DENSO
Crafting the Core

異なる開発組織における継続的統合に向けたQueuing統合の提案と適用

白木　徹*
株式会社デンソー
ICT技術1部
toru_shiraki@denso.co.jp

林 健吾
株式会社デンソー
先進安全技術4部
kengo_hayashi@denso.co.jp

目次

1. 背景

- HUD (Head Up Display)とは？

デンソーHP
https://www.denso.com/jp/ja/products-and-services/com/human-machine-interface/

- HUDの世界市場

	2016年見込	2025年予測
市場規模	747億円	3,058億円
2015年比	1.5倍	6.2倍

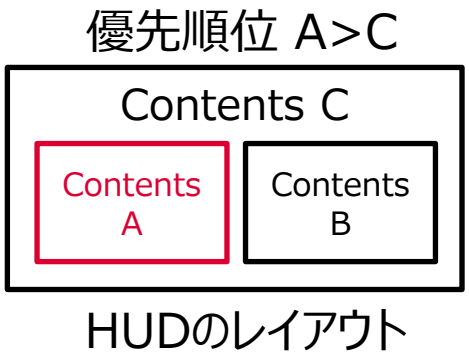
富士キメラ総研
2017年3月2日 https://www.fcr.co.jp/pr/17020.htm

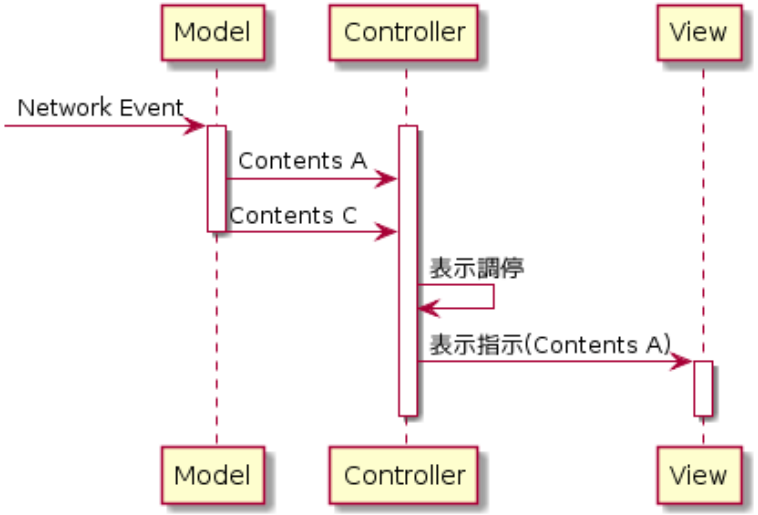
HUDの市場規模の増加　　自動車システムの複雑化

表示するコンテンツ数の増加と搭載車両の増加によって
ソフトウェアの開発量が急増

1. 背景（デンソーHUD開発の現状）

- MVC(Model View Controller)アーキテクチャを採用
  1. Model，Viewはコンテンツごとに独立して開発が可能
  2. Controllerは調停すべき全てのコンテンツを対象に一括して開発が必要



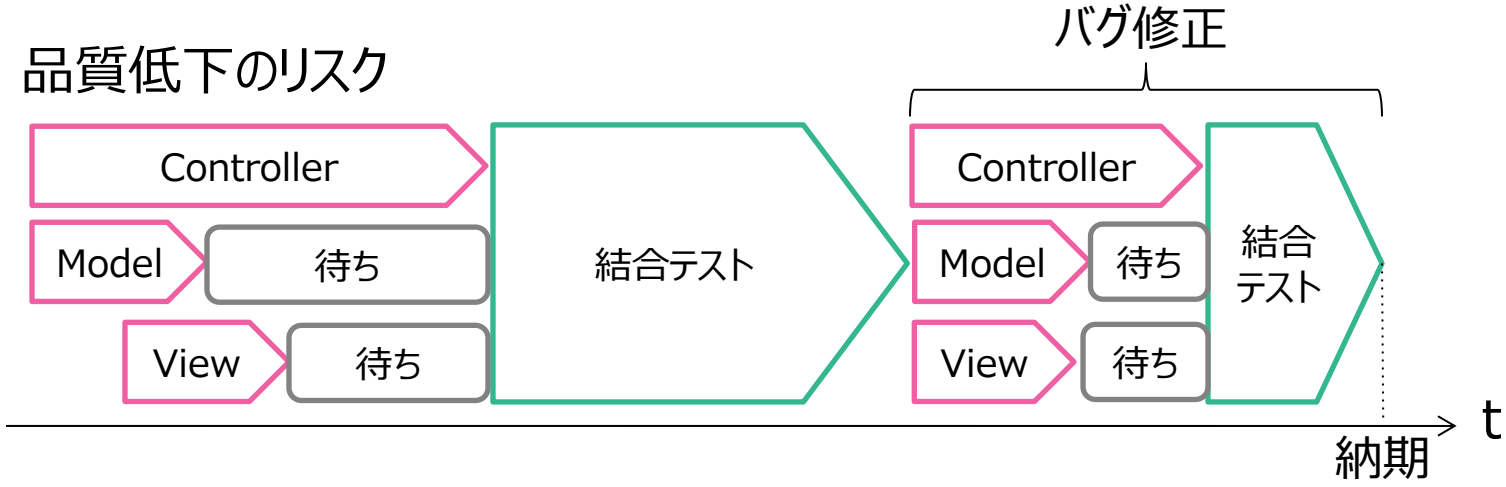


- BI（ビッグバン統合）による開発
  1. 必要となるスキルやツールセットは各サブシステムで異なる
  2. 各サブシステムは異なる組織が開発する
- Viewの開発量が増加
  - 製品個別でViewを変更したいという強い顧客要求

開発量が増加してもBIで開発終盤に結合テストを実施

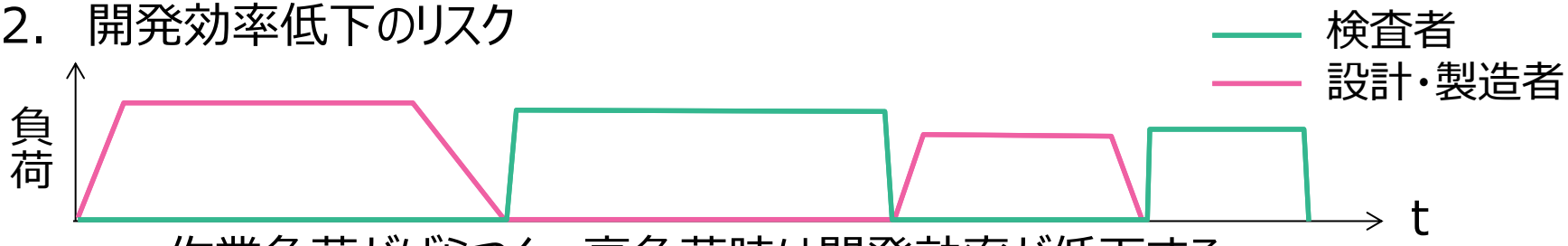
2. 改善したいこと(BIによる開発プロセスのリスク)

