

インキュベーション型プロダクトに対応した ソフトウェア開発リスクマネジメント

Risk management of software development for R&D

NTT ソフトウェアイノベーションセンタ
NTT Corporation, NTT Software Innovation Center
○秦泉寺 久美¹⁾ 羽室 大介²⁾
○Kumi Jinzenji¹⁾ Daisuke Hamuro²⁾

Abstract The software product of research and development (R&D) are used in various conditions, from incubation to commercial. Not only product type but also its quality divers. The crucial difference between incubation type products and commercial products is that the latter can uniformly define the release quality standards for business while the former could not set up unique quality standards. Under such circumstances, R&D organizations are accountable for product towards business companies which adopt R&D products. In this paper, we propose software development risk management scheme based on software development standards featuring software quality diversity.

1. はじめに

研究開発によって生みだされた成果はプロダクトとして事業へ提供(プロダクトアウト)され、そのまま利用される、あるいは機能追加されるなどしてサービスへ導入されていく。研究開発プロダクト(以降、プロダクト)は事業サイドの要求を実現するニーズ志向のものもあるが、将来を見据えたインキュベーション(シーズ志向)型も多数存在する。また、プロダクトの事業での利用条件は、単純なデモンストレーション用途から商用としてサービスに導入されるものまで多様である。さらに、プロダクトの種類においては業務アプリケーション、メディア処理エンジン、OSS、ミドルウェア、組み込みシステムなど多種多様である。このような研究開発プロダクト特有の状況下においても、開発する過程で予算や期間超過、予定した機能が実現されない、逆に余計な機能・非機能を作りこむリスク、事業サイドへ提供後に品質面で問題を生じるリスクは存在し、安定した開発の遂行と事業サイドとのトラブルの未然防止が課題であった。インキュベーション型プロダクトと商用プロダクトとの決定的な差異は、後者が事業へのリリース基準を一律に定義しやすいのに対し、前者はさまざまなプロダクト品質を是としているため、一律な品質基準を定められない点にある。本稿では、こうした課題下でのソフトウェアリスク回避施策の制度化と実施効果について述べる。また、施策の確実な運用方法と形骸化防止、施策の推移についても述べる。なお、本稿で述べるプロダクトはすべてソフトウェアプロダクトである。

2. インキュベーション型プロダクトのためのソフトウェアリスク回避施策

多種多様な利用状況、プロダクトの種類に加えて、インキュベーション型のプロダクトが多く

NTT ソフトウェアイノベーションセンタ ソフトウェア開発技術プロジェクト

Software Engineering Project, Software Innovation Center, NTT Corporation

東京都港区芝浦3-4-1 グランパークタワー33F Tel: 03-6712-9315 e-mail: kumi.jinzenji.hz@hco.ntt.co.jp
Granpark Tower 33F, 3-4-1 Shibaura, Minato-ku, Tokyo Japan

1)NTT ソフトウェアイノベーションセンタ ソフトウェア開発技術プロジェクト 主任研究員
Dr. Sci., Senior Research Engineer, Software Engineering Project, Software Innovation Center, NTT Corp.

2)NTT ソフトウェアイノベーションセンタ ソフトウェア開発技術プロジェクト プロジェクト
マネージャ Project Manager, Software Engineering Project, Software Innovation Center, NTT Corp.

