

JaSST'19 kansai 基調講演



# テスト運営を乗り切るための テストマネジメント入門 ～テストプロジェクトマネージャという働き方～

2019/7/5

バルテス株式会社

R&C部 部長  
石原 一宏

# 挨拶と今日のテーマ

# バルテスとは

商号	バルテス株式会社
所在地	大阪本社、東京本社、名古屋オフィス、福岡オフィス
資本金	9,000万円
設立	2004年4月19日
社員数	423名 (2019年3月末現在) (グループ連結・契約社員含む)
事業内容	ソフトウェアテストサービス コンテンツテストサービス テストコンサルティング
関連会社	バルテス・モバイルテクノロジー株式会社 VALTES Advanced Technology, Inc. (Philippines)

グループスローガン

**VALTES**

FOR QUALITY CONFIDENCE

お客様の品質に対する自信を  
揺るぎないものにしたい

# バルテストグループの提供するサービス



さまざまな角度から、**ソフトウェアの品質向上**を支援しています

## 石原 一宏（いしはら かずひろ）

### <所属>

バルテス株式会社 R&C部 部長 兼 上席研究員  
ソフトウェアテスト技術の研究開発、および教育セミナー・  
コンサルティングサービスを担当

### <大学との取り組み>

大阪大学 土屋達弘教授とテストツール『Qumias』を開発  
宮崎大学 非常勤講師として講義を担当

### <著書>

『いちばんやさしいソフトウェアテストの本』（共著、技術評論社）  
『ソフトウェアテストの教科書』（共著、ソフトバンク・クリエイティブ）

# こんな悩みはないですか？

普段、ソフトウェアテストの運営で、  
こんな**悩み**はないですか？

人・リソースが  
足りない

想定外のトラブルに  
悩んでいる

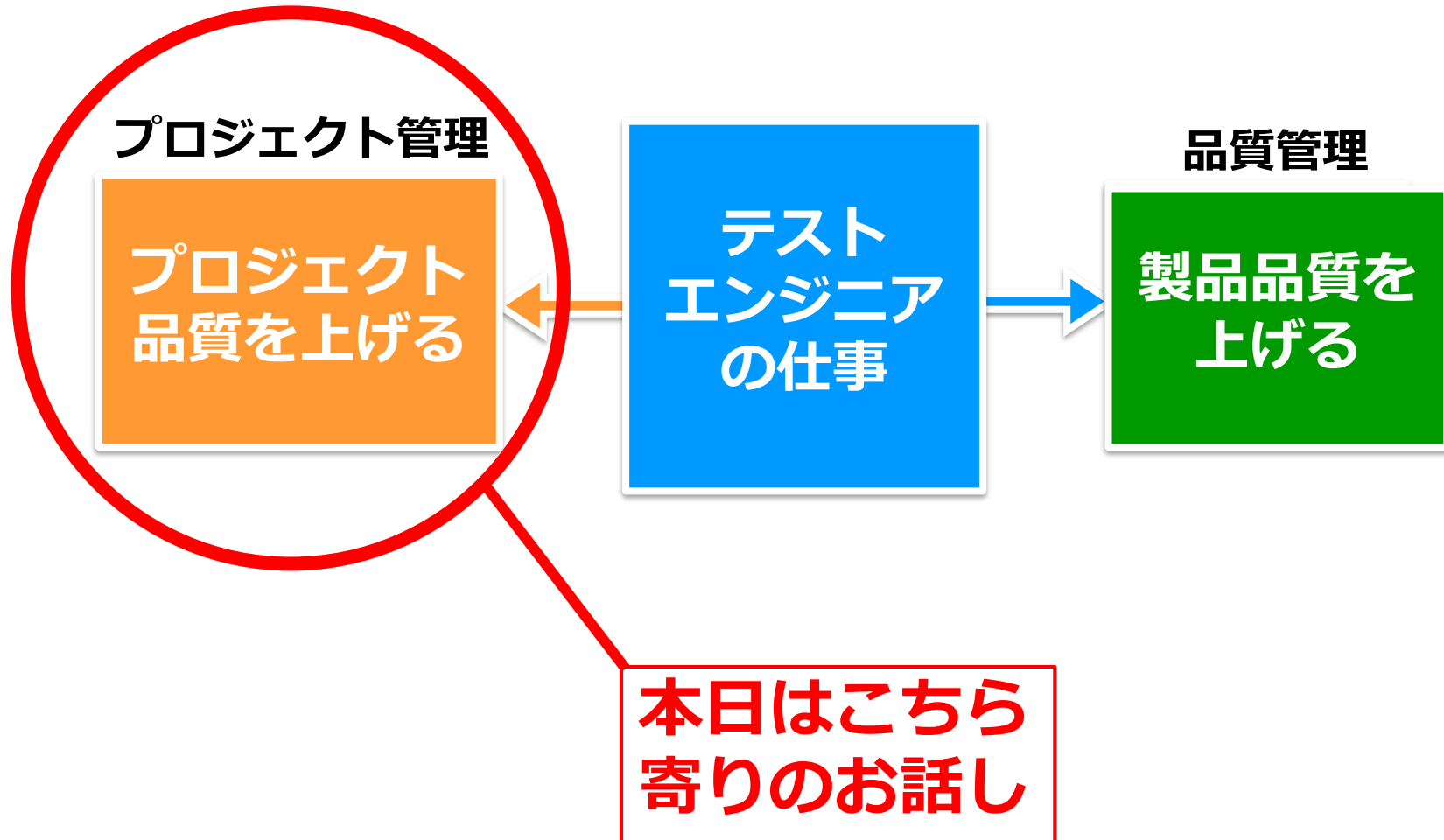
仕様がバラバラ。  
変更がたくさん

見積りが大変





# 「プロジェクト管理」と「品質管理」

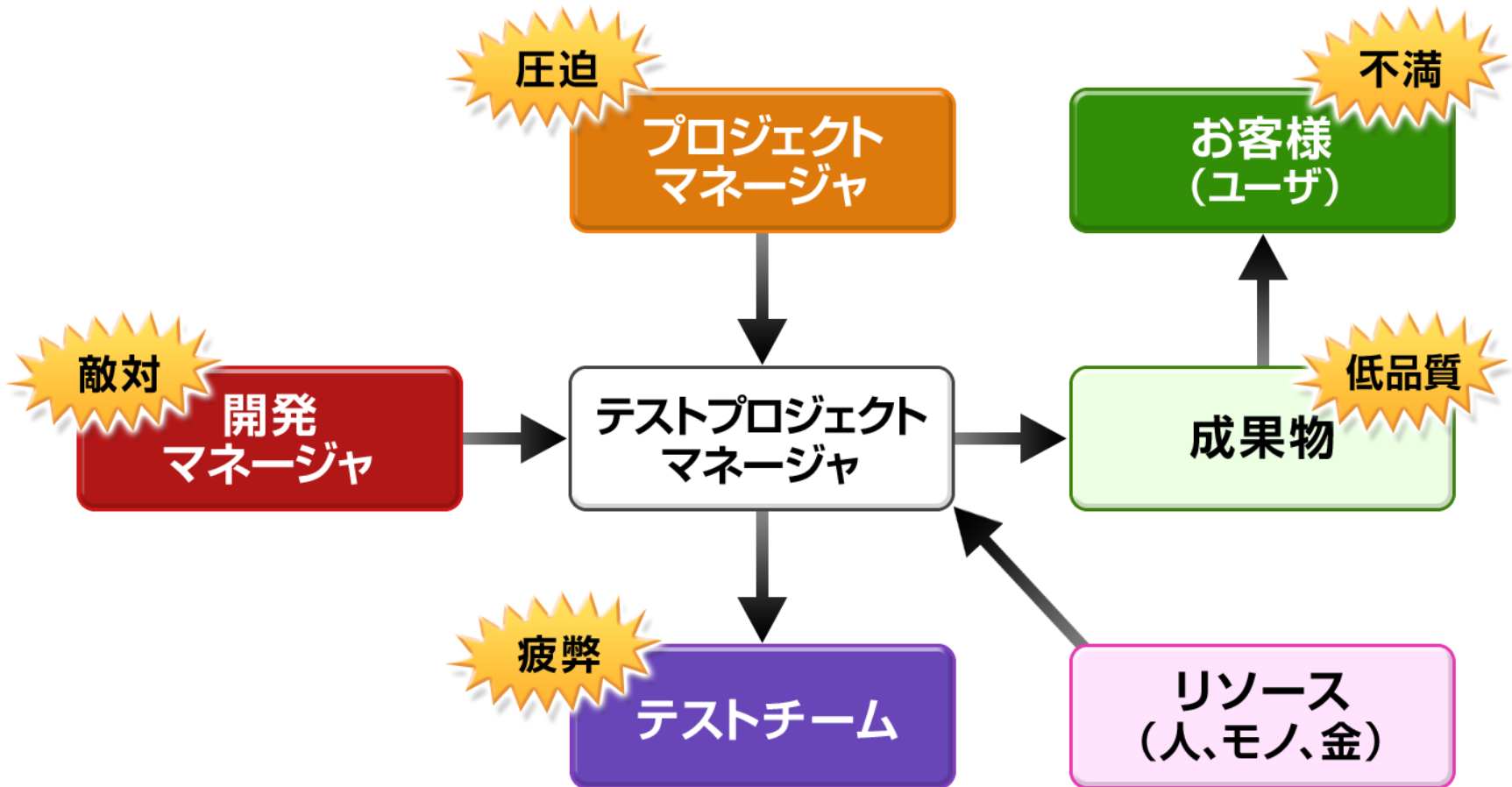


# テストにおけるProject Management



# テストプロジェクト管理の問題点 ①

## ▶ テストプロジェクトマネージャを取り巻く現状



## テストプロジェクト管理の問題点 ②

- ▶ なぜ、このような現状にあるのか
- ▶ テストプロジェクト全体を見渡せていない
  - テストプロジェクト全体を俯瞰した
    - 計画立案
    - 実施管理が不十分

