

ソフトウェア品質シンポジウム2025

制約のある大規模 ソフトウェア開発 における、 アジャイル型開発 導入の取り組み

○ 高橋 拳矢 (E-mai:kenya.takahashi.ja@ds.mhi.com)
武島 正典 (E-mai:masanori.takeshima.t8@ds.mhi.com)

2025/09/26

三菱重工業株式会社





- 1. はじめに**
- 2. 当社のソフトウェア開発状況**
- 3. 従来の開発について**
- 4. 開発プロセスの制約/検討**
- 5. 開発プロセスの構築**
- 6. 品質保証部門の役割**
- 7. 開発プロセスの効果**



本発表では、さまざまな制約の中でアジャイル型開発を導入した際の取り組みと成果について説明する。

制約下での導入という特性から、一般的なアジャイル型開発とは異なる視点も含まれているが、大規模開発における実践例やその有効性を紹介する。これにより、アジャイル型開発の導入をためらっている方々に新たな視点を提供することを目指す。



出典：<https://agilemanifesto.org/iso/en/manifesto.html>

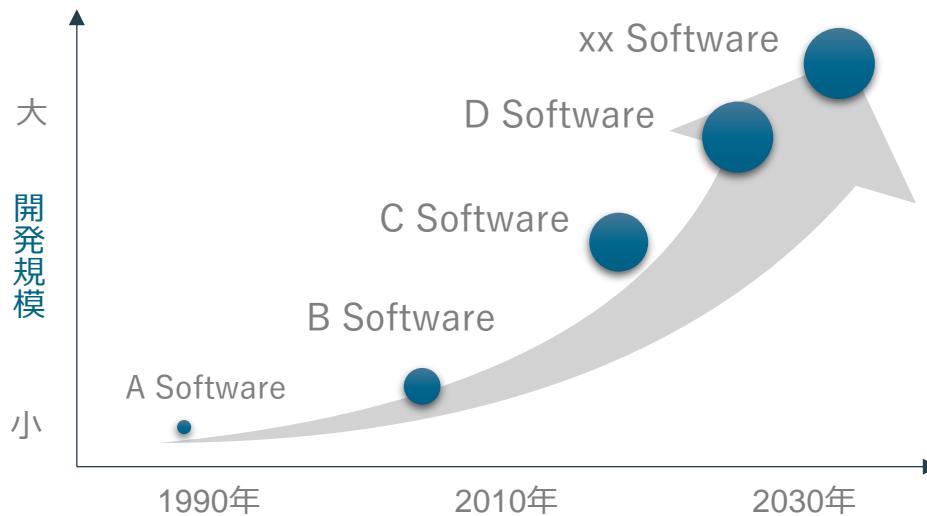
2. 当社のソフトウェア開発状況



航空機事業環境

- 航空機に搭載する表示/通信/飛行制御等の組込みソフトウェア
- 航空機を模擬したシミュレータソフトウェア など

当社は安全性/信頼性の高いソフトウェアを開発しているが、
その重要性と複雑さは年々増しており、開発規模は拡大している。



3. 従来の開発について



従来開発の悩み

ソフトウェア(SW)の大規模化

開発関係者/関係部門の拡大

SW規模/影響範囲の拡大

認識齟齬起因のバグ増加

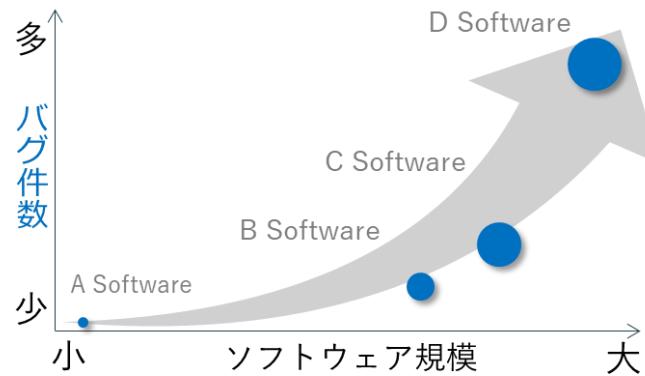
SW修正時間の長期化

「大規模ソフトウェア開発」
⇒ 手戻りが増加し、リリーススケジュール/納期に影響

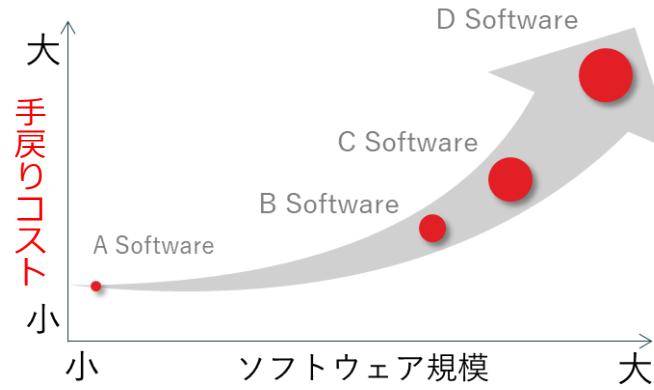
3. 従来の開発について



従来開発の悩み



認識齟齬の抑制



迅速なソフトウェア修正

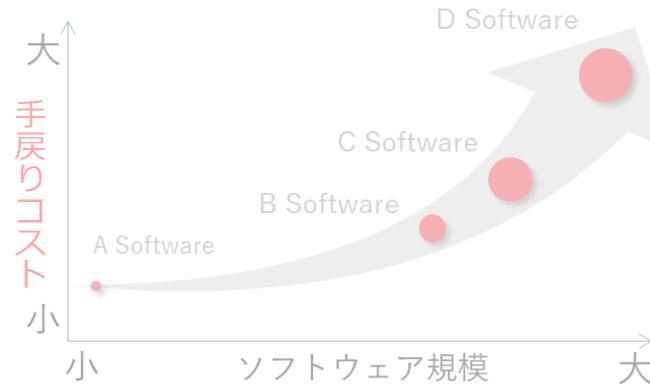
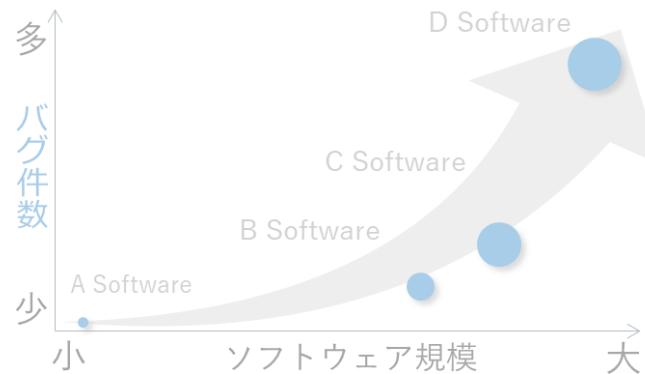


実現するための開発プロセスを構築する

3. 従来の開発について



従来開発の悩み



認識齟齬の抑制

迅速なソフトウェア修正

チームの対話の活性化

迅速な対応力

アジャイル型開発の導入を検討



開発プロセスの制約と課題

制約

①顧客との契約内容により、システム開発の大枠は
ウォーターフォール型が要求されている
(ソフトウェア開発工程内のプロセス検討が必要)

制約

②安全性・信頼性が求められる製品である
(リリースに向けてソフトウェア全域のテストが必須)

