

日本XPユーザグループ関西 (XPJUG関西) 主催  
「XP祭りin関西2012」

アジャイル開発における計画と管理  
～アジャイルのプロジェクトマネジメント～

2012年4月7日(土) 大阪市 福島区民センター

株式会社システム インテグレータ  
代表取締役 社長 梅田弘之



設立	:1995年3月14日
本社	:埼玉県さいたま市南区沼影1-10-1 ラムザタワー7階
大阪支社	:大阪府大阪市中央区今橋2-1-10 ダイセンビル2階
社員数	:110名(役員7名含む)

## ■ 創業のバックボーン

- ・ 1992年に日本初のERP「ProActive」を企画・開発した2人が独立

## ■ 沿革 ■

- ・ 1995年03月 株式会社システムインテグレータ創業開始
- ・ 1996年03月 日本初ECサイト構築パッケージ「SI Web Shopping」をリリース
- ・ 1997年08月 開発支援ツール「SI Object Browser」をリリース
- ・ 2004年05月 日本初のWeb-ERP「GRANDIT」を開発
- ・ 2006年07月 大阪支社開設
- ・ 2006年08月 MIJS設立
- ・ 2006年12月 東証マザーズ上場
- ・ 2008年02月 「SI Object Browser」「同ER」の中国語版を中国で販売開始
- ・ 2008年05月 「SI Web Shopping 中国語版」を中国で販売開始
- ・ 2008年11月 総合プロジェクト管理システム「OBPM」をリリース
- ・ 2009年06月 「OBPM」を中国現地企業に販売
- ・ 2010年04月 「OBPM」のクラウド提供を開始