# 本セミナーの目標

- ① ソフトウェアテストの全体計画立案ができるようになる
- ② テスト実施中の管理ポイントをおさえる

# テスト管理の必要性

## ▶ プロジェクトを成功させるための要因

良いテスト設計ができて、プロジェクトは成功しない

(理由)

プロジェクトには様々な制約があり、  
制約を守れていなければ成功とはいえないため



**「制約内」でプロジェクトを成功させるためには、  
「テスト業務運営スキル・管理スキル」が必要不可欠**

※ 制約 : ヒト/モノ/金/時間

# テスト管理とは

## ▶ テスト業務運営・管理スキルとは

語句の意味を確認してみると…

**運営**：機能を発揮させることができるように、  
組織をまとめて動かしていくこと

**管理**：ある規準などから外れないよう、  
全体を統制すること

(共に出典：大辞泉)



**ある規準にそってプロジェクトを統制し、  
動かしていくスキル**

# テスト運営前に収集すべき情報

# 全体計画立案の前に確認すること

## ▶ 計画立案前の確認ポイント

- テストの目的・制約
- リスクの識別（リスク管理）



# テストの目的・制約の確認 ①

テスト全体計画に入る前に、開発プロジェクト計画書を確認し、**テストの目的や制約**を明らかにする

計画書の項目	主な確認観点
開発の対象・目的	<ul style="list-style-type: none"> <li>新規開発か、派生開発か</li> <li>類似製品があるか、独自性高いか</li> </ul>
使われている技術	<ul style="list-style-type: none"> <li>既存の技術か、新技術か</li> </ul>
対象特有の専門知識	<ul style="list-style-type: none"> <li>どのくらいの学習時間が必要か</li> </ul>
セキュリティ、安全性のレベル	<ul style="list-style-type: none"> <li>セキュリティレベルは高いか、低い</li> </ul>
開発規模	<ul style="list-style-type: none"> <li>開発ボリュームはどのくらいか</li> </ul>
開発スケジュール	<ul style="list-style-type: none"> <li>テストの開始日・終了日はいつか</li> <li>製品のリリース日はいつか</li> </ul>
人員構成	<ul style="list-style-type: none"> <li>テストにかけられるリソースはどのくらいか</li> </ul>

