道具であると同時に、人を深く傷つける凶器にもなります。この節では、とくに見落とされがちな「言葉のリスク」を、正面から扱います。

■ 3つのよくある誤解

💡 ポイント：言葉にまつわる3つの誤解

- ・ **「飲み会の席だから、業務外で問題ない」** → 業務の内外にかかわらず、名誉毀損や侮辱は成立します。場所はありません。
- ・ **「本当のことを言っただけ」** → ここは多くの人が誤解しています。日本の名誉毀損は、たとえ**内容が事実でも**、公然と人の評価を下げれば成立します（公益目的などの例外はあります）。「事実なら何を言ってもいい」わけではないのです。
- ・ **「クローズドなチャットだから外に出ない」** → スクリーンショット一枚で、いくらでも外に出ます。削除しても、相手の画面には残っています。「仲間内だけ」は、もう存在しないと考えましょう。

■ 侮辱罪の厳罰化と、特定される時代

侮辱罪は、2022年に大きく厳罰化されました。それまでは比較的軽い処罰でしたが、改正後は**1年以下の拘禁刑、または30万円以下の罰金**まで引き上げられました。その背景には、SNS上の中傷で人が命を絶つという、痛ましい出来事がありました。

また、「匿名なら大丈夫」も、もはや通用しません。発信者の情報を開示させる手続きは年々スピードアップしており、匿名の投稿でも、書いた人が特定される時代です。

■ 呼び捨て・あだ名・「いじり」

意外に見落とされるのが、日常の呼び方や「いじり」です。呼び捨て、嫌がるあだ名、容姿や失敗を笑いものにする「いじり」、盛り上げているつもりでも、本人が嫌なら、それはハラスメントの芽です。健全な職場では、立場に関係なく敬意ある言葉づかいが保たれています。

「たかが言葉」と思うかもしれませんが、けれど言葉は、消えずに心に残り続けます。しかも今は文字として記録・拡散され、一度ネットに出ると完全には消えません（デジタルタトゥー。くわしくは第14章で扱います）。軽い気持ちの一行が、何年もあとに自分の名前とともに掘り起こされることもあります。発する前・送る前の一呼吸が、未来の自分を守ります。

📖 事例から学ぶ：SNSの中傷で、命が失われた出来事から

ある人が、SNS上の中傷に苦しみ、自ら命を絶ちました。投稿者には賠償が命じられ、こうした出来事が侮辱罪の厳罰化（2022年）につながりました。「有名人だから」「みんな言っている」は通用しません。言葉は人の命さえ奪う力を持つ、そのことを重く受け止める必要があります。

「有名人なんだから、少しくらい批判されて当然」「みんな言っているから」、こうした考えは、完

全な誤りです。

→ **あなたにできる一步**：投稿ボタン・送信ボタンを押す前に、3秒だけ止まる。「これを、本人の目の前で、面と向かって言えるか？」と自分に問う。言えないなら、書かない。それだけで、ほとんどの事故は防げます。

事例から学ぶ：「仲間内のチャット」が証拠になった

ある職場で、中途入社と同僚について、生え抜きの社員グループがクローズドな社内チャットで「外様が来た」「前の会社はたいしたことない」などと繰り返し書き込んでいました。そのやり取りは記録として残っており、後に大きな問題になりました。「閉じた場所だから外に出ない」という思い込みが、いかに危ういかを示しています。

→ **あなたにできる一步**：チャットやメールは「いつか誰かに読まれる前提」で書く。逆に、自分が理不尽な書き込みを「された」側になったら、その画面こそが、あなたを守る証拠になります。

なお、SNS・情報発信のルールと、デジタルタトゥーの怖さは、第14章でさらに詳しく扱います。

12-7 誠実さ—嘘をつかない・隠さない・ごまかさない

ここで、ハラスメントから少し視点を移します。けれど、根っこは同じ「人と組織の健全さ」の話です。テーマは、**誠実さ（正直であること）**。技術の世界で長く信頼される人に共通するのは、頭の良さや器用さよりも、まず「誠実であること」です。逆に、一度の嘘やごまかしが、積み上げた信頼を、一瞬で崩してしまうことがあります。

新人が最初につまずきやすいのが、「ミス隠してしまふ」ことです。これは、能力の問題ではなく、「怒られたくない」「がっかりされたくない」という、とても人間的な気持ちから起こります。だからこそ、誰にでも起こりうる。その前提で、次の事例を読んでください。

事例から学ぶ：小さなミスを「隠した」ことで、致命傷になった

ある新人が本番環境で設定を誤り、データの一部を消してしまいました。

ところが新人は、「怒られる」と怖くなり、自分一人でも何とかしようと、別の操作を重ねてしまいます。これが裏目に出て、被害はどんどん広がりました。報告は半日遅れ、復旧は翌日までかかる大ごとになった。

最初のミスは、本当に小さかったのです。それを致命傷にしたのは、ミスそのものではなく、「隠したこと」でした。

→ **あなたにできる一步**：ミスは「早く・正直に」。これが鉄則です。報告の速さが、被害の大きさを決めます。「隠す」という選択は、必ず、何倍も怖い結果を連れてきます。そして覚えておいてください。早く正直に報告できる新人を、まともな先輩は、決して見捨てません。

嘘・データの改ざん・「できていないのにできたと報告する」ことは、技術者として絶対に越えてはいけない一線です。私たちのつくるシステムは人の生活や安全に関わり、一つの嘘が後で大きな事故につながることもあります。これは第21章で学ぶ「予防倫理」の中心です。

誠実さは地味で、すぐには評価されにくいものです。正直に「分かりません」と言うのは、その場ではかっこ悪く感じるかもしれませんが、けれど長い目で見れば、これほど確実に信頼を積み重ねる方法はあ

りません。「あの人の報告は正確だ」「都合の悪いことも隠さない」と思われる人に、大きな仕事と重要な情報が集まります。逆に一度ごまかすと、その後どれだけ正しくても疑われます。誠実さは、コツコツしか貯まらないのに一度で失う、繊細な財産です。

✦ 考えてみよう

想像してみてください。「正直にミスを言ったら、激しく怒られて、人格まで否定される職場」と、「正直に言えば、まず一緒に対処してくれて、その後で再発防止を考える職場」。

あなたが安心して、長く力を出せるのは、どちらでしょうか。その答えは、実は、次の節からつながっていく「健全な組織」「心理的安全性」の話と、まっすぐにつながっています。誠実さは、個人の心がけであると同時に、それを支える「場の空気」とセットなのです。

12-8 公正であること—機会と評価は公平に開かれているか

健全な職場には、もう一つ、欠かせない要素があります。それが**公正さ（フェアであること）**です。具体的には、**チャンスと評価が、個人の好き嫌いではなく、きちんとした基準と事実にもとづいて、みんなに公平に開かれている**こと。これは、働く人が安心して力を出すための、土台になります。

考えてみれば、当たり前のことです。どれだけ頑張っても、結局は「気に入られているかどうか」で仕事や評価が決まるなら、誰も、まっすぐ努力しようとは思えなくなります。逆に、「やったことが、ちゃんと見てもらえる」と信じられる職場では、人は安心して挑戦できます。

💡 ポイント：公正さの"ものさし"

健全な職場かどうかは、次のような点で見えてきます。

- ・ 評価や、仕事の割り振りに、**説明できる基準**があるか。「なぜこの評価なのか」を、本人が納得できる形で説明してもらえるか。
- ・ 一部の「お気に入り」だけが、よい仕事やチャンスを独占していないか（えこひいき）。
- ・ **実力や事実ではなく、"立場や仲の近さ"だけ**で、物事が決まっていないか。
- ・ 私的な仲の良さが、そのまま人事や評価に持ち込まれていないか（公私混同）。

逆に言えば、好き嫌いで人を厚遇したり冷遇したりすることは、それ自体が、ハラスメント（特定の人を「人間関係から切り離す」「過小な要求しか与えない」）に育っていくことがあります。**公平さは、単なる優しさではなく、組織を強くし、人を育てるための"仕組み"**なのだと考えてください。

少し想像してください。同じ失敗でも、お気に入りは笑って許され、そうでない人は厳しく責められる。同じ成果でも、声の大きい人だけが評価される。こうなると、職場から正直さとやる気が失われます。「どうせ見ていないなら手を抜こう」となるからです。公正さを欠いた職場が弱るのはこのため。あなたが将来評価する側になったら、「好き嫌いを持ち込まない」と決めるだけで、チームを健全に保てます。

そして、これも「ものさし」の一つです。もし将来、「どうも、特定の人ばかりが優遇されているな」「自分のこの評価は、何を基準にしているんだろう」と感じることがあったら、その違和感は、決して、わがままや甘えではありません。健全さを測る、まっとうな感覚です。

12-9 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ ハラスメントが成立するのは、する側の「意図」と受け手の「苦痛」、どちらを基準にするでしょうか。
- ・ 「正当な指導」と「ハラスメント」の分かれ目を、思い出せるだけ（向ける先・場・程度…）。
- ・ 「誠実さ（隠さない）」と「公正さ（えこひいきしない）」は、なぜ大切なのでしょうか。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、もう一度つけてみましょう。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、これからの学びや仕事の中で、少しずつ埋めていきましょう。次の第13章では、健全な組織をつくる側の視点を学びます。

第13章 健全な組織をつくるために—知っておくこと、できること（25日目・1日）

🎯 この章のゴール

この章では、「健全な組織とはどういうものか」を知り、あなた自身がその健全さをつくり、守る一員になるために、知っておきたいことと、今日からできることを学びます。前章（第12章）の「人を傷つけない」を受けて、ここでは「健全な場を見分け、育てる。そして自分も大切にする」へ視野を広げます。

終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ①【**考え方がわかる**】健全な組織と、そうでない組織の違いが、自分の言葉でわかる（ふつうの人でも空気がしだいで流される、という仕組みも含めて）。
- ②【**ものさしがもてる**】健全さの「ものさし」を持ち、よい点も、気になる点も見えるようになる。
- ③【**動ける**】健全な場を育てる小さな行動（声をかける・相談する・記録する）と、困ったときの相談先がわかる。

💡 まず自己診断—読む前に、いまの自分を知ろう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「健全な組織とはどういうものか」を、自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
- ② 「みんなやっているから」と言われたら、あなたはこうだと思いますか？」を、いまの考えで一言だけ書きます（正解は気にしません）。

読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

はじめに、ひとつだけ。この章は、「健全な組織とはどういうものか」を知り、あなた自身がその健全さをつくる一員になるための章です。多くの職場は、完璧ではなくても健全に営まれていて、日々の仕事の多くは、支え合いと信頼の上に成り立っています。そして、その「健全さ」は、誰か一人が与えてくれるものではなく、そこで働く一人ひとりが、毎日少しずつ育てているものです。

もちろん、どんな世界にも、まれに「健全とは言いにくい状況」が生まれることはあります。そんなときに、自分を大切にしながら、健全さを取り戻す側に立てるように、この章では、知っておきたいことと、今日からできる小さな行動を、一緒に確かめていきます。新人のうちから、あなたはもう、健全な組織づくりの立派な担い手です。どうか、肩の力を抜いて読んでください。

13-1 組織の健全さを支えるもの—そして、人が流される仕組みを知る

ここで、視野をぐっと広げます。これまでは「個人と個人」の話が中心でした。ここからは、「組織」そのものが道を外していくとき、何が起こるのか、を見ていきます。少し難しく聞こえるかもしれませんが、**健全な組織をつくり、守るために、ぜひ知っておいてほしい話**です。

まず、健全な組織の姿を確かめておきましょう。健全な組織では、ルールが守られ、人が尊重され、悪い知らせも安心して報告できます。多くの組織は、完璧ではなくても、この方向を目指して日々運営されています。そのうえで、まれに起こる「そうではない状態」を、これから見ていきます。誰かを責めるためではなく、もし出会ったときに気づけるように、です。

まず、大事な事実から。ハラスメントが横行している組織は、たいていの場合、**組織の統治（ガバナンス）が弱っています**。「人を恐怖や圧力で動かす文化」と「不正やごまかしが起きる土壌」は、

別々のものではなく、同じ根から生えた、一本の木の枝なのです。

そして、もっとも恐ろしいのは、そういう組織では、**もともと善良だった、ごく普通の人たちが、いつのまにか不正に加担してしまう**ということです。「自分は絶対にそんなことしない」と、今は思うでしょう。でも、人間には、そう簡単に抗えない「重力」のようなものが働きます。

💡 **ポイント：人を流す3つの「重力」**

- ・ **同調圧力**：「みんなやっているから」。周りが全員そうしていると、自分一人だけ違う行動をとるのは、驚くほど難しくなります。
- ・ **権威への服従**：「上の指示だから」。立場が上の人の指示には、たとえ違和感があっても、つい判断を預けて従ってしまいます。
- ・ **沈黙の空気**：「言っても無駄」「波風を立てたくない」。誰もが薄々おかしいと思っ
ても、誰も口に出さない。その沈黙が、問題を温存します。

これらは、あなたが弱いから働く力ではありません。**人間なら誰でもかかる重力**です。だからこそ、「こういう力が働くんだ」と、先に知っておくこと自体が、抵抗力になります。

不正は、研究上、「**機会・動機・正当化**」の3つがそろったときに起きやすいと言われます（不正のトライアングル）。「やろうと思えばできる状況（機会）」「やらざるをえない事情（動機・たとえば過大なノルマ）」「自分を納得させる言い訳（正当化・「みんなやっている」「会社のためだ）」」。この3つがそろう職場は、危険です。

不正のトライアングルー どれか一つを断てば防げる

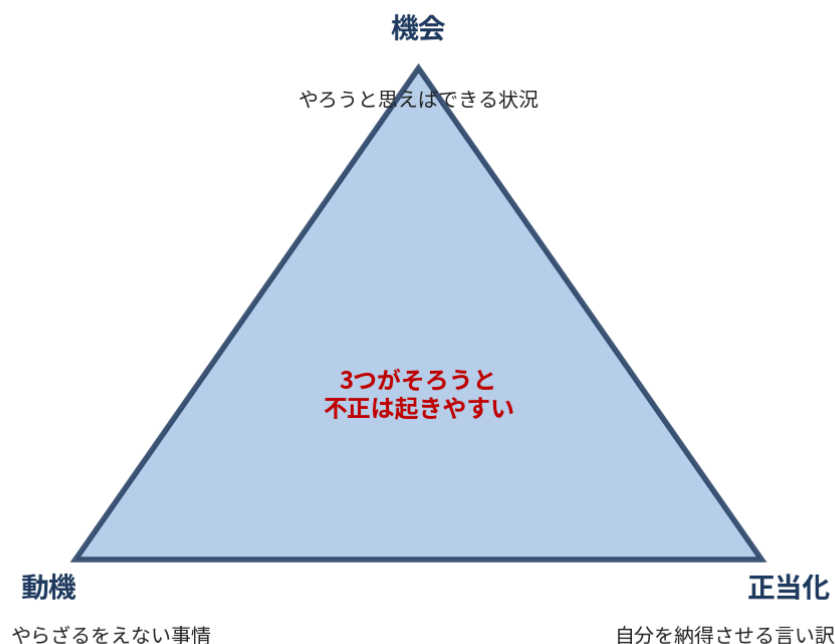


図 不正のトライアングル

「自分だけは流されない」、そう思うのは、自然なことです。その気持ちは大切です。でも、少しだけ、具体的に想像してみてください。配属されたばかりの職場で、まわりの人みんなが、当たり前のように、あるルール違反をしている。あなたは新人で、その分野の知識もまだ浅く、その場にいる全員が、あなたと逆の行動をとっている。この状況で、たった一人、「それは、まずいのではないですか」と声を上げる。それが、どれほど難しいことか。だからこそ、責任感が強く、まじめで、和を大切にする「ふつうのいい人」ほど、「波風を立てたくない」「自分さえ我慢すれば」と考えて、少し

ずつ巻き込まれていきます。これは、人間という生き物の性質です。だから、こういう「重力」が働くのだと、配属前の今、知っておくことに、大きな意味があります。知ってさえいれば、その場に立ったとき、「あ、これが、あのとき習った重力だ」と、一歩引いて自分を見られるからです。その一歩の引きが、あなたを守ります。

事例から学ぶ：恐怖支配が、組織ぐるみの不正を生んだ（報道された事例）

ある大手企業では、達成できないと降格や遠方への転勤が待っているという強い圧力が、日常的にかけられていました。その恐怖の下で、現場の責任者が、部下に対して不正な処理を強いるようになり、それがいつしか「当たり前」になっていきました。後に設置された第三者委員会は、その原因として「**達成不可能な過大な目標**」「**ガバナンス（統治）の機能不全**」「**経営陣に逆らえず、忖度するゆがんだ企業風土**」を挙げています。

ここで大事なのは、不正に関わった多くの従業員は、最初から「悪人」だったわけではない、ということです。逆らえない空気の中で、少しずつ、巻き込まれていったのです。

→ **あなたにできる一歩**：「みんなやっている」「指示されたから」は、後になって、あなたを守ってはくれません（「指示だった」は、法的にも免責になりません）。もし「これはおかしいのでは」と感じたら、まず、その事実を記録する。そして、信頼できる人に相談する（具体的な動き方は、次の13-3で渡します）。気づけたあなたは、もう、流されかけていません。

なお、コーポレートガバナンス、コンプライアンス、内部統制の仕組みそのものは、第15章でくわしく扱います。

13-2 健全な職場・リーダーの「ものさし」—違和感に気づく力

前の章（第12章）では、ハラスメント・感情・言葉・誠実さ・公正を学びました。組織の健全さも、その延長にあります。この節では、それらを、**配属後に、あなた自身が使える一本の"ものさし"**に束ねます。

最初に、大切なことわりを。このものさしは、**「誰かを採点し、裁くため」のものではありません**。誰かを「あの人はダメだ」と決めつけるための道具ではないのです。これは、あくまで**あなた自身が、自分の身を守り、健全に働き続けるための**、心の中の基準点です。この一点を、どうか間違えないでください。

なぜ「ものさし」を先に渡すのか。もう一度言います。人は、基準を知らないと、目の前でおかしなことが起きていても、「社会人とは、こういうものなのかな」と受け入れてしまうからです。先に「健全とはこういうものだ」と知っていれば、ギャップに気づけます。

そして、もう一つ、とても大切なことわりがあります。このものさしは、「100点満点の、完璧な職場」を探すためのものではありません。完璧な職場も、完璧なリーダーも、この世には存在しないからです。誰にだって、機嫌の悪い日はあるし、言い方を間違える日もあります。一度や二度、ものさしから外れたからといって、「ここはダメな職場だ」と決めつける必要は、まったくありません。見てほしいのは、それが「たまたまの一回」なのか、それとも「くり返し・当たり前のよう」に「そう」になっているのか。そして何より、誰かに「それはおかしい」と指摘されたときに、直そうとするのか、それとも開き直って相手を黙らせようとするのか。その態度の違いこそが、健全な職場と、そうでない職場とを分ける、いちばん本質的な分かれ目です。完璧さではなく、誠実に直そうとする姿勢があるかどうか。そこを見てください。

知っておこう：健全な職場・リーダーの「ものさし」

- ・ **公正**：チャンスと評価が、好き嫌いではなく、基準と事実にもとづいて、公平に開かれている。
- ・ **言行一致**：言ったことを守る。そして、自分の誤りを、誤りだと認められる。
- ・ **責任**：仕事を任せても、結果の責任は、最後は自分が引き受ける（部下に押し付けない）。
- ・ **敬意**：立場や年齢に関係なく、敬意のある言葉づかいが保たれている。
- ・ **誠実**：事実をねじ曲げない、隠さない、ごまかさない。
- ・ **学びとルール**：教育や、コンプライアンス（ルールを守ること）を、軽んじず大切にする。
- ・ **公私の区別**：私的な好き嫌いや仲の良さで、人事や評価を動かさない。

これらが「ある」職場は、多少忙しくても、安心して力を出せます。逆に、これらが**続けて欠けている**と感じたら、それは、あなたが気にしすぎなのではなく、環境のほうに理由があるサインかもしれません。

◆ 考えてみよう：「違和感」は、大切な情報

もし配属先で、このものさしが繰り返し欠けていると感じたら。「気のせいかな」「自分が新人だから、まだ分かっていないだけかな」と、その感覚を打ち消してしまう前に、どうか、その違和感を、そっとメモしておいてください。

経験を重ねた多くの人が、振り返ってこう言います。「あのとき感じた"なんだか変だ"は、たいてい当たっていた」と。たとえ誰かに「考えすぎだよ」と言われても、あなたが感じたことは、消えません。**あなたの感覚を、信じていい。**その上で、次の節（13-3・13-4）で、具体的にどう動けばいいかを、一緒に確認していきましょう。

13-3 健全さを守り、自分も守る—染まらないための構え

健全な組織は、つくるだけでなく、守ることも大切です。もし、ものさしから大きく外れた状況に出会ったら、自分を保ちながら、健全さを取り戻す側に立ちましょう。大事なのは、たった二つ。**あなたが、その色に染まらないこと。そして、絶対に、一人で抱え込まないこと。**この二つを実現するための、具体的な構えを5つ、渡します。

知っておこう：健全さを守り、自分も守る5つの構え

1. **記録する**：いつ・どこで・誰が・何と言った（した）か・その場に誰がいたかを、事実だけ、淡々とメモする。感情的な言葉ではなく、事実を。日記アプリでも、手帳でも構いません。この記録は、後であなたを守る、いちばん確実な味方になります。
2. **抱えない・相談する**：信頼できる人に話す。先輩でも、同期でも、家族でも、社内外の窓口でもいい。一人で背負わないことが、何より大切です。
3. **加担しない「線」を、先に決めておく**：「これだけは、できない」という一線を、平時のうちに、自分の中に持っておく。実際に迫られてから考えるのは、とても難しいからです。判断に迷ったら、第10章・第21章で学んだ手順、「事実を確認 → 関係者を考える → 選択肢を出す → 原則に照らす → 後で堂々と説明できるか」、を思い出してください。
4. **逃げ道を確保する**：記録と、相談した履歴を残しておく。そして、これは強く言いたいのですが、**最後の手段としての異動や、転職も、自分を守るための、まっとうで正当な選択**です。「逃げ」ではありません。自分の心と体を壊してまで、しがみつくなすべき場所はありません。
5. **「指示」や「空気」は、盾にならないと知る**：「指示されたから」「みんなやっていたから」は、

後になって、あなたを守ってはくれません。法的にも、免責にはなりません。最後にあなたを守るの
は、あなた自身の判断と、残しておいた記録です。

ここで、当事者の声から生まれた、二つの大切な補足をします。

💡 **ポイント：「有能で、人気のある人」でも、加害することがある**

「あの人は仕事ができるから」「みんなに好かれているから」、まさかハラスメントなんて、そう思
いたくなります。しかし、現実には、有能で人望のある人が加害者であるケースは、少なくありませ
ん。そして、そういうとき、周りは「あの人がそんなことするはずない」と被害者を信じてくれず、
被害を受けた人が、かえって孤立してしまうことがあります。

だからこそ、**記録**が効きます。そして、**一人ではなく、複数人で**声を上げることが、状況を動かす力
になります。「自分だけがおかしいと感じているのかも」と思ったら、まず、信頼できる誰かに、そ
っと確かめてみてください。

💡 **ポイント：客先常駐・受託の現場では（板挟みに備える）**

受託の仕事では、お客さまの会社に常駐して働くことがよくあります。このとき、「自社の上司」と
「客先の担当者」の両方から、違う方向の圧力を受け、「いったい誰に相談すればいいんだ」と、板
挟みになることがあります。

基本の原則を覚えておいてください。**あなたの雇い主は、自社です。**だから、まずは**自社の上司や、
自社の窓口**に相談します。客先からの理不尽な言動は、本来、あなたの会社があなたを守るべき対象
（カスハラ）です。一人で板挟みを抱え込まず、「こういうことがあって困っている」と、自社に上
げてよいのです。それは、告げ口でも、弱音でもありません。

💡 **ポイント：そのまま使える「言い方」の例**

いざというとき、言葉は意外と出てこないものです。いくつか、あらかじめ"持って"おきましょう。

- ・ **確認したいとき**：「すみません、1分だけ確認させてください」
- ・ **無理な依頼を断るとき**：「いま〇〇を抱えていて、このままだと質が落ちてしまいます。優先
順位を一緒に決めていただけますか」
- ・ **相談を切り出すとき**：「少しご相談したいことがあって、お時間をいただけますか」
- ・ **おかしいと感じて社内窓口に伝えるとき**：「こういうことがあって、困っています。記録も残
してあります」

完璧な言い方でなくて、大丈夫。**"言い出す"こと自体が**、あなたを守る第一歩です。

最後に、いちばん大切なことを。ここまで紹介した「構え」は、あなたを、疑い深い人や、いつも
身構えて戦う人にするためのものでは、けっしてありません。むしろ、逆です。これらは、**安心し
て、のびのびと働くための**備えです。お守りを一つポケットに入れておくと、かえって落ち着いて出
かけられる、あれと同じです。現実には、あなたが会える職場や先輩の、ほとんどは、あなたを大切
に育てようとしてくれるはずです。その温かい毎日の中で、ごくまれに「あれ、これはおかしいぞ」
という場面に出くわしたとき、この備えさえあれば、あなたは、自分を見失わず、流されず、自分の
足で立っていられます。

13-4 困ったときに—相談先・証拠・周りの一歩

「もしも」のときのために、**相談先の地図**を持っておきましょう。すべてを覚える必要はありません。「こういう場所が、いくつもある」と知っておくだけで、いざというときの安心感が、まるで違います。

知っておこう：相談先マップ

〈社内〉 直属の上司／その上の上司／人事／ハラスメント相談窓口／産業医や保健師／（あれば）社内のカウンセリング窓口（EAP）。

〈社外〉 各都道府県の労働局にある「総合労働相談コーナー」（無料・予約不要で相談できます）／法テラス（法律相談）／労働組合や、個人で入れる社外のユニオン。

ポイントは、**一つのルートでうまくいかなくても、別のルートがある**ということ。相談は、一度きりでも、一か所きりでもありません。

「いきなり、外部の窓口相談するのは、ハードルが高い」と感じるかもしれません。それで、まったく構いません。まずは、信頼できる先輩や同期、あるいは家族に、「ちょっと、聞いてほしいことがあるんだけど」と話してみる。それだけで、立派な第一歩です。人に話すと、自分の頭の中が整理され、「これは、自分が我慢して済ませる話ではないんだ」と、はっきり気づけることがあります。相談とは、必ずしも、すぐに答えや解決をもらうためだけのものではありません。一人で抱えていた重さを、半分だけでも、誰かと分け合うこと。そのこと自体に、心を守る大きな意味があります。「こんなことで相談していいのかな」と迷うくらいが、ちょうどいいのです。

正直に、お伝えしておきます。

ポイント：相談しても、すぐに思い通りにならないことも、ある

残念ながら、勇気を出して相談しても、すぐには期待どおりに解決しないこともあります。これは、現実として知っておいてください。でも、だからといって、相談に意味がないわけでは、決してありません。**相談した記録、残しておいた事実の記録**、それらが、次の一歩で、確実にあなたを支えます。一か所がだめでも、社外の窓口は、いつでも、あなたのために開いています。あきらめなくて、大丈夫です。

■ あなたが「見た側」になったとき

ハラスメントは、当事者だけの問題ではありません。周りの人の小さな一歩が、職場の空気を、大きく変えます。

たとえば、誰かがきつく言われている場面で、「それ、ちょっと言い過ぎじゃないですか」と、やわらかく挟む一言。あるいは、つらそうな同僚に、「大丈夫？」とそっと声をかけること。これだけで、加害がエスカレートするのを止めたり、被害を受けた人を、孤立から救ったりできます。

ただし、**無理は禁物**です。その場で止めるのが怖ければ、それで構いません。後で、見たことを記録して、相談に回すだけでも、十分に立派な行動です。

そして、もう一つ、とても大切なこと。**被害を受けた人を、絶対に責めないこと**。「あなたにも、何か隙があったんじゃない？」「気にしすぎだよ」、こうした言葉は、たとえ善意でも、被害者をもう一度傷つける「二次被害」になります。

考えてみよう

もし、あなたの同期が、ある日ぽつりと「最近、ちょっとつらくて……」と打ち明けてくれたら。あ

あなたは、何と言ってあげたいですか。

立派なアドバイスは、ありません。たいていの場合、いちばん力になるのは、「話してくれて、ありがとう」という、たった一言です。同期や仲間どうしで、こうして支え合えること（ピアサポート）は、つらいとき、本当に大きな救いになります。あなたにとっても、あなたの周りの誰かにとっても。

事例から学ぶ：ある新入社員の過労自殺

ある会社に入った、ひとりの新入社員がいました。長時間労働と、職場での厳しい言動が重なる中で、心を病み、入社して間もない冬に、自ら命を絶しました。亡くなる前のひと月の残業は、月100時間を超えていたと言います。この出来事は労災（仕事が原因の災害）と認められ、会社の責任が問われました。長時間労働を見直す社会的な動きが大きく進む、ひとつのきっかけにもなりました。

胸が痛むのは、彼女がとても優秀で、責任感の強い人だったことです。まじめな人ほど、「自分がもっと頑張れば」と、一人で抱え込んでしまいます。だからこそ、はっきり伝えておきたいのです。

→ **あなたにできる一歩**：どんな仕事にも、あなたの心と体より大切なものは、ありません。「おかしい」と思うほど忙しいとき、つらくて仕方ないときは、休んでいい。逃げていい。相談していい。それは、負けでも、甘えでもなく、自分を守るための、いちばん大事な行動です。（働きすぎのサインや、相談・休み方は、第16章でくわしく扱います。）

なお、心や体の不調のサインと、そのケア・休み方・相談のしかたは、第16章でさらにくわしく扱います。

13-5 心理的安全性と、あなたがリーダーになるとき

重いテーマが続きました。最後は、ぐっと明るい方向を向いて、この章を締めくくりましょう。これまで見てきた「健全でない職場」の、ちょうど裏返しに、**良いチームの姿**です。

第6章で学んだ**心理的安全性**を、覚えているでしょうか。「このチームでは、質問しても、失敗を正直に言っても、反対意見を述べても、罰せられたり、馬鹿にされたりしない」、そう信じられる状態のことでした。これは、この章で扱ってきたハラスメントや恐怖支配の、まさに正反対にあるものです。

ポイント：心理的安全性は「ぬるい職場」ではない

ここはよく誤解されます。心理的安全性が高い、とは、「厳しいことを言わない、なれ合いの、ぬるい職場」という意味では、まったくありません。

むしろ逆です。**高い目標を持ちながら、同時に、安心して本音を言い合える**。それが心理的安全性の高いチームです。安心して「それ、おかしくないですか」と言えるからこそ、ミスや問題を早く見つけられ、結果として高い成果を出せる。研究でも、それが示されています（→第6章）。心理的安全性は、優しさであると同時に、チームの「強さ」そのもののなのです。

なぜ、この明るい話を、つらいテーマばかりが続いたこの章の、最後に置いたのか。それは、この章で本当に伝えたかったゴールが、「こわい職場から、自分の身を守る」ことだけでは、ないからです。本当のゴールは、その先にあります。それは、**あなた自身が、誰かにとって「安心できる存在」になること**。質問されたら、ばかにせず、ちゃんと答える。失敗を打ち明けられたら、責める前に、まず受け止める。後輩ができたなら、自分がされて嫌だったことを、しない。たったそれだけのこと

で、あなたは、職場の空気を、少しずつ、でも確実に、良いほうへ変えていきます。今、新人のあなたが感じている「こんな職場だったらいいのに」という、その素直な願い。それは数年後、あなた自身の手で、それを実現するための、いちばん確かな原動力になります。被害から身を守る知識は、いつか、良い場をつくる力に変わるのです。

もちろん、心理的安全性をいま用意する第一の責任は、組織と、人を率いる側にあります。新卒が安心して質問し、失敗を報告できる場をつくるのは、迎える側の仕事です。あなたが将来その担い手になるのと同時に、いまの組織にも、その場を整える責任があります。

そして、いつか、たぶん、あなたが思うよりずっと早く、あなたにも、後輩ができ、人を率いる立場になる日が来ます。

◆ 考えてみよう：未来のあなたへ

数年後、あなたが誰かを指導する立場になったとき。この章を読んで「こんなことを、されたら嫌だな」と感じたことを、自分が、していないでしょうか。

今日、あなたが感じた「嫌だな」「つらいな」という気持ちは、消してしまうものではありません。それは、**未来のあなたが、良いリーダーになるための、いちばん確かな教科書**です。されて嫌だったことを、しない。してほしかったことを、する。それだけで、あなたは、誰かにとっての「健全な職場」を、自分の手でつくる側に回れます。

13-6 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう（本を閉じて、思い出せるだけで大丈夫）

- ・ 人が「流されて」不正などに加担してしまう、3つの力（重力）を、思い出せるだけ。
- ・ 「健全な職場・リーダーのものさし」を、思い出せるだけ挙げてみましょう（公正・言行一致・責任・敬意・誠実…）。
- ・ もし「なんだか変だ」と感じたら、まずする2つは何でしたか（記録する／一人で抱えず相談する）。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた、1～5の点を、もう一度つけてみましょう。点が上がっていれば、それが、この章であなたが手に入れたものです。まだ自信が持てないところは、これからの仕事の中で、少しずつ身につけていけば、それで十分です。

第14章 情報を扱う責任（26日目・1日）

🎯 この章のゴール

この章では、情報を扱う責任、機密の扱いと、自分を守る自衛を学びます。

終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ①【考え方がわかる】情報は「機密度に応じて扱う」とわかる。
- ②【知っている】SNS・社外での発信に注意がいて知っている。
- ③【構えがわかる】「自分にも起こりうる」と考え、自衛する構えがわかる。

💡 まず自己診断—読む前に、いまの自分を知ろう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「情報の扱いを自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
- ② 「情報漏洩の事故は、自分にも起こりうると思いますか？」を、いまの考えで一言だけ書きます（正解は気にしません）。

読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

✦ よくある勘違いと、その反証

【勘違い】「自分のような新人は、情報事故の当事者にはならない」→

【反証】実際の事故の多くは、悪意のない普通の人々が、普通の業務やSNSで起こしています（14-5の実例）。

「自分は大丈夫」という油断こそ、最大のリスク要因です。

14-1 情報へのアクセスと責任—組織人として知っておくべきこと

エンジニアは「情報に最もアクセスしやすい職種」の一つです。このアクセス権限は業務遂行のためのものであり、その権限を適切に使うことが、組織と顧客への責任を果たすことになります。

- ・ **「アクセスできる」と「アクセスしていい」は別のことを意味します**
- ・ 業務上必要な範囲でアクセスすることが、信頼される専門家としての姿勢です

情報へのアクセスしやすさは、エンジニアの“力”であると同時に“責任”です。鍵をたくさん預かっている人ほど、その鍵をどう使うかが問われます。便利さの裏側に責任があることを、最初に意識しておきましょう。

- ・ テスト時は本番データではなく、マスキング済みのデータを使う習慣が組織を守ります

「テストに本番データを使わない」のも、同じ理由です。本物の個人データをテスト環境に置くと、そこから漏れる危険があります。手間でも、ダミーやマスキング済みのデータを使う。面倒な一手間が、大事故を防ぎます。

そして、もし情報のミス、誤送信、紛失、見られてはいけないものを見られた、が起きたら、隠さず、すぐ報告してください。情報事故は、初動の速さが被害の大きさを決めます。「怒られたくな

い」で黙ると、傷口が広がります。早く言える人が、結果として自分も会社も守ります。

📁 事例から学ぶ：ある大手通信会社の子会社の情報漏洩事件（2023年）から学べること

その子会社の元派遣社員が、システムの管理者アカウントを悪用してサーバにアクセスし、数百万件規模の個人情報をUSBメモリーで不正に持ち出した事件です。

持ち出しは長期間に及びました。

親会社の社長が引責辞任し、監督官庁から行政指導を受けました。

この教訓から学べること：「アクセスできる立場」を正しく使うことが、自分と組織と顧客を守ります。

📁 事例から学ぶ：ある大手教育サービス会社の個人情報漏洩事件（地方裁判所の判決：2023年2月）から学べること

2014年に発覚した、業務委託先の技術者が顧客情報（数千万件規模）を不正に持ち出し、名簿業者に売却した事件です。外部からの攻撃ではなく、内部からの持ち出しでした。

漏えいから約9年後の2023年2月、地方裁判所は、多数の原告に対し、一人あたり数千円・総額で1,000万円を超える賠償を命じました（本件をめぐる訴訟は複数あります）。

→学べること：情報事故は、外部からの攻撃だけでなく、アクセスできる人の手からも起こります。だからこそ「最小権限」と持ち出しの管理が、組織と顧客、そして自分を守ります。

■ 最小権限—「必要な分だけ」の原則

情報セキュリティの基本原則の一つに、「最小権限」という考え方があります。仕事に必要な情報・システムにだけアクセスでき、それ以外は持たない・触れない、という考え方です。権限が大きいほど便利な反面、事故やミスの際の被害も大きくなります。「自分はどこまでアクセスしてよいのか」を正しく理解し、必要のない情報には近づかない、それが、自分を守る第一歩です。

新人のうちは、「これ、見ていいんだっけ?」と迷う場面が必ずあります。迷ったら、見る前に確認する。勝手に「たぶん大丈夫」で進めない。これだけで、多くの事故を防げます。

✦ 考えてみよう：その情報、いま必要?

仕事で何かの情報にアクセスするとき、「これは今の作業に本当に必要か」と、一度だけ自分に聞いてみましょう。

“なんとなく”“つい”で見るクセをなくすだけで、あなたは情報事故からぐっと遠ざかります。

14-2 情報セキュリティの基本

2023年度の個人情報漏洩・紛失事故件数は13,279件（個人情報保護委員会）で過去最多を記録しました。**情報セキュリティを日常的に意識することが、組織と自分を守る力になります。**

件数が増えているのは、攻撃が増えたからだけではなく、私たちが扱うデータが増え、つながる仕組みが増えた分、漏れる経路も増えているのです。便利さとリスクは、いつも背中合わせ。だからこそ、一人ひとりの基本動作が、これまで以上に大切になっています。

情報の種類	例	扱い方の原則
機密情報	営業秘密・開発中の製品情報・個人情報	社外持ち出し禁止・暗号化・アクセス権限の制限
社外秘	社内の経営情報・人事情報	社外への開示禁止・適切な管理
社内公開	一般的な業務情報	社内では共有できますが社外には出しません
公開情報	プレスリリース・公開仕様書	制限なし

情報には“重み”があります。誰でも見てよい情報（公開）から、社外秘、そして個人情報や営業秘密のように厳重に守るべきものまで。同じ書類でも、中身によって扱いが変わります。「この情報はどのレベルか」を意識する、それが、適切に扱う第一歩です。

迷ったら、上位（厳しいほう）に倒すのが原則です。「これは社外秘かな、公開していい情報かな」と迷ったら、まず社外秘として扱っておけば安全です。ゆるく扱って漏らすより、厳しく扱いすぎるほうが、ずっとましです。

情報を守る基本は、特別な知識より「日々の小さな習慣」です。たとえばパスワード。同じものを使い回していると、どこか一つのサービスから漏れたとき、芋づる式に全部が破られます。だから、パスワードは使い回さず、大事なサービスには二段階の確認（MFA）をかけておく。家のドアと金庫で別の鍵を使うのと同じ発想です。

もう一つ知っておきたいのが、「あなたを狙ってくる」攻撃です。本物そっくりのメールでパスワードを入力させる（フィッシング）、業務を装って添付ファイルを開かせる（標的型メール）、手口は年々巧妙です。「少しでも怪しい」と感じたら、リンクや添付を開く前に、送信元を確かめ、社内の窓口に相談する。慌てさせて判断を奪うのが、攻撃の常套手段です。

💡 ポイント：守るのは「会社のため」だけじゃない

情報を守ることは、会社のためだけではありません。漏れた情報の先には、迷惑をこうむる「お客様」や「同僚」がいます。

そして、事故を起こせば、あなた自身のキャリアも傷つきます。情報を守る習慣は、まわりの人と、未来の自分を守る習慣でもあります。

✦ 考えてみよう：あなたの“鍵”は大丈夫？

あなたが今使っているサービスで、同じパスワードを使い回しているものはありますか。

大事なものから一つずつ、別のパスワードや二段階確認に変えてみましょう。今日できる、いちばん簡単な自衛です。

14-3 SNS・情報発信のルール

SNSは個人の発言が瞬時に広がるメディアです。

社会人になっても、あなたのSNSアカウントは「一個人」と同時に「会社の人間」として見られます。

■ 社会人のSNS利用で守るべき基本原則

- ・ 業務上知り得た情報を投稿しない：顧客名・プロジェクト内容・社内の出来事は非公開

- ・ 会社・同僚・顧客への誹謗中傷をしない：「愚痴」として書いたことが証拠になる

とくに気をつけたいのが「愚痴」です。つらいとき、つい吐き出したくなる気持ちは自然ですが、SNSに書いた愚痴は、消えずに残り、誰かに見られます。愚痴は、SNSではなく、信頼できる人に直接こぼす。それが、自分を守る賢いやり方です。

- ・ 「会社の見解」と「個人の意見」を明確に分ける：会社を代表するような発言は慎む
- ・ 写真・動画に注意する：職場の写真に機密情報・個人情報映り込んでいないか確認する
- ・ 過去の投稿も審査される：就職活動前後の投稿は採用・評価に影響することがある

「鍵付きアカウントだから大丈夫」は誤りです。スクリーンショットで拡散されるリスクは常にあります。

投稿前に「上司・顧客の目に触れて恥ずかしくないか」を自問する習慣を持ちましょう。

迷ったら、投稿しない、これがいちばん簡単で、いちばん確実なルールです。発信は、消えないリンクのようなもの。出してから後悔するより、出す前に一呼吸おくほうが、ずっと楽です。

「プライベートのSNSなのに、なぜ会社の人間として見られるのか」と感じるかもしれません。理由はシンプルで、見る側には“その人のプライベートと仕事”の区別がつかないからです。プロフィールに会社名や仕事の話があれば、その投稿は「〇〇社の人の発言」として受け取られます。理不尽に思えても、それが現実。だからこそ、公私のあいだに、自分で線を引いておく必要があります。

事例から学ぶ：何気ない一枚が…

ある人が、職場での自撮りをSNSに投稿。背景のホワイトボードに、顧客名と進行中の案件が写り込んでいました。本人に悪気はなく、数分で消しましたが、その前にスクリーンショットが拡散。小さな油断が、大きな問題になりました。

→ あなたにできる一歩：職場で写真を撮る・上げるときは、背景に何が写っているかを必ず確認する。

14-4 SNS・インターネット上の情報発信と法的リスク

SNSは個人の発信力を劇的に高める一方で、一度の不適切な投稿が個人のキャリアと会社の信頼を大きく傷つけるリスクがあります。**「会社員」としての発信は、個人のものであっても会社に影響することを常に意識しましょう。**

■ 業務中・業務に関連する発信のリスク

- ・ 守秘義務違反になりえる発信：「今日の客先、めちゃくちゃ大変だった。〇〇業界の企業がこういう問題を抱えているとは」、顧客が特定できる情報は守秘義務違反です
- ・ 技術情報の無意識な漏洩：「今自社で〇〇というシステムを作っている」「〇〇の脆弱性を発見した」、未公表の技術情報・セキュリティ情報は絶対に外部で言及しません
- ・ 写真からのリスク：会社のPC画面・ホワイトボード・資料が写り込んだ写真をSNSに投稿することは、機密情報漏洩につながります
- ・ 「会社の意見ではありません」の注意：このフレーズがあっても、会社名・肩書きを開示した状態での発言は、会社の評判に影響します

■ 退職後も続く義務—転職・退職時の注意点

- ・ 守秘義務は退職後も継続：在職中に知った営業秘密・顧客情報・技術情報は、退職後も守秘義務が継続します。不正競争防止法違反で損害賠償を請求されるリスクがあります

- ・ 技術ブログでの注意：前職での「失敗談」「学び」を発信する場合でも、会社名・プロジェクト名が特定できる情報は避けます
- ・ OSSでの注意：前職の業務で作ったコードをOSS化することは、職務発明制度との関係で問題になりえます。必ず事前に確認します

前職の経験を発信すること自体は、悪いことではありません。むしろ学びの共有には価値があります。ただ、「会社名・案件名・人が特定できる情報」は外す。一般的な学びとして書く分には問題なく、特定の情報を出すと違反になりうる、この線引きを覚えておきましょう。

- ・ 転職先への情報持ち込み：前職の顧客リスト・設計書・ソースコードを転職先に持ち込むことは、不正競争防止法違反・情報財産侵害として刑事事件になりえます

「前の会社で作ったものだから、自分のもの」と思いがちですが、業務で作った成果物の多くは、会社に権利があります。退職時に、顧客リストや設計書、ソースコードを持ち出すのは、たとえ悪用するつもりがなくても、重大な違反になりえます。きれいに去ることは、次の職場での信頼にもつながります。

■ 個人SNSアカウントの管理

- ・ 鍵垢・プライベート設定の限界：「友人限定」の投稿でも、友人がスクリーンショットを撮ることで外部に漏れます。「全世界に公開される」前提で投稿内容を判断します（第12章で学んだ内容との接続）
- ・ 過去の投稿のリスク：就職活動・採用時に過去のSNS投稿がチェックされることがあります。不適切な過去の投稿は定期的に見直すことを習慣にしましょう

「もう消したから大丈夫」と思っても、誰かが保存していれば残ります。だから、消すことより、出す前に止まることが大事。投稿ボタンの前の3秒が、いちばん確実な対策です。

◆ 考えてみよう：その投稿、3年後に見られても平気？

SNSやネットへの書き込みは、消したつもりでも残ることがあります（デジタルタトゥー）。

「3年後、就職や昇進のときに、この投稿を見られても平気か」を、投稿前に一度想像してみましょう。

14-5 情報を「漏らさない」ために—自分とキャリアを守る実践

情報セキュリティやコンプライアンスについて、会社が十分に教えてくれるとは限りません。しかし、**会社の教育不足は、あなたの免罪符にはなりません。**情報を漏らして法律で裁かれるのも、SNSで身元が特定されて転職市場で苦労するのも、最終的にはあなた自身だからです。守るための柱は、突き詰めると2つだけです。

- ・ 情報セキュリティを学んで、自分で守る

情報を守ることは、難しい技術の話だと身構えなくて大丈夫です。本質は「相手の大切なものを、自分の大切なものと同じように扱う」という、人としての誠実さ。お客様や同僚の情報を、自分の家族の情報だと思って扱う、その気持ちがあれば、自然と慎重になれます。

- ・ 情報セキュリティを軽んじる先輩や同僚からは、距離をとる

■ 軽く扱くと、何を失うのか

業務で知った情報を不適切に扱うと、刑事罰の対象になりえます。代表的なものに、不正競争防止法（営業秘密侵害罪）は最大10年の拘禁刑または2,000万円の罰金、金融商品取引法のインサイダー取引は最大5年の拘禁刑または500万円の罰金（利益は没収）、個人情報保護法の不正提供・盗用

は最大1年の拘禁刑または50万円の罰金、があります（前科がつきます）。これに民事の損害賠償が加わり、故意による不法行為に基づく賠償は自己破産しても消えないことがあります。さらに懲戒解雇（退職金なし）と、ネット上に残る名前と顔（デジタルタトゥー）。前科・賠償・懲戒解雇・デジタルタトゥーは「自分には起きない最悪の話」ではなく、実際に新人が当事者になってしまう例も少なくありません。

📖 事例から学ぶ：情報漏洩は「悪人」ではなく「普通の人の一瞬」から

【SNS投稿による漏えい（近年）】入社直後の新人が、機密保持誓約書や社員番号・所属が写った入社書類を、軽い気持ちでSNSのストーリーズ（24時間で消える機能）に投稿。フォロワーがスクリーンショットで保存して別のSNSへ転載し、数時間で数百万回も閲覧される“炎上”になりました。本人の名前や顔まで特定され、消したくても情報は残り続けました。こうしたSNSがらみの新人の漏えいは、近年たびたび問題になっています。

【何気ない一枚から（近年）】別の若手が、職場で撮った写真や短い動画をSNSに上げたところ、背景のホワイトボードに顧客の氏名や業務目標が写り込み、広く拡散。「鍵アカウントだから」「すぐ消えるから」は通用しません。

【ある大手教育サービス会社（2014年。14-1で触れた事件の刑事の側面）】委託先（グループ会社）で働いていた派遣のシステムエンジニアが、顧客の個人情報（数千万件規模）を持ち出し、名簿業者に売却した事件です。本人は逮捕・起訴され、控訴審で実刑判決（懲役・罰金）が言い渡され、確定しました。会社にも多額の補償と信用失墜が及びました。たった一人の出来心が、本人の人生と巨大企業を同時に傷つけた事例です。

【ある自治体（全住民が数十万人規模・2022年）】給付金事務の委託先社員が、自治体の個人データを無断でUSBメモリにコピー。作業後の飲み会で泥酔し、路上で寝込んで、全住民分のデータが入ったかばんごと紛失しました（約3日後に発見、不正利用は確認されず）。自治体は委託元企業に多額の損害賠償を請求。「ルール無視の持ち出し」と「作業後の一杯」という気の緩みが、全住民規模の事故として大きく報道されました。

これらに共通するのは、派手な悪意ではないということです。入社して嬉しかった気持ち、通知が来た瞬間の机の上、作業後の一杯、ほんの出来心、そんな普通のシーンが、会社と顧客、そして自分の人生を一発で変える地雷になりました。SNSなら「鍵アカウントだから大丈夫」「匂わせ程度なら」も成り立ちません。スクショは永遠に残り、断片情報も組み合わせれば特定されます。投稿前に画像を拡大して背景を確認し、業務時間中に職場や客先で自撮りをしないこと。そして、データの持ち出しは決められた方法だけで行うこと、これが自分を守る最低ラインです。

■ 飲み会・電車・カフェでの会話

顧客名・案件内容・金額・トラブル・人物名は、社外では一切口にしない。これだけがルールです。周囲に誰がいるかはわかりません（同業者、顧客、その株主、SNSに書く人）。地方都市は世間が狭いぶん、リスクはさらに高くなります。**先輩が話していても、あなたは話さない。**なお、事実に基づかない他人の悪口・噂話は、名誉毀損になりえます（第12章12-4参照。事実かどうかは関係なく、伝播性があれば成立しうる）。噂話が始まったら話題を変えるか席を立ち、自分が新しい伝達者にならないことが、自分を守ります。

電車やカフェ、エレベーターの中も同じです。「ここなら大丈夫」と思った場所に、たまたま関係者がいることは珍しくありません。とくに資料を広げての打ち合わせや、スピーカーホンでの通話は要注意。場所を選ぶことも、情報を守る技術のうちです。

■ 「うちは非上場だから関係ない」は危険な誤解—インサイダー取引

インサイダー取引は、上場企業の未公表の重要情報を業務上知った人が、その情報で株を売買する犯罪です。盲点になるのが、「会社関係者」には上場会社の役職員だけでなく、子会社や、上場会社と契約している取引先・その従業員まで含まれること、そして会社関係者から情報を受けた人（第一次情報受領者）も規制対象になることです。つまり、自社が非上場でも、上場企業A社のシステムを開発・保守しているだけで、あなたはA社についての「会社関係者」になりえます。取引先の打ち合わせで上場企業の未公表情報を耳にすれば、その株の売買が規制されます。立ち聞きや飲み会で聞いた場合も対象になりえます。業務で関わる企業（およびその親会社・子会社・主要取引先）の株は、業務に関わる期間は触らない。これが自衛です。

知っておこう—自分を守る情報セキュリティの基本

- ・パスワードは使い回さない／パスワードマネージャを使う／多要素認証（MFA）は必ずON
- ・メールは送信前に宛先を必ず確認。「全員返信」と「差出人だけ返信」の取り違えに注意
- ・Excelの「非表示」セルやシートにはデータが残る。顧客情報を含むファイルは暗号化＋パスワードを別経路で
- ・業務データを私物デバイス・個人クラウド・個人のメッセージアプリに入れない（シャドーIT）
- ・「至急」「添付確認」系メールは疑う。リンクはURLを確認してからクリック
- ・紙やノートPCを電車・カフェに置き忘れない。退社時はクリアデスク
- ・最重要：誤送信や事故に気づいたら、隠さず必ず即上司に報告する。隠蔽は、何倍も重い問題になります

最後に、いちばん難しく、いちばん効くこと。情報セキュリティを軽んじる先輩・職場からは距離をとってください。事故が起きたとき、近くにいた人ほど巻き添えを食らいます。「先輩に言われてやりました」は、刑事責任の場面では通用しません。**周りの大人が正しいとは限らない**、これは、新卒が最初に身につけてほしい感覚です。明らかに法令違反レベルでマズいと感じたら、公益通報者保護法に守られた内部通報（社内窓口がなくても労基署など行政機関への通報が可能）という選択肢もあります。

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・自分のSNS投稿に「会社や顧客の情報が混じっていないか」を一度見直す。

この章でいちばん覚えておいてほしいのは、「情報事故は、悪人ではなく“普通の人の一瞬”から起こる」ということです。嬉しくてつい投稿した、急いでいて宛先を間違えた、よかれと思って家族に話した、どれも、特別に悪い人の話ではありません。

ポイント：AIの時代こそ、情報の扱いは重くなる

AIに何かを相談するとき、社外秘の情報や顧客データを、そのまま入力していませんか。便利なツールほど、情報が“どこに残る”リスクがあります。

何を入力してよくて、何はだめか、その線引きを持っておくことが、これからの情報リテラシーです（会社のルールを必ず確認しましょう）。

◆ 考えてみよう：いちばん守りたい情報は？

あなたが将来扱う情報の中で、「これだけは絶対に漏らしてはいけない」と思うものを、一つ想像してみましょう。

それを守るために、今日からできる小さな習慣は何でしょうか。

14-6 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ 情報は、何に応じて扱いを変えるか。
- ・ SNS・社外発信で気をつけることは。
- ・ 「自分にも起こりうる」とは、どういうことか。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、これからの学びや仕事の中で、少しずつ埋めていきましょう。

第15章 会社と社会を守るルール（27日目・1日）

🎯 この章のゴール

この章では、会社と社会を守るルール、ガバナンスとコンプライアンスを学びます。
終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ① **【違いがわかる】** ガバナンスとコンプライアンスの違いがわかる。
- ② **【知っている】** 会社のルール（情報管理など）が自分にも関わると知っている。
- ③ **【知っている】** 触れてはいけない行為があると知っている。

💡 まず自己診断—読む前に、いまの自分を知ろう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「コンプライアンスを自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
 - ② 「コンプライアンスは、会社にとって「コスト」だと思いますか？」を、いまの考えで一言だけ書きます（正解は気にしません）。
- 読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

✦ よくある勘違いと、その反証

【勘違い】「コンプラ＝面倒な“守りのコスト”」→

【反証】コンプライアンスは“信頼の前提”です。

一度の違反で、長年積み上げた信用と事業が吹き飛びます。

守ることは縛りではなく、会社とあなた自身を守る投資。

「バレなければいい」ではなく「なぜそのルールがあるか」で考えられる人が信頼されます。

15-1 コーポレートガバナンスとコンプライアンス

企業が社会から信頼を得て持続的に成長するためには、適切なガバナンス（統治）とコンプライアンス（遵法）が不可欠です。これらを理解することで、組織の健全性に貢献できます。**ガバナンスとコンプライアンスは、優秀な人ほど軽視しません。**一度の違反で、積み上げた技術も信頼も失われるからです。

■ コーポレートガバナンス

- ・ 内部統制、不正・ミスが起きにくい仕組みをつくること。健全な組織を守る基盤です
- ・ 透明性、適切な情報開示が、長期的な信頼と企業価値を高めます

内部統制と聞くと難しそうですが、身近なものです。経費を申請したら上司の承認が要る、システムの本番反映は別の人がチェックする、重要な操作は記録が残る、これらは全部、内部統制です。

「一人の判断やミスが、そのまま事故にならないように」という工夫であり、承認やチェックを“面倒な手続き”ではなく“みんなを守る仕組み”と捉えれば、向き合い方が変わります。

ガバナンスは、株主・顧客・社員・取引先・社会といった、会社に関わる人たち（ステークホルダー）

一) への責任を果たすための仕組みでもあります。誰か一人の利益のためでなく、関わる全員にとって健全であること、それが、長く続く会社の条件です。

事例から学ぶ：ある精密機器メーカーの粉飾決算事件（2011年）から学べること

20年以上にわたる損失隠しが発覚し、株価が急落した事例です。この教訓から学べること：内部統制が機能している組織は、長期的に信頼と価値を維持できます。

■ コンプライアンス

- ・ 狭義：法律・規則を守ること
- ・ 広義：法律だけでなく、社会的規範・倫理・顧客の期待に応えること
- ・ **コンプライアンスを守ることが、組織の長期的な信頼と競争力を高めます**

逆に、たった一度のルール違反が、長年かけて築いた信頼を一瞬で崩すこともあります。ニュースになる不祥事の多くは、特別に悪い人ではなく、「これくらいなら」「みんなやっている」という小さな緩みから始まっています。だからこそ、新卒のうちから「小さなルールこそ守る」感覚を持っておくことが、将来の自分を守ります。

■ ガバナンスを支える仕組み

ガバナンス（統治）は、経営者が暴走したり、組織が道を外したりしないように“経営を律する仕組み”です。重要な意思決定をチェックする取締役会、業務を監査する監査役や内部監査、経営を監視する株主、こうした存在が、その役割を担います。

その健全さを現場で支えるのが内部統制です。一人で全部を決めさせない（職務分掌）、お金の動きは二人で確認する（ダブルチェック）、記録を残す、こうした地味な仕組みが、不正やミスの起きにくい組織をつくります。新卒のあなたが日々守る手順やチェックも、この内部統制の一部です。

コンプライアンスも「法律さえ守ればよい」ではありません。法律に書いていなくても、社会の常識や顧客の期待を裏切らないこと、広い意味での誠実さが含まれます。「法律的にはセーフでも、お客さんが知ったらどう思うか」を考えられることが、本当のコンプライアンスです。

✦ 考えてみよう：「バレなければいい」？

もし、ルール違反が「誰にもバレなければ問題ない」としたら、何が起きるでしょうか。

信頼は、見ていないところでの誠実さで決まります。あなたが「見られていなくても守る」人だと、まわりは安心して仕事を任せられます。

15-2 ISMS・QMS—組織を守る「マネジメントシステム」を理解する

「ISMS」「QMS」という言葉を入社後に耳にすることがあります。これらは国際規格にもとづく組織の「マネジメントシステム（管理の仕組み）」であり、会社が社会・顧客に対して品質や情報セキュリティを保証するための土台です。認証を取得している会社では、社員一人ひとりの日常業務がこの仕組みの中で動いています。

