

組込みソフトウェア開発データ白書の 取り組みと最新の分析結果

SEC調査役 三原 幸博 SEC研究員 松田 充弘 SEC研究員 田代 宣子

IPA/SECでは、2013年より「組込みソフトウェア開発データ白書」編纂事業を進めている。この事業は組込みシステム製品の開発企業から提供を受けたプロジェクト管理データを分析する活動で、2015年の初版に続き今年の11月には2017年版を刊行することができた。2017年版の特徴は、分析対象データ件数が累計で400件を超えたことにより、製品の特性ごとの分析ができるようになったことである。本稿では、その分析結果の中から、リアルタイム性の強弱や自然環境からの影響を受けるかどうかなどの組込みシステム製品の特性によって、生産性や信頼性の指標の傾向に違いが見られたことを紹介する。

1 取り組みの経緯

SECは、これまで日本の組込みソフトウェア開発現場のベストプラクティスを収集し、それを整理した「組込みシステム開発技術リファレンスESxRシリーズ」(図1)を発行してきた。中でも、標準プロセスを定義した「ESPR」が広く参照されるようになったことで、組込みソフトウェア開発作業を可視化し、生産性や信頼性のメトリクスを業界で共有する標準的な枠組みができた。その結果、組込みソフトウェア開発に特化した生産性や信頼性の指標の目安を知りたい、自社の開発力を他社と比べたいなどの意見や要望に応えるための体制作りの準備が整ってきた。このような経緯のもとで、社内の機密情報として扱われていたプロジェクト管理データを国内組込み業界に役立てるためにSECに集めて、みんなで分析する「製品・制御システム定量データ収集・分析WG」が立ち上がった。その活動成果として、組込みソフトウェア開発データ白書の初版を2015年に発行することができた。これを機に収集分析活動に賛同する企業が増

え、提供データ件数も累計416件まで増加し、初版に比べて深いところの分析ができるようになったため、第2版「2017年版」をこの11月に発行した。



図2 組込みソフトウェア開発データ白書2017

2 収集データの分布と分析対象

まず、白書2017年版編纂のために収集したプロジェクトデータ416件のプロファイルの特徴から、生産性や信頼性などを分析するために絞り込んだ対象範囲を示す。

(1) 改良(派生)開発

収集データの内訳は、新規開発7%、改良(派生)開発93%であったため、2015年版と同じく、改良(派生)開発を分析対象とした。

(2) 開発言語

収集データの開発言語は2015年版と同じく、圧倒的にC言語とC++言語が多く使われているため、言語の対象をC言語とC++言語に絞り込んだ。2015年版と同じく、C++言語はC言語の代用を目的にする傾向であったため、区別せずに分析した。

