

CEATEC JAPAN

2006年度活動報告

組込み系ソフトウェア開発の 課題分析と提言

～ 大規模化、短納期化、多機種開発にどう立ち向かうか～
(JEITA活動報告)

2007年10月2日

社団法人
電子情報技術産業協会
ソフトウェア事業委員会
ソフトウェア事業基盤専門委員会

1. 複雑化するソフトウェア事業環境

- ハードウェア (メインフレーム、サーバ、PC) の日本国内出荷実績：
2兆6,500億円余り (JEITA 2005年度統計)
- 日本国内主要ベンダによるソフトウェア・サービス
(システム開発、ミドルウェア、アウトソーシング等) の市場規模：
5兆3,000億円余り (JEITA 2005年度統計)
- 組込みソフトウェア開発規模 (内製を含む)：
2005年度 約2.7兆円 (経済産業省推定)
- ソフトウェア事業の事業基盤の複雑さが益々増大：
 - オープン化の進展
 - IT とネットワークの融合の進行
 - オープンソースソフトウェア活用の高まり
 - オフショア等のソフトウェアリソースの新たな段階への対応 等々
- 情報端末機器、デジタル家電、携帯電話、カーエレクトロニクス等を中心に、組込み系ソフトウェアの領域・開発規模・開発速度が急激に増加。
- 組込み系ソフトウェアが事業全体の成否を決める鍵となりつつある。
組込みシステムの開発費の40%が組込みソフトウェア開発費

2. ソフトウェア事業基盤専門委員会

■ ソフトウェア事業委員会：

- ソフトウェア業界の共通課題を、ソフトウェア事業の視点と日本の産業力強化の観点から検討
- 2005年4月 活動開始

□ 3つの専門委員会：

ソフトウェア事業戦略専門委員会：

- ・ソフトウェア産業の国際競争力向上と市場活性化のための課題

ソフトウェアリソース対応専門委員会：

- ・ソフトウェア開発に必要な要員リソースに関する諸問題

ソフトウェア事業基盤専門委員会：

- ・ソフトウェア事業運営の基盤を確立・強化する共通的な課題

我が国の強みの源泉・価値創出のキーと言われる
「組み込み機器のソフトウェア(組み込み系ソフトウェア)」に焦点
この分野での開発力強化等を図るための調査活動を実施

2.1 組込み系ソフトウェアに関する活動

- 2005年度の活動：
 - 組込み系ソフトウェア開発に関する基本的な実態調査とその分析と提言
 - 品質確保
 - 外部委託活用
 - OSS(オープンソースソフトウェア)
- 2006年度の活動：
 - 品質確保の問題に焦点を絞り込み、アンケート調査を実施し、実態把握、課題分析、課題解決に向けての提言：
 - 品質の計測と管理の仕組みとしての「品質施策」
 - 大規模化/多機種開発/システム化を見据えた「品質プロセス」
 - ドイツ・フラウンホーファー協会IESE(実験的ソフトウェア工学研究所)と2回に渡るディスカッションを実施(6月、11月)
 - ソフトウェア工学分野の研究で欧州No.1との評価
- 2007年度の活動：
 - 課題解決に向けた先進的事例・成功事例の調査・収集
 - その一環として・・・IESE / JEITA共同ワークショップ開催

3. 組み込み系ソフトウェアに関する問題認識

- 「組み込み系ソフトウェアは我が国の強みの源泉であり価値創出のキー」と言われているが、組み込み対象となるハードウェア機器は強いとしても、そのソフトウェア開発力が国際的に見ても本当に強いのだろうか。
- 「擦り合わせ」なるものが日本の開発力の強みと言われているが、急激に増大している開発規模や短納期化の状況の中で、現在でもそれが強みになっているのだろうか。
- 何を強くすれば、我が国の組み込み系ソフトウェア開発力の国際競争力を強化し、真に「我が国の強みの源泉」たりうるものにできるのだろうか。

3.1 品質施策

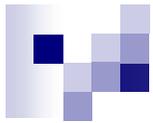
- 大規模化、短納期化、多機種開発等の組み系システム開発の環境変化に対して、品質施策の面で対応できているのだろうか
 - 品質施策全般
 - 目的や投資対効果を明確にして運用しているか
 - 品質施策の標準化は行われているか
 - 具体的な品質施策や利用ツールについても調査
 - 特に、品質計測に重点をおいて
 - 開発規模の増大に対する計測対象や計測工程の変化
 - 外部委託企業との品質計測の連携
 - 短納期化に対する品質計測の効率化
 - 多機種開発に対する品質計測の共通化や水平展開

3.2 品質プロセス

- 大規模化、短納期化、多機種開発、システム化に対応すべき組込み系システム開発での品質プロセスはどうなっているのか
 - 大規模化・短納期化
 - 大規模なソフトウェアに対処するための開発プロセス、
 - 上流工程での品質の作りこみの施策
 - ソフトウェア全体を俯瞰する技術、ソフトウェアアーキテクトの育成方法
 - 多機種開発
 - 差分開発の状況や再利用の方法
 - プロダクトラインエンジニアリングの日本における取り組み状況
 - システム化
 - ハードウェアとソフトウェアの擦り合わせの実態と方法
 - 「メカニクス/エレクトロニクス/ソフトウェアの3層化されたV字モデル」
「システムとしての品質を段階的に検査する品質ゲートモデル」も意識

4. アンケート調査の概要

- 組込み系ソフトウェアの開発に関する以下の設問で構成：
 - 回答対象プロジェクトの概要に関する設問(23問)
 - 品質施策の課題に関する設問(24問)
 - 品質プロセスの課題に関する設問(26問)
 - 2005年度アンケート調査に比べ、意識的に自由記述回答の設問を増し、できる限り開発現場の生の声を調査結果の反映するようにした。
- アンケート調査期間：2006年10月20日～11月上旬
- アンケート調査対象プロジェクトの範囲：
 - 情報家電機器、携帯機器、自動車や産業機器等の組込みソフトウェアが搭載された機器(組込み機器)を開発するプロジェクト
 - 組込み機器に搭載されるソフトウェア(組込みソフトウェア/OS/ミドルウェア等)
 - 10名以上の要員規模の開発プロジェクト
(新規開発、既存製品の改良、機種展開等)
- アンケート調査の回収結果：
 - 回答社数：JEITAの関連委員会参加企業32社
 - 回答プロジェクト数：59プロジェクト



5. 回答プロジェクトの概要

5.1 製品分野、プロセッサ種別、組込みOS

■ 対象とする製品分野

- AV機器 (TV、DVD、デジタルカメラ等) : 20プロジェクト
- コンピュータ周辺機器/OA機器 (プリンタ、スキャナ、複写機等) : 16プロジェクト
- 通信端末機器 (固定電話機、携帯電話等) : 6プロジェクト
- その他 : 17プロジェクト

■ プロセッサ種別

- 「SH」「MIPS」「x86」「ARM」「PowerPC」など種々のプロセッサを対象

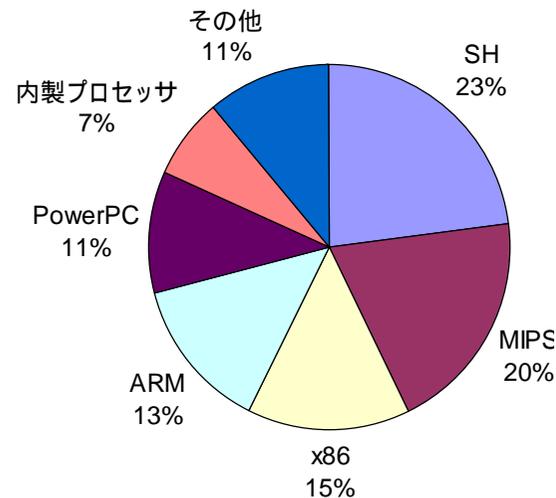


図5.1 対象プロセッサ

■ 組込みOS

- 「μITRON」、「Linux」、「VxWorks」の順に多い
- 2005年度に比べて、Linuxの割合が増加
- Windows系も合わせて15%と2005年度より増加

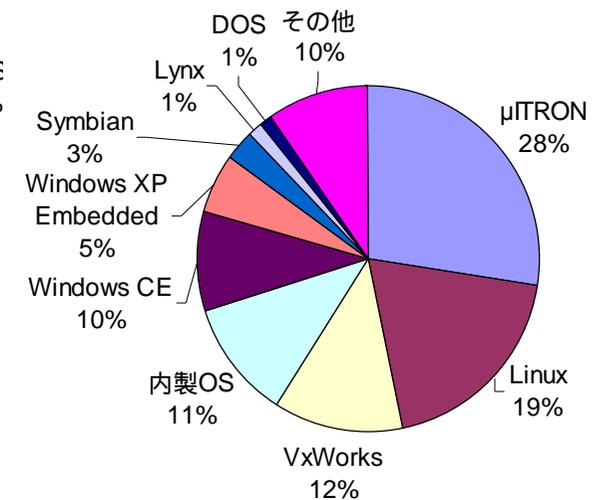


図5.2 対象とする組込みOS

5.2 開発言語、開発規模、開発期間

■ 開発言語

- 2005年度と比べて、Cの比率は同程度、C++が微増、アセンブラが微減

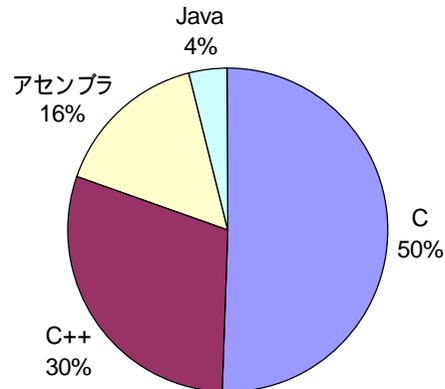


図5.3 開発言語

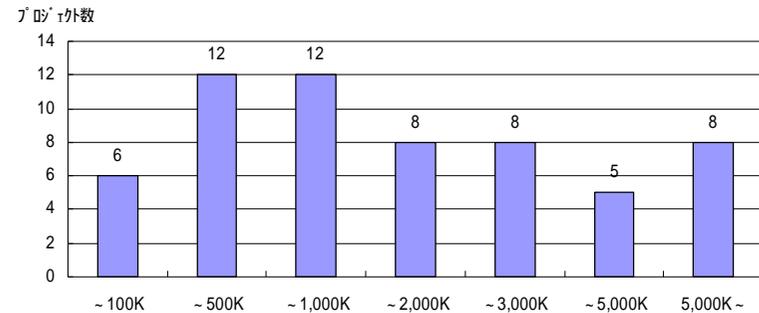


図5.4 開発規模(完成規模)

