

システム開発の足腰を鍛えるソフトウェアエンジニアリング教育

Software Engineering Education building up Physique of System Development

宮 脇 亨

要 約 2004年頃から、ソフトウェアエンジニアリング教育を見直す動きが盛んになっている。現在、情報システムは、変化の激しいビジネス環境に対応した新たなビジネスモデルの創出と業務効率化の両面を実現する役割を担うだけでなく、社会生活の身近なところに普及し、便利、安心を提供するインフラとして位置づけられている。ひとたび情報システムに障害が発生すると、ビジネスの停止だけでなく、社会生活へ多大な影響を及ぼしかねない状況にある。情報システムを開発する立場としては、複雑で変化するビジネス要求に応えながら、社会インフラとしての責務を意識したシステム作りが求められている。

本稿では、主に情報システムの開発や運用・保守を提供する立場から、現在の情報システムを支える人材をどのように育成していくべきか、といった視点で、その中心として位置づけるソフトウェアエンジニアリング教育について述べる。

Abstract Movement to review the software engineering education has been activated for several years. The information system not only plays the role to achieve both sides of the creation of a new business model responding to the business environment with an extreme change and the workforce optimization, but also spreads to the immediate surroundings of the social life, and is positioned as an infrastructure offering the convenience and safety. Once any failure occurs in the information system, it comes down to not only the discontinuance of business, but also a situation in which a considerable influence on the social life might be exerted. There is a call for the system building conscious of the obligation as the society's infrastructure while satisfying the complex and changing business needs, from the standpoint to develop the information system.

This paper discusses the software engineering education positioning as the center in the aspect how it is necessary to develop human resources who support a present information system from the standpoint to offer mainly the development, operation and maintenance of the information system.

1. はじめに

インターネットの普及、ネットワークの高速大容量化、センサー技術の進化など、今やITは社会生活の身近なところまで普及し、便利、安心を提供するインフラとして位置づけられている。企業システムとともに発展してきた情報システムは、変化の激しいビジネス環境に対応した新たなビジネスモデルの創出と業務効率化の両面を実現する役割を担っているだけでなく、社会的責任を問われている。情報システムがひとたび障害が発生すると、ビジネスの停止だけでなく、社会生活へ多大な影響を及ぼしかねない状況にある。

情報システム開発を担うITベンダや企業の情報システム部門は、これまでも銀行システムやチケット予約システムに代表される、いわゆるミッションクリティカルなシステムづくりを

おこなってきたが、最近では、ビジネス環境の変化に晒された経営からの、複雑なサービスを短時間で提供したいという要求に応えながら、社会インフラとしての責務を意識した、安心、安全なシステム作りのプレッシャーと戦う必要が出てきている。

ソフトウェアエンジニアリングは、ソフトウェアの開発、運用、保守に対して、系統的で規律をもった定量的なアプローチの適用である¹⁾。複雑な要求への対応と煩雑な開発作業に追われている開発者にとって、系統的で規律をもった定量的なアプローチを適用することは有効である。単にソフトウェアエンジニアリングツールの適用による恩恵だけではなく、エンジニアリング手法や品質の基礎、マネジメントなど開発者が意識すべきことが示されているため、知識ベースとしても、実行における指針としても、参照すべき事項が多い。

本論では、まず、IT 産業にかかわる IT 人材の育成について、現在の産官学の取り組み状況を確認し、次に、これまでの教育環境がどのように変化してきたかについて考察する。その上で、現在の日本ユニシスでは、どのようなソフトウェアエンジニアリング教育に取り組んでいるかを述べ、現在抱えている課題を明らかにして、今後、こういった方向に向かっていこうと考えているかを述べる。

2. IT 人材育成の現状と方向性

経団連の提言「産官学連携による高度な情報通信人材の育成強化」^[1]では、組み込みソフトウェアを含むソフトウェアの開発・利用に携わる情報サービス業の競争力への危機感が叫ばれている。情報サービス産業は、人的資源への依存度が高く、競争力を高めるには、質・量への対応が必要で、今後の IT 化のアキレス腱になりかねない、という懸念からである。政府も、「IT 新改革戦略」を策定し、戦略の重点分野として高度 IT 人材の育成を示しており、IT に関する人材育成は、日本の産業を活性化させる重要な位置を占めていると言える。

2.1 官が主導となった取り組み

産業界や政府の提言を受ける形で、各省が主導となった取り組みが始まっている。経済産業省では、「情報サービス・ソフトウェア産業維新～魅力ある情報サービス・ソフトウェア産業の実現に向けて」^[2]で人材育成への取り組みについて方向性を示すとともに、産業構造審議界の情報経済分科会において人材育成ワーキンググループを形成し、高度 IT 人材情報処理試験やスキル標準を核とした「人材育成プラットフォーム」の構築を推進している。「人材育成プラットフォーム」は、自律的な高度 IT 人材育成を促す全体的メカニズムの構築を目的としており、以下の要素で構成されている。

- 1) IT 産業の収益力向上に基づく人材需給のメカニズム
- 2) 高度 IT 人材の具体像（キャリアとスキル）の可視化と共有
- 3) 実践的かつ先端的な人材育成手法の確立
- 4) 客観性の高い人材評価メカニズムの構築
- 5) 産学連携による実践的教育システムの構築
- 6) 国際的な人材育成メカニズムの確立
- 7) 産学官連携の仕組みと職業人コミュニティの確立

