
『ソフトウェア・プロセス品質』の考え方と 実践を通じた考察

(株) 富士通北陸システムズ

■ 執筆者 Profile ■



後藤 玄次

1987年 (株) 富士通北陸システムズ入社
言語/帳票などのミドルウェア開発に
従事
2006年 品質検証部へ異動
2008年 現在 UNIX サーバ関連製品検査担当

■ 論文要旨 ■

完成後のソフトウェア成果物に対する第三者検証の効果は、比較的良く認知されている。しかし、ソフトウェア成果物の場合、最終製品が複雑、実態が容易に目に見えない、加えて、限られたテスト項目の実施だけで、必要とするソフトウェアの幅広い品質全部を保証できるものではないという特質がある。このため、「完成品に基づく品質保証」のみではなく、「プロセスに基づく品質保証」に重点を置くことが、ソフトウェアの品質保証のあり方として重要であると考えられる。

上記考えのもと、第三者によるソフトウェア・プロセス品質支援に対する考え方、及び取組の実践を通じた結果から、“考え”と“実際”の FIT&GAP を考察し、その有効性と可能性について論述する。

■ 論文目次 ■

1. はじめに	《 3》
2. 開発プロセスから見た現状の問題点	《 3》
3. 開発プロセス点検の実施内容	《 4》
3. 1 目的	
3. 2 点検実施手順	
3. 3 点検の観点	
3. 4 点検の実施	
3. 5 点検結果の定量表現による可視化	
3. 6 点検結果のフィードバックと報告	
4. 開発プロセス点検実施上の工夫	《 9》
4. 1 開発プロセス良好率の扱い	
4. 2 開発プロセス点検結果データ蓄積への取組	
5. 評価と課題	《 11》
6. おわりに	《 11》

■ 図表一覧 ■

図 1 ソフトウェア開発プロセス点検実施時期.....	《 5》
図 2 開発プロセス点検結果連絡票.....	《 8》
図 3 開発プロセス良好率の部門別推移.....	《 10》
表 1 BSBOKプロセスとSDEMプロセスの対応表	《 4》

1. はじめに

ソフトウェア開発では、プロジェクトにおけるQCD（品質、コスト、納期）のバランスを定義し、それらを管理状態におくことで、開発プロセスを確実に推進し、予定期間内、予定コスト内で高品質プロダクトを実現することが求められる。しかし、システムが複雑化し、技術進歩が高速化するなかで、さらなる短納期化、開発コストの削減が求められる今日のソフトウェア開発において、開発部門のプロジェクト管理者がすべてを受け止めて、その自助努力のみで開発プロセスの質を高め、最終プロダクトの品質確保を遂行することは、大変困難である。そのため、プロジェクトメンバーのみではなく、品質検証部などの支援部門がこれを支援するための活動を展開することが必要となる。

当社のソフトウェア事業本部には、ミドルウェア製品を開発する部門（以降、本稿では「ミドルウェア系開発部門」と称する）と、SE部門パッケージ製品／フィールドSI製品を開発する部門（以降、本稿では「フィールド系開発部門」と称する）がある。当社品質検証部では、ミドルウェア製品に対する製品検査業務を行う一方で、SE部門開発製品に対する第三者検証を行い、作成されたSI成果物に対する検証を行っている。

近年、フィールド系開発部門において、工程遅延が発生したり、レビュー、テストの充分性が不透明なまま工程を進めているプロジェクトが増えてきており、最終成果物の第三者検証だけでは未然に防止できない品質問題や、外部仕様書の不備により最終成果物の第三者検証が成立しえない問題が顕在化してきた。

そこで、成果物検証よりも早い段階、つまり、成果物の作成過程である開発プロセスに対して、第三者としての検証を実施する『ソフトウェア開発プロセス点検活動』に取り組むこととした。

本稿では、2006年から実施している『ソフトウェア開発プロセス点検活動』の取組を振り返り、プロセス品質向上活動の実際のノウハウと、実践を通して得られた効果や可能性について論述する。