(3) SLOC規模

分析対象の規模感をつかむために、2017年版の収集データの開発規模の分布を図3に示す。



図1 組込みシステム開発技術リファレンスESxRシリーズ

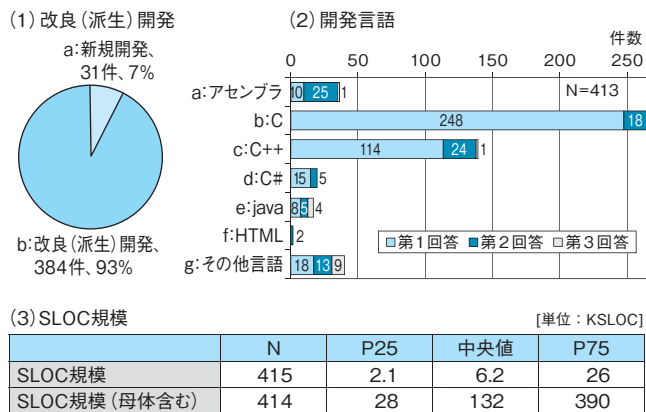


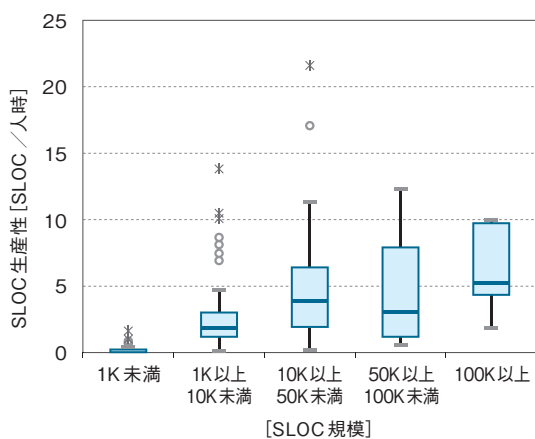
図3 分析対象の範囲

3 組み込みソフトウェア開発全般の特徴

まず、組み込み製品の特性ごとの生産性や信頼性の傾向を見る前に、組み込みソフトウェア開発全体として、生産性や信頼性の特徴を把握する。

(1) 規模別SLOC 生産性

SLOC生産性はエンジニア人が単位時間(または期間)当たりどれだけの行数のプログラムを開発できるのかを表す指標である。生産性を計算する作業対象の工程は、組織やプロジェクトの位置付けにより異なるが、ここでは、アーキテクチャ設計、詳細設計、実装及び単体テスト、結合テスト、総合テストのソフトウェア開発5工程を対象に生産性を計算している。図4から分かるように、生産性は、規模が小さいところでは低く、規模が大きくなるにつれて高くなる傾向が見られる。ただし、ある程度の規模を超えると生産性の伸びは小さくなっていく。



開発5工程 言語C/C++ (改良・派生開発)

SLOC規模	N	P25	中央	P75
全体	130	0.54	1.86	4.35
1K未満	31	0.01	0.01	0.20
1K以上10K未満	48	1.13	1.90	3.06
10K以上50K未満	34	1.97	3.91	6.46
50K以上100K未満	10	1.75	3.18	7.86
100K以上	7	4.34	5.19	9.79

図4 SLOC規模別SLOC生産性

(2) 信頼性

「組み込みソフトウェア開発データ白書」では、信頼性の指標は、結合テストと総合テストにおいて検出するバグ件数をSLOC規模で正規化した「テスト検出バグ密度」で表している。エンタプライズ系のソフトウェア開発データ白書は、出荷後のバグ件数を信頼性の指標値としているが、組み込みシステムの場合、多くは、出荷後にバグを出さないように結合テストと総合テストで品質を高めるため、信頼性指標の定義を変えている。図5に、SLOC規模とテスト検出バグ密度の傾向を示す。バグ密度は、プログラム1,000行(1K)当たりのバグ件数と定義しているため、1K以下の小規模SLOCのプロジェクトでは、バグ1件が密度に大きく影響しばらつきが大きくなる。そのため、「テスト検出バグ密度」は、ある程度SLOC規模が大きいプロジェクトの指標に向いている。図5左側の結合テストのバグ密度は、SLOC規模に関係がありそうな傾向が見えているが、総合テストの場合は、SLOC規模との関係があまりないように見られる。その理由として、総合テストは、デバッグを目的とするのではなく、バグがないことを確認する目的で行っていることが想定できる。

4 組み込み分野とエンタプライズ分野の比較

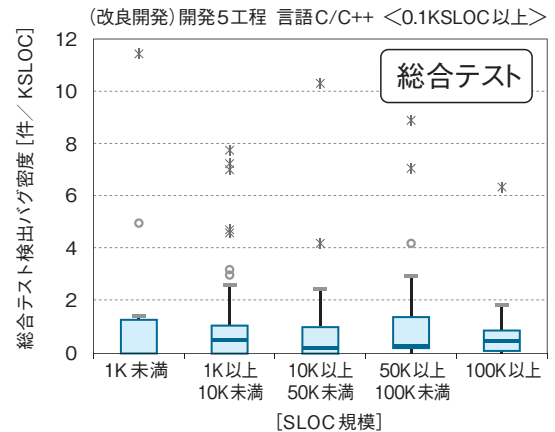
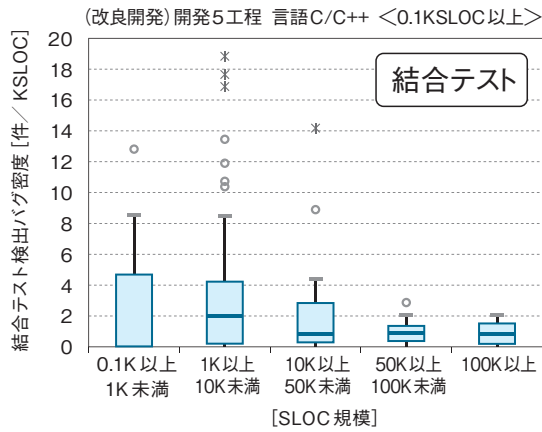
組み込みソフトウェア開発全般の生産性や信頼性の傾向が分かると、次はメトリクスの分析で先行しているエンタプライズ分野と比較して、組み込みソフトウェア開発の特徴を捉える。エンタプライズ分野の「ソフトウェア開発データ白書2016-2017」分析結果と比較して特徴の違いが見えたものを、幾つか紹介する。