【キーワード：】ソフトウェア品質、リスク管理、制度設計、インキュベーション

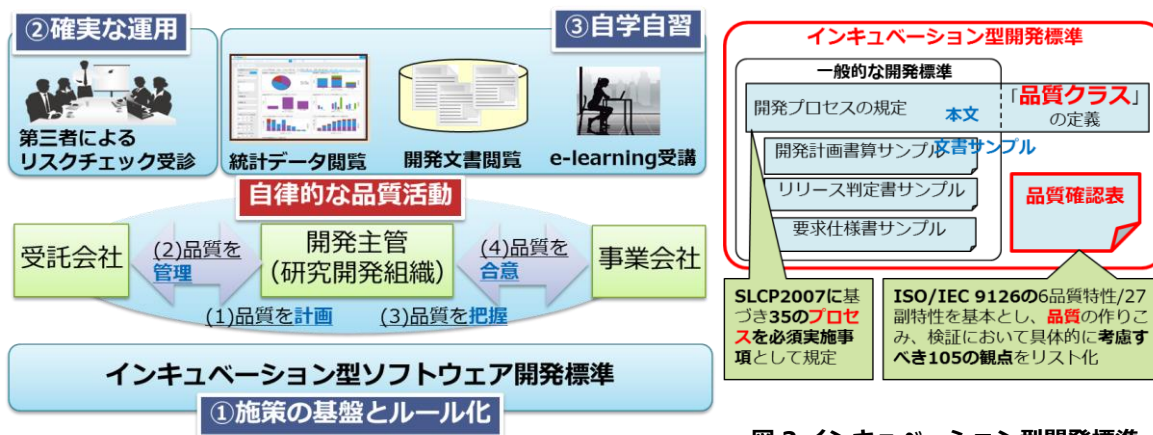


図1 リスクマネジメント施策の全体像

図2 インキュベーション型開発標準

表1 品質クラスの定義

品質クラス	サービス導入の可能性からみた定義
A	事業会社がほぼそのままサービス導入でき、インフラやネットワークサービスのようにシステムを止めてはいけないような極めて厳しいSLA(Service Level Agreement)下で使用されることを想定したプロダクト
B	事業会社がほぼそのままサービス導入できるが、パッケージやソリューションのようにアプリケーションソフトウェアの再起動がある程度許容される状況下で使用されることを想定したプロダクト
C	実証実験のように使い方が一部限定され、事業会社がサービス導入するには機能・非機能要件やアーキテクチャの一部の見直しと一部の追加試験などを要するプロダクト
D	機能のデモンストレーションのように使い方が極めて限定され、事業会社がサービス事業導入するには、機能・非機能要件およびアーキテクチャ、試験などについて抜本的な見直しを要するプロダクト
E	品質見解が無いプロダクト

を占めることから各種トラブルは以下に起因していると考えた。

- 開発主管（研究所）が想定していない使い方を事業サイドがすること
- 開発主管が品質の作りこみと検証の度合いを把握していないこと

そのため、利用条件を明確化し、品質の作りこみと検証の度合いを念頭においた開発をすることを徹底する必要があった。ここでの品質とは狭義の品質である信頼性だけでなく、ISO/IEC 9126[1]で述べられている6つの品質特性を含む広義の品質である。筆者らはこれらの課題解決へのアプローチとしてインキュベーション型プロダクトの開発・事業導入のためのリスクマネジメント施策と確実な運用のための制度設計を行った[2][3]。図1に施策の全体像を示す。当施策は以下の3つパートから成る。

- 1) 開発主管組織が品質責任を負うための開発標準の導入とルール化
- 2) 施策の確実な運用のためのシステムを用いた案件管理と第三者によるリスクチェック
- 3) 開発主管の自学自習を目的とした集合知共有、開発文書公開と e-learning

なお、NTT の研究開発組織におけるソフトウェア開発の多くが受託開発であり、研究開発組織は開発主管として発注側に立つ。

2.1 開発主管組織が品質責任を負うための開発標準の導入とルール化

一般的なソフトウェア開発標準はソフトウェア開発に標準的なプロセス/タスクを分類・定義し、それに基づいて人に依存しないソフトウェア開発を目的とする。筆者らは図2に示すようにSLCP2007[3]の一般的な開発プロセスに加え、「品質クラス」の概念と品質クラスを実現するための具体的な観点を一覧化した「品質確認表」を導入した。

