

第3期 品質保証部長の会 成果発表会

ソフトウェア品質保証の肝
～品質保証の仕組みを効果的に運用するための
経験に基づいた勘所～

2012年12月4日
SQiP品質保証部長の会
グループ1

【参考】 品質保証部長の会 グループ1メンバ

大石 一士	キヤノン電子株式会社
岡本 卓	株式会社日立ハイテクソリューションズ
小田 明	株式会社日立製作所
鎌倉 洋一	株式会社富士通アドバンストソリューションズ
川田 葉子	株式会社構造計画研究所
佐藤 孝司	日本電気株式会社
藤川 昌彦	アズビル株式会社
森本 広美	富士通株式会社
渡邊 範夫	株式会社菱化システム

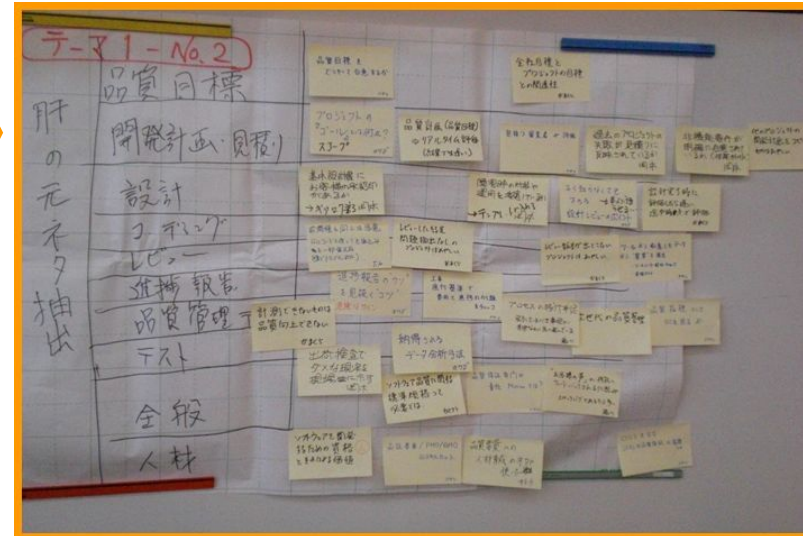
目次

- ◆ 当グループの活動
- ◆ “ソフトウェア品質保証の肝”とは・・・
- ◆ 具体例
 - ① 品質目標に関する肝
 - ② メトリクスに関する肝
 - ③ テストにおける品質判断に関する肝
- ◆ その他の肝の紹介（「ソフトウェア品質保証の肝」本編から抜粋）
- ◆ まとめ
- ◆ メンバの感想 ← ※9/13 SQiPシンポジウム発表以降に追加

当グループの活動

◆ テーマ検討（3月）

- 品質保証に関する悩みを抽出
→ 広範囲に、たくさん出てきた
→ “この際、全部、整理してみよう”



◆ 内容の検討(4月～6月)

- 1件毎に悩みの内容と、各メンバーの経験をもとに肝は何かを検討
- グループ1以外のメンバーからも、幅広く肝を収集

◆ 結果まとめ(7月～8月)

- 「ソフトウェア品質保証の肝」として検討結果をまとめた



返 数: 005
発 行: 2012/07/03

ソフトウェア品質保証の肝

～ 品質保証の仕組みを効果的に運用するための経験に基づいた勘所集 ～

SQIP ソフトウェア品質保証部長の会 3期グループ1

1

“ソフトウェア品質保証の肝”とは・・・①

◆ “品質保証プロセス/仕組みはわかっているが、現場でうまく運用できない”

- 品質保証部門長の“悩みが尽きない”!
- “わかっちゃいるけど、うまくいかない”!



◆ 一方、これらの悩みを、豊富な経験から解決してきた事例も、たくさん持っている



⇒ これらの悩みと解決ノウハウを収集し、“肝”として整理した

“ソフトウェア品質保証の肝”とは・・・② 肝の一覧

“品質保証の肝” 一覧							
1.品質目標	①品質目標が立てられない	②品質目標はお仕着せではうまくいかない	③全社目標とプロジェクトの目標との関連性	④品質目標はリアルタイム評価すべき			
2.開発計画	①「計画書をタイムリーに更新しない		②他のプロジェクトの開発計画のコピペは怪しい		③非機能要件が明確に合意されているか		
3.見積もり	①見積もりの評価はベテランやデルファイ法を使う			②過去のプロジェクトの失敗が見積もりに反映されているか			
4.設計・コーディング	①お客様の声を機能へフィードバックする	②基本設計書にお客様の承認印があるか	③障害時の対策や運用を考慮しているか	④“前機種と同じ設計”は要注意。	⑤ツールに使用されるな	⑥設計書はどこまで書くべきか	⑦コード変更の影響範囲を見積もれない
5.レビュー	①機能面以外の設計レビューのポイント	②レビューで問題指摘が無い	③レビュー状況の評価は早めに行う	④レビュー会議に関係ない人が	⑤レビュー会議で、初めてレビュー対象	⑥レビューの場で、検討会が	⑦レビューをどう評価して
6.テスト	①出荷検査で品質の実態を現場に示す		②テストの見積りができない	③テストを終了していいかどうか判断が出来ない		④テストフェーズが区切れない	
7.進捗報告	①進捗報告のウソを見抜くコツ		②まずい進捗報告の例	③プロセスの移行判定で完了しているべき事項が未決なのに先へ進んでいるのは許してはいけない			
8.品質管理	①計測できないものは品質向上できない	②納得されるデータ分析手法とは	③品質指標として何を見るか	④基準値は閾値。アクション無ければ意味無し	⑤バグ分析で品質を見極める		
9.人材	①ソフトウェアを開発するための資格とそれによる価値		②品質要員・PMO/QMOのスキルセット		③品質改善推進者を必要と認めつつ、その人材を確保することに消極的		
10.全般	①品質保証部門の責任ミッションとは	②組織の品質目標の達成が品質保証活動の目的	③ISO9001が形骸化	④品質システムのPDCAが上手く回らない	⑤開発部門と品質保証部門は犬猿の仲	⑥開発のアウトソーシングでは何を管理	

