

SEC journal

51

巻頭言

澁谷 裕以 特定非営利活動法人ITコーディネータ協会 会長

所長対談

**企業のデジタル化を推進するために
～CDOの役割とは～**

神岡 太郎 一橋大学 商学研究科 教授 / CDO Club Japan 顧問

論文

**ISBSGデータを用いた見積もり研究に対する
IPA/SECデータを用いた外的妥当性の評価**

山田 悠斗 大阪大学 大学院情報科学研究科

江川 翔太 大阪大学 大学院情報科学研究科 / ソニーグローバルマニュファクチャリング&オペレーションズ株式会社

松本 真佑 大阪大学 大学院情報科学研究科 / 角田 雅照 近畿大学 理工学部情報学科

楠本 真二 大阪大学 大学院情報科学研究科

提案依頼書に含まれる無理難題の分類

門田 暁人 岡山大学大学院自然科学研究科 / 住吉 倫明 岡山大学工学部情報系学科

神谷 芳樹 みたに先端研

特集

ソフトウェア開発の定量的管理

「ソフトウェア開発の定量的管理」特集にあたって

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(エンタプライズ系)

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

NECにおける定量的管理

～ソフトウェア品質会計とNECデータ白書に基づく改善活動の展開～

SPI活動と連携した定量的プロジェクトマネジメントの実践

ソフトウェア定量的管理にかかわる学術研究事例

報告

Embedded Technology West 2017 (ETWest2017) 出展報告

INCOSE国際シンポジウム2017開催模様

ソフトウェア高信頼化センターの地域支援活動について

IT利活用を支えるソフトウェア技術の地方への展開状況報告

Column

ハドソン川の奇跡

1

巻頭言

ITを経営の力とするということ、あるいはITコーディネータの役割について

澁谷 裕以 特定非営利活動法人ITコーディネータ協会 会長

2

所長対談

企業のデジタル化を推進するために ～ CDOの役割とは～

神岡 太郎 一橋大学 商学研究科 教授 / CDO Club Japan 顧問

10

論文

ISBSGデータを用いた見積もり研究に対する
IPA/SECデータを用いた外的妥当性の評価

山田 悠斗 大阪大学 大学院情報科学研究科

江川 翔太 大阪大学 大学院情報科学研究科 / ソニーグローバルマニュファクチャリング&オペレーションズ株式会社

裕本 真佑 大阪大学 大学院情報科学研究科 / 角田 雅照 近畿大学 理工学部情報学科

楠本 真二 大阪大学 大学院情報科学研究科

提案依頼書に含まれる無理難題の分類

門田 暁人 岡山大学大学院自然科学研究科 / 住吉 倫明 岡山大学工学部情報系学科 / 神谷 芳樹 みに先端研

26

特集：ソフトウェア開発の定量的管理

「ソフトウェア開発の定量的管理」特集にあたって

野中 誠 東洋大学 経営学部 経営学科 教授 / 山下 博之 SECシステムグループリーダー

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 (エンタプライズ系)

峯尾 正美 SEC 研究員 / 佐伯 正夫 SEC 専門委員 / 山下 博之 SEC システムグループリーダー

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

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

NECにおける定量的管理

～ソフトウェア品質会計とNECデータ白書に基づく改善活動の展開～

宮崎 義昭 日本電気株式会社 / 菅田 直美 日本電気株式会社

SPI活動と連携した定量的プロジェクトマネジメントの実践

佐藤 浩明 TOiNX (トイックス) 東北インフォメーション・システムズ株式会社 開発運用本部 業務管理部 業務統括課 主査

ソフトウェア定量的管理にかかわる学術研究事例

野中 誠 東洋大学 経営学部 教授 / 亀井 靖高 九州大学 システム情報科学研究院 准教授

大平 雅雄 和歌山大学 システム工学部 准教授

50

報告

Embedded Technology West 2017 (ETWest2017) 出展報告

荒川 明夫 SEC プロモーショングループ 主任

INCOSE国際シンポジウム2017開催模様

SEC journal 編集部

ソフトウェア高信頼化センターの地域支援活動について

IT利活用を支えるソフトウェア技術の地方への展開状況報告

久保 忠伴 SEC 調査役 / 藤原 由起子 SEC 研究員 / 小長谷 義浩 SEC 研究員

56

Column

ハドソン川の奇跡

松田 晃一 IPA 顧問

57

書籍紹介

58

編集後記

SEC journal 論文募集/国家試験 エンベデッドシステムスペシャリスト試験のご案内

ITを経営の力とすること、 あるいはITコーディネータの役割について



しぶや ひろゆき
澁谷 裕以

特定非営利活動法人ITコーディネータ協会 会長

少し前のことになるが、日経コンピュータ誌が「システム開発の成功率」について2度調査したことがある。その結果は、以下の通り、実に7割ものシステム開発が失敗という結果であった。

- 2003年の調査・・・73.3%の失敗率
- 2008年の調査・・・68.9%の失敗率

私自身、40年間のシステム屋ライフの中で、そんなに失敗していたわけではないが、あのときちょっと気を許せば失敗だったな、という思いは山のようにあるので、この数字はそんなに驚きではない。むしろ、それだけ難しい仕事だという認識が世の中全般に薄い、あるいはシステム側の発信が足りないことが問題ではないかと思っている。

失敗の現れ方は、サービスイン時期の延期であったり、大幅な予算超過であったり様々であろうが、原因は大半が、できあがったシステムに対して「こんなもの作ってくれと頼んだ覚えはない」という、システム用語で言う「要件ギャップ」の問題であることが多い。

そうなる原因は、発注者の「思い」を深く掘り下げて具体的な形にしないままシステム開発を始めることにある。

人間の「思い」というのは、最初から具体的ではない。例えば、家を建てる施主が「明るい台所が欲しい」と言ったとする。私は、システム開発に対する最初の要望はほとんどこれくらいのレベルではないかと思う。気持ちは分かる。「明るい台所」は良い。しかし、建築家はこのままで図面を描くことはできない。

そこで、建築家は施主に対して次のような質問を繰り返して、施主の要望を具体化していく。

「それって、窓が大きくて陽がさんと降り注ぐような台所をイメージされていますか？」

「あるいは、台所のタイルがキラキラしているような感じですか？」

「それとも、カウンターキッチンで、家族と対話しながら料理ができるような雰囲気ですか？」

ITを経営の力としていくためには、このような対話の積み重ねが必須なのだ。対話の中で、経営者の思いが深く掘り下げられ、システムとして具体化し現場で威力を発揮する。そのようにしてITは初めて経営の力となっていく。

ITコーディネータの役割は、中小企業経営者の良き対話相手になることに尽きる。

経営者は最初からITを「自分ごと化」しているわけではない。会社の売り上げを上げる、利益を増やすということに対しては、正に経営の根幹として「自分ごと」そのものだと思うが、そのためにITがどのような領域で、どのように役に立つのか、イメージできる経営者は少ない。

それは、経営者がITコーディネータとの対話を通じて少しずつIT化の対象領域を「見える化」し、「自分ごと化」していくべきものだと思う。

今ほど中小企業経営の中でITを経営の力としていくことが強く求められていることはない。中小企業において人手不足は共通する深刻な課題であり、この最も効果的な解決策はITの活用による生産性の向上である。しかしながら、中小企業の経営者にとって、ITはむしろ遠ざけたいもの、あるいは最近のデジタルネイティブの若者に任せておけば何とかしてくれるものになっていることが大きな問題だと思う。

IT導入に苦手意識をお持ちの中小企業の経営者の皆様に、こう申し上げたい。

中小企業の経営者は、どうぞITなど気にしないでご自分の夢を語ってください。ご自分の会社をどのような会社にしたいか、何が自社のアピールポイントなのか、語ってください。私たちITコーディネータはその思いをしっかり受け止め、経営者と対話をしながらITを経営の力として活用する「かたち」を作り、一緒に成果を上げていきますから、と。

私たちITコーディネータは経営者との対話を重ねながら、IPA/SECの知の蓄積も活かし、中小企業がITを経営の力とできるような専門家として役立つ支援をまいります。

企業のデジタル化を推進するために ～ CDOの役割とは～

一橋大学 商学研究科 教授
CDO Club Japan 顧問
神岡 太郎

IPA/SEC所長
松本 隆明

IoT、AI、ビッグデータなどの急激な技術革新に対応すべく、今や企業は事業活動を進める上で大きな変換点を迎えようとしている。そのためのキーワードはデジタル化である。企業においてデジタル化を推進する司令塔とも言うべきCDO (Chief Digital Officer) を設置する企業も徐々にではあるが増えつつある。日本における企業のデジタル化の必要性・重要性を早くから提唱され、CDO Club Japanの立ち上げにも尽力された神岡先生にデジタル化のもたらすもの、デジタル化を進める上でのリスクなどについてお話を伺った。

デジタル化が意味するものは何か

松本 「デジタルトランスフォーメーション」「デジタルディストラクション」といったキーワードをよく耳にするようになり、最近デジタル化が大変注目されつつあります。デジタル化がもたらすものと今後の方向性、CDOの役割や育て方といったことについてお話ができればと思います。はじめに、そもそもデジタル化とはどういうことを意味している言葉なのか、神岡先生はどうお考えですか？



神岡 太郎 (かみおか たろう)

1990年北海道大学大学院博士課程単位取得退学(行動科学専攻)。同年一橋大学商学部講師、その後助教授を経て、2005年より一橋大学商学研究科教授。マーケティング、デジタル、データ、情報システムが企業全体としてどう機能させるか、企業の競争力にどのように結びつくのか、どのように構造転換すべきかを研究対象としている。とくに、CDO (Chief Digital/Data Officer)、CMO (Chief Marketing Officer) の役割について関心を持っている。研究論文以外に、共著として『マーケティング立国ニッポンへ』(日経BP社、2013年)、『CIO学』(東大出版会、2007年)など多数。高度ICT活用人材育成推進会議座長(総務省)、トレーサビリティ・サービス推進協議会座長(国土交通省)など歴任。博士(工学)。

神岡 デジタルという言葉は非常にあいまいで、色々なステークホルダが、色々な意味で語っています。その人がデジタル化という言葉にどういう意味を込めているかということは、常に注意しておく必要があるでしょう。

デジタルという言葉を意

図的にITと区別して使っている人も多いですね。デジタルには何かを変えるという意味が裏に込められています。それによってユーザの活動が変わる、価値の作り方が変わる、経験が変わる、といったような意味を持つ技術のことをデジタルと呼んでいます。他方ITは、効率化したい、組織全体を最適化したい、といったところにフォーカスがあるように思います。デジタルは人間の活動を変えるというところに意識があり、ビジネスオリエンテッドであり、フォアグラウンドです。ITは、どちらかと言うとバックグラウンドで、その面では効率化以外にもセキュリティや安心・安全ということがあります。

もちろん両者は密接に関係していて、その関係をどうするかということは今後大きな問題になるでしょう。我々はエンゲージメントという言葉を使っていますが、人と人、機械と人の関係の仕方、関係性と言ってもお互いにアクティブにするような関係、例えばソーシャルメディアの「いいね」のように、単に良いと思っただけでなく「いいね」ボタンを押すというアクションを含めた関係ですが、このような経験をどう設計するかはデジタルの問題になります。しかし、そのプラットフォームを支えるのはITの問題になり、その境界があってないようなものになります。ただテクノロジーとして見れば、ベースは一緒なんです。その設計と使い方が違うということです。デジタルは、アジャイルな作り方と言うか、最初から計画してできないような設計の仕方をしなければなりません。設計が決まってそれに従ってシステムを作っていくのではなく、作ろうとしているビジネスのやり方自体があいまいなので、作るシステムは最初から決まって

いないわけです。ご存じのように、昔のIT戦略というのはビジネス戦略が確定した上で作っていったわけですが、デジタル戦略の場合は、IT戦略とデジタル戦略は同時、双方向的であるという考え方です。デジタルを設計することはビジネスをどうするかということと同時にスタートするものだということですね。

手持ちの資産ではなく顧客から発想する

松本 これまでITは業務効率化など目的がはっきりしていました。目的があってIT導入が進んでいました。しかし最近はそのではありません。ITの世界にSystems of Record (SoR) と Systems of Engagement (SoE) という言葉がありますが、SoRからSoEへ変えていこうという流れがあり、どういったエンゲージメントの仕組みをITで作りに出していくかということを考えながら設計していく。そういうことだと考えれば良いのでしょうか。

神岡 SoEとSoRはハイモダリティITとしてガートナーらが強調していますね。対応していると思います。CIO (Chief Information Officer) はSoR、CDOはSoEだと言う人もいます。設計の仕方として、どこをスタートにするのか、SoEは活動している人をスタートポイントにして設計していくわけです。今大きな変革を迫られているのは、これまで成長してきた価値を生んできたトラディショナルな企業です。それに対して、デジタルディスラプターと呼ばれているのがUber (ウーバー) やAirbnb (エアビーアンドビー) などの新興企業です。ほとんど資産を持たないことを前提としたシェアードエコノミーです。古いビジネスに置き換えれば、Uberは世界で最も大きなタクシー会社かもしれないし、Airbnbは世界で最も大きなホテル会社かもしれない。しかし、それらはうまくデジタルを使っているだけではありません。もしそうなら、既にある企業でなぜできなかったのでしょうか。トラディショナルな企業は、基本的には資産を持ってしまっているのです。そこから出発せざるを得ない。例えば従業員がいます。この人たちは明日からいらないということとはできない。ところがUberやAirbnbは全く資産を持っていないところからスタートするので、まず顧客を起点にする。顧客がどういう経験をすれば一番便利だろう、そこにデジタルをどう使えば良いのだろうということをスタートポイントにします。こういう作り方が最近のデジタルの基本ではないでしょうか。サービスを提供する側の視点で設計するとうまくいかない。デジタルを本当にうまく使おうとしているのは、それを使う人から発想する能力を持っている人だと思います。

デジタルが拓いた コ・クリエーションの時代

松本 デジタルトランスフォーメーションということが急速に

注目されるようになってきた理由は何だとお考えになりますか？

神岡 いや私は5,6年前からずっと言っているんです(笑)。世界ではデジタルトランスフォーメーションという言葉は非常によく使われていました。ところが、日本でだけはなぜか使われていなかった。最近は色々な事例が出てきたということ、そして、デジタルが色々な意味で民主化してきたということでしょう。デジタルが最初に破壊したのはメディアと広告です。

松本 確かにGoogleのようなものが生まれて、それまでの広告とは違うやり方になりました。

神岡 CDOが生まれたのは2003年とされています。MTVネットワークスというメディア会社で生まれ、その後もCDOはメディア業界で生まれていきます。従来、メディア業界はマスメディア、いわゆる4媒体を中心にやってきましたが、デジタルに変わらなければならない。日本はマスメディアが非常に強いのですが、海外では、例えばイギリスで言うと、既にネット広告がテレビ広告を抜いています。海外ではそれに対応しないといけない、という状況で動いています。今の学生は言葉すら知らないのですが、DTP (Desk Top Publishing) の登場は大変衝撃的でした。それまでチラシは専門の印刷業者に頼んで初めてできたんです。ところがこれが自分のコンピューターの画面でできて、それを編集して、そのまま印刷すればプロダクトになってパブリッシングができるんです。同じようなことがインターネットの世界で起きて、それまではマスメディアを通さないと組織は顧客にメッセージを伝えることはできませんでしたが、今は自分が放送局であるがごとくメッセージを伝えることができます。同じようなことが、3Dプリンターを使って、製造業と流通の間の垣根を取り除くかもしれません。あるいは、ソーシャルメディアを使えば、組織や企業ではなくて個人が発信できる。YouTubeを使えば個人がコンテンツを創ってそれをライブで配信することができるわけです。完全に民主化されている。そういう圧力が企業に対して大きくかかっている。顧客、一般人の活動は、デジタルで個人が使える環



松本 隆明 (まつもと たかあき)

1978年東京工業大学大学院修士課程修了。同年日本電信電話公社(現NTT)に入社、オペレーティング・システムの研究開発、大規模公共システムへの導入SE、キャリア共通調達仕様の開発・標準化、情報セキュリティ技術の研究開発に従事。2002年に株式会社NTTデータに移り、2003年より技術開発本部本部長。2007年NTTデータ先端技術株式会社常務取締役。2012年7月より独立行政法人情報処理推進機構(IPA)技術本部ソフトウェア高信頼化センター(SEC)所長。博士(工学)。

境になっているのに、トラディショナルな企業が出しているサービスはそれに追いついていません。行政サービスも同じです。一般の市民は携帯電話やスマートフォンを持って日常的に活動しているのに、いまだに役所の窓口に行って、紙に書いて順番を待って、許認可を申し込んで何日も待って返事がくる。

松本 webやインターネットの大きな力は、双方向ということですね。今までサービスの提供は提供業者から顧客への一方の流れだったのが、双方向にインタラクションが取れるようになってきました。

神岡 そうだと思います。その双方向の力をどう使うか。企業間でもそうですし、企業と顧客の間でもそうです。Co・クリエーション(Co-Creation)という言い方をしますが、企業がイノベーションを起こすときには、顧客起点で起こさないといけない。この技術が良いからではなくて、この技術が顧客にどういう意味があるか、と発想して、顧客起点でイノベーションを起こす必要があります。マーケティングの世界ではそういうことが起こっていて、少し古い例ですが、あるハンバーガーチェーンが新しいバーガーを作ったときのことで。新商品のネーミングは、マーケティング的には非常に重要なのですが、消費者と一緒に考えただけです。募集して、意見交換し、投票をする。そういうことが簡単にできてしまうわけです。デジタルという環境があって、みんながデバイスを持っていて、それが双方向につながる。そしてそれはスパイラルのようになっているのだと思います。お互いが色々なところで影響し合う。だから変化が激しい。今はやっていて、次何がはやるかは分からない。昔は広告代理店のようなトレンドセッターがいて、はやりを作っていました。ところが今はすべての人が誰でもトレンドセッターになれる可能性があります。予測不可能、あるいは予測がすごく難しい環境になっています。企業が自分たちだけでものを考えて判断するのは非常に危険になっています。顧客とエンゲージしながら、あるいはコラボレーションするパートナーとネットワーク的に関係し合いながら活動していく中で、デジタルは呼吸するがごとく我々の生活に入ってきているわけです。

松本 昔は企業がPRや広告の戦略を立てて、こういう広告を打って顧客をこういう方向に誘導していこうということができましたが、今は口コミが先行してマーケットを作っていく。企業にとっては将来が読みにくくなっていますね。

神岡 極めて読みにくいですね。変化の激しさ、伝達の激しさ、予測のしにくさはデジタルでなければ解決できないのですが、そのデジタルが更にスピードをアップさせて不確実な世界を作り上げています。企業はこのデジタルと共存して生きていかなければいけないわけです。

なぜCIOではなくCDOが必要なのか

松本 その意味でも企業におけるCDOの意義が増していると思います。最近ではCDOを置く企業が増え、日本でも2015年にはゼロだったのが、2016年には大手企業の7%で採用されていると聞きました。

神岡 その数字はCDOという肩書を持っている人だけではなくてデジタルのリーダーがいるという意味だと思います。CDOという肩書を持っている人はまだそれほど多くありません。ただ、CIOやITの担当者とは別にデジタルの担当者を置いているところはそのくらいあるでしょう。

松本 CDOの役割というのは何なのか、CIOやCTO(Chief Technology Officer)とは違う、あるいはCMO(Chief Marketing Officer)とも違う役割を担っていると思います。どのように考えたら良いのでしょうか？

神岡 CIOでもCMOでもCTOでも、CDOのような役割を果たすことは可能です。ただ、現状を言いますと、CTOはやはりテクノロジーを作る責任者です。CMOもデジタルに深くかかわっていますが、デジタル・コミュニケーションの部分がメインになります。一番議論の対象となるのはCIOとの関係でしょうか。CDOの役割は、デジタルをキーにして新しい価値を作っていくことです。新しい市場を作っていく、新しいサービスを作っていく、あるいは新しい製品の中にデジタルの機能を入れることによって、その製品がサービス化していく。例えば車も、それを使っている人の情報が全部取れ、そこで色々なサービスをしていくサービスマシンのようになるかもしれない。そういうことを考えていくということですね。それをこれまでのCIOがやってこなかったんです。実際には、デジタル関係、広い意味ではITの予算がCIO以外のところでものすごく増えている。ビジネスを進めている事業部でCIOの管轄下でないITがものすごく増えている。大きなエンタプライズのシステムを作る以外のところで、例えば、新しい製品に組み込みたい、新しいサービスを作るのに使いたいということで、色々な事業部がバラバラにやり出したんです。これは企業としては歓迎できない。自由にイノベーションを起こすことは必要ですが、ある程度企業の戦略にのっとってやっていく必要があります。企業の戦略に結び付いて、製品の中にデジタルを入れていくとか、サービスにデジタルを入れていくとか、そういうことをつないでいく人が必要なんです。そこにCDOの存在意義があります。それから、CDOは新しいものを作るために古いものを壊さなければいけない。壊さないと新しいものは生めない。こちらに移ってくださいというトランスフォーメーションをやらなければいけない。実はこれは、もともとCIOとCMOが連携しながら実現できるのではないかと期待されていたのですが、残念ながら実現しませんでした。

松本 確かにCIOというと社内の情報システム部門というイメージがあって社内向きですね。外を向いていない感じがあります。

神岡 その通りです。ところが10年程前は、CIOにもそういうことが期待されていたんです。CIOにイノベーションとか、リインベンションとか、トランスフォーメーションとか、私はビジネストランスフォーメーションという言い方をしていたのですが、それをCIOがやるべきだということを私自身も書いていた。しかし結局CIOは、ITのインフラとかメンテナンスとか、トラディショナルなデベロップメントとかセキュリティで手一杯になってしまった。手一杯なので、新しいデジタルの世界を時間的に担えない。もう一つはスキルセットが大きく違う。松本所長のご専門だと思いますが、設計の仕方が全然違うので、部下ができたとしてもついて来られない。設計をどうやるかというノウハウも分からない。企業の中に昔で言うITを設計するノウハウ以外に、デジタルに対応した設計をする仕組みがまだ確立できていないんです。デジタルのトップがIT部門に作ってくださいと言ってもIT部門がそれに応えられない現状があります。

CDOが備えるべきスキルとは何か

松本 CDOに求められるスキルとはどういうものなのでしょうか。新しいサービスにつながるようなユニークな発想や幅広い見方といったスキルが求められますね。

神岡 どういう人材が必要かと言うと、当然デジタルやテクノロジーに対する理解やセンスが求められます。ビジネスのセンスも必要です。後は自社だけでできないのは明らかなので、外との連携ができること。更にこれが一番大きいと思いますが、組織を変えることができるか、組織の活動を変えることができるか、ですね。組織の価値観を変えることができるか、その点のリーダーシップですね。

松本 それはCEO(Chief Executive Officer)の役割ではないのですか？

神岡 そうです。ですからCEOがやっても良い仕事だと思います。ただCEOは今動いているビジネスを見なければいけないので、変革させるためのリーダーを常に横に置いておくということです。CDOが次に何になるかと言うとCEOになる確率が非常に高い。変革をすることはCEOに求められている大きな能力ですが、これから5年から10年のCEOには、デジタルを使って会社を変えていく能力が求められています。CDOはCEOの片腕だと考えて良いと思います。

松本 CDOに求められる能力というのは、非常に幅広いものだと思います。技術的な知識はもちろんですが、外との連携を図るパイプや力を持っていないといけませんし、なおかつ顧客志向、

顧客目線で顧客が求めているものが分かる、そんな市場感覚も持っていないといけないかもしれません。本当にスーパーマンしかできないのではないのでしょうか。

神岡 はい、とても難しい役割です。先ほどお話が出ましたが、CIOの中にもCDO的な役割を果たしている人がいると同時に、CMOの中にもCDO的な役割を果たしている人がいます。しかし、CMOはこれまでのマーケティングのテリトリーでコミュニケーションの部分の役割を負っている。コミュニケーションにもデジタルトランスフォーメーションは必要ですが、このコミュニケーションに基づいて新しいものづくりをすとか、新しいシステムを作っていくとか、新しい製品を作っていくとか、イノベーションにつなげていくためにはCMOではテクノロジーの力が足りないんです。

松本 CDOも以前はマーケティングや営業出身者になる場合が多かったけれど、最近はテクノロジーをバックグラウンドにした人がCDOになるケースが増えていると聞きました。

神岡 その通りですね。マーケティングのデジタルチャネル、インターネットのコミュニケーション以外のところでも、AIやビッグデータ、IoT、更にVRやARなどあらゆるものを使わなければいけません。マーケティングの人が今まで扱ってきたテクノロジーよりも更に広い範囲のテクノロジーを理解し、使いこなさないとイケない。そういう意味では、CTOの中にもCDOになれる素質のある人がいる。テクノロジーに詳しいだけではなくて、トランスフォーメーションにも向いている人がいる。そういう人はCDOになれる可能性がかなりあると思います。

CDOをいかに育てていくか

松本 そういう人材はどのように育てれば良いのでしょうか？

神岡 それは大問題です(笑)。今まで誰もやったことがありませんし、誰も方法が分からない職です。あるCEOはCDOを実際に起用する際に、4つの条件を求めました。「イノベーションできる人」「シリコンバレーとつながっている人」「自分でビジネスをやったことがある人」「大企業にいたことがある人」です。

松本 大企業にいたことがある人というのはおもしろいですね。

神岡 大企業にいたことがない人は、変革するときに大企業ではどういうところが大変か分からないんです。ずっとスタートアップで新しい事業をゼロから作ってきた人には、変革という感覚がない。古いものを壊して新しいものを作る難しさが分からないことが多いのです。

松本 ゼロからなら新しいものが作れるかもしれないけれど、既にあるものをどう変えていくかということには慣れていないわけですね。ところで、技術が分かる人をマーケティングやイノベーションが分かるように育てていけば良いのか、もともとマー

ケティングやイノベーションをやってきた人に新しい技術を覚えていてもらうのが良いのか、その点はどうお考えですか？

神岡 その疑問を私は10年以上持ち続けているのですが、両方の可能性があるとしか言いようがありません。どちらが近道ということはありません。それぞれの人が違う意見を持っています。テクノロジーのバックグラウンドがある人が良いと言う人がいれば、いやビジネスが分からなければダメで、テクノロジーは外部の人を使えば良い、と言う人もいます。その会社がどういうターゲットに向けて新しい製品を作っていくか、ということにも関係すると思います。飛行機のエンジンを作っているような会社は、テクノロジーのことが相当分かっているなければダメでしょうし、化粧品を作っている会社ならそこまでテクノロジーのことを知らなくても良い、ということがあるかもしれません。

松本 トランスフォーメーションという意味で言うと、社内で育てるより外部から招聘したほうが良いということになりませんか？

神岡 そういうマインドを持ったCEOは多いですね。実際、日本の今のCDOはほぼ全員が外部からです。海外の例を見ても、外部からの場合のほうが多いですね。ただ、当然コンフリクトは起こります。それをどう乗り越えるのか。私は古いものと新しいものが同居するのは仕方がないと考えていて、この矛盾を常に内包しながら進んでいくしかないと思っています。今新しいものもいつか古くなります。すると次のものに変えていかなくてはなりません。そしてまた次のものに。古いものがある日突然新しいものに全部入れ替わるということはないわけです。古いものを維持しながら、徐々に新しいものに変えていくというトランスフォーメーションを、少なくともここ10年間くらいは繰り返しやっていくしかない、そういう時代なのではないかと思っています。

リスクを取ることが必要に

松本 デジタルトランスフォーメーションを進めていく上では、リスクもありますね。ある程度経営基盤を安定化させた上で取り組みないと、いきなりガラリと変えてしまって、しかしそのトランスフォーメーションが失敗して事業全体が行き詰まってしまっては企業の存続にもかわります。その見極めのようなものが求められるのではないかと思います。

神岡 その感覚は経営者に最も求められるでしょう。ここまではリスクを取ろうという線引きですね。今までの経営はリスクを最大限避けて進みましょうというやり方でした。それがある程度リスクを取りながら進みましょうというスタイルに変わってきたような気がします。ある程度不確実性がある状態にもかかわらず前に進む、しかし玉砕してしまうわけにはいかないのが、リスクを見てやっぱりやめようということも含め、舵取りをしていく。今まではあらゆる

ることのプロセスが明確に分かれていました。ここからここまでは何々をする時期、次のここからここまでは何々をする時期、そしてこの時期でリスクをゼロにしてスタート、というようなイメージです。ITのシステムも、ウォーターフォール型で作ってこの段階が終わったらこれ、この段階が終わったらこれ、という風に進んでいく。しかし今はこの段階が終わってもリスクを引きずりながら続いていく、つまり確定していないものを内包しながら次のステップに進んでいくということです。これは今確定しなければいけないのか、後回しで良いのか、迷いながら進めていくことになります。その辺りの判断力がCDOの資質として色々なビジネスの色々な場面で求められるのではないかと思います。

松本 スタートアップの企業であれば、リスクがすべてなので覚悟して進んでいくわけですが、ある程度の経営基盤を持った企業がデジタル化を進めていく上では、リスクをどう取れば良いのか分からないことが予想されます。そのような問題に直面したときのガイドのようなものはあるのでしょうか？

神岡 多くの企業が直面する問題ですね。どこまでリスクを取れば良いのか、という質問はよくいただきます。しかし、リスクの部分はいくら論理的に計算してここまでと分かってても、どこかでえいやと割り切ることが必要です。その感覚を持っていないければなりません。経営というのはロジックとクリエイティブ、サイエンスとアートの両方を持っていないとできない。ロジックがないと、組織が納得して動いてくれない。しかし、ステイブ・ジョブスなどもそうだったと思いますが、ある場面ではかなり無茶なことをして経営を進めてきたことでしょう。その辺りのリスクの取り方の感覚が重要になってくるのではないかと思います。

アジャイルへの誤解を解く

松本 我々IPA/SECではソフトウェアの開発を中心にしていますが、最近はや件定義もあいまいなまま進めざるを得ないことが多くあります。では、どの程度まであいまいにしておいて良いのか、ガイドがあるかと言ったらそんなものはないんですね。感覚的にこの辺りまで決めておけば良いかな、ということでスタートしています。そのあいまい性をどこまで、どの工程まで引きずることができるのか、そこが難しくなっています。アジャイル的にスパイラルアップで繰り返していくというやり方も有効なのかなと思います。

神岡 ある種のフォーマットを作って、設計をフォーマルにしていくことはできますが、しかしそれはリスクそのものが減るということではありません。間違いが減るということでしょうか。

