

TOSHIBA

Leading Innovation >>>

管理&改善らくらくパッケージの 部門展開事例

～即効性もあり中長期的にも効果のある
プロセス改善の取り組み～

2012年10月11日
東芝セミコンダクター&ストレージ社
システムソフトウェアセンター
岸 利至

Rev1.0

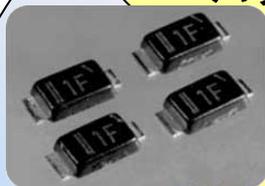
東芝セミコンダクター&ストレージ社のご紹介

製品の一例

東芝S&S社 製品群

ディスクリート半導体

汎用CMOSロジック/小信号デバイス/
パワーデバイス/オプトデバイス



メモリ

フラッシュメモリ/MCP(Multi
Chip Package)



ロジックLSI

マルチメディアSoC/カスタムSoC



ストレージプロダクツ

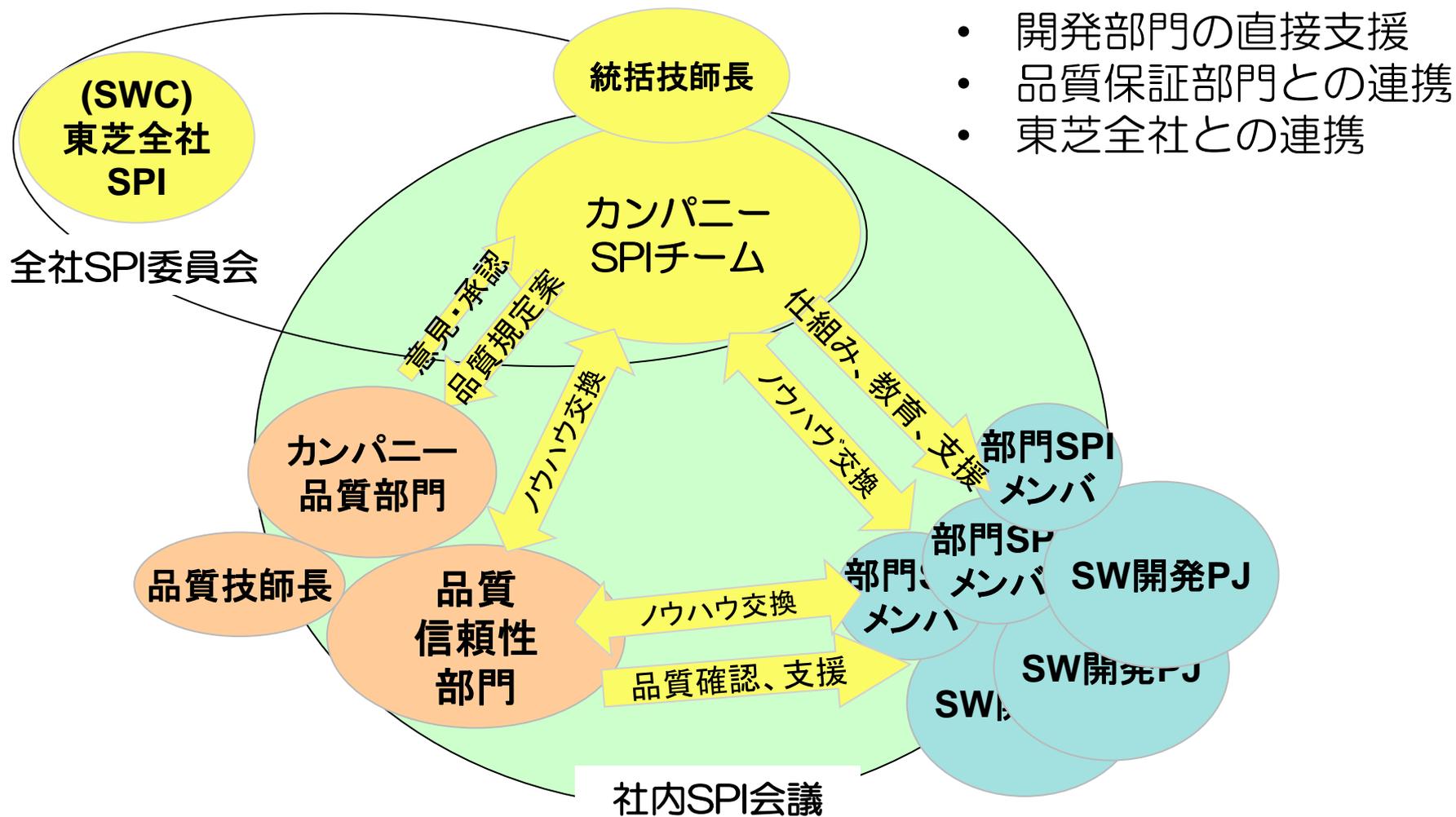
SSD/HDD



アナログ・イメージングIC

アナログIC/CMOSイメージセンサ
マイコン

カンパニー-SPIチームとソフトウェア関係組織関連図

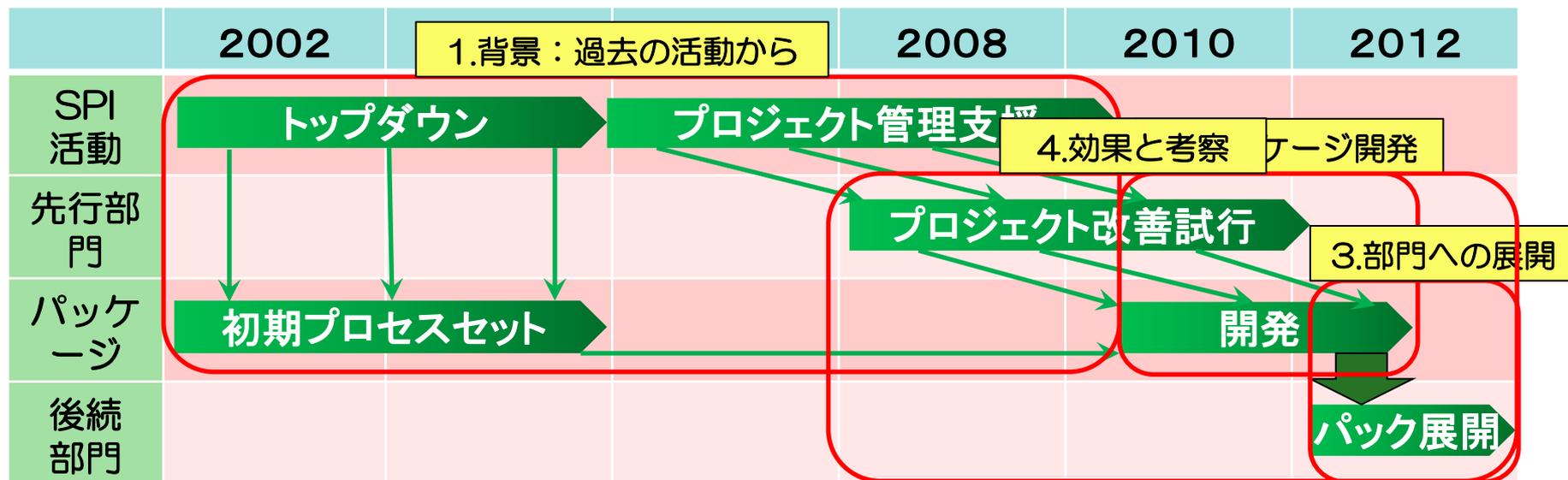


- 部門SPIの支援
- 開発部門の直接支援
- 品質保証部門との連携
- 東芝全社との連携

もくじ

「管理」 & 「改善」 軽量化パッケージの開発と部門展開

1. 背景：過去の活動から
2. パッケージ開発
3. 部門への展開
4. 効果と考察



1. 背景：過去の活動から

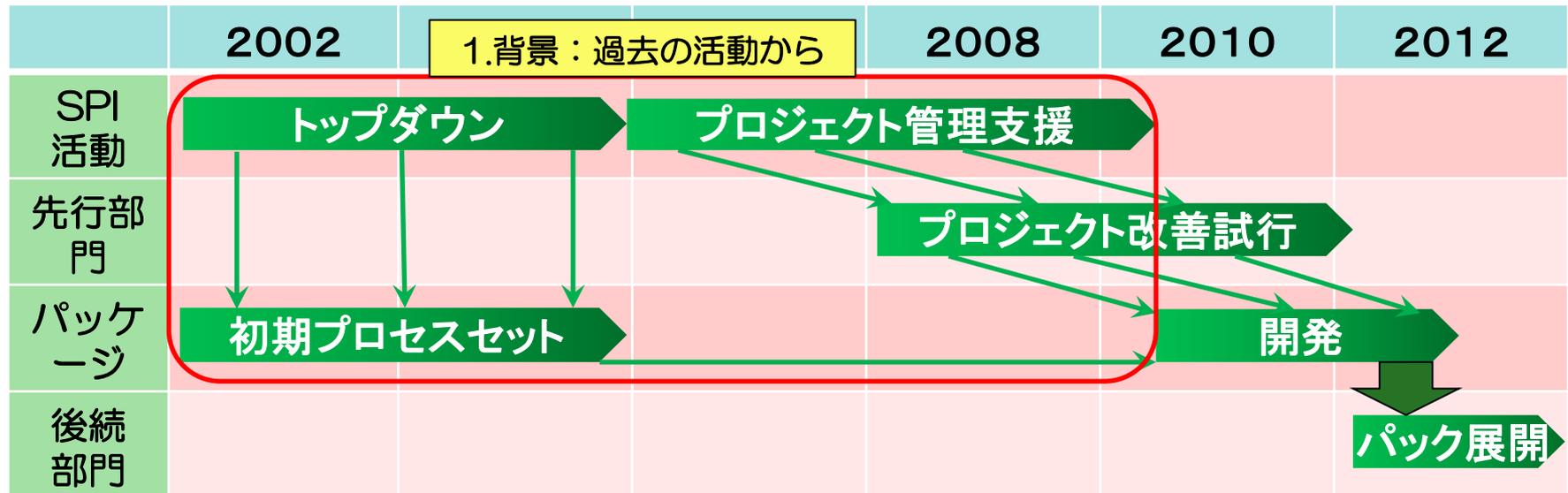
「管理」 & 「改善」 軽量化パッケージの開発と部門展開