10カテゴリ、44個の肝

“ソフトウェア品質保証の肝”とは・・・③ 全体イメージ

特徴:
全て、現場の悩み(本音)であり、
現場で実践した解決方法・ノウハウ(肝)
である

具体例

①品質目標に関する肝

②メトリクスに関する肝

③テストにおける品質判断に関する肝

具体例① 品質目標に関する肝(1)

◆ 悩み

- 品質目標がどんなものかわからない
- 品質目標が無くても開発できるので、品質目標を立てようと思わない
- 品質目標が行動目標になってしまうので、定量的な判定が難しい
- そもそも、品質データが集められない

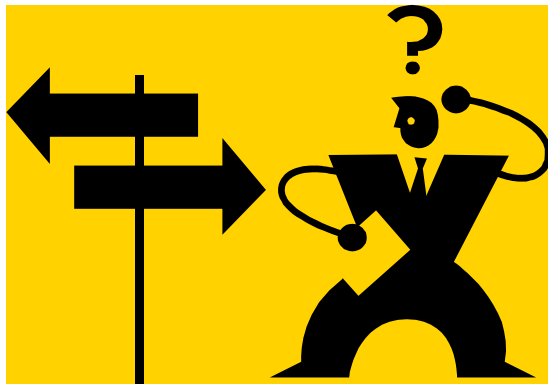


■ 例えば・・・

- 「レビュー工数は XX人H/ページ以上」、
「バグ抽出数を XX件/規模以上」
を目標と決めているが、うまくいかない

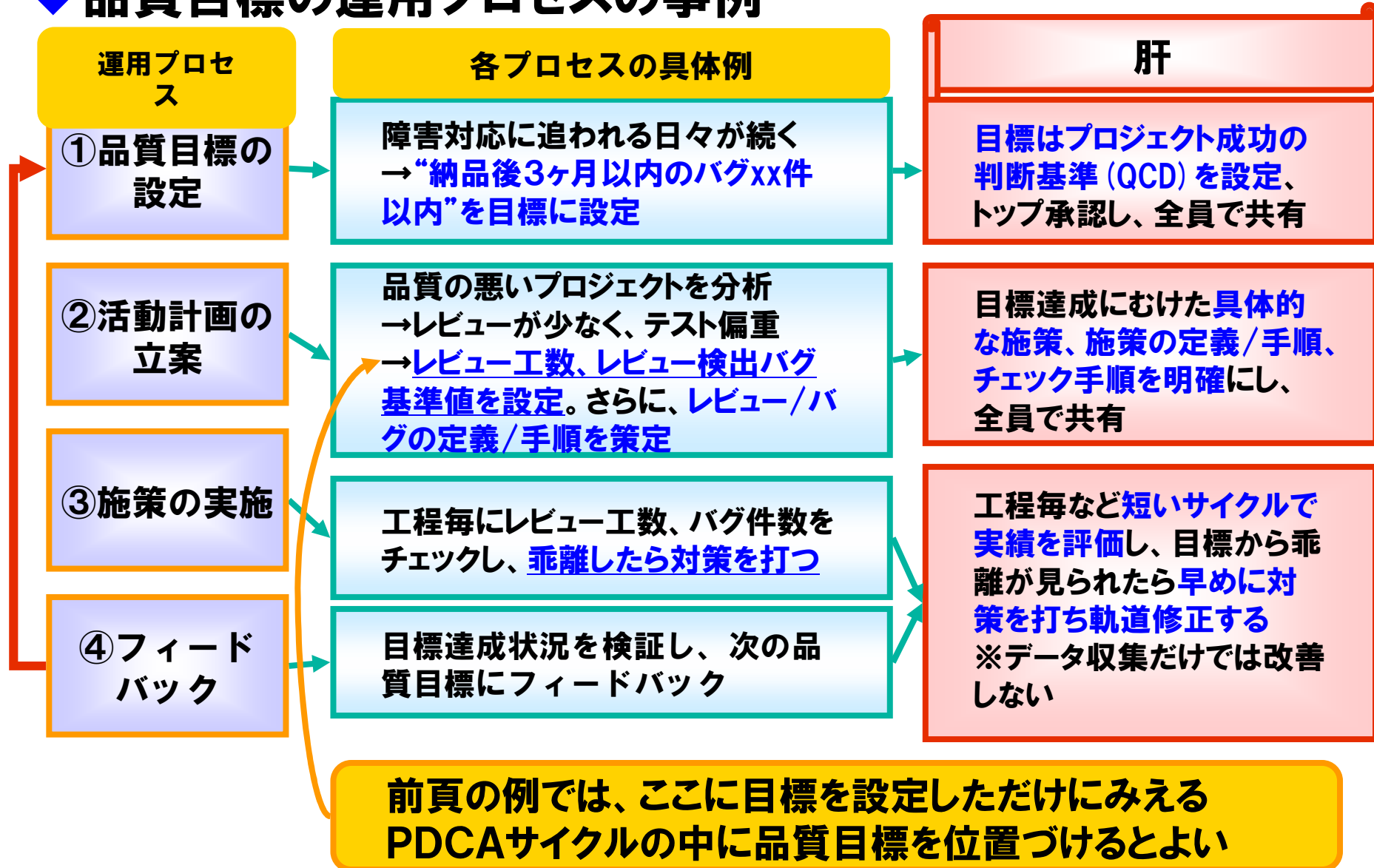
→ 品質目標の運用プロセスに、問題がない
だろうか？

→ 次頁に事例を紹介



具体例① 品質目標に関する肝(2)