## テストの目的・制約の確認 ②

### ▶ テスト全体計画書立案におけるインプット（全般）

#### ▶ 開発ドキュメント

- 開発プロジェクト計画書
- 過去の開発プロジェクト

#### ▶ テストドキュメント

- 過去のテストドキュメント

#### ▶ ユーザー情報

- 利用環境
- ユーザーの業務リテラシ
- ユーザーの製品リテラシ

#### ▶ コンプライアンス情報

- 法規
- 標準
- 規格

#### ▶ ヒアリング

- プロジェクトマネージャ
- 開発担当者
- 過去のテスト担当者
- 業務や製品の有識者

# リスク管理の概要 ①

## ▶ リスクとは

損失や被害など望ましくない出来事の起こる  
**可能性**のこと

➡ 品質目標に影響を与えるトラブルの「芽」

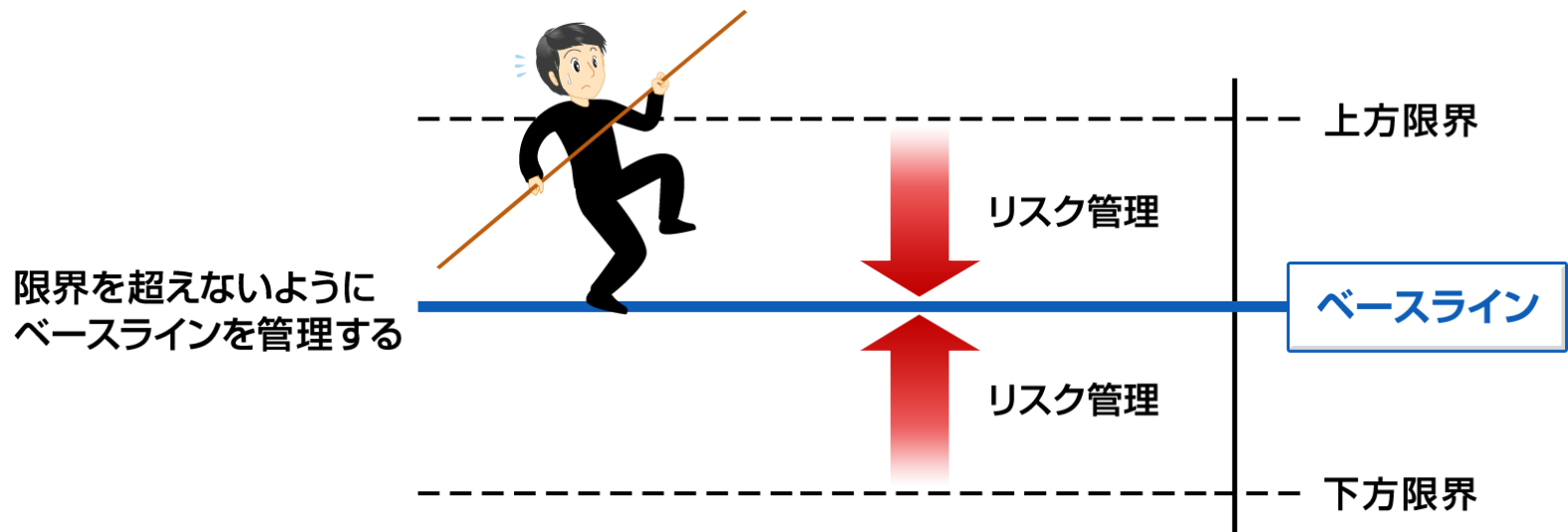
## リスクの例

- メンバーが揃わない
- メンバーが技術に不慣れ
- 仕様がなかなか決まらない

## リスク管理の概要 ②

### ▶ リスク管理の必要性

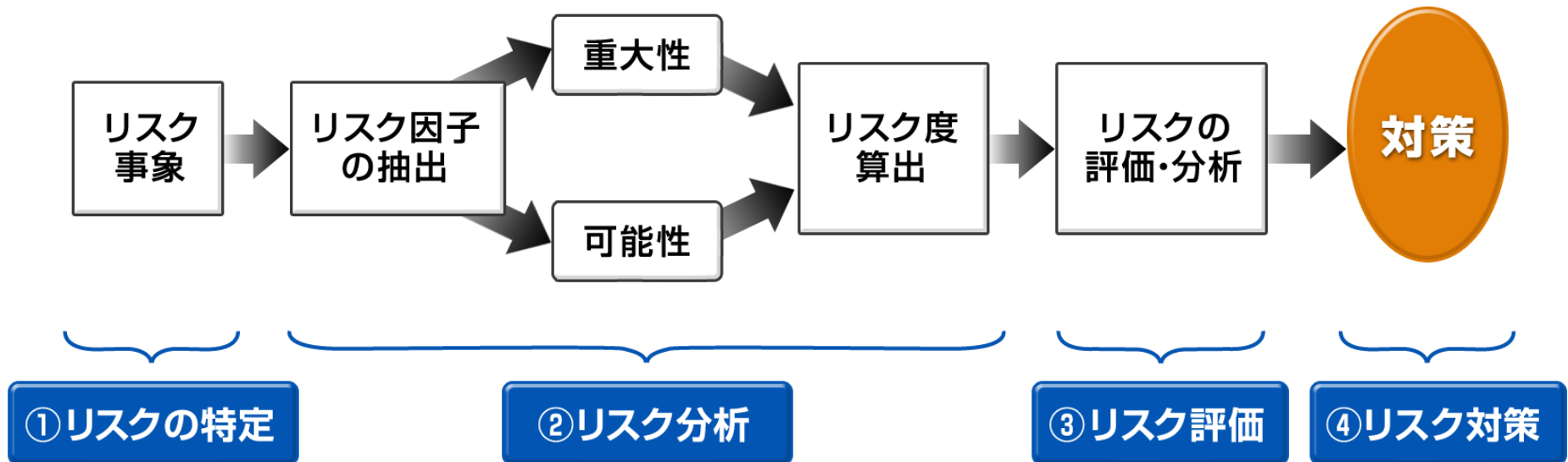
- プロジェクトを中断・頓挫から守る、外乱から守る
- 顧客に対して悪影響を与えないよう対策を講じる



テスト全体計画でリスクに対して  
『セーフティネット』を張る

# リスク管理とは

## ▶ リスク管理の流れ



# リスクの特定

## ▶ 予見できるリスクをできるだけ洗い出す作業

### ① どこにリスクがあるのか？

#### ✓ プロジェクトリスク と プロダクトリスク

- プロジェクト：人手不足、テスト機材不足
- プロダクト：原因不明の致命的な不具合

#### ✓ コントロール内のリスク と コントロール外のリスク

- コントロール内：過去のトラブル事例
- コントロール外：天災による開発遅延

### ② そのリスクにどのような意味があるのか？

# リスク分析

## ▶ リスク分析の主な内容

- リスクをもたらす要因（リスク因子）を洗い出す
- リスクの重要性、可能性、リスク度などを算出



# リスク評価

## ▶ リスク評価

➡ リスクがもつ重大性、可能性、リスク度などを  
評価・分析し、リスクを測定する行動を指す

※ FMEAの考え方が有効

### FMEAの例

①	②	③	④	⑤		⑥	⑦	⑧	⑨
機器名	機能	故障モード	原因	影響		検出法	故障モード発生頻度	致命度	対策
				1次影響	2次影響				

IEC FMEA規格

### リスク評価の例

機能	リスク評価
不完全燃焼検知	6
ガス漏れ検知	6
水漏れ検知	5
水量調整	5
ガス調整	4
温度調整	4
騒音	3
リモコンの表示	2
端末機器との通信	1

