【産業競争力懇談会２００９年度推進テーマプロジェクト報告】

エンタプライズ・ソフトウェア
生産革新プロジェクト

２０１０年３月１２日

産業競争力懇談会（ＣＯＣＮ）

【エグゼクティブサマリー】

エンタプライズ・ソフトウェア生産革新プロジェクト

1. エンタプライズ・ソフトウェアのビジョン

エンタプライズ・ソフトウェアは銀行、証券、病院、鉄道など大規模かつ社会基盤を支える情報システム、いわゆる「重要インフラ」システムの重要な一要素である。また、すべての産業にとってもインフラ的な位置付けで利用されている。このように、エンタプライズ・ソフトウェアは一般の国民には直接見えにくいが、社会基盤的なシステムや日本の産業を下支えしている、国民にとってはなくてはならない存在である。

今までのエンタプライズ・ソフトウェア開発は、ソフトウェア発注者、ソフトウェア運用者・利用者、経営者等様々なステークホルダが関わっているが、実際の開発では設計や開発の視点で不具合やバグを減らすことを重視してきた。実は、ステークホルダによってエンタプライズ・ソフトウェアの品質の視点は異なる。それぞれのステークホルダの観点による品質を満足したとき、そのソフトウェアは正しい品質のレベルであると言える。今後は、エンタプライズ・ソフトウェアのライフサイクル全体を見据えた開発方法が必要になる。

また、情報サービス産業におけるビジネスという観点では、近年、欧州を中心として標準化の動きが活発である。日本もこの標準化の動きに乗り遅れることなく、逆に先行して国際標準化も視野に入れて本プロジェクトを進めることにより、近い将来、国内の多くの企業が参入障壁を乗り越える技術力を保有できることを確信する。

本プロジェクトでは、「世界最先端の革新的ソフトウェア開発手法の確立」を目標として、ソフトウェア開発者だけではなくその他のステークホルダの視点も考慮した技術を先行開発し、それをベースに成長著しい新興国という市場で欧州や米国と競争する意識で活動する。

2. プロジェクトの背景

【エンタプライズ・ソフトウェア開発の現状】

- ソフトウェアの複雑性（ソフトウェアが大きくて複雑なことがさらなる複雑性をもたらし、結果として信頼性が低下）と変更容易性（ソフトウェアは柔軟に変更することが可能）は、旧態依然として人のスキルに依存しやすいレビューや人海戦術によるテストに頼っている日本のエンタプライズ・ソフトウェア生産現場の改革を著しく阻害している。
- 要件定義工程や上流の設計工程は様々なステークホルダとの合意プロセスが必要であることから、この合意プロセスが不十分である。
- 要件定義や設計時にエンタプライズ・ソフトウェア要求の暗黙知が存在しており、

それを正しく分析して仕様書や設計書に反映できていない。

【エンタプライズ・ソフトウェア運用の現状】

最近 1 年間で発生したシステム障害のうち、実に約半数の障害がソフトウェア不具合が原因である。その原因を分析すると、以下の 2 つの原因が浮かび上がる。

- ソフトウェア発注者のビジネス環境や外部状況の変化に対応できていない、あるいはそれに対応するための仕様変更を行うときにソフトウェア不具合が含まれてしまう。
- 誤作動が国民生活や社会経済活動に影響を及ぼすソフトウェアを体系的に構築するソフトウェア技術が現実の要求に充分に対応できていない。

【エンタプライズ・ソフトウェア生産革新に関連する日本の取組み状況】

エンタプライズ・ソフトウェア生産革新に関連する、日本の要求分析技術やモデル記述・検証技術、自然言語処理等の取組みは一定の成果が出ており、要素技術は揃ってきている。これから我々のプロジェクトは、これらの要素技術を組み合わせ、統合していく次のステップに移らなければならない。

3. ソフトウェア生産革新の課題整理

エンタプライズ・ソフトウェアの現状を考慮し、エンタプライズ・ソフトウェア生産革新プロジェクトは、すでにあるいくつかの要素技術を組み合わせて統合し、さらにその成果を実証する実証実験を実施するために、上流工程での信頼性向上に重点を置いた、以下の 6 つの研究開発課題を解決する必要がある（図 i 参照）。

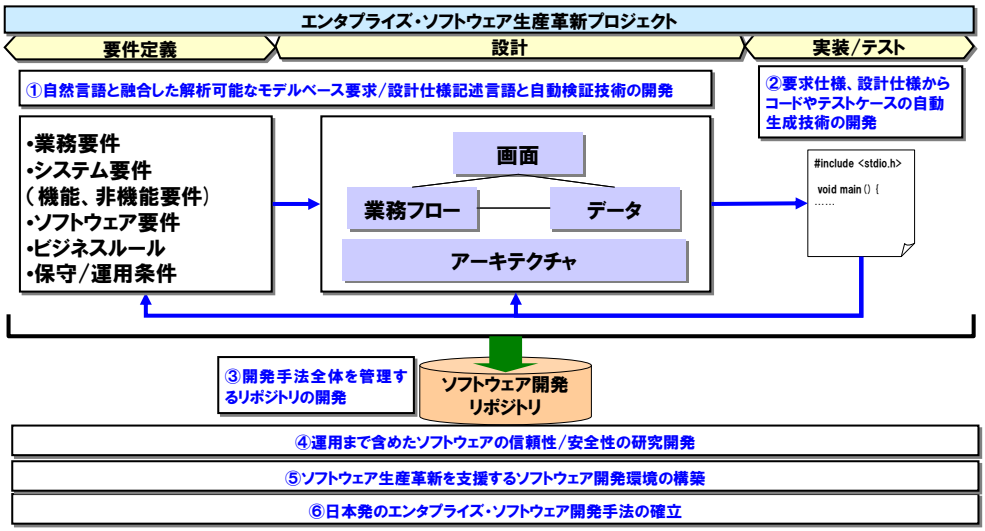


図 i エンタプライズ・ソフトウェア生産革新プロジェクトの研究開発課題全体像

① 自然言語と融合した解析可能なモデルベース要求／設計仕様記述言語と自動検証技術の開発

② 要求仕様、設計仕様からコードやテストケースの自動生成技術の開発
③ 開発手法全体を管理するリポジトリの開発
④ 運用まで含めたソフトウェアの信頼性／安全性の研究開発
⑤ ソフトウェア生産革新を支援するソフトウェア開発環境の構築
⑥ 日本発のエンタプライズ・ソフトウェア開発手法の確立

これらの研究開発課題が解決し、世界最先端の革新的ソフトウェア開発手法を実現すると、以下の効果が期待される。

(1) 社会的基盤を担うエンタプライズ・ソフトウェアにおいて、30%程度が開発フェーズでの問題発生が原因となるシステムトラブルである。よって、本手法の実現により、ソフトウェア・システムの不具合が最大で 30%減少し、一般国民の社会生活の不安を和らげることになる。

(2) 開発上流工程から高い品質、高い信頼性を確保できるので、開発費を圧迫しているテスト工数を大幅に削減することができる。その結果、ソフトウェア開発企業の収益性が向上する。さらに、現状 1 億行を超えているエンタプライズ・ソフトウェアを含む 17.8 兆円の情報サービス産業市場において、コスト超過分 1 兆円のうち 8000 億円の無駄の削減、及び 1 年間に発生するシステムトラブルのうちの 30%である 1120 億円の損失防止を実現する。

(3) 信頼性技術の有無が参入障壁となっているエンタプライズ･ソフトウェアの市場を開拓することができる。近年、欧州を中心として機能安全、あるいはセキュリティに関わる製品の標準化制定と実施が活発化し、そこではソフトウェアの信頼性を客観的に保証する技術としてモデルベース開発を推奨している。今後は、要求モデル、設計モデルを利用して開発されたソフトウェアだけが市場流通できる方向に向かっていくであろう。逆に、本研究課題の成果によって、すでに存在する標準化への対応とともに、新たな標準化制定と実施を目指し、国内の多くの企業が先行的に参入障壁を乗り越える技術力を保有できる。

4. エンタプライズ・ソフトウェア生産革新の提言

【エンタプライズ・ソフトウェア生産革新への産学官推進体制の確立】

現在、原子力発電等、他産業でも日本の国家プロジェクトが世界各国と競争しているが、産学官の連携不足のため、思うような成果が出ていない。このような中、国を支える高信頼で安全な社会インフラシステムを構築することは日本の国家プロジェクトとして非常に重要な施策である。それを実現し、世界と競争するためには、国、大学機関、企業が一体となって取組みを進めなければならない。

本プロジェクトの実現のためには、要求分析技術、モデル記述・検証技術、自然言語処理技術等の要素技術を組み合わせ、統合していく応用研究や実証開発が必要である。そのためには各企業が連携した連合体制確立や、本プロジェクトの研究開発に関連する有識者

が在籍する大学・関連機関と連携しての共同研究実施、推進コミュニティの体制等を作ることが必要である。

また、日本としては、エンタプライズ・ソフトウェア開発手法の国際標準化を目指し、この開発手法を活用したエンタプライズ・ソフトウェアの信頼性/安全性を向上させるところで競争するべきである。このような産業界が普及展開を主導して取り組む市場化フェーズの場合は、まず、産業界が協調領域となり得る研究分野を提示し、かつ産業界もそれ相応の資金を提供するようにしてリスクを背負いながら、産業界が学界や官界を先導して活動しなければならない。このような産学官が連携したトライアングル体制を形成すべきである。

さらに、エンタプライズ・ソフトウェアを含む大規模なシステムの実証実験等を行うためには、多額の予算の確保や予算が実証実験に使用されることが明示的に了承されていることが必要である。本プロジェクトでも今後、明示的に研究予算がつくこのような大規模な実証実験等を実施し、産業界への普及展開を加速させるべきである。

【エンタプライズ・ソフトウェア分野への研究開発投資】

エンタプライズ・ソフトウェアはすべての産業に多大な影響を及ぼすインフラ的な存在であり、欧州や米国はそれを踏まえて情報通信産業等に積極的な研究開発投資を実施している。一方、日本の政府が投資する政府研究開発投資のうち、情報通信産業への研究予算は、最近ではほぼ横ばいか若干減少傾向である。情報通信技術やエンタプライズ・ソフトウェア分野において、欧州や米国との競争に打ち勝っていくためには、さらなる研究開発投資を行うことが重要である。

【実現までのロードマップ】

本プロジェクト実現に向けて、第 3 章で述べている 6 つの研究開発課題を解決するためには、研究開発課題の順序付けを行って段階的に進める革新的ソフトウェア開発手法の研究開発と、それを行うための基盤整備及び支援を実施しなければならない。

このように進めることによって、本提言は世界をリードする施策に成長させることができると考える（図 ii 参照）。

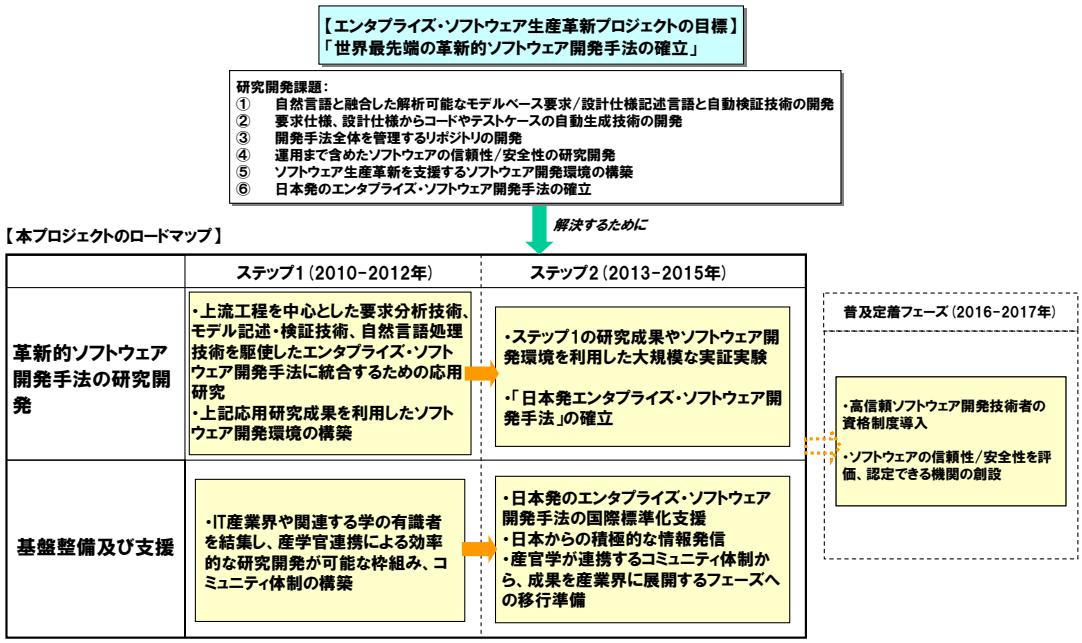


図 ii　本プロジェクト実現までのロードマップ

【目　次】

はじめに

ソフトウェアは、一般的にシステムを構成する一要素であるが、当初はメインフレームと呼ばれる汎用大型コンピュータに付随している程度の代物であった。しかし、ハードウェアの高性能化やネットワーク技術の進歩とともに、ソフトウェアも進歩を続け、今ではソフトウェアは大規模化、複雑化している。以前はハードウェアによって実現していたが、今やソフトウェアによって実現している機能も存在し、システムにおけるソフトウェアの役割は非常に大きいものになっている。

また、企業の業務システムや情報システム、あるいは社会基盤を支える情報システムにはエンタプライズ・ソフトウェアが入っている。つまり、エンタプライズ・ソフトウェアは、すべての企業活動を支えるインフラ的な存在であり、また一般国民が日常生活を営む上でも非常に重要な位置付けとなっている。