このうち、育成という観点では、制度や連携といったハード面の充実に加え、実践的かつ先端的な人材育成手法の確立が急務であると指摘しており、実践技術としてのソフトウェアエン

エンジニアリングをいかに適用していくかが重要視されている。モデル化技術やオープンシステム技術、組み込みシステム関連技術などの主要なスキルセットとあわせて、特に、ソリューション系人材にとっては、情報システム開発・運用の生産性、信頼性の向上が顧客に提供する価値の中心であるとの考えからである。

また、文部科学省では、産学連携による人材育成カリキュラムの構築を目指して、「先導的 IT スペシャリスト育成推進プログラム」を始めている。このプログラムは、大学院を核に、他の大学や民間企業等との連携を基本とした体制を構築し、人的・物的資源を集約したカリキュラムの策定や、教育プロジェクトの開発・実施等を行うことを条件としている。現在、筑波大学、東京大学、名古屋大学、大阪大学、九州大学、慶應義塾大学の大学院が拠点校となり、周辺に連携校を置く形になっている。連携校では、教員の融通、授業内容や教育方法の改善・向上をさせるファカルティ・ディベロップメント(FD)²に関して協力している。また、IT 産業の発展への貢献という観点から、ソフトウェアエンジニアリング教育の徹底と、IT を活用していく視点で経営学をはじめ IT 以外の専門領域と IT のダブルメジャーを習得するなどのカリキュラム対応を進めている。

2.2 学の取り組み

わが国の高等教育機関における取り組みは、先進諸国と比較して十分なレベルに達していないと指摘されている³。わが国では、高度 ICT 人材に関する国家戦略が弱いだけでなく、大学・大学院教育で、米国や韓国、インド、中国のような資金、人材、ノウハウを含めた産学官の協力体制がなく、教員も研究によって評価される傾向があるため、十分な教育を実践できていないと言われている。

産業界からの高等教育機関への要望は強い。産業界ではインドのように高度 ICT 技術者の育成を国家施策として実施している国への脅威を感じている。最初からグローバルで活躍することを前提としているインドのような新興国の台頭によって、国内 IT ユーザー企業自身が、いずれ標準化された低廉な IT 商品にスイッチし、国内 IT サービス企業はハイエンドな高付加価値分野でしか生き残れないのではないかと、といった強い懸念を抱いているからである。企業としてもより高度な人材を早く育成するカリキュラムへの対応を迫られているが、入社時点で諸外国に比べて 3～4 年の遅れを取っているのが現状であり、早期に高度な IT プロフェッショナルを育成していかないといけないという課題を抱えている。

このような中、先の文部科学省の主導による施策が、ようやく 2006 年から開始されたところである。ソフトウェアエンジニアリングの知識体系にもとづく教育カリキュラムの編成に加え、個別の技術にフォーカスするだけでなく、PBL(Project Based Learning)手法のような、現実的にエンジニアが共同してシステムを開発していくという点にフォーカスした実践的な教育手法を取り入れながら、システム開発プロジェクトにおけるさまざまな課題に取り組むといった育成についての試行を進めている。

2.3 産の取り組み

IT の進化によって、IT そのものがビジネスを変容させるパワーをもち、経営と IT が一体となってビジネス変革にチャレンジしていく時代になって、IT 産業においても、開発・生産・提供の基本構造に変貌がもたらされた結果、新たなスキルセットが要求されている^[3](表 1)。

表1 新たに求められるスキルセットについて

	スキルセット項目	概要
1	サービス型への対応	SaaS (Software as a Service) に代表されるようにITを用いた顧客へのサービスの提供方法に変化が起きており、サービスを連携、組み合わせていくSOA (Service Oriented Architecture) のような設計技術への取り組みが必要となり、プロセス、データについての知識やスキルを要求されている
2	オープンな環境への対応	主要なオープンコミュニティでの開発や製品間の統合などが増加する中、相互運用、連携に必要な標準化への対応能力や国際的なコミュニケーション能力などが要求される
3	モジュール化への対応	ソフトウェア産業の競争力を維持するためには、作り込みが中心のシステム開発から、データの標準化や業務の標準化を推進し、共通部品の再利用を促進させていく必要がある。そのためには、業務や製品への深い知見と、システムをいくつかのモジュールとして捉え、統合し、最適化していく能力が求められている
4	コモディティなITの進化への対応	仮想化技術の進展など、それまで制約を受けていた部分が開放された結果、社会にどのようなインパクトがおきるのかをテクノロジーを軸に予測しながら、その上で展開されるビジネスへの対応をどうしていくかなどの見通し、提案やサービス提供を行う能力が必要となってきた。こういったITパワーを捉えるためには、基礎的な技術要素である、OS、コンパイラ、データベース、ネットワーク等へのスキルが改めて必要となっている