1. 背景：過去の活動から

2. パッケージ開発

3. 部門への展開

4. 効果と考察



活動初期：トップダウン

2002年～2004年

トップダウンアプローチでプロセス展開



CMMでいうレベル3相当の仕組み構築をめざし、下記を作成して開発者に展開

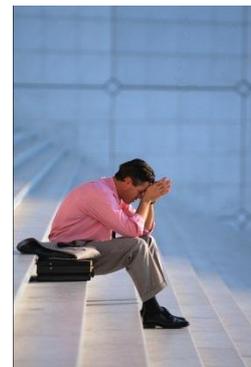
- SW-CMMレベル3相当の**部門規定**
- 開発のやり方をわかりやすくまとめた**ハンドブック、ガイド**
- 帳票作成のための**テンプレート、記載実例**
- **マイルストーン管理ツール**で文書の承認を行いながら開発を進める癖
- SW-CMMレベル3相当の各種**プロセス教育**

➡ 開発者のやらされ感、不満などはあったものの、これらは継続して活用&定着

活動中期：個別プロジェクト支援 2005年～2010年

景気の悪化、リーマンショック → 活動縮小

- ◆SPI人員削減
- ◆プロジェクト直接支援中心の作業



→ 直接支援を繰り返すうちに、大枠でプロジェクトの支援手順は、下記の4ステップから成っている事を発見。

1. PJ状況把握
2. 把握した状況を見える化
3. 状況と問題を共有
4. 問題に対する対策

➡ このパターンで支援をすると問題解決していく



何を見えるようにすると良いのか？

現実に終わると思われる日（完了見込み）が示されると、初めて本気で対策を検討し始める。

納期までに「何ができるのか」を考えるためには、「**できないこと**」がどのくらいなのかを全員で共有して、**目をそむけず**に対策を検討することができて、初めてプロジェクトは加速し始める。

