

# Technology Reports

## スマートフォン向けアプリケーション開発における 統一的な定量的開発管理プロセスの導入

ドコモにおけるスマートフォン向けアプリケーション開発の初期においては、多数のアプリケーションの早期導入が求められる環境とマルチベンダ化の進展により、開発現場ごとに開発管理の詳細度にばらつきが生じ、開発管理プロセスが属人化していた。そこで、ベンダ各社の開発管理状況の報告様式および部内開発完了時の品質報告様式を統一化し、標準的な定量的開発管理プロセスを部内に導入・定着化を行った。本稿では上記プロセス改善活動について解説する。

移動機開発部  
とさき たかし やまだ よしひろ  
戸崎 貴資 山田 善大  
はっとり ひろゆき  
服部 弘幸

### 1. まえがき

近年、携帯電話業界はフィーチャーフォンからスマートフォンへのシフトが急速に進み、ドコモとしても多数のスマートフォン向けアプリケーションを早期に開発・導入する必要がある、要求される品質・コスト・納期が多様化してきていた。そのため、従来開発委託していた発注先ベンダ（以下、ベンダ）では、開発リソースの不足・要求への不適應などの理由から新規ベンダの採用が増加し、急速にマルチベンダ化が進んだ。しかし開発管理においては、早期サービスインが優先され、そのプロセスは部内各アプリケーション開発担当者およびベンダに依存し、統一性なく属人的なものとなっていた。そのため、開発管理情報の詳細度の相違から商用リリース判定会議での品質報告内

容もばらつきがあり、品質判定に必要な情報を質疑応答により確認することが多く、リリース判定に時間を要した。さらに、品質確認漏れを発生させる可能性が高い状況であった。

そこで、プロセス改善チームにて、ベンダの開発状況の報告様式、商用リリース判定時の品質報告様式を統一化し、開発現場に導入、各種施策により定着化を実施した。また、これらの様式を用いた定量的開発管理<sup>\*1</sup>プロセスが組織内に浸透するよう継続的に啓発活動を実施した。

本稿では、これまで実施したプロセス改善活動について解説する。

### 2. 開発プロセスと課題

2.1 アプリケーション開発に関する組織構成・開発上の役割分担  
アプリケーション開発に関する組

織構成・役割分担の概略を図1に示す。主な部内組織としては、各アプリケーション開発担当、これを横断して取りまとめるPMO（Project Management Office）<sup>\*2</sup>担当、および維持管理担当が存在する。アプリケーション開発担当は、サービス主管部門の要求条件を基に要件定義書<sup>\*3</sup>を策定し、ベンダにソフトウェア開発（基本設計～総合テスト）を委託する。また、ベンダの開発が完了し、社内受入れテスト完了後に、組織長とPMO担当長、維持管理担当長が商用リリース可否を判定する。

### 2.2 対処すべき課題

図1の①におけるアプリケーション開発担当一ベンダ間の開発管理は、定期的な開発状況の情報共有ミーティング（以下、情報共有MTG）こそ実施されていたものの、開発管

©2016 NTT DOCOMO, INC.  
本誌掲載記事の無断転載を禁じます。

\*1 定量的開発管理：客観的なデータ・事実に基づく開発管理手法。定量的開発管理の導入有無は開発プロジェクトの成否に大きな影響を与える。

\*2 PMO：組織内における個々のプロジェクトマネジメントを横断的に統括・管理・支

援する専門部署。

\*3 要件定義書：顧客が望んでいる機能や仕様などについて、その概略をまとめた文書。後工程の成果物はすべて要件定義書の要件を満たすように作成されるため、開発の原典となる。

