

派生開発カンファレンス2016

# 並行開発におけるSCRUMを応用した リソース最適化手法の提案

車載ソフトウェアの車両展開  
への適用とその効果

林 健吾  
(株)デンソー

kengo\_hayashi@denso.co.jp



横浜市開港記念会館, 2016.5.27.Fri

# 発表内容

- 1.事例背景
- 2.人的リソース計画における課題
- 3.関連研究
- 4.車両展開開発における反復性
- 5.SCRUMを応用したリソース最適化手法
- 6.評価
- 7.考察
- 8.まとめ

# 1. デンソーの主要製品と事例の対象製品

## ● 安心・安全

### 走行支援用センシングシステム、

ABS用アクチュエータ、  
ヘッドランプコントロールシステム、  
エアバッグ用センサ、  
車両周辺監視システム…

## ● 環境

ハイブリッド車・電気自動車用製品、  
エンジンマネジメントシステム、  
スタータ、オルタネータ…

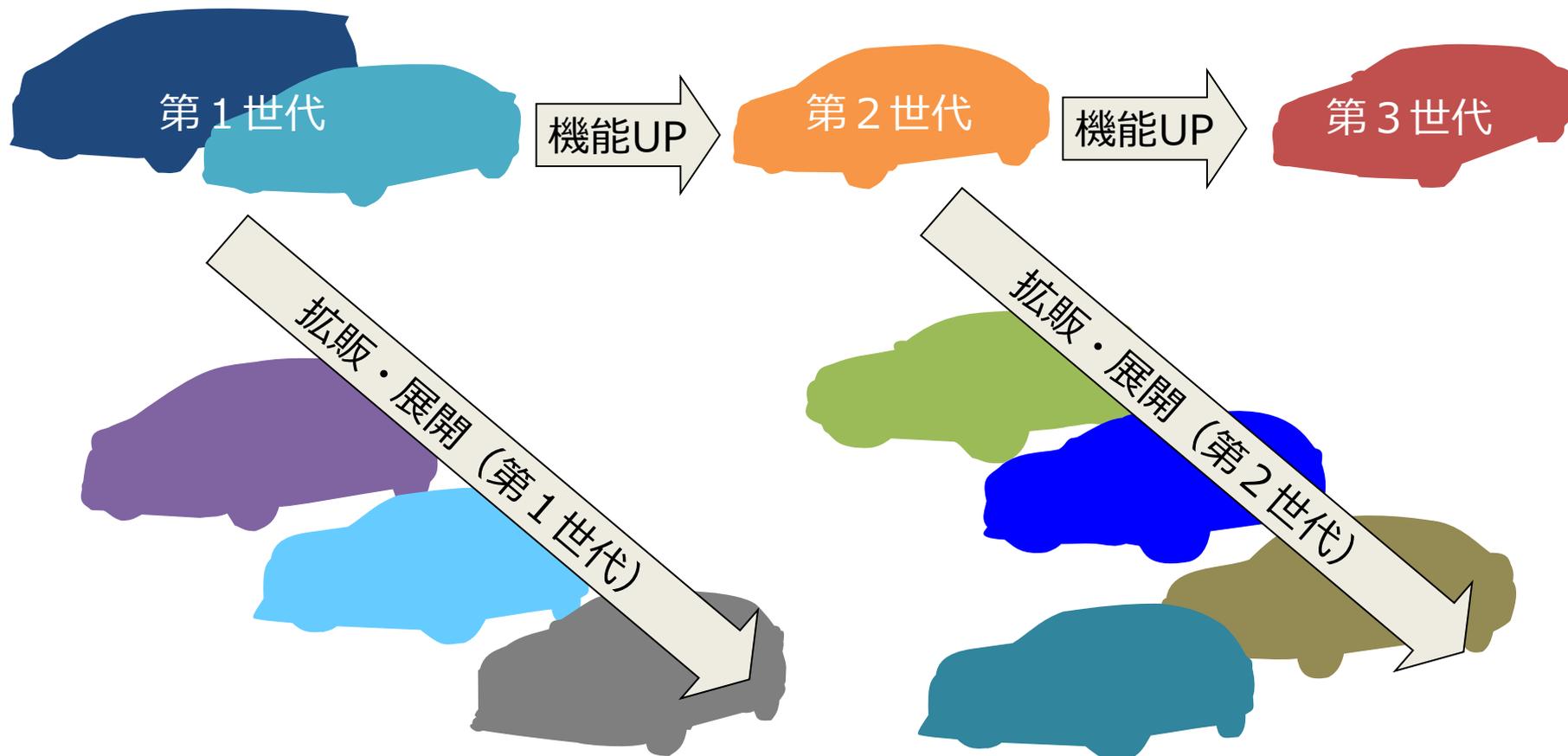
