

あなたの計測、役立っていますか？

－事例に学ぶ計測の失敗要因－

2012年10月11日

JASPIC 特別会員
岩見 好博

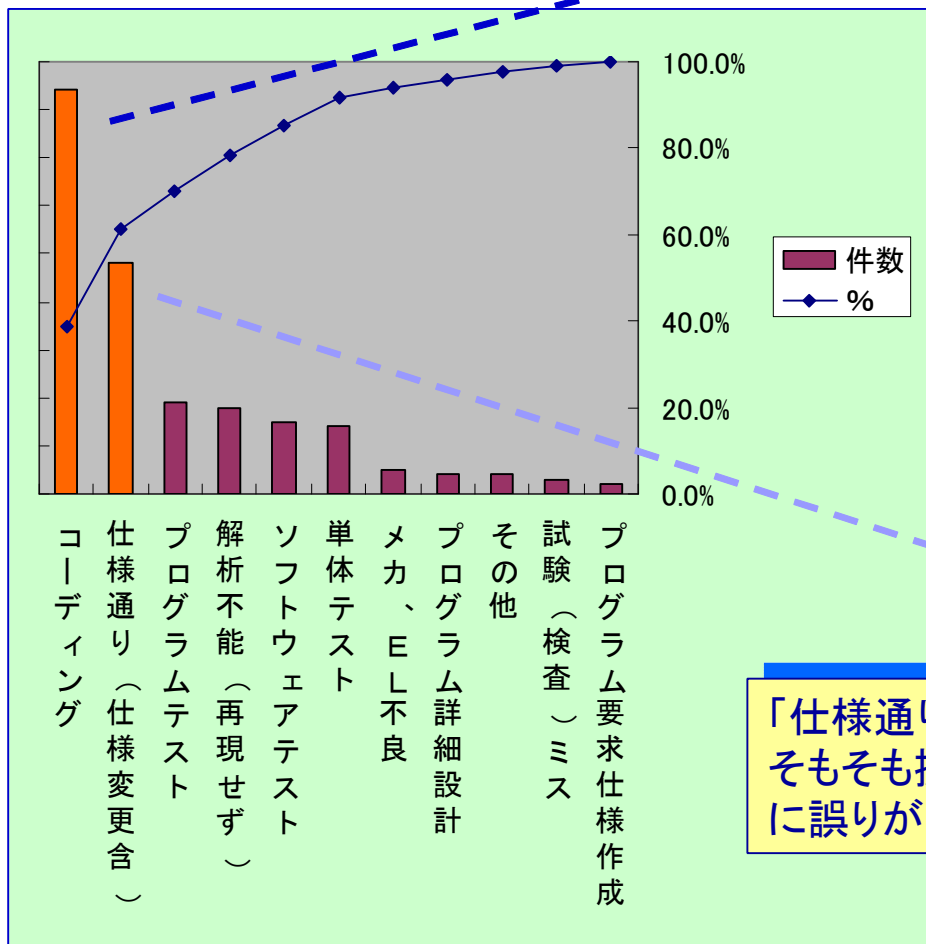
計測できないものは改善できない

- ソフトウェアプロセス改善を成功させるにはプロセスの計測が必要となる。しかし、計測が十分に活用できていない開発現場も多い。
- 例えば、計測の目的を明確にしないまま計測項目を設定すると、改善に役立つ情報を得られないだけでなく、間違った改善策につながることも。また、複数の尺度で計測しないと誤った判断につながる。
- 事例から、計測の前提条件、項目の設定、計測方法、計測結果の使い方の問題点、その改善策を示す。