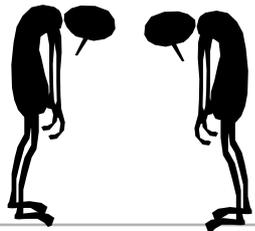


できない量を明確にして、
完了見込みを示し続ける

管理がなされていない

管理がなされている

状況が把握できず完了見込みが
示せない
QCDが達成できるかわからない
「がんばれ」しか言えない



どのくらいできないかがわかる
目標としたQCDとのすりあわせ
ができる
具体的な対策が検討できる。



構想 & 試運用

構想：「できない量を明確にして、完了見込みを示し続ける」方法について検討

- redmineでバグの管理はしている。 → **作業タスクについても同様に管理**できないか
- **タスクの消化数を計測**する事で、完了見込みを示すことはできないか



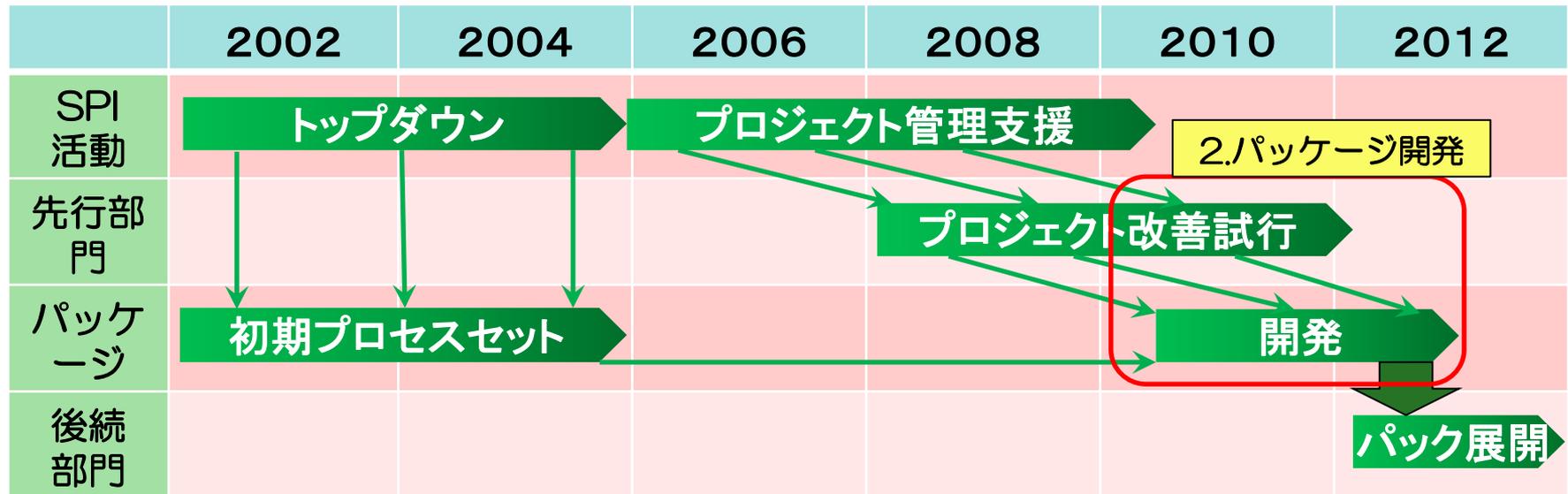
試運用

- 2004年までに作り上げた帳票類と教育を適用し、簡単な導入説明資料でキックオフを行うことから始め、構想を**実プロジェクトに試運用**を行い、試行錯誤を繰り返しながら、パッケージの原型を作っていた。

2. パッケージ開発

「管理」 & 「改善」 軽量化パッケージの開発と部門展開

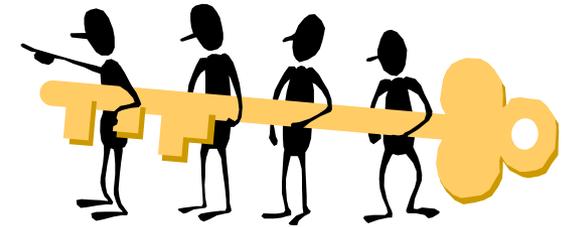
1. 背景：過去の活動から
2. パッケージ開発
3. 部門への展開
4. 効果と考察



パッケージ構成検討

- 構想のパッケージ化を検討
- 作りこみの活動に人をアサイン

3本柱



軽量化パック

仕組み

プロセス、管理手法、ツール

教育

PJ管理教育、プロセス教育

サポート

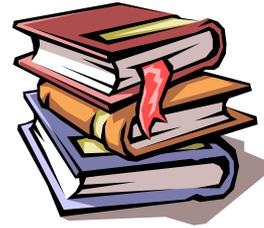
SPI支援、PJ管理支援

見込みを
示し続ける
プロジェクト
マネジメント

仕組みとは：A) 標準プロセス

仕組みとしては下記3つ

- A) 標準プロセス
- B) 開発管理手法
- C) ツール



A) 標準プロセス

- ✓ 規定、テンプレート、ガイド

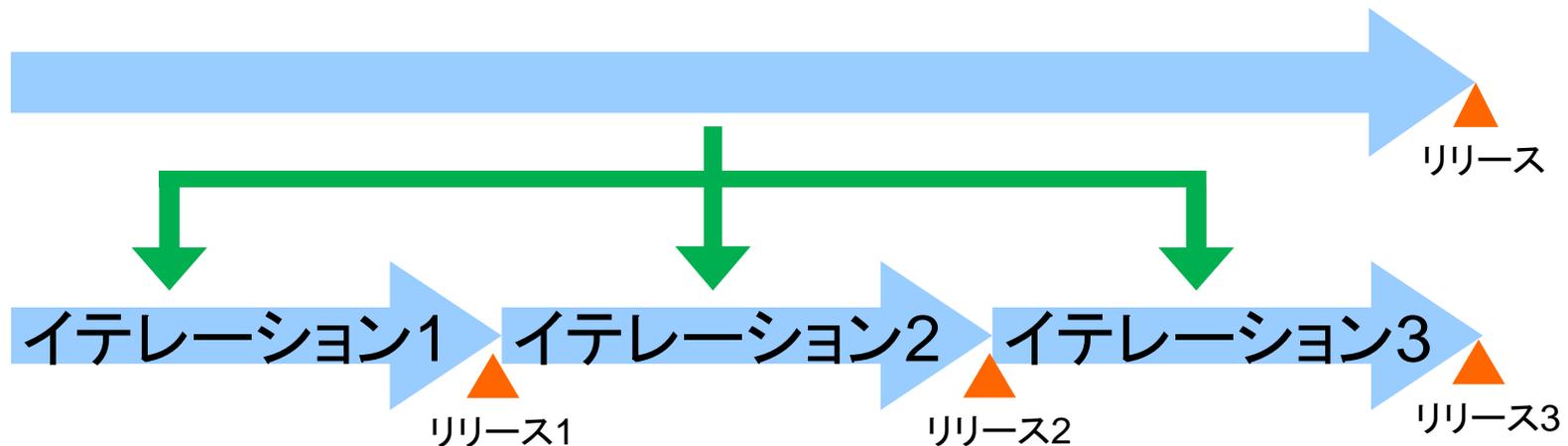
活動初期のころに作成したSW-CMMのレベル3相当のものが既にあり、記載実例もある。

