

情報化が進展する状況下での 労働者の技能形成に関する考察 ——ソフトウェア技術者を中心に

古谷 真介
(大阪産業大学准教授)

本稿では、1990年代末と2020年代初頭におけるソフトウェア技術者の技能形成について考察した。技能形成に関する先行研究（東京都立労働研究所（1987）、小池（1991）など）では、プログラミングなどの下流工程の易しい作業から要件定義や基本設計などの上流工程の難しい作業の経験を経ることで、技能が形成され、そして中核となる技能は、上流工程の作業を遂行するものであると指摘している。これら先行研究は1980年代の調査に基づいている。IT産業の進化は速いと言われているから、現代日本のIT産業においては、先行研究とは違ったものとなっている可能性がある。そこで、1990年代末のメガバンクのシステム保守および2020年代初頭のIoTシステムの開発プロジェクトの事例を取りあげ、先行研究の議論を検討した。その結果、その技能については、プログラミングなどの下流工程の作業、およびユーザの要求を仕様・設計に転換する上流工程の作業を遂行するものであるとし、それらはOJTをつうじて形成されることを確認した。そして下流工程における技能形成は、3つの時点においても、定型化されていることを指摘した。しかし、上流工程については、下流工程と比較すれば、先行研究が想定するような単線的なものではなく、ソフトウェアの規模と複雑さ、およびプロジェクトの編成状況に応じて、異なるパターンが存在する可能性を指摘した。

目次

- I はじめに
- II ソフトウェアの特徴と開発過程
- III 1980年代における技能形成に関する調査研究
- IV 1990年代末のメガバンクのシステム保守の事例
- V 2020年代初頭のIoTシステム開発プロジェクトの事例
- VI おわりに

I はじめに

現代日本社会とIT産業においては、技術革新が進展している。たとえば、スマートフォンなどの小型、高性能、かつ低価格化したコンピュータが普及し、オペレーティング・システムはLinux、

Windows、あるいはクラウド・コンピューティング・サービスとなり、プログラミング言語はCOBOLなどからPythonなどに取り替わりつつある。さらには、無線などの通信網も文字や音声だけでなく、動画も安価に送受信することができるようになった。またタッチパネルやネットワーク接続可能なセンサーなどの入出力装置も低価格化が進み、普及した。それらによって、工場などの機械設備にセンサーを取り付け、それらの稼働情報を容易に収集することが可能となり、IoTシステムが登場した¹⁾。そしてAIなどを用いて、蓄積した大量のデータから傾向を読み解き、システム全体、さらには社会全体を最適化しようとする動きもある。

このような技術革新が進展していることから、

経済産業省を中心に DX が唱導されている²⁾。従来の技術をレガシー・システムと呼び、これらを新しい技術にもとづいたものに刷新し、企業成長を図らなければならないとの主張がみられる。そして日本社会は、この革新の波に乗り遅れてしまうのではないかと、とも議論されている。

上に述べたような事態が進展したことで、労働者の技能とその形成過程には、どのような変化がみられるのであろうか。それに関するさまざまな言説があるが、その実態を丁寧に観察したうえでの議論は、少ないように思われる。そこで、本稿では、DX と呼ばれる技術革新の鍵となっているソフトウェアとその開発を担っている技術者に焦点をあて、とくに OJT に注目して、その技能形成について考察する。

以下では、次の順に考察する。まず始めに、ソフトウェアの製品としての特徴およびその開発過程とその作業内容について確認する。これをソフトウェア技術者の技能形成の考察の手掛かりとする。次に、3つの先行研究を確認する（東京都立労働研究所 1987；雇用職業総合研究所 1987；小池 1991）。そして 1990 年代末のメガバンクのシステム保守と 2020 年代初頭の IoT システムのプロジェクト³⁾の事例を取りあげる。それら 2つの事例における技能とその形成過程を明らかにし、先行研究と比較し、考察する。最後に、DX の影響は幅広い労働者に及びうると考えられるから、

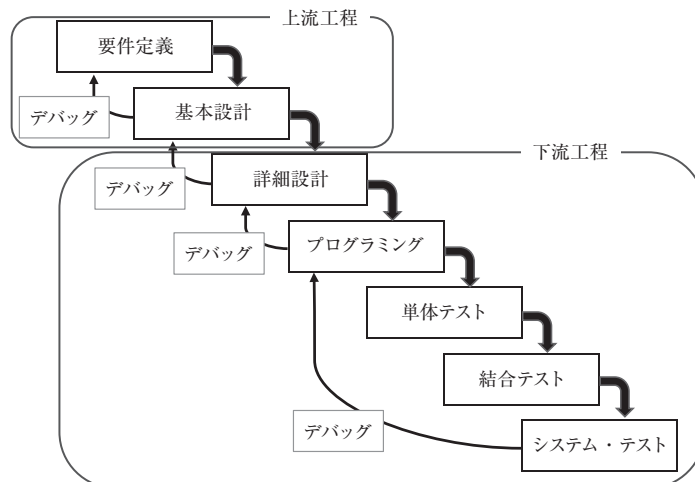
他の職種の労働者や企業にとっての含意についても触れることにする。

II ソフトウェアの特徴と開発過程

まずソフトウェアの特徴とその開発過程を一瞥しておこう。その特徴は、ソフトウェアには形や重さがないということである。ソフトウェアが動いている様を手で触って、耳で聞いて確認することができない。そのことから、工業製品と比較すれば、設計情報などの伝達で、伝言ゲームが発生し易いと考えられる。いかにして、技術者たちは、ユーザのシステムに対する要求を動作するプログラムに手際よく転換するのか。これが焦点となる。言い換えれば、社内外への技術者間の設計情報の伝達費用を最小限化することであろう。