表1のようなスキルセットの変化は、IT産業で働く人材へのスキルチェンジと専門性、および、多様な技術への対応を要求している。そのため、それぞれ専門家を有しながら、全体として品質を上げるといった難易度の高いシステム開発を実現する必要がある。したがって、プログラマ SE シニアSEといった単純な階層と、データベースとネットワークという単純な棲み分けでは、人材を捉えるのが難しくなっているのが現状である。そこで、人的資源の可視化を目指し、経済産業省で定めるITスキル標準(ITSS)の自社への適用が進んでいる。アプリケーションスペシャリスト、データベーススペシャリストなどの高度な専門技術を有する人材に加え、システム全体の構造と技術的な意思決定への責任をもつITアーキテクト、高難度なプロジェクトマネジメント技術を有するプロジェクトマネージャなどに別れている^[4]。

3. 教育環境の変遷

コンピュータが本格的に商用利用されてまだ30年程度である。製造業をはじめ他の産業と比較しても、はるかに少ない時間しか経っていない。それだけに、これまでどのような教育環境であったのかを振り返ることは十分ではないかもしれない。しかし、他の産業にはない急激な技術の進展と社会への浸透を考えると、わずか30年程度とはいえ、どのような環境にあったかを捉えることは、育成が急務であるとされているIT産業の今後について考える上で必要ではないかと考える。ここでは、汎用機を中心としていた時代と、オープンシステムと呼ばれるクライアントサーバー型が広く登場した時代、インターネットが普及したあとの情報システムの時代(現在)の三つの時代にわけ、対象領域と技術ドメイン、開発テスト環境、教育の特徴といった観点で考察する。

3.1 汎用機中心の時代

汎用機時代は、プロプライエタリな環境にあり、ベンダ独自のハードウェア、ソフトウェアが前提となっていた。個々のベンダは、独立した環境で独自性を磨き、高価なシステムを利用して業務効率化をいかに実現するかが、主軸であった時代である。開発にあたっては、コンピュータに詳しい専門技術者が業務システムの構築から運用まで、顧客の機械化担当部門（たとえば、情報システム部）と共同で実施していた時代である。

3.1.1 対象領域への対応

業務効率化が主体であり、業務ドメインへの深い理解が必要である。オンラインシステムと呼ばれる、遠隔からデータ収集しコンピュータ上にデータベースとして集中管理する形態のシステムでは、業務アプリケーションも集中管理され、大量のトランザクションをいかに効率的に処理するかが、もっとも重要な課題であった。オンラインでのトランザクション処理、大量データのバッチ処理効率化などがもっとも高い関心事である。

3.1.2 技術ドメイン

独自アーキテクチャが中心で、ネットワークはクローズ。業務ドメインを深く理解し、いかにコンピュータ上で実現するかが関心事であり、業務で使われるデータをどのようにデータベースに格納していくか、構造化分析・設計・プログラミングといったトップダウンアプローチによって、ステップバイステップで詳細化していくウォーターフォール型開発プロセスによる開発が主流である。

3.1.3 開発やテストの環境

高価な汎用機上で開発・テストを実施することが前提で、開発では人的資源以外に、コンピュータリソースにかかる費用が大きく、いかにコンピュータを使わずに開発やテストを進め、コーディング段階までにいかにバグを検出できるかに重点を置いたことが、この時代の特徴である。綿密な設計図を描くことによるプログラムの論理構造の事前チェック、プログラムコードとほぼ同一のレベルまで記述した設計書による机上の実行シミュレーションが求められる。その後、第四世代言語^{*4}の登場により、コード生産率は飛躍的に向上し、プロトタイピング技術を活用した開発や、エンドユーザー部門に近い業務担当でもコンピュータを扱えるような環境が準備されたが、原則、プログラムは、「リスト上で考えるもの」であり、不具合が発生すると、人が紙上をサーチし原因を見つけた。すべてのコードが人の頭にほぼ入るぐらい実行コードのイメージができた時代である。

3.1.4 教育

固定化されたアーキテクチャを前提とした教育が中心となっていた。他社の技術や標準化された技術への追随は必要だが、あくまで自社技術への理解を深めることが主目的であり、言語の教育とプロダクト（製品）教育が多い。現在のように、同じ言語がさまざまな分野で適用されるのではなく、適用分野に応じて言語が分かれている点も特徴的である。たとえば、高度なシミュレーションが必要な分野は FORTRAN、業務アプリケーション系は COBOL、ミドルウェアなどのプロダクトはアセンブラが特殊言語であった。基本的なアーキテクチャに変化が

ないため、特化した分野で、長い時間をかけて、個々の技術を習得し、修練させていけばよく、組織が変動しなければ、先駆者が後輩に技術を伝承することも、日々の職場での OJT として自然に機能したと想定できる。

3.2 オープンシステム時代

オープンシステム時代と呼べる時期は明確ではないが、ほぼ 1990 年代に入ってから UNIX システムの商用利用が広がっている。当初は、汎用機で集中処理する分野以外に適用されることも多く、独立したクライアントサーバー環境として稼働していることもあった。オープンと呼ばれるのは技術特性もさることながら、ソースコードが安価に配布されたということにも起因しており、そういった背景もあって、研究機関での利用が中心となっていた時代もあった。汎用機によるシステムとはまだ独立した環境にあったが、汎用機では不向きな分野にコンピュータの適用範囲を広げたのがオープンシステムであった。また、Windows ベースの PC や PC サーバーの普及も始まった。