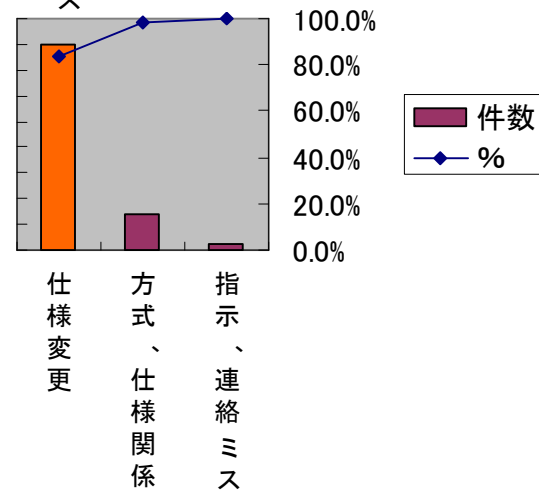
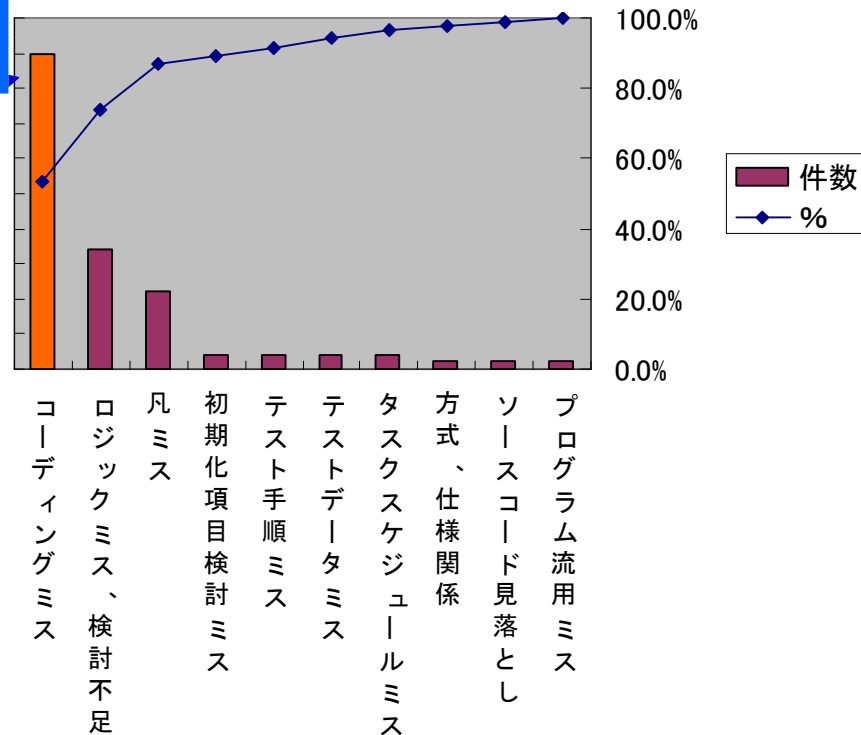
事例 1

「コーディング」とは、
原因？ 不具合作り
込みフェーズ？

原因別障害件数分析



「仕様通り」とは、
そもそも提示仕様
に誤りがあった



開発工程別見積り・実績工数

	見積工数 人日単位	実績工数 人日単位	工数比	組込み系 標準値
調査、検討		21.2	2.4%	14%
基本設計		40.3	4.6%	11%
詳細設計		2.7	0.3%	21%
試験仕様書		4.9	0.6%	
コーディング	226.7	290.9	33.3%	29%
単体試験	138.3	435.3	49.8%	
結合試験		26.5	3.0%	25%
障害仕様変更				
文書整理				
工程管理		5.6	0.6%	
その他直接		47.3	5.4%	
	365	874.7	100.0%	100.0%

事例1 改善策

- 計測する前に開発フェーズを見直す。
 - 定義したフェーズが守られないことも多い
- 欠陥の分類基準をよく吟味する。
 - 欠陥部位、原因、作り込みフェーズがよく混在している
 - まず欠陥の種類で分類し、作り込みや検出フェーズ、原因(記述)を追加情報とするとよい
- ベンチマーキング
 - 業界標準、品質のよい他部門や他社と比較してみるとプロセスの問題点が見えてくる

表 2.7 欠陥記録のログ例

欠陥型番号		
10 文書化	60	チェック
20 シンタックス	70	データ
30 ビルド、パッケージ	80	機能
40 割り当て	90	システム
50 インタフェース	100	環境

