

# 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

比嘉 定彦  
株式会社アドバンテスト  
[sadahiko.higa@advantest.com](mailto:sadahiko.higa@advantest.com)

## 要旨

開発工程の中で発生する「待ち」や「遅れ」の状態をムダであるとし、(「欠陥」と同様に)改善する手法として、トヨタ開発方式[1]というのがある。我々は、必要な部分から順次カスタマイズし、開発現場へ導入した。その結果、トヨタ開発方式は開発の「待ち」、「遅れ」、「欠陥」を未然に防ぐ有効な解決法であることを確認した[2][3][4]。

開発の「待ち」、「遅れ」、「欠陥」の改善は、プロジェクト管理が扱う範囲全体から見ると一部の領域にすぎない。しかし、プロジェクト管理の実施効果は製品開発のQCD実績で把握される点を考慮すると、開発の「待ち」、「遅れ」、「欠陥」を改善する活動領域は重要である。

我々は数年に渡るトヨタ開発方式の利用実績を基に、開発の「待ち」、「遅れ」、「欠陥」を改善するための要点を体系化して「PM(Project Management)改善ナビ(以降PM改善ナビと記述する)」へ組み込み、開発現場で適用を試みた。

本報告では、開発の「待ち」、「遅れ」、「欠陥」を防ぐプロセス改善を提案し、その改善活動を支援する「PM改善ナビ」を開発現場に適用した結果を紹介する。

## 1. はじめに

本報告では、「開発の"待ち", "遅れ", "欠陥"を防ぐプロセス改善」について論じ、次に「PM 改善ナビを利用した"待ち", "遅れ", "欠陥"の改善活動」について論じる。考察は前/後者毎に、まとめは2者をまとめて論じる。

前者の「開発の"待ち", "遅れ", "欠陥"を防ぐプロセス改善」では、課題ばらし\*1の十分性と、プランニング\*2との整合性確保が、改善のベース(土台)になることを論じたうえで、「欠陥」、「遅れ」、「待ち」を防ぐ改善原則(3種類)を紹介する。

\*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]。(補足:開発の阻害要因となる点は

リスクと同じだが、開発目標実現のために克服/解決が不可避となる点は、リスクの対極に位置する。)

\*2) 課題を解決するための対策手順(段取りや作戦)を明確にすること

後者の「PM 改善ナビを利用した"待ち", "遅れ", "欠陥"の改善活動」では、改善活動を支援するPM改善ナビへ集約した知識(分析項目や定量指標)等について紹介する。

本報告の内容を正しく理解して頂くため、はじめに、WBSやレビュー、テスト等のソフトウェア開発管理で使われる用語が果たす役割について、一般と本報告での違いを述べる。これは、(繰り返しになるが)本報告内容の理解のためであり、プロジェクト管理の定石やその効用を否定する意図はないことを予め明確にしておく。

開発工期はWBS(Work Breakdown Structure)の総量と総マンパワーを基に算定する通念(一般論)が有るが、実際にはWBS以外の要因の方が開発工期へ大きく影響する。実際に要する期間は課題の大きさ(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)の累積に依存し、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になる。加えて、開発品質はレビューとテストで如何に欠陥を除去するかによる、という通念(一般論)が有るが、実際には欠陥の作り込みを抑制するプロセスが開発製品の品質(良し悪し)に大きく関わる。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であり、「課題ばらしの質の向上」の取り組みこそが開発(作込み)品質向上の決め手になる。

プロジェクトの管理はWBSをベースにした方法論が一般的である。しかし、開発工期の支配要因ではないWBSを一般論の通りに管理しても、(前述した理由により)期待する開発遅延抑制の効果を得るのは難しい。加えて、ソフトウェアの品質管理は「レビュー」と「テスト」をベースにした方法論が一般的である。しかし、作込み品質向上の決め手とはならない「レビュー」と「テスト」を一般

