組込みソフトウェア開発
データ白書
2019

プロジェクトの成功/失敗要因を
データから検証しているか!

独立行政法人情報処理推進機構
社会基盤センター

組込みソフトウェア開発
データ白書
2019

プロジェクトの成功／失敗要因を
データから検証しているか!

独立行政法人情報処理推進機構
社会基盤センター

IPA Better Life with IT

本書の内容に関して

商　標

刊行にあたって

「組込みソフトウェア開発データ白書」の2019年版が刊行となりました。本白書は、IPAの社会基盤センターが中心となり、組織や企業をまたがって収集した多数の開発データを整理、分析したものです。貴重な開発プロジェクト・データベースとなっており、本書は前回2017年に続き3回目の発刊となります。

デジタルトランスフォーメーション（DX）の必要性が広く叫ばれるようになった昨今、企業は、既存のビジネスから脱却して、新しいデジタル技術を活用することによって新たな価値を生み出していくことが求められています。また、DXは、「Connected Industries」が示すように、サイバー空間だけでなく、様々な機器や設備などフィジカルな領域も巻き込んで進んでいきます。日本企業はこの領域で多くの実績があり、様々なビジネスノウハウを蓄積しており大きなチャンスとなる可能があります。

個々の企業がDXを着実に実行し、「Connected Industries」を実現していくためには、組込みソフトウェアの果たす役割はますます重要なものとなってきました。そして、組込みソフトウェアの開発においては、技術者の職人的な作業から脱却し組織的な開発への移行を推進することがより一層重要となっており、定量データを活用して工学的に行っていく必要があります。

「組込みソフトウェア開発データ白書」では、定量データに基づいたソフトウェア管理の推進を目的とし、指標値となるように、プロジェクトデータと分析結果を公開しています。本書を用いれば、自社の開発データと比較することで、プロジェクト計画の参考とすることができます。また、自社の開発プロジェクトでの欠点や弱点もわかり、その後のプロジェクトにフィードバックすることが可能になります。

2019年版では、2017年版に対して新たに提供を受けたデータを加え、累計599件のプロジェクトデータを分析するとともに、新たな分析テーマとして、製品リリース後の品質が良いプロジェクトと悪いプロジェクトの差異分析を行いました。この分析結果を参考にすると、自社の定量データと市場品質データを照らし合わせることによって、次の開発に収集したデータを生かせることが可能となります。本書はこのような視点で活用いただければと思います。

IPAでは、本書以外にもエンタプライズ分野向けに、定量データに基づいたソフトウェア開発管理の推進を目的とした、「ソフトウェア開発データ白書」を発刊しています。さらに、定量データの活用法や分析手法など定量的な品質管理の方法について解説した「定量的品質予測のススメ」や、定量データから得られた品質マネジメントに関するいくつかの知見をまとめた「ソフトウェア開発データが語るメッセージ」など、定量分析による実践的なプロジェクト管理方法の手引きを多数刊行しています。これらの出版物は、以下のIPAのウェブサイトにて公開していますので、ダウンロードの上ぜひご活用をお願いします。

- 『ソフトウェア開発データ白書』
  https://www.ipa.go.jp/sec/publish/tn12-002.html
- 『定量的品質予測のススメ』
  https://www.ipa.go.jp/sec/publish/tn08-004.html
- 『続 定量的品質予測のススメ』
  https://www.ipa.go.jp/sec/publish/tn10-004.html
- 『ソフトウェア開発データが語るメッセージ2015』
  https://www.ipa.go.jp/sec/reports/20150925.html
- 『ソフトウェア開発データが語るメッセージ「設計レビュー・要件定義強化のススメ」』
  https://www.ipa.go.jp/sec/reports/20170331.html
- 『ソフトウェア開発データが語るメッセージ2017』
  https://www.ipa.go.jp/sec/reports/20180306.html
- 『組込みソフトウェア向けプロジェクトマネジメントガイド［定量データ活用編］』
  https://www.ipa.go.jp/sec/publish/tn15-002.html

最後に、本白書の成果は、これまで門外不出としてほとんど公開されることのなかった貴重なプロジェクトデータを惜しみなくご提供くださった多くの企業や、データ収集・分析に関わる多くの知見を持ち寄り検討していただいた関係者の皆さまのおかげです。2019年の刊行に際して改めて敬意を表すとともに、この場をおかりしましてご協力に対して厚く御礼申し上げます。

独立行政法人情報処理推進機構
社会基盤センター
センター長　**片岡　晃**

組込みソフトウェア開発データ白書2019

目次

刊行にあたって 5

1章 本書の目的と位置付け 9

1.1 はじめに ～本書刊行の目的～ 10
1.2 なぜデータの定量管理が求められるのか 10
1.3 組込みソフトウェア開発を取り巻く現状 10
1.4 公的機関の立場を活かしたIPAの取り組み 11
1.5 2019年版について 11
1.6 想定読者 11
1.7 収集データについて 12
1.8 構成 14
1.9 利用の留意点 14

2章 「組込みソフトウェア開発データ白書2019」のメッセージとポイント 15

2.1 「組込みソフトウェア開発データ白書2019」のメッセージ 16
2.2 「組込みソフトウェア開発データ白書2019」のポイント 17

3章 分析について 23

3.1 分析の進め方 24
3.2 分析に関する事前の取り決め 26
3.3 分析結果の取り扱い 28

4章 収集データのプロファイル 33

4.1 プロファイル一覧 34
4.2 開発プロジェクトの全般的な特徴 37
4.3 利用局面 39
4.4 システム特性 40
4.5 開発の進め方 44
4.6 ユーザ要求管理 45
4.7 要員の経験 46
4.8 規模 47
4.9 工期 48
4.10 工数 50
4.11 体制 52
4.12 信頼性 53
4.13 実施工程の組み合わせパターン 58
4.14 プロジェクト成否 59

5章 プロジェクトの主要要素の統計 61

5.1 位置付け 62
5.2 SLOC規模 63
5.3 工数 71

6章 工数、工期、規模の関係の分析 79
6.1 位置付け 80
6.2 工数と工期 81
6.3 SLOC規模と工数 83
6.4 工程別の工期、工数 85
7章 生産性の分析 87
7.1 位置付け 88
7.2 SLOC生産性 89
7.3 工程別SLOC生産性 92
8章 信頼性の分析 93
8.1 位置付け 94
8.2 SLOC規模と検出バグ密度 95
8.3 テストケース数と検出バグ数、テスト工数 102
8.4 SLOC規模とテスト工期 108
8.5 工数あたりのテストケース数、検出バグ数 110
9章 信頼性と生産性の関係 113
9.1 位置付け 114
9.2 テスト検出バグ密度と生産性 115
10章 製品の特性別の分析 119
10.1 位置付け 121
10.2 リアルタイム性（時間制約） 123
10.3 自然環境からの影響度合い 130
10.4 ユーザの多様性 136
10.5 法規等による規制度合い 142
10.6 M2Mの有無 148
10.7 ネットワーク接続の有無 154
10.8 稼動（非停止、オンデマンド） 160
10.9 オンライン保守の可否 166
10.10 障害リスク（TYPE） 172
11章 品質評価別の分析 177
11.1 位置付け 178
11.2 品質実績の評価 179
付録 195
付録A データ項目の定義 196
付録B 用語集 215
付録C 参考文献・参考情報 217
図表一覧 218
監修 234

1章　本書の目的と位置付け

1.1　はじめに ～本書刊行の目的～ …… 10
1.2　なぜデータの定量管理が求められるのか …… 10
1.3　組込みソフトウェア開発を取り巻く現状 …… 10
1.4　公的機関の立場を活かしたIPAの取り組み …… 11
1.5　2019年版について …… 11
1.6　想定読者 …… 11
1.7　収集データについて …… 12
1.8　構成 …… 14
1.9　利用の留意点 …… 14

1章　本書の目的と位置付け

1.1　はじめに　～本書刊行の目的～

IPAは「作業計画と進捗（Delivery）」「コスト見込と実績（Cost）」「品質計画と実績（Quality）」の改善を目的に、組込みソフトウェア業界に対して定量データの活用推進を働きかけている。開発プロジェクトの定量的な管理がなされていない企業・組織に対して、定量的データに基づく開発を行わなければ、日本の組込みソフトウェアの発展が停滞するとの危機感を持ったからである。また、すでにデータの定量管理を導入済みの企業・組織に対し、その指標となる具体的な数値（指標値）を提供する必要があったことも理由の一つである。

1.2　なぜデータの定量管理が求められるのか

定量管理導入の最大のメリットは、QCD可視化によるリスクマネジメントの実現である。組込みソフトウェア開発の黎明期には、ソフトウェアは「見えないもの」「動作させなければ分からないもの」といわれ、職人が経験と勘で作っていた。そのため、ソフトウェアは「属人的」といわれ、その内容の不透明さから、経営リスクの一因となっていた。

定量管理の導入により可視化が実現すれば、ソフトウェア開発は、特定の開発者への依存から組織的形態へ転換が可能となり、ソフトウェアの大規模化や複雑化に対応できる。特に、色々なものがつながるIoT（Internet of Things）時代において、ますます大規模化・複雑化する組込みソフトウェア開発には、定量管理が必要不可欠となる。

もう1つ重要なのは、ビッグデータ活用の視点である。定量管理の導入により、プロジェクト管理データを収集できれば、それらを分析することで新たな知見やインテリジェンスを得られる。今後は人工知能（AI）がプロジェクト管理データからQCDを判断し、プロジェクトコントロールを支援すると予想される。その際、多くのデータを人工知能に学習させることが、より効率的なプロジェクトコントロールを実現するカギとなる。

1.3　組込みソフトウェア開発を取り巻く現状

色々なものがつながり新たな価値を創出する『つながる世界』やAIの活用など、技術の発展とソフトウェアによる付加価値向上という流れの中で、組込みソフトウェア開発は、複雑化・高度化の一途をたどっている。

2018年「組込み／ IoTに関する動向調査」[1]において、「開発の課題」のトップは「設計品質の向上」であり、「設計能力（量）の向上」、「市場の拡大、新規市場の開拓」、「新製品・新技術の開発」、「開発コストの削減」を挙げた企業の2倍近くある。

「設計品質の向上」に対する解決策としては、第1位「技術者のスキル向上」、第2位「プロジェクトマネージャのスキル向上」、第3位「開発手法・開発技術の向上」、第4位「技術者の確保」、第5位に「管理手法・管理技術の向上」が挙がっている。

解決策の経年変化を見ると、「技術者のスキル向上」、「開発手法・開発技術の向上」は減少傾向、「管理手法・管理技術の向上」は増加傾向である。このことから複雑化・高度化する開発においては管理の重要性が改めて見直されていると考えられる。

[1] 2018年「組込み／ IoTに関する動向調査」 https://www.ipa.go.jp/ikc/reports/20190327.html

1.4 公的機関の立場を活かしたIPAの取り組み

IPAは、現在の社会基盤センターの前身であるソフトウェアエンジニアリングセンター設立以来、組込みソフトウェア産業界のベストプラクティスを収集／整備し、「組込みソフトウェア開発リファレンスガイド ESxRシリーズ」を発行してきた。同書は産業界に広く普及し、各社の組込みソフトウェア開発標準の制定等に活用されてきた。このような流れの中、組込み開発業界内でも企業内のプロジェクト管理データを共有する基盤ができつつあると考えた。

IPAは、これまでのESxRシリーズ編さん活動を通して、企業にとっての機微情報を公的機関の立場で責任を持って取り扱うことに対する信頼を得ていた。そのため、プロジェクト管理データについても特定の企業や業種に偏ることなく、さまざまな企業のデータを収集／分析することが可能であると判断し、2013年より「組込みソフトウェア開発データ白書」の編さん活動を開始し、初版を2015年に発行した。

1.5 2019年版について

本書「組込みソフトウェア開発データ白書2019」は、前回2017年版に続き3回目の発行となる。本書では2017年版で分析したデータベースに、新たに提供を受けたデータを加え、累計599件のプロジェクトデータを分析している。

2019年版では、分析対象データの標本数の増加により2017年版よりも深い分析が可能になったため、品質視点でのプロジェクトの成功／失敗の要因分析を新たに追加している。

1.6 想定読者

本書は、主として以下の読者を想定している。

○プロジェクトマネージャ／プロジェクトリーダ

ソフトウェア開発プロジェクトを成功させるためには、定量データを用いたプロジェクト管理の推進が重要になる。プロジェクトを仕切るマネージャ／リーダは、本書のデータを定量データの1つとして参考／活用してほしい。

○プロジェクトマネジメントオフィス／品質保証部門

本書は、プロジェクト管理データのデータベース構築やプロジェクトのベンチマーキングにも活用できるように、収集したデータの定義とデータを分析した際の条件を分析毎に掲載している。収集データの定義は、付録A「データ項目の定義」に纏めている。プロジェクトマネジメントオフィスや品質保証部門など、プロジェクト管理に責任を持つ部門には特に、本書を参考／活用してほしい。

1.7 収集データについて

本書に掲載したデータは、国内企業15社から収集した、599件のプロジェクトデータである。対象は、組込みソフトウェアを開発するプロジェクトである。組込みソフトウェアの範囲は、機器や製品に組み込まれたソフトウェアと、プラントやインフラを制御するソフトウェアとしている。

1.7.1 収集の基本方針

収集データの項目は主に実績に関するものであるが、主な要素（規模、工数、工期等）では計画データについても収集した。

重点収集したデータの詳細は下記に示す通りである。

- 開発プロジェクトの種別：新規開発、改良（派生）開発
- 製品の特性：リアルタイム性（時間制約）、自然環境からの影響度合い、ユーザの多様性、法規等による規制度合い、M２Mの有無、ネットワーク接続の有無、稼動（非停止、オンデマンド）、オンライン保守の可否
- 実績の評価：Q（品質）、C（コスト）、D（工期）各計画に対する実績の評価
- 開発言語：アセンブラ、C、C++、C#、Javaなど
- CPUアーキテクチャ：シングルチップかマルチチップか、シングルコアかマルチコアかなど
- OSアーキテクチャ：リアルタイムOS利用か非リアルタイムOS利用か
- 開発対象プラットフォーム：Windows系、Linux系、ITRON系、OSレスなど
- 規模：SLOC（コード行数）
- 工数：各工程の工数
- 工期：プロジェクト開始年月日と終了年月日、各工程の開始年月日と終了年月日
- 信頼性データ：テストケース数、テスト検出バグ数、レビュー工数、レビュー指摘件数

1.7.2 収集方法

データ提供企業と秘密保持契約を結び、組込みソフトウェア開発プロジェクトを対象にプロジェクト実績データを収集した。

なお、収集したデータは無作為抽出されたものではなく、各データ提供企業が任意に選択したものである。

◆収集期間

2013年10月～ 2019年3月

◆収集データ項目

付録Aに、収集に使用したデータ項目の定義を示す。

1.7.3 データの精査

データの収集活動においては、次に示すようにデータの精度向上に注力した。

- 収集するデータは、データ提供企業の品質保証部門や生産管理部門で精査、またはレビューを受けた信頼できるデータを前提としている。
- IPAではデータの精査を実施。異常値や誤記と見られるデータについては、データ提供企業に問い合わせた。この作業を何度か繰り返し、分析の基礎データとした。

1.7.4 データ提供状況

◆データ提供企業件数

本書に収録したデータの提供元は15企業。

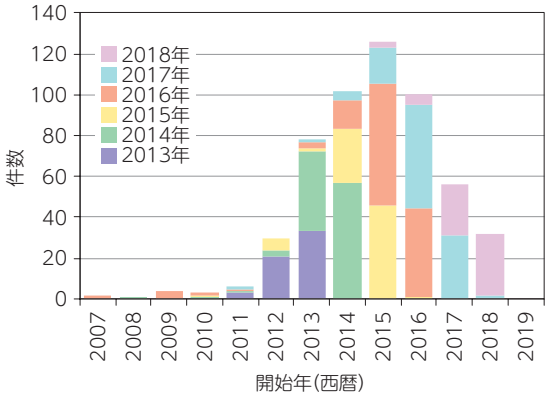
◆データ提供件数

本書に収録したデータは累計599件。

◆プロジェクトデータ収録年度別、プロジェクト開始年・終了年別クロス集計

本書に収録したデータのプロジェクトの開始年と終了年について、収録した年度別のデータ件数を図表1.7-1及び図表1.7-2に示す。また、プロジェクトの開始年と終了年をクロス集計したものを図表1.7-3に示す。

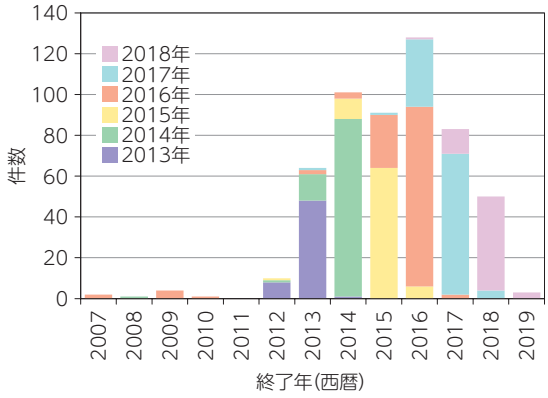
図表1.7-1 プロジェクトの開始年ごとの収録年度別データ件数



N=540

※集計対象データ：
8-4-1_プロジェクト全体工期(実績)_開始年月[日]

図表1.7-2 プロジェクトの終了年ごとの収録年度別データ件数



N=538

※集計対象データ：
8-4-2_プロジェクト全体工期(実績)_終了年月[日]

図表1.7-3 収録年度別のプロジェクト開始年及び終了年のクロス集計

データ収録年度	時期	プロジェクトの開始年及び終了年ごとの件数													総計
		2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	
2013年	開始年	0	0	0	0	3	21	33	0	0	0	0	0	0	57
	終了年	0	0	0	0	0	8	48	1	0	0	0	0	0	57
2014年	開始年	0	1	0	1	1	3	39	57	0	0	0	0	0	102
	終了年	0	1	0	0	0	1	13	87	0	0	0	0	0	102
2015年	開始年	0	0	0	1	0	6	2	26	46	1	0	0	0	82
	終了年	0	0	0	0	0	1	0	10	64	6	0	0	0	81
2016年	開始年	2	0	4	1	1	0	3	14	59	43	0	0	0	127
	終了年	2	0	4	1	0	0	2	3	26	88	2	0	0	128
2017年	開始年	0	0	0	0	1	0	1	5	18	51	31	2	0	109
	終了年	0	0	0	0	0	0	1	0	1	33	69	4	0	108
2018年	開始年	0	0	0	0	0	0	0	0	3	5	25	30	0	63
	終了年	0	0	0	0	0	0	0	0	0	1	12	46	3	62

1.8 構成

本書の構成は以下のとおりである。

- 1章：本書の目的と位置付け
- 2章：「組込みソフトウェア開発データ白書2019」のメッセージとポイント
- 3章：分析について
- 4章：収集データのプロファイル
- 5章：プロジェクトの主要要素の統計
- 6章：工数、工期、規模の関係の分析
- 7章：生産性の分析
- 8章：信頼性の分析
- 9章：信頼性と生産性の関係
- 10章：製品の特性別の分析
- 11章：品質評価別の分析

6章～9章では、組込みシステム全体の傾向を分析。10章では、製品の特性別の傾向を分析している。2019年版で新たに追加した11章では、品質実績の評価別の傾向を分析している。分析対象は改良（派生）開発のみであり、開発言語は組込みソフトウェア開発の大半で使われるC言語とC++言語に限定している。

1.9 利用の留意点

本書に収録されている内容は、組込みソフトウェア開発関連企業から提供していただいたプロジェクトデータを統計的に分析したものであるが、業界の平均値や標準値ではない。組込みソフトウェア開発に定量的な管理データの収集を既に導入、もしくは、これから導入しようとしている企業や組織が、データの収集や収集データの分析を行う際の参考値として、本書が利用されることを想定している。

※　※　※

本書が、組込みソフトウェア開発各企業における定量データの活用やQCD改善に少しでも貢献できれば幸いである。

2章 「組込みソフトウェア開発データ白書2019」のメッセージとポイント

2.1 「組込みソフトウェア開発データ白書2019」のメッセージ …… 16

2.2 「組込みソフトウェア開発データ白書2019」のポイント …… 17

2章 「組込みソフトウェア開発データ白書2019」のメッセージとポイント

「組込みソフトウェア開発データ白書」では、2015年の発行以来、組込みソフトウェア開発プロジェクトの管理データを収集し統計的な分析を行い、その結果を公開している。2019年版では、成功プロジェクトと失敗プロジェクトに着目して新たな分析を行った。この章では、今回の分析結果から「組込みソフトウェア開発データ白書2019」が発するメッセージと、2019年版分析結果のポイントを掲載する。

2.1 「組込みソフトウェア開発データ白書2019」のメッセージ

◆サブタイトルに込めた意図

プロジェクトの成功／失敗要因をデータから検証しているか！

■成功するプロジェクトと失敗するプロジェクトには違いがあるのか？

「組込みソフトウェア開発データ白書2019」では、“成功したプロジェクトと失敗したプロジェクトとでは作業傾向が異なる”と仮定し、QCD（品質、コスト、工期）の各視点からプロジェクトデータの比較分析を行った。想定していた分析結果が得られたのはもちろんだが、全く想定外の傾向に気づかされることもあった。これはデータ検証をしてみないと分からないことである。

一例として、Qの視点でリリース後品質が良い場合と悪い場合を比べたところ、上流工程の設計レビューの違いによって、リリース後品質に異なる傾向が見られた。

■プロジェクトの失敗要因は突き止められるはず

リリース後品質が悪かった場合、レビュー体制が整っていなかったのか、要求仕様書が充実していなかったのか等、何らかの原因がプロジェクトや組織にあったと考えられる。それらをデータから突き止めることができれば、改善が可能である。

■自組織のデータで分析して成功要因、失敗要因を究明してほしい

ソフトウェア開発の作業内容等、取り組みの見直しを考えている企業や組織の方は、組込みデータ白書2019の分析事例を参考に、自組織で蓄積した成功／失敗双方のプロジェクトデータを比較し、開発過程でどのような違いがあるのかを調べてほしい。

その結果から、要因究明のために現状作業を注意深く見直し、プロジェクトを成功に導いてほしい。それと同時に、データ検証の重要性を今一度考えるきっかけにしてほしい。

2.2 「組込みソフトウェア開発データ白書2019」のポイント

2019年度版では、成功／失敗をテーマに、新たにプロジェクトを分析しているが、その結果、リリース後品質の視点で成功したプロジェクトと失敗したプロジェクトでは、作業傾向に違いが見られた。分析結果の詳細データは11章に示しているので、ここでは、分析結果のポイントを掲載する。

2.2.1 プロジェクトの成功率

(1) プロジェクト成功の定義とその割合

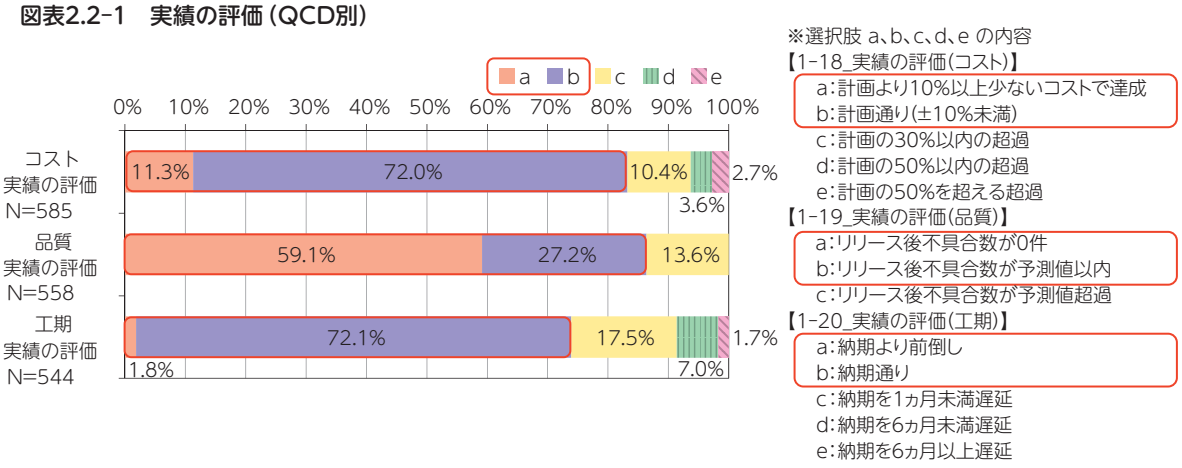
プロジェクトの成功／失敗の視点を、コスト、品質、工期（納期）に置き、それぞれの実績評価（計画値との差異）をa、b、cの３段階またはa、b、c、d、eの５段階で表した。実績評価のaとbを成功、c以下を失敗とみなした。図表2.2-1にその定義と評価結果を示す。成功は赤枠で囲んでおり、コストを例にすると、選択肢「a:計画より10%以上少ないコストで達成」と選択肢「b:計画通り（±10%未満）」が成功となる。

評価の結果は、

- コストの視点で成功したプロジェクトは全体の約80%
- 品質の視点で成功したプロジェクトは全体の約85%
- 工期の視点で成功したプロジェクトは全体の約70%

となっている。

図表2.2-1　実績の評価（QCD別）



(2) QCD各視点のプロジェクト成功と失敗の意味

組込み白書2019発行のために収集したプロジェクトデータのうち、ほとんどのプロジェクトがQCD目標を定めている。そこで、QCDそれぞれについて成功／失敗を評価することの意味を考えた。

コスト目標遵守の重要性は、プロジェクトの性質に依存する。要求仕様書に基づいた厳密なコスト見積もりによる請負契約のプロジェクトでは最重要視される。一方、必要な追加機能が判明した時に予算超過が認められるプロジェクトの場合、当初のコスト目標を遵守することが必ずしも重要ではない。また、マーケティングの観点で製品の付加価値を追求しながら開発するプロジェクトでは、製品の販売計画に対して実績を評価することに意味があり、開発コスト実績の評価は無意味である。

品質の場合、試作品と量産品とでは目標レベルは異なるが、組込みソフトウェアを開発する場合、有形無形にかかわらず品質目標が存在する。リリース後の品質問題は、経営上のマイナスになり得るため、品質が作り込まれていることを確認した上でリリースする。よって品質面の成功／失敗の評価は、以降の開発プロジェクトで同じ失敗を起こさないために非常に重要である。

工期目標は、予め決められた納期を短縮しなければならないケースや、逆に機能の追加や他の外的要因により納期延長を指示される等、変更を余儀なくされる場合がある。また、品質目標を達成できずに納期遅延を起こすこともある。工期目標の遵守は、プロジェクト計画の変更によるコスト増加や品質目標の達成状況に影響されるため、工期視点単独でのプロジェクト評価はあまり意味が無い。

このような考えに基づいて、プロジェクトの成功と失敗をQCDの各視点で分析した結果を次項に示す。

2.2.2 品質視点でのプロジェクトの成功と失敗

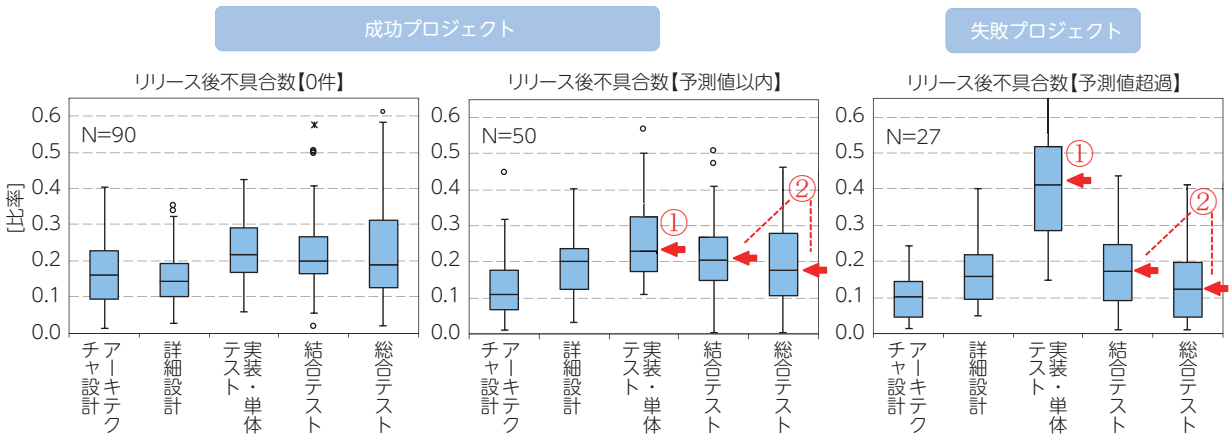
品質実績の評価の視点で、プロジェクトデータを成功プロジェクトと失敗プロジェクトに分けて、生産性、設計レビュー、テストケース数、テスト検出バグ件数、工数の工程別配分を分析し比較した。その結果は11章に掲載しているが、その中から、(1)工程別工数比率、(2)レビュー工数密度とレビュー指摘密度、(3)テストケース密度とテスト検出バグ密度について考察した結果を以下に示す。

(1)工程別工数比率

●ポイント1●　リリース後の品質が悪いプロジェクトは、実装・単体テスト工程の作業工数比率が高く、テスト工程の作業工数比率が低い。

リリース後品質とプロジェクト期間中の設計、実装、テストの工程に配分した実績工数の比率に関係があるかどうか分析した。

図表2.2-2　品質実績の評価別の工程別の実績工数の比率



図表2.2-2に、品質実績の評価別の工程別工数比率を比較した結果を示す。成功プロジェクトのうちリリース後不具合数【0件】と【予測値以内】では、傾向に違いが見られなかったが、失敗プロジェクトとみなしたリリース後不具合数【予測値超過】と比べると違いがあった。【予測値超過】は、【予測値以内】に比べて実装・単体テストに配分した工数比率が高く(①を比較)、結合テスト及び総合テストに配分した工数比率が低かった(②を比較)。

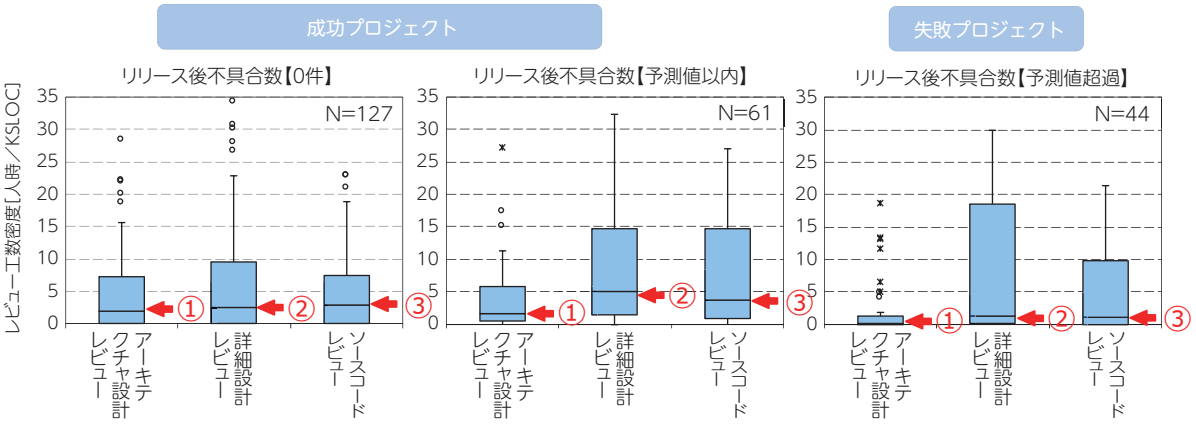
この結果から、実装・単体テストに予定より工数を使ってしまうと結合テストや総合テストで品質を検証する工数が削られ、結果的に品質が担保されないままリリースされてしまうことが考えられる。実装・単体テストに予定より多くの工数を使ってしまうのは、アーキテクチャ設計や詳細設計の品質が低いことが要因として考えられる。

(2) レビュー工数密度とレビュー指摘密度

●ポイント2●　リリース後の品質が良いプロジェクトは、アーキテクチャ設計レビューに工数を多くかけている。

“品質は上流工程で作り込まれる”といわれるため、設計やコーディングレビューのレビュー工数密度とレビュー指摘密度について、リリース後品質の良否に関係があるかどうかを分析した。

図表2.2-3　品質実績の評価別の工程別レビュー工数密度

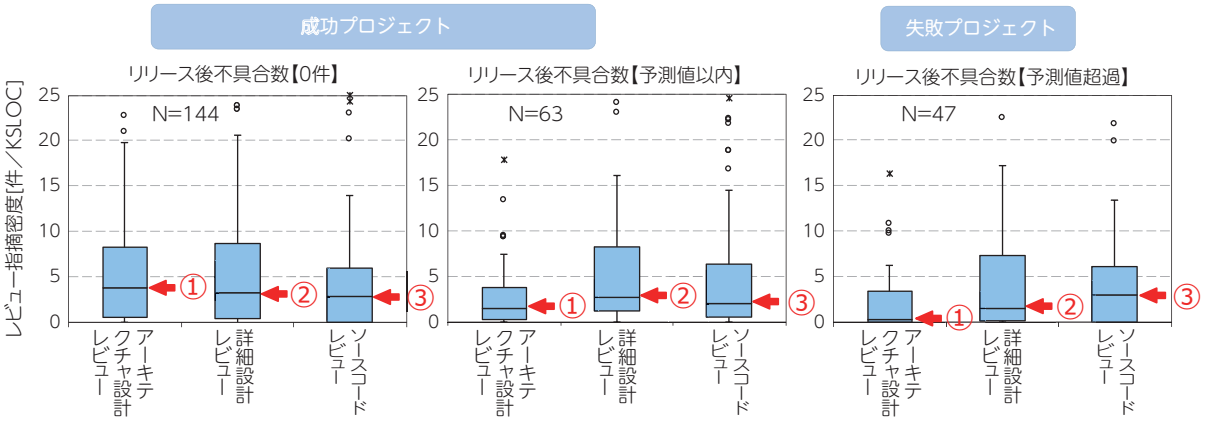


図表2.2-3に、品質実績の評価別の「レビュー工数密度」を比較した結果を示す。

アーキテクチャ設計レビューの工数密度を比較すると（①を比較）、リリース後不具合数【0件】や【予測値以内】の場合に比べて、リリース後不具合数【予測値超過】は、箱ひげ図の上辺が低く中央値も0に近い。この傾向から、リリース後に不具合が予測値よりも多く発生したプロジェクトは、アーキテクチャ設計レビューの工数が著しく少ないことが分かる。

詳細設計レビュー（②を比較）やソースコードレビュー（③を比較）を同じ様に比べて見ると、リリース後不具合数【予測値超過】の箱ひげ図（②、③）の中央値は、ともに低い位置にあるが、箱ひげ図の上辺が高い位置にあるため、レビュー工数密度が低いとは言い切れない。

図表2.2-4　品質実績の評価別の工程別レビュー指摘密度



図表2.2-4に、品質実績の評価別の「レビュー指摘密度」を比較した結果を示す。

アーキテクチャ設計レビューのレビュー指摘密度を比較すると（①を比較）、箱ひげ図の中央値では、リリース後不具合数【予測値超過】は、他と比べて低い位置にあるが、箱ひげ図の上辺と下辺は、リリース後不具合数【予測値以内】と変わらない。

詳細設計レビュー（②を比較）やソースコードレビュー（③を比較）を同じ様に比べて見ると、リリース後不具合数【0件】、【予測値以内】、【予測値超過】の間にはっきりとした違いが見られない。

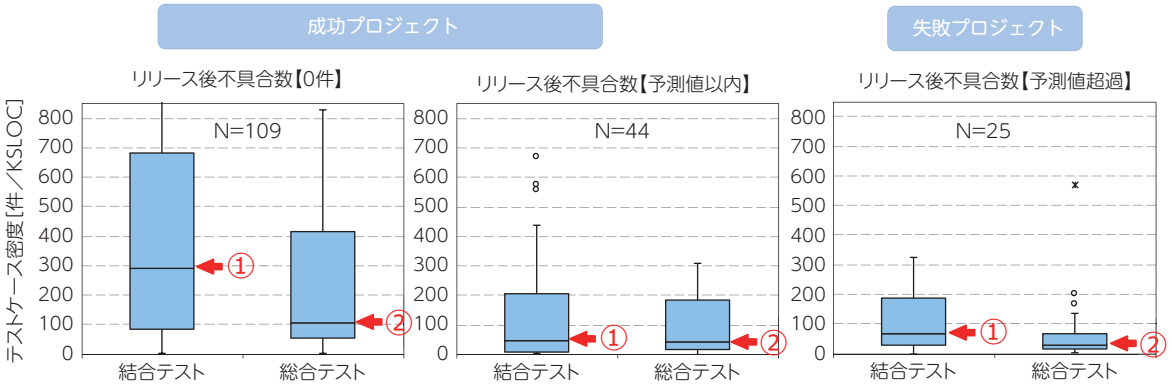
一般に、“レビュー指摘はその内容が重要であり、指摘件数を上げても品質は良くならない”といわれるように、この結果からリリース後の品質とレビュー指摘密度との間に関連は見られない。

(3) テストケース密度とテスト検出バグ密度

●ポイント3●　テストケース密度とテスト検出バグ密度だけでは、テスト作業の良否を判断できない。

組込みシステム開発の品質指標として、結合テストや総合テストのテストケース密度とバグ密度を測り、バグが十分検出されているかを評価する考え方がある。そこで、この考え方が有効かどうかを分析した。図表2.2-5、図表2.2-6は、結合テストと総合テストで実施したテストケースの密度と検出したバグ密度が、リリース後品質の良否に関係しているかどうかを示している。

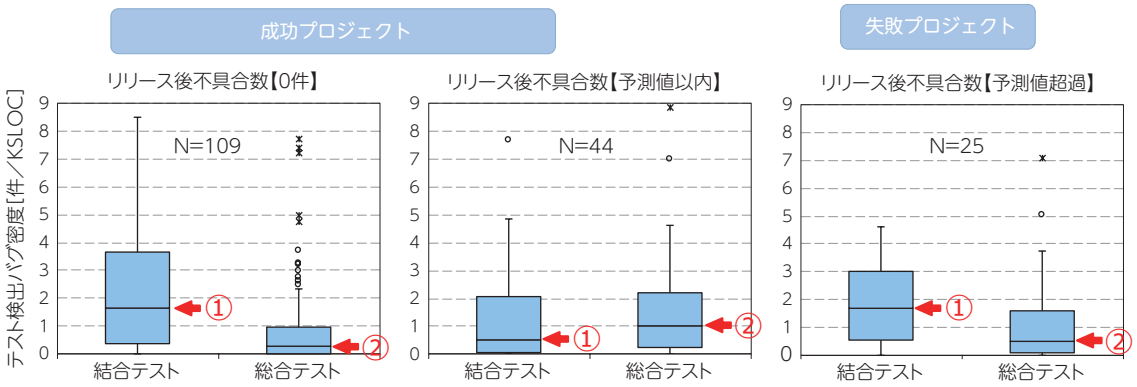
図表2.2-5　品質実績の評価別のSLOC規模あたりのテストケース数



図表2.2-5に、品質実績の評価別の「テストケース密度」を比較した結果を示す。

結合テスト（①）、総合テスト（②）それぞれの場合を比較すると、リリース後不具合数【0件】の場合のみが、【予測値以内】や【予測値超過】よりも、テストケース密度が高いことが分かり、過剰品質の可能性が伺える。そのため、リリース後不具合数【予測値以内】と【予測値超過】の2者で比べてみると、結合テスト（①）、総合テスト（②）ともに中央値に大きな差が無い。つまり、テストケース密度だけでは、リリース後の品質は判断できないと言える。

図表2.2-6　品質実績の評価別のSLOC規模あたりの検出バグ数



図表2.2-6に、品質実績の評価別の「テスト検出バグ密度」を比較した結果を示す。

リリース後不具合数【0件】、【予測値以内】、【予測値超過】のテスト検出バグ密度を、結合テスト（①）、総合テスト（②）で比較したが、プロジェクトの成功や失敗の評価に関係する傾向が掴めない。

結果として、テストケース密度とテスト検出バグ密度だけでは、リリース後品質を測れないことが分かる。

2.2.3 コストと工期視点でのプロジェクトの成功と失敗

●ポイント4●　コストが超過したプロジェクトと予算内に収まったプロジェクトに、作業内容の違いは無い。

コスト計画に対する実績評価の視点で、プロジェクトデータを成功プロジェクトと失敗プロジェクトに分けて、生産性、設計レビュー、テストケース数、テスト検出バグ件数、工数の工程別配分を比較したが、コスト評価が成功か失敗かにかかわらず、両者間で作業傾向に違いは見られなかった。

●ポイント5●　工期が遅延したプロジェクトと納期内に納まったプロジェクトに、作業内容の違いは無い。

工期（納期）に対する実績評価の視点でプロジェクトデータを成功プロジェクトと失敗プロジェクトに分けて、生産性、設計レビュー、テストケース数、テスト検出バグ件数、工数の工程別配分を比較した。その結果、工期評価の成功・失敗の如何を問わず、両者間に作業傾向の違いは見られなかった。

3章　分析について

3.1　分析の進め方 …… 24

3.1.1　分析の観点及び方針
3.1.2　分析の手順

3.2　分析に関する事前の取り決め …… 26

3.2.1　データ項目の取り扱いに関する取り決め
3.2.2　その他の取り決め

3.3　分析結果の取り扱い …… 28

3.3.1　共通事項
3.3.2　基本統計量
3.3.3　回帰分析
3.3.4　箱ひげ図

3章　分析について

この章では、分析の進め方と分析に関する事前の取り決め、分析結果の取り扱いについて説明する。
4章以降では、分析結果を考察して得られた傾向や推測できる要因を“●”、“●”、“●”、…等の印とともに箇条書きで示した。

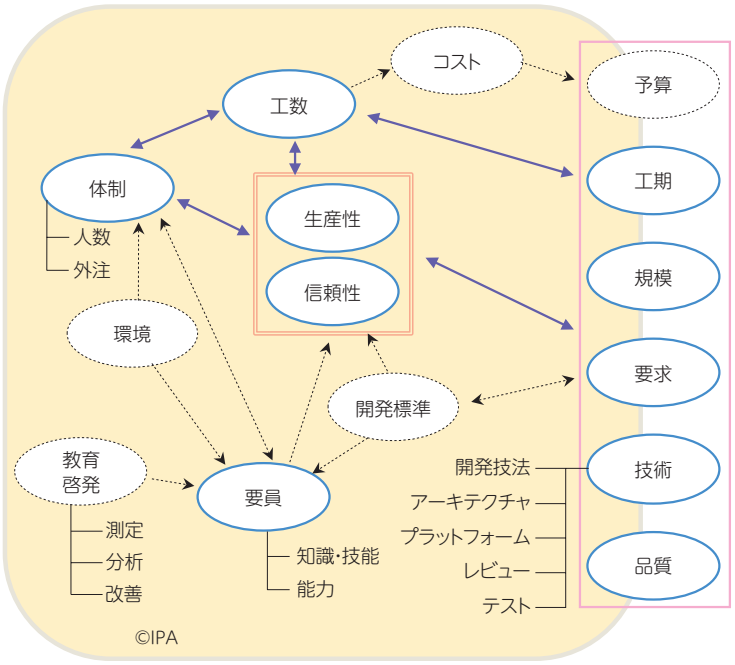
3.1　分析の進め方

本書における分析は「ソフトウェア開発の関係者間で共通認識を形成するための代表的な要素」に着目し、要素間の関係を明らかにするアプローチをとっている。

3.1.1　分析の観点及び方針

ソフトウェア開発プロジェクトの特徴を表す代表的な要素（楕円で示す）と、要素間の主な影響関係（矢印で示す）を図表3.1-1に示す。オレンジ色の二重枠で囲んだ生産性と信頼性は収集データから導出した。開発プロジェクトに影響を与える要素群は、ピンク色の枠で囲んだ。枠内は相互に関連するが、複雑になるので矢印は表示しない。これらの要素群はプロジェクトの成否を左右する要因であり、成功に導くためには、プロジェクトの様々なデータを収集し、各要素の関係を丹念に分析していく必要がある。

図表3.1-1　代表的な要素と、要素間の主な関係



3.1.2 分析の手順

分析の手順を以下に示す。

(1) 収集データの精査

収集データ1件ごとに精査を実施し、分析に必要なデータの不足やデータ間の不整合などを修正する。例えば、プロジェクトの特性を示すデータの不足、データの合計値が合わないなどの不整合が生じた場合は、可能な限りデータ提供元に確認し、適正なデータを入手し直す。

(2) 収集データのプロファイル確認

収集データの各項目について、回答結果の分布傾向を統計グラフや表で視覚的に把握する。選択肢による回答結果の分布は、円グラフまたは棒グラフで把握する。件数や量など度数の分布はヒストグラムで把握する。件数や量などの度数の分布は、必要に応じて正規化する。

(3) 組込み分野全体の傾向分析

収集データ全体に対して分析を行い、工期、工数、規模、生産性や信頼性の傾向について全体像を把握する。

(4) 製品の特性別の傾向分析

組込みソフトウェア開発における生産性や信頼性の傾向は、製品の特性によって左右されると考えられるため、製品の特性で層別し、細分化した分析を行うことにより、データ傾向を明らかにする。

(5) 品質評価別の傾向分析

品質に焦点を当て、成功したプロジェクトと失敗したプロジェクトを比較すると、作業内容の傾向が異なると考えられるため、品質実績の評価で層別した分析を行う。

3.2 分析に関する事前の取り決め

この節では、分析時のデータ項目の取り扱いなどに関する事前の取り決めを示す。

3.2.1 データ項目の取り扱いに関する取り決め

分析の前提として、データ項目ごとに必要な取り決めを以下に示す。データ項目、導出項目に関する詳細な定義は、付録A.3を参照していただきたい。

◆SLOC規模

- コード行数の単位で表す規模は「SLOC規模」と呼ぶ。SLOC はSource Lines of Code の略である。
- コメント行及び空行を含まないコード行数（SLOC）を使用する。
- 提出された規模にコメント行または空行を含む数値データは、コメント行及び空行の比率（提出された値）をもとにして計算した行数を、提出値（コメント行または空行を含む）から引いて算出した行数とする。
- 1,000行の単位で表すものを「KSLOC」と表記する。
- 複数言語の場合でも、合計値のコード行数を使用する。また、プログラミング言語は、言語名が明確に記載されているデータを使用する。特に明記しない限りは、プログラミング言語の種類は混在している。
- SLOC規模をプログラミング言語の種類で層別する場合は、データ項目の「主開発言語1」を用いて、「言語C」「言語C++」「言語C/C++以外」に分類する。
- SLOC規模を対象とした分析では、データに含まれる開発言語の種類を前提条件として記載する。
- 本書では、「新規開発」「改良（派生）開発」の規模の計測対象範囲は次の通りとする。
  - 新規開発では、システム全体を規模の対象範囲とする。
  - 改良（派生）開発においては、改修部分（追加・変更・削除）のみを規模の対象範囲とし、母体は対象範囲外とする。
  - 改良（派生）開発の規模は、変更・追加部分を計測対象範囲として規模を計算して扱っているため、システム全体規模ではない。したがって、改良（派生）開発の規模として示す数値が小さい傾向があっても、単純にシステム全体が小規模であることを意味するものではないことに注意していただきたい。
- 4章以降の分析の際にSLOC規模を扱う場合、原則、導出指標（付録A.3）の実効SLOC実績値を用いる。

◆工数

- 工数は、「社内工数」及び「外部委託工数」の合計値を使用する。社内工数には、「開発」「管理」「その他」及び「作業配分不可」のすべての工数を含める。
- 人月換算は、工数単位が人月の場合は、提出された変換係数を使用した値である。
- 開発5工程（アーキテクチャ設計から総合テストまでの工程）の作業がすべて行われているプロジェクトでは、「該当する5工程」の工数の合計値を使用する。
- 生産性を分析する場合には、開発5工程の工数を使用する。
- 分析の目的に応じて、プロジェクト全体の工程の工数を分析対象とする場合がある。
- 4章以降の分析の際に工数を扱う場合、導出指標（付録A.3）の実績工数（開発5工程など）を用いる。

◆工期

- プロジェクト全体の開始日から終了日までの日数を月数に直したもの。
- 分析の目的に応じて、アーキテクチャ設計開始から総合テスト終了までの開発5工程の期間を分析対象とする場合もある。

◆月あたりの要員数

- 付録A.3 に「月あたりの要員数」として示す導出指標であり、次の計算式で算出される。

  工数が人月で与えられている場合：

  1ヵ月あたりの要員数＝実績工数（プロジェクト全体）÷実績月数（プロジェクト全体）

  工数が人時で与えられている場合：

  1ヵ月あたりの要員数＝実績工数（プロジェクト全体）÷実績月数（プロジェクト全体）÷人時換算係数

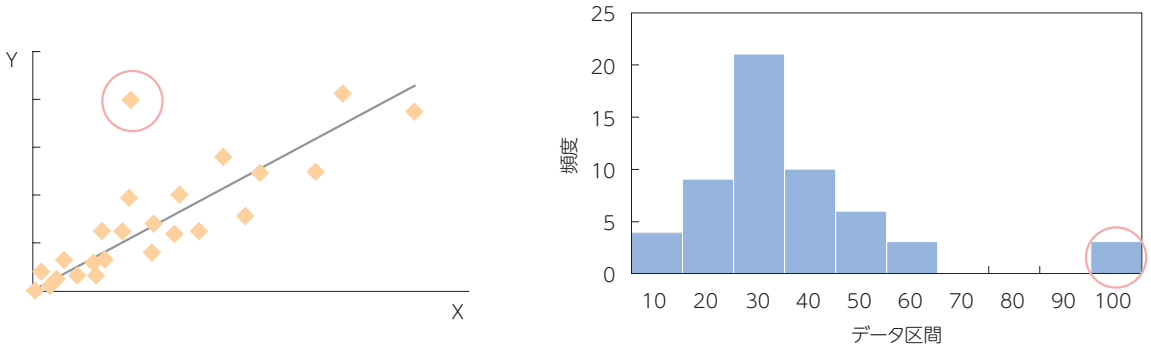
- 人時換算に関しては、上記の工数と同様である。

3.2.2 その他の取り決め

◆外れ値の取り扱い

平均や分布から外れているデータを、分析の対象から恣意的に除外することはしない。分析の対象となるデータは、「外れ値を除外する理由を明らかにする」というプロセスを経て開示する。分析結果のグラフや表において、分析の対象となるデータ数は「N」として示す。

図表3.2-1 外れ値の例



3.3 分析結果の取り扱い

この節では、分析結果の取り扱いとして、掲載基準や評価の目安、基本的な見方などを示す。

なお、分析結果から導出された数値については、「数字の一人歩き」が起こらないよう、読者の方々には、分析結果とプロファイル情報（対象データ、特性などの情報）を一対のものとして扱うなどの配慮をお願いしたい。

3.3.1 共通事項

◆分析結果の掲載基準

- 分析対象の標本数が10件以上であること。
- ただし、複数の層別の分析結果を並べて示す場合、いずれかの層別の標本数が10件以上であれば掲載することがある。
- 同様に、基本統計量や箱ひげ図に並べて表示する場合も、いずれかの標本数が10件以上であれば掲載することがある。
- 分析対象の標本が特定の企業のデータに偏らないこと。
- 基準を満たしていなくても、目的によって掲載することがある。その場合、該当箇所にその旨を記載する。

◆単位の表記

グラフや図表での単位の表記は、次に示す通りとする。

図表3.3-1　単位の表記

データ	単位の基本的な表記
SLOC規模	[SLOC]
1,000SLOC単位のSLOC規模	[KSLOC]
人時単位の工数	[人時]
人月単位の工数	[人月]

データ	単位の基本的な表記
月数単位の工期	[月]
テスト検出バグ数	[件]
要員数	[人]

◆分析結果の掲載方式

6章以降の分析結果の掲載方式を以下に示す。

- **使用データの掲載方式**

  分析対象データの抽出条件について、以下の例に示すような方式を採用する。

例：条件1 ～ 3のAND条件で抽出した標本をもとに、データ1とデータ2の関係を分析する場合

■層別定義	■対象データ
• 条件1（←1つめの抽出条件を表す）	• X軸：データ1（←その関係を分析する1つめのデータの名称を表す）
• 条件2（←2つめの抽出条件を表す）	• Y軸：データ2（←その関係を分析する2つめのデータの名称を表す）
• 条件3（←3つめの抽出条件を表す）	

分析対象データが導出指標の場合は、「■対象データ」において「データ1（導出指標）」のように表記する。データの定義は、付録A.2及び付録A.3を参照していただきたい。

- **導出指標の例**
  - SLOC生産性
  - テスト検出バグ密度
  - 月あたりの要員数
  - 外部委託比率

- **分析結果の表現方式**
  - 「基本統計量」：統計量（数値）でデータの傾向を示す。
  - 「散布図」：データの散らばり具合や傾向を示す。
  - 「箱ひげ図」：中央値、25パーセンタイルと75パーセンタイルで分布の傾向を視覚的に示す。
  - 分析対象数を「N」で示す。拡大分布図でも非表示を含めた分析対象数を示す。ただし、範囲限定ヒストグラムは表示数を示す。

3.3.2 基本統計量

◆基本統計量の掲載基準

対象となっている標本数が10件以上であること。ただし、複数の層別のデータを併記する際に、いずれかの層別のデータが10件以上である場合は記載する。

◆基本統計量の表記

図表3.3-2に示すいずれかの形式で、対象とするデータについての「基本統計量」を掲載する。

「項目」はデータ名称を表し、「N」は件数、「最小」は最小値、「P25」は25パーセンタイル、「中央」は中央値、「P75」は75パーセンタイル、「最大」は最大値、「平均」は平均値、「標準偏差」は標準偏差を示す。

「項目」のデータ名称は付録Aのデータ項目定義に従うが、表記は「項番_名前」または「名前」とする。例えば「1-5_開発プロジェクトの種別」あるいは「開発プロジェクトの種別」のように表記する。なお、用語の詳細は付録Bを参照していただきたい。

図表3.3-2 基本統計量の表

項目	N	最小	P25	中央	P75	最大	平均	標準偏差

N	最小	P25	中央	P75	最大	平均	標準偏差

項目	N	P25	中央	P75

N	P25	中央	P75

◆基本統計量の評価の目安

評価の目安として、図表3.3-3を使用する。

図表3.3-3 基本統計量を使用した場合の判断の目安

	項目	判断の目安
1	データ数Nの量	データ数は層別あたり、最低でもN≧10、望ましいのはN≧30とする
2	統計量の代表値の採択	分布は非対称性が大きいと見られるため、平均値より中央値を採択する

本書では、散布図や箱ひげ図など視覚的に傾向を捉える図表とともに、基本統計量も合わせて記載している。これにより的確なデータ値を把握することができる。また、1つの項目だけではなく、いくつかの層別された項目に対して表すことで、傾向を捉えることができる。

例）工程別の実績工数の比率の基本統計量（改良（派生）開発）

比率が高い工程には「それだけ長い作業時間を要する」ということになる。

［比率］

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	228	0.01	0.09	0.16	0.23	0.68	0.17	0.11
詳細設計	228	0.01	0.10	0.15	0.21	0.67	0.16	0.09
実装・単体テスト	228	0.03	0.17	0.23	0.33	0.79	0.26	0.12
結合テスト	228	0.01	0.16	0.20	0.27	0.58	0.22	0.10
総合テスト	228	0.00	0.09	0.16	0.26	0.61	0.18	0.13

3.3.3 回帰分析

◆回帰分析結果の掲載について

本書では回帰分析結果は掲載しないが、2つのデータ項目間に相関が見られる場合に相関係数Rを掲載する。

図表3.3-4　相関関係を確認した場合の判断の目安

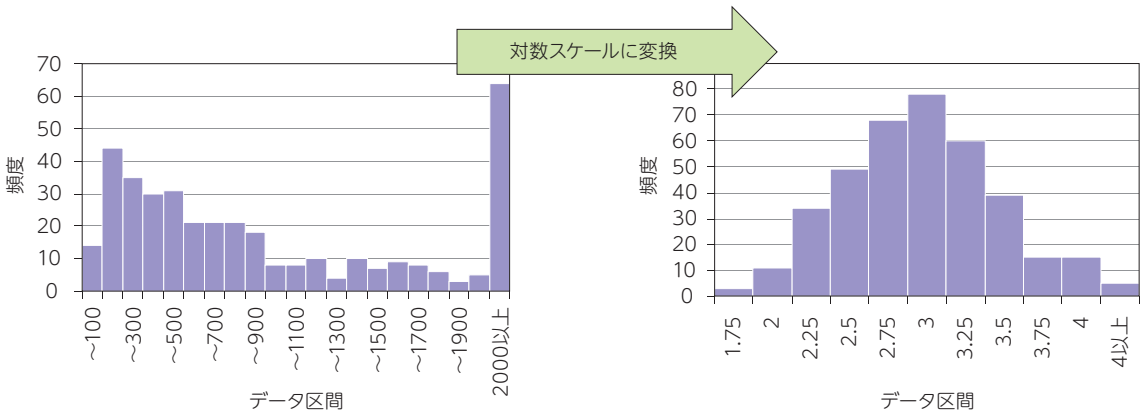
	項目	判断の目安
1	データ数Nの量	データ数は層別あたり、N≧30とする
2	相関の見方	｜相関係数R｜≧0.85：強い関係 0.85＞｜相関係数R｜≧0.70：やや強い関係 ｜相関係数R｜＜0.70：強い関係は認められないが要継続観察
3	相関の有意性	P値＜0.05とする（危険率5%で相関が有意と判断できる）

◆対数スケールで見る理由

ソフトウェア開発プロジェクトのデータは正規分布していないことが多い。しかし、対数に変換するとほぼ正規分布と見なせることが多い。よって、対数スケールに変換すると「正規分布」であることを前提としている相関係数の有意性を求めることができる。本書では、散布図を掲載する場合、必要に応じて、対数スケール表示を並記している。対数スケール表示に相関が見られる場合は、相関係数Rを掲載する。

※対数スケールで見る理由の詳細は古山恒夫「プロジェクトデータ分析の指針と分析事例」,『SEC journal No.3』, p.6-13, 2005に記載されている。

図表3.3-5　通常スケールを対数変換したときの分布



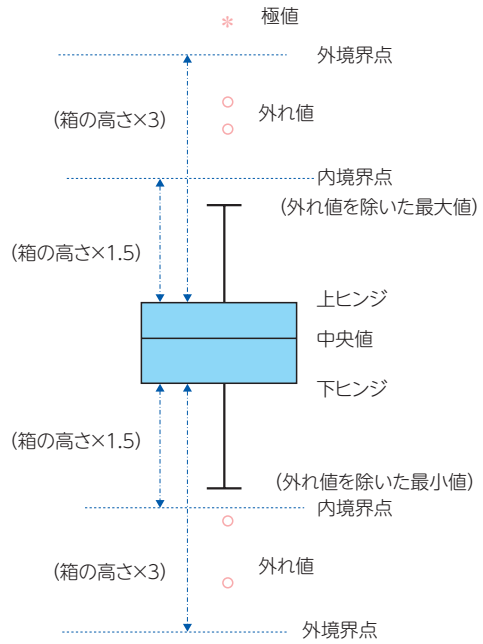
3.3.4 箱ひげ図

◆箱ひげ図の表記

箱ひげ図は、中央値だけでなくバラツキも比較できるため、傾向を視覚的に捉えたい場合に有効である。図表3.3-6に示す通り、箱ひげ図は、「箱」とそれに付随した「ひげ」から構成される。

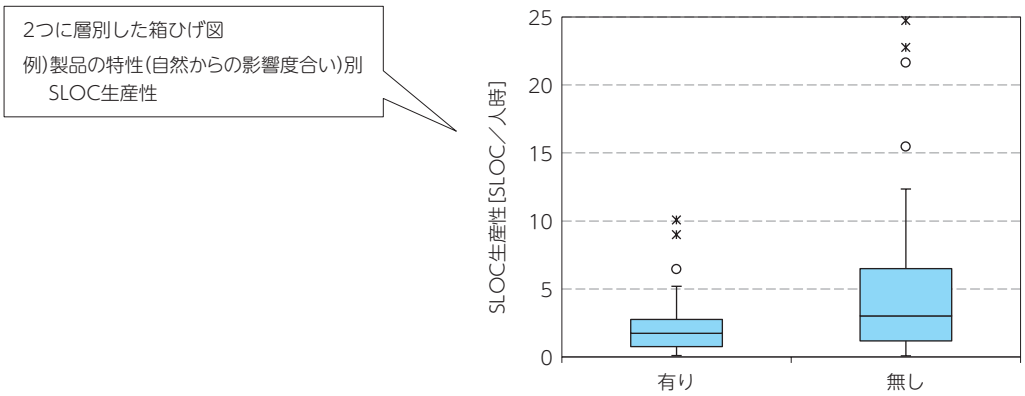
箱の上端は上ヒンジと呼ばれ、上から全体の25%に相当するデータの位置（値）を示す。箱の下端は下ヒンジと呼ばれ、下から25%に相当するデータの位置（値）を示す。上下50%の境目は中央値であり、箱の中のその位置に横線を引いて示す。外れ値を除いた最大値と最小値までを、ひげとして表す。

図表3.3-6　箱ひげ図のサンプル



◆箱ひげ図の使い方

箱ひげ図は、いくつかの層別されたデータを比較する場合に用いると、傾向の違いが視覚的に分かる。



例）2つに層別した箱ひげ図を見比べることで、明らかにデータの傾向が異なる点を視覚的に理解できる。

4章　収集データのプロファイル

4.1　プロファイル一覧 …… 34
4.2　開発プロジェクトの全般的な特徴 …… 37
4.3　利用局面 …… 39
4.4　システム特性 …… 40
4.5　開発の進め方 …… 44
4.6　ユーザ要求管理 …… 45
4.7　要員の経験 …… 46
4.8　規模 …… 47
4.9　工期 …… 48
4.10　工数 …… 50
4.11　体制 …… 52
4.12　信頼性 …… 53
4.13　実施工程の組み合わせパターン …… 58
4.14　プロジェクト成否 …… 59

4章　収集データのプロファイル

本章には、IPAで収集したプロジェクトデータのプロファイル情報（付録A「データ収集項目」）を掲載する。

4.1　プロファイル一覧

プロファイル				収集データ項目
4.2　開発プロジェクトの全般的な特徴	開発プロジェクトの種別		図表4.2-1	1-5_開発プロジェクトの種別
	開発システムの経過年数・保守年数		図表4.2-2 図表4.2-3	1-5a_経過年数 1-5b_保守する年数
	開発プロジェクトの概要		図表4.2-4	1-7_開発プロジェクトの概要
	新技術を利用する開発か否か		図表4.2-5	1-11_新技術を利用する開発か否か
4.3　利用局面	利用形態		図表4.3-1	2-1_利用形態
	障害リスク		図表4.3-2	2-2_障害リスク
4.4　システム特性	製品の特性	リアルタイム性（時間制約）	図表4.4-1	3-1-1_製品の特性_リアルタイム性（時間制約）
		自然環境からの影響度合い	図表4.4-2	3-1-2_製品の特性_自然環境からの影響度合い
		ユーザの多様性	図表4.4-3	3-1-3_製品の特性_ユーザの多様性
		法規等による規制度合い	図表4.4-4	3-1-4_製品の特性_法規等による規制度合い
		M２Mの有無	図表4.4-5	3-1-5_製品の特性_M２Mの有無
		ネットワーク接続の有無	図表4.4-6	3-1-6_製品の特性_ネットワーク接続の有無
		稼動（非停止、オンデマンド）	図表4.4-7	3-1-7_製品の特性_稼動（非停止、オンデマンド）
		オンライン保守の可否	図表4.4-8	3-1-8_製品の特性_オンライン保守の可否
	市販パッケージ利用の有無		図表4.4-9	3-2_（市販）パッケージ利用の有無
	CPUアーキテクチャ		図表4.4-10	3-7_CPUアーキテクチャ
	OSアーキテクチャ		図表4.4-11	3-7a_OSアーキテクチャ
	開発対象製品のソフトウェアプラットフォーム		図表4.4-12	3-8_開発対象製品のソフトウェアプラットフォーム
	開発言語		図表4.4-13	3-9_主開発言語
4.5　開発の進め方	開発ライフサイクルモデル		図表4.5-1	4-1_開発ライフサイクルモデル
	類似プロジェクトの参照の有無		図表4.5-2	4-2_類似プロジェクトの参照の有無
	ツールの利用有無		図表4.5-3 図表4.5-4	4-3_プロジェクト管理ツールの利用 4-4_構成管理ツールの利用 4-5_設計支援ツールの利用 4-6_ドキュメント作成ツールの利用 4-7_デバッグツールの利用 4-7a_テストツールの利用 4-7b_バグ管理ツールの利用 4-8_コードジェネレータの利用
4.6　ユーザ要求管理	ユーザ要求		図表4.6-1 図表4.6-2	5-1_要求レベル（信頼性） 5-2_要求レベル（使用性） 5-3_要求レベル（性能・効率性） 5-4_要求レベル（保守性） 5-4a_要求レベル（移植性） 5-4b_要求レベル（セキュリティ）
4.7　要員の経験	要員の経験		図表4.7-1 図表4.7-2	6-1_要員スキル_製品分野の経験 6-2_要員スキル_分析・設計経験 6-3_要員スキル_言語・ツール利用経験 6-4_要員スキル_開発プラットフォームの使用経験 6-5_要員スキル_製品プラットフォームの使用経験
4.8　規模	SLOC（コード行数）実績値		図表4.8-1 図表4.8-2 図表4.8-3	7-5_SLOC実績値
	SLOC（母体含むコード行数）実績値		図表4.8-4 図表4.8-5 図表4.8-6	7-6_SLOC実績値（母体）

プロファイル			収集データ項目
4.9 工期	プロジェクト全体の開発期間	図表4.9-1 図表4.9-2	8-4-3_プロジェクト全体工期（実績）_月数
	開発6工程の開発期間	図表4.9-3 図表4.9-4	8-2-3-1_要求定義_月数 8-2-3-2_アーキテクチャ設計_月数 8-2-3-3_詳細設計_月数 8-2-3-4_実装・単体テスト_月数 8-2-3-5_結合テスト_月数 8-2-3-6_総合テスト_月数
	開発5工程（要求定義を除く）の開発期間	図表4.9-5 図表4.9-6	8-2-3-2_アーキテクチャ設計_月数 8-2-3-3_詳細設計_月数 8-2-3-4_実装・単体テスト_月数 8-2-3-5_結合テスト_月数 8-2-3-6_総合テスト_月数
4.10 工数	プロジェクト全体の工数の実績値（人時換算）	図表4.10-1 図表4.10-2 図表4.10-3	9-4-35_社内実績工数_プロジェクト全体 9-8-7_外部委託工数_プロジェクト全体 9-2_人時換算係数
	プロジェクト全体の工数の実績値（人月換算）	図表4.10-4 図表4.10-5 図表4.10-6	
	人月ー人時換算係数	図表4.10-7 図表4.10-8	9-2_人時換算係数
4.11 体制	外部委託工数比率	図表4.11-1 図表4.11-2	9-8-7_外部委託工数_プロジェクト全体 9-4-35_社内実績工数_プロジェクト全体
	月あたりの要員数	図表4.11-3 図表4.11-4	8-4-3_プロジェクト全体工期（実績）_月数 9-4-35_社内実績工数_プロジェクト全体 9-8-7_外部委託工数_プロジェクト全体 9-2_人時換算係数
4.12 信頼性	結合テスト検出バグ現象密度と原因密度	図表4.12-1 図表4.12-2 図表4.12-3 図表4.12-4 図表4.12-5 図表4.12-6	10-3-1_結合テスト検出バグ現象数 10-3-2_結合テスト検出バグ原因数 7-5_SLOC実績値
	総合テスト検出バグ現象密度と原因密度	図表4.12-7 図表4.12-8 図表4.12-9 図表4.12-10 図表4.12-11 図表4.12-12	10-4-1_総合テスト検出バグ現象数 10-4-2_総合テスト検出バグ原因数 7-5_SLOC実績値
	総合テスト検出バグ現象密度と原因密度（母体含む）	図表4.12-13 図表4.12-14 図表4.12-15 図表4.12-16 図表4.12-17 図表4.12-18	10-4-1_総合テスト検出バグ現象数 10-4-2_総合テスト検出バグ原因数 7-6_SLOC実績値（母体）
	品質保証の体制	図表4.12-19	10-5_品質保証体制
	品質基準、レビューの有無	図表4.12-20 図表4.12-21	10-7_定量的な出荷品質基準の有無 10-8_第三者レビューの有無
	テスト計画書、レビューの有無	図表4.12-22 図表4.12-23	4-17_テスト計画書の有無 4-18_テスト計画書のレビューの有無

プロファイル			収集データ項目
4.13　実施工程の組み合わせパターン	実施工程の組み合わせパターン	図表4.13-1 図表4.13-2	9-3-1_プロジェクト総工数に含まれるフェーズ_要求定義 9-3-2_プロジェクト総工数に含まれるフェーズ_アーキテクチャ設計 9-3-3_プロジェクト総工数に含まれるフェーズ_詳細設計 9-3-4_プロジェクト総工数に含まれるフェーズ_実装・単体テスト 9-3-5_プロジェクト総工数に含まれるフェーズ_結合テスト 9-3-6_プロジェクト総工数に含まれるフェーズ_総合テスト
4.14　プロジェクト成否	実績の評価（QCD）	図表4.14-1 図表4.14-2 図表4.14-3 図表4.14-4 図表4.14-5 図表4.14-6 図表4.14-7	1-15_計画の評価（コスト） 1-16_計画の評価（品質） 1-17_計画の評価（工期） 1-18_実績の評価（コスト） 1-19_実績の評価（品質） 1-20_実績の評価（工期）

4.2 開発プロジェクトの全般的な特徴

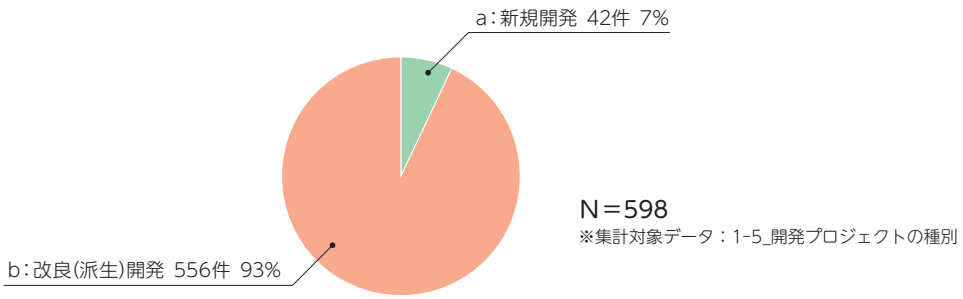
この節では、開発プロジェクトの基本的な属性を示す。

- 開発プロジェクトの種別
- 開発システムの経過年数・保守年数
- 開発プロジェクトの概要
- 新技術を利用する開発か否か

(1)開発プロジェクトの種別

収集したプロジェクトデータが、新規開発プロジェクトのものか改良開発や派生開発のものなのかを示す。

図表4.2-1 開発プロジェクトの種別



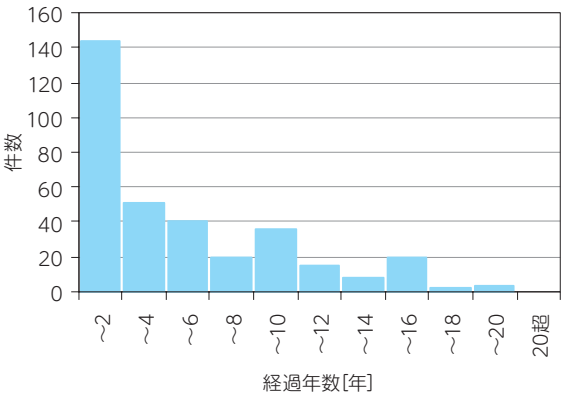
N=598

※集計対象データ:1-5_開発プロジェクトの種別

(2)開発システムの経過年数・保守年数

経過年数は、改良(派生)開発の場合に、これまで保守または維持管理されていた期間(年数)を示す。保守年数は、プロジェクト終了後に保守または維持管理しなければならない年数(製品寿命)を示す。

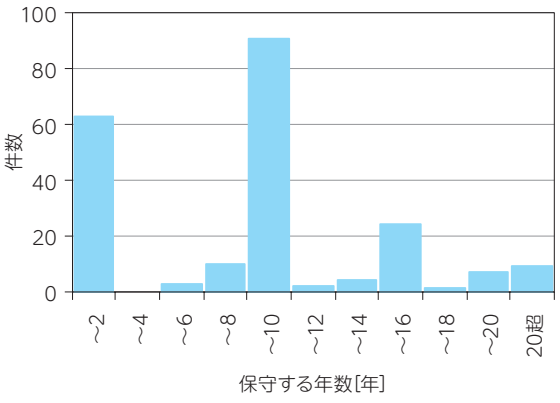
図表4.2-2 開発システムの経過年数



N=339

※集計対象データ:1-5a_経過年数

図表4.2-3 開発システムの保守年数



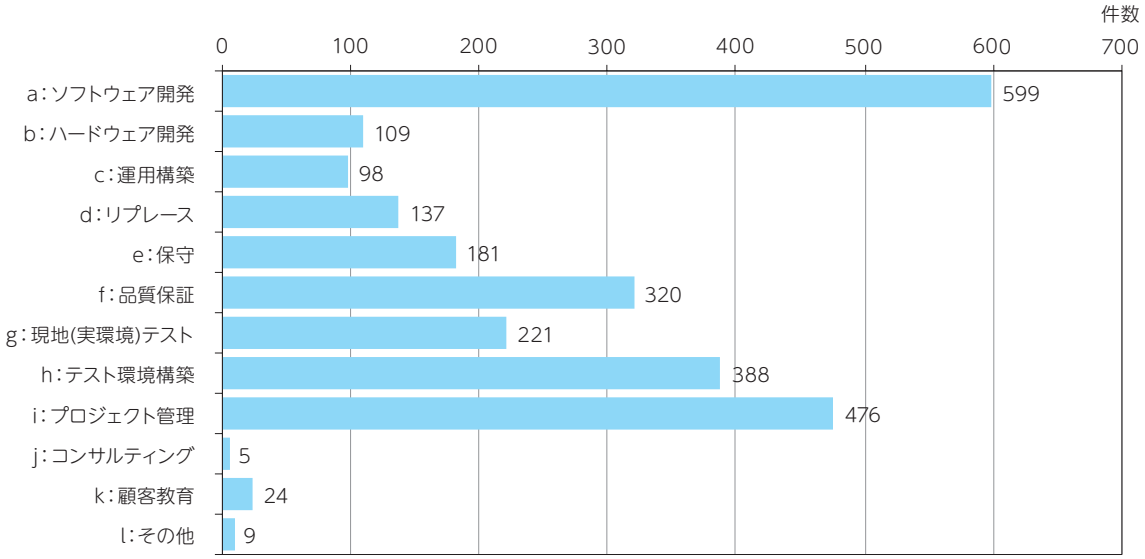
N=214

※集計対象データ:1-5b_保守する年数

(3)開発プロジェクトの概要

収集したプロジェクトデータのプロジェクトに含まれるすべての作業内容集計結果を示す。

図表4.2-4　開発プロジェクトの概要



N＝599

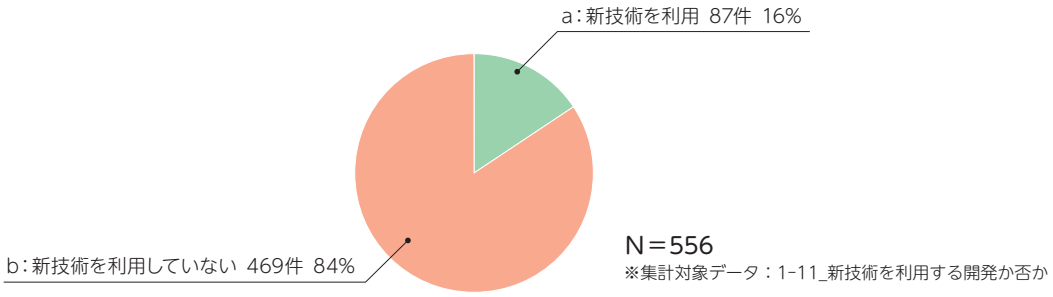
※集計対象データ：1-7_開発プロジェクトの概要

- 組込み開発のプロジェクト形態の特徴の1つとして、プロジェクト内にハードウェア開発とソフトウェア開発が同居している場合がうかがえる。
- 品質保証やプロジェクト管理が行われていない場合があるが、研究開発プロジェクトなど、試作を目的とした場合と考えられる。

(4)新技術を利用する開発か否か

収集したプロジェクトデータのうち、新技術を利用した開発プロジェクトがどれだけあるかを示す。新技術とは、プロジェクトメンバにとって経験のない技術を利用している場合である。

図表4.2-5　新技術を利用する開発か否か



N＝556

※集計対象データ：1-11_新技術を利用する開発か否か

4.3 利用局面

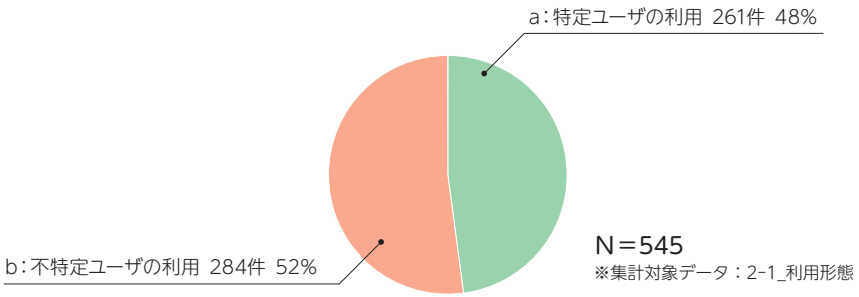
この節では、利用局面に関する以下の属性を示す。

- 利用形態
- 障害リスク

(1)利用形態

開発対象の製品やシステムの利用者が、特定ユーザか不特定ユーザかを示す。

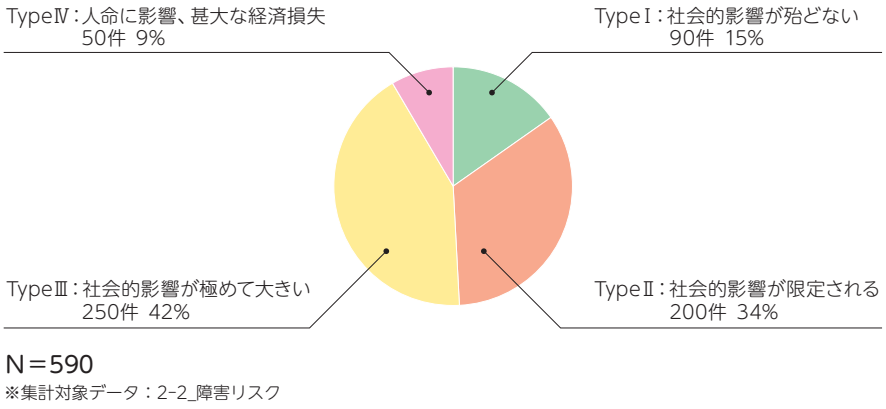
図表4.3-1　利用形態



(2)障害リスク

ソフトウェア不具合に起因した障害が発生した場合の影響度合い(人的損害・経済損失など)を把握し、開発対象の製品やシステムをTypeⅠ～TypeⅣの4種類に分類している。

図表4.3-2　障害リスク



4.4 システム特性

この節では、開発した組込みシステムの特性を示す以下のプロファイルを掲載する。

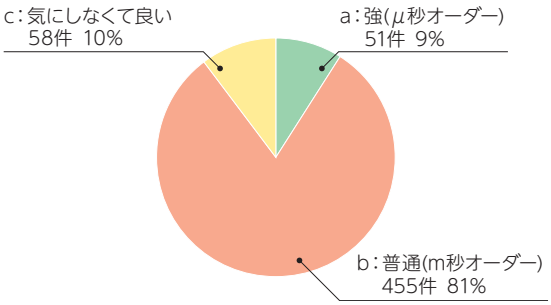
- 製品の特性
- 市販パッケージ利用の有無
- CPUアーキテクチャ
- OSアーキテクチャ
- 開発対象製品のソフトウェアプラットフォーム
- 開発言語

(1)製品の特性

リアルタイム性(時間制約)	：強(μ秒オーダー)／普通(m秒オーダー)／気にしなくて良い
自然環境からの影響度合い	：気象、地形、位置(緯度・経度・高度)等の影響の有無
ユーザの多様性	：年齢・性別・人種・地域等の不特定多数のユーザの多少
法規等による規制度合い	：米国FDA規制、機能安全規格等のソフトウェア品質に関わる規格の有無
M2Mの有無	：M2M通信機能の有無
ネットワーク接続の有無	：他の装置やシステムとネットワークを介して接続する機能の有無
稼動(非停止、オンデマンド)	：非停止システムか、又はオンデマンドで使用されるものかどうか
オンライン保守の可否	：プログラム更新をオンライン保守機能により実施可能かどうか

これらの特性は、収集データを分析する場合の層別・分類項目に用いる。

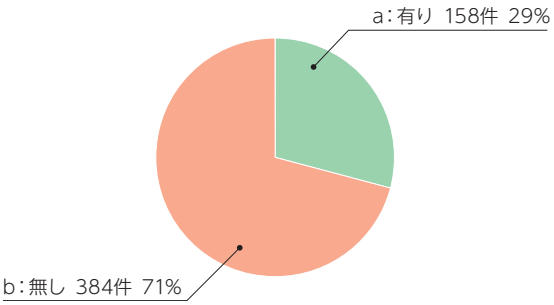
図表4.4-1　製品の特性: リアルタイム性(時間制約)



N＝564

※集計対象データ：3-1-1_リアルタイム性(時間制約)