松本 おっしゃる通りです。IPA/SECでは『つながる世界の開発指針』というものを作っています。IoT時代になると色々なものがつながり、つながることによるリスクが起きてきます。技術

的にもそうですし、またビジネス的にも起こり得ることです。開発者が想定もしなかったものを利用者が勝手につないでしまうケースとか、あるいは自分のところのシステムのトラブルがほかのシステムに悪い影響を与えてしまうとか、色々なことが考えられます。その際に考えるべき視点を17個ほど示しています。

神岡 それは重要ですね。

松本 ただそれを必ず全部対処してくださいとは言えません。企業によってどこまでリスクを取るかということは違うので、そこは企業にお任せする。ただ、少なくとも検討はしてください、と言っています。

神岡 それは非常に重要です。チェックリストのような形にされているのだと思いますが、こういうリスクがあるということを知った上で次に進みますか、ということですよ。今企業は色々なところとコラボレーションして、様々なオープンAPIを使ってつないでいます。このAPIを使ったときに、こういうことが起こり得ますよ、そういう世界であなたはやっているんですよ、ということ意識させるための設計のフレームワークは必要だと思います。

松本 その辺りまで、ある程度きちんと出していく必要があると思っています。

神岡 ソフトウェア開発と同じように、経営やビジネスのやり方に対して、アジャイル、あるいはアジリティということについて、私はもう20年くらい前から言ってきました。しかし、アジャイルを知っていて、これが重要だと分かっているということと、アジャイルを実施できるかということの間にはものすごく大きな差が存在します。これをどうやってやるかということはリーダーの仕事です。マインドセットを変えなければいけないし、カルチャーを変えなければいけない。こういう失敗はあり得るということは理解しておきましょう、あるいはこういう失敗が起きたときはそれを許容し、次にどうつなげるかということに意識を向けましょう、というように、組織のカルチャーや行動様式が変わらないとできないんです。ITを担当する人も、設計があいまいなままビジネス側から言われたことでもすぐに拒否するのではなく、一緒に考えていきましょうとか、設計側からビジネス側にフィードバックするような流れを作るとか、仕事の仕方も全部変えなければいけないのではないかと考えています。

松本 そうですね。IPA/SECもアメリカのカーネギーメロン大学と定期的に議論をしています。彼らは国防総省(DoD)の仕事のアジャイルで進めたいという提案をしたそうです。しかし当初は全く受け入れてもらえず、国防システムに関してそんないい加減な開発はやめてくれと言われたそうです。そこで考え方から変えいく努力をしたそうです。まず誤解を解く。アジャイルと言うと、いい加減に作ってとにかく世の中に出して悪いところは直せば良い、といったものとして捉えがちだけれど、そんなことはない。アジャイルでやる場合もきちんと設計書は作り、

試験もやって品質を確保して出していくんです、ということを書いて、徐々に理解が深まっていったそうです。

神岡 計画的に作り上げるシステムは絶対に必要ですが、例えば国防であっても環境の変化に応じてやっていかなくてはならないことがあるわけですね。一度計画を作ったら3年先にできるものを目指してそのまま進むということではない。アジャイルには誤解があるんですね。そもそもアジャイルは完璧ではないけれど、こういうメリットがある、ということはある。アジャイルの本質は何かということがもう少し伝わると良いですね。

松本 全部アジャイルでやれと言っているわけではないですよ。要求がなかなか決めにくいようなケースはアジャイルでやっていったほうが、よりスピードアップできるということだと思います。アジャイルはまだまだ進化していく過程にあるものだと思います。

神岡 これまでは、ITは日本でも海外でもコアコンピタンスではなかった。しかし、今ビジネスではITが最初から設計に入らなくてはならず、どういう新製品を作るか、新しいサービスを作るかという最初の戦略的なところから一緒にやらないといけない。そういう意味ではITやデジタルがコアコンピタンスに入ってきていると思います。ビジネス側も意識を変え、組織もやり方も変えなければならないと思います。

松本 個人的に気になっているのは、IoTにしろAIにしろ、とにかくこれを使うんだ、と手段が目的化してしまって、本来何のためにどういう事業やサービスを展開すべきなのかという議論が後回しになっているのではないかと、ということです。

神岡 同感です。今経営者は非常に大きなプレッシャーを感じていますね。ITの部門長やCIOを呼んでAIを使って何かできないのか、と聞いたりしている。ただ、CDOの幾つかの活動を聞いていると、テクノロジーについて社内の各部門にレクチャーしたり、経営者向けに今のデジタルはこうなっているといったワークショップをかなり行っているようです。そういう知識がすべての経営層に理解されていないと新しいことはできません。AIを入れるとすべてが解決するかなのようなイメージで「うちもやりましょう」と短絡的になってしまう。CDOの幾つかの役割をケーススタディー的に分類して示した論文があって、一つはアントレプレナー、もう一つはエバンジェリスト、つまり伝道師、更にコーディネーターという役割に分かれると書いていました。

松本 なるほど。文化と言うと大げさかもしれませんが、考え方を変えさせるような伝道師の役割を担う面もあるということですね。

神岡 ルールだけ変えてもダメで、価値観から変えていかなければならないということですね。CDOはそれをやる人なんだと思います。

松本 IPA/SECの中でも、最近はシステムズエンジニアリング

について取り組みを進めていますが、かなりデジタル化の考え方に近いんじゃないかと思っています。

神岡 私もとても親和性が高い気がしています。

松本 システムズエンジニアリングでは、開発すべき機能から考えるのではなく、開発する目的志向で俯瞰的に見るということを強調しています。何のために作るのかという目的を明確にしてから開発すべきものを決めていきましょうということで、正にデジタル化の考えに近いと思います。どういう新しいビジネスやサービスを考えていくのか、それに伴ってどうやってトランスフォーメーションをしていくのかを考えて設計していくということなので、すごく親和性が高いと感じます。

神岡 今までは、こういうのを作ってください、と言われて、いかに効率良く作っていくかということを考えていた。これがどう使われるのかというパーパスオリエンテッドなところまでエンジニアリングが入っていくということだと思います。必要なことだし大事なことだと思いますね。

同時に求められる組織再編

松本 日本のシステムズエンジニアリングへの取り組みは遅れています。ドイツのフラウンホーファー 研究機構に委託して比較的取り組みが進んでいるドイツの状況の調査をしたことがあって、企業がシステムズエンジニアリングの考え方を取り入れるときに苦労したことは何だったかと聞くと、組織の再構築が一番の問題だったという回答でした。つまりビジネスやサービスを変えていくのに伴って企業内の組織を変えていかなければならない、それが一番大変だったと。資金面やリソースの問題もありますが、最大の問題は組織再編です。

神岡 組織というとストラクチャーの問題、ガバナンスの問題、その両方があると思います。あるいは広い意味でシステム、仕組み、手順といった問題も絡みますね。

松本 CDOは社内の組織を変えていくということまで担う必要があるということでしょうか？

神岡 ある意味ではそうですが、そこまでCDOが一人で担うのは無理だと思います。GEなどもそうですが、デジタルトランスフォーメーションをしようとしてCDOが苦しんでいます。彼らの考え方はCEO、CDO、CMO、CTO、みんな含めてデジタルトランスフォーメーションを担うということです。

松本 確かにCDOだけで全部こなすのは無理ですね。

神岡 CDOが中心となって進めるけれども、すべての役員がかかわってサポートしながら進めていくという考え方です。現在は、チーフデータオフィサーも重要になっています。同じCDOなので紛らわしいですが、これはCIOが次になる職業になる可能性がある

と思います。今企業の中にはデータが山のように溜まっています。企業からも一般の消費者からも入ってくる。ところがこれを全然使いこなせていません。宝の山なのに整理もできず、データガバナンスもない状況です。IBMなどではチーフデジタルオフィサー、チーフデータオフィサー、更にチーフアナリティックオフィサーが存在します。チーフアナリティックオフィサーは、集められたデータを分析して何か答えを出していくという存在です。

松本 それだけ役割分担が明確になってきているのですね。

神岡 同じデジタルやIT関係でも、それだけオフィサーが増えています。今まではCIOに全部任せきりだった部分を、変革はチーフデジタルオフィサー、データを使って何かをするのはチーフデータオフィサー、そのデータを持って資産として何かをするというときに、経営層からこういうことがしたい、だったらどうすれば良いかという質問があったら、チーフアナリティックオフィサーがチーフデジタルオフィサーやチーフデータオフィサーと連携して社内のデータをかき集めてきて分析して、こうすべきですという答えを出す。このように役割分担をするようになりました。

松本 確かに、これからはデータ駆動型社会になると言われていますが、じゃあ具体的にデータをどう活用するか、という具体策になかなか落とし込んでいません。

神岡 手始めにデータサイエンティストを作りましようと言っていますが、作っても社内で孤立しているのが現状です。

松本 分析の手法は身に付くかもしれないけれど、どういう観点から何のためにデータを分析するのかという目的を明確にしないと、データのためのデータ分析になってしまいますよね。

神岡 その通りです。先ほどフラウンホーファーのお話を伺いましたが、組織がそれに伴っていないければなりません。データサイエンティストだけ雇ってきても何もできません。この人たちが組織の中で、どういう役割を果たし、ほかと組織的に連携していくのか、そこが明確になっていないと有効に使えないと思います。

CDOというブランドを確立する

松本 先ほど海外では外部からCDOを招聘している企業が多いと伺いました。しかし日本は人材の流動率が極端に低く、特有の組織風土もあります。外部の人が入るのはなかなか難しい。その中で、CDOが日本に根付いていくためにどのような変化が必要だとお考えですか？

神岡 日本の場合、CIOとCMOは企業に縛られていて動かないことが非常に多いのですが、CDOは動くものだ、というブランド、あるいは認知を早く作ったほうが良いと思います。CDOという

のはこういう職業で、企業を渡り歩くものなのだ、と。その企業にないものを持ってきてくれる。業種も違う、経験も違う人が動いていく職業なんですよ、ということをやうまく定着させていくと良いと思います。

松本 そういう情報交流の場のようなものがあると良いかもしれませんね。CDOの人たちの情報交流も必要でしょうし、それ以外の、ひょっとして自分に向いているかもしれないと思った人が、そのキャリアパスを作れるような集まりがあると良いですね。

神岡 CDOはある意味スーパーマンで、その意味ではまだ一人もいない。今はその一部を持っている人ばかりだと思います。残りはこれから学んでいかなければいけないので、本当はCEOがそういうことを理解して、失敗を覚悟して起用し、育てて欲しいと思っています。大きくまとめて言えば、CDOというのはリスクを取ってやっていく仕事なのということ。だから、少しぐらいミスがあっても前に進んでいけるタイプの人が向いているのかもしれない。

松本 そういう存在なんだというブランディングが重要ですね。IPA/SECでも、システムズエンジニアリングのようなシステム思考の考え方を広げていかなければと考えています。そのためにもシステム指向ができる人の「スキルセット」を定めて、体系的にきちんと育成していくことが重要です。

神岡 デジタル化やデジタルトランスフォーメーションが成り立つためには、それをシステムとして作れる人がいなければいけません。ところが、この人たちが今まで通りの開発や設計しかできなければ、いくら良いCDOがいてもそれはできないと思います。

松本 そうですね。我々にも反省すべき点があって、今までは要件定義は漏れなくきっちりやりましょう、ということをやずっとやってきたんです。

神岡 いえ、大学の教育も同じです。私もそうでした。ちゃんと計画を作ってチェックして、おかしなところがあったら、一から計画を作り直して、という教育をしてきました。今は計画にばかり時間がかかるなら先に進んだほうが良い、ということです。ただし、いい加減なまま進まないという歯止めは必要です。

松本 きっちりやるんだけれど、あいまいなまま進めざるを得ない所もあるということを意識しながら進めるんだと。そのリスクの存在を意識するのとならないのでは全然違いますからね。これは日本の特性だと思いますが、日本は現場力が強いこともあり、なかなかやり方を変えながらないところがありますよね。今まで通りやっていたら現場の力でちゃんとしたものが作れるんだという文化が強い。そこも変えていかなければならないという気がします。

神岡 同感ですね。基本的には二つだと思います。一つは危機

感を共有すること。このままいくとみんなタイタニックになってしまいますよ、ということをみんなが理解する。もう一つは、インセンティブです。どういうものが良いか、それは分からないけれども、今やっていることを壊すわけですから、それなりの動機付けが必要でしょう。

松本 そうですね。世代もどんどん変わりつつあるし、若い人にはSNSなどが当たり前の世界になっているので、そういう人たちが中心になるような形にしていかなければいけないのかもしれないですね。

神岡 デジタルネイティブと呼ばれる人たちですね。その人の感覚でものを作ったり、やっていくことが自然になっていくでしょう。そういう人たちをやうまく使い、活躍してもらおう。我々のような年寄りがそれを支える(笑)。日本ではイノベーションはなかなか起こりませんでした。イノベーション専門のチームを作ったりするんだけど、私はどちらかと言うとチームと言うより全員がイノベーションにかかわっていくような仕組みが良いのではないかと思います。日本の中間層は優秀で、こだわりも持っています。その人たちの考えていることをどうやって結集できるか、それが日本的なイノベーションの進め方ではないでしょうか。それが、CDOとかCIO、CEOのやるべきことであり、そこにもデジタルが使えるのではないかと、思います。

松本 そうですね。日本は突出した個人が引っ張るというよりは、集合体で、組織の力でやっていくところが強みですから、そういう発想が重要になるかもしれないですね。日本的なイノベーションの進め方になるかもしれません。今日は大変貴重なお話をありがとうございました。



ISBSGデータを用いた見積もり研究に対する IPA/SECデータを用いた外的妥当性の評価

山田 悠斗^{※1}江川 翔太^{※1, ※2}松本 真佑^{※1}角田 雅照^{※3}楠本 真二^{※1}

ソフトウェア開発プロジェクトの開発工数・期間の見積もり技術に関する研究が盛んに行われている。その際、評価のためのデータとして豪州の団体が世界各国より収集したプロジェクトデータであるISBSGデータが用いられることが多い。一方、一般に既存研究で得られた知見に対する追試 (replication study) が重要であるとされている。追試を行うことによって特定の条件や環境において得られた知見に対する再現性や、異なる実験条件から得られる知見の差異を調査することができる。また、追試を行い様々な条件から得られた知見を統合することによって、新規開発プロジェクトにおいてそれと類似した条件での知見を再利用することが可能となる。本稿では、ISBSGデータが用いられている見積もりの既存研究を対象にして別データを用いた追試を実施する。既存研究とは異なるデータセットを用いて実験を行うことで、得られる知見に差異が生じるかを調査する。過去5年間でISBSGデータが利用されていた論文22本の中から4本の論文を選択し、IPAが日本の各種ベンダ企業より収集したプロジェクトデータであるIPA/SECデータを用いて追試を行った。その結果、2本の論文では既存研究と類似した知見が、残りの2本の論文では既存研究と異なる知見が得られた。

Evaluation of External Validity Using IPA/SEC Dataset for Effort Estimation Studies Using ISBSG Dataset

Yuto Yamada^{※1}, Shota Egawa^{※1, ※2}, Shinsuke Matsumoto^{※1},
Masateru Tsunoda^{※3}, Shinji Kusumoto^{※1}

Many research studies of effort and duration estimation of the software development project have been conducted. In the studies, ISBSG dataset which includes actual software development data collected by the Australian organization from all over the world has been frequently used to evaluate the results. In the field of empirical software engineering, replication studies for the knowledge and experience finding from past studies are very important. By conducting the replications, we can evaluate the repeatability of the results of past studies got from the specific context and the difference between the results and the replication ones. This paper conducted replication studies for past ones of software effort estimation where ISBSG dataset were used in the evaluation. We selected four existing studies from twenty-two papers published within last five years. In the replications, we used IPA/SEC dataset that Information-technology Promotion Agency (IPA), Japan, have collected from Japanese software development companies. As the results, one replication produced the similar results to previous two and other two did different ones. Then, we discussed the reasons of the different results.

※1 大阪大学 大学院情報科学研究科

※2 ソニーグローバルマニュファクチャリング&オペレーションズ株式会社 設計部門 ソフトウェア設計部 テスト技術2課

※3 近畿大学 理工学部情報学科

1 はじめに

ソフトウェア工学はソフトウェアの開発、運用、保守に関して体系的、定量的にその応用を考察する分野であり、この分野で扱われている技術の中にソフトウェア開発の見積もり [Radatz1990] がある。ソフトウェア開発においては初期の段階から全体のコストや工数・開発期間などを正確に見積もることが重要であるとされており [Boehm2000]、見積もりの誤りがプロジェクト失敗へとつながる場合がある。この問題を解決するためにソフトウェア開発の見積もりに関する研究が盛んに行われている。見積もりに関する研究の評価には豪州の団体が世界各国より収集したプロジェクトデータであるISBSGデータ [ISBSG] が用いられることが多い。過去5年間における主要国際会議の論文誌を調査した結果では、見積もりに関する研究が行われている83本の論文のうち22本の論文でISBSGデータが利用されていた。

また、実証的ソフトウェア工学 [Kitchenham2008] の分野では、既存研究で得られた知見に対する追試 (replication study) が重要であるとされている [Silva2014, Shull2008]。研究に対する追試とは、ある研究に関して、実験の条件や環境を部分的に変更して実験を再現することである。追試を行うことによって特定の条件において得られた知見が別の条件においても再現できるかどうか、異なる条件では別の知見が得られるかどうかなどを調査することができる。これらを調査することで、研究成果に対する妥当性の評価を行うことができる。また、様々な条件から得られた知見を統合することで、新規開発プロジェクトにおいてそれと類似した条件での知見を再利用することが可能となる。

そこで、本研究ではISBSGデータが用いられており、開発工数・期間の見積もりに関する既存研究を対象にした追試を実施する。追試を行うにあたって明らかにするResearch Questionを3つ設定した。(RQ1)は、過去5年間における見積もり研究のうちISBSGデータが用いられた研究はどの程度存在するか、である。(RQ2)は、追試が可能で、現場でもすぐに適用できる研究はどの程度存在するか、である。(RQ3)は、ISBSGデータを用いた場合とIPA/SECデータを用いた場合から得られる知見に差は生じるか、である。異なる知見が得られた場合は結果が異なった原因の考察や追加の調査につなげることができ、同様の知見が得られた場合は既存研究の外的妥当性がより高められたと判断することができる。

以降、2.では研究の背景となる関連研究について述べる。3.では追試を行うための準備事項について述べる。4.では選択した論文に対して追試を行った結果と考察について述べる。5.では追試全体に対する考察について述べる。6.では主な結果と今後の課題をまとめる。

2 準備

本節では研究の背景となる諸用語や関連研究について簡単に述べる。

2.1 線形重回帰分析

ソフトウェア開発規模の見積もりには、多変量回帰分析の一手法である線形回帰分析が多く用いられている [Boehm1981]。線形回帰分析では予測対象となる目的変数と予測に必要とする説明変数との関係を一次式で表したモデルが作成される。一般的なモデルは式(1)の形で表される。目的変数である \hat{Y} には予測の対象となる工数などが、説明変数である X にはソフトウェアの規模や要因といった予測対象を導くために必要となる要素が当てはめられる。 a_i と b は予測対象の実測値と予測値の残差が最小となるように決められる。

$$\hat{Y} = a_1X_1 + a_2X_2 + \dots + a_nX_n + b \quad (1)$$

精度の推定には以下の式(2)~(6)によって求められる5つの評価指標 [Foss2003] であるAR (Absolute Residuals), MRE (Magnitude of Relative Error), MER (Magnitude of Error Relative to the estimate), BRE (Balanced Relative Error), Pred (25) が多く用いられる。MREは実測値から見た予測値の相対誤差を、MERは予測値から見た実測値の相対誤差を表す。BREは過大見積もりや過小見積もりに対しバランスの良い評価を行うことができる。AR, MRE, MER, BREは値が小さいほど、Pred (25)は値が大きいほど見積もり精度が良いと評価される。また、各指標の平均値は指標名の先頭にM, 中央値にはMdを付与することとする。例えば、MREの平均値はMMRE, 中央値をMdMREと示す。

$$AR = |\text{実測値} - \text{予測値}| \quad (2)$$

$$MRE = \frac{AR}{\text{実測値}} \quad (3)$$

$$MER = \frac{AR}{\text{予測値}} \quad (4)$$

$$BRE = \begin{cases} MRE & (\text{予測値} - \text{実測値} \geq 0) \\ MER & (\text{予測値} - \text{実測値} < 0) \end{cases} \quad (5)$$

$$\text{Pred}(25) = \frac{\text{評価指標の値が } 0.25 \text{ 以下であるデータ数}}{\text{全データ数}} \quad (6)$$

回帰モデルの予測精度を表す指標として、ほかに決定係数がある。これは重相関係数の2乗に等しく、説明変数が目的変数をどの程度説明できるかを表す。この値が大きいほど説明変数と目的変数の相関が強く、得られたモデルの予測能力が高いことを意味する。

2.2 ファンクションポイント法

ソフトウェアの規模を見積もる手法の一つにファンクションポイント法 [IFPUG] がある。この手法では、まずソフトウェアの持つ機能から5種類の基本機能要素を抽出し、それぞれの処理内容の複雑度からファンクションポイント (以降, FP) と呼ばれる点数を付ける。このFPから工数などの推定が行われる。5種類の基本機能要素とは以下に示す要素のことを言う [Kashimoto2000]。

内部論理ファイル (ILF): 計測対象のアプリケーション内で更新される, 論理的な関連を持ったデータの集合

外部インターフェースファイル (EIF): 計測対象のアプリケーションによって参照されるデータの集合 (更新されない)

外部入力 (EI): 計測境界外からのデータ入力によってILFの更新を行う処理

外部出力 (EO): 計測境界外へのデータ出力を含む処理のうち, 出力に派生データを含むもの

外部参照 (EQ): 計測境界外へのデータ出力を含む処理のうち, 出力に派生データを含まないものであり, 処理がILFを更新しないもの

アプリケーション全体でのFPの合計をアプリケーションFPと言い, アプリケーションFPにシステムの特性を考慮に入れて調整を加えた値を調整済みアプリケーションFPと言う。FP値は見積もり研究における規模の尺度としてよく利用されている。

2.3 ISBSGデータ

ISBSGデータは, 豪州の団体であるISBSG (The International Software Benchmarking Standards Group) [ISBSG] が世界24カ国に存在する組織や企業から実開発のデータを収集・整理したデータセットである。データの項目には開発工数や開発言語, 開発プラットフォームなどが含まれる。また, FPに対する品質ランクも収録されており, 信頼性の高いデータを選別することができる。リリースごとにデータ数は異なるが, 最新のデータセットには5,000以上のプロジェクトデータが118項目に分けて蓄積されている。

2.4 IPA/SECデータ

IPA/SECデータは, 独立行政法人情報処理推進機構 [IPA] が日本に存在する組織や企業から実開発のデータを収集し, 整理したデータセットである。データの項目には開発工数や開発言語など, ISBSGデータと共通した項目が多く含まれる。また, データの品質に関するランクも収録されており, 信頼性の高いデータを選別することができる。2014-2015版では, 3,541プロジェクトのデータが194項目に分けて蓄積されている。

2.5 実証的ソフトウェア工学

実証的ソフトウェア工学 [Kitchenham2008] とは, ソフトウェア開発現場での作業や実績に対する計測, 定量化とその評価, そしてフィードバックによる改善という実証的手法を行う研究分野である。ソフトウェア開発の課題である生産性の向上や品質の確保に対する有用なアプローチとして注目されている。

(1) 妥当性についての議論

実証的ソフトウェア工学では, 実在するソフトウェア開発データを用いたケーススタディを通じて提案手法の評価が行われることが多いが, このとき妥当性に関する議論が行われなければならない。妥当性には以下の分類 [Robert2013] が存在する。

内的妥当性

研究成果が研究の際に操作した要因から影響を受けている程度を指す

外的妥当性

ある研究から得られた成果を, 違った母集団, 環境, 条件へ一般化し得る程度を指す

構成概念妥当性

結果を得るために行った操作が適切である程度を指す

信頼性

他者が同様の手順で研究を行った場合, 研究結果が再現可能となる程度を指す

実証的ソフトウェア工学における妥当性に関する研究として, 文献 [Egawa2016] の研究がある。この研究では, 見積もり研究において研究成果の外的妥当性がどの程度意識されているかを調査するため, 過去の研究論文を対象とした網羅的なレビューが行われている。調査の結果, 対象となる89本の論文のうち, 研究成果の外的妥当性についての議論を行っていない研究論文が26本存在しており, 結論部分においてのみ言及している論文が31本存在することが示されている。このことから, 見積もりの研究に携わる研究者は, 研究成果の外的妥当性に関してより注意を払うべきであるということが主張されている。

(2) 追試 (replication study)

実証的ソフトウェア工学の分野では, 既存研究で得られた知見に対する追試 (replication study) が重要であるとされている [Silva2014, Shull2008]。研究に対する追試とは, ある研究に関して, 実験の条件や環境を部分的に変更して実験を再現することである。追試を行うことによって, 特定の条件において得られた知見が別の条件においても再現できるか, 異なる条件や環境では別の知見が得られるかなどを調査することができる。追試によって異なる知見が得られた場合は結果が異なった原因の考察や追加調査につなげることができる。同様の知見が得られた場合は既存研究の研究成果の外的妥当性がより高められたと判断することができる。また, 追試を行い, 様々な条件から得られた知見を統合することで, 新規開発プロジェクトにおいてそれと類似した条件での知見を再利用することが可能となる。

3 追試の準備

本節では追試を行うために必要な準備事項について述べる。

3.1 Research Questionの設定

追試を行うにあたって明らかにしたい問をResearch Question (RQ) として設定する。本研究では, 以下の3つを設定した。

- **RQ1**: 過去5年間における見積もり研究のうちISBSGデータが用いられた研究はどの程度存在するか
- **RQ2**: 追試が可能で, 現場でもすぐに適用できる研究はどの程度存在するか

- **RQ3**：ISBSGデータを用いた場合とIPA/SECデータを用いた場合から得られる知見に差は生じるか

RQ1は、ISBSGデータには複数企業のデータが含まれているため、ほかのデータセットに比べて外的妥当性が高いと思われるが、それがどの程度使用されているのかを明らかにするために設定した。RQ2に関しては、工数見積もりは現場での利用可能性が高くないといけませんが、実際には適用が難しい研究成果も多いと思われる。適用可能な研究がどの程度の割合で存在するかを明らかにするために設定した。RQ3に関しては、IPA/SECデータは日本国内の複数の企業から収集したデータセットであるため、このデータセットで成立する知見は日本国内の企業に適用してもある程度成立すると考えられる。ISBSGデータで得られた知見が、IPA/SECデータ、すなわち日本の企業でもどの程度同様に得られるのかを調査するために設定した。

3.2 論文の選択

見積もりに関する研究の評価にはISBSGデータが用いられることが多い。過去5年間における以下の主要国際会議の論文誌を調査した結果では、見積もりに関する研究が行われている83本の論文のうち22本の論文でISBSGデータが使用されていた。これらの論文を追試の対象とする。

- ACM Transactions On Software Engineering and Methodology
- ASIA-PACIFIC Software Engineering Conference
- Empirical Software Engineering and Measurement
- International Conference on Software Engineering
- International Conference on Software Maintenance
- IEEE Transactions on Software Engineering
- Information and Software Technology
- Journal of Systems and Software

22本の論文の中から、追試が可能であり、かつ追試を行うのにふさわしい論文を選択する必要がある。選択のために提示した条件と、各条件を満たしている研究を順に抽出した際の論文数の推移を表1に示す。表の通り、最終的に4本の論文 [Tsunoda2012-1, Lavazza2013, López-Martín2015, Tsunoda2012-2] が追試の対象として選択され、それぞれに関して追試を行った。4本の論文をそれぞれ、生産性に基づくモデル作成に関する論文 [Tsunoda2012-1]、FPの簡易推定に関する論文 [Lavazza2013]、ニューラルネットワーク (以降、NN) を利用した見積もりに関する論文 [López-Martín2015]、カテゴリ変数を用いたモデルに関する論文 [Tsunoda2012-2] と名称付けて説明する。

ただし、本論文では紙面の都合上、追試の詳細な結果に関しては、生産性に基づくモデル作成に関する論文、FPの簡易推定に関する論文、NNを利用した見積もりに関する論文についてのみ報告する。

表1 追試の対象となる論文の条件と抽出された論文数

論文の条件	論文数
評価の段階でプロジェクトデータが用いられている	20
入手可能なデータセットが用いられている	19
使用されているデータセットが5種類未満である	16
再現実験のために必要な情報が正確に記載されている	11
ISBSGデータの入手可能な情報が用いられている	9
現場での適用容易性が高い	4

3.3 生産性に基づくモデル作成に関する論文の概要

回帰分析に基づく工数見積もりモデルに関する研究が行われており、対象プロジェクトの生産性も考慮に加えた工数見積もりモデルの作成方法を提案している [Tsunoda2012-1]。なお、ここでの生産性はFPを工数で割った値として定義される。

通常、工数見積もりモデルを作成する際にはデータセットに蓄積されたデータすべてを対象として回帰分析を行い、1つのモデルを作成する。提案手法ではデータセットを生産性の高さによって複数に分類し、それぞれからモデルを作成する。そしてテストデータが持つ生産性の値の高さに基づき、複数のモデルのうち1つを選択して工数の見積もりを行う。現場で提案手法を実施する際、現行プロジェクトの生産性の推測はプロジェクトマネージャが行う。プロジェクトマネージャが生産性の推測を誤る確率を「推測誤り率」とする。

実験では、以下に示す3種類のモデルを作成し、見積もりの精度を比較する。Twoレベルモデルを用いて対象プロジェクトの見積もりを行う場面を図1に示す。また、実験を行う際の条件を表2に示す。

Noレベルモデル

生産性を考慮せずに回帰分析を行う従来の見積もり手法を用いるモデル

Twoレベルモデル

生産性の値の高さによってデータセットをHigh (高)、Low (低) の2段階に分類して提案手法を用いるモデル

Threeレベルモデル

生産性の値の高さによってデータセットHigh (高)、Medium (中)、Low (低) の3段階に分類して提案手法を用いるモデル

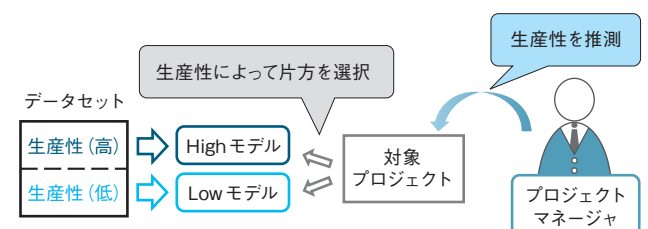


図1 Twoレベルモデルを用いた見積もり手法

表2 生産性に基づくモデル作成に関する論文における実験の条件

入力	FP, 開発言語, 開発種別, プラットフォーム
出力	工数
使用するモデル	Noレベルモデル, Twoレベルモデル, Threeレベルモデル
評価指標	MRE, MER, BRE