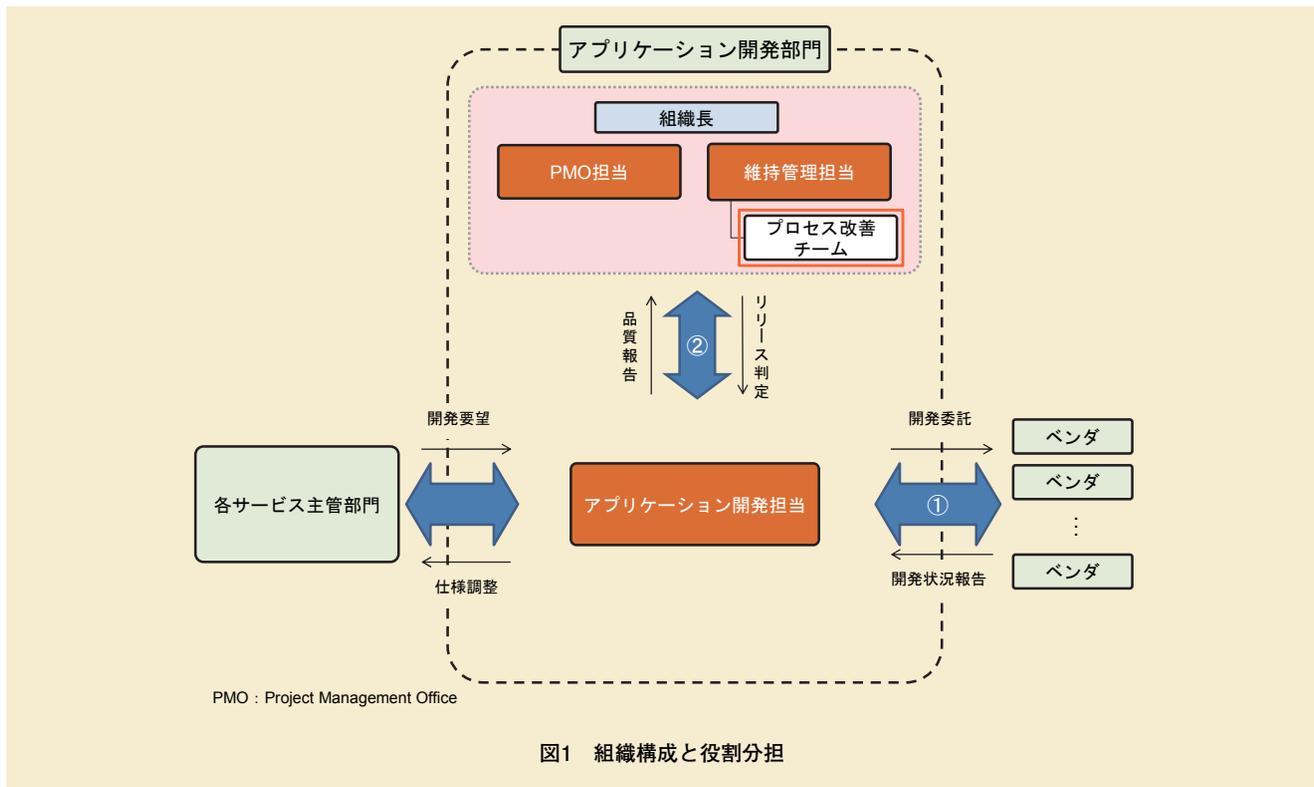


図1 組織構成と役割分担

理方法が統一性なく属人的であったため、ベンダからドコモへの開発状況報告は、定量的・客観的な報告と定性的・主観的な報告が混在し、詳細度の観点からばらつきが大きかった。そのため、開発現場によっては、ベンダの開発状況や潜在リスクをドコモ側が正確に把握できず、適切な対策やリスク予防策が実施されない懸念が生じていたことが大きな課題であった。

また、ベンダから提示される情報の詳細度の相違から、図1の②リリース判定プロセスにおいても品質報告内容にばらつきがあり、客観的な品質判定をリリース判定者が行うためには報告内容を適切に解釈し、報告者に質疑応答しながら不足する情報を補うことが必要であることも

大きな課題であった。

### 3. プロセス改善活動

部内ヒアリングの結果、開発管理プロセスとリリース判定時の品質報告内容にばらつきがある根本原因は、ベンダの開発状況の報告書およびリリース判定時の品質報告書が統一されていないことだけでなく、定量的開発管理プロセスに関する知識・ノウハウを組織的に展開する仕組みが整備されていないことも大きな要因であることが分かった。

そこで、ベンダからアプリケーション開発担当者への報告用の開発状況報告書および、商用リリース判定用の品質報告書の統一様式を制定・導入したうえで、本様式を用いた定量的開発管理手法を部内に普及

していくこととした。

#### 3.1 統一様式の作成と初期導入

##### (1) 開発状況報告書の作成

開発状況報告書の様式作成にあたり、10社以上の多種多様なベンダへの早期適用と管理コストの増加の抑制を考慮し、最小限のメトリクス<sup>\*4</sup>（規模・進捗・品質予実）のみ報告必須とし、また広く普及しているMicrosoft® Excel<sup>®</sup><sup>\*5</sup>ファイルフォーマットを採用した。

開発状況報告書は毎週ベンダからドコモのアプリケーション開発担当者に提示され、毎週の情報共有MTGの場でベンダとアプリケーション開発担当者間で開発状況を共有・透明化することにより、顕在化したリス

\*4 **メトリクス**：ソフトウェアおよび開発プロセスの品質を定量的に把握するために定義された測定方法および測定尺度。開発規模やプロセスの実施に要した時間・工数などがある。

\*5 **Microsoft® Excel®**：米国Microsoft Corporationの米国およびその他の国における商標または登録商標。

クおよび潜在リスクに対する双方の対策を議論する材料となる。Excelの機能を最大限活用し、ベンダ・ドコモ双方でリスクの認識漏れがないよう、懸念箇所を赤や黄などの色の網掛けで自動的にアラート表示し、入力規則や条件式などによる入力ミスのチェックや記入必須箇所の網掛け表示などにより、ベンダが入力する際に迷わないよう工夫した。

様式の構成は①サマリ部、②開発機能・規模管理部、③進捗予実管理\*6

部、④品質予実管理部に分かれており、目的ごとにシートを分けた構成としている。図2に開発状況報告書におけるサマリ部と品質予実管理部を示す。その他については文献[1]を参照されたい。

- ①サマリ部は有限の情報共有MTGを効率的に行うため、サマリ部を見るだけでその時点における開発状況とリスク概要を共有できる体裁とした。具体的には、規模・進捗・品質データのダイ

ジェスト情報と現状の課題・ベンダ側のアクションなどを掲載している (図2(a))。

- ②開発機能・規模管理部は、適切な管理単位\*7に分割した開発機能一覧と各管理単位の開発計画時および各開発工程終了時点の開発規模を管理している。規模の推移を確認することで、開発リスクを把握できる。
- ③進捗予実管理部は管理単位ごとの進捗予実を管理し、遅れが生