そしてそのような特徴をもったソフトウェアは、図 1 に示される工程を経ることで開発される。すなわち、要件定義、基本設計、詳細設計、プログラミング、単体テスト、結合テスト、ならびにシステム・テストの工程である。これらの工程では、ソフトウェア技術者が、ユーザの業務の流れを明らかにし（要件定義）、設計し（基本、詳細設計）、プログラムを組み（プログラミング、単体テスト）、正しく動作するかどうかの試験を行う（結合、システム・テスト）。そして単体、結合、ならびにシステム・テストにおいて、設計どおり

図 1 ソフトウェア開発の工程編成



出所：有澤（1988：10-15）と玉井（2004：13-18）から筆者作成

に動作しなかったときには、デバッグと呼ばれる作業が行われる。この作業では、その原因を明らかにし、その対処策を考え、プログラムを修正するものである。この作業では、単にプログラムを修正するにとどまらない。必要であれば、詳細設計あるいは基本設計を再検討し、それを修正することになる⁴⁾。こうして基本設計どおりの動作を確認すると、ユーザは、そのシステムを運用することになる。

上に説明した工程編成は、ウォーターフォール型の開発手法と呼ばれる。ソフトウェア工学などの教科書で説明されているものであるが、いくつかの注意が必要である。上の7つの工程は、業種あるいは企業によって呼称が異なる。たとえば、基本設計は外部設計あるいはシステム設計、詳細設計は内部設計、そしてプログラミングはコーディング、実装、あるいは製造などと呼ばれることがある。また上の工程のうち、要件定義から基本設計までを上流工程、そして詳細設計からシステム・テストまでを下流工程と呼ぶこともある。さらに要件定義からシステム・テストまで順番に行う場合、基本設計から結合テストまでの工程を繰り返し行った後に、システム・テストを行う場合、あるいは要件定義から基本設計までの工程と詳細設計から結合テストまでの工程が並行して行う場合もある⁵⁾。

Ⅲ 1980年代における技能形成に関する調査研究

それでは次に、上のようなソフトウェアの特徴と工程編成をふまえて、技術者の技能とその形成過程に関する先行研究を確認しよう。東京都立労働研究所(1987)、雇用職業総合研究所(1987)、小池(1991)の3つの調査研究があげられる。

まず、東京都立労働研究所(1987)では、技術者を「基本設計、詳細設計、ならびにプログラム設計」を担っているシステムエンジニアと「プログラム設計、コーディング」を担っているプログラマに大別する。そしてプログラマについては「電算機オペレート、データ入力→コーディング→プログラム設計→システムの詳細設計→シ

ステムの基本設計→管理」という業務の経験を経て、システム・エンジニアになることを指摘した。システム・エンジニアについては「コーディング→プログラム設計→システムの詳細設計→システムの基本設計→管理」という業務の経験を経ていることを指摘した入社すると、新人技術者の大半がプログラマとなり、その後経験を積み、システム・エンジニアとなることを明らかにした⁶⁾。

雇用職業総合研究所(1987)では、年齢とともに職種がかわり、担当する業務もより高度化していることを指摘している。たとえば、担当業務別では、経験年数が短いときには、「プログラムコーディングと単体テスト」と「プログラムの修正・保守」を担い、経験年数が長くなると「総合的計画管理的業務」と「プロジェクトの計画と管理」へと、下流から上流工程の仕事を担うことを指摘した⁷⁾。

最後に、小池(1991)について見よう。まず、ソフトウェア開発が「プランニング」「システム設計」「プログラム設計」「プログラミング」「テストとメンテナンス」の工程からなっていると、その中で最も重要かつ「むずかしい」工程は「プランニング」と「システム設計」であるとしている。そして、その工程の作業を遂行する技能は、「業務分析」および「それにもとづき顧客を説得する折衝力」であるとしている。その技能は、さまざまなソフトウェア開発のプロジェクトに所属し、前任の技術者がユーザと折衝する時に、同行し、それを観察することなどで知識と経験を蓄積し、さまざまな業種の業務あるいは同一業種の異なる企業での類似した業務の流れを分析することで、技能が形成されていることを指摘した。その過程は、プログラミング、詳細設計、基本設計、そして顧客との折衝を経ていることを明らかにした⁸⁾。

これらの研究以外にも戸塚・中村・梅澤(1990)もあげられる。この研究では、ソフトウェア開発を担う組織が「流動的プロジェクトチーム方式」によって編成されていることから、技能形成が進みにくいのではないかと言う論点を考察したうえで、先行研究と同様の過程を指摘した⁹⁾。それら

の研究を、OJTに注目して要約すれば、技術者については、ソフトウェア開発の下流工程の易しい仕事を経験し、それに習熟するとより上流工程の難しい高度な仕事を経験することで技能の形成が進むと整理して良いであろう。

ソフトウェア開発の企業は、採用の時点などで、技術者の技能を見極めることは容易ではない。そうであるならば、実際に働かせてみて、その実績などを評価することが、次善の策となるであろう。易しい仕事を課し、それができるようになると、より難しい仕事を課し、その実績を評価する。そうすることで、その技術者の技能を見極める。このような方法が、技能情報の収集に関する取引費用を軽減することになる。OJTによる技能形成は、企業にとっては、妥当と思われる¹⁰⁾。

以上の先行研究は、1980年代の調査にもとづくものである。その時代に、技術者たちは、大型コンピュータに接続されたオンライン端末とCOBOLなどのプログラミング言語を用いて、ソフトウェアを開発していた。一部の企業では、オンライン端末ではなく、コンピュータを操作する場合には、パンチカードなどを利用していた時代である。レガシー・システムの時代の調査研究である。2023年現在のソフトウェア開発は、インターネットなどのネットワークの利用が前提となり、ソフトウェア開発の支援ツールも進化を遂げ、普及し、スマートフォンやウェアラブル・デバイス向けに開発されている。このような技術革新が進展したことにより、技能とその形成過程にも変化が生じているかもしれない。また2000年以降、ソフトウェア技術者の技能形成に関する研究は、乏しいように思われる¹¹⁾。そこで、次にインターネットの普及期の1990年代末とDX時代の2020年代初頭に行った調査を取りあげて、ソフトウェア開発の技術者の技能とその形成過程についてみよう。