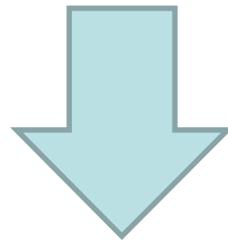
# アジャイル開発を実践する際の心構え

# 我々はなんのためにやっているか

**【サッカーをやる目的】勝つこと！（サッカーのルールの中で）  
そのために必要なこと・・・個人技とチームプレー、そして戦術。**

**【ソフト開発の目的】良いものを作ること（コストと納期の制約の中で）  
そのために必要なこと・・・アジャイル手法とプロジェクト管理**

**アジャイル開発自体を目的とするな！**



**全てのプラクティスを忠実に実行するのが  
良いわけではない**

## 【犬小屋を作るなら】

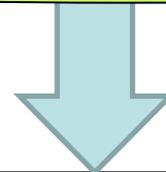
完成イメージ図を元に、作りながらアイデアを盛り込みながら作成



## 【マンションを作る場合】

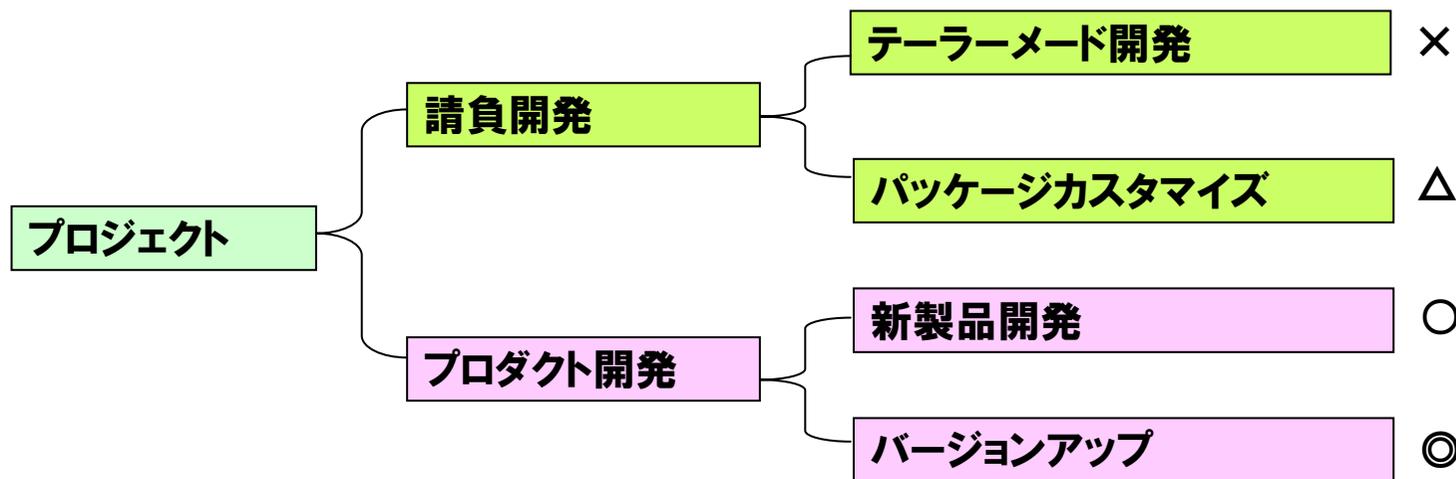
意匠図、構造図、工事仕様書、構造計算書など、さまざまな設計書にもとづいて建築

**ウォーターフォールを敵視するな！**

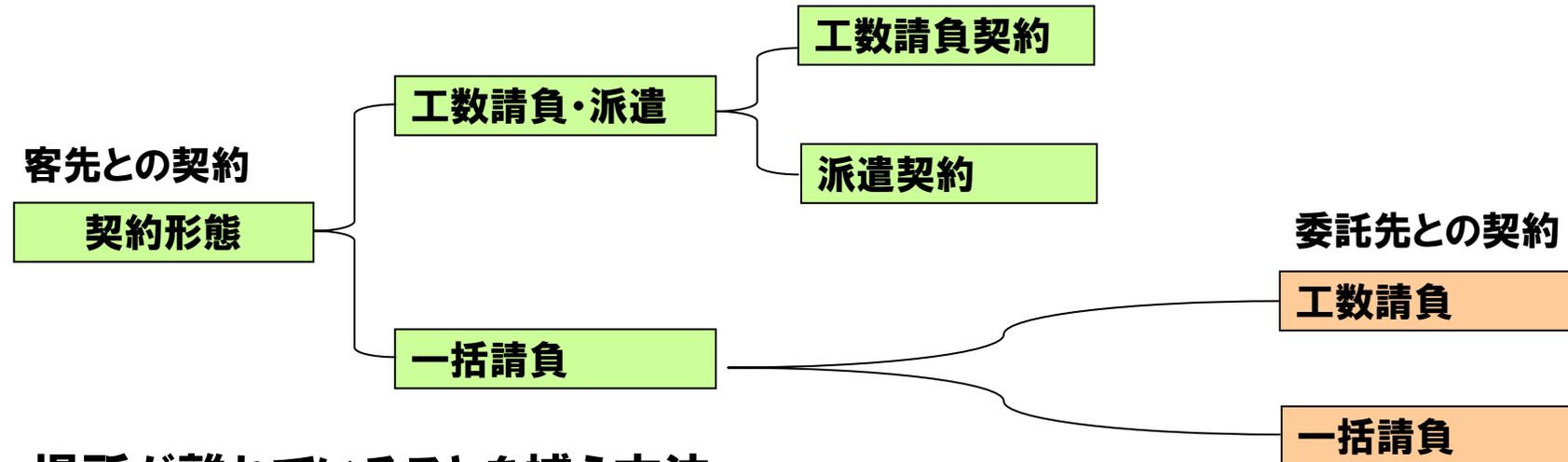


**ある規模のシステムを構築するには、ウォーターフォールとアジャイル、両方の良さを生かす必要がある**

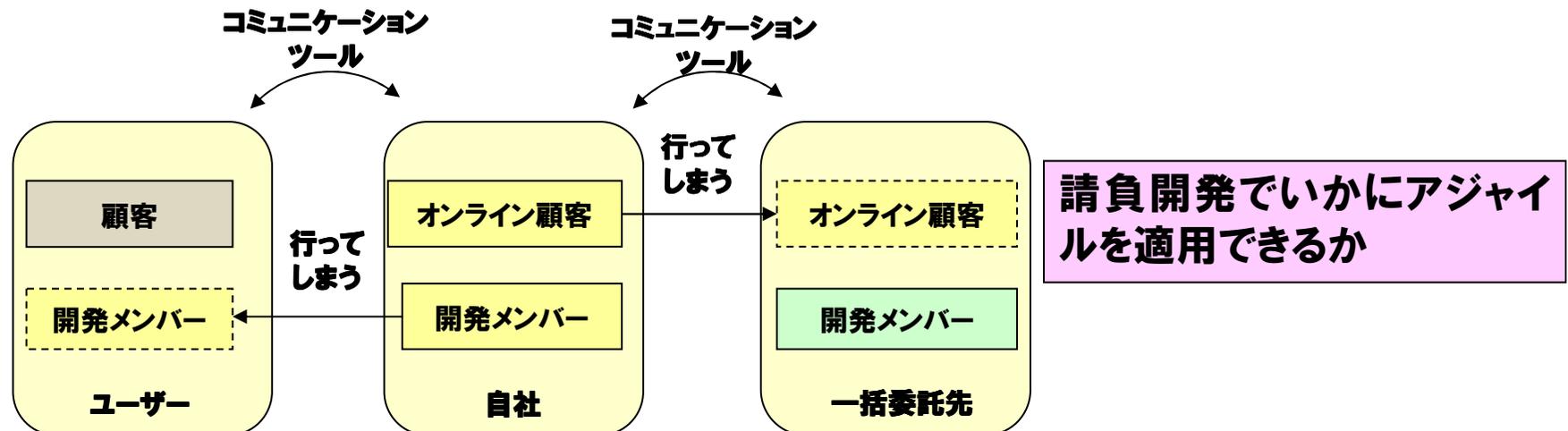
- アジャイル開発の最大の利点・・・「満足度の高いシステムが出来上がる」  
→「前向き品質」の必要なプロダクト開発に向いているが、  
請負開発でも取り入れる工夫をして行こう。
- アジャイルで請負開発を行う際の課題・・・
  - (1) 最初にスコープを契約で取り決めて、どこかで仕様を確定しないと、  
いつまでも変更作業が続くリスクがある。  
→客先常駐型でアジャイルを敢行する。  
→ウォーターフォールで仕様決めの際にモックを作ったりして確認。
  - (2) 忙しい顧客、各部門にまたがる顧客にオンライン顧客になってもらったり、  
開発途中のチェックを頻繁に頼みづらい。  
→最初にアジャイル開発で行うことをきちんと説明して時間を確保してもらう。



- ・アジャイル開発がやりにくい契約・・・「一括請負契約」  
 →アジャイル開発は“変更”が前提なので、スコープの確定が難しい。



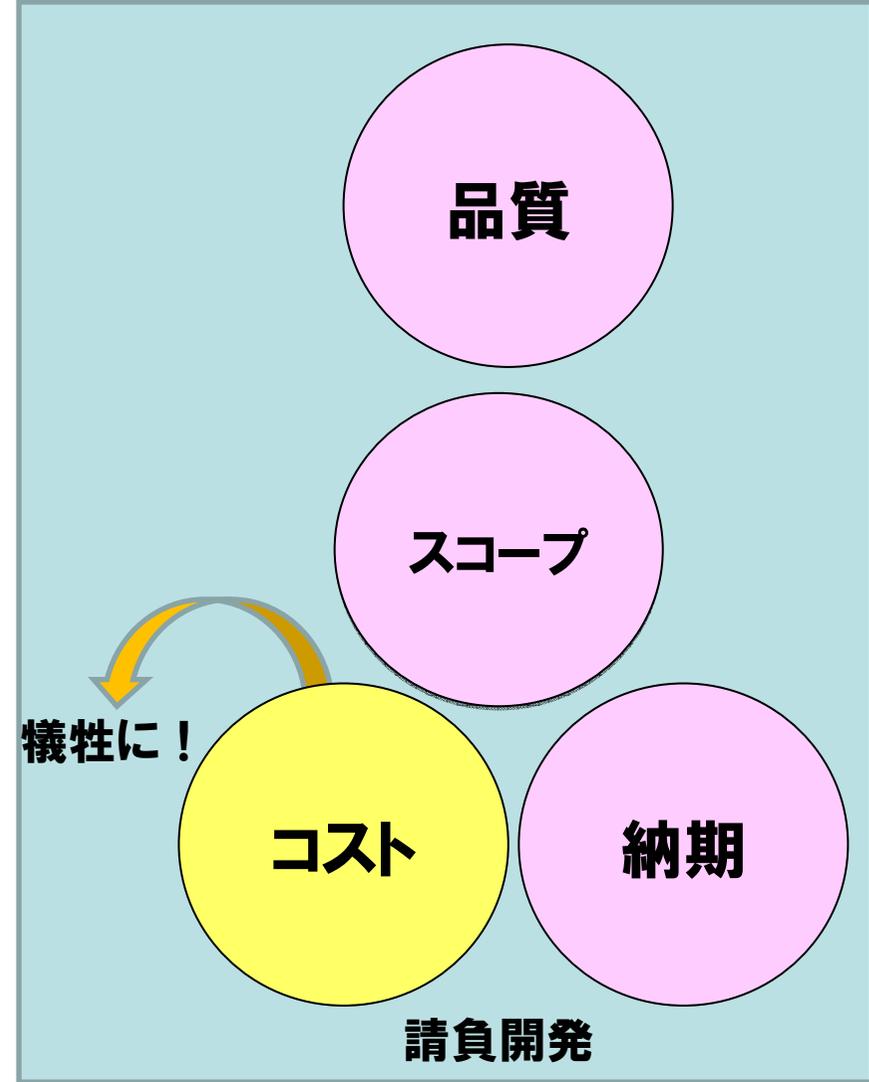
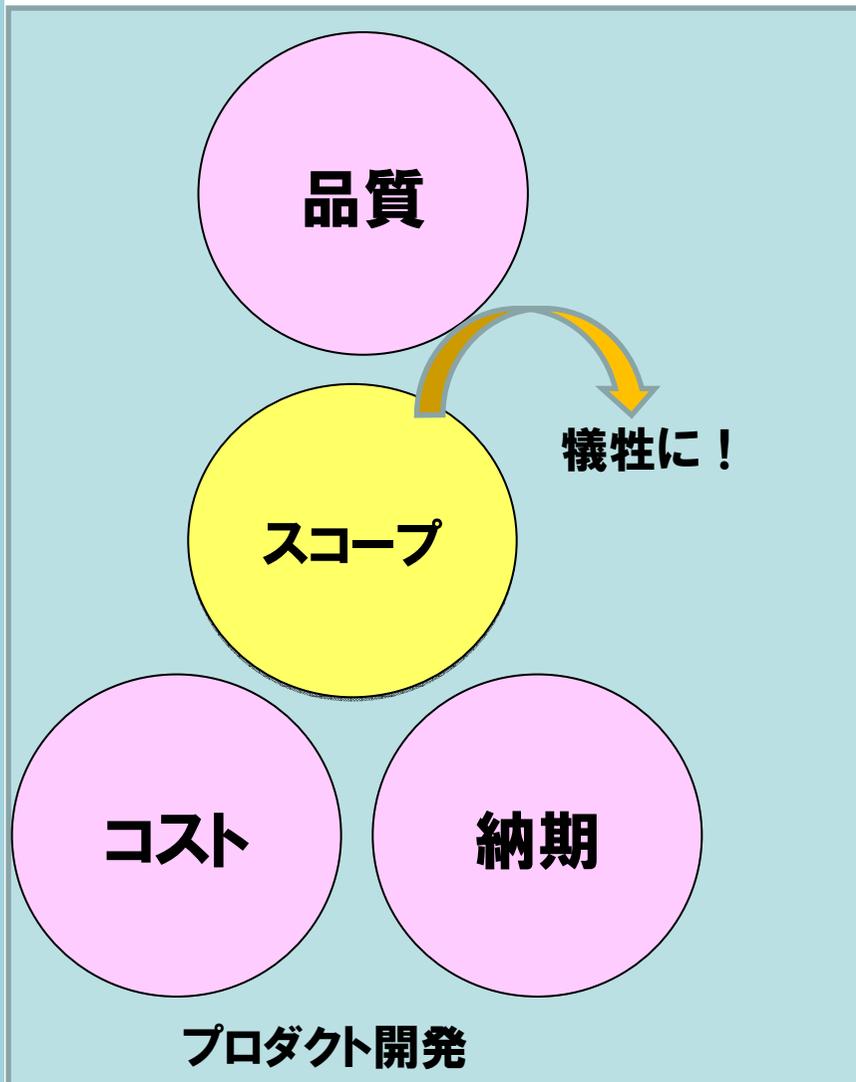
- ・場所が離れていることを補う方法・・・  
**Web-TV会議や進捗確認、バグトラッキングシステムなどのツールを積極活用**



