Quality Agent

組込みソフトウェア産業実態調査から見る

組込みソフトウェア開発現場の課題

株式会社クオリティ・エージェント編

目　次

1. はじめに 1
1.1 組込みソフトウェアの不具合対策費 1
2. 組込みソフトウェア市場について 3
2.1 ソフトウェア産業の市場規模 3
2.2 組込みソフトウェアの市場規模 4
2.3 組込みソフトウェア技術者数 6
3. 組込みソフトウェア開発の現状 10
3.1 開発規模 10
3.2 人員構成 12
3.3 組込み製品出荷後の不具合原因 15
3.4 外部委託状況 16
3.5 組込みソフトウェアの受託状況 21
3.6 外部調達のミドルウェア 22
3.7 利用した手法・技法 26
3.8 使用ツール 28
4. 組込みソフトウェア開発の課題と解決策 33
4.1 設計品質の向上 33
4.2 人材育成 35
4.2.1 新人社員教育 40
4.2.2 スキルチェンジ 40
4.2.3 語学力 42
4.3 外部委託 43

4.3.1 外部委託側の課題 46
4.3.2 受託側の課題 50
4.4 外部調達 52
5. おわりに 54
付録 1. 参考資料 55

1. はじめに

経済産業省では、2004 年度より組込みソフトウェアの開発力の推進に取り組んでおり、その中で、2004 年 10 月に「独立行政法人情報処理推進機構（IPA）」内に、ソフトウェアエンジニアリングセンター（SEC）を発足させ、組込みソフトウェアの開発力強化に向けての様々な活動を行っています。その活動のひとつとして、毎年、国内外における組込みソフトウェア開発を行う企業を対象として、組込みソフトウェア産業実態調査（以下、実態調査）を行っています。2010 年度は、事業責任者、プロジェクト責任者、技術者個人、海外の 4 つのカテゴリ別に調査を行い、その結果を、実態調査結果としています。

なお、実態調査結果の詳細については、以下のサイト上にありますので、ご興味のある方は、是非、見てみて下さい。

http://www.meti.go.jp/policy/mono_info_service/joho/ESIR/index.html

本書では、この実態調査の結果を元に、現在の組込みソフトウェアの市場及び開発状況から、組込みソフトウェアの品質面からの問題点を抽出し、その問題解決について考えていきたいと思います。

1.1 組込みソフトウェアの不具合対策費

私たちの身の回りにある携帯電話やスマートフォンなどの情報端末、テレビ、デジタルカメラなどの家電製品、乗用車やバイク、電車などの乗り物、その他にも多くの産業機器には、何らかの組込みソフトウェアが搭載され、製品を購入した利用者は気づかないうちに組込みソフトウェアを利用しており、いまや組込みソフトウェアを使っていることは分かりませんが、身近な存在となっています。

図 1-1 組込みソフトウェア搭載例

組込みソフトウェアは、製品に組み込まれるという性格上、制御機器や人命にかかわる機器類、さらにそれらの機器からネットワークに接続されることも多く、何らかの問題が発生した場合には、機器そのものを入れ替えたり、さまざまな機器類へ影響を及ぼしたりと、不具合が発生した時の影響が大

きくなることも少なくありません。

以下のグラフは、組込みソフトウェアにおける不具合が発生したときの対策費用の合計を示したものです。

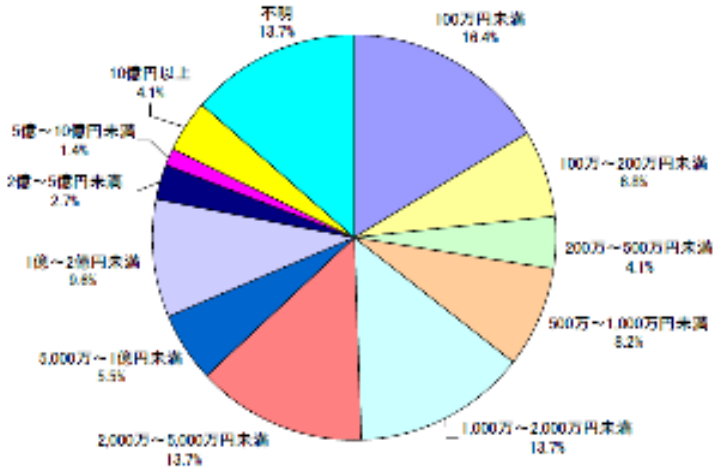


図 1-2 不具合が発生したことによる対策費の総合計（2008 会計年度）

このグラフを見て分かるように、不具合が発生したことによる対策費として 1,000 万円以上かけている企業が 50.7%もあり、そのうち、17.8%が 1 億円を、更に、4.1%が 10 億円以上もの費用をかけていることが分かります。

製品に組み込まれるという組込みソフトウェアの性質上、製品出荷後に不具合が発生してしまうと、製品そのものを交換したり、交換はしなくとも、通信費などを負担してのソフトウェアの入れ替え作業が発生したりと、多くの対策費がかかることがあります。また、近年では、多くの機器と接続されていたり、ネットワークに接続されていたりすることも少なくないため、不具合等が発生した際には、更に対策費が上昇することが考えられます。このため、組込みソフトウェアの品質を確保するということは、顧客満足の観点のみならず、経営リスクともなり得る重要な課題であると言えるでしょう。

2. 組込みソフトウェア市場について

本章では、組込みソフトウェア市場について、解説します。

2.1 ソフトウェア産業の市場規模

まず、組込みソフトウェア市場規模を見る前に、それを取り巻くソフトウェア業界全体の動向を見ていきましょう。

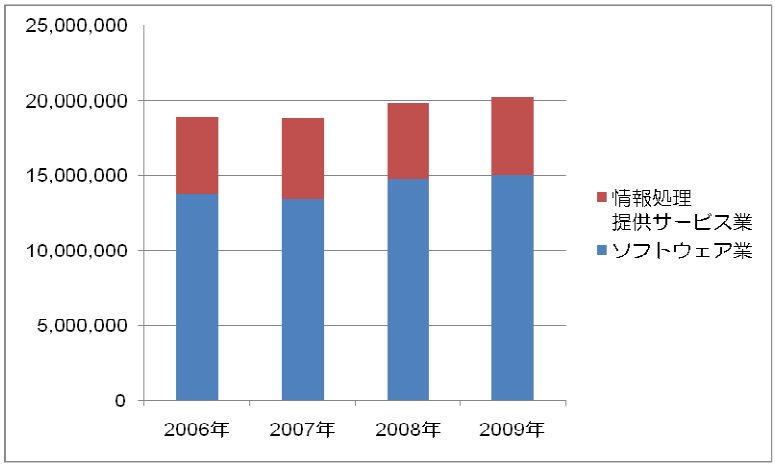


図 2-1 ソフトウェアおよび情報処理サービス業の売上高

「図 2-1 ソフトウェアおよび情報処理サービス業の売上高」は、経済産業省からの統計データである特定サービス産業実態調査結果からのデータとなっており、ソフトウェア業および情報処理サービス業の売上高の推移を示したものです[1]。このグラフを見ると、ソフトウェアおよび情報処理サービス業の売上高は、2006 年に 18 兆 8951

[1] 経済産業省「特定サービス産業実態調査結果」については、以下のサイトにあります。
http://www.meti.go.jp/statistics/tyo/tokusabizi/index.html
元データでは、元号表示となっていますが、組込みソフトウェア実態調査結果に合わせるために、本書では、西暦での表示に変更してあります。

億円、2007 年に 18 兆 8261 億円（前年比：-0.37%）、2008 年に 19 兆 8454 億円（前年比：+5.41%）、2009 年に 20 兆 2264 億円（前年比：+1.92%）と推移しています。

2006 年から 2007 年にかけて、一旦、前年度比で減少したものの、2009 年には、売上高が 20 兆円を超え、2006 年と比較すると、プラス 7.05%の増加率となっており、ソフトウェアおよび情報処理サービス産業全体が成長していることが分かります。

今後も、国内 IT 大手企業が、中国をはじめとするアジア圏へのシステムサービス拡充する動きも活発化しており、それに伴い、ソフトウェアおよび情報処理サービス業全体の市場規模も成長していくと考えられます。

2.2 組込みソフトウェアの市場規模

それでは、組込み関連産業は、どうでしょう。

以下に、国内総生産額と組込み関連産業生産額の推移を示します。

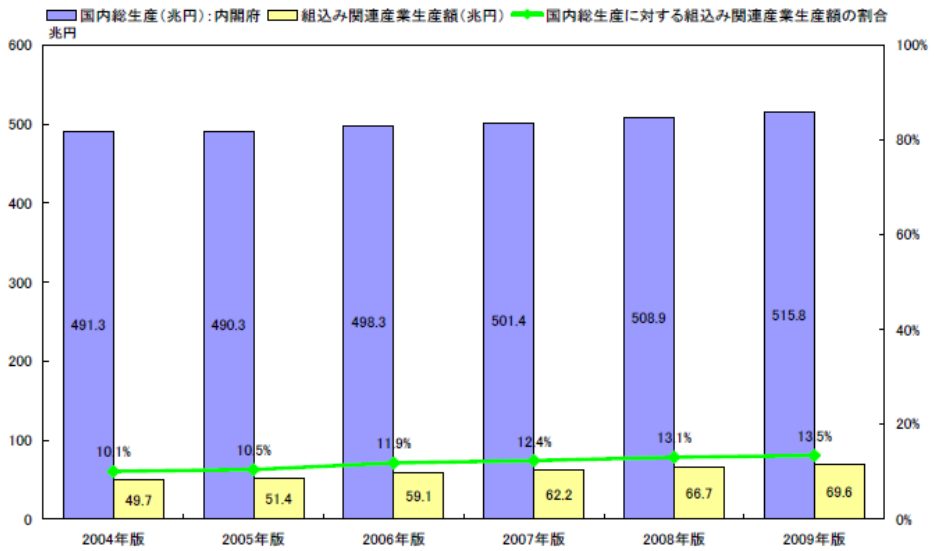


図 2-2 国内総生産額と組込み関連産業生産額の推移

このグラフは、2009 年の実態調査結果からの引用となりますが、組込み関連産業全体で、2004 年と比較して、生産額で 47.9 兆円から 69.6 兆円と 19.9 兆円の増加（増加率：+40.0%）、国内総生産に対する割合は、10.1%から 13.5%（増加率：+33.7%）と増加してきています。

組込み関連産業は、カーエレクトロニクス関連における環境に配慮したハイブリッド車や電気自動車などの新たな車種の増加、携帯電話を含む情報端末機器でのスマートフォン市場の拡大、3D テレビなどの新たなデジタル家電製品の増加などが見込まれており、これらの製品を中心に市場規模は、ソフトウェアおよび情報処理サービス業全体としての成長と共に、組込みソフトウェア自体も、更に成長していくことが予想出来ます。

組込み産業全体の市場規模が拡大していく中、組込みソフトウェアそのものの市場規模はどうなるでしょう。

以下に、組込み製品開発費と組込みソフトウェア開発費の推移を示します。

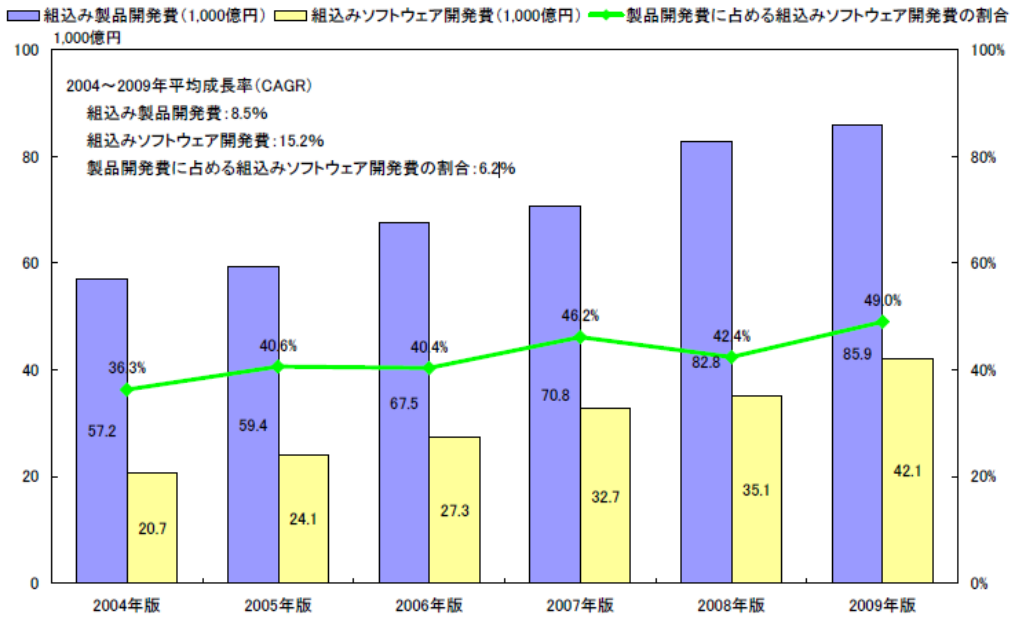
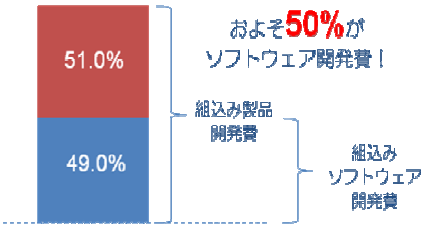


図 2-3 組込み製品開発費と組込みソフトウェア開発費の推移

組込み製品開発費は、2004 年の 5 兆 7200 億円から 2009 年には 8 兆 5900 億円（増加率：+50.2%）、組込みソフトウェア開発費は、2004 年の 2 兆 700 億円から 2009 年には 4 兆 2100 億円（増加率：+103.4%）と、それぞれ増加し続けています。

組込み製品開発費に対する組込みソフトウェア開発費の割合は、その年により増減はあるものの、調査開始時点から概ね増加を続けており、2004 年の 36.3%から 2009 年の 49.0%と、組込み製品開発費の中で、組込みソフトウェアの開発費がおよそ半分を占めるまで伸びてきています。



これは、組込み製品自体の製造コストに大きな変化がないと想定した場合、組込みソフトウェアに対する開発コストが増大していることを示しており、組込み製品自体の価格に大きく影響を与えてしまうことになります。つまり、今後、組込み製品そのものの価格を抑えるためには、組込みソフトウェアの開発費用を削減していく必要があり、組込み製品を扱っている企業では、そのための施策を必要としていることを示しているのです。

2.3　組込みソフトウェア技術者数

組込みソフトウェアの市場規模は、間違いなく拡大しており、今後も拡大し続けることが予想されますが、組込みソフトウェアの市場規模が大きくなるにつれて、当然のことながら、そこに従事する組込みソフトウェア技術者不足が懸念されます。

以下は、組込みソフトウェア技術者数（現状人数と不足人数）の推移を示しています。

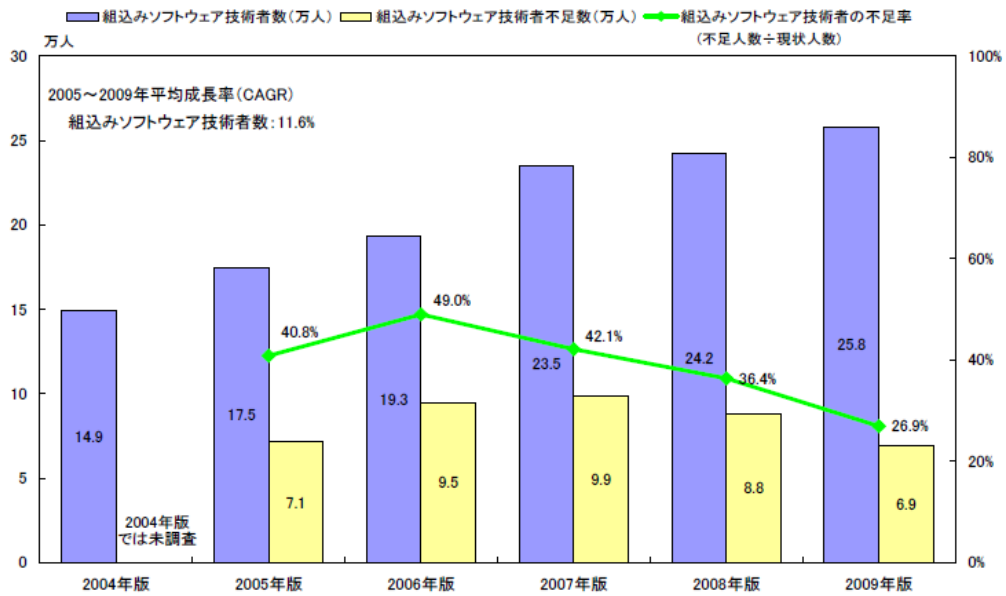


図 2-4 組込みソフトウェア技術者数(現状人数と不足人数)の推移

市場規模の拡大と共に、年々、組込みソフトウェア技術者数も増加しており、2004年の 14.9 万人から 2009 年の 25.8 万人と、6 年間で、10.9 万人（およそ 1.73 倍）も増加しています。平均成長率が 11.9%（図内左上参照）となっていますので、技術者数は、急増していると言っても過言ではありません。

