

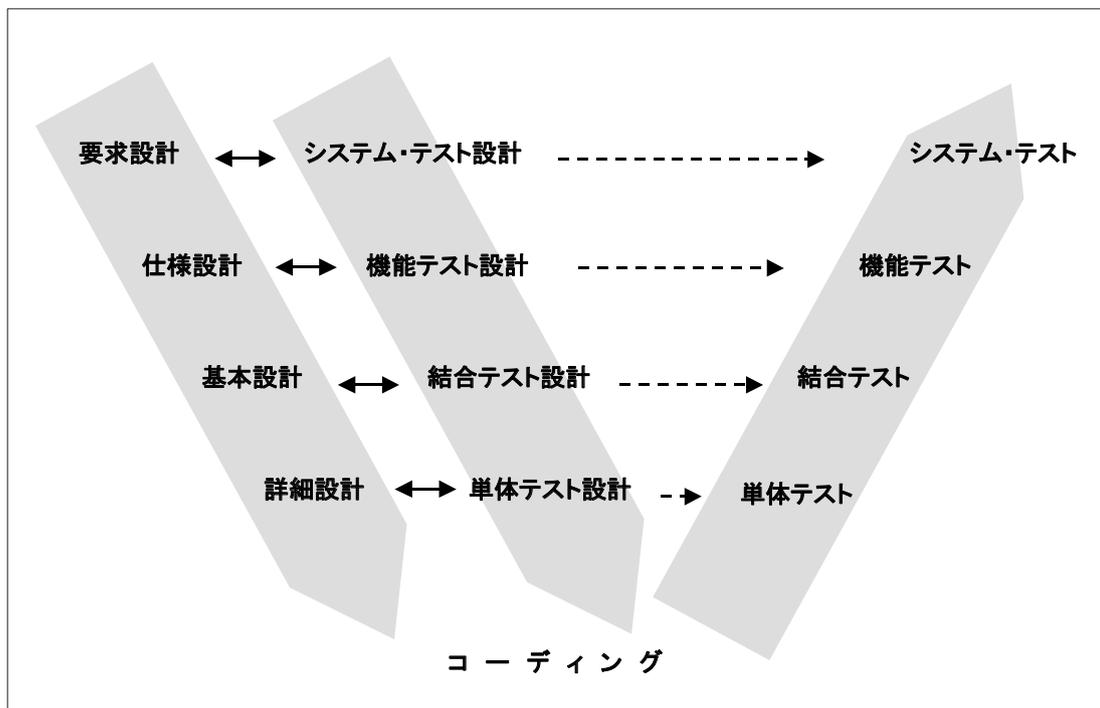
ソフトウェア・テストの技術動向と課題

近年、ソフトウェア・テストは、リスクマネジメントとして、あるいは、品質保証活動の一環として強く意識されるようになった。ソフトウェアに起因する事故は、社会問題や大きな経済損失を招くことすらあるが、それらは、極端な言い方をすれば、テストが不十分であったために生じたと言ってもよい。自動車を含めたほとんどの機械は、今や組込システムと呼ばれるソフトウェアで動くようになっている。このようなテストについての需要の増大にこたえる形で、ソフトウェア・テストを専門とする企業や組織も出現している。

テストの手法も、図表のように、上流工程を取り込むようになり、静的テストの重視や自動化などが行われている。

ここでは、このような技術面の改良だけでなく、1) 事故調査委員会の設置による経験知の共有、2) 第三者機関によるソフトウェア・テストの監査、3) 法制度の整備、4) 用語・記述・表現の研究および基本知識の普及を提案する。このような取り組みは、長期に見れば、日本製品を含めた日本のシステムの品質を見直し、品質に関わる作業を企業内部の活動にとどめず、広く世の中に提供できる活動、いわば、「品質」産業とでも呼べるような活動にまで高まる可能性を秘めている。

開発とテストの W モデル (最近のモデル)



科学技術動向研究センターにて作成

に着手して、実際に採用する標準規格やガイドラインについても世界的に告知する活動を行うための組織である。

本稿では、ソフトウェア・テストの位置付けを示し、特に、最近、国民の安全・安心に大きく関わるようになった「組込システム」と呼ばれるソフトウェアの技術分野において、テストがいかに重要かと