# プロジェクト管理の3要素とスコープの4項目 System Integrator

● 計画通りに行かないとき、何を犠牲にするか。

アジャイル開発でも、請負契約ならコストを犠牲に





# プロジェクト管理の基礎を知っておく

PMBOK (Project Management Body of Knowledge)  
&  
CMMI (Capabiliti Maturity Model Integration)

## アジャイル開発においてもプロジェクト管理の基本は同じ

### •PMBOKの功績

1. プロジェクト管理というもやもやしたものを整理してくれた  
⇒9つの知識エリアと5つのプロセス
2. プロセスを管理する重要さを提案してくれた  
⇒QCD実現のために何を管理すべきか

### •PMBOKの課題

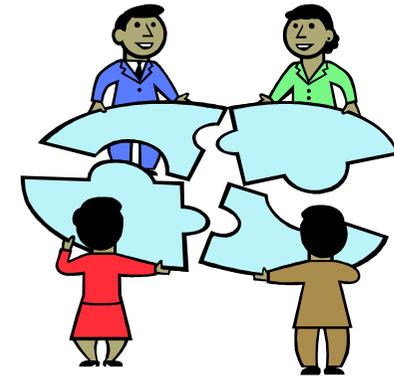
1. あくまでも知識(Knowledge)なので、そのまま現場では使えない  
⇒実現するための手段(ツール)が必要
2. 個別のプロジェクトの管理だけで、組織のプロジェクト管理力向上まではカバーしていない  
⇒複数のプロジェクトを横串で見たり、束ねて見たり、標準化など
3. 9つの知識エリアをバラバラに解説しているだけ  
⇒相互の関連性があるので、そこを抑えないと合理化できない

## ・ソフトウェア開発を行う組織のプロセス成熟度

レベル1	初期状態	出発点レベル(混沌とした、行き当たりばったり)
レベル2	管理された状態	反復してプロセスを実行できる
レベル3	定義された状態	プロセスが標準プロセスとして明示的に定義されている
レベル4	定量的に管理された状態	プロセス管理が実施され、さまざまなタスクを最適化して改善している
レベル5	最適化している状態	継続的に自らのプロセスを最適化し改善している

## ・CMMIのポイント

1. CMMIとCMMがある
2. レベル3が平均というわけではない
3. 会社単位でなく組織単位で取れる
4. 1回取って終わりというものではない
5. 日本はあまり積極的ではないが、米国や中国、インドでは認識が高い



# 統合管理 Integration Management

## ・プロジェクトの登録

- ・登録タイミング
- ・承認行為
- ・登録の単位
- ・予定と実績
- ・プロジェクト計画書

## ・プロジェクトの変更

- ・再承認プロセス

## ・プロジェクトの終結

- ・終結タイミング
- ・保証期間と保守
- ・レビューの実施と結果登録
- ・失敗原因の記録と改善活動
- ・プロジェクト完了報告書

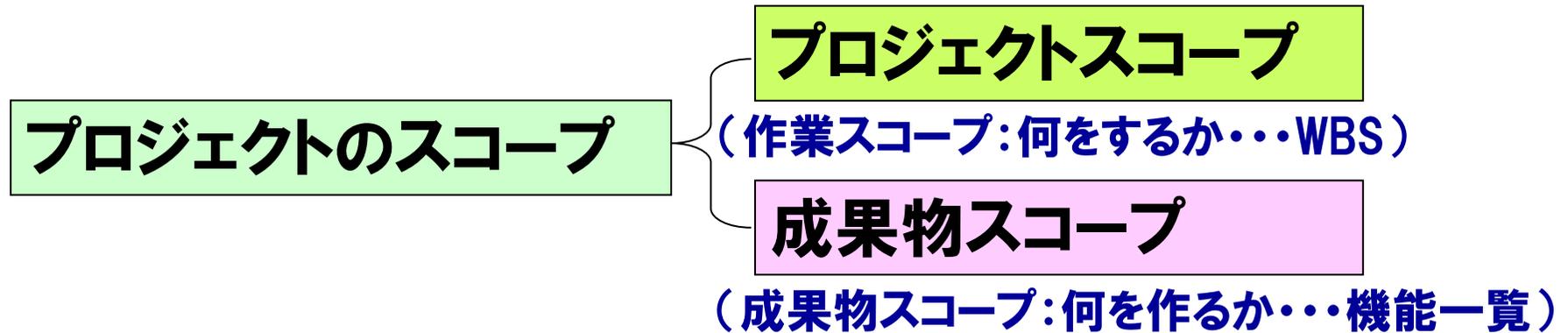
### 【大切なこと】

- ① 組織で、ルールを定める
- ② ルールを浸透、徹底する
- ③ ルール違反を取り締まる



# スコープ管理 Scope Management

WBS: Work Breakdown Structure (作業分解図)



- ウォーターフォール開発におけるスコープ計画  
要件定義工程でWBSと機能一覧を定義し、基本設計工程で確定



- アジャイル開発におけるスコープ計画

要件定義工程でWBSと機能一覧を定義し、以降順次スコープを決定してゆく。

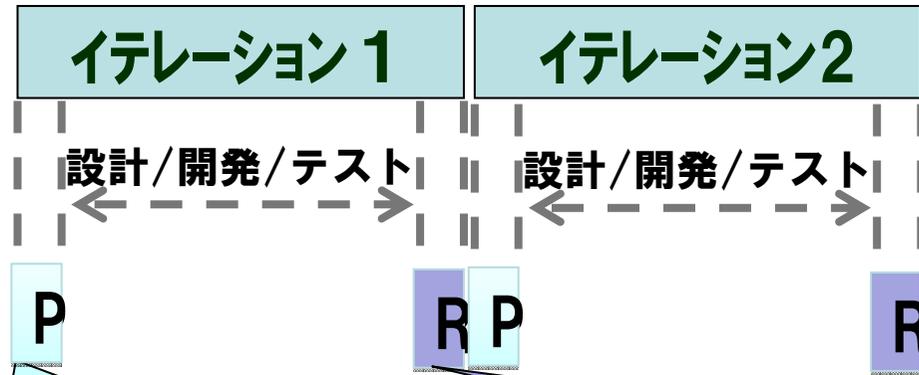




# スケジュール管理 Time Management

# スケジュールの計画(アジャイル開発の場合)

- アジャイルにおける計画…「リリース計画」と「イテレーション計画」
- 計画－実行－反省（調整）のフラクタル（相似形）な構造



変更分と新規分で優先順位を再設定し、タスク再構成

**P**  
**プランニングミーティング**  
 (ストーリーのタスク化と見積)

**R P** **R**  
**レトロスペクション**  
 (振り返りと調整)

デイリースクラム	毎朝15分程度。メンバーが前日やったこと、今日やることを報告。長くなりそうな場合は別ミーティングに切出す
スプリント (イテレーション)	2週間程度。初日にプランニング(計画)ミーティングを行いストーリーのタスク化と見積、最終日にレトロスペクション(振り返り)を実施。
マイルストーン	内部リリース。この程度の機能が実装されてなければならないという開発者側で設定する大きな目標(中だるみ防止)
リリース	外部へのリリース。リリースに必要なタスクやイベントもきちんと洗い出して計画する。

## ■ プランニングミーティング…初日に実施、2～3時間程度

1. タスクの決定（ミーティング前に）
2. 前イテレーションのタスク（チケット）完了の確認
3. メンバーの予定作業時間の確認…有給休暇や他の作業なども考慮
4. イテレーションのゴールを共有…全体像と今どこにいるかを確認
5. タスクを説明し工数を見積もる…プランニングポーカーなど複数で見積
6. タスクを割り当てる（ミーティング後に）

