

トヨタ開発方式を利用した ソフト開発の継続改善活動

株式会社 **アドバンテスト**
品質保証本部 ソフトウェアQA部

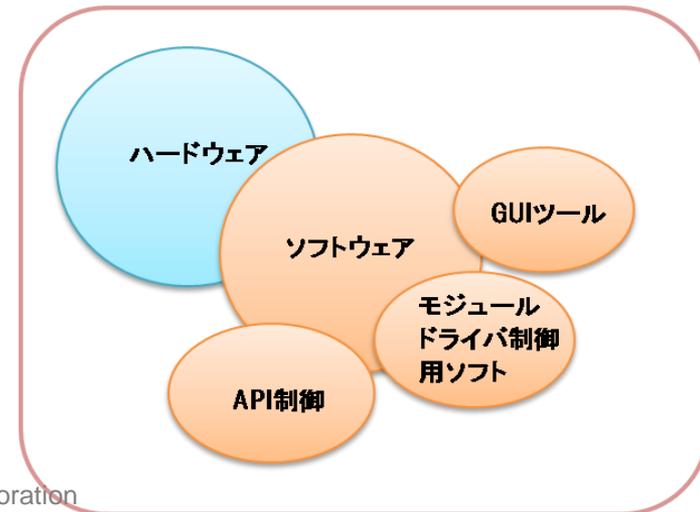
比嘉 定彦

アドバンテストの紹介

弊社は主に半導体試験装置の製造/販売を行っております



弊社の開発体制



自己紹介

勤務地: 株式会社アドバンテスト 群馬R&Dセンター

所属: 品質保証本部 ソフトウェアQA部

業務: ソフトウェアプロセス改善の推進

<履歴>

- ・1990年代からメトリクスを利用した欠陥の早期検出活動を展開
- ・2002年から、CMMの社内普及活動を実施。その後、EVM や CCPM を利用した開発管理プロセスの向上活動を展開
- ・2011年からトヨタ開発方式を利用した開発管理プロセスの向上活動を展開し現在に至る



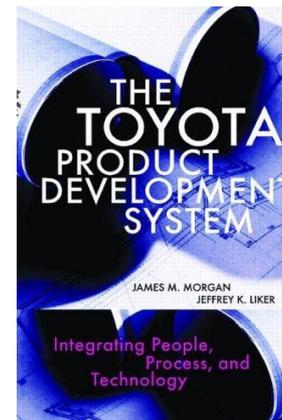
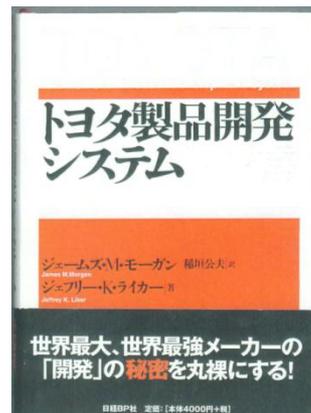
はじめに...

本発表の内容は、弊社でトヨタ開発方式*1について調査/検討し、ソフト開発現場で実践した内容と結果をまとめたものである。

*1) 次の文献にて、著者が名付けた開発方式。

”トヨタ製品開発システム” James M.Morgan、Jeffrey K.Liker 著

著者はトヨタ自動車の開発部門に出向き調査した結果を仕組と事例を関連付けて整理し、“トヨタ製品開発方式”と名づけた。トヨタ開発方式は開発部門で発展したものであり、トヨタ生産システムとは別のものである。



1. 背景
2. 改善前の状態
3. 改善前の状態をもたらした原因
4. 変更内容・対応策
5. 変更や対応策の実施内容
6. 変更・改善後の状態と効果
7. 改善活動の妥当性確認