制約

③システム要求は確定済
(細部仕様の変更・追加等に対する柔軟な対応が必要)

課題

④仕様検討は社内で、
製作はパートナー会社という
分業制が多い
(ワンチームの開発体制が
とりづらい)

4. 開発プロセスの制約/検討



開発プロセスの検討

①アジャイル型開発はシステム要件定義の後に適用
顧客を除いた関係者にてスクラムを組織

制約

②安全性・信頼性が求められる製品である
(リリースに向けてソフトウェア全域のテストが必須)

制約

③システム要求は確定済
(細部仕様の変更・追加等に対する柔軟な対応が必要)

課題

④仕様検討は社内で、
製作はパートナー会社という
分業制が多い
(ワンチームの開発体制が
とりづらい)



開発プロセスの検討

①アジャイル型開発はシステム要件定義の後に適用
顧客を除いた関係者にてスクラムを組織

②統合(結合)テスト以降はウォーターフォール型開発

制約

③システム要求は確定済
(細部仕様の変更・追加等に
対する柔軟な対応が必要)

課題

④仕様検討は社内で、
製作はパートナー会社という
分業制が多い
(ワンチームの開発体制が
とりづらい)



開発プロセスの検討

①アジャイル型開発はシステム要件定義の後に適用
顧客を除いた関係者にてスクラムを組織

②統合(結合)テスト以降はウォーターフォール型開発

③実現すべき機能を
段階的に積み上げていく
製作方式を適用

課題

④仕様検討は社内で、
製作はパートナー会社という
分業制が多い
(ワンチームの開発体制が
とりづらい)



