

情報システム開発と教育についての一考察

A Study of Information System Development and Education

高林 茂樹

TAKABAYASHI Shigeki

The information systems have recently become more complex. For many modifications during system development, the cost is high and the period is long compared with the initial estimate. Therefore, we want to have a method of system development with the easy functions of modification. There is the agile system development method as an example. As we modify the system quickly using this method, all members have to understand the whole system and technical knowledge, and communicate with each other for the adjustments.

1. はじめに

コンピュータが単なる計算をする道具から、情報システムの中心となって情報処理をするようになってから、およそ半世紀が経過した。そしてシステム開発もプログラムの作成だけでなく、人とコンピュータの役割を分担し、情報の流れを整理し加工・提供して、システム全体の効率化を図るようになった。そこで、システム開発の方法として、ウォーターフォール・モデルやプロトタイプ・モデルなどが提案されて、大規模なシステム開発で用いられてきた。しかし、システム開発の途中での変更が多く、当初の見積りを金銭的にも期間的にもオーバーすることが多くなってきた。また、システムが複雑になるにつれ、システムの発注者と受託者の間で、システムの利用者側の要求が、システムの開発側に十分に理解されないまま設計・開発にはいつてしまう事態も発生している。たとえば、日本経済新聞によれば、「S銀行が、開発を依頼していたI社が債務を履行しなかったとして、損害賠償を求める裁判を東京地方裁判所に起こし、約74億円の

賠償をI社に命じる判決が2012年3月に出た。」ことが報道されている。システムの複雑化とともに、あらゆる場面を想定して、完璧なものを作ることがむずかしくなっている。開発の途中で、性能がよく価格も安いハードウェアやソフトウェアが現れることもある。あるいは、社会情勢の変化、人々の意識の変化、法律や規則の変更などにより、当初予測のできなかった開発環境の変化が起こることがある。コミュニケーションが悪くて、開発の変更をせざるを得ないこともある。ソフトウェアでは、購入後にアップデートによって不具合の修正を行うことも多くなっている。もちろんこの中には、性能アップのための修正もあるが、コンピュータウイルスの攻撃に対する備えが不十分でそれを補うものも多い。

このように複雑化した情報システムの開発では、途中での変更に対処しやすいものが求められる。また、利用者と開発者間のコミュニケーション、利用者間、開発者間のコミュニケーションも重要である。情報機器の普及そしてネットワークの広がりや高速化により、今後、ますますシステム開発の需要が増え、システム開発者の育成が必要となる。情報処理学会や日本技術者教育認定機構では、大学教育の質保障という観点から、国際的相互承認に耐えるレベルの教育が検討され一部実施されている。^[1]しかし、これだけでは多様で進歩の速いシステム開発の人材育成のために十分とは思えない。ウォーターフォール・モデルでのシステム開発は、仕事を細分化して与えられた範囲の作業をする能力があればよかったが、変更に対処しやすいと言われるアジャイル開発法によるシステム開発では、1人1人が全体を理解できる幅広い知識と、コミュニケーション能力や調整能力などが必要とされる。この論文では、変化に対応しやすいシステム開発の方法とそれに従事するシステム開発者の教育について考察した。

次に、情報システム、システム開発に必要な知識や技術、システム開発方法について、その推移および現状について述べる。

2. 情報システム開発方法の現状

2.1 情報システムの推移と現状

1950年代は、EDP (Electronic Data Processing) システムと言われたバッチ処理システムで企業の経理処理や給与計算などが行われた。1950年代の後半にはそれまで個別に行われていたコンピュータ処理をオンラインで統合処理する時代となった。1960年代にはJR (当時の国鉄) の座席予約や銀行の業務が、大規模なオンラインリアルタイムシステムとして稼働した。経営の

