開発の"待ち", "遅れ", "欠陥"を防ぐプロセス改善

～PM改善ナビを利用した"待ち", "遅れ", "欠陥"の改善活動～

株式会社アドバンテスト
品質保証本部　商品品質保証部

比嘉　定彦

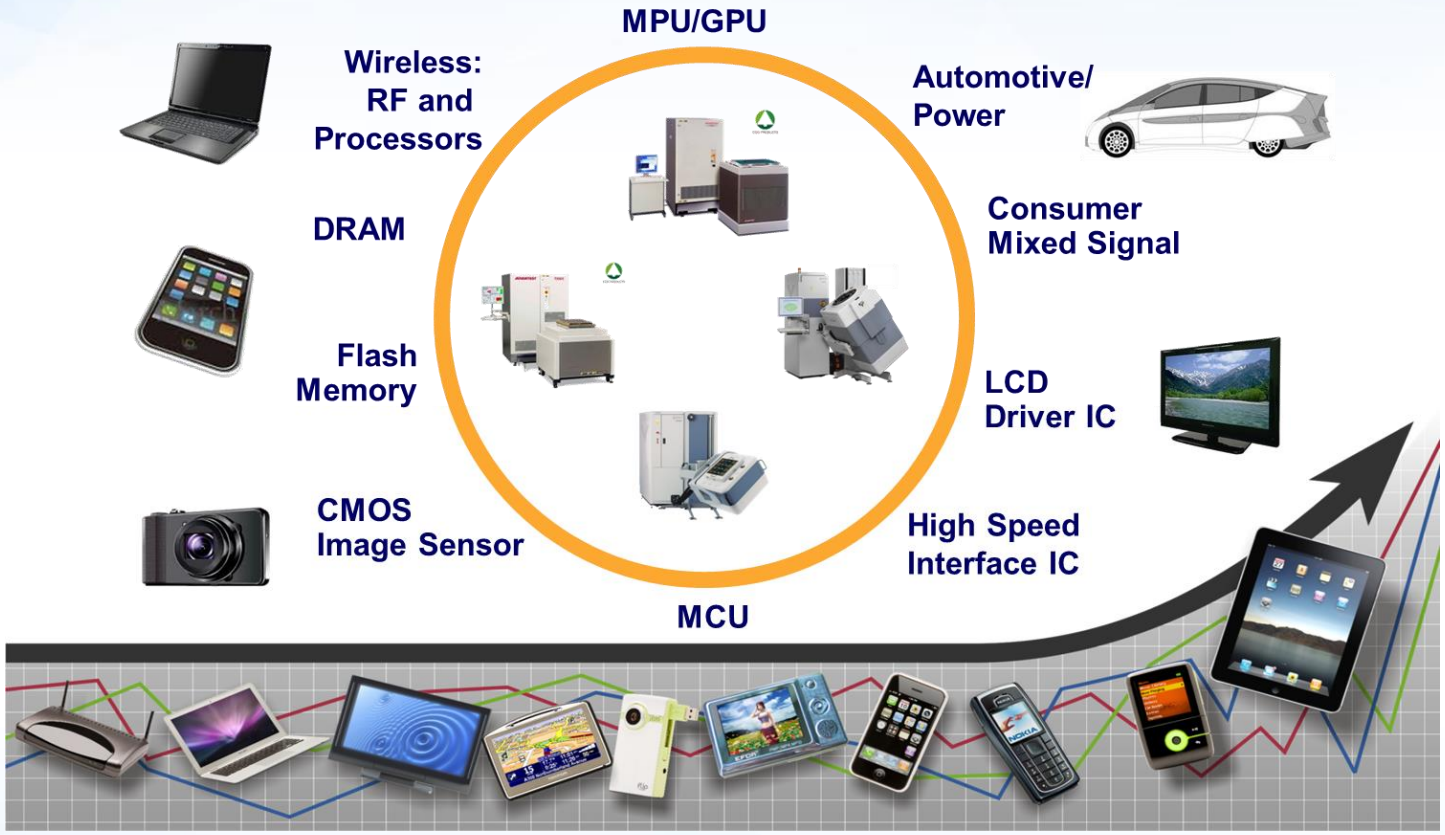
ADVANTEST

All Rights Reserved – Advantest Corporation

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

半導体テスト・システム: アプリケーション



→ 活動領域は半導体試験装置のソフトウェア開発

ADVANTEST

All Rights Reserved – Advantest Corporation

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
所属：品質保証本部 商品品質保証部

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[1]を利用した開発管理プロセスの向上活動を展開。

ADVANTEST

All Rights Reserved – Advantest Corporation

はじめに（聞き所）

◆開発工期と品質を同時に改善していく方法論として，トヨタ開発方式*[1]があります。
*）”トヨタ製品開発システム”James M.Morgan、Jeffrey K.Liker著

・開発現場ではしばしば「予定通りに進まない」や「不具合が多い」という問題に遭遇
↓
・問題を改善するため、必要な部分から順次 トヨタ開発方式をカスタマイズし利用
↓
・結果、トヨタ開発方式は 開発工期と品質を改善するための解を提示していることを確認
↓
・今日は、トヨタ開発方式を利用した開発工期と品質の改善活動 で得た経験を紹介

◆技術者の皆様が、プロセス改善について ヒントになることがあれば幸いです。

ADVANTEST

All Rights Reserved – Advantest Corporation

はじめに（続き）

本報告の内容を理解して頂くためのお願いです。

開発管理の考え方について一般論とのちがい[5]

一般論：開発品質は レビューとテストで欠陥を除去して確保

↓

本報告：レビュー指摘欠陥やテスト検出バグの本質は後から分かった課題

⇒ 課題ばらし*1の質向上の取り組みが開発品質向上の決め手

*1)技術的に曖昧な点や懸念事項を事前に明確にすること[5]

一般論：開発工期は，WBSの総量と総マンパワーを基に算定

↓

本報告：開発工期は，課題の大きさ（課題を解決して得られた付加価値の大きさ)と質（課題の新規性や複雑性)の累積に依存

⇒ 課題の重さを把握する方法（課題の見積方法)が 予定した工期で開発する決め手

プロジェクト管理の定石や効用を否定する意図はありません。

ADVANTEST

All Rights Reserved – ADVANTEST CORPORATION

目次

ADVANTEST

All Rights Reserved – ADVANTEST CORPORATION

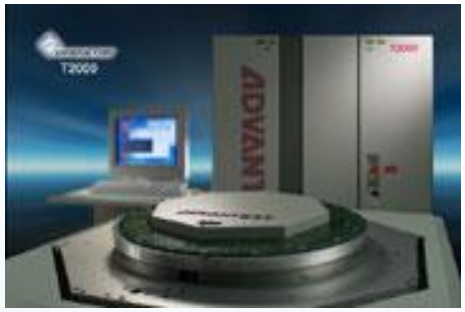
１．開発の"待ち", "遅れ", "欠陥"を防ぐプロセス改善

1-1. 課題

1-2. 改善の土台

1-3. 解決策

1-4. 改善事例



ADVANTEST

All Rights Reserved – ADVANTEST CORPORATION

1-1．課題

開発現場では，しばしば次のような問題に遭遇

- 実装後に不具合が多発（開発終盤に起きると品質/納期の両面に悪影響）
- 予定した日程で機能を実現できず（特に新規性が高い機能の場合）
- 提供されたものが使用要件を満たさず作業着手に遅れ

トヨタ開発方式[1]では 3つとも開発の流れの阻害要因（＝改善対象）

1点目の問題は〝欠陥〟
2点目の問題は〝遅れ〟
3点目の問題は〝待ち〟

ADVANTEST

All Rights Reserved – Advantest Corporation

1-1．課題(続き)