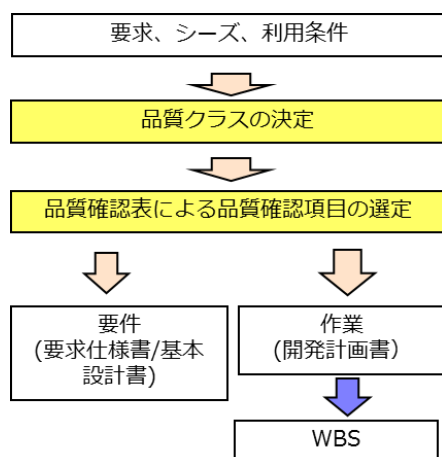


図3 開発主管での品質計画フロー

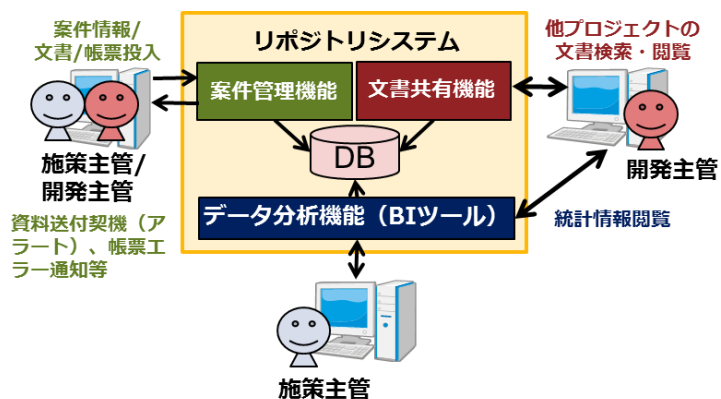


図4 システムによる運用

品質クラスの定義を表1に示す。品質クラスとは「想定される利用方法」による5段階のレベルであり、プロダクトをそのまま事業導入できるかによって二つに大別できる。品質クラスA、Bはそのまま事業導入できる品質である。品質クラスAは厳格なSLA(Service Level Agreement)を遵守し、停止を許容しない。一方、品質クラスBはリスタートを許容する。品質クラスC、D、Eはそのままでは事業導入できない品質である。品質クラスCは使い方を限定した利用条件下での実証実験を想定したクラスである。一方、品質クラスDはデモンストレーションを想定したクラスである。また、これら品質クラスA~Dは必ず品質見解をもつこととし、品質見解の無いプロダクトはすべて品質クラスEと位置付けた。

品質確認表は、各品質クラスを実現するための具体的な品質観点としてISO/IEC 9126[1]の6品質特性/27副特性を105の具体的な品質確認項目にブレークダウンし、確認観点と各品質クラスの推奨基準とともにチェックリスト化したものである。表2に品質確認表における品質確認項目の一部を示す(推奨基準と確認観点は割愛)。図4に開発主管における具体的な品質計画フローを示す。開発主管は要求、シーズと共にプロダクトの利用条件を想定し、品質クラスを決定する。次に品質確認表を用い、105の品質確認項目のどれを品質に含めるかを決定する。選択された品質確認項目に基づき、具体的な作業(テスト、レビューなど)と要件(機能/非機能)に落としこみ、開発計画書と要求仕様書/基本設計書に纏め上げる。このように、品質を計画し把握することは、スコープ(要件、作業)を計画し把握することに他ならない。また、開発計画書、要求仕様書、リリース判定書を主要三文書と位置付け、上位品質クラス(A、B)向けの雛型を開発標準付録として提供した。それにより、下位品質クラス(C、D)にて開発をするときに、雛型から考慮しない要件、作業に関する記述を削除していくが、削除した分がリスクになることを認識する効果を狙っている。

また、プロダクトアウトを想定して開発をする時は、開発主管が品質を計画し、管理し、把握し、提供前には事業と品質を合意するという一連の品質活動が実施されることを狙いとし、以下の2つをルール化した。

- プロダクトアウト想定した開発は開発標準を適用し、品質クラスA~Dを宣言すること
- 要件確定時およびリリース判定時に所属組織長がプロジェクトレビューを実施すること

このように、インキュベーション型プロダクトにおいてはリリースするための一律な品質基準を定めることができないという課題を、開発主管の自律的品質活動とプロダクトの責任を開発主管組織が負うことで解決を図った。

2.2 施策の確実な運用のためシステムを用いた案件管理と第三者によるリスクチェック

施策遵守のために、要件確定後、リリース判定後、リリース後指定期間経過後に施策主管(筆者ら)がモニタリングした。但し、施策主管は一般的な品質管理部門と異なり、品質ゲートのと