1. 背景



1. 背景

弊社では、1990年代よりソフトウェアプロセス(規模/工数/品質)を定量化する活動を行い、定量化の概念(メトリクス)が普及してから久しい。

メトリクスが普及した結果、品質は改善されてきたが工期の改善は課題という状態が続いた。

2. 改善前の状態

3. 改善前の状態をもたらした原因



2. 改善前の状態

品質目標は達成した一方で開発工期は延長(分納など)。

3. 改善前の状態をもたらした原因

デザインレビューの徹底だけでは目標工期と目標品質の両立(同時達成)はできない。



4. 変更内容・対応策



4.変更内容・対応策（1）

我々は『目標工期と品質の両者を共に達成できる取り組みが必要』という状況の中で、トヨタ開発方式に出合った。

トヨタ開発方式では不良（欠陥）だけではなく、開発工程の中で発生する“**待ち**”や“**遅れ**”の状態も改善の対象としている。

トヨタ開発方式(原則3)にある「7つのムダ」と対応施策に着目し、社内で活動し易いようにソフト開発向けにカスタマイズして利用し、現状を改善することにした。

4.変更内容・対応策 (2)

7つのムダ	対応施策(トヨタ開発方式)	社内用にカスタマイズ
作り過ぎ	プロセスロジックと マイルストーン要求に基づく 平準化と同期化	部署間の同期化と平準化を導出する マイルストーンを設定し維持する
在庫		
手持ち	品質／技術データベース から知識を引き出す	課題ばらし*(検討プロセス)の質を 確保／向上する
加工	標準化したマイルストーンと 詳細計画	マイルストーンを使って開発をブレイクダウン し、総量ばらし*／課題ばらしを実施。 課題ばらしした結果をもとに作成した作業 計画は無駄を排除しチームの目標を網羅
動作		
運搬	製品開発価値の流れの改善	マイルストーン完了遅れなどを分析し、 開発の阻害要因を取り除く
不良	リーン製品開発への転換	プロセスを継続的に改善