1. 品質低下のリスク



結合テストの開始が遅い→ソフトウェアの修正期間が十分にない

2. 開発効率低下のリスク

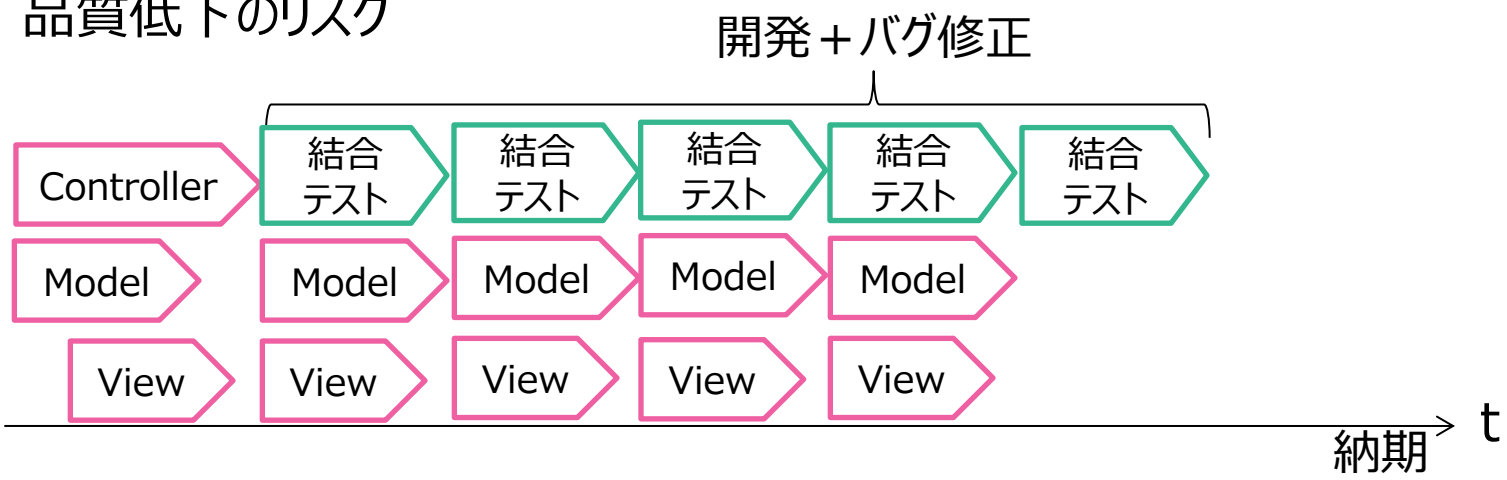


作業負荷がばらつく→高負荷時は開発効率が低下する

システムの統合タイミングを分散・早期化して
インクリメンタルな開発に移行

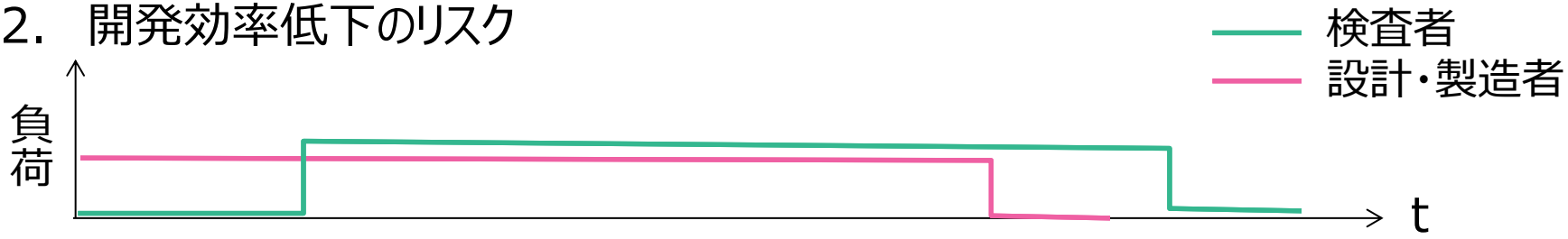
2. 改善したいこと(インクリメンタル開発)