## ● 快適・利便

カーナビゲーションシステム、  
スマートキー、車両運用システム、  
カーエアコンシステム…



# 1. 走行支援センシングシステム製品の世界

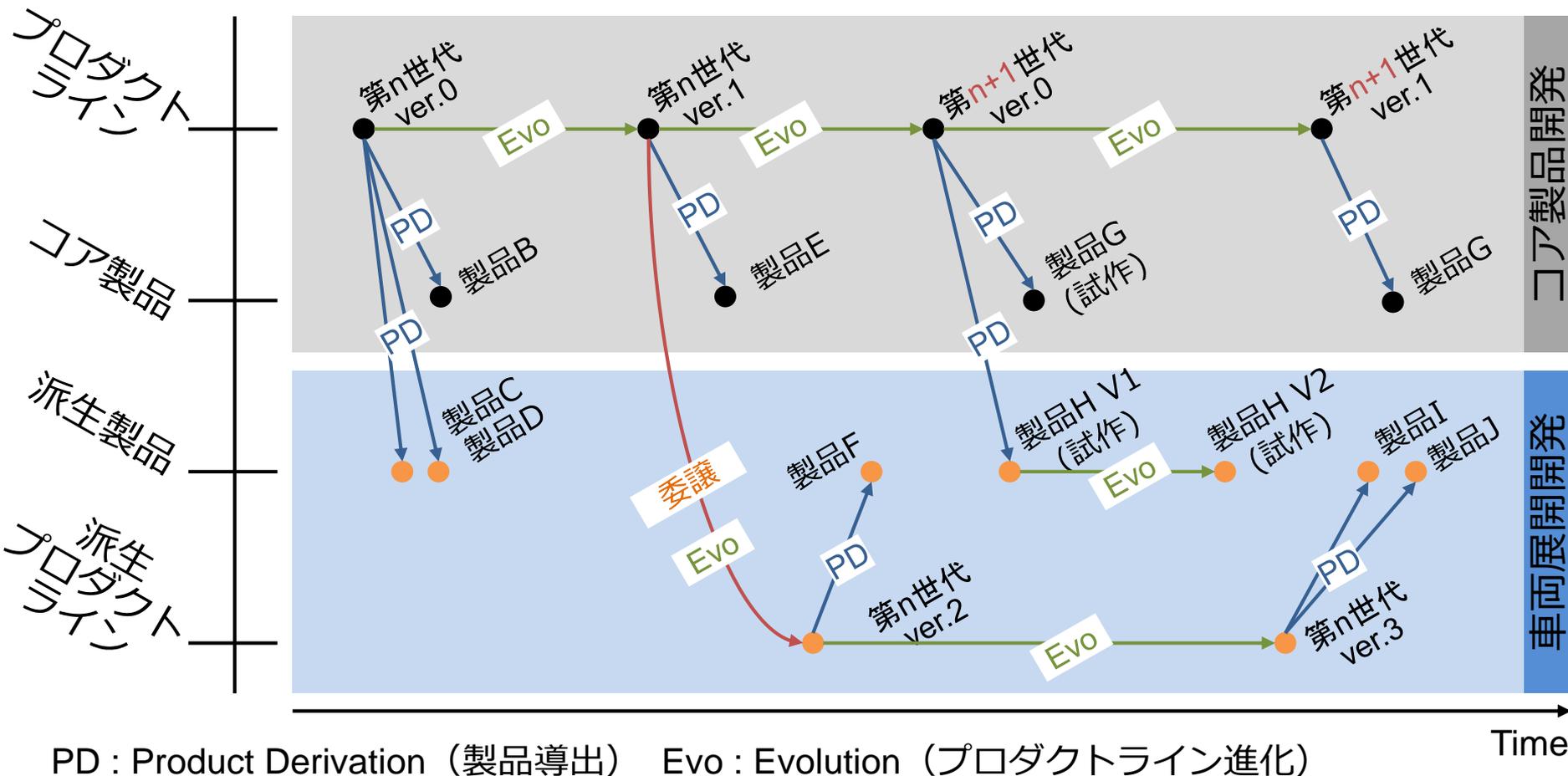
技術競争が活発でこれから拡大が期待される領域



機能性を向上しながら多くの車両に展開していく開発体制が必要

# 1. 製品開発の組織と分担

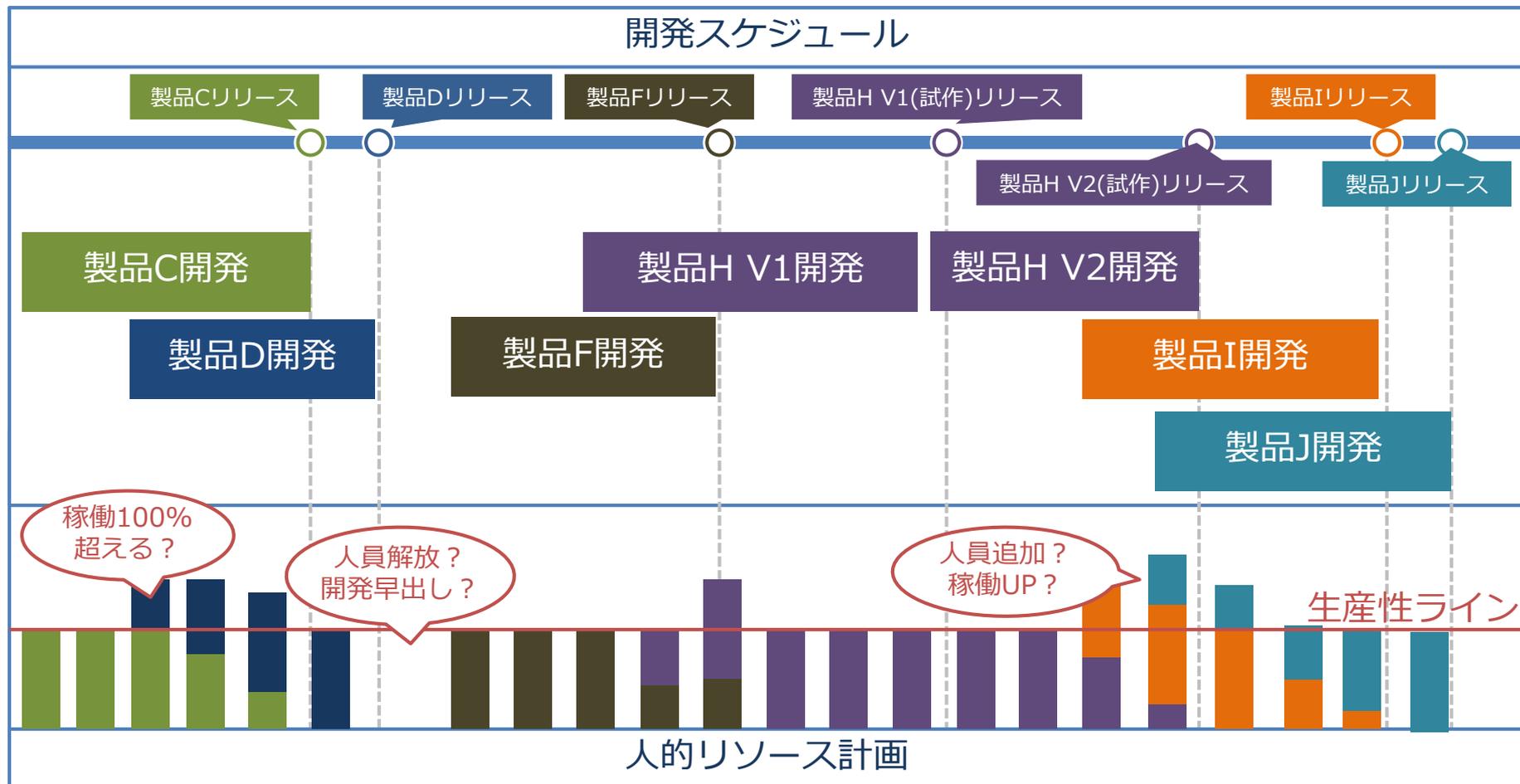
コア製品開発と車両展開開発の2つの組織で開発を分担



多様な製品を並行して開発する車両展開開発が事例の対象

## 2. 人的リソース計画における課題

並行開発を数多くやりくりするには人的リソース計画が重要



多様な製品開発の開発量の見積りが困難⇒納期リスクの発生

### 3. 関連研究① XDDPとプロダクトライン開発

## XDDP

- ・ 派生開発に対応した後戻りの少ない開発方法を提供
- ・ 要求仕様特定後の開発計画は安定するが、計画時点での見積もりや開発管理は未規定

