電子書籍ダイジェスト版

インプレスの技術メディア
「Think IT」の人気連載を書籍化。
はじめに〜第1章を収録!

エンジニアなら
知っておきたい

システム設計とドキュメント

クラウド時代のシステム仕様の
まとめかたを解説

梅田弘之著

いま、どんな情報を共有すべきか

チームプレーでシステムを
構築するために、準備すべき情報と
顧慮すべき課題がわかる!

インプレス

はじめに

エンジニアは、ともすると目の前のシステム開発作業に追われて、技術が狭い範囲にとどまってしまいます。先日、社内の若手エンジニアとラウンドテーブルを行った際に、「非機能要件ってどんな項目がある？」「データ中心設計ってなに？」「ゼロトラストの考え方って？」などのボールを投げかけたのですが、意外ときちんと答えられる人が少なくて焦りました。実際、システムエンジニアが知っておくべき知識は非常に幅広く、それらを理解したうえでシステムを設計・開発するのと、知らないで作業しているのとでは雲泥の差があります。引き出しが少ないと、本質を捉えたシステムが作れないし、自身の成長に繋がりにくいし、なによりシステム作りの楽しさが半減すると思います。

そこで本書は、次の4つのテーマを意識して執筆しました。

・SEが知っておくべき幅広い知識や技術を学ぶ

本書では、システム設計書やドキュメントの作成など設計業務を中心に解説していますが、より幅広い知識を数多く盛り込んでいます。単なる設計テクニックだけでなく、システムエンジニアが知っておくべき知識や技術も一緒に学ぶ本だと考えてお読みいただければと思います。

・知識や技術を全体像として理解する

知識や技術は、IT用語辞典で断片的に読んでも本質までは理解できません。そのため本書では、できるだけ俯瞰的に技術を説明しています。「システム開発のドキュメント体系」「非機能要件の定義項目」「運用設計書の項目」などを整理して解説していますので、体系的に全体像を理解してください。

・取り巻く世界や本質を理解する

本書では、「プロジェクトの失敗要因」や「システム開発の実態」などをデータを使って解説しています。目の前のシステムだけでなく、ちょっと目を離して周囲の状況を理解したうえで取り組んだ方が良いからです。また、「設計書を作

成する理由」や「アジャイル開発における設計書」など原点に立ち返って本質を理解することも重視しています。

・知識や技術を柔軟に使いこなす力をつける

エンジニアは、とかく0か1で考えがちですが、世の中ははるかに複雑で、システム開発のやり方も常に変化しています。昔からやっているやり方を疑いを持たずに続ける代わりに、ツールや新しい手法を使って効率的にシステムを作る柔軟な感覚を磨いてください。

ぜひ、本書で幅広い知識を全体像として身に着け、柔らかい頭でそれらを活用するスマートなシステムエンジニアになってください。心から応援しています。

2022年1月　梅田弘之

本書の対象読者

本書では、読者として次のような方々を想定しています。

- これから設計書も書くことになったプログラマー
- もっと効率的な設計書の書き方を模索したいシステムエンジニア
- 設計書の標準化を進めて、組織的に設計効率を上げたいリーダー
- アジャイル開発における設計書のあり方に悩んでいるリーダー
- 保守・運用コストが肥大化し、保守運用負荷を軽減したいマネージャー
- RFPに非機能要件をきちんと書きたいユーザー
- 運用設計書を標準化したい運用担当者

今まで、なんとなく見よう見まねで開発設計書や運用設計書を作成していたという人も多いと思いますが、一度、原点に立ち戻ってシステム設計書を書く意義や書き方について見つめ直してみましょう。そして、令和時代に合った良い設計書の書き方を理解し、自分の仕事に活かしてください。

contents

contents

contents

第1章

システム開発プロジェクトを成功させるリスク管理

システム開発のプロジェクトは、この数十年で進化しているのでしょうか。プロジェクト失敗の話は巷でよくありますし、「だからSIerはだめなんだ」という論調もよく聞きますが、実態はどうなんでしょう。エンジニアなら、まず、自分たちのシステム開発がどのような状況にあるかを知っておきたいと思います。そこでシステム開発のドキュメントという本書のテーマに入る前に、プロジェクトを失敗しないポイントについて解説します。

プロジェクトの成功率はどう変化したか

最初に質問です。「この15年でプロジェクトの成功率は変化したでしょうか」（図1-1）

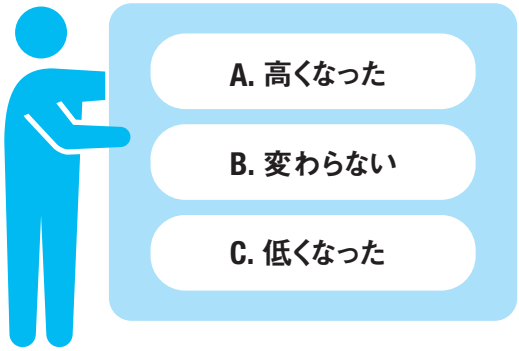

図1-1 プロジェクトの成功率は15年でどう変化したか

(1) プロジェクトの成功率の変化

システム自体が複雑化して難易度が増している感じはします。また、「最近のエンジニアは技術の基本ができていなくて…」などと嘆くベテランの声も時々聞きます。一方で、昔の方が炎上プロジェクトの話をよく聞いたような気もします。

