

「ベンチマーキングによる生産性・ 品質の向上」

大日本スクリーン製造 株式会社

技開センター ソフト開発グループ

粕渕 清孝

- 経緯
- ソフトウェア開発データ白書
- 事例紹介
- まとめ

大日本スクリーン製造株式会社

京都府京都市 1943年設立 東証一部

売上 1,641億円 (連結2010年3月期)

従業員数 4,679名 (連結2010年3月期)

事業

半導体製造装置 (洗浄装置世界シェアNo. 1)

FPD製造装置

プリント基板製造装置

印刷製版機器・IT関連 (CTP世界シェアNo. 1)

ソフト開発グループ (CMMI Level 3)

社内ソフトウェアの開発



半導体製造装置

愛あ永の

ヒラギノフォント



LeafThrough

今まで

- ・ 日欧米間での競争
- ・ 品質は外部失敗コストとして表出し競争力に影響
- ・ 下流工程で品質確保の傾向

GAP
解消上流工程からの戦略的な
品質確保と生産性向上が必要ソフトウェア工学的
アプローチ
ベンチマーキング

今から

- ・ コスト競争力の高いアジアからの新規参入
- ・ コストは内部失敗コストとして内在し競争力に影響
- ・ 上流工程からの品質確保により失敗コストを削減

とは言っても、
同業他社や競
合のデータは
ない。
そこで...

経済産業省IPA発行

2009年版PDF公開中

<http://sec.ipa.go.jp/publish/index.html>

比較データプロファイル(2009年版PDF)

