

15-B-7

メトリクス分析手法を用いた試験品質向上の取組み¹

1. 概要

製品開発において QCD（品質/コスト/納期）の確保は当然であるが、工程が逼迫している受託開発においては、納期(D)が優先される場合が少なくない。これは、顧客と合意した期日に対し、納期遅延が顧客にとっても受託側企業にとっても大きな損失になることが理由であることは言うまでもない。

受託開発において、工程遅れの多くは上流設計の遅れであり、我々の所属部門では、特に開発期間が 1 年を超える新規開発において、上流工程に時間を費やしてしまう傾向がある。この結果、実装以降の計画が遅れ、納期に向け残された期間で試験を実施しなければならない。このような状況になると、総合試験の期間を確保するため、ソフトウェア間の段階的な結合試験が十分実施できないまま総合試験に進んでしまうことになる。最終的にはシステム試験工程で、本来結合試験等で解決されているべき不具合が顕在化する可能性が出てくる。

そこで、決められた納期に対し、逼迫している試験工程の中で品質を如何に確保するかについて検討した。所属部門のデータから試験に関するメトリクスと品質の関係について、「統計的手法」を用いた分析を行った結果、「結合試験の期間がシステム試験時の不具合発生に影響している」ということが実績データから分かった。その結果、設計者、管理者等関係者に対しソフトウェアの結合試験の位置づけを再認識してもらい、品質向上に繋がる適正な試験期間の指標を設定することができた。

本稿では、試験に関するメトリクス分析のアプローチ及びその有用性と試験品質の向上に向けた取組み事例を述べる。

2. 取組みの目的

2.1. システムの特徴と背景

所属する部門は社会インフラ系の受託開発を行っている。

2011 年度から 2013 年度上期における所属部門の開発プロジェクトの実績データを確認したところ、図 15-B-7-1 に示すとおり、「既存システム改良開発」に該当するプロジェクト数が約 70%を占めていた。これに対して「新規システム開発」は約 25%であり、プロジェクト

¹ 事例提供: 株式会社東芝 社会インフラシステム社 福田 伊津子 氏

数で見ると「既存システム改良開発」の方が多いことが分かる。

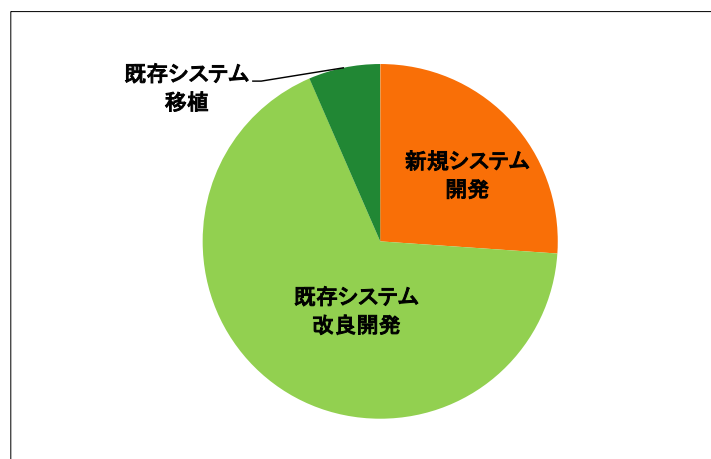


図 15-B-7-1 プロジェクト数の割合(2011～2013 年度上期実績)

また、図 15-B-7-2 に示すとおり、全プロジェクトのソフトウェア実績規模（ソースコード量）についても、総量は「既存システム改良開発」の方が多。これはプロジェクト数が多いことによるものであるが、「新規コードの作成」に着目すると「新規システム開発」の方が圧倒的にソースコードの量が多いことも分かる。

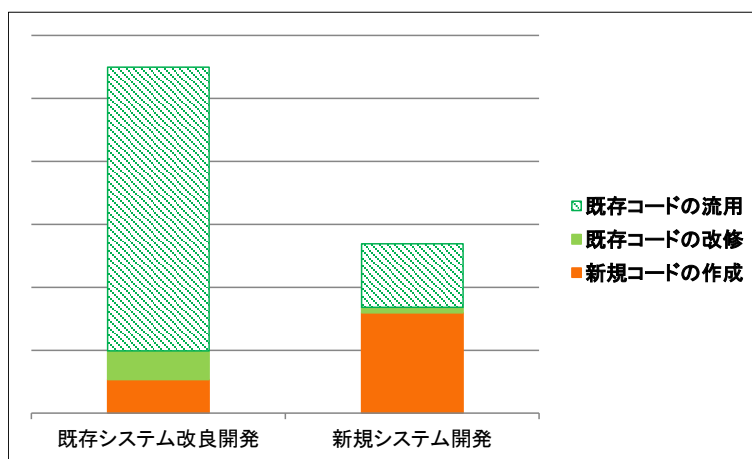


図 15-B-7-2 全プロジェクトのソースコード量 (2011～2013 年度上期実績)

さらに、開発中の不具合検出数を確認しても、図 15-B-7-3 に示すとおり、結合試験及び総合試験の不具合検出総数は「新規システム開発」の方が2倍近いことが確認できた。

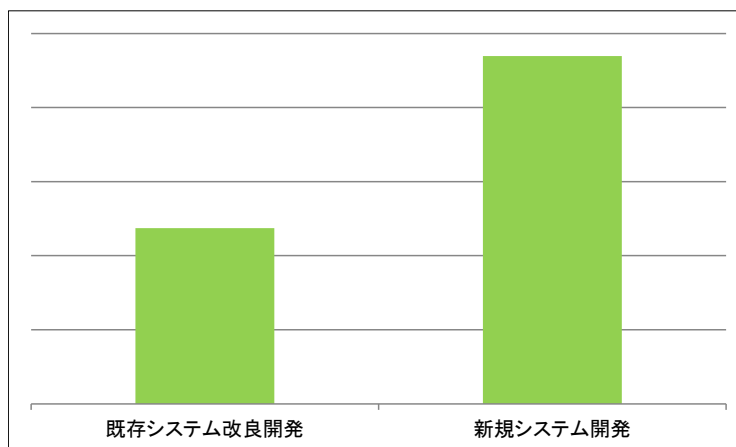


図 15-B-7-3 全プロジェクトの不具合総数（2011～2013 年度上期実績）

以上のことから「新規システム開発」の品質を向上させる必要性があると考えた。

2.2. 「新規システム開発」における不具合発生状況の分析

「新規システム開発」の品質について、現状の試験工程に対する不具合発生状況を一部のプロジェクトに対し調査し、分析を行った。なお、ソフトウェア開発の工程の定義は表 15-B-7-1 のとおりとする。総合試験後、製品であるハードウェアに組み込みシステム試験が実施される。

