
第23回ソフトウェア生産における品質管理シンポジウム
チュートリアル

ソフトウェアの品質管理を定量的に行うために

2004年12月1日

東芝 ソフトウェア技術センター

小笠原秀人

(hideto.ogasawara@toshiba.co.jp)

背景と目的

- 背景

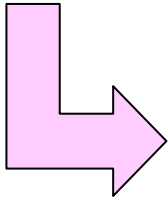
- ソフトウェア開発における定量化については、さまざまな研究がされてきた。また、定量化に関する実践結果も数多く発表されている。
- しかしながら、実際の開発の場では、必ずしも定量的な品質管理が効果的・効率的に行われているとは限らない。

- 目的

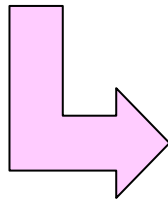
- 定量的な品質管理を実践するための基本的な考え方を理解する。
- 参加者間でのオープンな議論の中で、定量的な品質管理を実践するためのヒントを得る。

チュートリアル構成

指標について考える



どんな指標があるのか？



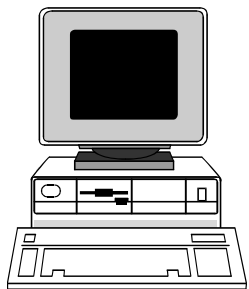
計測活動をする際に
注意すべきこと

指標を理解しよう

測ることは大切！

コンビニの売れ筋商品は？
POSシステムで、
販売情報をリアルタイムに集計

コンピュータの性能って何？



CPU

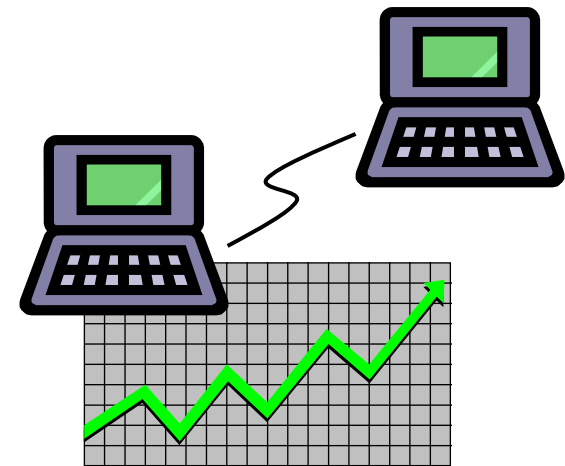
**MHz

CD-ROM

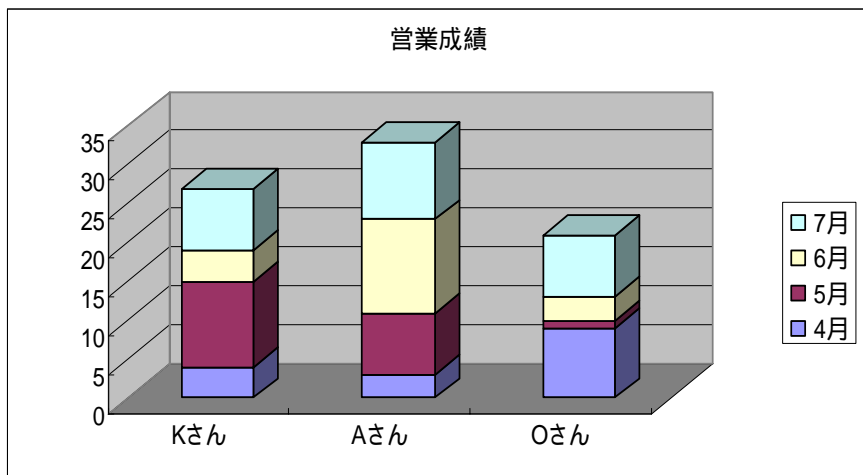
**倍速

通信速度

** kbps



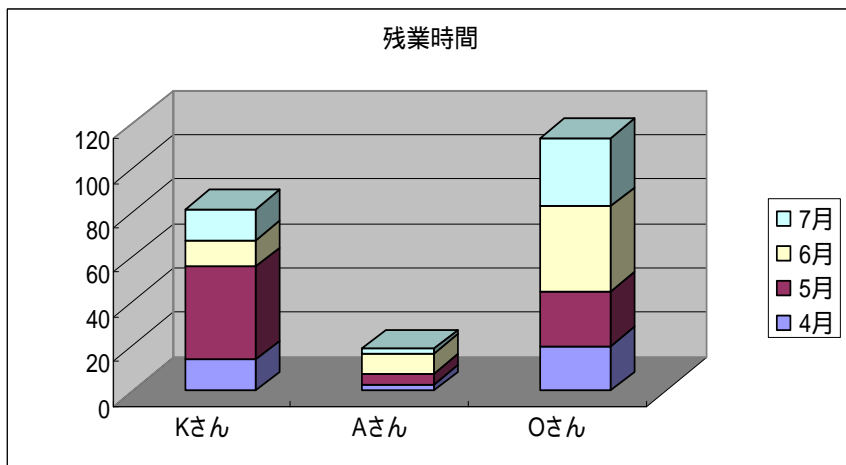
営業成績トップは誰？



あなたの健康は？

	安心	少し心配
肥満	⓪	
血圧	⓪	
心電図	⓪	
脂質		☹
糖代謝	⓪	

誰が一番仕事をしているの？



測れるもの！

製品の特徴

体の状態

作業の実績

作業時間

人々の動向

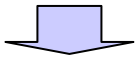
.....

世の中のいろいろなことは、
おおよそ測ることができる

指標とは？

指標：めじるし
(岩波 国語辞典 第二版)

ものごとの状態を、めじるし(指標)を使って
判断をする。これが、指標管理である。



基準と比較する

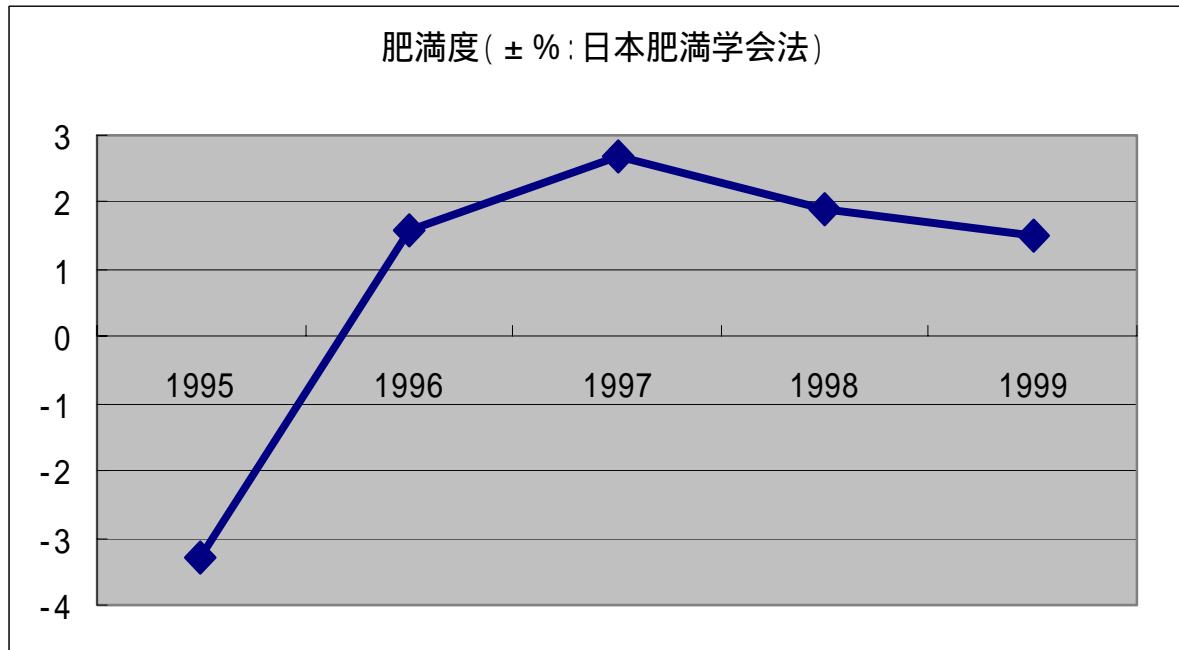
(品質)指標管理の考え方(重要!)