このように、エンタプライズ・ソフトウェアは産業界や一般国民にとって非常に重要な位置付けになってきたにもかかわらず、エンタプライズ・ソフトウェアでは、旧態依然の人のスキルに依存した人海戦術による開発が主となっている。特に、エンタプライズ・ソフトウェア開発の上流工程は、ソフトウェア発注者、受注者の合意形成が重要であるが、その合意形成がきちんとできないために、ソフトウェアの不具合が発生する大きな要因となっている。このようなソフトウェア不具合が発生すると、産業界、あるいは一般国民の日常生活に大きな被害が出ることは、最近のシステムトラブルの事例からも明らかである。

本プロジェクトは、このような背景のもと、「世界最先端の革新的ソフトウェア開発手法の確立」を目指し、検討を行った。すでに、日本には要求分析技術、モデル記述・検証技術、自然言語処理技術等、革新的ソフトウェア開発手法を実現するための要素技術の研究開発は実施されており、本プロジェクトを実現するための素材は揃っている。我々はソフトウェア開発におけるニーズを産業界から提示、これらの研究開発技術を有する有識者を一つのプロジェクトに結集し、上流工程の信頼性向上に重点を置いた日本発のエンタプライズ・ソフトウェア開発手法を構築する。

この革新的ソフトウェア開発手法の実現により、日本国内で競争するのではなく、日本の産学官が一丸となって成長著しい新興国市場をターゲットに欧米と競争していき、国際競争力を向上させるようにすべきである。本プロジェクトを実現するために、関係各位のご理解とご協力をお願いする次第である。

【プロジェクトメンバー】

プロジェクトリーダー：片山　卓也（北陸先端科学技術大学院大学　学長）

メンバー：	深谷　哲司（株式会社東芝）
	篠原　郁二（日本電気株式会社）
	亀尾　和弘（株式会社日立製作所）
	銀林　純（富士通株式会社）
	丸山　勝巳（国立情報学研究所）
（事務局）	中島　震（国立情報学研究所）
	神谷　慎吾（株式会社ＮＴＴデータ）
	塚本　英昭（株式会社ＮＴＴデータ）

1. エンタプライズ・ソフトウェアのビジョン

1.1. エンタプライズ・ソフトウェアの位置付け

エンタプライズ・ソフトウェアは銀行、証券、病院、鉄道など大規模かつ社会基盤を支える情報システム、いわゆる「重要インフラ[1]」システム[11]の重要な一要素である。また、すべての産業にとってもインフラ的な位置付けで利用されている（図 1参照）。例えば、製造業、流通業であればサプライチェーンマネージメントシステム等、自動車産業であればITSや調達・生産システム等に含まれるソフトウェアがエンタプライズ・ソフトウェアに対応する。このように、エンタプライズ・ソフトウェアは一般の国民には直接見えにくいが、社会基盤的なシステムや日本の産業を下支えしている、国民にとってはなくてはならない存在である。

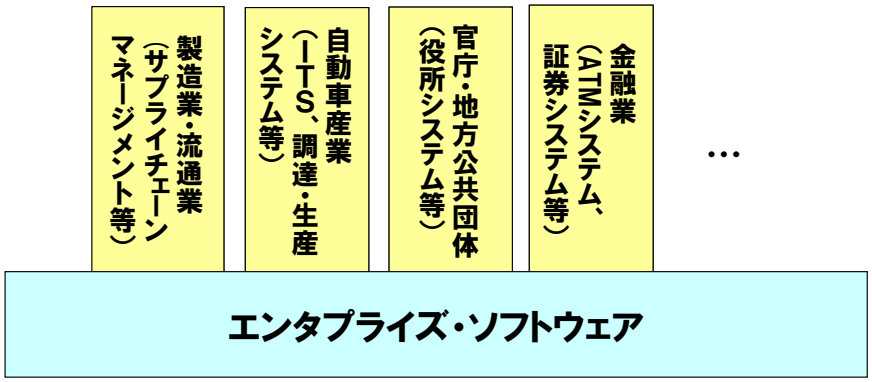


図 1　産業界におけるエンタプライズ・ソフトウェアの位置付け

また、エンタプライズ・ソフトウェアを含む 2006 年の情報サービス市場規模は、国内生産額 17.8 兆円、従業員数 74 万人であり、エンタプライズ・ソフトウェアを含む情報サービス産業は日本を代表する一大産業である。これは 2006 年で 95 兆円の市場規模である情報通信産業のうちの 18.7%を占めており、2006 年の通信業（郵便、固定電気通信、移動電気通信、電気通信に付帯するサービス）の市場規模（国内生産額 17.7 兆円）に匹敵する[14][15]。このように、日本を代表する、エンタプライズ・ソフトウェアを含んだ情報サービス産業自体をさらに活性化させることも我々のプロジェクトの責務である。

ここで、本プロジェクトで定義するエンタプライズ・ソフトウェアを明示的に定義する。エンタプライズ・ソフトウェアは、発注者がソフトウェアの要求を提示し、受注者がその

[1] 他に代替することが著しく困難なサービスを提供する事業が形成する国民生活及び社会経済活動の基盤であり、その機能が停止、低下又は利用不可能な状態に陥った場合に、我が国の国民生活又は社会経済活動に多大なる影響を及ぼすおそれが生じるもの。「情報通信」、「金融」、「航空」、「鉄道」、「電力」、「ガス」、「政府・行政サービス（地方公共団体を含む）」、「医療」、「水道」、「物流」の10 分野。

要求に提案する受発注の契約形態によって開発するソフトウェアと定義する。このようなソフトウェアには、企業活動を営むための業務システムや情報システムのソフトウェア、及び社会基盤を支えている重要インフラシステムのソフトウェアが存在する。受発注の契約形態が存在しない、自動車や家電等のハードウェアに組み込まれているソフトウェア（いわゆる、組込みソフトウェア）やパッケージソフトウェアは含まれない。また、最近、SaaS（Software as a Service）やクラウドコンピューティング技術が注目されているが、これらも一般的には受発注の契約形態ではないため、これらの技術を構成するソフトウェアも含まれない（表 1参照）。

表 1　本プロジェクトにおけるエンタプライズ・ソフトウェアの定義

エンタプライズ・ソフトウェア	**業務システムや情報システムのソフトウェア** **重要インフラシステムのソフトウェア**
組み込みソフトウェア	自動車や家電等のハードウェアに組み込まれているソフトウェア
その他	パッケージソフトウェア SaaS やクラウド技術を構成するソフトウェア

1.2. エンタプライズ・ソフトウェアのビジョン

ソフトウェアは、ハードウェアやネットワーク技術の進歩により、受発注ソフトウェアであるエンタプライズ・ソフトウェアやパッケージソフトウェア、組込みソフトウェアから、よりオープンな SaaS やクラウドコンピューティング技術へと発展し、その形態は様々である。しかし、これらの技術の出現によって、バグが全くないソフトウェアや、運用時にオペレータが絶対に間違えないことを保証するソフトウェアを開発することはできない。特に、今までのエンタプライズ・ソフトウェア開発は、ソフトウェア発注者、ソフトウェア運用者・利用者、経営者等様々なステークホルダが関わっているが、実際の開発では設計や開発の視点のみで不具合やバグを減らすことを重視してきた。実は、ステークホルダによって、エンタプライズ・ソフトウェアの品質の視点は異なる。ソフトウェア開発者の品質の視点は不具合、バグの有無であり、ソフトウェア発注者の品質の視点は正しいソフトウェアを作っているか否か、ソフトウェア運用者・利用者の品質の視点はソフトウェアを含むサービスの劣化の評価、経営者の品質の視点はソフトウェアを含むシステムに対するビジネスとしての有効性の評価である。それぞれのステークホルダが対象のソフトウェアに対して、上記のそれぞれの観点で満足したとき、そのソフトウェアは正しい品質のレベルであると言える。今後は、エンタプライズ・ソフトウェアのライフサイクル全体を見据えた開発方法が必要になる。ソフトウェアを含むサービスの劣化を抑えることができれば、ソフトウェア開発者だけでなく、ソフトウェア発注者、運用者、経営者等すべてのステークホルダの不安を解消することになるであろう。

また、情報サービス産業におけるビジネスという観点では、近年、欧州を中心として標準化の動きが活発である。ソフトウェアの信頼性を客観的に保証する技術としてモデルベース開発を推奨している。現状では推奨のレベルに留まっているが、今後は、要求モデル、設計モデルを利用して開発されたソフトウェアだけが市場流通できる方向に向かっていくであろう。現状は欧州市場ではじまりつつある動きであるが、今後、アジアを含む世界市場で同様な状況になると予測される。日本が先行して本プロジェクトを進めることにより、近い将来、国内の多くの企業が参入障壁を乗り越える技術力を保有できることを確信する。

本プロジェクトでは、「世界最先端の革新的ソフトウェア開発手法の確立」を目標として、ソフトウェア開発者だけではなくその他のステークホルダの視点も考慮した技術を先行開発し、それをベースに成長著しい新興国という市場で欧州や米国と競争する意識で活動する。

2. プロジェクトの背景

2.1. エンタプライズ・ソフトウェア開発の現状

エンタプライズ・ソフトウェア開発は、しばしばビルや家屋等の建物を建てる建設プロセスと比較される。建設プロセスは、建築を依頼する発注者の要求を調査・分析・定義し(要件定義工程)、それを元に設計図を書き(設計工程)、設計図から実際の建物構築を施工(製造工程)、最後に建物が設計図や契約書通りに完成しているかを検査する(テスト工程)。建物は建設途中で取り壊すことはできないため、基本的には工程を遡ってやり直すことは許されない。システムを構築するエンタプライズ・ソフトウェアの開発に対してもいまだに上記のような開発プロセスモデルが主流であり、この開発プロセスモデルを一般的に「ウォーターフォールモデル」と呼ぶ。しかし、ブルックス氏が著した書籍「人月と神話」[3]で述べているように、ソフトウェアは建物とは根本的に異なる以下の本質が存在する。

- 複雑性
  ソフトウェアは大きくて複雑なことそれ自身が問題である。その複雑性がさらなる複雑性をもたらし、その結果、ソフトウェアの信頼性が低下する。
- 変更容易性
  ソフトウェアは柔軟に変更することが可能である(ソフトウェアのソースコードは簡単に修正することができる)。ゆえに常に変更に対する圧力がかかる。
- 不可視性
  ソフトウェアは目に見えない。
- 順応性
  ソフトウェアはハードウェア他さまざまなものと関係を保ち続ける。

特に、ソフトウェアの複雑性と変更容易性への対応は、旧態依然として人のスキルに依存しやすいレビューや人海戦術によるテストに頼っている日本のエンタプライズ・ソフトウェア生産現場の改革を著しく阻害しており、これらの本質を見極めたうえで克服する努力が必要である。

また、「ウォーターフォールモデル」は別名、「V字モデル」と呼ばれる。V字モデルは要件定義から設計工程までを品質の作り込み工程、テスト工程を品質の検証工程と考え、各々対応関係にあるアクティビティで検証を実施するモデルである。そのため、要件定義工程で挿入されたソフトウェアの不具合は、テスト工程の最終段階である受け入れテストの検証によって発見され、修正することになる。2002 年 5 月に発行された米国の国立標準技術研究所のPlanning Report[8]によると、品質の作り込み工程で不具合を発見・修正したときのコストを 1 とすると、受け入れテストの段階で発見・修正した場合はその 15 倍、さらにテストをすり抜けてしまったソフトウェアの不具合を実際のシステムで運用している段階で発見･修正した場合はその 30 倍のコストがかかるという試算も提示されている(図 2参

照)。つまり、V字モデルは不具合の発見・修正が後工程にいけばいくほど、大きな手戻りやコストの増大が発生する問題がある。

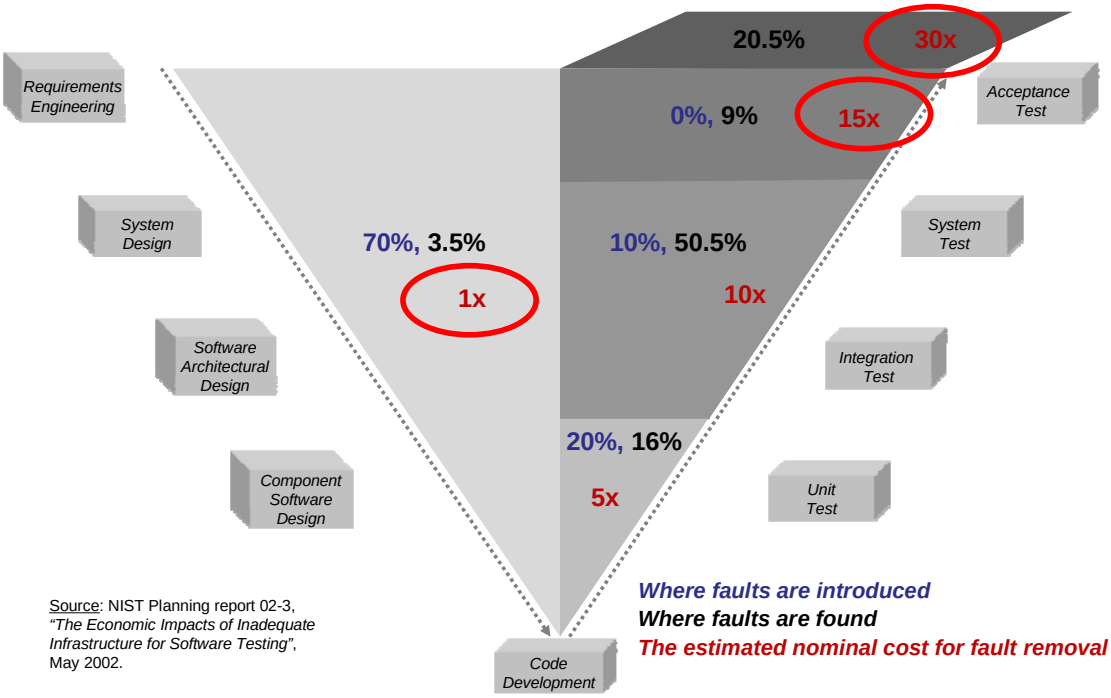
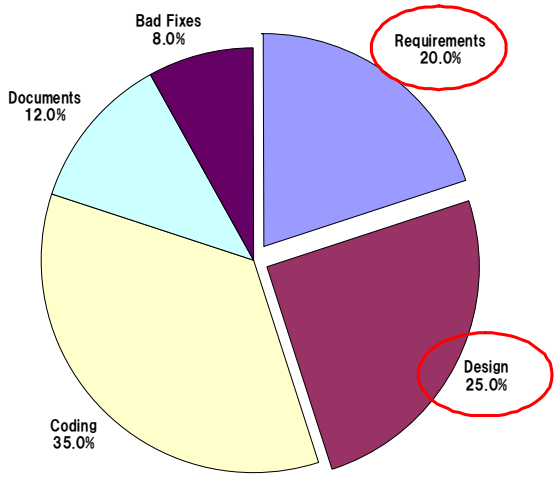