いうことを述べる。次に、テスト技術の現状のうち、特に注目すべきものを紹介し、また、ソフトウェア・テスト産業に関わる最近の動向として、ソフトウェア・テスト専門企業の出現について述べる。さらに、ソフトウェア・テストの抱える最大の課題として、ソフトウェア・テスト設計者や技術者の置かれている状況と、その教育の

現状を論じる。ソフトウェア・テスト技術を核とした品質保証産業の育成や、その前提としてのソフトウェア品質に対する評価や制度の整備、また、一般に分かりやすくするための工夫についても言及し、このような取り組みが、長期に見て、「品質」産業と呼べるような活動につながり得ることを指摘する。

2 ソフトウェア構築全般における テストの位置付け — デバッグとの相違点 —

ソフトウェア工学の中で、ソフトウェア・テストの占める位置は、これまでは、決して大きいとは言えなかった。それは、以下に述べるようなデバッグ作業との相違点を認識していなかった、ということに起因していると考えられる。

ソフトウェアを構成するプログラムが思ったように動かないという困った現象自体は、プログラムが作られる初期段階から見受けられる。不具合の生じている個所をバグと呼ぶことから、そのような不具合を取り除く作業をデバッグ(debug)と呼んでいる。ソフトウェア・テストは、デバッグに係るが、デバッグとは違う独立した作業として取り扱わなければいけない。このことは、ソフトウェア・テストの古典と言われているMyersの「ソフトウェア・テストの

技法」(1979年出版)でも強調されていることの一つである⁷⁾。デバッグとテストの相違点は、図表1のようにまとめることができる。

デバッグもテストも品質向上という大きな目標は共通なのだが、デバッグの目的はプログラムのバグを修正することであるのに対し、テストの目的はシステムを含めた全体の不具合を発見することである。両者は全く方向性が異なることに注意しなければならない。

例えば、プログラム自体にバグがなくても、テストで不具合を指摘されることがあり得る。デバッグにおいては、与えられた仕様を満たすプログラムであるかどうかの問題になり、テストにおいては利用者にとっての不都合が問題になるからである。

このことを、ソフトウェア以外

の製品を例に説明しよう。例えば、「防水」と書かれた製品を、利用者によっては、海や入浴剤の入った風呂に持ち込むことがあり得る。その場合、単なる水に対する防水機能だけでなく、塩分その他化学物質に対する耐久性までも要求される。これらも備えておかないと、利用者にとっては「防水」とは言えないであろう。

デバッグでの不具合は、多くの場合、意図したプログラムの論理構造が実際に書いたプログラムで実現していないとか、書き間違えた、ということに起因している。一方、ソフトウェア・テストが対象とするシステムの不具合とは、利用者にとって不都合なこと、危険なことであり、ソフトウェア・テストではそれらが生じないかどうかまで考えなければならない。

図表1 デバッグとソフトウェア・テストの相違点

| | デバッグ | ソフトウェア・テスト |
|------|------------|----------------------|
| 目的 | バグをなくす | 品質リスクを評価して、期待品質を保証する |
| 対象 | プログラム | (ソフトウェアを含んだ) システム |
| 作業 | 不具合を解消する | 不具合を見つける |
| 作業者 | プログラマ | テスト技術者 |
| 開始時点 | プログラムを書いた後 | テスト計画は、要求仕様の時点から作成 |
| 終了時点 | (なし=永遠に続く) | テスト担当者の判断 |

科学技術動向研究センターにて作成

4 組込システムにおけるソフトウェア・テストの重要性

ソフトウェア・テストの重要性は、特に最近、組込システム (embedded systems)、あるいは組込ソフトウェア (embedded software) と呼ばれる技術分野で、認識されるようになった。この技術分野では、パソコンのOSやワードプロセッサと同じようには、ソフトウェアを使っているという意識を利用者が持たない。これらは何らかの機械システムの中に埋め込まれていて、その働きを支えるソフトウェアが組込ソフトウェアであり、そのようなシステムを組込システムという。

かつてコンピュータが高価であった時代には、このような技術は、特別な場合、例えばアポロ宇宙船の制御であるとか原子炉の制御などのようなところにしか使われなかった。ところが、マイクロプロセッサの低価格化と普及により、様々な機械の中にマイクロプロセッサが使われるようになり、それに伴って、組込ソフトウェアの使用領域も広がった。現在では、ほとんどの機械製品に組込ソフトウェアが使われている。例えば、すでに乗用車には数十台のマイクロプロセッサが使われており、そ