開発プロセスの検討

①アジャイル型開発はシステム要件定義の後に適用
顧客を除いた関係者にてスクラムを組織

②統合(結合)テスト以降はウォーターフォール型開発

③実現すべき機能を
段階的に積み上げていく
製作方式を適用

④社内/パートナー会社の
関係者全員 参加型の
レビュー会とする



開発プロセスの検討

①アジャイル型開発はシステム要件定義の後に適用
顧客を除いた関係者にてスクラムを組織

②統合(結合)テスト以降はウォーターフォール型開発

③実現すべき機能を
段階的に積み上げていく
製作方式を適用

④社内/パートナー会社の
関係者全員 参加型の
レビュー会とする

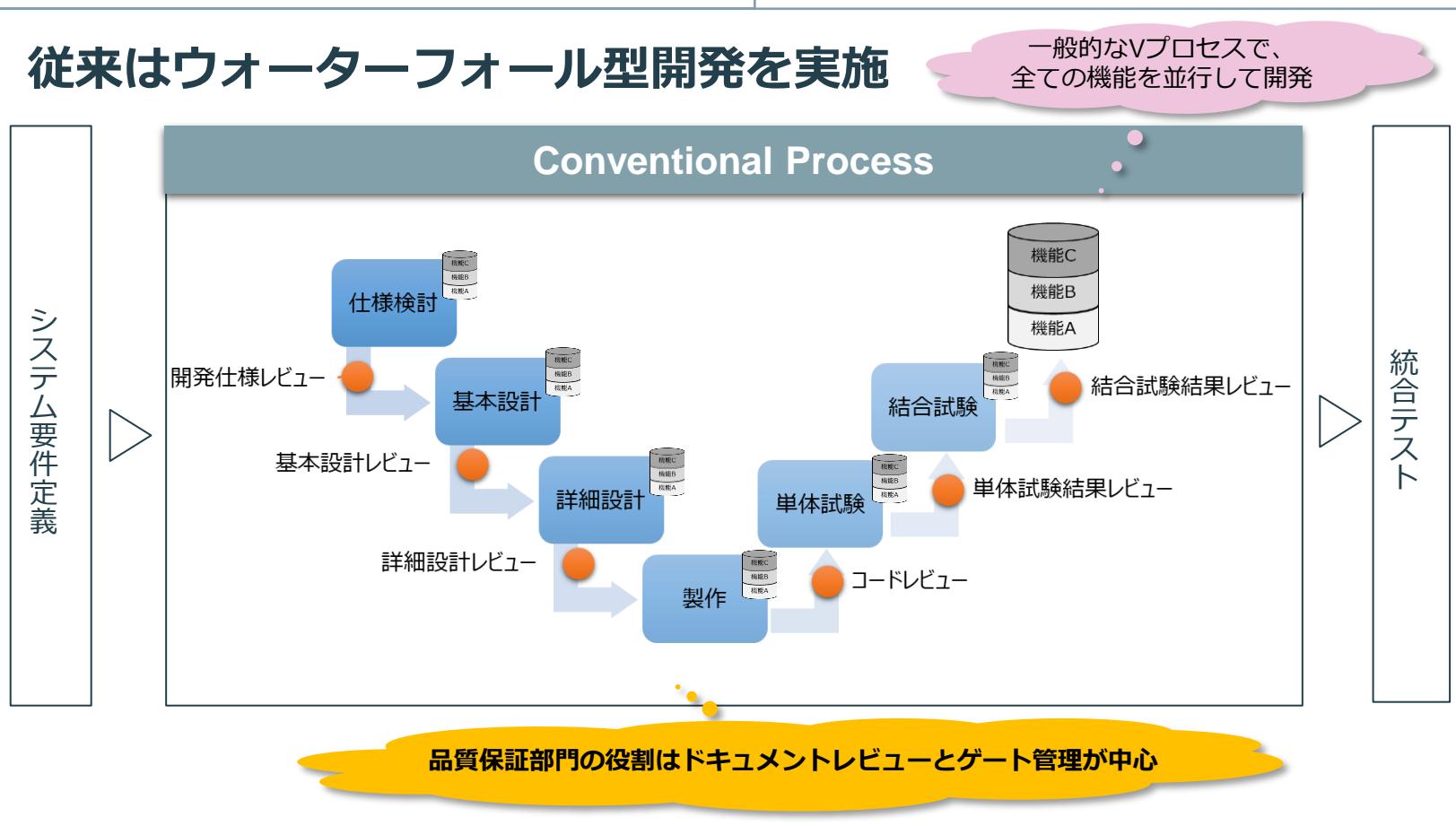
5. 開発プロセスの構築



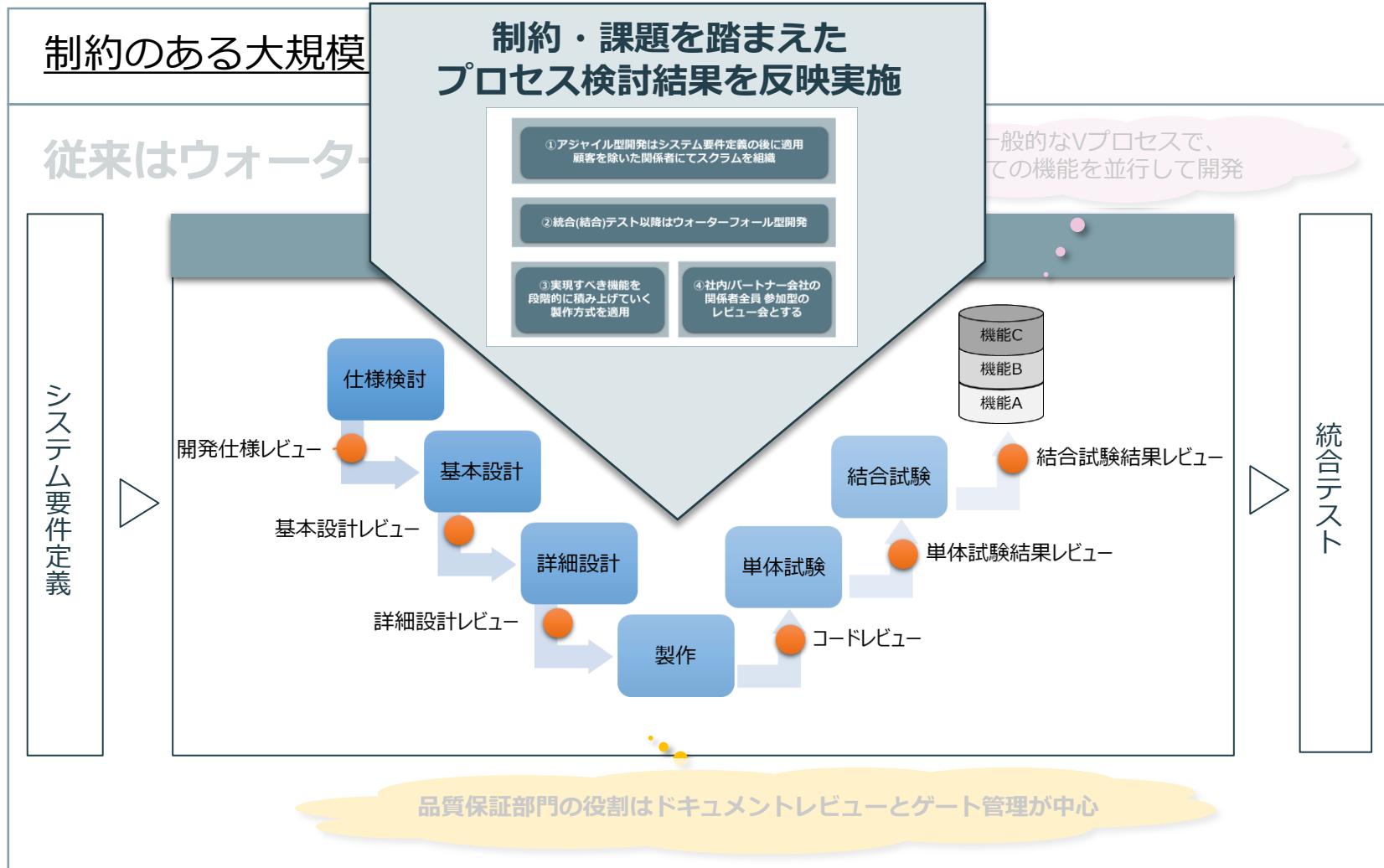
制約のある大規模ソフトウェア開発