1. 品質低下のリスク



統合時期を早めてバグを早期検出→ソフトウェアの修正期間の確保

2. 開発効率低下のリスク



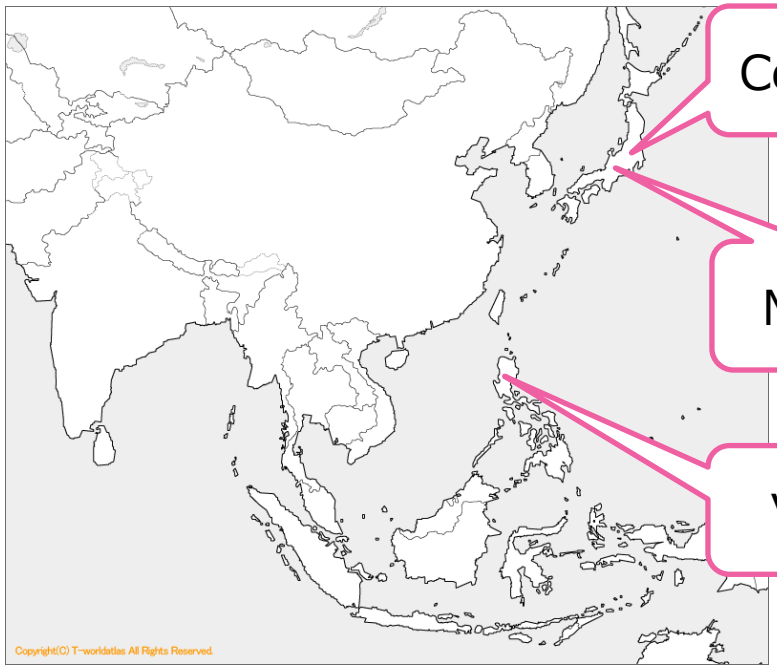
設計・製造・検査を短いサイクルで回す→作業負荷を平準化し開発効率向上

インクリメンタルな開発方法に移行してリスクを低減

3. 経緯

- 継続的統合（CI：Continuous Integration）※へ移行したい

  ※実行可能ソフトウェアとして継続的に統合することで，バグを早期に検出修正することでソフトウェアの品質を高めるアプローチ



１コンポーネントの規模は**大きく**，**分割開発できない**

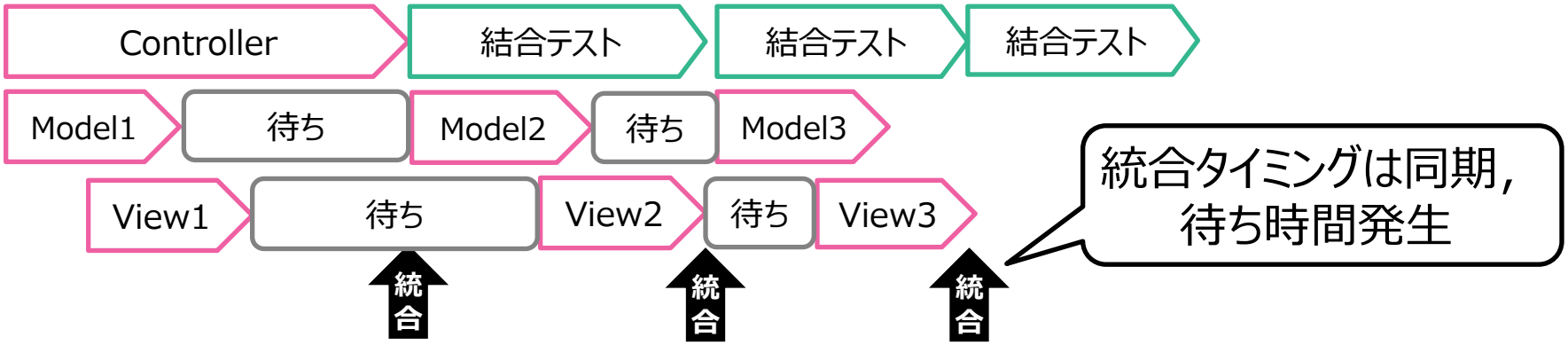
１コンポーネントの規模は**小さく**，開発スピードも**早い**

１コンポーネントの規模は**小さく**，開発は**タイムラグがある**

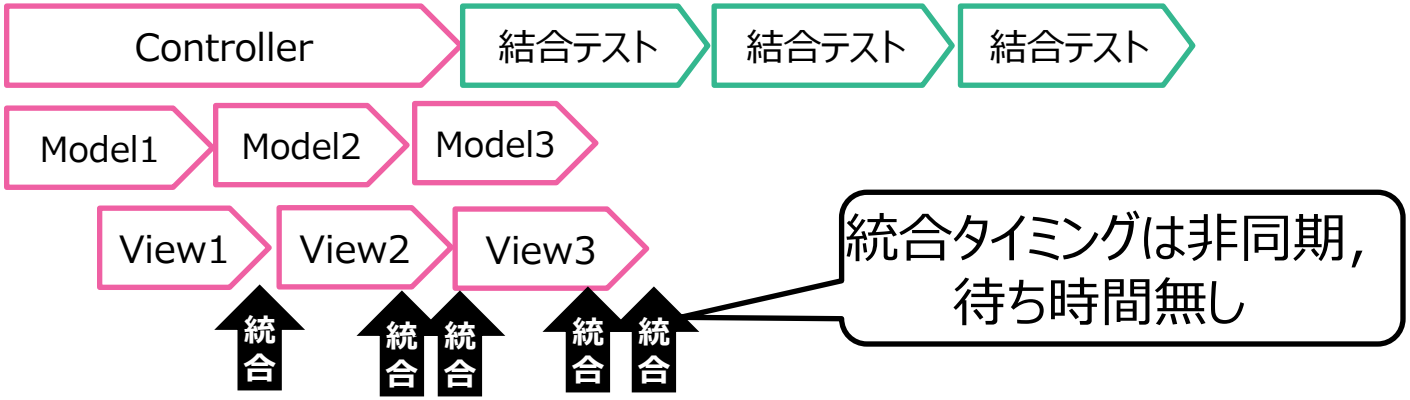
異なる開発組織，開発速度，開発粒度のため完成タイミングがコンポーネントごとに異なる