3.2.1 対象領域への対応

UNIX という共通のオペレーティングシステムを軸に、C 言語やテキストエディタなどインタラクティブで標準的な開発環境が登場した。汎用機に比べると比較的安価にシステムを利用可能であったため、研究者を中心に、個人が自由に環境を試すことが可能となった。また、グラフィカルなインターフェースが可能となったこともあり、汎用機では実現が面倒だった分野で適用が進み、業務への適用範囲が広がった。

オープンシステム上で構築するシステムは比較的規模も小さかったこともあるが、汎用機の基幹系システムとは違い、新しい技術による新しい業務への適用と、汎用機システムの一部をダウンサイズする動きが活発となった。プロプライエタリな世界からオープンな世界への技術的チャレンジと、あらたな業務への適用方法の検討など、柔軟な対応が開発上必要となってきた。また、Windows ベースの PC の普及と PC サーバーによるクライアントサーバー型システムが普及しはじめ、よりエンドユーザーに近いところでの業務のコンピュータ化に対応する必要が出てきた。

3.2.2 技術ドメイン

主に UNIX や C 言語、Windows、LAN などのオープン化された技術が中心で、これらの技術ではハードウェアとのインターフェースが隠蔽されたので、ハードウェアから独立してソフトウェア開発ができるようになった。また、クライアントが GUI ベースになり、画面設計とデータベース設計を一緒に進めていく RAD 開発も多くなった。新しい言語が登場し、画面を作成しながら業務システムを構築できるようになったことに加え、個人が独習しやすい環境ができたことによって、開発へのハードルが下がったことも特徴的である。

ただ、スケールアップする仕組みや、対障害性、大量のトランザクションを効率的に処理するといった点では、相変わらず汎用機による集中システムが中心であり、いわゆるクライアントサーバー型システムによって、すべての汎用機システムをダウンサイジングするようなブームもあったが、現実的には適用範囲が限られ、結果として、さまざまなシステムが分散して乱立する傾向が進んだ時期である。

3.2.3 開発やテストの環境

VB (Visual Basic) に代表されるように、クライアントサーバー型のシステムを構築する場合、高価なコンピュータシステムは必要なく、ある程度の性能があれば、個人の PC 上で開発が可能となった。また、画面を作成し、データを定義し、データベースへ配置するといった一連の作業も、開発支援ツールの発達により、シームレスに実施することができるようになったため、開発の生産量は飛躍的に上がるとともに、詳細なコードを書かなくても、プログラムを稼働させることができるようになっている。

3.2.4 教育

ビジネスとして、企業システム構築の中にクライアントサーバー型システムが含まれるようになったことを受けて、教育体系の中に新しい開発言語とプロダクトが含まれるようになった。この時代は、汎用機とオープンシステムが混在し、両者に向かって教育体系を考える必要があったが、おそらく当時は、両者の本質的な違いをあまり意識せず、言語やプロダクトの違いにのみ注目し、教育を進めていたのではないかという懸念がある。つまり、教育を実施する前提として最終的に同じアーキテクチャに向かっている場合と、まったく違ったアーキテクチャで構築することを目的とした場合では、教育のあり方にも違いが出て当然なのに、それに対応した教育環境（研修も OJT も）を作ることができなかったのではないかと考えるからである。

汎用機時代の開発の特徴として、コンピュータをいかに使わないで開発を進めるか、を軸に開発をしていたが、オープンシステムでは、トライ＆エラーを繰り返しながら徐々に作りたいものに近づけていくという開発方法であるため、それぞれの開発に対する考え方に距離感ができてしまう。対象となるシステムの規模や適用範囲もさることながら、開発方法とともに培ってきた文化が、それまでの常識では測れない開発のやり方に直面したとき、うまく融合できなかったのではないかと想像する。

3.3 インターネット時代

インターネットの普及と、UNIX サーバーや PC サーバーなど、安価にハードウェアを調達できる環境が整ったことで、ハードウェア構成の考え方から、ソフトウェア（特にミドルウェア）に搭載すべき仕組みへの要求が変化してきている。利用者は不特定であり、トランザクション量がピークが読みにくい。回線も専用ではなく、一般のネットワークを経由するケースもあり、ボトルネックが読みにくい。さらに、セキュリティ対策など新たな課題もクローズアップされている。このような中、IT の利用は加速度的に進み、メールや Web によるコミュニケーションの変化、システムの提供する価値が、提供側から消費者側に移ったことを受け、ビジネスサービスの提供を担う位置づけとして、システム化への要求はますます複雑になってきている。

3.3.1 対象領域への対応

汎用機による基幹系システムの構築、業務効率化の実現とオープンシステムによる適用業務範囲の拡大を受けて、企業の情報システムは分散や統合など、企業全体を意識したシステムの構築が必要となってきている。複数の IT 環境で構築されたシステムを、業務の統合やサービ

スの統合という形で効率化し、新たな効果を生み出す方向に力が注がれ、IT は経営戦略の重要なツールとなってきた。現在の IT による価値は、業務効率化と新たなビジネス価値の供出にある。守るべきところを守り、攻めるべきところを攻めるといったメリハリ感を、システム開発者側も意識せざるを得ない状況にある。