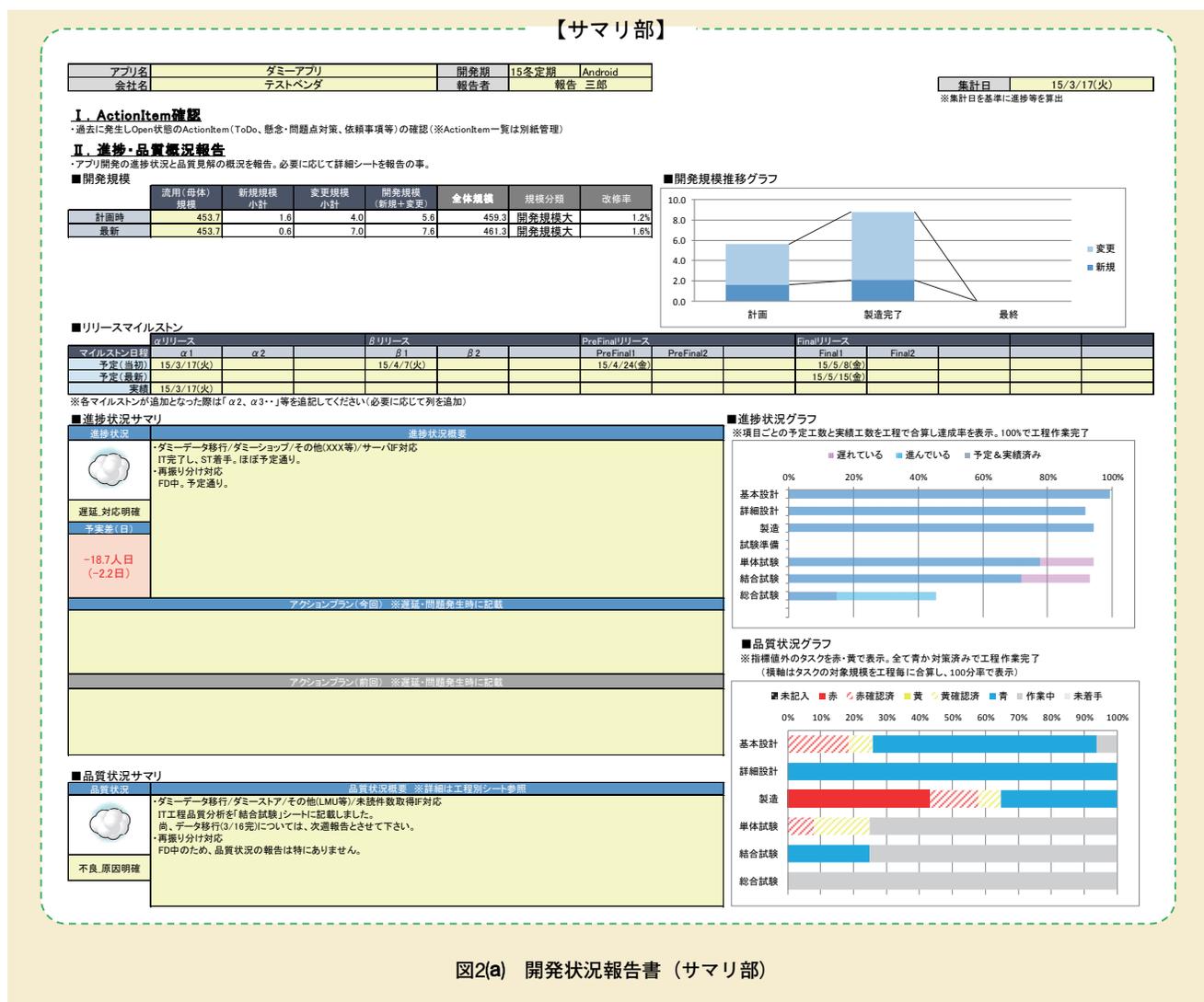


図2(a) 開発状況報告書 (サマリ部)

\*6 予実管理：予定と実績の差異を管理すること。

\*7 管理単位：品質の良し悪しを測定し、必要に応じて品質向上施策を実施するための、ソフトウェアの構成上の単位。

じた機能・開発工程と原因・対処時期を共有可能とする。進捗を見える化する工夫により、進捗リスクを把握できる。

④品質予実管理部は開発工程単位・管理単位ごとにレビュー密度\*8・テスト密度\*9およびレビュー指摘密度\*10・バグ密度\*11の品質指標値の目標と実績および、結合テスト以降の工程にお

ける試験状況を管理する。品質指標値の目標からの乖離に対する分析結果と対策を共有することで品質リスクへの早期対策が可能となる。ベンダ・ドコモ双方で品質リスク対応漏れを防ぐための工夫として、それらをゾーン分析\*12した結果を視覚的に表現している(図2(b))。なお、品質予実管理部の作成にあ

たり文献[2]を参考にした。

また、開発規模が小さくリスクが低いプロジェクト向けに、管理コストと開発管理の詳細度のバランスを考慮した簡略版の開発状況報告書様式を作成した。通常版と簡略版の開発状況報告書は、開発規模・開発費にしきい値を定めたくえで使い分けて運用している。一般的に開発状況