従来はウォーターフォール型開発を実施

一般的なVプロセスで、全ての機能を並行して開発



5. 開発プロセスの構築



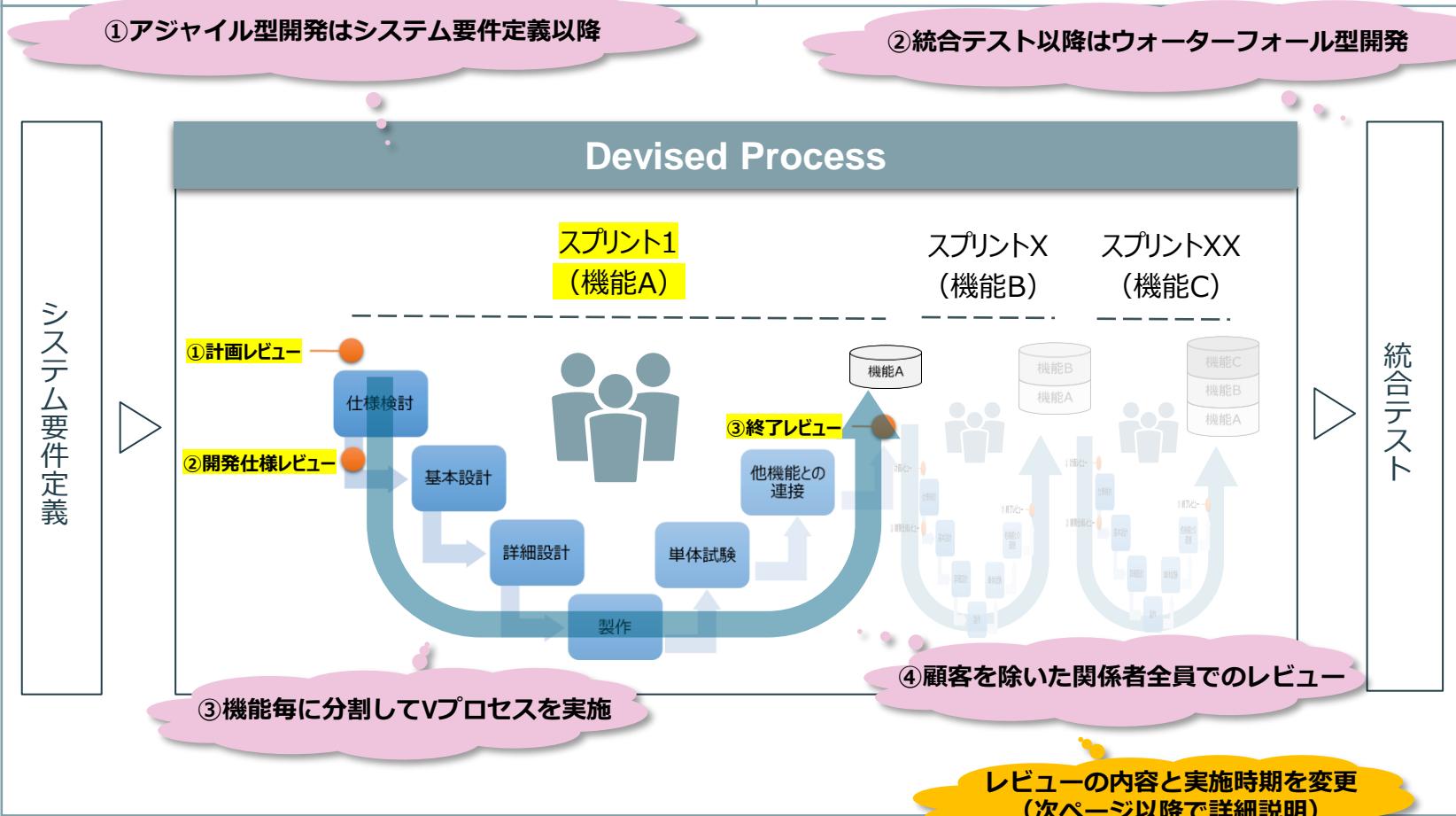
5. 開発プロセスの構築



制約のある大規模ソフトウェア開発

①アジャイル型開発はシステム要件定義以降

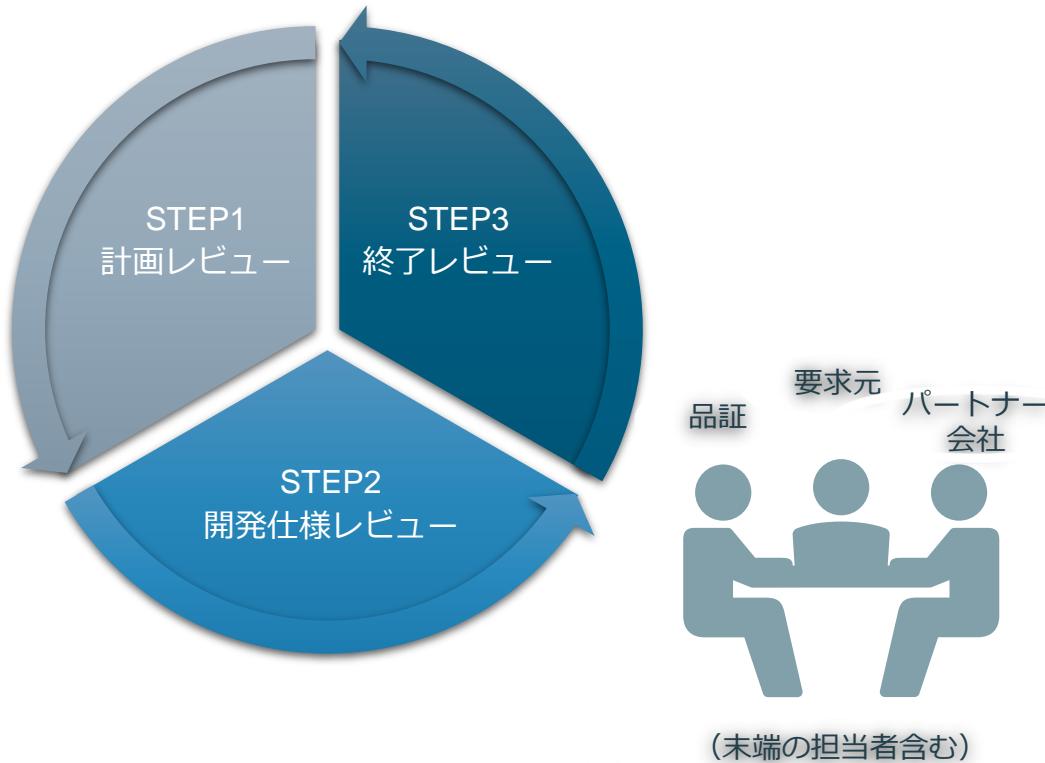
②統合テスト以降はウォーターフォール型開発



5. 開発プロセスの構築



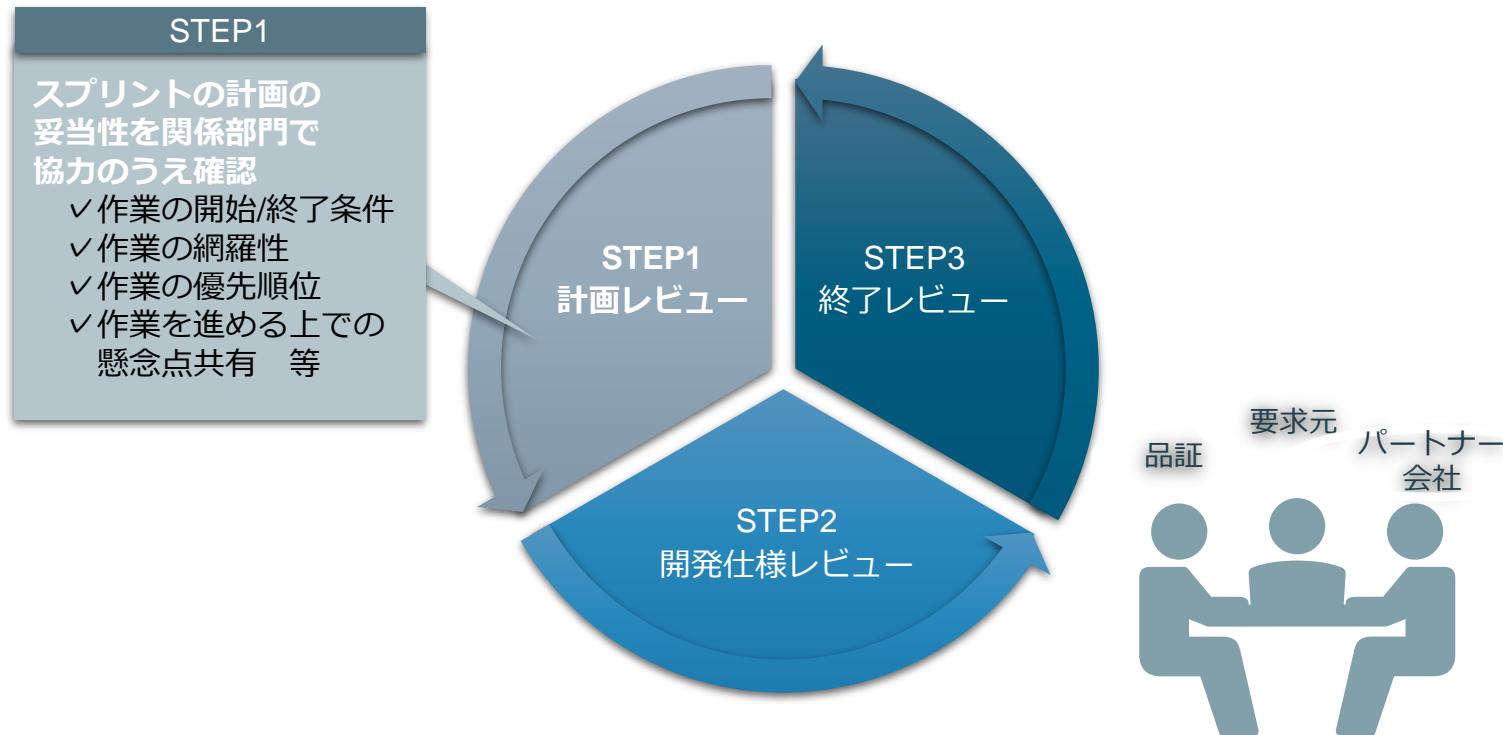
制約のある大規模ソフトウェア開発



5. 開発プロセスの構築



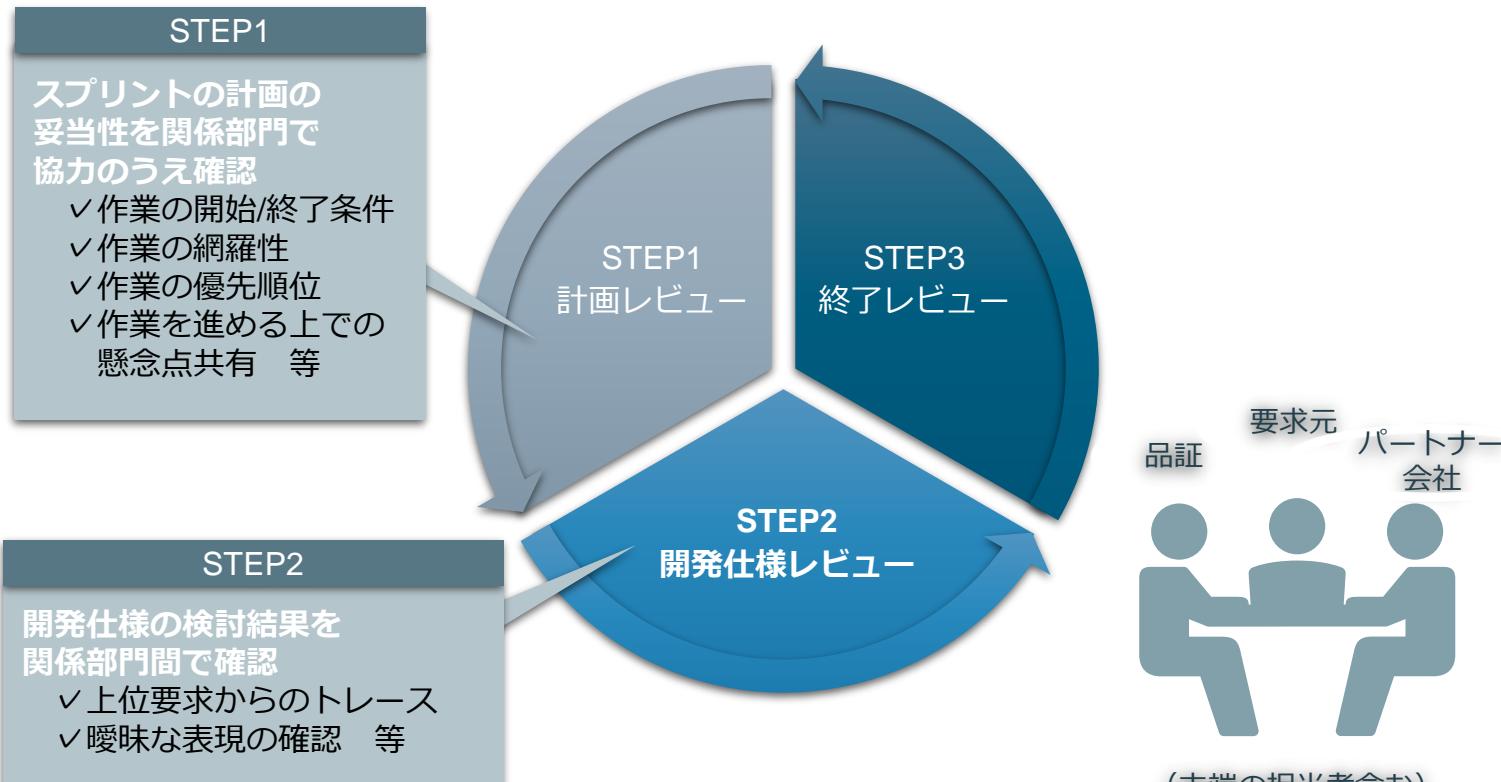
制約のある大規模ソフトウェア開発



5. 開発プロセスの構築