足切り線

# リスクへの対応策

## ▶ リスク評価を基に、リスクによるトラブルを未然に防ぐための対策を講じる

リスクの例	対策の例
メンバーが揃わない	<ul style="list-style-type: none"><li>• 社内外にメンバー集めをアプローチする</li></ul>
メンバーが技術に不慣れ	<ul style="list-style-type: none"><li>• スキルのあるメンバーを探す</li><li>• メンバーをトレーニングする</li></ul>
仕様がなかなか決まらない	<ul style="list-style-type: none"><li>• 協議の場を持つ</li><li>• スケジュールを見直す</li></ul>

# テストプロジェクトを計画・管理する際の「肝」

# 目を光らせるべきマネジメント領域 ①

## PMBOK®Guide（第6版）の知識エリア

プロジェクト統合マネジメント

プロジェクト・スコープ・マネジメント

プロジェクト・スケジュール・マネジメント

プロジェクト・コスト・マネジメント

プロジェクト品質マネジメント

プロジェクト資源マネジメント

プロジェクト・コミュニケーション・マネジメント

プロジェクト・リスク・マネジメント

プロジェクト調達マネジメント

プロジェクト・ステークホルダー・マネジメント

"Project Management Institute, A Guide to the Project Management Body of Knowledge, (PMBOK® Guide) – Sixth Edition, Project Management Institute Inc., 2018"