一方、組込みソフトウェア技術者の不足人数は、2007 年の 9.9 万人の不足をピークに、2008 年に 8.8 万人（前年比：-11.1%）、2009 年に 6.9 万人（前年比：-21.5%）と減少傾向にあります。
しかし、組込みソフトウェアの不足率は、2009 年でも、26.9%も不足しており、慢性的に組込みソフトウェア技術者は不足していると言えます。

少しずつ組込みソフトウェア技術者不足の改善は行われているものの、組込みソフトウェア産業の市場規模拡大に対して、人材の育成や確保が追い付いていない結果と言えるかも知れません。

「図 2-5　従業員の平均増減数」は、企業内における従業員の増減数を示したもので、また、「図 2-6　事業部門の従業員の増減技術者比率」には、その比率を示したものです。

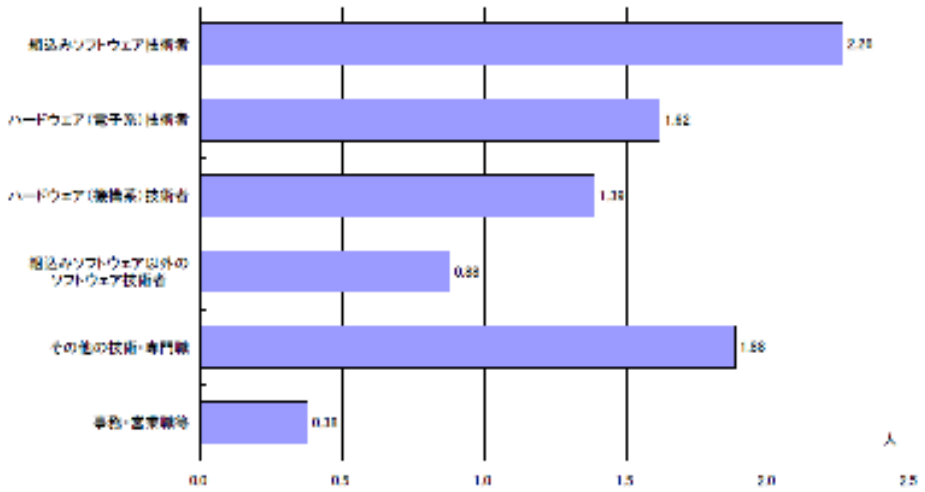


図 2-5 従業員の平均増減数

このグラフを見ると、組込みソフトウェア技術者の平均増減数が 2.26 人と、他の技術者に比べても顕著に増加していることが分かります。

組込みソフトウェア以外のソフトウェア技術者は、0.88 人と 1 人に満たない状況であるにも関わらず、組込みソフトウェア技術者が、2.26 人の増加があることを考えてみても、企業内において、組込みソフトウェア技術者を求め、採用を行ってきたことが分かります。

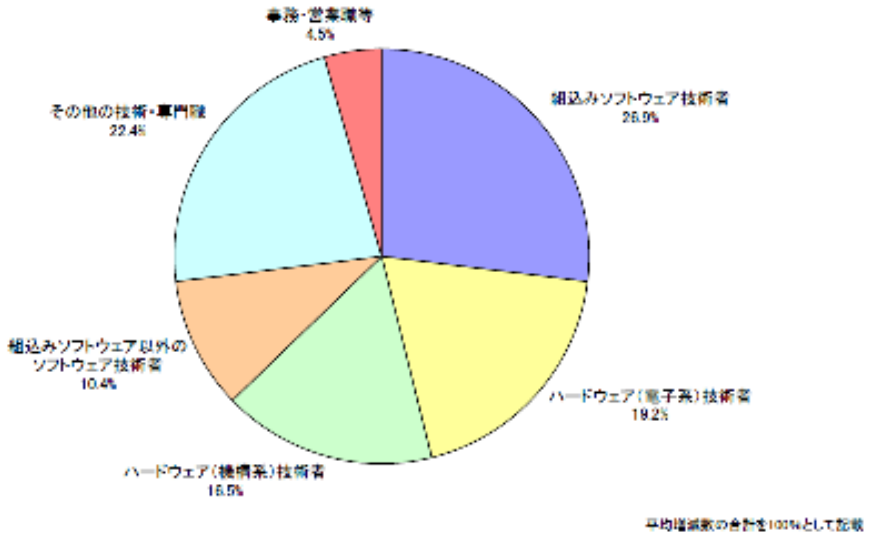


図 2-6 事業部門の従業員の増減技術者比率

では、このように増えてきた組込みソフトウェア技術者を企業としては、どのように採用したのかについても見てみることにしましょう。

以下に、事業部門の技術者の増員方法別平均増員数を示します。

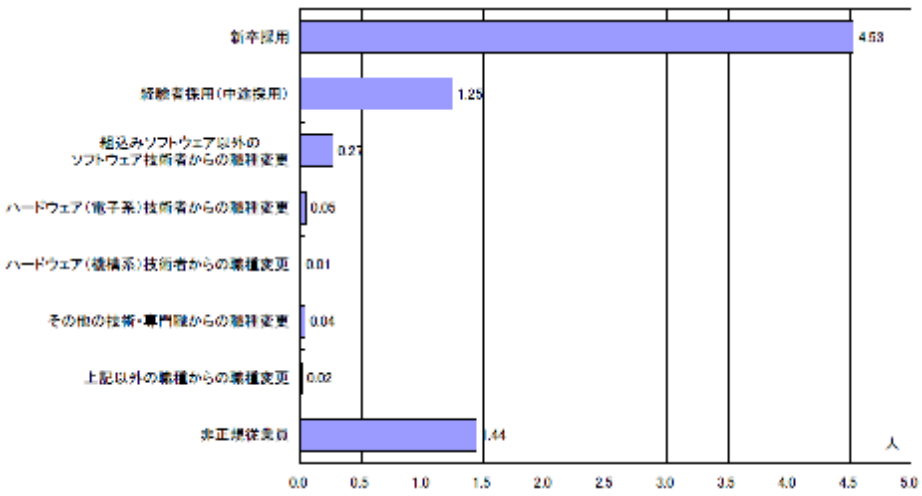


図 2-7 事業部門の技術者の増員方法別平均増員数

「図 2-7 事業部門の技術者の増員方法別平均増員数」からは、その多くが、新卒採用（4.53 人）として人材を採用していますが、経験者採用（中途採用）（1.25 人）や非正規従業員（1.44 人）も比較的高い人数となっています。

企業では、新卒採用し人材を育成していくことと、経験者採用や非正規従業員として、即戦力となる人材を必要としていることが分かりますが、スキルチェンジによる人材活用はあまり行えていない現実も見えてきます。これは、組込みソフトウェアの技術力習得が難しいか、もしくは、企業内において人材育成がうまく機能していない可能性がありますが、どちらにしても、不足している組込みソフトウェア技術者を育成するための社内の環境を整備していく必要があると言えます。もし、企業内における人材育成環境が整っていないとするならば、新規採用した人材が、その後、優秀な組込みソフトウェア技術者として育成することも難しいものになる可能性もあるため、企業にとって人材育成環境整備は、重要な課題と言えるでしょう。

組込みソフトウェア技術者は、企業から求められていますし、今後の組込みソフトウェアの市場規模の拡大も考えた場合には、この傾向は、更に続くものと考えられます。そのためにも、組込みソフトウェア技術者を育成するための仕組みづくりや環境整備についても、企業内で構築していく必要があるのかも知れません。

3. 組込みソフトウェア開発の現状

本章では、組込みソフトウェア開発の現状を解説します。

3.1 開発規模

開発規模にもさまざまなデータがありますが、ここでは、開発するソフトウェアそのものを見ていきましょう。「図 3-1 全行数（新規開発と既存の合計）」は、組込みソフトウェアの全行数（新規開発と既存の合計）を示しています。

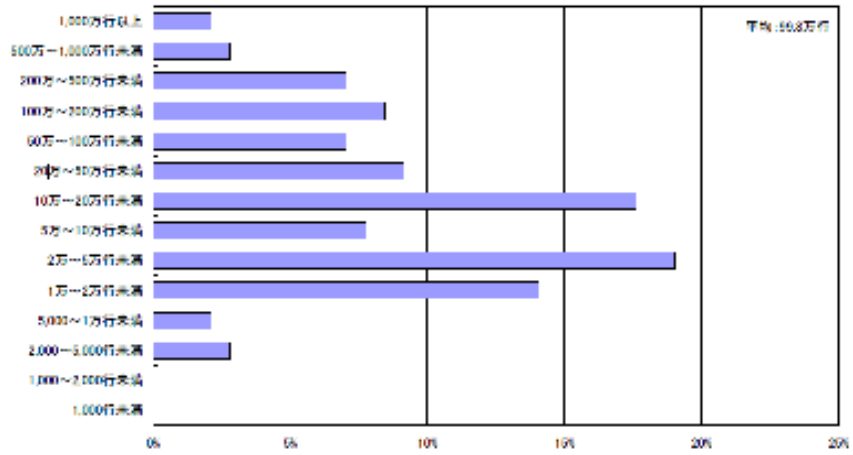


図 3-1 全行数（新規開発と既存の合計）

2009 年の実態調査結果では、開発規模の平均行数は 181 万行となっており、2010 年は、2009 年と比較すると減少はしていますが、それでも平均行数は、およそ 100 万行（99.8 万行）もあり、また、1,000 万行以上の組込みソフトウェアも 2.1%も存在しています。

この 1,000 万行という行数については、単純に比較することは出来ませんが、OSS（Open source software）の OS である Linux が、1,000 万行以上ありますので、それに匹敵するほどの巨大な組込みソフトウェアが作られていることになります。

なお、1,000 万行以上の組込みソフトウェアとしては、高級車に搭載される車載ソフトウェアが有名ですが、その行数は、既に数千万行を超えており、2015 年には、1 億行にまで達すると言われています。この高級車には、ECU（Electric Control Unit）[2]が 100 個以上も搭載されており、それぞれのシステムを組込みソフトウェアによって制御しています。

1 億行を 100 個の ECU で単純に割っても、ECU1 個あたりに、100 万行にもおよぶ巨大な組込みソフトウェアが動作していることになります[3]。

また、新規開発規模（「図 3-2　新規開発行数」参照）を見てみると、2010 年は、45.4 万行となっています。2009 年の実態調査結果では、新規開発規模は 21.7 万行となっており、増加率が 109.2%と、ほぼ倍増していることが分かります。更に、全行数（新規開発と既存の合計）の平均行数 99.8 万行と新規開発規模の平均行数 45.4 万行との割合を考えると、単純に計算することは出来ませんが、およそプログラムの半分（45.5%）が新規に開発されていると考えることができます。

なお、2009 年は、全行数が 181 万行、新規開発行数が 21.7 万行となっており、その割合は 11.98%となります。2010 年の 45.5%と比べると、大幅に新規開発の比率が増えていることが分かります。

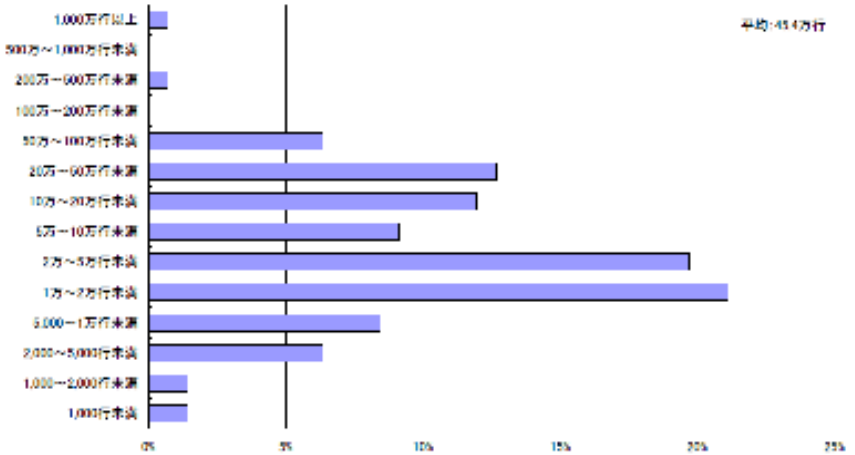


図 3-2　新規開発行数

新規開発が多いということは、新たな不具合が含まれる可能性が高くなるばかりではありません。新規開発部分は、既存部にも少なからず影響を与えることがあるため、

[2] 自動車向け電子制御ユニットで、エンジン、ブレーキ、電池などの車載システムごとにエンジン ECU、ブレーキ ECU などが搭載され、各システムを制御する役目を果たします。いわゆる車の頭脳とも言うべき装置です。

[3] 実際には、単純計算ではありませんので、あくまでも参考程度にお考えください。

新規開発部の品質だけではなく、既存部への影響範囲部分の品質についても、対策を考えていく必要があります。新規開発が大きくなるということは、影響を与える範囲も大きくなってくる可能性が高くなるため、品質を確保するために必要となるコストも増大する可能性が高くなることを示していると言えます。

このように、組込みソフトウェアの開発規模は肥大化してきており、これは、組込み製品の高機能化、高性能化によるものと言われています。このような開発規模の肥大化の状況は、今後も少なからず続くものと考えられますし、また、大きくなった組込みソフトウェアを置き換えることは、コスト面からも難しくなってきます。このため、巨大な組込みソフトウェアに対し、新規開発と既存を切り分け、新規開発部分への品質確保と、既存部への影響範囲の特定とその品質確保を求められていると言えるでしょう。

多くの書籍では、新規開発のプロジェクトに対する品質確保の方法を示しています。これは、既存部分の品質確保には、その企業ごとのノウハウが含まれていることも少なくないため、一般的な話として出し難い側面を持っており、組込みソフトウェア開発企業では、このノウハウをいかに早く蓄積出来るのかどうかについても、十分に検討していく必要があるでしょう。

3.2　人員構成

それでは、組込みソフトウェアのプロジェクトにおける人員の構成を見てみることにしましょう。

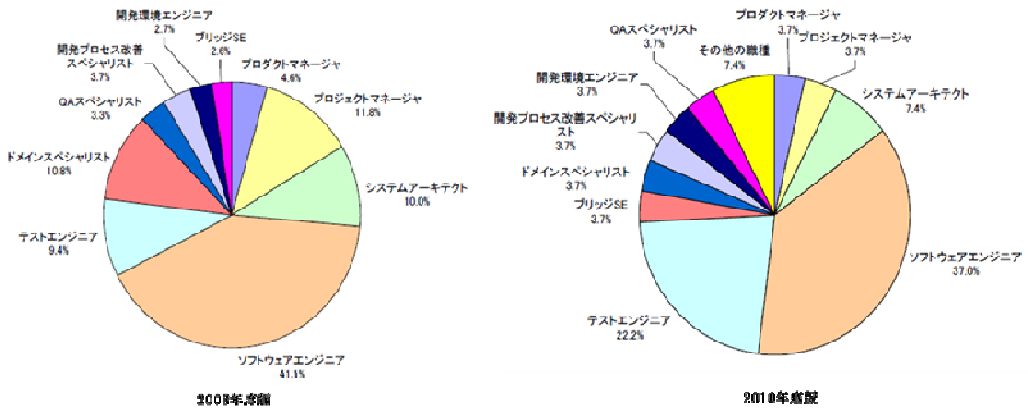


図 3-3 プロジェクトの職種構成比較

「図 3-3　プロジェクトの職種構成比較」は、2008 年（左）と 2010 年（右）のプロジェクトの職種別人員構成の比較です。

2008 年では、ソフトウェアエンジニア 41.1%に対して品質に関わる人員はテストエンジニア（9.4%）、QA スペシャリスト（3.3%）の合計 12.7%しかいません。

つまり組込みソフトウェアを作る人員（ソフトウェアエンジニア）に対して、品質の確認を行う人員（テストエンジニア、QA スペシャリスト）が、30.9%の人数しかいなかったことになります。人数比率としては、約 3:1 の比率となっており、組込みソフトウェアを作成することを中心に考え、プロジェクトへ人員を投入していると言えるかも知れません。

ところが、2010 年には、ソフトウェアエンジニア 37.0%に対して、テストエンジニアが 22.2%、QA スペシャリストが 3.7%と合計で 25.9%と、比率としては、およそ 10:7 にまで増加していることが分かります。（「図 3-4　開発人員と品質関連人員のバランス」参照）

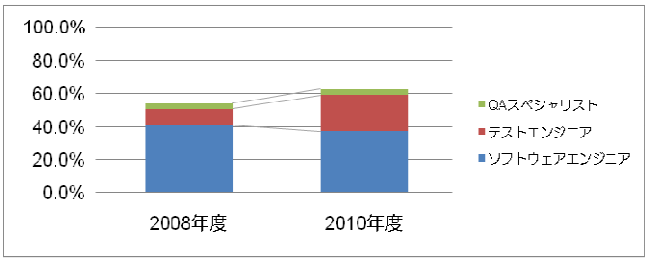


図 3-4 開発人員と品質関連人員のバランス

2008 年に比べると、ソフトウェアエンジニアは 4.1%の減少、それに対して、テストエンジニアが 12.8%増加、QA スペシャリストは 0.4%増加の合計で 13.2%も増加しています。この結果を見てみると、2008 年に比べ、組込みソフトウェアの品質に対する意識が高まり、企業としての取り組みによって、人材を登用してきた結果であると言えるかも知れません。