◆ 品質目標の運用プロセスの事例



具体例② メトリクスに関する肝(1)

◆ 悩み

- メトリクスとして何を測ればよいか？
- 収集したデータをどう評価すればよいか？



◆ 肝

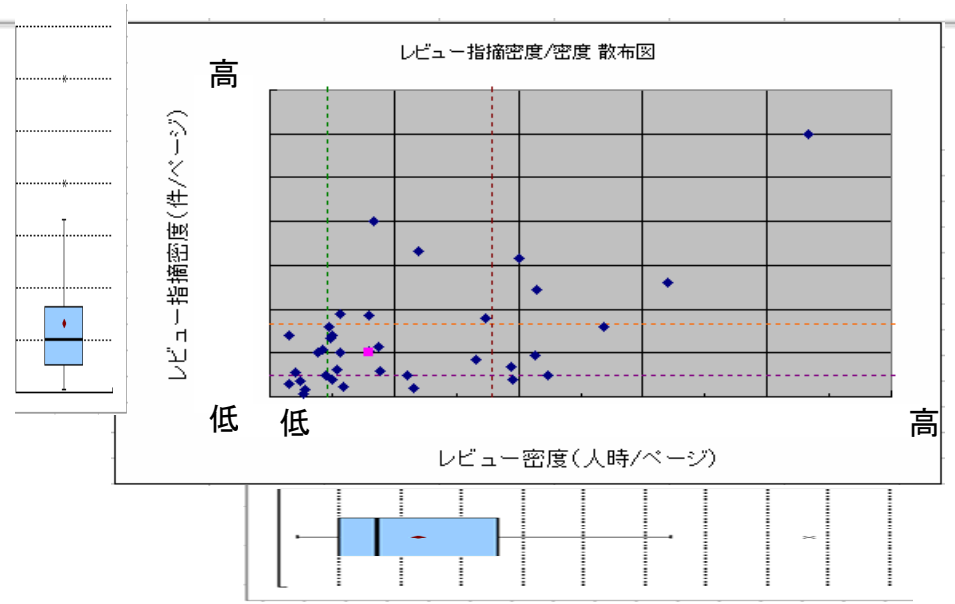
- 「データを平均値だけで見ない」
→ 個別データに“はぐれ値・外れ値”などの異常を探す
- 「データ上、何の問題もなし」
→ 逆に怪しい。成果物を見て実態を探る
- 基準値との比較だけではわからない場合が多い
→ 複数のデータを組み合わせた分析が必要
※ 次頁に事例を紹介

具体例② メトリクスに関する肝(2)

◆ レビュー量が多いこと、レビューの指摘数が多いことを、ばらばらに評価しても品質を判断できない

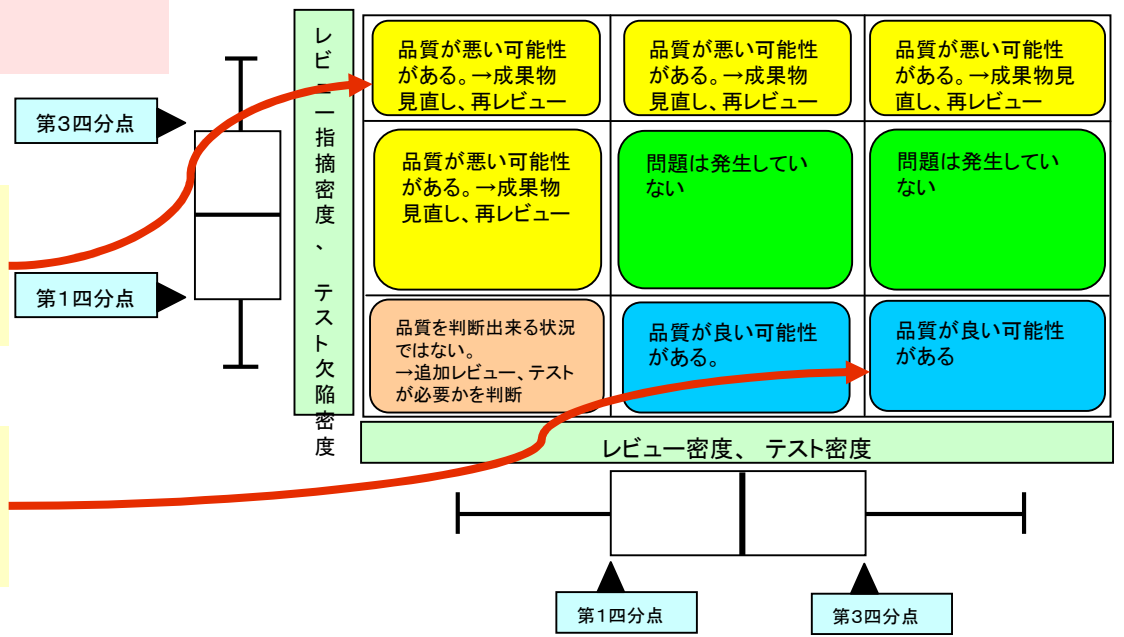


◆ レビュー量と、指摘数との関係を組み合わせて品質を判断



レビュー密度:小 & 指摘密度:大
→品質が悪い可能性あり

レビュー密度:大 & 指摘密度:小
→品質が良いといえる



具体例② メトリクスに関する肝(3)