客観的な数値で確かめたいですね。実は『日経コンピュータ』が2003年、

2008年、2018年にアンケート調査した結果を『日経ビジネス電子版』(共に日経BPの媒体)が公開しているデータがあります。**図1-2**は、その結果を筆者がグラフ化したものです。これによると2003年には26.7％しかなかったプロジェクト成功率が、2018年には2倍の52.8％まで高まっています。つまり正解は「A:高くなった」で、なんと15年で成功率は2倍になっているのです。

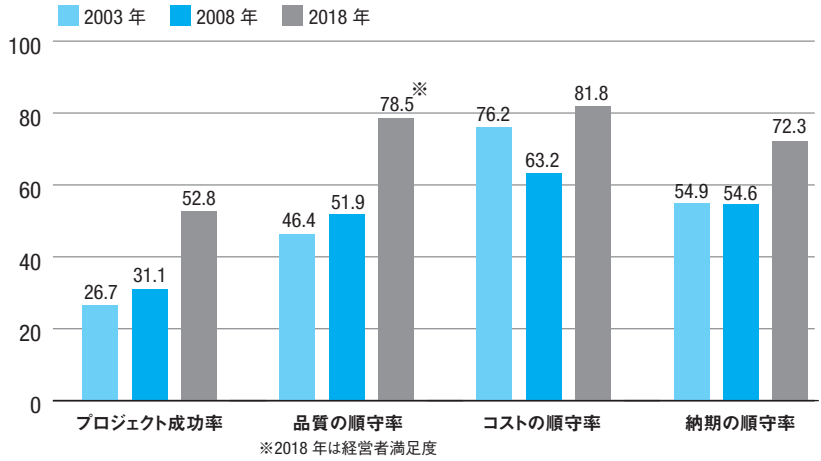

図1-2 プロジェクトの成功率の変化
(数値は『日経ビジネス電子版』「プロジェクト失敗の理由、15年前から変わらず(2018年3月8日掲載)」より引用)

この調査の対象は、一般企業のシステム部門です。直近のプロジェクトのQCD、すなわち「Quallity：品質」「Cost：コスト」「Delivery：納期」について評価してもらった結果をもとにしており、QCDの3要素がすべて順守できた場合を“成功”として成功率を算出しています。2018年は調査対象に業務部門やIT企業も加わり、「品質の自己評価」を「経営者の満足度」に換えて質問していますが、概ね2018年が最も高いという傾向は変わっていません。

(2) プロジェクトの成功率が高くなった理由

成功率が上がった要因は何でしょうか。2008年の調査結果記事によると、成功率が上がった理由の1つが「プロジェクトにおける定量管理の普及」と解説さ

れています。**図1-3**では2003年と2008年の調査データしかありませんが、プロジェクト管理の3要素であるQCDそれぞれで定量管理を導入した割合が増えています。

実は私の会社（株式会社システムインテグレータ。以降SI社と表記）では、2008年11月に統合型プロジェクト管理システム「SI Object Browser PM」を発売開始しています。発売以来13年間さまざまな企業のプロジェクト管理のシステム化に携わってきているのですが、実際、2008年よりも現在の方が、着実に定量管理が定着していると感じています。IT業界は、この15年で着実に定量管理を取り入れて進化しているのです。

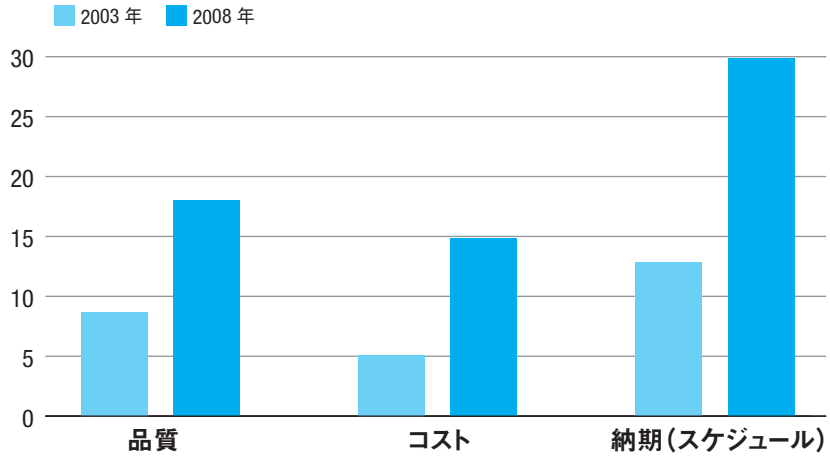

図1-3 定量管理の普及度
（数値は『日経ビジネス電子版』「プロジェクト失敗の理由、15年前から変わらず（2018年3月8日掲載）」より引用）

プロジェクトの規模と成功率の関係

プロジェクトの成功率に関してもう1つ質問です。「プロジェクトの失敗率は、規模が大きいほど高くなるでしょうか」(**図1-4**)

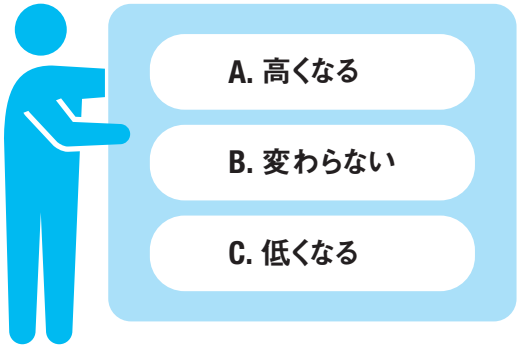

図1-4 プロジェクトの失敗率は規模が大きいほど高くなるか

私の知り合いで「うちの会社はプロジェクトの請負金額が5千万円を超えると失敗が増える」と嘆く経営者がいます。自分の会社に当てはめてみてもこれは言えていて、プロジェクト管理力が上がるにつれて、この上限金額（規模）が高まっていった覚えがあります。