図 2　V 字モデルにおけるソフトウェア不具合のコスト分析

そこで、エンタプライズ・ソフトウェアの開発プロジェクトにおける品質問題の発生原因を調査したところ、20%が要件定義工程、25%が設計工程での問題であるという結果であった（図 3参照[10]）。つまり、エンタプライズ・ソフトウェア開発におけるソフトウェア不具合の半分近くは、要件定義や設計工程の上流工程で埋め込まれている。上記「V字モデル」の特徴から、このことはかなり後工程段階に進まなければ、その不具合が発見・修正されないことを示しており、現在のエンタプライズ・ソフトウェア開発は非常に非効率的であることが指摘できる。

なぜ、要件定義や設計工程の上流工程で、ソフトウェアの不具合が多く埋め込まれるのであろうか。ソフトウェア品質問題の原因を調査した結果として「要件定義が不十分」、「システムの設計が不正確」という原因が上位を占めている（図 4参照[10]）。これらの原因をさらに分析すると、要件定義工程や上流の設計工程は様々なステークホルダとの合意プロセスが必要であることから、この合意プロセスが不十分であることが分かる。つまり、要件定義工程や上流の設計工程の課題は、ソフトウェア受注者のみでは解決できないものも含んでいるかもしれない。また、他の要因として要件定義や設計時にエンタプライズ・ソフトウェア要求の暗黙知が存在しており、それを正しく分析して仕様書や設計書に反映できていない点も挙げられる。最近では、中国やインド等海外の企業と一緒にエンタプライズ・ソフトウェアを開発するオフショア開発が盛んになってきている。特に、オフショア

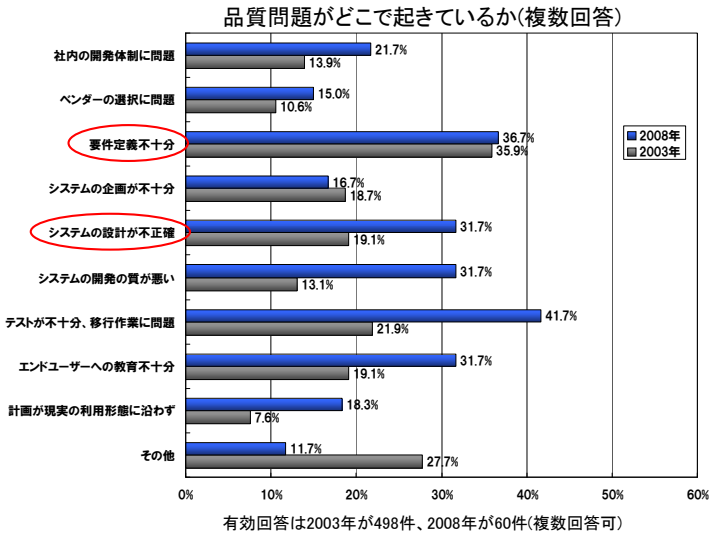
先の海外企業にとって、国ごとに慣習やしきたりが異なるエンタプライズ・ソフトウェア要求を暗黙知のまま進めることは非常にリスクが大きい。技術やスキルを向上させることにより暗黙知を正しく分析・抽出し、ソフトウェア発注者、受注者すべてのステークホルダで共有できるようにしなければならない。



(出典)"SOFTWARE QUALITY IN 2008: A SURVEY OF THE ART"(2008年、SRP Software Productivity Research LLC)(JaSST Japan Symposium on Software Testing ソフトウェアテストシンポジウム資料)、ガートナーコンサルティング分析

※経産省:情報システム・ソフトウェアの信頼性及びセキュリティの取組強化に向けて中間報告書より抜粋

図 3　ソフトウェア欠陥発生フェーズ分析



(出典)日経コンピュータ(2008年12月1日号):第2回プロジェクト実態調査(平成20年8月～9月)

※経産省:情報システム・ソフトウェアの信頼性及びセキュリティの取組強化に向けて中間報告書より抜粋

図 4　情報システム・ソフトウェアの品質問題分析

2.2. エンタプライズ・ソフトウェア運用の現状

第1.1節でも述べているように、近年、あらゆる産業分野でソフトウェアが使われるようになり、共通的・実践的な科学技術としてソフトウェアの重要性が高まっている。携帯電話や情報家電などの各種情報機器に組み込まれた、組込みソフトウェアが急速に拡大していることは周知の事実であるが、情報システム等を構成するコンピュータ機器にも当然エンタプライズ・ソフトウェアが入っている。また、機能が停止、低下または利用不可能な状態に陥った場合には我が国の国民生活及び社会経済活動に多大なる影響を及ぼす「重要インフラ」システムにもエンタプライズ・ソフトウェアが存在する。こうしたソフトウェアは、システム自体の大規模化やネットワーク化により、規模や複雑度が増大する傾向にある。例えば、大手都市銀行の合併に伴う情報システム統合では、4 年半という歳月をかけて、総費用が3,300 億円、開発工数14 万人月、ピーク時にはシステムエンジニアを6,000人も投入して開発を行い、その規模や複雑度が莫大であることがよく分かる。

しかし、時にソフトウェアが正しく動作しないことにより、システム障害が発生する。JUAS（日本情報システムユーザ協会）が 2009 年 7 月に発行した「ソフトウェアメトリクス 2009」[2]によると、最近 1 年間で発生したシステム障害のうち、実に約半数の障害がソフトウェア不具合が原因であるという調査結果が出ている(図 5参照)。

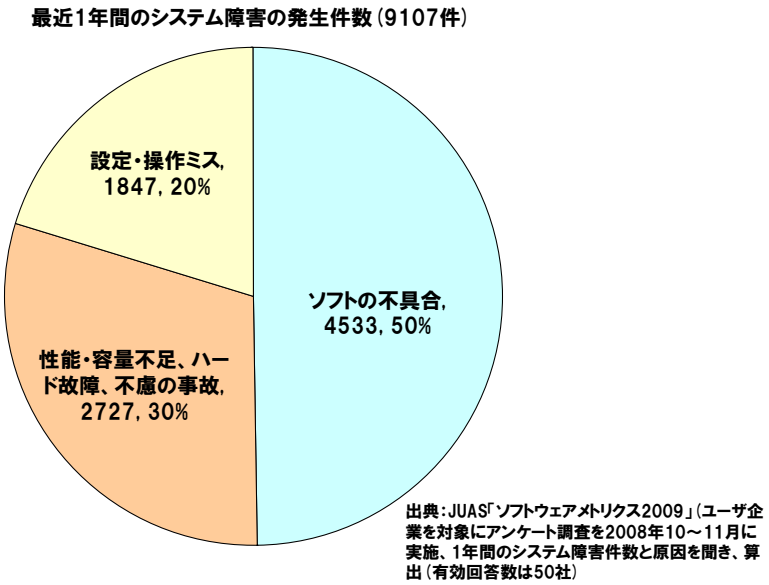


図 5　最近 1 年間に発生したシステム障害の原因分類

また、昨今のソフトウェア不具合が原因であるシステム障害は、事によると数百億円規模の損害賠償問題を発生させたり、交通機関に支障が出て数十万人の移動に影響が及んだりする場合もある（表 2参照)。これらのシステム障害は、ソフトウェアを含むシステムが

ソフトウェア発注者のビジネス環境や外部状況の変化に対応できていない、あるいはそれに対応するための仕様変更を行うときにソフトウェア不具合が含まれてしまうことがその主な原因である。また、誤作動が国民生活や社会経済活動に影響を及ぼすソフトウェアを体系的に構築するソフトウェア技術が現実の要求に充分に対応できていないことも一因である。

表 2　ソフトウェア不具合における最近の国内トラブルの事例

日付	不具合事例	不具合原因
2007年3月12日	りそな銀行と埼玉りそな銀行のネットバンクで障害が発生し、1万9000件の振り込みが処理ができず	他の銀行や支店などに振り込む際に起動するプログラムの一部に不具合
2007年5月23日	NTT東日本,NTT西日本合わせてひかり電話318万回線が3時間半にわたり不通	中継機器交換時にコマンド誤入力でデータが破壊
2007年5月27日	全日空国内線予約システム故障 130便が欠航、影響は4万人以上、キャンセルなどによる損失4.5億円	空港端末とホストコンピュータをつなぐルータ管理プログラムの設定ミスによるメモリー故障が発端で全システムに波及
2007年10月12,18日	首都圏鉄道の自動改札機と窓口処理機の障害のべ727駅、260万人の足に影響	中央コンピュータからのデータをICカードに書き込むプログラムにミス
2008年2月1日	NTT「緑の電話」の一部に障害	電話機本体に搭載されたソフトウェアの不具合
2008年3月15日	親和銀行（ふくおかファイナンシャルグループ傘下）のATMトラブル	指静脈認証ソフトウェアの不具合
2008年7月22日	東証でシステム障害、TOPIX先物など売買停止	データを蓄積する容量の上限値のパラメータ設定ミス
2008年9月12日	大和証券の取引所との接続不具合で注文通らず	先物取引に関連した株式注文システムのプログラムを変更

さらに、先ほどの図 5からも分かるように、人為的な設定や操作ミスから生じるシステム障害も全体の 20%を占めている。ソフトウェアの大規模化による運用の高度化とも相まって、人為的ミスから大きなトラブルを生じさせうる状況に、危機感も高まっている。

2.3. エンタプライズ・ソフトウェア生産革新に関連する日本の取組み状況

現在、エンタプライズ・ソフトウェア生産革新に関連する日本の取組み状況は次の通りである[9]。

上流工程における要求分析技術等の研究開発状況は、ゴール指向手法の研究においてAGORA 手法や変更管理手法が世界的に認められ、一定の成果が出ている。また、産業界でもIT ベンダ各社が集まった「実践的アプローチに基づく要求仕様の発注者ビュー検討会」や「非機能要求グレード検討会」の活動を通して、「発注者ビューガイドライン」、「システム基盤の非機能要求に関するグレード表」という成果を公開している。

モデル記述・検証技術等の研究開発状況は、従来、数学的理論の研究に強みがあったこともあり、特に代数仕様記述の分野で世界をリードしている。また、最近ではモバイルフェリカチップの開発、証券会社向けシステムへの VDM(仕様記述言語)適用や、コピー複合機内制御ソフトウェアや IC カードへの SPIN（モデル検査）適用、電力関連のシステムへの SMV(モデル検査)適用等、徐々にではあるが応用研究や実証プロジェクトも散見されるようになった。また、モデル検査に興味をもった企業が集まった「モデル検査によるソフトウェアテストの実践研究会」では、ツールの開発・公開や応用事例の蓄積を進めている。さらに、独立行政法人情報処理推進機構ソフトウェアエンジニアリングセンター(以下、IPA/SEC)では、既存のモデル記述・検証技術を利用して、ソフトウェア開発環境を構築する取り組みを始めようとしている。

自然言語処理の研究開発状況は、1980 年代から 90 年代初頭までに機械翻訳に多大な資源を投入した結果、言語処理技術の水準は向上し、現在もその水準は維持している。現在は、Web 中のテキストからの評判分析や高度検索といったテキストマイニング技術に関心が集中しており、自然言語処理をエンタプライズ・ソフトウェア開発に応用するような研究開発は行われていない。

このように、エンタプライズ・ソフトウェア生産革新を実現するためのいくつかの要素技術は揃ってきている。これから我々のプロジェクトは、これらの要素技術を組み合わせ、統合していく次のステップに移らなければならない。

3. ソフトウェア生産革新への研究開発課題

第 2 章ではエンタプライズ・ソフトウェア生産革新プロジェクトの背景として、エンタプライズ・ソフトウェア開発や運用の現状、及び、エンタプライズ・ソフトウェア生産革新に関連する日本の取り組み状況を述べた。このような状況の中、エンタプライズ・ソフトウェア生産革新プロジェクトは、すでにあるいくつかの要素技術を組み合わせて統合し、さらにその成果を実証する実証実験を実施するために、上流工程での信頼性向上に重点を置いた、以下の 6 つの研究開発課題を解決する必要がある（図 6参照）。