- ◆ バグの作り込み工程と発見工程のマトリクスから、当該工程でどれほどバグを抽出したか、次工程以降にどれほど流出しているかを把握し、どの工程の品質が悪いかを判断する

		作りこみ工程					見逃し率(%)	妥当性(%)	飛び値有り
		要件定義	仕様作成	システム設計	プログラム設計	コーディング			
発見工程	要件定義	0					-	-	-
	仕様作成	14	14				26.3	50.0	-
	システム設計	0	5	24			0.0	82.8	-
	プログラム設計						-	-	-
	コーディング						-	-	-

メトリクス	内容	評価参考値
レビュー妥当性	当該工程のレビューで発見すべき指摘を当該工程で抽出した率	80%以上
レビュー見逃し率	当該工程のレビューで発見すべき指摘が次工程に流出した率	20%以下
試験妥当性	当該工程の試験で発見すべき指摘を当該工程で抽出した率	80%以上
試験見逃し率	当該工程の試験で発見すべき指摘が次工程に流出した率	20%以下
指摘飛び値有無	該当する工程よりも、2つ以上先で発見された欠陥の有無	無いこと

具体例③ テストにおける品質判断に関する肝(1)

悩み

- ◆ **テストを終了してよいかの判断が出来ない**
 - テストで摘出すべきバグを見積もれないので、いつテストを終了してよいか分からない
 - テストが十分か過剰か分からない
 - 不安なので、期限ぎりぎりまでテストをする
 - 期限になったら、「やるだけのことはやった」とテストを終了する

肝

- ◆ **どれだけテストをするのかは、V字プロセスにより各設計書を入力としたテスト項目を作成し、全てのテスト項目を消化することを最低限とする**
- ◆ **さらに、バグの収束状況から、テストが十分か、追加施策が必要かを判断する**
- ◆ **ただし、品質の判断は単純ではなく、多次元の組み合わせ分析で判断する**

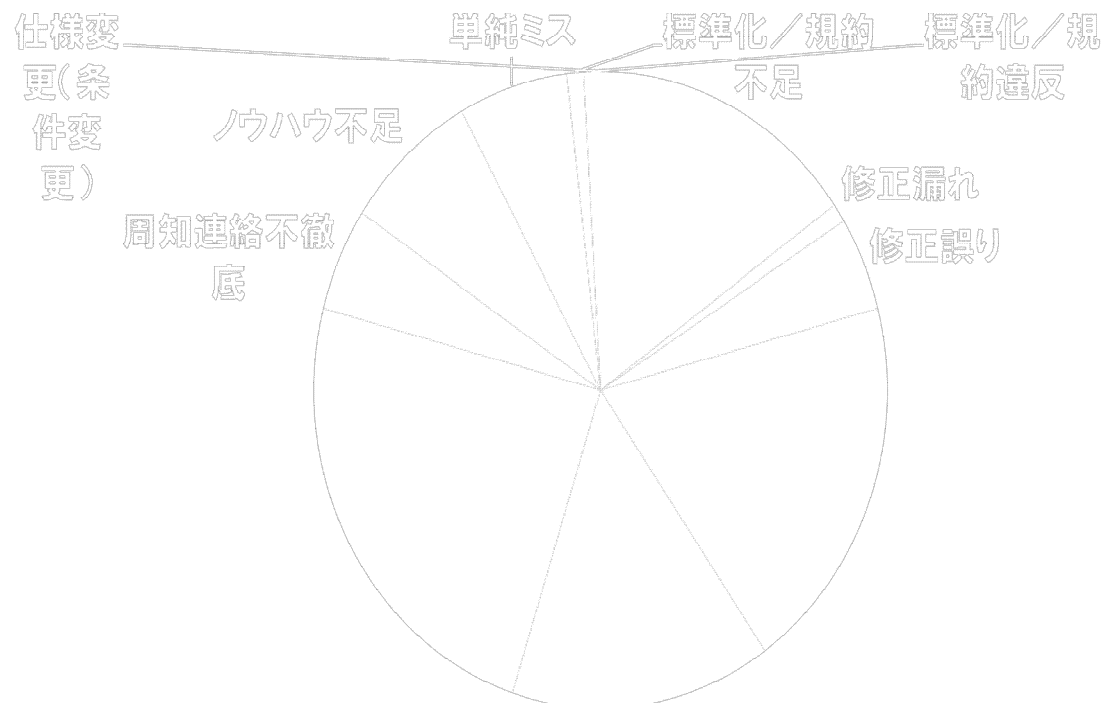
[→次頁に事例を紹介](#)

具体例③ テストにおける品質判断に関する肝(2)

