はじめに

1990年代初めのこと、当時私が働いていた職場のマネジャーは「生涯一プログラマーで終わるような人は、うちにはいらない」と何度も語っていたことがありました。エンジニアであっても技術がわかるだけではダメだ。コミュニケーション能力、管理能力が必要だ。プログラマーはプロジェクトマネジャーやコンサルタントへの通過点であって、一生やるような仕事ではない。という考え方です。

その考え方にも一理あることは否定しませんが、私は当時その発言を聞いて大きな違和感を感じていました。「プログラミングというのはそんなに価値のない仕事だろうか？　誰にでもできるような簡単な仕事だろうか？」……私にはそうは思えませんでした。

実際、優秀なプログラマーと平凡なプログラマーの生産性は10倍どころではないぐらい違います。それを一緒くたにして「プログラマーは通過点でしかない」と言い放つことには同意できませんでした。

しかし一方で、技術志向の人間がコミュニケーションをおろそかにするのも、それはそれで問題が起きることにもその後気づかされました。これは最近のことですが、ある金融機関の情報システム部でマネジャーをつとめる人物に取材したところ、「技術がないのにコミュニケーション能力だけがあって重宝されてしまう奴も確かにいる。そういう奴を出世させてしまうと、技術の裏付けがないから最後にはコケるんだけれど、それまでの間、会社にも関係者にも多大な迷惑をかけることになる。そんな事態を防ぐためにも、本物の技術力を持ったエンジニアにこそ、コミュニケーション能力も身につけて欲しい。そのために会社としてお金をかけてサポートしている」という話を聞くことができました。

確かに、技術力があってもそれが周囲に認められないのでは、本人にとっても周囲にとっても不幸なことです。口達者な人間を目指す必要はありませんが、きちんと伝えられるだけのコミュニケーション能力は持っておくべきでしょう。

エンジニアが「職務上、必要なことをきちんと伝えられるコミュニケーション」をするために最も重要なのは、「誰に」「何を」「何のために」伝えるのかを常に意識し、その情報を整理整頓しておくことです。これはコミュニケーションの手段が文書であろうと会話であろうと変わりません。したがって、本書は主に「文書」を通じたコミュニケーションを想定して書いていますが、本書で推奨する習慣は口頭での説明力を伸ばすためにも本質的に役に立ちます。

「プログラムは論理的に正しくコードを書けば動くけれど、人間にはどう書けば／話せば伝わるのかわからない。だから難しいし苦手だ」という苦手意識のあるエンジニアは少なくないと思いますが、実際にやってみれば意外に「やれる」ものです。本書では、「1日あたり数分のちょっとした努力の継続でできる、ライティング力アップ法」を紹介しています。ぜひ、食わず嫌いで敬遠することなく「伝わるドキュメントが書けるエンジニア」への道を歩んで行ってください。

開米瑞浩

目次
CONTENTS

はじめに 2

第1章
この本の使い方

1.1 「文書が書けない」悩みも人それぞれ 10
1.2 悩み別・本書の使い方ガイド 11
1.3 「紙を書く」ことを馬鹿にする者に未来はない 16
1.4 技術力を活かすためにも、書く力を身につけよう 17

第2章
事例で探る「わかりやすく書く」ための6ポイント

2.1 技術者といえども「ドキュメント」を避けては仕事ができない 20
2.2 「文章のテクニック」に走る前にやるべきことがある 21
2.3 「文章を書く前にやるべき9割」とは何か？ 22
2.4 最も基本的なポイントは6つ 23
2.5 事例 次世代ストレージ・システムの意義 24
2.6 ポイント1 分類してラベルをつける 25
「ラベル」は大量の情報の断片を関連づける手段／「問題」に直面していない人間は「ラベル」を必要としない／ラベルを考える努力は、問題を主体的に解決しようとする努力／「とりあえず考えたラベル」はたいていどこかおかしい／分類・ラベリングは「文章を書く」前に行うのが基本
2.7 ポイント2 意味のある順番で並べる 36
2.8 ポイント3 数量化してグラフを使う 38
2.9 ポイント4 メッセージを入れる 39
「メッセージ」は絞り込むほど効果が高い／「メッセージを絞り込む」ためには相手を知らなければならない
2.10 ポイント5 相手の価値観を考える 43