3. 経緯

1. 開発速度が早いサブシステムで**待ちが発生**し開発効率は悪化



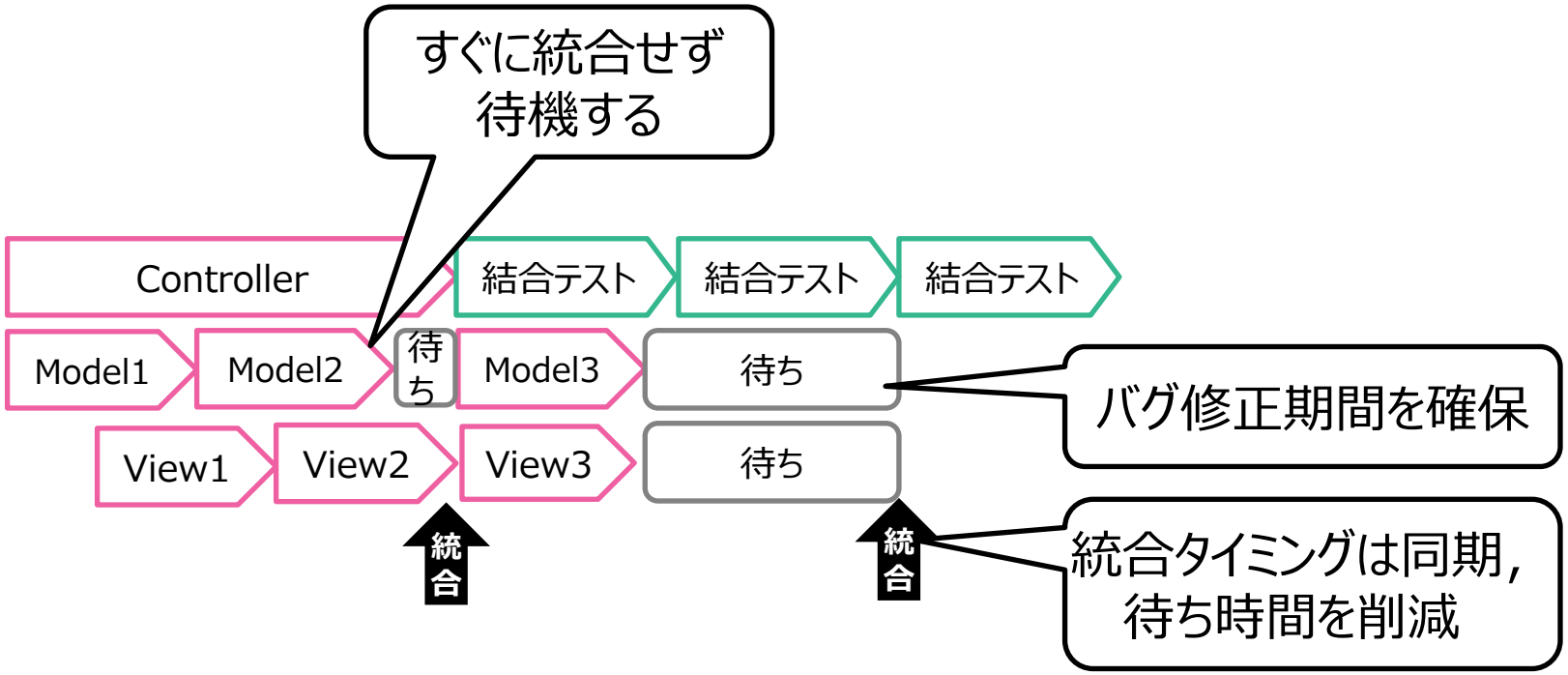
2. 不整合のある組合せが生じ，**不具合が無秩序**に生じて品質確保が困難



各サブシステムの開発速度を妨げない統合方法を考案

3. 経緯（アプローチ）

統合タイミングは同期しつつ，待ち時間を少なくする

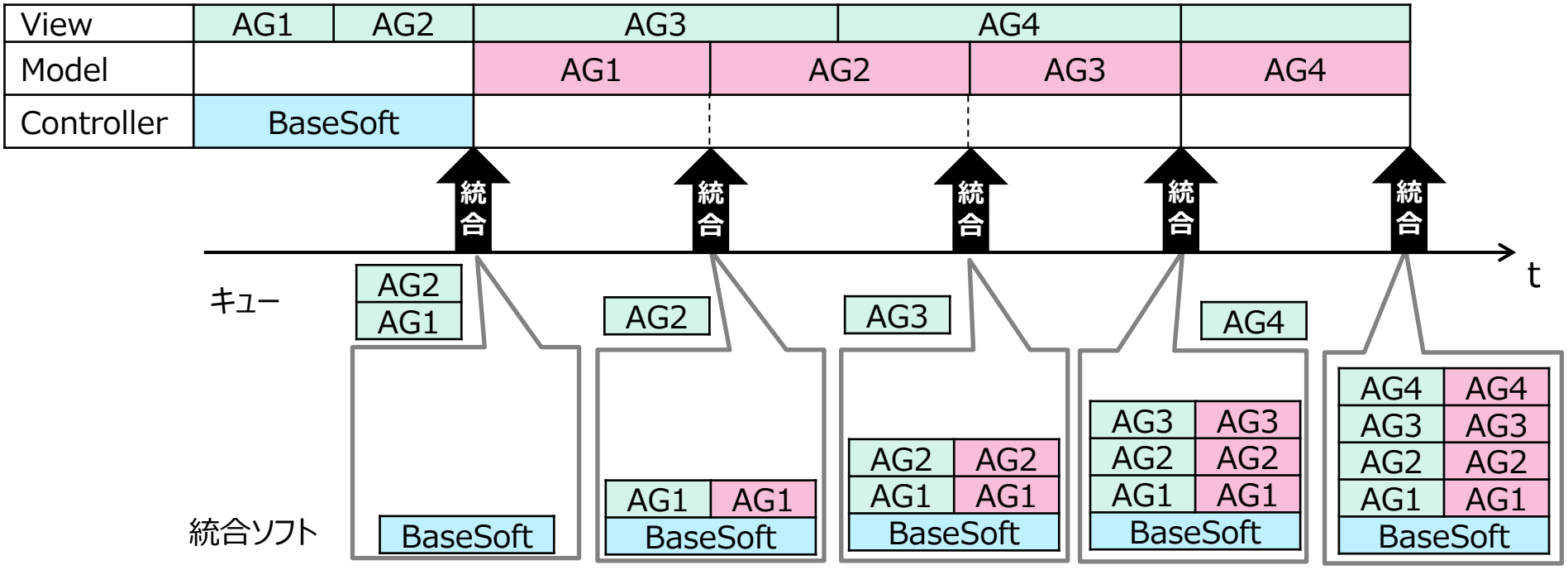


統合タイミングをルールに従って同期させるQueuing統合を提案

4.1. QI(Queuing 統合)の導入

統合のルール
① AGの開発順序はサブシステム間で同じ