(1) 工程別 工数・工期の比率

図6は、ソフトウェア開発のアーキテクチャ設計～総合テストまでの開発5工程の各工程に配分する工数と工期の比率を組み込み分野とエンタプライズ分野で比較したものである。

左上の「組み込み」の工数比率と左下の「エンタプライズ」の工数比率を比べてみると、「エンタプライズ」は、製作工程(組み込みの実装単体テスト工程)に突出して工数をかける傾向が読み取れる。一方、「組み込み」では、実装単体テスト工程に最も工数をかけるという点では、「エンタプライズ」と同じであるが、アーキテクチャ工程やテスト工程にも「エンタプライズ」よりも工数をかける傾向が見える。

次に右側の工期について、「組み込み」と「エンタプライズ」を比べると、「エンタプライズ」では、工数と同様に製作工程に最も長い期間をかけるのに対して、「組み込み」では、実装・単体テスト工程は、リソースを増やすなどして、期間を圧縮している傾向が見える。また、両者の箱ひげ図の高さを比べると、「組み込み」のほうが高さの範囲が広く、中央値の位置も全体的に高くなっていることが分かる。この理由は、「組み込み」の場合、工程の終了と次工程の開始が重なっている標本データが大半を占めるが、「エンタプライズ」の場合は、工程の終了と次工程の開始が重なる標本データはごく少数であるためと分かった。



SLOC規模	N	P25	中央	P75
全体	168	0.140	1.120	3.100
0.1K以上 1K未満	19	0.000	0.000	4.643
1K以上 10K未満	73	0.193	2.020	4.237
10K以上 50K未満	50	0.267	0.837	2.838
50K以上 100K未満	16	0.438	0.841	1.308
100K以上	10	0.182	0.643	1.497

SLOC規模	N	P25	中央	P75
全体	147	0.027	0.331	1.072
1K未満	15	0.000	0.000	1.279
1K以上 10K未満	59	0.000	0.465	1.047
10K以上 50K未満	47	0.054	0.290	1.027
50K以上 100K未満	17	0.199	0.262	1.331
100K以上	9	0.103	0.542	0.799