リリース9のISBSGデータ内の593データを対象として実験が行われた。その結果、推測誤り率が38%以下の場合、つまり現場のプロジェクトマネージャが生産性の推測を誤る可能性が低いと判断できる場合には、従来の手法よりも提案手法のほうが工数見積もりにおける見積もり精度が向上するという知見が得られている。

3.4 FPの簡易推定に関する論文の概要

FPの計測方法に関する研究が行われており、従来よりも簡易化されたFP推定モデルを提案している [Lavazza2013]。FPを計算する際に抽出される5種類の基本機能要素のうち、FPと最も相関の強い要素の規模のみを説明変数とした簡易FP推定モデルを作成する。相関関係の調査にはケンドールの順位相関係数とスピアマンの順位相関係数を用いる [Croux2010]。実験を行う際の条件を表3に示す。

表3 FPの簡易推定に関する論文における実験の条件

入力	FP, EI, EO, EQ, ILF, EIF
出力	FPと各基本機能要素の相関係数
評価指標	ケンドールの順位相関係数, スピアマンの順位相関係数

リリース11のISBSGデータが持つ600以上のデータを対象として実験が行われた結果、EIにおけるケンドールの順位相関係数が0.658、スピアマンの順位相関係数が0.839となり、FPと最も相関が強くなった。

3.5 NNを利用した見積もりに関する論文の概要

NN [Richard1987] を利用したソフトウェア開発期間の見積もりモデルの精度を調査している [López-Martín2015]。NNとは、人間の脳が問題を解く際の振る舞いを計算機上のシミュレーションによって表現したネットワークモデルである。今回は以下の2種類のNNであるMLP (Multilayer Perceptron) とRBFNN (Radial Basis Function Neural Network) を使用する [Park2003]。これらのモデルの精度を、重回帰モデルを使用した場合と比較して調査する。実験を行う際の条件を表4に示す。なお、表4におけるPred (25) は、ARに対する値である。

MLP

内部のニューロンが入力層、中間層、出力層に分かれており、ループせず単一方向にのみ信号が伝播するネットワーク

RBFNN

MLPの中間層で放射基底関数を用いて出力を計算するネットワーク

表4 NNを利用した見積もりに関する論文における実験の条件

入力	FP, 社内要員数
出力	開発期間
使用するモデル	重回帰モデル, MLP, RBFNN
評価指標	AR, Pred (25)

リリース11のISBSGデータ内の49データを対象として実験が行われた結果、2種類のNNモデルはいずれも重回帰モデルと比べて、MAR, MdARに関して精度が6%以上高くなった。このことから、NNを利用することによって開発期間の見積もり精度を

より高めることができると言える。

3.6 カテゴリ変数を用いたモデルに関する論文の概要

工数見積もりにおけるカテゴリ変数の扱い方に関する研究が行われている [Tsunoda2012-2]。カテゴリ変数とは性別、職業など一般に数や量では測れない変数を指す。回帰モデルの説明変数としてカテゴリ変数を使用する際には、対応方法の異なる様々なモデルが用いられる。以下の4種類のモデルを対象とし、これらの工数見積もりモデルを、カテゴリ変数を使用しないモデルと比較することで評価する。

ダミー変数化を用いたモデル

カテゴリ変数から0.1の2値を取る複数のダミー変数を生成し、それらを説明変数として回帰モデルを作成する。

層別を用いたモデル

変数の値の組み合わせにより、データセット内のデータをサブセットに分割する。そしてそれぞれのサブセットから回帰モデルを作成する。

交互作用モデル

ある説明変数の値によって、ほかの説明変数の効果に変化することを交互作用と言う。今回はダミー変数化を用いたモデルに、各ダミー変数とFPの積により作成した説明変数を加えたモデルを作成する。

階層線形モデル

グループごとにまとまりがあるデータを分析する際に用いられる。層別によって分割したサブセット間の関係性を考慮に加えて、サブセットごとに切片と傾きが変化するモデルを作成する。

リリース9のISBSGデータ内の558データを対象として実験を行った結果、カテゴリ変数を使用する4種類のモデルはいずれもカテゴリ変数を使用しない場合と比べて精度が約10%向上した。また、4種類のモデル間での精度の差は5%未満となり、見積もりにおいて同程度の精度が得られた。

3.7 使用する統計ツール

追試の中で回帰分析などの統計処理を行う際は、統計分析のフリーソフトであるR [Rproject] を使用する。

4 追試の結果

本節では、対象となる4本の論文に対してIPA/SECデータで追試を行った結果及び考察について説明する。

4.1 生産性に基づくモデル作成に関する論文

(1) データの選別基準

IPA/SECデータに蓄積されているデータのうち、表5に示す既存研究と同様の選別基準に従って抽出された189データを使用する。「本データの信頼性」とは、当該プロジェクトデータ

の合理性や整合性に関する信頼度を4段階(A~D)で評価した値であり、最も信頼度が高い場合はAと評価される。

表5 生産性に基づくモデル作成に関する論文でのIPA/SECデータの選別基準

項目	選別基準
FP実績値の計測手法	IFPUG
本データの信頼性	AもしくはB
FP実績値(調整前)	欠損していない
主開発言語グループ	欠損していない
開発プロジェクトの種別	欠損していない
開発対象プラットフォームのグループ	欠損していない

(2) 結果と考察

各モデルの説明変数にはFPの対数、主開発言語グループ、開発プロジェクトの種別、開発対象プラットフォームのグループを用いる。追試における実験の結果を表6に示す。TwoレベルモデルとThreeレベルモデルの値は、推測誤り率が0%の時点での数値である。更に、推測誤り率を増加させた際の見積もり精度の推移を図2、図3に示す。図2は横軸に推測誤り率が、縦軸にMBREの数値が示されている。図3は横軸に推測誤り率が、縦軸にMdBREの数値が示されている。

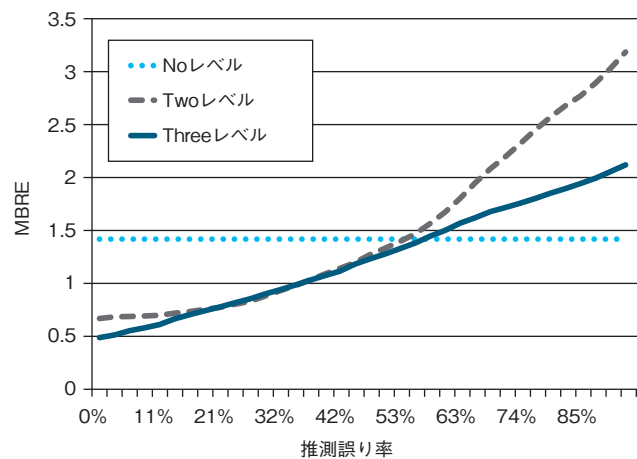


図2 追試における推測誤り率の増加によるMBREの推移

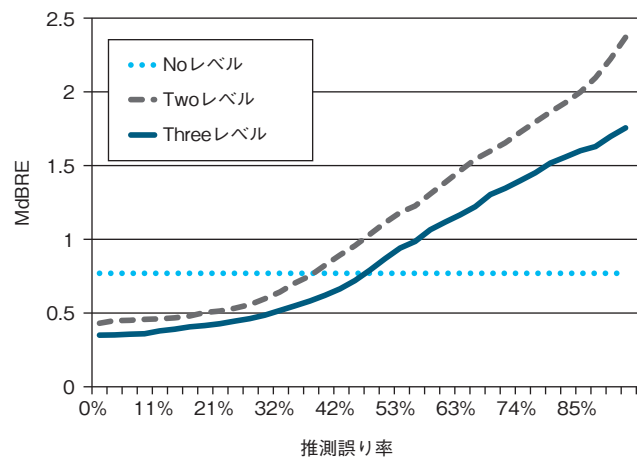


図3 追試における推測誤り率の増加によるMdBREの推移

表6 生産性に基づくモデル作成に関する論文の追試結果

レベル	MMRE	MdMRE	MMER	MdMER	MBRE	MdBRE
No	1.00	0.53	0.85	0.54	1.41	0.77
Two	0.50	0.33	0.46	0.35	0.65	0.43
Three	0.38	0.28	0.35	0.29	0.47	0.35

表6を見ると、推測誤り率が0%のときはいずれの評価指標でもNoレベルモデルよりTwo, Threeレベルモデルのほうが見積もりの精度が高い。図2, 図3を見ると、推測誤り率が増加するほどTwo, Threeレベルモデルの精度は低下している。しかし、推測誤り率が37%以下の時点では、MBREとMdBREのどちらにおいてもTwo, ThreeレベルモデルのほうがNoレベルモデルより精度が高い。

以上のことから、推測誤り率が低い状態、つまり現場のプロジェクトマネージャが生産性の推測を誤る可能性が低いと判断できる場合には、従来の手法よりも提案手法のほうが工数見積もりにおける見積もり精度が高くなると考えられる。これは既存研究から得られた知見と同様の知見である。

海外からデータを収集したISBSGデータと日本からデータを収集したIPA/SECデータで同様の知見が得られたことから、追試を行うことにより既存研究の研究成果の外的妥当性が高められたと言える。

4.2 FPの簡易推定に関する論文

(1) データの選別基準

IPA/SECデータに蓄積されているデータのうち、表7に示す既存研究と同様の選別基準に従って抽出された、122データを使用する。

表7 FPの簡易推定に関する論文でのIPA/SECデータの選別基準

項目	選別基準
本データの信頼性	AもしくはB
EI	欠損していない
EO	欠損していない
ILF	欠損していない

(2) 結果と考察

追試における実験の結果を表8に示す。ケンドールとスピアマンの順位相関係数を用いた際の、FPと各基本機能要素の相関係数の値が示されている。

表8 FPの簡易推定に関する論文の追試結果

要素	ケンドール	スピアマン
EI	0.663	0.836
EO	0.607	0.791
EQ	0.656	0.850
ILF	0.655	0.837
EIF	0.384	0.525

既存研究でISBSGデータを用いた場合、FPと最も相関の強い要素としてEIが得られている。しかし表8より、IPA/SECデータを用いた場合におけるFPと最も相関の強い要素は、ケンドールの順位相関係数の場合はEI、スピアマンの順位相関係数の場合

はEQとなった。このことから、FPと関連の強い基本機能要素は必ずしもEIではなく、データセットにより異なることが分かる。

よって、基本機能要素を用いて簡易FP推定モデルを作成する際は、使用するデータセットごとに適した要素を選択する必要があると考えられる。

4.3 NNを利用した見積もりに関する論文

(1) データの選別基準

IPA/SECデータに蓄積されているデータのうち、既存研究と同様の選別基準に従って抽出された、表9に示す36データを使用する。

表9 NNを利用した見積もりに関する論文でのIPA/SECデータの選別基準

項目	選別基準
本データの信頼性	AもしくはB
実績月数(開発期間)	2カ月以上
社内ピーク要員数	2人以上
開発対象プラットフォームのグループ	Windows系
主開発言語	第3世代言語
開発プロジェクトの種別	新規開発

(2) 結果と考察

各モデルの説明変数またはNNの入力値には調整済みアプリケーションFPの対数と社内ピーク要員数の対数を用いる。追試における実験の結果を表10に示す。

なお、リリース11のISBSGデータを用いて実験の再現を行った結果、こちらの準備した実験環境でNNモデルを用いると、MdARとPred(25)に関しては重回帰モデルより精度が向上したが、MARに関しては精度がやや低下した。これは、NNモデルを用いた際に、ごく一部のプロジェクトデータに関して本来の値と大きく外れた異常な予測値が発生しやすくなったためである。元となる論文では、NNモデルの動作がブラックボックス化されていることや、最適なノード数などを見つけることが困難であるということが問題として述べられている。このことから、NNモデルの環境や得られる結果を完全に一致させることは困難であると考えられる。よって今回は、ごく一部の異常値を無視すれば元の実験と類似した傾向の結果が得られたことから、元の論文と近い実験環境を構築できたと判断し、この実験環境で追試を行う。異常値による影響を減らすために、MARよりも異常値に関して頑健であるとされているMdARとPred(25)の値からモデルの精度を比較する。

表10を見ると、Pred(25)とMdARのどちらにおいても重回帰モデルより精度が向上しているNNモデルはRBFNNモデルのみである。これは既存研究と異なる結果である。

表10 NNを利用した見積もりに関する論文の追試結果

モデル	ニューロン数	MAR	MdAR	Pred(25)
重回帰	—	0.33	0.27	0.44
MLP	5	0.44	0.30	0.50
RBFNN	14	0.37	0.20	0.53

次に、追試ではMLPモデルの精度が重回帰モデルより低下した原因を考える。既存研究ではNNモデルの入出力値の関係性を調査している。その際、入力値である調整済みアプリケーションFPの対数と社内ピーク要員数の対数を説明変数とし、出力値である開発期間の対数を目的変数とする重回帰モデルを、データセットに蓄積されたすべてのデータに対して回帰分析を行い作成している。そのモデルが以下の式(7)である。AFP(Adjusted Function Points)は調整済みアプリケーションFPを、MTS(Max Team Size)は社内ピーク要員数を、Durationは開発期間を指す。このモデルの決定係数は0.560となった。

$$\ln(\text{Duration}) = 0.150 + 0.438 \times \ln(\text{AFP}) - 0.187 \times \ln(\text{MTS}) \quad (7)$$

追試においても同様のモデルを作成した。そのモデルが以下の式(8)であり、決定係数は0.398となった。

$$\ln(\text{Duration}) = -0.669 + 0.435 \times \ln(\text{AFP}) - 0.205 \times \ln(\text{MTS}) \quad (8)$$

既存研究でISBSGデータから得られたモデルのほうが追試でIPA/SECデータから得られたモデルより決定係数が高いことから、入出力の相関関係はIPA/SECデータよりISBSGデータのほうが強いことが分かる。また、NNモデルは入出力値の相関関係が強いほど予測精度が高くなるとされている[Hitokoto2012]。更に、RBFNNモデルはMLPモデルより安定した学習が可能であるとされている[Murakami2008]。以上のことから、既存研究よりも入出力値の相関関係が弱くなった影響をMLPモデルのみが受け、MLPモデルの見積もり精度が低下したと考えられる。

4.4 カテゴリ変数を用いたモデルに関する論文

紙面の都合上、詳細な結果は省略する。生産性に基づくモデル作成に関する論文と同様に、既存研究から得られた知見と類似した知見が得られた。

5 考察

ある手法を用いた際のコンテキストが、どの程度自身の環境に対応しているかを特定することは容易ではない、という課題が存在する。実際に2本の論文では、既存研究とはコンテキストが異なるデータで構成されたIPA/SECデータを用いた場合は異なる結果が得られた。ただし、FPの簡易推定に関する論文では、ISBSGデータを業種で層別した場合は業種ごとに異なる結果が得られたことから[Yamada2016]、コンテキストを特定するには層別を行い、データのフィルタリングを行うことが有効であると考えられる。

また、上記の問題に対し、コンテキストに強く依存しない手法を用いることも一つの解決策であると考えられる。類似した知見が得られた2本の論文では比較的コンテキストの依存が強くないため、この手法に関してはコンテキストを厳密に合致させなくても見積もり改善の効果が期待できる。

6 おわりに

本稿では、ISBSGデータが評価に用いられている見積もりの既存研究に対する追試を実施した。設定したRQへの回答として、RQ1については、ISBSGデータが用いられた研究は22本存在するという結果が得られた。RQ2については、4本の論文 [Tsunoda2012-1, Lavazza2013, López-Martín2015, Tsunoda2012-2] が選択された。RQ3については、2本の論文では既存研究と異なる知見が得られたことから、追加調査も交えてその原因に対する考察を行った。ほかの2本の論文では既存研究と類似した知見が得られたことから、既存研究の研究結果の外的妥当性が高められたという結果が得られた。以上のことから、IPA/SECデータで同様の知見が得られた研究については開発現場でも同様の成果が得られることが期待でき、かつ、適用も容易なので、工数見積もりの精度を高めるために適用すべきであると言える。

今後の課題としては、まず見積もり分野におけるほかの既存研究に対する追試が考えられる。実証的ソフトウェア工学では妥当性への脅威に関する議論が重要となるが、その必要性に対する認識はいまだに不十分である。よって、今回のような妥当

性に関する議論を継続的に行うことで、その必要性に対する認識をより広めていくべきである。また、今回の追試から得られた知見に対する追加調査も今後の課題として考えられる。例えば、FPの簡易推定に関する論文について、各業種と導かれた要素との関連性に対する調査が挙げられる。更に、外的妥当性をより高めるために、今回用いたデータセットとは異なる種類のデータセットで追試を行うことも必要であると考えられる。加えて、実環境に対する実験結果の対応度合いをいかにして特定するかという課題に対する、より詳細な調査が考えられる。様々な実環境との適応度合いや期待できる効果などを調査するには、実企業の協力を得た更なる追試が必要となる。

謝辞

本研究を行うにあたり、データを提供していただくと共に多大なご助言をいただきました独立行政法人情報処理推進機構技術本部ソフトウェア高信頼化センターの関係各位に深く感謝申し上げます。

本研究は一部、日本学術振興会科学研究費補助金基盤研究(S)(課題番号:25220003)、及び基盤研究(C)(課題番号:16K00113)の支援を受けている。

【参考文献】

- [Radatz1990] J. Radatz, A. Geraci, and F. Katki. IEEE standard glossary of software engineering terminology. IEEE Std, 610.12-1990.
- [Boehm2000] B. Boehm, C. Abts, and S. Chulani. Software development cost estimation approaches - A survey. Annals of software engineering, 10, 1-4, 177-205, 2000.
- [ISBSG] International Software Benchmarking Standards Group (ISBSG). <http://www.isbsg.org>.
- [Kitchenham2008] B. Kitchenham, et al. Evaluating guidelines for reporting empirical software engineering studies. ESE, 13, 1, 97-121, 2008.
- [Silva2014] F. QB Da Silva, et al. Replication of empirical studies in software engineering research: a systematic mapping study. ESE, 19, 3, 501-557, 2014.
- [Shull2008] F. J Shull, J. C Carver, S. Vegas, and N. Juristo. The role of replications in empirical software engineering. ESE, 13, 2, 211-218, 2008.
- [Tsunoda2012-1] M. Tsunoda, A. Monden, J. Keung, and K. Matsumoto. Incorporating expert judgment into regression models of software effort estimation. In APSEC, 1, 374-379. IEEE, 2012.
- [Lavazza2013] L. Lavazza, S. Morasca, and G. Robiolo. Towards a simplified definition of Function Points. IST, 55, 10, 1796-1809, 2013.
- [López-Martín2015] C. López Martín and A. Abran. Neural networks for predicting the duration of new software projects. JSS, 101, 127-135, 2015.
- [Tsunoda2012-2] M. Tsunoda, S. Amasaki, and A. Monden. Handling categorical variables in effort estimation. In ESEM, 99-102. ACM, 2012.
- [Boehm1981] Barry W Boehm. Software engineering economics. 197. Prentice-hall Englewood Cliffs (NJ), 1981.
- [Foss2003] T. Foss, E. Stensrud, B. Kitchenham, and I. Myrteit. A simulation study of the model evaluation criterion MMRE. TSE, 29, 11, 985-995, 2003.
- [IFPUG] IFPUG: Function Point Counting Practices Manual, Release 4.3. IFPUG, 2010.
- [Kashimoto2000] 柏本, 楠本, 井上, 鈴木, 湯浦, 津田. イベントトレース図に基づく要求仕様書からのファンクションポイント計測手法. 2000.
- [IPA] IPA 独立行政法人情報処理推進機構. <https://www.ipa.go.jp>.
- [Robert2013] Robert K Yin. Case study research: Design and methods. Sage publications, 2013.
- [Egawa2016] 江川翔太. 見積もり研究における外的妥当性の調査を目的とした系統的レビューと追試. 修士学位論文, 大阪大学, 2016.
- [Croux2010] C. Croux and C. Dehon. Influence functions of the Spearman and Kendall correlation measures. Statistical methods & applications, 19, 4, 497-515, 2010.
- [Richard1987] Richard P Lippmann. An introduction to computing with neural nets. ASSP Magazine, IEEE, 4, 2, 4-22, 1987.
- [Park2003] J. Park, R. G Harley, and G. Kumar Venayagamoorthy. Adaptive-critic-based optimal neurocontrol for synchronous generators in a power system using MLP/RBF neural networks. TIA, 39, 5, 1529-1540, 2003.
- [Rproject] R: The R Project for Statistical Computing.
- [Hitokoto2012] 一言, 桜庭, 小野. ニューラルネットワークを用いた洪水予測システムの開発. こうえいフォーラム: 日本工営技術情報, 20, 67-72, 2012.
- [Murakami2008] Masayuki Murakami. Practicality of modeling systems using the IDS method: Performance investigation and hardware implementation. PhD thesis, The University of Electro-Communications, 2008.
- [Yamada2016] 山田悠斗. ISBSGリポトリを用いた見積もり研究に対する追試. 特別研究報告, 大阪大学 基礎工学部, 2016.

提案依頼書に含まれる無理難題の分類



門田 暁人^{※1}



住吉 倫明^{※2}



神谷 芳樹^{※3}

提案依頼書 (Request For Proposal ;RFP) には、発注者 (ユーザ) による無理難題とも言える要求がしばしば含まれ、ソフトウェア開発受注後に大きな問題となる場合がある。本論文では、受注者 (ベンダ) の視点に立ち、RFPに含まれる無理難題を抽出・分類し、どのような無理難題が存在するかを明らかにする。また、その結果を事例集としてまとめることで、ベンダのリスク管理、及び、ユーザ自身によるRFPのチェックに役立つ。地方公共団体や病院システムなどの5つのRFPを分析した結果、抽出した無理難題を、(1)実績を求める要求、(2)技術的に実現が難しい要求、(3)仕様の分からない既存システムの移行、連携を求める要求、(4)将来の課題への対応を求める要求、の4つに分類した。とくに、(3)と(4)は、4つのRFPに含まれており、ベンダにとって損失発生の大きなリスクとなり得ることが分かった。

Classifying Unreasonable Requirements in Request For Proposals

Akito Monden^{※1}, Toshiaki Sumiyoshi^{※2}, Yoshiki Mitani^{※3}

Request For Proposals (RFPs) often contain unreasonable requirements that can cause troubles after software development has started. From the viewpoint of software developer companies, this paper identifies and classifies such unreasonable requirements from existing RFPs, and provides a collection of cases to help risk assessment in developer companies and to help improving RFPs by software purchaser companies. As a result of analyzing five RFPs written by public organizations, hospitals and so on, we were able to classify unreasonable requirements into 4 categories: (1) requirements of past installation experience, (2) requirements that are difficult to fulfill, (3) requirements to cooperate with existing systems whose specifications are unknown, (4) requirements to cope with future issues. Especially, (3) and (4) were contained in four RFPs, which can cause financial damage to software developer companies.

1 はじめに

提案依頼書 (Request For Proposal, 以後はRFPとする) とは、ソフトウェア開発プロジェクトの上流工程において、発注企業 (ユーザ) が、発注先の開発会社 (ベンダ) に解決したい課題や達成したい目的・目標を伝えると共に、具体的な技術仕様の提案を依頼するための資料である。

RFPは開発プロジェクトのベースとなる重要な書類であるが、RFPの記述が十分でないことに起因する品質問題が発生し、失

敗プロジェクトとなる場合が少なくない [Roth]。例えば、契約締結後に、ユーザの意図している要求とベンダが理解した要求との間のギャップが明らかになることがある [Port]。また、保守に関する要求がRFPに十分に記載されていなかったために、ユーザが十分な保守サービスを受けられないケースもある。このような問題を回避するために、従来ユーザ支援を目的として、提案依頼書に要求を漏れなく記述するための方法が研究されてきた。例えば、機能要求の抜けや漏れをなくすための工夫 [Saito1] [Sakamaki] [Wakabayashi] や、非機能要求の記述を評

※1 岡山大学大学院自然科学研究科

※2 岡山大学工学部情報系学科

※3 みに先端研

価する枠組み [Saito2] が提案されている。

一方、そのようなRFPの記述の漏れや正確さとは別に、RFPには受注者にとって無理難題とも言えるリスクの高い要求が含まれている場合があり、やはりプロジェクト失敗の原因となり得る [Mitani]。例えば、「将来の機能拡張などにおけるデータ移行時に特別な費用が発生しないこと」あるいは、「同時処理件数の増加により、レスポンスに影響を与えないように考慮されていること」といった記述が挙げられる。ベンダが「できます」「やります」の精神でこのような記述を含む案件に応札すると、後で大きな問題が発生することとなる。

そこで、本論文では、地方公共団体や病院システムなどの5つのRFPを対象として、ベンダの視点から無理難題と考えられる要求を抽出・分類し、どのような無理難題が存在するかを明らかにすると共に、事例集としてまとめることで、ベンダのリスク管理に役立てることを目指す。また、無理難題を整理することは、ユーザ自身によるRFPのチェックにも役立つと期待される。

なお、本論文は、筆者らの研究会原稿 [Sumiyoshi] をベースに全面的に改訂したものであり、本論文では抽出したRFPをより適切なものへと再分類すると共に、従来研究との比較、及び、無理難題を書きあらためる記述例を追加し、論文としての体裁を整えている。

以降、2節では、RFPに含まれる無理難題の例とその問題点を述べる。3節では、無理難題の分類方法とその結果を述べる。4節では、従来のRFPの評価結果との比較を行う。5節は、無理難題を書きあらためる記述例を示す。最後に、6節では、まとめと今後の課題について述べる。

2 提案依頼書における無理難題

本論文におけるRFPの無理難題とは、多くのベンダにとって実現が困難な要求や、実現のために多大な費用負担を強いられる可能性のある要求を指す。そのような要求は、例え正確に、かつ漏れなく記述されていたとしても、開発そのものの失敗やベンダが多大な損害を受ける原因となるため、受注前にその存在を認識しておく必要がある [Mitani]。

無理難題の例として、「将来の機能拡張などにおけるデータ移行時に特別な費用が発生しないこと」や「法改正などの制度改正に伴うバージョンアップは、別途費用が発生しないこと」といった要求が挙げられる。これら要求では、将来どのような機能拡張や法改正が行われるか明らかでないにもかかわらず、

無償で作業を行うことを強要しており、ベンダが大きな金銭的損失を被るリスクがある。また、別の例として、「同時処理件数の増加により、レスポンスに影響を与えないように考慮されていること」が挙げられる。この要求では、同時処理の増加件数の具体的な数字が示されていないため、文字通り解釈すると、同時処理数が無限に増大した場合においてもレスポンスに影響を与えないことが求められ、その実現は不可能となる。

財務担当が完備され、適切にリスク管理が行われている企業においては、こうした要求条件では応札そのものできないと考えられる。しかし、実際にはリスク管理不在の企業が「何でもできます、やります」方式で応札する可能性があり、大きな問題となる。

従来、RFPに要求を漏れなく正確に記述するための方法が研究されてきたが、無理難題の評価や分類についての研究は、筆者の知る限り行われていない。

3 無理難題の分類

3.1 題材

本論文では、地方自治体や病院などの一般公開されている5つのRFPを題材とした(表1)。これらは、文献 [Saito] においても非機能要求の記述の評価に用いられており、その評価結果との比較が可能なことから採用した。表1に示されるように、これら5つのRFPのドメインは互いに異なっており、その規模も様々である。

表1 題材となるRFP

RFP	頁数
A. 市基幹システム構築事業 (自治体)	59
B. 町立図書館情報システム (図書館)	11
C. 独法救済給付業務システム (政府機関)	21
D. 大学キャンパス教育・研究用システム (大学)	71
E. 広域連合総合医療情報システム (病院)	160

3.2 手順

次の手順により、無理難題に該当すると思われる要求を抽出し、分類した。

1. 多くのベンダにとって実現が困難な要求や、実現のために多大な費用負担を強いられる可能性のある要求を抽出する。ただし、実現が困難であるかどうかや、費用負担を強いら

れる可能性があるかどうかは客観的な基準を設けることが難しいことから、本論文では、誤検出(擬陽性)を避けることを主眼において、明らかに実現が困難である、若しくは費用負担が発生する要件のみを抽出した。その結果、5つのRFPから合計74件の要求記述を抽出した。

2. 抽出した要求のうち、記述そのものの品質の問題(あいまいな記述、矛盾する記述など)を除外する。結果として、35件の無理難題が残された。
3. 残った要求の分類を行う。35件の無理難題を傾向が似ているものを同じグループとして分類した結果、5つの分類に集約された。
4. 実務者を交えた議論により、分類の見直しを行う。分類間の独立性や各無理難題がいずれの分類に属するかについて精査した結果、無理難題を4つに再分類した。

3.3 分類結果

得られた分類は次のU1～U4である。

(U1) 実績を求める要求

例として、「医療情報システムとして、XXX病院以上の規模・機能の病院において、相当数の安定稼働実績のあるソフトウェアであること」のように、導入するソフトウェアに対して、当該ドメインにおける実績を求める要件が挙げられる。この例では、大規模病院にシステム納入した実績のある企業でないと応札ができないため、多くの企業を実質的に入札から締め出していることとなる。ただし、一般に受注者の能力を判定する判断材料の一つとして実績を求める場合があり、必ずしも無理難題とならない場合があることから、このU1については、主観的な判断に左右されやすい点に注意する必要がある。

(U2) 技術的に実現が難しい要求

例として、2節で例示した「同時処理件数の増加により、レスポンスに影響を与えないよう考慮されていること」や、「今回導入する情報システムは、マルチベンダ環境での利用を保証すること」といった要求が挙げられる。後者については、「マルチベンダ環境」は無数に想定されるため、事実上実現は不可能となる。また、「ID、パスワードの漏洩が発生しない仕組みを用意する」といったセキュリティに関する要求も、あらゆる攻撃を想定して100%漏洩を防ぐことは不可能であることから、具体的な防御の範囲を指定していない場合はこのU2に含まれる。