2.11 ポイント6 情報量を減らす 45
2.12 図解の元になった文章はどうする？ 46
2.13 難解なテクニックは不要だが覚えただけでは使えない 47

第3章

「書く力」向上作戦の進め方

3.1 まず何から始めればいい？ 50
大きな労力がかかるものは後回しでかまわない
3.2 一番手軽で応用が効く方法は「とりあえず分割してみる」こと 51
3.3 分割に困ったときこそ、成長のチャンス！ 55
3.4 分割することはイコール分類すること 57
3.5 自分の理解を要約したラベルをつける 58
3.6 分割で悩むような文章は捨てて手っ取り早く成果を出そう 60
3.7 さらに3段構えで「書く力」のパワーアップ作戦を考える 61
3.8 ステップ1 動機を確認する 62
銀の弾丸を求めず地道な継続を！／絞り込めば効率よく成果が上がる
3.9 ステップ2 型を真似る 65
「まっすぐ順番につながる型」はどこにでもある
3.10 ステップ3 引き出しを増やす 69
3.11 必要なのは「情報」であって「文章」ではない 70
3.12 「良い文章」はかえって読者を誤解させやすい 73
3.13 新聞や雑誌の記事をお手本にするのは大きな勘違い 73

第4章

ワークフローの中での文書の役割を考える

4.1 エンジニアの声が経営者を動かせないのはなぜ？ 76
[解決策1]「言って」ダメなら「書いて」みる／[解決策2]「書いて」ダ

メなら「わかりやすく書いて」みる／[解決策3] 誰が何のために読むのかを想定して書く
4.2 ビジネスの基本は状況把握・方針立案・行動提案の3段階 78
4.3 使うタイミングを間違えていませんか？ 80
4.4 「状況把握」段階では情報量の削減をしてはいけない場合がある 80
4.5 「方針立案」段階では典型的な「型」がある 82
[理由1] 課題と解決策を語っていなかった／ [理由2] 問題の本質を語っていなかった
4.6 「行動提案」は、方針を相手の受け取りやすい形に整えること 86
4.7 状況把握の「型」は千差万別 89
4.8 「Simple is best」が常に正しいわけではない 90
4.9 現実をありのままに表現した図が必要なときもある 94

第5章

IT関連ドキュメント構造化事例集

5.1 見て、真似て、考えて、身につけよう 96
5.2 事例1 誤配送連絡 97
5.3 事例2 クラスタリング技術の結集 99
5.4 事例3 システム構成テンプレート化構想 103
5.5 事例4 ファイルサーバー性能測定ツール 106
5.6 事例5 電池の持ち時間をグラフで表す 108
5.7 事例6 パソコン用インタフェース 111
5.8 事例7 通信機器の障害防止策 115
5.9 事例8 Apacheの設定ファイルに関する説明文（1） 119
5.10 事例9 Apacheの設定ファイルに関する説明文（2） 122
5.11 事例10 Apacheの設定ファイルに関する説明文（3） 123
5.12 事例11 Apacheの動作に関する説明文 125

5.13 事例12 SSIDとは 128
5.14 事例13 マルチSSID機能 130
5.15 事例14 インシデント対応活動 133
5.16 事例15 機密性・完全性・可用性 135
5.17 事例16 ウイルス、スパムメール対策 137
5.18 事例17 個人情報保護法ガイドライン 140

第6章

ドキュメンテーション作業チェックリスト

6.1 文書作成中に悩んだら、いつでも確認！ 148
6.2 対象者・ワークフロー・表現の3分野でチェックする 150
6.3 「対象者確認」…誰がいつ何のために読むものか？ 151
6.4 「ワークフロー確認」…状況把握、方針立案、行動提案 153
頻出パターン：並列／順序／工程／ペア概念／因果関係・相関関係／全体と部分／抽象・具体／特性・適性／目標・手段・働き・効果／目的・目標・手段／定性・定量／用途・機能・仕組み／フロー＆コメント
6.5 「表現調整」…用語の選択、概念のまとまり、数値表現 164
概念のまとまりを示す／適切な用語の選択／数値の表現

