

# 品質マネジメント

[プロジェクトマネージャー]

## ソフトウェアの品質とバグ

### バグ混入と除去のサイクル

ソフトウェアのバグは開発のどのどの工程でも入り込む可能性

- ・ 工程
  - ・ 要件定義
  - ・ 基本設計
  - ・ 詳細設計
  - ・ コーディング

バグの潜伏期間が長いほど、除去するコストは増大する

成熟したプロセスでは、各工程の終わりには、必ず品質管理作業が含まれる

バグを除去する作業

- ・ 要件定義レビュー
- ・ 設計レビュー
- ・ コード・レビュー
- ・ 単体試験 (UT)
- ・ 結合試験 (IT)
- ・ システム試験 (ST)
- ・ 受入試験 (AT)

### 品質マネジメントの作業

- ・ 適切な品質管理活動を計画
- ・ 正しく実行し管理
- ・ ほとんどのバグが「プロセス中」(ソフトウェアの納入前)に摘出されるようにすること

品質マネジメントにおける手続き的アプローチ

#### プロセス

#### レビュー

- ・ 人間の手による構造化されたプロセス

#### 試験

- ・ バグを発見するためにソフトウェア(またはその一部)を実行するプロセス

#### ガイドライン

レビューや試験の手続きやガイドラインが定められる

## 品質について

「手続き的アプローチ」

- ・バグを除去するため
- ・語るができない
  - ・バグを除去した割合
  - ・手続きが完了した後のソフトウェアの品質
  - ・手続きを実行しただけ

有効性を判断する材料にはならない

最終コードの品質を評価することは出来ない

- ・問題点
  - ・プロジェクトマネージャーが品質を定量的に評価するための手段がない

CMM レベル3 では、適切な品質マネジメントの手続き的アプローチが求められる

品質マネジメントにおける定量的アプローチ

バグ特定プロセスの効果をもっとよく評価

手続きは行われたかというよりも高いレベルの方法論が必要

バグデータを評価に使う必要

バグデータを分析

ソフトウェアの品質を判断

追加の試験やレビューが必要かを判断

評価指標データの分析を洗練させる

定量的な目標を達成

定量的なデータに基づいた管理

2つの重要な側面

定量的な品質目標を設定すること

開発プロセスを定量的に管理

- ・目標を達成するため

CMM レベル4 では、適切な定量的品質マネジメントが求められる

## 定量的に管理する手法

### ソフトウェア信頼性モデル (software reliability mode)

- ・試験の最終段階のエラーデータによってソフトウェアの信頼性を推測
  - ・信頼性が受け入れられるか、追加試験が必要か示す
- ・必要な信頼性目標を達成するためにどれくらいの試験が必要かを評価
- ・残念ながら
  - ・試験において試験項目の種類に制約を与える
  - ・大規模なシステムにしか適用できないことが多い
  - ・信頼性の予測には役に立つが、品質マネジメントにはそれほど価値がない

### 優れた品質マネジメントのアプローチ

#### 早期警告

- ・可能な選択肢が少なくなってしまうプロジェクトの後半ではなく、早い段階で警告を発することが出来るものである必要
- ・タイムリーな介入が可能になる
- ・予測
  - ・プロジェクトの各工程でいくつかのパラメータ値を予測することが不可欠
  - ・プロセス実行時に発生するデータ

プロセスがその能力を十分発揮できるように実施されているかを確認するのに利用できる

- ・パラメータを管理
  - ・最終製品が必要な品質を満足できるようにする

#### 工数や工期のマネジメントと非常に近くなる

- ・品質以外の2大重要成功要因
- ・納入するソフトウェアの品質目標の設定
- ・各工程で選択したパラメータ値が推測
  - ・マイルストーンが確定

予測があていれば、最終のソフトウェアの品質が必要なレベルを満たすように設定

- ・プロジェクト遂行中に、パラメータの実測値と比較し、期待通りに進捗しているかを判断

#### バグ除去率 (defect removal efficiency)

- ・品質管理活動の効果を測るのによい
- ・定義は様々
- ・バグ除去効率 (DRE)
  - ・品質管理 (QC) 活動のため
  - ・残存していたバグの総数に対する品質管理活動によって発見されたバグの数
  - ・プロセス内効率 (in-process efficiency)

プロジェクトのライフサイクル全体に対するバグ除去率

- ・残バグ密度 (defect density delivered)