■ マネジメントシステムとは何か

マネジメントシステムとは、「ある目的（品質・情報セキュリティ等）を継続的に達成・維持・改善するための、組織全体の仕組み」のことです。個人の努力や属人的なスキルに頼るのではなく、仕

組みとして品質や安全を保証する考え方です。

マネジメントシステム	規格	目的	社員への影響
QMS（品質マネジメントシステム）	ISO 9001	製品・サービスの品質を継続的に保証し、顧客満足を高める	品質手順・チェックリスト・是正処置の義務が生まれます
ISMS（情報セキュリティマネジメントシステム）	ISO/IEC 27001	情報資産を守り、情報漏洩・改ざん・滅失のリスクを管理する	情報の取り扱いルール・アクセス管理・インシデント報告の義務が生まれます
PIMS（個人情報保護マネジメントシステム）	JIS Q 15001（プライバシーマーク）	個人情報を適切に取り扱い、プライバシーを保護する	個人情報の取り扱い手順・廃棄ルールの遵守が求められます

■ QMS（品質マネジメントシステム）—ISO 9001

QMSとは、顧客が求める品質を「仕組みとして」安定的に提供し、継続的に改善していくための組織全体の管理体系です。ISO 9001は世界で100万社以上が認証を取得している国際規格です。

QMSの核心にある考え方：

- ・ PDCA（Plan-Do-Check-Act）、計画→実行→確認→改善のサイクルを回し続けること
- ・ プロセスアプローチ、仕事を「プロセス（手順の連鎖）」として設計し、各プロセスの品質を管理すること
- ・ リスクにもとづく思考、問題が起きてから対応するのではなく、起きる前にリスクを特定して対処すること
- ・ 継続的改善、現状に満足せず、常によりよい状態を目指すこと（第5章で学んだ「カイゼン」そのものです）

QMSの核にあるPDCA（計画→実行→確認→改善）も、特別なものではありません。やってみて、結果を確かめ、次に活かす、この当たり前のサイクルを、組織ぐるみで止めずに回し続けることが、品質を支えます。あなたの日々の小さな改善提案も、その一部です。

社員として求められること：

- ・ 手順書・チェックリスト・作業標準は「面倒なもの」ではなく、「品質を保証するための組織の約束」です
- ・ 問題が発生したとき、「是正処置（再発防止策）」を記録・報告する義務があります
- ・ 手順から逸脱するとき、勝手に判断せず、上長・品質担当者に確認します
- ・ 改善提案を出す文化、現場で気づいた改善点を提案することが、QMSを活かす最も大切な行動です

■ ISMS（情報セキュリティマネジメントシステム）—ISO/IEC 27001

ISMSとは、組織が保有する情報資産（顧客データ・システム・設計情報・営業秘密等）を守るための管理体系です。「守るべき情報を特定し、リスクを評価し、対策を講じ、継続的に改善する」という仕組みです。

ISMSが守る3つの特性（情報セキュリティの三要素）：

特性	意味	崩れると何が起きるか
機密性 (Confidentiality)	許可された人だけが情報にアクセスできること	顧客情報の漏洩・営業秘密の流出
完全性 (Integrity)	情報が正確で改ざんされていないこと	データの不正改ざん・システムの誤動作
可用性 (Availability)	必要なときに情報・システムを使えること	システム障害・ランサムウェアによる業務停止

社員として求められること：

- ・ 情報分類、社内の情報が「機密・社外秘・社内公開・公開」のどこに属するかを理解し、それぞれの取り扱いルールを守ります（第14章で詳述）
- ・ アクセス管理、自分に付与されたアクセス権限の範囲を理解し、業務上必要な情報のみにアクセスします
- ・ インシデント報告、情報の紛失・誤送信・不審アクセスに気づいたとき、速やかに報告します（小さなことでも隠さない）

情報セキュリティの三要素は、身近に言い換えられます。機密性＝「見るべき人だけが見られる」、完全性＝「勝手に書き換えられていない」、可用性＝「必要なときに使える」。顧客名簿が外に漏れれば機密性、データが改ざんされれば完全性、システムが止まれば可用性が損なわれます。どれが欠けても、信頼は崩れます。

事例から学ぶ：手順を飛ばした“近道”が事故に

忙しさのあまり、決められた確認手順を飛ばして本番作業を進めた結果、設定ミスに気づけず障害が発生、という事故は、どの現場でも起こりえます。手順は「遅くするためのもの」ではなく「事故を防ぐためのもの」。急ぐときほど、手順が効きます。

→ あなたにできる一歩：急いでいるときこそ、チェックリストを声に出して確認する。

- ・ 定期的な教育・訓練への参加、ISMSでは全社員への定期的なセキュリティ教育が義務づけられています

「毎年同じような研修で面倒」と思うかもしれません。でも、攻撃の手口は年々巧妙になり、人は時間がたつと気がゆるみます。定期的な教育は、その“ゆるみ”をリセットするためのもの。受け身で受けるのではなく、「今の自分の弱点はどこか」を一つ持ち帰るつもりで参加すると、ぐっと役に立ちます。

- ・ パスワード管理・画面ロック・USBの扱いなど、日常業務に組み込まれたルールを守ることが、組織全体のISMSを支えます

■ なぜ社員一人ひとりに関わるのか

ISMS・QMSがあっても、社員が守らなければ機能しません。逆に言えば、社員一人ひとりの日常行動の積み重ねが、組織全体のISMS・QMSを支えています。

よくある誤解	実際はこうです
「ISMS・QMSは管理部門や品質担当の仕事」	認証を受けた組織のすべての社員が対象です。自分の業務に関わるルールを守ることが求められます

よくある誤解	実際はこうです
「手順書通りにやれば思考は不要」	手順書は最低限の保証。なぜその手順が必要かを理解した上で守ることが本質です
「認証を取ればそれで終わり」	定期的な審査があります。継続的に改善を続けることが認証維持の条件です
「問題が起きなければ何もしてなくていい」	問題が起きる前にリスクを把握・対処することがマネジメントシステムの核心です

ISMS・QMSへの理解と貢献は、入社1年目から求められます。「**規則を守る**」だけでなく、「**なぜその規則があるか**」を理解した上で行動できる社員が、組織の信頼と品質を本当に支えます。

💡 ポイント

ISMS・QMSは「制約」ではなく「プロとしての証明」です。顧客や社会から「この会社の製品・サービスは信頼できる」と認められるための、組織全体での約束です。その約束を支えているのは、現場で働く一人ひとりの日常の行動です。

⚖️ 知っておこう—法律の紐づけ

個人情報保護法・不正アクセス禁止法、ISMSで定める情報取り扱いルールは、これらの法律上の義務を満たすための実践でもあります。ルールを守るとは、法的義務を果たすことでもあります。

■ なぜ「認証」をとるのか

ISMSやQMSの認証（ISO/IEC 27001／ISO 9001）は、飾りではありません。「この会社は、情報や品質を“仕組みで”守っています」と第三者が保証してくれる“信頼の証明書”です。多くの取引では、この認証が取引の条件になっています。つまり認証は、会社が仕事を得るための土台でもあるのです。

📖 事例から学ぶ：隠さず報告したから、大事に至らなかった

ある社員が、宛先を間違えてメールを送りそうになり、すぐ上長とセキュリティ窓口に報告しました。会社は速やかに相手へ連絡し、影響を最小限に抑えられました。もし「怒られたくない」と黙っていたら、被害はずっと大きくなっていたはずです。

→ あなたにできる一歩：情報のミスは、小さくても“すぐ・正直に”報告する。それがISMSを生きた仕組みにします。

💡 ポイント：ルールは「自分を守る盾」でもある

手順書やチェックリストは、面倒に見えて、実は「その通りにやれば責められない」というあなたの盾です。

決められた通りにやって起きた問題は、個人の責任ではなく、仕組みの改善対象になります。だから、勝手な自己判断で外れないことが、自分を守ります。

15-3 インサイダー取引

インサイダー取引とは、会社の内部情報（まだ公開されていない重要事実）を利用して株式などの有価証券を売買する行為で、金融商品取引法で厳しく禁止されています。

なぜここまで厳しいのでしょうか。それは、内部情報を知る人だけが得をすると、株式市場の公平が崩れ、一般の投資家が安心して参加できなくなるからです。市場全体の信頼を守るためのルール、そう理解すると、「自分には関係ない」とは思えなくなります。だからこそ、未公開の情報に触れたら、それを“自分の胸にしまっておく”のが基本です。

■ なぜエンジニアに関係するか

ITエンジニアは、業務の性質上、上場企業の未公開情報に接触する機会があります。

- ・ 決算数字・業績予測のシステムを扱う
- ・ M&A・新製品発表前のシステム開発に関わる
- ・ 顧客企業の内部データを取り扱う業務委託

このような情報を知った状態で株取引をすると、知らなかったでは済まない法的責任が生じます。

たとえば、こんな“うっかり”があります。担当した上場企業の好決算をシステム作業で先に知り、つい家族に「あの会社、今度すごいよ」と話してしまう。家族がその株を買えば、あなたも処分の対象になりえます。悪気の有無は関係ありません。重要事実の例は、決算の大きな変動、合併・買収（M&A）、新製品・大型契約、業績予想の修正など。「これは公表前かもしれない」と感じたら、まず口を閉じ、相談する習慣を。

💡 ポイント：迷ったら“触らない・話さない・相談する”

インサイダーも、独禁も、情報も、「グレーかな」と感じたら、自分で判断せず、いったん止めて相談する。

これが、知らないうちに加害者になることを防ぐ、いちばん確実な方法です。

項目	内容
禁止行為	未公開の重要情報を知りながら株取引をすること
「重要事実」の例	決算の大幅な変化、M&A、新製品・サービスの発表、重大な事故・不祥事など
罰則	5年以下の拘禁刑または500万円以下の罰金（法人は5億円以下）
情報伝達も禁止	自分で取引しなくても、他人に情報を流して取引させることも違反

■ 「うっかり」も罪になる—インサイダーの怖さ

インサイダー取引は、「もうけてやろう」という悪意がなくても成立します。たとえば、仕事で知った未公開情報を、家族や友人に何気なく話し、その人が株を買えば、あなたも巻き込まれます。SNSへの書き込みも同じです。「自分は取引していないから大丈夫」とは限りません。

守り方はシンプルです。仕事で知った未公開の重要情報は、口外しない・SNSに書かない・それに関係する株は触らない。迷ったら、取引の前に必ず上長やコンプライアンス窓口相談する。

15-4 独占禁止法・カルテル

独占禁止法は、市場の公正な競争を守るための法律です。

「カルテル（価格協定）」「談合（入札調整）」「不当廉売」などを禁止しています。

■ エンジニアが意識すべき場面

- ・ 競合他社の関係者との会話：「最近うちは単価を～にしている」という情報交換は違反になりうる
- ・ 入札・見積もりへの関与：**他社と事前に価格や落札者を調整することは談合として厳禁**
- ・ 取引先への不当な圧力：「うちのサービス以外使わないよう強制する」行為も問題になりうる

エンジニアが直接的に関わる機会は少ないですが、業界勉強会・カンファレンスでの

他社エンジニアとの会話の中で不用意に競争情報を交換することにも注意が必要です。

違反した場合、個人・法人ともに課徴金や刑事罰の対象になります（カルテル・談合は個人に5年以下の拘禁刑、法人に最大5億円以下の罰金が科されうるほか、別途課徴金の対象になります）。

やってはいけない会話を、はっきり覚えておきましょう。「うちは次の入札、〇〇円で出すよ」「お互い、客は取り合わないようにしましょう」「値段、そろえない?」、これらは、雑談でも違反を疑われます。違反すれば、会社に巨額の課徴金、関わった個人にも刑事罰が及ぶことがあります。あいまいに話を合わせず、「その話ではできません」と席を外す勇気を持ってください。

知っておこう：受託の現場で関わる「下請法」

独占禁止法と関連して、立場の強い発注者が弱い受注者に無理を強いることを防ぐ法律（下請法）があります。受託の現場では、「一方的な値引き」「無理な納期」「タダでの追加作業の強要」などが問題になります。

自分が将来、発注する側になったときも、この視点を忘れないでください。

15-5 公益通報制度

公益通報者保護法は、会社の不正を通報した従業員を保護する法律です（2022年改正で強化）。

- ・ 対象となる通報：法令違反行為（横領・談合・食品偽装・個人情報漏洩 など）
- ・ 通報先：社内の通報窓口、外部の行政機関、または一定条件下で報道機関・弁護士
- ・ 保護の内容：通報を理由とした解雇・降格・減給・嫌がらせなどの不利益な扱いは禁止
- ・ 2022年改正のポイント：常時使用労働者301人以上の事業者に公益通報対応体制の整備と従事者の指定を義務化（300人以下は努力義務）。通報者情報の保護を強化

エンジニアとして、システムや業務の中で不正を発見することがあるかもしれません。

そのとき「見て見ぬふりをする」とは、問題を共有した責任を問われる可能性があります。

組織の不正に直面したとき、どこに相談すればよいかを事前に知っておくことが重要です。

通報には、大きな心理的ハードルがあります。「告げ口みたいで気が引ける」「自分が不利になるのでは」、当然の不安です。だからこそ、一人で抱えないこと。まずは信頼できる先輩や同期に話すだけでも、頭が整理されます。そして、社内窓口が頼れないときのために、社外の相談先（行政機関・弁護士・労働局）があると知っておく。声を上げることは、会社を裏切る行為ではなく、会社を守る行為です。

■ ルールは、誰のためにあるのか

この章で見てきたルールは、どれも「会社を縛るため」だけのものではありません。お客様の信頼を守り、働く人を守り、そして長い目で見れば、あなた自身を守るためにあります。AIがどれだけ進んでも、「信頼される組織か」を決めるのは、ルールを誠実に守る一人ひとりです。ここでも土台にあるのは“体系”、なぜそのルールがあるかを理解して動ける人が、これからますます求められます。

（第10章で学んだ「内部通報制度」も合わせて確認してください）

見て見ぬふりは、ときに「黙認」として、自分の責任を問われることもあります。とはいえ、いきなり大きく動く必要はありません。「おかしい」と感じたら、まず事実を記録して、信頼できる人に相談する。その小さな一歩が、自分と組織の両方を守ります。

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・ 研修で習ったルール（情報の扱いなど）のうち、「自分にも関わるもの」を1つ挙げてみる。

通報には順序があります。まずは社内の通報窓口へ。それが機能しない、あるいは握りつぶされそうなときは、行政機関や、一定の条件のもとで外部（弁護士・報道）へ、と段階があります。大切なのは、一人で抱え込まず、「正しいことを伝えた人は守られる仕組みがある」と知っておくことです（組織の不正への向き合い方は第13章でも扱いました）。

◆ 考えてみよう：あなたの仕事と、ルールのつながり

この章のルール（ガバナンス・ISMS/QMS・インサイダー・独禁・公益通報）のうち、あなたが将来いちばん関わりそうなものはどれでしょうか。

一つ選んで、「自分の仕事のどんな場面に関わるか」を想像してみましょう。

15-6 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ ガバナンスとコンプライアンスの違いを、ざっくり。
- ・ 会社のルールは、自分にどう関わるか。
- ・ 触れてはいけない行為を、1つ。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、これからの学びや仕事の中で、少しずつ埋めていきましょう。

第16章 働く権利・環境・社会への責任（28日目・1日）

🎯 この章のゴール

この章では、働く権利・環境・社会への責任など、自分を守るルールを学びます。

終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ① **【知っている】** 労働時間や残業の基本ルールがあると知っている。
- ② **【知っている】** 有給・育休などの権利があり「使ってよい」と知っている。
- ③ **【大切さがわかる】** 不調のサインに早く気づき相談する、が大切とわかる。

💡 まず自己診断—読む前に、いまの自分を知ろう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「働く権利と環境を自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
- ② 「有給や残業のルールは、誰のためにあると思いますか？」を、いまの考えで一言だけ書きます（正解は気にしません）。

読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

✦ よくある勘違いと、その反証

【勘違い】「権利は会社が与えてくれるもの。

主張すると角が立つ」→

【反証】労働者の権利は、法律であなたに保障されたものです（会社の温情ではありません）。

正しく知り、正しく使うことは、わがままではなくプロの自己管理。

知らないことだけが、唯一あなたを不利にします。

16-1 労働法の整理と補強

自分の権利を知ること、健康的に・長く・よりよく働けるようになります。

権利・制度	内容	知っておくべきポイント
有給休暇	会社に年5日取得させる義務（労働基準法39条7項）	会社が拒否できる場面は「時季変更権」の行使のみです
残業代	1.25～1.5倍の割増賃金	みなし残業制の会社でも、超過分は請求できます
育児休業	子が1歳（条件により2歳）まで取得できます	男性も取得できます。取得を理由とした不利益は違法です
育児休業給付	休業中、給与の概ね67%（6ヶ月まで）・50%（以降）。加えて2025年4月から「出生後休業支援給付」により、両親がともに一	雇用保険から支給されます

権利・制度	内容	知っておくべきポイント
	定期間取得する等の条件で実質手取り10割相当（最大28日）の上乗せがあります	
介護休業	対象家族1人につき93日間（3回まで分割可）	突然来る介護に備えて知っておくことが大切です

■ 多様性と無意識の偏見

- ・ アンコンシャスバイアスを定期的に見直す習慣が、公平な判断力を育てます
- ・ 異なる背景・スキルセット・視点を持つメンバーが集まるチームは、複雑な問題をより多角的に解決できます

16-2 文書管理と証跡—記録を残す意義

ここでは要点だけを押さえます。記録・証跡の実務は、第7章（文書）や第14章（情報）とも重なります。

- ・ **正確な記録を残す習慣が、後から問題が生じた際に組織を守ります**
- ・ 文書を誠実に保管することが、組織の信頼性を高めます

📖 事例から学ぶ：品質データの正確な記録から学べること

複数の製造大手が品質データの改ざんを行い、社会的信頼を大きく損なった事例があります。この教訓から学べること：品質データを正確に記録・管理することが、組織の信頼を守り、顧客・社会への責任を果たす基盤になります。

16-3 社会・環境への責任

ここでは概要だけを押さえます。エンジニアとしての社会・環境への責任は、第30章でもう一度ふれます。

- ・ SDGs、**自分の仕事がSDGsのどの目標と関係するかを意識することで、仕事の社会的意義が見えてきます**
- ・ 省エネなシステム設計・データセンターの電力消費を意識することが、エンジニアとして社会に貢献する行動になります
- ・ 地方企業として地域社会に貢献できる機会を意識することが、より豊かな仕事につながります

16-4 労働時間の法律—「何時間まで働かせてよいか」を理解する

労働基準法が定める労働時間のルールは、社員を守るための最低限の基準です。入社前からこの知識を持っていることで、自分の権利を守り、将来管理職になったときに部下の権利を守ることができます。

■ 法定労働時間と所定労働時間の違い

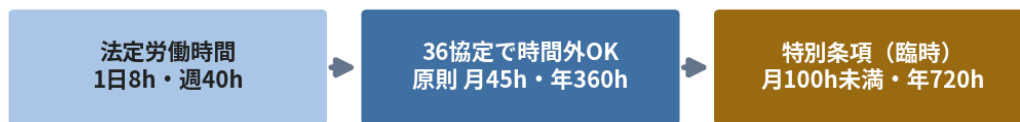
- ・ 法定労働時間：法律が定める上限、1日8時間・1週40時間（労働基準法32条）
- ・ 所定労働時間：各会社が就業規則で定める労働時間、法定労働時間の範囲内であれば会社が自由に設定できます（例：1日7.5時間・週37.5時間）
- ・ 重要：所定労働時間を超えた分は「残業」、法定労働時間を超えた分は「法定外残業」として1.25倍以上の割増賃金が必要です

■ 36協定—「残業させてよい」の許可証

「36（サブロク）協定」とは、労使間で締結する「時間外・休日労働に関する協定」のことです（労働基準法36条）。**この協定なしに法定労働時間を超えた残業をさせることは違法です。**

- ・ 36協定がある場合：月45時間・年360時間を上限として残業させることができます（原則）

働く時間のルール — 36協定と時間外労働の上限



※ 36協定なしの時間外労働は違法。特別条項にも上限（複数月平均80h等）があり、青天井ではない。

残業代は1円も損せず受け取る権利がある（割増賃金）。

図 36協定と時間外労働の上限

- ・ 特別条項：繁忙期など特別な事情がある場合は、月100時間未満・年720時間以内まで認められます（2019年働き方改革で上限が法律で義務化）
- ・ あなたができること：入社後に「会社の36協定の内容」を確認します。月何時間まで残業が認められているかを把握しておきましょう
- ・ 注意：「36協定を結んでいるから何時間でも残業させてよい」ではありません。協定の範囲内でも、健康被害が生じない配慮が必要です

■ 時間外労働と割増賃金の計算

残業代は「1円も損せず受け取る権利がある」ということを知っておきましょう。みなし残業（固定残業代）がある会社でも、超過分は追加で請求できます。

- ・ 通常の時間外労働（月45時間以内）：基本給の25%増し（1.25倍）
- ・ 月60時間超の時間外労働：基本給の50%増し（1.5倍）、中小企業には2023年4月から適用
- ・ 深夜労働（22時～5時）：基本給の25%増し、時間外労働と重複すれば50%増し
- ・ 休日労働（法定休日）：基本給の35%増し
- ・ みなし残業制（固定残業代）の仕組み：「月30時間分の残業代を含む月給〇〇万円」という形で、30時間を超えた残業が発生した場合は超過分を追加で支払う義務があります

16-5 有給休暇—「取れる権利」を正しく理解する

有給休暇は法律で保障された権利です。「上司に申し訳ない」「忙しいから取れない」という感覚は多くの場合、誤解や組織の文化的な問題から来ています。権利として正しく理解し、活用しましょう。

■ 有給休暇の発生と付与日数

- ・ 入社後6ヶ月で10日発生（所定労働日数の8割以上出勤していることが条件）
- ・ 1年6ヶ月：11日、2年6ヶ月：12日、3年6ヶ月：14日、4年6ヶ月：16日、5年6ヶ月：18日、6年6ヶ月以上：20日
- ・ 有効期限：2年間（付与日から2年で失効します）
- ・ 会社によっては入社と同時に全日数を付与するところもあります。就業規則で確認しましょう

■ 年5日取得義務—2019年から法律が変わりました

2019年4月の労基法改正で、会社は年10日以上の有給が付与された社員に「年5日は取得させる義務」を負うようになりました（社員の権利を守る強制ルールです）。

- ・ 会社が違反した場合：30万円以下の罰金（対象者1人につき）
- ・ 「社員が申請しないから取得させなくてよい」は違法、会社が積極的に有給取得を促す義務があります
- ・ 時季指定権：会社は「有給の時季を変更する権利（時季変更権）」を持ちますが、代替日を提示する必要があります。単純な「拒否」はできません

■ 有給休暇の申請の仕方

有給休暇の申請は労働者の権利であり、「理由」を会社に告知する義務はありません。ただし、チームへの影響を考えた上での申請が、職場の信頼関係を保ちます。

- ・ 前日申請：法律上は可能ですが、業務への影響を最小化するため余裕を持って申請する方が望ましいです
- ・ 申請理由：法律上の義務なし、「私用のため」で十分です。プライバシーに関わる理由を具体的に言う必要はありません
- ・ 半日・時間有給：就業規則で認められていれば取得可能です。会社によって異なります
- ・ 有給が取りにくい文化の会社：自分だけで抱え込まず、先輩や同期に相談したり、会社の就業規則・36協定を確認したりすることが第一歩です

16-6 育児・介護・その他の制度—ライフイベントに備える

育児・介護は「遠い未来の話」ではありません。入社後数年以内に結婚・出産を迎える人も多く、介護は突然訪れることもあります。制度を知っておくことで、必要なときに迷わず使えます。

■ 育児休業制度の全体像

- ・ 取得できる期間：子どもが1歳になるまで（保育所に入れない等の場合は最長2歳まで延長可能）
- ・ 男性も取得できます：「男性育休」は法律で保障されています。取得率向上のため、2022年に「産後パパ育休」制度が創設されました
- ・ 産後パパ育休（出生時育児休業）：子の出生後8週間以内に、最大4週間取得可能。2回まで分割取得できます

- ・ 2025年以降の動向：男性の育児休業取得率の政府目標は2025年に50%、2030年に85%です（女性は既に約8割の取得実績）。直近の男性取得率は令和6年度で40.5%と過去最高を更新しています

■ 育休取得時に知っておくべき権利

- ・ 不利益取り扱いの禁止：育休取得を理由とした降格・給与減額・解雇は違法です（育児介護休業法）
- ・ マタハラ・パタハラの禁止：妊娠・出産・育休を理由とした嫌がらせはハラスメントです。第12章で学んだ内容と接続します
- ・ 社会保険料の免除：育休中は健康保険・厚生年金の保険料が免除されます（会社負担分も含む）
- ・ 年金への影響：育休中は保険料免除でも、将来の年金額に影響しない措置があります

■ 介護休業—突然来る準備のために

介護休業は「育児休業より身近でない」と感じるかもしれませんが、30代以降になると親の介護が突然始まることがあります。制度を知っておくことで、慌てずに対応できます。

- ・ 介護休業：対象家族1人につき93日間取得できます（3回まで分割可能）、介護が必要な状態になった時点で取得できます
- ・ 介護休暇：1年度に最大5日（対象家族が2人以上の場合は10日）の短期休暇です。通院の付き添い等に使います
- ・ 介護休業給付金：雇用保険から休業開始時賃金の67%が支給されます
- ・ フレックスタイム・テレワーク：介護との両立のために、働き方の柔軟化を会社に相談できます

16-7 メンタルヘルス—自分の心を守ることも仕事のうち

メンタルヘルスの問題は増えています。2022年度の労働安全衛生調査では、仕事や職業生活で強い不安・悩み・ストレスを感じる事柄が「ある」労働者は82.2%に上りました。メンタルヘルスは「弱さ」ではなく「職業的なリスク管理」と捉えることが大切です。

■ ストレスの早期サインを知る

メンタルヘルスの問題は、多くの場合、身体・行動・思考のサインとして現れます。早期に気づき、対処することが最も重要です。

- ・ 身体のサイン：睡眠の変化（眠れない・眠りすぎ）、食欲の変化、体の疲れが取れない、頭痛・胃腸の不調が続く
- ・ 行動のサイン：遅刻・欠勤が増える、ミスが増える、仕事への集中力が落ちる、人との関わりを避けるようになる
- ・ 思考・感情のサイン：何をしても楽しくない、気力がわかない、自分を責めすぎる、将来への強い不安が続く

■ 相談できる場所—一人で抱え込まない

- ・ 社内の相談先：直属の上司、人事担当者、産業医（常時50人以上の事業場には配置義務があります）
- ・ ストレスチェック制度：常時50人以上の事業場では年1回のストレスチェックが義務づけられています。結果は会社に通知されません（本人の同意なし）

- ・ EAP（従業員支援プログラム）：外部の専門機関によるカウンセリングサービスを導入している会社もあります。就業規則・社内ポータルで確認しましょう
- ・ 社外の相談先：「こころの健康相談統一ダイヤル」（0570-064-556）、各都道府県の子どもの健康相談センター
- ・ 傷病手当金：メンタルヘルスの不調により休職した場合でも、健康保険の傷病手当金（給与の3分の2×通算1年6ヶ月）を受給できます

■ 職場のメンタルヘルスに貢献する

自分のメンタルヘルスを守るだけでなく、チームの心理的安全性（第6章で学んだ内容）を育てることで、チーム全体のメンタルヘルスに貢献できます。

- ・ 「大丈夫？」の声かけ：元気がなさそうな同僚に一声かけることが、相談のきっかけをつくります
- ・ 「無理しなくていい」の文化：締め切りや業務量に追われているチームメンバーに、自分が余裕があれば「何か手伝えることある？」と聞ける関係性をつくります
- ・ 残業の連鎖を断ち切る：全員が残っているから自分も残る、という連鎖は、意識的に「帰る」人が一人いるだけで変わります

■ 心身のセルフチェックと、あなたが使ってよい外部リソース

メンタル不調の難しさは、当事者ほど自分の不調に気づきにくいことです。頭で気づく前に、身体がシグナルを出します。次のような状態が「3つ以上、2週間続く」ときは、無理をせず専門家への相談を検討してください（最初の数か月は誰でも疲れます。1～2項目が一時的に当てはまるのは自然なことです）。

- ・ 朝、布団から出るのに30分以上かかる
- ・ 食事の味がわからない、食欲が極端に落ちた・増えた
- ・ 出社前に動悸や吐き気がする
- ・ 休日に何もする気が起きず、ただ横になっている
- ・ 趣味や好きだったことに興味が湧かない
- ・ 涙が突然出る、または感情が平板になる
- ・ 飲酒量が増えた、薬・サプリで眠ろうとする
- ・ 「消えたい」という考えが頭に浮かぶ（これは1つでも、すぐ相談してください）

覚えておいてほしいことがあります。労働基準監督署、産業医、ハラスメント相談窓口、自治体のメンタルヘルス相談、弁護士・社労士、これらは、上司の許可なく使えるあなたの権利です。**「内部で解決しよう」と止める人がいたら、それ自体が注意すべきサインです。**相談窓口は変わることがあるので、厚生労働省「まもろうよ こころ」で最新の窓口を確認しておくことが確実です。自分のことだけでなく、同期や後輩のサイン（表情が乏しくなった、雑談に来なくなった、遅刻が増えた等）に気づいたら、「最近どう？」「ご飯行こう」と声をかけてみてください。深刻な相談は一人で抱え込まず、産業医・人事・専門機関につなぐ。気づいて、つなぐ。それがあなたにできる最大の支援です。

16-8 副業・兼業—複数の仕事を持つ時代への理解

政府の「働き方改革」を受け、副業・兼業を認める企業が増えています。2023年時点で、厚生労働省の「副業・兼業の促進に関するガイドライン」は、原則として副業・兼業を認める方向性を示しています。

■ 副業を始める前に確認すること

- ・ 就業規則の確認：まず自社の就業規則で副業が認められているかを確認します。**無許可での副業は規則違反になることがあります**
- ・ 競業禁止義務：在籍中は自社の競合他社での業務は原則として禁じられています。退職後も一定期間の制限がある場合があります
- ・ 守秘義務：副業先での業務に自社の機密情報・顧客情報を使うことは厳禁です
- ・ 健康管理：副業・本業合算の労働時間が過重になることで、健康被害や本業のパフォーマンス低下が生じるリスクがあります。両方の会社が労働時間を管理する義務を負います

■ 副業の税金と確定申告

- ・ 20万円超の副業収入がある場合：確定申告が必要です
- ・ 住民税の注意：副業収入が会社に知られる主なルートは「住民税の通知」、確定申告時に「副業分の住民税を自分で納付する」設定をしないとリスクを下げられます
- ・ 社会保険：副業先でも一定以上の収入・労働時間があると、副業先でも社会保険加入義務が生じる場合があります

16-9 テレワーク・多様な働き方と法律

テレワークやフレックス・裁量労働制など多様な働き方が広がっています。便利な一方で「労働時間の管理が難しい」「過重労働になりやすい」というリスクも伴います。

■ テレワーク時の労働時間管理

- ・ **自宅での業務時間も「労働時間」、会社は適切に管理する義務があります**
- ・ 業務時間外の連絡：テレワーク中でも、就業時間外の業務指示・連絡への対応を義務付けることは問題になりえます
- ・ 費用負担：テレワークに必要な通信費・光熱費等については、会社が負担するか補助する取り扱いが望ましいとされています
- ・ 安全配慮義務：テレワーク中であっても、会社は社員の安全・健康を守る義務を負います

■ フレックスタイム制・裁量労働制

- ・ フレックスタイム制：コアタイム（全員が働くべき時間帯）内は出勤し、それ以外の時間は自分で調整できます。月単位で所定時間を満たせばよい制度です
- ・ 裁量労働制：業務の進め方・時間配分を本人の裁量に委ねる制度、「みなし労働時間」が設定されており、実際の労働時間にかかわらず一定時間働いたとみなされます。IT・企画・研究職に多い制度です
- ・ 注意：裁量労働制であっても、深夜・休日労働の割増賃金は必要です。また、健康被害が生じた場合の会社の安全配慮義務は変わりません

事例から学ぶ：テレワーク中の過重労働と安全配慮義務違反—近年の事例から学べること

テレワーク導入後に、業務時間外の深夜まで連絡が来る・休日にも対応を求められる等の環境により体調を崩した社員が、会社に損害賠償を求めた事例が増加しています。

テレワークは「見えないところで働かせやすい」リスクを持つため、会社は明確な業務時間のルール設定・健康状態の把握・長時間労働の防止を、より意識的に行う必要があります。

この教訓から学べること：自分がテレワーク中であっても「今日はここまで」と業務を終わらせる自律的な時間管理と、過重な要求を受けたときに相談・報告する勇気が、自分の健康と権利を守りま

す。

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・ 有給休暇は「いつ・何日もらえて、どう申請するか」を、会社の制度で調べてみる。

16-10 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ 労働時間や残業の基本ルールを、ざっくり。
- ・ 自分が使える権利（有給など）を、思い出せるだけ。
- ・ 不調のとき、どうするのが大切か。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、これからの学びや仕事の中で、少しずつ埋めていきましょう。

第4部 総合ワークショップ（29～30日目・2日）

第4部の学びを統合する大型ケーススタディです。「法律・倫理・ハラスメント・情報管理」を横断する複合的な問題を議論します。

▶ やってみようーワーク・演習

やってみよう①：「倫理的ジレンマ」のグループ議論 架空の企業シナリオ（複数の法律・倫理問題が絡み合った状況）を提示します。

・法的にはどうか（関連法律を特定します） ・倫理的にはどうか（4層の正しさで評価します） ・どんな行動が自分・組織・社会を守るか の3軸で分析し、グループごとに発表します。

やってみよう②：実際の企業事例の分析 次の中から1つ選びます（候補：①ある食品会社の集団食中毒事件、

②ある地方銀行の不正融資事件、

③同社個人情報流出事件）。

「何が起点で」「誰が」「どこで」「どんな判断をしたか」を分解し、「自分が同じ立場だったらどうしたか」を議論します。

達成基準：再発を防ぐ具体策を、研修で学んだ「正しさの4層」に紐づけて1つ以上提案できること。

第4部（第10～16章）で学んだ「法律・倫理・ハラスメント・情報管理・コンプライアンス・労働権」を統合する大型ケーススタディです。複数の法律・倫理問題が絡み合うシナリオを用いて、「知識を判断に活かす力」を養います。

■ 演習①：複合ケーススタディー「あなたは どうする？」（1日目・5時間）

以下の「架空の会社のシナリオ」を読み、複数の問題点を法律・倫理・組織の観点から分析します。

【シナリオ：株式会社デジタルソリューションズの1週間】

- ・ 月曜日：上司から「このシステムにはセキュリティの脆弱性があるが、コスト削減のため対応を後回しにしよう。顧客には伝えなくていい」と指示を受けた
- ・ 火曜日：同僚が「業務で使ったOSSのライセンス条件を確認していない」と話しているのを聞いた
- ・ 水曜日：先輩から「テストのために本番DBのデータをコピーしていい」と言われた
- ・ 木曜日：自分がチームリーダーから「毎日22時まで残業しろ。残業代は出ないが、これが当社の文化だ」と言われた
- ・ 金曜日：取引先から個人情報が含まれた顧客リストをメールで受け取ったが、誤送信の可能性はある

【各シナリオへの分析項目】

- ・ ①関連する法律（個人情報保護法・労働基準法・不正競争防止法等）を特定する
- ・ ②「4層の正しさ」のどの層の問題かを分析する
- ・ ③あなたは どう行動するか、具体的なセリフ・行動を決める
- ・ ④最悪の場合何が起きるか、法的・倫理的・組織的なリスクを予測する

■ 演習②：ハラスメント事例の分析と対処（2日目・3時間）

第12章で学んだハラスメントの知識を、実際の場面に適用する演習です。

- ・ ケース分析：5種類のハラスメント事例（パワハラ・セクハラ・マタハラ・カスハラ・その他）について「これはハラスメントか？」を判断し、その根拠を示します
- ・ 対処法のロールプレイ：「ハラスメントを受けたとき」「ハラスメントを目撃したとき」「ハラスメントと指摘されたとき」の3つの立場を実践練習します
- ・ 相談体験：仮定の相談窓口ロールプレイ、相談する側と受ける側に分かれて、安心して話せる面談を体験します

▶ やってみようーワーク・演習

▶ 第4部総合ワークショップ 振り返り

- ①ケーススタディで「判断が最も難しかった」シナリオはどれか、なぜ難しかったか
- ②「4層の正しさ」は実際の判断の場面で使えそうか、使いにくい場面はどこか
- ③ハラスメント演習で「自分が気づいていなかった視点」は何か
- ④第4部全体で最も印象に残った「知識」と、それを活かす「行動」を書く

第5部 自分を育てる（31～34日目・4日間）

第5部では、成長し続けるための姿勢とキャリアの設計を学びます。ビジネスパーソンの価値は、入社時点の能力ではなく、学び続ける力で決まります。新卒の時期に「成長の習慣」を持てるかどうか、5年後・10年後の市場価値を分けます。そして、あなた自身の成長は、会社の成長の一部です。**会社にいる以上、あなたには、自分の立場でビジネス・組織・技術の成長に貢献する役割があります。**

入社時点の能力差は、数年で逆転します。**決め手は、才能ではなく、学び続ける習慣です。**この部では、成長を止める「見えない壁」への向き合い方と、自分のキャリアを自分で設計する考え方を学びます。そして、成長の環境を整えるのは、あなた自身の努力と、それを支える会社の責任の、両方であることも忘れないでください。

第17章 成長し続けることの重要性と阻害要素への対応（31～32日目・2日）

🎯 この章のゴール

この章では、成長し続けること、成長を阻む壁と、学びを身につけるコツを学びます。
終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ① **【気づける】** 成長を阻む内側の壁（完璧主義・比較など）に気づける。
- ② **【コツがわかる】** 「わかった」を「できる」に変えるにはアウトプットが要る、とわかる。
- ③ **【考え方がわかる】** フィードバックは成長のヒント、とわかる。

💡 まず自己診断—読む前に、いまの自分を知ろう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「成長し続けることを自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
- ② 「「成長する」とは、具体的に何をすることだと思えますか？」を、いまの考えで一言だけ書きま

す（正解は気にしません）。
読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

17-1 成長し続けることで広がる4つの人生の可能性

会社が成長し続けなければならないのと同じく、個人も成長し続ける必要があります。競争相手が成長する以上、立ち止まることは後退です。

人生の側面	成長し続けた場合	成長が止まった場合
人としての人生	好奇心・感動・視野が広がり続け、毎日に新しい発見があります	好奇心・感動・視野が失われ、毎日が同じ繰り返しになります
社会人としての人生	時代の変化に対応でき、選択肢が増え続けます	時代の変化に取り残され、選択肢が狭まり続けます
社員としての人生	「この人がいると違う」という存在になれる	「いてもいなくても同じ人」になります
専門家としての人生	技術・知識が積み上がり、自分の価値を自分で証明できます	技術・知識が陳腐化し、自分の価値を見失います

成長は義務ではなく、自分の人生を豊かにするための、自分への約束です。 今日から少しずつ積み上げることが、将来の大きな違いを生みます。

とはいえ、毎日全力で頑張る続ける必要はありません。成長は短距離走ではなく、長いマラソンです。疲れたら休んでいい。大事なのは、完全に止まってしまうこと。ゆっくりでも歩き続けた人が、いちばん遠くまで行きます。

「成長し続けなければ」と聞くと、しんどく感じるかもしれません。でも、ここで言う成長は、出

世や評価のことだけではありません。新しいことを知る楽しさ、できなかったことができる喜び、視野が広がる感動、成長は本来、人生を豊かにする“楽しいもの”です。義務ではなく、自分への贈り物だと捉えてみてください。

事例から学ぶ：技術変化と継続的な学びから学べること

「その仕事は一生続く」という前提が崩れた事例は多くあります。
この教訓から学べること：継続的な学びが自身のキャリアを守り、広げます。
変化を脅威ではなく機会として捉えることができる人が、長期的に活躍できます。

◆ 考えてみよう：1年後の自分

1年後、「こうなっていたら嬉しい」という小さな目標を、一つ思い浮かべてみましょう。大きな夢でなくて構いません。

「これができるようになりたい」という具体的な一つが、今日の学びに方向を与えてくれます。

17-2 どんな環境でも成長を続ける方法

成長機会の少ない環境に配属されても、成長の方法は自分で設計できます。

同時に、新卒が成長できる環境を整えるのは、迎える組織の責任でもあります。「環境を言い訳にしない」は新卒の心構えであって、組織が環境整備を怠ってよい理由にはなりません。

配属先は、自分では選べないことも多いものです。でも、その環境の中で「何を学べるか」は、自分で選べます。与えられた場所を嘆くより、その場所で拾えるものを探す、その視点が、どんな配属でもあなたを成長させます。

成長の供給源を分散させる設計：

- ・ 上司・先輩からの指導（状況により得られにくいこともあります）
- ・ 社外の人、技術コミュニティ・勉強会・同業他社の人
- ・ 書籍・論文・オンラインコース、自己投資による学び
- ・ 同僚・同期、同じ立場での学び合い
- ・ 顧客・現場、実際の問題から学ぶ
- ・ どんな仕事にも「工夫・改善・観察の余地」があります。作業を仕事に変えることで、どんな環境でも学べます（第5章との接続）
- ・ **環境を言い訳にせず、自分でできることを探して動く姿勢が、長期的に自分のキャリアを守り広げます**

とはいえ、「環境のせいにするな」を、自分を追い込む言葉にしないでください。本当に成長を妨げる環境（学べない・つぶされる）なら、第18章でも触れますが“動く”という選択もあります。「自分でできることを探す」と「環境を変える」は、どちらも前向きな一手です。

事例から学ぶ：成長型マインドセットから学べること

スタンフォード大学のキャロル・ドゥエック教授の研究によると、「努力で成長できる（成長型）」と考える人は、困難な課題にも前向きに取り組み、長期的に大きな成果を上げることが示されています。

す。

「自分には無理」という思い込みに気づき、「どうすれば成長できるか」を考えることが、可能性を広げます。

◆ よくある勘違いと、その反証

【勘違い1】「成長＝時間が経てば身につく」→

【反証】同じ年数でも差がつくのは、“意図的な練習”の差です。

できることの反復ではなく、「少し背伸びした課題」に挑み、フィードバックで修正する、この負荷が成長をつくります。

【勘違い2】「成長＝右肩上がり」→

【反証】成長は階段状で、必ず“伸びを感じない停滞期（プラトー）”が来ます。

停滞は失敗ではなく、次の飛躍の前兆。

ここで辞めた人だけが伸び悩みます。

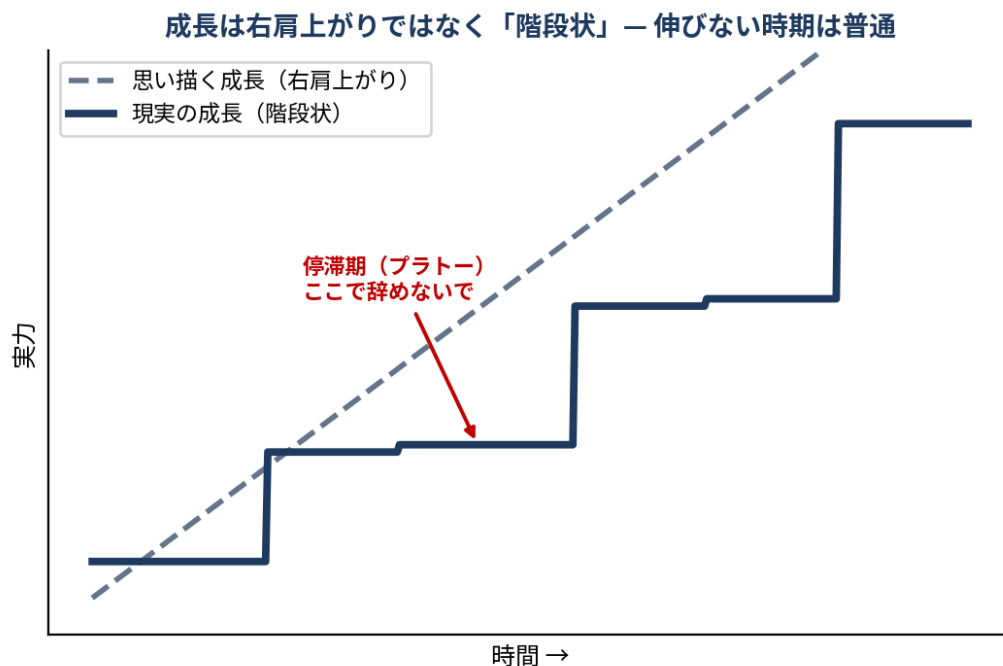


図 成長は階段状（停滞期＝プラトー）

▶ 経験を成長に変える—KPTで振り返る

経験は、振り返って言葉にして初めて“学び”になります。週に一度、3つだけ書きましょう。

【Keep】うまくいった・続けたいこと

【Problem】困った・うまくいかなかったこと

【Try】次に試すこと（Problemに対する具体的な一手）

ポイントは、感想（「大変だった」）で止めず、必ずTry＝次の行動まで落とすこと。これがOJT（特別章特-4）でそのまま使う、成長の基本動作です。

成長の機会は、上司や会社が与えてくれるのを待つものではありません。指導してくれる先輩、社外の勉強会、書籍やオンライン講座、同期との学び合い、そして目の前の仕事そのもの、供給源は、思った以上にたくさんあります。一つに頼らず、いくつかを自分で組み合わせておくと、どんな環境でも学び続けられます。

💡 ポイント：どんな仕事も“学びの教材”になる

「この仕事からは学べない」と感じる作業にも、たいてい工夫・改善・観察の余地があります。「もっと速くできないか」「なぜこの手順なのか」と一歩踏み込むだけで、同じ作業が学びの教材に変わります。

環境を言い訳にしないこの姿勢が、長い目で大きな差を生みます。

17-3 成長の習慣をつくる

習慣	内容	所要時間
振り返り	その日の学びを3行書きます（何を学んだか・何に気づいたか・次にどう活かすか）	5分/日
インプット	本・ニュース・技術記事・専門情報を読みます	15～30分/日
アウトプット	学んだことを誰かに話す・ブログに書く・社内でも共有します	週1～2回
対話	社内外の人と意図的に話す・コミュニティに参加します	週1回以上

▶ やってみよう—ワーク・演習

やってみよう：自分の成長の可能性を広げるための計画を立てます

- ①現在自分の成長に影響を与えている「外側の要因」「内側の要因」をそれぞれ2つ書きます
- ②各要因に対して「自分にできる行動」を1つ書きます
- ③「今週から始める小さな習慣」を1つ決めます

✦ 考えてみよう：今日からの“小さな習慣”

学びを続けるために、「これなら毎日できそう」という小さな習慣を一つ決めてみましょう。5分の読書でも、1行の振り返りでも構いません。

大切なのは、大きさではなく、続けられること。

17-4 成長を阻む「内側の壁」—自分で自分を止めないために

成長の機会は外側にあふれています。しかし**多くの場合、成長を妨げているのは環境よりも「自分の中の思い込みや感情」です**。代表的な阻害要因を知ること、意識的に対処できます。

ここからは、成長を妨げる“外側”ではなく“内側”の壁を見ていきます。少し耳が痛いかもしれませ

んが、安心してください。これらの壁は、特別に弱い人だけの話ではなく、誰の中にもあるものです。大切なのは、なくすことではなく、「あ、今これが出ているな」と気づけること。気づけた瞬間に、壁は半分崩れています。

■ 阻害要因①：コンフォートゾーンへの執着

コンフォートゾーンとは「慣れ親しんで安心できる状態」のことです。成長にはコンフォートゾーンの外に出ることが必要ですが、人間は本能的に変化を避けようとします。

- ・ 「今の自分にはまだ早い」という思い込み：新しい仕事・役割・技術を「まだ準備ができていない」と断ってしまうことで、成長の機会を逃します
- ・ 対処法：「少し怖い・難しいと感じること」は成長のサインです。「伸び代がある」と捉え直すことで、挑戦への姿勢が変わります
- ・ 実践：来週、「やったことのない」仕事を一つ引き受けることを決めましょう

■ 阻害要因②：完璧主義—「できてから見せる」の罠

特にエンジニアに多いのが、「完璧に仕上げてから見せよう」という完璧主義です。これは品質意識の裏返しですが、成長を遅らせる原因になります。

- ・ 完璧主義の弊害：着手が遅れる・フィードバックが遅くなる・完成までのプロセスで間違った方向に進み続けるリスクが高まる
- ・ 「80%の完成度で見せる」習慣：完璧でなくても早い段階で見せることで、方向修正が容易になり、結果として高品質なものが完成します。アジャイル開発の精神と同じです
- ・ 対処法：「完璧にしてから」を「まず動くものを作ってから」に言い換えます

完璧主義そのものは、悪いものではありません。品質を大事にする気持ちの裏返しです。問題なのは、それが“抱え込み”や“着手の遅れ”につながる時。完璧を目指す気持ちは大切にしつつ、「途中で見せる」勇気とセットにする、それが、完璧主義を“武器”に変えるコツです。

■ 阻害要因③：固定型マインドセット—「自分には向いていない」

スタンフォード大学のキャロル・ドゥエック教授が提唱した「マインドセット理論」によると、人のマインドセットには2種類あります。

- ・ 固定型マインドセット（Fixed Mindset）：「才能は生まれつき決まっている」という思い込み、失敗を「自分はダメだという証拠」として捉えます
- ・ 成長型マインドセット（Growth Mindset）：「努力と経験で才能は伸びる」という信念、失敗を「学びの機会」として捉えます
- ・ ITエンジニアとの関連：「自分はプログラミングに向いていない」「数学が苦手だから無理」という固定型の思い込みは、学習の機会を自分で閉ざします。多くの優れたエンジニアは「得意だったから始めた」のではなく「やり続けたから得意になった」のです
- ・ 対処法：「できない」を「まだできない」に言い換える習慣をつけます。この小さな言葉の変化が思考を変えます

■ 阻害要因④：比較による自己否定

SNSやチームの中で「あの人はもっとできる」という比較が自己否定につながりやすい時代です。比較は成長の燃料になることもありますが、過度な比較は学習意欲を奪います。

- ・ 他者比較より「過去の自分比較」：1年前の自分と比べて何が成長したかを定期的に確認します。日々の小さな成長は、他者との比較では見えにくいのです

- ・ 「上位互換を見つけて諦める」の罠：「すでに自分よりすごい人がいるから」という理由で学習をやめることは、自分の成長を他者に委ねることです
- ・ 対処法：「3ヶ月前の自分に教えられることが1つでもあれば、成長している」という基準を持ちます

SNSは、他人の“良いところだけ”が並ぶ場所です。そこと比べて落ち込むのは、編集された映画の名場面と、自分の日常を比べるようなもの。比べるなら、過去の自分とだけ。今日のあなたは、確実に、何かを一つ知っています。

📖 事例から学ぶ：完璧主義を手放して伸びた人

あるエンジニアは、「完璧に仕上げてから見せたい」と抱え込み、いつも提出がぎりぎりでした。あるとき思い切って、6割の段階で先輩に見せるようにしたところ、早い段階で方向を直せて、結果はむしろ良くなり、気持ちも楽に。完璧主義を手放したことが、成長の加速装置になったのです。

→ あなたにできる一歩：完成を待たず、“途中”を見せてフィードバックをもらう。

✦ 考えてみよう：あなたの“内側の壁”は？

4つの壁（安心への執着・完璧主義・「向いていない」という思い込み・人との比較）のうち、自分にいちばん当てはまりそうなのはどれでしょうか。

一つ自覚するだけで、次に出てきたとき、対処しやすくなります。

17-5 学習の技術—効果的に、長く、楽しく学ぶために

「学ぶべきことはわかっているが、続かない」という悩みは多くの社会人が持っています。学習の技術を知ることで、同じ時間でより多くを吸収し、より長く学び続けられます。

「読んだ」と「わかった」と「できる」は、全部違います。本を読んで分かった気になっても、実際にやってみると詰まる、それが普通です。だからこそ、学んだら小さく試す。手を動かして“できる”まで持っていく一手間が、知識を本物の力に変えます。

■ インプットの質を高める—「読んだ」を「わかった」に変える

- ・ アウトプット前提のインプット：「後で誰かに説明する」「ブログに書く」という前提で読むと、理解の深さが大幅に変わります
- ・ 「わからない箇所」を大切に：わからない部分を飛ばして読み進めると、後の理解が浅くなります。「なぜわからないのか」を問うことが深い理解につながります
- ・ 技術書の読み方：まず「目次」を読んで全体構造を把握→気になる章を深く読む→実際にコードを動かしながら読む、という順序が最も効率的です
- ・ 情報源の多様化：技術ブログ・公式ドキュメント・書籍・カンファレンス動画・コミュニティの議論、それぞれ強みが違います。一つの情報源に偏らないようにします

■ アウトプットで学びを定着させる

人は「学んだこと」より「使ったこと」を定着させます。インプットだけでは知識は定着しません。

- ・ 技術ブログ・Zenn・Qiita：学んだことを記事にすることで、自分の理解の穴が見えます。「人に説明できるレベル」の理解を目指します

- ・ 社内での知識共有：LT（ライトニングトーク）・勉強会での発表は、最良のアウトプットの機会です
- ・ OSSへの貢献：バグ報告・ドキュメント修正・小さな機能追加から始められます。世界中の開発者からフィードバックを受けられる最高の学習環境です
- ・ ペアプログラミング・コードレビュー：他者のコードを読み・自分のコードを読まれることで、自分の「当たり前」が問い直されます

アウトプットの王様は、「人に教えること」です。教えようすると、自分の理解の穴がはっきり見えます。完璧に分かっていなくても大丈夫。学んだことを、後輩や同期に一つ説明してみる、それだけで、学びは何倍も深く定着します。

■ 習慣化の技術—「やる気」に頼らない学習継続

- ・ 小さく始める：「毎日30分勉強する」より「毎日5分コードを書く」から始めます。継続することで習慣になり、自然に時間が伸びていきます
- ・ トリガーを設定する：「朝コーヒーを飲みながら技術記事を読む」「電車で移動中にPodcastを聞く」、既存の習慣に紐づけると継続しやすくなります
- ・ 記録する：毎日何を学んだかを記録することで、継続の達成感が生まれます。GitHubのコントリビューションカレンダーはこの原理を活用しています
- ・ 学習コミュニティに入る：一人で学ぶより、同じ目標を持つ人と学ぶ方が継続率が大幅に上がります。勉強会・技術コミュニティへの参加が効果的です

学びを長く続けるいちばんのコツは、「楽しむ」ことです。義務感だけだと続きません。少しでも面白いと感じる分野から始める、学んだことをすぐ試して「できた」を味わう、仲間と一緒にやる、楽しさは、最高の継続エンジンです。

💡 ポイント：「やる気」より「仕組み」

続かないのは、意志が弱いからではありません。続く人は、やる気に頼らず“仕組み”を作っています。「いつ・どこで・何分」を決め、既存の習慣にくっつける（朝のコーヒー中に記事の一つ、など）。

小さく始めて、止めない。それだけで十分です。

17-6 フィードバックを受け取る力—成長の最大の加速装置

成長が速い人の共通点は「フィードバックの受け取り方がうまい」ことです。**フィードバックは、自分が見えていない自分を教えてくれる最も価値の高い情報源です。**

逆に、フィードバックを「攻撃」と受け取って身構えてしまうと、せっかくの成長のチャンスを自分で閉じてしまいます。受け取り方ひとつで、同じ一言が“伸びる燃料”にも“傷”にもなる。だからこそ、受け取る技術を持っておく価値があります。

■ フィードバックを受け取る心構え

- ・ 「攻撃された」と感じないために：フィードバックは「あなた（人格）への批判」ではなく「行動・成果物への評価」です。自分とフィードバックの対象を切り離して考えます
- ・ まず「ありがとうございます」：防御的な言い訳より、まず感謝します。理解できない部分は「どのようにすれば改善できますか」と前向きな質問で返します
- ・ 全部取り入れる必要はない：フィードバックは参考情報です。複数人から同じ指摘がある場合は優先度が高く、一人だけの場合は「そういう見方もある」として考えます

そして、いちばん成長が速いのは、「フィードバックを待つ人」ではなく「取りにいく人」です。「ここ、どうでしたか」「もっと良くするには？」と自分から聞きにいく。求められたフィードバックは、相手も答えやすく、あなたの成長意欲も伝わります。

同時に、フィードバックの質を保つのは、与える側の責任です。組織や指導する側には、人格ではなく行動と成果物に向けて、具体的で改善につながる形で伝える責任があります。良いフィードバックは、取りにいく新卒と、誠実に返す組織の、両方があって成長の燃料になります。

- ・ フィードバックを求める習慣：「この方向性で進めていますが、何か懸念点がありますか」と積極的にフィードバックを求めることが、成長の速度を上げます

✦ 考えてみよう：最近もらったフィードバック

最近、誰かから言われた「アドバイス」や「指摘」を一つ思い出してみましょう。そのとき、素直に受け取れましたか。

もし身構えてしまったなら、次はまず「ありがとうございます」から始めてみませんか。

📖 事例から学ぶ：「フィードバックを活かした成長」の実例から学べること

ある新卒エンジニアは、コードレビューで毎回多数の指摘を受けていました。

最初は「自分はダメだ」と落ち込みましたが、指摘をすべてメモし「なぜそう指摘されたか」の理解に集中し始めました。

3ヶ月後、指摘の数は半分以下になり、1年後には自分がレビューをする側になりました。

この教訓から学べること：フィードバックの量は「相手が自分の成長に投資している量」でもあります。

多くのフィードバックを受ける時期は、最も成長できる時期です。

💡 ポイント

成長に必要なのは「才能」ではなく「成長し続けようとする意志と習慣」です。毎日少しずつ積み上げる力の力は、数年単位で見ると想像以上に大きいです。

入社1年目の今日から始める「小さな習慣」が、5年後・10年後のキャリアと人生の質を決めます。

▶ やってみようワーク・演習

やってみよう追加①：自分の成長阻害要因を特定する

以下の問いに正直に答えてみましょう。

- ①「やりたいが後回しにしていること」を1つ書きましょう。なぜ後回しにしているか
- ②「自分には向いていないかも」と思っていることは何か、それは本当に「向いていない」のか、それとも「まだ試していない」だけか
- ③最後に「フィードバックをもらった後、防御的になった」のはどんな場面か、その時本当はどう動けばよかったか