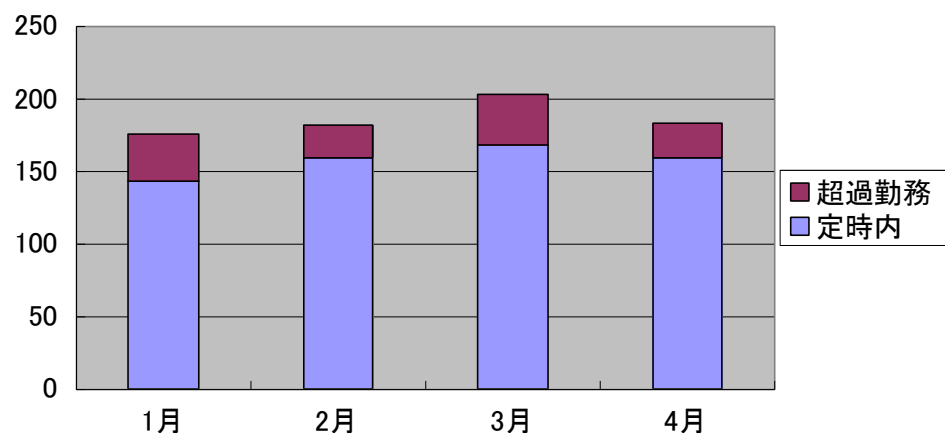
学生:	学生3	日付:	1/19
プログラム:	Standard Deviation	プログラム番号:	1
インストラクタ:	Humphrey	言語:	C

プロジェクト	日付	番号	型	作り込み	除去	修正時間	修正参照
1	1/19	1	20	Code	Comp.	1	

説明: Missing semicolon.

事例2

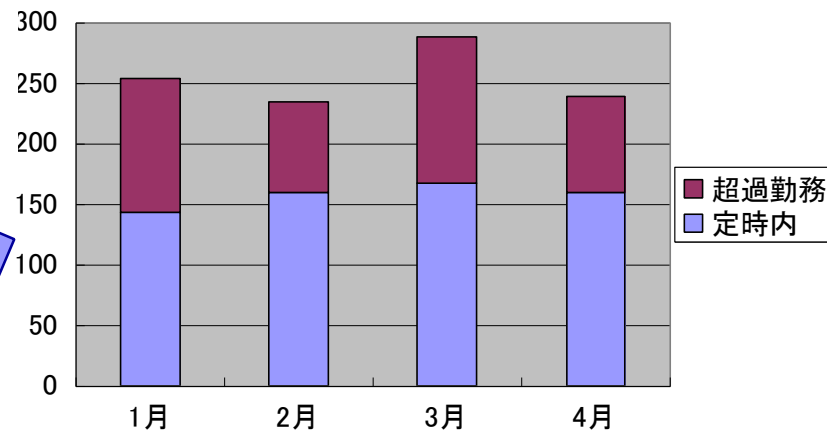
- 開発費用集計システムから提供される作業時間で生産性をみると実感よりいい数値になっていた。よく調べると残業時間が36協定時間を超えないよう調整されていた。