## ■ デイリースクラム

1. 昨日何をしたか？
2. 今日何をするか？
3. 進捗を妨げている問題は何か？

## ■ レトロスペクションミーティング（ふりかえり）…イテレーション最終日

1. 何をしたか
2. 何ができていないか。どんな課題が残っているか
3. ここまでは順調か。どうすればタスクを完了できそうか

## ●初期のアジャイルブーム

「ウォーターフォールはダサイ」→すべてアジャイルでやるべき

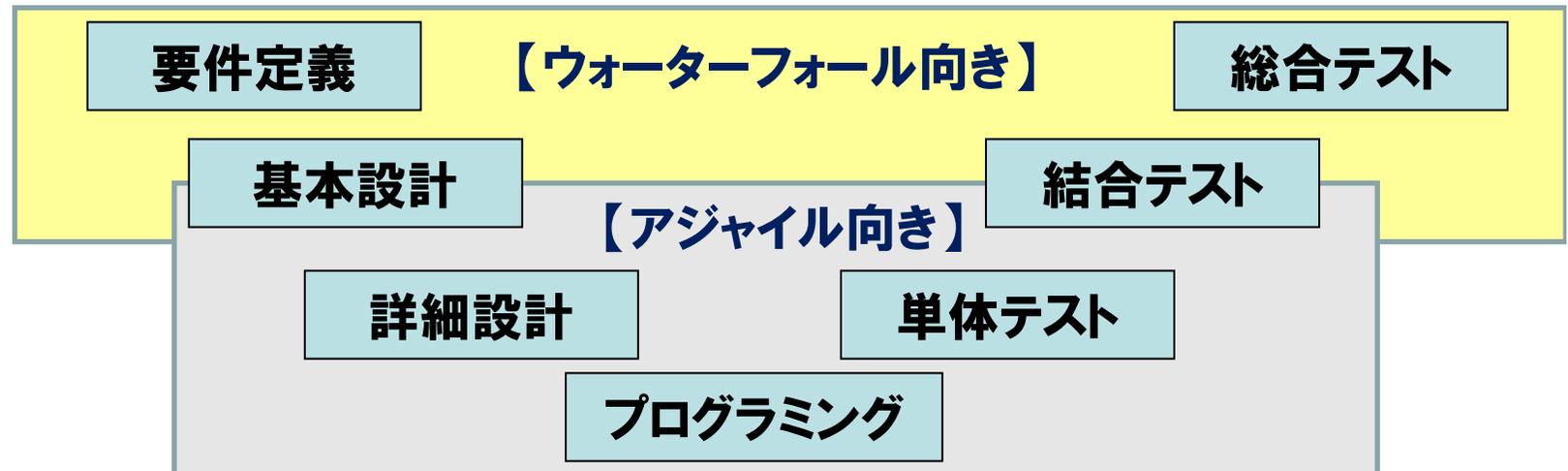
しかし、規模の比較的大きな開発では

最初(要件定義や基本設計)と最後(結合テストや総合テスト)はやりにくい。

## ●最近のトレンドは“ハイブリッド”

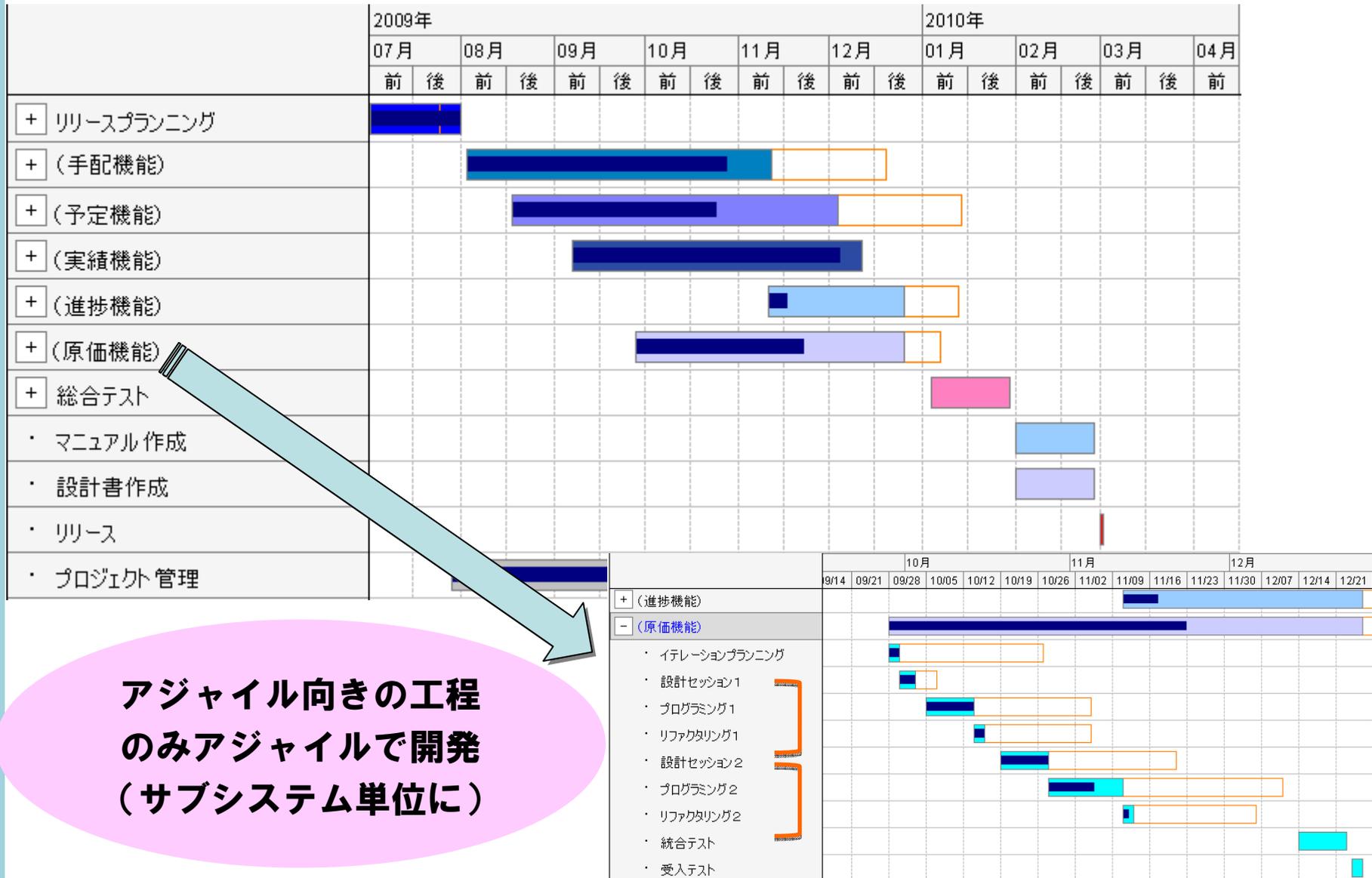
「blending agile and non-agile techniques and practices to create a hybrid methodology that fits larger organizations」