のための組込ソフトウェアのプログラム総量は 1000万行を超えると言われている^{16~18)}。組込システムは急速に拡大し、今や、我が国における産業規模は総額で62兆円、国内総生産比率で12%に達する¹⁹⁾。この報告によれば、組込システム関連企業の従業員数は 471万人にのぼり、これは全産業比率では9%に当たる。

産業として無視できない規模を備えているだけでなく、その品質が国民生活に重大な影響を与えるという点で、組込システムのソフトウェア・テストは重要視されている。組込ソフトウェアの不具合は、冒頭に述べた自動改札機の不具合の例でも分かるように、担当者の予想をはるかに超えて大きな影響が及ぶことがある。携帯電話の不具合では、その修正作業だけでなく、製品の回収・交換作業を伴うために、膨大な費用がかかっている。このような実際の例によって、組込ソフトウェアのテストは、それ以外の企業の情報システムのテストなどよりも、注目を浴びるようになってきている。組込ソフトウェアのテストを専業とする企業も設立され、順調に業績を伸

ばしている。

組込システムの技術分野では、設計者がハードウェアとソフトウェアと両方について詳しくなければならぬ。原因がハードウェアとソフトウェアの両方にわたる可能性があり、テストにおいては、片方だけを個別にテストして済ませるわけにはいかない。

なお、インターネットの普及と企業の情報システムが様々な関係者と連携するという最近の動きは、組込システムではない従来の情報システムをも、組込システムと同じように、社会の様々な分野まで影響を及ぼしかねない性質を持つものへと変貌させている。つまり、企業システムの組込システム化、とでもいうべき現象が生じている。一企業内のシステムの問題が、一企業を超えて社会へ広がってしまう危険性が生じている。端的な例は、ウィルスソフトの感染である。現在では、個人あるいは企業が対策を怠ることによって、ネットワークにつながったあらゆる個人や企業に悪影響を及ぼしてしまう可能性がある。

5 ソフトウェア・テスト技術の最近の動向

ソフトウェア・テストには、これまでの技術開発の蓄積から、いろいろな種類がある。これらは、テストが行われる環境、テストの対象、テストの目的、テストの手段、方法などによって分類されている。ここでいう環境とは、ソフトウェア開発段階を指す場合もあれば、テストの実施方法を指す場合もある。また、参考文献^{20~22)}な

どのソフトウェア・テストの教科書でも、取り上げられるテストの種類や呼び名が微妙に異なっている場合がある。個別のソフトウェア・テスト技術や技法については、それらの教科書や参考文献に譲り、ここでは、最近のいくつかの動きについて述べる。

5 - 1

テストのプロセスとソフトウェア開発のプロセスの連動

従来のソフトウェア・テストは、ソフトウェアが開発された後に行うものとして、図表2のようなV字モデルで考えられていた。しか

し、このように、開発が終わってからテストするのは、不具合の修正が開発の後になってしまう。つまり、上流段階で生じた不具合が、テストの最後に見つかるために、修正の時間が膨大になるという欠点がある。

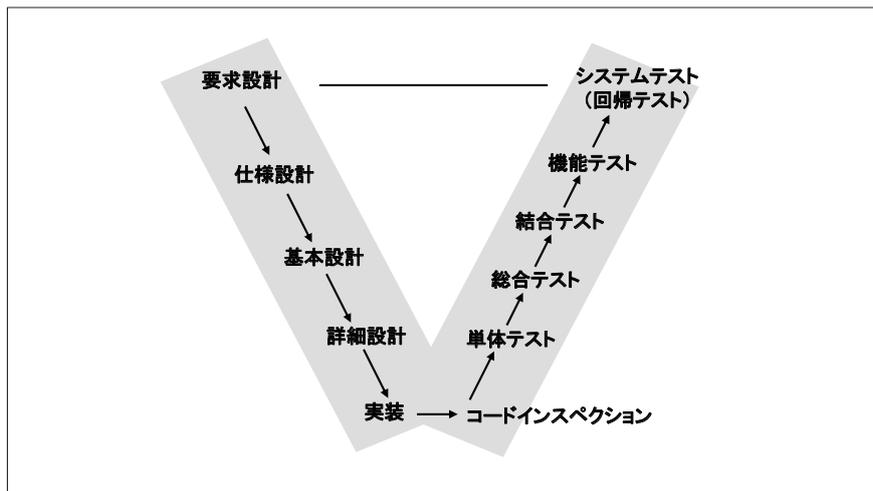
テスト実施の段階においては、闇雲にテストをして、その結果を集めただけでは効果は出ない。日本で開発されたソフトウェア・テスト手法の代表例の一つに、HAYST法がある²³⁾。これは実験計画法に基づいた手法で、事前に効率的なテストを実施するにはどうすればよいかを計画することによって、効果を出すというものである。