第7章

人間の認知情報処理モデルの観点から

7.1 人の脳内には2種類の情報処理回路がある 172
7.2 同時処理型・継次処理型はそれぞれ「得意技」が違う 173
7.3 「文章」は継次処理型に適した表現手段 174
7.4 ネットワーク型の構造は図解でなければ表現できない 174
7.5 ネットワーク型情報を図解するときは継次処理型向けの手を加える 176

7.6 図解に決まった手法はない 177

第8章

ドキュメントだけでは コミュニケーションは改善しない

8.1 「コミュニケーション」って何だろう？ 180
8.2 『カッツの3能力』から「コミュニケーション」を考える 181
8.3 テクニカル・コミュニケーションでは「文書」情報の山ができる 182
8.4 ヒューマン・コミュニケーションでは短時間・多頻度の顔合わせが重要 183
8.5 「ヒューマン」ができていないと「テクニカル」にも支障が出る 184
8.6 「目標をイメージさせる」のがコンセプチュアル・コミュニケーション 187
8.7 コンセプチュアル・スキルの核は価値観、抽象化、語彙 189
「抽象化」のスキルは鍛えておこう／「語彙」を増やすためにしておきたいこと
8.8 お互いに切磋琢磨するコミュニティに属していますか？ 192
8.9 ドキュメントだけではコミュニケーションは改善しない 196

あとがき 198

1章

この本の使い方

システムの開発や運用の仕事に携わっていると、報告書、説明書、提案書などの「文書」を書く必要に迫られます。そこで、「必要な情報が短時間で正確に伝わるようなわかりやすい文書を書く力」があると仕事が楽になりますが、「文書」といってもその種類はさまざまです。普段、提案書を書かない人が提案書の書き方を学んでもあまり役に立ちません。そこでまずは自分が普段どんな種類の文書を書くことに悩んでいるかを思い出してみましょう。この章でいろいろな「悩みの声」を読み、自分に近いものを探してください。

1章 この本の使い方

1.1 「文書が書けない」悩みも人それぞれ

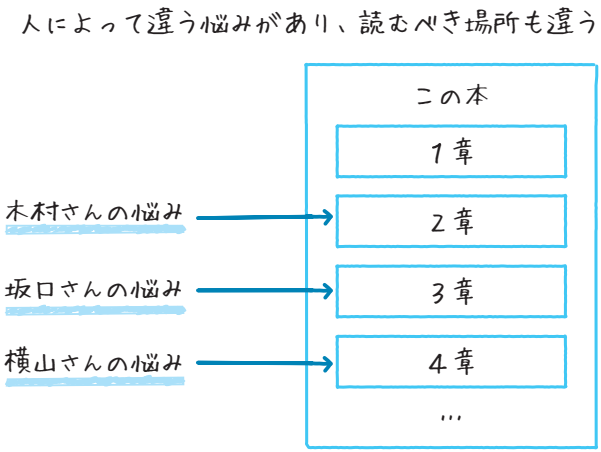
私は1988年から15年間ITエンジニアとして仕事をした後、2003年からは企業研修業界に転じて「文書作成能力」「わかりやすく書く力」の向上研修を手がけてきました。そのため研修依頼もIT業界からのものが多く、「プログラムは書けるけれど、文書は書けない。報告書、説明書、提案書を書くのが苦手……」というITエンジニアの悩みをよく聞きます。しかしその悩みを具体的に聞けば聞くほど、人によって千差万別だなあ、と思わざるを得ませんでした。

たとえばある会社から研修の依頼を受けた際、先方の部長さんはこんな話をされていました。

> このところ部下の多くから「説明書、提案書を書くのが苦手」という声を聞いています。「何を書くべきか、コンテンツの整理ができていない」のと「コンテンツを伝わりやすい形に表す方法がわからない」のと、人によって苦手分野は違うようなのですが……

はい、確かにその2つは違う問題です。ですので、同じ方法では解決できません。

この会社のケースに限らず、「わかりやすく書けない」「せっかく書いた文書が役に立たない」悩みは人によってずいぶん違います。ということは、本書の中でどれが役に立つかも人によって違うはずです。そこで第1章では「悩みの種類に応じて、本書のどこを読むべきか」をご案内することにします。



