

IT 資格道場

CompTIA Project+ 学習用 テキスト

予備知識ゼロから合格ラインへ —— 読んで理解する1冊目

無料サンプル(第1章まで)

CompTIA Project+ (PK0-005) 公式 Exam Objectives 準拠

2026年10月版

ご利用にあたって

- このテキストは「IT資格道場」(www.shikaku-doujou.com)の購入者限定テキストのうち、学習用テキストの第1章までを収録した**無料サンプル**です。どなたでもご覧いただけます。
- 無料サンプルも著作物です。転載・改変しての再配布・二次販売はできません。
- AI モデルの学習データとしてのご利用(学習目的の入力・データセット化を含む)はできません。
- 製品版(学習用テキスト・頻出度別ノート・暗記用ノートの3点)は、ご購入者を識別できる透かし入りのPDFとして提供されます。
- 本PDFはスマートフォン・タブレットでの閲覧に合わせたA5版面・文字サイズで組版しています。

目次

試験の骨組みと、このテキストの読み方 [0]

試験の基本情報

4つの分野と配点

読み方

第1章 Project Management Concepts

[1.1/1.2/1.3/1.4/1.5/1.6/1.7/1.8/1.9/1.10/1.11]

プロジェクトの特性と手法・フレームワーク [1.1]

アジャイルとウォーターフォールの比較 [1.2]

変更管理プロセス [1.3]

リスクマネジメント [1.4]

課題管理 [1.5]

スケジュールの作成と管理 [1.6]

品質管理とパフォーマンス管理 [1.7]

コミュニケーション管理 [1.8]

会議運営 [1.9]

チームと資源の管理 [1.10]

調達とベンダー選定 [1.11]

サンプルはここまでです

試験の骨組みと、このテキストの読み方 [0]

試験の基本情報

CompTIA Project+ は、プロジェクトマネジメントの基礎を問う、特定の手法や製品に偏らない入門の資格です。試験コードは PK0-005 (V5)。

項目	内容
試験コード	PK0-005 (V5)
問題数	最大 90 問
出題形式	選択式 (択一・複数選択) とパフォーマンスベース問題 (PBQ)
試験時間	90 分
合格点	100～900 のスケールスコアで 710

パフォーマンスベース問題は、対応付けや並べ替えのように、操作や判断を実際の作業に近い形で問う問題です。時間を取られやすい。迷ったら後回しにして、先に選択式を片付ける進め方が向きます。合格点の 710 は、難易度で調整されたスケールスコア。正答数そのものではありません。

試験の版や形式、受験料は変わることがあります。申し込みの前に、公式の試験ページで最新の情報を確認してください。

4 つの分野と配点

分野	配点	対応する章
1.0 Project Management Concepts	33%	第 1 章

分野	配点	対応する章
2.0 Project Life Cycle Phases	30%	第 2 章
3.0 Tools and Documentation	19%	第 3 章
4.0 Basics of IT and Governance	18%	第 4 章

学習の時間も、この配点に合わせて配るのが合理的だ。分野 1 と分野 2 で全体の 6 割を超えるので、まずこの 2 つを固めてください。

読み方

各章の見出しの末尾にある [1.1] のような記号は、公式の出題範囲の細目の番号です。見出しの語は、試験で目にする英語の用語を併記した。計算の問題は、本文の例を自分の手で解き直すと定着します。

第1章 Project Management Concepts

[1.1/1.2/1.3/1.4/1.5/1.6/1.7/1.8/1.9/1.10/1.11]

CompTIA Project+ (PK0-005) で最も配点が大きい分野で、全体の 33% を占めます。試験は選択式とパフォーマンスベースの問題で構成されます[^c1-1]。範囲は手法の選択、変更と課題の扱い、見積りの計算、契約の種類の見分けなどに及びます。用語を覚えるだけでなく、場面を読んで「次に何をするか」を言えることが目標です。

プロジェクトの特性と手法・フレームワーク [1.1]

1.1-1 プロジェクトの特性とプログラム・ポートフォリオ

プロジェクトは、独自の成果物を作るために期間を区切って行う取り組みです。CompTIA の出題範囲は特性として、開始と終了・独自性・理由と目的の 3 つに加えて、プログラムの一部であること・ポートフォリオの一部であることを挙げています。前の 3 つは次のとおり。

- **開始と終了がある**: 終わりの日付や完了条件が決まっており、いつまでも続く仕事ではありません。終わりのない仕事は運用 (オペレーション) です。
- **独自性 (unique)**: 前回と全く同じ成果物にはなりません。毎月の給与計算は繰り返しなので運用です。
- **理由と目的がある**: 事業上の狙いがあるため始まります。

プログラムは、関連する複数のプロジェクトを束ねて管理し、個別に管理しては得られない効果を狙う単位です。**ポートフォリオ**は、プロジェクトやプログラムを組織の戦略目標に照らして選び、まとめて管理する単位です。階層は「ポートフォリオ ⊃ プログラム ⊃ プロジェクト」で、逆の包含ではありません。

隣との違いは範囲です。新しい基幹システムの導入(プロジェクト)、それを含む全社の業務刷新(プログラム)、会社が抱える全投資案件の一覧(ポートフォリオ)、と並べると区別できます。間違えやすいのは運用との区別で、保守や日次バッチのように期限なく繰り返す仕事は、開始と終了がないのでプロジェクトではありません。

1.1-2 予測型の手法:ウォーターフォール・SDLC・PRINCE2

ウォーターフォールは、要件定義、設計、実装、テスト、導入と、工程を順に流す予測型(計画駆動)の手法です。前の工程が終わってから次へ進み、全体の計画を最初に固めます。

SDLC (Software Development Life Cycle) は、ソフトウェアが生まれて退役するまでの段階を表す枠です。ウォーターフォールはその代表的な進め方の1つで、SDLCの段階をアジャイルで反復して回すこともできます。SDLCは進め方の名前ではありません。段階の枠の名前です。

PRINCE2 は PProjects IN Controlled Environments の略で、構造化されたプロジェクトマネジメントの手法です^[^c1-2]。特徴は、プロジェクトをステージに区切り、ステージの境目で続行の可否を判断する点と、許容範囲を超えそうな時だけ上位へ報告する例外管理(manage by exception)です。ビジネス上の正当性を続けて確かめる原則もあります。

予測型の共通点は、スコープ・スケジュール・予算を計画で先に決め、変更は変更管理で扱うことです。間違えやすいのは、PRINCE2を「アジャイルの一種」とする選択肢です。PRINCE2はステージと文書で統制する予測型の枠で、アジャイルの手法ではありません。

1.1-3 アジャイル系の手法:スクラム・カンバン・XP・SAFe

アジャイルの出発点は 4 つの価値です。「プロセスやツールよりも個人と対話」「包括的なドキュメントよりも動くソフトウェア」「契約交渉よりも顧客との協調」「計画に従うことよりも変化への対応」を、右側より左側に重く置きます[^c1-3]。

スクラムは、固定長のスプリント(1 か月以内)で作業を回す枠組みです。スクラムチームは、プロダクトオーナー 1 人、スクラムマスター 1 人、開発者で構成され、通常 10 人以下です[^c1-4]。プロダクトバックログを順に並べ、スプリントごとに使える増分(インクリメント)を作ります。

カンバンは、作業の流れをボードで見える化し、仕掛かり(WIP)に上限を置く管理手法です。固定長の反復も固定の役割も要りません。新しい作業は余力が空いた時に引く(プル)形で始めます[^c1-5]。

XP(エクストリーム・プログラミング)は、ペアプログラミング、テスト駆動開発、継続的インテグレーション、短い間隔のリリース、リファクタリングなど、開発の技術的な作法を重視する手法です。

SAFe(Scaled Agile Framework)は、多数のチームにアジャイルを拡大して適用する枠組みです[^c1-6]。複数チームが同じ周期で計画を合わせる PI プランニングなどを持ちます。

覚え分けは、役割と反復があるのがスクラム、流れと WIP 上限がカンバン、技術の作法が XP、大規模化が SAFe です。「スプリントがある」と言われたらスクラムで、カンバンにスプリントはありません。

スクラムの定める要素を表にまとめます。

区分	内容
イベント	スプリント計画、デイリースクラム(15分)、スプリントレビュー、スプリントレトロスペクティブ
作成物	プロダクトバックログ、スプリントバックログ、インクリメント
役割	プロダクトオーナー、スクラムマスター、開発者
品質の基準	完成の定義(Definition of Done) : インクリメントが満たすべき品質の状態を正式に書いたもの

見分けの表です。

手法	型	単位	特徴の一言
ウォーターフォール	予測型	工程(フェーズ)	前の工程の完了が次の条件
PRINCE2	予測型	ステージ	境目で続行を判断し、例外だけ上へ報告
スクラム	アジャイル	スプリント(1か月以内)	役割と4つのイベントがある
カンバン	アジャイル	作業項目の流れ	固定の反復なし、WIPの上限
XP	アジャイル	短い反復	ペアプログラミングやテスト駆動などの技術の作法

手法	型	単位	特徴の一言
SAFe	アジャイルの拡大	複数チームの周期	多数のチームの計画をそろえる

場面で選ぶ例です。保守チームに、障害の修正依頼が不定期に次々と届き、固定の反復では優先順位が追いつかない状況なら、カンバンが向きます。新製品を、顧客のフィードバックを受けながら2週間ごとに出していくなら、スクラムで、カンバンの連続的な流れではありません。数十のチームが1つの大きな製品に取り組むなら、SAFeを検討します。要件が契約で固定され、工程ごとの承認が義務づけられた基幹システムの更改なら、予測型です。

1.1-4 DevOps と DevSecOps

DevOps は、開発 (Dev) と運用 (Ops) が分断せず協力し、ビルド、テスト、リリースを自動化して、変更を速く安全に届ける考え方と実践です。**DevSecOps** は、そこにセキュリティ (Sec) を最初から組み込み、セキュリティの検査を開発の流れの各段階に置く形です。セキュリティの検査は後半の監査に回さず、コードを書く段階から脆弱性の検査を走らせるので、後付けではありません。

どちらも自動化されたパイプライン (CI/CD) と深く結びつきます。CI/CD の中身は 4.5 で扱うので、ここでは詳しく述べません。ウォーターフォールのように「開発が終わってから運用へ渡す」引き継ぎを壊すのが DevOps の狙いです。間違えやすいのは DevOps を「役職」や「ツール名」とする選択肢で、文化と実践の名前です。