論の通りに実施しても、(前述した理由により)期待する作込み品質向上の効果を得るのは難しい。また、欠陥の発生原因分析は、課題ばらしの振り返りと比べて属人性が高く(分析者への依存度が高い)、即時性と網羅性が十分得られないため、作込み品質向上の効果は限られる。

我々は、一般の知識体系を利用して改善に活用するのではなく、トヨタ開発方式の実践によって自前で得た経験や結果を体系化し、開発現場の改善活動で利用することにした。

## 2. 開発の"待ち", "遅れ", "欠陥"を防ぐプロセス改善

開発現場では、しばしば次のような問題に遭遇する。

- ・実装後に不具合が多発した(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なかった(特に新規性が高い機能の場合)
- ・他部署から提供されたものが使用要件を満たさず、作業の着手が遅れた

トヨタ開発方式[1]では、1つ目の問題は"欠陥", 2つ目は"遅れ", 3つ目は"待ち"と捉え、3つとも開発の流れを阻害する要因と認識し、改善する対象としている。我々は "欠陥", "遅れ", "待ち"の問題に対し、トヨタ開発方式が提示する解決方法を開発現場の改善活動で利用することにした。

### 2.1. 問題の分析\_その1

本章では"欠陥", "遅れ", "待ち"の問題の解決に向けて、分析した経過を述べる。

トヨタ開発方式は次の3つの改善原則を提示している。  
[1]

- ・原則 1: 問題を前段階で解決することにより、プロセス中の手戻りを無くす「"欠陥"の改善」。
- ・原則 2: 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「"遅れ"の改善」。
- ・原則 3: 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「"待ち"の改善」。

上述した3つの改善原則を参考に、弊社で"欠陥", "遅れ", "待ち"の問題を分析した結果を次に示す。

"欠陥"の分析:

ある製品(派生機種の開発)で、開発終盤に発生した

不具合(手戻り)が課題ばらしの質(前段階のプロセス)に関係したかどうかを問い、調査した結果、大半が課題ばらしに起因する問題だった。そこで、課題ばらしの質と不具合の発生がどのように関係したかを問い、調査した結果、課題の抽出漏れが多い部分から不具合が多く発生していた。

"遅れ"の分析:

ある製品(新製品の開発)で、テーマ毎の遅延の大きさが見積りの精度に関連したかどうかを問い、調査した結果、新規性が高くなるにつれて工数の増加傾向が顕著だった。そこで、工数の増加が顕著になる理由を問い、調査した結果、機能追加と同様の課題ばらしと見積りを(新機能に対しても)一様に行っていた。

"待ち"の分析:

ある製品(新ハードのサポート)でソフト評価作業の着手が遅れたことについて、ハード開発部署と事前に摺合せた内容を問い、調査した結果、使用条件について事前に打合せを行い、ソフトが使用する機能について合意していた。しかし、動作条件については、ハード開発側の考えとソフト開発側では差異が有った。

### 2.2. プロセス改善の土台

本章では、トヨタ開発方式を利用したプロセス改善のベース(土台)となる課題ばらしの十分性と、プランニングとの整合性について述べる。

“欠陥”，”遅れ”，”待ち”を防ぐ土台(その1)は、次に示すとおり課題ばらしの十分性を確保することである。

- ・開発当初(～計画時)に、開発全体を通して 心配や不明な点を十分に洗い出し技術課題を明確化する(理由-> 必要な課題ばらしが漏れていると、後から分かった課題となって顕れる)。

“欠陥”，”遅れ”，”待ち”を防ぐ土台(その2)は、次に示すとおり課題と計画の整合性を確保することである。

- ・抽出した課題の解決をベースにした作戦(優先順位の設定等)を立てる(理由-> 課題と計画の不整合は、課題ばらしの効果を減少させる)。

課題と計画の間の整合性確保は一見、出来て当たり前様である。しかし、技術課題がチーム内で共有されていなかったり、課題の解決プランが担当者により明確化されていなかったり等の場合があり、実行するのは簡単ではない。