# 目を光らせるべきマネジメント領域 ②

## テスト活動

テストプロジェクトの  
マネジメント

リスクの特定および  
分析の活用

テスト計画の  
作成と実装

テスト活動のモニタ  
リングとコントロール

コスト・メリットを  
基にしたテスト活動の  
ビジネスケースの提案

## ステークホルダ連携

チーム内やステークホルダとの  
コミュニケーションの確立

テスト状況の  
評価と報告

## テストプロセス・テストチーム改善

必要な  
スキルの開発

必要な  
人材の調達

プロセスの  
改善

参考) 『Advanced Level シラバス日本語版 概要 Version2012.J01』 p.9

# ソフトウェアテスト全体計画の5W2H

## ✓ When いつ

- 開始条件と終了条件
- スケジュール

## ✓ Where どこで

- 作業場所

## ✓ Who 誰が

- 人員構成

## ✓ What 何を

- テスト対象の確認
- テスト対象範囲・非対象範囲

## ✓ How どのように

- テスト環境
- コミュニケーションルール

## ✓ How Much いくら

- 工数見積

## Why なぜ

他の4W2Hで定めた事柄の  
理由・根拠などを考慮

# テストの開始と終了（When） ①

## ▶ 開始条件

- テストプロジェクトを開始してよいか
- テストフェーズを進めてよいか

## ▶ 終了条件

- 次のテストフェーズに移ってよいか
- テストプロジェクトを終了してよいか

## ▶ 中断条件と再開条件

- 中断条件に該当する事象が発生したか
- 再開条件が満たされているか

### 明確な条件を記載

- プロセス監査的な条件
- 品質メトリクスによる条件



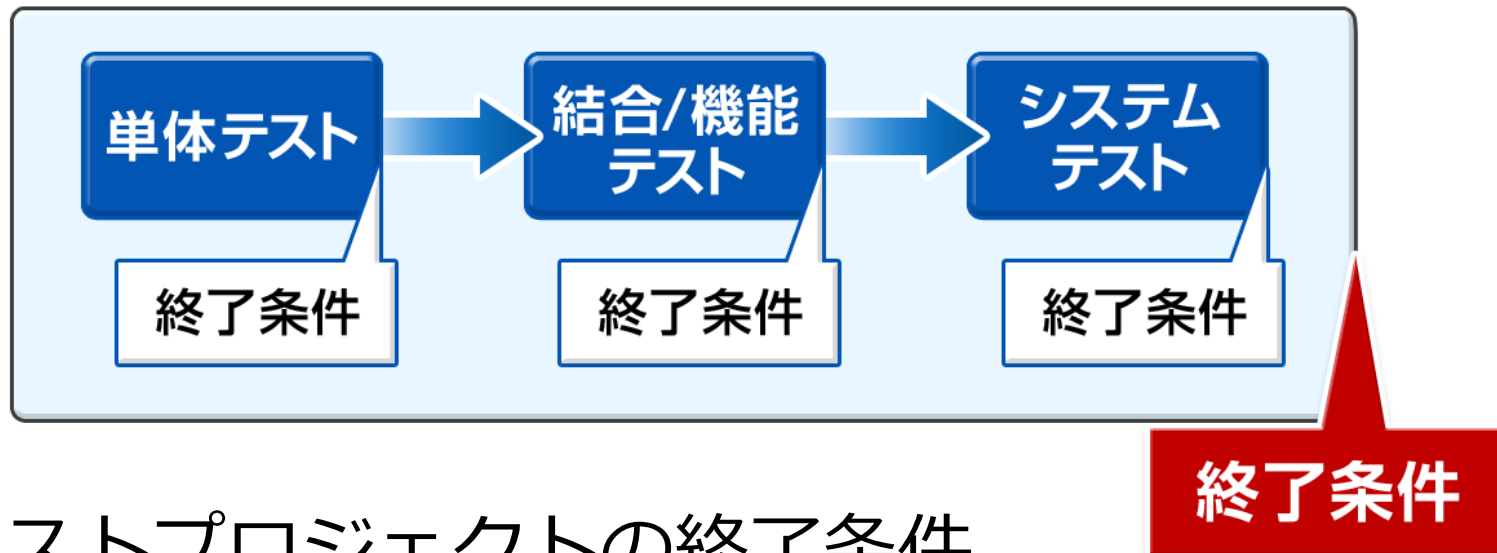
品質目標にしたがって決定



# テストの開始と終了（When） ②

## ▶ テストプロジェクトとテストフェーズの終了条件

テストプロジェクト



## ▶ テストプロジェクトの終了条件

- テスト対象が求められる品質目標を満たしていること
- 各種のメトリクスが満たされていること など

# テストの開始と終了 (When) ③

## ▶ テストフェーズの終了条件

### ✓ 単体テスト

- 計画で策定したパス網羅率を100%達成

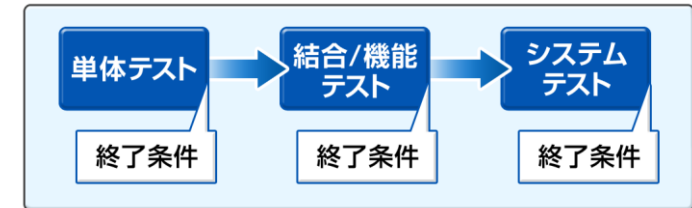
### ✓ 結合/機能テスト

- 全機能のテスト終了