やってみよう追加②：学習計画を立てる

今後3ヶ月の「学習の習慣化計画」を作りましょう。

- ・ 毎日取り組む学習：内容・時間帯・時間
- ・ 週1回のアウトプット：何を・どこで（ブログ/社内共有/勉強会等）
- ・ 月1回の振り返り：何が成長したか・計画を修正するか

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・ 今日学んだことを1つ、誰かに（または自分に）声に出して説明してみる。

学び続ける力は、AIの時代にいっそう価値を持ちます。AIは知識を一瞬で出してくれますが、それを“自分の力”にできるかは、あなたが学び続けるかどうか次第。変化が速い時代に、いちばん陳腐化しないのは、特定の知識ではなく「学び続ける力」そのものです。

💡 ポイント：成長に必要なのは、才能より“続ける意志”

毎日少しずつでも、学び、ふり返り、人に説明し、フィードバックを受け取る、この地道な習慣の積み重ねが、数年後に大きな差になります。

スタートの速さより、止まらないことが、いちばん効きます。

17-7 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ 成長を阻む「内側の壁」を、思い出せるだけ。
- ・ 「わかった」を「できる」に変えるには何が要るか。
- ・ フィードバックは、どう受け取るとよいか。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、これからの学びや仕事の中で、少しずつ埋めていきましょう。

第18章 キャリアプラン—自分の成長を設計する（33～34日目・2日）

この章のゴール

この章では、キャリア、自分の成長を、自分で設計するという考え方を学びます。

終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ① **【考え方がわかる】** キャリアは「自分でつくるもの」という考え方がわかる。
- ② **【知っている】** 自分の強みを3つの層で整理できる、と知っている。
- ③ **【知っている】** 面談を成長に使える、と知っている。

まず自己診断—読む前に、いまの自分を知ろう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

① 現時点での「キャリアの考え方を自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。

② 「キャリアは、誰がつくるものだと思いますか？」を、いまの考えで一言だけ書きます（正解は気にしません）。

読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

18-1 キャリアは「自分がつくるもの」です

「キャリアは会社が与えてくれるもの」という時代は終わりました。

終身雇用の崩壊、技術の急速な変化、副業・転職の一般化。

現代において、キャリアは「自分で設計し、自分で育てるもの」です。

これは不安ではなく、自由を意味します。

どんな強みを伸ばすか、どんな仕事をしたいか、どんな人生を歩みたいか、

主体的に選択できる時代だということです。

もう一つ。キャリアに「出遅れ」はありません。スタートの早い遅いや、配属が希望どおりかどうかは、長い目で見れば小さな差です。大切なのは、どこにいても“自分の力を伸ばす姿勢”を持ち続けること。その姿勢さえあれば、どんな配属先も、あなたの成長の舞台になります。

考え方	内容
受動的キャリア	配属された部署で言われた仕事をこなす。評価は会社任せ
能動的キャリア	自分がどうなりたいかを考え、今の仕事・学習・人脈を設計する

入社直後からキャリアを設計する必要はありません。

しかし「今の仕事が自分の成長にどうつながっているか」を意識するだけで、同じ仕事でも得られるものが大きく変わります。

3年後・5年後の自分をぼんやりとでもイメージしながら働くことが、キャリア設計の第一歩です。今日の積み重ねが、未来の自分を作ります。

✦ よくある勘違いと、その反証

【勘違い1】「キャリア＝出世（役職が上がること）」→

【反証】出世は選択肢の一つにすぎません。

専門を究める（スペシャリスト）、領域を越える（越境）、地元で腰を据える、どれも対等なキャリアです。

第2章2-7で見たとおり、管理職にならない専門職の道も用意されています。

【勘違い2】「キャリアは最初に決めて、まっすぐ進むもの」→

【反証】多くの人のキャリアは、偶然の出会いや想定外の異動で形づくられます（計画的偶発性）。大事なものは“決め切ること”より、目の前の仕事に誠実に取り組み、来た機会に開いておくことです。

💡 ポイントー「何を大切にするか」は人それぞれでよい

キャリアに唯一の正解はありません。

「成長して大きな仕事がしたい」も、「定時で帰って家族との時間を大切にしたい」も、「地元で長く働きたい」も、上下のない対等な選択です。

大切なのは、世間や他人の物差しでなく、“自分が何を大切にしたいか（価値観）”を自分の言葉で持つこと。

それがブレない軸になります（第20章で深めます）。

「自由」には、少しでも責任がともないます。自分で選べるということは、選ばなかった道の結果も、自分のものになるということ。とはいえ、最初から完璧に選ぶ必要はありません。やってみて、合わなければ選び直す、その繰り返しがキャリアです。大切なのは、誰かに決めてもらうのを待つのではなく、小さくても自分で選ぶ経験を積むことです。

そして、キャリアは階段を一段ずつ上がるような一直線のものではありません。寄り道、回り道、思いがけない異動が、あとで大きな意味を持つこともよくあります（「点と点が、あとでつながる」と言われます）。今やっていることが将来どう役立つかわからなくても、目の前の仕事に誠実に取り組む価値は、ここに 있습니다。

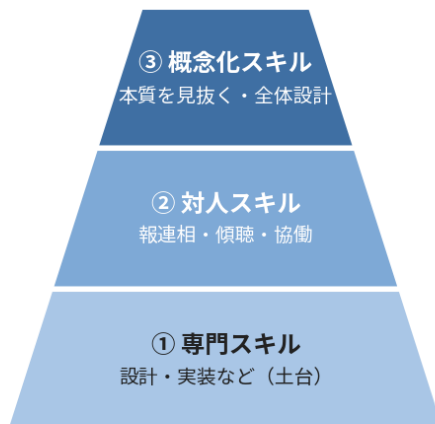
✦ 考えてみよう：あなたが大切にしたいこと

将来「こういう働き方ができたら幸せだ」と思える状態を、一つ想像してみましょう。お金・やりがい・人との関わり・自由な時間、どれを大事にしたいかは、人によって違います。正解はありません。

いまの素直な気持ちを、一言だけメモしておきましょう。それが、これからのキャリアを選ぶときの“羅針盤”になります。

18-2 スキルの三層構造と成長の方向

スキルの三層（①が土台、③は上の役割ほど効く）



まず縦棒（得意）を一本つくる

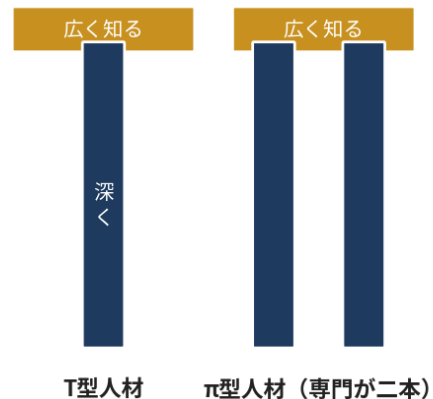


図 スキルの三層とT型・π型人才

キャリアを設計するには、まず「自分の強みはどこにあるか」を把握する必要があります。

ビジネスパーソンのスキルは、テクニカル（専門）・ヒューマン（対人）・コンセプチュアル（概念化）の3つの層に分けられます（3層の詳しい説明は第21章21-6で扱いました）。キャリアの設計では、この3層を「自分の強みはどこにあるか」の見取り図として使います。

①専門スキルは目立ちやすいですが、実際に長期的なキャリアを支えるのは②③です。

「どんな環境でも学び続けられる」という学習習慣は、技術変化の激しいIT業界では最大の武器になります。

新卒の段階では3つを均等に育てようとするより、まず①の基礎固めをしながら、②③を日常業務を通じて意識的に鍛えるのが現実的です。

自分の強みは、自分では気づきにくいものです。「ありがとう」と言われた場面や、人から頼られることを思い出すと、ヒントが見つかります。同僚や先輩に「自分の良いところはどこだと思う？」と聞いてみるのも、立派な強み発見の方法です。焦らなくて大丈夫。まず半年やってみて、見えてきた強みを伸ばしていけば、それで十分です。

■ 管理職だけが“上”ではない—道は一つではない

キャリアの道は「管理職になること」だけではありません。多くの会社には、人をまとめる管理職の道と、技術や専門性をとことん深める専門職（スペシャリスト）の道という、二つの道（複線型キャリア）が用意されています。どちらが上ということではなく、自分の強みと「大切にしたいこと」に合うほうを選べばよいのです。今すぐ決める必要はありません。

複線型キャリアー道は一つではない

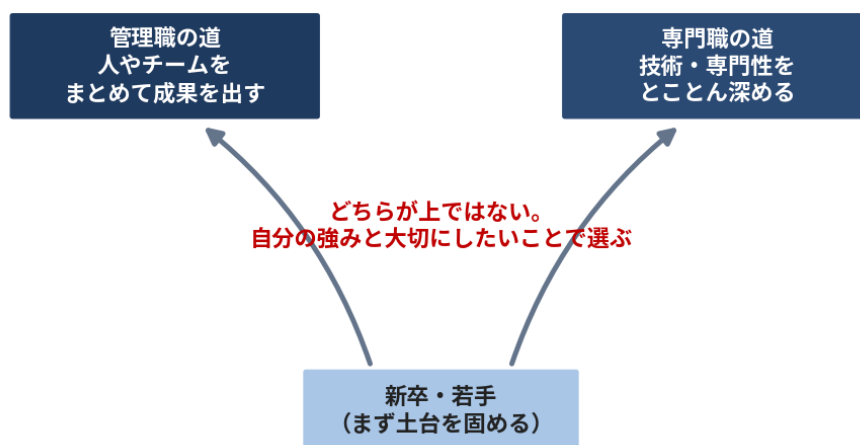


図 複線型キャリア (管理職／専門職)

■ T型・π型人材と、「持ち運べる力」

スキルの伸ばし方には、よく知られた形があります。一つの専門を深く持ちつつ周辺も広く知る「T型人材」、専門を二つ持つ「π（パイ）型人材」。新卒のうちは、まず一本の縦棒（得意分野）をつくることを意識すると、土台が安定します。広げるのは、その後でも遅くありません。

もう一つ大切なのが、「持ち運べる力（ポータブルスキル）」です。会社や職種が変わっても通用する力、論理的に考える力、人に分かりやすく伝える力、学び続ける習慣など。特定のツールや製品の知識は数年で古くなりますが、これらは長く効きます。自分の“市場価値”（ほかの場所でも通用する度合い）は、この持ち運べる力をどれだけ積み上げたかで決まっていきます。

💡 ポイント：今すぐ役立つ力と、長く効く力

目の前の仕事に必要な「専門スキル」は、もちろん大切です。それと並行して「持ち運べる力（考える・伝える・学ぶ）」を意識して鍛えておくと、5年後・10年後の選択肢がぐっと広がります。

日々の仕事は、その両方を同時に鍛える練習の場、そう捉えれば、ふだんの業務の見え方が変わります。

18-3 新卒3年間のステージ設計

「入社3年間」はキャリアの土台を作る最も重要な時期です。

この期間をどう過ごすかが、その後のキャリアの幅と深さを大きく左右します。

時期	テーマ	目標
1年目	吸収と適応	業務の基本を覚える。小さな仕事を確実にやり切る。 チームのやり方を理解し、信頼を積み重ねる
2年目	貢献と自	自分で考えて動けるようになる。後輩に教えられるレベルになる。

時期	テーマ	目標
	立	担当領域で「この人に聞けばわかる」という存在になる
3年目	専門性と方向性	自分の強みが見えてくる。スペシャリスト／マネジメントなどキャリアの方向性を意識し始める。社外との接点を持つ

ただし、これはあくまで一般的な目安です。

大切なのは「今自分はどのステージにいるか」を意識しながら働くことです。

💡 ポイント：比べる相手は、他人ではなく“過去の自分”

同期や周り比べて焦る必要はありません。成長のペースは人それぞれです。

見るべきは「半年前の自分より、何ができるようになったか」。過去の自分と比べられる人が、長く伸び続けます。

- ・ 1年目に「もっと面白い仕事がしたい」と焦る必要はない：基礎なき応用はない
- ・ 3年以内の転職を否定しない：ただし「逃げの転職」より「攻めの転職」を選ぶ
- ・ 「石の上にも三年」は一律の正解ではない：環境が成長を妨げているなら動く判断も必要

■ 1年目・2年目・3年目で、意識することは変わる

1年目は「吸収と適応」。小さな仕事を確実にやり切り、チームのやり方を覚え、信頼を積みます。背伸びより、土台づくりの時期です。

2年目は「自走と工夫」。指示を待つのではなく、自分で段取りし、「もっと良くできないか」と改善を回します。後輩のサポートも始まります。

3年目は「自分の強みを言葉にする」。これまでの経験を棚卸しし、「自分はこれが得意」「次はここを伸ばしたい」を語れるようにします。ここが、その後の方向を決める分かれ目になります。

📦 事例から学ぶ：環境を変えて伸びた先輩

ある若手は、最初の配属先で「学べることが少ない」と感じ、悩んでいました。すぐ辞めるのではなく、まず半年、与えられた仕事をやり切り、その間に身につけたい力を言葉にして上司に相談。結果、希望に近いチームへ異動でき、そこで力を伸ばしました。

→ あなたにできる一歩：もし今の環境に迷ったら、まず「ここで身につけられること」をやり切りつつ、希望を言葉にして相談してみる。逃げではなく、準備したうえで動くことが、次の成長につながります。

■ 社内で、キャリアを広げる

キャリアを広げる方法は、転職だけではありません。多くの会社には、別の部署を経験するジョブローテーション、やりたい仕事に手を挙げる社内公募、新しい役割への立候補など、社内で挑戦できる仕組みがあります。まずは「こういうことに挑戦してみたい」と、ふだんから上司や面談で口に出しておくこと。チャンスは、意思表示をしている人のところに回ってきやすいものです。

新卒のうちは「与えられた場所で頑張る」ことが基本ですが、それは「希望を言うてはいけない」という意味ではありません。目の前の仕事を誠実にこなしながら、行きたい方向をそっと示しておく、この両立が、社内でのキャリアを少しずつ広げていきます。

18-4 リスキリングと自己投資

社会人の学びは「終わりません」、技術・法律・社会は変わり続けます。 リスキリングとは新しいスキルを意図的に獲得することです。

学ぶ対象は、技術だけではなく、ビジネスの知識（自社や業界の仕組み）、対人の力（伝える・聴く）、お金や法律の基礎、どれも、長く働くうえで効いてきます。「自分に今いちばん足りないものは何か」を考え、技術以外にも目を向けると、学びの幅が広がります。

なお、資格は目的ではなく、学びの“きっかけ”や“地図”として使うと効果的です。取ること自体がゴールにならないよう、「それで何ができるようになりたいか」を一緒に考えておきましょう。そのうえで、興味のある分野から一步、踏み出してみてください。

自己投資の種類	内容	具体例
時間の投資	学ぶ時間を意図的に確保します	毎朝30分の技術書・週1回の勉強会
お金の投資	書籍・資格・セミナーへの支出を「投資」と捉えます	技術書・資格受験費・カンファレンス参加費
関係の投資	社内外のネットワークを育てます	コミュニティ参加・OSSへの貢献・メンターを探す

▶ やってみようワーク・演習

やってみよう①：自分の強みと成長したい方向を書き出します「今の自分が得意なこと」と「3年後になりたい姿」を書き出し、ギャップを埋めるための行動を3つ特定します。

やってみよう②：3年間のキャリアロードマップを作ります「6ヶ月後・1年後・3年後の目標」と「そのために今月・今週・今日から始めること」をロードマップの形で整理します。グループで共有し、相互にフィードバックします。

■ AI時代に、陳腐化しない学び方

AIが多くの作業を助けてくれる時代には、「答えを出す力」だけでなく「問いを立て、答えが本当かを確かめる力」が大切になります。だからこそ、個々のツールの使い方を追いかけるだけでなく、その土台にある“仕組み”や“体系”を理解しておくことが、長く効く学びになります。これは、この研修全体を貫く考え方、「AI時代にこそ、体系（全体像と原理）が効く」、と、まっすぐつながっています。

学びを続けるコツは、大きく始めないことです。毎朝15分、週に一つ新しい言葉を調べる、月に一冊だけ読む、小さく、でも止めない。「いつ・どこで・何分やるか」を決めて、生活のルーティンに組み込むと、やる気に頼らずに続けられます。

💡 ポイント：学びは「習慣」にすると続く

やる気に頼ると、学びは続きません。通勤時間に記事を一つ、寝る前に5分ふり返る、小さな習慣を一つ決めて、生活に埋め込みましょう。

小さな積み重ねが、数年後に大きな差になります。「続けられる量」から始めるのが、いちばんの近道です。

■ 社外とつながる—アウトプットと学びの輪

学びは、一人で抱え込むより、外に出すと加速します。勉強会に参加する、学んだことを短く書いて発信する、社外の人と話す、こうした「アウトプット」は、理解を深めるだけでなく、思いがけない出会いや機会を運んでくれます。最初は読むだけ・聞くだけで十分。少しずつ、輪の中に入っていきましょう。

18-5 1on1・評価面談の活かし方—成長の機会として使い倒す

1on1・評価面談は「評価される場」ではなく「自分の成長を設計し、サポートを得る場」です。
この認識の違いが、1on1の質を大きく変えます。

■ 1on1を最大限活かすための準備

- ・ 毎回の議題を自分で持っていく：「最近考えていること」「詰まっていること」「相談したいこと」を1〜2つ準備します。「特に何もありません」で終わる1on1は機会の損失です
- ・ キャリアの話を避けない：「3年後にどんなエンジニアになりたいか」「そのために今のプロジェクトで何を伸ばしたいか」を定期的に話します
- ・ フィードバックを求める：「今の自分の強みと改善点を教えてください」と定期的に聞きます。評価面談の前ではなく、普段の1on1で聞くことが重要です

■ 評価面談でやるべきこと

- ・ 自己評価の根拠を準備：「〇〇のプロジェクトで〇〇の問題を解決し、〇〇の効果を生んだ」という具体的な実績を数字付きで準備します
- ・ 次の成長目標を自分から提案：「次の半期で〇〇を伸ばしたい。そのために〇〇に挑戦したいが、機会をもらえないか」と自分から提案します
- ・ 評価への不満があるとき：感情的になる前に「評価の根拠を教えてください」と質問します。理解できたら受け入れ、納得できなければ「どうすれば次の評価につながりますか」と前向きに問います

📖 事例から学ぶ：小さな発信が道をひらいた

ある若手は、学んだことを社内チャットに短くメモする習慣を続けていました。やがて「詳しい人」として相談が集まり、勉強会の講師や新しいプロジェクトへの声がかかるように。特別なことはしていません。学びを少しだけ外に出し続けたことが、機会を呼び込んだのです。

→ あなたにできる一歩：今日学んだことを一言だけ、誰かに共有してみる。

💡 ポイント：キャリアは「点検」しながら

立てた計画は、半年〜1年に一度、見直して大丈夫です。状況も、あなたの興味も変わります。大事なのは、立てっぱなしにせず、ときどき立ち止まって「今の自分はどこにいるか」を点検すること。完璧な計画より、点検する習慣です。

✦ 考えてみよう：あなたの“持ち運べる力”

今のあなたが、会社や職種が変わっても通用しそうな力（考える・伝える・学ぶ・人と働く…）を、一つ挙げてみましょう。

それを、これからの3年で、どんな場面で伸ばせそうですか。

事例から学ぶ：キャリアを意図的に設計した人とそうでない人の10年後から学べること

同期入社で同じ技術スキルを持つAさんとBさんの10年後。

Aさんは毎年の評価面談で「次に挑戦したいこと」を上司と話し合い、社内勉強会での登壇・後輩育成・プロダクト企画への参画を積み重ね、エンジニアリングマネージャーに。

Bさんは与えられた仕事を誠実にこなし続けたが、「次にやりたいこと」を言語化しておらず、上司や制度の側もそれを拾い上げられなかったため、同じポジションに留まり続けました。

この教訓から学べること：キャリアは「与えられるもの」ではなく「設計して勝ち取るもの」です。意図的な設計と、それを周囲に伝える勇気が、キャリアの選択肢を広げます。

ポイント

キャリアプランは「完璧に立てる必要はない」し「一度立てたら変えてはいけない」ものでもありません。大切なのは「今の自分の強み・弱み・やりたいことを定期的に問い直し、次の一歩を意識して踏み出すこと」です。

入社1年目からキャリアを意識することは、決して早すぎません。

今日から少しずつ、自分のキャリアの設計者になりましょう。

▶ やってみよう—ワーク・演習

やってみよう追加：キャリアビジョンの具体化

以下のフレームワークで、自分のキャリアビジョンを整理しましょう。

【強み・弱みの整理】

- ・今の自分の「技術的な強み」を3つ書きましょう
- ・今の自分の「伸ばしたいスキル」を3つ書きましょう

【3年後のビジョン】

- ・3年後に「どんな仕事を・どんな形でやっていきたいか」を1段落で書きましょう
- ・そのビジョンに向けて「今年やること」「来年やること」「再来年やること」を1つずつ書きましょう

【市場価値の可視化計画】

- ・GitHubプロフィール・技術ブログ・資格のうち、今年取り組むものを1つ決めましょう
- ・3ヶ月後の具体的な目標（記事を3本書く・資格試験を申し込む等）を設定しましょう

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・3年後の自分が「できるようになっていたこと」を1つ書いてみる。

■ ロールモデルとメンターを持つ

成長を速める近道の一つが、「お手本になる人（ロールモデル）」と「相談に乗ってくれる先輩（メンター）」を持つことです。「あの人のこういうところを見習いたい」という相手を、社内に一人見つけてみましょう。正式なメンター制度がなくても大丈夫。尊敬できる先輩に、ときどき相談させてもらうだけで、見える世界が変わります。一人に絞らず、場面ごとに「この人のここ」を学ぶ“いいとこ取り”でも構いません。

■ ふり返り（リフレクション）の習慣

もう一つ、地味だけれど効くのが「ふり返り（リフレクション）」です。一日の終わりや週末に、「うまくいったこと」と「次はこうしたいこと」を一つずつ書き留める。たったこれだけで、同じ経験から学べる量が大きく変わります。経験は、ふり返って初めて“学び”になります。やりっぱなしにせず、立ち止まる時間を、自分のために少しだけ取りましょう。

◆ 考えてみよう：3年後の自分への、小さな一歩

3年後、「できるようになっていた」ことを一つだけ挙げてみましょう。

そのために、今月できそうな小さな一歩は何でしょうか。大きな計画より、今日動ける小さな一歩のほうが、未来を変えます。

18-6 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ キャリアは、誰がつくるものか。
- ・ 自分の強みは、どんな層で整理できるか。
- ・ 面談は、何に使えるか。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、これからの学びや仕事の中で、少しずつ埋めていきましょう。

第6部 人生という視点で、働くことを設計する（35～37日目・3日間）

第6部では、仕事を人生という長い時間軸の中に位置づけます。ビジネスパーソンとして働く約40年を、お金・価値観・健康の面から設計します。早い時期に全体像を持つことが、後悔のない選択につながります。

働くことは、人生の一部であって、すべてではありません。お金の不安に振り回されないための知識、何を大切にするかという価値観、そして心と体の健康。この三つを早いうちに見渡しておく、目の前の仕事にも落ち着いて向き合えます。**長い道のりを、最初に地図を持って歩き出すための部です。**

第19章 ライフプラン—人生とお金を設計する（35～36日目・2日）

この章のゴール

この章では、ライフプランとして、人生とお金の全体像のつかみ方を学びます。

終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ①【イメージできる】手取り・固定費・貯められる額の全体像がイメージできる。
- ②【知っている】長期でお金を増やす考え方（NISAなど）があると知っている。
- ③【知っている】税金・社会保険の最低限を知っている。

まず自己診断—読む前に、いまの自分を知ろう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

① 現時点での「人生とお金の設計を自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。

② 「お金のことは、いつから考え始めるとよいと思いますか？」を、いまの考えで一言だけ書きます（正解は気にしません）。

読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

19-1 人生の全体地図を描く

入社初日から人生全体を見渡す習慣を持つことが、後悔のない選択につながります。早く意識を持つほど、選択肢が広がります。

ライフステージ	主な出来事	主な支出・準備
22歳 入社	社会人のスタート	一人暮らしの生活費・自己投資の開始
20代（スキル形成期）	技術・ビジネススキルの習得	資格取得費・コミュニティ参加費・自己投資
30代（安定・転換期）	結婚・住宅取得・子育て開始・親の老いの兆候	住宅ローン・育児費用・介護の準備
40代（充実・負荷期）	教育費ピーク・住宅ローン継続・親の介護開始	教育費・ローン返済・介護費用の並走
50代（仕上げ期）	老後準備の本格化・子どもの独立・健康管理	老後資金の積み立て・健康投資
60代（移行期）	定年・再雇用・年金受給開始	退職金の活用・生活費の見直し

- ・ 人生100年時代、40年働いて40年生きる現実があります。
- ・ 早く始めるほど有利になります。その理由が「複利」です（次の事例で具体的に見ます）。始める時期の違いだけで、将来の金額には大きな差が生まれます

「複利」とは、増えた分にもさらに利息がつく仕組みのこと。雪だるまが転がりながら大きくなるイメージです。だから、金額より「時間」が効きます。少額でも、早く始めた人ほど、最後は大きく育つ。お金の準備において、思い立った今日という日が、いつだって、いちばん早いタイミングなの

です。迷うくらいなら、まず少額から始めてみましょう。

📖 事例から学ぶ：「20代から始めた積立投資」の複利効果—数字で見る時間の力

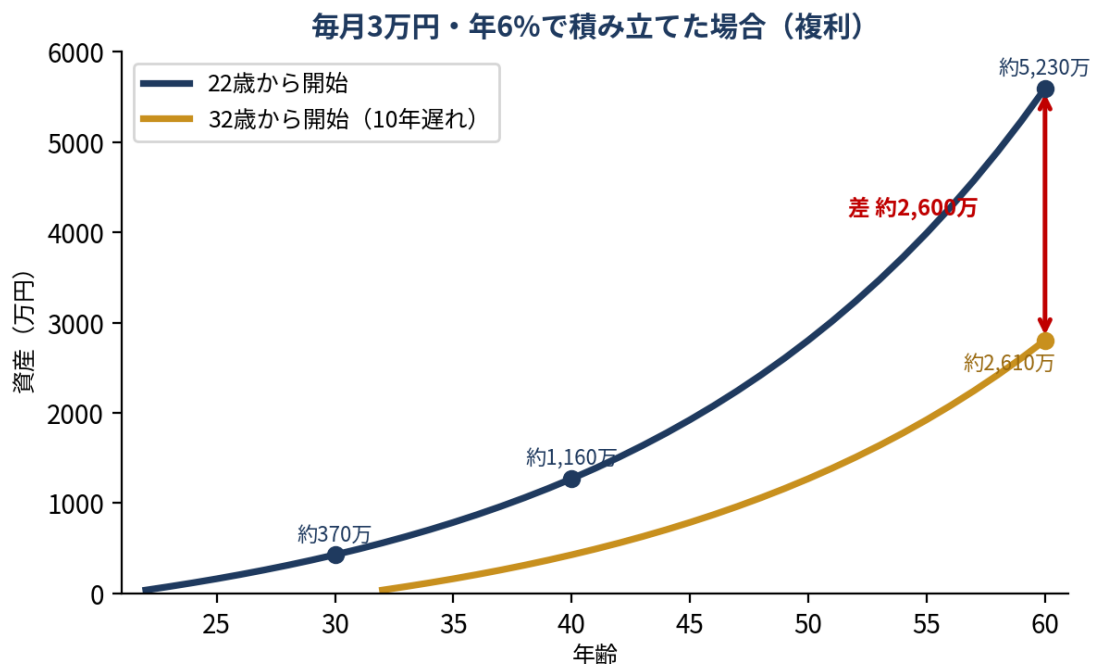
22歳から毎月3万円を全世界株式インデックス等で年6%運用できたと仮定した場合：30歳時点で約370万円、40歳時点で約1,160万円、60歳時点で約5,230万円になります（毎月積立・複利での試算）。

同じ投資を32歳から始めた場合、60歳時点では約2,610万円、開始を10年遅らせるだけで約2,600万円の差になります。

年6%はあくまで仮定であり、市場は変動し元本割れもありえます。

この教訓から学べること：「今は生活費で精一杯」という状況でも、月1〜3万円から始めることで、30〜40年後の生活は大きく変わります。

最大の敵は「あとから始めればいい」という先送りです。



※ 年6%はあくまで仮定。市場は変動し元本割れもありえる。大切なのは「金額」より「始める時期」。

図 毎月3万円・年6%で積み立てた場合（複利）

人生の地図は、今ぴったり描けなくて大丈夫です。むしろ、これから何度も書き換わっていきます。大事なのは、正確に予言することではなく、「お金や人生には大きな流れがある」と知っておくこと。地図があれば、迷ったときに「今どのあたりか」が分かり、落ち着いて選べます。

✦ 考えてみよう：5年後の自分

5年後、どんな生活をしていきたいですか。一人暮らし、貯金、やってみたいこと、ざっくりで構いません。

ぼんやりとでも“行き先”を思い描くと、今のお金との付き合い方が、少し変わってきます。

19-2 お金のリアル—手取り・固定費・3年で貯められる額

お金の話は、ぼんやりした「不安」を「数字」に変えると、急に扱えるようになります。まずは“いま”の自分のお金から見ていきましょう。

「お金の不安」は、正体が見えないから大きく感じます。手取りはいくらで、固定費はいくらで、毎月いくら残るか、数字にしてみると、不安は「対処できる課題」に変わります。怖いのは、知らないこと。知れば、たいていのことは扱えるようになります。

■ 額面と手取りは違う―「引かれてから」が自分のお金

給与は、額面（総支給）から社会保険料と税金が引かれ、残りが手取りとして振り込まれます。新卒1年目は前年の所得がないため住民税がかからず、2年目から住民税が加わって手取り率が下がる、これは多くの新卒が驚く点です。

項目	1年目のイメージ	補足
額面（月・総支給）	22万円	※金額はあくまで一例です
健康保険・厚生年金・雇用保険	約 -3.3万円	厚生年金は会社が半額負担。料率は年・地域で変動します
所得税	約 -0.3万円	収入により変わります
住民税	1年目は0円	2年目から課税（月1万円前後の例も）
手取り（おおよそ）	約 18万円前後	額面のおよそ8割。2年目以降は住民税ぶん下がります

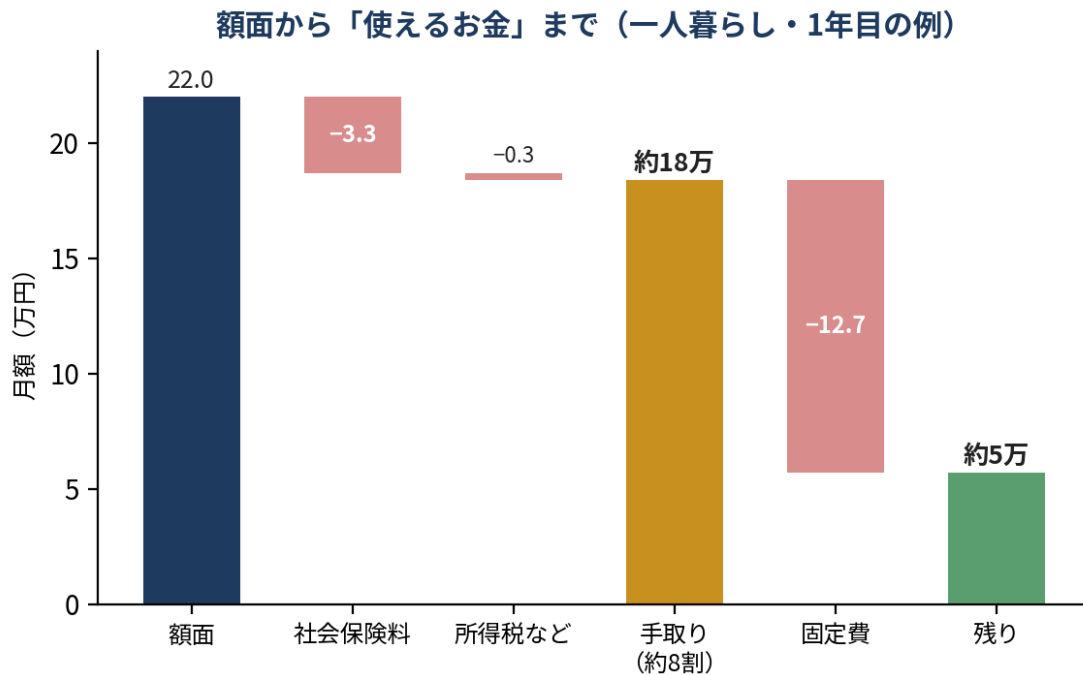
正確な金額は、毎月の給与明細と、第9章（数字・財務）の考え方で必ず自分の数字を確認してください。ここで伝えたいのは、「額面＝使えるお金ではない」という感覚です。

お金を考えるとき、つい「額面（月22万円など）」で考えがちですが、実際に使えるのは、そこから税金や社会保険料を引いた「手取り」です。額面と手取りには、けっこうな差があります。だから、家賃や生活の計画は、必ず手取りを基準に立てる。これを知らずに額面で考えると、「思ったより残らない」と慌てることになります。

■ 固定費を知れば、毎月の余白が見える

手取りから、毎月ほぼ自動で出ていくお金（固定費）を引いた残りが、自由に使える・貯められるお金です。下は一人暮らしの一例です。

固定費	月額の一例	見直しの勘所
家賃（管理費込み）	6.0万円	手取りの3割以内が目安。住む場所で大きく変わります
水道・光熱	1.2万円	季節で変動します
通信（スマホ・回線）	1.0万円	格安プランで下げやすい代表格です
食費	3.0万円	自炊の比率で変わります
奨学金返済	1.5万円	人により有無・額が異なります
合計（固定費）	12.7万円	手取り18万なら、残りは約5.3万円



※ 額面がそのまま使えるわけではない。手取りから、家賃などの固定費も自動で出ていく。

図 額面から手取り・残額まで（一人暮らし・1年目の例）

地方で働く場合、家賃や通勤コストが都市部より低く、同じ手取りでも“残る額”が大きくなりやすい。ただし家賃や生活費は住む地域や暮らし方で変わるため、これは一例です。

家計をラクにするコツは、毎月変わる「使うお金」より、ほぼ固定で出ていく「固定費」を先に見直すことです。家賃・通信費・サブスクなど、固定費は一度下げれば、何もしなくても毎月効きます。節約というと「我慢」を思い浮かべますが、固定費の最適化は、我慢せずに余白を作る、いちばん賢い方法です。

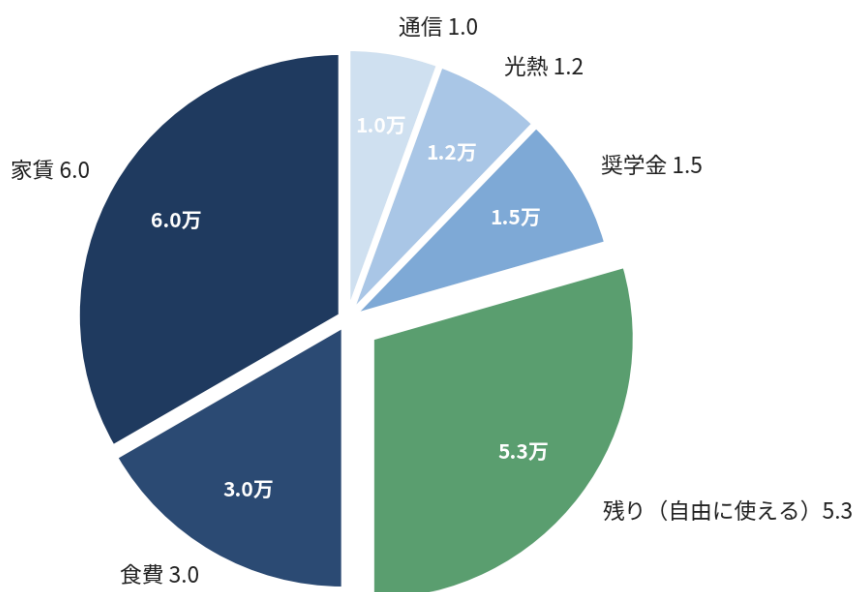
■ 3年で、いくら貯められるか

残った約5万円のうち、毎月3万円を先に貯蓄・投資に回す（先取り貯蓄）と決めると、3年で約108万円。これをNISA（19-6）で長期運用すれば、複利が時間を味方につけてくれます。大切なのは金額の大小より、「手取りが入ったら“先に”貯蓄を取り分ける」習慣を1年目からつくることです。

▶ やってみよう—自分の「お金の地図」を1枚で

- ①自分の手取り（見込みでよい）
- ②固定費の合計
- ③残り（②を①から引く）
- ④そのうち毎月いくらを先取りで貯蓄・投資に回すか、この4つを書き出してみましょう。数字にすると、漠然としたお金の不安は「設計できる対象」に変わります。

手取り（約18万）の使いみち — 自動で出ていく固定費と「余白」



※ 固定費（家賃・通信など）を先に押さえると、自由に使える「余白」が見える

図 手取りの使いみち（固定費と余白）

19-3 人生の四大支出を知る

支出	おおよその目安	備えのポイント
結婚	300～500万円	式・指輪・新生活の初期費用。早めから少しずつ積み立てると安心です
住宅	3,000～5,000万円	35年ローンの月々の負担を計算します。頭金・諸費用も必要です
教育	子ども一人あたり1,000～2,000万円	公立か私立かで大きく変わります。早めの準備が選択肢を広げます
老後	夫婦で2,000万円以上が目安	早くから少しずつ積み立てることで、複利の恩恵を受けられます

📖 事例から学ぶ：老後2,000万円問題（2019年）から学べること

金融庁の報告書が「老後に年金以外で2,000万円が必要」とした内容が社会に大きな衝撃を与えました。

この教訓から学べること：若いうちから資産形成を始めることで、複利の力を最大限に活かします。「いつか」は、たいてい永遠に来ません。

NISA口座は今日つくれて、複利は待ってくれません。

結婚・住宅・教育・老後、人生には、まとまったお金が要る場面が、いくつかやってきます。怖が

る必要はありません。大事なのは、「いつ・どれくらい要りそうか」を、早めにざっくり知っておくこと。早く知れば、その分、長い時間をかけて少しずつ備えられます。時間は、お金の最大の味方です。

四大支出（結婚・住宅・教育・老後）の金額を見て、「そんなに！」と不安になるかもしれません。でも、これらは一度に全部来るわけではなく、人生に分散してやってきます。来る時期もだいたいの予想がつく。だから、慌てず、来るものから順に、少しずつ備えればいいのです。

19-4 社会保険と資産形成の基本

■ 社会保険の仕組みと意味

保険の種類	何のためにあるか	知っておくべきポイント
健康保険	病気・ケガの医療費補助（原則3割負担）	傷病手当金（休業中の所得補償・通算1年6ヶ月）・出産手当金も活用できます
厚生年金	老後・障害・遺族の生活保障	会社が半額負担します。老後だけでなく障害・死亡時の保障もあります
雇用保険	失業・育休・スキルアップ支援	育休給付金（最大67%）も活用できます。転職時の失業給付の条件も知っておきます
労災保険	業務中・通勤中の事故の補償	全額会社負担です。業務中のケガは必ず利用できます

社会保険料は、給料から強制的に引かれるので「損」に感じるかもしれません。でも、これは病気・ケガ・老後・失業など、いざというときにあなたを支える“みんなで支え合う仕組み”です。引かれている分、あなたも守られている。そう知っておくと、給与明細の見え方が変わります。

19-5 働き方と制度の変化を見通す

- ・ 定年延長・再雇用・70歳就業時代、2021年の高年齢者雇用安定法改正により、70歳まで就業機会を確保する努力義務が企業に課されました
- ・ 副業・兼業、就業規則を確認した上で取り組むことで、スキルアップと収入の両方が得られます
- ・ 介護休業制度（最大93日間・3回まで分割可）、知っておくことで、いざというときに安心して活用できます

働き方の制度は、年々増え、変わっていきます。今すぐ全部覚える必要はありません。大事なのは、「こういう制度がある」と頭の隅に置いておくこと。いざ必要になったとき、「たしか、そんな制度があったな」と思い出して調べられれば、それで十分です。

▶ やってみようワーク・演習

やってみよう①：自分のライフイベント表を書く 22歳から65歳まで、自分に起こりうるライフイベントを年齢と一緒に書き出します。

「起きてほしいこと」と「備えるべきこと」を分けて考えます。

やってみよう②：10年後までの資金計画の骨格を描く 「今の収入」「毎月の支出」「貯蓄額」を確認

認し、「10年後に何のためにいくら必要か」を考えます。
 具体的な数字は仮でも大丈夫です。
 「お金のことを考える習慣」を始めることが目的です。

19-6 NISAとiDeCo—今すぐ始めるべき資産形成の実践

「老後2,000万円問題」という言葉が示すように、公的年金だけでは豊かな老後の生活は難しい時代になりました。

■ 新NISA（2024年～）—投資利益が非課税になる制度

- ・ NISAとは：株式・投資信託等への投資で得た利益・配当が非課税になる政府の制度、通常は利益の約20%が課税されます
- ・ 2024年からの新NISA：年間投資枠360万円・生涯投資枠1,800万円に大幅拡充。非課税期間が無期限になりました
- ・ つみたて投資枠（年間120万円）：長期・積立・分散投資に適した投資信託が対象、毎月一定額を積み立てる「ドルコスト平均法」に最適です
- ・ 成長投資枠（年間240万円）：上場株式・投資信託が対象、ある程度知識がついてから活用するのがおすすめです
- ・ 最初の一步：証券口座を開設してNISAを設定し、全世界株式インデックスファンドを月1万円積み立てから始めることが、最もシンプルで効果的な方法です

ここで大事なのは、「始めたら、ほったらかしでいい」ということです。毎日値動きを見て一喜一憂する必要はありません。むしろ、ほったらかしのほうが、長期ではうまくいきやすい。設定したら、あとは淡々と続ける、それが、忙しい社会人に向けたやり方です。

NISAやiDeCoと聞くと難しく感じるかもしれませんが、最初の一步はとても小さくて大丈夫です。「月1万円を、全世界に分散した投資信託で、コツコツ積み立てる」、まずはこれで十分です。金額の大きさより、早く始めて、長く続けることが、いちばん効きます。

■ iDeCo（個人型確定拠出年金）—税金を減らしながら老後の準備をする

- ・ iDeCoとは：毎月の掛金が全額「所得控除」になる老後のための積立制度、掛金が多いほど所得税・住民税が安くなります
- ・ 節税効果の例：年収400万円の人が毎月2万円積み立てると、年間約4.8万円の節税になります（所得税率10%・住民税10%の場合）
- ・ 受け取り方：60歳以降に一時金または年金として受け取れます。受取時にも税制優遇があります
- ・ 注意点：60歳まで原則引き出せません。急な出費への備え（生活費6ヶ月分の貯蓄）を確保してから始めましょう
- ・ 会社員の掛金上限：月2.3万円（企業型DCがない場合）、会社の制度を確認します。なお拠出限度や企業型DC併用ルールは2024年以降も改正が続くため、最新の限度額は国民年金基金連合会等で要確認です

■ 投資の基本原則—「損しないため」ではなく「長期で増やすため」

- ・ 長期投資：投資期間が長いほど、複利の効果と「時間分散」によるリスク低減が働きます。20年・30年の視点で考えます

- ・ 分散投資：1つの銘柄・1つの国・1つの資産に集中しない、全世界株式インデックスファンドは数千社に分散投資できます
- ・ 積立投資（ドルコスト平均法）：毎月一定額を積み立てることで、価格が高いときは少なく・安いときは多く買えます。「底値を狙う」より確実な方法です
- ・ 低コストファンドを選ぶ：信託報酬（管理費用）が年0.1%のファンドと1%のファンドでは、30年後の資産額に大きな差が生まれます
- ・ 「やってはいけないこと」：短期売買・信用取引・FX・仮想通貨の投機は投資ではなく投機です。入社後しばらくは「インデックス投資の長期積立」だけで十分です

投資には、必ず値動き（リスク）があります。大事なのは、リスクをゼロにすることではなく、「正しく怖がる」こと。短期で大儲けを狙う投機（FXや仮想通貨の短期売買など）と、長期でコツコツ育てる投資は、まったくの別物です。新人のうちは、「生活を守る現金」を先に確保し、投資は“余裕資金で、長期・分散・積立”が鉄則です。

💡 ポイント：投資より先に“現金のクッション”

投資を始める前に、まず「生活費の半年分」くらいの現金を、すぐ使える形で確保しておきましょう。急な出費や、もしものとき、この“クッション”があるだけで、無理な判断をせずに済みます。守りを固めてから、攻めに回る、これがお金の基本です。

19-7 人生の各ステージでの備え—保険と緊急予備資金

資産形成と並んで重要なのが「リスクへの備え」です。保険は「起きると困る事態」に対して支払い能力を確保するための仕組みです。必要な保険だけに加入することがポイントです。

■ 独身・入社1～3年目に必要な保険

- ・ 公的健康保険（健康保険）：会社員は健康保険に自動加入、3割負担で医療を受けられます。高額療養費制度により月の自己負担に上限があります
- ・ 傷病手当金：病気・ケガで働けなくなったとき、給与の3分の2が通算1年6ヶ月支給されます（2022年改正で「支給開始日から通算」に）、公的保険でこの保障があるため、若い独身者は医療保険の必要性が低いです
- ・ 生命保険：独身で扶養家族がいない場合、基本的に不要です。「死んだときに困る人」がいてから考えます
- ・ 緊急予備資金：生活費6ヶ月分を普通預金・定期預金で確保することが最優先、**投資より先に「生活を守る現金」を確保します**

■ ライフイベントに合わせた保険の見直し

- ・ 結婚・子どもができたとき：配偶者・子どもが収入に依存する状況になったら生命保険を検討します。定期保険（掛け捨て）が最もコスト効率が良いです
- ・ 住宅購入時：火災保険は必須（地震保険は任意ですが推奨）。団体信用生命保険（団信）で住宅ローン中の死亡・高度障害がカバーされます
- ・ 保険の見直し：ライフステージが変わるたびに保険を見直します。不要な保険に払い続けることは資産形成の機会損失です

19-8 税金の基本—社会人として知っておくべき最低限の知識

会社員として最初の数年間は「会社が税金の手続きをしてくれる」ため、税金を意識しにくいです。しかし副業・投資・転職・フリーランス化など、様々な場面で税の知識が必要になります。

■ 主要な税金と会社員との関係

- ・ 所得税：年収から控除を引いた「課税所得」に対して課される税金、会社が毎月の給与から「源泉徴収」して代わりに納めます。年末調整で精算されます
- ・ 住民税：前年の所得に対して課される都道府県・市区町村への税金、4月～翌3月分を翌年6月～翌年5月に分割して給与から天引きされます
- ・ 社会保険料：健康保険・厚生年金・雇用保険の保険料、給与から天引きされます。標準報酬月額に応じて決まります

■ 確定申告が必要になる場面

- ・ 副業収入が20万円を超えたとき：給与以外の所得が年20万円超の場合、確定申告が必要です
- ・ 年収が2,000万円を超えたとき：高収入になると年末調整の精算が不十分になります
- ・ 医療費控除：1年間の医療費が10万円を超えたとき、超過分を所得控除として申告できます
- ・ ふるさと納税：ワンストップ特例（5自治体以内）なら確定申告不要ですが、6自治体以上は確定申告が必要です

確定申告と聞くと身構えるかもしれませんが、今は会計アプリや国税庁のサイトで、ガイドに沿って進めればできるようになっています。最初は戸惑っても、一度やれば要領がつかめます。「自分で申告できる」ことは、お金の自立への大きな一歩です。

- ・ 株式等の売却益・配当：NISA以外での投資利益は確定申告が必要な場合があります（特定口座・源泉徴収ありを選べば不要）

💡 ポイント

ライフプランは「完璧に立てる」必要はありません。大切なのは「考え始めること」と「早く動き始めること」です。

NISAの口座開設・iDeCoの手続き・緊急予備資金の積み立て、どれも今日から始められます。入社1年目から少しずつ始めた積み重ねが、20年後・30年後のあなたの選択肢を大きく広げます。

▶ やってみようワーク・演習

やってみよう追加：マネープランを立てる

以下のステップで、自分の入社後の財務計画を作りましょう。

①収支を整理する：

- ・ 手取り月収の概算を確認する
- ・ 固定支出（家賃・食費・交通費・通信費等）を書き出す
- ・ 毎月いくら「使えるお金」があるか計算する

②資産形成の優先順位を決める：

- ・ 緊急予備資金（生活費6ヶ月分）を貯めるまで：毎月いくら貯める？

- ・ NISAを始める：毎月いくら・どんなファンドを買う？
 - ・ iDeCoを始める（任意）：毎月いくら掛ける？
- ③10年後の目標を設定する：
- ・ 30歳時点での貯蓄・投資残高の目標を決める
 - ・ その目標のために今日から変える「1つの行動」を決める

✦ 考えてみよう：続けたいお金の使い方

お金は「がまんするため」だけのものではありません。

あなたが「これだけは続けたい」と思うお金の使い方（趣味、好きなものへの支出、自分への投資など）を、1つ思い浮かべてみましょう。

それを無理なく続けるには、毎月いくらくらい確保しておきたいですか。

ざっくりで構いません。

お金の計画は、削ることより「大事にしたいものを続けるため」と考えると、ぐっと前向きになります。

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・ これからの毎月の固定費（家賃・通信など）をざっくり見積もり、「先取りで少し貯める」金額を考えてみる。

💡 ポイント：お金の知識は、これからの“生きる力”

お金の制度は、これからも変わっていきます。AIに聞けば調べられる時代だからこそ、「何を聞けばいいか」「何に気をつけるか」という土台が大切です。

完璧に覚えなくていい。「自分のお金は、自分で考える」、その姿勢こそが、いちばんの資産です。

19-9 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ 手取り・固定費・貯められる額の間関係を、ざっくり。
- ・ 長期でお金を増やす考え方とは。
- ・ 給料から引かれるもの（税金・社会保険）を、思い出せるだけ。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、これからの学びや仕事の中で、少しずつ

埋めていきましょう。

第20章 仕事と人生を統合する—「なぜ働くのか」を問い直す（37日目・1日）

この章のゴール

この章では、仕事と人生について、「なぜ働くのか」を自分の言葉にすることを学びます。終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ① **【言える】** 「なぜ働くのか」を、自分の言葉で一言で言える。
- ② **【考え方がわかる】** 変化に合わせて学び続ける姿勢が要る、とわかる。
- ③ **【ふり返れる】** 序章で書いた「大切にしたいこと」と今をくらべられる。

まず自己診断—読む前に、いまの自分を知ろう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「働く意味を自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
- ② 「働く意味は、立派な答えを一つに決めないといけないと思いますか？」を、いまの考えで一言だけ書きます（正解は気にしません）。

読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

20-1 「生活のため」を超えた働く動機

「なぜ働くのか」という問いに、正直に向き合ったことがありますか。

多くの人は「生活のため」「お金のため」と答えます。それは正直な答えであり、否定すべきものではありません。

しかし、長い職業人生を支えるには、それだけでは不十分なことが多いのです。

心理学者マズローは、人間の欲求を5つの階層に整理しました。

マズローの欲求5段階 — 下が満たされると上を求める

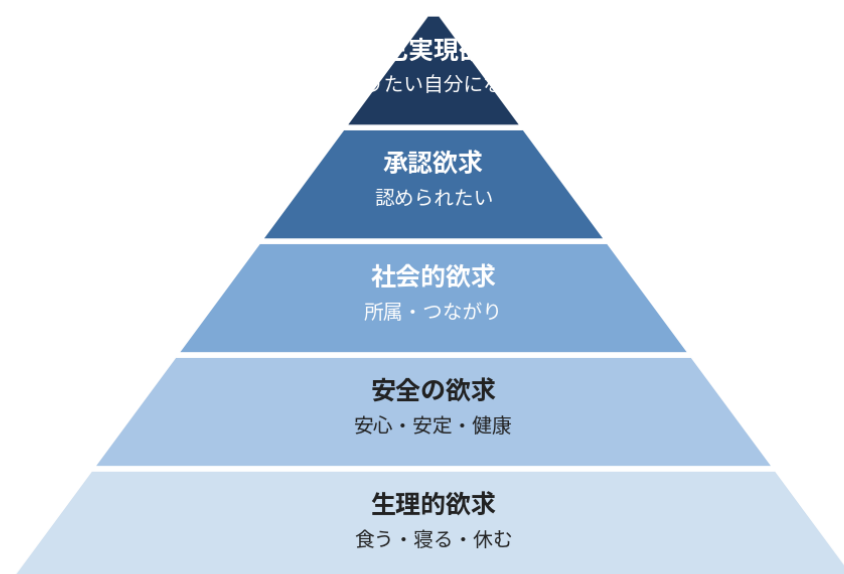


図 マズローの欲求5段階

生存・安全・所属という基本的な欲求が満たされると、人は「承認」「自己実現」を求めるようになります。

マズローの欲求段階は、順番に一段ずつ上がる階段というより、「土台が安定すると、より高い欲求に目が向く」という傾向の話です。生活の不安が大きいうちは、まず安定を求めるのが自然。だから、新卒のうちに「自己実現は？」と焦らなくて大丈夫。まず土台を整えながら、少しずつ上の問いに向き合えば十分です。

仕事は、自己実現の最も大きな舞台の一つです。

とはいえ、「仕事＝自己実現でなければならない」とも思わなくて大丈夫です。仕事は人生の大切な一部ですが、すべてではありません。仕事以外に大切なものがあること自体が、健やかさの証。仕事を“人生の主演”にも“ただの手段”にもしすぎず、自分に合った距離で付き合ってください。

働く動機の種類	特徴
外発的動機 (外からの報酬)	給与・評価・昇進・他者からの承認。安定をもたらすが、報酬が下がったり失われたりすると働く意欲が落ちやすい
内発的動機 (内からの欲求)	仕事そのものへの興味・成長の喜び・誰かの役に立つ実感。長期的な充実感・やりがいの源。外部環境に左右されにくい

どちらが良い・悪いというものではありません。

ただ、長く・楽しく・誇りを持って働き続けている人のほとんどは、

内発的動機、「これが好き」「これで誰かの役に立てる」という感覚、を育てています。

動機に「正解」はありません。「お金のため」も「成長のため」も「誰かの役に立ちたいから」も、全部同時にあっていい。しかも、それは時期によって変わります。だから、今ここで完璧な答えを出す必要はなく、「自分は今、何を大事にしているかな」と、ときどき自分に聞いてみる、それで十分です。

外発的動機（給与・評価）も、内発的動機（好き・役に立つ）も、どちらも大切で、両方あっていいのです。ただ、外発的動機だけだと、報酬や評価が止まったときに、やる気も止まってしまう。一方、「これが好き」「誰かの役に立てる」という内発的な火種を一つでも持っている、つらい時期も乗り越えやすくなります。

面白いのは、内発的動機は「与えられる」のを待たなくても、自分で育てられるということです。同じ作業でも、「この工夫で誰かが楽になる」と意味づけたり、得意なやり方を少し足したりする、こうして自分の手で仕事をやりがいのあるものに作り変えることを、ジョブ・クラフティングと呼びます。仕事を“させられるもの”から“自分のもの”に変える、小さな工夫です。

✦ 考えてみよう：あなたの“火種”

これまでの人生で、「時間を忘れて夢中になった」ことを思い出してみましょう。勉強でも、趣味でも、人の手伝いでも構いません。

その“夢中になれた要素”の中に、あなたが仕事でやりがいを感じるヒントが隠れています。

📖 事例から学ぶ：「お金のため」から始まってよい

あるエンジニアは、最初は「給料のため」と割り切って働いていました。でも、担当したシステムを使う人の「助かった」という声に触れるうち、「もっと役に立ちたい」という気持ちが芽生え、それが今の原動力になっています。動機は、後から育つこともあるのです。

→ あなたにできる一歩：今は「お金のため」で大丈夫。目の前の仕事の先にいる「使う人」を、ときどき想像してみる。

20-2 変化に適応し続ける自分をつくる

社会人人生の中には必ず「想定外の変化」が来ます。技術の変化・組織の変化・個人の変化、それらを成長の機会として捉えられる人が、長期的に活躍できます。

- ・ 組織に依存しすぎない「自分の軸」、技術力・外部ネットワーク・倫理観・学ぶ習慣

「自分の軸」は、一つに絞らなくて大丈夫です。技術の軸、人とのつながりの軸、学び続ける軸、いくつか持っておくと、どれか一つが揺らいでも、ほかが支えてくれます。新卒のうちは、まず「学び続ける軸」を一本、確実に育てておくで安心です。

- ・ 失敗・挫折を「財産」にする思考法、**失敗から何を学んだかが、その人の成長を決めます**
- ・ 変化を脅威ではなく機会として捉える習慣が、長期的なキャリアを豊かにします

もちろん、すべての変化を一人で前向きに受け止めるのは大変です。不安なときは、同期や先輩に話す、信頼できる人に相談する、それも立派な“適応する力”です。変化に強い人は、一人で抱える人ではなく、上手に人を頼れる人でもあります。

📖 事例から学ぶ：経済変動を乗り越えた人たちの共通点から学べること

バブル崩壊・リーマンショック・コロナ禍を経験しながらも成長し続けた人たちに共通するのは、「自分で考えて動く力」と「変化を学びに変える習慣」でした。

特定の技術・会社・ポジションへの依存ではなく、「変化に適応し続ける力」が本当の安定をもたらします。

変化に強い人は、「変わらないために頑張る人」ではなく、「変わることを前提にしている人」です。技術も、会社も、自分も、必ず変わります。だからこそ、特定の会社やツールにだけ依存せず、「どこでも通用する力（考える・学ぶ・人と働く）」と「学び続ける習慣」を土台にしておく。その土台があれば、変化はむしろチャンスになります。

「失敗を財産にする」とは、失敗を美化することではありません。失敗したら、まず落ち込んでいい。そのうえで、「次はどうするか」を一つだけ書き留める。この“一つだけ”の積み重ねが、同じ失敗を繰り返さない自分を作ります。失敗を隠す人は学べず、失敗を直視できる人だけが伸びます。

事例から学ぶ：変化を味方にした先輩

ある人は、長く使ってきた技術が時代遅れになりかけたとき、嘆くのではなく、半年かけて新しい分野を独学。最初は不安でしたが、「学び方」を知っていたので乗り換えられ、結果として市場価値がむしろ上がりました。