表 15-B-7-1 所属部門のソフトウェア開発工程

工程名称	外部設計	内部設計	コーディング／テスト		結合テスト	システム テスト	
所属部門 工程名称	ソフトウェア開発工程						システム 試験
	基本設計	詳細設計	コード作成 &デバッグ	単体試験	結合試験	総合試験	

(1) システム試験で発生した不具合の分析

プロジェクト A において、総合試験終了後、システム試験で発生した不具合に対する要因工程を分析した。ここでの要因工程とは混入工程ではなく、除去すべき工程を示している。分析の結果、図 15-B-7-4 のとおり、単体試験、結合試験及び総合試験で除去できたであろう不具合が 40%以上あった。

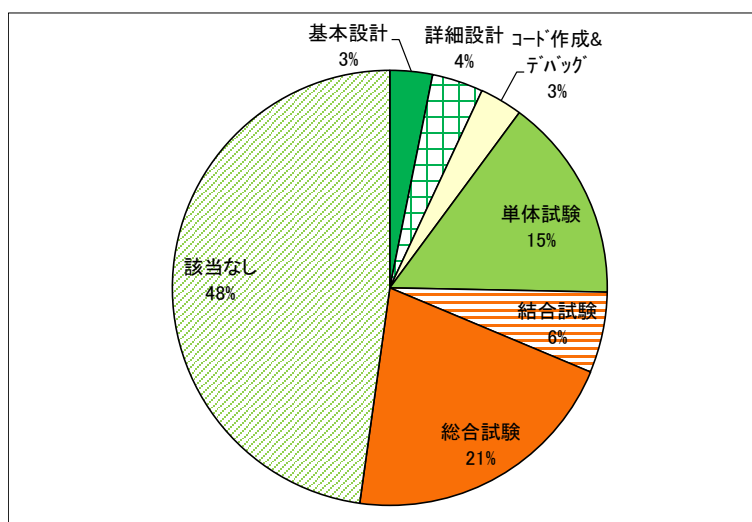


図 15-B-7-4 システム試験で発生した不具合に対する要因工程の割合

(2) 結合試験・総合試験で発生した不具合の分析

別のプロジェクト B に対し、結合試験及び総合試験で発生した不具合を機能別に分析した。図 15-B-7-5 に示すとおり、機能 3、機能 4 は結合試験での不具合検出件数は少ないが、総合試験では多く検出されている。このことから、結合試験で不具合が十分出し切れていない機能については総合試験で不具合が多く発生していると推定することができる。

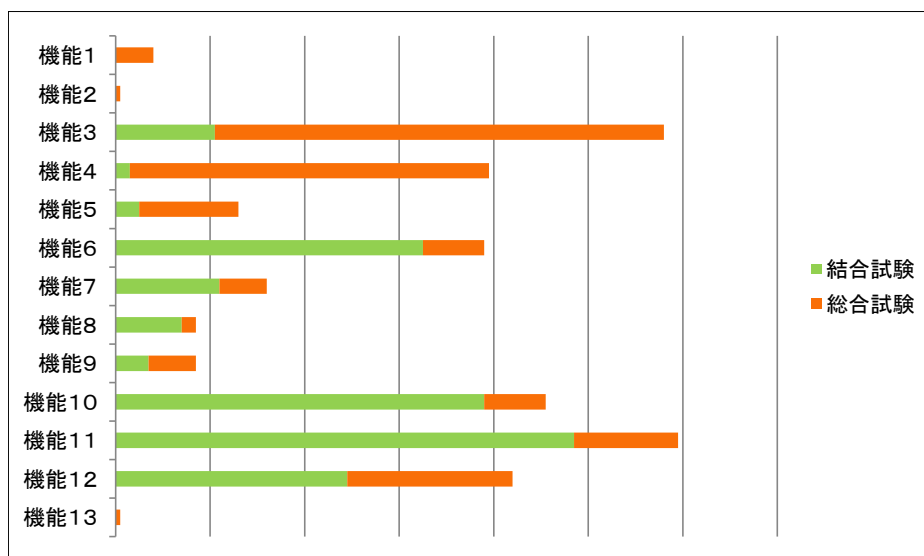


図 15-B-7-5 試験工程内で発生した不具合数（機能別）

(3) 上流工程で作りこんだ不具合の検出結果の分析

さらに、プロジェクト C において、不具合の混入工程に対するその除去状況を確認した。

図 15-B-7-6 で示す V 字モデルの各設計工程において、対応する試験工程で検証を行う

ことを前提として、各設計工程で混入した不具合は対応する試験工程で除去できればよいものとした。しかし、本プロジェクトではグラフ右の棒グラフに示すとおり、実際には十分な除去ができず、次工程に流出してしまっていた。