アジャイルとウォーターフォールの比較 [1.2]

1.2-1 手法を選ぶ基準:変更への許容度(要求・予算・スケジュール)

最初の基準は、変更をどこまで受け入れられるかです。要求、予算、スケジュールの3つについて見ます。

観点	予測型(ウォーターフォール)	アジャイル
要求	開始時に固定できる	進めながら変わる前提
予算	総額を先に承認する	反復ごとの投資で調整する
スケジュール	最終期日を先に決める	期日か範囲のどちらかを動かす

要求が明確で変わりにくく、期日と金額が契約で固定されているなら、予測型を選んで構いません。要求がはっきりせず、顧客のフィードバックで中身が変わる見込みなら、アジャイルが向きます。

間違えやすいのは「アジャイルは計画しない」という誤解です。アジャイルも計画しますが、計画を短い周期で作リ直します。また予算が厳しい場面でも、要求が曖昧ならアジャイルです。変更の許容度が選択の第一の軸で、予算の大小だけでは決まりません。

1.2-2 手法を選ぶ基準:環境要因(文化・開発・業界標準)

2つ目の基準は環境要因で、出題範囲は文化、開発、業界標準の3つを挙げています。

文化: 組織が権限委譲、自己組織化、頻繁な対話を受け入れるかどうかです。上意下達で承認を重ねる文化にアジャイルを持ち込むと、チームが決められずに止まります。

開発の事情: 成果物の性質です。ハードウェアや建設のように、後から作り直すと費用が大きい物は、アジャイルの小刻みな作り直しに向きません。ソフトウェアのように、小さく作って確かめられる物はアジャイルに傾きます。

業界標準: 規制や標準が、文書、承認、監査の証跡を決めている業界です。医療機器や金融のように規制が重い分野では、予測型の文書と承認の統制が求められます。アジャイルを使う場合も、規制に合う文書を成果物に含めません。

場面の問では、文化や規制の記述が決め手で、手法の名前は手がかりになりません。「規制当局への文書提出が必須」と書かれていれば、予測型か、予測型の統制を持つ混合型が答えです。

1.2-3 チーム構成の違い

観点	ウォーターフォール	アジャイル (スクラム)
プロダクトの責任者	スポンサーや顧客の窓口が要件を承認	プロダクトオーナーがバックログの順序と価値に責任を持つ
役割と責任	職能ごとに分かれ、引き継ぐ	多職能のチームで自己管理する
チームの規模	大きくなりやすい	小さい (スクラムは通常 10 人以下)
資源の割り当て	工程ごとに人が入れ替わる	チームが専念し、同じ人が反復を続ける

プロダクトオーナーは、チームが作る物の価値を最大にする責任者です。バックログの内容と順序を決めます[^c1-4]。プロジェクト・マネージャーは予測型の中心で、計画、予算、スケジュールの責任を負います。スクラムにプロジェク

ト・マネージャーという役割はありません。開発者を含むスクラムチームは多職能で、誰が何をいつどう進めるかを自分たちで決めます(自己管理)。

資源の「専念」は、アジャイルの特徴です。複数案件を兼任させると反復の速度(ベロシティ)が安定しません。兼任が多い職能型の組織ではアジャイルを回しにくくなります。

1.2-4 コミュニケーションの違い

予測型では、書面の計画、状況報告書、変更要求などの文書が中心で、報告は定期的に上位へ流れます。引き継ぎも文書で行います。

アジャイルでは、対面または同じ画面での対話が中心です。毎日のスタンドアップ、スプリントレビュー、振り返り(レトロスペクティブ)が決まった頻度で開かれ、タスクボードやバーンダウンチャートで状況が誰にでも見える形が基本で、状況が一部の人だけに止まることはありません。顧客も反復ごとに動く成果物を見てフィードバックを返します。

観点	予測型	アジャイル
主な手段	文書・報告書	対話・見える化
頻度	計画された定期報告	毎日・反復ごと
顧客との接点	節目(マイルストーン)	継続的

間違えやすいのは「アジャイルは文書を一切作らない」です。必要な文書は作りますが、価値の最小限に絞ります。

場面で選ぶ例を3つ並べます。

場面	選ぶ手法	決め手
規制当局が、設計書と試験記録を工程ごとに承認する医療機器のソフトウェア	予測型	業界標準が文書と承認を求める
利用者の反応を見ながら機能を決めたいスマートフォンのアプリ	アジャイル	要求の変わる前提と、短い周期でのフィードバック
基盤は固定だが、画面は顧客の要望で変わるシステム	混合（基盤は予測型、画面はアジャイル）	部分ごとに変更の許容度が違う

予算、スケジュール、要求の3つのうち、どれが固定でどれが動かせるかを先に書き出すと、手法が決まり、迷う余地は多くありません。3つとも固定なら予測型で、要求だけが動くならアジャイルです。固定された予算と期日の中で要求が変わるなら、範囲の優先順位を付けて、入らない物は次に回す運営になります。

変更管理プロセス [1.3]

1.3-1 変更要求の作成・受付と変更管理ログへの記録

変更管理は、承認されたベースライン（スコープ、スケジュール、予算）への変更を、統制した手順で扱う仕組みです。誰でも思いつきで変えてよいわけではありません。手順を通した変更だけが計画に入ります。例外は設けません。

出題範囲が挙げる流れの最初の2つは次のとおりです。

- 1. 変更要求を作成または受け付ける:** スポンサー、顧客、チーム、ベンダーなど誰でも出せます。
- 2. 変更管理ログに記録する:** 要求を受けたら、その場でID、日付、提出者、内容を記録します。

ここで大事ななのは、受け付けた段階で作業を始めないことです。依頼が口頭やメールで届いても、記録の前に作業へ入ってはいけません。記録のない変更は追跡できません。間違いやすいのは、小さな変更だからと記録を省く選択肢です。小さな変更の積み重ねがスコープクリープになります。

1.3-2 予備レビューと影響評価・推奨の記録

記録の次は、**予備レビュー**です。変更要求が完全か、範囲内か、重複や却下すべきものではないかを、まず簡単に確かめます。ここで不備があれば、影響評価へは進めません。提出者へ差し戻します。

予備レビューを通った要求は、**影響評価**に進みます。変更がスコープ、スケジュール、予算、品質、リスク、資源にどう響くかを見積もります。評価の結果、**推奨**(承認、却下、保留、修正して再提出)を添えて文書に残します。

流れは「記録 → 予備レビュー → 影響評価 → 推奨を記録」の順で、途中を飛ばしてはいけません。影響評価は承認の前に行い、承認の後に回してはいけません。承認してから影響を調べるのは順序の誤りです。間違いやすいのは、推奨を決める者と承認する者を混同することです。推奨は評価した側の意見で、承認は意思決定者の仕事です。

1.3-3 意思決定者の決定と変更管理委員会(CCB)へのエスカレーション

推奨が出たら、誰が決めるかを定めます。**意思決定者**は変更の大きさで決まり、いつも同じ人とは限りません。小さな変更ならプロジェクト・マネージャー、大きな変更ならスポンサーです。

変更管理委員会 (CCB: Change Control Board) は、変更を審査して承認または却下する権限を持つ会議体です。スポンサー、顧客、主要な職能の代表などで構成されます。出題範囲の表現は「該当する場合は CCB へエスカレーションする」です。つまり全ての変更が CCB に行くわけではありません。計画で

決めた権限の範囲を超える変更、たとえば予算や期日の閾値を超える、契約に関わる、といった変更だけが CCB に上がります。

間違えやすいのは、PM が自分の権限を超える変更を独断で承認する選択肢です。権限を超えるならエスカレーションです。

1.3-4 承認後の流れ

承認された後の手順は次の順です。出題範囲の並びと同じで、順序は入れ替えられません。

1. **承認の状況を変更管理ログに記録する** (承認、却下、保留のどれか)
2. **変更の状況を関係者に伝える** (提出者とステークホルダー)
3. **プロジェクト計画を更新する** (スコープ、スケジュール、予算、リスク登録簿など。ベースラインも改訂)
4. **変更を実施する**
5. **実施を検証する** (変更が意図どおり入ったかを確認する)
6. **展開を伝える** (変更が実際に反映されたことを周知する)

場面の問で多いのは、並べ替えです。「計画の更新」は実施の前、「検証」は実施の後、「展開の周知」は最後です。却下の場合も、ログへの記録と提出者への連絡は行います。間違えやすいのは、承認の直後に実施へ進み、計画の更新を後回しにする順序です。更新前に実施すると、チームは古いベースラインを見て作業し、成果物が計画とそろいません。

変更管理ログは、1 件が 1 行の表で、次のような形になります。

ID	内容	提出者	受付日	影響評価	推奨	決定者	状況
CR-007	帳票を2種追加する	顧客の業務担当	3/4	期日 +8 日、費用 +120 万円	承認(予備の範囲を超えるため CCB へ)	CCB	承認済み・計画更新待ち
CR-008	ログイン画面の文言修正	テスター	3/6	影響なし	承認	PM	実施済み・検証済み

場面で組み合わせて解く例です。実装の途中で顧客の担当者が電話で「帳票を2種増やしてほしい」と頼み、開発リーダーが「できそうです」と答えて着手しました。最初の誤りは、記録の前に着手したことです。正しくは、依頼を変更管理ログに記録し、予備レビューで範囲と重複を確かめ、影響評価で期日と費用への響きを出し、推奨を添えます。影響が PM の権限を超える大きさなら CCB へ上げ、承認の結果をログに記録してから、状況を伝え、計画を更新し、実施し、検証し、展開を周知します。この順のどこかが欠けた選択肢は、場面の間では誤りになります。

1.3-5 プロダクトの変更とプロジェクトの変更・スコープクリープ

プロダクトの変更は、作る成果物そのものの仕様の変更で、進め方の変更ではありません。たとえば画面の項目の追加です。**プロジェクトの変更**は、作り方や枠組みの変更で、期日、予算、資源、手順の変更です。プロダクトの変更がプロジェクトの変更を引き起こすことは多く、画面の項目を足すと工数と期日に響きます。

スコープクリープは、承認なしにスコープがじわじわ広がる現象で、誰かが承認した変更ではありません。1回の大きな変更とも違い、気づかないうちに進みます。顧客が「ついでにこれも」と頼み、チームが善意で応じる場面が典型です。対処は、変更管理の手順に載せることです。**スコープ変更**は、手順を通して承認されたスコープの増減で、これは問題ではありません。違いは承認の有無です。