IV 1990年代末のメガバンクのシステム保守の事例¹²⁾

1 メガバンクのシステム保守プロジェクトの編成と開発工程

それでは、1990年代末のメガバンクにおけるTSB社のシステム保守プロジェクトAを事例として取りあげよう¹³⁾。このプロジェクトは、マネージャT氏に率いられており、3つのチームから編成されていた¹⁴⁾。この3つのチームの中で、共通関数の開発・保守チームを対象とする。このチームは、1990年10月時点では、5名の技術者から編成されていた。TSB社の技術者が、入社10年目以上、9年目以上、4-5年目、ならびに新人がそれぞれ1名、そして協力会社の技術者が1名であり、そしてこのチームのリーダーは、マネージャのT氏が兼務していた。

このチームの業務である共通関数の開発・保守について説明しておこう。まず、システム保守とは、既存のシステムに対する修正あるいは機能追加といった業務を指す。具体的には、不具合などの修正、法律の改正と制度の変更などへの対応、ならびに新しい金融商品・サービスへの対応などである。その規模は、スマートフォンのメニューに表示されるボタン位置の変更といった小さなものから、新たな金融規制への対応などの大きなものまでである。そして、共通関数とは、プログラムの共通部品である。基本設計の工程において、システムを構成するプログラムの中で共通した処理の部分抽出し、これを関数と呼ばれるプログラムとして定義する。その関数を他のプログラムに組み込んで用いる。1つの共通関数についてみれば、開発・保守するプログラムの規模は小さいが、複雑なものである。多くのプログラムに組み込まれていることから、1,2ステップ程度の修正であったとしても、システムの広い範囲に影響する。この影響範囲を考慮したうえで、保守・開発を行わなければならないからである。

さらに、ここで言う規模とは、工数で示される。工数とは、1人の標準的な技術者が1カ月間で行える作業量である。一般的には人月という単

位で示される¹⁵⁾。そして複雑さについては、連携するシステムの数である。連携するシステムの数が多くなれば、複雑さが高まる。また1つのプログラム内における分岐あるいは繰り返しの条件が多い場合なども、複雑さを高めることになる。

次に開発工程についてみよう。メガバンクにおいて、新規開発あるいは大規模な機能追加・改善などの保守の工程は、要件の確定、機能設計、技術設計、詳細設計、コーディング、単体テスト、結合テスト、ならびにシステム・テストという8つの工程からなっている。それらの工程の作業内容は、以下のとおりである。まずメガバンクのシステム企画部が機能追加と改善の要求項目に優先順位をつけ、高いものをTSB社のプロジェクトに伝える(要件の確定)。次に、TSB社はメガバンクの担当者から聞き取りし、機能を定義し、そしてその工数を見積もる。メガバンクのシステム企画部は、その見積の工数、優先順位、ならびにシステム開発の予算から、その要求をシステムとして実現するかを判断する(機能設計)。

そのシステム開発・保守を行うことが決定されると、TSB社のプロジェクトは、ソフトウェアの構造と開発するプログラムなどを定義する(技術設計)。そしてプログラムの1行毎の処理の流れを明らかにしたモジュール仕様書を作成し(詳細設計)、プログラムを組み(コーディング)、組んだプログラムをテストする(単体テスト)。そして、作成したプログラムを結合してテストを行う(結合テスト)。結合テストが完了すると、日常業務でのシステムの稼働条件と同じ条件のもとで、テストを行う(システム・テスト)。最後に、ユーザがリリースを決定すれば、エンド・ユーザが、そのシステムを日常業務で利用することになる。

以上がメガバンクにおける新規あるいは大規模な機能追加と改善における工程とその作業内容である。図1に示したウォーターフォール型の開発手法である。ただし、すべての開発・保守の業務がすべての工程を経るとは限らない。たとえば、共通関数の開発・保守チームは、規模は小さいが、複雑なプログラムを開発・保守している。そのことから、実際に行われる作業は、多くの場合には、技術設計工程以降の作業、あるいは機能設

計の一部変更とそれ以降の作業に限られる。8つのすべての工程の作業を行っていない。またエディタやデバッガなどのソフトウェア開発支援ツールが高性能となり、普及したことによって、詳細設計から単体テストの工程は、1つの工程として取り扱うようになった。つまり、モジュール仕様書を作成し、プログラムを組むという過程を経ずに、技術設計書から直接プログラムを組んでいた。

2 メガバンクのシステム保守プロジェクトにおける新人技術者の技能形成

プロジェクト配属後の新人技術者の技能形成についてみよう¹⁶⁾。共通関数チームにおいては、マネージャのT氏は、チームに新人技術者が配属されると、前任技術者の中からOJTトレーナを選出し、その指導のもとで作業を行わせた。OJTトレーナとなる技術者は、入社5年目から9年目の技術者である。そして、プロジェクトがおかれている状況によって作業内容が変わることがあるが、OJTトレーナは、新人技術者には、期間が長く、リスクが低いプログラムを割り当てる。その技術設計とサンプルのプログラムを新人技術者に渡し、それらを参照させて、プログラムを組ませている。そのプログラムは、OJTトレーナが組めば1週間程度のプログラムであり、かつユーザのシステムの一部ではあるが、そのプログラムに不具合が生じたとしても、開発スケジュールなどに与える影響が小さいものである。そのような条件に合ったコーディング作業が存在しないときには、帳票類のプリントアウト、画面レイアウトを示した仕様書・設計書、ならびにプログラムのリストを渡し、プログラムを読ませて、間違いなどがあれば、それらの文書を修正させた。新人技術者の技能形成は、前任技術者から割り当てられたプログラムを組むこと、あるいはドキュメント管理などの付帯業務から始まる。