(品質)指標を管理するのではなく、

指標で品質を管理する。

指標のメリットは？

- 現在の状況を数値で把握できる
- 過去と比較できる
- 基準と比較して、値の良し / 悪しを判断できる



↑
± 15%の
範囲内:
理想の範囲
↓

データは組合わせて見ることが大切

単独の計測結果が指標になる場合もあるし、計測結果の組合わせで指標となる場合もある。

例) 熱があるかどうか

測るもの : 熱

指標 : 熱

基準 : 37度

判定 : 37度以上なら熱がある

例) 肥満かどうか

測るもの : 体重、身長

指標 : 肥満度 (体重と身長から計算する)

基準 : $\pm 15\%$

判定 : $\pm 15\%$ の範囲外の場合は肥満

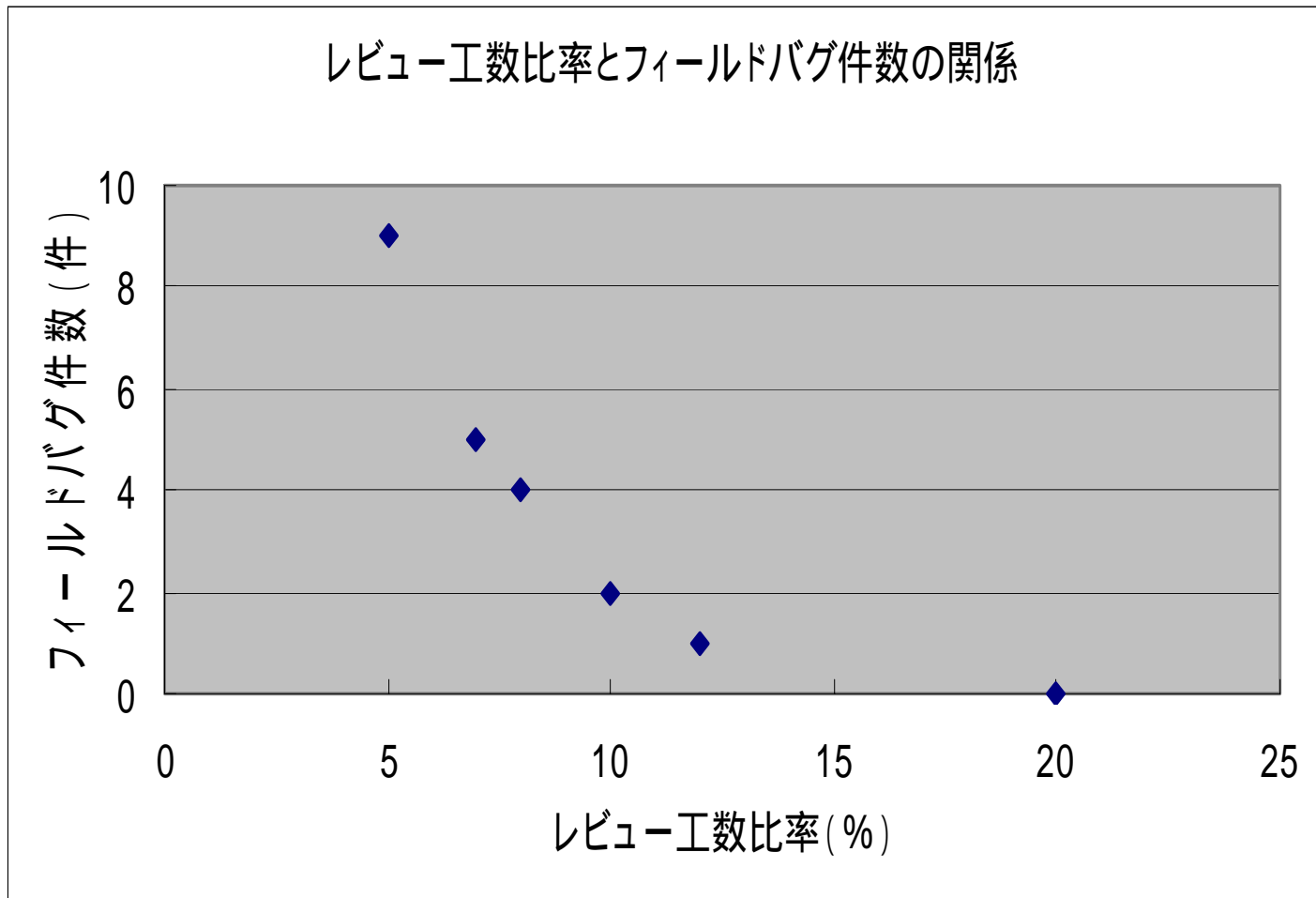
ある部門において、開発物件毎の
レビュー工数比率を測ってみました。

開発物件	レビュー工数比率
A	5%
B	20%
C	10%
D	12%
E	8%
F	7%
:	:



多い、少ない
しかわからない

フィールドバグ件数と一緒にレビュー工数比率 を見てみると...



データを知る

予定

バグ密度、etc.

実績

ソフトウェア
(規模、難易度、経験、etc.)

バグ情報
(**機能、重要度、検出工程、
検出すべき工程、修正時間、etc.)

プロセスデータ
(作業工数、レビュー時間、
テスト件数、etc.)

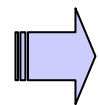
生産性、レビュー効率、etc.

信頼度推定、etc.

ソフトウェアも測ることはできる！

でも、うまく使われていない。なぜ？

- 指標の使い方がわからない
- 指標の定義があいまい
- とにかくコーディング、テスト
- 属人性が高いから測っても無駄
(結局は人の問題)



指標は使えないのか？

できるプロジェクトマネージャは、 指標を大事にする

ソフトウェア開発も総合(システム)テストの
フェーズに入りました。
途中経過のデータは以下のとおりでした。

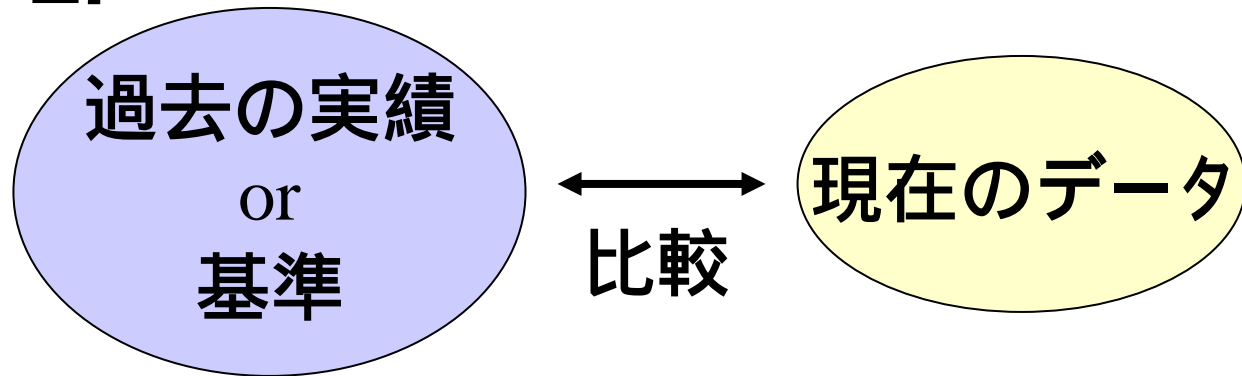
言語	C
規模	100Kstep
検出バグ数	150件
テスト項目数	1200件

このデータをどう見ますか？

これだけの情報では判断できない！

大事なことは、比較と内容の吟味。

【指標管理】



【内容の検討】

負荷テスト、大容量テスト、
性能テスト、マニュアルテスト、...

テスト種類

機能別、重要度別、
バグの種類、...

バグの内容

指標管理の目的

プロジェクトのQCDを確保する

Q:Quality, C:Cost, D:Delivery

指標管理の効果

- リスクを適切なタイミングでチェックできる
- 進捗を定量的に評価できる
- 各フェーズの達成基準を明確化できる
- 主観的ではなく、客観的な判断ができる
(データに語らせ、データに聞く)

どんな指標があるのか？

- **基本は予定と実績の比較**
 - 開発計画時に計画されるもの
 - 工程、スケジュール、成果物、etc.
 - 開発時に報告されるもの
 - 工数、スケジュールの進捗率、成果物量、etc.
- **可視化することが大事**
 - DR実施日の予定と実績、成果物の予定と実績、要件の追加・変更・削除の数、レビュー会議の予定と実績、etc.

