

9. ソフトウェア品質会計 における品質要求と評価

誉田直美 (日本電気 (株))

ソフトウェア品質会計

ソフトウェア品質会計 (以降、「品質会計」と記す) とは、NECのIT製品向けのOSおよび汎用ミドルソフトウェア製品を開発する部門で考案されたソフトウェア品質管理手法である。本稿では、品質要求と評価の具体的実践事例として、ソフトウェア品質会計における品質要求と評価について解説する。

品質会計の語源は、バグを負債とみなすことからきている。負債は利子で膨らまないうちに早期に返済するほうが経済的である。ソフトウェアのバグも同様に、1件のバグが複数のバグを生まないうちに、設計・製造で作り込まれたバグを早期にレビューやテストで摘出すると後戻りが少なくすむ。すべての負債を返済したとき、すなわちすべてのバグを摘出したとき、そのソフトウェアを出荷する。

この考え方を実現するため、品質会計は、バグ件数を主軸とした摘出目標管理により、開発するソフトウェアの品質作り込み状況を管理する。管理精度をあげるため、品質会計では、開発中に作り込まれ

るバグ件数を常に精度よい推定値に保持する必要がある。そのために品質会計が準備する技法を表-1に示す。品質会計は「バグ目標管理技法」および「バグ予測・見直し技法」から構成される。バグ目標管理技法は、「上工程品質会計」および「テスト工程品質会計」の2種類があり、これらは品質会計の中心軸をなす管理技法である。バグ予測・見直し技法には、5種類の技法が含まれており、開発中に発生するさまざまな事象に対応して予測バグ件数を見直すことができるようになっている。

品質会計で想定する開発プロセスはウォーターフォールモデルである (図-1 参照)。基本設計からコーディングまでを上工程と呼び、単体テストからシステムテストまでをテスト工程と呼ぶ。上工程の各工程においては、「設計」と「レビュー」という2つの作業を実施する。当該工程で当該工程のバグをバグとして認識し、計測開始するのは、当該工程の「レビュー」以降である。品質会計は、開発の進行に応じて、表-1に示した品質会計の各技法を組み合わせ適用する (図-2 参照)。

品質会計の基本的な考え方は、「バグは作り込まない。作り込んだバグは素早く摘出する」である。この考え方のもと、上工程では「作り込んだバグは次の工程までに摘出する」ことを、テスト工程では、「作り込んだバグは、すべて摘出してから出荷する」ことを原則とする。品質会計では、上工程におけるレビューでのバグ摘出を重要視しており、そのための目標が「上

カテゴリ	技法	使用法と特徴
バグ目標管理技法	上工程品質会計	上工程 (設計~製造) 用のバグ目標管理技法。バグの摘出工程と作り込み工程の両面からバグを目標管理する。
	テスト工程品質会計	テスト工程用のバグ目標管理技法。テスト開始時に残存する、プログラム全体の総バグ数を目標管理する。
バグ予測・見直し技法	帰帰型バグ予測モデル	開発開始時に、今回の開発で作り込むであろう総バグ数を予測するためのバグ予測技法。
	品質判定表	開発途中に発生する変化を考慮して、バグ目標値を見直すバグ目標値見直し技法。上工程品質判定表とテスト工程品質判定表がある。
	バグ傾向分析	摘出したバグをさまざまな観点から整理することにより、バグの摘出傾向に偏りが無いかを分析する技法。
	バグ分析と1+n施策	バグ1件ごとに真因を分析することにより、開発上の細かい抜け・漏れを発見し、その抜け・漏れに対して、集中的なレビューやテストにより残存するバグを摘出する技法。バグ分析と1+n施策はセットで用いる。
	バグ収束判定	テスト度合いに対する累積摘出バグ数の推移により、バグ収束を判定する技法。