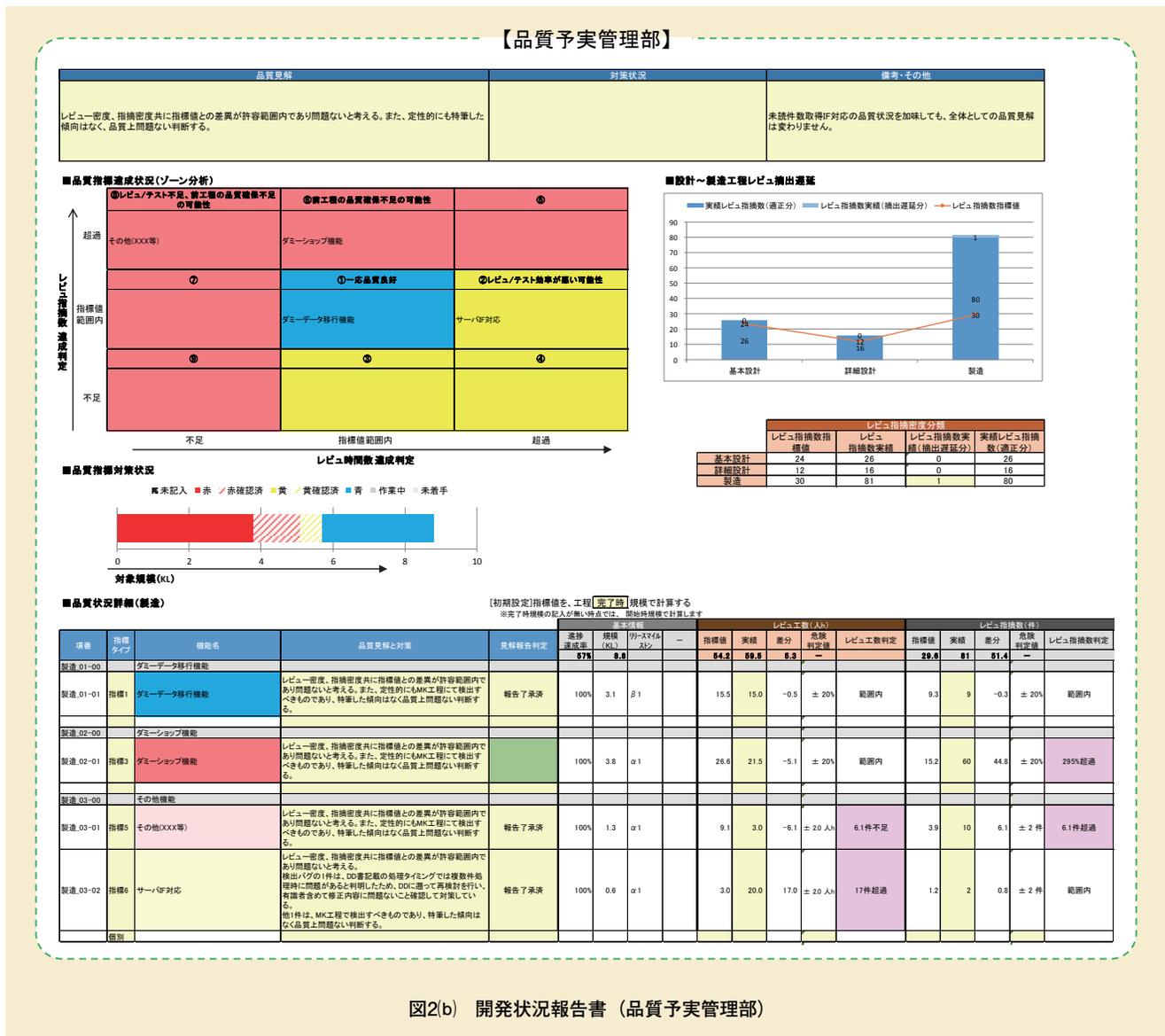


図2(b) 開発状況報告書 (品質予実管理部)

\*8 **レビュー密度**: レビュー対象物の規模あたりのレビュー量。レビュー量の十分性を示す尺度となる。ここでのレビューとは、開発の中間成果物(設計ドキュメントまたはソースコード)を、作成者を含めた複数人で閲読する作業。多角的な視点で欠陥や問

題を抽出することで、成果物の品質を高める効果がある。

\*9 **テスト密度**: プログラム製造規模あたりのテスト項目数。工程ごとのテストの深度を示す尺度となる。ここでのテストとは、作成したプログラムを、コンピュータ上で実

際に動かし、想定通りの結果が得られるかチェックする作業。

の報告粒度が細かく精緻であるほど、リスクの早期発見・対応が可能となる一方で管理コストが高くなる。開発リスクが低いプロジェクトに適用する簡略版の開発状況報告書は報告内容の詳細度よりも管理コストの削減に重点をおき、管理すべき工程の簡素化や品質指標値をドコモが定める簡素な値とすることにより、開発機能・規模・進捗・品質のすべての情報をシート1枚で簡潔に整理した構成については文献[1]を参照されたい。