### 2.3. 問題の解決\_その1

本章では、開発の"待ち", "遅れ", "欠陥"を防ぐプロセス改善について述べる。

トヨタ開発方式は"欠陥","遅れ","待ち"を改善するため、次の6つの行動原則を提示している。

⇒設計の時間差解除[1]

-> 開発を分割して設計情報を段階的に解除し、(一定期間内の取り組み対象を絞り込み集中することで)開発の流れを作る

⇒組織横断活動の同期化[1]

-> 組織毎の入/出力と相互依存性をもとに、同時進行している活動がうまく統合するように調整する

⇒顧客価値の探究と実現[1]

-> 要求と真のニーズを抽出して顧客価値を創りこみ、実現する。

⇒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

※:弊社では、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

⇒知識の再利用[1]

-> 知識を形式化して再利用し、技術課題の抽出/解決力を向上する

※:弊社では、課題ばらしの再利用を知識の再利用とした

⇒LAMDA サイクル\*3[6]の利用

\*3)PDCA サイクルを開発向けに発展させたもので、LAMDA サイクル2回が PDCA 1回に相当する。

- Look : 観察する
- Ask : 問いかける
- Model : モデル化する
- Discuss: 話し合う
- Act : 行動する

-> 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

前述した改善原則(項 2.1)は、上述した6種の行動原則(トヨタ開発方式の知見)から、問題解決に必要なものを選択する事で実現する。



図1 トヨタ開発方式の6つの行動原則[1][6]

続いて、弊社で"欠陥", "遅れ", "待ち"の問題の改善方法を検討した結果(行動原則の選択を含む)を示す。

"欠陥"の改善:

不具合が多発した問題に対しては、課題ばらしの質を

一定水準へ改善すること(セットベース化)で、開発終盤の不具合を減らせるか確認する。加えて、どの程度水準へ改善すれば不具合が減るのか、その水準値も把握する。【セットベース開発】

"遅れ"の改善:

遅れるテーマが多発した問題に対しては、新規性の高いテーマの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して遅れを抑制する。

【LAMDA サイクルの利用】

"待ち"の改善:

評価作業の着手が遅延した問題に対しては、ソフト開発側の期待値を明確にしてハード開発へ伝え(期待値との差異を確認または生じないようにする)、部署間のマイルストーンを設定する。【組織横断活動の同期化】加えて、テストを細分化してハード開発と並行して進められる様にする(ハードの機能を段階的に提供してもらい、ソフトの評価を逐次進められる様にする)。【設計の時間差解除】

## 2.4. 改善事例

本章では、開発の“欠陥”、“遅れ”、“待ち”を改善した事例を紹介する。

最初に、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した事例(“欠陥”の改善例)を紹介する。品質要求水準が高い製品開発で課題ばらしの質確保に取り組んだ結果、品質が向上した。この活動で使用した改善指標と結果(改善効果)を以下に示す。

表 1 改善指標と改善効果

測定項目	
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)
指標2.	パフォーマンス<品質> = 最終マイルストーンで行った不具合修正件数 / 開発工数[人月]

改善前 / Rev. A	改善後 / Rev. A+1
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
0.13 件/人月	0.02 件/人月

90 人月の製品開発

45 人月の製品開発

上述した改善について、経過を以下に示す。

活動にあたり、初めに課題ばらしの質を測定するための計測基準を設け(全チームで統一)、あるレビジョン(Rev.A)から計測を開始した。結果を分析したところ、最

終マイルストーンで行った不具合修正が特定ユニットに集中し、このユニットの課題ばらしの質は一定レベルに届いていなかった。一方、課題ばらしの質が一定レベル確保されていた他のユニットでは、最終マイルストーンでの不具合修正は殆ど無かった。不具合が多発した特定ユニットの振返り(分析)記録では、約8割が課題ばらしの問題(事前抽出が可能)となっていたことに加え、改善案(振返り結果)を実行することで、他のユニットと同一水準へ改善できることが分かった。不具合修正が多発した特定ユニットの開発チームは、次のレビジョン(Rev.A+1)で改善策を実施した。結果、課題ばらしの質は一定レベルを超え、品質が大幅に向上した(最終マイルストーンでの不具合修正は無くなった)。製品全体においては最終マイルストーンで行った不具合修正は1件のみとなり、品質が大幅に向上した。