大量のトランザクション処理や大量データを格納するだけでなく、大量データを分析し経営に活かす、トランザクションを分析し顧客の行動分析を行うなど、より複雑な要求への対応が必要となっている。

3.3.2 技術ドメイン

汎用機中心であった大量のトランザクション処理を、新たな技術によって置き換え可能な時期に来ている。UNIX や Windows は、すでに企業の基幹系システムの基盤として普及し始め、銀行やチケット予約など汎用機の独壇場であったミッションクリティカル分野への進出も始まっている。一時、分散していたシステムは再び集中傾向にあり、サーバーは分散しながらもシステムそのものは集中管理していくといった仕組みも多くなっている。Web サーバーやアプリケーションサーバー、データベースサーバーが連携してひとつのアーキテクチャを実現する仕組みであることを理解しておく必要がある。

またデバイスの進化が、携帯端末やゲーム機、センサー情報など大量のデータ処理を要求しており、より詳細なデータ分析をしながら、経営への価値を生み出す仕組みに添えていかなければいけない。

3.3.3 開発やテストの環境

開発環境は、モデル化技術の進化や設計図からのコードやスタブ自動生成、テストの自動実行など、あらゆる面で進化している。PC の性能が進化したこともあり、ほとんどの開発を自分の PC 環境で行うことができる。最終的なプログラムモジュールは、サーバー環境にあわせて配置するだけで実行でき、複雑なアプリケーションを手元で開発可能である。統合開発環境 (IDE : Integrated Development Environment) と呼ばれる対話型の開発環境では、コンパイラ、テキストエディタ、デバッガなどが統一して使えるようになっており、ソフトウェア構成管理との連携も可能である。

また、オブジェクト指向言語の適用が進み、プログラムのモジュール化が進んだため、分散環境で並列に開発し、統合していくことが可能となった。WebTV 会議などのインフラを活用すれば、インタラクティブなコミュニケーションも簡単に取ることができるため、遠隔地でのコードレビューなども可能となり、最早、インターネットに接続できれば、距離は問題にならないようになっている。

変化する要求に対応するため、繰返し開発や段階的開発などの手法に加え、ライトウエイトなプロセス、アジャイルなプロセスなどが登場してきた。

3.3.4 教育

現在は、主流となった言語やアーキテクチャを中心に教育が進んでいる。Java や C# といった言語をベースに、Web システムの基本、ネットワークの基本、J2EE や .NET フレームワーク、データベースシステムなど広範囲にわたり、全体のアーキテクチャを構成する技術への

理解を深めていく必要がある。

現在、教育を行っていく上での難しさは、多様な技術やプロダクトを極めていくだけではなく、全体をどう構成していくかといった広い技術的な視野を持ち、変化するビジネス要求に対応していかなければならない状況で、自分の得意領域を修練していくことに加えて、他の技術領域への見識を持って対応できる人材をいかに育成していくか、である。

こういった人材の典型として、近年登場したアーキテクトと呼ばれる職種^{*6}は、広範な技術と問題領域をカバーする人材として期待されている。

4. 日本ユニシスのソフトウェアエンジニアリング教育の状況

日本ユニシスでは、ビジネス環境の変化、技術者に求められるスキルや知識の多様化が進む中、従来のやり方ではエンジニアの育成に十分な効果が期待できないとの危機感から、新しい人材育成のスキームとして、人的資産を可視化する「人材モデル」と育成計画や個々人のキャリアデザインのベースとなる「キャリアパス」を定めて育成カリキュラムを策定し、実施、運用している^[5]。

4.1 人材モデルとベーシックキャリアパス

日本ユニシスの人材モデルは、事業を遂行する上で必要となる人材を役割の観点からモデル化したもので、このモデルを軸に、育成の方向性と社員のキャリア形成の指針として、ベーシックキャリアパスを示している。人材モデルは、原則としてITスキル標準とモデルの切り分け方、レベル基準などを合わせているが、一部、日本ユニシス独自に定めたモデルがある^{*6}。その中の一つが「ソフトウェアエンジニアリングスペシャリスト」だが、このモデルは、システム職全員が入社してまず目指すべきモデルとして、キャリアパス上で位置づけている(図1)。

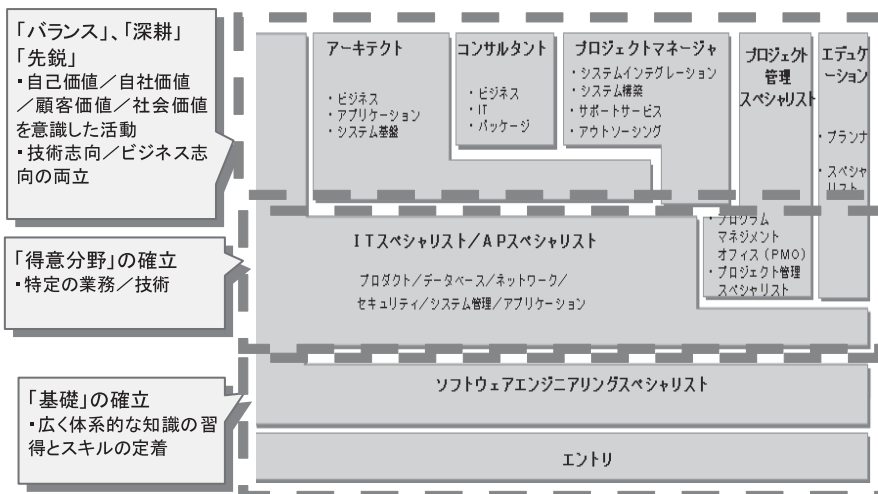


図1 人材モデルとベーシックキャリアパス

ソフトウェアエンジニアリングスペシャリストは、ソフトウェアエンジニアリング技術を用い、開発環境の構築・整備を行うとともに、構築フェーズでは物理設計やプログラミングを担当するが、同時にプログラム開発要員の指導の役割を指す。