開発計画書の例

- 基本情報
- 変更履歴
- プロジェクト情報
- 体制
- 外注計画
- 納期
- 開発内容(開発機能、工数、担当者)
- 開発環境
- 開発スケジュール(予定と実績)
- 開発ボリューム(予定と実績)
- 検査計画
- 品質目標と実績
- 進捗報告会(予定と実績)
- インスペクション計画(予定と実績)
- DR計画(予定と実績)
- 納品物管理
- リスク管理
- 構成管理
- 開発計画作成時に参考にした資料
- 開発に必要な資料
- 開発に必要な技術
- トレーニング
- 必須トレーニング受講状況

予実差の見方

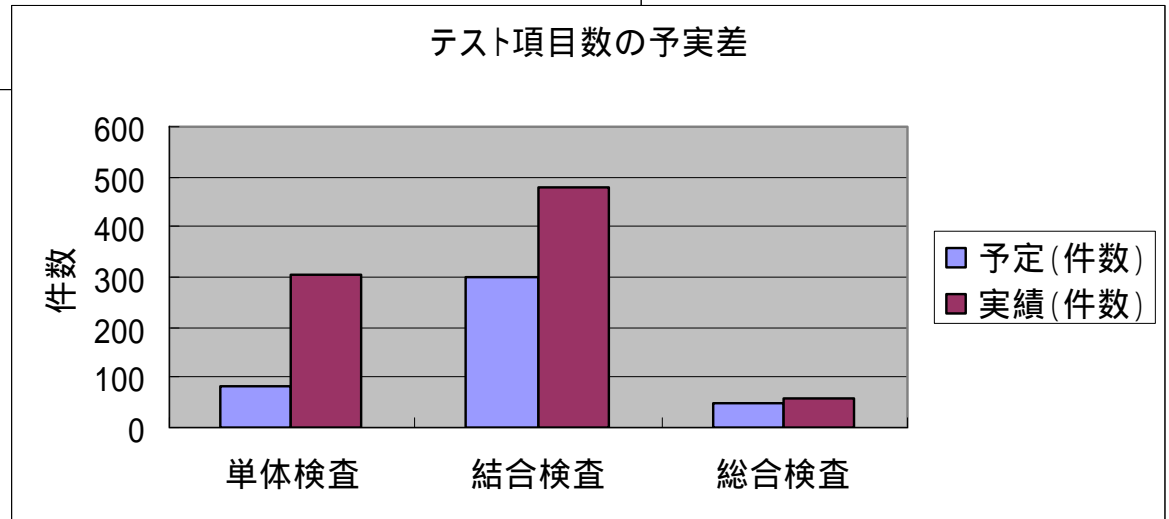
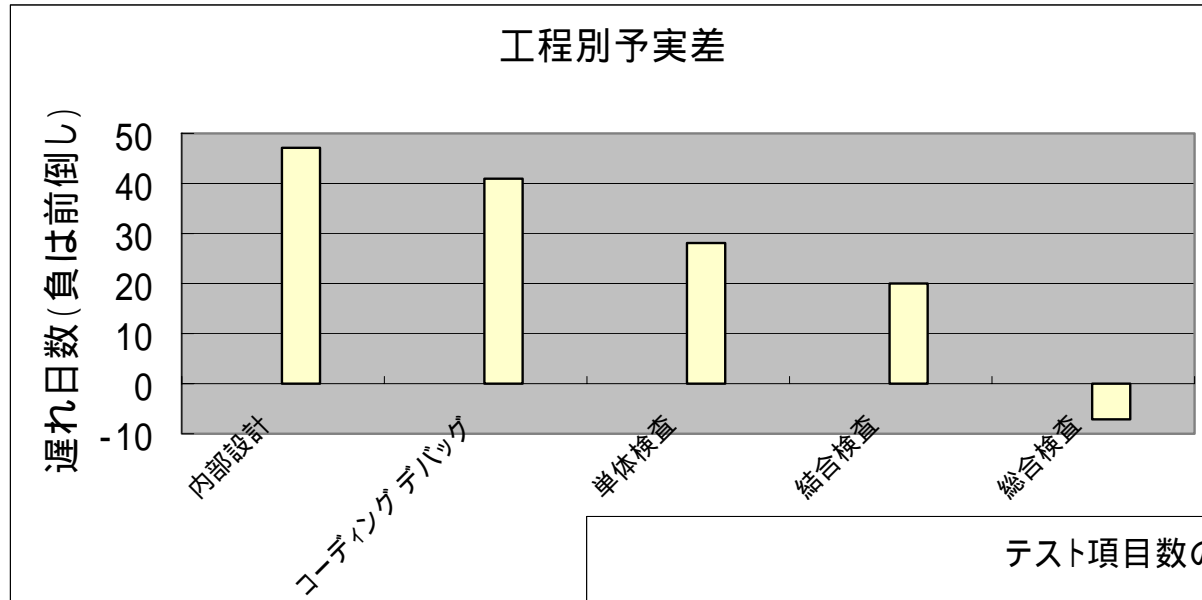
- 基本は工程の節目で、予定と実績を比較して、活動状況を判断する
- 例
 - 開発スケジュール(計画と実績)

	内部設計	コーディング/デバック	単体検査	結合検査	総合検査
完了予定(初期)	2004/4/28	2004/5/19	2004/6/1	2004/6/30	2004/7/28
完了予定(改訂)	-	-	-	-	-
完了実績	2004/5/14	2004/6/29	2004/6/29	2004/7/20	2004/7/21
差分	47	41	28	20	-7

– 検査計画と実績

		単体検査(件数)	結合検査(件数)	総合検査(件数)
予定	新規	80	300	50
	改造			
	流用		1000	
	全体	80	1300	50
実績	新規	307	477	59
	改造			
	流用		1046	
	全体	307	1523	59

可視化することが大事

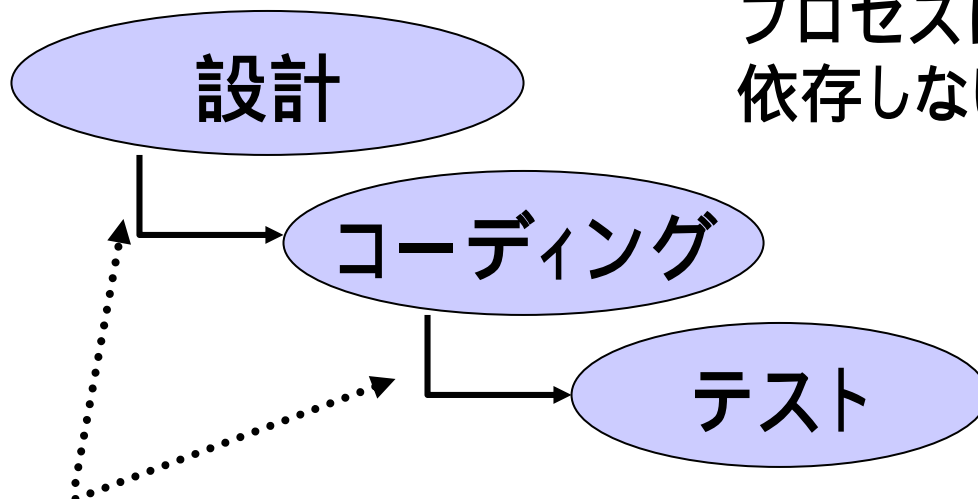


データを上手に活用するポイント

- データを重複して入力させない
- 入力されたデータをいろいろな用途で活用する
- 必ずしもツール(システム)が必要ではない。ツール化は、プロセスが固まってから。
 - 例: Excelシートの開発計画書・報告書
 - 計画と実績の入力 プロジェクト状況報告シート(グラフ表示) プロジェクトデータ(プロジェクトデータベースの元データ)
 - プロジェクトデータは、プロジェクト終了後、プロセスデータベースとしてライブラリアンが蓄積