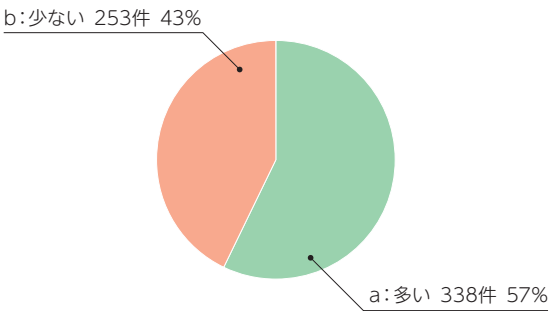
図表4.4-2　製品の特性: 自然環境からの影響度合い



N＝542

※集計対象データ：3-1-2_自然環境からの影響度合い

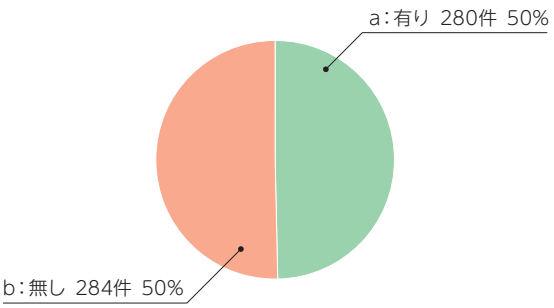
図表4.4-3　製品の特性: ユーザの多様性



N＝591

※集計対象データ：3-1-3_ユーザの多様性

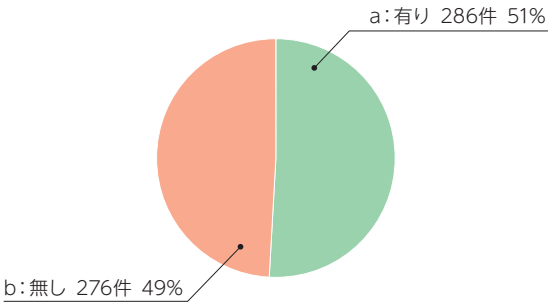
図表4.4-4　製品の特性: 法規等による規制度合い



N＝564

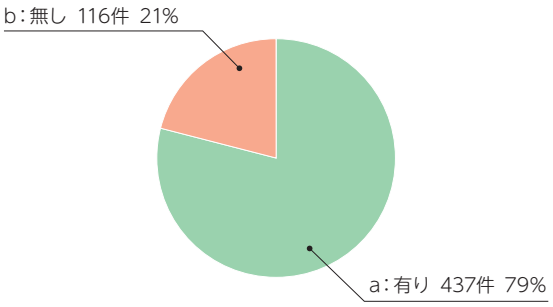
※集計対象データ：3-1-4_法規等による規制度合い

図表4.4-5　製品の特性: M2Mの有無



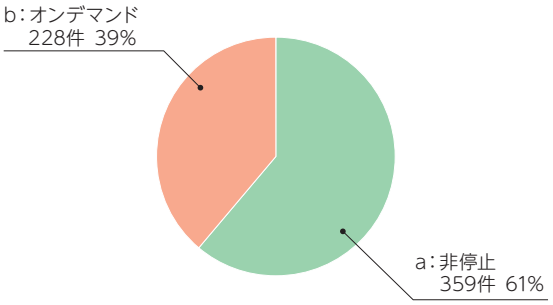
N＝562
※集計対象データ：3-1-5_M2Mの有無

図表4.4-6　製品の特性: ネットワーク接続の有無



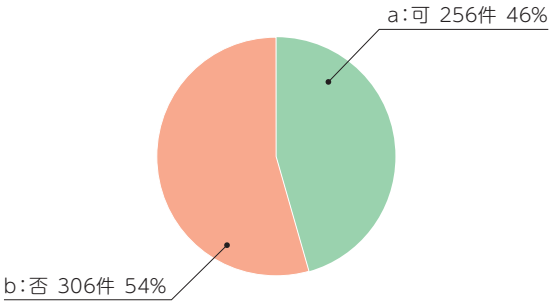
N＝553
※集計対象データ：3-1-6_ネットワーク接続の有無

図表4.4-7　製品の特性: 稼動（非停止、オンデマンド）



N＝587
※集計対象データ：3-1-7_稼動（非停止、オンデマンド）

図表4.4-8　製品の特性: オンライン保守の可否

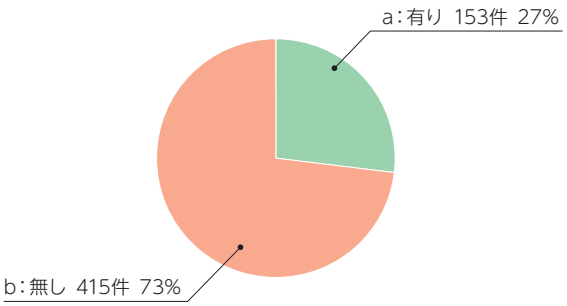


N＝562
※集計対象データ：3-1-8_オンライン保守の可否

(2) 市販パッケージ利用の有無

パッケージソフト（市販・オープンソフト）の利用の有無を示す。ただし、自社開発したパッケージソフトは除く。

図表4.4-9　市販パッケージ利用の有無

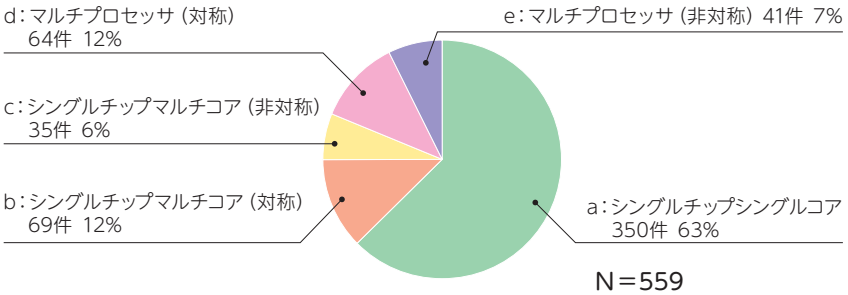


N＝568
※集計対象データ：3-2_（市販）パッケージ利用の有無

(3) CPUアーキテクチャ

開発対象の製品やシステムに組み込まれているCPUが、シングルチップかマルチチップ(マルチプロセッサ)か、シングルチップの場合は、コアが1つ(シングルコア)か複数(マルチコア)か、また、マルチプロセッサ、マルチコアの場合は、対称型か非対称型を示す。

図表4.4-10　CPUアーキテクチャ



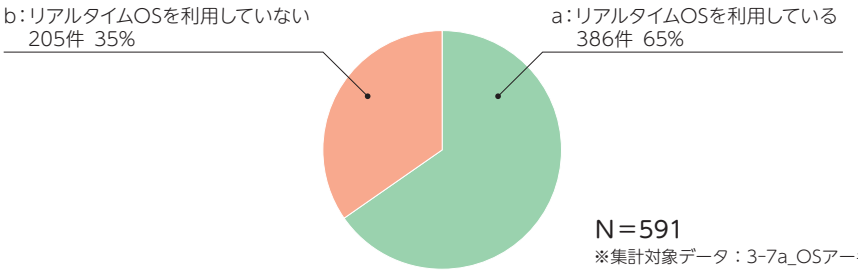
N=559

※集計対象データ:3-7_CPUアーキテクチャ

(4) OSアーキテクチャ

開発対象の製品やシステムに組み込まれているOSが、リアルタイムOSか非リアルタイムOSかを示す。

図表4.4-11　OSアーキテクチャ



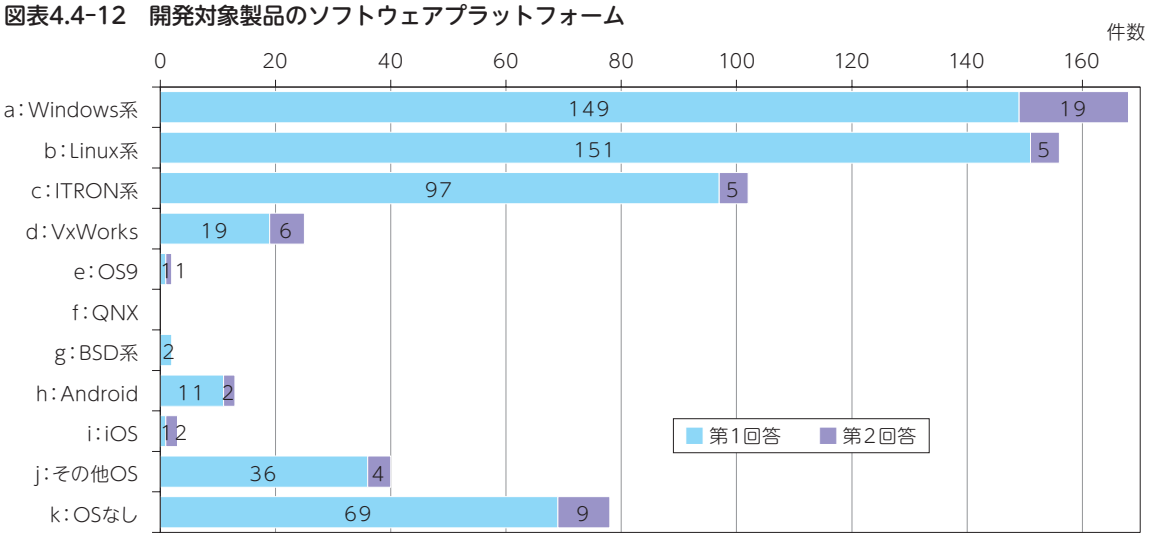
N=591

※集計対象データ:3-7a_OSアーキテクチャ

(5) 開発対象製品のソフトウェアプラットフォーム

開発対象の製品やシステムに組み込まれているOSの種類を示す。

図表4.4-12　開発対象製品のソフトウェアプラットフォーム

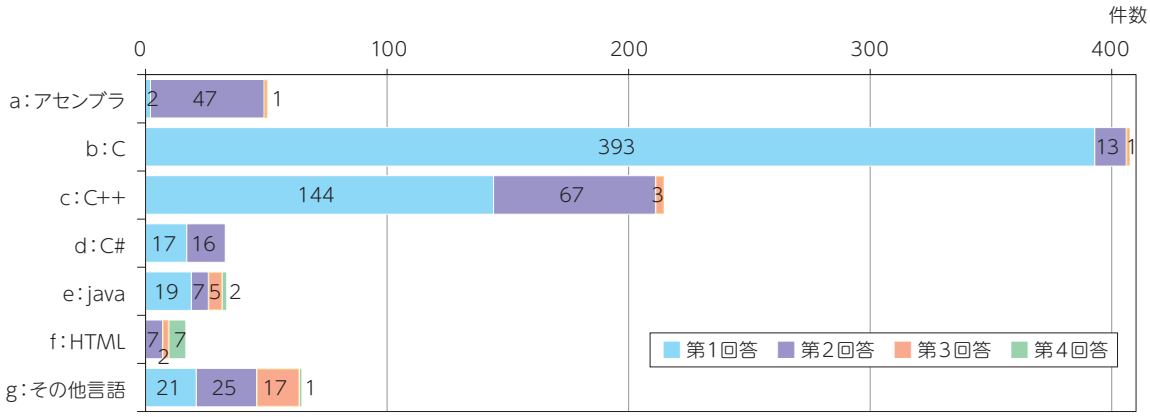


N=536

※集計対象データ:3-8_主たる開発対象OS

(6)開発言語

図表4.4-13　開発言語



N＝596

※集計対象データ：3-9_主開発言語

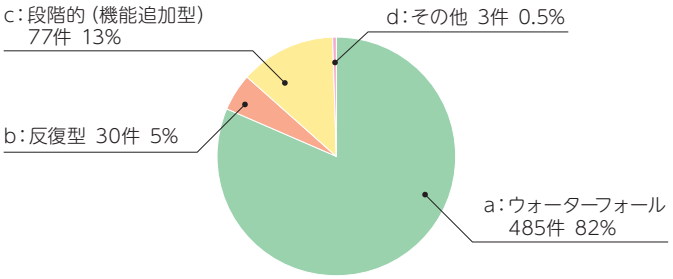
4.5 開発の進め方

この節では、開発プロジェクトにおける開発作業の進め方に関する、以下のプロファイルを掲載する。

- 開発ライフサイクルモデル
- 類似プロジェクトの参照の有無
- ツールの利用有無

(1)開発ライフサイクルモデル

図表4.5-1　開発ライフサイクルモデル

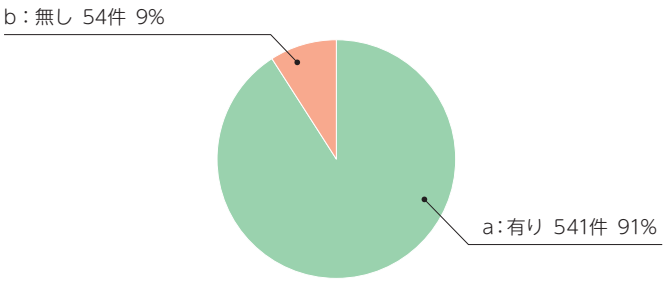


N＝595
※集計対象データ：4-1_開発ライフサイクルモデル

(2)類似プロジェクトの参照の有無

類似プロジェクトのソフトウェア資産(プログラム、各工程で作成された文書、プロジェクト管理資料、テストデータなど、開発プロジェクトによって作成されたものすべてを含む)のいずれかを参照したか否かを示す。

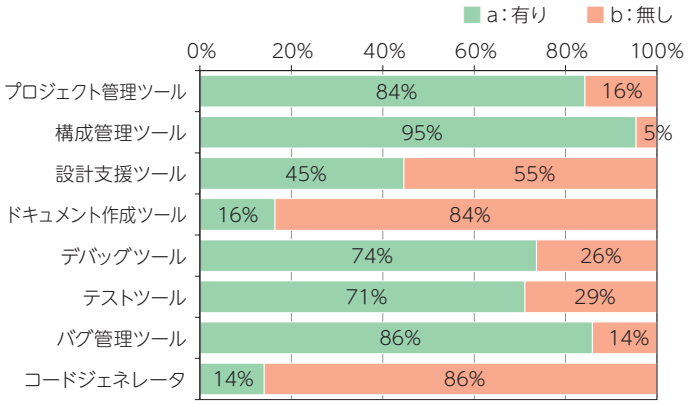
図表4.5-2　類似プロジェクトの参照の有無



N＝595
※集計対象データ：4-2_類似プロジェクトの参照の有無

(3)ツールの利用有無

図表4.5-3　ツールの利用有無



図表4.5-4　ツールの利用有無一覧

集計対象データ		a:有り	b:無し	N
4-3	プロジェクト管理ツール	496	93	589
4-4	構成管理ツール	563	27	590
4-5	設計支援ツール	257	319	576
4-6	ドキュメント作成ツール	93	474	567
4-7	デバッグツール	394	141	535
4-7a	テストツール	384	156	540
4-7b	バグ管理ツール	431	71	502
4-8	コードジェネレータ	79	484	563

- プロジェクト管理ツールは、独自に作成したツールやチケット駆動型のTracやRedmineなどが使われている。
- 構成管理ツールは、収集したプロジェクトの中ではSubversionが最も多く使われている。
- バグ管理ツールは、TracやRedmineが多く使われている。

4.6 ユーザ要求管理

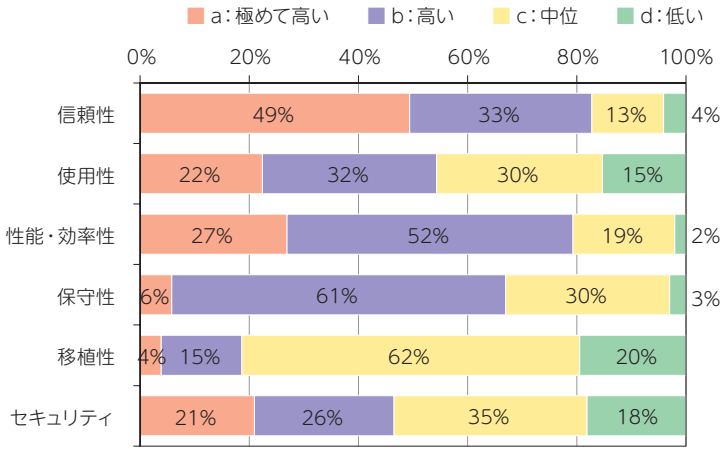
この節では、製品品質に対するユーザ要求のレベルに関する、プロファイルを掲載する。

(1) ユーザ要求

製品品質に対するユーザ要求事項は、信頼性、使用性、性能･効率性、保守性、移植性、セキュリティの6つの特性で示す。

信頼性　：システムの故障の頻度、故障状態からの回復時間・影響を受けたデータの修復などに関する、要求の厳しさ。
使用性　：利用者にとって製品仕様が理解しやすいか、適用法を習得しやすいか、管理しやすいか、またグラフィカル・デザインなどが魅力的であるかなど。
性能・効率性　：製品を利用する際の応答時間・処理時間・処理能力、及びディスク・メモリのハードウェア・その他の資源の使用量などに関する、要求の厳しさ。
保守性　：製品の修理に関して、故障箇所・原因の特定のしやすさ、変更作業のしやすさ、修正の際の予期せぬ影響の防止、修正の妥当性の確認のしやすさなどに関する、要求の厳しさ。
移植性　：移植性に関する、要求の厳しさ。
セキュリティ　：セキュリティに関する、要求の厳しさ。

図表4.6-1　要求レベル



図表4.6-2　要求レベル一覧

集計対象データ	a：極めて高い	b：高い	c：中位	d：低い	N
5-1　信頼性	286	193	76	24	579
5-2　使用性	130	185	176	89	580
5-3　性能・効率性	156	304	108	12	580
5-4　保守性	33	348	171	17	569
5-4a　移植性	22	84	352	111	569
5-4b　セキュリティ	119	145	201	103	568

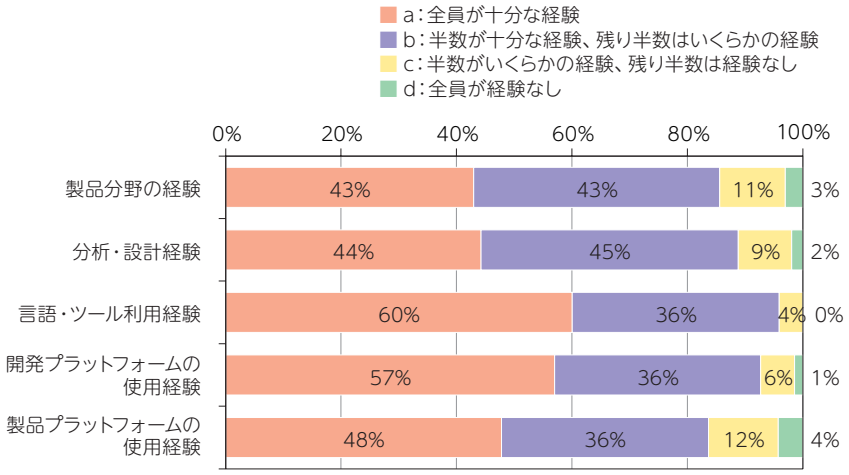
4.7 要員の経験

この節では、開発プロジェクトに携わる要員の経験に関する、プロファイルを掲載する。

(1)要員の経験

要員の経験を製品分野の経験、分析・設計の経験、言語・ツール利用経験など5つのカテゴリで示す。製品プラットフォームは、開発対象製品のプラットフォームで搭載OSやミドルウェアなどを意図し、開発プラットフォームは、クロス開発環境を意図する。

図表4.7-1　要員の経験



図表4.7-2　要員の経験一覧

	← 良い			悪い →	
集計対象データ	a	b	c	d	N
6-1　製品分野の経験	242	240	64	17	563
6-2　分析・設計経験	249	251	52	11	563
6-3　言語・ツール利用経験	338	202	23	0	563
6-4　開発プラットフォームの使用経験	321	201	33	8	563
6-5　製品プラットフォームの使用経験	269	202	68	24	563

※選択肢a、b、c、dの内容
a：全員が十分な経験、b：半数が十分な経験、残り半数はいくらかの経験、c：半数がいくらかの経験、残り半数は経験なし、d：全員が経験なし

4.8 規模

この節では、開発したソフトウェアの規模に関する、以下のプロファイルを掲載する。

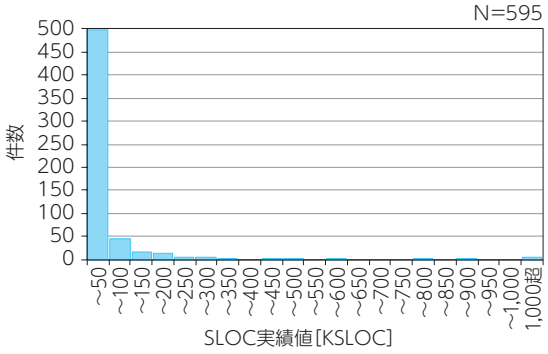
- SLOC（コード行数）実績値
- SLOC（母体含むコード行数）実績値

(1) SLOC（コード行数）実績値

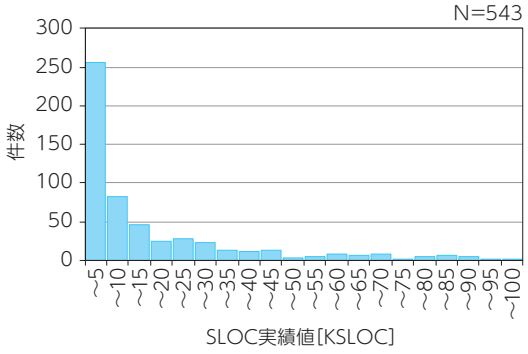
収集したプロジェクトのソフトウェア規模（SLOC実績値）の分布を示す。

改良（派生）開発の場合、改造せずにそのまま再利用した既存コードを除いた実績値。「追加／新規」分の行数と「変更」して作成した行数の合算に相当するが、SLOC実績値には、既存製品から「削除」した行数も含める。

図表4.8-1　SLOC（コード行数）実績値（全体、50KSLOC刻み）



図表4.8-2　SLOC（コード行数）実績値（100KSLOC以下、5KSLOC刻み）



右に、SLOC実績値の軸を拡大したものを示す。

図表4.8-3　SLOC（コード行数）実績値の基本統計量

[KSLOC]

N	最小	P25	中央	P75	最大	平均	標準偏差
595	0.001	2.292	6.754	27.200	1551.100	40.907	130.795

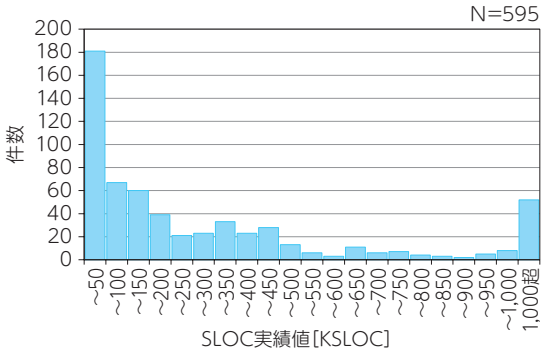
※集計対象データ：実効SLOC実績値（導出指標）

(2) SLOC（母体含むコード行数）実績値

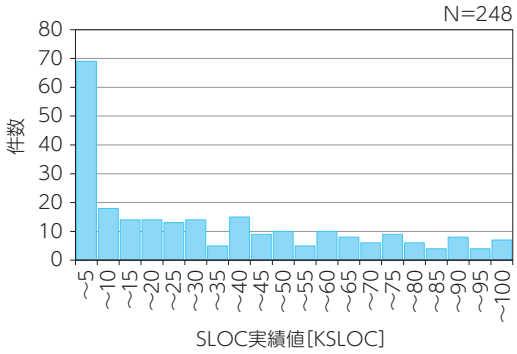
収集したプロジェクトの母体を含む全体のソフトウェア規模（SLOC実績値）の分布を示す。

改良（派生）開発の場合、改造せずにそのまま再利用した母体の既存コードを含む実績値。

図表4.8-4　SLOC（母体含むコード行数）実績値（全体、50KSLOC刻み）



図表4.8-5　SLOC（母体含むコード行数）実績値（100KSLOC以下、5KSLOC刻み）



右に、SLOC実績値の軸を拡大したものを示す。

図表4.8-6　SLOC（母体含むコード行数）実績値の基本統計量

[KSLOC]

N	最小	P25	中央	P75	最大	平均	標準偏差
595	0.005	35.650	137.533	396.580	26400.000	447.482	1430.695

※集計対象データ：7-6_SLOC実績値(母体)

4.9 工期

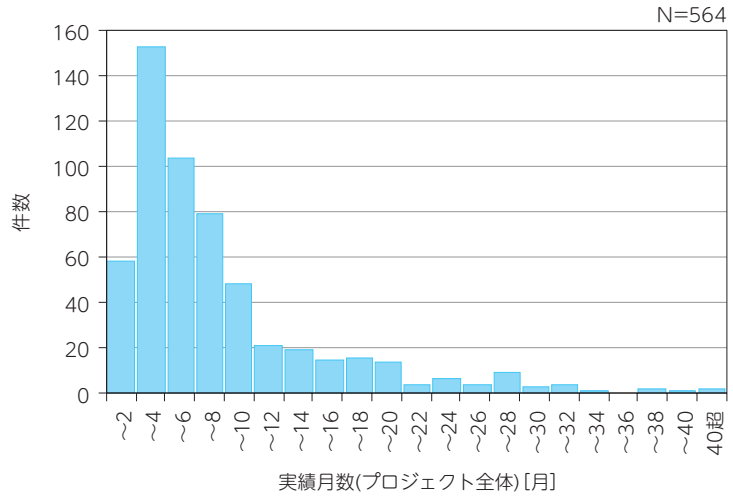
この節では、開発プロジェクトの工期に関する、以下のプロファイルを掲載する。

- プロジェクト全体の開発期間
- 開発6工程の開発期間
- 開発5工程（要求定義を除く）の開発期間

(1)プロジェクト全体の開発期間

プロジェクト全体の開発期間の分布を示す。実施する工程には依存しない。

図表4.9-1　プロジェクト全体の開発期間



図表4.9-2　プロジェクト全体の開発期間の基本統計量

[月]

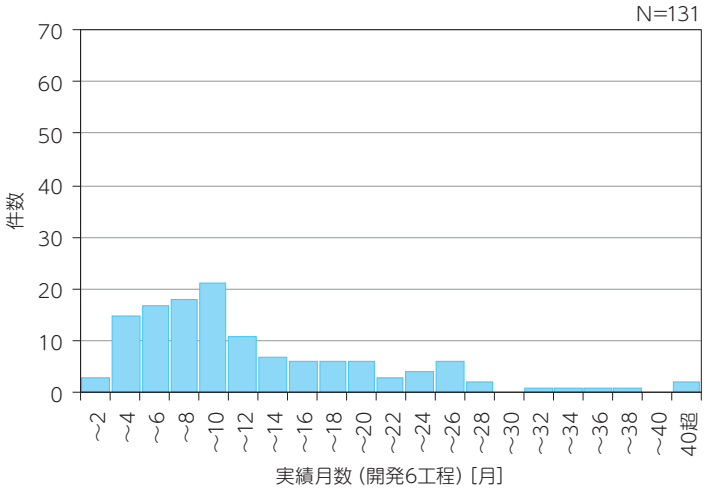
N	最小	P25	中央	P75	最大	平均	標準偏差
564	0.56	3.22	5.45	9.01	50.73	7.75	7.03

※集計対象データ：8-4-3_プロジェクト全体工期（実績）_月数

(2)開発6工程の開発期間

要求定義～総合テストまでの開発6工程を実施しているプロジェクトの工期の分布を示す。

図表4.9-3 開発6工程の開発期間



図表4.9-4 開発6工程の開発期間の基本統計量

[月]

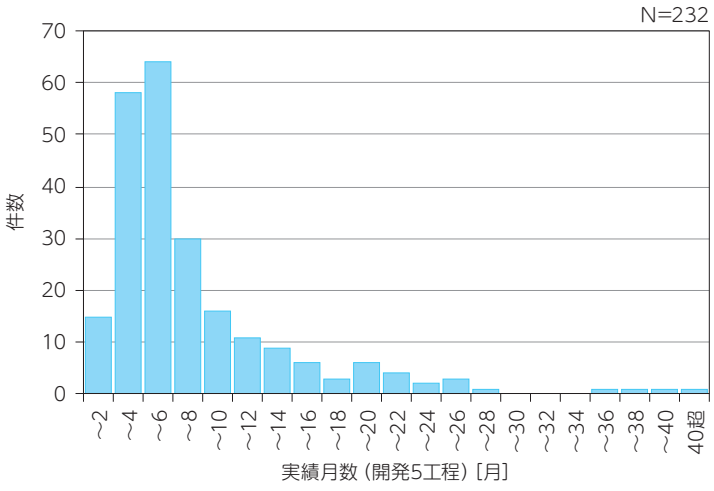
N	最小	P25	中央	P75	最大	平均	標準偏差
131	0.60	5.85	8.77	16.02	50.73	11.82	8.76

※1：集計対象データ：実績月数(開発6工程)（導出指標）
※2：開発6工程プロジェクト131件を対象とする
※3：開発6工程プロジェクト：要求定義～総合テストの6工程が含まれているプロジェクト

(3)開発5工程（要求定義を除く）の開発期間

アーキテクチャ設計～総合テストまでの開発5工程を実施しているプロジェクトの工期の分布を示す。

図表4.9-5 開発5工程（要求定義を除く）の開発期間



図表4.9-6 開発５工程（要求定義を除く）の開発期間の基本統計量

[月]

N	最小	P25	中央	P75	最大	平均	標準偏差
232	0.50	3.50	5.25	8.66	41.57	7.48	6.63

※1：集計対象データ：実績月数（開発5工程）（導出指標）
※2：開発5工程プロジェクト232件を対象とする
※3：開発5工程プロジェクト：アーキテクチャ設計～総合テストの5工程が含まれているプロジェクト

4.10 工数

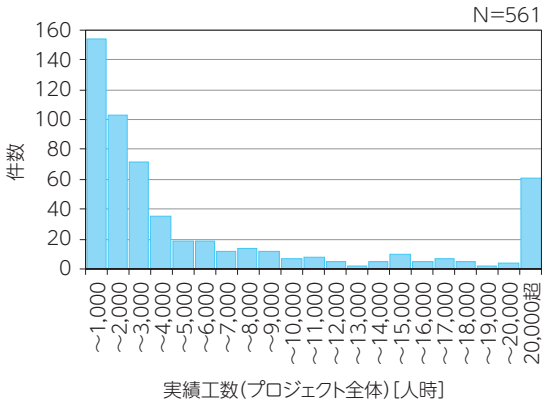
この節では、開発プロジェクトの工数に関する、以下のプロファイルを掲載する。

- プロジェクト全体の工数の実績値（人時換算）
- プロジェクト全体の工数の実績値（人月換算）
- 人月―人時の換算係数

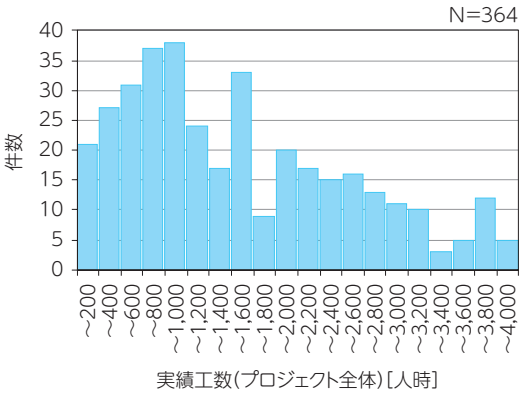
(1) プロジェクト全体の工数の実績値（人時換算）

プロジェクト全体の工数の分布を示す。実施する工程には依存しない。

図表4.10-1　プロジェクト全体の工数の実績値（人時換算）（全体、1,000人時刻み）



図表4.10-2　プロジェクト全体の工数の実績値（人時換算）（4,000人時以下、200人時刻み）



右に、実績工数の軸を拡大したものを示す。

図表4.10-3　プロジェクト全体の開発期間の基本統計量

[人時]

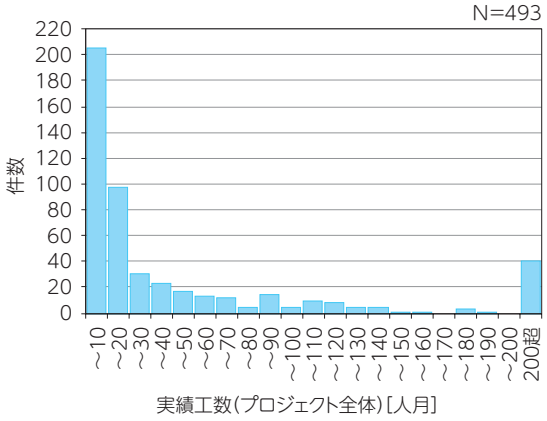
N	最小	P25	中央	P75	最大	平均	標準偏差
561	6	907	2,278	7,365	292,000	10,041	26,553

※集計対象データ：実績工数（プロジェクト全体）（導出指標）の人時換算値。工数が人時で提出されている場合は、提出値を使用。工数が人月で提出されている場合は、「9-2_人時換算係数」から人時に換算

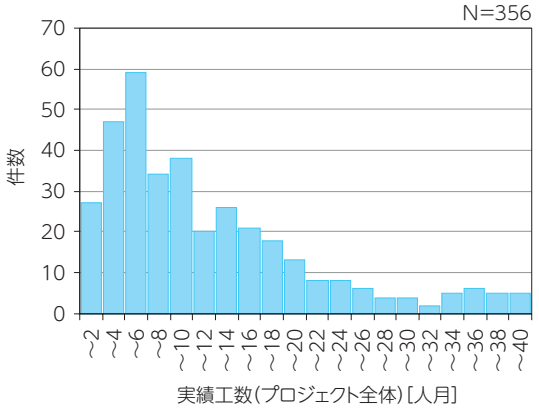
(2) プロジェクト全体の工数の実績値 (人月換算)

プロジェクト全体の工数の分布を人月に換算して示す。実施する工程には依存しない。

図表4.10-4 プロジェクト全体の工数の実績値 (人月換算) (全体、10人月刻み)



図表4.10-5 プロジェクト全体の工数の実績値 (人月換算) (40人月以下、2人月刻み)



右に、実績工数の軸を拡大したものを示す。

図表4.10-6 プロジェクト全体の工数の実績値の基本統計量 (人月換算)

[人月]

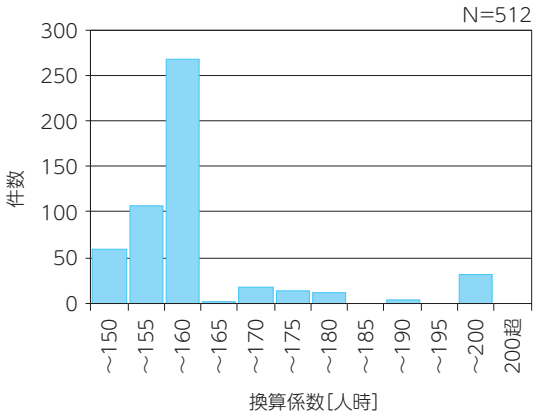
N	最小	P25	中央	P75	最大	平均	標準偏差
493	0.04	5.80	13.40	48.14	1,825.00	63.30	169.68

※集計対象データ：実績工数 (プロジェクト全体) (導出指標) の人月換算値。工数が人月で提出されている場合は、提出値を使用。工数が人時で提出されている場合は、「9-2_人時換算係数」から人月に換算

(3) 人月―人時の換算係数

工数1人月を人時に換算する係数の分布を示す。

図表4.10-7 人月－人時換算係数



図表4.10-8 人月－人時換算係数の基本統計量

[人時]

N	中央	平均	標準偏差
512	160.00	159.89	16.87

※集計対象データ：9-2_人時換算係数

4.11 体制

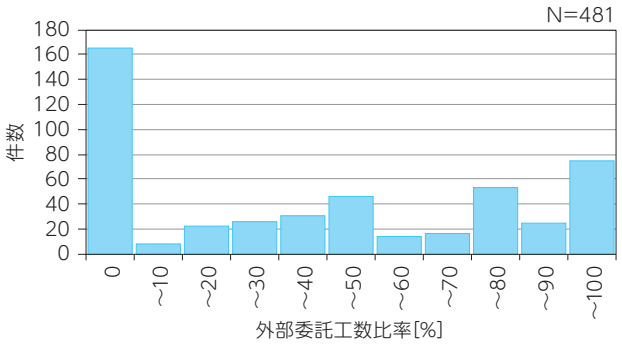
この節では、プロジェクトの開発体制に関する、以下のプロファイルを掲載する。

- 外部委託工数比率
- 月あたりの要員数

(4)外部委託工数比率

プロジェクト全体工数のうち、外部委託した工数比率の分布を示す。外部委託工数は実施した工程にかかわらずプロジェクト全体の外部委託工数合計値を、実績工数(プロジェクト全体)で割った値である。

図表4.11-1　外部委託工数比率



図表4.11-2　外部委託工数比率の基本統計量

[%]

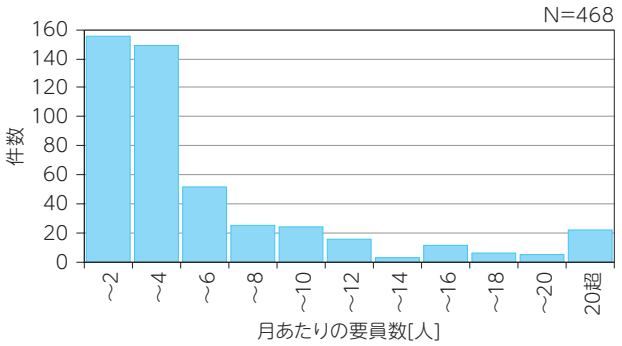
N	最小	P25	中央	P75	最大	平均	標準偏差
481	0.00	0.00	37.00	79.00	100.00	41.42	38.20

※1：集計対象データ：外部委託工数比率(導出指標)
※2：外部委託工数を明示的に“0”で回答しているものは“0%”として分布に記載

(5)月あたりの要員数

月あたりの要員数は、工数と工期の月数から算出する数値である。詳しくは付録A.3に導出指標「月あたりの要員数」に示す。

図表4.11-3　月あたりの要員数



図表4.11-4　月あたりの要員数の基本統計量

[人]

N	最小	P25	中央	P75	最大	平均	標準偏差
468	0.01	1.52	2.90	5.86	129.63	6.00	11.35

※1：集計対象データ：月あたりの要員数(導出指標)
※2：工数1人月を人時単位に換算する係数が示されているものが対象

4.12 信頼性

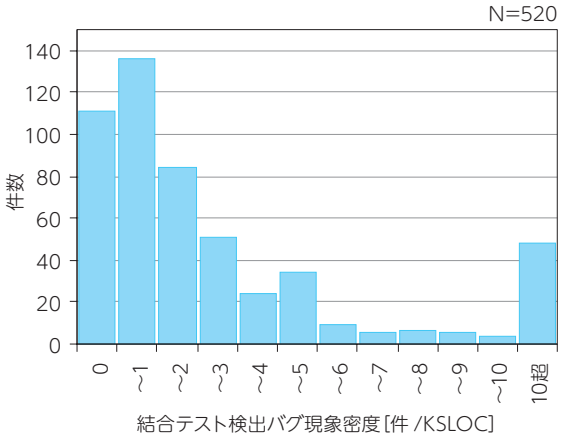
この節では、開発したソフトウェアの信頼性に関する事項として、以下のプロファイルを掲載する。

- 結合テスト検出バグ現象密度と原因密度
- 総合テスト検出バグ現象密度と原因密度
- 総合テスト検出バグ現象密度と原因密度（母体含む）
- 品質保証の体制
- 品質基準、レビューの有無
- テスト計画書、レビューの有無

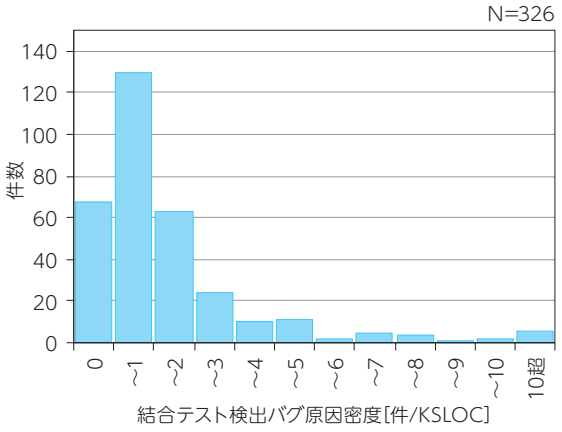
(1)結合テスト検出バグ現象密度と原因密度

結合テストで検出したバグ現象密度とバグ原因密度(1KSLOCあたりの件数)の分布を示す。

図表4.12-1　結合テスト検出バグ現象密度



図表4.12-2　結合テスト検出バグ原因密度



図表4.12-3　結合テスト検出バグ現象密度と原因密度の基本統計量

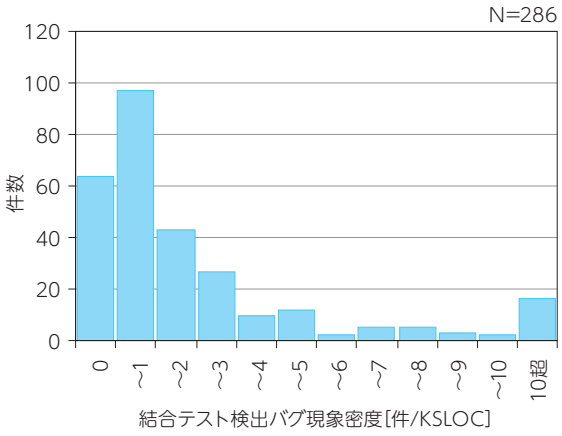
[件/KSLOC]

	N	P25	中央	P75
結合テスト検出バグ現象密度(導出指標)	520	0.140	1.072	3.316
結合テスト検出バグ原因密度(導出指標)	326	0.089	0.690	1.740

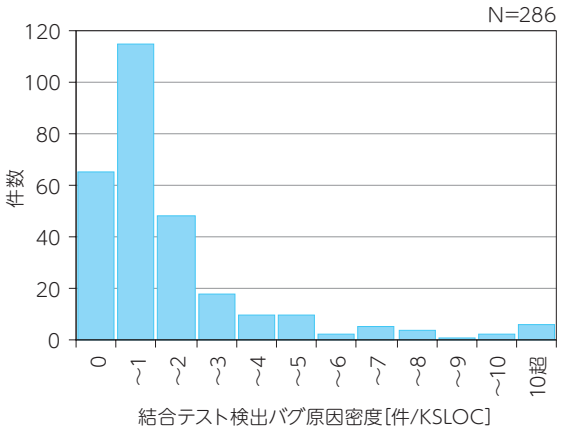
※集計対象データ：結合テスト検出バグ現象密度(導出指標)、結合テスト検出バグ原因密度(導出指標)

以下に、結合テスト検出バグ現象密度と原因密度の両データがそろっている標本を対象に分布を比較する。

図表4.12-4　結合テスト検出バグ現象密度と原因密度の比較(現象密度)



図表4.12-5　結合テスト検出バグ現象密度と原因密度の比較(原因密度)



図表4.12-6　結合テスト検出バグ現象密度と原因密度の比較の基本統計量

[件/KSLOC]

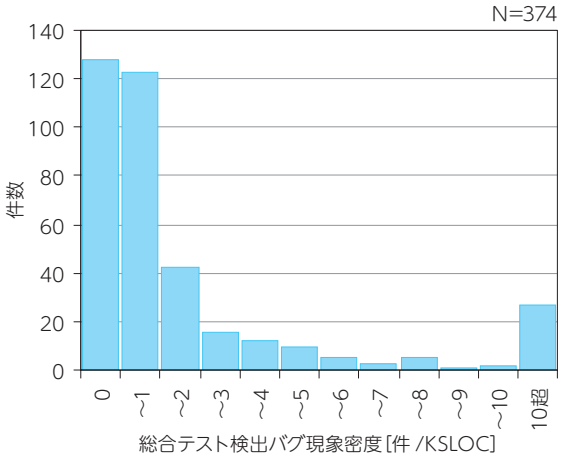
	N	P25	中央	P75
結合テスト検出バグ現象密度(導出指標)	286	0.072	0.740	2.328
結合テスト検出バグ原因密度(導出指標)	286	0.052	0.585	1.768

※集計対象データ：結合テスト検出バグ現象密度(導出指標)、結合テスト検出バグ原因密度(導出指標)

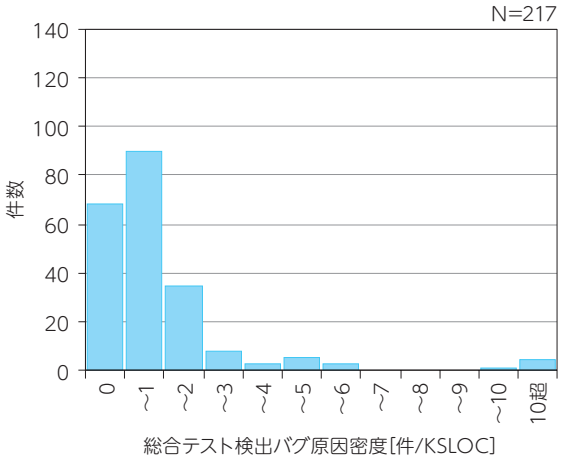
(2)総合テスト検出バグ現象密度と原因密度

総合テストで検出したバグ現象密度とバグ原因密度(1KSLOCあたりの件数)の分布を示す。

図表4.12-7 総合テスト検出バグ現象密度



図表4.12-8 総合テスト検出バグ原因密度



図表4.12-9 総合テスト検出バグ現象密度と原因密度の基本統計量

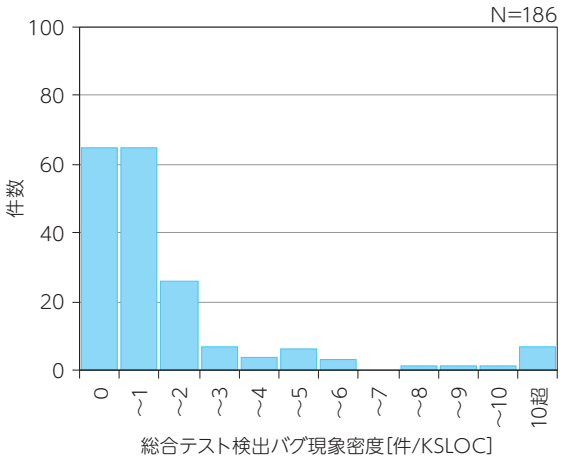
[件/KSLOC]

	N	P25	中央	P75
総合テスト検出バグ現象密度(導出指標)	374	0.000	0.285	1.675
総合テスト検出バグ原因密度(導出指標)	217	0.000	0.329	1.059

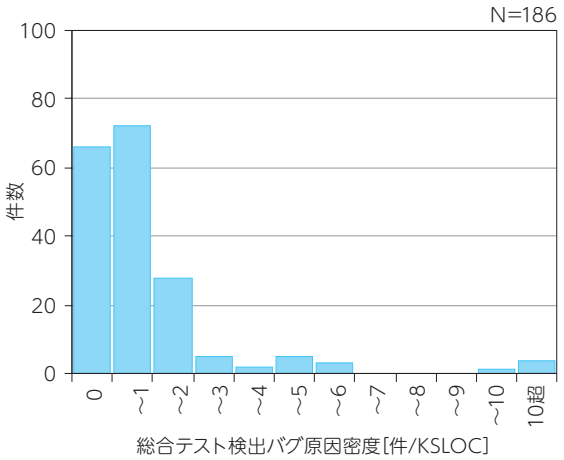
※集計対象データ:総合テスト検出バグ現象密度(導出指標)、総合テスト検出バグ原因密度(導出指標)

以下に、総合テスト検出バグ現象密度と原因密度の両データがそろっている標本を対象に分布を比較する。

図表4.12-10 総合テスト検出バグ現象密度と原因密度の比較(現象密度)



図表4.12-11 総合テスト検出バグ現象密度と原因密度の比較(原因密度)



図表4.12-12 総合テスト検出バグ現象密度と原因密度の比較の基本統計量

[件/KSLOC]

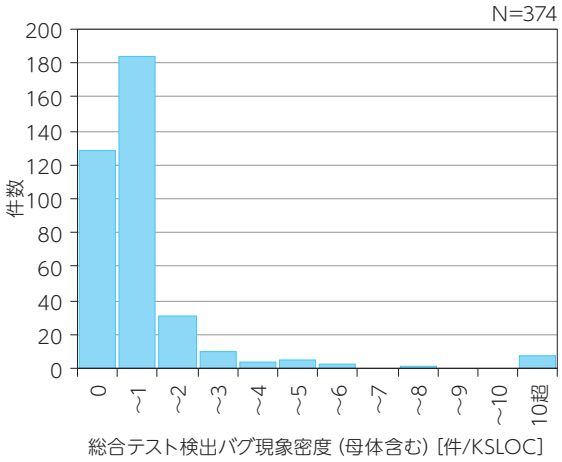
	N	P25	中央	P75
総合テスト検出バグ現象密度(導出指標)	186	0.000	0.299	1.279
総合テスト検出バグ原因密度(導出指標)	186	0.000	0.275	1.048

※集計対象データ:総合テスト検出バグ現象密度(導出指標)、総合テスト検出バグ原因密度(導出指標)

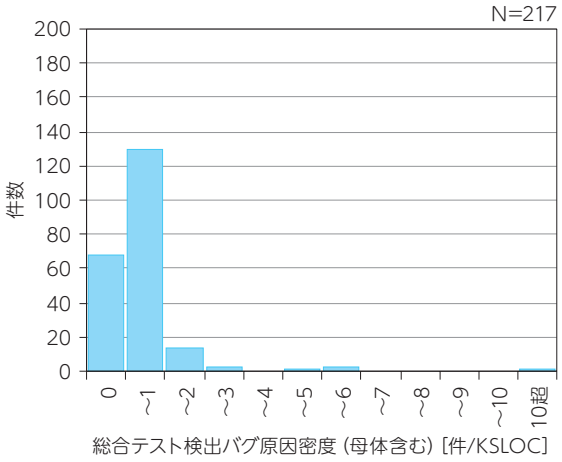
(3)総合テスト検出バグ現象密度と原因密度(母体含む)

総合テストで検出したバグ現象密度とバグ原因密度(母体含む1KSLOCあたりの件数)の分布を示す。

図表4.12-13　総合テスト検出バグ現象密度(母体含む)



図表4.12-14　総合テスト検出バグ原因密度(母体含む)



図表4.12-15　総合テスト検出バグ現象密度と原因密度(母体含む)の基本統計量

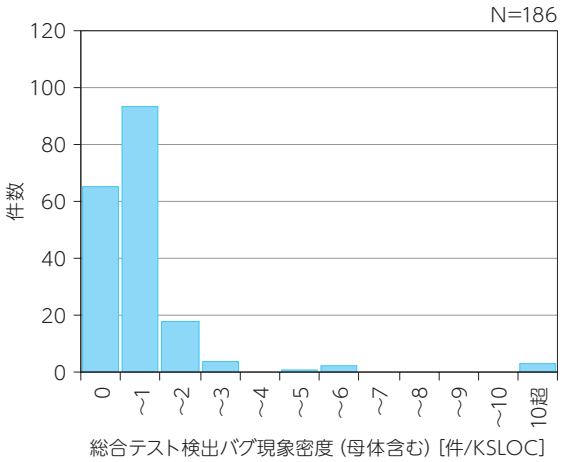
[件/KSLOC]

	N	P25	中央	P75
総合テスト検出バグ現象密度(母体含む)(導出指標)	374	0.000	0.012	0.352
総合テスト検出バグ原因密度(母体含む)(導出指標)	217	0.000	0.027	0.274

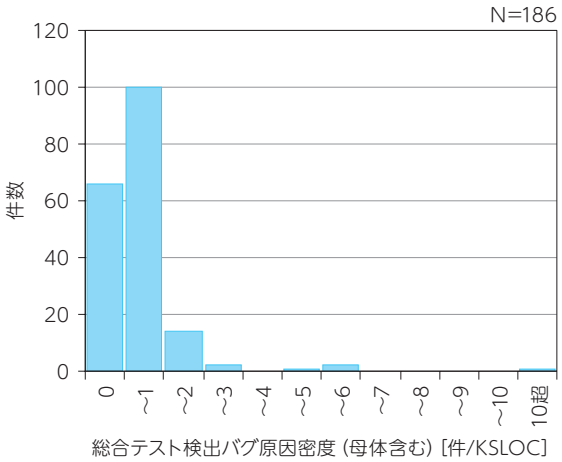
※集計対象データ：総合テスト検出バグ現象密度(母体含む)(導出指標)、総合テスト検出バグ原因密度(母体含む)(導出指標)

以下に、総合テスト検出バグ現象密度と原因密度(母体含む)の両データがそろっている標本を対象に分布を比較する。

図表4.12-16　総合テスト検出バグ現象密度と原因密度(母体含む)の比較(現象密度)



図表4.12-17　総合テスト検出バグ現象密度と原因密度(母体含む)の比較(原因密度)



図表4.12-18　総合テスト検出バグ現象密度と原因密度(母体含む)の比較の基本統計量

[件/KSLOC]

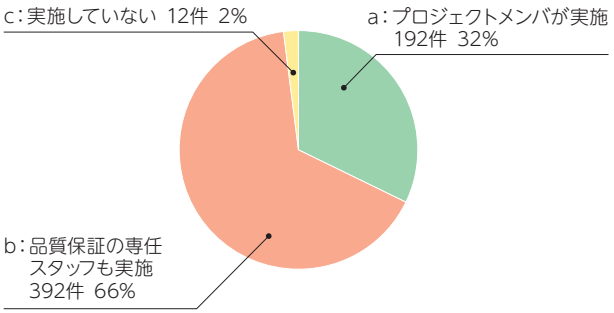
	N	P25	中央	P75
総合テスト検出バグ現象密度(母体含む)(導出指標)	186	0.000	0.018	0.383
総合テスト検出バグ原因密度(母体含む)(導出指標)	186	0.000	0.016	0.239

※集計対象データ：総合テスト検出バグ現象密度(母体含む)(導出指標)、総合テスト検出バグ原因密度(母体含む)(導出指標)

(4) 品質保証の体制

品質保証の体制を示す。

図表4.12-19 品質保証の体制

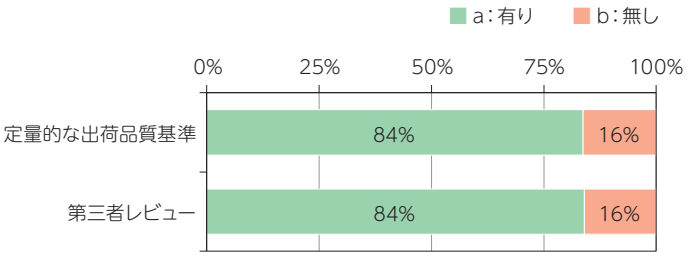


N＝596
※集計対象データ：10-5_品質保証体制

(5) 品質基準、レビューの有無

対象プロジェクトにおいて定量的な出荷品質基準が設定されていたか、また、プロジェクトの期間中、各工程のレビューなどで第三者（例：品質保証部門、PMO）によるレビューを実施したか否かを示す。

図表4.12-20 品質基準、レビューの有無



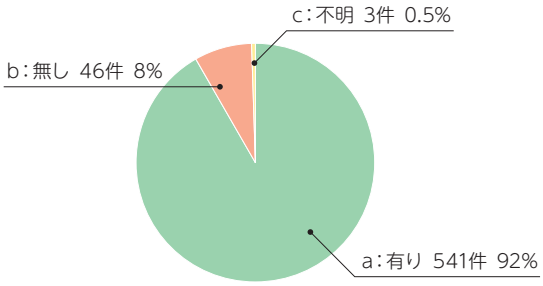
図表4.12-21 品質基準、レビューの有無一覧

集計対象データ	a：有り	b：無し	N
10-7 定量的な出荷品質基準	500	97	597
10-8 第三者レビュー	498	95	593

(6) テスト計画書、レビューの有無

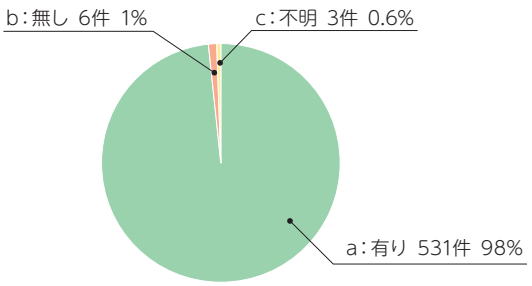
テスト計画書を作成しているか、また作成している場合は、テスト計画書のレビューを実施しているかどうかを示す。

図表4.12-22 テスト計画書の有無



N＝590
※集計対象データ：4-17_テスト計画書の有無

図表4.12-23 テスト計画書のレビューの有無



N＝540
※1 集計対象データ：4-18_テスト計画書のレビューの有無
※2「テスト計画書の有無」において、「有り」に該当するデータを対象

4.13 実施工程の組み合わせパターン

この節では、実施工程の組み合わせパターンの結果を示す。

実施工程の組み合わせパターンとは、開発プロジェクトにおける実施工程の有無が同じものをグルーピングしたパターンである。

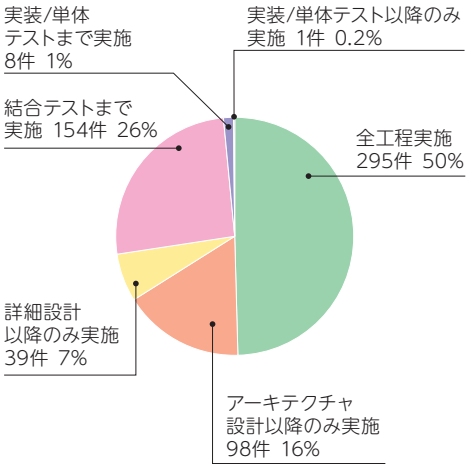
図表4.13-1　実施工程の組み合わせパターン

実施工程の組み合わせパターン	工程						新規開発	改良（派生）開発	合計
	要求定義	アーキテクチャ設計	詳細設計	実装・単体テスト	結合テスト	総合テスト			
全工程実施	○	○	○	○	○	○	18	174	295
	○	⇒	○	○	○	○	1	10	
	○	×	○	○	○	○	3	42	
	○	○	⇒	○	○	○	0	5	
	○	○	×	○	○	○	0	2	
	○	⇒	○	⇒	○	○	0	1	
	○	⇒	⇒	○	○	○	0	3	
	○	○	○	○	×	○	0	2	
	○	⇒	⇒	○	⇒	○	0	1	
	⇒	○	○	○	○	○	0	20	
	○	⇒	○	○	⇒	○	0	1	
	⇒	⇒	⇒	⇒	○	○	0	6	
	⇒	⇒	⇒	⇒	⇒	○	1	5	
アーキテクチャ設計以降のみ実施	×	○	○	○	○	○	1	84	98
	×	○	○	○	×	○	0	3	
	×	○	⇒	○	○	○	0	3	
	×	○	×	○	○	○	0	3	
	×	⇒	○	○	○	○	0	2	
	×	○	○	○	⇒	○	1	0	
	×	○	⇒	○	⇒	○	0	1	
詳細設計以降のみ実施	×	×	○	○	○	○	2	32	39
	×	×	○	○	⇒	○	0	1	
	×	×	○	○	×	○	0	2	
	×	×	⇒	○	×	○	0	1	
	×	×	⇒	⇒	⇒	○	0	1	
結合テストまで実施	○	○	○	○	○	×	2	29	154
	○	○	⇒	○	○	×	0	4	
	○	⇒	○	○	○	×	0	1	
	○	×	○	○	○	×	0	10	
	○	×	⇒	○	○	×	1	0	
	⇒	○	○	○	○	×	0	5	
	×	○	○	○	○	×	4	60	
	×	○	⇒	○	○	×	0	2	
	×	⇒	○	○	○	×	0	7	
	×	×	○	○	○	×	5	23	
	×	×	×	○	○	×	0	1	
実装・単体テストまで実施	⇒	⇒	⇒	○	×	×	0	1	8
	×	○	○	○	×	×	0	3	
	×	×	○	○	×	×	2	1	
	×	×	×	○	×	×	0	1	
実装・単体テスト以降のみ実施	×	×	×	○	○	○	0	1	1
合計							41	554	595

[凡例] 各工程のフェーズ有無の記入記号（○、×、⇒）の解釈に従って、表に工程（フェーズ）の有無を表している。

※集計対象データ：
9-3-1_プロジェクト総工数に含まれるフェーズ_要求定義
9-3-2_プロジェクト総工数に含まれるフェーズ_アーキテクチャ設計
9-3-3_プロジェクト総工数に含まれるフェーズ_詳細設計
9-3-4_プロジェクト総工数に含まれるフェーズ_実装・単体テスト
9-3-5_プロジェクト総工数に含まれるフェーズ_結合テスト
9-3-6_プロジェクト総工数に含まれるフェーズ_総合テスト

図表4.13-2　実施工程の組み合わせパターンの比率



N＝595

※集計対象データ：
9-3_プロジェクト総工数に含まれるフェーズ

- 実施工程の組み合わせにいろいろなパターンがある理由は、以下のような状況があるためだと推測する。
  - 全工程を1つの開発体制で行っている。
  - 要求定義、総合テストを別の体制で行っている。

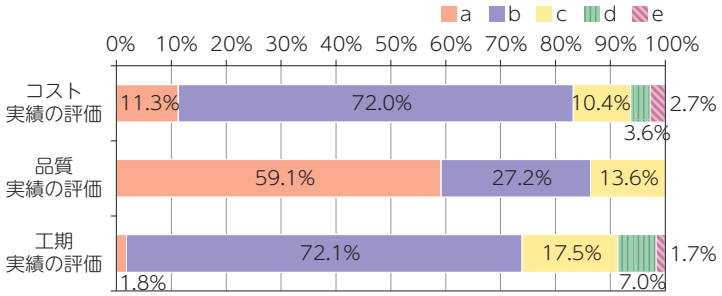
4.14　プロジェクト成否

この節では、プロジェクトの成否に関する実績の評価を掲載する。

(1)実績の評価 (QCD別)

コスト、品質、工期別にプロジェクト計画に対する実績の評価を示す。

図表4.14-1　実績の評価 (QCD別)



図表4.14-2　実績の評価 (QCD別)一覧

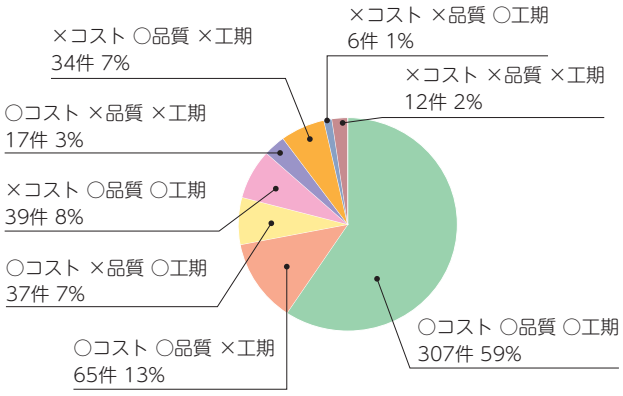
	←良い				悪い→	
集計対象データ	a	b	c	d	e	N
1-18　実績の評価 (コスト)	66	421	61	21	16	585
1-19　実績の評価 (品質)	330	152	76			558
1-20　実績の評価 (工期)	10	392	95	38	9	544

※選択肢a、b、c、d、e の内容
【1-18_実績の評価 (コスト)】　a：計画より10%以上少ないコストで達成、b：計画通り (±10%未満)、c：計画の30%以内の超過、d：計画の50%以内の超過、e：計画の50%を超える超過
【1-19_実績の評価 (品質)】　a：リリース後不具合数が0件、b：リリース後不具合数が予測値以内、c：リリース後不具合数が予測値超過
【1-20_実績の評価 (工期)】　a：納期より前倒し、b：納期通り、c：納期を1ヵ月未満遅延、d：納期を6ヵ月未満遅延、e：納期を6ヵ月以上遅延

(2)実績の評価 (QCD組み合わせパターン別)

コスト(Q)、品質(C)、工期(D)それぞれの実績の評価に対して成功 (○) と失敗 (×) を定義し、○と×の組み合わせパターン別にプロジェクトの割合を示す。

図表4.14-3　実績の評価 (QCD組み合わせパターン別)



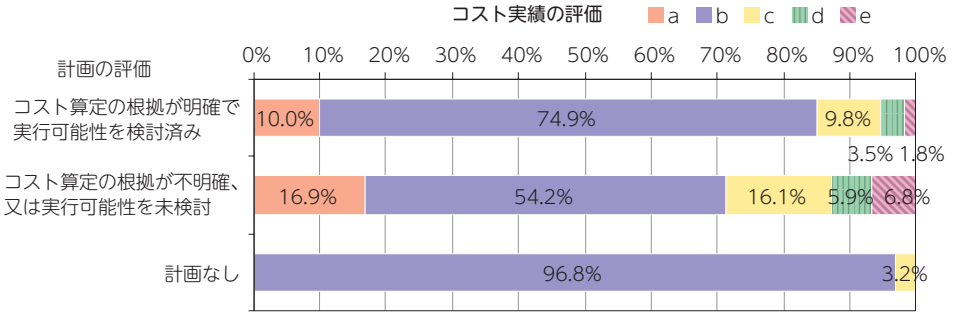
N＝517
※集計対象データ：1-18～20_実績の評価 (QCD)
※成功 (○) 失敗 (×) の定義

	成功 (○)	失敗 (×)
コスト	a：計画より10%以上少ないコストで達成 b：計画通り (±10%未満)	c：計画の30%以内の超過 d：計画の50%以内の超過 e：計画の50%を超える超過
品質	a：リリース後不具合数が0件 b：リリース後不具合数が予測値以内	c：リリース後不具合数が予測値超過
工期	a：納期より前倒し b：納期通り	c：納期を1ヵ月未満遅延 d：納期を6ヵ月未満遅延 e：納期を6ヵ月以上遅延

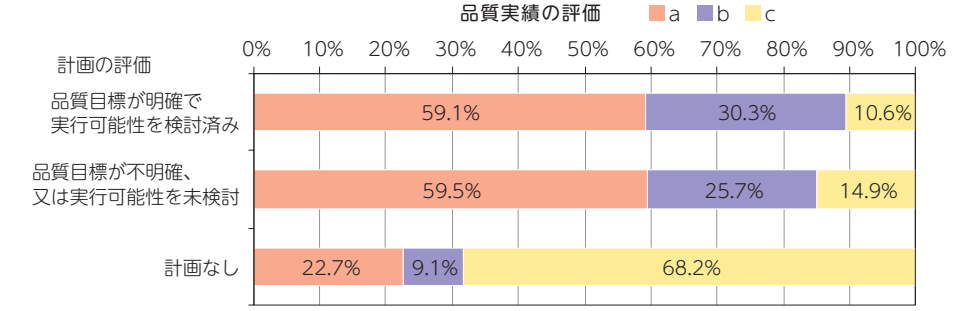
(3) 計画の評価と実績の評価 (QCD別)

コスト、品質、工期別にプロジェクト計画の評価と実績の評価との関係を示す。

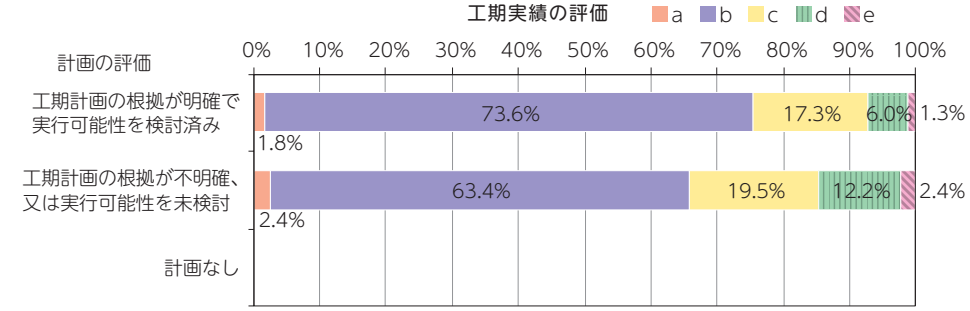
図表4.14-4　計画の評価と実績の評価 (コスト)



図表4.14-5　計画の評価と実績の評価 (品質)



図表4.14-6　計画の評価と実績の評価 (工期)



図表4.14-7　計画の評価と実績の評価 (QCD別) 一覧

		←(良い) 実績の評価 (悪い)→					
計画の評価		a	b	c	d	e	N
コスト	コスト算定の根拠が明確で実行可能性を検討済み	40	299	39	14	7	399
	コスト算定の根拠が不明確、又は実行可能性を未検討	20	64	19	7	8	118
	計画なし	0	30	1	0	0	31
品質	品質目標が明確で実行可能性を検討済み	250	128	45			423
	品質目標が不明確、又は実行可能性を未検討	44	19	11			74
	計画なし	5	2	15			22
工期	工期計画の根拠が明確で実行可能性を検討済み	8	331	78	27	6	450
	工期計画の根拠が不明確、又は実行可能性を未検討	2	52	16	10	2	82
	計画なし	0	0	0	0	0	0

※集計対象データ：1-15 ～ 17_計画の評価 (QCD)
1-18 ～ 20_実績の評価 (QCD)
※選択肢a、b、c、d、e の内容
【1-18_実績の評価 (コスト)】
a：計画より10%以上少ないコストで達成
b：計画通り (±10%未満)
c：計画の30%以内の超過
d：計画の50%以内の超過
e：計画の50%を超える超過
【1-19_実績の評価 (品質)】
a：リリース後不具合数が0件
b：リリース後不具合数が予測値以内
c：リリース後不具合数が予測値超過
【1-20_実績の評価 (工期)】
a：納期より前倒し
b：納期通り
c：納期を1 ヵ月未満遅延
d：納期を6 ヵ月未満遅延
e：納期を6 ヵ月以上遅延

5章　プロジェクトの主要要素の統計

5.1　位置付け …… 62
5.2　SLOC 規模 …… 63
5.2.1　SLOC規模の分布：改良（派生）開発、開発5工程
5.2.2　製品の特性別のSLOC規模の分布：改良（派生）開発、開発5工程
5.2.2.1　リアルタイム性（時間制約）別のSLOC規模
5.2.2.2　自然環境からの影響度合い別のSLOC規模
5.2.2.3　ユーザの多様性別のSLOC規模
5.2.2.4　法規等による規制度合い別のSLOC規模
5.2.2.5　M2Mの有無別のSLOC規模
5.2.2.6　ネットワーク接続の有無別のSLOC規模
5.2.2.7　稼動（非停止、オンデマンド）別のSLOC規模
5.2.2.8　オンライン保守の可否別のSLOC規模
5.2.3　障害リスク別のSLOC規模：改良（派生）開発、開発5工程
5.2.4　実績の評価別のSLOC規模の分布：改良（派生）開発、開発5工程
5.2.4.1　コスト実績の評価別のSLOC規模
5.2.4.2　品質実績の評価別のSLOC規模
5.2.4.3　工期実績の評価別のSLOC規模
5.3　工数 …… 71
5.3.1　工数の分布：改良（派生）開発、開発5工程
5.3.2　製品の特性別の工数の分布：改良（派生）開発、開発5工程
5.3.2.1　リアルタイム性（時間制約）別の工数
5.3.2.2　自然環境からの影響度合い別の工数
5.3.2.3　ユーザの多様性別の工数
5.3.2.4　法規等による規制度合い別の工数
5.3.2.5　M2Mの有無別の工数
5.3.2.6　ネットワーク接続の有無別の工数
5.3.2.7　稼動（非停止、オンデマンド）別の工数
5.3.2.8　オンライン保守の可否別の工数
5.3.3　障害リスク別の工数：改良（派生）開発、開発5工程
5.3.4　実績の評価別の工数：改良（派生）開発、開発5工程
5.3.4.1　コスト実績の評価別の工数
5.3.4.2　品質実績の評価別の工数
5.3.4.3　工期実績の評価別の工数

5章　プロジェクトの主要要素の統計

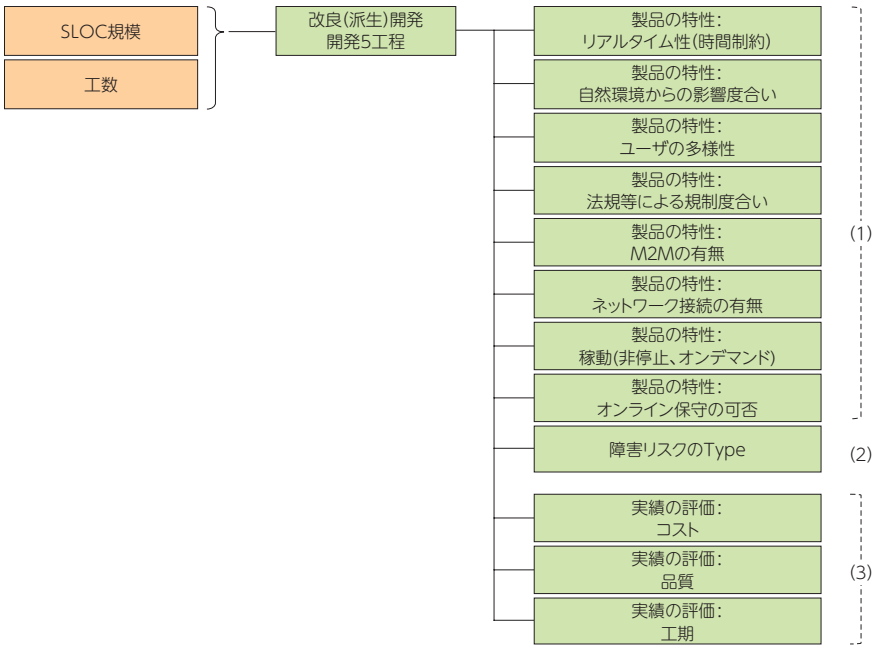
5.1　位置付け

この章では、収集データの分布傾向を大局的に見る手段として、プロジェクトの規模を表すSLOC規模や工数の分布状況を示す。6章以降、生産性や信頼性について散布図や箱ひげ図で示される分析結果を考察する際に、工数の分布やSLOC規模の分布を俯瞰的に捉えることで、分布のバラつきの状況の確認が可能となる。また、10章以降、プロジェクトの特性に応じた生産性や信頼性の分析結果を考察する際の参考となるように、工数の分布やSLOC規模の分布を幾つかのプロジェクト特性で層別した分布状況も併せて示す。

生産性は、単位工数あたりに開発可能なプログラムのSLOC規模を示す指標であり、開発工程の範囲を開発5工程（ソフトウェアアーキテクチャ設計～ソフトウェア総合テスト）に定めている。そのため、この章で示すSLOC規模や工数の分布は、開発5工程の作業が行われたプロジェクトデータを対象とする。

図表5.1-1に件数、分布の掲載対象と層別の種類を示す。

図表5.1-1　件数、分布の掲載対象と層別の種類



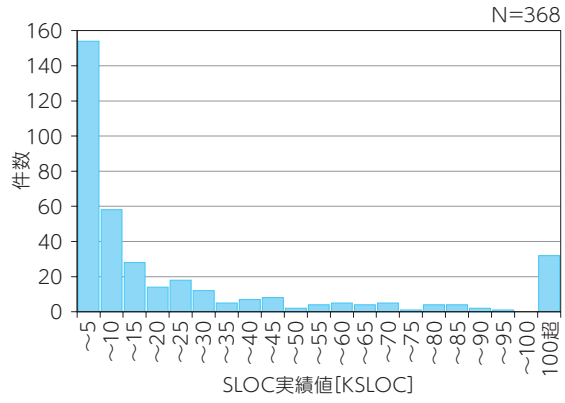
(1)製品の特性（リアルタイム性（時間制約）、自然環境からの影響度合い、ユーザの多様性、法規等による規制度合い、M２Mの有無、ネットワーク接続の有無、稼動（非停止、オンデマンド）、オンライン保守の可否）ごとに、層別した対象データの件数、分布、統計量を示す。
(2)障害リスク（TypeⅠ：社会的影響が殆どない、TypeⅡ：社会的影響が限定される、TypeⅢ：社会的影響が極めて大きい、TypeⅣ：人命に影響、甚大な経済損失）で層別して、対象データの件数、分布、統計量を示す。
(3)実績の評価（コスト、品質、工期）ごとに、層別した対象データの件数、分布、統計量を示す。

5.2 SLOC規模

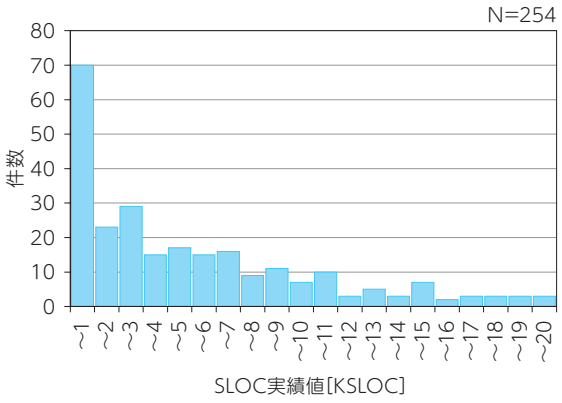
ここでは、改良（派生）開発で開発5工程（アーキテクチャ設計～総合テスト）の作業の行われたプロジェクトを対象に、SLOC規模データの分布及び基本統計量を示す。

5.2.1 SLOC規模の分布：改良（派生）開発、開発5工程

図表5.2-1 SLOC規模の分布（改良（派生）開発、開発5工程、全体、5KSLOC刻み）



図表5.2-2 SLOC規模の分布（改良（派生）開発、開発5工程、20KSLOC以下、1KSLOC刻み）



図表5.2-3 SLOC規模の基本統計量（改良（派生）開発、開発5工程）

[KSLOC]

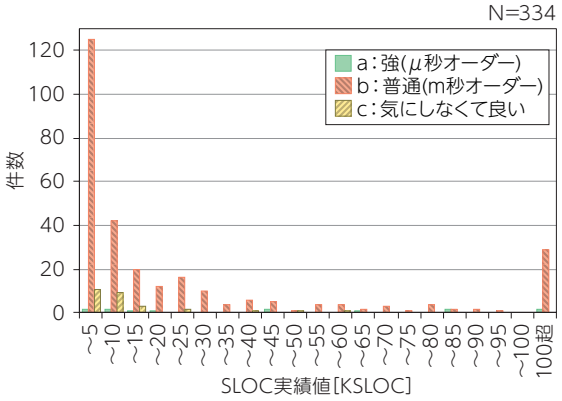
N	最小	P25	中央	P75	最大	平均	標準偏差
368	0.001	1.973	6.989	27.178	851.700	34.543	87.772

5.2.2　製品の特性別のSLOC規模の分布：改良（派生）開発、開発5工程

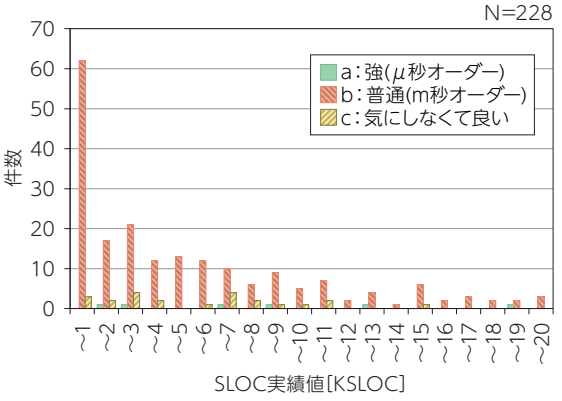
5.2.2.1　リアルタイム性（時間制約）別のSLOC規模

リアルタイム性（時間制約）別にSLOC規模データの分布及び基本統計量を示す。

図表5.2-4　リアルタイム性(時間制約)別SLOC規模の分布（改良（派生）開発、開発5工程、全体、5KSLOC刻み）



図表5.2-5　リアルタイム性(時間制約)別SLOC規模の分布（改良（派生）開発、開発5工程、20KSLOC以下、1KSLOC刻み）



図表5.2-6　リアルタイム性（時間制約）別SLOC規模の基本統計量（改良（派生）開発、開発5工程）

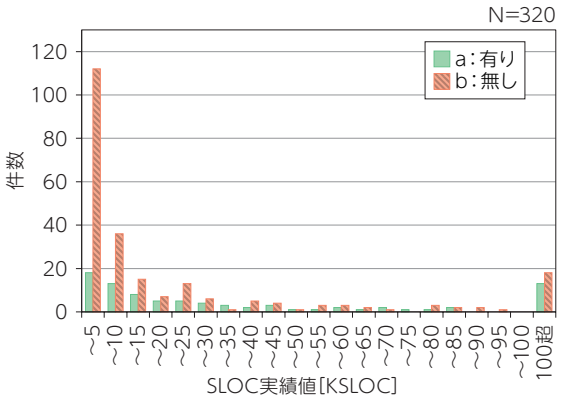
[KSLOC]

リアルタイム性（時間制約）	N	最小	P25	中央	P75	最大	平均	標準偏差
a：強（μ秒オーダー）	13	1.950	8.502	40.880	83.641	192.200	53.686	56.388
b：普通（m秒オーダー）	293	0.001	1.500	6.970	28.193	851.700	37.883	96.462
c：気にしなくて良い	28	0.161	2.607	6.260	10.470	56.834	11.016	13.960

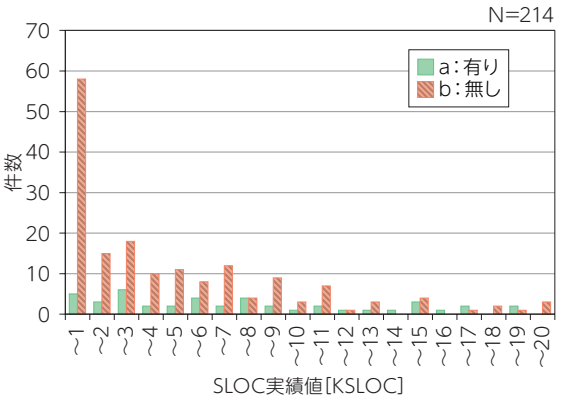
5.2.2.2　自然環境からの影響度合い別のSLOC規模

自然環境からの影響度合い別にSLOC規模データの分布及び基本統計量を示す。

図表5.2-7　自然環境からの影響度合い別SLOC規模の分布（改良（派生）開発、開発5工程、全体、5KSLOC刻み）



図表5.2-8　自然環境からの影響度合い別SLOC規模の分布（改良（派生）開発、開発5工程、20KSLOC以下、1KSLOC刻み）



図表5.2-9　自然環境からの影響度合い別SLOC規模の基本統計量（改良（派生）開発、開発5工程）

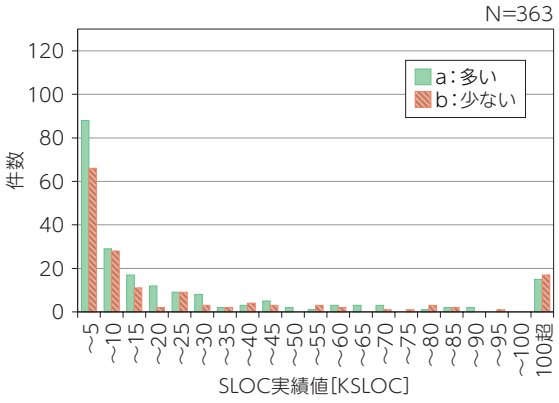
[KSLOC]