## ソフトウェアプロダクトライン開発 (SPLE)

- ・ 同種同系列のソフトウェア製品の多様性を吸収
- ・ アーキテクチャの研究は盛んだが、個々の製品開発における管理の連携は未確立

複数の派生製品開発を統括した開発管理手法が必要

### 3. 関連研究② SCRUM

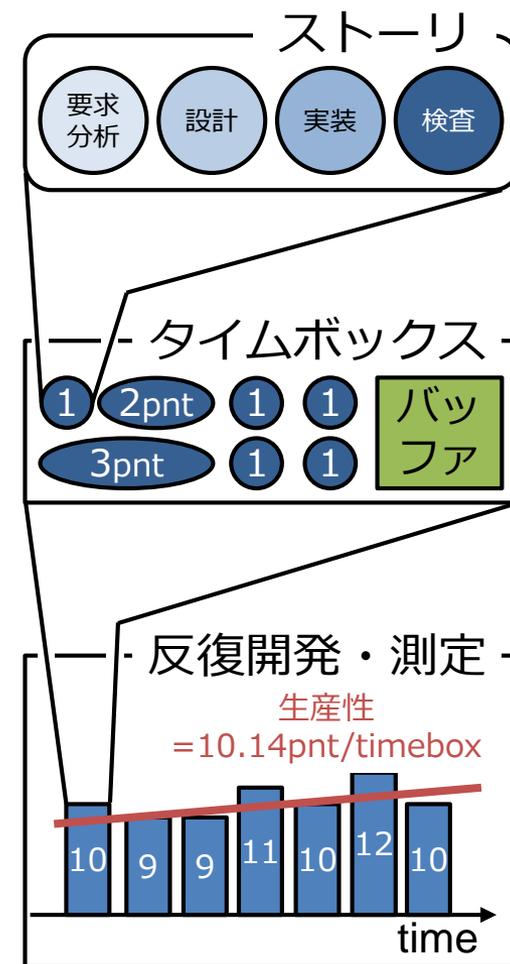
反復性に着目した開発管理手法を提供

#### ① ストーリ：反復の単位

フィーチャ単位で開発サイクルを包括  
開発規模を相対的に見積もって管理

#### ② タイムボックス：ばらつきの吸収 期間を区切った反復的な開発管理 ばらつきを吸収して生産性を安定化

- 短期開発では生産性安定前に開発が完了
- 並行開発での適用事例が少数



派生開発で反復性を見出せれば開発管理手法として応用可能

## 4. 本事例のゴールとアプローチ

### ゴール

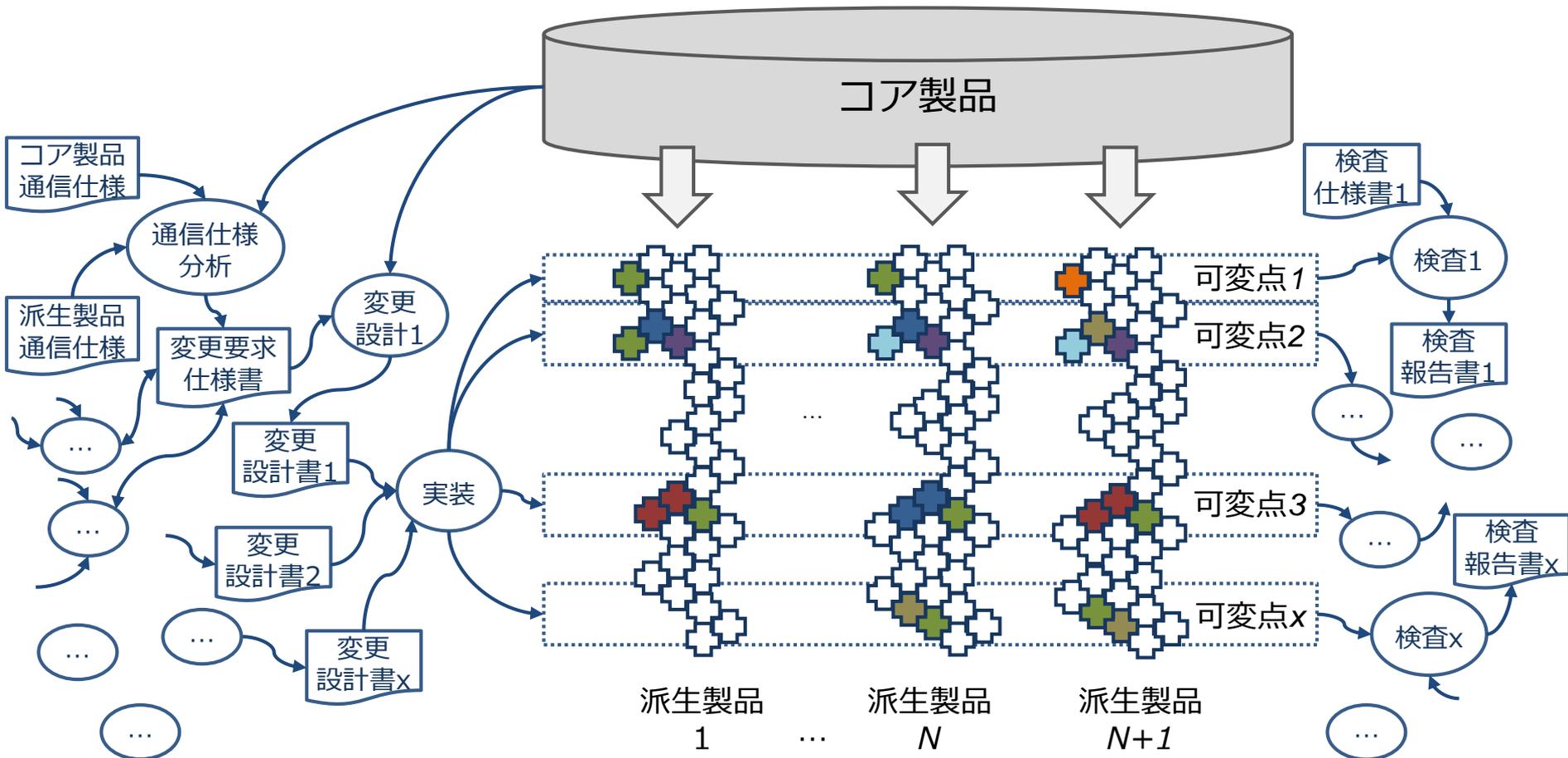
車両展開開発において、  
生産性と開発量をより正確に見積り、  
人的リソース計画と管理を実現する