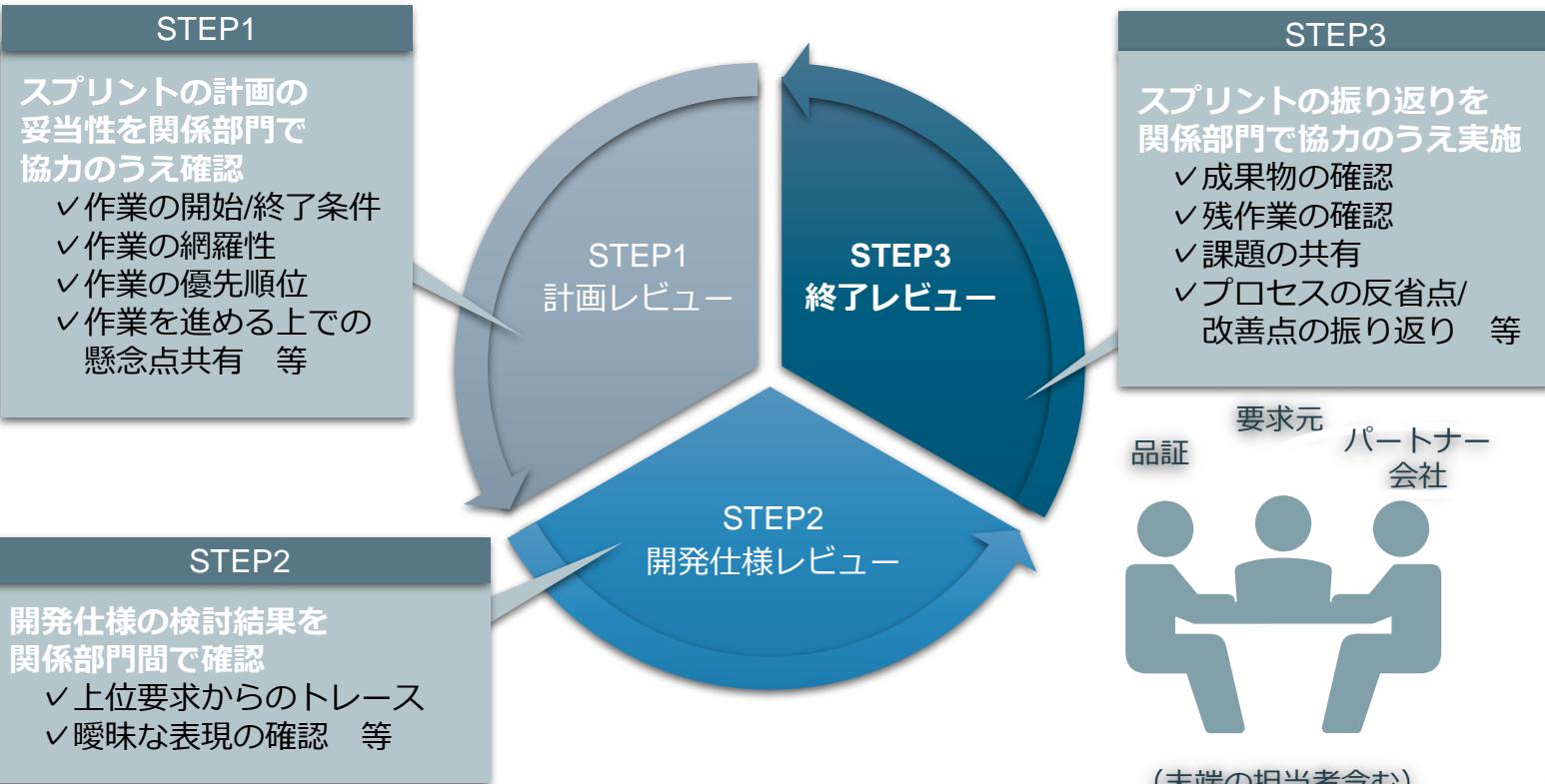
制約のある大規模ソフトウェア開発



5. 開発プロセスの構築



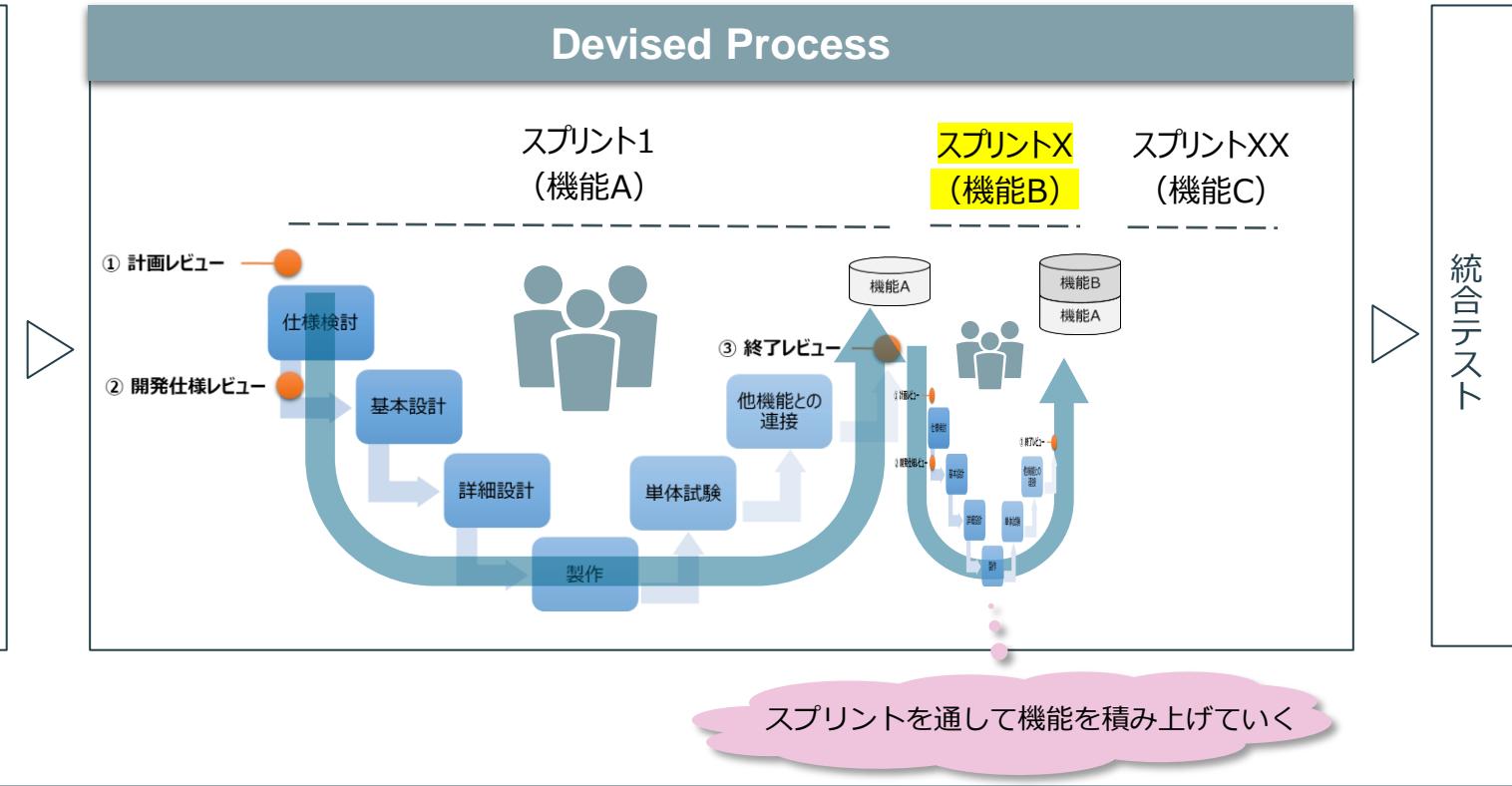
制約のある大規模ソフトウェア開発



5. 開発プロセスの構築



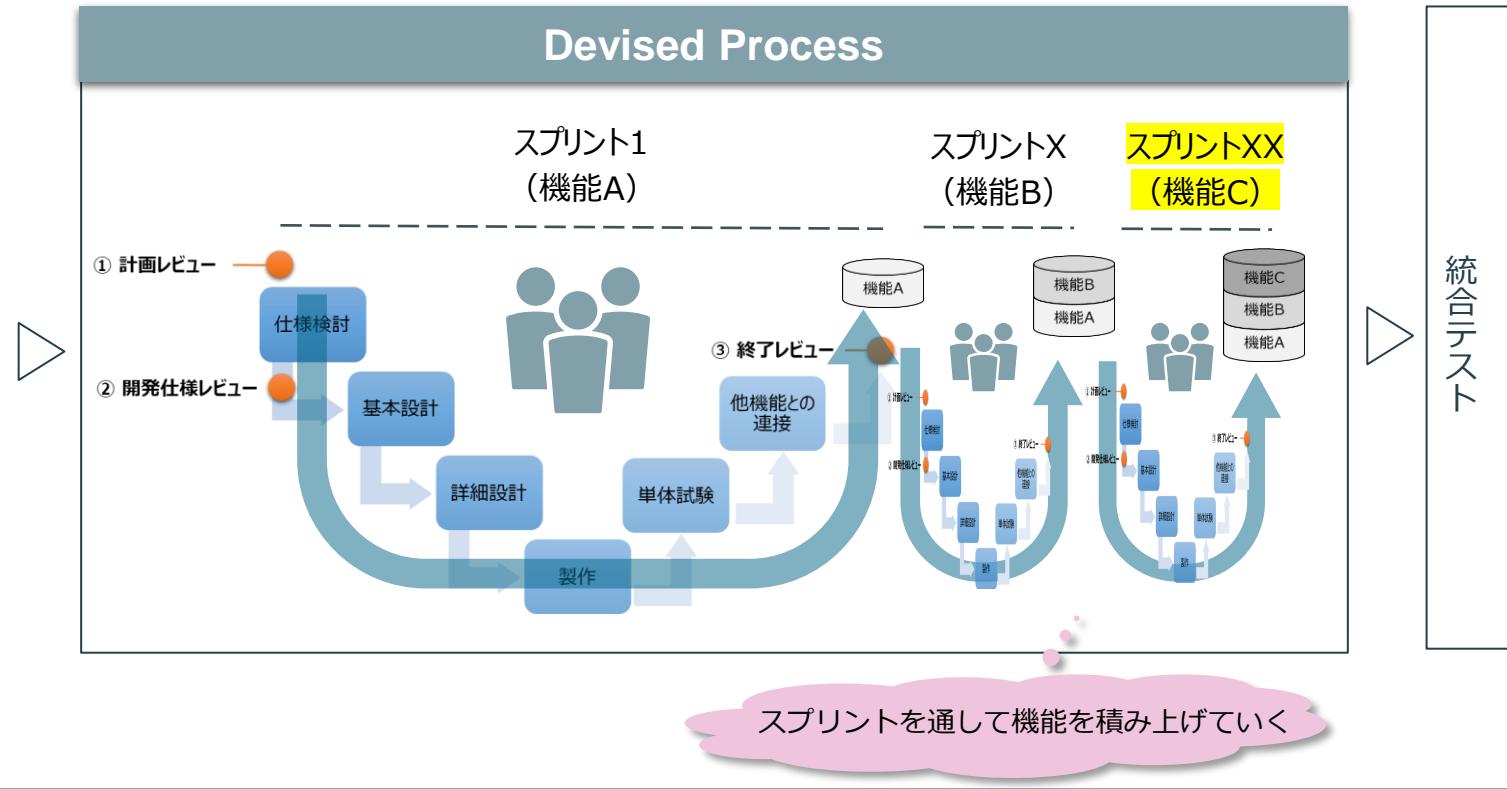
制約のある大規模ソフトウェア開発



5. 開発プロセスの構築



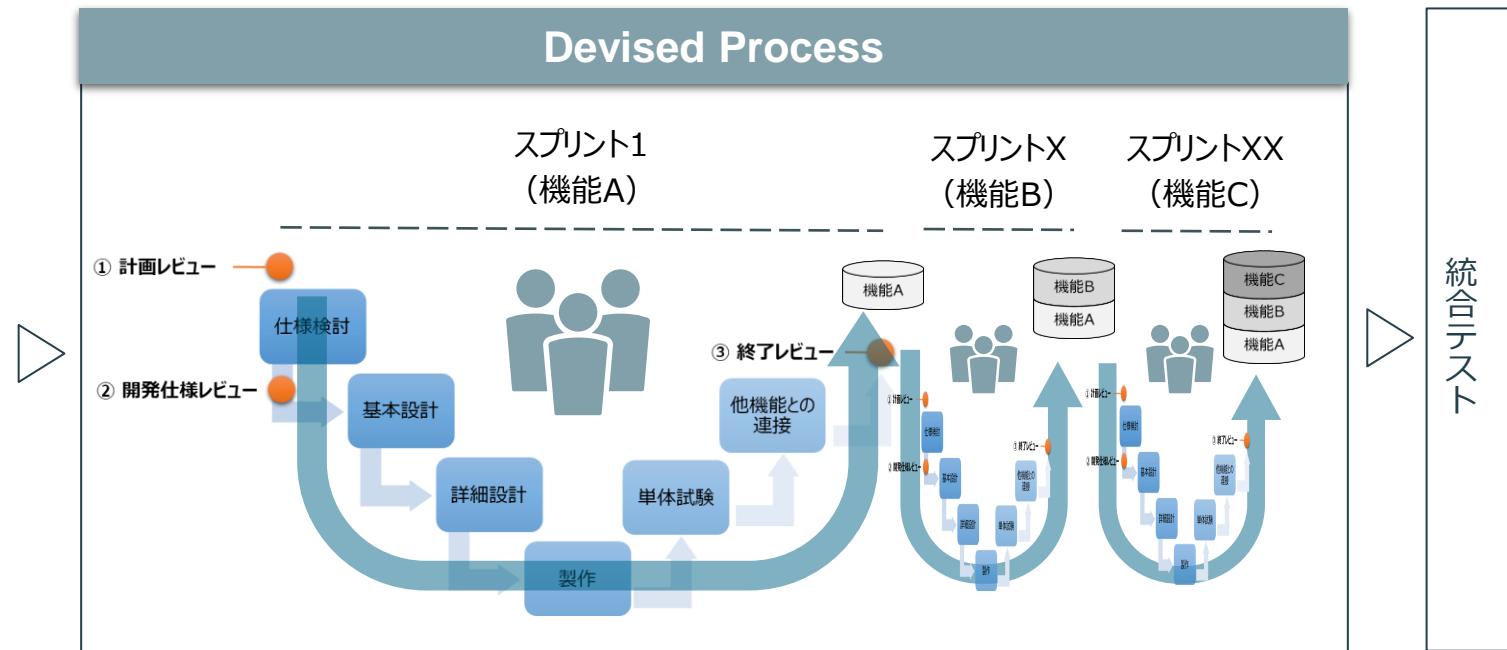
制約のある大規模ソフトウェア開発



5. 開発プロセスの構築



制約のある大規模ソフトウェア開発

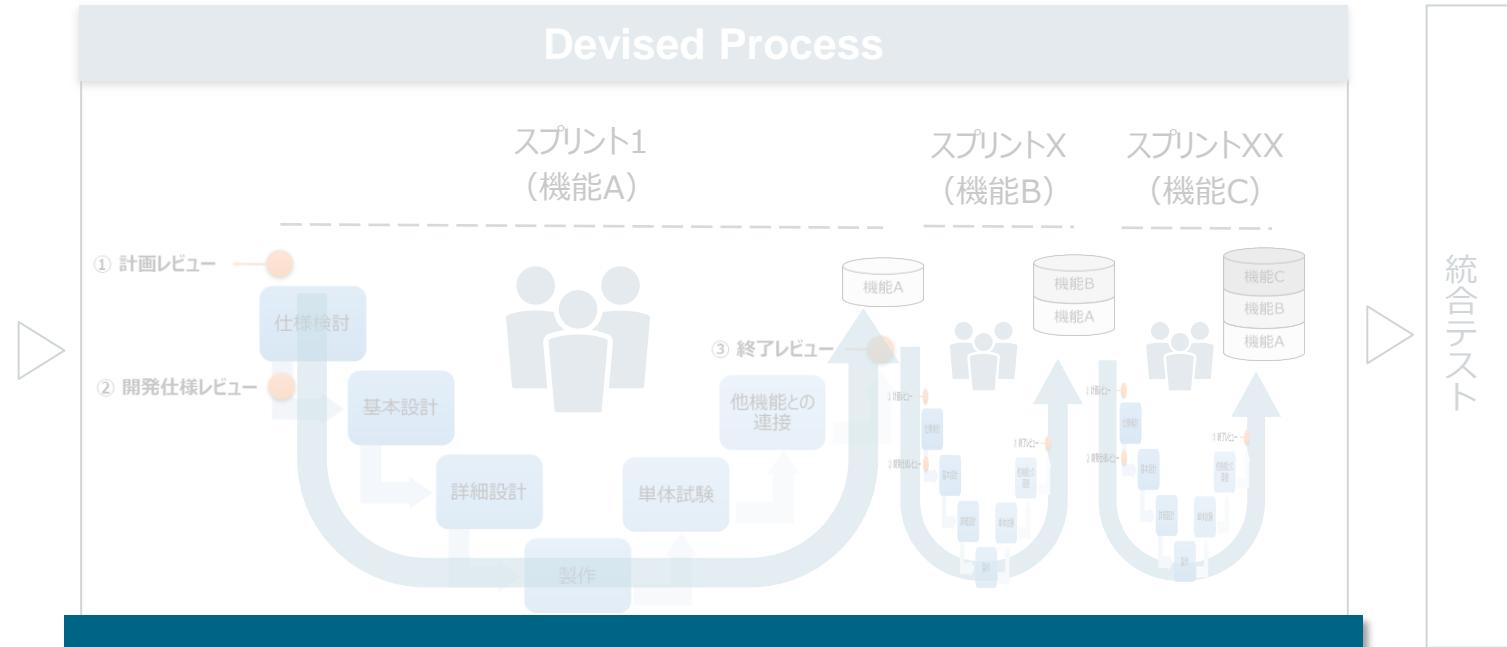


以上のようなプロセスを構築のうえ運用

5. 開発プロセスの構築



制約のある大規模ソフトウェア開発