ところが、少し問題があります。

その前に、1 つのプロジェクトに関わる人数を見てみましょう。（「図 3-5　プロジェクトの投入人数：全体（社内＋社外）」参照）

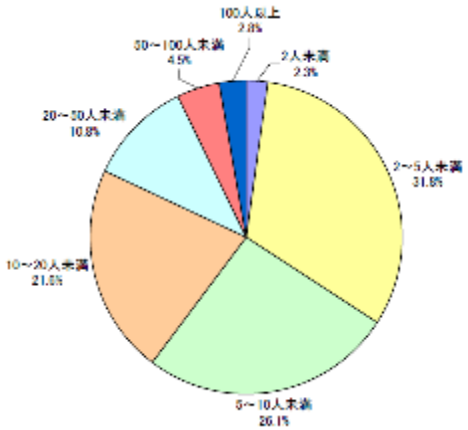


図 3-5 プロジェクトの投入人数：全体（社内＋社外）

プロジェクトに投入される人数は、「2~5 人未満」がもっとも多く 31.8%となって

おり、次いで、「5~10 人未満」の 26.1%、「10~20 人未満」の 21.6%となっており、平均としては、「5 人~10 人未満」となります。

ここで、先ほどの「図 3-3　プロジェクトの職種構成比較」と合わせて見てみると、ソフトウェアエンジニア（2~4 名）およびテストエンジニア（1~2 名）以外は、プロジェクト投入人数が 10 人だったとしても 1 人にも満たないことになります。つまり、ソフトウェアエンジニアおよびテストエンジニア以外の職種の人が行うべき作業というのは、誰かがその役割を担っていることになり、これらの作業を、プロジェクトマネージャが担っていると考えるのが妥当ではないかと思います。

これは、2010 年に限った話ではありませんが、本来は、多くの職種の方々が行うべき作業を、プロジェクトマネージャが担うことになっているのであれば、必然的に、広範な知識を有していなければいけなかったり、技術的にも高いレベルでなければいけなかったりと、プロジェクトマネージャへの要求が拡大している可能性が高くなります。これによって、プロジェクトマネージャが足りない、プロジェクトマネージャになれる人材がいないといった問題に発展する可能性も高くなっていると考えられます。また、多くの中小企業では、職種別に作業を割り当てられるだけの人材を登用することは難しいものとなりますので、慢性的にプロジェクトマネージャ不足は続くのではないかと思われます。

次に、プロジェクトの総工数を見てみましょう。（「図 3-6　プロジェクトの総工数」参照）

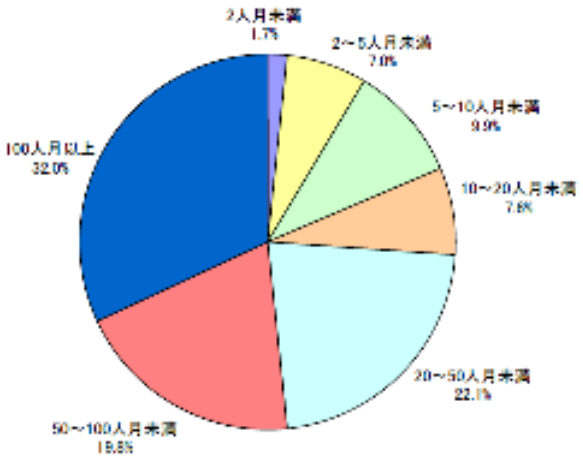


図 3-6 プロジェクトの総工数：全体（社内＋社外）

プロジェクトの総工数は、もっとも比率が高いのが、「100 人月以上」の 32.0%、次いで、「20~50 人未満」が 22.1%、「50 人~100 人未満」が 19.8%となっており、

50 人月以上もの工数が発生する大きなプロジェクトが半分以上もあることが分かり、工数の面からも組込みソフトウェアの開発規模が大きくなっていることが見えてきます。

「図 3-6　プロジェクトの総工数：全体（社内＋社外）」と「図 3-5　プロジェクトの投入人数：全体（社内＋社外）」の両データを合わせて見ると、平均的なプロジェクトの総工数は「50 人~100 人月未満」であり、そのプロジェクトへ投入されている人員は「5 人~10 人未満」となりますので、平均的なプロジェクト期間としては、10 ヶ月程度と言えるでしょう。この 10 ヶ月というプロジェクトの期間が早いかどうかは議論が分かれるところだと思いますが、この期間で、「図 3-1　全行数（新規開発と既存の合計）」の約 100 万行を取り扱うのだとすれば、平均で月当たりに 10 万行を開発しなければいけないことになります。

更に、ここに、「図 3-3　プロジェクトの職種構成比較」からソフトウェアエンジニアの人数を求めると約 4 人となりますので、1 人あたり 2.5 万行／月、1 時間当たり 156 行[4]を取り扱う必要があることになります。なお、この行数は、既存部と新規部の両方の合計行数となるため、「図 3-2　新規開発行数」で考えてみると、新規開発行数は、平均行数は 45.4 万行ですので、1 人あたり 1.135 万行／月、1 時間あたり約 71 行を新たに作成しなければいけません。

プログラミング言語によって異なりますが、1 時間あたりで、156 行あるいは 71 行程度の行数であれば、それほど多くはないと感じることでしょう。

しかし、実際のプロジェクト開発を考えた場合には、プロジェクト期間である 10 ヶ月すべてを、ソフトウェアエンジニアが開発していることはありえませんので、このため、実際には、3~4 倍（213~624 行）程度の行数になってしまうことが考えられます。

3.3　組込み製品出荷後の不具合原因

「図 3-7　製品出荷後の不具合の割合」は、2009 年（左）と 2010 年（右）の組込み製品出荷後の不具合の原因の占める割合を示しています。

[4] 1 ヶ月の稼働日数を 20 日、1 日あたり 8 時間として計算しています。

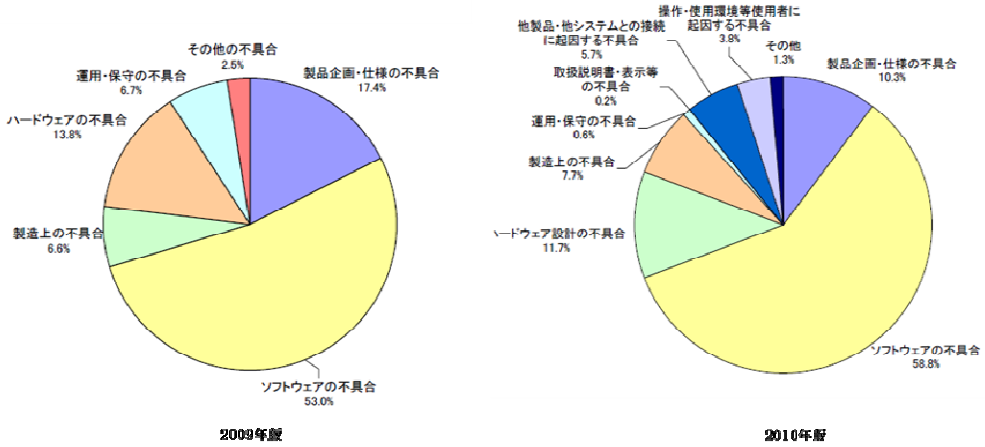


図 3-7 製品出荷後の不具合の割合

2009 年と 2010 年での項目に若干の差異はありますが、「ソフトウェアの不具合」が、2009 年には 53.0%、2010 年には 58.8%と、5.8%（増加率：10.9%）も増加していると共に、製品出荷後の不具合として大きな割合を占めていることが分かります。

しかし、実態調査結果は、組込みソフトウェアに関わる企業への調査であるため、「ソフトウェアの不具合」が多いからといって、それだけを問題視することは出来ません[5]が、それでも、「図 1-2 不具合が発生したことによる対策費の総合計（2008 会計年度）」を考えた場合には、ここに記述されている原因に起因して多くの対策費がかかっているのですから、組込みソフトウェアへの品質向上への取り組みは、重要な位置づけであると言わざるを得ないでしょう。

3.4 外部委託状況

「図 3-8 外部委託の割合と委託先割合」に、外部委託の割合と外部委託先の割合を示しています。

[5] 組込みソフトウェアだけを作っているソフトウェアベンダーの場合には、ハードウェアに起因する問題は少なくなる傾向があるため、必然的に、すべてソフトウェアの不具合となってしまうことがあるためです。

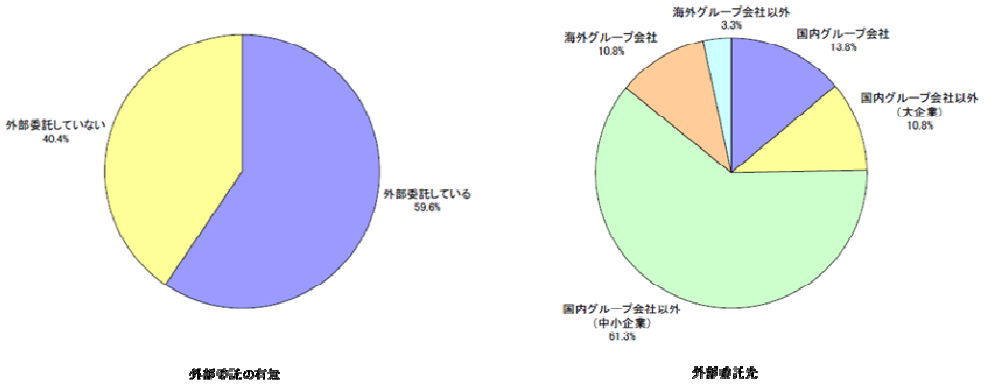


図 3-8 外部委託の割合と委託先割合

外部委託の割合を見てみると、実に 59.6%が外部に開発を委託している状況となっています。そのうち，グループ会社ではない国内企業に委託している割合は、国内グループ会社以外（大企業）の 10.8%と、国内グループ会社以外（中小企業）の 61.3%の合計 72.1%が占めています。

では、どのような場合に、外部へ委託するのでしょう。

「図 3-9 外部委託の理由（委託先：国内大企業）」から「図 3-11 外部委託の理由（委託先：海外企業）」に、その理由を示します。

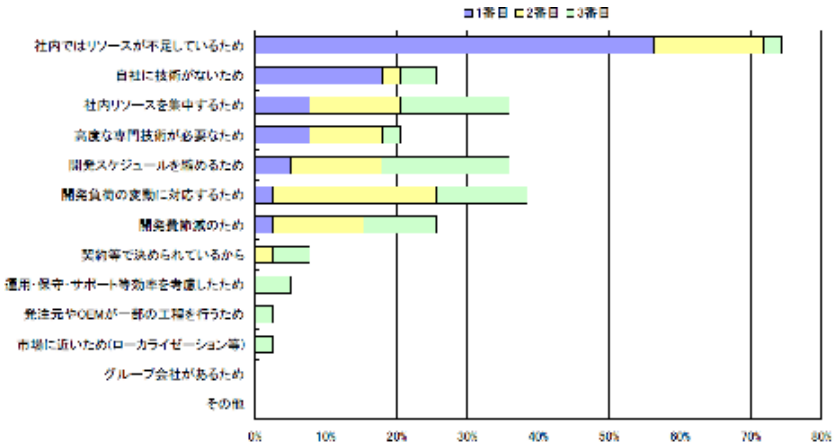


図 3-9 外部委託の理由（委託先：国内大企業）

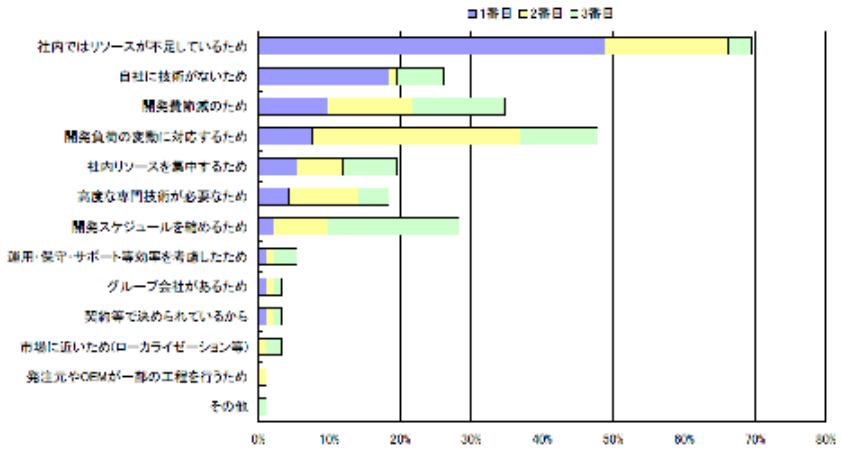


図 3-10 外部委託の理由(委託先:国内中小企業)

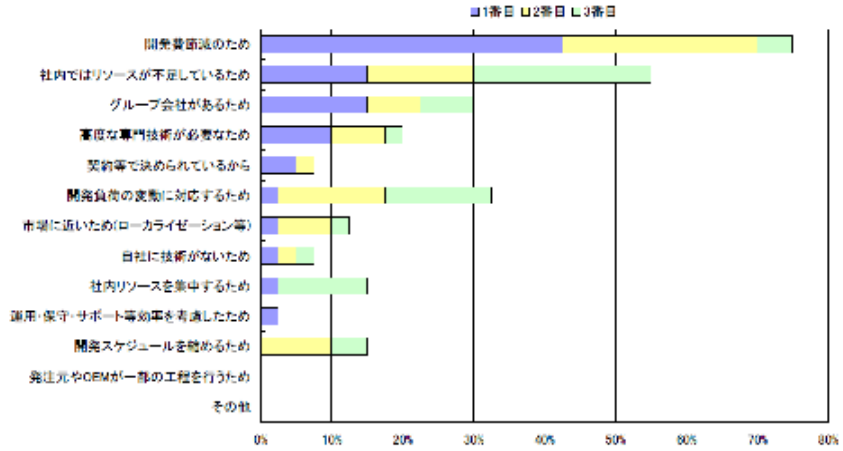


図 3-11 外部委託の理由(委託先:海外企業)

外部委託の理由としては、国内企業と海外企業で、明確に理由が異なっています。

国内企業の場合は、外部委託の理由として、「社内でのリソースが不足している」(大企業:56.4%、中小企業:48.9%)、「自社に技術がないため」(大企業:17.9%、中小企業:18.5%)が大きな理由となっています。つまり、自社内の技術不足を補うために外部へ委託していることが分かります。

海外企業の場合には、「社内ではリソースが不足しているため」(15.0%)、「高度な専門技術が必要なため」(10.0%)と、自社内の技術不足を補うために委託することもありますが、圧倒的に「開発費節減のため」(42.5%)に海外企業へ委託していることが分かります。ただし、気になる点として、海外企業の場合には、「グループ会社

があるため」（15.0%）という理由についても若干高い比率になっています。
　国内企業の場合、この「グループ会社があるため」という理由については、大企業で0.0%、中小企業で 1.1%という結果であることを考えてみても、本来は、外部委託したくはないもののグループ会社があるために、仕方なく海外のグループ企業へ委託している場合もあるかも知れません。