間違えやすいのは、善意の追加を「顧客満足のため」と許す選択肢です。承認されていない追加は、無償であっても統制の外にあります。アジャイルでは、バックログに追加して優先順位を付ける形で受け付けます。

リスクマネジメント [1.4]

1.4-1 一般的なリスク要因

リスクは、起きれば目標にプラスかマイナスの影響を与える、まだ起きていない不確かな事象で、すでに起きた問題ではありません。出題範囲が挙げる一般的な要因は8つあります。

要因	典型的に起きること
新規プロジェクト	前例がなく、見積りや手順の不確実性が高い
新しい経営陣(経営交代・新任の幹部)	優先順位や予算の方針が変わる
規制の変化(規制環境の変化)	新しい法令でスコープや文書が増える
デジタルトランスフォーメーション(DX)	業務とシステムを同時に変えるため、抵抗や移行の混乱が起きる
インフラのEOL(サポート終了)	保守が切れ、脆弱性や互換性の問題が出る
合併と買収(M&A)	システム統合や方針統一が課題になる

要因	典型的に起きること
組織再編	役割や指揮系統が変わり、担当者が入れ替わる
大規模なサイバーセキュリティ事件	計画外の対応と是正が割り込む

場面の文に「新任の CIO が就任した」「ベンダーが製品の保守を終了すると発表した」と出れば、それがリスクの原因です。間違えやすいのは、これらを課題(すでに起きた問題)と取ることです。まだ影響が出ていなければリスクとして登録します。

1.4-2 既知のリスクと未知のリスク、コンティンジェンシー計画とフォールバック計画

既知のリスクは、特定されて分析できるリスクで、想定外のものではありません。リスク登録簿に載せ、対応を決めます。**未知のリスク**は、まだ誰も想定していないリスクで、個別の対応を事前に決められません。未知のリスクには、プロジェクト全体に持たせるマネジメント予備 (management reserve) で備え、使う時は上位の承認を受けます。既知のリスクの残りには、コンティンジェンシー予備を割り当てます。コンティンジェンシー予備の範囲内で使うなら、ベースラインは変わらず、上位の承認も要りません。

コンティンジェンシー計画は、リスクが実際に起きた時(あるいは起きる合図の**トリガー**が出た時)に実行する対応です。**フォールバック計画**は、一次の対応が効かなかった時の第二の手段です。どちらの手順も、トリガーが出る前には発動しません。たとえば主要部品の納品遅れに備えて代替の仕入先に切り替える手順がコンティンジェンシー計画で、その代替先も間に合わない時に出荷を延期して顧客に通知する手順がフォールバック計画です。

区別は順番です。コンティンジェンシーが先で、フォールバックが後です。間違えやすいのは、これらを「リスクが起きる前に実行する対策」とする選択肢です。起きる前に実行するのは軽減や回避で、コンティンジェンシーはトリガーが出るまで動きません。

1.4-3 マイナスのリスクへの対応

マイナスのリスク（脅威）への対応は 4 つあります。何もせずに受け入れる受容も、その 1 つです。

対応	内容	例
受容 (Accept)	何もせず受け入れる。予備を積む場合もある	影響が小さく、対応費用の方が高い
回避 (Avoid)	原因を取り除くか計画を変え、リスクをなくす	実績のない技術の採用をやめる
軽減 (Mitigate)	発生確率か影響を小さくする	試作を先に作って不具合を早く見つける
転嫁 (Transfer)	影響の負担を第三者に移す	保険に入る、定額契約にして超過を受注側が負う

見分けの軸は、リスクがなくなるか（回避）、小さくなるか（軽減）、持ち主が変わるか（転嫁）、そのままか（受容）です。**転嫁はリスクを消しません**。責任と費用の負担を移すだけで、リスクそのものは残ります。保険に入っても事故は起こり得ます。また転嫁は通常、相手に対価を払います。

場面の問で「費用超過のリスクを、定額契約で受注側に負わせる」なら転嫁、「主要部品の調達先を 2 社に分ける」なら軽減です。キーワードだけで選ばず、何が変わるかで決めなければなりません。

1.4-4 プラスのリスクへの対応

プラスのリスク(好機) への対応も 4 つで、マイナスの鏡像です。

対応	内容	例
受容 (Accept)	好機が来れば受けるが、積極的には動かない	想定より早く部材が値下がりしたら歓迎する
強化 (Enhance)	発生確率か効果を高める	早期割引の条件を満たすよう発注時期を前倒しする
活用 (Exploit)	確実に起こす	値下がり見込みの部材を、先に契約して確定させる
共有 (Share)	好機を得意な第三者と分け合う	共同事業体を組み、成功の利益を分配する

マイナスの側と並べると、回避に当たるのが活用、軽減に当たるのが強化、転嫁に当たるのが共有で、受容は共通です。強化は確率を上げる努力、活用は確率を 100% にする行動です。間違えやすいのは、強化と活用の取り違えです。「確実に実現させる」と書かれていれば活用で、強化は確実とまでは言いません。

1.4-5 定性的リスク分析

定性的リスク分析は、数値でなく、高・中・低の段階でリスクに優先順位を付ける作業です。基本は**発生確率×影響度**で、確率と影響度の 2 つの軸の表(確率・影響度マトリクス)に載せます。たとえば確率が「高」で影響度が「中」のリスクは、確率も影響度も「低」のリスクより上に置きます。確率と影響額を掛けて金額を出す計算は、1.4-6 で扱う側の作業だ。

出題範囲は、2 つの観点を追加で挙げています。**相互関連性**(Interconnectivity) は、あるリスクが他のリスクとつながっている度合いで、

単独の確率や影響度では表せません。1 つが起きると連鎖して他も起きやすくなるリスクは、単独の評価より優先度を上げます。**検出可能性** (Detectability) は、リスクが起きる前や起きた直後に気づける度合いです。気づきにくいリスクは、手を打つ時間がないので優先度を上げます。

場面では、確率と影響度が同程度の 2 つのリスクを並べ、「他の作業へ波及する」「兆候が見えない」と書かれた方を先に扱う答えを選びます。

確率・影響度マトリクスに載せる前の素材は、次のような表の形で、まだ順位は付いていません。

リスク	確率	影響額	期待値(確率×影響額)	相互関連性	検出可能性
R1 ベンダーの納期遅延	40%	50,000	20,000	高 (R3 を誘発)	低 (兆候が見えにくい)
R2 担当者の離職	10%	100,000	10,000	中	中
R3 テスト環境の確保の遅れ	30%	30,000	9,000	低	高 (予定表で見える)

期待値だけで並べると R1、R2、R3 の順です。R1 は他のリスクを誘発し、兆候も見えにくいので、順位はさらに上がります。R3 は期待値が小さく、検出もしやすいので、定期的確認で足ります。場面の問では、期待値に近い 2 つのリスクを並べて、相互関連性か検出可能性で順位を決めさせます。

1.4-6 定量的リスク分析

定量的リスク分析は、リスクの影響を数値 (金額や日数) で評価する作業で、高・中・低の段階付けではありません。出題範囲は、定量 (Quantitative) の下

にシミュレーションを置き、それと並べて影響分析（確率と影響）と状況・シナリオ分析を挙げています。ここでは、金額や日数で表せる3つの方法としてまとめます。

- **シミュレーション:** 見積りの幅（最小・最頻・最大）を乱数で何千回も試し、完了日や総費用の確率分布を得る方法（モンテカルロ法）。「予算内に終わる確率は70%」といった答えが出ます。
- **影響分析:** 確率と影響額を掛けて期待金額（EMV）を求める方法。確率30%、影響40,000ドルなら12,000ドル。複数の事象を合計する時は、脅威を負、好機を正の値で書きます（3.3-5）。
- **状況・シナリオ分析:** 「ベンダーが2週間遅れたら」など、前提を変えた場合の結果を比べる方法。

定性は順位付けで、定量は金額や日数の算出です。定量分析は手間がかかるため、定性で絞った上位のリスクに行います。間違えやすいのは、定性の結果である「高・中・低」を定量と取ることです。

1.4-7 リスクと課題・リスクと変更のつながり

リスクは未来の不確定、課題は今起きている問題です。リスクが現実になった時に、課題へ変わるのであって、別々の物が並んでいるわけではありません。この時はリスク登録簿から課題ログへ移し、リスクのコンティンジェンシー計画を実行します。課題として起きたものにリスクの確率はなく、確率は100%です。

リスクと変更のつながりは2つあります。1つは、リスクへの対応が、スコープ、スケジュール、予算の変更を要する場合で、変更管理の手順に載せます。もう1つは、承認した変更が新しいリスクを生む場合で、承認すれば安全になる

とは限りません。影響評価でリスクの変化も見ておかなければなりません。たとえば「ベンダーを替える」変更は、新ベンダーの習熟のリスクを増やします。

間違えやすいのは、リスクへの対応を PM の判断だけで計画に入れる選択肢です。ベースラインを変えるなら変更要求です。

1.4-8 リスクの役割と責任

リスクごとに**オーナー**（責任者）を 1 人決めます。オーナーの仕事は、リスクの監視、トリガーの確認、対応の実行、状況の報告で、登録するだけでは終わりません。担当は、そのリスクに最も影響を与えられる人です。登録簿に名前を書いて初めて責任が生まれます。

エスカレーションの点は、リスクが PM やオーナーの権限を超える時に、より上位へ上げる条件と相手です。たとえば「影響額が予備を超える」「期日に影響する」場合にスポンサーへ報告する、と計画で決めておきます。上位が対応を決めて返すまで、オーナーは待つだけではありません。

間違えやすいのは、オーナー不在のリスクを「全員の責任」とする選択肢です。担当者が決まっていないリスクは、誰も動きません。

課題管理 [1.5]

1.5-1 課題の役割と責任・エスカレーションの経路・課題の追跡

課題は、すでに起きて目標に影響している（または影響しつつある）問題です。課題ごとにオーナーを決め、解決の期限を付けます。**エスカレーションの経路**は、担当者から PM、スポンサー、運営委員会のように、どの順で誰に上げるかを事前に定めたものです。オーナーの権限で解決できない、期限を過ぎる、他の作業に波及する場合に上げます。

課題の追跡とは、**課題ログ**に記録して状況を更新し続けることで、一度書けば終わりではありません。項目は ID、内容、日付、オーナー、優先度、状況、期限、解決の記録です。口頭の報告や記憶に頼らず、ログが唯一の台帳です。課題ログのツールの扱いは 3.1 で出てきます。