自然環境からの影響度合い	N	最小	P25	中央	P75	最大	平均	標準偏差
a：有り	85	0.030	5.786	18.129	58.514	851.700	56.879	114.219
b：無し	235	0.001	1.089	5.800	21.535	779.200	30.617	83.273

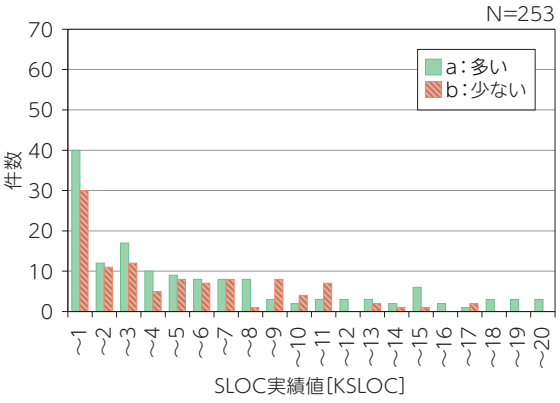
5.2.2.3 ユーザの多様性別のSLOC規模

ユーザの多様性別にSLOC規模データの分布及び基本統計量を示す。

図表5.2-10 ユーザの多様性別SLOC規模の分布（改良（派生）開発、開発5工程、全体、5KSLOC刻み）



図表5.2-11 ユーザの多様性別SLOC規模の分布（改良（派生）開発、開発5工程、20KSLOC以下、1KSLOC刻み）



図表5.2-12 ユーザの多様性別SLOC規模の基本統計量（改良（派生）開発、開発5工程）

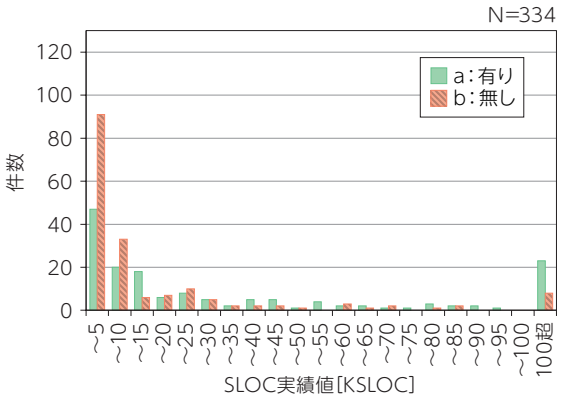
[KSLOC]

ユーザの多様性	N	最小	P25	中央	P75	最大	平均	標準偏差
a：多い	205	0.001	1.980	6.949	24.900	851.700	30.413	78.450
b：少ない	158	0.001	1.733	6.872	28.686	779.200	39.750	99.468

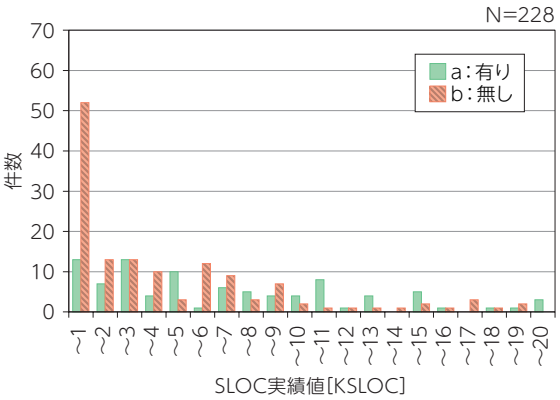
5.2.2.4 法規等による規制度合い別のSLOC規模

法規等による規制度合い別にSLOC規模データの分布及び基本統計量を示す。

図表5.2-13 法規等による規制度合い別SLOC規模の分布（改良（派生）開発、開発5工程、全体、5KSLOC刻み）



図表5.2-14 法規等による規制度合い別SLOC規模の分布（改良（派生）開発、開発5工程、20KSLOC以下、1KSLOC刻み）



図表5.2-15 法規等による規制度合い別SLOC規模の基本統計量（改良（派生）開発、開発5工程）

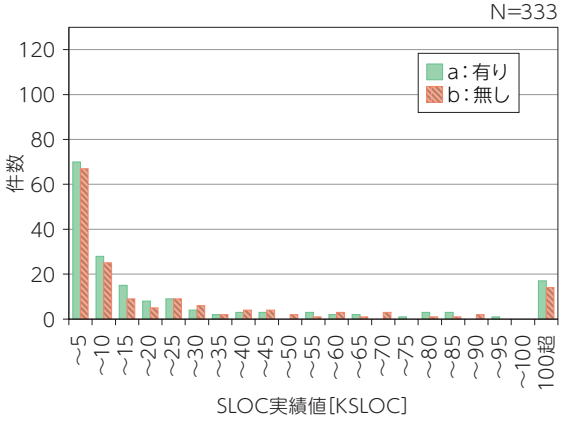
[KSLOC]

法規等による規制度合い	N	最小	P25	中央	P75	最大	平均	標準偏差
a：有り	158	0.030	4.345	12.941	51.397	779.200	49.097	94.173
b：無し	176	0.001	0.553	3.919	16.327	851.700	24.710	87.417

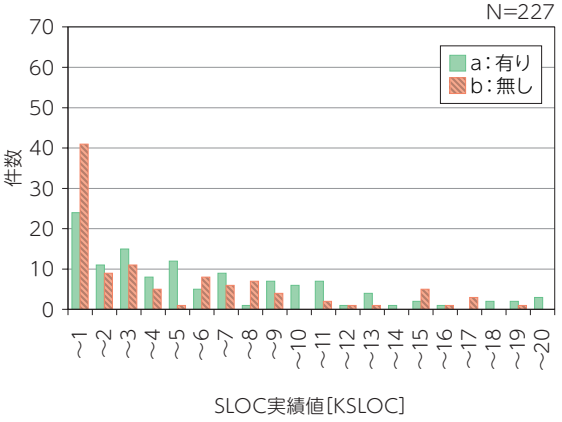
5.2.2.5 M２Mの有無別のSLOC規模

M２Mの有無別にSLOC規模データの分布及び基本統計量を示す。

図表5.2-16 M２Mの有無別SLOC規模の分布（改良(派生)開発、開発5工程、全体、5KSLOC刻み）



図表5.2-17 M２Mの有無別SLOC規模の分布（改良(派生)開発、開発5工程、20KSLOC以下、1KSLOC刻み）



図表5.2-18 M２Mの有無別SLOC規模の基本統計量（改良（派生）開発、開発5工程）

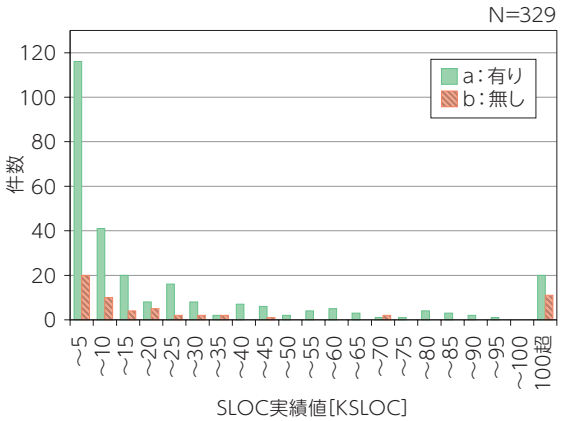
[KSLOC]

M２Mの有無	N	最小	P25	中央	P75	最大	平均	標準偏差
a：有り	174	0.013	2.643	8.510	26.639	779.200	37.256	90.817
b：無し	159	0.001	0.854	6.978	28.253	851.700	35.349	92.457

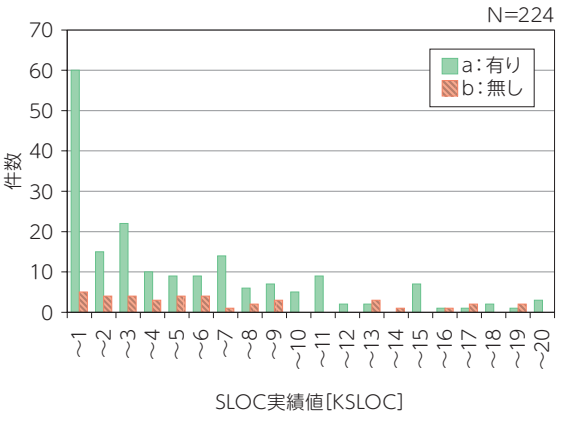
5.2.2.6 ネットワーク接続の有無別のSLOC規模

ネットワーク接続の有無別にSLOC規模データの分布及び基本統計量を示す。

図表5.2-19 ネットワーク接続の有無別SLOC規模の分布（改良（派生）開発、開発5工程、全体、5KSLOC刻み）



図表5.2-20 ネットワーク接続の有無別SLOC規模の分布（改良（派生）開発、開発5工程、20KSLOC以下、1KSLOC刻み）



図表5.2-21 ネットワーク接続の有無別SLOC規模の基本統計量（改良（派生）開発、開発5工程）

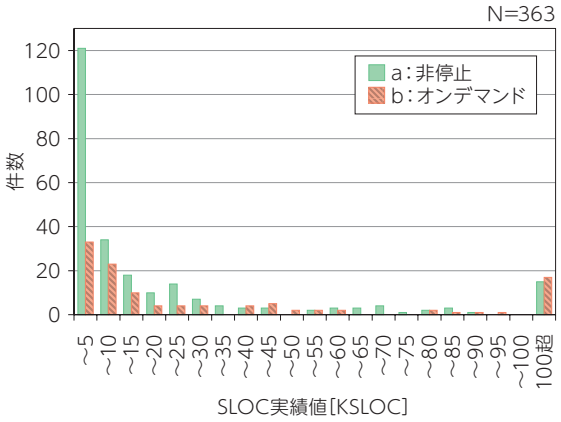
[KSLOC]

ネットワーク接続の有無	N	最小	P25	中央	P75	最大	平均	標準偏差
a：有り	270	0.001	1.513	6.799	26.850	595.000	30.097	67.542
b：無し	59	0.001	3.673	8.910	31.128	851.700	65.673	159.276

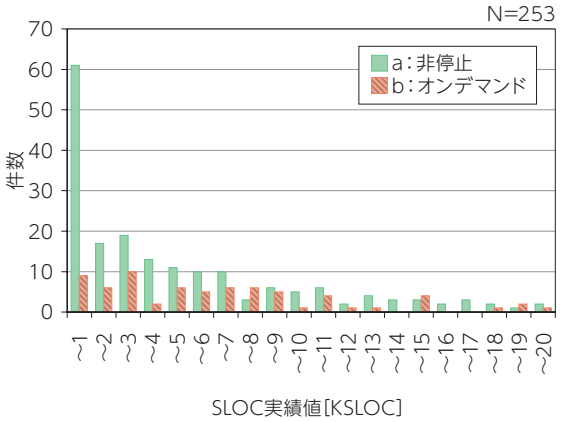
5.2.2.7 稼動（非停止、オンデマンド）別のSLOC規模

稼動（非停止、オンデマンド）別にSLOC規模データの分布及び基本統計量を示す。

図表5.2-22 稼動（非停止、オンデマンド）別SLOC規模の分布（改良（派生）開発、開発5工程、全体、5KSLOC刻み）



図表5.2-23 稼動（非停止、オンデマンド）別SLOC規模の分布（改良（派生）開発、開発5工程、20KSLOC以下、1KSLOC刻み）



図表5.2-24 稼動（非停止、オンデマンド）別SLOC規模の基本統計量（改良（派生）開発、開発5工程）

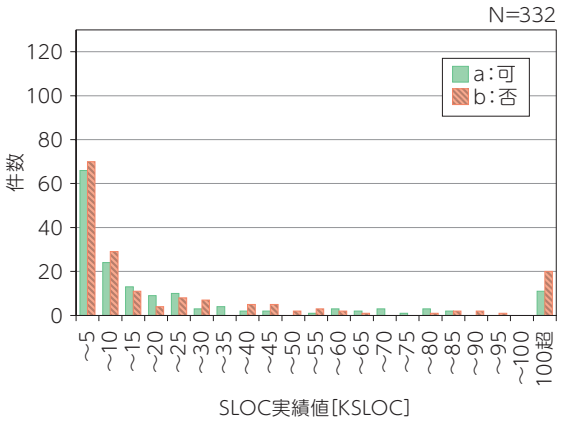
[KSLOC]

稼動（非停止、オンデマンド）	N	最小	P25	中央	P75	最大	平均	標準偏差
a：非停止	248	0.001	1.095	5.295	21.027	595.000	23.977	58.581
b：オンデマンド	115	0.071	4.397	10.493	44.588	851.700	57.121	128.378

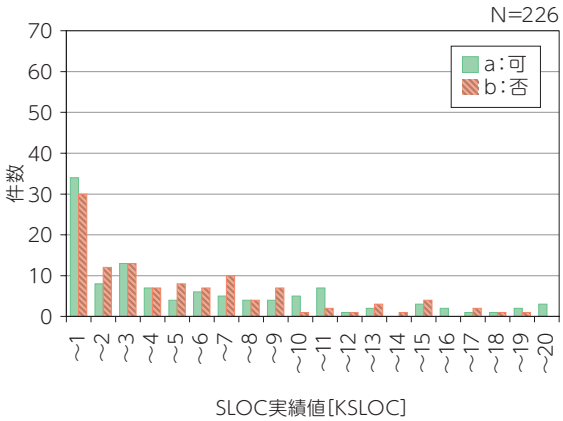
5.2.2.8 オンライン保守の可否別のSLOC規模

オンライン保守の可否別にSLOC規模データの分布及び基本統計量を示す。

図表5.2-25 オンライン保守の可否別SLOC規模の分布（改良（派生）開発、開発5工程、全体、5KSLOC刻み）



図表5.2-26 オンライン保守の可否別SLOC規模の分布（改良（派生）開発、開発5工程、20KSLOC以下、1KSLOC刻み）



図表5.2-27 オンライン保守の可否別SLOC規模の基本統計量（改良（派生）開発、開発5工程）

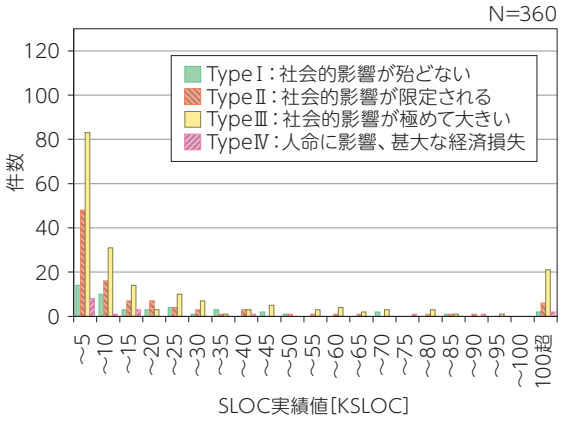
[KSLOC]

オンライン保守の可否	N	最小	P25	中央	P75	最大	平均	標準偏差
a：可	159	0.001	1.650	7.470	22.667	251.300	23.858	40.073
b：否	173	0.001	2.251	6.978	35.400	851.700	48.023	119.963

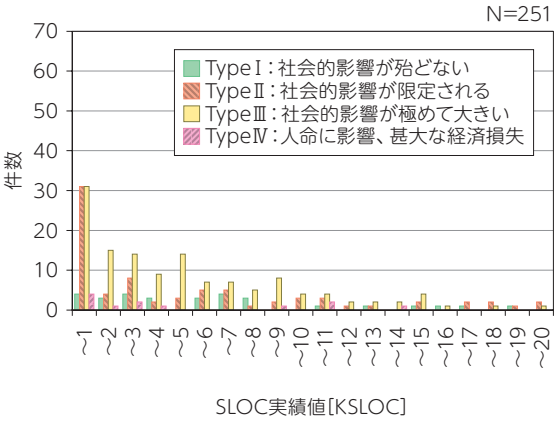
5.2.3 障害リスク別のSLOC規模：改良（派生）開発、開発5工程

障害リスク（Type）別にSLOC規模データの分布及び基本統計量を示す。

図表5.2-28 障害リスク（Type）別SLOC規模の分布（改良（派生）開発、開発5工程、全体、5KSLOC刻み）



図表5.2-29 障害リスク（Type）別SLOC規模の分布（改良（派生）開発、開発5工程、20KSLOC以下、1KSLOC刻み）



図表5.2-30 障害リスク（Type）別SLOC規模の基本統計量（改良（派生）開発、開発5工程）

[KSLOC]

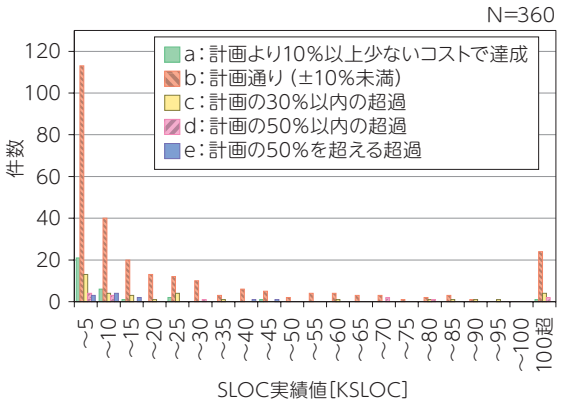
障害リスク	N	最小	P25	中央	P75	最大	平均	標準偏差
TypeⅠ：社会的影響が殆どない	46	0.300	3.418	7.596	25.637	147.488	21.644	29.600
TypeⅡ：社会的影響が限定される	102	0.001	0.379	5.485	18.950	595.000	28.251	80.351
TypeⅢ：社会的影響が極めて大きい	195	0.002	2.328	7.470	28.511	851.700	40.001	102.196
TypeⅣ：人命に影響、甚大な経済損失	17	0.030	1.950	8.160	37.710	170.700	34.827	55.419

5.2.4 実績の評価別のSLOC規模の分布：改良（派生）開発、開発5工程

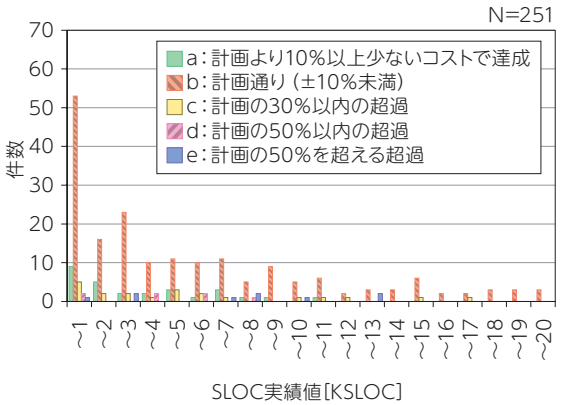
5.2.4.1 コスト実績の評価別のSLOC規模

コスト実績の評価別にSLOC規模データの分布及び基本統計量を示す。

図表5.2-31 コスト実績の評価別SLOC規模の分布（改良（派生）開発、開発5工程、全体、5KSLOC刻み）



図表5.2-32 コスト実績の評価別SLOC規模の分布（改良（派生）開発、開発5工程、20KSLOC以下、1KSLOC刻み）



図表5.2-33 コスト実績の評価別SLOC規模の基本統計量（改良（派生）開発、開発5工程）

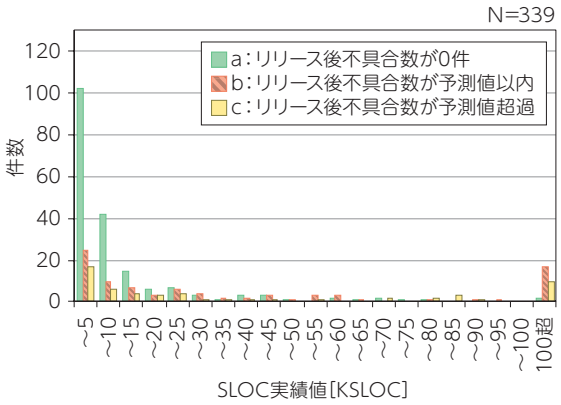
[KSLOC]

コスト実績の評価	N	最小	P25	中央	P75	最大	平均	標準偏差
a：計画より10%以上少ないコストで達成	32	0.026	0.936	2.800	6.759	140.915	9.899	25.087
b：計画通り（±10%未満）	269	0.001	1.900	7.370	27.200	595.000	31.192	68.105
c：計画の30%以内の超過	35	0.025	3.243	10.785	45.076	851.700	56.860	146.133
d：計画の50%以内の超過	13	0.161	3.560	7.470	69.100	779.200	93.097	203.360
e：計画の50%を超える超過	11	0.028	4.937	7.792	12.641	44.375	13.105	13.793

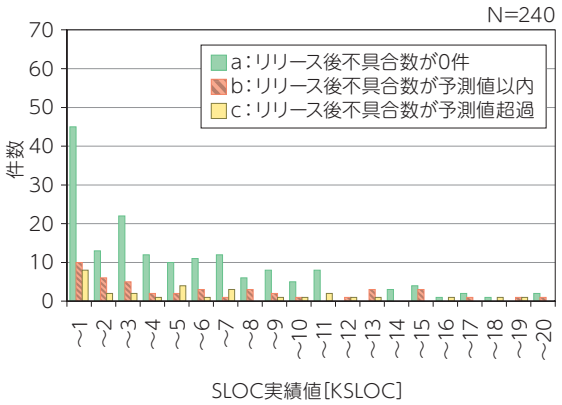
5.2.4.2 品質実績の評価別のSLOC規模

品質実績の評価別にSLOC規模データの分布及び基本統計量を示す。

図表5.2-34 品質実績の評価別SLOC規模の分布（改良（派生）開発、開発5工程、全体、5KSLOC刻み）



図表5.2-35 品質実績の評価別SLOC規模の分布（改良（派生）開発、開発5工程、20KSLOC以下、1KSLOC刻み）



図表5.2-36 品質実績の評価別SLOC規模の基本統計量（改良（派生）開発、開発5工程）

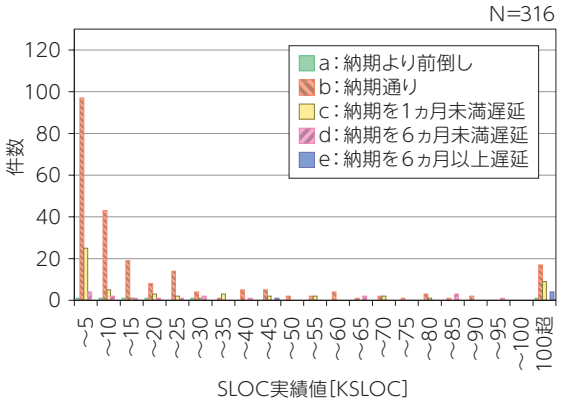
[KSLOC]

品質実績の評価	N	最小	P25	中央	P75	最大	平均	標準偏差
a：リリース後不具合数が0件	192	0.001	1.170	4.300	10.025	595.000	13.053	45.447
b：リリース後不具合数が予測値以内	90	0.002	3.913	19.850	59.247	473.000	57.838	96.755
c：リリース後不具合数が予測値超過	57	0.001	4.500	17.114	80.000	779.200	56.303	111.379

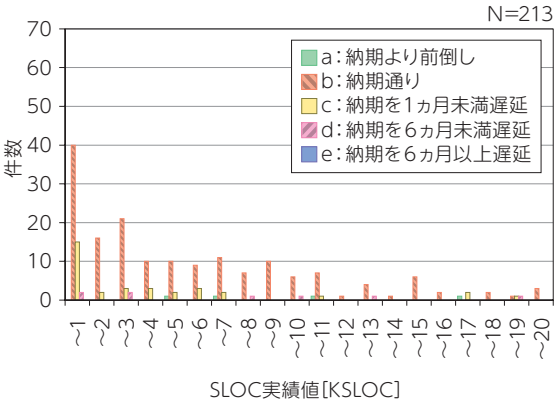
5.2.4.3 工期実績の評価別のSLOC規模

工期実績の評価別にSLOC規模データの分布及び基本統計量を示す。

図表5.2-37 工期実績の評価別SLOC規模の分布（改良（派生）開発、開発5工程、全体、5KSLOC刻み）



図表5.2-38 工期実績の評価別SLOC規模の分布（改良（派生）開発、開発5工程、20KSLOC以下、1KSLOC刻み）



図表5.2-39 工期実績の評価別SLOC規模の基本統計量（改良（派生）開発、開発5工程）

[KSLOC]

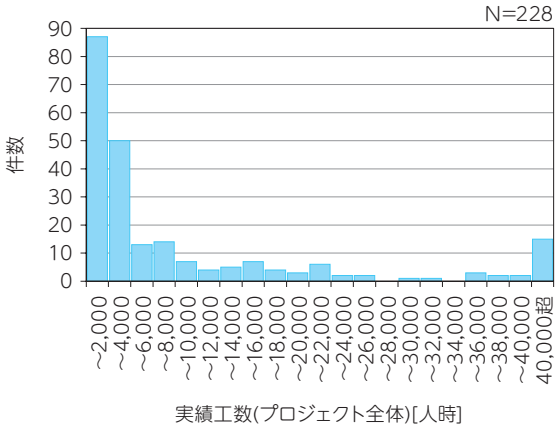
工期実績の評価	N	最小	P25	中央	P75	最大	平均	標準偏差
a：納期より前倒し	6	4.800	7.178	13.390	25.628	851.700	153.016	312.562
b：納期通り	231	0.001	2.328	6.978	21.185	595.000	29.753	71.008
c：納期を１ヵ月未満遅延	56	0.001	0.027	6.227	46.850	194.200	35.252	52.306
d：納期を６ヵ月未満遅延	18	0.071	8.103	23.952	60.576	94.000	35.149	32.260
e：納期を６ヵ月以上遅延	5	44.375	134.000	165.000	468.047	779.200	318.124	271.055

5.3 工数

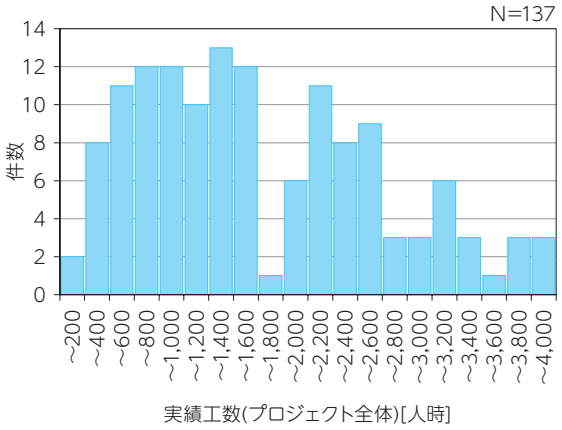
ここでは、改良（派生）開発で開発5工程（アーキテクチャ設計～総合テスト）の作業の行われたプロジェクトを対象に、工数データの分布及び基本統計量を示す。

5.3.1 工数の分布：改良（派生）開発、開発5工程

図表5.3-1 工数の分布（改良（派生）開発、開発5工程、全体、2,000人時刻み）



図表5.3-2 工数の分布（改良（派生）開発、開発5工程、4,000人時以下、200人時刻み）



図表5.3-3 工数の基本統計量（改良（派生）開発、開発5工程）

[人時]

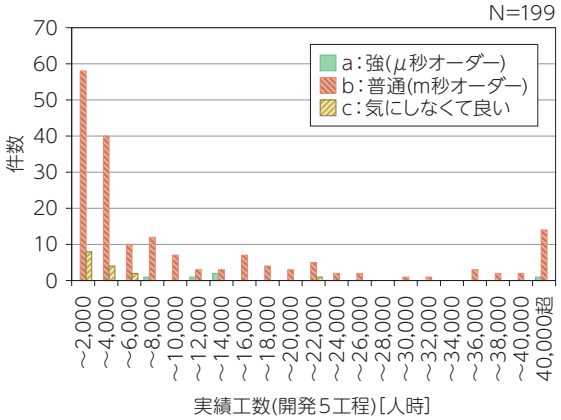
N	最小	P25	中央	P75	最大	平均	標準偏差
228	35	1,290	2,597	9,458	109,410	10,069	17,153

5.3.2　製品の特性別の工数の分布：改良（派生）開発、開発5工程

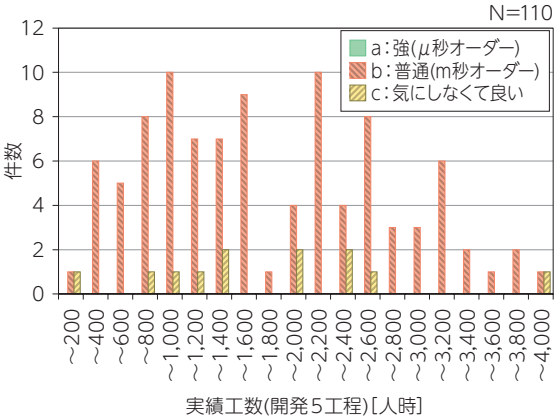
5.3.2.1　リアルタイム性（時間制約）別の工数

リアルタイム性(時間制約)別に工数データの分布及び基本統計量を示す。

図表5.3-4　リアルタイム性（時間制約）別工数の分布（改良（派生）開発、開発5工程、全体、2,000人時刻み）



図表5.3-5　リアルタイム性（時間制約）別工数の分布（改良（派生）開発、開発5工程、4,000人時以下、200人時刻み）



図表5.3-6　リアルタイム性（時間制約）別工数の基本統計量（改良（派生）開発、開発5工程）

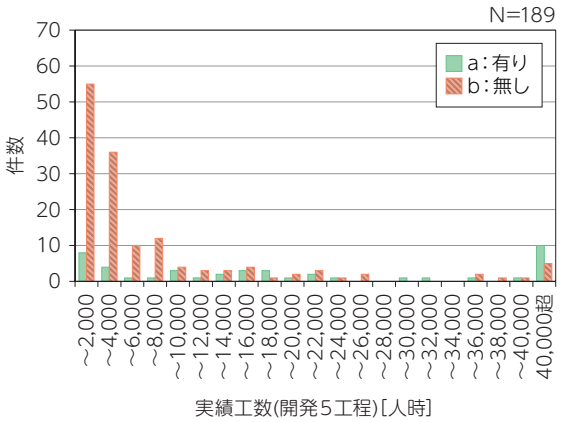
[人時]

リアルタイム性（時間制約）	N	最小	P25	中央	P75	最大	平均	標準偏差
a：強（μ秒オーダー）	5	6,770	11,135	12,563	13,542	42,211	17,244	12,696
b：普通（m秒オーダー）	179	142	1,432	3,082	14,462	109,410	11,786	18,667
c：気にしなくて良い	15	35	1,217	2,000	3,265	20,410	3,383	4,766

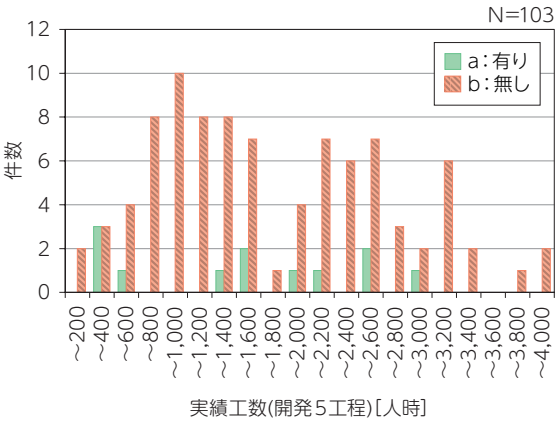
5.3.2.2　自然環境からの影響度合い別の工数

自然環境からの影響度合い別に工数データの分布及び基本統計量を示す。

図表5.3-7　自然環境からの影響度合い別工数の分布（改良（派生）開発、開発5工程、全体、2,000人時刻み）



図表5.3-8　自然環境からの影響度合い別工数の分布（改良（派生）開発、開発5工程、4,000人時以下、200人時刻み）



図表5.3-9　自然環境からの影響度合い別工数の基本統計量（改良（派生）開発、開発5工程）

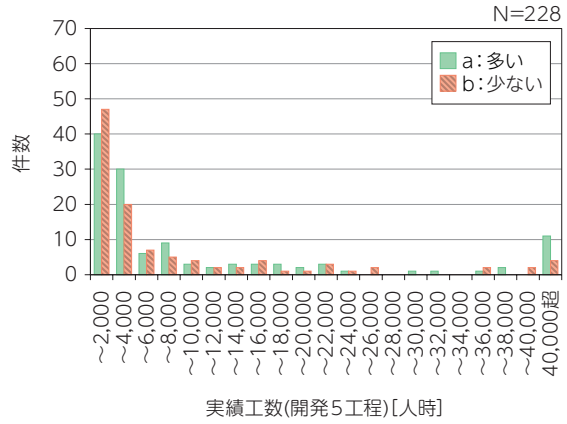
[人時]

自然環境からの影響度合い	N	最小	P25	中央	P75	最大	平均	標準偏差
a：有り	44	262	2,839	15,811	36,079	109,410	24,296	26,255
b：無し	145	35	1,302	2,575	6,889	76,704	7,582	12,728

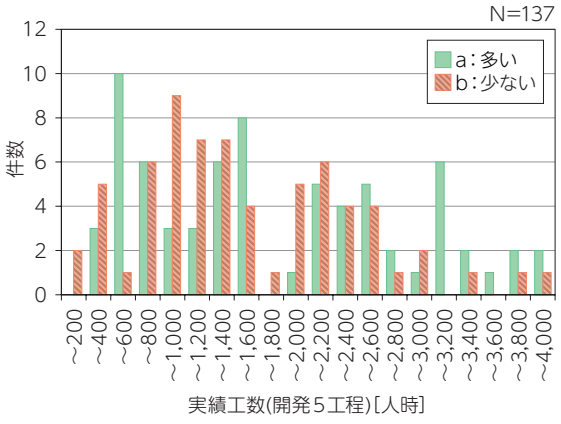
5.3.2.3 ユーザの多様性別の工数

ユーザの多様性別に工数データの分布及び基本統計量を示す。

図表5.3-10 ユーザの多様性別工数の分布（改良（派生）開発、開発5工程、全体、2,000人時刻み）



図表5.3-11 ユーザの多様性別工数の分布（改良（派生）開発、開発5工程、4,000人時以下、200人時刻み）



図表5.3-12 ユーザの多様性別工数の基本統計量（改良（派生）開発、開発5工程）

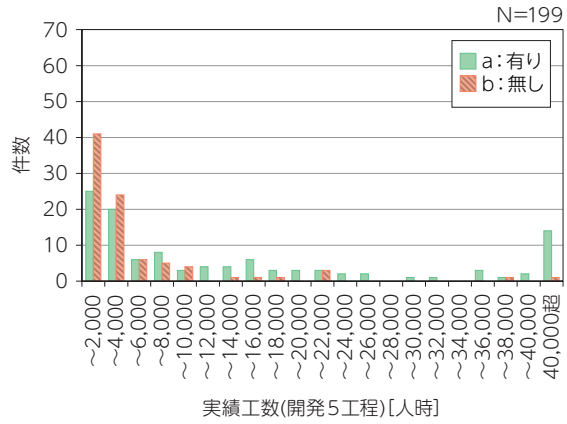
[人時]

ユーザの多様性	N	最小	P25	中央	P75	最大	平均	標準偏差
a：多い	121	269	1,392	3,082	12,563	109,410	11,823	19,840
b：少ない	107	35	1,125	2,209	8,273	76,704	8,086	13,207

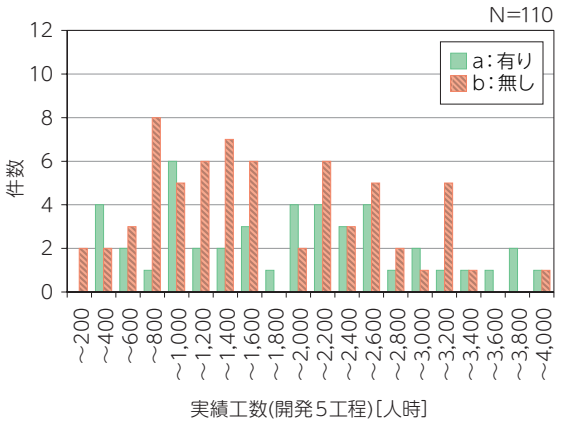
5.3.2.4 法規等による規制度合い別の工数

法規等による規制度合い別に工数データの分布及び基本統計量を示す。

図表5.3-13 法規等による規制度合い別工数の分布（改良（派生）開発、開発5工程、全体、2,000人時刻み）



図表5.3-14 法規等による規制度合い別工数の分布（改良（派生）開発、開発5工程、4,000人時以下、200人時刻み）



図表5.3-15 法規等による規制度合い別工数の基本統計量（改良（派生）開発、開発5工程）

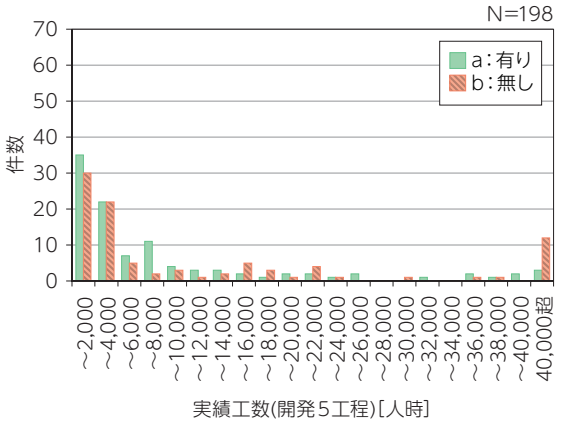
[人時]

法規等による規制度合い	N	最小	P25	中央	P75	最大	平均	標準偏差
a：有り	111	262	2,176	6,889	21,340	109,410	16,525	21,540
b：無し	88	35	1,095	2,106	4,374	62,640	4,686	8,461

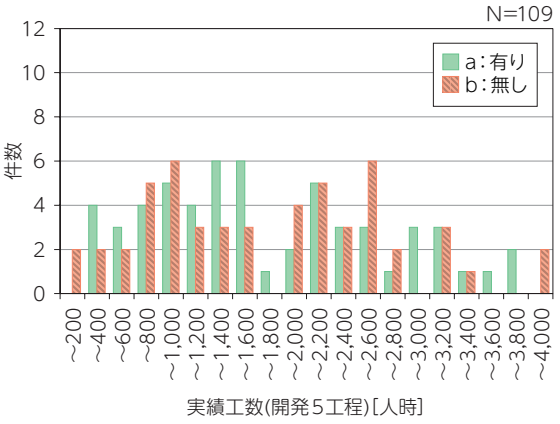
5.3.2.5 M2Mの有無別の工数

M2Mの有無別に工数データの分布及び基本統計量を示す。

図表5.3-16 M2Mの有無別工数の分布（改良（派生）開発、開発5工程、全体、2,000人時刻み）



図表5.3-17 M2Mの有無別工数の分布（改良（派生）開発、開発5工程、4,000人時以下、200人時刻み）



図表5.3-18 M2Mの有無別工数の基本統計量（改良（派生）開発、開発5工程）

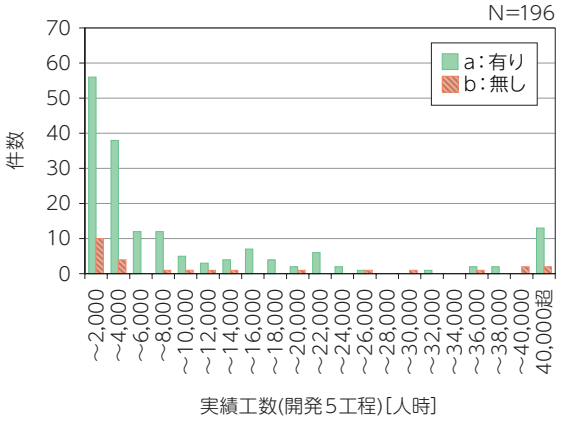
[人時]

M2Mの有無	N	最小	P25	中央	P75	最大	平均	標準偏差
a：有り	104	262	1,430	3,193	8,747	76,704	8,699	12,654
b：無し	94	35	1,431	3,049	15,991	109,410	14,265	22,216

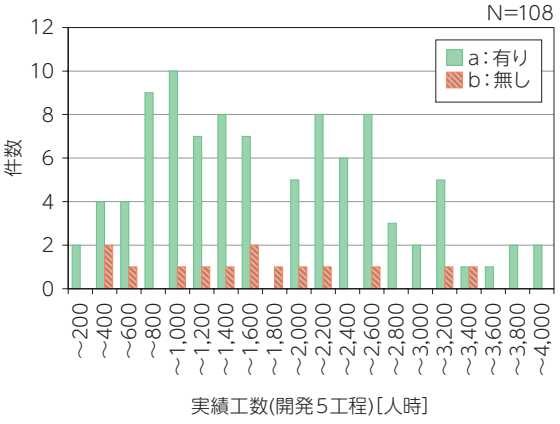
5.3.2.6 ネットワーク接続の有無別の工数

ネットワーク接続の有無別に工数データの分布及び基本統計量を示す。

図表5.3-19 ネットワーク接続の有無別工数の分布（改良（派生）開発、開発5工程、全体、2,000人時刻み）



図表5.3-20 ネットワーク接続の有無別工数の分布（改良（派生）開発、開発5工程、4,000人時以下、200人時刻み）



図表5.3-21 ネットワーク接続の有無別工数の基本統計量（改良（派生）開発、開発5工程）

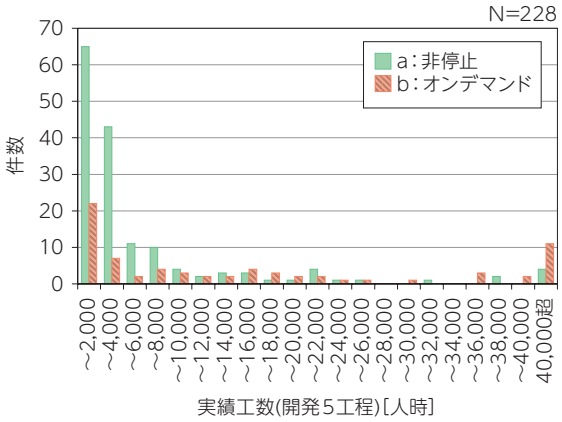
[人時]

ネットワーク接続の有無	N	最小	P25	中央	P75	最大	平均	標準偏差
a：有り	170	35	1,387	3,051	12,988	109,410	10,838	17,704
b：無し	26	326	1,434	3,241	23,232	76,704	15,033	20,415

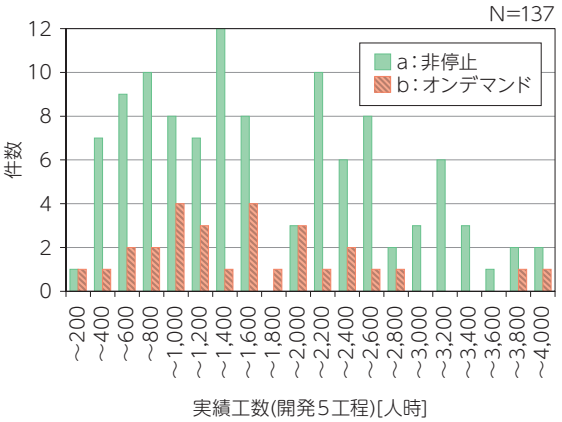
5.3.2.7 稼動（非停止、オンデマンド）別の工数

稼動（非停止、オンデマンド）別に工数データの分布及び基本統計量を示す。

図表5.3-22 稼動（非停止、オンデマンド）別工数の分布（改良（派生）開発、開発5工程、全体、2,000人時刻み）



図表5.3-23 稼動（非停止、オンデマンド）別工数の分布（改良（派生）開発、開発5工程、4,000人時以下、200人時刻み）



図表5.3-24 稼動（非停止、オンデマンド）別工数の基本統計量（改良（派生）開発、開発5工程）

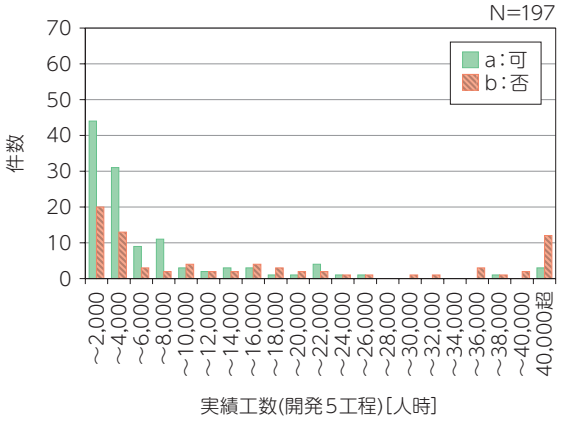
[人時]

稼動（非停止、オンデマンド）	N	最小	P25	中央	P75	最大	平均	標準偏差
a：非停止	156	142	1,127	2,330	5,600	65,130	6,035	10,490
b：オンデマンド	72	35	1,614	8,759	23,096	109,410	18,810	24,118

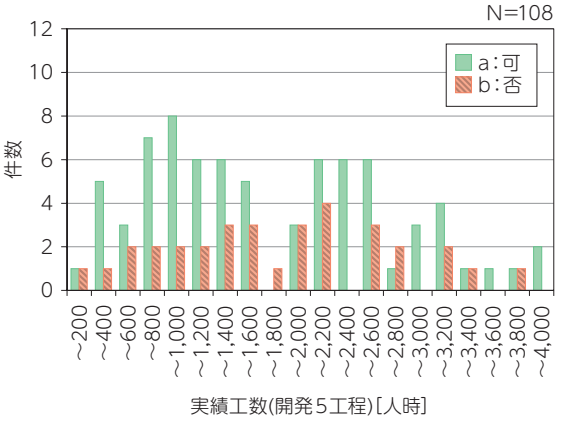
5.3.2.8 オンライン保守の可否別の工数

オンライン保守の可否別に工数データの分布及び基本統計量を示す。

図表5.3-25 オンライン保守の可否別工数の分布（改良（派生）開発、開発5工程、全体、2,000人時刻み）



図表5.3-26 オンライン保守の可否別工数の分布（改良（派生）開発、開発5工程、4,000人時以下、200人時刻み）



図表5.3-27 オンライン保守の可否別工数の基本統計量（改良（派生）開発、開発5工程）

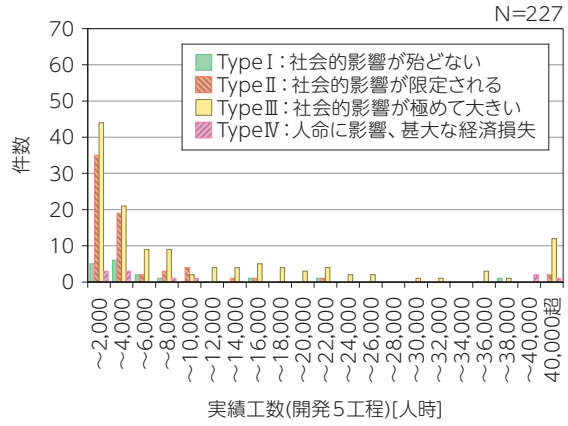
[人時]

オンライン保守の可否	N	最小	P25	中央	P75	最大	平均	標準偏差
a：可	118	142	1,202	2,548	6,783	65,130	6,337	9,965
b：否	79	35	2,040	8,784	26,565	109,410	18,944	23,929

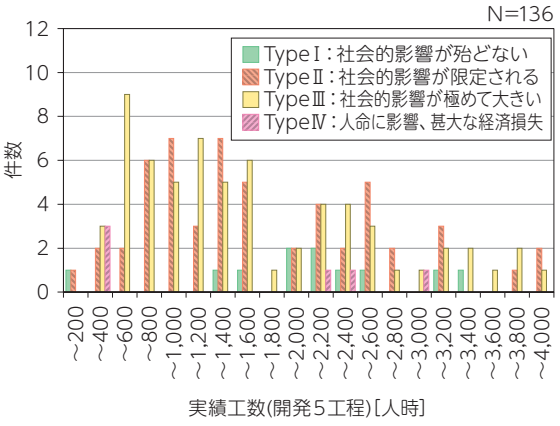
5.3.3 障害リスク別の工数：改良（派生）開発、開発5工程

障害リスク（Type）別に工数データの分布及び基本統計量を示す。

図表5.3-28 障害リスク（Type）別工数の分布（改良（派生）開発、開発5工程、全体、2,000人時刻み）



図表5.3-29 障害リスク（Type）別工数の分布（改良（派生）開発、開発5工程、4,000人時以下、200人時刻み）



図表5.3-30 障害リスク（Type）別工数の基本統計量（改良（派生）開発、開発5工程）

[人時]

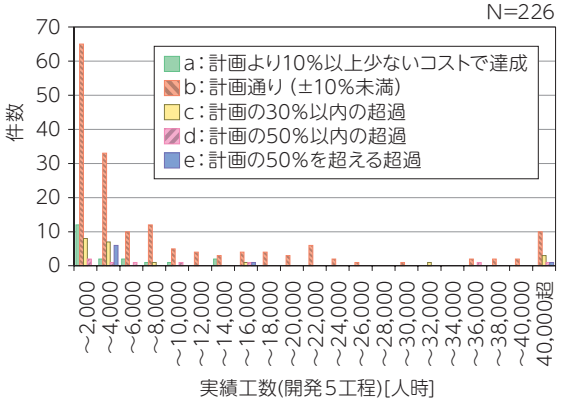
障害リスク	N	最小	P25	中央	P75	最大	平均	標準偏差
TypeⅠ：社会的影響が殆どない	17	35	1,902	2,600	4,500	36,973	6,524	9,260
TypeⅡ：社会的影響が限定される	68	142	953	1,966	3,301	65,130	4,882	10,902
TypeⅢ：社会的影響が極めて大きい	131	269	1,372	4,284	15,961	109,410	12,972	19,671
TypeⅣ：人命に影響、甚大な経済損失	11	262	1,274	2,940	23,959	48,000	13,693	17,872

5.3.4 実績の評価別の工数：改良（派生）開発、開発5工程

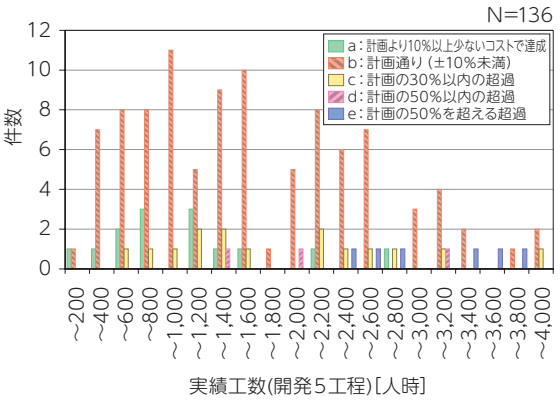
5.3.4.1 コスト実績の評価別の工数

コスト実績の評価別に工数データの分布及び基本統計量を示す。

図表5.3-31 コスト実績の評価別工数の分布（改良（派生）開発、開発5工程、全体、2,000人時刻み）



図表5.3-32 コスト実績の評価別工数の分布（改良（派生）開発、開発5工程、4,000人時以下、200人時刻み）



図表5.3-33 コスト実績の評価別工数の基本統計量（改良（派生）開発、開発5工程）

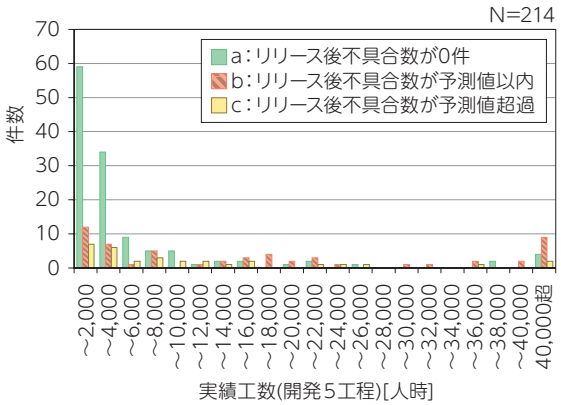
[人時]

コスト実績の評価	N	最小	P25	中央	P75	最大	平均	標準偏差
a：計画より10%以上少ないコストで達成	20	35	695	1,240	4,431	13,542	3,284	4,018
b：計画通り（±10%未満）	169	142	1,331	2,575	11,135	68,260	9,593	14,751
c：計画の30%以内の超過	21	470	1,354	2,318	6,770	109,410	16,769	31,470
d：計画の50%以内の超過	8	1,254	2,772	7,175	19,944	76,704	18,332	24,290
e：計画の50%を超える超過	8	2,245	2,651	3,434	6,692	48,000	10,234	14,888

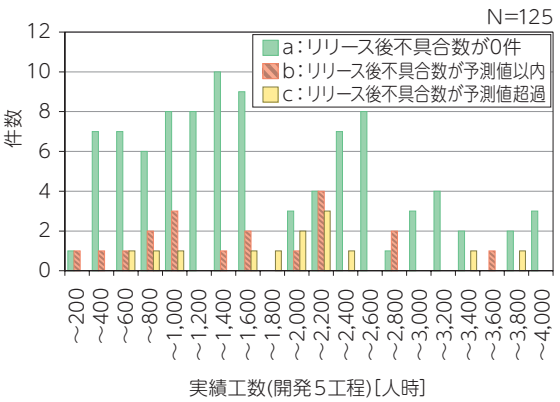
5.3.4.2 品質実績の評価別の工数

品質実績の評価別に工数データの分布及び基本統計量を示す。

図表5.3-34 品質実績の評価別工数の分布（改良（派生）開発、開発5工程、全体、2,000人時刻み）



図表5.3-35 品質実績の評価別工数の分布（改良（派生）開発、開発5工程、4,000人時以下、200人時刻み）



図表5.3-36 品質実績の評価別工数の基本統計量（改良（派生）開発、開発5工程）

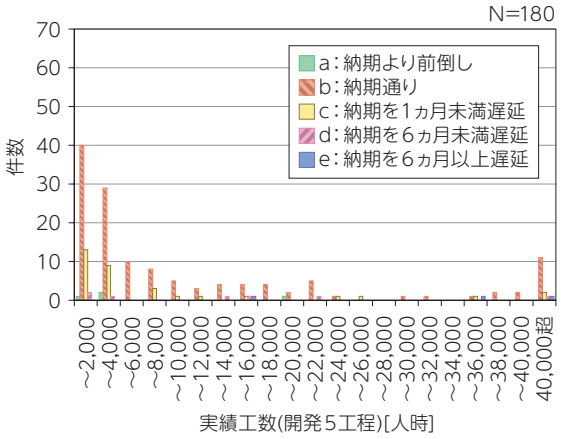
[人時]

品質実績の評価	N	最小	P25	中央	P75	最大	平均	標準偏差
a：リリース後不具合数が0件	127	142	1,038	2,209	4,416	62,640	5,732	10,971
b：リリース後不具合数が予測値以内	56	35	2,080	13,731	28,971	109,410	21,486	25,553
c：リリース後不具合数が予測値超過	31	470	2,152	6,787	13,776	48,000	10,807	12,166

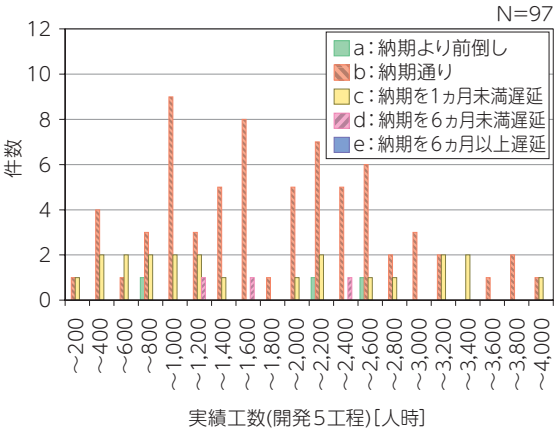
5.3.4.3 工期実績の評価別の工数

工期実績の評価別に工数データの分布及び基本統計量を示す。

図表5.3-37 工期実績の評価別工数の分布（改良（派生）開発、開発5工程、全体、2,000人時刻み）



図表5.3-38 工期実績の評価別工数の分布（改良（派生）開発、開発5工程、4,000人時以下、200人時刻み）



図表5.3-39 工期実績の評価別工数の基本統計量（改良（派生）開発、開発5工程）

[人時]

工期実績の評価	N	最小	P25	中央	P75	最大	平均	標準偏差
a：納期より前倒し	4	718	1,739	2,309	6,667	19,056	6,098	7,511
b：納期通り	133	35	1,570	3,626	14,333	109,410	12,228	19,138
c：納期を１ヵ月未満遅延	33	142	956	2,668	7,851	65,130	8,820	14,752
d：納期を６ヵ月未満遅延	7	1,010	1,879	12,563	17,966	48,000	14,466	15,430
e：納期を６ヵ月以上遅延	3	15,853	25,047	34,240	55,472	76,704	42,266	25,482

6章　工数、工期、規模の関係の分析

6.1　位置付け 80

6.2　工数と工期 81

6.2.1　開発言語別の工数と工期：改良（派生）開発、開発5工程

6.2.2　開発言語C/C++の工数と工期：改良（派生）開発、開発5工程

6.3　SLOC 規模と工数 83

6.3.1　開発言語別のSLOC規模と工数：改良（派生）開発、開発5工程

6.3.2　開発言語C/C++のSLOC規模と工数：改良（派生）開発、開発5工程

6.4　工程別の工期、工数 85

6.4.1　工程別工期：改良（派生）開発

6.4.2　工程別工数：改良（派生）開発

6章　工数、工期、規模の関係の分析

6.1　位置付け

この章では、工数と工期の関係、SLOC規模と工数の関係、ならびに工期や工数の工程別比率を組込みシステム全体で把握するための分析を行った。

図表6.1-1に、分析対象と層別のパターンを示す。

図表6.1-2には、要素データ“SLOC規模”と“工数”それぞれの分布状況の参照先を示す。

図表6.1-1　分析対象と層別のパターン

分析対象		層別のパターン				項番号
		開発種別	開発工程	開発言語	その他層別	
工数	工期	改良（派生）開発	開発５工程	全言語	開発言語別	6.2.1
				C/C++	−	6.2.2
SLOC規模	工数			全言語	開発言語別	6.3.1
				C/C++	−	6.3.2
工程別工期				全言語	−	6.4.1
工程別工数				全言語	−	6.4.2

図表6.1-2　主な要素データと参照先の節番号

要素データ	参照先の節番号
SLOC規模	5.2
工数	5.3

6.2　工数と工期

この節では、工数と工期の関係を示す。本節で使用するデータのうち、その名称に「導出指標」と付記されたものについては、付録A.3にてその定義や導出方法を説明する。

工数と工期は、3.2.1項「データ項目の取り扱いに関する取り決め」で記載した開発5工程の工数とその期間を対象とする。

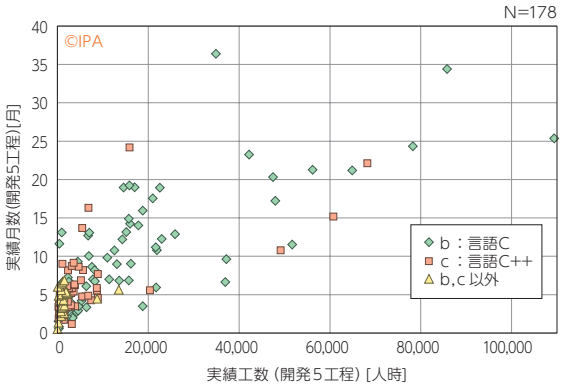
6.2.1　開発言語別の工数と工期：改良（派生）開発、開発5工程

ここでは、改良（派生）開発のプロジェクトを対象に、実績工数とその工期（月数）の関係を、開発言語ごとに示す。

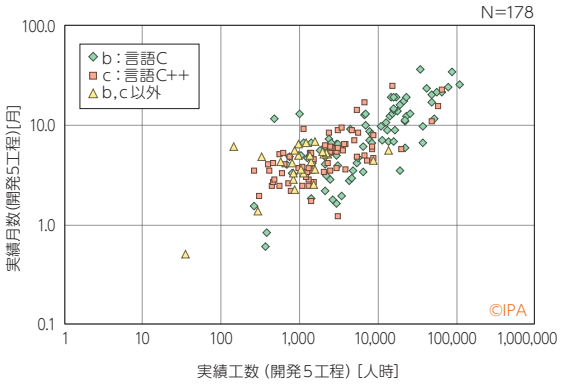
■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1が回答されているもの
- 実績工数（開発5工程）＞0
- 実績月数（開発5工程）＞0

■対象データ
- X軸：実績工数（開発5工程）（導出指標）
- Y軸：実績月数（開発5工程）（導出指標）

図表6.2-1　開発言語別の工数と工期（改良（派生）開発、開発5工程）



図表6.2-2　開発言語別の工数と工期（改良（派生）開発、開発5工程）対数表示



※開発言語別の工数と工期について、対数化した場合の相関係数は次のようになる。
- b：言語C　R=0.76（P<0.05）
- c：言語C++　R=0.68（P<0.05）
- b,c以外　R=0.59（P<0.05）

図表6.2-3　開発言語別の実績月数の基本統計量（改良（派生）開発、開発5工程）

[月]

開発言語	N	P25	中央	P75
b：言語C	85	4.20	6.83	12.90
c：言語C++	67	3.37	4.73	5.98
b,c以外	26	3.39	4.67	5.33

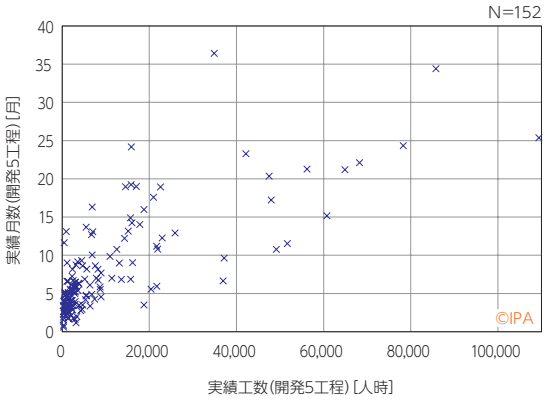
6.2.2　開発言語C/C++の工数と工期：改良（派生）開発、開発5工程

ここでは、改良（派生）開発で言語Cと言語C++のプロジェクトを対象に、開発5工程での実績工数とその工期（月数）の関係を示す。

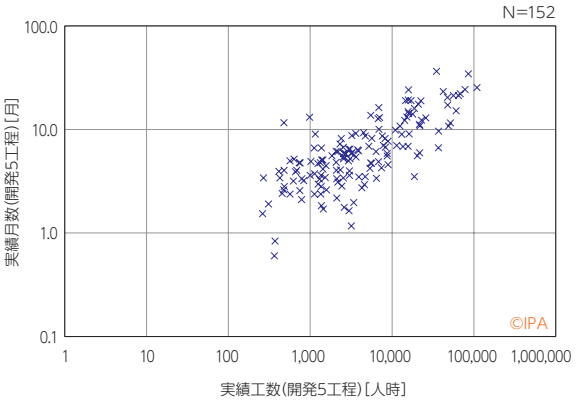
■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 実績工数（開発5工程）＞0
- 実績月数（開発5工程）＞0

■対象データ
- X軸：実績工数（開発5工程）（導出指標）
- Y軸：実績月数（開発5工程）（導出指標）

図表6.2-4　開発言語C/C++の工数と工期（改良（派生）開発、開発5工程、開発言語C/C++）



図表6.2-5　開発言語C/C++の工数と工期（改良（派生）開発、開発5工程、開発言語C/C++）対数表示



※開発言語C/C++の工数と工期について、対数化した場合の相関係数は次のようになる。
R=0.75（P<0.05）

図表6.2-6　開発言語C/C++の実績月数の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++）

［月］

N	P25	中央	P75
152	3.50	5.68	9.68

6.3　SLOC規模と工数

この節では、SLOC規模と工数の関係を示す。本節で使用するデータのうち、その名称に（導出指標）と付記されたものについては、付録A.3にてその定義や導出方法を説明する。

SLOC規模と工数の関係から生産性が分かるため、工数は、3.2.1項「データ項目の取り扱いに関する取り決め」で記載した通り開発5工程の工数を使用する。

本節は、7.2節「SLOC生産性」と対で見るとよい。

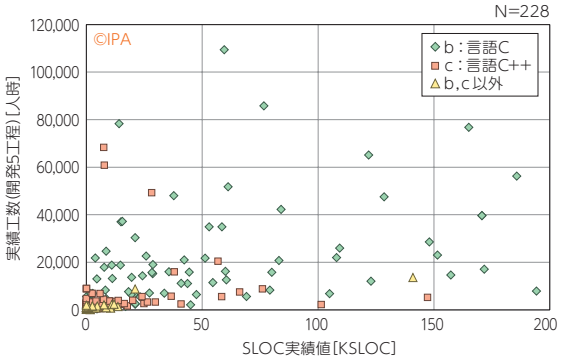
6.3.1　開発言語別のSLOC規模と工数：改良（派生）開発、開発5工程

ここでは、改良（派生）開発のプロジェクトを対象に、SLOC規模と工数の関係を開発言語別に示す。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1が回答されているもの
- 実効SLOC実績値 ＞ 0
- 実績工数（開発5工程）＞ 0

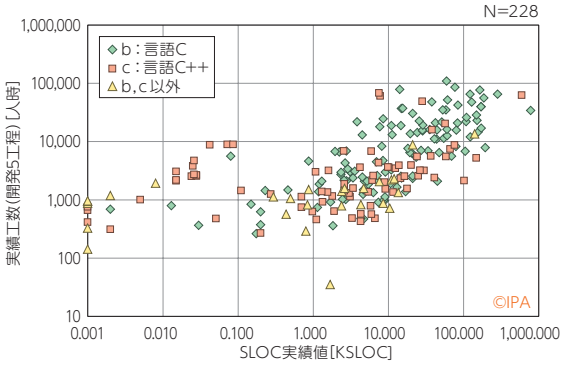
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：実績工数（開発5工程）（導出指標）

図表6.3-1　開発言語別のSLOC規模と工数（改良（派生）開発、開発5工程）



※表示されていないものが3点ある

図表6.3-2　開発言語別のSLOC規模と工数（改良（派生）開発、開発5工程）対数表示



※開発言語別のSLOC規模と工数について、対数化した場合の相関係数は次のようになる。
b：言語C　　R=0.72（P<0.05）
c：言語C++　　R=0.39（P<0.05）
b,c以外　　R=0.40（P<0.05）

図表6.3-3　開発言語別の実績工数の基本統計量（改良（派生）開発、開発5工程）

[人時]

開発言語	N	P25	中央	P75
b：言語C	123	1,991	6,440	18,810
c：言語C++	78	1,138	2,370	3,925
b,c以外	27	803	1,056	1,572

●ここでは、言語Cと言語C++の違いによりSLOC規模と工数の分布に違う傾向が見えている。開発言語は、一般に、製品の特性や特定の機能などの用途に応じて使い分けられている。そのため、SLOC規模と工数の分布に違う傾向が見える要因は、言語能力の違いよりも製品の特性や特定の機能などの用途に依存すると考えられる。なお、7章以降の生産性や信頼性の分析では、言語Cと言語C++ の違いは追究せずに２つの言語を混在させて分析している。

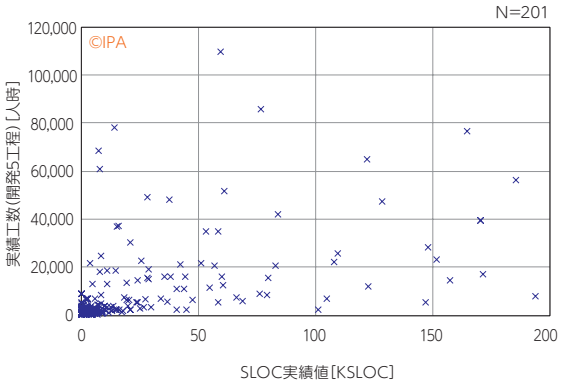
6.3.2 開発言語C/C++のSLOC規模と工数：改良（派生）開発、開発5工程

ここでは、改良（派生）開発で言語Cと言語C++のプロジェクトを対象に、SLOC規模と工数の関係を示す。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 実効SLOC実績値 ＞ 0
- 実績工数（開発5工程）＞ 0

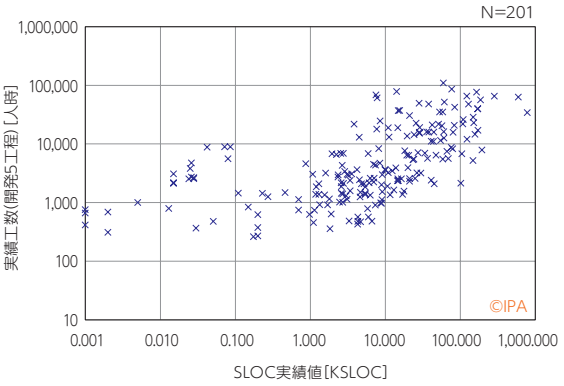
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：実績工数（開発5工程）（導出指標）

図表6.3-4 開発言語C/C++のSLOC規模と工数（改良（派生）開発、開発5工程、開発言語C/C++）



※表示されていないものが3点ある

図表6.3-5 開発言語C/C++のSLOC規模と工数（改良（派生）開発、開発5工程、開発言語C/C++）対数表示



※開発言語C/C++のSLOC規模と工数について、対数化した場合の相関係数は次のようになる。
R=0.60（P<0.05）

図表6.3-6 開発言語C/C++の実績工数の基本統計量（改良(派生)開発、開発5工程、開発言語C/C++）

[人時]

N	P25	中央	P75
201	1,431	3,193	12,960

6.4　工程別の工期、工数

本節では、工期、工数が各工程に配分された比率を示す。対象工程は、アーキテクチャ設計から総合テストまでの開発5工程とする。

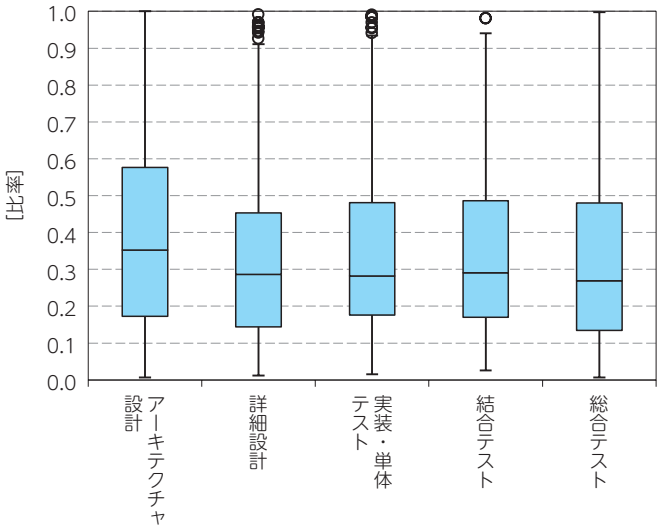
6.4.1　工程別工期：改良（派生）開発

ここでは、開発5工程における、改良（派生）開発の工程別の実績月数の比率を示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 開発5工程について、各工程の実績月数にすべて記入があり、各月数が0より大きい

■対象データ
- 8-2-3-2_アーキテクチャ設計_月数、
  8-2-3-3_詳細設計_月数、
  8-2-3-4_実装・単体テスト_月数、
  8-2-3-5_結合テスト_月数、
  8-2-3-6_総合テスト_月数
- 実績月数（開発５工程）
  （総合テスト終了年月［日］－アーキテクチャ設計開始年月［日］）（導出指標）

図表6.4-1　工程別の実績月数の比率（改良（派生）開発）箱ひげ図



図表6.4-2　工程別の実績月数の比率の基本統計量（改良（派生）開発）

［比率］

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	208	0.01	0.17	0.35	0.58	1.00	0.40	0.27
詳細設計	208	0.01	0.14	0.29	0.45	0.99	0.34	0.26
実装・単体テスト	208	0.02	0.18	0.28	0.48	0.99	0.35	0.25
結合テスト	208	0.03	0.17	0.29	0.49	0.98	0.35	0.23
総合テスト	208	0.01	0.13	0.27	0.48	1.00	0.34	0.26

●改良、派生開発のアーキテクチャ設計工程では、母体への影響範囲の調査に時間をかけていると見られ、詳細設計や実装より工期が長く取られている。

6.4.2　工程別工数：改良(派生)開発

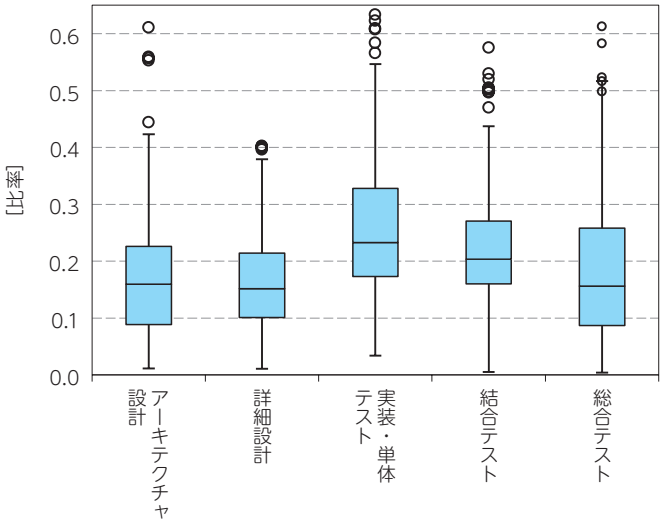
ここでは、開発5工程における改良（派生）開発の工程別の実績工数の比率を示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 工程別の実績工数にすべて記入があり、各値が0より大きい

■対象データ
- 9-4-30_社内実績工数_アーキテクチャ設計、
  9-4-31_社内実績工数_詳細設計、
  9-4-32_社内実績工数_実装・単体テスト、
  9-4-33_社内実績工数_結合テスト、
  9-4-34_社内実績工数_総合テスト
- 9-8-2_外部委託工数_アーキテクチャ設計、
  9-8-3_外部委託工数_詳細設計、
  9-8-4_外部委託工数_実装・単体テスト、
  9-8-5_外部委託工数_結合テスト、
  9-8-6_外部委託工数_総合テスト

※各工程の実績工数は、社内、外部委託の実績工数合計の人時換算値を使用。

図表6.4-3　工程別の実績工数の比率（改良（派生）開発）箱ひげ図



図表6.4-4　工程別の実績工数の比率の基本統計量（改良（派生）開発）

[比率]

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	228	0.01	0.09	0.16	0.23	0.68	0.17	0.11
詳細設計	228	0.01	0.10	0.15	0.21	0.67	0.16	0.09
実装・単体テスト	228	0.03	0.17	0.23	0.33	0.79	0.26	0.12
結合テスト	228	0.01	0.16	0.20	0.27	0.58	0.22	0.10
総合テスト	228	0.00	0.09	0.16	0.26	0.61	0.18	0.13

●工程別の工数比率の傾向をエンタプライズ系と比べると（参考文献 [2]”ソフトウェア開発データ白書2018-2019”）、実装・単体テストにかける工数比率が小さい。

7章　生産性の分析

7.1　位置付け 88

7.2　SLOC生産性 89

7.2.1　開発言語別のSLOC生産性：改良（派生）開発、開発5工程

7.2.2　開発言語C/C++のSLOC生産性：
改良（派生）開発、開発5工程、開発言語C/C++

7.3　工程別SLOC生産性 92

7.3.1　工程別SLOC生産性：
改良（派生）開発、開発5工程、開発言語C/C++

7章　生産性の分析

7.1　位置付け

この章では、SLOC生産性に関する傾向を組込みシステム全体で把握するための分析を行った。「SLOC生産性」は、SLOC規模を開発5工程の工数で除算したものである。すなわち、人時あたりのSLOC規模、または人月（人時への変換は9-2_人時換算係数を使用）あたりのSLOC規模である。

図表7.1-1に分析対象と層別のパターンを示す。
図表7.1-2には、要素データ“SLOC規模”と“工数”それぞれの分布状況の参照先を示す。

図表7.1-1　分析対象と層別のパターン

分析対象		層別のパターン				項番号
		開発種別	開発工程	開発言語	その他層別	
SLOC規模	SLOC生産性	改良（派生）開発	開発5工程	全言語	開発言語別	7.2.1
				C/C++	−	7.2.2.1
				C/C++	母体含むSLOC規模別	7.2.2.2
工程別SLOC生産性				C/C++	−	7.3.1

図表7.1-2　主な要素データと参照先の節番号

要素データ	参照先の節番号
SLOC規模	5.2
工数	5.3

7.2 SLOC生産性

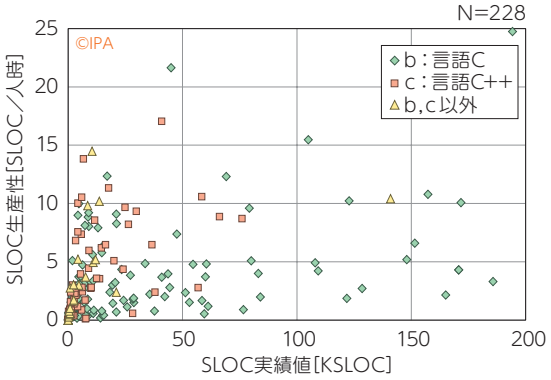
7.2.1 開発言語別のSLOC生産性：改良（派生）開発、開発5工程

ここでは、改良（派生）開発プロジェクトを対象に、SLOC規模とSLOC生産性の関係を開発言語別に示す。開発言語は、収集データ件数が多い言語C、C++、それら以外で分類する。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1が回答されているもの
- 実効SLOC実績値 ＞ 0
- SLOC生産性 ＞ 0

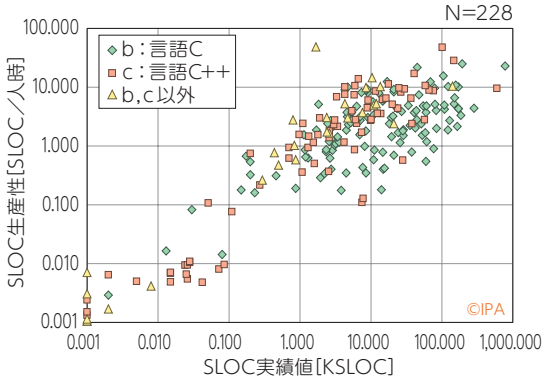
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：SLOC生産性
  （SLOC／実績工数（開発5工程））（導出指標）

図表7.2-1 開発言語別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程）



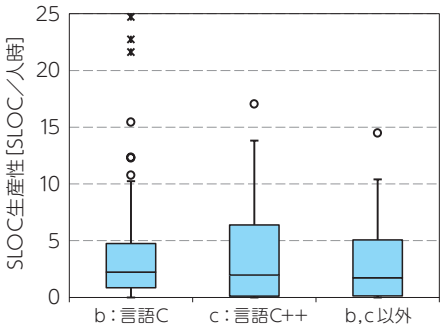
※表示されていないものが6点ある

図表7.2-2 開発言語別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程）対数表示



※開発言語別のSLOC規模とSLOC生産性について、対数化した場合の相関係数は次のようになる。
b：言語C　R=0.73（P<0.05）
c：言語C++　R=0.92（P<0.05）
b,c以外　R=0.95（P<0.05）

図表7.2-3 開発言語別SLOC生産性（改良（派生）開発、開発5工程）箱ひげ図



図表7.2-4 開発言語別SLOC生産性の基本統計量（改良（派生）開発、開発5工程）

[SLOC／人時]

開発言語	N	P25	中央	P75
b：言語C	123	0.85	2.22	4.76
c：言語C++	78	0.11	1.98	6.39
b,c以外	27	0.14	1.72	5.07

● 開発言語の違いにより、SLOC規模と生産性の分布に違う傾向が見えている。開発言語は、一般に、製品の特性や特定の機能などの用途に応じて使い分けられているため、SLOC規模と生産性の分布に違う傾向が見える要因は、言語能力の違いよりも製品の特性や特定の機能などの用途に依存すると考えられる。そのため、本章の分析では、言語Cと言語C++の違いは追究せずに2つの言語を混在させて分析している。

7.2.2 開発言語C/C++のSLOC生産性：改良（派生）開発、開発5工程、開発言語C/C++

ここでは、改良（派生）開発で言語Cと言語C++のプロジェクトを対象に、SLOC生産性を示す。

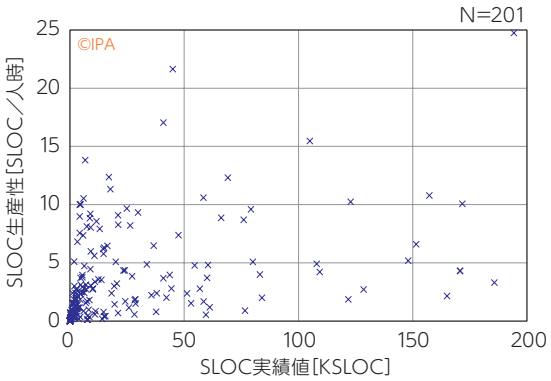
7.2.2.1 SLOC規模とSLOC生産性

ここでは、SLOC規模とSLOC生産性の関係をSLOC規模別に示す。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良(派生)開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 実効SLOC実績値 ＞ 0
- SLOC生産性 ＞ 0

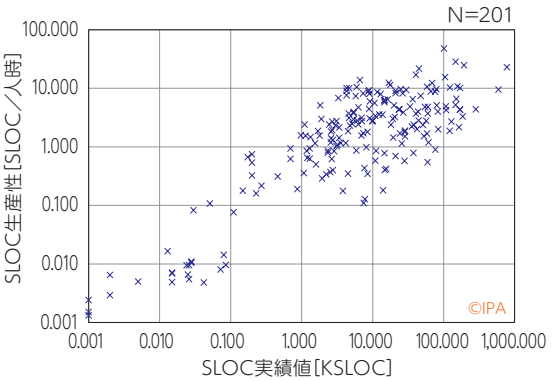
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：SLOC生産性
  （SLOC／実績工数（開発5工程））（導出指標）