▲図1　本書の読み方も人それぞれ

この本は、1ページ目から順番に読んでいただく必要はありません。この章の案内を参考にして「今の自分の悩み」に直結しそうな場所を探して読んでください。ある悩みが解決すれば今度は別な悩みが浮上するはずで、そのときはまた同様に「今の悩み」に直結する場所を読んでください。そうやってその都度その都度「目の前の悩み」を解決するための手がかりを探し、実践していけば、いつの間にか「書けない」という悩みは解消していることでしょう。

1.2 悩み別・本書の使い方ガイド

それでは具体的な悩み別に本書のどこが関係するかをご案内します。

悩み：ユーザーから要求仕様のヒアリングをすると箇条書きがたくさんできますが、箇条書きのままだと全体としてどう関連付いているかがわからず、ユーザーとイメージを共有できません。そこで全体のイメージを要領よくまとめた図に表したいのですが、どう書いていいのかわからず困っています。

答え：箇条書きや文章で書かれた情報を元にして、そのイメージがわかる図解をしたい、という場合は、第2章「事例で探る「わかりやすく書く」ための6ポイント」の最初の2つが重要です。後は第5章「IT関連ドキュメント構造化事例集」から、似たケースを探してみてください。

悩み：自社の現行のシステムには重大な懸念事項があるため、早急に対応する必要があることを報告したのですが、取締役に「お前の話は要するに何が言いたいのかわからん！」と言われてしまい、かといってどう伝えればわかってくれるのか見当もつかないので、気力がくじけてしまいそうです。

答え：自社のシステムに関する悩みだそうですが、顧客のシステム向けでも同様の問題は起きますね。この場合は現場の技術的な問題を把握しているエンジニアと、ビジネス上の判断をしなければいけないマネジメント（経営者）層の意識の違いを埋めなければいけません。まずは、第4章「ワークフローの中での文書の役割を考える」で、組織を動かすための基本的なコミュニケーションには「状況把握」「方針立案」「行動提案」の3種類の型があることを知ってください。それから、第2章「事例で探る「わかりやすく書く」ための6ポイント」のポイント3～6が重要です。

悩み：ユーザーさんに提出するシステム仕様の説明書や障害報告書について、「専門用語ばかりでわかりにくい」と言われてしまいます。

答え：専門用語の使い方についての悩みも多いです。第6章「ドキュメンテーション作業チェックリスト」の「適切な用語の選択」に関する解説をごらんください。また、この問題はユーザーが認識しやすいパターンを応用することで解決することもあります。同じ第6章の「頻出パターン」や第5章「IT関連ドキュメント構造化事例集」か

ら類似例を探してみてください。

悩み：特定のシチュエーションで悩んでいるというよりは、ありとあらゆる場面で「文書を書く」のが苦手です。根本的にスキルが足りないと思うのでなんとかしたいのですが、何から手をつけて、どんな段取りで進めていったらいいのかイメージがわきません。

答え：四方八方、できないことばかりだと、何から始めていいのかわからず立ちすくんでしまいますよね。第3章「「書く力」向上作戦の進め方」で、そのあたりの悩みにお答えしています。人間は少しでも成果があればやる気が出ますので、とにかくすぐにやれることから手をつけて、少しずつでも「うまくいった！」という成果を出していくのがポイントです。

悩み：私が「わかりやすい」と感じる文書と、顧客や同僚がそう感じる文書が食い違うことが多くて不思議です。これには何か理由があるのでしょうか。自分の感覚で書いたものが通じない場面が多いと、書くのが怖くなってしまいます。原因がわかればまだ納得もできるし対処のしようもあると思うのですが。

答え：このケースに当てはまるかどうかはわかりませんが、第7章「人間の認知情報処理モデルの観点から」を読んでみてください。人間が見聞きして得た情報を「理解」するまでの脳内での情報処理の形態には大まかに同時処理型と継次処理型の2種類があり、それぞれ得意技が違い、「わかりやすい」と感じるパターンも違うという話です。