課題ばらしの質と“欠陥”,“遅れ”との関係

⇒ 開発の流れを阻害する”欠陥”と”遅れ”は、深層では繋がっている

＜　良い　←　課題ばらしの質　→　悪い　＞

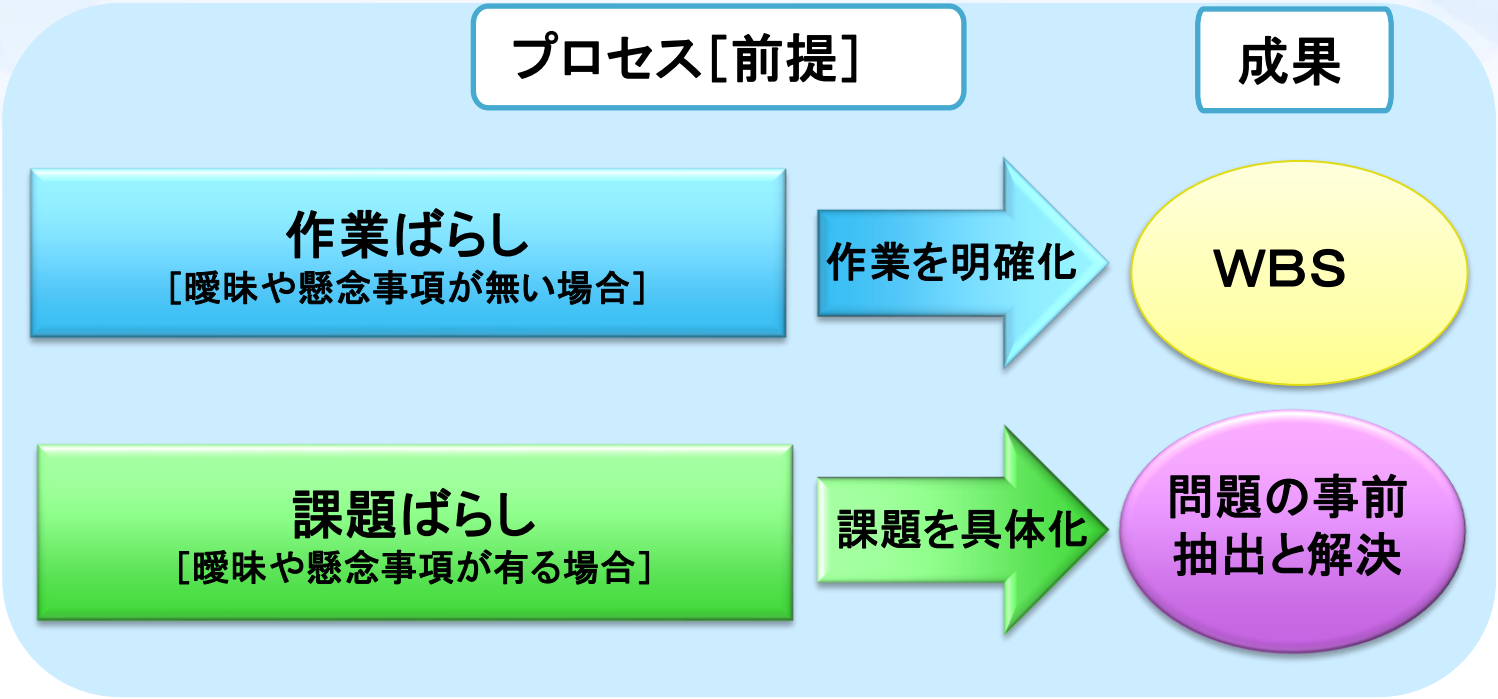
充分な課題ばらし　　不十分な課題ばらし（抽出し切れていない～実施漏れ）

レビュー / テストの欠陥(手戻り)は無し	課題ばらし後に新たな課題が見つかる	作業終了後に必要だった課題が見つかる
↓	↓	↓
遅れも不具合も無し	遅れ現象	不具合現象

All Rights Reserved – Advantest Corporation

1-2.プロセス改善の土台（課題ばらしの十分性）

課題ばらしの十分性を確保し、手戻りを最小化[5]



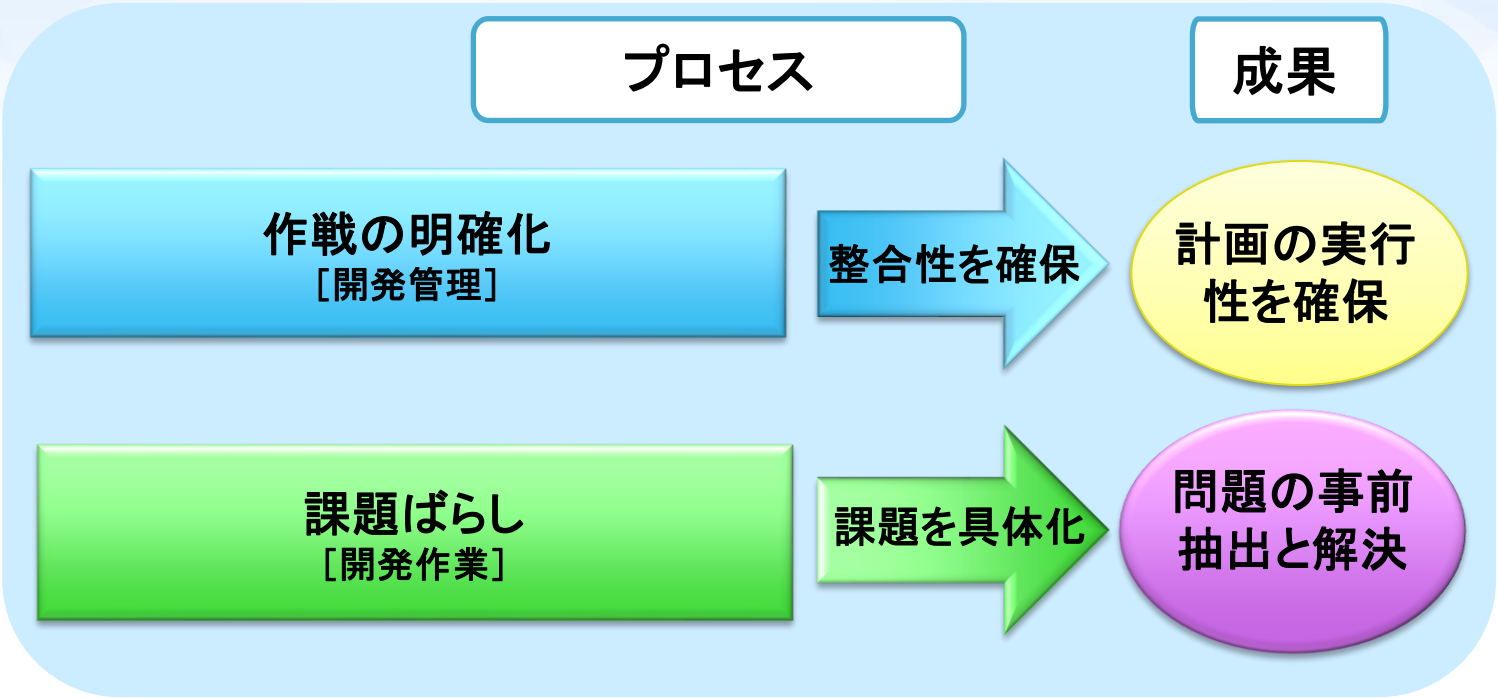
プロセス改善の土台(その1)⇒ 課題ばらしの十分性を確保

・技術的に曖昧な点や懸念事項は事前に明確にする
理由：懸念事項が有るのにWBSで対応した場合，後から分かる課題が発生

All Rights Reserved – ADVANTEST CORPORATION

1-2.プロセス改善の土台（課題と計画の整合性）

課題と計画の整合性を確保し、計画の実行性を確保[5]



プロセス改善の土台(その2) ⇒ 課題と計画の間の整合性を確保

・抽出した課題の解決をベースにした作戦（計画）を立てる
理由 ⇒ 課題の解決プランと一致しない場合，計画の実行性が確保されない

ADVANTEST

All Rights Reserved – ADVANTEST CORPORATION

課題と計画の整合性(補足)

抽出した課題に基き、課題の解決をベースにした開発プランを策定する

開発のスムーズな流れ(開発の実行性確保)は、
開発プランの質(課題と計画の整合性)が左右する。

よくある計画不備のケース:

チーム:

技術課題(チームレベル)の解決プラン(期待成果・時期を含む)がチーム内で共有されていない。

-> 課題解決目標(課題対応マイルストン)がチームリーダの頭の中だけに在り、メンバが認識していない(または認識が十分でない)ことがある

担当者 :

(最小粒度に分割した) 課題の解決プランが作業計画上で明確化されていない。

-> WBSが計画から抜けることは稀だが、技術課題への対応が計画に記述(登録)されないことはめずらしくない

ADVANTEST

All Rights Reserved - Advantest Corporation

1-3．解決策(トヨタ開発方式に基づく改善)

３つの改善原則[1]