(2)品質報告書の作成

商用リリース判定用の様式である

品質報告書の作成にあたり、データ収集・加工の容易さと開発状況報告書との親和性から同様にMicrosoft Excelファイルフォーマットを採用した。また、アプリケーション開発担当者の品質報告書の作成稼働増加を抑制しつつリリース判定者が客観的な品質判定ができる最低限のデータを掲載することを考慮した。具体的には品質報告書のソースデータは、アプリケーション開発担当者が開発管理中に使用してきた開発状況報告書および受入れテストのテスト項目数・不具合数のデータのみとし、追加の情報収集を不要とした。また、

開発状況報告書の実績データを掲載することで、リリース判定者は上流工程における品質の作込みプロセスも含めた総合的な品質判定が可能となった。

図3に品質報告書の様式を示す。

品質報告書は、①提供アプリ名、開発規模などの概要を記載した「アプリ概要」、②開発状況報告書を元データとした「開発工程における品質情報」のダイジェスト（簡易版の開発状況報告書のプロジェクトでは省略）、③ベンダの総合テスト以降のテストと社内受入れテストのテスト数・不具合数から成る「信頼度成

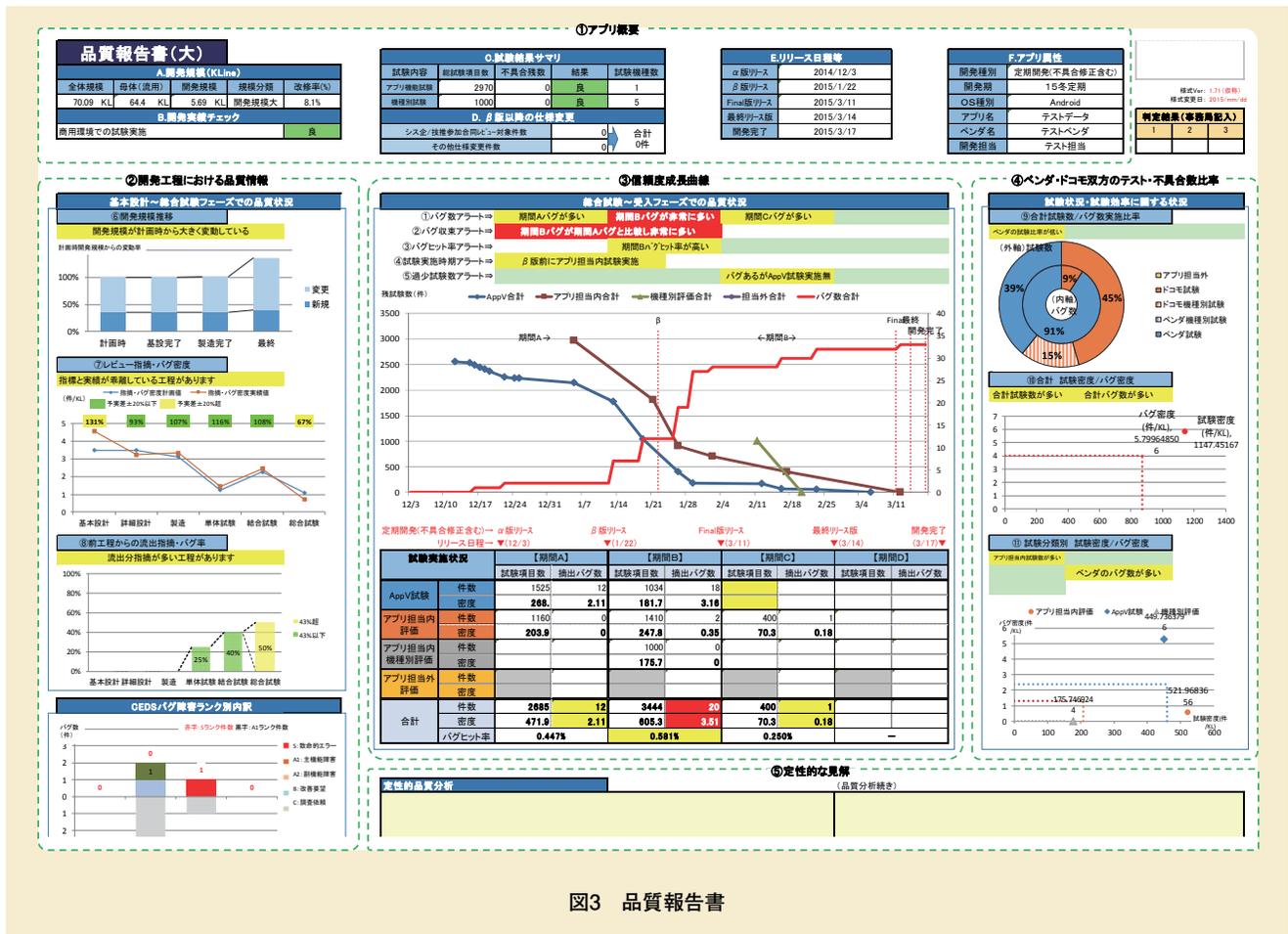


図3 品質報告書

- \*10 レビュー指摘密度：レビュー対象物の規模あたりのレビューによる指摘数。レビューによる問題箇所の指摘の抽出度およびレビュー対象物の品質を判断する尺度となる。
- \*11 バグ密度：プログラム製造規模あたりのバ