品質工学においては、「品質の作り込みは、上流工程で行うべし」と言われている。テストにおいても、このような上流工程での取り組みが行われるようになり、最近では、図表3のようなWモデルが、従来のV字モデルにとって代わるようになった。

図表3の左側の大きな下向きの2つの矢印の方向は、古典的なソフトウェア開発のウォーターフォール開発、すなわち、科学技術動向 2006年10月号「情報通信技術と「思想」- 科学技術の能力としての「思想」-」²⁴⁾で述べられているような線形的な開発に沿ったものであるが、そのそれぞれの段階で両方向の矢印(←→)が示しているようなテスト開発が行われる。また、開発が全て終わってからソフトウェア・テストが始まるのではなく、開発を始める前の要求を定める段階から、システム・テスト設計を行う。要求項目がテストで確認できる特性を備えているかどうかというテスト可能性の検討を含めて、破線の矢印で指されているような、最終的なシステム・テストの設計を行うのである。システム・テスト自体は、従来のV字モデルと同様に最終工程で行われるが、その設計は上流工程から行われる点が従来方式とは異なる。

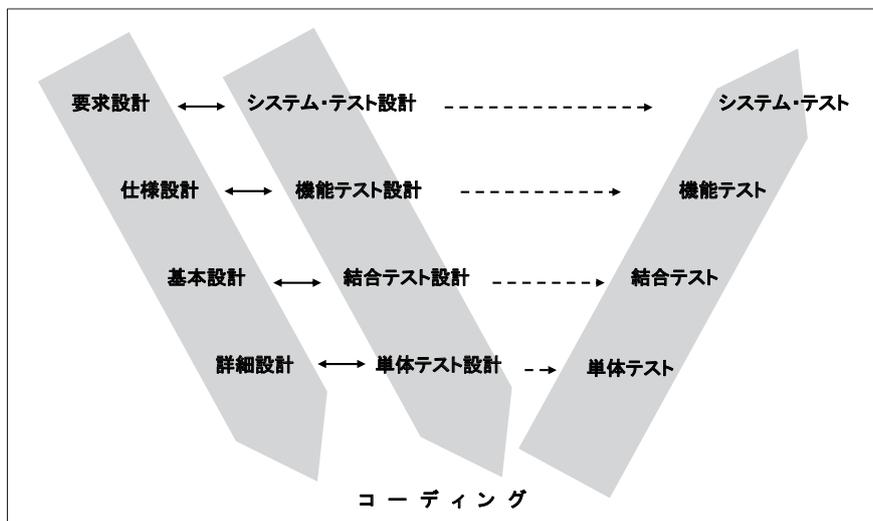
Wモデルは、単体プログラムの開発段階では、アジャイル開発手

図表2 開発とテストのV字モデル (従来のモデル)



科学技術動向研究センターにて作成

図表3 開発とテストのWモデル (最近のモデル)



科学技術動向研究センターにて作成

法と同じように、プログラム設計に並行してテスト・プログラムを作成する。ただし、アジャイル法では、プログラムを作るプログラマがテストを考えるが、Wモデルでは、プログラマではなくテスト技術者がテストを作成する、という点が異なる。すなわち、テストという視点からプログラム作成をとらえる、ということになる。

本来、テストが必要でない程の特別品質のソフトウェアが最初から作られるのが理想的であるという考えもある。しかし、どのような製品やシステムでも、テストが不要になることは、将来的にもあり得ないと思われる。組み込まれたシステムの機能の高度化などの

要求により、ソフトウェアがますます複雑化し、巨大化している状況にあっては、製品やシステムの利用に関して全ての場合を事前に網羅的に考え、あらゆる場合を想定した完全なプログラムを作るのは不可能である。そこで、リスクマネジメントのために、でき上がりがつつある、あるいはまた、でき上がったソフトウェアを実際に動かしてテストし、そのような予想外の状況に対して、どの程度製品やシステムが安全に動作するか確認することの重要性は極めて大きい。