レビュー内容に 進捗/課題管理が追加されたことから、
レビューにおける品証部門の役割も変化



品質保証部門 ▶ シープドッグ(牧羊犬)の役割



出典：PowerPointストック画像



品質保証部門 ▶ シープドッグ(牧羊犬)の役割

群れを守るべく行動し、その務めをかき乱されることもない
チームの健康と幸せこそが最高の優先順位

妨害は様々な発生源から到来

- ☞ 要求元の判断のみで作業の優先順位が決定
- ☞ 他ソフトウェアの開発進捗に伴う影響
- ☞ 作業中に判明した新たな事象



品質保証部門は以下のようにチームを問題解決に導く

直面している課題/問題 等について議論の要点をインプットし、
作業の優先順位をつけ、取り組むべきことに専念させ、
紛争(問題)を調停する。

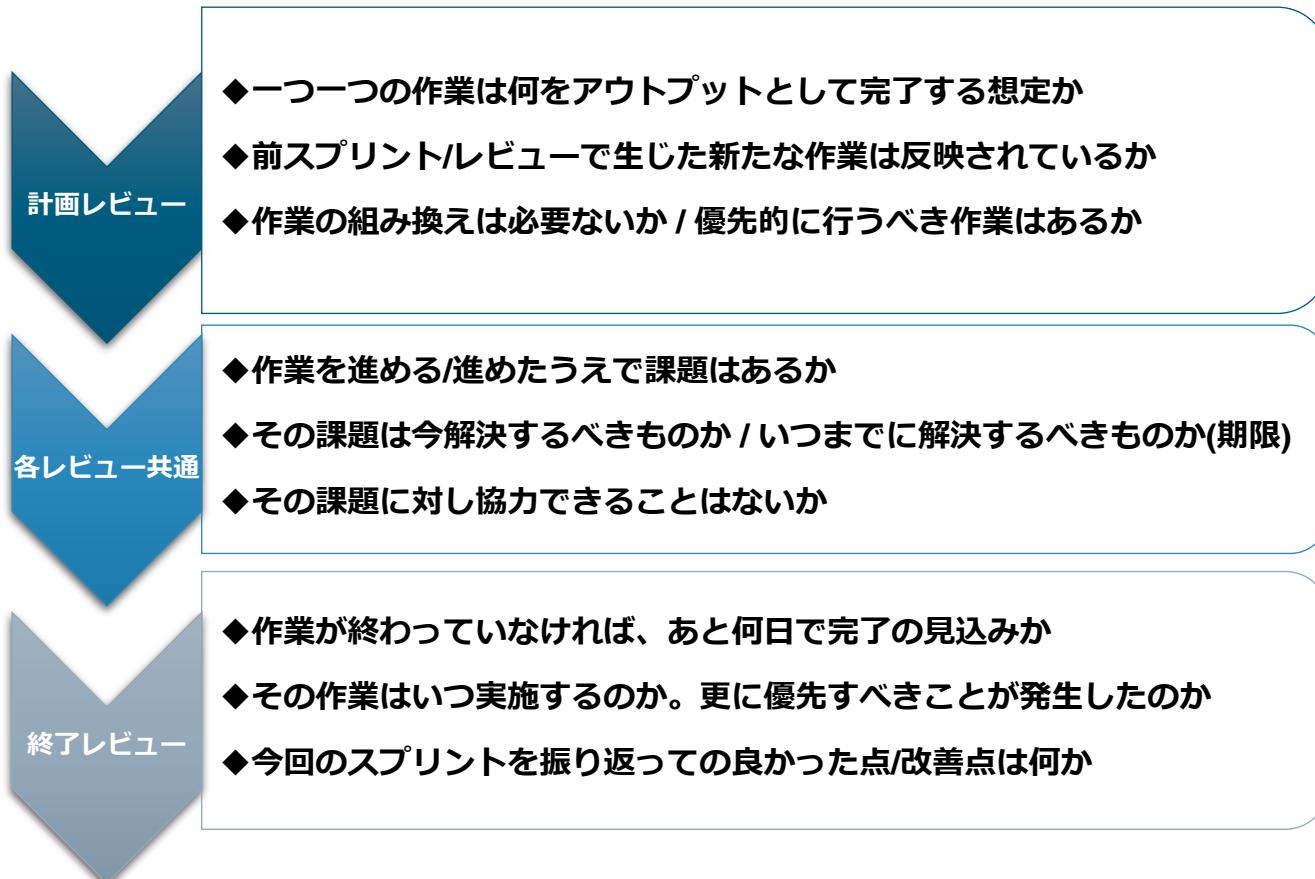


出典：PowerPointストック画像