図5 SLOC規模別のテスト検出バグ密度

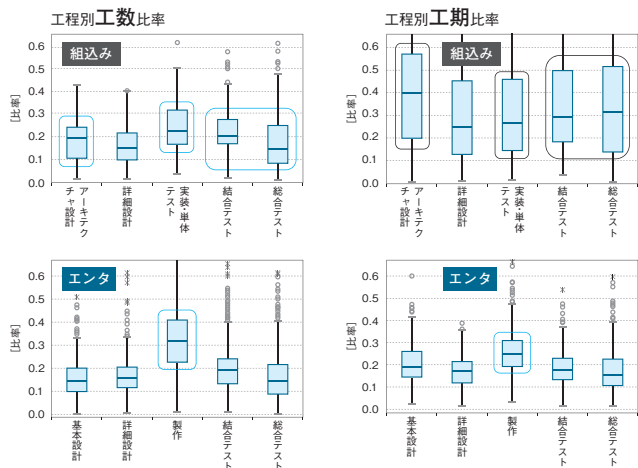


図6 工程別工数・工期の比率

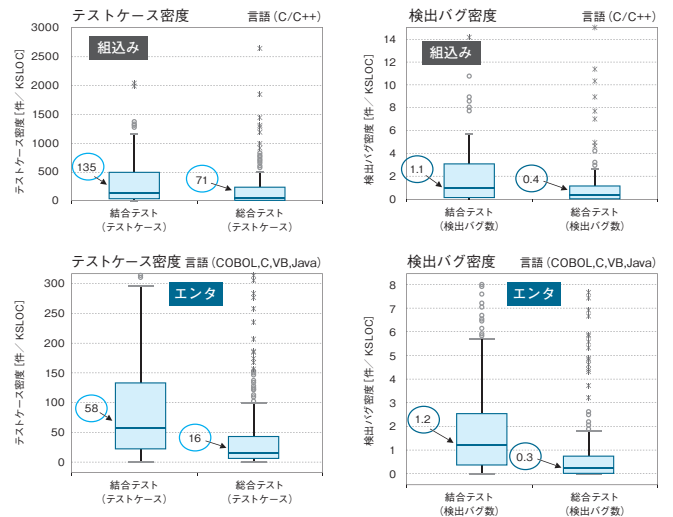


図7 テストケース密度と検出バグ密度

(2) 規模当たりのテストケース数、検出バグ数

図7は、結合テスト及び総合テストにおけるテストケース密度 (KSLOC規模 (母体規模は含まない) 当たりのテストケース数) とテスト実施により検出するバグ密度 (KSLOC規模当たりの検出バグ件数) を「組み込み」と「エンタプライズ」で比較したものである。テストケース密度を「組み込み」と「エンタプライズ」で比べてみると、中央値にて、結合テストで135:58と2倍以上、総合テスト71:16で4倍以上の違いが見られる。

一方で、検出バグ密度 (図の右上:「組み込み」、右下:「エンタプライズ」) では、結合テスト、総合テスト共に、「組み込み」「エンタプライズ」両分野の差異が見られない。

5 製品の特性ごとの分析

2017年版では、製品の特性を表す次の8つの視点で、プロジェクトデータを層別し、生産性や信頼性の指標の傾向を分析している。

- ①リアルタイム性 (時間制約) の強弱
- ②自然環境からの影響度合い
- ③ユーザの多様性
- ④法規などによる規制度合い
- ⑤M2Mの有無
- ⑥ネットワーク接続の有無
- ⑦稼動 (非停止、オンデマンド)
- ⑧オンライン保守の可否

この8つの製品特性のうち、①リアルタイム性の強弱や②自然環境からの影響度合いの違いで、生産性や信頼性の指標値が異なる傾向が見えている。一方、③ユーザの多様性の違いや④オンライン保守の可否では、想定した傾向が見られなかった。これらの特性のうち、②自然環境からの影響度合いと⑧オンライン保守の可否について、分析結果を紹介する。

5.1 自然環境からの影響の有無

屋外で使用し自然環境からの影響を受ける車載機器と、自宅で使用する家電製品のソフトウェアを開発する生産性(1人月でアーキテクチャ設計から総合テストまでの5工程を経て、ソフトウェアを完成させることができるプログラム行数)について、違いがあるとの想定のもと、分析した。

(1) SLOC規模と生産性

図8は、自然環境からの影響を受けるかどうかの視点でSLOC規模と生産性の関係を散布図(対数表示)に表したものである。「3 組込みソフトウェア開発全般の特徴」に示した通り、SLOC生産性はSLOC規模に依存する傾向が見られることから、同じSLOC規模の範囲で生産性を比べてみる。図8のSLOC規模(X軸)10.0~100.0前後の範囲でSLOC生産性(Y軸)を見てみると、自然環境からの影響【あり】◆と【なし】■に比べて、生産性が低くなる傾向が見られる。

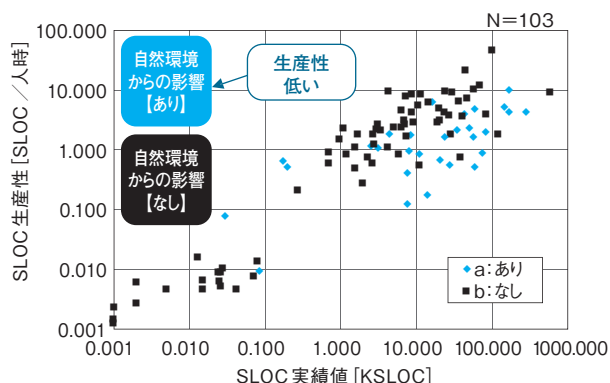


図8 自然環境からの影響を受ける場合のSLOC生産性

(2) テスト検出バグ密度

次に前項(1)で分析した同じ標本について、結合テストで検出したバグ密度と総合テストで検出したバグ密度の分布の違いがあるかどうか調べた。その結果は図9の散布図(対数表示)に示す。