図表7.2-5 SLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++）



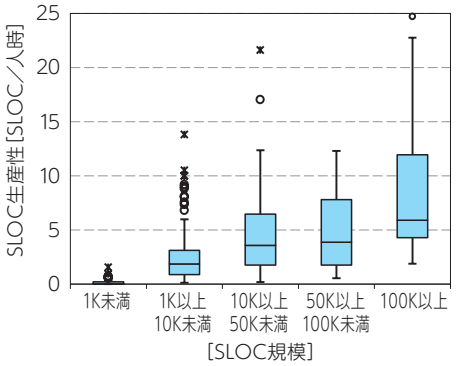
※表示されていないものが5点ある

図表7.2-6 SLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++）対数表示



※開発言語C/C++のSLOC規模とSLOC生産性について、
対数化した場合の相関係数は次のようになる。
R=0.85（P＜0.05）

図表7.2-7 SLOC規模別SLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++）箱ひげ図



図表7.2-8 SLOC規模別SLOC生産性の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++）

[SLOC／人時]

SLOC規模	N	P25	中央	P75
全体	201	0.63	2.15	4.92
1K未満	35	0.01	0.01	0.20
1K以上10K未満	77	0.86	1.85	3.12
10K以上50K未満	51	1.75	3.58	6.46
50K以上100K未満	18	1.75	3.86	7.80
100K以上	20	4.29	5.89	11.95

●生産性は、SLOC規模が大きくなると、高くなる傾向が見られる。

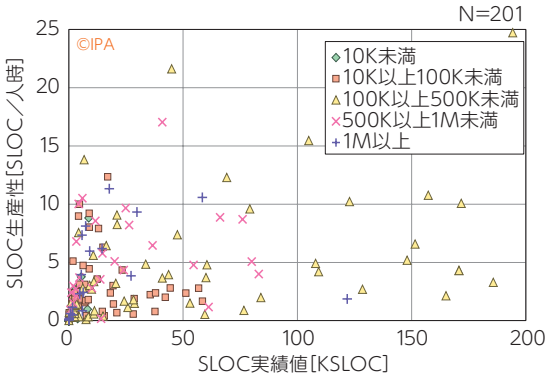
7.2.2.2 母体SLOC規模別のSLOC規模とSLOC生産性

ここでは、SLOC規模とSLOC生産性の関係を母体含むSLOC規模別に示す。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 実効SLOC実績値 ＞ 0
- 7-6_SLOC 実績値（母体）＞ 0
- SLOC生産性 ＞ 0

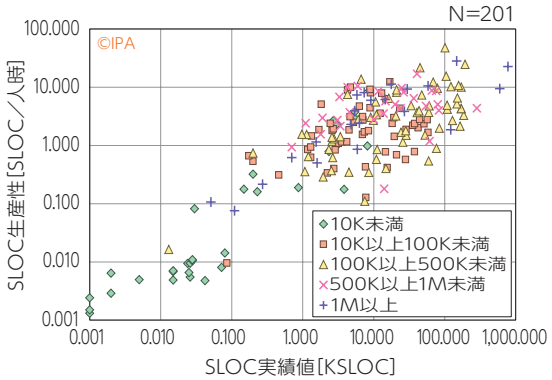
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：SLOC生産性
  （SLOC／実績工数（開発5工程））（導出指標）

図表7.2-9 母体SLOC規模別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++）



※表示されていないものが5点ある

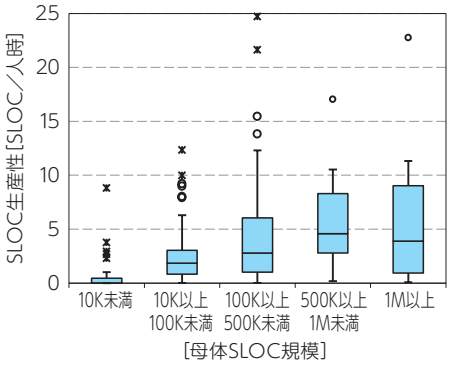
図表7.2-10 母体SLOC規模別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++）対数表示



※母体SLOC規模別のSLOC規模とSLOC生産性について、対数化した場合の相関係数は次のようになる。

10K未満	R=0.94（P<0.05）
10K以上100K未満	R=0.46（P<0.05）
100K以上500K未満	R=0.68（P<0.05）
500K以上1M未満	R=0.31（P≧0.05）
1M以上	R=0.87（P<0.05）

図表7.2-11 母体SLOC規模別SLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++）箱ひげ図



図表7.2-12 母体SLOC規模別SLOC生産性の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++）

[SLOC／人時]

母体SLOC規模	N	P25	中央	P75
10K未満	32	0.01	0.01	0.44
10K以上100K未満	56	0.83	1.86	3.03
100K以上500K未満	63	1.02	2.80	6.04
500K以上1M未満	28	2.79	4.58	8.30
1M以上	22	0.93	3.90	9.03

● 母体SLOC規模別に生産性の分布（P25からP75）を比べると、母体SLOC規模が大きくなるにつれてP75は高くなり生産性の高い標本が増えている。P75が高くなるのは、母体SLOC規模が大きくなるに従い、実効SLOC実績値の大きいデータが増えるためであり、母体SLOC規模の影響を受けているとは言えない。

7.3 工程別SLOC生産性

本節では、改良(派生)開発で言語Cと言語C++のプロジェクトを対象に、開発5工程の工程ごとのSLOC生産性(SLOC規模を各工程の実績工数で除算した値)を示す。

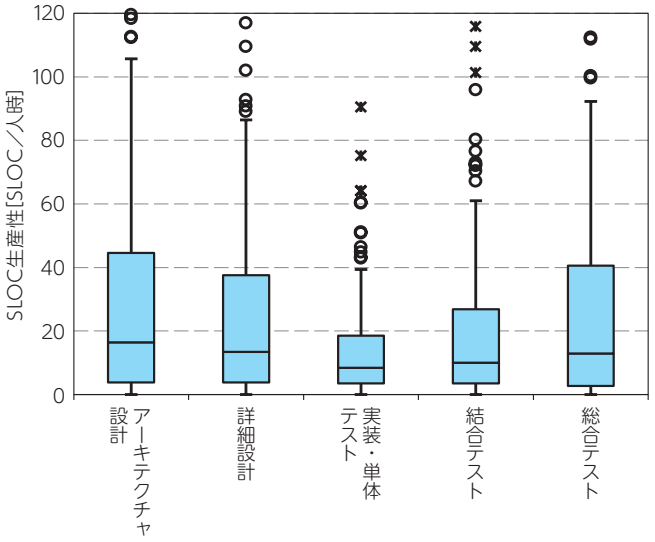
7.3.1 工程別SLOC生産性：改良(派生)開発、開発5工程、開発言語C/C++

ここでは、開発5工程(アーキテクチャ設計～総合テスト)の作業が行われたものを対象とする。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良(派生)開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 実効SLOC実績値 ＞ 0

■対象データ
- アーキテクチャ設計SLOC生産性
  (SLOC／実績工数 アーキテクチャ設計)(導出指標)
- 詳細設計SLOC生産性
  (SLOC／実績工数 詳細設計)(導出指標)
- 実装・単体テストSLOC生産性
  (SLOC／実績工数 実装・単体テスト)(導出指標)
- 結合テストSLOC生産性
  (SLOC／実績工数 結合テスト)(導出指標)
- 総合テストSLOC生産性
  (SLOC／実績工数 総合テスト)(導出指標)

図表7.3-1 工程別SLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++)箱ひげ図



図表7.3-2 工程別SLOC生産性の基本統計量(改良(派生)開発、開発5工程、開発言語C/C++)

[SLOC／人時]

工程	N	P25	中央	P75
アーキテクチャ設計	201	3.83	16.38	44.61
詳細設計	201	3.83	13.45	37.57
実装・単体テスト	201	3.49	8.38	18.53
結合テスト	201	3.47	10.03	26.84
総合テスト	201	2.69	12.92	40.53

8章　信頼性の分析

8.1　位置付け …… 94

8.2　SLOC規模と検出バグ密度 …… 95

8.2.1　結合テスト検出バグ密度

8.2.2　総合テスト検出バグ密度

8.3　テストケース数と検出バグ数、テスト工数 …… 102

8.3.1　SLOC規模あたりのテストケース数、検出バグ数：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

8.3.2　SLOC規模あたりのテスト工数：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

8.4　SLOC規模とテスト工期 …… 108

8.4.1　SLOC規模と結合テスト工期：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

8.4.2　SLOC規模と総合テスト工期：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

8.5　工数あたりのテストケース数、検出バグ数 …… 110

8.5.1　工数あたりのテストケース数、検出バグ数：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

8章　信頼性の分析

8.1　位置付け

この章では、信頼性に関する傾向を組込みシステム全体で把握するための分析を行った。

分析対象は、テスト検出バグ数（総合テスト検出バグ現象数、結合テスト検出バグ現象数）、テストケース数、テスト工数、テスト工期、SLOC規模の各要素で、各要素間の関係を分析した。図表8.1-1に、分析対象と層別のパターンを示す。

図表8.1-2には、要素データ“SLOC規模”の分布状況の参照先を示す。

テスト検出バグ数をSLOC規模で正規化した「テスト検出バグ密度」は、1,000行（1KSLOC）あたりのテスト検出バグ数（単位：件／KSLOC）で表すが、SLOC規模が1,000行（1KSLOC）以下の小さい標本の場合、1,000行で正規化したテスト検出バグ密度は、実際に検出した件数より高くなるため、本章の分析では、分析対象の標本を0.1KSLOC以上に限定している。

図表8.1-1　分析対象と層別のパターン

分析対象		層別のパターン				項番号
		開発種別	開発工程	開発言語	その他層別	
SLOC規模	テスト検出バグ密度	改良（派生）開発	－	全言語	開発言語別	8.2.1.1 8.2.2.1
			開発5工程	全言語	開発言語別	8.2.1.2 8.2.2.2
				C/C++	－	8.2.1.3 8.2.2.3
					母体含むSLOC規模別	8.2.2.4
テストケース密度	テスト検出バグ密度		開発5工程	C/C++	－	8.3.1.1 8.3.1.2
テスト工数				C/C++	－	8.3.2.1 8.3.2.2
SLOC規模	テスト工期		開発5工程	C/C++	－	8.4.1 8.4.2
工数あたりのテストケース数	工数あたりのテスト検出バグ数		開発5工程	C/C++	－	8.5.1

図表8.1-2　主な要素データと参照先の節番号

要素データ	参照先の節番号
SLOC規模	5.2

8.2 SLOC規模と検出バグ密度

ここでは、SLOC規模の実績データが計測されているプロジェクトを対象に、テスト検出バグ密度について示す。テスト検出バグ密度は、KSLOC(1,000行)あたりの検出バグ現象数とする。

8.2.1 結合テスト検出バグ密度

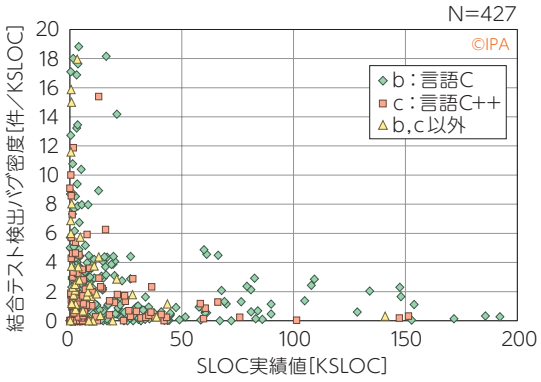
8.2.1.1 開発言語別のSLOC規模と結合テスト検出バグ密度：改良(派生)開発、0.1KSLOC以上

ここでは、改良(派生)開発プロジェクトを対象に、SLOC規模と結合テスト検出バグ密度の関係について、開発言語(C、C++、C/C++以外)別に示す。

■層別定義
- 1-5_開発プロジェクトの種別がb：改良(派生)開発
- 3-9_主開発言語1が回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- 10-3-1_結合テスト検出バグ現象数 ≧ 0

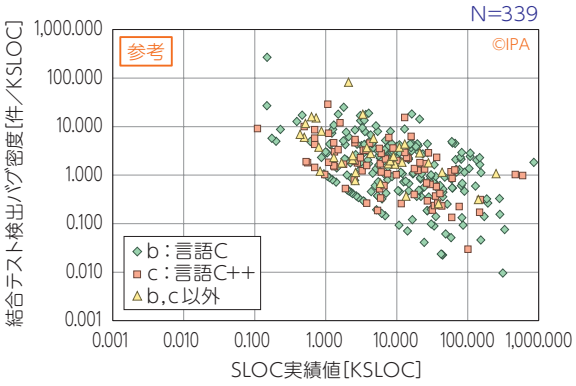
■対象データ
- X軸：実効SLOC実績値(導出指標)
- Y軸：結合テスト検出バグ密度(SLOCあたりの検出バグ数)(導出指標)

図表8.2-1 開発言語別のSLOC規模と結合テスト検出バグ密度(改良(派生)開発、0.1KSLOC以上)



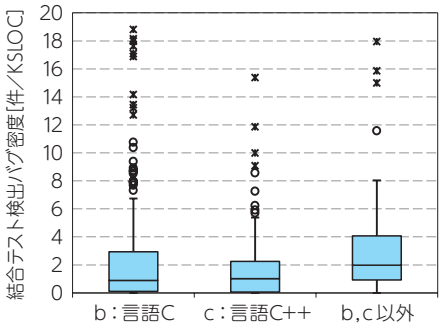
※表示されていないものが14点ある

図表8.2-2 開発言語別のSLOC規模と結合テスト検出バグ密度(改良(派生)開発、0.1KSLOC以上)対数表示



【参考】対数表示では検出バグ密度が"0件／KSLOC"より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

図表8.2-3 開発言語別結合テスト検出バグ密度(改良(派生)開発、0.1KSLOC以上)箱ひげ図



図表8.2-4 開発言語別結合テスト検出バグ密度の基本統計量(改良(派生)開発、0.1KSLOC以上)

[件／KSLOC]

開発言語	N	P25	中央	P75
b：言語C	282	0.103	0.894	2.930
c：言語C++	106	0.056	1.017	2.245
b,c以外	39	0.919	1.975	4.069

●ここでは、開発5工程に関係なく結合テストを実施しているプロジェクトを対象にしている。開発5工程すべてを実施したプロジェクトの場合は次項に示す。

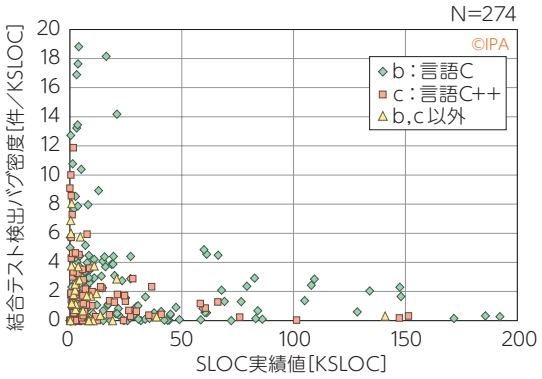
8.2.1.2 開発言語別のSLOC規模と結合テスト検出バグ密度：改良（派生）開発、開発5工程、0.1KSLOC以上

ここでは、改良（派生）開発で開発5工程のプロジェクトを対象に、SLOC規模と結合テスト検出バグ密度の関係について、開発言語（C、C++、C/C++以外）別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1が回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- 10-3-1_結合テスト検出バグ現象数 ≧ 0

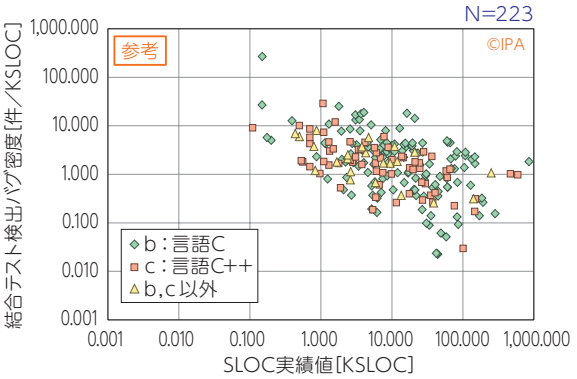
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：結合テスト検出バグ密度（SLOCあたりの検出バグ数）（導出指標）

図表8.2-5 開発言語別のSLOC規模と結合テスト検出バグ密度（改良（派生）開発、開発5工程、0.1KSLOC以上）



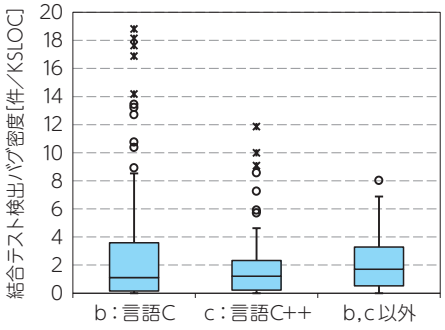
※表示されていないものが10点ある

図表8.2-6 開発言語別のSLOC規模と結合テスト検出バグ密度（改良（派生）開発、開発5工程、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件／KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

図表8.2-7 開発言語別結合テスト検出バグ密度（改良（派生）開発、開発5工程、0.1KSLOC以上）箱ひげ図



図表8.2-8 開発言語別結合テスト検出バグ密度の基本統計量（改良（派生）開発、開発5工程、0.1KSLOC以上）

[件／KSLOC]

開発言語	N	P25	中央	P75
b：言語C	163	0.159	1.110	3.586
c：言語C++	84	0.214	1.206	2.319
b,c以外	27	0.523	1.711	3.277

●言語Cと言語C++を比べると、分布のバラつきは言語Cの方が大きいが、中央値の差異は小さい。

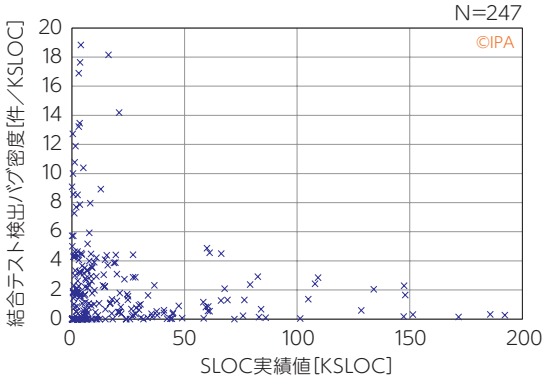
8.2.1.3 開発言語C/C++のSLOC規模と結合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、改良（派生）開発で言語Cと言語C++のプロジェクトを対象に、SLOC規模と結合テスト検出バグ密度の関係をSLOC規模別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良(派生)開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 実効SLOC実績値 ≧ 0.1KSLOC
- 10-3-1_結合テスト検出バグ現象数 ≧ 0

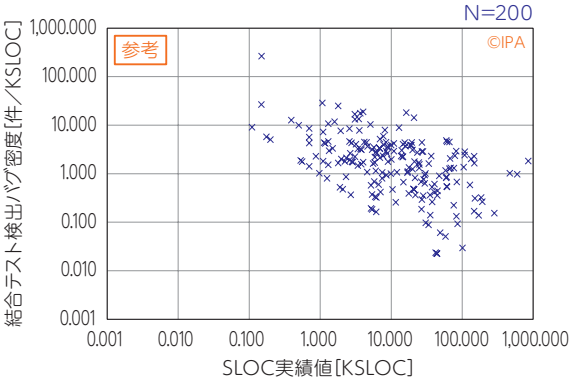
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：結合テスト検出バグ密度（SLOCあたりの検出バグ数）（導出指標）

図表8.2-9 SLOC規模と結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



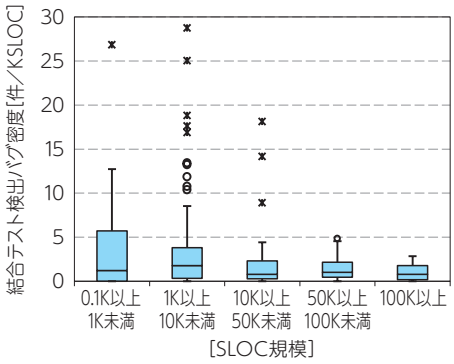
※表示されていないものが9点ある

図表8.2-10 SLOC規模と結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件／KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

図表8.2-11 SLOC規模別結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表8.2-12 SLOC規模別結合テスト検出バグ密度の基本統計量（改良（派生）開発、開発5工程、0.1KSLOC以上）

[件／KSLOC]

SLOC規模	N	P25	中央	P75
全体	247	0.177	1.165	3.141
0.1K以上1K未満	28	0.000	1.227	5.714
1K以上10K未満	107	0.328	1.779	3.817
10K以上50K未満	74	0.271	0.798	2.319
50K以上100K未満	20	0.460	1.011	2.150
100K以上	18	0.194	0.783	1.786

●結合テスト検出バグ密度は、SLOC規模が大きくなるとSLOCK規模による違いが見られない。

8.2.2　総合テスト検出バグ密度

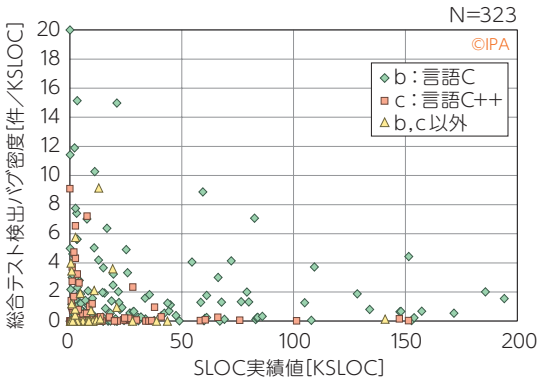
8.2.2.1　開発言語別のSLOC規模と総合テスト検出バグ密度：改良（派生）開発、0.1KSLOC以上

ここでは、改良（派生）開発プロジェクトを対象に、SLOC規模と総合テスト検出バグ密度の関係について、開発言語（C、C++、C/C++以外）別に示す。

■層別定義
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1が回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- 10-4-1_総合テスト検出バグ現象数 ≧ 0

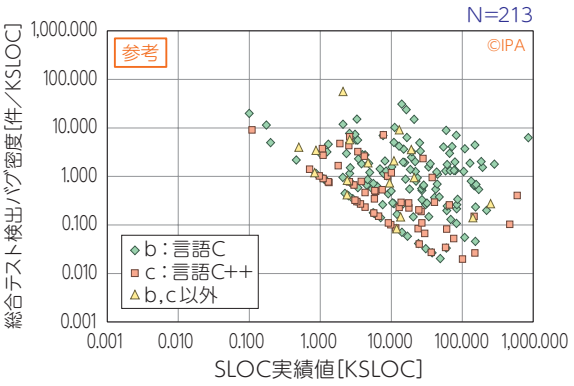
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：総合テスト検出バグ密度（SLOCあたりの検出バグ数）（導出指標）

図表8.2-13　開発言語別のSLOC規模と総合テスト検出バグ密度（改良（派生）開発、0.1KSLOC以上）



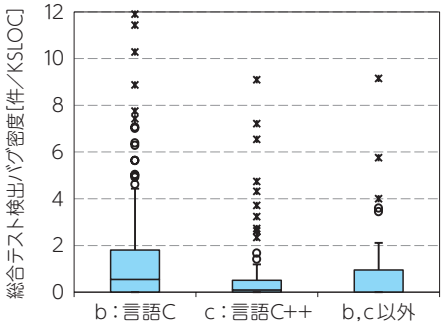
※表示されていないものが10点ある

図表8.2-14　開発言語別のSLOC規模と総合テスト検出バグ密度（改良（派生）開発、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件／KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

図表8.2-15　開発言語別総合テスト検出バグ密度（改良（派生）開発、0.1KSLOC以上）箱ひげ図



図表8.2-16　開発言語別総合テスト検出バグ密度の基本統計量（改良（派生）開発、0.1KSLOC以上）

[件／KSLOC]

開発言語	N	P25	中央	P75
b：言語C	192	0.000	0.548	1.802
c：言語C++	94	0.000	0.094	0.505
b,c以外	37	0.000	0.000	0.952

●ここでは、開発5工程に関係なく総合テストを実施しているプロジェクトを対象にしている。開発5工程すべてを実施したプロジェクトの場合は次項に示す。

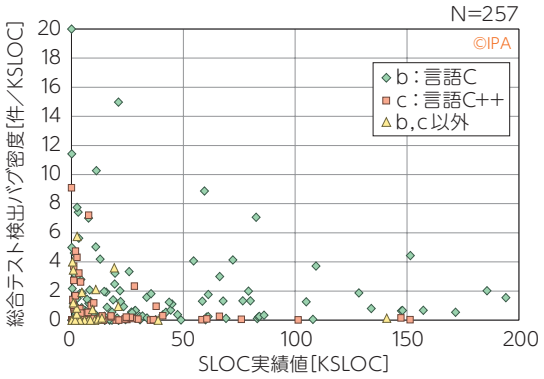
8.2.2.2 開発言語別のSLOC規模と総合テスト検出バグ密度：改良（派生）開発、開発5工程、0.1KSLOC以上

ここでは、改良（派生）開発で開発5工程のプロジェクトを対象に、SLOC規模と総合テスト検出バグ密度の関係について、開発言語（C、C++、C/C++以外）別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1が回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- 10-4-1_総合テスト検出バグ現象数 ≧ 0

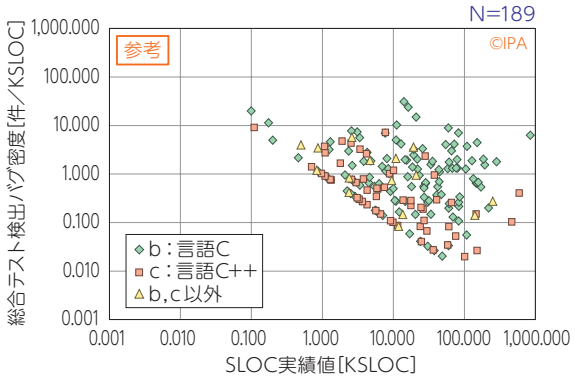
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：総合テスト検出バグ密度（SLOCあたりの検出バグ数）（導出指標）

図表8.2-17 開発言語別のSLOC規模と総合テスト検出バグ密度（改良（派生）開発、開発5工程、0.1KSLOC以上）



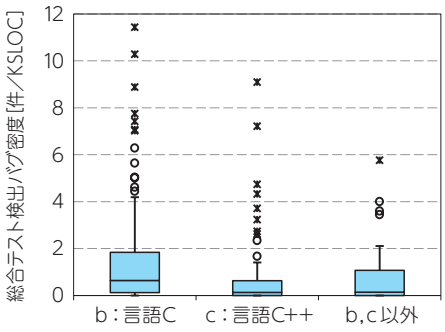
※表示されていないものが8点ある

図表8.2-18 開発言語別のSLOC規模と総合テスト検出バグ密度（改良（派生）開発、開発5工程、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件／KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

図表8.2-19 開発言語別総合テスト検出バグ密度（改良（派生）開発、開発5工程、0.1KSLOC以上）箱ひげ図



図表8.2-20 開発言語別総合テスト検出バグ密度の基本統計量（改良（派生）開発、開発5工程、0.1KSLOC以上）

[件／KSLOC]

開発言語	N	P25	中央	P75
b：言語C	148	0.125	0.640	1.837
c：言語C++	82	0.000	0.130	0.633
b,c以外	27	0.000	0.142	1.074

- 言語Cと言語C++を比べると、言語Cの方がバラつきが大きくなっている。
- 総合テスト検出バグ密度は言語Cよりも言語C++の方が低い傾向が見られるが、製品特性によるバグ密度の差異が現れたものと考えられる。

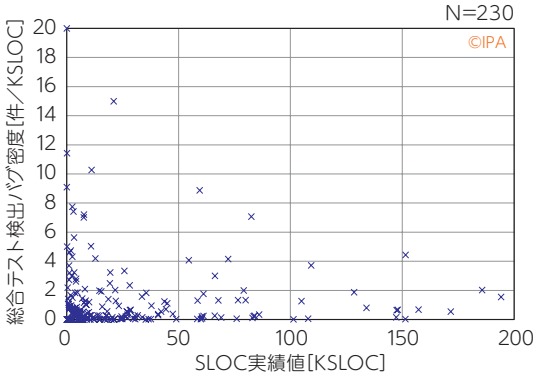
8.2.2.3 開発言語C/C++のSLOC規模と総合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、改良（派生）開発で言語Cと言語C++のプロジェクトを対象に、SLOC規模と総合テスト検出バグ密度の関係をSLOC規模別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 実効SLOC実績値 ≧ 0.1KSLOC
- 10-4-1_総合テスト検出バグ現象数 ≧ 0

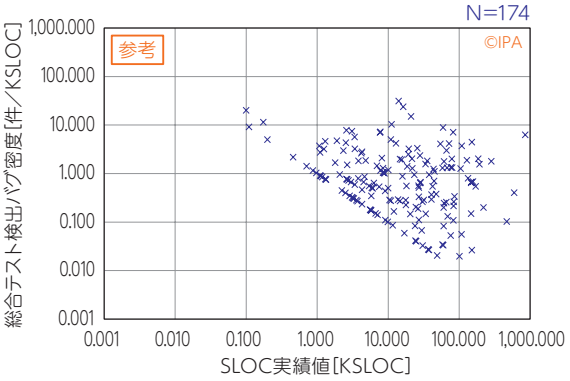
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：総合テスト検出バグ密度（SLOCあたりの検出バグ数）（導出指標）

図表8.2-21 SLOC規模と総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



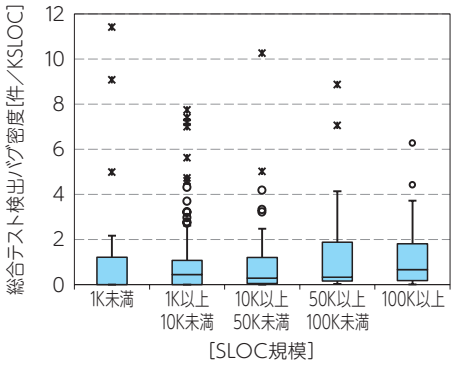
※表示されていないものが5点ある

図表8.2-22 SLOC規模と総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件／KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

図表8.2-23 SLOC規模別総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表8.2-24 SLOC規模別総合テスト検出バグ密度の基本統計量（改良（派生）開発、開発5工程、0.1KSLOC以上）

[件／KSLOC]

SLOC規模	N	P25	中央	P75
全体	230	0.022	0.357	1.330
1K未満	24	0.000	0.000	1.214
1K以上10K未満	92	0.000	0.457	1.083
10K以上50K未満	71	0.050	0.290	1.212
50K以上100K未満	23	0.164	0.337	1.881
100K以上	20	0.187	0.674	1.814

●中央値で比較すると、総合テスト検出バグ密度にSLOC規模別の違いは見られない。

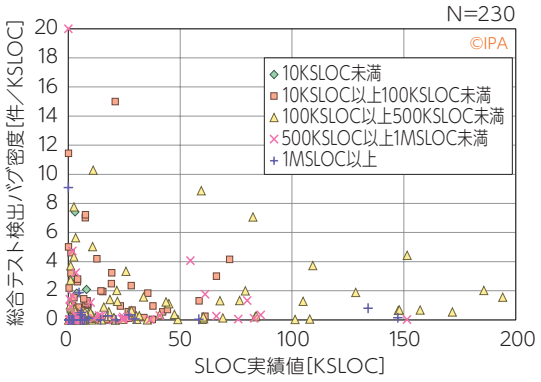
8.2.2.4 母体SLOC規模別のSLOC規模と総合テスト検出バグ密度：改良（派生）開発、開発5工程、0.1KSLOC以上

ここでは、前項8.2.2.3で分析したデータを対象に、SLOC規模と総合テスト検出バグ密度の関係を母体SLOC規模別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 実効SLOC実績値 ≧ 0.1KSLOC
- 7-6_SLOC実績値（母体） ＞ 0
- 10-4-1_総合テスト検出バグ現象数 ≧ 0

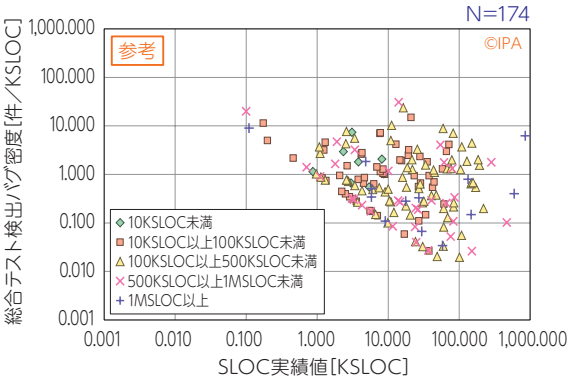
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：総合テスト検出バグ密度（SLOCあたりの検出バグ数）（導出指標）

図表8.2-25 母体SLOC規模別のSLOC規模と総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



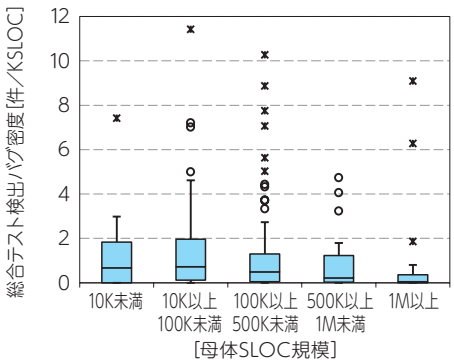
※表示されていないものが7点ある

図表8.2-26 母体SLOC規模別のSLOC規模と総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件／KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

図表8.2-27 母体SLOC規模別総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表8.2-28 母体SLOC規模別総合テスト検出バグ密度の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

[件／KSLOC]

母体SLOC規模	N	P25	中央	P75
10K未満	13	0.000	0.667	1.828
10K以上100K未満	66	0.118	0.723	1.964
100K以上500K未満	87	0.048	0.493	1.293
500K以上1M未満	40	0.027	0.218	1.222
1M以上	24	0.000	0.051	0.359

●母体SLOC規模別の総合テストバグ密度は、同じ実効SLOC値の範囲で見た場合、違いは見られない。

8.3 テストケース数と検出バグ数、テスト工数

本節では、結合テスト、総合テストの2工程について、規模あたりのテストケース数、検出バグ現象数、及び規模あたりのテスト工数を示す。対象プロジェクトは、開発言語がCもしくはC++の改良(派生)開発プロジェクトとする。

8.3.1 SLOC規模あたりのテストケース数、検出バグ数：改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上

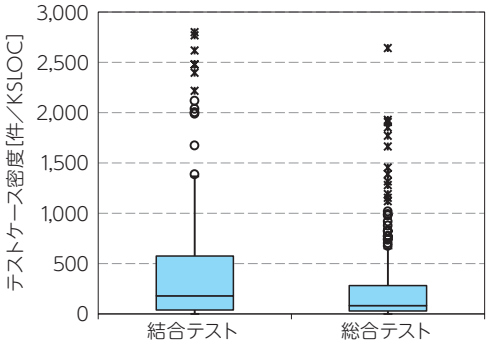
8.3.1.1 SLOC規模あたりの結合・総合テストケース数、検出バグ数

ここでは、改良(派生)開発で言語Cと言語C++のプロジェクトを対象に、SLOC規模あたりのテストケース数と検出バグ数を示す。

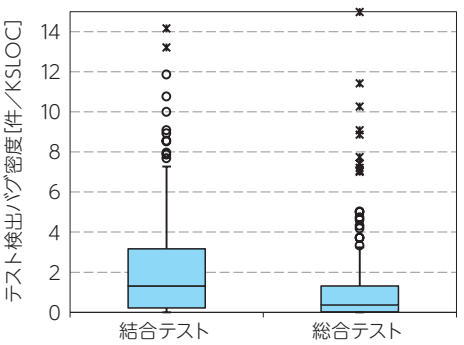
■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良(派生)開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 結合テスト、総合テストの両方を実施
- 実効SLOC実績値 ≧ 0.1KSLOC

■対象データ
- テストケース数
  (10-1-1_結合テストケース数、
  10-2-1_総合テストケース数)
- 検出バグ現象数
  (10-3-1_結合テスト検出バグ現象数、
  10-4-1_総合テスト検出バグ現象数)

図表8.3-1 SLOC規模あたりのテストケース数(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図



図表8.3-2 SLOC規模あたりの検出バグ数(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図

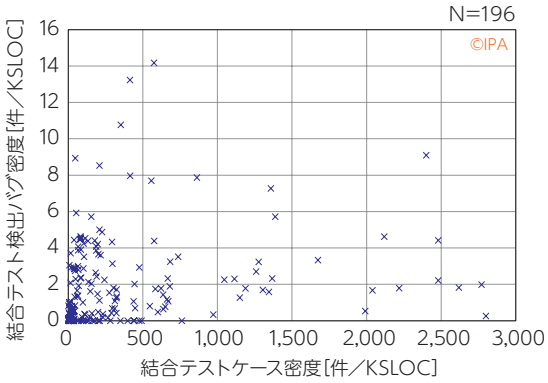


図表8.3-3 SLOC規模あたりのテストケース数、検出バグ数の基本統計量(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)

[件／KSLOC]

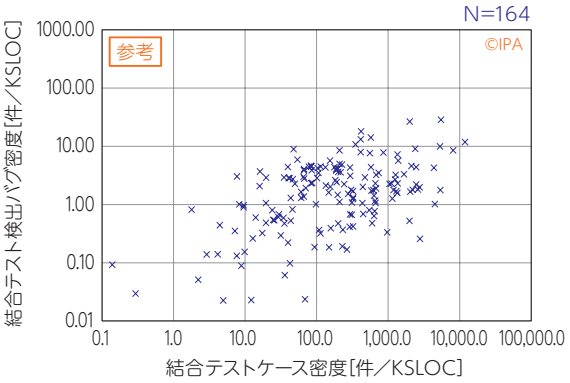
	N	P25	中央	P75
結合テストケース密度	196	40.673	181.104	575.663
総合テストケース密度	196	30.591	82.848	282.940
結合テスト検出バグ密度	196	0.216	1.311	3.159
総合テスト検出バグ密度	196	0.020	0.357	1.317

図表8.3-4　結合テストケース密度と検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

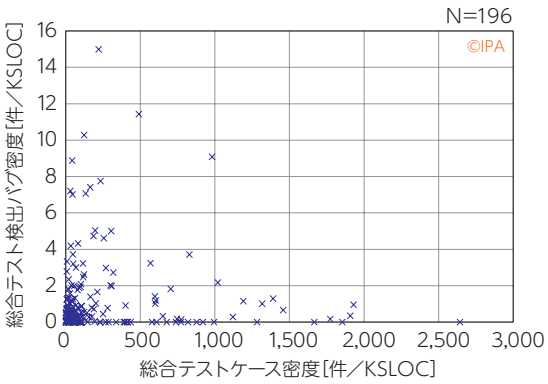


※表示されていないものが12点ある

図表8.3-5　結合テストケース密度と検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示

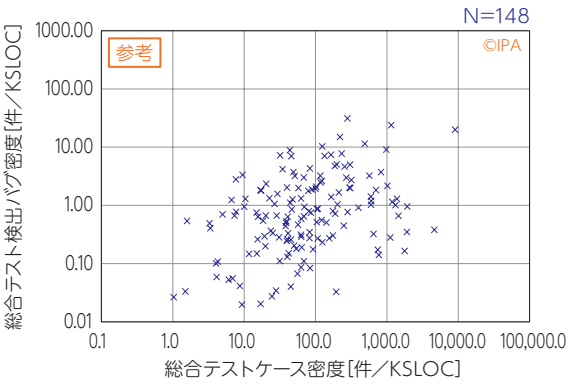


図表8.3-6　総合テストケース密度と検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



※表示されていないものが5点ある

図表8.3-7　総合テストケース密度と検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件／KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

●テストケース密度とテスト検出バグ密度の関係を示す。

8.3.1.2　母体含むSLOC規模あたりの結合・総合テストケース数、検出バグ数

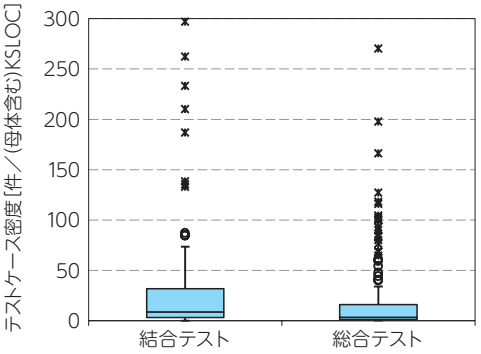
ここでは、改良(派生)開発で言語Cと言語C++のプロジェクトを対象に、母体を含むSLOC規模あたりのテストケース数と検出バグ数を示す。

改良(派生)開発での結合テストは、新規部分や改造部分を中心にテストが行われ、総合テストは、母体を含んだ全体のテストが行われると考えられるので、SLOC規模あたりのテストケース数や検出バグ数は母体を含んだSLOC規模で算出した。

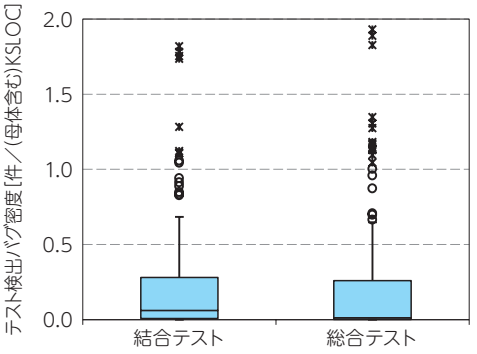
■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良(派生)開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 結合テスト、総合テストの両方を実施
- 実効SLOC実績値 ≧ 0.1KSLOC
- 7-6_SLOC実績値(母体) ＞ 0

■対象データ
- テストケース数
  (10-1-1_結合テストケース数、
  10-2-1_総合テストケース数)
- 検出バグ現象数
  (10-3-1_結合テスト検出バグ現象数、
  10-4-1_総合テスト検出バグ現象数)

図表8.3-8　母体含むSLOC規模あたりのテストケース数(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図



図表8.3-9　母体含むSLOC規模あたりの検出バグ数(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図

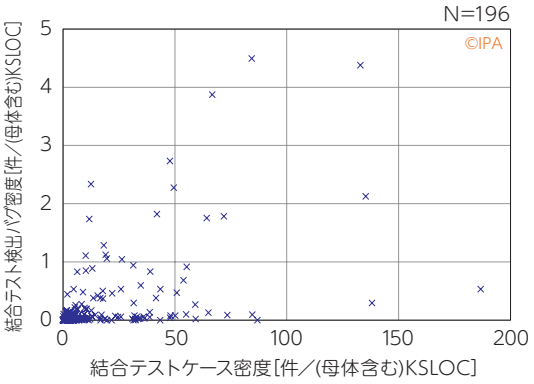


図表8.3-10　母体含むSLOC規模あたりのテストケース数、検出バグ数の基本統計量(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)

[件／(母体含む)KSLOC]

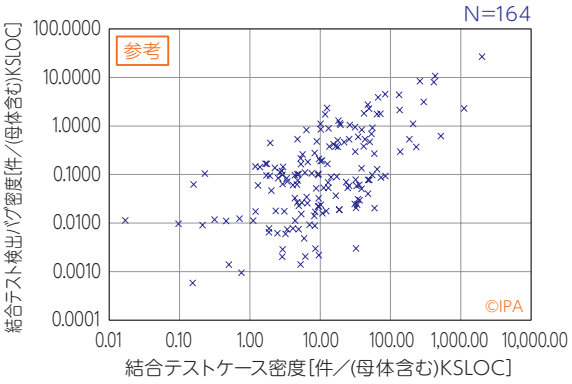
	N	P25	中央	P75
結合テストケース密度	196	2.932	8.686	31.569
総合テストケース密度	196	1.000	3.401	16.028
結合テスト検出バグ密度	196	0.008	0.061	0.280
総合テスト検出バグ密度	196	0.001	0.013	0.260

図表8.3-11　母体含む結合テストケース密度と検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

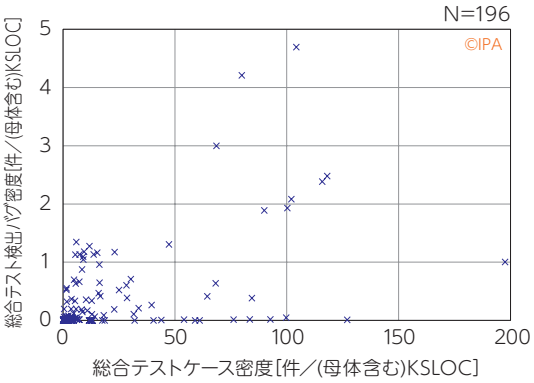


※表示されていないものが10点ある

図表8.3-12　母体含む結合テストケース密度と検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示

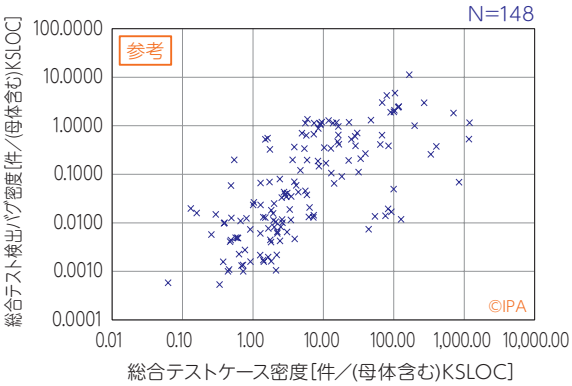


図表8.3-13　母体含む総合テストケース密度と検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



※表示されていないものが11点ある

図表8.3-14　母体含む総合テストケース密度と検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件／KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

8.3.2 SLOC規模あたりのテスト工数：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

8.3.2.1 SLOC規模あたりの結合テスト・総合テスト工数

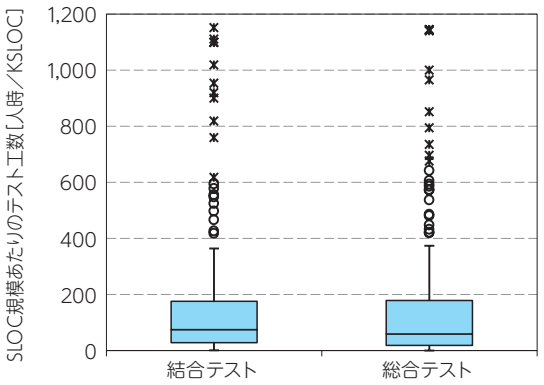
ここでは、改良（派生）開発で言語Cと言語C++のプロジェクトを対象に、SLOC規模あたりのテスト工数の実績値を示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 結合テスト、総合テストの両方を実施
- 実効SLOC実績値 ≧ 0.1KSLOC
- 実績工数（総計人時）結合テスト ＞ 0
- 実績工数（総計人時）総合テスト ＞ 0

■対象データ
- 結合テスト工数
  （9-4-33_社内実績工数_結合テスト、
  9-8-5_外部委託工数_結合テスト）
- 総合テスト工数
  （9-4-34_社内実績工数_総合テスト、
  9-8-6_外部委託工数_総合テスト）

※結合テスト、総合テスト実績工数は、社内、外部委託の実績工数合計の人時換算値を使用。

図表8.3-15　SLOC規模あたりのテスト実績工数（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表8.3-16　SLOC規模あたりのテスト実績工数の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

［人時／KSLOC］

	N	P25	中央	P75
結合テスト	229	27.87	74.30	175.00
総合テスト	229	17.90	59.26	177.99

8.3.2.2 母体含むSLOC規模あたりの結合テスト・総合テスト工数

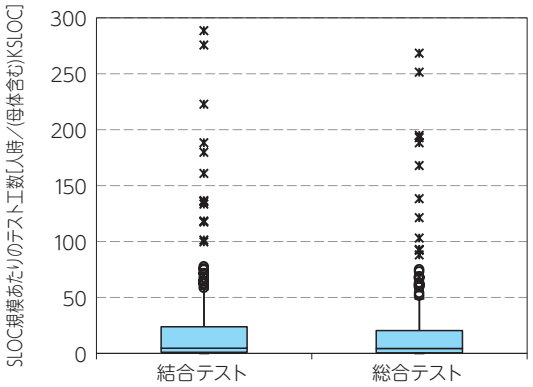
ここでは、改良(派生)開発で言語Cと言語C++のプロジェクトを対象に、母体を含むSLOC規模あたりのテスト工数の実績値を示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良(派生)開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 結合テスト、総合テストの両方を実施
- 実効SLOC実績値 ≧ 0.1KSLOC
- 7-6_SLOC実績値(母体) ＞ 0
- 実績工数(総計人時)結合テスト ＞ 0
- 実績工数(総計人時)総合テスト ＞ 0

■対象データ
- 結合テスト工数
  (9-4-33_社内実績工数_結合テスト、
  9-8-5_外部委託工数_結合テスト)
- 総合テスト工数
  (9-4-34_社内実績工数_総合テスト、
  9-8-6_外部委託工数_総合テスト)

※結合テスト、総合テスト実績工数は、社内、外部委託の実績工数合計の人時換算値を使用。

図表8.3-17 母体含むSLOC規模あたりのテスト実績工数(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図



図表8.3-18 母体含むSLOC規模あたりのテスト実績工数の基本統計量(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)

[人時／(母体含む)KSLOC]

	N	P25	中央	P75
結合テスト	229	1.08	4.74	23.94
総合テスト	229	0.58	4.32	20.37

8.4 SLOC規模とテスト工期

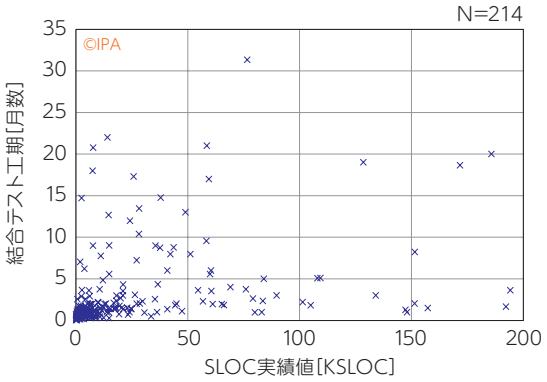
8.4.1 SLOC規模と結合テスト工期：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、改良（派生）開発で言語Cと言語C++のプロジェクトを対象に、SLOC規模と結合テスト実績月数の関係を示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 実効SLOC実績値 ≧ 0.1KSLOC
- 8-2-3-5_結合テスト_月数 ＞ 0

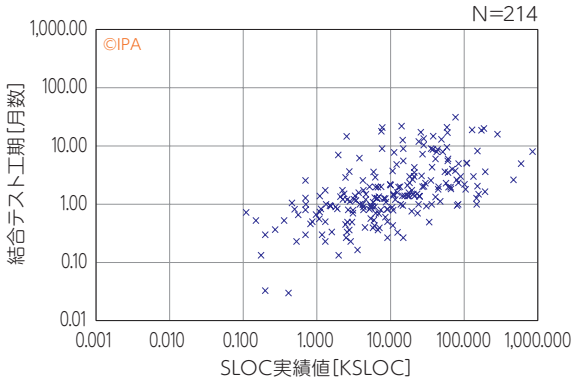
■対象データ
- 8-2-3-5_結合テスト_月数

図表8.4-1 SLOC規模と結合テスト実績月数（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



※表示されていないものが4点ある

図表8.4-2 SLOC規模と結合テスト実績月数（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



※SLOC規模と結合テスト工期について、対数化した場合の相関係数は次のようになる。
R=0.59（P＜0.05）

図表8.4-3 SLOC規模と結合テスト実績月数の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

［月数］

N	P25	中央	P75
214	0.81	1.39	3.00

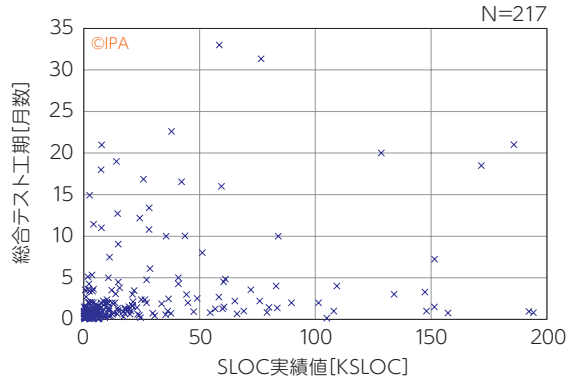
8.4.2 SLOC規模と総合テスト工期：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、改良（派生）開発で言語Cと言語C++のプロジェクトを対象に、SLOC規模と総合テスト実績月数の関係を示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 実効SLOC実績値 ≧ 0.1KSLOC
- 8-2-3-6_総合テスト_月数 ＞ 0

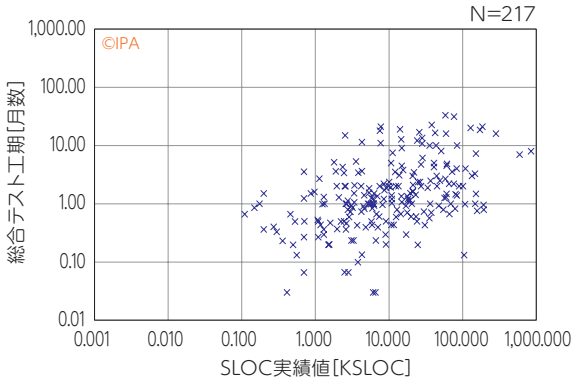
■対象データ
- 8-2-3-6_総合テスト_月数

図表8.4-4 SLOC規模と総合テスト実績月数（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



※表示されていないものが3点ある

図表8.4-5 SLOC規模と総合テスト実績月数（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



※SLOC規模と総合テスト工期について、対数化した場合の相関係数は次のようになる。
R=0.46（P＜0.05）

図表8.4-6 SLOC規模と総合テスト実績月数の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

［月数］

N	P25	中央	P75
217	0.60	1.25	3.00

8.5 工数あたりのテストケース数、検出バグ数

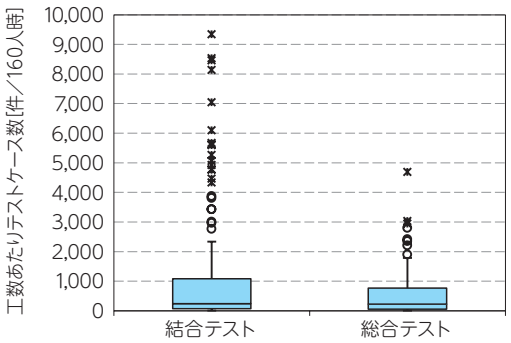
8.5.1 工数あたりのテストケース数、検出バグ数：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、改良（派生）開発で言語Cと言語C++のプロジェクトを対象に、結合テスト及び総合テストについて、工数（1人月＝160人時）あたりのテストケース数と検出バグ数を示す。

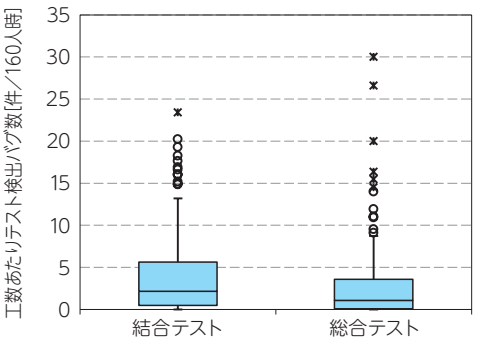
■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 実効SLOC実績値 ≧ 0.1KSLOC
- 実績工数（結合テスト）＞ 0
- 実績工数（総合テスト）＞ 0

■対象データ
- テストケース数
  （10-1-1_結合テストケース数、
  10-2-1_総合テストケース数）
- 検出バグ現象数
  （10-3-1_結合テスト検出バグ現象数、
  10-4-1_総合テスト検出バグ現象数）

図表8.5-1 工数あたりのテストケース数（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表8.5-2 工数あたりの検出バグ数（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図

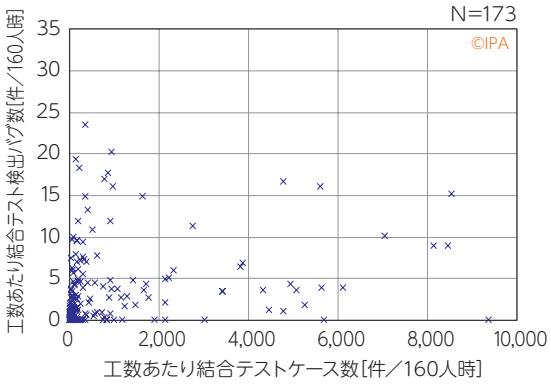


図表8.5-3 工数あたりのテストケース数の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

［件／160人時］

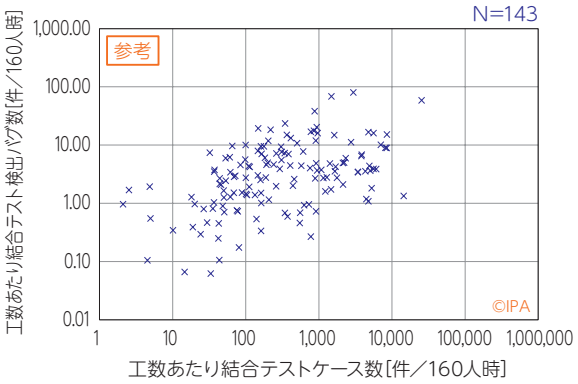
	N	P25	中央	P75
工数あたり結合テストケース数	173	72.24	240.78	1081.71
工数あたり総合テストケース数	173	50.45	224.68	762.36
工数あたり結合テスト検出バグ数	173	0.47	2.16	5.63
工数あたり総合テスト検出バグ数	173	0.09	1.09	3.58

図表8.5-4　結合テスト工数あたりのテストケース数と検出バグ数（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

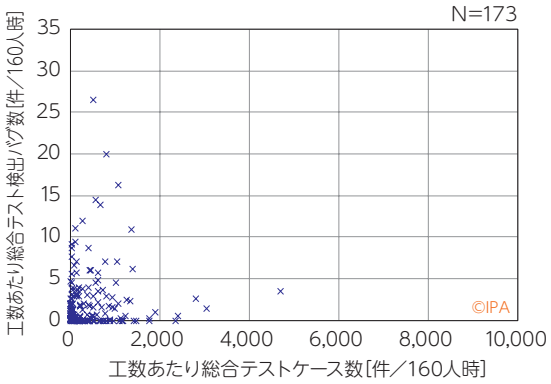


※表示されていないものが6点ある

図表8.5-5　結合テスト工数あたりのテストケース数と検出バグ数（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示

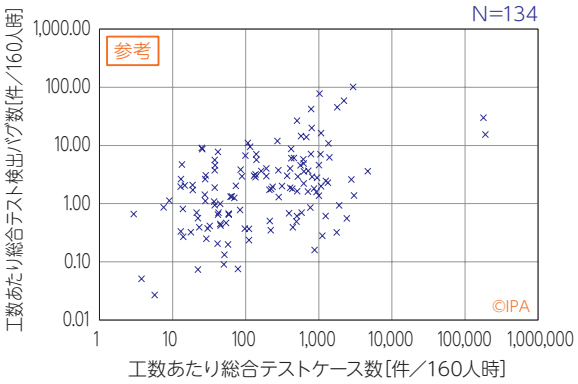


図表8.5-6　総合テスト工数あたりのテストケース数と検出バグ数（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



※表示されていないものが8点ある

図表8.5-7　総合テスト工数あたりのテストケース数と検出バグ数（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では工数あたり検出バグ数が“0件／KSLOC”より大きい標本を対象にしている。工数あたり検出バグ数の増減傾向が視覚的に分かりやすい。

9章　信頼性と生産性の関係

9.1　位置付け …… 114

9.2　テスト検出バグ密度と生産性 …… 115

9.2.1　結合テスト検出バグ密度とSLOC生産性：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

9.2.2　総合テスト検出バグ密度とSLOC生産性：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

9.2.3　テスト検出バグ密度別SLOC規模とSLOC生産性：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

9章　信頼性と生産性の関係

9.1　位置付け

この章では、7章「生産性の分析」の結果と8章「信頼性の分析」の結果との関係性を調べた。

本章で扱った主な情報は、テスト検出バグ密度、SLOC規模、SLOC生産性（SLOC規模／工数）である。本章で取り上げる分析対象の組み合わせと層別のパターンを図表9.1-1に示す。

図表9.1-2には、要素データ“SLOC規模”の分布状況の参照先を示す。

図表9.1-1　分析対象と層別のパターン

<table>
<tr><th colspan="2" rowspan="2">分析対象</th><th colspan="4">層別のパターン</th><th rowspan="2">項番号</th></tr>
<tr><th>開発種別</th><th>開発工程</th><th>開発言語</th><th>その他層別</th></tr>
<tr><td>テスト検出バグ密度</td><td>SLOC生産性</td><td rowspan="2">改良（派生）開発</td><td rowspan="2">開発5工程</td><td rowspan="2">C/C++</td><td>–</td><td>9.2.1
9.2.2</td></tr>
<tr><td>SLOC規模</td><td>SLOC生産性</td><td>テスト検出バグ密度別</td><td>9.2.3</td></tr>
</table>

図表9.1-2　主な要素データと参照先の節番号

要素データ	参照先の節番号
SLOC規模	5.2

9.2 テスト検出バグ密度と生産性

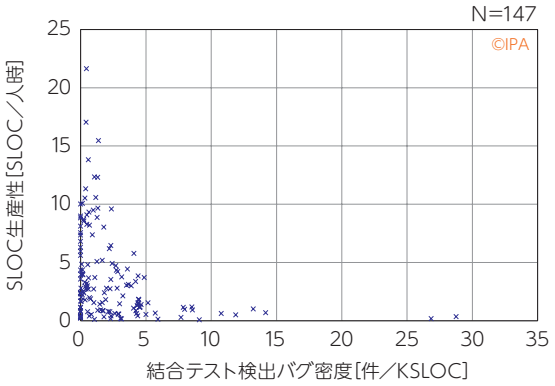
9.2.1 結合テスト検出バグ密度とSLOC生産性：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、改良（派生）開発プロジェクトを対象に、結合テスト検出バグ密度とSLOC生産性の関係について示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 実効SLOC実績値 ≧ 0.1KSLOC
- 10-3-1_結合テスト検出バグ現象数 ≧ 0

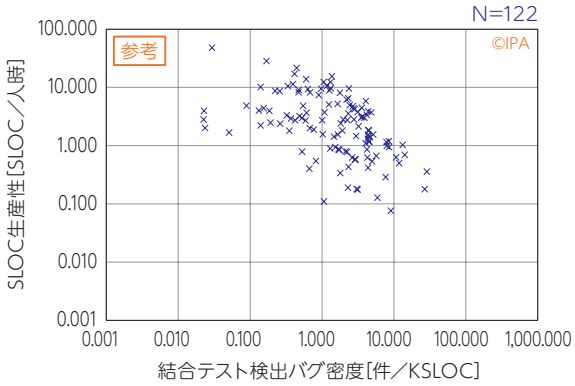
■対象データ
- X軸：結合テスト検出バグ密度（SLOCあたりの検出バグ数）（導出指標）
- Y軸：SLOC生産性（SLOC／実績工数（開発5工程））（導出指標）

図表9.2-1 結合テスト検出バグ密度とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



※表示されていないものが2点ある

図表9.2-2 結合テスト検出バグ密度とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件／KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

●結合テスト検出バグ密度が高くなると、生産性は下がる傾向が見られる。

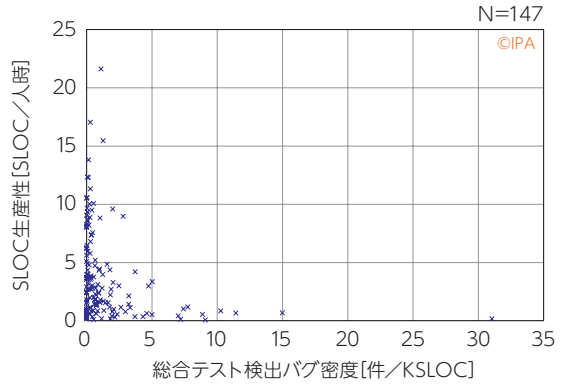
9.2.2 総合テスト検出バグ密度とSLOC生産性：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、改良（派生）開発プロジェクトを対象に、総合テスト検出バグ密度とSLOC生産性の関係について示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 実効SLOC実績値 ≧ 0.1KSLOC
- 10-4-1_総合テスト検出バグ現象数 ≧ 0

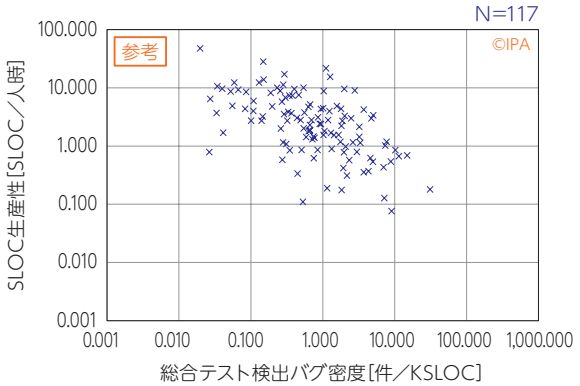
■対象データ
- X軸：総合テスト検出バグ密度
  （SLOCあたりの検出バグ数）（導出指標）
- Y軸：SLOC生産性
  （SLOC／実績工数（開発5工程））（導出指標）

図表9.2-3　総合テスト検出バグ密度とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



※表示されていないものが2点ある

図表9.2-4　総合テスト検出バグ密度とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件／KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

●総合テスト検出バグ密度が高くなると、生産性は下がる傾向が見られる。

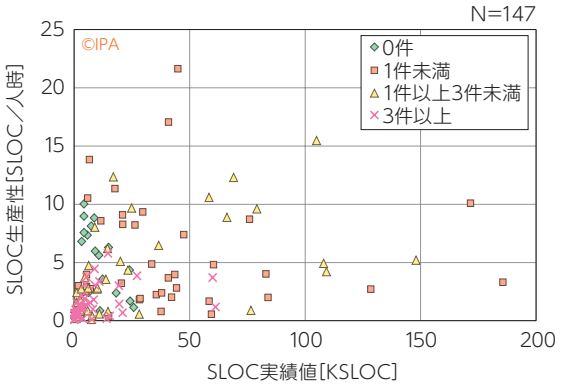
9.2.3 テスト検出バグ密度別SLOC規模とSLOC生産性：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模とSLOC生産性の関係をテスト検出バグ密度別に示す。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 実効SLOC実績値 ≧ 0.1KSLOC
- SLOC生産性 ＞ 0
- 10-3-1_結合テスト検出バグ現象数 ≧ 0
- 10-4-1_総合テスト検出バグ現象数 ≧ 0

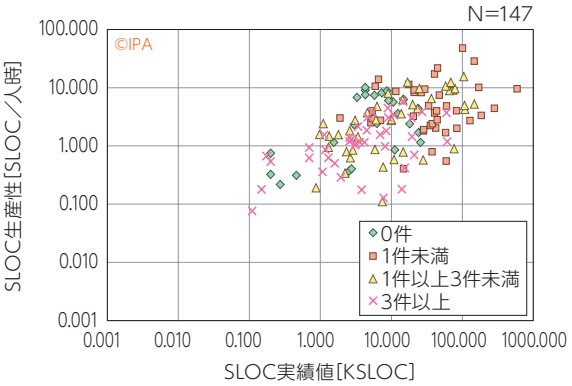
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：SLOC生産性
  （SLOC／実績工数（開発5工程））（導出指標）

図表9.2-5 結合テスト検出バグ密度別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



※表示されていないものが3点ある

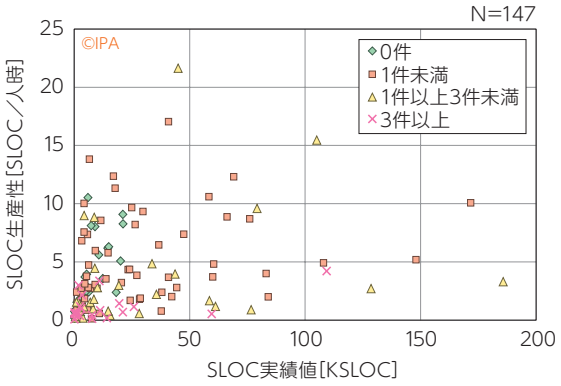
図表9.2-6 結合テスト検出バグ密度別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



※結合テスト検出バグ密度別のSLOC規模とSLOC生産性について、対数化した場合の相関係数は次のようになる。

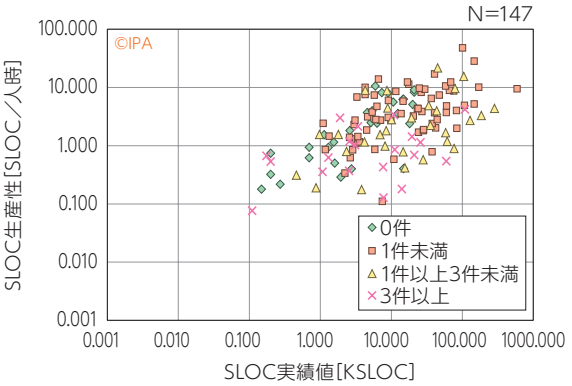
0件	R=0.60（P＜0.05）
1件未満	R=0.14（P≧0.05）
1件以上3件未満	R=0.58（P＜0.05）
3件以上	R=0.51（P＜0.05）

図表9.2-7 総合テスト検出バグ密度別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



※表示されていないものが4点ある

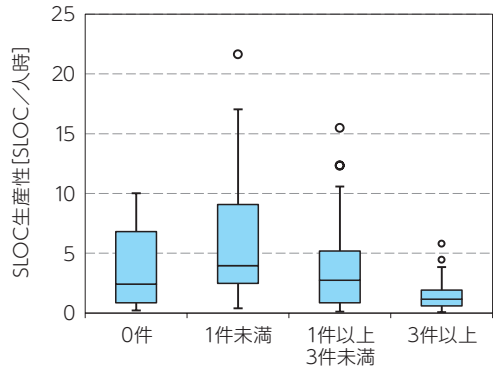
図表9.2-8 総合テスト検出バグ密度別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



※総合テスト検出バグ密度別のSLOC規模とSLOC生産性について、対数化した場合の相関係数は次のようになる。

0件	R=0.78（P＜0.05）
1件未満	R=0.52（P＜0.05）
1件以上3件未満	R=0.51（P＜0.05）
3件以上	R=0.38（P≧0.05）

図表9.2-9　結合テスト検出バグ密度別SLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図

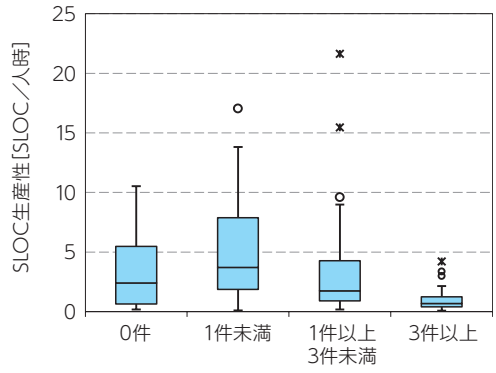


図表9.2-10　結合テスト検出バグ密度別SLOC生産性の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

[SLOC／人時]

結合テスト検出バグ密度	N	P25	中央	P75
0件	25	0.85	2.43	6.81
1件未満	41	2.48	3.97	9.08
1件以上3件未満	41	0.86	2.74	5.19
3件以上	40	0.60	1.16	1.92

図表9.2-11　総合テスト検出バグ密度別SLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表9.2-12　総合テスト検出バグ密度別SLOC生産性の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

[SLOC／人時]

総合テスト検出バグ密度	N	P25	中央	P75
0件	30	0.65	2.41	5.48
1件未満	67	1.87	3.71	7.89
1件以上3件未満	30	0.92	1.74	4.27
3件以上	20	0.42	0.68	1.25

●結合テスト、総合テストともに、バグ密度が高くなるにつれてSLOC生産性が下がる傾向が見られる。

10章 製品の特性別の分析

10.1 位置付け …… 121

10.2 リアルタイム性（時間制約） …… 123
- 10.2.1 リアルタイム性（時間制約）別の工数と工期：改良（派生）開発、開発5工程、開発言語C/C++
- 10.2.2 リアルタイム性（時間制約）別のSLOC規模と工数：改良（派生）開発、開発5工程、開発言語C/C++
- 10.2.3 リアルタイム性（時間制約）別の工程別工数：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
- 10.2.4 リアルタイム性（時間制約）別のSLOC規模とSLOC生産性：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
- 10.2.5 アルタイム性（時間制約）別のSLOC規模と結合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
- 10.2.6 リアルタイム性（時間制約）別のSLOC規模と総合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

10.3 自然環境からの影響度合い …… 130
- 10.3.1 自然環境からの影響度合い別の工数と工期：改良（派生）開発、開発5工程、開発言語C/C++
- 10.3.2 自然環境からの影響度合い別のSLOC規模と工数：改良（派生）開発、開発5工程、開発言語C/C++
- 10.3.3 自然環境からの影響度合い別の工程別工数：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
- 10.3.4 自然環境からの影響度合い別のSLOC規模とSLOC生産性：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
- 10.3.5 自然環境からの影響度合い別のSLOC規模と結合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
- 10.3.6 自然環境からの影響度合い別のSLOC規模と総合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

10.4 ユーザの多様性 …… 136
- 10.4.1 ユーザの多様性別の工数と工期：改良（派生）開発、開発5工程、開発言語C/C++
- 10.4.2 ユーザの多様性別のSLOC規模と工数：改良（派生）開発、開発5工程、開発言語C/C++
- 10.4.3 ユーザの多様性別の工程別工数：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
- 10.4.4 ユーザの多様性別のSLOC規模とSLOC生産性：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
- 10.4.5 ユーザの多様性別のSLOC規模と結合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
- 10.4.6 ユーザの多様性別のSLOC規模と総合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

10.5 法規等による規制度合い …… 142
- 10.5.1 法規等による規制度合い別の工数と工期：改良（派生）開発、開発5工程、開発言語C/C++
- 10.5.2 法規等による規制度合い別のSLOC規模と工数：改良（派生）開発、開発5工程、開発言語C/C++
- 10.5.3 法規等による規制度合い別の工程別工数：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
- 10.5.4 法規等による規制度合い別のSLOC規模とSLOC生産性：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
- 10.5.5 法規等による規制度合い別のSLOC規模と結合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
- 10.5.6 法規等による規制度合い別のSLOC規模と総合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

10.6　M2Mの有無 …… 148
10.6.1　M2Mの有無別の工数と工期：改良（派生）開発、開発5工程、開発言語C/C++
10.6.2　M2Mの有無別のSLOC規模と工数：改良（派生）開発、開発5工程、開発言語C/C++
10.6.3　M2Mの有無別の工程別工数：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
10.6.4　M2Mの有無別のSLOC規模とSLOC生産性：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
10.6.5　M2Mの有無別のSLOC規模と結合テスト検出バグ密度：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
10.6.6　M2Mの有無別のSLOC規模と総合テスト検出バグ密度：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
10.7　ネットワーク接続の有無 …… 154
10.7.1　ネットワーク接続の有無別の工数と工期：改良（派生）開発、開発5工程、開発言語C/C++
10.7.2　ネットワーク接続の有無別のSLOC規模と工数：改良（派生）開発、開発5工程、開発言語C/C++
10.7.3　ネットワーク接続の有無別の工程別工数：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
10.7.4　ネットワーク接続の有無別のSLOC規模とSLOC生産性：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
10.7.5　ネットワーク接続の有無別のSLOC規模と結合テスト検出バグ密度：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
10.7.6　ネットワーク接続の有無別のSLOC規模と総合テスト検出バグ密度：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
10.8　稼動（非停止、オンデマンド） …… 160
10.8.1　稼動（非停止、オンデマンド）別の工数と工期：改良（派生）開発、開発5工程、開発言語C/C++
10.8.2　稼動（非停止、オンデマンド）別のSLOC規模と工数：改良（派生）開発、開発5工程、開発言語C/C++
10.8.3　稼動（非停止、オンデマンド）別の工程別工数：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
10.8.4　稼動（非停止、オンデマンド）別のSLOC規模とSLOC生産性：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
10.8.5　稼動（非停止、オンデマンド）別のSLOC規模と結合テスト検出バグ密度：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
10.8.6　稼動（非停止、オンデマンド）別のSLOC規模と総合テスト検出バグ密度：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
10.9　オンライン保守の可否 …… 166
10.9.1　オンライン保守の可否別の工数と工期：改良（派生）開発、開発5工程、開発言語C/C++
10.9.2　オンライン保守の可否別のSLOC規模と工数：改良（派生）開発、開発5工程、開発言語C/C++
10.9.3　オンライン保守の可否別の工程別工数：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
10.9.4　オンライン保守の可否別のSLOC規模とSLOC生産性：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
10.9.5　オンライン保守の可否別のSLOC規模と結合テスト検出バグ密度：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
10.9.6　オンライン保守の可否別のSLOC規模と総合テスト検出バグ密度：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
10.10　障害リスク（Type） …… 172
10.10.1　障害リスク（Type）別の工数と工期：改良（派生）開発、開発5工程、開発言語C/C++
10.10.2　障害リスク（Type）別のSLOC規模と工数：改良（派生）開発、開発5工程、開発言語C/C++
10.10.3　障害リスク（Type）別の工程別工数：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上
10.10.4　障害リスク（Type）別のSLOC規模とSLOC生産性：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

10章　製品の特性別の分析

10.1　位置付け

組込みシステムは、リアルタイム性の高い処理が要求されるものや自然の影響を受ける環境で利用されるものなど、製品によって特性が異なるため、その特性を考慮したうえで開発作業を行う。

この章では、第4章4.4項(1)に定義した8つの「製品の特性」と4.3項(2)に定義した「障害リスク」に対し、それぞれの特性の違い別に生産性や信頼性等を分析した。図表10.1-1に分析対象と層別のパターンを示す。図表10.1-2には、要素データ“SLOC規模”と“工数”それぞれの分布状況の参照先を示す。

図表10.1-1　分析対象と層別のパターン

分析対象		層別のパターン				項番号
		開発種別	開発工程	開発言語	その他層別	
工数	工期	改良(派生)開発	開発5工程	C/C++	リアルタイム性(時間制約)別	10.2.1
					自然環境からの影響度合い別	10.3.1
					ユーザの多様性別	10.4.1
					法規等による規制度合い別	10.5.1
					M2Mの有無別	10.6.1
					ネットワーク接続の有無別	10.7.1
					稼動(非停止、オンデマンド)別	10.8.1
					オンライン保守の可否別	10.9.1
					障害リスク別	10.10.1
SLOC規模	工数	改良(派生)開発	開発5工程	C/C++	リアルタイム性(時間制約)別	10.2.2
					自然環境からの影響度合い別	10.3.2
					ユーザの多様性別	10.4.2
					法規等による規制度合い別	10.5.2
					M2Mの有無別	10.6.2
					ネットワーク接続の有無別	10.7.2
					稼動(非停止、オンデマンド)別	10.8.2
					オンライン保守の可否別	10.9.2
					障害リスク別	10.10.2
工程別工数		改良(派生)開発	開発5工程	C/C++	リアルタイム性(時間制約)別	10.2.3
					自然環境からの影響度合い別	10.3.3
					ユーザの多様性別	10.4.3
					法規等による規制度合い別	10.5.3
					M2Mの有無別	10.6.3
					ネットワーク接続の有無別	10.7.3
					稼動(非停止、オンデマンド)別	10.8.3
					オンライン保守の可否別	10.9.3
					障害リスク別	10.10.3

分析対象		層別のパターン				項番号
		開発種別	開発工程	開発言語	その他層別	
SLOC規模	生産性	改良（派生）開発	開発5工程	C/C++	リアルタイム性（時間制約）別	10.2.4
					自然環境からの影響度合い別	10.3.4
					ユーザの多様性別	10.4.4
					法規等による規制度合い別	10.5.4
					M2Mの有無別	10.6.4
					ネットワーク接続の有無別	10.7.4
					稼動（非停止、オンデマンド）別	10.8.4
					オンライン保守の可否別	10.9.4
					障害リスク別	10.10.4
SLOC規模	テスト検出バグ密度	改良（派生）開発	開発5工程	C/C++	リアルタイム性（時間制約）別	10.2.5 10.2.6
					自然環境からの影響度合い別	10.3.5 10.3.6
					ユーザの多様性別	10.4.5 10.4.6
					法規等による規制度合い別	10.5.5 10.5.6
					M2Mの有無別	10.6.5 10.6.6
					ネットワーク接続の有無別	10.7.5 10.7.6
					稼動（非停止、オンデマンド）別	10.8.5 10.8.6
					オンライン保守の可否別	10.9.5 10.9.6

図表10.1-2　主な要素データと参照先の節番号

要素データ	参照先の節番号
SLOC規模	5.2
工数	5.3

10.2　リアルタイム性（時間制約）

この節では、改良（派生）開発で開発５工程（アーキテクチャ設計～総合テスト）の作業が行われたプロジェクトを対象に、リアルタイム性（時間制約）別による分析を示す。

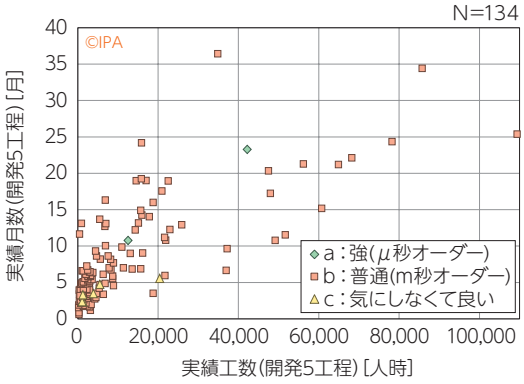
10.2.1　リアルタイム性（時間制約）別の工数と工期：改良（派生）開発、開発5工程、開発言語C/C++

ここでは、実績工数とその工期（月数）の関係をリアルタイム性（時間制約）別に示す。

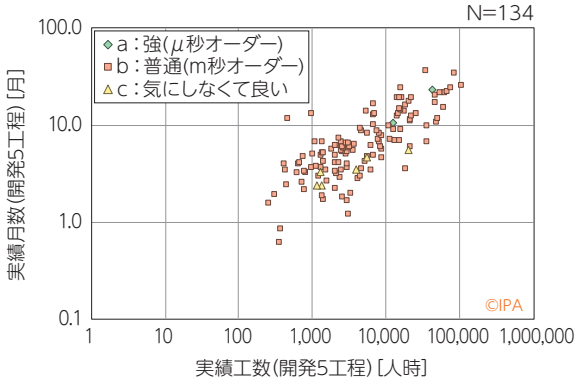
■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-1_リアルタイム性（時間制約）が回答されているもの
- 実績工数（開発5工程）＞０
- 実績月数（開発5工程）＞０