原則１:”欠陥(手戻り)”の改善

⇒問題を前段階で解決することによってプロセス中の手戻りを無くす

原則２:”遅れ(アウトプットの遅延)”の改善

⇒仕事や作業が足並みを揃えて進むように調子を合わせる仕組を作る

原則３:”待ち(インプットの遅延)”の改善

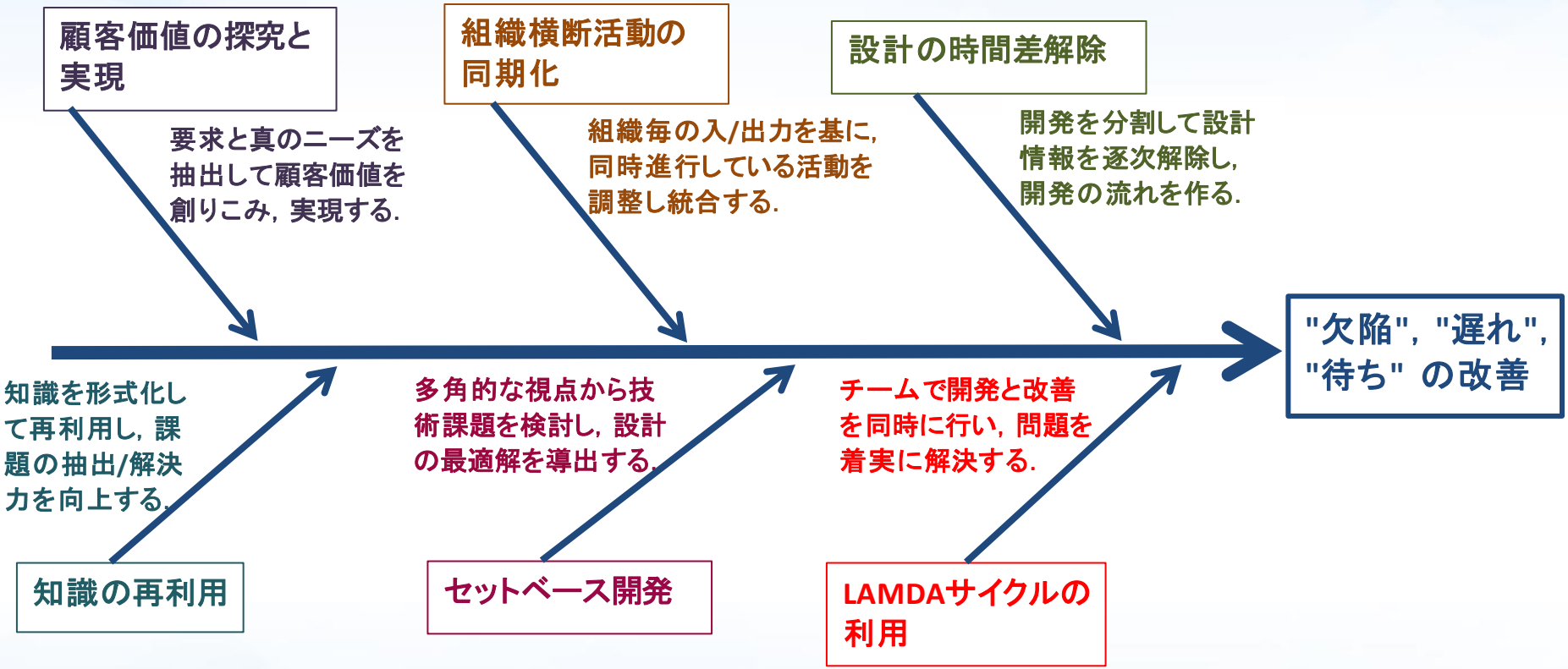
⇒横断的な活動を同期させ、必要な情報やものが必要な時にやり取りされるようにする

ADVANTEST

All Rights Reserved – Advantest Corporation

1-3．解決策(続き)

トヨタ開発方式の6つの行動原則[1][6]



All Rights Reserved – Advantest Corporation

行動原則の説明１

◇設計の時間差解除[1]

開発を分割して設計情報を逐次解除し，開発の流れを作る.

＜開発の分割例１（機能を段階的に網羅）＞

マイルストン（抜粋）	期日(予定)	区分
M1:新モジュールで従来機能 が動作する	2016/07/29	終了
M2:新モジュールの測定パターンが設定できる	2016/08/01	着手
M3:新モジュールの測定パターンが実行できる	2016/08/22	着手
M4:新モジュールの測定データが取得できる	2016/09/26	着手
M5:新モジュールの多点測定機能が使用できる	2016/11/18	終了

＜開発の分割例２（動作条件を段階的に網羅）＞

マイルストン（抜粋）	期日(予定)	区分
M1:システムBUS のPIOが動作する	2016/08/26	終了
M2:システムBUS の初期化処理が動作する	2016/08/29	着手
M3:システムBUS の割込み処理が動作する	2016/09/21	着手
M4:システムBUS のDMAが動作する	2016/11/30	終了
M5:システムBUS の機種依存動作を排除する	2016/12/28	終了

ADVANTEST

All Rights Reserved – Advantest Corporation

行動原則の説明２

◇セットベース開発[1]

多角的な視点から技術課題を検討し，設計の最適解を導出する.

⇒課題ばらしの質が一定以上に達した状態をセットベース化された状態とした

単一案を逐次改良する方法は多数の問題が発見され、開発期間を使い果たすまで続けられる。これに対しセットベース開発は、広範囲の設計選択肢を十分に検討し一定のペースで収束させる[1]。