間違えやすいのは、気づいた人がログに載せずに直す選択肢です。記録しないと、再発の分析や報告ができません。

1.5-2 課題と変更のつながり

課題の解決策が、ベースラインの変更を要することがあります。たとえばテストで重大な欠陥が見つかり(課題)、修正のために期日を 2 週間延ばす場合です。この時は、課題の解決の一部として変更要求を出し、変更管理の手順(記録、評価、承認)を通します。逆に、承認された変更が新しい課題を生む場合もあります。

課題を解決するのに、スコープや予算の枠内で収まるなら変更要求は要りません。枠を超えるかどうかで決まります。間違えやすいのは、課題だから緊急に直して構わない、という選択肢です。ベースラインを動かすなら手順が要ります。

1.5-3 課題の優先順位付け

優先順位は、次の観点に合わせて決めます。

観点	問い
深刻度 (severity)	その課題が、どれほど大きな損害か
プロジェクトへの影響	スコープ、スケジュール、予算、品質のどれにどれだけ響くか
緊急度 (urgency)	いつまでに対応しないと悪化するか

観点	問い
組織への影響範囲	他のプロジェクトや部門、顧客にまで及ぶか
エスカレーション	上位の判断が要るか

深刻度と緊急度は別の軸で、同じ物差しではありません。深刻度が高く、期限に余裕がある課題もあれば、深刻度は低いけど今日中に直さないと間に合わない課題もあります。両方が高いものから対応します。クリティカルパス上の課題は、全体の完了日に直結するため上位に置きます。

場面の問では、「本番停止（深刻度高）、顧客全体に影響（範囲広）」の課題と、「画面の誤字（深刻度低）」を並べ、前者を選ばせます。範囲が組織の外に及ぶなら、エスカレーションも欠かせません。

場面で組み合わせる例です。本番の決済機能が止まり（深刻度が高い）、全国の顧客が購入できない（影響範囲が広い）一方で、管理画面のボタンの位置のずれも報告されています（深刻度が低い）。さらに、来週の発表会で使う画面の表示の不具合が出ています（深刻度は中程度だが、緊急度は高い）。優先は、決済の停止、発表会の画面、ボタンの位置の順です。決済の停止は、組織の外に及ぶので、エスカレーション経路でスポンサーにも直ちに上げます。同時に、リスク登録簿に「決済基盤の障害」があり、コンティンジェンシー計画が決まっていれば、それを実行します。計画が無いなら、回避策で売上の損失を止め、並行して根本原因を調べます。

1.5-4 解決計画と結果の記録

出題範囲は解決計画の下に、コンティンジェンシー計画の実行、根本原因分析、優先順位付け（1.5-3）、回避策の 4 つを置いている。優先順位付けは前の節で見たので、ここでは残りの 3 つを扱います。どれか 1 つで済ませてはいけません。

- **コンティンジェンシー計画の実行:** 課題が、すでにリスクとして登録され、対応が決まっていた場合に実行します。
- **根本原因分析:** 症状でなく原因を探します。「なぜ」を繰り返す 5 回のなぜや、特性要因図 (3.3) を使います。原因を直さないと、同じ課題が再発するので、症状の手当てだけでは足りません。
- **回避策 (work-around):** 根本原因が直るまでの一時的な手段です。代替の手順で作業を止めずに続けます。回避策は暫定で、根本解決ではありません。

回避策と根本原因の対応を同時に進めるのが定石で、回避策だけで終わるのは誤りです。解決したら**結果を記録**します。原因、取った策、結果、得た教訓を課題ログに書き、教訓は振り返り (1.7) に回します。間違えやすいのは、回避策で動いたので課題を閉じる選択肢です。根本原因は残ったままで、課題は解決していません。

課題ログの 1 件は、次のような形です。

ID	内容	オーナー	優先度	状況	期限	対応
IS-012	受け入れ環境のデータが壊れた	テスト担当	高	対応中	3/8	回避策: 前日のバックアップを復元。 根本原因:夜間バッチの設定誤りを調査中
IS-013	仕様書の誤字	設計担当	低	未着手	3/20	次の改訂で直す

オーナーがいること、期限があること、回避策と根本原因が別の欄に書かれていることが、台帳として整っている目印です。リスクが現実になって IS-012 のような課題になった場合は、元のリスクの番号を課題の備考に残し、両方の台帳でつながりを追えるようにします。

スケジュールの作成と管理 [1.6]

1.6-1 マイルストーンと活動の特定・スプリントゴール

活動は、所要時間と資源を持つ作業の単位です。**マイルストーン**は、所要時間が 0 の重要な節目（承認の取得、設計の完了、本番稼働など）で、作業そのものではありません。「到達した」という事実を表す印です。次のマイルストーンまでに何を終えるべきかを見て、必要な活動を洗い出します。

アジャイルでは、各スプリントに**スプリントゴール**（そのスプリントの単一の目的）を置き、ゴールのないスプリントはありません。スプリントバックログは、スプリントゴール、選んだバックログ項目、その実現の計画で構成されます[^c1-4]。予測型のマイルストーンに当たるのが、反復のゴールと反復の終わりに出るインクリメントです。間違いやすいのは、マイルストーンを数日かかる作業と取る選択肢です。所要時間は 0 です。

1.6-2 依存関係の種類と課題のエスカレーション

依存関係は、ある活動が別の活動や条件に左右される関係です。性質は 4 つあり、「強制か任意か」と「外部か内部か」の 2 つの軸で決まる。1 つの依存関係が「強制で外部」のように、両方の軸の性質を同時に持つこともあります。

種類	別名	内容	例
強制	ハードロジック	作業の性質や契約で順序が決まる	基礎工事の後でなければ壁を立てられない

種類	別名	内容	例
任意	ソフトロジック	慣行や好みで決めた順序。変えられる	先に画面を作ってから帳票を作る習慣
外部		プロジェクトの外の者や出来事に依存する	ベンダーの納品、規制当局の承認
内部		プロジェクトの中で、チームが管理できる	コーディングの後にチーム内でテスト

任意の依存関係は、短縮のために変更できるので、期日を縮める時に最初に見直します。強制と外部は変えられません。外部の依存関係は管理の範囲外なので、遅れそうなら早めに状況を確認し、課題としてエスカレーションします(出題範囲が同じ列に Issue escalation を置いているのはこのためです)。

間違えやすいのは、強制と外部の取り違いです。「納品は注文の後」は強制(性質上の順序)ですが、「ベンダーの納期」は外部です。

1.6-3 先行・後続の関係

先行は先にある活動、**後続**は後ろの活動です。関係は 4 種類で、「どちらの終わりが始まりか」で名前が付きます。

関係	意味	例
終了-開始 (FS)	先行が終わったら後続が始まる	設計が終わったら実装を始める
開始-開始 (SS)	先行が始まったら後続も始められる	工事が始まったら進捗の記録も始める
終了-終了 (FF)	先行が終わるまで後続も終われない	執筆が終わるまで校正も終われない

関係	意味	例
開始-終了 (SF)	先行が始まるまで後続は終われない	新システムが稼働するまで、旧システムの運用を終えられない

最も多いのは FS で、SF はほとんど使いません。見分ける手順は、「先行の」とちら側と、「後続の」とちら側かを、前から順に読むことです。SF は「先行の開始」と「後続の終了」を結びます。間違いやすいのは FF と SF で、FF は「終わり同士」、SF は「先行の始まりと後続の終わり」です。

1.6-4 資源の負荷 (リソースローディング)

リソースローディングは、各資源 (人、設備) を、期間ごとにどの活動へどれだけ割り当てたかを見る分析です。1 日の稼働を 8 時間とすると、同じ日に 8 時間の作業を 2 つ割り当てた人は $16 \text{ 時間} \div 8 \text{ 時間} = 200\%$ で、**過負荷**に当たり、1 人で 2 倍は働けません。稼働率が 100% を超える状態が過負荷で、100% ちょうどは満杯ですが過負荷ではありません。

過負荷の解消には 2 つの手があります。**資源の平準化** (リソースレベリング) は、作業を後ろにずらして負荷を均す方法で、期日が延びることがあります。**資源の円滑化** (スムージング) は、余裕 (フロート) の範囲内だけで調整する方法で、完了日は動きません。

方法	完了日	クリティカルパス
平準化	延びることがある	変わることがある
円滑化	変わらない	変わらない

場面では、「期日を動かせない」なら円滑化か資源の追加、「期日が延びてもよいから過負荷を解消したい」なら平準化が答えです。負荷の山は、グラフ (ヒ

ストグラム)で見ると分かります。

資源の負荷の読み方は、表で確かめます。担当者 A の週の稼働を 40 時間とします。

週	タスク T1	タスク T2	合計	稼働率
第 1 週	30 時間	0	30 時間	75%
第 2 週	30 時間	20 時間	50 時間	125%
第 3 週	0	20 時間	20 時間	50%
第 4 週	0	0	0	0%

第 2 週は 50 時間で、稼働率が $50 \div 40 = 125\%$ になり、1 人では担えません。T2 に 1 週間の余裕があるなら、T2 を 1 週後ろへずらして第 3 週と第 4 週に置き、各週 20 時間にします。第 2 週は 30 時間で 75% に下がり、完了日は動きません。これが円滑化 (スムージング) です。T2 に余裕がなく、ずらすと完了日が延びるなら、平準化の判断になり、期日への影響を承認してもらいます。

1.6-5 見積りの技法とコンティンジェンシー予備・バッファ

技法	内容
類推見積り	過去の似たプロジェクトを基準にする。速いが粗い
パラメトリック見積り	単位あたりの量×数量で出す (例: 1 画面 3 日×20 画面=60 日)
ボトムアップ見積り	小さな作業ごとに見積もり、積み上げる。正確だが手間がかかる
三点見積り	楽観値 (O)、最頻値 (M)、悲観値 (P) から求める

技法	内容
専門家の判断	経験者の知見を使う

三点見積りには 2 つの式があります。PERT の加重平均は $(O + 4M + P) \div 6$ で、三角分布は $(O + M + P) \div 3$ です。 $O=4$ 日、 $M=6$ 日、 $P=14$ 日なら、PERT は $(4 + 24 + 14) \div 6 = 7.0$ 日、三角分布は $(4 + 6 + 14) \div 3 = 8.0$ 日になり、PERT は最頻値を 4 倍に重みづけるため、単純平均とは一致しません。標準偏差は $(P-O) \div 6 = 1.67$ 日で、幅が広いほど不確実です。式は問題の指定に従って使い分けます。