こうした経営者の感覚は、『日経コンピュータ』の「ITプロジェクト実態調査2018」を『日経クロステック』(日経BPの媒体）が発表しているグラフでもはっきり表れています。**図1-5**を見てください。上のグラフは開発期間と成功率の関係を表したものですが、「3か月以上～6か月未満」の失敗率が33.7％なのに対し、「1年半以上～2年未満」になると失敗率65.3％と2倍にも跳ね上がる傾向がわかります。

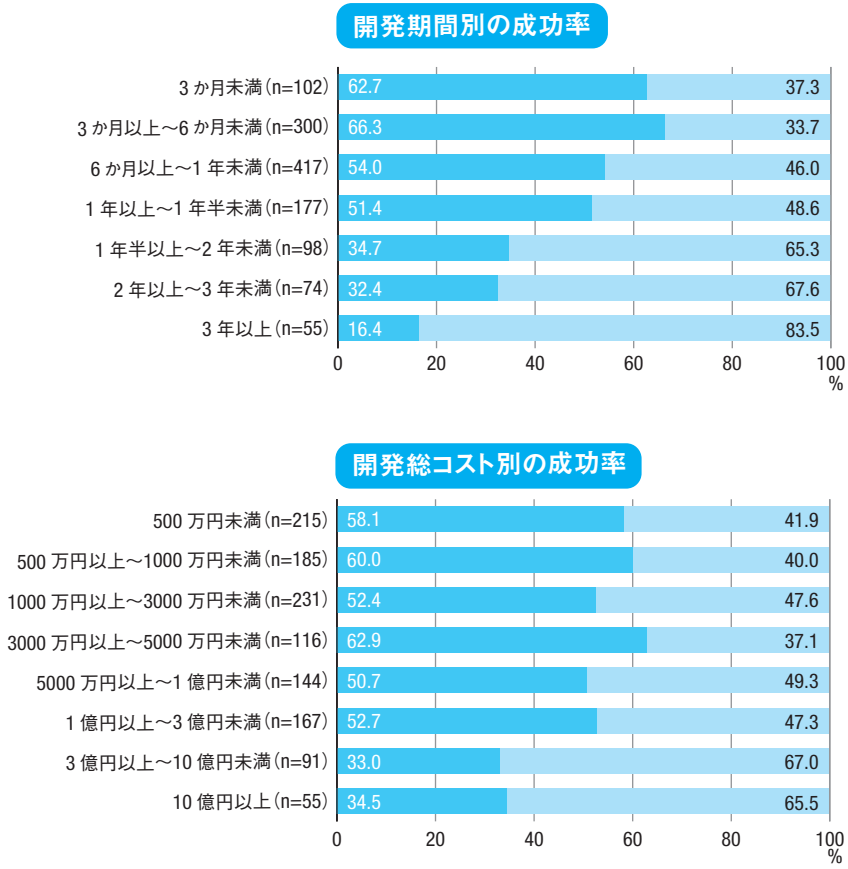

図1-5 プロジェクトの規模と成功率の関係
(出典:日経コンピュータ2018年3月1日号 ITプロジェクト実態調査2018)

下のグラフは期間の代わりに開発総コストを規模の指標に用いたものです。こちらも「1億円以上〜3億円未満」の失敗率47.3％に対して、「3億円以上〜10億円未満」は20ポイントも高い67％に膨れ上がっているのが見て取れます。

ということで正解は「A:高くなる」です。特に1年半を超えるプロジェクト、3億円以上のプロジェクトになると成功率はかなり低くなる。そう心してプロジェクトに向き合う必要がありますね。

最近の状況

図1-2や図1-5は、日経コンピュータがアンケートを実施した2018年発表のデータですが、　日本情報システム・　ユーザー協会（JUAS：Japan Users Association of Information Systems）から最新のデータが発表されていますので見比べてみましょう。

(1) 企業IT動向調査報告書2021

図1-6～図1-8は同協会が公開している「企業IT動向調査報告書2021」にあるプロジェクト規模別の成功率を表したものです。図1-6が工期、図1-7が予算、図1-8が品質とQCD別に調査報告しているので、図1-2の値と比較しやすいと思います。また、プロジェクト規模は100人月未満、100～500人月、500人月以上と工数別に3段階に分かれています。図1-5の下のグラフはコスト別なので、これと比較するためにざっくり1人月100万円とすると、1億円未満、1～5億円、5億円以上という分類になります。