② ModelとViewのAGが揃ったらすぐに統合する
③ ModelとViewのAGのタイミングが合わない場合はキューに加える

凡例
AG ：Application Group
(M)：Model
(V)：View
(C)：Controller



結合テストで発見するバグの解析対象が局所化できる

4.2. HUD開発への適用方法

事前計画の立案

- 統合のタイミングをタイムボックス（1週間）で区切る

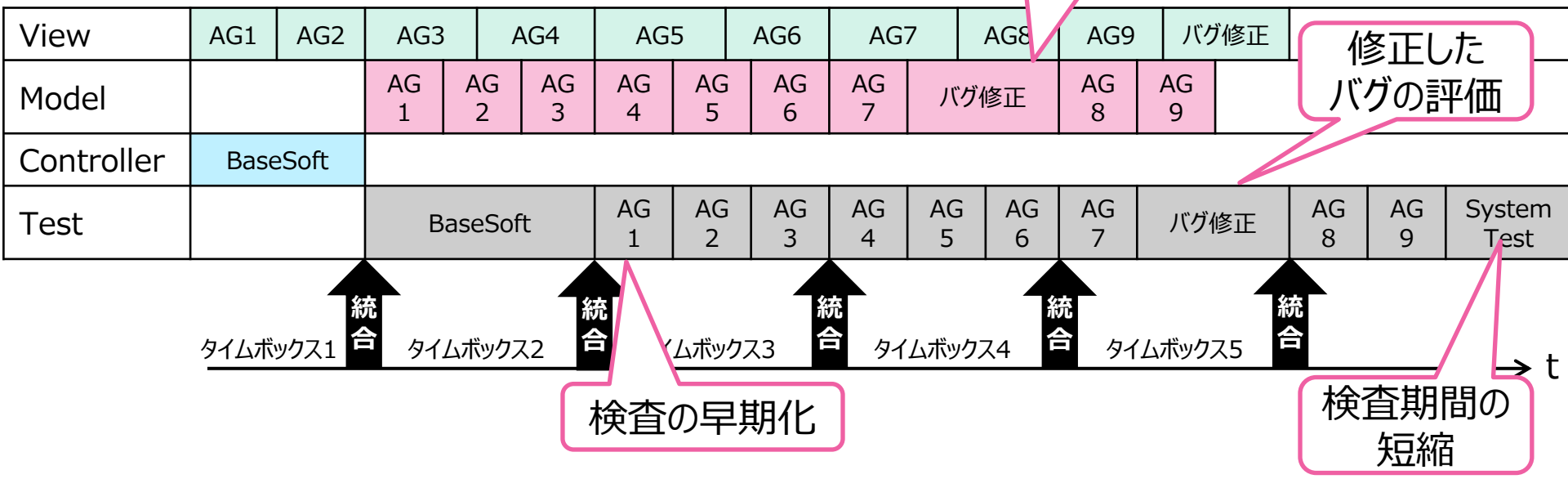
- 開発メンバーへの要求の出し方を変更
  - BIでは，統合までに開発が完了していればよく，開発順序は無秩序
  - 計画に対してAG開発が完了した時点でアウトプットを出すよう徹底