プロジェクト全体のバグ混入率がわかっているならば、DRE により計算できる

- ・QC の DRE や全体の DER はプロジェクトの最後にしか計算できない

プロセスを評価し改善の余地がある分野を発見するには役に立つ指標

これだけでは品質マネジメントに適さない

プロジェクト遂行中の品質管理には使えない

## バグ予測に基づく定量的品質マネジメント

### バグの予測 (defect prediction)

定量的な品質マネジメント手法の一つ

残バグ密度 (delivered defect density)

- ・品質の目標として設定
- ・プロセスを定量的に管理して期待される残バグ密度目標を満足するようにする
- ・各バグ摘出作業におけるバグの発見数を予測
  - ・品質の目標が設定されると各工程におけるバグ数が予測される
  - ・予測値に達することにより目標とする品質が確保できる
- ・実際のバグ数を予測値と比較
  - ・開発プロセスが品質目標を達成する方向に向かっているかのベンチマーク
- ・1つのキー・ファクターに依存
  - ・どれだけ各工程におけるバグ数を正確に予測できるか
  - ・利用する概念

バグ除去効率

バグ混入率 (defect injection rate)

- ・バグの数はシステム規模の見積から推測できる

各工程のバグ混入率がわか(予測でき)るならば

各品質活動(QC)におけるDREがわか(予測でき)るならば

- ・定量的品質マネジメントの目的で利用

バグ観測曲線 (observed defect pattern)

- ・バグの発生率は工数配分と同じようなパターンを持つことがわかっている
  - ・どちらもレーリー曲線を書く
  - ・スタート時点で発見されるバグの数は少ない
  - ・徐々に増加を続け試験工程くらいでピークに達し
  - ・再び減り始める
  - ・滑らかな曲線ではなく、上り下りの階段状グラフとなる
- ・バグ発生分布 (percentage distribution)
  - ・プロセス上、バグは決められた時点で発見される

- ・ 曲線は各工程で発見されるバグ数の割合で表すことも出来る

各品質管理活動のバグ発生率

活動によって発見されたバグの割合に置き換えることが出来る

- ・ プロジェクト全体で混入したバグの総数が予測できれば、各工程で発見されるバグの数も予測できる
- ・ バグ混入率
  - ・ 1 プロジェクトのバグ総数はソフトウェア規模とともに増加

正規化した値がバグ混入率 ( 単位規模あたりのバグ数 ) を表す

- ・ 規模の見積と、バグ混入率があれば、バグ総数を見積もることが出来る
- ・ バグ混入率は予測可能

大前提

バグ混入率はプロジェクト間でほとんど変わらない

過去のバグ混入率のデータが現行プロジェクトのバグ混入率の予測に使える

他の評価指標同様にかなりばらつきがある

不確実性を念頭に置く

実データが範囲外の場合

事実はプロジェクトマネジメントの注意を喚起するのに利用

定量的品質マネジメント計画

重要

品質目標を設定すること

途中のマイルストーンでのバグ数を予測すること

過去プロジェクトのデータを取得

プロセス・データベース (PDB)

- ・ バグデータ
  - ・ 規模と工数により正規化
  - ・ 規模により正規化された情報をバグ数の予測に使うには規模を FP で見積もる必要
  - ・ FP が直接入手できない場合、工数の見積とプロジェクトの生産性予測値を規模の見積に利用する
  - ・ 工数によって正規化されたバグ混入率を生かせば、工数の見積を直接利用することも出来る

プロセス・ケイパビリティ・ベースライン (PCB)

品質目標の設定

プロジェクトの品質目標

- ・プロジェクトが「納入しようとする」バグの数
- ・受入試験におけるバグの予測値
- ・過去のデータから計算された値に設定することも出来る
  - ・標準のプロセスが修正せずに使われ、品質においても標準的な結果が期待できることが前提
- ・標準と異なる品質目標を設定
  - ・プロセスを品質目標にあわせて修正する必要がある
- ・2つの重要な情報源
  - ・類似プロジェクトの過去のデータ
  - ・PCBのデータ
- ・品質のターゲットをファンクションポイントあたりのバグ数で設定
  - ・1. 品質目標をFPで設定する
  - ・2. プロジェクトの生産性を見積もる
  - ・3. 規模をFPで見積もる(生産性の予測×見積工数)
  - ・4. 受入試験でのバグ数を予測(品質目標×見積規模)