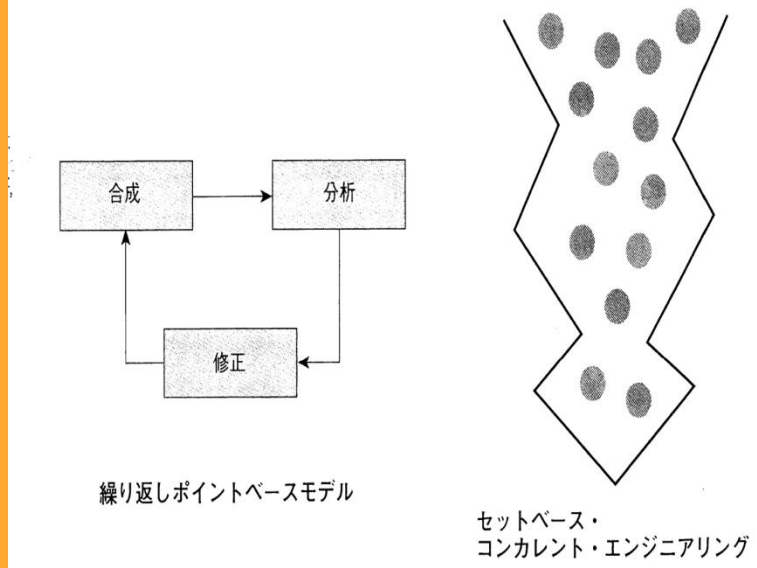


図 4-1 設計の2つのモデル

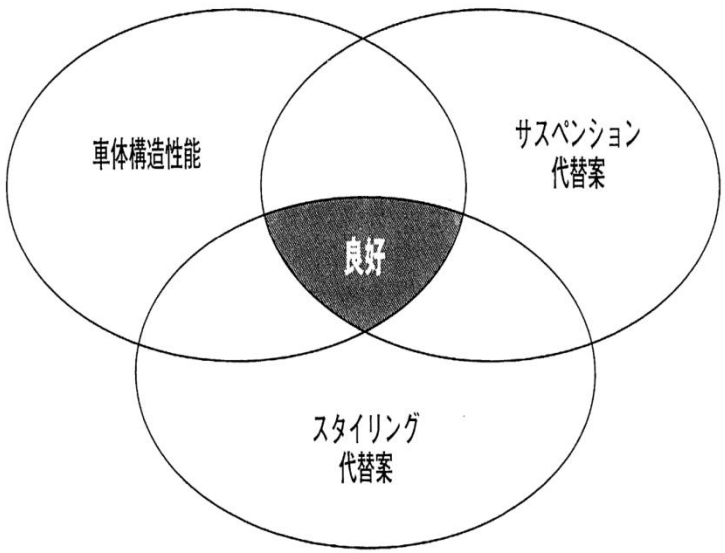


図 4-2 セットベースの視点――各機能別グループの良い部分の積集合を見つける

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

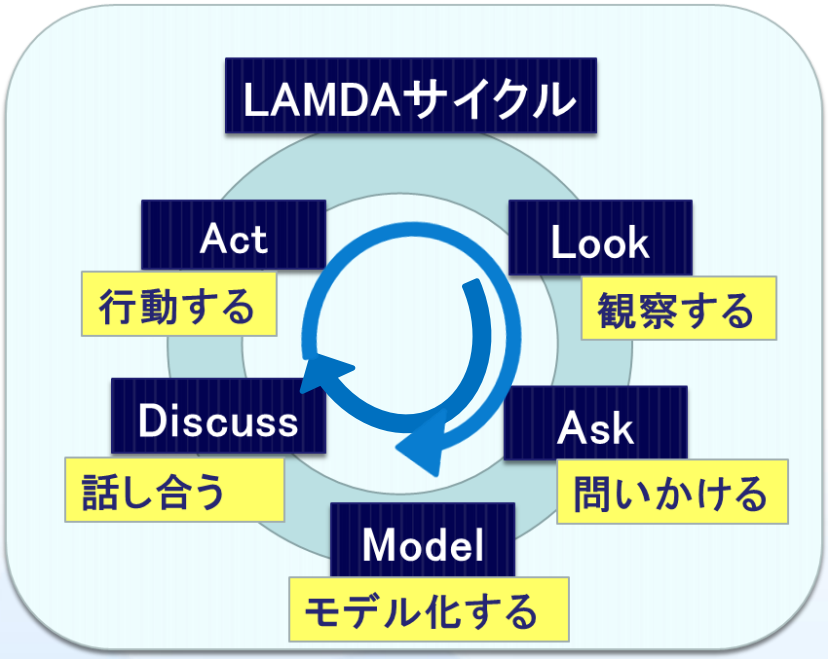
ADVANTEST

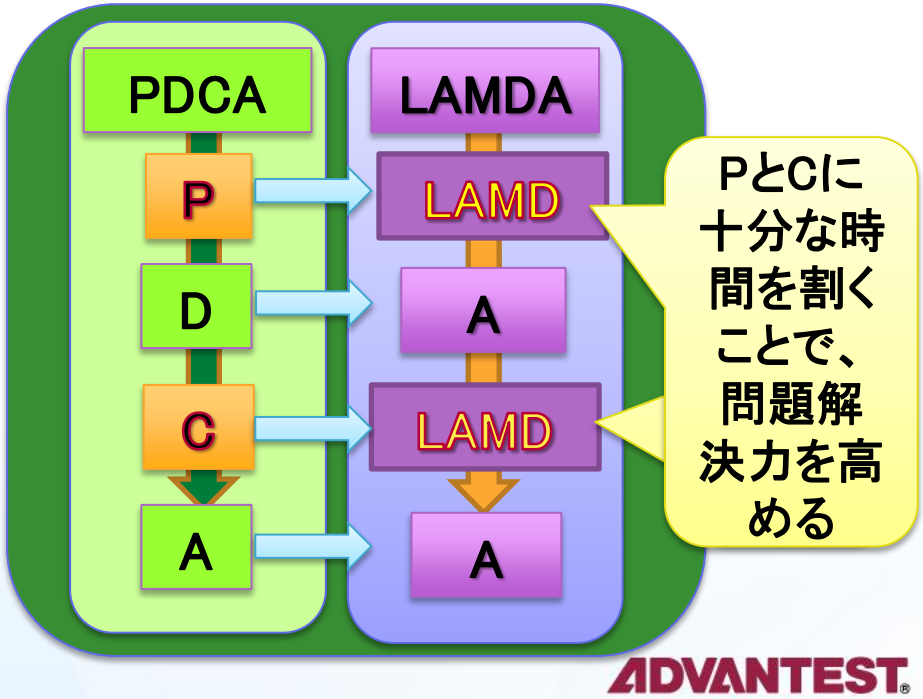
行動原則の説明３

◇LAMDAサイクル[6]

チームで開発と改善を同時に行い，問題を着実に解決する.

LAMDAサイクルは、リーン製品開発の基礎となる仕事の進め方で、
PDCAサイクルを開発設計向けに発展させたもの。
LAMDAサイクル2周分が PDCAサイクル1回分に相当する[6]。





ADVANTEST

All Rights Reserved – Advantest Corporation

1-3．改善事例（問題の分析）

- ”欠陥”の改善：
  問題内容：ある派生機種の開発で，開発終盤に不具合が多発
  分析結果：課題抽出漏れが多い所から不具合が多く発生

- ”遅れ”の改善：
  問題内容：ある新製品開発で，予定通りに終了しないテーマが多発
  分析結果：新機能に対し機能追加と同様の見積りを実施

- ”待ち”の改善：
  問題内容：ある新ハード対応の開発で，ソフト評価作業の着手が遅延
  分析結果：ハード/ソフト開発間で動作条件の考えに差異

ADVANTEST

All Rights Reserved – ADVANTEST CORPORATION

1-3．改善事例（問題の解決策）

◆ ”欠陥”の改善

課題ばらしの質を一定水準へ改善し，開発終盤の不具合を減らす．

※：弊社では，課題ばらしの質が一定以上に達した状態がセットベース化された状態

行動原則の選択 ⇒【セットベース開発】

◆ ”遅れ”の改善

見積りをチームで改善し，見積りと実際の課題の大きさの差を縮小．

※：チームで課題を抽出し（Look），大きさを問い掛けて（Ask），見積案を複数出し（Model），
見積りを収れんし（Discuss），計画へ反映（Act）

行動原則の選択 ⇒【LAMDAサイクルの利用】

◆ ”待ち”の改善

期待値を明確にしてハード開発と摺合せ，部署間マイルストンを設定．
テストを細分化し，ハード開発と並行してソフトの評価を逐次進める．

行動原則の選択 ⇒【組織横断活動の同期化】【設計の時間差解除】

ADVANTEST

All Rights Reserved – ADVANTEST CORPORATION

1-3．改善事例(”欠陥”の改善)

課題ばらしの質を一定レベル確保し，開発終盤での不具合修正を抑制

改善テーマ(要約)：
課題ばらしの効果を定量的に評価できる仕組みを作り、QCDを継続的に向上できるようにする。

	測定項目	改善前(中)/ Rev.A		改善後 / Rev.A+1
指標1.	課題ばらしの質 ＝事前にばらせた課題 /（事前にばらせた課題＋後から分かった課題）	一部のユニットで一定レベル(0.8*)に到達せず	⇒	全てのユニットが一定レベル(0.8*)をクリア
指標2.	パフォーマンス<品質> ＝ 最終マイルストンで行った不具合修正件数 / 開発工数[人月]	0.13 件/人月	⇒	0.02 件/人月

定量化： 課題ばらしの質を測定するためのデータ計測基準を設定(全チームで統一)し、あるRev. から計測を開始。

分析： 製品開発の終了まで計測した結果、最終マイルストンの不具合修正が特定ユニット(一定レベル未達)に集中。
├→ 振返り記録では約8割が課題ばらしの問題。”T”の実行により他ユニットと同一水準へ改善が見込める。
└→ 一定レベルをクリアした他ユニットでは、最終マイルストンでの不具合修正が殆ど無し。

対策： 不具合修正が多発した特定ユニットで課題ばらしの”T”(改善課題)を実施。 他のユニットは現状を維持。

結果： 次のレビジョンで、特定ユニットの課題ばらしの質が大幅に改善(最終マイルストンでの不具合修正はゼロ)。
製品全体では、最終マイルストンでの不具合修正 1件のみ へ大幅に品質が向上。

ADVANTEST

All Rights Reserved - Advantest Corporation

1-3．改善事例（”遅れ”の改善）

課題ばらしスキルの差をチームで補い,見積精度のばらつきを低減

（開発規模：約180人月、開発期間：約30ヶ月、機能追加やオプション対応で月１回リリース）

＜設定した改善策と結果＞

① 全ての課題を対象に新規性を確認（大／中／小）
→新規性が“大”（従来製品には無い機能等）について、②以降の対策を実施

② 担当者の課題ばらし結果を関係者で再確認

③ 工数バッファを持った計画を策定（バッファ量はチームで共有）
→分割テーマ毎に、通常の見積に対し 1W又は2Wのバッファを設定（2Wを超える場合は次のリリースへ繰り越す）

④ 見積工数と実績工数の差を確認し分析
→使用したバッファの中身を分析し、バッファ使用率を下げる改善策を検討

リリース期日達成率[%]*1

改善する前		改善後
0%	⇒	86%

*1）リリース期日達成率
＝ 期日達成リリース数 / リリース数 ×100[%]

All Rights Reserved – Advantest Corporation

1-3．改善事例(”待ち”の改善)

ソフト開発側でマイルストン(部署間および部署内)を設定し，着手遅れを抑制

Y	• “動作実績のあるデバイス実測評価” というマイルストンが遅延した。
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	• 次期レビジョンで、オンライン評価の開始条件(ハードの完成度調査)をマイルストン化し対応した結果、ハード評価遅れに対してソフト開発側でマクロ計画を調整し、狙い通りの影響回避ができた。

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

*) 日本能率協会コンサルティング 技術KI用語

ADVANTEST

All Rights Reserved – Advantest Corporation

1-3．改善事例(考察)

◆ 課題ばらしの質が確保されると後から分かる課題が減り，不具合が発生しにくくなる．

→ “欠陥(開発中)”の改善に関しては，6種の行動原則から「セットベース開発」を選択し，実施することが有効

◆ 新規性が高いテーマでスキルの差を補い見積り精度が確保されると，遅れが生じにくくなる．

→ “遅れ”の改善に関しては，6種の行動原則から，「LAMDAサイクルの利用」を選択し，実施することが有効

◆ ハード開発とソフト開発が並行して進む際，ソフト開発側でマイルストン(部署間および部署内)を設定し調整すると，着手遅れを抑制できる．

→ “待ち”の改善に関しては，6種の行動原則から，「組織横断活動の同期化」と「設計の時間差解除」を選択し，実施することが有効

ADVANTEST

All Rights Reserved – ADVANTEST CORPORATION

2．PM改善ナビを利用した“待ち”，“遅れ”，“欠陥”の改善活動

2-1. 課題

2-2. 解決策

2-3. 適用事例

ADVANTEST

All Rights Reserved – ADVANTEST CORPORATION

2-1．課題

“欠陥”,“遅れ”,“待ち”を改善する方法と効果の確認を踏まえ，

開発全体へ知見を普及するのに必要なことは？

・改善活動で得た知見を参照しやすい形で提供する仕掛けが必要である
（プロジェクトに張り付く方法では、開発全体へ知見を普及できない）

知見の参照が必要になるケースは？

・“欠陥”,“遅れ”,“待ち“ の改善を進める指標や問題分析の観点を確認

・課題ばらしの振返りを定着させるのに必要な活動や振返りの視点を確認

上記の仕掛けを作り試用することを次の課題に設定

・開発管理において問題分析や改善検討時の道標になってほしいという願いをこめて，仕掛けに対し「PM（Project Management）改善ナビ」と命名

All Rights Reserved – Advantest Corporation

2-2．解決策（PM改善ナビの適用対象）

①Agile方式の開発
②中規模以上の開発【30人月以上，工期は6ヶ月以上】

PM改善ナビの画面紹介（“欠陥”,“遅れ”,“待ち”の改善 に共通する画面）

■課題ばらしと振返り【中～大規模の製品開発 <工期は半年～数年>】

1つの開発を複数の小さな開発[illegible]出し（切り出された部分をイテレーション*1と呼ぶ）、段階的に機能や検証レベルを上げていく開発において、課題ばらしと振返りを次の通[illegible]ロ的な（イテレーション単位の）活動を示し、横方向はミクロ的な（イテレーションの中の）活動を示す。
[illegible]それぞれが独立した小さなプロジェクトであり、分析/設計/実装/評価等の作業から構成される[8]。