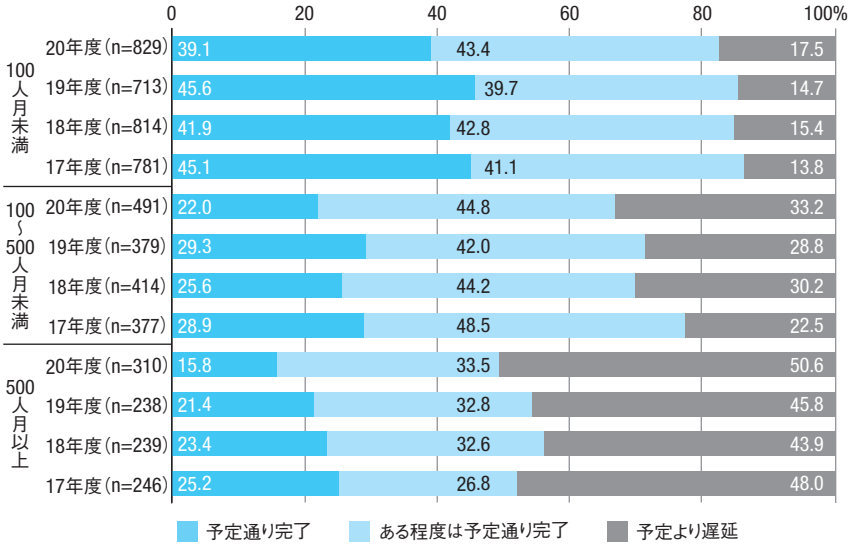

図1-6　プロジェクト規模別システム開発の工期遵守状況　（出典：JUAS企業IT動向調査報告書2021）

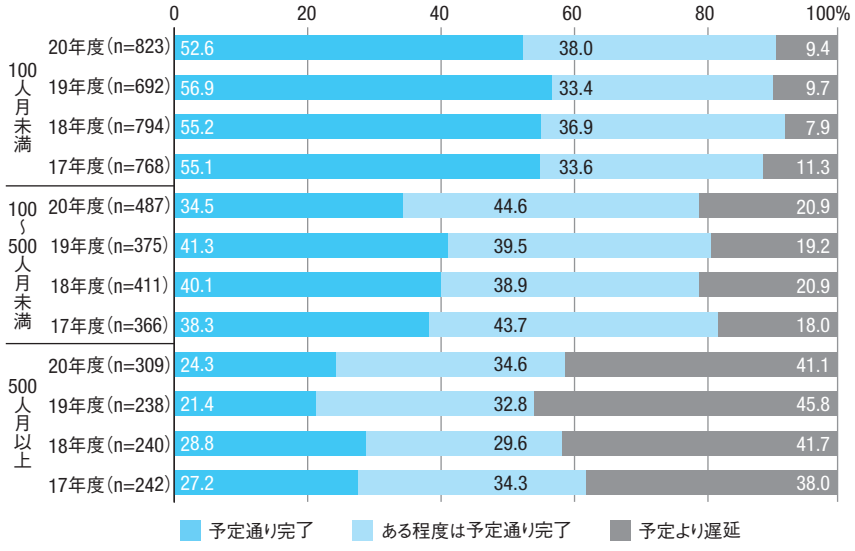

図1-7 プロジェクト規模別システム開発の予算遵守状況 （出典：JUAS企業IT動向調査報告書2021）

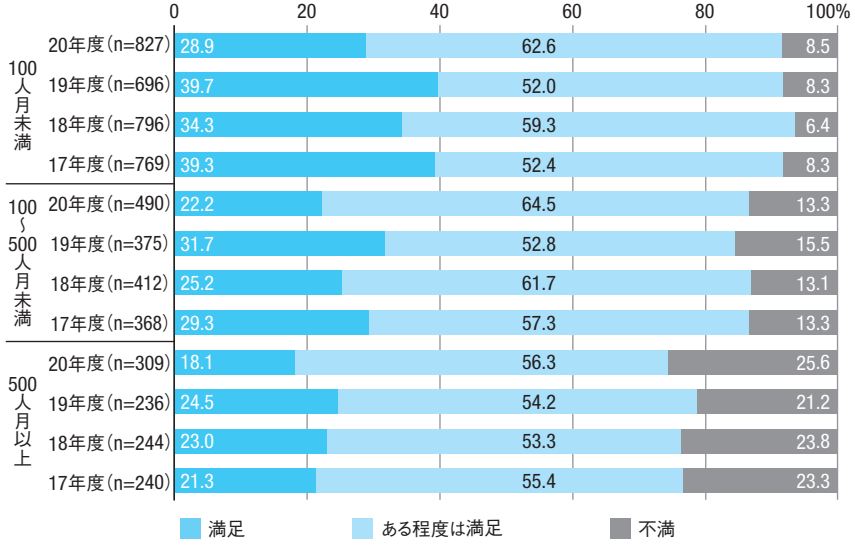

図1-8 プロジェクト規模別システム開発の品質満足度 （出典：JUAS企業IT動向調査報告書2021）

(2) QCD別のプロジェクト成功率

図1-2の2018年の値で見ると、納期の順守率72.3％、コストの順守率81.8％、品質の満足度78.5％となっています。図1-6～図1-8は分類の仕方が違うので正確に比較はできないのですが、2018年度の100～500人月のデータで比較してみましょう。「ある程度」という評価も成功に含めると表1-1のような値となり、QCDともにほぼ同じような成功率となっているのが確認できます。

	日経コンピュータ2018年	JUAS2018年（右2列は内訳）		
		計	予定通り／満足	ある程度
納期の遵守率	72.3%	69.8%	25.6%	44.2%
コストの遵守率	81.8%	79.0%	40.1%	38.9%
品質の遵守率	78.5%	86.9%	25.2%	61.7%

表1-1 日経コンピュータとJUAS調査データの比較（QCD）

(3) 規模別のプロジェクト成功率

図1-5では規模が大きくなるとプロジェクトの成功率が低くなる傾向がはっきり出ていました。これに関してもJUASのデータと比較してみましょう。表1-2は日経コンピュータとJUASの調査結果を比較したものです。JUASのデータは図1-7の予算（工数）における最新2020年の値を用いています。また、日経コンピュータの方は5000万円～1億円、1億円～3億円、3億円～10億円の3つをピックアップしています。