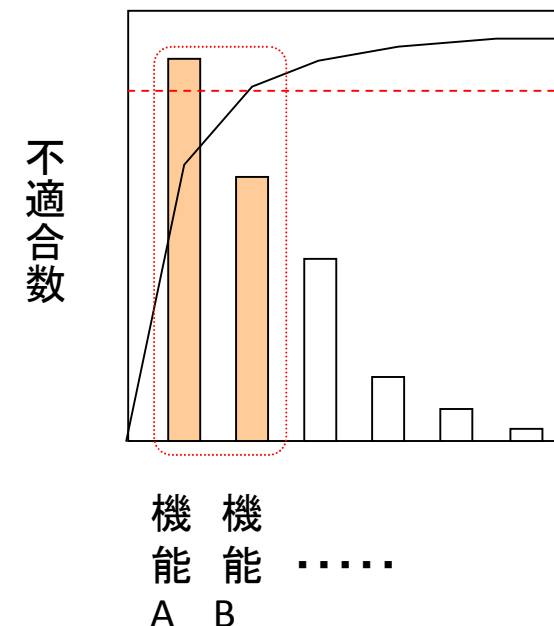
◆バグの傾向分析により、品質の弱点箇所を判断する

●バグ原因別分布、機能別分布、人別分布などに詳細化

- ・ 特定の箇所に集中している場合、該当箇所を再点検
- ・ 特定箇所がなく、全般的に同様の傾向の場合、プロセス全般を再点検



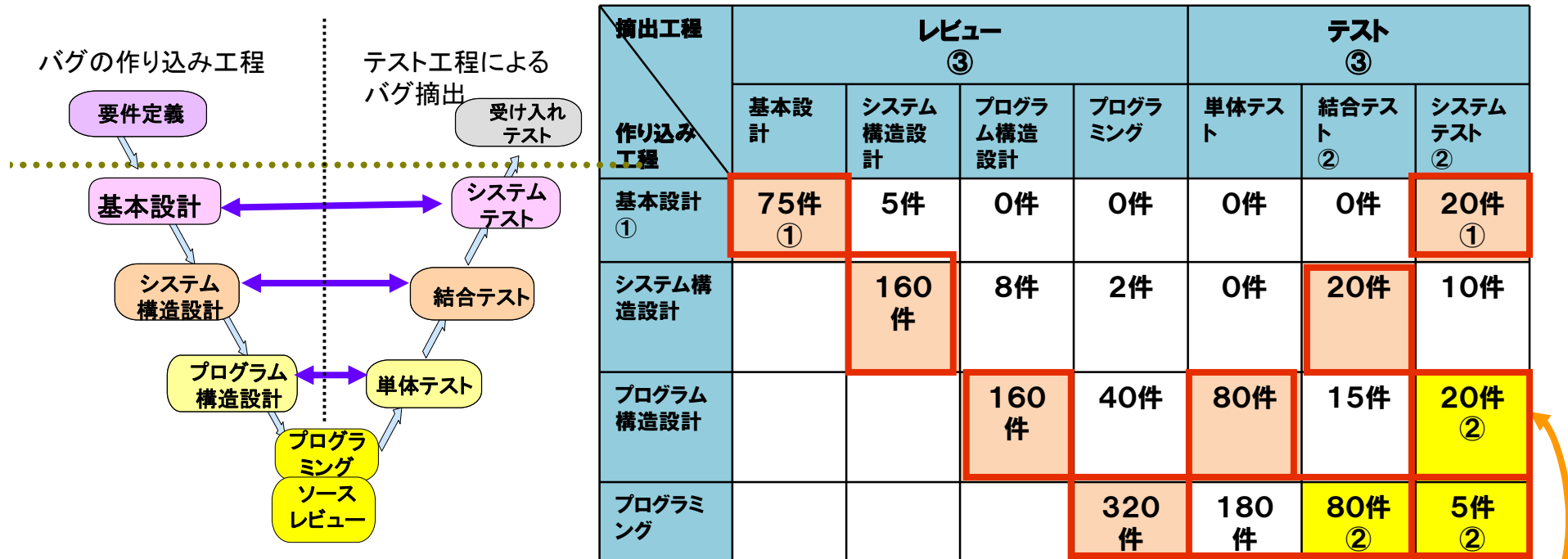
バグ原因の分布による傾向分析



機能別分布による傾向分析

具体例③ テストにおける品質判断に関する肝(3)

- ◆ テストフェーズと、バグの作り込み工程の相関表で評価する
 - 作り込み工程に対応したテスト工程で摘出されているか？ → ①
 - 該当テスト工程にふさわしくない、作り込み工程のバグはないか？ → ②
 - レビューの方が、テストよりも多く摘出しているか？ → ③