マクロの活動

原始要求
要求分析
要求仕様と実現課題（最終ゴール）
プランニング
イテレーション計画
Iteration-1
Iteration-2
Iteration-3
：
Iteration-n
最終Iteration
開発全体の振返り

2）以降、要求仕様と実現課題（最終ゴール）を ″要求″ と略記します。

*3）Iterationは、WW**やSPRINT、alpha/beta,等、自部署で使っている名称へ置き換えてください。
以降、Iteration計画を ″計画*″ と略記します。

ミクロの活動

【課題ばらし*1】	【タスク*2】	【振返り*3】
Iteration-1 タスク用の 課題抽出（①）と見積（②）と作戦（③）、リスク抽出と作戦（③）	Iteration-1 のタスク	Iteration-1 の振返り
Iteration-2 タスク用の 課題抽出（①）と見積（②）と作戦（③）、リスク抽出と作戦（③）	Iteration-2 のタスク	Iteration-2 の振返り
Iteration-3 タスク用の 課題抽出（①）と見積（②）と作戦（③）、リスク抽出と作戦（③）	Iteration-3 のタスク	Iteration-3 の振返り
：	：	：
Iteration-n タスク用の 課題抽出（①）と見積（②）と作戦（③）、リスク抽出と作戦（③）	Iteration-n のタスク	Iteration-n の振返り
最終Iteration タスク用の 課題抽出（①）と見積（②）と作戦（③）、リスク抽出と作戦（③）	最終Iterationのタスク	最終Iterationの振返り

*4）課題とは技術的に[illegible]曖昧な点や懸念事項を指す[5]。（補足：開発成果を出す際の障害になる点はリスクと同じだが、開発目標実現の[illegible]克服/解決が不可避となる点は、リスクの対極に位置する。）

*5）設計/実装/評価等の作業が1つ以上存在する

*6）振返りを対象外とする場合の判断はリーダが行う

ADVANTEST

All Rights Reserved – Advantest Corporation

2-2．解決策（PM改善ナビの機能_課題ばらしの振返り）

クリックすると改善領域を選択する画面が表示される

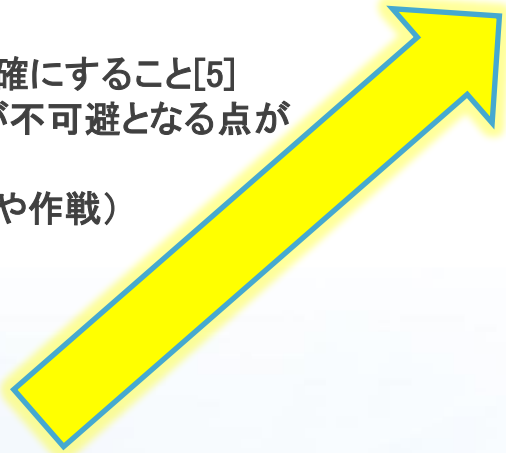
下の図において、横方向にミクロ的な（分割した開発中の）活動 を示す

【課題ばらし*1】	【タスク*2】	【振返り*3】
teration-1 タスク用の 課題抽出（①）と見積（②）と作戦（③）、リスク抽出と作戦（③）	Iteration-1 のタスク	Iteration-1 の振返り
teration-2 タスク用の 課題抽出（①）と見積（②）と作戦（③）、リスク抽出と作戦（③）	Iteration-2 のタスク	Iteration-2 の振返り
teration-3 タスク用の 課題抽出（①）と見積（②）と作戦（③）、リスク抽出と作戦（③）	Iteration-3 のタスク	Iteration-3 の振返り
： ：	： ：	： ：
teration-n タスク用の 課題抽出（①）と見積（②）と作戦（③）、リスク抽出と作戦（③）	Iteration-n のタスク	Iteration-n の振返り
最終Iteration タスク用の 課題抽出（①）と見積（②）と作戦（③）、リスク抽出と作戦（③）	最終Iterationのタスク	最終Iterationの振返り

*1）
課題ばらしは、技術的に曖昧な点や懸念事項を事前に明確にすること[5]
-> 抽出した課題は、開発目標実現のために克服/解決が不可避となる点がリスクの対極に位置する
-> 作戦は、抽出した課題を解決するための手順（段取りや作戦）

*2）
タスクは、設計/実装/評価等の作業が1つ以上存在する

*3）
振返りは原則，分割した開発毎（終了時）に実施する
-> 課題ばらしの振返りの実践が“欠陥”と“遅れ”の改善を牽引



ADVANTEST

All Rights Reserved - Advantest Corporation

2-2．解決策（PM改善ナビの機能_課題ばらしの振返り〈続き〉）

改善領域の選択

	＜＜＜ 振返りの注力点 ＞＞＞		
	①課題の抽出	②課題の見積	③作戦
欠陥	○		○
遅れ	△	○	○
待ち	△		○

【振返り対象〈例〉】

欠陥の振返り ← レビュー時に課題の抽出漏れを指摘された。

遅れの振返り ← 課題の解決を後続のイテレーションへ繰り越した。

待ちの振返り ← 1Wを超える着手遅れが発生した。

All Rights Reserved – Advantest Corporation

2-2．解決策（課題ばらしの振返り<補足>）

課題ばらしの振返りの実践が“欠陥”と“遅れ”の改善を牽引

″欠陥″の改善：

レビュー指摘欠陥やテスト検出バグの本質は，後から分かった課題（課題の抽出漏れ）

↓＜対応＞

-> 後から分かる課題を減らすための振返りを行い，課題ばらしの質向上に取り組む

″遅れ″の改善：

設計/実装やテスト作業の遅れの大元は，課題の大きさの把握誤り（課題の見積誤り）．

↓＜対応＞

-> 課題の大きさと見積りを合致させるための振返りを行い，見積精度の向上に取り組む

ADVANTEST

All Rights Reserved – Advantest Corporation

2-2．解決策（PM改善ナビの機能_問題分析）

4つのガイドワード（①～④）に基づくメッセージ[7]

①アクション(全体または一部)が実行されない
⇒作業項目（課題ばらしの実施項目）に抜けがないか確認
逐次開発できるように開発が（複数に）切り出されていることが前提
「設計の時間差解除」

②正しくないアクションが実行される
⇒作業手順（課題ばらしの手順）が顧客要求の実現から逸脱していないか確認「顧客価値の実現」

③アクションのタイミングや順序が正しくない
⇒作業する時期が全体計画から逸脱していないか確認
「組織横断活動の同期化」

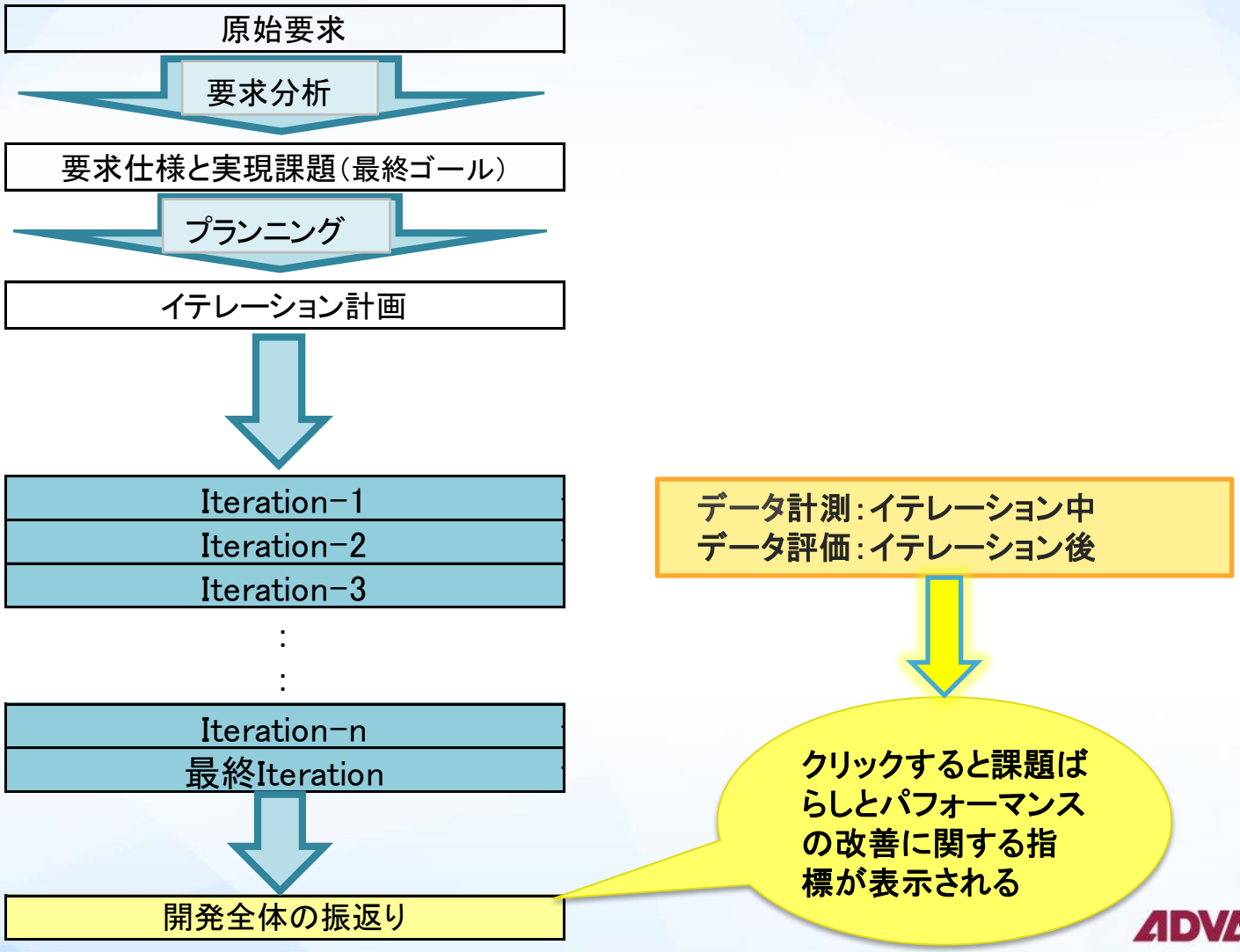
④アクションを早く止め過ぎるか長く続け過ぎる
⇒作業（課題ばらし）のペースが早すぎまたは遅すぎないか確認【分析や検討を早く止め過ぎると，予定期間で作業が終了しなかったり，作業終了後に不具合が発生する可能性が高くなる】
「セットベース開発」,「LAMDAサイクルの利用」