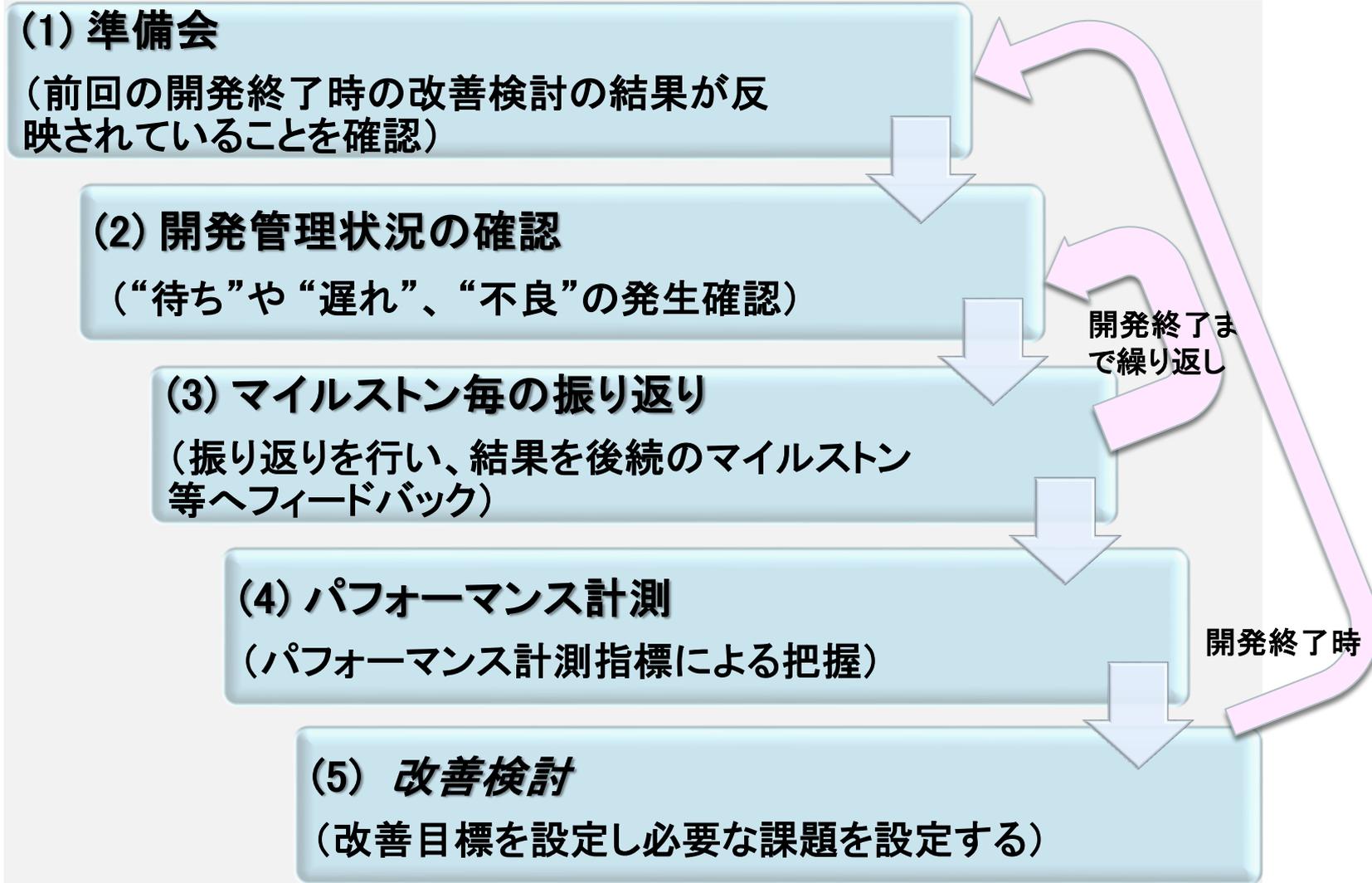
5 . 変更や対応策の実施内容



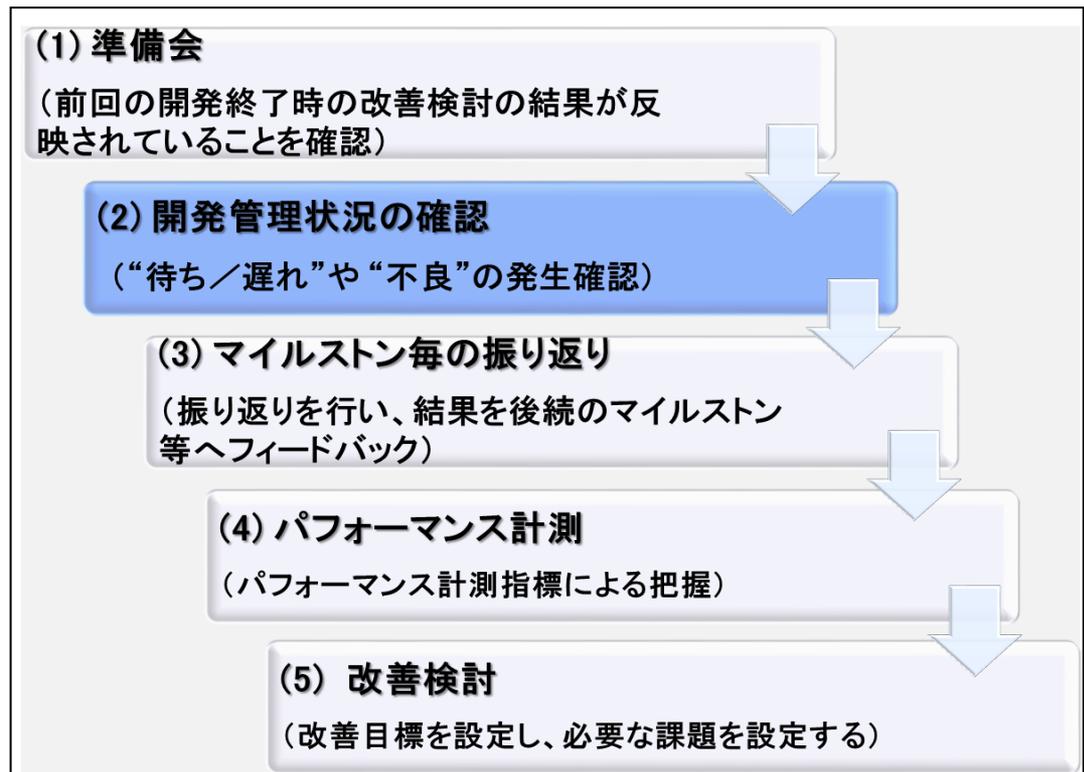
5.1 活動の手順



5.1 活動の手順



5.2 開発管理状況の確認



5.2 開発管理状況の確認

マイルストーン期日のタイミングでF2Fを開催し、開発リーダー(課長)へ
“待ち”や“遅れ”、“不良”の有無とその理由を確認。

例)