　海外企業の気になる点は別として、外部委託の主な理由は、国内企業には技術力を、海外企業にはコスト削減を求めていることが分かります。

国内企業	技術力不足の補てん
海外企業	コスト削減

図 3-12 外部委託の大きな理由

　では、外部委託したときに、課題となることは何があるでしょう。

「図 3-13　外部委託の課題（委託先：国内大企業）」から「図 3-15　外部委託の課題（委託先：海外企業）」に、外部委託の課題を示します。

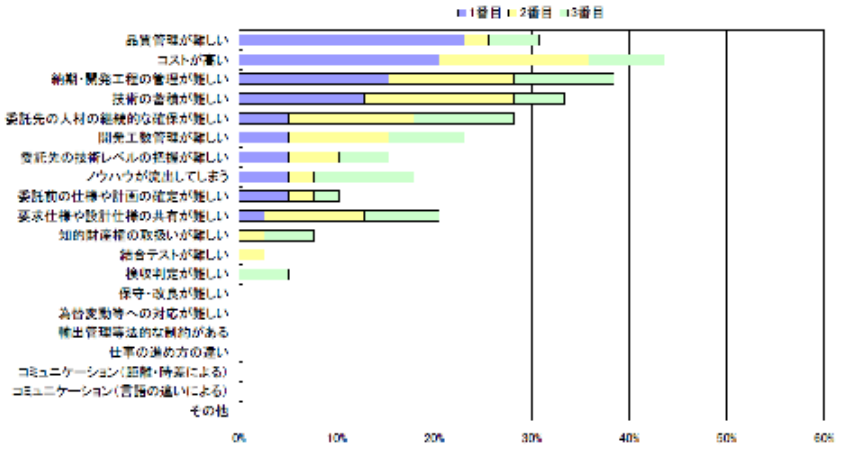


図 3-13 外部委託の課題（委託先：国内大企業）

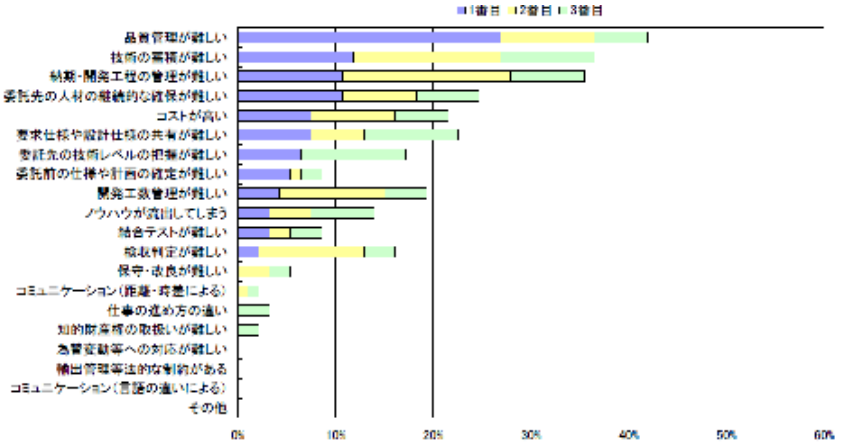


図 3-14 外部委託の課題（委託先：国内中小企業）

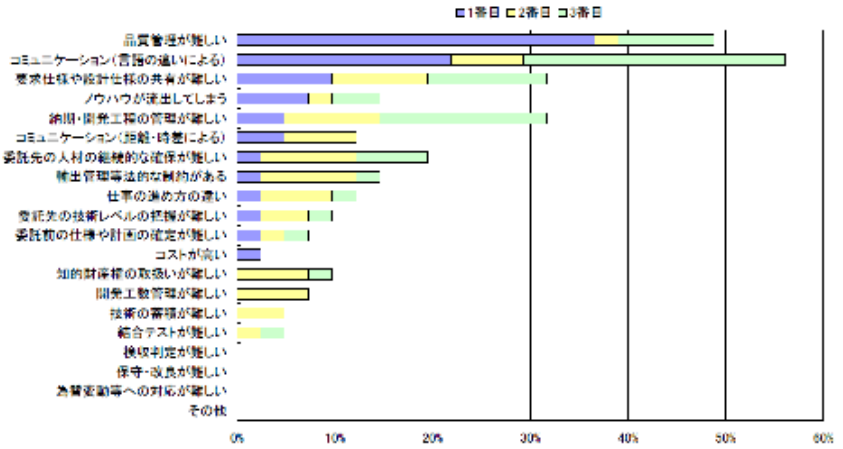


図 3-15 外部委託の課題（委託先：海外企業）

外部委託の課題としてもっとも多いのが、国内企業および海外企業すべての結果において、「品質管理が難しい」（大企業：23.1%、中小企業：26.9%、海外企業：36.6%）がトップに挙げられています。

また、「納期・開発工程の管理が難しい」（大企業：15.4%、中小企業：10.8%、海外企業：4.9%）という課題も上位に挙げられており、企業内に不足している技術力を外部に求めなければいけないものの、マネジメントが難しい現実との間のジレンマがあるようです。このため、企業は、外部委託する際の品質管理、工程管理などのマネジメント部分の強化を行っていく必要があります。

もう 1 点は、開発コストを抑えるために、海外企業へ委託する場合には、「コミュニケーション（言語の違いによる）」（22.0%）、「コミュニケーション（距離・時差）」

（4.9%）、「要求仕様や設計仕様の共有が難しい」（9.8%）などの情報共有も含めたコミュニケーションの課題が大きな割合を占めています。ミスコミュニケーションによる組込みソフトウェアの品質低下は大きな課題のひとつと言え、単に優秀なブリッジ SE 等の人材の配置だけではなく、距離や時差を考慮した情報共有やコミュニケーションの取り方にまで踏み込んだ仕組み作りが必要となります。

また、海外企業へ委託する際には、「ノウハウが流出してしまう」（7.3%）という課題を持っている企業も少なくありません[6]。このため、国内外に関わらず、自社内における貴重な財産である技術情報を含めたノウハウを流出させないための施策は、常に考えておかなければいけません。

3.5　組込みソフトウェアの受託状況

「図 3-16　受託の有無と発注元別売上げ比率」に、受託の状況を示します。

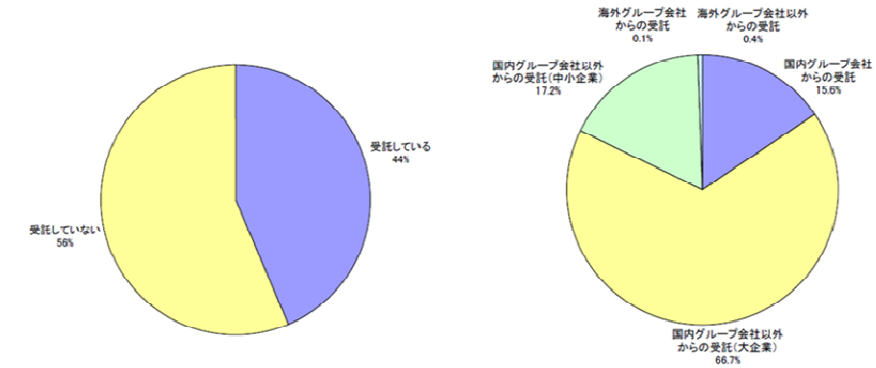


図 3-16　受託の有無と発注元別売上げ比率

組込みソフトウェアの受託の有無としては、「受託している」が 44%あり、受託元としては、圧倒的に「国内グループ会社以外からの受託（大企業）」が多く、66.7%もあります。大企業では、一部あるいは全部を外部委託することが多いと言え、「3.4 外部委託状況」のデータも含めて考えると、委託された業務に対し、その業務を受託した中小企業は、優れた技術を有している企業が多く、中小企業の活性化が、ひいては組込みソフトウェア業界そのものの活性化に繋がるとも言えそうです。

[6] 実態調査結果では、あくまでも課題として捉えているだけであり、本当に、ノウハウが流出してしまった結果ではありませんので、ご注意下さい。

しかしながら、課題として挙がっているように、中小企業がしっかりとした管理体制を持っていない可能性が高いとも言えます。このため、中小企業に対するマネジメント方法の浸透が、今後の組込みソフトウェアの品質面における強化のポイントとなってくるかも知れません。

3.6 外部調達のミドルウェア

「図 3-17 プロジェクトで利用したミドルウェア」に、プロジェクト内で使用したミドルウェアの割合を示します。

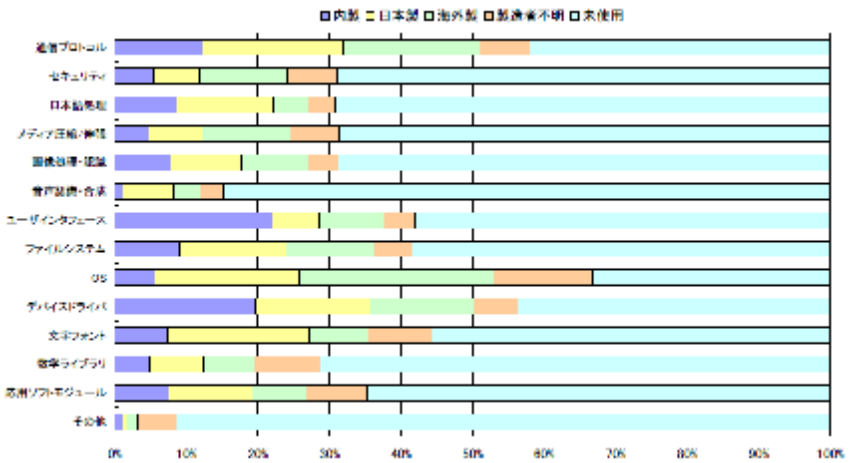


図 3-17 プロジェクトで利用したミドルウェア

組込み OS は、限られたリソースで動作させる必要がある点と、必要となる機能が製品によって異なる点から機能は最小限しか持っていないため、必要とされる機能ごとに、ミドルウェアを利用することになります。

「図 3-17 プロジェクトで利用したミドルウェア」から各ミドルウェアの内製と外部調達（日本製+海外製+製造者不明）の割合を抜き出したものを、「表 3-1 ミドルウェアの外部調達割合」に示します。

表 3-1 ミドルウェアの外部調達割合

ミドルウェア	内製	外部調達	外部調達割合
通信プロトコル	12.6%	47.0%	78.9%
セキュリティ	5.5%	26.3%	82.7%

ミドルウェア	内製	外部調達	外部調達割合
日本語処理	8.7%	22.4%	72.0%
メディア圧縮/伸張	4.9%	27.4%	84.8%
画像処理・認識	8.2%	24.6%	75.0%
音声認識・合成	1.1%	14.2%	92.8%
ユーザインタフェース	22.4%	20.3%	47.5%
ファイルシステム	9.3%	33.4%	78.2%
OS	6.6%	72.6%	91.7%
デバイスドライバ	21.3%	39.9%	65.2%
文字フォント	7.7%	38.8%	83.4%
数学ライブラリ	4.9%	24.1%	83.1%
応用ソフトモジュール	7.7%	28.4%	78.7%
その他	1.1%	7.6%	87.4%

この表を見ると、どのミドルウェアも、内製より外部調達の割合が、圧倒的に多い状況となっていることが分かります。

しかし、これらのミドルウェアの中で、ユーザインタフェースだけは、内製率が高い状況となっています。これまでも、組込みソフトウェアは、多くが機器に組み込まれると説明してきたとおり、表立って見える部分は極めて少ない状況です。このため、ユーザインタフェースは、組込みソフトウェア開発会社にとって、自社の特徴を活かしたものを作っていることが考えられ、内製している比率が高くなっているのではないかと思われます。また、少ないものであるがために、共通化されたユーザインタフェースが存在しておらず、内製せざるを得ないのが現状だと思われます。

ただし、例えば、携帯電話のように、ある程度決められた仕様のもとでユーザインタフェースを作らなければいけないこともあります。このような場合には、定められたユーザインタフェースのミドルウェアを利用することがあると言えるでしょう。

それでは、ミドルウェアを外部から調達している理由も見てみましょう。（「図 3-18　ミドルウェアを外部から購入・使用した理由：日本製」および「図 3-19　ミドルウェアを外部から購入・使用した理由：海外製」参照）

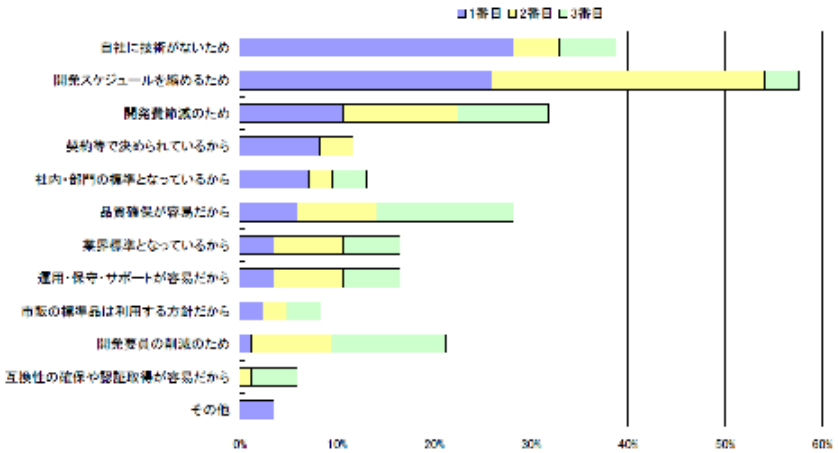


図 3-18 ミドルウェアを外部から購入・使用した理由：日本製

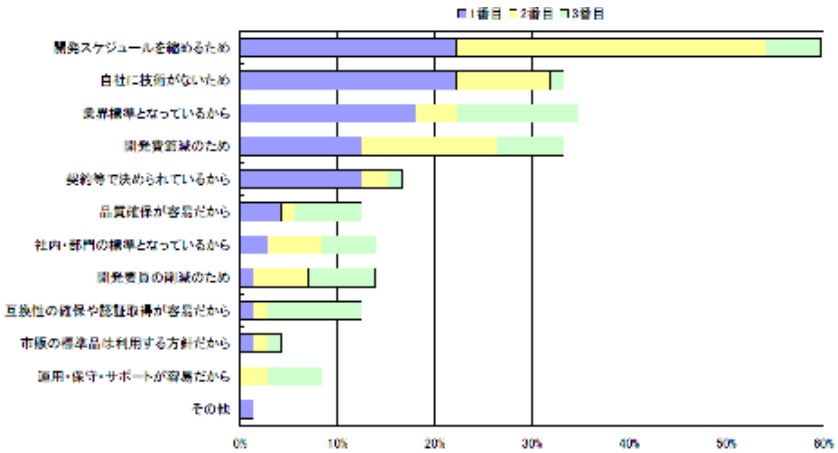


図 3-19 ミドルウェアを外部から購入・使用した理由：海外製

ミドルウェアを調達する理由として、日本製、海外製共に、「自社に技術がないため」（日本製：28.2%、海外製：22.2%）[7]が挙げられています。また、自社で内製するには、「開発スケジュールを縮めるため」（日本製：25.9%、海外製：22.2%）や「開発費節減のため」（日本製：10.6%、海外製：12.5%）の理由からも、時間的にも資金的にも余裕がないため、ミドルウェアを外部から調達する必要があるようです。

[7] ここでの数値は、1 番目に選択されている理由の数値を使用しています。

また、「契約等で決められているから」（日本製：8.2%、海外製：12.5%）という理由が多いのも特徴的です。これは、特に海外製のミドルウェアで上位に位置づけられている「業界標準となっているから」（日本製：3.5%、海外製：18.1%）という理由にも関係しており、ミドルウェアが契約上（あるいは業界標準として）定められているため、そのミドルウェアを利用する企業では、確実にその対応を入れなければいけないことになります。もちろん、決まったミドルウェアを利用し続けることによって、企業にとってのノウハウが蓄積されることも考えられますが、逆に、ミドルウェアが変更された場合には、蓄積されたノウハウを、一から築き直す必要も出てくることになります。

このような契約上あるいは業界標準としての縛りによる制約条件というのも、品質面では重要なことと言えます。

次に、ミドルウェアを調達したときの課題についても見てみることにしましょう。

（「図 3-20　ミドルウェアを外部から購入・使用したときの課題：日本製」および「図 3-21　ミドルウェアを外部から購入・使用したときの課題：海外製」参照）

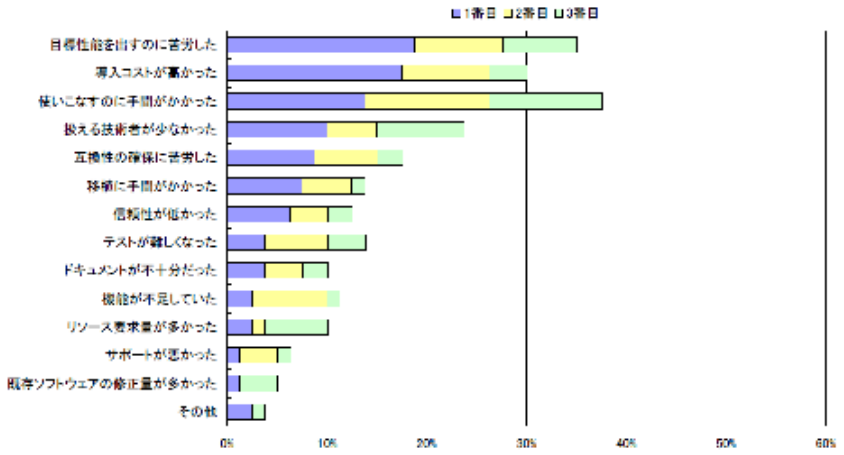


図 3-20 ミドルウェアを外部から購入・使用したときの課題：日本製

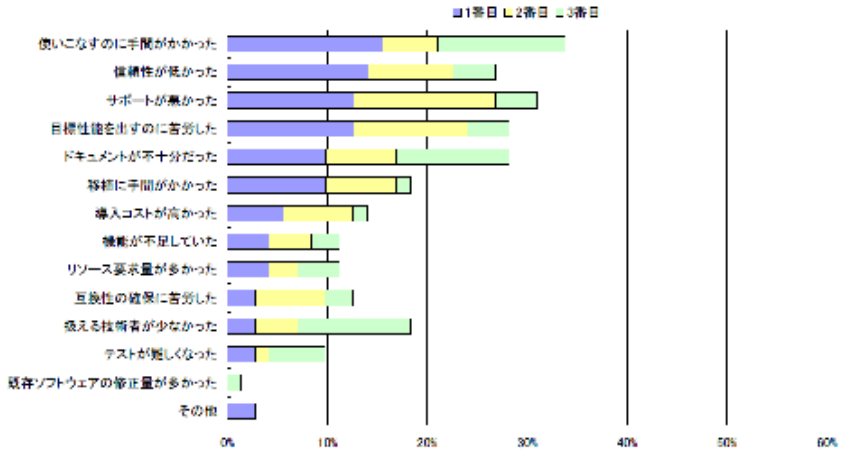


図 3-21 ミドルウェアを外部から購入・使用したときの課題：海外製

これらは、ミドルウェアを外部調達し、その課題として挙げられているものですが、その中でも目立つものとして、「使いこなすのに時間がかかった」（日本製：13.8%、海外製：15.5%）と、「目標性能を出すのに苦労した」（日本製：18.8%、海外製：12.7%）といった課題が挙げられます。