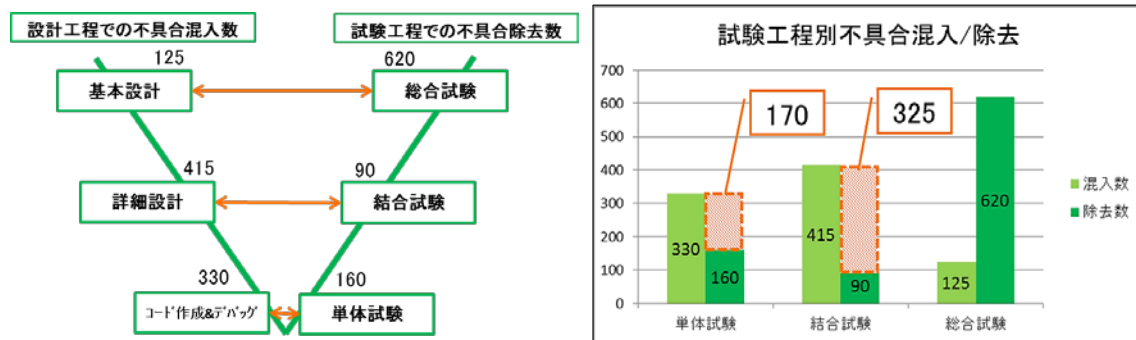


図 15-B-7-6 混入工程に対する不具合除去状況

2.3. ソフトウェア開発期間の実績把握

試験の実施状況の確認として、箱ひげ図を利用し全プロジェクトの開発工程期間の期間比率を確認した。図 15-B-7-7 のとおり試験期間で見ても、結合試験が全体工程に対し比率が低く、結合試験期間が少ないように思われた。ここで、詳細設計から単体試験については、主としてベンダに委託し、図 15-B-7-7 に示すデータは社内の一部の開発工程に対するデータであるため、期間の比較は行っていない。

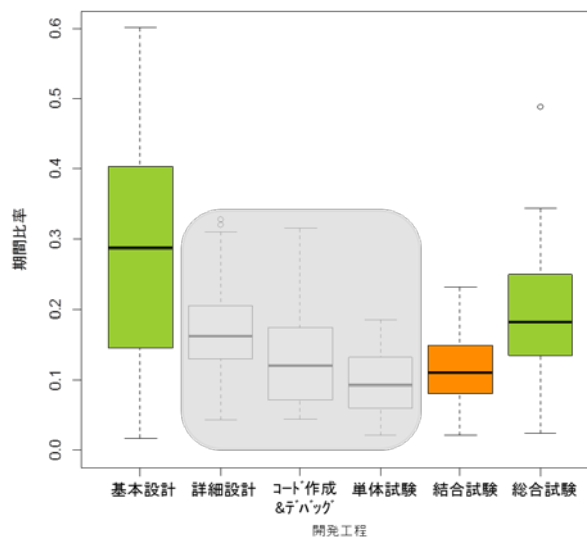


図 15-B-7-7 ソフトウェア開発期間比率（2011～2013 年度上期実績）

2.4. 現状の不具合分析からの課題

2.2.及び 2.3 から「結合試験を充実させる必要があるのではないか」という仮説を立てることができた。試験期間が少ないことにより、徐々に結合度を上げていくような段階的な結合試験が十分に実施されず、不具合が次工程に流出しているのではないかという仮説に対し、実績データを用いた定量的な分析アプローチを実施し以降で検証する。

3. 取組みの対象、適用技術・手法、評価・計測

本取組みは前述のとおり所属部門の試験品質向上を目的とし、試験工程に絞り検討を行った。試験品質の向上を対象とするため試験工程で取扱うメトリクスデータを分析する。この分析においては、2011 年度から 2013 年度上期までの所属部門の実績データを元に実施した。

適用技術・手法としてメトリクス分析手法を用いた統計的アプローチを実施し、定量的な評価を実施している。ここで示すメトリクスとは、ソフトウェアメトリクスを指し、ソフトウェア開発工程や作成する成果物の定量的データを計測し、管理することである。統計手法として、相関分析、検定（t 検定）を用いた。また、ツールとして統計解析ソフトウェア R を採用した。R はオープンソースで国際共同開発されており、機能的にも市販のソフトウェアに引けを取らない他、大勢の専門家によって吟味されており信頼性も高いことが知られている [1]。

さらに、取組みの達成度については、結合試験の実施期間が試験品質に影響しているか、統計的分析手法を用いて評価する。この際、試験品質のメトリクスとして不具合データを使用した。

4. 取組みの実施及び実施上の問題、対策・工夫

4.1. 課題解決に対するアプローチ

現状の不具合情報、ベンチマーク結果を元に、実績データの分析を実施、図 15-B-7-8 に示す流れで品質向上に向けた施策を検討した。

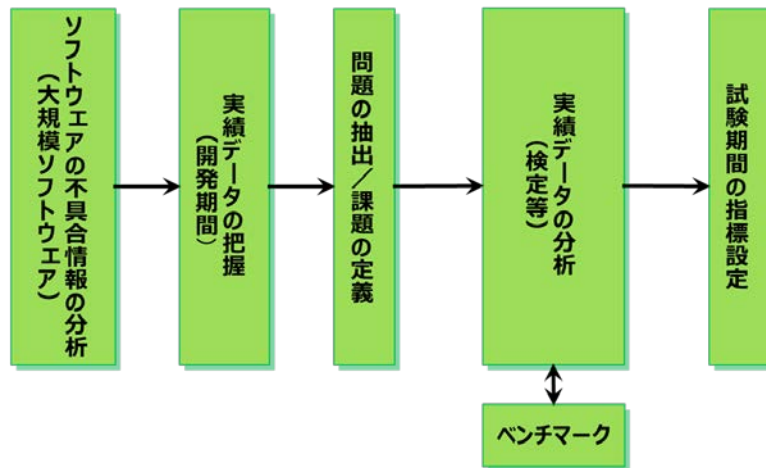


図 15-B-7-8 課題解決に対する実施の流れ

現状の不具合傾向から導出した仮説「結合試験を充実させる必要があるのではないか」について、試験品質に関わるメトリクスを用いて充実化の必要性の検討を行った。

「結合試験を充実させる」とは、結合試験の十分性を確保しているということであり、要求仕様に対する網羅性、妥当性、必要性等が十分確認されているということになる。所属部門では試験の十分性を確保するため要求仕様及び設計から試験項目を展開し、それらを項目ごとに設定しているが、2.取組みの目的 で述べた課題について、次の観点で検証していくこととした。