悩み：情報処理業界の展示会に自社ブースを出すことになりまして、製品紹介プレゼンもしなければいけません。プレゼンの準備をしているのですが、何人かに聞いてもらったところ、理路整然としているけ

れどイマイチ入り込めない、興味を引きつける何かが欲しい、インパクトが弱い、などと言われてしまっています。

答え：製品紹介プレゼンというのは、端的に言えば「うちの商品を買ってください」とアピールする場です。ただ単に「買ってください」では説得力がないので、「御社の課題を解決できます。その理由はこれこれです」といった説明が必要ですね。これに関連のあるパートはいくつかあります。第2章「事例で探る「わかりやすく書く」ための6ポイント」のうち、「メッセージを入れる」と「相手の価値観を考える」「情報量を減らす」の後半3つ。第4章「ワークフローの中での文書の役割を考える」の中の「方針立案」「行動提案」のパート。第6章「ドキュメンテーション作業チェックリスト」の中の「関係者確認」や「表現調整」のパート。などが該当します。

悩み：後輩のITエンジニア向けの教育用資料を作ろうとしています。ついては、WebサーバーやDBMSなどの設定方法について、単に項目ごとの手順を示すだけでなく、全体としてどう動くのかイメージがわくような形に書いてやりたいのですが、参考になる例はありませんか

答え：第5章「IT関連ドキュメント構造化事例集」から、Apacheの設定ファイルや動作についての事例、パソコン用インタフェースの事例、通信機器の障害防止策の事例、個人情報保護法ガイドラインの事例などが参考になると思います。

悩み：仕様書を受け取って、理解したつもりで作り始めて、後で誤解していたことがわかって手戻りを強いられたことが何度もあります。そのたびに「気をつけて読めよ！」と注意されるのですが、やっぱり同じミスをしてしまうので、自己嫌悪になりそうです。「書く」悩みよりは「書かれているものを読めない」悩みなのですが、何か打

つ手はありますか。

答え：「書く」よりも「読む」ほうがうまくいかない悩みということですが、実は「書く」力と「読む」力の間には密接な関係があります。「わかりやすく書く」ためには、書くための根拠になる情報をきちんと整理分類しなければいけないので、「読む」力が必要なのです。そのため、本書の中でも「書く」力を磨くために推奨している習慣の多くは、実際には「正確に読む」ために役に立つ習慣そのものです。

特に、第2章「事例で探る「わかりやすく書く」ための6ポイント」の最初の2つのポイントを読んで実践してみてください。

悩み：見込み客へのシステム提案をすることになり、提案書を書かなければなりません。当社の技術力は同業者間では高く評価されているのですが、先方の意思決定権者には技術の細部に入り込んだ説明をしても通じそうにありません。当社の技術力が先方のビジネスにどう役に立つかということを、どう整理して伝えたら良いか悩んでいます。

答え：第5章「IT関連ドキュメント構造化事例集」の「通信機器の障害防止策」の事例や「ファイルサーバー性能測定ツール」の事例、それから第4章「ワークフローの中での文書の役割を考える」の中の「方針立案」「行動提案」のパートが参考になります。

いかがでしょうか。「おお、これこれ」という、思い当たるものがあれば、そこから読んでみてください。もちろん、時間があるなら最初から順番に読んでみるのも良い方法です。

1.3 「紙を書く」ことを馬鹿にする者に未来はない

何にしても、「書く力」は今後のIT技術者にとって非常に重要なスキルになってきます。その上で少々気になるのは、IT企業のマネジャー職から現場のエンジニアまで、このことを理解していない傾向がちらほら見られることです。たとえば私は本書を書くためにあるSIerのインフラ技術者を取材した際、「うちの部長は“要するに紙を書くだけだろ？”とか言って、ドキュメント作成を馬鹿にする傾向があるんですよ」という嘆きを聞いたことがありました。「紙を書く」という表現は他にも系列の違う複数社で聞いたことがあり、IT業界では慣用句としてよく使われているようです。だいたいの場合良い意味ではなく、エンジニアにとっては雑務であるとか、誰でもできるハンパ仕事、のようなニュアンスで聞くことが多いですね。

