プロジェクトマネジメント学会 中国支部
2019年5月　セミナー

HITACHI
Inspire the Next

アジャイル概説と適用のポイント

～日本の特徴に合わせたアジャイル適用～

2019年5月24日

株式会社　日立ソリューションズ
ハナブサ　シゲオ
英　繁雄

HITACHI
Inspire the Next

Contents

HITACHI
Inspire the Next

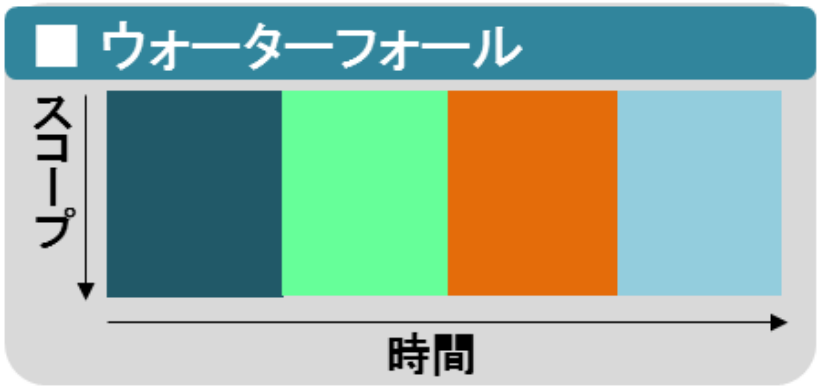
1. アジャイル開発概説

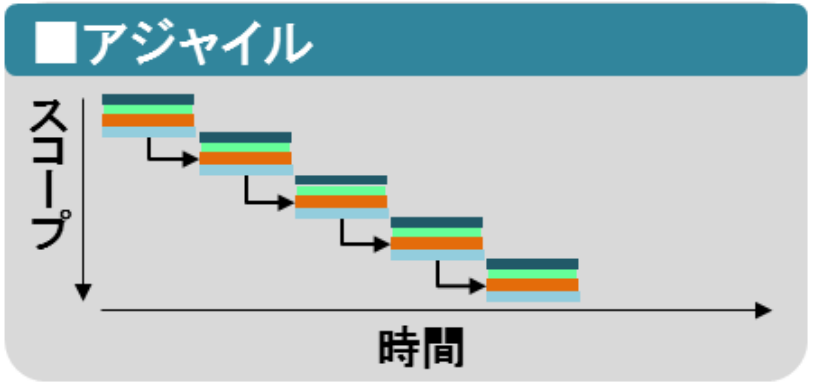
1-1　アジャイル開発とは？

HITACHI
Inspire the Next

顧客・市場の要求にしたがって，優先度の高い機能から順に、
要求・開発・テスト(・リリース)を短い期間で繰り返しながら、
システムの構築・改善を行う手法

原則として、事前に開発の詳細な計画は作らず、
1～4週間という一定の短い周期で要求・開発・テストを繰り返しながら動作可能なソフトを作り上げる。





1-2 アジャイルソフトウェア開発宣言

HITACHI
Inspire the Next

私たちは、ソフトウェア開発の実践あるいは実践を手助けする活動を通じて、よりよい開発方法を見つけだそうとしている。
この活動を通して、私たちは以下の価値に至った。

・プロセスやツール	よりも	**個人と対話**を
・包括的なドキュメント	よりも	**動くソフトウェア**を
・契約交渉	よりも	**顧客との協調**を
・計画に従うこと	よりも	**変化への対応**を

価値とする。すなわち、左記のことがらに価値があることを認めながらも、私たちは右記のことがらにより価値をおく。

Kent Beck
Martin Fowler
Jon Kern
Jeff Sutherland
Mike Beedle
James Grenning
Brian Marick
Dave Thomas
Arie van Bennekum
Jim Highsmith
Robert C. Martin
Alistair Cockburn
Andrew Hunt
Steve Mellor
Ward Cunningham
Ron Jeffries
Ken Schwaber

HITACHI
Inspire the Next

アジャイル宣言の背後にある原則

http://agilemanifesto.org/iso/ja/principles.html

私たちは以下の原則に従う:

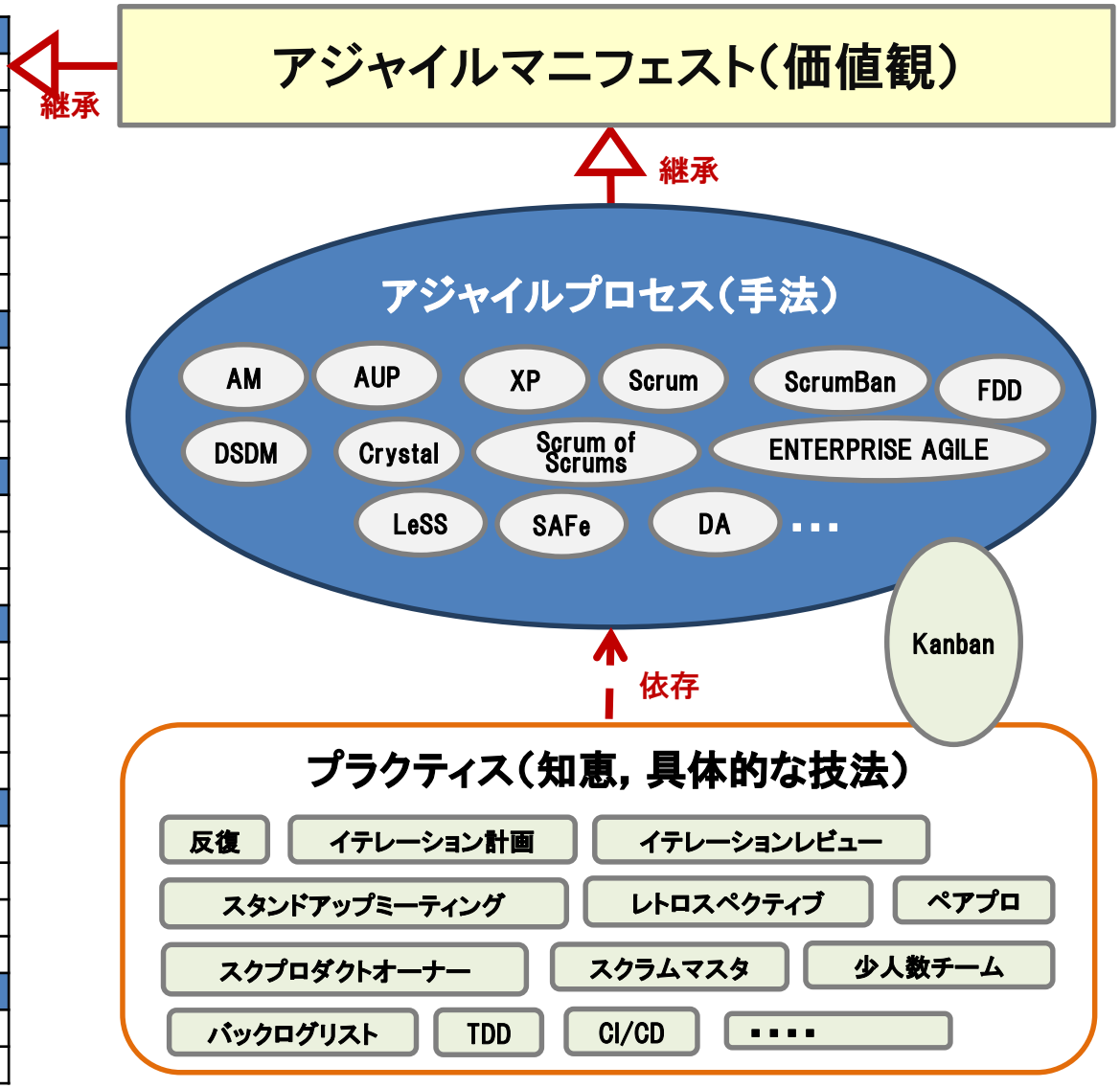
① 顧客満足を最優先し **価値のあるソフトウェアを早く継続的に提供** します。
② **要求の変更はたとえ開発の後期であっても歓迎** します。
③ 動くソフトウェアを、2-3週間から2-3ヶ月という **できるだけ短い時間間隔でリリース** します。
④ ビジネス側の人と開発者は、プロジェクトを通して **日々一緒に** 働かなければなりません。
⑤ **意欲に満ちた人々** を集めてプロジェクトを構成します。
環境と支援を与え仕事が無事終わるまで彼らを信頼します。
⑥ 情報を伝えるもっとも効率的で効果的な方法は **フェイス・トゥ・フェイス** で話をすることです。
⑦ **動くソフトウェアこそが進捗の最も重要な尺度** です。
⑧ アジャイル・プロセスは **持続可能な開発** を促進します。
一定のペースを継続的に維持できるようにしなければなりません。
⑨ 技術的卓越性と優れた設計に対する不断の注意が **機敏さ** を高めます。
⑩ **シンプルさ**（ムダなく作れる量を最大限にすること）が本質です。
⑪ 最良のアーキテクチャ・要求・設計は、**自己組織的なチーム** から生み出されます。
⑫ チームがもっと効率を高めることができるかを **定期的に振り返り**、
それに基づいて自分たちのやり方を最適に調整します。

1-4 アジャイルの本質はムダ取り

HITACHI
Inspire the Next

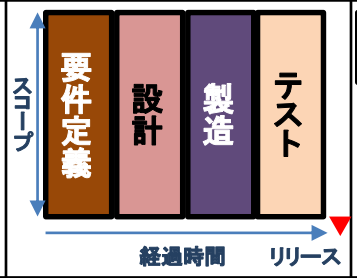
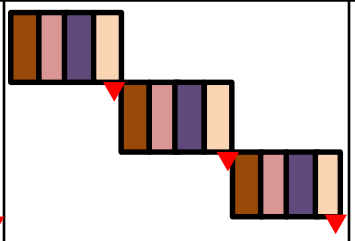
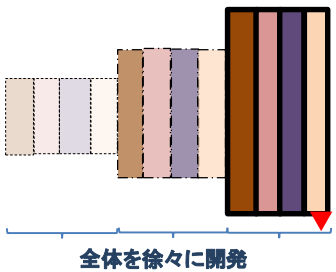
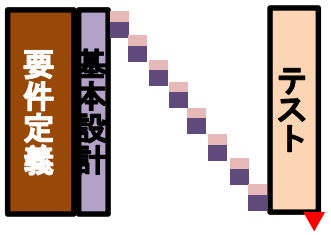
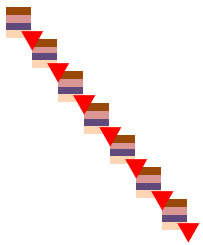
リーンの原則とツール(思考)

原則1:ムダをなくす
ツール1:ムダの認識
ツール2:バリューストリームマッピング
原則2:知識を作り出す(学習効果を高める)
ツール3:フィードバック
ツール4:イテレーション
ツール5:同期
ツール6:集合ベース開発
原則3:決定をできるだけ遅らせる
ツール7:オプション思考
ツール8:最終責任時点
ツール9:意思決定
原則4:できるだけ早く提供する
ツール10:プルシステム
ツール11:待ち行列理論
ツール12:遅れのコスト
原則5:人を尊重する(チームに権限を与える)
ツール13:自発的決定
ツール14:モチベーション
ツール15:リーダーシップ
ツール16:専門知識
原則6:品質(統一性)を作り込む
ツール17:認知統一性
ツール18:コンセプト統一性
ツール19:リファクタリング
ツール20:テスティング
原則7:全体を最適化する
ツール21:計測
ツール22:契約



1-5 開発モデルの比較

HITACHI
Inspire the Next

	ウォーターフォール	インクリメンタル	イテレーティブ（スパイラル含む）	ハイブリッドアジャイル	アジャイル(Scrum)
					
反復	1回のみ	反復（数カ月程度）	反復（数カ月程度），完成するまで繰り返す	アジャイル期間は短い反復（2～4週間の一定の間隔）	短い反復（1～4週間の一定の間隔）
目的	各工程を確実に完成させる（確実さを重視）	早期リリース（できたところから早めにリリース）	未知なものを作りながら完成させていく	確実に完成させる，変更受入れ前提，ムダを除外する	ムダを除外する（効率重視）
成果物	要件定義書，設計書，ソースコード，テスト報告書など	要件定義書，設計書，ソースコード，テスト報告書など	要件定義書，設計書，ソースコード，テスト報告書など	要件定義書，基本設計書，ソースコード，テスト報告書など	ソースコード,必要最低限のドキュメント
体制	プロジェクトマネージャー，設計者,プログラマ,品質保証部員の役割分担が一般的	ウォーターフォールと同様	設計者とプログラマは同一人物が理想	・アジャイル牽引者必要 ・ユーザーの定期確認 ・アジャイル期間は、メンバー固定が理想	・アジャイル牽引者必要 ・ユーザーの常時確認 ・メンバー固定が理想
要件定義	事前確定	反復ごとに事前確定	変更を前提	事前確定，ただし変更許容	反復の中で確定していく
リリース	最後に一度	反復ごとにリリース（完成の意味）	最後に一度	最後に一度	反復ごとにリリース（完成の意味）
相性	委託開発，大規模開発	大規模開発の分割リリースやサービスの早期提供	仕様に未知な部分がある新製品開発	委託開発（要件の変更に対応が必要な場合）	小規模開発、内製開発
プラクティス	―	―	―	TDD*1,CI/CD*2,ペアプログラミング,バックログリスト,カンバンなどプラクティスを選択	TDD*1,CI/CD*2,ペアプログラミング,バックログリスト,カンバンなどプラクティスを選択

*1 TDD:Test Driven Development（テスト駆動開発）
*2 CI/CD:Continuous Integration/Continuous Delivery（継続的インテグレーション/継続的デリバリー）

1-6 価値を強く意識する

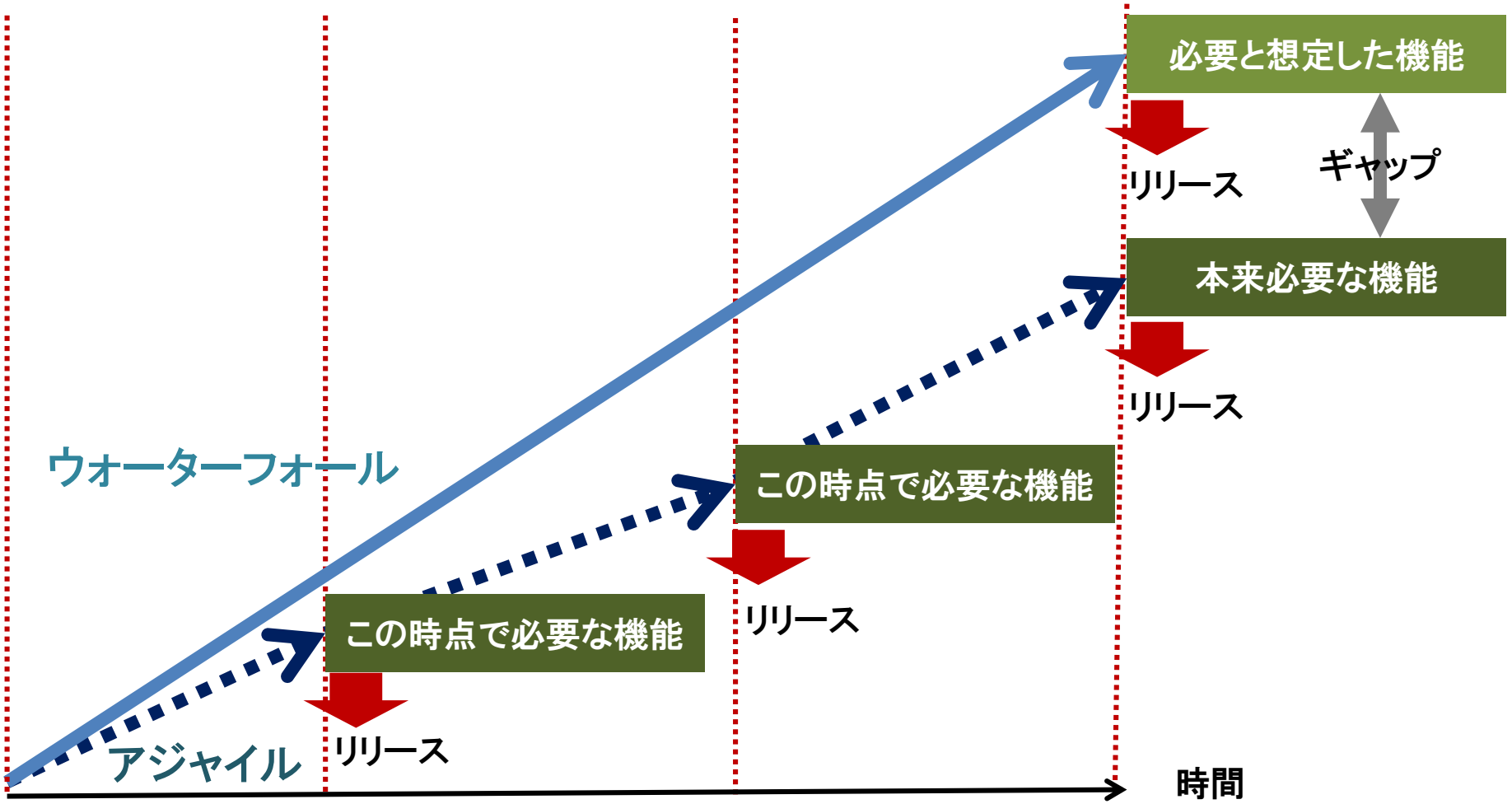
HITACHI
Inspire the Next

Q(Quality)
品質
＋
C(Cost)
コスト
＋
D(Delivery)
納期
＋
V(Value)
価値
S(Satisfaction)
満足度

- ビジネス価値で優先度を決める
- 部分的であっても早期にリリースし、ビジネス価値を早く得る
- コストとスケジュールを固定して、リリースする機能を調整する
- 本当に必要な機能は、半分もない
- 最初に、完璧な仕様はできない(必ず変更が入る)ものである
- ムダな作業は止める
- 確認は、動作するシステムで行う
- コミュニケーションは密にする(スタンドアップミーティング)