2. 開発プロセスからみた現状の問題

市場の短納期化の要請やアジャイル開発モデルの導入により、かつてウォーターフォール開発モデルが標榜した、秩序だった開発プロセスが崩れつつある。新しい開発手法に適合した開発プロセスについても確立できていない状況にある。このため、近年の開発現場では以下の開発プロセスがおろそかになりつつある。

- ・仕様を決めて、設計書のレビューを行ってから開発に着手
- ・品質を判断して次工程に進めてよいか判定する工程完了基準の適用

しかし、ミドルウェア系開発部門においては、仕様書やソースプログラムを記述すれば必ずレビューするといった、最も根本的な基本プロセスはしっかり守られている。この理由として、検査部門の存在があげられる。最終的に検査部門における出荷検査で合格とならない限り製品出荷は行えないことがルール化されている。レビューや十分なテストを怠り、安易に納期を重視した開発を行えば、検査段階で数多くの問題が指摘され、その修正

に多大な時間や苦勞の負担が発生することをミドルウェア製品の開発者は痛切に感じている。

安易な開発作業を行えば、必ず検査に引っかかることを理解しているので、基本プロセスは決して崩さない。放置すれば、安易に流れ、崩れていく開発プロセスをしっかりと固めて守る意味で検査部門の存在の意味は大きい。

一方、フィールド系開発部門においては、歯止めとなる検査プロセスがルール化されておらず、納期やコストとの兼ね合いから、品質プロセスの充分性に疑問をもつケースが散見される。CMMで述べられる未熟な組織の状態におちいることすらありうる。以下、品質関連部分を引用する。

『未熟な組織では、成果物品質を判断するためや、成果物またはプロセスの問題を解くための客観的基準がない。そのため、成果物品質を予測することは困難である。プロジェクトが予定より遅れてくるとレビューやテストのような品質を高めるための活動がしばしば削減されるかまたは省略される。』

3. ソフトウェア開発プロセス点検の実践内容

3.1 目的

『ソフトウェア開発プロセス点検』は、当社ソフトウェア事業本部で採用する二つのプロセス標準（BSBOK、SDEM）で規定されるプロセスが、規定どおりに実施されているかどうかを第三者である品質検証部員がチェックし、徹底させる仕組みである。開発プロジェクトに潜在するプロセス上の問題を早期発見して対策へのトリガーとすることを目的とする。当社で採用する二つの開発プロセス標準である、BSBOKとSDEMは、開発工程の基本的な部分ではほぼ対応させて考えられる。表1に工程の対応を示す。

一般的な名称	BSBOK 工程	SDEM 工程
要件分析	調査 RS	システム化計画 SP
基本設計	基本設計 BD	システム要件定義 RD
外部仕様設計	機能設計 FD	ユーザーインタフェース設計 UI
内部仕様設計	構成設計 SD	システム構造設計 SS
詳細仕様設計	詳細設計 DD	プログラム構造設計 PS
コーディング	コーディング MK1	プログラミング PG
単体テスト	モジュールテスト MK2	プログラムテスト PT
結合テスト	インタフェーステスト MK3	
統合テスト	コンポーネントテスト CT	結合テスト IT
機能テスト		
総合テスト	システムテスト ST	システムテスト ST
運用テスト	(現場試験 FT)	運用テスト OT
製品検査	検査 PT	(対応工程なし)

表1 BSBOK プロセスと SDEM プロセスの対応表

『ソフトウェア開発プロセス点検』のコンセプトは、「開発のプロセス品質追及によりプロダクト品質を向上する」という考え方であり、具体的には、以下を目的に開始した。

- ・開発プロセス上に見え隠れするリスクを早期発見・対策し、開発プロセスをあるべき

姿に保つ品質問題の発生を未然に防止

- ・開発プロセス作業の良し悪しを定量データで表現する試みにより、開発プロセスの品質状態という把握しづらい対象を測定する際のモノサシとして活用
- ・プロジェクト管理者が複数プロジェクトを兼任する場合、手薄になりがちな個々のプロジェクトの開発プロセスチェックの支援
- ・第三者がチェックすることにより状況の透明性が向上し、必要に応じてタイムリーなエスカレーションを可能とする