→ あなたにできる一歩：今のうちから「新しいことを学ぶ筋肉」を、小さく動かし続けておく。

20-3 序章で書いた「大切にしたいこと」と照らし合わせる

この研修を通じて、あなたはどう変わりましたか。何が変わらなかったでしょうか。

- ・ 44日前の自分が書いた「大切にしたいこと」を読み返します
- ・ 研修を通じて、新しく加わった「大切にしたいこと」はありますか
- ・ 「知識」として学んだことが「自分の価値観」として根付きましたか

▶ やってみよう—ワーク・演習

最終ワーク①：「10年後の自分への手紙」を書く 10年後の自分へ、今の自分から手紙を書きます。

「今、自分は何を大切にしているか」「10年後にどんな社会人・エンジニア・人間になりたいか」「そのために今日から何をするか」を書きます。

これは自分だけが読むものです。

最終ワーク②：「自分の行動指針」を一枚にまとめる（終章へ接続） 第10章で学んだ「正しさの4層」を自分の言葉で表現した「行動指針」を作ります。

終章での発表に向けて仕上げます。

✦ 考えてみよう：変わったこと、変わらなかったこと

44日前の自分と、今の自分。「考え方が変わったこと」を一つ、「変わらず大切にしたいこと」を一つ、挙げてみましょう。

変化に気づけることも、軸を持ち続けられることも、どちらもあなたの成長です。

20-4 ウェルビーイング—豊かな仕事人生のために

「ウェルビーイング（Well-being）」とは、身体的・精神的・社会的に良好な状態を指します。単に「病気でない状態」ではなく、人生全体が充実している状態です。仕事はウェルビーイングに大きく影響しますが、仕事だけがウェルビーイングを決めるわけではありません。

■ 働くこととウェルビーイングの関係

- ・ 仕事はウェルビーイングの重要な柱：達成感・成長・つながり・承認、これらは仕事を通じて得られる代表的なウェルビーイングの要素です
- ・ 「お金のため」の先にあるもの：マズローの欲求5段階説が示すように、生存・安全の欲求が満たされると、人は「帰属」「承認」「自己実現」を求めます。仕事はこれらすべてを満たせる活動です
- ・ 過度な仕事集中のリスク：**仕事だけがアイデンティティになると、異動・降格・転職・定年のタイミングで大きなアイデンティティクライシス起きるリスクがあります**
- ・ ワーク・ライフ・インテグレーション：「仕事とプライベートを分ける（ワーク・ライフ・バランス）」から「仕事もプライベートも自分の人生として統合する」という考え方が広がっています

■ 自分なりの「豊かさ」を定義する

「豊かさ」の定義は人によって異なります。年収・ポジション・家族・趣味・社会貢献、「何が自分にとっての豊かさか」を問い続けることが、後悔のない人生の設計につながります。

- ・ ポジティブ心理学（マーティン・セリグマン）が提唱する豊かさの5要素（PERMAモデル）：
- ・ P (Positive Emotion)：喜び・感謝・安らぎ等のポジティブな感情
- ・ E (Engagement)：何かに没頭できている状態、「フロー体験」とも呼ばれます
- ・ R (Relationships)：良質な対人関係、仕事・プライベート双方での信頼できる関係
- ・ M (Meaning)：自分の人生に意味・目的があるという感覚
- ・ A (Achievement)：目標の達成・成長の実感
- ・ 仕事を通じてこの5要素がどれだけ満たされているかを定期的に振り返ることが、働き方の設計に役立ちます

そして、豊かさは人と比べるものではありません。SNSで見る誰かの華やかな生活と比べて、落ち込む必要はない。あなたにとっての「これがあれば満たされる」を知っていること、それ自体が、いちばんの豊かさです。

ウェルビーイングは、仕事だけで作るものではありません。健康、家族や友人とのつながり、趣味や休息、これらは、仕事の支えにもなり、いざ仕事がつまづいたときの“逃げ場”にもなります。仕事に全部を賭けると、その仕事でつまづいたとき、自分の全部が崩れてしまう。複数の柱を持っておくことが、長く健やかに働くコツです。

PERMAの5要素は、覚えるためのものではなく、「自分の今の状態を点検する物差し」として使えます。最近、心が動く瞬間（P）はあったか。没頭できる時間（E）はあるか。信頼できる人（R）はいるか。意味（M）を感じられるか。達成（A）の実感はあるか。一つでも弱っていたら、そこを少し手当てする、それがセルフケアの第一歩です。

20-5 人生の意味と仕事—「何のために働くか」の問いと向き合う

この問いに「正解」はありません。しかし「自分なりの答えを持っている人」と「持っていない人」では、困難な状況での立ち回りの質が大きく変わります。

「何のために働くか」を問い続けることは、しんどいことではありません。むしろ、その問いがあるから、迷ったときに立ち返る場所ができます。答えが出なくても、問い続けている限り、あなたは自分の人生のハンドルを、自分で握っていられます。

■ 働く意味を問いつける哲学的背景

東西の哲学者・思想家が「働く意味」について様々な視点を提供しています。いくつかのフレームを知っておくことで、自分の問いを深める助けになります。

- ・ ヴィクトール・フランクル「意味への意志」：人は「意味があるから生きられる」、どんな状況でも意味を見出す能力が人間の本質です。**仕事に意味を見出せる人は、困難な状況でも前進できます**
- ・ 宮本武蔵「五輪書」の「道」：「剣の道」に象徴されるように、仕事を単なる生業ではなく「道（Way）」として捉える日本的な職業観は、長期的な技術の習得・品格の向上と深く結びついています
- ・ 孔子「論語」：「之を知る者は之を好む者に如かず。之を好む者は之を楽しむ者に如かず」、知っているだけより好きな方が、好きなだけより楽しめる方が上だという考え。仕事を「楽しめる状態」に近づけることが理想です

■ 働く動機の進化—入社1年目から始まる変化

働く動機は人生のステージによって変化します。それは成長の証でもあります。

- ・ 入社1～3年目：「会社・チームの役に立ちたい」「早く一人前になりたい」、外発的・即時的な動機が中心です
- ・ 3～7年目：「自分の専門性を活かしたい」「この分野で価値を生み出したい」、内発的動機が育ってきます
- ・ 10年以上：「自分の仕事が社会や次世代に何を残せるか」、貢献・レガシーへの意識が深まります
- ・ 重要：動機の変化を「以前の自分とのギャップ」として否定しない、成長と環境の変化によって動機が変わることは自然です

動機が変わるのは、あなたが成長し、見える世界が広がった証拠です。1年目に「役に立ちたい」と思っていた人が、数年後に「この分野を究めたい」と思うようになる、それは“ブレた”のではなく、“育った”のです。変化を、前向きに受け止めてください。

「働く意味」は、難しく考えなくても、ふとした瞬間に見つかります。自分の作ったものを誰かが使ってくれた、「ありがとう」と言われた、困っていた人が助かった、そうした小さな実感の積み重ねが、「なぜ働くのか」の答えを、少しずつ形づくっていきます。今すぐ立派な答えを出す必要はありません。

フランクルや孔子の言葉を、難しく受け取らなくて大丈夫です。要するに、「意味を感じられると、人は頑張れる」「好きでやれると、もっと続く」というシンプルな話。自分の仕事の中に、ほんの少しでも「意味」や「好き」を見つけられると、毎日が変わります。

20-6 「正しく・誠実に・よく」生きるということ—44日間の学びの統合

この研修で学んできたことを、最後に一度統合して振り返ります。技術スキル・法律知識・ビジネスマナーはすべて手段です。「何のために」それらを使うかが、あなたの仕事の価値と人生の質を決めます。

■ この研修で学んだことの本質

- ・ 報連相（第4章）、情報を正直に・早く・正確に伝えることで、チームと社会を機能させます
- ・ 品質（第5章）、「動けばいい」ではなく「価値を届ける」設計が、プロとしての誇りです
- ・ ハラスメント（第12章）、人を傷つけない。人を守る。それが組織と社会の健全性を保ちます

- ・ 情報を扱う責任（第14章）、アクセスできる力には責任が伴います
- ・ 働く権利（第16章）、自分の権利を知ることが、長く健康的に働く基盤です
- ・ 正しさの4層（第10章）、迷う場面で「人としての正しさ」を最優先することが、後悔のない判断をもたらします

■ 「良い仕事人」であることと「良い人間」であることは同じ

この研修で繰り返し伝えてきたメッセージは、「技術力と倫理観は別物ではない」ということです。

- ・ 信頼は毎日の小さな積み重ねで作られます。一度の大きな成果より、毎日の誠実な行動の方が信頼を築きます

この44日間で学んだことの多くは、すぐには使わないかもしれませんが、でも、配属後に「あ、これか」と思い出せば十分です。研修は、答えを暗記する場ではなく、これから出会う問いに備えて“引き出し”を作る場でした。困ったら、いつでもこのテキストに戻ってきてください。

- ・ 失敗は財産です。問題は失敗することではなく、失敗から学ばないことです。失敗を隠さず、正直に向き合い、次に活かすことが成長です
- ・ **「わからない」は最強の武器です**、「わからない」と言える誠実さが、本当の信頼と深い学びをもたらします
- ・ あなたの仕事は社会とつながっています。どんな小さなコードも、どんな小さなサポートも、誰かの生活に影響しています。その意識が、仕事への誇りと責任感を育てます

✦ 考えてみよう：あなたの仕事の“向こう側”

あなたが将来つくるもの・支えるものの“向こう側”には、それを使う誰かがいます。その人は、どんな場面で、あなたの仕事に助けられるのでしょうか。

顔の見えない「ユーザー」を、一人の人として想像してみましょう。それが、働く意味の手がかりになります。

📖 事例から学ぶ：仕事と人生を統合して生きた人から学べること

日本のソフトウェアエンジニアリングの先人たちは、技術への情熱と社会への責任感を両立させながら、日本のIT産業の基盤を作りました。

また、世界的に見ても、Linuxカーネルを開発したリーナス・トーバルズは「自分が使いたいものを作る」という純粋な動機から出発し、現在は世界中のサーバ・スマートフォン・自動車のOSとなる世界的な影響力を持つソフトウェアを生み出しました。

この教訓から学べること：「自分が正しいと思うことを、誠実に、継続して行う」、これが仕事と人生を統合する最もシンプルな原則です。

💡 ポイント

この研修を終えるあなたへ、必要な「引き出し」は、もう持っています。

あとは、現場で少しずつ取り出していくだけです。

44日間で学んだことは、すべて「人としての正しさ」という一つの軸に収束します。

難しい判断に迷ったとき、誰も見ていない場面での選択、チームが困っているとき、その都度、自分

の軸を思い出してください。
あなたの仕事は社会とつながっています。
誠実に、誇りを持って、よく働いてください。

▶ やってみようーワーク・演習

最終ワーク追加：「自分の職業哲学」を書く

以下の問いに対して、自分の言葉で答えを書いてみましょう。（正解はありません）

- ①「なぜ働くのか」、あなた自身の言葉で教えてください
- ②「どんなエンジニアでありたいか」、技術だけでなく、人間として・社会人として
- ③「自分の仕事が10年後の誰かの生活にどう影響するか」、想像してみましょう
- ④「この研修で最も心に残ったこと」、それはなぜ心に残ったのか
- ⑤「明日から一つだけ変える行動」、具体的に決めてください

書き終えたものを封筒に入れて、1年後に開封する「タイムカプセル」にしましょう。

✦ 考えてみよう：「働いてよかった」と思う瞬間

「なぜ働くのか」は、立派な答えを今すぐ出さなくて大丈夫です。

代わりにこう想像してみましょう。あなたが「この仕事をしていてよかった」と感じるのは、どんな瞬間だと思いますか。

お金以外で1つ挙げてみてください。

「ありがとうと言われた」「できなかったことができるようになった」、どんな小さなことでも構いません。

その感覚が、これから働くうえでの、あなたなりの羅針盤になります。

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・「自分はなぜ働くのか」を、いまの言葉で一言だけ書いてみる。

そして「なぜ」を持つことは、AIの時代にますます効いてきます。AIは「どうやるか（手段）」をどんどん肩代わりしてくれます。だからこそ、「何のためにやるのか（目的・意味）」を持っている人が、これからますます強くなります。あなた自身の「なぜ働くのか」は、AIには決められない、あなただけの羅針盤です。

💡 ポイント：答えは、変わっていい

「なぜ働くのか」の答えは、一度決めたら固定するものではありません。年齢や経験とともに、変わっていったって当然です。

大切なのは、ときどき立ち止まって、自分に問い直すこと。問い続けること自体が、あなたの人生を、あなたのものにしていきます。

20-7 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ 「なぜ働くのか」を、自分の言葉で一言で。
- ・ これから大切になる姿勢は何か。
- ・ 序章で書いたことと、今の自分は違うか。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、これからの学びや仕事の中で、少しずつ埋めていきましょう。

第7部「特論：新卒社員向け ITエンジニア入門 –エンジニアとしての入口に立つために知っておくべきこと」（38～43日目・6日間）

第7部では、ソフトウェアエンジニアという専門職に固有のテーマ、あるべき姿・倫理・体系・開発プロセス・品質・セキュリティ・伝える力・法律・生成AI・成長、を扱います。ここまで築いたビジネスパーソンとしての土台の上に、専門家としての土台を体系的に重ねます。各章は〈概要 → 歴史と背景 → 詳細解説 → コラム → 討議〉の順で進み、巻末には自習用の特別資料A～Hを置きます。

ここから先は、エンジニアという専門職の世界に踏み込みます。とはいえ、暗記大会ではありません。個々の技術は、やがて移り変わります。だからこそ、変わらない「考え方の地図（体系）」を持つことを目指します。地図を持つ人は、新しい技術が出てきても、それが地図のどこにあるのかをすぐにつかめます。**AI時代にこそ、この地図が効きます。**

第21章 本当のエンジニアとは—あるべき姿・倫理・必須スキル（38日目）

🎯 この章のゴール

この章では、「本当のエンジニアとは何か」、「作業」と「仕事」の違い、エンジニアに大切な心がまえ、そして「できる」と「やってよい（倫理）」の違いを学びます。

終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ① **【違いがわかる】** 「作業」と「仕事」（作業者とエンジニア）の違いを、自分の言葉で一言で言える。
- ② **【思い出せる】** エンジニアに大切な“心がまえ”を、3つ思い出して言える。
- ③ **【考え方がわかる】** 「できる」と「やってよい（倫理）」は違う、ということがわかる。

💡 まず自己診断—読む前に、いまの自分を知ろう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「作業者とエンジニアの違いを自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
- ② 「「作業」と「仕事」のいちばん大きな違いは何か」を、いまの考えで一言だけ書きます（正解は気にしません）。

読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

21-1 概要—「作業」ではなく「仕事」をする／この章で目指すエンジニア像

この特論の最初の章では、技術の中身に入る前に、もっとも大切な問いを置きます。「**本当のエンジニアとは何か**」です。

ここまでの研修で、皆さんは「作業」ではなく「仕事」をする人になろう、という言葉は何度も聞いてきました。第7部でもこの問いは消えません。むしろ、技術を学べば学ぶほど重みを増します。なぜなら、技術は「作業」を速くする道具にもなれば、「仕事」の質を上げる土台にもなるからです。同じツールを持っても、言われた通りに手を動かすだけの人と、目的を理解して最善を選ぶ人とは、生み出す価値がまったく違ってきます。

この違いを、本書では「**作業者**」と「**エンジニア**」と呼び分けます。最初に強調しておきたいのは、これは**個人の優劣の話ではない**ということです。誰かが怠けているからでも、先輩や会社が悪いからでもありません。多くの場合、これは仕事の任せ方・教わり方・評価のされ方といった**構造の問題**です。だからこそ、構造を理解し、自分の意志で「エンジニア」の側に立つことができます。本章はその第一歩です。

この章で目指すエンジニア像を、先に一枚で示しておきます。

- ・ **目的を理解して動く**、「何を作るか」だけでなく「なぜ作るか」を問える
- ・ **専門職としての責任を持つ**、「できる」と「やってよい」を区別できる（倫理）
- ・ **体系で考える**、場当たりでなく、先人が築いた知識の体系を使える
- ・ **学び続ける**、技術の寿命が短い時代に、陳腐化しない力を育て続ける

そしてこの特論全体を貫く中心となる考えが、次の一文です。

💡 ポイント

AI時代にこそ、ソフトウェアエンジニアリング（体系）が効く。

コードを書く作業の多くをAIが助けてくれる時代だからこそ、「何を作るべきか」「その品質でよいか」「どこに任せ、どこを自分で判断するか」を決める力、つまり体系に裏打ちされた判断力、が、エンジニアの価値の中心になります。本章はその土台となる、エンジニアの「あるべき姿・倫理・必須スキルの全体像」を扱います。

なお、この特論は**知識を体系的に頭に入れる座学**です。コードを書く・ツールを操作するといった実技は別科目やOJTで鍛えます。本章で身につけるのは、技術を学んでいくための「見取り図」と「構え」です。

21-2 歴史と背景—「プログラマ→SE→ソフトウェアエンジニア」の変遷と技術者倫理の成立

ソフトウェアを作る職業の呼び名は、時代とともに変わってきました。ごく大づかみに言えば、命令を書き写す「**プログラマ**」から、システム全体を設計し顧客と合意する「**システムエンジニア (SE)**」へ、そして開発・運用・保守までを工学として体系的に担う「**ソフトウェアエンジニア**」へと、担う責任の範囲が広がってきた、という流れです。

その背景には、1968年にソフトウェア工学 (software engineering) という言葉が広まり、「作る」を**規律ある工学**にしようという意志が生まれたことがあります（第23章で扱います）。技術が社会に大きな影響を持つようになると、専門職としての**倫理綱領**も成文化されていきました。これが「技術者倫理」の系譜です。

押さえておきたいのは、呼び名や肩書きそのものより、「**責任の範囲が広がってきた**」という**方向性**です。皆さんが立つのは、その延長線上です。

21-3 あるべきエンジニア像とプロフェッショナルリズム

■ そもそも「エンジニアリング」とは何か

ソフトウェアエンジニアリングには、広く使われてきた定義があります。国際的な用語標準 (IEEE の用語集、およびそれを引き継ぐSWEBOK) では、おおむね次のように説明されます。

💡 ポイント

ソフトウェア工学とは、ソフトウェアの開発・運用・保守に対して、**体系的で、規律があり、定量的なアプローチ**を適用すること。

ここで大事なのは「体系的・規律ある・定量的」という3語です。思いつきや勘ではなく、**先人が積み上げた知識体系に沿って、再現できるやり方で、測りながら作る**、これがエンジニアリングの核です。アート（個人の感性に依る創作）とは異なり、エンジニアリングは**他者と共有でき、検証でき、改善できる**ことを目指します。

■ プロフェッショナルリズム—新卒1年目から「プロ」である

第1部で学んだ通り、プロフェッショナルとは役職ではなく**姿勢**です。エンジニアにおけるプロ意識は、次のような態度に表れます。

- ・ **成果に責任を持つ**：「やってみた」で終わらず、相手の期待に応える結果まで引き受ける

- ・ **約束を守る**：見積もり・納期・品質について、できることとできないことを誠実に伝える
- ・ **自分の判断を説明できる**：「なぜそうしたか」を言葉にできる（できないものは、まだ自分のものになっていない）
- ・ **学び続ける**：知らないことを認め、調べ、人に聞き、身につける
- ・ **境界を守る**：「できる」ことと「やってよい」ことを区別する（後述の倫理）

これらは特別な才能ではなく、**初日から選べる構え**です。

■ エンジニアの5つのマインドセット

あるべきエンジニア像は、技術スキル以前の「**態度・価値観**」に表れます。ソフトウェア技術者論でしばしば挙げられるのが、次の5つのマインドセットです（クラフツマンシップ＝職人気質の議論などで整理されてきたもの）。

1. **自走力（オーナーシップ）**、指示を待つのではなく、自分の責任範囲を自分で前に進める
2. **価値への貪欲さ**、コードを書くこと自体でなく、その先の**顧客価値**にこだわる（第25章の「品質＝顧客価値」と直結）
3. **成長への欲求**、誰かに言われなくても、自ら学び続けようとする
4. **チーム貢献**、レビューやメンタリングなど、チーム全体の成果に貢献する（T型人才）
5. **技術的好奇心**、「面白そう」という内発的な動機で、新しい技術へ手を伸ばす

💡 ポイント

なお本特論では、これらの態度を日々呼び戻すための「合言葉」として、**目的志向／当事者意識／誠実さ／体系で考える／学習者であり続ける**という言い方も繰り返し使います。

上の5つのマインドセット（あるべき態度）を、現場の判断の場面で思い出すための平易な標語、と捉えてください。

両者は別物ではなく、同じ「態度」を、原理（マインドセット）と日々の合言葉という二つの言い方で示しているだけです。

21-4 技術者倫理の基礎—綱領と誠実さ

■ なぜエンジニアに倫理が必要か—「できる」と「やってよい」は違う

エンジニアは、人々の生活や安全、財産、個人情報に深く関わるシステムを作ります。技術的に「できる」ことと、社会的・倫理的に「やってよい」ことは、必ずしも一致しません。**力（技術）には責任が伴う**、これが技術者倫理の出発点です。

倫理には2つの面があります。

- ・ **予防倫理**：してはいけないことを避ける（不正・隠蔽・手抜き・データの無断利用など）
- ・ **志向倫理**：よりよい社会のために、なすべきことを積極的に目指す

新卒のうちはまず予防倫理（やってはいけない一線を守る）を確実に身につけ、経験とともに志向倫理（社会に資する技術の使い方）へと視野を広げていきます。この志向倫理は、終盤の第30章（社会・環境への責任）でもう一度具体的に回収します。

■ 代表的な倫理綱領

専門職としての倫理は、国内外の学会・団体が**綱領**として成文化しています。代表的なものを押さえておきましょう（本文では概要にとどめ、詳細は巻末Fを参照）。

- ・ **ACM/IEEE-CS ソフトウェア工学倫理・専門職実践綱領**（ACM＝米国計算機学会、IEEE-CS＝IEEEコンピュータ部会。両者が共同で1999年に公表、version 5.2）。ソフトウェア技術者向けに作られた綱領で、次の**8原則**から成ります。

1. **公衆（PUBLIC）**、公共の利益を最優先する
 2. **依頼者・雇用者（CLIENT AND EMPLOYER）**、公益と矛盾しない範囲で依頼者・雇用者の最善の利益のために行動する
 3. **製品（PRODUCT）**、成果物と関連する改変が高い専門基準を満たすようにする
 4. **判断（JUDGMENT）**、専門的判断の誠実さと独立性を保つ
 5. **管理（MANAGEMENT）**、倫理的な開発・保守のマネジメントを推進する
 6. **専門職（PROFESSION）**、専門職としての評判と公益を高める
 7. **同僚（COLLEAGUES）**、同僚に公正であり、支え合う
 8. **自己（SELF）**、生涯にわたり学び続け、倫理的に実践する
- ・ **ACM 倫理・職業行動綱領**（一般版。2018年6月改訂）。AI・自動化・社会的影響など現代的論点を反映。
 - ・ **情報処理学会 倫理綱領**（日本。2022年6月改定）。日本の情報処理技術者の行動規範。

綱領は丸暗記するものではありません。「**迷ったときに立ち返る共通の物差し**」として存在を知り、必要なときに読み返せることが大切です。なかでも第1原則「公衆の利益を最優先する」は、すべての判断の土台になります。

■ 多様性と包摂

エンジニアが作るものは、多様な人々に使われます。年齢・性別・障害の有無・言語・文化が異なる人にとって使えるか（アクセシビリティ）、特定の集団に不利益や偏見を生まないか、こうした視点も、現代の技術者倫理に含まれます。「自分にとって使いやすい」が「みんなにとって使いやすい」とは限らない、という想像力を持つことです。

21-5 迷ったときの判断—説明責任と倫理的意思決定

■ 板挟みは必ず起こる

現場では、倫理的な迷いが必ず生まれます。たとえば、納期に間に合わせるため、テストを省いて出荷したい雰囲気がある。客先で「このデータ、ついでに別の用途にも使っていい?」と聞かれる。先輩のコードに重大な不具合を見つけたが、指摘しづらい。こうした**板挟み**は、能力ではなく状況の問題として誰にでも起こります。

■ 倫理的意思決定の手順

迷ったとき、感情や空気だけで決めないための簡単な手順があります。

1. **事実を確認する**、何が起きているか、推測と事実を分ける
2. **関係者を洗い出す**、誰に影響するか（利用者・顧客・チーム・社会）
3. **選択肢を挙げる**、「やる/やらない」だけでなく中間案も
4. **原則に照らす**、綱領（とくに「公衆優先」）や社内ルールと突き合わせる
5. **説明できるかを問う**、「この判断を、後で堂々と説明できるか」

最後の「説明責任」がとくに重要です。**自分の判断を、根拠とともに第三者に説明できるか**、これがプロの判断の試金石になります。

■ ひとりで抱え込まない—相談という技術

新卒のうちは、自分だけで結論を出す必要はありません。むしろ、**早めに相談すること自体がプロの行動**です。「これは相談すべきか」を判断する基準は単純で、**少しでも倫理・安全・法律・顧客との約束に関わると感じたら、迷わず上長や先輩に共有する**、これで十分です。抱え込んで一人で背負うことは、誠実さでも責任感でもありません。早く声を上げた人が、結果としてチームと自分を守ります。

なお、組織として不正を見つけたときに守られる仕組みとして、日本には**公益通報者保護法**があります（2022年6月施行の改正で、一定規模以上の企業に通報対応体制の整備が義務づけられました。さらに2025年に改正法が成立し、通報を理由とする不利益取扱いへの刑事罰の新設などを盛り込んだ改正が2026年12月に施行される予定です）。新卒がいきなり使う制度ではありませんが、「正しいことを伝えた人が守られる仕組みがある」と知っておくことは、誠実さを支えてくれます。

21-6 必須スキルの全体像—テクニカル/ヒューマン/コンセプチュアル

ここからは、エンジニアに必要な「スキルの全体像」を描きます。細かい技術の習得は別科目とOJTに譲り、本章では**何を伸ばしていくべきか**を俯瞰して押さえます。

■ 3種類のスキル

仕事に必要なスキルは、古くから3つに分けて整理されてきました（経営学者ロバート・カッツの古典的な整理）。

- ・ **テクニカルスキル（技術力）**、プログラミング、設計、テスト、ツールの操作など、専門の「手」の力
- ・ **ヒューマンスキル（対人力）**、報連相、傾聴、合意形成、チームでの協働など、人と働く力
- ・ **コンセプチュアルスキル（概念化力）**、物事を構造でとらえ、本質を見抜き、抽象化して考える力

新卒のうちはテクニカルスキルに目が向きがちですが、現場で評価される人は**3つのバランス**が取れています。とくにヒューマンスキルとコンセプチュアルスキルは、技術が陳腐化しても価値が落ちにくい力です。

■ T型人才を目指す

理想は「**T型**」の形です。横棒は**広く浅い知識の幅**（IT全般を見渡せること）、縦棒は**一つ以上の深い専門**（これなら任せろという領域）。新卒の最初の数年は、まず横棒を広げて全体像をつかみ（この特論がまさにその役割を担います）、並行して縦棒を一本作っていく、この順序が遠回りに見えて確実です。

21-7 主要BOKの見取り図—机に常備する3大BOK

エンジニアの世界には、先人が知識を体系的にまとめた「**知識体系（BOK：Body of Knowledge）**」があります。一人の経験には限界がありますが、BOKを使えば、業界全体が積み上げてきた知恵を借りられます。「**困ったらBOKに立ち返る**」、これがエンジニアの強力な習慣です。

新卒が知っておくべき「3大BOK」は次の通りです。

- ・ **PMBOK（ピンボック）**、プロジェクトマネジメントの知識体系。**進め方**（計画・スケジュール・リスク・ステークホルダー管理など）をまとめたもの。
- ・ **SWEBOK（スウェボック）**、ソフトウェアエンジニアリングの知識体系。**作り方**（要求・設計・構築・テスト・保守など）をまとめたもの。第23章で詳しく扱います。

- ・ **SQuBOK (スクボック)**、ソフトウェア品質の知識体系（日本発）。**品質の作り込み方**をまとめたもの。第25章で詳しく扱います。

今すべてを読む必要はありません。「**こういう体系が存在する**」と知り、必要になったときに開ける、それが第一歩です。本特論は、この3大BOKを新卒向けに案内する旅でもあります。

21-8 コンピュータサイエンス基礎の全体像

エンジニアの土台には、**コンピュータサイエンス (CS) の基礎**があります。これも一気に深掘りはせず、「どんな領域があるか」の全体像を押さえます。国際的なCS教育の標準ガイド (CS2023) も、CSを十数の知識領域に整理しています。新卒がまず見取り図として持っておきたい領域を、噛み砕いて挙げます。

- ・ **アルゴリズムと計算量**、同じ処理でも「速い/遅い」「重い/軽い」がある。なぜかを考える土台
- ・ **データ表現**、0と1、文字コード（文字化けの正体）、数値の扱いなど、コンピュータが情報をどう持つか
- ・ **ネットワークとWeb**、データが届く仕組み、HTTP、クライアントとサーバ
- ・ **データベース (DB)**、データを安全・効率的に貯め、取り出す仕組み
- ・ **オペレーティングシステム (OS)**、プログラムが動く土台。メモリ・プロセス・ファイル
- ・ **並行・分散**、同時に動く・複数の機械で動くときの難しさ（現代システムの核心）
- ・ **クラウドの類型**、IaaS/PaaS/SaaS（インフラ/開発実行基盤/ソフトウェアを、それぞれサービスとして借りる形態）など、どこまでを自前で持ち、どこから借りるかの違い

これらは「知っている」と、起きている現象の理由が分かる」基礎です。深掘りは基本情報技術者試験 (FE) などのシラバスや別科目に委ね、本章では**全体を見渡せること**を目標にします。

21-9 自走する力—調べる・読む・学ぶ

技術の世界では、誰も全部は知りません。だからこそ、**分からないことを自分で調べ、学び、進める力 (自走力)** が決定的に重要です。これは才能ではなく、訓練できる技術です。

- ・ **調べる**、公式ドキュメント・一次情報にあたる習慣。検索やAIの答えを鵜呑みにせず、出典を確認する
- ・ **読む**、他人のコードや設計書、エラーメッセージを「読んで理解する」力。書く力より先に伸ばしたい
- ・ **学ぶ**、新しい技術を、目的から逆算して効率よく身につける（学び方は第30章で深めます）

ここでもAIは強力な助けになりますが、**AIは「調べる」を速くしてくれても、「何を確かめるべきか」は人間が決める**という関係を忘れないでください。自走力の中心は、最後まで自分の頭で確かめる姿勢です。

21-10 問題解決とデバッグの思考

エンジニアの日常は、問題解決の連続です。とくに「動かない・思った通りにならない」を切り分ける**デバッグの思考**は、すべての土台になります。実装は別科目で鍛えますが、考え方はここで押さえます。

- ・ **現象を正確につかむ**、「動かない」ではなく「何を入れたら、何が起き、何を期待したか」を言葉にする
- ・ **仮説を立てる**、原因の候補を挙げる

- ・ **切り分ける**、一度に一つだけ変えて、原因の範囲を狭める（あれこれ同時に変えない）
- ・ **検証する**、仮説が正しいか確かめ、違えば次の仮説へ
- ・ **記録する**、分かったこと・試したことを残す（次の自分とチームのために）

この「仮説→検証」の往復は、デバッグだけでなく、設計・調査・トラブル対応など、あらゆる場面で使える汎用の思考法です。

21-11 段取る力ー分解・見積もり・優先順位

最後は、仕事を**段取る力**です。大きな仕事をそのまま抱えると、人は固まってしまう。プロは仕事を扱える大きさに分け、順番をつけ、見通しを立てます。

- ・ **分解する**、大きなタスクを、半日～1日でできる小さな単位に割る
- ・ **見積もる**、各タスクにどれくらいかかるかを、根拠とともに見立てる。外れても、見積もりと実績の差から学べる（見積もりの誠実さは第23章でも扱います）
- ・ **優先順位をつける**、「重要×緊急」で並べ、まず重要なものから着手する
- ・ **見通しを共有する**、「いつ・何が・どこまで」を先に伝える。遅れそうなら**早く**言う

📖 事例から学ぶ：「半日で終わるはず」が3日かかる

新卒が最初につまずくのが、見積もりの甘さと抱え込みです。

「設定を1か所変えるだけ、半日で終わる」と思った作業が、いざ始めると関連箇所が芋づる式に出てきて、影響範囲の調査・修正・確認に3日かかった、現場でよくある話です。

これは能力の問題というより、**目に見えていなかった作業（前提の調査・他人への確認・テスト・手戻り）を見積もりに入れ忘れた**ために起きます。

だから見積もりは「分解してから」行い、少しでも雲行きが怪しくなったら、抱え込まずに**早めに状況を共有する**。

外れた見積もりは、実績との差を記録して次に活かせば、立派な学びの材料になります。

段取りは「作業」を「仕事」に変える技術です。同じ手を動かすにも、段取りがある人は、迷いと手戻りが格段に少なくなります。

【コラム】受託のビジネス構造（人月・稼働率・原価率）／客先データと倫理綱領

ここまで述べたエンジニア像や倫理は、業種を問わない普遍的な話です。最後に、皆さんが多く関わることになる**受託開発（SI）**の現実に、少しだけ接続しておきます。

受託は「時間（人の働き）」を売るビジネスだという面があります。よく出てくる言葉を押さえておきましょう。

- ・ **人月（にんげつ）**、「一人が1か月働く量」を単位に、開発規模や費用を見積もる考え方。「3人月の案件」なら、おおよそ一人で3か月、三人で1か月分の仕事量、という目安です。
- ・ **稼働率**、エンジニアの働く時間のうち、売上につながる業務に充てられている割合。
- ・ **原価率**、売上に対して、人件費などの原価が占める割合。

これらは経理の話に見えて、実は皆さんの仕事と直結します。たとえば「見積もりを誠実に行う」ことは、人月の前提を崩さないための倫理でもあります。「手戻りを減らす段取り」は、稼働を価値に変える行為です。**作業を速くこなすだけの人は人月の数字にしかありませんが、品質と段取りで手戻りを減らすエンジニアは、同じ人月でより大きな価値を生みます。**ここに「作業者とエンジニアの差」が、ビジネスの数字としても表れます。

もう一つ、受託で必ず効いてくるのが**倫理**です。受託では、**客先の本番データや機密情報**に触れる

機会が少なくありません。「技術的にアクセスできる」と「見てよい・使ってよい」ことは別だ、21-4で学んだ「できる≠やってよい」が、ここで現実の重みを持ちます。客先データの目的外利用、無断のコピー、SNSへの不用意な投稿などは、信頼と契約を一瞬で壊します。倫理綱領の第1原則「公衆の利益を最優先する」と、契約上の**守秘義務（NDA：Non-Disclosure Agreement＝秘密保持契約）**は、受託エンジニアにとって絵空事ではなく、日々の判断基準です。

💡 ポイント

受託の構造は本章の主演ではありません。詳しい契約や法律は第28章で扱います。ここでは「普通のエンジニア像が、受託の数字と倫理にそのまま効いてくる」ことだけ感じ取ってください。

21-12 考えてみよう（グループ討議）

教室内では、手を動かす実技は行いません。次のテーマについて、3～4人のグループで話し合い、考えを言葉にしてみましょう。**正解を当てる場ではなく、自分の経験と言葉で語ることが目的**です。

✦ 考えてみよう：討議1：「作業で終わった/仕事にできた」経験

これまでの人生（アルバイト・部活・学業・この研修を含む）で、

- ・ 「言われたことをこなしただけ（作業）」で終わってしまった経験
- ・ 「目的を考えて、自分なりに価値を足せた（仕事）」と感じた経験

をそれぞれ一つずつ思い出し、**何が両者を分けたのか**をグループで共有してください。本章の「5つのマインドセット」のどれが効いていたか（あるいは欠けていたか）も照らし合わせてみましょう。

✦ 考えてみよう：討議2：今いちばん足りない力

21-6～21-11で挙げたスキル（テクニカル／ヒューマン／コンセプチュアル、自走力、問題解決、段取り）のうち、

- ・ **自分が今いちばん足りない**と感じる力はどれか
- ・ それを伸ばすために、明日から始められる**小さな一歩**は何か

を一人ずつ宣言し、グループでフィードバックし合ってください。

✦ 考えてみよう：（倫理ミニケース）相談すべきか考える

次のケースについて、21-5の手順（事実→関係者→選択肢→原則→説明できるか）で考えてみましょう。

締め切り当日。先輩から「テストは一部省いていいから、とにかく今日リリースして」と言われた。あなたはテスト不足で重大な不具合が残る可能性に気づいている。

このとき、誰に・何を・どう伝えるか。

「黙って従う」「黙って自分で全部直す」以外の中間案はないか。

グループで複数の選択肢を出してみてください。

なお、このケースで指示を出した先輩自身も、上からの納期圧力に板挟みになっているのかもしれない。

誰かを責めるためではなく、構造ごと安全な方へ動かすために、どう相談を切り出すかという視点で考えると、現実に近い答えが見えてきます。

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・ 仕事を頼まれたら、手をつける前に「これは何のためにやるのか（目的）」を一度たしかめる。

21-13 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ 「作業」と「仕事」（作業者とエンジニア）のいちばん大きな違いを、一言で。
- ・ エンジニアに大切な“心がまえ”を、思い出せるだけ（3つ書ければ十分）。
- ・ 「できる」と「やってよい」は、なぜ違うのか。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、次の章やこれからの仕事の中で、少しずつ埋めていきましょう。

第22章 IT産業の現在地と未来（39日目）

🎯 この章のゴール

この章では、自分が働くIT業界が「いまどんな形か」、そして「これからどう変わるか」を学びます。

終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ① **【イメージできる】** 自分が働くIT業界がいまどんな形か（受託が多いなど）を、ざっくり言える。
- ② **【理由がわかる】** 技術はすぐ古くなる→だから「学び続ける力」が大事、という理由がわかる。
- ③ **【知っている】** 受託からでも伸びる道がある、と知っている（1つ言える）。

💡 まず自己診断—読む前に、いまの自分を知らう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「いまのIT業界がどんな形かを自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
 - ② 「IT業界で働く人は、どんな会社が多いと思いますか？」を、いまの考えで一言だけ書きます（正解は気にしません）。
- 読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

22-1 概要—なぜ業界の現在地を知るのか

自分がどの船に乗り、その船がどの海を、どちらへ進んでいるのか、それを知らずに漕ぎ続けるのは不安なものです。本章では、皆さんが身を置く**IT産業の現在地と方向**を概観します。

業界を俯瞰して見る意味は3つあります。第一に、**自分の仕事が産業構造のどこにあるか**を理解できること。日々の作業に追われると、自分が大きな流れのどこにいるのかを見失いがちです。第二に、**産業がどちらへ向かっているか**を踏まえて、学ぶことを選べること。流れに逆らう努力より、流れを読んだ努力のほうが報われます。第三に、変化に振り回されるのではなく、**変化を読んで備えられる**ことです。

ただし、本章で示すのは大づかみな見取り図です。産業の細部や個別企業の事情までは扱いません。新卒のうちは、まず「**全体としてどうなっていて、どこへ動いているか**」を感覚としてつかむ、それで十分です。細部は、現場で働きながら肉付けしていきましょう。

💡 ポイント

なお、本章で挙げる数値・予測は「〇年時点」のものです。産業の状況は動きます。大切なのは個々の数字の暗記ではなく、**構造と方向を読み取る視点**を持つことです。

22-2 歴史と背景—情報処理の加速

コンピュータが実用化されて以来、情報処理の世界は**加速**を続けてきました。計算能力は長年にわたって指数的に向上し（半導体の集積度が一定期間ごとに倍増していくという経験則が長く語られてきました）、かつては高価な大型計算機でしか動かなかった処理が、今や手元の端末やクラウドで当たり前になります。記憶容量も通信速度も同様に桁違いに伸び、この「速く・安く・大きく」の流れ

が、産業の前提を何度も塗り替えてきました。

この加速は、単に「機械が速くなった」だけの話ではありません。性能の向上が、新しい使い方を可能にし（大量データの分析、動画配信、そして近年の生成AIなど）、新しい使い方が新しいビジネスと職種を生み、それがさらなる需要と投資を呼ぶ、という連鎖を生んできました。つまり、技術の加速は産業構造の変化と一体です。だから、技術の動向を知ることは、産業の行方を読むことと切り離せません。

象徴的なのが、**技術や知識の「寿命」の短縮**です。かつては、一度習得した技術や製品知識が、何十年も通用した時代もありました。しかし今は、主要な技術やツールの入れ替わりが速く、「**学んだ技術の賞味期限が数年**」とも言われます（年数は概説であり、分野により大きな幅があります。基盤的な知識ほど寿命が長く、特定の製品・流行ほど短い傾向があります）。

この「加速」は、本特論全体を貫く考えと直結します。個々の技術はすぐ古くなる。だからこそ、**陳腐化しにくい体系・考え方・学び方**（第23章・第30章）が効く、という構図です。歴史が教えるのは、「速くなり続けてきた」という事実と、「これからも速くなる」という見通しです。であれば、新卒が身につけるべきは「今いちばん新しい技術」そのものより、**新しい技術を次々に取り込んでいける土台**だと分かります。

もう一つ、歴史から見えるのは「**抽象化の積み重ね**」です。機械語からアセンブリ、高水準言語、フレームワーク、クラウド、そしてAIによる支援へと、エンジニアはより低レベルな作業を道具に任せ、より高い視点で考えられるようになってきました。各時代で「これで楽になる」と言われ、実際に生産性は上がってきましたが、それでエンジニアが不要になったわけではありません。**任せられる作業が変わり、人間が考えるべきことがより上流・本質へ移ってきた**のです。生成AI（第29章）も、この長い流れの最新の一步として捉えると、冷静に向き合えます。

22-3 産業の構造—Slerピラミッドとバリューチェーン

■ 日本のIT産業の際立った特徴—人材の偏在

日本のIT産業には、世界と比べて際立つ特徴があります。**IT人材の所属の偏り**です。ある調査（総務省の白書等）では、日本ではIT人材の約**7割がIT企業（ベンダー）側**に、約3割がそれを使うユーザー企業側にいるとされます。米国ではこれがほぼ逆で、多くの人材がユーザー企業側にいるとされます。

この違いは、産業のかたちを大きく左右します。米国型では、ユーザー企業が自社内にエンジニアを抱え、自分たちでシステムを作る（内製する）のが基本です。これに対し日本型では、ユーザー企業が「**IT企業に開発を発注（委託）する**」受託開発（SI）が大きな位置を占めてきました。皆さんの多くが関わるのが、この受託の世界です。

■ Slerピラミッドと多重下請け

Slerピラミッド（多重下請けの構造）

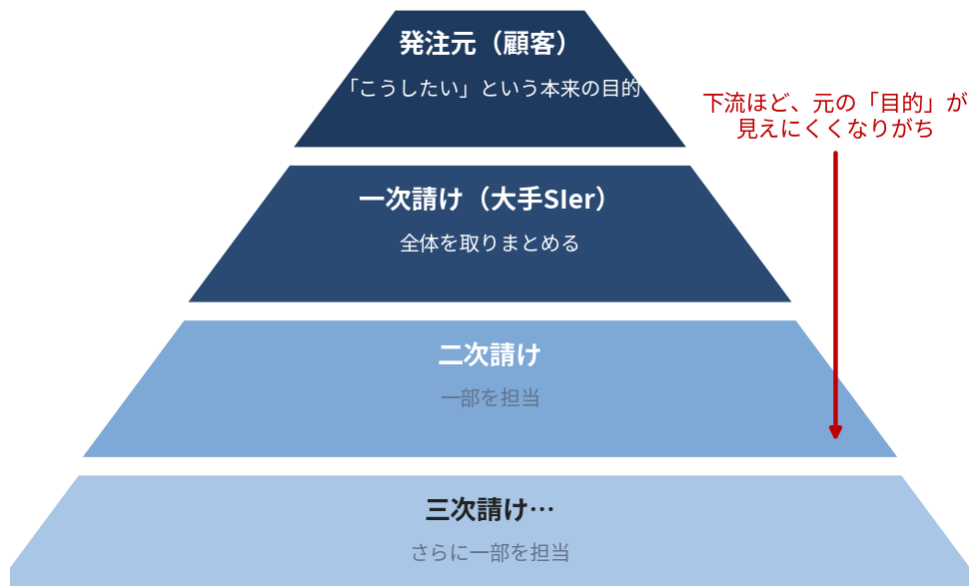


図 Slerピラミッド（多重下請けの構造）

受託の構造を特徴づけるのが、**多層の下請け構造**です。大きな案件を、発注元から一次請けの大手Slerが受け、その一部を二次請けへ、さらに三次請けへ……と仕事が流れていきます。この階層構造は、しばしば「**Slerピラミッド**」と呼ばれます。

この構造には合理性もあります（需要の波に応じて人を集めやすい、専門を分担できる等）。一方で、下層ほど条件が厳しくなりがち・上流の意図が伝わりにくい・価値が見えにくい、といった構造的な課題も指摘されてきました。ここで大切なのは、これを**善悪で語らず、構造として理解すること**です（本章コラム、第28章の取引法とも関わります）。

具体的にイメージしてみましょう。ある大規模システムの案件を、発注元が大手Sler（一次請け）に任せ、一次請けが全体設計と取りまとめを担い、実装の各部分を二次請け・三次請けの会社が分担する、こうした形がよくあります。下流に行くほど「決められた範囲を、決められた仕様で作る」性格が強まり、「なぜそれを作るのか」という全体の意図や顧客の顔が見えにくくなりがちです。これは、第21章で述べた「作業者とエンジニアの差」が**構造として生まれやすい場所**でもあります。だからこそ、下流にいても「この機能は全体の中で何のためか」を問う姿勢（第23章の全体像）が、差を生みます。構造は変えにくくても、構造の中での自分の向き合い方は選べるのです。

■ バリューチェーンと多様なプレイヤー

ソフトウェアが価値を生むまでの流れ（バリューチェーン）には、さまざまなプレイヤーがいます。受託で作る**Sler**、自社サービスを持つ**事業会社（プロダクト企業）**、開発の基盤を提供する**クラウド事業者**、特定領域の製品を持つ**ソフトウェアベンダー**など。同じ「ITエンジニア」でも、どのプレイヤーで働くかによって、仕事の性質は大きく変わります。

なかでも産業構造を大きく変えたのが**クラウド**の普及です。かつては、システムを動かすには自前で設備（サーバ等）を用意する必要があり、初期投資も運用負担も大きいものでした。今は、必要な分だけ計算資源を借りる形（クラウド）が一般化し、**小さなチームでも、世界規模のサービスを作れる**ようになりました。この変化は、起業のしやすさや、内製化（後述）の追い風にもなっています（クラウドの類型=IaaS/PaaS/SaaS等は21-8参照）。

■ グローバルとの対比

世界に目を向けると、巨大なプラットフォーム企業（検索・SNS・クラウド・スマートフォンの基盤などを握る企業群）が、産業の中心で大きな影響力を持っています。クラウドやAIの基盤、開発に使う道具の多くも、こうしたグローバルなプレイヤーが提供しているのが現状です。私たちが日々使う技術の土台が、世界とつながっていることを意識しておきましょう。

その上で、日本のIT投資には「**攻めより守りに偏りがち**」という特徴も指摘されてきました。既存業務の効率化やコスト削減（守り）への投資に比べ、新しい価値や事業を生み出す（攻め）ための投資が相対的に少ない、という傾向です。前述の内製化・DX（デジタルトランスフォーメーション）は、この「守り偏重」から「攻め」へと重心を移そうとする動きとも読めます。こうした大きな流れの中に、自分の仕事を置いて眺めてみると、見え方が変わってきます。

22-4 働き方とつくり方の現在

つくり方・働き方も、変化のただ中にあります。代表的な3つの潮流を見ましょう。

■ アジャイル化

変化を前提に、小さく作って早く確かめ、フィードバックで方向を修正する進め方（アジャイル）が広く取り入れられるようになりました（第24章）。背景には、「最初にすべてを決めきれない」ソフトウェアづくりの現実と、市場の変化の速さがあります。ただし第24章で詳しく見るように、受託・組込みなど、計画的な進め方（ウォーターフォール/V字）が適する領域も依然として多くあります。

■ リモート・分散

同じ場所に集まらず、離れた場所のメンバーが協働する働き方が一般化しました。これは働く場所の自由をもたらす一方で、**コミュニケーションのあり方**を変えました。対面でその場で済んでいた確認が、文字での非同期のやり取りに置き換わります。

■ オフショア・ニアショア

コスト削減や人材確保のため、開発の一部を海外拠点（オフショア）や地方拠点（ニアショア）に委ねる形も広く使われます。これは、言語・文化・時差を越えた協働を意味し、仕様を明確に書く力（第23章・第27章）や、プロセスを揃える力（第24章）が、より一層問われる働き方でもあります。

これらの潮流に共通するのは、「**どれか一つが正解**」ではなく、**目的や状況に応じて使い分けられるようになった**、ということです。選択肢が増えたぶん、エンジニアには「今の状況に、どの進め方・働き方が合うか」を理解し、選ぶ力が求められます。

新卒にとっての含意も押さえておきましょう。リモートや分散が一般化した現場では、「見て覚える」「その場で気軽に聞く」機会が、以前より減っていることがあります。だからこそ、**自分から発信し、質問し、文字で記録を残す**姿勢（第27章・第30章）が、これまで以上に成長を左右します。働き方の自由は、裏返せば「自分で動く力」をより強く求める、ということでもあるのです。「環境が整っていないから学べない」ではなく、「この環境で、自分はどう学びを取りにいくか」を考える、その構えが、変化の時代の新卒には効きます。

22-5 人材市場と学びの変化

■ 人材の需給—不足の見通しと、その変数

IT人材は、全体として不足が見込まれています。経済産業省の試算（2019年公表）では、2030年に相当規模（試算には幅がありますが、数十万人規模）の不足が生じうるとされました。ただしこ

の試算には重要な但し書きがあります。「**労働生産性が向上すれば、不足は軽減しうる**」というものです。つまり、**AIなどによる生産性向上が、需給を左右する大きな変数**になります。「人が足りないから安泰」と単純に考えるのではなく、「一人あたりが生み出せる価値が問われる」と読むのが正確です。

この点は、新卒のキャリア観にも関わります。「IT人材は不足しているから、就職すれば安泰だ」という見方は、半分正しく半分危ういものです。たしかに人手は足りていません。しかし、AIが定型的な作業を肩代わりするようになると、「数を埋めるだけの人」の価値は相対的に下がり、「**少ない人数で大きな価値を生み出せる人**」の価値が上がっていきます（第29章・第30章）。不足という追い風はありますが、その風に乗れるかどうかは、結局のところ「生み出せる価値」次第、この厳しさと希望の両方を、冷静に受け止めておきましょう。

■ 内製化・DXという構造変化

大きな潮流として、**内製化・DX**があります。ユーザー企業が、外注に頼り切るのではなく、自分たちで開発・改善できる力を持つとする動きです。デジタル技術を、単なるコスト削減（守り）ではなく、事業価値の創出（攻め）に使うという流れとも言えます。

ただし、この変化は一様ではありません。近年の調査（IPA＝情報処理推進機構のDX動向2025等）では、上流の「企画」は内製化が進む一方で、「開発」や「テスト」といった実装工程は、なお外部（受託）への依存が根強い、という結果も示されています。ここから読み取れるのは、「**受託（SI）の役割は当面続く。しかし同時に、上流の内製化という構造変化も着実に進む**」という両面です。受託を否定的に煽る必要も、変化に目を背ける必要もありません。両方を冷静に踏まえることが、これからの身の振り方を考える土台になります。

■ 学びの変化—「第三の学び場」と新しい世代

学ぶ環境も変わりました。日本では高校で「**情報I**」が必修化され（2022年度から）、さらに大学入学共通テストでも「情報」が出題されるようになり（2025年から）、プログラミングなどの素養を持つ世代が社会に入り始めています。これは、新卒に求められる前提が少しずつ上がっていくことを意味します。

また、学びの場そのものが広がりました。会社（OJT）でも学校でもない、「**第三の学び場**」、社外の勉強会・技術コミュニティ・OSS・オンライン学習、が、エンジニアの成長に欠かせない場になっています（第30章）。「会社が教えてくれるのを待つ」受け身の姿勢ではなく、**自ら学び場を選び、外へ出ていく**ことが、当たり前求められる時代です。

加えて、技術の寿命が短い時代（22-2）には、一度学んで終わりではなく、**学び直し（リスキリング）**が生涯にわたって必要になります。これは新卒にとって、ある意味では朗報です。後から入ってくる人も、先に入った人も、新しい技術の前では「学び続ける必要がある」点で同じだからです。スタート地点の差より、**学び続けられるかどうか**が、長い目で効いてきます。「今のスキルが足りない」ことを過度に恐れるより、「これから学び続ける仕組みを自分に作れるか」に目を向けましょう。会社の側も、人材を「採って終わり」ではなく「育て、学び直しを支える」方向へと、考え方を变えつつあります。

22-6 未来の見通し—技術トレンドと産業の行方

未来を正確に予言することは、誰にもできません。しかし、専門の調査機関が毎年「**戦略的に注目すべき技術トレンド**」を公表しており、方向感をつかむ参考になります。たとえばある調査機関が示した2026年向けのトレンドでは、**AIネイティブな開発、複数のAIが協調して働くシステム（マルチエージェント）、AIを前提としたセキュリティや、情報の出所・真正性の検証**などが上位に挙げられました（2025年時点の見通し。年が変われば内容も変わります）。

こうした予測の使い方には、コツがあります。「**当たる/外れる**」を競うものではなく、**変化の方向感をつかむための材料**として使うことです。とくに新しい技術には、**過度な期待と、その後の幻滅というサイクル（ハイプ）**がつきものです。話題になっているからと飛びつくのでも、よく分からないからと無視するのでもなく、「**これは何の問題を解くのか**」「自分の現場にとって意味があるか」を冷静に見極める、その目を持つことが大切です。

■ 新しい職種の登場と、求められる力の変化

技術の変化は、**新しい職種**も生み出します。たとえばクラウドの普及はクラウドの専門家を、データ活用の高まりはデータエンジニアやデータサイエンティストを、そして生成AIの普及は、AIを上手に使いこなす役割（AIを使った開発を支援する役割など）を生んでいます。「〇年前には存在しなかった職種」が次々に現れるのが、この産業の常です。

ここで重要なのは、職種名を追いかけることではありません。新しい職種が生まれても、その根底で求められるのは、**問題を定義する力・体系で考える力・学び続ける力**といった、本特論で扱ってきた普遍的な土台です。職種は変わっても、土台のある人は、新しい役割へ移っていきます。逆に言えば、流行の職種名だけを追いかけ、土台を持たない人は、その職種が変質したときに行き場を失います。「**何の職種か**」より「**どんな価値を生み出せるか**」、この視点を持っておきましょう（キャリアの考え方は第30章）。

■ 変化の時代に、新卒は何に備えるか

ここまで見てきた産業の現在地と方向を、新卒の備えとしてまとめ直すと、次のようになります。

第一に、**変わらないもの（土台）に投資すること**。個々の技術は移り変わりますが、コンピュータサイエンスの基礎、ソフトウェアエンジニアリングの体系、要求を捉える力、品質と倫理の感覚、こうした土台は長く効きます（第21章・第23章）。流行を追うことと、土台を固めることは、どちらも必要ですが、優先すべきは土台です。

第二に、**変化を前提に、学び続ける仕組みを自分に作る**こと。賞味期限の短い時代には、「学校を出たら学び終わり」という発想自体が通用しません。自分なりの学び方（第30章）を確立した人が、長く活躍します。

第三に、**産業の構造を理解した上で、自分の立ち位置を前向きに選ぶ**こと。受託か内製か、上流か下流か、専門特化かジェネラリストか、置かれた構造を嘆くのではなく、「この構造の中で、自分は何を積み上げるか」を主体的に考える。これは次のコラムのテーマでもあります。

💡 ポイント

この章で繰り返し立ち返る軸：トレンドの中身（個々の技術）は、すぐに移り変わります。
変わらないのは、「**変化を読み、必要なものを学び続ける力**」です。
新しい技術は、それを評価し使いこなす土台（体系・学び方）があってこそ、価値になります。

（コラム）受託の位置取りと伸びる4つの道

受託（SI）について、否定も美化もせず、**構造として**捉えましょう。22-3・22-5で見たとおり、受託の役割は当面続きます。一方で、「言われたものを作る」ことに最適化されすぎると、**価値起点で考える発想（プロダクト思考）が育ちにくい**、という構造的な弱点も指摘されます。「何を作れば顧客や利用者にとって価値があるか」を自ら問う機会が、構造的に少なくなりがちだからです。

ここで大切なのは、置かれた構造を理解した上で、**前向きに伸びる道**を選ぶことです。受託を起点に伸びていく方向として、たとえば次の4つが考えられます。

1. **上流シフト**、要件定義・設計など、上流の価値ある工程に強くなる。「何を作るか」を顧客と決める力は、代替されにくい価値です（第23章・第27章の力が効く）。
2. **専門特化**、品質・セキュリティ・特定の業務領域（ドメイン）など、希少で価値ある専門を深める。受託で多様な現場に触れられることは、むしろ専門を見つける好機にもなります（第30章のキャリア資本）。
3. **プロダクト／内製支援**、価値起点でものを考え、顧客の内製化を支援する側、あるいは自社プロダクトを持つ側へ回る。
4. **保守・運用のストック**、長く動かし続ける関係を、継続的な価値と収益に育てる。作って終わりでなく、育て続ける関係は、信頼と事業の安定を生みます（23-8）。

これらは排他的ではなく、組み合わせられます。たとえば「特定の業務領域に詳しくなり（専門特化）、その知識を武器に上流の要件定義に強くなる（上流シフト）」といった重ね方です。大切なのは、**今いる場所を出発点として、どの方向に一步踏み出すかを自分で決める**ことです。受託の現場は、多様な顧客・業務に触れられるという意味で、その出発点としての強みも持っています。