(U3) 仕様の分からない既存のシステムの移行、連携を求める要求

例として、「現行図書館システムのデータを漏れなく、完全に新システムに移行すること」、「本業務において改修などを実施際には、既存システムのソフトウェア構成やシステム開発言語などを踏襲し拡張などにも対応できること」などが挙げられる。既存システムの詳細な仕様が別資料として与えられていない限り、妥当な見積もりが困難となる。このような場合、過去に開発を担当した企業でないと応札が困難であるため、実質的には多くの企業を入札から締め出していることとなる。なお、本論文では、5つのRFPすべてにおいて論文作成時点では別資料が確認できなかったため、既存のシステムの移行、連携を求める要求についてはすべてU3として列挙している。

(U4) 将来の課題への対応を求める要求

例として、「将来の機能拡張などにおけるデータ移行時に特別な費用が発生しないこと」、「法改正などの制度改正に伴うバージョンアップは、別途費用が発生しないこと」、「将来連合立病院になる5病院において連携できるシステムであること」などが挙げられる。これらの要求は、未定義の条件によって将来発生する作業を、費用が顕在化しない形で実施するように求めており、ベンダにとって将来大きな問題となる。また、別の例として、「納入成果物の構成、詳細については、受注後、XXXと協議し取り決めること」が挙げられる。この要求では、納品物の詳細が示されていないにもかかわらず、契約を先にすることを求めている点が問題である。

なお、本分類を決めるにあたって、3.2節の手順3.では、(U5)として「リスクを受注者に負わせる(費用負担を求める)要件」を設けていた。しかし、手順4.における再検討の結果、U5とU1～U4は両立する可能性がある(例えば、「仕様の分からない既存システムとの連携に関して費用負担を求める」要件はU3とU5の両方に属する)ことから、U5を削除し、U5に分類していた要件をU1～U4に再分類した。また、将来の課題U4についても、U1～U3と両立する可能性があるが、将来の課題に対する要件であることが明確な無理難題については、例えU1～U3と両立する場合においてもU4を優先して分類することとした。そのほかの、2つ以上の分類に属する可能性がある無理難題については、いずれが主であるかを(主観的にはあるが)検討し、U1～U4のいずれかに分類した。

本論文で分類した35件の無理難題を、事例集として付録に付ける。また、各RFPについて、U1～U4の件数を表2に示す。

表2 RFPに含まれる無理難題の件数

RFP	無理難題の種別				合計	頁当たり件数
	U1. 実績	U2. 技術	U3. 既存	U4. 将来		
A. 自治体	3	0	5	2	10	0.169
B. 図書館	1	0	2	0	3	0.273
C. 政府機関	0	1	2	2	5	0.238
D. 大学	0	4	0	1	5	0.070
E. 病院	3	4	1	4	12	0.075
合計	7	9	10	9	35	0.109

表3 「A.自治体」の無理難題の記述個所の分析

RFP中の項目	要件数	無理難題の種別					合計	要件当たりの無理難題(%)
		U1. 実績	U2. 技術	U3. 既存	U4. 将来			
提案範囲・スケジュール	15	0	0	0	1	1	6.67	
システム	302	2	0	4	1	7	2.32	
提案書作成	82	0	0	0	0	0	0	
見積書作成	68	1	0	1	0	2	2.94	
サービスレベル	78	0	0	0	0	0	0	
ハードウェア	395	0	0	0	0	0	0	
合計	940	3	0	5	2	10	1.06	

いずれのRFPにおいても、少なくとも3件の無理難題が含まれており、RFPに無理難題が含まれることは決してまれなことではないことが分かった。

最も件数の多い無理難題はU3の「仕様の分からない既存のシステムの移行、連携を求める要求」であった。U3は既存システムにかかわっていない企業を実質的に入札から締め出すことになるため問題である。また、そのような意図がユーザにないとしても、既存システムの仕様の重要性をあまり理解していない、若しくは伝えなくても大丈夫であるとユーザが判断していることになるため、開発を成功に導く上で大きな問題である。

また、U4の「将来の課題への対応を求める要求」も、4つのRFPに含まれていた。このU4は、システム発注にかかる費用を固定化したいというユーザの思いから出たものである可能性がある。将来の改修、保守に費用がかかることは当然であるという認識がユーザ側にないとしたら、大きな問題である。

表2の右端の列に、RFP1頁当たりの無理難題の件数を示す。分析対象としたRFPでは、頁当たりの無理難題の件数は0.070～0.273の範囲となった。およそ3.7～14頁に1件の無理難題を含んでいることとなる。「A.自治体」「B.図書館」「C.政府機関」は、「D.大学」「E.病院」と比べると高い値となっている。前者は後者に比べて頁数が少ないことから、頁数が少ないがむしろ無

理難題の密度が大きいと言える。受注者にとっては、RFPの頁数が少ないからといって安心できるものではなく、受注時のリスクが高い場合があると言える。

3.4 無理難題の記述個所の分析

表2の無理難題のうち、とくに頁当たり件数の多かった「A.自治体」「B.図書館」「C.政府機関」について、RFPのどの部分に記述されていたかを分析する。具体的には、各RFPの節立てに従って要件を分類し、各分類における無理難題の密度(要件当たりの無理難題の件数)を分析する。

「A.自治体」の分析結果を表3に示す。このRFPでは、要件の1.06%に無理難題が含まれていた。多くの無理難題は「システム要件」に含まれており、中でも4件の「U3.既存」が含まれていた。本システムは、既存システムの再構築となることから、既存システムの利用、引き継ぎ、データ移行に関する無理難題が生じたと考えられる。また、本システムは、外部団体とのデータ連携が必要なことから、外部団体における仕様の分からないシステムとの連携に関しての無理難題が生じたと考えられる。また、「システム要件」には「U1.実績」が2件含まれていたが、これらは、システムの構成要素となるソフトウェアなどに関しての実績を求める無理難題であった。一方、サービスレベルに関する要件

表4 「B.図書館」の無理難題の記述個所の分析

RFP中の項目	要件数	無理難題の種別					要件当たりの無理難題(%)
		U1. 実績	U2. 技術	U3. 既存	U4. 将来	合計	
ハードウェア	95	0	0	0	0	0	0
ソフトウェア	3	1	0	0	0	1	33.3
データ作成・管理・移行	16	0	0	1	0	1	6.25
運用	25	0	0	0	0	0	0
Webサイト作成	19	0	0	0	0	0	0
その他	6	0	0	1	0	1	16.7
合計	164	1	0	2	0	3	1.83

表5 「C.政府機関」の無理難題の詳細分析

RFP中の項目	要件数	無理難題の種別					要件当たりの無理難題(%)
		U1. 実績	U2. 技術	U3. 既存	U4. 将来	合計	
システム	18	0	0	0	1	1	5.56
性能・信頼性・拡張性・安全性	13	0	1	2	0	3	23.1
テスト	21	0	0	0	0	0	0
移行	5	0	0	0	1	1	20.0
運用・保守	5	0	0	0	0	0	0
ハードウェア	26	0	0	0	0	0	0
ソフトウェア	22	0	0	0	0	0	0
その他・特記事項	58	0	0	0	0	0	0
合計	168	1	0	2	2	5	2.98

表6 非機能要件の記述の評価点(文献[Saito2])と無理難題の件数

提案依頼書	各評価者による評価点(100点満点)			無理難題の件数
	エキスパート	教員	エンブラ系SE	
A. 自治体	2.71	5.99	6.88	10
B. 図書館	27.31	32.79	33.40	3
C. 政府機関	18.91	15.16	16.91	5
D. 大学	5.06	5.88	5.34	5
E. 病院	42.75	44.08	64.90	12

には無理難題が計上されていないが、個々のサービスのレベルについて具体的な値が挙げられていない項目が多く、現時点では無理難題とは判断できなかったことによるものである。4節でも示すように、本RFPではサービスレベルを含む非機能要件全般に関する記述が不十分であったことから、将来的なリスクとなる可能性がある。また、ハードウェア要件についても無理難題を含んでいないが、これは、PCやその周辺機器について、具体的な要件が明確に記載されており、無理難題の生じる余地がなかったためである。

「B.図書館」は、合計164件の要件を含んでいた(表4)。3件の

無理難題のうち、「データ作成・管理・移行」に関する要件に、現行システムからのデータ移行に関する1件が含まれていた。これは「A.自治体」と同様であり、データ移行は無理難題の温床となる可能性がある。ソフトウェア要件には、システムの構成要素となるソフトウェアについての過去の実績を求める無理難題が含まれていた。また、「その他」要件は、雑多な要件の寄せ集めであったが、既存のネットワーク・電源などの工事に関して、受託者に費用負担を求める無理難題が含まれていた。

「C.政府機関」は、合計168件の要件を含んでいた(表5)。「性能・信頼性・拡張性・安全性」のカテゴリに全5件のうちの3件の

無理難題が含まれていた。性能要件について「既存システムの要件に合わせる」とだけ記載されており、拡張要件についても「既存システムのソフトウェア構成を踏襲」することが記載されており、既存システムに関する説明がないために、無理難題となっていた。また、信頼性要件については、「稼働中システムの正常動作を保証し、通常業務に重大な影響を与えないこと」と記載されており、正常動作を100%保証することは現実的に困難であることから、無理難題となっていた。本RFPでは、「性能・信頼性・拡張性・安全性」に関して全体的に記述が不足していることから、3件もの無理難題が生じたと考えられる。

4 従来研究との比較

文献[Saito2]では、本論文と同じ題材を用い、RFPに含まれる非機能要件が漏れなく正確に記述されているかについての評価を行っている。本節では、文献[Saito2]の結果と本論文の結果を比較することで、要件が漏れなく記述されていることと、無理難題が含まれることの関係について論じる。

表6に、文献[Saito2]における非機能要件の記述の評価点(100点満点)と、無理難題の件数を示す。表に示されるように、最も評価点の高いE.病院が、最も無理難題の件数が多いという結果となっている。その一方で、評価点の低いA.自治体も、無理難題が10件と多いことから、要求が漏れなく正確に記述されていることと、無理難題が含まれることは独立であると言える。無理難題の密度に着目した場合も同様であり、B.図書館は2番目に評価点が高いが、表2の右端の欄に示されるように、無理難題の密度は最も高い。これらの結果より、しっかりと書かれた(あるいは書かれているように見える)RFPであっても、無理難題が含まれている可能性があることから、ベンダはその見極めを確実に行うべきであると言える。

5 無理難題の記述の改善に向けて

RFPに無理難題が含まれる原因の一つとして、RFPの形式がフリーフォーマットかつ自然言語で記述されていることが挙げられる。また、ユーザがソフトウェア開発に熟知していないことにより、妥当な記述が行えていないことが挙げられる。

将来的には、無理難題を記述しないためのガイドライン、あるいは、無理難題とならないような記述のテンプレートを策定することが望ましいと言える。本論文では、その第一歩として、

無理難題を書きあらためる記述例を示す。

(U1) 実績を求める要求

記述例「…〇〇の実績を有すること。〇〇に関する直接的な実績がない場合には、類似する実績や技術についてのエビデンスにより、〇〇の実施に支障がないことを示すこと。」

(U2) 技術的に実現が難しい要求

稼働率や稼働品質(応答時間、スループット、ターンアラウンド時間など)に関する要件については、具体的な数値や範囲を示す。

セキュリティに関する要件については、完全なセキュリティの実現は難しいため、最低限実現すべき要件を具体的に記述する、若しくは、順守すべきセキュリティ基準を明記すると共に、セキュリティが破られた際の対応に関する要件を記述する。

(U3) 仕様の分からない既存のシステムの移行、連携を求める要求

ユーザは既存システムの仕様を提供すること。

(U4) 将来の課題への対応を求める要求

記述例「納入後のシステムの保守・運用については、別途保守契約を結ぶこととする。また、新たな要求によるシステムの変更、改良については、別途委託契約を締結し、それに基づいて実施するものとする。」

また、「システム改修などを施しても、改修など機能を含めて全機能が使用できること」といった要求は、「システム改修時の回帰テストに必要なドキュメントとツールを提供すること」といった要求に変更する。

更に、U1～U4のいずれにおいても、現時点で不明確な要求についてのリスクや費用を受注者に負わせる代わりに、次のような文言を明記することが望ましい。「本システムの実現に際して第三者との協議や契約が必要となることが想定される場合には、その時期や費用見積もり、費用負担も含めてユーザとベンダが別途協議し解決する。」

6 まとめ

本論文では、主にベンダの立場から受注時のリスク評価に役立てることを目的として、RFPに含まれる無理難題を抽出し、その分類を試みた。地方自治体や病院などの5つの提案依頼書を

分析した結果、無理難題を4つに分類し、事例集として付録に示した。また、結果を従来研究と比較し、要求が漏れなく正確に記述されていることと、無理難題が含まれることは独立であることを示した。最後に、無理難題を書きあらためる記述の例を示した。

なお、本論文は、無理難題の抽出、及び、分類は筆者らの主観に基づいており、厳密な定義に基づく分類は行っていない。ただし、厳密な定義に基づく分析でなくとも、無理難題の具体例や類型を示すことができれば、ベンダ、ユーザの役に立つと期待される。今後は、より多くのRFPを分析して無理難題の分類を洗練させていくと共に、事例集をより充実させていくことが望ましい。また、本論文の5つの事例は、そのすべてが公的機関、あるいはそれに準じる組織のものであり、今後、民間企業のRFPについても分析することが課題となる。また、無理難題の記述を避けるためのガイドラインについても検討していくことが今後の課題となる。

付録 無理難題の事例集

(U1) 実績を求める要求

A. 市基幹系システム構築事業

- XXXシステムについては、運用実績があり現在稼働しているもので提案すること。
- 安定した動作実績があるものを使用すること。
- SEの要件は、別記定めるXXXを実現する上で運用上支障がない体制を確保することとし、十分な自治体経験を有するSEを配置すること。

B. 町立図書館情報システム

- XXX専用パッケージとして実績を有するもので構築すること。

E. 広域連合総合医療情報システム

- 受注者として、相当数の導入実績と運用保守実績のあるソフトウェアであること。
- 医療情報システムとして、安定稼働実績のあるハードウェアであること。
- 医療情報システムとして、XXX病院以上の規模・機能の病院において、相当数の安定稼働実績のあるソフトウェアであること。

(U2) 技術的に実現が難しい要求

C. 独法救済給付業務システム

- 稼働中システムの正常動作を保証し、通常業務に重大な影響を与えないこと。

D. 大学キャンパス教育・研究用システム

- 一般に有用性が認められているソフトウェアが利用できなければならない。また、国際的に流通しているフリーウェア（無償、有償を問わず）が利用できなければならない。
- 画像情報などの転送が遅滞なきようネットワークの安定化、高速化を図ること。
- ID、パスワードの漏洩が発生しない仕組みを用意する。
- サーバにあるホームページコンテンツやユーザファイルなどが消去、改ざんされない。

E. 広域連合総合医療情報システム

- 今回導入する情報システムは、マルチベンダ環境での利用を保証すること。
- 同時処理件数の増加により、レスポンスに影響を与えないよう考慮されていること。
- 稼働年数の経過などによるデータ量の増加に伴って、レスポンスに影響が出ないように考慮されていること。
- 医事会計システムにおいて、診療データ以外はすべて永久保存データとして管理できること。

(U3) 仕様の分からない既存のシステムの移行、連携を求める要求

A. 市基幹系システム構築事業

- 業務に必要な外部団体・金融機関などとのデータ連携機能を有すること。
- 既存システムにおけるデータ形式を考慮した上で、移行が容易であること。
- 新システムのテスト実施に際し、現行システムの利用が必要となる場合はその費用を負担すること（既存業者と直接契約すること）。
- 新システムの構築業者に選定された業者で、現行システムの改修が困難な場合は既存業者からの見積もりに基づき、その費用を負担すること（必須）。
- 現在、XXXで受け付けているすべての業務（現行システム、情報系システム（グループウェア、セキュリティシステムなど））

の業務システムのサポートにあたり必要な引き継ぎ費用を負担すること

B. 町立図書館情報システム

- 館内のネットワーク、電源などの工事については図書館と協議の上、必要であれば受託者の責任において実施すること。
- 現行図書館システムのデータを漏れなく、完全に新図書館システムに移行すること。

C. 独法救済給付業務システム

- 既存システムの要件に合わせること。
- 本業務において改修などを施す際には、既存システムのソフトウェア構成やシステム開発言語などを踏襲し拡張などにも対応できること。

E. 広域連合総合医療情報システム

- 既存システムに蓄積された必要なデータを安全かつ確実に移行できること。

(U4) 将来の課題への対応を求める要求

A. 市基幹系システム構築事業

- 将来における機能の拡張、機器の増設などに柔軟に対応できるように、原則として広く利用されている国際的な標準に基づく技術を採用し、汎用性、拡張性のあるシステムとすること。

- 提案するシステムは、後期高齢者制度対応などの各種法改正に対応していること。

C. 独法救済給付業務システム

- 納入成果物の構成、詳細については、受注後、XXXと協議し取り決めること。
- システム改修などを施しても、改修など機能を含めて全機能が使用できること。

D. 大学キャンパス教育・研究用システム

- 将来性、運用性、保守性や上記要件を考慮した上で必要と思われる機器、資材、工事などに関しては見積りの範囲とする。

E. 広域連合総合医療情報システム

- 将来の機能拡張などにおけるデータ移行などに特別な費用が発生しないこと。
- 法改正などの制度改正に伴うバージョンアップは、別途費用が発生しないこと。
- 将来連合立病院になる5病院において連携できるシステムであること。
- データベースサーバ、アプリケーションサーバのOSは、オープン環境下のスタンダードなものを使用すること。また、開発途中で陳腐化することがないように十分な実績があり、かつ将来においてもその発展が見込まれるものであること。

【参考文献】

- [Mitani] 神谷芳樹, "RFPで垣間見たソフトウェア・エンジニアリングの現実", IT記者会レポート, June.2014., <http://itkisyakai.com/detail.php?id=330&pid=7>
- [Port] D. Port, A. Nikora, J. H. Hayes, L. G. Huang, "Text Mining Support for Software Requirements: Traceability Assurance", Proc. 44th Hawaii International Conference on System Sciences, pp.1-11, 2012.
- [Roth] Bud Porter-Roth著, 渡部洋子訳, "RFP入門 一初めての提案依頼書", 日経BP, 2004.
- [Saito1] 斎藤篤史, 伊藤恵, "BABOKを活用したチェックシートによるRFP評価の試み", 日本ソフトウェア科学会第31回大会(2014年度)講演論文集, 2014.
- [Saito2] 齊藤康廣, 門田暁人, 松本健一, "非機能要件に着目したRequest For Proposal (RFP) の評価", SEC journal, Vol.10, No.3, pp.30-37, Sep. 2014.
- [Sakamaki] 酒巻弘晃, 横山真一郎, 劉功義, 石井信明, 田村智幸, 牛嶋一朗, 加藤俊, 笠井直貴, 木野泰伸, "ステークホルダーの識別とRFPの評価方法に関する研究", プロジェクトマネジメント学会誌, Vol.13, No.3, pp.26-31, June 2011.
- [Sumiyoshi] 住吉倫明, 門田暁人, "ソフトウェア開発の提案依頼書における無理難題の分析", 電子情報通信学会技術報告, ソフトウェアサイエンス研究会, No. SS2016-52, pp. 95-100, Jan. 2017.
- [Wakabayashi] 若林沙弥, 伊藤恵, "開発要求の精度向上のための提案依頼書の作成支援", 日本ソフトウェア科学会第32回大会講演論文集, 2015.

「ソフトウェア開発の定量的管理」 特集にあたって

東洋大学 経営学部 経営学科 教授 野中 誠

SECシステムグループリーダー 山下 博之

ソフトウェア開発のプロセスやプロダクトの状況を見える化することで、品質や開発効率低下の予兆を検知して対策を講じたり、技法や手順、組織体制の改善に取り組んだりすることが期待される。本特集では、ソフトウェア開発の定量的管理に関する産業界の最近の取り組み事例を中心に、IPA/SECの取り組みや学術研究の事例を紹介する。

品質の高いソフトウェアやシステムを効率良く開発するためには、適時・適切な管理が必要となる。具体的には、開発の経過や結果を見える化することにより、品質低下や効率の兆候をいち早く検知し速やかに対策を講ずることができる。あるいは、組織の体制や手続きなどの問題点を発見し改善することができる。

ソフトウェア開発データの測定と分析に基づく定量的管理は、経過を対象とするインプロセス計測と、結果を対象とするベンチマーキングに大別できる。前者は進行中プロジェクトの把握と改善を図ることを主眼とし、分析結果を当該プロジェクトの進捗管理やリスク管理などに用いる。後者は品質マネジメントなどの改善を図ることを主眼とし、分析結果は組織の開発プロセス標準や管理基準などに反映する。それぞれ、以前から各所で取り組みが行われ、成功事例も数多く報告されている。一方、ICTの進展は目覚ましく、ソフトウェアそのものが高度化・複雑化し続けている。また、ソフトウェア開発の手法やその支援環境も刻々と進化している。最近では、ビッグデータ解析やAIなどの関連技術も導入されつつある。

このような背景から企画した本特集では、ソフトウェア開発の定量的管理に関し、IPA/SECにおける推進のための継続的な取り組み内容と、産業界及び学界における先進的な取り組み事例を紹介する。

本特集では、次の記事を掲載している。

IPA/SECによる「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(エンタプライズ系)」では、10年以上にわたって蓄積している開発データを用いるベンチマーキング、「ソフトウェア開発データ白書」の取り組みを説明すると共に、蓄積データの分析により得られた新たな知見などを紹介している。同じくIPA/SECによる「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」では、エンタプライズ系に比べて

遅れていた組込みソフトウェア開発の定量的管理推進のために開始した、ベンチマーキングにより得られた分析結果を紹介している。これらの公的ベンチマークに蓄積された統計分析結果はソフトウェア開発の各局面で“参考値”として利用できる。また、ベンチマーキングのやり方そのものも、各組織におけるデータ収集や分析を行う上で参考となるものである。

日本電気株式会社の宮崎らによる「NECにおける定量的管理～ソフトウェア品質会計とNECデータ白書に基づく改善活動の展開～」では、独自の“ソフトウェア品質会計”技法に基づく取り組みの概要を説明すると共に、社内データ白書の分析から得られた知見などを紹介している。東北インフォメーション・システムズ株式会社の佐藤による「SPI活動と連携した定量的プロジェクトマネジメントの実践」は、各部門から収集した開発データの解析結果をフィードバックし、ソフトウェアプロセス改善に役立ててもらうことにより、定量的管理を全社的に定着させた事例である。興味深いのは、これら2つの成功事例は、いずれも開発データの分析結果に基づくプロセス改善活動に全社的に取り組んでいる点である。読者には大いに参考になると思われる。

東洋大学の野中らによる「ソフトウェア定量的管理にかかわる学術研究事例」では、産業界におけるソフトウェア定量的管理にとって参考となり、役立つことが期待される学術研究の事例について、筆者である野中・亀井・大平がそれぞれかかわった最近の研究事例と知見を紹介し、産業界でのソフトウェア定量的管理に期待することを述べる。

読者の方々には、本特集で紹介した事例などを参考に、各社・組織のソフトウェア開発事情を考慮した上で、それぞれのソフトウェア開発を更に進化させるための定量的管理を進めていただけることを期待する。

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(エンタプライズ系)

SEC研究員 峯尾 正美 SEC専門委員 佐伯 正夫 SECシステムグループリーダー 山下 博之

SEC設立以来、定量的に管理されたソフトウェア開発データを業界から広く収集・分析し、ソフトウェアの信頼性・生産性向上のための統計データを「ソフトウェア開発データ白書」として公開している。その最新版である「ソフトウェア開発データ白書2016-2017」では、4,000プロジェクト以上のデータに基づく統計情報のほかに、生産性、信頼性の変動要因の多面的な分析結果や、代表的な3業種について業種ごとの統計情報を掲載した業種編を公開した。また、ソフトウェアの定量的プロジェクト管理を推進するため、ベンチマーク情報を品質マネジメントなどに活用するための手引きと事例集、上流工程の強化が信頼性向上につながることを定量的に示した分析結果を公開した。

1 ソフトウェアの定量的プロジェクト管理概要

1.1 プロジェクト管理における動的管理、静的管理

ソフトウェアの定量的プロジェクト管理とは、ソフトウェア開発プロセスにおいて作成される成果物や作業量を測定することで得られる種々の定量データを用いたソフトウェア開発管理技法の一つであり、「動的管理」と「静的管理」に大別される。

一般的に「プロジェクト管理」としては、図1に示すPDCAサイクルが知られている。この中で、ソフトウェアの「動的管理」とは、プロジェクト実施中(In Process)に得られる種々の定量的・定性的情報をもとに、プロジェクトの進行を管理していく手法であり、ソフトウェア開発においては、計画値と実績値の比較による進捗管理や、テスト工程中の発生不良からソフトウェアの品質を予測し、必要な対策を講ずる「ソフトウェア品質予測」などの様々な技法がある。一方「静的管理」とは、完了したプロジェクト(Post Process)を分析し、得られる種々の情報をもとに、当該プロジェクトの問題点を分析し、今後のプロジェクト推進上の改善点を検討したり、次のプロジェクトの計画策定時の妥当性の確認などを実施したりするものである。

このような「静的管理」の基本となる考え方が、ソフトウェア開発データ白書などを用いた「ベンチマーキング」である。

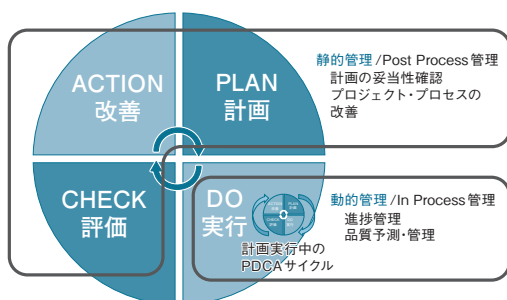


図1 PDCAサイクルと動的管理・静的管理

1.2 ベンチマーキング

ベンチマーキングとは、「良い成績を収めているプロジェクト群と比較し、それらのやり方(開発プロセス、マネジメント・プロセス、組織の特性など)を参考にして、自組織の業務改善及び組織の改善を進めること」である。図2にソフトウェア開発におけるベンチマーキングの位置付けを示す。

完了したプロジェクトを分析して得られた各種情報をリポジトリ(データベース)に蓄積したものを「ベンチマーク」と呼ぶ。

ベンチマークは、自組織のプロジェクトデータを蓄積したものを使用するのが基本であるが、それらが整備されていない場合や、他組織の値を参考値とする場合は、外部のベンチマークを用いる。

これらのベンチマークを用いて、

- ①プロジェクト計画の妥当性評価
- ②プロジェクト・マネジメントの改善検討
- ③組織のマネジメント改善検討など

が可能となる。

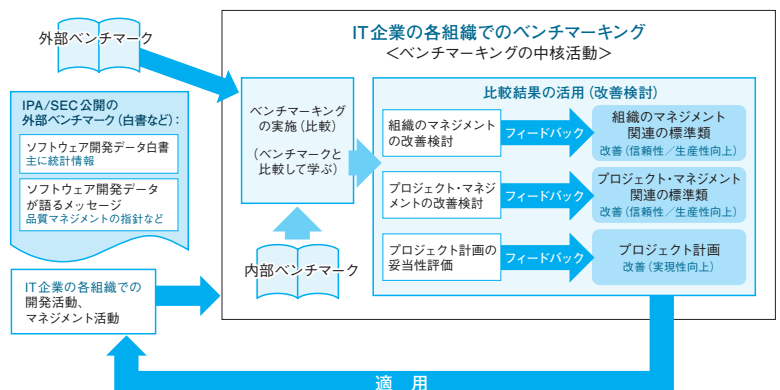


図2 ソフトウェア開発におけるベンチマーキングの位置付け

2

「ソフトウェア開発データ白書 2016-2017」の概要

2.1 概要

IPA/SECでは、ソフトウェア開発における定量的プロジェクト管理の普及促進の一環として、国内約30社からエンタープライズ系のソフトウェア開発のプロジェクトデータを収集し、整理・分析した「ソフトウェア開発データ白書」(以下データ白書)を定期的に発行している。

2005年に最初の「ソフトウェア開発データ白書2005」を発行し、9冊目にあたる最新版「ソフトウェア開発データ白書2016-2017」(図3)を2016年10月1日に発行した。なお、収集されたプロジェクトデータは、累計で、4,067件となった。

データ白書の目次は、以下の通りである。

- 1章 背景と本書の目的
- 2章 収集データについて
- 3章 分析について
- 4章 収集データのプロフィール
- 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 工程別の分析
- 8章 生産性の分析
- 9章 信頼性の分析
- 10章 その他の分析



図3 データ白書 2016-2017

2.2 収集しているデータ項目

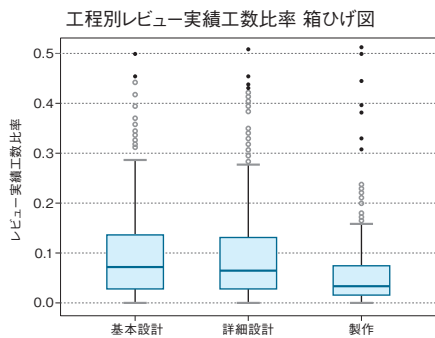
データ白書では、工数・工期・規模など定量的な情報のみならず、システム特性(対象業種・業務など)、ユーザ要件管理(要求仕様の明確さ、要求レベルなど)など定性的な情報も併せて収集している。これによって3節で示すような多面的な分析を可能としており、これは、他に類を見ないIPA/SECデータ白書の強みとなっている。

2.3 統計データの掲載例

①工程別分析例

データ白書での統計データ掲載例として、工程別の分析例(レビュー実績工数比)を図4に示す。これらの統計値は、データの特徴に合わせて、グラフ、散布図や箱ひげ図と共に、基本統計量が掲載されている。

これらの値は、例えばレビュー時間設定時の参考値(ベンチマーク)として使用することができる。



工程別レビュー実績工数比率の基本統計量 [比率]

レビュー実績工数比率	N	最小	P25	中央	P75	最大	平均	標準偏差
基本設計	543	0.002	0.030	0.074	0.136	0.692	0.104	0.111
詳細設計	505	0.002	0.032	0.067	0.130	0.625	0.097	0.094
製作	369	0.000	0.017	0.035	0.076	0.836	0.070	0.112