上記の事例は、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ことを裏付けた。加えて、“欠陥”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であることを示した。

次に、課題の見積スキルの差をチームで補い、見積精度のばらつきを低減した事例(“遅れ”の改善例)を紹介する。新規性が高い製品開発で見積精度のばらつき低減に取り組んだ結果、工期遅れを低減した(開発規模は約180人月、アジャイル方式により月1回のリリース)。この活動で行われた改善策(4点)を以下に示す。

- ① 全ての課題を対象に新規性を確認(大/中/小)  
→新規性“大”(従来製品には無い機能等)のテーマについて②以降の対策を実施
- ② 担当者の課題見積り結果を関係者で再確認し、チームで課題を抽出して大きさを見積る  
※: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)
- ③ 数バッファを持った計画を策定(チームで共有)  
→テーマ毎に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 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

上記の改善(見積精度ばらつきの低減)を行った結果、次のとおり、ほぼ期日通りにリリースできるようになった。

#### リリース期日達成率[%]\*4

改善する前	⇒	改善後
0%		86%

#### \*4) リリース期日達成率

= 期日達成リリース数 / リリース数 × 100[%]

上記の事例は、新規性が高いテーマの見積り精度が確保されると、課題の大きさと見積りの差が縮小し、遅れが生じにくくなることを裏付けた。加えて、“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であることを示した。

最後に、情報やものの必要時期と内容(機能や検証の網羅度)について提供部署の合意を(事前に)取り、開発中に生じた変化にも適切に対応した事例(“待ち”の改善例)を紹介する。評価用のハードウェアが提供されたが(完成度が低いため)、予定通りに作業に着手できなかった問題を振返り\*5、次の開発で改善策を実施した結果、待つ状態を最小化した(開発規模は10人月位から100人月迄の範囲内)。この活動で行われた振返りの結果(改善策)と効果を以下に示す。

\*5) 振返り[5] → 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” 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” という情報があつたが、ソフトウェアテストに耐える状態では無かつた。

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において、キャプチャが100%動作可能なハード状態で、評価を開始する”。

加えてテストを細分化し、ハード開発と並行して進むソフト開発側の対応に柔軟性を持たせる。

次期開発で上記の改善を行った結果、ハード開発遅れに対してソフト開発側の影響を最小化できた。

上記の事例は、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ーン(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ことを裏付けた。加えて、“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であることを示した。



□待ちの振返り ← 1Wを超える着手遅れが発生した。

## 2)分析対象アクション:

"欠陥", "遅れ", "待ち"の画面別に分析対象アクションを表示する。

【画面】 【分析対象アクション】

"欠陥"の改善 → 課題ばらし(技術的に不明や心配なことを事前に明確化すること)

"遅れ"の改善 → 課題の見積(課題の重さを把握し, 重さに応じた工数を設定すること)

"待ち"の改善 → 同期化作業(必要な時に必要なものがやりとりされるように, 関連する機能や作業と同期を取ること)

## 3) ”ナビ”の分析構造:

"欠陥", "遅れ", "待ち" を分析しやすくする様に, 4つのガイドワード\*6(①~④)に基づくメッセージを表示する。

\*6) [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)に抜けないか確認. 1つの開発が複数の小さな開発の集合体となるよう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(課題ばらしや見積)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(課題ばらしや見積)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」