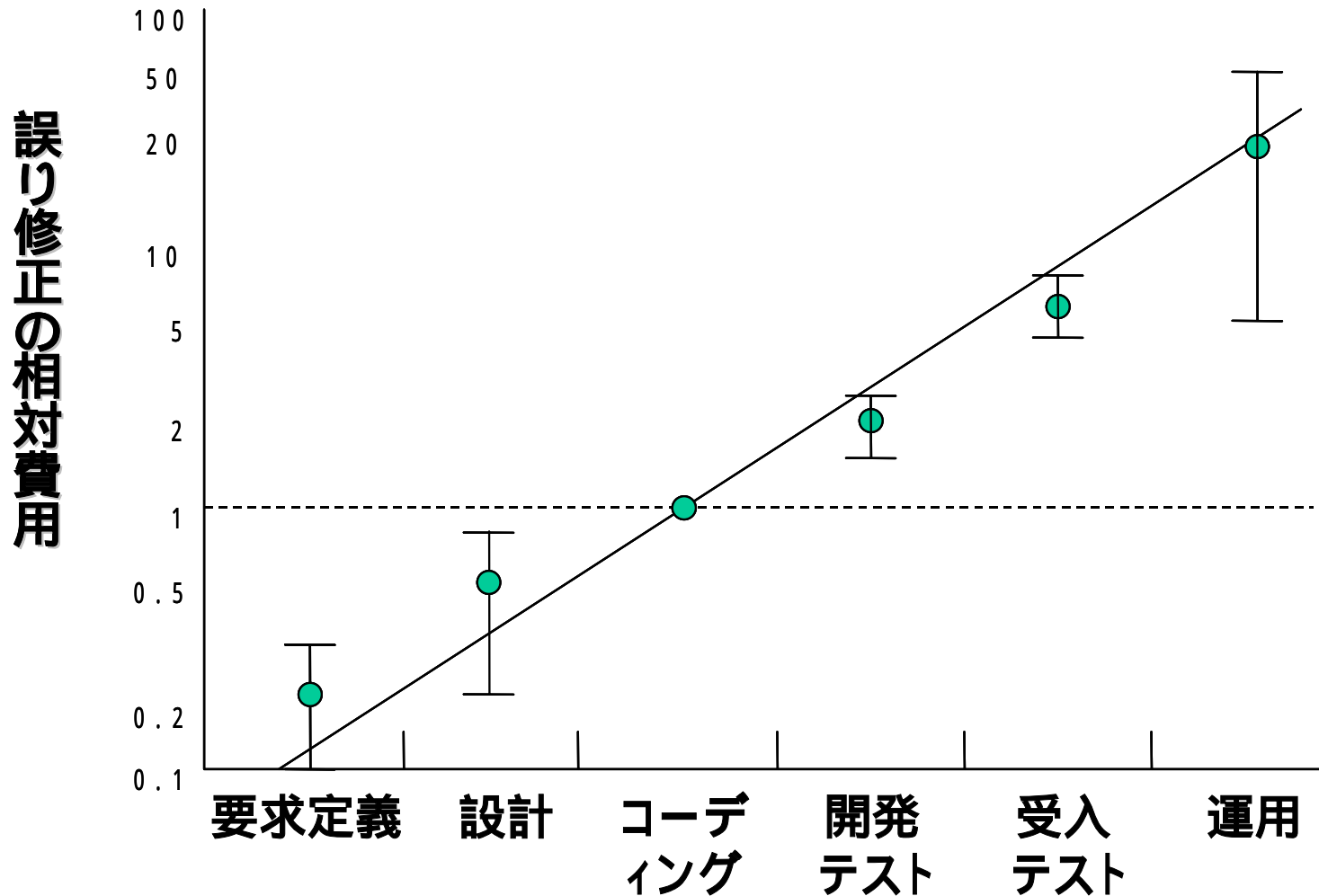
上流工程での品質保証の考え方

この流れは、
開発ライフサイクル
プロセスには
依存しない！

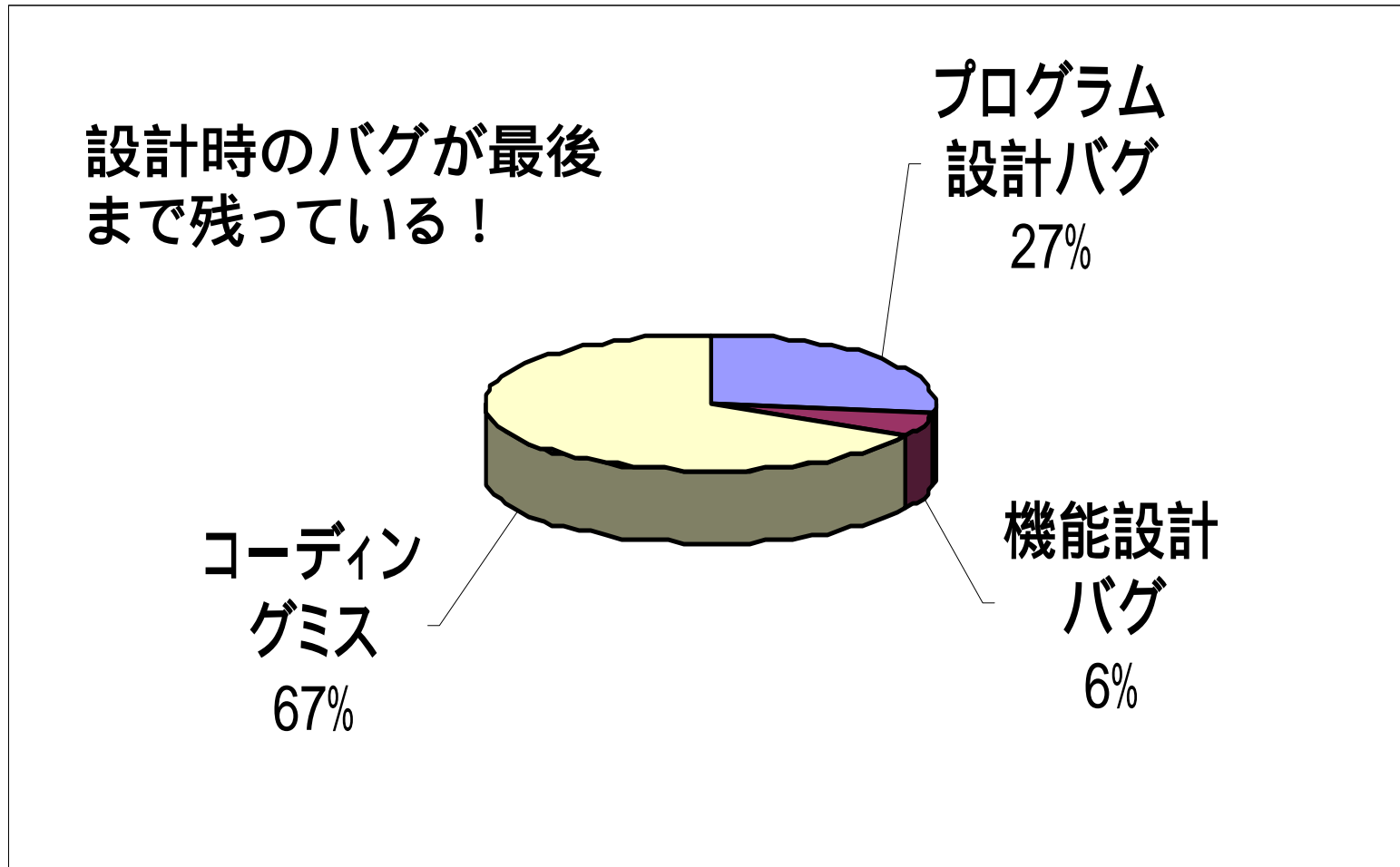


バグを作り込まない！
『後工程はお客様』の考えが大事

エラー修正に要するコスト



結合試験以降でのバグ件数比率

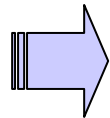


伊鹿倉、若木：“バグ予測法適用による品質評価”、第16回ソフトウェア生産における品質管理シンポジウム、日科技連、1997

上流工程から品質を作り込むには

- 要求仕様、設計書のレビュー
- テスト前のコードレビュー
- ユーザーズマニュアルのレビュー
- テスト仕様書のレビュー

など、主要な成果物のレビューを
タイミングよく、適切な方法で行なうこと。



レビューがもっとも大事

中流～下流工程での品質保証の考え方

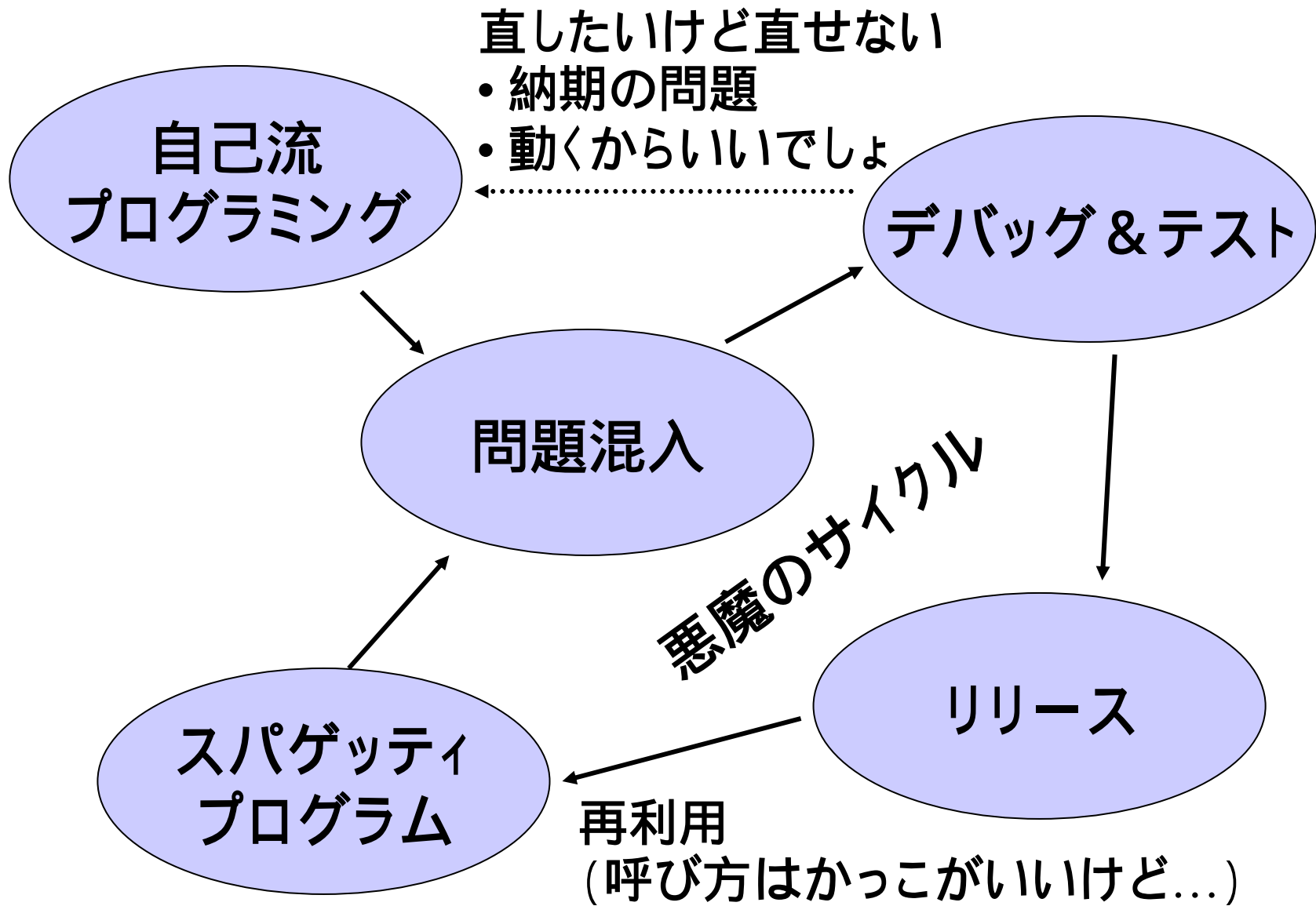
バグは出すもの。

バグを憎んで、人を憎まず！

検出されたバグ情報をもとに、品質、進捗、工数、納期などを、さまざまなデータを駆使して、いろいろな観点から予測することが大切。

こんなソースプログラムをどう思いますか？

- 1モジュールのサイズが1000ステップ以上
- モジュール構造が複雑
- コメントがない
- インデントがされていない
- 生データ(20, 124, 777など)がたくさんある