＜プロジェクトB のマイルストーン管理表(抜粋)＞

マイルストーン	期日(予定)	達成日	確認結果
M1.オフライン単体テストが完了しオンライン単体テスト開始	2012/10/19	2012/10/25	遅れ
M2.オンライン単体テストが完了しオンライン総合テスト開始	2012/11/02	2012/11/02	OK
M3.新規版対応のため、* * へオフライン環境を提供する	2012/11/06	2012/11/20	遅れ
M4.従来版で動作実績のあるデバイス実測評価が開始できる	2012/12/07	2012/12/27	待ち

M1ではリソースの不足による遅れが発生。

→“遅れ”

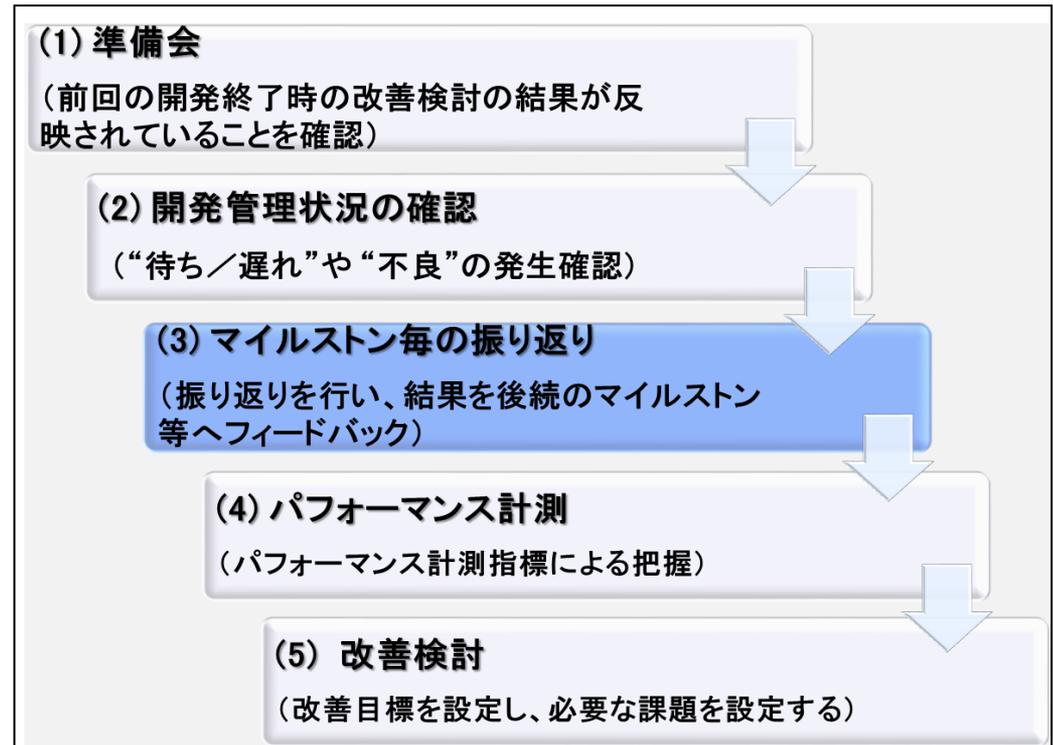
M3では課題ばらしの不足による遅れが発生。

→“遅れ”

M4ではハード遅れによる評価開始遅れが発生。

→“待ち”