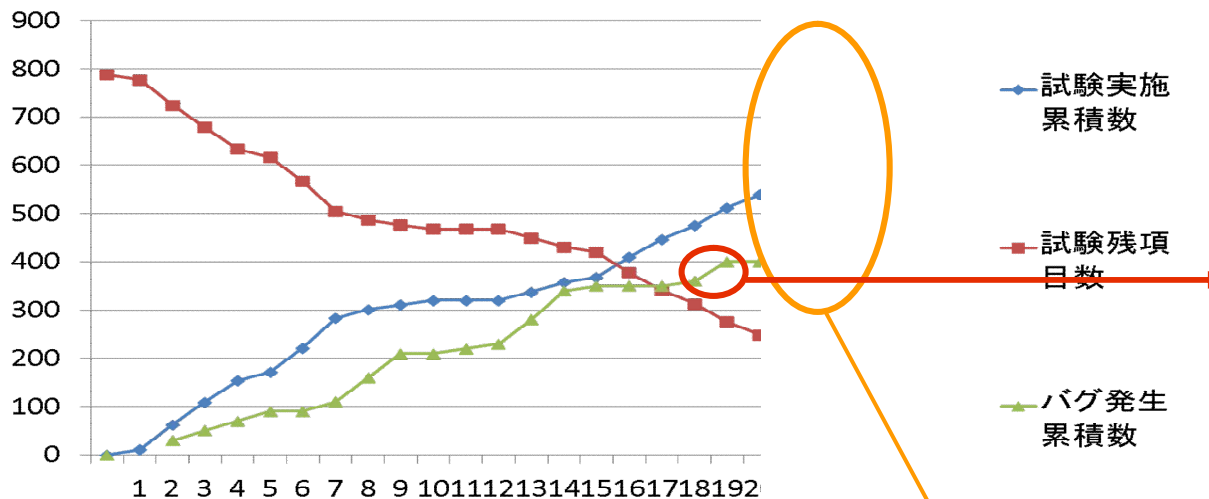
②の例：「システムテスト」には、作り込み工程が「プログラム構造設計」や「プログラミング」のバグがでている。当該工程のレビューや単体テストの不足が考えられる

具体例③ テストにおける品質判断に関する肝(4)

◆ **テスト項目の消化と、バグの発生件数の収束状況を評価する**
ただし、グラフだけで判断しない

⇒ **テスト終盤で発生しているバグの内容を確認する**
なぜテスト終盤で出たのか？、なぜこれまでのレビューやテストで見つけれなかったのか？を分析して、弱点箇所を探し、対策を打つ

テスト項目消化数とバグ累積数のグラフ



経験的には、予定テストを消化し、バグの対処が全て完了しても、さらにテストを追加し、もうバグが抽出されないことを確認する

■ **開発終盤で発生したバグの内容を確認する**

- ① **メインルート**のバグがシステムテストで検出されていないか？
⇒ **重大なテスト漏れの懸念**
- ② 同様な勘違いを複数のプログラムでしていないか？
⇒ **仕様書に欠陥があるかもしれない**
- ③ **バグの内容と、原因の分類**に矛盾がないか？
⇒ **バグ票の信憑性に問題があるかも**

※ **数字には表れてこない問題が見えてくる**

「ソフトウェア品質保証の肝」本編から、その他の肝を紹介



“品質保証の肝” 一覧

1.品質目標	①品質目標が立てられない	②品質目標はお仕着せではうまくいかない	③全社目標とプロジェクトの目標との関連性	④品質目標はリアルタイム評価すべき			
2.開発計画	①「計画書をタイムリーに更新しない	②他のプロジェクトの開発計画のコピペは怪しい		③非機能要件が明確に合意されているか			
3.見積もり	①見積もりの評価はベテランやデルファイ法を使う		②過去のプロジェクトの失敗が見積もりに反映されているか				
4.設計・コーディング	①お客様の声を機能へフィードバックする	②基本設計書にお客様の承認印があるか	③障害時の対策や運用を考慮しているか	④“前機種と同じ設計”は要注意。	⑤ツールに使われるな	⑥設計書はどこまで書くべきか	⑦コード変更の影響範囲を見積もれない
5.レビュー	①機能面以外の設計レビューのポイント	②レビューで問題指摘が無い	③レビュー状況の評価は早めに行	④レビュー会議に関係ない人が	⑤レビュー会議で、初めてレ	⑥レビューの場で、検	⑦レビューをどう評価していいか
6.テスト	①出荷検査で品質の実態を現場に示す	②テストの見積りができない	③テストを終了していいかどうか判断が出来ない	④テストフェーズが区切れない			
7.進捗報告	①進捗報告のウソを見抜くコツ	②まずい進捗報告の例	③プロセスの移行判定で完了しているべき事項が未決なのに先へ進んでいるのは許してはいけない				
8.品質管理	①計測できないものは品質向上できない	②納得されるデータ分析手法とは	③品質指標として何を見るか	④基準値は閾値。アクション無ければ意味無し	⑤バグ分析で品質を見極める		
9.人材	①ソフトウェアを開発するための資格とそれによる価値		②品質要員・PMO/QMOのスキルセット		③品質改善推進者を必要と認めつつ、その人財を確保することに消極的		
10.全般	①品質保証部門の責任ミッションとは	②組織の品質目標の達成が品質保証活動の目的	③ISO9001が形骸化	④品質システムのPDCAが上手く回らない	⑤開発部門と品質保証部門は犬猿の仲	⑥開発のアウトソーシングでは何を管理	

【肝】見積もりの評価はベテランやデルファイ法を使う

【肝の説明】

◆ 見積もりが適切かを判断するのは非常に難しい

→

- 見積もりの経験豊富なベテランに審査してもらう

- 複数人で見積もられた合意(デルファイ法)を使う

◆ 見積りは、金額を見積もる前に工数を見積もる。その工数に単価をかけて金額とする

◆ 見積りは、工期も見積りが必要である。工数と工期には関係がある。(COCOMO II、JUASによる工期モデルを参考にする。ただし、プロジェクトが大きい場合には合わない)

◆ 新規開発と改造では、見積もりの基準値は異なる

【肝】過去のプロジェクトの失敗が見積もりに反映されているか

【肝の説明】

- ◆ QCDで失敗したプロジェクトの原因を分析して見積もり時に反映するとよい
 - システムの特性や、お客様の体質
 - プロマネの資質や、メンバーのスキルこれらからリスクを想定して、見積もりに反映する
- ◆ 計画策定時の見積もり値と、開発終了時の実績との差を分析して、見積りの精度を向上させる
- ◆ 工程終了毎に、見積もりと実績の差を確認しておく
くと、見積もりの弱点が見えてくる