表-1 品質会計の各技法とその特徴

9. ソフトウェア品質会計における品質要求と評価

工程バグ摘出率 80%」である。上工程バグ摘出率とは、出荷前までの全摘出バグ数に対する上工程での摘出バグ数の比率をいう。

品質会計によるソフトウェア品質向上の特徴は、以下の3点にある。

- ① レビューでのバグ摘出による早期品質確保
- ② 的確なテスト完了判断
- ③ 品質会計を軸とした全体的な仕組みの構築と運用による品質の底上げ

上記のうち①および②は、品質会計そのものが持つ特徴であり、品質会計を適用することにより直接的に得られる効果である。③の全体的な仕組みは、品質会計による効果を最大化するために、品質会計を適用する組織が構築する必要がある。図-3は、品質会計を軸としたソフトウェアライフサイクル全体にわたる仕組み（品質保証体系）の例である。品質保証体系とは、品質会計を適用することによって得られたさまざまなデータに基づき、自組織のソフトウェア開発上の強みや弱みを分析し、弱みを改善し強みを活かすようにソフトウェア開発全体を仕組みとして整備したものをいう。仕組みとは、管理方法だけでなく、設計技術やテスト技術などソフトウェア開発にかかわるすべての技術を組み合わせた一連のものをいう。

品質保証体系とは、品質会計を適用し、品質保証体系を構築して運用することにより、1年間に顧客で発生する出荷後バグ件数を1/20に低減し、以降そのレベルを維持する実績を持つ。品質会計は、現在、NECグループ全体に渡って、標準的な品質管理手法として適用されている。

品質保証体系を構築して運用することにより、1年間に顧客で発生する出荷後バグ件数を1/20に低減し、以降そのレベルを維持する実績を持つ。品質会計は、現在、NECグループ全体に渡って、標準的な品質管理手法として適用されている。