💡 ポイント

ここでも「作業員 vs エンジニア」は構造の問題であり、個人・先輩・自社を責めるものではありません。

構造を理解し、自分の意志で前向きな道を選ぶ、それが、伸びるエンジニアの姿勢です。

技術や産業の変化は、不安の種であると同時に、立ち位置を選び直せる好機でもあります。

22-7 考えてみよう（グループ討議）

教室内では手を動かす実技は行いません。3～4人のグループで、考えを言葉にしてみましょう。

✦ 考えてみよう：討議1：数年で寿命が縮む時代に、何を学ぶか

「学んだ技術の賞味期限が数年」という時代に、**何を学べば長く役に立つか**。

「すぐ古くなりそうなもの」と「長く効きそうなもの」を分けて挙げ、なぜそう考えるかを話し合ってください。

本特論を貫く中心の考え（体系・考え方・学び方が陳腐化しにくい）を、自分の言葉で説明できるようになるのが狙いです。

✦ 考えてみよう：討議2：自分はどの船に乗っているか

22-3の産業構造の中で、「自分（会社・配属）はどこに位置するか」を考える。

その位置の**強みと弱み（構造的な課題）**を挙げ、コラムの「伸びる4つの道」のどれに関心があるかを共有してください。構造を嘆くのではなく、「ここから何を積むか」を前向きに考えるのが狙いです。

✦ 考えてみよう：討議3：その新技術は、何の問題を解くのか

最近よく耳にする技術の言葉（バズワード）を一つ取り上げる。

それが「**何の問題を解く技術なのか**」「自分たちの現場にとって意味がありそうか」を、22-6のハイプの視点で話し合ってください。流行に飛びつくのでも無視するのでもなく、冷静に評価する練習です。

📖 事例から学ぶ：10年でこんなに変わった

少し前まで、音楽はCDを買い、映画はレンタルし、道に迷えば紙の地図を広げていました。
 いまは同じことをスマホ1台がこなします。
 IT産業は「あって当たり前」のものを、わずか数年で丸ごと入れかえてしまう速さで動いています。
 だからこそ、いま新しく見える技術も数年で古くなり、逆に古く見える知識（仕組みの理解）は長く役に立ちます。

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・ 気になる技術の言葉を1つ選び、「これは何の役に立つの？」と自分なりに考えてみる。

22-8 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ 日本のIT業界の特徴を、ざっくり一言で（受託が多い、など）。
- ・ なぜ「学び続ける力」が大事なのか。
- ・ 受託から伸びる道を、1つ。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、次の章やこれからの仕事の中で、少しずつ埋めていきましょう。

第23章 ソフトウェアエンジニアリングの体系（SWEBOK V4）（39日目）

🎯 この章のゴール

この章では、「コードが書ける」と「エンジニアとして働ける」は別だということ、そしてものづくりの全体像（体系）を学びます。

終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ①【違いがわかる】「コードが書ける」と「エンジニアとして働ける」は別、という意味がわかる。
- ②【イメージできる】ものづくりは「作る」前後（決める・確かめる・直し続ける）まで広い、と言える。
- ③【考え方がわかる】先人のまとめ（体系）を使うと迷いにくい、という考え方がわかる。

💡 まず自己診断—読む前に、いまの自分を知ろう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「エンジニアの仕事の広さを自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
- ② 「「コードが書ける」ことと「エンジニアとして働ける」ことは、同じだと思いますか？」を、いまの考えで一言だけ書きます（正解は気にしません）。

読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

23-1 概要—なぜ体系を学ぶのか／「書ける」と「働ける」は別物

第21章で、エンジニアは「体系で考える」専門職だと述べました。本章では、その体系の正体、**ソフトウェアエンジニアリングの知識体系**、を一枚の見取り図として広げます。この第7部のなかでも、本章は**背骨**にあたります。以降の各章（プロセス・品質・セキュリティ・ドキュメント…）は、いずれも本章で広げる見取り図のどこかに位置づくからです。だからこそ、ここはじっくり腰を据えて読んでください。

なぜ体系を学ぶのでしょうか。理由はシンプルです。**「コードが書ける」ことと「ソフトウェアエンジニアとして働ける」ことは、別物**だからです。

学生時代のプログラミングを思い出してください。多くの場合、課題は「仕様が与えられ」「一人で作り」「動けば完成」でした。ところが現場のソフトウェアづくりは、その前後が圧倒的に長いのです。動くコードを書く「構築」は、全体のごく一部にすぎません。その前には「**何を作るべきかを決める（要求）**」「**全体の骨組みを考える（アーキテクチャ・設計）**」があり、その後には「**品質を保証する（テスト）**」「**動かし続ける（運用・保守）**」があり、全体を貫いて「**進め方を管理する（マネジメント）**」「**安全を確保する（セキュリティ）**」が走っています。しかも、ほとんどの仕事は**一人ではなくチームで、一度きりではなく長期間**続きます。

この全体を見渡し、つなげて考えられる人が、エンジニアです。逆に、与えられた一部分だけを「作業」としてこなす人は、いくらコードが速く書いても、全体の中で自分が何をしているのかを見失います。**体系を学ぶとは、この「全体の見取り図」を手に入れること**にほかなりません。

この章で繰り返し立ち返る軸は、特論全体と同じです。

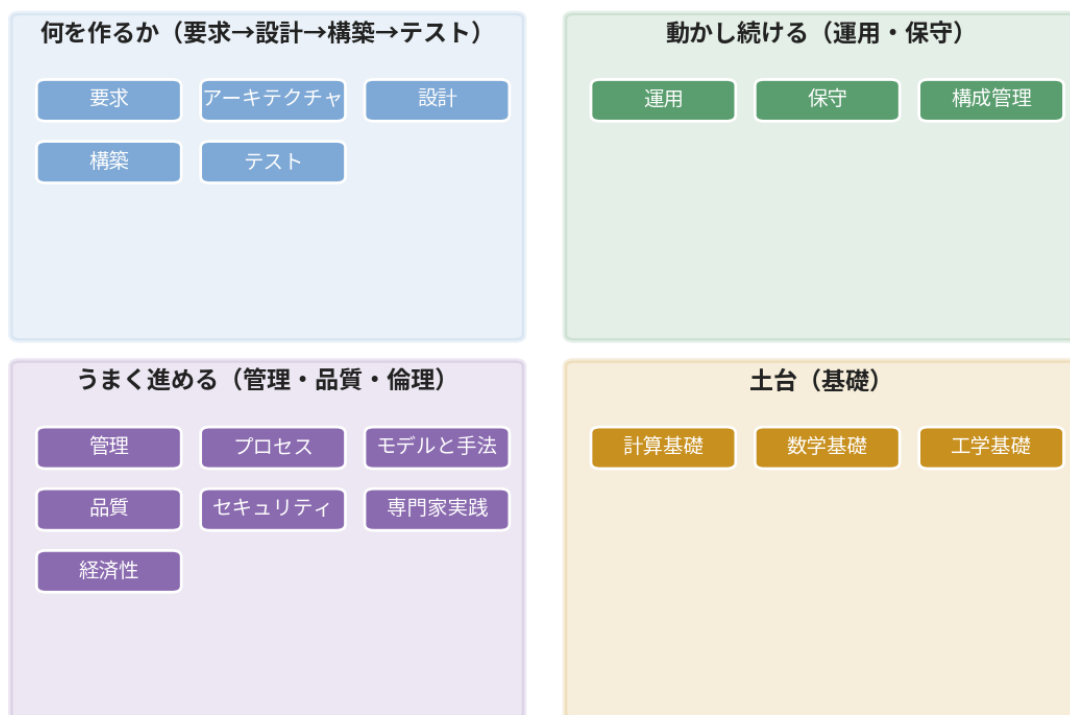
💡 ポイント

AI時代にこそ、体系が効く。

コードを書く作業は、AIが大きく助けてくれるようになりました。だからこそ、「何を・なぜ・どの品質で作るか」を判断する力、つまり体系に裏打ちされた全体観、が、人間のエンジニアの価値の中心になります。体系を知らない人はAIの出力をそのまま受け取るしかありませんが、体系を持つ人はそれを評価し、直し、使いこなせます。たとえば「このコードは動くが、保守性が低い」「この設計はテストしづらい」「この要求は非機能要件が抜けている」、こうした判断はすべて体系の言葉です。

本章では、まず18の知識領域を「**こういう全体像がある**」と俯瞰し、続いて中核領域（要求・設計・構築・テスト・運用保守・管理）の**考え方**を一つずつ解説します。さらに章の後半では、「一つの機能が、この地図をどう一周するか」を通し例で追います。なお、一つひとつの**手の動かし方**（実装・ツール操作）は別科目やOJTに譲り、本章は全体を見渡し、つなげて考えるための「鳥の目」を養うことに徹します。

ソフトウェアエンジニアリングの体系（SWEBOK 18の知識領域）－ 全体の地図



※ たった一つの機能を作るだけでも、この地図のあちこちに触れている

図 ソフトウェアエンジニアリングの体系（SWEBOK 18の知識領域）

23-2 歴史と背景—ソフトウェア危機とソフトウェア工学の誕生

ソフトウェアエンジニアリングという分野には、生まれた理由があります。1960年代、コンピュータの普及とともにソフトウェアが急速に大規模・複雑になり、納期遅延・予算超過・品質トラブルが各地で多発しました。これは「**ソフトウェア危機 (software crisis)**」と呼ばれ、1968年にNATOが主催した会議で「**software engineering (ソフトウェア工学)**」という言葉が広まる契機になったとされます。

その精神は明快です。ソフトウェアづくりを、個人の勘や職人芸だけに任せるのではなく、**規律ある「工学」として体系化しよう**、この意志から、本章で扱うSWEBOKのような知識体系が育ってきました。

ここで立ち止まりたいのは、「半世紀以上前の問題が、なぜ今も語られるのか」という点です。ソフトウェア危機が指摘した「**複雑さと変化にどう立ち向かうか**」という問いは、今も解決していません。むしろシステムは当時より桁違いに複雑になり、変化も速くなっています。だからこそ、先人が積み上げた体系は「古い知識」ではなく、**今の私たちが立つ土台**なのです。

23-3 SWEBOK V4 全体地図—18の知識領域

■ SWEBOKとは

SWEBOK (スウェボック/Software Engineering Body of Knowledge) は、ソフトウェアエンジニアリングの知識を体系的に整理したガイドです。IEEE Computer Society が発行し、無償で入手できます。最新版は **V4.0 (2024年10月公開)** で、**18の知識領域 (KA: Knowledge Area)** から構成されます。

版を追うごとに領域が増えてきたこと自体が、この分野の広がりをお話します。2004年版は10領域、V3.0 (2013年) は15領域、そしてV4で18領域。とくにV4で新しく独立した領域が、**アーキテクチャ・セキュリティ・運用**の3つです。クラウドや大規模サービスが当たり前になり、これらの重要度が増したことを反映しています。「体系は固定された石板ではなく、現実に合わせて更新される生きた地図だ」と捉えてください。

■ 18の知識領域は「どんな問いに答えるか」

領域名を眺めるだけでは、頭に入りません。そこで、**各領域が"どんな問いに答えるための知識か"**という形で一望してみます（このまとめ方・分類は理解のための補助線で、SWEBOK本体の正式な区分とは異なります）。

<何を・どう作るか>

#	知識領域	答える問い
1	ソフトウェア要求	何を作るべきか？ 本当に必要なものは何か？
2	ソフトウェアアーキテクチャ ★V4新設	全体をどんな骨組みで構成するか？
3	ソフトウェア設計	内部をどんな構造に分けるか？
4	ソフトウェア構築	どう作れば読みやすく直しやすいか？
5	ソフトウェアテスト	正しく動くと、どうやって確かめるか？

<動かし続ける>

#	知識領域	答える問い
6	運用 ★V4新設	本番で安定して動かし続けるには？
7	ソフトウェア保守	どう直し、変化に合わせて育てるか？
8	構成管理	変更を誰がどう安全に管理するか？

<うまく進める>

#	知識領域	答える問い
9	エンジニアリング管理	品質・費用・納期をどう両立させるか？
10	プロセス	どんな進め方の型を選ぶか？
11	モデルと手法	考えをどう図や手法で表すか？
12	ソフトウェア品質	良し悪しを何で測り、どう作り込むか？
13	ソフトウェアセキュリティ ★V4新設	どうやって安全に作るか？

<土台>

#	知識領域	答える問い
14	専門家としての実践	プロとして・倫理的にどう振る舞うか？（第21章）
15	経済性	コストと価値をどう判断するか？
16	コンピューティング基礎	計算機の仕組みの土台は？
17	数学的基礎	論理・離散数学などの土台は？
18	工学的基礎	測定・実験など工学共通の土台は？

18も領域があると圧倒されるかもしれませんが、「問い」の形にすると、どれも当たり前のことだと気づくはず。良いものを作るために、人類が「これは考えておくべきだ」と積み上げてきた問いのリスト、それがSWEBOKの正体です。

■ この見取り図の使い方——一つの機能が、全体を一周する

全体像を持つ真価は、「自分の仕事がどこか」を指させることにあります。具体例で一周してみましょう。

あなたが「**会員が自分の注文履歴を一覧で見られる**」という機能を担当したとします。一見、画面とデータを作るだけの「作業」に見えますが、体系の目で見ると、こうなります。

1. **要求 (KA1)** : そもそも「履歴」とは何件まで？ いつからの分？ 誰に見せて誰に見せない？、ここが曖昧だと、後で必ず作り直しになります。
2. **アーキテクチャ・設計 (KA2,3)** : 履歴データはどこから取るか、どの部品が責任を持つか。表示と取得を分けておけば、後で仕様が変更しても直しやすい。
3. **モデル (KA11)** : 画面・サーバ・データの会話の流れをシーケンス図にして、チームと認識をそろえる。
4. **構築 (KA4)** : 読みやすく、後で直しやすいコードで実装する。
5. **構成管理 (KA8)** : 変更履歴を残し、他の人の作業と安全に統合する。
6. **テスト (KA5)** : 履歴が0件のとき、大量にあるとき、他人の履歴が混ざらないか、観点を決めて確かめる。
7. **品質 (KA12)** : 1万件あっても3秒で出るか（性能）、他人に見えてしまわないか（セキュリティ）。
8. **運用・保守 (KA6,7)** : 本番で遅くなっていないか見張り、仕様変更（例：表示件数の変更）にに応じていく。
9. **管理・経済性 (KA9,15)** : いつまでに、どれだけの工数で、を見積もり、見通しを共有する。

10. **専門家実践 (KA14)** : 他人の注文情報を扱う以上、「見てよいデータか」を倫理として問う。

たった一つの機能が、18領域のほぼ全域に触れているのがわかります。「**作業者はこのうち4番(構築)だけを見ますが、「エンジニア」は1~10のつながりを見て、最も手戻りの少ない順路を選びます。**

💡 ポイント

AIに丸投げすると、どこで事故るか。

この機能をAIに丸投げしたら、と考えてみましょう。

AIは4番(構築)のコードはすばやく出してくれます。

しかし、1番(要求: 誰に見せて誰に見せないか)が曖昧なまま投げれば、AIは"それらしい"履歴画面を作るものの、他人の注文が見えてしまう不具合は防げません。

事故は「コードの書き方」ではなく、その前後の**要求・テスト・品質 (KA1,5,12) の見落とし**で起きます。

どこを任せ、どこを自分で握るかを決められるのが、体系を持つ人です(生成AIは第29章で深めます)。

本研修では、このうち中核となる領域(要求・設計・構築・テスト・品質・セキュリティ・構成管理・専門家実践)を重点的にたどりま。残りの領域も「存在する」と知っておくだけで、いざ必要になったときに学びにいきます。

23-4 要求と要件定義—何を作るべきかを決める

ソフトウェアづくりの出発点は、コードではなく「**何を作るべきか**」を決めることです。これを**要求 (requirements)**と呼びます。ここを間違えると、どれだけ美しいコードを書いても、誰も使わないものができあがります。実際、開発の手戻りや失敗の多くは、コードの書き間違いではなく、**この上流のすれ違いに原因がある**と言われます。

なぜ上流のミスは怖いのか。それは、**間違いは下流に行くほど直すのが高くてつ**からです。要求段階の勘違いは、紙の上で直せば一瞬です。しかし、それが設計・実装・テストを通り抜けてリリース後に発覚すると、調査・修正・再テスト・顧客対応まで、何十倍もの手間に膨らみます。出発点で1度ずれた方向は、進むほど大きく開いていく、だから要求に時間をかけることは、急がば回れの王道なのです。

■ 機能要件と非機能要件

要求には大きく2種類あります。

- ・ **機能要件**、「何ができるか」(例: 注文を登録できる、商品を検索できる、月次の集計を出せる)
- ・ **非機能要件**、「どのくらいの品質で動くか」(例: 3秒以内に表示される、同時に1,000人が使える、個人情報を守られる、障害時に復旧できる)

新卒がとくに意識したいのは**非機能要件**です。これは「言われないと気づきにくいのに、抜けると大事故になる」厄介な存在です。「動くものは作ったが、遅くて使い物にならない」「平常時は捌けたが、セールの集中アクセスで落ちた」、こうした失敗は、機能ではなく非機能の見落としから生まれます。機能要件が「何をするか」なら、非機能要件は「**どう在るべきか**」。後者は第25章の品質特性(性能・信頼性・セキュリティ等)と一対一でつながります。

■ 要求を固める技術

- ・ **引き出し（要求獲得）**、顧客や利用者へのヒアリング。ここで初心者がやりがちなのが「**言われたことをそのまま書く**」ことです。プロは、その奥にある**本当の目的（なぜそれが欲しいのか）**まで掘り下げます。顧客は「ドリルが欲しい」と言いますが、本当に欲しいのは「穴」かもしれない、という有名なたとえの通り、顧客自身、本当に必要なものを言葉にできていないことが多いのです。「なぜ?」を5回繰り返す、といった姿勢が効きます。
- ・ **優先順位づけ**、要求は、すべてが同じ重みではありません。「絶対に要る／あると良い／なくても良い」を分け、限られた時間を本当に重要なものに使います。「全部いますぐ」は、たいてい「何も決めていない」と同じです。
- ・ **合意形成とスコープ**、「これを作る／作らない（スコープ）」をはっきりさせ、関係者で認識をそろえる。曖昧なまま進めると、後で必ず食い違いします。
- ・ **受け入れ条件を決める**、「どうなったら"できた"と言えるか」を、作る前に合意しておく。ゴールを先に描くから、迷わず走れます。
- ・ **文書化とトレーサビリティ**、決めたことを書き残し、「この機能は、どの要求から来たのか」を後からたどれるようにする（トレーサビリティ）。これにより、要求が変わったとき、影響範囲を追えます。受託では、この文書が後の検収の土台になります（本章コラム・第28章参照）。

💡 ポイント

要求は「最初に一度決めて終わり」とは限りません。アジャイルな進め方では、要求を少しずつ確かめながら育てます（第24章）。ただしどんな進め方でも、「何のために作るのか」を問い続ける姿勢は共通です。

23-5 アーキテクチャと設計—設計原則とデータ設計

要求が固まったら、次は**どう作るか**を考えます。大きく分けて、全体の骨組みを決める「**アーキテクチャ**」（V4で独立領域に）と、内部の構造を決める「**設計**」があります。家にたとえれば、間取りや構造・耐震の方針を決めるのがアーキテクチャ、各部屋の使い勝手を作り込むのが設計です。骨組みを後から変えるのは大工事になる、だからアーキテクチャの判断は重く、慎重さが要ります。

■ すべての設計の根っこ—抽象化とモジュール化

良い設計の根底には、たった2つの考え方があります。

- ・ **抽象化**、細部を隠し、「何をするか」だけを見せる。利用する側は中身の作りを知らなくても使える（車の運転にエンジンの構造の知識が要らないのと同じ）。
- ・ **モジュール化**、大きなものを、意味のあるまとまり（部品）に分ける。分けることで、人は複雑さに立ち向かえます。

人間が一度に理解できる量には限りがあります。だから、複雑なシステムを「適切に分け、細部を隠す」ことが、設計のすべての出発点になります。

■ よい設計の感覚—凝集度と結合度

分け方の良し悪しを語る古典的な物差しが、**凝集度（ぎょうしゅうど）**と**結合度（けつごうど）**です。

- ・ **高凝集**、一つのまとまり（部品）が、関連する仕事だけをきちんと担っている状態。よい。
- ・ **低結合**、部品どうしの依存が少なく、独立して直せる状態。よい。

目指すのは「**高凝集・低結合**」です。たとえば、一つの部品が「データの保存も、画面表示も、メ

ール送信も」何でも抱え込んでいると（低凝集）、どこか一つを直すたびに全体に影響が及びます。逆に、役割ごとに部品が分かれ、互いの依存が薄ければ（低結合）、一か所の変更が他に波及せず、変更も再利用もテストもしやすくなります。「変更に強い構造」は、長く使われるソフトウェアの生命線です。

📦 事例から学ぶ：ある画面の表示部品が、ついでにメール送信処理まで抱えていた（低凝集）

ある日「メール文面を少し変えるだけ」のはずが、表示処理に巻き込まれて画面が崩れ、修正とテストが半日仕事に。

役割が分かれていれば、メール側だけ直して済んだはずです。

「小さな変更が、関係ない所まで壊す」のは、低凝集・高結合のわかりやすい症状です。

■ 設計原則 SOLID

オブジェクト指向設計でよく引かれる5つの原則の頭文字をまとめたものが **SOLID** です。新卒は「こういう原則群がある」と知っておけば十分です（深掘りは別科目へ）。

- ・ **S**：単一責任の原則（一つの部品は一つの責任を持つ）
- ・ **O**：開放閉鎖の原則（拡張に開き、修正に閉じる）
- ・ **L**：リスコフの置換原則（派生は基底と置き換え可能に）
- ・ **I**：インターフェース分離の原則（使わない機能に依存させない）
- ・ **D**：依存性逆転の原則（具体でなく抽象に依存する）

5つを丸暗記する必要はありません。根っこはどれも同じで、「**変更に強く、理解しやすい構造を保つ**」ための知恵です。先ほどの「高凝集・低結合」を、より具体的な指針に落としたもの、と捉えてください。なお、こうした「よくある設計問題への定番の解き方」を名前付きで整理した**デザインパターン**というカタログも存在します（これも、いま暗記する必要はなく「ある」と知っておけば十分です）。

📦 事例から学ぶ：単一責任の原則（S）を破り、一つの部品が「料金計算」と「請求書PDFの出力」を兼ねていると、税率が変わっただけでPDF処理まで巻き込んでテストし直すことになります

責任を一つに絞ってあれば、変更は計算側だけで完結します。SOLIDは結局、「**変更したい所だけを安全に変えられる**」状態を保つための知恵です。

■ 代表的なアーキテクチャの型（俯瞰）

全体の骨組みにも、よく使われる「型」があります。型の名前より、**それぞれが何を得て何を失うか（トレードオフ）**を感じ取るのが狙いです。

- ・ **階層化（レイヤード）**、画面・処理・データなど役割ごとに層を分ける。わかりやすく、もっとも基本的。
- ・ **クライアント／サーバ**、利用者側（クライアント）と提供側（サーバ）に分ける。Webの基本形。
- ・ **モノリス（一枚岩）対 マイクロサービス**、全機能を一つにまとめる作り（モノリス）は単純だが大きくなると重い。機能ごとに小さく分けて独立させる作り（マイクロサービス）は柔軟だが、分けた分だけ管理が複雑になる。

「最新だから良い」のではありません。**規模・チーム・変化の速さに合った骨組みを選ぶ**のがアーキテクチャの判断であり、ここに「正解は一つではない」という工学の現実が表れます。

■ データ設計—ER図と正規化

多くのシステムの中心には**データ**があります。データの構造を設計する基本が、**ER図（実体関連図）**でデータ同士の関係を表すことと、**正規化**でデータの重複や矛盾を防ぐことです。

たとえば顧客の住所を、注文ごとに何度もコピーして持つと、引っ越し一つで全部を直す羽目になり、直し漏れれば矛盾が生まれます。「**同じ情報を二か所に持たない**」、この単純な規律（正規化の考え方）が、後々のデータ不整合という、見つけにくく厄介なバグを未然に防ぎます。データ設計のまずさは、運用が始まってから静かに効いてくるので、上流での丁寧さが効きます。

23-6 モデルで考える—UMLで設計を伝える

設計は、頭の中だけでは他者に伝わりません。言葉だけの説明は、人によって解釈が割れます。そこで、考えを**図（モデル）**にして共有する共通言語が、**UML（統一モデリング言語）**です。UMLには十数種類の図がありますが、新卒はまず**代表的な3つを「読める」**ことを目標にします（自分で精密に描けるのは後でよい）。

- ・ **クラス図**、システムの登場人物（データやモノ）と、その関係を表す。「静的な構造」を示します。「このシステムには何があるか」を一望できます。
- ・ **シーケンス図**、処理が時間に沿って、誰から誰へどうやり取りされるかを表す。「動きの流れ」を示します。「ログイン時に、画面とサーバとDBがどう会話するか」といった流れが追えます。
- ・ **ユースケース図**、誰（利用者）が何をできるかを表す。「要求の見取り図」。

大事なのは、図は**目的のための道具**だということです。きれいなUMLを描くこと自体が目的ではありません。「設計の意図を、他者に正確に・速く伝える」ために図を使う、この感覚を持ってください。実務では、厳密な記法より「伝わること」を優先して、簡略化した図やホワイトボードの落書きで議論することも多くあります。図は、頭の中の設計を**チームの共有物にする**ための言語なのです（書いて伝える技術は第27章で深めます）。

23-7 構築と構成管理—バージョン管理とコードレビュー

■ 構築（コンストラクション）

いよいよコードを書く段階が**構築**です。ただしプロの構築は「動けばいい」ではありません。**読みやすく・直しやすく・他者が理解できる**コードを書くことが、品質の一部です。

なぜなら、コードは「書く時間」より「**後で読まれ・直される時間**」のほうがずっと長いからです。あなたが書いた1行は、半年後の別の誰か（あるいは未来のあなた自身）が読み、直すことになります。意味のわかる命名、チームで決めたコーディング規約、必要十分なコメント、こうした作法は、未来の読み手への思いやりであり、第21章の「同僚への公正さ」の実践でもあります。「自分だけがわかるコード」は、チームの資産にはなりません。

■ 技術的負債という考え方

ここで一つ、現場で頻出する重要概念を押さえます。**技術的負債（technical debt）**です。これは、「**今は速いが、後で利息を払うことになる、手抜きや先送り**」のたとえです。締め切りに追われて雑な作りで通すと、目先は速く進みます。しかしその箇所は、後で誰かが触るたびに余計な時間（＝利息）を取られ、放置するほど積み上がって、やがて身動きが取れなくなります。

負債は必ずしも悪ではありません（戦略的に借りることもあります）。大事なのは、**借りていると自覚し、返す計画を持つ**ことです。「とりあえず動いた」で放置しないこと、これが、長く健全にソフトウェアを育てる鍵です。

■ 構成管理とバージョン管理

複数人で、長期間、安全に開発を進めるための仕組みが**構成管理**です。その中心が**バージョン管理システム**（現在の事実上の標準は **Git**）。「いつ・誰が・何を・なぜ変えたか」の履歴を残し、複数人の変更を安全に統合し、必要なら過去の状態に戻せます。これがなければ、大人数の開発は「最新ファイルはどれ？」の混乱で破綻します。

💡 ポイント

ここでは構成管理の**概念・作法・規律**を扱います。

CI/CD（継続的インテグレーション／継続的デリバリー）・DevOps・ブランチ運用といった「**プロセスとしての回し方**」は第24章で扱います。

「道具と概念」が本章、「回し方」が次章、という分担です。

■ コードレビューと、技術的対立の扱い

書いたコードを他者が見て改善する**コードレビュー**は、品質を作り込む強力な習慣です。効果は一つではありません。欠陥を早期に見つける、知識をチームで共有する、特定の人しか触れない「属人化」を防ぐ、そして書き手が学んで育つ。

新卒にとくに強調したいのは、レビューでの**意見の対立は、人格の否定ではない**ということです。「このコードはこうした方がよい」は「あなたはダメだ」ではありません。技術的な議論は、より良い答えに到達するための協働です。指摘する側は**事実と理由**で語り（「動かない」ではなく「この入力でこう落ちる」）、される側は**防御的にならず**に学ぶ、この文化が、強いチームを作ります。最初自分のコードを直されるのが怖いものですが、「レビューは自分への攻撃ではなく、品質への投資」と捉え直せると、成長が一気に加速します。

23-8 運用と保守—動かし続ける責任

ソフトウェアは、作って終わりではありません。**動かし続け、直し続ける**ことまでがエンジニアの責任です。むしろシステムの一生のうち、作る期間より**運用・保守の期間のほうがずっと長い**のが普通です。「リリースはゴールではなくスタート」と言われるゆえんです。

■ 運用（オペレーション）

V4で独立領域になった**運用**は、「システムが本番で安定して動き続けるようにする」仕事です。鍵になる考え方が**Observability（可観測性）**、システムの状態（ログ・指標・異常）を「見えるようにしておく」ことです。見えなければ、異常にも気づけず、起きてしまった障害の原因もたどれません。健康診断の数値を取っていなければ、体調の異変に気づけないのと同じです。

障害が起きたときの**初動（検知 → 影響の確認 → 対応 → 記録）**も運用の中核です。とくに「記録」は軽視されがちですが、同じ障害を二度起こさないための学び（第24章のふりかえり）の材料になります。「誰が悪いか」ではなく「仕組みのどこに穴があったか」を記録する、これが第25章の品質文化にもつながります。

■ 保守の4類型

保守には、目的の異なる4つの型があります（国際規格の整理に基づく一般的な分類）。

- ・ **是正保守**、見つかった不具合を直す
- ・ **適応保守**、環境の変化（OSの更新・法改正・外部サービスの仕様変更など）に合わせる
- ・ **完全化保守**、性能や使い勝手を改善する

- ・ **予防保守**、将来の問題を未然に防ぐ（顕在化する前に手を打つ）

「保守＝バグ直し（是正）」だけだと思われがちですが、実際には**環境変化への追従や改善のほうが多い**ことも珍しくありません。動いているシステムを、変化に合わせて育て続ける、これも、新規開発に劣らず立派なエンジニアリングです。他人が作ったシステムを読み解いて安全に直す力は、むしろ高度な能力であり、「保守は地味」という先入観は現場に出る前に捨てておきましょう。

23-9 マネジメント・プロセス・経済性

技術だけでは、プロジェクトは成功しません。**うまく進める**ための領域も、体系の正式な一部です。

- ・ **QCD**、Quality（品質）・Cost（費用）・Delivery（納期）。この3つは互いに引っ張り合う関係（トレードオフ）にあります。たとえば納期を縮めようとするれば、品質か費用（人員）に皺寄せがいきます。何を優先するかを**関係者で合意する**ことが、管理の核心です。「全部を最大化する」は幻想だと知ることが、現実的な判断の出発点です。
- ・ **WBS（作業分解構造図）**、大きな仕事を、管理できる小さな作業に分解して見通しを立てる手法（第21章「段取る力」の、プロジェクト規模版）。
- ・ **リスク管理**、「うまくいかないかもしれないこと」を先に洗い出し、起きる前に備える。問題が起きてから慌てるのではなく、起きうる問題を先回りして考えるのがプロです。
- ・ **見積もりの誠実さ**、経済性の領域で、最も新卒に効くのがこれです。見積もりは「願望」や「相手の期待に合わせた数字」ではなく、**根拠に基づく誠実な見立て**であるべきです。できないことを「できます」と言ってしまうと、後で全員が苦しみます。**できないことをできると言わない**、これは技術であると同時に、第21章で学んだ倫理（誠実さ）そのものです。

プロセス（進め方の型。ウォーターフォール／アジャイル等）については、重要なので第24章でまとめて扱います。本章では「進め方そのものも、選び・設計する対象である（KA10）」とだけ押さえておきます。

23-10 エンジニアのロールはこんなに広い

「ソフトウェアエンジニア」と一口に言っても、現場の役割（ロール）は多様で、年々広がっています。代表的なものを挙げます（呼称や境界は会社・時期により異なります。○年時点の一般的な呼び名として捉えてください）。

- ・ **アプリケーション／バックエンド／フロントエンドエンジニア**、機能を作る
- ・ **SRE（Site Reliability Engineer）**、システムの信頼性・運用を専門に担う
- ・ **DevOpsエンジニア**、開発と運用をつなぎ、自動化を進める
- ・ **データエンジニア**、データの基盤・パイプラインを作る
- ・ **機械学習（ML）エンジニア**、AI／機械学習モデルを開発・運用する
- ・ **QA／テストエンジニア**、品質を専門に支える（第25章）
- ・ **セキュリティエンジニア**、安全を専門に支える（第26章）

新卒のうち、どれか一つに早く絞る必要はありません。まず全体像（横棒）を広げ、その中で「これを深めたい」という縦棒を見つけていきます（第21章のT型、キャリアは第30章）。

ここで大事なのは、**どのロールも、本章の体系という共通言語の上に立っている**という点です。フロントエンドであれデータであれ、要求があり、設計があり、品質があり、保守がある。体系を持っていれば、別のロールの人と会話でき、必要に応じて自分の専門を移していくこともできます。逆に体系のない「一芸」だけだと、その芸が陳腐化したときにつぶしが効きません。

〔コラム〕仕様の握りと検収／保守は受託の収益源

本章の体系は業種を問わず普遍ですが、**受託開発（SI）**の現実に接続すると、いくつかの領域がとりわけ重い意味を持ちます。

「**仕様の握り**」と**検収**。受託では、「何を作るか（要求・仕様）」を顧客と**文書で合意**することが、後の**検収**（顧客が成果物を確認し、契約通りかを判断する手続き）の土台になります。仕様が曖昧なまま進むと、「言った/言わない」の水掛け論になり、追加作業のコストを誰が負担するかもめまします。23-4で学んだ「要求の文書化と合意」は、受託では**自分とチームを守る防具**でもあるのです。開発の途中で仕様変更が出たら、口頭で済ませず、**変更として記録し、合意し直す**、この一手間、後の大きなトラブルを防ぎます（契約面は第28章）。

保守は受託の収益源。23-8で「保守の期間は長い」と述べました。受託ビジネスでは、この保守・運用フェーズが**継続的な収益の柱**になることが少なくありません。作って納めて終わりではなく、「動かし続け、育て続ける」関係こそが、顧客との信頼と事業の安定を生みます。だからこそ、後から保守する人（多くは未来のあなたやチーム）が困らないコード・ドキュメントを残すことは、技術であると同時にビジネス上の価値そのものです。「読みやすいコードを書く」「技術的負債を残さない」は、人柄の問題ではなく経営の問題でもあるのです。

💡 ポイント

「作業者」は目の前の構築だけを見ますが、「エンジニア」は要求から保守までの一生（ライフサイクル）を見て、価値を最大化します。受託の数字は、その差を映す鏡です。

23-11 考えてみよう（グループ討議）

教室内では手を動かす実技は行いません。3～4人のグループで、考えを言葉にしてみましょう。

✦ 考えてみよう：討議1：自分の仕事は18KAのどこか

これまでの経験（アルバイト・制作・学業の課題・この研修の演習など）で「ソフトウェアやシステムに関わる作業」をしたことがあれば、それを思い出してください。

なければ「これからやってみたい開発」を想像でかまいません。

- ・ その作業は、23-3の**18の知識領域のどこ**に位置するか
- ・ その作業の**前後（上流・下流）**には、どの領域の仕事があったはずか（例：自分は「構築」をしたが、その前の「要求」や「設計」は誰がどう決めていたか）

を全体像の上で指さしながら話し合ってください。「自分の担当は全体のどこか」を意識することが、視野を広げる第一歩です。

✦ 考えてみよう：討議2：体系を知る人と知らない人で、AIの使い方はどう変わるか

同じAIコーディング支援ツールを、「体系を知らない人」と「体系を知る人」が使ったとき、結果にどんな差が生まれるだろうか。

要求・設計・テスト・品質といった領域を念頭に、「体系を持つ人だからこそできるAIの使いこなし方」を具体的に挙げてみてください。

本特論を貫く中心の考え「AI時代にこそ体系が効く」を、自分の言葉で説明できるようになるのが狙いです（生成AIは第29章で深めます）。

✦ 考えてみよう：討議3：手戻りは、どの工程で生まれるか

「作ったのに作り直しになった」という失敗（実体験でも、見聞きした話でも可）を一つ取り上げる。

その手戻りは、**どの領域のすれ違い**が原因だったか（要求の取り違い？ 設計の甘さ？ テスト不足？）を、全体像の上で特定してみましょう。

そして「もし上流のどこで気づけていれば、手戻りを防げたか」を話し合ってください。

手戻りの多くが上流に根を持つことを、実感として掴むのが狙いです。

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・ 身のまわりの「作るもの」を1つ思い浮かべ、「作る前」と「作った後」に必要なことを書き出してみる。

23-12 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ 「コードが書ける」と「働ける」は、なぜ別なのか。
- ・ ものづくりは「作る」の前後にどんな仕事があるか（思い出せるだけ）。
- ・ 「体系（先人のまとめ）」があると、なぜ助かるのか。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、次の章やこれからの仕事の中で、少しずつ埋めていきましょう。

第24章 開発プロセスとアジャイル（40日目）

🎯 この章のゴール

この章では、開発の「進め方」にはいくつかの型があること、そして「唯一の正解はない」ことを学びます。

終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ① **【知っている】** 進め方に型（ウォーターフォール／アジャイルなど）がある、と知っている。
- ② **【一言で言える】** アジャイルは「小さく作って早く確かめる」考え方、と一言で言える。
- ③ **【考え方がわかる】** 進め方に唯一の正解はなく、状況で選ぶ、とわかる。

💡 まず自己診断—読む前に、いまの自分を知らう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「開発の進め方の型を自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
- ② 「ものを作る「進め方」に、決まったやり方は1つだけだと思いますか？」を、いまの考えで一言だけ書きます（正解は気にしません）。

読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

24-1 概要—なぜプロセスを学ぶのか

第23章で、ソフトウェアづくりは「要求 → 設計 → 構築 → テスト → 運用・保守」という広がりを持つことを地図で見ました。本章のテーマは、その地図を **どんな順序・進め方で回すか、つまり開発プロセス**です。

同じ材料（要求）と同じ腕（技術）を持っていても、進め方が悪ければプロジェクトは失敗します。作る順番、誰がいつ何を決めるか、変化にどう対応するか、これらを定めるのがプロセスです。プロセスは「お役所的な手続き」ではありません。**大人数で、長期間、変化のなかで、安全にソフトウェアを作り続けるための知恵**です。

新卒がプロセスを学ぶ理由は3つあります。第一に、**自分が今どの工程にいて、次に何が起きるか**を理解できること。第二に、現場で飛び交う言葉（スプリント、CI/CD、レトロ…）の意味がわかること。第三に、そして最も大事なものは、**「なぜこの進め方をしているのか」を考えられるようになる**ことです。

💡 ポイント

本章で繰り返し立ち返る軸：プロセスにも「これが唯一の正解」はありません。

AIが開発を速くする時代でも、**「何を・どの順で・どう確かめながら作るか」という判断**は人間に残ります。

むしろ速く回せる時代だからこそ、回し方の良し悪しを見極める目（体系）が効きます。

24-2 歴史と背景—開発プロセスの系譜

開発プロセスは、「**複雑さと変化に、どう立ち向かうか**」という問いへの答えとして進化してきました。大づかみな系譜は次の通りです。

- ・ **ウォーターフォール（滝）**、要求→設計→製造→テストと、上流から下流へ順に進める古典的な型。各工程を固めてから次へ進むため計画は立てやすい反面、後から要求が変わると手戻りが大きい。
- ・ **アジャイル**、2001年の「アジャイルソフトウェア開発宣言」に象徴される、小さく作って早く確かめ、変化を歓迎する考え方。
- ・ **DevOps**、開発（Dev）と運用（Ops）の壁を壊し、自動化で素早く・安全に届け続ける文化と実践（2009年ごろから広まる）。

注意したいのは、これは「**古いものが新しいものに置き換わった**」という単純な話ではないということです。今でもウォーターフォール的な進め方が適する場面は数多くあります（後述）。歴史は「新旧の勝ち負け」ではなく、「**立ち向かう問題に応じて道具が増えてきた**」と捉えるのが正確です。

24-3 受託開発の流れ全体——一枚の絵で押さえる

まず、多くの新卒が最初に関わる**受託開発（SI）の典型的な流れ**を、一枚の絵として頭に入れましょう。進め方がアジャイルでもウォーターフォールでも、登場する工程そのものは共通です。

要件定義 → 設計 → 製造（実装） → テスト → リリース → 運用・保守

- ・ **要件定義**、顧客と「何を作るか」を決め、文書で合意する（23-4）
- ・ **設計**、「どう作るか」。全体の骨組み（アーキテクチャ）から内部構造まで（23-5）
- ・ **製造（実装）**、実際にコードを書く（23-7）
- ・ **テスト**、正しく動くか確かめる。単体→結合→システム→受入と段階を踏む（第25章）
- ・ **リリース**、本番環境に出して、利用者が使えるようにする
- ・ **運用・保守**、動かし続け、直し、育てる（23-8）

ウォーターフォールはこれを**一直線に一度だけ**通します。アジャイルはこの流れを**小さく何度も**繰り返します。DevOpsは、とくに「製造→テスト→リリース→運用」の往復を**自動化で高速・安全に**回します。どの進め方も、この一枚の絵のどこをどう回しているか、と捉えると整理できます。

💡 ポイント

この流れの各工程が「契約」「責任」とどう結びつくかは、受託では極めて重要です。本章コラムと第28章で扱います。

24-4 アジャイルの価値観

■ アジャイルとは「素早い」だけではない

「アジャイル（agile）」は「機敏な」という意味ですが、**ただ速いこと**を指すではありません。本質は、「**最初にすべてを決めきれない**」という現実を受け入れ、**小さく作って早く見せ、フィードバックで方向を修正しながら、本当に価値のあるものへ近づく**という考え方です。

■ アジャイルソフトウェア開発宣言

2001年に表明された「アジャイルソフトウェア開発宣言」は、4つの価値を示しました。**左の項目も大事だが、それ以上に右を重んじる**、という形で書かれています。

- ・ プロセスやツールよりも、**個人と対話**を
- ・ 包括的なドキュメントよりも、**動くソフトウェア**を

- ・ 契約交渉よりも、**顧客との協調**を
- ・ 計画に従うことよりも、**変化への対応**を

宣言には、これを支える**12の原則**も添えられています（顧客満足を最優先する、変化を歓迎する、動くソフトを短い間隔で届ける、関係者が日々協働する、ふりかえって改善する、など）。

新卒が誤解しやすいのは、宣言の読み方です。「ドキュメントは要らない」「計画しなくていい」という意味では**ありません**。「ドキュメントのための開発」ではなく「動くソフトのための開発」を、と**優先順位**を示しているのです。

■ リーンと継続的デリバリ

アジャイルの背景には、**リーン**（日本の製造業の生産方式に源を持つ、ムダを省き流れをよくする考え方）があります。「作りすぎのムダ」「手戻りのムダ」を減らし、価値を**継続的に・絶え間なく届ける（継続的デリバリ）**、この発想が、後述のDevOpsへとつながっていきます。

24-5 スクラムとカンバンの基礎

アジャイルは「考え方」であり、それを実践する具体的な「型（フレームワーク）」がいくつかあります。代表が**スクラム**と**カンバン**です。

■ スクラム—リズムで回す

スクラムは、開発を**スプリント**と呼ぶ一定期間（例：1～2週間）で区切り、その単位で「計画 → 開発 → 確認 → ふりかえり」を繰り返す型です。主要要素を地図として挙げます。

- ・ **役割**：プロダクトオーナー（何を作るかの優先順位に責任）／スクラムマスター（チームがうまく回るよう支援）／開発チーム
- ・ **作成物**：プロダクトバックログ（やりたいことの一覧）／スプリントバックログ（今回やること）／インクリメント（毎回の成果）
- ・ **会議（イベント）**：スプリントプランニング（計画）／デイリースクラム（毎日の短い同期）／スプリントレビュー（成果の確認）／レトロスペクティブ（ふりかえり。24-9）

ポイントは、**短いリズムで「計画と確認」を繰り返す**ことです。これにより、大きくずれる前に軌道修正できます。

💡 ポイント

デイリースクラムは「進捗報告会」ではありません。

典型的には各メンバーが15分以内で「

①**昨日やったこと**

②**今日やること**

③**困っていること（障害）**」を手短に共有し、滞っている点を早く見つけて助け合う場です。たとえば「外部APIの仕様待ちで止まっている」と一言出せば、その日のうちに誰かが動けます。長い議論はその場でせず、必要な人だけ後で集まる、これが短時間で回すコツです。

■ カンバン—流れを見える化する

カンバンは、作業を「未着手 / 進行中 / 完了」のように見える化し（カンバンボード）、**仕掛かり中の作業量に上限を設けて流れをよくする**型です。一度に多くを抱え込まず、「終わらせること」に集中します。スクラムのような時間の区切りを持たず、流れを連続的に整えるのが特徴です。

■ どちらが正解か

どちらが上ということはありません。リズムを作りたいならスクラム、流れの改善に集中したいならカンバン、と**目的に合わせて選ぶ／組み合わせる**ものです。「型を回すこと」が目的化しないよう、「何のためにこの型を使うのか」を見失わないことが大切です。

24-6 DevOpsと継続的デリバリー開発と運用をつなぐ

■ なぜDevOpsが生まれたか

かつて、**開発チーム (Dev)** と **運用チーム (Ops)** は分かれ、しばしば対立しました。開発は「早く新機能を出したい (変化)」、運用は「安定して動かしたい (安定)」、目的が逆を向いていたからです。開発が作ったものを運用に「投げて渡す」と、本番でトラブルが起き、互いを責め合う……。この壁を壊し、**両者を一つの流れとしてつなぐ文化と実践**が**DevOps**です。

DevOpsの核心は、技術以前に**文化**にあります。「自分たちが作ったものは、自分たちで動かし、責任を持つ」。開発と運用が同じゴール（利用者に価値を安定して届ける）を共有する、これが出発点です。

■ CI/CDパイプライン—自動化で速く・安全に

DevOpsを支える代表的な仕組みが、**CI/CD**です。

- ・ **CI (継続的インテグレーション)**、コードの変更を**こまめに統合**し、そのたびに**自動でビルドとテスト**を回す。「直したつもりが別を壊した」をすぐ検知できる（第25章のテスト自動化と直結）。
- ・ **CD (継続的デリバリー/デプロイ)**、テストを通った変更を、**自動で・いつでも・安全にリリース**できる状態に保つ（さらに自動で本番へ出すのが継続的デプロイ）。

これら一連の自動化された流れを**パイプライン**と呼びます。人手の作業を自動化することで、「速さ」と「安全（人為ミスの削減）」を同時に得るのが狙いです。**速さと品質はトレードオフではなく、自動化によって両立できる**、これがDevOpsの重要なメッセージです。

💡 ポイント

ここでは「プロセスとしての回し方」を扱います。バージョン管理やコードレビューといった**構成管理の概念・作法**は23-7で学びました。本章はそれを「どう運用に乗せるか」の話です。

24-7 構成管理をプロセスに組み込む—ブランチ戦略・プルリクエスト・GitOps

23-7で、変更履歴を管理する**構成管理 (Git等)** の概念を学びました。本節は、それを**チームのプロセスとしてどう回すか**を扱います。

- ・ **ブランチ戦略**、複数人が同時に開発するとき、作業を枝分かれ（ブランチ）させ、安全に本流へ合流させる**ルールの決め方**。「いつ枝を切り、いつ・どうやって本流に戻すか」をチームで揃えることで、混乱と衝突を防ぎます。代表的な型がいくつかありますが、大事なのは「**チームで一つのやり方に合意し、守る**」ことです。
- ・ **プルリクエスト (マージリクエスト) 運用**、自分の変更を本流に取り込む前に、**レビューを依頼し、自動テストを通す**仕組み。コードレビュー（23-7）と自動チェック（CI）を、変更を取り込む手続きそのものに**組み込む**ことで、品質を仕組みで守ります。
- ・ **GitOps**、構成管理（Git）を「正（ただししい状態）の置き場所」とし、**Gitへの変更をきっかけに、環境への反映を自動化する**考え方。「何が正しい状態か」をGitで一元管理し、人の手作業を減らします。

これらに共通するのは、「**良い習慣を、個人の心がけでなく、仕組み（プロセス）として強制する**」という発想です。人は忘れ、間違えます。だからこそ、レビューやテストを「やったほうがいい努力目標」ではなく「**通らなければ先に進めない関所**」として組み込む、ここに、プロセス設計の知恵があります。

24-8 ウォーターフォールとアジャイルの使い分け・ハイブリッド

■ どちらが優れているのか？

ここまでアジャイルを多く語りましたが、「**アジャイルが正しく、ウォーターフォールは古くて悪い**」という理解は誤りです。両者は、向いている状況が異なります。

- ・ **ウォーターフォールが向く**：要求が初めから明確で変わりにくい／全体計画と文書が重視される／途中で大きく変えにくい（例：法規制対応、ハードウェアと連動する開発、安全性が最重要のシステム）
- ・ **アジャイルが向く**：要求が固まりきらない／早く出して市場やユーザーの反応で学びたい／変化が前提（例：新規Webサービス、改善を続けるプロダクト）

■ V字モデルとハイブリッド

ウォーターフォールを、品質保証の視点で描き直したのが**V字モデル**です。左側を上流から下流へ降りる開発工程（要件→基本設計→詳細設計→製造）、右側を下流から上流へ昇るテスト工程（単体→結合→システム→受入）とし、**同じ高さの左右を対応づけます**（例：要件定義 ↔ 受入テスト）。「各設計工程に、それを検証するテストが対応する」という考え方で、組み込みや大規模システムで広く使われます。

ウォーターフォール：一直線に一度だけ



計画→開発
→確認→改善

アジャイル：小さく何度も繰り返す

V字モデル：左の各工程と、同じ高さの右のテストが対応する



例：要件定義 ↔ 受入テスト / 詳細設計 ↔ 単体テスト

図 開発プロセス（ウォーターフォール／アジャイル）とV字モデル

現実のプロジェクトでは、両者を組み合わせる**ハイブリッド**も一般的です。たとえば「全体は計画的に（ウォーターフォール）進めつつ、各部分の作り込みは反復的に（アジャイル）回す」といった形です。**型に現実を合わせるのではなく、現実に合わせて型を選び・混ぜる**、これがプロの態度です。

知っておこう

〔参考情報〕 **Automotive SPICE (A-SPICE)**、**組込み・車載のデファクト・プロセス**

世の中はアジャイル一辺倒ではありません。

とくに**自動車（車載ソフト）**のように安全性が最重要で、ハードウェアと密に連動する組込み開発では、**段階を踏む厳密なプロセス（V字モデル）**と、その**プロセスの能力を客観的に評価する枠組み**が重視されます。

その代表が **Automotive SPICE (A-SPICE)** です。

- **何か**：自動車業界・サプライヤ向けの**プロセスアセスメント／参照モデル**。

完成車メーカー（OEM）が、サプライヤの開発プロセスの成熟度を評価する際に用い、車載開発では**事実上の標準（デファクト）**になっています。

- **出自**：ドイツ自動車工業会の品質管理組織（**VDA QMC**）が公開。

国際規格 **ISO/IEC 15504 (SPICE)**／現在の **ISO/IEC 330xx** 系に準拠し、プロセス自体は **ISO/IEC 12207**（ソフトウェアライフサイクル）をベースに自動車向け拡張を加えたもの。

- **考え方**：「プロセス次元（どんなプロセスがあるか）」と「能力次元（どれだけ成熟して回っているか）」の2軸でとらえ、能力を**0～5の6段階**で評価します（おおむね、L0=実施されていない → L1=実施されている → L2=計画・管理されている → L3=組織で標準化されている → L4=定量管理 → L5=継続的改善）。

プロセス成熟度を段階で捉えるモデルとしては、**CMMI（能力成熟度モデル統合）**も同じ系譜の有名な枠組みです。

新卒は詳細を覚える必要はありません。

押さえてほしいのは、「**アジャイルが合う領域もあれば、A-SPICEのように厳密なプロセスと客観評価が求められる領域もある**」という事実です。

配属先によっては、こうしたプロセス標準への準拠が日常になります（版や評定の詳細は実務・別科目で）。

24-9 ふりかえりとカイゼン—レトロスペクティブ

どんな進め方をとるにせよ、最も大切な習慣が**ふりかえり（レトロスペクティブ）**です。一定の区切り（スプリントの終わりなど）で、チームが立ち止まり、「**うまくいったこと／いかなかったこと／次に試すこと**」を話し合い、自分たちのやり方そのものを改善します。

ここで効くのが、第25章で学ぶ品質管理の源流、**カイゼン（継続的改善）**と**TQM（総合的品質管理）**の思想です。「品質はプロセスで作り込む／全員参加で高める」という考えは、製品だけでなく**チームの働き方そのもの**にも当てはまります。ふりかえりは、いわば「チームの働き方の品質」を、全員参加で少しずつ良くしていく営みです。

ふりかえりを健全に行う鍵は、第25章でも触れる**心理的安全性**です。「失敗を責める場」になれば、人は本音を言わなくなり、改善は止まります。**人を責めず、仕組みを直す**、この一点を守れるかどうかで、ふりかえりが宝になるか、形だけの儀式になるかが決まります。第21章の「誠実さ（都合の悪い事実こそ共有する）」が、ここでも土台になります。

24-10 受託でアジャイルをやる難しさ—契約・準委任

最後に、受託開発ならではの難所に触れます。アジャイルは「変化を歓迎する」進め方ですが、受

託は「何を・いくらで・いつまでに作るか」を契約で取り決める」世界です。ここに緊張があります。

- ・ **請負契約**は、原則「**完成した成果物**」に対して責任と対価が定まります。「何を作るか」を先に固める必要があり、「変化を歓迎する」アジャイルと相性が悪い面があります。

契約形態の違い — 「誰が指示するか」「完成責任があるか」

	請負契約	準委任契約	(労働者) 派遣
指揮命令する人	受託側（自社）	受託側（自社）	派遣先
成果物の完成責任	あり	なし（善管注意義務）	なし
対価の対象	完成した成果物	業務の遂行	労働力の提供

※ 偽装請負（契約は請負なのに発注側が直接指示）に注意

図 契約形態の違い（請負・準委任・派遣）

- ・ **準委任契約**は、「成果物の完成」ではなく「**業務の遂行（労務の提供）**」に対する契約です。要求が変わりうるアジャイル開発では、こちらのほうが馴染みやすいとされます。

ここで現場に多いすれ違いが、「**契約は準委任なのに、実際には成果物の完成を暗に求められる**」というものです。準委任は本来「業務を誠実に遂行すること」に対する契約で、請負のような完成責任までは負いません。それでも顧客が請負と同じ「完成」を期待していると、「やると言ったのに終わっていない」という認識のずれがトラブルになります。だからこそ、**契約形態が何を意味するか（何に対して責任を負うのか）を、現場の進め方とお金の両面で揃えておくことが大切です**（偽装請負と並ぶ頻出の論点。第28章）。

つまり、受託でアジャイルを成立させるには、**進め方（アジャイル）と契約形態（請負／準委任）の整合**が欠かせません。

新卒がここで学ぶべきは、「**良い進め方**」も、**契約・お金・責任という土台と噛み合っ**て初めて**機能する**ということです。技術・プロセス・契約は、つながっています。

〔コラム〕受託アジャイルの落とし穴／客先指定のランチ運用・CI

受託アジャイルの落とし穴。受託でアジャイルを掲げたものの、実態は「**ただの行き当たりばった**り」になってしまう、これはよくある失敗です。アジャイルは「計画しない」「ドキュメントを書かない」ことの言い訳ではありません（24-4）。むしろ、短いリズムでの計画・確認・ふりかえりという規律を要求します。「アジャイルだから仕様書はなくていい」「アジャイルだから見積もりは要らない」と顧客・現場が都合よく解釈すると、スコープは際限なく膨らみ（スコープクリープ）、誰も全体を把握できなくなります。受託では、変化を扱うための**最低限の合意の仕組み**（何を・どの優先順で・どこまで今回やるか）を、顧客とともに持つことが欠かせません。

客先指定のランチ運用・CI。受託・客先常駐では、開発の進め方そのものが**顧客に指定される**ことが少なくありません。「ランチはこの戦略で」「コミットメッセージはこの書式で」「CIはこのツールで、このチェックを通すこと」、自社の慣れたやり方と違って、まずは**客先のルールに従い、正確に守る**のがプロです。やり方の善し悪しを論じる前に、「決められた作法を、漏れなく守れる」ことが信頼の第一歩になります（改善提案は、信頼を得てから、相手の土俵の言葉で）。

 ポイント

プロセスは「自社の好み」ではなく「その現場で合意されたルール」。郷に入っては郷に従う、その上で、より良い回し方を提案できる人が、受託で重宝されるエンジニアです。

24-11 考えてみよう（グループ討議）

教室内では手を動かす実技は行いません。3～4人のグループで、考えを言葉にしてみましょう。

◆ 考えてみよう：討議1：受託でアジャイルを成立させる条件

「受託開発でアジャイルをうまく回すには、何が必要か」をグループで洗い出す。

24-10（契約・準委任）、コラム（落とし穴・客先ルール）をヒントに、「顧客側に必要なこと」「**自社・チームに必要なこと**」「**契約面で必要なこと**」の3つの観点で条件を挙げてみてください。「変化を歓迎する」と「契約で約束する」の緊張を、どう両立させるかが論点です。

◆ 考えてみよう：討議2：この進め方は、何に向いているか

身近な「ものづくり」や「プロジェクト」（学園祭の出し物、制作課題、アルバイト先の改善など）を一つ選ぶ。

それは**ウォーターフォール的**（先に全部決めて一直線）と**アジャイル的**（小さく試して直す）のどちらが向いていたか、なぜそう思うかを話し合ってください。

24-8の「使い分け」を、身近な例で実感するのが狙いです。

A-SPICEのように「厳密な手順と検証」が必要な場面（安全に関わる等）も想像してみましょう。

◆ 考えてみよう：討議3：その関所は、何を守っているか

CI・コードレビュー・プルリクエストといった「通らなければ先に進めない関所」（24-6, 24-7）を一つ取り上げる。

その関所が「もしなかったら」どんな問題が起きるかを想像し、関所が**何を・誰を守っているのか**を言葉にしてください。「面倒な手続き」に見えるものの裏にある意図を読み解く練習です。

 事例から学ぶ：旅行の計画にたとえると

旅行には2つのやり方があります。

出発前にすべての宿・電車・店を予約してその通りに動くやり方（ウォーターフォール）と、初日の宿だけ決め、現地で様子を見ながら次を決めるやり方（アジャイル）です。

どちらが良い・悪いではなく、行き先がはっきりしているか、途中で変わりそうかで向き不向きが変わります。

ソフトウェアの作り方も、これと同じ考え方です。

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構

いません。

- ・ 身近な作業を1つ選び、「先に全部決める」と「小さく試しながら進める」の、どちらが向くか考えてみる。

24-12 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ 開発の進め方の型を、思い出せるだけ。
- ・ アジャイルを一言で言うとは？
- ・ なぜ「進め方に唯一の正解はない」のか。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、次の章やこれからの仕事の中で、少しずつ埋めていきましょう。