分野では、経営情報システム（MIS:Management Information System）が構築され、大量のデータを加工して経営に関する情報を提供することができるようになり、それにはオンラインやデータベース技術が用いられた。さらに1970年代になると経営の意思決定支援を行う意思決定支援システム（DSS:Decision Support System）が構築された。1980年代後半から、情報システムを利用して、企業の経営戦略を支援するために、戦略情報システム（SIS:Strategic Information System）が構築された。1990年代になると、サーバーとLANで結合された多数のパソコンからなるクライアントサーバーシステムが主流となってきた。さらに組織外や海外にある多数のコンピュータがネットワークによって組織内のコンピュータと接続できるようになった。インターネットの利用も始まり、家庭でもパソコンが入り始め、電子メールやWebは、組織内あるいは組織間の情報交換を変えたばかりではなく、企業と消費者の間で新たなコミュニケーションが可能になり、ネットショッピングやネットオークションなど新しいビジネスモデルが創造された。さらに、90年代後半の携帯電話の普及とインターネットとの接続で、コミュニケーション・スタイルに新たな変化がもたらされた。このような情報をめぐる環境の激しい変化により、システム開発もそれまで以上にスピードが要求されるようになり、システムの新規開発や変更も多くなった。

2.2 情報システム開発の知識や技術の推移と現状

情報システム開発のためには、さまざまな知識や技術が必要とされ、時代によってそのニーズも変化してきた。情報システム開発の知識や技術について、それを認定する情報処理技術者試験を例に見ていくことにする。^[2]

1969年にプログラムの設計、高度のプログラムの作成のできる人を対象にした第一種情報処理技術者認定試験とプログラム設計書に基づくプログラムの作成を対象にした第二種情報処理技術者認定試験が実施された。1種、2種はいずれもプログラマを対象とした試験であり、情報処理技術者のなかで重要な役割を担うべき、情報処理システムの分析、設計に従事するシステムエンジニアを対象としたものは、1971年から実施された特種情報処理技術者試験である。情報処理システムの監査に主として従事する者を対象とした情報処理システム監査技術者試験は1986年から実施された。システム監査は、監査対象から独立した監査人が、情報システムを信頼性、安全性及び効率性の観点から総合的に点検・評価し、関係者に助言・勧告するものであり、コンピュータ・セキュリティの確保とシステムの有効活用を図る上で極めて有効な手段であり、情報化社会の健全化に大きく貢献するものである。情報処理システムが急速にネットワーク化する中

で、オンラインシステムの設計・構築・エキスパートシステムの構築等、新たに高度な専門的知識を持つ技術者が必要となってきたため、1988年からネットワークを利用する情報処理システムの分析、設計に主として従事する者を対象としたオンライン情報処理技術者試験が実施された。プログラミングの知識・技術から情報処理システムの分析、設計、監査そしてネットワークの知識・技術へと要求が増加してきた。

1994年からは、試験が11区分になり、経営戦略に立脚した情報システム化計画の立案に主として従事する者を対象とするシステムアナリスト試験、情報システムの監査に主として従事する者を対象とするシステム監査技術者試験、情報システム開発のプロジェクト管理に主として従事する者を対象とするプロジェクトマネージャ試験、個別アプリケーションのシステム化計画、情報システムの分析、設計に主として従事する者を対象とするアプリケーションエンジニア試験、情報システムの運用管理に主として従事する者を対象とするシステム運用管理エンジニア試験、情報システムの開発に主として従事する者を対象とするプロダクションエンジニア試験、ネットワークシステムの構築、維持、技術支援に主として従事する者を対象とするネットワークスペシャリスト試験、データ資源の管理及びデータベースシステムの構築、維持、技術支援に主として従事する者を対象とするデータベーススペシャリスト試験、高度情報処理技術者を目指し、情報システムの開発、保守、運用のいずれか又は複数の業務に従事する者を対象とする第一種情報処理技術者試験、高度情報処理技術者を目指し、情報システムの開発、保守、運用のいずれか又は複数の業務に従事する者を対象とする第二種情報処理技術者試験、情報システムの利用者の立場で、エンドユーザコンピューティングの推進に従事する者を対象とする初級システムアドミニストレータ試験が実施された。1996年からは、マイコン応用システムエンジニア試験と上級システムアドミニストレータ試験も実施されるようになった。この時期、経営戦略に立脚した情報システム開発能力や情報システム開発のプロジェクト管理能力が要求されるようになった。