1-7　価値のある機能をタイミングよくリリースする

HITACHI
Inspire the Next

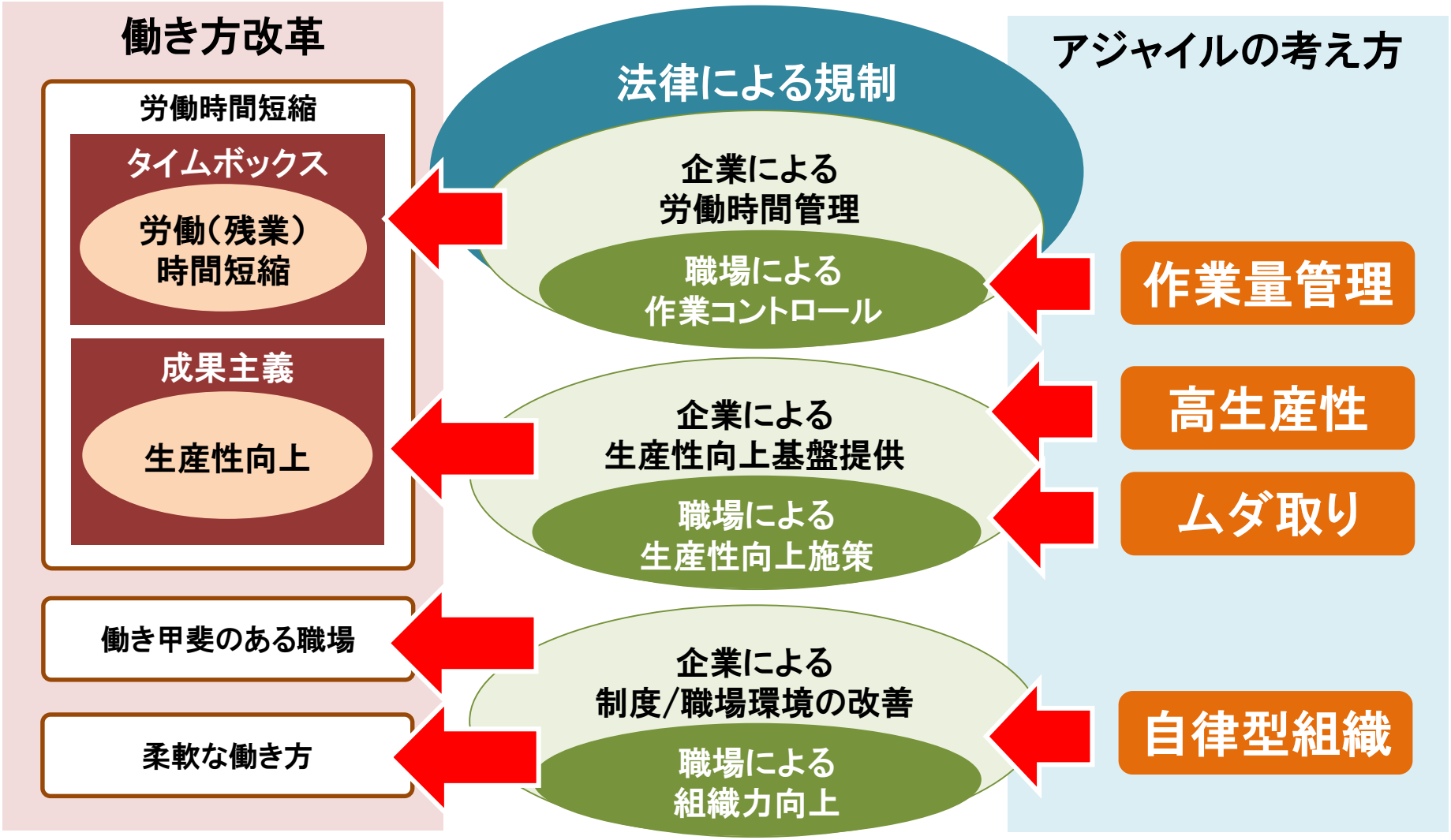


アジャイルは、再調整しながら必要な機能をタイミングよくリリースする。ビジネス利益を早期に得て、無駄を作らない。

1-8 働き方改革のためにも必要な考え方

HITACHI
Inspire the Next

アジャイルの考え方は働き方改革の手段でもある



1-9　第1章のまとめ

HITACHI
Inspire the Next

① リーン(ムダ取り)がアジャイルの本質である

② アジャイルは、他の反復開発とは異なる価値観がある

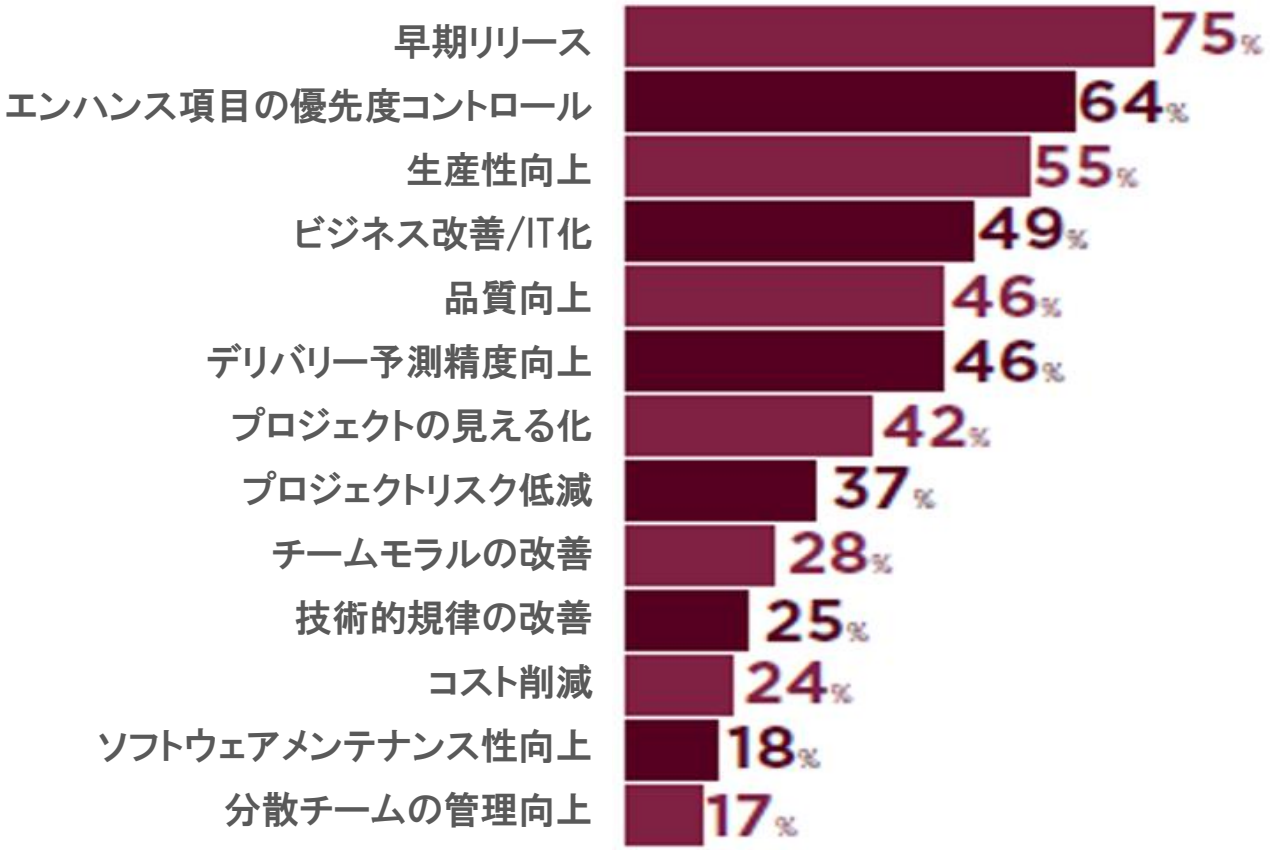
③ ビジネス価値を優先して早くタイミングよくリリースする

④ アジャイルは、働き方改革にも必要な考え方である

HITACHI
Inspire the Next

2. アジャイルの適用状況

2-1　アジャイル適用の目的

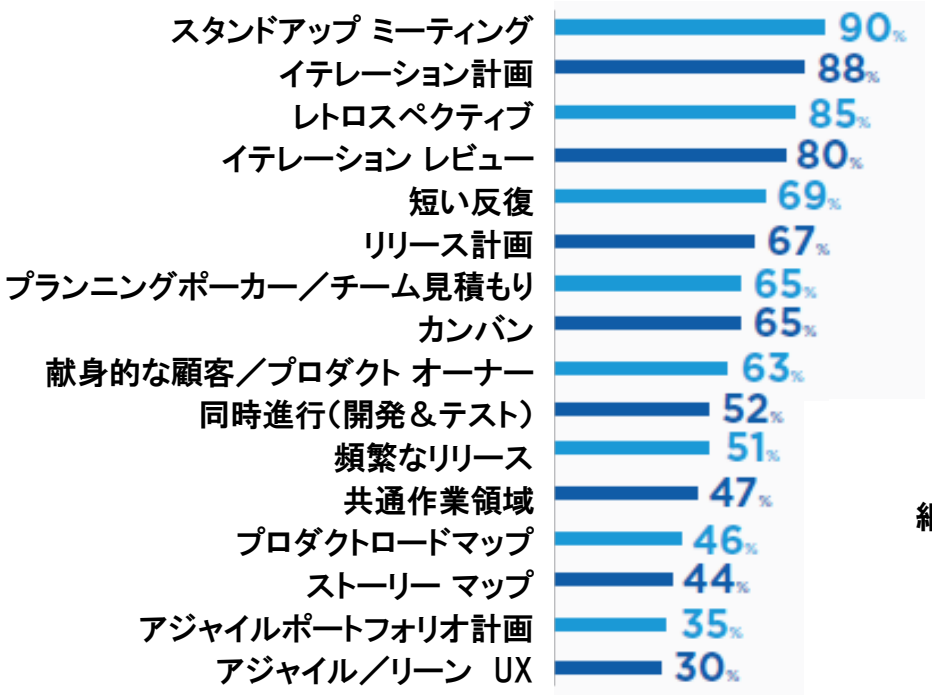


出典：One-12th-Annual-State-of-Agile-Report(COLLABNET　VERSIONONE社)から日本語に翻訳
https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report

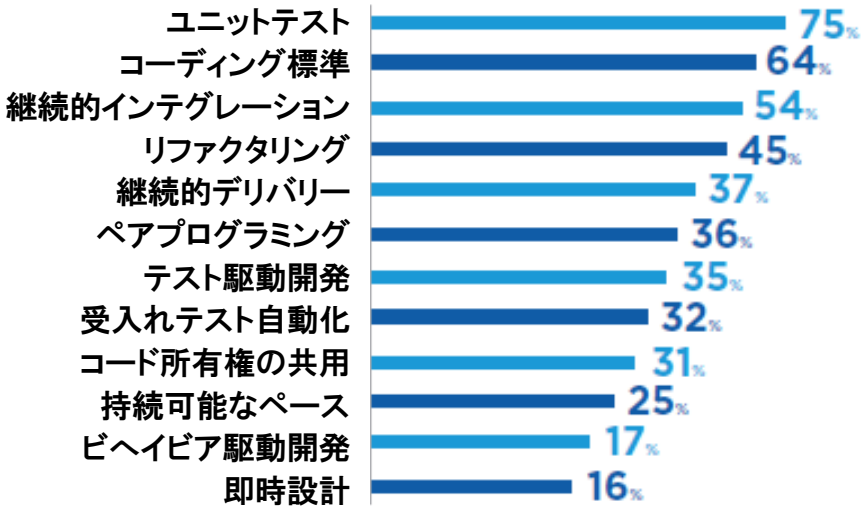
2-2 アジャイル開発のプラクティス適用状況

HITACHI
Inspire the Next

採用したアジャイルテクニック



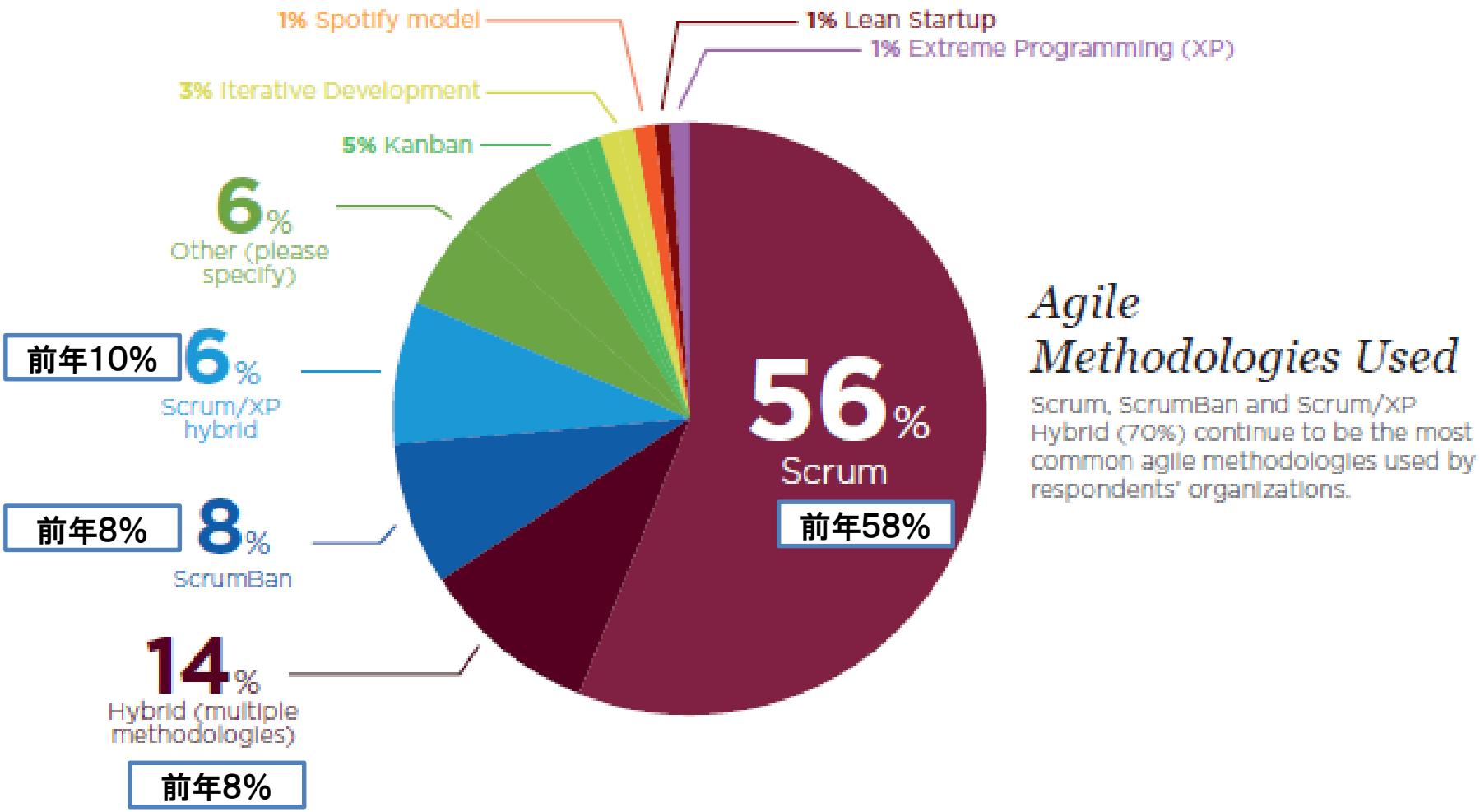
採用した技術プラクティス



出典：One-12th-Annual-State-of-Agile-Report（COLLABNET　VERSIONONE社）から日本語に翻訳
https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report

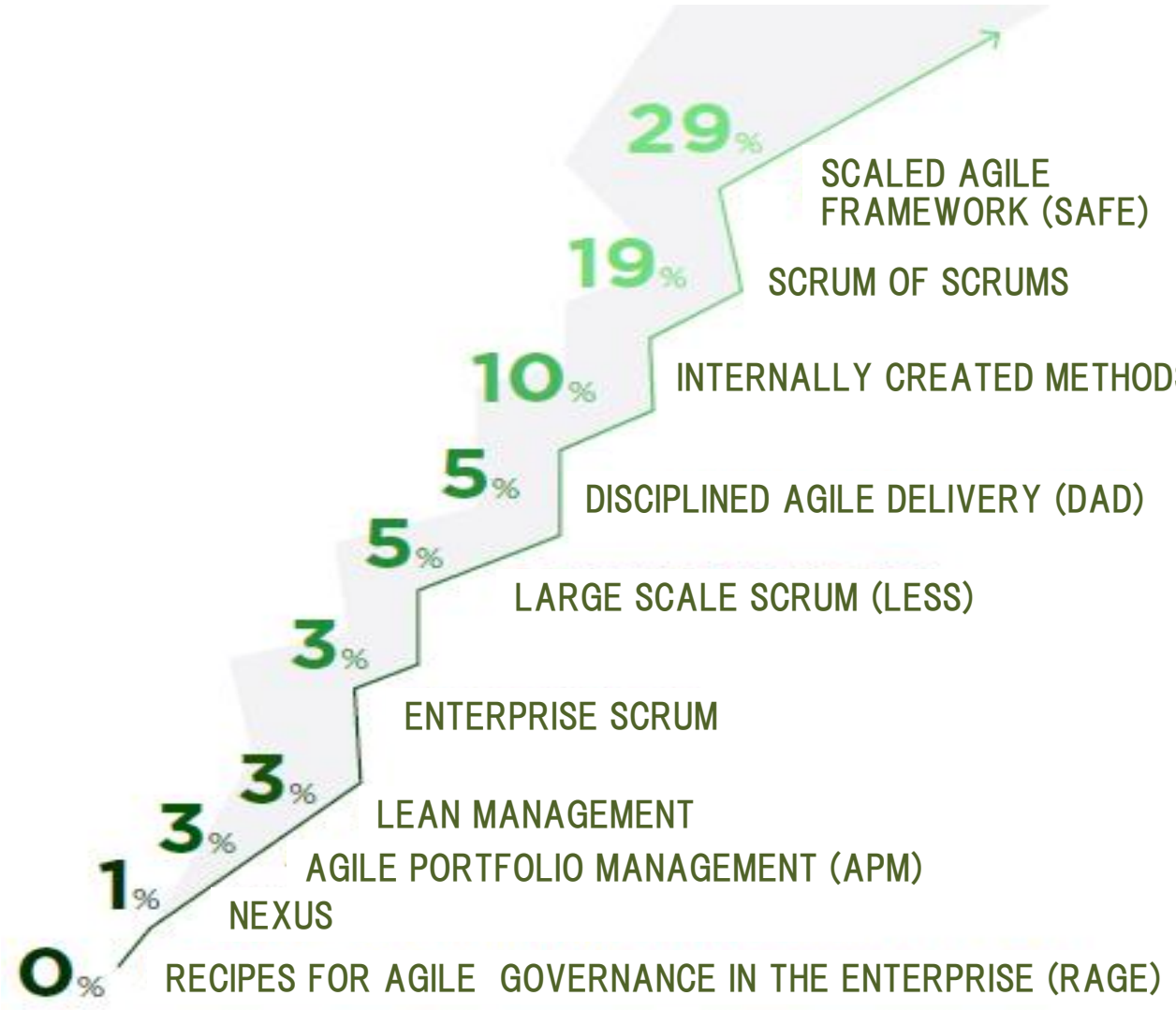
2-3 アジャイル開発手法の適用状況

HITACHI
Inspire the Next



出典：One-12th-Annual-State-of-Agile-Report(COLLABNET　VERSIONONE社)から日本語に翻訳
https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report

2-4 Scaling Agile（大規模開発アジャイル）



出典：One-12th-Annual-State-of-Agile-Report(COLLABNET VERSIONONE社)から日本語に翻訳
https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report