■ 開発規模

- 約半数(49%)のプロジェクトが1,000K SLOCを超える規模
- 5,000K SLOCを超える規模のプロジェクトが8プロジェクトもある

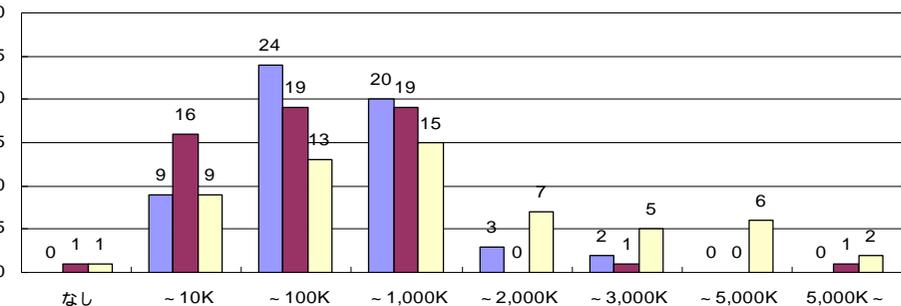


図5.4 開発規模(新規作成、変更、流用)

■ 開発期間

- 回答プロジェクトの3/4近くが1年以下の開発期間

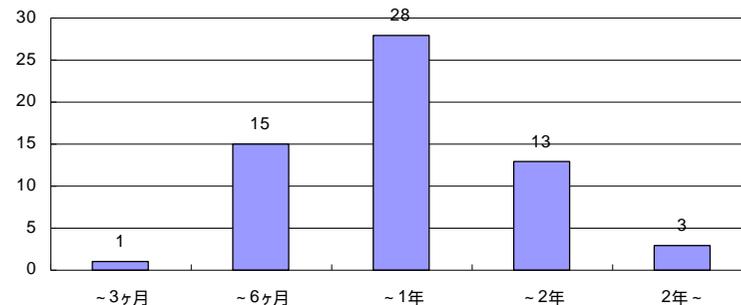


図5.5 開発期間

5.3 開発対象製品数、対象ハードウェアの新規性

■ 開発対象製品(機種)数

- 回答プロジェクトの8割近くが2機種以上のハードウェアを対象
- 回答プロジェクトの2/3近くが2機種以上のソフトウェアを開発
- 1プロジェクトで複数機種のソフトウェアを開発する多機種開発の実態が浮き彫りに

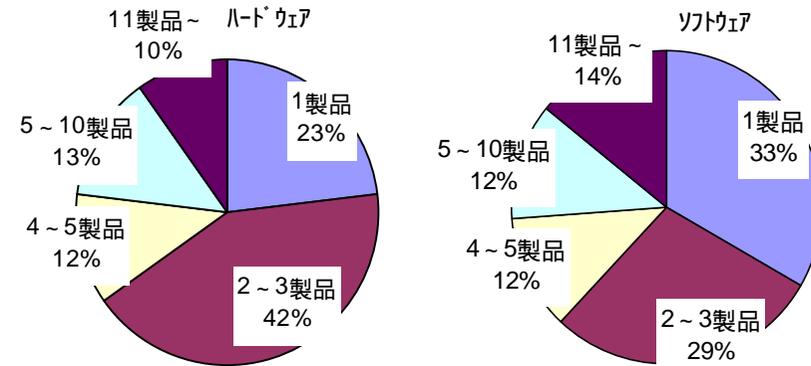


図5.6 開発対象製品(機種)数(ハードウェア、ソフトウェア)

■ 対象ハードウェアの新規性

- 新規開発:改造:市販品:その他 = 39% : 45% : 16% : 0%
- 新規開発 + 改造が主要な部分を占め、独自のハードウェアが依然として差別化要因となっている

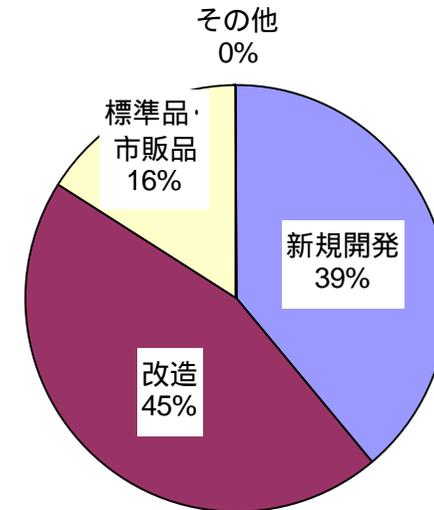


図5.7 開発対象ハードウェアの新規率

5.4 投入要員数、投入工数比率、分野別技術者構成

■ 投入要員数

- 回答プロジェクトの85%が100名以下の要員
- 15プロジェクト(26%)でピーク時要員が100人超

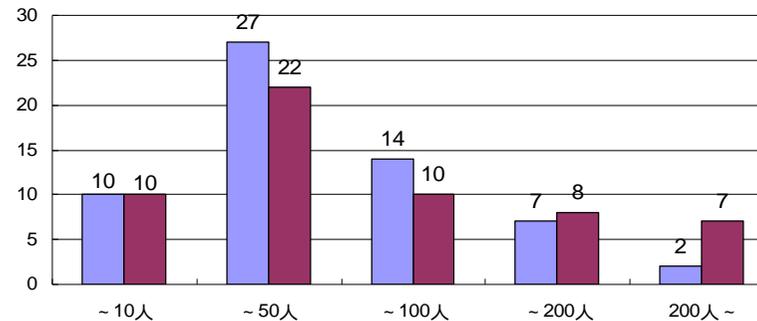


図5.8 開発プロジェクトの平均要員数、ピーク時要員数

■ 工程別投入工数比率

- 設計:実装:テスト:保守=3:3:3:1
- 保守工数が小さく、テスト工数が大きめ

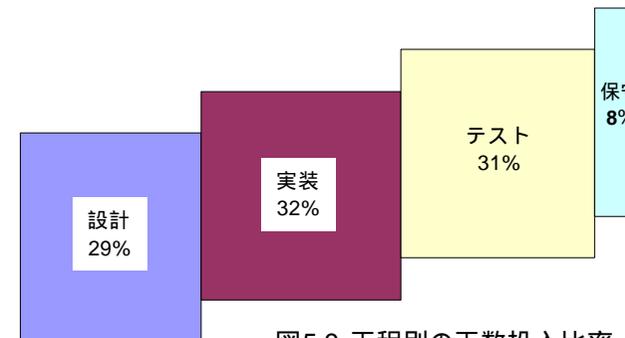


図5.9 工程別の工数投入比率

■ 分野別技術者構成

- 回答プロジェクトでは、システム全体の開発要員の2/3がソフトウェア技術者

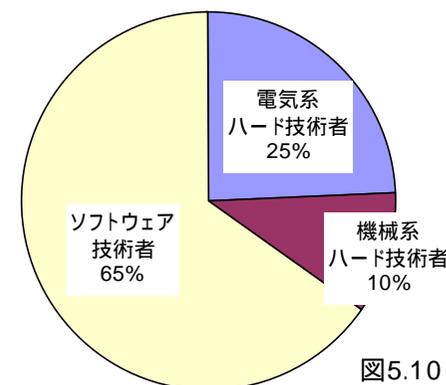


図5.10 分野別技術者構成

5.5 外部委託工数の利用率、外部委託工程

■ 外部委託工数の利用率

- 約7割(69%)のプロジェクトで、全体工数の25%以上を外部委託に依存
- 全体工数の50%以上を外部委託に依存しているプロジェクトが、回答プロジェクトの45%に及んでいる

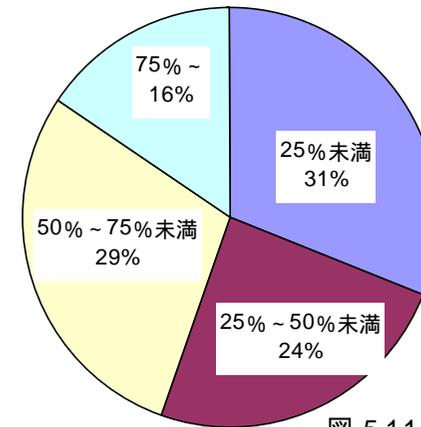


図 5.11 外部委託工数の利用率

■ 外部委託工程

- コーディング・デバッグからテストに至る 下流工程の外部委託が大きな割合を占めている
- 仕様設計、基本設計などの 上流工程にも外部委託が採用されている。
- 2005年度と同様の傾向だが、テスト仕様設計工程での使用比率が6%から10%に増加。上流工程へのシフト傾向とテスト工数増大に対する施策の影響が出ている可能性も考えられる