コンティンジェンシー予備は、既知のリスクに備える時間や予算の上乗せです。**バッファ**は、スケジュールの安全余裕で、不確実な作業の見積りに混ぜず、後ろにまとめて置く考え方があります。決める時は、リスクの大きさ(定量分析の結果)から求め、勘では決めません。見積りのたびに各担当者が少しずつ余裕を足すと、全体に無駄が積み上がります。

1.6-6 ストーリーの見積り

アジャイルの作業は、大きさの順に 3 段あります。**エピック**は、複数のスプリントに渡る大きな要求です。**ユーザーストーリー**は、利用者の視点で書いた、1 スプリント内で終わられる要求です。**タスク**は、ストーリーを実現する具体的な作業で、ポイントでなく時間で見積もり、ストーリーのようにポイントは付けません。

ストーリーポイントは、ストーリーの相対的な大きさを表す単位で、日数や時間ではありません。基準のストーリーに比べて「その 2 倍」「3 倍」と決め、フィボナッチ数列(1、2、3、5、8、13)の値がよく使われます。チームで札を同時に出して合わせる**プランニングポーカー**で決めます。数字が離れたら、理由を話し合って再度出します。

チームが1スプリントで終わるポイントの合計が**ベロシティ**です。ベロシティが20で、バックログの残りが100ポイントなら、 $100 \div 20 = 5$ スプリントで終わる見込みになりますが、バックログが増えれば、この見込みはそのまま使えません。間違えやすいのは、ポイントを時間に換算して個人を比べることです。ポイントはチーム内の相対値で、他チームとは比べられません。

1.6-7 クリティカルパス分析(経路の計算・余裕)

クリティカルパスは、プロジェクトの最初から最後まで**の経路のうち、最も長い所要期間の経路**です[^c1-7]。この経路の期間が、プロジェクトの最短の完了期間を決めます。クリティカルパス上の活動の**余裕(フロート)**は0で、遅れるとそのまま完了日が遅れます。非クリティカルな活動には余裕があり、その範囲内の遅れなら完了日に影響しません。

次のネットワークで求めます。活動の期間は日数で、すべて終了-開始の関係です。

活動	期間	先行
A	3	なし
B	4	A
C	2	A
D	5	B
E	6	C
F	2	DとE

経路はA→B→D→Fが $3+4+5+2=14$ 日、A→C→E→Fが $3+2+6+2=13$ 日です。最長の14日がクリティカルパスで、完了までは14日です。CとEは余裕が1日あり、余裕が0の活動ではありません。順方向(早い開始・早い

終了)と逆方向(遅い開始・遅い終了)で計算すると、Cの余裕は遅い開始4－早い開始3=1、Eは6－5=1です。

この結果から3つの判断ができます。Dが2日遅れれば、完了は16日になります。Eが1日遅れても、余裕の範囲なので完了は14日のままです。Eが3日遅れれば、A→C→E→Fが16日になり、クリティカルパスはA→B→D→Fのままではありません。

期間を縮めるには、**クラッシング**(資源を足して縮めます。費用が増える)と**ファストラッキング**(順番の作業を並行にします。手戻りのリスクが増える)があります。縮めるのは、クリティカルパス上の活動だけです。間違えやすいのは、余裕が0の経路が複数ある場合で、どれも遅らせられません。

期間を縮める例です。先ほどのネットワークで、Dを5日から3日へ2日縮める(クラッシング)と、A→B→D→Fは $3+4+3+2=12$ 日になります。A→C→E→Fは13日のままなので、クリティカルパスが入れ替わり、全体は14日から13日へ、1日しか縮みません。縮めたのは2日でも、効いたのは1日です。次に縮めるなら、新しいクリティカルパスのCかEの活動を対象にします。縮めるたびに、クリティカルパスが入れ替わっていないかを確認めます。

1.6-8 スケジュールのツールとスケジュールの維持

スケジュールの**ツール**は、活動、依存関係、資源、期間を入力して、ガントチャートやクリティカルパスを自動で出す専用のソフトウェアです。表計算でも作れますが、依存関係と余裕の再計算は手作業になり、ツールほど確実ではありません。アジャイルではタスクボードとバックログのツールを使います。

維持の観点は次の5つです。

- **予備の消費**: コンティンジェンシー予備やバッファの使われる速さ。進捗より速く減っているなら、問題の兆候です。

- **ケイデンスへの影響:** ケイデンスは、スプリントや定例の一定のリズムです。途中での変更や中断は、リズムを崩し、ベロシティを下げます。
- **予測:** 現在の実績 (ベロシティ、SPI) から、完了日を予測し直します。
- **公開と共有:** 最新の版を決まった場所に出し、関係者への周知まで行い、古い版は残しません。古い版が出回ると、判断を誤ります。
- **更新:** 実績を反映し、残りの作業を見直します。

間違えやすいのは、遅れた作業の穴を、バッファを使わずに黙って吸収する選択肢です。遅れは報告し、予備を使ったことも記録します。

1.6-9 スプリント計画とバックログの優先順位付け

スプリント計画は、スプリントの最初に行う会議で、「なぜこのスプリントに価値があるか」「何を終わるか」「どう進めるか」を決める場であり、途中で開く会議ではありません。1 か月のスプリントなら最大 8 時間で、それより短いスプリントは短くします^[^c1-4]。入力プロダクトバックログの上位の項目とチームのベロシティで、出力はスプリントゴールとスプリントバックログです。スプリントバックログには、過去の実績 (ベロシティ) と使える時間に収まる量だけを入れます。

プロダクトバックログは、順番を付けた、変わり続ける項目の一覧で、順序と中身の責任はプロダクトオーナーにあります^[^c1-4]。優先順位を決める要素は、顧客への価値、リスク (早く片付けたい不確実なもの)、依存関係、工数、期限です。価値が高く工数が小さい項目から取るのが定石で、リスクの大きい項目は早めに取り、不確実性を先に減らします。

間違えやすいのは、スプリントの途中で追加を受けることです。スプリントゴールを脅かす追加は、次のスプリントに回します。

1.6-10 ベースラインの改訂と再ベースライン

ベースラインは、承認されたスコープ、スケジュール、予算の基準で、実績との比較に使う物差しですから、勝手には動かさせません。**ベースラインの改訂** (revise baseline) は、承認された変更を反映して、基準の一部を更新することです。変更管理の手順を通し、元の値と変更履歴を残します。

再ベースライン (rebaseline) は、遅れや超過が大きく、元の基準が実態と合わなくなった時に、基準全体を新しく作り直すことです。過去の差異を捨てて新しい基準に置き換えるため、実績の評価が見えにくくなります。実施にはスポンサーなど上位の承認が必要で、PM の一存では行えません。

	改訂	再ベースライン
範囲	一部(承認された変更)	全体(基準を作り直す)
契機	個々の変更	差異が大きすぎて基準が機能しない
差異の履歴	引き継ぐ	新しい基準からやり直す

間違えやすいのは、少し遅れただけで再ベースラインする選択肢です。まず変更管理で改訂し、それでも基準が使えない時に限ります。

品質管理とパフォーマンス管理 [1.7]

1.7-1 振り返り・教訓とスプリントレビュー

スプリントレビューは、スプリントの終わりに、できたインクリメントをステークホルダーに見せ、フィードバックを得て、次のバックログを調整する会議です。1か月のスプリントで最大 4 時間です[^c1-4]。見るのは成果物です。

スプリントレトロスペクティブ(振り返り)は、チームが自分たちの進め方(人、関係、プロセス、ツール)を点検し、次のスプリントでの改善を決める会議です。1か月のスプリントで最大3時間です[1-4]。見るのは進め方です。

教訓(lessons learned)は、プロジェクトで得た知見の記録で、終了時にまとめるだけでなく、全期間を通して集めます。次のプロジェクトで使うので、組織の記録に残します。

区別は対象です。レビューは成果物(作った物)を見る会で、進め方(作り方)を見る会ではありません。間違えやすいのは、会の対象の取り違いです。レビューにはステークホルダーが出て、レトロスペクティブはチームの内側で行います。

1.7-2 SLA と KPI・OKR

SLA(Service Level Agreement)は、サービスの提供者と利用者が、サービス水準を数値で合意した文書です。可用性 99.9%、障害の受付から 30 分以内に応答、のような目標と、満たせなかった時の扱い(返金など)を書きます。外部の顧客とだけでなく、社内の部門間でも結ぶので、社外限定の文書ではありません。

KPI(Key Performance Indicator)は、目標への達成度を測る主要な指標です。バグの検出数、顧客の満足度、納期の遵守率などで、目標値と比べて状況を判断します。

OKR(Objectives and Key Results)は、定性的で意欲的な目的(Objective)と、その達成を測る数値の成果(Key Results)を組にした目標管理の手法です。例えば Objective「顧客に選ばれる安定したサービスにする」と、Key Results「障害による停止を月 1 回以下にする」です。

	SLA	KPI	OKR
性質	提供者と利用者の約束	状況を測る指標	挑戦的な目標と測定
相手	提供者と利用者(社外・社内とも)	組織内の管理	組織内の目標設定

間違えやすいのは、SLA と KPI の取り違えで、両者は同じ物ではありません。SLA は「約束した水準」で、KPI は SLA を守れているかを測る指標にもなります。

1.7-3 コストとスケジュールのパフォーマンス(コスト差異・スケジュール差異の計算)

アードバリュー管理 (EVM) は、計画、実績の作業量、実績の費用の 3 つを比べる手法です。

略語	名前	意味
BAC	完成時の予算	全体の予算
PV	計画価値	今日までに終えるはずだった作業の予算
EV	出来高	今日までに実際に終えた作業の予算
AC	実績原価	今日までに実際に使った費用

差異は、出来高を基準に引き算します[^c1-8]。

- **コスト差異 $CV=EV-AC$:** 負なら予算超過、正なら予算内です。
- **スケジュール差異 $SV=EV-PV$:** 負なら遅れ、正なら先行です。

指数は割り算で、**CPI = EV ÷ AC**、**SPI = EV ÷ PV** です。1 より小さければ悪い状況です。

BAC = 100,000、PV = 40,000、EV = 30,000、AC = 36,000 の場合は、CV = 30,000 - 36,000 = -6,000、SV = 30,000 - 40,000 = -10,000 で、どちらも負なので、予算内でも予定どおりでもありません。CPI = 30,000 ÷ 36,000 = 0.83 (正確には 0.8333...)、SPI = 30,000 ÷ 40,000 = 0.75 です。