### アプローチ

プロダクトライン開発の派生開発に存在する、  
開発を跨いで現れるプロセスの反復性に着目し、  
SCRUMを応用したリソース最適化手法を提案する

# 4. 車両展開開発における反復性

派生製品導出のためにコア製品を基に開発が反復される



コア製品の可変点を中心に製品固有だが類似のプロセスが反復される

# 5. SCRUMを応用したリソース最適化手法

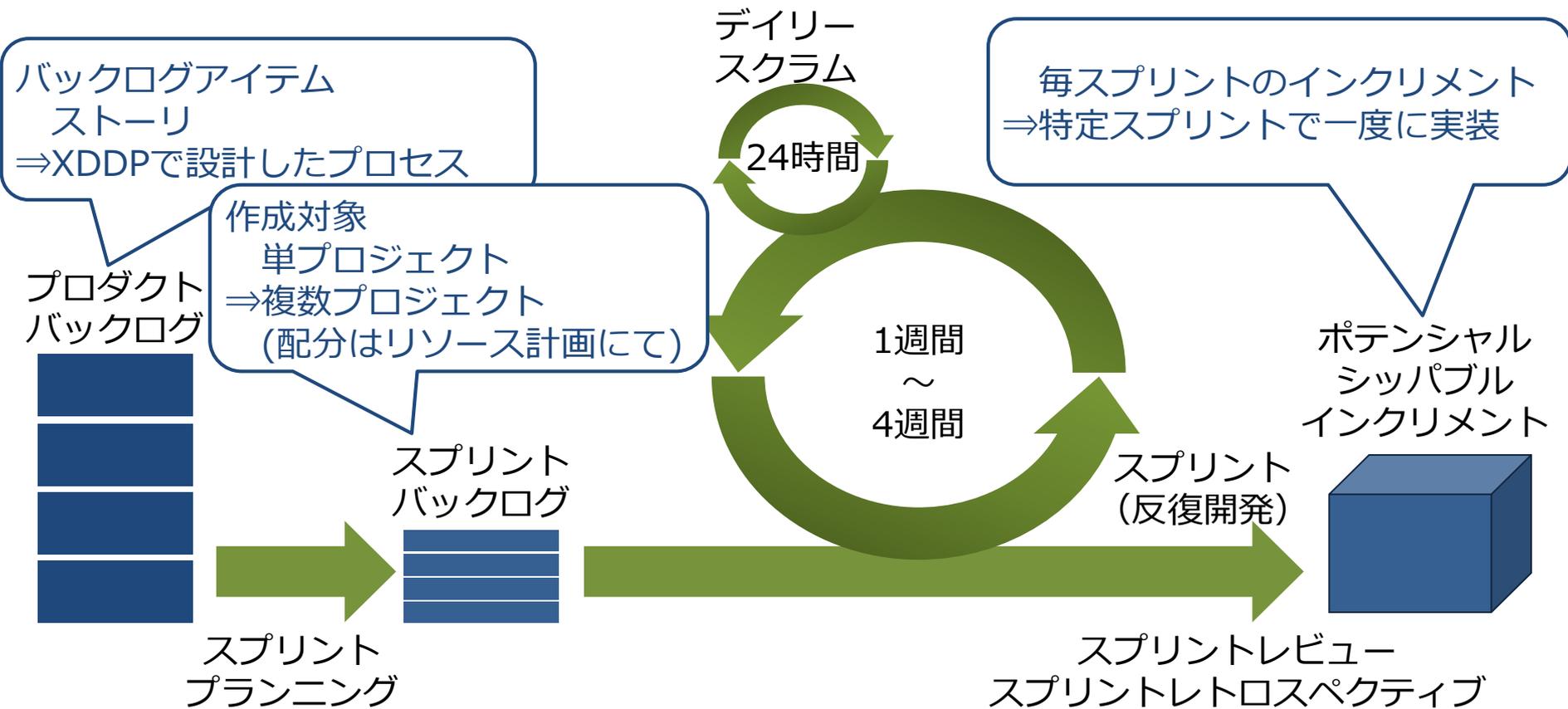
## SPLE + XDDPにSCRUMを開発管理に加えた手法を提案する

	階層	手法	利用技術と着目点
提案	人的リソース管理	—	<ul style="list-style-type: none"><li>・タイムボックス単位で把握した<b>生産性と開発量</b>を元に、<b>並行開発</b>をまとめて人的資源を分配する</li></ul>
提案	開発管理	<b>SCRUM</b> (部分適用)	<ul style="list-style-type: none"><li>・<b>設計プロセス</b>を<b>反復対象</b>としてタイムボックスコントロールで<b>生産性と開発量</b>を見通す</li></ul>
既存	プロセスアーキテクチャ	XDDP	<ul style="list-style-type: none"><li>・開発ドメインと<b>可変点</b>に着目して、<b>再利用(反復)可能なプロセス</b>を設計する</li></ul>
既存	プロダクトアーキテクチャ	SPLE	<ul style="list-style-type: none"><li>・<b>可変点を分析</b>することで、改造対象部品に<b>反復性</b>を与える</li></ul>

反復性に着目して4つの階層で人的リソース管理につなげる

# 5. SCRUMフレームワークの部分適用

## SCRUMのフレームワークを部分的に適用する



反復する対象を変えたことに開発手法を適応させる

## 5. バックログに登録するプロセスアイテム①

大分類	分類	プロセス	概要	反復性
要求分析	-	通信仕様差分を抽出する	コア製品仕様との差分を分析して変化点を抽出する	派生製品開発単位で反復する
要件開発	通信判別系 変異体追加	エンジン種類を追加する	通信Xのデータからエンジン種類の変異体を追加する	入出力は異なるが、開発内で類似のプロセスを反復する
		駆動方式を追加する	通信Yのデータから駆動方式の変異体を追加する	
	-	T/Mでシフト判定を変更する	トランスミッションを判定してシフト判定方法を切り替える	一度可変点を開発したら反復されない
リリース	-	通信を回帰テストする	システムの通信セットを全数検査する	派生製品開発単位で反復する

XDDPでプロセスを設計してアイテム登録する

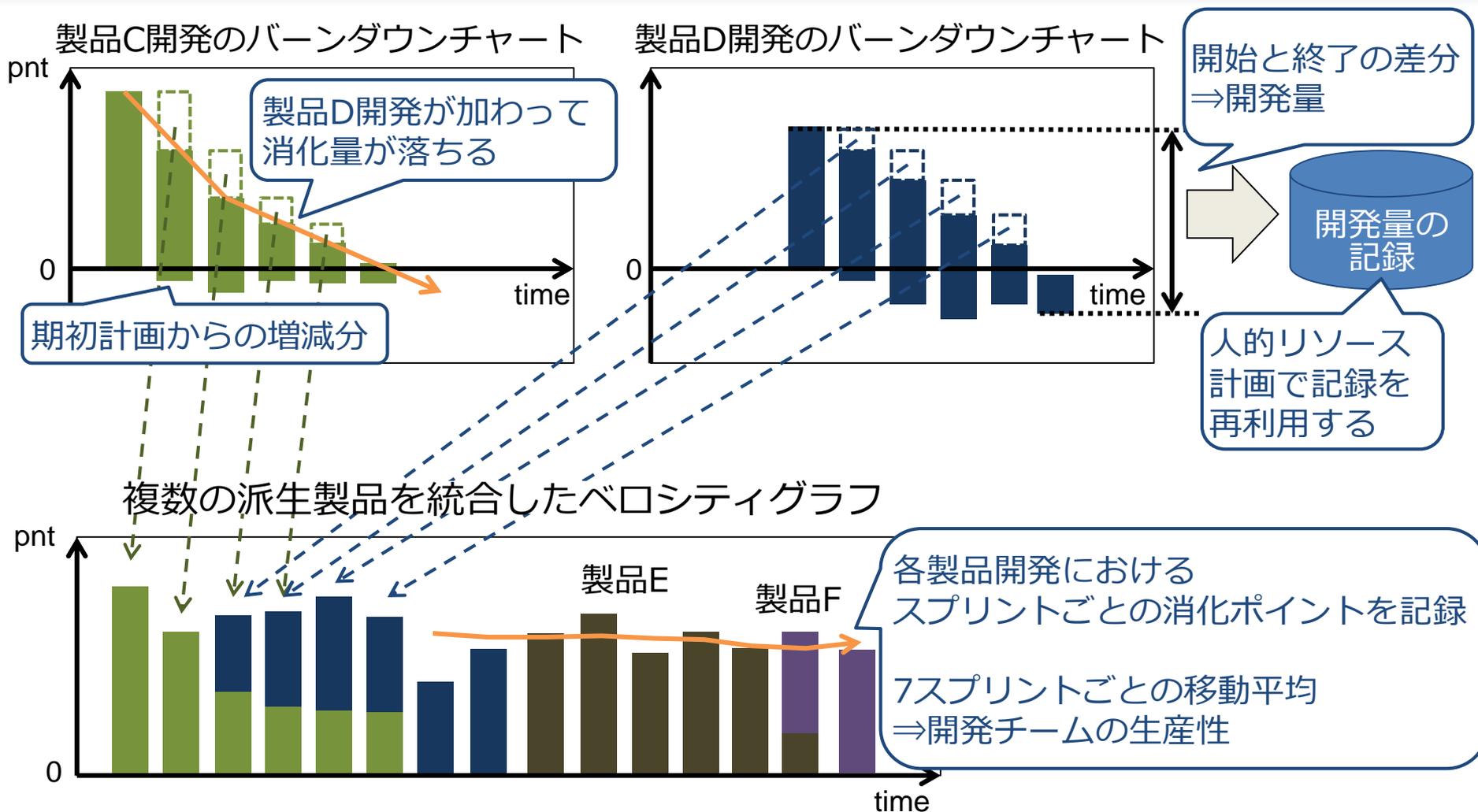
## 5. バックログに登録するプロセスアイテム②

要件開発はプロセスをさらに分割して登録する

分割方法	概要
上流工程	要求仕様化・設計・検査設計の単位で分割する
実装	実装プロセスはすべての上流工程が完了した時点で実行する
下流工程	検査設計は上流工程に含め、検査実施のみを切り出す

