3. フォーマルメソッド導入の意義と効果の概要

本章では、ソフトウェア開発における従来のソフトウェア・テストとフォーマルメソッドを比較して、フォーマルメソッド導入の効果と意義について概要を示す。

対象読者	(1) 発注者（CIO, CTO 等） (2) ベンダー上級管理者等
目的	フォーマルメソッドの効果と導入意義について、フォーマルメソッドの前提知識無しに分かりやすく説明する。
想定知識	ソフトウェア開発の概要
得られる知見等	● 現状のソフトウェア開発における問題点 ● 現状問題点に対するフォーマルメソッド導入の効果と意義の概要

開発現場にフォーマルメソッドの導入を図るためには、フォーマルメソッドの効果について、それを直接利用するベンダーから理解されるだけでなく、システム開発の発注者からも理解が得られることが期待される。システム納入後にソフトウェアの不具合に起因する事故が発生した場合、その損害は、発注者であるシステム利用者にも影響をおよぼし、企業価値の毀損などにつながるため、フォーマルメソッドの適用検討は、生産性向上のみならず、事故リスクの低減などシステム利用者による長期的な視点でコストと効果を判断することが必要なためである。

効果とコストに関する詳しい説明は第 5 章にまとめるが、本章では、その中でも重要な点を技術的な知識を用いず説明する。

フォーマルメソッドの効果と意義を示すために、従来のソフトウェア開発の問題点を挙げる。関連する問題点は以下のような事が挙げられる。

- ソフトウェア・テストに関する問題点

  従来のソフトウェア・テストでは、どんなにテスト件数を増やしても、不具合が無いことを保証することが難しい。特に、処理タイミングに関わる再現性の無い動作については、テスト件数を増やすだけでは不具合が存在しないことを保証できない。

  人命に関わるセーフティクリティカル・システムの制御、電力等の重要インフラ、決済、基幹システム、セキュリティ、膨大な普及数になる家電などの場合、ソフトウェアの不具合に起因する障害は極めて大きい損害をもたらす場合がある。このような場合に、明確に定義した性質が満たされることを保証することは大きな意味がある。

- 開発プロセスに関する問題点

  ソフトウェア・テストは、設計、コーディングを経てプログラムが実行できるようになった段階で初めて行うことができる。そのためソフトウェア・テストで発見された不具合を修正するためには、開発の上流工程の要求仕様や設計仕様に遡って作業をやり直すことが必要になるこ

ともありトータルコストが増大するリスクがある。これにより納期遅延などの深刻な問題につながる。

以上のような問題に対してフォーマルメソッドは、以下のような効果が期待できる。

- ソフトウェア・テストの弱点の補強

  フォーマルメソッドによりソフトウェアの設計仕様と検証性質を明確に記述し、ツールを用いて検証が成功すれば、設計仕様が検証性質を満たすことを保証することができる。セーフティクリティカル・システムなど人命に関わるシステムにおいては、ソフトウェアが特定の性質を満たすことを保証することは極めて重要な意味を持つ。

  一方で、従来のソフトウェア・テストでは、不具合の存在が発見されても、その原因を特定することが困難な場合がある。フォーマルメソッドの一部の手法では、不具合に至るソフトウェアの実行履歴を再現することで、不具合の原因の特定に有用な情報を得ることができる。

  フォーマルメソッドは、ソフトウェアの信頼性や安全性等について論理的な検証により、客観的・科学的な保証を与えるためのアプローチとして利用することも有用である。

- 開発プロセスの改善

  フォーマルメソッドの利用により、プログラムを作成する前に、要求仕様や設計仕様を記述した段階で、設計仕様が要求仕様を満たすことの検証や、矛盾の発見などの解析を行うことができる。これにより開発の上流工程で早期に不具合を発見することができるようになり、開発の手戻りによる工数増大と納期遅延リスクを抑えることが可能になる。

以上のようなフォーマルメソッドの効果のうち、定量的な評価が難しいものもあるが、大まかな規模を捉える上で以下のように考えることができる。

- 事故損害リスクの低減

  フォーマルメソッドの特徴的な効果である検証性質が満たされることを保証したり、不具合の原因を特定することによるソフトウェアの品質向上は、最終的にはソフトウェアの不具合に起因するIT事故による損害リスクの低減として具現化される。

  IT事故の損害リスクは、ソフトウェアを利用する事業者やベンダーが十分認識していると思われがちであるが、現実には、IT事故によってもたらされる事業者やベンダーに対する信用失墜を考慮すると、一般的な認識に比べて想像以上に損害リスクの規模が大きいことが分かってきている。具体的には、情報セキュリティ経済学などの進展により、IT事故における復旧コスト、業務中断による営業機会損失、リコール、損害賠償等の直接的コスト以上に、信用失墜により将来に渡るビジネス機会の損失などの潜在的な損害が大きいことが示されている。詳しくは 5 章に示すが、具体的には、直接コストに対して信用失墜による損失を含む企業価値全体のコストは3倍程度になる場合がある結果が示されている[16]。

- 開発プロセスの改善

[16] 第 5 章参照。

ソフトウェア開発におけるテスト工程のコストはバラつきはあるが 5 割程度と言われている[17],[18]。また、NIST報告書 では、開発コストの 80%は、欠陥を特定して修正することに費やされると見積もっている。フォーマルメソッドの適用により開発の上流工程に重点を置くフロントローディング[19]により、開発工数の大幅な削減が達成される実例が示されている。分野による違いは大きいが、航空分野で適用されている実例では、フォーマルメソッドに基づく言語を用いて設計し、自動コード生成を活用することにより、コーディング、レビュー、テスト工程の 7～9 割の工数が削減されている。一方で、鉄道分野においては、設計から実装に至るまでの工程を、フォーマルメソッドを用いた検証を徹底することにより、自動生成されたコードの単体テストを全く行わない事例などもある[20]。

以上に示したようなフォーマルメソッド導入効果に関しては、適用対象や適用の仕方に応じて、メリットの規模に違いがみられるため、一概に大きなメリットが得られるとはいえないが、実践的な応用事例として、フォーマルメソッドの導入判断の参考とできる重要な情報である。より具体的な情報と考え方については、5 章を参照すると良い。

[17] ROBERT M. HIERONS, et.al., Using Formal Specifications to Support Testing, ACM Computing Surveys, Vol. 41, No. 2, pp. 9-76, 2009

[18] W.S. Humphery, "Winning with Software. An Executive Strategy", Addison-Wesley, 2001

[19] 開発の上流工程の要求分析や設計に重点を置いて集中的に負荷・資源を投入することにより、下流工程で発生する負荷を減らし、トータルの工数を下げる活動を指す。システム開発や製品製造で用いられてきた方法であるが、ソフトウェアの分野でも取り込もうとする動きがある。

[20] Meteor, A Successful Application of B in a Large Project, FM'99, Vol. I, LNCS 1708, pp. 369-387, 1999.