予測は、**EAC** (完成時総コスト見込み)、**ETC** (残りの見込み)、**VAC** (完成時差異) です。現在の効率が続くなら $EAC = BAC \div CPI = 100,000 \div 0.8333 = 120,000$ です。ETC = EAC - AC = 84,000、VAC = BAC - EAC = -20,000 で、VAC が負なので、この例は予算内に収まりません。差異が今回限りで、残りは計画どおり進むなら $EAC = AC + (BAC - EV) = 36,000 + 70,000 = 106,000$ です。

覚え方は、差異は引き算で単位が金額 (スケジュール差異も金額)、指数は割り算で単位がありません。間違えやすいのは、SV の引く順 (PV を先に書かない) と、CV を「AC - EV」と逆に書く誤りです。

判断の例です。先ほどの数値 (CPI が 0.83、SPI が 0.75) は、費用でも日程でも状況が悪いことを示します。BAC ÷ CPI の EAC は 120,000 で、予算を 20,000 超えます。残りの作業で予算内に収めるのに必要な効率を表す TCPI (To-Complete Performance Index) は、 $(BAC - EV) \div (BAC - AC) = 70,000 \div 64,000 = 1.09$ で、1 を超えていて、楽な水準ではありません。これまでの効率が 0.83なのに、残りで 1.09 を出さなければ予算に収まりません。実現が難しいので、PM は予算の見直し (変更要求) か、スコープの縮小を、スポンサーに諮ります。遅れの原因がクリティカルパス上かどうかも見ます。パス外の遅れでも、余裕 (フロート) の範囲内なら完了日には響きません。

1.7-4 監査と検査

監査は、独立した立場の者が、プロセスや記録が決まりやルールに合っているかを点検する活動です。品質の監査、調達の監査、リスクの監査などがあります。見るのは「決めたやり方を守っているか」です。

検査は、成果物そのものを調べ、要求を満たしているかを判定する活動で、手順の遵守を見る監査とは対象が違い、記録の突き合わせではありません。測定、レビュー、ウォークスルーなどの形で行います。

	監査	検査
対象	プロセス・記録	成果物
問い	手順どおりに行ったか	要求を満たしているか
実施者	独立した者	担当チームや品質担当

場面では、「製造記録が手順書と合っているか」は監査、「完成した画面が仕様を満たすか」は検査です。

1.7-5 テスト計画とテストの種類

テスト計画は、何を、どの順で、誰が、どの環境で、どの基準で合格とするかを決めた文書です。テストは、テスト実施、欠陥の記録、修正、再テストのサイクルを繰り返します。

種類	目的	実施の場面
単体テスト	最小の部品 (関数やモジュール) が正しく動くか	開発者が実装直後に行う
スモークテスト	主要な機能が動くかを素早く確かめ、詳しいテストに進めるか判断する	ビルドの直後

種類	目的	実施の場面
回帰テスト	変更のあとに、既存の機能が壊れていないか	修正や機能追加の後
ストレステスト	通常を超える負荷をかけ、限界と壊れ方を見る	本番前
性能テスト	想定で負荷で、応答時間や処理量が基準を満たすか	本番前
ユーザー受け入れテスト (UAT)	利用者や顧客が、業務の要求を満たすと確かめる	引き渡しの直前

見分けは、「壊れていないか」なら回帰、「限界まで負荷」ならストレス、「基準の速さ」なら性能、「顧客が確認」なら UAT で、名前の印象だけでは選べません。UAT は開発側でなく利用者側が行い、合格が受け入れ（サインオフ）の根拠になります。間違いやすいのはストレスと性能で、通常の想定内の負荷が性能、想定を超えるのがストレスです。

1.7-6 検証と妥当性確認

検証 (verification) は、「正しく作ったか」を確かめます。仕様や設計どおりに作れているかを、レビュー、検査、テストで判定します。**妥当性確認** (validation) は、「正しい物を作ったか」を確かめる作業で、仕様との照合だけでは終わりません。顧客や利用者の本来のニーズを満たしているかを、受け入れや実際の利用で判定します。

仕様どおりに完璧に作っても、仕様が顧客の望みと違えば、検証は通っても妥当性確認は通りません。単体テストや設計レビューは検証、UAT は妥当性確認に当たります。終結フェーズでの成果物の確認と顧客のサインオフも妥当性確認です。間違いやすいのは、「作った側が確かめる＝検証、顧客が確かめる＝妥当性確認」の対応を逆にすることです。

1.7-7 稼働後のサポートと保証期間

稼働後のサポートは、本番稼働の後、利用者が新しい仕組みに慣れ、初期の不具合が直るまでの間、プロジェクトチームが手厚く支える期間です。問い合わせの窓口を置き、不具合に素早く対応して、運用の担当へ引き継ぐまで、プロジェクトの責任は終わりません。

保証期間 (warranty period) は、納品後の一定期間、納品物の欠陥を、売り手が追加費用なしで直す期間です。契約で期間と範囲を決めます。保証の後の支援は、有償の**保守契約**で受けます。

間違えやすいのは、稼働後のサポートが終わるまでプロジェクトを閉じない選択肢で、運用へ移した後は、サポートの終了まで待つ必要はありません。サポート期間と引き継ぎの完了を計画に入れ、運用へ移してから閉じます。移す前に閉じてはいけません。

コミュニケーション管理 [1.8]

1.8-1 伝達の方法の評価

伝達の方法は 4 つの軸で評価し、場面に合うものを選びます。

軸	一方	他方
時間	同期(会議、電話など、同時にやり取りする)	非同期(メール、チケット、録画など、時間差でやり取りする)
形	書面(記録が残る、正確)	口頭(速く、ニュアンスが伝わる)
格式	公式(契約、報告書、変更要求など、正式な手続き)	非公式(雑談、チャットの軽い連絡)

軸	一方	他方
相手	外部(顧客、ベンダー、規制当局)	内部(チームや社内の部門)

選び方の目安は次のとおりです。複雑で、すぐに合意が要る議論は同期の口頭です。後で証拠が要る決定や、変更の承認は書面で、公式にします。時差のある相手への連絡は非同期の書面で、即時の返事は期待できません。外部の相手には、契約や信用に関わるので、公式の書面を基本にします。

間違えやすいのは、口頭で合意した内容を、書面に残さない選択肢です。口頭で決めても、後で確認のメールを書面に残します。

1.8-2 コミュニケーションの基盤と手段の整備

基盤(プラットフォーム)と**手段**(モダリティ)は、情報を流すための道具と型です。メール、チャット、ビデオ会議、共有ドライブ、Wiki、課題管理ツール、ダッシュボードなどから、プロジェクトに合うものを選んで整えるのであり、全部を使う必要はありません。

整える時に決めることは4つです。どの情報をどの道具で流すか、誰が使えるか(アクセス権)、どれくらいの速さで返事をするか(応答の目安)、記録はどこに残すか、です。ツールは増やしすぎず、1つの目的に1つの道具を割り当てます。同じ連絡が複数の場所に散ると、見落としが出ます。

間違えやすいのは、新しい道具を、全員が使えることを確かめずに採用することで、導入しただけでは使われません。アクセス権と習熟の教育を先に整えます。

1.8-3 伝達の障壁と克服

障壁	例	克服の手
言語	母語が違い、専門用語が通じない	平易な表現、用語集、通訳や翻訳、書面での確認
時差と地理	重なる勤務時間が少ない	重なる時間帯に会議を置く、会議時間を交代で負担する、非同期の記録を厚くする
技術	回線が細い、ツールが使えない、アクセスできない	手段を代える、事前に接続を試す、代替の連絡手段を決める
文化の違い	上下関係の感覚、直接的か婉曲か、祝日や慣習	相手の慣習を学ぶ、確認を重ねる、祝日を計画に入れる

克服の原則は、相手に合わせて手段を変えることと、理解の確認を取ることで。相手がうなずいても、理解したとは限りません。内容を言い換えて返してもらいます。

場面では、「米国と日本のチームで、定例会議の時刻が片方の深夜になる」なら、時間の交代制か非同期の併用が答えです。「片方だけ負担する」は不公平で、士気を落とします。

1.8-4 伝達の記録の維持

伝達の記録（メール、議事録、決定の記録、報告書）は、プロジェクトの証跡です。維持の観点は3つで、保管だけではありません。

- **セキュリティ:** 内容の機密度に合わせてアクセスを制限し、暗号化し、権限のない者に流れないようにします。

- **完全性:** 記録が改ざんされず、最新の版と古い版の区別がつくようにします。版管理を行い、承認の記録は変更不可の形で保管します。
- **保管(アーカイブ):** 保存期間を決めて保管し、期間が過ぎたら処分します。法令や契約が定める期間は、短くしてはいけません。

間違えやすいのは、個人のチャットや私用のメールに、決定を残す選択肢です。組織の管理下でない場所は、退職や削除で失われ、監査にも出せません。

1.8-5 伝達の統制(伝達の問題のエスカレーション・コミュニケーション計画の改訂)

伝達の統制は、計画どおりに情報が届いているかを監視し、ずれを直すことです。伝達の問題(情報が届かない、誤解される、遅れる)は、すぐに原因を調べ、解決できなければ上位へエスカレーションします。たとえば、顧客が重要な連絡を受け取っていないなら、PM が直接確認し、必要ならスポンサーが顧客に連絡します。

コミュニケーション計画の改訂は、ステークホルダーの増減、役割の変更、手段の変更、フィードバックで見つかったずれが起きた時に行います。改訂した計画は、関係者に周知しなければ意味がありません。計画は作って終わりではありません。間違えやすいのは、問題が起きてもメンバーに個別に口頭で伝えるだけで、計画を直さない選択肢です。仕組みを直さないと、同じ問題が再発します。

場面から方法を選ぶ例です。

場面	選ぶ方法
日本と米国のチームが、重なる勤務時間のない状況で、仕様の質問を毎日やり取りする	非同期の書面(課題管理ツールやメール)。返事の期限を決める

場面	選ぶ方法
設計方針で意見が割れ、すぐに合意が要る	同期の会議（ビデオ会議）で口頭。結論は書面で残す
顧客に、納期の延期を正式に通知する	外部向けの公式な書面。口頭は補助
チーム内の軽い進捗の確認	内部の非公式なチャット