スプリントに対して作業規模が小さくなるよう分割する

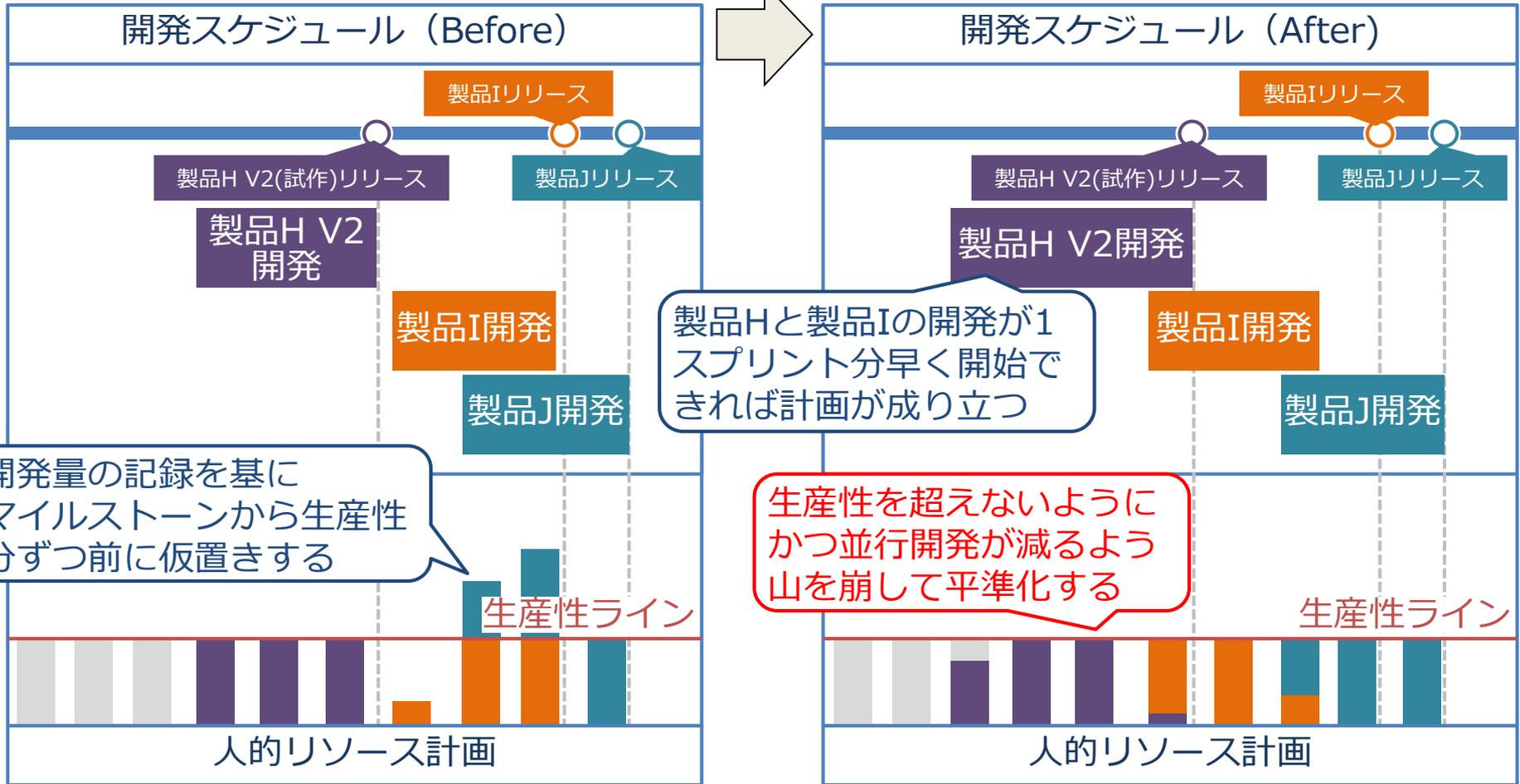
# 5. 開発量の記録と生産性の計測



統合測定したベロシティが生産性 ⇒ 計画のガイドライン

# 5. 人的リソース計画と平準化

## 開発量と生産性から人的リソース計画を立案する



開発量と生産性が把握できれば従来の平準化で計画できる

## 5. 計測結果の振り返りによる規模の正確さ向上

- ・ スプリントでのストーリーの学習は働かない
- ・ 次の派生開発に向けて見積りと実績を振り返り、  
見積りの粒度と各プロセスの規模感を調整する

プロセス	見積り(pnt)	実績(H)	見積り(pnt)
A	3	56.00	13
B	1	7.50	2
C	1	6.25	2
D	2	10.25	3
E	1	3.75	1

3pntは小さい

これくらいが1pntの目安かも

- ・ 実績はばらつく⇒規模の見直しは大まかで十分
- ・ フィボナッチ数列\*などを利用して更新する

\*フィボナッチ数列…{1,2,3,5,8,13, …}と、2つの前の数字を加えて作られる数列

見積りにフィードバックを掛けて計画の正確さを向上させる

## 6. 適用プロジェクト概要

### 提案手法を適用した車両展開開発

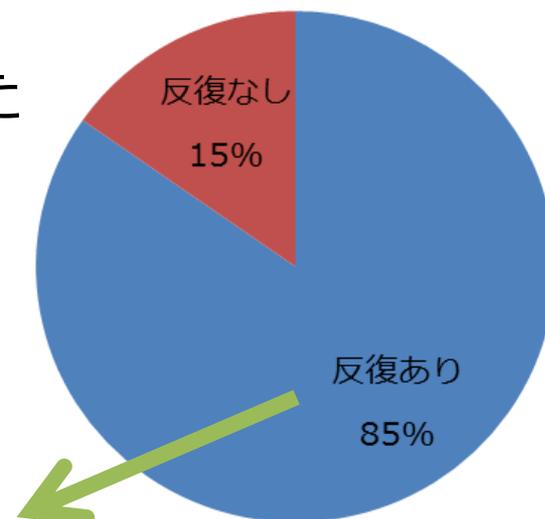
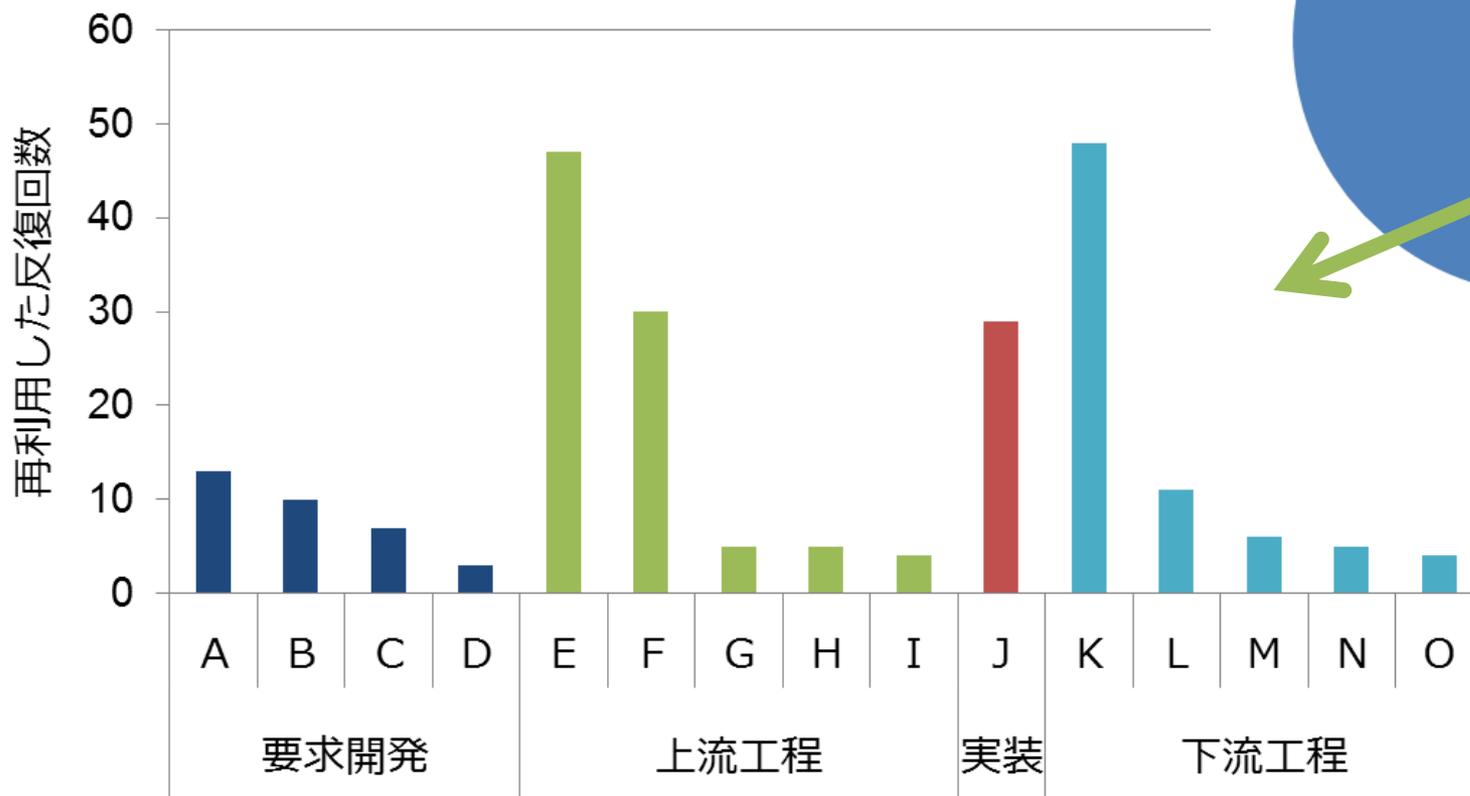
適用期間	2015年7月～2016年4月
適用月数	10か月
スプリント単位	2週間
スプリント数	22スプリント
適用プロジェクト数	11
開発コード量(KLOC)	1～20