➡ この部分については、特に**新たな活動は必要なし**

仕組みとは：B) 開発管理手法

B) 開発管理手法

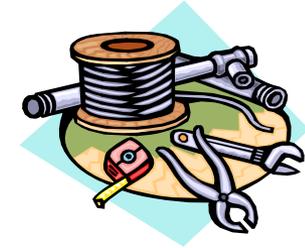
- ◆問題のあるプロジェクトの支援手順の4つのステップを短期にタイムリに実行しながら、**毎週のリズムを決めて開発を進めることができる手法を取り入れた。**
- ◆作業タスクをredmine上で管理し、**タスクの消化数で開発速度やできなかったタスクをコントロール**するような管理の手法を取り入れた。



仕組みとは：C) ツール

C) ツール

- タスク管理ツール：redmine
- 構成管理ツール：subversion
- 可視化ツール：自製ツール



Redmine、subversionについてはツールを触るというレベルではできていたので、あとは

- ✓ 適切なプロセス
- ✓ チームで決められたルールに沿って開発を進める
- ✓ 状況を見えるようにするための仕掛け

をすることがSPIチームとしての、キーとなる活動。

可視化ツール

Web表示

PJ情報見える化

リスク見える化

設計見える化

実装見える化

テスト見える化

バグ見える化

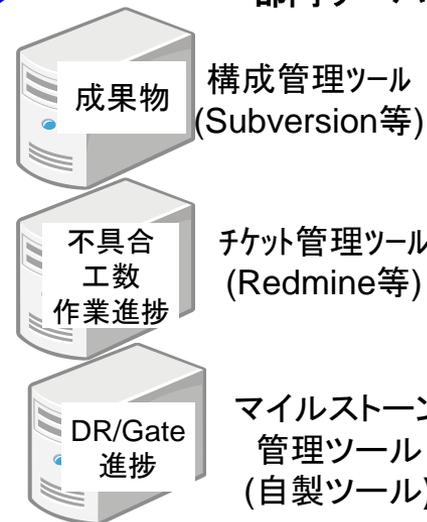
作業見える化

集約表示 & 誰でもいつでも気軽にチェック

新たなツールの導入コスト不要

既設サーバ・ツール利用

部門サーバ



データ収集
省力化

自動収集

共通DB
共通メトリクス

- ・過去の類似PJデータの参照が容易
- ・現在の複数PJデータの俯瞰、集計が容易

- ・データを自動収集
- ・状況をWeb上でグラフ表示
- ・定期的にステータスレポート

仕組みとは：C) ツール：分析改善ガイド (抜粋) 仕組み

No	表示イメージ	活用例	開発計画					
			設計	実装	テスト	出荷後	上級管理者/SQ	プロジェクトリーダー
			活用時期		活用対象者			
2.6.1	<p>バグ件数と改修件数の推移</p> <p>バグ見える化：バグ推移</p> <p>バグ発生状況、収束状況 バグ発生と修正の乖離</p>	<p>【バグ対応状況の確認】</p> <p>バグ件数とその改修推移を確認することで、バグ収束やバグ改修を視覚的に判断することができます。 DR/Gate日程予実などと照らし合わせて確認することによって、不具合収束と改修が、あるマイルストーンに間に合うかの判断にも利用できます。 → DR/Gate日程の確認は、「2.1.1 プロジェクト情報」で行うことができます。</p> <p>! 修のフォローについてはバグの滞留日数の多いものから対策すると効果的です。 → バグの滞留日数の確認は「2.6.4 バグ滞留日数ヒストグラム」で行うことができます。</p> <p>いわゆる、バグの滞留日数は差</p>						
2.6.4	<p>バグ滞留日数</p> <p>バグ見える化：滞留日数ヒストグラム</p> <p>この部分は強調して見せたい!</p> <p>PJ進捗状況 全体傾向</p>	<p>【チケット別滞留日数の確認】</p> <p>バグの滞留日数をチケット別に確認し、滞留日数の多いものから対策を行うことができます。 滞留日数が大きいチケットに絞り込むことにより、チケットの優先度付けを行うことができます。 チケット別の滞留日数については、各チケットへ展開した「バグ滞留日数ヒストグラム」で確認することができます。</p> <p>滞留日数が大きいものから内容を確認して、不具合の修正を加速</p>						