- ・ 試験項目の洗い出しの観点で問題（不足など）はないか
- ・ 試験実施時間が予定通り確保できていないのではないか

また、試験工程として結合試験及び総合試験を対象とし、部門で収集している次のメトリクスを用い分析することとした。

- ・ 不具合件数
- ・ 試験項目数
- ・ 試験日数（期間）
- ・ ソフトウェア規模（ステップ数）

4.2. 不具合数と相関する要因分析

(1) 工数と工期に関するベンチマーク

試験工程の品質を把握するために一般的に捉えるメトリクスとして、不具合件数が挙げられる。この他、不具合検出率も品質状況を把握するための有用なメトリクスである[2]。不具合検出率は試験工数に対する検出できた不具合件数であり、日ごとの監視をすることで、その傾きで品質の安定度合いが判断できる。

所属部門では試験工数が取得できていないため、「実績期間（日数）」で検討しても問題ないか、文献調査を行った。調査の結果、図 15-B-7-9 のとおり、「工期（月数）」は工数の

3乗根に概ね比例する傾向が見られる。」ことを確認することができた [3]。

このことから試験品質の検討に際し、試験工数でなく試験期間で検討しても問題ないと判断した。

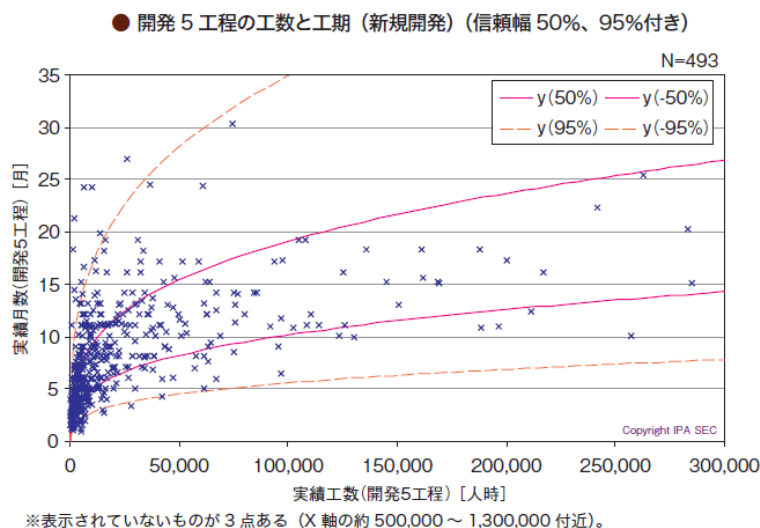


図 15-B-7-9 工数と工期の関係（新規開発）

出典：IPA/SEC（独立行政法人情報処理推進機構 ソフトウェア高信頼化センター）

「ソフトウェア開発データ白書 2010-2011」 p.124 図表 6-3-2

(2) 所属部門の試験期間の分析

試験期間が品質に影響を及ぼしているかを試験期間内に検出した不具合数、試験期間、試験項目数のメトリクスに対し分析した。結合試験・総合試験の工程に分け、ソフトウェア規模（ステップ数）で正規化して分析している。図 15-B-7-10 に示すとおり、結合試験工程では試験期間と不具合数の相関が 0.77 に対し、試験項目数と不具合数の相関は 0.33、総合試験工程では試験期間と不具合数の相関が 0.68 に対し、試験項目数と不具合数の相関は 0.25 となっていた。つまり、実績データでは、不具合数は試験項目数よりも試験期間に強い相関があったことを示している。

この結果から、以降の分析では試験期間に着目して分析する。

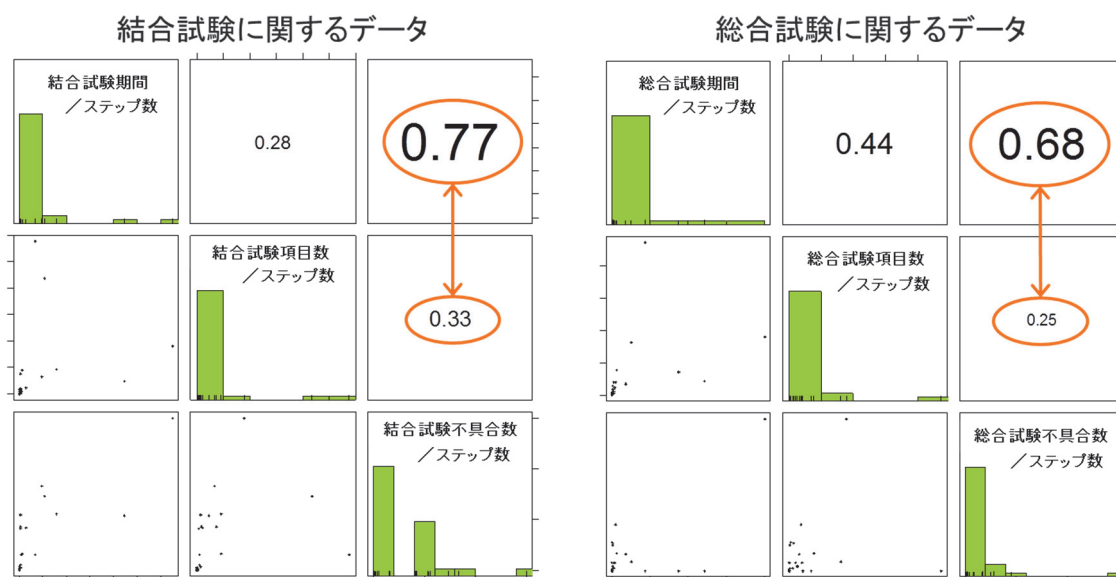


図 15-B-7-10 試験に関するメトリクスの相関

4.3. 試験期間を用いた分析

試験品質に関する分析を進めるにあたり、試験期間について、指標となるベンチマークの分析を行った。表 15-B-7-2 及び図 15-B-7-11 に示すとおり、一般的に試験工程の期間比率は、「新規開発」ではプロジェクト期間の約 25%を占めていることがわかった。