調整前だと生産性は
 $3,200\text{step} / 1,017\text{h} = 3.1$

生産性
 $3,200\text{step} / 744\text{h} = 4.3$

平均3.6より
いいね！

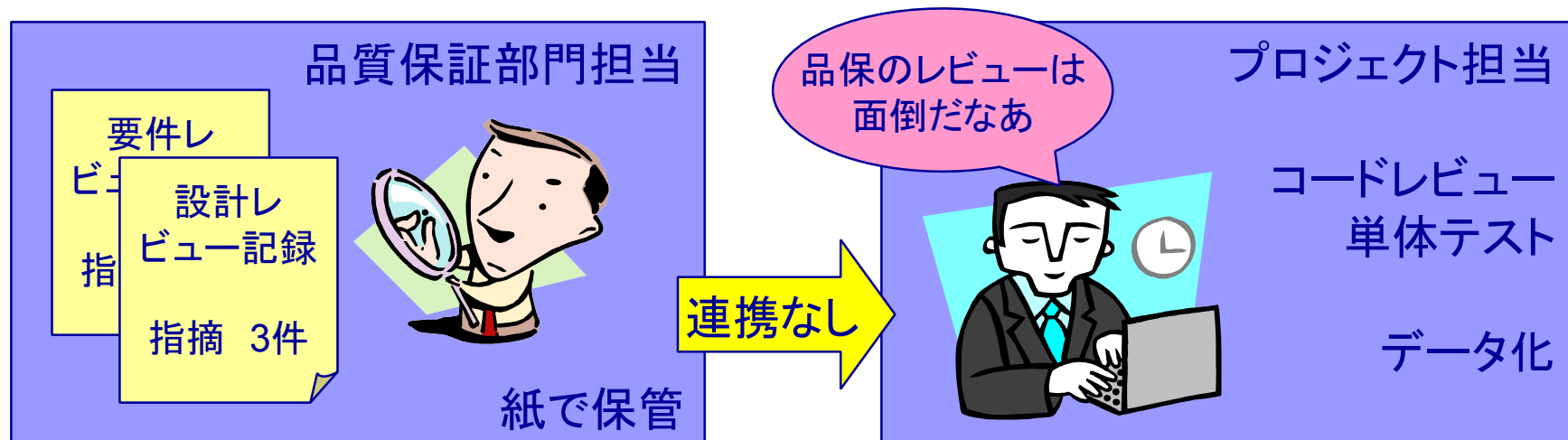


事例2 改善策

- 計測の目的を周知する。
 - 実際の生産性を知り、改善につなげるのが目的
- データの集計方法を事前によく確認する。
 - 工数集計システムの計算ロジックを検証する
 - 工数集計データの改ざんは、不正な会計処理とみなされることも。
- 発見の手がかり
 - 36協定時間を超える残業がない
 - 期末に近づくと36協定時間一杯の残業となっている

事例3

- 要件レビュー、設計レビュー記録に指摘件数が記載されていたが、品質の先行指標としては活用されていなかった。
 - 同社では設計レビューまでを品質保証部門が主査、それ以降はプロジェクトが主査するため、コードレビュー以降の不具合件数だけをデータ化していた