- グ検出数。工程ごとのバグの抽出度合いおよびテストの強化または再テストの要否を判断する尺度となる。
- \*12 ゾーン分析：与えられた分析のテーマをある特徴に着目した視点によってゾーンに分割し、ゾーンごとに分析を行う分析手法。

ゾーンに分割することにより、全体に行う施策と比較してきめ細かく対応できるようになる。

長曲線<sup>\*13</sup>], ④「ベンダ・ドコモ双方のテスト・不具合数比率」, ⑤「定性的な見解」で構成される。Excelの機能を最大限活用し、判定者にとって見易く、記入者にとって簡易に記入可能な様式となるよう工夫した。例えば入力規則や条件式などの機能を利用した入力チェックを実現している。また、数値データ入力のみで自動的に視認性の高いグラフを表示し、リスクが大きい品質データに赤色または黄色の網掛けでアラート表示し品質リスクを見える化した。

### (3)本様式の初期導入活動

本様式導入の際には、文献[3]を参考に、トップダウンとボトムアップの両アプローチを使い分けた。

まず、多様な開発現場への早期導入のため、強制力の強いトップダウンアプローチを活用し、全開発現場への一斉導入を進めた。

一方、ボトムアップアプローチとして、プロセス改善チームのメンバーがアプリケーション開発担当者一ベンダ間の情報共有MTGに参加し、導入背景・様式記入方法の説明、アプリケーション開発担当者への例示も兼ねてベンダの記入方法および報告内容に対する確認・指摘を行い、アプリケーション開発担当者・ベンダを含めた開発現場への直接的支援を行うことで、トップダウンアプローチの弊害である強制感や改善取組みへの反発を極力抑えることを心掛けた。

このように両アプローチを使い分けることで、約半年間で全開発現場

に本様式を導入した。

## 3.2 定量的開発管理の定着化

本様式の導入完了後、定量的開発管理手法の定着・継続的改善・高度化に主眼をおき、下記に記す3点の施策を実施した。

### (1)本様式含めたドキュメントのさらなる改善と整備

開発担当者の管理スキルにかかわらず進捗・品質を把握でき、ベンダが記入し易くなるよう、本様式を適宜改版した。その際、公式・非公式にアプリケーション開発担当者およびベンダにヒアリングし、開発現場の意見・提案を積極的に取り入れ、プロセス改善活動への主体的な参加意識を醸成し、改版による変化を受け入れ易いよう考慮した。また、ベンダ向けに開発状況報告書の記入上のガイドラインを、アプリケーション開発担当者向けには、開発状況報告書や品質報告書を読み解き定量的開発管理を実践するためのノウハウ集を作成し、本様式を効果的に利用するための参考ドキュメントを充実させた。

また、開発実績データを蓄積し、品質報告書の品質アラートに過去実績の統計データを活用することで、各プロジェクトをより客観的に品質判定可能とした。例えば、図3の③信頼度成長曲線にて、期間ごとのバグ密度が過去の統計データの75~85パーセント<sup>\*14</sup>の場合に黄アラート、85パーセントを超える場合には品質リスクが高いデータとして赤アラート表示した。

### (2)アプリ開発管理サポートツールの導入

開発現場への本様式の浸透に従い、より効率的に、より精緻に開発管理をしたいという現場のニーズが大きくなり、また、プロセス改善チームメンバーにとって全プロジェクトの主要開発データ（信頼性・生産性・規模・工数など）の収集・集計に伴う負担が大きくなってきた。そこで、アプリケーション開発担当者向けの①「開発管理サポート機能」と、プロセス改善チーム向けの②「データ集計サポート機能」の2種類の機能から構成される「アプリ開発管理サポートツール」を構築した。図4にアプリ開発管理サポートツールの概略図を示す。なお、本ツールはアジャイル開発<sup>\*15</sup>により、プロセス改善の進捗を測りながら柔軟かつ早期に機能追加している。

①「開発管理サポート機能」はアプリケーション開発担当者の開発管理の支援を目的としており、ヒアリングや開発現場の観察を重ねながら、開発管理上の課題を解決する機能を搭載している。具体的には、プロセス改善チームや部内有識者のノウハウを基に開発計画内容を分析し進捗遅延リスク・品質リスク要因を抽出する機能、情報共有MTG効率化のため直近の開発状況報告書との差分を明示する機能、開発状況報告書を基に品質報告書を自動作成する機能、受入テストで発生した不具合情報を分析する機能、当該プロジェクトの

\*13 信頼度成長曲線：プロジェクトの進捗状態・品質状況の確認などに用いるグラフ。横軸に日付やテスト時間またはテストケース数、縦軸に累積バグ発見数をとる。S字の成長曲線を描くことが多い。

\*14 パーセンタイル：計測値の分布（ばらつき）を小さい数字から大きい数字に並べ替え、パーセント表示することによって、小さい数字から大きな数字に並べ変えた計測値においてどこに位置するかを測定する単位。例えば65パーセンタイルであれば、