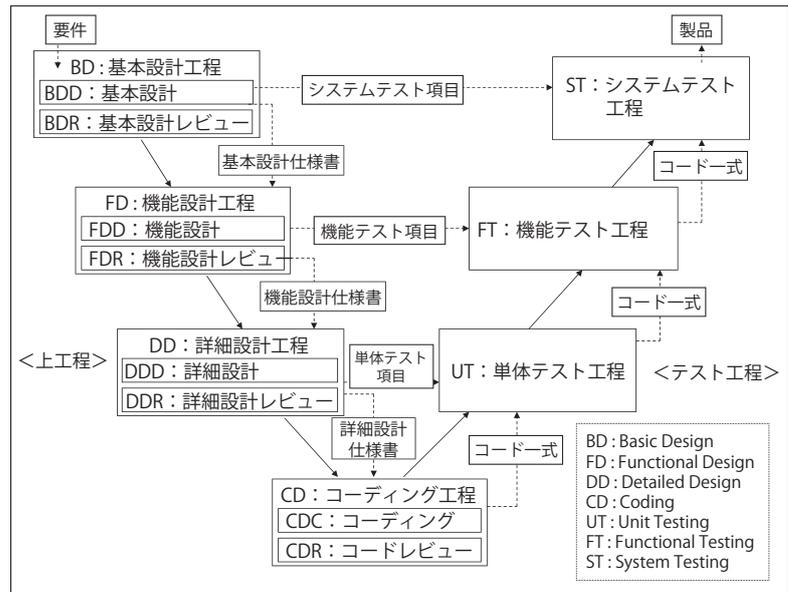


図-1 品質会計の想定する開発プロセス

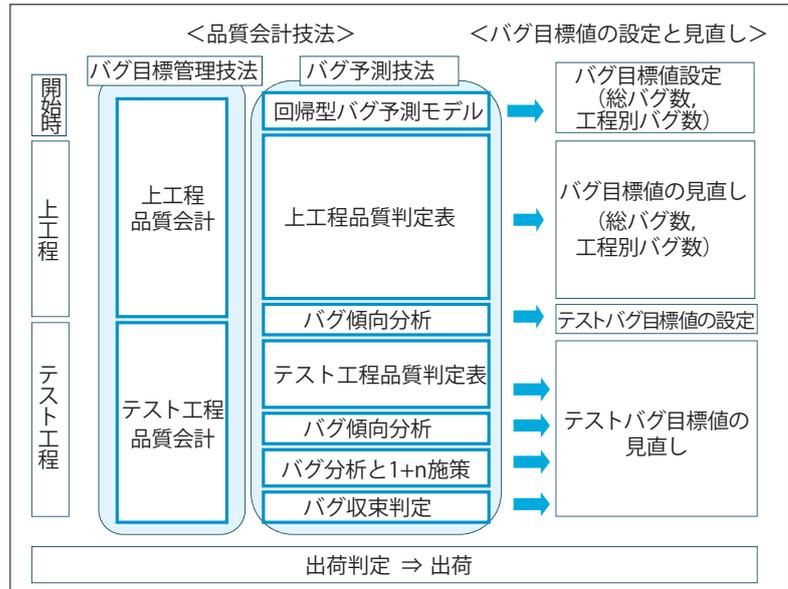


図-2 ソフトウェアライフサイクルと適用する品質会計技法

ソフトウェア品質会計における品質要件の把握と評価

● 内部品質・外部品質・利用時の品質要件の把握と評価

品質会計における、品質要件の把握と分析評価を図-4に、品質メトリクス一覧を表-2に示す。

品質会計では、要件を入力としてソフトウェア開発を開始する。要件から導出される各品質特性の要求に基づき、機能要求および非機能要求という視点からソフトウェアを設計し、設計仕様書として作成