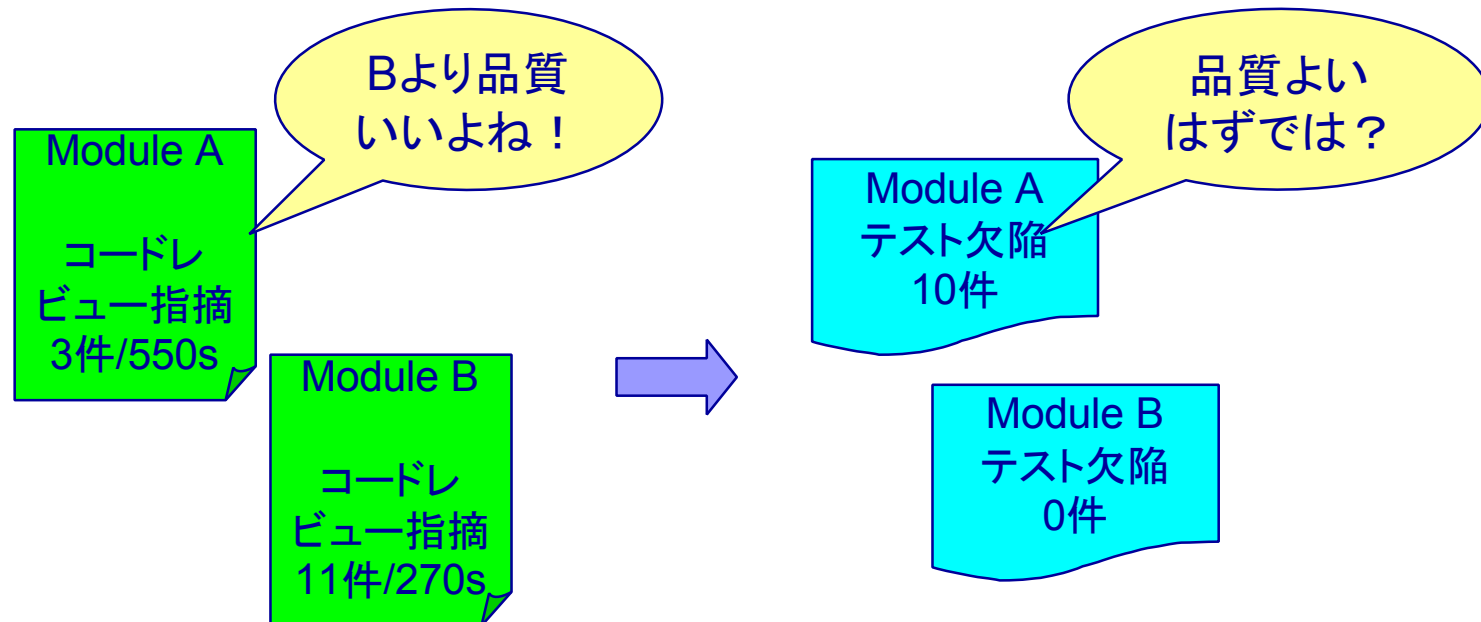


事例3 改善策

- 要件レビュー、設計レビュー指摘件数がソフトウェア品質の先行指標として有益であることを周知する。
 - 品質の悪い部分が推測でき、事前に対応できる
- プロセスデータを関係者で共有する。
 - レビュー指摘件数をデータ化する
 - 関係部門間の連携も必要
- 指摘内容も重要なので、レビュー記録は必ず残しておく。
 - 指摘内容が再発防止策に役立つことが多い

事例4

- レビュー指摘件数が品質の先行指標にならない！



- コードレビューで指摘は少なかったモジュールAに、テストで多くの欠陥が見つかった。
 - レビュー作業の品質も見ないと！

事例4 改善策

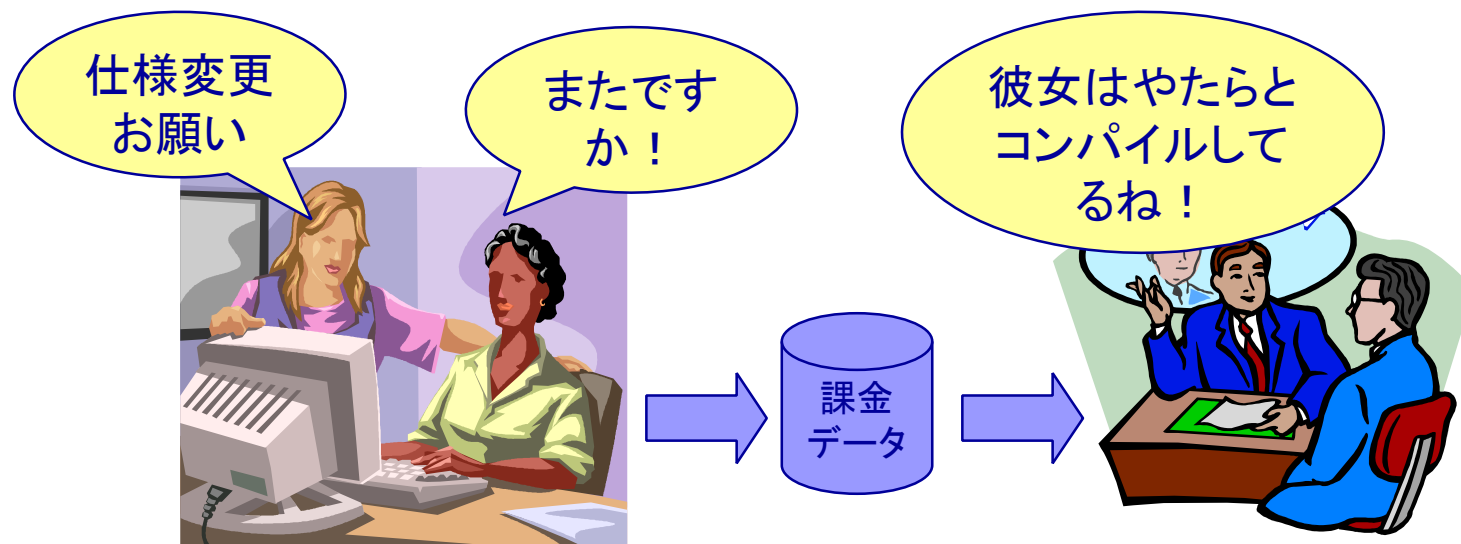
- プロセス品質をひとつの尺度だけで評価しない。
 - 関連性のある複数の尺度を用いる
- レビュー作業の品質を知るため、指摘件数だけでなくレビューレートも見る