ADVANTEST

All Rights Reserved – ADVANTEST CORPORATION

2-2．解決策（PM改善ナビの機能_定量指標）

下の図において、縦方向にマクロ的な（分割開発単位の）活動 を示す



All Rights Reserved – Advantest Corporation

2-2．解決策（PM改善ナビの機能_定量指標 <続き>）

■開発全体の振返り

現状(または改善結果)を把握し、目標と課題を設定して改善を進める.

1.〝欠陥〟の改善【手戻りを減らす】

課題抽出率＝事前に抽出した課題 /（事前に抽出した課題＋後から分かった課題）
⇒目安は、0.85　※）開発全体の工数/工期が最小になる点
欠陥発生率＝作業終了後に他部署から指摘された不具合件数 / 開発規模[人月]
⇒目安は、0.1～0.2以下（開発規模<人月数等>に応じて目安を設定）

2.〝遅れ〟の改善【計画精度を向上】

課題解決率＝期間内に解決した課題 /（期間内に解決した課題＋繰り越した課題）
⇒目安は、0.85　※）開発全体の工数/工期が最小になる点
遅延発生率＝遅延発生イテレーション数 / イテレーション総数
⇒目安は、0.1以下（遅れの検出点<1W/2W以上,等>は期間の長さに応じて設定）

3.〝待ち〟の改善【予定通りに着手】

同期化設定率＝同期化点の設定数 / 同期化が必要な点の数
⇒目安は、1.0　※）〝待ち〟のリスクは漏れなく捉える
同期化成功率＝対策が成功した同期化点の数 / 対策を実施した同期化点の数
⇒目安は、0.85以上（同期化点の計測粒度<原則>はイテレーション単位）

備考）数年間・数十のプロジェクト実績を基に目安値を記載。

ADVANTEST

All Rights Reserved - Advantest Corporation

2-3．適用事例（試用した部門）

- 部門A：開発基盤が整備され（部門の改善手順書有り）、
  改善活動のマンネリ防止が課題。
- 部門B：新たな開発基盤を整備中(SCRUM導入)で、
  振返りの定着 が課題。
- 部門C：(開発OSが)異なるチームが合流し、
  リーダやメンバの早期立上げが課題。

ADVANTEST

All Rights Reserved – ADVANTEST CORPORATION

2-3．適用事例(効果)

◆ 開発基盤あり / マンネリ防止が課題の部門Ａでは

開発現場の改善手順書に不足している点(または無い点)を補完

⇒ 改善手順書に定量指標を追加(２点：課題ばらしの質，パフォーマンス)

◆ 新基盤導入中 / 振返りの定着が課題の部門Ｂでは

整備中(SCRUM)の基盤上で，振返りの基準(タイミングと対象)を明確化

⇒開発現場の規定に，基準を追加(２点：振返りはSPRINT毎に実施，課題抽出漏れは振返り必須)

◆ チームビルディング中の部門Ｃでは

新たに加わった開発チームが，開発基盤の意義を理解し実行するための課題を明確化

⇒ チームの改善課題を設定(２点：課題や見積結果はチームで共有，振返り結果も共有する)

ADVANTEST

All Rights Reserved – ADVANTEST CORPORATION

2-3．適用事例（考察）

PM改善ナビは，課題ばらしの振返りを定着させることや
課題ばらしの質を向上していくうえで有効

◆ 課題ばらしの質の向上に寄与
理由：部門Aで、課題ばらしの質の評価指標を使うようになった

◆ 課題ばらしの振返りの定着に寄与
理由：部門Bで、新たな基盤へ振返りの実施（SPRINT毎）を組み込んだ

◆ 課題ばらしの定着に寄与
理由：部門Cで，新たに合流したチームが課題の共有化に取り組んだ

ADVANTEST

All Rights Reserved – ADVANTEST CORPORATION

3．まとめ

“欠陥”,“遅れ”,“待ち”を防ぐプロセス改善導入のポイント

ステップ3:「課題ばらしの質向上」に取り組む

⇒ 改善した結果，開発チームが
「不具合が減った」,「遅れなくなった」,「狙い通りの対処ができた」
という実感を持つことが何よりも大切.

↑

ステップ2:「課題ばらし(見積含む)の振返り」を定着させる

⇒ 担当者が後から分かった課題を分析した後，チームで分析結果に対する意見交換を行い，チームとしての改善策を決めると効果的.

↑

ステップ1:「課題ばらし」を定着させる

⇒ チームで課題ばらしをフォローし，苦手な人も課題ばらしができるようにする.
取組みは，LAMDAサイクルの利用が効果的.
-> Look　-> Ask　-> Model　-> Discus　-> Act
テーマを確認し課題は何かを問い掛け，課題を出し合って洗練し，課題を登録

ADVANTEST

All Rights Reserved – Advantest Corporation

4．今後の課題

“欠陥”,“遅れ”,“待ち”を改善する知見の形式化を踏まえ，

改善に関する知見は現状の内容で充分か？

・開発後に発見される不具合の低減に寄与する内容を追加する必要あり
（現状の内容は，開発中に起きた問題に関する改善情報）

新たな課題は，開発後の不具合を低減する課題ばらしの明確化

・6種の行動原則の中から，「顧客価値の実現」と「知識の再利用」を選択

ADVANTEST

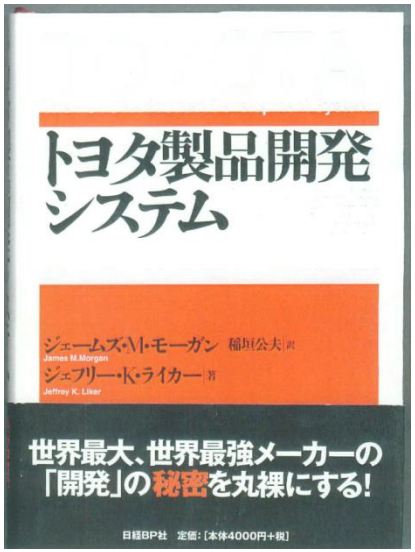
All Rights Reserved – Advantest Corporation

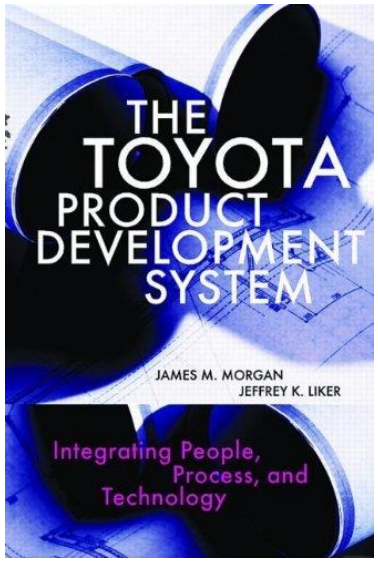
ご清聴ありがとうございました。

All Rights Reserved - Advantest Corporation

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





All Rights Reserved - Advantest Corporation

参考文献(続き)