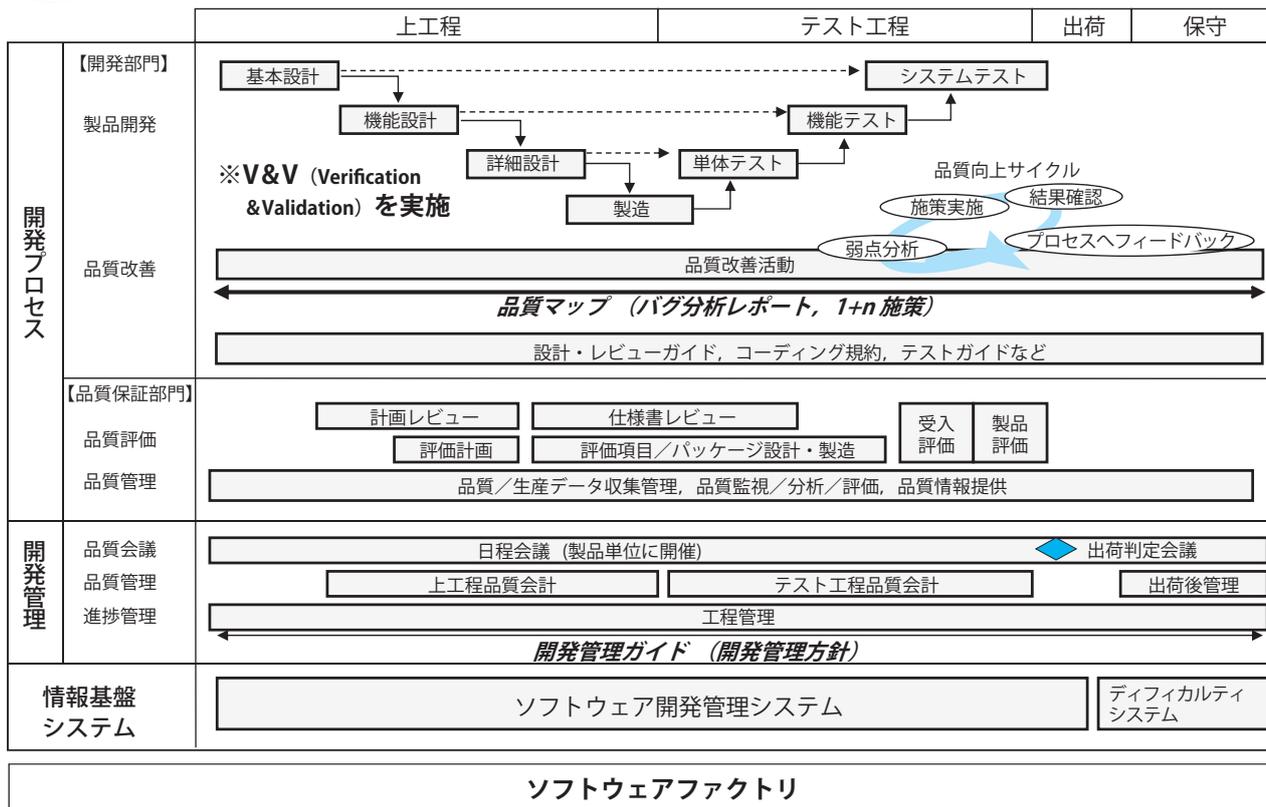


図-3 品質会計を軸とした品質保証体系の例

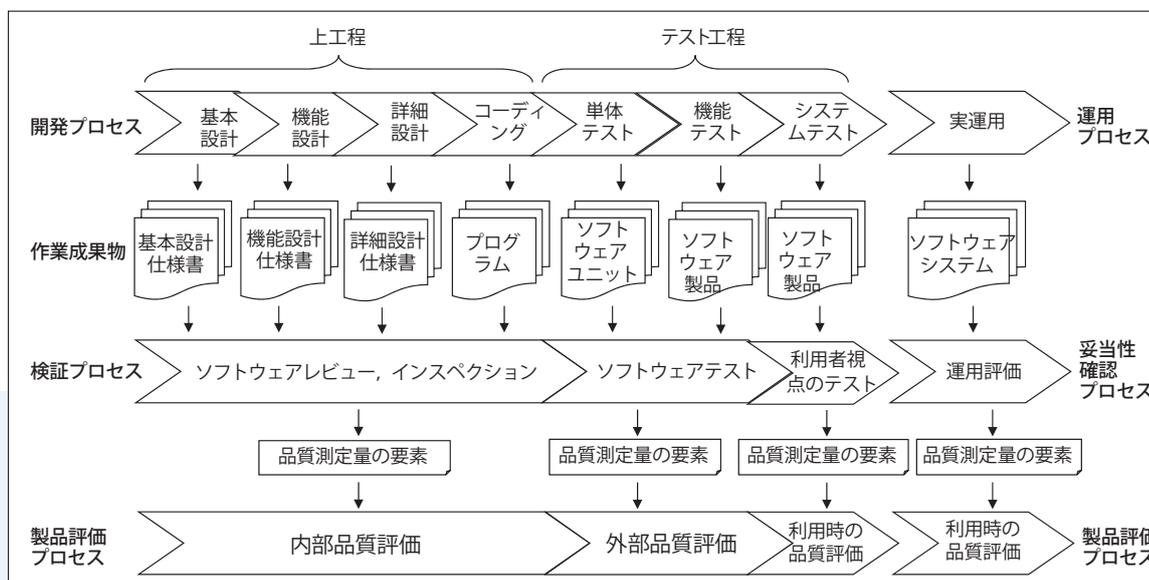


図-4 品質会計における品質要件の把握と分析

する。上工程における各設計工程で作成された設計仕様書は、同じ工程のレビューでその内容を確認する。品質会計では、設計仕様書の修正点はすべてバグとしてカウントするため、機能要求および非機能要求のバグはどちらも1件のバグとして扱う。すなわち品質会計においては、バグ数が主要な品質測定量であり、バグ数はすべての品質特性の品質測定量を包含する総合的な品質測定量と考えることができ

る。品質会計の重視するレビューでの早期バグ抽出は内部品質評価に該当し、「上工程バグ抽出率 80%」という目標は、内部品質評価によるバグ抽出を全体の80%以上とすることを意味する。内部品質評価とは、ソフトウェアの静的な属性の能力を主にレビューによって評価することをいう²⁾。

品質会計では、各工程で実施する「バグ傾向分析」(表-1参照)において、バグを品質特性やバグが作

9. ソフトウェア品質会計における品質要求と評価

内部品質メトリクス	単位	外部品質メトリクス	単位
設計・製造工数	人H	テスト工数	人H
レビュー工数	人H	テスト項目数	項目
仕様書作成量	頁	規模	Line
規模	Line	バグ数	件
バグ数	件	品質特性別バグ数	件
工程別作り込みバグ数	件	工程別作り込みバグ数	件
工程別抽出バグ数	件	工程別抽出バグ数	件

利用時の品質メトリクス	単位
バグ数	件
うち重大バグ数	件
バグ作り込み原因	-
バグ見逃し原因	-
品質特性別バグ数	件
バグ収束率	テスト終盤の収束度 / テスト全体の収束度 ※収束度 = バグ数 / テスト項目数