【フォレスター・リサーチ調査レポートのエグゼクティブサマリ】



# ハイブリッドなスケジュール計画

## WFとAGを組み合わせた例・・・サブシステムの中でn回のイテレーション

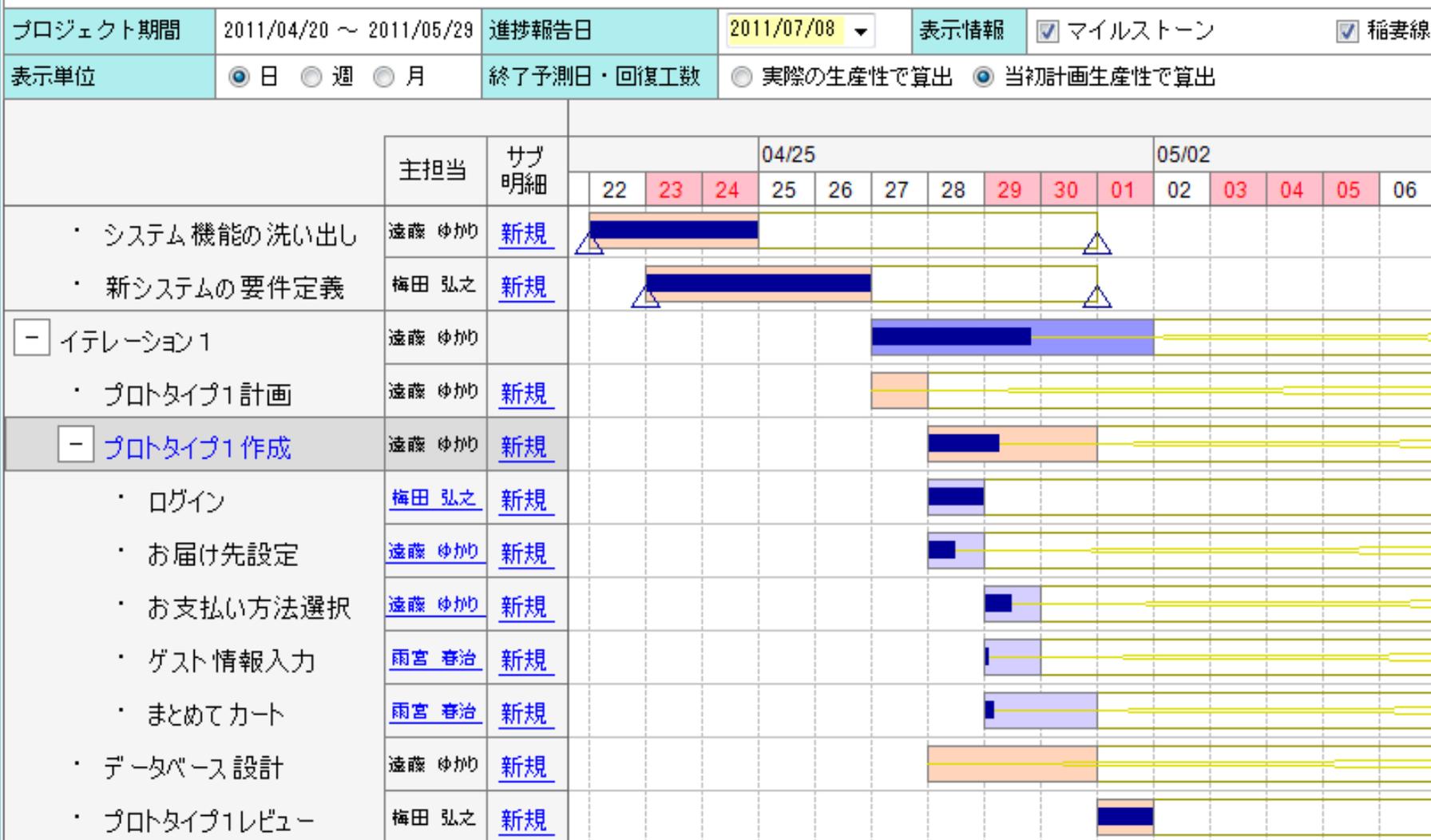


アジャイル向きの工程のみアジャイルで開発  
(サブシステム単位に)