キャリアパスとしては、入社後にまず新人研修でシステム開発に必要となる広く体系的な知識を習得したあと、ソフトウェアエンジニアリングスペシャリストを目指し、各種研修や現場でのOJTを通してスキルを定着させるようにした上で、各人が得意分野を磨き、スペシャリストとして活躍することを期待している。実際にビジネスにおいてパフォーマンスを発揮するのはこの段階からになるが、得意分野だけではなく、広く体系だったものの見方もあわせて習得するようにしていく必要があると考えている。これは、ビジネスを遂行する上で、エンジニアとして技術を磨くだけではなく、情報システムに関するサービスを提供できる「プロフェッショナル」としての活動を期待してのことである。

その後、ハイパーフォーマーとして活躍し、高い価値を提供できるようになるためには、自己価値をしっかりと確認し、広く顧客価値や自社価値、社会価値を意識した行動を身につけるとともに、バランスや深耕、先鋭といった感覚を磨くことを期待しているが、あくまで技術者としての軸足がぶれないようにしなければならないとも考えている。最終的に、日本ユニシスのエンジニアには、さまざまな価値やビジネス要求に対していかにして応えていくか、その中で、どのような技術を選択していくべきか、ITがどのような影響をもたらすのか、といった技術者としての責務を果たす必要があるからである。その意味で、まず基礎の確立は必須であり、表面的な技術習得や技術の偏りがないようにキャリアを形成していく必要がある。

4.2 基礎教育としてのソフトウェアエンジニアリング

ソフトウェアエンジニアリングスペシャリストとしての知識やスキルを定着させるために、新人研修を含めた5年間程度、これまで若年層に実施してきたカリキュラムを義務教育として整備しなおし実施している^{*7}(表2)。研修の流れとしては、まず新人研修を経たあと、残り2.5年を義務教育、次の2年を義務教育として実施するスキームとし、必須科目として受講する仕掛けを構築している。義務教育化した背景には、若年層での基礎の確立を重視していることと、実際の開発現場での体験を体系的に整理し、スキルとして定着させていくという、育成サイクルの実現を目途としている。

新人研修では、必ずしも一致していない知識レベルをあわせていくために、ITの基礎から開始し、言語や構造化プログラミングを通して基本を学び、Webの仕組みとWebベースでの開発をUMLで実施するなど徐々にレベルをアップしていく形を取っている。最後に、実際のプロジェクトを模して、システム開発における役割を意識させながら、プロジェクト演習を行っていく。プロジェクト演習は、いわゆるPBL手法をベースにしているが、実際の開発現場に近いケースを導入することでより実践的な効果を狙うものである。新人の中には、情報工系で知識も十分であり、プログラミング経験もあるものがあるが、ビジネスとしてのシステム開発、プロジェクトによるシステム開発は、なかなか高等教育機関では教育できない実践的な研修カリキュラムであり、ここでプロフェッショナルになるための最初の一步を踏み出させる形をとっている。

その後の義務教育では、技術系や人間系の研修に加え、米国ユニシス社が長年、知識ベースとして蓄積し、ナレッジ化されたTEAMMethod^{*8}を活用し、知識を体系的に教えていくことに加えて、社内のビジネスプロセスや見積りなど、実業務での役割を意識したカリキュラムも提供している。

表2 義務教育のカリキュラム

	義務教育 I	義務教育 II
ソフトウェアエンジニアリング系	<ul style="list-style-type: none"> ソフトウェアエンジニアリング概論 テストとレビュー TEAMDesign 開発工程リバース セキュア開発 IT基礎 (eLearning) 	<ul style="list-style-type: none"> プロジェクトマネジメント入門 システム開発ビジネスプロセス概要 見積り技法基礎 プロジェクト計画とコントロール
技術系	<ul style="list-style-type: none"> Microsoftテクノロジー .NETテクノロジー Javaテクノロジー Linux Oracleデータベース CISCO JPI 	<ul style="list-style-type: none"> オブジェクト指向 モデリング技術 (ネットワーク技術) (開発技法) (セキュリティ)
人間系	<ul style="list-style-type: none"> ロジカルシンキング サービスマインドコミュニケーション 	<ul style="list-style-type: none"> ファシリテーション基礎 交渉力基礎

4.3 ハイエンド人材モデルに向けての教育

ハイパフォーマンスを発揮し、技術者として高レベルな知識やスキル、経験を有する人材(ハイエンドな人材)を育成していくためには、いくつかの条件がある。

- ・ソフトウェアエンジニアリングの基礎を確立しており、開発・保守・運用のライフサイクルで、系統的に、定量的なアプローチができること。
- ・専門性をもって価値を生み出すことができること。
- ・上記二つを条件に、多様な価値を受け入れ、柔軟性とバランス、深耕、先鋭といった感覚を磨いていくこと。