3. 2 点検実施手順

『ソフトウェア開発プロセス点検』は、成果物を作成するための各開発工程を監視する活動であり、開発計画完了（開発の立ち上がり）時、及び各工程の完了時に点検を実施する。

開発計画完了（開発の立ち上がり）時には、開発計画書を基に各工程の点検スケジュールを設定するとともに、見積開発量などの基礎データを把握する。

各工程の完了時では、進捗資料及び開発データを点検し、プロセスの充分性を検証する。

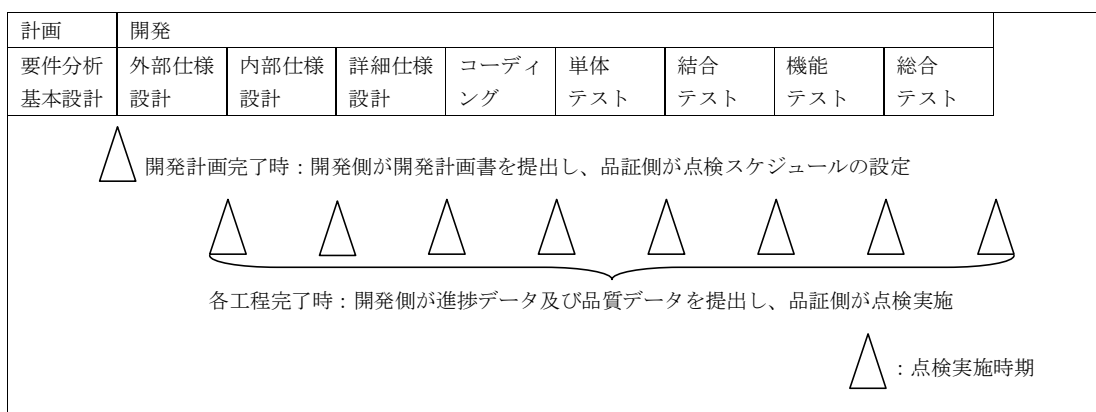


図1 ソフトウェア開発プロセス点検実施時期

3. 3 点検の観点

『ソフトウェア開発プロセス点検』での点検内容や観点が属人的になり、点検担当者によるバラツキが発生することを防止し、点検結果を均質なものとするため、チェック観点をリストに整理した。各チェック項目についても、結果判定に曖昧さが入る余地を極力排除するため、品証部員全員の経験から得られた意見を集約し、設定した。

なお、通常のソフトウェア開発プロセスは、見積、プロジェクト計画・管理、要件管理、開発（設計、プログラム作成、レビュー、テスト）、品質管理、進捗管理、構成管理など、様々な要素が存在する。本取組では、これら各分野を網羅的にチェックするのではなく、当社の実態を踏まえたうえで、品質に直結すると考えられる可能な限りプリミティブな項目を厳選し、比較的短時間で、広範囲なプロジェクトをチェック対象としてカバーすることに重点を置き、必要最小限のチェック項目としていることが特徴である。追試可能性の観点より、本取組で使用したチェック項目を以下に掲載する。

1. 開発プロジェクト管理

(1) タイム

- a) 漏れなく各開発工程のスケジュールが記載されているか。
- b) 予定線表に対する現在の実績状況が記述されているか。
- c) 各工程の実施手順を遵守しているか。
- d) 各工程の実施手順が遵守されていない場合、その理由が明確になっているか。
- e) 作業の状況（ステータス）が正しく記述されているか（完了、作業中、未着手、計画中など）。
- f) 工程に遅れがないか。
- g) 遅れがある場合、遅れの期間が明確になっているか。
- h) 遅れがある場合、遅れの原因が明確になっているか。
- i) 遅れがある場合、対策は明確になっているか（スケジュール調整、要員調整、開発機能調整 など）。
- j) 前月の進捗状況と比較して、進んでいない作業については、その理由が明確になっているか。
- k) マニュアル記事執筆の作業有無が明確になっており、作業がスケジュールされているか。