ここで、「開発に際しては、テストを考える」という原則を、「開発の前に、テストを作り、そのテストを通せるように開発を進める」と言

い換えるならば、3章で述べたTDDというテスト駆動開発の方法論の枠組みに当てはめることができる。

目に見えないソフトウェアの開発においては、このようなテスト可能性の追求やテスト駆動の方法論が特に有効である。また、これらは、法律や制度のような広い意味のソフトウェアにおいても有用であろう。現在、多くの技術分野で進められている「見える化」や「達成度評価」などにおいて、テストできるかどうかの一つの指針になってきていることは、その証左と考えることもできるだろう。

5 - 2

静的テストの重視

ソフトウェア・テストには、動的テストと静的テストがある。動的テストとは、プログラムを実行して結果を解析するテストである。一方、静的テストとは、プログラムを実行しないでプログラムテキストの解析などでテストを行うものである。

ソフトウェア工学の体系をまとめた“Guide to the Software Engineering Body of Knowledge”²⁵⁾に書かれている「テスト」とは、基本的に実際に動作させて判定する動的なもの、とされており、この本では静的テストは検証(Verification and Validation)に含まれている。しかし、それ以外の多くのソフトウェア・テストの教科書では、静的テストもテ

ストの一部として解説されている。

静的テストは、「レビュー」「インスペクション」というような人の目による検査とともに、「メトリクス」「コーディング規則違反検査」「静的解析」などといったコンピュータを使った分析やデータ処理なども含む。静的テストは、その分野の知識を備えた専門家でないと実施できないため敷居は高いが、開発部隊へ迅速なフィードバックができる、という効果がある。すなわち、開発に係る人員の能力を含めた開発の体質そのものを高める上で効果がある。

システム開発の品質向上では、上流工程での品質確保が必要であり、より有効である¹¹⁾。同じことがソフトウェア・テストにおいても言える。設計段階でのテスト、特に、時間的にきわどい状況での試験技術として、モデル・チェック(モデル検査)という技法が注目を集めている²⁶⁾。モデル・チェックは、ソフトウェアそのものをテストするのではなく、ソフトウェアを含めたシステムを状態遷移機械として表現したときに、その時間的な遷移関係と、このシステムが満たすべき性質を時制論理式で表現して、これを検証器(verifier)を使って検証するものである(モデル・チェックをソフトウェア・テスト技術に含めることが適当かどうかについては異論もある。多くの教科書では、モデル・チェックまでは含まれていないことが多い)。この技術分野は、自動定理証明という機械証明技術として長い歴史があるが、最近では各種の

ツールが開発され、実際に有効な結果を出している。よく使われているツールにはSPINなどがある²⁷⁾。

5 - 3

ソフトウェア・テストの自動化

一般に、ソフトウェア・テストは、ソフトウェアの規模が大きくなると指数関数的に作業量が増える。それは、複数の条件が重なった場合のテスト個数の増大を考えれば容易に想像がつく。そこで、テストを自動化して、人間の作業量を減らし、同時にテスト作業の高品質化あるいは高能率化を図ろうという考えが自然と出てくる。

ソフトウェア・テストの自動化には、次のような種類がある。

- インターネット上の負荷テストなど入力負荷の自動生成
- TTCN-3などテスト・スクリプト言語によるテストの自動化
- UTP(UML Testing Profile) などによるテストケースの自動生成

一方、特にツールに頼った自動化の導入には批判もある²⁸⁾。これには、場当たりの自動化やテスト・ツールの導入が、必ずしも予期した成果を上げていないという背景がある。自動化が成功するためには、熟練した技術者の養成などそれなりの準備が必要であり、成果を出すための工夫が必要となる。

6 ソフトウェア・テスト産業

ソフトウェア・テストの重要性への認識が増すとともに、テスト作業を外部に委託する企業も出てきている。その主な理由としては、1) テストに伴う作業コストを低減するため、2) 比較的短時間に作業

量が集中することから、その人員調節のため、3) 社内よりも外部のほうがテストの専門知識が高いため、などが挙げられる。

特に、最近の組込システムにおいては、テスト作業が膨大な量に

上ることから、外注への依存度が高まっている。このような背景のもとで、2005年にIT検証産業協会(IVIA)²⁹⁾が発足し、2008年3月末現在で47社が加盟している。IVIAは技術部会、標準化部会、教