- 各タイムボックス内で開発するAGを決める
  - 開発の優先順位を定める

4.3. HUD開発への適用

- AG開発の優先順位を定めて，事前に統合計画を立案
  1. Controllerを優先して開発
  2. 移植などの開発が容易なAGを開発
  3. 変更規模が少ないAGを開発
  4. 変更規模が多い，新規開発となるAGを開発

※タイムボックス＝1週間



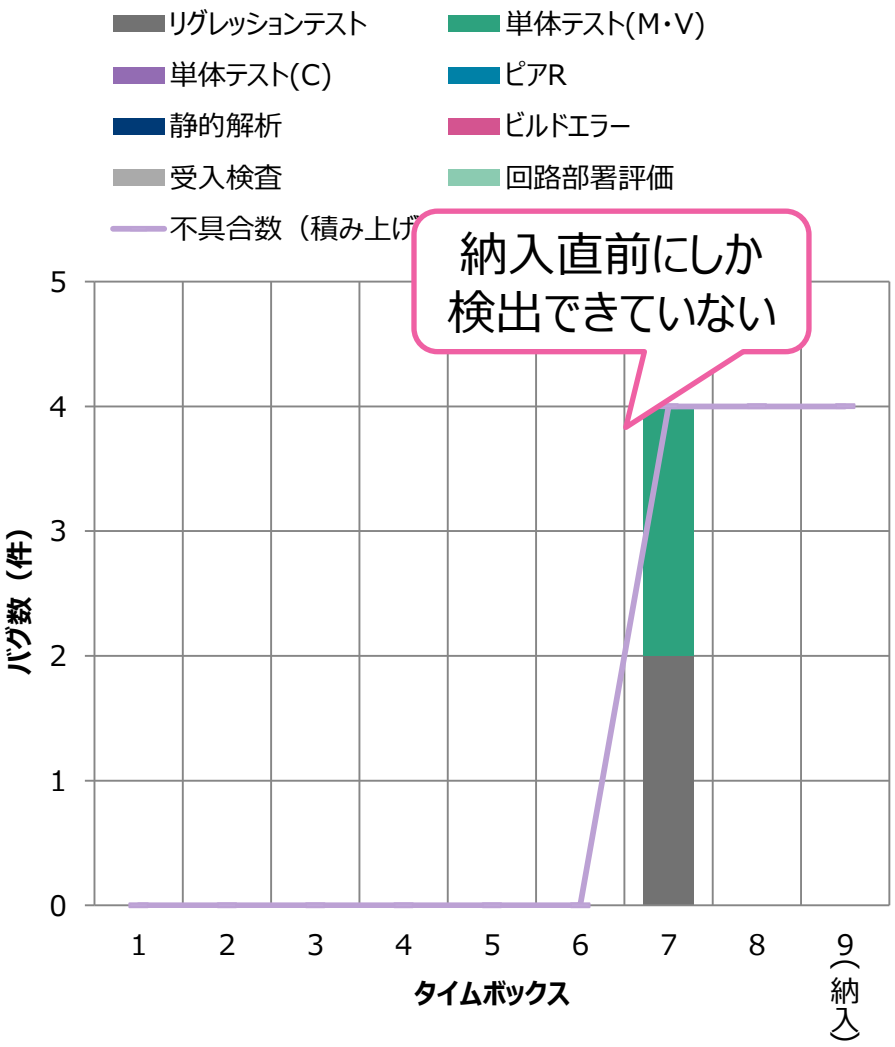
検査の早期化，結合テストの評価期間の短縮が可能

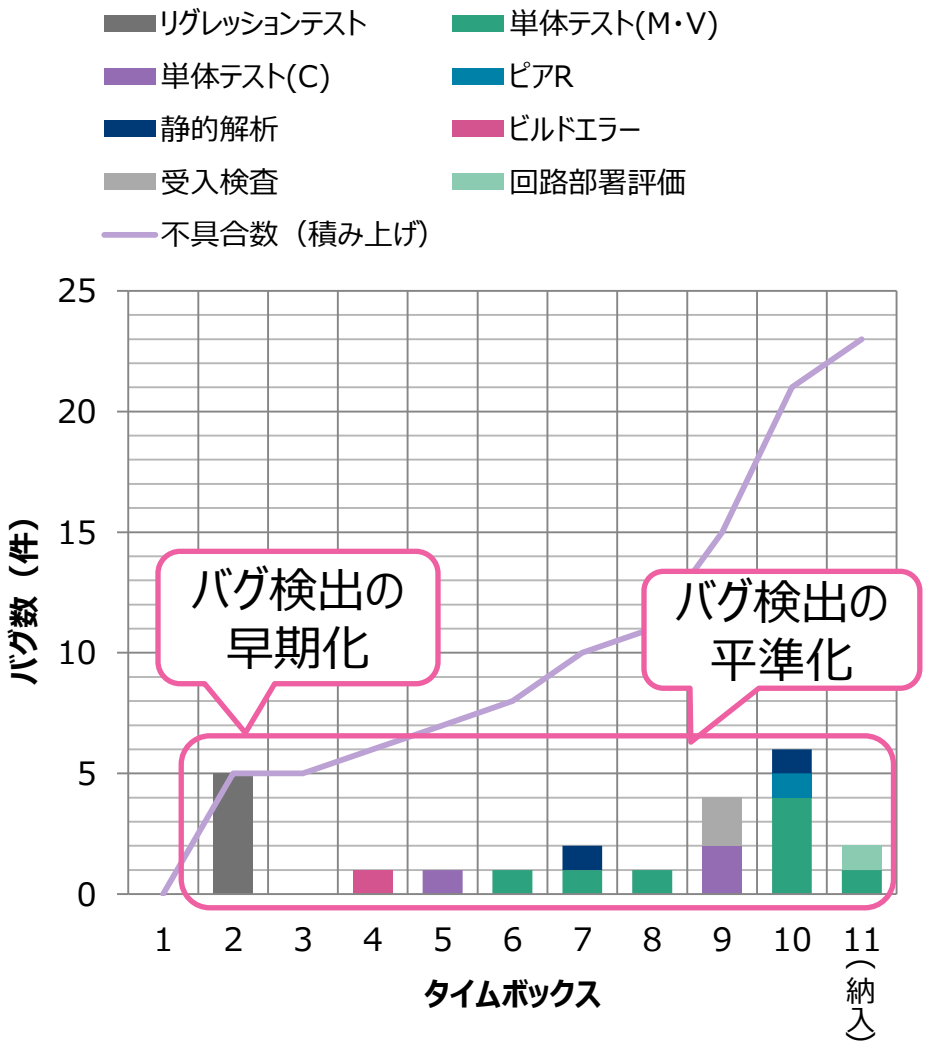
5.1. 評価条件

- 評価対象
  - 同一の開発車両，同一の開発メンバ
    ※開発時期によって開発規模は異なる

- 評価指標
  1. バグ曲線
  2. 工程別の月毎の開発工数の推移

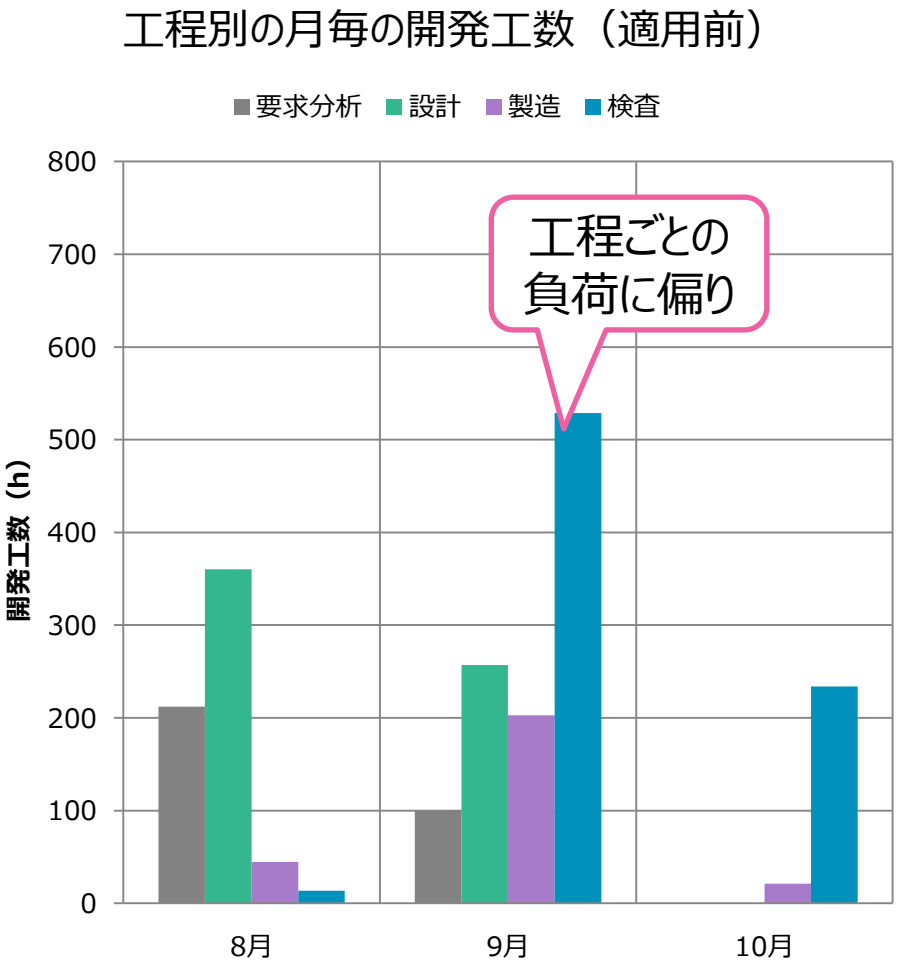
5.2. 評価結果[バグ検出の早期化]