ITの仕事というのは、きちんと動くコードを書いたりテストをしたりする負荷が極めて大きいのに対して、プログラムと違ってコンパイルエラーを起こすわけでもなく、多少の誤字脱字があっても人間が読むときに補ってくれてしまう「ドキュメント」はどうしても軽視される傾向があります。その結果が「紙を書く」という慣用句なのでしょう。しかし、今後はそういうわけにはいきません。というのは、全般にシステム開発に使える時間が短くなっているため、コミュニケーションギャップが致命傷になるケースが増えているからです。開発期間1年のプロジェクトでの1週間のロスはなんとかなりますが、1ヶ月の中の1週間を無駄にしたら深刻な結果を生んでしまいます。

別な見方をすると、今後は「現実には使われない、無駄なドキュメント」を書いている余裕はありません。これまでの開発現場ではその種の「誰も見ない文書」が山のようにあったのが実態でした。私が経験した中では、あるITベンチャーが有名なオブジェクト指向開発の方法論を導入し、社員に研修を受けさせてその方法論に即したドキュメントを量産したものの、結局誰も使わなくなったケースがありました。これでは「紙を書く仕事」が軽視されるのも無理はありません。「方法論」そのものは有益なものであったとしても、それを何も考えずに丸呑みコピペのような形で導入するとそういう事態を招きます。

ですから、重要なのは「自分の現場では何が必要なのかを自分で考える」ことです。この本も含めて、教科書丸写しのようなやり方をするのはやめましょう。本章で既に紹介したように、「文書が書けない悩みも人それぞれ」です。「今はどの情報が誰に理解されていないのか」を個別に考え、それに必要な文書はどんなものかを考えて書くようにしてください。この本はそのためのヒントが得やすいように構成してあります。

大事なことなのでもう一度書きますが、この本も含めて教科書的方法論を鵜呑みにするようなことはやめましょう。それは容易にデスマーチ（長時間残業の連続）を招くやり方です。

1.4 技術力を活かすためにも、書く力を身につけよう

そうは言っても自分はコードを扱う技術者であって、「紙を書く」ことに力を入れるのは気が乗らない、という方も少なからずいることと思います。そんな方は以下の2つの質問を自問自答してみてください。

質問1： あなたがあらかじめ予想して警告していたにも関わらず、それを会社が聞いてくれなかったために予想通り「困った事態」が起きてしまい、残念な思いをしたことはありませんか？

質問2： 自分が過去にあれこれ調べて理解した技術に関して、同僚がさんざん悩んで失敗していたことを後で知り、「相談してくれればよかったのに」と思ったことはありませんか？

この2例はどちらも、「せっかくの技術力が活かされていない」というケースになります。「技術力」は人の役に立ってナンボです。1つ目の例ではもし自分が危険性を「わかりやすく説明する」ことができていたら、困った事態は避けられたかもしれません。2つ目の例ではもし自分が調べたことを簡単なレポートにして社内ポータルにでも載せていれば、相談してくれていたかもしれません。ちなみにこれは実際に私の知人が実践していた方法です。30年以上前

に高卒で大手電機メーカーに就職した彼は、業務から得られた知見を技術レポートにまとめて報告し続けた結果、その能力が認められて高卒ながら新分野の重要な新製品開発に抜擢されたそうです。

「技術力」を活かすためには、あなたの技術的知見からどんなことが言えるのかを周囲に理解してもらわなければなりません。また、あなたが技術を持っていることを他の人に知らせることが必要です。それができないと、せっかくの技術力が活かされずに埋もれてしまいます。そのために役に立つのが「わかりやすく書く力」なのです。そしてその力は毎日数分ずつでかまわないので、意識してトレーニングすることで意外に短期間で向上します。食わず嫌いにならずに、ぜひチャレンジしてみてください。そのためのやさしい手引き書になるように、ということを意図して私はこの本を書きました。

改めて書きますが、悩みが人それぞれであるように、本書の使い方も人それぞれです。「目の前の悩みに今すぐ答えること」と「じっくり時間をかけて『わかりやすく書くスキル』を伸ばすこと」の双方に、ぜひ本書を活用してください。