22社、2327プロジェクト

- ・ 金融・保険が多い、次に製造、情報通信業と続く

新規開発 : 56.3 %, **改良開発** : 38.7 %

規模(中央値) : 51.4 KSLOC, 417 FP

工期(中央値) : 6.1ヶ月, **工数**(中央値) : 46.0人月

- ・ 基本設計～総合テスト(ベンダ確認)の5工程

工程は SLCP-JCF2007とも対応考慮

製品 A

- 制御ソフトウェア

- バージョン5から8の総合テスト

 - 用意したテストを全て実施

- バージョン9の総合テスト

 - 納期が厳しく、テストを全て実施できない

 - テスト戦略による優先順にテストを実施

 - 要求レビューへテスト設計者参加

 - テストレビュー実施

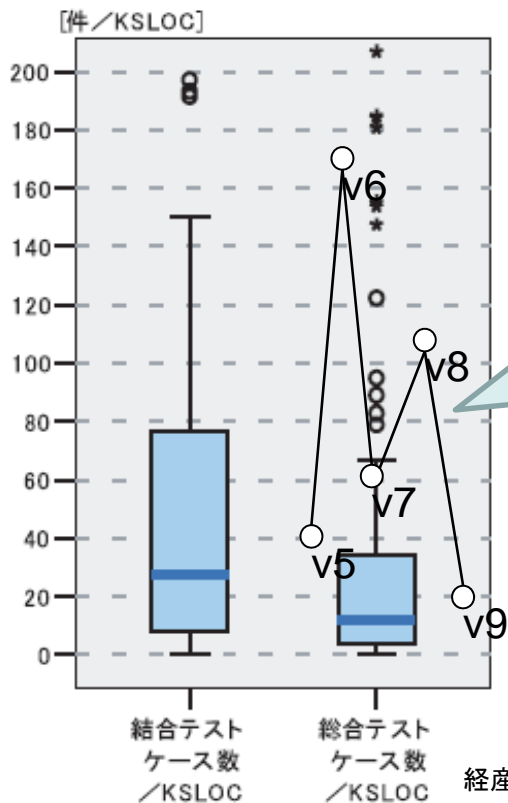
総合テストの効率について調査

- 少ないコストで品質を確保できたかどうかの確認

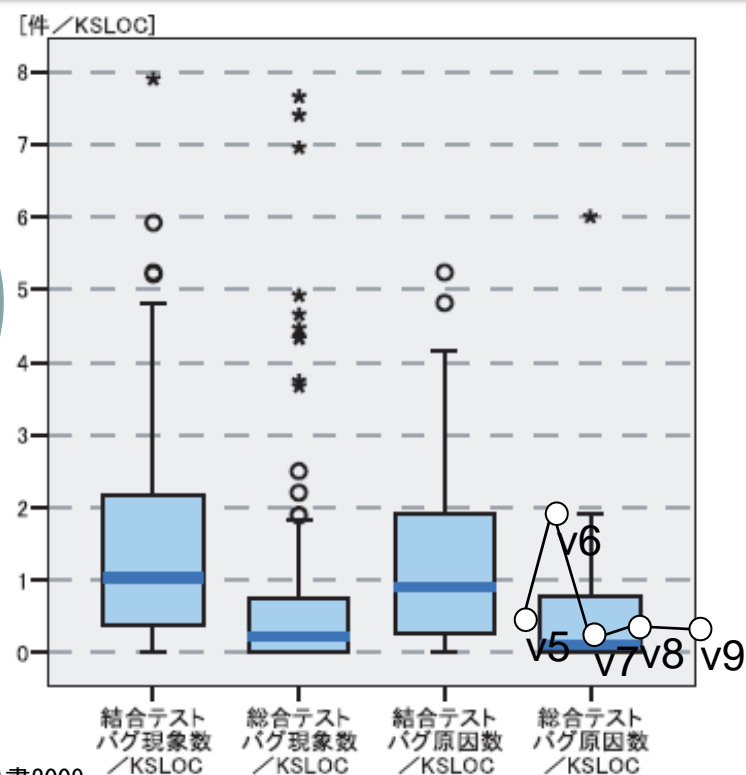
事例 1 - テスト分析

Ver.	SLOC	ケース数	検出数	検出%
V5	27896	1134	13	1.15
V6	1026	176	2	1.14
V7	15583	919	2	0.22
V8	25337	2731	7	0.26
V9	42183	831	12	1.44

- ・規模あたりのテストケース数がばらついている。小さな数値の場合網羅性が気になる。
- ・V6は規模あたりのテストケース数、バグ原因数が大きい。ただし有効変更行数が小さいので、一旦要注意に留めた。
- ・V9は規模あたりのテストケース数が小さいが、検出率が高い。結合テスト後の品質が低い、または総合テストの効率が低いことが考えられる。



経産省IPA ソフトウェア開発データ白書2009



追加調査

- 「V9結合テスト後の品質が以前と比べて低い」？
→ 「以前と変わらないはず」
- 「V9総合テストの効率が高いのでは」？
→ 「高いとは考えていない、設計変更の影響範囲考慮不足による不具合が減らない」
…どうということか？