しての役割はない。モニタリングには開発標準適用の証跡として開発主管から主要三文書を中心とした開発文書を施策主管に提出する形態をとった。図4に示すように運用のためにシステムを構築し管理を行った。「案件管理機能」を設け、プロダクトアウトをオーソライズする会議主管からの情報をシステムにあらかじめ入力し、会議体の日付を基準として特定の日数の経過後に開発主管にアラートメールを自動発出する。アラートメールが所定回数を越えた場合、開発主管組織長にエスカレーションメールが送信される。さらに定期的な上位会議体への報告を行い、より確実な運用を実施した。

要件確定後およびリリース判定後については、開発主管から提出された主要三文書群に対して施策主管が第三者的にリスクチェックを実施した。要件確定時のリスクは70超の項目を3段階で評価、リリース判定後のリスクは19の項目を5段階で評価している。リスクが高いと思われる案件については施策主管が開発主管に対してフォローアップを実施した。表3にリリース判定後リスクチェックの19の項目を記す。(チェック観点の詳細については本稿では割愛する。)

2.3 開発主管の自学自習を目的とした集合知共有、開発文書公開と e-learning

モニタリングのために提出された開発文書群をリポジトリシステムの「文書共有機能」上に公開し、アクセス制限付きで開発主管が閲覧することを可能にした。文書提供者である開発主管への情報還元だけでなく、開発文書雛型や一時的な研修では得られない他者の経験そのものを生きた教材にすることを狙いとしている。文書共有機能は検索機能を持たせ、開発費用、品質クラス、組織、開発期間3節で述べるリスクレベル、品質レベルなどでプロダクトの検索が可能である。また、Excel 帳票のあらゆるデータはBI ツールによって集約され、集合知として開発主管に対し

表2 品質確認表における確認項目(抜粋)

品質特性	品質副特性	確認項目	品質特性	品質副特性	確認項目	
信頼性	成熟性(試験の妥当性)	①試験項目数	保守性	解析性	①マニュアル(保守・解析)	
		②ホワイトボックス試験(カバレッジ)			②進行状況の確認	
		③ブラックボックス試験			③障害時の処理履歴	
		④非機能試験/システム試験			④解析情報のダンプ	
	成熟性(障害収束性)	①収束性	変更性	①変更対応容易性		
障害許容性	回復性	②障害摘出	②変更内容確認容易性			
		③制限事項	③プログラムやデータの独立性			
		④ファイル破壊防止	安定性	①変更に対する自身への影響		
		①エラーチェック	②変更に対する他への影響			
使用性	理解性	②エラー時の継続運用	③変更のキャンセル			
		③重要入力の確認	試験性	①確認試験要領		
		④ファイル破壊防止		②回避性		
		①UNDO/REDOの実施		③確認性		
	②チェックポイントの構築	④自己診断機能				
	③ソフトウェア閉塞	⑤テスト・シミュレーション機能				
	習得性	運用性	④障害解析用の記録	移植性	適応性	①検証実績
			①チュートリアル			②多言語対応
			②デモ			③OSSなどの導入
			③操作メニュー		設置性	①マニュアル(導入)
④マニュアル(障害時)			②インストール機能			
①マニュアル(通常操作時)			③留意事項の明確化			
魅力性	資源効率性	②ユーザレベル	共存性	置換性	①共存できないソフトウェア	
		③ヘルプ機能			②共存条件	
		①デフォルト設定			③共存するための変更	
		②ガイダンス指示			④データへの排他アクセス	
		③操作の確認			⑤データ運用の矛盾なき統合	
④障害の通報	時間効率性	⑤運用状況の記録	①データ移行			
①魅力的なユーザインタフェース		②移行前共存ソフトウェアへの非影響				
②魅力的な機能		③移行前ソフトへの互換性				
③魅力的な非機能		④移行前共存ソフトウェアへのユーザインタフェース維持				
①応答時間		⑤移行前システムへの復旧				
②スループット	資源効率性	④過負荷状態での効率性	①メモリ			
③ターンアラウンド		②CPU				
⑤縮退運転時の効率性		③HDD				
④過負荷状態での効率性		④縮退運転時の効率性				