2-5　第2章のまとめ

HITACHI
Inspire the Next

① アジャイル適用の目的を明確にする

② 目的を達成するためのプラクティスを採用する

③ Scrumベースのアジャイル手法が多い

④ 大規模向けのアジャイル手法も提唱されてきた

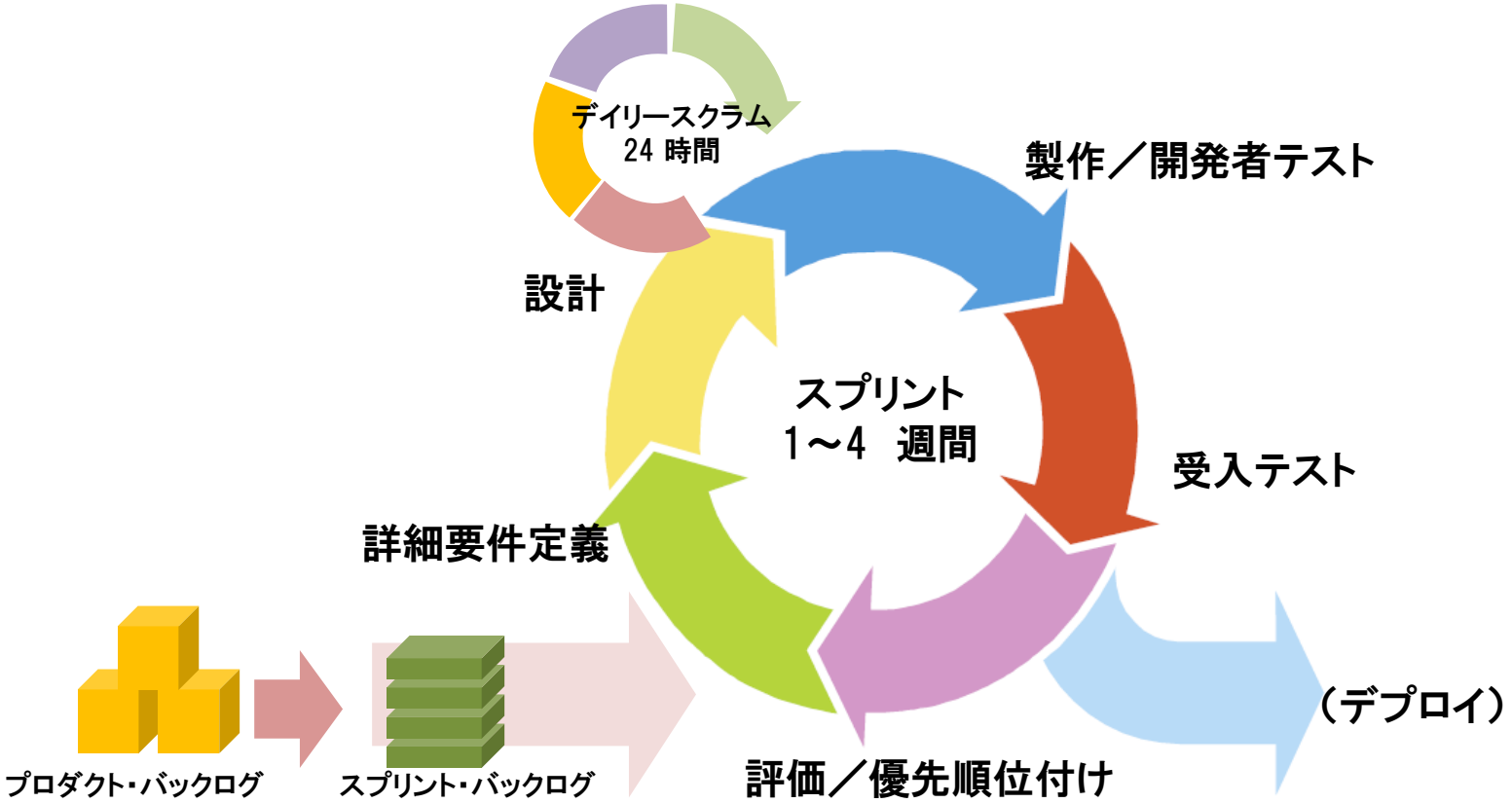
HITACHI
Inspire the Next

3. 主なアジャイル手法の概説

3-1　Scrumのサイクル

HITACHI
Inspire the Next

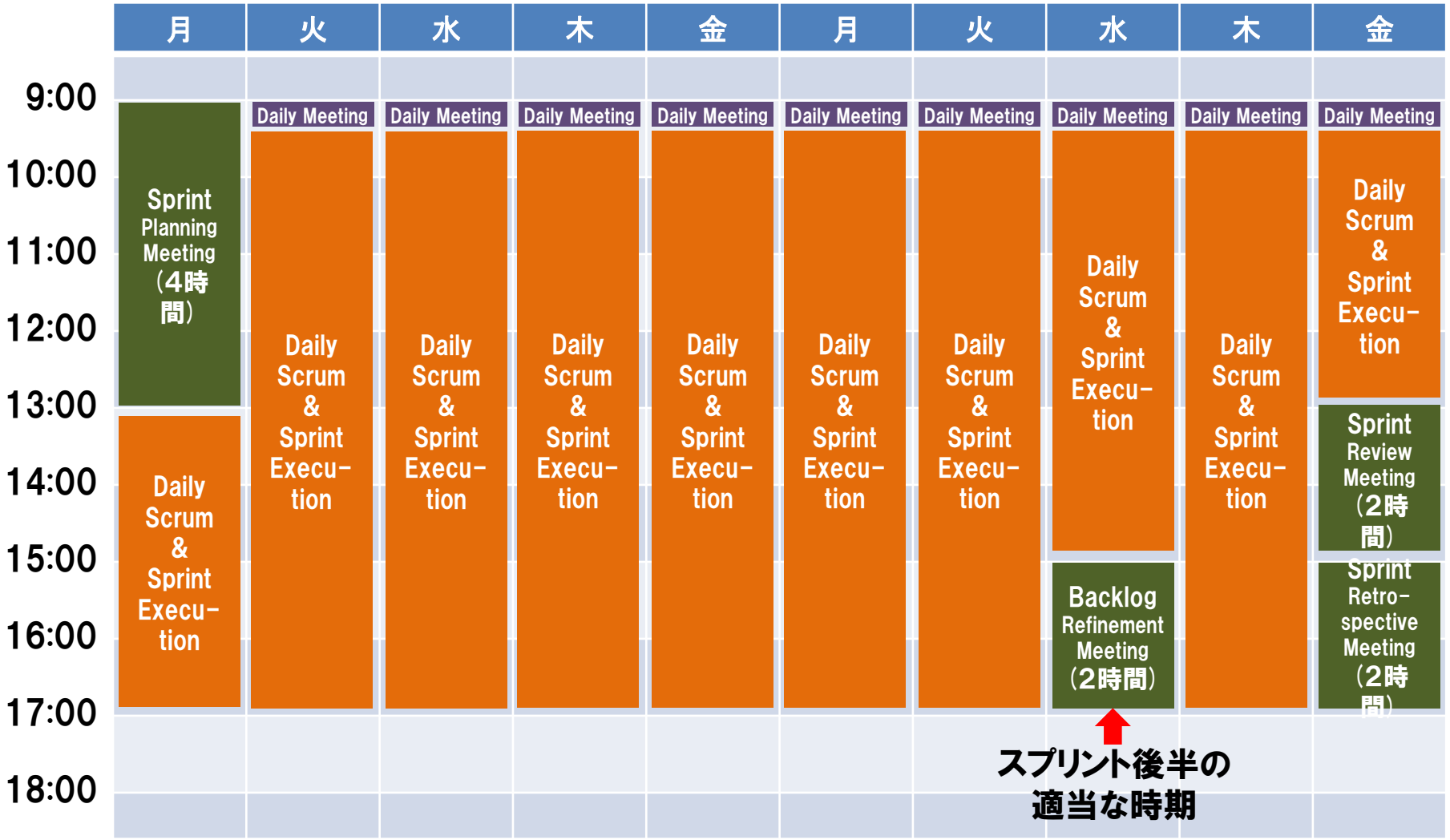
（Scrumで言うスプリントとイテレーションは同じ反復の意味）



3-2 Scrumのイテレーション(スプリント)スケジュール

HITACHI
Inspire the Next

Scrumのスプリント(2週間の場合)の時間目安



3-3 Scrumの会議体

HITACHI
Inspire the Next

タスクの詳細化を行い、イテレーション期間の作業を明確化する

イテレーション計画

当該イテレーションで実装するプロダクトバックログアイテムを決定し、イテレーションバックログに詳細化する

仕様確認レビュー

当該イテレーションで開発したプロダクトについて、要求された仕様通りに作成できているか確認を行う

バックログ リファインメント

プロダクトバックログの優先順位を確認し、次回以降スプリントでの対応可否を決定する

スタンドアップミーティング

チームで、作業状況の確認、問題の報告、タスクの割当てのための15分程度の打合せをする

全員が1分程度で次のことを発言する

① 昨日行ったこと
② 本日行うこと
③ 抱えている課題

レトロスペクティブ

スプリントの反省点、プロセスの改善策などについて議論を行う

KEEP(継続:良かった点)	TRY (挑戦:改善点)
PROBLEM(問題:反省点)	

3-4 プラクティスの効果

HITACHI
Inspire the Next

目的に合わせてプラクティスを選択する

#	プラクティス	早期市場投入	変更対応	生産性向上	品質向上	見える化	リスク回避	スキルアップ
1	反復(イテレーション)	○	○				○	
2	レトロスペクティブ			○	○			○
3	スタンドアップ ミーティング					○	○	
4	フェーズ分割				○		○	
5	ペアプログラミング				○			○
6	テスト駆動開発(TDD)		○		○		○	
7	リファクタリング				○			
8	継続的インテグレーション/デリバリー(CI/CD)		○	○	○	○	○	
9	回帰テスト		○		○		○	
10	チケット管理					○	○	
11	進捗/コスト/品質/変更管理				○	○	○	
12	イテレーション計画			○	○			
13	仕様確認レビュー				○		○	
14	体制						○	○

凡例 ○:効果が期待できる

3-5　第3章のまとめ

HITACHI
Inspire the Next

① まずは　Scrum　を知ろう

② 短い計画された反復を繰り返す

③ 反復ごとにSprint Reviewでプロダクトオーナーと動作するシステムで仕様を確認する

④ プラクティスの効果を知り、採用するプラクティスを決定する

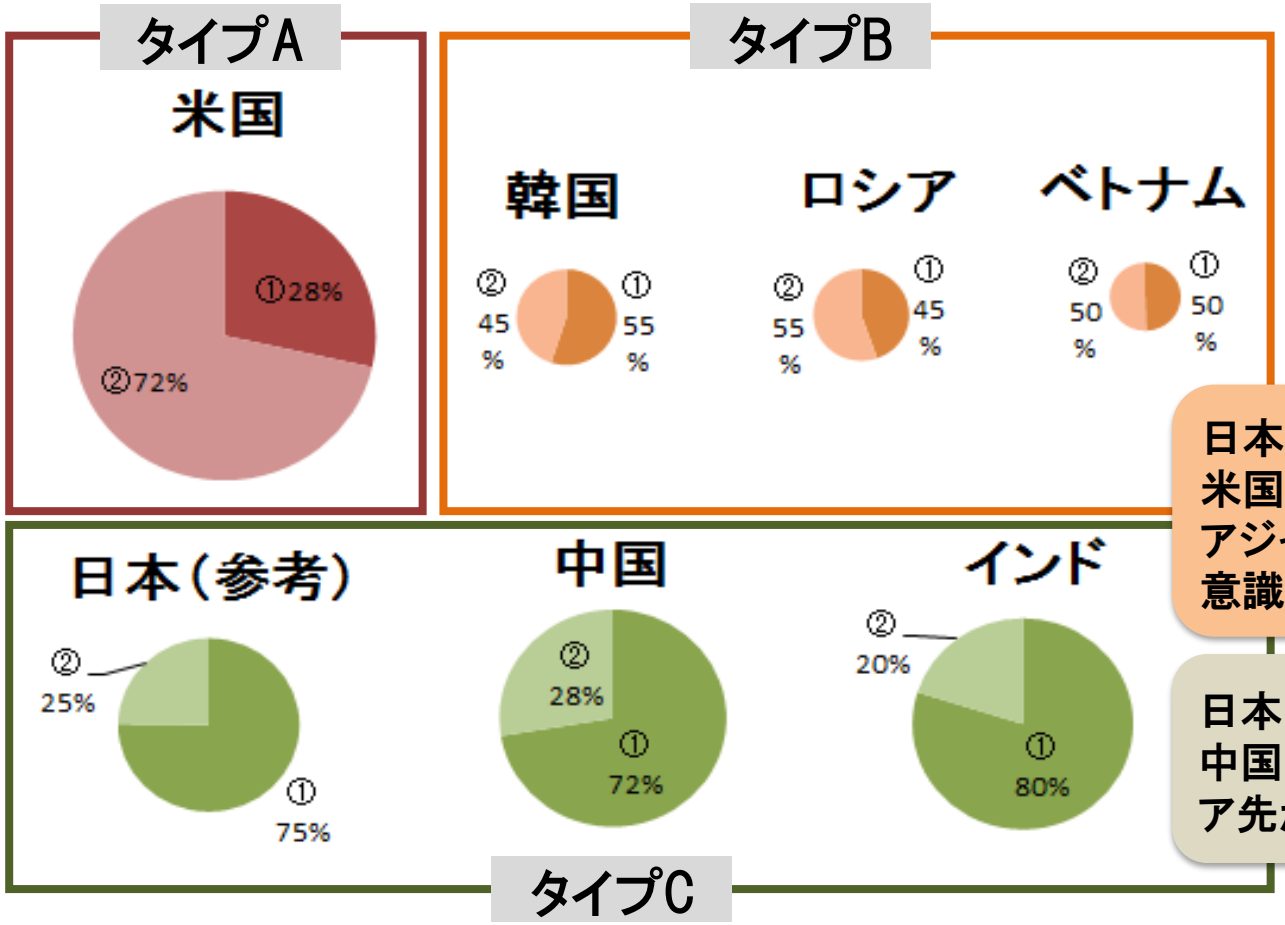
HITACHI
Inspire the Next

4. 日本のソフトウェア開発の特徴

4-1　各国のIT技術者の勤務先比較

HITACHI
Inspire the Next

日本はITサービス企業に75%、米国はユーザ企業に72%



凡例
① ITサービス企業に勤務
② ユーザ企業に勤務

日本は、極端な委託開発傾向
米国は、極端な内製傾向
アジャイル手法には、委託開発を意識したものが少ない

日本は、大手SIベンダが多い
中国・インドのSIベンダはオフショア先が多い

出典：各国統計資料（米国労働省労働統計局等）
公知情報（NASCOMM、アジア情報化レポート、RUSSOFT、IPA IT人材白書2010）
その他：「ガートナー／Enterprise IT Spending by Vertical Industry Market, Worldwide, 2008-2014, 2Q10 Update」の内部サービスコスト、及び「平均給与単価（後述）」に基づく推計値から（独）情報処理推進機構：IT 人材白書 調査報告書 グローバル化を支えるIT 人材確保・育成施策に関する調査，p.44，図表3-14，http://www.ipa.go.jp/files/000010609.pdf（2016年8月1日現在）に纏めたもの

4-2 日本のソフトウェア開発の特徴

HITACHI
Inspire the Next

日本独自の文化に対応が必要である

① 予算化の方法

- 年度または期単位に全体予算を申請する場合が多い（随時予算追加困難）

② 契約上の課題

- 指揮命令系統（ユーザー企業との全員同席は問題となりやすい）
- 請負契約の場合、成果物責任による要件追加・変更に対する契約が必要
  （変更受入れ困難）

大規模開発では、工程ごとの縦割りの体制となることがある

- 要件確定、設計、製作、製作のオフショア発注など工程が縦割りとなるケースが多い
  （要件定義、設計、製作の同時進行困難）
- 設計者とプログラマという役割分担が多く、設計書で仕様伝達となる
  （ドキュメント省略の限界）

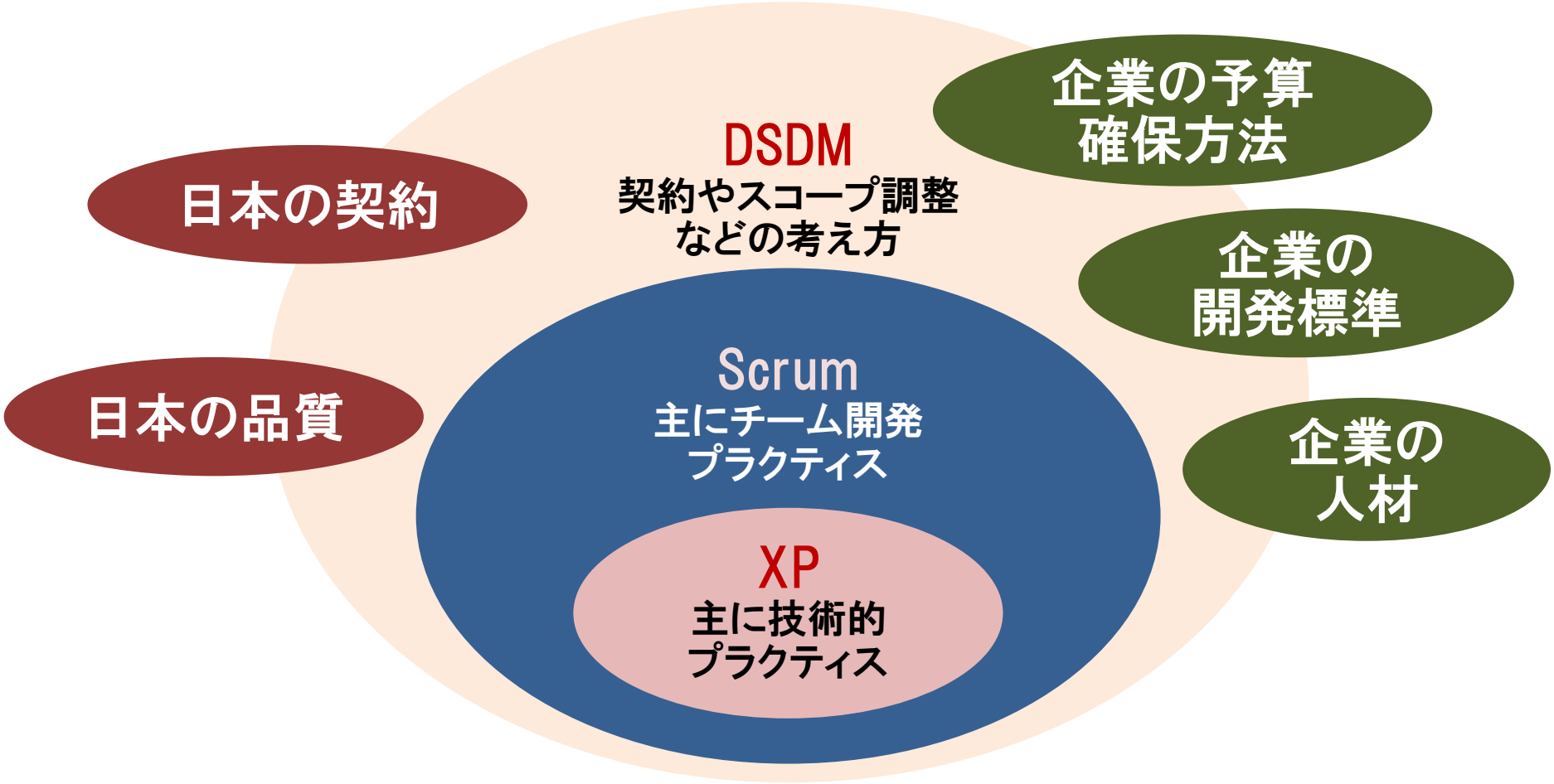
③ 品質の厳格さ

- 「バグは付きもの」では済まされない「バグはあってはならない」
- テスト十分性の結果報告が求められる。（テスト工程省略困難）

4-3 アジャイルに必要な追加の考え方

HITACHI
Inspire the Next

Scrumではカバーできない範囲がある



DSDM: Dynamic Systems Development Method
XP:eXtreme Programming

4-4 DSDMのフレームワークの8つの原則

HITACHI
Inspire the Next

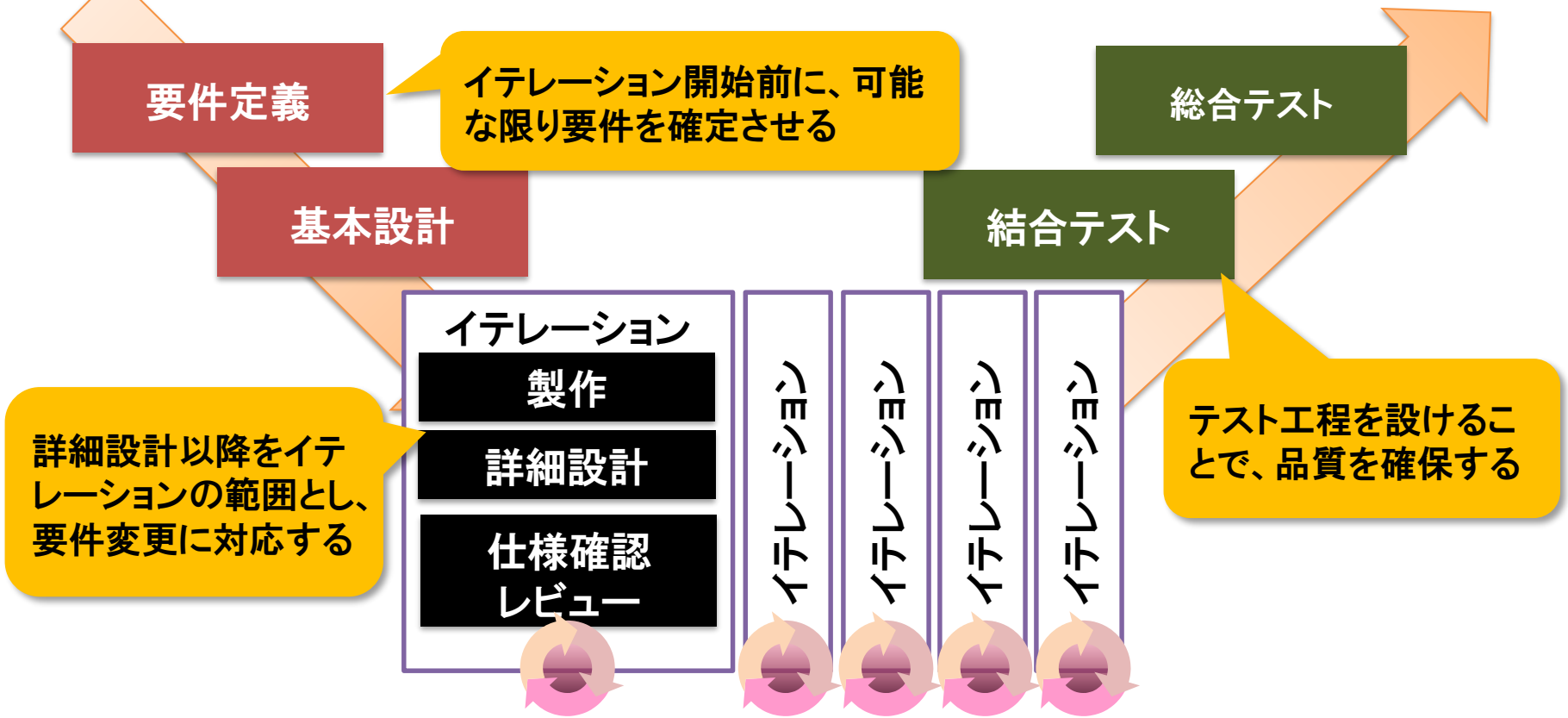
- ビジネスニーズの焦点を当てる
- 予定どおり提供する
- 協力する
- 品質は絶対に妥協しない
- 基盤となる機能から段階的に開発する
- 反復開発
- 明確で継続的なコミュニケーションをする
- コントロールをしっかり行う