ベーシックキャリアパスにもあるように、まず、ソフトウェアエンジニアリングスペシャリストとしての基礎ができていることが条件となって、各スペシャリストとして専門領域を深めていく設定になっている。その後、そのままスペシャリストとしてさらに極めるか、ハイエンドに設定されたモデル(アーキテクト、プロジェクトマネージャなど)に進むことになるが、前提となる人材モデルで一定以上のレベルを超えていなければならない仕組みとしている。

たとえば、アーキテクトは、最適なアーキテクチャ設計を行うために、ソフトウェアの要求をしっかりと捉え、要求を分析し、技術的な観点から実現可能性を評価し、いかにシステムを作り上げるかを構想していかなければならない。開発プロセスはどうすべきなのか、テストはどのように実施すべきなのか、開発におけるすべてのシーンにおいて、構想をしていく必要がある。こういった構想を現実のものとして着地させるためには、広い知識やスキルだけではなく、スペシャリストとしての得意領域をしっかりと身につけていることが必要となっている。

4.4 時代のギャップに対応する施策

現在は、新人教育からの流れとして義務教育制度やキャリアパスで目標人材を示すなど、さまざまな施策による育成体系を提供しているが、こういう体系になったのは、ITスキル標準

の公表に合わせて大幅改訂をおこなった 2003 年からである。それ以前は 1998 年頃からスキル調査を行い、人材モデルを通して人的資源の可視化を行ってきたが、ちょうどこの時期から 5 ~ 6 年間はインターネットが個人や職場の環境に急速に普及しはじめ、情報システムも、J2EE や .NET ベースに移行しはじめ、技術者の役割が大きく変化していたころであり、人材モデルをベースに体系だった研修を実施しにくい時代でもあった。

1990 年代前半かそれ以前の入社となると、汎用機中心かオープンシステム中心かの差はあるものの、明確な目標人材という点ではいずれもなかった時期で、入社後はプログラムを書き、その後要件定義や設計を行う SE となり、技術を磨いていく中で、上級 SE になっていくといった流れが暗黙に意識されてきた時期であった。

問題は、急に新しい体系を設定されても、個々人がまず自分がどの人材モデルなのかが分からないことである。たとえばずっと汎用機に携わってきた人にとって、現在のような細かに設定された人材モデルは、なじまない部分があるからで、実際にレベル判定をしようとした場合にも、汎用機上での経験やスキルは十分なのに、現在のシステム開発の環境にある程度重点を置いたレベル設定条件下では、スキルや知識が十分でない、という結果となる。単に最近の技術や流れを知らないことや、細分化した人材モデルにうまく当てはまらないことが、本当にその人のパフォーマンスの評価と言えるのか、その点については、レベル診断の方法についても少し配慮が必要だろう。

しかし、少なくともスキル、知識レベルだけで見れば、最新の技術動向や開発環境に目を向けてほしいという考えから、スキルチェンジを目指した教育環境を準備している。最新の開発とはこういったことをやるのだ、という体験を通して個人のスキルや経験と融合させ、新技術習得へのきっかけになるようにしている。たとえば、Java 言語で IDE を使ってプログラムを作成し、デバッグ、そして実行シミュレーションや実行環境への配置、また、オブジェクト指向の考え方や UML を使った要件定義や設計書の書き方などである。

5. 今後取り組むべき課題と施策の方向性

日本ユニシスでは、現在、人材の可視化戦略の中で、今後、どのような人材をどれだけ育成すべきかを考え、育成計画につなげている。しかし、時代の変遷によって、求められていた人材像が違ってきていたこともあり、新しい人材を育成してだけでなく、今の人材をいかにスキルチェンジしていくか、についても課題となっている。

新しい人材に関しては、システム開発に従事する「プロフェッショナル」を目指して、ソフトウェアエンジニアリングの基礎をしっかりと固め、個々のスペシャリティを磨くことで、パフォーマンスを出し、さらにハイレベルな人材への道を示すことで、個人の成長と会社としての成長を一致させていくように考えている。

しかし、今の人材に関しては、個々人のレベルまで踏み込むと、技術シフトや体系的なソフトウェアエンジニアリングの習得が十分できていないケースがあり、人によってばらつき感が出てくるため、このギャップを埋める施策を打つとともに、世代間で発生しているギャップについても、どのような傾向が見られるかを仮説検証していくことが必要であると考えている。

日本ユニシスでは、人材ポートフォリオを策定し、IT 人材を事業戦略に沿って適性に配置することによって最大の効果を上げる取り組みとして、人材ポートフォリオマネジメントに取り組み始めているが⁷⁾、こういった取り組みに加えて、世代間のスキルギャップをいかにして

埋めていくか、スキルチェンジの効果的な方法はどうすればいいのか、といった施策を打っていくことで、今後、新しい人材の育成と今の人材のスキルチェンジを融合して進めていくことになるのではないかと考えている。