■対象データ
- X軸：実績工数（開発5工程）（導出指標）
- Y軸：実績月数（開発5工程）（導出指標）

図表10.2-1　リアルタイム性（時間制約）別の工数と工期（改良（派生）開発、開発5工程、開発言語C/C++）



図表10.2-2　リアルタイム性（時間制約）別の工数と工期（改良（派生）開発、開発5工程、開発言語C/C++）対数表示



※リアルタイム性（時間制約）【b：普通（m秒オーダー）】の工数と工期について、対数化した場合の相関係数は、R=0.76（P＜0.05）となる。

図表10.2-3　リアルタイム性（時間制約）別の実績月数の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++）

［月］

リアルタイム性（時間制約）	N	P25	中央	P75
a：強（μ秒オーダー）	2	13.89	17.02	20.14
b：普通（m秒オーダー）	126	3.60	5.90	11.08
c：気にしなくて良い	6	2.60	3.38	4.42

●リアルタイム性の違いによる工数と工期の関係は、標本数が少ないため、比較できない。

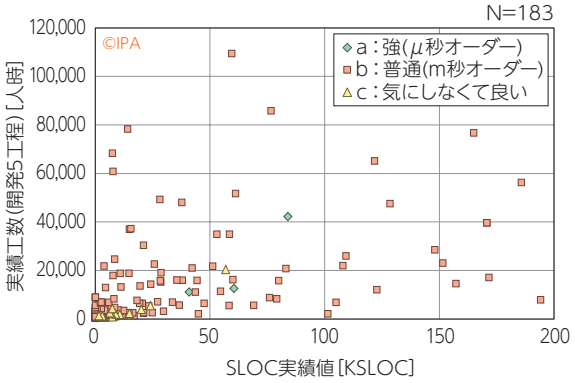
10.2.2　リアルタイム性（時間制約）別のSLOC規模と工数：改良（派生）開発、開発5工程、開発言語C/C++

ここでは、SLOC規模と工数の関係をリアルタイム性（時間制約）別に示す。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-1_リアルタイム性（時間制約）が回答されているもの
- 実効SLOC実績値 ＞ 0
- 実績工数（開発5工程）＞ 0

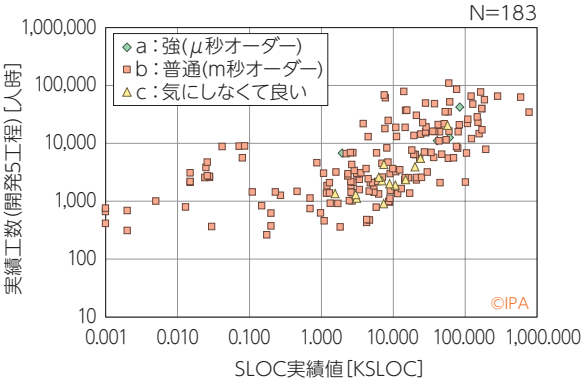
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：実績工数（開発5工程）（導出指標）

図表10.2-4　リアルタイム性（時間制約）別のSLOC規模と工数（改良（派生）開発、開発5工程、開発言語C/C++）



※表示されていないものが3点ある

図表10.2-5　リアルタイム性（時間制約）別のSLOC規模と工数（改良（派生）開発、開発5工程、開発言語C/C++）対数表示



※リアルタイム性（時間制約）【b：普通（m秒オーダー）】、【c：気にしなくて良い】のSLOC規模と工数について、対数化した場合の相関係数は次のようになる。
b：普通（m秒オーダー）　R=0.63（P<0.05）
c：気にしなくて良い　R=0.82（P<0.05）

図表10.2-6　リアルタイム性（時間制約）別の実績工数の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++）

［人時］

リアルタイム性（時間制約）	N	P25	中央	P75
a：強（μ秒オーダー）	4	10,044	11,849	19,975
b：普通（m秒オーダー）	166	1,544	3,547	15,742
c：気にしなくて良い	13	1,349	2,245	3,935

●工数の増加はSLOC規模に依存するという一般的な傾向は、「リアルタイム性：気にしなくて良い」の場合に、比較的強く出ているが、「リアルタイム性：普通（m秒オーダー）」の場合には強いとは言えない。工数が増える要因はSLOC規模の他にもあることが伺える。

10.2.3 リアルタイム性（時間制約）別の工程別工数：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、工程別の実績工数の比率をリアルタイム性（時間制約）別に示す。

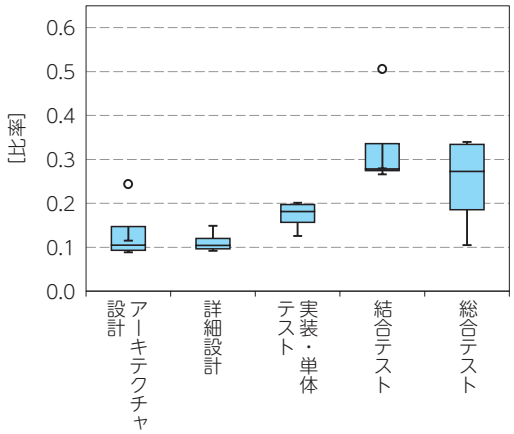
■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-1_リアルタイム性（時間制約）が回答されているもの
- 工程別の実績工数にすべて記入があり、各値が0より大きい
- 実効SLOC実績値 ≧ 0.1KSLOC

■対象データ
- 9-4-30_社内実績工数_アーキテクチャ設計、
  9-4-31_社内実績工数_詳細設計、
  9-4-32_社内実績工数_実装・単体テスト、
  9-4-33_社内実績工数_結合テスト、
  9-4-34_社内実績工数_総合テスト
- 9-8-2_外部委託工数_アーキテクチャ設計、
  9-8-3_外部委託工数_詳細設計、
  9-8-4_外部委託工数_実装・単体テスト、
  9-8-5_外部委託工数_結合テスト、
  9-8-6_外部委託工数_総合テスト

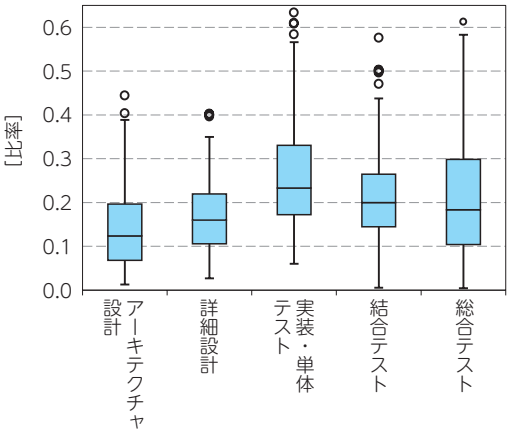
※各工程の実績工数は、社内、外部委託の実績工数合計の人時換算値を使用。

図表10.2-7 リアルタイム性（時間制約）別の工程別の実績工数の比率（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図

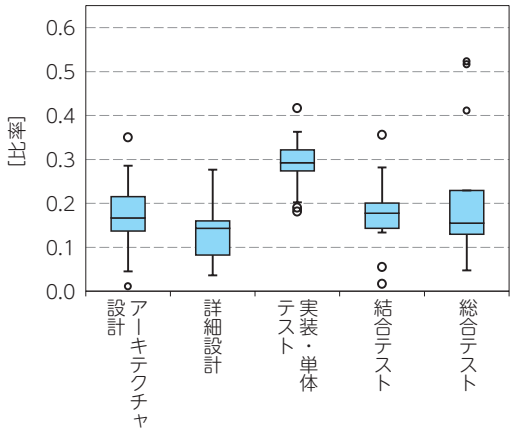
リアルタイム性（時間制約）【強（μ秒オーダー）】



リアルタイム性（時間制約）【普通（m秒オーダー）】



リアルタイム性（時間制約）【気にしなくて良い】



図表10.2-8　リアルタイム性（時間制約）別の工程別の実績工数の比率の基本統計量
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

リアルタイム性（時間制約）【強（μ秒オーダー）】

［比率］

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	4	0.09	0.09	0.11	0.15	0.24	0.14	0.06
詳細設計	4	0.09	0.10	0.10	0.12	0.15	0.11	0.02
実装・単体テスト	4	0.13	0.16	0.18	0.20	0.20	0.17	0.03
結合テスト	4	0.27	0.27	0.28	0.34	0.51	0.33	0.10
総合テスト	4	0.10	0.19	0.27	0.33	0.34	0.25	0.10

リアルタイム性（時間制約）【普通（m秒オーダー）】

［比率］

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	145	0.01	0.07	0.12	0.20	0.44	0.14	0.09
詳細設計	145	0.03	0.11	0.16	0.22	0.67	0.17	0.09
実装・単体テスト	145	0.06	0.17	0.23	0.33	0.79	0.27	0.13
結合テスト	145	0.01	0.14	0.20	0.26	0.58	0.21	0.10
総合テスト	145	0.00	0.10	0.18	0.30	0.61	0.21	0.13

リアルタイム性（時間制約）【気にしなくて良い】

［比率］

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	13	0.01	0.14	0.17	0.22	0.35	0.17	0.09
詳細設計	13	0.04	0.08	0.14	0.16	0.28	0.14	0.07
実装・単体テスト	13	0.18	0.27	0.29	0.32	0.42	0.29	0.07
結合テスト	13	0.02	0.14	0.18	0.20	0.36	0.18	0.09
総合テスト	13	0.05	0.13	0.16	0.23	0.52	0.22	0.15

- 工程別工数比率では、リアルタイム性を気にしなくて良い場合には、実装・単体テストにかける工数比率が高く、結合・総合テストでは低い傾向が見られる。

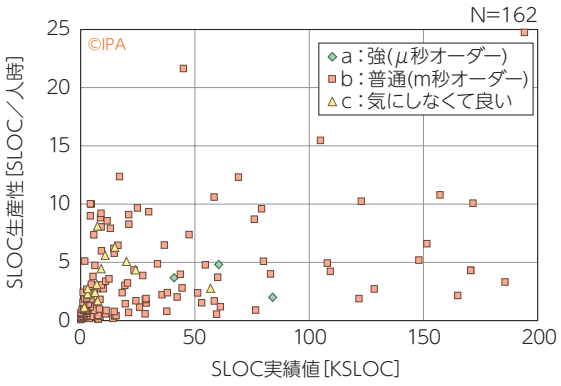
10.2.4　リアルタイム性（時間制約）別のSLOC規模とSLOC生産性：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模とSLOC生産性の関係をリアルタイム性（時間制約）別に示す。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-1_リアルタイム性（時間制約）が回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- SLOC生産性 ＞ 0

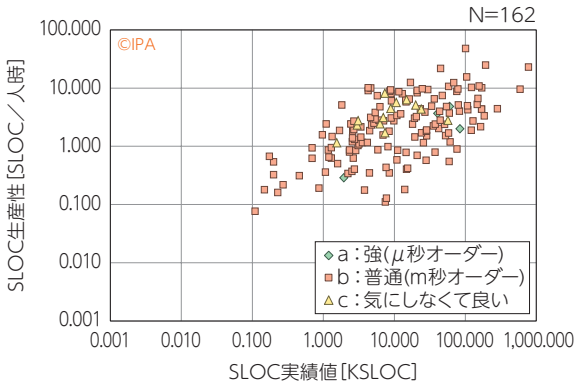
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：SLOC生産性
  （SLOC／実績工数（開発5工程））（導出指標）

図表10.2-9　リアルタイム性（時間制約）別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



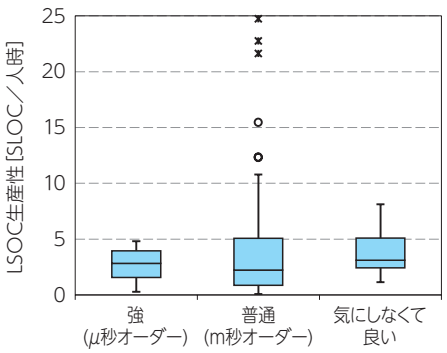
※表示されていないものが4点ある

図表10.2-10　リアルタイム性（時間制約）別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



※リアルタイム性（時間制約）【b：普通（m秒オーダー）】、【c：気にしなくて良い】のSLOC規模とSLOC生産性について、対数化した場合の相関係数は次のようになる。
b：普通（m秒オーダー）　R=0.61（P＜0.05）
c：気にしなくて良い　R=0.51（P≧0.05）

図表10.2-11　リアルタイム性（時間制約）別SLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.2-12　リアルタイム性（時間制約）別SLOC生産性の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

[SLOC／人時]

リアルタイム性（時間制約）	N	P25	中央	P75
a：強（μ秒オーダー）	4	1.56	2.83	3.96
b：普通（m秒オーダー）	145	0.86	2.22	5.08
c：気にしなくて良い	13	2.43	3.12	5.09

●リアルタイム性を気にしなくて良い場合には、強（μ秒オーダー）や普通（m秒オーダー）に比べて生産性が高い傾向が見られる。

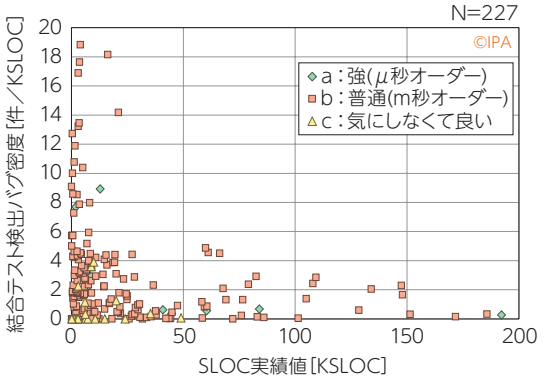
10.2.5 リアルタイム性（時間制約）別のSLOC規模と結合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模と結合テスト検出バグ密度の関係について、リアルタイム性（時間制約）別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-1_リアルタイム性（時間制約）が回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- 10-3-1_結合テスト検出バグ現象数 ≧ 0

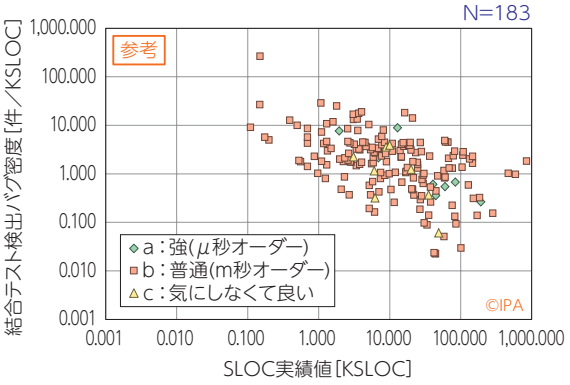
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：結合テスト検出バグ密度（SLOCあたりの検出バグ数）（導出指標）

図表10.2-13 リアルタイム性（時間制約）別のSLOC規模と結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



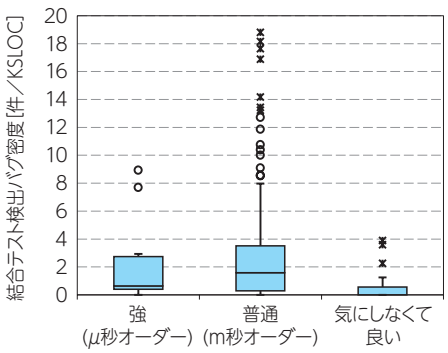
※表示されていないものが9点ある

図表10.2-14 リアルタイム性（時間制約）別のSLOC規模と結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件/KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

図表10.2-15 リアルタイム性（時間制約）別結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.2-16 リアルタイム性（時間制約）別結合テスト検出バグ密度の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

[件／KSLOC]

リアルタイム性（時間制約）	N	P25	中央	P75
a：強（μ秒オーダー）	10	0.404	0.645	2.743
b：普通（m秒オーダー）	197	0.294	1.579	3.516
c：気にしなくて良い	20	0.000	0.000	0.563

●リアルタイム性を気にしなくて良い場合には、強（μ秒オーダー）や普通（m秒オーダー）に比べて結合テスト検出バグ密度が低い傾向にある。

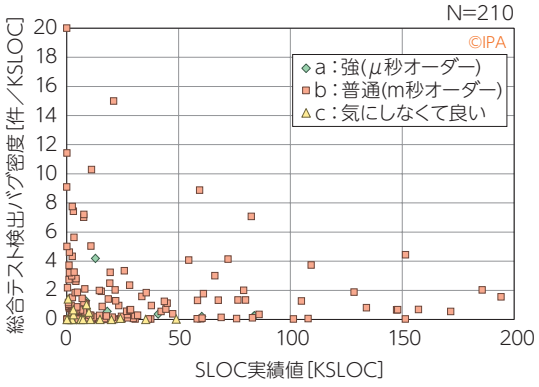
10.2.6　リアルタイム性（時間制約）別のSLOC規模と総合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模と総合テスト検出バグ密度の関係について、リアルタイム性（時間制約）別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-1_リアルタイム性（時間制約）が回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- 10-4-1_総合テスト検出バグ現象数 ≧ 0

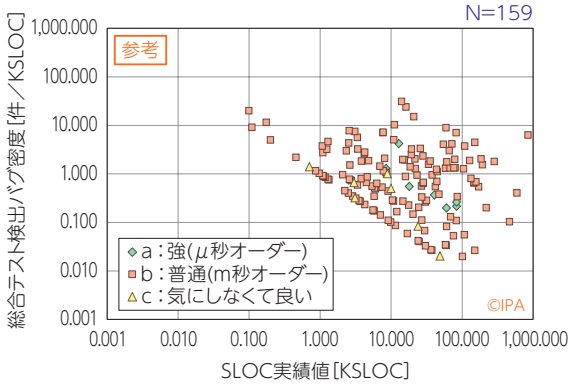
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：総合テスト検出バグ密度
  （SLOCあたりの検出バグ数）（導出指標）

図表10.2-17　リアルタイム性（時間制約）別のSLOC規模と総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



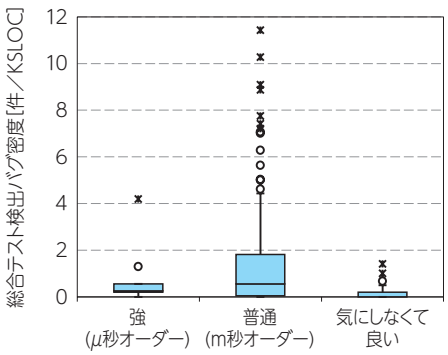
※表示されていないものが7点ある

図表10.2-18　リアルタイム性（時間制約）別のSLOC規模と総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件/KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

図表10.2-19　リアルタイム性（時間制約）別総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.2-20　リアルタイム性（時間制約）別総合テスト検出バグ密度の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

[件／KSLOC]

リアルタイム性（時間制約）	N	P25	中央	P75
a：強（μ秒オーダー）	9	0.199	0.262	0.552
b：普通（m秒オーダー）	182	0.044	0.557	1.818
c：気にしなくて良い	19	0.000	0.000	0.203

●リアルタイム性を気にしなくて良い場合には、強（μ秒オーダー）や普通（m秒オーダー）に比べて総合テスト検出バグ密度が低い傾向にある。

10.3 自然環境からの影響度合い

この節では、改良(派生)開発で開発５工程（アーキテクチャ設計～総合テスト）の作業の行われたプロジェクトを対象に、自然環境からの影響度合い別による分析を示す。

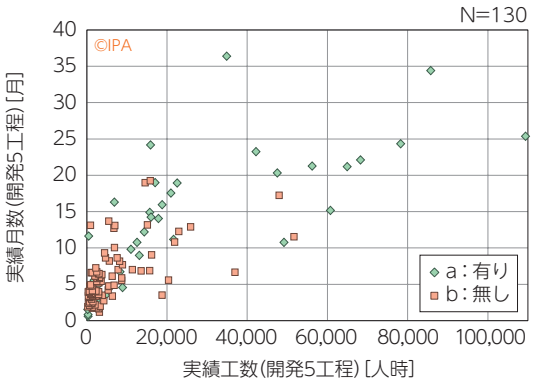
10.3.1 自然環境からの影響度合い別の工数と工期：改良（派生）開発、開発5工程、開発言語C/C++

ここでは、実績工数とその工期（月数）の関係を自然環境からの影響度合い別に示す。

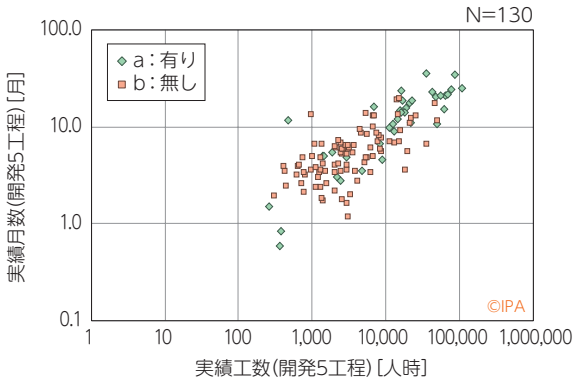
■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-2_自然環境からの影響度合いが回答されているもの
- 実績工数（開発5工程）＞ 0
- 実績月数（開発5工程）＞ 0

■対象データ
- X軸：実績工数（開発5工程）（導出指標）
- Y軸：実績月数（開発5工程）（導出指標）

図表10.3-1 自然環境からの影響度合い別の工数と工期（改良（派生）開発、開発5工程、開発言語C/C++）



図表10.3-2 自然環境からの影響度合い別の工数と工期（改良（派生）開発、開発5工程、開発言語C/C++）対数表示



※自然環境からの影響度合い別の工数と工期について、対数化した場合の相関係数は次のようになる。
a：有り　R=0.86（P<0.05）
b：無し　R=0.62（P<0.05）

図表10.3-3 自然環境からの影響度合い別の実績月数の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++）

［月］

自然環境からの影響度合い	N	P25	中央	P75
a：有り	38	5.19	13.12	20.00
b：無し	92	3.40	5.20	6.90

●自然環境からの影響【有り】の場合、【無し】の場合に比べて、工数の増加に伴い工期が長くなる傾向が顕著である。

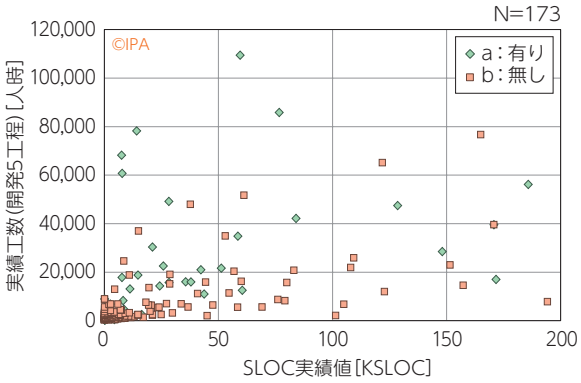
10.3.2 自然環境からの影響度合い別のSLOC規模と工数：改良（派生）開発、開発5工程、開発言語C/C++

ここでは、SLOC規模と工数の関係を自然環境からの影響度合い別に示す。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-2_自然環境からの影響度合いが回答されているもの
- 実効SLOC実績値＞ 0
- 実績工数（開発5工程）＞ 0

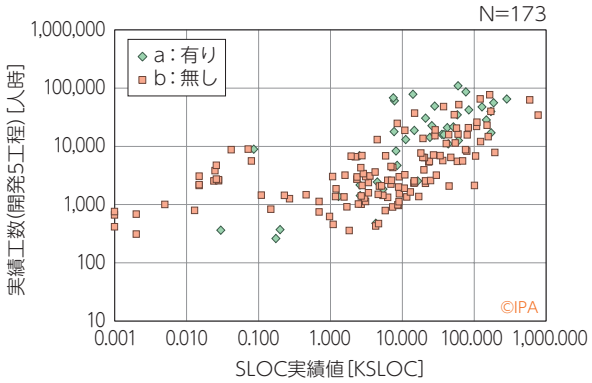
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：実績工数（開発5工程）（導出指標）

図表10.3-4 自然環境からの影響度合い別のSLOC規模と工数（改良（派生）開発、開発5工程、開発言語C/C++）



※表示されていないものが3点ある

図表10.3-5 自然環境からの影響度合い別のSLOC規模と工数（改良（派生）開発、開発5工程、開発言語C/C++）対数表示



※自然環境からの影響度合い別のSLOC規模と工数について、対数化した場合の相関係数は次のようになる。
a：有り　R=0.76（P＜0.05）
b：無し　R=0.58（P＜0.05）

図表10.3-6 自然環境からの影響度合い別の実績工数の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++）

[人時]

自然環境からの影響度合い	N	P25	中央	P75
a：有り	42	3,383	15,961	38,438
b：無し	131	1,436	2,963	7,338

●工数の増加はSLOC規模に依存するという一般的な傾向は、自然環境からの影響【有り】場合では、比較的強く出ているが、自然環境からの影響【無し】の場合では弱い。

10.3.3 自然環境からの影響度合い別の工程別工数：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、工程別の実績工数の比率を自然環境からの影響度合い別に示す。

■層別定義

- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-2_自然環境からの影響度合いが回答されているもの
- 工程別の実績工数にすべて記入があり、各値が0より大きい
- 実効SLOC実績値 ≧ 0.1KSLOC

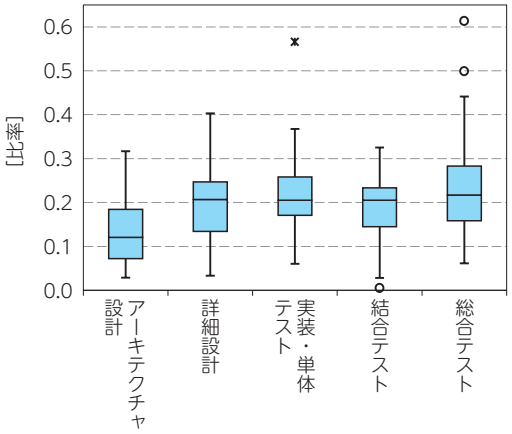
■対象データ

- 9-4-30_社内実績工数_アーキテクチャ設計、
  9-4-31_社内実績工数_詳細設計、
  9-4-32_社内実績工数_実装・単体テスト、
  9-4-33_社内実績工数_結合テスト、
  9-4-34_社内実績工数_総合テスト
- 9-8-2_外部委託工数_アーキテクチャ設計、
  9-8-3_外部委託工数_詳細設計、
  9-8-4_外部委託工数_実装・単体テスト、
  9-8-5_外部委託工数_結合テスト、
  9-8-6_外部委託工数_総合テスト

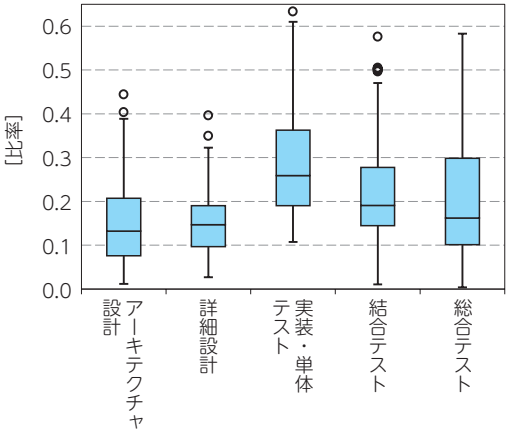
※各工程の実績工数は、社内、外部委託の実績工数合計の人時換算値を使用。

図表10.3-7　自然環境からの影響度合い別の工程別の実績工数の比率（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図

自然環境からの影響度合い【有り】



自然環境からの影響度合い【無し】



図表10.3-8　自然環境からの影響度合い別の工程別の実績工数の比率の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

自然環境からの影響度合い【有り】　[比率]

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	40	0.03	0.07	0.12	0.18	0.32	0.13	0.08
詳細設計	40	0.03	0.13	0.21	0.25	0.67	0.21	0.11
実装・単体テスト	40	0.06	0.17	0.21	0.26	0.57	0.22	0.09
結合テスト	40	0.01	0.14	0.21	0.23	0.33	0.19	0.08
総合テスト	40	0.06	0.16	0.22	0.28	0.61	0.24	0.12

自然環境からの影響度合い【無し】　[比率]

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	112	0.01	0.08	0.13	0.21	0.44	0.15	0.09
詳細設計	112	0.03	0.10	0.15	0.19	0.40	0.15	0.07
実装・単体テスト	112	0.11	0.19	0.26	0.36	0.79	0.29	0.14
結合テスト	112	0.01	0.14	0.19	0.28	0.58	0.22	0.11
総合テスト	112	0.00	0.10	0.16	0.30	0.58	0.20	0.13

●自然環境からの影響度合い【有り】の方は、【無し】に比べて総合テストにかける工数比率が高い。

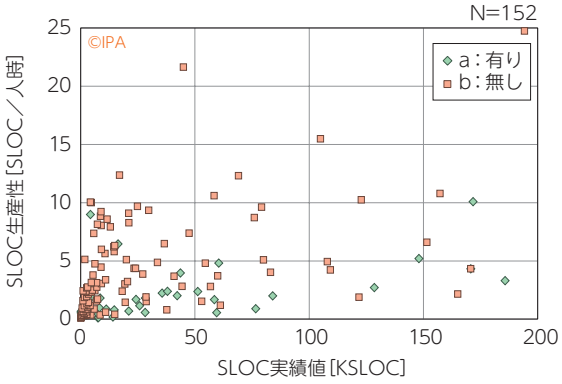
10.3.4　自然環境からの影響度合い別のSLOC規模とSLOC生産性：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模とSLOC生産性の関係を自然環境からの影響度合い別に示す。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-2_自然環境からの影響度合いが回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- SLOC生産性 ＞ 0

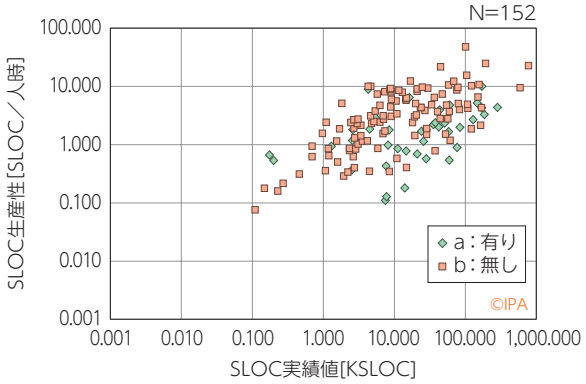
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：SLOC生産性
  （SLOC／実績工数（開発5工程））（導出指標）

図表10.3-9　自然環境からの影響度合い別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



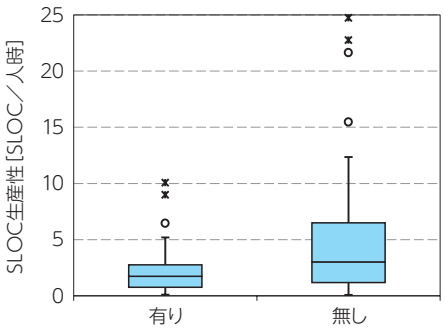
※表示されていないものが4点ある

図表10.3-10　自然環境からの影響度合い別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



※自然環境からの影響度合い別のSLOC規模とSLOC生産性について、対数化した場合の相関係数は次のようになる。
a：有り　R=0.45（P＜0.05）
b：無し　R=0.70（P＜0.05）

図表10.3-11　自然環境からの影響度合い別SLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.3-12　自然環境からの影響度合い別SLOC生産性の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

[SLOC／人時]

自然環境からの影響度合い	N	P25	中央	P75
a：有り	40	0.76	1.75	2.76
b：無し	112	1.18	3.02	6.50

●自然環境からの影響度合い【有り】の方は、【無し】に比べて生産性が低くなる傾向にある。

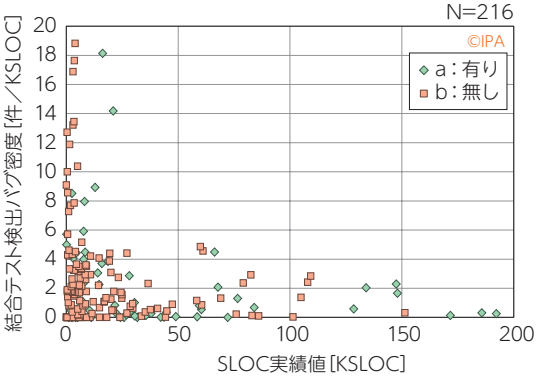
10.3.5 自然環境からの影響度合い別のSLOC規模と結合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模と結合テスト検出バグ密度の関係について、自然環境からの影響度合い別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-2_自然環境からの影響度合いが回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- 10-3-1_結合テスト検出バグ現象数 ≧ 0

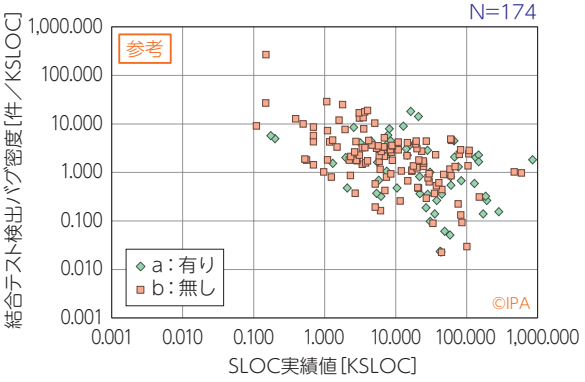
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：結合テスト検出バグ密度（SLOCあたりの検出バグ数）（導出指標）

図表10.3-13 自然環境からの影響度合い別のSLOC規模と結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



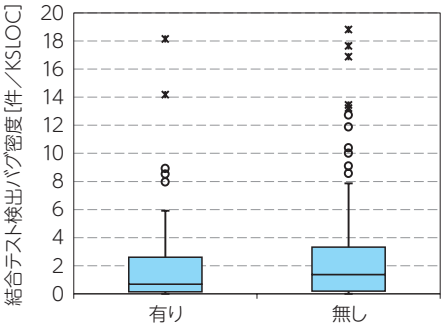
※表示されていないものが9点ある

図表10.3-14 自然環境からの影響度合い別のSLOC規模と結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件/KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

図表10.3-15 自然環境からの影響度合い別結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.3-16 自然環境からの影響度合い別結合テスト検出バグ密度の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

[件／KSLOC]

自然環境からの影響度合い	N	P25	中央	P75
a：有り	67	0.140	0.691	2.606
b：無し	149	0.193	1.372	3.333

●結合テスト検出バグ密度は、自然環境からの影響度合い【有り】【無し】で顕著な違いは見られない。

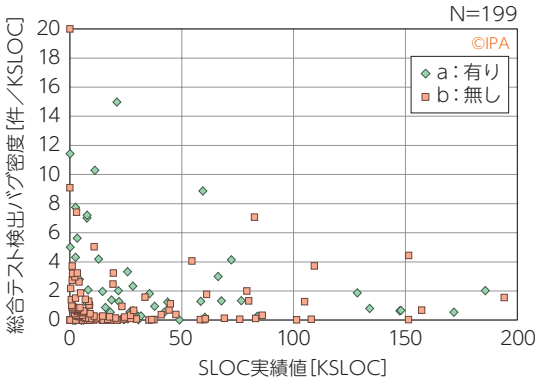
10.3.6 自然環境からの影響度合い別のSLOC規模と総合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模と総合テスト検出バグ密度の関係について、自然環境からの影響度合い別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-2_自然環境からの影響度合いが回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- 10-4-1_総合テスト検出バグ現象数 ≧ 0

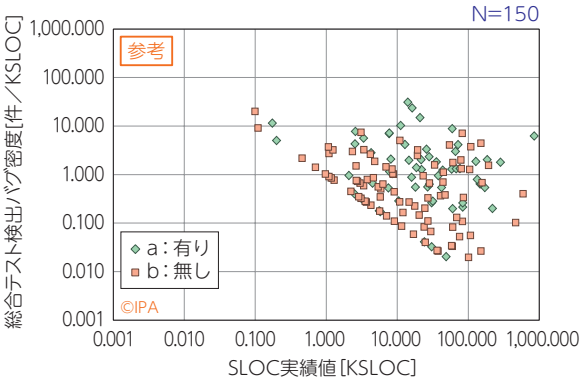
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：総合テスト検出バグ密度（SLOCあたりの検出バグ数）（導出指標）

図表10.3-17 自然環境からの影響度合い別のSLOC規模と総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



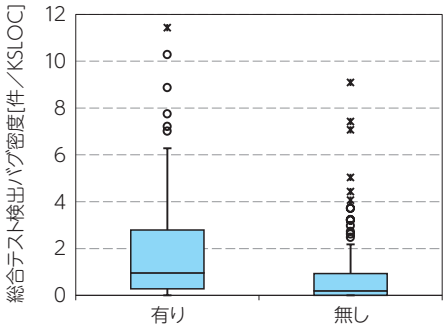
※表示されていないものが7点ある

図表10.3-18 自然環境からの影響度合い別のSLOC規模と総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件/KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

図表10.3-19 自然環境からの影響度合い別総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.3-20 自然環境からの影響度合い別総合テスト検出バグ密度の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

[件／KSLOC]

自然環境からの影響度合い	N	P25	中央	P75
a：有り	69	0.283	0.957	2.795
b：無し	130	0.000	0.189	0.934

●総合テスト検出バグ密度は、自然環境からの影響度合い【有り】の場合に高くなる傾向が見られる。

10.4 ユーザの多様性

この節では、改良(派生)開発で開発5工程(アーキテクチャ設計～総合テスト)の作業の行われたプロジェクトを対象に、ユーザの多様性別による分析を示す。

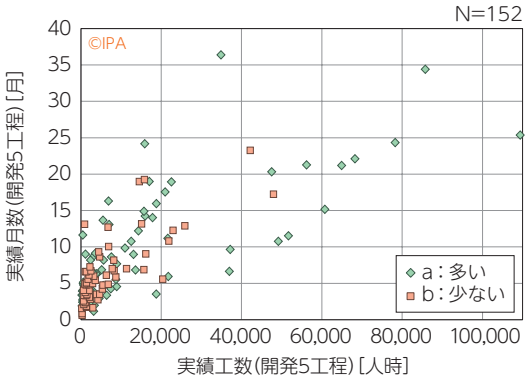
10.4.1 ユーザの多様性別の工数と工期：改良(派生)開発、開発5工程、開発言語C/C++

ここでは、実績工数とその工期(月数)の関係をユーザの多様性別に示す。

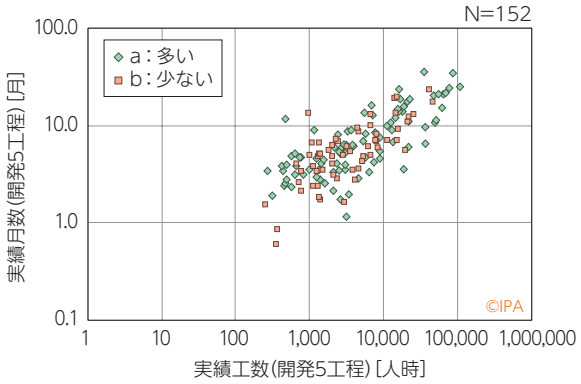
■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良(派生)開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-3_ユーザの多様性が回答されているもの
- 実績工数(開発５工程) ＞ 0
- 実績月数(開発５工程) ＞ 0

■対象データ
- X軸：実績工数(開発5工程)(導出指標)
- Y軸：実績月数(開発5工程)(導出指標)

図表10.4-1 ユーザの多様性別の工数と工期(改良(派生)開発、開発5工程、開発言語C/C++)



図表10.4-2 ユーザの多様性別の工数と工期(改良(派生)開発、開発5工程、開発言語C/C++)対数表示



※ユーザの多様性別の工数と工期について、対数化した場合の相関係数は次のようになる。
a：多い　　R=0.75 (P<0.05)
b：少ない　　R=0.75 (P<0.05)

図表10.4-3 ユーザの多様性別の実績月数の基本統計量(改良(派生)開発、開発5工程、開発言語C/C++)

[月]

ユーザの多様性	N	P25	中央	P75
a：多い	92	3.60	5.85	11.26
b：少ない	60	3.41	5.23	7.47

●ユーザの多様性【多い】【少ない】いずれの場合も、工数の増加に伴い工期が長くなる傾向が出ている。

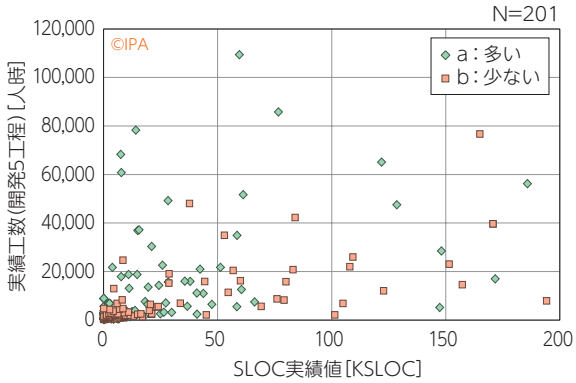
10.4.2　ユーザの多様性別のSLOC規模と工数：改良（派生）開発、開発5工程、開発言語C/C++

ここでは、SLOC規模と工数の関係をユーザの多様性別に示す。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-3_ユーザの多様性が回答されているもの
- 実効SLOC実績値 ＞ 0
- 実績工数（開発5工程）＞ 0

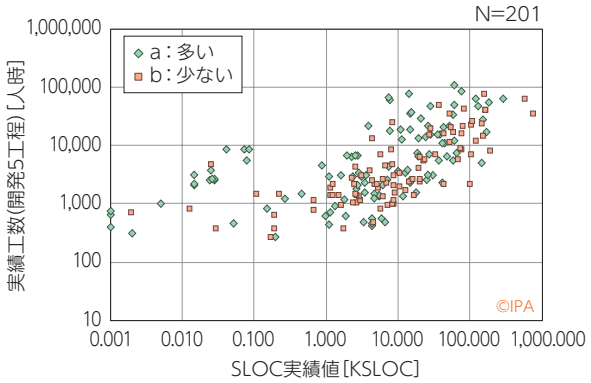
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：実績工数（開発5工程）（導出指標）

図表10.4-4　ユーザの多様性別のSLOC規模と工数（改良（派生）開発、開発5工程、開発言語C/C++）



※表示されていないものが3点ある

図表10.4-5　ユーザの多様性別のSLOC規模と工数（改良（派生）開発、開発5工程、開発言語C/C++）対数表示



※ユーザの多様性別のSLOC規模と工数について、対数化した場合の相関係数は次のようになる。
a：多い　　R=0.57（P<0.05）
b：少ない　　R=0.73（P<0.05）

図表10.4-6　ユーザの多様性別の実績工数の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++）

[人時]

ユーザの多様性	N	P25	中央	P75
a：多い	109	1,564	3,605	14,333
b：少ない	92	1,377	2,545	9,404

●工数の増加はSLOC規模に依存するという一般的な傾向は、ユーザの多様性【少ない】の場合、比較的強く出ているが、ユーザの多様性【多い】では、SLOC規模が小さい場合でも工数が多くかかっているケースが見られるため、その傾向が強いとは言えない。

10.4.3 ユーザの多様性別の工程別工数：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

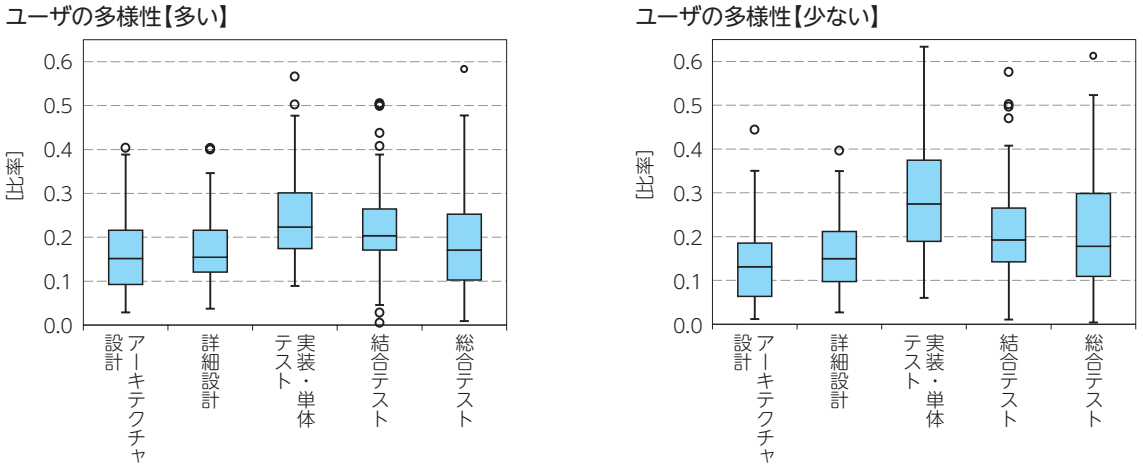
ここでは、工程別の実績工数の比率をユーザの多様性別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-3_ユーザの多様性が回答されているもの
- 工程別の実績工数にすべて記入があり、各値が0より大きい
- 実効SLOC実績値 ≧ 0.1KSLOC

■対象データ
- 9-4-30_社内実績工数_アーキテクチャ設計、
  9-4-31_社内実績工数_詳細設計、
  9-4-32_社内実績工数_実装・単体テスト、
  9-4-33_社内実績工数_結合テスト、
  9-4-34_社内実績工数_総合テスト
- 9-8-2_外部委託工数_アーキテクチャ設計、
  9-8-3_外部委託工数_詳細設計、
  9-8-4_外部委託工数_実装・単体テスト、
  9-8-5_外部委託工数_結合テスト、
  9-8-6_外部委託工数_総合テスト

※各工程の実績工数は、社内、外部委託の実績工数合計の人時換算値を使用。

図表10.4-7　ユーザの多様性別の工程別の実績工数の比率（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.4-8　ユーザの多様性別の工程別の実績工数の比率の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

ユーザの多様性【多い】　[比率]

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	91	0.03	0.09	0.15	0.22	0.40	0.16	0.09
詳細設計	91	0.04	0.12	0.15	0.22	0.67	0.18	0.09
実装・単体テスト	91	0.09	0.17	0.22	0.30	0.57	0.25	0.09
結合テスト	91	0.01	0.17	0.20	0.26	0.51	0.22	0.10
総合テスト	91	0.01	0.10	0.17	0.25	0.58	0.20	0.12

ユーザの多様性【少ない】　[比率]

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	88	0.01	0.06	0.13	0.19	0.44	0.14	0.09
詳細設計	88	0.03	0.10	0.15	0.21	0.40	0.16	0.08
実装・単体テスト	88	0.06	0.19	0.28	0.37	0.79	0.29	0.14
結合テスト	88	0.01	0.14	0.19	0.27	0.58	0.20	0.10
総合テスト	88	0.00	0.11	0.18	0.30	0.61	0.21	0.14

●ユーザの多様性【少ない】の場合、実装・単体テストにかかる工数比率は、【多い】と比較して少し高い傾向が見られる。

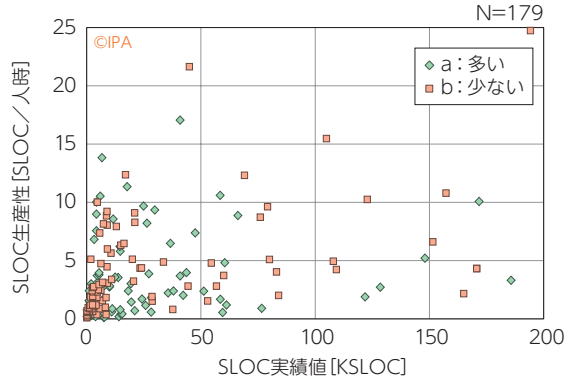
10.4.4 ユーザの多様性別のSLOC規模とSLOC生産性：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模とSLOC生産性の関係をユーザの多様性別に示す。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-3_ユーザの多様性が回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- SLOC生産性 ＞ 0

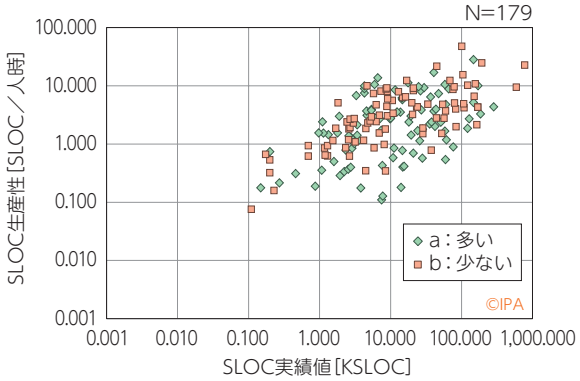
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：SLOC生産性
  （SLOC／実績工数（開発5工程））（導出指標）

図表10.4-9 ユーザの多様性別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



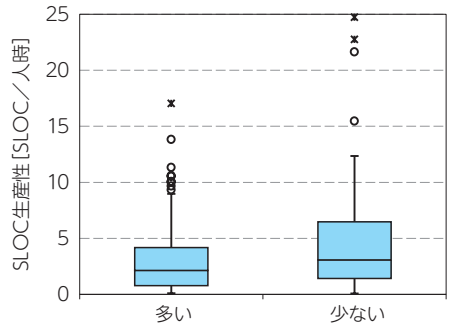
※表示されていないものが5点ある

図表10.4-10 ユーザの多様性別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



※ユーザの多様性別のSLOC規模とSLOC生産性について、対数化した場合の相関係数は次のようになる。
a：多い　R=0.46（P<0.05）
b：少ない　R=0.71（P<0.05）

図表10.4-11 ユーザの多様性別SLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.4-12 ユーザの多様性別SLOC生産性の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

［SLOC／人時］

ユーザの多様性	N	P25	中央	P75
a：多い	91	0.79	2.14	4.17
b：少ない	88	1.43	3.08	6.49

●ユーザの多様性別の生産性について、顕著な違いは見られない。

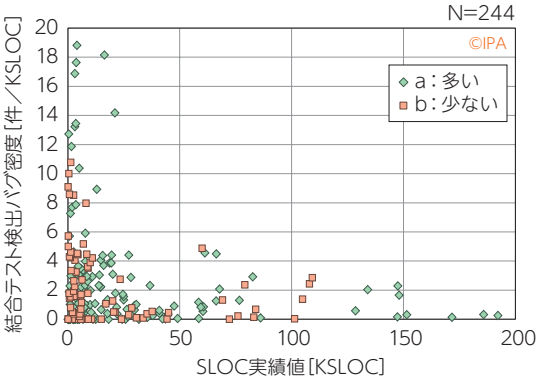
10.4.5 ユーザの多様性別のSLOC規模と結合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模と結合テスト検出バグ密度の関係について、ユーザの多様性別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-3_ユーザの多様性が回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- 10-3-1_結合テスト検出バグ現象数 ≧ 0

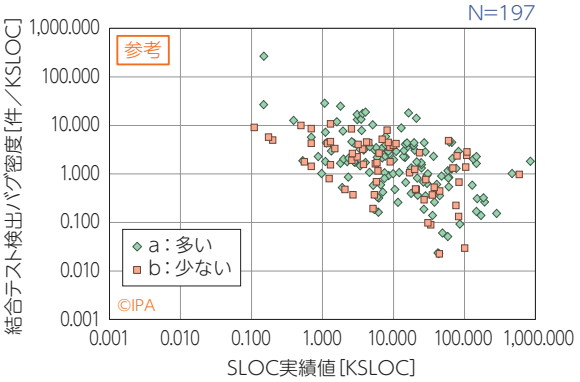
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：結合テスト検出バグ密度（SLOCあたりの検出バグ数）（導出指標）

図表10.4-13 ユーザの多様性別のSLOC規模と結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



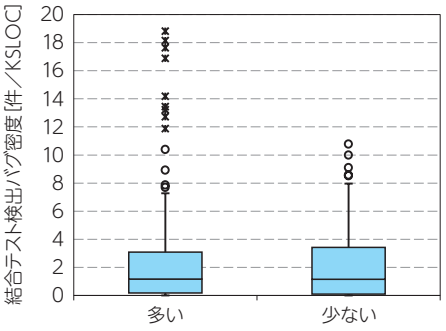
※表示されていないものが9点ある

図表10.4-14 ユーザの多様性別のSLOC規模と結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件/KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

図表10.4-15 ユーザの多様性別結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.4-16 ユーザの多様性別結合テスト検出バグ密度の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

[件／KSLOC]

ユーザの多様性	N	P25	中央	P75
a：多い	161	0.186	1.165	3.089
b：少ない	83	0.093	1.152	3.425

●ユーザの多様性別の結合テスト検出バグ密度について、顕著な違いは見られない。

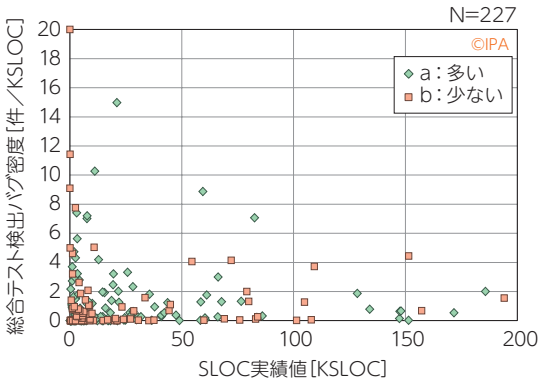
10.4.6 ユーザの多様性別のSLOC規模と総合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模と総合テスト検出バグ密度の関係について、ユーザの多様性別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-3_ユーザの多様性が回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- 10-4-1_総合テスト検出バグ現象数 ≧ 0

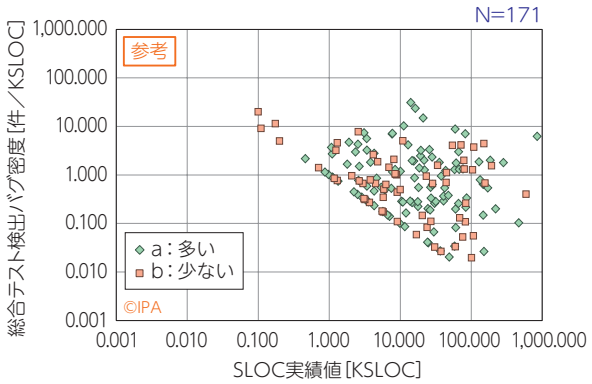
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：総合テスト検出バグ密度（SLOCあたりの検出バグ数）（導出指標）

図表10.4-17 ユーザの多様性別のSLOC規模と総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



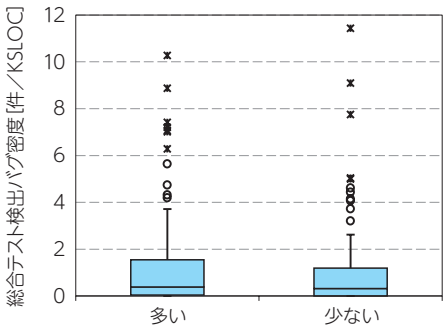
※表示されていないものが7点ある

図表10.4-18 ユーザの多様性別のSLOC規模と総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件/KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

図表10.4-19 ユーザの多様性別総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.4-20 ユーザの多様性別総合テスト検出バグ密度の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

［件／KSLOC］

ユーザの多様性	N	P25	中央	P75
a：多い	140	0.039	0.389	1.549
b：少ない	87	0.000	0.323	1.189

●ユーザの多様性別の総合テスト検出バグ密度について、顕著な違いは見られない。

10.5 法規等による規制度合い

この節では、改良(派生)開発で開発5工程(アーキテクチャ設計～総合テスト)の作業の行われたプロジェクトを対象に、法規等による規制度合い別による分析を示す。

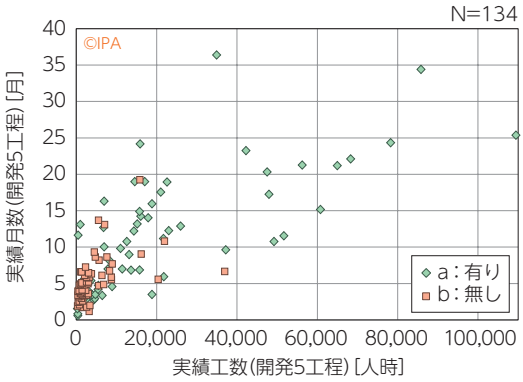
10.5.1 法規等による規制度合い別の工数と工期：改良(派生)開発、開発5工程、開発言語C/C++

ここでは、実績工数とその工期(月数)の関係を法規等による規制度合い別に示す。

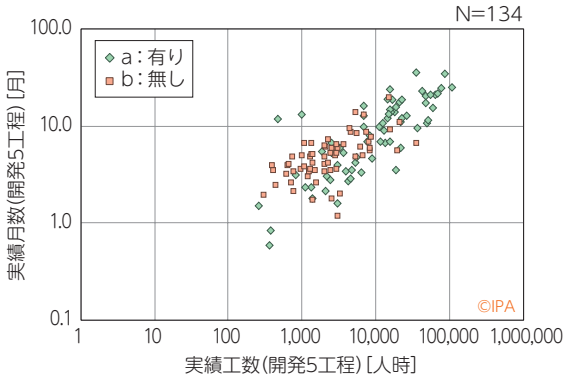
■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良(派生)開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-4_法規等による規制度合いが回答されているもの
- 実績工数(開発5工程) ＞ 0
- 実績月数(開発5工程) ＞ 0

■対象データ
- X軸：実績工数(開発5工程)(導出指標)
- Y軸：実績月数(開発5工程)(導出指標)

図表10.5-1 法規等による規制度合い別の工数と工期(改良(派生)開発、開発5工程、開発言語C/C++)



図表10.5-2 法規等による規制度合い別の工数と工期(改良(派生)開発、開発5工程、開発言語C/C++)対数表示



※法規等による規制度合い別の工数と工期について、対数化した場合の相関係数は次のようになる。
a：有り　R=0.80 (P<0.05)
b：無し　R=0.60 (P<0.05)

図表10.5-3 法規等による規制度合い別の実績月数の基本統計量(改良(派生)開発、開発5工程、開発言語C/C++)

[月]

法規等による規制度合い	N	P25	中央	P75
a：有り	69	3.97	9.63	15.17
b：無し	65	3.43	4.90	6.33

●法規等による規制度合い【有り】の場合、【無し】に比べて工数の増加に伴い工期が長くなる傾向が顕著である。

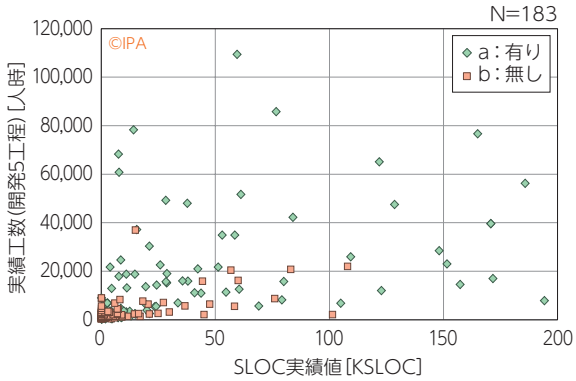
10.5.2 法規等による規制度合い別のSLOC規模と工数：改良（派生）開発、開発5工程、開発言語C/C++

ここでは、SLOC規模と工数の関係を法規等による規制度合い別に示す。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-4_法規等による規制度合いが回答されているもの
- 実効SLOC実績値 ＞ 0
- 実績工数（開発5工程）＞ 0

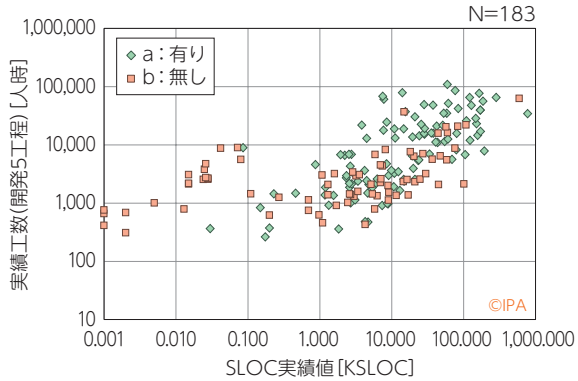
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：実績工数（開発5工程）（導出指標）

図表10.5-4 法規等による規制度合い別のSLOC規模と工数（改良（派生）開発、開発5工程、開発言語C/C++）



※表示されていないものが3点ある

図表10.5-5 法規等による規制度合い別のSLOC規模と工数（改良（派生）開発、開発5工程、開発言語C/C++）対数表示



※法規等による規制度合い別のSLOC規模と工数について、対数化した場合の相関係数は次のようになる。
a：有り　R=0.71（P<0.05）
b：無し　R=0.51（P<0.05）

図表10.5-6 法規等による規制度合い別の実績工数の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++）

[人時]

法規等による規制度合い	N	P25	中央	P75
a：有り	107	2,272	6,892	21,720
b：無し	76	1,324	2,330	4,924

●工数の増加はSLOC規模に依存するという一般的な傾向は、法規等による規制【有り】の場合に比較的強く出ているが、法規等による規制【無し】の場合は、弱いと言える。

10.5.3 法規等による規制度合い別の工程別工数：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

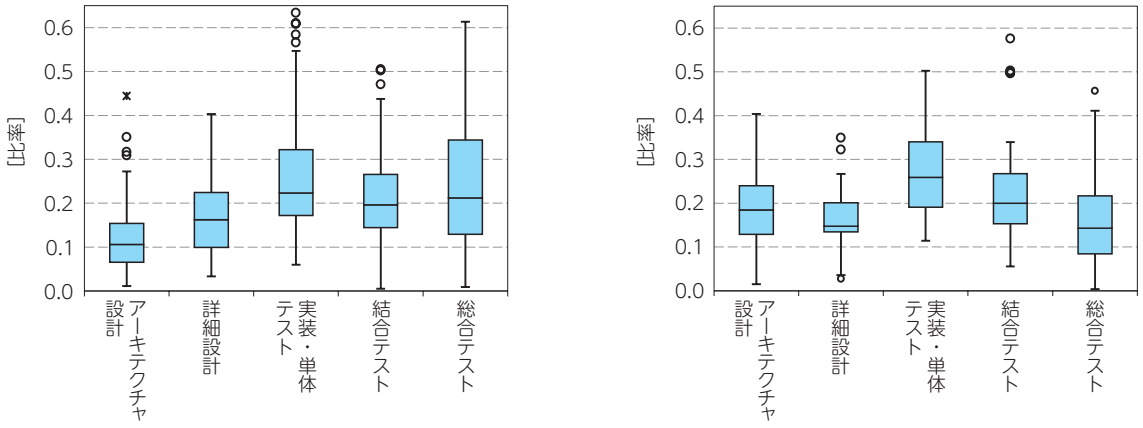
ここでは、工程別の実績工数の比率を法規等による規制度合い別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-4_法規等による規制度合いが回答されているもの
- 工程別の実績工数にすべて記入があり、各値が0より大きい
- 実効SLOC実績値 ≧ 0.1KSLOC

■対象データ
- 9-4-30_社内実績工数_アーキテクチャ設計、
  9-4-31_社内実績工数_詳細設計、
  9-4-32_社内実績工数_実装・単体テスト、
  9-4-33_社内実績工数_結合テスト、
  9-4-34_社内実績工数_総合テスト
- 9-8-2_外部委託工数_アーキテクチャ設計、
  9-8-3_外部委託工数_詳細設計、
  9-8-4_外部委託工数_実装・単体テスト、
  9-8-5_外部委託工数_結合テスト、
  9-8-6_外部委託工数_総合テスト

※各工程の実績工数は、社内、外部委託の実績工数合計の人時換算値を使用。

図表10.5-7　法規等による規制度合い別の工程別の実績工数の比率（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.5-8　法規等による規制度合い別の工程別の実績工数の比率の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

法規等による規制度合い【有り】　[比率]

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	105	0.01	0.07	0.11	0.15	0.44	0.12	0.08
詳細設計	105	0.03	0.10	0.16	0.22	0.67	0.17	0.10
実装・単体テスト	105	0.06	0.17	0.22	0.32	0.63	0.26	0.13
結合テスト	105	0.01	0.14	0.20	0.27	0.51	0.21	0.10
総合テスト	105	0.01	0.13	0.21	0.34	0.61	0.24	0.14

法規等による規制度合い【無し】　[比率]

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	57	0.02	0.13	0.18	0.24	0.40	0.19	0.09
詳細設計	57	0.03	0.13	0.15	0.20	0.35	0.16	0.07
実装・単体テスト	57	0.11	0.19	0.26	0.34	0.79	0.28	0.12
結合テスト	57	0.06	0.15	0.20	0.27	0.58	0.22	0.11
総合テスト	57	0.00	0.08	0.14	0.22	0.46	0.16	0.10

●法規等による規制度合い【有り】の方が、総合テストにかける工数比率が高い傾向が見られる。また、アーキテクチャ設計にかける工数比率では【無し】の方が、高い傾向が見られる。

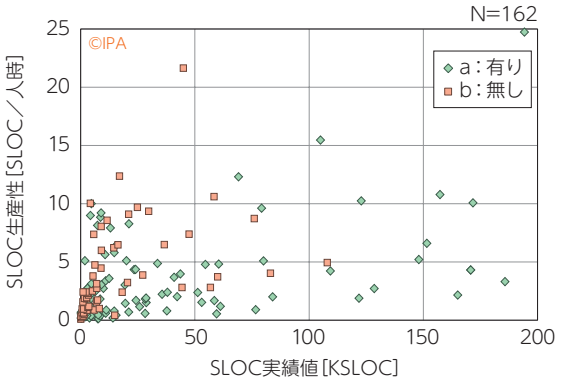
10.5.4 法規等による規制度合い別のSLOC規模とSLOC生産性：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模とSLOC生産性の関係を法規等による規制度合い別に示す。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-4_法規等による規制度合いが回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- SLOC生産性 ＞ 0

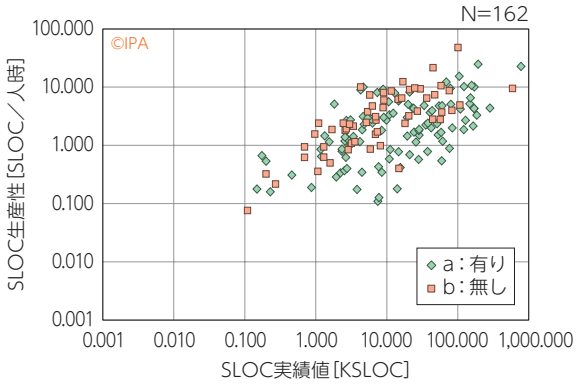
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：SLOC生産性
  （SLOC／実績工数（開発5工程））（導出指標）

図表10.5-9 法規等による規制度合い別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



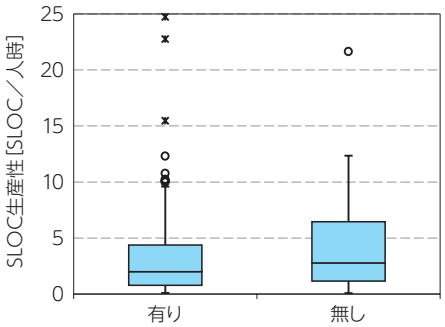
※表示されていないものが4点ある

図表10.5-10 法規等による規制度合い別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



※法規等による規制度合い別のSLOC規模とSLOC生産性について、対数化した場合の相関係数は次のようになる。
a：有り　R=0.57（P<0.05）
b：無し　R=0.77（P<0.05）

図表10.5-11 法規等による規制度合い別SLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.5-12 法規等による規制度合い別SLOC生産性の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

［SLOC／人時］

法規等による規制度合い	N	P25	中央	P75
a：有り	105	0.79	2.01	4.38
b：無し	57	1.16	2.78	6.47

●法規等による規制度合い【有り】の方が、生産性が低くなる傾向が見られる。

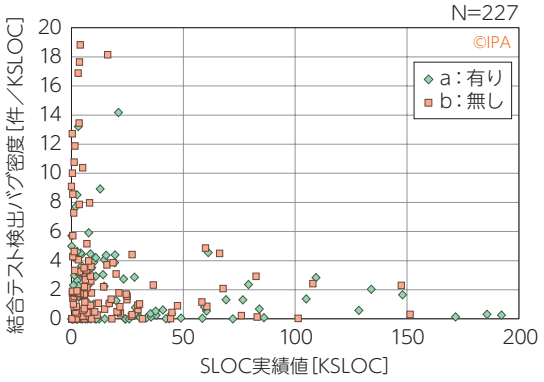
10.5.5 法規等による規制度合い別のSLOC規模と結合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模と結合テスト検出バグ密度の関係について、法規等による規制度合い別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-4_法規等による規制度合いが回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- 10-3-1_結合テスト検出バグ現象数 ≧ 0

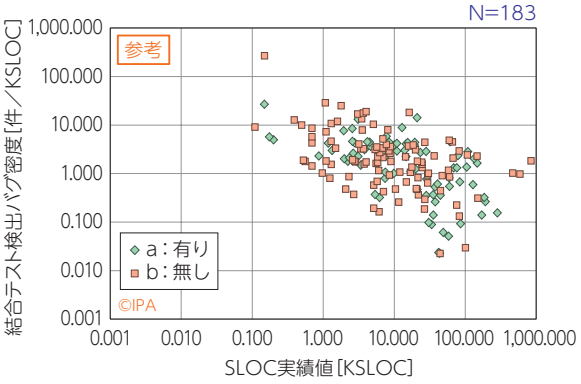
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：結合テスト検出バグ密度（SLOCあたりの検出バグ数）（導出指標）

図表10.5-13 法規等による規制度合い別のSLOC規模と結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



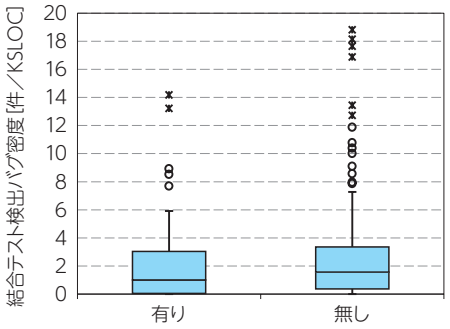
※表示されていないものが9点ある

図表10.5-14 法規等による規制度合い別のSLOC規模と結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件/KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

図表10.5-15 法規等による規制度合い別結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.5-16 法規等による規制度合い別結合テスト検出バグ密度の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

[件／KSLOC]

法規等による規制度合い	N	P25	中央	P75
a：有り	101	0.051	1.010	3.046
b：無し	126	0.366	1.571	3.361

●結合テスト検出バグ密度は、法規等による規制度合い【有り】【無し】で顕著な違いが見られない。

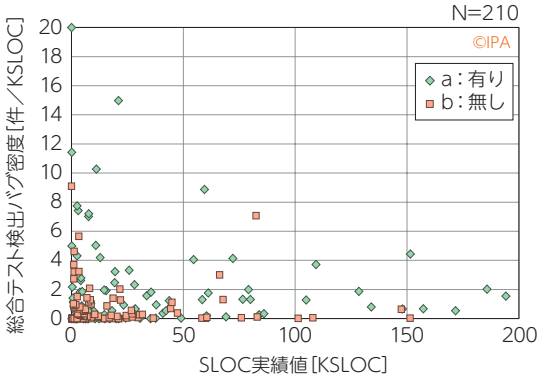
10.5.6 法規等による規制度合い別のSLOC規模と総合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模と総合テスト検出バグ密度の関係について、法規等による規制度合い別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-4_法規等による規制度合いが回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- 10-4-1_総合テスト検出バグ現象数 ≧ 0

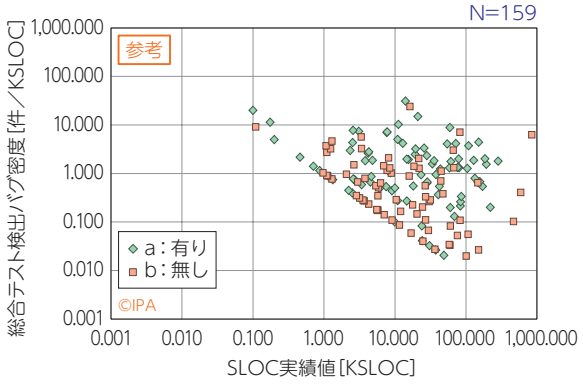
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：総合テスト検出バグ密度（SLOCあたりの検出バグ数）（導出指標）

図表10.5-17 法規等による規制度合い別のSLOC規模と総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



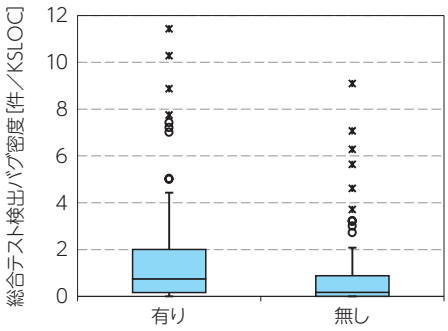
※表示されていないものが7点ある

図表10.5-18 法規等による規制度合い別のSLOC規模と総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件/KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

図表10.5-19 法規等による規制度合い別総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.5-20 法規等による規制度合い別総合テスト検出バグ密度の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

[件／KSLOC]

法規等による規制度合い	N	P25	中央	P75
a：有り	107	0.164	0.752	2.008
b：無し	103	0.000	0.175	0.883

●総合テスト検出バグ密度は、法規等による規制度合い【有り】の場合に高くなる傾向が見られる。

10.6 M２Mの有無

この節では、改良（派生）開発で開発5工程（アーキテクチャ設計～総合テスト）の作業の行われたプロジェクトを対象に、M２Mの有無別による分析を示す。

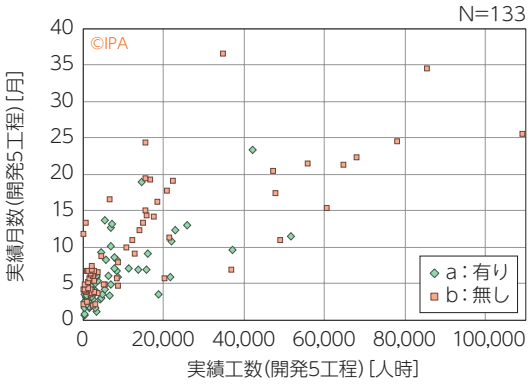
10.6.1 M２Mの有無別の工数と工期：改良（派生）開発、開発5工程、開発言語C/C++

ここでは、実績工数とその工期（月数）の関係をM２Mの有無別に示す。

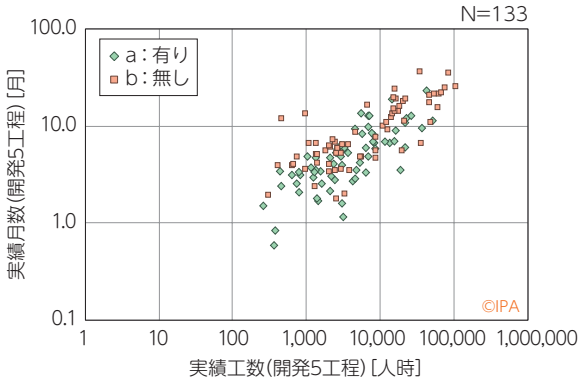
■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-5_M２Mの有無が回答されているもの
- 実績工数（開発5工程）＞０
- 実績月数（開発5工程）＞０

■対象データ
- X軸：実績工数（開発5工程）（導出指標）
- Y軸：実績月数（開発5工程）（導出指標）

図表10.6-1　M２Mの有無別の工数と工期（改良（派生）開発、開発5工程、開発言語C/C++）



図表10.6-2　M２Mの有無別の工数と工期（改良（派生）開発、開発5工程、開発言語C/C++）対数表示



※M２Mの有無別の工数と工期について、対数化した場合の相関係数は次のようになる。
a：有り　　R=0.76（P<0.05）
b：無し　　R=0.77（P<0.05）

図表10.6-3　M２Mの有無別の実績月数の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++）

［月］

M２Mの有無	N	P25	中央	P75
a：有り	66	3.12	4.83	7.00
b：無し	67	4.73	6.63	15.03

●M２M【有り】【無し】いずれの場合も、工数の増加に伴い工期が長くなる傾向がやや強く出ている。

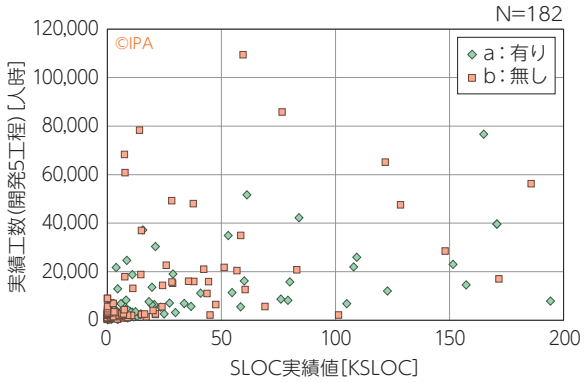
10.6.2 M2Mの有無別のSLOC規模と工数：改良（派生）開発、開発5工程、開発言語C/C++

ここでは、SLOC規模と工数の関係をM2Mの有無別に示す。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-5_M2Mの有無が回答されているもの
- 実効SLOC実績値 ＞ 0
- 実績工数（開発5工程）＞ 0

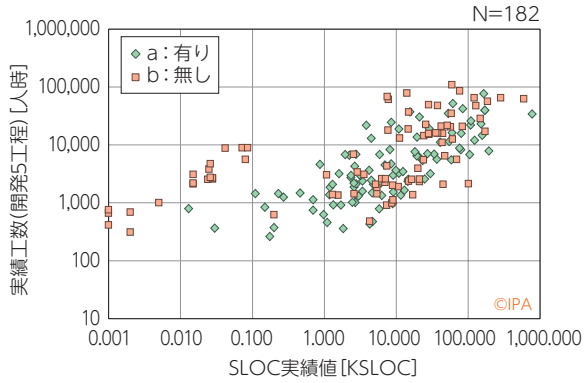
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：実績工数（開発5工程）（導出指標）

図表10.6-4 M2Mの有無別のSLOC規模と工数（改良（派生）開発、開発5工程、開発言語C/C++）



※表示されていないものが3点ある

図表10.6-5 M2Mの有無別のSLOC規模と工数（改良（派生）開発、開発5工程、開発言語C/C++）対数表示



※M2Mの有無別のSLOC規模と工数について、対数化した場合の相関係数は次のようになる。
a：有り　R=0.75（P<0.05）
b：無し　R=0.61（P<0.05）

図表10.6-6 M2Mの有無別の実績工数の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++）

[人時]

M2Mの有無	N	P25	中央	P75
a：有り	99	1,442	3,193	8,518
b：無し	83	2,106	4,332	18,339

●工数の増加はSLOC規模に依存するという一般的な傾向は、M2M【有り】の場合、比較的強く出ているが、M2M【無し】ではSLOC規模が小さい場合でも工数が多くかかっているケースが見られるため、その傾向が強いとは言えない。

10.6.3 M２Mの有無別の工程別工数：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

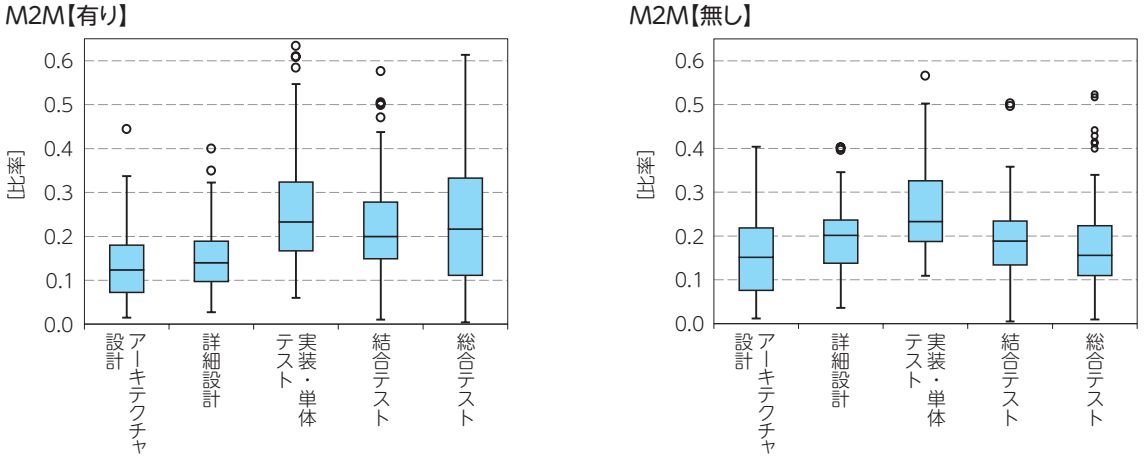
ここでは、工程別の実績工数の比率をM２Mの有無別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-5_M２Mの有無が回答されているもの
- 工程別の実績工数にすべて記入があり、各値が0より大きい
- 実効SLOC実績値 ≧ 0.1KSLOC

■対象データ
- 9-4-30_社内実績工数_アーキテクチャ設計、
  9-4-31_社内実績工数_詳細設計、
  9-4-32_社内実績工数_実装・単体テスト、
  9-4-33_社内実績工数_結合テスト、
  9-4-34_社内実績工数_総合テスト
- 9-8-2_外部委託工数_アーキテクチャ設計、
  9-8-3_外部委託工数_詳細設計、
  9-8-4_外部委託工数_実装・単体テスト、
  9-8-5_外部委託工数_結合テスト、
  9-8-6_外部委託工数_総合テスト

※各工程の実績工数は、社内、外部委託の実績工数合計の人時換算値を使用。

図表10.6-7　M２Mの有無別の工程別の実績工数の比率（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.6-8　M２Mの有無別の工程別の実績工数の比率の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

M２M【有り】　[比率]

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	97	0.02	0.07	0.12	0.18	0.44	0.13	0.08
詳細設計	97	0.03	0.10	0.14	0.19	0.40	0.15	0.07
実装・単体テスト	97	0.06	0.17	0.23	0.32	0.63	0.27	0.13
結合テスト	97	0.01	0.15	0.20	0.28	0.58	0.22	0.11
総合テスト	97	0.00	0.11	0.22	0.33	0.61	0.23	0.14

M２M【無し】　[比率]

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	64	0.01	0.08	0.15	0.22	0.40	0.16	0.10
詳細設計	64	0.04	0.14	0.20	0.24	0.67	0.20	0.10
実装・単体テスト	64	0.11	0.19	0.23	0.33	0.79	0.27	0.12
結合テスト	64	0.01	0.13	0.19	0.23	0.50	0.19	0.10
総合テスト	64	0.01	0.11	0.16	0.22	0.52	0.18	0.12

●M２Mの有無【有り】の方が、総合テストにかける工数比率が高い傾向が見られる。

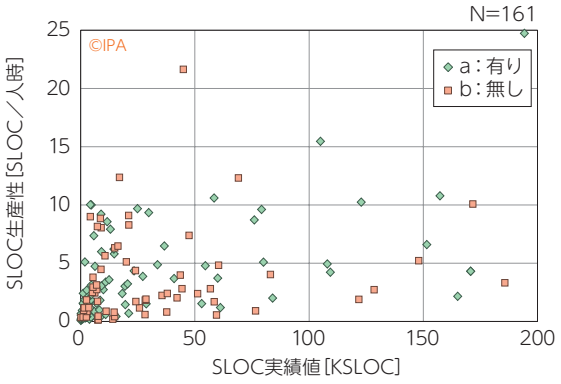
10.6.4 M２Mの有無別のSLOC規模とSLOC生産性：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模とSLOC生産性の関係をM２Mの有無別に示す。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-5_M２Mの有無が回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- SLOC生産性 ＞ 0