5.3 マイルストーン毎の振り返り



5.3 マイルストーン毎の振り返り

終了遅延等が発生したマイルストーンについて“待ち”や“遅れ”、“不良”の内容を確認し、カスタマイズ内容に基づく視点により振り返りを実施。

(1) 振り返りの視点

待ち

- ・ 部署間のマイルストンの設定方法に問題は無いか

遅れ

- ・ マイルストーンと作業計画の整合性に問題は無いか

不良

- ・ 課題ばらしの質を確保する仕掛けに問題は無いか

5.3 マイルストーン毎の振り返り

(2) マイルストンの達成遅れ(“待ち”)についての例

振り返り(YWT*) → Y: やったこと、W: わかったこと、T: つぎにやること

Y

- ・ “動作実績のあるデバイス実測評価が開始できる” というマイルストーンが遅延した。

W

- ・ ハード開発から”キャプチャできる” という情報があったが、ソフトウェアテストに耐える状態では無かった。

T

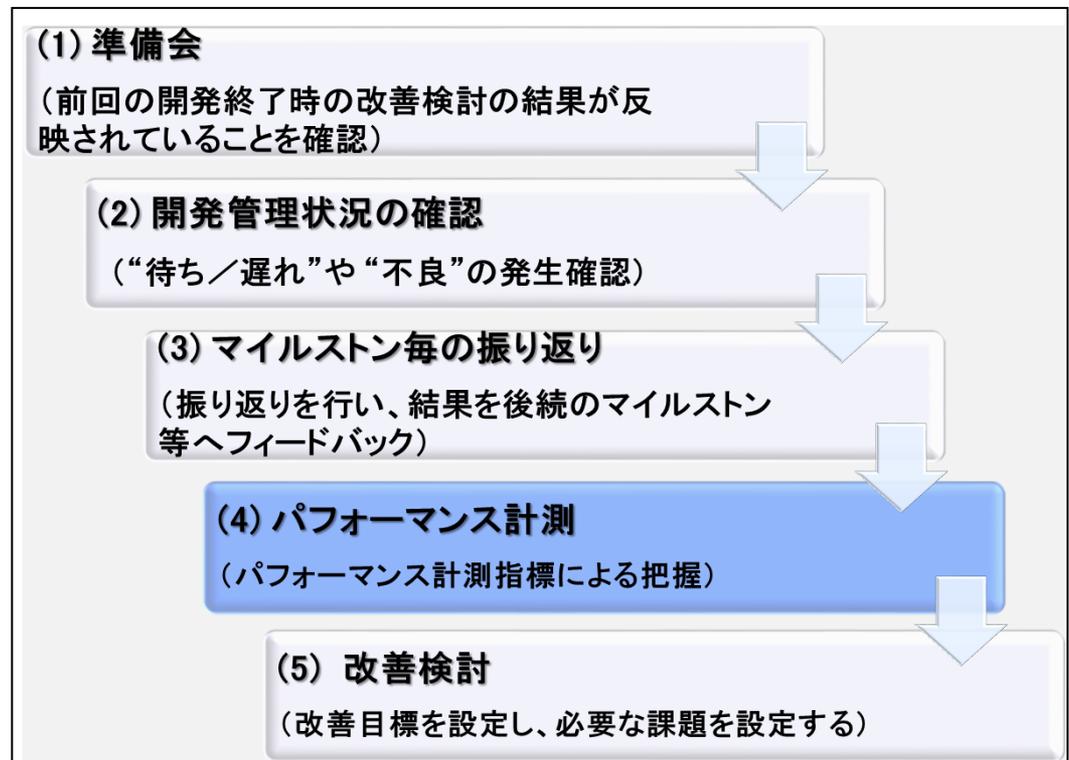
- ・ ソフト開発側の期待値を明確にした部署間マイルストーンを設定する。
例)
“動作実績のあるデバイス実測評価において、キャプチャが100%動作可能なハード状態で開始できる”

効果

- ・ 開発部門間の活動の同期化による改善

*) 日本能率協会コンサルティング 技術KI用語

5.4 パフォーマンス計測



5.4 パフォーマンス計測

(1) パフォーマンス計測方法

カスタマイズした施策の利用結果は、次のパフォーマンス改善指標で計測し、計測結果を基に今後の改善検討を行うようにした。

① “待ち”や“遅れ”を改善する指標 ⇒ マイルストーン遅延率（以降遅延と略す）

- ・ 他部署要因込みの遅延

$$= \left(\frac{\text{遅延が発生した全てのマイルストンの数}}{\text{マイルストンの総数}} \right) \times 100[\%]$$

- ・ 自部署要因のみの遅延

$$= \left(\frac{\text{課題ばらしを含む自部署要因で遅延が発生したマイルストンの数}}{\text{マイルストンの総数}} \right) \times 100[\%]$$