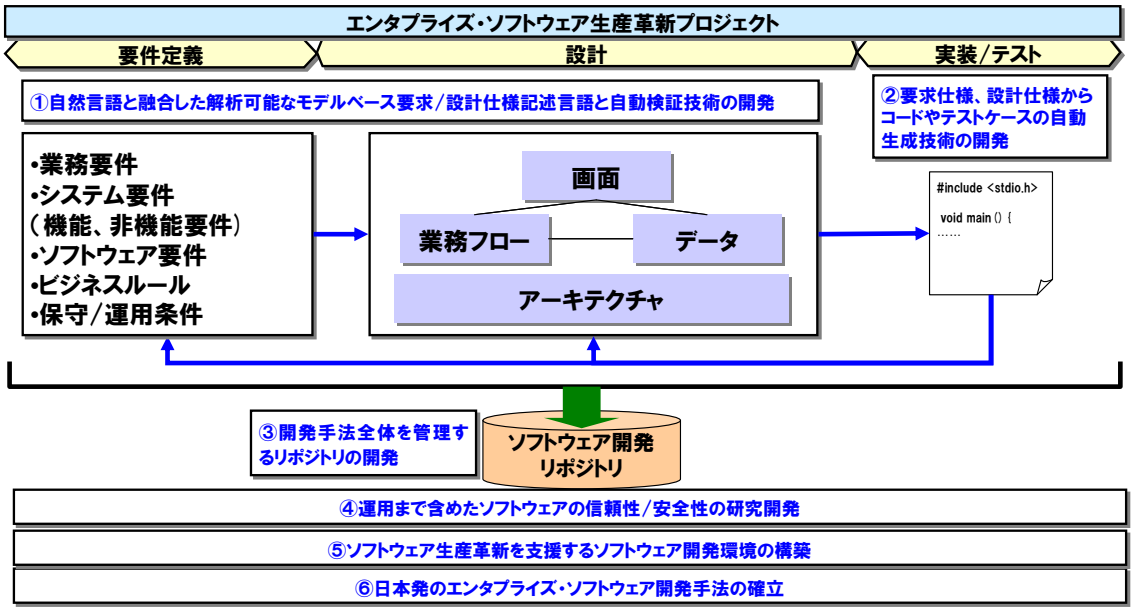


図 6　エンタプライズ・ソフトウェア生産革新プロジェクトの研究開発課題全体像

① 自然言語と融合した解析可能なモデルベース要求／設計仕様記述言語と自動検証技術の開発

解析可能なモデルベースの要求／設計仕様を記述・利用することで、要件定義工程や設計工程等の上流工程、いわゆる品質の作り込み工程で高い品質を保証する技術を適用する。厳密な仕様記述を定義するとともに、自然言語やUMLの要素を取り入れた、一般のソフトウェア技術者や関連するステークホルダのスキルレベルに合った表現形式を分析する。また、仕様と表現形式を明確に分離するために、自然言語やUMLの解析を駆使した仕様と表現形式間の変換ロジックを開発し、ステークホルダのスキルレベルにあったドキュメントを生成することを目指す。これにより、従来までは数学的スキルを持った一部のソフトウェア開発技術者のみが扱えた仕様記述・検証技術が、すべての一般ソフトウェア技術者でも扱え、関連するステークホルダに対しても仕様を簡単に理解することができる（図 7参照）。

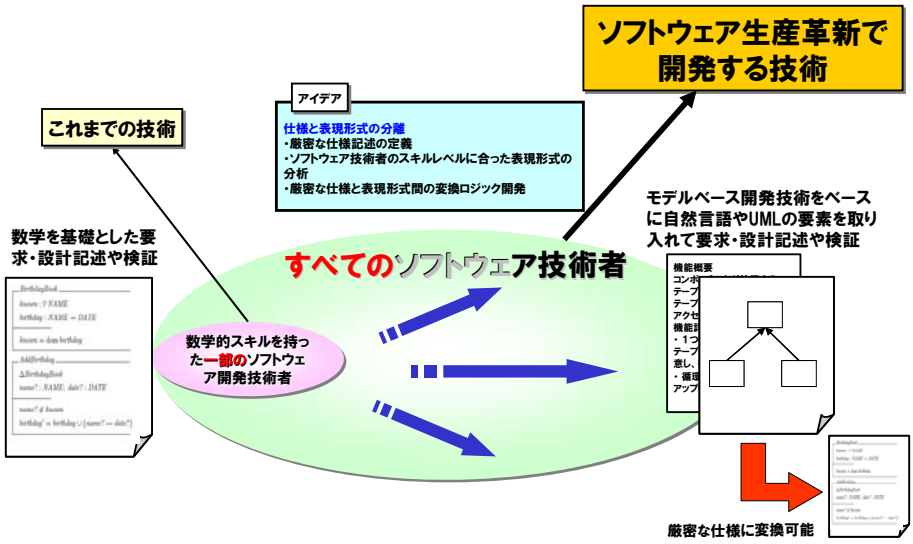


図 7　自然言語と融合した解析可能なモデルベース要求／設計仕様記述言語と自動検証技術開発による効果

検証技術で使用する「正しさ」の保証には、「正しくソフトウェアを作っているか」を確認すること、「正しいソフトウェアを作っているか」を確認することの 2 種類がある。これまでは、それら個々の検証技術が独立して発展してきた。しかし、上流工程から一貫して「正しさ」を保証するソフトウェア開発技術を目指している我々の研究開発課題は、「正しくソフトウェアを作っているか」の確認と「正しいソフトウェアを作っているか」の確認の両者を組み合わせていかなければならない。

② 要求仕様、設計仕様からコードやテストケース[2]の自動生成技術の開発

第 2.1 節でも述べているように、エンタプライズ・ソフトウェア開発現場におけるプログラミングやテストは旧態依然として人海戦術に大きく頼っている。例えば、ソフトウェア技術者が要求仕様や設計仕様を読み込んだ上で、そこから仕様を分析してテストケースを抽出しているが、これは非常に時間がかかる作業であり、かつテストケースの網羅性や整合性はソフトウェア技術者のスキルに依存する。この原因の 1 つが自然言語で記述された要求仕様や設計仕様の曖昧性であり、①は上流工程において要求仕様や設計仕様をコンピュータでも解析可能なモデルで記述し、検証することを目指す。さらに、本研究開発課題では解析可能な要求仕様や設計仕様を下流工程に適用することを目指すものであり、解析可能な要求仕様や設計仕様からソースコードやテストケースを自動的に抽出する。これにより、今まではソフトウェア技術者のスキルや人海戦術に頼っていたプログラミングやテストの下流工程からソフトウェア技術者を解放し、より上流工程の知的生産活動にシフト

[2]ソースコードに対してどのような値を入力すると、どのような結果になるのかを記述したもの

するように促すことが可能となる（図 8参照）。

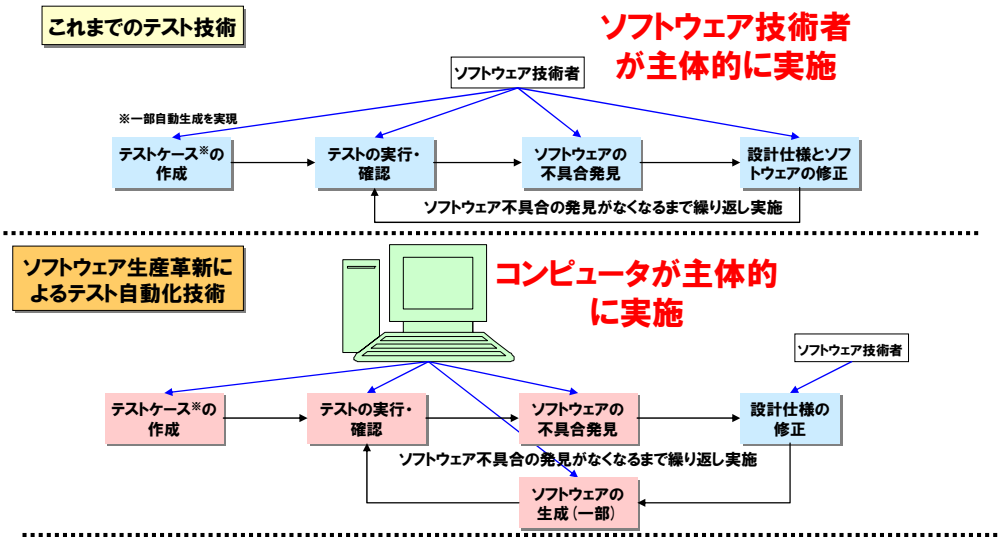


図 8　要求仕様、設計仕様からコードやテストケースの自動生成技術開発による効果

③　開発手法全体を管理するリポジトリの開発

複数のソフトウェア技術者が分担して記述した、解析可能なモデルベースの要求仕様／設計仕様、およびそこから生成されたソースコードやテストケースを管理するリポジトリを開発する。本技術は、分担して記述した仕様の統合や後工程の仕様に情報を引き継ぐトレーサビリティ適切に管理し、要求変更に対応した仕様の影響等を分析して、開発のライフサイクル全体の整合性を取ることを可能とする。また、保守開発においても、本リポジトリで管理する解析可能な要求仕様／設計仕様を再利用、変更すべき箇所を分析、明確化することにより、どの部分を修正すべきかを把握することを可能とする。

④　運用まで含めたソフトウェアの信頼性/安全性の研究開発

保守や運用まで含めたソフトウェアの信頼性/安全性の研究では、システムに割り付けられた役割と運用者に割り付けられる役割を含めた要求分析手法の研究、運用者起因の失敗についても対応可能なシステム設計手法の研究などが求められる。本提言で課題として認識している運用まで含めた「正しさ」の保証を行うためには、以下に例示するような一連の研究開発が求められることになろう。

- 運用まで含めたシステムモデル化の研究
- 運用者に割り付けられる役割まで含めた要求分析方法の研究
- 運用者による「間違い」「攻撃」などを含めた振る舞いのシミュレーションの開発
- ソフトウェア・システム設計において運用を含めた「正しさ」の実現を支援するツールチェーンの構築

⑤　ソフトウェア生産革新を支援するソフトウェア開発環境の構築

ソフトウェア開発環境は、プログラム生産性の向上を主軸として発達してきた。フレームワークとの統合、開発方法論やワークフローとの統合、ビジュアライズ、自動化といった機構である。品質強化の観点では、問題追跡ツールとの連携、テストの自動生成、テストの自動実行などが開発されてきている。

その一方で、「正しさ」を支援する環境は、「正しさ」を保証するツールの現場への浸透不足とも相まって、進歩していないのが現状であった。よって、①で述べている″上流工程から「正しさ」を保証するソフトウェア開発技術″を確立するとともに、それを支援するツールの開発や、開発方法論への「正しさ」保証の取り込みなどをソフトウェア・ツールチェーンの形で組み入れる研究開発を実施する必要がある。特に、ソフトウェア開発環境はエンタプライズ・ソフトウェアの生産現場で活動している一般の技術者が利用されなければ意味がない。そのため、一般の技術者が利用しやすい開発環境を常に意識しながら研究開発を進めるべきである。

⑥　日本発のエンタプライズ・ソフトウェア開発手法の確立

①～④の技術開発成果や⑤のソフトウェア開発環境を組み合わせて日本発のエンタプライズ・ソフトウェア開発手法を確立すべきである。現状のエンタプライズ・ソフトウェア生産現場での方法とすり合わせることと、今までの日本に特化した開発方法からよりグローバルを意識した開発方法を上手に融合させることにより、エンタプライズ・ソフトウェアの信頼性/安全性を向上させる新たなエンタプライズ・ソフトウェア開発手法を確立する。この新しいエンタプライズ・ソフトウェア開発手法は、海外のオフショア先でも利用可能なだけではなく、世界の中でも先進的な技術として、日本がリードできるような技術に成熟させる。また、それとともに、エンタプライズ・ソフトウェア開発手法の国際標準化を目指し、この開発手法を活用したエンタプライズ・ソフトウェアの信頼性/安全性を評価、認定する市場を形成すべきである。

これらの研究開発課題が解決し、世界最先端の革新的ソフトウェア開発手法の実現により、以下の効果が期待される。

(1) 社会的基盤を担うエンタプライズ・ソフトウェアにおいて、30%程度が開発フェーズでの問題発生が原因となるシステムトラブルである[18]。よって、本手法の実現により、ソフトウェア・システムの不具合が最大で 30%減少し、一般国民の社会生活の不安を和らげることになる。

(2) 開発上流工程から高い品質、高い信頼性を確保できるので、開発費を圧迫しているテスト工数を大幅に削減することができる。その結果、ソフトウェア開発企業の収益性が向上する。さらに、現状 1 億行を超えているエンタプライズ・ソフトウェアを含む 17.8

兆円の情報サービス産業市場において、コスト超過分 1 兆円のうち 8000 億円の無駄の削減、及び 1 年間に発生するシステムトラブルのうちの 30%である 1120 億円の損失防止を実現する。

(3) 信頼性技術の有無が参入障壁となっているエンタプライズ・ソフトウェアの市場を開拓することができる。近年、欧州を中心として機能安全、あるいはセキュリティに関わる製品の標準化制定と実施が活発化し、そこではソフトウェアの信頼性を客観的に保証する技術としてモデルベース開発を推奨している。今後は、要求モデル、設計モデルを利用して開発されたソフトウェアだけが市場流通できる方向に向かっていくであろう。現状は欧州市場ではじまりつつある動きであるが、今後、アジアを含む世界市場で同様な状況になると予測される。逆に、本研究課題の成果によって、すでに存在する標準化への対応とともに、新たな標準化制定と実施を目指し、国内の多くの企業が先行的に参入障壁を乗り越える技術力を保有できる。