表3 リスクチェック表におけるチェック観点

チェック観点		
1プロダクトの範囲	1.1要件定義(基本設計)の明確性	機能要件 非機能要件 外部インタフェース要件 システム/ソフトウェア構成の要件 データ要件
2開発プロセス	2.1開発プロセス定義の妥当性	開発プロセスの定義
	2.2開発プロセス実施の妥当性	開発プロセスの遂行
3品質の検証と分析	3.1品質作り込みの妥当性	要件定義の品質確保 コーディングの品質確保
	3.2試験準備の妥当性	試験環境
	3.3試験実施結果の妥当性	単体試験による検証 結合試験による検証
	3.4バグ抽出と修正の妥当性	バグ修正や機能追加等による影響有無の検証(回帰試験) システム試験による検証 抽出バグの解析と展開
4品質評価	4.1品質評価の妥当性	抽出バグ数の妥当性 目標品質の達成状況 プロダクトのリスクと制限事項 総合評価の結論

て常時公開されている。コスト情報、開発中（要件確定時、リリース判定時）およびリリース後に生じた不具合状況、利用状況が可視化されている。一方、さらなる自学自習の取り組みとして、これまでのリスクチェックおよびフォローアップで得られた知見を基に e-learning を構築し提供している。施策の目的、品質の考え方、スコープ管理、開発管理、開発方法論（ウォータフォール、アジャイル）をリスクマネジメントの観点に紐付けることで施策を補強している。

3. 実施結果

本節では本施策および制度設計において、下記3つの観点から効果を考察する。なお、評価対象は、施策開始直後の2014年7月1日から2018年6月30日までの4年間にリリースされたプロダクトである。品質クラスA～Dのプロダクト数比率は、12.5%、40.2%、26.7%、20.6%であった。

- リスクマネジメントの達成状況
- システム導入による確実な運用状況
- 自学自習状況

3.1 リスクマネジメントの達成状況

品質確認表を用いた品質観点に対するセルフチェック結果と主要三文書からの第三者によるリスクチェック結果は数値化されており、この数値を可視化することで研究開発組織全体のソフトウェア開発の様子が見えてくる。ここでは品質レベルとリリース判定時のリスクレベルの2つの指標を用いて評価する。以下、品質レベルとリスクレベルの算出方法を記す。

(1) 品質レベルの算出：

それぞれの品質観点項目において、どれだけ品質を考慮するかを0～3までの数値で表現する。これを便宜上「品質考慮値」と呼ぶ。3は最大限考慮する、0は全く考慮しない、その他は中間値とする。品質考慮値を105項目分足し合わせて1項目あたりの平均値を算出したものを品質レベルと呼ぶ。105項目すべてを最大限考慮する場合に3.0となり、どの品質観点も考慮しない場合が0となる。品質レベルは包括的な値ではあるが、プロダクトの品質の考慮度合いを直感的に把握できる。

(2) リスクレベルの算出：

リリース判定後のリスクは19の観点に対し、5:全く記載が無い、4:なんらかの実施した形跡がある、3:数値的な記載がある、2:数値に対して見解がある、1:見解が妥当である、という観点で評価した。これらを便宜上「リスク値」と呼ぶ。リスク値の総計を項目数19で除した値をプロダクトのリスクレベルとした。すべての項目のリスク値が1の場合はリスクレベルが1となる。また、すべての項目のリスク値が5の場合はリスクレベルが5となる。品

質レベルと同様、包括的な指標ではあるが、リスクを直感的に把握できる。

図5に評価期間にプロダクトアウトされた案件の各確品質確認項目の品質考慮値の平均を品質クラス毎に示す(本来は棒グラフで表すべきだが、4つの品質クラス毎比較の視認性を向上するために折れ線グラフを用いた)。一部逆転現象があるものの、各品質特性における品質考慮値はおおむね品質クラスA、B、C、Dの順に高い。図6に案件毎の品質レベル(自己評価結果)とリスクレベル(第三者評価結果)の散布図を示す。品質レベル(品質考慮度合い)が高いとリスクレベルは低いという結果が可視化された。また、適度に分散していることから、プロダクト毎に必要な非機能要件を検討し、プロセスを選択している様子が伺える。図7にリスクレベルの全プロダクトの平均値の4年間の推移を示す。施策開始から2年目で各品質クラスにリスクレベルの上昇が見られるが、その後減少傾向にある。高品質クラスで平均”2”を下回っており、各リスクチェック項目において平均的に何らかの品質見解が得られている様子が分かる。このように、開発においてはリスクマネジメントが実施されていると考えられる。