- 定量データ：消化項目数、不具合検出件数

### ✓ システムテスト

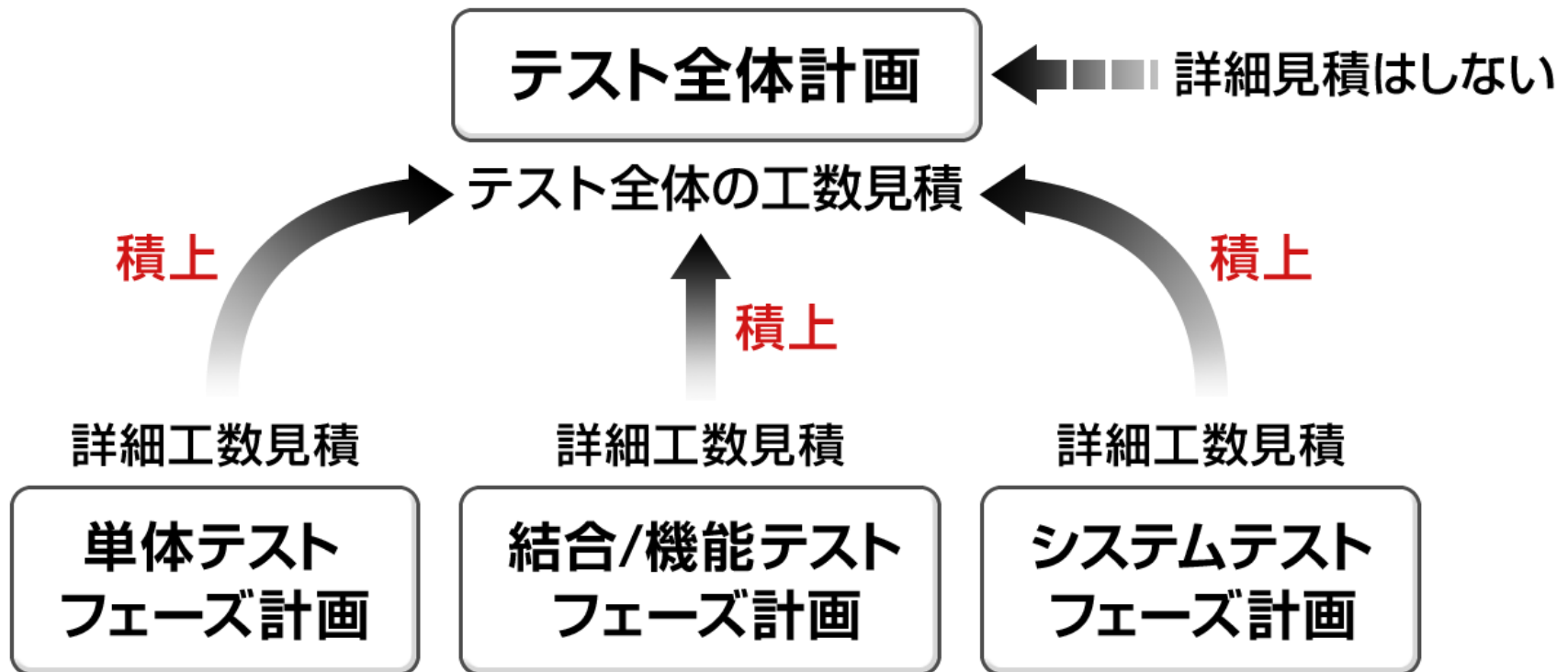
- 品質目標に見合うテストが実施済みされたこと
- 品質目標に見合う性能が確認されていること
- 検出した不具合がすべて対応済みであること

テストプロジェクト



# 工数見積 (How Much)

- ▶ テスト全体計画では**詳細見積**は行わない
- ▶ 詳細見積は個別のテストフェーズの見積を積算する



# スケジュール (When) ①

## ▶ スケジュール表

タスク		工数		時期		
タスク	期間	2008年4月	2008年5月	2008年6月		
単体テスト	20日	<div></div>				
結合/機能テスト	15日	<div></div>	<div></div>			
システムテスト	30日		<div></div>	<div></div>		
テストサマリレポート作成	5日			<div></div>		

**WBS**  
(Work Breakdown Structure)

## スケジュール（When） ②

### ▶ スケジュール作成上のポイント

- ✓ 品質を犠牲にしないこと
- ✓ プロジェクト全体のマイルストーンにあわせて考える
- ✓ 期限から逆算して考える
- ✓ 経験ベースで考える
- ✓ 実作業以外の事項を考慮する（調査・準備、休暇など）
- ✓ 不具合の修正時間を見込んでおく
- ✓ テスト用のビルドリリースのタイミングを考慮する
- ✓ スケジュール上のクリティカルパスに注意する

# 人員構成（Who） ①

- ▶ テストチーム内を どのような**組織編成**にし、  
どのような**人員構成**にするか
- ▶ 目的や工程に合わせて**人員構成**を組み替える  
あるいは**必要な教育やトレーニング**を計画する



## 人員構成（Who） ②

### ▶ テストに必要なテストチーム外の人員

例：

- テストチームの成果に対するレビュアー（外部有識者）
- 特定技術のスペシャリスト
- ファシリティ管理者

時期と作業規模を明らかにして  
打診できるようにしておく



## 人員構成（Who） ③

### ▶ 人員構成の不備による問題

例：

- スキルに偏りがあり、テストできない部分がある
- マネージメントできる人がいないので、進捗が遅れる  
問題を抱え込んでしまう性格のメンバーが多く、
- 報告が遅れる  
有識者による外部レビューが行われず、
- 要求を誤解したテストを実施してしまう