第25章 品質管理とソフトウェアテスト（40日目）

🎯 この章のゴール

この章では、「品質」とは何か（お客さんにとっての価値）、そして品質は「作る過程で作り込む」ものを学びます。

終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ①【**考え方がわかる**】品質とは結局「お客さんにとっての価値」、という考え方がわかる。
- ②【**たとえられる**】品質は「テストで作る」のではなく「作る過程で作り込む」、と言える（味見のたとえ）。
- ③【**感覚がわかる**】不具合は早く見つけるほど安い、という感覚がわかる。

💡 まず自己診断—読む前に、いまの自分を知ろう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「品質とは何かを自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
- ② 「「品質が良い」とは、どういうことだと思いますか？」を、いまの考えで一言だけ書きます（正解は気にしません）。

読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

25-1 概要—なぜ品質が最優先か／QA（作り込む）とテスト（検出）の違い

■ 品質第一は「当たり前」—品質とは顧客価値そのものだから

なぜ品質が最優先なのか。突き詰めると、答えはこうです。**品質とは、顧客満足であり、顧客価値そのものだから**です。

「品質が高い」とは、欠陥が少ないという技術的な話に閉じません。それは「**顧客が求めるものを満たし、顧客が満足し、顧客にとっての価値になっている**」ことを意味します（第23章で学んだ「要求」とも一直線につながります。要求とは「顧客にとっての価値」を言葉にしたものでした）。

ここで思い出してほしいのが、第21章の「作業でなく仕事をする」です。**エンジニアの仕事とは、価値を作り出すこと**にほかなりません。価値を作り出すのがエンジニアなのだから、その価値の中身である品質を第一に置くのは、特別な心がけではなく**当たり前**のことなのです。「品質か、それ以外か」を天秤にかけている時点で、すでにずれている、品質は、エンジニアが生み出すものの本体です。

だからこそ、受託開発（SI）の世界の古い言葉も腑に落ちます。「**品質・コスト・納期（QCD）は大事だが、最後に残るのは品質だ**」。納期に間に合っても、コストが安くても、**品質の悪いもの＝顧客価値のないものを納めれば、信頼は失われ、後の手直しでかえって高くつきます**。一度「あの会社のものは品質が低い」と思われたら、その評価を覆すのは容易ではありません。品質を作り込むことは、顧客価値を作り、信頼を築く、エンジニアの仕事の核心そのものなのです。

ここで新卒に最初に整理してほしいのが、よく混同される2つの言葉の違いです。

- ・ **品質保証（QA：Quality Assurance）＝品質を「作り込む」**

開発の最初から最後まで、プロセス全体を通じて品質を確保する活動。要求の段階から「良いものを作る仕組み」を整える、**予防**の考え方です。

・ **テスト (Testing) = 品質を「検出する」**

作ったものに欠陥がないかを確認、見つける活動。QAの一部であり、**確認・検出**の手段です。

この2つの関係を、料理にたとえてみましょう。テストは「**できあがった料理の味見**」です。味見は大切ですが、味見をいくら繰り返しても、まずい料理が美味しくなるわけではありません。美味しさは、食材選び・下ごしらえ・調理という**プロセス全体**で作られます。ソフトウェアも同じで、**品質はテストで作られるのではなく、開発プロセス全体で作られる**のです。これが本章を貫く、最も大事な考え方です。

💡 **ポイント**

AI時代の補足：AIは大量のコードやテストを高速に生み出せます。

だからこそ「**何をもって"良い"とするか（品質の基準）**」「**何を確かめるべきか（テストの観点）**」を決める人間の判断が、ますます重要になります。

AIが生成したコードやテストを鵜呑みにせず評価できるのは、品質の体系を持つ人だけです。

ここでも「AI時代にこそ体系が効く」が当てはまります。

25-2 歴史と背景—デミングからTQMへ／テストの目的の変遷

品質管理の考え方には、ものづくりの歴史が流れ込んでいます。統計的品質管理を築いたシューハート、それを発展させた**W.E.デミング**。1950年に日本科学技術連盟（日科技連）がデミングを招いた品質管理セミナーは、日本の品質管理が根づく契機になったとされます。やがて日本で全社的品質管理（TQC）が発展し、**TQM（総合的品質管理）**として世界へ広がりました。その精神は「**品質は検査で選り分けるのではなく、プロセスで作り込む／全員参加で高める**」というものです。

注目したいのは、この「**作り込む／全員参加**」という思想が、製造業からソフトウェアへ受け継がれている点です。25-1で述べた「品質はプロセス全体で作る」も、「品質は品質保証部門だけの仕事ではなく、開発者全員の責任」も、源流はここにあります。

ソフトウェアの**テスト**の目的も、時代とともに高度化してきました。おおまかには「**デバッグ（不具合を取る）→ 不具合の検出（積極的に見つける）→ 品質の証明・予防**」へと、捉え方が深まってきたとされます。とくに大事な発想の転換が、「**テストは"欠陥がないこと"を示すためではなく、"欠陥を見つける"ために行う**」というものです。欠陥を見つけられたテストは「成功」、この一見逆説的な見方が、現代テストの土台にあります。

25-3 品質管理の体系—SQuBOKと品質特性（ISO/IEC 25010:2023）

■ SQuBOK—日本発の品質知識体系

品質管理にも、第23章で見たような知識体系があります。日本発の**SQuBOK（スクボック／ソフトウェア品質知識体系ガイド）**です。日科技連が策定し、最新は**第3版（V3、2020年11月発行）**。品質マネジメント（仕組み・進め方）と品質技術（具体的な技法）の両輪を体系化しています。第23章のSWEBOKが「作り方全体」の地図なら、SQuBOKは「品質」を深掘りした地図、と位置づけられます。第3版では、ユーザビリティ・セーフティ・セキュリティといった専門的な品質や、AI・IoT・クラウドといった新しい応用領域も扱われるようになりました。

■ 品質には「測れる軸」がある—ISO/IEC 25010:2023

「品質が良い」とは、漠然とした言葉ではありません。国際規格 **ISO/IEC 25010** は、ソフトウェアの製品品質を**複数の特性（軸）**に分けて定義しています。2023年の改訂版では、製品品質モデルは**9つの特性**で整理されています。

1. **機能適合性**、必要な機能が正しく備わっているか
2. **性能効率性**、速さ・資源の効率
3. **互換性**、他のものと共存・連携できるか
4. **相互作用性**、使いやすさ（旧「使用性 Usability」を改称）
5. **信頼性**、落ちずに正しく動き続けるか
6. **セキュリティ**、安全に守られているか（第26章）
7. **保守性**、直しやすさ・育てやすさ
8. **柔軟性**、環境変化への適応（旧「移植性 Portability」を改称）
9. **安全性 (Safety)**、2023年版で新設。人や環境への危害を防ぐ

新卒が9特性を暗記する必要はありません。大切なのは、「**品質には機能の正しさ以外にも、速さ・使いやすさ・安全性など複数の軸がある**」と気づくことです。「品質が良い」と言うとき、自分はどの軸の話をしているのか、これを意識できるだけで、議論が具体的になります。

とくに、機能の正しさ以外の軸（性能・信頼性・セキュリティ・保守性・安全性など）は、23-4で学んだ**非機能要件**にそのまま対応します。非機能要件は見落とされやすいので、品質を語るときの**チェックリスト**として、この9特性を使うとよいでしょう。

知っておこう

規格は改訂されます。「2023年版では9特性」のように、年・版を意識して扱ってください。古い資料は8特性（2011年版）で書かれていることがあります。

25-4 品質を作り込む—コスト・メトリクス・文化

■ 品質コスト—「早く見つけるほど安い」

品質には費用がかかりますが、その構造を知ると判断が変わります。古くから知られる経験則として、**欠陥は見つかるのが遅いほど、修正コストが跳ね上がる**というものがあります。よく語られる桁感はこうです。同じ1件の欠陥でも、**要求段階で気づいて直せば「1」**の手間で済むものが、**設計段階では数倍、コードになってからは十倍前後、リリース後に発覚すると数十～百倍**にまで膨らむ、というものです（あくまで桁のイメージで、正確な倍率は状況によります）。リリース後の修正には、調査・修正・再テストに加え、顧客対応・信頼回復まで乗ってくるからです。「上流の1の手間を惜しんで、下流で100払う」、これが品質を軽視したときに起こることです。

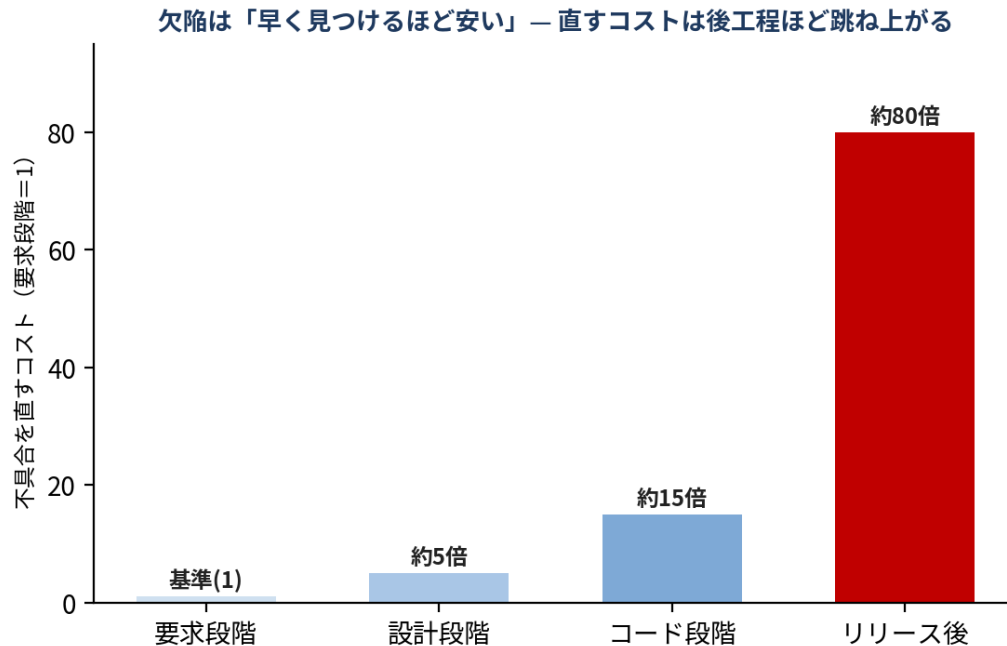


図 欠陥を直すコストは後工程ほど跳ね上がる

品質コストは、ざっくり次のように分けて考えられます。

- ・ **予防のコスト** (良いものを作る仕組み：教育・レビュー・標準化など)
- ・ **評価のコスト** (確かめる：テスト・検査)
- ・ **失敗のコスト** (不具合への対応：手戻り・障害対応・信頼の損失)

予防と評価に適切に投資するほど、失敗のコストは下がります。「テストする時間をもったいない」のではなく、「テストしない方が、後で高くつく」、この感覚が大切です（討議で扱います）。品質は「コスト」ではなく「投資」だ、と捉え直してください。

■ メトリクス—勘ではなく事実で語る

品質は「測って」改善します。欠陥の数や発生箇所、テストがどれだけ網羅できているか（カバレッジ）などの**メトリクス（指標）**を使い、勘ではなく事実で語ります。

ただし、メトリクスには落とし穴もあります。**数字が目的化すると、人は数字に合わせて行動を歪めます**（「テスト件数」をノルマにすると、意味の薄いテストが量産される、など）。メトリクスは「現実を映す鏡」であって「達成すべきノルマ」ではない、この区別が、健全な測定には欠かせません。

■ 文化—心理的安全性が品質を支える

最後に効くのは**文化**です。とくに**心理的安全性**、「不具合や失敗を、責められずに報告できる空気」、が品質を支えます。ミスを隠す組織では欠陥が地下に潜り、もっと大きな事故になります。逆に「早く言ってくれてありがとう」と言える組織では、問題が小さいうちに表に出ます。

第21章で学んだ「都合の悪い事実こそ早く正直に共有する（誠実さ）」が、品質文化の土台です。そして「**人を責めず、仕組みを直す**」、この姿勢は、第24章のふりかえり（カイゼン）や、第23章で触れた障害対応の記録とも一直線につながります。品質は、技術であると同時に、チームの文化なのです。

25-5 ソフトウェアテストの体系—JSTQB（CTFL）とISO/IEC/IEEE 29119

ここからは、品質を「検出」する側、**テスト**の体系を扱います。テストもまた、属人的な勘ではな

く、**共通言語と体系**を持っています。本節は、新卒・若手がまず学ぶ国際標準的な入口である **JSTQB認定テスト技術者資格 Foundation Level (CTFL : Certified Tester Foundation Level)** の体系に沿って、その骨格を概観します（資格そのものは25-7で扱います）。

■ テストの共通言語—JSTQB/ISTQBと国際規格

- ・ **JSTQB/ISTQB**、テスト技術者の知識を体系化し、資格として認定する仕組み。**ISTQB**が国際組織、**JSTQB**がその日本組織で、用語・プロセス・技法に**世界共通の言葉**を与えています。新卒の入口が **Foundation Level (CTFL)** で、その学習内容（シラバス）は、本節で見る骨格そのものです（本書記載時点の最新は **CTFL v4.0**＝2023年版。日本ではv4.0準拠の試験が2024年11月から。版は更新されるので最新を確認）。
- ・ **ISO/IEC/IEEE 29119**、ソフトウェアテストの**国際規格**（全5部：概念と用語／テストプロセス／テストドキュメント／テスト技法／キーワード駆動テスト）。「どんなテストにも使える、国際合意のテストの型」です。

新卒は中身を暗記する必要はありません。「テストにも世界共通の体系と用語がある」と知り、CTFLの骨格を俯瞰できれば十分です。以下、CTFLの主要な柱（テストの基礎・SDLC（ソフトウェア開発ライフサイクル）全体を通したテスト・静的テスト・テスト分析と設計・テストのマネジメント・ツール）に沿って見ていきます。

■ テストの基礎—テストの7原則

CTFLは、まず「テストとは何か」を押さえます。その核に、有名な**テストの7原則**があります（CTFL v4.0準拠。新卒がとくに腑に落としたい考え方です）。

1. **テストは欠陥があることは示せるが、欠陥がないことは示せない**、テストを通っても「バグゼロの証明」にはならない
2. **全数テストは不可能**、あらゆる入力のは試せない。だから賢く絞る（後述の技法）
3. **早期テストで時間とコストを節約する**、上流で早く確かめるほど安い（シフトレフト。25-4の品質コストと一致）
4. **欠陥の偏在（クラスタリング）**、欠陥は特定の箇所に集中しがち。そこを重点的に攻める
5. **テストの弾力性の低下（殺虫剤のパラドックス）**、同じテストを繰り返すだけでは、新しい欠陥を見つけられなくなる。テストも見直し続ける
6. **テストは状況次第**、最適なテストは、対象や文脈（安全重視か、素早さ重視か等）によって変わる
7. **「欠陥ゼロ」の落とし穴**、欠陥がなくても、要求やニーズを満たさなければ意味がない（25-1の「品質＝顧客価値」と一致）

■ SDLC全体を通したテスト—テストレベルとテストタイプ

テストは、確かめる「単位の大きさ」で段階に分けられます（**テストレベル**。CTFL v4.0の用語）。

- ・ **コンポーネントテスト**（従来「単体／ユニットテスト」）、部品ひとつが正しく動くか
- ・ **コンポーネント統合テスト**（従来「結合テスト」の一部）、部品どうしをつないで正しく連携するか
- ・ **システムテスト**、システム全体が要求を満たすか
- ・ **システム統合テスト**、他システム・外部サービスとの連携が正しいか
- ・ **受入（うけいれ）テスト**、利用者・顧客の視点で、使えるものになっているか（受託では検収につながる重要な関所）

小さい単位から確かめ、徐々に大きくしていくと、問題の原因を切り分けやすくなります（第21章のデバッグの考え方と同じ）。また、確かめる「観点」による分け方が**テストタイプ**で、**機能テスト**（何ができるか）、**非機能テスト**（速さ・使いやすさ・安全性など。25-3の品質特性に対応）、**ブラックボックステスト**（中身を見ず外から）、**ホワイトボックステスト**（中身の構造に基づく）などがあります。

新卒が現場で最初に出会う重要な2つも押さえておきましょう。修正後に「その不具合が本当に直ったか」を確かめるのが**確認テスト（リテスト）**、修正によって「**別のところを壊していないか**」を確かめるのが**リグレッションテスト（回帰テスト）**です。とくにリグレッションテストは、変更のたびに繰り返し必要になるため、後述の自動化・CIともっとも相性がよい領域です。「直したつもりが別を壊す」事故は頻発するので、この2つはセットで覚えてください。

■ 静的テスト—動かさずに見つける

CTFLは、プログラムを動かす「動的テスト」だけでなく、**動かさずに成果物を確かめる「静的テスト」**も重視します。これはレビューや静的解析にあたり、欠陥を上流で安く見つけれられます。詳細は次の25-6で扱います。

■ テスト分析と設計—テスト技法

「どこを・どう確かめるか」を、勘でなく系統立てて選ぶための**技法**があります。CTFLは大きく3つに分類します（深掘りと実装は別科目・OJTへ）。

- ・ **ブラックボックス技法**（中身を見ず、入出力の仕様から設計）
- ・ **同値分割法**、入力を「同じ結果になるグループ」に分け、代表値で確かめる
- ・ **境界値分析**、間違いが起きやすい「境目」（上限ぴったりとその前後など）を重点的に確かめる
- ・ **デシジョンテーブルテスト**、条件の組み合わせを表にして、漏れなく確かめる
- ・ **状態遷移テスト**、状態の移り変わり（ログイン前/後など）に沿って確かめる
- ・ **ホワイトボックス技法**（中身の構造に基づく）
- ・ **ステートメントテスト／ブランチテスト**、コードの命令や分岐をどれだけ通したか（網羅率＝カバレッジ）で測る
- ・ **経験ベースの技法**（人の知見を活かす）
- ・ **エラー推測／探索的テスト／チェックリストベースドテスト**、経験から「ありそうな弱点」を狙う

技法の根っこにある問いは一つです。「**すべては試せない（原則2）。では、限られた手数で、どこを突けば欠陥を最も多くあぶり出せるか**」。テストは「全部やる」ものではなく、「**賢く絞る**」ものなのです。なお、CTFL v4.0では、開発初期に**受入基準やユーザーストーリーを用いてテストを考える協調的なアプローチ**（ATDD＝受け入れテスト駆動開発など）も扱われ、「テストは後工程」ではなく「最初から作り込みに関わる」という現代的な位置づけが強まっています（第24章のアジャイル・25-1のQAと一致）。

■ テストのマネジメントと自動化・ツール

CTFLは、テストの**計画・モニタリング・欠陥マネジメント**、そして**リスクに基づいて優先順位をつける考え方（リスクベースドテスト）**も扱います。「すべてを一律にテストせず、影響と発生可能性の高いところを重点的に」、これは第23章のリスク管理とも通じます。

また、繰り返すテスト（とくにコンポーネントテスト）は、**自動化**して変更のたびに自動で回す仕組み（**CI＝継続的インテグレーション**）に乗せると効果が跳ね上がります。「直したつもりが別を壊

した」をすぐ検知できるからです（CI/CDのプロセス面は第24章）。一般に、土台の自動テストを厚く、上位を薄く積む形（テストピラミッド）が効率的とされます。CTFLでも**テストツール**は独立した柱の一つですが、ツールはあくまで手段であり、「何を・なぜテストするか」を考える人の判断が主役である点は変わりません。

テストピラミッドー土台（単体テスト）を厚く、上ほど薄く

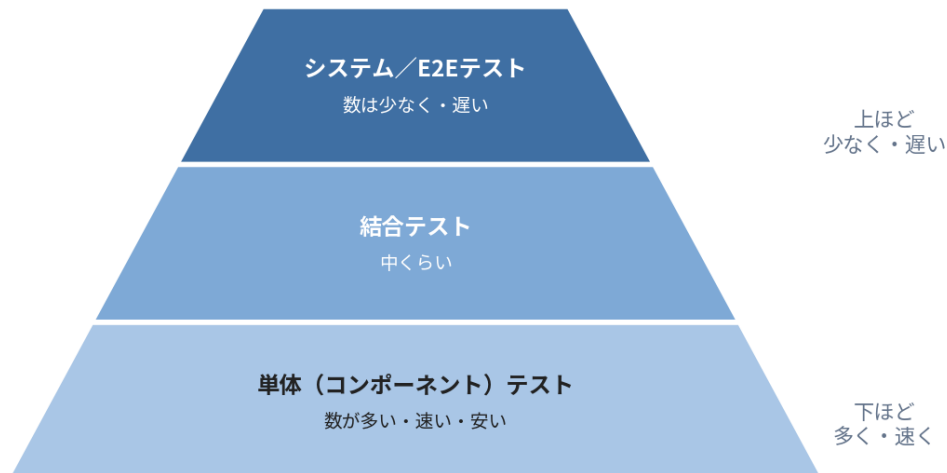


図 テストピラミッド

これらCTFLの骨格は、25-1～25-4で述べた品質の考え方と、見事に響き合っています。テストは「品質を検出する」体系的な営みであり、その共通言語を持つことが、第一歩になります。

25-6 レビューと静的解析—動かす前に見つける

テストの多くは「動かして」確かめますが、**動かす前に欠陥を見つける**強力な手段があります。それが**レビュー**と**静的解析**です。第23章で触れたコードレビューを、品質の観点から深めます。

■ レビューの種類

成果物（要求書・設計書・コードなど）を人の目で確かめる活動がレビューです。形式の硬さに応じて呼び名があります。

- ・ **ウォークスルー**、作成者が説明し、参加者が気づきを述べる、比較的ゆるやかな形式
- ・ **インスペクション**、役割と手順を決め、記録を取りながら系統的に行う、最も厳格な形式

レビューの利点は、**動かなくても・コードになる前でも欠陥を見つけられる**ことです。要求書のレビューで誤りを潰せれば、25-4で見た「上流で直すほど安い」を地で行けます。

■ 観点表—見るべき所を漏らさない

レビューを「なんとなく眺める」で終わらせないために、**観点表（チェックリスト）**を使います。「この種の成果物では、ここここを必ず見る」という観点をあらかじめ用意しておくことで、人による見落としのばらつきを減らせます。これも「勘でなく体系で」の一例です。

■ 静的解析

プログラムを**実行せずに**、ツールでコードを自動的に調べ、怪しい箇所（バグの兆候・規約違反・危険なパターン）を機械的に指摘するのが**静的解析**です。人のレビューと組み合わせると、機械が得意な網羅的チェックを任せ、人は設計の意図など本質に集中できます。

25-7 品質・テストは、全エンジニアの「当たり前の技術」

■ もはや専門職だけのものではない

かつて、テストや品質保証は「**専門の担当者（テストエンジニア・QAエンジニア）の仕事**」と考えられがちでした。開発者はコードを書き、テストは別の人に任せる、そういう分業のイメージです。

しかし、いまや状況は変わりました。本章でここまで見てきた知識、QAとテストの違い、品質特性、品質コスト、テストレベルと技法、レビュー、自動化とCI、は、**テストエンジニアやQAエンジニアでなくても、エンジニアであれば実践して当たり前の技術**になっています。

その背景には、本章や第24章で見てきた変化があります。**コンポーネント（単体）テストは作った本人が書くのが当たり前になり、CIで変更のたびに自動でテストを回し（24-6）、コードレビューで全員が品質を見る（23-7）**。テストや品質の作り込みは、開発の流れそのものに織り込まれました。「自分はテスト担当ではないから関係ない」は、もはや通用しません。**自分が書いたものの品質に、自分で責任を持つ**、これが現代エンジニアの標準的な構えです（25-1の「品質はエンジニアの仕事の本体」と一直線につながります）。

専門のQA／テストエンジニアは、もちろん今も重要です。彼らはより高度な品質戦略やテスト設計を担います。しかしそれは「品質は専門家に丸投げ」を意味しません。**全員が品質の作り込みを実践し、その上で専門家がさらに深める**、という関係です。

■ さらに学ぶための道標

その上で、品質・テストを体系的に学び続けたい人のために、具体的な道標を示します。**受験が必須というわけではありませんが**、「次に何を学ばいいか」の目印になります。

- ・ **JSTQB認定テスト技術者資格（Foundation Level）**、テストの基礎的な共通言語を体系的に学べる、新卒・若手の入口。国際資格ISTQBと整合しており、世界で通用します。上位にAdvanced Levelがあります。
- ・ **JCSQE（ソフトウェア品質技術者資格）**、SQuBOKを基盤とする、品質技術者の資格（日科技連）。初級・中級などがあります。

資格そのものが目的ではありません。大切なのは、「**品質とテストには、こうして体系的に学べる道が整っている**」と知り、必要に応じてその地図を開けることです。そして、その基礎（Foundationレベルの知識）は、いまや**どのロールのエンジニアにとっても共通の教養**になりつつあります。第30章の資格戦略ともあわせて考えてみてください。

（コラム）品質＝契約不適合責任と検収

本章の品質論は普遍的ですが、**受託開発（SI）**では、品質が**契約・お金**に直結します。

受託では、納めた成果物が「契約で約束した内容（仕様）」を満たしているかを顧客が確認する手続きを**検収**と呼びます。そして、納めたものが契約の内容に適合していなかった場合の作り手の責任を、法律では**契約不適合責任**といいます（この用語・制度の詳細は第28章で扱います。ここでは「品質が法的責任に直結する」という接続だけ押さえてください）。

つまり受託エンジニアにとって、品質は「良心」や「プライド」の問題であると同時に、**契約上の義務**でもあるのです。だからこそ、

- ・ 25-1の「品質はプロセス全体で作り込む」は、検収を一発で通すための備えになる
- ・ 第23章の「仕様を文書で握る」は、「何をもって適合とするか」の基準そのものになる
- ・ 25-4の「上流で早く欠陥を潰す」は、納品後の手直し（無償対応になりがち）を減らし、利益を守る

品質を作り込めるエンジニアは、会社の信頼と利益を守るエンジニアでもあります。「テストは面倒な作業」ではなく、「自分と会社を守る仕事」なのだと捉え直してください。

💡 ポイント

「作業者」は「動いたから完成」と考えますが、「エンジニア」は「契約で約束した品質を満たし、検収を通り、その後も問題を出さない」までを見据えます。品質への向き合い方に、その差がはっきり表れます。

25-8 考えてみよう（グループ討議）

教室内では手を動かす実技は行いません。3～4人のグループで、考えを言葉にしてみましょう。

✦ 考えてみよう：討議1：テストを省いた失敗を「品質コスト」で説明する

身の回り（アルバイト・制作・学業、または見聞きた話）で、「**確認を省いたために、後で大きな手戻りになった**」経験の一つ取り上げてください。そして、

- ・ もし「確認（テスト・レビュー）」をしていたら、どこで・どれだけの手間で防げたか（**予防・評価のコスト**）
- ・ 実際には、後でどれだけの手間・損失になったか（**失敗のコスト**）

を見比べ、25-4の「品質コスト」の言葉で説明してみてください。「早く確かめるほど安い」を、自分の体験で腑に落とすのが狙いです。

✦ 考えてみよう：討議2：その「品質が良い」は、どの軸の話か

「あのアプリは品質が良い／悪い」と感じたサービスを、各自ひとつ挙げる。

その「良い・悪い」は、25-3の**9つの品質特性**のうちどの軸の話か、グループで分類してみましょう。

「速い（性能効率性）」「使いやすい（相互作用性）」「よく落ちる（信頼性）」など、漠然とした感想を**品質の言葉に翻訳する**練習です。

✦ 考えてみよう：討議3：全部はテストできない。どこを突くか

あるログイン画面（IDとパスワードを入れる）をテストするとして、限られた時間で「ここは必ず確かめる」という観点を挙げる。

25-5の技法（同値分割・境界値・条件の組み合わせ）を思い出しながら、「賢く絞る」とはどういうことかを話し合ってください。

「思いつく限り全部」ではなく、「**欠陥が潜みやすい所を狙う**」発想に立てるかが鍵です。

📦 事例から学ぶ：出す前に、ひと手間

料理は出す前に味見をし、レポートは提出前に読み返し、出かける前に持ち物を確かめます。

どれも「作り終えてから、世に出す前に確かめる」ひと手間です。

ソフトウェアのテストも同じで、作った後そのまま渡すのではなく、「ちゃんと動くか・困らないか」を先に確かめる作業です。

味見を飛ばした料理がお客さんに出てしまう、それが「テストをしない」ということです。

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・ 最近「使いやすい／使いにくい」と感じたアプリを1つ思い出し、それが品質のどの面（速さ・わかりやすさなど）かを言ってみる。

25-9 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ 「品質」とは結局、誰にとっての何か。
- ・ 「味見」のたとえで、品質はどこで作られると言える？
- ・ 不具合は、いつ見つけると安いか。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、次の章やこれからの仕事の中で、少しずつ埋めていきましょう。

第26章 セキュリティの基礎（41日目）

この章のゴール

この章では、セキュリティは「作る全員の責任」であること、そして守りの基本の考え方を学びます。

終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ① **【考え方がわかる】** セキュリティは一部の担当でなく「作る全員」の責任、とわかる。
- ② **【つながりがわかる】** 「できる（技術的に）」と「やってよい」は違う、がセキュリティでも効く、と言える。
- ③ **【大切さがわかる】** 事故に気づいたら隠さず・すぐ相談、が大切とわかる。

まず自己診断—読む前に、いまの自分を知ろう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「セキュリティは誰の仕事かを自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
- ② 「システムの安全を守るのは、専門の担当者だけの仕事だと思いますか？」を、いまの考えで一言だけ書きます（正解は気にしません）。

読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

ポイント

※攻撃の手順そのものは扱いません。本章は「守る側の考え方」を学ぶ座学です。具体的な防御実装・攻撃技術は、安全な文脈の別科目・OJTで扱います。

26-1 概要—なぜ全エンジニアの責任か

セキュリティは「セキュリティ担当者だけの仕事」ではありません。**作るすべてのエンジニアの責任**です。なぜなら、脆弱性（ぜいじゃくせい＝弱点）は、設計やコードの一行から生まれるからです。どれだけ優秀な専門家が後から守りを固めても、土台に穴があれば守りきれません。

ここで鍵になる考え方が、**セキュリティ・バイ・デザイン**、「後付けで守る」のではなく、**最初から（設計の段階から）安全を組み込む**という発想です。第25章で「品質はプロセス全体で作り込む」と学びましたが、セキュリティもまったく同じで、**安全も後から足すのではなく作り込むもの**です。リリース直前に「セキュリティチェック」を一回やれば済む、という性質のものではありません。

新卒に最初に持ってほしい感覚は、「**自分が書くものは、誰かに狙われうる**」という健全な警戒心です。インターネットにつながるシステムは、世界中から不特定多数にアクセスされます。攻撃の多くは、特定の誰かを狙うというより、機械的に弱点を探し回るものです。だから「こんな小さな機能、無名のシステム、誰も攻撃しないだろう」という油断こそが、事故の入口になります。規模の大小は、狙われない理由にはなりません。

そしてもう一つ、セキュリティが守るのは「システム」だけではない、という点も大切です。その先には、サービスを使う**人々**がいます。個人情報が増えれば、実在する誰かが被害を受けます。セキュリティを守ることは、突き詰めれば**利用者を守ること**であり、第21章の倫理（公衆の利益を最優

先する)と一直線につながっています。本章は、全エンジニアが持つべきセキュリティの基礎を、考え方を中心に学びます。

💡 ポイント

この章で繰り返し立ち返る軸：AIは便利ですが、安全でないコードを生成することもあります（第29章）。

だからこそ、生成物の安全性を評価できる「守りの体系」を持つ人が、AI時代にはいっそう重要になります。

守りの知識がなければ、AIが作った穴に気づけません。

26-2 歴史と背景—性善説から「バイ・デザイン」へ

インターネットは、もともと信頼できる研究者の仲間内で使う前提（いわば性善説）で設計されました。「みんな善意で使う」という前提のうえに、利便性を優先して作られていたのです。その前提を崩した象徴的な事件が、**1988年のモリスワーム**です。これは史上初の大規模なインターネットワーム（自己増殖して広がる不正プログラム）とされ、当時のネットの相当部分を機能停止に追い込み、「**ネットワークは攻撃されうる。セキュリティを本気で考えるべきだ**」という認識を社会に広めました（この事件を契機に、緊急対応を担う組織（CERT：Computer Emergency Response Team）が生まれたとされます）。

その後、Webの普及とともに攻撃も急増しました。**2001年には、Webアプリのセキュリティ向上を目指す非営利団体OWASP（Open Worldwide Application Security Project）が発足**し、代表的な脅威をまとめた「OWASP Top 10」などで広く知られるようになります。近年は、ソフトウェアが多数の外部部品（ライブラリ）の上に成り立つことから「**サプライチェーン（供給網）の安全**」が大きな論点となり、さらにAIの普及で、AIを狙う・AIが生み出す新たなリスクも登場しています。

時代が進むにつれ、攻撃は「愉快犯的なもの」から「**組織的で、金銭や情報を狙う本格的なもの**」へと変質してきました。データを人質に金銭を要求する手口や、広く使われるソフトの弱点を突いて一度に多数を狙う手口など、攻撃は高度化・産業化しています。つまり、セキュリティはもはや一部の専門家の関心事ではなく、**ソフトウェアを作る全員が前提として意識すべき現実**になった、ということです。

歴史の流れは「**性善説 → 攻撃の現実化 → セキュリティ・バイ・デザイン**」と要約できます。ここから学ぶべきは、**利便性と安全はしばしば相反し（トレードオフの関係にあり）、初期に安全を軽視すると後で高くつく**ということです。これは第25章の品質コスト（早く対処するほど安い）と、まったく同じ構造です。脆弱性も、設計段階で気づけば一行の修正で済むものが、本番稼働後に悪用されれば、情報漏えい・サービス停止・信頼失墜という甚大な損失になりえます。

26-3 基礎概念—守りの共通言語

セキュリティを語り、考えるための基本の言葉を押さえましょう。これらは、何かを設計・実装するときの「観点」になります。

■ CIA—情報セキュリティの3要素

情報セキュリティで守るべきものは、3つの頭文字で表されます。

- ・ **機密性（Confidentiality）**、許可された人だけが見られること。（例：他人の個人情報が見えてしまわない）

- ・ **完全性 (Integrity)**、情報が正しく、改ざんされていないこと。（例：残高や記録が勝手に書き換えられない）
- ・ **可用性 (Availability)**、必要なときに使えること。（例：攻撃でサービスが止まらない）

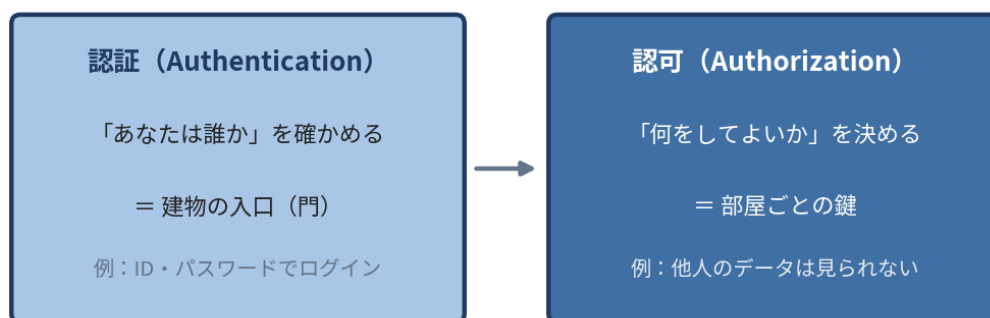
「守る」とは、この3つを守ることです。何かを設計するとき、「この機能で、3要素のどれが、どう脅かされうるか」を考えるのが出発点になります。3つは時に綱引きの関係にもなります（厳重にしすぎると使いにくくなる＝可用性とのバランス）。

■ 認証と認可—似て非なるもの

- ・ **認証 (Authentication)**、「**あなたは誰か**」を確かめること（ログインなど）
- ・ **認可 (Authorization)**、「**あなたに何の権限があるか**」を決めること（この人は閲覧だけ、管理者は編集も可、など）

この2つの混同は、典型的な事故のもとです。「ログインできた（認証された）」ことと「何でもしてよい（認可された）」ことは、まったく別です。たとえば、ログイン後に他人のIDを指定すると他人のデータが見えてしまう、これは「認証は通っているが、認可のチェックが抜けている」典型的な弱点です。

認証と認可は別物 — 「門は通ったが、鍵は別」



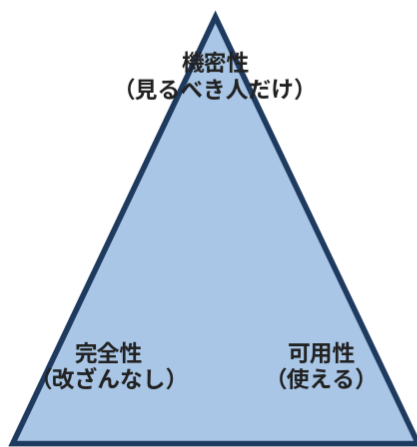
新卒の事故で多い：認証は通っているのに、認可のチェックが抜けて他人のデータが見えてしまう

図 認証と認可の違い

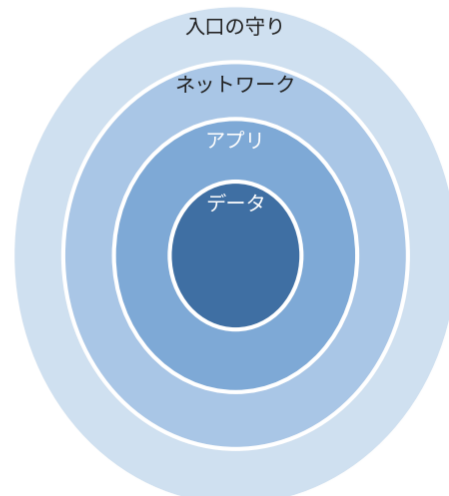
■ 守りの基本原則

- ・ **最小権限の原則**、人にもプログラムにも、必要最小限の権限だけを与える。「念のため全部許可」をしない。万一そこが破られても、被害を最小に抑えられます。
- ・ **多層防御**、一つの守りが破られても、次の守りで止める。守りは一枚の壁でなく、何重にも。

情報を守る考え方（CIAと多層防御）



CIA：3つは時に綱引き（トレードオフ）



多層防御：一枚破られても次で止める

図 情報を守る考え方（CIAと多層防御）

- ・ **脅威モデリング（超入門）**、作る前に「**誰が・何を狙い・どう攻めうるか**」を想像し、弱点に先回りして備える考え方。「攻撃者の目で自分の設計を眺める」習慣です。
- ・ **「信頼せず、検証する」**、外部から来るもの（入力・データ・通信）を無条件に信じない。これは次節の脅威対策に直結します。

脅威モデリングは難しそうに聞こえますが、入口は単純な自問です。たとえば「会員がメッセージを送れる」機能を作るとき、こう問います。「**悪意のある人なら、ここで何をしようとするか?**」。他人になりすまして送れないか（認証）、他人宛のメッセージを読めないか（認可）、おかしい入力ですべてシステムを誤動作させられないか（インジェクション）、大量送信でサービスを止められないか（可用性）。完璧な分析でなくてよいのです。「**攻撃者の視点で一度立ち止まって考える**」その習慣**自体**が、多くの初歩的な穴を塞ぎます。新卒のうちから、機能を作るたびにこの自問を挟むクセをつけましょう。

26-4 主要な脅威を知る—OWASP Top 10

代表的なWebの脅威は、OWASPが「**OWASP Top 10**」として整理・公開しています（数年ごとに更新されます。本書記載時点では2025年版が最新ですが、公開・反映の時期には留意し、参照時は最新版を確認してください。順位や項目は版により変わります）。新卒の目標は、全項目の暗記ではなく、「**こういう種類の弱点があり、なぜ危ないのか**」を理解し、「疑ってかかる」習慣を持つことです。代表的な考え方を挙げます。

- ・ **アクセス制御の不備**、本来アクセスできないはずのデータや操作に、手が届いてしまう弱点（認可の失敗）。近年も最重要級の脅威とされます。「権限のチェックを、すべての入口で漏れなく行えているか」が鍵です。
- ・ **設定ミス（セキュリティ設定の不備）**、初期設定のまま使う、不要な機能やアカウントを開けたままにする、といった設定の甘さから生じる穴。2025年版では順位が上昇したとされます。高度な攻撃でなく、単純な見落としが入口になる点が怖いところです。
- ・ **インジェクション**、外部からの入力に紛れ込ませた不正な命令を、システムが誤って実行してしまう類の弱点（代表例にデータベースを狙うものや、利用者の画面で悪用されるものがあり

ます)。守りの基本は、「外部からの入力を、そのまま信用せず、無害化して扱う」ことです。

- ・ **ソフトウェアサプライチェーンの問題**、自分で書いていない外部部品（ライブラリ等）に由来する弱点（26-5）。2025年版で新しいカテゴリとして加わったとされ、重要性が増しています。

これらに共通する教訓は、驚くほどシンプルです。「**入力を信用しない**」「**初期設定を見直す**」「**権限を絞る**」「**使っている部品を把握する**」。派手な攻撃技術より、こうした基本の徹底こそが、現実の多くの事故を防ぎます。

■ 守りは「ゼロか百か」ではない—リスクに基づいて考える

ここで一つ、現実的な考え方を補っておきます。**セキュリティに「100%安全」はありません**。あらゆる脅威に最高の対策を施すのは、コストの面でも使いやすさの面でも不可能です。だから実務では、「**何を・どれだけ守るか**」を、リスクに基づいて判断します。

判断のものさしは、おおむね「**起きる可能性の高さ × 起きたときの影響の大きさ**」です。たとえば、誰でも見てよい公開情報の保護より、個人情報や決済情報の保護のほうに、重点を置くべきです。すべてを一律に守ろうとして力尽きるのではなく、**守るべき重要なものを見極め、そこに資源を集中する**、この優先順位の発想は、第23章のリスク管理や第25章の品質コストとも通じます。新卒のうちは細かな判断はできなくてよいですが、「守りには優先順位がある」「重要な情報ほど厚く守る」という感覚は持っておきましょう。

なお、Webだけでなく、生成AIを使うシステムには固有の新しい脅威もあります（たとえば、AIへの入力に悪意ある指示を紛れ込ませて意図しない動作を引き出す、といった類のもの）。これらもOWASPがLLM向けの整理を公開するなど、体系化が進んでいます。詳細は第29章に譲りますが、「新しい技術には新しい脅威が伴う」と知っておきましょう。

ここで持ち帰ってほしいのは、**脅威の種類と「自分の作るものにも同じ弱点がないか」と問う習慣**です。攻撃の詳細を学ぶのは、それを防ぐためであり、安全な文脈（別科目）で行います。「攻撃を知る」と「攻撃を行う」ことはまったく別であり、後者は第28章で見る法律にも触れる行為だという点も、忘れないでください。

26-5 依存・サプライチェーンの基礎

現代のソフトウェアは、自分で書くコードよりも、**外部のライブラリ（部品）**のほうが多いことすらあります。便利な反面、その部品に弱点があれば、それは**そのままあなたのシステムの弱点**になります。これが**サプライチェーンのリスク**です。自分の家の鍵をいくら厳重にしても、使っている宅配ボックスに欠陥があれば、そこから侵入されうる、そんなイメージです。

- ・ **ライブラリの脆弱性**、広く使われている部品に、後から弱点が見つかることがあります。見つかった弱点は公表され、攻撃者も知るところとなります。だから、**使っている部品を古いまま放置せず、更新する**ことが基本の守りになります。
- ・ **SBOM（エスボム／Software Bill of Materials）**、「ソフトウェア部品表」。自分のソフトウェアが、どの部品を・どのバージョンで使っているかを一覧にしたものです。食品の「原材料表示」にあたります。これがあると、ある部品に弱点が公表されたとき、「自分のシステムは影響を受けるか」を素早く確認できます。

新卒の実務的な要点は、「**自分が何を使っているかを把握する**」こと。当たり前のようですが、把握できていなければ、いざ弱点が公表されても、自分が危ないかどうかすら分かりません。「把握する」がサプライチェーン対策の第一歩です（※AIが生成したコードの安全性は、別の論点として第29章で扱います。AIは、実在しない部品や古く危険な部品を勧めることがあるからです）。

26-6 日常のセキュリティ習慣

高度な対策の前に、日々の習慣が守りの土台です。これは新卒が今日から実践できることであり、実は事故の多くは高度な攻撃ではなく、こうした基本の隙から起きます。

- ・ **パスワード**、使い回さない。長く・推測されにくいものにする。サービスごとに異なるパスワードにし、パスワード管理ツールを活用するのも有効です。なぜ使い回しが危険かというと、一つのサービスから漏れたパスワードが、他のサービスへの侵入に使い回されるからです。
- ・ **多要素認証（MFA）**、パスワードに加え、もう一つの要素（スマホへの確認など）で守るしくみ。パスワードが漏れても、もう一段の守りで防げます。可能な限り有効化しましょう。
- ・ **フィッシング**、本物そっくりを装ったメールやサイトで、情報を盗もうとする手口。「**急かす（今すぐ・期限切れ）／うますぎる話／不審なリンクや添付**」は疑うサインです。少しでも怪しければ、リンクを踏む前に、送信元を確認し、必要なら周囲や担当に相談を。
- ・ **端末・私物クラウドの扱い**、会社や顧客のデータを、私物の端末や個人のクラウドサービスに不用意に置かないこと。紛失・誤共有・退職時の持ち出しといった、漏えいの典型的な入口になります。
- ・ **物理的な隙とのぞき見**、画面を開いたまま離席しない（画面ロックの習慣）、公共の場で機密を含む画面を開かない・大声で話さない、といった物理的な注意も立派なセキュリティです。高度なハッキングより、「肩越しにのぞかれた」「離席中のPCを操作された」といった単純な経路で情報が漏れることもあります。
- ・ **ソーシャルエンジニアリング**、技術ではなく「人」の隙を突く手口にも注意が必要です。たとえば、関係者を装った電話やメールで情報を聞き出す、緊急を装って手順を飛ばさせる、といったものです。フィッシングもこの一種です。「**正規の手順や本人確認を、雰囲気や急かしで飛ばさせようとするもの**」を疑う感覚を持ちましょう。

こうした習慣は、第21章の「できる≠やってよい」「誠実さ」とも地続きです。「便利だから」と安易にルールを外れない、その小さな判断の積み重ねが、自分と顧客を守ります。新卒のうちは「これは大丈夫だろうか」と迷う場面が多いはずですが、**迷ったら自己判断せず確認する**のが正解です。なお、こうした個人情報の保護は、習慣であると同時に法律上の要請（個人情報保護法など。第28章）とも重なります。「ルールだから守る」のではなく、「なぜ守るのか（誰の何を守っているのか）」を理解すると、判断がぶれなくなります。

26-7 インシデント対応の初動

どれだけ備えても、事故（インシデント）は起こりえます。「絶対に事故を起こさない」ことより、「**起きたときに正しく動けること**」が、現実には重要です。新卒が覚えておくべきは、難しい技術ではなく、**正しい順番と心構え**です。

1. **検知・気づく**、「いつもと違う」「おかしい」と感じる感度を持つこと。違和感を見過ごさない。
2. **報告する**、隠さず、ためらわず、すぐに上長・関係者へ伝えること。**第一報は速さが命**で、完璧な情報を待つ必要はありません。「何が起きたか分からないが、こういう異常がある」で十分です。
3. **記録する**、いつ・何が起きたか、自分がどう操作したかを残すこと。これは原因究明と再発防止の貴重な材料になります。記憶は薄れるので、その場で残します。

最もやってはいけないのは、「**自分のミスかもしれない**」と怖くなって、**隠したり、こっそり直そうとしたりすること**です。報告が遅れるほど被害は広がり、対応も後手に回ります。だからこそ、第25章で学んだ**心理的安全性**、「早く言ってくれてありがとう」と言える文化、が、セキュリティでも決定的に効きます。報告した人を責めない組織ほど、事故に強いのです。

もう一つ、初動を速くするための備えがあります。それは、「何かあったら、誰に・どう連絡するか」を、事故が起きる前に知っておくことです。配属されたら、自社や客先の連絡ルール（報告先、緊急連絡の手順）を早めに確認しておきましょう。いざというとき、「誰に言えばいいのか分からない」が、報告の遅れを生みます。平時の備えが、有事の速さを決めます。

なお、暗号技術・高度な認証技術・ゼロトラストといった専門領域は、本章では扱いません。それらは、守りをさらに深めたい人のための別科目で学びます。本章で持ち帰ってほしいのは、**全エンジニアが持つべき基礎の構え**、「自分の作るものは狙われうる」という警戒心、守りの共通言語、日々の習慣、そして隠さず報告する誠実さ、です。これらは特別な才能ではなく、初日から選べる態度です。

【コラム】客先の本番データ・多重下請けの責任分界

受託・客先常駐では、セキュリティが極めて重い意味を持ちます。

客先の本番データ。受託では、顧客の**本番データ（実在の個人情報や機密を含むことがある）**に触れる場面があります。第21章の「技術的にアクセスできる≠見てよい・使ってよい」が、ここで現実の重みを持ちます。本番データのコピー・持ち出し・目的外利用・不用意な共有は、顧客の信頼と契約を一瞬で壊し、法的責任（第28章）にも直結します。「テストのためにちょっとだけ本番データを使う」も、許可と手順なしには危険です。**扱う前に、許可と取り扱いルールを確認する**、これが鉄則です。やってしまってから「知らなかった」では取り返しがつきません。

多重下請けの責任分界。多層の委託構造（第22章）では、「どこからどこまでが誰の責任か（責任分界点）」が曖昧になりがちです。曖昧なまま進むと、事故が起きたときに対応が遅れ、責任の押し付け合いになりかねません。「自分（自社）が守るべき範囲はどこか」「セキュリティの取り決め（誰が何を守り、何かあったらどう連絡するか）はどうなっているか」を、**作業を始める前に確認する**ことが、結果的に自分とチームを守ります。

💡 ポイント

セキュリティは「面倒な制約」ではなく、顧客の信頼を預かる受託エンジニアの土台です。

守りを作り込める人は、信頼されるエンジニアです。

逆に、どれだけ機能を速く作れても、一度の漏えいで信頼を失えば、それまでの価値は吹き飛んでしまいます。

26-8 考えてみよう（グループ討議）

教室内では手を動かす実技は行いません。3～4人のグループで、考えを言葉にしてみましょう。

✦ 考えてみよう：討議1：身近な「ヒヤリ」の、何が危ないか

日常で経験した・見聞きした「ヒヤリとしたこと」（怪しいメール、パスワードの使い回し、USBメモリの紛失、公共Wi-Fiでの作業、SNSへのうっかり投稿など）を一つ持ち寄る。

それは**CIA（機密性・完全性・可用性）のどれを、どう脅かすか**を分析し、「どうすれば防げたか」を話し合ってください。脅威を「自分ごと」として捉え、抽象的なルールを具体的な行動に結びつけるのが狙いです。

✦ 考えてみよう：討議2：「ちょっとだけ」をどう断るか

「本番データを、テストのためにちょっとコピーしていい？」と現場で言われた、という場面を想像す

る。

26-7・コラム・第21章の倫理を踏まえ、「**なぜ危険か**」「**どう対応するのが正しいか**（誰に確認するか、どんな代替手段があるか）」をグループで考えてください。

「断りにくさ」も含めて、現実的にどう振る舞うかを話し合しましょう。

新人だからこそ言える「確認させてください」の一言の大切さも考えてみてください。

◆ 考えてみよう：討議3：もし自分のミスで事故が起きたら

「自分の操作が原因で、何か異常が起きたかもしれない」と気づいた場面を想像する。

26-7を踏まえ、「すぐ報告する」と「様子を見て自分で直そうとする」で、その後どう違ってくるかを話し合ってください。

報告をためらわせるものは何か、それを乗り越えるには組織と個人に何が必要かも考えてみましょう。

📦 事例から学ぶ：家の戸締まりと同じ

家では鍵をかけ、合鍵を誰にでも渡さず、知らない人が来てもむやみにドアを開けません。

情報セキュリティもこれと同じで、パスワードは「鍵」、使い回しは「同じ鍵を家中・会社中で使う」こと、あやしいメールのリンクは「知らない人にドアを開ける」ことにあたります。

特別な知識より先に、この「戸締まりの感覚」を持てるかどうかが入ります。

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・ 自分のパスワードに使い回しがないか確認し、大事なサービスで2段階の確認（MFA）を1つ有効にしてみる。

26-9 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ セキュリティは、誰の責任か。
- ・ 「できる」と「やってよい」は、セキュリティでもなぜ違うのか。
- ・ 事故に気づいたら、まず何をするか。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、次の章やこれからの仕事の中で、少しずつ埋めていきましょう。

第27章 ドキュメントとプレゼンテーション (41日目)

🎯 この章のゴール

この章では、「書く・伝える」も大事な技術であること、そしてわかりやすく書くコツを学びます。終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ①【理由がわかる】「書く・伝える」も大事な技術、の理由がわかる。
- ②【思い出せる】わかりやすい文章のコツ（結論を先に・短く・一つずつ）を思い出して言える。
- ③【役立ちがわかる】図や議事録が後で自分とチームを助ける、とわかる。

💡 まず自己診断—読む前に、いまの自分を知ろう (2分)

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「伝える文章のコツを自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
- ② 「「わかりやすい文章」とは、どんな文章だと思いますか？」を、いまの考えで一言だけ書きます（正解は気にしません）。

読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

27-1 概要—なぜ「書く・伝える」が技術なのか

「エンジニアの仕事はコードを書くことだ。文章やプレゼンは苦手でもいい」、もしそう思っているなら、本章はその考えを大きく変える章になります。**書く・伝える力は、エンジニアの中核スキル**です。むしろ、技術が高度になるほど、それを他者に伝える力が成果を左右します。

■ 伝わらないことのコスト

ソフトウェアづくりは、第23章で見たとおり**チームで・長期間**行う営みです。あなたの考え（設計の意図、見つけた問題、提案）は、伝わらなければ存在しないのと同じです。伝わらないことは、はっきりとコストがあります。

- ・ 仕様が伝わらず、誤ったものが作られる（手戻り）
- ・ 設計の意図が残らず、半年後に誰も直せなくなる
- ・ 障害の状況がうまく報告されず、対応が遅れる
- ・ 良い提案が、伝え方のせいで通らない

これらはすべて、時間・お金・信頼の損失です。「良いものを作ったのに評価されない」の多くは、技術力ではなく**伝える力**の問題です。

たとえば、ある機能の設計意図を一行も残さずにコードだけ書いたとします。半年後、別のメンバー（あるいは記憶の薄れた自分）がそこを直そうとして、「なぜこう作ったのか」が分からず、調査に半日かかる。あるいは「触ると何か壊れそう」と恐れて誰も手を付けられなくなる。たった数行のメモがあれば防げたはずの損失が、繰り返し発生します。**書く・伝えるは、未来の時間を節約する投資**なのです。

■ 文書は「成果物」であり「検収対象」である

とくに受託開発では、ドキュメントは「おまけ」ではありません。要件定義書・設計書・テスト仕様書・マニュアルといった文書は、それ自体が**契約上の成果物**であり、**検収の対象**です（第23章・

第28章)。「動くものさえあればいい」ではなく、**約束した文書を、約束した品質で納める**こともエンジニアの仕事なのです。文書の不備で検収が通らない、ということも現実に起こります。

新卒が陥りがちな誤解に、「書くのは作業の後の、面倒な事後処理」というものがあります。しかし実際は逆で、**書くこと自体が考えることであり、設計や合意の一部**です。要件を文章にしようとして初めて、曖昧だった点に気づく。設計を図にしようとして初めて、矛盾が見える。つまり、書く力はそのまま「考える力・気づく力」でもあるのです。「コードは書けるが、説明や文書は苦手」という状態は、実は「自分のしていることを、まだ十分には理解できていない」サインかもしれません。書いて伝えられて初めて、その仕事は本当に自分のものになります。

💡 ポイント

本章で繰り返し立ち返る軸：AIは文章作成を強力に助けます。

しかし「何を・誰に・なぜ伝えるか」を決めるのは人間です。

読み手を想定し、要点を見極め、AIの下書きを評価して整える、その判断力こそが、伝える技術の核心であり、AI時代にますます効きます。

27-2 歴史と背景—テクニカルライティングとドキュメント文化の発展

「分かりやすく正確に書く」技術は、**テクニカルライティング**として体系化されてきました。もともとは製品マニュアルや仕様書を、誰が読んでも同じ意味に取れるように書く実務技術として発展したものです。

ソフトウェアの世界では、ドキュメントへの向き合い方も変化してきました。かつては「分厚い文書を最初に固める」文化が主流でしたが、アジャイル(第24章)の広まりとともに「**動くソフトを重視し、ドキュメントは必要十分に**」という考え方が浸透しました。さらに近年は、ドキュメントをコードと同じように管理する「**Docs as Code**」(後述)の発想が広がっています。共通して言えるのは、「**文書のための文書**」をやめ、「**読み手の役に立つ文書**」を目指す方向に進んできた、ということです。

27-3 テクニカルライティングの基礎

技術文書には、小説やエッセイとは異なる「良さ」の基準があります。**面白さや美しさではなく、正確さと分かりやすさ**です。新卒がまず身につけたい原則を挙げます。

■ 明確・簡潔・一貫

- ・ **明確**、一通りの意味にしか取れないように書く。「適宜」「なるべく早く」のような曖昧語を避け、「○日までに」と具体的に。
- ・ **簡潔**、一文を短く。一文に詰め込まない。削れる言葉は削る(この研修資料の方針も同じです。「言葉を削っても同じ意味が伝わるか」が簡潔さの試金石)。
- ・ **一貫**、同じものは同じ言葉で呼ぶ。文体・表記・用語を揃える。

■ 結論を先に(結論先・要点先)

技術文書では、**結論や要点を先に書く**のが鉄則です。読み手(とくに忙しい上司や顧客)は、結論を早く知りたいからです。「背景を長々と説明してから最後に結論」ではなく、「結論 → 理由 → 詳細」の順に。報告・連絡・相談(第1部の報連相)も同じ構造です。