4-5 ハイブリッド アジャイルの考え方（基本パターン）

HITACHI
Inspire the Next

日本の委託開発でもアジャイル適用を可能にする

■ ウォーターフォールに、アジャイルによる反復開発を部分的に適用する
（詳細設計～単体テストを適用した場合の例）



- 委託契約を考慮している
- 顧客への納品に向けた作業を考慮している

4-6 ハイブリッドアジャイルの目的

HITACHI
Inspire the Next

変更対応、手戻り作業防止、利用者満足度向上に効果がある

目的

- 要件の追加や変更に対応する
- 仕様齟齬による手戻り作業防止
- 利用者満足度の向上
- リリースまでの期間を短縮
- 開発コストの削減（ムダな作業の排除）

- ハイブリッドアジャイルでは、大きな期待はできない
- 現状の開発プロセスを少しでも改善する

方法

- 一度仕様を確認するが、追加や変更は許容する
- 仕様確認は開発中にも頻繁に行う
- イテレーションごとにスコープ調整を行う
- テストエ程は、しっかり行う
- ムダな作業を認識し、排除する

4-7 アジャイルの適用パターン（一例紹介）

HITACHI
Inspire the Next

目的によって「アジャイル」の適用パターンはさまざまである

CASE 01 早期リリース、開発効率向上

CASE 02 変更受入れ、仕様認識齟齬のリスク低減

CASE 03 UX向上、ユーザビリティ向上

CASE 04 アジャイルとウォーターフォールの並行

4-8 第4章のまとめ

HITACHI
Inspire the Next

① 米国は内製主体、日本は委託主体

② 法律は遵守する

③ アジャイルは委託開発には敷居が高い

④ 委託開発時には、ハイブリッドアジャイルなどで少しでも改善を考える

HITACHI
Inspire the Next

5. 優先度付けの考え方

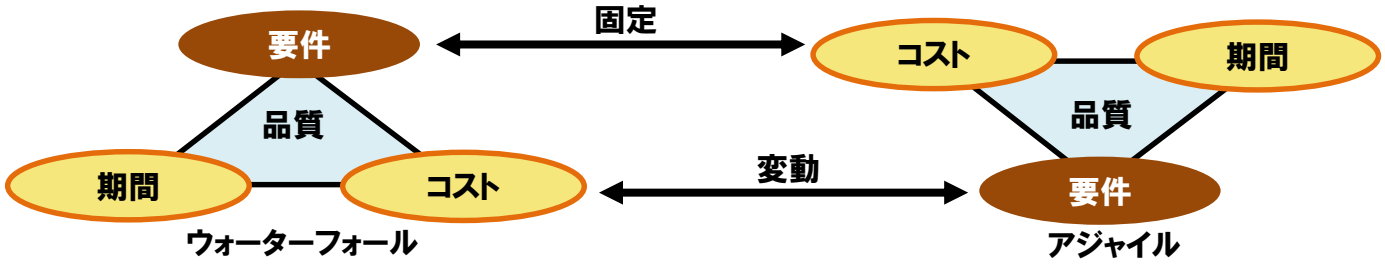
5-1　DSDMの推奨する作業優先度の付け方

HITACHI
Inspire the Next

DSDM：Dynamic Systems Development Method

時間とコストを固定し、要件の量をコントロールする

- ウォーターフォールの場合、要件を固定し、時間とコストをコントロールする
- アジャイルは「タイムボックス型」と呼ばれ、コストと時間を固定し、その範囲内で実現可能な要件をコントロールする



MoSCoW法による推奨優先度付け

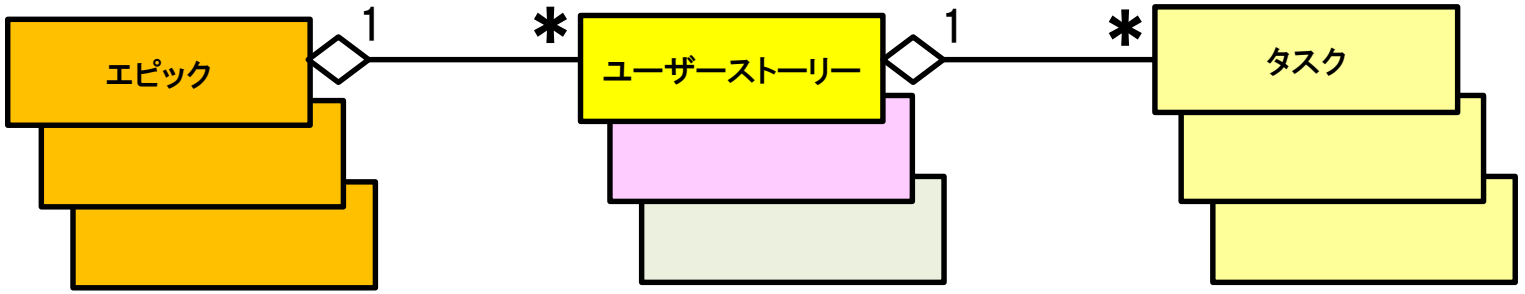
MUST：必ず実施（なければ業務が成り立たないもの）　[60%以内が目安]
SHOULD：可能な限り実施（業務に特に影響は無いが、あると効果的（価値を生む）なもの）
COULD：実施すると良い（業務に特に影響は無く、効果（価値）は少ないがあると良いもの）
　　[20%程度が目安]
WOULD：特に必要ない（開発対象外にするもの）

100%

優先度の高いMUSTから対応を行うのが原則であり、（　　）内の記述は判定の目安である

5-2　チケットでバックログリストに優先度付けする

HITACHI
Inspire the Next



- 一般的には、ユーザーストーリーで優先度を付ける
- ユーザーストーリーが多すぎる場合は、優先度付け作業の効率を考えて、ユーザーストーリーを纏めた一つ上のレベルである「エピック」で行っても良い

● MoSCoW法による優先度付けの例

エピック	ユーザーストーリー	優先度
アンケートを作成する	誰でもアンケートを作成することができる	Must
	質問の回答方法は、ラジオボタン、チェックボックス、テキストボックスを指定できる	Must
	質問に必須回答と任意回答を設定できる	Must
	作成したアンケートは、作成者が削除できる	Must
	作成中のアンケートは、一時保存できる	Should
	過去に作成したアンケートを再利用できる	Could
	EXCELで作成したアンケートを取り込める	Would
アンケートに回答する	:	:
:	:	:

5-3　開発範囲とリリース時期を計画する

HITACHI
Inspire the Next

① MoSCoWで優先度付けする
② 提供する機能のビジネス価値を見積もる
③ 機能を実装するための開発工数を見積もる
④ ①～③を考慮して、最終的に実装する機能と開発計画を作成する

ビジネス価値(K￥/年) ＝
削減効果(時間) ＊ 回数(回/年) ＊ 人数(人) ＊ 全社加工費単価(K￥)

見積もり開発工数(人日) ＝基本設計 から 総合テスト までの見積もり工数

● ビジネス価値算出の例

ビジネス価値　　見積もり開発工数

題名	ビジネス価値	ビジネス価値	単位効果(分	回数/年間	対象者	見積り開発工	基本設計(UI	詳細設計(DE	実装	テスト(UT)	テスト(CT)	テスト(ST)	見積り開発工	見積り開発工
実	0	0	0	0	0	128	0	24	36	36	24	8	128	17
副	1,283	167	5	4	500	121	33	16	24	24	16	8	121	16
副	92	12	60	12	1	168	40	24	36	36	24	8	168	23
副	31	4	10	24	1	275	19	48	72	72	48	16	275	37
副	1,283	167	20	1	500	161	33	24	36	36	24	8	161	22
副	15	2	10	12	1	202	26	32	48	48	32	16	202	27
副	370	48	60	48	1	147	19	24	36	36	24	8	147	20
基	0	0	0	0	0	372	60	56	84	84	56	32	372	50
匠	0	0	0	0	0	50	10	8	12	12	8	0	50	7
	3,075	399				1624	240	256	384	384	256	104	1,624	218

ユーザーストリー
⋮

5-4 意識を変える

HITACHI
Inspire the Next

ビジネス価値の高い機能を優先して、最低限の機能から数カ月単位に早くリリースする

コミュニケーションは密にする。毎日の状況確認、週単位の動作するシステムでの状況確認

● 某システムのリリース例

4カ月	5カ月	5カ月	5カ月
• 全社共通機能	• 部門独自機能 • グループ会社対応	• 残機能の追加 • UI/機能改善	• 他システム連携
最低限の機能	利用範囲の拡大	ブラッシュアップ	他システム連携