2001年からは、経営戦略に基づく情報戦略の立案、システム化全体計画及び個別システム化計画の策定を行うとともに、計画立案者の立場から情報システム開発プロジェクトを支援し、その結果を評価する者を対象としたシステムアナリスト試験、情報システム開発プロジェクトの責任者として、プロジェクト計画の作成、要員などプロジェクト遂行に必要な資源の調達、プロジェクト体制の確立及び予算・納期・品質などの管理を行い、プロジェクトを円滑に運営する者を対象としたプロジェクトマネージャ試験、情報システム開発プロジェクトにおいて、プロジェクト計画に基づいて、業務要件分析からシステム設計、プログラム開発、テストまでの一連のプロセスを担当する者を対象としたアプリケーションエンジニア試験、情報システム開発プロジェク

トにおいて、内部設計書・プログラム設計書を作成し、効果的なプログラムの開発を行い、単体テスト・結合テストまでの一連のプロセスを担当する者を対象としたソフトウェア開発技術者試験、情報システム基盤（業務システム共有のシステム資源）の構築・運用において中心的役割を果たすとともに、個別の情報システム開発プロジェクトにおいて、固有技術の専門家として開発・導入を支援する者を対象としたもので、ネットワーク分野のテクニカルエンジニア（ネットワーク）試験、データベース分野のテクニカルエンジニア（データベース）試験、システム管理の分野のテクニカルエンジニア（システム管理）試験、マイクロプロセッサやシステム LSI などの分野のテクニカルエンジニア（エンベデッドシステム）試験、情報セキュリティに関する基本的な知識をもち、情報セキュリティ管理の現場責任者として、情報セキュリティを保つための施策を計画・実施し、その結果の評価を行う者を対象とした情報セキュリティアドミニストレータ試験、そして以前からの上級システムアドミニストレータ試験、初級システムアドミニストレータ試験、システム監査技術者試験が実施された。2006年に情報セキュリティ分野のテクニカルエンジニア（情報セキュリティ）試験が加わって、情報セキュリティに関する試験が2つとなり、安全性が一段と重視されるようになった。

2009年からは、高度な IT 人材の養成のため試験区分の統合や内容の改訂があり、職業人が共通に備えておくべき情報技術に関する基礎的な知識をもち、情報技術に携わる業務に就くか、担当業務に対して情報技術を活用していこうとする者を対象とした IT パスポート試験、高度 IT 人材となるために必要な基本的知識・技能をもち、実践的な活用能力を身に付けた者を対象とした基本情報技術者試験、高度 IT 人材となるために必要な応用的知識・技能をもち、高度 IT 人材としての方向性を確立した者を対象とした応用情報技術者試験、高度 IT 人材として確立した専門分野をもち、企業の経営戦略に基づいて、ビジネスモデルや企業活動における特定のプロセスについて、情報技術を活用して改革・高度化・最適化するための基本戦略を策定・提案・推進する者、また、組込みシステムの企画及び開発を統括し、新たな価値を実現するための基本戦略を策定・提案・推進する者を対象とした IT ストラテジスト試験、高度 IT 人材として確立した専門分野をもち、IT ストラテジストによる提案を受けて、情報システム又は組込みシステムの開発に必要な要件を定義し、それを実現するためのアーキテクチャを設計し、情報システムについては開発を主導する者を対象としたシステムアーキテクト試験、高度 IT 人材として確立した専門分野をもち、システム開発プロジェクトの責任者として、プロジェクト計画を立案し、必要となる要員や資源を確保し、計画した予算、納期、品質の達成について責任をもってプロジェクトを管理・運営する者を対象としたプロジェクトマネージャ試験、それぞれの固有技術を活

用し、最適な情報システム基盤の企画・要件定義・開発・運用・保守において中心的な役割を果たすとともに、固有技術の専門家として、情報システムの企画・要件定義・開発・運用・保守への技術支援を行う者を対象にしたネットワークスペシャリスト試験、データベーススペシャリスト試験、エンベデッドシステムスペシャリスト試験、情報セキュリティスペシャリスト試験、高度IT人材として確立した専門分野をもち、情報システム全体について、安定稼働を確保し、障害発生時には被害の最小化を図るとともに、継続的な改善、品質管理など、安全性と信頼性の高いサービスの提供を行う者を対象としたITサービスマネージャ試験、そして以前からあるシステム監査技術者試験が実施されている。