2年目になると、トレーナなどの前任技術者に指導されずに、割り当てられたプログラムを組むことが求められる。そして、規模が小さくかつ簡素なプログラムであれば、機能設計および技術設計などのより上流工程における作業の一部あるい

は全部を割り当てられ、前任の技術者などから助言をもらいながら作業を行うことになる。

3 メガバンクのシステム保守プロジェクトにおける3年目以降の技能形成——ユーザの業務に関する知識の蓄積

3年目以降の技能形成については、ソフトウェア開発の規模と複雑度、およびプロジェクト編成によって異なる。メガバンクの事例の場合には、「理想的な条件」が整えば、おおよそ以下のとおりとなる。まず「理想的な条件」とは、規模が新規の開発で50人月、あるいは保守で30人月であり、かつプロジェクトの編成がマネージャ、基本設計を1人で担える入社5-6年目の技術者、ならびに技術設計とプログラムを組める技術者から編成されているというものである。これらの条件が整えば、技術者は、プログラミングと単体テストであれば1-2年程度、技術設計であれば2-4年目、機能設計であれば入社5年以上で独力で作業を担えるようになる。システム保守であれば、連携する他のシステムへの影響を考慮しなければならないので、複雑なものとなるので、1年ほど遅れて、入社3-4年目で、技術設計を担えるようになる¹⁷⁾。

しかし、そのような「理想的な条件」が整うことは、システム開発・保守の受注状況によるから、不確かである。たとえば、システムの開発から保守の段階へと移行すると、多くの場合には、プロジェクトはベテランと若手技術者、あるいはベテラン技術者と協力会社の技術者のみから編成されてしまう。そのような編成となると、作業分担が固定化してしまい、新人あるいは若手技術者の技能形成の機会が乏しくなってしまうことがある。または、若手技術者が、規模が小さく、簡素な開発・保守に限られるだろうが、コーディングから単体テストまでの作業に習熟せずに、より早い時期に技術・機能設計の作業に携わることになる可能性もある。つまり、コーディングと単体テスト、技術設計、機能設計という順に進まないこともある。

規模と複雑さ、およびプロジェクトの編成によって、技能形成のパターンに違いが生じるが、

技術者達は、3年目以降は、より上流工程の作業を行う。ユーザの業務に関する知識を蓄積し、それらを元にしてシステムの要求から機能・技術設計書を作成する。その要求を仕様・設計書あるいはプログラムへ転換している。具体的には、共通関数チームの場合には、前任技術者が、作業を割り当てた経験年数の短い技術者に、その機能に対応するユーザの業務、その必要性、機能改善点、ならびにそれによって予想される効果を説明する。これが、ユーザの業務知識を蓄積し、システムに対する要求が、どのような形で仕様書あるいは設計書に転換されているかを知る機会となっている。経験年数の短い技術者は、それらの情報を元にして、前任技術者から助言をもらいながら、必要であれば機能設計の一部、技術設計、プログラミング、単体テストを行う。マネージャのT氏の場合には、1980年代半ばの入社2年目で技術設計と機能設計の一部修正などが主な作業となり、コーディング作業に携わる機会が減少していた。3年目となると、当時銀行業で開発が進んでいた第3次オンライン・システムと呼ばれる大規模開発のプロジェクトに参加し、前任技術者から助言を受けながら100人月程度の複雑でない定時一括処理のシステムの保守について、コーディングなどの作業を担当せずに、機能設計の一部変更からシステム・テストまでのすべての工程を管理・監督した。

このようなことが繰り返され、やがて前任者から助言なしに保守を行うことになる。システム保守が機会となって、技術者は、ユーザの業務の知識を蓄積し、それらを仕様・設計に転換し、その技能が形成される。そしてその技術者が、OJTトレーナあるいはサブリーダーとなり、配下の技術者を持つようになると、自身が、これまで前任技術者から受けたことを、新人技術者あるいは部下の技術者に伝える。1990年代末には、以上のことが繰り返され、技能が形成されていた¹⁸⁾。

V 2020年代初頭のIoTシステム開発プロジェクトの事例¹⁹⁾

1 IoTシステムのプロジェクト編成と開発工程

次に、2020年代初頭のIoTシステムの開発におけるソフトウェア技術者の技能形成をみることにしよう。このIoTシステムは、P社によって、開発、運用管理されている²⁰⁾。サーバレスと呼ばれるクラウド・コンピューティング・サービス、Webページの移動を行わずに画面遷移するシングルページ・アプリケーション、ならびにプログラミング言語にはPythonといった新しい技術を用いたシステムである。

それを担っているプロジェクトは、2020年春の時点で、P社が10名、協力会社が3名、合計13名の技術者から編成されている。このうち協力会社の1名が2019年にプロジェクトに配属された新人技術者である。

このIoTシステムの開発工程は、図2のとおりである。企画と開発の2つの工程に分かれる。それらの工程における作業内容は、次のとおりである。まず企画の工程は経営判断によって集められたP社の技術者が中心となり、事業の領域の探索、技術と開発手法の選択、基本設計、ならびに企画書の作成などの作業を行った。