簡単な例で比べてみましょう。悪い書き出しは「先週からA機能の調査を進めており、関連するログを確認し、再現環境も用意して……(中略)……つまり原因はBでした」。良い書き出しは「**結**

論：原因はBです。 対応として〇〇を提案します。以下、調査の経緯です」。後者なら、読み手は最初の一行で要点をつかみ、詳細を読むかどうかを自分で選べます。新卒のうちは前者になりがちなので、「**書き終えたら、結論を一番上に動かせないか**」と見直すクセをつけましょう。

■ 一文一義——一文に一つのことだけ

新卒の文章でとくに多いのが、「一文が長く、複数のことが詰まっている」ケースです。「AはBで、CのときはDになるが、Eの場合はFも考慮して……」と続くと、読み手は迷子になります。**一つの文には一つのことだけ**を書き、文を短く切る。これだけで、ぐっと読みやすくなります。

■ 事実と意見・推測を分ける

技術文書では、**事実（観測できたこと）と、意見・推測（自分の考え）を、はっきり分けて書く**ことが重要です。とくに障害報告では致命的で、「サーバが落ちた（事実）」と「たぶんメモリ不足だと思う（推測）」が混ざると、読み手は判断を誤ります。「～を確認した」「～と考えられる」「～と推測する」と語尾を書き分けるだけでも、文章の信頼性は大きく上がります（第21章の「誠実さ」「事実を重んじる」の実践です）。

■ 用語を統一する

同じ対象を、ある場所では「利用者」、別の場所では「ユーザー」「お客様」と呼ぶと、読み手は「違うものか?」と混乱します。**用語集を持ち、表記を統一する**ことは、地味ですが品質に直結します（第25章の「一貫性」とも通じます）。

■ 箇条書きと表を使い分ける

並列する項目は、ただらと文章にせず**箇条書き**に。条件と結果の組み合わせは**表**に。形式を選ぶこと自体が「分かりやすさ」の設計です。ただし、箇条書きは「項目の関係（順序・因果）」が表せないで、流れを説明したいときは文章のほうが向きます。**中身に合った形式を選ぶ**のも、書く技術の一部です。

■ 読み手を分析する

最も大切なのが、「**誰が読むのか**」を意識することです。同じ内容でも、相手によって書き方は変わります。

- ・ 相手は**技術者か、非技術者（顧客の業務担当者など）か**
- ・ 相手は**前提知識をどこまで持っているか**
- ・ 相手は**何を知りたい・何を判断したいのか**

専門用語を並べれば技術者には伝わりますが、顧客には伝わりません。**読み手の立場に立って言葉を選ぶ**、これが、伝える技術のすべての出発点です。

■ 書いたら、読み返す（推敲）

最後に、地味ですが効く習慣を一つ。**書いたら、必ず読み返す**ことです。書いた直後の文章には、必ず削れる言葉・曖昧な箇所・二重の意味があります。「自分が読み手だったら、これで一度で分かるか」と問いながら読み返すと、多くの不備は自分で直せます。可能なら、**声に出して読む**と、一文の長すぎや不自然さに気づきやすくなります。重要な文書は、提出前に第三者にざっと読んでもらう（第23章のレビューと同じ）のも有効です。「一発で完璧に書ける人」はいません。良い書き手とは、**読み返して直せる人**のことです。

27-4 エンジニアの文書とDocs as Code

エンジニアが実際に書く文書には、さまざまな種類があります。代表的なものを押さえましょう（書き方の詳細・テンプレートは別科目やOJTで）。それぞれ「誰が・何のために読むか」が違うことに注目してください。

- ・ **要件定義書・設計書**、何を・どう作るかを記録し、合意の土台にする（第23章）
- ・ **README**、そのプロジェクトの「入口」。何のためのもので、どう使い、どう動かすかを最初に読む人へ示す
- ・ **API (Application Programming Interface) 仕様**、他のプログラムから使うための「取扱説明書」。入力・出力・約束事を正確に
- ・ **障害報告（インシデントレポート）**、いつ・何が起き・影響は・原因は・どう対応し・再発防止は、を事実に基づいて記録（第23章の障害対応、第25章の「人を責めず仕組みを直す」と接続）
- ・ **議事録**、会議で「何が決まり・誰が・いつまでに何をするか」を残す（決定事項とToDoが命。発言の逐語録ではない）

これらに優劣はありません。共通するのは、いずれも「**後から読む誰か**」のために書く、という点です。READMEは初めてプロジェクトに触れる人のために、API仕様はその機能を使う別のエンジニアのために、障害報告は原因を追い再発を防ぐ人のために書きます。だから、書く前に「**これは誰が、どんな状況で読むのか**」を一瞬考えるだけで、何を書き・何を省くかが定まります。たとえば障害報告で「誰が悪かったか」を書く必要はありません。必要なのは「何が起き、なぜ起き、どう防ぐか」です（第25章の「人を責めず、仕組みを直す」）。文書の良し悪しは、書き手の努力量ではなく、**読み手の役に立ったか**で決まります。

■ Docs as Code—文書をコードのように扱う

近年広がっているのが、**Docs as Code（ドキュメント・アズ・コード）**という考え方です。文書を、コードと同じツール・作法で管理します。

- ・ 文書を**Markdown**などのテキストで書く
- ・ **バージョン管理（Git）**で履歴を残す（23-7）
- ・ **レビュー**を通してから反映する（23-7、本章27-7）
- ・ 必要なら**自動で公開・更新**する（第24章のCI/CDの考え方）

こうすると、文書がコードと一緒に育ち、「コードは新しいのに文書が古い」という、よくある問題を防ぎやすくなります。文書を「書いて終わりの紙」ではなく「**コードと共に保守する生きた成果物**」として扱う、この発想を知っておきましょう。

💡 ポイント

文書には「鮮度」があります。古くて間違った文書は、無いよりたちが悪いこともあります。だからこそ、更新され続ける仕組み（Docs as Code）が効くのです。

27-5 図で伝える

言葉だけでは伝わりにくいことも、一枚の図にすると一瞬で伝わります。「**百聞は一見にしかず**」は、技術の世界でこそ当てはまります。エンジニアがよく使う図を挙げます（描き方の詳細は別科目・第23章のUMLも参照）。

- ・ **構成図（システム構成図）**、どんな要素（サーバ・DB・外部サービスなど）が、どうつながっているかを示す。全体像をつかむ入口

- ・ **フロー図（フローチャート）**、処理や業務の流れを、順序と分岐で示す
- ・ **シーケンス図**、登場人物（画面・サーバ・DBなど）の間で、時間に沿ってどうやり取りされるかを示す（23-6）
- ・ **ER図**、データ同士の関係を示す（23-5）

図を描くときの心構えは、文章と同じです。**目的と読み手に合わせ、伝えたいこと以外を削る**こと。あれもこれも盛り込んだ図は、かえって何も伝えません。「この図で、相手に何を分かってほしいのか」を一つに絞ると、図は格段に伝わるようになります。きれいに描くこと自体が目的ではなく、**理解を助けることが目的**です。

新卒に覚えておいてほしいコツが2つあります。一つは、**抽象度をそろえる**こと。一枚の図の中で、大きな構成要素と細かい部品を混ぜて描くと、見る人は混乱します。「全体像を見せる図」と「特定部分を詳しく見せる図」は、分けて描きます。もう一つは、**言葉と図を補い合わせる**こと。図だけ・文章だけで完結させようとせず、図で全体の関係を示し、文章で要点と意図を補う。両者がかみ合ったとき、理解はもっとも速くなります。なお、図にも「鮮度」があります。実態が変わったのに古い図が残っていると、かえって誤解を生むので、設計が変わったら図も直す意識を持ちましょう。

27-6 プレゼンテーション

書く技術と並ぶのが、**話して伝える技術**、プレゼンテーションです。設計レビュー、進捗報告、提案、勉強会など、エンジニアが人前で説明する場面は想像以上に多くあります。

■ 構成—伝わる順番に組み立てる

プレゼンも、文書と同じく**構成が9割**です。基本は「**結論・要点を先に**」。聞き手が知りたいことから話します。話の見取り図（今日は何を・どんな順で話すか）を最初に示すと、聞き手は安心して付いてこられます。

「相手は誰で、何を知りたいのか、聞いた後にどうなってほしいのか（理解？承認？行動？）」、このゴールを先に決めると、話す内容と順番が定まります。たとえば「予算の承認がほしい」プレゼンと「技術内容を理解してほしい」勉強会では、同じ題材でも構成がまったく変わります。前者は結論（承認してほしいこと）と判断材料を前に、後者は理解を積み上げる順序を重視します。「自分が話したいこと」ではなく「**相手に、どうなってほしいか**」から逆算する、これがプレゼン構成の出发点です。

■ スライド—1枚1メッセージ

- ・ **1枚のスライドで言いたいことは1つ**に絞る
- ・ 文字を詰め込みすぎない（スライドは話の支えであって、台本ではない）
- ・ 図やグラフで視覚的に（27-5）

スライドは「読ませる」のではなく「見せる」もの。聞き手がスライドの長文を読み始めると、あなたの話は耳に入らなくなります。詳しい情報は別途配る資料に回し、スライドは要点だけにとどめます。

■ デモ—動くものは雄弁

ソフトウェアのプレゼンでは、**実際に動かして見せるデモ**が強力です。百の説明より、動く一画面が伝えます。ただしデモには事故がつきもの（うまく動かない、通信が切れる等）。**事前に必ず試し、失敗したときの代替（録画や画面）も用意する**のがプロの備えです。

■ リハーサルと時間管理

本番前に、**声に出して通して練習する**ことを軽視しないでください。頭の中で完璧でも、口に出すと詰まる箇所が必ずあります。また、与えられた時間を守るのもプロの礼儀です。「時間が足りず大事な結論まで行けなかった」を防ぐため、最初に結論を置き（前述）、時間配分を決めて臨みます。緊張するのは当たり前、準備の量だけが、緊張を自信に変えてくれます。

■ 質疑—分らないことは正直に

質疑応答で大切なのは、**誠実さ**です。分からない質問に、その場しのぎの当てずっぽうで答えてはいけません（第21章「できないことをできると言わない」）。「確認して後ほど回答します」と正直に言える人のほうが、信頼されます。

27-7 レビューと非同期コミュニケーション

現代の開発は、全員が同じ場所・同じ時間にいるとは限りません（リモート・分散、第22章）。だからこそ、**文字で・非同期に（その場にいらなくても）伝える**力が重要になります。

- ・ **PR（プルリクエスト）の説明**、「何を・なぜ変えたか」をレビュアーに伝える文章。変更の意図が書かれていないPRは、レビューする側を困らせます。コードだけでなく**説明文も成果物**です（23-7のレビュー文化）。
- ・ **チケット（課題管理）**、タスクや不具合を、後から見ても分かるように記述する。「再現手順・期待する結果・実際の結果」がそろっていると、対応がぐっと速くなります。
- ・ **チャット**、速くて便利ですが、流れて消えます。**重要な決定は、チャットで済ませず、記録に残る場所（ドキュメント・チケット）に転記する**のが鉄則です。

■ 非同期コミュニケーションの作法

その場で聞き返せない非同期では、**最初のメッセージの質**がすべてを決めます。「これ見てください」だけでは相手は動きません。**背景・目的・お願いしたいこと・期限**をひとまとめにして伝えると、無駄な往復が消えます。相手の時間を奪わない書き方は、思いやりであると同時に、チーム全体の速度を上げる技術です。

たとえば「バグ直しました、確認お願いします」だけでは、相手は何をどう確認すればよいか分かりません。「**〇〇の不具合（チケット番号）を修正しました。原因はAで、Bのように直しました。△△の画面で再現しないことをご確認ください。今週中にレビューいただけると助かります**」、ここまで書いて、初めて相手は迷わず動けます。

もう一つ、非同期では「**相手はすぐには反応できない**」前提で動くことも大切です。返事が来ないことを「無視された」と捉えず、急ぎなら別の手段（同期的な相談）に切り替える、相手のタイムゾーンや勤務時間を尊重する、といった配慮も、分散時代の働き方の作法です。文字のコミュニケーションは記録が残る強みがある一方、表情や声色が伝わらないぶん、**冷たく・きつく受け取られやすい**点にも注意します。丁寧さと感謝の一言が、文字のやり取りでは効きます。

【コラム】客先フォーマット・議事録が身を守る

受託開発では、ドキュメントが「身を守る盾」になる場面が数多くあります。

客先フォーマットに従う。受託・客先常駐では、ドキュメントの様式（テンプレート・記載項目・粒度）が**顧客に指定される**ことがよくあります。自社の慣れた書式と違って、まずは**客先のフォーマットに正確に従う**のがプロです（第24章コラムの「客先ルールに従う」と同じ精神）。様式を守ることは、読み手（顧客）への配慮であり、検収をスムーズにする備えでもあります。

議事録が身を守る。とりわけ強力なのが**議事録**です。打ち合わせで「言った/言わない」が後で問題になったとき、その場で取り、共有し、相手の確認を得た議事録が、あなたとチームを守ります。

要点は3つ、**決定事項／宿題（誰が・いつまでに）／保留事項**を明確にすること。そして会議後すぐに共有し、「認識違いがあればご指摘ください」と一言添えること。これだけで、後の多くのトラブルを未然に防げます。仕様変更の経緯も、議事録に残っていれば追加費用の交渉の根拠になります（第23章「仕様の握り」、第28章の契約と接続）。

💡 ポイント

「書くのが面倒」で省いた一手間が、後で何十倍ものトラブルになって返ってくる、これは第25章の品質コストとまったく同じ構造です。文書は、未来の自分とチームへの投資です。

27-8 考えてみよう（グループ討議）

教室内では手を動かす実技は行いません。3～4人のグループで、考えを言葉にしてみましょう。

✦ 考えてみよう：討議1：悪い文書・悪い説明を直す

「分かりにくい」と感じた説明（取扱説明書、学内の通知、アプリの案内、過去に自分が書いた文章など）を一つ持ち寄る。

それが**なぜ分かりにくいのか**を、27-3の原則（明確・簡潔・一貫・結論先・読み手分析）で診断し、**どう直せば伝わるか**をグループで具体的に書き換えてみてください。

「なんとなく分かりにくい」を、原則の言葉で説明できるようになるのが狙いです。

✦ 考えてみよう：討議2：1分で要点を伝える

各自、最近取り組んだこと（研修の学び、趣味、アルバイト等）を一つ選ぶ。

それを「**結論先・1分**」で説明する練習をします。一人ずつ話し、聞き手は「結論が先にあったか」

「要点が絞れていたか」をフィードバック。短く・要点から伝えることの難しさと効果を体感してください。

✦ 考えてみよう：討議3：この決定、どこに残す？

あるチャットのやり取りで「じゃあ仕様はAで確定しましょう」と決まった、という場面を想像する。

この決定を、このままチャットに流すとどうなるか。**どこに・どう記録すれば**後で困らないかを話し合ってください。27-7「重要な決定は記録に残す」を、具体的な行動に落とすのが狙いです。

📖 事例から学ぶ：引き継ぎノートにたとえると

アルバイトを辞めるとき、次の人のために手順をメモした経験はありませんか。

よいメモは、その場にいない人が読んでも同じように動けます。

逆に「自分だけが分かるメモ」は、後から来た人を困らせます。

仕事のドキュメントもまったく同じで、「自分のためのメモ」ではなく「後から読む人のための説明」になっているかが、良し悪しの分かれ目です。

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・ 短い連絡文を1つ書き、「結論が先頭にあるか」「一文が長すぎないか」を見直してみる。

27-9 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ 「書く・伝える」は、なぜ大事な技術なのか。
- ・ わかりやすい文章のコツを、思い出せるだけ。
- ・ 議事録は、後で何の役に立つか。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、次の章やこれからの仕事の中で、少しずつ埋めていきましょう。

第28章 最低限知っておくべき法律（42日目）

この章のゴール

この章では、法律を学ぶ理由（自分と仲間を守る・迷ったら相談する）、そして身近な法律の考え方を学びます。

終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ①【**目的がわかる**】法律を学ぶのは「自分と仲間を守る」「迷ったら相談する」ため、とわかる。
- ②【**知っている**】会社で書いたものの権利は原則会社のもの、と知っている。
- ③【**構えがわかる**】「おかしいかも」と感じたら一人で抱えず相談、という構えがわかる。

まず自己診断—読む前に、いまの自分を知ろう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「エンジニアと法律の関係を自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
 - ② 「法律は「破らなければ問題ない」と思いますか？」を、いまの考えで一言だけ書きます（正解は気にしません）。
- 読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

知っておこう

※本章は法律の**全体像**を示すもので、法的助言ではありません。

記載内容は2026年時点のものであり、法令は改正されます。

実際の判断は、必ず最新の公式情報・社内の法務・専門家に確認してください。

とくに**個人情報保護法の見直し・AI規制・取適法**など動きの速い領域は、各節でも「記載時点」を断っています。

28-1 概要—なぜ法を知るか／法の全体像

エンジニアがなぜ法律を学ぶのでしょうか。弁護士になるためではありません。理由は、第21章で学んだ一言に集約されます。「**できる**」と「**やってよい**」は違う。技術的に可能なことのすべてが、法的・倫理的に許されるわけではありません。

法を知る目的は2つです。第一に、**知らずに一線を越えてしまうことを防ぐ（自衛）**こと。「知らなかった」は、多くの場合、言い訳になりません。第二に、**おかしいと気づき、適切に相談できる**ようになることです。新卒に求められるのは、条文を暗記して自分で判断することではありません。

「これは法に関わるかもしれない」という"におい"を嗅ぎ分け、立ち止まって相談すること。これができれば十分です。

■ 法の全体像

本章で扱う法を、ざっくり見渡すと次のようになります。

- ・ **知的財産**、成果物の権利は誰のものか（著作権・OSSライセンスなど）
- ・ **情報・秘密の保護**、不正アクセス禁止・営業秘密・個人情報

- ・ **受託の取引**、契約のかたちと責任（請負/準委任・契約不適合・派遣/取適法）。本章で厚めに扱います
- ・ **働く人の法**、労働基準法など
- ・ **これからの法**、AI規制・輸出管理など、動きの速い領域

このうち、皆さんが**受託（SI）の現場で日々直面するのが「取引」と「働き方」の法**です。本章は、そこを厚めに扱います。

💡 ポイント

強調しておきたいのは、法律を「脅し」として覚えないことです。罰則を恐れて萎縮するのではなく、「**自分と顧客と仲間を守る道具**」として、前向きに知っておきましょう。

28-2 歴史と背景—情報化と法整備の歩み

法律は、社会の変化を追いかけて整備されてきました。コンピュータやインターネットが普及するにつれ、それまでの法では想定していなかった問題、データの不正利用、個人情報の漏えい、ソフトウェアの権利、ネット上の犯罪、が次々と現れ、その都度、新しい法律や改正が積み重ねられてきました。

近年も、データやAIといった新しい技術に合わせて、法は動き続けています。だからこそ本章で重要なのは、個々の条文を覚えることより、「**技術が進むと、それを律する法も変わり続ける**」という**構造を理解し、常に最新を確認する姿勢を持つ**ことです。「数年前に習った法律知識が、今も同じとは限らない」、これは技術の陳腐化（第22章）とまったく同じ構図です。

28-3 知的財産—成果物の権利は誰のものか

ソフトウェアやドキュメントといった「**知的財産**」には、権利があります。新卒が押さえるべき要点を概観します。

■ 著作権と職務著作

プログラムやドキュメントは、**著作権**で保護される著作物になりえます。ここで新卒が誤解しやすいのが、「自分が書いたコードは自分のもの」という感覚です。実際には、**会社の業務として作ったものは、原則として会社に著作権が帰属する**仕組みがあります（いわゆる**職務著作／法人著作**。一定の要件のもとで、著作者が法人となる）。つまり、業務で書いたコードを勝手に持ち出したり、個人で公開したりすることはできません。

■ OSSライセンス

第30章でも扱うOSS（オープンソース）は「自由に使える」イメージがありますが、**それぞれに利用条件（ライセンス）があります**。「無料＝何でも自由」ではありません。ライセンスの種類によって、「改変したら公開が必要」「著作権表示を残す必要がある」など、守るべき条件が異なります。条件を守らずに使うと、ライセンス違反になります。

ごく大づかみに言うと、OSSライセンスには性格の異なる種類があります。一方には、**利用や改変の自由度が高く、著作権表示などを残せば商用利用も比較的ゆるやかに認める**タイプ（よく「寛容型」と呼ばれます）。もう一方には、**そのソフトを組み込んで配布する場合、自分のコードも同じ条件で公開することを求める**タイプ（「コピーレフト型」と呼ばれます）。後者を、条件を知らずに自社製品に組み込んでしまうと、「自社のソースコードも公開せねばならない」といった事態になりかねません。これは受託では顧客との契約とも衝突しうる、現実的なリスクです。

ですから新卒の基本動作はシンプルです。**使う前に、その部品のライセンスを確認する。判断に迷ったら、自分で決めずに先輩や社内ルールに確認する。**「みんな使っているから大丈夫だろう」で進めないことです（詳細な種類・条件は別途学習します）。

■ 生成AIが書いたコードの扱い

生成AIが出力したコードの権利関係や、学習データとの関係は、**新しく、まだ定まりきっていない論点**を含みます（第29章）。「AIが書いたから自由に使える」と単純に考えず、**会社の方針やライセンス、機密の観点を確認する**慎重さが必要です。ここは特に変化が速いので、最新の社内ルールに従ってください。

28-4 情報セキュリティ関連法と営業秘密

第26章のセキュリティは、法律とも結びついています。

■ 不正アクセス禁止法

他人のIDやパスワードを無断で使ったり、アクセス権限のないシステムに侵入したりする行為は、**不正アクセス禁止法**で禁じられています。「技術的にできるか」ではなく「**権限があるか**」が境界です（第26章の認可と同じ発想）。興味本位の"試し"も、権限がなければ違法になりえます。

■ 不正競争防止法と営業秘密

会社や顧客の重要な秘密情報は、**不正競争防止法**の「**営業秘密**」として保護されることがあります。営業秘密として守られるには、一般に3つの要件があるとされます。

- ・ **秘密管理性**、秘密として管理されていること
- ・ **有用性**、事業に役立つ情報であること
- ・ **非公知性**、公に知られていないこと

近年は、ビッグデータの保護を念頭に「**限定提供データ**」という枠組みも設けられ（2018年改正で新設、その後2023年改正で拡充）、データの不正取得・利用への対策が強化されてきました。新卒の実務上の要点はシンプルで、**業務で知った秘密情報を、外に持ち出さない・漏らさないこと**。退職時に持ち出す、SNSにうっかり書く、こうした行為が法的責任につながります（守秘義務は契約=NDAとも結びつきます。28-6）。

28-5 個人情報保護法

システムは、しばしば**個人情報**（氏名・連絡先・購買履歴など、特定の個人を識別できる情報）を扱います。その取り扱いを定めるのが**個人情報保護法**です。

2022年4月施行の改正では、**漏えい等が起きた際の個人情報保護委員会への報告義務・本人への通知義務**が定められ、違反への**罰則も強化**されました（法人に対する重い罰金など）。さらにこの法律は「**3年ごとに見直す**」とされており、2026年時点では、課徴金制度の導入などを含む次の改正に向けた検討が進められています（検討中であり、内容・時期は流動的です。最新は公式情報を確認してください）。

新卒が押さえておきたい実務的なポイントを、もう少し具体的に挙げます。

- ・ 「**個人情報**」は**思うより広い**、氏名そのものだけでなく、他の情報と組み合わせて特定の個人を識別できるもの（メールアドレス、顧客ID+αなど）も該当しえます。「これは個人情報ではないだろう」という自己判断は危険です。
- ・ **目的の範囲で使う**、個人情報は、集めたときに示した利用目的の範囲で使うのが原則です。「ついでに別のことにも使う」は、目的外利用になりえます（第21章のデータ倫理と同じ）。

- ・ **安全に管理する**、漏えい・紛失を防ぐための管理（安全管理措置）が求められます。第26章のセキュリティ習慣が、ここで法的な要請とも重なります。
- ・ **委託先も管理対象**、受託では、顧客から個人情報の取り扱いを委託される立場になることがあります。委託された側にも適切な管理責任が及ぶ、という点は重要です。

新卒が持つべき構えは、「**個人情報**は**"預かりもの"**であり、**目的外に使わない・むやみに持ち出さない・漏らさない**」という基本です。便利だからと安易に扱わない、第26章のデータの扱い、第21章の倫理と一直線です。

28-6 受託の取引法①—契約形態と成果物責任

ここからは、受託（SI）エンジニアが日々直面する**取引の法**を、厚めに扱います。まず**契約のかたち**です。

■ 請負と準委任—何に責任を負うかが違う

受託でよく使われる契約には、大きく2種類あります（第24章でも触れました）。

- ・ **請負契約**、「**仕事の完成**」に責任を負う。約束した成果物を完成させて初めて、対価を受け取れる。完成責任があるぶん、「何を作れば完成か」を明確にする必要がある。
- ・ **準委任契約**、「**業務の遂行（適切に作業すること）**」に責任を負う。成果物の完成自体は約束しない（善管注意義務をもって業務を行う）。要求が変わりうる開発（アジャイル等）と馴染みやすい。

この違いは、責任の重さ・お金の受け取り方・進め方に直結します。たとえば、同じ「3か月の開発」でも、請負なら「3か月後に動く〇〇を完成させること」が約束であり、完成しなければ対価をめぐる問題になります。準委任なら「3か月、適切な技術者として誠実に開発業務を行うこと」が約束で、完成そのものは契約の対象ではありません。どちらが良い・悪いではなく、**作るものの性質（要求が固いか・変わりうるか）に応じて選ぶ**ものです。「自分の現場はどちらの契約か」を知ることは、自分の責任範囲を理解する第一歩です。

■ 契約不適合責任—「仕様に書いてあるか」が鍵

納めた成果物が、契約で約束した内容と違っていた場合の責任を、**契約不適合責任**といいます（2020年4月施行の改正民法で、それまでの「瑕疵担保責任」から変わりました）。発注者（顧客）は、不適合があった場合、**修正（追完）・代金減額・損害賠償・契約解除**などを求めることができます。通知には期間の定めがあります（一般に、不適合を知った時から一定期間内）。

新卒に重要なのは、その**実務的な含意**です。改正前は「隠れた欠陥を知っていたか」といった点が争われがちでしたが、改正後は「**契約・仕様書に何と書いてあるか（契約の内容に適合しているか）**」が判断の中心になりました。つまり、**要件定義書・仕様書がいかに大切か**ということです（第23章・第27章）。「仕様に書いていないこと」をめぐる争いを防ぐには、何を作るかを文書で明確に握っておくしかありません。

具体的に考えてみましょう。「検索が速いこと」とだけ口頭で言われ、文書化しなかったとします。納品後に顧客が「遅い、契約不適合だ」と主張したとき、何をもって「速い」とするかが仕様がないと、水掛け論になります。逆に「検索結果が3秒以内に表示されること」と仕様に明記し合意していれば、適合か不適合かを客観的に判断できます。**非機能要件（第23章）を仕様に落とすこと**が、こうした場面で効くのです。

■ SLA・検収・NDA—現場で出会う言葉

- ・ **SLA（サービス品質保証）**、とくに運用・保守で、「どの程度の品質・可用性を保証するか」を取り決めるもの（例：稼働率、障害対応時間）。非機能要件（第23章）と結びつきます。

- ・ **検収**、顧客が成果物を確認し、契約どおりかを判断する手続き。ここを初めて「納品完了」になります。検収基準は、仕様で明確にしておくべきものです。
- ・ **NDA（秘密保持契約）**、業務で知った秘密を外に漏らさないことを約束する契約。28-4の営業秘密の保護と結びつき、受託では当たり前に関わられます。

なお、契約書などへの**電子署名**は、法律（電子署名及び認証業務に関する法律。2001年施行）により、一定の要件を満たせば紙の押印に相当する効力を持つとされています。電子契約が一般化した今、知っておくとよい知識です。

28-7 受託の取引法②—派遣法と取適法

受託、とくに**客先常駐**で働くとき、避けて通れないのが「**誰の指揮命令で働くか**」の問題です。

■ 偽装請負・二重派遣—なぜ問題か

働き方には、法的に異なる類型があります。ごく大づかみに言うと、

- ・ **請負・準委任**では、あなたは**自社の指揮命令**で働きます（顧客から直接、業務の細かい指示は受けない建前）。
- ・ **労働者派遣**では、あなたは**派遣先（顧客）の指揮命令**で働きます。これには**労働者派遣法**に基づくルールがあります。

問題になるのが、**契約は請負なのに、実態は顧客から直接指揮命令を受けている**ケース。これは「**偽装請負**」と呼ばれ、法的に問題があります。また、派遣された人をさらに別へ派遣する「**二重派遣**」も禁じられています。これらは、働く人の保護や責任の所在を曖昧にするため規制されています。

なぜこれらが規制されるのでしょうか。指揮命令と雇用責任の所在が曖昧になると、働く人が不安定な立場に置かれ、労働時間や安全、トラブル時の責任の所在がうやむやになるからです。つまりこれらの規制も、根っこは**働く人を守る**ためにあります。

新卒が覚えるべきは、難しい線引きそのものではありません。「**契約上の自分の立場と、実際の指示系統が食い違ってないか**」に違和感を持ち、**おかしいと思ったら相談する**、これです。たとえば「請負で来ているのに、客先の人から毎日細かく作業を割り振られ、勤怠まで管理されている」と感じたら、それは相談すべきサインかもしれません（コラムで具体例を扱います）。一人で「これは普通なのかな」と抱え込まないことが大切です。

■ 取適法（旧・下請法）—2026年改正

多重下請け構造（第22章）における、立場の弱い受託側を守る法律があります。従来「**下請代金支払遅延等防止法（下請法）**」と呼ばれていたものが、**2026年1月施行で「中小受託取引適正化法（取適法）」へ改称・拡充**されました。用語も「親事業者→委託事業者」「下請事業者→中小受託事業者」へと改められています。

この改正では、適用範囲が広がり（従来の資本金基準に加え従業員数基準が追加）、「**価格交渉に応じない・一方的に価格を決める**」といった行為が明確に禁止されるなど、立場の弱い側を守る方向が強化されました。新卒が細部を覚える必要はありませんが、「**取引には、弱い立場を守るための法がある**」「**かつての"下請法"は名前が変わった**」ことは知っておきましょう（2026年時点の情報。最新は公式情報を確認）。

28-8 労働法の基礎と、これからの法

■ 労働法の基礎—働く自分を守る

働く人を守る基本が**労働基準法**です。IT現場に関わる代表的な点を挙げます。

- ・ **36（さぶろく）協定**、法定労働時間（1日8時間・週40時間）を超えて働かせるには、労使の協定と届出が必要です。
- ・ **時間外労働の上限規制**、働き方改革に伴い、残業時間に上限が定められました（原則、月45時間・年360時間。特別な事情があっても、月100時間未満・年720時間などの上限があり、違反には罰則があります）。

これらは「会社を縛るルール」であると同時に、「**働くあなた自身を守るルール**」です。第30章の「燃え尽きずに続ける」とも結びつきます。自分の働き方が、これらの枠を大きく超えていないか、気になったら、社内規程を確認し、相談してよいのです。

■ これからの法—AI規制・輸出管理・PL法

技術の最前線では、法も急速に整備されつつあります。

- ・ **AI規制**、日本は2025年に、研究開発と活用を推進する基本法的な法律（AI推進法）を整えました。これは**罰則を伴わないソフトな枠組み**で、自主的なガバナンスを促す性格が強いものです。一方、EUは**リスクに応じて規制し、違反に制裁金を科すハードな法（EU AI Act、2024年成立）**を整えました。**日本＝ソフトロー／EU＝ハードロー**という対比が、2026年時点の大きな構図です（いずれも動きが速く、最新確認が必要）。
- ・ **輸出管理**、高度な技術や製品は、安全保障の観点から、海外への提供に規制がかかることがあります（安全保障貿易管理）。グローバルな開発に関わるなら、頭の片隅に置いておきましょう。
- ・ **製造物責任法（PL法）の注記**、製品の欠陥で損害が生じた場合の責任を定める法律ですが、**ソフトウェア単体は無体物のため、原則としてPL法の対象外**とされています。ただし、機器に**組み込まれたソフト**の不具合が製品の欠陥と見なされる場合など、関わってくる論点もあります（組み込み・IoT＝モノのインターネットでは要注意）。過度に恐れる必要はありませんが、「ソフト単体は原則対象外」という整理を知っておきましょう。

（コラム）受託成果物の権利帰属／検収トラブルと仕様書／客先常駐で指揮命令を受けたら

受託の現場で、法が顔を出す典型例を3つ挙げます。

成果物の権利は誰のもの？ 受託で作ったプログラムの著作権が、発注者（顧客）と受託者（自社）のどちらに帰属するかは、**契約で取り決められます**。「作ったのは自分たちだから自社のもの」とは限りません。流用・再利用したいときに権利関係でつまづかないよう、契約上どうなっているかを意識しましょう（28-3・28-6）。

検収トラブルと仕様書。「これは仕様に入っていたはず」「いや聞いていない」、検収時のもめごとの多くは、仕様の曖昧さから生まれます。28-6で見たとおり、契約不適合の争点は「仕様にと書いてあるか」です。だからこそ、**要件・仕様を文書で握り、変更は記録して合意し直す**（第23章・第27章の議事録）ことが、最大の予防策になります。文書は、あなたとチームを守る盾です。

客先常駐で、顧客から直接指示を受けたら？ 契約が請負・準委任なのに、客先で顧客から細かい業務指示を直接受け続ける、これは偽装請負(28-7)の懸念があります。新卒が一人で判断するのは困難です。「**指示系統がおかしいかも**」と感じたら、**抱え込まず、自社の上長に必ず相談すること**。これは、あなたを守るための正しい行動です。

💡 ポイント

ここでも姿勢は同じです。法を恐れて萎縮するのではなく、「おかしい」に気づき、相談する。**できる≠許される**を忘れず、自衛と相談を習慣にしましょう。

28-9 考えてみよう（グループ討議）

教室内では手を動かす実技は行いません。3～4人のグループで、考えを言葉にしてみましょう。

◆ 考えてみよう：討議1：このケースは何法に関わるか

次のケースについて、「**どの法律・ルールに関わりそうか**」「**どうすべきか（誰に相談するか）**」をグループで考えてください。正確な法解釈を当てる場ではなく、「**法に関わる"に"おいに気づく**」練習です。

1. 前職（前のアルバイト先）で使っていた顧客リストを、今の仕事に使おうとしている同僚がいる
2. 「無料で使えるから」と、ライセンスを確認せずにネットで見つけたコードを製品に組み込もうとしている
3. 客先常駐先で、契約は請負なのに、顧客の担当者から毎日細かく作業指示を受けている
4. テストのために、顧客の本番データ（個人情報を含む）を自分のPCにコピーしようとしている

◆ 考えてみよう：討議2：仕様書は、なぜ自分を守るのか

28-6・コラムを踏まえ、「検収でもめた」想定場面の一つ作る。

仕様書や議事録があった場合と、なかった場合で、何がどう変わるかを話し合ってください。「面倒な書類」が「身を守る盾」になる構造を、自分の言葉で説明できるようになるのが狙いです。

📁 事例から学ぶ：拾った画像、使っていい？

ネットで見つけた写真やイラストを、よさそうだからとSNSや資料にそのまま貼る、身近によくある場面ですが、これは著作権の侵害になりうる行為です。

「ネットにあるもの＝自由に使える」ではありません。

仕事ではさらに、他社の文章・ロゴ・コード・契約に関わるものを扱います。

「使う前に、使ってよい条件かを確認する」習慣が、自分と会社を守ります。

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・ ネットで見つけたコードや画像を使うとき、「使っていい条件か」を一度確認するクセをつける。

28-10 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ 法律を学ぶのは、何のためか。
- ・ 会社で書いたコードの権利は、原則だれのものか。
- ・ 「おかしいかも」と感じたら、どうするか。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、次の章やこれからの仕事の中で、少しずつ埋めていきましょう。

第29章 生成AIによる開発の現在（42日目）

🎯 この章のゴール

この章では、生成AIが「変えるもの」と「変えないもの」、そしてAIとの付き合い方を学びます。終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ①【**性質がわかる**】AIは便利だが「もっともらしく間違える」ことがある、とわかる。
- ②【**区別できる**】AIが変えるのは“やり方”で、“何を・誰が責任をもつか”は人に残る、と言える。
- ③【**考え方がわかる**】「AIが決めたから」は言い訳にならない、とわかる。

💡 まず自己診断—読む前に、いまの自分を知ろう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「AIとの付き合い方を自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
 - ② 「AIの答えは、いつも正しいと思いますか？」を、いまの考えで一言だけ書きます（正解は気にしません）。
- 読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

💡 ポイント

※本章で挙げる製品名・年代・数値は「〇年時点」の例です。この分野は変化が極めて速く、具体的な道具はすぐに移り変わります。だからこそ本章は、**変わりにくい「考え方」**に重心を置きます。

29-1 概要—AIはエンジニアに何をもたらすか

生成AIの登場は、ソフトウェアづくりに大きな変化をもたらしました。コードの下書き、調査、文書作成、テストの生成、かつて時間のかかった多くの作業を、AIが高速に肩代わりしてくれるようになりました。これは脅威でしょうか、福音でしょうか。

本章の答えはシンプルです。**手段は変わる。しかし判断は残る。**

AIが変えたのは「どうやって作るか（手段）」です。コードを一行ずつ手で書く比率は下がるかもしれません。しかし、「**何を作るべきか**」「**その品質でよいか**」「**どこを信じ、どこを疑うか**」「**どこまで任せ、どこで人が責任を持つか**」、こうした**判断**は、むしろAIの登場で重要さを増しました。AIは答えを大量に出しますが、その答えが正しいか・安全か・目的に合うかを決めるのは人間だからです。

そして、その判断の質を支えるのが、本特論で学んできた**ソフトウェアエンジニアリングの体系**です。だから本章の結論を先に言えば、

💡 ポイント

AI時代にこそ、体系が効く。

この章は、AIの仕組みと現在の使われ方を「考え方」として押さえ、最後にこの結論へ戻ってきます。なお、AIツールを実際に操作する演習は別科目で行います。本章は座学として、向き合い方を学

びます。

29-2 歴史と背景—推論から学習、そして生成へ

生成AIは、ある日突然現れたわけではありません。AIという分野には、70年近い積み重ねがあります。

ごく大づかみには、AIは「**推論 → 知識 → 学習**」とパラダイムを変えてきました。1956年のダートマス会議で「人工知能」という分野が生まれ（推論・探索の時代）、専門家の知識をルール化する試み（エキスパートシステム）を経て、データから自ら学ぶ「機械学習・深層学習」が花開きました。転機の一つが**2017年のTransformer**という技術で、これが現在の大規模言語モデル（LLM）の土台になりました。そして**2022年11月のChatGPT公開**を境に、生成AIが一気に社会へ普及し、近年（2025年ごろ）は、人がゴールを与えるとAIが自律的に作業を進める「**エージェント**」型へと発展しています。

この歴史から学ぶべき教訓が2つあります。第一に、AIは**ブームと冬の時代を繰り返してきた**こと。過度な期待と幻滅のサイクルを知れば、今の熱狂も冷静に見られます。第二に、生成AIは**数十年の技術蓄積の上に立つ**ものであり、魔法ではないこと。仕組みに裏打ちされた道具だと理解することが、正しく使う出発点です。

なお、本章で挙げる製品名や時期は、あくまで「〇年時点」の例にすぎません。この分野の具体的なツールや性能は、半年・一年で大きく変わります。だからこそ本章は、最新ツールの使い方を覚えることを目的とせず、**どんなツールが来ても通用する向き合い方**、AIの性質を理解し、判断と責任を人が持ち、検証を欠かさない、を学ぶことに重心を置きます。具体的なツールの習熟は、その時々で最新版で、別科目やOJTで行ってください。陳腐化するのは道具であって、向き合い方ではありません。

29-3 LLMの仕組み超入門—なぜ間違えるのか

AIを使いこなすには、「**なぜ間違えるのか**」を理解しておくことが決定的に重要です。仕組みを知らずに使うと、AIの誤りに気づけません。

■ LLMは「確率的に、次の言葉を予測している」

現在の生成AI（LLM）は、ものすごく大づかみに言えば、**膨大な文章から学んだパターンをもとに、「次に来そうな言葉」を確率的に予測して並べている**仕組みです。意味を「理解」して答えているというより、もっともらしい続きを生成している、というイメージに近いものです（これは説明のための単純化で、実際の仕組みはより複雑です）。

この性質から、重要な帰結が導かれます。**AIは「正しいこと」ではなく「もっともらしいこと」を出す**、だから、堂々と間違えることがあるのです。

AIの答えは、しばしば流暢で自信に満ちていますが、その**流暢さは正しさの保証ではありません**。人間は「自信ありげに、整った言葉で語られると、つい信じてしまう」傾向があります。AIはまさにそこを突いてくる（意図せずとも）ので、「もっともらしいから正しい」という思い込みを、意識的に外す必要があります。とくに、自分がよく知らない領域ほど、AIの誤りを見抜けません。だからこそ、後述する「検証」と「基礎力」が効いてくるのです。

■ ハルシネーション

AIが、事実と異なることを、さももっともらしく生成してしまう現象を**ハルシネーション（幻覚）**と呼びます。存在しない関数やライブラリ、誤った仕様を、自信ありげに提示することがあります。これは「バグ」ではなく、確率的に言葉を生成する仕組みの**性質**です。だから、**AIの出力は常に検証**

が必要なのです。

関連して、セキュリティ上の新しいリスクも生まれています。たとえば、AIが間違えて提案しがちな（実在しない）部品名を、攻撃者が先回りして悪意あるものとして用意しておく、という手口（スロップスクワッティングと呼ばれる）も報告されています。「AIが勧めたから安全」とは限らないのです。

■ RAG—外部の知識で補う

AIの「知らないこと・古いこと・もっともらしい嘘」を補う代表的な工夫が、**RAG（検索拡張生成）**です。これは、AIに答えさせる前に、**信頼できる資料を検索して渡し、その内容に基づいて答えさせる**という考え方です。社内文書や正確な資料を根拠にAIを使う場面で広く使われます。仕組みの詳細より、「**AIの弱点（知識の限界・嘘）を、外部の正しい情報で補う発想がある**」と理解しておきましょう。

■ もう一つの限界—「知識のカットオフ」と文脈の限り

LLMには、ほかにも知っておくべき限界があります。一つは、学習した時点より新しいことは（そのままでは）知らない、という点です。最新の出来事や、ごく最近変わった仕様について、古い情報のまま答えることがあります。もう一つは、一度に扱える情報量に限りがある、という点です。長大な文脈をすべて完璧に踏まえてくれるとは限りません。

こうした限界を知っていれば、使い方も変わります。「最新情報や正確性が重要な場面では、AIの答えを鵜呑みにせず一次情報で確かめる」「重要な前提は、省略せずきちんと与える」、仕組みの理解が、そのまま賢い使い方につながるのです。**道具の限界を知る者が、道具を使いこなす**、これは、どんな技術にも共通する原則です。

29-4 AI開発の進化と新しいスタイル

■ 補完 → チャット → エージェント

AIによる開発支援は、関わり方の「深さ」で段階的に進化してきました（年代は概況）。

1. **コード補完**、行や関数単位で続きを提案する（初期の支援）
2. **チャット型支援**、質問や生成を対話で依頼し、人がコピーして使う
3. **統合・マルチファイル編集**、「こうしたい」と書くと、複数ファイルにまたがる変更をAIが提案し、人がレビューする
4. **エージェント型**、人は**ゴールを与えるだけ**で、AIが自分で計画し、ツールを使い（ファイル操作やコマンド実行）、結果を評価して進める

注意したいのは、進化するほど**人の関わりが「手を動かす」から「方向づけと検証」へ移る**ことです。エージェントが自律的に動くほど、「ゴールを正しく与える力」と「結果を正しく評価する力」が問われます。手放して任せきりにできる、という意味ではありません。

たとえるなら、AIエージェントは「優秀だが、こちらの意図を完全には汲めず、ときどき自信満々に的外れなことをする部下」のようなものです。優秀な部下を活かせるかは、上司（あなた）が「何をしてほしいか」を的確に伝え、上がってきた成果を適切にチェックできるかにかかっています。指示が曖昧なら、優秀な部下でも見当違いの仕事をします。つまり、AIが賢くなるほど、それを使う人間の**「任せ方とチェックの力」**（マネジメントに近い力）が問われるようになる、ということです。これは第23章のマネジメントや、第27章の「伝える力」とも通じます。

■ AI駆動・仕様駆動・バリエーション

AIを使った開発の進め方にも、いくつかの考え方が登場しています（2025年ごろの整理）。

- ・ **AI駆動開発**、AIを開発の中心に据える、広い概念
- ・ **仕様駆動開発（SDD）**、「**仕様（要求・受入基準）を明示してから、AIに実装させる**」進め方。「仕様 → 計画 → タスク分解 → 実装」と段階を踏み、AIの暴走や手戻りを減らす。明示と検証を重ねる規律的なやり方
- ・ **パイプコーディング**、「なんとなくこう動いてほしい」という感覚的な要望で、AIに任せて作らせる手法（2025年ごろに広まったとされる）。素早い試作には向くが、仕様も検証も曖昧になりやすい

ここに本質的な対比があります。**パイプコーディング（感覚・速さ）⇔ 仕様駆動開発（明示・検証）**。試作なら前者でもよいですが、品質と責任が問われる本番では、後者の規律、つまり「**何をやるかを言語化し、結果を検証する**」、が効きます。これは、第23章の「要求」や第25章の「品質」と、まったく同じことを言っているのに気づくでしょう。**新しい道具が出て、効く規律は変わらない**のです。

29-5 AIを使いこなすスキル

AIを「使う」のは簡単ですが、「**使いこなす**」には固有のスキルが要ります。新卒が意識したい5つを挙げます。

- ・ **仕様を書く力**、曖昧な要望ではなく、要求・前提・受入基準を言語化する力。これはSDDの中核であり、第23章・第27章で学んだことそのものです。AIへの指示の質は、要求を言葉にする力で決まります。
- ・ **プロンプト／コンテキスト設計**、AIに、必要な前提・制約・参照情報を過不足なく渡す力。AIは「言われていないこと」は汲み取れません。何を伝えれば望む結果が出るかを設計します。
- ・ **評価・検証力（Evals）**、生成物を読んで、正しさ・安全性・意図への適合を判断する力。AI時代に**最も重要度が上がるスキル**です。出てきたものを評価できなければ、AIは使いこなせません。
- ・ **任せる／任せない判断**、どこをAIに委ね、どこを人が責任を持つかの線引き。これは技術であると同時に、責任の問題でもあります（29-6）。
- ・ **基礎力の維持**、AIに頼りきると、自分の基礎が痩せます。あえて自分で書く・調べる時間を持ち、「AIの答えを評価できる地力」を保つことが、長い目で効きます。

気づいてほしいのは、この5つのうち多くが、本特論で繰り返し学んできた力（要求を書く・伝える・品質を評価する・責任を持つ・学び続ける）だということです。**AIを使いこなす力の土台は、エンジニアの基礎そのもの**なのです。

とくに新卒が陥りやすい罠が、「基礎力の維持」を軽んじることです。AIがすぐ答えを出してくれるので、自分で考え、調べ、書く機会が減ります。すると一見、速く仕事が進んでいるように見えます。しかし、自分で書いた経験の薄い人は、AIの出力の良し悪しを評価できません。これは「**理解の空白**」と呼べる状態で、大量のコードを受け入れているのに、中身を本当には分かっていない、という危うさを生みます。新卒のうちは、あえてAIに頼りきらず、自分で手を動かして基礎を作る時間を確保することが、長い目で見た「AIを使いこなす力」への投資になります。**AIに任せられるようになるために、まず自分でできるようになっておく**、この順序が大切です。

■ AIを「学びの加速」に使う―壁打ち・個別教師として

AIは「仕事を片づける道具」であると同時に、「学びを加速する相棒」にもなります。分からない概念を噛み砕いてもらう個別教師として、自分の理解をぶつけて試す壁打ち相手として使えます。

ただし答えを写すだけでは基礎は育ちません。自分で思い出し・考え・間違える負荷こそが学びの本体です。学びを加速するAIの詳しい使い方は、第30章30-3で扱います。

29-6 AIの使いどころと注意点

AIは強力ですが、使い方を誤ると大きなリスクになります。新卒が守るべき原則を挙げます。

■ 必ず検証する—「検証の儀式化」

29-3で見たとおり、AIは堂々と間違えます。だから、**AIの生成物は必ず人がレビュー・検証する**、これを、気分次第ではなく**必ず通す手順（儀式）**にすることが大切です。「仕様を書く → AIに実装させる → 人が検証する」という品質ゲートを、習慣として置きます。とくに本番に投入するものは、チェックリストで漏れなく確認します（第25章の品質、第23章のレビュー）。

「動いた」と「正しい・安全」は別物です。ある調査（2025年時点）では、AIが生成したコードは人手のものより脆弱性が多い傾向が報告されています。**「AIが書いたから大丈夫」は通用しません**。むしろ「AIが書いたものは、人が書いたもの以上に疑ってかかる」くらいの慎重さが、現時点では適切です。

検証を「儀式化」するとは、具体的にはこういうことです。「生成されたコードを、一行ずつ自分で読んで意味を理解する」「テストを通す（第25章）」「セキュリティの観点で見る（第26章）」「想定外の入力でどうなるか確かめる」、こうした確認を、面倒でも毎回・必ず通す手順として組み込みます。とくに危ういのが、**よく分からないまま"動いたから"採用してしまう**ことです。理解できないコードを本番に入れるのは、地雷を埋めるようなものです。「自分が説明できないコードは採用しない」を、新卒のうちの原則にしましょう。

■ 機密・著作権を守る

- ・ **機密情報をAIに入れない**、会社や顧客の機密・個人情報を、許可なく外部のAIに入力してはいけません。第26章・第28章で学んだ情報保護そのものです。必ず会社のAI利用ポリシーに従ってください。
- ・ **著作権・ライセンス**、生成されたコードの権利関係には、まだ定まりきらない論点があります（第28章）。鵜呑みにせず、社内ルールを確認します。

■ ガバナンスと「AIが決めた、は免責にならない」

組織としてAIを使う際には、ルール（ガバナンス）が必要です。多くの会社が、AIの利用に関するポリシー（何に使ってよいか、何を入力してはいけないか、どのツールを使うか）を定めています。新卒がまずすべきは、**配属先のAI利用ポリシーを確認し、それに従う**ことです。「便利だから」と自己判断で社外のAIに機密を入れる、といった行為が、重大な事故になりえます。

参考までに、「どこを任せ、どこを人が握るか」の大づかみな線引きを、表で示します（あくまで出発点の目安であり、討議1で自分たちの基準を考えてください）。

AIに任せやすい（人が検証すれば可）	人が責任を持って判断する
調査・情報収集の入口	設計上の重要な判断（アーキテクチャ等）
コード・文書の下書き（たたき台）	本番リリースの可否判断
定型的なテストコードの生成	セキュリティ・個人情報に関わる判断
別の説明の仕方・言い換え	顧客への説明・約束

この線は固定ではありません。**影響の大きさ・検証のしやすさ・責任の所在**という3つのものさしで、現場ごとに引き直します。影響が大きく、検証が難しく、責任が重いものほど、人が握る側に寄せる、これが基本の考え方です。

そして最も大切な原則がこれです。**「AIが決めたことだから」は、責任の免除にならない**。AIの出

力を採用すると決めたのは人間であり、その結果の責任は人間が負います。AIは道具であって、責任の主体ではありません。これは第21章の倫理・説明責任と完全に地続きです。「なぜそれを採用したのか」を、自分の言葉で説明できる状態を保ちましょう。将来、AIの活用がどれだけ進んでも、この一点、**最終的な判断と責任は人間にある**、は変わりません。むしろAIに任せる範囲が広がるほど、「任せた結果を引き受ける」という人間の責任は重くなる、と考えるべきです。

29-7 AI時代にこそ体系が効く

最後に、本特論全体を貫く考えへ戻ります。なぜ「AI時代にこそ、ソフトウェアエンジニアリングの体系が効く」のでしょうか。

理由は、ここまでの各節がすべて示してきました。

- ・ AIは「何を作るべきか」を決めてくれない → **要求の体系**（第23章）が要る
- ・ AIの出力が良いか判断するには → **品質・テストの体系**（第25章）が要る
- ・ AIの生成物の安全性を見極めるには → **セキュリティの体系**（第26章）が要る
- ・ AIに正しく指示し、結果を伝えるには → **書く・伝える力**（第27章）が要る
- ・ AIに任せた結果の責任を負うには → **倫理・専門職の自覚**（第21章）が要る

つまり、**AIという強力な道具は、それを使う人の"地力"を増幅する**のです。体系を持つ人が使えば、生産性も品質も跳ね上がります。一方、体系のない人が使えば、間違いや危険も増幅されてしまう。同じ道具を持っても、**体系の有無が、使い手の差を決定的に広げる**、これがAI時代の現実です。

これを、本特論では「**刈り取り**」と呼んできました。これまで積み上げてきた体系という土台があるからこそ、AIという強力な道具の恵みを、安全に・大きく刈り取れるのです。逆に言えば、AIがあるからと体系を学ばずにいると、その差は開く一方になります。**学び続ける理由（第30章）は、AI時代にこそ強まっている**のです。

ここで、新卒が抱きがちな不安に触れておきます。「AIがこんなに賢いなら、自分が学ぶ意味はあるのか」「エンジニアの仕事は奪われるのではないか」という問いです。歴史を振り返れば（第22章）、新しい技術は繰り返しエンジニアの作業を肩代わりしてきました。高水準言語、フレームワーク、クラウド、そのたびに「楽になる」と言われ、実際に生産性は上がりましたが、エンジニアの仕事はなくなるどころか、より高い視点へと移ってきました。AIも同じ流れの中にあります。**置き換えられるのは「指示通りに作業する」部分であり、残り・むしろ価値が増すのは「課題を定義し、判断し、責任を負う」部分**です。だからこそ、本特論で学んだ「作業でなく仕事をする」エンジニアの態度が、AI時代にいっそう効いてきます。AIを恐れる必要はありません。恐れるべきは、AIに頼って自分の判断力を育てないことのほうです。

【コラム】受託でのAI活用と顧客機密・成果物責任

受託（SI）でAIを活用するとき、とりわけ重い論点が2つあります。

顧客機密。 受託では、顧客の機密情報・本番データ・仕様書に触れます。これらを、許可なく外部のAIサービスに入力することは、情報漏えいに直結し、契約違反・法的責任（第28章）にもなりえます。「便利だから」「ちょっとだけ」は禁物です。**顧客との取り決めと自社のAIポリシーを必ず確認し、許可された範囲・ツールでのみ使う**、これが鉄則です。

成果物責任。 第23章・第28章で見たとおり、受託では成果物に責任（契約不適合責任など）が伴います。AIが生成したコードであっても、それを納めた以上、責任を負うのは作り手です。「AIが書いたので」は、顧客に対する言い訳になりません。だからこそ、AI生成物にも**人による検証**（29-6）を必ず通し、品質を担保する必要があります。AIは生産性を上げますが、**責任までは肩代わりしてくれない**のです。

 ポイント

AIを上手に使う受託エンジニアとは、「速く作れる人」ではなく、「**速く作りつつ、機密と品質と責任を守れる人**」です。その土台は、結局のところ本特論で学んだ体系と倫理にあります。

29-8 考えてみよう（グループ討議）

教室内では手を動かす実技は行いません。3～4人のグループで、考えを言葉にしてみましょう。

◆ 考えてみよう：討議1：任せてよい仕事／人が責任を持つ仕事

ソフトウェア開発のさまざまな作業（調査、下書き、テスト生成、設計判断、本番リリースの可否判断、顧客への説明など）を思い浮かべる。

それぞれを「**AIに任せてよい（人が検証すれば可）**」「**人が責任を持つべき**」に仕分けし、その理由を話し合ってください。

線引きの基準（影響の大きさ、検証のしやすさ、責任の所在など）を、自分の言葉で言えるようになるのが狙いです。

◆ 考えてみよう：討議2：AI時代に、あえて鍛えるべき体系領域は何か

「AIがやってくれるから、もう学ばなくていい」と思える領域と、「AIがあるからこそ、むしろ鍛えるべき」領域を分けてみる。

29-7をヒントに、「自分が意図的に体系を深めたい領域」を一つ選び、なぜそれがAI時代に効くのかを説明してください。本特論を貫く中心の考えを、自分ごととして引き受けるための討議です。

 事例から学ぶ：AIの「もっともらしいウソ」

ある人がAIに調べ物を頼んだところ、実在しない本のタイトルと著者を、さも本当らしく答えてきました。

AIは「それらしい文章」を作るのが得意で、それが事実かどうかは保証しません。

これは性能が悪いのではなく、AIのそういう性質です。

だからこそAIの答えはそのまま信じず、「本当か」を別の方法で確かめる、この一手間が、AIを安全に使うコツです。

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・ AIに何かを聞いたら、その答えが本当か、別の方法で1つ確かめてみる。

29-9 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ AIはなぜ「もっともらしく間違える」のか。
- ・ AIが変えるもの／人に残るものは、それぞれ何か。
- ・ 「AIが決めたから」は、言い訳になるか。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、次の章やこれからの仕事の中で、少しずつ埋めていきましょう。

第30章 活躍できるエンジニアへの成長（43日目）

🎯 この章のゴール

この章では、「学び続ける力」がいちばん古くならないこと、そして学びのコツと長く続けるための心がけを学びます。

終わったとき、次の3つが「わかる・言える」状態を目指します（むずかしく考えなくて大丈夫です）。

- ① **【理由がわかる】** 技術は古くなるが「学び続ける力」は古くならない、とわかる。
- ② **【知っている】** 学びにはコツ（思い出す・間をあける・人に説明する）がある、と知っている。
- ③ **【大切さがわかる】** 長く続けるには無理をしない・困ったら頼る、が大切とわかる。

💡 まず自己診断—読む前に、いまの自分を知らう（2分）

次の2つをやってみましょう（むずかしく考えなくて大丈夫です）。

- ① 現時点での「学び続けることの意味を自分の言葉で説明できる自信の程度」を、1～5で選んで紙にメモします（5＝説明できる／3＝なんとなく／1＝まだ）。
- ② 「身につけた技術は、ずっと役に立ち続けると思いますか？」を、いまの考えで一言だけ書きます（正解は気にしません）。