日経コンピュータ2018年		JUAS2020年（右2列は内訳）			
			計	予定通り／満足	ある程度
5000万円～1億円	50.7%	100人月未満	90.6%	52.6%	38.0%
1億円～3億円	52.7%	100～500人月	79.1%	34.5%	44.6%
3億円～10億円	33.0%	500人月以上	58.9%	24.3%	34.6%

表1-2 日経コンピュータとJUAS調査データの比較（規模別）

JUASの調査においても100人月未満で90.6％の成功率だったものが、100～500人月で79.1％、500人月以上で58.9％と規模が大きくなるにつれて成功率が下がる傾向がはっきり出ていますね。ここで日経コンピュータの値に比べてJUASの成功率が高く見えるのは、成功の定義に「予定通り完了」だけでなく「ある程度予定通り完了」も加えているからでしょう。それを抜くとだいぶ近い値になります。納期や品質と違い、コスト（予算順守状況）に関しては「ある程度」は予算順守してないのが明らかで、成功に含めにくいからだと推測されます。

(4) 年度別のプロジェクト成功率

図1-2では、2003年と比べて2018年のプロジェクト成功率が約2倍になっていました。この傾向は現在も続いているのでしょうか。

図1-6の工期順守状況で2017年度～2020年度の4年間の成功率を見てみましょう。100人月未満の規模で「予定通り完了」を比較すると、17年度45.1％、18年度41.9％、19年度45.6％、20年度39.1％とほぼ変わっていない状況になっています。それどころか500人月以上で見ると17年度25.2％だったものが23.4％、21.4％、15.8％とだんだん悪くなっています。図1-2を見て成功率が2倍になったとよろこんだのですが、それでもQCDとも成功するのは5割程度でした。この数年はその程度の成功率で足踏みしている状況が読み取れます。

なお、2020年度の順守度や品質満足度が極端に低くなっていますが、これはコロナ禍の影響が大きいとレポートでは分析しています。

プロジェクト失敗の原因

少し掘り下げてプロジェクト失敗の原因も考察してみましょう。**図1-9**は、同じく日経コンピュータの「ITプロジェクト実態調査2018」を日経クロステックが公開しているグラフです。1745件のプロジェクトのうち計画よりも遅延が発生した513件について原因を複数回答で答えてもらっています。

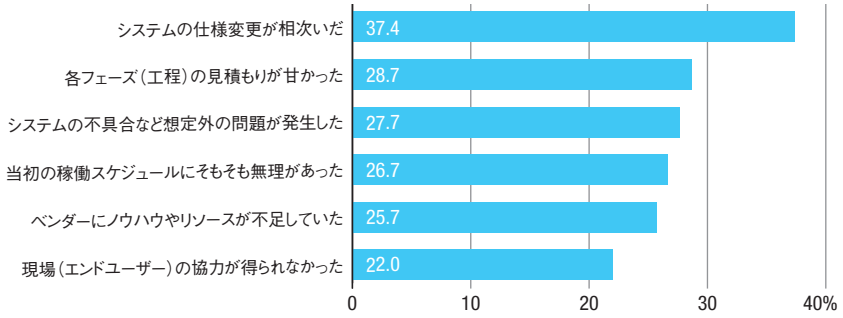

図1-9 スケジュールが遅延した原因
（出典：日経コンピュータ2018年3月1日号 ITプロジェクト実態調査2018）

遅延原因のトップは、「システムの仕様変更が相次いだ」というものです。これは、それ自体が根本的な原因というよりも、例えば「要件定義や基本設計が甘かった」などによって生じたものと考えられます。次点の「各フェーズ（工程）の見積もりが甘かった」はどうでしょうか。こちらは、見積もり基準がない、見積もりがKKD（勘と経験と度胸）で行われている、複数の人で見積もりレビューを行っていない、などの原因が考えられます。

このように納期遅れの直接的原因も、「トヨタのなぜなぜ分析」で掘り下げると本来の原因が見つかります。そのうえで再発防止の対策を行うことが重要になるわけです。

プロジェクト失敗が生じやすい工程

失敗はどの工程で生じやすいのでしょうか。**図1-10**は日経クロステックが公開している「4年前と変わったか？プロジェクト成功率の実態」において、「あなたが関わった案件のなかで、納期遅れやコスト超過が最も発生しやすい工程はどれですか」という問いに対する1288件の回答を、V字モデル中で示したものです。

これによると、要件定義26％、設計23％、製造27％、テスト24％とどの工程でもまんべんなく発生しており、記事では「“地雷”は全工程に」と表現しています。ただし、これは、この設問が「最も発生しやすい工程」を尋ねているからであり、それらの元となる要因が上流工程にあることも多いと思います。

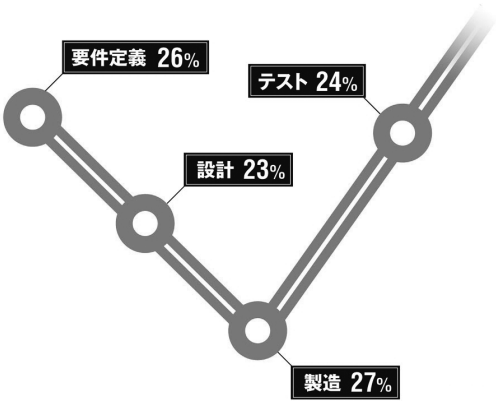

図1-10 納期遅れやコスト超過が発生しやすい工程（n=1288）
（出典：日経コンピュータ2014年10月16日号）

COLUMN

V字モデルとは

V字モデル（V Model）とは、ウォーターフォール型のソフトウェア開発における「開発工程」と「テスト工程」の対応関係をV字上に表したものです。ウォーターフォールモデルの各工程は図1-11のような対応関係になっており、右側のテスト工程がそれぞれ左側のどの開発工程の内容を検証しているかをV字で表したものです。例えば「詳細設計書」に書かれている内容が正しく動作するかを「単体テスト」で検証するという関係がパっとわかります。