事業の企画書が承認されると、開発の工程が始まる。開発の工程には、開発（基本設計の詳細化、プログラミング・単体テスト、ならびにシステム・

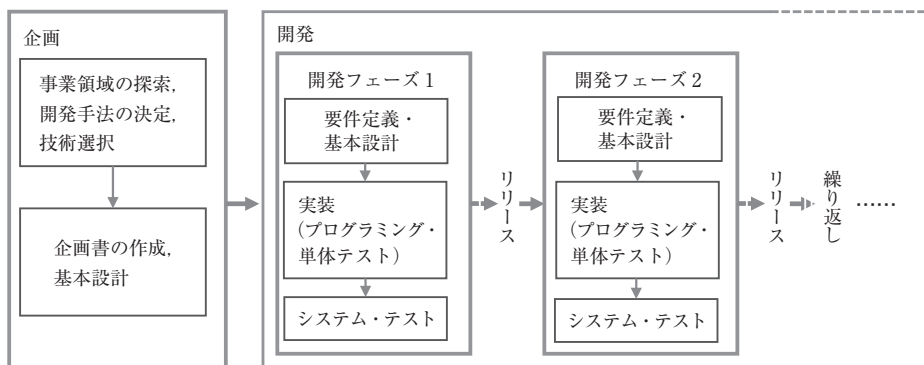
テスト）、図2には示していないが、プラットフォームの運用管理、開発管理、ならびに営業支援の業務がある。営業支援と開発管理の業務に関しては、プロジェクト設立時のP社の技術者が中心に担当し、その後参加したP社と協力会社の技術者がそれらの業務から発生する付帯業務を担っている。開発と運用管理の業務に関しては、P社と協力会社の技術者が協働で担っている。とくに、開発チームの技術者たちは、3カ月を1つの開発サイクルとして、開発の業務を繰り返し行っている。

2 IoTシステム開発プロジェクトにおける新人技術者の技能形成

このP社のプロジェクトでは、技能形成の機会を自社の技術者にとどまらず、協力会社の技術者にも、提供していた²¹⁾。以下では、2019年にプロジェクトに配属された協力会社の新人技術者を対象として、その技能形成を明らかにしよう。

まず、協力会社のリーダ役の技術者が、新人技術者の配属が決まると、P社の技術者と相談して、その教育・訓練の計画を立てた。その内容は、次のとおりである。期間は、おおよそ6-8カ月間（開発サイクルにして2,3回）とした。この期間中は、協力会社のリーダの指導のもとで、リスクが伴わない周辺の作業から始まり、クライアント側の表示系のプログラムを組み、やがてサーバ側のプログラムを組むというものであった。具体的には、まず開発サイクルのシステム・テストに参

図2 P社IoTシステムの開発工程の編成



*経営会議で承認後、開発工程が始まる

出所：労働政策研究・研修機構（2022：86）の図3-4-1より。

加させた。この作業を通じて、開発するシステムの全体像を、新人技術者に理解させることが目的であった。そして開発作業で用いるツールをExcelのマクロで組ませた。ここまでの期間は3-4カ月程度であった。次に、IoTシステムの簡単な表示系のプログラムを担当させた。クライアント側の表示系プログラムについて、1つの開発サイクルのすべての作業を行うように指導した。その過程で、リーダから指示を受けて、分からないことがあれば、相談しながら、開発した。そして徐々に難しい作業を行わせた。サーバ側のプログラムの開発作業を担うように指導した。そこでも表示系と同様に、リーダの指導の下、相談しながら、作業を遂行した。さらに新人技術者には、サーバの運用管理の業務を割り当てた。具体的には、リーダは、サーバの稼働ログにエラーが記録された場合に、そのエラーがIoTシステムのどのプログラムを実行している時に生じたものなのかを、新人技術者に調査させた。こうした作業を通じて、新しい技術と開発しているシステムに関する知識を取得させた。

新人技術者は、以上の教育・訓練を経て、開発サイクルを担うことになった²²⁾。

3 IoTシステム開発プロジェクトにおけるユーザの業務に関する知識の蓄積

上に述べた新しい技術とIoTシステムに関する知識と経験を蓄積する以外にも、競合他社などの製品・サービスなどの知識を収集している²³⁾。そしてメガバンクの事例と同様に、ユーザの業務に関する知識も蓄積している。IoTシステムの場合であれば、たとえば、ボイラーなどの設備の利用方法を知り、理解することになる。そうした知識が乏しいままでは、開発している機能の基本設計の情報を正確に把握することができないことにつながる。そして、そのことは不十分な情報と理解のもとで、基本設計あるいは詳細設計を作成し、そしてプログラムを組むことになってしまう。それは、ユーザにとって、使い勝手の悪い機能となる、あるいはその開発サイクルの実装作業が無駄になってしまうことにもなる。ユーザの業務を理解したうえで、基本設計と詳細設計を作

成し、プログラムを組む必要がある。

これらの知識の取得とそれらを元にして基本・詳細設計を作成する技能は、IoTシステムの開発サイクルを続けていく中で形成される。たとえば、協力会社のリーダは、IoTシステムの機能追加あるいは修正の時に要件定義と基本設計などの文書中に、ボイラーなどに関する用語が頻出していることに気づいた。それらのほとんどは、初見の用語であった。そこでリーダは、まずはGoogleなどを用いてWebサイトの情報を検索し、それらの情報に目をとおして、理解に努めた。それでも分からない場合には、プロジェクトの先任技術者たちに質問した。必要であれば、関連する文献を探し、これを参照した。こうして、開発しながら、IoTシステムがユーザの業務において、どの部分に用いられて、どのように利用されているのかを理解し、そうしてユーザ業務に関する知識を蓄積し、それらを仕様・設計などに転換する技能が形成されているのである。

VI おわりに

1990年代末のメガバンクのシステム保守および2020年代初頭のIoTシステムのプロジェクトを取りあげて、ソフトウェア技術者の技能とその形成過程について明らかにした。それらの事例を整理し、先行研究と比較し、検討しよう。

1 下流工程と上流工程の技能形成

まず本稿で明らかにした事例から、ソフトウェア開発の技術者の技能は、さしあたり2点に整理できるであろう。すなわち、ソフトウェア開発の基礎的技能、およびユーザの業務を仕様・設計などに転換する技能である。