考察

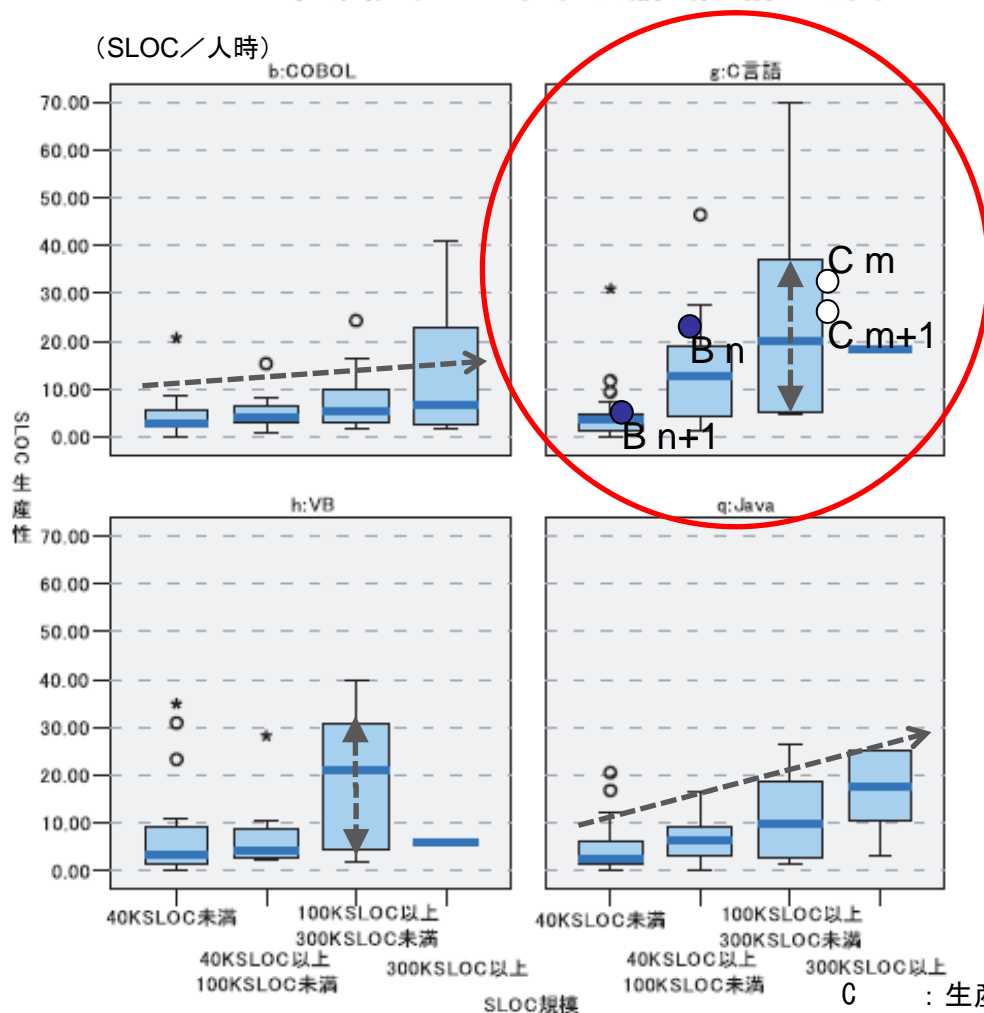
- V9総合テストの効率は高いが、リリース後にデグレが多く実感できない。
- 短期的には回帰テストで品質を確保していく。
- 中長期的には変更要求・設計の影響範囲を要求定義・設計工程にて見極めていく必要あり。

- ▶ 製品 B
 - ▶ データ処理ソフトウェア
 - ▶ バージョン n と $n + 1$
- ▶ 製品 C
 - ▶ データ処理 + 制御ソフトウェア
 - ▶ バージョン m と $m + 1$

- ▶ 生産性・テスト効率について調査

事例 2 - 生産性分析

図表 6-7-33 ● SLOC 規模別 SLOC 生産性
(改良開発、主開発言語別) 箱ひげ図



Ver.	KSLOC	SLOC/人時	偏差値(*)
製品B n	61.0	22.6	65
製品B n+1	25.6	6.5	53
製品C m	184.6	26.9	58
製品C m+1	209.9	31.6	61

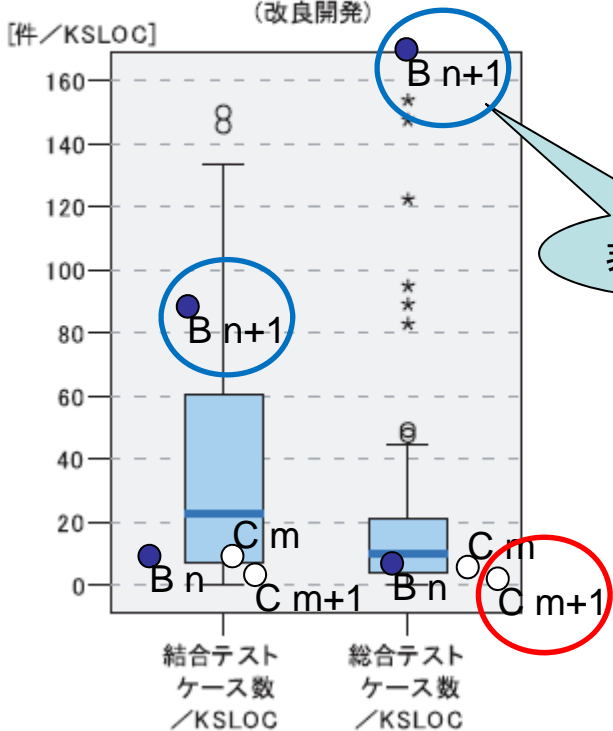
(*)みなしで算出。

- ・製品Bのバージョンnとn+1を比較すると生産性が1/3になっている。ただし**規模が違うため単純に比較はできない**。
- ・製品Cのバージョンmとm+1は規模が同じで生産性もほぼ同じである。