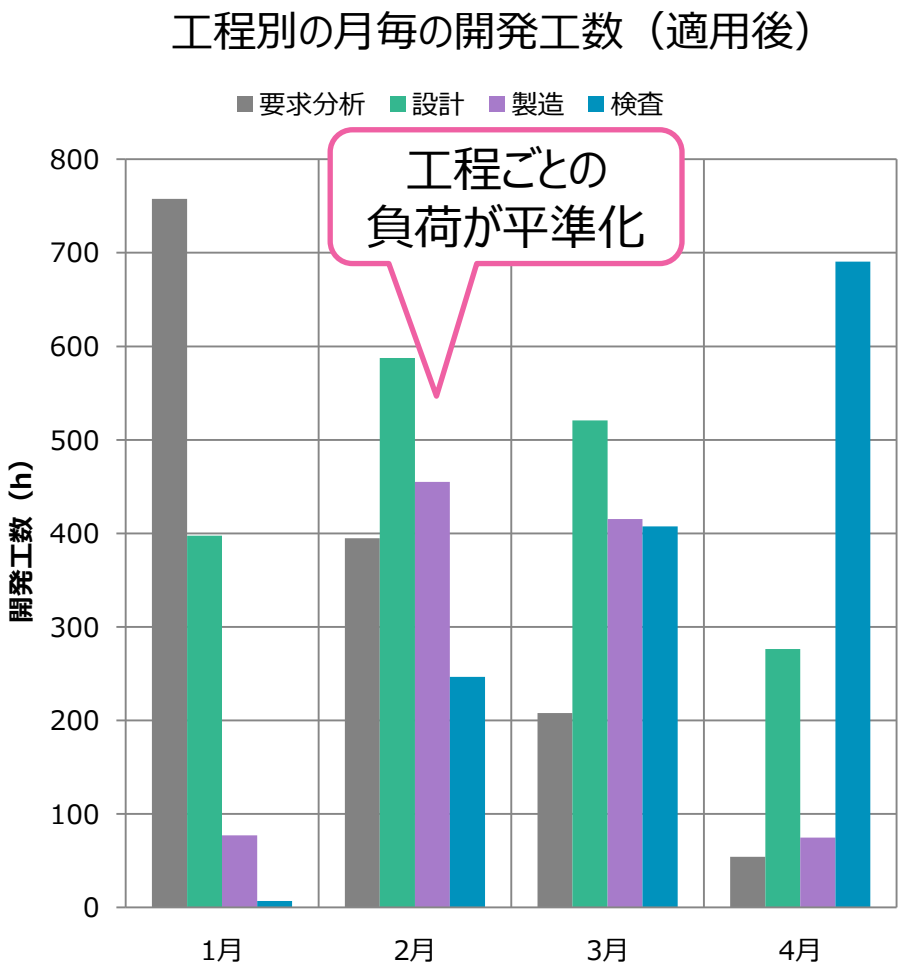




バグ検出の早期化，バグ検出の平準化を実現

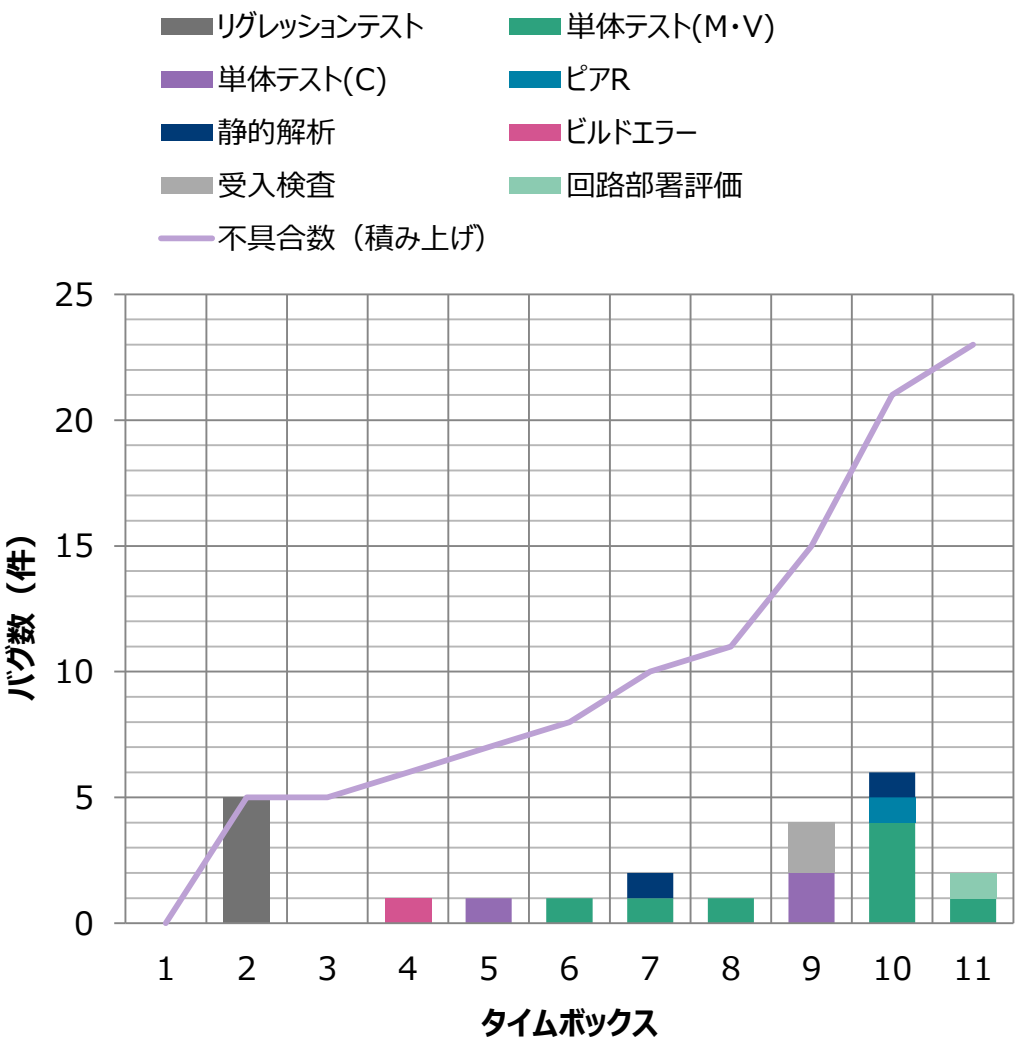
5.3. 評価結果[工程別の月毎の開発工数]