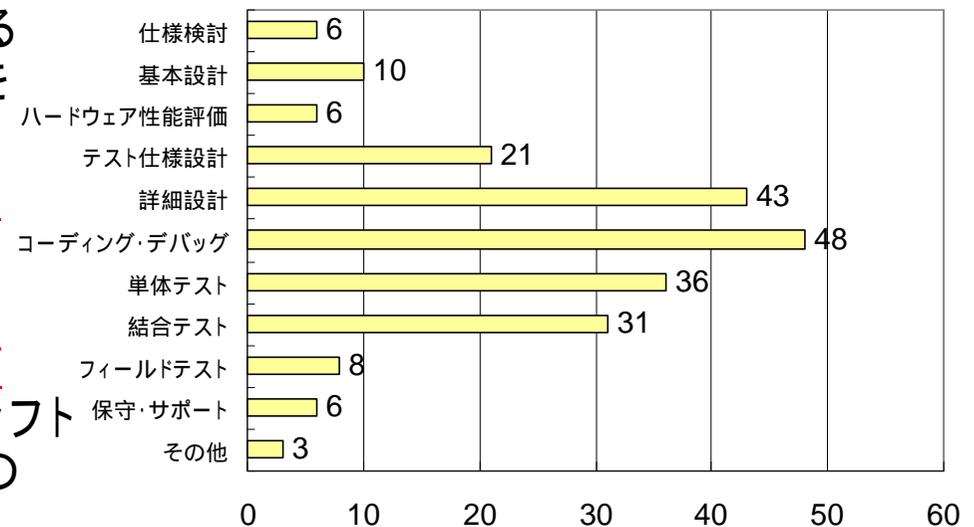


図 5.12 外部委託する工程

5.6 外部委託先、海外の外部委託先

■ 外部委託先

- 外部委託先の7割が国内企業
- 海外企業への外部委託は2割程度
- グループ会社でない受託開発会社への外部委託が5割

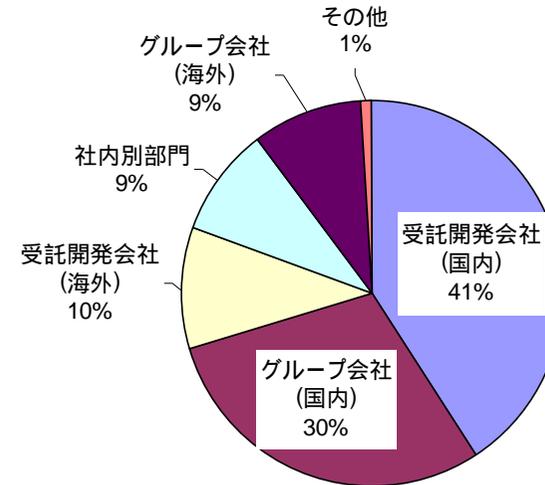


図5.13 外部委託先比率

■ 海外の外部委託先

- 中国 (12プロジェクト) が最多、次いで北米 (6プロジェクト)、インド (5プロジェクト)
- 北米、西欧への外部委託は、コア技術の開発委託の可能性がうかがえる

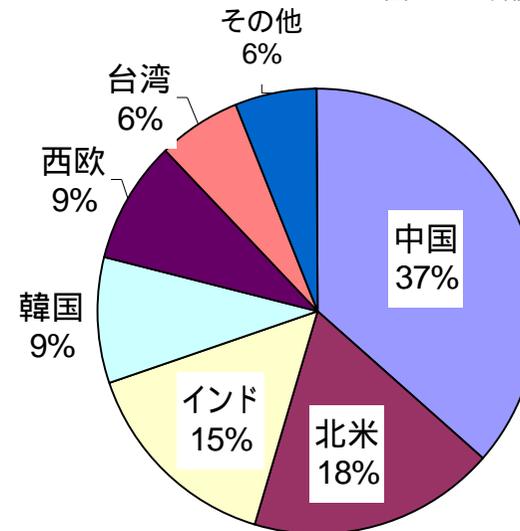


図 4.14 海外の外部委託先

5.7 OSSの利用状況

■ OSSの利用状況

- 製品での利用は 32%、2005年度より増加傾向
- 開発環境での利用も含めると 52% になる
- 製品で利用している大半のプロジェクトでは、全プログラム規模の10%程度に利用
- 全プログラム規模の60%超にOSSを利用しているプロジェクトもある

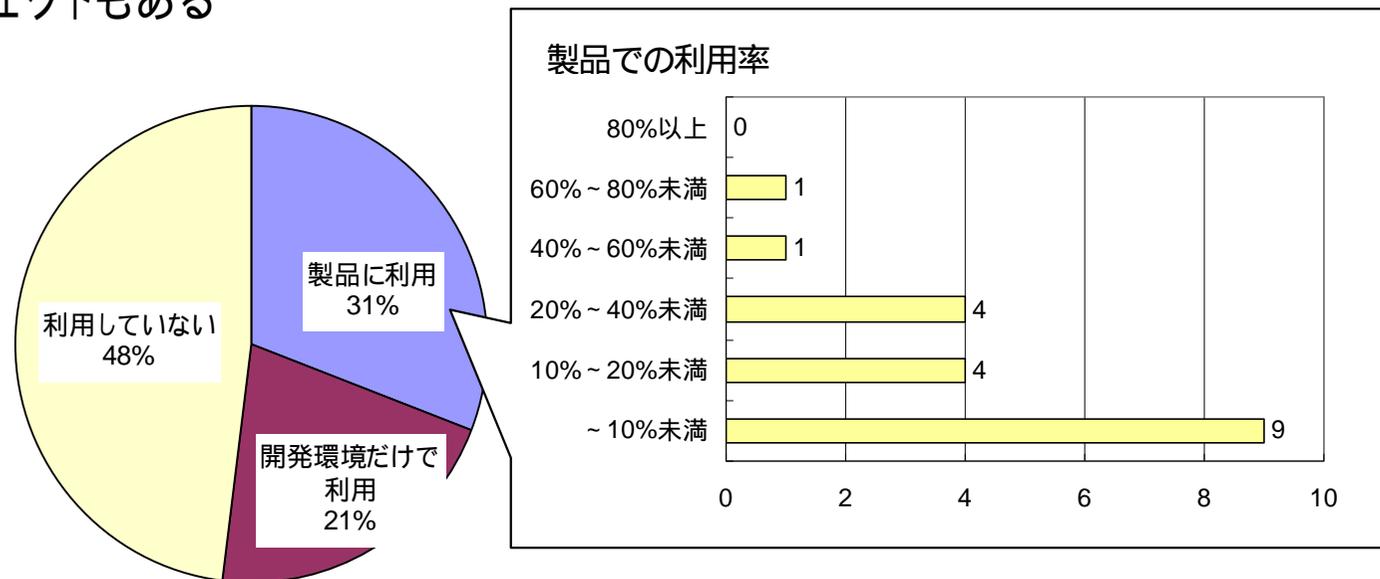


図5.15 OSSの利用状況

5.8 品質状況、品質要求レベル、品質問題の重要度

■ 品質状況

- 回答プロジェクトの半数強(54%)で、出荷後に重度障害を検出、依然として品質確保困難

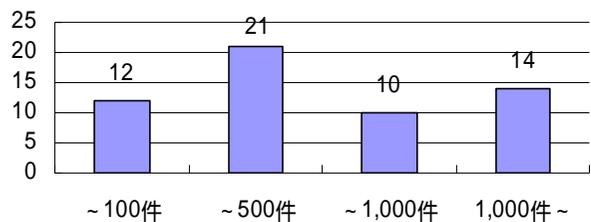


図5.16 出荷前障害件数

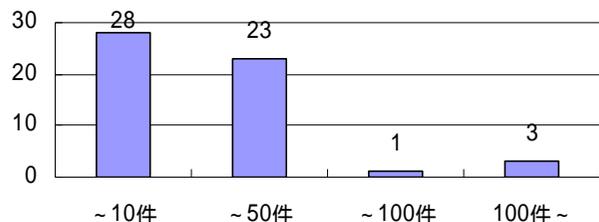


図5.17 出荷後障害件数(軽度)

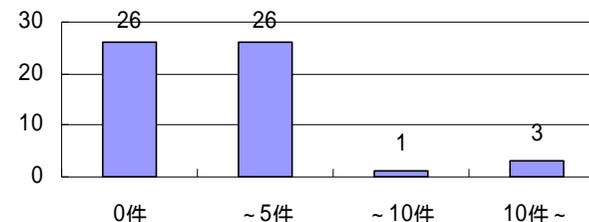


図5.18 出荷後障害件数(重度)