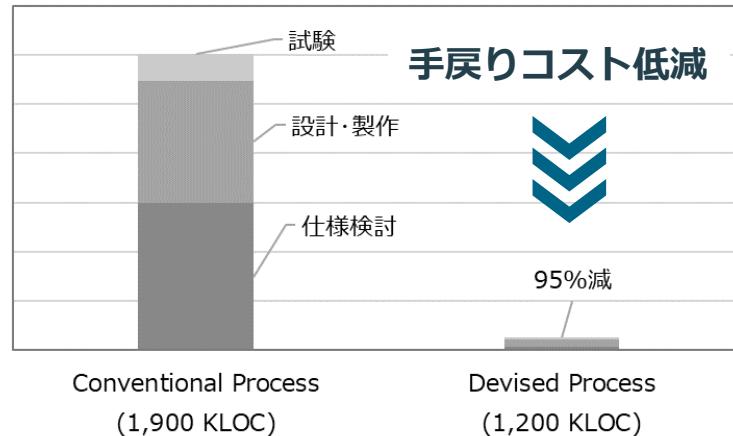
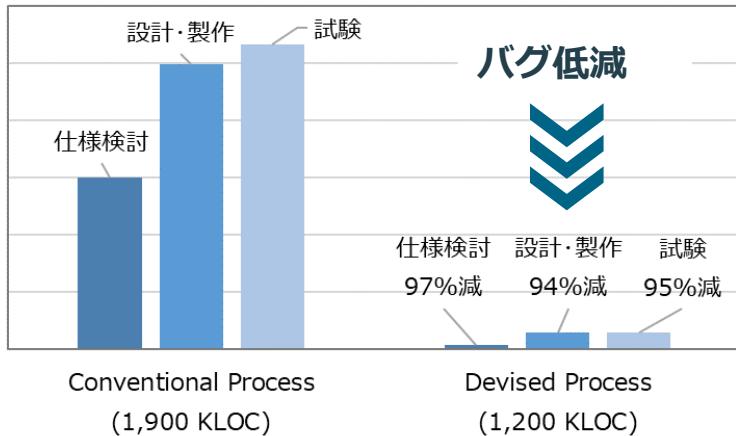
品質保証部門 ▶ シープドッグ(牧羊犬)の役割

(例) スプリントでのシープドッグ(=品質保証部門)としての確認観点





定量的な効果：従来開発との比較



バグの発生件数と原因内訳

手戻りに要したコスト

▶ 認識齟齬起因のバグ低減＆手戻りコスト低減 達成



顕著な効果：One Team 開発体制による迅速な開発実現

誰もがフラットに物事が言える/聞ける環境の提供

要求部門／品質保証部門に加えてパートナー会社(末端の作業担当含)も参加し、開発の実態に即した対話の活性化を実現

頻繁なプロセス振り返りによる、運用と並行したタイムリーな改善反映

関係者全員が当事者意識を持って情報共有・作業を促進





定性的な効果：関係者への聞き取りで確認



強み



弱み

- ◆ 認識齟齬の早期発見/解消
- ◆ 問題共有の容易さ
- ◆ 作業管理の容易さ
- ◆ レビュー品質の向上
- ◆ レビューに関する労力の増加
- ◆ 運用面は個人のスキル/積極性に依存
- ◆ アジャイル開発への抵抗



定性的な効果：関係者への聞き取りで確認

当初の目的の「認識齟齬起因のバグ低減」と
「手戻り作業の低減」を実現できた

一方で、今回構築したプロセスの強みだけでなく、
弱みも浮き彫りになった

21/21
22/22

今後は、開発対象ソフトウェアの
規模/特徴を踏まえたうえで
従来プロセス/今回プロセスを適宜選択のうえ
開発に臨んでいく

- ◆ レビュー品質の向上
- ◆ アジャイル開発への抵抗

MOVE THE WORLD FORWARD MITSUBISHI
HEAVY
INDUSTRIES
GROUP