読み終えたら、章末の「理解度チェック」で同じことをやって、読む前とくらべます。

30-1 概要—なぜ学び続けるのか（一生もののOS／陳腐化しない力）

第7部の旅も終わりに近づきました。ここまで、エンジニアの姿・体系・プロセス・品質・セキュリティ・伝える技術・法律を学んできました。しかし、これらの「知識」には寿命があります。第22章で見たとおり、技術の賞味期限はどんどん短くなっています。今日学んだことの一部は、数年で古くなるでしょう。

では、何が残るのか。**学び続ける力そのもの**です。本章のテーマは、個々の技術ではなく、それらを取り込み続ける**土台（OS）**、どう学び、どう伸び、どう続けるか、です。これは一度身につければ一生使える、最も陳腐化しにくい力です。

💡 ポイント

本章で繰り返し立ち返る軸：AIが知識やコードを安く速く提供する時代、「知っていること」の価値は下がり、「**学び方・考え方・判断**」の価値が上がります。

皮肉なことに、AI時代に最も陳腐化しないのは、AIには代わりにくい「**学び続けるあなた自身**」なのです。

新卒の今、すべての答えを持っている必要はありません。必要なのは、「分からないことを、自分で学んでいける」という確信です。本章はその道筋を渡します。

少し視点を変えると、こうも言えます。第23章で学んだソフトウェアエンジニアリングの体系が「**何を作るか**」の知識だとすれば、本章で扱う学び方・成長・続け方は、その知識を取り込み・更新し続ける「**エンジン**」にあたります。前者は燃料、後者はそれを動力に変える仕組み。エンジンが回り続ける限り、知識はいくらでも入れ替え・更新できます。だからこの章は、第7部の締めくくりにふさわしい、最も長持ちするテーマなのです。

30-2 歴史と背景—OSS・コミュニティの系譜

エンジニアが学び続けられるのは、**知識を共有する文化**が業界にあるからです。その大きな源流が、**OSS（オープンソースソフトウェア）**とコミュニティの歴史です。

大づかみには、1983年の**GNUプロジェクト**（ストールマンによるフリーソフトウェア運動）に始まり、1991年の**Linux**（リーナス・トーバルズ）が共同開発の力を示し、1998年に「**オープンソース**」という言葉と考え方（OSI=Open Source Initiative）が広まりました。

その根底にある思想は、「**共有することで、全員が豊かになる**」というものです。先人が公開してくれたコード・知識・ドキュメントの上に、私たちは立っています。学び続けるとは、この巨大な共有財産から受け取り、いつか自分も返していく営みでもあります（30-7「外へ開く」につながります）。

30-3 学びの科学

「努力しているのに伸びない」、その原因は、根性ではなく**学び方**にあるかもしれません。学習には、科学的に効果が確かめられた原則があります。代表的なものを、新卒の学びに役立つ形で紹介します。

■ 意図的練習（Deliberate Practice）

ただ漫然と手を動かすことは、上達とは別物です。効果的な練習には、(1)**今の実力よりわずかに難しい課題に挑む**（ストレッチ）、(2)**結果へのフィードバックを得る**、(3)**改善点を意識して繰り返す**、という特徴があります。「毎日やっているのに伸びない」ときは、この3要素が欠けていないかを疑ってみましょう。

■ 記憶に残る学び方—想起・間隔・交互・精緻化

学習科学の知見として、次のような方法が定着を高めると知られています。

- ・ **想起練習**、読み返すより、「**思い出す**」こと（自分に小テストをする）が定着する
- ・ **間隔練習**、一度に詰め込むより、**間を空けて**復習する。少し忘れた頃に思い出すのが効く
- ・ **交互練習**、似た問題ばかりでなく、**違う種類を混ぜる**。短期的には難しいが、長期的に身につく
- ・ **精緻化**、新しいことを、**既に知っていることと結びつけ、自分の言葉で言い換える**

共通するのは、「**少し難しい=望ましい困難**」が学びを深める、という点です。スラスラできる楽な復習（再読・詰め込み）は、実は効果が薄いのです。試験前に教科書を何度も読み返したのに点が伸びなかった、という経験に心当たりがあれば、それは努力不足ではなく、方法（再読は楽だが定着しにくい）の問題だったのかもしれません。

これらの原則は、新卒の現場学習にそのまま使えます。たとえば、教わったことをその日のうちに「見ずに思い出して書く」（想起）、一度やった作業を数日後にもう一度たどり直す（間隔）、学んだ技術を「なぜそう動くのか」既知の知識と結びつけて説明してみる（精緻化）。OJTや自主学習を、ただ量をこなすのでなく、こうした原則で設計できる人が、同じ時間で大きく伸びます。

■ メタ認知—「分かったつもり」を破る

最も大切なのが**メタ認知**、自分の理解度を、自分で監視する力です。読んで「分かった気」になるのは危険な錯覚です。自分に説明させてみる、白紙に書き出してみる（章末の「白紙再生」）と、本当に分かっているかが分かります。「**分からないことが分かる**」ことが、**次に学ぶ出発点**になります。

■ アウトプットが最大のインプット

これらの原則を、新卒が日々の学びに落とすなら、答えは一つに集約されます。**アウトプット**することです。教わったことを人に説明する、学んだ技術でミニ作品を作る、調べたことをメモにまとめる。これらはすべて「想起」「精緻化」「意図的練習」を同時に満たします。「インプット（読む・聞く）ばかりで伸びない」と感じるなら、それはアウトプットが足りないサインです。とくに「**人に教える**」のは最強の学習法とも言われます。教えようとする、自分の理解の穴がはっきり見えるからです。新人のうちから、学んだことを後輩や同僚に説明する機会を、むしろ歓迎しましょう。それは相手のためであると同時に、自分の学びを最も深める行為です。

■ AIを学びの相棒にする―壁打ち・個別教師として使い、依存で基礎を痩せさせない

いまや学びの強力な相棒になるのが、生成AIです。分からない概念を噛み砕いてもらう「**個別教師**」として、自分の理解をぶつけて試す「**壁打ち相手**」として、24時間付き合ってくれます。たとえば「この理解で合っているか、間違いを指摘して」と自分の言葉を投げれば、それ自体が精緻化と想起の練習になります。

ただし、ここに落とし穴があります。AIに答えを出させて写すだけでは、楽な再読と同じで定着しません（30-3の「望ましい困難」が消えてしまう）。自分で思い出す・考える・間違える、という負荷こそが学びの本体だからです。AIが答えを肩代わりするほど、皮肉にも自分の基礎は痩せていきます。だから、線を引いて使います。

- ・ **任せてよい**：調べる入口、たたき台、別の説明の仕方、用語の整理
- ・ **自分でやるべき**：思い出すこと、考えて答えを出すこと、「なぜ」を問うこと、最終的な理解の確認

AIに「答え」をもらうのではなく、AIを使って「**自分の理解を鍛える**」。この向き合い方は、ツールが変わっても陳腐化しません。そして、基礎を自分の中に持つ人ほどAIの出力の良し悪しを評価でき、使いこなせます。本特論全体を貫く考え「AI時代にこそ体系が効く」は、学び方そのものにも当てはまるのです（生成AIの仕組みと向き合い方は第29章）。

30-4 自分の現在地を測る―成長段階と技術ロードマップ

学び続けるには、「**今、自分はどこにいるのか**」を知ることが助けになります。

■ ドレイファスのスキル習得モデル

スキル習得は段階を踏む（飛ばせない）。今の自分はどこか？

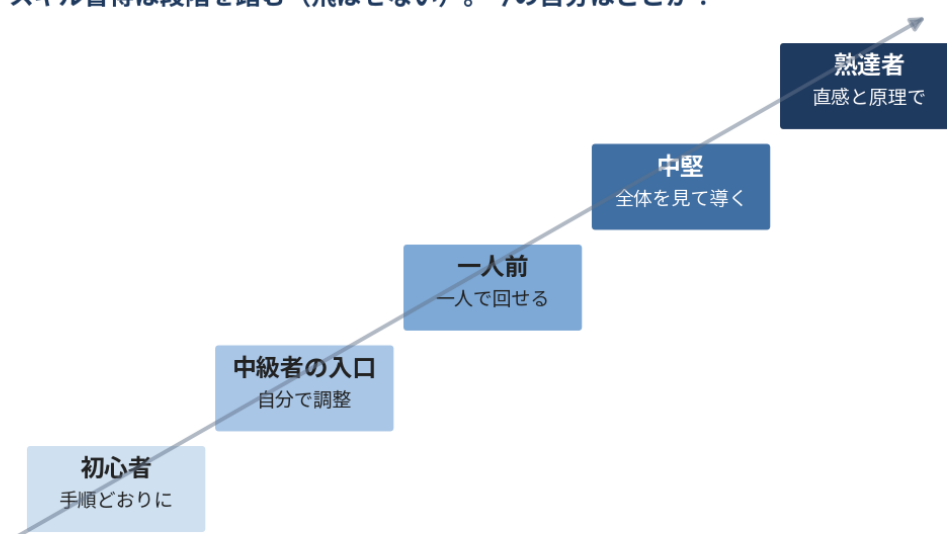


図 ドレイファスのスキル習得の5段階

スキルの習得段階を5つで捉える、よく知られたモデルがあります（ドレイファスモデル）。

1. **初心者**、ルールや指示に忠実に従う段階
2. **中級者の入り口**、学んだルールを、実際の状況に当てはめ始める
3. **一人前**、自分で計画を立てて進められる（学ぶことの多さに圧倒されがちな段階でもある）
4. **中堅**、状況を俯瞰し、何が重要かを見極められる
5. **熟達者**、経験に裏打ちされ、直観的に流れるように動ける

新卒の多くは、1～2の段階にいます。大事なのは、ここで「**自分はまだ初心者だ**」と正しく認識することです。初心者がルールに忠実なのは、悪いことではなく自然なことです。段階を飛ばすことはできません。「今の段階」を知り、「次の段階に進むには何が要るか」を考える、この自己評価が、成長の羅針盤になります。

■ 節目を設計する

成長は、ただ待っていても訪れません。「半年後にここまで」「1年後にこれができる」という**節目（マイルストーン）**を自分で設計し、ふりかえる（第24章のレトロを自分に対して行う）。この習慣が、漫然とした日々を、成長の階段に変えます。

節目を考えると、ドレイファスの段階が指針になります。たとえば初心者の段階では「まず基本の型・手順を、迷わず実行できるようになる」ことが目標です。中級者の入り口に立てば、「教わった型を、実際の状況に応じて使い分けられる」ことを目指します。背伸びしすぎた目標（いきなり熟達者のように振る舞う）は挫折を招き、易しすぎる目標（できることの反復）は成長を生みません。

「**今の自分より少しだけ難しい**」ところに**目標を置く**、これは30-3の意図的練習・望ましい困難と、まったく同じ考え方です。自分の段階を正しく見て、次の一步を適切な高さに設計する。これができる人は、独学でも着実に伸びていきます。

30-5 キャリアをどう設計するか

■ 「情熱を追う」より「価値を積む」—キャリア資本

「好きなことを仕事に」とよく言われますが、キャリアの研究では、むしろ「**希少で価値あるスキ**

ルを磨くと、面白い仕事と裁量が後から付いてくる」という考え方が示されています。この「希少×価値あるスキル」を**キャリア資本**と呼びます。

新卒へのメッセージは明確です。最初から「天職」を探して悩むより、目の前の仕事で**代替されにくいスキル**を意図的に積み上げること。受託の下流からのスタートでも、品質・設計・伝える力・特定領域の知識を積み重ねれば、それは確かな資本になります。資本が貯まるほど、仕事を選べる自由が増えていきます。

■ キャリアは一本道ではない—ICと管理職

経験を積むと、キャリアは専門職（IC：個人貢献者）と管理職に分かれます。「出世＝管理職」ではなく、技術を究めるスペシャリストの道も、正当で価値あるキャリアです（複線型キャリアの詳しい考え方は第18章18-2で扱いました）。

■ 市場価値と資格戦略

自分の市場価値は「どれだけの価値を生み出せるか」で決まります。資格はゴールそのものではなく、断片的な知識を体系立てて学ぶ手段の一つです（資格との付き合い方は第18章18-4で扱いました）。

30-6 メンターとロールモデルの見つけ方

成長は一人では加速しません。**メンター（導いてくれる人）**や**ロールモデル（目標になる人）**の存在が、学びを大きく後押しします。

- ・ **メンターやロールモデルの持ち方（任命は不要、まず「教わり上手」になる、ロールモデルは複数でよい）は第18章で扱いました。ここではエンジニアならではの点を補います。**
- ・ **社外にも師はいる**、書籍の著者、コミュニティの発信者、OSSの開発者。直接知らなくても、その仕事ぶりから学べます（30-7）。

新卒が忘れがちなのは、**質問は迷惑ではない**ということです。適切な準備（自分で調べた上で、要点を絞って聞く。第27章の伝える技術）をすれば、質問はむしろ歓迎されます。聞ける人は、伸びます。先輩の多くは、後輩に頼られ、教えることに喜びを感じるものです（教えることは教える側の学びにもなる、と先に述べたとおりです）。「迷惑をかけたくない」という遠慮が、かえって成長を遅らせ、ときには大きな手戻りを招くこともあります。**抱え込むより、適切に頼る**、これも、長く活躍するための技術の一つです。

そして、いつか自分が教わった分を、後輩に返す番が来ます。メンターやロールモデルから受け取ったものを、次の世代へ渡していく。この循環こそが、エンジニアの文化（30-2のOSSの精神）を支えています。受け取るだけでなく、渡す側にも回っていく、その意識を、早いうちから持っておきましょう。

30-7 外へ開く—発信とコミュニティ

学びは、自分の中に閉じ込めるより、**外へ開く**ほうが加速します。

- ・ **発信する**、学んだことをブログやメモにまとめる。**人に説明できて初めて、本当に理解したと言える**（30-3の精緻化・想起そのもの）。
- ・ **登壇する・話す**、勉強会やLT（短い発表）で話すと、理解が一段深まり、仲間とつながれます。
- ・ **コミュニティに参加する**、社外の勉強会やOSSは「**第三の学び場**」（会社でも学校でもない学びの場）です。同じ技術に取り組む仲間から得る刺激は、社内だけでは得られません。

- ・ **インナーソース**、OSSの「共有して良くする」やり方を、社内に持ち込む考え方。自分のチームの外にも、知識を開いていく。

発信には勇気が要ります。「こんな初歩的なことを書いていいのか」とためらうかもしれません。しかし、**初心者の視点は、同じ初心者にとって最も価値がある**ものです。熟練者は、初心者がどこでつまづくかを、もう忘れてしまっています。今まさにつまづいて乗り越えたあなたの記録こそ、次の誰かを救います。完璧を待たず、学んだことを小さく外に出す習慣が、長い目で大きな差を生みます（受け取った共有財産を、少しずつ返していく、30-2の精神です）。

外に開くことには、もう一つ実利があります。**社外の基準を知ること、自分の現在地が正しく測れる**のです。社内だけにいると、「井の中の蛙」になりがちです。社外の勉強会やコミュニティに出ると、「世の中にはこんなにできる人がいる」と刺激を受け、同時に「自分の強みはここだ」とも気づけます。この健全な比較が、次の学びの目標を与えてくれます。ただし、他者との比較で落ち込む必要はありません。人それぞれ歩んできた道は違います。比べるべきは「他人」ではなく「昨日の自分」、外の世界は、自分を測る物差しとして使い、自分を責める道具にはしないことです。

30-8 燃え尽きずに続ける

最後に、最も大切なテーマを扱います。**長く続けること**です。どんなに優れた力も、燃え尽きてしまっては発揮できません。

■ バーンアウトは「構造の問題」である

バーンアウト（燃え尽き）は、世界保健機関（WHO）の国際疾病分類で「**適切に管理されなかった慢性的な職場ストレスから生じる症候群**」とされ、**職業上の現象**として位置づけられています（医学的な病気そのものではなく、職場の文脈で起きる現象とされます）。エネルギーの枯渇、仕事への否定的・距離を置く感情、効力感の低下、こうした状態が特徴とされます。

ここで強くお伝えしたいのは、**燃え尽きは「根性が足りない」「弱いから」ではなく、働き方・環境という構造の問題**だということです。本特論を貫く「作業者とエンジニアの差は構造の問題で、個人を責めない」という姿勢と、まったく同じです。自分を責める必要はありませんし、周囲の誰かが不調なときも、その人の性格のせいにははいけません。

■ 持続可能に働くために

長く健やかに続けるための土台を、一生もののスキルとして持っておきましょう。

- ・ **境界を引く**、休む時間を確保する。常に全力では続きません。回復も仕事のうちです。
- ・ **健康・メンタル・お金**、体と心の健康、そして生活の基盤となるお金の知識は、キャリアを支える土台です。不調のサインに早く気づき、ためらわず相談する（第21章の「早めの相談」）。
- ・ **助けを求めることは強さ**、つらいときに「つらい」と言えること、人や制度に頼れることは、弱さではなくプロの自己管理です。一人で抱え込まないでください。

エンジニアのキャリアは、短距離走ではなく長距離走です。最初の数年で燃え尽きてしまえば、せっかく積んだ力も発揮できません。逆に、健やかなペースを保てれば、複利のように学びが積み上がります。**「速く走る」より「走り続ける」**、これが、長い目で見たときに最も成果を生む戦略です。無理を美德としない働き方は、自分のためであると同時に、チームと顧客のためでもあります。

■ より広い責任—社会と環境

長く働くなかで、視野は自分やチームを超えて広がっていきます。自分の作るものが社会に与える影響、技術が環境やエネルギーに与える負荷、こうした**社会的・環境的な責任**に思いを向けることも、成熟したエンジニアの姿です。

これは、第21章で導入した**志向倫理**、「してはいけないことを避ける（予防倫理）」の先にある、「**よりよい社会のために、なすべきことを積極的に目指す**」姿勢の、具体的な実践にほかなりません。新卒のうちはまず予防倫理（守るべき一線）から始めますが、経験を重ねるなかで、こうした志向倫理を担う側へと育っていきます。本特論の倫理の縦糸は、ここで最後の結び目を迎えます。

【コラム】受託から伸びる道

「受託の下流から始めると、成長に限界があるのでは」、そう感じる人がいるかもしれません。しかし、それは構造の一面にすぎません。受託という現場には、伸びるための材料が豊富にあります。

- ・ **多様な現場・業務知識に触れられる**、受託は、案件ごとに異なる業種・業務に関わります。金融、製造、流通…と幅広いドメイン知識（業務の知識）は、それ自体が希少なキャリア資本になります。
- ・ **顧客と向き合う力がつく**、要求を引き出し、合意し、説明する経験（第23章・第27章）は、どこでも通用する力です。技術だけの人にはない強みになります。
- ・ **下流からでも「希少×価値」は積める**、目の前の仕事で、品質・設計・伝える力を意図的に磨けば、代替されにくい人材へ近づきます。「言われた作業」を「価値を生む仕事」に変える、第21章の問いが、ここでも効きます。

大切なのは、置かれた場所を嘆くことではなく、**そこで何を意図的に積み上げるか**です。環境は選べないことも多いですが、何を学ぶかは自分で選べます。

たとえば、同じ「指示された機能を作る」仕事でも、「なぜこの機能が必要なのか」を顧客の業務から考える人は、知らぬ間にドメイン知識と上流の視点を蓄えています。「もっと読みやすく直せないか」を考える人は、設計・品質の力を磨いています。日々の小さな問いの積み重ねが、数年後に大きな差になります。受託の現場は、与えられる仕事の幅が広いぶん、こうした「意図的な積み上げ」の素材に事欠きません。**環境を言い訳にせず、環境を素材にする**、それが、どんな場所でも伸びる人の共通点です。

💡 ポイント

ここでも「作業員 vs エンジニア」は構造の問題であり、個人・先輩・自社を責めるものではありません。構造を理解した上で、前向きに自分の資本を積む、それが受託から伸びる道です。

30-9 考えてみよう（グループ討議）

教室内では手を動かす実技は行いません。3～4人のグループで、考えを言葉にしてみましょう。

✦ 考えてみよう：討議1：無理なく10年続ける働き方

「10年後も、心身ともに健やかにエンジニアを続けている」ために、何が必要か。

30-8をヒントに、「**続けるために避けたこと**」「**続けるために大切にしたいこと**」を出し合ってください。

燃え尽きが構造の問題であることを踏まえ、「個人の頑張り」だけでなく「環境・働き方・助けの求め方」の観点も入れて話し合しましょう。

✦ 考えてみよう：討議2：自分の現在地と、次の一歩

30-4のドレイファス5段階で、「今の自分はどのあたりか」を各自考える。

そして「次の段階に進むために、これから半年で意図的に学びたいこと」を一つ、具体的な節目として宣言してください。

30-3の学びの科学（想起・間隔・交互・意図的練習）を、その学習計画にどう組み込めるかも添えてみましょう。

◆ 考えてみよう：討議3：受け取った財産を、どう返すか

30-2・30-7をヒントに、「自分が今、誰かの共有（先輩・書籍・OSS・ネットの記事）に助けられている例」を挙げる。

その上で、「**自分はどんな小さな形で、学びを外に返せるか**」（チーム内のメモ共有、後輩への説明、勉強会での一言など）をグループで具体的に考えてみてください。

📖 事例から学ぶ：上達のしかたは部活と同じ

楽器でもスポーツでも、上手な人は「基礎をくり返し、できなかった所をふり返り、人に教えることで自分も深める」という同じ道を通っています。

仕事の力もこれと同じで、特別な才能より、地道な反復・ふり返り・人への説明の積み重ねで伸びます。

学生時代に何かを練習して上達した経験があれば、その手応えは、そのまま仕事の成長にも使えます。

▶ やってみよう（できそうな人は）

学んだことを、ひとつ行動に移してみましょう。むずかしく考えず、続けられそうな小さな1つで構いません。

- ・ 今日学んだことを1つ、誰かに（または自分に）声に出して説明してみる。

30-10 理解度チェック（読む前の自分とくらべよう）

この章の理解を、かんたんに確かめます。むずかしいことはしません。

■ 思い出して書いてみよう

本を閉じて、次のことを思い出して紙に書いてみましょう（書けたら本文を開いて確かめます）。

- ・ なぜ「学び続ける力」がいちばん古くならないのか。
- ・ 学びのコツを、思い出せるだけ。
- ・ 長く続けるために、大切なことは何か。

■ 読む前の自分とくらべよう

冒頭の「まず自己診断」でつけた1～5の点を、いまもう一度つけます。点が上がっていれば、それがこの章でわかったことです。まだ自信のないところは、次の章やこれからの仕事の中で、少しずつ埋めていきましょう。

終章 研修の総ざらい—正しいエンジニアとして、現場へ（44日目・1日）

🎯 この章のゴール

この章では、44日間の学びをふり返し、総ざらいワークショップでまとめ、正しいエンジニアとして現場へ向かう一歩を決めます。

終わったとき、次の3つができる状態を目指します（むずかしく考えなくて大丈夫です）。

- ①【**ふり返れる**】44日間で学んだことを、自分の言葉でふり返れる。
- ②【**くらべられる**】序章で書いた「大切にしたいこと」と、今の自分をくらべられる。
- ③【**一歩を決められる**】配属後に向けて、まず何を大事にするかを1つ決められる。

💡 44日間の自己採点—研修を終えるあなたへ

各章の冒頭でつけた自己評価を覚えていますか。いま、研修全体を通しての自分を振り返ってください。

この研修で得たのは「知識」です。

それを「行動」に変えるのは、明日からのあなた自身です。

完璧である必要はありません。

迷ったら、このテキストを開き、章末の「現場への持ち出し」を読み返してください。

そのとき読む文章は、入社初日に読んだものとは、違う意味を持っているはずです。

第1の柱 研修の総ざらい—序章から、全44日を

終-1 序章から、全44日をたどり直す—部ごとのふり返し

44日間は、序章と7つの部に分かれていました。最初はバラバラの知識に見えたかもしれませんが、けれど、そのすべては「正しく・よく働くとは、どういうことか」という、**一つの問いでつながっています**。ここで、序章から順に、一度たどり直してみましょう。細かい内容は、忘れていてかまいません。「どこで・何を学んだか」さえ思い出せれば、現場で必要になったとき、いつでもこの本に戻ってこられます。**総ざらいは、暗記の確認ではありません。**“もう、自分はこれを持っている”と確かめる時間です。

■ 序章 なぜあなたはここにいるのか（出発点）

すべての始まりでした。「**入社は入学ではない**」。あなたは、教わってお金を払う側から、**価値を提供して対価を得る側**になった。そう確認し、「自分が大切にしたいこと」を、自分の手で書きました。あの一枚の紙が、44日間の、そして**これから続く長い職業人生の、出発点**です。今のあなたなら、あのとき書いた言葉に、どんな意味を見いだすでしょうか。

■ 第1部 土台をつくる（第1～3章）

社会人の自覚、会社という組織のしくみ、そしてビジネスマナー。ここでは「**学生からビジネスパーソンへ**」の切り替えを学びました。働くとは、お金を払って教わる側から、価値を提供して対価を受け取る側になること。マナーも、細かい作法の暗記ではなく、**相手を大切にすることを形にすること**

でした。派手さはありませんが、ここが、**すべての土台になります**。

■ 第2部 動き方を身につける（第4～7章）

報連相、仕事と作業の違い、人と働く信頼の築き方、文書・プレゼン・会議。毎日くり返し使う**“基本動作”**を身につけました。一つひとつは地味ですが、ここが固いほど、仕事は速く・正確に・気持ちよく回ります。じつは新人のうちに、**いちばん差がつくところ**でもあります。

■ 第3部 考える力・数字の力（第8～9章）

ロジカルシンキングと問題解決、そして数字・財務の基本。感覚や思いつきではなく、**筋道と数字で考え、伝える力**です。「**なぜ、そう言えるのか**」を一步深く問い、根拠で語れること。これが、判断の質と、提案の説得力を支えます。

■ 第4部 正しさを知り、身につける（第10～16章）

正しさの4層、法律、ハラスメント、健全な組織のつくり方、情報を扱う責任、会社と社会を守るルール、そして働く権利。ここでは「何が正しいか」を**自分で判断するための軸**を、徹底して鍛えました。**本書のいちばん太い背骨**であり、現場で迷ったとき、**何度でも立ち返ってくる場所**です。

■ 第5部 自分を育てる（第17～18章）

成長し続ける姿勢と、それを妨げる“内側の壁”、そしてキャリアの設計。会社から与えられるのを待つのではなく、**自分の軸で学び、伸び続けるための章**でした。いわば、**学び方そのものを学んだ時間**です。

■ 第6部 人生という視点で働く（第19～20章）

ライフプラン（お金と人生）、そして「なぜ働くのか」。40年続く職業人生を、**長い目で設計する**視点を持ちました。仕事は人生の大切な一部ですが、**そのすべてではない**。そのバランスを、自分なりに考え始めました。

■ 第7部 特論・ITエンジニア入門（第21～30章）

エンジニアのあるべき姿と倫理、IT産業の現在地、ソフトウェアエンジニアリングの体系、開発プロセス、品質、セキュリティ、ドキュメント、法律、生成AI、そして成長。専門家としての誇りと、実力の土台を築きました。全体を貫く考えは「AI時代にこそ、体系が効く」。道具がどれだけ変わっても、**考え方と学び方は、古びません**。

終-2 4つの正しさで束ね直す一人・社会人・専門家・社員として

学んだことは多岐にわたりますが、第10章の「正しさの4層」で束ねると、すっきり整理できます。あなたは、**4つの顔を同時に持っています**。ひとりの人として、社会人として、専門家（エンジニア）として、そして社員（組織人）として。それぞれに、大切にすべきことと、それを学んだ場所があります。

層	大切にすること（要素）	学んだ場所	崩れると
① 人として	誠実さ・品格・倫理観／人を傷つけない／感情の制御／正直で隠さない	第10・12・6章	人として信頼されなくなる
② 社会人として	法令・契約・社会規範を守る／マナー・報連相／考える力・数字	第1～9・11・16章	法的責任・社会的信頼を失う
③ 専門家として	技術者の責任と誇り／体系・プロセス・品質・セキュリティ・伝える力・AI	第21～30章	システム・社会・利用者に被害

④ 社員として	組織のルールと社会の常識の両立／健全な組織づくり・情報・ガバナンス／長く働く設計	第2・13～15・17～20章	組織の秩序と外部の信頼を失う
---------	--	-----------------	----------------

この4つは、ときにぶつかります。**会社のルールと法律が食い違うとき**、組織の都合と自分の倫理観がぶつかるとき、迷ったら、優先順位①人として→②社会人として→③専門家として→④社員として、に戻ってください。いちばん下で全体を支える土台は、**いつでも「人としての正しさ」**です。

この4つの層は、別々のものではありません。いちばん下の「人として」を土台に、その上に「社会人として」「専門家として」「社員として」が積み重なっています。だから、上の層で迷ったら、一つ下に降りて考えてみる。

終-3 到達の自己点検—白紙に書き出して、たしかめる

総ざらいの最後に、「どれだけ自分のものになったか」を軽く確かめておきましょう。点数をつけるためではありません。**これから何を伸ばせるかを知るため**です。

▶ やってみよう：白紙再生と、自己診断の見返し

本を閉じて、次の3つを紙に書いてみましょう。

- ① 44日間で「最も大切だと思ったこと」を、一言で。
- ② 「正しさの4層（人・社会人・専門家・社員）」を、思い出せるだけ。
- ③ 配属後、最初に大事にする行動を1つ。

次に、各章の冒頭でつけた「自己診断」の点数を見返します。上がってれば、それがこの研修でわかったこと。空欄や低い点は、これから伸ばせる“のびしろ”です。

点が低くても、空欄があっても、落ち込む必要はありません。それは、**これから伸ばせる“のびしろ”**が見つかった、ということです。むしろ、現場に出る前に自分の弱いところを知っておけるのは、**とても幸運なこと**です。

終-4 困ったら、ここへ戻る—逆引きマップ

配属後、困ったときのために、**「こんなときは、ここへ戻る」という地図**を置いておきます。全部を覚える必要はありません。「あのへんに書いてあった」と思い出して、開き直せば十分です。

こんなとき	戻る場所（章・層）
何が正しいか迷う／板挟みになった	第10章「正しさの4層」（迷ったら ①人として を最優先）
報告・相談の仕方に迷う	第4章 報連相／第6章 人と働く・信頼
ハラスメントや理不尽に出会った	第12章（人を傷つけない）／第13章（健全な組織）
情報・データの扱いが不安	第14章 情報を扱う責任／第15章 会社と社会を守るルール
法律・契約が絡みそう	第11章（法律の基本）／第28章（最低限の法律）
成長や学び方に悩む	第17章 成長／第30章 学び続ける
キャリア・お金・働く意味に迷う	第18章 キャリア／第19章 ライフプラン／第20章 なぜ働くのか
技術・設計・品質・セキュリティ	第21～30章（特論）
心が折れそう・燃え尽きそう	第20章／第30章／特別章 特-6（停滞・つまづき）

この地図は、覚えるためのものではありません。「困ったら開く」ためのものです。配属後、壁にぶつかったら、まずこのページを思い出してください。一人で抱え込まずにすむ、それだけで、ずいぶん楽になります。

第2の柱 総ざらいワークショップ—学びを「自分のもの」に

終-5 ワークショップの進め方

総ざらいの仕上げに、学んだことを“自分の言葉”にして外に出すワークショップを行います。知識は、**使って・語って、はじめて自分のものになります**。完璧なプレゼンが目的ではありません。「自分はこれを大切にすると、**自分の言葉で言い切れること**、それがゴールです。

■ 進め方（1日の流れ）

午前は、これまでのテキストとメモ、そして終-1～4の総ざらいを見返して、発表①②を準備します。午後は、一人ずつ発表し、グループで相互フィードバックを行います。緊張しなくて大丈夫。**うまく話すことより、正直に語ることが大切です。**

終-6 発表① わたしの行動指針—4層で自分の判断を語る

一人ずつ、「自分が大切にしたい行動指針」を発表します（一人5分）。コツは、**抽象的な決意ではなく、4層それぞれの“具体的な行動ルール”**として語ることです。

たとえば、こんな形です。①人として、「困ったら、隠さず、すぐ相談する」。②社会人として、「『できる』と『やっていい』は違う。迷ったら法とルールを確認する」。③専門家として、「コードを書く前に、『何のために作るのか』を一度問う」。④社員として、「決めたことは口頭で済ませず、記録に残す」。

あわせて、「今の自分にいちばん足りない層・力はどれか」「配属後の90日で、それをどう埋めるか」も一言添えます。これは、次の特別章（配属後OJTガイド）への、**自分なりの宿題**になります。

✦ 考えてみよう：序章の自分と、今の自分

1日目に書いた「大切にしたいこと」を読み返しましょう。

変わったこと、変わらなかったこと、その両方が、あなたの44日間の成果です。

終-7 発表② グループ・ケーススタディ—4層で優先順位を論じる

グループで、「現場で起こりそうな板挟み」を1つ取り上げ、4層の優先順位（①人として→②→③→④）でどう判断するかを話し合い、発表します。正解を当てるのではなく、**『なぜその順で判断したか』を言葉にすることが目的です。**

📦 例題：こんな板挟み、あなたならどうする？

- ・AIが生成したコードに、出どころ（ライセンス）の分からない部分が混じっている。納期は明日。どうする？（③品質・専門家と②法令と①誠実さがぶつかります）
- ・客先で口頭の追加依頼。仕様書には無い。引き受ける？（②契約・顧客と④組織の整理が要りま

す)

→ どちらも「①人として誠実か」をいちばん上に置くと、筋が通ります。

迷ったら、いちばん下で全体を支える土台、「人としての正しさ」に戻る。これが、**4層という道具の、いちばん大事な使い方**です。

ここから先は、現場へ

終-8 ここからは現場—特別章（配属後OJT）への橋

ここまでが、研修です。**本当に、おつかれさまでした**。ここから先は、配属されて現場に出てからの話になります。詳しい歩き方は、このあとの特別章「配属後OJTガイド」にまとめました。

送り出す側として、最後に一つだけ。**研修と現場は、地続きです**。今日学んだことは、現場で“使って”はじめて本物になります。だからこの研修は、**ゴールではなく、準備でした**。

終-9 正直に言う—現場には、困難が待っている

ここで、正直なことを一つだけ言わせてください。これだけ学んでも、現場に出れば、**最初は大変です**。「研修であんなにできたのに、現場では全然できない」、多くの先輩が、そこでつまづきました。でも、それでいいのです。最初の数十日、いや数ヶ月、**できないのは“当たり前”**です。あなたが劣っているからではありません。**誰もが通る道です**。

現場の困難の多くは、技術そのものより、**「板挟み」の形でやってきます**。納期と品質。言われたことと、正しいこと。あなたの前を歩いている先輩たちも、その前で何度も迷いました。そして、迷いながらも、越えてきました。今あなたのまわりにいる人は、みんな、**その困難を越えてきた人たち**です。

だから、覚えておいてください。つまづいたら、**一人で抱えないこと**。そして、終-4「逆引きマップ」に戻ってくること。困難の越え方は、**もう、あなたの手の中にあります**。

第3の柱 仕上げメッセージ

終-10 それでも、越えていける—正しいエンジニア像を、守り、極める

困難は、あなたを苦しめるためにあるものではありません。**あなたを成長させる“入口”**です。長く活躍する人は、困難をそう捉えています。そして、**その困難を越える武器を、あなたはもう持っています**。

思い出してください。第21章で学んだ「本当のエンジニア」とは、技術が飛び抜けて上手い人のことではありませんでした。**技術を、正しく・誠実に・人のために使える人**のことです。困難の前で、この姿を守り抜けるか、そこに、**本物のエンジニアの真価**が表れます。困難は、あなたを“本物”にしてくれる試金石でもあるのです。

迷ったら、4層の優先順位に戻ってください。④社員として→③専門家として→②社会人として、そして、いちばん上の①人として。どんなときも、**土台は「人としての正しさ」**です。技術やAIがどれだけ進んでも、この軸は古びません。**むしろAIの時代だからこそ**、「何が正しいか」を自分で考えられる人が、**いちばん強いのです**。

終-11 これから数十年の、エンジニア人生と、人としての人生のために

もう一度、心に刻んでおいてください。あなたが新卒として過ごしたこの44日間は、これからの数十年、エンジニアとしての人生、そして一人の人間としての人生の、**土台になる時間でした**。あなたは今日、その土台を、**確かに手にしたのです**。

💡 現場へ持っていく「7つ+α」—4つの正しさで束ねる

- ① 一人の人として — 倫理／誠実さ（迷ったら相談、隠さない）
- ② 社会人として — 法令遵守／顧客第一
- ③ 専門家として — 品質第一／自走力（自分で学び、動く）／変化を恐れない
- ④ 組織の一員として — 学び続け、教える／仲間と協働する（一人で抱えない）／成長／健やかに続ける

ひとことで言えば、倫理。法令遵守。顧客第一。品質第一。自走力。学び続け、教える。成長。この言葉たちを、ときどき思い出してください。どれも、あなたが44日間で確かに学んだ、**「正しいエンジニア」の姿そのもの**です。

完璧でなくて、いいのです。誰かと比べなくて、いいのです。**できない日があってもいい**。迷ったら、ここへ戻ってくればいい。**土台はもう、あなたの中にあります**。技術はいつか古くなります。でも、正しく・誠実に・人のために、**その構えは、これから数十年、あなたを裏切りません**。

研修は、今日まで。仕事は、明日から。あなたの長い物語が、ここから始まります。

ひとりのエンジニアとして、胸を張って、いってらっしゃい！

特別章 配属後OJTガイドー研修の学びを、現場で育てる

🎯 この章のゴール

この章では、配属後のOJTを、あなた自身が主体的に運転し、自分の成長のために使えるようになることを目指します。

終わったとき、次の3つが「わかる・言える」状態を目指します。

- ①【**運転するとわかる**】OJTは「受けるもの」ではなく「自分で設計し、運転するもの」だと言える。
- ②【**自分から動かせる**】計画の共同策定、指導役の活用、フィードバック、振り返りを、自分から動かせる。
- ③【**携えられる**】「正しさの4層」とあるべき姿を、現場でも自分事として持ち続けられる。

特-0 これは、あなたがOJTを「運転する」章

研修の最終日まで、あなたは「研修生」でした。けれど配属されたその日から、あなたはもう研修生ではありません。一人のエンジニアとして、現場の仕事に加わります。OJT（オン・ザ・ジョブ・トレーニング）は「現場で働きながら学ぶ」しくみですが、誤解しないでください。それは**練習ではなく、もう本番の仕事**です。あなたが書く一行、あなたが返す一言の先には、お客様がいて、**その先に世の中があります**。

この章は、これまでの章とは少し役割が違います。第1章から終章までは「知っておくこと」を渡す章でした。この特別章は、渡されたものをあなた自身が現場で使い、自分を育てていくための章です。だから主語は、ずっと「あなた」です。**OJTは、受けるものではありません**。あなたが**自分で設計し、運転するもの**です。

「運転する」とは、こういうことです。多くの現場では、あなたのために指導役（OJTの担当者）が育成計画を用意してくれています。それはありがたい下書きですが、敷かれたレールを、目を閉じたまま走るためのものではありません。走り出す前にレールを確かめ、行き先をすり合わせ、あなた自身が伸びたい方向も乗せて、一緒に、**より良い一本に敷き直す**。それが、この章で身につけてほしい構えです。

✦ なぜ「運転する」のか

OJTは、会社があなたに行う、**最初の育成投資**です。指導役の時間、現場の余力、任される機会、その一つひとつが、あなたに投じられています。

だから、受け身で消化するのはもったいない。投資には、**主体性と「価値」で応える**。これが、**ビジネス感覚を持ってOJTに関わる**ということです（→第9章・第5章）。

罪悪感を持つ必要はありません。あなたは、**投資される価値のある存在**です。だからこそ、自分の手で運転し、宛先であるお客様と、その先の世の中に、**価値を返していきましょう**。

同時に、新卒を育てることは、会社が負う責任です。投資という言葉は、新卒が一方的に恩を返す関係を意味しません。育てる責任と、機会を活かす主体性が噛み合う、対等な関係としてOJTをとらえてください。

そしてもう一つ。運転するのは、作業の段取りだけではありません。研修で確かめた「本当のエンジニアとは何か」（→第21章）、そして「正しさの4層」（→第10章）。あの構えを、現場で自分の手で実践していくこと。それも、あなたが運転します。座学で終わらず、**現場で生きるものに変わ**

えるのは、あなた自身です。

🌀 運転席に座る、3つの構え

主体性：OJTは「受ける」のではなく「自分で設計し、運転する」。計画は、指導役との共同策定です。

宛先意識：自分の仕事の先にいるお客様と、その先の世の中を、いつも意識する。

4層を携える：人として、社会人として、専門家として、社員として。この4つの目を、現場でも手放さない（→第10章）。

特-1 最初の2週間、観察して「運転席」に座る

配属直後は、わからないことだらけで当然です。最初の2週間は、無理に成果を出そうとするより、**まず「観察」してください**。観察といっても、ぼんやり眺めることではありません。運転席に座るために必要な地図を、手に入れる時間です。

■ 二つの地図を手に入れる

手に入れたい地図は、二つあります。一つは「**権限の地図**」。どこまでは自分で判断してよく、どこからは相談すべきか、その線引きです。新人がいちばん事故を起こしやすいのは、この線がわからないまま、よかれと思って一人で進めてしまうときです。もう一つは「**時間とゴールの地図**」。指導役が用意した育成計画の、行き先と区切りです。「最初の1ヶ月で、何ができるようになってほしいのか」を知っているかどうかで、毎日の作業の意味が、まるで変わります。

ここで大切なのは、計画はたいてい「指導役の頭の中」にあって、あなたにすべては共有されていない、ということです。だから、**あなたから確かめにいきます**。確かめにいくことは、**生意気ではありません**。むしろ指導役は、自分の計画をはっきりさせる助けとして、歓迎してくれます。

📁 計画を共同で磨く、4つの問い

- ① このOJTで、最初の1ヶ月・3ヶ月のゴールは、こういう理解で合っていますか。（時間とゴールの地図）
 - ② どこまでは自分で進めてよく、どこからは相談したほうがよいですか。（権限の地図）
 - ③ 私は半年後・1年後に、こういう力を伸ばしたい、こうなりたいと思っています。この計画に、どう織り込めそうでしょうか。（あなたのwill）
 - ④ この計画は、何のため、誰のためのものでしょうか。（宛先）
- ③のwillは「要求」ではなく「提案」です。いま明確でなくても構いません。仮の願いを置いて、走りながら更新していきましょう。

■ 部門の「宛先」を知る

もう一つ、最初に押さえておくとしち上がり速くなることがあります。配属された部門が「**誰に、何を届けているのか**」です。あなたのチームの仕事の先には、必ずお客様がいて、その先に世の中があります。自分の作業がその流れのどこにあるのかが見えると、「言われたから作る」が「**価値を届けるために作る**」に変わります。これは、ビジネス感覚を持って働くことの第一歩であり、社員として、専門家として、現場に貢献していくための土台です（→第9章）。

「すぐに大きな貢献をしなければ」と気負う必要はありません。一年目の貢献は、派手な手柄ではなく、**自分の仕事の宛先を外さないこと**、それ自体です。宛先を知り、そこへ向けて丁寧に一歩を運

ぶ。その積み重ねが、いちばん確かな貢献になります。

▶ 最初の2週間でやること（小さく）

- ・ 育成計画のゴールと区切りを、4つの問いで指導役と確かめる。
- ・ 「ここまでは自分で、ここからは相談」の線を、一度紙に書いて、指導役に見てもらう。
- ・ 「この部門は、誰に何を届けているか」を一行で書いてみる。わからなければ、それを質問にする。

特-2 指導役を「使い倒す」

指導役は、あなたを助けるためにいます。けれど、指導役も、自分の仕事を抱えた一人の社会人です。黙って待っていれば手取り足取り教えてもらえる、とは限りません。だからこそ、**あなたから関わりに入ります**。これを、ここでは「指導役を使い倒す」と呼びます。失礼な意味ではありません。用意された助けを、遠慮して使わないほうが、よほどもったいないのです。

■ 報連相は、自分から設計する

報告・連絡・相談（→第4章）は、指導役のリズムに合わせるだけでなく、**あなたから設計できます**。どのくらいの頻度で、どのタイミングで、どのくらいの細かさで報告すると、お互いに楽か。最初に一度すり合わせておくと、「いつ声をかけていいかわからない」という新人のいちばんの悩みが、ぐっと軽くなります。

■ 1on1は、自分から申し込んでいい

定期的に対面で話す時間（1on1）は、成長にとっても効きます。ところが、これは指導役によっては開かれないことがあります。忙しさにまぎれて、いつの間にか流れてしまう。そんなときは、待たないでください。**あなたから申し込んでいいのです**。「来週、十五分だけでも、お時間をいただけますか。最近の振り返りと、次の目標を相談したいです」。この一言で、成長の場は、自分の手で作れます。

もっとも、1on1や報連相の場を定期的に関開くのは、本来、指導する側と組織の役割でもあります。あなたが自分から動いてよいのは確かですが、それは指導側が場を用意しなくてよい理由にはなりません。

▶ 使い倒すための、ひとこと集

- ・ 相談したいとき：「五分だけ相談いいですか。〇〇について、AとBで迷っています。私はAかなと思うのですが」。
- ・ 進み具合を伝えるとき：「〇〇、ここまで終わりました。次は△△に進みます。気をつける点があれば教えてください」。
- ・ 1on1を申し込むとき：「振り返りと次の目標を相談したいので、近いうちにお時間をいただけますか」。

ポイントは、「丸投げの質問」ではなく「**自分の考えを添えた相談**」にすることです。「どうすればいいですか」ではなく「私はこう考えたのですが、どうでしょう」。これだけで、指導役にとって「**教えやすい新人**」になり、得られる助けの量が変わってきます。

特-3 フィードバックを「取りに行く」

フィードバック（評価や助言）は、与えられるのを待つものだと思っていませんか。受け身で待っていると、それは半年に一度の面談まで来ないかもしれません。それでは遅すぎます。フィードバックは、**あなたから、軽く、こまめに、取りに行くもの**です。

取りに行くコツは、三つあります。一つ、具体的に聞く。「どうでしたか」ではなく「この報告書、わかりやすさの点でどうでしたか」。二つ、早く聞く。完成してからではなく、**途中の段階で「方向は合っていますか」**と聞くと、手戻りが減ります。三つ、受け取り方です。耳の痛いことを言われても、**反射的に言い訳をしない**。「ありがとうございます、次はこうします」と、感謝と次の一手で受ける。これができる人のところに、人はもっと教えたくくなります。

◆ フィードバックを取りに行く、一言

- ・ 途中で：「方向はこれで合っていますか。早めに確かめたくて」。
- ・ 終わった後で：「今回、もっとよくできた点があれば教えてください」。
- ・ 受け取ったら：「ありがとうございます。次は〇〇を直してみます」。

フィードバックは、あなたを否定するものではありません。あなたの仕事を、宛先であるお客様にとって、より良いものにするための、**いちばん安い材料**です。取りに行った分だけ、あなたは速く育ちます。

特-4 振り返りを「自走」させる

同じ一年を過ごしても、ぐんと伸びる人と、そうでない人がいます。**差は才能ではなく、振り返りを習慣にしているかどうか**です。経験は、ただ通り過ぎるだけでは学びになりません。振り返って言葉にして、初めて、次に使える教訓になります。

■ 一日の終わりに、四行

むずかしい振り返りは続きません。一日の終わりに、四行だけ書きましょう。

▶ 振り返りの四行ループ

- ① 今日やったこと（経験）：
- ② 気づいたこと、うまくいった・いかなかったこと：
- ③ そこから言える教訓：
- ④ 次に試す一手（明日の、具体的な行動）：

ポイントは、②の感想で止めず、必ず④の**「次の一手」まで落とすこと**です。これが、経験を成長に変える基本動作になります（→第17章）。

■ 少し背伸びした課題を選ぶ

振り返りと合わせて、もう一つ。できることの反復だけでは、伸びはゆるやかになります。ときどき**「少し背伸びした課題」**に、手を挙げてみてください。完全にできることでも、まったく無理なことでもなく、頑張れば届きそうなところ。その負荷とフィードバックが、成長をいちばん大きくします。

■ 「これは誰のため」を口癖にする

振り返りの軸に、研修で学んだ4層を置いてみましょう（→第10章）。今日の自分は、人として誠実だったか。社会人としてルールを守れたか。専門家として品質にこだわられたか。社員として、部門やお客様に、どう貢献できたか。とくに「**この仕事は誰のため、どんな価値のためか**」を口癖にすると、**作業が仕事に変わり**、宛先を外さなくなります。

特-5 90日から1年の歩き方

最初の勢いは、二、三ヶ月で必ず落ち着きます。そこからの歩き方が、一年目の後半を決めます。鍵は「立てた計画を、**置きっぱなしにしない**」ことです。

■ 30日ごとに、計画と宛先を結び直す

特-1で共同策定した計画は、一度作って終わりではありません。30日ごとに、指導役と短く見直しましょう。できるようになったこと、次に伸ばしたいこと、そして仮に置いた「will」を、いまの実感で更新する。ルールは、**走りながら、一緒に敷き直していい**のです。これが、共同策定を続けるということです。

30日カード（月に一度、五分で）

この30日でできるようになったこと：

まだ難しい、次に伸ばしたいこと：

いまの「will」（伸ばしたい方向・なりたい姿）：

指導役にすり合わせたいこと：

■ 現場の外にも、一歩を出す

もう一つ、後半でぜひ始めてほしいことがあります。現場の外に、小さく触れることです。OJTでは、その現場のやり方を主に教わります。それは立ち上がりには不可欠です。けれど、**現場のやり方が、世界のすべてではありません**。一つの現場に最適化されすぎると、**その現場でしか通用しない人**になってしまいます。

月に一度でいい。社外の勉強会をのぞく、技術書を一冊読む、同期と学び合う。そして、現場で覚えたことを「外の言葉（体系や標準）」で捉え直してみる（→第23章・第30章）。「うちのやり方」を相対化する目が、あなたを、一つの現場に閉じない一人のエンジニアに育てます。順番だけ、間違えないでください。**まず現場で信頼を築くのが先**。そのうえで、外とつなぐのです。

特-6 停滞も、つまずきも、通過点

ここで、正直なことを言います。現場には、研修にはなかった難しさが待っています。できると思っていたことが、できない。叱られる。自分だけ取り残されている気がする。そういう日が、きっと来ます。先に知っておいてほしいのです。それは、**あなたが特別に駄目だからではありません**。ほとんどのエンジニアが、**同じ場所を通過してきました**。

■ 伸びを感じない時期は、必ず来る

成長は、右肩上がりの直線ではありません。階段のように、伸びを感じない平らな時期（プラトー）をはさみながら進みます。**停滞は失敗ではなく、次の飛躍の前の踊り場**です。ここで「自分は向いていない」と決めつけて辞めてしまう人だけが、伸び悩みます。平らな道は、登っている途中の証拠だと、思い出してください。

■ つまづいたときの、受け止め方

放置されている気がするとき。まず、自分から声をかけてみる（特-2）。それでも変わらないなら、抱え込まないでください。ミスをして落ち込んだとき。ミスは、まだ取り返しがつくうちに見つかった、ありがたい学びです。**隠さず、すぐ報告する**。これは、健全な組織で働くために、いちばん大切な構えでした（→第13章）。「向いていないかもしれない」と不安になったとき。たいていそれは、まだ何も「できない」最初の数十日に、誰もが感じる感覚です。三ヶ月後の自分に、もう一度聞いてみてください。

一人で抱えないために

- ・ 困りごとは、指導役にも相談する。言いにくければ、人事や相談窓口という道もあります（→第12章・第16章）。
- ・ 「おかしいかもしれない」と感じることを見聞きしたら、一人で判断せず相談する（→第15章）。
- ・ どの章に戻ればいいのか迷ったら、終章の逆引きマップ（終-4）が、地図になります。
- ・ 相談は、弱さではありません。問題が小さいうちに動ける、強さです。

越えた人がいます。たくさん、います。**あなたも、越えていきます。**

特-7 現場で使う「正しさ」を、自分事として生きる

研修で、「正しさの4層」を学びました（→第10章）。人として、社会人として、専門家として、社員として。座学では、整理された話に聞こえたかもしれませんが、けれど現場では、この4層が、**毎日、具体的な場面として問われます。**

■ 現場は「やり方」は教えても、「あるべき姿」は常には示さない

大事なことを言います。現場は「ここでの仕事のやり方」は、丁寧に教えてくれます。けれど「こういうエンジニアであるべきか」は、毎回、手本として示してくれるとは限りません。だからこそ、その基準は、**あなた自身が携えて、現場に持ち込むもの**です。研修で確かめた「本当のエンジニア」の姿（→第21章・終-10）を、誰かが見せてくれるのを待つのではなく、**あなたが体現する**。それが、4層を自分事にするということです。

■ 迷ったら、優先順位に立ち戻る

4層には、優先順位があります（→第10章）。①人として、②社会人として、③専門家として、④社員として。現場で板挟みになったとき、たとえば「納期のために、品質や正直さを、少しでも曲げてしまおうか」と心が揺れたとき、この順番を思い出してください。**④を守るために①を破ってはいけません。**いちばん上にあるのは、いつも**「人として誠実であること」**です。

迷ったときの、判断の順番

- ① **人として**：嘘をつかない、人を傷つけない、隠さない。
 - ② **社会人として**：法律とルールを守る（→第11章・第15章）。
 - ③ **専門家として**：品質に責任を持つ、ごまかさない。
 - ④ **社員として**：組織に貢献する。
- 迷ったら、上の層を優先する。下の層のために、上の層を曲げない。

■ 不正やハラスメントを見聞きしたとき

もし、不正やごまかし、ハラスメントを見聞きしたら、**見て見ぬふりをする**ことと、**空気を読むことは、違います**。まずは一人で抱えず、**信頼できる人や窓口**に相談してください（→第12章・第15章・第16章）。声を上げるのは、勇気がいります。でも、その勇気を支えるためにこそ、相談先があります。

気をつけたいのは、自分の基準を持つことと、**独りよがりになることは、違う**ということです。現場のやり方が自分と違って、すぐに「現場が間違っている」と決めつけない。まず学び、信頼を築き、そのうえで、どうしても譲れない場面では、**①人として誠実であることに立ち返る**。静かに、しかし確かに、基準を手放さない。それが、現場で「正しさ」を生きるということです。

特-8 OJTは、ずっと続く

OJTには、「いつ終わり」という明確な線はありません。半年や一年で「独り立ち」と呼ばれる日は来ますが、現場で学び続ける営みそのものは、終わりません。むしろ、**ここからが、本当の始まり**です。

技術は、これからも変わり続けます。いま現場で覚えるやり方も、数年で古くなるものがあるでしょう。けれど、**この特別章で運転してきた構えは、古くなりません**。自分で計画を磨くこと。助けを取りに行くこと。振り返って、次の一手を出すこと。停滞を、通過点と知っていること。そして、お客様と世の中を宛先に置き、4層を携えて、誠実に働くこと。これらは、何年たっても、**あなたを裏切りません**。

だから、現場の中で深く学びながら、現場の外にも開いていってください（→第30章）。一つの現場の流儀を超えて、業界やコミュニティ、普遍的な体系とつながる。現場で得た経験を外の言葉で捉え直し、外で得たものを現場に持ち帰る。その往復が、あなたを「一つの会社のやり方ができる人」から「**どこでも通用する一人のエンジニア**」へと育てていきます。

もう一度、確かめておきましょう。OJTは、受けるものではなく、あなたが運転するものでした。レールは、走る前に確かめ、走りながら、より良いものに敷き直していい。あなたの仕事の先には、いつもお客様がいて、その先に、世の中があります。

そして、あなたはもう、研修生ではありません。配属されたその日から、一人のエンジニアとしての道を、あなたは歩き始めています。さあ、顔を上げて、最初の一步を。

いってらっしゃい。現場で、会いましょう！

特-付録 OJT行動チェックリスト

印刷して、机のそばに。OJTの前・最中・あとに、くり返し使ってください。チェックは、できたら付ける。つかない項目こそ、次に伸ばすところです。

① 現場に入る前に（準備）

- ☐ OJTは会社の最初の育成投資。主体性で応えると知っている（特-2）
- ☐ 育成計画のゴールと区切りを、指導役と確かめる準備をした（特-1）
- ☐ 「ここまで自分で／ここから相談」の線を、言葉にしてみた（特-2）
- ☐ 自分のwill（伸ばしたいこと）を、仮で一つ書いた（特-1）
- ☐ 配属先が「誰に何を届けるか」を、調べて一行で書いた
- ☐ 報連相もlonlも、自分から動かしていいと知っている（特-2）
- ☐ 迷ったら「正しさの4層」の①人として、に立ち返れる（第10章）

② 現場に入った後に（毎日・毎週）

- ☐ 計画を指導役と共同で確かめ、willもすり合わせた（特-1）
- ☐ わからないことは、自分の考えを添えて相談している（特-2）
- ☐ 詰まったら一人で抱えず、目安の時間で手を止め相談する（特-6）
- ☐ フィードバックは途中で、自分から取りに行っている（特-3）
- ☐ lonlは自分から申し込み、聞きたいことを用意している（特-2）
- ☐ 一日の終わりに「四行ループ」で振り返っている（特-4）
- ☐ 月に一度は現場の外（体系・業界）にも触れている（特-5・特-8）

③ OJTが一段落したら（自己振り返り）

- ☐ 区切りごとに、計画とwillを自分で更新してきた（特-5）
- ☐ 昔できなかったことで、今できることを三つ言える
- ☐ 伸び悩んだ時期も、通過点として越えてきた（特-6）
- ☐ 「正しいエンジニア像」を実践できた場面がある（特-7・第21章）
- ☐ お客様と世の中という宛先を、見失わずにいられた（終-4）
- ☐ 指導役に学びと感謝を、自分の言葉で返した（特-2）
- ☐ 次の半年で伸ばしたいwillを、言葉にできる（特-1）

【指導役の方へ これは評価表ではありません】

このリストは、新卒が自分のOJTを自分で運転するための、自己点検です。
ここに並ぶ「自分から相談・lonl・フィードバック」を、あなたは受け取りやすくできていますか。
新卒が線を引き直し、willを語り、外とつながるとき、あなたの一言が背中を押します。
手本を示しきれない場面もあります。そのときは隠さず、一緒に「あるべき基準」を探してください。

地方中小企業 新卒エンジニア向け

新卒研修シラバス

Syllabus for New-Graduate Engineers

正しく、誠実に、人のために働く。



クオリティアーツ

<https://quality-arts.com/>

© 2026～ クオリティアーツ
無償公開版（無保証・自己責任利用）