- ・ チームメンバは第3スプリントで増員して以降変動なし
- ・ 発表者は開発チームのリーダーとして開発に参加

適用した実プロジェクトのデータで提案手法の有効性を評価

## 6. 反復性の評価

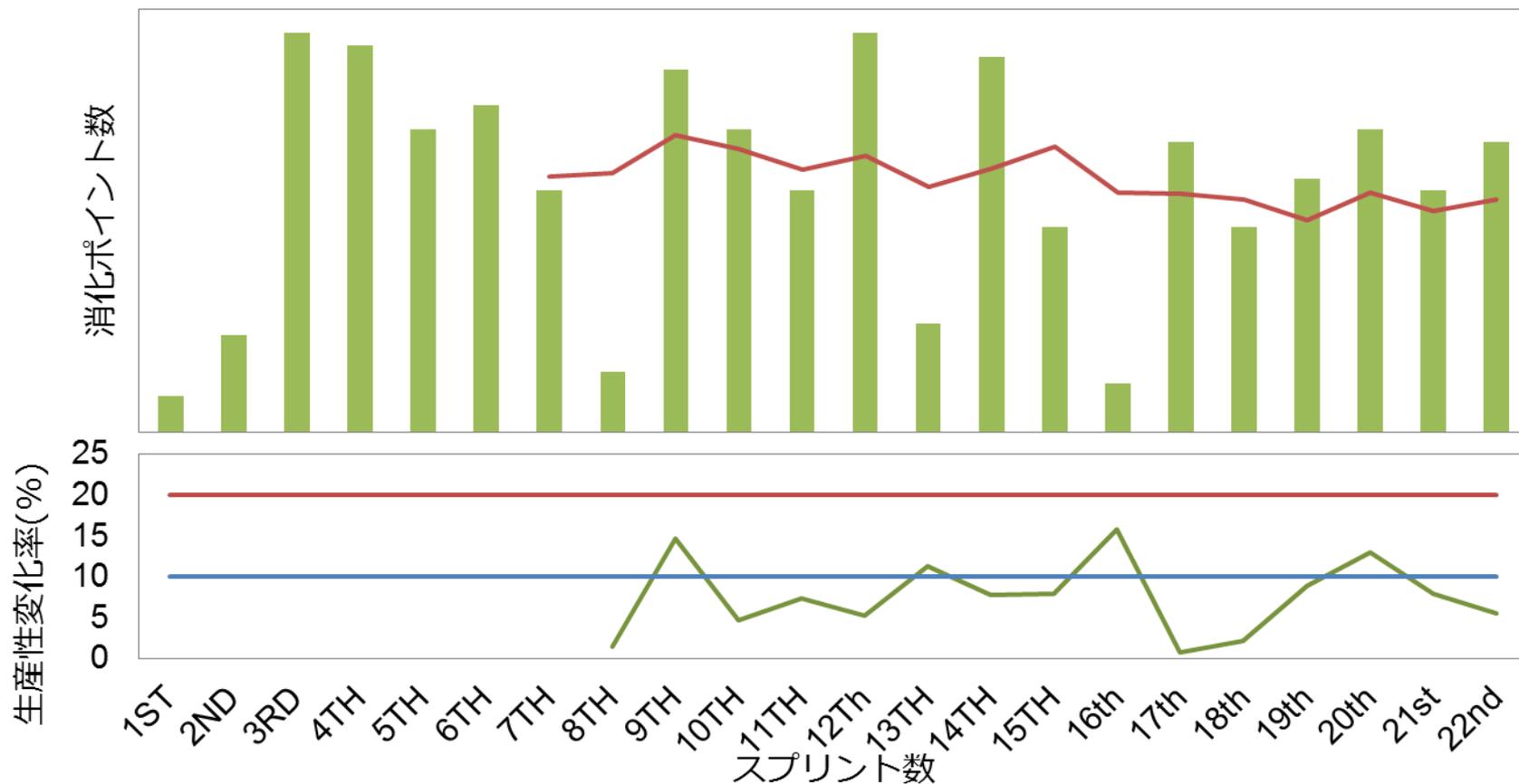
- ・ 類似の成果物、類似の手順を実行するプロセスの反復回数を集計して評価した
- ・ 反復性あり = プロセスが再利用された



車両展開開発のプロセスに反復性があることが確認された

## 6. 生産性の安定性評価

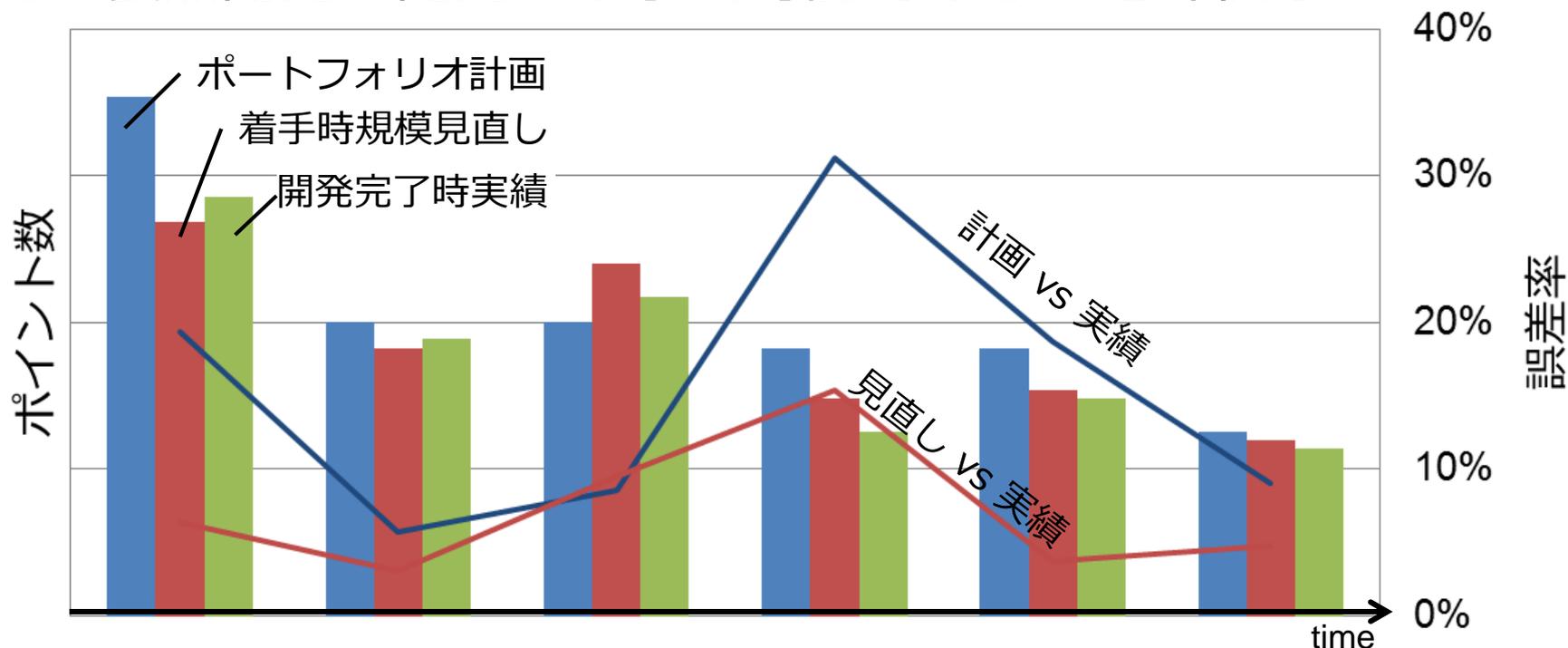
ベロシティ(7スプリント分の移動平均)の変化率から、生産性が予測可能であるかを評価した