(2) コスト／スコープ

- a) 開発量の実績見込（または実績）が当初見積を超過していないか。
- b) 超過している場合、超過工数が明確になっているか。
- c) 超過している場合、超過の原因が明確になっているか。
- d) 超過している場合、対策が明確になっているか（工数調整、開発機能調整、スケジュール調整、要員調整 など）。

(3) 人的資源

- a) 担当者の作業分担において、無理な工数配分になっていないか（作業集中など）。

(4) リスク

- a) 開発量が見積を超過している場合、計画する対策内容について、品質へのリスク分析とその回避策が充分検討されているか。
- b) 「問題なし」と報告されている場合、その根拠が明確（定量データと辻褃が合っているなど）になっているか。

2. 仕様確定（設計工程で点検）

- a) 仕様書枚数が著しく少ないか。
- b) 作成しなければならない仕様書が作成されているか。
- c) 仕様FIXされた後、次工程に進んでいるか。

3. レビュー（設計工程、コーディング工程で点検）

- a) 必要なレビューが実施されているか（実績日の記入など）。
- b) 品質目標値（レビュー時間、検出バグ数）を設定しているか。
- c) 品質目標値の設定根拠は適切か。
- d) 品質実績値（レビュー時間、検出バグ数）が、目標値の許容範囲内（±20%など）か。範囲外の場合はその理由が明確か。

- e) 検出されたバグ分類（論理、I/F、単純、冗長、改善 など）を分析し、対策が実施されているか。
- f) 検出されたバグの原因工程を分析し、対策が実施されているか。
- g) 検出されたバグがすべて対応済か。

4. テスト（テスト工程で点検）

- a) 品質目標値（テスト項目数、検出バグ数）を設定しているか。
- b) 品質目標値の設定根拠は適切か。
- c) 品質実績値（テスト項目数、検出バグ数）が、目標値の許容範囲内（±20%など）か。範囲外の場合はその理由が明確か。
- d) 検出されたバグ分類（論理、I/F、単純、冗長、改善 など）を分析し、対策が実施されているか。
- e) 検出されたバグの原因工程を分析し、対策が実施されているか。
- f) 検出されたバグが特定の機能または担当者に偏って発生していないか分析し、対策が実施されているか。
- g) 検出されたバグがすべて対応済か。

3. 4 点検の実施

『ソフトウェア開発プロセス点検』は、開発者から提出された進捗資料や開発データを、検証担当者がチェックリストに従ってしたがってチェックを行い、その結果を記録する。この際、各項目のチェック結果は、以下の3パターンで表現する。

○：問題なしと判定

×：問題ありと判定 または 記載事項やデータがなく判定できない

－：判定対象外

本取組でのポイントとして、進捗資料へ必要事項が記載されていない場合や、必要データが提示されていない場合は、判定不可能とはせず、問題ありとしている点である。これら、必要情報が提示されない事実は、開発プロセスとして問題視している、「状況の不透明化」へ直結するためである。

点検結果は、「開発プロセス点検結果連絡票」としてまとめ、開発元へのフィードバックに利用すると同時に、サーバに蓄積する。「開発プロセス点検結果連絡票」を図2に示す。「開発プロセス点検結果連絡票」では、チェックした結果のみではなく、チェック観点やチェック項目そのものを明記して開発元へ開示する。これにより、チェック結果で良し悪しを測るのみではなく、開発側へ最初からチェック項目の内容を意識したプロセスの実施を促す効果が期待できる。

【開発プロセス点検結果連絡票】

製品名:		開発(発注)・単体状況:	
V/A:		開発(発注)・単体状況:	
プラットフォーム:		開発(発注)・単体状況:	
コンポーネント名:		開発(発注)・単体状況:	
機能名:		開発(発注)・単体状況:	
開発種別:		開発(発注)・単体状況:	
開発担当者:		開発(発注)・単体状況:	
開発担当者:		開発(発注)・単体状況:	
点検担当者:		開発(発注)・単体状況:	
点検実施日:		開発(発注)・単体状況:	