■ 品質要求レベル

- 回答プロジェクトの2/3で「最重要」、「重要」レベル

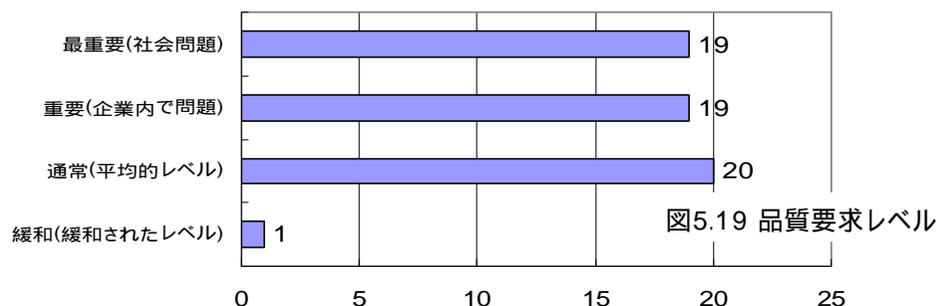


図5.19 品質要求レベル

■ 品質問題の重要度

- 回答プロジェクトの9割がビジネス上の大きな課題と認識
- スケジュール遅れ、競争力低下、コスト上昇、ブランド価値低下

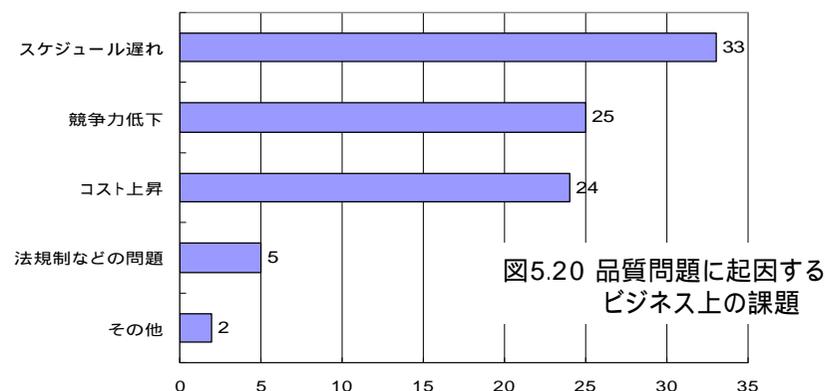


図5.20 品質問題に起因するビジネス上の課題

5.9 品質の要因、重視する品質項目

■ 品質の要因

- 最も重要な要因は「大規模化」
- 次に重要な要因は「短納期化」

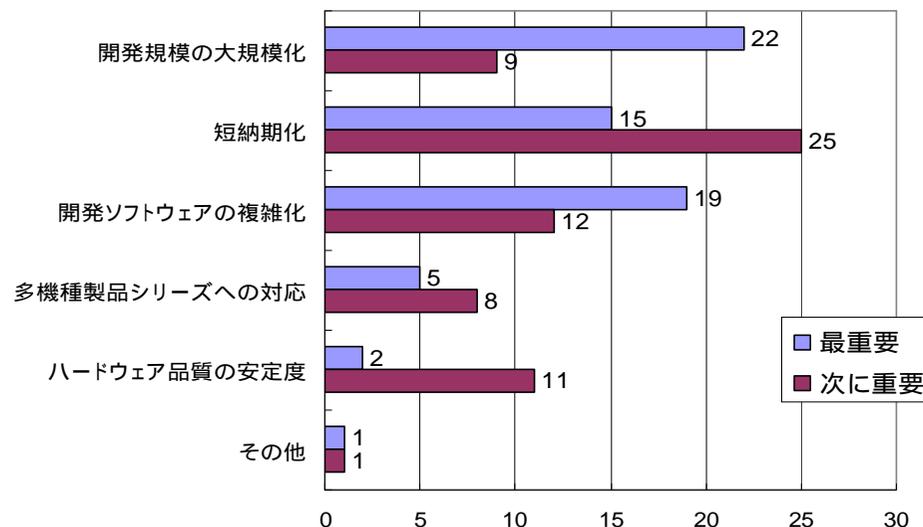


図5.21 品質の要因

■ 重視する品質項目

- 最も重視する項目は「機能安全性」
- 次に重視する項目は「ユーザビリティ」
- 機能保証は当然となり、さらにその先の安全性、ユーザビリティに重点がシフト

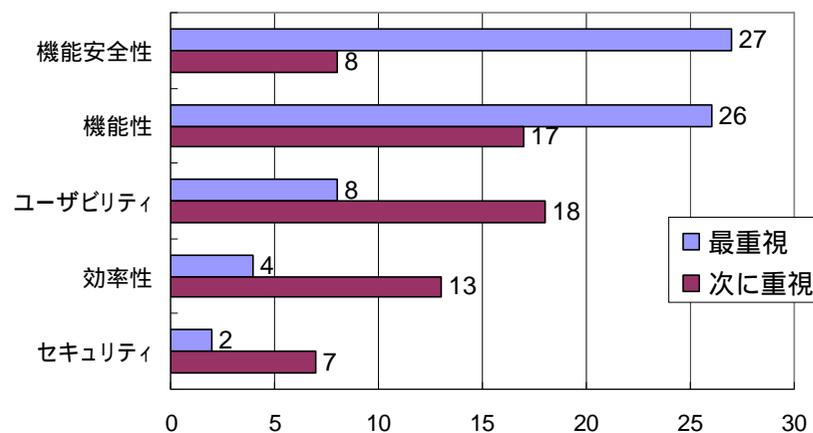


図5.22 重視する品質項目

6. 品質施策の状況と課題分析

■ 品質施策は下流工程から上流工程へ

- 要求分析・設計工程重視へ
 - 現状は下流(試験工程重視・不具合情報)中心
- 品質計測は、リスク対策を目的として一定のコストを掛けている
 - 生産性計測とは連携していない
- 仕様変更の計測は半数以下で、品質との関係を重視していない
- 多機種開発に対する施策が弱い

■ 具体的品質施策

- ツールの利用状況
 - コードメトリクスツールの使用は多いが、その他のテストツール使用は少ない
 - プロジェクト管理ツールや構成・版数管理ツールの使用は多い
- テスト項目数の削減により、品質施策の実施を効率化している
- コーディング規約など、標準化の重要性は認識されている
 - 外部委託先との共有も進んでいる

6.1 品質計測 (1)

■ 重点は下流工程から上流工程へ

- 設計工程は現在も将来も重視する
 - 現在71%、将来76%
- 品質施策は要求分析工程重視へ
 - 現在13%、将来21%
- システム試験重視から当たり前の段階へ
 - 現在58%、将来29%
- 品質計測実施は下流(試験工程)中心
 - 設計46%、試験90%で計測実施
 - 設計工程は重視するが、品質計測は未実施が多い

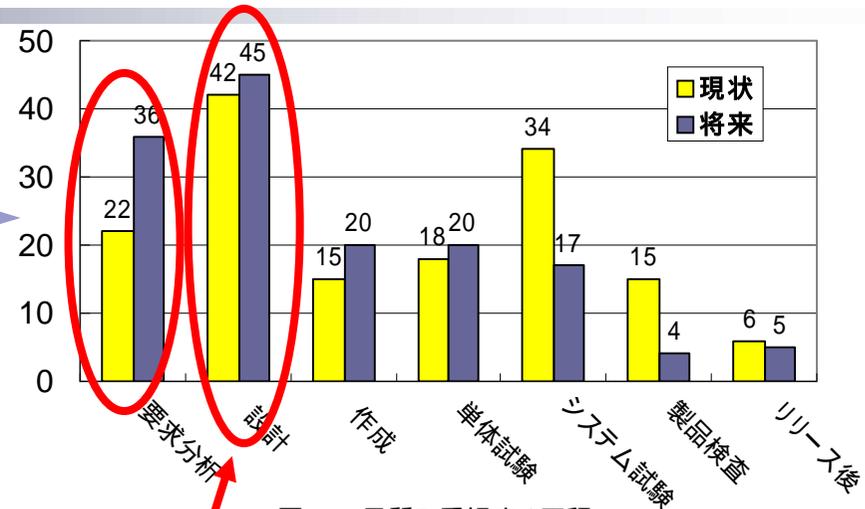


図6.1 品質で重視する工程

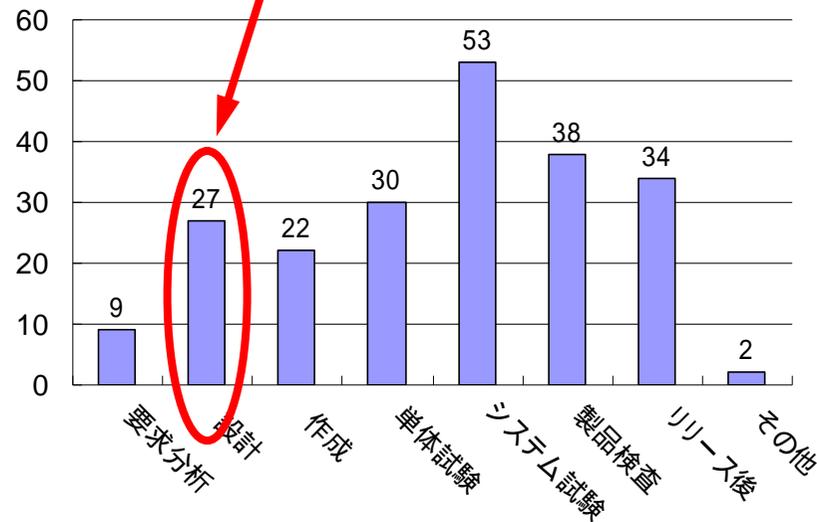


図6.2 品質計測をしている工程

6.1 品質計測 (2)

- 計測は一定のコストを掛け、その目的はリスク対策
 - 計測コストは一定量掛ける
 - 半数が全体の1～5%の計測コストを掛ける
 - リスク対策として実施
 - リスク対策66%
 - 生産性計測と連携していない 84%

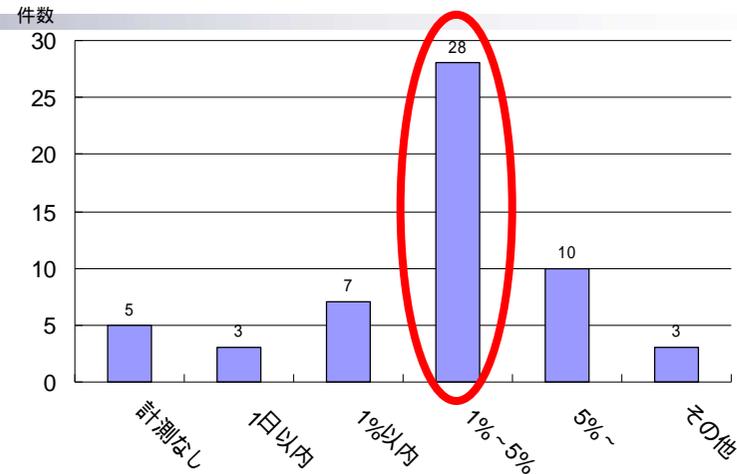


図 6.3 品質計測のコスト

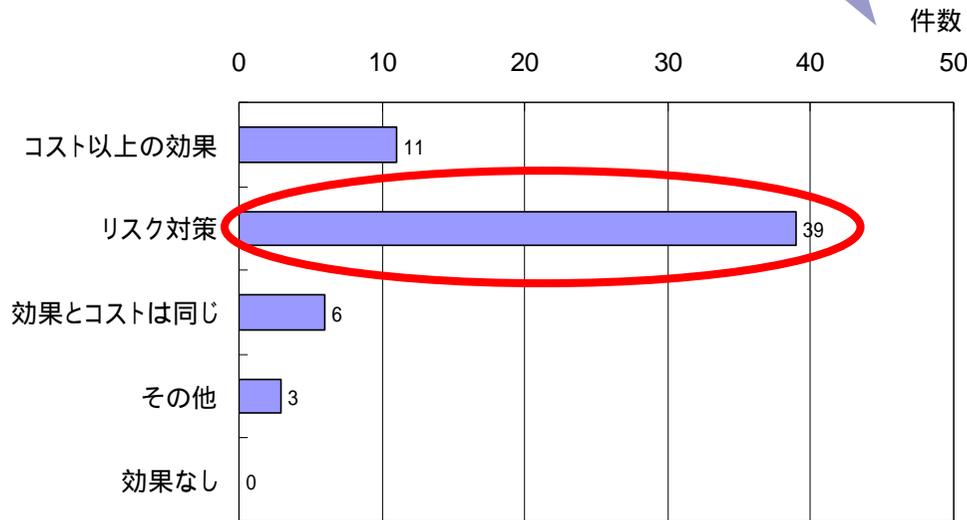


図 6.4 品質計測のコスト効果

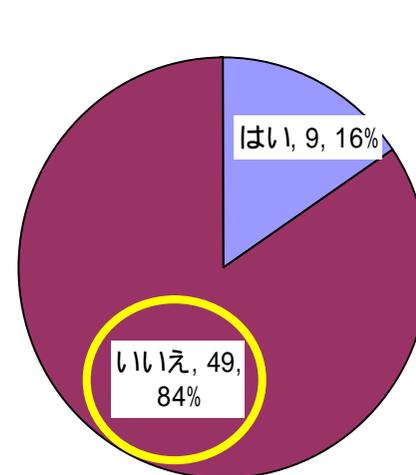


図 6.5 生産性計測との連携

6.1 品質計測 (3)