## 4)定量指標の利用:

課題ばらしの質とパフォーマンスを見える化する指

標(数値化方法と目安値\*7)を表示する。

\*7) 数年間・数十のソフト開発プロジェクトの実績(確認/調査結果)を基に目安値を設定

"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

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

"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

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

"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

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

## 3.4. 適用事例

本章では, ”ナビ”の適用例(試用結果)を述べる。

部門A: 開発基盤が整備され(部門内の改善手順書有り), 改善活動のマンネリ防止が課題  
効果 ⇒ 開発現場で使われている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: 課題ばらしの質の評価, パフォーマンスの評価)

部門B: 新たな開発基盤を整備中(SCRUM 導入)で, 振返り(SPRINT 毎)の定着が課題  
効果 ⇒ 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,  
課題抽出漏れのレビュー指摘は  
振返り必須)

部門C:開発製品が異なるチームが合流し,リーダーや  
メンバの早期立上げが課題

効果 ⇒ 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為の  
課題(必須部分)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し,  
振返り結果もチームで共有)

部門Aでは課題ばらしの質の評価指標を使うようになり,  
”ナビ”が課題ばらしの質の改善に寄与することを確認した。  
部門Bでは SPRINT 毎に振返りを行うことが組織のルール化となり,  
”ナビ”が改善の土台を構築することに寄与することを確認した。

上記の試用結果から,“ナビ”は目標(利用モデル)に対し有効であることを確認した。

#### 4. まとめ

開発工期と品質の改善に取り組む際,開発の流れを阻害する  
”待ち”, ”遅れ”, ”欠陥”を防止していくことが課題となる。  
“欠陥”, “遅れ”, “待ち”を防ぐプロセス改善の導入は次の3つの  
ステップで進めると良い。

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。定着させる為には,  
LAMDA サイクルの利用が効果的である。

※:チームでテーマを確認し(Look)

チームへ課題を問い(Ask)

課題を出し合い(Model)

話し合っ洗練し(Discuss)

チームとしての課題を登録(Act)

ステップ2:「課題ばらしの振返り」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  
的である。

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  
大切である。

#### 5. 今後の課題

今後, ”欠陥”, ”遅れ”, ”待ち”の改善を進める上での課題は  
何かを問い, 開発が置かれている状況を調査した。結果, 労働時間  
短縮の流れの中で保守工数の割合が増加していた(理由は開発終了  
後の欠陥の減少よりも開発稼働時間の減少の方が顕著なため)。弊  
社での保守作業は開発作業とのマルチタスクになる場合が多く, 保  
守工数の割合増加は開発スループットの低下要因となる。そこで,  
我々はライフサイクル全体で(開発終了後の不具合発生低減を含め  
て) ”欠陥”, ”遅れ”, ”待ち”の改善に取り組むことを新たな課  
題とした。

開発終了後の欠陥を減少させるために必要なことを問い, 調査した  
結果, 大半は課題ばらしに起因する問題であった(レビュー・テストで  
検出できず市場へ流出)。加えて開発終了後に発見される不具合は  
複雑度が高く,(開発終了後の不具合修正は全て改善策の報告対象と  
なっているが)前段階での解決が容易でないことを確認した。とはい  
え, 分析・検討時に抽出が漏れてしまった課題は, レビュー・テスト  
時には捕捉がさらに難しい。

現在, 開発終了後の不具合を低減するための課題ばらしについて検  
討を進めている。この改善策を導き出すため, 取組みはトヨタ開発  
方式が提示する問題解決方法(6種の行動原則)の中から, 次の2つを  
選択した。

- ・顧客価値の探究と実現
- ・知識の再利用

上記の取組みにより, 開発終了後の不具合を低減する課題ばらし  
方法を明確化していきたい。

#### 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] 比嘉定彦他:SQiP2013\_報文\_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\_B3-1 トヨタ開発方式  
の深堀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」を遅らせる, 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】  
～組み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～