表 15-B-7-2 工程別の実績月数の比率の基本統計量（新規開発）

● 工程別の実績月数の比率の基本統計量（新規開発）

工程	N	最小	P25	中央	P75	最大	平均	[比率]
								標準偏差
基本設計	131	0.016	0.155	0.227	0.301	0.522	0.232	0.105
詳細設計	131	0.026	0.143	0.191	0.246	0.645	0.200	0.091
製作	131	0.047	0.211	0.263	0.359	0.902	0.285	0.116
結合テスト	131	0.016	0.094	0.143	0.185	0.386	0.145	0.068
総合テスト（ベンダ確認）	131	0.014	0.074	0.121	0.183	0.571	0.138	0.088

出典：IPA/SEC 「ソフトウェア開発データ白書 2010-2011」 p.202 図表 8-1-2

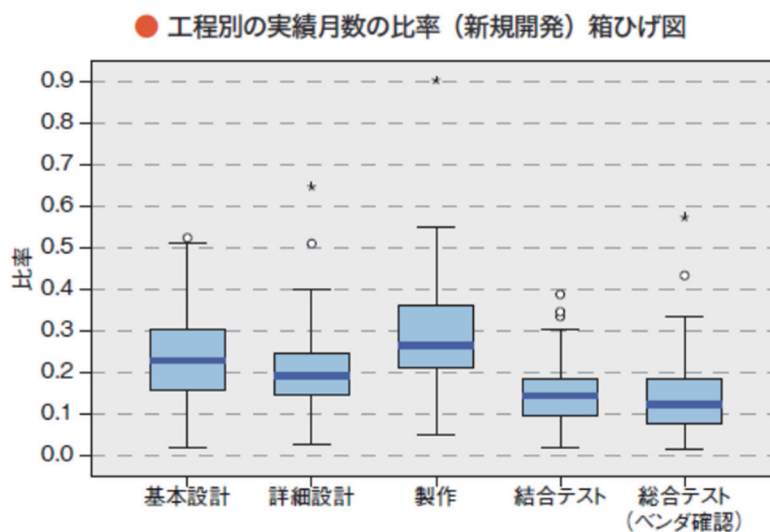


図 15-B-7-11 工程別の実績月数の比率（新規開発）

出典：IPA/SEC「ソフトウェア開発データ白書 2010-2011」 p.201 図表 8-1-1

また、図 15-B-7-12 に示す「改良開発」においても試験工程の期間比率はプロジェクト期間の約 30%であり、新規開発と大きな差はなかった。

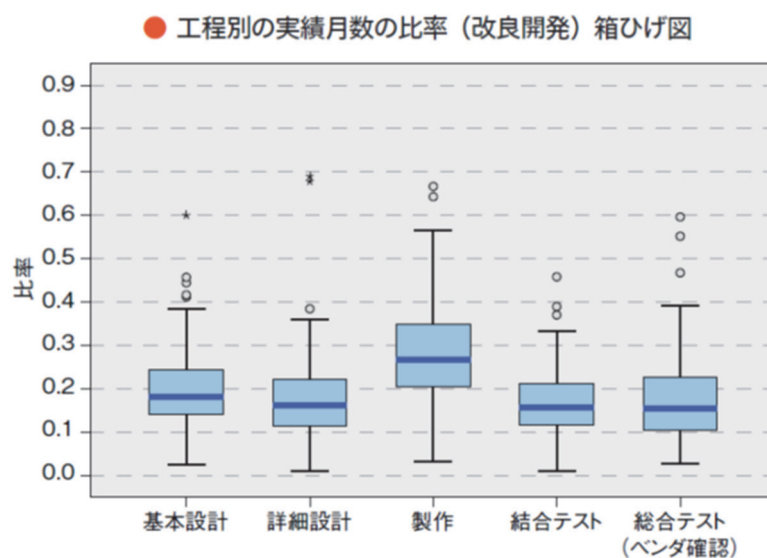


図 15-B-7-12 工程別の実績月数の比較（改良開発）

出典：IPA/SEC「ソフトウェア開発データ白書 2010-2011」 p.203 図表 8-1-4

さらに工数についても再度確認をしてみると、表 15-B-7-3 に示すとおり、「新規開発」、「再開発・改修」、いずれにおいても試験工程の工数比率はプロジェクト期間の約 30%を占めていた。

表 15-B-7-3 工程別の実績工数比率

規模別工程別工数比											
開発種別	全体工数	件数	実装工数を1とした比率			合計を100%とした比率			平均工数		
			設計工数	実装工数	テスト工数	設計工数	実装工数	テスト工数	設計工数	実装工数	テスト工数
新規開発	<10人月	13	0.81	1.00	0.53	21.48%	54.37%	24.15%	1.17	2.97	1.32
	<50人月	80	0.68	1.00	0.57	26.97%	50.29%	22.74%	5.56	10.37	4.69
	<100人月	32	0.80	1.00	0.68	28.74%	45.12%	26.14%	17.64	27.69	16.04
	<500人月	56	0.73	1.00	0.82	28.10%	42.39%	29.51%	59.87	90.32	62.87
	>=500人月	25	2.24	1.00	1.24	21.66%	40.87%	37.47%	220.20	415.37	380.85
	未回答	5	0.47	1.00	0.59	13.93%	56.22%	29.86%	32.98	133.14	70.72
	合計	211	0.90	1.00	0.73	23.80%	42.32%	33.88%	47.62	84.65	67.78
再開発・改修	<10人月	7	0.69	1.00	1.27	20.88%	40.58%	38.54%	0.99	1.92	1.83
	<50人月	59	0.64	1.00	1.07	23.76%	43.11%	33.13%	6.46	11.72	9.01
	<100人月	43	0.77	1.00	1.05	23.71%	44.62%	31.66%	13.34	25.09	17.81
	<500人月	53	1.03	1.00	1.21	25.15%	40.69%	34.16%	47.42	76.74	64.43
	>=500人月	17	0.64	1.00	0.69	25.49%	45.50%	29.01%	204.51	365.06	232.71
	未回答	3	0.61	1.00	0.71	23.61%	52.70%	23.69%	5.21	11.63	5.23
	合計	182	0.78	1.00	1.07	25.10%	43.57%	31.33%	38.28	66.44	47.78
合計	<10人月	20	0.77	1.00	0.79	21.29%	49.98%	28.73%	1.11	2.60	1.50
	<50人月	140	0.66	1.00	0.79	25.40%	46.61%	27.98%	5.97	10.95	6.58
	<100人月	77	0.77	1.00	0.89	25.79%	44.81%	29.40%	14.95	25.98	17.05
	<500人月	111	0.87	1.00	1.00	26.78%	41.61%	31.61%	53.62	83.31	63.29
	>=500人月	42	1.59	1.00	1.02	23.00%	42.49%	34.51%	213.85	395.01	320.89
	未回答	8	0.52	1.00	0.64	14.44%	56.03%	29.53%	22.57	87.57	46.16
	合計	398	0.84	1.00	0.89	24.33%	42.80%	32.87%	43.02	75.69	58.13