一方、リリース後の状況からリスクマネジメントの結果を論じる。ここでは2014年7月1日～2017年6月30日の3年間にリリースされたプロダクトを対象とする。図8にリリース後1年間

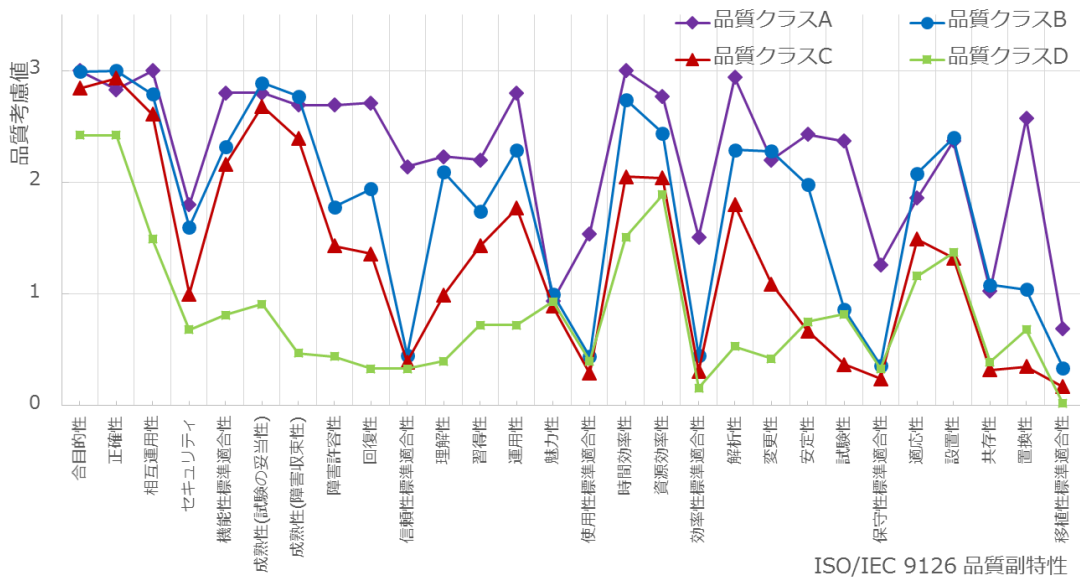


図5 品質クラス毎の平均的な品質の考慮状況

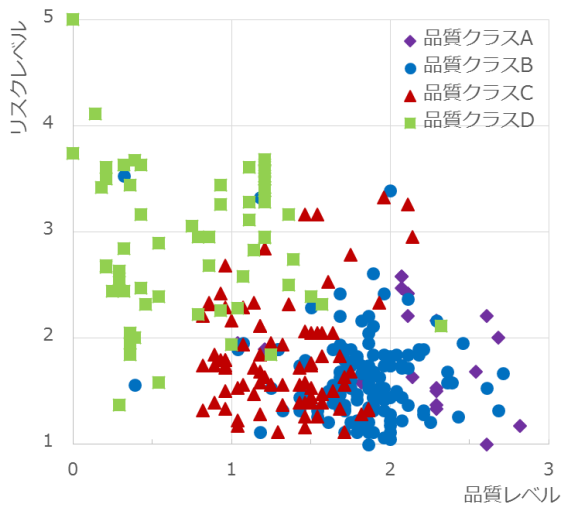


図6 品質レベル vs リスクレベル

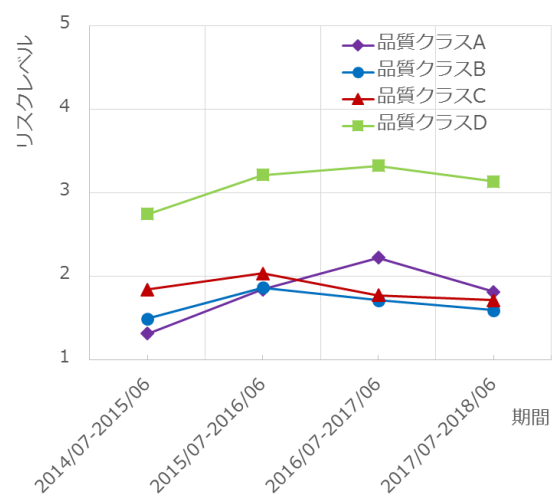


図7 リスクレベル平均値の経年変化

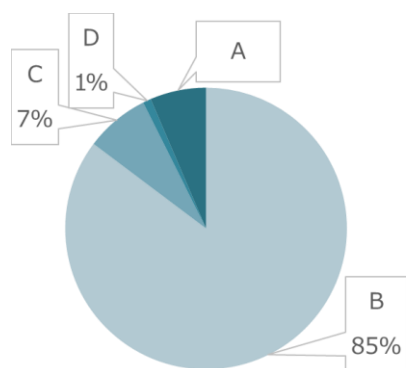


図 8 問題処理票の割合

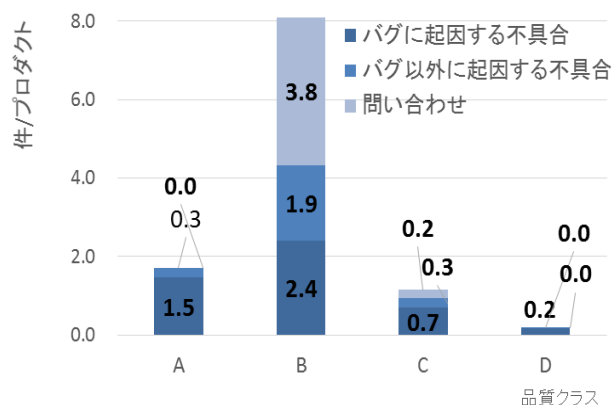


図 9 品質クラス別問題処理票内訳

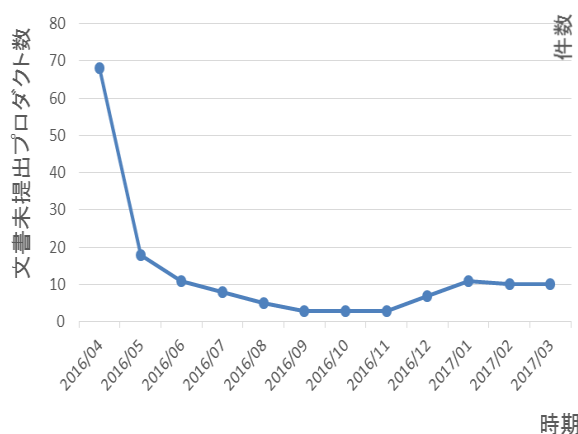


図 10 文書未提出プロダクト件数の推移

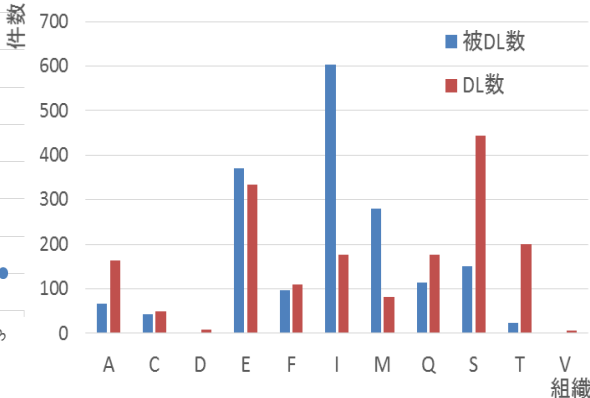


図 11 文書ダウンロード状況

に報告された問題処理票件数のクラス別割合を記す。また、図 9 に品質クラス別の問題処理票の内訳を 1 案件あたりの件数で示す。内訳は①バグに起因する不具合、②バグ以外に起因する不具合、③問い合わせの 3 つである。問題処理票件数については総件数、1 プロダクトあたり件数とも品質クラス B が突出している。ここで②バグ以外に起因する不具合に着目すると（③問い合わせを除くと）品質クラス B の不具合事象の半数が相当する。本不具合は、インフラ設定ミス、性能、他社製品に起因するものであり、非機能に含まれるものである。当社における品質クラス A は基本的にネットワーク案件であり、長年の開発経験から比較的非機能要件が見えやすいこと、品質クラス C, D においては利用方法を厳密に限っていることが想定されるため、非機能における不具合の指摘が少なかったと推測する。一方で、品質クラス B はこれからのサービスにつながる事案であり、非機能における研究開発組織と事業サイドとの意識あわせ、すなわち品質の作りこみと検証における「合意」が今後いっそう必要になると考える。