8 - 1

事故調査委員会の設置による経験知の共有

社会的に影響のあるソフトウェア・システムで事故が起こった場合には、例えば、航空機や鉄道の事故のために設置されている航空・鉄道事故調査委員会のような第三者機関によって、ソフトウェア・システムを徹底的に解析し、同様の事故を起こさないように、社会として学習できる仕組みが必要である。このような委員会の設置方法には、受発注者共同型（型で例えるならば、医局コンファレンス型）、当事者を含まない公平な外部組織（例えば、日本学術会議の情報学委員会あるいは（社）情報処理学会などの関係学会）などが考えられる。そのミッションは責任追及ではなく、あくまで事故解析・分析に限定することが肝要である。なお、現在、政府など公的機関のシステム発注では、工程段階ごとの分割発注が多い。このため、完成後に事故が発生すると、関係者が多岐にわたるといえる問題がある。このようなソフトウェアでは、一括発注の場合とは異なる調査分析上の工夫も必要となる。

8 - 2

第三者機関によるソフトウェア・テストの監査

社会的に影響のあるソフトウェア・システムには、発注者および受注者以外の外部の第三者を含めたテスト監査が必要と考えられ

る。社会的に影響のあるソフトウェア・システムを構築した場合には、外部の第三者機関にテストを監査してもらい、その結果を第三者機関が保管することにより、不都合が生じた場合に速やかに問題解決に取り組めるような仕組みが必要ではないだろうか。このような第三者機関には、ソフトウェア・テストの部分を含めて、総合的な品質評価とリスクマネジメントが評価できる能力が求められる。

8 - 3

法制度の整備

現在の製造物責任(PL)法の制度は、ソフトウェアには適用されない。しかし、PL法の適用を望む声もあり、組込システムの場合には、ハードウェアとソフトウェアとが複雑に絡み合って事故に至る可能性もあるので、ソフトウェアに対しても製造責任を問う法制度の整備が行われる可能性がある。その場合には、現行のソフトウェアライセンス条項(End User License Agreement,EULA)におけるような、全面的な製造者免責条項をどう取り扱うかを含めて、事故被害の最小化を目的とした整備が望まれる。特に、ハードウェアと異なるソフトウェアの特殊性、すなわち、使用者あるいは外部者による改変をどう取り扱うか、技術開発を推進しながら、想定される被害をどう抑えるかなどの課題を解決していかなければならない。

8 - 4

用語・記述・表現の研究および基本知識の普及

ソフトウェアの事故や問題は、ともするとソフトウェアの専門家だけの問題であるとして、組織全体で扱われないことが多い。このことがソフトウェア品質の放置につながっていると思われる。このような事態を改善し、いわゆるe-Japan戦略の成果を国民すべてのものにするために、この技術分野の用語・記述・表現法などの研究を進め、ソフトウェアの事故や問題を一般の人や組織の長に分かりやすく説明できるものにする努力が必要である。また、それらを一般に教育し、普及啓蒙していく活動も必要である。

このような取り組みは、長期的に見れば、日本製品を含めた日本のシステムの品質を見直し、品質に関わる作業を企業内部の活動にとどめず、広く世の中に提供できる活動、いわば、「品質の産業化」とでも呼べるような活動にまで高める可能性を秘めている。ソフトウェア・テストをこのような品質産業という文脈に置いて考えるならば、ソフトウェア・テスト技術の研究開発や、ソフトウェア・テストの研究者・設計者・技術者・管理者の養成という投資は、日本の産業競争力を強化するだけでなく、新たな知的産業を創造し、日本のみならず世界の安全・安心に寄与できる可能性につながるだろう。

9

おわりに

ソフトウェア・テストは、従来から、ソフトウェア開発における最終工程として行われていたが、

ともすれば、プログラムの問題点を解消する作業であるデバッグと混同されてきた。しかし、最近に

なって、リスクマネジメントとして、あるいは、品質保証活動の一環としてのソフトウェア・テスト

が強く意識されるようになった。

それは、ソフトウェアを組み込んだシステムが広く使われるようになり、想定外の使われ方によるトラブルも頻発して、場合によっては、我々の日常生活にまで深刻な影響を及ぼすようになったからである。このような問題は、単なるデバッグ(あるいは、デバッグのみで済むという認識)では解決できず、システム内外で様々なケースを想定したソフトウェア・テストが行われなければ解決できない。品質の良い信頼性の高いソフトウェアを作るために技術開発や制度・環境を整備することと同時に、ソフトウェア・テストの技術開発や専門家の養成、さらにその制度や環境を整えることが、どうしても欠かせない要素になる。実際に、ソフトウェア・テストの重要性とその困難さから、ソフトウェア・テストを専業とする企業が新たに生まれており、このような産業は確実に発展している。