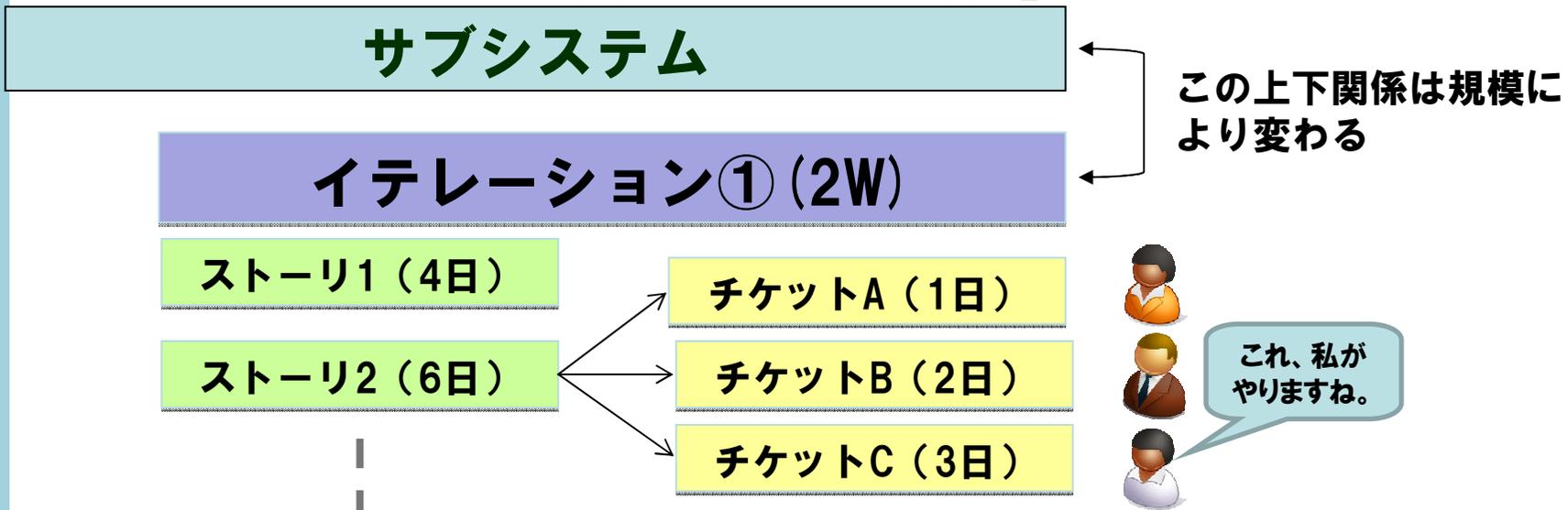
# 作業スコープと成果物スコープのWBS管理

## ■ガントチャートのWBS展開の原則

成果物のあるものは成果物スコープ、ないものは作業スコープ を管理

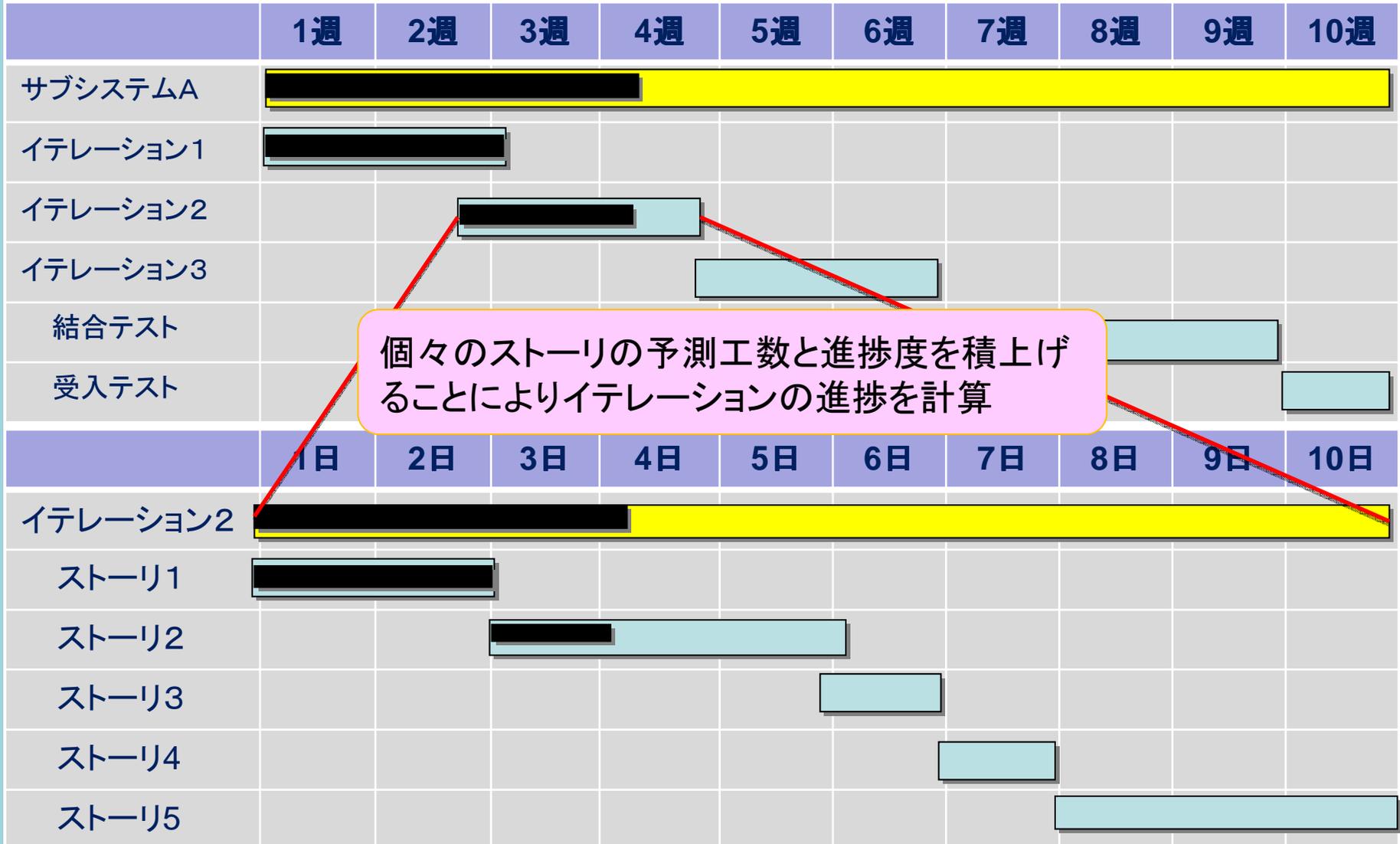


## ■ アジャイルにおける計画…「リリース計画」と「イテレーション計画」



# アジャイル開発における進捗管理

チケット>ストーリー>イテレーション>サブシステム>プロジェクト全体と進捗積上

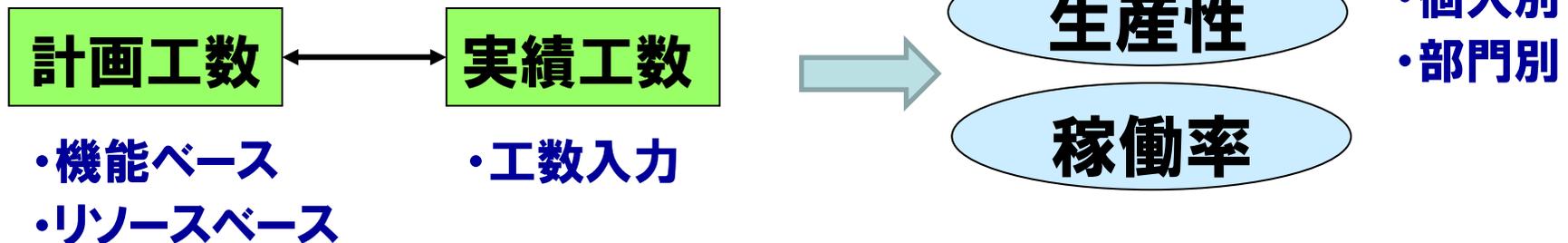


## ■階層積上計算

ストーリー>イテレーション>サブシステム>プロジェクト全体へと進捗を積上

上位タスクの進捗率 =  $\Sigma$  計画工数  $\times$  進捗率 (%)

## ■計画と実績の対比

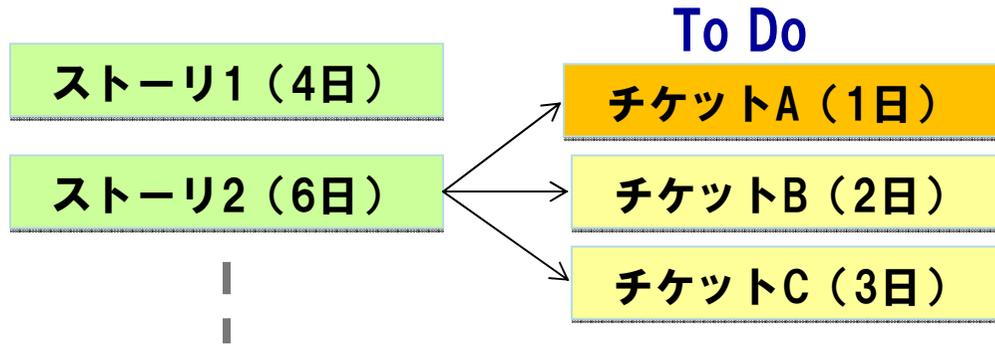


## ■進捗報告の3つのやり方



## プロジェクト管理と現場の作業管理

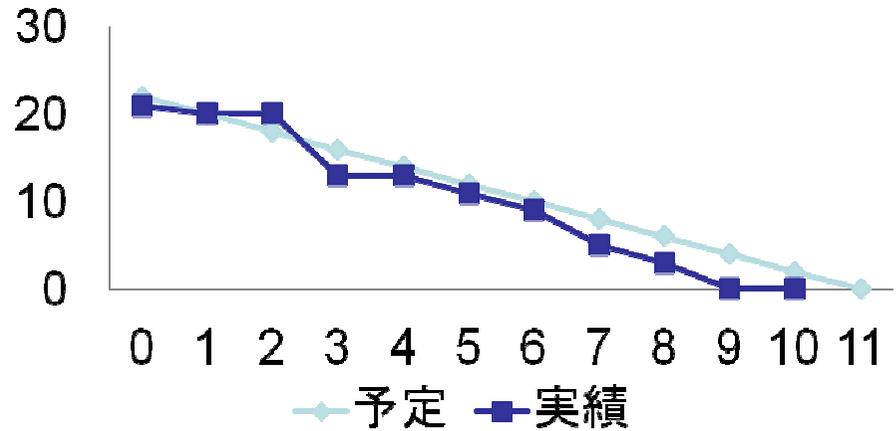
### イテレーション① (2W)



**【大切なこと】**  
 プロジェクト管理と  
 現場の作業管理を分ける



ストーリー	ToDo	仕掛中	要変更	完成
a.	a2. a3.	a.4	a1.	a5. a6.
b.	b1	b2. b3.	b4.	b5.



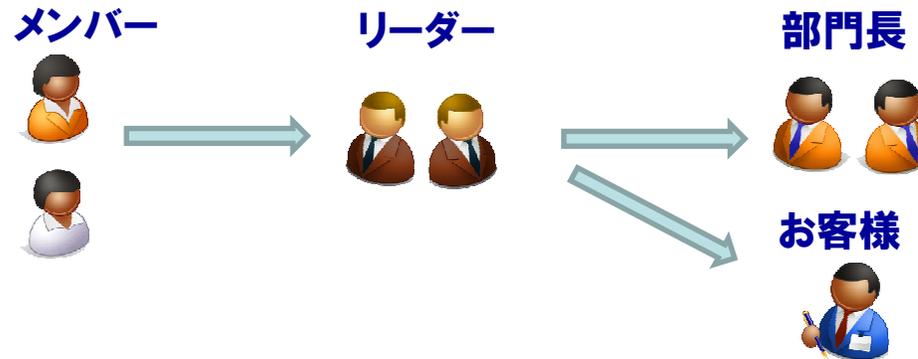
スクラムボード



バーンダウンチャート

# 進捗報告は2つのレイヤーで

■メンバー⇒プロジェクトリーダー⇒部門長/お客様・・・逆ウォーターフォール



## ■進捗報告の問題点と解消手段

進捗報告漏れ・・・アラートが上がる仕組み

進捗がいい加減・・・細かなタスクの進捗を積み上げる仕組み

あいまいな報告・・・状況を数値化して見せる仕組み

報告が進捗だけ・・・QCDの3要素を報告できる仕組み

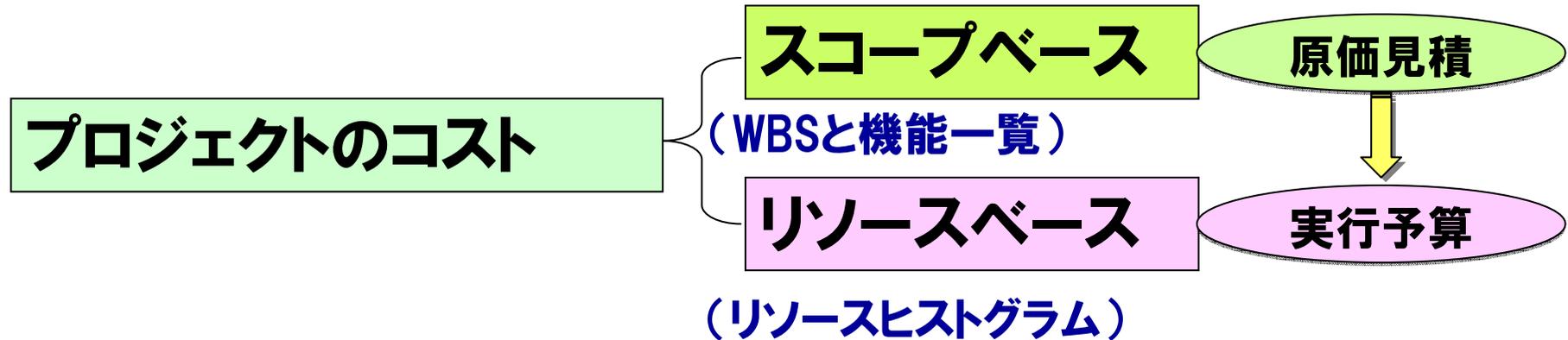
報告内容が急変・・・進捗報告を履歴管理する仕組み

報告を見てない・・・上司が報告を確認し、コメントを残せる仕組み



# コスト管理 Cost Management

## ■ スコープベースとリソースベースの関係性を理解



スコープベースの工数見積 ≒ リソースベースの工数見積  
 実行予算(正確な原価) => リソースベースの原価見積

## ■ 工数見積 2つのタイミング

要件定義終了時

ファンクションポイント法など

ユーザファンクションタイプ	個数	重み付け係数
外部入力	1	4
外部出力	2	5
内部論理ファイル	1	10
外部インタフェースファイル	0	7
外部照会	0	4