工程	分類	点検内容	結果	改善理由(箇所)/対象外の理由/指摘事項
管理	(1) 資料・他	品質管理手続表に、該当する月の開発元提出予定が記入されているか。		
		「予定工程実施計画表(開発元提出)」が提出(予定予定に)提出されているか。		
		開発データ資料(機能単位チェック表、品質管理表、等)が提出(予定予定に)提出されているか。		
		開発データ資料には、必要なデータが記載されているか。		
		開発データ資料で、開発データに不足はないか(開発ステップ数設計が含まない、記載よりも多〜記載の数が多い等)。		
(2) タイム	(1) 品質管理手続表に、該当する月の開発元提出予定が記入されているか。	前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
(3) コスト	(1) 品質管理手続表に、該当する月の開発元提出予定が記入されているか。	前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
(4) 人的資源	(1) 品質管理手続表に、該当する月の開発元提出予定が記入されているか。	前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
(5) リスク	(1) 品質管理手続表に、該当する月の開発元提出予定が記入されているか。	前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
(6) マニユア	(1) 品質管理手続表に、該当する月の開発元提出予定が記入されているか。	前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
(7) 仕様確定	(1) 品質管理手続表に、該当する月の開発元提出予定が記入されているか。	前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
(8) レビュー	(1) 品質管理手続表に、該当する月の開発元提出予定が記入されているか。	前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
(9) SS	(1) 仕様確定	前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
(10) SS/PV	(1) レビュー	前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
(11) PT	(1) テスト	前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
(12) IT	(1) テスト	前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
(13) ST	(1) テスト	前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
(14) ST	(1) テスト	前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		
		前月の指摘が反映されているか。		

前月指摘の対応状況: 対応済、 未対応、 対応中/対応不十分、 保留、 中止

○:合格、●:改善部/記載なし、×:対象外

図2 開発プロセス点検結果連絡票

3.5 点検結果の定量表現による可視化

『ソフトウェア開発プロセス点検』の実施結果から開発プロセスの良し悪しを表す定量的指標として、開発プロセス良好率を設定した。これは、「開発プロセス点検結果連絡票」を基に次の規則で設定した。

$$\text{良好率} [\%] = \frac{\text{チェック結果○の数}}{\text{チェック結果○の数} + \text{チェック結果×の数}}$$

非常にシンプルな考え方であるが、この良好率を、製品単位

、プロジェクト単位で集計分析することで、開発プロセス品質状況の傾向を把握したり、開発チーム自らが改善の指標として活用するという、これまでのプロセス監視に比較して、

結果の活用度を向上することを目指した。

3. 6 点検結果のフィードバックと報告

点検結果の開発チームに対するフィードバックは、点検結果に応じて次に示す二つのパターンで実施する。

(1) e-mail による「開発プロセス点検結果連絡票」の送付

全プロジェクトを対象として点検の実施にあわせて毎回行う。送付先としては、プロジェクトが属する部門の幹部社員及び開発リーダーとしている。

(2) 対面形式による報告とヒアリング

点検の結果、問題ありと判断したプロジェクトに対しては、「開発プロセス点検結果連絡会」を開催し、Face to Face での議論の場を設定する。開発リーダーと実施するが、必要に応じて幹部社員への参加も要請する。

「開発プロセス点検結果連絡票」は公開サーバへ蓄積することにより、指摘したまま放置することなく、前回指摘事項への対応状況なども、逐次フォローしている。

また、経営層への報告も定期的（1回/月）に設定した。これまでは、抽象度の高い開発プロセスの良し悪しを、経営層向けの報告としてまとめるのは、困難を要した。例えば、プロセスの良し悪しは、見積りから始まり、スケジュール面、仕様確定度合い、レビューの充実度、テストの充分性など多岐に渡る要素全体で評価する必要がある、端的に良い悪いを表現する手段がなかった。本取組で開発プロセス良好率という数値表現を導入したことにより、開発プロセスの状況をダイジェストとして報告することを標榜した。経営層に対して開発プロセス状況が定期報告されているという事実は、開発部門への適切な緊張感醸成にもつながる。