表-2 品質会計の主要な品質メトリクス一覧

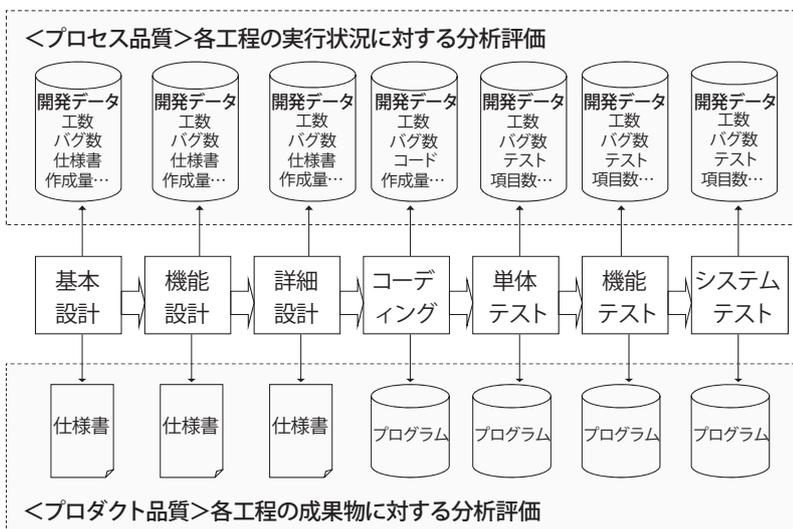


図-5 プロセス品質とプロダクト品質

り込まれた原因などの視点で分類して品質特性上の弱点を評価する。また、「品質判定表」（表-1参照）により、レビューやテストの程度に対する抽出されたバグ数を分析することにより品質を判定し、必要であればバグ抽出目標値を見直す。これらの結果は、随時バグ抽出目標値に反映する（「バグ傾向分析」および「品質判定表」の詳細については文献1）を参照）。

単体テストおよび機能テストは、外部品質評価に該当する。一方、テストの最終段階で実施するシステムテストは、利用者視点のテストであり、疑似的な利用時の品質評価と考えることができる。外部品質評価とは、ソフトウェアを実行することによりその能力を評価することであり、利用時の品質とは、利

用者ニーズが満足される程度をいう²⁾。

システムテストで抽出される当該ソフトウェアにとって重要なバグは、「バグ分析と1+n施策」（表-1参照）によって、1件ごとに作り込み原因と見逃し原因を分析し、類似のバグ抽出を実施する。バグ分析とは、バグ1件ごとの根本原因分析であり、1+n施策とは根本原因に基づく類似のバグ抽出である。また、「バグ収束判定」（表-1参照）により、テストの程度に対する抽出バグ数の推移をバグ収束率という観点で評価し、基準値を満足していることを確認する（「バグ分析と1+n施策」および「バグ収束判定」の詳細については文献1）を参照）。

● プロセス品質とプロダクト品質

プロセス品質とプロダクト品質の両面からの品質把握は、品質管理の要諦と言えるものである。

品質会計においても、プロセス品質とプロダクト品質の両面から品質状況を分析把握することが重要と考える（図-5参照）。プロセス品質と

プロダクト品質は、どちらか一方では正しく品質評価することができない。両方の視点から分析評価することによって初めて、的確に品質評価ができる。それは、品質会計を軸とした品質保証体系という仕組みのなかで実現する。

プロセス品質は、各工程で実施すべき項目を確実に実施しているかを分析把握することであり、各工程の開発実施状況を計測して得られた測定量を分析することによって評価される。たとえば、工数やテスト項目数などにより、当該工程で実施すべきことが実際に実施されたかを確認することは、プロセス品質分析の基本である。一方、プロダクト品質は、各工程の成果物そのものを評価した結果である。たとえば、各工程で作成される仕様書やプログラムの

出来を確認することである。プロダクト品質の代表的なものとして、最終ソフトウェア製品に対する品質保証部門による利用者視点の評価がある。

図-3の品質保証体系では、開発部門とは独立した品質保証部門による客観的な立場からの品質評価が、品質要件の把握と分析の鍵を握る。品質保証部門は、当該組織が実施する全プロジェクトを横断的に監視する立場にあるため、過去のプロジェクトの実施結果や測定量などの情報を保有する。過去のデータに基づく分析により、精度のよい分析結果を得ることができる。品質保証部門によるプロセス品質とプロダクト品質両面からの評価を、開発の全行程を通じて実施することにより、確度の高い品質評価が実現できる。