反復性のある開発を対象とした場合に生産性は予測可能

## 6. 開発量の推定評価

類似の製品開発での開発量の記録から、  
先の製品開発の開発量が予測可能であるかを評価した

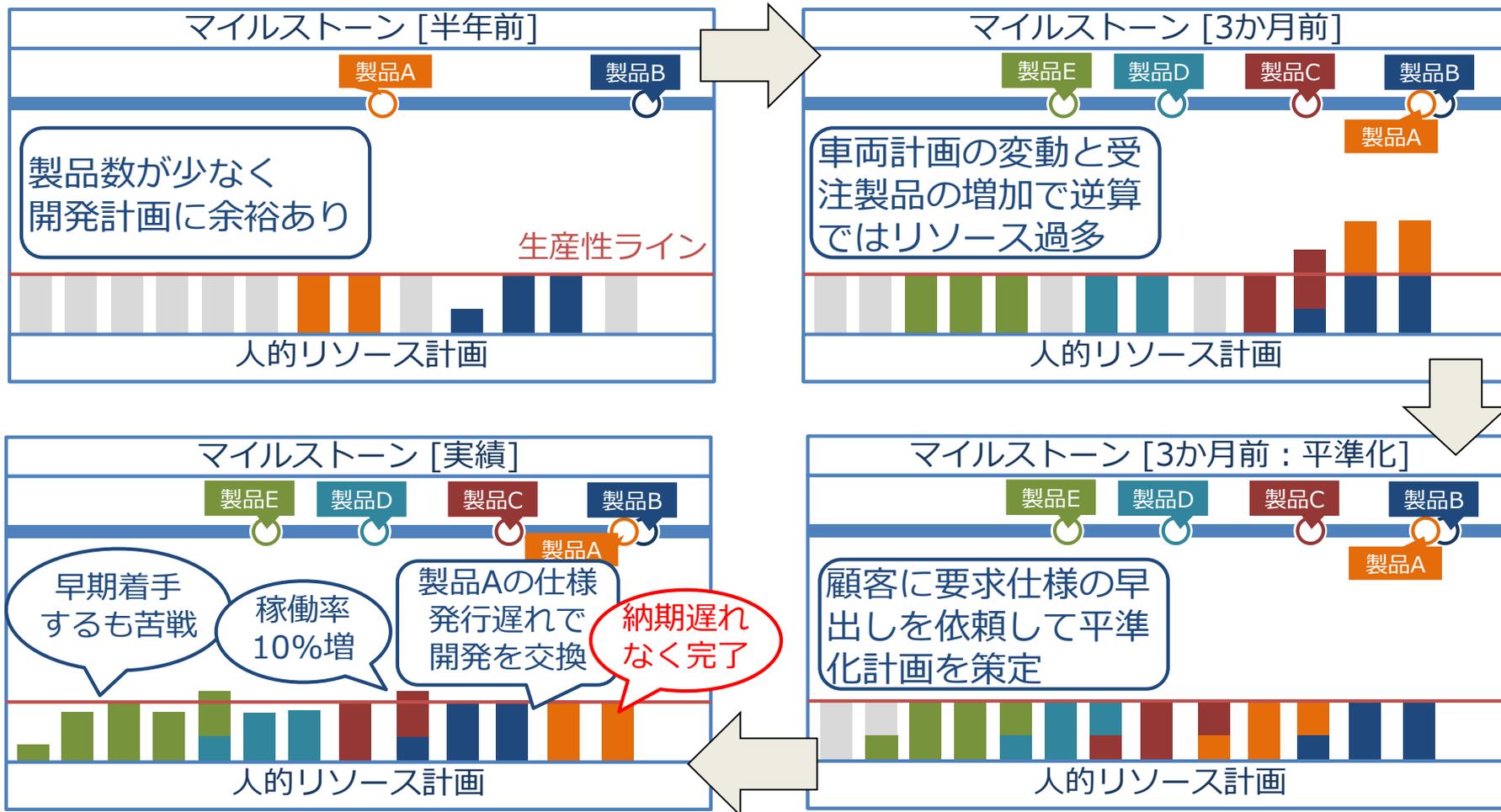


11のプロジェクトの内、類似する6プロジェクトを選び、  
人的リソース計画と着手時の規模見直し、実績を比較した

見積り誤差30%以下の正確さで開発量が予測可能

# 6. 人的リソース計画と管理例

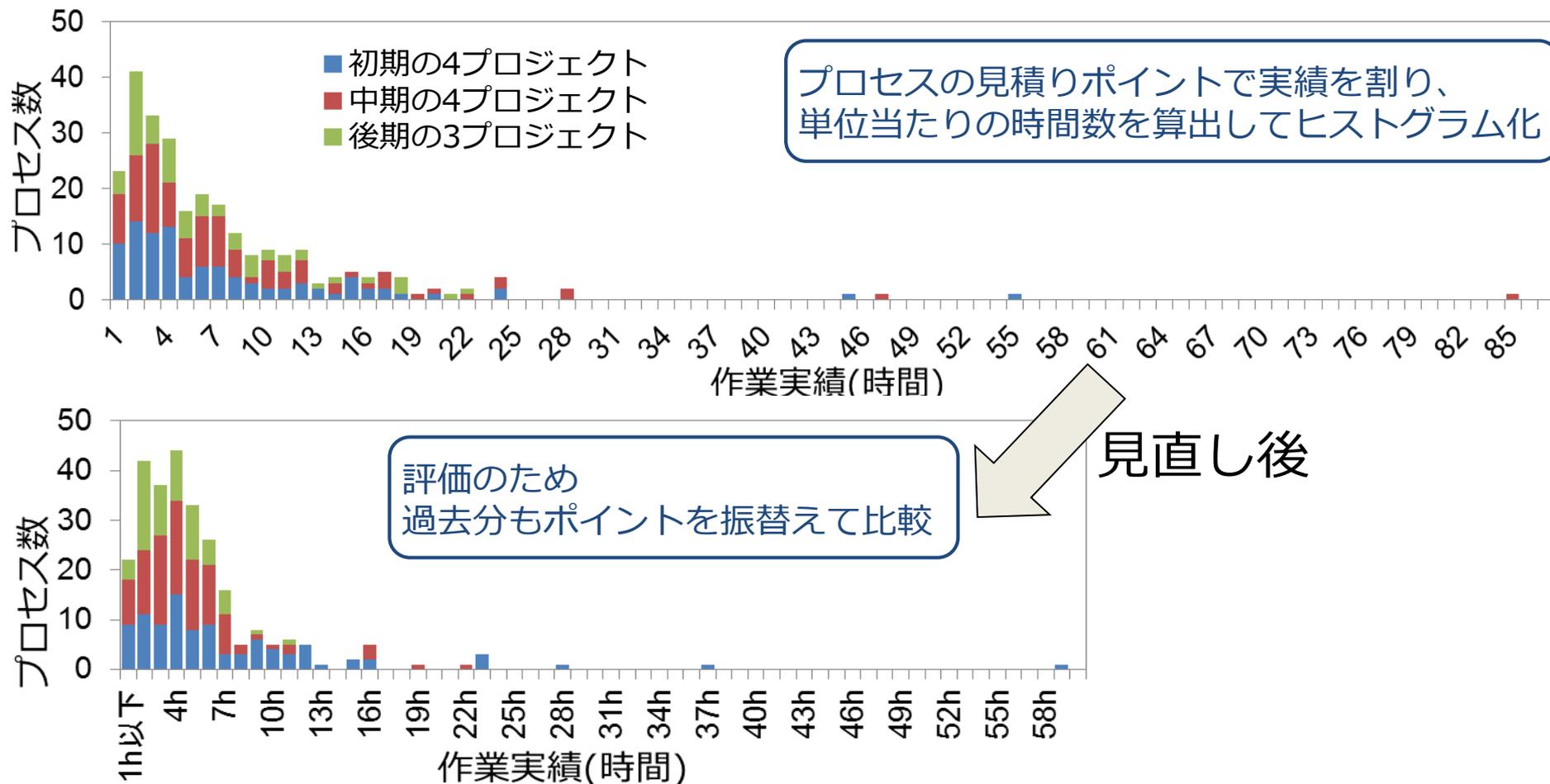
## ある時期の人的リソース計画の推移を示す



平準化した計画をもとに納期遅れなく開発を完了させられた

# 6. フィードバックの効果①

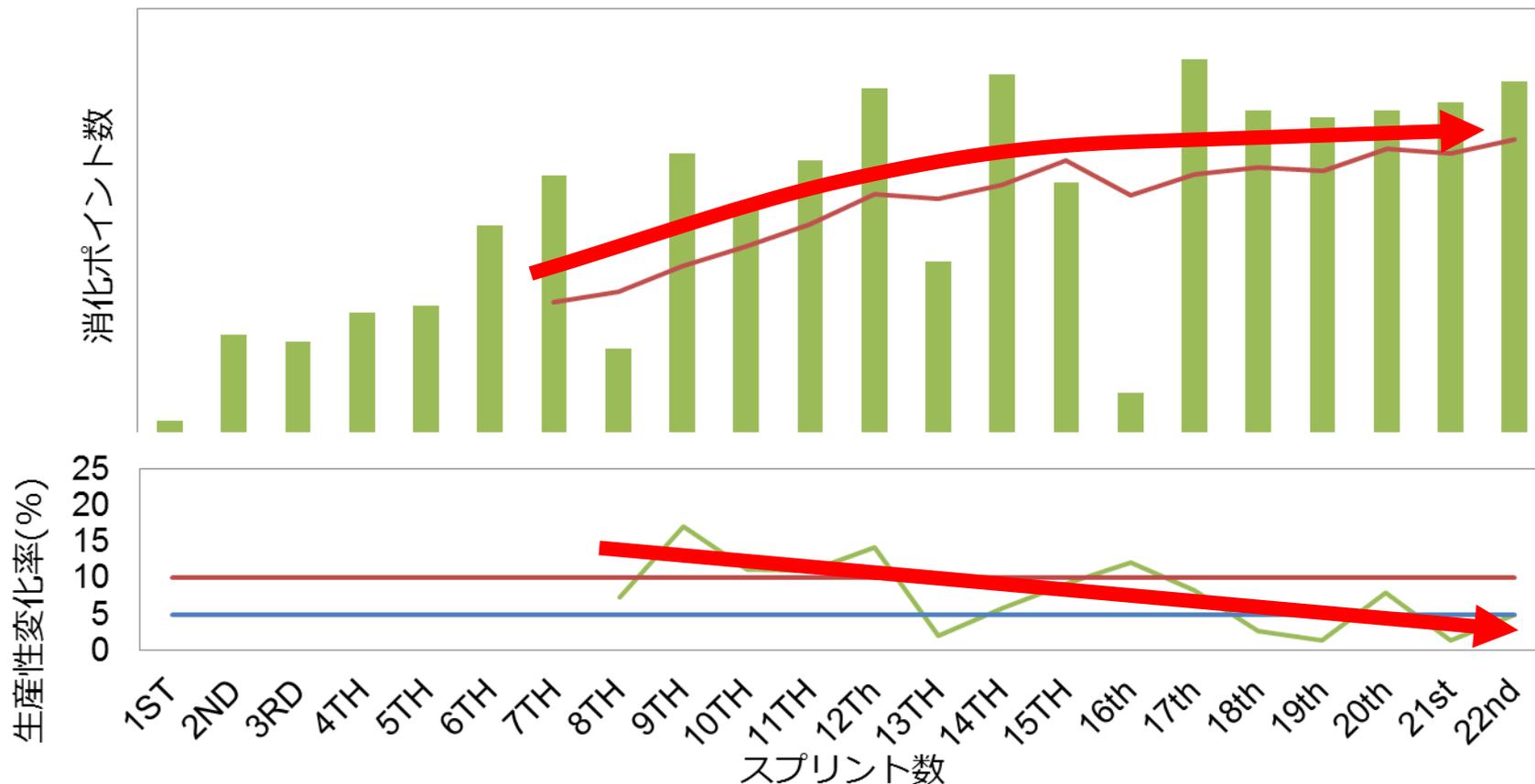
規模粒度の見直し⇒見積り正確さへの影響を評価した



規模当たりの実績のばらつきが軽減され管理し易くなる

## 6. フィードバックの効果②

規模(アイテムのポイント数)を見直すことによる、  
ベロシティ = 生産性の変化を再計算して評価した



開発成熟度が向上する様子など開発状況が見える化できる

## 7. 考察①

- 車両展開開発の反復性
  - 車両展開開発は車載通信や車両個性への適合が多い