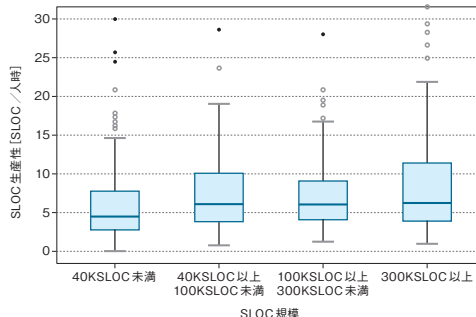
図4 データ白書での統計データ掲載例

②生産性の掲載例

ソフトウェア生産性の掲載例として、図5に開発規模別のSLOC※1生産性を示す。

SLOC生産性(SLOC/人時)の中央値は、5.4SLOC/人時である。また、SLOC生産性においては、規模による大きな差は見られない。

SLOC規模別SLOC生産性(新規開発、主開発言語グループ) 箱ひげ図



SLOC規模別SLOC生産性の基本統計量(新規開発、主開発言語グループ)

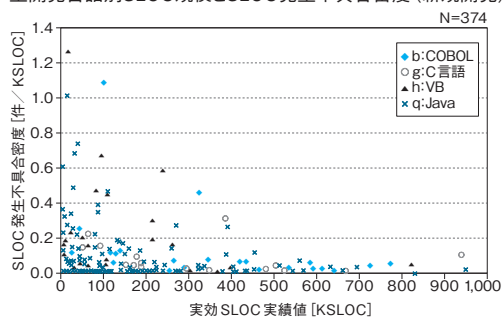
SLOC規模	単位	N	最小	P25	中央	P75	最大	平均	標準偏差
全体		635	0.0	3.2	5.4	8.9	325.2	8.5	16.6
40KSLOC未満		206	0.0	2.6	4.2	7.6	44.7	5.9	6.0
40KSLOC以上100KSLOC未満	SLOC/人時	139	0.7	3.6	5.9	9.8	71.7	8.3	8.9
100KSLOC以上300KSLOC未満		149	1.1	3.8	5.9	9.0	41.6	7.6	6.5
300KSLOC以上		141	0.9	3.6	6.1	11.4	325.2	13.6	32.3

図5 開発規模別のSLOC生産性

③信頼性の掲載例

ソフトウェア信頼性の掲載例として、図6にSLOC規模とSLOC発生不具合数の散布図を示す。

主開発言語別SLOC規模とSLOC発生不具合密度(新規開発)



	N	最小	P25	中央	P75	最大	平均	標準偏差
全体	374	0.000	0.000	0.012	0.062	2.413	0.072	0.191
40KSLOC未満	116	0.000	0.000	0.000	0.059	1.269	0.075	0.185
40KSLOC以上100KSLOC未満	86	0.000	0.000	0.017	0.070	0.741	0.073	0.134
100KSLOC以上300KSLOC未満	88	0.000	0.000	0.018	0.078	2.413	0.099	0.292
300KSLOC以上	84	0.000	0.003	0.015	0.038	0.476	0.040	0.082

図6 SLOC規模とSLOC発生不具合数

なお、データ白書では、「信頼性」は、システム稼働後の「発生不具合密度」を指標として用いている。

3 生産性・信頼性変動要因分析

データ白書2016-2017では、新たな試みとして、生産性や信頼性に関係すると思われる変動要因(生産性:22種類、信頼性:23種類)をリストアップし、生産性・信頼性との相関関係を多面的に分析した。

3.1 生産性変動要因分析

生産性変動要因分析の結果を、表1に示す。

表1 生産性変動要因分析結果

通番	区分	*	変動要因候補	通番	区分	*	変動要因候補
1	業種	◎	業種	14	開発プロセス	◎↓	設計文書化密度
2	QCD要求	×	信頼性の要求レベル	15		△↓	設計レビュー工数密度
3		×	性能・効率性の要求レベル	16		◎↓	設計レビュー指摘密度
4		◎↓	重要インフラタイプ	17		◎↓	テスト密度
5	実現手段	×	アーキテクチャ	18		◎↓	テスト検出不具合密度
6		○	主開発言語	19		×	上流工程での不具合検出比率
7	実現手段	◎	プラットフォーム	20	ユーザ要求管理	×	要求仕様の明確さ
8		×	開発フレームワークの利用	21	×	ユーザ担当者の要求仕様関与	
9	実施体制	◎↓	月当たりの要員数	22	組織の成熟度	×	定量的な出荷品質基準の有無
10		×	外部委託比率				
11		×	PMスキル				
12	実施体制	×	テストスキル				
13		◎	品質保証体制				

*: Welchのt検定結果 (P値) ◎:1%有意、○:5%有意、△:10%有意、×:有意でない、↑:正の相関、↓:負の相関

分析方法としては、基本的に、変動要因ごとにグループ化し、それぞれの生産性の差を、Welchのt検定にて検定した。なお、本表において、「×」となっているものは、一般的に相関がないというわけではない。分析対象のSEC蓄積データにおいては、統計的に見て今回設定した基準に照らして有意性が低い、ということを示していることに留意されたい。

分析結果の例を図7に示す。

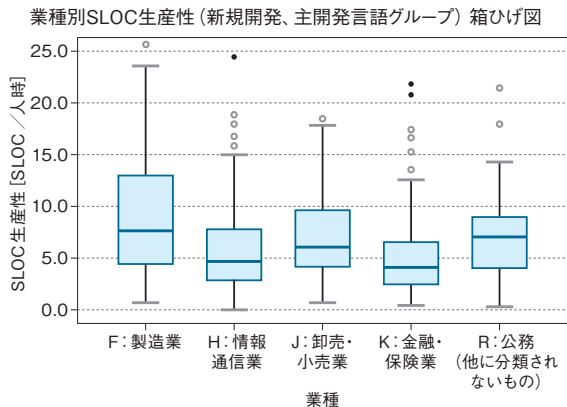


図7 生産性変動要因分析：業種

業種ごとの生産性の分析の結果、「製造業はほかよりSLOC生産性が高い」「金融・保険業はほかよりSLOC生産性が低い」ことが判明した。

これらは、業種ごとにQCD要求(信頼性の要求レベルなど)やユーザ要求管理に差があり、その結果、開発プロセス(とくに品質保証プロセス)に差が生じ、生産性の差として現れたものと想定されるが、別途分析は必要である。

なお、開発プロセスでの設計文書化密度(頁/KSLOC)、テスト密度(テスト項目数/KSLOC)などの相関は、「負の相関」であり、例えば、設計文書化密度が「高い」ほど、生産性は「低い」ことを示している。

3.2 信頼性変動要因分析

信頼性変動要因分析の結果を表2に示す。

分析方法は、生産性の変動要因分析と同様である。

この分析の結果、業種、テストスキル、設計レビュー工数密度、上流工程での不具合検出比率などが信頼性と相関のある変動要因であることが明らかになった。

表2 信頼性変動要因分析結果

通番	区分	*	変動要因候補	通番	区分	*	変動要因候補
14	開発プロセス	×	設計文書化密度	14	開発プロセス	×	設計文書化密度
15		◎↑	設計レビュー工数密度	15		◎↑	設計レビュー工数密度
16		×	設計レビュー指摘密度	16		×	設計レビュー指摘密度
17		×	テスト密度	17		×	テスト密度
18		×	テスト検出不具合密度	18		×	テスト検出不具合密度
19		◎↑	上流工程での不具合検出比率	19		◎↑	上流工程での不具合検出比率
20	ユーザ要求管理	×	要求仕様の明確さ	20	ユーザ要求管理	×	要求仕様の明確さ
21		×	ユーザ担当者の要求仕様関与	21	ユーザ要求管理	×	ユーザ担当者の要求仕様関与
22	組織の成熟度	×	定量的な出荷品質基準の有無	22	組織の成熟度	×	定量的な出荷品質基準の有無
23		◎↓	テスト検出能率	23	組織の成熟度	×	定量的な出荷品質基準の有無

*: Welchのt検定結果 (P値) ◎:1%有意、○:5%有意、△:10%有意、×:有意でない、↑:正の相関、↓:負の相関

分析結果の例として、「上流工程での不具合検出率」の例を図8に示す。

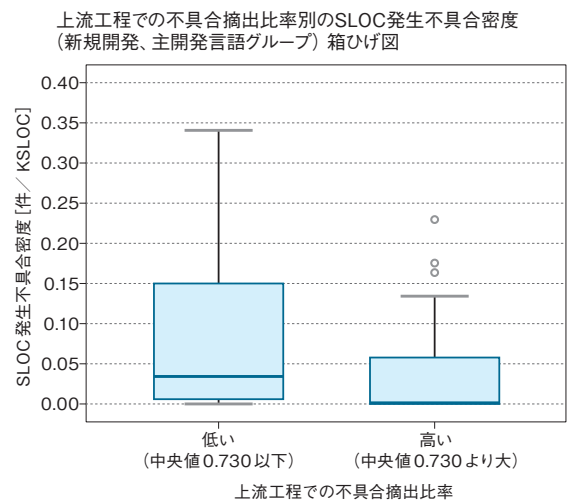


図8 信頼性変動要因分析：上流工程での不具合検出率

ここで、「上流工程での不具合検出率」とは、上流工程(基本設計～製作)でのレビュー指摘件数を、開発工程全体(基本設計～総合テスト(ベンダ確認))でのレビュー指摘件数とテスト工程でのバグ数の合計で割ったものである。つまり、上流工程で、どれだけ不良が検出できているかを示す。

図8で示すように、上流工程での不良検出率の中央値(0.73)で全体を2分すると、検出率が高いグループのほうが、SLOC発生不具合密度が低い(信頼性が高い)ことが判明した。

4 業種ごとの分析

データ白書2016-2017では、新たな試みとして、ベンチマークとして使用されることの多い第6章「工数、工期、規模の関係の分析」から第9章「信頼性の分析」までを、データ件数の多い3業種「金融・保険業」「情報通信業」「製造業」のデータに限定して分析した「業種編」を分冊として発行した(図9)。

データ件数は、本編(業種混在)が4,067件であるのに対して、それぞれ、「金融・保険業」(1,151件)、「情報通信業」(649件)、「製造業」(641件)である。

業種編作成の効果として、業種に細分することで、より自部門・

自プロジェクトと近い条件でのベンチマークの参照が可能になることが挙げられる。



金融・保険業編

情報通信業編

製造業編

図9 業種編

5 「統計指標に基づく品質マネジメント実践集」

データ白書や個々の企業が持つ内部ベンチマーク情報を活用した「定量的管理による信頼性向上のヒントや具体的な改善事例集、または品質マネジメントのための具体的なベンチマーキング方法の手引き」として、「統計指標に基づく品質マネジメント実践集」(以下「本書」)を作成し、公開した。

本書の特長は、以下の通りである。

- ①統計指標に基づいた品質マネジメントの代表的なシーン(15シーン)に沿って具体的なベンチマーキング方法を解説
- ②IT企業における具体例(13事例)を含む、豊富なベンチマーキングの具体例(32事例)を掲載
- ③プロジェクト・マネジメント／組織改善につなげていく改善を重視した具体的なベンチマーキング方法を掲載

データ白書と併せて本書を用いることで、ベンチマーキングを実施する際の参考となれば幸いである。

6 「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』」

「ソフトウェア開発データが語るメッセージ」(以下「メッセージ」)は、ソフトウェア開発の定量的管理の主な目的である「プロジェクト計画の妥当性評価」や「組織の品質マネジメントの改善」に向けて、最新の白書に掲載したデータを分析し、そこから導いたプロジェクトの評価や改善の指針となる考察結果をまとめたものである。

本メッセージでは、データ白書で分析した信頼性変動要因の結果を更に深掘りし、上流工程(本書では、要件定義、基本設計、詳細設計、製作工程)の強化が信頼性向上のために重要であることを初めて「定量的」に示した。

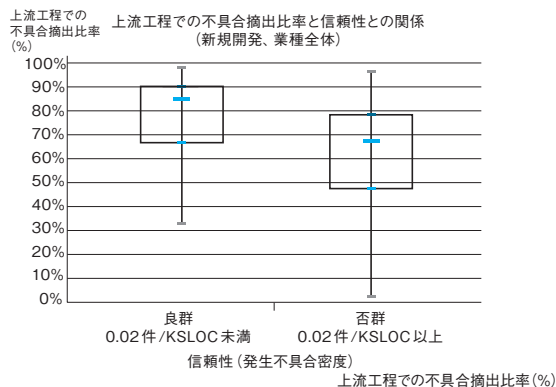
主なポイントは以下の通りである。

- 上流工程(基本設計～製作)での不具合摘出比率を高めることによって、信頼性向上が期待できる。

上流工程での不具合摘出比率(中央値)は、信頼性が高いグループでは約85%だったのに対し、低いグループでは約66%

であった(図10)。このことから、以下のメッセージを導き出した。

「プロジェクト計画／再計画や品質マネジメント改善等のシーンにおいて、上流工程での不具合摘出比率の目標を、目安として85%程度に高めて設定することを目指そう」



発生不具合密度	N	最小	P25	中央	P75	最大	平均	標準偏差	上流工程での不具合摘出比率(%)	
									信頼性(発生不具合密度)	上流工程での不具合摘出比率(%)
良群 0.02件/KSLOC未満	43	0.0%	65.9%	84.6%	89.4%	97.8%	73.4%	25.5%		
否群 0.02件/KSLOC以上	43	1.7%	46.7%	66.3%	78.4%	95.3%	59.3%	26.8%		

図10 上流工程での不具合摘出比率と信頼性

7 おわりに

ソフトウェア定量的プロジェクト管理の一環として、データ白書並びに関連する活動、成果物を紹介した。これらの成果物の効用としては、以下の3点と考える。

- ①統計分析結果を参考値(ベンチマーク)として利用
- ②データ収集、分析方法を参考にする
- ③分析結果(メッセージ)をプロセス改善のために利用

これらの成果物が、ソフトウェアの生産性・信頼性の向上に少しでも貢献できれば幸いである。

知るために測り 変わるために知り 先んずるために変わる

Capers Jones

※1 SLOC : Source Lines Of Code

参考文献

- [1] Capers Jones : ソフトウェア開発の定量的手法(第3版)、共立出版、1993
- [2] IPA/SEC : 定量的品質予測のススメ、オーム社、2008、<https://www.ipa.go.jp/sec/publish/tn08-004.html>
- [3] IPA/SEC : 続定量的品質予測のススメ、オーム社、2011、<https://www.ipa.go.jp/sec/publish/tn10-004.html>
- [4] IPA/SEC : ソフトウェア開発データ白書2016-2017、IPA/SEC、2016、<https://www.ipa.go.jp/sec/publish/tn12-002.html>
- [5] IPA/SEC : 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」、IPA/SEC、2017、<https://www.ipa.go.jp/sec/reports/20170331.html>
- [6] IPA/SEC : 統計指標に基づく品質マネジメント実践集、IPA/SEC、2016、<https://www.ipa.go.jp/sec/reports/20160701.html>

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

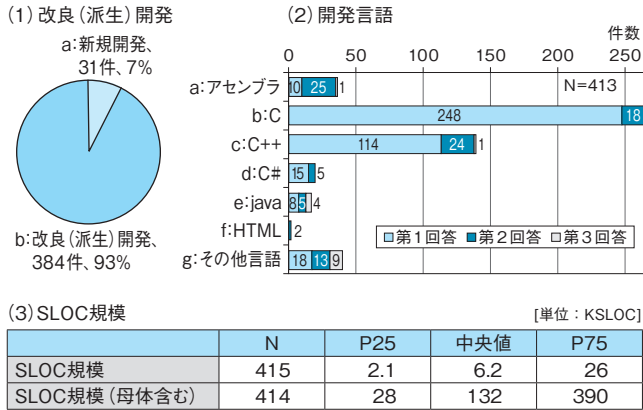


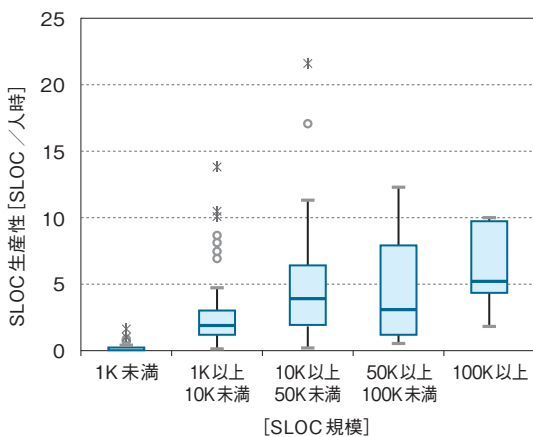
図3 分析対象の範囲

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。



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

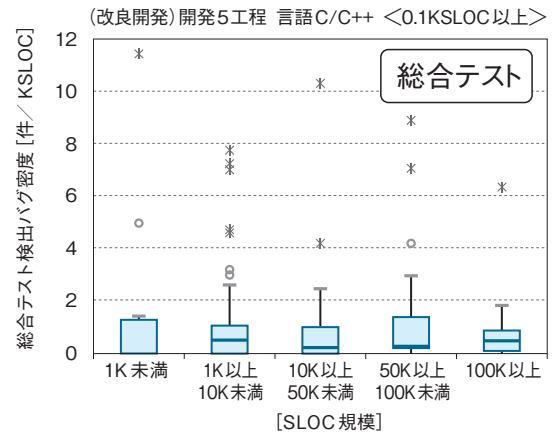
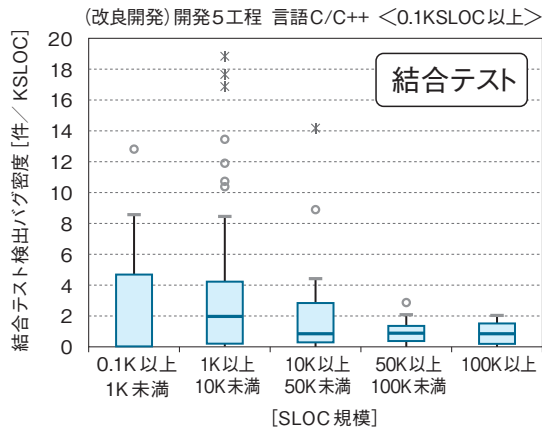
組み込みソフトウェア開発全般の生産性や信頼性の傾向が分かると、次はメトリクスの分析で先行しているエンタプライズ分野と比較して、組み込みソフトウェア開発の特徴を捉える。エンタプライズ分野の「ソフトウェア開発データ白書2016-2017」分析結果と比較して特徴の違いが見えたものを、幾つか紹介する。

(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

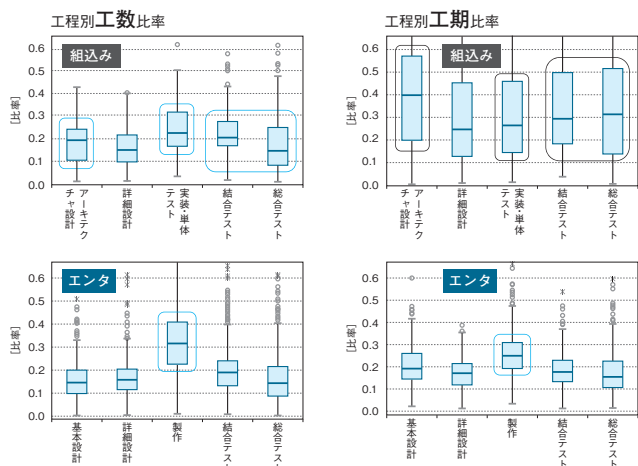


図6 工程別工数・工期の比率

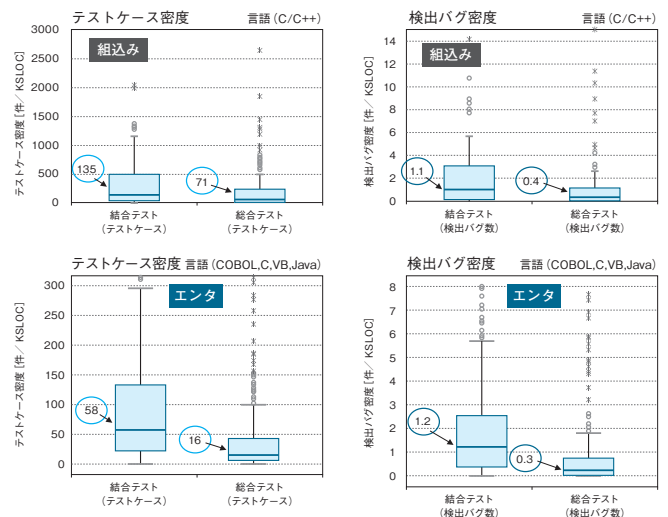


図7 テストケース密度と検出バグ密度

(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。

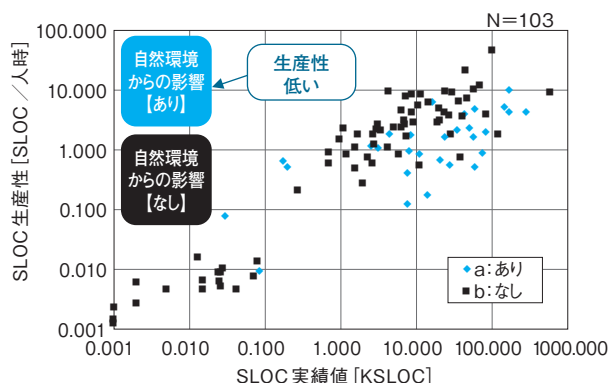
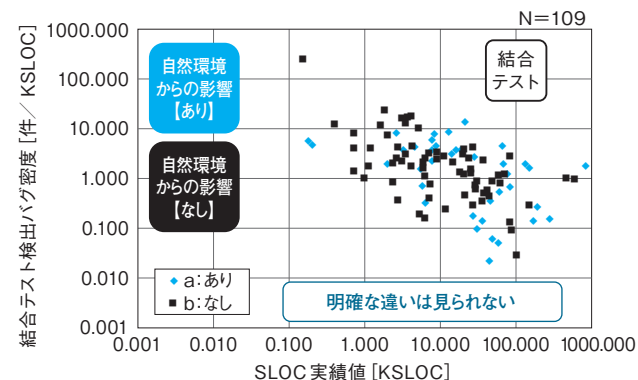


図8 自然環境からの影響を受ける場合のSLOC生産性



(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、総合テストにかか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ていて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。

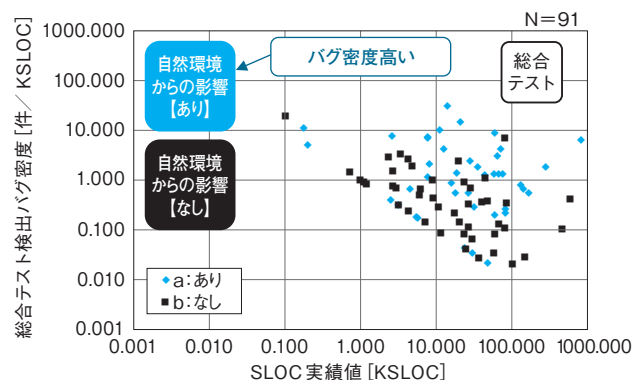


図9 自然環境からの影響を受ける場合のテスト検出バグ密度

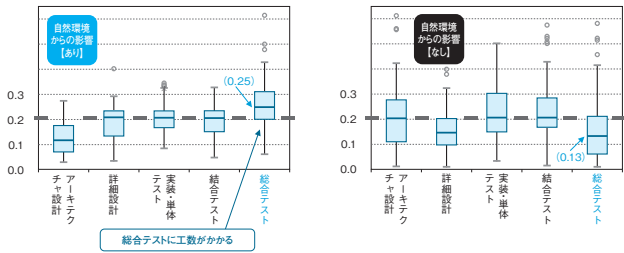


図10 自然環境からの影響を受ける場合の工数実績の工程比率

(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性を散布図(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。

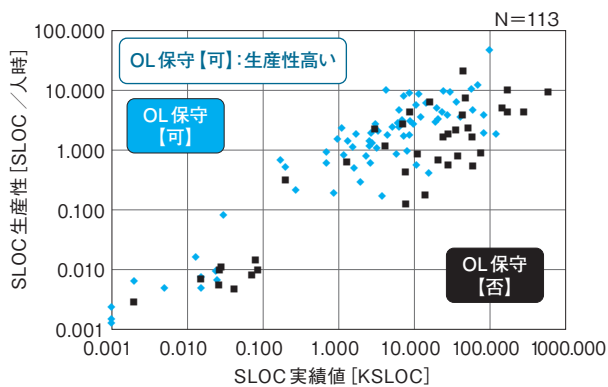


図11 オンライン保守が可能な場合のSLOC生産性

(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。

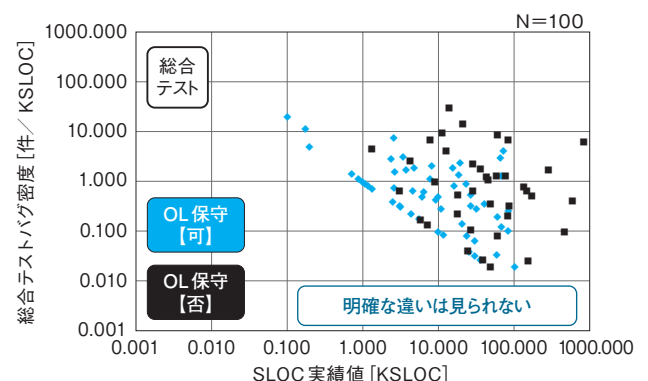
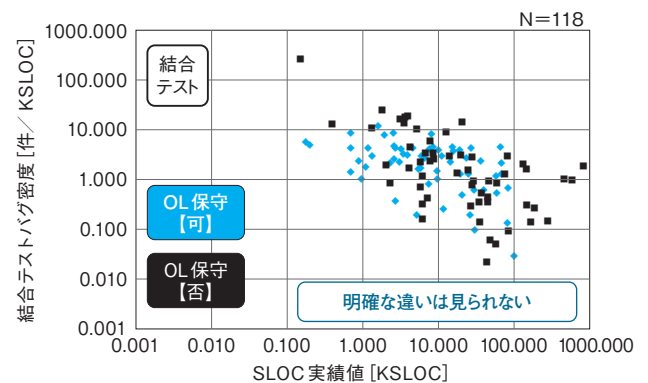


図12 オンライン保守が可能な場合のテスト検出バグ密度

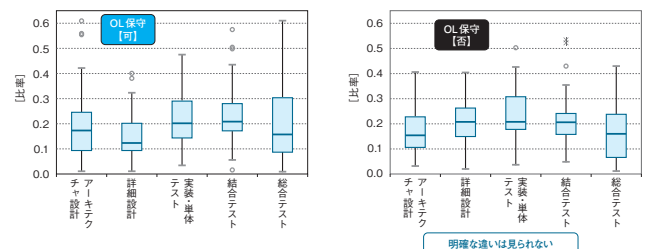


図13 オンライン保守が可能な場合の工数実績の工程比率

NECにおける定量的管理

～ソフトウェア品質会計とNECデータ白書に基づく改善活動の展開～

日本電気株式会社 宮崎 義昭 日本電気株式会社 誉田 直美

弊社ではソフトウェア開発全域において、長年にわたり、品質会計という標準的な品質管理手法を適用し、高品質なソフトウェア開発に努めてきた。また、開発途中で測定した膨大なプロジェクトデータを、グループ各社を含む全体規模で共通的に収集し、分析を加えた結果をデータ白書としてまとめ毎年発行している。

この分析から、上工程重視により品質向上と生産性向上が両立可能であること、CMMI成熟度レベルを向上することにより品質向上できることなど、多くの知見を得ることができた。更に、これらの知見を基に、組織的なプロセス改善のPDCAサイクルを回すことにより、ソフトウェア開発の品質と生産性の向上に取り組んでいる。本稿では、これまでの活動の概要と成果及び今後に向けた展開について報告する。

1 はじめに

NEC及びNECグループ各社では、高品質ソフトウェアを開発するための品質管理技法である「ソフトウェア品質会計」の適用を基盤とし、ソフトウェア品質の改善に向けた様々な取り組みを継続的に行っている。ソフトウェア開発途中で収集しているプロジェクトデータをグループ横断的に収集し、統計的な分析と考察を加えた結果を各組織にフィードバックすることで、データに基づくプロセス改善活動を全社的に展開している。これは、一部の事業部門において以前から行っていた活動を、2011年度から、グループ全体に対象範囲を広げ発展させてきたものである。現在では、グループ全体の開発量のうち、8割を超えるプロジェクトデータを収集するに至っている。

2 ソフトウェア品質会計

2.1 ソフトウェア品質会計の位置付け

このデータに基づくプロセス改善活動は、「ソフトウェア品質会計」(以降、品質会計)という弊社独自のソフトウェア開発の品質管理技法が基盤となっている [Honda2010]。品質会計は、グループ全体で標準的な品質管理技法として適用しており、この共通基盤があるからこそ、広範囲の事業領域において横断的な視点での取り組みが可能となったと考える。

2.2 品質会計の概要

品質会計は、基本設計からコーディングまでの工程(以降、上工程)で作込まれたバグを負債と考え、これをレビュー及びテストにおいて摘出/修正することで負債を返済し、残存バグがなくなり負債がゼロになった時点で出荷する、という考え方に基づく。その特徴は、上工程のレビューを充実させ、早い段階でより多くのバグを摘出する「上工程重視」と、開発途中の品質状況や実際に摘出したバグの分析により的確に「テスト終了判断」を行うことにある。

品質会計を構成する基本的な技術体系を図1に、また各技法の使用法と特徴を表1に示す。「バグ目標管理技法」は、開発途中のレビューやテストで摘出されるバグ数を目標管理するための技法であり、「上工程品質会計」「テスト工程品質会計」から成

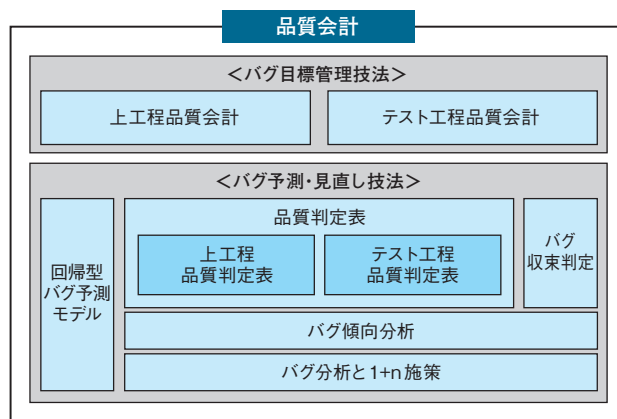


図1 品質会計の技術体系

る。「バグ予測・見直し技法」は、開発状況に応じて残りのバグ件数を予測し適宜見直しを行う技法であり、「回帰型バグ予測モデル」「品質判定表」「バグ傾向分析」「バグ分析と1+n施策」「バグ収束判定」の5つから成る。