- 意味のない変数名(xxx, yyyなど)がたくさんある
- 古いバージョンのコードがコメントになっている
- フルパスでインクルードファイルが指定されている



QACの品質指標(1) 静的解析結果と修正率の例

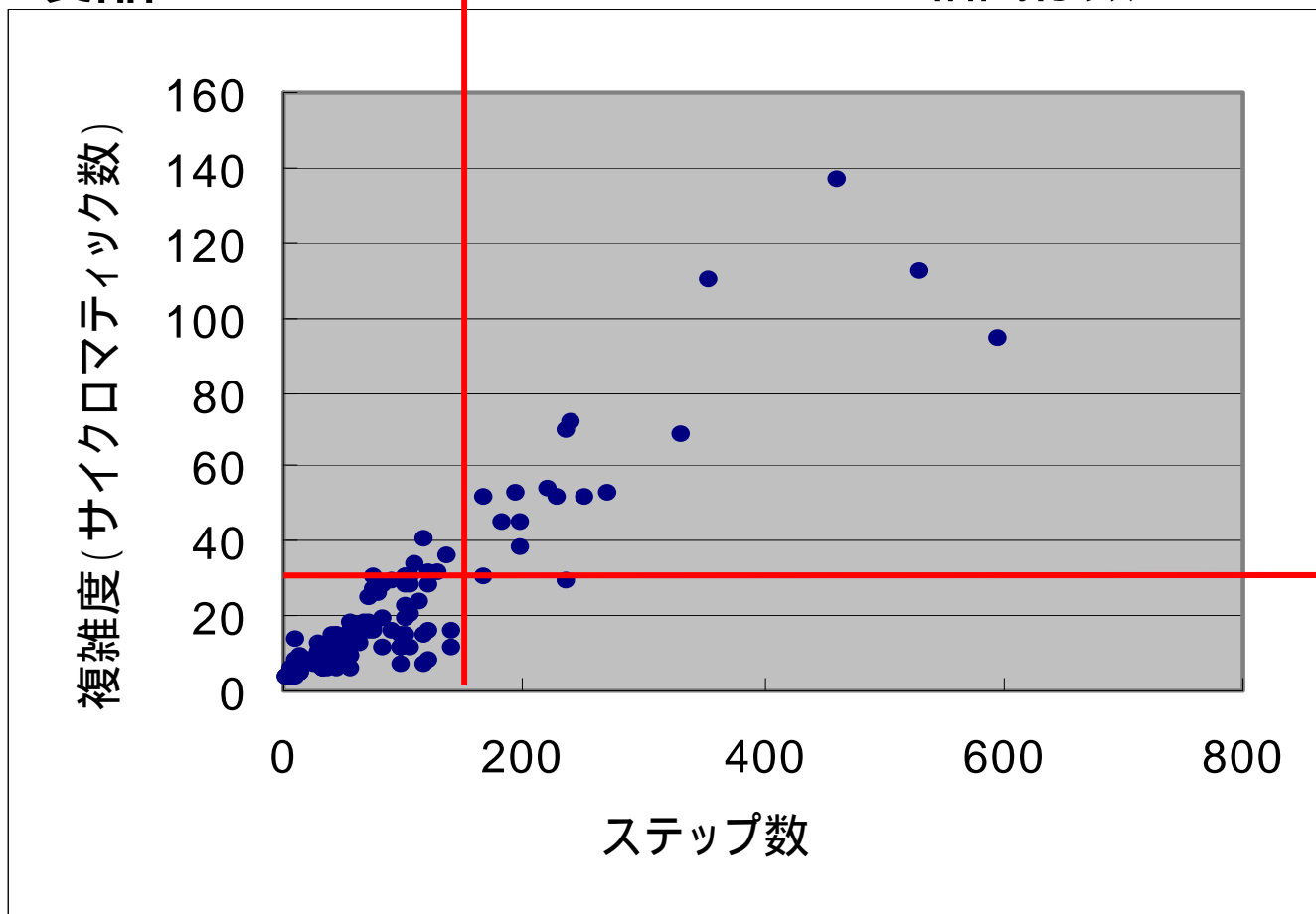
警告メッセージ		出現 件数	対応 件数
番号	内容		
3611	‘***’は内部リンクと外部リンクを同時に持ちます。	3	3
2002	用心すべきです。‘default’をこの‘switch’文に追加して下さい。	13	2
2003	すべての‘case’文は明示的な‘break’文で終わるべきです。	24	11
3409	マクロ本体に演算子が含まれています。マクロ本体をカッコで囲んでください。	16	8
合計		56	24
修正率		43%	

QACの品質指標(2) ステップ数と複雑度の例

製品A

150

モジュール数:132
相関係数:0.92

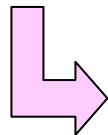


30

テスト工程での品質指標

効率的、効果的なテスト実施に必要なこと

- テスト仕様書の作成、レビュー
- 設計 / 開発チームとテストチームの役割分担明確化
- 不具合管理ルーチンの確立
- 品質指標にもとづくテスト管理



テスト項目密度、テスト工数比率
バグ密度、バグ修正率 など

バグデータは宝の山！

そこから何がわかるのか？

低品質なモジュール、重大なバグの出現頻度、
収束の時期、人の能力(適性)、
バグの入り込んだ時期、
バグの発生間隔、バグ修正時間、...