5-5　第5章のまとめ

HITACHI
Inspire the Next

① ビジネス価値で優先度付けする

② コストと期間を固定する考え方にする

③ 優先度の低いものは、見送る場合もある

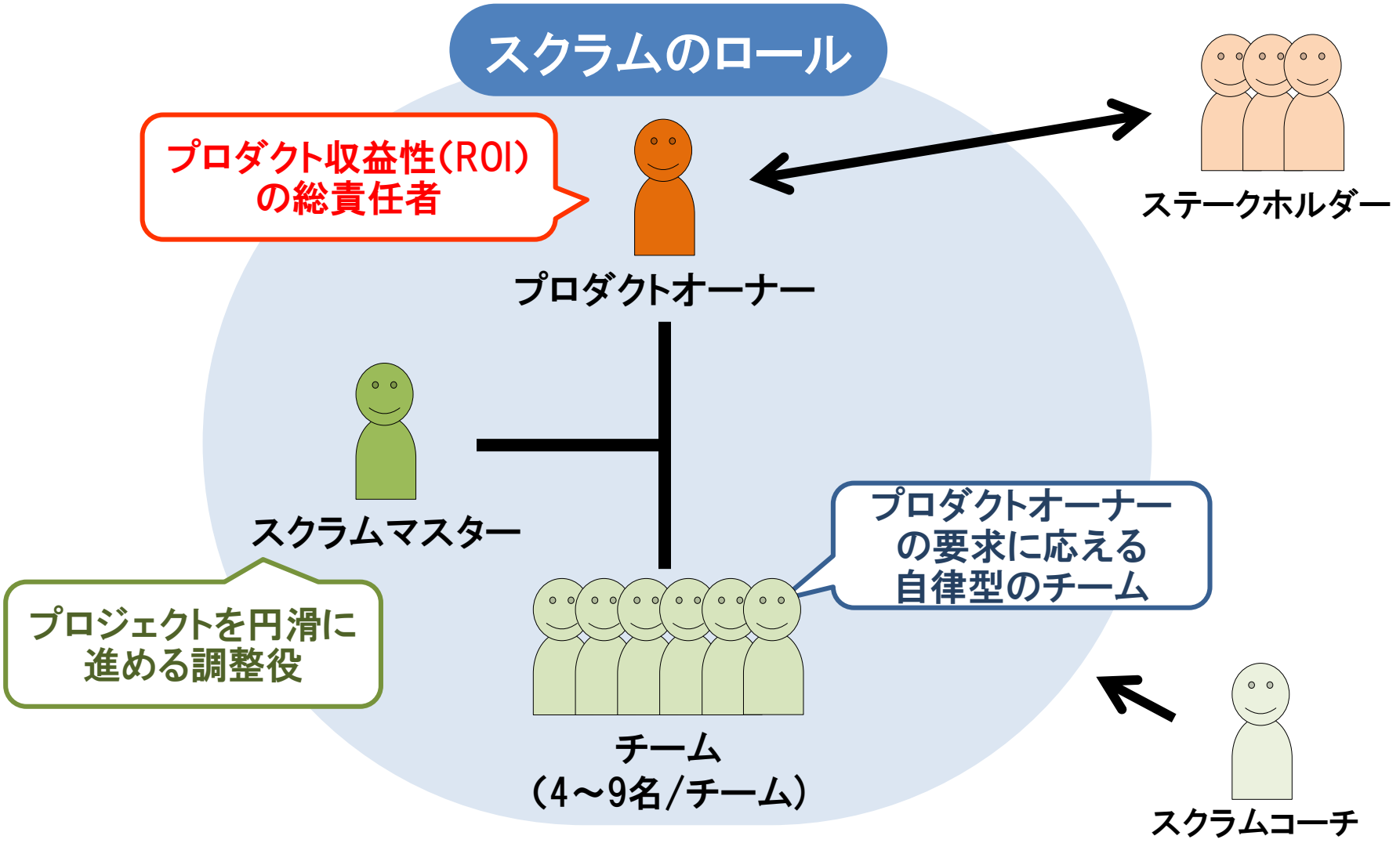
④ 段階的に早くリリースして、ビジネス価値を早期に得る

HITACHI
Inspire the Next

6. 体制の考え方

HITACHI
Inspire the Next

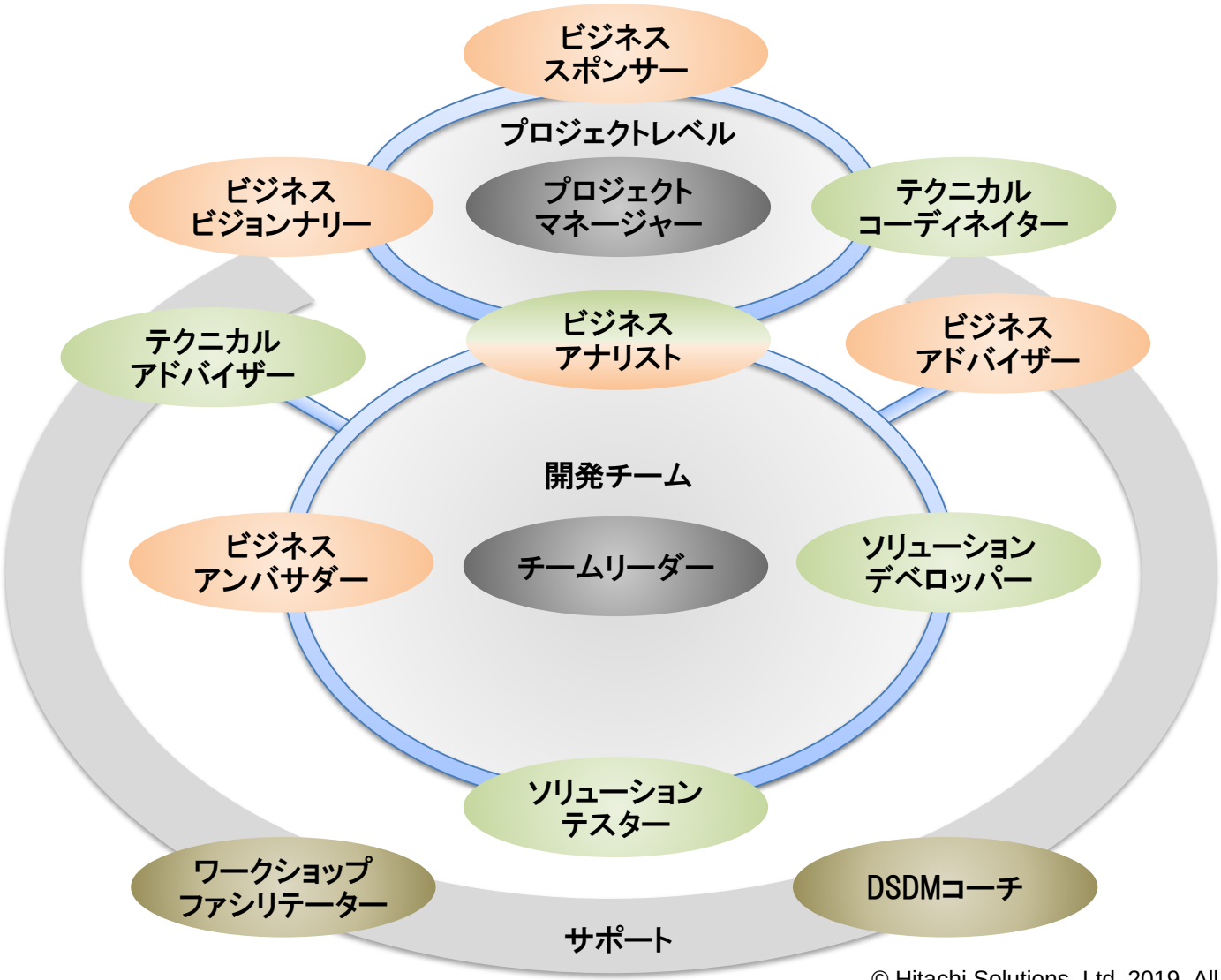
小規模な内製開発向きである



6-2 DSDMの体制

HITACHI
Inspire the Next

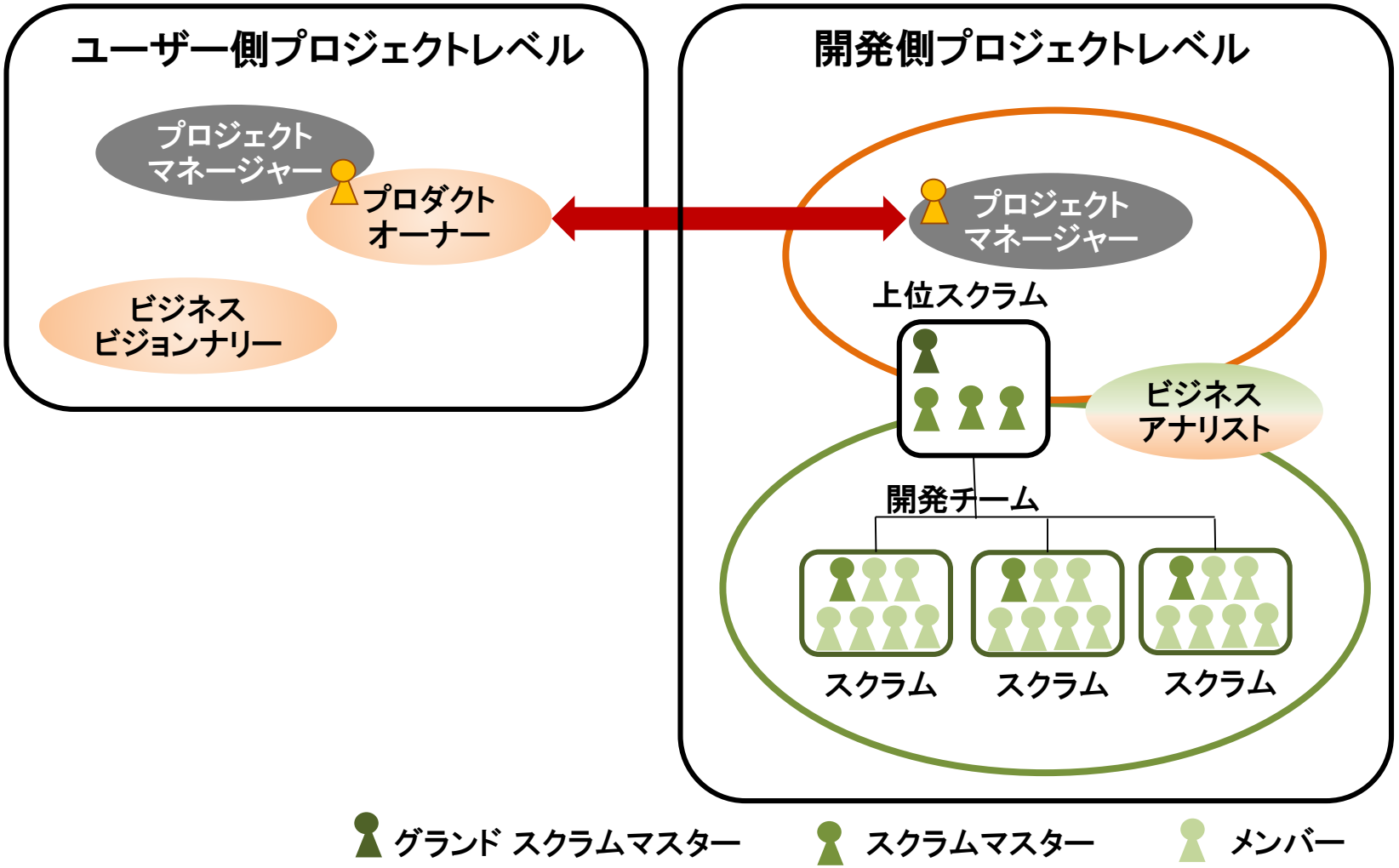
大規模や委託開発を意識した体制である



6-3 大規模・委託開発時のプロジェクト体制例

HITACHI
Inspire the Next

アジャイル手法に拘らず、プロジェクトに適した体制をにする



6-3 主なプロジェクトメンバの役割

HITACHI
Inspire the Next

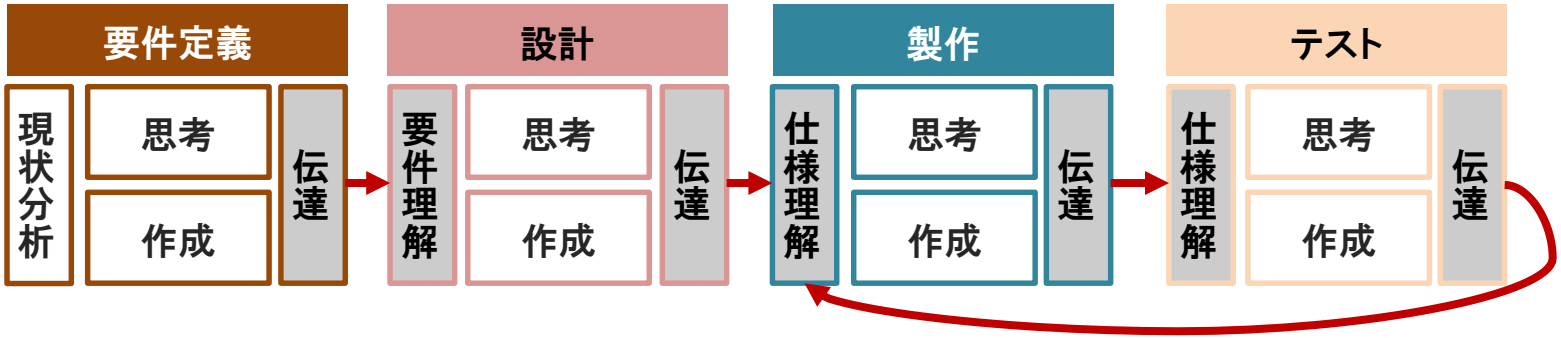
役割は兼任しても構わない

役割(ロール)	作業概要
・プロダクトオーナー ・ビジネススポンサー	・プロジェクトの発注者で成果物の決定を行う
・プロジェクトマネージャー	・プロジェクトの責任者 ・プロジェクト計画を策定し、それを実行するための体制を整える ・人材や資源、予算、スケジュール、品質などを管理し、プロジェクトを円滑に進める
・ビジネスビジョンナリー ・ビジネスアンバサダー	・要件を作成する人／要件を伝える人
・ビジネスアナリスト	・要件をまとめて仕様を作成する ・実装する機能の調整などを行い、仕様を確定させる ・開発者に仕様を伝える役割も担う
・スクラムマスター ・チームリーダー	・チーム内のメンバーをまとめ ・成果物に責任を持ち、発生する技術的な課題を解決する
・メンバー ・ソリューションデベロッパー ・ソリューションテスター	・機能を実装するプログラミング担当者 ・開発ツールの環境設定、データベース設計や構成管理などを行う担当者も含む

HITACHI
Inspire the Next

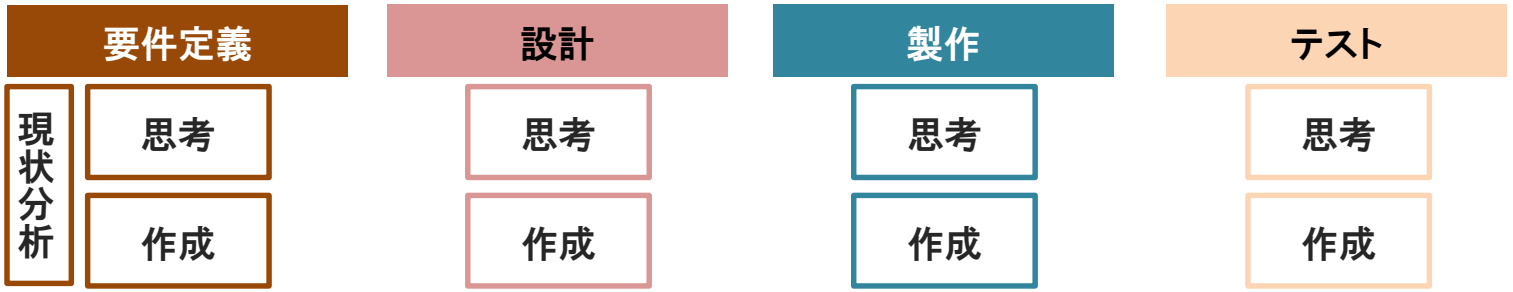
理想は同一人物で全工程を担当する

工程ごとに担当者が異なる場合



伝達と理解の
時間はムダ

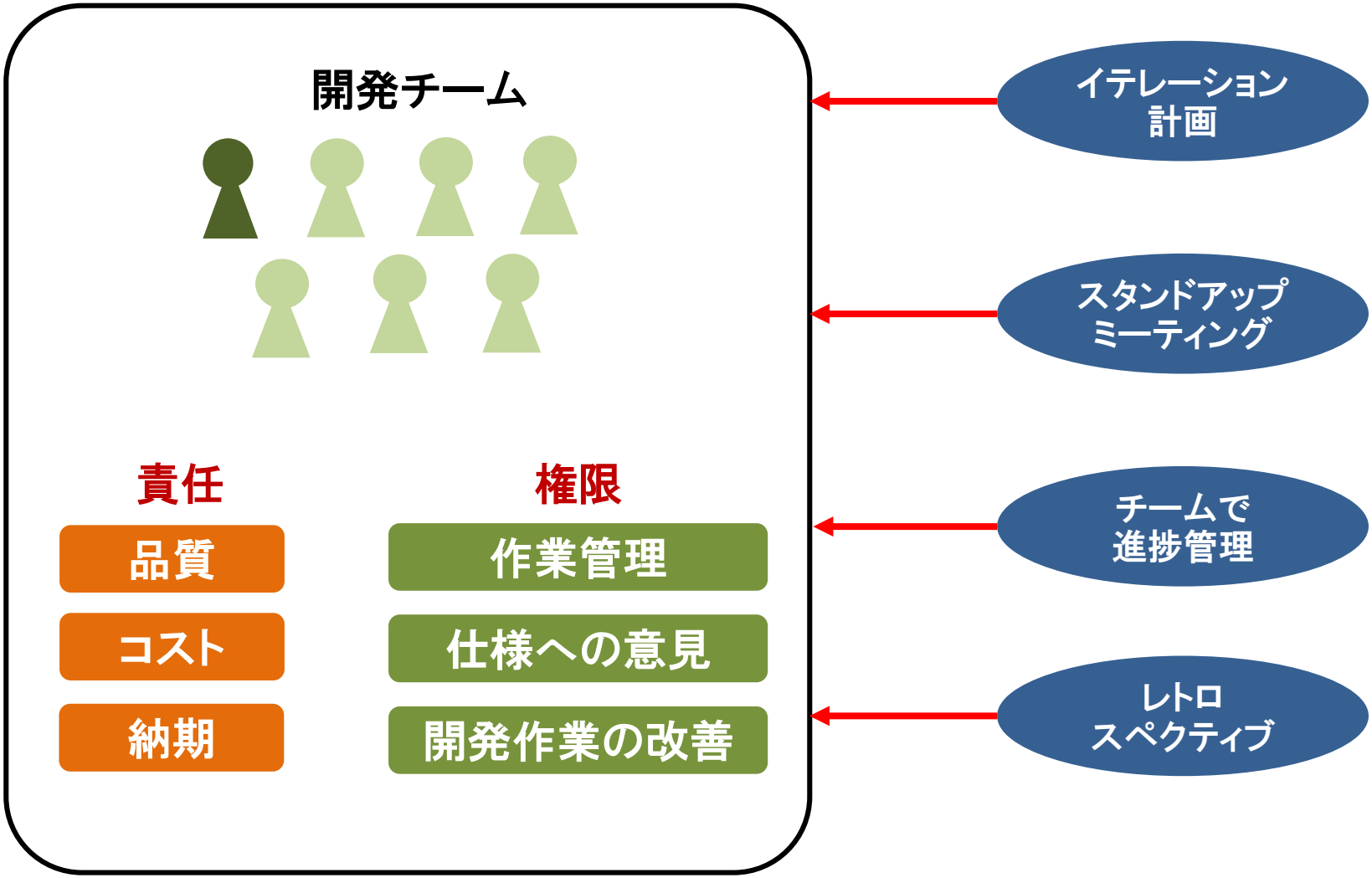
全工程が同じ担当者の場合



6-5 自律型組織

HITACHI
Inspire the Next

チーム意識を強く持たせ、責任と権限を与える



6-6 第6章のまとめ

HITACHI
Inspire the Next

① 委託開発であれば、委託元と委託先の体制が必要

② 大規模プロジェクトでは、階層化や専任チームを設ける

③ チームは少人数にする

④ 自律型組織にする

7. 進捗管理の考え方

HITACHI
Inspire the Next

見える化と管理効率を上げる

① ガントチャート

- ウォーターフォールでよく使われる
- 機能やWBS単位に作業の開始と終了が線で表され、作業単位に進捗を管理する

② カンバン

- アジャイルプロセスでよく使われる
- プロジェクトルームの壁などにストーリーカードを貼り付ける(ツールでもよい)
- 作業状況が縦に分割され、進捗によりストーリーカードを左から右へ移動させていく

③ バーンダウンチャート

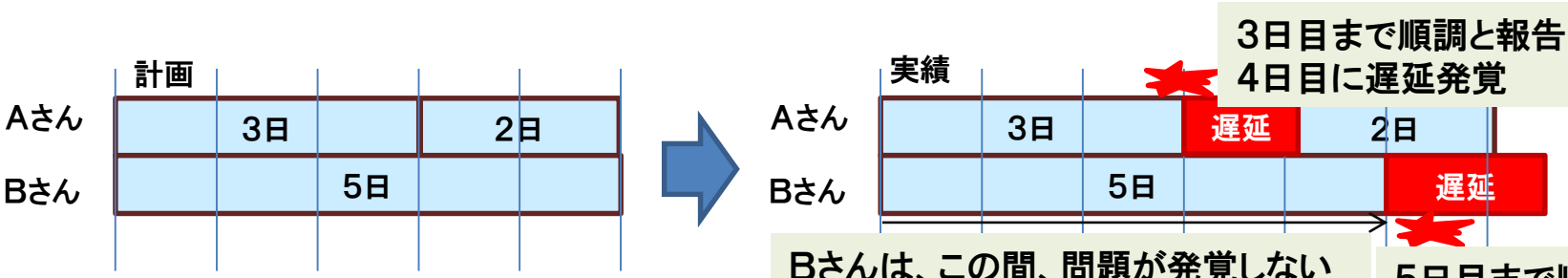
- アジャイルプロセスでよく使われる
- イテレーション単位、チーム単位の残作業量をグラフ化し、チームの開発スピードを管理する

HITACHI
Inspire the Next

個人の進捗ではなく、チームの開発スピードを管理する

ガントチャート

WBSで機能単位に個々のノルマをガントチャートで管理する

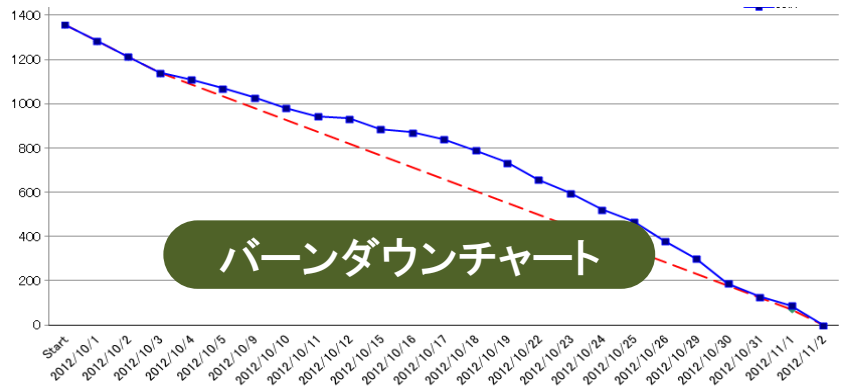


進捗と問題の発覚が遅れやすい。遅延発生により計画は全て崩れる。

ガントチャートは、全体スケジュールを報告するのに向いている。

チケット管理

チームでタスクを消化し、作業時間を日々管理する

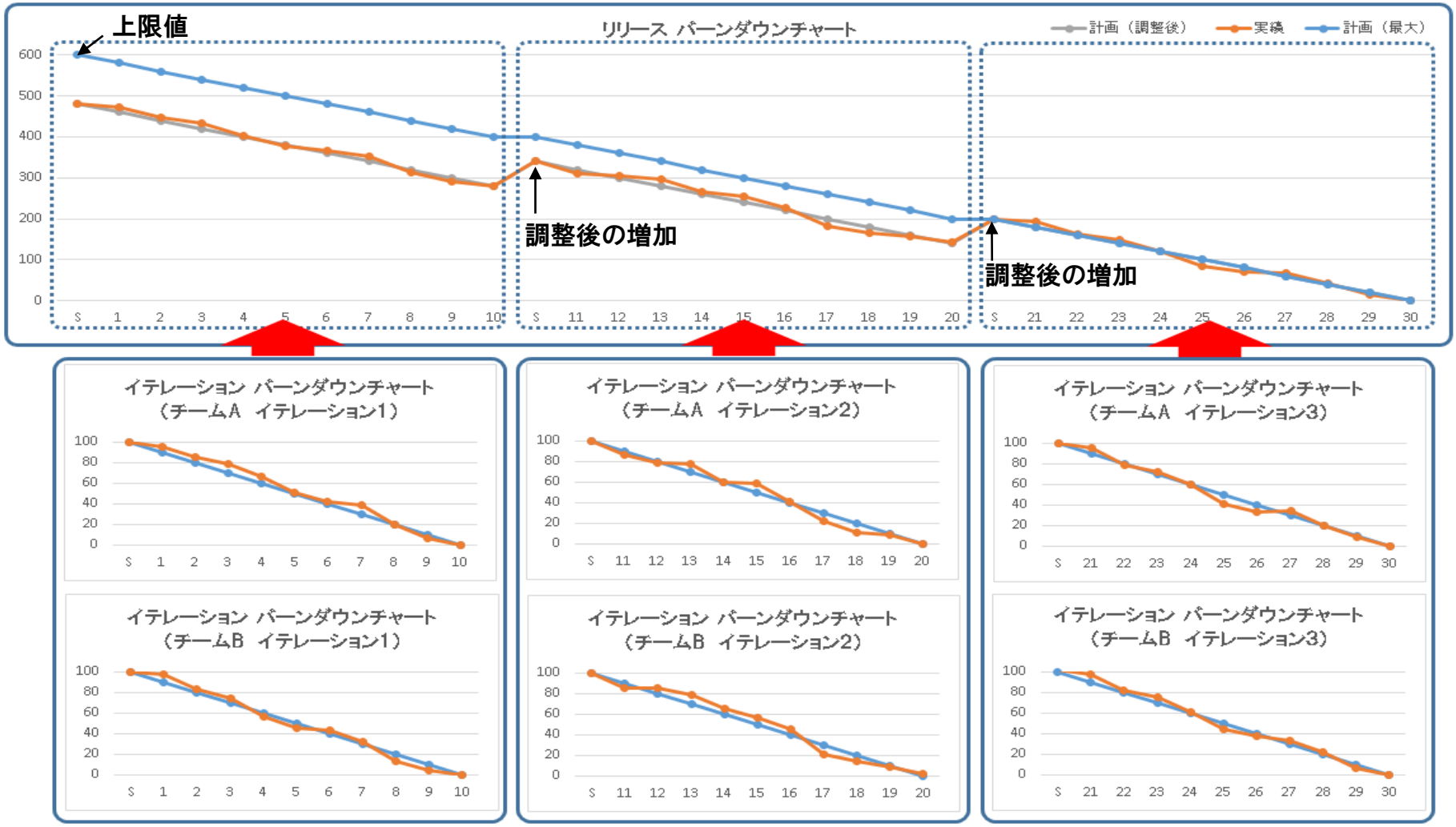


チケット管理とバーンダウンチャートは、日々の進捗把握と開発スピードコントロールに向いている

7-3 リリースバーンダウンとイテレーションバーンダウン

HITACHI
Inspire the Next

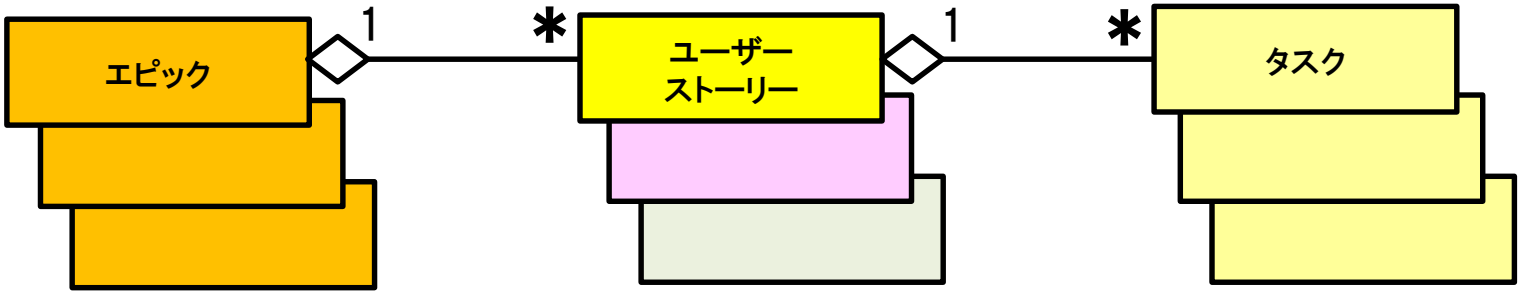
プロジェクト全体とチームごと、イテレーションごとを管理する



7-4 チケット管理

HITACHI
Inspire the Next

管理できるようにデータ化したものがチケット(付箋やツール)である

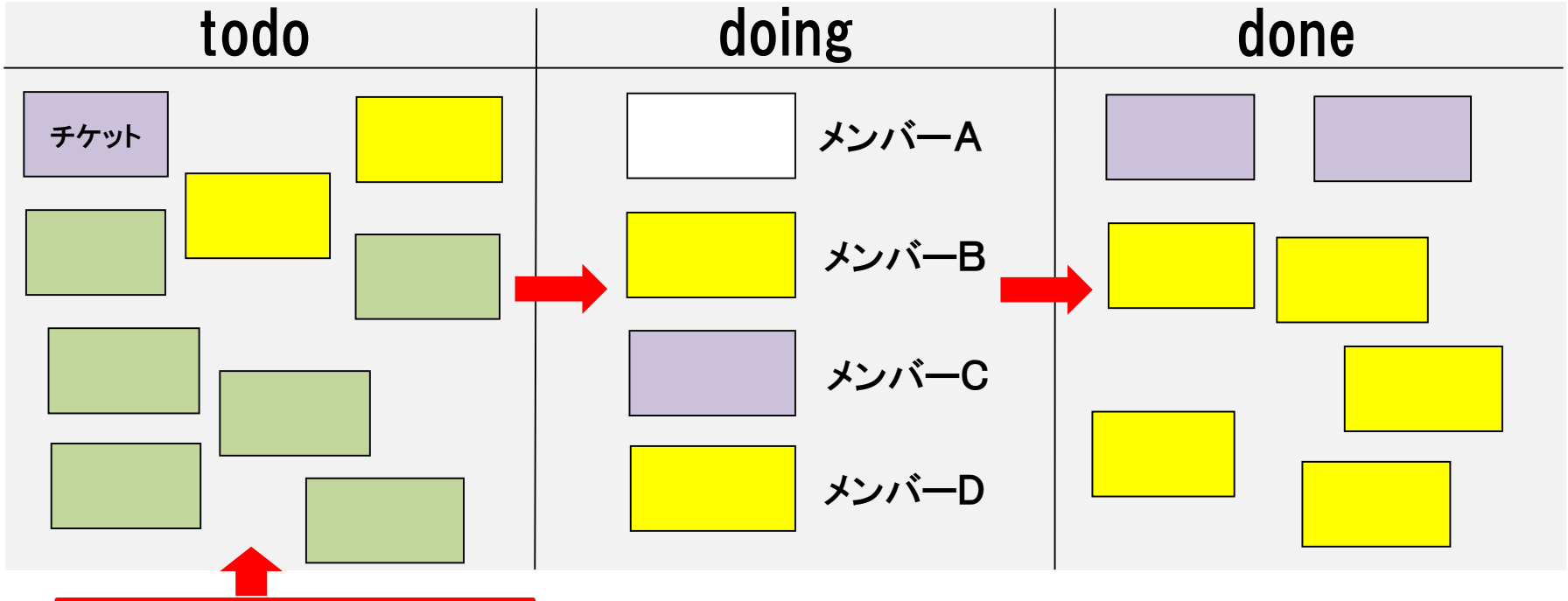


エピック	ユーザーストーリー	タスク
大まかな要件	機能要件	具体的な作業
例： ・実施依頼発信 ・実施依頼の回答 ・部門実施状況の確認 ・全社実施状況の確認 ・実施依頼の終了 ・実施依頼の削除 ：	例： ・スタッフ部門の人が実施依頼を作成する ・作成した実施依頼を上長が承認する ・実施依頼を発信する ：	例： ・仕様設計 ・実施依頼作成画面の作成 ・ロジック作成 ・テストコード作成 ・単体テスト実施 ：