[2]比嘉定彦他：SQiP2013_報文集_B3-1「トヨタ開発方式の利用による
ソフト開発のQCD向上活動」，(財)日本科学技術連盟，2013
[3]比嘉定彦他：SQiP2014_報文集_A3-3 トヨタ開発方式の深掘りによる
ソフト開発のQCD向上活動，(財)日本科学技術連盟，2014
[4]比嘉定彦他：SQiP2015_報文_B3-1 トヨタ開発方式の包括的利用に
によるソフト開発のQCD向上活動，(財)日本科学技術連盟，2015
[5]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
[6]稲垣公夫：開発戦略は「意思決定」を遅らせろ，中経出版，2012
[7]ET2015/IoT2015併催：IPAセミナー
http://www.ipa.go.jp/sec/seminar/20151119.html
IPAセミナー【第4部】
～組込みシステムの安全解析と障害原因診断分析の統合アプローチ／
STAMP～
[8] Frank Buschmann / Regine Meunier / Hans Rohnert / Peter Sommerlad /
Michael Stal /
, Pattern-Oriented Software Architecture ソフトウェア開発のためのパターン
体系，
近代科学社，2000

All Rights Reserved - Advantest Corporation

＜付録　課題ばらしと作業ばらしの比_製品Aと A’＞

欠陥率＝工程終了後に他部署から指摘された不具合の件数 / 人月

製品A				
開発規模	課題ばらしが有るインクリメント数（中盤～）	課題ばらしが無いインクリメント数（同左）	課題ばらし無しの割合	欠陥率
90人月	49	27	約36%	0.131

製品A’				
開発規模	課題ばらしが有るインクリメント数	課題ばらしが無いインクリメント数	課題ばらし無しの割合	欠陥率
45人月	46	11	約19%	0.02

最適化点は、課題ばらしを網羅し 余剰ゼロの点。

無駄ゼロの課題ばらし網羅点"X"
36%＞"X"＞19%

ADVANTEST

All Rights Reserved – ADVANTEST CORPORATION

2-1．課題(PM改善ナビの利用対象)

「PM(Project Management)改善ナビ」は，Agile開発向けに製作

調査結果：
- トヨタ開発方式の行動原則「設計の時間差解除」は，逐次開発を進める点でAgile方式と符合
- 逐次開発の終了時に目標に対する実績の差を精査し，次の逐次開発の目標を設定する点についても Agile方式と符合

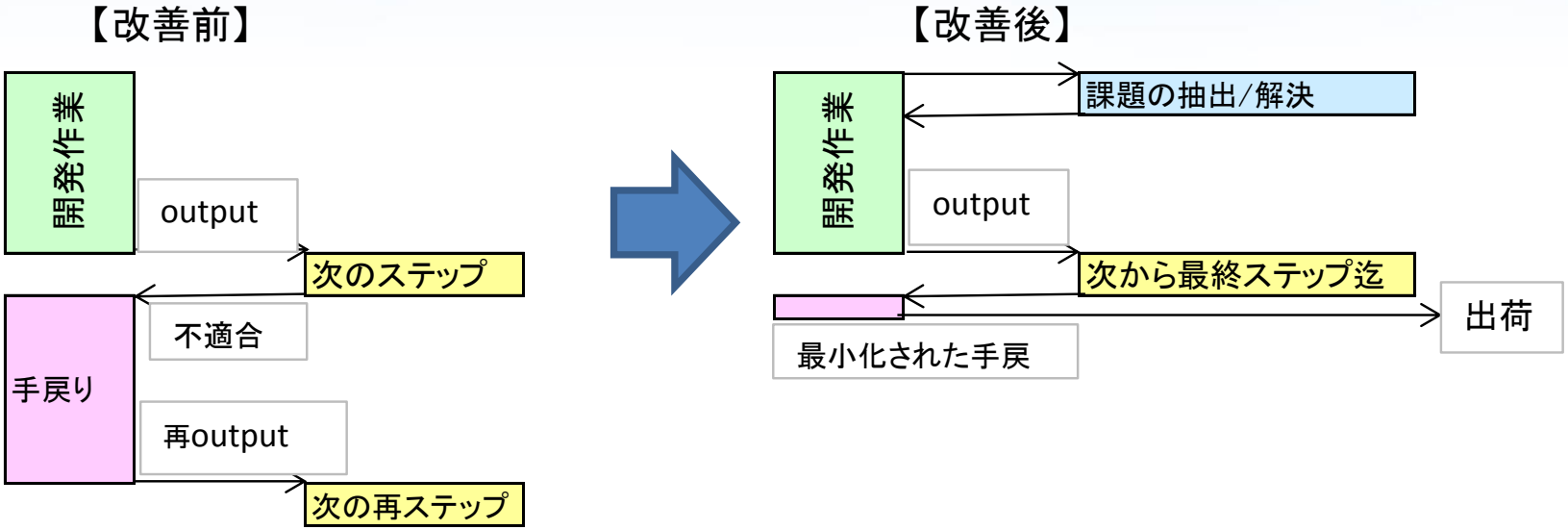
検討結果：
- Agile方式で開発している組織ではトヨタ開発方式に基く知見を利用しやすい

All Rights Reserved – Advantest Corporation

“欠陥”の改善（先頭部分の画面紹介）

「欠陥」の改善

原則：問題を前段階で解決することによってプロセス中の手戻りを無くす[1]。



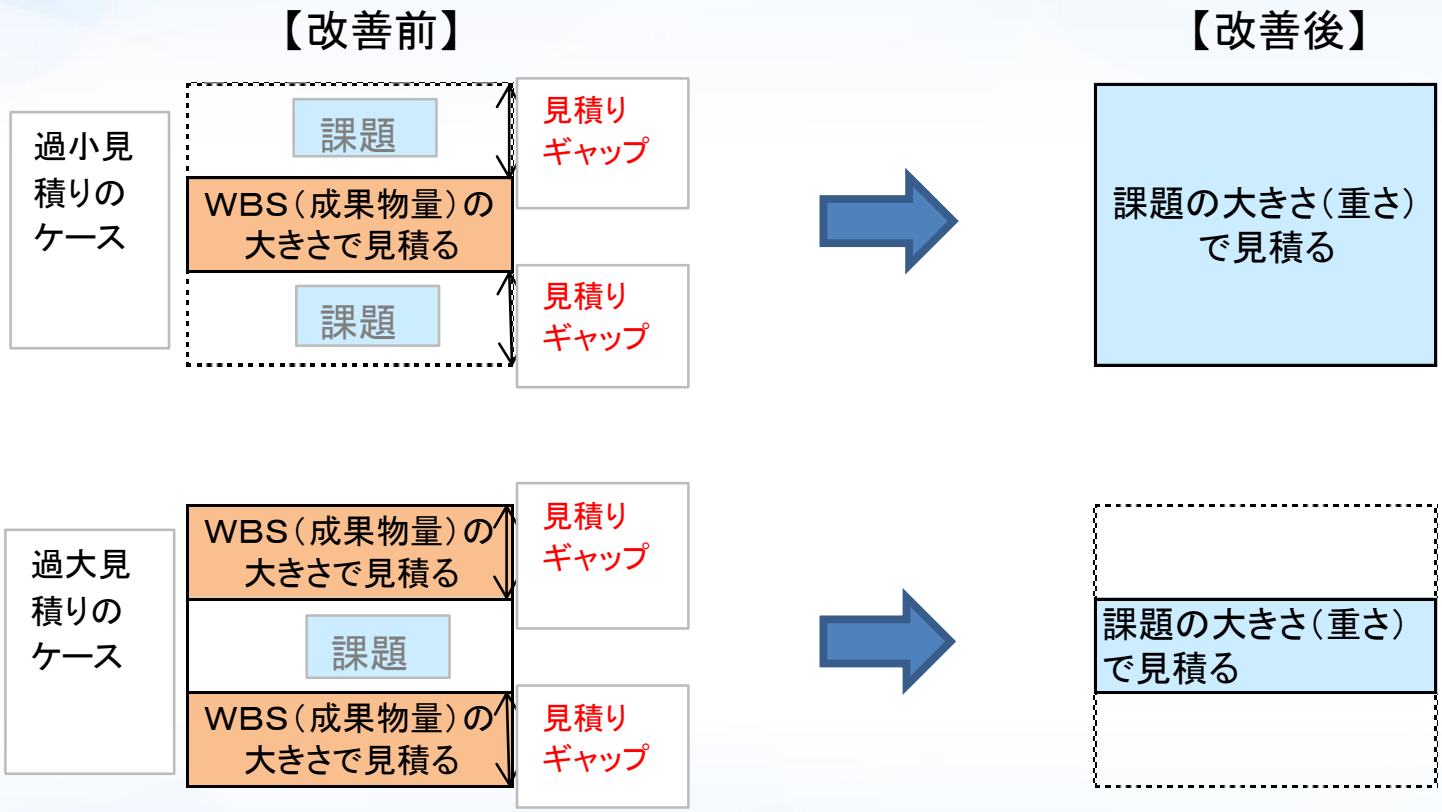
理由：品質の良し悪しは、欠陥の作り込みを抑制するプロセスの充分性に大きく依存し
課題の抽出/解決の取り組み方(方法)が品質向上の決め手になる[1]。