モジュール	指摘件数	規模	時間	レート
Module A	3	550	0.5H	1,100
Module B	11	270	1.3H	207.6

- モジュールAのレビューアはスピード違反！！
- レビューレート(成果物規模/レビュー時間)が早すぎ、レビューの品質がよくなかった

事例5

- 管理者がモジュールソースコードのコンパイル回数をこっそりと開発者の人事考課に使っていた。
 - 要件や設計の詰め甘さに起因する仕様変更が多発したモジュールの開発者は、他責にもかかわらず評価を下げられ、やる気をなくしていた。



事例5 改善策

- プロセスデータを開発者の評価には使わない。
 - 正直に事実を入力してもらう必要がある
 - プロセスデータを入力しないことは責めてよい
- 評価に使っていると思われるでもいいけない。
 - でないと、正しい数値が報告されなくなる
- プロセスデータが自分の評価に使われるのではないかとの不安が計測に対する拒否反応の原因の一つ。
 - 開発リーダーは、マネジメントに個人を特定できるプロセスデータを開示してはならない

まとめ

- 計測の目的をよく理解する
 - 何のために計測するのかを考える
 - 目的を関係者に周知する
- データがおかしいと感じたら、よく調べる
 - 健全な懐疑心が必要
 - 収集方法、集計方法に問題ないか、特に既存のデータ
 - 実際の開発プロセスは妥当か
 - 複数の尺度で検証する
- プロセスデータを開発者の評価には使わない
 - そう思われるだけで正しいデータが得られない

プロセス計測のスキーム（おまけ）

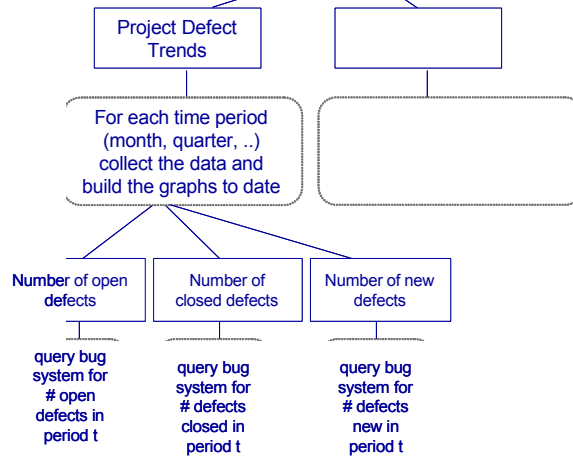
Issue: Maturity (quality, stability, growth) of the product

PM and SW Manager and Validation Manager need data about product maturity at points throughout the project to be able to predict readiness for the next stage

see diagrams – 4 charts; all have time as X axis;
1. Project Reqs Progress; 2. reqts volatility; 3. # open defects, amount of code change; 4. **defect trends (open, closed, new)**

Ready for validation if:
- code changes have stopped
- no high severity defects open
- 100% reqts passed integration (or set established by RTM [similar rules for transitions to other phases and release])

When the code is going through a lot of flux, expect to see an increase in defects; if not, check if testing is being done or other reasons. Projects usually show a standard curve for open defects; if it doesn't appear, the code may be experiencing feature creep, ... etc...



bug tracking database; CM system (for code change); initial count of requirements in SRS and SCR system (for reqts added, changed, deleted)

Information Need

Decision Criteria & Indicator

Analysis Model

Derived Measure(s)

Functions

Base Measures

Measurement Methods

The Process (Entities & Attributes)

計測の目的を定義

目的にあった
最小限の計測項目

source: TeraQuest,2002

ご清聴ありがとうございました