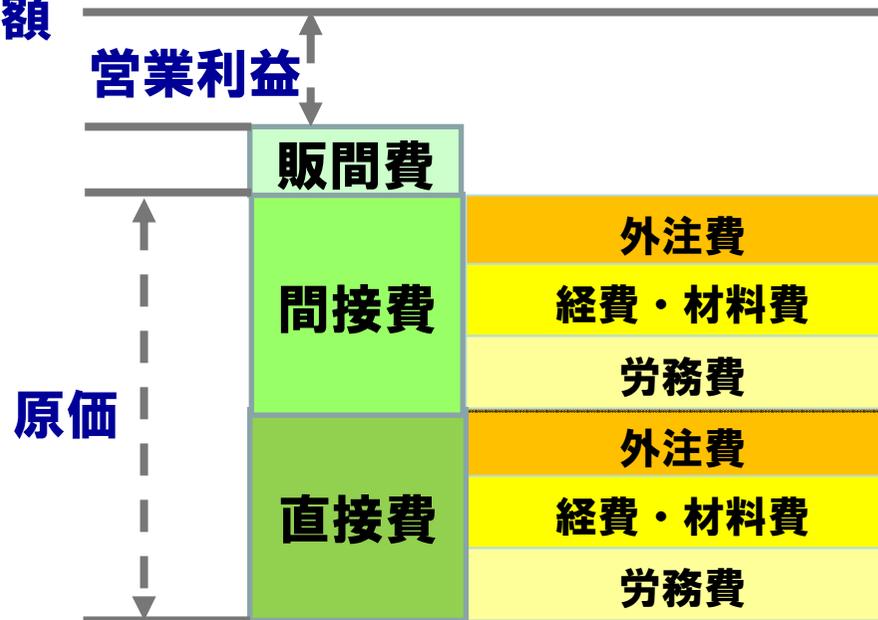
イテレーション開始時

プランニングポーカーなど

1	2	3	5
8	13	20	?