- 仕様変更の計測意識不足 40%で未実施
 - 品質に与える影響をあまり考慮していない
- 不具合情報と他の情報との連携は将来に
 - 現在 22%、将来51%

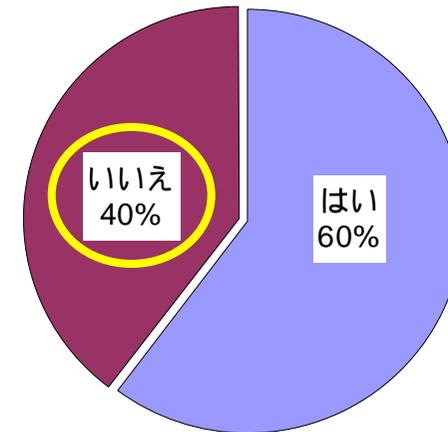


図6.6 仕様変更の計測

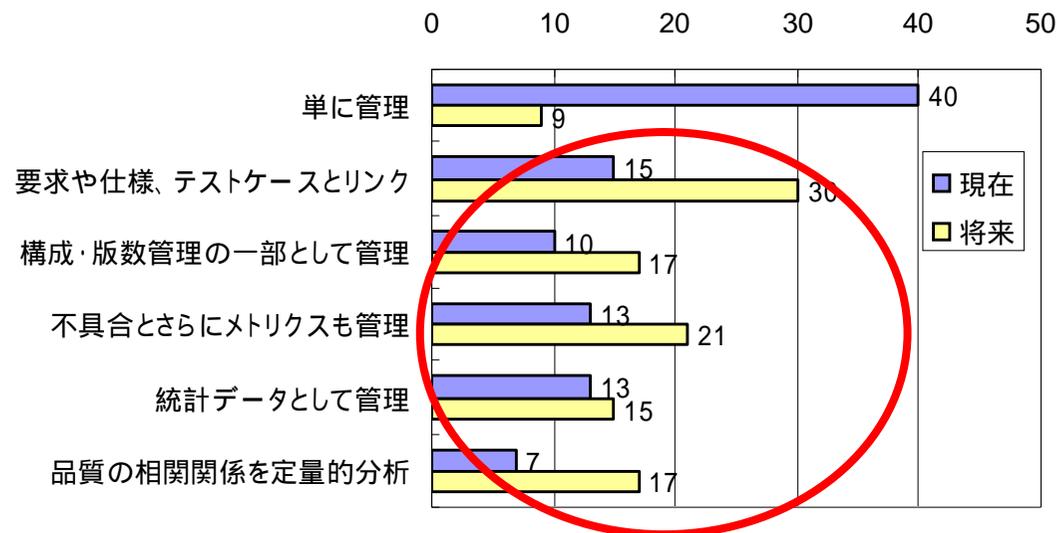


図6.7 品質計測の手法

6.2 品質計測の傾向

■ 品質計測の傾向

品質計測の調査で、選択回答が多かった(50%以上)主な項目

- 計測対象は不具合情報
- 単体試験～リリース後の各工程で品質を計測
- リスク対策のため、全体の1%～5%のコストをかけて計測
- 不具合情報として、作り込んだ工程、発見したバージョン、対策内容を記録
- 将来は仕様やコード、テストケースと連携させて不具合情報を管理
- 品質施策として重視している工程は、現在は設計とシステム試験の工程で、将来は要求分析と設計の工程を重視する

6.3 具体的品質施策 (1)

■ ツールの利用状況

- コードメトリクスツールを35%が使用
- 独自ツールを27%が使用
- それ以外のテストツールの使用は少ない
- プロジェクト管理ツールは52%が使用と多い
- 構成・版数管理ツールは76%が使用と多い

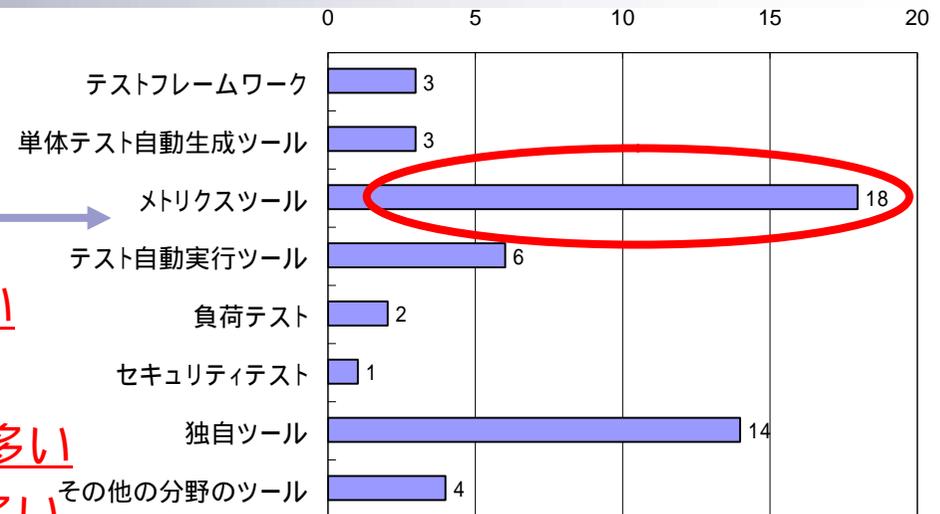


図6.8 テストツールの使用

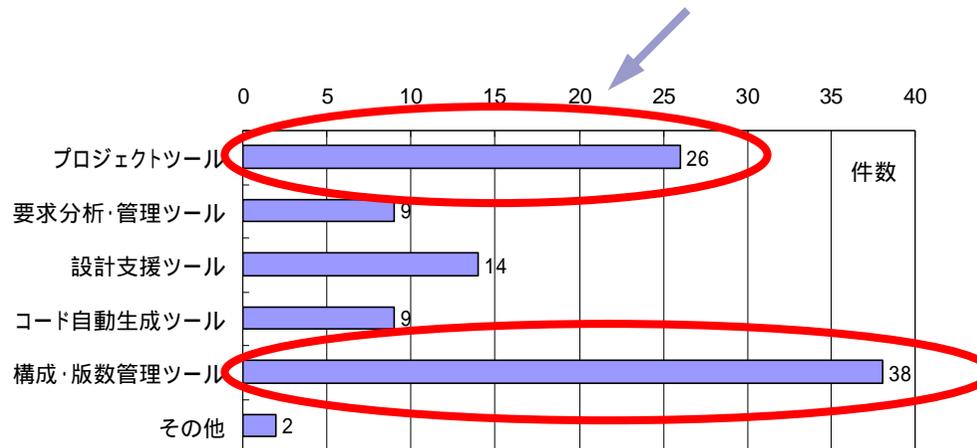


図6.9 その他のツールの使用

6.3 具体的品質施策 (2)

■ 品質施策の実施効率化

- テストケースの項目数削減は高い実施率
- 標準化の重要性を91%が認識
 - コーディング規約は品質課題に有益
 - 外部委託先と規約共有は41%
- 複数同時並行開発のためには施策を水平展開
- 製品全体の品質向上にはプロセス改善

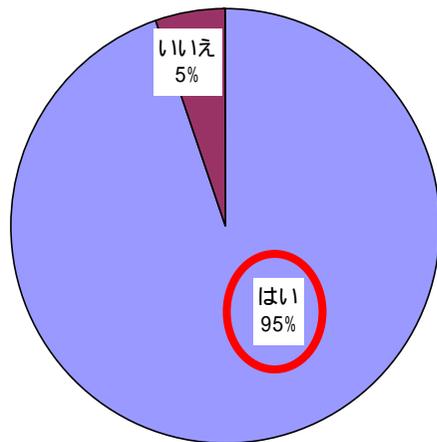


図6.11 品質に関する標準化が課題解決に役立つか

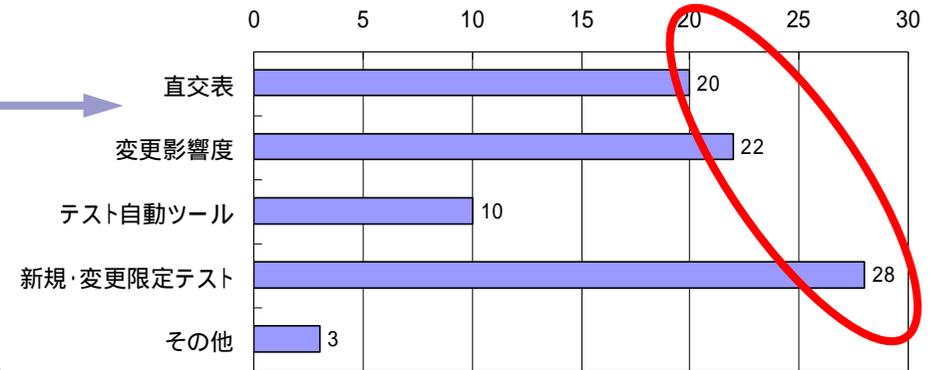


図6.10 テストの効率化施策



図6.12 外部委託先と共有する品質施策

6.4 具体的品質施策の傾向

■ 具体的品質計測の傾向

具体的品質施策の調査で、選択回答が多かった(50%以上)主な項目

- 品質の計測はExcelや独自ツールで行う
- 品質の計測と分析はともに開発メンバが行っている
- 異常データは通常データと同様に扱う
- 構成・版数管理ツールを使用
- テスト品質向上のために、設計者とテスト担当者との相互レビューを実施
- テスト品質向上のために、新規・変更部分のテストケース作成に留意
- 大規模化・複雑化のため、テストは十分とは言えない
- 品質施策に関する標準化は重要で、外部委託先と共有している