4. エンタプライズ・ソフトウェア生産革新の提言

4.1. エンタプライズ・ソフトウェア生産革新への産学官推進体制の確立

日本は少子高齢化がかなりの速度で進行しており、すでに 2005 年より日本の総人口は減少期に突入している。今までは、情報サービス産業はソフトウェア発注者からの要求にきめ細かく対応したカスタムメイドのソフトウェアを開発していたため、ソフトウェア受注者は国内に閉じた形で競争していても産業が成立していた。しかし、このような状況下では、情報サービス産業でも日本の閉じた領域で競争していては、いずれ衰退することは目に見えている。よって、情報サービス産業に関わる企業は、日本の閉じた領域の中で競争するだけではなく、世界を見据え、世界中の関連する企業と競争していくグローバルな視点を持つことが重要である。現在、原子力発電等、他産業でも日本の国家プロジェクトが世界各国と競争しているが、産学官の連携不足のため、思うような成果が出ていない。このような中、国を支える高信頼で安全な社会インフラシステムを構築することは日本の国家プロジェクトとして非常に重要な施策である。それを実現し、世界と競争するためには、国、大学機関、企業が一体となって取組みを進めなければならない。また、昨今の長引く経済不況の下、多くの企業では研究開発に関する環境も急速に悪化しており、研究費の削減、研究人員を含む体制の縮小等、大幅な見直しを迫られている。各企業とも競争領域となる研究開発分野については、1 企業で閉じた研究開発は避けられない。しかし、協調領域となり得る研究分野については、中国、インド、東南アジア、南米等これから成長するであろう新興国をターゲット市場として、EU や米国を競争相手とすべきである。第 4.2 節でも述べているように、本プロジェクトに関連する要素技術はある程度、揃っている。本プロジェクトの実現のためには、それらの要素技術を組み合わせ、統合していく応用研究や実証開発が必要である。そのためには各企業が連携した連合体制確立や、本プロジェクトの研究開発に関連する有識者が在籍する大学・関連機関と連携しての共同研究実施、推進コミュニティの体制等を作ることが必要である。

日本が EU や米国との競争に勝つための手段としては、日本発の標準化を目指すことが必要である。EU は、すでにセキュリティ評価規格 ISO15408(コモンクライテリア)や機能安全規格 IEC61508 を先導しており、その分野においては世界をリードしている。よって、日本としては、先ほどの研究開発課題で出たエンタプライズ・ソフトウェア開発手法の国際標準化を目指し、この開発手法を活用したエンタプライズ・ソフトウェアの信頼性/安全性を向上させるところで競争するべきである。このような産業界が普及展開を主導して取り組む市場化フェーズの場合は、まず、産業界が協調領域となり得る研究分野を提示し、かつ産業界もそれ相応の資金を提供するようにしてリスクを背負いながら、産業界が学界や官界を先導して活動しなければならない。このような産学官が連携したトライアングル

体制を形成すべきである。

通常、研究開発を時系列で見た場合、大きく分けて基礎研究と応用研究に分類される。しかし、研究成果を産業界のビジネスで有効的に活用できるように進めるために、応用研究の中をさらに細分化し、産業界のニーズを取り入れ基礎研究成果を統合する応用研究フェーズ、実開発に対する実証実験等を行う実証フェーズ、産業界が中心となって普及展開する市場化フェーズが必要である。H21 年度の科学技術関係予算（3 兆 5,548 億円）の内訳（表 3参照）を見ると、（1）でも述べているように、政策課題対応型研究開発の取り組みが存在する。この施策自体も重点分野を絞って優先的に予算配分を実施するという観点から重要ではあるが、この施策ではどの時系列フェーズにおける予算であるのかが判断できない。その他を見ても、明示的に実証フェーズや市場化フェーズであると分かる研究予算は存在しない。特に、エンタプライズ・ソフトウェアを含む大規模なシステムの実証実験等を行うためには、多額の予算の確保や予算が実証実験に使用されることが明示的に了承されていることが必要である。本プロジェクトでも今後、明示的に研究予算がつくこのような大規模な実証実験等を実施し、産業界への普及展開を加速させるべきである。

表 3　H21 年度科学技術関係予算の内訳

H21年度科学技術関係予算（3兆5,548億円）

大学等の基盤的経費、科学研究費補助金等の基礎研究	政策課題対応型研究開発（重点推進8分野）	システム改革等（人材育成、理解増進、産学官連携、知的財産、地域イノベーション等）
1兆4,769億円	1兆6,869億円	3,910億円

4.2.　エンタプライズ・ソフトウェア分野への研究開発投資

EU の FP7(2007 年から 2013 年まで)全体の研究開発予算は 505 億 2,100 万ユーロであり、そのうち、情報通信技術（ＩＣＴ）への研究開発予算は、90 億 5,000 万ユーロで全体予算の 29.5%に相当する。また、情報通信技術の研究開発予算を時系列でみると、過去の FP5, FP6 は 4 年計画、FP7 は 7 年計画と期間の長さは異なるが、単純に 1 年単位で比較した場合、情報通信産業への研究開発予算は FP5 では 9 億ユーロ/年、FP6 では 9 億 9,600 万ユーロ/年、FP7 では 12 億 9,000 万ユーロと情報通信産業への研究開発予算は増加傾向である。

また、米国連邦政府が投資する情報通信技術に関連する NITRD の研究予算も時系列で見てみると、2007 年度 31.2 億ドル、2008 年度 33.7 億ドル、2009 年度 35.7 億ドルと米国でも情報通信技術関連の研究予算は増加傾向である。

一方、日本の政府が投資する政府研究開発投資では、各国の科学技術戦略を踏まえた我

が国の戦略や国民からの期待などを考慮して、ライフサイエンス、情報通信、環境、ナノテクノロジー・材料、エネルギー、ものづくり技術、社会基盤、フロンティアの重点推進 8 分野に優先的に予算配分を実施するH21 年度の政府研究開発予算（政策課題対応型研究開発費）が存在する。その予算は全体で 1 兆 6,869 億円であるが（表 3参照)、そのうち情報通信分野への研究開発予算は 1,580 億円で全体予算の 9.4%である。また、情報通信分野の研究開発予算を時系列でみると、2006 年度予算は 1,726 億円（9.7%)、2007 年度予算は 1,681 億円（9.9%)、2008 年度予算は 1,613 億円（9.3%）であり、ほぼ横ばいか若干減少傾向である。情報通信分野はさらに、ネットワーク、ユビキタス、デバイス・ディスプレイ等、セキュリティ及びソフトウェア、ヒューマンインターフェイス及びコンテンツ、ロボット、研究開発基盤の 7 領域に分かれる。本プロジェクトに最も関連がある領域は、セキュリティ及びソフトウェア領域であり、組込みソフトウェア開発、オープンソース・ソフトウェア環境整備の他にも、ソフトウェアエンジニアリングを活用したソフトウェアの信頼性への取り組みが行われている。しかし、その取り組みはステークホルダとの合意作業が必要である企画や要件定義工程等の上流工程やソフトウェア開発以外の保守、運用まで考慮する視点が欠けており、取組み活動としてはまだ不十分な状況である。

第 1.1 節でも述べているように、エンタプライズ・ソフトウェアはすべての産業に多大な影響を及ぼすインフラ的な存在であり、EU や米国はそれを踏まえて情報通信産業等に積極的な研究開発投資を実施している。日本にとってもそれは同様であるので、本プロジェクトの研究開発を加速させるためにも、情報通信技術やエンタプライズ・ソフトウェア分野にさらなる研究開発投資を行うことが重要である。

4.3. 具体的な施策例

ここでは、エンタプライズ・ソフトウェア生産革新プロジェクトをより加速させることができると考える具体的な施策例を紹介する。

（施策例１）IPA/SEC プロジェクトとの連携

第 2.3 節では現在の IPA/SEC の取り組みを述べたが、この取り組みは既存のモデル記述・検証技術を利用しているため、既存の要素技術を組み合わせて統合する応用研究が主体である我々のプロジェクトとは競合はしない。むしろ、我々のプロジェクトと IPA/SEC の取り組みとは補完関係にあるので、IPA/SEC のソフトウェア環境構築の成果を我々のプロジェクトに利用できるようにしていただく、あるいはソフトウェア環境を拡張することが可能となるような、強固な連携関係を築くべきである。

（施策例２）ソフトウェア調達における解析可能なモデル記述の推進

エンタプライズ・ソフトウェアの調達は、通常、ソフトウェア発注者が自然言語で記述する RFP(Request for Proposal)に対して、ソフトウェア受注者も自然言語で記述する提案書(Proposal)を作成、提案し、それを審査した後に契約に至る。発注者、受注者がともに、RFP や提案書に書かれるソフトウェアの要求事項そのものを解析可能なモデルで記述し、双方が理解することができれば、自然言語による曖昧性が取り除かれ、「正しいソフトウェアを作っているか」を確認するときの基準となる。また、ソフトウェア要求事項を解析可能なモデルで記述することによって構造化されるため、次のソフトウェア調達時の RFP や提案書に再利用しやすい利点がある。

上記を実現するためには、RFP や提案書に対して解析可能なモデル記述を推進するガイドラインを作成したり、公共系のエンタプライズ・ソフトウェア調達における RFP や提案書作成は解析可能なモデルで記述することを指定する等の施策を実施する必要がある。

（施策例３）技術成果に対する公共系システム開発への試行適用の推進

ソフトウェア開発技術の研究開発成果が真のエンタプライズ・ソフトウェア開発現場で利用できるようにするためには、産業界で実証実験や試行適用が必要である。しかし、エンタプライズ・ソフトウェアはソフトウェア発注者が発注したものをソフトウェア受注者が作るのであり、そのようなビジネスの枠組みの中で実証実験や試行適用を実施することは非常に難しい。そこで、例えば公共系のソフトウェア開発において、ソフトウェア開発技術の成果が試行適用に耐えることができるかを判定する第三者機関を設置し、審査が通った成果については、その技術成果を試行適用可能となる契約形態が望ましい。

4.4. 本プロジェクト実現までのロードマップ

本プロジェクト実現に向けて、第 3 章で述べている 6 つの研究開発課題を解決するためには、研究開発課題の順序付けを行って段階的に進める革新的ソフトウェア開発手法の研究開発と、それを行うための基盤整備及び支援を実施しなければならない（図 9参照)。

6 年後の 2015 年に「世界最先端の革新的ソフトウェア開発手法の確立」を目指し、2010-2012 年の 3 年間は、実証実験を行うための準備を行うステップ 1 と位置付ける。革新的ソフトウェア開発手法の研究開発は、上流工程を中心とした要求分析技術、モデル記述・検証技術、自然言語処理技術を組み合わせ、エンタプライズ・ソフトウェア開発手法に統合するための応用研究、及びその成果を利用したソフトウェア開発環境を構築する。基盤整備及び支援は、上記技術開発を効率的に行うために、IT 産業界や本プロジェクトの研究開発に関連する学の有識者を結集し、産学官連携による効率的な研究開発が可能な枠組み、及びコミュニティ体制を構築する。

また、2013-2015 年の 3 年間は、ステップ 1 の成果を利用して実証実験を行うステップ 2 と位置付ける。革新的ソフトウェア開発手法の研究開発は、ステップ 1 の研究成果や構築したソフトウェア開発環境からエンタプライズ・ソフトウェア開発手法を検討し、その手法が実開発の開発現場で利用可能かどうかを検証するための大規模な実証実験を行う。基盤整備及び支援は、新興国を市場とした欧米との競争に打ち勝つために、日本発のエンタプライズ・ソフトウェア開発手法の国際標準化への準備支援や積極的安情報発信、及び成果を産業界に展開するための移行準備を行い、次の普及定着フェーズへ移行させるための足掛かりを築く。

このように進めることによって、本提言は世界をリードする施策に成長させることができると考える。

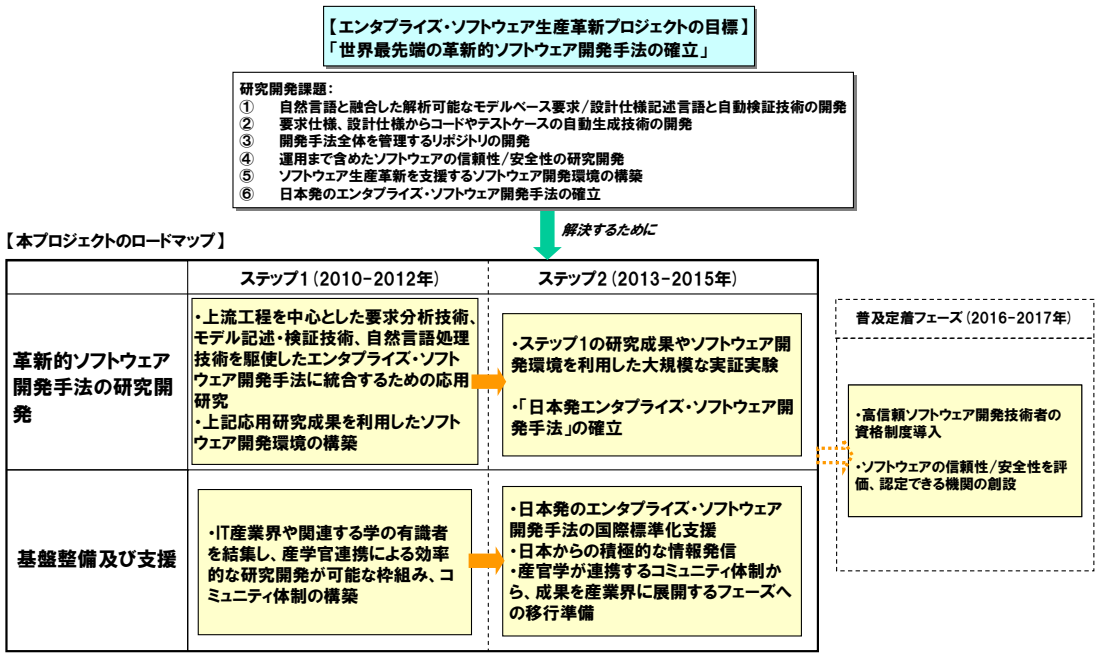


図 9　本プロジェクト実現までのロードマップ

おわりに

我々は、企業の業務システムや情報システム、あるいは社会基盤を支える情報システム、いわゆる「重要インフラ」システムの重要な一要素であり、また、すべての産業にとってもインフラ的な位置付けであるエンタプライズ・ソフトウェアに対象を絞り、エンタプライズ・ソフトウェアの開発現場やエンタプライズ・ソフトウェアの運用に関する現状を分析し、そこからエンタプライズ・ソフトウェア生産革新プロジェクトにおける研究開発課題を抽出した。研究開発課題は以下の 6 つである。