さまざまな理由によってミドルウェアを使わなければいけない状態になったものの、それを使いこなすあるいは目標性能を出すまでに多くの時間や労力を割いていることが分かります。この点からも、外部調達したミドルウェアに対し、どのように自社内の組込みソフトウェアに取り込んでいくのかは大きな課題であると言えます。

また、海外製に対しては、「信頼性が低かった」（14.1%）、「サポートが悪かった」（12.7%）、「ドキュメントが不十分だった」（9.9%）など外的要因による課題も挙げられており、外的要因による品質低下をどのように防いでいくのかも、ミドルウェアを利用する企業にとっては、重要な課題であると言えます。ただし、これらの課題については、言語の違いなどによるコミュニケーションの課題である可能性もある点を考慮したものでなければいけません。

3.7 利用した手法・技法

「図 3-22 プロジェクトで利用した手法・技法」は、プロジェクト内で利用した手法や技法についてまとめたものです。

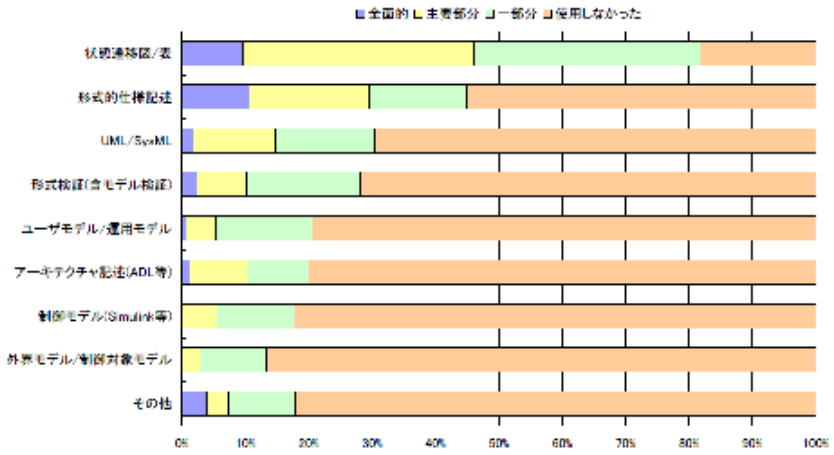


図 3-22 プロジェクトで利用した手法・技法

全面的に利用している手法や技法は、ごくわずかであることが分かります。

しかし、「状態遷移図／表」については、全面的、主要部分、一部分をすべて合わせると 81.9%も利用されています。組込みソフトウェアの場合、ハードウェアを制御することが多いため、その状態遷移を把握するために有効に活用されていると考えて良いと思いますが、その他の手法などは、使われることが少ないのが現状のようです。

必ずしも特定の手法や技法を使う必要はありませんが、「3.1 開発規模」にも記載したとおり、年々、開発規模が大きくなってきている状況で、手法や技法を使わずに、組込みソフトウェアの開発を行っていくことは、不具合等が潜在してしまう可能性を高めてしまうことからも、今後、特にテスト工程は、ますます難しくなってくることが予想出来ます。手法などは、定められた方法で作成していくため、そこからのイレギュラーなパターンを導き出しやすいのですが、もし、それが無ければ、すべてを網羅的に確認していく必要があります。

しかし、現実的には、すべてを網羅的に確認することは出来ません。

そこで、確認すべき項目を抽出していくことになりますが、この抽出作業は、あくまでも不具合が潜在する確率を下げていくだけのことであり、漏れてしまうケースもあり得ます。このため、手法などを利用し、現在行っている確認から、更に確率を下げていくようにしていく必要があると考えられます[8]。

[8] プログラム上から不具合をゼロにすることは出来ないと言われており、このため、どれだけ確認をしても不具合は潜在してしまうと言われていますが、極限まで不具合の潜在を下げることが出来るのかが重要になります。

3.8　使用ツール

続いて、プロジェクト内で利用しているツールについて見てみましょう。（「図 3-23　プロジェクトで使用したツール（内製+日本製+海外製+不明）」参照）

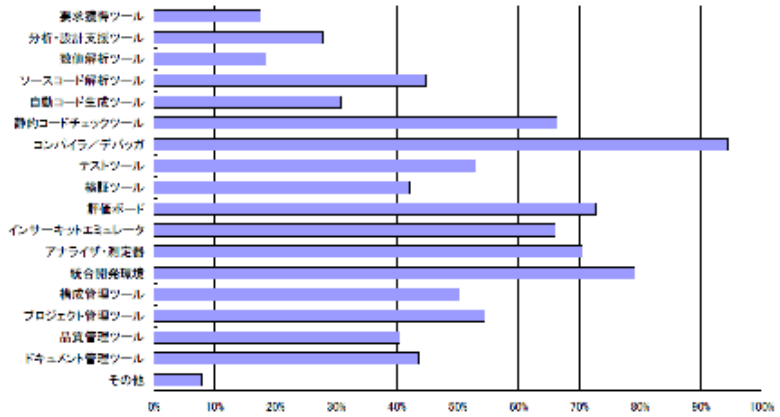


図 3-23　プロジェクトで使用したツール（内製+日本製+海外製+不明）

コンパイラ/デバッガや開発統合環境などソフトウェアエンジニアが使用するツールが多く使用されていますが、プロジェクト管理、品質管理などの管理ツールの利用の割合も 40%~50%と過去と比較しても増加してきているようです。

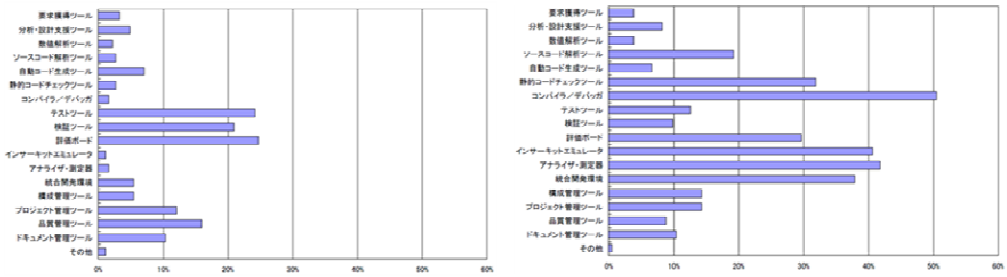


図 3-24 使用ツールの内製と外部調達の比較

「図 3-24 使用ツールの内製と外部調達の比較」は、左側が内製ツール、右側が日本製の外部調達ツールです。[9]

[9] グラフは、日本製のもののみを使用していますが、海外製の外部調達ツールもほぼ同じような傾向となっています。

「コンパイラ／デバッガ」および「総合開発環境」、「インサーキットエミュレータ（In-circuit emulator：ICE）」、「アナライザ・測定器」は、外部からの調達が多く、「評価ツール」および「検証ツール」、「評価ボード」については、内製が多いことが分かります。ただし、「評価ボード」については、内製だけではなく、外部調達でも多く、自社で組込み製品を作っている企業では、内製したものを使い、逆に、組込み製品を作っていない企業では、外部調達により評価ボードを利用しているためだと思われます。

内製が多い「評価ツール」や「検証ツール」については、自社の組込みソフトウェアの特質に合わせる形でツールを作ることが多く、そのため、多くのノウハウが蓄積されている部分とも言えます。逆に言うと、ノウハウが詰まっているために、必然的に汎用性が低くなり、他への転用が利かなかったり、新しいことへの対応に手間取ったりすることも少なくありません。なお、各種ツールは、組込み製品の根本的な品質に直結する大事な部分でもあることから、ツールに対して、高品質のツールでなければいけない（あるいは求められる）ことは言うまでもありません。

ツールを外部から調達している場合の理由を、以下に示します。

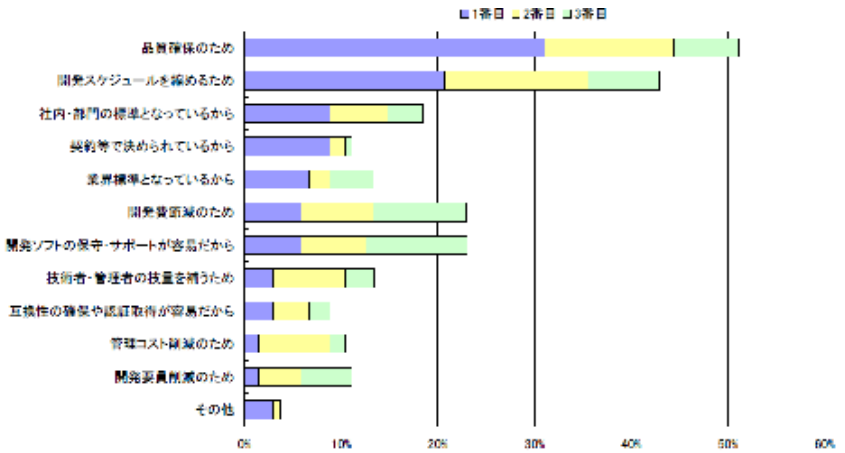


図 3-25 ツールを外部から購入・使用した理由：日本製

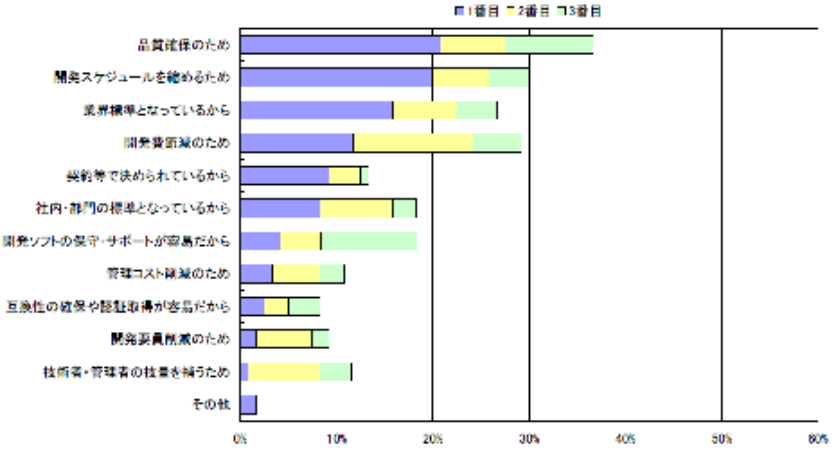


図 3-26 ツールを外部から購入・使用した理由：海外製

日本製と海外製で多少の違いはあるものの、多くの理由が「品質確保のため」（日本製：31.1%、海外製：20.8%）と「開発スケジュールを縮めるため」（日本製：20.7%、海外製：20.0%）となっています。つまり、外部から調達するツールに対しては、そのツールの絶対的な品質のもとで、ツールを利用し、それによって、高品質な組込みソフトウェアと開発効率を向上させていきたいという思いがあることが分かります。

しかしながら、「図 3-27 ツールを外部から購入・使用したときの課題：日本製」や「図 3-28 ツールを外部から購入・使用したときの課題：海外製」に示すように、いくつかの課題があります。

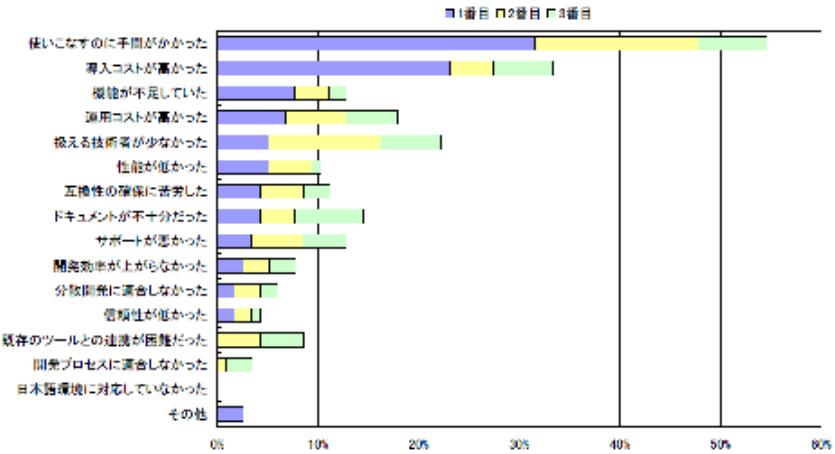


図 3-27 ツールを外部から購入・使用したときの課題：日本製

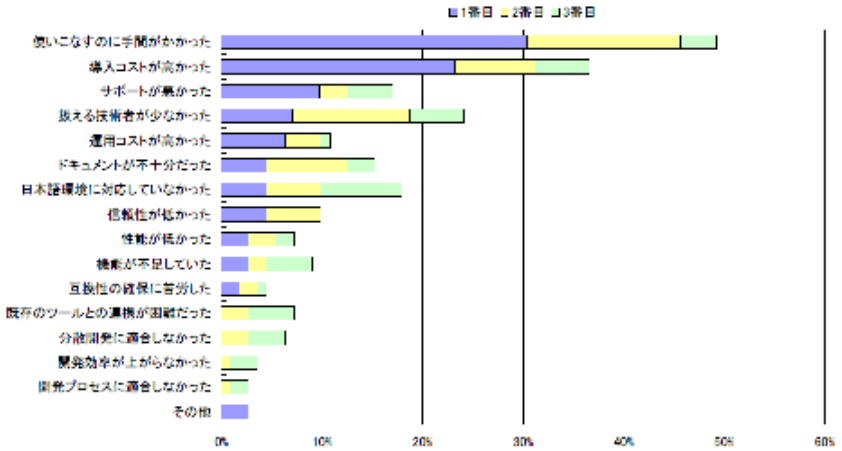


図 3-28 ツールを外部から購入・使用したときの課題：海外製

日本製、海外製のどちらのツールでも、多くの理由が、「使いこなすのに手間がかかった」（日本製：31.6%、海外製：30.4%）、「導入コストが高かった」（日本製：23.1%、海外製：23.2%）、「運用コストが高かった」（日本製：6.8%、海外製：6.3%）となっています。

「使いこなすのに手間がかかった」という課題が示すように、本来のツールを使用目的である「開発スケジュールを縮めるため」が、実現出来ていない（あるいは実現することが難しい）状態になっていることがうかがえます。

「導入コストが高かった」についてですが、多くのツールは、出荷数量が、決して多くはありません。このため、比較的高い価格に設定されていることが多いのが現状です。ツールを導入する企業にとっては、その高い価格が課題となっていることが分かりますし、「使いこなすのに手間がかかった」ために、初期稼働までの人件費等によって、必然的に、導入コストを押し上げてしまうこともあると考えられます。

また、ツールの中には、使用される環境によって、都度設定をし直す必要のあるものも少なくありません。このような場合には、その環境に合わせ設定変更を行わなければならず、必然的に運用コストが高くついてしまうことになってしまうのです。
これまでのデータが示しているように、組込みソフトウェア開発におけるコスト削減は、重要な課題となっています。このため、ツール利用によってコストを押し上げてしまうことは、決して望ましいことではありません。

また、「3.5 組込みソフトウェアの受託状況」にもあるように、海外製のツールでは、「サポートが悪かった」（海外製：9.8%）、「扱える技術者が少なかった」（海外製：7.1%）、「ドキュメントが不十分だった」（海外製：4.5%）という課題も挙げられています。海外製のもの（ミドルウェアやツール）に対しての原因のひとつが、使用している言語によるものなのかについては、今後、精査していく必要があるのかも知れません。

もし、言語によるツール利用の課題として存在しているのであれば、ツール自体のローカライズを行うか、もしくは、ツールを利用する要員に対する言語習得が必要になってくることが考えられます。

費用対効果としては、ツール提供会社によるツールのローカライズを待つことよりも、要員に対する言語習得を行わせた方が良いでしょう。また、言語習得については、ツールだけに限った話ではなく、「3.4 外部委託状況」や「3.6 外部調達のミドルウェア」にも関係することであり、海外とのやり取りが発生する状況においては必要不可欠なことであるため、今後、言語習得は、組込みソフトウェアに携わる人たちにとって、重要な要素のひとつになるかも知れません。

4. 組込みソフトウェア開発の課題と解決策

さて、ここまで組込み製品および組込みソフトウェアの現状を見てきました。これまでも、それぞれの項目で課題を挙げていますが、本章では、それらをまとめてみたいと思います。

4.1 設計品質の向上

その前に、実態調査結果から、企業が組込みソフトウェア開発の課題として挙げているものを見てみましょう。

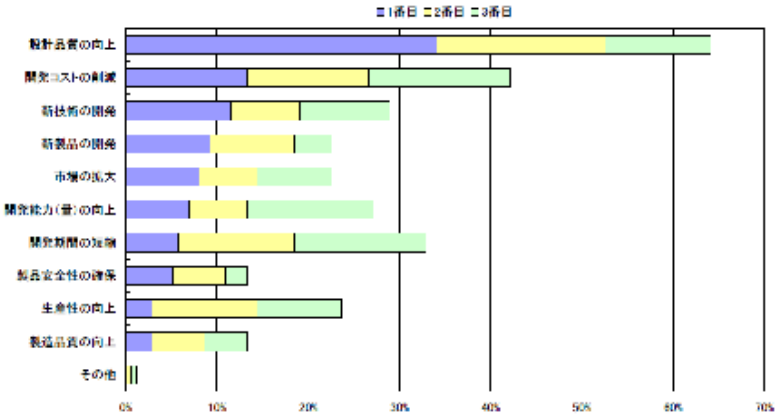


図 4-1 組込みソフトウェア開発の課題

まず、大きな割合であるのが、「設計品質の向上」であり 34.1%（2 番目：18.5%、3 番目：11.6%）あります。3 番目までの課題と合わせると、64.2%を占めており、「製品品質の向上」の 2.9%（2 番目：5.8%、3 番目：4.6%）と比較しても、格段に大きな割合を占めていることが分かります。