図9左側は、結合テストで検出したバグ密度の分布を自然環境からの影響【あり】◆と【なし】■に層別している。同様に図の右側は総合テストのバグ密度分布を示す。結合テスト、総合テストを比較すると、結合テストのバグ密度は、自然環境からの影響の【あり/なし】の違いは見られないが、総合テストの場合、明らかに自然環境からの影響【あり】◆のほうのバグ密度が高い。その要因を推測すると、総合テストは一般に実際に製品が使用される環境でテストを行うため、自然環境からの影響を受ける製品は、実際の環境で使用して初めて検出される問題が多いことが考えられる。

(3) 工数実績の工程比率

前々項(1)で自然環境からの影響を受けるほうが、生産性が下がる傾向が見られたが、どの工程に工数がかかるのかを調べるために、工数実績の工程比率を図10に示す。

自然環境からの影響【あり】と【なし】の工程比率を見比べると、総合テストの工程比率は、【あり】で0.25、【なし】で0.13であり、生産性を下げる要因は、総合テストにかける工数の割合が高くなるためと判明した。これは、総合テストで検出するバグが多いことで作業が増えることを裏付けている。

5.2 オンライン保守の可否

スマートフォンは、アプリの更新ができる。出荷後にプログラム更新が可能な製品は、出荷後に不具合が見つかったもソフトウェア更新が行えるため、コストや納期を優先する場合は、適切な品質が確保されたと判断できた時点でテストを終了していると仮定した。そのため、オンライン保守が可能な製品のテスト検出バグ密度は、オンライン保守が不可能な製品に比べて、違う傾向が見られるはずだと想定し、分析を行った。

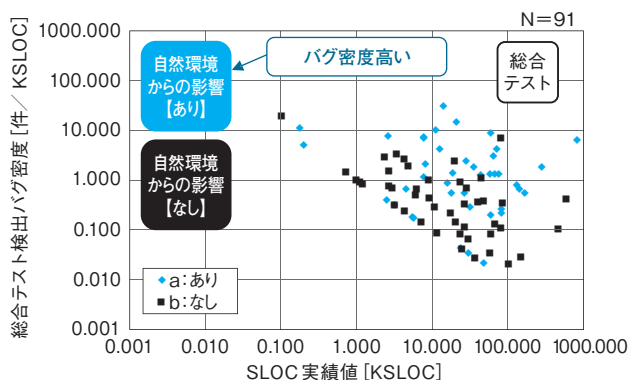
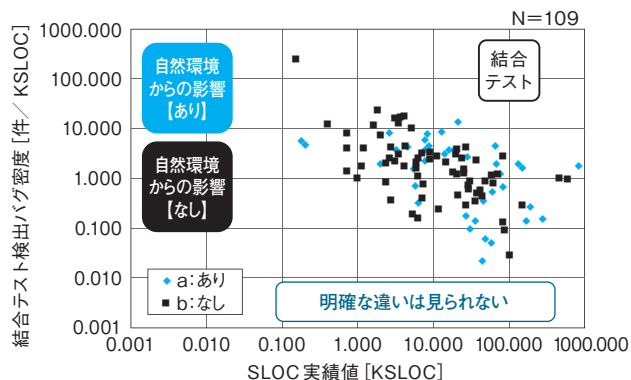


図9 自然環境からの影響を受ける場合のテスト検出バグ密度

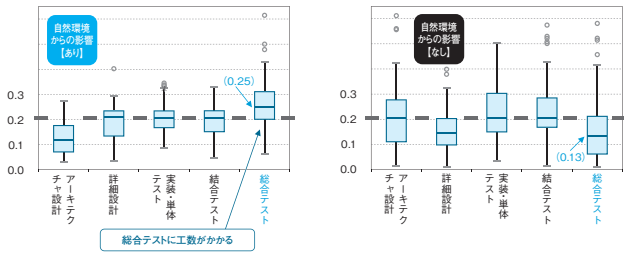


図10 自然環境からの影響を受ける場合の工数実績の工程比率

(1) SLOC規模と生産性

図11は、オンライン保守が可能かどうかの視点でSLOC規模と生産性の関係を散布図(対数表示)に表したものである。【可】◆の分布は【否】■に比べて生産性が高い傾向が見られる。その要因をテスト工程にかける作業負荷が小さいと想定し、次の(2)テスト検出バグ密度と(3)工数実績の工程比率を分析した。