グラフのどのあたりを見ればよいか

グラフの目的、分析方法、改善例などをわかりやすく説明

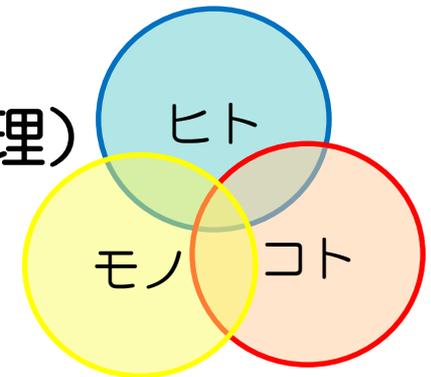
どの工程で活用するのか

教育とは

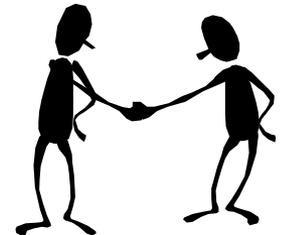
まずは教育により、全員の意識共有、知識共有、基盤技術の底上げ！

SPIチームが既に提供しているプロセス教育の中から、最も基本的な3種類の講座をパックに入れる必要がある

- ◆開発プロセス基礎教育（基本）
- ◆プロジェクト管理関連教育（ヒトとコトの管理）
- ◆構成管理関連教育（モノの管理）



- ・今回追加で必要だったのはプロジェクト管理関連の教育
- ・PMPの資格を持っていてプロジェクト管理について詳しい人がSPIメンバとして参加



教育とは：プロジェクト管理教育体系

教育

	初級者	中級者	上級者	管理者	
	PJL経験:0~1回 管理人数:~3人程度	PJL経験:2回以上 管理人数:~10人程度	PJL経験:複数回 管理人数:数十人規模	GPM、SM、... 管理人数:数十人規模	プロジェクト管理の...
導入	導入(e-learning) 製品開発とプロジェクト管理				必要性の再認識
基礎	一般教養(e-learning) 管理技術基礎 1~7章				基礎概念の共有
入門	実践入門教育(座学) 初級者のためのプロジェクト管理				勘所を知る
実践	実践詳細教育(e-learning) 計画と進捗管理実践				知識向上に応える
発展	グループ討議(座学) プロジェクト事例検討				事例と経験に学ぶ

教育とは：教育メニューと受講対象者

教育

	担当者 プロパ 駐在担当者	サブリーダー	全体リーダー 主任、マネージャ
プロセス管理	ソフトウェア開発のQMS基礎(プロセスの必要性と詳細) (e-Learning)		
		ソフトウェア開発のメトリクス基礎教育 (e-Learning)	
プロジェクト管理	プロジェクト管理(導入編)教育 (e-Learning)		
		初級者のプロジェクトマネジメント教育 (座学)	
	計画と進捗管理実践 (e-Learning + コンサル)		
		プロジェクト管理(基礎編)教育 (e-Learning)	
構成管理	構成管理教育 (座学)		
	構成管理ツール教育 (e-Learning + コンサル)		

パッケージに入れる教育メニューと受講対象者の目安を決めた

サポートとは：

サポートする時に必要なモノとして、

- らくらくパックメニュー
- らくらくパック展開手順説明資料
- kizashi（可視化ツール）構築と設定ガイド
- プロジェクト管理、構成管理、プロセス開発、教育実施時の各導入（キックオフ）資料。
- サポートに必要な各種ガイドとテンプレート。
 - （担当一覧表テンプレート、構成管理計画作成ガイド、チケットによるプロジェクト管理ガイド、教育受講管理テンプレート）

を作りこんだ。



➡ 俗人的な要素があるので、最低限の成果物を準備

軽量化パックメニュー

開発プロジェクトを運営するために**最低限必要**な環境／仕組みと育成のパッケージ

1. 準備／計画立案

- システム導入計画案
- 体制案
- 展開スケジュール案
- 予算計画
- 関係者への説明

2. 教育

- プロジェクト管理
- ツール操作
- プロセス
- 構成管理

3. 実践

- 規定、テンプレート、ガイド、ルール作成支援
- ツール構築と設定支援
- オンジョブ実践支援
(タスク管理、構成管理、プロジェクト管理者支援)
- 定期的なフィードバック