② “不良(欠陥)”を改善する指標 ⇒ 他部署から指摘された不具合

$$= \text{工程終了後に他部署から指摘された不具合件数}$$

補足) 通常の開発規模は1リリース当り10～100人月、期間は6ヵ月～1年

5.4 パフォーマンス計測

(2) パフォーマンス計測例

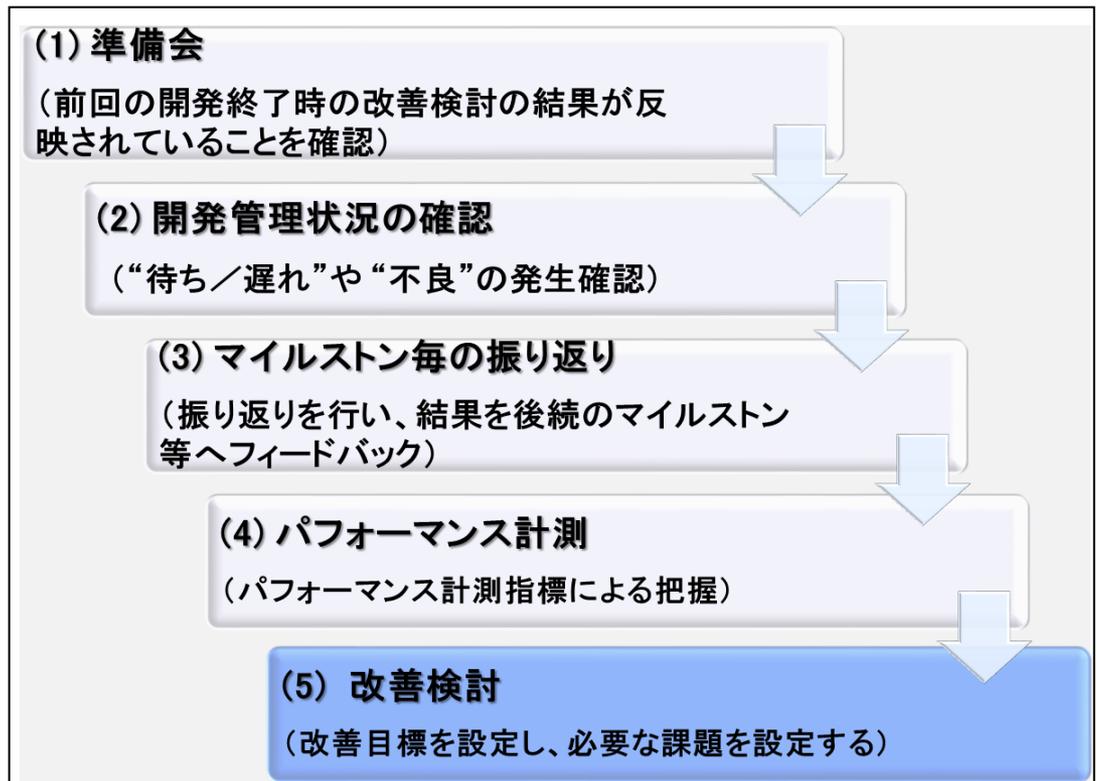
<プロジェクトB のマイルストーン管理表(抜粋)>

マイルストーン	期日(予定)	達成日	確認結果
M1.オフライン単体テストが完了しオンライン単体テスト開始	2012/10/19	2012/10/25	遅れ
M2.オンライン単体テストが完了しオンライン総合テスト開始	2012/11/02	2012/11/02	OK
M3.新規版対応のため、* *へオフライン環境を提供する	2012/11/06	2012/11/20	遅れ
M4.従来版で動作実績のあるデバイス実測評価が開始できる	2012/12/07	2012/12/27	待ち

計算例 (マイルストーン総数=13個)

- ① 他部署要因込みの遅延 = $3/13 = 23\%$
- ② 自部署要因のみの遅延 = $2/13 = 15\%$
- ③ 不具合 = 0 件

5.5 改善検討



5.5 改善検討

(1) 現状の把握と今後の改善目標設定

パフォーマンス計測結果を基に現状を把握し、今後の改善目標を設定する。

例) <プロジェクトA>

1) 現状の把握

- ・他部署要因込みの遅延 : $3/10 = 30\%$
- ・自部署要因のみの遅延 : $1/10 = 10\%$
- ・他部署から指摘された不具合 : 1件

意見交換結果



2) 今後の改善目標: 現状値を半減する。

- ・他部署要因込みの遅延 : 30→15%
- ・自部署要因のみの遅延 : 10→5%

5.5 改善検討

(2) 改善に必要な課題の設定

改善目標を達成するのに必要な課題を検討/設定する。

例)