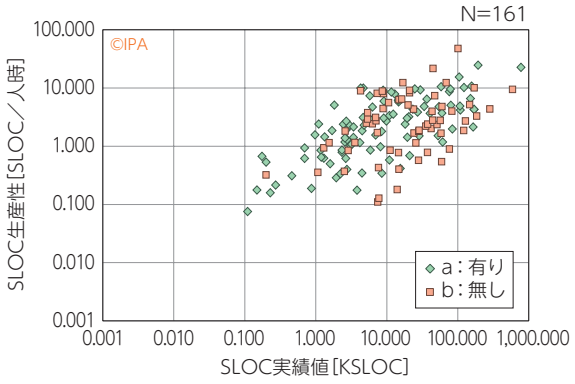
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：SLOC生産性
  （SLOC／実績工数（開発5工程））（導出指標）

図表10.6-9　M２Mの有無別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



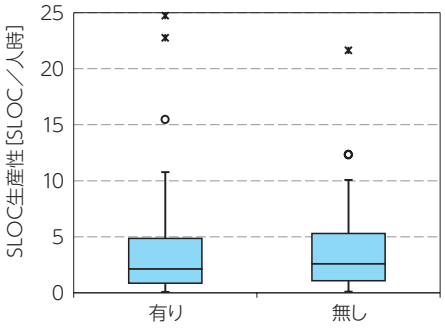
※表示されていないものが4点ある

図表10.6-10　M２Mの有無別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



※M２Mの有無別のSLOC規模とSLOC生産性について、対数化した場合の相関係数は次のようになる。
a：有り　　R=0.71（P＜0.05）
b：無し　　R=0.41（P＜0.05）

図表10.6-11　M２Mの有無別SLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.6-12　M２Mの有無別SLOC生産性の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

［SLOC／人時］

M２Mの有無	N	P25	中央	P75
a：有り	97	0.86	2.15	4.86
b：無し	64	1.09	2.59	5.30

●M２Mの有無別の生産性について、違いは見られない。

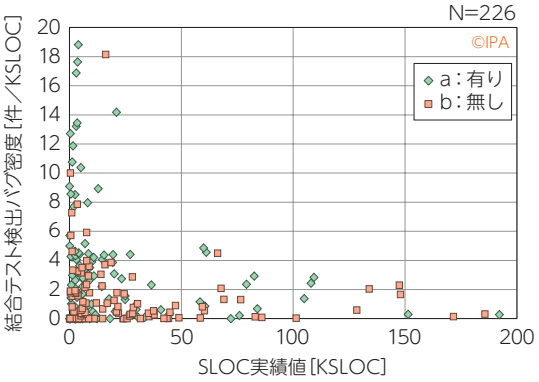
10.6.5 M２Mの有無別のSLOC規模と結合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模と結合テスト検出バグ密度の関係について、M２Mの有無別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-5_M２Mの有無が回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- 10-3-1_結合テスト検出バグ現象数 ≧ 0

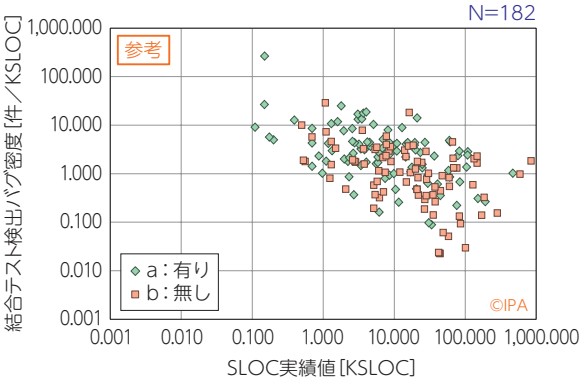
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：結合テスト検出バグ密度（SLOCあたりの検出バグ数）（導出指標）

図表10.6-13　M２Mの有無別のSLOC規模と結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



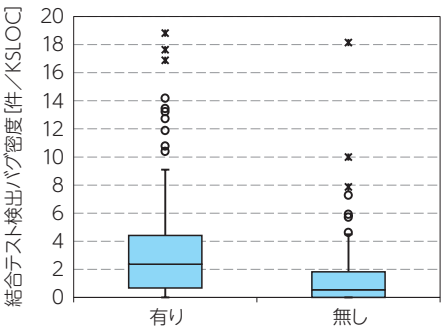
※表示されていないものが9点ある

図表10.6-14　M２Mの有無別のSLOC規模と結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件/KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

図表10.6-15　M２Mの有無別結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.6-16　M２Mの有無別結合テスト検出バグ密度の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

［件／KSLOC］

M２Mの有無	N	P25	中央	P75
a：有り	113	0.679	2.363	4.412
b：無し	113	0.000	0.546	1.829

●M２Mの有無【有り】の方が、結合テスト検出バグ密度が高い傾向が見られる。

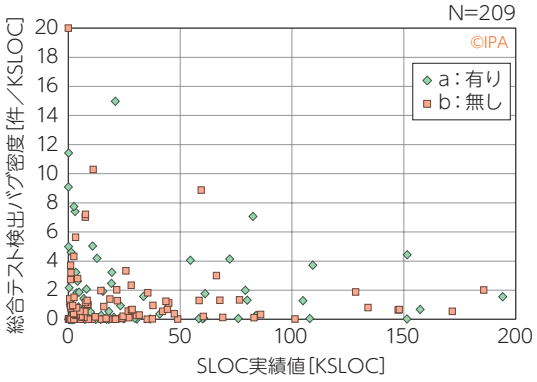
10.6.6 M2Mの有無別のSLOC規模と総合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模と総合テスト検出バグ密度の関係について、M2Mの有無別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-5_M2Mの有無が回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- 10-4-1_総合テスト検出バグ現象数 ≧ 0

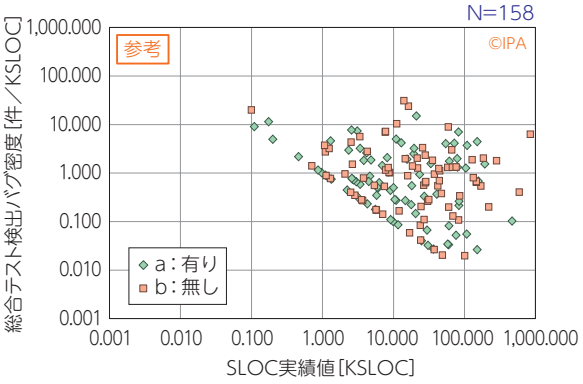
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：総合テスト検出バグ密度（SLOCあたりの検出バグ数）（導出指標）

図表10.6-17 M2Mの有無別のSLOC規模と総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



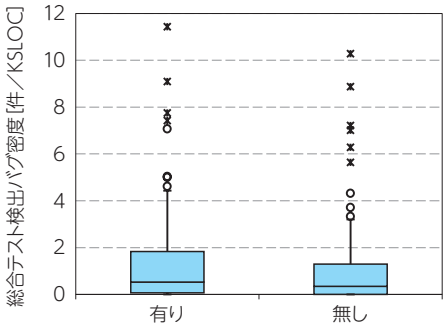
※表示されていないものが5点ある

図表10.6-18 M2Mの有無別のSLOC規模と総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件/KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

図表10.6-19 M2Mの有無別総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.6-20 M2Mの有無別総合テスト検出バグ密度の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

[件／KSLOC]

M2Mの有無	N	P25	中央	P75
a：有り	96	0.064	0.530	1.837
b：無し	113	0.000	0.348	1.299

●総合テスト検出バグ密度は、M2Mの有無による傾向の違いは見られない。

10.7 ネットワーク接続の有無

この節では、改良(派生)開発で開発5工程(アーキテクチャ設計～総合テスト)の作業の行われたプロジェクトを対象に、ネットワーク接続の有無別による分析を示す。

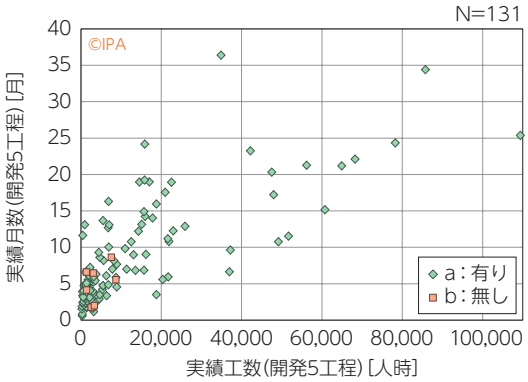
10.7.1 ネットワーク接続の有無別の工数と工期：改良(派生)開発、開発5工程、開発言語C/C++

ここでは、実績工数とその工期(月数)の関係をネットワーク接続の有無別に示す。

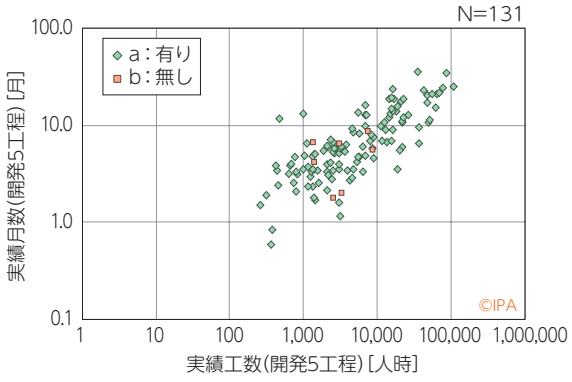
■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良(派生)開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-6_ネットワーク接続の有無が回答されているもの
- 実績工数(開発5工程) ＞ 0
- 実績月数(開発5工程) ＞ 0

■対象データ
- X軸：実績工数(開発5工程)(導出指標)
- Y軸：実績月数(開発5工程)(導出指標)

図表10.7-1 ネットワーク接続の有無別の工数と工期(改良(派生)開発、開発5工程、開発言語C/C++)



図表10.7-2 ネットワーク接続の有無別の工数と工期(改良(派生)開発、開発5工程、開発言語C/C++)対数表示



※ネットワーク接続【a：有り】の工数と工期について、対数化した場合の相関係数は、R=0.78 (P<0.05)となる。

図表10.7-3 ネットワーク接続の有無別の実績月数の基本統計量(改良(派生)開発、開発5工程、開発言語C/C++)

[月]

ネットワーク接続の有無	N	P25	中央	P75
a：有り	124	3.50	5.87	11.56
b：無し	7	3.05	5.53	6.53

●ネットワーク接続の有無別の工数と工期の関係について、標本数が少ないため、比較はできない。

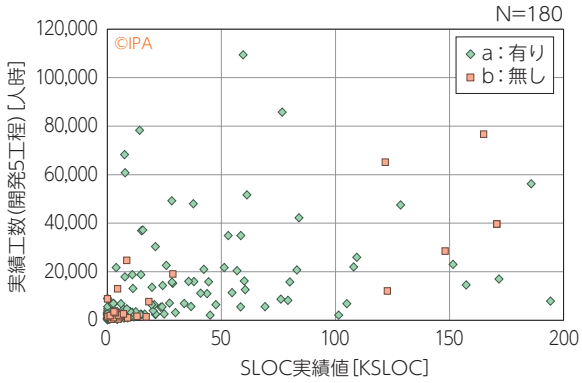
10.7.2 ネットワーク接続の有無別のSLOC規模と工数：改良（派生）開発、開発5工程、開発言語C/C++

ここでは、SLOC規模と工数の関係をネットワーク接続の有無別に示す。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-6_ネットワーク接続の有無が回答されているもの
- 実効SLOC実績値 ＞ 0
- 実績工数（開発5工程）＞ 0

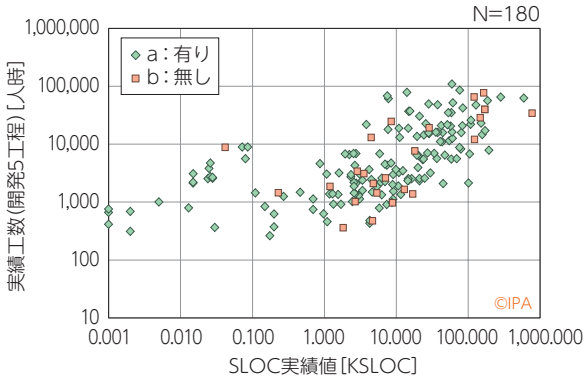
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：実績工数（開発5工程）（導出指標）

図表10.7-4 ネットワーク接続の有無別のSLOC規模と工数（改良（派生）開発、開発5工程、開発言語C/C++）



※表示されていないものが3点ある

図表10.7-5 ネットワーク接続の有無別のSLOC規模と工数（改良（派生）開発、開発5工程、開発言語C/C++）対数表示



※ネットワーク接続の有無別のSLOC規模と工数について、対数化した場合の相関係数は次のようになる。
a：有り　R=0.62（P<0.05）
b：無し　R=0.64（P<0.05）

図表10.7-6 ネットワーク接続の有無別の実績工数の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++）

[人時]

ネットワーク接続の有無	N	P25	中央	P75
a：有り	155	1,558	3,626	14,462
b：無し	25	1,440	3,400	24,624

●工数の増加はSLOC規模に依存するという一般的な傾向は、ネットワーク接続【有り】【無し】いずれの場合も顕著ではない。

10.7.3 ネットワーク接続の有無別の工程別工数：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

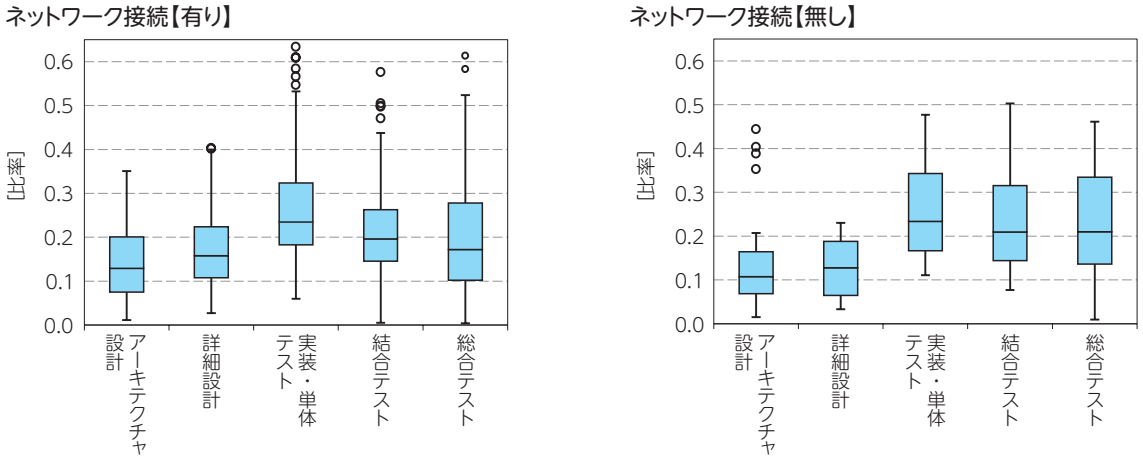
ここでは、工程別の実績工数の比率をネットワーク接続の有無別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-6_ネットワーク接続の有無が回答されているもの
- 工程別の実績工数にすべて記入があり、各値が0より大きい
- 実効SLOC実績値 ≧ 0.1KSLOC

■対象データ
- 9-4-30_社内実績工数_アーキテクチャ設計、9-4-31_社内実績工数_詳細設計、9-4-32_社内実績工数_実装・単体テスト、9-4-33_社内実績工数_結合テスト、9-4-34_社内実績工数_総合テスト
- 9-8-2_外部委託工数_アーキテクチャ設計、9-8-3_外部委託工数_詳細設計、9-8-4_外部委託工数_実装・単体テスト、9-8-5_外部委託工数_結合テスト、9-8-6_外部委託工数_総合テスト

※各工程の実績工数は、社内、外部委託の実績工数合計の人時換算値を使用。

図表10.7-7　ネットワーク接続の有無別の工程別の実績工数の比率（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.7-8　ネットワーク接続の有無別の工程別の実績工数の比率の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

ネットワーク接続【有り】　[比率]

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	135	0.01	0.08	0.13	0.20	0.35	0.14	0.08
詳細設計	135	0.03	0.11	0.16	0.22	0.67	0.18	0.09
実装・単体テスト	135	0.06	0.18	0.23	0.32	0.79	0.27	0.13
結合テスト	135	0.01	0.15	0.20	0.26	0.58	0.21	0.10
総合テスト	135	0.00	0.10	0.17	0.28	0.61	0.20	0.14

ネットワーク接続【無し】　[比率]

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	24	0.02	0.07	0.11	0.16	0.44	0.15	0.12
詳細設計	24	0.03	0.06	0.13	0.19	0.23	0.12	0.06
実装・単体テスト	24	0.11	0.17	0.23	0.34	0.48	0.26	0.11
結合テスト	24	0.08	0.14	0.21	0.32	0.50	0.23	0.12
総合テスト	24	0.01	0.14	0.21	0.33	0.46	0.23	0.13

●ネットワーク接続【有り】の場合は、【無し】に比べて「アーキテクチャ設計」や「詳細設計」の工数比率が高く、「結合テスト」や「総合テスト」の工数比率が低い傾向が見られる。

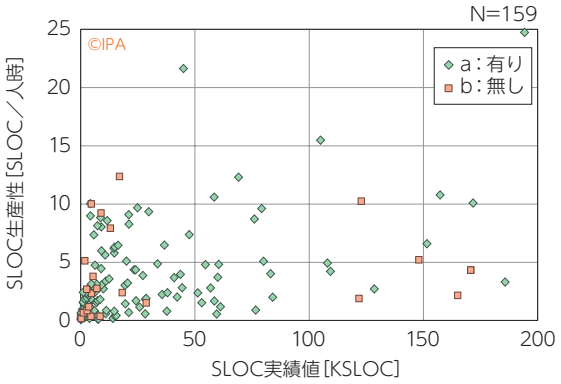
10.7.4　ネットワーク接続の有無別のSLOC規模とSLOC生産性：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模とSLOC生産性の関係をネットワーク接続の有無別に示す。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-6_ネットワーク接続の有無が回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- SLOC生産性 ＞ 0

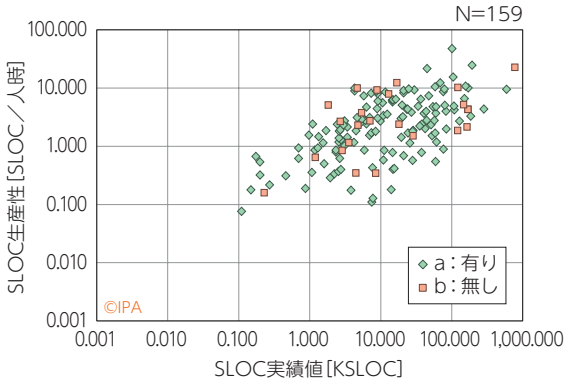
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：SLOC生産性
  （SLOC／実績工数（開発5工程））（導出指標）

図表10.7-9　ネットワーク接続の有無別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



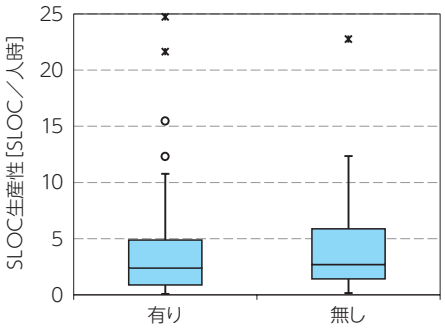
※表示されていないものが4点ある

図表10.7-10　ネットワーク接続の有無別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



※ネットワーク接続の有無別のSLOC規模とSLOC生産性について、対数化した場合の相関係数は次のようになる。
a：有り　R=0.60（P＜0.05）
b：無し　R=0.56（P＜0.05）

図表10.7-11　ネットワーク接続の有無別SLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.7-12　ネットワーク接続の有無別SLOC生産性の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

［SLOC／人時］

ネットワーク接続の有無	N	P25	中央	P75
a：有り	135	0.88	2.39	4.89
b：無し	24	1.42	2.70	5.87

●ネットワーク接続の有無別の生産性に違いは見られない。

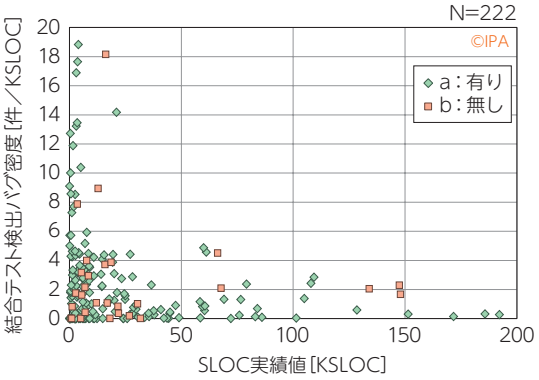
10.7.5 ネットワーク接続の有無別のSLOC規模と結合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模と結合テスト検出バグ密度の関係について、ネットワーク接続の有無別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-6_ネットワーク接続の有無が回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- 10-3-1_結合テスト検出バグ現象数 ≧ 0

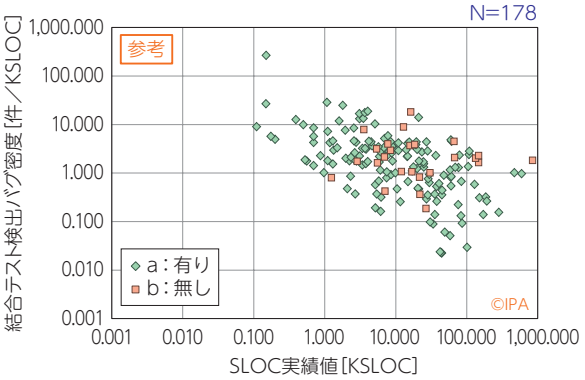
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：結合テスト検出バグ密度（SLOCあたりの検出バグ数）（導出指標）

図表10.7-13 ネットワーク接続の有無別のSLOC規模と結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



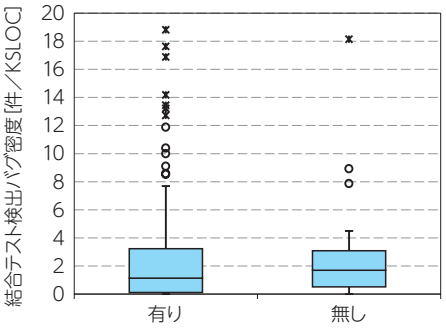
※表示されていないものが9点ある

図表10.7-14 ネットワーク接続の有無別のSLOC規模と結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件/KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

図表10.7-15 ネットワーク接続の有無別結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.7-16 ネットワーク接続の有無別結合テスト検出バグ密度の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

[件／KSLOC]

ネットワーク接続の有無	N	P25	中央	P75
a：有り	192	0.124	1.131	3.237
b：無し	30	0.517	1.698	3.096

●ネットワーク接続【有り】と【無し】双方の「結合テスト検出バグ密度」は、SLOC規模が同じ範囲内で比較すると、【無し】の方が幾分高い傾向が見られる。

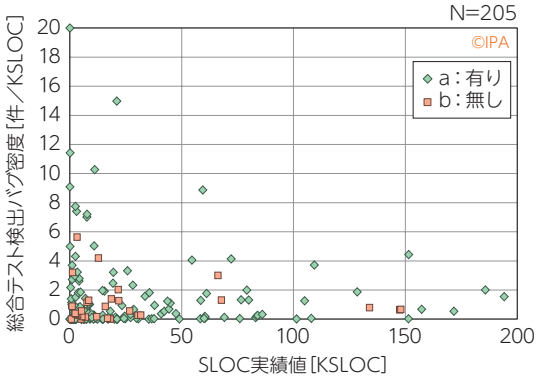
10.7.6 ネットワーク接続の有無別のSLOC規模と総合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模と総合テスト検出バグ密度の関係について、ネットワーク接続の有無別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-6_ネットワーク接続の有無が回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- 10-4-1_総合テスト検出バグ現象数 ≧ 0

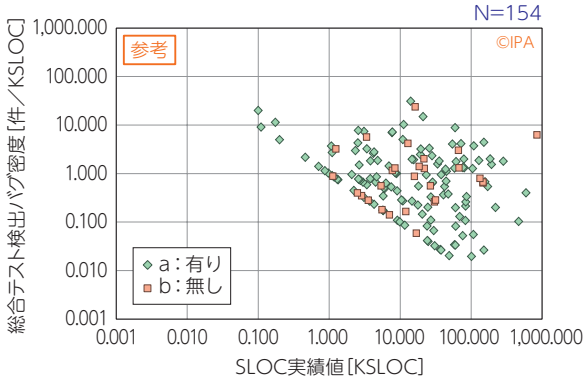
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：総合テスト検出バグ密度（SLOCあたりの検出バグ数）（導出指標）

図表10.7-17 ネットワーク接続の有無別のSLOC規模と総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



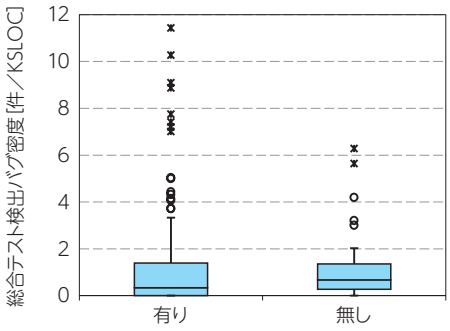
※表示されていないものが7点ある

図表10.7-18 ネットワーク接続の有無別のSLOC規模と総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件/KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

図表10.7-19 ネットワーク接続の有無別総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.7-20 ネットワーク接続の有無別総合テスト検出バグ密度の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

[件／KSLOC]

ネットワーク接続の有無	N	P25	中央	P75
a：有り	174	0.000	0.341	1.389
b：無し	31	0.272	0.669	1.354

●ネットワーク接続【有り】と【無し】双方の「総合テスト検出バグ密度」は、SLOC規模が同じ範囲内で比較すると、【無し】の方が高い傾向が見られる。

10.8 稼動（非停止、オンデマンド）

この節では、改良（派生）開発で開発5工程（アーキテクチャ設計～総合テスト）の作業の行われたプロジェクトを対象に、稼動（非停止、オンデマンド）別による分析を示す。

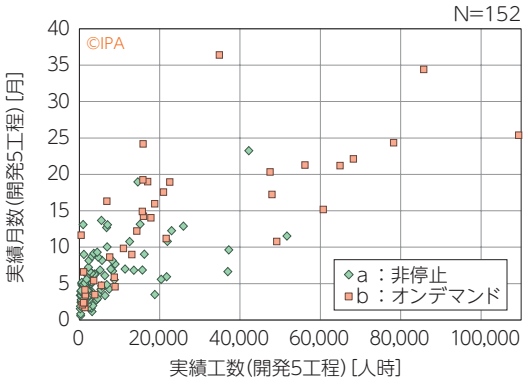
10.8.1 稼動（非停止、オンデマンド）別の工数と工期：改良（派生）開発、開発5工程、開発言語C/C++

ここでは、実績工数とその工期（月数）の関係を稼動（非停止、オンデマンド）別に示す。

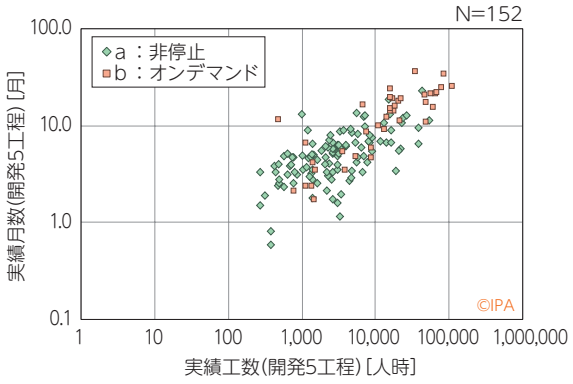
■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-7_稼動（非停止、オンデマンド）が回答されているもの
- 実績工数（開発5工程）＞0
- 実績月数（開発5工程）＞0

■対象データ
- X軸：実績工数（開発5工程）（導出指標）
- Y軸：実績月数（開発5工程）（導出指標）

図表10.8-1 稼動（非停止、オンデマンド）別の工数と工期（改良（派生）開発、開発5工程、開発言語C/C++）



図表10.8-2 稼動（非停止、オンデマンド）別の工数と工期（改良（派生）開発、開発5工程、開発言語C/C++）対数表示



※稼動（非停止、オンデマンド）の工数と工期について、対数化した場合の相関係数は次のようになる。
a：非停止　R=0.62（P<0.05）
b：オンデマンド　R=0.82（P<0.05）

図表10.8-3 稼動（非停止、オンデマンド）別の実績月数の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++）

［月］

稼動（非停止、オンデマンド）	N	P25	中央	P75
a：非停止	113	3.40	4.97	6.87
b：オンデマンド	39	5.63	14.03	19.12

●稼働【オンデマンド】の場合、【非停止】の場合に比べて、工数の増加に伴い工期が長くなる傾向が強く出ている。

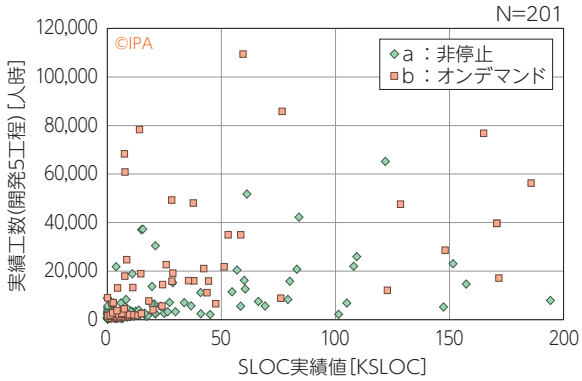
10.8.2 稼動（非停止、オンデマンド）別のSLOC規模と工数：改良（派生）開発、開発5工程、開発言語C/C++

ここでは、SLOC規模と工数の関係を稼動（非停止、オンデマンド）別に示す。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-7_稼動（非停止、オンデマンド）が回答されているもの
- 実効SLOC実績値 ＞ 0
- 実績工数（開発5工程）＞ 0

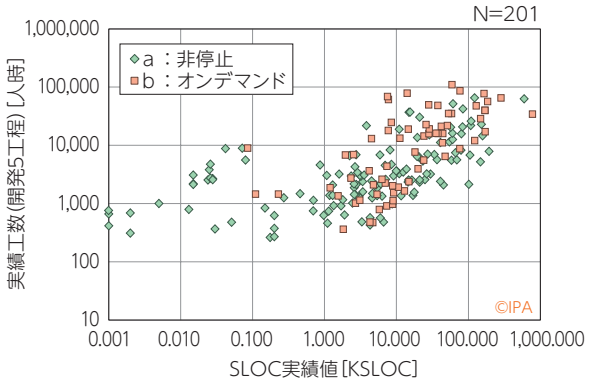
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：実績工数（開発5工程）（導出指標）

図表10.8-4 稼動（非停止、オンデマンド）別のSLOC規模と工数（改良（派生）開発、開発5工程、開発言語C/C++）



※表示されていないものが3点ある

図表10.8-5 稼動（非停止、オンデマンド）別のSLOC規模と工数（改良（派生）開発、開発5工程、開発言語C/C++）対数表示



※稼動（非停止、オンデマンド）のSLOC規模と工数について、対数化した場合の相関係数は次のようになる。
a：非停止　R=0.25（P＜0.05）
b：オンデマンド　R=0.56（P＜0.05）

図表10.8-6 稼動（非停止、オンデマンド）別の実績工数の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++）

［人時］

稼動（非停止、オンデマンド）	N	P25	中央	P75
a：非停止	134	1,337	2,587	6,561
b：オンデマンド	67	1,951	11,013	26,565

●工数の増加はSLOC規模に依存するという一般的な傾向は、【非停止】【オンデマンド】いずれの場合も、顕著ではない。

10.8.3 稼動（非停止、オンデマンド）別の工程別工数：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、工程別の実績工数の比率を稼動（非停止、オンデマンド）別に示す。

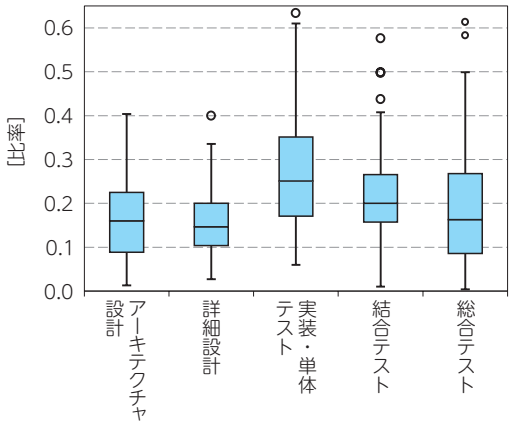
■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-7_稼動（非停止、オンデマンド）が回答されているもの
- 工程別の実績工数にすべて記入があり、各値が0より大きい
- 実効SLOC実績値 ≧ 0.1KSLOC

■対象データ
- 9-4-30_社内実績工数_アーキテクチャ設計、9-4-31_社内実績工数_詳細設計、9-4-32_社内実績工数_実装・単体テスト、9-4-33_社内実績工数_結合テスト、9-4-34_社内実績工数_総合テスト
- 9-8-2_外部委託工数_アーキテクチャ設計、9-8-3_外部委託工数_詳細設計、9-8-4_外部委託工数_実装・単体テスト、9-8-5_外部委託工数_結合テスト、9-8-6_外部委託工数_総合テスト

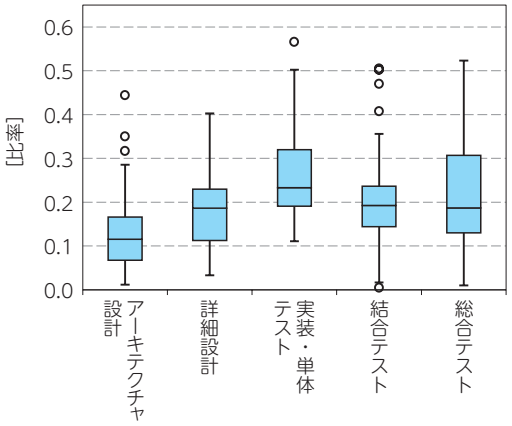
※各工程の実績工数は、社内、外部委託の実績工数合計の人時換算値を使用。

図表10.8-7　稼動（非停止、オンデマンド）別の工程別の実績工数の比率（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図

稼動【非停止】



稼動【オンデマンド】



図表10.8-8　稼動（非停止、オンデマンド）別の工程別の実績工数の比率の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

稼動【非停止】　[比率]

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	113	0.01	0.09	0.16	0.22	0.40	0.16	0.09
詳細設計	113	0.03	0.10	0.15	0.20	0.40	0.16	0.07
実装・単体テスト	113	0.06	0.17	0.25	0.35	0.79	0.28	0.14
結合テスト	113	0.01	0.16	0.20	0.27	0.58	0.21	0.09
総合テスト	113	0.00	0.09	0.16	0.27	0.61	0.19	0.13

稼動【オンデマンド】　[比率]

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	66	0.01	0.07	0.12	0.17	0.44	0.13	0.08
詳細設計	66	0.03	0.11	0.19	0.23	0.67	0.19	0.11
実装・単体テスト	66	0.11	0.19	0.23	0.32	0.57	0.26	0.09
結合テスト	66	0.01	0.14	0.19	0.24	0.51	0.21	0.11
総合テスト	66	0.01	0.13	0.19	0.31	0.52	0.22	0.12

●稼動【非停止】の方が、アーキテクチャ設計にかける工数比率が高い。一方、詳細設計にかける工数比率は、稼動【オンデマンド】の方が高くなっている。

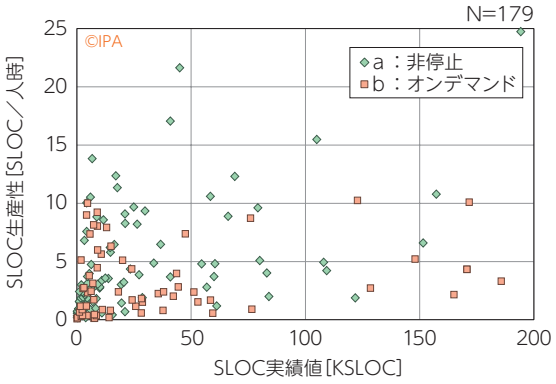
10.8.4 稼動（非停止、オンデマンド）別のSLOC規模とSLOC生産性：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模とSLOC生産性の関係を稼動（非停止、オンデマンド）別に示す。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-7_稼動（非停止、オンデマンド）が回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- SLOC生産性 ＞ 0

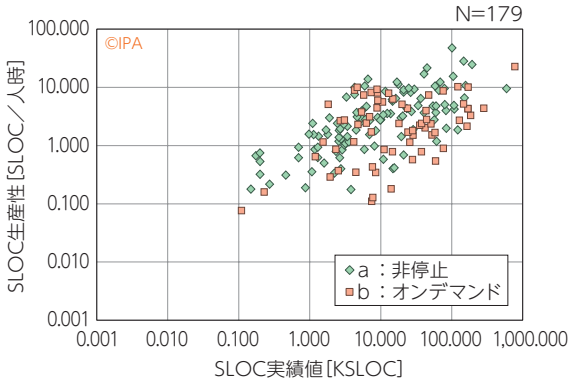
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：SLOC生産性
  （SLOC／実績工数（開発5工程））（導出指標）

図表10.8-9 稼動（非停止、オンデマンド）別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



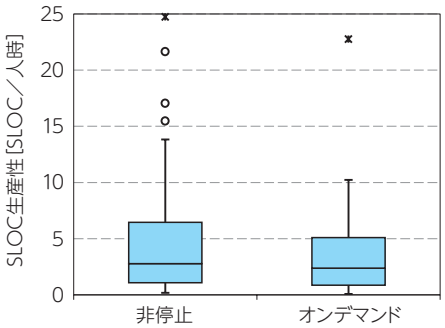
※表示されていないものが5点ある

図表10.8-10 稼動（非停止、オンデマンド）別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



※稼動（非停止、オンデマンド）別のSLOC規模とSLOC生産性について、対数化した場合の相関係数は次のようになる。
a：非停止　　R=0.70（P＜0.05）
b：オンデマンド　R=0.46（P＜0.05）

図表10.8-11 稼動（非停止、オンデマンド）別SLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.8-12 稼動（非停止、オンデマンド）別SLOC生産性の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

［SLOC／人時］

稼動（非停止、オンデマンド）	N	P25	中央	P75
a：非停止	113	1.09	2.78	6.46
b：オンデマンド	66	0.87	2.39	5.10

●稼動（非停止、オンデマンド）別のSLOC生産性について、傾向の違いが見られない。

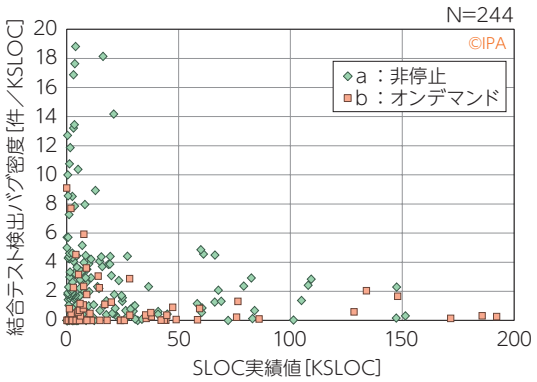
10.8.5 稼動（非停止、オンデマンド）別のSLOC規模と結合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模と結合テスト検出バグ密度の関係について、稼動（非停止、オンデマンド）別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-7_稼動（非停止、オンデマンド）が回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- 10-3-1_結合テスト検出バグ現象数 ≧ 0

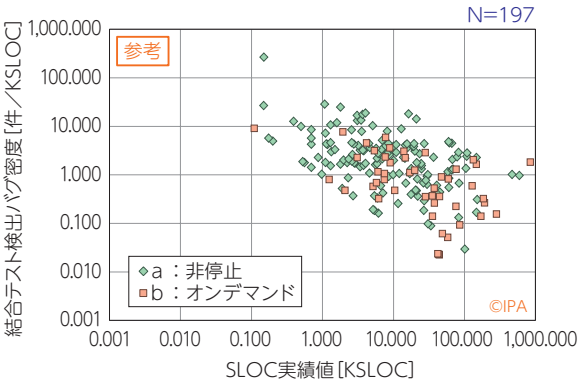
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：結合テスト検出バグ密度（SLOCあたりの検出バグ数）（導出指標）

図表10.8-13 稼動（非停止、オンデマンド）別のSLOC規模と結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



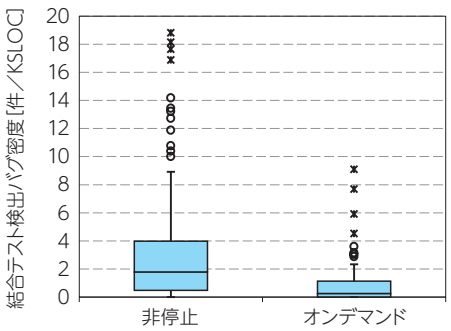
※表示されていないものが9点ある

図表10.8-14 稼動（非停止、オンデマンド）別のSLOC規模と結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件/KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

図表10.8-15 稼動（非停止、オンデマンド）別結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.8-16 稼動（非停止、オンデマンド）別結合テスト検出バグ密度の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

[件／KSLOC]

稼動（非停止、オンデマンド）	N	P25	中央	P75
a：非停止	173	0.474	1.786	3.978
b：オンデマンド	71	0.000	0.265	1.131

●結合テスト検出バグ密度は、稼動【非停止】の方が高い傾向にある。

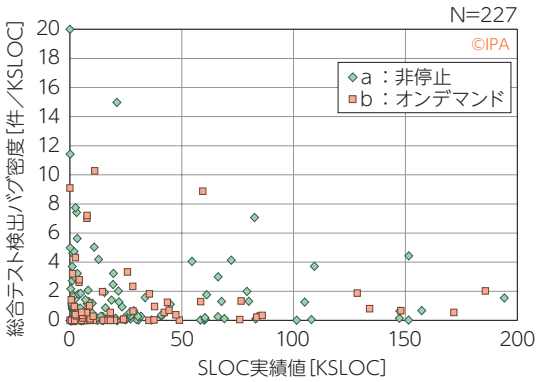
10.8.6 稼動（非停止、オンデマンド）別のSLOC規模と総合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模と総合テスト検出バグ密度の関係について、稼動（非停止、オンデマンド）別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-7_稼動（非停止、オンデマンド）が回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- 10-4-1_総合テスト検出バグ現象数 ≧ 0

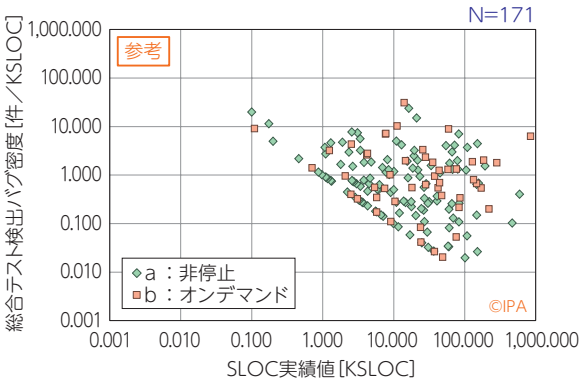
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：総合テスト検出バグ密度（SLOCあたりの検出バグ数）（導出指標）

図表10.8-17 稼動（非停止、オンデマンド）別のSLOC規模と総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



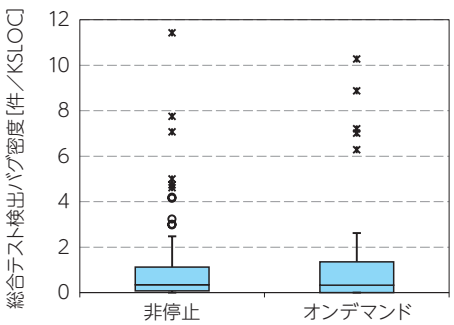
※表示されていないものが7点ある

図表10.8-18 稼動（非停止、オンデマンド）別のSLOC規模と総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件/KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

図表10.8-19 稼動（非停止、オンデマンド）別総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.8-20 稼動（非停止、オンデマンド）別総合テスト検出バグ密度の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

[件／KSLOC]

稼動（非停止、オンデマンド）	N	P25	中央	P75
a：非停止	156	0.034	0.357	1.316
b：オンデマンド	71	0.000	0.379	1.601

●稼動(非停止、オンデマンド)別の総合テスト検出バグ密度について、違いは見られない。

10.9 オンライン保守の可否

この節では、改良(派生)開発で開発5工程(アーキテクチャ設計～総合テスト)の作業の行われたプロジェクトを対象に、オンライン保守の可否別による分析を示す。

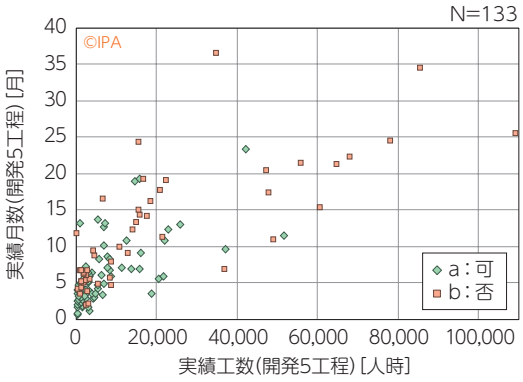
10.9.1 オンライン保守の可否別の工数と工期：改良(派生)開発、開発5工程、開発言語C/C++

ここでは、実績工数とその工期(月数)の関係をオンライン保守の可否別に示す。

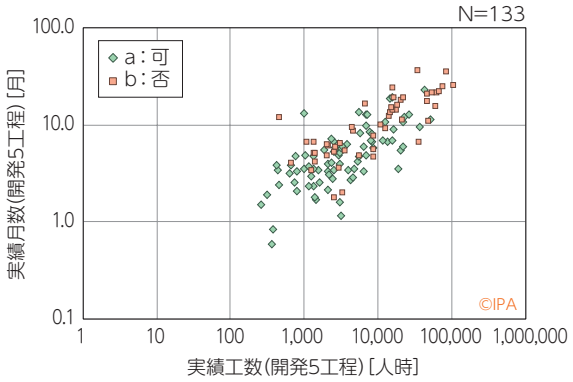
■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良(派生)開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-8_オンライン保守の可否が回答されているもの
- 実績工数(開発5工程) ＞ 0
- 実績月数(開発5工程) ＞ 0

■対象データ
- X軸：実績工数(開発5工程)(導出指標)
- Y軸：実績月数(開発5工程)(導出指標)

図表10.9-1 オンライン保守の可否別の工数と工期(改良(派生)開発、開発5工程、開発言語C/C++)



図表10.9-2 オンライン保守の可否別の工数と工期(改良(派生)開発、開発5工程、開発言語C/C++)対数表示



※オンライン保守の可否別の工数と工期について、対数化した場合の相関係数は次のようになる。
a：可　R=0.71 (P<0.05)
b：否　R=0.77 (P<0.05)

図表10.9-3 オンライン保守の可否別の実績月数の基本統計量(改良(派生)開発、開発5工程、開発言語C/C++)

[月]

オンライン保守の可否	N	P25	中央	P75
a：可	83	3.25	4.77	6.93
b：否	50	5.43	9.57	17.00

●オンライン保守【可】【非】いずれの場合も、工数の増加に伴い工期が長くなる傾向は、やや強く出ている。

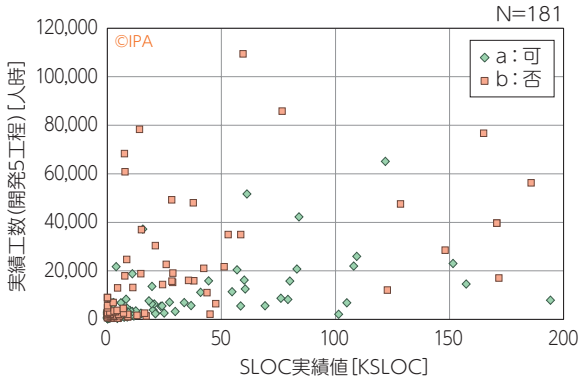
10.9.2 オンライン保守の可否別のSLOC規模と工数：改良（派生）開発、開発5工程、開発言語C/C++

ここでは、SLOC規模と工数の関係をオンライン保守の可否別に示す。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-8_オンライン保守の可否が回答されているもの
- 実効SLOC実績値 ＞ 0
- 実績工数（開発5工程）＞ 0

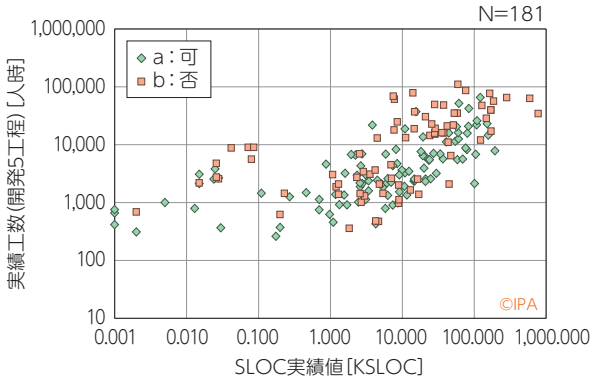
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：実績工数（開発5工程）（導出指標）

図表10.9-4 オンライン保守の可否別のSLOC規模と工数（改良（派生）開発、開発5工程、開発言語C/C++）



※表示されていないものが3点ある

図表10.9-5 オンライン保守の可否別のSLOC規模と工数（改良（派生）開発、開発5工程、開発言語C/C++）対数表示



※オンライン保守の可否別のSLOC規模と工数について、対数化した場合の相関係数は次のようになる。
a：可　R=0.66（P<0.05）
b：否　R=0.59（P<0.05）

図表10.9-6 オンライン保守の可否別の実績工数の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++）

[人時]

オンライン保守の可否	N	P25	中央	P75
a：可	106	1,411	2,952	6,877
b：否	75	2,080	8,940	29,437

●工数の増加はSLOC規模に依存するという一般的な傾向は、オンライン保守【可】【非】いずれの場合も、顕著ではない。

10.9.3 オンライン保守の可否別の工程別工数：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

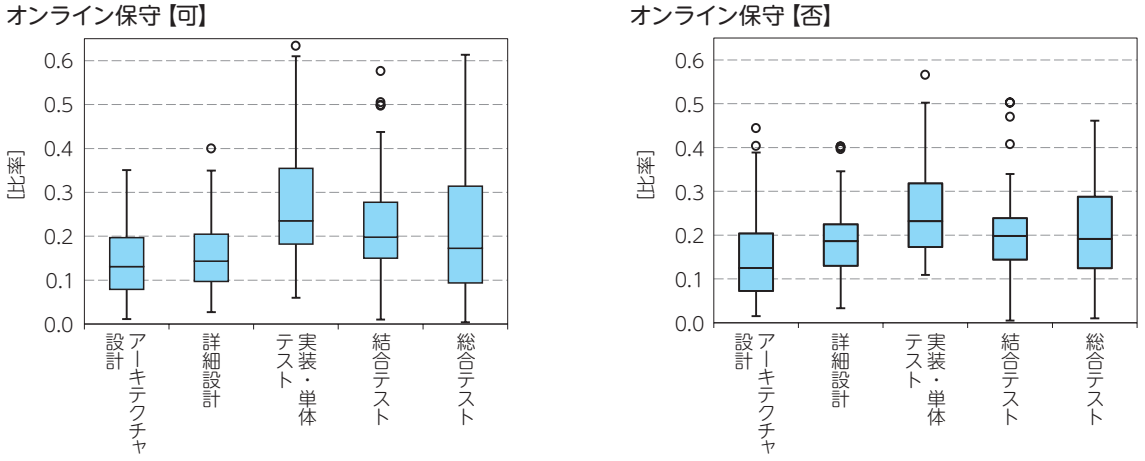
ここでは、工程別の実績工数の比率をオンライン保守の可否別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-8_オンライン保守の可否が回答されているもの
- 工程別の実績工数にすべて記入があり、各値が0より大きい
- 実効SLOC実績値 ≧ 0.1KSLOC

■対象データ
- 9-4-30_社内実績工数_アーキテクチャ設計、
  9-4-31_社内実績工数_詳細設計、
  9-4-32_社内実績工数_実装・単体テスト、
  9-4-33_社内実績工数_結合テスト、
  9-4-34_社内実績工数_総合テスト
- 9-8-2_外部委託工数_アーキテクチャ設計、
  9-8-3_外部委託工数_詳細設計、
  9-8-4_外部委託工数_実装・単体テスト、
  9-8-5_外部委託工数_結合テスト、
  9-8-6_外部委託工数_総合テスト

※各工程の実績工数は、社内、外部委託の実績工数合計の人時換算値を使用。

図表10.9-7　オンライン保守の可否別の工程別の実績工数の比率（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.9-8　オンライン保守の可否別の工程別の実績工数の比率の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

オンライン保守【可】　[比率]

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	94	0.01	0.08	0.13	0.20	0.35	0.14	0.08
詳細設計	94	0.03	0.10	0.14	0.20	0.40	0.15	0.07
実装・単体テスト	94	0.06	0.18	0.24	0.35	0.79	0.28	0.14
結合テスト	94	0.01	0.15	0.20	0.28	0.58	0.22	0.11
総合テスト	94	0.00	0.09	0.17	0.31	0.61	0.21	0.15

オンライン保守【否】　[比率]

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	66	0.02	0.07	0.12	0.20	0.44	0.15	0.10
詳細設計	66	0.03	0.13	0.19	0.22	0.67	0.19	0.11
実装・単体テスト	66	0.11	0.17	0.23	0.32	0.57	0.25	0.10
結合テスト	66	0.01	0.14	0.20	0.24	0.50	0.20	0.10
総合テスト	66	0.01	0.12	0.19	0.29	0.46	0.21	0.11

●オンライン保守の可否別の工程別工数の配分比率について、バラつきに差はあるものの大きな違いは見られない。

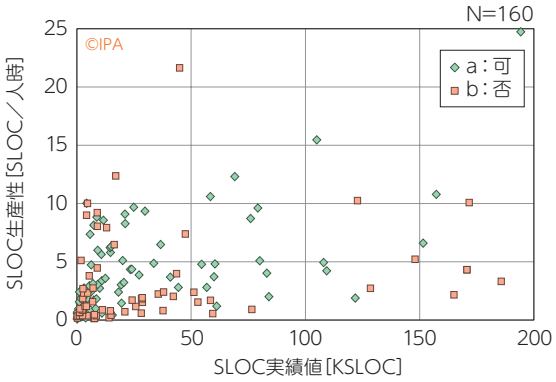
10.9.4 オンライン保守の可否別のSLOC規模とSLOC生産性：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模とSLOC生産性の関係をオンライン保守の可否別に示す。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-8_オンライン保守の可否が回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- SLOC生産性 ＞ 0

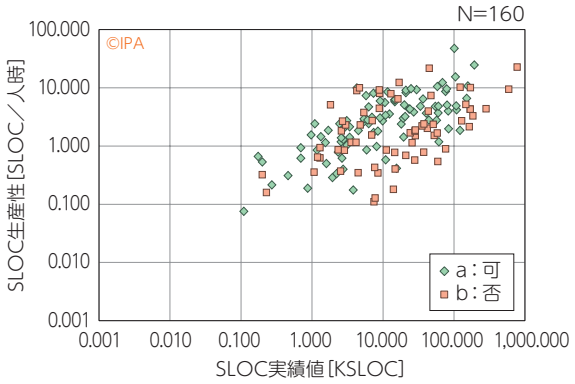
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：SLOC生産性
  （SLOC ／実績工数（開発5工程））（導出指標）

図表10.9-9 オンライン保守の可否別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



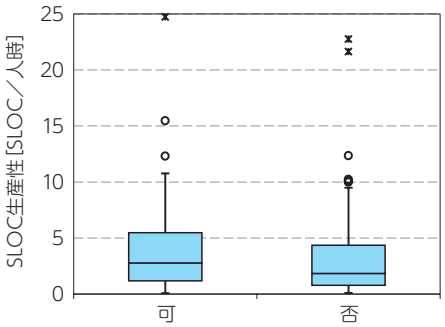
※表示されていないものが4点ある

図表10.9-10 オンライン保守の可否別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



※オンライン保守の可否別のSLOC規模とSLOC生産性について、対数化した場合の相関係数は次のようになる。
a：可　R=0.73（P＜0.05）
b：否　R=0.49（P＜0.05）

図表10.9-11 オンライン保守の可否別SLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.9-12 オンライン保守の可否別SLOC生産性の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

［SLOC ／人時］

オンライン保守の可否	N	P25	中央	P75
a：可	94	1.18	2.79	5.48
b：否	66	0.78	1.85	4.36

● オンライン保守の可否別の生産性について、SLOC規模が10KSLOC以上の範囲では、オンライン保守【可】の方が高い傾向にある。

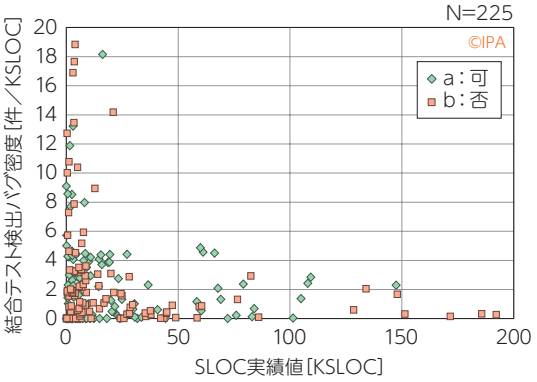
10.9.5 オンライン保守の可否別のSLOC規模と結合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模と結合テスト検出バグ密度の関係について、オンライン保守の可否別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-8_オンライン保守の可否が回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- 10-3-1_結合テスト検出バグ現象数 ≧ 0

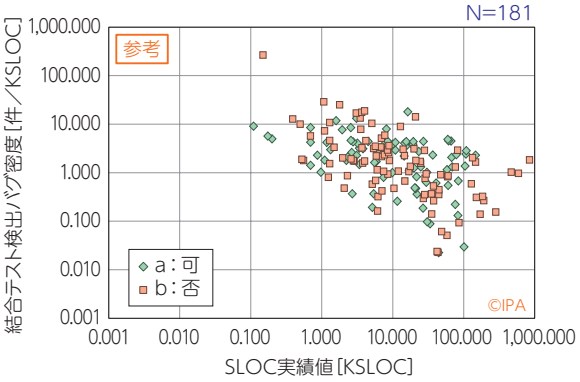
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：結合テスト検出バグ密度（SLOCあたりの検出バグ数）（導出指標）

図表10.9-13 オンライン保守の可否別のSLOC規模と結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



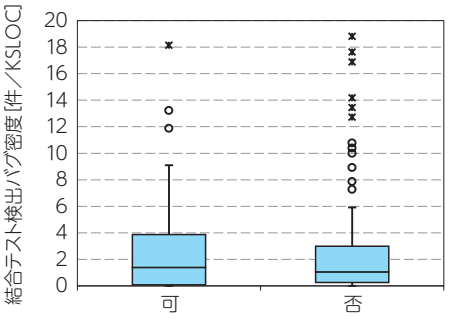
※表示されていないものが8点ある

図表10.9-14 オンライン保守の可否別のSLOC規模と結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件/KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

図表10.9-15 オンライン保守の可否別結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.9-16 オンライン保守の可否別結合テスト検出バグ密度の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

[件／KSLOC]

オンライン保守の可否	N	P25	中央	P75
a：可	110	0.091	1.400	3.871
b：否	115	0.264	1.064	2.993

●オンライン保守の可否別の結合テスト検出バグ密度について、傾向の違いは見られない。

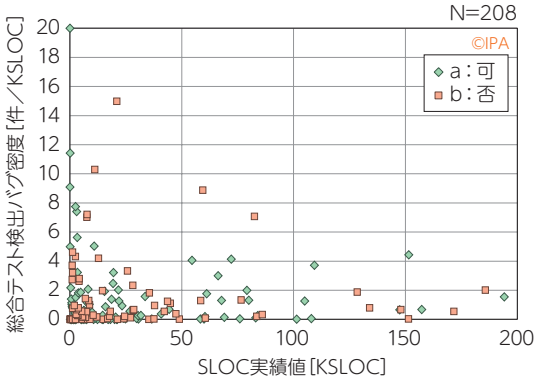
10.9.6 オンライン保守の可否別のSLOC規模と総合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模と総合テスト検出バグ密度の関係について、オンライン保守の可否別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 3-1-8_オンライン保守の可否が回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- 10-4-1_総合テスト検出バグ現象数 ≧ 0

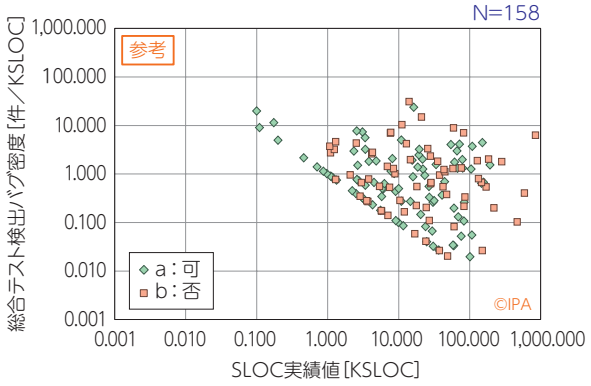
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：総合テスト検出バグ密度（SLOCあたりの検出バグ数）（導出指標）

図表10.9-17 オンライン保守の可否別のSLOC規模と総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



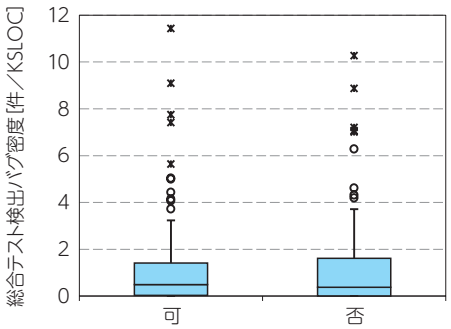
※表示されていないものが7点ある

図表10.9-18 オンライン保守の可否別のSLOC規模と総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件/KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

図表10.9-19 オンライン保守の可否別総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.9-20 オンライン保守の可否別総合テスト検出バグ密度の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

［件／KSLOC］

オンライン保守の可否	N	P25	中央	P75
a：可	117	0.033	0.493	1.408
b：否	91	0.000	0.379	1.614

●オンライン保守の可否別の総合テスト検出バグ密度について、傾向の違いは見られない。

10.10 障害リスク (Type)

この節では、改良(派生)開発で開発5工程(アーキテクチャ設計～総合テスト)の作業の行われたプロジェクトを対象に、障害リスク(Type)別による分析を示す。

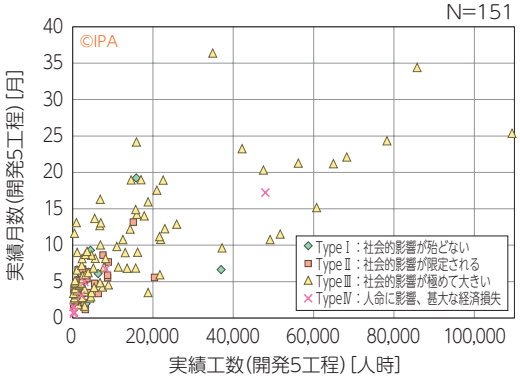
10.10.1 障害リスク(Type)別の工数と工期：改良(派生)開発、開発5工程、開発言語C/C++

ここでは、実績工数とその工期(月数)の関係を障害リスク(Type)別に示す。

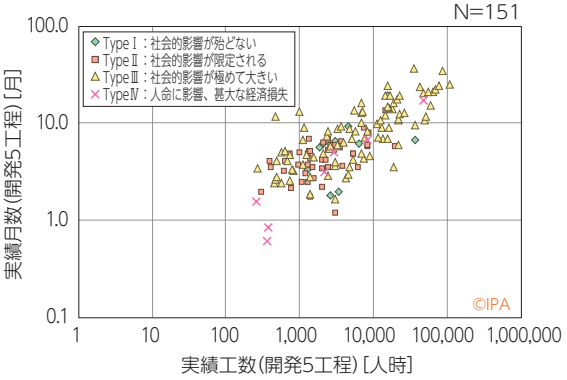
■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良(派生)開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 2-2_障害リスクが回答されているもの
- 実績工数(開発5工程) ＞ 0
- 実績月数(開発5工程) ＞ 0

■対象データ
- X軸：実績工数(開発5工程)(導出指標)
- Y軸：実績月数(開発5工程)(導出指標)

図表10.10-1 障害リスク(Type)別の工数と工期(改良(派生)開発、開発5工程、開発言語C/C++)



図表10.10-2 障害リスク(Type)別の工数と工期(改良(派生)開発、開発5工程、開発言語C/C++)対数表示



※障害リスク(Type)別の工数と工期について、対数化した場合の相関係数は次のようになる。

TypeⅠ：社会的影響が殆どない	R=0.51 (P≧0.05)
TypeⅡ：社会的影響が限定される	R=0.51 (P<0.05)
TypeⅢ：社会的影響が極めて大きい	R=0.75 (P<0.05)
TypeⅣ：人命に影響、甚大な経済損失	R=0.95 (P<0.05)

図表10.10-3 障害リスク(Type)別の実績月数の基本統計量(改良(派生)開発、開発5工程、開発言語C/C++)

[月]

障害リスク	N	P25	中央	P75
TypeⅠ：社会的影響が殆どない	10	3.87	6.12	6.59
TypeⅡ：社会的影響が限定される	46	3.38	4.42	5.56
TypeⅢ：社会的影響が極めて大きい	88	4.25	8.17	13.08
TypeⅣ：人命に影響、甚大な経済損失	7	1.18	3.10	5.85

●障害リスクが高くなると、工数の増加に伴い工期が長くなる傾向が強く出ている。

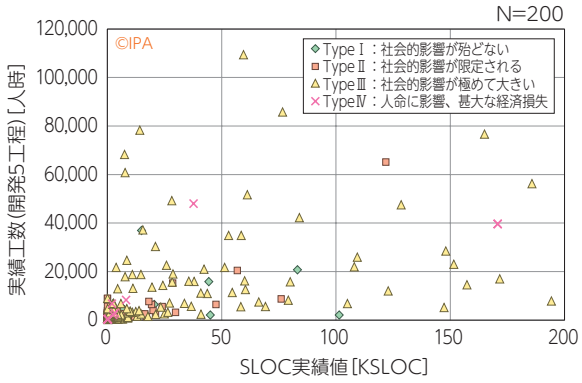
10.10.2 障害リスク（Type）別のSLOC規模と工数：改良（派生）開発、開発5工程、開発言語C/C++

ここでは、SLOC規模と工数の関係を障害リスク（Type）別に示す。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 2-2_障害リスクが回答されているもの
- 実効SLOC実績値 ＞ 0
- 実績工数（開発5工程）＞ 0

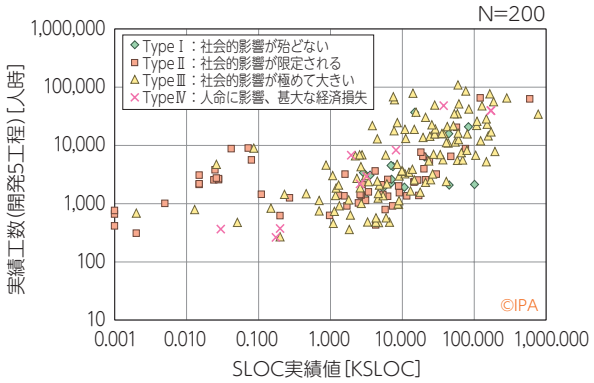
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：実績工数（開発5工程）（導出指標）

図表10.10-4 障害リスク（Type）別のSLOC規模と工数（改良（派生）開発、開発5工程、開発言語C/C++）



※表示されていないものが3点ある

図表10.10-5 障害リスク（Type）別のSLOC規模と工数（改良（派生）開発、開発5工程、開発言語C/C++）対数表示



※障害リスク（Type）別のSLOC規模と工数について、対数化した場合の相関係数は次のようになる。

Type Ⅰ：社会的影響が殆どない	R=0.42（P≧0.05）
Type Ⅱ：社会的影響が限定される	R=0.47（P<0.05）
Type Ⅲ：社会的影響が極めて大きい	R=0.64（P<0.05）
Type Ⅳ：人命に影響、甚大な経済損失	R=0.95（P<0.05）

図表10.10-6 障害リスク（Type）別の実績工数の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++）

［人時］

障害リスク	N	P25	中央	P75
Type Ⅰ：社会的影響が殆どない	15	2,104	3,082	5,438
Type Ⅱ：社会的影響が限定される	57	1,331	2,123	3,626
Type Ⅲ：社会的影響が極めて大きい	118	1,583	5,565	17,661
Type Ⅳ：人命に影響、甚大な経済損失	10	824	4,855	31,788

● 工数の増加はSLOC規模に依存するという一般的な傾向を障害リスク（Type）別で見ると、【Type Ⅳ】／【Type Ⅲ】は、【Type Ⅱ】／【Type Ⅰ】よりも比較的その傾向が強く出ている。

10.10.3 障害リスク（Type）別の工程別工数：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、工程別の実績工数の比率を障害リスク（Type）別に示す。

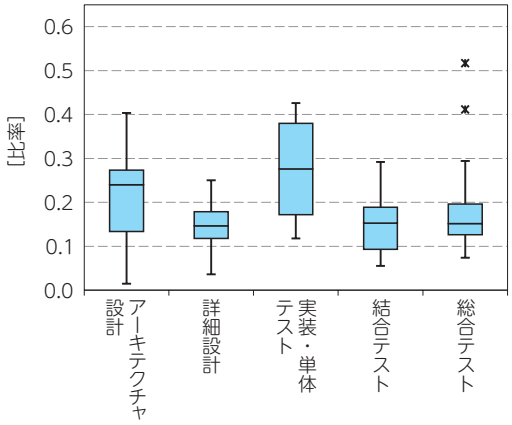
■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 2-2_障害リスクが回答されているもの
- 工程別の実績工数にすべて記入があり、各値が0より大きい
- 実効SLOC実績値 ≧ 0.1KSLOC

■対象データ
- 9-4-30_社内実績工数_アーキテクチャ設計、
  9-4-31_社内実績工数_詳細設計、
  9-4-32_社内実績工数_実装・単体テスト、
  9-4-33_社内実績工数_結合テスト、
  9-4-34_社内実績工数_総合テスト
- 9-8-2_外部委託工数_アーキテクチャ設計、
  9-8-3_外部委託工数_詳細設計、
  9-8-4_外部委託工数_実装・単体テスト、
  9-8-5_外部委託工数_結合テスト、
  9-8-6_外部委託工数_総合テスト

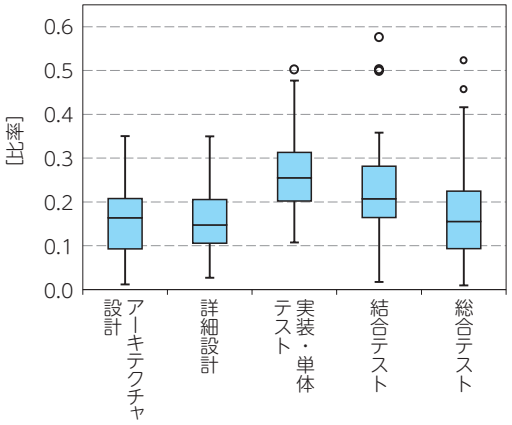
※各工程の実績工数は、社内、外部委託の実績工数合計の人時換算値を使用。

図表10.10 7　障害リスク(Type)別の工程別の実績工数の比率（改良(派生)開発、開発５工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図

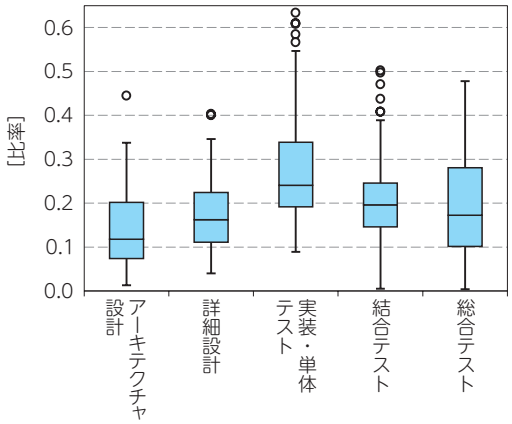
【TypeⅠ：社会的影響が殆どない】



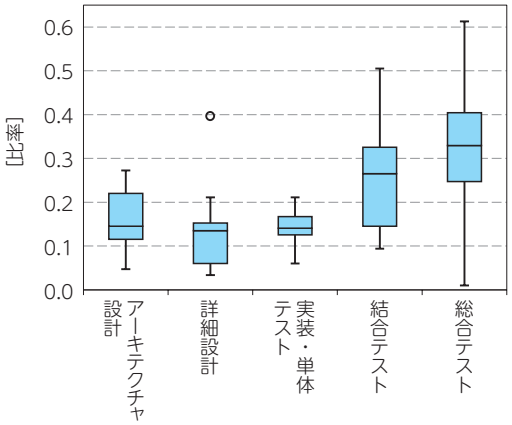
【TypeⅡ：社会的影響が限定される】



【TypeⅢ：社会的影響が極めて大きい】



【TypeⅣ：人命に影響、甚大な経済損失】



図表10.10-8 障害リスク(Type)別の工程別の実績工数の比率の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)

【TypeⅠ:社会的影響が殆どない】 [比率]

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	15	0.02	0.13	0.24	0.27	0.40	0.21	0.11
詳細設計	15	0.04	0.12	0.15	0.18	0.25	0.14	0.06
実装・単体テスト	15	0.12	0.17	0.28	0.38	0.79	0.30	0.17
結合テスト	15	0.06	0.09	0.15	0.19	0.29	0.15	0.07
総合テスト	15	0.07	0.13	0.15	0.20	0.52	0.20	0.12

【TypeⅡ:社会的影響が限定される】 [比率]

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	41	0.01	0.09	0.16	0.21	0.35	0.16	0.08
詳細設計	41	0.03	0.11	0.15	0.21	0.35	0.16	0.07
実装・単体テスト	41	0.11	0.20	0.26	0.31	0.50	0.26	0.10
結合テスト	41	0.02	0.16	0.21	0.28	0.58	0.24	0.11
総合テスト	41	0.01	0.09	0.16	0.22	0.52	0.18	0.12

【TypeⅢ:社会的影響が極めて大きい】 [比率]

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	113	0.01	0.07	0.12	0.20	0.44	0.14	0.09
詳細設計	113	0.04	0.11	0.16	0.22	0.67	0.18	0.09
実装・単体テスト	113	0.09	0.19	0.24	0.34	0.63	0.28	0.12
結合テスト	113	0.01	0.15	0.20	0.25	0.50	0.21	0.09
総合テスト	113	0.00	0.10	0.17	0.28	0.48	0.20	0.12

【TypeⅣ:人命に影響、甚大な経済損失】 [比率]

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	9	0.05	0.12	0.14	0.22	0.27	0.15	0.07
詳細設計	9	0.03	0.06	0.13	0.15	0.40	0.14	0.11
実装・単体テスト	9	0.06	0.13	0.14	0.17	0.21	0.14	0.04
結合テスト	9	0.09	0.15	0.27	0.33	0.51	0.25	0.12
総合テスト	9	0.01	0.25	0.33	0.40	0.61	0.31	0.17

● 標本数がそろっていないため参考扱いではあるが、高品質が求められる障害リスク【TypeⅣ】の場合には、テスト(結合テスト及び総合テスト)にかける工数比率が高い。反対に品質要求が通常レベルの障害リスク【TypeⅠ】の場合は、実装・単体テストにかける工数比率が高い。

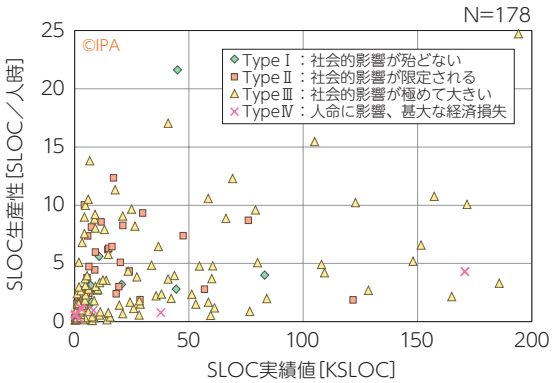
10.10.4 障害リスク（Type）別のSLOC規模とSLOC生産性：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模とSLOC生産性の関係を障害リスク（Type）別に示す。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 2-2_障害リスクが回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- SLOC生産性 ＞ 0

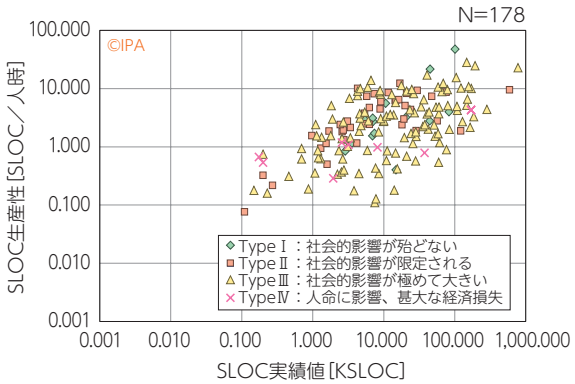
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：SLOC生産性
  （SLOC ／実績工数（開発5工程））（導出指標）

図表10.10-9 障害リスク（Type）別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



※表示されていないものが5点ある

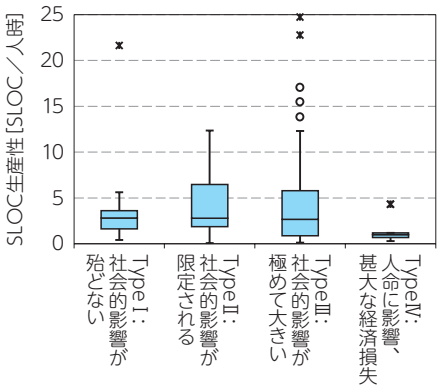
図表10.10-10 障害リスク（Type）別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



※障害リスク【TypeⅠ：社会的影響が殆どない】、【TypeⅡ：社会的影響が限定される】、【TypeⅢ：社会的影響が極めて大きい】、のSLOC規模とSLOC生産性について、対数化した場合の相関係数は次のようになる。

TypeⅠ：社会的影響が殆どない	R=0.65（P＜0.05）
TypeⅡ：社会的影響が限定される	R=0.75（P＜0.05）
TypeⅢ：社会的影響が極めて大きい	R=0.55（P＜0.05）

図表10.10-11 障害リスク（Type）別SLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表10.10-12 障害リスク（Type）別SLOC生産性の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

[SLOC ／人時]

障害リスク	N	P25	中央	P75
TypeⅠ：社会的影響が殆どない	15	1.63	2.80	3.61
TypeⅡ：社会的影響が限定される	41	1.86	2.78	6.46
TypeⅢ：社会的影響が極めて大きい	113	0.86	2.67	5.79
TypeⅣ：人命に影響、甚大な経済損失	9	0.67	0.98	1.19

●標本数がそろっていないため参考扱いではあるが、障害リスク(Type)が高くなるに従い、生産性が低くなる傾向が見られる。

11章 品質評価別の分析

11.1 位置付け 178

11.2 品質実績の評価 179

11.2.1 品質実績の評価別の工数と工期：
改良（派生）開発、開発5工程、開発言語C/C++

11.2.2 品質実績の評価別のSLOC規模と工数：
改良（派生）開発、開発5工程、開発言語C/C++

11.2.3 品質実績の評価別の工程別工数：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

11.2.4 品質実績の評価別のSLOC規模とSLOC生産性：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

11.2.5 品質実績の評価別のSLOC規模と結合テスト検出バグ密度：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

11.2.6 品質実績の評価別のSLOC規模と総合テスト検出バグ密度：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

11.2.7 品質実績の評価別のSLOC規模あたりのテストケース数、検出バグ数：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

11.2.8 品質実績の評価別の工程別レビュー指摘密度：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

11.2.9 品質実績の評価別の工程別レビュー工数密度：
改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

11章　品質評価別の分析

11.1　位置付け

本章では、第4章4.14項「プロジェクト成否」の品質実績の評価結果別にプロジェクトデータを層別し、生産性や信頼性を分析した。

コストや工期に関しては、実績評価別の分析結果に違いが見られなかったため、掲載しない。

図表11.1-1に、分析対象と層別のパターンを示す。図表11.1-2には要素データ“SLOC規模”と“工数”それぞれの分布状況の参照先を示す。

図表11.1-1　分析対象と層別のパターン

分析対象		層別のパターン				項番号
		開発種別	開発工程	開発言語	その他層別	
工数	工期	改良（派生）開発	開発5工程	C/C++	品質実績の評価別	11.2.1
SLOC規模	工数					11.2.2
工程別工数						11.2.3
SLOC規模	生産性					11.2.4
SLOC規模	テスト検出バグ密度					11.2.5 11.2.6
テストケース密度	テスト検出バグ密度					11.2.7
SLOC規模	工程別 *注1） レビュー指摘密度					11.2.8
SLOC規模	工程別 *注1） レビュー工数密度					11.2.9

注1）本章では、品質実績の評価結果別にレビュー工数密度とレビュー指摘密度を相対的に比較している。収集したプロジェクトデータのレビュー指摘件数の粒度や指摘内容の重要度は任意であり、レビュー体制やレビューアのスキルも異なるため、本章に掲載したレビュー指摘密度やレビュー工数密度は、一般的な参考値には適さない。

図表11.1-2　主な要素データと参照先の節番号

要素データ	参照先の節番号
SLOC規模	5.2
工数	5.3

11.2 品質実績の評価

この節では、改良(派生)開発で開発5工程(アーキテクチャ設計～総合テスト)の作業が行われたプロジェクトを対象に、品質観点でプロジェクト実績を評価した結果別の分析を示す。

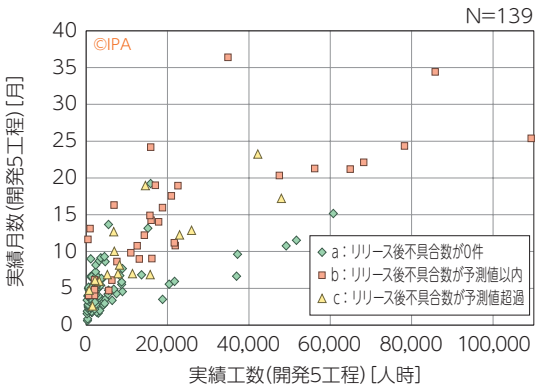
11.2.1 品質実績の評価別の工数と工期：改良(派生)開発、開発5工程、開発言語C/C++