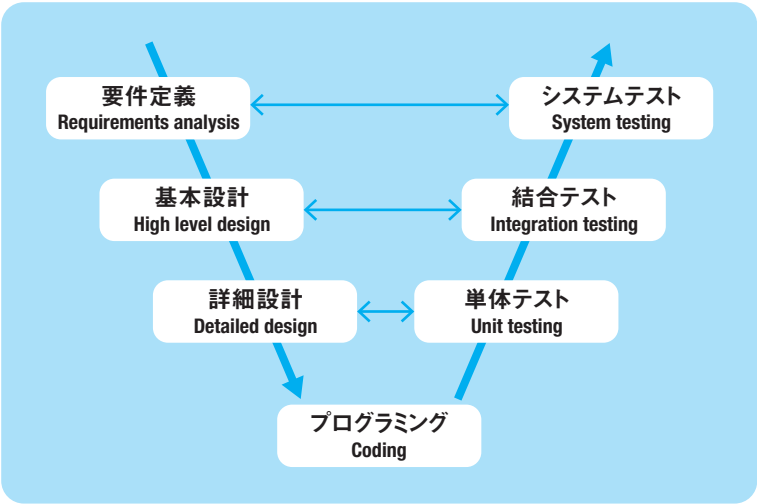

図1-11 V字モデル

プロジェクト失敗を防ぐには

図1-9のようなトラブルを起こさないために、各社ともいろいろ工夫して対策を行っています。私の会社でも何度も“痛い目”に遭いながら、その都度対策を強化したり改善したりしています。参考までにSI社が行っている対策を表1-3に紹介します。地味ですがこういうことも1つのノウハウなので、対策内容に関してここで説明しましょう。

スケジュールが遅延した原因	SI社が行っている対策
システムの仕様変更が相次いだ	各工程の最後に「工程QA（後述）」のマイルストーンを設け、上流工程の遅延を下流に持ち越さない
各フェーズ（工程）の見積もりが甘かった	総コスト1億円以上の案件は、見積もり提出時に「見積もり審査会」を実施
システムの不具合など想定外の問題が発生した	「リスク標準」を持ち、プロジェクトスタート時にリスクの洗い出しを行う
当初の稼働スケジュールにそもそも無理があった	「見積もり審査会」のチェック項目に設定して、リスクをチェックしている
ベンダーにノウハウやリソースが不足していた	
現場（エンドユーザー）の協力が得られなかった	

表1-3 SI社で行っているトラブル防止対策

(1) 工程QA

製造業ではクオリティゲートという品質管理手法が取られています。これは製品が完成するまでの各段階で、ゲートを設けて品質を審査し、各工程で不具合を後工程に流さないという考え方です。

システム開発においてもこの考え方は有効です。ウォーターフォールであれば「要件定義」「基本設計」「詳細設計」などの各工程の最後に「品質レビュー」というゲートをマイルストーンとして置き、次の工程に移行して良いかどうか審査します。スケジュール遅延原因トップの「システムの仕様変更が相次いだ」というトラブルを招く主な要因は、上流工程の不備のしわ寄せですので、これを防ぐ方

法としてクオリティゲートの手法を取り入れているのです。

具体例で説明しましょう。**図1-12**はある実際のプロジェクトのガントチャートです。一番上のラインを見ると、要件定義、基本設計、詳細設計など各工程の最後にQA（Quality Assurance）というマイルストーンを設け、「自工程の品質保証」のための審査を行っている様子がわかります。

図1-12 各工程で設置されるクオリティゲート（マイルストーン）

COLUMN

QAとQC

QAはQuality Assuranceの略で、日本語では「品質保証」です。工程ごとにQAを設けるということは、“各工程のアウトプットが問題ないことを保証する”という意味になります。一方、QCはQuality Control、すなわち「品質管理」です。こちらは“各工程の品質を保証できるレベルに高める管理作業”という意味です。昔はQCという言葉がよく使われていましたが、最近ではQCとQAをきちんと使い分けて用いるようになっています。

通常、各工程のクオリティゲートでチェックする項目は標準化されています。SI社の例を見てみましょう。「詳細設計QA」というマイルストーンをクリックすると**図1-13**のような画面が表示されます。品質基準名「⑤QA（詳細設計）」というのが品質のチェック項目で、このマイルストーンに紐づいています。

図1-13 マイルストーンと品質基準の連携

「⑤QA（詳細設計）」という品質基準名をクリックすると、**図1-14**のような品質審査のチェック項目が表示されます。QAレビューはこの品質基準判定画面を使って行われ、レビュー日時、参加者、総合評価、コメントなどを記載してレビュー実績を登録し、これが品質保証のエビデンスとなります。下段の表部分に詳細設計QAにおける標準チェック項目が表示されるので、各項目で合否を確認したうえで次工程に進めるかどうかを判定します。

図1-14 品質審査のレビュー実績

(2) 見積もり審査会

「そもそも見積もりが甘かった」「初めからスケジュールに無理があった」「ノウハウやリソースが不足していた」「現場（エンドユーザー）の協力が得られなかった」という原因は、どれもプロジェクト失敗あるあるですね。こうしたリスクに気づかずにプロジェクトを進めて、後で取り返しできない状態になるのは絶対に防ぎたいものです。そのためにSI社が行っているのが「見積もり審査会」です。