# テスト対象の確認 (What)

## ▶ テスト対象について理解する

### ✓ 開発プロジェクト計画を確認する

- どのような目的、経緯で開発されるのか
- 最も重要な品質特性は何か（品質の優先順位）

#### できていない場合

- 何をテストすればよいのかわからない
  - 品質の優先順位がわからない
- ➡ 結果として、適切なテストができない

# テスト対象範囲・非対象範囲（What）

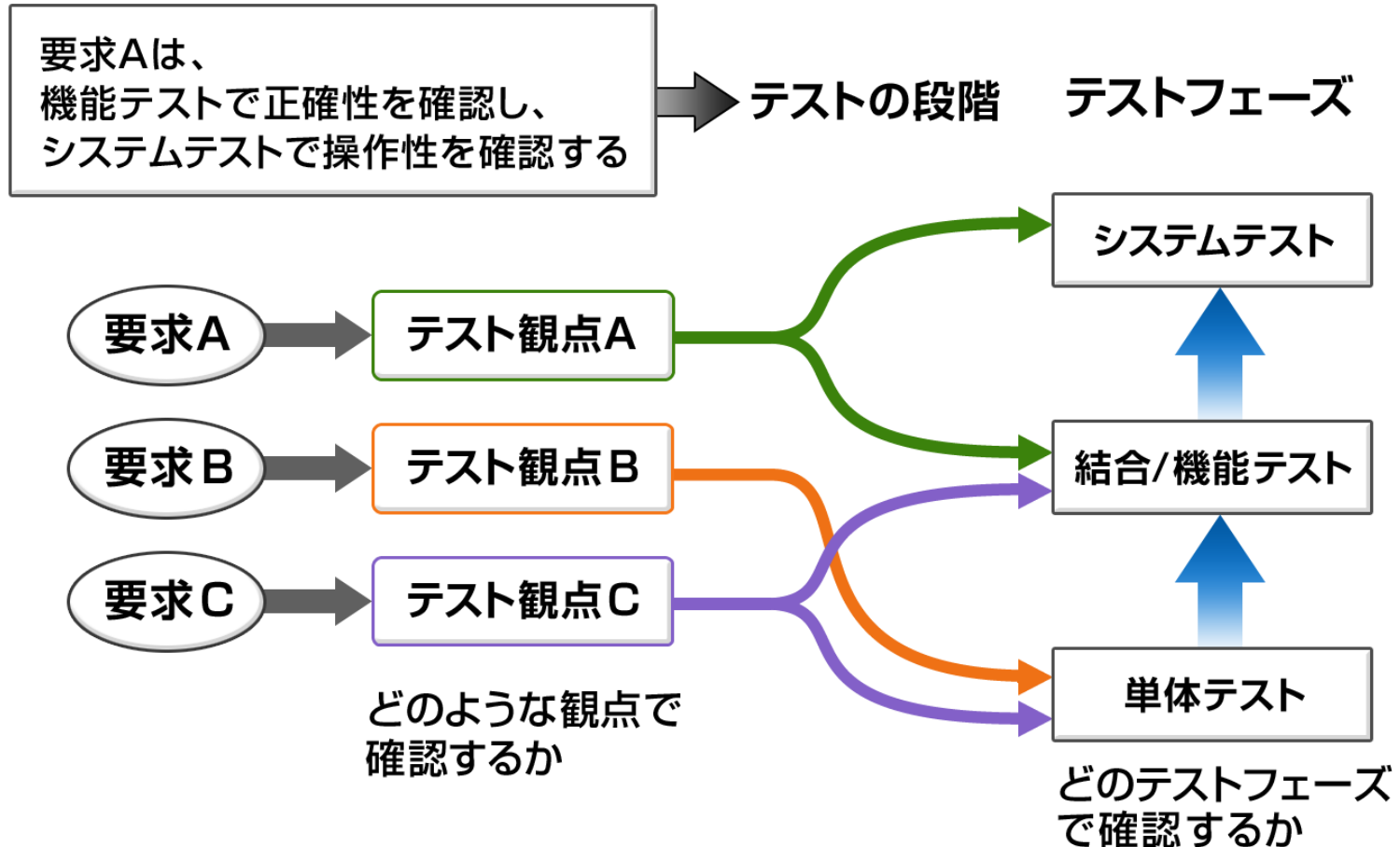
## ▶ テストの対象・非対象を明確にしておく事項

- 機能
- 他のシステムとの連携範囲
- 品質特性

できていない場合

- テストした／しないのトラブルが発生
  - ➡ 非対象範囲は書かないのではなく、  
書いた上で非対象であることを明示する

# テストフェーズと段階の構築 ①



品質はテストフェーズが進むにつれ、段階的に確保されていく

# テストフェーズと段階の構築 ②

## テスト観点リスト

### ▶ ロジック

- ロジック
- 条件分岐

### ▶ インタフェース

- 上位インタフェース
- 下位インタフェース
- モジュール間インタフェース
- 疎通
- 接続
- 切断
- 中断

### ▶ 機能性

- 基本機能
- 基本設定
- 入力バリエーション
- 出力バリエーション
- 設定バリエーション

### ▶ 障害

- 障害時動作
- 障害復旧

### ▶ 業務

- 初期移行
- 代表業務
- 業務フローバリエーション
- システム起動
- システム終了

### ▶ 環境

- 構成
- 互換性
- 両立性

### ▶ パフォーマンス

- 負荷
- セキュリティ
- 性能測定
- リソース測定

### ▶ 使用性

- 習得
- 魅力
- 効率

## ▶ テスト全体計画

👉 各テストフェーズで  
何をするのか、しないのかを明らかにする

単体テスト、結合テスト : 開発者側で行う  
機能テスト、システムテスト : 評価側でやる

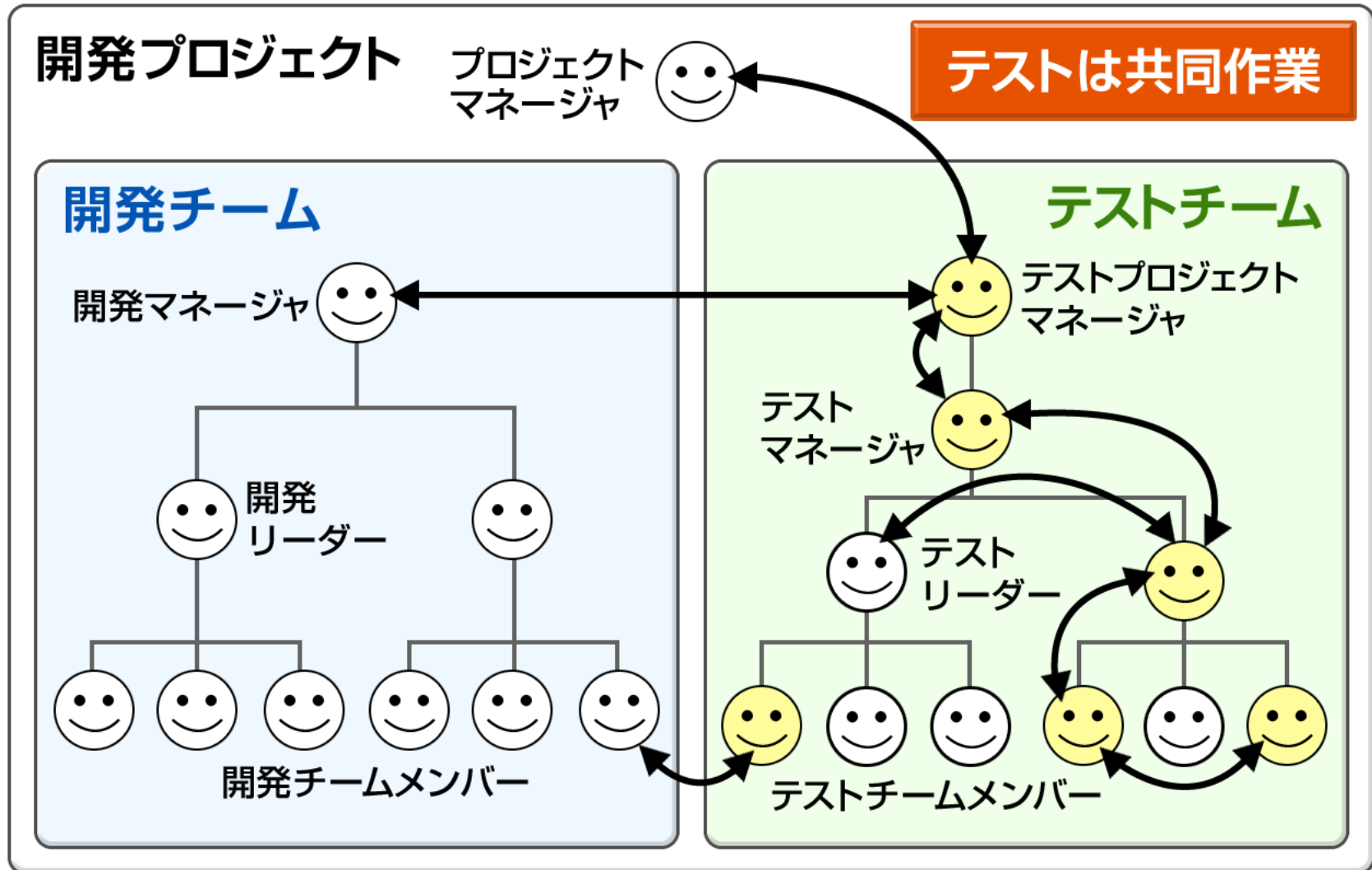
プロセス標準があったとしても、  
プロジェクトごとに確認することが大切！

# テスト環境（How）

## テスト環境を計画する際のポイント

- 可能なかぎり**実利用環境と同じ**にする
- **テスト専用**の環境を用意する
- 開発チームから**提供**されるのか、  
テストチームで**調達**するのか決める
- **利用可能期間**を明確にする

# コミュニケーション・ルール（How） ①





## コミュニケーション・ルール（How） ②

- ▶ コミュニケーション手段を決めておく
  - ➡ 組織体制と同時に確認する
- ▶ コミュニケーションで利用する情報の種類
  - 進捗情報
  - 不具合情報
  - 品質情報
  - トラブル（リスク）に関する情報

# コミュニケーション・ルール（How） ③

## ▶ ルールとして決めておくこと

- 経路 （誰から誰へ）
- タイミング （即時、日次、週次、マイルストーンごと）
- 手段 （メール、電話、ドキュメント、会議）
- 責任 （承認者、開始・中止などの判断）