各開発種別の合計における「合計を 100%とした比率」をみると、再開発・改修の方が設計工数のウェイトが高くなっている。

出典：社団法人 日本情報システム・ユーザー協会

「ソフトウェア開発管理基準に関する調査報告書（ソフトウェアメトリクス調査）」

（平成 23・10・18 財情第 3 号） p.133 図表 6-162

ベンチマークとした上述の結果に対し、図 15-B-7-13 に示すように社内の試験（結合試験＋総合試験）期間比率は平均 30%程度であり、試験期間として少なくないと判断できた。しかし、図 15-B-7-7 に示すとおりソフトウェアの結合試験は約 12%の期間比率であり、総合試験の期間比率と比較すると少ないことが分かった。これを開発種別ごとに確認すると、図 15-B-7-13 に示すとおり、「新規システム開発」で、結合試験と総合試験の期間比率（中央値）に大きな差があることが分かった。

また、「既存システム改良開発」においても「総合試験」と比較すると「結合試験」の期間比率が低い。

表 15-B-7-1 に示している結果では、結合試験と総合試験の期間比率に大きな差はなく、結合試験の期間比率（中央値）は 0.143、総合試験の期間比率（中央値）は 0.121 であった。所属部門の実績データはベンチマーク結果と比較すると結合試験の期間比率が低いことが分かった。ここからソフトウェアの結合試験期間を見直し、試験を充実させることが品質向上への糸口とし、さらに「結合試験期間が十分確保できていない」という仮説を立て以降の分析を行う。

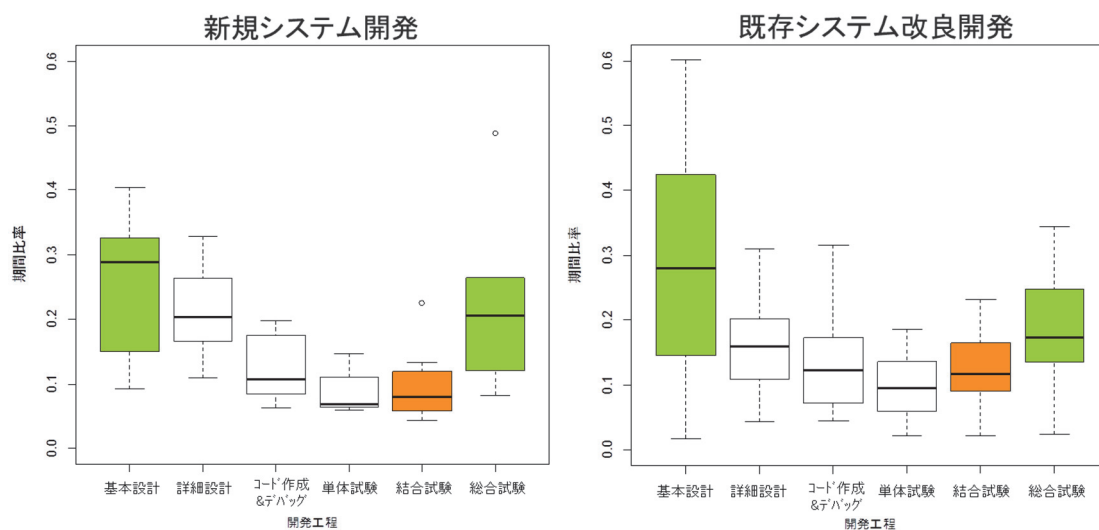


図 15-B-7-13 所属部門におけるソフトウェア開発期間比率（開発種別別）

4.4. 試験品質に関するメトリクス分析

(1) 相関分析による確認

試験品質を把握するため、試験期間（日数）、試験項目数及びシステム試験時の不具合件数をソフトウェアメトリクスとして相関関係を確認した。ここで、システム試験の不具合件数については、ソフトウェア規模（ステップ数）で正規化していない。ソフトウェアの開発中に混入した不具合の除去は総合試験終了までに実施することが試験品質に求められることであるため、システム試験で検出された不具合はソフトウェア規模に関係なくソフトウェアの試験品質の不足不具合とした。

これらのソフトウェアメトリクス間には、図 15-B-7-14 に示すとおり、あまり明瞭な相関関係は見られなかった。相関があるということは、2 変数に関連があり連動する関係になっているということであるが、ここではシステム試験時の不具合件数が試験期間比率に連動していることを厳密に分析したわけではない。そこで、別の手法の選択し以降において分析した。