重なる勤務時間がない相手との会議は、時刻を交代で負担し、議事録と録画を共有して、会議に出られない人を補います。言語の壁があるなら、会議の前に資料を配り、会議の後に書面で確認を取るので、口頭だけには頼りません。

会議運営 [1.9]

1.9-1 協働型の会議

協働型は、参加者が一緒に考え、案や要求を作る会議です。

種類	内容
ワークショップ	参加者が共同作業で成果物（要求、設計、計画）を作る
フォーカスグループ	選んだ利用者や専門家に、製品への意見や期待を聞く
JAD/JAR	顧客と開発者が同席して要求を決める（JAD: Joint Application Development）、または成果物を共同で見直す（JAR: Joint Application Review）
ブレインストーミング	他人の案を批判せず、まず多くのアイデアを出し、後で絞る

区別は、誰が何をするかです。フォーカスグループは、司会が質問し、参加者が意見を述べる聞き取りで、合意を作る会ではありません。JAD は、要求の合意を目指す共同作業です。ブレインストーミングは、批判を後回しにして量を出します。間違いやすいのは、ブレインストーミングの途中で案の良し悪しを議論することです。評価は出し切った後です。

1.9-2 情報共有型の会議

情報共有型は、情報を伝え、状況を知らせる会議で、意思決定の場ではありません。

- **デモ・発表:** 完成した機能や成果物を見せ、ステークホルダーのフィードバックを得ます。スプリントレビューでの実演が代表例です。
- **スタンドアップ:** チームが毎日、短時間 (15 分以内) で、進捗、次にやること、障害物を共有します。スクラムのデイリースクラムは 15 分です[^c1-4]。立って行うのは、長引かせないためです。
- **状況(ステータス)会議:** PM がスケジュール、予算、リスク、課題の現状を、関係者に報告します。

スタンドアップは、問題を解決する場ではなく、見つけて、別の場で解決する場です。深い議論が始まったら、後で関係者だけが残って話します。間違いやすいのは、スタンドアップを PM への報告会にして長引かせる運営で、15 分を超えると本来の形ではありません。

1.9-3 決定型の会議

決定型は、判断や優先順位を決める会議です。

- **リファインメント:** プロダクトバックログの項目を、詳しく書き、分割し、見積りと優先順位を付ける作業です。次のスプリントで取り組める状態にします。

- **タスクの設定:** 誰が何をいつまでにやるかを決めます。
- **プロジェクト運営委員会:** スポンサーや主要な幹部で構成され、方針、予算、大きな変更、エスカレーションされた問題を決める会議体。

運営委員会は、PM の権限を超える判断を扱います。変更管理委員会 (CCB) と役割が重なる場合もあります。間違えやすいのは、リファインメントを、スプリント計画と同じものとする選択肢です。リファインメントは、計画の前に、バックログを準備する作業です。

会議の種類は場面から選ぶもので、名前の印象では選べません。

場面	会議
顧客と開発者が同席して、画面の要求を決めたい	JAD
新製品への利用者の期待を、選んだ数人から聞きたい	フォーカスグループ
解決策の案を、批判なしで大量に出したい	ブレインストーミング
完成した機能を、顧客に実際に動かして見せたい	デモ (スプリントレビュー)
次のスプリントに入れる項目を、詳しく書き、見積もりたい	リファインメント
予算の大幅な増額を決めたい	プロジェクト運営委員会
毎日、障害物を共有したい	スタンドアップ

1.9-4 議題の設定と公開・会議の役割・タイムボックス

議題は、会議の前に決めて、公開しておく物で、当日に決めるものではありません。目的、話す項目、各項目の時間、必要な準備、決めること、を書きます。議

題を事前に配ると、準備した人が参加し、会議が短くなります。

役割は3つです。**ファシリテーター**は、議題どおりに進め、全員に発言の機会を与え、脱線に戻します。**書記**(スクライプ)は、決定事項とアクションを記録する役で、全発言の記録係ではありません。**参加者**は、会議の目的に必要な人だけで、準備をして臨みます。PM がファシリテーターと書記を兼ねないほうが、議論に集中できます。

タイムボックスは、時間の上限を先に決めて、その中で終える運営です。議題ごとにも、会議全体にも置きます。時間がきたら、議論を打ち切り、結論を出すか、別の場に回し、延長はしません。スクラムのイベントも、すべてタイムボックスです。間違えやすいのは、結論が出るまで時間を延ばす運営です。

1.9-5 アクションアイテム・議事録・フォローアップ

アクションアイテムは、会議で決めた、個別の行動です。必ず、担当者と期限を付けます。「検討する」のような担当者のない行動は、実行されません。

議事録は、日時、参加者、議題、決定事項、アクションアイテム(担当と期限)を記録した文書です。会議の後、早く、参加者と必要な関係者に配り、誤りがあれば直してもらいます。発言を全て書き起こすのではなく、決定と行動を残します。

フォローアップは、アクションアイテムの進捗を確かめ、次の会議の冒頭で状況を報告し、未完了のものを追う活動です。間違えやすいのは、会議を開いて議事録を出せば終わりとする選択肢で、実行の確認が済むまで終わりではありません。実行の確認までが会議運営です。

チームと資源の管理 [1.10]

1.10-1 組織構造

構造	指揮系統	PM の権限	資源
機能型	機能部門の長に報告する	小さい(調整役)	部門の人が兼務する
マトリクス型	機能部門の長と PM の両方に報告する	中くらい(弱い・均衡・強い幅がある)	部門から借りる
プロジェクト型 (Projectized)	PM に報告する	大きい	プロジェクトの専任で、終了後に解散する

マトリクス型は、2 人の上司を持つため、優先順位の対立が起きやすい構造です。PM の権限が強いほどプロジェクト型に近づき、機能部門の長の権限が強いほど機能型に近づきます。

場面では、「PM に予算も人事の権限もなく、各部門の長の許可が要る」なら機能型です。「チーム全員が PM の下に専任で配属された」ならプロジェクト型で、機能型やマトリクス型ではありません。間違えやすいのは、マトリクス型を、PM の権限が最も強い型とする選択肢です。最も強いのはプロジェクト型です。

1.10-2 資源のライフサイクル

資源のライフサイクルは、次の順です。

- 1. **取得:** 必要性を評価し(ニーズ評価)、購入、リース、採用、借用で得ます。
- 2. **保守:** 稼働を保つための点検、更新、パッチ、教育。

- 3. **廃棄・終了:** ハードウェアの**廃棄**では、データの消去、資産台帳からの除却、規則に従った処分を行います。ソフトウェアの**EOL** (サポート終了) は、更新が止まる日です。
- 4. **後継の計画:** EOL の前に、次の資源 (後継の機器、製品、担当者) を計画し、移行します。

EOL を過ぎたソフトウェアは、更新が出ず、脆弱性が放置されるため、リスクです (1.4-1)。ハードウェアを廃棄する時に、データを消去せず手放すと、情報が流出します。間違えやすいのは、取得の前のニーズ評価を飛ばして、先に購入する選択肢です。

1.10-3 資源の種類と重要度

軸	種類
性質	人的 (人)、物的 (設備、機器、施設)、資本 (資金)
所属	内部 (自社)、外部 (ベンダー、契約者)
専有	共有 (他の仕事と兼ねる)、専任 (このプロジェクトだけ)

重要度 (クリティカリティ) は、その資源がなければプロジェクトが止まるかどうか、代わりが見つかるかで決まります。1 人しかいない専門家や、調達に数か月かかる機器は、重要度が高い資源です。重要な資源は、早く確保し、代替を用意しておかなければなりません。共有の資源は、他の仕事と競合して使えない期間が出るので、事前に予約します。外部の資源は、契約の管理が要ります。

1.10-4 ギャップ分析

ギャップ分析は、あるべき姿と現状の差を見つける分析です。出題範囲は 3 つの観点を挙げます。

- **機能(フィーチャー)**: 求める機能と、現行の製品やシステムが持つ機能の差。
- **スキル**: 必要な技能と、チームが持つ技能の差。差があれば、教育、採用、外部の専門家で埋めます。
- **稼働率(Utilization)**: 必要な作業量と、使える資源の稼働の差。稼働率が 100% を超えれば不足、低すぎれば余剰です。

分析の後に、ギャップを埋める計画を立て、実行します。間違えやすいのは、ギャップの種類の取り違いです。「新しい言語を使える人がいない」はスキルのギャップで、「製品に必要な帳票機能がない」は機能のギャップです。

1.10-5 チームの成長段階と勢いの維持・フィードバック

チームは 5 段階を経て成長し、最初から機能するわけではありません。

段階	状態	PM の関わり方
形成(Forming)	顔合わせの段階。役割や目標が不明で、様子見	方向を示し、目標と役割を明確にする
混乱(Storming)	意見や進め方の対立が出る	対立の仲介、対話の促進
統一(Norming)	規範ができ、協力が整う	規範の定着を後押しする
機能(Performing)	自律して成果を出す	権限を委譲し、障害を除く
散会(Adjourning)	完了して解散する	成果の認知、教訓の記録、次の配属

順番は、形成、混乱、統一、機能、散会です。新しいメンバーが入ると、前の段階に戻ることがあります。

勢いの維持は、小さな成功を認め、障害を早く取り除き、進捗が見える化し、チームの士気を保つ活動です。**フィードバック**は、チームと個人の成果を、具体的に、早く、定期的に伝えることです。良かった点と、改善する点を、事実に基づいて話し、人格への批判にはしません。間違えやすいのは、対立が出た混乱の段階を、問題と取ることです。成長の通常の過程で、避けるのではなく導きます。

1.10-6 役割と責任(スポンサー・ステークホルダー・経営層・PMO・PM・プログラマネージャー)

役割	責任
スポンサー	資金と権限を提供し、プロジェクトを承認する。障害の解決を助け、最終的な責任を持つ
ステークホルダー	プロジェクトに影響を与えるか、影響を受ける個人や団体(顧客、利用者、経営層、規制当局など)
経営層(シニアマネジメント)	戦略と優先順位を決め、資源を配分する。方針を示す
PMO	組織のプロジェクト管理の標準、テンプレート、ガバナンス、支援を提供する部署
プロジェクト・マネージャー(PM)	プロジェクトの計画、実行、統制を行い、目標の達成に責任を持つ
プログラマネージャー	関連する複数のプロジェクトを、プログラムとして束ねて調整する