4. 結果分析／報告

- 実践結果の定量的分析、報告、評価

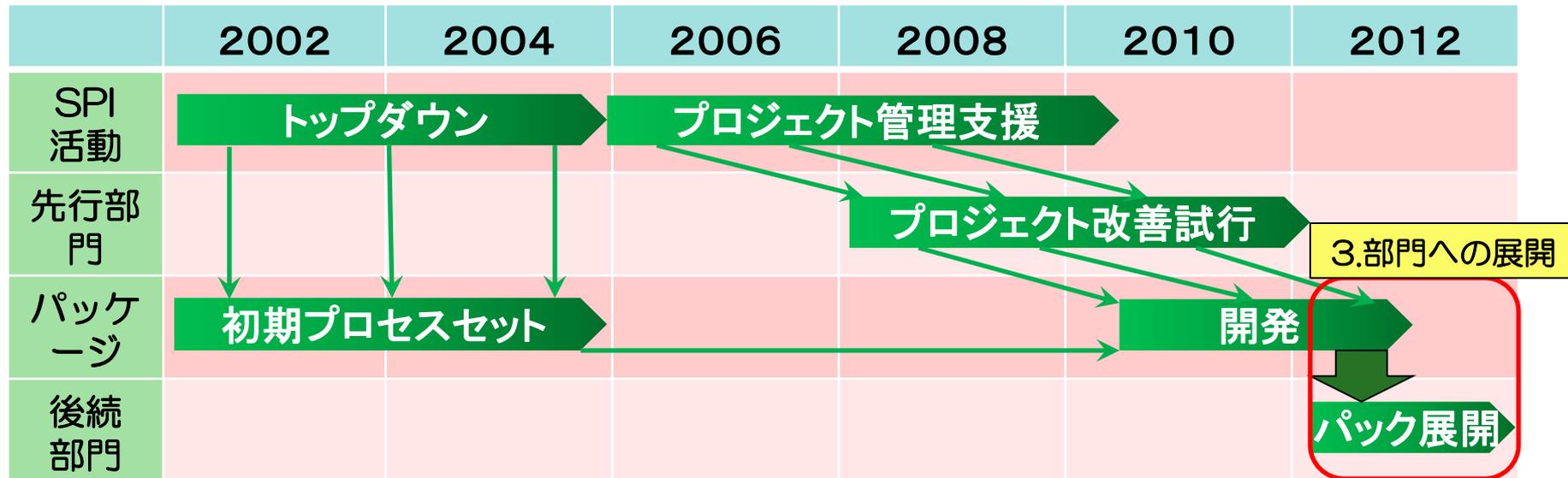
5. 改善検討／提案

- 評価を元に改善策検討／提案
- 次回の目標設定

3. 部門への展開

「管理」 & 「改善」 軽量化パッケージの開発と部門展開

1. 背景：過去の活動から
2. パッケージ開発
3. 部門への展開
4. 効果と考察



部門への展開：展開部門概要

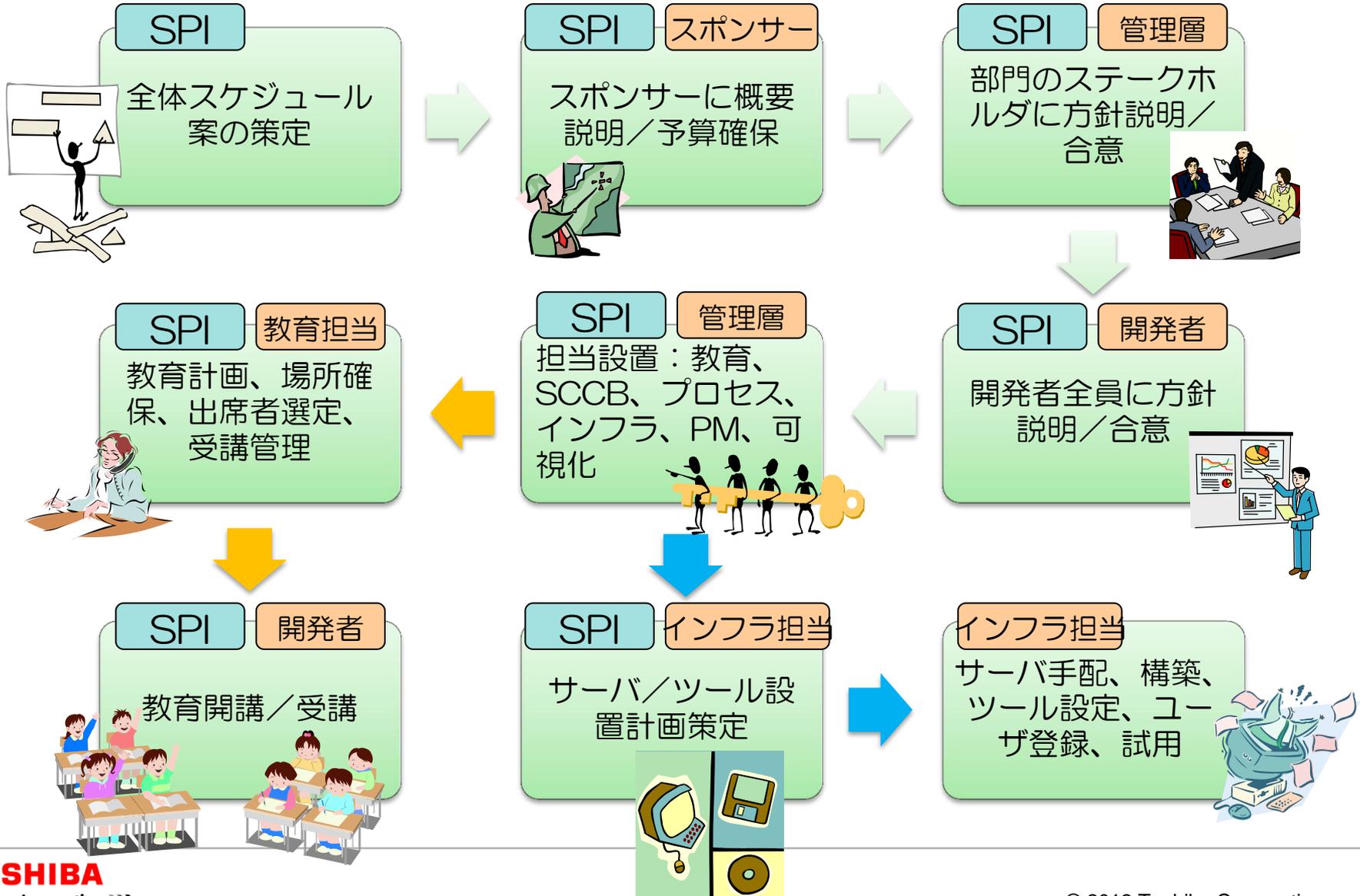
今回のパッケージを展開した開発部門の概要は下記

- ▶ 組み込み系のファームウェア開発部門
- ▶ 約50～70人規模×3部門 = 約180名
- ▶ C言語
- ▶ 組織の標準プロセスが整備されているとは限らない