表1 品質会計を構成する技法とその特徴

カテゴリ	技法	使用法と特徴
バグ目標管理技法	上工程品質会計	上工程（設計～製造）用のバグ目標管理技法。バグの抽出工程と作り込み工程の両面からバグを目標管理する。
	テスト工程品質会計	テスト工程用のバグ目標管理技法。テスト開始時に残存する、プログラム全体の総バグ数を目標管理する。
バグ予測・見直し技法	回帰型バグ予測モデル	開発開始時に、今回の開発で作り込むであろう総バグ数を予測するためのバグ予測技法。
	品質判定表	開発途中に発生する変化を考慮して、バグ目標値を見直すバグ目標値見直し技法。上工程品質判定表とテスト工程品質判定表がある。
	バグ傾向分析	抽出したバグを様々な観点から整理することにより、バグの抽出傾向に偏りが無いかを分析する技法。
	バグ分析と1+n施策	バグ1件ごとに真因を分析することにより、開発上の細かい抜け・漏れを発見し、その抜け・漏れに対して、集中的なレビューやテストにより残存するバグを抽出する技法。 バグ分析と1+n施策はセットで用いる。
	バグ収束判定	テスト度合いに対する累積抽出バグ数の推移により、バグ収束を判定する技法。

3 プロジェクトデータの収集と分析

3.1 収集データ項目の標準化

組織横断的なデータの収集と分析には、収集するデータ項目の標準化が不可欠である。データ項目は、開発するソフトウェアの規模や種類、工数やバグ数などの様々な管理項目に対する計画値と実績値に加え、統計的な分析に利用するための、組織名や業種などのプロジェクトに付随する属性情報、更に、開発が完了し出荷後に抽出されたバグ数などが含まれる。これらのデータ項目に対して、その意味と具体的な測定方法を定義することで、組織によるばらつきを最小限に抑えるようにしている。しかし、各事業領域によってソフトウェアの種類、開発方法、更には歴史的な経緯も様々であり、同じ名前のデータ項目が異なる意味で使われている例や、一部のデータ項目が収集されていない例もあり、データの精度には少なからず課題があるのも事実である。この点についても、継続的に共通化の方向に向けて改善を進めている。

3.2 データの収集と分析

各組織が管理するプロジェクトデータは、後述する全社的なデータ白書の作成に合わせて年1回の周期で収集し、多角的な視点から分析する。なお、ここで収集したプロジェクトデータの一部は、IPA/SECが発行するデータ白書用に提供させていた

3.3 NECデータ白書の発行

分析の結果は、年次レポートである「NECグループ版ソフトウェア開発データ白書」として発行し、グループ内に公開している。データ白書を作成する主要な目的は二つある。一つ目は定点観測である。どれだけの規模のソフトウェアをどのように開発しているのか、この情報を毎年蓄積することで、過去と比較して何が変わり、何が変わらないかが見えるようになる。この結果は、ソフトウェアを取り巻くビジネス環境が急速に変化する中において、それぞれの事業体やグループ全体の視点での経営判断にも重要な情報となる。

二つ目の目的は、ソフトウェア開発の品質／生産性に影響を与える要因を分析し、今後のプロセス改善の方針を考える材料を提供することである。収集したデータを成功プロジェクトと失敗プロジェクトの二つの層に分け、成功と失敗に大きく影響する要因を分析する。グループ全体のデータを対象にすることにより、まとまったデータ件数を確保できるため、事業部単位では実施しにくい細かな層別による傾向分析が可能となる。毎年、新たな視点を加えながら分析と考察を行いメッセージとして発信しており、これらの分析結果は、全社基準値などへ反映することにより、組織的なプロセス改善活動に活用している。

3.4 データ白書による効果

弊社において、データに基づくプロセス改善活動が、取り組み直後から順調に進んだかと言うと必ずしもそうではない。当初は、組織横断的にプロジェクトデータを収集し分析することのメリットが十分に理解されず、データ収集や提供に消極的な組織も存在した。弊社の事業領域は、特定の顧客向けにカスタム開発した業務システム、サーバのOSやミドルウェアのような汎用ソフトウェア製品、更に、特定の機器向けの組込みソフトウェアなど、非常に広範囲にわたっている。事業領域が違えば、そのソフトウェアを開発するための開発プロセスや開発技術に違いが出るのは当然と言える。そのような条件の中で、データに基づく改善活動がグループ全体の取り組みとして定着した要因としては、ソフトウェア開発における品質を作り込むポイントには、ソフトウェアの種類によらず共通な面が多く、大量のデータを集め適切に統計的な分析を行えば、有益な情報が得られることが徐々に認知されてきたことが大きい。分析の結果から、データのばらつきの許容範囲や、複数のデータの関係が数式で示されることによって、これまで経験と勘に頼っていたノウハウが見える化され、基準値として設定したり、客観的な根拠を持った説明が可能になるなどの効果が現れてきた。

以下に、これまでのデータ白書の分析から得られた知見や、データに基づくプロセス改善活動の成果の一部を紹介する。

4 データ分析により得られた知見

4.1 上工程重視による品質向上と生産性向上の両立

ソフトウェアエンジニアリングにおいて、上流工程における品質確保の重要性は古くから言われている。品質会計では、上工程で作り込む全バグ数に対する、上工程終了までにレビューで摘出するバグ数の割合を上工程バグ摘出率と定義し、上工程バグ摘出率80%を目標とすることを推奨している。しかしながら、この目標の達成は、レビューに時間がかかるためにコストが増加し、品質が良くなっても生産性が悪化するのではないかと懸念の声が多かった。これに対して、上工程バグ摘出率が高いほうが品質が良いだけでなく生産性も高くなることを実証し、上工程バグ摘出率80%を達成することの重要性を示すことができた[Maruyama2017]。具体的には以下の通りである。

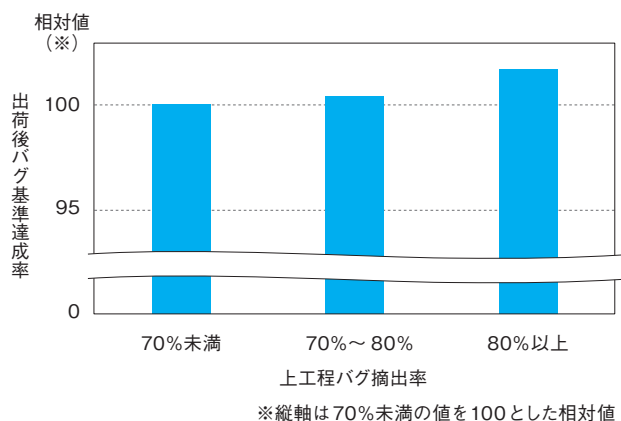


図2 上工程バグ摘出率と出荷後バグ基準達成率

図2は、データを上工程バグ摘出率70%未満、70~80%、80%以上の3つの層に分け、出荷後バグ基準達成率を比較したグラフである。出荷後バグ基準とは、出荷後12カ月に顧客で発生するバグ数を開発規模で除した数値の目標値である。この出荷後バグ基準を達成したプロジェクトの比率を、出荷後バグ基準達成率と呼ぶ。図2によると、上工程バグ摘出率が高いほど、出荷後バグ基準達成率は高くなり、品質が良くなるのが分かる。更に、上工程バグ摘出率が高いほど、単位規模当たりの開発工数は減少し、生産性は高くなる(図3)。その理由は、後戻り工数の削減にある。上工程バグ摘出率で層別した3つの層を比較すると、単位規模当たりのテスト項目数には差は見られないことが分析の結果から得られている。一方で、テスト工程で摘出されたバグ数は、上工程バグ摘出率が高くなるほど減少し(図4)、同様にテスト工程の工数も減少している(図3)。これは、上工程でより多くのバグを摘出することにより、テスト開始時点に

潜在するバグ数が減少するため、同等のテスト項目を実施しても、バグ修正のための後戻り工数が削減されたと考えられる。これが、生産性向上につながったのである。弊社では、これらの結果をもとに、上工程バグ摘出率に対して各組織で具体的な目標値を設定し、実績を評価する取り組みを全社的に推進している。

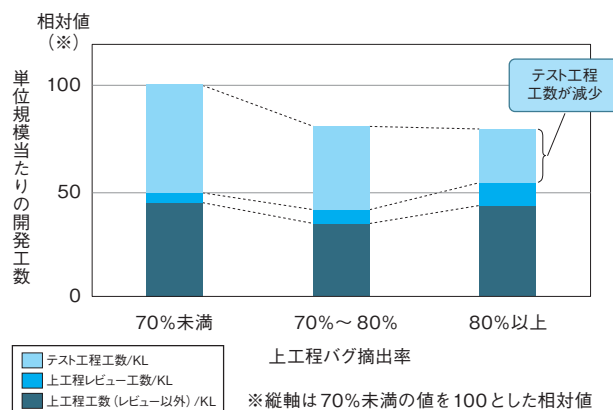


図3 上工程バグ摘出率と開発工数

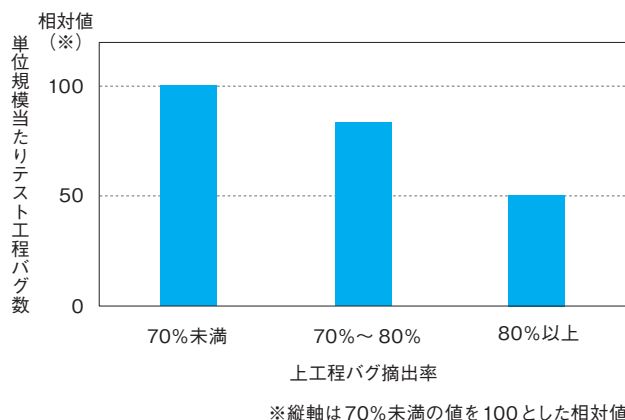


図4 上工程バグ摘出率とテスト工程バグ数

4.2 CMMI成熟度レベルと品質

CMMIの成熟度レベルが上がると各組織が開発するソフトウェアの品質が良くなることは、過去の先行研究においても示されており、これを弊社のデータで検証し、成熟度レベルが上がると、プロジェクトの特性によらず出荷後バグ基準達成率が高くなるという結果を得た[Yanagida2015]。

更に、プロジェクトの成否につながる具体的な要因を明らかにする目的で詳細な分析を行った[Yanagida2017]。成熟度レベル別に分類木を構築した上で、有意差検定及び相関分析を組み合わせて、出荷後バグ基準に対する達成と未達成に影響を与える要因を複合的に分析した。その結果、分類木において、成熟度レベル1と成熟度レベル2では共に開発規模が最も影響が大きい要因であり、達成と未達成を分ける開発規模の境界値は、成熟度レベル2は成熟度レベル1の4倍の値となった。これは、成熟

度レベル1と2の組織は開発規模が大きくなるとプロジェクトを制御できなくなること、成熟度レベル2の組織は、レベル1と比べて4倍の規模のプロジェクトまでは制御できることを示している。また、レベル3になると、開発規模は影響要因ではなくなる。以上の考察から、成熟度レベルの向上は安定したソフトウェア開発に必須と判断した。

上記の分析結果に基づき、弊社では、各組織の成熟度レベルに応じて、遂行中のプロジェクトにおける品質確保の施策と、組織の成熟度レベルを上げることの両面で改善計画を立て、推進する取り組みを行っている。

4.3 自律的なプロセス改善活動の推進

次に、各事業部が自律的に改善活動を進めるための取り組みについて紹介する。2年前から、各事業部において自組織のプロジェクトデータを自ら分析し、改善計画へ反映する取り組みを行っている。これは、グループ全体のデータと自組織データの比較分析を行うことで、グループ内における自組織の位置付けや課題を客観的に把握してもらうことを目的としている。また、この施策を実行できるよう、各事業部に統計分析ができる人材を育成するための教育プログラムを考案し実施している。これにより、各事業部の改善活動に、定量的な目標設定と実績確認のPDCAサイクルを回す仕組みを組み込むことが可能になった。

5 今後に向けた取り組み

5.1 対象領域の拡大

本稿では、ウォーターフォールモデルによるソフトウェア開発プロジェクトを対象にした改善の取り組みについて説明した。現在では、プロジェクトデータの収集と分析の対象範囲をアジャイル開発や、プラットフォーム構築^{*1}のプロジェクトまで拡大している [Honda2016] [Haraguchi2016]。これらの新たな領域に対しても、ウォーターフォールモデル開発と同様に、プロジェクトを特徴付ける品質指標データ項目を定義し、プロジェクトの成否との関係や、遂行途中に管理すべき項目について分析と検討を進めている。今後、データの蓄積が進むことで、新たな領

域でも、組織的な改善サイクルが軌道に乗るものと期待している。

5.2 分析から得られた知見の有効活用

本節では、データ白書を中心とした改善活動から得られた知見を応用した二つの取り組みを紹介する。

一つは、過去データを利用したシステムの提供である。このシステムは、過去データの分析から得られた成功プロジェクトに基づく見積もり機能を備えており、現在進行中のプロジェクトを過去データの中にプロットすることで、自プロジェクトの立ち位置の確認が可能である。また、事業領域ごとの統計値の表示など、収集した過去データを直接利用できるよう工夫し、単にデータを参照できるだけでなく現場で有効に利用することを目指したものである。

もう一つは、データ分析で得られた知見を徹底的に利用し、高品質ソフトウェアを効率的に開発できるよう支援するプロジェクト管理システムの提供である。品質会計技法に基づく品質作り込み状況の自動判定、開発途中ででき上がったソースコードの複雑度やネストの深さなどによる品質自動判定など、開発途中でリアルタイムに収集した品質指標データに基づく問題の有無をシステムが自動判定し警告を出す仕組みである。更に、今まで収集した膨大なデータを使用し、AI(人工知能)によるプロジェクト成否の予測自動判定の実証実験を行っている。AIによる品質予測は非常に高い精度を示しており、本システムによる効果向上に寄与するものと期待している。

これらの取り組みは、いずれも、これまでのデータ収集と分析の蓄積があって初めて可能になるものである。

6 まとめ

組織的な改善活動を継続的に行うためには、活動の成果を常に定量的に管理できるようにすることが必要不可欠であり、今後も、ソフトウェア開発組織としての更なるレベルアップに向けた全グループ横断での活動を推進する。

^{*1} アプリケーションを動作させるために必要なハードウェア(サーバ、ストレージ、ネットワークなど)、OS、ミドルウェアを含むICT基盤の設計及び構築

参考文献

- [Honda2010] 菅田直美、ソフトウェア品質会計、日科技連、2010
- [Maruyama2017] 丸山志保、「レビュー重視と品質・生産性の関係分析」、ソフトウェア品質シンポジウム2017、2017
- [Yanagida2015] 柳田礼子、「効率的な品質改善に向けたCMMI成熟度レベル別の要因分析」、ソフトウェア品質シンポジウム2015、2015
- [Yanagida2017] 柳田礼子、野中誠、菅田直美、「CMMI成熟度レベル別に見たソフトウェア品質の良否にかかわる要因」、SECジャーナル No.49、2017
- [Honda2016] 菅田直美、「アジャイルと品質会計」、情報処理学会デジタルプラクティス Vol.7 No.3、2016
- [Haraguchi2016] 原口剛、「Approach to Build Prediction Model for System platform development projects」、ProMAC2016、2016

SPI活動と連携した定量的プロジェクトマネジメントの実践

TOiNX(トインクス) 東北インフォメーション・システムズ株式会社 開発運用本部 業務管理部 業務統括課 主査 佐藤 浩明

ソフトウェア開発プロジェクトを成功に導くためには、自社のメトリクスを活用した「定量的プロジェクトマネジメント」の実践が有用である。一方で、メトリクスの解析と活用には、多くの検討事項がある。本稿では、定量的プロジェクトマネジメントの実践に向けたSPI活動の教訓や、メトリクスの活用事例について紹介する。

1 はじめに

ソフトウェア開発プロジェクトを成功に導くためには、自社の実績データによるメトリクスを、計画段階から活用する「定量的プロジェクトマネジメント」の実践が有用である。勘と経験だけでは、高精度な計画を策定し、プロジェクトのQCDを正確に監視・制御することは困難である。

一方で、組織の標準プロセスが定着しない状態では、データのばらつきが大きく、メトリクスの精度向上は期待できない。SPI(Software Process Improvement)活動と、データ解析活動は表裏一体の関係である。開発プロセスの成熟度向上には、この2つの活動が連携している必要がある。

本稿では、東北インフォメーション・システムズ株式会社(以下、当社)におけるSPI・データ解析活動の経緯やメトリクスの活用事例を、データ解析に携わる実務者の観点から紹介する。

2 SPI・データ解析活動の経緯

当社は、2001年度からCMMI®(Capability Maturity Model Integration)に基づくSPI活動を開始した。2003年度には成熟度レベル2を、2008年度にはレベル3を達成している。また、2010年度には、一定の実績データが蓄積できたため、データ解析を本格化するための「メトリクス分析支援ツール」を開発した。以降、毎年100パターンを超えるメトリクス解析結果を、プロジェクトに提供している。

約15年間にわたるSPI活動であるが、標準プロセスの定着を実感できたのは、活動開始後10年以上が経ってからである。そして、この契機は、メトリクスの公開と活用にある。定量的プロジェクトマネジメントの実践には、様々なアプローチがある。しかし、当社の経験から言えることは、いかに厳格な計測・収集・解析のプロセスを定義しても、SPI・データ解析活動は浸透しない。重要なことは、プロジェクトが要求するメトリクスを、常にフィードバックし続けることである。

3 メトリクス解析結果とは

当社では、プロジェクトに対して、以下のメトリクス解析結果を提供している。以降、各メトリクスの概要を紹介する。

3.1 「プロセス実績ベースライン」と「プロセス実績モデル」

プロセス実績ベースライン(以下、基準値)と、プロセス実績モデル(以下、予測モデル)は、最も初歩的なメトリクスである。これらは、図1で示すように、社内のポータルサイトで参照できる。

基準値とは、各メトリクスの幾何平均値と上下限値を、開発タイプ別に解析したものである。主に、開発タイプごとの傾向把握に利用するが、後述する予測モデルを確立できない場合、暫定的に予測にも用いる。

予測モデルとは、見積もりや妥当性評価に用いる予測ツールである。開発タイプ別の層別による単回帰分析で、標準的な予測値と任意の信頼率による上下限値を算出する。べき乗モデル($y = b_0 \times X^{b_1}$)を採用しており、プロジェクト規模を反映した予測が可能である。本モデルは、寄与率、係数 b_1 のp値、残差の傾向などを確認し、一定の条件を満たしたもののだけが公開の対象となる。

3.2 メトリクス応用ツール

メトリクス解析結果の応用として、表1に示す各種のツールを開発し、プロジェクトに提供している。本ツールは、以下の①から③のタイプに分類することができる。

3.2.1 プロジェクト特性に応じた重回帰分析(①)

タイプ①は、OSや顧客などのプロジェクト特性を組み合わせた層別に対して、複数の説明変数(FP、SLOC、テストケース数など)を用いた重回帰分析の仕組みである。各種の特性を反映しており、予測モデルより格段に精度が高い。代表的なツールとして、工数見積もりの妥当性を評価するための「工数評価ツール」がある。

3.2.2 複数メトリクスの統合(②)

レビューやテストなどは、単一の指標で品質を判断することが難しい。このため、複数の基準値を組み合わせた「ゾーン分析ツール」が有効になる。タイプ②である本ツールは、開発タイプ、対象工程、欠陥・指摘の重要度に基づく9つのゾーンを設定し、今、どのゾーンに該当するのかが確認できる。

3.2.3 プロジェクト特性に応じた基準の再設定(③)

プロジェクトは、多様な特性があり、必ずしも組織の基準値に合致するとは限らない。タイプ③は、工程進行中の実績情報から、プロジェクト特性を反映した基準値を再設定し、管理するものである。上流工程でのレビューを管理する「u管理図作成ツール」や、統合テスト工程以降で利用する「信頼度成長モデル」などがある。

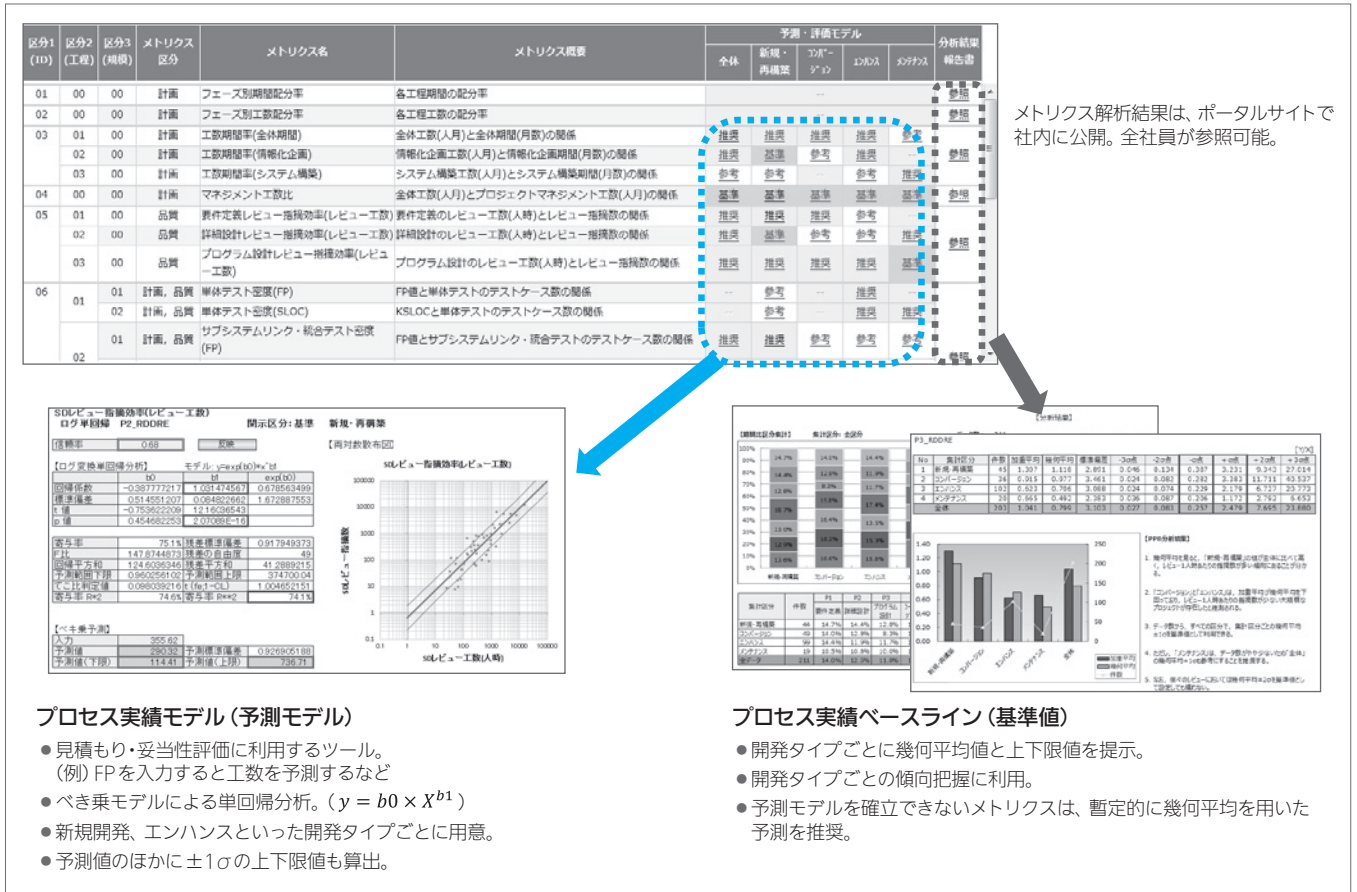


図1 基準値・予測モデル

表1 メトリクス応用ツール

No.	カテゴリ	ツール名称	タイプ	機能概要
1	工数・規模	工数評価ツール	①	PJ特性に応じた工数と、その上下限値を予測。PJ初期段階の工数見積りの妥当性評価に利用。
2		工数生産性評価モデル	①	PJ特性に応じた生産性を評価。
3		FP値 (IFPUG) 推定	—	画面や帳票数などの限定された情報からFP値を推定。
4	WBS/EVM	ライン数カウント	—	空白行・コメント行を除いたライン数をカウント。修正行のみカウントすることも可能。
5		WBSツール	—	WBSとEVMの作成を支援。
6		統合EVMグラフ	—	複数のWBS/EVMツールのEVMグラフを統合。
7		WBS要員展開ツール	—	要員ごとの計画・実績工数を月別(日別)に展開。
8	品質	検出欠陥予測ツール	①	統合テストの総検出欠陥数をPJの特性に応じて予測。
9		ゾーン分析(レビュー)ツール	②	要件定義～PGM設計のレビューのデータから工程完了時の品質を予測。
10		テスト進捗管理表(ゾーン分析ツール)	②	テスト工程の進捗管理とゾーン分析を組み合わせたツール。
11		u管理図作成ツール	③	各種レビューや単体テストのデータからu管理図を作成。
12		信頼度成長モデル	③	指数型モデルを用いた統合テスト以降の定量的品質管理を支援。

4 SPI・データ解析活動の教訓

当社のSPI・データ解析活動では、多くの失敗を経験した。そして、このほとんどが、活動初期における初歩的な考慮不足が原因である。ここでは、本活動の主な教訓を記載する。

4.1 開発プロジェクトとの合意形成

データ計測の主体は、あくまでプロジェクトである。正確な実績データの蓄積には、開発現場の協力が不可欠である。しかし、計測が習慣となっていない組織では、計測を不要な作業と捉え、敬遠する傾向がある。

データ解析の最大の難関は、計測開始からメトリクスを提供するまでに、ある程度の期間を要する点である。このため、活動初期にメトリクスの活用で「何ができるのか」「どう役立つのか」を、開発現場と共有することが重要である。将来のビジョンを具体的に伝え、計測に対するモチベーションを維持することがねらいである。

当社の場合、この共有が不十分であった。このため、初期に収集したデータは、欠損が多い状態となった。現在は、後追いではあるが、プロジェクトが求める（または使用頻度が低い）メトリクスを毎年調査し、要求に応じたメトリクスを拡充している。

4.2 データ計測規則の定義

通常、計測規則は活動の初期で定義するが、当社は、この定義が不完全であった。レビューやテストで検出した欠陥の重要度や、テストケースの粒度などが典型例である。

しかもこの問題は、データ蓄積時に判明せず、メトリクス解析時に初めて表面化した。例えば、「テスト密度(=テストケース数/規模尺度)」の解析で、散布図を作成した際に、予想外の集落やばらつきが散見された。この原因を調査した結果、ある部門では、テストケース数を計測すべき個所に、テストのシナリオ数を登録していたことが判明した。このように、計測の慣行は、プロジェクトや部門ごとに大きく異なることがある。

この状況を踏まえ、組織共通の計測規則を再定義し、徹底した。しかしながら、初期に蓄積した数十プロジェクトのデータは、規則に準じたものかどうかをさかのぼって確認できず、利用できないものとなった。

また、品質・工数・工期に影響を与える定性要因(質的データ)を、初期段階から計測しなかったことも大きな失敗である。量的データの解析は、層別が基本である。ところが、層別を繰り返した場合でも、いずれ壁にぶつかることがある。

この解決策は、要件の確定度や顧客の関与度といった定性要因を、量的データと併せて解析することである。当社は、定性要因をダミー変数^{*1}化した重回帰分析も適用しており、量的データだけよりも、高精度な予測になることを確認している。しかし、この定性要因の蓄積が遅れたため、初期のデータを解析の対象にできないという問題がある。

4.3 実績データの鮮度と精度の確保

当初、プロジェクトが計測したデータは、プロジェクト完了時に「完了報告」として取りまとめ、SPI担当者に提供する運用であった。しかし、この運用が、データの鮮度と精度を損なう原因となった。このため、各種の実績データを一元管理するためのシステム(以下、Co.守り)を開発すると共に、データの登録プロセスを見直した。

4.3.1 実績データの鮮度の確保

当社のプロジェクトは、年々大規模化し、工期も長期化する傾向にある。このような状況で、プロジェクト完了時に、上流工程のデータをさかのぼって計測することは現実的でない。このため、Co.守りへのデータの登録を、該当工程の完了後10営業日以内とした。

また、当社では、各工程で標準プロセスに対する遵守状況を、第三者が評価する品質保証活動を実施している。この評価では、実績データの登録状況も確認し、未登録の場合は、非遵守課題として指摘し、プロジェクトに是正対応を催促している。

4.3.2 実績データの精度の確保

最終工程完了時には、Co.守りを經由して、完了報告に対する同意依頼がSPI担当者に届く。SPI担当者は、登録された約300のデータ項目をすべて確認し、指摘や疑問がある場合は、この依頼を何度でも差し戻す。

この確認は、メトリクスがあるからこそ可能である。実績データは、それ単品では判断が難しい。例えば、テストケース数という基本測定量だけでは、その値が多いのか少ないのかを判別できない。しかし、メトリクス「テスト密度」で、テストケース数とFPの2変数の関係を確認することで、要注意データが特定できるのである。

4.4 データ解析技法の習得と効率化

当社は過去に、統計技法を理解せずに解析を行った。データの分布を確認せずに、算術平均値と標準偏差で、基準値と上下限値を設定するといった誤りである。しかし、こうして得た結果は、プロジェクトの実情と乖離し、予測や評価に利用できないものであった。

ITメトリクスは、そのままの値(真値)で正規分布に近似することが少ない。当社では、多くのメトリクスが対数正規分布に近似することを確認している。このような性質を持つデータは、事前に変数変換などの処理が必要である。統計学では周知の知識や技法を習得しなければ、解析に不要な時間を費やすだけでなく、誤った結果を導くことになる。更に、ITメトリクスの解析では、統計技法に加えて、プロジェクト特性やIT分野固有の技術的知見(FPや欠陥の性質)も考慮する必要がある。

メトリクス分析支援ツールの開発にあたっては、事前に解析手順や技法を十分に習得した。一般的なメトリクスの解析手順を、図2に示すが、本ツールは、このすべての機能を網羅している。

*1 定量的ではない質的・属性的なデータを「0」と「1」だけの数列に変換すること。



図2 ITメトリクスの解析手順

5 メトリクスの活用事例

メトリクスの活用形態は以下の3段階を経た。以降、図3で示す、統合テスト工程を例に、当社の品質管理の変遷を紹介する。

- ① 第1段階(自社メトリクス非活用)
- ② 第2段階(自社メトリクスの初期活用)
- ③ 第3段階(自社メトリクスの応用活用)

第1段階におけるテスト方針書・計画書では、テストの範囲・方法・環境などを記載する。しかし、品質の評価は、工程完了時におけるテストケースの消化状況の確認や、欠陥の原因・傾向分析が中心であり、勘と経験に基づくものである。