企業の経営戦略に基づいて、ビジネスモデルや企業活動における特定のプロセスについて、情報技術を活用して基本戦略を策定・提案・推進する人もこの試験の対象とするようになった。なお、多様化する脅威や情報セキュリティ対策の追加、クラウドコンピューティングの急速な普及、Webアプリケーション開発の拡大と関連技術の進歩などの観点から、2012年5月にVer.5として、出題範囲等の改訂も行われている。^[2]

2.3 情報システム開発方法の推移と現状

プログラムの開発方法では、対象となるものをどのように人間が把握し、設計し、プログラムとして実現するかという視点から構造化技法（構造化プログラミング）とオブジェクト指向開発法（オブジェクト指向プログラミング）がある。構造化技法は、個々の処理を小さな単位に分解し、階層的な構造にしてプログラムを記述する手法である。オブジェクト指向開発法は、ソフトウェアを相互作用するオブジェクトの集まりとしてモデル化するものである。

システムの開発方法では、開発プロセスから見ると次のものがある。

(1)ウォーターフォール・モデル

一般に、開発プロジェクトを時系列順に、「要求定義」「外部設計（概要設計）」「内部設計（詳細設計）」「プログラミング」「テスト」「運用」などの作業工程に分割し、原則として前工程が完了しないと後工程に進まないことで、前工程の成果物の品質を確保し、前工程への後戻りを最小限にする方法である。最初にしっかり設計をして、変更が無いようにしないと後戻りが発生する。

(2)プロトタイピング・モデル

将来完成する予定に近いモデル（プロトタイプ）を作成することで、顧客がそのシステムを評価することができ、また、設計者はプロジェクトの初期段階でユーザーからフィードバックを得ることができる。しかし、どの程度の完成度のプロトタイプを作成するかで費用や期間が変わる。

(3)反復型開発・モデル

反復型開発・モデルには各種あるが、スパイラル・モデルは、トップダウン設計とボトムアップ設計の長所を生かしたソフトウェア開発工程のモデルで、設計とプロトタイピングを繰り返して開発していく手法である。スパイラル・モデルは大規模プロジェクトでよく使われる。小さいプロジェクトでは、アジャイル開発の方が良く使われるようになってきた。

アジャイルソフトウェア開発宣言に合意している代表的なアジャイル開発手法には次のようなものがある。^{[3] [7]}

- ・エクストリーム・プログラミング (Extreme Programming, XP)
- ・スクラム (Scrum)
- ・クリスタルファミリー (Crystal Family)
- ・フィーチャー駆動型開発 (Feature-Driven Development, FDD)
- ・適応的ソフトウェア開発 (Adaptive Software Development, ASD)

アジャイル開発は、1週間から1カ月程度の短い期間を単位として、それを反復しながら段階的にソフトウェアを開発する手法であるが、オブジェクト指向で使われる再利用部品群、デザインパターンなどの技術は、高品質で保守や再利用がしやすいソフトウェアを手早く作るための開発手法であるから、アジャイル開発は、オブジェクト指向の開発法となっている。

3. 変化に対応しやすいシステム開発の方法と教育

3.1 変化に対応できるプログラミング

プログラム作成では、ケント・ベックらによって提唱されているソフトウェア開発方法であるエクストリーム・プログラミングが、途中での変更にも対応が柔軟にできる方法として使われている。エクストリーム・プログラミングで示されている4つの価値とは「コミュニケーション」、「シンプル」、「フィードバック」、「勇気」の4つである。これら4つの関係を式で表すと「コミュニケーション+シンプル+フィードバック=勇気」となる。^[4]これは、コミュニケーションができ、プログラムの設計がシンプルで、さらに、変更を行った結果が即座にフィードバックされる仕組みがあってはじめて勇気を持って乗り越えることができることを示している。そして、エクストリーム・プログラミングは経験に基づいて有用性が立証された実践項目を「12のプラクティス」として次のように示している。^[4]