【肝】“前機種と同じ”設計は要注意

【肝の説明】

◆「前機種と同じコードを移植／流用する」場合は危険



◆前機種との些細な違いが、重大な影響を与えることがある

- 前機種と環境や運用が異なる部分があるはず。その部分を徹底的にレビューやテストで確認すべき。

◆コードクローン検出ツールなど、どれだけ流用部分があるかをチェックするだけでも、影響範囲を特定することが可能

【肝の説明】

◆ソースコード検証ツールが莫大な量の警告を検出して
も、担当者も管理者もやる気が失せてしまう



◆ツールの警告の内容を理解し、どの警告に対応すべきかを、あらかじめ合意しておく

◆ツールの実行時期は、対象となる設計書やコードが
完成した時点では遅く、要所要所で適用すべき

【肝】機能面以外の設計レビューのポイント

【肝の説明】

- ◆ **設計者本人に説明させる。** あいまいな箇所や自信がなければ、しゃべらせるだけで問題箇所がわかる。本人が自分自身で気づく場合もある
- ◆ **設計の入力情報も確認する。**（旧版や、間違った情報の参照を防ぐ）
- ◆ **第三者が理解できない文章は最初から不適合とする**
- ◆ **書かれていないことを聞く**（例：性能，容量，，）
- ◆ **必要なことが書かれていないことが判明したら、レビューを進めてはいけない。** 追記することを指摘し、レビューで書かれていない内容を深追いしない

【悩み】進捗報告のウソを見抜くコツ

【悩みの説明】

- ◆ 担当者は進捗を%で報告するが、数値の根拠がなく、適当な値と判明

【解決のヒント】

①進捗を示す%の算出方法を定義する

- 設計工程なら仕様書の作成ページ数/ 予定全体ページ数
- コーディング工程なら作成規模/ 予定全体規模
- テスト工程なら実施テスト項目数/ 予定全体テスト項目数
- 1週間以下に作業を細分化したWBSの終わった数/ 全体数

※消費した工数で進捗を評価せず、どれだけの成果物(アウトプット)を生産したかで進捗を評価するべき。EVMは有効。ただし、過信は禁物。

②%の報告だけでなく、中間成果物を必ず出させ、内容をチェックする

- 全ての内容をチェックすることは無理だが、サンプリングしてでも見る
- 内容の正しさを見るわけではないので、チェック者は必ずしも技術スキルが高い必要はない

③過去に同様の問題があったチームや人の報告には、集中的に内容をチェックする

【肝】 基準値は閾値。アクション無ければ意味無し**【肝の説明】**

- ◆メトリクスは、データを収集するだけではなく、**実績値が基準値を超えた場合に、どのようなアクションを取るかまで、ルールを決めておくこと。**基準値は閾値と考えて、閾値を超えた場合に何かのアクション(分析と対策の実施と報告)をとらなければ意味がない
- ◆基準や閾値という言葉が、**現場の抵抗がある場合には、「参考値」と置き換えると良い。**参考なので、強制力は無いが、現場が自らアクションを取る方向に誘導する。(品質保証部の腕の見せ所)自ら決断したアクションならば、現場も達成しようと頑張る。

【悩み】品質改善推進者は必要と認めながらも、その人財を確保することに消極的である

【悩みの説明】

- ◆ トップから「品質改善の成果を見せろ」と迫られるが、何を成果として見せていいのか分からない
- ◆ 改善施策が、どれだけ品質に影響を与えたかを定量的に示すことが難しいので、説明に自信を持ってない
- ◆ 従って、積極的に人財を確保するためのアピールができない

【解決のヒント】

- ◆ **現場に理解者を作る(時間がかかる)**
 - 10年もすれば、現場の理解者が、裁量権をもった人財に育つ。
 - その間、絶えず改善努力を続ける必要がある。組織の20%の人の理解を得られれば、改善が停止するリスクは低くなる。
- ◆ **組織変更等で、改善が振り出しに戻らないように、組織横断的に理解者を得ておくことが必要**
- ◆ **改善成果が経営数値に結びついているように見せることが必要。経営層は経営数値が変化しないと納得しない**

【悩み】品質保証部門の責任ミッションとは何か？

【悩みの説明】

- ◆ ISO9001、トラブル対応、品質データ分析・・・トップから矢継ぎ早に業務を命じられる。何でも屋なのか？どこを向いて仕事をすればよいのか、わからない

【解決のヒント】

- ◆ 品質保証部門の業務は多岐に渡り、組織の中の位置づけは各々のケースで決める必要があり、様々な意見がある
- ◆ 少なくともトップ含めて組織的な合意が必須
- ◆ 基本的には組織の品質目標を策定し目標達成をミッションとすべき
- ◆ 業務の方向性は以下の2種類がある(障害対応系は別として)
 - ①今の品質を磨き、出荷品質を確保する立場で品質管理や出荷判定をになう
 - ②品質の良い開発をするためのプロセス改善を主導し、“開発のやり方”を改善し、“強い開発集団”を作ること
- ◆ ミッションの一つは「製品が品質要求事項を満たして必要十分な情報を提供できる状態であることを、事実として証明できる体系的な活動を継続すること」