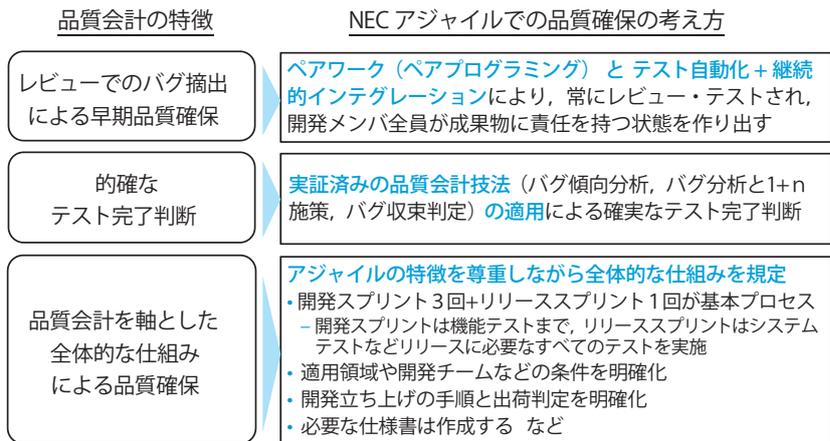


図-6 品質会計と NEC アジャイル

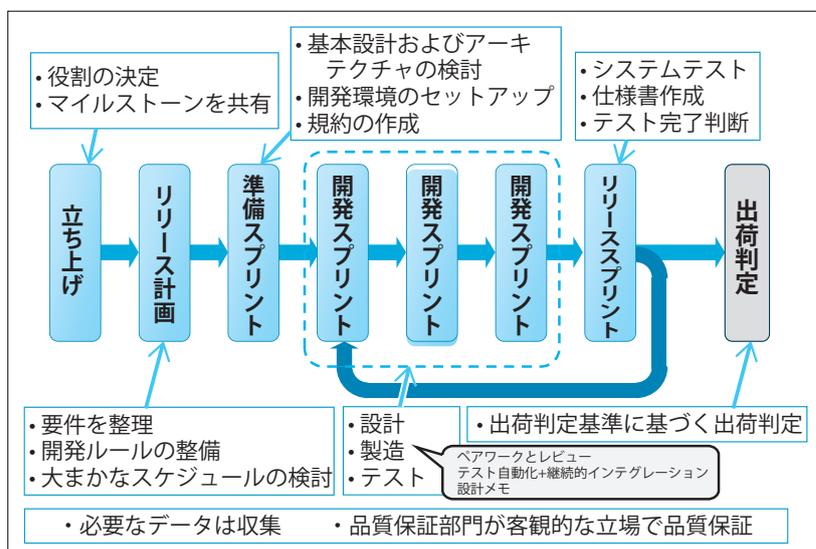


図-7 NEC アジャイルの全体像

アジャイルモデルにおける品質要件の把握と評価

● NEC アジャイルの考え方

NECでは、品質会計の品質確保の考え方をアジャイルモデルへ適用した社内標準を策定している。これを「NEC アジャイル」³⁾と呼ぶ。NECアジャイルの目的は、アジャイルモデル適用時の高いプロジェクト成功確率の確保である。

NECアジャイルは、アジャイル開発手法の1つであるスクラム⁴⁾をベースとしており、加えて品質会計が重要視する品質確保のための3つの特徴を、アジャイル開発に適用したものである（図-6参照）。

NECアジャイルの全体像を図-7に示す。アジャイルモデルでは、詳細な開発機能（以降、ストーリーと呼ぶ）の一覧表（以降、プロダクトバックログ

と呼ぶ）を開発開始時に作成する。このプロダクトバックログのなかから、優先順位の高いストーリーの順に開発を進める。

「品質会計を軸とした全体的な仕組みで品質確保」という観点に対して、「開発スプリント3回+リリーススプリント」を品質確保のための1つのサイクルとして定義している。スプリントとは期間を一定化した1回の繰り返し開発であり、通常1～2週間である。1回の開発スプリントで開発に着手したストーリーを完了するのが基本である。NECアジャイルでは1回の「開発スプリント」で、品質会計の想定する開発プロセス（図-3参照）のうち、機能設計から機能テストまでを同時に実施する。この開発スプリントを3回繰り返し、3回分の開発済みストーリーに対して、「リリーススプリント」でシス