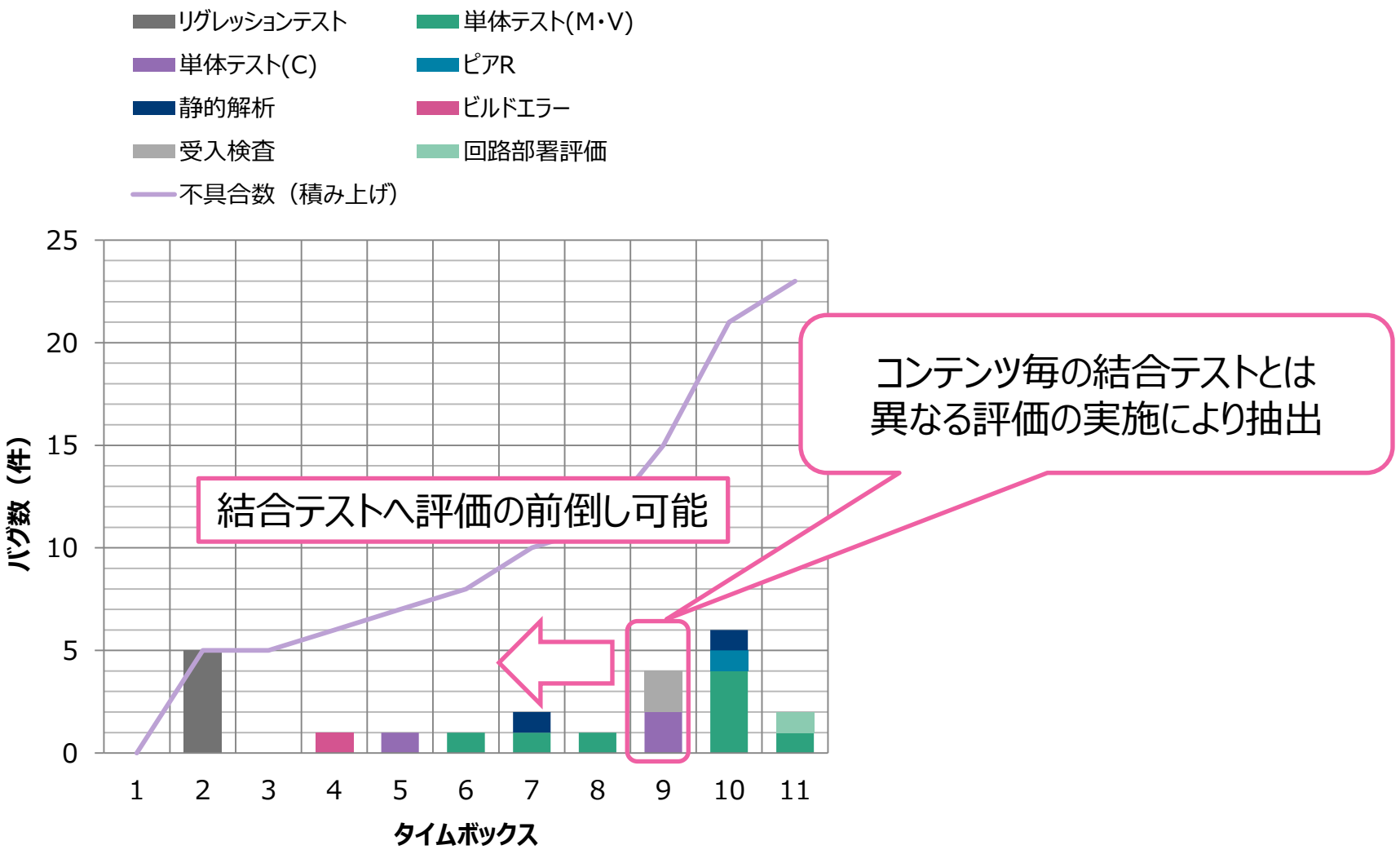


工程ごとの負荷が平準化できている

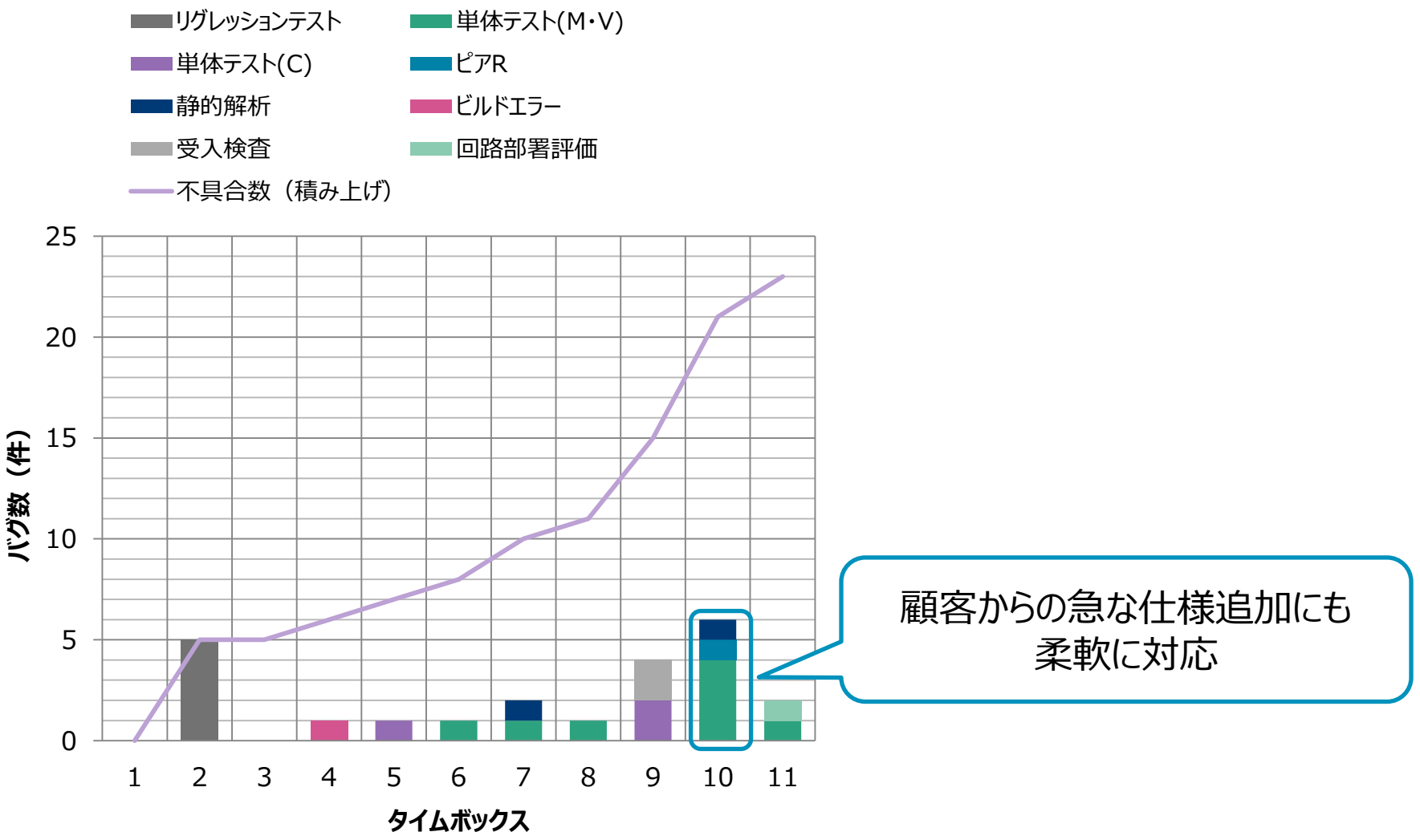
6. 考察①



6. 考察①



6. 考察①



6. 考察②

インクリメンタル開発導入の副次的効果

＋ 1. 事前に計画を立案し開発期間が一定期間に区切られることで
誰が何をいつまでに開発するのかが明確化

- 開発チームの開発能力の把握とチーム全体のリソースの管理が容易
- 急な仕様変更に対する対応可否の判断が可能

－ 2. 開発スケジュールの管理と開発メンバへの周知が大変

- 仕様変更やバグの発生状況によって計画の見直しやリソースの割当を
都度実施することによる管理コストが増加

→管理コスト増加を低減する施策が必要

上手くいった要因：事前の統合計画の立案

7. まとめと今後の課題

- まとめ
  - 異なる開発組織における継続的統合に向けたQIを導入した
  - バグ検出の早期化と開発チームの負荷の平準化を行うことができた
  - 納期の達成と品質の確保をすることができた

- 今後の課題
  - 管理コスト増加
    - プロジェクト管理ツールやデイリースクラムの導入を検討する
    - メンバの入れ替わりがあったときでも本開発方法を適用できるよう，ガイドラインやプロセス定義の整備を進め，開発プロセスの標準化を進める

DENSO

Crafting the Core