① 自然言語と融合した解析可能なモデルベース要求./設計仕様記述言語と自動検証技術の開発
② 要求仕様、設計仕様からコードやテストケースの自動生成技術の開発
③ 開発手法全体を管理するリポジトリの開発
④ 運用まで含めたソフトウェアの信頼性／安全性の研究開発
⑤ ソフトウェア生産革新を支援するソフトウェア開発環境の構築
⑥ 日本発のエンタプライズ・ソフトウェア開発手法の確立

上記研究開発課題は、要素技術を組み合わせ、統合する応用研究の色彩が強く、また、本プロジェクトに関連する日本の取組み状況より、すでに上記を解決するための要素技術は揃っている。よって、我々は上記課題を解決するために、産業界が主導し、本プロジェクトの研究開発課題を解決するための様々な有識者が結集する産学官推進体制を確立させることと、本プロジェクトの研究開発を加速させるために、エンタプライズ・ソフトウェア分野にさらなる研究開発投資が重要である旨の提言を述べた。また、その他にも本プロジェクトをより加速させることができると考える具体的な施策例についても紹介した。

今後は、本提言を主要各省庁、産業界、大学などに紹介・説明し、賛同やご協力を得ながら、実現に向けての活動を実施していく予定である。

参考文献

[1] 基盤ソフトウェア技術戦略の産官学協力委員会、提言書「安心・安全な社会を実現する基盤ソフトウェア作りを目指して」、2009 年

[2] (社)日本情報システム・ユーザー協会（ＪＵＡＳ）編　「ソフトウェアメトリックス 2009」、2009 年

[3] Frederick Phillips Brooks, Jr.著、滝沢徹・富沢昇・牧野祐子翻訳 「人月の神話 – 狼人間を撃つ銀の弾はない」(20 周年記念増訂版、新装版)、ピアソンエデュケーション、2002 年

[4] 中島震、ソフトウェア工学の道具としての形式手法, NII TR-007J

[5] (財) 日本自動車研究所: 自動車電子システムの海外動向調査報告書, ITS 規格化 S08-1

[6] EU 研究開発に関するポータルサイト
http://cordis.europa.eu/home_en.html

[7] U.S. NITRD ポータルサイト
http://www.nitrd.gov/

[8] 米国国立標準技術研究所 Planning Report 02-3 ″The Economic Impacts of Inadequate Infrastructure for Software Testing″、2002 年

[9] 独立行政法人科学技術振興機構、「電子情報通信分野　科学技術・研究開発の国際比較 2009 年版」、2009 年

[10]経済産業省、高度情報化社会における情報システム・ソフトウェアの信頼性及びセキュリティに関する研究会、中間報告書「情報システム・ソフトウェアの信頼性セキュリティの取り組み強化に向けて　～豊かで安全・安心な高度情報化社会に向けて～」、2009 年

[11]経済産業省、「情報システムの信頼性向上に関するガイドライン第 2 版」、2009 年
http://www.meti.go.jp/press/20090324004/20090324004.html

[12]文部科学省、「平成 21 年度版科学技術白書」、2009 年
http://www.mext.go.jp/b_menu/hakusho/html/hpaa200901/1268148.htm

[13]独立行政法人新エネルギー・産業技術総合開発機構（ＮＥＤＯ）、「技術戦略マップ 2009」、2009 年
http://www.nedo.go.jp/roadmap/index.html

[14]経済産業省、特定サービス産業実態統計調査
http://www.meti.go.jp/statistics/tyo/tokusabizi/index.html

[15]総務省　「ICT の経済分析に関する調査報告書」、2008 年
http://www.soumu.go.jp/johotsusintokei/link/link03.html

[16]総務省、平成 21 年度科学技術研究調査
http://www.stat.go.jp/data/kagaku/2009/index.htm

[17]NICT パリ事務所、「EU の第 7 次枠組み計画における情報通信技術研究の動向調査」、2007 年
[18]経済産業省、独立行政法人情報処理推進機構、社団法人日本情報システム・ユーザ協会、「重要インフラ情報システム信頼性研究会報告書」、2009 年
http://sec.ipa.go.jp/reports/20090409.html

付録

1. ソフトウェア生産革新技術への各国の取り組み状況

1.1. はじめに

ソフトウェア生産革新の要点は、ニーズ探索・基礎および応用研究・実証および市場化の３つの段階からバランスよく技術開発を進めることにある。必要とされる技術像を明確にすることが肝要であり、ソフトウェア開発に関わる産業界と基礎研究を担う学界の連携が重要な役割を果たす。基礎研究の成果は産業界での利用に耐えるものだけが生き残る。一方、科学の成果として得られている理論的な限界を無視した「錬金術」のような技術はあり得ないことも事実である。残念ながら、ソフトウェアは一般の実感からかけ離れた実体感のない工業製品であることから、人々の心の中に錬金術が忍び込む隙間が生じる。ソフトウェアならびに関連する研究分野での成果を踏まえた技術の理解が大切である。

本節ではソフトウェア生産革新技術に関して、産学連携を中心とする研究開発支援の枠組みと技術発展方向の２つの観点から、欧米の状況を整理する。研究開発支援の枠組みに関しては、EU が戦略的な取り組みを行っており、系統的な情報を入手することができる。一方、US は世界的な企業を中心とする産学連携の研究活動やエンジェルと呼ばれるベンチャー企業投資家による資金提供が活発であり実体が見えにくい。以下では、欧州の取り組みを中心に紹介する。EU が主導する枠組みは、オープン国際標準化を前提としたものであり、ロードマップ作成・予算配分・研究推進に関して、科学技術に関わる制度設計の観点からも興味ある調査対象である。技術発展の方向については、ソフトウェア生産革新技術として、欧米を中心に研究開発が活発化している２つの技術、自動検証とソフトウェア発展の要点を解説する。

1.2. 欧州における研究開発支援の枠組みと現状

欧州では EU が主導する戦略的な研究開発支援の重要性が高い。1984 年ルクセンブルグ宣言により、産官学連携によるイノベーション実現、企業単独あるいは垂直統合型からオープンな EU 域内の国際分業へと転換することが示された。これが、FP （Framework Programs）として現在まで続いている。相前後して、当初フランスが主導して、市場志向の研究開発ネットワークに力点を置く EUREKA がはじまった。基礎研究から支援の対象とする FP、市場志向が明確な EUREKA の２本立てになっているといえる。さらに、各国の独自研究予算によるプロジェクトがあることも付記する。

2000 年リスボン宣言により、「世界で最も活発かつ競争力のある知識立脚型社会」に向けて、研究領域と研究助成の大枠を設定し、研究者をネットワーク化していく方法として、ERA（European Research Area） の活動がはじまった。GDP の 3%が研究開発投資の目標値として示された。このうち、2%は民間投資としており、そのためか、欧州の研究プロジェクトは（EU 域内）多国籍であると同時に産学連携のものが多い。2005 年から研究領域決定の仕組みが明確化され、民間主導でロードマップを作成する ETP（European Technology Platform） の活動がはじまった。研究開発投資目標値を達成するためには、民間投資を促進することが大切であり、ETP ロードマップが FP の研究投資に反映される。さらに、2007 年からは重点分野への集中投資と市場化を促進する JTI（Joint Technology Initiative）が動きはじめた。ETP と JTI の組み合わせによって産官学連携の動きを加速・推進する仕組みが整った。現在の対象分野は 6 項目であり、その 1 つとして、情報分野では、組込みシステムがあげられている。自動車・交通、航空・宇宙、鉄道、モバイル、等の共通基盤技術として、広範な産業分野に影響を与えることが理由である。

ソフトウェア技術は、EU FP においては、ICT（Information and Communication Technology、「情報技術」と「通信技術」）の領域に入る。膨大な研究開発支援投資が実施され、現在、2007 年からはじまった FP7 による研究支援が中心活動になっている。研究実施については複数の方法を並立させている。学界の研究者が中心となる研究活動だけではなく、ICT-FP7 プログラムとして実施される大型案件では、産学連携プロジェクトの形をとる。後に、ソフトウェア関連の ICT-FP7 プログラムについて技術内容を紹介する。中長期的な研究動向の良い整理になっている。

ICT-FP7 の支援対象は純粋な研究開発プロジェクトだけではなく、いくつかの形態がある。IP（Integrated Project）は技術研究開発と産業界での実証・展開を組み合わせた大型プロジェクト、STReP（Specific Targeted Research Project）は特定研究テーマの実行、NoE（Network of Excellence）はある分野の知識集積を目的とするコンソーシアム的な活動である。NoE としては、組込みシステム分野では ARTIST2（FP7 でも継続し現在は ArtistDesign）、ソフトウェア技術分野では S-Cube、などがある。活発な活動で知られる ARTIST2 は学界と産業界をつなぐ役割を担い、国際学会・ワークショップの主催・協賛、技術教育コースの提供、標準化に向けた作業などを行っている。すなわち、技術マップの整備、研究開発活動だけではなく、教育・技術移転までを一貫して行う。

ICT-FP7 プログラムは 7 年間の計画で、2007 年に公募、2008 年からプロジェクトがはじまり、予算総額はEUR 9.1Bである。2007-2008 年の 2 年間だけで EUR 2B（約 2,800 億円）に達し、7 つのチャレンジ領域とFET（Future and Emerging Technologies）に分

類される。チャレンジ1のPervasive and Trustworthy Network and Service Infrastructures（EUR 585M）はソフトウェア技術を含み、6つのサブチャレンジ項目に細分されている。2009-2010年は総額EUR 557M（約700億円）を図 10に示した6サブチャレンジ項目に割り当てる。

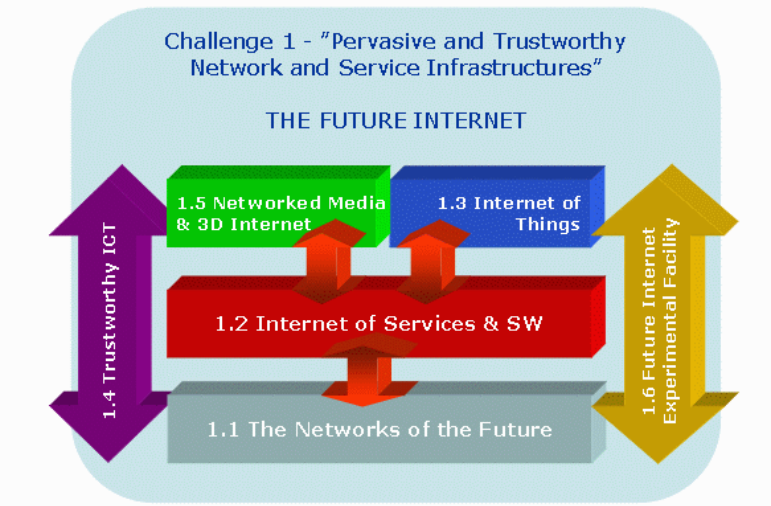


図 10　ICT-FP7 チャレンジ1の概要

ソフトウア生産革新技術は、チャレンジ1.2 Internet of Service & SW （図 10）の構成中、Service/Software Engineering : Complexity, Dependabilityが関連し、現在、8プロジェクト（図 11参照）が動いている。その中で、DEPLOYはIPであって、2007年からの5年間でEUR 17.885M（約23億円）の予算規模を持つEU内国際プロジェクトである。2004年から3年間実施されたFP6 RODINの後継であり、鉄道輸送（Siemens社）、自動車（Bosch社）、宇宙（Space Systems社）、ビジネス情報（SAP社）での実用化を目指す産学連携プロジェクトである。その他の7つは2007年からの3カ年プロジェクト（STReP）であり、各々EUR 3.5MからEUR 5.5Mの予算配分になっている。

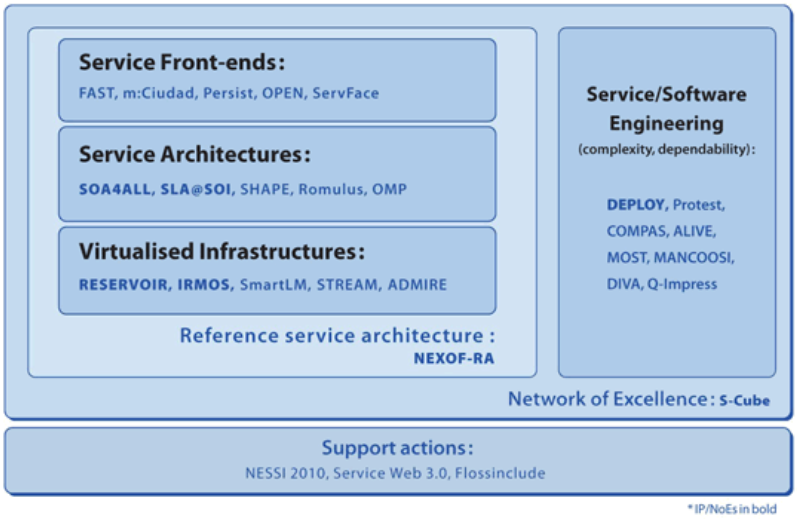


図 11　チャレンジ 1.2 の具体的な研究プロジェクト

現在審査中の 2009-2010 年公募では、チャレンジ 1.2 について、図 12のようなビジョンが示されている。チャレンジ 1 総額予算の 20%強に相当するEUR 110M（約 150 億円）を予定している。革新性（Highly Innovative）の要件として、検証（Verification）が重要な技術観点になっていることがわかる。