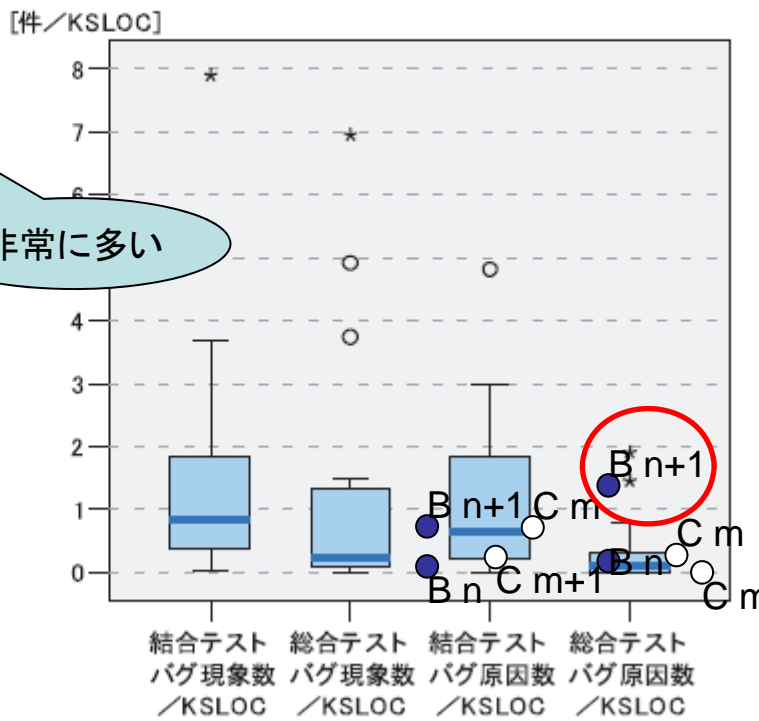
事例 2 - 生産性分析

Ver.	KSLOC	結合テスト			総合テスト		
		ケース	検出数	検出%	ケース	検出数	検出%
製品B n	61.0	563	4	0.71	508	7	1.38
製品B n+1	25.6	2246	19	0.85	4977	44	0.88
製品C m	184.6	1392	159	11.42	250	32	12.80
製品C m+1	209.9	665	49	7.37	263	13	4.94

SLOC規模あたりのテスト ケース数 (改良開発)
[件/KSLOC]



SLOC規模あたりの検出バグ数 (改良開発)



検出率 : 7% (データ白書)

- ・製品Bは検出率が低い。Ver. nに比べてn+1の結合・総合テストのケース数が多い。Ver. n+1のバグ原因数が多い。
- ・製品Cは検出率が高い。結合・総合テストのケース数が少ない。バグ原因数が少ない。

事例 2 - 生産性分析

Ver.	要求定義	基本設計	詳細設計	製作	結合テスト	総合テスト
製品B n			84.5			15.5
製品B n+1			75.2			24.8
製品C m			97.1			2.9
製品C m+1			99.8			0.2
データ白書	対象外	16.1	17.4	37.0	16.7	12.8

結合+総合テストは工数・工期とも全体の約1/3（データ白書）

- ・製品Bは総合テスト工程の工数割合が高い。
Ver. n+1の結合・総合テストでのケース数が多いが、検出率は低い。
Ver. n+1の総合テストでのバグ原因数が多く、**結合テストではケース数が多い割に妥当なテストが実施されなかった可能性**がある。
- ・製品Cは総合テスト工程の工数割合が低い。
結合・総合テストでのケース数が少ないが、検出率は高い。
結合・総合テストで妥当なテストが実施されなかった可能性**がある。**
単体テスト工程までに高い品質を確保している、もしくはリリース後に不具合が多く発生する可能性がある。