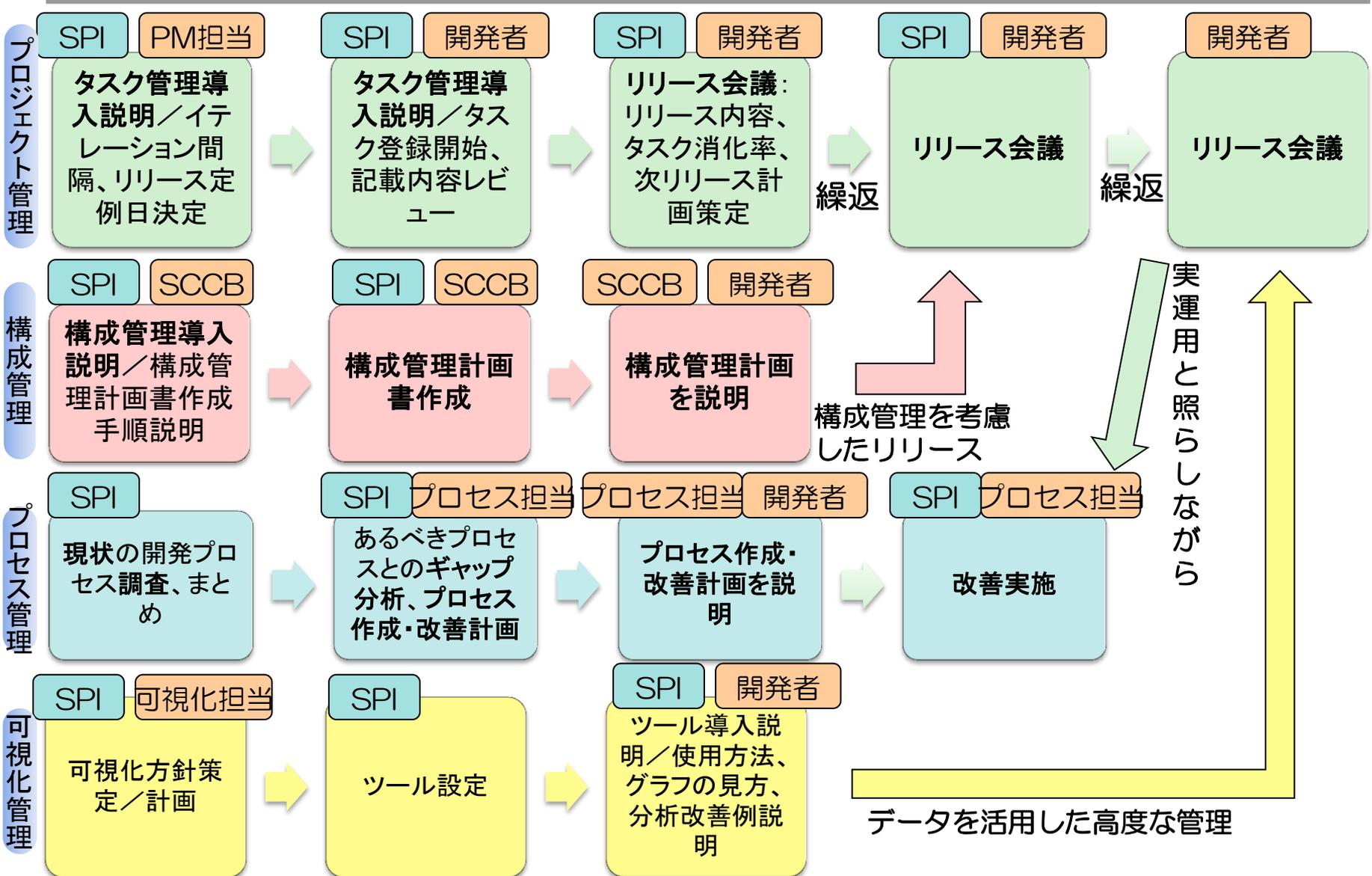


部門への展開：step0

Step0：立ち上げ準備、底上げ（約一か月でここまで実施）



部門への展開：step1



展開時の苦勞

きれいに展開したように書いているが、実際には苦勞

- ▶ プロセスは無くても、いままで俗人的にやっていた文化があり、改善後の姿と現状との違いを示しながら**開発者の視点**での説明が必要
- ▶ 変えたくないと思われる場合に、説得力やアプローチを変えて説明する**臨機応変さ**が必要
- ▶ **エンジニアリング**に入り込んだ話になった時に、理解し、会話ができる程度の**知識と実務経験**が必要

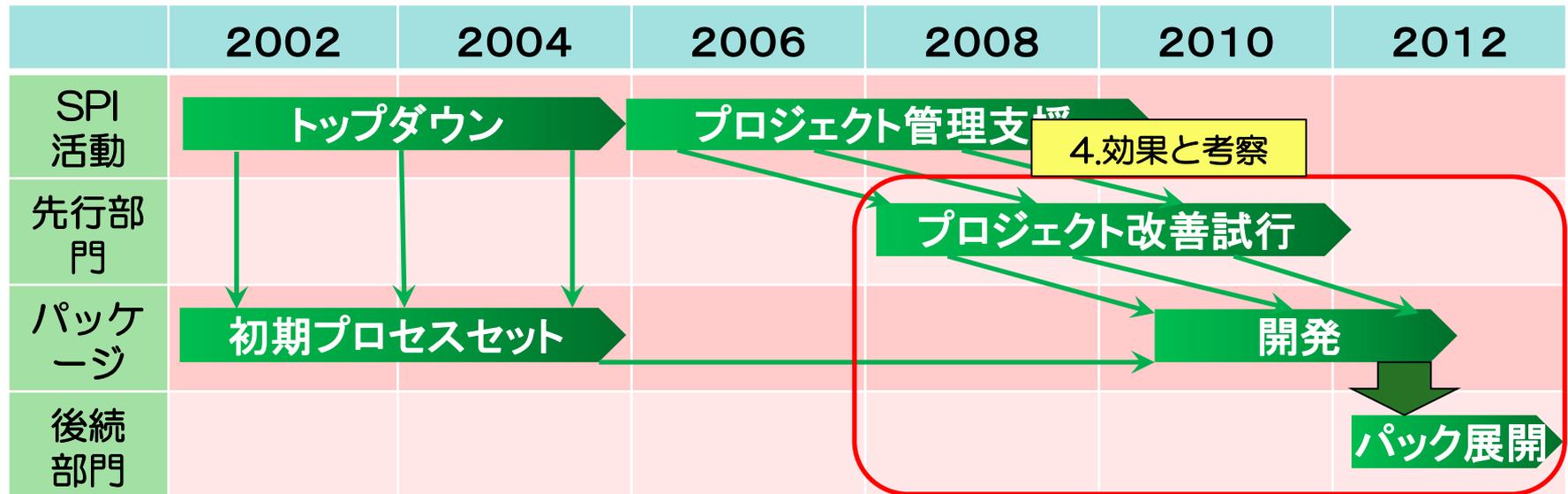
➡ 今後さらなるSPIチームの**スキルアップ**が必要。



4. 効果と考察

「管理」 & 「改善」 軽量化パッケージの開発と部門展開

1. 背景：過去の活動から
2. パッケージ開発
3. 部門への展開
4. 効果と考察



4. 効果と考察

このように、**上級管理者でも理解しやすい**プロジェクト管理という側面からのアプローチで、**最低限必要なプロセス**を抽出し、**開発者になじみやすい形**で展開を行うための導入パッケージを構築し、部門へ展開した。

即効性

パックが先に準備されていることで、物量の大量投入により一気に制覇するようなやり方で適用することができた。

部門側の意識も含めて一気に変えていく活動を、**4か月という短期間で実施**することができた。

中長期的効果

先行してパックの原型となるものを実践していたプロジェクトでは、安定してこの方法を実践しつづけており、**数年間納期遅れゼロ**を実現している。（顧客バグもほぼゼロ）

4. 効果と考察

今後

- ▶ このパッケージは、ソフトウェアに依存しない内容になっているため、今後はHW部門や海外ベンダへの展開も視野に入れて活動を検討していく。
- ▶ 可視化ツールで蓄積されたデータを元に、ビジネス上の勝ち負けを定量的に指標にする手法を検討していく。