これまで、ソフトウェアエンジニアリングは、ともすればアカデミックなものとして現場とは疎遠な意識を持たれたり、実際はしっかりと工学的にアプローチをしているのに、それがどのような領域のどういった分野でのことなのかを意識せずに進めていたりということもあった。その反省から、改めてソフトウェアエンジニアリングをシステム開発の足元を固める要素として取り上げ、各世代に向けての効果的な研修や、再学習するきっかけを作ることが求められていると感じている。

6. おわりに

本稿執筆のきっかけは、昨今の社会的に大きく取り上げられたシステム障害の中で、人的ミス起因としているケースがあったことである。最近のシステム開発を担う人へのプレッシャーは大きく、そのためか、「Java プログラムができます」、「テストはいわれた指標にしたがってこれだけやりました」といった、何ができるとか、やったとかという方向に向かいがちになり、本来、何がシステム開発に一番大事で、そのためにはどのようにすべきかといった思考が欠けているのではないかと、という危機感を覚えたからである。

システムは、開発だけではなく、運用し、保守も含めて考えなければいけない。安心、安全を提供する情報システムを開発時点の品質だけで評価していいのか、保守・運用プロセスに入って、改善を施していくときの品質はどのように評価するのか、といったことにもしっかりと目を向ける必要がある。

ソフトウェアエンジニアリングは一度学べばいいというものではない。日頃から常に意識して行動すべきものとして捉えるべきで、継続して学習していく姿勢が大事である。さらに高い価値を生み出していくためには、自己価値を確認し、広く顧客価値や自社価値、社会価値を意識した行動を通して、バランスや深耕、先鋭といった感覚を養っていく必要がある。

こういった日頃の意識の持ち方、行動のあり方の変容をもって、システム作りに臨むことが、最終的に、すこしでも人手に起因するシステム障害を減らすことになれば、と思っている。

- * 1 IEEE では、ソフトウェアエンジニアリングとは、ソフトウェア開発、運用、保守に対して、系統的で規律をもった定量的なアプローチと、そのアプローチに対する研究・としている。
- * 2 文部科学大臣の諮問機関である中央教育審議会の大学分科会では、FD を「教員が授業内容・方法を改善し、向上させるための組織的な取組の総称」と定義
- * 3 経団連の意見書による指摘。あくまで先進諸国に比べてだが、新興国の台頭もあり、日本が遅れをとっているのも確かである。
- * 4 米国ユニシス社では、MAPPER、LINC という第四世代言語を保有。コンピュータで実現したいことをパラメータ等の仕様で定義すれば、コードが自動生成(ジェネレート)される。
- * 5 IT スキル標準では、人材モデルを職種として定義している。後述する日本ユニシスの人材モデルは、役割モデルである。
- * 6 「ビジネスアーキテクト」、「サポートサービスプロジェクトマネージャ」、「ソフトウェアエンジニアリングスペシャリスト」、「プログラムマネジメントオフィス」、「プロジェクト管理スペシャリスト」である。
- * 7 これまでも体系的なカリキュラムが用意されていたが、義務教育化という形で実施しはじめたのは、2007 年からである。
- * 8 TEAMMethod は、米国ユニシス社で開発されたソフトウェアエンジニアリングに関する知識体系である。開発のあらゆる局面で、必要となる知識や行動指針が体系的にまとめられて

おり、日本ユニシスの開発基礎を担っている。

- 参考文献**
- [1] 日本経済団体連合会，産官学連携による高度な情報通信人材の育成強化，意見書，日本経済団体連合会，2005年6月21日，<http://www.keidanren.or.jp/japanese/policy/2005/039/index.html>
 - [2] 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，情報サービス・ソフトウェア産業維新～魅力ある情報サービス・ソフトウェア産業の実現に向けて～，経済産業省，2006年9月14日，<http://www.meti.go.jp/committee/materials/g60922aj.html>
 - [3] 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人材育成ワーキンググループ，高度IT人材の育成をめざして（報告書案），経済産業省，2007年4月12日，P3，<http://www.meti.go.jp/committee/materials/g70418cj.html>
 - [4] 独立行政法人情報処理推進機構 ITスキル標準センター，ITスキル標準 V2 2006，経済産業省，2006年10月31日，http://www.ipa.go.jp/jinzai/itss/download_V2_2006.html
 - [5] 吉田恵美，「人材モデルによる人的資源の可視化と人材育成指針の提供」，ユニシス技報，日本ユニシス，Vol.26 No.3 通巻91号，2007年2月，P15～29
 - [6] ITトップガンプロジェクト，ソフトウェアエンジニアリング講座，日経BP社，2007年2月，
 - [7] 堆 圭介，「人材ポートフォリオの開発」，ユニシス技報，日本ユニシス，Vol.26 No.3 通巻91号，2007年2月，P30～43

執筆者紹介 宮 脇 亨 (Tohru Miyawaki)

1984年入社。地域金融機関向けソリューションの開発・適用に従事後，ソリューション主管部を経て，1997年にJava Project 参画。その後，開発技法 LUCINA の策定，開発標準策定コンサルティングサービス，顧客システム企画支援など，主に顧客のIT部門向けコンサルティングサービスに従事。現在，人材育成部 HR 戦略推進室で，IT 技術者育成についての戦略立案などを担当。