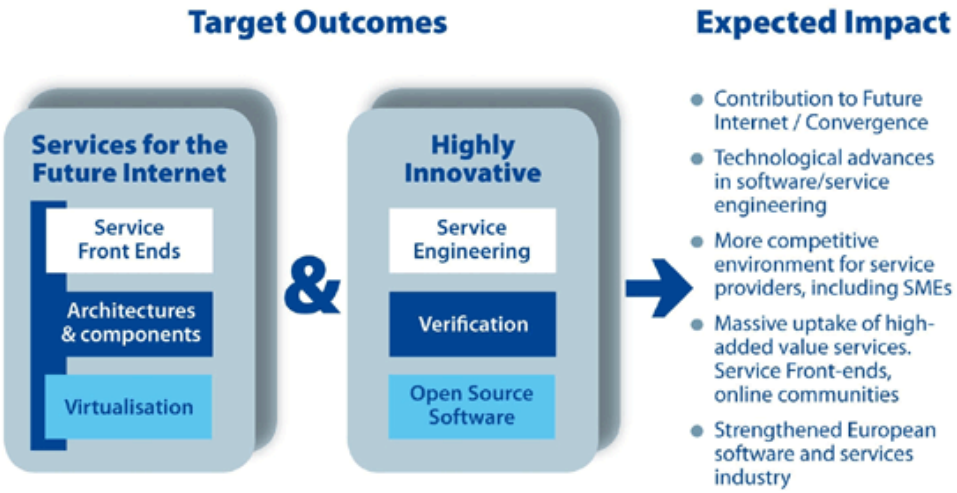


図 12　チャレンジ 1.2 のビジョン（2009-2010）

ICP-FP7 の FET は 7 つのチャレンジ分野にとらわれることなく ICT 全領域を対象とし、新しいアイデア・萌芽的なテーマや長期的な研究課題に投資する。FET-Open はボトムアップ（シーズ指向）の探索的研究を、また、FET-Proactive は社会あるいは産業界のニーズから描かれた長期研究課題への初期的な研究支援を担う。学術研究として興味深い研究テーマが多い。FET-Open への提案は随時受け付けという公募方法による。現在、17 テーマ

（STReP）が実行中である。FET-Proactive は領域カテゴリを明示して研究提案を募集する。ソフトウェア技術の分野では、最近数年、研究が活発になっている「自己適応型システム（self-* systems)」に関係するカテゴリとして、2007 年募集の PERADA、2009 年募集の AWARENESS（EUR 15M を計画）が目立つ。将来のソフトウェア・システム像を描いて長期的な研究投資を行っていることがわかる。

研究活動を実施する側の大学・研究機関が中心の集まりとして、ERCIM（European Research Consortium for Informatics and Mathematics）がある。1985 年以来、研究機関コンソーシアムとして活動している。ERCIM では FP への入力となるロードマップを学界の立場から作成し、この活動には EU 域外部の専門家も参加している。現時点では FP8 に向けたロードマップを世界中の英知を結集して議論している。さらに、研究者の ICT-FP 等への研究提案応募の支援、FP プロジェクトの実施運営を行う。また、ERCIM-WG で活動している研究グループの多くが FP7 の予算配分を受けている。ソフトウェア革新生産技術に関係する WG としては、FMICS（Formal Methods for Industrial Critical Systems)、MLQA（Models and Logics for Quantitative Analysis)、STM（Security and Trust Management)、Software Evolution（EVOLVE)、SERENE（Software Engineering for Resilient Systems)、Dependable Software-intensive Embedded Systems などがある。研究活動の成果は著名な国際学会ならびに各 WG が主催するワークショップで公表される。なお、これらのワークショップは学術集会であり研究発表を含めて参加はオープンである。

最後に、各国個別の研究プロジェクトについても簡単に触れる。ソフトウェア生産革新の分野では、2002 年からはじまった英国グランドチャレンジの影響が大きい。いわゆる学界中心の活動であり、研究開発支援の枠組みではない。具体的なプロジェクトテーマとしては、Verified Compiler と Verified Repository がある。前者は自動検証機能を組み込んだコンパイラ言語とツールの研究開発、後者は多様な検証事例を収集して利用可能とする活動である。Verified Repository では異なる手法で検証したコンポーネントを組み合わせて新たなシステムを構成する際に、既存の検証結果を再利用できるようにするための「形式手法のインターオペラビリティ」という新しい研究を含む。英国グランドチャレンジは IEE、BCS、IFIP 等の専門学会を活用して、世界中の研究者を巻き込む活動になっている。

ドイツ国内の研究プロジェクトとして、 Verified in Germany をキャッチフレーズとした Verisoft / VerisoftXT を紹介する。自国の産業力強化を目的にはじめられた産学連携の実証プロジェクトである。産学連携というと、普通は、学側が提供した要素技術をもとに産業界パートナーが実証研究を行うというイメージが強い。しかし、VerisoftXT は逆であって、産業界で開発中のプログラム自動検証ツールを用いる点が興味深い。マイクロソフト研究所が開発中の VCC（Verified C Compiler）を用いて、Windows 仮想化ミドルウェア

Hyper-V の全体を検証する。使っている技術は 1970 年代に考案されたもので、標準理論といってよい。一方、Cプログラムの性質を具体的に表現することが難しい。技術の難しさが要素技術から適用技術に移ってきている。この新たな課題である適用技術の研究開発に大学が寄与する。さらに、Hyper-V の検証作業を通して得たノウハウを活用して、他のプロジェクト参加企業がリアルタイム OS のセキュリティ認証を行うことを計画している。セキュリティの分野ではコモンクライテリアで規定されたセキュリティ品質を達成することが必須である。一方、最高レベル（EAL6 あるいは 7）を達成することは技術的にチャレンジングであり、VerisoftXT の目標（Verified in Germany）になっている。

1.3. U.S. における研究開発支援の枠組みと現状

従来、北米では、トップダウンな国家戦略による分野（セキュリティ関連）、産業の基盤分野（マイクロプロセッサ）で、高度な信頼性を達成する技術の研究開発が進められてきた。1990 年代後半、Microsoft 社がソフトウェアの信頼性向上に対する研究投資を活発化させた頃から研究対象の分野が広がった。同時に、実用レベルの適用を目指す方向への転換が目立つ。

国家戦略的な面では、重要技術分野に関する提言書が発行されている。ソフトウェア生産革新に関連するディペンダブルシステムの分野では、2007 年に National Academies から “Software for Dependable Systems” が発表された。広い意味でのディペンダビリティを高めるために必要な技術開発、組織体制等に関わる議論が含まれている。このような提言書が予算配分や研究の方向性に影響を持つと思われる。

研究戦略の中心を担う NITRD（Networking and Information Technology Research and Development）は、DARPA, NSF, NIST, NASA, NSA といった 13 の機関を取りまとめる形をとる。PCA（Program Component Area）と呼ぶ 8 つの研究分野を扱っている。その中で、ソフトウェア技術に関わる分野は、HCSS（High Confidence Software and Systems）、CSIA（Cyber Security and Information Assurance）、SDP（Software Design and Productivity）などである。大学・公的研究機関への予算配分は NSF が主である。ソフトウェア技術は NSF-CISE の分野になる。膨大な件数であることから、具体的な研究内容の全貌を把握することは難しい。発表される研究論文に NSF 支援であることがクレジットされているので関連するプロジェクトであることがわかる。

NITRD の具体的な活動の中には、研究センター設立への予算配分もある。CSIA で複数のセンター活動があり、たとえば、TRUST（Team for Research in Ubiquitous Secure Technology）は次世代組込みシステムの分野で、全米の有力な研究大学の教員がメンバとなり産学連携の研究活動を行っている。欧州での NoE（Network of Excellence）と同様な役割とも考えられる。TRUST について特筆すべきことは、欧州の研究グループ交流（ARTIST 等）との窓口になっていることであり、数回のワークショップを開催している。偶然か、これらの欧米の交流に相前後して、著名な研究者が北米から欧州に移籍した例がある。一方、欧州からはポスドク相当の若手研究者が北米の著名な大学に職（研究に専念する助手等）を得る事例が多い。人的な交流も活発であることがみてとれる。

2010 年度予算計画書（2009 年 5 月）によると、NITRD 全体で$3,926M（2009 年度実

績予測値は$3,882M)、すなわち約 3,500 億円である。内訳は、HCSS が$215.0M（同$320.1M)、CSIA が$342.5M（同$320.1M)、SDP が$120.6M（同$113.0M）となっている。同書では各 PCA の研究テーマの要点が明記されている。2010 年度版にも、ソフトウェア生産革新技術に関係する技術課題が HCSS の High-confidence systems and foundations of assured computing など、ディペンダビリティ関連としてあげられている。特徴的なことは、将来の応用システムとして CPS（Cyber Physical Systems）を描き、これを多様な要素技術研究の統一的な目標にしていることである。すなわち、CPS 研究を推進する中で、ディペンダブルシステムや組込みシステム一般で必要な要素技術の研究開発を行っている、と言える。

北米では産学の人的交流が柔軟に行われており、明示的、トップダウンな方法をとらなくても、従来から産学連携の研究活動が見られた。教員雇用の特徴をいかし、学期外の夏期休暇期間やサバティカル制度を利用して、教員が企業研究所に滞在し研究活動に専念する。さらに、博士課程学生が夏季インターンとして企業研究所で働く制度になっており、著名大学の優秀な学生は引く手数多である。学生側にとっては、企業ニーズに合った研究活動に触れる良い機会である。インターン成果が著名国際学会で発表されることも多い。実際、自動検証の技術（並行システムのロジック・モデル検査）が学側から産業界に浸透していく過程（1990 年代）では、夏季インターン学生が果たした役割が大きいと聞く。

北米には世界規模の企業（Microsoft, Intel, IBM, 等）が多数あるため、産学連携の活動は企業側が主導権を握っているという印象が強い。以前より、先に述べたサバティカルやインターン学生の制度が活用されている。また、人材交流も活発であり、テニュアでない若手教員は所属組織を移っていく。研究テーマへの興味、研究費獲得という２つの面があると聞く。さらに、企業側は、ビジネスの戦略上必要であるが自社にない先端技術について、若手教員に対して意欲的な人材獲得を行う。このように、草の根的に産学連携が有効に働いていることが北米の特徴であろう。

世界規模で活動している企業では、国際的な NoE を自社の活動の中で展開できるという強みがある。たとえば、Microsoft Research はアメリカ西海岸の研究所以外に世界各地に拠点を持つ。ソフトウェア生産革新技術については基礎的な研究から行う英国ケンブリッジ研究所、さらに、実用レベルの技術開発を行うドイツ・アーヘンの EMIC（European Microsoft Innovation Center）などと協力体制にある。歴史的には IBM は古くから世界各地に研究所を持つ。なかでも、形式手法と呼ばれる分野を切り拓いたのは、IBM ウィーン研究所のグループであることは良く知られている。現在、イスラエル・ハイファ研究所の VLSI 形式検証グループが著名である。

1.4. ソフトウェア生産革新の技術発展方向

ソフトウェア生産革新を達成する中核的な技術は、客観的かつ系統的な開発の方法である。数理論理学に基づく基礎理論をもとにソフトウェアの表現と解析の技術として、1970年代から研究が続けられている。約40年の積み重ねによって、ソフトウェアの信頼性向上に必須の自動検証技術が産業界での実用化を目指す技術開発の段階に達している。以下、自動検証技術に基づく信頼性向上の技術の概要と発展の方向について、欧米での主要な研究活動を引用しつつ説明する。

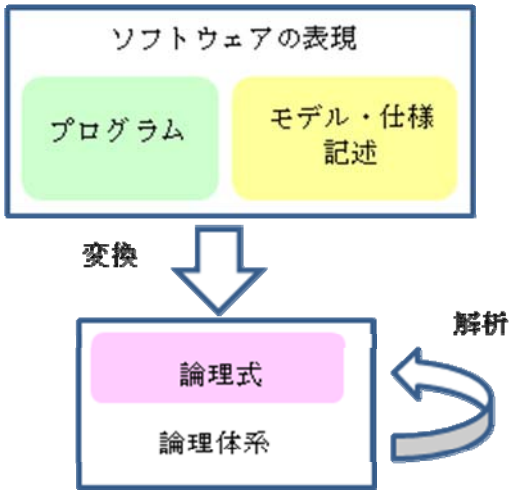


図 13　モデル・仕様表現あるいはプログラムと解析・検査エンジン

信頼性向上の技術は、ソフトウェアを表現する技術（形式仕様言語、プログラミング言語）と解析・検査する技術（形式検証）からなる。図 13に模式的に示すように、技術者が作成するソフトウェア表現の記述（モデルあるいは仕様と呼ぶ）を、自動検証可能な論理体系に翻訳して検査する方法がとられる。ところが、数理論理学の理論的な成果によって、モデルや仕様記述あるいはプログラムの自動検証は不可能であることがわかっている。万能な方法を目指すことは錬金術を求めることに等しい。理論的な限界の中で、産業界での利用に耐える工学的に有用なツールの研究開発に関心が移っている。

ソフトウェアの信頼性向上への応用に適した論理体系にはいくつかあることが知られている。表現力あるいは複雑さによって、自動検証が可能な命題論理から自動検証ができない論理体系までがある。自動検証できない体系を用いる場合は技術者が対話的に検査の作業を行う必要があり、その結果、用いている論理体系に習熟した技術者が必要となる。

ソフトウェア開発の上流工程では多種多様なモデルや仕様記述を作成する。一方、最終

成果物はプログラミング言語で書かれたプログラムである。上流工程の生成物とプログラムは性格を異にすることから、研究開発の流れが 2 つに分かれている。どちらも重要な技術であり、一方があれば他方は不要というわけではない。図 14では、上流工程での信頼性向上とプログラム検査の２つの方向に技術開発が分化していることを示す。1990 年代初頭に欧米の研究者が集まり会議が開催された。欧州は上流工程生成物の信頼性向上に、北米はプログラム検査の研究開発に力点を置くことが議論された。実際、その後、そこでの議論の方向にしたがった研究活動が見られる。