9. ソフトウェア品質会計における品質要求と評価

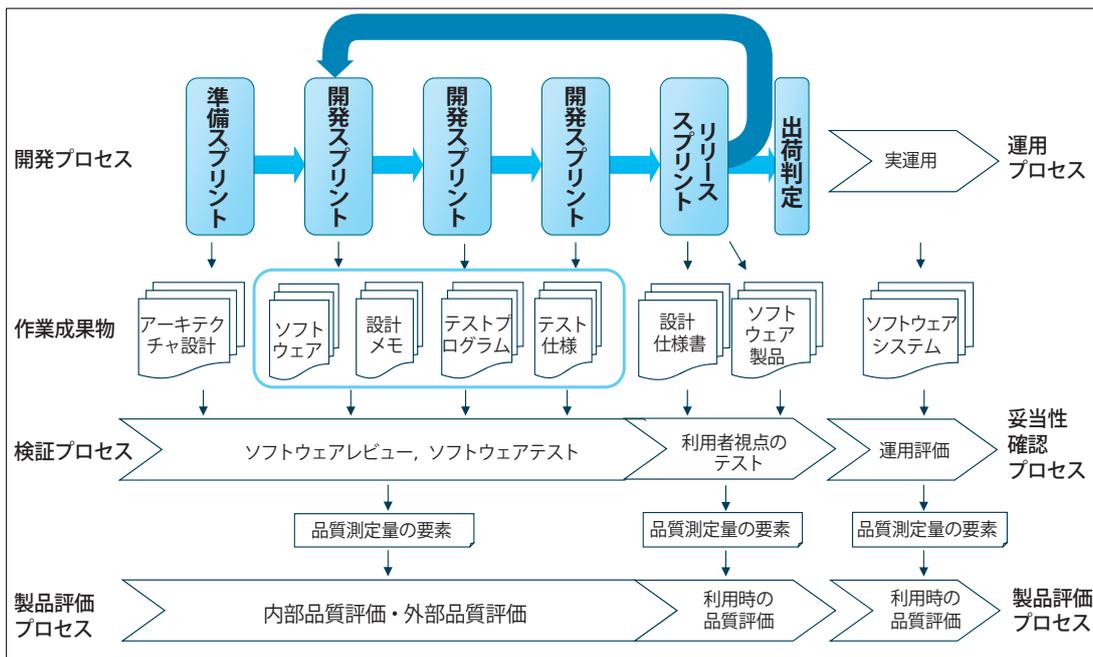


図-8
NECアジャイルの
品質要件の把握と
分析

テムテストを実施する。顧客の要望があれば、その後出荷判定を実施して、顧客へこのサイクルで開発したソフトウェアを提供する。この「開発スプリント3回+リリーススプリント」の1サイクルは1~2カ月であるため、その頻度で顧客へ動くソフトウェアを提供可能である。

「レビューでのバグ摘出による早期品質確保」の考え方は、各開発スプリントにおいて、ペアワークとテスト自動化+継続的インテグレーションを必須化することにより実現する。設計やコーディングをペアワークで実施し、設計やコーディングの作業者とそのレビューアが2人で作業することにより常にレビューされている状態を作り出して、レビューによる早期品質確保と同じ効果を狙う。さらにテスト自動化+継続的インテグレーションにより、動くソフトウェアを保証する。

「的確なテスト完了判断」は、品質会計の想定するシステムテストで実行すべき項目をそのままリリーススプリントへ適用することにより、ウォーターフォールモデルと同等の品質確保を実現する。