バグチケットや意見チケットなど何でもチケット化はできる

HITACHI
Inspire the Next

カンバンはプル型、開発者の最適ペースを可能にする



- カードウォールやタスクボード は、進捗を見える化するだけのツール
- カンバンは、**プル型システム**（後工程引き取り方式） その結果、最適ペースを可能にする
- カンバンによる進捗の管理は、タスクボードによって見える化することができる

委託開発の契約やワークスタイル改革に
たいへん重要な考え方である

7-6 チケット粒度の調整

HITACHI
Inspire the Next

チケットの数を減らして管理しやすくする

ユーザーストーリー

ユーザーストーリーでは粒度が大きく進捗が解りづらい

ユーザーストーリー

- タスク(設計)
- タスク(画面)
- タスク(ロジック)
- タスク(データ)
- タスク(テストコード)
- タスク(テスト)

タスクに分割するとチケットが多くなりすぎる

ユーザーストーリー
45%

事例

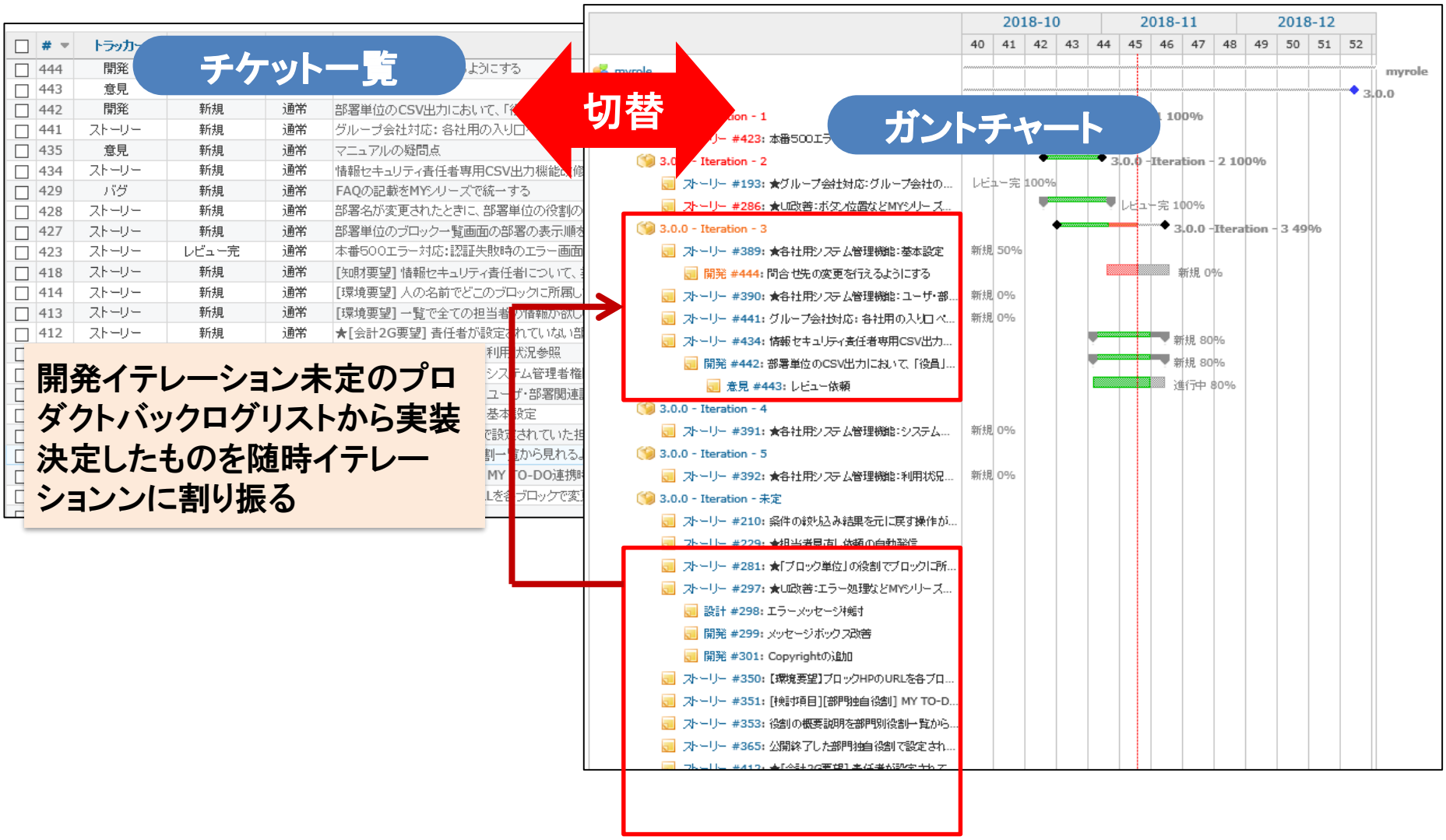
マイルストーン	重み(例)
設計	15%
画面	15%
ロジック	15%
データ	15%
テストコード	20%
テスト	20%

少ないチケットで進捗を把握しやすい

7-7　バーンダウンチャートを仕様しない進捗管理例

HITACHI
Inspire the Next

2，3人の開発者であれば、ガントチャートでも良い



7-8 第7章のまとめ

HITACHI
Inspire the Next

① 進捗管理方法を使い分ける

② すぐに崩れる進捗計画はムダ

③ イテレーションごとにチームの開発スピードを見る

④ チケット管理で進捗管理力を高める

HITACHI
Inspire the Next

8. 設計ドキュメントの考え方

HITACHI
Inspire the Next

設計書を作成する理由があるものは作成する

① 工程で決めたドキュメントを全ての機能に作成する必要はない

- 単純な機能であれば、ラフな画面仕様だけでもよい
- 複雑な機能であれば、詳細な設計書が必要な場合もある
- 納品物でなければ記述方法の形式に捉われる必要はない

② ドキュメントが誰のために必要なのかで判断する

- 設計者とプログラマが同一人物であれば、設計書は無くてもよいぐらいである
- プログラマが委託先であれば、しっかりとした設計書が求められる

③ 委託開発で、納品ドキュメントとしているものは作成する

- 開発中はラフな記述でも構わない。納品前に清書すればよい

8-2 必要な設計書の作成

HITACHI
Inspire the Next

機能により必要な設計書を作成する

機能	基本設計				詳細設計			
	画面仕様	ユースケースシナリオ	・・・	DB設計	詳細仕様	・・・	クラス図	シーケンス図
ユーザーストーリーA	○	○	・・・	○	△	・・・	×	×
ユーザーストーリーB	○	○	・・・	○	○	・・・	○	○
ユーザーストーリーC	○	△	・・・	△	△	・・・	×	×
ユーザーストーリーD	×	○	・・・	△	×	・・・	×	×
：	：	：		：	：	：	：	：

凡例　○：作成する　△：簡略作成する　×：作成しない

請負契約であれば、前頁の考え方と合わせて、基本設計のXXドキュメントは全て作成、詳細設計のXXドキュメントは必要により作成などの契約にすることも検討する。

8-3　第8章のまとめ

HITACHI
Inspire the Next

① 成果物(設計書)の量を減らす

② 設計書は、変更されるものとして捉えておく

③ 必要な設計書は作成する

④ 納品物であれば、納品までに清書すればよい。
開発中は、ラフな状態でも構わない。

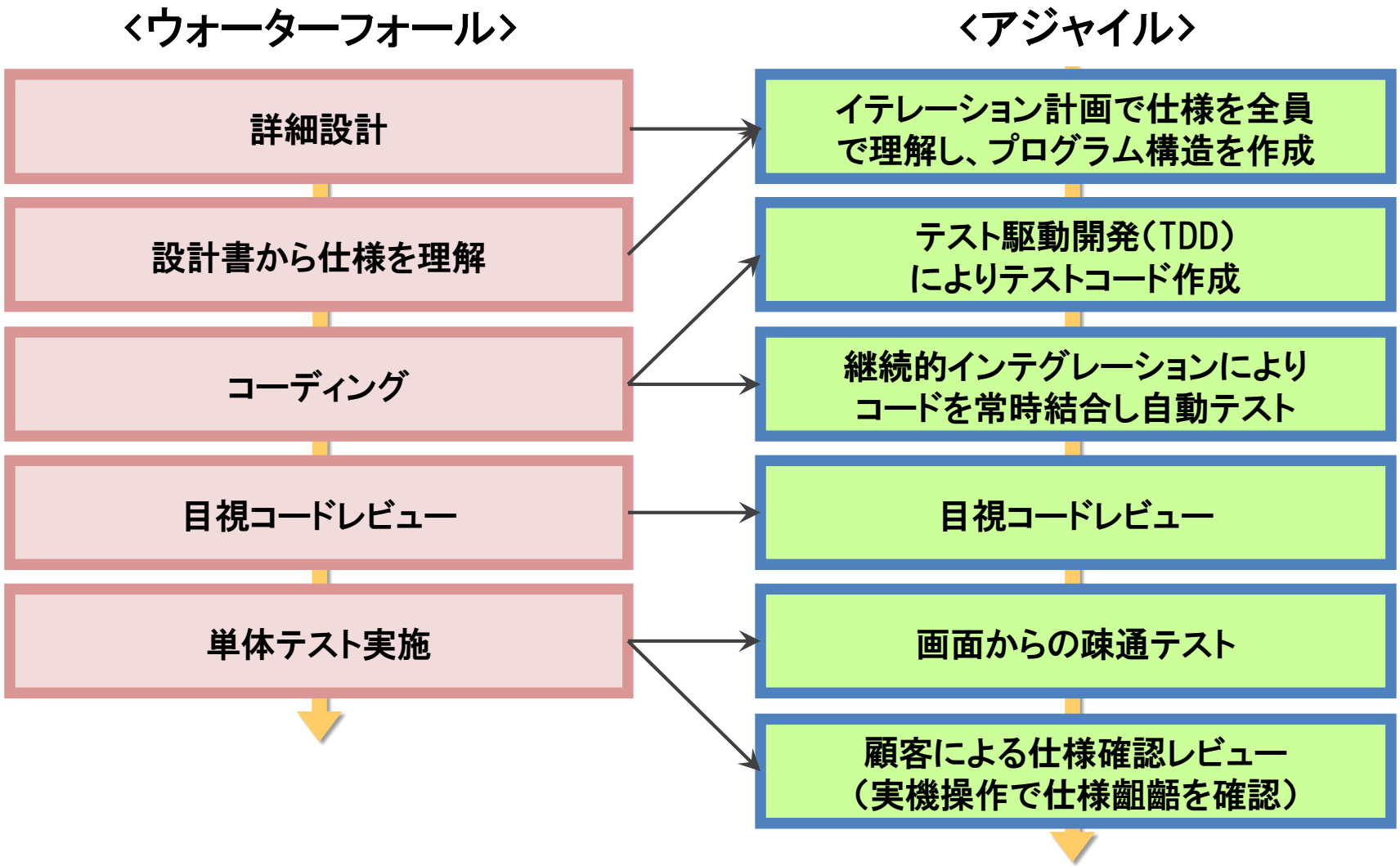
HITACHI
Inspire the Next

9. 品質の考え方

9-1　品質確保手順（詳細設計～単体テストの範囲）

HITACHI
Inspire the Next

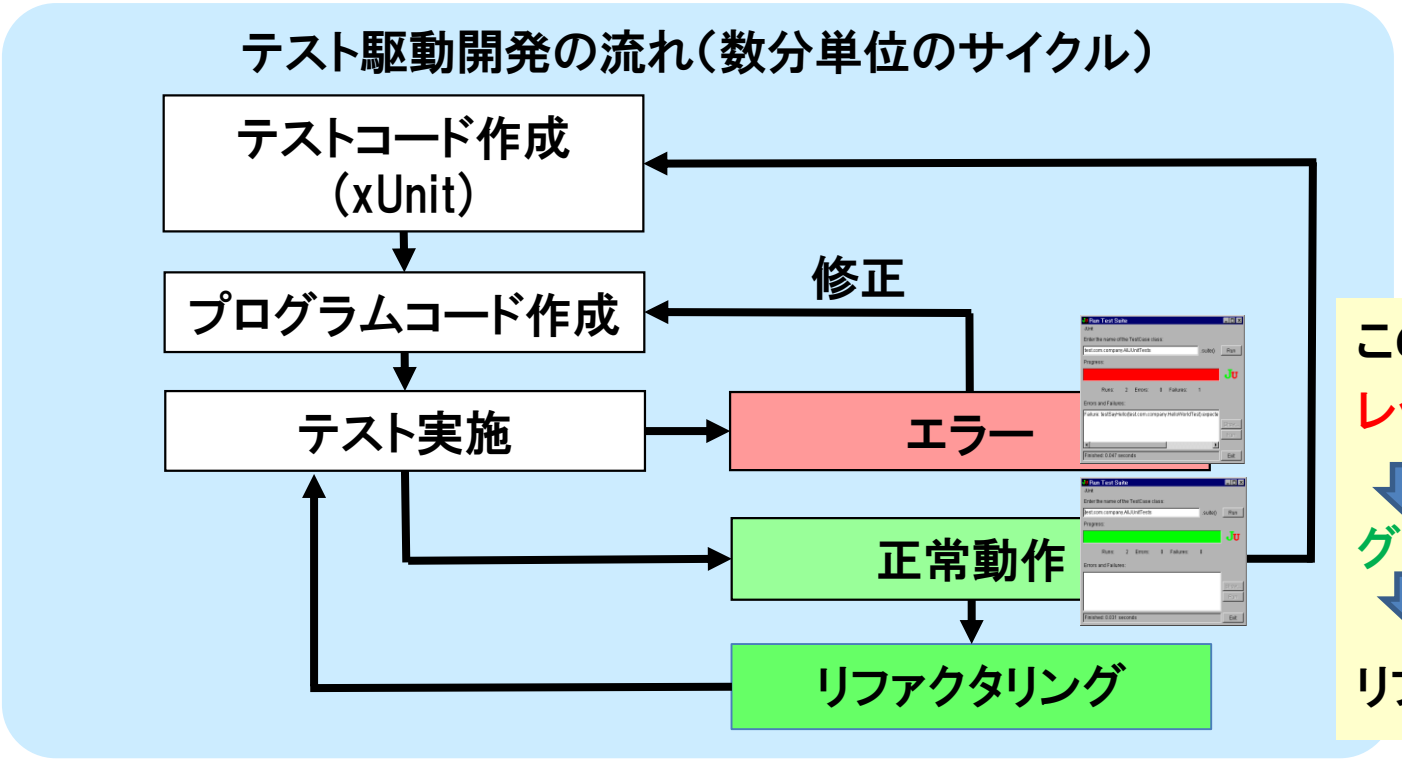
アジャイルは品質向上に効果的である



HITACHI
Inspire the Next

動作するきれいなコードを作成する、設計・開発の手法でもある

- ソースコードを書く前に、テストコードを用意し、テストコードを通過するように プログラムコードを書く方法
- テストケースを追記しながら繰返してソースコードを完成させる
- 常に「動作するきれいなコードを作成(リファクタリング)」しながら完成させるため、テストコードを先に作成するテストファーストがさらに進化し、テスト駆動となった

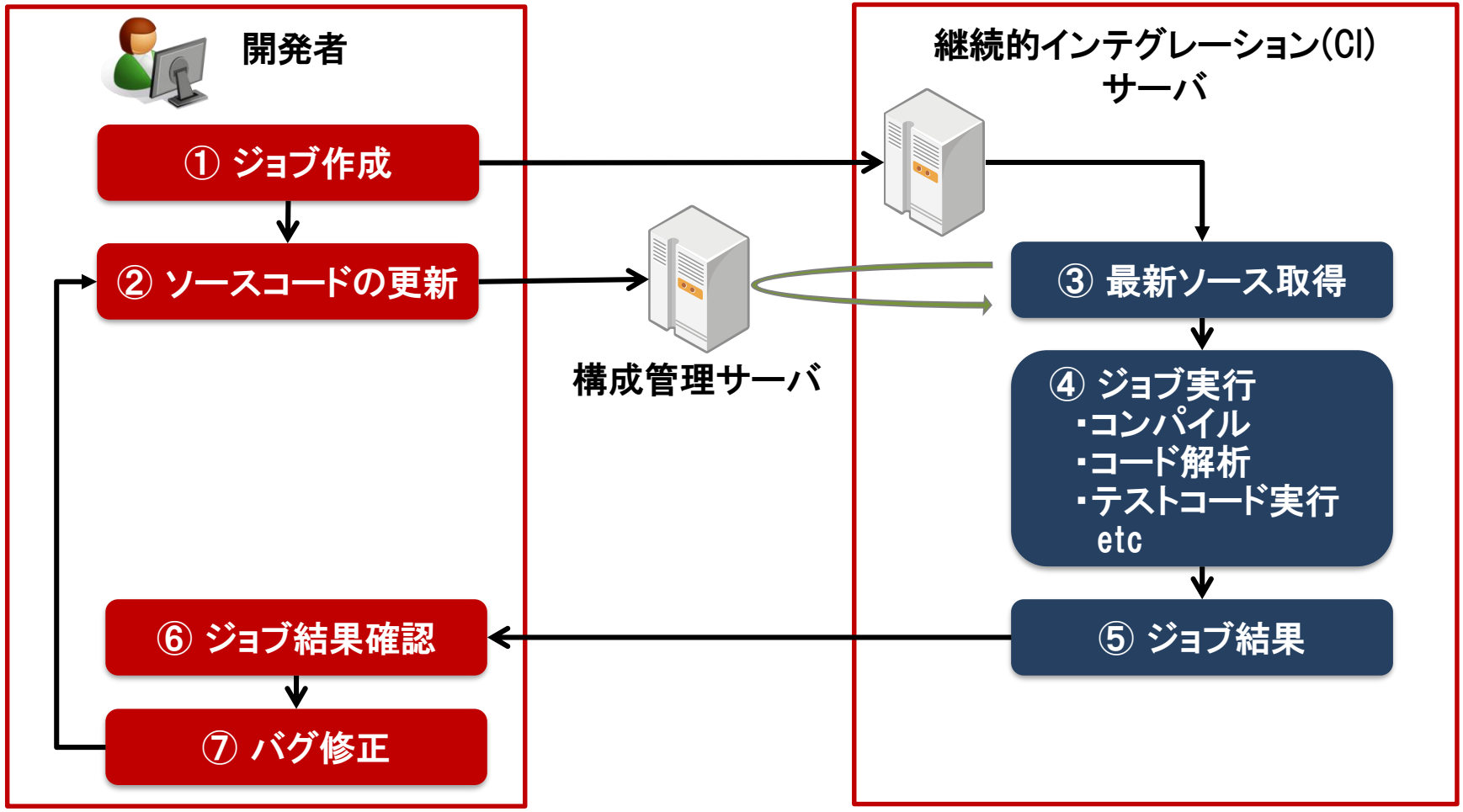


9-3 継続的インテグレーション(CI:Continuous Integration)