  - 開発プロセスの反復性が高く、パターン化できた
- 生産性の予測可能性
  - 反復性が低いフィーチャ開発が入ると不安定となる
  - 反復性があればタイムボックス管理が有効に働く
- 開発量の予測可能性
  - 成熟度を考慮することで予測可能性向上の余地あり

## 7. 考察②

- 管理性

- 生産性と開発量は十分な正確さを備えた
- 確かな計画は確かな開発の礎となる

- 見積り粒度と規模のフィードバック

- 初期の見積りは正確さに欠ける
- 初期見積りは次回以降も参考にされる傾向にある
- 適切なフィードバックが必要不可欠

- 成熟度の測定

- 生産性を定めることで成熟度が数値化された
- チームの開発状態を測れる副次効果が期待できる

## 8. まとめ

### ・ゴール

- 車両展開開発において生産性と開発量をより正確に見積り、人的リソース計画と管理を実現する

### ・アプローチ

- プロダクトライン開発の派生開発に存在する、開発を跨いで現れるプロセスの反復性に着目し、SCRUMを応用したリソース最適化手法を提案する

### ・適用

- 車載ソフトウェアの車両展開開発

### ・効果

- 管理性を向上させて人的リソース計画を実現できた
- 開発成熟度などの開発状態を数値化する可能性を得た

***DENSO***