- ・計画ゲーム (The Planning Game) ビジネス優先度と技術的見積により次回リリースの範囲を早急に決める。
 - ・小さなリリース (Small Releases) 新バージョンを非常に短いサイクルでリリースする。
 - ・メタファー (Metaphor) システムの動きを示すメタファーを共有する。
 - ・シンプルデザイン (Simple Design) シンプルに設計にする。
 - ・テストイング (Testing) プログラマは継続的にユニットテストを書き、顧客は機能テストを書く。
 - ・リファクタリング (Refactoring) システムの動作を変えることなくシステムを再設計する。
 - ・ペアプログラミング (Pair Programming) コードは2人のプログラマにより一台のマシンで行う。
 - ・共同所有権 (Collective Ownership) 誰でもどのコードでも修正できる。
 - ・継続的インテグレーション (Continuous Integration) 一日に何回もビルドし、テストを100%パスさせる。
 - ・週40時間 (40-Hour Week) 週40時間以上の仕事はよい結果を生まない。
 - ・オンサイト顧客 (On-Site Customer) 顧客をフルタイムでチームに加える。
 - ・コーディング標準 (Coding Standards) 全プログラマがコーディング標準に従う。
- これらは、次に述べるようにアジャイル開発法に影響を与えている。

3.2 変化に対応できるシステム開発

システム開発において、途中での変更に対応が柔軟にできる方法として、アジャイル開発の方法が使われている。

2001年に、ケント・バックやジェイムス・グレニングなどのアジャイル開発の提唱者17名が集結し「アジャイル宣言 (Agile Manifesto)」が発表された。それには、4つの価値と12の原則が、次のように示されている。^{[5] [6] [7]}

4つの価値

私たちは、ソフトウェア開発の実践を手助けする活動を通じて、より良い開発手法を見つけ出そうとしている。この活動を通して私たちは以下の価値に至った。

- ①プロセスやツールよりも個人との対話を、
- ②包括的なドキュメントよりも動くソフトウェアを、
- ③契約交渉よりも顧客との協調を、

④計画に従うことよりも変化への対応を
価値とする。

12の原則

私たちは以下の原則に従う。

- ①顧客満足を最優先し、価値のあるソフトウェアを早く継続的に提供する。
- ②要求の変化は、たとえ開発の後期であっても歓迎する。変化を味方につけることによって、顧客の競争力を引き上げる。
- ③動くソフトウェアを、2～3週間から2～3カ月というできるだけ短い期間でリリースする。
- ④ビジネス側の人と開発者は、プロジェクトを通して日々一緒に働く。
- ⑤意欲に満ちた人々を集めてプロジェクトを構成する。
- ⑥情報を伝える最も効率的で効果的な方法は、フェイス・トゥ・フェイスで話をするところである。
- ⑦動くソフトウェアこそが進捗の最も重要な尺度である。
- ⑧アジャイル・プロセスは持続可能な開発を促進する。一定のペースを継続的に維持できるようにしなければならない。
- ⑨技術的卓越性とすぐれた設計に対する不断の注意が機敏性を高める。
- ⑩シンプルさ（ムダなく作れる量を最大限にすること）が本質である。
- ⑪最良のアーキテクチャ・要求・設計は、自己組織的なチームから生み出される。
- ⑫チームがもっと効率を高めることができるかを定期的に振り返り、それに基づいて自分たちのやり方を最適に調整する。

ここでは、「個人との対話」、「動くソフトウェア」、「顧客との協調」、「変化への対応」が強調され、形式よりも実践を重視している。

アジャイル開発の効果としては、次のことがあげられている。^[3]

短期間にフィードバックを繰り返すことにより、ユーザーの要望を的確に把握し満足度の高いシステムを開発できる。無駄な文書化よりも動くソフトウェアを早期にリリースすることを優先し、実際に動くソフトウェアを利用して要求の確認を行いながら開発を進めることで要求の把握が容易になる。

優先度の高い機能から徐々に開発していくことで、仕様の欠陥を早期に発見できるほか、作りすぎや要件の盛り込みすぎといった無駄が省けるため、開発コストが削減できる。各イテレーション（反復）のタイミングで実装する機能の計画を見直し、その評価を行うことができる。

ビジネス環境は絶えず変化しているが、先が予測できないような状況であっても、変化に柔軟で短納期で修正が可能なシステムであれば、環境変化に適合できるシステムライフサイクルを実現できる。

アジャイル開発導入時の注意点としては、見積もった工数が膨らんだ場合の負担に関する調整など、それまでの契約形態とは異なる方策が必要になることがあるといわれる。システム開発を開始する前に、システムの導入の目的であるビジネスゴールを検討した上で、プロジェクト全体像を捉えることが重要である。要求や作業の優先度をしっかり評価して作業順序を決定することも重要である。また、開発の効果を高めるには、1人で多くの役割が可能な開発者の比率を高めることが重要となる。