All Rights Reserved – Advantest Corporation

“遅れ”の改善（先頭部分の画面紹介）

「遅れ」の改善

原則：仕事や作業が足並みを揃えて進むように調子を合わせる仕組みを作る[1]。



理由：開発に要する期間は課題の大きさ(課題を解決し得られる付加価値の大きさ) と質（課題の新規性や複雑性)に依存し、課題を正確に見積る事が遅延を抑制する決め手になる[5]。

ADVANTEST

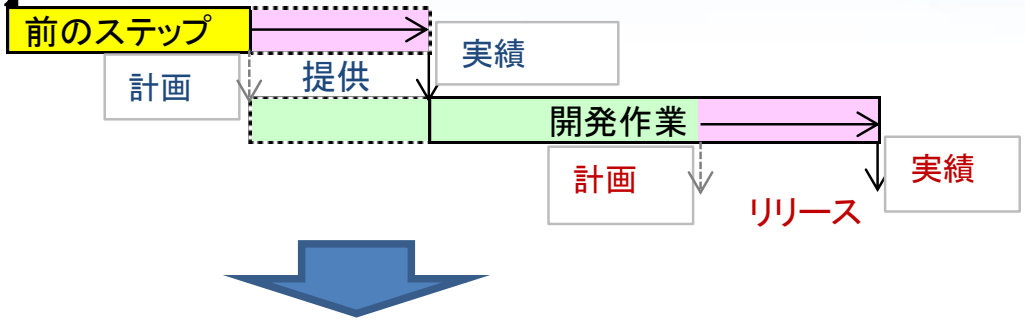
All Rights Reserved – Advantest Corporation

“待ち”の改善（先頭部分の画面紹介）

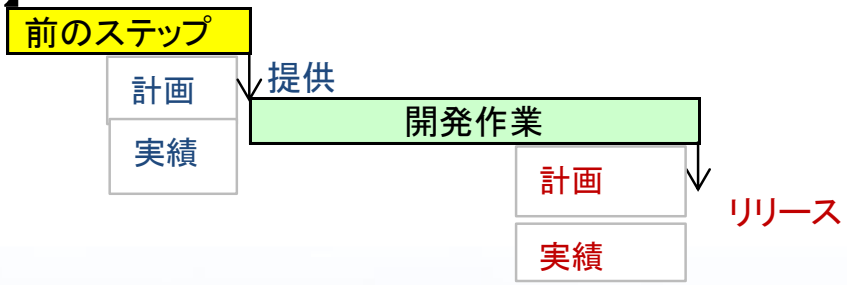
「待ち」の改善

原則：横断的活動（複数の組織や複数の担当）を同期させ、必要な情報やものが必要な時にやり取りされるようにする[1]。

【改善前】



【改善後】



理由：プロセスのアウトプットは全て（他の機能や階層等にとっての）インプットとなる。アウトプットがインプットの要求を満たさない場合は手戻りとなり、修正される迄の間、それを使うインプットは待たされる[1]。

All Rights Reserved – Advantest Corporation

付録（行動原則の補足３）

◇知識の再利用

知識を形式化して再利用し，課題の抽出/解決力を向上する.

⇒課題ばらしの再利用を知識の再利用とした

パターン[8]の概念を利用して形式化する

ソフトウェア アーキテクチャ のための パターン[8]	
前提	設計上の課題が発生する状況
課題	その前提で繰り返し現れる設計上の課題 −容易に追加/変更できる −他機能に影響を及ぼさない
解決策	設計課題を解決するための処置

応用

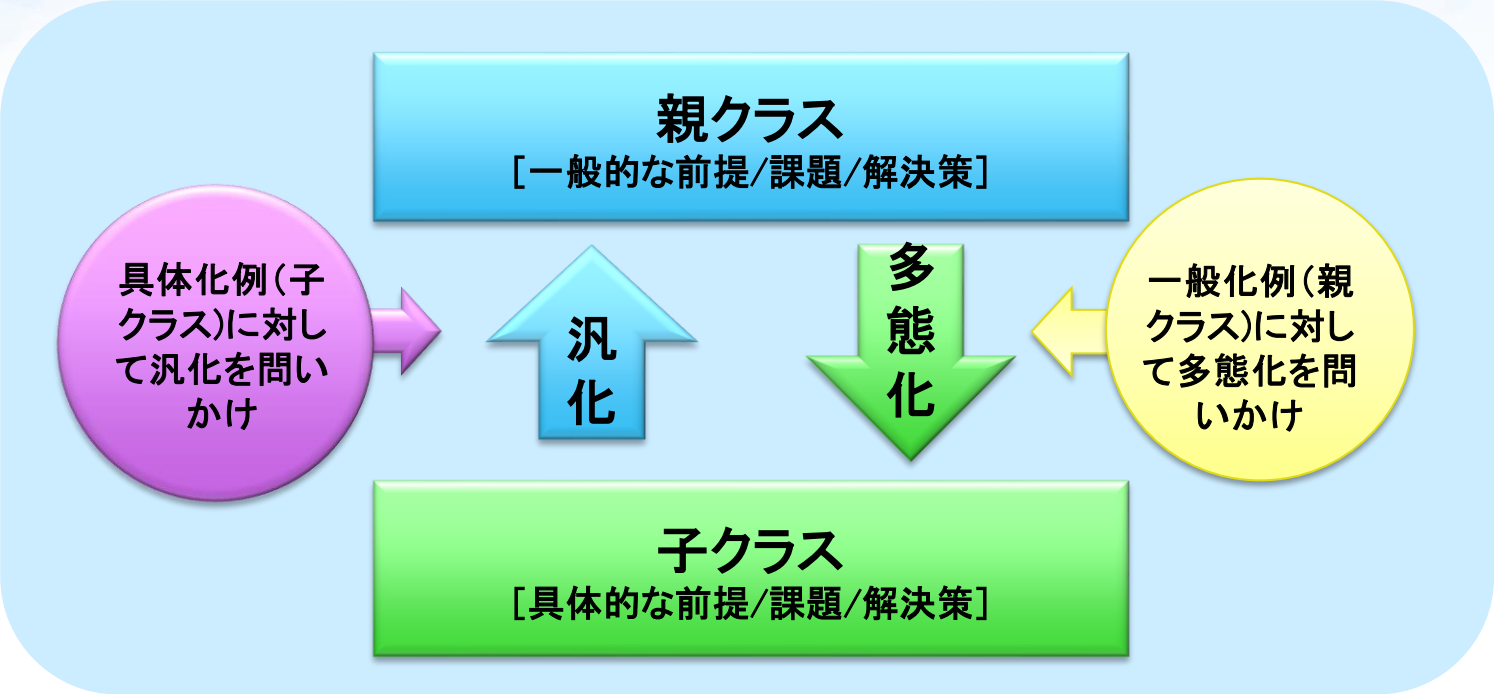
課題ばらしを形式化するためのパターン	
前提	課題ばらしが必要となる状況
課題	その前提で繰り返し現れる抽出すべき課題 −開発の阻害要因を解消/回避できる −相互作用の範囲/内容を特定できる
解決策	抽出した課題の解を導くアプローチ

All Rights Reserved − ADVANTEST CORPORATION

付録（行動原則の補足３<続き>）

◇知識の再利用

汎化と多態化をセットにして形式化し、再利用時の負荷を軽減する。



汎化のポイント　：再利用時の多態化が可能となる抽象度で記述する。
多態化のポイント：再利用時の課題ばらしに合うようにカスタマイズする。

All Rights Reserved – ADVANTEST CORPORATION

付録_課題ばらしの再利用_開発終了後の不具合低減編

パターン	課題ばらしの再利用（親クラス）	←クラス名
・タイトル ・前提 ・課題	・ハード制限チェック対応における条件抜けの防止 ・動作条件の組み合わせによりハードの最大(制限)構成が変わる。 ・動作条件の組み合わせによりハードの最大(制限)構成が変わる場合、従来の方法では対応できない事がある。	←メンバ
・解決策	ケースＡ：ハード構成が特定の動作モードだけで決まる場合、既存の方法で対応。 ケースＢ：ハード構成が複数の動作条件の組み合わせで変化する場合（又は可能性がある場合）、影響を受ける条件を洗い出し、最適な対応方法を検討。	←メソッド

汎化(つまり、‥)↑　　↓ 多態化(例えば、‥)

パターン	課題ばらしの再利用（子クラス）	←クラス名
・タイトル ・前提 ・課題	・ABCxnn && Nway以上のHW制限チェックにおける条件抜けの防止 ・動作条件の組み合わせによりABC構成(がxnnになるかどうか)が変わる。 ・ABC構成(がxnnになるかどうか)が変わる場合、問い合わせだけでは対応できない事がある。	←メンバ
・解決策	ケースＡ：ABCxnnになるのはＡの動作で決まる場合、ビットサイズの問い合わせで対応。 ケースＢ：ABCxnnになるのは次の動作条件の組み合わせによる場合、**_Checkにより対応。 ・機能Ａの動作設定 ・機能Ｂに関するモード ・機能Ｃのビット数とビット幅 ・機能Ｃと機能Ｄの対応に関するモード	←メソッド

ADVANTEST

All Rights Reserved – Advantest Corporation