ここでは、実績工数とその工期(月数)の関係を品質実績の評価別に示す。

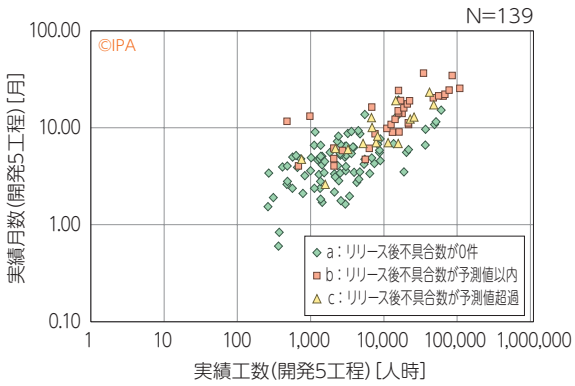
■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良(派生)開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 1-19_実績の評価(品質)が回答されているもの
- 実績工数(開発5工程) ＞ 0
- 実績月数(開発5工程) ＞ 0

■対象データ
- X軸：実績工数(開発5工程)(導出指標)
- Y軸：実績月数(開発5工程)(導出指標)

図表11.2-1 品質実績の評価別の工数と工期(改良(派生)開発、開発5工程、開発言語C/C++)



図表11.2-2 品質実績の評価別の工数と工期(改良(派生)開発、開発5工程、開発言語C/C++)対数表示



※品質実績の評価別の工数と工期について、対数化した場合の相関係数は次のようになる。

a：リリース後不具合数が0件	R=0.60 (P<0.05)
b：リリース後不具合数が予測値以内	R=0.75 (P<0.05)
c：リリース後不具合数が予測値超過	R=0.82 (P<0.05)

図表11.2-3 品質実績の評価別の実績月数の基本統計量(改良(派生)開発、開発5工程、開発言語C/C++)

[月]

品質実績の評価	N	P25	中央	P75
a：リリース後不具合数が0件	89	3.30	4.73	6.43
b：リリース後不具合数が予測値以内	34	8.98	13.57	20.00
c：リリース後不具合数が予測値超過	16	6.68	7.58	12.75

●リリース後不具合数【予測値超過】の場合、工数の増加に伴って工期が長くなる傾向が強い。

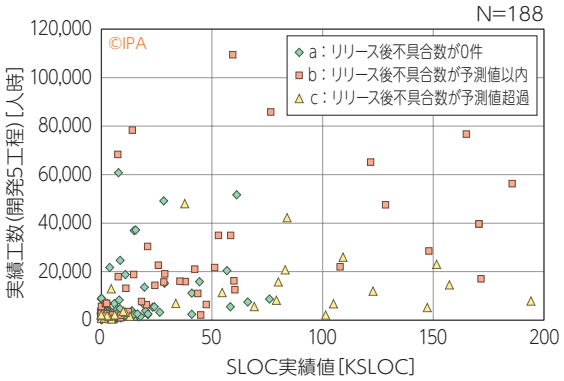
11.2.2 品質実績の評価別のSLOC規模と工数：改良（派生）開発、開発5工程、開発言語C/C++

ここでは、SLOC規模と工数の関係を品質実績の評価別に示す。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 1-19_実績の評価（品質）が回答されているもの
- 実効SLOC実績値 ＞ 0
- 実績工数（開発5工程）＞ 0

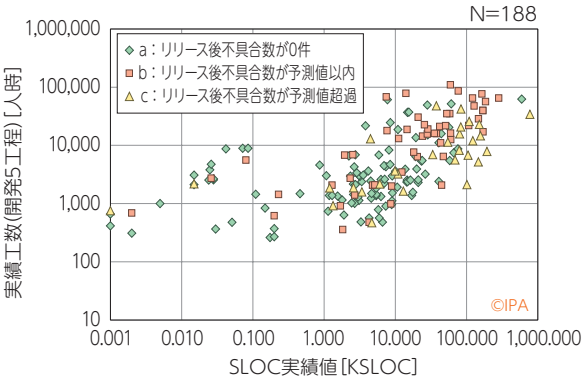
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：実績工数（開発5工程）（導出指標）

図表11.2-4　品質実績の評価別のSLOC規模と工数（改良（派生）開発、開発5工程、開発言語C/C++）



※表示されていないものが3点ある

図表11.2-5　品質実績の評価別のSLOC規模と工数（改良（派生）開発、開発5工程、開発言語C/C++）対数表示



※品質実績の評価別のSLOC規模と工数について、対数化した場合の相関係数は次のようになる。

a：リリース後不具合数が0件	R=0.46（P＜0.05）
b：リリース後不具合数が予測値以内	R=0.70（P＜0.05）
c：リリース後不具合数が予測値超過	R=0.67（P＜0.05）

図表11.2-6　品質実績の評価別の実績工数の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++）

［人時］

品質実績の評価	N	P25	中央	P75
a：リリース後不具合数が0件	106	1,336	2,543	5,215
b：リリース後不具合数が予測値以内	53	2,720	15,719	30,368
c：リリース後不具合数が予測値超過	29	2,128	6,787	14,591

●工数の増加はSLOC規模に依存するという一般的な傾向は、リリース後不具合数が【予測値以内】【予測値超過】いずれにも見られるが、【0件】の場合は比較的その傾向は弱い。

11.2.3 品質実績の評価別の工程別工数：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、工程別の実績工数の比率を品質実績の評価別に示す。

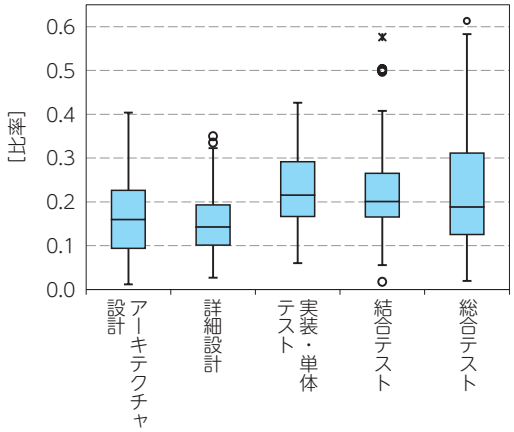
■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 1-19_実績の評価（品質）が回答されているもの
- 工程別の実績工数にすべて記入があり、各値が0より大きい
- 実効SLOC実績値 ≧ 0.1KSLOC

■対象データ
- 9-4-30_社内実績工数_アーキテクチャ設計、
  9-4-31_社内実績工数_詳細設計、
  9-4-32_社内実績工数_実装・単体テスト、
  9-4-33_社内実績工数_結合テスト、
  9-4-34_社内実績工数_総合テスト
- 9-8-2_外部委託工数_アーキテクチャ設計、
  9-8-3_外部委託工数_詳細設計、
  9-8-4_外部委託工数_実装・単体テスト、
  9-8-5_外部委託工数_結合テスト、
  9-8-6_外部委託工数_総合テスト

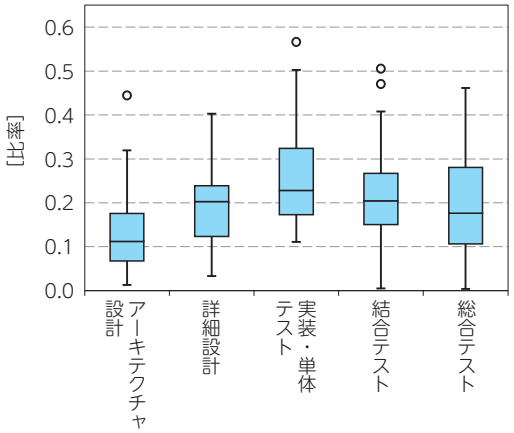
※各工程の実績工数は、社内、外部委託の実績工数合計の人時換算値を使用。

図表11.2-7 品質実績の評価別の工程別の実績工数の比率（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図

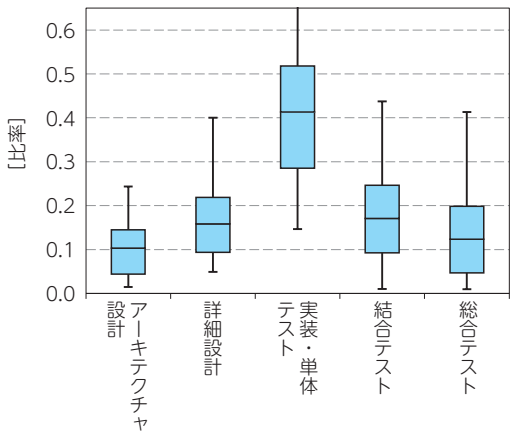
【リリース後不具合数が0件】



【リリース後不具合数が予測値以内】



【リリース後不具合数が予測値超過】



図表11.2-8　品質実績の評価別の工程別の実績工数の比率の基本統計量
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

【リリース後不具合数が0件】

［比率］

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	90	0.01	0.09	0.16	0.23	0.40	0.17	0.09
詳細設計	90	0.03	0.10	0.14	0.19	0.35	0.15	0.07
実装・単体テスト	90	0.06	0.17	0.22	0.29	0.43	0.23	0.08
結合テスト	90	0.02	0.17	0.20	0.26	0.58	0.22	0.10
総合テスト	90	0.02	0.13	0.19	0.31	0.61	0.23	0.14

【リリース後不具合数が予測値以内】

［比率］

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	50	0.01	0.07	0.11	0.18	0.44	0.13	0.09
詳細設計	50	0.03	0.12	0.20	0.24	0.67	0.20	0.11
実装・単体テスト	50	0.11	0.17	0.23	0.32	0.57	0.26	0.11
結合テスト	50	0.01	0.15	0.20	0.27	0.51	0.21	0.10
総合テスト	50	0.00	0.11	0.18	0.28	0.46	0.20	0.13

【リリース後不具合数が予測値超過】

［比率］

工程	N	最小	P25	中央	P75	最大	平均	標準偏差
アーキテクチャ設計	27	0.02	0.04	0.10	0.15	0.24	0.10	0.07
詳細設計	27	0.05	0.09	0.16	0.22	0.40	0.17	0.10
実装・単体テスト	27	0.15	0.29	0.41	0.52	0.79	0.41	0.16
結合テスト	27	0.01	0.09	0.17	0.25	0.44	0.17	0.10
総合テスト	27	0.01	0.05	0.12	0.20	0.41	0.15	0.12

- リリース後不具合数が【予測値以内】の場合には、「結合テスト」や「総合テスト」にかけた工数比率が低く、「実装・単体テスト」にかけた工数比率が高い。

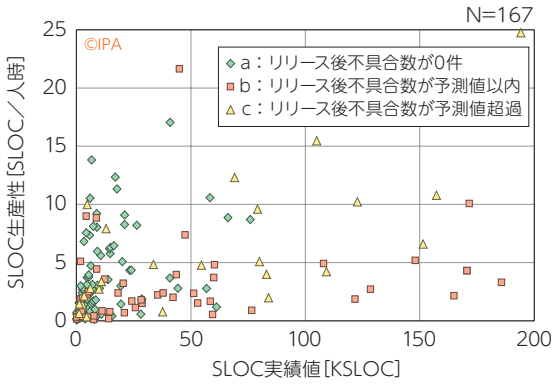
11.2.4 品質実績の評価別のSLOC規模とSLOC生産性：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模とSLOC生産性の関係を品質実績の評価別に示す。

■層別定義
- 開発5工程のそろっているもの
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 1-19_実績の評価（品質）が回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- SLOC生産性 ＞ 0

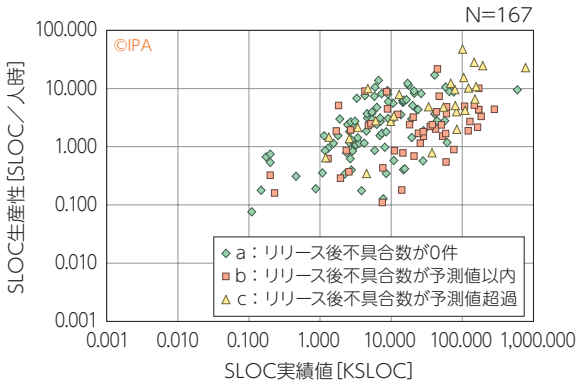
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：SLOC生産性
  （SLOC／実績工数（開発5工程））（導出指標）

図表11.2-9 品質実績の評価別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



※表示されていないものが5点ある

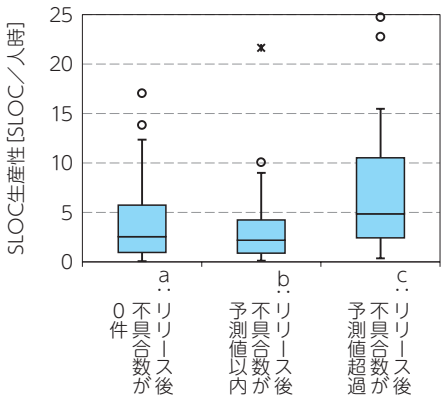
図表11.2-10 品質実績の評価別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



※品質実績の評価別のSLOC規模とSLOC生産性について、対数化した場合の相関係数は次のようになる。

a：リリース後不具合数が0件	R=0.59（P<0.05）
b：リリース後不具合数が予測値以内	R=0.50（P<0.05）
c：リリース後不具合数が予測値超過	R=0.70（P<0.05）

図表11.2-11 品質実績の評価別SLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表11.2-12 品質実績の評価別SLOC生産性の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

［SLOC／人時］

品質実績の評価	N	P25	中央	P75
a：リリース後不具合数が0件	90	0.95	2.55	5.74
b：リリース後不具合数が予測値以内	50	0.87	2.19	4.22
c：リリース後不具合数が予測値超過	27	2.43	4.86	10.51

● リリース後不具合数が【0件】と【予測値以内】の場合は、【予測値超過】の場合に比べて生産性が低い傾向にある。

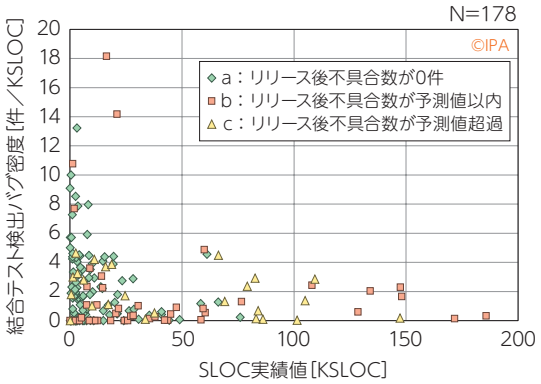
11.2.5 品質実績の評価別のSLOC規模と結合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模と結合テスト検出バグ密度の関係を品質実績の評価別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 1-19_実績の評価（品質）が回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- 10-3-1_結合テスト検出バグ現象数 ≧ 0

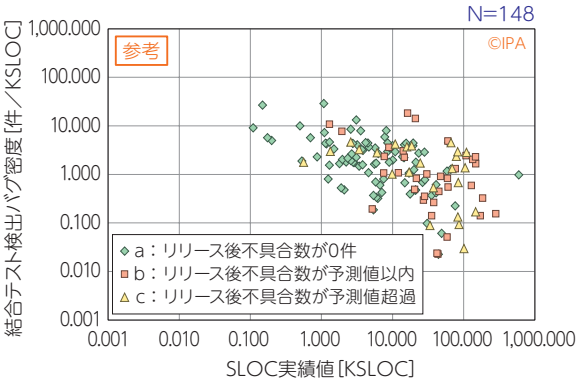
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：結合テスト検出バグ密度（SLOCあたりの検出バグ数）（導出指標）

図表11.2-13 品質実績の評価別のSLOC規模と結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



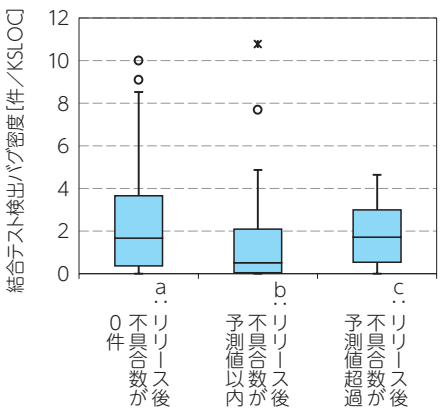
※表示されていないものが4点ある

図表11.2-14 品質実績の評価別のSLOC規模と結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件/KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

図表11.2-15 品質実績の評価別結合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表11.2-16 品質実績の評価別結合テスト検出バグ密度の基本統計量（改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

[件／KSLOC]

品質実績の評価	N	P25	中央	P75
a：リリース後不具合数が0件	109	0.367	1.667	3.656
b：リリース後不具合数が予測値以内	44	0.044	0.517	2.090
c：リリース後不具合数が予測値超過	25	0.530	1.708	2.992

●結合テスト検出バグ密度は、リリース後不具合数が【予測値以内】よりも【予測値超過】の方が高い傾向にある。【0件】の場合は、SLOC規模の分布が【予測値以内】や【予測値超過】と異なり、比較対象とならない。

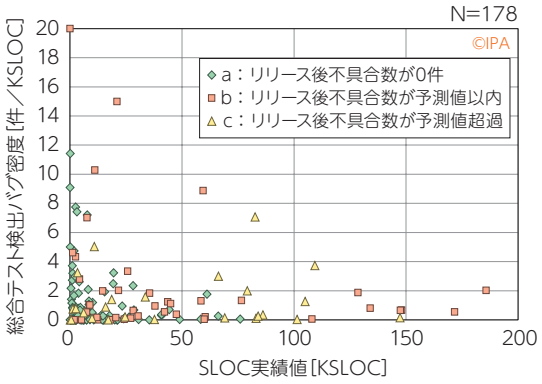
11.2.6 品質実績の評価別のSLOC規模と総合テスト検出バグ密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模と総合テスト検出バグ密度の関係を品質実績の評価別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 1-19_実績の評価（品質）が回答されているもの
- 実効SLOC実績値 ≧ 0.1KSLOC
- 10-4-1_総合テスト検出バグ現象数 ≧ 0

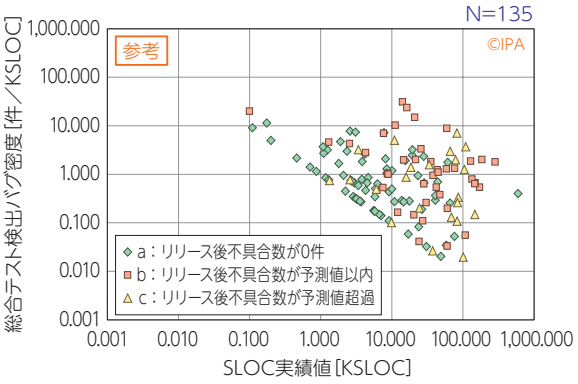
■対象データ
- X軸：実効SLOC実績値（導出指標）
- Y軸：総合テスト検出バグ密度（SLOCあたりの検出バグ数）（導出指標）

図表11.2-17 品質実績の評価別のSLOC規模と総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



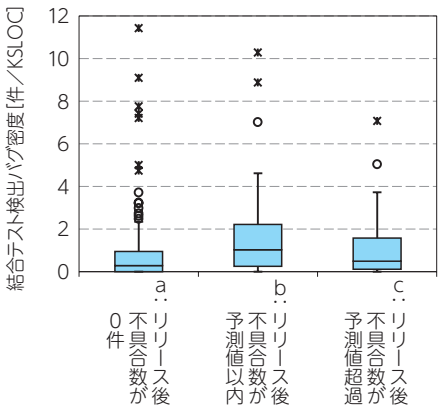
※表示されていないものが4点ある

図表11.2-18 品質実績の評価別のSLOC規模と総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件/KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

図表11.2-19 品質実績の評価別総合テスト検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表11.2-20 品質実績の評価別総合テスト検出バグ密度の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

[件／KSLOC]

品質実績の評価	N	P25	中央	P75
a：リリース後不具合数が0件	109	0.000	0.282	0.942
b：リリース後不具合数が予測値以内	44	0.246	1.023	2.216
c：リリース後不具合数が予測値超過	25	0.108	0.493	1.575

●総合テスト検出バグ密度は、リリース後不具合数が【予測値以内】よりも【予測値超過】の方が低い傾向にある。【0件】の場合は、SLOC規模の分布が【予測値以内】や【予測値超過】と異なり、比較対象とならない。

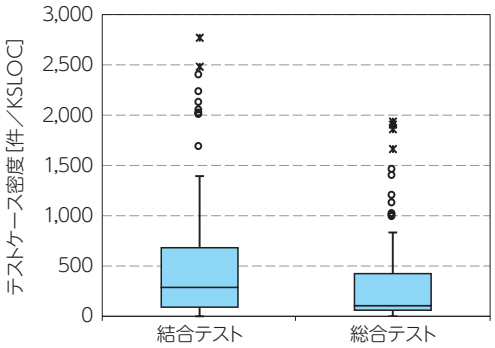
11.2.7 品質実績の評価別のSLOC規模あたりのテストケース数、検出バグ数：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

ここでは、SLOC規模あたりのテストケース数と検出バグ数を品質実績の評価別に示す。

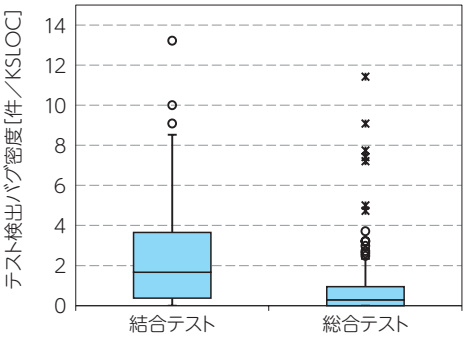
■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 1-19_実績の評価（品質）が回答されているもの
- 結合テスト、総合テストの両方を実施
- 実効SLOC実績値 ≧ 0.1KSLOC

■対象データ
- テストケース数
  （10-1-1_結合テストケース数、
  10-2-1総合テストケース数）
- 検出バグ現象数
  （10-3-1_結合テスト検出バグ現象数、
  10-4-1_総合テスト検出バグ現象数）

図表11.2-21 品質実績の評価【リリース後不具合数が0件】のSLOC規模あたりのテストケース数（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表11.2-22 品質実績の評価【リリース後不具合数が0件】のSLOC規模あたりの検出バグ数（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図

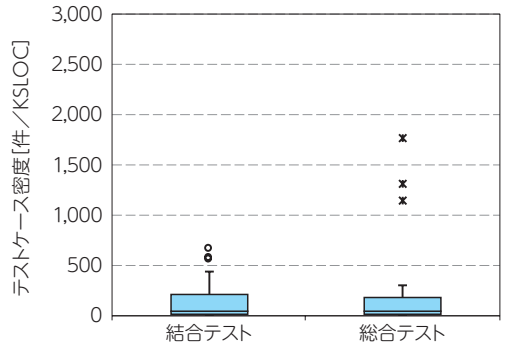


図表11.2-23 品質実績の評価【リリース後不具合数が0件】のSLOC規模あたりのテストケース数、検出バグ数の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

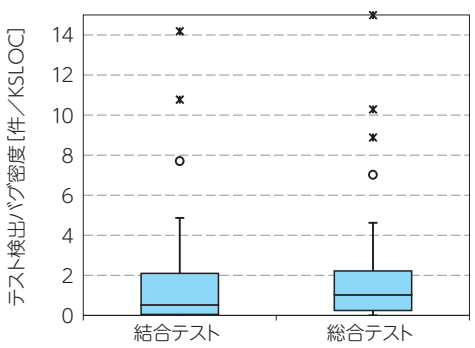
［件／KSLOC］

	N	P25	中央	P75
結合テストケース密度	109	84.622	289.756	681.132
総合テストケース密度	109	52.609	105.161	415.637
結合テスト検出バグ密度	109	0.367	1.667	3.656
総合テスト検出バグ密度	109	0.000	0.282	0.942

図表11.2-24 品質実績の評価【リリース後不具合数が予測値以内】のSLOC規模あたりのテストケース数（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表11.2-25 品質実績の評価【リリース後不具合数が予測値以内】のSLOC規模あたりの検出バグ数（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図

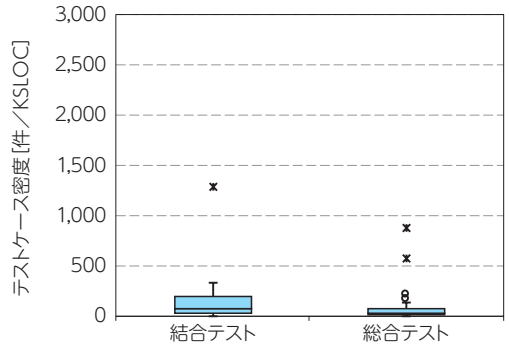


図表11.2-26 品質実績の評価【リリース後不具合数が予測値以内】のSLOC規模あたりのテストケース数、検出バグ数の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

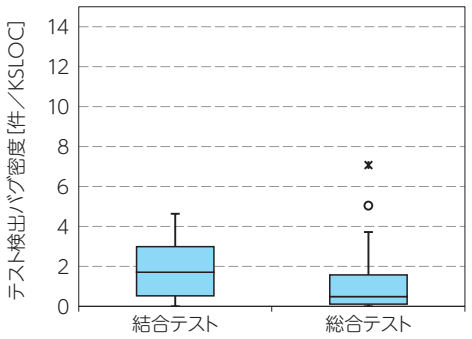
[件／KSLOC]

	N	P25	中央	P75
結合テストケース密度	44	9.785	44.858	204.100
総合テストケース密度	44	15.682	40.860	182.549
結合テスト検出バグ密度	44	0.044	0.517	2.090
総合テスト検出バグ密度	44	0.246	1.023	2.216

図表11.2-27 品質実績の評価【リリース後不具合数が予測値超過】のSLOC規模あたりのテストケース数（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表11.2-28 品質実績の評価【リリース後不具合数が予測値超過】のSLOC規模あたりの検出バグ数（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表11.2-29 品質実績の評価【リリース後不具合数が予測値超過】のSLOC規模あたりのテストケース数、検出バグ数の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

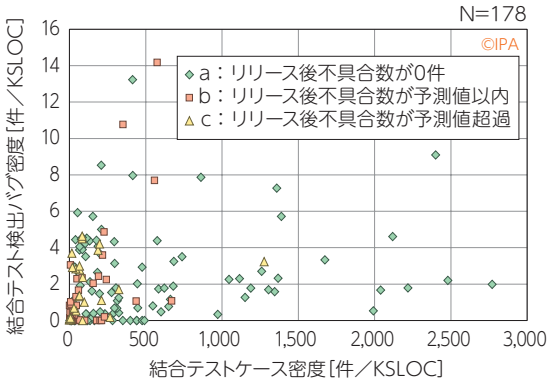
[件／KSLOC]

	N	P25	中央	P75
結合テストケース密度	25	29.762	67.521	188.218
総合テストケース密度	25	14.959	29.266	68.641
結合テスト検出バグ密度	25	0.530	1.708	2.992
総合テスト検出バグ密度	25	0.108	0.493	1.575

- リリース後不具合数が【予測値以内】と【予測値超過】のテストケース密度について、中央値で比較すると、「結合テスト」は、【予測値超過】の方が高く、「総合テスト」は、【予測値以内】の方が高い。リリース後不具合数が【0件】の場合は、SLOC規模の分布が【予測値以内】や【予測値超過】と異なるため中央値で比較することはできない（11.2.5項の図表11.2-13及び図表11.2-14、11.2.6項の図表11.2-17及び図表11.2-18参照）。

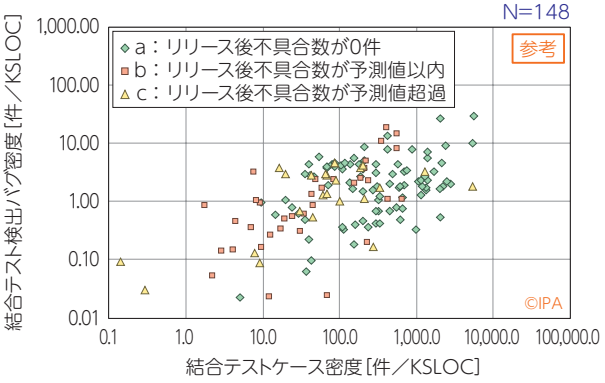
以下に、テストケース密度とテスト検出バグ密度の関係を示す。

図表11.2-30　品質実績の評価別の結合テストケース密度と検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

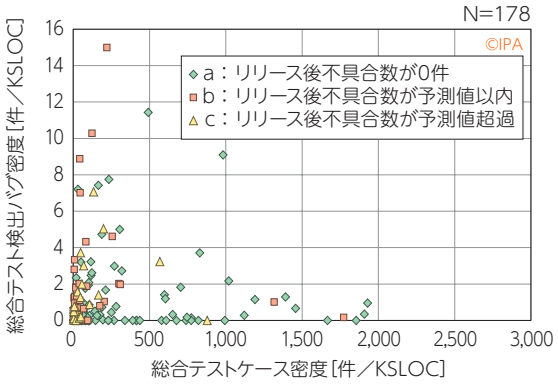


※表示されていないものが7点ある

図表11.2-31　品質実績の評価別の結合テストケース密度と検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示

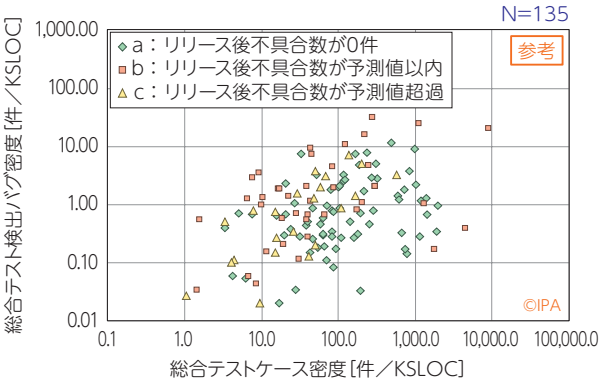


図表11.2-32　品質実績の評価別の総合テストケース密度と検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



※表示されていないものが5点ある

図表11.2-33　品質実績の評価別の総合テストケース密度と検出バグ密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示では検出バグ密度が“0件/KSLOC”より大きい標本を対象にしている。SLOC規模に対する検出バグ密度の増減傾向が視覚的に分かりやすい。

11.2.8 品質実績の評価別の工程別レビュー指摘密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

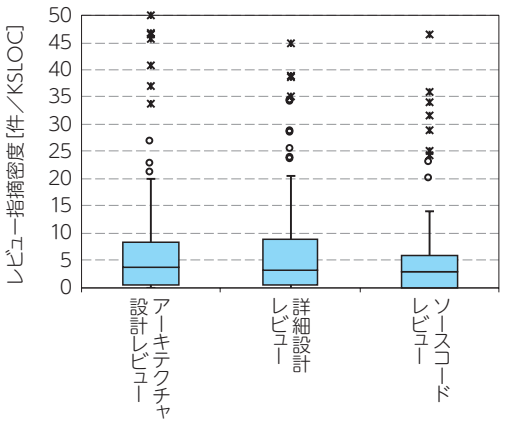
ここでは、工程別レビュー指摘密度を品質実績の評価別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 1-19_実績の評価（品質）が回答されているもの
- アーキテクチャ設計レビュー、詳細設計レビュー、ソースコードレビューの何れかを実施
- 実効SLOC実績値 ≧ 0.1KSLOC

■対象データ
- レビュー指摘件数
  （9-7-2_アーキテクチャ設計レビュー指摘数、
  9-7-3_詳細設計レビュー指摘数、
  9-7-4_ソースコードレビュー指摘数）

※各工程のレビュー指摘数データがない場合、“0”で補完する。

図表11.2-34 品質実績の評価【リリース後不具合数が0件】の工程別レビュー指摘密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図

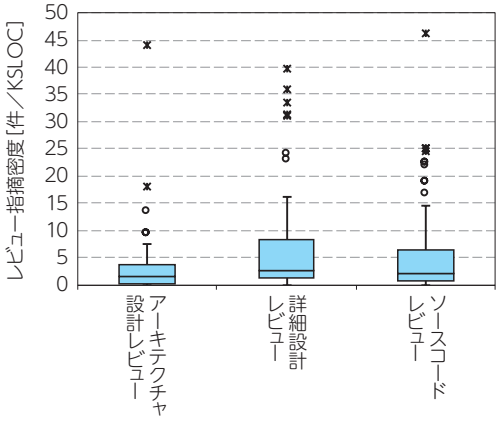


図表11.2-35 品質実績の評価【リリース後不具合数が0件】の工程別レビュー指摘密度の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

[件／KSLOC]

	N	P25	中央	P75
アーキテクチャ設計レビュー	144	0.52	3.75	8.28
詳細設計レビュー	144	1.15	3.96	9.46
ソースコードレビュー	144	0.44	3.32	6.36

図表11.2-36 品質実績の評価【リリース後不具合数が予測値以内】の工程別レビュー指摘密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図

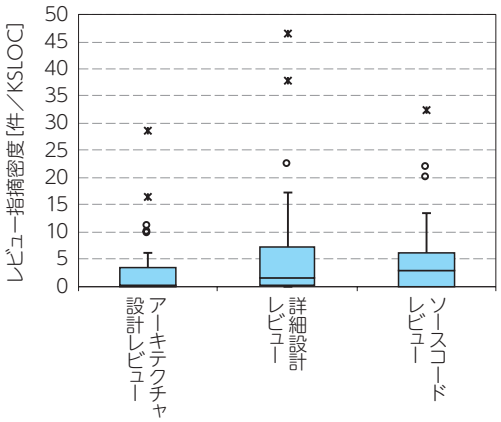


図表11.2-37 品質実績の評価【リリース後不具合数が予測値以内】の工程別レビュー指摘密度の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

[件／KSLOC]

	N	P25	中央	P75
アーキテクチャ設計レビュー	63	0.29	1.53	3.81
詳細設計レビュー	63	1.22	2.72	8.30
ソースコードレビュー	63	0.59	1.98	6.33

図表11.2-38　品質実績の評価【リリース後不具合数が予測値超過】の工程別レビュー指摘密度（改良（派生）開発、開発５工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表11.2-39　品質実績の評価【リリース後不具合数が予測値超過】の工程別レビュー指摘密度の基本統計量（改良（派生）開発、開発５工程、開発言語C/C++、0.1 KSLOC以上）

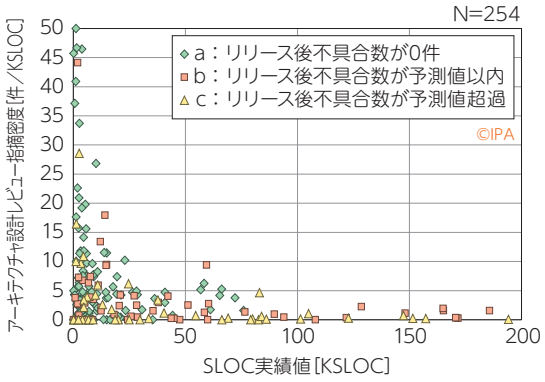
［件／KSLOC］

	N	P25	中央	P75
アーキテクチャ設計レビュー	47	0.00	0.23	3.40
詳細設計レビュー	47	0.13	1.48	7.33
ソースコードレビュー	47	0.00	3.01	6.05

● リリース後不具合数が【予測値以内】と【予測値超過】を比べると、アーキテクチャ設計レビューの指摘密度が高い（中央値）のは、【予測値以内】の方であり、ソースコードレビューの指摘密度が高い（中央値）のは、【予測値超過】の方である。中央値で見ると、上流工程でのレビュー指摘密度が高いとリリース後の品質が良い傾向が見られるが、P25からP75の範囲で比べると大きな違いは見られない。

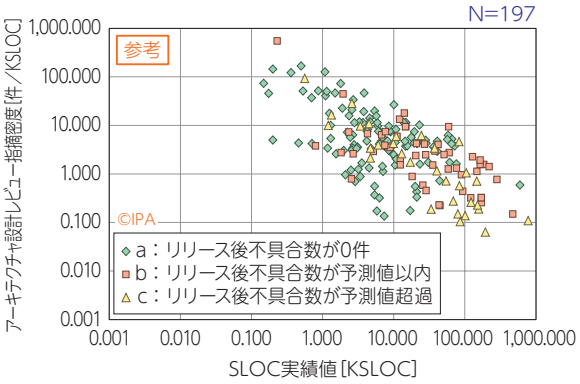
以下に、SLOC規模とレビュー指摘密度の関係を示す。

図表11.2-40 品質実績の評価別のアーキテクチャ設計レビュー指摘密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

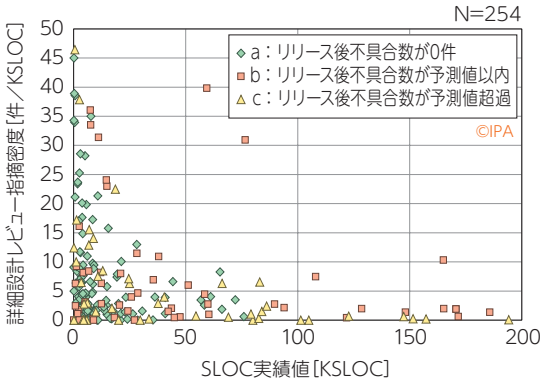


※表示されていないものが16点ある

図表11.2-41 品質実績の評価別のアーキテクチャ設計レビュー指摘密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示

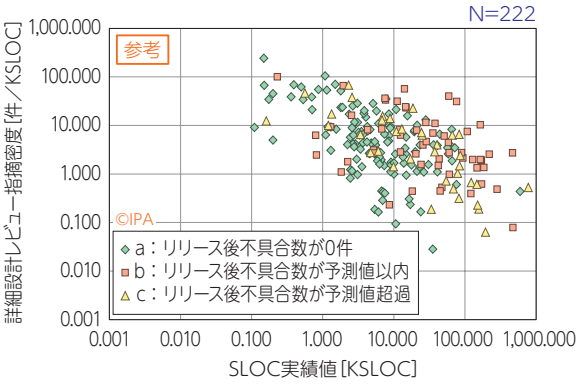


図表11.2-42 品質実績の評価別の詳細設計レビュー指摘密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

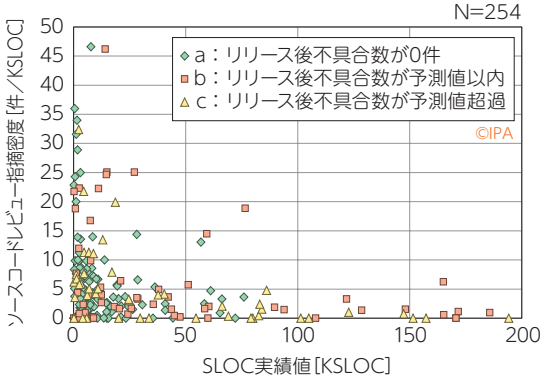


※表示されていないものが19点ある

図表11.2-43 品質実績の評価別の詳細設計レビュー指摘密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示

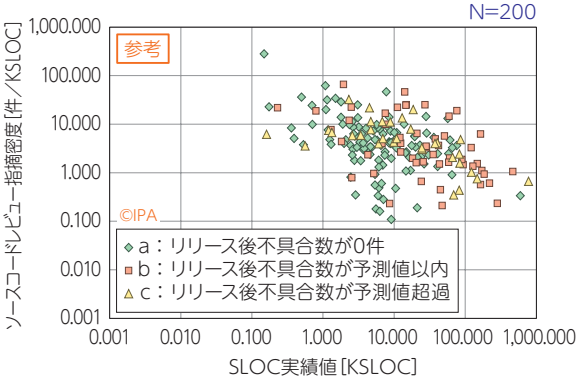


図表11.2-44 品質実績の評価別のソースコードレビュー指摘密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



※表示されていないものが9点ある

図表11.2-45 品質実績の評価別のソースコードレビュー指摘密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示ではレビュー指摘密度が“0件/KSLOC”より大きい標本を対象にしている。SLOC規模に対するレビュー指摘密度の増減傾向が視覚的に分かりやすい。

11.2.9 品質実績の評価別の工程別レビュー工数密度：改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上

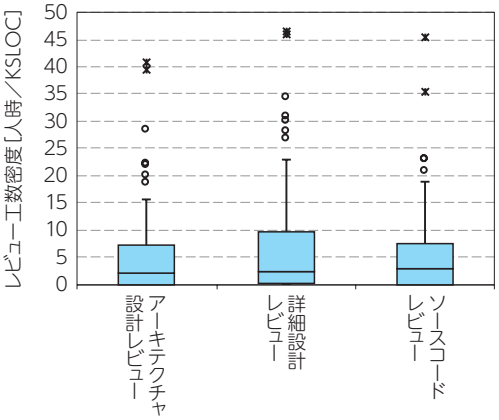
ここでは、工程別レビュー工数密度を品質実績の評価別に示す。

■層別定義
- 開発5工程のフェーズ有無がすべて○
- 1-5_開発プロジェクトの種別がb：改良（派生）開発
- 3-9_主開発言語1がb：言語C、c：言語C++のいずれか
- 1-19_実績の評価（品質）が回答されているもの
- アーキテクチャ設計レビュー、詳細設計レビュー、ソースコードレビューの何れかを実施
- 実効SLOC実績値 ≧ 0.1KSLOC

■対象データ
- レビュー実績工数
  （9-5-2_アーキテクチャ設計レビュー実績工数、
  9-5-3_詳細設計レビュー実績工数、
  9-5-4_ソースコードレビュー実績工数）

※各工程のレビュー実績工数データがない場合、“0”で補完する。

図表11.2-46　品質実績の評価【リリース後不具合数が0件】の工程別レビュー工数密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図

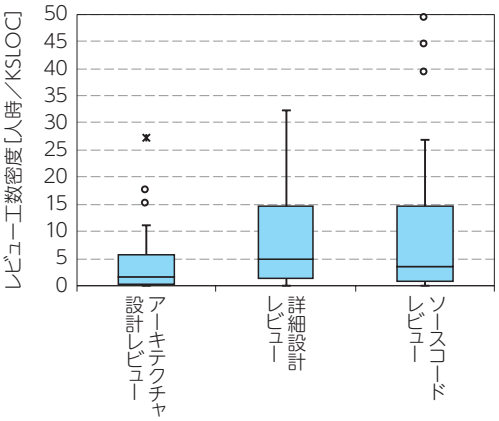


図表11.2-47　品質実績の評価【リリース後不具合数が0件】の工程別レビュー工数密度の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

［人時／KSLOC］

	N	P25	中央	P75
アーキテクチャ設計レビュー	127	0.01	2.06	7.34
詳細設計レビュー	127	0.06	2.45	9.52
ソースコードレビュー	127	0.02	2.88	7.54

図表11.2-48　品質実績の評価【リリース後不具合数が予測値以内】の工程別レビュー工数密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図

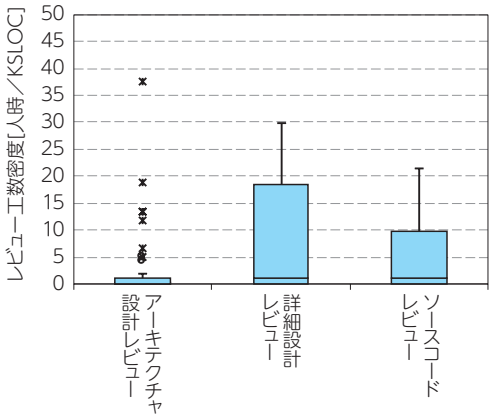


図表11.2-49　品質実績の評価【リリース後不具合数が予測値以内】の工程別レビュー工数密度の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

［人時／KSLOC］

	N	P25	中央	P75
アーキテクチャ設計レビュー	61	0.43	1.63	5.79
詳細設計レビュー	61	1.39	5.01	14.69
ソースコードレビュー	61	0.92	3.68	14.66

図表11.2-50　品質実績の評価【リリース後不具合数が予測値超過】の工程別レビュー工数密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図



図表11.2-51　品質実績の評価【リリース後不具合数が予測値超過】の工程別レビュー工数密度の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

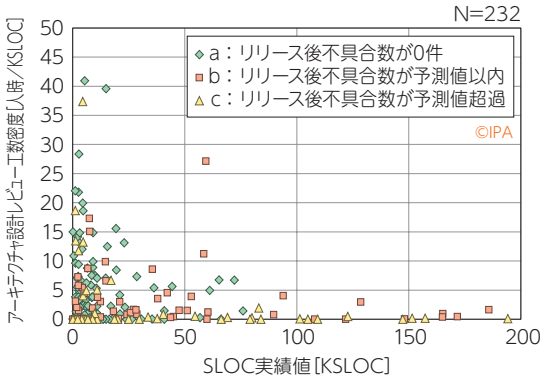
［人時／KSLOC］

	N	P25	中央	P75
アーキテクチャ設計レビュー	44	0.00	0.10	1.24
詳細設計レビュー	44	0.09	1.25	18.59
ソースコードレビュー	44	0.00	1.07	9.90

- リリース後不具合数が【予測値以内】と【予測値超過】を比べると、【予測値以内】の方がアーキテクチャ設計レビューの工数密度が高い（P25、中央値、P75全て）傾向にある。

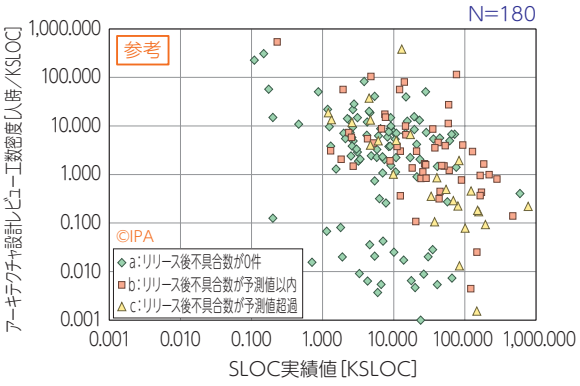
以下に、SLOC規模とレビュー工数密度の関係を示す。

図表11.2-52　品質実績の評価別のアーキテクチャ設計レビュー工数密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

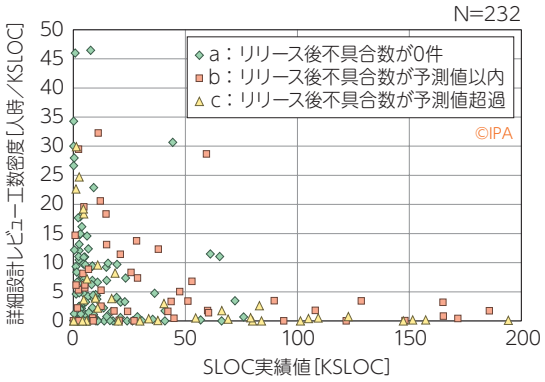


※表示されていないものが18点ある

図表11.2-53　品質実績の評価別のアーキテクチャ設計レビュー工数密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示

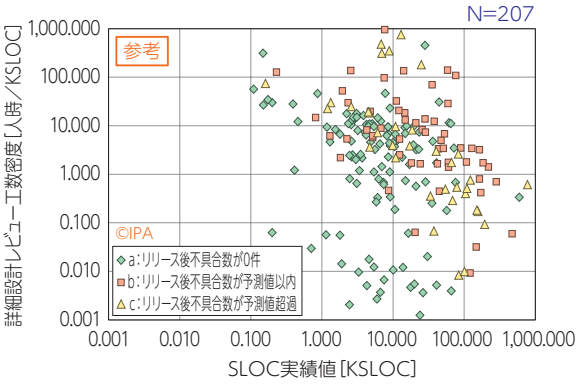


図表11.2-54　品質実績の評価別の詳細設計レビュー工数密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）

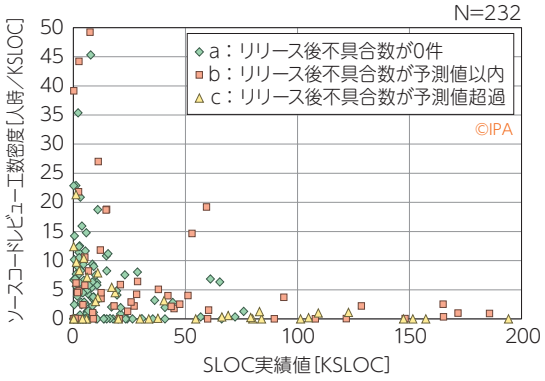


※表示されていないものが24点ある

図表11.2-55　品質実績の評価別の詳細設計レビュー工数密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示

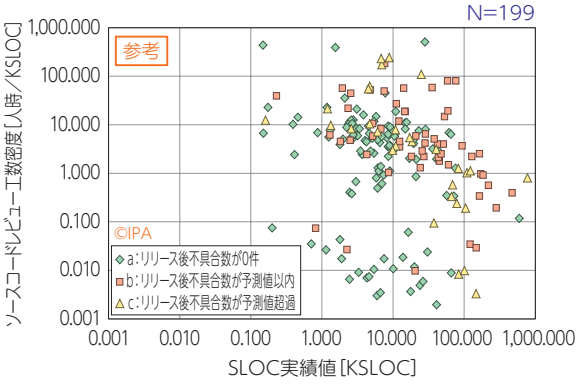


図表11.2-56　品質実績の評価別のソースコードレビュー工数密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）



※表示されていないものが23点ある

図表11.2-57　品質実績の評価別のソースコードレビュー工数密度（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示



【参考】対数表示ではレビュー工数密度が“0件/KSLOC”より大きい標本を対象にしている。SLOC規模に対するレビュー工数密度の増減傾向が視覚的に分かりやすい。

付　録

付録A　データ項目の定義 …… 196

A.1　工程の呼称
A.2　データ項目定義 Version 4.12
A.3　導出指標の名称と定義

付録B　用語集 …… 215

付録C　参考文献・参考情報 …… 217

C.1　参考文献
C.2　参考情報

付録

付録A　データ項目の定義

A.1　工程の呼称

次の表に、本データ項目定義で使用されているソフトウェア開発工程の名称を示す。これらの名称は『組込みソフトウェア開発　プロセスガイド(ESPR)』に定義されている。

工程	ESPRのソフトウェア・エンジニアリング・プロセス	ESPRの定義
要求定義	ソフトウェア 要求定義	このアクティビティは、 システム要求仕様とハードウェア仕様をもとに、ソフトウェア要求仕様を検討する際の制約条件を確認した上で、ソフトウェアに必要となる機能面の要求を明確にし、また、ソフトウェアに必要な非機能面の要求も合わせて明確にする。 実際のソフトウェア開発では開発期間やリソース面の制約、あるいは、システムとしての制約などがあるため、これらを考慮したうえで、個々の機能、非機能要求に優先順位をつけていく。 これらの検討結果を整理し、ソフトウェア要求仕様書を作成する。 作成したソフトウェア要求仕様書はあらかじめ確認ポイントを決めて確認を行い、確認した内容を整理して内部確認レポートを作成する。
アーキテクチャ設計	ソフトウェア・ アーキテクチャ 設計	このアクティビティは、 ソフトウェア要求仕様をもとに、実現すべき機能、非機能要求を確認する。 また、要求実現上の制約などを仕様実現の視点で確認する。 要求仕様を実現するためのソフトウェアとしての振る舞いと構成を検討し、ソフトウェアを構成する機能ユニット間のインタフェースを設計する。 また、ソフトウェアを実装するハードウェアを考慮し、性能やメモリ量なども検討し見積もる。 これらの検討結果を整理し、ソフトウェア・アーキテクチャ設計書を作成する。 作成したソフトウェア・アーキテクチャ設計書はあらかじめ確認ポイントを決めて確認を行い、確認した内容を整理して内部確認レポートを作成する。また、関係者を交えて共同レビューを実施し、レビューの結果を整理して共同レビュー報告書を作成する。
詳細設計	ソフトウェア 詳細設計	このアクティビティは、 ソフトウェア要求仕様とソフトウェア・アーキテクチャ設計をもとに、当該システムとして提供する機能を実現するために、組込みソフトウェアを実装の視点からいくつかの部分(プログラムユニットと呼ぶ)に分割し、それぞれのプログラムユニットで行うべき処理内容や論理処理を明確にするとともに、それぞれのプログラムユニット間の関係を明確にし、そのインタフェースを決め、実装の視点から、使用するメモリ量の見積もりを出す。 これらの検討結果を整理し、ソフトウェア詳細設計書を作成する。 作成したソフトウェア詳細設計書はあらかじめ確認ポイントを決めて確認を行い、確認した内容を整理して内部確認レポートを作成する。 また、関係者を交えてハードウェア仕様との整合性の確認を行い、確認した内容を整理してハードウェア仕様との整合性の確認レポートを作成する。

工程	ESPRのソフトウェア・エンジニアリング・プロセス	ESPRの定義
実装・単体テスト	実装・単体テスト	このアクティビティは、 組込みソフトウェアのプログラム実装を行うための開発環境や再利用するプログラムなどを用意し、実装したプログラムユニットの単体レベルでの確認の準備を行い、各プログラムユニットをソフトウェア詳細設計に従って実装し単体テストを実施する。 また実装完了したプログラムユニットの実装内容および単体テスト結果は、あらかじめ確認ポイントを決めて確認を行い、確認した内容を整理して内部確認レポートを作成する。
結合テスト	ソフトウェア 結合テスト	このアクティビティは、 個々のプログラムユニットを結合するためのmakeファイルをはじめとするソフトウェア結合の準備をして、ソフトウェア結合レベルのテストの準備を進め、個々のプログラムユニットの結合と結合テストを行い、その結果を確認する。 ソフトウェア結合テスト結果はあらかじめ確認ポイントを決めて確認を行い、確認した内容を整理して内部確認レポートを作成する。
総合テスト	ソフトウェア 総合テスト	このアクティビティは、 結合が完了したソフトウェアが、ソフトウェア要求仕様に記載された機能動作を実現しているかどうかを確認するためのテスト仕様を準備し、実際にそのテスト仕様に従い、ソフトウェアを動作させ、ソフトウェアとしての総合的なテストを行う。 ソフトウェア総合テスト結果については、テストの合否判定基準に従って確認を行い、確認した内容を整理して内部確認レポートを作成する。 また、関係者を交えてソフトウェア総合テスト結果の共同レビューを実施してソフトウェア開発の最終確認を行い、レビューの結果等を整理してソフトウェア開発完了報告書を作成する。

A.2　データ項目定義 Version 4.12

この節では、本白書で使用しているデータ項目の定義を示す。本書で扱ったプロジェクトデータは、この定義に従って収集し、分析を行った。

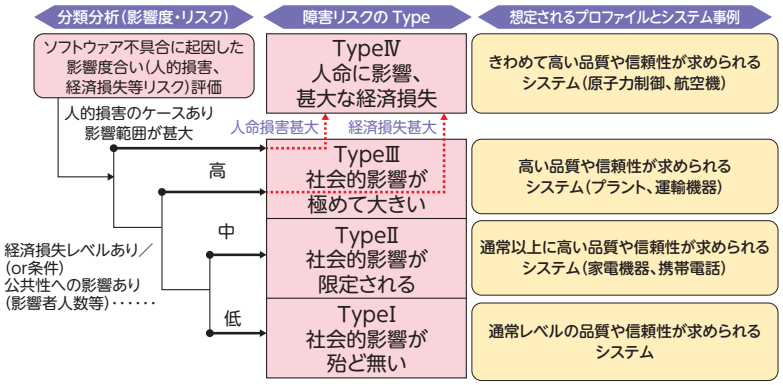
表の「データ名称」列は、データ項目の名称を示す。名称は“項番_名前”という書式となっている。「定義」列は、データ項目の定義の説明である。「回答内容、選択肢」列は、データ収集フォームでの回答方法（質問内容）を示しており、選択式の場合は選択肢の一覧を、自由記入の場合は（　　）と記載している。また、自由記入の場合の回答例や補足説明を記載したものもある。

(1) 開発プロジェクト全般

データ名称	定義	回答内容、選択肢
1-1_各社採番のプロジェクトID	各社にてプロジェクトを識別するためのID。サブシステムの識別にも利用。 ※サブシステム単位でデータを捕捉できている場合に、それらを集約しないこと。	1、2、3、・・・：全体システムの場合 1-1、1-2、・・・：全体システム1のサブシステムの場合
1-5_開発プロジェクトの種別	開発プロジェクトの種別（新規か改良か）。 ※別製品のソフトウェアを流用して開発する場合も「b：改良（派生）開発」とする。	a：新規開発：ベースとなる既存製品が存在せず、新規の開発を行うもの。 b：改良（派生）開発：既存製品が存在し、機能追加・変更など改修を伴う次期製品開発を行うもの。
1-5a_経過年数	改良（派生）開発の場合は、これまでに保守または維持管理されて来た期間（年数）を記載する。 ※1年未満の期間は切り上げる。	（　　）年
1-5a_保守する年数	開発プロジェクト終了後に保守または維持管理しなければならない年数（製品寿命）を記載する。 ※1年未満の期間は切り上げる。	（　　）年
1-6_開発作業場所	「受託開発」の場合は、その開発作業場所。	a：顧客先 b：自社 c：その他（具体的記述）
1-7_開発プロジェクトの概要	開発プロジェクトに含まれる作業を全て記載する。	a：ソフトウェア開発 b：ハードウェア開発 c：運用構築 d：リプレース e：保守 f：品質保証 g：現地（実環境）テスト h：テスト環境構築 i：プロジェクト管理 j：コンサルティング k：顧客教育 l：その他（具体的名称）
1-8_外部委託先情報	外部委託が有る場合に、外部委託先の情報を主要なものから3つまで記載する。 ※系列＝資本関係有りの企業	a：日本企業（グループ内／系列） b：日本企業（グループ外／系列外） c：海外企業（グループ内／系列） d：海外企業（グループ外／系列外） e：外部委託なし
1-9_外部委託先国名	1-8が「c：海外企業（グループ内/系列）」、「d：海外企業（グループ外/系列外）」の場合に、国名を記述する（複数記入可）。 例. 中国、インド	（　　）
1-10_新規外部委託先か否か	新規の協力会社へ委託したか否か。 ※1-8で外部委託先がある場合（a～d）に記載する。	a：外部委託先の中に新規の委託先が含まれる b：外部委託先はすべて2回以上の委託実績がある

データ名称	定義	回答内容、選択肢
1-11_新技術を利用する開発か否か	新しい技術を利用する開発か否か。 〔補足説明〕プロジェクトメンバにとって経験のない技術を利用している場合に、「a：新技術を利用」を選択する。	a：新技術を利用 b：新技術を利用していない
1-12_開発プロジェクトチーム内での役割分担・責任所在の明確さ	開発プロジェクトチーム内での役割分担・責任所在の明確度合い。 例. a：文書で明確に定義されている b：文書で定義されていないが明確である c：曖昧な部分があり、認識が相違する場合がある d：不明確であり、問題となる場合がある	a：非常に明確 b：概ね明確 c：やや不明確 d：不明確
1-13_達成目標と優先度の明確さ	納期・品質・技術開発等の達成目標と優先度の明確度合い。 〔補足説明〕「a：非常に明確」とは、Q（品質）、C（コスト）、D（納期）の数値目標および優先度が、文書で明確に定義されていること。 例. a：文書で明確に定義されている b：文書で定義されていないが明確である c：曖昧な部分があり、認識が相違する場合がある d：不明確であり、問題となる場合がある	a：非常に明確 b：概ね明確 c：やや不明確 d：不明確
1-15_計画の評価（コスト）	プロジェクト計画時のコスト計画の妥当性を評価する。	a：コスト算定の根拠が明確で実行可能性を検討済み b：コスト算定の根拠が不明確、又は実行可能性を未検討 c：計画なし
1-16_計画の評価（品質）	プロジェクト計画時の出荷後品質の目標の妥当性を評価する。	a：品質目標が明確で実行可能性を検討済み b：品質目標が不明確、又は実行可能性を未検討 c：計画なし
1-17_計画の評価（工期）	プロジェクト計画時の工期計画の妥当性を評価する。	a：工期計画の根拠が明確で実行可能性を検討済み b：工期計画の根拠が不明確、又は実行可能性を未検討 c：計画なし
1-18_実績の評価（コスト）	コスト計画に対する実績の評価。	a：計画より10%以上少ないコストで達成 b：計画通り（±10%未満） c：計画の30%以内の超過 d：計画の50%以内の超過 e：計画の50%を超える超過
1-19_実績の評価（品質）	品質計画（開発完了時品質の目標）に対する実績の評価。	リリース後不具合数が a：0件 b：予測値以内（1件以上の場合） c：予測値超過
1-20_実績の評価（工期）	工期計画に対する実績の評価。定めた又は顧客（OEM含む）と合意した納期に対する遅延状況で評価する。	a：納期より前倒し b：納期通り c：納期を1ヵ月未満遅延 d：納期を6ヵ月未満遅延 e：納期を6ヵ月以上遅延
1-22_総括コメント	提供データについて、分析時に考慮すべき点やIPAへの連絡事項など。 例1. 外部委託があるが、比率が分からず、記入していない。 例2. 社内の開発工数に実機環境構築対応作業を約3割含む。	（　　） ※全角256文字まで。

(2) 利用局面

データ名称	定義	回答内容、選択肢
2-1_利用形態	開発した製品の利用形態(特定ユーザの利用か、不特定ユーザの利用か)。	a：特定ユーザの利用 b：不特定ユーザの利用
2-2_障害リスク	製品のTypeソフトウェア不具合に起因した影響度合い(人的損害・経済損失等)を把握して、システムを分類。	TypeⅣ：人命に影響、甚大な経済損失 TypeⅢ：社会的影響が極めて大きい TypeⅡ：社会的影響が限定される TypeⅠ：社会的影響が殆どない 	
2-3_製品分類	製品の分類 a：AV機器 (TV、DVD、デジタルカメラ、オーディオ機器等) b：家電機器 (電子レンジ、炊飯器、エアコン、洗濯機、冷蔵庫等) c：個人用情報機器 (PDA、電子手帳、GPS、カーナビ等) d：教育機器、娯楽機器 (ゲーム機、電子楽器、電子辞書、玩具ロボット等) e：コンピュータ周辺機器/OA機器 (プリンタ、複写機/複合機、ストレージ機器等) f：業務用端末機器 (POS機器、金融端末、自動改札機、自動販売機等) g：民生用通信端末機器 (固定電話機、携帯電話端末等) h：通信設備機器等 (ルータ、通信網用スイッチ、放送機器、無線機器等) i：運輸機器/建設機器 (自動車、船舶、飛行機、オートバイ、ブルドーザ/ショベル等) j：工業制御/FA機器/産業機器 (プラント制御、工業用ロボット、縫製機械等) k：設備機器 (エレベータ/エスカレータ、照明機器、空調機器等) l：医療機器 (診断・検査装置、個人用健康管理機器、福祉・介護機器等) m：分析機器・計測機器等 (分光光度計、ロジックアナライザ、電子顕微鏡等) n：その他の応用機器製品	a：AV機器 b：家電機器 c：個人用情報機器 d：教育機器、娯楽機器 e：コンピュータ周辺機器/OA機器 f：業務用端末機器 g：民生用通信端末機器 h：通信設備機器等 i：運輸機器/建設機器 j：工業制御/FA機器/産業機器 k：設備機器 l：医療機器 m：分析機器・計測機器等 n：その他の応用機器製品(任意記入) o：非回答

(3) システム特性

データ名称		定義	回答内容、選択肢
3-1_製品の特性	3-1-1_リアルタイム性（時間制約）	リアルタイム性（時間制約）の強弱。	a：強（μ秒オーダー） b：普通（m秒オーダー） c：気にしなくて良い
	3-1-2_自然環境からの影響度合い	気象、地形、位置（緯度・経度・高度）等の影響の有無。	a：有り b：無し
	3-1-3_ユーザの多様性	年齢・性別・人種・地域等の不特定多数のユーザの多少。	a：多い b：少ない
	3-1-4_法規等による規制度合い	米国FDA規制、機能安全規格等のソフトウェア品質に関わる規格の有無。	a：有り b：無し
	3-1-5_M2Mの有無	M2M通信機能の有無。	a：有り b：無し
	3-1-6_ネットワーク接続の有無	他の装置やシステムとネットワークを介して接続する機能の有無。	a：有り b：無し
	3-1-7_稼動（非停止、オンデマンド）	非停止システムか、又はオンデマンドで使用されるものか。	a：非停止 b：オンデマンド
	3-1-8_オンライン保守の可否	プログラム更新をオンライン保守機能により実施可能かどうか。	a：可 b：否
3-2_（市販）パッケージ利用の有無		当該プロジェクトにおける（市販・オープンソフト）パッケージソフトの利用の有無。 ※自社開発したパッケージソフトは除く。	a：有り b：無し
3-3_（市販）パッケージの初回利用か否か		（市販・オープンソフト）パッケージが「有り」の場合、そのパッケージを初めて利用するのか否か。	a：初回利用 b：過去に経験有り c：経験度合いがわからない
3-4_（市販）パッケージの名称		（市販・オープンソフト）パッケージ利用「有り」の場合、パッケージの名称。	（　　）
3-5_パッケージの機能規模の比率		（市販・オープンソフト）パッケージ利用「有り」の場合、システム全体の機能規模に対するパッケージの機能規模の概算比率（厳密な値でなくても良い）。	約（　　）%
3-6_パッケージのカスタマイズの度合い		（市販・オープンソフト）パッケージ利用「有り」の場合、カスタマイズ金額÷パッケージの金額。フリーソフトの場合は、100%と記入。	（　　）%
3-7_CPUアーキテクチャ		CPUの個数。	a：シングルチップ シングルコア b：シングルチップ マルチコア（対称） c：シングルチップ マルチコア（非対称） d：マルチプロセッサ（対称） e：マルチプロセッサ（非対称）
3-7a_OSアーキテクチャ		リアルタイムOS利用の有無。	a：リアルタイムOSを利用している b：リアルタイムOSを利用していない
3-8_OSの種類		主たる開発対象のOS。 ※主なものが複数ある場合は2つまで記入。	a：Windows系（CE等も含む） b：Linux系 c：ITRON系 d：VxWorks e：OS9 f：QNX g：BSD系 h：Android i：iOS j：その他OS（任意記入） k：OSなし

データ名称		定義	回答内容、選択肢
3-9_主開発言語	3-9-1_主開発言語1	主たる開発言語。 ※規模の大きい順に5つまで選択。 ※Web系のCGI、Javaアプレット、EJBなど、選択肢にないものは、「その他言語」を選び、具体的名称を記述すること。	a：アセンブラ b：C c：C++ d：C# e：java f：HTML g：その他言語（任意記入）
	3-9-2_主開発言語2		
	3-9-3_主開発言語3		
	3-9-4_主開発言語4		
	3-9-5_主開発言語5		
3-10_DBMSの利用		当該プロジェクトにおいて組込み向け汎用DBMSを使用したか否か。	a：Oracle b：SQL Server系 c：Empress d：SQLite e：その他DB（任意記入） f：無し

(4) 開発の進め方

データ名称	定義	回答内容、選択肢
4-1_開発ライフサイクルモデル	サイクルモデル開発ライフサイクルモデル。	a：ウォーターフォール b：反復型 c：段階的（機能追加型） d：その他（具体的名称）
4-2_類似プロジェクトの参照の有無	プロジェクト計画時に過去に実施した類似プロジェクトを参照したか否か。 ※類似プロジェクトは存在したが、参照できなかった場合は「無し」とする。 〔補足説明〕類似プロジェクトのソフトウェア資産（プログラム、各工程で作成された文書、プロジェクト管理資料、テストデータ等、開発プロジェクトによって作成されたもののすべてを含む）のいずれかを参照したか否か。	a：有り b：無し
4-3_プロジェクト管理ツールの利用	開発におけるプロジェクト管理ツールの利用の有無。	a：有り（具体的名称） b：無し
4-4_構成管理ツールの利用	開発における構成管理ツールの利用の有無。 ※構成管理ツールの例：ClearCase、CVS、Subversion、PVCS、SCCS、VSSなど。	a：有り（具体的名称） b：無し
4-5_設計支援ツールの利用	開発における設計支援ツールの利用の有無。	a：有り（具体的名称） b：無し
4-6_ドキュメント作成ツールの利用	開発におけるドキュメント作成ツールの利用の有無。 ※MS-Wordや、テキストエディタ等の一般文書作成ツールは除く。	a：有り（具体的名称） b：無し
4-7_デバッグツールの利用	開発におけるデバッグツールの利用の有無。 ※社内製ツールで具体的名称を明記できない場合は、“社内開発ツール”も可。	a：有り（具体的名称） b：無し
4-7a_テストツールの利用	テスト工程におけるテストツールの利用の有無。 ※社内製ツールで具体的名称を明記できない場合は、“社内開発ツール”も可。	a：有り（具体的名称） b：無し
4-7b_バグ管理ツールの利用	バグ修正管理のためのツールの利用の有無。 ※バグ管理ツールの例：チケット管理ツール、MS-Excelなど。	a：有り（具体的名称） b：無し
4-8_コードジェネレータの利用	コードジェネレータの利用の有無。 ※社内製ツールで具体的名称を明記できない場合は、“社内開発ツール”も可。	a：有り（具体的名称） b：無し
4-9_プロジェクト計画書再利用率	プロジェクト計画書の再利用したページ数÷全ページ数 ※手をつけたかどうかに関係なく他から持ってきたものを再利用と定める。	（　　）%

データ名称	定義	回答内容、選択肢
4-10_要求仕様書再利用率	要求仕様書の再利用したページ数÷全ページ数。 ※手をつけたかどうかに関係なく他から持ってきたものを再利用と定める。	(　　)%
4-11_アーキテクチャ設計書再利用率	アーキテクチャ設計書の再利用したページ数÷全ページ数。 ※手をつけたかどうかに関係なく他から持ってきたものを再利用と定める。	(　　)%
4-12_詳細設計書再利用率	詳細設計書の再利用したページ数÷全ページ数。 ※手をつけたかどうかに関係なく他から持ってきたものを再利用と定める。	(　　)%
4-13_ソースコード再利用率	再利用したSLOC÷全SLOC。 ※手をつけたかどうかに関係なく他から持ってきたものを再利用と定める。	(　　)%
4-14_コンポーネント再利用率	ソフトウェアコンポーネント(ライブラリ等)の再利用率(概数)。 再利用した機能規模÷システム全体の機能規模。 ※機能モジュールの数で計算。粒度は問わない。 ※手をつけたかどうかに関係なく他から持ってきたものを再利用と定める。	約(　　)%
4-15_テストケース再利用率_結合テスト	結合テストにおいて再利用したテストケース数÷全テストケース数。 ※手をつけたかどうかに関係なく他から持ってきたものを再利用と定める。	(　　)%
4-16_テストケース再利用率_総合テスト	総合テストにおいて再利用したテストケース数÷全テストケース数。 ※手をつけたかどうかに関係なく他から持ってきたものを再利用と定める。	(　　)%
4-17_テスト計画書の有無	テスト計画書の有無。 〔補足説明〕テスト計画が作成されていれば良い。テスト計画文書の形は問わない。(1冊のテスト計画書にまとめられていても良いし、テスト工程ごとにテスト計画書が作成されていても良い。また、プロジェクト計画書の中にテスト計画を含める形でも構わない。)	a：有り b：無し c：不明
4-18_テスト計画書のレビューの有無	テスト計画書のレビューの有無。 〔補足説明〕誰によるレビューかは問わない。(プロジェクト内レビュー、ユーザレビュー、第三者レビューのいずれであっても良い。)	a：有り b：無し c：不明
4-19_網羅性測定の有無	テスト工程での網羅性測定の有無。	a：有り b：無し c：不明
4-20_仕様カバレッジ	仕様に規定された事項のうち、テストを実施した割合。 〔補足説明〕「要求仕様」に着目したテストカバレッジ。要件カバレッジともいう。	a：100% b：80%以上 c：50%以上 d：50%未満 e：不明 f：未測定
4-21_機能カバレッジ	仕様を実現するために必要な機能のうち、テストを実施した割合。	a：100% b：80%以上 c：50%以上 d：50%未満 e：不明 f：未測定
4-21a_非機能カバレッジ	仕様を実現するために必要な非機能(性能、感度、精度等)のうち、テストを実施した割合。	a：100% b：80%以上 c：50%以上 d：50%未満 e：不明 f：未測定

データ名称	定義	回答内容、選択肢
4-22_コードカバレッジ（命令網羅）	コード内の全ての命令のうち、テストを実施した割合。	a：100% b：80%以上 c：50%以上 d：50%未満 e：不明 f：未測定
4-23_コードカバレッジ（分岐網羅）	コード内の全ての分岐のうち、テストを実施した割合。	a：100% b：80%以上 c：50%以上 d：50%未満 e：不明 f：未測定
4-24_コードカバレッジ（条件網羅）	コード内の全ての条件の真偽の組み合わせのうち、テストを実施した割合。	a：100% b：80%以上 c：50%以上 d：50%未満 e：不明 f：未測定

(5) ユーザ要求管理

データ名称	定義	回答内容、選択肢
5-1_要求レベル（信頼性）	システムの故障の頻度、故障状態からの回復時間・影響を受けたデータの修復などに関する、要求の厳しさ。	a：極めて高い b：高い c：中位 d：低い
5-2_要求レベル（使用性）	利用者にとってソフトウェアが理解しやすいか、適用法を習得しやすいか、運用管理しやすいか、またグラフィカル・デザインなど魅力的であるかなどに関する、要求の厳しさ。	a：極めて高い b：高い c：中位 d：低い
5-3_要求レベル（性能・効率性）	製品を利用する際の応答時間・処理時間・処理能力、及びディスク・メモリのハードウェア・その他の資源の使用量などに関する、要求の厳しさ。	a：極めて高い b：高い c：中位 d：低い
5-4_要求レベル（保守性）	製品の修理に関して、故障箇所・原因の特定のしやすさ、変更作業のしやすさ、修正の際の予期せぬ影響の防止、修正の妥当性の確認のしやすさなどに関する、要求の厳しさ。	a：極めて高い b：高い c：中位 d：低い
5-4a_要求レベル（移植性）	移植性に関する、要求の厳しさ。	a：極めて高い b：高い c：中位 d：低い
5-4b_要求レベル（セキュリティ）	セキュリティに関する、要求の厳しさ。	a：極めて高い b：高い c：中位 d：低い
5-5_法的規制の有無	法的規制の有無。	a：個別法レベルの規制あり b：一般法レベルの規制あり c：規制なし ※個別法の例．消防法、電波法
5-6_要求管理の有無	要求とのトレーサビリティをツール等で管理しているか。	a：有（ツール等の名称） b：無

(6)要員等スキル

データ名称	定義	回答内容、選択肢
6-1_要員スキル_ 製品分野の経験	開発する製品・技術分野に関するプロジェクトメンバの経験の度合い。 〔補足説明〕外部委託部分がある場合、外部委託要員を含めて評価する。	a：全員が十分な経験 b：半数が十分な経験、残り半数はいくらかの経験 c：半数がいくらかの経験、残り半数は経験なし d：全員が経験なし
6-2_要員スキル_ 分析・設計経験	プロジェクトメンバの仕様分析・設計の経験の状況。 〔補足説明〕外部委託部分がある場合、外部委託要員を含めて評価する。	a：全員が十分な経験 b：半数が十分な経験、残り半数はいくらかの経験 c：半数がいくらかの経験、残り半数は経験なし d：全員が経験なし
6-3_要員スキル_ 言語・ツール利用経験	プロジェクトメンバの言語・ツールの経験の状況。 〔補足説明〕外部委託部分がある場合、外部委託要員を含めて評価する。	a：全員が十分な経験 b：半数が十分な経験、残り半数はいくらかの経験 c：半数がいくらかの経験、残り半数は経験なし d：全員が経験なし
6-4_要員スキル_ 開発プラットフォームの使用経験	プロジェクトメンバの開発プラットフォームの使用経験の状況。 〔補足説明〕外部委託部分がある場合、外部委託要員を含めて評価する。	a：全員が十分な経験 b：半数が十分な経験、残り半数はいくらかの経験 c：半数がいくらかの経験、残り半数は経験なし d：全員が経験なし
6-5_要員スキル_ 製品プラットフォームの使用経験	プロジェクトメンバの製品プラットフォームの使用経験の状況。 〔補足説明〕外部委託部分がある場合、外部委託要員を含めて評価する。	a：全員が十分な経験 b：半数が十分な経験、残り半数はいくらかの経験 c：半数がいくらかの経験、残り半数は経験なし d：全員が経験なし

(7)システム規模

データ名称	定義	回答内容、選択肢
SLOC計画値の推移		
7-1_プロジェクト計画時	プロジェクト計画時のSLOC計画値。	(　　)SLOC
7-3_アーキテクチャ設計後	アーキテクチャ設計終了後のSLOC計画値。	(　　)SLOC
SLOC実績値 ※SLOCの単位はLine（KiloLineではない）。SLOC値にはコメント行、空行は含めない。		
7-5_SLOC実績値	総合テスト完了時のSLOC値。 ※改良(派生)開発の場合、改造せずにそのまま再利用した既存コードを除いた実績値。(7-7)の「追加・新規」と(7-8)の「変更」を合算したコード行数に相当するが、SLOC実績値の考え方を現行のエンタプライズ系「ソフトウェア開発データ白書」に合わせるため、既存製品から削除したコード行数(7-9)も含める。ただし(7-9)の「削除」が管理されている場合のみ。	(　　)SLOC ※(7-7)(7-8)(7-9)を加算したもの【自動計算】
7-6_SLOC実績値 (再利用した母体含む全体)	改造せずにそのまま再利用したコード行数を含めた全体のSLOC値を記述。	(　　)SLOC
7-7_SLOC実績値(追加・新規)	追加・新規のSLOC値を記述。	(　　)SLOC
7-8_SLOC実績値(変更)	既存コードを変更して作成したSLOC値を記述。	(　　)SLOC
7-9_SLOC実績値(削除)	既存コードから削除したSLOC値を記述。	(　　)SLOC

データ名称		定義	回答内容、選択肢
7-10_SLOC計画値（再利用した母体含む全体）		改造せずにそのまま再利用するコード行数を含めた全体のSLOC値を記述。	（　　）SLOC
7-11_SLOC計画値（追加・新規）		追加・新規のSLOC値を記述。	（　　）SLOC
7-12_SLOC計画値（変更）		既存コードを変更して作成するSLOC値を記述。	（　　）SLOC
7-13_SLOC計画値（削除）		既存コードから削除するSLOC値を記述。	（　　）SLOC
言語別SLOC実績値 ※7-5_SLOC実績値に対し、開発言語別の上位5言語について、プロジェクト内で使用言語の規模の多いものから順に記載する。			
7-14_言語別SLOC実績値	**7-14-a1_言語名称**	「3-9_主開発言語」で選択した各開発言語について、そのSLOC実績値を記述する。	• 言語名称【自動入力】 a：アセンブラ b：C c：C++ d：C# e：java f：HTML g：その他言語（任意記入） • SLOC実績値 （　　）SLOC
	7-14-a2_SLOC実績値		
	7-14-b1_言語名称		
	7-14-b2_SLOC実績値		
	7-14-c1_言語名称		
	7-14-c2_SLOC実績値		
	7-14-d1_言語名称		
	7-14-d2_SLOC実績値		
	7-14-e1_言語名称		
	7-14-e2_SLOC実績値		
	7-14f_コメント行比率	ソースコードのうちコメント行の比率。	5%, 10%, 15%, 20%, 25%, 30%, 35%, 45%, 50%, 55%, 60%, 65%, 70%, 75%, 80%, 90%, 95%
設計書の文書量（実績値）			
7-15_プロジェクト計画書		プロジェクト計画書の実測ページ数。	（　　）ページ
7-16_要求仕様書		要求仕様書の実測ページ数。	（　　）ページ
7-17_アーキテクチャ設計書		アーキテクチャ設計書の実測ページ数。	（　　）ページ
7-18_詳細設計書		詳細設計書の実測ページ数。	（　　）ページ
その他規模指標			
7-19_ユースケース数		ユースケース数。単純（7-19-1）、平均的（7-19-2）、複雑（7-19-3）の3段階で記述。	単純：（　　） 平均：（　　） 複雑：（　　）

(8) 工期

データ名称		定義	回答内容、選択肢
工程別工期 ※要求定義、アーキテクチャ設計、詳細設計、実装・単体テスト、結合テスト、総合テスト			
8-1_工程別工期（計画）	**8-1-1-1_要求定義_開始年月［日］**	各工程の開始年月［日］	yyyy/mm/dd　※ddは省略可能
	8-1-1-2_アーキテクチャ設計_開始年月［日］		yyyy/mm/dd　※ddは省略可能
	8-1-1-3_詳細設計_開始年月［日］		yyyy/mm/dd　※ddは省略可能
	8-1-1-4_実装・単体テスト_開始年月［日］		yyyy/mm/dd　※ddは省略可能
	8-1-1-5_結合テスト_開始年月［日］		yyyy/mm/dd　※ddは省略可能
	8-1-1-6_総合テスト_開始年月［日］		yyyy/mm/dd　※ddは省略可能
	8-1-2-1_要求定義_終了年月［日］	各工程の終了年月［日］	yyyy/mm/dd　※ddは省略可能
	8-1-2-2_アーキテクチャ設計_終了年月［日］		yyyy/mm/dd　※ddは省略可能
	8-1-2-3_詳細設計_終了年月［日］		yyyy/mm/dd　※ddは省略可能
	8-1-2-4_実装・単体テスト_終了年月［日］		yyyy/mm/dd　※ddは省略可能
	8-1-2-5_結合テスト_終了年月［日］		yyyy/mm/dd　※ddは省略可能
	8-1-2-6_総合テスト_終了年月［日］		yyyy/mm/dd　※ddは省略可能