6.5 品質施策に関する提言

- コストと効果を考慮した品質施策
 - 効果に見合ったコストで品質施策を実施する
 - 明確なゴールを設定して、効果を検証しながら実施する
 - プロジェクトの規模に応じたツールを積極的に利用する
- ソフトウェア開発の上流工程へのシフト
 - 品質施策は、効果が大きい上流工程へシフトする
 - 上流工程での品質の定量化とその計測および分析が重要
- 多機種開発への適用
 - 多機種開発を前提として、各プロジェクトへの水平展開を促進するツールを利用する
- 品質関連標準化の共有
 - 品質施策に関する手法やツールは、費用対効果の面からも、外部委託先と共有する

7. 品質プロセスの状況と課題分析

■ ソフトウェアの大規模化

- 開発体制や開発プロセスの整備が追いついていない
- 設計の文書化 / モデル化を行い、上流工程を重視していく方向である
- ソフトウェア全体を統括する仕組みが弱く、人を介したコミュニケーションが中心で、開発手法として定式化されていない

■ 多機種開発

- 従来型のウォーターフォールプロセスの採用が多く、もはや実態に合わないプロセスを、現場開発者が適時、工夫変更しながら開発を続けている
- プロダクトラインエンジニアリングという体系的な技法には期待が大きいですが、本格的な導入はまだこれから

■ システム化と擦り合わせ

- ハードウェアの仕様決定時期は開発の後半でも発生している
- 仕様や、ハードウェアとソフトウェア双方に関わる活動の計画策定と管理もハードウェアが主導している
- ソフトウェア開発を配慮した形の、ハードウェアとソフトウェアの統合開発プロセス・開発方法論が必要となっている

7.1 ソフトウェアの大模化（1）

■ 大規模化の状況とその対策

- 88%のプロジェクトで規模増大傾向
- プロジェクト管理手法の導入には効果もあるが、課題もあり

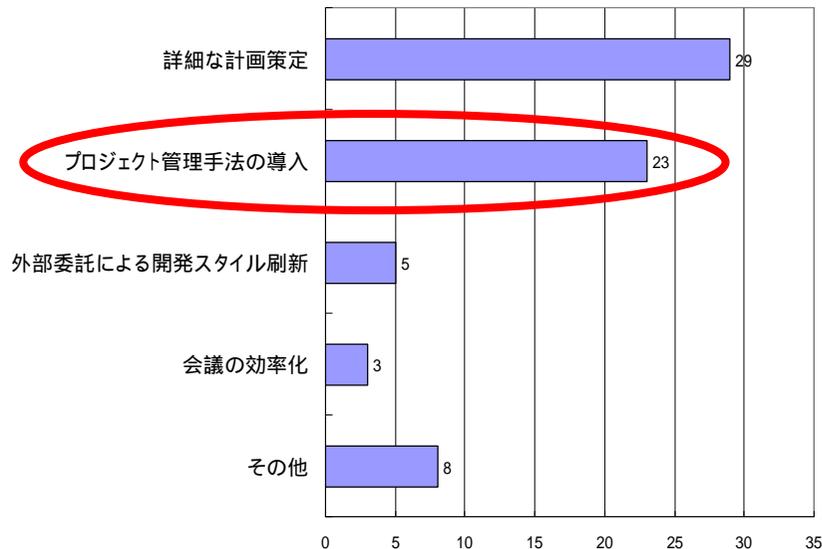


図7.2 ソフトウェア開発規模増加に対する施策

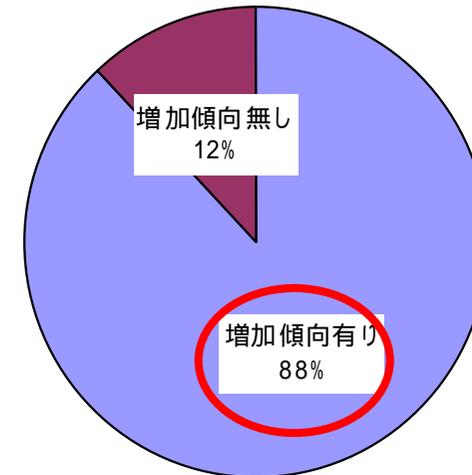


図7.1 ソフトウェア規模の増加傾向

7.1 ソフトウェアの大規模化 (2)

■ 全体を把握する仕組み(文書化)

- 全体構造を表記したドキュメントは83%が保有
- 図解表現は79%が採用
- 可視化意識は高いが、ツールを利用した体系的設計には至らず

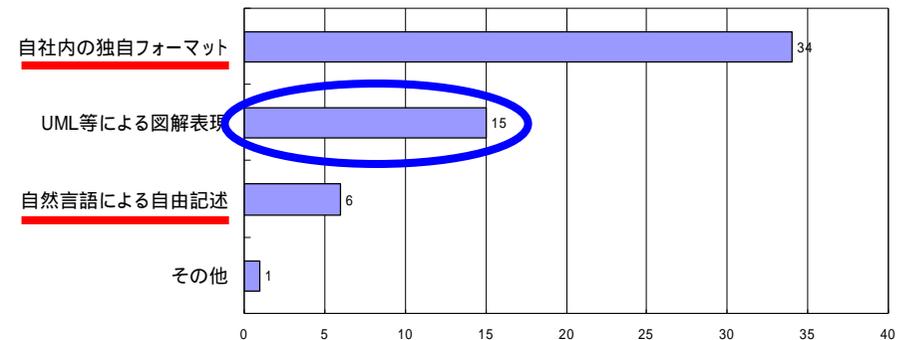


図7.3 全体構造を表記した設計書の記法

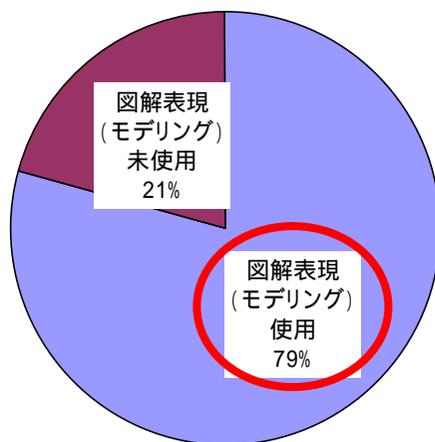


図7.4 図解表現(モデリング)の使用

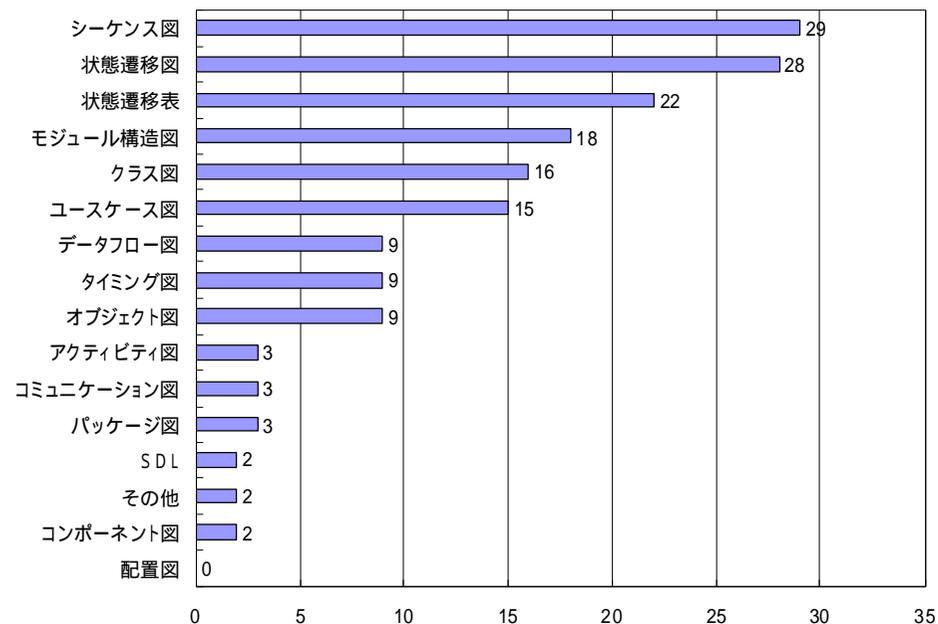


図7.5 図解表現(モデリング)の表記法

7.1 ソフトウェアの大規模化 (3)

■ 開発全体を統括する仕組み

- 工程の移行は60%が責任者の総合判断
- モジュール間インタフェースは、設計を進めながら、担当者同士や会議で決められる
- 55%のプロジェクトでソフトウェアアーキテクトが存在
- 全体統括を意識するも、アーキテクトの育成は手つかずの状況

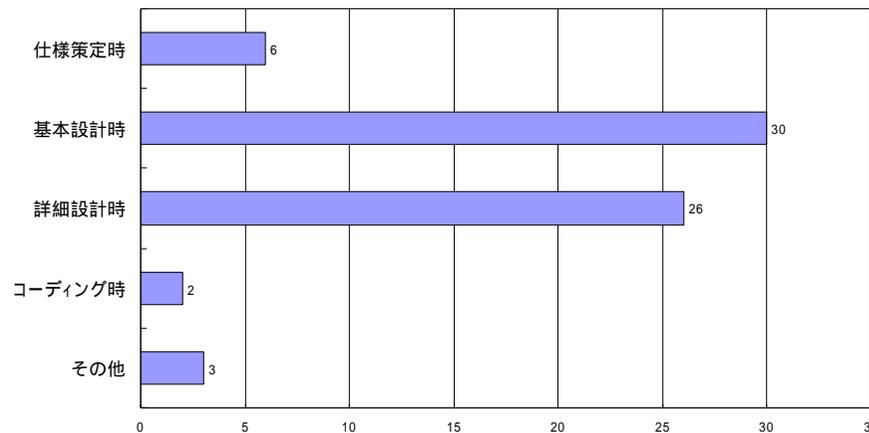


図7.7 ソフトウェアのモジュール間インタフェース仕様が決まる工程



図7.6 設計工程から次工程への移行判断方法

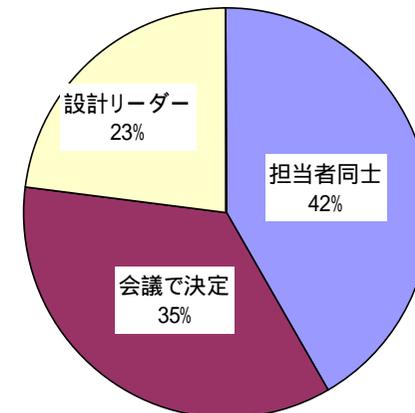


図7.8 ソフトウェアのモジュール間インタフェース仕様を決定する人

7.2 多機種開発 (1)