最小値から数えて65%に位置する値を指す。

\*15 アジャイル開発：アジャイル開発宣言に基づく開発方法論であり、迅速かつ適応的にソフトウェア開発を行う軽量な開発手法群の総称。



図4 アプリ開発管理サポートツール

開発データを分析した開発振り返り用データを提供する機能などが搭載されている。本機能により、開発管理業務が効率化し、開発現場の自発が促され、開発現場に定量的開発管理が定着した。なお、各アプリケーション開発担当者のツール使用履歴ログを自動収集し、ログ分析することで、ツールの有効性と浸透度を容易に検証する仕組みもあわせて構築した。これにより、ツールの普及度が低い開発現場への個別説明や、利用度が低い機能の改善などの早期の意思決定を可能としている。

- ②「データ集計サポート機能」は開発状況報告書に記載されている開発データを収集・蓄積し、整理・出力する機能である。本機能により、多種多様な開発現場におけるデータを容易に集計・蓄積し、統計データとして活用することが可能となった。

(3)メリハリを付けた人的サポート  
ドキュメントの整備とツールによる支援、開発現場の経験・ノウハウの蓄積により、各開発現場の主体的・自立的な定量的開発管理への移行が進んだため、徐々にプロセス改善チームの開発現場への参加頻度を減らすこととした。一方、本様式改版時やアプリ開発管理サポートツールへの新機能搭載時には、部内全体への説明会を開催した。説明会の場で、定量的開発管理の必要性の啓発、開発状況報告書・品質報告書からリスクを分析する具体的なノウハウの紹介、開発管理のベストプラクティス・ワーストプラクティスを共有した。

このように人的サポートの機会を意図的に減らしつつも、プロセス改善の要は“人”であることを意識し、規模や難易度の観点から開発リスクの大きいプロジェクトや担当者入替えなどのリスクが高まる局面では、開発現場への人的サポートを手厚く

行い、メリハリを付けたサポートを実施した。

## 4. プロセス改善活動の成果

これまでのプロセス改善活動の成果を以下に解説する。

### 4.1 開発管理の適正化とリリース判定の客観化

ベンダとアプリケーション開発担当者が客観的・定量的な指標を用いて開発状況を共有し、リスクを互いに認識することで進捗・品質面で早期の対策が可能となった。また、全開発プロジェクトに統一的品質報告書の導入・全プロジェクトの過去の統計品質データを基にした品質アラート条件を設定することで、各開発プロジェクトを時系列かつ横並びで客観的な品質判定が可能となった。進捗管理の改善の具体的な成果を図5に示す。開発状況報告書を導入した全プロジェクトの全情報共有MTGのうち、1日超の進捗遅れを報

告した情報共有MTGの占める割合を進捗遅延発生率と定義する。導入当初の半年間は進捗遅延発生率が55%を超えていたが、経年で改善し、特に直近では20%以下に抑えられ、プロセス改善活動が進捗管理の改善に寄与したことが分かる。

品質面の改善の具体的な成果を図6に示す。2つの箱ひげ図<sup>\*16</sup>は、初期半年間と直近半年間の全開発プロジェクトについて、ベンダの総合テスト以降の全テストにおけるバグ密度のデータを集計したものである。例として、中央値について直近半年間では初期半年間の2/3に低下している。定量的開発管理の浸透により、上流工程での品質の作り込みが促進され、ソフトウェア品質が向上したことが確認できる。

## 4.2 開発データの統計資料の整備

アプリ開発管理サポートツールのデータ集計サポート機能を用いて、全プロジェクトの主要開発データ(信頼性・生産性・規模・工数など)を集計・整理したソフトウェア開発データ白書(部内版)を作成し、部内公開した。内容・構成は独立行政法人情報処理推進機構ソフトウェア高信頼化センター(IPA/SEC: Information-technology Promotion Agency, Japan/Software Reliability Enhancement Center)<sup>\*17</sup>が発行しているソフトウェア開発データ白書[4]を参考とした。ソフトウェア開発データ白書(部内版)は最新データの追加に伴い過去3回改版し、ベンダへ