4. 開発プロセス点検実施上の工夫

4. 1 開発プロセス良好率の扱い

『ソフトウェア開発プロセス点検』は幾度と無く実施手法の改善を試みてきた結果、本稿で説明してきた手法になった。まだ基礎データが充実していないため、開発プロセス良好率が何%であればよいのかといった基準が、明確化されていない点が、現状の問題である。当社ソフトウェア事業本部においては、ミドルウェア系開発部門とフィールド系開発部門があり、ミドルウェア系開発部門は、製品検査の存在が歯止めとなり、一定の開発プロセス品質が保たれていることは、本稿「2. 開発プロセスからみた現状の問題」で既述したとおりである。

本取組では、この当社ソフトウェア事業本部内の部門間における差異を利用し、まず、ミドルウェア系開発部門の開発プロジェクト点検結果から導き出される開発プロセス良好率を、ひとつの指標としてフィールド系開発部門の状態を判断することとした。過去に3ヶ月間の実施結果をもとにベンチマークした結果を、図3に示す。

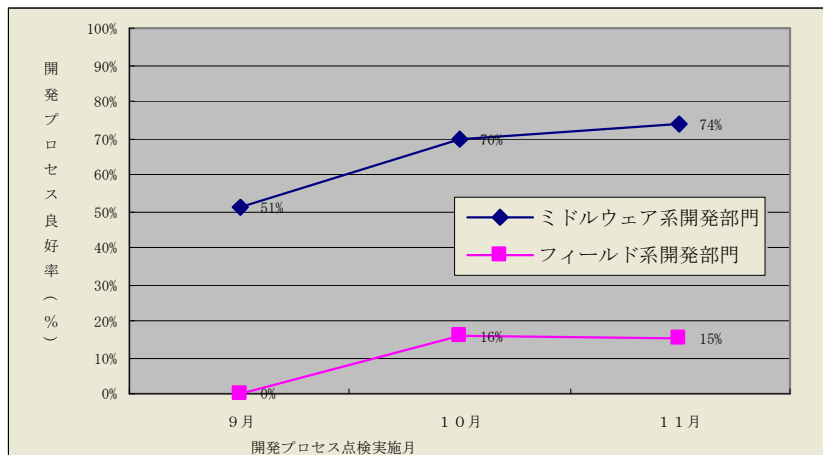


図3 開発プロセス良好率の部門別推移

ミドルウェア系開発部門の開発プロセス良好率は、点検3ヶ月目において、74%となっており、ソ事本ミドルウェア開発での標準的な開発プロセス良好率基準と仮定した。これに対して、フィールド系開発部門では、2ヶ月間連続で約15%と、ミドルウェア系開発部門に比較して著しく開発プロセス品質が悪いといえる。

これは、毎バージョン同一の製品を手がけ、ほぼ、同等のメンバ構成で機能エンハンスを繰り返すミドルウェア系開発に比較し、毎案件に新たなスコープやメンバ構成で実施するフィールドSI案件プロジェクトの難易度の差が如実に現れており、中でも顕著な傾向として以下の状況が明確になった。

- ・開発量を定量的に管理して体制とのバランスをトータルで調整しきれていない
- ・開発工程の区切りや意味づけが甘い
- ・レビュー、テストといった品質活動の充分性をデータで検証していない

品質管理担当者としてフィールド系開発部門の開発プロセス品質を向上させる活動を開始する根拠として、開発部門へ状況を説明する材料として、本データを上手く活用することで、事態に対する理解の共有を促すことも、プロセスを定量的に可視化することの効果の一つと考えている。

4.2 開発プロセス点検結果データ蓄積への取組

前項で述べたとおり、フィールド系開発部門の点検結果の分析から、何点か問題傾向が把握できたが、そのなかでも、「レビュー、テストといった品質活動の充分性をデータで検証していない」問題に着目した。

開発プロセス品質を見る場合の最も基本的なポイントは、レビューの充分性、テストの充分性である。『ソフトウェア開発プロセス点検』では、レビューの充分性、テストの充分性を判断する手法として、レビュー努力と検出バグの兼ね合い、テスト努力と検出バグの兼ね合い、それぞれからレビュー・テストの充分性を可視化する品質管理図の手法を用いている。品質管理図は非常にベーシックな手法である。開発者も無意識のうちに品質指標と実績値の比較の中で行っている考え方である