ソフトウェア開発の基礎的技能とは、プログラミングとその前後を含む工程の作業を遂行するものである。取りあげた2つの事例を整理すれば、それらの作業内容は、仕様・設計書などのドキュメント管理、開発ツールの作成、ならびに仕様・設計書を読み、動作するプログラムを組むことである。これは、東京都立労働研究所(1987)と雇用職業総合研究所(1987)の下流工程を遂行する

技能であり、小池（1991）の「基礎専門レベル」の技能に相当するであろう²⁴⁾。

ユーザの業務を仕様・設計などに転換する技能については、蓄積されたユーザの業務知識を参照して、システムに対する要求を基本・詳細設計などへと転換するものである。仕様・設計を理解あるいは作成するためには、その元となる情報であるユーザの業務に関する知識が必要である。この知識に乏しければ、システムに対する要求を、要件定義および基本・詳細設計として適切に表現することができず、それらを元にしてプログラムを組む技術者に正しく設計情報を伝達することが困難となるであろう。その結果、納期までに間に合わず、不具合の多いあるいは使い勝手の悪いソフトウェアとなってしまう、改修あるいは破棄を余儀なくされてしまう。このような事態を防ぐものである。具体的には、メガバンクの事例では、システム保守の時に前任技術者の助言などの形で得ていた。そしてIoTシステムの事例では開発サイクルの中で、担当する機能の基本設計の中で初見の用語などが出現した時に、Webサイトの検索、前任技術者などへの質問、ならびに関連する文献から知識を得ていた。そしてそれらの情報を元にして、基本設計を行っていた。以上は、東京都立労働研究所（1987）と雇用職業総合研究所（1987）の上流工程を遂行する技能、そして小池（1991）の「業種なり企業の仕事の流れの分析力」と「折衝力」に相当するものであろう。そして、この技能は、ユーザの要求をソフトウェアとして具現化すること、および開発計画の立案と管理にかかわると考えられるから、基礎的技能と比較すれば、より重要なものと考えられる。

次に、技能形成については、ひとまず、ソフトウェア技術者は、易しい作業から難しい作業へと経験を積むことで、技能が形成されていると言えるであろう。2つの事例を整理すれば、入社後およそ1,2年間で、新人技術者は、前任技術者の指導のもとで、開発スケジュールに影響が小さい下流工程の作業を中心に行う。プログラミングと詳細設計書などのドキュメントの作成・修正などの作業を経験する。ここまでは、ソフトウェア開発の基礎的技能の形成にあたる。そしてそれらの作

業に習熟すると、前任技術者に助言をもらいながら、より上流工程である基本・詳細設計および要件定義の作業を経験する。ユーザの業務に関する知識を蓄積し、システムに対する要求をとりまとめ、基本・詳細設計書を作成あるいは改訂・修正し、プログラムを組む。やがて、OJTトレーナになる、あるいは部下の技術者を持つようになる、新人・配下の技術者に指導・助言し、自身が割り当てられた下流工程の作業については独力で、より上流工程の作業については、前任技術者から助言をもらい行うことになる。これらが繰り返されることで、やがて、上流工程の作業についても、独りで行うことになる。これらは、仕様・設計へ転換する技能の形成にあたる。

2つの事例から安易に一般化することは慎まなければならないが、ソフトウェア開発の基礎的技能の形成については、その期間が短くなり、指導・助言をもらいながら基本・詳細設計の作業を担う時機が早期化していると考えられる。この時機は、メガバンクの事例では2年目以降、IoTシステムの事例ではプロジェクト配属後半年たらずであった。おそらく、この早期化については、ソフトウェア開発支援ツールなどの進歩により、労力が軽減したこと、そしてそれによってOJTの内容と方法が定型化されたことによるものではないかと思われる。基礎的技能の形成については1980年代よりも、定型化が進み、期間が短くなり、より上流工程の仕事を担う時機の早期化が進んでいるものと考えられる。

そして仕様・設計へ転換する技能については、2つの事例はほぼ同じと考えて良いであろう。メガバンクの事例のところで指摘したように、「理想的な条件」が整えば、東京都立労働研究所（1987）などが指摘するとおりの、下流から上流工程へという形で技能形成が進む。しかし、仕様・設計へ転換する技能の形成については、ソフトウェア開発の基礎的技能の形成と比較すれば、定型化されていないと考えられる。

2 規模と複雑さ、およびプロジェクト編成に応じた技能形成

2つの事例を一般化するのは慎重でなければな

らないが、年あるいは四半期などの期間でみたとき、ソフトウェア開発・保守のプロジェクトには、さまざまな大きさと複雑さの開発と保守があり、さまざまなプロジェクト編成の状況がある。技術者の技能形成の段階に応じた大きさ、複雑さのソフトウェア開発があるとは限らない。たとえば、煩雑さをさけるために、プロジェクトの編成状況は除いて、規模と複雑さに応じて、技能形成が、どのように進むのかを考えてみよう。メガバンクの事例では、マネージャのT氏は、2年目には、コーディング作業が減り、機能設計と技術設計の修正を、3年目以降になると100人月程度のシステム保守を、前任技術者の助言を受けながら、すべての工程を担当していた。IoTの事例では、新人技術者は、リーダーの指導のもとで、簡単なプログラムについて基本設計の詳細化、プログラミング・単体テスト、テストという1つの開発サイクルのすべての作業を行っていた。この点から、規模が小さく、かつ簡素なシステム開発・保守が継続して存在すれば、早い時期に、基本・詳細設計などの上流工程の一部の作業を担っていると言えるであろう。この場合には、基礎的技能的習熟度は低いが、広い範囲の工程の作業を経験し、繰り返すことで、技能形成が進む可能性がある。それに対して、規模が大きく、かつ複雑なシステム開発・保守が継続してあれば、前者と比較すれば、基礎的技能的習熟度は高いが、狭い範囲の工程の作業を経験し、繰り返すことで、技能が形成される可能性がある。