これらはソフトウェアの「品質の産業化」であり、ソフトウェア・テストの研究者・設計者・技術者・管理者の養成という投資は、日本の産業競争力を強化するだけでなく、安全・安心に寄与できる新たな知的産業を創造しようということを強調しておきたい。

謝 辞

本稿をまとめるにあたり、下記の方々にお会いして貴重な情報や意見を頂戴した。ここにあらためて謝意を表したい。香川大学工学部信頼性情報システム工学科 古川善吾 教授、電気通信大学電気通信学部システム工学科 西康晴 講師、日本電子専門学校オープンソース教育システム 小菅貴彦 主任研究員、情報処理推進機構(IPA) ソフトウェア・エンジニアリング・センター(SEC) 鶴保証城 所長、組込み系プロジェクト 門田浩 プロジェクトリーダー、平山雅

之 研究員、JPCERTコーディネーションセンター 伊藤友里恵 業務統括、椎木孝斉 主席分析官、富樫一哉 チーフシステムアーキテクト、古田洋久 情報セキュリティアナリスト、株式会社豆蔵 大西建児 シニア・コンサルタント、有限会社デバッグ工学研究所 松尾谷徹 代表、ソニー株式会社テレビ事業本部 高橋寿一 主任技師、株式会社ベリサーブ 浅井清孝 社長、勝又信人 部長、佐々木方規 部長(順不同)。

参考文献

- 1) 日経 ITpro, 東証の新派生売買システムが復旧、原因はメモリーの初期化エラー <http://itpro.nikkeibp.co.jp/article/NEWS/20080212/293526/>
- 2) 毎日新聞、自動改札機トラブル: 首都圏の通勤・通学客は戸惑いの表情、2007年10月12日
- 3) 朝日新聞、1文字分のミスで大トラブルに 首都圏改札機トラブル、2007年10月28日
- 4) 有賀貞一、前回の教訓を生かせず、また起きた自動改札機のトラブル、IT業界の進路、NikkeiNet、2007年10月16日 http://it.nikkei.co.jp/business/column/aruga_gyokai.aspx?n=MMIT0z000016102007
- 5) 有賀貞一、「スイカ」改札通れないトラブル・説明責任を果たすべき、IT業界の進路、NikkeiNet、2006年12月5日 http://it.nikkei.co.jp/business/column/aruga_gyokai.aspx?n=MMIT0z000004122006
- 6) 深谷直彦・古川善吾・西康晴編、特集 ソフトウェアテストの最新動向、情報処理、Vol.49、No.2、126-173、2008年2月
- 7) Glenford Myers, The Art of Software Testing, 松尾正信訳、ソフトウェア・テストの技法、近代科学社、第1版は1979年、第2版は2004年、翻訳は2006年

- 8) SQuBOK策定部会編、ソフトウェア品質知識体系ガイド-SQuBOK Guide-、オーム社、2007年
- 9) JIS X 0160:1996ソフトウェアライフサイクルプロセス、日本工業規格、1996年
- 10) JIS X 0170:2004システムライフサイクルプロセス、日本工業規格、2004年
- 11) 黒川利明、情報システム構築の品質・信頼性向上のために—上流工程の"ビジネスルール"と要求工学を検討する、科学技術動向 No.32、特集1、2003年11月号
- 12) (独)情報処理推進機構ソフトウェア・エンジニアリング・センター編、経営者が参画する要求品質の確保~超上流から攻める IT化の勘どころ~, SEC BOOKS、オーム社、2005年
- 13) 林晋・黒川利明、二つの合理性と日本のソフトウェア工学、科学技術動向、No.42、特集1、2004年9月号
- 14) Kent Beck, Test Driven Development: By Example, Addison-Wesley Professional, 2002(長瀬訳、テスト駆動開発入門、ピアソンエデュケーション、2003年)
- 15) Susumu HAYASHI, PAN YiBing, Masami SATO, Kenji MORI, SUL Sejeon and Shusuke HARUNA, Test Driven Development of UML Models with SMART modeling system, <<UML>>2004- The Unified Modeling Language, Lecture Notes in Comp. Sci., No.3273, Springer-Verlag, pp.395-409, 2004
- 16) 鍛冶克彦、情報政策の新展開とSECへの期待、SEC Forum 2006、2006年、https://sec.ipa.go.jp/download/dl.php?filename=event/2006/20060612/secforum2006_0_kazi.pdf
- 17) 産業構造審議会情報経済分科会 情報サービス・ソフトウェア