＜マイルストーン遅延率を半減するための課題(工期の改善)＞

1) 外部要因 (影響を小さくする)

・ハード部署が問題(課題)へ対応するスケジュールを監視する。

2) 自部署要因 (リスクを低減する)

・開発環境や評価環境を整備するためのマイルストーンを設定する。

＜他部署から不具合を指摘されないようにするための課題(品質の改善)＞

3) 設計要因 (課題ばらしの質を向上する)

・不明確な部分の制御仕様について資料を作り勉強会を行う。

5.6 改善検討結果の活用



5.6 改善検討結果の活用

改善検討を行った結果を基に次期開発の計画へフィードバックし活用
例)

改善検討結果

ソフト開発側の期待値を明確にした部署
間マイルストーンを設定することにより、ハー
ド要因によるソフト評価トラブルを抑制する。

次の開発計画

新規モジュールの対応で、ハードの完成
度を評価するマイルストーンを設定した。

次の開発実績

ハードの完成度を評価するマイルストーンを
設定した結果、ハード要因によるソフト評
価トラブルは発生しなかった。

6 . 変更・改善後の状態と効果



6. 変更・改善後の状態と効果

活動した初回のリリース版と次に活動したリリース版（現在活動中）で取得したパフォーマンス値を比較することによって本活動の効果を確認する。

＜活動の初回と2回目のパフォーマンス値の比較結果＞

プロジェクトA	活動初回	活動2回目 (現在14個中10個終了)
他部署要因込みの遅延	30%(3/10)	0%(0/14)
自部署要因のみの遅延	10%(1/10)	0%(0/14)
他部署から指摘された不具合	1	TBD
プロジェクトB	活動初回	活動2回目 (現在12個中9個終了)
他部署要因込みの遅延	54%(7/13)	0%(0/12)
自部署要因のみの遅延	38%(5/13)	0%(0/12)
他部署から指摘された不具合	1	TBD

プロジェクトA,B共に現時点では遅延と不具合は発生していない。
現在迄の経過から活動2回目終了時の結果を推測すると、工期と品質の改善目標(前回実績の半減)を達成できる見込み。

7. 改善活動の妥当性確認



7. 改善活動の妥当性確認 (1)

パフォーマンスの向上実績が見込める状況になった要因を明確化するため各プロジェクトで実施した改善内容とその効果を確認した。

プロジェクトA

- 補正データ転送の実現方法についてハード部門と事前に検討会を開催
→課題ばらしの質の確保でき、手戻りの発生を防止できた
- 補正データ転送のオンライン評価を前倒しで(実装の期間中に)実施
→マイルストーンに対する作業計画を網羅でき、遅れを防止できた

プロジェクトB

- 実装以後の開発進捗は、機能別にマイルストーンを設定し管理
→“待ち”や“遅れ”の伝播の防止により、開発遅延を抑制できた
- オンライン評価の開始条件(ハードの完成度調査)をマイルストーン化
→関係部門間の依存関係を把握し同期化することで待ちを抑制できた

7. 改善活動の妥当性確認 (2)

<改善されたこと>

関係部門間の依存関係を把握する率が高くなり同期化が進んだ



“待ち”の改善

課題ばらしの質の確保が進んだ



“不良”の改善

マイルストーンに対する作業計画の網羅性と整合性が改善された



“遅れ”の改善

“待ち”や“遅れ”、“不良”の改善に必要な“T”(振返り結果)を出すようになった



工期と品質を共に改善する方策の導出



工期と品質を共に改善する効果を確認

改善目標を設定する際に、開発リーダーから次の意見が出された。
『ハード開発部署との依存関係を基にマイルストーンを設定しても、ハード開発が大きく遅れた場合は同期化を維持することが難しい』
ハードの開発リスクが大きな場合に必要となるマイルストーンの細分化や同期化等の対策を検討する必要がある。

ご清聴ありがとうございました。

参考文献

- [1] James M.Morgan and Jeffrey K.Liker,
The TOYOTA Product Development System
トヨタ製品開発システム(翻訳), 日経BP社, 2007

- [2] 中村素子, 勝田博明: 技術者・エンジニアの知的生産性向上,
日本能率協会マネジメントセンター出版, 2009