この技能形成については、OJTを通じて形成されるにしても、その内容、方法、ならびに期間などは、ソフトウェア開発の基礎的技能和比較すれば、定型化されたものではないと考えられる。

つまり、仕様・設計へ転換する技能については、規模と複雑さ、そしてプロジェクトの編成に規定されて、形成が進むと考えられる。東京都立労働研究所(1987)、雇用職業総合研究所(1987)、ならびに小池(1991)では、下流工程から上流工程へ、あるいは易しい仕事から難しい仕事へと技能が形成されると単線的に描いていた。しかし、そのような単線的なパターンではないと思われる。OJTによって技能形成される点は同じでは

あるが、開発の規模と複雑度、およびプロジェクト編成によって、作業の割り当てが変わってくる。その結果として、OJTの内容と期間も変わり、技能の習熟度と指導・助言を受けずに担える工程の範囲が異なると考えられる。小池(2005)に即して言えば、技能の深さと幅が異なってくると考えられる。

さらに、規模が大きく、かつ複雑なソフトウェア開発であり、プロジェクトの技術者の役割に変化がない場合には、仕様・設計へ転換する技能形成が進まず、そして職能資格制度が実年齢に重きをおいて運用されていれば、技能がともなわないマネージャ、あるいはマネージャになろうとしない技術者を多数生み出してしまっていることも考えられる。技術者の機会主義的行動に手を拱き、技能形成に問題が生じている可能性もある。この点については、本稿の目的を超えているので、言及するにとどめたい。

以上の考察は、限られた事例から導き出したものである。これらをソフトウェア技術者全般の傾向とするのは、性急である。本論でも触れたように、ソフトウェア技術者の技能形成過程が1つの企業内にとどまっていないことを念頭におけば、IT産業の分業構造を把握し、その中に事例を位置づけたうえで、より一層の丁寧な考察が求められる。

3 DX時代の含意

ソフトウェア技術者の技能形成が上のようなものであるとするならば、DXが進展している状況において、どのような含意があるのだろうか。新しい技術の登場により、それまでの技能が陳腐化し、そのもとでの技能形成を図らなければならない、という議論がある。再教育訓練あるいはリスキリングである。さまざまな新しい技術が存在するが、たとえば、新しいプログラミング言語であれば、文法などの学び直しの期間が必要である。しかし、本稿で明らかにしたように、プログラミングなどの技能形成は定型化されていた。この点から、他のプログラミング言語によるソフトウェア開発の経験があれば、新しいプログラミング言語への習熟期間は短いと考えられる²⁵⁾。

それでは、最後に、他の産業と他の職種における労働者の技能形成については、どうなるのであろうか。DX と呼ばれる革新によって、新しい製品・サービスが生まれ、新しい仕事が生まれている。そしてその仕事には、プログラミング、統計、AI などの知識と経験が求められる、などと議論されることがある。おそらくは新しい仕事には新しい技術を利用してソフトウェアを開発し、それを運用することになるであろう。そのためには、ユーザの業務を明らかにし、仕様・設計書を書き、プログラムを組むことになる。それらの作業においては、要求を仕様・設計へ転換する技能が重要なものであることを指摘した。

そこからの類推となるが、プログラミング、統計、AI などの知識も必要であろうが、まずは、ソフトウェアとして実現したい業務に関する情報を、技術者に適切に伝えなければならない。技術者と折衝することが求められる。または、システム化の意図を汲み取り、それをシステムとして具現化する実績がある企業を選定する、さらには業務で利用できるパッケージ・ソフトウェアを探し出す力ということになるのであろう。

これらを担えるのは、OJT を通じて得られた業務に関する知識と経験を蓄積した労働者であろう。さらには、AI が適切に動作するためには、データを適切に蓄積し、適切に学習させる必要がある。それらを担えるのは、知識と経験を蓄積した労働者であろう。そのような労働者を確保するためには、新しい技術に関する知識の習得を促すための諸制度を整備し、企業内外の研修プログラムを活用することになるであろう。そしてさらに、より重要なことは、企業ないし経営者が新しい技術のもとでの OJT の機会を用意することと思われる。

- 1) IoT システムとは、工場における設備などにネットワーク接続可能なセンサーを取り付け、その稼働状態に関するデータを収集・蓄積し、それらの中からみたいデータを取り出し、可視化し、分析し、設備の効率化に寄与するというものである。
- 2) 経済産業省 (2018) を参照。また DX という用語は、Stolterman and Fors (2004) が初出だと言われている。
- 3) ソフトウェア開発において、プロジェクトとは作業組織の単位であり、受注案件毎、製品毎、あるいはサービス毎に数名から数十名の技術者から編成されている。多くのソフトウェア企業では、受注元の業種毎にこのプロジェクトをいくつか集め、

課あるいは部が編成されている。開発・保守するソフトウェアの規模が大きくなれば、プロジェクトは複数のチームと呼ばれる組織から編成されるようになる。そして、予算・人事・納期に関する権限を持つプロジェクト・マネージャ、開発するソフトウェアの設計などに関する権限を持つプロジェクト・リーダー、リーダーから権限を委譲されたサブリーダー、そしてメンバーから編成されている。