⇒ 集めないことには直感でしか言えない

バグの内容を分類(層別)しよう！

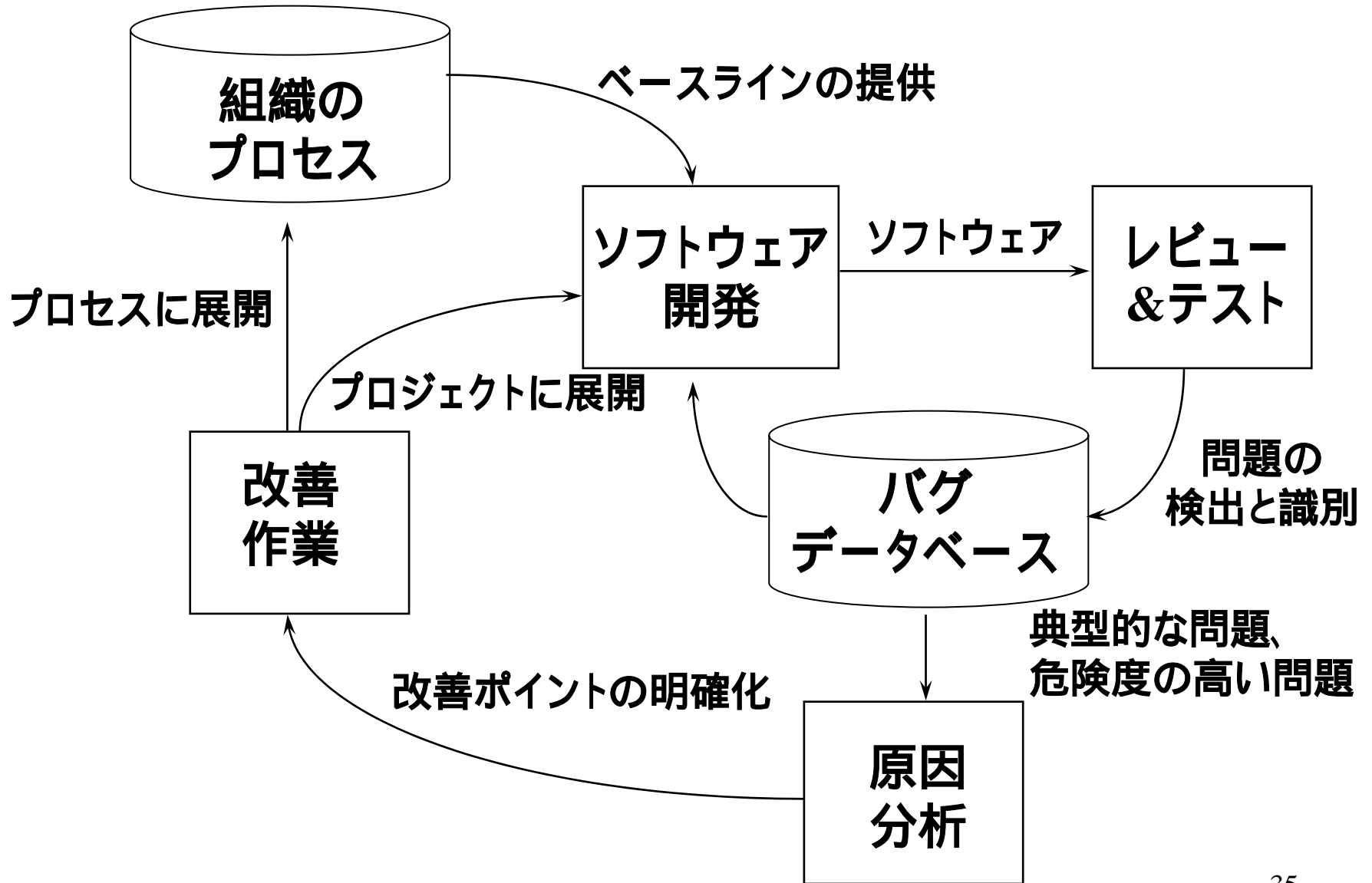
バグ情報の例

- (1) 試験区分
- (2) 発見日、修正日、発見者、修正者
- (3) 発生頻度
- (4) バグランク
- (5) バグ原因
- (6) 変更理由
- (7) 問題の混入時期・発見すべき工程

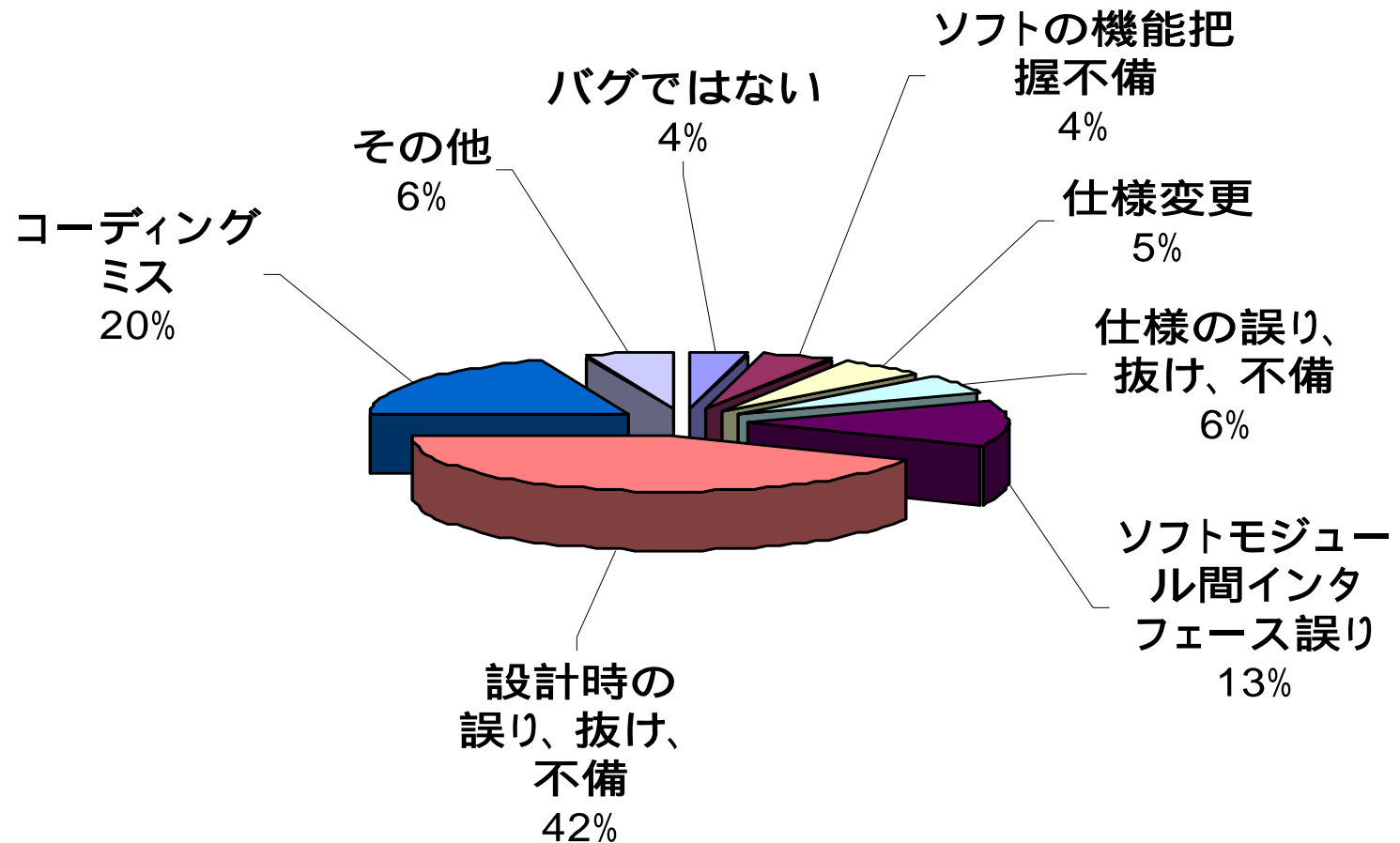
バグ原因の例

- コミュニケーション誤り
- 場所や処理の抜け
- タイミング問題
- 初期化エラー
- 不十分なチェック
- 境界条件忘れ
- 順序誤り
- 仕様の誤解
- その他