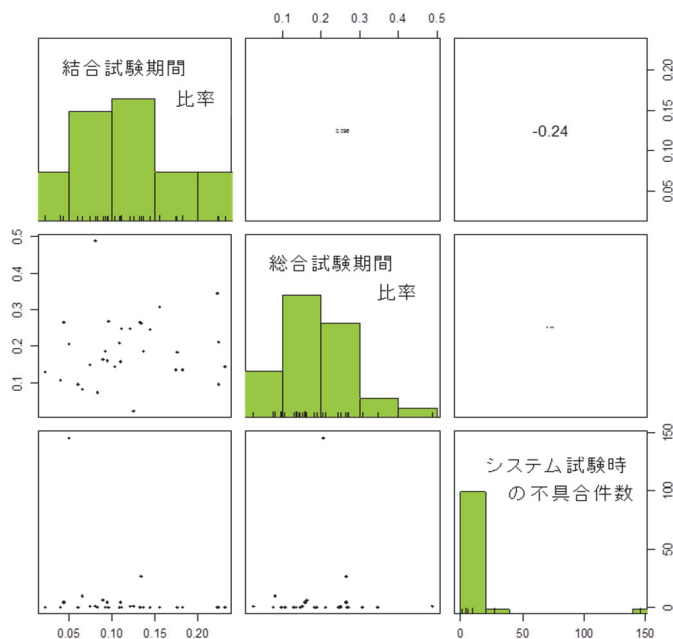


図 15-B-7-14 試験関連メトリクスの相関分析

(2) 2 群の平均値の差の確認

次のアプローチとして、2 群の平均値の差の検定という分析手法を選択した。ここでは、システム試験時の不具合の「無」、「有」の群（グループ）に分け確認をした。

各試験の期間比率をシステム試験時の不具合発生無、有の 2 群に分け、箱ひげ図により視覚的に確認したところ、図 15-B-7-15 に示すとおり結合試験の期間比率で不具合「有」と「無」に違いがあった。

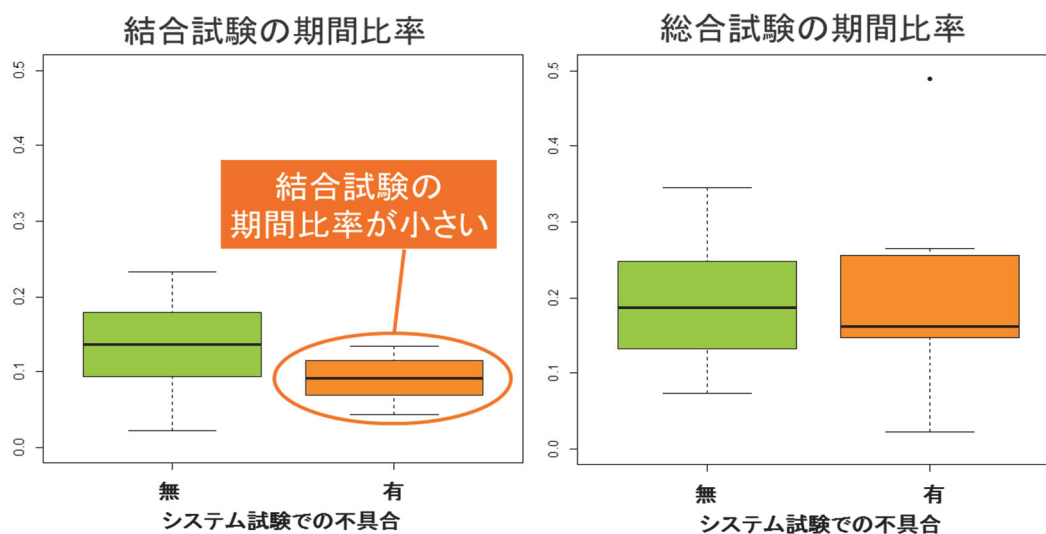


図 15-B-7-15 試験期間比率（不具合無／有の差）

相関分析は一つひとつのデータに着目し2つの変数が協調して動くかどうかを見ている。これに対し、2群の平均値の差の検定では、一つひとつのデータではなく、データ群とデータ群で比較している。つまり、相関分析の方が一つひとつの動きに敏感であり、2群の平均値の差の検定は鈍感ではあるといえる。鈍感であるとは、統計手法の用語では「ロバストである」と表現する。ロバストな手法の方が誤差などによる無意味なデータ変動にかく乱されることなく真の結果を導きやすい[2]。

今回のソフトウェアのデータ分析においては、2群の平均値の差の検定というアプローチが有効に使えるといえる。

(3) 正規性の確認

(2) 2群の平均値の差の確認に示すとおり、結合試験の期間比率に対して、システム試験時の不具合の無、有で差があることが視覚的に分かった。この検定を実施するにあたり、まずは正規性を確認した。これは、正規性の有無で使用する分析法を変えるためである。

図 15-B-7-16 に示すとおり、シャピローウィルクの正規性の検定結果は、p 値（偶然に起こり得る確率）が 0.05 より大きかった。このことから、システム試験においては、「不具合無」、「不具合有」ともに結合試験の期間（日数）比率には正規性がないとは言えないことが分かった。

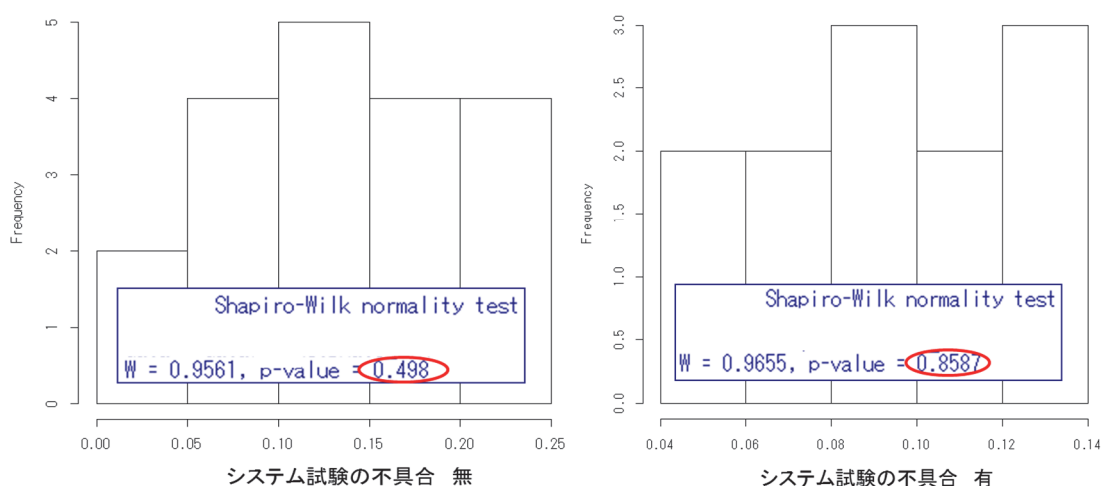


図 15-B-7-16 正規性の検定

(4) 検定の実施

システム試験の不具合「無」、「有」ともに結合試験の期間（日数）比率に正規性がないとは言えないため、分析としてはt検定を採用した。

検定を行った結果、p 値（偶然に起こり得る確率）が 0.05 以下であり、システム試験時の不具合の有無でソフトウェアの結合試験の期間比率に統計的に有意な差があること