事例 2 - 生産性分析

➤ 実は…

➤ 製品B

- 生産性がばらついているのは、Ver. n+1で OS・ライブラリ変更に伴う回帰テスト増のため。
- 全体的に十分テストが行われているが、結合・総合テストに関して重複した内容が多い。
- テストレビュー等により開発チームとテストチーム間の交流を深めれば、棲み分けによりテスト効率や全体の生産性を高くできる。

➤ 製品C

- 生産性が高いように見えるが…
- Ver. mでは結合・総合テストケース数が少ない割に検出率が高いため、テスト検出バグ数は平均的である。単体テスト後の品質が低い可能性がある。またリリース後の品質も低い可能性がある。(心配である)
- Ver. m+1では単体テストを行い状況が改善しているが、また全体的にテストが不足している。

- ▶ 製品 D
 - ▶ 制御ソフトウェア
 - ▶ 品質が良くなってきている(と聞いている)

- ▶ 問題のある工程の見える化
 - ▶ 問題工程を特定し、改善に注力したい

事例3 - 欠陥除去モデル

欠陥除去率 = 開発中に検出した「欠陥」の総数 ÷ 開発~納品後に発生する「欠陥」の総数
ソフトウェア品質を知る上での重要なファクタ

欠陥除去工程	欠陥作り込み工程			合計	欠陥除去率
	要求定義	設計(※)	実装(※)		
要求レビュー	197			197	55%
設計レビュー	125			125	36%
テスト	31	114	63	208	69%
リリース後	4	71	17	92	100%
合計	357	185	80	622	85%

工程毎の欠陥除去率

全体の欠陥除去率

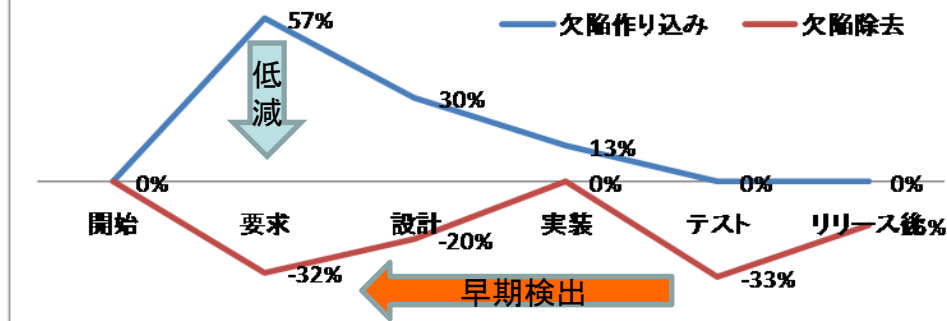
(※)