■ 多機種開発の方法

- 差分開発は77%のプロジェクトで行われている
- 多機種開発へは共通化により対応している
 - 工程の共通化を含むリソースの共通化
 - ソフトウェア資産の共通化・再利用

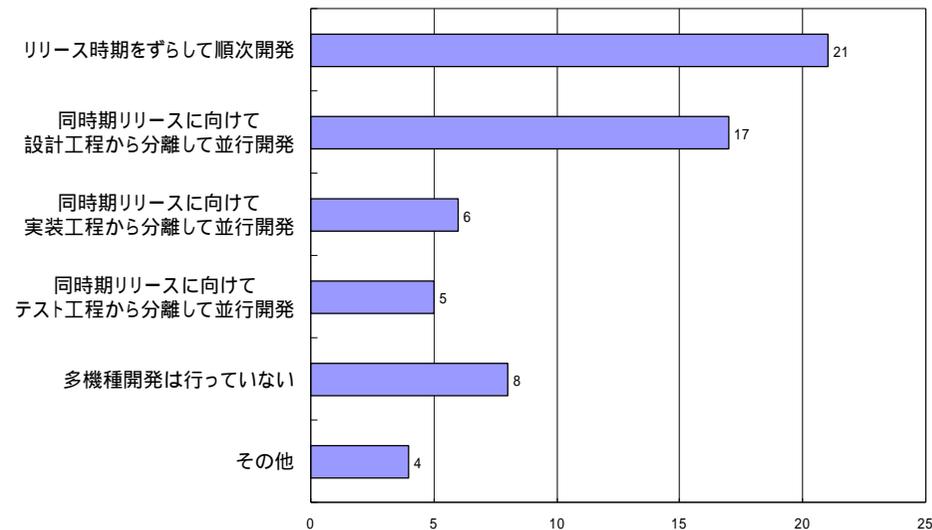


図7.10 多機種開発の方法

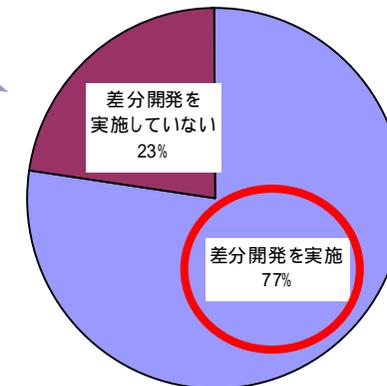


図7.9 差分開発の実施状況

7.2 多機種開発 (2)

■ 多機種開発の開発プロセス

- 開発プロジェクトの82%で開発プロセスがある
 - ウォーターフォールプロセスの採用が最も多い。
 - 増分型、反復型を併用するケースも多い。
- 画一的な開発プロセスモデルが、プロジェクトに必ずしも適合しない状況がうかがえる
 - 伝統的、制度的な理由で、特定の開発プロセスモデルが採用される場合、現場が不適合を訴える傾向が強くなる

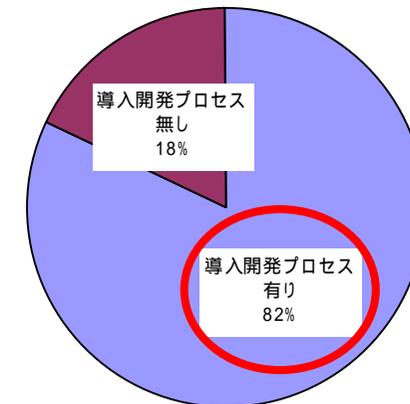


図7.11 導入している開発プロセスの有無

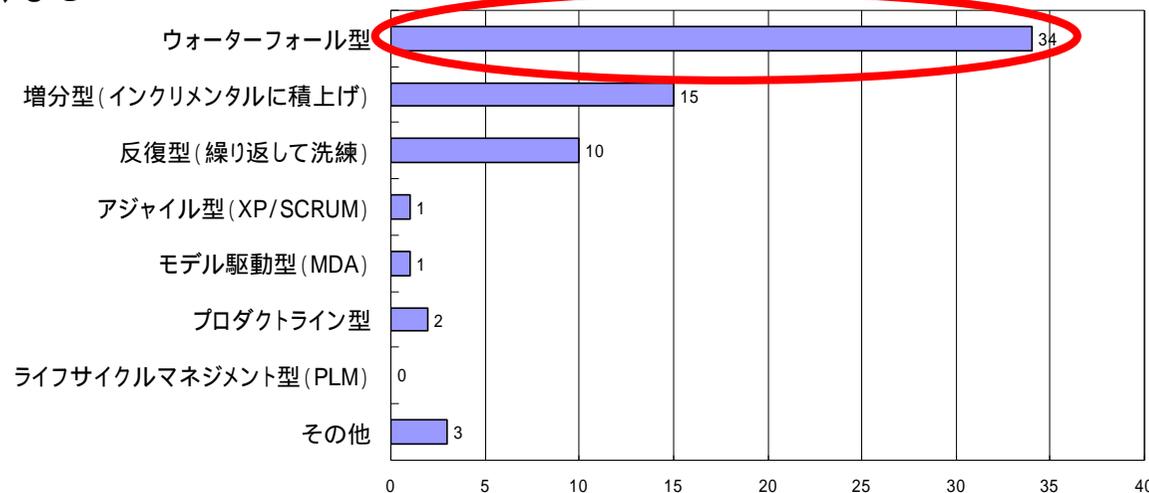


図7.12 導入している開発プロセス

7.2 多機種開発 (3)

■ 多機種開発のこれからの取組み

- プロダクトラインエンジニアリングの導入は11%と、まだ多くはない
- プロダクトラインエンジニアリング実践の要件としては、再利用への取り組みが最も進んでいる

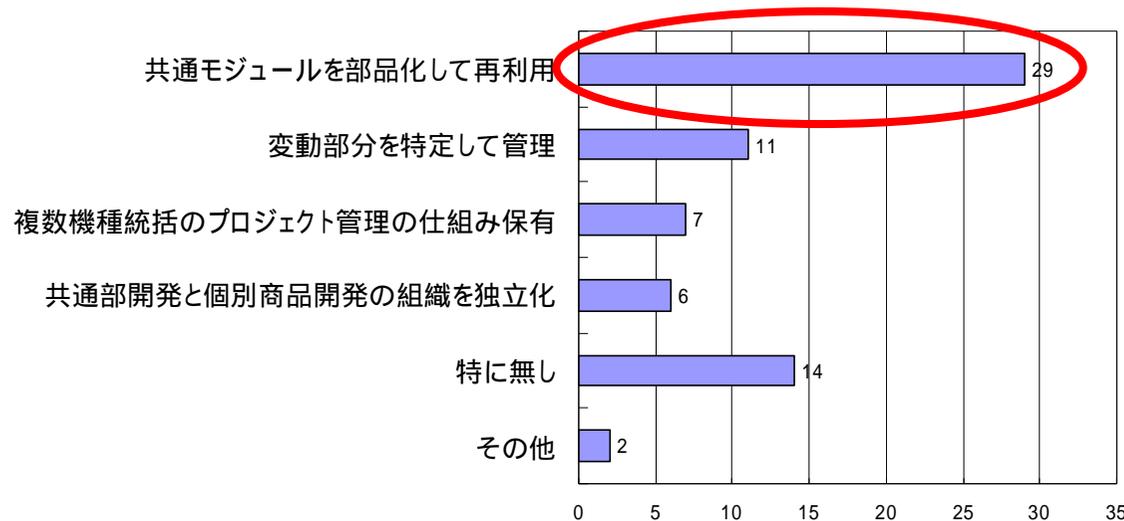


図7.14 多機種開発での工夫点

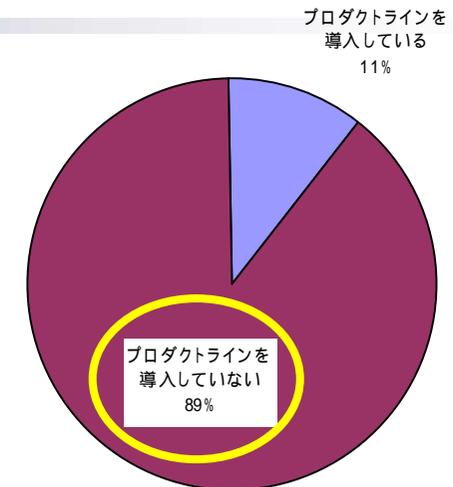


図7.13 プロダクトラインエンジニアリング導入状況

7.2 多機種開発 (4)

■ 多機種開発のプロジェクトマネジメント

- ハードウェアとソフトウェアを一体で管理しているプロジェクトは41%
- 開発メンバーがプロジェクト管理者を兼任するケースが52%
 - ソフトウェア担当者による兼任が半数