一般的に、不具合検出は、遅れれば遅れるほど、手戻り修正に要する時間とコストは増大することは多くの方が知るところとなっており、「開発コストの削減」（1 番目：13.3%、2 番目：13.3%、3 番目：15.6%）を行うためにも、「設計品質の向上」が最重要課題であると言えるでしょう。

解決策として、以下のようなことを聞いたことはありませんか？

- 設計および製造の不具合情報を確実に収集する（情報収集）
- 事実に基づき、収集されたデータから傾向を分析する（傾向分析）
- 原因を突き止め、その原因を分析する（原因分析）
- 未然に防止出来るように、課題を発見する方法を見つける（未然防止策）

これらの解決策は、重要ではあるものの難しい側面があります。ひとつには、それぞれの具体的方法が分からない点、もうひとつは、企業ごとのノウハウとして蓄積されていることであるため、外部が知ることが出来ない（外部と共有出来ない）点です。どちらも、企業内にデータの蓄積が無い場合には、知る由もありません。このため、難しいことになってしまうのです。また、これらの内容については、常に新しいもの（対策）へとアップデートしていく必要があります。環境の変化が激しい組込みソフトウェアの開発現場において、その時点では、良いとされているものでも、変化に伴ってやり方自体を見直す必要があるのです。

それらの点も含めて、以下のような解決策があります。

ひとつには、人材育成を行うことが考えられます。
設計を行うのは人間であるため、その人材を教育する必要は少なからず発生しますし、最低限必要なことです。人材育成については、次節の「4.2 人材育成」にて詳しく解説しますので、そちらを参照してください。

もうひとつは、手法・技法の導入、ツールの導入です。

「3.7 利用した手法・技法」や「3.8 使用ツール」でも記載しているとおり、まだまだ利用している企業が少ないのが現状です。ツールについては、導入コストの問題もあるため、難しい側面もありますが、手法・技法については、もちろん人的コストの面は考えなければいけませんが、導入することは検討する余地があります。

なぜ、手法や技法、ツールの導入が必要なのでしょうか？

設計に限らず、多くの場面で人が介在してしまうソフトウェアの開発現場では、ミスコミュニケーションによって不具合が入り込んでしまう可能性があり得るのが事実です。このため、可能な限り、人を介在させないようにするか、ミスコミュニケーションが発生しない仕組み作りが大切なこととなります。

手法・技法が、なぜミスコミュニケーションの低減に繋がるかと言うと、ひとつの方法を提示することによって、ステークホルダー間での言語による考え方の違い、曖昧さを最小限にしていくことに繋がるのです。特に、日本語は、曖昧さが多い言語でもあり、話し言葉の中や仕様の中でも、「だいたい」、「おおよそ」、「〇〇くらい」、「〇〇など」といった曖昧な言葉が出てきます。また、擬音語や擬態語といった読み手の受け取り方ひとつで変わってくるような表現もあります。これらの言葉をすべて排除出来る訳ではありませんが、特定の手法や技法によって、最小限に出来るのであれば、検討する価値はあるのではないでしょうか？

ミスコミュニケーションが発生しない仕組み作りとは、例えば、プロセス品質を向上させることも方法のひとつとして挙げられます。プロセスをしっかりと定着させ、そのプロセスを監視していくことで、設計品質を保つ環境を作り上げていくことに繋がっていくと考えられます。

ただし、この方法で間違えていけないのは、特定のプロセスだけが正しいものではないということです。一般的な開発プロセスモデルとして、ウォーターフォールモデルを採用する企業が多いのですが、組込みソフトウェアを利用している企業で、純粋なウォーターフォールモデルを適用出来ている企業は、実は少なく、そのため、開発プロセスを、無理やりこのウォーターフォールモデルに合わせ込んでいる可能性もあるのでないかと考えています。

特に、「3.4 外部委託状況」のような外部へ委託する場合には、委託先企業が、このような状態に陥っていないかを把握する必要があります。もちろん、委託先企業の開発プロセスを、自社の開発プロセスに合わせるようにすることも出来るかも知れませんが、企業規模、企業風土の異なる場合には、どうしても無理が出てきますので、そのプロセス品質は疑わしいものになるかも知れません。

このため、その企業風土も含めた組織体質に合った開発プロセス（設計に関わるプロセスも含まれます）を明確にし、そのプロセスで、何をするのか、どのようにするのかを決めていく方が良いでしょうし、また、委託先企業との間では、お互いの開発プロセスを明確にした上で、相手の開発プロセスを尊重させ、品質を確保出来る体制を考えていく方が良いと思います。

組込み製品のうち、ハードウェアを作る企業は、どうしてもメーカといった大企業が、発注元であることが多いでしょう。

しかし、その一部分である組込みソフトウェアを作る企業は、大企業が自社で組込みソフトウェアを作る場合以外では、メーカほどの大きな企業ではないことは、「3.4 外部委託状況」や「3.5 組込みソフトウェアの受託状況」からも明白です。そのような場合、企業規模が異なる体制のもとで、同じプロセスで物事が進むとは考えない方が無難なはずですが、どうしても、自分たちの開発プロセスを優先させ、それを外部にも強要[10]していることもあり得ます。このような場合には、どこかに歪が生まれる可能性は少なからずあり得ると考えられるのではないでしょうか。

4.2 人材育成

組込みソフトウェア技術者の不足率は減少傾向にあるものの、それでも、まだ不足

[10] 強要という言葉は悪いですが、委託先の中小企業の中には、開発プロセスさえ明確に定められていない場合も、少なからずあるため、仕方なく合わせさせている場合もあり得ます。

しています（「2.3 組込みソフトウェア技術者数」参照）。不足している人材を確保するためには、大きく 2 つの方法があり、1 つは「外部より調達する」ことと、もう 1 つは「自社で育成する」ことがあります。

先の「外部より調達する」については、外部委託といった形で対応することも出来ますし、中途採用を行うことで達成出来ますが、たとえ外部から調達しても、その人たちが、期待している能力を発揮出来ないことも考えられるため、やはり「自社で人材を育成する」ことが重要になってきます。

「自社で人材を育成する」ことにも、大きく「新入社員教育」と「社内人材教育」の 2 つに分類することが出来ますが、どちらの場合であっても、その人たちが持っていなければいけないスキルを明確にすることが、第一に必要なこととなります。

経済産業省では、組込みソフトウェアの開発力強化の一環として、組込みソフトウェア開発に関する最適な人材育成や人材の有効活用を実現するための指標や仕組みとして、組込みスキル標準（ETSS：Embedded Technology Skill Standards）を定めています。組込みスキル基準では、組込みソフトウェア開発に必要なスキルを明確化・体系化したものであり、指標として次の 3 つの要素を提供しています。

表 4-1 組込みスキル基準要素

要素	説明
スキル標準	組込みソフトウェアに関する技術スキルを、技術要素、開発技術、管理技術の 3 つのスキルカテゴリごとに階層的に整理しており、各スキル項目に対して、4 段階でスキル診断することで、技術者のスキル分布を定量的に可視化することが出来ます。
キャリア標準	組込みソフトウェア開発に関して、10 の職種と 12 の専門分野に分類。要求される責任性と専門性の度合いを 7 段階のレベルで表します。 各職種が担う職掌を果たすためには、どのような要素（スキル、知識、責任性、･･･など）が必要なのかを明確化しています。
教育研修標準	組込みソフトウェア開発のスキルアップ・キャリアアップを支援する教育カリキュラムのフレームワークを定義しています。

スキル標準の技術要素は、通信、マルチメディア、ストレージ（記憶）、などの製品開発に必要な構成要素、開発技術は、要求分析、設計、実装、テストといった開発の流れ、管理技術は、プロジェクト管理と開発プロセス管理の流れに従って分類されています。

「図 4-2　ETSS スキル基準（IPA SEC ホームページより）」は、これらの 3 つのスキルカテゴリを表した図となっています。

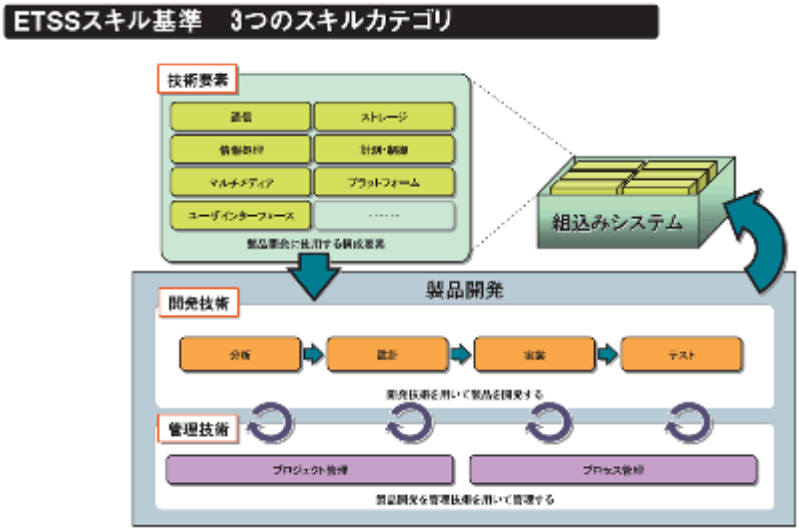


図　4-2　ETSS スキル基準（IPA SEC ホームページより）

キャリア基準は、各職種のレベル毎に必要な関連要素を明確化するものであり、開発に必要な職種と責任の範囲を定義しています。職種の例としては、プロダクトマネージャ、プロジェクトマネージャ、ドメインスペシャリスト、システムアーキテクト、ソフトウェアエンジニア、ブリッジエンジニア、サポートエンジニア、QA スペシャリスト、テストエンジニアといった組込みソフトウェアで必要とされる職種に分類されています。

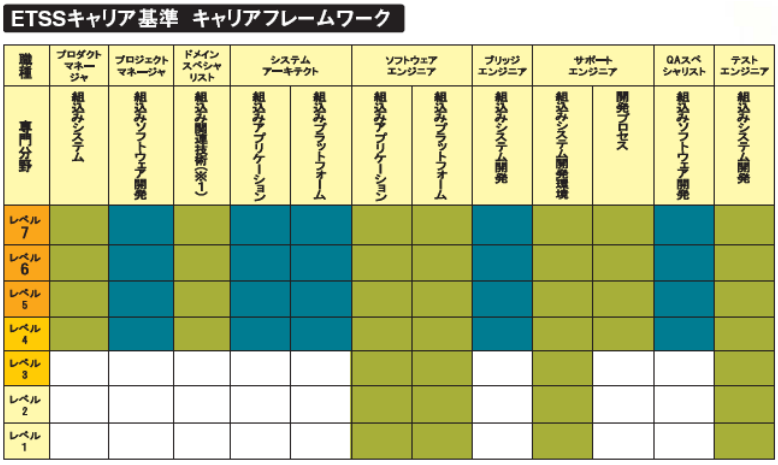


図　4-3　ETSS キャリア基準（IPA SEC ホームページより）

教育研修標準は、スキルアップ・キャリアアップを支援する教育カリキュラムのフレームワークだけではなく、組込みソフトウェア未経験者向けのガイドラインも対象

としています。

図 4-4 ETSS 教育研修標準（IPA SEC ホームページより）

それでは、ETSS の枠組みを、どのように活用していけばよいのでしょうか？

うまく活用するためには、マネジメントシステムの教育について考えるのが良いのではないかと思っています。
以下の図は、マネジメントシステムでの教育（ISO9001 6.2.2.力量、教育・訓練及び認識より）を段階的に示したものです。

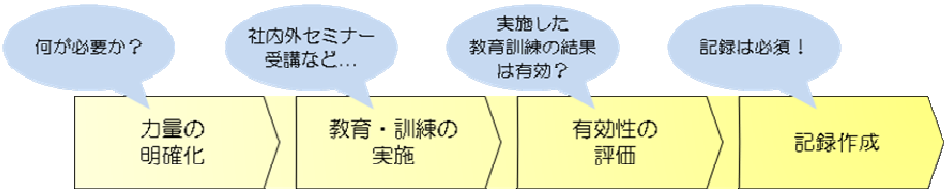


図 4-5 力量、教育・訓練及び認識

まず、最初の「力量の明確化」というのは、ETSS におけるキャリア基準を利用し、どの部署に、どのような職種の人材が必要であるのかを決定します。その後、スキル基準を利用し、その職種での必要なスキルを明確にしていくというやり方です。

一般的には、このスキルと給与が一緒になっていることが少なくありません。もちろん、それがダメということではありませんが、スキルと給与を合わせた場合には、エキスパートタイプ（特定のスキルのみが得意なタイプ）の人材と、ジェネラルタイプ（様々なスキルに対し、満遍なくこなせるタイプ）での違いを考慮しなければ、少なからず反発が出てくることが予想されるため、利用することが難しくなってくる可能性があることは留意しておくべきでしょう。

スキルが明確になったところで、実際に教育や訓練を行うことになりますが、ここでも、一工夫が必要です。

例えば、ソフトウェアエンジニアを育成する場合、どうしてもプログラミング言語を習得させることが多くなると思いますが、品質やテストに関する教育を行った後で、

プログラミング言語の習得を行わせても良いのではないかと考えています。

不具合の原因を追及すると、多くが「設計ミス」や「考慮不足」が挙げられます。
「設計ミス」については、「4.1 設計品質の向上」でも挙げられているとおりですので、そちらを参照して頂ければと思いますが、もう一方の「考慮不足」については、例えば、初期化漏れや限界値や制限値等の比較文の間違いなどが挙げられます。これらの問題は、すべて人為的なミスになりますので、教育によって、少なからず軽減させていくことが可能です。

なぜ可能なのかというと、簡単に言えば、プログラムを作るときに「気にする」からです。プログラムを作成する場合、個々のソフトウェアエンジニアによりさまざまなプログラムが出来あがります。このとき、初期化漏れや限界値、制限値[11]を気にしているか、気にしていないかは、品質面で大きな差となって表れてきます。これは、経験上、間違いやすいと思われる部分ではありますが、特に、新入社員のような経験が少ない人たちにとっては、それらの間違いやすいことを教育されているだけで、大きな違いが生まれてくることになるのです。

このように、何かのスキルを身につける場合には、他のスキルの中で、先に覚えておいた方が良いものが少なからず存在しているはずであり、これらの順序も考慮しながら、教育を行うようにしてみても良いのではないかと思われます。

有効性の評価については、自己申告だけでも構いませんが、実務の中で、上司や先輩、同僚などから評価（他者からの評価）を行うようにしても良いのではないでしょうか？ただし、自己申告であれ、他者からの評価であれ、主観的な評価が含まれてしまう可能性があることも考慮すべきであり、どちらか一方だけの評価にしてしまうと、評価が乖離している個所、つまり、何か問題があるかも知れない点を見逃すことにもなりかねません。評価が乖離していなくても問題が無いことはありませんが、評価が乖離している個所に比べ、格段に問題がある可能性は低くなります。このため、少なくとも評価が乖離している個所を正しく把握し、その乖離箇所について、本人にも認識させていくこともスキルアップを図るためには必要なのではないかと思われます。

さて、ここまでで、誰がどのようなスキルを身につけたのか、あるいは、必要スキルのうち、どの程度までの熟練度があるのかを把握することが出来ると思います。その把握出来たスキルを、確実に蓄積していく（記録する）必要があります。

組込みソフトウェア開発会社に限った話ではありませんが、異なる部署にいる必要スキルのある人材を有効活用出来ていない例が良く見受けられます。もちろん、その

[11] 良くある不具合の代表例として挙げていますが、初期化漏れや限界値、制限値だけではありません。あくまでも代表的な例として捉えて下さい。

人材が、他の作業を行っている場合には仕方がありませんが、そうではない場合も、少なからずあります。もし、記録として存在しているのであれば、プロジェクト等で要員を確保する際にも有効利用することは出来るはずです。

また、いくらスキルを身に着けていたとしても、状況等の変化により再教育・再訓練が必要になることもあることもあります。

しかし、再教育・再訓練では、教育する内容が、新たな教育とは大きく異なるはずです。例えば、過去に C++のプログラミングスキルを身につけた人材がいた場合、その過去の時点で利用していた開発環境と、現時点での開発環境では大きく異なる可能性は少なくありません。このとき、C++のプログラミングそのものの教育を行うよりも、開発環境についての教育を行う方が有効です。このように、記録をしておくことで、より有効な方法で、教育を行うことが出来れば、コスト面でのメリットも出てくることでしょう。

ETSS のような規定を利用し、技術者のスキルやキャリアを定量化することにより、企業としては、人材の状況が把握できるようになり、効率的な人材配置及び人材育成を可能に出来るでしょうし、組込みソフトウェア技術者としては、自分のキャリアやスキルがどのような状態にあるのかを客観的に把握できるため、どの点を伸ばせばよいかの目標設定がしやすくなり、モチベージョンの向上も期待することが出来ます。