3.3 システム開発と教育

情報システムの開発での教育において基本となるのは次の3つである。

- ①学校教育でのカリキュラム
- ②認定試験等での個人のレベル把握
- ③実際の開発現場での OJT

これらは独立したものではなく、連携しながら人材を育成していく必要がある。システム開発では、規模が大きくなり、変化に対応できる反復型の開発の実践的な教育は、期間的に十分なことをするには困難な面もあるが、プログラミングでは、規模も小さくできるので、教育に取り入れることが比較的容易である。情報処理の基礎知識を修得後、システム開発の方法の説明とその実践の1つとして変更を反復的に経験しながらプログラミングをすることで、変更に対応する方法を身につけることができると考える。また、しっかりしたシステムを設計するためには、ウォーターフォール・モデルについての教育も基礎として大切である。技術的な知識とともにコミュニケーション能力も重要である。情報処理学会の「情報専門教育カリキュラム標準 (07)」^[8] 中にも「対人関係の能力」として「コミュニケーション能力」、「インタビュー、質問、傾聴」、「プレゼンテーションの技能」、「執筆能力」などが書かれている。カリキュラムでは、日本技術者教育認定機構 (JABEE) のような機関が、国際的なレベルで、情報専門の技術者教育プログラムを認定することも必要である。個人の能力を認定する情報処理技術者試験などのような試験も必要である。また、学会等での発表や議論も必要である。そして実際のシステム開発では、様々なケースがあり、OJT が重要である。

4. おわりに

人々の活動のグローバル化とボーダーレス化、組織の大規模化と効率化、個人のライフスタイルの多様化などによって、情報システムは大きく変化した。特に、インターネットの登場は、情報システムの利用者の範囲を広げ、情報システムの多様化、複雑化をもたらした。その結果、情報システムの設計、開発、運用、保守、改良には多大なコストと時間を要するようになった。また、システム障害や不正侵入などが発生した場合の影響も大きく、セキュリティ対策への要求がさらに高まった。このような情報システムを取り巻く環境の変化に対して、情報システムの開発でもより速い対応が求められている。変化に対応しやすいと言われるアジャイル開発法なども、今後、さらに改良されることを期待したい。そのために事例を数多く収集し、分析することである。

情報システムをどのように設計・開発し、運用していくかは、多くの組織や人々の生活を左右するものである。また、情報技術の進歩およびそれに伴う社会の変化は、今後想像を越えるものとなるであろうが、それを受け入れるだけの柔軟さと包容力は備えている必要がある。そして変化・進化に対応できるシステム開発力が社会を変えていくことにもなる。変化の激しい時代には、システム開発に参加するすべてのメンバーが、変化に迅速に対応することができるようになることが重要である。そのためには、

- ①専門的な知識・技術
- ②システム全体を把握できる能力
- ③提案・調整のできるコミュニケーション能力

を備えた人材の育成をすることが必要である。

参考文献

- [1] 情報処理学会「大学教育の質保証」情報処理 Vol.53 No. July 2012
- [2] 情報処理推進機構「情報処理技術者試験」2012 <http://www.ipa.go.jp/>
- [3] 柏木雅之、山下博之「アジャイル開発を実案件に生かすための基礎知識」ZDNet 2011 http://japan.zdnet.com/development/sp_agile2011/35003090/

- [4] 鍋健児 「eXtreme Programming の魅力を探る」 オブラブ 2001
http://objectclub.jp/community/XP-jp/xp_relate/xp-intro
- [5] 平一橋範哉 「ソフトウェアの新たな開発手法、アジャイル開発って」 ZDNet 2006
http://japan.zdnet.com/sp/sp_06sp0130/20248727/
- [6] Jonathan Rasmusson 近藤修平 他訳 「アジャイルサムライ」 オーム社 2011
- [7] Dean Leffingwell 玉川憲 他訳 「アジャイル開発の本質とスケールアップ」 翔泳社 2010
- [8] 情報処理学会 「情報専門学科におけるカリキュラム標準「J07」最終報告」 情報処理学会第70回全国大会シンポジウム 2008