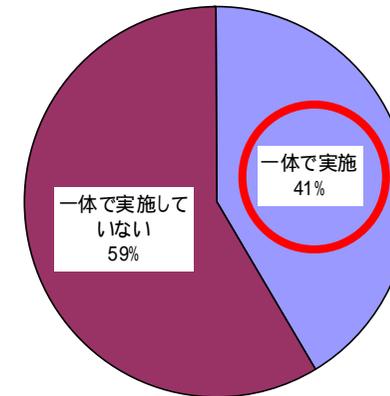


図7.15 プロジェクト管理はハードウェアとソフトウェアで一体か？

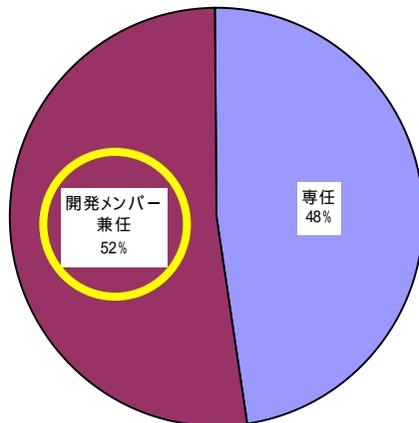


図7.16 プロジェクト管理者は専任か？
開発メンバー兼任か？

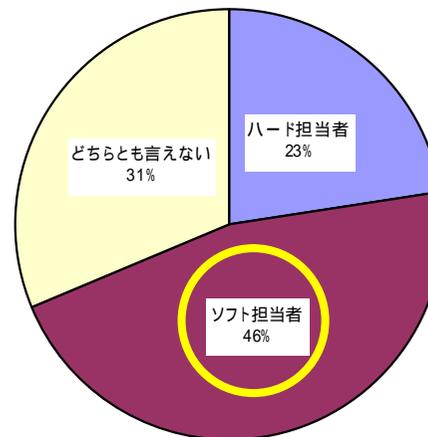


図7.17 プロジェクト管理者が兼任の場合、ハード担当者、ソフト担当者のどちらが担当か？

7.3 システム化と擦り合わせ (1)

■ ハードウェアとの協調

- ハードウェアとのインタフェース仕様決定は、1/4のプロジェクトで、詳細設計以降に
- 仕様決定者はハード担当者が多い
- ハードの変更はコーディング時まで発生し、ソフトが対応している
- ハードウェア主導の開発体制が多い

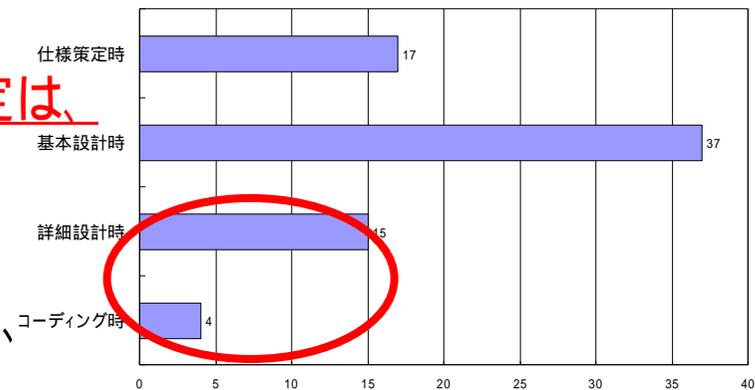


図7.18 ハードウェアとのインタフェース仕様の決定工程



図7.19 ハードウェアとのインタフェース仕様の決定者

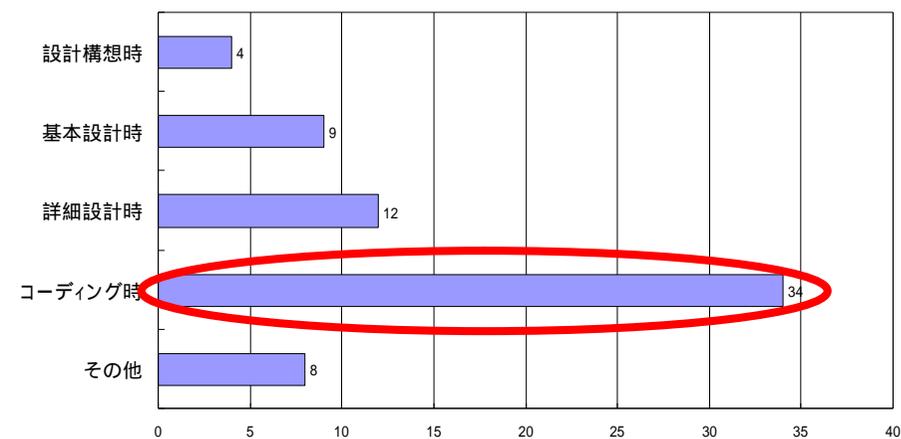


図7.20 ハードウェアの仕様変更時期はどの工程まで発生？

7.3 システム化と擦り合わせ (2)

- ハードウェアとソフトウェアを統合する開発プロセス
 - ハードウェアに起因する不具合をソフトウェアで対応するケースがある
 - インタフェース仕様の不備
 - ハードウェア性能や機能の補完
 - ソフトウェア側でハードウェアへの依存度を下げる工夫も
 - ハードウェア側との全体プロセスを考慮した工夫は少ない
 - ハードウェアとソフトウェアを統合した開発プロセスが存在していない

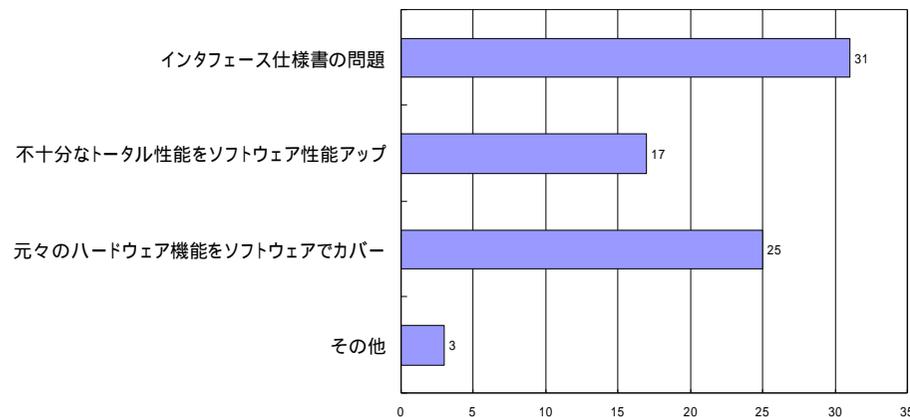


図7.21 システム総合試験で発生するソフトウェア問題点のうち、ハードウェアに起因する問題点

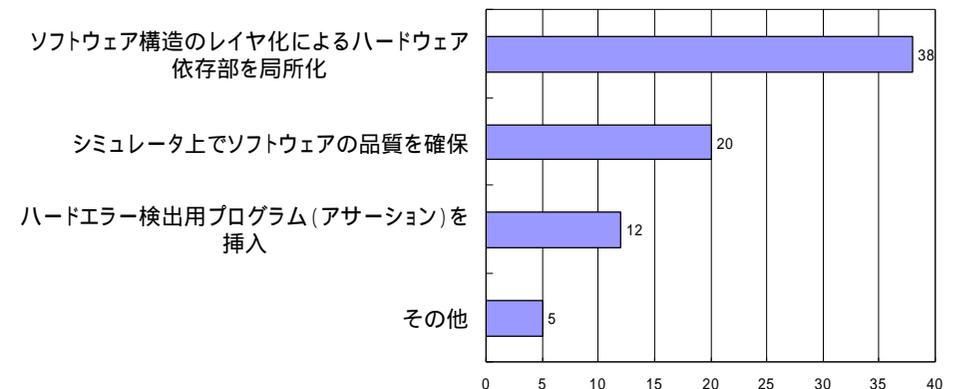


図7.22 ハードウェアの影響を局所化し、ソフトウェアの品質を確保する工夫

7.3 システム化と擦り合わせ (3)

■ ハードウェアとソフトウェアの擦り合わせ方法

- 34%のプロジェクトでは、ハードウェアとソフトウェアが関連する活動が事前に計画されていない
- 76%のプロジェクトでは、ハードウェアとソフトウェアの部門と一緒に、計画策定や各工程での確認をする機会がある
- システムアーキテクトは25%のプロジェクトで存在
- システムアーキテクト不在の場合は、ハードウェア担当部門が主導
- システムアーキテクトを育成する仕組みはない



図7.23 ハードウェアとソフトウェアの両方が関連する活動が事前に計画され明確になっているか？

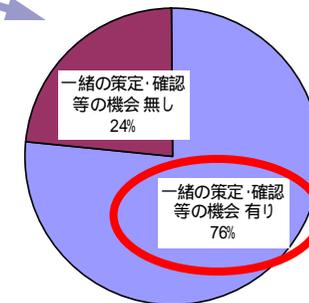


図7.24 ハードウェアとソフトウェアの担当部門が一緒になっての計画策定等を実施する機会

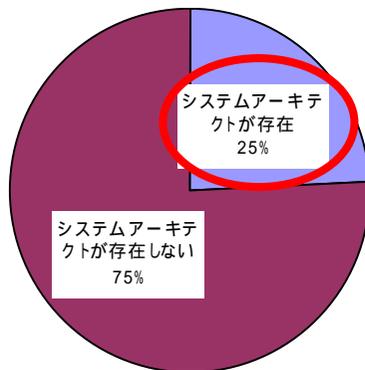


図7.25 システムアーキテクトの存在



図7.26 システムアーキテクト不在の場合、ハードウェアとソフトウェア間の機能分担等に責任を持つ部門は？

7.4 品質プロセスに関する提言

- 大規模化に適合する開発プロセスの事例収集と周知展開の仕組み
 - ベストプラクティスなどの知識化を行い、周知する仕掛けが必要
 - 実践的な大規模化の取り扱い事例
 - 再利用性向上の先行事例
- 多機種開発を考慮した工学的な開発プロセスの研究
 - アーキテクチャやプロダクトラインエンジニアリングの研究
 - 日本の実情に即した実用化の研究
- 日本の強みと弱みを意識した方向性の提示と共有
 - プロアクティブな擦り合わせへの進化
 - アーキテクチャを中心としたトップダウンアプローチのモデル化
 - 両アプローチの融合による国際競争力の向上
- アーキテクトの育成が急務
 - 位置付けの明確化とプログラムの整備により、戦略的に育成