TOSHIBA

Leading Innovation >>>

以下は参考

1. 改善活動基本方針

調査



現状(コト、モノ)の見える化

- プロセス管理
 - ヒアリング
- 構成管理
 - Subversionによる成果物作成状況の確認

➢ 成果物確認

動き(ヒト)の見える化

- プロジェクト管理
 - redmineによるタスク消化状況の確認

- 運用ルール策定、見せ方の工夫
(例: Kizashiによる自動監視など)

改善



Q

C

プロセスの最適化

- 部門共通項目の一元ルール化
- グループ固有ルールの明確化
- 仕組み上のリスク特定
- 技術スキルの特定

D

技術とルールの最適化

- フラグ管理の効率化
- 派生管理の効率化
- テスト効率向上
- 実現性のある計画の立案

ゴール



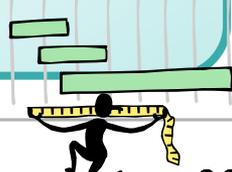
シナジーを発揮できる部門への転換

- 部門を横断した開発体制・技術
- 共通語・共通文化の発展

トレードオフがあるため
同時に連携しながら実施

TATがカギとなるビジネス状況への対応

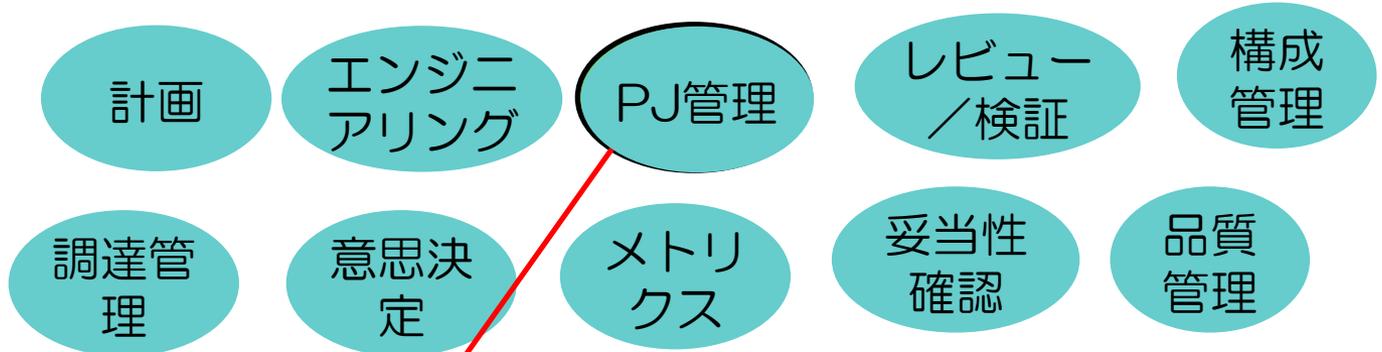
- タイムリーな開発
- 迅速かつ確実なリリース



目指すアプローチ



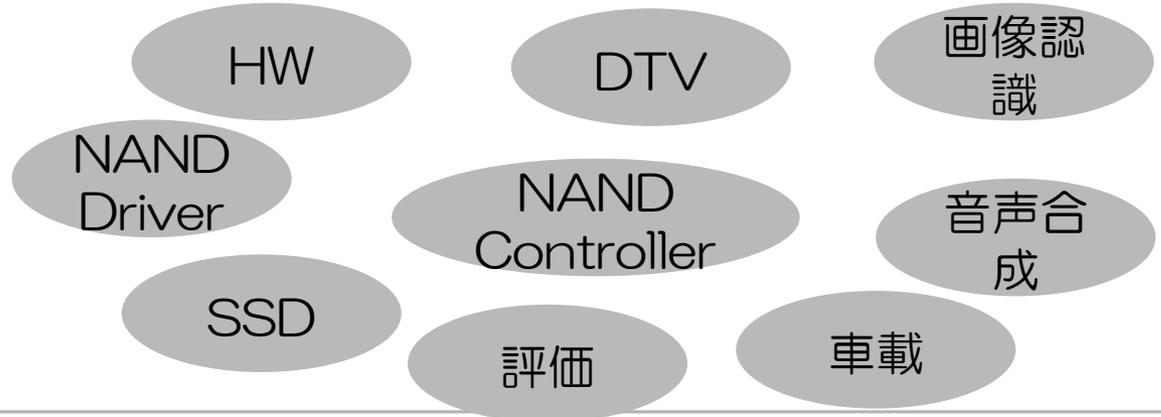
プロセス担当



共通言語



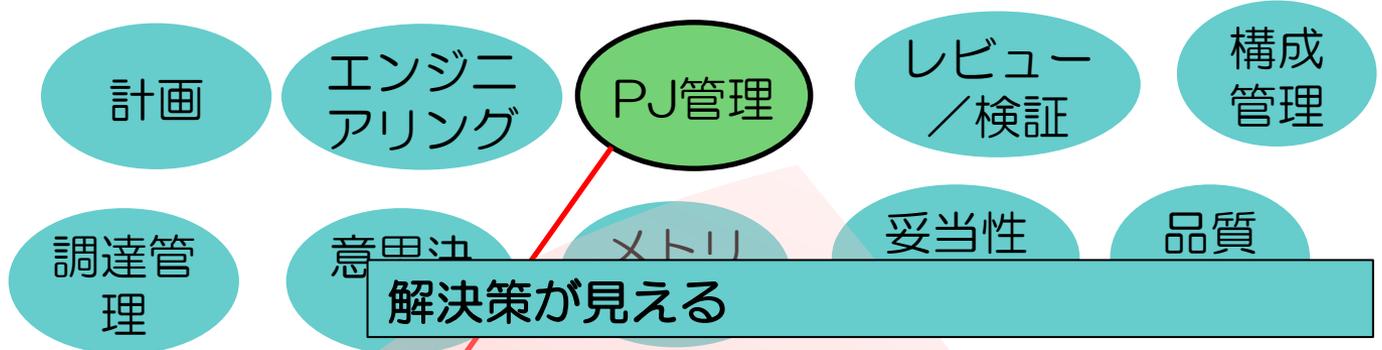
開発担当



目指すアプローチ



プロセス担当



共通言語



開発担当