PMは1つのプロジェクトの納品に責任を持ち、**プログラクマネージャー**は複数のプロジェクト間の調整と、プログラムの効果に責任を持ちます。**PMO**は個別の案件を実行する部署ではなく、標準と支援の部署です(実行を担う形のPMOもありますが、基本は標準化と支援です)。**スポンサー**は日々の作業の管理者でなく、資金と権限の提供者です。間違えやすいのは、予算の承認や、PMの権限を超える障害の除去をPMが単独で担う選択肢です。スポンサーの仕事です。

1.10-7 役割と責任(プロダクトオーナー・スクラムマスター・プロダクトマネージャー・BA・SME)

役割	責任
プロダクトオーナー	製品の価値を最大にする責任者。バックログの内容と順序を決める[^c1-4]
スクラムマスター	スクラムを定義どおりに確立し、チームの効果に責任を持つ。障害を除き、チームを支える(命令する管理者ではない)[^c1-4]
プロダクトマネージャー	製品の市場、戦略、ライフサイクル全体(企画から終了まで)を担う。個別のプロジェクトを越える
ビジネスアナリスト (BA)	現場の課題とニーズを調べ、要求を引き出して文書にし、業務と技術の間をつなぐ
SME (専門家)	その分野に深い知識を持ち、助言する。要求や設計、受け入れの確認に関わる

プロダクトオーナーとプロダクトマネージャーは似て見えますが、前者はスクラムチームの中でバックログを預かる役割、後者は製品全体の戦略を担う役割です。間違えやすいのは、スクラムマスターをPMと同じ「チームの管理者」

とする選択肢で、両者は同じ役割ではありません。スクラムマスターは、作業の指示や進捗の管理をせず、スクラムが機能するよう支えます。

1.10-8 役割と責任(アーキテクト・開発者・テスター・エンドユーザー・コアチームと拡張チーム)

役割	責任
アーキテクト	システム全体の構造と技術の方針を設計する
開発者・エンジニア	設計に基づき、成果物を作る
テスター・QA 担当	品質を確かめ、欠陥を見つけ、報告する
エンドユーザー	成果物を実際に使う人。要求の出し手であり、受け入れテストの担い手

コアチーム(運用側のメンバー)は、プロジェクトの作業に、ほぼ専任で、日常的に関わるメンバーです。**拡張チーム**(機能部門のメンバー)は、機能部門に所属し、必要な時だけ、一部の作業で支援します。機能部門のメンバーは、拡張側の関係者として扱います。コアチームは、計画の会議と日々の作業に出ますが、拡張チームは、専門の助言や特定の成果物のために関わるだけで、日々は動きません。

間違えやすいのは、テスターの品質の責任を、「テスターだけが品質を負う」と取ることです。品質の責任はチーム全体にあり、テスターは確認の専門家です。

調達とベンダー選定 [1.11]

1.11-1 資源の調達方法

方法	内容	向く場面
構築 (Build)	自社で作る	独自の要求があり、技術を社内に持つ。競争力の源泉
購入 (Buy)	既製品を買い、所有する	標準的な機能で足り、長期に使う
リース	期間を決めて借りる (所有しない)	初期費用を抑えたい、陳腐化が早い機器
サブスクリプション・従量課金	使った分や期間に応じて払う	利用量の変動する、クラウドのサービス

選ぶ時の観点は、費用（初期と継続）、期間、柔軟性、保守、所有権です。**購入**は初期費用が大きい代わりに、長く使えば単価が下がります。**リースとサブスクリプション**は、初期費用が小さい代わりに、長く使うと、購入より総額が大きくなることもある。間違えやすいのは、リースを所有と取ることです。期間の終わりに返却します。

1.11-2 探索の文書 (RFP・RFB・RFQ・RFI)

買い手が、ベンダーに情報や価格や提案を求める文書です。何を求めるかで使い分けます。

文書	求めるもの	向く場面
RFI (情報提供依頼)	ベンダーの能力や製品の情報	選択肢を探る段階。応答は申込みではなく、契約にならない[^c1-9]

文書	求めるもの	向く場面
RFQ (見積依頼)	価格の見積り	仕様が明確な物品の価格を比べる。見積りは申込みでなく、買い手が受けても契約にならない[^c1-10]
RFB (入札依頼)	条件に合う入札 (価格)	仕様が明確で、価格が決め手。最低価格の適格な入札者に決める[^c1-11]
RFP (提案依頼)	解決策の提案 (技術と価格)	要求を満たす方法に幅があり、提案を比べて決める。評価基準と重みを示す[^c1-12]

流れは、RFI で市場を知り、要求が固まれば RFQ か RFB か RFP を出す形で、最初から RFP とは限りません。見分けは、「提案」なら RFP、「価格だけ」なら RFQ、「入札」なら RFB、「情報」なら RFI です。間違いやすいのは、仕様が曖昧で解決策に幅のある時に RFQ を出すことで、価格だけでは比べられません。

1.11-3 ベンダーの評価

評価の基本の軸は、**最良価値** (best value) と**最低価格** (lowest cost) です。最低価格は、要求を満たす提案のうち、最も安いものを選びます。最良価値は、価格と、技術、実績、品質などを合わせて、総合で最も得なものを選びます[^c1-12]。要求が明確で差が出にくい物品は最低価格、複雑なサービスは最良価値が向きます。

評価の手法は次のとおりです。

手法	内容
費用便益分析	費用と得られる便益を比べて選ぶ

手法	内容
市場調査	市場にどんなベンダーと製品があるか調べる
競合分析	ベンダー同士を比べる
資格 (Qualifications)	認定、免許、経験が要求に合うか
事前審査済み (Prequalified)	以前に審査を済ませた承認済みのベンダー一覧から選ぶ
デモ	実際の製品を動かして確かめる
技術的手法	提案された作り方や方法が、実現可能で適切か
能力 (物理と財務)	設備、人員、財務の体力があるか
照会先 (References)	過去の顧客に実績を問い合わせる

事前審査済みのベンダーから選ぶと、審査の手間と時間を省けます。間違えやすいのは、最低価格のベンダーを、機械的に選ぶ選択肢です。財務の体力や実績に問題があれば、価格が安くても、納品できないリスクは消えません。

1.11-4 契約の種類

種類	支払い	費用の超過リスク	向く場面
定額 (Fixed price)	合意した総額	売り手が負う	範囲が明確で、見積りが確かな時[^c1-13]
実費精算 (Time and material)	時間単価×実作業時間＋材料の実費	買い手が負う (上限額を置く場合がある)	作業の量や期間が事前に見積れない時[^c1-15]

種類	支払い	費用の超過リスク	向く場面
単価 (Unit price)	単位あたりの価格 × 実際の数値	数量の増減で総額 が変わる	数量が事前に確定 しない反復的な作 業
コストプラス (Cost plus)	実費 + 報酬 (固定、 または成果に応じ た額)	買い手が負う	範囲が不明確で、研 究的な作業[¹⁴]

定額は、買い手にとって費用が確定し、売り手が超過を負います。範囲が変わると、変更管理を通して価格を直します。コストプラスは、売り手に費用を抑える動機が弱いため、買い手が費用を監視します。実費精算は、定額とコストプラスの中間の性質で、時間単価は固定され、時間と材料が増えれば総額も増えます。

場面の問は、「範囲が明確で、予算が決まっている」なら定額、「範囲が決まらず、探りながら進める」ならコストプラスか実費精算で、逆の組み合わせは成り立ちません。範囲が固まった時点で、変更管理を通して定額へ切り替えることもできます。実費精算のまま続けなければならない決まりはありません。間違えやすいのは、定額契約を、買い手が費用のリスクを負う契約と取ることです。定額では、売り手が超過を負います。

1.11-5 契約の文書

文書	内容
MSA (基本サービス契約)	取引全体の共通の条件 (支払い、責任、秘密保持、紛争の解決) を、あらかじめ決める枠の契約。個別の仕事はこの枠の下で結ぶ
PO (発注書)	買い手が、品目、数量、価格を示して購入を指示する文書

文書	内容
TOR (業務範囲を定めた付託条件)	取り組みの目的、範囲、役割、条件を定めた文書
SOW (作業範囲記述書)	個別の仕事の作業内容、成果物、期間、受け入れの基準を書いた文書
NDA (秘密保持契約)	共有した秘密情報を、第三者に漏らさない約束
保守契約と保証	納品後の不具合の修正や更新の条件。保証は欠陥の無償修正、保守は有償の継続支援

関係は、MSA が枠で、その下に SOW と PO が個別に置かれる形です。NDA は、提案や見積りの段階から、情報を共有する前に結びます。SOW は成果物と受け入れの基準を明記するため、後の紛争の防止に役立ち、省けません。間違えやすいのは、MSA と SOW の取り違えて、MSA は全体の条件、SOW は 1 つの仕事の範囲です。

場面で組み合わせる例です。要求に幅があり、複数のベンダーの解決策を比べたい会社は、まず RFI で市場を調べ、RFP を出します。提案を、技術的手法、実績、財務の体力、価格で評価し（最良価値）、デモを経て 1 社を選ぶ。最低価格だけでは決めません。範囲がまだ固まらないなら、実費精算かコストプラスで契約し、範囲が固まれば、定額に切り替える選択もあります。取引全体の条件は MSA に置き、最初の仕事の内容と成果物は SOW に書きます。提案の前に、NDA を結んで、機密の要件を共有します。購入の発注は POで行います。この組み合わせが、場面の間の標準の流れだ。

サンプルはここまでです

続き(第2章～巻末)は、Project+ 問題集のご購入で、頻出度別ノート・暗記用ノートと合わせて3点すべてダウンロードできます。

SAMPLE

CompTIA Project+ 学習用テキスト(無料サンプル)

版

2026年10月版(CompTIA Project+ (PK0-005) 公式 Exam Objectives 準拠)

発行

IT資格道場(<https://www.shikaku-doujou.com>)

お問い合わせ

info@shikaku-doujou.com

CompTIA、Project+、および関連するロゴ・認定名は、CompTIA(The Computing Technology Industry Association, Inc.)の商標または登録商標です。記載されているその他の会社名・製品名・サービス名は、各社の商標または登録商標です。

本教材は公式シラバスの内容を基に作成したもので、CompTIAとの提携・承認関係はありません。

本文・図表・設問はすべてオリジナルの著作物です。無断での転載・複製・再配布・二次販売はできません。お気づきの点はinfo@shikaku-doujou.com までご連絡ください。