HITACHI
Inspire the Next

常に動作する状態を維持し、テストなどを自動化する

- 作成した動作確認済みのコードは常時、継続的に結合する
- 常時、コードのビルド，テストを自動実行する



9-4 アジャイル適用時の品質指標の考え方

HITACHI
Inspire the Next

品質管理の考え方を大きく変える必要はない

事例

テスト工程		テストケース密度	バグ密度	品質評価方法
単体テスト	テスト駆動開発（ロジックテスト ）	0件/Kstep	0件/Kstep	C0/C1メジャー ※100%原則（100%未達の場合は、目視で確認）
	画面からの疎通テスト	従来の単体テスト指標のn%と決め、n件/Kstepとする		指標の達成具合で判断
結合テスト		ウォーターフォールと同様の指標とするn件/Kstep		
総合テスト				

単体テスト

- テスト駆動開発（TDD）の適用部分は、従来の品質指標は当てはまらない
- C0/C1のカバレッジで品質を判断
- 単体テストに画面からの疎通テストを追加し、テストコードでテストできない箇所を補う

結合テスト、総合テスト

- 結合テスト以降は、ウォーターフォール同様の品質指標でテストを行う

当該フェーズの実績をもとにして、次フェーズ以降の品質指標値を見直す

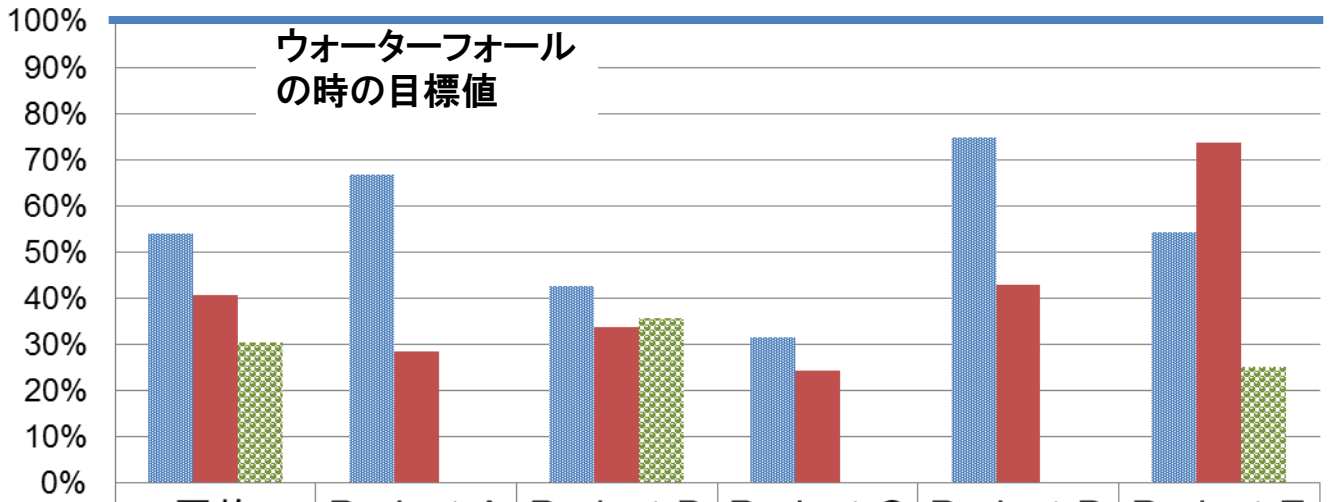
9-5 アジャイル導入時の品質実績

HITACHI
Inspire the Next

ウォーターフォールよりバグが出にくい傾向がある

事例

目標値に対するバグ実績比率



	平均	Project-A	Project-B	Project-C	Project-D	Project-E
単体テスト	54%	67%	43%	32%	75%	54%
結合テスト	41%	29%	34%	24%	43%	74%
総合テスト（ベンダ確認）	30%	0%	36%	0%	0%	25%

アジャイル品質確保のプラクティスを実践したプロジェクト

9-6　第9章のまとめ

HITACHI
Inspire the Next

① アジャイルでもテスト工程を設ける

② テストコードは全てをテストできるわけではない

③ 顧客側の人にも頻繁に動作確認できるようにする

④ テストは厳しく効率よく

HITACHI
Inspire the Next

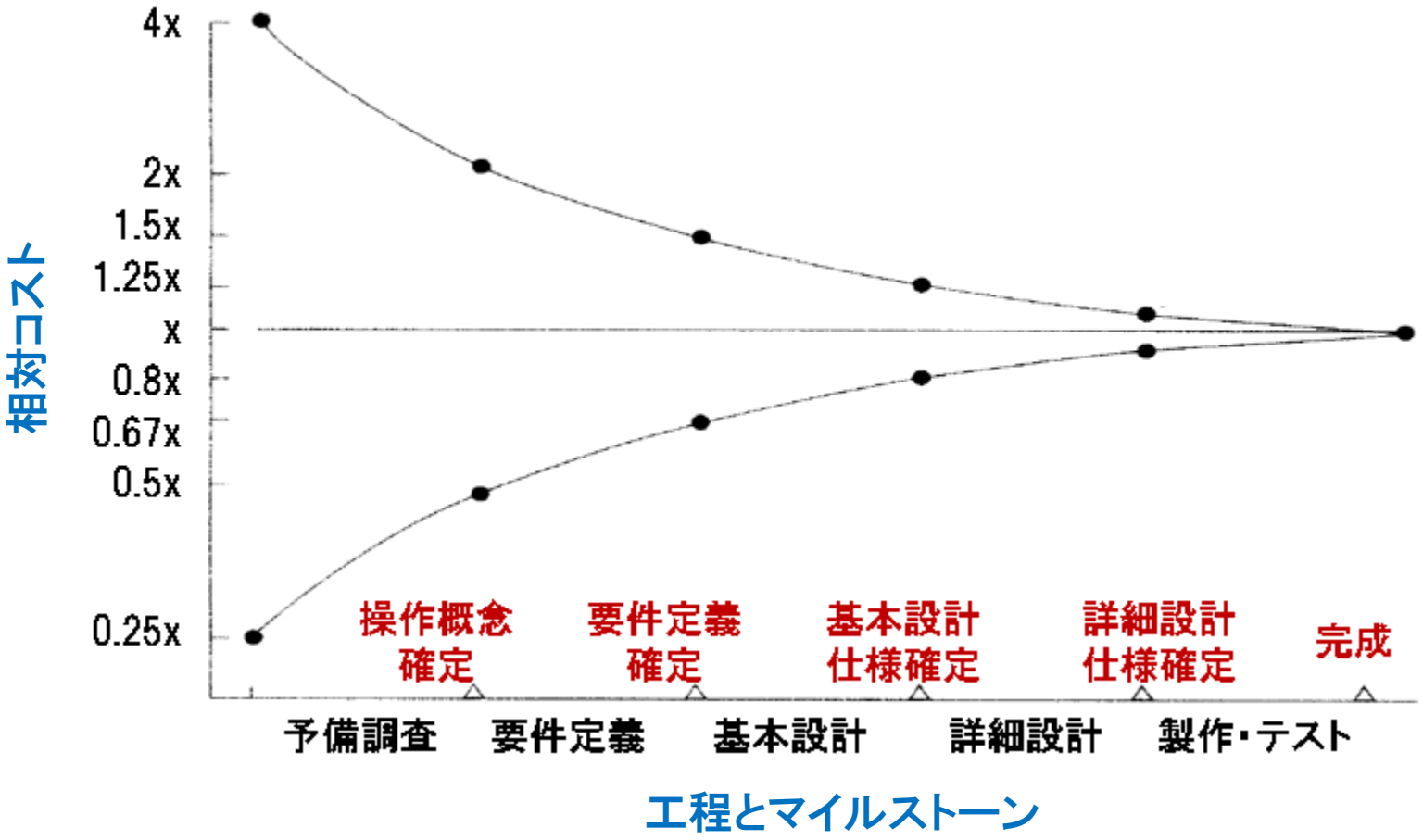
10. 見積もりの考え方

10-1 バリーベームの法則

HITACHI
Inspire the Next

早い段階での見積もりは実績との差も大きい

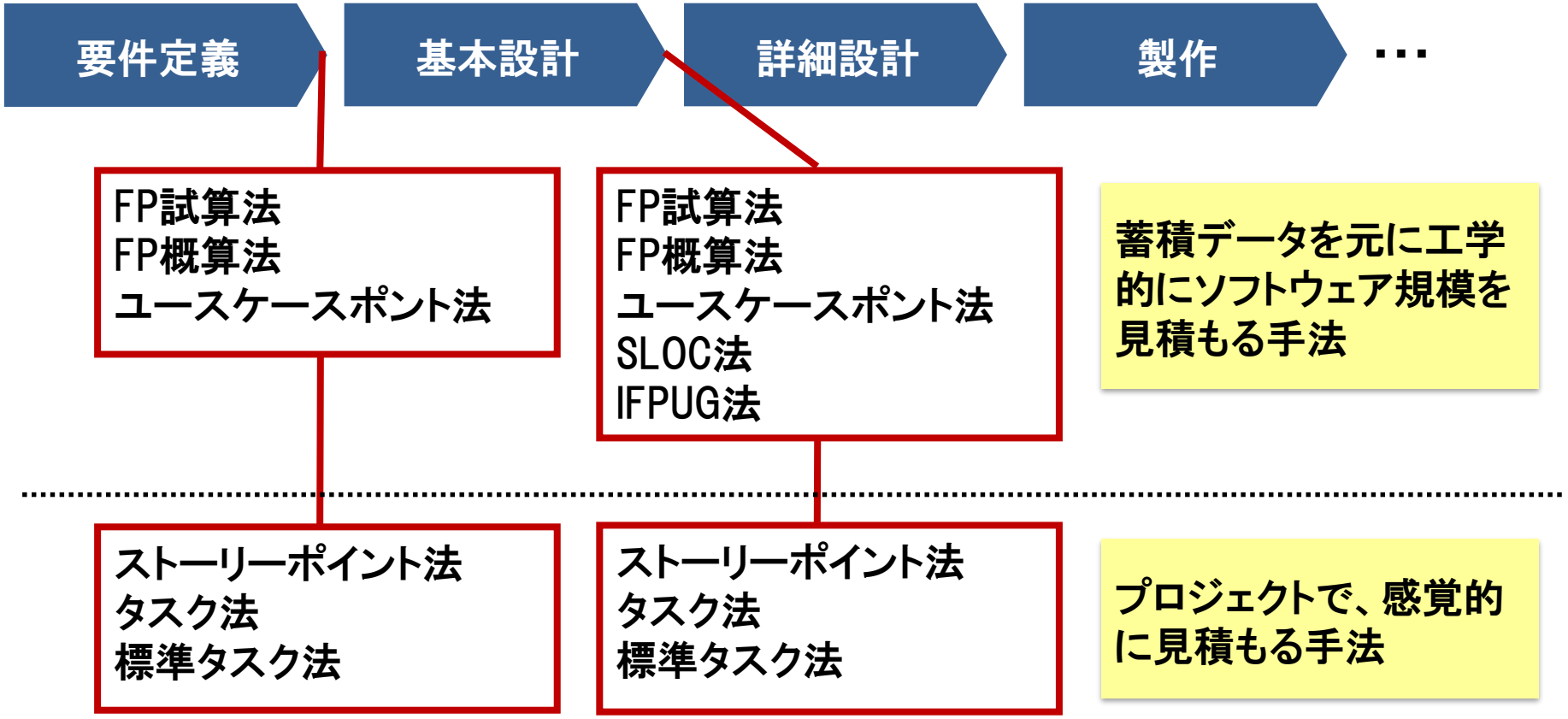
バリー・ベームの法則(1981年発表)



10-2 見積もり手法の適用可能時期

HITACHI
Inspire the Next

工程により使える見積もり手法は異なる



HITACHI
Inspire the Next

機能単位にタスクポイントを見積もる

機能	機能規模	複雑度	調整タスクポイント
AAA機能	大	複雑	3
BBB機能	中	普通	1
CCC機能	小	単純	0.4
： ：			
ZZZ機能	中	複雑	2
合計			226

変更分の見積もりに相性がよい

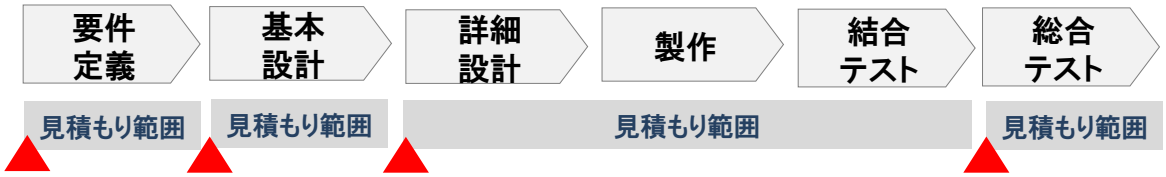
10-4 見積もり範囲とタイミング

HITACHI
Inspire the Next

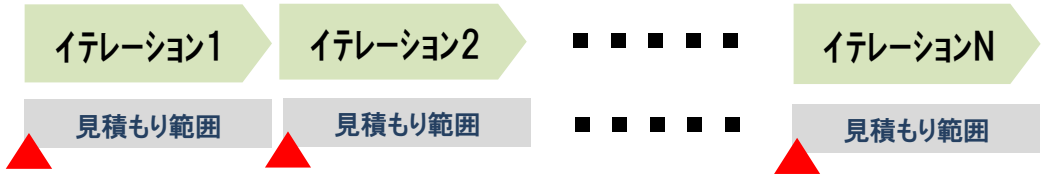
見積もりは、分割して行うほうがコストオーバーリスクが少ない

凡例 ▲ :見積もり
△ :追加や変更分の見積もり

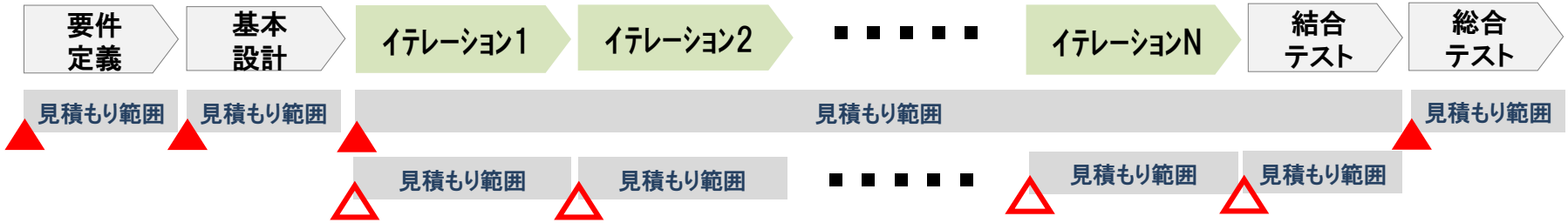
ウォーターフォール



アジャイルプロセス



ハイブリッドアジャイル



10-5 第10章のまとめ

HITACHI
Inspire the Next

① 契約のための見積もり手法を決める

② 変更分の見積もりは、タスク法を勧める

③ アジャイルの見積もり手法は契約には向かない

④ 工程やイテレーションで分割して見積もる

HITACHI
Inspire the Next

11. 契約の考え方

11-1　日本の委託開発契約

HITACHI
Inspire the Next

法律は遵守する

請負契約

契約のための見積もりが必要
業務完成義務
瑕疵担保責任
指揮命令系統

準委任契約

契約のための見積もりが必要
善管注意義務
報告義務
指揮命令系統

→ ユーザ企業からの委託先担当者へ直接指揮命令はできない

民法改正

2017年5月26日に「民法の一部を改正する法律」が成立（6月2日公布）した。「この法律は、公布の日から起算して三年を超えない範囲内において政令で定める日から施行する。…」と定められおり、2020年6月までには施行されることになる。

- 担保責任：「瑕疵」を「契約の内容に適合しないもの」に変更し、瑕疵担保責任を債務不履行責任となる。
- 請負の出来高報酬：注文者が受ける利益の割合に応じて報酬を請求できる（出来高報酬）が明文化される。
- 請負の瑕疵担保：瑕疵担保責任の請求が「目的物を引き渡してから1年以内に権利行使」から「契約に適合しないことを知ってから1年以内に通知」となる。
- 準委任：成果完成型と履行割合型が新設される。