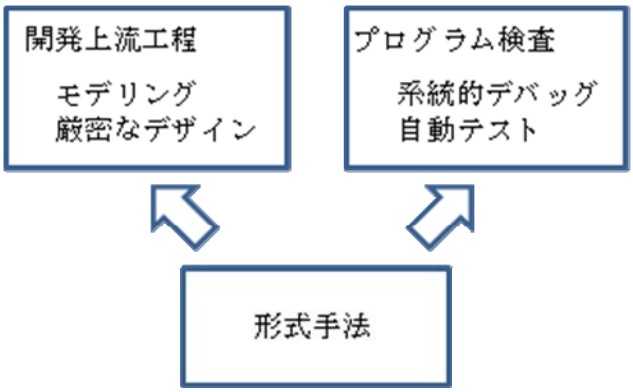


図 14　２つの方向への発展

プログラム検査の技術は、作成されたソースプログラムを入力とし、メモリリーク等のプログラム共通性質の検査（sanity checking）ならびに個々のプログラムが実現するアプリケーション性質が満たされているかを判定する。検査の仕組みを理解しなくても使える自動解析ツールのコマンドとして実現することが期待される。一方、プログラミング言語はチューリング機械等価といわれ、一般に自動検証することは不可能であることがわかっている。そこで、検査性質を限定する、プログラムの書き方を制限する、検査に必要な情報を補助的に与える、等の工夫を盛り込んだ専用ツールとする。さらに、不具合がないことの検証は難しい場合が多く、その結果、系統的な自動デバッグ・ツールとして開発されることもある。形式検証技術を基にした自動デバッガである。

現在、プログラム検査の研究を精力的に行っているのは、マイクロソフト研究所である。膨大な OS プログラムの資産を持つことから、プログラム自動検査技術による信頼性の確保を重要視している。2001 年頃から、SLAM, SAL, Terminator などの専用ツールを開発し、2009 年には VCC（Verified C Compiler）を公にした。これらのツールは、1999 年にオクスフォード大学から英国ケンブリッジのマイクロソフト研究所に移った T. Hoare 博士の Verified Compiler 計画を実現する具体的な研究事例とも考えることができる。最新の VCC 関連では Windows 仮想化ミドルウェア Hyper-V（約 10 万行）の正しさを検証する実証研

究がある。その他、北米では、SRI、VLSI 関連の Cadence 社や NEC Laboratories America でもソフトウェア信頼性向上技術やプログラム検証の研究が進められている。

開発上流工程のモデルや仕様記述を対象とする研究は、プログラム検査に関わる研究に比べて、多様性が大きい。開発対象ソフトウェアの種類・性質ならびに開発の方法に依存して、上流工程で作成するモデルや仕様記述が異なるからである。その中で、FP7 の DEPLOY プロジェクトが研究者の間で注目を集めている。DEPLOY は形式仕様言語 Event-B を中心とするツールと開発方法の全体を取り扱い、システム分析に関して、リファインメント・モデリング（refinement-based modeling）と呼ぶ新しい方法を提案する。正しさの確認が容易な単純な初期仕様から出発し、段階的な詳細化手法で具体的な仕様を作成していく。詳細化の各段階に対応するリファインメントの正しさを確認することで正しい仕様を導出する。従来は、リファインメント関係の検証に対話的な検査の方法を用いていた。DEPLOY では、後に述べるような最新の自動検証エンジン（SMT エンジン）を統合する研究も行われている。

産業界の視点では、現在、プログラムの高い信頼性を達成する方法として、プログラム・テスティングを採用している。テスト対象システム（System Under Test）の規模が大きくなると共に、テスト規模が膨大になる。テスティングに頼った信頼性確保の技術だけは、開発費用が膨れ上がり、ソフトウェア産業の収益を圧迫する。その結果、先に述べたプログラム自動検査と適切にすみ分けることで、テスティング費用の削減が期待されている。そのためには、テスト自動生成の技術である SBT（Specification-Based Testing）や MBT（Model-Based Testing）の技術開発が必要である。SUT が満たすべき仕様やモデルの記述から単体テスト・データを自動生成する。仕様やモデル記述からテスト生成に必要な情報を抽出することで、適切なカバレッジ基準を満たす無駄のないテスト・データを自動生成する。仕様やモデル記述の与え方、選択したカバレッジ基準の違いにより、多くの方法が研究されている。たとえば、2007-2008 年 ICT-FP7 のチャレンジ 1.2 では PROTEST プロジェクトがある。マイクロソフト研究所では後に述べる Z3 をエンジンとして用いた SBT ツールを C#プログラミング言語向けに開発している。

プログラム検査、モデルや仕様記述の検査、テスト自動生成は、技術者からみた場合の使い方は大きく異なるが、ツールを実現する基本技術、要素技術には共通点がある。将来必要となるツールの実現に際して必須となる技術要素が整理できてきたということもできる。図 13を再び参照すると、模式的に示されているように、論理系を対象とした検証エンジンが共通基盤技術になっていることがわかる。原理的に自動検証できる場合であっても、検査に要する計算時間が膨大になり、ツール作成の観点からは検査打ち切りとすることが多い。すなわち、自動検証の技術は、うまくいけば成功するが、そうでない場合ツールは

検査に失敗し、検査対象が正しいか誤っているかの判断ができない、というものである。なお、通常のソフトウェアを検査対象としている限り、多くの場合、自動検査は成功する、ということが経験的にわかっている。

自動検証ツールは、命題論理式を対象とする SAT エンジン、SAT の基本アルゴリズムを拡張して決定可能な問題の決定手続きを組み込んだ SMT エンジン（Satisfiability Modulo Theory）、一般の 1 階論理に対する部分手続きを組み込んだエンジンなどがある。2000 年頃から SAT エンジンの性能を競うコンテストが開催されて性能向上技術の発展に寄与した。商用 SAT エンジンの中にはコンテストに参加しないため、その性能を客観的に評価できないと批判されるものもある。また、SMT エンジンに関しても同様なコンテストがあるが、毎年のようにチャンピオンが入れ替わるほど技術発展のペースが速い。米国 SRI（Stanford Research Institute）の Yices、Microsoft Research の Z3 が高い性能を示すことで著名である。Z3 は非常に高速なエンジンであり、マイクロソフト研究所の各種ツールの基本エンジンになっている。面白いことに、Yices と Z3 のツール開発者は同一であり、属人的な技術ノウハウが重要な世界になっていることがわかる。一方、スイスの University of Lugano 等を中心にオープンソースのプロジェクト OpenSMT が始まった。世界中の英知を結集した新たな技術革新が期待できる。今更、全く新しい検証エンジンをゼロから研究開発をはじめることは考えにくい。

1.5. ソフトウェア発展の技術発展方向

前節に述べた自動検証を中核とするシステム信頼性向上の方法は、ソフトウェア開発過程で利用する技術である。一方、利用者からみたシステムの信頼性は、開発過程だけではなくシステム運用を含むライフサイクル全体に関わる。開発過程で不具合を減らし信頼性を向上させたとしても、それは、開発時に想定した範囲内での話である。ここでの疑問は、「システムの利用形態を開発時に完全に予測できるのであろうか」、というものである。予測できない状況が発生すれば、開発時で実施する信頼性保証の技術では十分でない。

著名なソフトウェア専門家である M. Jackson は、"Any useful computing system interacts with the world outside the computer."と述べている。計算システムが行う外部とのやりとりの中には、利用者の入力情報を含む。以前の大型計算機システムであればオペレータ教育を受けた専任者がシステム利用者であった。一方、今後ますます重要となる大容量オープン・ネットワークを前提としたソフトウェア・システムでは、利用者像を特定することができない。一般利用者は、思い違い、不注意、悪戯、など、気まぐれな振る舞いを示すかもしれない。さらに、悪意を持って「攻撃」することさえある。このような開

発時に予測できなかった利用者の振る舞いに対して、システムがどのように振る舞うかも予測できない。予測できないという意味において、システムの信頼性は低下する。

一般に、システムは初期開発終了後も継続的に保守、機能拡張される。新たな機能の追加や当初想定していたシステム要件の変化などに起因する。ソフトウェア・システムの本質は変化に対する発展である。現状の技術では、機能拡張などの開発作業を伴う。その改修の結果、安定作動していたソフトウェアに新たに不具合が混入するなどの問題が発生している。機能拡張を含む改修作業がシステムの信頼性を低下させることになる。この問題に系統的な解決策を与える技術として「ソフトウェア発展（Software Evolution）」の研究分野がある。1998 年片山らが創設した国際ワークショップ IWPSE（International Workshop on Principle of Software Evolution）によって研究分野として明確な位置を得た。現在、ERCIM の EVOLVE などに引き継がれて、ソフトウェア・プロダクトライン工学（Software Product Line Engineering）の技術との関連性が明らかにされてきた。

利用者の振る舞い、および、システム要件の変化は、見方を変えると、サービスイン後のシステム信頼性に関わる問題と理解することができる。図 15は、開発時の信頼性を向上させる技術がカバーする範囲を示し、実行時になってはじめてわかる問題への対応が新たに必要であることを示している。

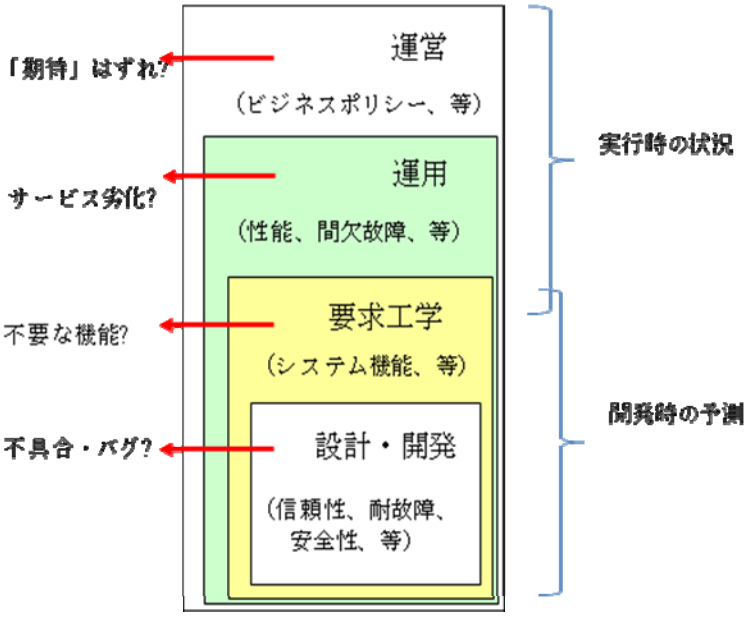


図 15　開発時の予測と実行時に発生する問題

ソフトウェア発展に関する研究の流れの中で、これらを統一的に取り扱う概念的な仕組みである「自己適応型システム（self-adaptive systems）」が提案された。研究者によって

関心が少し異なり、self-sustainable、self-evolving 等、いろいろな呼び方をされている。最近では、総称して、「self-* systems 」と呼ぶので、本稿でも、これに従う。似た概念は、過去に、人工知能などの隣接技術分野で議論されてきた。エンタープライズシステムではディペンダビリティ向上という明確な目的を持つ技術として理解されている。実用的な観点からも重要であり、2000 年頃には、IBM が新しいシステム化の考え方として、Autonomic Computing を提唱した。システム運用管理の自動化から生まれた概念であるが、新しい見方をソフトウェア・システムに導入するものである。

Self-* systems に関するソフトウェア生産技術からの研究は、歴史を遡ると、1995 年に S. Fickas（米国オレゴン大学）たちが提案した「要求モニタリング」に端を発する。ソフトウェア・システムがサービスイン後、何らかの方法で自身の要件定義に変化が発生するかをモニターする。次に、その変化を吸収、対応するような機能変更を自身に引き起こす。予測できなかった利用者の振る舞いを要件変化と捉えれば、先に述べた２つの問題を同じ枠組みの特殊な場合として論じることも可能である。

容易に想像できるように、本格的な self-* systems の実現には解決すべき技術課題が多い。Self-* systems は具体的な技術を示すキーワードではなく、autonomic computing と同様に、将来のシステム像を描いた研究分野を指し示すと考えるべきであろう。先に述べた要求モニタリングの考え方が提案されて 10 年ほどが経過した 2005 年頃から、欧米の大学からコンセプトデモといえる研究成果が公開されはじめた。対象を限定した self-* systems の構成要素技術に関する具体的な研究が進められている。現在、いくつかの国際ワークショップが並立している段階であり、大規模な国際会議の中心テーマのひとつになる日も近い。北米では先に述べたオレゴン大学の他、ミシガン州立大学、CMU、UC アーバインなど、英国ではインペリアルカレッジ、ニューキャッスル大学など、また、ERCIM の SERENE や FP7 の NoE である S-Cube などの活動がみられる。主として、要求工学、ソフトウェア・アーキテクチャ、耐故障システム等を技術背景として self-* systems の諸要素にチャレンジしている。さらに、FP7 では FET-Proactive 領域カテゴリのひとつに設定し、2009 年の公募でも課題に入っている。将来のシステムでの必須技術として研究開発投資が活発に行われている。

最後に、図 15を再び参照する。第１．４節で紹介した技術は、図 15の「設計・開発」すなわち、「開発時の予測」の正しさ、あるいは、予測に従って構築したシステムの正しさを確認することで信頼性向上を達成する基本技術である。一方、第１．５節で述べた技術は、図 15の「運用」や「運営」など「実行時の状況」という観点から信頼性向上を達成するための新しいアプローチである。これに対しても、第１．４節で論じた技術を応用することも考えられ、実際、そのような研究が進んでいる。

産業競争力懇談会（ＣＯＣＮ）

東京都千代田区丸の内一丁目６番６号　〒100-8280
日本生命丸の内ビル（株式会社日立製作所内）
Tel：03-4564-2382　　Fax：03-4564-2159
E-mail：cocn.office.aj@hitachi.com
URL：http://www.cocn.jp/
事務局長　中塚隆雄