第2段階では、計画時に予測モデルを使って組織の基準値を目標として追加する。典型的には、開発規模に応じたテストケース数を予測する「テスト密度(=テストケース数/規模情報)」や、どの程度の欠陥を検出するべきかを予測する「欠陥検出密度(=検出欠陥数/規模情報)」を用いる。

品質の評価は、工程進行中から、図4の「ゾーン分析ツール」を活用する。本ツールは、統計的な予測ではないものの、目標となるゾーン(⑤)に対して、どのように推移しているのかを確認できる。工程完了時は、第1段階の分析に加えて、組織の基準値

との乖離を分析する。しかし、評価は、あくまでも組織の基準値との比較にしかすぎない。

第3段階では、メトリクス応用ツールである「検出欠陥予測ツール」や「信頼度成長モデル」を用いる。検出欠陥予測ツールは、図5のように、規模情報と特性を入力すると、この条件を満たす最も高精度の予測式を自動選択し、欠陥数を予測する。このツールにより、自プロジェクトの特性に応じた目標の欠陥数が設定できる。

更に、この欠陥数を入力値として、図6に示す「信頼度成長モデル」が利用できる。本ツールは、計画時に欠陥数とテストケース数を入力することで、指数型モデルに応じた検出欠陥数と上下限値の基準を設定する。テスト進行中は、欠陥の検出状況を反映し、完了時の検出欠陥数、潜在欠陥数、欠陥除去率と、目標の欠陥除去率を達成するための追加テストケース数などを予測する。また、組織の基準との乖離が大きい場合、プロジェクトの欠陥検出状況から、プロジェクト特性に応じた基準を再設定できる。

このように、第3段階では、組織の基準だけではなく、自プロジェクトの特性を反映した基準で、品質が評価できる。そして、評価は工程完了時ではなく、テスト進行中から継続的に実施するのである。

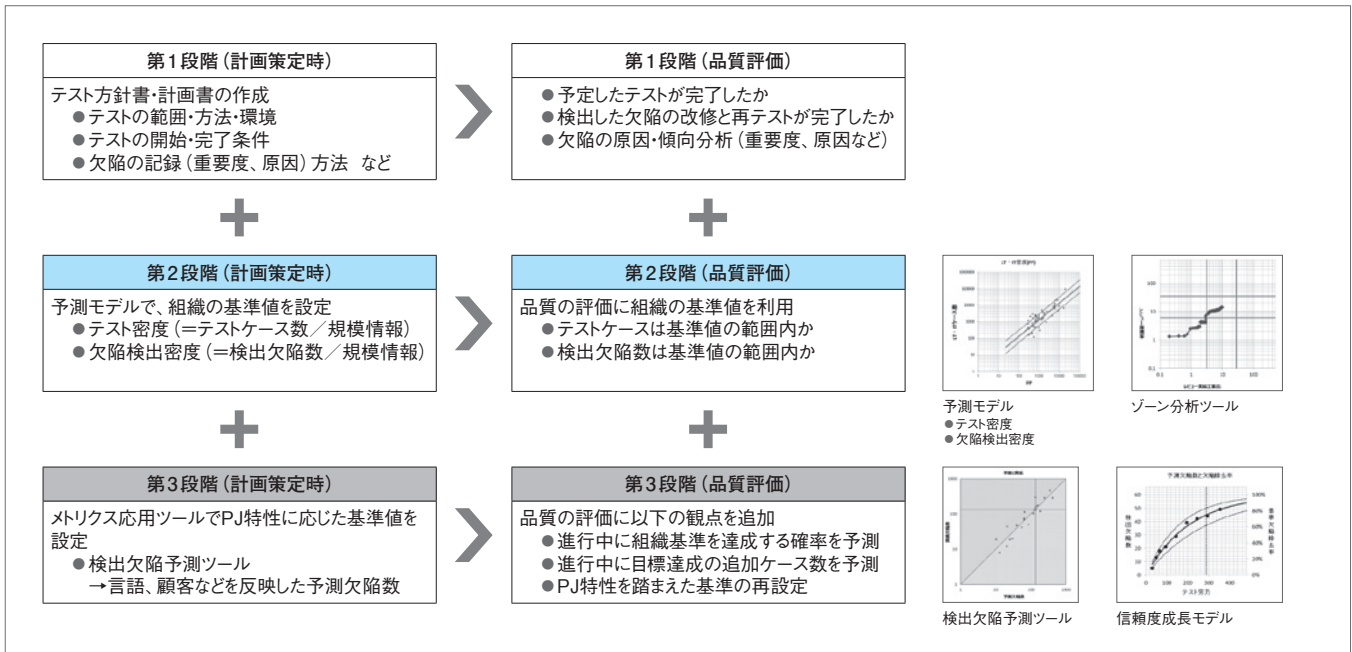


図3 メトリクスを活用した品質管理

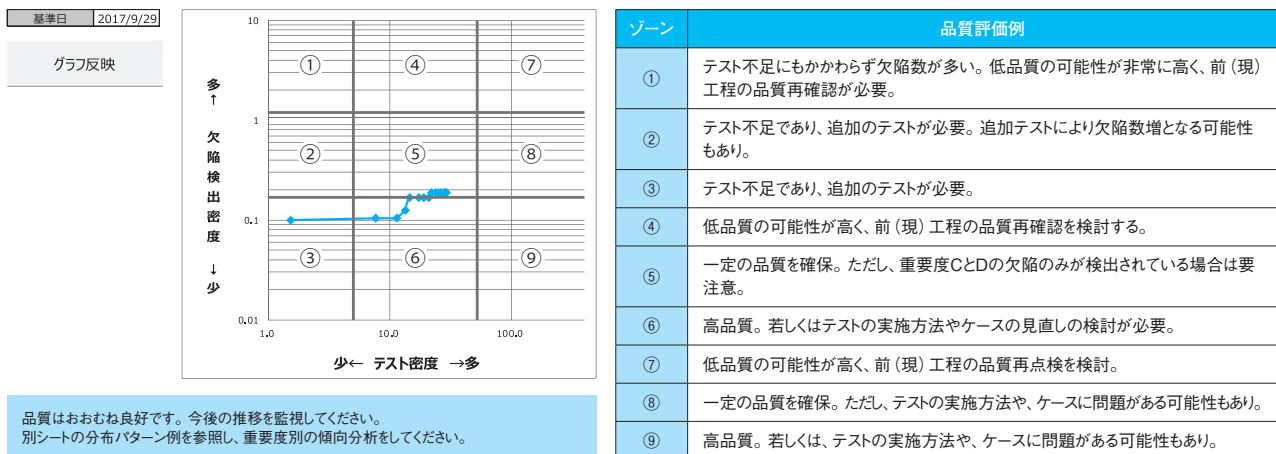


図4 ゾーン分析ツール

Ver 2.00
TOINX 検出欠陥予測ツール(統合テスト)

【規模】
FP入力タイプ: 直接入力
直接入力FP値: 1000
SLOC (K): 50
LT数+IT数: 800

【プロジェクト特性】
開発タイプ: 新規開発
主OS: Solaris
主言語: Java
顧客: 顧客A

【予測欠陥数と上下限値】
予測欠陥数: 59
下限: 36
上限: 97

【選択欠陥数モデル】
モデル順位: 1
モデルID: N030F_DM
開発タイプ: 新規開発
主OS: Java
顧客: 顧客A
規模要因: F
データ数: 20
寄与率: 76.89%
上下限係数: 0.5

図5 検出欠陥予測ツール

SRGM for TOiNX V130

データ反映

分析対象データ **工程A1**

[計画値] **自動**

基準潜在欠陥数	80
予定テスト労力	100.0
形状パラメータ	0.023
位置パラメータ	0.000

[設定と予測]

基準設定 実績反映

SRGM予測 記録 消去

テスト労力追加 基準再設定

[実績情報]

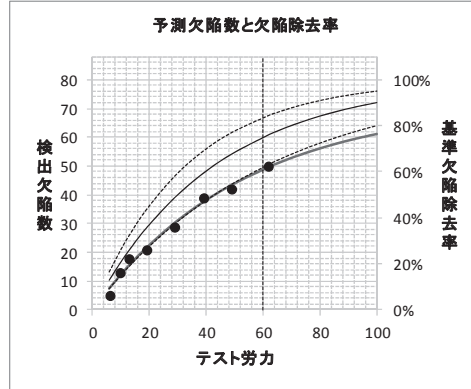
テスト労力	62
検出欠陥数	50
テスト進捗率	62.0%

[データ分割]

分割点探索 分割実行

テスト労力分割点 分割消去

指数モデル: $m(t)=D(1-EXP(-b(t-t_0)))$



[処理結果] SRGM予測完了

[予測結果]

完了時基準達成確率	19%
完了時欠陥除去率	85%
潜在欠陥数	72
完了時検出欠陥数	61
信頼区間下限	55
信頼区間上限	68
現時点欠陥除去率	69%
除去率目標達成テスト労力	121.9
追加テスト労力	21.9
形状パラメータ	0.019
位置パラメータ	0.244

[テスト実績及び欠陥数管理限界]

No.	テスト 労力	検出 欠陥数	予測 欠陥数	除去率	基準欠陥		
					下限	基準	上限
1	6	5	7	9%	7	10	13
2	10	13	12	15%	12	16	21
3	13	18	15	19%	15	21	26
4	19	21	22	27%	21	28	35
5	29	29	30	38%	30	39	46
6	39	39	37	47%	37	47	55
7	49	42	43	54%	44	54	62
8	62	50	50	62%	51	61	68
9	65		51	64%	52	62	69
10	68		52	65%	53	63	70
11	71		53	66%	54	64	70
12	74		54	68%	56	65	71
13	77		55	69%	57	66	72
14	80		56	70%	58	67	73
15	83		57	71%	59	68	73
16	86		58	72%	60	69	74
17	89		59	73%	61	70	74
18	92		59	74%	62	70	75
19	95		60	75%	63	71	75
20	100		61	76%	64	72	76

初期段階では、組織の基準を利用。ただし、組織の基準との乖離が大きい場合は、指数型モデルによる基準の再設定が可能である。

テスト計画時

- 工程完了時の予測潜在欠陥数とテストケース数を入力することで、指数型モデルに応じた検出欠陥数と上下限值（組織の基準）を設定。

テスト進行中・完了時

- 検出欠陥数が組織の基準の管理限界内に入っているかどうかを視覚的に確認。
- テストの進捗が60%の時点で、テスト完了時の検出欠陥数、潜在欠陥数、欠陥除去率を予測。
- 目標となる欠陥除去率（90%）の達成に必要な追加テストケース数を予測。

図6 信頼度成長モデル

6 おわりに

メトリクスの活用によって、マネジメントのスタイルは変化する。各種の判断は、「定性・主観」から「定量・客観」となり、そのタイミングは「工程完了時」から「工程進行中」にシフトする。更に、品質やプロセスの目標は、「組織の基準」から「プロジェクト特性に応じた基準」に変わる。

開発プロジェクトでは、課題やリスクを工程完了時に特定しても手遅れである。より早期に要注意や異常な状態を特定し、是正する必要がある。事実、プロジェクトがデータ解析者に要望するのは、このようなメトリクスである。

一方で、高精度なメトリクスを提供するためには、組織レベルでのプロセス定着を推進する、SPI活動との連携が必須である。

従って、データ解析者のミッションは幅広く、蓄積した実績データを解析することは活動の一部にしかすぎない。

プロセスの改善機会は、常にプロジェクトからである。プロジェクトに有用なメトリクスを、いかにタイムリーに提供できるのかが、SPI・データ解析活動を加速するためのキーである。

参考文献・引用転載

[1] 佐藤浩明：ソフトウェア開発プロジェクトの生産性評価に関する事例—真の価値と工数削減効果を計測する手法、ユニシス技報 Vol.36 No.2 通巻129号、日本ユニシス株式会社、2016年

[2] 梶山昌之・合田英二・千野智子：ソフトウェア開発プロジェクトの計数管理フレームワークによる定量的管理、プロジェクトマネジメント学会誌 Vol.13、一般社団法人プロジェクトマネジメント学会、2011年

ソフトウェア定量的管理にかかわる 学術研究事例

東洋大学 経営学部 教授 **野中 誠**

九州大学 システム情報科学研究所 准教授 **亀井 靖高**

和歌山大学 システム工学部 准教授 **大平 雅雄**

産業界でのソフトウェアの定量的管理に参考となり、役立つことが期待される学術研究の事例について、筆者ら3名がそれぞれかかわった最近の研究事例と、そこから読み取れる知見を紹介する。その上で、産業界でのソフトウェア定量的管理に期待することを最後に述べる。

1 はじめに

ソフトウェア工学の一分野として、ソフトウェア開発に関する大量かつ多様なデータの分析を通じて、提案手法の有効性を実証したり、データに見られる関係性を抽出して知見を得たりする学術研究が盛んに行われている。そうした学術研究は、オープンソースソフトウェア (OSS) のリポジトリを主な分析対象としたMSR (Mining Software Repositories) 研究などを中心に、2000年代になってから多くの研究者が取り組んでいる。OSS開発と産業界におけるプロプライエタリなソフトウェア開発の境界が融合しつつある今日において、MSRなどで積み重ねられてきた学術研究の知見は産業界においても有用であると考えられる。

本稿では、産業界におけるソフトウェア定量的管理にとって参考となり、役立つことが期待される学術研究の事例について、筆者ら3名がそれぞれかかわった最近の研究事例と、そこから読み取れる知見を紹介する。まず、亀井がかかわったモダンコードレビュー記録の分析について紹介する。次に、大平及び亀井がかかわった不具合修正タスクの最適割り当て問題に関する研究を紹介する。続いて、野中がかかわった組織の能力成熟度別に見たりリリース後品質の善し悪しを分けるポイントに関する研究を紹介する。そして、亀井がかかわった直近の研究である、ソフトウェア進化とバグ予測の関係に関する研究を紹介する。最後に、これらを踏まえた上で、産業界でのソフトウェア定量的

管理に期待することを述べる。それぞれの研究事例の詳細については、参考文献に挙げた情報をもとに原文をご参照されたい。

2 モダンコードレビュー記録の分析

Shimagakiら^[1]は、ソニーモバイルコミュニケーションズ株式会社でのソフトウェア開発における約20,000件のコミット記録を対象に、コードレビューを実施したものとそうでないものを比較し、その違いがリリース後欠陥数に影響しているかどうかを分析している。その結果、コードの規模、コミッターの人数、及びリリース前に検出したバグ数がそれぞれ多くなるとリリース後欠陥数は増える傾向にあるが、コードレビューが未実施であってもリリース後欠陥数が増えるわけではないという、ソフトウェア品質管理の観点からの期待感とは異なる結果を示している。一方で、OSSの外部コードをレビューせずに取り込んだ場合、リリース後欠陥数が増える傾向にあるという、こちらは品質観点からの期待感に沿った結果を示している。また、得られた分析結果とその含意を開発者に伝えたところ、約7割の開発者がこれに賛同したことを示している。

この分析結果から、「内部コードのレビューは不要である」ということを筆者らが主張したいのではないことに留意されたい。ここで述べたいのは、このような分析の実施基盤としてWeb上のコードレビューツールGerritが利用されており、データ入力

のための人手による追加コストが不要であるという点である。そして、分析の結果として得られた情報を、開発プロセス改善のためのコミュニケーションツールとして利用している点である。OSSの開発では、地理的・時間的に分散した開発者がGerritのようなツールを使ってコードレビューを行うことが一般的である。このようなコードレビューは、従来型の対面によるフォーマルなインスペクションに対比して「モダンコードレビュー」と呼ばれている。Gerritのようなツールを使えば、レビュー実施記録の基礎データを容易に収集することができ、定量的管理に利用できるだけでなく、開発プロセス改善への有用なインプットとして利用できる。この研究事例は、そうした改善サイクルをスムーズに回すためにツール利用の観点から貢献できることを示唆している。

そして、モダンコードレビューと呼ばれる最近の取り組みを、組織的な定量的管理の文脈に位置付けることの必要性を述べておきたい。組織的な定量的管理の仕組みは、ひとつたび確立されると組織の慣性力が働くために、データ収集や分析の枠組みを変えていくのが難しくなりがちである。一方で、OSSの開発においては、モダンコードレビューのようにツールや開発スタイルの進化が著しい。定量的管理の仕組みが技術の進化に取り残されることのないよう、技術動向や学術研究の動向に目を向けておく必要がある。そのような意味合いからも、この研究事例を捉えていただけると良い。

3 不具合修正タスクの最適割り当て

柏ら^[2]は、不具合管理システムに報告された不具合の修正タスクを担当者に割り当てるときに、特定の担当者に修正タスクが集中することなく、かつ、プロジェクト全体での不具合修正時間をできるだけ短縮できるようなタスク割り当て手法を提案している。具体的には、不具合の生じたコンポーネントの分類、対応の優先度、その分類における開発者の過去の不具合修正時間などを入力情報として、整数計画法によりタスク割り当て案を作成するというものである。この手法をMozilla Firefox及びEclipse Pluginプラットフォームなどの大規模OSS開発プロジェクトの実績データに適用してシミュレーションを行った結果、一部の開発者に修正タスクが集中するという問題を緩和でき、プロジェクト全体での不具合修正時間を従来の方法に比べて約35～50%削減できる可能性があることを示している。

この手法は、TracやRedmineなどのタスク(チケット)管理システムがあれば適用可能であり、応用範囲は広いと言える。そのほかに、この手法を適用するベースとして、それぞれの開発者がそれぞれのコンポーネントの分類に対してどのくらいの不具合修正時間を必要としたのかといったパラメータ値が必要である。ただし、提案手法ではその中央値のみを用いていることから、典型的な修正時間の推定値を初期のパラメータ値とし、手法を運用しながら修正時間の実測値に基づいてパラメータ値を適宜更新していけば良いだろう。このように、この研究事例で示された手法は、実際のソフトウェア開発に適用する上でのハードルが低いことも特徴である。

実際のソフトウェア開発において、不具合修正タスクが特定の担当者に集中し過ぎないようにタスクを割り当てるといったこと自体については、日常的に行われているだろう。しかし、この手法が示すように、定量的なデータに基づいて不具合修正のタスク割り当て案を作成し、これを実際のタスク割り当ての際の参考情報として意思決定しているといった取り組みは、筆者らの主観的観測としてはほとんどないように思える。ソフトウェアの定量的管理はアクションに結び付けることができこそ意味がある。そのため、ソフトウェア開発におけるデータに基づく意思決定を支える一つの方法として、このような研究に目を向けていただけると良い。

この研究事例では、OSSを対象に手法の有効性を検証している。筆者がこのような研究事例を産業界の実務者に紹介したときに、過去には「OSSの開発と産業界でのソフトウェア開発は異なるため、参考にならない」という批判をしばしば頂戴した。しかし、それは一昔前の話である。今日においてはOSSを取り入れたソフトウェア開発が当たり前となり、OSSとプロプライエタリなソフトウェアの関係性をソフトウェア開発戦略の中に位置付けることが重視される。より積極的には、OSSへの貢献を含めたオープンイノベーションも重要な戦略になるであろう。「OSSで有効性が示されたことは実務においても有効である可能性が高い」という認識を持つことが必要であろう。

4 組織の能力成熟度別に見た品質の善し悪しを分けるポイント

柳田ら^[3]は、日本電気株式会社における522件のソフトウェア開発プロジェクトデータを対象に、CMMI(Capability Maturity

Model Integration)のソフトウェア能力成熟度レベル1及び2のそれぞれについて、リリース後品質の善し悪しを分ける要因を分析している。その結果、リリース後品質の善し悪しを分ける最初の特徴的な要因は、いずれの成熟度レベルにおいてもソフトウェア開発規模であるが、その閾値はレベル1よりもレベル2のほうが4倍以上大きいことを示している。つまり、レベル1の組織に比べて、レベル2の組織では、ソフトウェア開発規模が多少大きくなっても品質問題が生じるリスクを低く抑えられることを示している。

また、レベル1においては、開発規模の次に位置付けられる特徴的な要因はレビュー指摘数であり、これが「少ない」ほどリリース後品質が良いという、ソフトウェア品質管理の観点からの期待感とは逆の結果が得られた。すなわち、レビューで多くのバグを指摘できていることよりも、そもそもバグの混入数が少ないことのほうがリリース後品質の良さに影響しているものと思われる。一方、レベル2においては、開発規模の次に位置付けられる特徴的な要因はテストバグ数であり、これが少ないほどリリース後品質が良いという結果が得られた。こちらについては、品質管理の観点からの期待感と整合した結果であると言える。

この研究事例を通して述べたいこととして、第一に、データの現れ方は分析対象のセグメントによって異なり、それに伴って定量的管理の着眼点や基準値も変える必要があるということである。この研究事例では、それぞれの能力成熟度レベルのセグメントにおいて、どの程度の規模を超えたときにプロジェクト管理レベルを高める必要があるかを示唆している。また、その次に着目すべきポイントは、レベル1の組織ではバグ混入を抑制することであり、レベル2の組織ではテスト重視であることを示唆している。改善施策を異なるセグメントに対して一律に展開しても必ずしもうまく行かず、それぞれの状況に合わせた施策が必要である。この研究事例は、それを定量的に示す一つの方法を示している。

第二に、データ解析手法はプロジェクトの実態把握に役立つということである。この研究事例では、分類木と呼ばれる手法を適用することで、リリース後品質の善し悪しを分ける要因の階層的な分析を行っている。分類木により影響要因を階層的に示すことで、分析結果を直観的に理解することが容易になる。ソフトウェアの定量的管理においては、データで把握した結果を組織へとフィードバックすることが必要であり、結果の「分かりやすさ」も重要な要素の一つである。

第三に、実際にデータ分析をしてみると、期待とは異なる関

係が見いだされることがある。この研究事例では、成熟度レベル1における第二の要素、すなわちレビュー指摘数が少ないほどリリース後品質が良いという関係が見いだされている。これを因果関係として捉えるべきではないことは言うまでもないが、このような事前の期待とは異なる結果は、開発プロセス改善に向けた議論の契機となり得る。

5 進化するソフトウェアにおけるバグ予測に対してどの範囲のデータを使用すべきか

McIntoshら^[4]は、OSSの中でも変更が多く進化の激しいQTとOpenStackを対象に、ソースコードのコミットに対してバグが含まれるかどうかの予測を、約3年半にわたる、およそ38,000件のコミットデータを対象に実施している。分析の結果、OSSが進化していくにつれて、バグ予測の精度が大幅に低下していくという現象を示している。分析により得られた知見として、進化の激しいソフトウェアにおいては、直近3カ月程度のデータに基づいてバグ予測モデルを構築することが望ましいことなどを示している。なお、多くのバグ予測研究ではファイルを予測の単位としているが、この研究ではより細かな変更を予測の単位としている。

この研究事例から得られる知見は、過去データをどこまで使用して予測モデルを構築すべきかという問いについて、OSSの事例を通して実践的に示している点である。そして、進化の激しいOSSの変更履歴データに基づくバグ予測の精度を保つには、3カ月程度が望ましいという一定の見解を示している点である。もちろん、この期間は予測対象の変化の度合いや、得られるデータサイズにも依存するが、一定の目安として有用な知見と言える。

なお、このような予測モデルへの入力データ範囲をどこまでとすべきかについては、工数見積り分野でも研究が行われており(例えば^[5])、一般にウィンドウングと呼ばれている。ただし、ソフトウェア開発プロジェクトを対象とした工数見積りモデルと、ソースコードを対象としたバグ予測とでは、入力データの連続性と変化量の観点で大きく異なる。工数見積りモデルにおけるウィンドウングに興味がある方は、天嵩^[5]の論文などを参照されたい。

6

産業界での ソフトウェア定量的管理への期待

本稿で紹介した研究事例を踏まえた上で、産業界におけるソフトウェア定量的管理への期待として、第一に、開発手法やツールの進化を定量的管理へと取り入れて、定量的管理の仕組みを進化させることが挙げられる。産業界におけるソフトウェアの定量的管理は、品質管理や品質保証の部門、若しくはプロジェクトマネジメントオフィスなどが担当することが多い。こうした部門は開発部門に比べて開発支援ツールを直接的に利用する機会が少なくなる傾向にあるため、開発現場でのツール利用と定量的管理の仕組みが乖離してしまいがちである。ツールの利用状況を把握し、ツールからのデータ抽出スキルを身に付けるなどして、開発と一体となって進化していくことのできる定量的管理の実現が期待される。あるいは、開発現場での工夫により生み出されたツールと連携した定量的管理の仕組みを、プロジェクトレベルや組織レベルへと引き上げ、組織内へと展開することが期待される。

第二に、意思決定を支援し、アクションに結び付く定量的管理の実現である。そして、その対象を、開発チームでの活動から組織的改善まで幅広く捉えることである。3節で紹介した不具合修正タスクの最適割り当ては主にプロジェクト単位で行われる活動であり、4節で紹介した品質の善し悪しを分けるポイントから得られた知見は組織的なプロセス改善にかかわる活動である。ソフトウェアの定量的管理と言えば後者の組織的な取り組みの例がなじみ深いと思われるが、プロジェクト単位やチーム単位での活動にも着目できると良い。ただ、そうした単位に

着目してデータ分析が行えたとしても、適切なアクションへと結び付けることは容易なことではない。しかし、アクションにまで結び付けなければ、定量的管理を行う意味は薄れてしまう。チームレベルから組織レベルまでの範囲に応じた、開発プロセスの改善や品質向上につながるアクションの実践が求められる。

そして第三に、ソフトウェア工学研究コミュニティとの交流である。本稿は、野中・亀井・大平の3名による記事だが、このほかにも国内にはソフトウェア定量的管理にかかわる研究を行っている研究者がいる。もちろん、国外にも多数いる。こうした研究者らとの交流は、国内であれば、例えば情報処理学会ソフトウェア工学研究会のソフトウェアエンジニアリングシンポジウムなどに参加すれば意見交換の機会が得られる。逆に、学術研究に携わる研究者が、例えばIPA/SECの活動や日本科学技術連盟のソフトウェア品質シンポジウムに出向くなどして、産業界と交流するための場作りが求められる。まずは、参照可能な論文や発表資料を相互に学び合うところから始めると良いだろう。

7 おわりに

本稿では、産業界でのソフトウェアの定量的管理に参考となり、役立つことが期待される学術研究の事例について、筆者らがかかわったものを中心に紹介した。本稿の内容が、産業界における学術研究への関心につながり、ソフトウェア定量的管理更なる進化への足がかりとなり、産官学での交流がよりさかんになることを期待する。

参考文献

- [1] Shimagaki, J., Kamei, Y., McIntosh, S. Hassan, A. E. and Ubayashi, N., A Study of the Quality-Impacting Practices of Modern Code Review at Sony Mobile, Proceedings of International Conference on Software Engineering (ICSE2016), Software Engineering in Practice (SEIP), pp. 212-221 (2016).
- [2] 柏祐太郎, 大平雅雄, 阿萬裕久, 亀井靖高, 大規模OSS開発における不具合修正時間の短縮化を目的としたバグトライージ手法, 情報処理学会論文誌, Vol. 56, No. 2, pp. 669-681 (2015).
- [3] 柳田礼子, 野中誠, 菅田直美, CMMI成熟度レベル別に見たソフトウェア品質の良否にかかわる要因の複合的分析, SEC journal, Vol. 13, No. 1, pp. 8-15. (2017).
- [4] McIntosh, S. and Kamei, Y., Are Fix-Inducing Changes a Moving Target? A Longitudinal Case Study of Just-In-Time Defect Prediction, IEEE Transactions on Software Engineering. (To Appear)
- [5] 天壽聡介, モデルに基づいた工数見積もりにおける重み付きMoving Window 法の評価, 情報処理学会ソフトウェア工学研究報告, Vol. 2012, No. 24, pp. 1-8 (2012).