3.2 システム導入による確実な運用状況

2016 年 4 月にシステムを導入する以前は、文書未提出プロダクトに対して開発主管、研究主管、オーソライズ会議主管との間で直接問い合わせや確認をしていた。しかし、全組織対応するためには無視できない稼働にかかる上に資料の提出遅延も多かった。しかし、当システムを導入することにより 1500 人時/年間の稼働削減が実現されており、資料提出遅延件数も劇的に減少した。図 10 にエスカレーションにまで至った案件の推移を示す。システム導入直後では約 70 件がエスカレーション対象であったが、導入後 3 ヶ月で約 10 件前後に落ち着き、システム導入の効果が得られている。なお、エスカレーション総数は全てのモニタリングタイミングでの総計である。

3.3 自学自習状況

開発標準対象になったプロダクトの約 5000 種類の開発文書がアップロードされ、約 1800 文書のダウンロードが確認された。図 11 に文書のダウンロード状況を示す。ダウンロードされる組織とダウンロードする組織で比較すると、前者の上位が組織 I、E、M であるが、後者の上位は組織 S、E、T である。必ずしもダウンロードする側・される側が同じ順位にはならず、開発経験の少ない組織が開発に慣れている組織の経験(=他人の経験)をうまく利用していることがわかる。e-learning については 2018 年度上期より開始しており、現在効果測定中である。

4. 結論と今後の展開