## ■プロジェクトの原価構造

契約金額



## 原価管理のポイント

- 原価実績計算の仕組み
- 原価見積で間接費も含める
- 実績だけでなくゴールを把握
- EVMグラフ
- 進行基準と完成基準
- 月中での原価発生額を把握

## ■原価管理と工数管理と勤怠管理

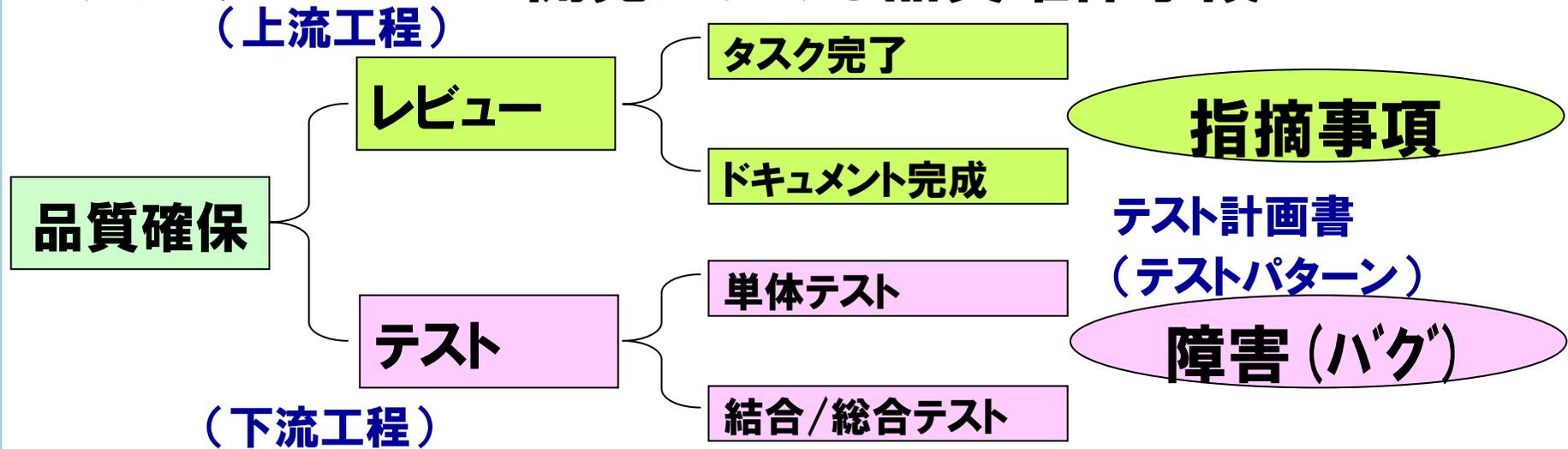
バラバラに管理するのは非合理的



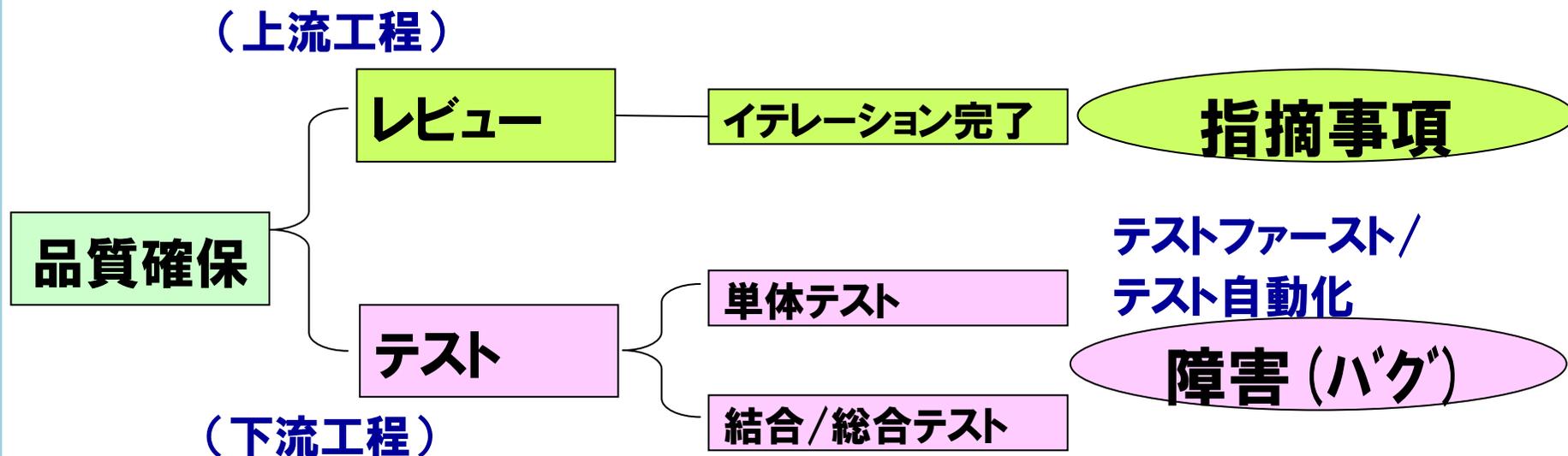
# 品質管理

## Quality Management

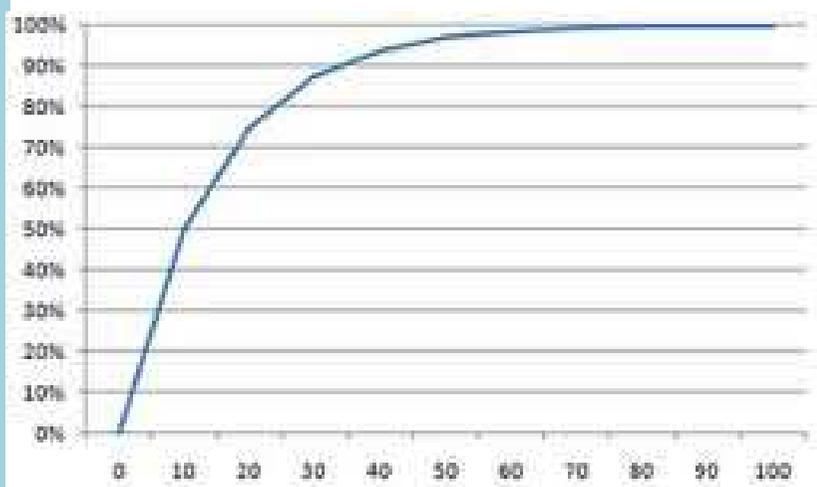
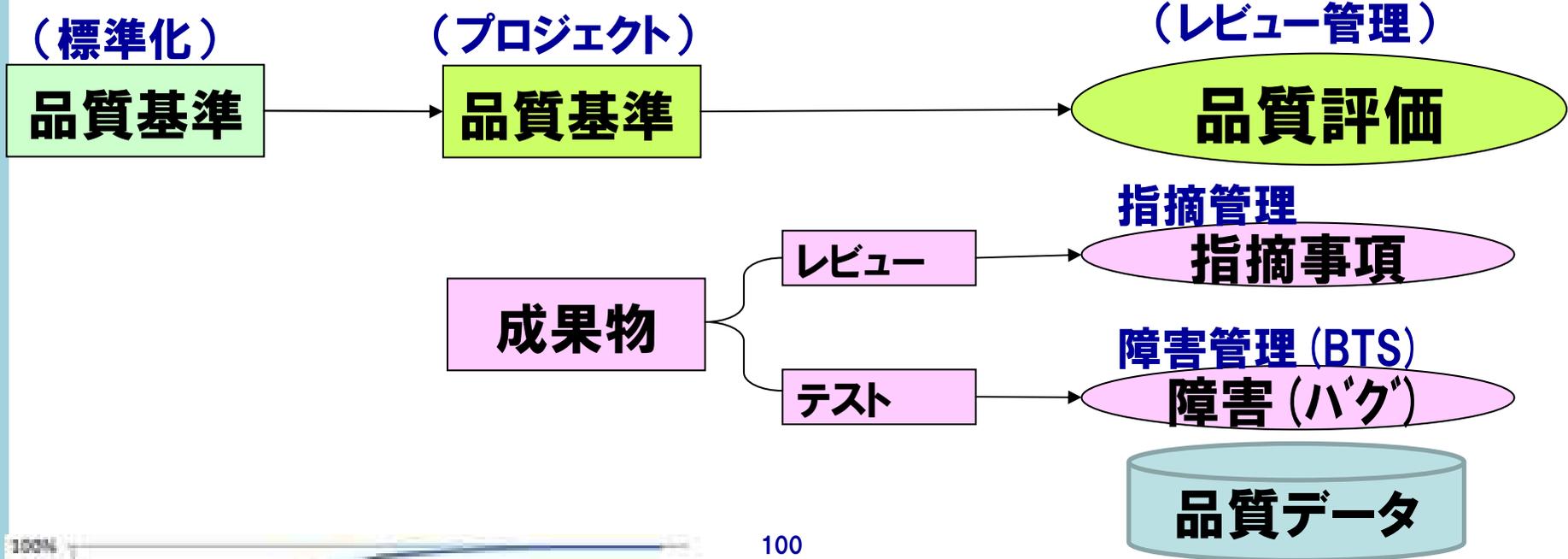
## ・ウォーターフォール開発における品質確保手段



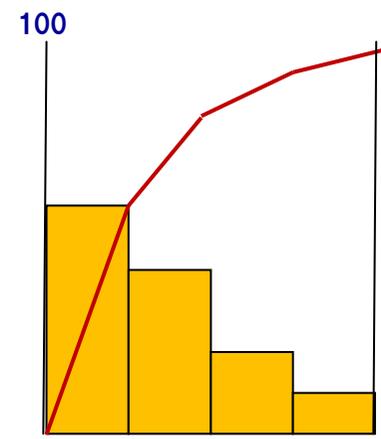
## ・アジャイル開発における品質確保手段



## 品質管理の体系化と結果の保存



バグ曲線



パレート図



# 要員管理

## Human Resource Management

## ■リソースヒストグラム

計画と実績

複数プロジェクトを担当している人の負荷管理

コストとの関連性

ガントチャート（計画工数）との関連性

勤怠実績との関連性

## ■スキル管理

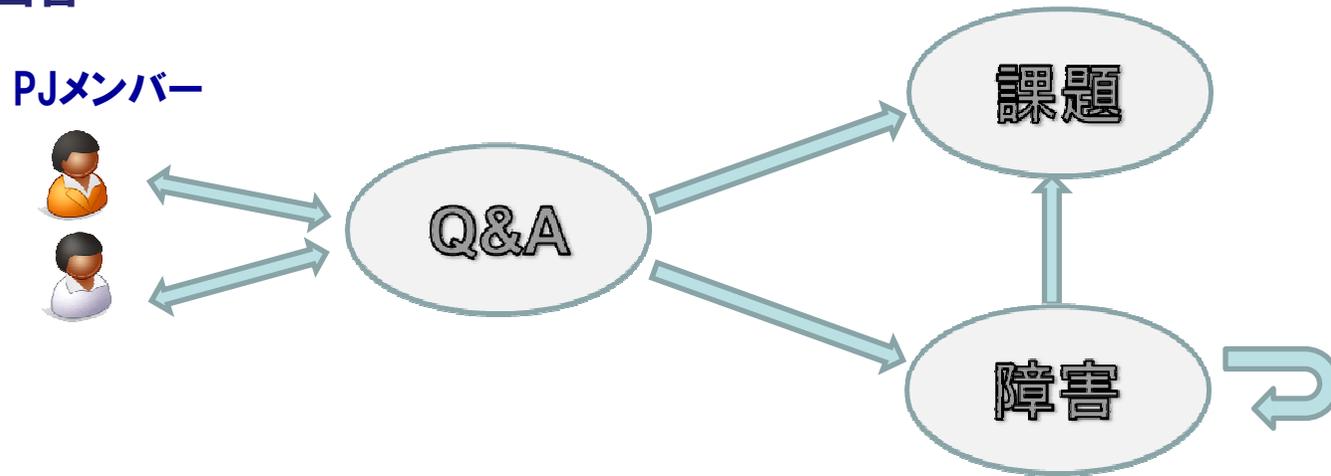
いつ、誰がスキルを評価するか

スキルとして捉える要素



# コミュニケーション管理 Communication Management

## ■質問&回答



## ■コミュニケーションの対象

メンバー間

メンバーとプロジェクトリーダー

プロジェクトリーダーと部門長

プロジェクトリーダーとユーザー



# リスク管理 Risk Management

## ■リスク管理（計画プロセス）

リスクの標準化

リスクの洗い出しと評価

リスクの共有と残管理

## ■課題管理（実行プロセス）

課題の登録と優先順位、担当者アサイン

課題の共有と残管理



# 調達管理 Procurement Management

## ■協力会社のスキル管理

会社と個人のスキルの共有

## ■要員（負荷）管理

工数実績と稼働率、生産性管理

## ■契約管理

契約管理

稼働実績と仕入計上

1. アジャイルは目的ではなく、勝つための手段
2. 実績のあるウォーターフォールを敵視せず、ハイブリッドで考えよう
3. 請負契約でもアジャイル開発ができるように工夫しよう
4. スコープにはプロジェクトスコープと成果物スコープがある
5. 成果物のあるタスクは、成果物で管理しよう
6. 繰返し型のアジャイル開発では最初（計画）と最後（振返り）が重要
7. リリース計画で全体を把握しながら、イテレーション計画で作業管理
8. スコープベースとリソースベースでコストを算出して比較しよう
9. 下流工程のテストとバグの関係は、上流工程ではレビューと指摘事項
10. テストファースト、テスト自動化など良いプラクティスは積極採用
11. プロジェクトの進捗は、詳細タスクの進捗を積上げ計算して求める
12. 進捗報告は2つのレイヤーで履歴型で内容を共有しよう
13. プロジェクトの原価構造を理解し、原価見積精度を上げよう
14. 品質基準や見積係数、WBSなどは標準化を図ろう
15. 品質データは蓄積してグラフなどで分析しよう

**「アジャイルにしたらプロジェクト管理が手を抜ける」 → ということはないなあ…**



# System Integrator

株式会社システムインテグレータ

TEL : 048-707-7061 (東京)

06-4706-5471 (大阪)

E-Mail: OBPM... [obpm@sint.co.jp](mailto:obpm@sint.co.jp)

URL: <http://www.sint.co.jp>