原因結果分析までつなげることが大事



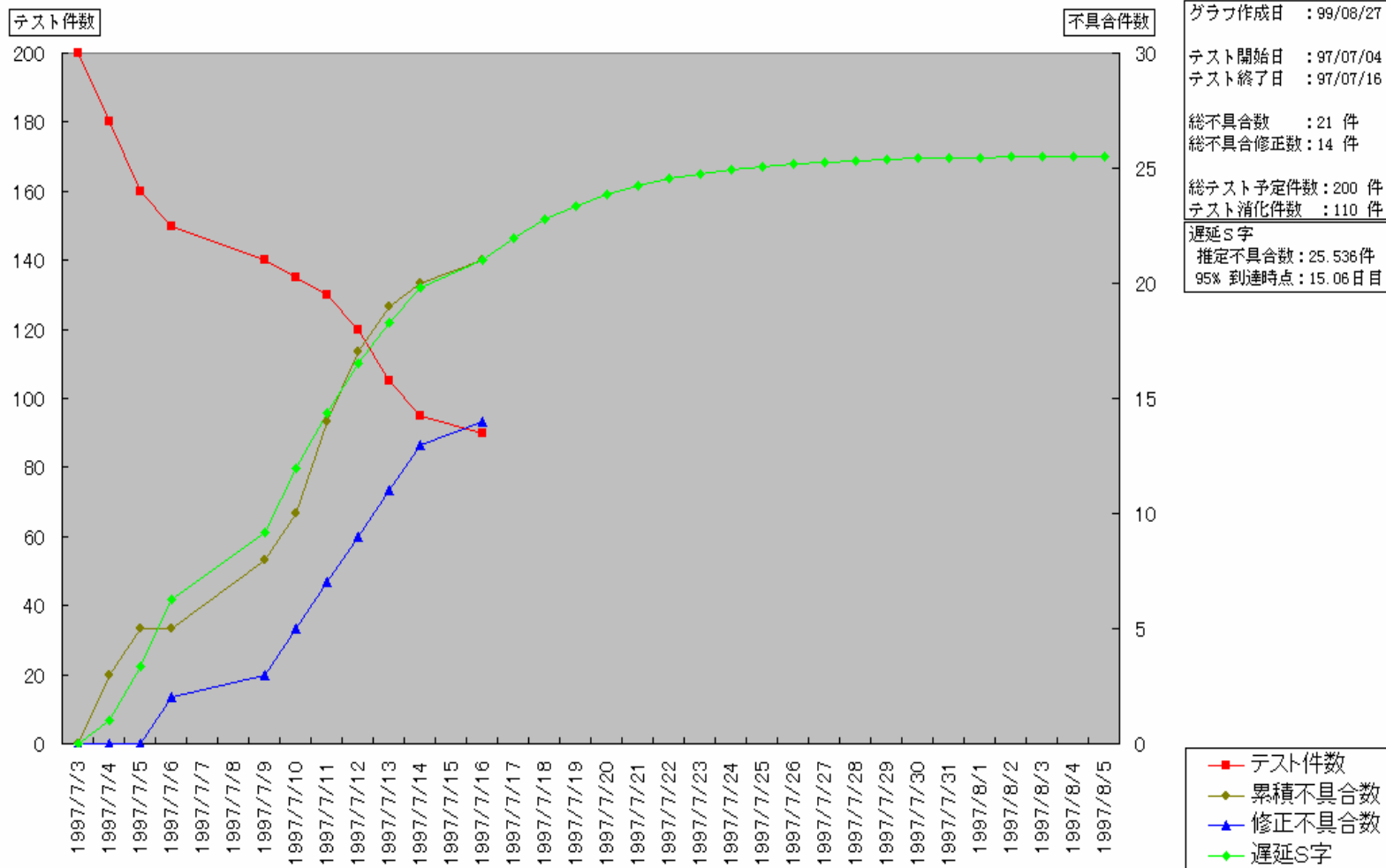
バグデータの分析例(1)



検出バグの原因分類

バグデータ分析例(2) 信頼度推定曲線

累積不具合数とテスト消化件数



上記グラフは信頼度推定ツールSQATPで作成したもの。
 SQATPについては、以下のサイトを参照のこと：

http://www.toshiba.co.jp/tech/software/sqatp/index_j.htm

過去から学び、未来を予測し、現在を把握する

ソフトウェア工学の世界では、米国を中心に、ソフトウェア品質計測が行なわれ、さまざまなデータが蓄積されています。

発生源別のバグ除去率(/Kstep : C言語)

発生源	潜在バグ	除去率(%)	引渡し時の潜在バグ数
要求定義	7.8	77	1.8
設計	9.7	85	1.5
ソースコード	13.7	95	0.7
ユーザ用文書	4.7	80	0.9
誤修正	3.1	70	0.9
合計	39.0	85	5.5

デバックの効果

工程	予想バグ数 (件/KStep)	検出バグ数 (件/KStep)	残存バグ数 (件/KStep)
単体デバック	10 ~ 20	6 ~ 16	3 ~ 4
結合デバック	3 ~ 4	2.4 ~ 3.2	0.6 ~ 0.8
システムデバック	0.6 ~ 0.8	0.5 ~ 0.7	0.1以下
運用テスト	0.1以下	0.05	0.05以下

石井康雄 著:「ソフトウェア工学入門」、pp.206、日科技連出版、1989.4

製品開発における品質管理に、このようなデータを活用できる仕組み(データの収集・蓄積方法とその使い方)の確立を目指してください。

計測活動実施上の留意点

• 計測項目の選定と活用のポイント

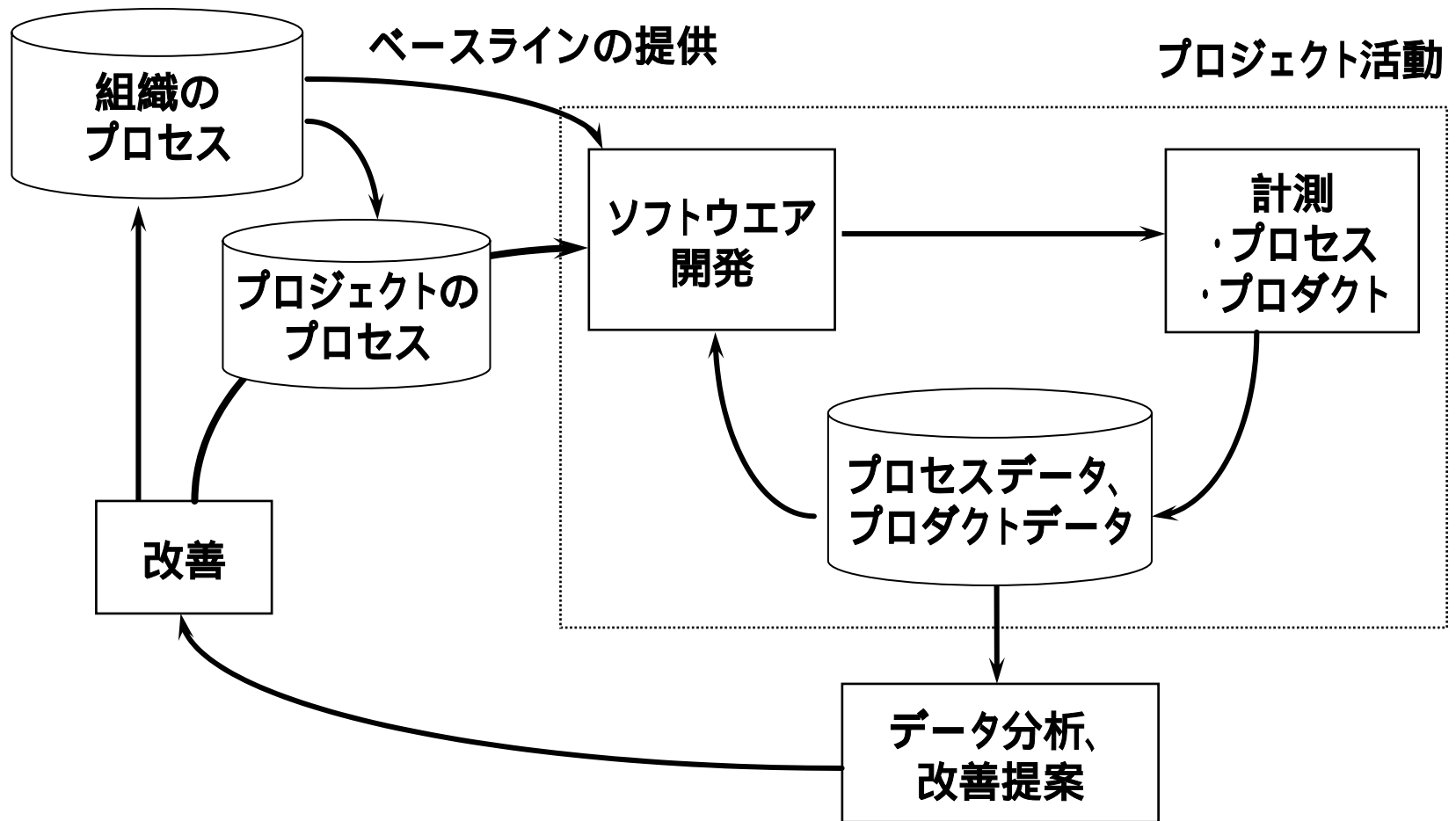
– 選定のポイント・計測の目的が明確か？

- 何を達成しようとしているか(目標)を確認し、どのようなデータが必要かを検討する
- 明確な目標を決めずに、データを集め過ぎない

– 活用のポイント・フィードバックは十分か？

- データをとりっぱなしにしない
 - データを分析し、プロジェクトや組織で活かせるように結果をフィードバックする
- どのタイミングで使うのか
 - 毎週、毎月、プロジェクト終了時、半年、1年、...
- データの使い方には気をつけて
 - 横並びで評価しない
 - 時間軸での変化を見る(同じデータを継続して収集することが大事)
 - 結果のデータのみを見てすぐ悪いと判断しない
 - » データを解釈、掘り下げて、次の改善へ結びつける