フィールド系開発部門での問題点をうけ、『ソフトウェア開発プロセス点検』の結果としてデータを蓄積し、それに対応する品質管理図を、「品質データ管理票」として開発元へ

フィードバックを試行開始した。開発者ができていない部分を実施するよう指導するのみではなく、可能な限り支援する観点で、プロセス品質の良し悪しをデータで可視化する手段の一つとして取り組んでいる。

5. 評価と課題

5.1 効果

本稿で述べてきた『ソフトウェア開発プロセス点検』への取り組みにより、次のとおり、一定の効果、及び今後への期待効果が明確になった。

- (1) 開発プロセス品質の数値化（開発プロセス良好率設定）により改善状態が明確化する効果

[本活動開始（2006年）後1年間での開発プロセス良好率改善実績]

開始時良好率：45% ⇒ 1年後良好率：63% : 18ポイント改善

- (2) 「開発プロセス点検結果連絡票」を定期的に関係部門へフィードバックすることで、開発者の活動へのコミット意識（日常的に関係部門へ取り組む基本姿勢）を促進する効果

[本活動開始（2006年）後1年間での開発データ提出率改善実績]

開始時提出率：48% ⇒ 1年後提出率：78% : 30ポイント改善

- (3) 開発プロセス品質の定量的可視化がさらに進展すれば、経営層、プロジェクトマネージャ、開発リーダー、開発メンバーが、役割や立場の違いを超えて、同じモノサシで開発プロセス作業品質を共有することが可能となる。このことは、開発プロセス品質の状態を共有するためのコミュニケーションが、より円滑・高精度になることを意味する。従って、問題の存在する作業プロセスを素早く認識できることから、品質リスクマネジメントの効率化という効果も期待できる。

5.2 課題

本取組は、実践を通じたプロセス品質の定量化モデルの構築途上段階であることは否めない。今後は、次の課題に取り組むことで、更に成熟させていきたい。

- (1) 開発プロセス良好率と成果物品質との相関関係明確化

開発プロセス良好率データを蓄積し、下流テスト工程または最終的な第三者検証での成果物品質との相関関係を明確にすることにより、開発途中でのプロセス品質データから、最終品質の予測が期待でき、事前にプロセスを修正するための、より正確なトリガーとできる。

- (2) 開発プロセス良好率と問題兆候への具体的対策のリンク

開発プロセス良好率をより細分化してゆき、分析し発見される問題兆候への具体的な対策案（事例）まで提供できる仕組みとする。

- (3) 標準プロセス（BSBOK、SDEM21）のFJH開発へのテーラリング指標

富士通で定義されている開発プロセス標準は、各組織や業務形態に応じてテーラリングして適用することが前提となっている。『ソフトウェア開発プロセス点検』を通して、努力しても向上しない良好率については、開発プロセス標準がミスマッチを起こしている可能性も考えられる。こうした開発標準のテーラリング指針として

も活用できる仕組みを構築。

6. おわりに

本稿で述べた手法やアプローチは、ある意味、特に目新しいものではないかもしれない。しかし、これを確実に実行して効果を生み出すには多くの時間と労力が必要である。更に、効果が出るまでは、開発担当者から、「意味があるのか?」、「面倒なだけ」などの、批判を伴うプレッシャーを常に受けることになり、忍耐という品質担当としてのマインドが求められることになる。しかし、「あたりまえのやるべきこと」を「確実に」実行し続けることを除いて、品質を良くする道はない。まさに、「銀の弾丸」はない以上、IT 業界の一員として、当社でも品質重視の文化を更に定着させるべく、日々、精進して行きたい。「質はどこでもものを云う」のである。

参考文献

- [1] 保田 勝通：“ソフトウェア品質保証の考え方と実際”，(財)日本科学技術連盟，1995
保田 勝通：“ソフトウェア品質保証の考え方と実際”，(財)日本科学技術連盟，1995
- [2] ソフトウェア能力成熟度モデル1.1版