これは「総コスト1億円以上の案件を対象に、見積もり提出前に審査を行う」というものです。PM（Project Manager）や部門長はもちろん、社長やPMO（次ページコラム参照）も参加して「見積もりを出して良いか」を審査しています。以前は大規模案件の受注時に審査していたのですが、見積もりを提示した後で「やっぱりやめます」とはなかなかできないので、見積もり提出前に行うルールに変えました。総コスト1億円以上としているのも、図1-5の開発コスト別の成功率を見るとそれなりに妥当性があるように思います。

COLUMN

PMO

PMOはProject Management Officeの略で、企業におけるプロジェクト管理を組織横断的に行う専門部隊です。昔は品質管理室というポジションはあっても、PMOという役割は明確ではありませんでした。プロジェクト管理の重要性が高まるにつれて、PMOを置こうという企業も増えていますが、コスト部門なのでPMOを設置していないソフトウェア会社もまだまだ多いようです。しかし開発者が50名以上いるような企業規模でしたら、PMOを設けてプロジェクト成功率を高める方がメリットが大きいと思います。

通常、PMOは全社レベルで設置しますが、SI社では全社PMOとは別に各事業部ごとにPMOを設けています。PMOの役割は企業によって違ったりもするのですが、SI社の場合は次の3つをPMOの役割として定義しています。

① リスクの高いプロジェクトを早期発見して対処させる
② プロジェクト管理が甘いプロジェクトを見つけて管理を徹底させる
③ 社員や組織のプロジェクト管理力を向上させ、標準化を推進する

図1-15は見積もり審査会のチェック項目です。あらかじめ見積もり担当者がPMと相談しながら表を埋めたうえで審査会に臨み、そこでさまざまな観点から意見や指摘が入ります。指摘をクリアするために持ち帰って再検討することや、場合によっては見積もり提出を見送るという判断に至ることもあります。

チェック分類	項番	チェック項目	評価	対応状況
1.体制	1	開発実施時にPM,PLは確保できる見込みか、スキルは問題ないか ・PM,PLは誰を予定しているか。		PM： PL： スキル面での懸念点：
	2	開発実施時に必要となる要員は確保できる見込みか、スキルは問題ないか ・開発実施時に必要な要員は何人で、そのうち何%確保できそうか（カバー率）		必要となる要員： 確保できそうな要員：　プロパー　人、協力会社　人（カバー率：%） スキル面での懸念点：
	3	一括発注はあるか。ある場合は、会社名を記載。 一括発注先は該当する業務の開発経験があるか、任せられる体制を組めるか（管理を含む） ・開発経験が浅い場合は、その分のリスクをカバーする策は何か		会社名： 開発経験： リスクのカバー策：
	4	当社が請け元として、他ののベンダーを統括するか。統括する場合、リスクヘッジはどのように考えているか。		リスクヘッジ：
	5	その他、体制面での懸念点は何か。		
2.スケジュール	6	・各工程のスケジュールが無理のある形で重なっていないか ・期間的にタイトな工程はどことどこで、きついスケジュールをキープするためにどのように対応する予定か。		工程名： 対応策：
	7	開発を進行する上で、特殊なスケジュールとなっていないか。 ・先方都合により、3ヶ月以上、開発が止まる期間があるなど		
	8	その他、スケジュール面での懸念点は何か。		
3.コスト	9	利益率（粗利）は30%以上を確保できるか。		予定粗利率：
	10	コストが膨れ上がるとしたら、どのような要因が考えられるか。 膨れ上がらないために、どのようなことをする予定か。		要因： 対策：
	11	必要経費は盛り込まれているか。		
	12	その他、コスト面での懸念点は何か。		
4.技術	13	経験やノウハウのない技術や他システム、ツールを使う予定があるか ある場合、どのようにして対処する予定か。		使用予定： 対処：

図1-15　見積もり審査会のチェック項目（一部）

少し内容を見てみましょう。チェック分類「1.体制」や「4.技術」には、図1-6の「ベンダーにノウハウやリソースが不足していた」というリスクを確認する内容も含まれています。同様に「2.スケジュール」欄では「当初の稼働スケジュールにそもそも無理があった」というリスクをチェックし、「3.コスト」欄は「各フェーズ（工程）の見積もりが甘かった」というリスクなどを防止するチェック内容が入っています。

図1-15では切れていますが、下の方には「6.ユーザー」欄があって「現場（エンドユーザー）の協力が得られなかった」というリスクについて確認しています。最後のチェック項目に「本PJで最もリスクと考えられることは何と何か」という直球の設問があり、これが意外と「システムの不具合など想定外の問題が発生した」というリスク要因をヘッジできたりしています。

(3) リスク標準

見積もり審査会は、見積もり提出前に単発で行うものです。一方、リスク管理としてはプロジェクトがスタートしたときにリスクを洗い出して、それらを減らしたり回避したりするための対策とモニタリングを行う必要があります。

こうしたプロジェクトリスクは、毎回イチから考えていたら漏れが生じるので、標準化しておくことが重要です。SI社でも図1-16のようなリスク標準を持って

います。リスク番号をクリックすると、具体的なリスク内容が定義されています。このようなリスク要因がプロジェクトの初期値としてセットされるので、当該プロジェクトに合うように編集してリスクをメンバーで共有してコントロールしています。