【悩み】開発のアウトソーシングでは何を管理すべきか**【悩みの説明】**

- ◆ 国内／国外のアウトソーシングが増えているが、どんな管理をすべきか分からない

【解決のヒント】

- ◆ 社内の管理方法を参考にし、契約時点で、明確に決める
- ◆ 品質を判定する指標を、発注前に双方で合意しておく
- ◆ 発注前に発注先のスキルを判定できる情報を得ておくことも重要
- ◆ 納品時だけでなく、要所要所で確認ポイントを設けておく
 - 危険なポイントを探るには、発注先とのQ & Aを通して、仕様理解が怪しそうなところ、何も質問がなかったところ、進捗が遅れているところ、特に、発注先リーダがメンバの実態を把握できていなさそうなところに注目する
 - レビューなどで中間成果物の出来や、工程遅れに対する対策の確認を行う

まとめ

- ◆ 結局、“ソフトウェア品質保証の肝”とは何か？
- ◆ 品質保証活動における悩みがなぜ生じるのか？

- 品質保証の仕組みに頼りすぎていないか？
- 本来の目的を明確にしないまま、管理などの“手段”を押し付けたり、目的化してしまっていないか？

⇒ 本来の目的を実現するためには、どのような仕組みを使って、どのように運用するかが、悩みの生じるところであり、各々の組織に適切な解を見つけることが肝となる

- ◆ 今後も、品質保証の経験を蓄え、「ソフトウェア品質保証の肝」を、さらに充実したいと思います

【編集後記】 検討メンバーの感想①

- ◆ 各社、色々な悩みがあり、また、各々の解を持っていることがわかりました
- ◆ 各社で微妙に手法は異なりますが、**根底の考え方は共通であり、それが肝ということが判ってきました**
- ◆ 今後は、不足している箇所の事例収集や、解が不十分な部分について、継続して議論して充実させていきたい



【編集後記】 検討メンバーの感想②

- ◆ いかにして、品証部門の価値を経営層に訴えるかという点で、管理職になって以来、悩んでいます
- ◆ 部員に言っていることは、“開発部門から信頼され、頼りにされるように仕事しよう。後出しジャンケンと言われないように、先ずはこちらがやることの、目的や手段の説明をしよう。我々が問題視するのは、開発している人でなく、アウトプットやプロセスである。両者の真ん中にアウトプットを置いて、それに対して議論しよう”。
- ◆ 「肝」としてまとめた資料を部内展開しました。自分達の抱える悩みを見直す良い機会となり、また、『肝』を書いてみることで自分達の頭の中を再整理でき、また、部員にとっても新鮮な刺激であったようです。いろいろと気付かされることも多く、どのようにブラッシュアップするかは今後の課題と思いますが今後も継続してブラッシュアップしていく必要性を痛感しました。

【編集後記】 検討メンバーの感想③

- ◆ 何処の品質保証部門も困っていること、考えていることはそんなに変わらないと感じました
- ◆ ハード品質保証部門から、「ソフトは何故、同じような問題を繰り返すのか」と問われます。ハードは、確実に品質を積み上げていますが、ソフトは、体制や人の入れ換え、プロセスの変更等で、突然、リセットされてしまう場合があります。それをどう見極めるかが、品質保証部門の見識と考えます。
- ◆ 一方で品質保証部門の立ち位置が、経営層の考え方により変化しています。「親身な第三者で」と言っていたのが、「もっと厳しく客観的に評価」へと大きく変わっています。品質保証部門も、良い製品を提供したい思いから、開発に踏み込み過ぎることもあります。結果、思いと職責が合わず、ジレンマを感じることも多いですが、宿命と考えています。

【編集後記】 検討メンバーの感想④

- ◆「組み込み」「エンタープライズ」等の分類に関係なく、品質保証に関する「悩み」や「肝」は各社各人共通のテーマであることがわかりました。このことは、部長の会という活動に重要な意義があることを示唆していると思います。
- ◆肝自体は、基本的に現場の活動に依存するわけですが、品質保証部門がいかに現場をサポートできるか、という品質保証部門のパワーやモチベーションが根本的な肝であることがわかりました。
- ◆「良い人材が配属を希望するような品質保証部門に如何にしていくか」に品質保証部門の真価であることを実感しました。

【編集後記】 検討メンバの感想⑤

- ◆品質保証活動に、“肝”は確かに存在すると考えていた
- ◆メンバと意見交換することで、浅くとも広範囲の“肝”を集めることができ、一定の成果があった
- ◆多くの成功と失敗の経験を、自分自身で一旦抽象化し、“肝”として身に付け、それを新たなプロジェクトに適用し、プロジェクトを成功に導ける人、高品質なプロジェクトに仕上げるることができる人が、“達人”とか“アーキテクト”と呼ばれるのだろう
- ◆最も確かな教育方法は、この“達人”からOJTを受けることだが、“達人”の持つ“肝”を、ドキュメントなどで形式知化するのも効果的だろう
- ◆次期は、この辺りを進めると、良いアウトプットになるかもしれない

【最後に】 一緒に「肝」を考えませんか？

◆ 来期も、“品質保証の肝”の活動を継続する予定です

◆ ぜひとも、

- この機会に部長の会に参加いただき
- 一緒に肝を議論しましょう
- そして、お互いに気付きを得たり、自部門内への展開などのフィードバックを通じて
- 日本全国のソフトウェア品質保証活動を盛り上げていきましょう！



ご清聴ありがとうございました

End