### 決まっていない場合

- 誰に連絡すればよいか分からず対応が遅れる
- 連絡がとれず対応が遅れる

# テストドキュメントの定義・設定 ①

▶ 「なぜテストドキュメントを作るのか？」を考えよう

## ✓ テストの目的達成のため

- どのようなドキュメントでテスト内容を定義するのか
- どのようなドキュメントでテスト結果を残すのか
- そのドキュメントは誰に提出するのか

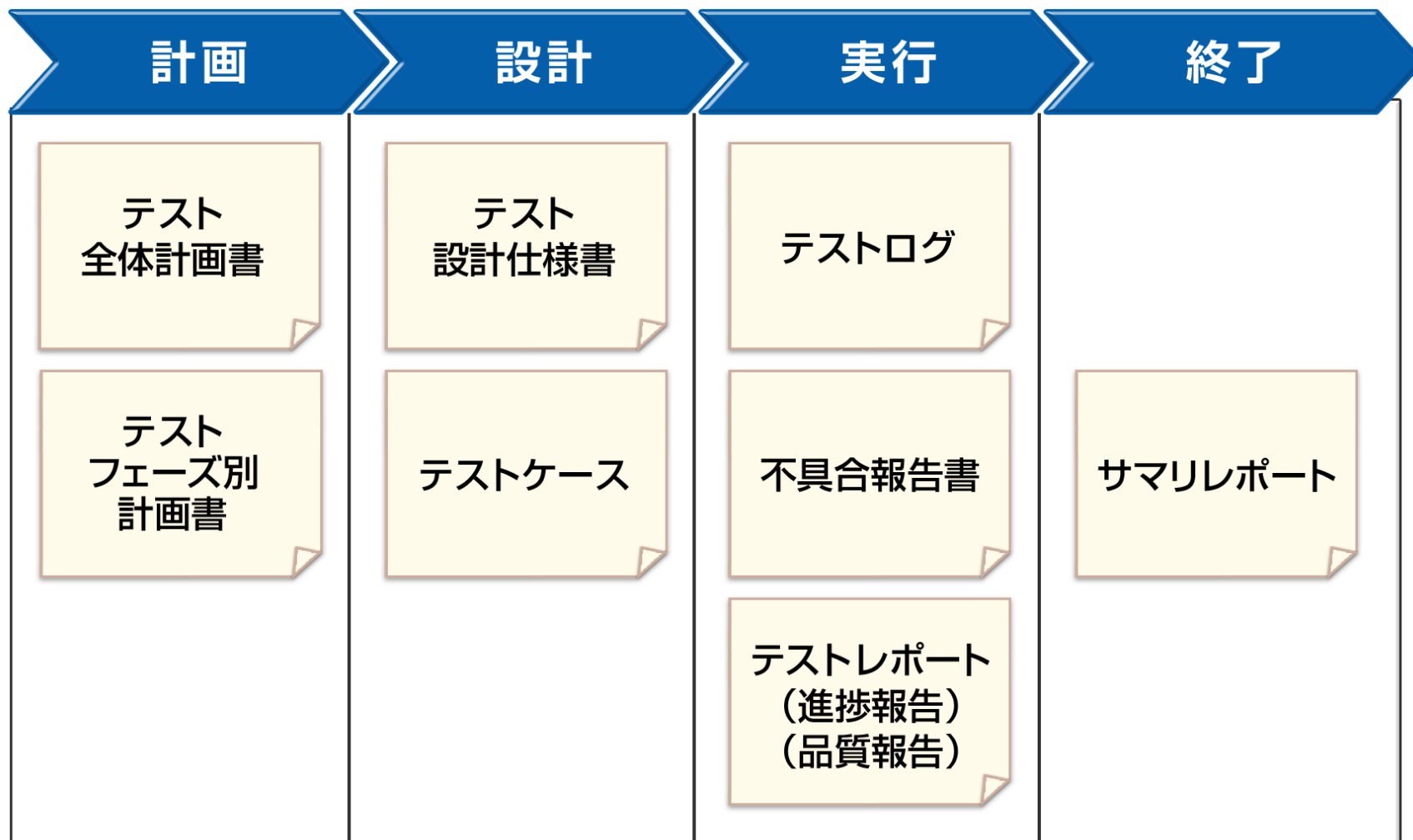
## ✓ サマリレポート作成のため

- サマリレポートに必要な情報を取得する
- 必要な情報の収集漏れを防止

## ✓ プロジェクトの振り返りのため

- プロダクト品質やプロジェクト品質を省みる

# テストドキュメントの定義・設定 ②



# 事例紹介を中心に

# 事例 1：概要設計書の品質にばらつき

## 課題

- 開発ベンダの概要設計書の品質にばらつき
- クライアントとの対面レビューを繰り返し実施
- スケジュール遅延発生

## 対応策

- クライアントの指摘をモニタリング
- 開発ベンダ側のレビューで発見すべき指摘事項を集計

## 結果

- モニタリング結果を開発ベンダへ申し入れ
- クライアントレビューの指摘件数激減！！

## 事例 2：テスト管理プロセス・方法が知りたい

### 課題

- これまでは開発ベンダの納品物を受け入れるのみ
- クライアント主導で結合テスト・システムテストの実施を計画
- 総合的なテストを計画、実施、管理を行ったことがない

### 対応策

- テスト管理標準を作成し、提供（プロジェクト参画後、1週間で）

### 結果

- ガイドライン資料として定着！！

## 事例3：システムテストの工数見積もり

### 課題

- システムテストのインプットとなる要求定義書がない
- 大規模すぎて見積もるのに時間がかかる
- そもそもシステムテストの見積もりをどうやればいいのか

### 対応策

- 超概算見積り⇒検証を繰り返す
- トップダウン的な方法（規模やIPA資料から見積り）
- ボトムダウン的な方法（業務フローから機能、確認項目を仮置き）
- 業務フローサンプル1本から詳細見積りを作成し、検証を繰り返す

### 結果

- 大幅な工数差異によるリスク低減！！



## 事例 4 : 課題管理の有効活用

### 課題

- 頓挫プロジェクトのリベンジで、失敗が許されない
- 品質強化への要求が高いが、品質向上へのノウハウは少ない
- プロジェクト規模が大きく、運営に不安がある

### 対応策

- 日常的に課題管理表へ起票し、週次定例でWBSと合わせてすり合わせ
- 4ヶ月で100件（1日約1件）の起票
- テストチームだけでなく、プロジェクト全体の運営に関わる課題も一覧化

### 結果

- PJ会議が課題やアジェンダに沿って進行  
⇒プロジェクト運営の効率化！！

最後に

テストプロジェクトにおいて  
周囲との**良い関係を実現**するために

テストプロジェクト

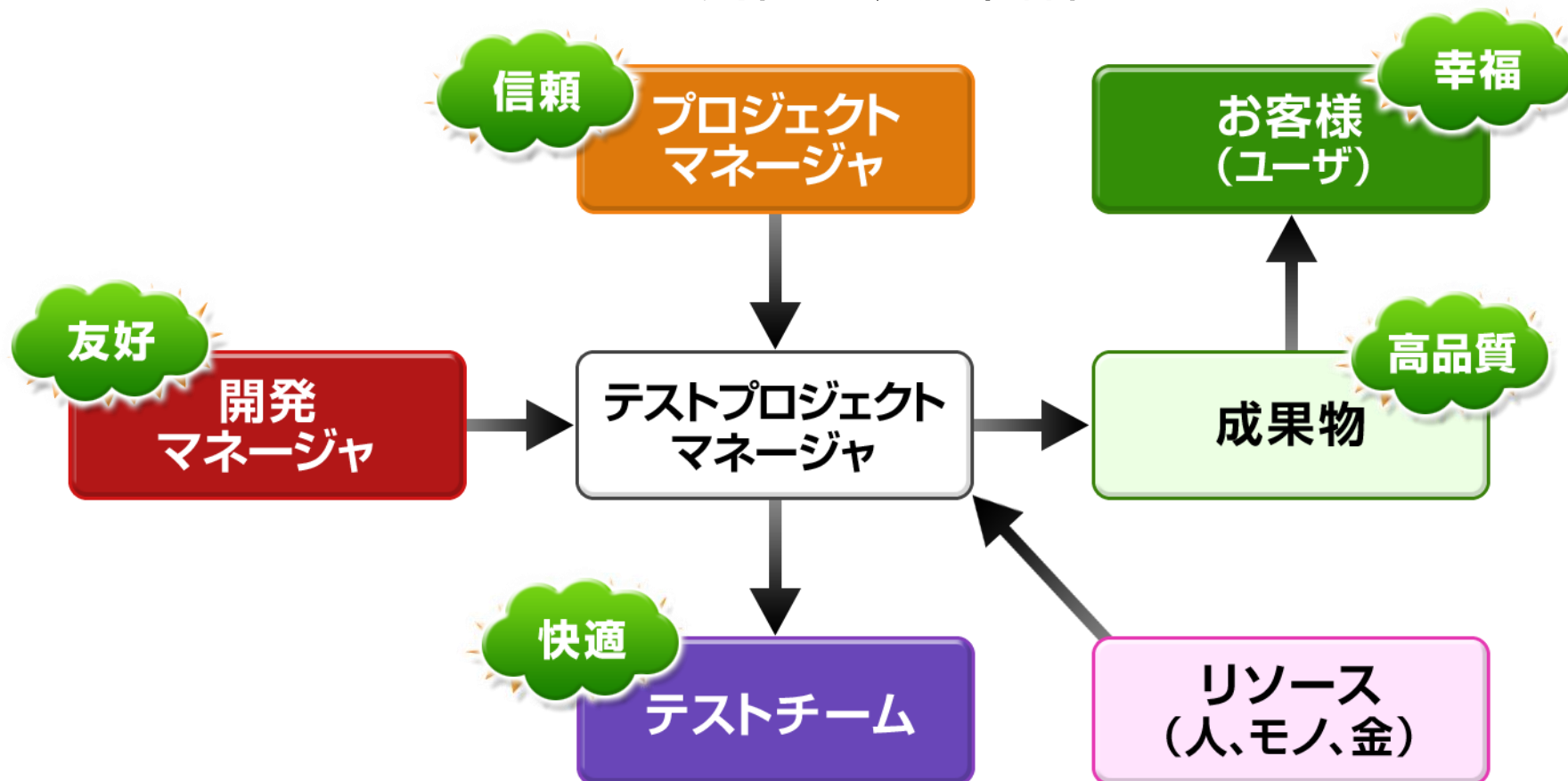
**計 画**  
**モニタリングとコントロール**  
**振返り**



**改善**

# テストプロジェクトマネージャが目指すもの ②

テストプロジェクトマネージャと周囲の良い関係



一歩ずつ、確実に  
向上させていきましょう。

ご清聴ありがとうございました