11-2 米国のコスト契約

HITACHI
Inspire the Next

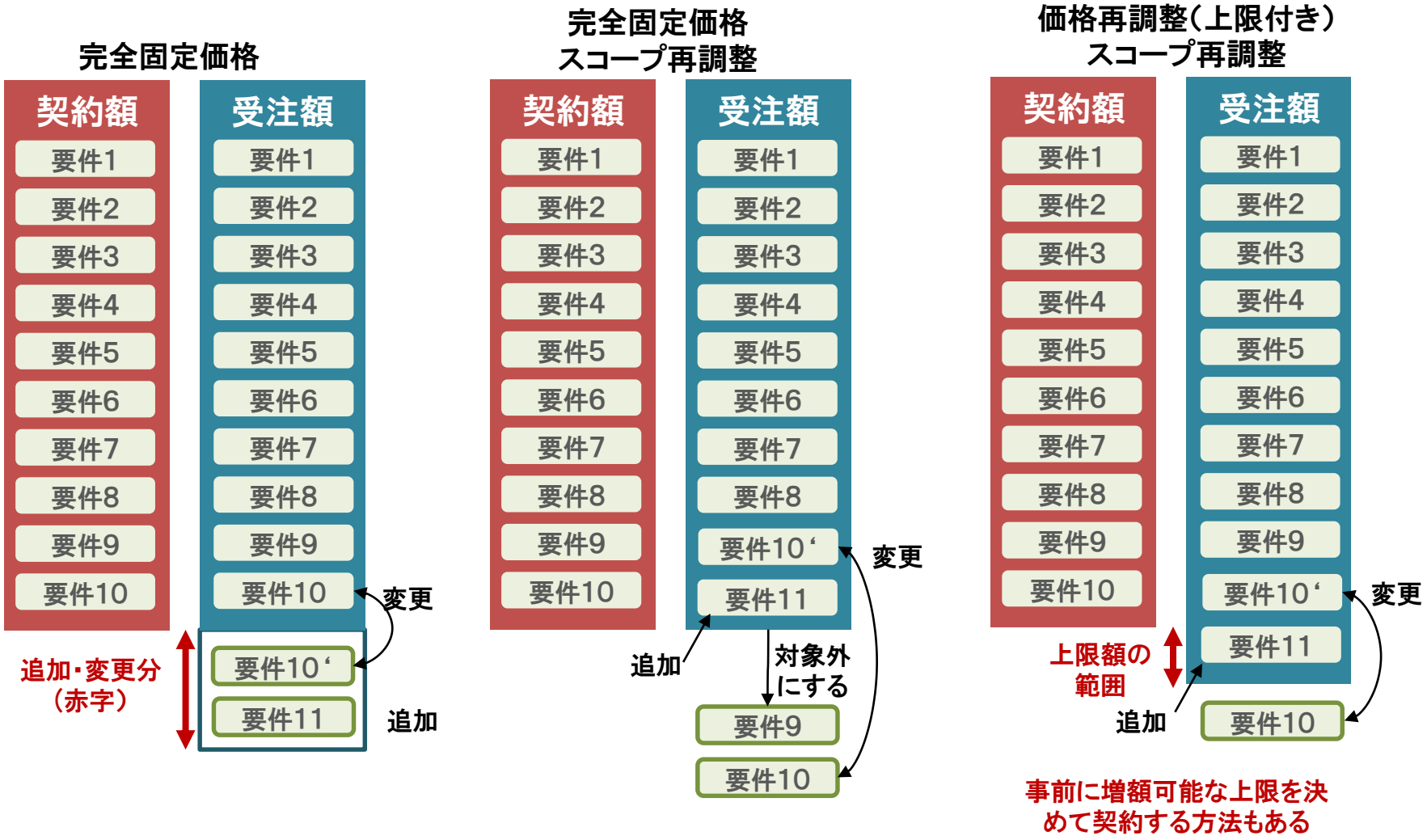
米国のコスト契約は多彩、再調整可能な契約がある

1. Fixed-Price Contracts （固定価格契約）	
1-1	Firm-fixed-price contracts （完全固定価格契約）
1-2	Fixed-price contracts with economic price adjustment （物価変動調整付固定価格契約）
1-3	Fixed-price incentive contracts （固定価格インセンティブ契約）
1-4	Fixed-price contracts with prospective price redetermination （予測価格再設定付固定価格契約）
1-5	Fixed-ceiling-price contracts with retroactive price redetermination （遡及価格再設定付固定最大価格契約）
1-6	Firm-fixed-price, level-of-effort term contracts （完全固定価格契約，期間内努力レベル契約）
2. Cost-Reimbursement Contracts （実費償還契約）	
2-1	Cost contracts （実費のみ契約）
2-2	Cost-sharing contracts （実費シェアリング契約）
2-3	Cost-plus-incentive-fee contracts （実費＋インセンティブフィー契約）
2-4	Cost-plus-award-fee contracts （実費＋表彰フィー契約）
2-5	Cost-plus-fixed-fee contracts （実費＋固定フィー契約）
3. Incentive Contracts （インセンティブ契約）	
3-1	Fixed-price incentive contracts （固定価格インセンティブ） ・Fixed-price incentive (firm target(固定目標)) contracts ・Fixed-price incentive (successive targets(連続目標)) contracts
3-2	Fixed-price contracts with award fees （表彰フィー付固定価格契約）
3-3	Cost-reimbursement incentive contracts （実費償還インセンティブ契約） ・Cost-plus-incentive-fee contracts （2-3と同一） ・Cost-plus-award-fee contracts （2-4と同一）
4. Indefinite-Delivery Contracts （未確定調達契約）	
4-1	Definite-quantity contracts （数量確定契約:調達時期未定）
4-2	Requirements contracts （要求契約:数量・調達時期とも未定、1者(社)から調達）
4-3	Indefinite-quantity contracts （数量未確定契約:数量・調達時期とも未定、複数者(社)から調達）
5. Time-and-materials ／ Labor-hour／Letter contracts （その他契約）	
5-1	Time-and-materials contracts （タイム・アンド・マテリアル契約）
5-2	Labor-hour contracts （労働時間契約）
5-3	Letter contracts （書簡契約）

11-3 再調整可能な契約

HITACHI
Inspire the Next

なければならない機能、もたらす価値の大きな機能の優先度が高い

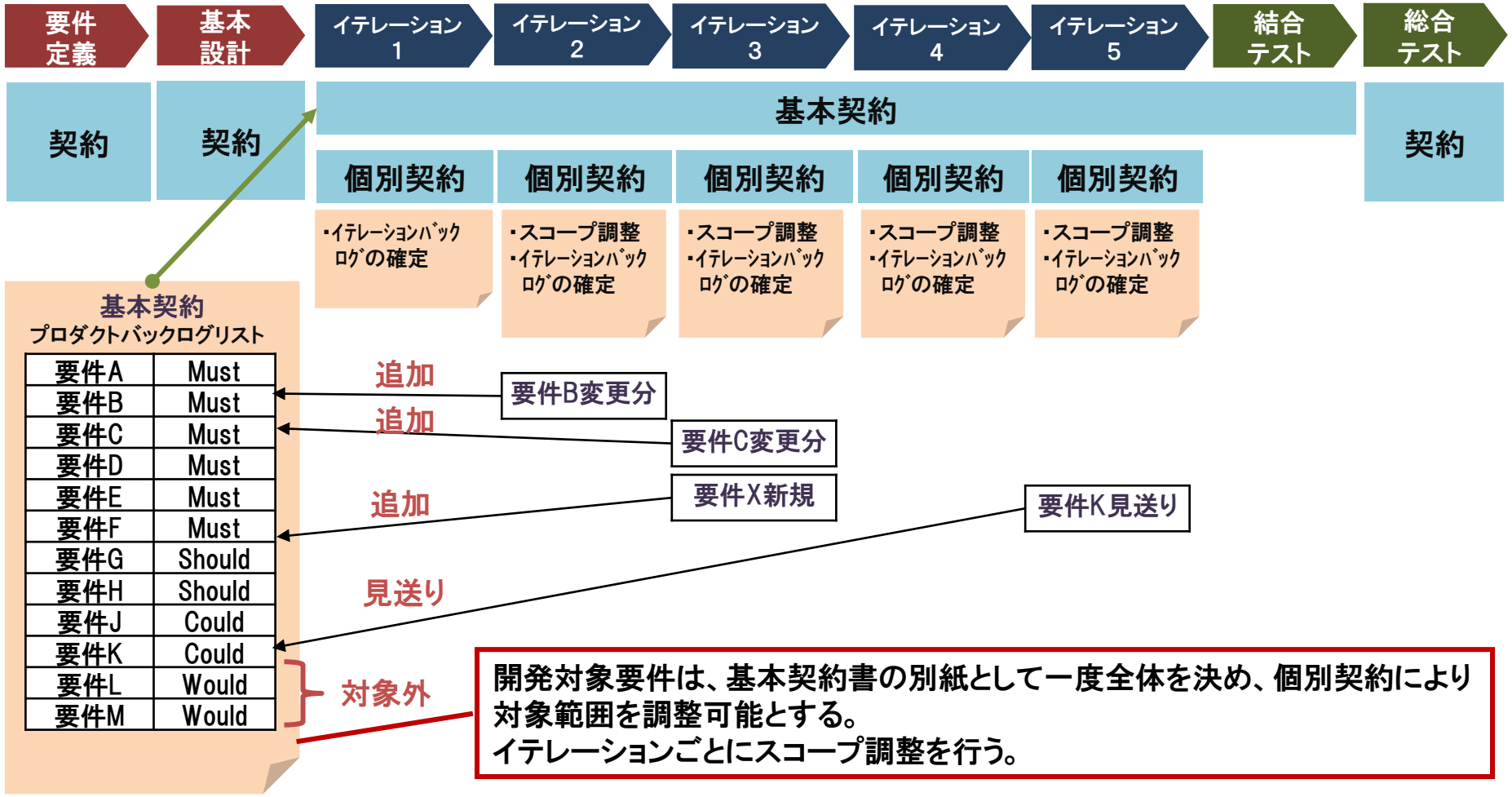


11-4 スコープ調整（基本契約と個別契約）

HITACHI
Inspire the Next

追加や変更、対象外分は個別契約でスコープ調整する（図は例）

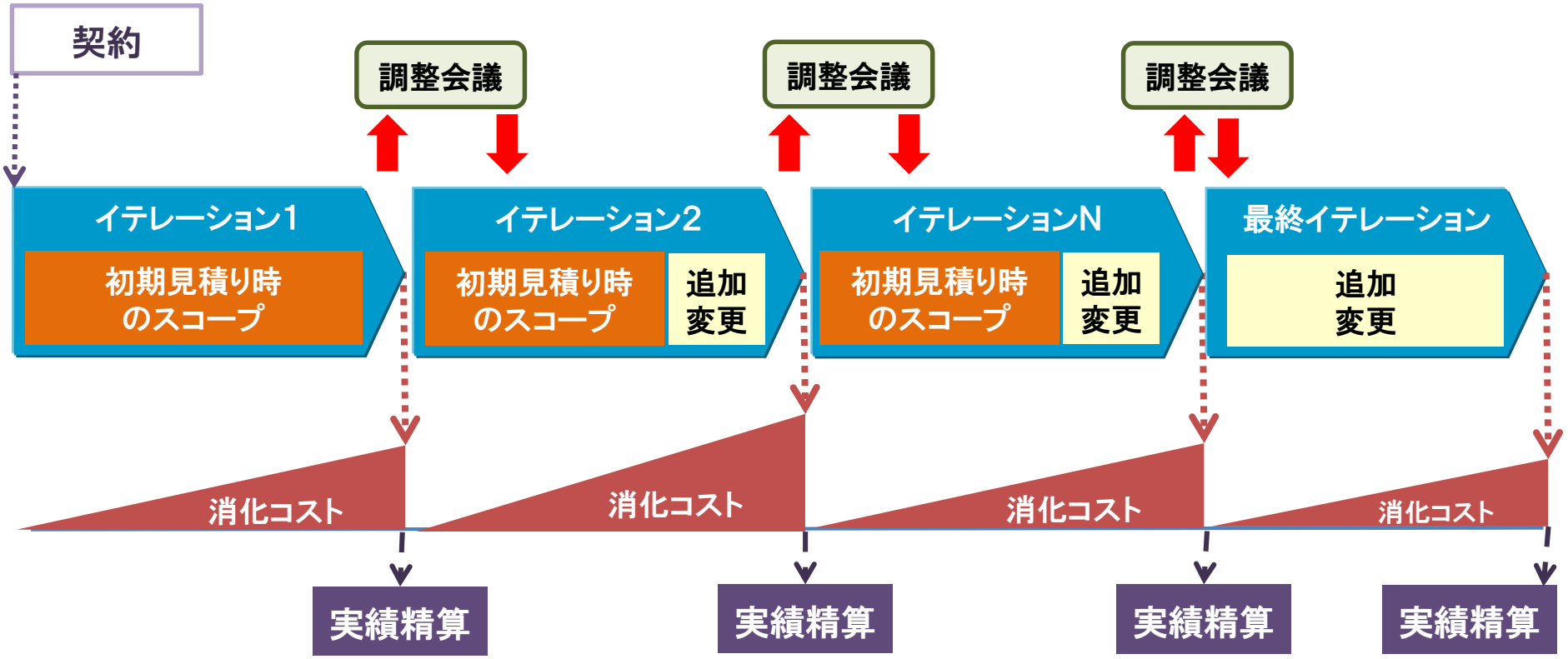
ハイブリッドアジャイル



11-5 価格再調整契約(実績コスト精算方式の場合)

HITACHI
Inspire the Next

イテレーションや月次などの短い期間ごとに、実績コストで精算する

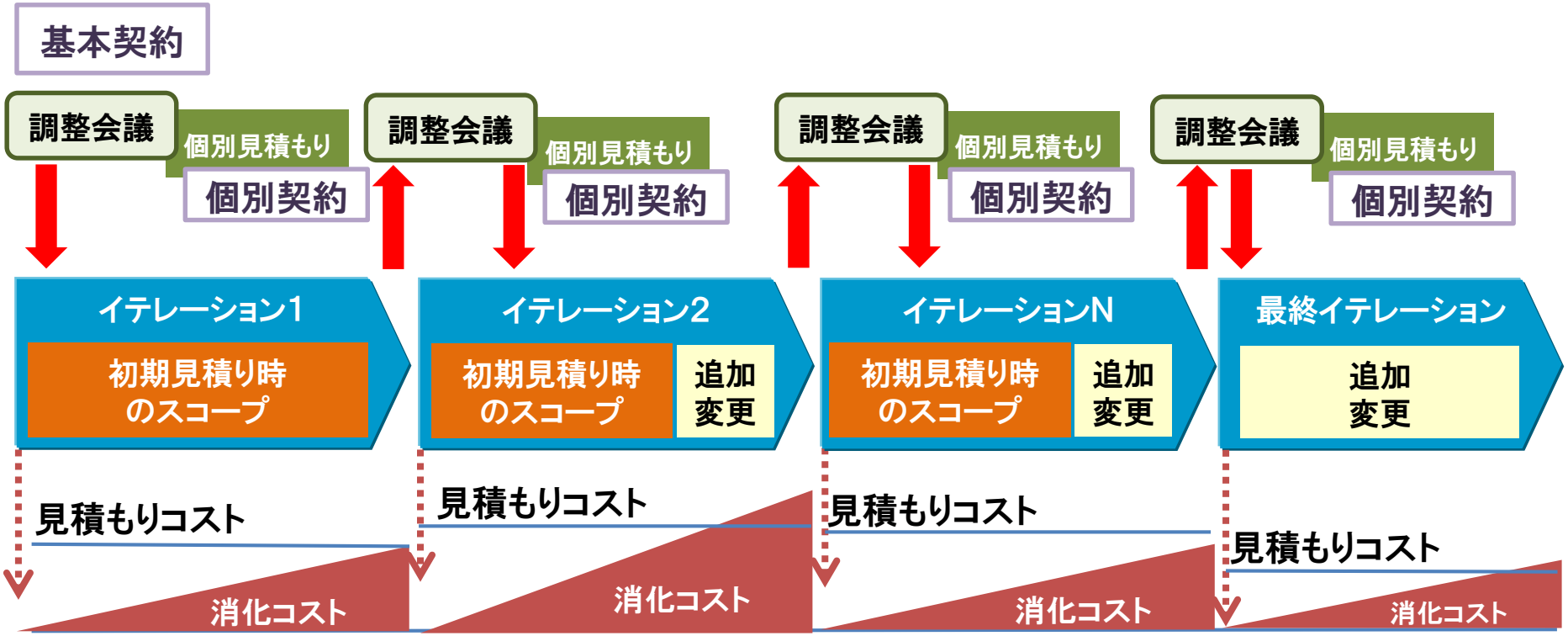


契約タイプ：準委任契約
ユーザー：予算超過のリスクあり
ベンダ：損失なし、粗利期待少ない

11-6 価格再調整契約(見積もりコストの場合)

HITACHI
Inspire the Next

イテレーションや月次などの短い期間で、見積もりコストで契約する

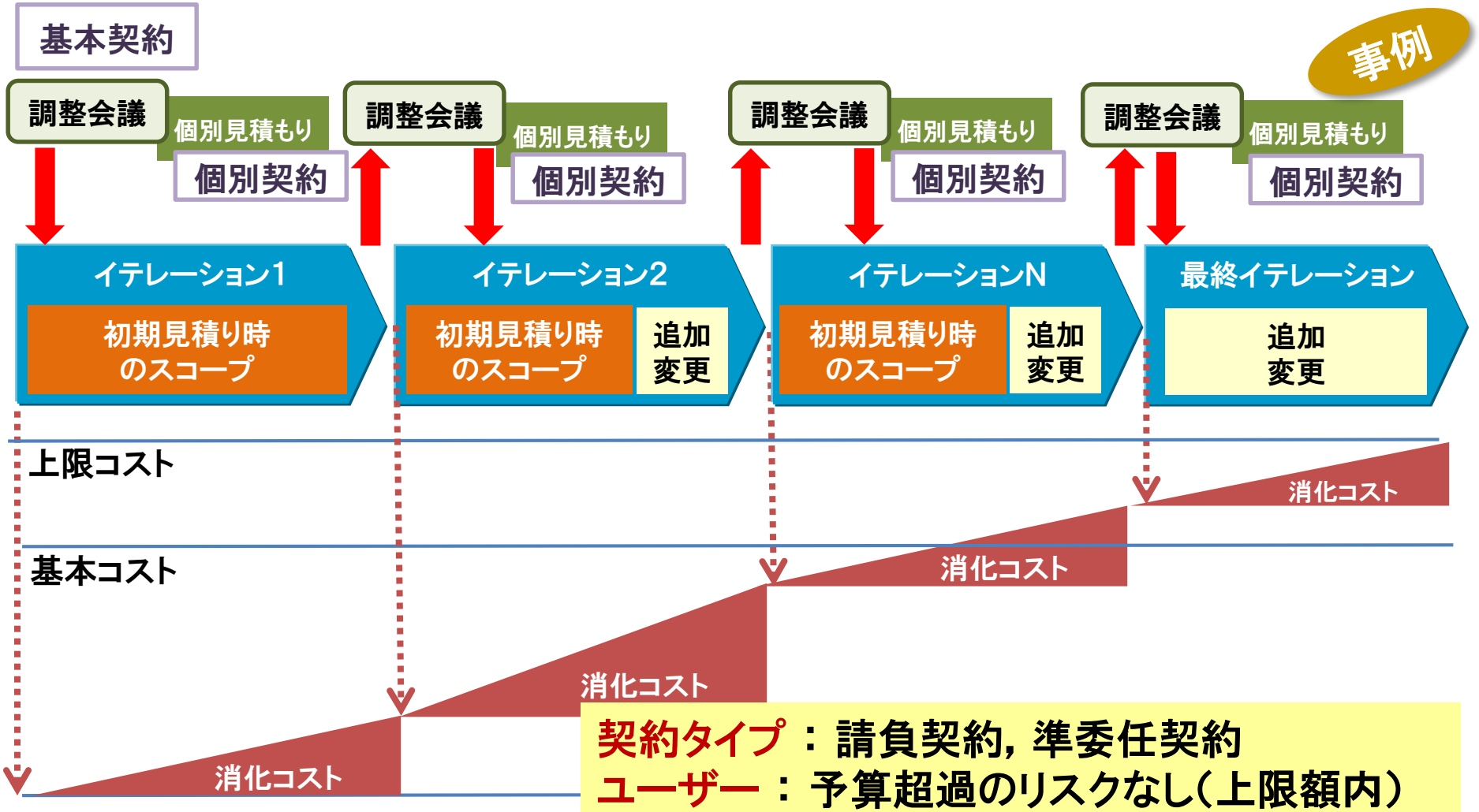


契約タイプ：請負契約，準委任契約
ユーザー：予算超過のリスクなし
ベンダ：損失リスクあり、粗利期待あり

11-7 価格再調整契約(目標コストの場合)

HITACHI
Inspire the Next

上限額までの範囲で追加や変更を調整する



契約タイプ：請負契約，準委任契約
ユーザー：予算超過のリスクなし(上限額内)
ベンダ：損失リスクなし、粗利期待あり

11-8 第11章のまとめ

HITACHI
Inspire the Next

① 委託開発では、再調整可能な契約にする

② 優先度を付ける（見送るものもある）

③ イテレーションごとにスコープの再調整を行う

④ 計画にはコストとスケジュールの「ゆとり」を設けておく

全体まとめ

HITACHI
Inspire the Next

- アジャイルプロセスの適用を目的にしない
- ビジネス価値の高い機能から早くリリースする
- プロジェクトに適した開発プロセスを適用すること
- アジャイルプロセスのノウハウは、プロセス改善の良き手本になる
- アジャイルプロセスは、進捗管理や品質管理が疎かになるわけではない
- 管理は、チケット管理や常時結合などの適用で厳しくなる
- プロジェクトのリスク防止には、優れた効果を発揮できる
- アジャイルプロセスは、進捗管理、コスト管理、品質管理、変更管理の方法を確定しておくことが重要
- 委託元と委託先で開発方法や変更要望に対する扱いを合意することと協力して進めることが重要

書籍紹介

単行本発行日：2013/10/14
発行所：株式会社リックテレコム
監修：長瀬嘉秀
著者：英 繁雄，奈加健次，平岡嗣晃，前川祐介
協力：関西電力株式会社
価格：単行本　￥2，808
　　Kindle版　￥2，592

単行本発行日：2017/10/23
発行：日経BP社
著者：英 繁雄
価格：単行本　￥2，592
　　Kindle版　￥2，400

HITACHI
Inspire the Next

ご清聴ありがとうございました

END

HITACHI

Inspire the Next