設計工程以降は、開発系協力企業に委託しているため、設計・実装工程での作り込み、除去は数値が取れなかった。

よって設計工程及び全体の欠陥除去率は上がる可能性がある。一方、今後の不具合発生状況によっては逆に悪くなる可能性もある。

ただしこれらは要求レビューでの欠陥除去率には影響しない。

欠陥除去状況



保田勝通, 奈良隆正, ソフトウェア品質保証入門, 日科技連出版社, 2008/05

要求工程での欠陥除去率は高くない。全体では普通程度。

C.Jonesによれば、CMMILレベル2の組織では、全体の欠陥除去率は85%
ミッションクリティカルなシステム(航空宇宙、軍事、インフラ)では95%

事例 3 - 欠陥除去モデル

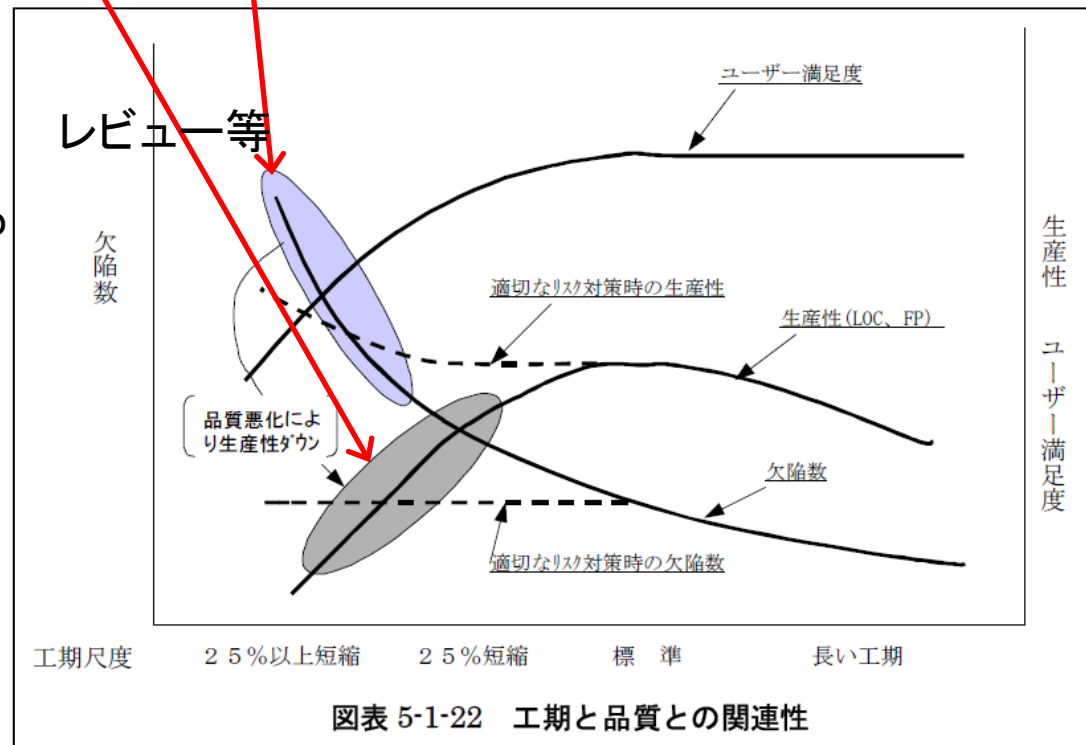
- 製品Dソフトウェア開発の特徴
 - 納期が厳しい + 仕様の安定性が低い + 協企依存度が高い
- 生産性・品質への影響
 - FP法や UCP法でいう環境要因係数(EF)が悪い値
 - ソフトウェア工学的には**生産性・品質**に悪影響が出るはず

現状

- 定期的な改善活動により
なんとか抑え込んでいる
- 問題発生を受けての改善
活動では近々限界発生

課題

- 要求定義工程の品質改善
- プロセス周知徹底による
全体最適の気づきと改善



■ 実は…

- コンポーネント毎に基本設計～結合テストまでを異なる開発系協力企業(グループ会社含む)に委託している。
- 総合テスト時にコンポーネントをまたがるような機能について度々問題が発生している。
- PJ内の原因分析では、開発系協力企業が影響範囲を見極められなかったと分析していたが…
- 要求定義工程からの影響範囲の見極めが本筋。社員が設計工程以降もレビューに参加するなどの工夫が必要。

データ白書ベンチマーク 必須データ

- ・ リリース後不具合検出数
- ・ 結合・総合テストの不具合検出数
- ・ 結合・総合テストのテストケース数
- ・ 規模 (FP or SLOC) (改良時変更規模)

← これは当然
Level 1
まずはここから
いかがでしょうか

データ白書ベンチマーク オプションデータ

- ・ 各工程レビューの指摘数
- ・ 各工程の工数 (品質コスト分類)
- ・ 欠陥作り込み工程と除去工程

↓ Level 2
より詳細な分析
ができます

ベンチマークのすすめ

- ・ 自分達の実力を知ることができる、ステークホルダーに説明できる
- ・ 気づきが改善の良いきっかけに (違う改善のきっかけに繋がることも)
- ・ 問題対処のプロダクト品質改善から未然防止のためのプロセス品質改善へシフト

SCREEN