	データ名称	定義	回答内容、選択肢
8-1_工程別工期(計画)	8-1-3-1_要求定義_月数	各工程の工期(月数) ※「工程別終了年月[日](計画)－工程別開始年月[日](計画)」で計算した月数	※自動計算
	8-1-3-2_アーキテクチャ設計_月数		※自動計算
	8-1-3-3_詳細設計_月数		※自動計算
	8-1-3-4_実装・単体テスト_月数		※自動計算
	8-1-3-5_結合テスト_月数		※自動計算
	8-1-3-6_総合テスト_月数		※自動計算
8-2_工程別工期(実績)	8-2-1-1_要求定義_開始年月[日]	各工程の開始年月[日]	yyyy/mm/dd　※ddは省略可能
	8-2-1-2_アーキテクチャ設計_開始年月[日]		yyyy/mm/dd　※ddは省略可能
	8-2-1-3_詳細設計_開始年月[日]		yyyy/mm/dd　※ddは省略可能
	8-2-1-4_実装・単体テスト_開始年月[日]		yyyy/mm/dd　※ddは省略可能
	8-2-1-5_結合テスト_開始年月[日]		yyyy/mm/dd　※ddは省略可能
	8-2-1-6_総合テスト_開始年月[日]		yyyy/mm/dd　※ddは省略可能
	8-2-2-1_要求定義_終了年月[日]	各工程の終了年月[日]	yyyy/mm/dd　※ddは省略可能
	8-2-2-2_アーキテクチャ設計_終了年月[日]		yyyy/mm/dd　※ddは省略可能
	8-2-2-3_詳細設計_終了年月[日]		yyyy/mm/dd　※ddは省略可能
	8-2-2-4_実装・単体テスト_終了年月[日]		yyyy/mm/dd　※ddは省略可能
	8-2-2-5_結合テスト_終了年月[日]		yyyy/mm/dd　※ddは省略可能
	8-2-2-6_総合テスト_終了年月[日]		yyyy/mm/dd　※ddは省略可能
	8-2-3-1_要求定義_月数	各工程の工期(月数) ※「工程別終了年月[日](実績)－工程別開始年月[日](実績)」で計算した月数	※自動計算
	8-2-3-2_アーキテクチャ設計_月数		※自動計算
	8-2-3-3_詳細設計_月数		※自動計算
	8-2-3-4_実装・単体テスト_月数		※自動計算
	8-2-3-5_結合テスト_月数		※自動計算
	8-2-3-6_総合テスト_月数		※自動計算
プロジェクト全体工期			
8-3_プロジェクト全体工期(計画)	8-3-1_開始年月[日]	プロジェクト開始年月[日](計画) 開始日＝工数が発生する日	yyyy/mm/dd　※ddは省略可能
	8-3-2_終了年月[日]	プロジェクト終了年月[日](計画) 終了日＝工数が発生する最後の日	yyyy/mm/dd　※ddは省略可能
	8-3-3_月数	「プロジェクト終了年月[日](計画)－プロジェクト開始年月[日](計画)」で計算した月数。	※自動計算
8-4_プロジェクト全体工期(実績)	8-4-1_開始年月[日]	プロジェクト開始年月[日](実績) 開始日＝工数が発生する日	yyyy/mm/dd　※ddは省略可能
	8-4-2_終了年月[日]	プロジェクト終了年月[日](実績) 終了日＝工数が発生する最後の日	yyyy/mm/dd　※ddは省略可能
	8-4-3_月数	「プロジェクト終了年月[日](実績)－プロジェクト開始年月[日](実績)」で計算した月数。	※自動計算

(9)工数(コスト)

データ名称	定義	回答内容、選択肢
9-1_工数の単位	工数の単位を人時、人月から選択する。 ※ここでの選択が、「9-4_社内実績工数」及び「9-8_外部委託工数」の単位となる。	a：人時、b：人月
9-2_人時換算係数	「人月」を「人時」にする場合の換算係数。 例．1人月＝160人時	1人月＝(　　)人時

データ名称		定義	回答内容、選択肢
9-3_プロジェクト総工数に含まれるフェーズ	9-3-1_要求定義	開発プロジェクトに「要求定義」～「総合テスト」までの各フェーズが含まれているか否か。該当フェーズに相当する作業の有無を記述。 【回答は次の定義から選択】 ○：作業があり、工数等のデータをこのフェーズの欄に記入する場合 ×：作業が無い場合 ⇒：作業があるが、当該フェーズに相当する作業工数は、他フェーズの欄に合算して記入する場合（後ろの「○」に含む）。ただし、当該工程が後ろの「○」工程に含まれず、工程配分不可に含む場合は工数を「0」とする。 ※全て工程配分不可の場合、対象工程を「⇒」と記入 複数フェーズの作業をまとめて1フェーズとして管理する場合や、データが合計でのみ把握できる場合、まとめた工数データは後工程の欄に両方の作業の合計工数を記録する。 例．アーキテクチャ設計・詳細設計・実装・単体テストのデータを合計で記入する場合は、アーキテクチャ設計は「⇒」、詳細設計は「⇒」、実装・単体テストに「○」を記入する。	○ × ⇒
	9-3-2_アーキテクチャ設計		○ × ⇒
	9-3-3_詳細設計		○ × ⇒
	9-3-4_実装・単体テスト		○ × ⇒
	9-3-5_結合テスト		○ × ⇒
	9-3-6_総合テスト		○ ×
9-4_社内実績工数	9-4-1_要求定義_開発	社員（社員と一緒に作業する派遣社員を含む）の工程別開発工数。	（　　）人時、又は人月
	9-4-2_アーキテクチャ設計_開発		（　　）人時、又は人月
	9-4-3_詳細設計_開発		（　　）人時、又は人月
	9-4-4_実装・単体テスト_開発		（　　）人時、又は人月
	9-4-5_結合テスト_開発		（　　）人時、又は人月
	9-4-6_総合テスト_開発		（　　）人時、又は人月
	9-4-7_プロジェクト全体_開発	開発工数（9-4-1 ～ 9-4-6）の合計。	※自動計算
	9-4-8_要求定義_管理	社員（社員と一緒に作業する派遣社員を含む）の工程別管理工数。	（　　）人時、又は人月
	9-4-9_アーキテクチャ設計_管理		（　　）人時、又は人月
	9-4-10_詳細設計_管理		（　　）人時、又は人月
	9-4-11_実装・単体テスト_管理		（　　）人時、又は人月
	9-4-12_結合テスト_管理		（　　）人時、又は人月
	9-4-13_総合テスト_管理		（　　）人時、又は人月
	9-4-14_プロジェクト全体_管理	管理工数（9-4-8 ～ 9-4-13）の合計。	※自動計算
	9-4-15_要求定義_その他	社員（社員と一緒に作業する派遣社員を含む）の工程別その他工数。 ※その他：開発、管理に分類されない工数。	（　　）人時、又は人月
	9-4-16_アーキテクチャ設計_その他		（　　）人時、又は人月
	9-4-17_詳細設計_その他		（　　）人時、又は人月
	9-4-18_実装・単体テスト_その他		（　　）人時、又は人月
	9-4-19_結合テスト_その他		（　　）人時、又は人月
	9-4-20_総合テスト_その他		（　　）人時、又は人月
	9-4-21_プロジェクト全体_その他	その他工数（9-4-15 ～ 9-4-20）の合計。	※自動計算
	9-4-22_要求定義_作業配分不可	社員（社員と一緒に作業する派遣社員を含む）の工程別作業配分不可工数。 ※作業配分不可：開発、管理、その他に分類されない工数 例．テスト環境構築、運用構築、移行、業務支援、コンサルティングなど	（　　）人時、又は人月
	9-4-23_アーキテクチャ設計_作業配分不可		（　　）人時、又は人月
	9-4-24_詳細設計_作業配分不可		（　　）人時、又は人月
	9-4-25_実装・単体テスト_作業配分不可		（　　）人時、又は人月
	9-4-26_結合テスト_作業配分不可		（　　）人時、又は人月
	9-4-27_総合テスト_作業配分不可		（　　）人時、又は人月

データ名称		定義	回答内容、選択肢
9-4_社内実績工数	9-4-28_プロジェクト全体_作業配分不可合計	作業配分不可工数(9-4-22 ～ 9-4-27)の合計。	※自動計算
	9-4-29_要求定義	要求定義の実績工数。 ※開発、管理、その他、作業配分不可工数(9-4-1, 9-4-8, 9-4-15, 9-4-22)の合計。	※自動計算
	9-4-30_アーキテクチャ設計	アーキテクチャ設計の実績工数。 ※開発、管理、その他、作業配分不可工数(9-4-2, 9-4-9, 9-4-16, 9-4-23)の合計。	※自動計算
	9-4-31_詳細設計	詳細設計の実績工数。 ※開発、管理、その他、作業配分不可工数(9-4-3, 9-4-10, 9-4-17, 9-4-24)の合計。	※自動計算
	9-4-32_実装・単体テスト	実装・単体テストの実績工数。 ※開発、管理、その他、作業配分不可工数(9-4-4, 9-4-11, 9-4-18, 9-4-25)の合計。	※自動計算
	9-4-33_結合テスト	結合テストの実績工数 ※開発、管理、その他、作業配分不可工数(9-4-5, 9-4-12, 9-4-19, 9-4-26)の合計	※自動計算
	9-4-34_総合テスト	総合テストの実績工数。 ※開発、管理、その他、作業配分不可工数(9-4-6, 9-4-13, 9-4-20, 9-4-27)の合計。	※自動計算
	9-4-35_プロジェクト全体	プロジェクト全体の実績工数。 ※各工程実績工数(9-4-29 ～ 9-4-34)の合計。	※自動計算
9-5_レビュー実績工数	9-5-1_要求定義	社内の工程別レビュー実績工数(社内工数の内数)。	(　　)人時
	9-5-2_アーキテクチャ設計		(　　)人時
	9-5-3_詳細設計		(　　)人時
	9-5-4_ソースコード		(　　)人時
	9-5-5_結合テスト		(　　)人時
	9-5-6_総合テスト		(　　)人時
	9-5-7_プロジェクト全体	工程別レビュー実績工数(9-5-1 ～ 9-5-6)の合計。	※自動計算
9-6_レビュー実績回数	9-6-1_要求定義	工程別レビュー回数。	(　　)回数
	9-6-2_アーキテクチャ設計		(　　)回数
	9-6-3_詳細設計		(　　)回数
	9-6-4_ソースコード		(　　)回数
	9-6-5_結合テスト		(　　)回数
	9-6-6_総合テスト		(　　)回数
	9-6-7_プロジェクト全体	工程別レビュー実績回数(9-6-1 ～ 9-6-6)の合計。	※自動計算
9-7_レビュー指摘件数	9-7-1_要求定義	工程別レビュー指摘数。	(　　)件数
	9-7-2_アーキテクチャ設計		(　　)件数
	9-7-3_詳細設計		(　　)件数
	9-7-4_ソースコード		(　　)件数
	9-7-5_結合テスト		(　　)件数
	9-7-6_総合テスト		(　　)件数
	9-7-7_プロジェクト全体	工程別レビュー指摘数(9-7-1 ～ 9-7-6)の合計。	※自動計算
9-8_外部委託工数	9-8-1_要求定義	外部委託の工程別開発工数(社内工数の外数)。 ※スパイラル開発の場合(「4-1：開発ライフサイクルモデル」でb:反復型、c:段階的を選択した場合)、例えば「詳細設計」、「実装・単体テスト」、「結合テスト」の3つの工程が複数回繰り返される場合は、各工程の工数を合算して各工程欄に記入する。但し、工程ごとに細かく工数が管理されている場合のみ。	(　　)人時、又は人月
	9-8-2_アーキテクチャ設計		(　　)人時、又は人月
	9-8-3_詳細設計		(　　)人時、又は人月
	9-8-4_実装・単体テスト		(　　)人時、又は人月
	9-8-5_結合テスト		(　　)人時、又は人月
	9-8-6_総合テスト		(　　)人時、又は人月

データ名称		定義	回答内容、選択肢
9-8_外部委託工数	9-8-7_プロジェクト全体	外部委託の工程別開発工数(9-8-1 ～ 9-8-6)の合計。	※自動計算
9-9_外部委託作業有無	9-9-1_要求定義	開発作業の外部委託の有無。	○、空白
	9-9-2_アーキテクチャ設計		○、空白
	9-9-3_詳細設計		○、空白
	9-9-4_実装・単体テスト		○、空白
	9-9-5_結合テスト		○、空白
	9-9-6_総合テスト		○、空白
9-10_外部委託金額比率	9-10-1_要求定義	外部委託工数が不明の場合に、全体金額に対する外部委託金額比率を工程別に記述。	(　　)%
	9-10-2_アーキテクチャ設計		(　　)%
	9-10-3_詳細設計		(　　)%
	9-10-4_実装・単体テスト		(　　)%
	9-10-5_結合テスト		(　　)%
	9-10-6_総合テスト		(　　)%
9-11_社内平均要員数	9-11-1_要求定義	社内の工程別平均要員数。	(　　)人
	9-11-2_アーキテクチャ設計		(　　)人
	9-11-3_詳細設計		(　　)人
	9-11-4_実装・単体テスト		(　　)人
	9-11-5_結合テスト		(　　)人
	9-11-6_総合テスト		(　　)人
9-12_社内ピーク要員数	9-12-1_要求定義	社内の工程別ピーク要員数。	(　　)人
	9-12-2_アーキテクチャ設計		(　　)人
	9-12-3_詳細設計		(　　)人
	9-12-4_実装・単体テスト		(　　)人
	9-12-5_結合テスト		(　　)人
	9-12-6_総合テスト		(　　)人
9-13_外部委託平均要員数	9-13-1_要求定義	外部委託の工程別平均要員数。	(　　)人
	9-13-2_アーキテクチャ設計		(　　)人
	9-13-3_詳細設計		(　　)人
	9-13-4_実装・単体テスト		(　　)人
	9-13-5_結合テスト		(　　)人
	9-13-6_総合テスト		(　　)人
9-14_外部委託ピーク要員数	9-14-1_要求定義	外部委託の工程別ピーク要員数。	(　　)人
	9-14-2_アーキテクチャ設計		(　　)人
	9-14-3_詳細設計		(　　)人
	9-14-4_実装・単体テスト		(　　)人
	9-14-5_結合テスト		(　　)人
	9-14-6_総合テスト		(　　)人
9-15_プロジェクト開発工数計画値（アーキテクチャ設計終了時点）		プロジェクト全体の開発工数(社内および外部委託)のアーキテクチャ設計終了時点の計画値。	(　　)

(10)品質

データ名称		定義	回答内容、選択肢
テストフェーズ別テストケース数			
10-1a_単体テストケース	10-1a-1_単体テストケース数	単体テストケース数	(　　)件
	10-1a-2_単体テストケース数定義	単体テストケース(数)の定義について補足(任意回答)	(　　)
10-1_結合テストケース	10-1-1_結合テストケース数	結合テストケース数	(　　)件
	10-1-2_結合テストケース数定義	結合テストケース(数)の定義について補足(任意回答)	(　　)
10-2_総合テストケース	10-2-1_総合テストケース数	総合テストケース数	(　　)件
	10-2-2_総合テストケース数定義	総合テストケース(数)の定義について補足(任意回答)	(　　)
テストフェーズ別検出バグ数			
10-3a_単体テスト検出バグ	10-3a-1_単体テスト検出バグ現象数	単体テスト検出バグ現象数	(　　)件
	10-3a-2_単体テスト検出バグ原因数	単体テスト検出バグ原因数	(　　)件
	10-3a-3_単体テスト検出バグ数定義	単体テスト検出バグ(数)の定義について補足(任意回答)	(　　)
10-3_結合テスト検出バグ	10-3-1_結合テスト検出バグ現象数	結合テスト検出バグ現象数	(　　)件
	10-3-2_結合テスト検出バグ原因数	結合テスト検出バグ原因数	(　　)件
	10-3-3_結合テスト検出バグ数定義	結合テスト検出バグ(数)の定義について補足(任意回答)	(　　)
10-4_総合テスト検出バグ	10-4-1_総合テスト検出バグ現象数	総合テスト検出バグ現象数	(　　)件
	10-4-2_総合テスト検出バグ原因数	総合テスト検出バグ原因数	(　　)件
	10-4-3_総合テスト検出バグ数定義	総合テスト検出バグ(数)の定義について補足(任意回答)	(　　)
10-5a_リリース後検出バグ	10-5a-1-1_検出バグ現象数_1ヶ月	リリース後(ソフトウェアの総合テスト完了後、量産工場、顧客等に引き渡し完了後)に報告された検出バグの総数。現象数と原因数に分ける。 それぞれの数は一定期間経過時点の累計で表す。つまり1ヶ月経過時点の合計値、3ヶ月経過時点の累計値、6ヶ月経過時点の累計値で表す。 ※1 例として、稼動後5ヶ月しか経過していない場合は、1ヶ月、3ヶ月の値のみ記入する。 ※2 総合テストが、引き渡し先で実施される場合や、リリース後検出バグが不明な場合は、記入しない。 〔補足説明〕 ・リリース後に引き渡し先で検出されたバグと自ら検出したバグの両方を対象とするが、非対処バグは含まない。 ・1ヶ月経過時点の件数≦3ヶ月経過時点の累計件数≦6ヶ月経過時点の累計件数でなければならない。	(　　)件
	10-5a-1-2_検出バグ現象数_3ヶ月		(　　)件
	10-5a-1-3_検出バグ現象数_6ヶ月		(　　)件
	10-5a-2-1_検出バグ原因数_1ヶ月		(　　)件
	10-5a-2-2_検出バグ原因数_3ヶ月		(　　)件
	10-5a-2-3_検出バグ原因数_6ヶ月		(　　)件

データ名称	定義	回答内容、選択肢
10-5_品質保証体制	開発中のソフトウェアまたは品質保証の体制。	a：プロジェクトメンバが実施 b：品質保証の専任スタッフも実施 c：実施していない
10-6_テスト体制	テストを実施する体制。 〔補足説明〕プロジェクトメンバの他に、第三者による評価テスト実施する体制か否か。さらに、スキルや員数が十分であるか。 ※専任のテスト部隊も含む。	a：プロジェクトメンバが実施。スキル、員数ともに十分 b：プロジェクトメンバが実施。スキルは十分、員数は不足 c：プロジェクトメンバが実施。スキルは不足、員数は十分 d：プロジェクトメンバが実施。スキル、員数ともに不足 e：第三者による評価テストも実施。スキル、員数ともに十分 f：第三者による評価テストも実施。スキルは十分、員数は不足 g：第三者による評価テストも実施。スキルは不足、員数は十分 h：第三者による評価テストも実施。スキル、員数ともに不足
10-7_定量的な出荷品質基準の有無	対象プロジェクトにおいて定量的な出荷品質基準が設定されていたか否か。	a：有り（具体的に記述） b：無し
10-8_第三者レビューの有無	第三者レビューを実施しているか否か。 ※第三者：プロジェクトに関係しない人員。例．品質保証部門、PMO。	a：有り b：無し
10-9_構成管理体制	構成管理の体制（成果物のバージョンを管理してビルドやリリースを行う作業体制）	a：プロジェクトメンバが実施 b：構成管理の専任スタッフ（ライブラリアン）が実施 c：実施していない

A.3　導出指標の名称と定義

付録A.2のデータ項目を組み合わせて定義した項目を以下に示す。
※「導出指標」は、JIS X0141：2004　ソフトウェア測定プロセスでは「導出測定量」と呼ばれている。

分類	名称	定義
規　模	実効SLOC実績値	原則、コメント行、空行を除いたSLOC値。 改良(派生)開発の場合は、改造せずに再利用した母体の部分を含まない。 7-5_SLOC実績値(各社提出値)を使用。 7-5_SLOC実績値=7-7_SLOC実績値(追加・新規)+ 7-8_SLOC実績値(変更)+ 7-9_SLOC実績値(削除) KSLOCは実効SLOC実績値を1,000行単位で表現したもの。
工　期	実績月数 (プロジェクト全体)	8-4-3_プロジェクト全体工期(実績)_月数のデータ。 ただし、8-4-3_プロジェクト全体工期(実績)_月数がない場合は、全工程別工期(8-2-3_月数)を合計した値を使用。
	実績月数 (開発5工程)	アーキテクチャ設計～総合テストの各工程別工期(8-2-3_月数)を合計した値。 8-2-3-2_アーキテクチャ設計_月数+8-2-3-3_詳細設計_月数+ 8-2-3-4_実装・単体テスト_月数+8-2-3-5_結合テスト_月数+ 8-2-3-6_総合テスト_月数 ただし、各工程が重複している期間は二重カウントしない。 開発5工程がすべて実施されたプロジェクトのみを対象に算出。
	実績月数 (開発6工程)	要求定義～総合テストの各工程別工期(8-2-3_月数)を合計した値。 2-3-1_要求定義_月数+8-2-3-2_アーキテクチャ設計_月数+ 8-2-3-3_詳細設計_月数+8-2-3-4_実装・単体テスト_月数+ 8-2-3-5_結合テスト_月数+8-2-3-6_総合テスト_月数 ただし、各工程が重複している期間は二重カウントしない。 開発6工程がすべて実施されたプロジェクトのみを対象に算出。
工　数	実績工数 (開発5工程)	アーキテクチャ設計～総合テストの各工程の工数を合計した値(単位は人時)。 9-4-30_アーキテクチャ設計+9-4-31_詳細設計+ 9-4-32_実装・単体テスト+9-4-33_結合テスト+ 9-4-34_総合テスト 詳細は本表後ろの※1を参照。 開発5工程がすべて実施されたプロジェクトのみを対象に算出。 なお、工数には社員工数(開発工数、管理工数、その他工数、作業配分不可工数)と外部委託工数を含む。
	実績工数 (開発6工程)	要求定義～総合テストの各工程の工数を合計した値(単位は人時)。 9-4-29_要求定義+9-4-30_アーキテクチャ設計+ 9-4-31_詳細設計+9-4-32_実装・単体テスト+ 9-4-33_結合テスト+9-4-34_総合テスト 詳細は本表後ろの※1を参照。 開発6工程がすべて実施されたプロジェクトのみを対象に算出。 なお、工数には社員工数(開発工数、管理工数、その他工数、作業配分不可工数)と外部委託工数を含む。
	実績工数 (プロジェクト全体)	実施した工程にかかわらず、プロジェクト全体の工数(単位は人時)。 なお、工数には社員工数(開発工数、管理工数、その他工数、作業配分不可工数)と外部委託工数を含む。 社内実績工数+外部委託工数で算出。 9-4-35_社内実績工数_プロジェクト全体+ 9-8-7_外部委託工数_プロジェクト全体
	外部委託比率	外部委託工数比率(次項を参照)のデータ。ただし、外部委託工数比率が算出できない場合は、9-10_外部委託金額比率のデータを使用。
	外部委託工数比率	実施した工程にかかわらずプロジェクト全体の外部委託工数合計値を、実績工数(プロジェクト全体)で割った値。 9-8-7_外部委託工数_プロジェクト全体÷実績工数(プロジェクト全体)

分類	名称	定義
生産性	SLOC生産性	人時あたりのSLOC数。 実効SLOC実績値÷実績工数（開発5工程）で算出。
	アーキテクチャ設計 SLOC生産性	人時あたりのSLOC数。 実効SLOC実績値÷実績工数（アーキテクチャ設計）で算出。
	詳細設計SLOC生産性	人時あたりのSLOC数。 実効SLOC実績値÷実績工数（詳細設計）で算出。
	実装･単体テストSLOC生産性	人時あたりのSLOC数。 実効SLOC実績値÷実績工数（実装･単体テスト）で算出。
	結合テストSLOC生産性	人時あたりのSLOC数。 実効SLOC実績値÷実績工数（結合テスト）で算出。
	総合テストSLOC生産性	人時あたりのSLOC数。 実効SLOC実績値÷実績工数（総合テスト）で算出。
信頼性	テスト検出バグ密度	KSLOCあたりのテスト検出バグ数。 検出バグ数÷実効SLOC実績値×1,000で算出。 検出バグ数は、結合テスト、総合テストをそれぞれ利用。 結合テスト検出バグ現象密度 = 10-3-1_結合テスト検出バグ現象数÷7-5_SLOC実績値 結合テスト検出バグ原因密度 = 10-3-2_結合テスト検出バグ原因数÷7-5_SLOC実績値 総合テスト検出バグ現象密度 = 10-4-1_総合テスト検出バグ現象数÷7-5_SLOC実績値 総合テスト検出バグ原因密度 = 10-4-2_総合テスト検出バグ原因数÷7-5_SLOC実績値 総合テスト検出バグ現象密度（母体含む）= 10-4-1_総合テスト検出バグ現象数÷7-6_SLOC実績値（母体） 総合テスト検出バグ原因密度（母体含む）= 10-4-2_総合テスト検出バグ原因数÷7-6_SLOC実績値（母体）
体　制	月あたりの要員数	実績工数（プロジェクト全体）÷実績月数（プロジェクト全体）÷人時換算係数で算出。 人時換算係数は、9-1_工数の単位が「b：人月」ならば9-2_人時換算係数_人時／人月を使用。

※1 実績工数(開発５工程、開発６工程)の図解

開発5工程：アーキテクチャ設計～総合テストの5工程がすべて実施されたプロジェクトに対して、下表の橙色の枠内のセルの工数を合算し、さらに人時へ換算した値を範囲とする。

開発6工程：要求定義～総合テストの6工程がすべて実施されたプロジェクトに対して、下表の緑色の枠内のセルの工数を合算し、さらに人時へ換算した値を範囲とする。

			← 開発5工程 →				
		← 開発6工程 →					
工数内訳		要求定義	アーキテクチャ設計	詳細設計	実装・単体テスト	結合テスト	総合テスト
社内実績工数	開発						
	管理						
	その他						
	作業配分不可						
外部委託工数	開発工数						

付録B 用語集

本書の分析で使われている用語について概要を記す。統計用語については『統計科学事典』(朝倉書店)などを参考にした。

●機能規模

(ソフトウェア測定-機能規模測定-JIS X0135-1:1999からの引用に基づく)
利用者機能要件を定量化して得られるソフトウェアの規模。利用者機能要件とは、利用者要件の部分集合であり、利用者の要求を満足するためにソフトウェアが実現しなければならない利用者の業務及び手順を表す。品質要件及び技術要件は除く。

●極値

箱ひげ図において、箱の上端または下端から、箱の長さの3倍超をもつケース。箱の長さは4分位範囲。

●四分位点(25、50、75のパーセンタイル)

確率分布または頻度分布を4等分する3個の値。小さい方から第1、第2及び第3四分位点と呼ぶ。第2四分位点は中央値である。

●正規分布(正規分布曲線)

平均を中心に常に左右対称となる分布形態。曲線は平均値で最も高くなり、左右に広がるにつれて低くなる。標準偏差の値が大きいほど曲線は扁平になり、小さいほど狭く高くなる。

●(単)相関係数

二つの変数xとyについて、両者の間に直線的な関連性が認められるとき、xとyの間には相関関係があるといい、相関関係の程度を示す数値を単相関係数という。単相関係数は−1から+1までの値をとる。単相関係数が−1もしくは+1に近いときは二つの変数の関係は直線的で、−1もしくは+1から遠ざかるに従って直線関係は薄れていき、0に近いときは変数の間にまったく直線的な関係はない。

●中央値(50パーセンタイル)

与えられたデータを大きさの順に並べたときに、大きいグループと小さいグループに同数ずつに2分する位置にあるデータの値をいう。データが偶数個の場合は中間に位置する2点、すなわち小さいグループの最大値と大きいグループの最小値の平均をもって中央値とする。特に非対称分布の場合に分布の位置を表すのに適したものである。また、外れ値の影響を受けることが少ない。

●箱ひげ図

中央値、4分位、外れ値に基づく要約図。箱は4分位数間の範囲であり、したがって箱にはデータの値の50%が含まれる。各箱から出る線(ひげ)は外れ値を除いたときの最大値、または最小値に向かって延びる。箱の中の横線は中央値を示している。詳細は3.3節を参照していただきたい。

●外れ値

箱ひげ図において、箱の上端または下端から箱の長さの1.5倍から3倍の間にある値をもつケース。箱の長さは4分位範囲。

●ヒストグラム

度数あるいは相対度数を縦軸に、階級値を横軸にとり、度数分布を棒グラフにしたもの。

●標準誤差

ある統計量Tの標本分布の標準偏差をTの標準誤差という。例えば、分散がσ^2に等しい分布から標本X_1、X_2、・・・、X_nから作られる標準平均（$X_1+X_2+\cdots+X_n$）÷nの標準誤差は$\sigma\div\sqrt{n}$である。ただし、標準誤差を標準偏差と同じ意味で（すなわち分散の平方根）使うこともある。

●標準偏差

分散の平方根（データのばらつきを表す）。

●分散

分布Fからの標本X1、X2、・・・、Xnについての偏差平方和（個々のデータから平均値を引いた値の2乗の合計）をデータ数で割った値。

●平均値（算術平均）

データを足し合わせ、データ数で割った値。

●SLOC規模、KSLOC

コード行数（SourceLines Of Code）の単位で表す規模はSLOC規模と呼び、1,000行の単位で表すものをKSLOCと表記する。

●25パーセンタイル

観測値の75%がその値以上であり、観測値の25%がその値以下に入る境界の値。

●75パーセンタイル

変数の観測値の25%がその値以上であり、観測値の75%がその値以下に入る境界の値。

付録C 参考文献・参考情報

C.1では、本書で参考とした文献、定義などの掲載されている書籍、関連する標準情報を示す。
C.2では、分析にあたり使用したソフトウェアについて示す。

C.1 参考文献

[1] IPAソフトウェア高信頼化センター，“組込みソフトウェア開発データ白書2017”，独立行政法人情報処理推進機構，2017年

[2] IPA社会基盤センター，“ソフトウェア開発データ白書2018-2019”，独立行政法人情報処理推進機構，2018年

[3] IPAソフトウェア・エンジニアリング・センター，“組込みソフトウェア開発向け 品質作り込みガイド（ESQR）Version1.1”，独立行政法人情報処理推進機構（IPA），2012年

[4] IPAソフトウェア・エンジニアリング・センター，“組込みソフトウェア向け開発 プロセスガイド（ESPR）Version2.0”，独立行政法人情報処理推進機構（IPA），2007年

C.2 参考情報

◆ソフトウェア

本書のデータ分析では次のソフトウェアを利用した。

- Microsoft® Excel 2016

  円グラフ、棒グラフ、ヒストグラム、箱ひげ図、基本統計量、相関係数、パーセンタイルに利用。

図表一覧

図表1.7-1　プロジェクトの開始年ごとの収録年度別データ件数 …… 13
図表1.7-2　プロジェクトの終了年ごとの収録年度別データ件数 …… 13
図表1.7-3　収録年度別のプロジェクト開始年及び終了年のクロス集計 …… 13
図表2.2-1　実績の評価（QCD別） …… 17
図表2.2-2　品質実績の評価別の工程別の実績工数の比率 …… 18
図表2.2-3　品質実績の評価別の工程別レビュー工数密度 …… 19
図表2.2-4　品質実績の評価別の工程別レビュー指摘密度 …… 19
図表2.2-5　品質実績の評価別のSLOC規模あたりのテストケース数 …… 20
図表2.2-6　品質実績の評価別のSLOC規模あたりの検出バグ数 …… 20
図表3.1-1　代表的な要素と、要素間の主な関係 …… 24
図表3.2-1　外れ値の例 …… 27
図表3.3-1　単位の表記 …… 28
図表3.3-2　基本統計量の表 …… 29
図表3.3-3　基本統計量を使用した場合の判断の目安 …… 29
図表3.3-4　相関関係を確認した場合の判断の目安 …… 30
図表3.3-5　通常スケールを対数変換したときの分布 …… 30
図表3.3-6　箱ひげ図のサンプル …… 31
図表4.2-1　開発プロジェクトの種別 …… 37
図表4.2-2　開発システムの経過年数 …… 37
図表4.2-3　開発システムの保守年数 …… 37
図表4.2-4　開発プロジェクトの概要 …… 38
図表4.2-5　新技術を利用する開発か否か …… 38
図表4.3-1　利用形態 …… 39
図表4.3-2　障害リスク …… 39
図表4.4-1　製品の特性：リアルタイム性（時間制約） …… 40
図表4.4-2　製品の特性：自然環境からの影響度合い …… 40
図表4.4-3　製品の特性：ユーザの多様性 …… 40
図表4.4-4　製品の特性：法規等による規制度合い …… 40
図表4.4-5　製品の特性：M2Mの有無 …… 41
図表4.4-6　製品の特性：ネットワーク接続の有無 …… 41
図表4.4-7　製品の特性：稼動（非停止、オンデマンド） …… 41
図表4.4-8　製品の特性：オンライン保守の可否 …… 41
図表4.4-9　市販パッケージ利用の有無 …… 41
図表4.4-10　CPUアーキテクチャ …… 42
図表4.4-11　OSアーキテクチャ …… 42
図表4.4-12　開発対象製品のソフトウェアプラットフォーム …… 42
図表4.4-13　開発言語 …… 43
図表4.5-1　開発ライフサイクルモデル …… 44
図表4.5-2　類似プロジェクトの参照の有無 …… 44
図表4.5-3　ツールの利用有無 …… 44
図表4.5-4　ツールの利用有無一覧 …… 44
図表4.6-1　要求レベル …… 45
図表4.6-2　要求レベル一覧 …… 45
図表4.7-1　要員の経験 …… 46
図表4.7-2　要員の経験一覧 …… 46
図表4.8-1　SLOC（コード行数）実績値（全体、50KSLOC刻み） …… 47
図表4.8-2　SLOC（コード行数）実績値（100KSLOC以下、5KSLOC刻み） …… 47
図表4.8-3　SLOC（コード行数）実績値の基本統計量 …… 47
図表4.8-4　SLOC（母体含むコード行数）実績値（全体、50KSLOC刻み） …… 47
図表4.8-5　SLOC（母体含むコード行数）実績値（100KSLOC以下、5KSLOC刻み） …… 47

図表4.8-6　SLOC（母体含むコード行数）実績値の基本統計量 …… 47
図表4.9-1　プロジェクト全体の開発期間 …… 48
図表4.9-2　プロジェクト全体の開発期間の基本統計量 …… 48
図表4.9-3　開発６工程の開発期間 …… 49
図表4.9-4　開発６工程の開発期間の基本統計量 …… 49
図表4.9-5　開発5工程（要求定義を除く）の開発期間 …… 49
図表4.9-6　開発5工程（要求定義を除く）の開発期間の基本統計量 …… 49
図表4.10-1　プロジェクト全体の工数の実績値（人時換算）（全体、1,000人時刻み） …… 50
図表4.10-2　プロジェクト全体の工数の実績値（人時換算）（4,000人時以下、200人時刻み） …… 50
図表4.10-3　プロジェクト全体の開発期間の基本統計量 …… 50
図表4.10-4　プロジェクト全体の工数の実績値（人月換算）（全体、10人月刻み） …… 51
図表4.10-5　プロジェクト全体の工数の実績値（人月換算）（40人月以下、2人月刻み） …… 51
図表4.10-6　プロジェクト全体の工数の実績値の基本統計量（人月換算） …… 51
図表4.10-7　人月－人時換算係数 …… 51
図表4.10-8　人月－人時換算係数の基本統計量 …… 51
図表4.11-1　外部委託工数比率 …… 52
図表4.11-2　外部委託工数比率の基本統計量 …… 52
図表4.11-3　月あたりの要員数 …… 52
図表4.11-4　月あたりの要員数の基本統計量 …… 52
図表4.12-1　結合テスト検出バグ現象密度 …… 54
図表4.12-2　結合テスト検出バグ原因密度 …… 54
図表4.12-3　結合テスト検出バグ現象密度と原因密度の基本統計量 …… 54
図表4.12-4　結合テスト検出バグ現象密度と原因密度の比較（現象密度） …… 54
図表4.12-5　結合テスト検出バグ現象密度と原因密度の比較（原因密度） …… 54
図表4.12-6　結合テスト検出バグ現象密度と原因密度の比較の基本統計量 …… 54
図表4.12-7　総合テスト検出バグ現象密度 …… 55
図表4.12-8　総合テスト検出バグ原因密度 …… 55
図表4.12-9　総合テスト検出バグ現象密度と原因密度の基本統計量 …… 55
図表4.12-10　総合テスト検出バグ現象密度と原因密度の比較（現象密度） …… 55
図表4.12-11　総合テスト検出バグ現象密度と原因密度の比較（原因密度） …… 55
図表4.12-12　総合テスト検出バグ現象密度と原因密度の比較の基本統計量 …… 55
図表4.12-13　総合テスト検出バグ現象密度（母体含む） …… 56
図表4.12-14　総合テスト検出バグ原因密度（母体含む） …… 56
図表4.12-15　総合テスト検出バグ現象密度と原因密度（母体含む）の基本統計量 …… 56
図表4.12-16　総合テスト検出バグ現象密度と原因密度（母体含む）の比較（現象密度） …… 56
図表4.12-17　総合テスト検出バグ現象密度と原因密度（母体含む）の比較（原因密度） …… 56
図表4.12-18　総合テスト検出バグ現象密度と原因密度（母体含む）の比較の基本統計量 …… 56
図表4.12-19　品質保証の体制 …… 57
図表4.12-20　品質基準、レビューの有無 …… 57
図表4.12-21　品質基準、レビューの有無一覧 …… 57
図表4.12-22　テスト計画書の有無 …… 57
図表4.12-23　テスト計画書のレビューの有無 …… 57
図表4.13-1　実施工程の組み合わせパターン …… 58
図表4.13-2　実施工程の組み合わせパターンの比率 …… 58
図表4.14-1　実績の評価（QCD別） …… 59
図表4.14-2　実績の評価（QCD別）一覧 …… 59
図表4.14-3　実績の評価（QCD組み合わせパターン別） …… 59
図表4.14-4　計画の評価と実績の評価（コスト） …… 60
図表4.14-5　計画の評価と実績の評価（品質） …… 60
図表4.14-6　計画の評価と実績の評価（工期） …… 60
図表4.14-7　計画の評価と実績の評価（QCD別）一覧 …… 60
図表5.1-1　件数、分布の掲載対象と層別の種類 …… 62
図表5.2-1　SLOC規模の分布（改良（派生）開発、開発5工程、全体、5KSLOC刻み） …… 63
図表5.2-2　SLOC規模の分布（改良（派生）開発、開発5工程、20KSLOC以下、1KSLOC刻み） …… 63

図表5.2-3　SLOC規模の基本統計量（改良（派生）開発、開発5工程） 63
図表5.2-4　リアルタイム性（時間制約）別SLOC規模の分布
（改良（派生）開発、開発5工程、全体、5KSLOC刻み） 64
図表5.2-5　リアルタイム性（時間制約）別SLOC規模の分布
（改良（派生）開発、開発5工程、20KSLOC以下、1KSLOC刻み） 64
図表5.2-6　リアルタイム性（時間制約）別SLOC規模の基本統計量（改良（派生）開発、開発5工程） 64
図表5.2-7　自然環境からの影響度合い別SLOC規模の分布
（改良（派生）開発、開発5工程、全体、5KSLOC刻み） 64
図表5.2-8　自然環境からの影響度合い別SLOC規模の分布
（改良（派生）開発、開発5工程、20KSLOC以下、1KSLOC刻み） 64
図表5.2-9　自然環境からの影響度合い別SLOC規模の基本統計量（改良（派生）開発、開発5工程） 64
図表5.2-10　ユーザの多様性別SLOC規模の分布（改良（派生）開発、開発5工程、全体、5KSLOC刻み） 65
図表5.2-11　ユーザの多様性別SLOC規模の分布
（改良（派生）開発、開発5工程、20KSLOC以下、1KSLOC刻み） 65
図表5.2-12　ユーザの多様性別SLOC規模の基本統計量（改良（派生）開発、開発5工程） 65
図表5.2-13　法規等による規制度合い別SLOC規模の分布
（改良（派生）開発、開発5工程、全体、5KSLOC刻み） 65
図表5.2-14　法規等による規制度合い別SLOC規模の分布
（改良（派生）開発、開発5工程、20KSLOC以下、1KSLOC刻み） 65
図表5.2-15　法規等による規制度合い別SLOC規模の基本統計量（改良（派生）開発、開発5工程） 65
図表5.2-16　M２Mの有無別SLOC規模の分布（改良（派生）開発、開発5工程、全体、5KSLOC刻み） 66
図表5.2-17　M２Mの有無別SLOC規模の分布
（改良（派生）開発、開発5工程、20KSLOC以下、1KSLOC刻み） 66
図表5.2-18　M２Mの有無別SLOC規模の基本統計量（改良（派生）開発、開発5工程） 66
図表5.2-19　ネットワーク接続の有無別SLOC規模の分布
（改良（派生）開発、開発5工程、全体、5KSLOC刻み） 66
図表5.2-20　ネットワーク接続の有無別SLOC規模の分布
（改良（派生）開発、開発5工程、20KSLOC以下、1KSLOC刻み） 66
図表5.2-21　ネットワーク接続の有無別SLOC規模の基本統計量（改良（派生）開発、開発5工程） 66
図表5.2-22　稼動（非停止、オンデマンド）別SLOC規模の分布
（改良（派生）開発、開発5工程、全体、5KSLOC刻み） 67
図表5.2-23　稼動（非停止、オンデマンド）別SLOC規模の分布
（改良（派生）開発、開発5工程、20KSLOC以下、1KSLOC刻み） 67
図表5.2-24　稼動（非停止、オンデマンド）別SLOC規模の基本統計量（改良（派生）開発、開発5工程） 67
図表5.2-25　オンライン保守の可否別SLOC規模の分布
（改良（派生）開発、開発5工程、全体、5KSLOC刻み） 67
図表5.2-26　オンライン保守の可否別SLOC規模の分布
（改良（派生）開発、開発5工程、20KSLOC以下、1KSLOC刻み） 67
図表5.2-27　オンライン保守の可否別SLOC規模の基本統計量（改良（派生）開発、開発5工程） 67
図表5.2-28　障害リスク（TYPE）別SLOC規模の分布
（改良（派生）開発、開発5工程、全体、5KSLOC刻み） 68
図表5.2-29　障害リスク（TYPE）別SLOC規模の分布
（改良（派生）開発、開発5工程、20KSLOC以下、1KSLOC刻み） 68
図表5.2-30　障害リスク（TYPE）別SLOC規模の基本統計量（改良（派生）開発、開発5工程） 68
図表5.2-31　コスト実績の評価別SLOC規模の分布
（改良（派生）開発、開発5工程、全体、5KSLOC刻み） 69
図表5.2-32　コスト実績の評価別SLOC規模の分布
（改良（派生）開発、開発5工程、20KSLOC以下、1KSLOC刻み） 69
図表5.2-33　コスト実績の評価別SLOC規模の基本統計量（改良（派生）開発、開発5工程） 69
図表5.2-34　品質実績の評価別SLOC規模の分布（改良（派生）開発、開発5工程、全体、5KSLOC刻み） 69
図表5.2-35　品質実績の評価別SLOC規模の分布
（改良（派生）開発、開発5工程、20KSLOC以下、1KSLOC刻み） 69
図表5.2-36　品質実績の評価別SLOC規模の基本統計量（改良（派生）開発、開発5工程） 69
図表5.2-37　工期実績の評価別SLOC規模の分布（改良（派生）開発、開発5工程、全体、5KSLOC刻み） 70

図表5.2-38　工期実績の評価別SLOC規模の分布
(改良(派生)開発、開発5工程、20KSLOC以下、1KSLOC刻み) …… 70
図表5.2-39　工期実績の評価別SLOC規模の基本統計量(改良(派生)開発、開発5工程) …… 70
図表5.3-1　工数の分布(改良(派生)開発、開発5工程、全体、2,000人刻み) …… 71
図表5.3-2　工数の分布(改良(派生)開発、開発5工程、4,000人以下、200人刻み) …… 71
図表5.3-3　工数の基本統計量(改良(派生)開発、開発5工程) …… 71
図表5.3-4　リアルタイム性(時間制約)別工数の分布
(改良(派生)開発、開発5工程、全体、2,000人刻み) …… 72
図表5.3-5　リアルタイム性(時間制約)別工数の分布
(改良(派生)開発、開発5工程、4,000人以下、200人刻み) …… 72
図表5.3-6　リアルタイム性(時間制約)別工数の基本統計量(改良(派生)開発、開発5工程) …… 72
図表5.3-7　自然環境からの影響度合い別工数の分布
(改良(派生)開発、開発5工程、全体、2,000人刻み) …… 72
図表5.3-8　自然環境からの影響度合い別工数の分布
(改良(派生)開発、開発5工程、4,000人以下、200人刻み) …… 72
図表5.3-9　自然環境からの影響度合い別工数の基本統計量(改良(派生)開発、開発5工程) …… 72
図表5.3-10　ユーザの多様性別工数の分布(改良(派生)開発、開発5工程、全体、2,000人刻み) …… 73
図表5.3-11　ユーザの多様性別工数の分布(改良(派生)開発、開発5工程、4,000人以下、200人刻み) …… 73
図表5.3-12　ユーザの多様性別工数の基本統計量(改良(派生)開発、開発5工程) …… 73
図表5.3-13　法規等による規制度合い別工数の分布
(改良(派生)開発、開発5工程、全体、2,000人刻み) …… 73
図表5.3-14　法規等による規制度合い別工数の分布
(改良(派生)開発、開発5工程、4,000人以下、200人刻み) …… 73
図表5.3-15　法規等による規制度合い別工数の基本統計量(改良(派生)開発、開発5工程) …… 73
図表5.3-16　M2Mの有無別工数の分布(改良(派生)開発、開発5工程、全体、2,000人刻み) …… 74
図表5.3-17　M2Mの有無別工数の分布(改良(派生)開発、開発5工程、4,000人以下、200人刻み) …… 74
図表5.3-18　M2Mの有無別工数の基本統計量(改良(派生)開発、開発5工程) …… 74
図表5.3-19　ネットワーク接続の有無別工数の分布
(改良(派生)開発、開発5工程、全体、2,000人刻み) …… 74
図表5.3-20　ネットワーク接続の有無別工数の分布
(改良(派生)開発、開発5工程、4,000人以下、200人刻み) …… 74
図表5.3-21　ネットワーク接続の有無別工数の基本統計量(改良(派生)開発、開発5工程) …… 74
図表5.3-22　稼動(非停止、オンデマンド)別工数の分布
(改良(派生)開発、開発5工程、全体、2,000人刻み) …… 75
図表5.3-23　稼動(非停止、オンデマンド)別工数の分布
(改良(派生)開発、開発5工程、4,000人以下、200人刻み) …… 75
図表5.3-24　稼動(非停止、オンデマンド)別工数の基本統計量(改良(派生)開発、開発5工程) …… 75
図表5.3-25　オンライン保守の可否別工数の分布(改良(派生)開発、開発5工程、全体、2,000人刻み) …… 75
図表5.3-26　オンライン保守の可否別工数の分布
(改良(派生)開発、開発5工程、4,000人以下、200人刻み) …… 75
図表5.3-27　オンライン保守の可否別工数の基本統計量(改良(派生)開発、開発5工程) …… 75
図表5.3-28　障害リスク(TYPE)別工数の分布(改良(派生)開発、開発5工程、全体、2,000人刻み) …… 76
図表5.3-29　障害リスク(TYPE)別工数の分布
(改良(派生)開発、開発5工程、4,000人以下、200人刻み) …… 76
図表5.3-30　障害リスク(TYPE)別工数の基本統計量(改良(派生)開発、開発5工程) …… 76
図表5.3-31　コスト実績の評価別工数の分布(改良(派生)開発、開発5工程、全体、2,000人刻み) …… 77
図表5.3-32　コスト実績の評価別工数の分布
(改良(派生)開発、開発5工程、4,000人以下、200人刻み) …… 77
図表5.3-33　コスト実績の評価別工数の基本統計量(改良(派生)開発、開発5工程) …… 77
図表5.3-34　品質実績の評価別工数の分布(改良(派生)開発、開発5工程、全体、2,000人刻み) …… 77
図表5.3-35　品質実績の評価別工数の分布(改良(派生)開発、開発5工程、4,000人以下、200人刻み) …… 77
図表5.3-36　品質実績の評価別工数の基本統計量(改良(派生)開発、開発5工程) …… 77
図表5.3-37　工期実績の評価別工数の分布(改良(派生)開発、開発5工程、全体、2,000人刻み) …… 78
図表5.3-38　工期実績の評価別工数の分布(改良(派生)開発、開発5工程、4,000人以下、200人刻み) …… 78

図表5.3-39 工期実績の評価別工数の基本統計量（改良（派生）開発、開発5工程） 78
図表6.1-1 分析対象と層別のパターン 80
図表6.1-2 主な要素データと参照先の節番号 80
図表6.2-1 開発言語別の工数と工期（改良（派生）開発、開発5工程） 81
図表6.2-2 開発言語別の工数と工期（改良（派生）開発、開発5工程）対数表示 81
図表6.2-3 開発言語別の実績月数の基本統計量（改良（派生）開発、開発5工程） 81
図表6.2-4 開発言語C/C++の工数と工期（改良（派生）開発、開発5工程、開発言語C/C++） 82
図表6.2-5 開発言語C/C++の工数と工期（改良（派生）開発、開発5工程、開発言語C/C++）対数表示 82
図表6.2-6 開発言語C/C++の実績月数の基本統計量
（改良（派生）開発、開発5工程、開発言語C/C++） 82
図表6.3-1 開発言語別のSLOC規模と工数（改良（派生）開発、開発5工程） 83
図表6.3-2 開発言語別のSLOC規模と工数（改良（派生）開発、開発5工程）対数表示 83
図表6.3-3 開発言語別の実績工数の基本統計量（改良（派生）開発、開発5工程） 83
図表6.3-4 開発言語C/C++のSLOC規模と工数（改良（派生）開発、開発5工程、開発言語C/C++） 84
図表6.3-5 開発言語C/C++のSLOC規模と工数
（改良（派生）開発、開発5工程、開発言語C/C++）対数表示 84
図表6.3-6 開発言語C/C++の実績工数の基本統計量
（改良（派生）開発、開発5工程、開発言語C/C++） 84
図表6.4-1 工程別の実績月数の比率（改良（派生）開発）箱ひげ図 85
図表6.4-2 工程別の実績月数の比率の基本統計量（改良（派生）開発） 85
図表6.4-3 工程別の実績工数の比率（改良（派生）開発）箱ひげ図 86
図表6.4-4 工程別の実績工数の比率の基本統計量（改良（派生）開発） 86
図表7.1-1 分析対象と層別のパターン 88
図表7.1-2 主な要素データと参照先の節番号 88
図表7.2-1 開発言語別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程） 89
図表7.2-2 開発言語別のSLOC規模とSLOC生産性（改良（派生）開発、開発5工程）対数表示 89
図表7.2-3 開発言語別SLOC生産性（改良（派生）開発、開発5工程）箱ひげ図 89
図表7.2-4 開発言語別SLOC生産性の基本統計量（改良（派生）開発、開発5工程） 89
図表7.2-5 SLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++） 90
図表7.2-6 SLOC規模とSLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++）対数表示 90
図表7.2-7 SLOC規模別SLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++）箱ひげ図 90
図表7.2-8 SLOC規模別SLOC生産性の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++） 90
図表7.2-9 母体SLOC規模別のSLOC規模とSLOC生産性
（改良（派生）開発、開発5工程、開発言語C/C++） 91
図表7.2-10 母体SLOC規模別のSLOC規模とSLOC生産性
（改良（派生）開発、開発5工程、開発言語C/C++）対数表示 91
図表7.2-11 母体SLOC規模別SLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++）箱ひげ図 91
図表7.2-12 母体SLOC規模別SLOC生産性の基本統計量
（改良（派生）開発、開発5工程、開発言語C/C++） 91
図表7.3-1 工程別SLOC生産性（改良（派生）開発、開発5工程、開発言語C/C++）箱ひげ図 92
図表7.3-2 工程別SLOC生産性の基本統計量（改良（派生）開発、開発5工程、開発言語C/C++） 92
図表8.1-1 分析対象と層別のパターン 94
図表8.1-2 主な要素データと参照先の節番号 94
図表8.2-1 開発言語別のSLOC規模と結合テスト検出バグ密度（改良（派生）開発、0.1KSLOC以上） 95
図表8.2-2 開発言語別のSLOC規模と結合テスト検出バグ密度
（改良（派生）開発、0.1KSLOC以上）対数表示 95
図表8.2-3 開発言語別結合テスト検出バグ密度（改良（派生）開発、0.1KSLOC以上）箱ひげ図 95
図表8.2-4 開発言語別結合テスト検出バグ密度の基本統計量（改良（派生）開発、0.1KSLOC以上） 95
図表8.2-5 開発言語別のSLOC規模と結合テスト検出バグ密度
（改良（派生）開発、開発5工程、0.1KSLOC以上） 96
図表8.2-6 開発言語別のSLOC規模と結合テスト検出バグ密度
（改良（派生）開発、開発5工程、0.1KSLOC以上）対数表示 96
図表8.2-7 開発言語別結合テスト検出バグ密度
（改良（派生）開発、開発5工程、0.1KSLOC以上）箱ひげ図 96

図表8.2-8　開発言語別結合テスト検出バグ密度の基本統計量
(改良(派生)開発、開発5工程、0.1KSLOC以上) …… 96
図表8.2-9　SLOC規模と結合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 97
図表8.2-10　SLOC規模と結合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 97
図表8.2-11　SLOC規模別結合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 97
図表8.2-12　SLOC規模別結合テスト検出バグ密度の基本統計量
(改良(派生)開発、開発5工程、0.1KSLOC以上) …… 97
図表8.2-13　開発言語別のSLOC規模と総合テスト検出バグ密度(改良(派生)開発、0.1KSLOC以上) …… 98
図表8.2-14　開発言語別のSLOC規模と総合テスト検出バグ密度
(改良(派生)開発、0.1KSLOC以上)対数表示 …… 98
図表8.2-15　開発言語別総合テスト検出バグ密度(改良(派生)開発、0.1KSLOC以上)箱ひげ図 …… 98
図表8.2-16　開発言語別総合テスト検出バグ密度の基本統計量(改良(派生)開発、0.1KSLOC以上) …… 98
図表8.2-17　開発言語別のSLOC規模と総合テスト検出バグ密度
(改良(派生)開発、開発5工程、0.1KSLOC以上) …… 99
図表8.2-18　開発言語別のSLOC規模と総合テスト検出バグ密度
(改良(派生)開発、開発5工程、0.1KSLOC以上)対数表示 …… 99
図表8.2-19　開発言語別総合テスト検出バグ密度
(改良(派生)開発、開発5工程、0.1KSLOC以上)箱ひげ図 …… 99
図表8.2-20　開発言語別総合テスト検出バグ密度の基本統計量
(改良(派生)開発、開発5工程、0.1KSLOC以上) …… 99
図表8.2-21　SLOC規模と総合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 100
図表8.2-22　SLOC規模と総合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 100
図表8.2-23　SLOC規模別総合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 100
図表8.2-24　SLOC規模別総合テスト検出バグ密度の基本統計量
(改良(派生)開発、開発5工程、0.1KSLOC以上) …… 100
図表8.2-25　母体SLOC規模別のSLOC規模と総合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 101
図表8.2-26　母体SLOC規模別のSLOC規模と総合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 101
図表8.2-27　母体SLOC規模別総合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 101
図表8.2-28　母体SLOC規模別総合テスト検出バグ密度の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 101
図表8.3-1　SLOC規模あたりのテストケース数
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 102
図表8.3-2　SLOC規模あたりの検出バグ数
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 102
図表8.3-3　SLOC規模あたりのテストケース数、検出バグ数の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 102
図表8.3-4　結合テストケース密度と検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 103
図表8.3-5　結合テストケース密度と検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 103
図表8.3-6　総合テストケース密度と検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 103
図表8.3-7　総合テストケース密度と検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 103

図表8.3-8　母体含むSLOC規模あたりのテストケース数
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 104
図表8.3-9　母体含むSLOC規模あたりの検出バグ数
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 104
図表8.3-10　母体含むSLOC規模あたりのテストケース数、検出バグ数の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 104
図表8.3-11　母体含む結合テストケース密度と検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 105
図表8.3-12　母体含む結合テストケース密度と検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 105
図表8.3-13　母体含む総合テストケース密度と検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 105
図表8.3-14　母体含む総合テストケース密度と検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 105
図表8.3-15　SLOC規模あたりのテスト実績工数
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 106
図表8.3-16　SLOC規模あたりのテスト実績工数の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 106
図表8.3-17　母体含むSLOC規模あたりのテスト実績工数
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 107
図表8.3-18　母体含むSLOC規模あたりのテスト実績工数の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 107
図表8.4-1　SLOC規模と結合テスト実績月数
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 108
図表8.4-2　SLOC規模と結合テスト実績月数
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 108
図表8.4-3　SLOC規模と結合テスト実績月数の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 108
図表8.4-4　SLOC規模と総合テスト実績月数
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 109
図表8.4-5　SLOC規模と総合テスト実績月数
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 109
図表8.4-6　SLOC規模と総合テスト実績月数の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 109
図表8.5-1　工数あたりのテストケース数
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 110
図表8.5-2　工数あたりの検出バグ数
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 110
図表8.5-3　工数あたりのテストケース数の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 110
図表8.5-4　結合テスト工数あたりのテストケース数と検出バグ数
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 111
図表8.5-5　結合テスト工数あたりのテストケース数と検出バグ数
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 111
図表8.5-6　総合テスト工数あたりのテストケース数と検出バグ数
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 111
図表8.5-7　総合テスト工数あたりのテストケース数と検出バグ数
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 111
図表9.1-1　分析対象と層別のパターン …… 114
図表9.1-2　主な要素データと参照先の節番号 …… 114
図表9.2-1　結合テスト検出バグ密度とSLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 115
図表9.2-2　結合テスト検出バグ密度とSLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 115

図表9.2-3　総合テスト検出バグ密度とSLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 116
図表9.2-4　総合テスト検出バグ密度とSLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 116
図表9.2-5　結合テスト検出バグ密度別のSLOC規模とSLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 117
図表9.2-6　結合テスト検出バグ密度別のSLOC規模とSLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 117
図表9.2-7　総合テスト検出バグ密度別のSLOC規模とSLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 117
図表9.2-8　総合テスト検出バグ密度別のSLOC規模とSLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 117
図表9.2-9　結合テスト検出バグ密度別SLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 118
図表9.2-10　結合テスト検出バグ密度別SLOC生産性の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 118
図表9.2-11　総合テスト検出バグ密度別SLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 118
図表9.2-12　総合テスト検出バグ密度別SLOC生産性の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 118
図表10.1-1　分析対象と層別のパターン …… 121
図表10.1-2　主な要素データと参照先の節番号 …… 122
図表10.2-1　リアルタイム性(時間制約)別の工数と工期
(改良(派生)開発、開発5工程、開発言語C/C++) …… 123
図表10.2-2　リアルタイム性(時間制約)別の工数と工期
(改良(派生)開発、開発5工程、開発言語C/C++)対数表示 …… 123
図表10.2-3　リアルタイム性(時間制約)別の実績月数の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++) …… 123
図表10.2-4　リアルタイム性(時間制約)別のSLOC規模と工数
(改良(派生)開発、開発5工程、開発言語C/C++) …… 124
図表10.2-5　リアルタイム性(時間制約)別のSLOC規模と工数
(改良(派生)開発、開発5工程、開発言語C/C++)対数表示 …… 124
図表10.2-6　リアルタイム性(時間制約)別の実績工数の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++) …… 124
図表10.2-7　リアルタイム性(時間制約)別の工程別の実績工数の比率
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 125
図表10.2-8　リアルタイム性(時間制約)別の工程別の実績工数の比率の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 126
図表10.2-9　リアルタイム性(時間制約)別のSLOC規模とSLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 127
図表10.2-10　リアルタイム性(時間制約)別のSLOC規模とSLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 127
図表10.2-11　リアルタイム性(時間制約)別SLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 127
図表10.2-12　リアルタイム性(時間制約)別SLOC生産性の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 127
図表10.2-13　リアルタイム性(時間制約)別のSLOC規模と結合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 128
図表10.2-14　リアルタイム性(時間制約)別のSLOC規模と結合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 128
図表10.2-15　リアルタイム性(時間制約)別結合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 128
図表10.2-16　リアルタイム性(時間制約)別結合テスト検出バグ密度の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 128

図表10.2-17　リアルタイム性（時間制約）別のSLOC規模と総合テスト検出バグ密度
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）…… 129
図表10.2-18　リアルタイム性（時間制約）別のSLOC規模と総合テスト検出バグ密度
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示 …… 129
図表10.2-19　リアルタイム性（時間制約）別総合テスト検出バグ密度
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図 …… 129
図表10.2-20　リアルタイム性（時間制約）別総合テスト検出バグ密度の基本統計量
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）…… 129
図表10.3-1　自然環境からの影響度合い別の工数と工期
（改良（派生）開発、開発5工程、開発言語C/C++）…… 130
図表10.3-2　自然環境からの影響度合い別の工数と工期
（改良（派生）開発、開発5工程、開発言語C/C++）対数表示 …… 130
図表10.3-3　自然環境からの影響度合い別の実績月数の基本統計量
（改良（派生）開発、開発5工程、開発言語C/C++）…… 130
図表10.3-4　自然環境からの影響度合い別のSLOC規模と工数
（改良（派生）開発、開発5工程、開発言語C/C++）…… 131
図表10.3-5　自然環境からの影響度合い別のSLOC規模と工数
（改良（派生）開発、開発5工程、開発言語C/C++）対数表示 …… 131
図表10.3-6　自然環境からの影響度合い別の実績工数の基本統計量
（改良（派生）開発、開発5工程、開発言語C/C++）…… 131
図表10.3-7　自然環境からの影響度合い別の工程別の実績工数の比率
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図 …… 132
図表10.3-8　自然環境からの影響度合い別の工程別の実績工数の比率の基本統計量
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）…… 132
図表10.3-9　自然環境からの影響度合い別のSLOC規模とSLOC生産性
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）…… 133
図表10.3-10　自然環境からの影響度合い別のSLOC規模とSLOC生産性
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示 …… 133
図表10.3-11　自然環境からの影響度合い別SLOC生産性
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図 …… 133
図表10.3-12　自然環境からの影響度合い別SLOC生産性の基本統計量
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）…… 133
図表10.3-13　自然環境からの影響度合い別のSLOC規模と結合テスト検出バグ密度
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）…… 134
図表10.3-14　自然環境からの影響度合い別のSLOC規模と結合テスト検出バグ密度
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示 …… 134
図表10.3-15　自然環境からの影響度合い別結合テスト検出バグ密度
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図 …… 134
図表10.3-16　自然環境からの影響度合い別結合テスト検出バグ密度の基本統計量
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）…… 134
図表10.3-17　自然環境からの影響度合い別のSLOC規模と総合テスト検出バグ密度
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）…… 135
図表10.3-18　自然環境からの影響度合い別のSLOC規模と総合テスト検出バグ密度
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示 …… 135
図表10.3-19　自然環境からの影響度合い別総合テスト検出バグ密度
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図 …… 135
図表10.3-20　自然環境からの影響度合い別総合テスト検出バグ密度の基本統計量
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）…… 135
図表10.4-1　ユーザの多様性別の工数と工期（改良（派生）開発、開発5工程、開発言語C/C++）…… 136
図表10.4-2　ユーザの多様性別の工数と工期
（改良（派生）開発、開発5工程、開発言語C/C++）対数表示 …… 136
図表10.4-3　ユーザの多様性別の実績月数の基本統計量
（改良（派生）開発、開発5工程、開発言語C/C++）…… 136
図表10.4-4　ユーザの多様性別のSLOC規模と工数（改良（派生）開発、開発5工程、開発言語C/C++）…… 137

図表10.4-5　ユーザの多様性別のSLOC規模と工数
(改良(派生)開発、開発5工程、開発言語C/C++)対数表示 …… 137
図表10.4-6　ユーザの多様性別の実績工数の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++) …… 137
図表10.4-7　ユーザの多様性別の工程別の実績工数の比率
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 138
図表10.4-8　ユーザの多様性別の工程別の実績工数の比率の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 138
図表10.4-9　ユーザの多様性別のSLOC規模とSLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 139
図表10.4-10　ユーザの多様性別のSLOC規模とSLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 139
図表10.4-11　ユーザの多様性別SLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 139
図表10.4-12　ユーザの多様性別SLOC生産性の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 139
図表10.4-13　ユーザの多様性別のSLOC規模と結合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 140
図表10.4-14　ユーザの多様性別のSLOC規模と結合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 140
図表10.4-15　ユーザの多様性別結合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 140
図表10.4-16　ユーザの多様性別結合テスト検出バグ密度の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 140
図表10.4-17　ユーザの多様性別のSLOC規模と総合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 141
図表10.4-18　ユーザの多様性別のSLOC規模と総合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 141
図表10.4-19　ユーザの多様性別総合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 141
図表10.4-20　ユーザの多様性別総合テスト検出バグ密度の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 141
図表10.5-1　法規等による規制度合い別の工数と工期
(改良(派生)開発、開発5工程、開発言語C/C++) …… 142
図表10.5-2　法規等による規制度合い別の工数と工期
(改良(派生)開発、開発5工程、開発言語C/C++)対数表示 …… 142
図表10.5-3　法規等による規制度合い別の実績月数の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++) …… 142
図表10.5-4　法規等による規制度合い別のSLOC規模と工数
(改良(派生)開発、開発5工程、開発言語C/C++) …… 143
図表10.5-5　法規等による規制度合い別のSLOC規模と工数
(改良(派生)開発、開発5工程、開発言語C/C++)対数表示 …… 143
図表10.5-6　法規等による規制度合い別の実績工数の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++) …… 143
図表10.5-7　法規等による規制度合い別の工程別の実績工数の比率
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 144
図表10.5-8　法規等による規制度合い別の工程別の実績工数の比率の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 144
図表10.5-9　法規等による規制度合い別のSLOC規模とSLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 145
図表10.5-10　法規等による規制度合い別SLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 145
図表10.5-11　法規等による規制度合い別SLOC生産性の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 145

図表10.5-12　法規等による規制度合い別のSLOC規模と結合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 145
図表10.5-13　法規等による規制度合い別のSLOC規模と結合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 146
図表10.5-14　法規等による規制度合い別結合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 146
図表10.5-15　法規等による規制度合い別結合テスト検出バグ密度の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 146
図表10.5-16　法規等による規制度合い別のSLOC規模と総合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 146
図表10.5-17　法規等による規制度合い別のSLOC規模と総合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 147
図表10.5-18　法規等による規制度合い別総合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 147
図表10.5-19　法規等による規制度合い別総合テスト検出バグ密度の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 147
図表10.5-20　法規等による規制度合い別総合テスト検出バグ密度の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 147
図表10.6-1　M2Mの有無別の工数と工期(改良(派生)開発、開発5工程、開発言語C/C++) …… 148
図表10.6-2　M2Mの有無別の工数と工期(改良(派生)開発、開発5工程、開発言語C/C++)対数表示 …… 148
図表10.6-3　M2Mの有無別の実績月数の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++) …… 148
図表10.6-4　M2Mの有無別のSLOC規模と工数(改良(派生)開発、開発5工程、開発言語C/C++) …… 149
図表10.6-5　M2Mの有無別のSLOC規模と工数
(改良(派生)開発、開発5工程、開発言語C/C++)対数表示 …… 149
図表10.6-6　M2Mの有無別の実績工数の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++) …… 149
図表10.6-7　M2Mの有無別の工程別の実績工数の比率
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 150
図表10.6-8　M2Mの有無別の工程別の実績工数の比率の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 150
図表10.6-9　M2Mの有無別のSLOC規模とSLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 151
図表10.6-10　M2Mの有無別のSLOC規模とSLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 151
図表10.6-11　M2Mの有無別SLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 151
図表10.6-12　M2Mの有無別SLOC生産性の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 151
図表10.6-13　M2Mの有無別のSLOC規模と結合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 152
図表10.6-14　M2Mの有無別のSLOC規模と結合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 152
図表10.6-15　M2Mの有無別結合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 152
図表10.6-16　M2Mの有無別結合テスト検出バグ密度の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 152
図表10.6-17　M2Mの有無別のSLOC規模と総合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 153
図表10.6-18　M2Mの有無別のSLOC規模と総合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 153
図表10.6-19　M2Mの有無別総合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 153

図表10.6-20　M２Mの有無別総合テスト検出バグ密度の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 153
図表10.7-1　ネットワーク接続の有無別の工数と工期
(改良(派生)開発、開発5工程、開発言語C/C++) …… 154
図表10.7-2　ネットワーク接続の有無別の工数と工期
(改良(派生)開発、開発5工程、開発言語C/C++)対数表示 …… 154
図表10.7-3　ネットワーク接続の有無別の実績月数の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++) …… 154
図表10.7-4　ネットワーク接続の有無別のSLOC規模と工数
(改良(派生)開発、開発5工程、開発言語C/C++) …… 155
図表10.7-5　ネットワーク接続の有無別のSLOC規模と工数
(改良(派生)開発、開発5工程、開発言語C/C++)対数表示 …… 155
図表10.7-6　ネットワーク接続の有無別の実績工数の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++) …… 155
図表10.7-7　ネットワーク接続の有無別の工程別の実績工数の比率
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 156
図表10.7-8　ネットワーク接続の有無別の工程別の実績工数の比率の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 156
図表10.7-9　ネットワーク接続の有無別のSLOC規模とSLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 157
図表10.7-10　ネットワーク接続の有無別のSLOC規模とSLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 157
図表10.7-11　ネットワーク接続の有無別SLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 157
図表10.7-12　ネットワーク接続の有無別SLOC生産性の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 157
図表10.7-13　ネットワーク接続の有無別のSLOC規模と結合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 158
図表10.7-14　ネットワーク接続の有無別のSLOC規模と結合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 158
図表10.7-15　ネットワーク接続の有無別結合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 158
図表10.7-16　ネットワーク接続の有無別結合テスト検出バグ密度の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 158
図表10.7-17　ネットワーク接続の有無別のSLOC規模と総合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 159
図表10.7-18　ネットワーク接続の有無別のSLOC規模と総合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 159
図表10.7-19　ネットワーク接続の有無別総合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 159
図表10.7-20　ネットワーク接続の有無別総合テスト検出バグ密度の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 159
図表10.8-1　稼動(非停止、オンデマンド)別の工数と工期
(改良(派生)開発、開発5工程、開発言語C/C++) …… 160
図表10.8-2　稼動(非停止、オンデマンド)別の工数と工期
(改良(派生)開発、開発5工程、開発言語C/C++)対数表示 …… 160
図表10.8-3　稼動(非停止、オンデマンド)別の実績月数の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++) …… 160
図表10.8-4　稼動(非停止、オンデマンド)別のSLOC規模と工数
(改良(派生)開発、開発5工程、開発言語C/C++) …… 161
図表10.8-5　稼動(非停止、オンデマンド)別のSLOC規模と工数
(改良(派生)開発、開発5工程、開発言語C/C++)対数表示 …… 161
図表10.8-6　稼動(非停止、オンデマンド)別の実績工数の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++) …… 161

図表10.8-7　稼動（非停止、オンデマンド）別の工程別の実績工数の比率
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図 …… 162
図表10.8-8　稼動（非停止、オンデマンド）別の工程別の実績工数の比率の基本統計量
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上） …… 162
図表10.8-9　稼動（非停止、オンデマンド）別のSLOC規模とSLOC生産性
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上） …… 163
図表10.8-10　稼動（非停止、オンデマンド）別のSLOC規模とSLOC生産性
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示 …… 163
図表10.8-11　稼動（非停止、オンデマンド）別SLOC生産性
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図 …… 163
図表10.8-12　稼動（非停止、オンデマンド）別SLOC生産性の基本統計量
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上） …… 163
図表10.8-13　稼動（非停止、オンデマンド）別のSLOC規模と結合テスト検出バグ密度
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上） …… 164
図表10.8-14　稼動（非停止、オンデマンド）別のSLOC規模と結合テスト検出バグ密度
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示 …… 164
図表10.8-15　稼動（非停止、オンデマンド）別結合テスト検出バグ密度
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図 …… 164
図表10.8-16　稼動（非停止、オンデマンド）別結合テスト検出バグ密度の基本統計量
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上） …… 164
図表10.8-17　稼動（非停止、オンデマンド）別のSLOC規模と総合テスト検出バグ密度
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上） …… 165
図表10.8-18　稼動（非停止、オンデマンド）別のSLOC規模と総合テスト検出バグ密度
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示 …… 165
図表10.8-19　稼動（非停止、オンデマンド）別総合テスト検出バグ密度
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図 …… 165
図表10.8-20　稼動（非停止、オンデマンド）別総合テスト検出バグ密度の基本統計量
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上） …… 165
図表10.9-1　オンライン保守の可否別の工数と工期（改良（派生）開発、開発5工程、開発言語C/C++） …… 166
図表10.9-2　オンライン保守の可否別の工数と工期
（改良（派生）開発、開発5工程、開発言語C/C++）対数表示 …… 166
図表10.9-3　オンライン保守の可否別の実績月数の基本統計量
（改良（派生）開発、開発5工程、開発言語C/C++） …… 166
図表10.9-4　オンライン保守の可否別のSLOC規模と工数
（改良（派生）開発、開発5工程、開発言語C/C++） …… 167
図表10.9-5　オンライン保守の可否別のSLOC規模と工数
（改良（派生）開発、開発5工程、開発言語C/C++）対数表示 …… 167
図表10.9-6　オンライン保守の可否別の実績工数の基本統計量
（改良（派生）開発、開発5工程、開発言語C/C++） …… 167
図表10.9-7　オンライン保守の可否別の工程別の実績工数の比率
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図 …… 168
図表10.9-8　オンライン保守の可否別の工程別の実績工数の比率の基本統計量
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上） …… 168
図表10.9-9　オンライン保守の可否別のSLOC規模とSLOC生産性
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上） …… 169
図表10.9-10　オンライン保守の可否別のSLOC規模とSLOC生産性
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）対数表示 …… 169
図表10.9-11　オンライン保守の可否別SLOC生産性
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上）箱ひげ図 …… 169
図表10.9-12　オンライン保守の可否別SLOC生産性の基本統計量
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上） …… 169
図表10.9-13　オンライン保守の可否別のSLOC規模と結合テスト検出バグ密度
（改良（派生）開発、開発5工程、開発言語C/C++、0.1KSLOC以上） …… 170

図表10.9-14　オンライン保守の可否別のSLOC規模と結合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 170
図表10.9-15　オンライン保守の可否別結合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 170
図表10.9-16　オンライン保守の可否別結合テスト検出バグ密度の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 170
図表10.9-17　オンライン保守の可否別のSLOC規模と総合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 171
図表10.9-18　オンライン保守の可否別のSLOC規模と総合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 171
図表10.9-19　オンライン保守の可否別総合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 171
図表10.9-20　オンライン保守の可否別総合テスト検出バグ密度の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 171
図表10.10-1　障害リスク(TYPE)別の工数と工期(改良(派生)開発、開発5工程、開発言語C/C++) …… 172
図表10.10-2　障害リスク(TYPE)別の工数と工期
(改良(派生)開発、開発5工程、開発言語C/C++)対数表示 …… 172
図表10.10-3　障害リスク(TYPE)別の実績月数の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++) …… 172
図表10.10-4　障害リスク(TYPE)別のSLOC規模と工数
(改良(派生)開発、開発5工程、開発言語C/C++) …… 173
図表10.10-5　障害リスク(TYPE)別のSLOC規模と工数
(改良(派生)開発、開発5工程、開発言語C/C++)対数表示 …… 173
図表10.10-6　障害リスク(TYPE)別の実績工数の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++) …… 173
図表10.10-7　障害リスク(TYPE)別の工程別の実績工数の比率
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 174
図表10.10-8　障害リスク(TYPE)別の工程別の実績工数の比率の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 175
図表10.10-9　障害リスク(TYPE)別のSLOC規模とSLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 176
図表10.10-10　障害リスク(TYPE)別のSLOC規模とSLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 176
図表10.10-11　障害リスク(TYPE)別SLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 176
図表10.10-12　障害リスク(TYPE)別SLOC生産性の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 176
図表11.1-1　分析対象と層別のパターン …… 178
図表11.1-2　主な要素データと参照先の節番号 …… 178
図表11.2-1　品質実績の評価別の工数と工期(改良(派生)開発、開発5工程、開発言語C/C++) …… 179
図表11.2-2　品質実績の評価別の工数と工期
(改良(派生)開発、開発5工程、開発言語C/C++)対数表示 …… 179
図表11.2-3　品質実績の評価別の実績月数の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++) …… 179
図表11.2-4　品質実績の評価別のSLOC規模と工数(改良(派生)開発、開発5工程、開発言語C/C++) …… 180
図表11.2-5　品質実績の評価別のSLOC規模と工数
(改良(派生)開発、開発5工程、開発言語C/C++)対数表示 …… 180
図表11.2-6　品質実績の評価別の実績工数の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++) …… 180
図表11.2-7　品質実績の評価別の工程別の実績工数の比率
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 181
図表11.2-8　品質実績の評価別の工程別の実績工数の比率の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 182

図表11.2-9　品質実績の評価別のSLOC規模とSLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 183
図表11.2-10　品質実績の評価別のSLOC規模とSLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 183
図表11.2-11　品質実績の評価別SLOC生産性
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 183
図表11.2-12　品質実績の評価別SLOC生産性の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 183
図表11.2-13　品質実績の評価別のSLOC規模と結合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 184
図表11.2-14　品質実績の評価別のSLOC規模と結合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 184
図表11.2-15　品質実績の評価別結合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 184
図表11.2-16　品質実績の評価別結合テスト検出バグ密度の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 184
図表11.2-17　品質実績の評価別のSLOC規模と総合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 185
図表11.2-18　品質実績の評価別のSLOC規模と総合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 185
図表11.2-19　品質実績の評価別総合テスト検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 185
図表11.2-20　品質実績の評価別総合テスト検出バグ密度の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 185
図表11.2-21　品質実績の評価【リリース後不具合数が0件】のSLOC規模あたりのテストケース数
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 186
図表11.2-22　品質実績の評価【リリース後不具合数が0件】のSLOC規模あたりの検出バグ数
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 186
図表11.2-23　品質実績の評価【リリース後不具合数が0件】のSLOC規模あたりのテストケース数、
検出バグ数の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 186
図表11.2-24　品質実績の評価【リリース後不具合数が予測値以内】のSLOC規模あたりのテストケース数
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 187
図表11.2-25　品質実績の評価【リリース後不具合数が予測値以内】のSLOC規模あたりの検出バグ数
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 187
図表11.2-26　品質実績の評価【リリース後不具合数が予測値以内】のSLOC規模あたりのテストケース数、
検出バグ数の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 187
図表11.2-27　品質実績の評価【リリース後不具合数が予測値超過】のSLOC規模あたりのテストケース数
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 187
図表11.2-28　品質実績の評価【リリース後不具合数が予測値超過】のSLOC規模あたりの検出バグ数
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 187
図表11.2-29　品質実績の評価【リリース後不具合数が予測値超過】のSLOC規模あたりのテストケース数、
検出バグ数の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 187
図表11.2-30　品質実績の評価別の結合テストケース密度と検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 188
図表11.2-31　品質実績の評価別の結合テストケース密度と検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 188
図表11.2-32　品質実績の評価別の総合テストケース密度と検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 188
図表11.2-33　品質実績の評価別の総合テストケース密度と検出バグ密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 188

図表11.2-34　品質実績の評価【リリース後不具合数が0件】の工程別レビュー指摘密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 189
図表11.2-35　品質実績の評価【リリース後不具合数が0件】の工程別レビュー指摘密度の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 189
図表11.2-36　品質実績の評価【リリース後不具合数が予測値以内】の工程別レビュー指摘密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 189
図表11.2-37　品質実績の評価【リリース後不具合数が予測値以内】の工程別レビュー指摘密度の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 189
図表11.2-38　品質実績の評価【リリース後不具合数が予測値超過】の工程別レビュー指摘密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 190
図表11.2-39　品質実績の評価【リリース後不具合数が予測値超過】の工程別レビュー指摘密度の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 190
図表11.2-40　品質実績の評価別のアーキテクチャ設計レビュー指摘密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 191
図表11.2-41　品質実績の評価別のアーキテクチャ設計レビュー指摘密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 191
図表11.2-42　品質実績の評価別の詳細設計レビュー指摘密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 191
図表11.2-43　品質実績の評価別の詳細設計レビュー指摘密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 191
図表11.2-44　品質実績の評価別のソースコードレビュー指摘密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 191
図表11.2-45　品質実績の評価別のソースコードレビュー指摘密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 191
図表11.2-46　品質実績の評価【リリース後不具合数が0件】の工程別レビュー工数密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 192
図表11.2-47　品質実績の評価【リリース後不具合数が0件】の工程別レビュー工数密度の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 192
図表11.2-48　品質実績の評価【リリース後不具合数が予測値以内】の工程別レビュー工数密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 192
図表11.2-49　品質実績の評価【リリース後不具合数が予測値以内】の工程別レビュー工数密度の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 192
図表11.2-50　品質実績の評価【リリース後不具合数が予測値超過】の工程別レビュー工数密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)箱ひげ図 …… 193
図表11.2-51　品質実績の評価【リリース後不具合数が予測値超過】の工程別レビュー工数密度の基本統計量
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 193
図表11.2-52　品質実績の評価別のアーキテクチャ設計レビュー工数密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 194
図表11.2-53　品質実績の評価別のアーキテクチャ設計レビュー工数密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 194
図表11.2-54　品質実績の評価別の詳細設計レビュー工数密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 194
図表11.2-55　品質実績の評価別の詳細設計レビュー工数密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 194
図表11.2-56　品質実績の評価別のソースコードレビュー工数密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上) …… 194
図表11.2-57　品質実績の評価別のソースコードレビュー工数密度
(改良(派生)開発、開発5工程、開発言語C/C++、0.1KSLOC以上)対数表示 …… 194

監修

独立行政法人情報処理推進機構（IPA）
社会基盤センター

IPAでは、新たな技術の動向や情報セキュリティの脅威、求められるIT人材など、ITに関するさまざまな調査・分析により、産業の発展や社会課題の解決につなげる指針やヒントを社会に発信するとともに、ITの利活用を促進させ、安全なIT社会の実現に貢献するための基盤づくりに取り組んでいます。

URL　https://www.ipa.go.jp/ikc/index.html
所在地　〒113-6591 東京都文京区本駒込2-28-8 文京グリーンコート センターオフィス

執筆

山下　博之　IPA社会基盤センター
久野　倫義　IPA社会基盤センター
五味　弘　IPA社会基盤センター
松田　充弘　IPA社会基盤センター
田代　宣子　IPA社会基盤センター

分析協力

製品・制御システム定量データ収集・分析WG

組込みソフトウェア開発データ白書 2019

2019年11月19日　発行

監 修 者　独立行政法人情報処理推進機構
社会基盤センター
発 行 人　片岡　　晃
発 行 所　独立行政法人情報処理推進機構
〒113-6591
東京都文京区本駒込二丁目28番8号
文京グリーンコート センターオフィス
URL https://www.ipa.go.jp/ikc/index.html

ISBN978-4-905318-71-2

ISBN978-4-905318-71-2

IPA 独立行政法人 情報処理推進機構
社会基盤センター