4.2.1 新人社員教育

新入社員を教育することは、企業にとって重要な役目のひとつでもあります。

社会人としてのマナーをはじめとして、それぞれの職種に合った教育をしていくことでしょう。期間は、企業によりさまざまなものではありますが、それぞれの職種に対する指針は必要となってくることは間違いありません。

このような場合に、ETSS のスキル標準を活用し、新入社員へ教育する科目を選定しても良いかも知れませんし、また、本章に出てくるいくつかの課題に対して教育するというのも有効では無いかと考えられます。

4.2.2 スキルチェンジ

組込みソフトウェア技術者へのスキルチェンジは、「2.3 組込みソフトウェア技術者数」にもあるように行われていないあるいは行われてはいるものの小さいものでしかないことが分かります。このため、組込みソフトウェア開発の現場では、他の技術職からの移動は、出来ていない状況であると言わざるを得ません。

しかし、人間にはどうしても職種に対する向き不向きがあるため、特定の技術職だけを行わせていることは、本人にとっても、企業にとっても良いものではありません。

このため、スキルチェンジが出来る環境を作ることも重要となってくると考えられますが、スキルチェンジについては、多くの部署がある環境や役割を分担出来る環境

を有する企業にとっては、出来ることであっても、少人数の企業では、難しい側面があるのも事実です。

しかし、ジョブローテーションのようなスキルチェンジではなく、その職種におけるスキルアッププランを考えることは、どのような会社であっても実現は可能だと思われ、その点でも ETSS の枠組みを利用することが可能です。

例えば、テストエンジニアを見ると、ETSS では、技術要素については、開発対象のプロダクトによって異なりますが、開発技術では、テストに関する知識やスキルの他にも、プロジェクトマネジメント（品質マネジメント、コミュニケーションマネジメント）やプロセスマネジメント（構成管理・変更管理）、パーソナルスキル（リーダーシップ、コミュニケーション、問題解決）などが求められています。
ところが、人によっては、パーソナルスキルは低いという人材が少なからず存在しており、そのようなときに、テストエンジニアとして、すべてのスキルを満たさなくてはいけないということではなく、テストに関する知識やスキルを伸ばすような形でスキルアップさせても良いかも知れません。このように、ETSS を活用することも出来るのではないでしょうか？

なお、「図 4-6　組込みソフトウェア開発課題解決の有効手段」は、事業責任者が考えている組込みソフトウェア開発における課題解決として有効だと考えている手段を示しています。

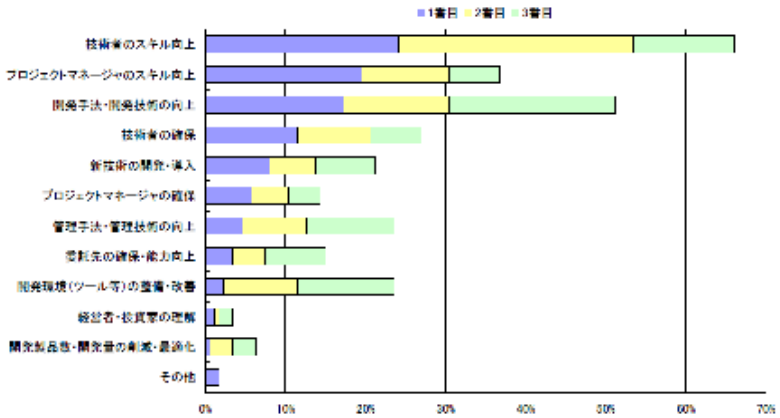


図 4-6　組込みソフトウェア開発課題解決の有効手段

事業責任者も、技術者のスキル向上が組込みソフトウェア開発の課題解決のためには、必要なことであるとの認識があります。このため、技術者のスキル向上のために、組込みソフトウェア開発会社によりさまざまな取り組みがされていることでしょう。

本調査結果からは、組込みソフトウェア開発会社が、どのような取り組みを行っているのかは見つけ出すことが出来ませんでしたが、ETSS を活用したスキルアッププランを策定することも、解決策のひとつとなり得るかも知れません。
しかし、「図 4-7　SEC 成果の導入状況」の結果からも分かるように、組込みスキル

標準（ETSS）については、導入済みが 8.0%、導入検討中が 13.8%と、両方を合算しても、21.8%と、ETSS を活用している企業は、まだまだ少ないことが分かります。また、ETSS に限らず、その他の SEC の成果についても同様の結果となっています。

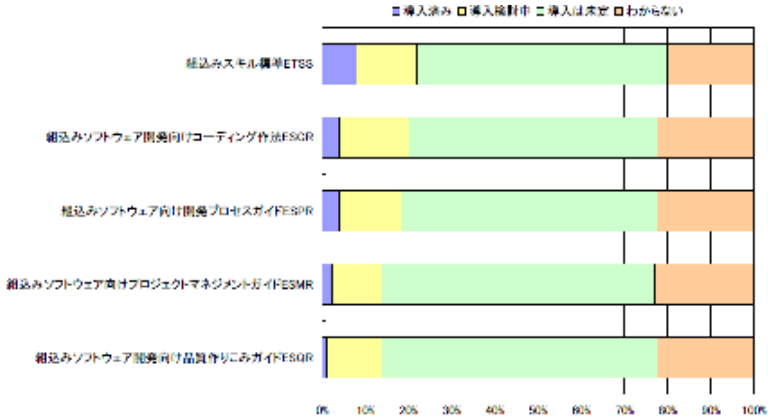


図 4-7 SEC 成果の導入状況

これらの根本的な原因を突き止め、ETSS のような SEC としての成果に対する認知度の向上および具体的な策の提示については、SEC にとっての大きな課題と言えそうです。

4.2.3 語学力

語学力については、ETSS では、スキルとして定義はされていませんが、もちろん、コミュニケーションスキルの一部として、語学力が必要であると言えるでしょう。

では、どの程度の語学力が必要になるのでしょうか？
TOEIC の運営機関である国際ビジネスコミュニケーション協会によれば、おおまかに分けて、以下のような語学力が必要であるというデータが出ています。

表 4-2 部門ごとの TOEIC での必要得点

部門	必要得点
技術部門	650～800
営業部門	600～750
海外部門	550～750

この表を見ると、平均すると、700 点前後は必要であると言えます。
しかし、仕事を行っていく上では、同時にコミュニケーションスキルも求められていることに注意しなければいけません。

コミュニケーション能力によっては、高得点を取っていたとしてもダメなケースもあることでしょう。日本の学校で習うような「読み書き」だけではなく、「話す聞く」といった能力も必要になってくることは間違いないでしょう。

「3.4 外部委託状況」や「3.6 外部調達のミドルウェア」、「3.8 使用ツール」の結果でも、語学力が必要ではないかと問題提起してきましたが、語学力は、組込みソフトウェア開発の現場では、今後、必要不可欠となりつつあると言えそうです。

それだけに、組込みソフトウェア開発会社にとっては、この語学力の習得については、最低限抑えておくべきスキルになるのかも知れません。

4.3 外部委託

「3.4 外部委託状況」で示した通り、多くの組込みソフトウェア開発会社が外部へ委託をしています。外部委託は、不足している技術力や社内のリソースを補える点やコストの点でメリットも多いため、企業や組織が存続するためには避けては通れないものであると言えるでしょう。

外部委託に対しては、2 つの見方があり、ひとつは、外部委託する側（外部委託側）と、もうひとつは、外部委託される側（受託側）があります。

「外部委託」と聞くと、どうしても外部委託側の話が中心になりがちですが、受託側の問題も解決していかなければ、そこから作り出される組込みソフトウェア全体としての課題を解決することは難しいでしょう。

このため、本節では、この両面からの課題と解決策を考えてみたいと思います。

まず、外部委託側としての課題ですが、「3.4 外部委託状況」で取り上げていますので、ここでは、その課題となる項目を列挙してみます。

- 品質管理、工程管理などのマネジメント
- コミュニケーション
- ノウハウ流出

これらの課題に対する解決策については、「4.3.1 外部委託側」を参照して下さい。

では、もう一方の受注側としては、どのような課題があるのでしょうか？

以下の「図 4-8　受託の課題（発注元：国内大企業）」および「図 4-9　受託の課題（発注元：国内中小企業）」に、受託側の企業が挙げている課題を示します。

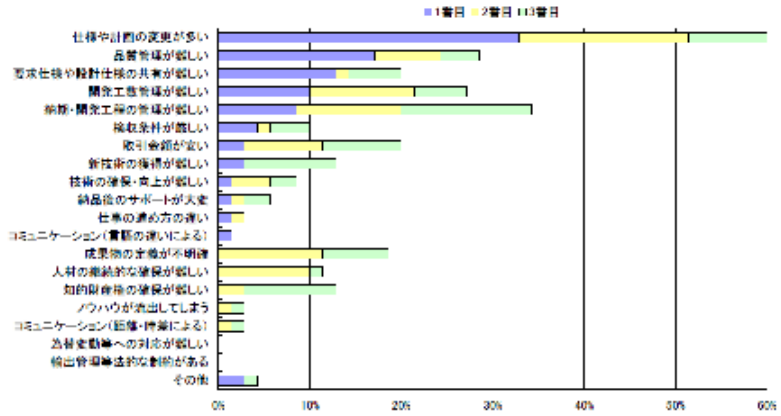


図 4-8 受託の課題(発注元:国内大企業)

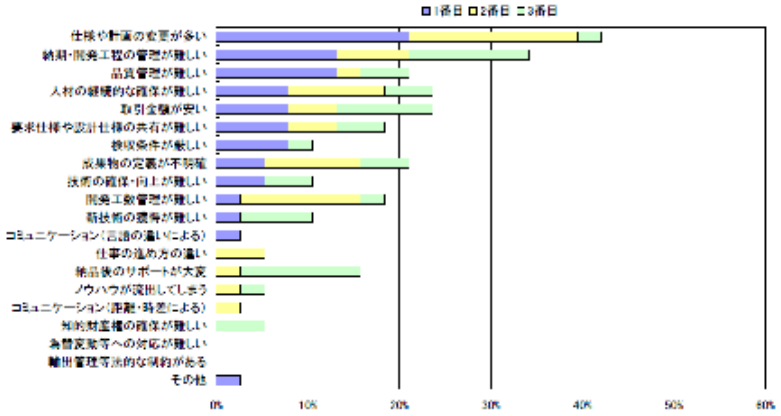


図 4-9 受託の課題(発注元:国内中小企業)

発注元が、国内大企業と国内中小企業によって多少の違いはあるものの、「仕様や計画の変更が多い」(国内大企業:32.9%、国内中小企業:21.1%)ことが、課題として多くの割合を占めている状況です。

また、「品質管理が難しい」(国内大企業:17.1%、国内中小企業:13.2%)や「納期・開発工程の管理が難しい」(国内大企業:8.6%、国内中小企業:13.2%)といった、外部委託側でも問題になっているマネジメントに関する課題も見えてきます。

組込みソフトウェアでは、開発の初期段階で、ソフトウェア要求が確定することは稀であり、開発途中で、機能追加などの仕様変更が発生することが多いのが現状です。これは、並行して開発されるハードウェアと協調させるための変更はもちろんですが、他のミドルウェアなどのソフトウェアとの競合によって開発の途中であっても、新しい仕様の取り込みなどにも対応することもあります。また、ハードウェアによる不具

合が見つかったとしても、それを組込みソフトウェア自体で、その不具合を吸収しなければいけないこともありえるのです。[12]

その他にも特徴的なところとしては、発注元が国内中小企業の場合には、「人材の継続的な確保が難しい」（7.9%）、「取引金額が安い」（7.9%）という課題が挙げられています。これらは、組込みソフトウェア開発だけの課題ではありませんが、外部委託の構造上で末端に位置する受託側では、単発的な作業の請負と、上位に位置する企業側の利益確保のために、単価が安く設定されてしまうことがあります。これらは、資本主義の仕組みの上では、仕方がない部分があるとしても、実は、組込みソフトウェア技術者を育成していく土壌としては課題があることは間違いがなく、組込みソフトウェア開発会社にとっては、優秀な人材が流出してしまう可能性が高くなる可能性があることを示しています。

以下の「図 4-10　受託の有無」は、有効回答数 193 事業部門（191 社）の経営者及び事業責任者向けの調査結果ですが、そのうちの 44%（およそ 85 事業部門）が、受託しており、「図 4-9　受託の課題（発注元：国内中小企業）」の「取引金額が安い」という課題に対し、1 番目の課題としている経営者や事業責任者は 7.9%、2 番目で 5.3%、3 番目で 10.5%と、全体で 27.3%（およそ 23 事業部門）の経営者や事業責任者が、課題と感じていることになります。

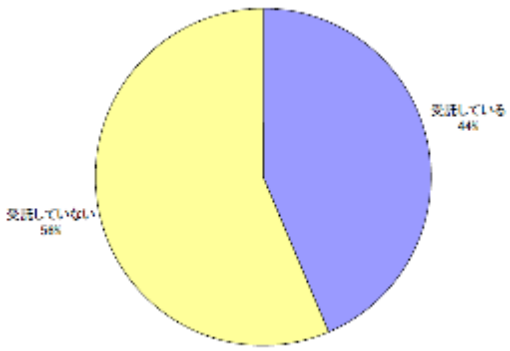


図 4-10　受託の有無

取引金額が安いと、必然的に、その企業の社員に対する給与水準も下げることになり、受託側では、どうしても給与面からの人離れが発生する可能性があります。もちろん、給与だけで人離れが発生する訳ではありませんが、客観的な評価として、給与の高い、低いというのが目安にされてしまうことも多いため、より高収入を求めるこ

[12] ハードウェアが確定してしまった段階で見つかった不具合に対して、費用面からもハードウェアを作り替えることが非常に難しくなってきてしまいます。もちろん、不具合の程度によりますが、ハードウェアの不具合を組込みソフトウェア側で吸収しなければいけなくなってしまうことがあるのです。

とも少なからず存在してしまうことになるでしょう。

4.3.1 外部委託側の課題

ここでは、外部委託側の課題に対する解決策を考えてみたいと思います。

4.3.1.1 マネジメントに関する課題

外部委託側としては、もっとも課題となるのが、品質管理や工程管理などのマネジメントの難しさが際立っていると考えてよいでしょう。

マネジメントに関する課題は、組込みソフトウェア開発に限った話ではなく、多くのソフトウェア開発企業でも、外部委託する際のマネジメントについて悩みを持たれている方が多いのが事実です。

さまざまな理由により外部委託したものの、途中段階での進捗状況や品質状況を確認してもまったく問題ないが、フタを開けて見ると出来あがっていない、品質が良くないなどの問題が、後から見つかるというものです。このようなことが発生する多くの場合、途中段階でのエビデンスの提供を求めていないことが多いように感じています。例えば、進捗管理の打ち合わせを実施しても、単に、「進捗どうですか？」と聞き、「進捗に問題ありません」という無駄なやり取りが存在していないでしょうか？

ただ単に、状況を問い合わせるのであれば、何も打ち合わせの場を設ける必要はありません。それよりも、「ソースコードを提出して下さい」、「ドキュメントを提出して下さい」という方が、委託先にとってはプレッシャーになると考えられます。もし、ソースコード管理ツールなどを利用しているのであれば、該当部分だけを取り出すことも容易に出来るのではないかと考えられます。

ただし、ここで間違ってはいけない点があります。
エビデンスを提出して貰う際に、多くの工数が発生してしまうようなやり方ではいけません。工数等を最小限に抑えるような施策は、絶対に必要なことですので、その点を間違わないようにしましょう。

このように、委託先での管理方法も把握しながら、エビデンスを取り出す方法を共に考え、共に運用していくことによって、マネジメントに関する課題を改善していくことは可能ではないかと考えられます。

4.3.1.2 選定基準

マネジメントの課題にも通じますが、外部委託する際に、選定基準を設ける企業もあることでしょう。

しかし、大企業の場合には、選定基準を設けて、外部委託先を決定することもありますが、中小企業の場合には、選定基準そのものが存在していないこともあります。もちろん、外部委託先を決定するのですから、何らかの理由によって決めている訳ですが、その多くが、担当者が決めてしまうことも少なくないでしょう。なお、この選定方法がいけない訳ではありません。

開発現場を知っている担当者は、間違いなく、求めていること（要求事項）が明確に分かっており、それに合致した外部委託先を選定することが出来るからです。

このように、選定基準がある場合、選定基準がない場合がありますが、そのどちらにも課題があります。

まず、選定基準がある場合を考えてみましょう。

ここでの課題は、本当に意味のある選定基準になっているかどうかです。

選定基準は、各企業が、それまでの経験などを元に定めていることが多く、与信管理のような会社そのものの経営状況や会社の規模、業績なども含めていることもあります。これらの情報は、委託先が突然倒産してしまうなど、資金を投入したにも関わらず、その対価（ソフトウェア開発を委託している場合には、プログラムなどの成果物を指します）を得られない状況が発生しないよう企業としてリスクをヘッジするために必要なことではあるので、不要だとは言いませんが、本来の外部委託をしたい理由から考えると少し異なる可能性はありそうです。

「3.4 外部委託状況」で説明したとおり、外部委託する理由は、国内企業に対しては「技術力」を求めています。「技術力」を求めた場合、技術力が高い企業すべてが、経営状況が良いとは限りません。本当に、技術力を求めているのであれば、その技術力が確実にあるのかどうかを確認する方が、効率が良いかも知れません。