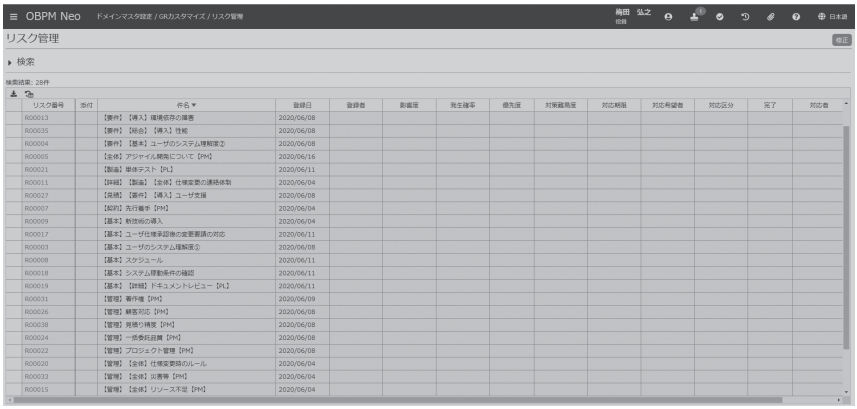

図1-16 プロジェクトリスク一覧

おわりに

本章では、プロジェクトの成功に向けて、特に定量管理とリスク管理を中心にお話しました。私は10数年前に「IT業界の近代化」を自分のライフテーマに掲げて取り組んできたのですが、このようにシステム開発の進歩をお伝えできるのは嬉しいことです。しかし、一方で2018年のデータを以ってしても「まだ半分が失敗している」というのは事実ですし、コロナ禍の影響もあって最近は足踏みしています。プロジェクトの失敗は「プロジェクト管理の強化」と「良い開発ドキュメント」の両輪がうまくまわって初めて防げます。次章からは、失敗しないための「開発ドキュメント」について一緒に考えて行きましょう。

続きをご覧になりたい方は「インプレスブックス」へ

シンクイット
Think IT ™

thinkit.co.jp

エンジニアのための
オープンソース実践活用メディア

“オープンソース技術の実践活用メディア”をスローガンに、インプレスグループが運営するエンジニアのための技術解説サイト。開発の現場で役立つノウハウ記事を毎日公開しています。

2004 年の開設当初からOSS（オープンソースソフトウェア）に着目、近年は特にクラウドを取り巻く技術動向に注力し、ビジネスシーンで OSS を有効活用するための情報発信を続けています。OSS に特化したビジネスセミナーの開催や、Web 連載記事の書籍化など、Web サイトにとどまらない統合的なメディア展開に挑戦しています。また、エンジニアを含むクリエイターの独立・起業、フリーランスなどの多様化する「働き方」や「IT で社会課題を解決する」等をテーマに、世の中のさまざまな取り組みにも注目し、解説記事や取材記事も積極的に公開しています。

- 本書は、インプレスが運営するWeb メディア「Think IT」に掲載された記事を再編集したものです。
- 本書の内容は、執筆時点までの情報を基に執筆されています。紹介した Web サイトやアプリケーション、サービスは変更される可能性があります。
- 本書の内容によって生じる、直接または関節被害について、著者ならびに弊社では、一切の責任を負いかねます。
- 本書中の会社名、製品名、サービス名などは、一般に各社の登録商標、または商標です。なお、本書では ©、®、™ は明記していません。

ご利用のお客様へ

このたびは弊社電子書籍ダイジェスト版をご利用いただきまして誠にありがとうございます。電子書籍ダイジェスト版のPDFファイル（以下「本PDFファイル」）の取り扱いに関し、以下のとおりご案内いたします。

●本PDFファイルの収録コンテンツ

本PDFファイルに収録されたコンテンツ（情報・資料・画像等）（以下「本コンテンツ」）は、無償または有償で、株式会社インプレス（以下「当社」）が認めた方法に従ってのみご利用いただけます。本コンテンツは、利用者様ご本人の個人的な使用の目的でのみ利用することができるものとし、当社の事前の書面による承諾なく、企業内、店舗、サイトなどにおいて特定または不特定の多数に利用させることのほか、著作権法で認められている私的使用の範囲を超えて複製、貸与、公衆送信その他の利用をすることはできません。

●ご利用方法

本PDFファイルは、ダウンロードを行われた利用者様ご本人のみがご利用いただけます。企業内での複数名による本PDFファイルのご利用については、別途有料サービスとしてご提供させていただきます。詳しくは当社までお問い合わせください。

●著作権

本コンテンツの著作権は、当社又は当該コンテンツの著作権者に帰属し、許可なく複製、転用、販売、蓄積等著作権法で認められている私的使用の範囲を超えて利用することはできません。また、本コンテンツの内容を変形、変更、加筆、修正等することは一切できません。

●商標など

本コンテンツに含まれる商標、ロゴ等は、当社または当該商標、ロゴ等の商標権者の商標です。本コンテンツには、TMマークまたは ® マークは明記していません。これらを私的使用以外の目的で無断に利用することはできません。

●免責事項

当社は、本コンテンツの内容について、妥当性や正確性について保証せず、一切の責任を負いません。また、本コンテンツの利用にあたり生じたいかなる損害についても、当社は一切の責任を負いません。本コンテンツをご覧いただくためのアプリケーション等のインストールに必要な接続等の費用は、利用者の自己負担で行うものとします。本コンテンツやURLは、予告なく変更または中止されることがあります。当社は、本コンテンツの変更、追加、中断または終了によって生じるいかなる損害についても責任を負いません。

impress

株式会社インプレス

法人営業局
customer@impressbm.co.jp
〒101-0051　東京都千代田区神田神保町1-105　神保町三井ビルディング