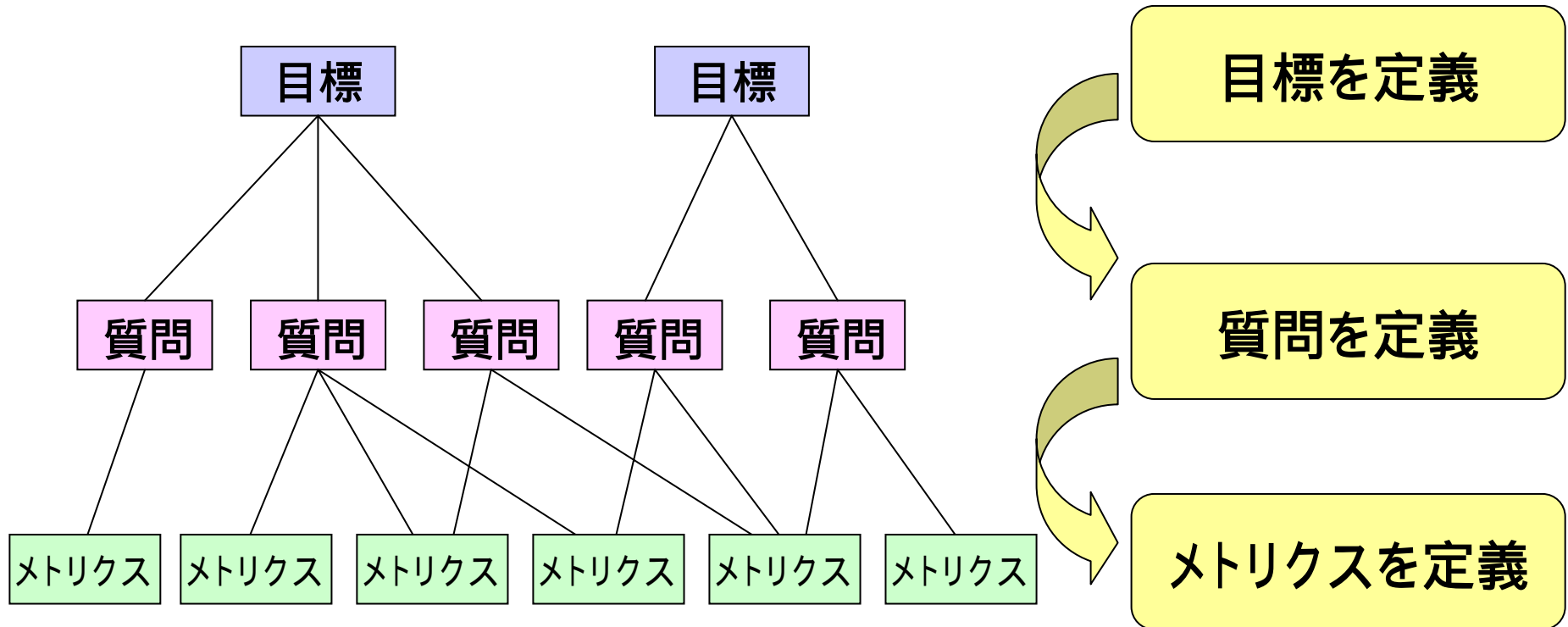
フィードバックが大切



GQM

- Goal/Question/Metric :
 - ソフトウェアのメトリクスを計測の目標と質問から得るという目標指向の測定法
- Goal(目標) :
 - 何を知り、改善したいかを定義
- Question(質問) :
 - 目標の達成を確認することができる質問を定義
- Metric(メトリクス) :
 - 質問に答えるために必要なメトリクスを定義

GQMの階層構造



- 目標 質問 指標 メトリクスの順番に定義
- 同一のメトリクスで複数の質問に答えることもできる

GQMの例

目標

意図：見積り精度の向上
論点：改善
対象(プロセス)：見積りプロセス
観点：マネージャの視点、SEPGの視点

必要に応じて、
サブ目標にブ
レークダウン

質問だけでなく、
目標を達成して
いるかどうかを
把握するのを助
ける傾向指示図
の図やグラフを
考えるとよい

サブ目標

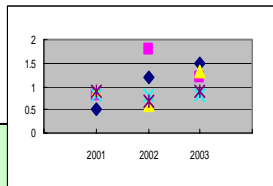
工数(コスト)の見積り精度の向上

サブ目標

測定リポジトリの利用率向上

質問

前年度と比べ、工数(コスト)
の見積り精度は上がったか？



メトリクス

- 工数(コスト)の見積りと実績(年度別)

質問

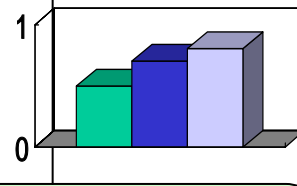
見積りに測定リポジトリは、
役立っているか？

メトリクス

- 工数(コスト)の予実差率、偏差(製品別)
- データ蓄積数(製品別)

質問

測定リポジトリは、
利用されているか？

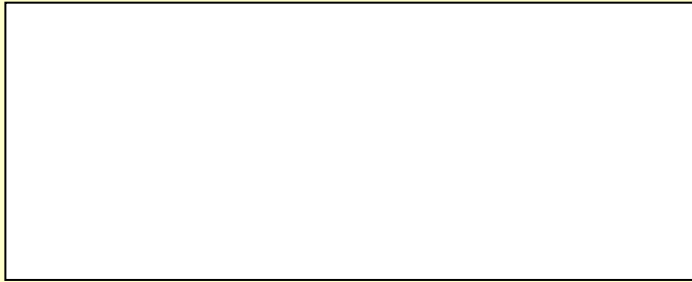


メトリクス

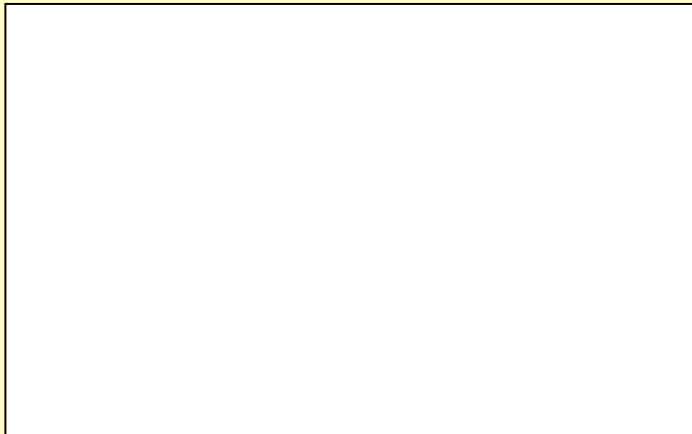
- 見積りデータの入力率
- 実績データの入力率

計測の目的と可視化のイメージ

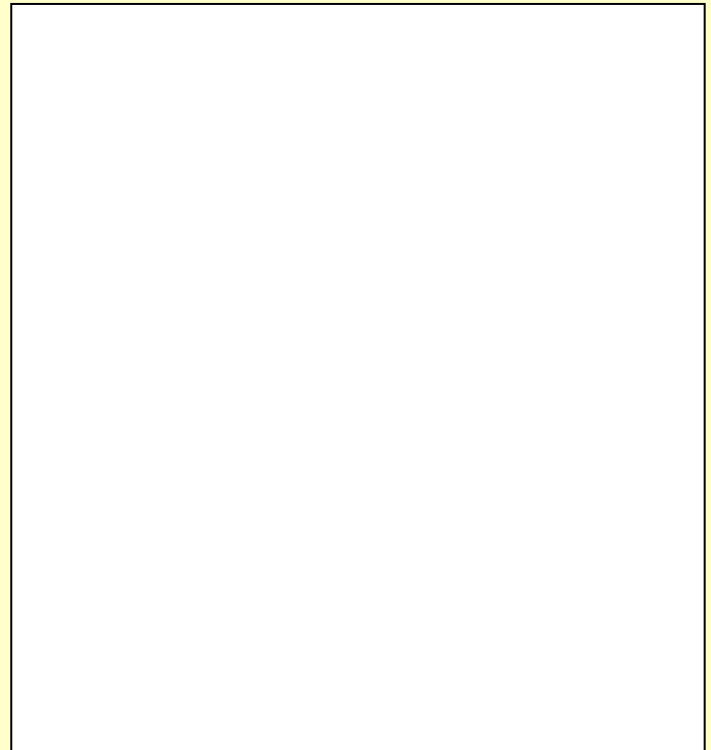
所属組織においてQCDのうち最も重要なのは、どれですか？



その理由



選択したQCDが改善しているイメージをグラフや図で定量的に示して下さい。



Goodbye

the end