- 小委員会中間とりまとめ報告、2006年6月、<http://www.meti.go.jp/press/20060613010/torimatome-hontai-set.pdf>
- 18) 山下徹編著、高度IT人材育成への提言、NHK出版、2007年
- 19) 経済産業省商務情報政策局 情報政策ユニット情報処理振興課、2007年版組込みソフトウェア産業実態調査報告書、2007年6月
- 20) 大西他、JSTQB教科書 JSTQB認定テスト技術者 Foundation Level試験、翔泳社、2007年
- 21) Cem Kaner, Jack Falk, Hung Quoc Nguyenテスト技術者交流会誌、基本から学ぶソフトウェアテスト～テストの「プロ」を目指す人のために～、日経BP社、2001年
- 22) Kaner, Cem, James Bach, Bret Pettichord (Eds.) テスト技術者交流会誌、ソフトウェアテスト293の鉄則、日経BP社、2003年
- 23) 吉澤正孝・秋山浩一・仙石太郎、ソフトウェアテスト HAYST法入門 品質と生産性がアップする直交表の使い方、日科技連、2007年
- 24) 林晋、情報通信技術と「思想」- 科学技術の能力としての「思想」-、科学技術動向、No.67、2006年10月号
- 25) Alain Abran and James W. Moore (Executive Editors), Pierre Bomrque and Robert Dupuis (Editors), Guide to the Software Engineering Body of Knowledge, 2004 Version, SWEBOK, IEEE Computer Society and IEEE, 2004 [亘理]
- 26) 産業技術総合研究所システム検証研究センター、4日で学ぶモデル検査、エヌ・ティー・エス、2006年
- 27) ON-THE-FLY, LTL MODEL CHECKING with SPIN: <http://spinroot.com/spin/whatispin.html>
- 28) 高橋寿一、知識ゼロから学ぶソフトウェアテスト、翔泳社、2005年
- 29) IT検証産業協会 (IVIA) ホームページ: <http://www.ivia.gr.jp/index.html>
- 30) Cem Kaner, Jack Falk, Hung Quoc Nguyen, Testing Computer Software, 2nd Edition, John Wiley & Sons, 1999 テスト技術者交流会誌、基本から学ぶソフトウェアテスト～テストの「プロ」を目指す人のために～、日経BP社、2001年11月26日
- 31) International Software Testing Qualifications Board (ISTQB) ホームページ: <http://www.istqb.org/>
- 32) Japan Software Testing Qualifications Board (JSTQB) ホームページ: <http://jstqb.jp/index.html>
- 33) 岡村久道、会社の内部統制、日経新聞出版社、2007年
- 34) 金融庁、会社法及び金融商品取引法ならびに両法施行規則等、2007年2月
- 35) 金融庁、財務報告に係る内部統制の評価及び監査基準、2007年2月
- 36) 金融庁、財務報告に係る内部統制の評価及び監査に関する実施基準、2007年2月
- 37) 経済産業省、システム管理基準追補版(財務報告に係るIT統制ガイドダンス)、2006年2月
- 38) 経済産業省、システム管理基準追補版(財務報告に係るIT統制ガイドダンス)追加付録、2007年12月

執筆者



黒川 利明

客員研究官

株式会社CSKホールディングスCSKフェロー
<http://www.csk.com/index.html>

●
東芝、IBMを経て現職。プログラミング言語、オブジェクト指向、メタデータなどの標準化に従事。システム開発の上流工程、サービス科学、科学技術コミュニティ、クラウド・コンピューティングにも関心がある



品川 萬里

客員研究官

法政大学 情報技術(IT)研究センター
学術担当教授

●
1980～2000年に電気通信行政に従事。電気通信市場のde-monopolise、地上波TV放送のデジタル化プロジェクト等に参画。その後SIビジネスにも従事。理科苦手の理科ディレクターとして自然像認識の変化と社会像認識の変化に関心。