● NECアジャイルの内部品質・外部品質・利用時の品質要求の把握と評価

NECアジャイルでは、各開発スプリントの開始

時に、開発するストーリーを品質特性に基づき分解するとともに、1~2時間程度の作業タスクに分解する。アジャイルモデルでは、設計、製造、テストを同時に実施するため、同一開発スプリント内で内部品質評価および外部品質評価を実施することになる。リリーススプリントではウォーターフォールモデルのシステムテストと同等の評価を実施するため、リリーススプリントが疑似的な利用時の評価という位置付けとなる（図-8参照）。

NECアジャイルの品質メトリクス一覧を表-3に示す。作業タスクごとに担当者名・時間・作業内容を記録するため、ウォーターフォールモデルのデータよりも開発途中の詳細なデータが得られる。また、ストーリーごとに集計して品質状況を詳細に分析することも可能である。

バグは、開発スプリントにおいて当該ストーリーの開発が完了となったとき以降に摘出されたバグを計測する。したがって、バグとしてカウントされるのは、開発完了したストーリーがデグレードした場合とシステムテストで問題が摘出された場合である。ウォーターフォールモデルで設計レビュー以降に摘出されたバグを計測しているのに対して、アジャイルモデルの特性を活かすために計測対象範囲を絞っている点が特徴である。

内部および外部品質メトリクス①	単位	内部および外部品質メトリクス②	単位
作業別工数	人H	テスト項目数	項目
・実装	人H	・自動	項目
・リファクタリング	人H	・手動	項目
・ドキュメント作成	人H	・システムテスト項目	項目
・テスト	人H	・リグレッションテスト実行数	回
・ペアワーク	人H	バグ数	件
・レビュー（全員）	人H	・うちデグレード	件
ストーリー	件	・うち重大	件
・Done	件	仕様書	ファイル数
・申し送り	件	・行数	行
ベロシティ	SP	・図の数	個
・全体ストーリーポイント	SP	・メモ数	個
・消化ストーリーポイント	SP	各種指標	-
規模	Line	・カバレッジ	%
・ソフトウェア	Line	・ネスト	-
・テストコード	Line	・複雑度	-

利用時の品質メトリクス	単位
バグ数	件
うち重大バグ数	件
バグ作り込み原因	-
バグ見逃し原因	-
品質特性別バグ数	件
バグ収束率	テスト終盤の収束度／テスト全体の収束度 ※収束度＝バグ数／テスト項目数

表-3
NEC アジャイルの
品質メトリクス一覧

リリーススプリントでは、計測されたバグに対して、ウォーターフォールモデルと同様に、品質特性ごとの観点でのバグ傾向分析やバグ分析等を実施するとともに、バグ収束判定により品質を評価する。出荷判定では、ウォーターフォールモデルと同じ出荷判定基準を適用して判定する。

NEC アジャイルでは、ウォーターフォールモデルと同様に、独立した開発部門と品質保証部門の両者からなる枠組みが適用されている。品質保証部門が客観的な立場でアジャイル開発全体を通じてプロセス品質とプロダクト品質を分析評価する。

今後の展望

ソフトウェア品質会計によるウォーターフォールモデルおよびアジャイルモデルでの品質要求と評価について紹介した。ソフトウェア開発の現場では、これらの開発プロセス以外にも、ビジネスの変化に

応じてさまざまな開発プロセスが考案され、実際に適用されている。今後も新たに登場する開発プロセスに対して、確実に品質要求および評価を実施し、顧客へ品質の良いソフトウェアを提供していく所存である。

参考文献

- 1) 菅田直美：ソフトウェア品質会計—NECの高品質ソフトウェア開発を支える品質保証技術—, 日科技連出版社, 東京 (2010).
- 2) ISO/IEC 25000 (JIS X 25000) : SQuaRE Series.
- 3) Kosaki, M., Kosumi, Y., Terada, T. and Honda, N. : Toward High-quality Agile Software Development : NEC's Agile Development Management Method, ProMAC2013 (Nov. 2013).
- 4) Jonathan Rasmusson (著), 西村直人 (監訳) : アジャイルサムライ—達人開発者への道—, オーム社 (2011).

(2013年10月19日受付)

● 菅田直美 (正会員) n-honda@ay.jp.nec.com

日本電気(株)入社後、IT系汎用ミドルソフトウェア製品の品質保証に従事。現在はNEC全社のソフトウェア品質向上に従事。主席品質保証主幹。工学博士。