2012 年 7 月に一部の組織において施策の運用が開始され、2014 年 4 月から NTT の研究開発組織全体の施策となって 4 年が経過し現在に至る。現在は運用が定着し、適用率はほぼ 100%を達成している。開発主管は開発標準を適用することで、品質の定義、計画、把握、提供先との合意という一連の品質活動を行い、開発主管組織長がプロダクトの品質とリスクの責任をもつという流れを形成できている。また、開発主管は他プロジェクトの経験と e-learning によって自学自習し、また開発という実践の場で開発作法を取得できている。その結果、開発が安定し、研究開発組織と事業サイド間で大問題化するようなトラブルは露見していない。一方で、今後新しいサービスにつながるプロダクトでは品質の合意がいっそう必要になると考える。

現在の NTT 研究開発組織でのプロダクトの開発はウォーターフォール型の開発が主流ではあるが、今後はアジャイル開発の増加が見込まれる。本稿で提示した品質の作りこみと把握の考え方は開発方法論によらず適用できる。リリース間隔を短くする代わりにプロジェクトスコープを縮小することが予想されるが、これはデリバリー期間に対して品質を下げることに他ならず、インキュベーション型開発標準における品質の考え方が今まで以上に重要になると考える。一方で、高頻度のリリースに耐えうるべく品質の検証結果を自動で取得・評価するなどの工夫が必要になると考える。

参考文献

- [1] ISO/IEC 9126 “Software engineering- Product quality- “
- [2] K. Jinzenji, N. Kasahara, T. Muraki, “R&D Software Development Standards and their Operation,” NTT Technical Review Jan. 2014 Vol. 12 No. 1
- [3] 秦泉寺、星野、 “インキュベーションのためのソフトウェア開発要領、” NTT 技術ジャーナル、2011 年 8 月。
- [4] “目標品質に基づくソフト開発標準を策定/NTT 研究所のプロダクトアウトを効率化、” ビジネスコミュニケーション 2014 Vol. 51 No. 9。
- [5] SLCP-JCF2007、” 共通フレーム 2007 “