が分かった。システム試験までに不具合を除去するためには、ソフトウェアの結合試験工程を充実させることが必要であることが分析に結果から導出された。

結合試験の期間比率について、「新規システム開発」、「既存システム改良開発」で分けて確認を行った。図 15-B-7-17 システム試験における不具合有無別の結合試験の期間比率に示すとおり、いずれも、「不具合無」の方が結合試験の期間比率が高かった。実績データでは、「新規システム開発」は「不具合無」のデータが少なかったため、検定を実施することはできなかった。このため、箱ひげ図による視覚的な確認のみとなったが、明らかに違いがあった。

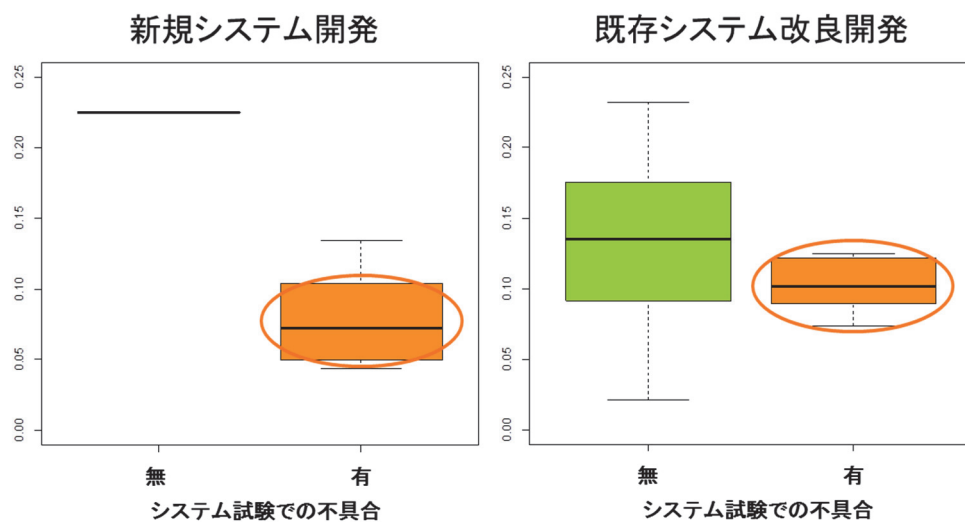


図 15-B-7-17 システム試験における不具合有無別の結合試験の期間比率

4.5. 必要な試験期間の設定

以上の分析結果から、ソフトウェアの結合試験の期間比率（日数）の基準を 15% とし、ソフトウェアの開発計画策定時に設定してもらうこととした。

本設定値は、システム試験時の不具合が「無」の場合の平均 13.8% を参考にした値である。

5. 達成度の評価、取り組みの結果

「結合試験期間が十分確保できていない」という仮説に対して、ソフトウェアの試験工程において（即ち、システム試験前までに）不具合を除去するためには、結合試験工程を充実させることが必要であることが実績データから導出された。試験期間と不具合数という収集可能かつ収集容易なメトリクスを用い、品質向上に寄与できる分析結果が得られたことは成果として大きいと考えている。

本成果については、設計部門、品質保証部門等関係部門とともに試験工程におけるソフトウェアの段階的な結合試験を十分実施することが品質向上に繋がるという再認識ができた。

結合試験の期間をかけることは後工程への期間を圧迫させるが、逆にこの期間をかけることにより総合試験での不具合対応等もなくなるという考え方で共通理解が得られている。現在、試験計画の策定時に設定した目標値を推奨している状況にある。

実績データの分析結果を用い、当たり前のことができていない現状を定量的に捉え、納期までの期間制約の中で品質向上に取り組む策を打出すことができた。

これらの結果として、選択した分析手法（2群に分けたアプローチ）の有効性を確認した。本アプローチはソフトウェアの品質向上を目的とした分析において、今後も活用できる。

6. 今後の取組みと考察

システム試験でソフトウェアの不具合をなくすためには、結合試験の期間を確保することが有効であることが分析結果から知見として得られた。本事例においては、工数が測定できなくとも期間データ（工期）を用い、不具合低減のための分析が実施できることを確認した。また、分析アプローチとして、2群の平均値の差による分析がソフトウェアのデータ分析において有効であり、不具合の有無による差を箱ひげ図で視覚化し、検定を実施することで結果が導きやすくなった。

今後は、完了プロジェクトからメトリクス収集し、実績を評価し、設定した15%に対する評価と設定値の見直しを実施しながら改善を図っていく。

今回は、管理者層、プロジェクトマネージャ、システム担当者がイメージしやすい「試験期間の比率」に着目した。期間と工数は概ね比例していることが分かり、統計的に分析することができた。

十分性を追及するためにも、さらに不具合の原因工程、原因及び試験項目数等のメトリクスを用い、統計手法を活用し品質向上・手戻り削減に繋がられる項目の分析を続けていきたい。

参考文献

- [1] 中澤港：Rによる統計解析の基礎、ピアソンエデュケーション、2003
- [2] 野中誠、小池利和、小室睦：データ指向のソフトウェア品質マネジメントーメトリクス分析による「事実にもとづく管理」の実践、日科技連、2012
- [3] IPA/SEC：ソフトウェア開発データ白書 2010-2011、2010
- [4] 社団法人 日本情報システム・ユーザー協会：ソフトウェア開発管理基準に関する調査報告書（ソフトウェアメトリックス調査）、2012.2

掲載されている会社名・製品名などは、各社の登録商標または商標です。

独立行政法人情報処理推進機構 技術本部 ソフトウェア高信頼化センター（IPA/SEC）