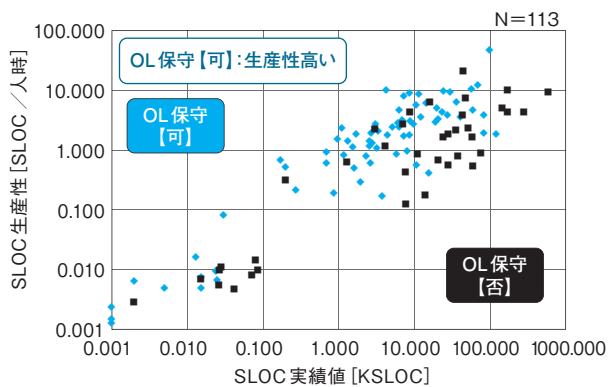


図11 オンライン保守が可能な場合のSLOC生産性

(2) テスト検出バグ密度

前項(1)で分析した同じ標本について、結合テストで検出したバグ密度と総合テストで検出したバグ密度の分布に違いがあるかどうか調べた。その結果は図12の散布図(対数表示)に示す。

【可】◆と【否】■の分布を結合テスト、総合テストで比べてみたが、仮定に反して、テスト検出バグ密度に明確な違いは見られない。

(3) 工数実績の工程比率

(1)のオンライン保守【可】のほうが生産性が高いという結果から、どこかの工程にかける工数が【可】のほうが高い傾向が見られることを想定したが、図13からは、明確な違いは見られなかった。

以上、「自然環境からの影響度合い」と「オンライン保守の可否」について、生産性や信頼性指標の傾向を分析した結果を紹介したが、そのほかの特性については、「組込みソフトウェア開発データ白書2017」に分析結果を掲載しているので、参照していただきたい。

2017年版でようやく製品の特性別の分析を行って、傾向の違いが見られるようになったが、傾向の違いの要因や想定通りの傾向が見られなかった特性については、更に標本データを収集して、引き続き「製品・制御システム定量データ収集・分析WG」の中で、その要因分析を進めていく。

開発現場の肌感覚が定量的に示されるようになると、「見えない」と言われがちな組込みソフトウェア開発のハードルが下がると考えられる。更に、開発目線のみならず、ビジネス目線においても、リスクの早期発見や低減に利用されることを期待したい。

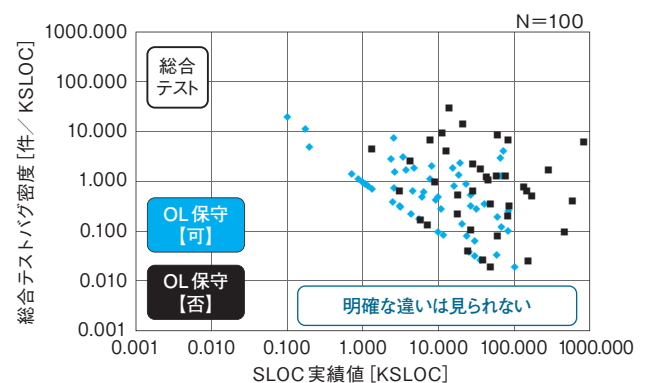
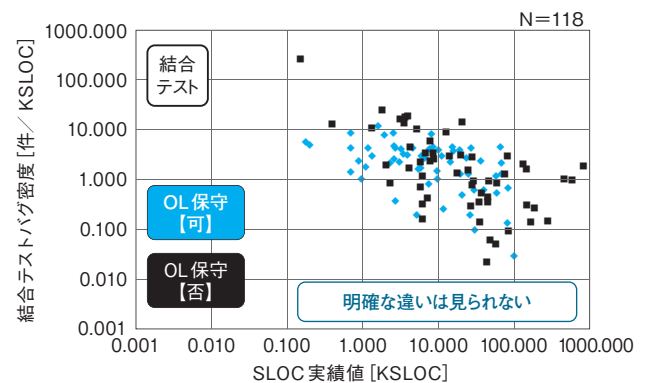


図12 オンライン保守が可能な場合のテスト検出バグ密度

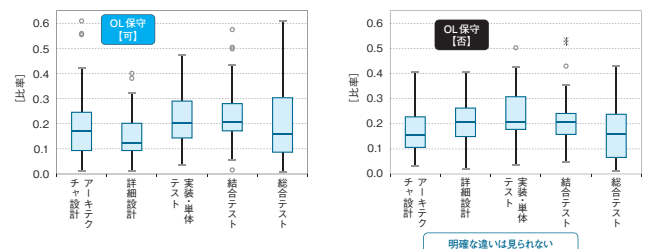


図13 オンライン保守が可能な場合の工数実績の工程比率