Embedded Technology West 2017 (ETWest2017) 出展報告

SECプロモーショングループ主任 荒川 明夫

IPA/SECは、2017年7月12日、13日の2日間、グランフロント大阪コングレコンベンションセンターにて開催された「Embedded Technology West 2017／組込み総合技術展 関西 (ETWest2017)」に出展した。また、隣接会場では、安全解析に関する取り組みをテーマとした「特別講演」、IoTにおける安全・安心に向けた取り組みなどを紹介する「IPAセミナー」をそれぞれ実施した。

1 展示会概要

Embedded Technology West (ETWest)とは、一般社団法人組込みシステム技術協会 (JASA) が主催する西日本で唯一となる最新テクノロジーの専門技術展である。組込みシステム開発にかかわる技術者や開発者向けに最新技術などの情報を発信しており、今回で12回目の開催となる。また、昨年より“つながる”技術の最先端を網羅する「IoT Technology West」が併催され、成長分野を支える先端技術にフォーカスした展示会として、開催された。

2 出展概要



IPA/SECは、事業成果の普及・啓発を目的として、本展示会に毎年出展している。

本年は、IPA展示ブースにてプレゼンを2日間22回、隣接会場ではIPAセミナーを4回、特別講演を2回、計28セッションを実施した。

また、SECの事業を中心にIPAで取り組んでいる組込みやIoTに関連する事業の資料配布を実施し、各事業成果を関西圏の方々に説明し、普及啓発を図った。

3 IPA展示ブース

本出展では、展示ブースを2分割し、ステージ(ブースプレゼン・SEC先端技術入門ゼミ、共通)とパネル展示エリアにて運営を行った。



展示コーナーでは、「つながる世界の開発指針」をはじめ、IoT時代のソフトウェア開発におけるポイントをまとめた事業成果展示や障害情報の共有促進、「STAMP」、システム再構築時の上流工程強化策、コーディング作法ガイドなどのパネルを配置、担当職員より来場者に説明を行った。

とくに、「つながる世界の開発指針」や「STAMP」への関心が高く、関連したステージを聴講する来場者の姿が多く見受けられた。

4 プレゼンステージ

今回のステージは、これまで個別で実施していたブースプレゼン（IPA職員や関係者による事業成果の説明）とSEC先端技術入門ゼミ（外部有識者による最新技術動向や事例などの紹介）をかけ合わせた、コラボレーション企画とした。

各セッションでは、個別にテーマを掲げ、「コードレビュー活用」「STAMP/STPA入門」「開発指針実践」「安心・安全入門」「STAMP/STPA事例」「利用時品質入門」を実施した。

いずれも多くの方に聴講いただき、2日間で1,286名の参加があった。

参加者からは、「短時間ながらも理解しやすい」「今まで知らなかった切り口で、参考になった」「IoTの安全の考え方に興味があったので参考にしたい」という意見が寄せられた。



5 IPAセミナー／特別講演

展示フロアに隣接する会場では、会期2日目の13日（木）にIPAセミナーを4部構成で開催し、延べ105名の参加があった。



今回のIPAセミナーは、理事長の富田より「頼れるIT社会の実現を目指して」と題して、IPA全体の事業や社会環境について、講演した。次にSEC所長の松本より「IoT時代に向けたシステム思考のアプローチ」と題し、SECの取り組みを紹介した。後半は、国立大学法人名古屋大学の森崎准教授より「『つながる世界の開発指針』の実践に向けた手引きの紹介」を、公立大学法人会津大学の兼本名誉教授からは、「システム理論に基づく新しい安全解析法STAMP/STPAの実践」を説明いただいた。

また、別会場では、各日特別講演と題して、日産自動車株式会社の岡田氏より「自動車機能安全開発へのSTAMP/STPAの適用事例」を、国立大学法人名古屋大学の高田教授からは、「IoT時代の到来に向けた安全・安心な社会の構築への取り組み」を講演いただき、SEC事業成果普及に協力いただいた。

6 ETWest2017を振り返って

今回の出展は、展示スペースが例年よりもコンパクトになり、動員が懸念されていたが、過去の出展の中で、最も多くの方にIPA展示ブースに足をお運びいただいた。展示会場で来場者の方から直接多くのご意見をいただいた。この貴重なご意見を次回以降の出展や今後の事業の参考としていきたい。

7 謝辞

IPAセミナー、IPA特別講演、展示ブースステージにご登壇いただいた外部講師の皆様、並びに関係団体の皆様には、本展示会出展に際し、多大なるご支援を賜りました。ここに深謝いたします。

- ▶ ETWest2017 IPA/SEC Webサイト
<https://www.ipa.go.jp/sec/events/20170712.html>
- IPAセミナー・ブースステージの講演資料がダウンロードできます

INCOSE国際シンポジウム2017 開催模様

SEC journal編集部

2017年7月15日から20日にかけて、オーストラリアのアデレード市において第27回INCOSE国際シンポジウムが開催された。システムズエンジニアリングに関しては最大規模のカンファレンスであり、今回は600名ほどの参加者があった。本会議は4日間、6トラックの並行セッションであり、それ以外にチュートリアルやワークショップも多数併設された。

INCOSE (International Council on Systems Engineering) は、システムを成功裏に実現できるようにする学際的な原理や実務慣行を整備し普及するために、1990年に設立された非営利組織である。システムズエンジニアリングを展開、発展させて、社会的及び技術的に複雑な問題に取り組むことをミッションとしている。2015年末の会員数が11,134名で、毎年着実に増加している。

このシンポジウムでは、エネルギー、自動車、運輸交通、医療、航空宇宙、防衛など、多様な産業や政府機関に所属する多数の専門家たちが一堂に会し、知識や経験、アイデアを交換する。社会を支える製品、サービスの企画、開発、生産、配備、運用にかかわる複雑な諸問題に対して、「システム」という視点で概念を整理し、知識を蓄積し、分野横断的なエンジニアとして協力する場になっている。

システムズエンジニアリングは、ベル研究所が発祥と言われており、アメリカの国防総省や航空宇宙産業の中で発展してきた。冷戦終結後、情報通信技術の発展も相まって、大規模で複雑なシステムが多数実現されるようになり、INCOSEの設立と拡大に象徴されるように、システムズエンジニアリングは新たな展開、発展を見せることになった。

とくに近年はIoT化や、AI技術の展開、自動運転実用化に向けての取り組みなど、これまで以上に複雑で大規模なシステムが構築されつつある。このような挑戦的な領域では、多様な専門家が協力し、社会的観点や人的要因も加味した対応が求められ、システムズエンジニアリングの重要性が増大している。システムの設計上の課題だけでなく、運用や改良、普及拡大、廃棄などに起因する課題を取り上げ、情報セキュリティや安全性なども考慮する必要がある。加えて国際規格の制定や、人材育成などへの取り組みも重要になっている。

今回のシンポジウムでも、このような幅広い課題を取り上げており、4件の基調講演、140件ほどの発表に加え、多数のパネルセッションが行われた。3日目の基調講演者、CMU/SEI^{*1}のPaul Nelsen所長は、IoT/AI/ロボットなどが発展した状況における「自律性 (Autonomy)」の重要性を指摘した。膨大な数の機器が接続された状況では、すべての機能やリンクが想定通り動作するとは限らない。それでも高い信頼性や安全性が求められ、自律分散型の設計が必要になってくる。谷口智彦内閣官房参与の基調講演では、新幹線の安全性、拡張性、正確性、経済性をもたらした設計上の重要決定を解説した。

今年2月に出版された「Integrating Program Management and Systems Engineering」について、INCOSE、PMI^{*2}、MIT^{*3}の3者提携の成果として報告されていた。当事者の発表や討議だけでなく、全体会議での紹介、展示スペースにおけるPMIによる解説などが行われていた。システムズエンジニアリングとプログラムマネジメントの知識領域は大きく重なっており、現実場面では両者の統合適用は必然であろう。

システムズエンジニアリングの適用領域の広さを反映し、発表のテーマは多岐にわたった。包装容器メーカーの業務改善、海底油田施設のコストダウン、複雑さに着目した安全性指標、アフリカでの井戸掘削のリスク管理、セウォル号の事故原因分析、汎用人工衛星コンポーネントを用いた人工衛星設計の最適化、軍の機構改革、鉱山用発破装置開発へのシステムズエンジニアリング適用など、見方によっては脈絡がないが、目的指向と全体俯瞰、複雑な対象のモデル化、多様な専門分野の統合など、システムズエンジニアリングのアプローチを利用して問題解決に取り組んだ事例が多数報告された。

一方、問題解決に取り組む手法やアプローチとして、システム思考、システム科学、MBSE^{※4}、モデリング、シミュレーション、システムダイナミクス、プロセス、プロダクトライン、リーダーシップ、アジャイルなどが取り上げられ、要件、アーキテクチャ、V&V^{※5}、調達など、ライフサイクルプロセスごとのテーマでも多くの発表が行われた。加えて、セキュリティ、安全性、レジリエンス、人的要素など、関連する特性に着目した議論も行われた。

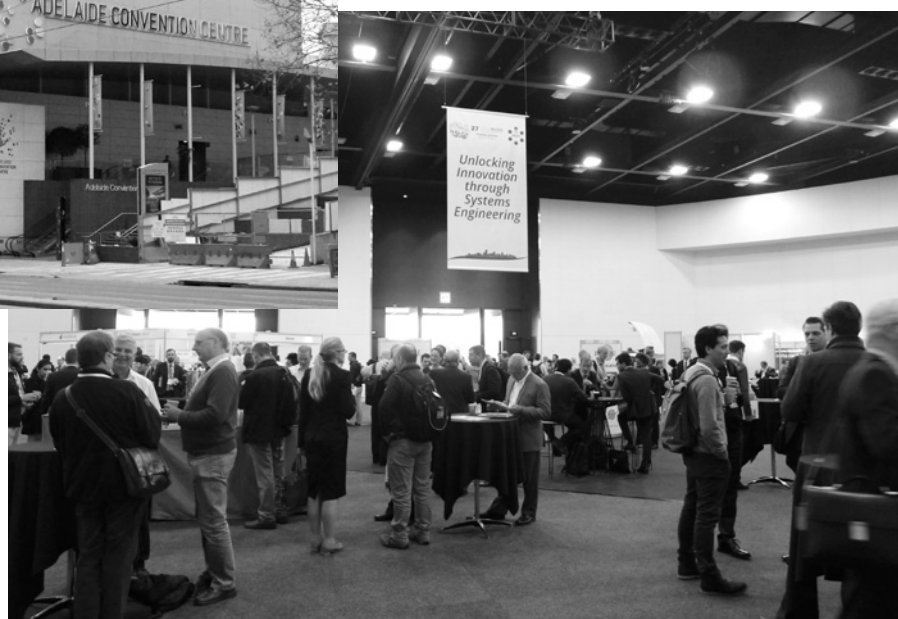
本会議の発表だけではなく、ワーキンググループや小規模なディスカッション、産業分野ごとのパネルや集会も

行われ、INCOSEのアクティブメンバが対面の機会を利用して打ち合わせを行ったり、活動仲間を勧誘したりする場でもあった。バンケット、レセプションはもちろん、ランチやコーヒブレイクなど、交流の機会が多数提供され、この分野の重鎮とも直接対話できることが、シンポジウムの重要な機能であった。

INCOSEは、システムズエンジニアリングで実施されるプロセス、アクティビティをまとめたSystems Engineering Handbookを公開している。不確実性をはらみ、大規模で複雑なシステムを構築、運用、発展させていくノウハウを体系的に整理する一方、その応用事例を共有、交換し、優れた実践の表彰も行っている。システムズエンジニアリングは「学際的」であると同時に、多くのエンジニアリングの交流のインフラである。INCOSE国際シンポジウムは、これからの時代を担う多様なエンジニアを啓発する貴重な機会を提供している。

脚注

- ※1 Carnegie Mellon University/Software Engineering Institute
- ※2 Project Management Institute
- ※3 Massachusetts Institute of Technology
- ※4 Model Based Systems Engineering
- ※5 Verification and Validation



ソフトウェア高信頼化センターの 地域支援活動について

IT利活用を支えるソフトウェア技術の地方への展開状況報告

SEC調査役 久保 忠伴 SEC研究員 藤原 由起子 SEC研究員 小長谷 義浩

少子高齢化、人口減少による対策として、地域経済の活性化のための地方支援策が活発に実施されている。SECでは、IoT時代における安全・安心なIT構築とその利活用に向けた地域支援活動を推進している。IT化を進める際のポイントや注意点、成功事例・失敗事例などを全国のIT関係者と共有するために、地域の団体との連携による共催セミナーの開催や講師派遣などの支援活動を行っている。ここでは、地域に向けた支援活動の方針や2017年度の計画と実施状況を報告する。

1 地域の状況と支援活動の方針

地域・中小企業の状況は、中小企業庁が課題を提示^{*1}、総務省はIoT時代に即した地域モデルを提示^{*2}している。これらを踏まえて、SECは、IT化による地域経済の活性化に向けて以下の方針で支援活動を推進している。

<基本方針>

夢のある安全・安心なIT社会の実現に向け、地域による、地域のための、地域に即した活動を支援

2 支援活動の戦略(3本の矢)

SECは、情報システムの信頼性向上を担う政策実施機関として、関連情報の収集・分析を行っている。更に、産学連携のハブとしてそこで得られた知見の共有・普及活動を行っている。地域支援に関しては、長年培ってきたSECの事業成果と知見を活用し、地域のイノベーションの実現を支援するために以下を地域支援活動戦略の3本の矢として推進している。

- ① 地域団体と連携した支援活動
(地域のIT関連団体、行政機関との連携)
- ② 地域に即したSEC技術資産の提供
(地域の特性、ニーズに応じた活動)
- ③ エリア戦略による計画的な支援活動
(地域の意欲を考慮して計画的に実施)

以下にその内容を解説する。

2.1 地域団体と連携した支援活動

地域主導の活動を行うために地方経済産業局・自治体、IPA関連団体、地域団体をパートナーとして以下の3つの観点から共創の実現を目指している。

- ① 政策連携：地方経済産業局や自治体が進めている政策課題解決策とSECの活動がシナジーを生み出すような活動連携
- ② IPA関連団体との連携：ITコーディネータ協会(ITCA)、組込みシステム技術協会(JASA)などとの連携
- ③ 地域団体との連携：地域活動の主体である団体(各県の情報サービス産業協会やNPO法人、協議会など)との連携

2.2 地域に即したSEC技術資産の提供

地域に普及するSECの技術資産としては、SECが現在取り組んでいるIoTなどの技術分野を中心に選定している(表1参照)。具体的には、IoT、システムズエンジニアリング、上流工程の強化、安全性解析手法(STAMP/STPA)、情報処理システムの障害対策、その他(見える化手法、定量的プロジェクト管理、ソフトウェアライフサイクルプロセス、プロセス改善手法などのITシステムの構築技術)があり、地域のニーズに即したテーマを組み合わせて提供している。

表1 SECが提供するIT技術資産

技術分野	内容
つながる世界 (IoT)	「つながる世界の開発指針」ほかIoTの安全・安心を確保するための考え方
上流工程の強化 (要件定義、再構築)	「ユーザーのための要件定義ガイド」ほかユーザー企業が抜け漏れのない要件定義を行うためのガイド
情報処理システムの 障害対策	「情報処理システム高信頼化教訓集」システムの障害事例の分析、体系化から得られる教訓
安全性解析手法 STAMP/STPA	「はじめてのSTAMP/STPA」、「はじめてのSTAMP/STPA 実践編」複雑なシステムの安全性評価に対応した新しい手法
システムズエンジニア リング	「経営者のためのシステムズエンジニアリングの薦め」IoT時代のビジネスに必要な新しい開発アプローチの必要性和有用性の説明
その他	定量的プロジェクト管理などのITシステムの構築技術全般

表1に関連した内容を、SEC BOOKSや報告書として公開している。
<https://www.ipa.go.jp/sec/publish/index.html>

2.3 エリア戦略による計画的な支援活動

地域支援を具体的に実施するにあたり、支援活動の効率性と効果性を考慮して、地域を8ブロック(北海道、東北、関東、中部、近畿、中国、四国、九州・沖縄)に分けて、以下の3点と地域の特性や地域産業の強みや弱みと事業機会に関するポートフォリオを意識した活動を進めている。

表2 今年度の実施状況

地域区分	開催地	区分	連携団体	開催時期	技術分野
北海道	札幌	講師派遣	札幌市	5月	情報処理システムの障害対策
		共催セミナー	北海道ソフトウェア技術開発機構、北海道経済産業局	9月	つながる世界 (IoT) と上流工程の強化(要件定義、再構築)
		講師派遣	北海道ソフトウェア技術開発機構	10月	上流工程の強化(要件定義、再構築)
東北	岩手	講師派遣	岩手県情報サービス産業協会	10月	上流工程の強化(要件定義、再構築)
関東	埼玉	講師派遣	埼玉ITコーディネータ	5月	つながる世界 (IoT)
	神奈川	講師派遣	横浜市、横浜企業経営支援財団	6月	つながる世界 (IoT)
	静岡	講師派遣	静岡大学 情報部、組込みソフトウェア技術コンソーシアム(HEPT)	10月	安全性解析手法 STAMP/STPA
近畿	大阪	講師派遣	ITC近畿会	7月	上流工程の強化(要件定義、再構築)
		講師派遣	関西情報センター、組込みシステム産業振興機構	10月	つながる世界 (IoT)
	京都	共催セミナー	電子情報技術産業協会(JEITA)	9月	安全性解析手法 STAMP/STPA
中国	鳥取	講師派遣	鳥取県情報産業協会	9月	つながる世界 (IoT) と定量的プロジェクト管理
九州・沖縄	大分	共催セミナー	ハイパーネットワーク社会研究所	5月	つながる世界 (IoT)
	福岡	講師派遣	九州組込みソフトウェアコンソーシアム(QUEST)	8月	つながる世界 (IoT)
	熊本	講師派遣	熊本県情報サービス産業協会	9月	情報処理システムの障害対策

- ① 意欲的だが、取り組みが計画段階の地域・団体
- ② 意欲的にIT化の取り組みを始めた地域・団体
- ③ 自主的に活発な活動を行っている地域・団体

3 2017年度の地域支援状況

3.1 地域との連携によるセミナー実施概要

2017年度は、IT化への積極的な取り組みを行っている5地区(北海道、東北、関東、近畿、九州・沖縄)を中心として、計画的に実施している。また、5地区以外の団体から要請があった場合には、個別に講師を派遣して対応している。実施状況を表2に示す。

3.2 共催セミナーの実施例

共催セミナーのテーマについては、地域ごとの特色を考慮して地域の共催団体と協力し、企画・開催している。実施結果を以下に示す。

(1) 九州地区(大分県) 2017年5月 開催

- テーマ：「デジタル時代に向けた大分イノベーション～OITA4.0の実現に向けて～」
- 技術分野：つながる世界 (IoT)
- 実施形態：共催セミナー
- 共催団体：ハイパーネットワーク社会研究所
- 参加者数：130名

「つながる世界の開発指針」やIoT事例の紹介、大分イノベーション実現への課題を討議した地元の方々とのパネルディスカッションを通して、地元企業の皆様のIoTへの関心を高めることができた。



図1 大分での会場風景とパネルディスカッションの様子

(2) 北海道地区(札幌) 2017年9月 開催

- テーマ：「デジタルイノベーションで創るスマートな北海道～経営者に必須な「IoT導入の指針とつながる世界の要件定義の重要性」の解説～」
- 技術分野：つながる世界 (IoT) と上流工程の強化
- 実施形態：共催セミナー
- 共催団体：北海道ソフトウェア技術開発機構、北海道経済産業局
- 参加者数：67名

つながる世界 (IoT) の留意点や上流工程の強化の重要性を解説し、新しいサービスやビジネスに取り組む際の安全・安心への関心を高めることができた。また、札幌市IoTイノベーション推進コンソーシアムの活動状況、北海道内地元企業のIoT事例の講演を通して、参加者の皆様へIoT等に関する道内の情報を紹介することができた。



図2 北海道セミナーの様子

なお、共催セミナーの詳細は下記Webサイトで公開している。
<https://sec.ipa.go.jp/seminar/index.html>

4 今後の課題

地域・中小企業に向けた支援活動は、点から線へ、線から面への展開戦略に沿った活動が効果を生むと考えられるため、地域団体との連携を重視している。また、今年度は、意欲的な活動を推進している地区を中心に活動している。今後、ほかの地区についても、ニーズの度合いや、地域の実情を探りながら支援活動を検討していきたい。

本稿の内容に関するお問い合わせ、地域IT活性化の検討についてのご質問・ご相談などは、下記までご連絡ください。(問合せ先：sec-pr@ipa.go.jp)

脚注

- ※1 <http://www.chusho.meti.go.jp/koukai/kenkyukai/smartsme/2017/170606smartsme02A.pdf>
- ※2 http://www.soumu.go.jp/main_content/000487304.pdf

ハドソン川の奇跡

IPA顧問 松田 晃一

もう一年ほど前になりますが、トム・ハンクス主演の「ハドソン川の奇跡」という映画が公開されました。ご覧になった方も多いと思いますが、実際に起きた飛行機事故の後日談を映画化したものです。

■ ハイテク機も野鳥の一撃で制御不能に

事故は2009年1月にニューヨーク・マンハッタンの上空で起こりました。離陸直後のUSエアウェイズ機にバードストライクが発生し全エンジンが停止。最先端技術の塊であっても、野鳥の一撃で一瞬にして破壊され危機的な状況に追い込まれるのですが、機長の冷静な決断と沈着な行動によって、ハドソン川に不時着。乗員乗客155名はもちろん、ほかに一人の犠牲者も出さず、奇跡の生還を遂げた事故。ハドソン川の奇跡として全世界に報道されたのでご記憶の方も多いのではないでしょうか。いかに最先端技術で造られた航空機であっても、人工物は自然には勝てないことをあらためて思い知らされた出来事でした。最後は人間の知恵と勇気が惨事を救ったわけで、人間の力の偉大さ、機械と人間の調和の重要性を感じたものです。

■ パイロットの判断は正しかったか？

さて、映画はここから始まります。事故の後、国家運輸安全局による事故原因の調査が行われ、その中で果たしてパイロットの判断が適切だったのか、もっと良い選択肢があったのではないかと、との疑問が生じてきます。すなわち、ハドソン川への不時着という方法ではなく、近くの設定の整った空港へ引き返して着陸することが可能であったのではないかと、ということです。奇跡の生還を果たした立役者、英雄として称賛された乗員達が一転厳しい追及を受けることとなります。

もし、空港へ引き返す決断をして、うまくたどり着くことができなかった場合には、ニューヨークの市街地へ墜落、乗客はもちろんその他の市民多数を巻き込む大惨事になる可能性があります。一方、厳寒のハドソン川への不時着水は、市民の犠牲は避けられるかもしれませんが、乗客の命を大きな危険にさらすこととなります。さて、どちらを選択すべきだったのか？正に究極の選択です。映画では、フライトシミュレータによって、空港へ引き返すと無事に着陸できることが示されるのですが…。

結末はここでは明かさないことにします。興味がある方は、どうぞ映画をご覧になって確かめてください。

■ 究極の選択は想定外か？

さて、今話題のAIにこのような究極の選択を委ねるとしたらどうなるのでしょうか？もちろん、航空機の操縦が完全にAIに任されてしまうことは、当分はないのですが、例えば実用化に向かって開発が進められている車の自動運転のケースはどうでしょうか。ブレーキやエンジンが故障したときに、対向してくる子供たちが乗ったスクールバスとの衝突を避けて、歩行者が歩く歩道にハンドルを切るのでしょうか？どちらを選んでも犠牲者が出るかもしれない究極の選択について、自動運転を設計している技術者はどのように考えて答えを出そうとしているのでしょうか？

福島原発事故の後で、想定外という言葉が繰り返し語られました。上のような究極の選択を迫られることは、想定外なのでしょうか？あるいは、想定はされるが発生確率は非常に低いとして設計から除外するのでしょうか？それとも、対策に本気で取り組もうとすると投資額が巨大になってしまうから、そんなことはほとんど起こらないだろうと楽観的に割り切るのでしょうか？

■ 技術者の社会的責任

「ちょっとでも可能性がある、そういうものを全部組み合わせていったらモノなんて作れない」ということも一面事実です。そのために、設計には一定の割り切りが行われます。しかし、設計上で割り切ったことと、それが実際には起こらないことは明らかに違います。設計上で想定した条件と現実との間には必ずギャップがあります。そのギャップをはっきりと認識して対応策を準備しておく必要があります。事は技術論ではありません。開発者だけに委ねて良い問題ではなく、法律家や倫理学者など多様な人々が議論に加わって社会的なコンセンサスを探るしか方法はないと思います。そのためには、技術者は問題が何であるかを異なる専門分野の人たちに分かりやすく説明し、共に協力して問題の解決策を探る必要があります。技術者が自らの狭い専門分野に閉じこもるのではなく、異なる分野の専門家たちと連携し、一般の人たちとも理解を深めるなど、問題の解決策を探るべく、広く社会と向き合っていく姿勢が今まで以上に求められるのではないのでしょうか。



野口 悠紀雄 著

ISBN : 978-4-532-35719-1
 日本経済新聞出版社
 四六判・360ページ
 定価1,800円(税抜)
 2017年1月18日刊

ブロックチェーン革命

～分散自律型社会の実現～

本書は、「超整理法」「超文章法」などで有名な一橋大学名誉教授 野口悠紀雄氏の著作である。

「ブロックチェーン技術」を、著者の言葉を借りてひと言で説明すると以下ようになる。

“私は、『仮想通貨革命』の「はじめに」で、「これは反乱ではありません。これは革命です」という言葉を引用した。インターネットが革命であったように、ブロックチェーンも革命だ。それはパラダイムの変革をもたらす。”

ブロックチェーンの本質

従来のインターネットでできなかったことが、2つある。第1は、貨幣など経済的に価値あるものを送ること。そして第2は、信頼性を確立することだ。ブロックチェーン技術は、経済的価値をインターネットで送ることができる。ブロックチェーンは「経済的価値のインターネット」だと言われている。

ブロックチェーンの特徴と応用

ブロックチェーンの適用対象は通貨に限らない。証券業界への導入も試みられている。これ以外の応用分野は多く、IoTや、サプライチェーン、そして医療や教育などにも及んでおり、その広がりや課題が解説されている。

また、IoTに関して重要なことが指摘されている。現在のIoTには経済的視点が欠落しており、IoTにはブロックチェーンの技術が不可欠であるということである。そして、分散型自律組織や分散市場が既に誕生しつつあり、ブロックチェーンの技術は、つながる世界の中で経済を変え、企業や組織を変え、社会を変える技術である。(久保 忠伴)



牧野 貴樹 他 編著

ISBN : 978-4627880313
 森北出版株式会社刊
 A5判・320ページ
 定価4,200円(税抜)
 2016年10月27日刊

これからの強化学習

「ポストビッグデータ」や「ポスト機械学習」の技術と言われる「強化学習」。最近よく見聞きするキーワードである。強化学習の提唱者と言われているRichard S. SuttonのReinforcement Learningが発行されたのは1998年であり、新しい考えではないのだが、AIや機械学習がそうであるように最近クローズアップされている。本書は多方面の専門家による共著であり、最新の研究動向をキャッチアップすると共に応用の可能性を俯瞰することができる。

未知の環境の中を探索しながら期待報酬を最大化するためのエージェントの行動原理が強化学習である。十分なデータが与えられることが前提とされる機械学習とは異なり、足りないデータを集めながら最適化していく。データを利用するだけでなく、必要なデータを自ら探索しに行くのが強化学習の特徴である。本書からは、探索任せで必要なデータを収集してくる夢の技術という印象を持つかもしれない。しかし、データの収集には、通常多くのコストと時間がかかるため、探索(explore)と利用(exploit)のトレードオフを考慮すべきであることを忘れてはいけないはずだ。

本書で解説されているアルゴリズムは、私にとって大変難しいものであったが、「不確かなときは楽観的に」とか「想定し得る環境の中から最も都合の良い環境を選ぶ」など、そのような行動を取るエージェントには人間味を感じ、親しみを覚えた。(遠藤 秀則)

編集後記

今号は定量的管理を特集しています。ソフトウェア・エンジニアリングの実践には、定量データの収集・分析・応用が基本であるとの考えから、SECでは2004年の設立当初より定量データを収集し、その成果である「ソフトウェア開発データ白書」を継続的に発行してきました。「ソフトウェア開発データ白書」などを用いて、定量的管理を実践されている皆様も多いと思いますが、今号の特集をご覧いただき、定量的管理に関する各界の最新の取り組みに触れていただくことにより、自社への活用のための新たなヒントが得られればうれしく思います。

本誌が届く頃、年末のイベントで忙しい方も多いかと思います。お体ご自愛ください。
(編集長)

編集部より

次世代のソフトウェア・エンジニアリングに関してなど、忌憚のないご意見をお待ちしております。下記のFAXまたはメールにてお気軽にお寄せください。

SEC journal 編集部 FAX : 03-5978-7517
e-mail : sec-journal_customer@ipa.go.jp

SEC journal 編集委員会

編集委員長 遠藤 秀則
編集委員 (50音順)
荒川 明夫
石橋 正行
江野村 亮輔
日下 保裕
佐藤 康彦
中尾 昌善
中谷 好寿
長谷川 佳奈子
三原 幸博
室 修治
山下 博之



冬のムーミン谷 (埼玉県飯能市)

撮影 : Y.Konagaya

SEC journal 第13巻 第3号 (通巻54号) 2017年12月1日発行

©独立行政法人情報処理推進機構 2017

編集兼発行人 独立行政法人情報処理推進機構
技術本部 ソフトウェア高信頼化センター
所長 松本 隆明
〒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16階
Tel : 03-5978-7543 Fax : 03-5978-7517
URL : <https://www.ipa.go.jp/sec/> e-mail : sec-journal_customer@ipa.go.jp

※本誌は「著作権法」によって、著作権等の権利が保護されている著作物です。

※本誌に掲載されている会社名・製品名は、一般に各社の商標または登録商標です。

SEC journal 論文募集

独立行政法人情報処理推進機構（IPA） 技術本部 ソフトウェア高信頼化センターでは、下記の内容で論文を募集しています。

論文テーマ

- ・ソフトウェア開発現場のソフトウェア・エンジニアリングをメインテーマとした実証論文または先導的な論文
- ・ソフトウェアが経済社会にもたらす革新的効果に関する実証論文

論文分野

品質向上・高品質化技術、レビュー・インスペクション手法、コーディング手法、テスト/検証技術、要求獲得・分析技術、ユーザビリティ技術、プロジェクト・マネジメント技術、設計手法・設計言語、支援ツール・開発環境、技術者スキル標準、キャリア開発、技術者教育、人材育成、組織経営、イノベーション

応募要項

締切り：1月・4月・7月・11月 各月末日

査読結果：締切り後、約1カ月で通知。「採録」と判定された論文はSEC journalに掲載されます。

応募方法：投稿は随時受付けております。応募様式など詳しくはHPをご覧ください。

<https://www.ipa.go.jp/sec/secjournal/papers.html>

SEC journal 論文賞

毎年「採録」された論文を対象に審査し、優秀論文にはSEC journal 論文賞として最優秀賞、優秀賞、所長賞を副賞と併せて贈呈します。

IoT時代に活躍する【組み込みシステムの腕利きエンジニア】を目指す！

国家試験 エンベデッドシステムスペシャリスト試験

高度な実践能力の証明に！

- ▶ 身近な場面を想定した出題を通して、最適な組み込みシステム実現のために必要となる高度な実践能力（レベル4）を問います。

レベル4の定義：専門分野において、自らのスキルの活用によって、独力で業務上の課題の発見と解決をリードするレベル。

技術要素

プロセッサ、メモリ、バス、計測・制御、リアルタイム OS、プラットフォーム、電気・電子回路、ネットワーク、セキュリティ

開発技術

- ・要求分析の実行とレビュー
- ・設計の実行とレビュー
- ・テストの実行とレビュー

管理技術

- ・開発環境マネジメント
- ・知財マネジメント
- ・構成管理、変更管理

- ▶ 近年の試験では、「無線通信ネットワークを使用した安全運転支援システム」、「3次元複写機」、「通信機能をもつ電子血圧計を用いた健康管理システム」、「非接触型ICカードを使用した入退場ゲートシステム」などのテーマを出題しました。
- ▶ 自動車、家電、モバイル機器などに搭載する組み込みシステムや重要インフラの制御システムを、ハードウェアとソフトウェアを適切に組み合わせて構築し、求められる機能・性能・品質・セキュリティなどを実現できる組み込みエンジニアを目指す方に最適です。

試験概要

【試験区分】 エンベデッドシステムスペシャリスト試験（情報処理技術者試験 高度試験の1区分として実施）

【日 時】 年1回の実施（毎年4月第3日曜日）

【申込受付】 毎年1月中旬から2月下旬（予定）までWEB・郵送で申込み受付

詳しくは、Webページをご覧ください。<https://www.jitec.ipa.go.jp/index.html>

試験概要の最新情報、過去問題、活用事例などをご紹介します。

IPA Better Life with IT

SEC journal No.51
第13巻第3号(通巻54号)
2017年12月1日発行

©独立行政法人情報処理推進機構

ISSN 1349-8622