経営状況などのリスクと技術力については、トレードオフの関係にはなっている訳ではありませんので、もし、どちらかを優先させなければいけないような状況になった場合に、どちらを優先するのかといった選定基準の設け方をしても良いと思われます。気をつけなければいけないのは、単に、定めている選定基準だけを見て判断をしてしまうと、求めている技術力を犠牲にしてしまうこともあるということを押さえておく必要があるということです。もちろん、それらを加味した形の選定基準を設ける方法もあるので、あらかじめ十分に検討しておくのが良いでしょう。

また、国外企業に対しては、「コスト削減」を求めていますが、単に、コスト削減だけを目指していては良いものは出来あがらないことを十分に考慮する必要があるでしょう。国外企業に委託することで、本当にコスト削減が出来るのかを総合的に判断すべきです。安易に、人件費が安いからと、国外企業に委託し、その管理コストが大幅

に増えてしまい、結局、国内企業とコスト面ではあまり変わらないという事例はたくさん存在しています。

次に、選定基準が無い場合です。

選定基準が無い場合に、気をつけなければいけないことは、担当者などの主観によって選定されてしまうことがある点と、一旦、選定されると、その後に、見直されることが少なくなってしまう点が挙げられます。

主観によって選定されてしまうと、本来必要である技術力が足りなくても選定してしまうことがあり得ます。この主観部分を、排除することが出来るかどうかは、とても重要なものとなります。

主観によって決定する場合には、委託先を決定する人によって、間違いなく選定基準にバラつきが生じることは覚悟しなければいけません。もちろん、それぞれの部署で、選定基準の異なる委託先になるというのは、十分に考えられる話ですので、それを否定するものではありませんが、あまりにもかけ離れた選定基準である場合には、マネジメントを行う上でも、非常にリスクが高くなってくる可能性はあります。

また、長期間、委託先を利用するケースの場合ですが、このような場合に、一旦、委託先を決定した後で、どうしてもズルズルと契約を引き延ばしてしまうケースもあります。主観で決めている場合には、その主観を覆すようなことの無い限りは、基本的に、委託先を継続してしまうことが多いのではないでしょうか？
もちろん、それが、絶対にダメだとは思いませんが、本当に、選定基準を満たすような力量を、持ち続けられているのかなどの評価は、必要ではないでしょうか？

選定基準を満たせるよう努力している委託先もたくさんありますが、長期間委託されている場合に、その環境に甘え、努力を怠る委託先も少なからず存在しています。そのような委託先に対して、プレッシャーを与えるためにも、再評価を行えるような仕組み、更には、その再評価を行うための基準は、必要ではないかと思います。

4.3.1.3 コミュニケーションの課題

コミュニケーションの課題として挙げられているのは、言語の違いによるものと、距離及び時間の違いによるものが課題として挙げられています。
しかし、この課題は、実態調査での質問事項として用意されているものですので、その 2 つだけがコミュニケーションに対する課題だとは考え難いと思っています。

社内あるいは社外のステークホルダー[13]とのコミュニケーションを円滑にし、活性化させていくことが重要だということに対して、異論を唱える方は少ないでしょう。また、何か問題が発生したときに、その原因のひとつとして、コミュニケーション不足やミスコミュニケーションを挙げる人も多いのではないでしょうか？

コミュニケーションについての課題を考える前に、まずは、コミュニケーションの目的について考えてみましょう。コミュニケーションのもっとも大きな目的としては、情報共有が考えられます。ステークホルダー間で、必要な情報が共有されることにより、円滑に作業が進められるというものです。

この目的を達成するために、言語の違いや距離や時間の違いが問題になっているというのは、当然のことだと思います。言語によってコミュニケーションが取れない（取り難い）というのは、最小限に抑えるようにしていかなければいけません。この課題については、「4.2.3 語学力」で触れていますので、そちらを参照して下さい。
距離については、離れた企業へと委託した場合に、直接的にコミュニケーションを取ることが出来ないことが問題になることがあります。特に、海外企業へ委託する際には、時差の関係から、コミュニケーションの問題が顕著に表れてくることがあります。

これらの解決のために、ツールの活用は、もはや必要不可欠であると言えるでしょう。コミュニケーションを活性化するためのツールとしては、代表的なものでは、メール、グループウェア、イントラネット、携帯電話などが挙げられます。また、最近では、遠隔会議システム（テレフォンカンファレンス用だけではなく、TV 電話会議システムなども含まれます）といったツールの導入も増加傾向にあります。
しかしながら、グループウェアやイントラネット、遠隔会議システムについては、自社内におけるツールの導入は出来ていたとしても、委託先との間で、これらのツールを有効に活用することが出来るかと言えば、セキュリティの問題やコストの問題からも、共有して使用出来る環境にすることは難しいことがあります。

このため、メールや携帯電話といった比較的安価な仕組みの範囲内でのステークホルダー間でのコミュニケーション方法を検討することが必要だと考えられます。
しかし、一旦、グループウェアやイントラネット、遠隔会議システムといったツールに慣れ親しんでいる場合には、安価な仕組みにすることが難しいと感じることがあります。会社の規模が大きくなればなるほど、ツール導入が進んでいる傾向があるため、委託先との間での異なる環境に順応出来ずに、ミスコミュニケーションへ発展してしまうことがあるかも知れません。

そうならないためにも、自社内のツールだけに頼るのではなく、あらかじめ比較的安価な仕組みを検討しておき、コミュニケーションを図るようにしても良いかも知れません。

[13] 企業等の利害と行動に直接あるいは間接的な利害関係を有する者を指します。

4.3.2 受託側の課題

「4.3 外部委託」では、データを元に受託側の課題を示しましたが、以下のような課題がありました。

- 仕様変更等が多い
- 品質管理、工程管理などのマネジメント
- 人材（単価が安い、継続的確保）に関する課題

ここでは、これらの課題について、解決策を探っていきたいと思います。

4.3.2.1 仕様変更等が多い

組込みソフトウェア開発特有の問題として、ハードウェアとの関連性が強いため、開発初期段階でソフトウェアの仕様を早い段階で確定しきれないことがあります。

例えば、ハードウェア開発と並行して組込みソフトウェアを開発する場合には、ハードウェアへの合わせ込みによる仕様変更や仕様追加が頻発する場合や、すでにハードウェア開発が終了していたとしても、ハードウェア側の問題を、ソフトウェア側で吸収させるために仕様変更を余儀なくされる場合[14]などもあります。

このため、上流工程における品質確保が難しくなり、下流工程にまで品質面での影響を与えることも少なくありません。このように、組込みソフトウェア開発の現場では、初期段階での仕様確定が出来ないために、仕様変更や仕様追加が多発することがあります。

さまざまなお客様から良く聞く対応方法として、大きく 2 つに分類されます。

ひとつは、仕様変更が無くなるように上流工程において、仕様を確定させる仕組みを作りだすようにする方法です。もうひとつは、『仕様変更があるのは当たり前』という想定のもとで、プログラムの書き方だけではなく、各ソースコードファイル、ドキュメントまでを含めた構成管理に至るまで、仕様変更し易い構造にしたり、アジャイル開発のような開発手法を取り入れ、早期に問題を解決する仕組みを導入したりする方法です。

根本的な解決としては、前者が有効であるのは間違いありません。

[14] ハードウェアを作り直すとコストが増大することがあり、その場合には、ソフトウェア側で、ハードウェアの問題を吸収する場合もあります。もちろん、すべてそのような対応を行う訳ではありません。

というのも、下流工程において問題が発生した場合に、その原因の多くが、設計書や仕様書類での不備などが挙げられることが多いからです。実際に、当社で蓄積している組込みソフトウェア開発のプロジェクトに関するデータでは、全体不具合の 30%~40%が、設計工程などの上流工程に起因しているという結果も出ています。

では、なぜこのようなことが発生してしまうのでしょうか？

ひとつには、設計手法についての教育体制が脆弱なことが挙げられます。
組込みソフトウェアだけではなく、多くのソフトウェアエンジニアを育成する場合、開発言語やアルゴリズムなどのプログラミングに関する技術的な教育は、多くの企業でも充実していますが、設計に関する教育は手薄になっていることが多いのではないかと思います。また、設計に関する教育を行っている場合であっても、設計手法などのテクニックに関する教育がほとんどです。
しかし、上流工程では、その多くが、コミュニケーションによって成り立つ部分が多いことから、体系的に教育することが出来ないのではないかと考えられます。

また、「設計手法」、「コミュニケーション」に対するそれぞれの教育は見受けられますが、それらをまとめた教育というのは、各段に少ないのが現状です。これらを解消するような教育というのも、今後必要になってくるかも知れません。

もうひとつには、日本語自体が、あいまいな言葉が多いため、設計書や仕様書への記述自体も、あいまいな記述になってしまうことも一因となっていると考えられます。問題発生時の原因として、「仕様不明瞭（曖昧）」、「仕様記述ミス」、「仕様記述漏れ」というのは、常にトップクラスに存在していないでしょうか？これらの問題は、先のコミュニケーションの課題にも絡んでくる場合もありますが、それぞれ施策を考えるようにした方が良いでしょう。

4.3.2.2 品質管理、工程管理などのマネジメントの課題

外部委託側の「4.3.1.1 マネジメントに関する課題」にもマネジメントの問題がありますが、受託側でも同様にマネジメントの問題があります。

特に、受託側では、委託先での管理と同じ方法を求められることも多いため、その違いを、受託側で吸収しなければいけないこともあり得ます。

では、本当にすべてを受託側のみで対応しなければいけないのか？という課題にもぶつかることでしょう。この点については、あらかじめ、それぞれのマネジメント方法について、開発そのものにも影響を与えるような管理はしないように交渉すべきでしょう。

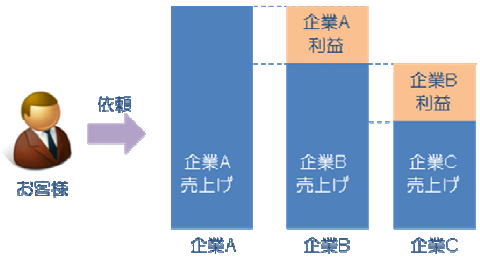
4.3.2.3 人材（単価が安い、継続的確保）に関する課題

人材に関する課題も多く存在しています。ここでは、まとめて「人材」というくくりにしていますが、「単価が安い」という課題と継続的確保では大きく意味合いが異なっています。

まず、単価が安いという点については、発注元が国内中小企業の場合に、上位に位置づけられている課題です。この課題は、組込みソフトウェア業界全体における構造にも問題があることがあり得ます。

例えば、国内中小企業が開発している組込みソフトウェアが、実は、大企業から委託を受けているような場合です。実際には、委託ではないとしても、お客様が、大企業であることは少なくありません。つまり、末端の委託先に行くまでの間に、何層もの構造が出来あがり、その層ごとで利益が取られることによって、必然的に末端にいる受託側が安い単価の中で、作業を行わなければいけないことも少なくないのではないかと思われます。

このような構造は決して健全なものではなく、末端の受託側企業からの人材流出や、雇用不足の問題が広がり、最終的に、組込みソフトウェア業界全体の不健全な体質にもなりかねないのではないかと考えられます。



組込みソフトウェアであれ、製品であれ、本来のお客様に対するより良い製品を提供することを目的としているはずであり、その目的を満たすための活動を、委託する側には考えて欲しいと切に願っています。

人材の継続的確保という点については、組込みソフトウェアの開発では、特定の分野に長けた人材が必要となることも少なくありません。このような人材は、その企業内でも稀有な存在となりがちであり、そのために、いくつものプロジェクトを任されていたり、引く手あまたの状態になったりすることもあるでしょう。そのために、必要なときに必ずしもその人材がいるとは限らない状態になることも考えられるため、このような人材になり得るように、社内の人材育成も必要になってくるでしょう。
しかし、人材流出によって、社内での人材の継続的確保が出来ないのだとするならば、少し深刻な問題と言えそうです。

4.4 外部調達

組込みソフトウェアの開発は、巨大化していることは、これまでも何度も言ってきていますが、巨大化していく中で、効率やコストの面からも、外部からソフトウェア

部品を調達するような開発になっていくことは明らかでしょう。近年では、スマートフォンの需要が拡大してきていますが、その中でも、スマートフォンを中心に搭載が進んでいる Android の存在は、更に、ソフトウェアの部品化を推し進めていくかも知れません。

かなり以前からソフトウェアの部品化と再利用については、議論されている点も多く、語り尽くされた感もありますが、今後の部品化は、自社開発ソフトウェアの部品化への流れではなく、あくまでも外部から調達する部品化されたソフトウェアを、いかに使いこなし、また、それらのソフトウェア部品と自社開発ソフトウェアを含めた総合的な品質を確保していくのかという点に推移していくと考えられますし、「3.6 外部調達のミドルウェア」に記載されている状況からも、すでに、そのような状況になっていると考えても良いでしょう。

しかし、部品化されるということは、その部分をブラックボックス化してしまうことにもなるため、組込みソフトウェアの総合的な品質確保は、かなり難しいものとも言えます。それゆえに、品質管理部門や品質保証部門の方々だけではなく、組込みソフトウェア開発の部門の方々も、それらの課題を考慮しながら、お客様へより良い組込みソフトウェア製品を提供するという目的を達成するために、何をすべきか、どのようにすべきかを考えていくことが試されていると言えるのではないでしょうか？

また、ちょっと矛盾してしまいますが、単にすべてを外部調達に頼るというのも考えモノです。というのも、外部調達を行っていては、独自性のある組込みソフトウェアを開発することにも制限を与えることになり得るからです。独自性を失うということは、すなわちユーザーにとっての魅力的な製品には、なり難いという側面を持っており、ユーザーからのニーズを失う可能性もあるのです。そのためにも、単にすべてを外部調達に頼ることなく、自社の技術力を向上させ、魅力ある組込みソフトウェアを作ることも検討すべきではないかと考えます。

5. おわりに

冒頭でも触れたように、近年、組込みソフトウェアは複雑化・多機能化によって規模が拡大している一方で、開発期間は短縮される傾向にあります。また、携帯電話や家電製品などの組込み製品の多くが、韓国や台湾、中国といったアジア諸国を中心として、どんどんと力をつけてきている状況です。今後も、組込みソフトウェア開発の競争は、もっと激化していくことでしょう。

日本は、資源が乏しい国のひとつであり、諸外国から原材料を調達し、加工し、付加価値を創造した製品として全世界に向けて発信してきました。そこで作られた製品に対し、更に、新しい息吹を与えることが出来たのが、組込みソフトウェアであり、これらは、世界に誇るべき創意工夫をしてきた伝統であると考えています。また、それが日本の強みだと考えています。「品質が良い」、「性能が良い」、「壊れない」といった付加価値は、今も健在であり、諸外国でも、日本の製品は良いものだという認識は、未だに変わってはいません。

本書では、産業実態調査の結果から、さまざまな課題があることを示してきましたが、これらの課題についても、日本の組込みソフトウェアに関わる皆さんが、創意工夫によって解決していくことが出来ると思っていますし、その一部に、私たちも協力することが出来ればと考えています。

2011 年 07 月
クオリティ・エージェント

付録1.　参考資料

本書では、以下の書籍類を参考にさせていただきました。

- ソフトウェアエンジニアリング基礎知識体系ーSWEBOK
  (訳)松本 吉弘 / オーム社
- ISO/IEC9000:2005 Quality management system - Fundamentals and vocabulary
- ISO/IEC9001:2008 Quality management systems – Requirements
- ISO/IEC12207:1995 Software Life Cycle Processes
- ISO/IEC15288:2002 System Life Cycle Processes
- JIS Q 0014:1999 情報処理用語ー信頼性、保守性及び可用性
- JIS Q 9000:2006 品質マネジメントシステム基本及び用語
- JIS Q 9001:2008 品質マネジメントシステムー要求事項
- ISO9001:2008(JIS Q 9001:2008) 要求事項の解説
  飯塚悦功（著)、住本守（著)、平林良人（著)、福丸典芳（著)、棟近雅彦（著）/ 日本規格協会
- 平成 21 年特定サービス産業実態調査結果
  経済産業省
- 組込みソフトウェア向け開発プロセスガイド
  独立行政法人　情報処理推進機構　ソフトウェア・エンジニアリング・センター編 / 翔泳社
- 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[計画書編]
  独立行政法人　情報処理推進機構　ソフトウェアエンジニアリングセンター編 / 翔泳社
- 組込みソフトウェア開発における品質向上の勧め
  独立行政法人　情報処理推進機構　ソフトウェアエンジニアリングセンター編 / 翔泳社
- 組込みスキル標準 ETSS 概説書 [2006 年度版]
  独立行政法人　情報処理推進機構　ソフトウェアエンジニアリングセンター編 / 翔泳社
- 組込みシステムの安全性向上の勧めー機能安全編
  独立行政法人　情報処理推進機構　ソフトウェアエンジニアリングセンター編 / 翔泳社
- 組込みスキル標準 ETSS 概説書[2008 年度版]
  独立行政法人　情報処理推進機構　ソフトウェアエンジニアリングセンター編 / 翔泳社
- 2004 年版～2010 年版組込みソフトウェア産業実態調査 報告書
  経済産業省, 独立行政法人　情報処理推進機構　ソフトウェアエンジニアリングセンター