図5 進捗管理面の改善成果

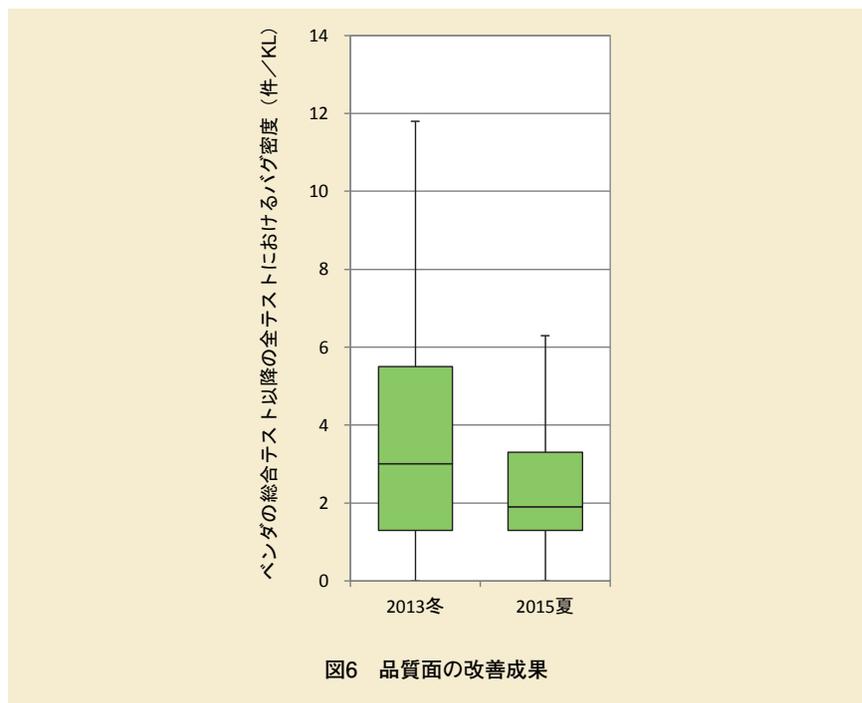


図6 品質面の改善成果

開示可能となるよう再構成した派生版も提供した。ソフトウェア開発データ白書(部内版)はアプリケーション開発担当者のみならず部内管理職にも広く浸透しており、開発完了後の振り返りや開発計画の妥当性確認などに引用され、開発のPDCAサイクルの強化に貢献している。

## 4.3 アプリケーション開発担当者が自立的に開発管理を適正化する仕組みの構築

品質報告書の浸透により、開発管理が適正であれば適切な品質報告書が出力され、逆に開発管理が不適切である場合は品質報告書に品質リスクを示すアラートが多数表示される

<sup>\*16</sup> 箱ひげ図：ばらつきのあるデータをわかりやすく表現するための統計学的グラフ。一般的には最小値、第1四分位点、中央値、第3四分位点および最大値を表現する。その場合、第1四分位点、第2四分位点(中央値)、第3四分位点は“箱”で表現され、最

小値・最大値は箱の両側に出た“ひげ”で表現される。

<sup>\*17</sup> 独立行政法人 情報処理推進機構 ソフトウェア高信頼化センター(IPA/SEC)：ソフトウェア開発における定量的プロジェクト管理の普及促進を目的に、開発プロセス

の標準化や見える化手法、定量的品質管理手法などの調査・検討を行っている組織。

ことがアプリケーション開発担当者に広く理解された。アプリ開発管理サポートツールの開発管理サポート機能の提供により、アプリケーション開発担当者は開発の途中段階で品質報告書を簡易な操作で即座に自動出力可能となったため、開発中に品質報告書のアラートを意識し、ベンダの進捗・品質報告に対してリスクを見極め、適切な追加対策を議論するような意識に変わり、開発現場が自立的に適切な開発管理を行うようになった。

#### 4.4 アプリ開発管理サポートツールによるシステム的な開発管理の浸透

アプリ開発管理サポートツールの利用ログの解析や開発現場へのヒアリングを重ねながら、継続的な機能改善を行い、説明会などによるアプリケーション開発担当者へのメリット理解の浸透などにより、ツールの利用者が拡大した。導入当初の3カ月後は想定ユーザの約2割であったが、導入の半年後には想定ユーザの約8割に拡大し、システム的な開発

管理が浸透した。ツールの利用が浸透し続けた結果、アプリケーション開発者の全員がツールを認知し利用している。

## 5. あとがき

本稿では、スマートフォン向けアプリケーション開発におけるプロセス改善活動について解説した。

マルチベンダへの適用を考慮した開発状況報告書と品質報告書の統一様式を作成し、早期に様式を定着化した。また、ドキュメント・ツールの整備・効率的な人的サポートにより、定量的開発管理プロセスが部内で定着した。これらプロセス改善活動の結果、開発管理の質が大きく向上し、進捗遅延の減少やソフトウェア品質の向上などの成果が得られた。

また、約2年間のプロセス改善活動を整理し、本成果をソフトウェア品質シンポジウム<sup>\*18</sup>にて発表した[1]。特に多種多様なベンダへの適用を考慮した様式はドコモならではの特色ある発表内容であり、本様式の詳細内容に対する質問も受け、参加者の高い関心を集めた。また、発注元企

業の立場での対外発表事例は珍しく、同様の発注元企業およびベンダにとっても知の共有につながり、ソフトウェア開発業界の発展の一助となった。

今後は、蓄積データ間の高度連携、管理スキル・意識のさらなる底上げ、およびアジャイル開発プロセスへの対応について、改善すべき課題として取り組む予定である。

## 文献

- [1] 戸崎 貴資, 山田 善大, 服部 弘幸: “発注元企業における開発プロセス改善活動,” ソフトウェア品質シンポジウム, 2015.
- [2] 独立行政法人情報処理推進機構ソフトウェア・エンジニアリングセンター編: “定量的品質予測のススメ,” SEC BOOKS, 2008.
- [3] 堀田 勝美, 関 弘充, 宮崎 幸生: “ソフトウェア品質保証システムの構築と実践,” ソフト・リサーチ・センター, 2008.
- [4] 独立行政法人情報処理推進機構: “ソフトウェア開発データ白書,” SEC BOOKS.
- [5] ソフトウェア品質シンポジウムホームページ.  
<http://www.juse.jp/sqip/symposium/>

\*18 ソフトウェア品質シンポジウム: 日本におけるソフトウェア品質に関する最大級のイベントであり、ソフトウェア品質に関する実践的な技術・経験・研究成果を共有し、意見交換が行われている。著名人による講演・パネルディスカッションのほか、一般

参加者からの発表も受け付けている。