- 4) デバッグ作業の詳細については、パッケージ・ソフトウェア開発の事例ではあるが、古谷 (2007: 55-58) を参照。
- 5) ソフトウェアの開発工程については、有澤 (1988: 10-15)、玉井 (2004: 13-18)、古谷 (2008: 124-125) を参照した。
- 6) 東京都立労働研究所 (1987: 26-33, 35-37) より。
- 7) 雇用職業総合研究所 (1987: 29-34) より。
- 8) 小池 (1991: 93-95) および小池 (2015: 78-85) より。
- 9) 戸塚・中村・梅澤 (1990: 127-129, 142) より。
- 10) Williamson (1975=1980: 129-130)、および小池 (2005: 156) を参照。
- 11) 管見の限りではあるが、ソフトウェア技術者のキャリア志向に焦点をあてた三輪 (2001) およびイノベーション・パターンに関する研究である生稲 (2012) などがある。これらの研究は、興味深く、学ぶべき点もあるが、ソフトウェア開発の技能形成には焦点をあてて、考察したものではない。
- 12) 本事例は、とくに断りのない限り、TSB 社 [2] にもとづいている。
- 13) TSB 社は、1999 年時点で、売上高 160 億円、従業員数 1172 名のソフトウェアの受託開発とネットワーク管理などを行う企業である。調査を行った 1999 年は、インターネット・ブームや IT バブルなどもあり、バブル崩壊後による売上高減少から、回復が始まった時期にあたる。
- 14) T 氏は、入社 20 年ほどのベテラン技術者である。入社以来、1980 年代後半の一時期を除いて、金融系システムの開発・保守業務に一貫してたずさわっていた。
- 15) TSB 社のメガバンクの開発・保守を担当している部門の技術者の間では、おおよそ 1 万人月以上を大規模、そして 100 人月以上を中規模との認識が共有されていた。TSB 社 [2] および [4] より。
- 16) 1990 年代末における TSB 社の新人技術者のプロジェクトの配属時期は、理工系・情報系学部卒業であれば夏頃、それ以外の学部卒業であれば秋頃であった。
- 17) TSB 社 [2] および TSB 社 [4] より。
- 18) 本論中では、OJT を中心にとりあげたが、Off-JT についても活発に行われている。たとえば、メガバンクの新人行員向けの銀行業務に関する研修会などに参加している。さらに 1999 年度であれば、A プロジェクトが属する部門長が中心となり、信託系の銀行業務に関する研修、そして新しい金融商品および制度の変更点について、ユーザから聴き取り、それをもとにして勉強会などを開催した。それ以外にも、プロジェクト内で銀行業務の用語集を作成し、それを利用することで知識を取得している。TSB 社 [2] および TSB 社 [4] より。
- 19) 本事例については、労働政策研究・研修機構 (2022) の聞き取り調査にもとづいている。
- 20) P 社は、エネルギー関連企業によって 1980 年代前半に設立された。売上高が約 400 億円、従業員数が約 1000 名の企業である。取引先は、親会社からの受注にとどまらず、広く民間企業から、システム開発およびネットワーク管理の業務を受注している。
- 21) この背景については、労働政策研究・研修機構 (2022: 90, 92-96) を参照。なお、TSB 社のメガバンクの事例においても、別のプロジェクトで、類似の事例があった。2000 年 2 月時点で、メガバンクからの強い要請によって、メガバンクとその子

会社の技術者が、TSB社のプロジェクトのチームに参加し、経験年数10年以上のベテラン技術者のもとで、システム保守の業務を行いながら、銀行業務と銀行システムに関する知識の取得に努めていた。TSB社の技術者が、OJTトレーナとなり、他社の新人技術者にたいして教育訓練を行っていた。TSB社[4]より。このような企業を超えた技能形成については、戸塚・中村・梅澤(1990:131-133)および自動車産業の事例になるが河野(2009)でも、指摘されている。

- 22) IoTシステムのプロジェクトにおいては、ソフトウェア開発の経験がある技術者であっても、新しく参加した場合には、1カ月ほどの教育・訓練の期間を設けている。その内容は、次のとおりである。はじめに、新たに配属された技術者は、ある機能のプログラムのソースコード、基本設計などの資料、ならびにサーバ側システムについて解説した動画などの教材が渡され、それをもとに自習する。つぎに、自習で得た知識を元に、プログラムを組むことになる。その時、分からないことがあれば、前任技術者に質問することになる。そしてこの期間が終わると、IoTシステムのソフトウェアを開発することになる。
- 23) 研修・セミナー・展示会などに参加して、競合他社のIoTシステムの情報収集についても収集に努めている。労働政策研究・研修機構(2022:93)より。
- 24) IoTの事例では、開発サイクルを担う技術者全員が、基本設計から単体テストまでの工程を1人の技術者が担当している。それにたいして、メガバンクのAプロジェクトの事例でも、詳細設計から単体テストまでの工程が1つの工程として扱われるようになっており、そして規模が小さく、簡素な開発・保守の案件であれば、経験年数が短い技術者でも、機能設計の修正程度であれば、担っていた。2つの事例からは、下流工程が統合される方向にあり、技術者に、より上流工程の作業を任せようとしていると解釈できる。しかし、メガバンクの事例では、仕様・設計に転換した経験豊富な技術者は、コーディング前後の工程の作業にほとんど携わっていなかった。おそらく、この違いは受注あるいはパッケージ・ソフトウェアの収益構造などに起因しているのではないかと思われる。また、小池(2005)の「統合方式と分離方式」に関する論文につながると思われるが、本論文では、言及することと定める。
- 25) 本稿では考察することができなかったが、開発管理にも影響を与える可能性がある。サーバレスの技術を用いると、ソフトウェア開発支援環境が大きく変化してしまう。このことから、工数の見積と実績に大きな乖離が生じる可能性を指摘できる。その結果、機能が不十分で、品質が低いソフトウェアが出荷される、あるいは出荷時期の延期が繰り返される可能性がある。この点については、1990年代半ばのパッケージ・ソフトウェア開発の事例ではあるが、古谷(2007:86-92)を参照。

参考文献

- 有沢誠(1988)『ソフトウェア工学』岩波書店。
 生稲史彦(2012)『開発生産性のディレンマ——デジタル化時代のイノベーション』有斐閣。
 経済産業省(2018)「DXレポート——ITシステム「2025年の崖」の克服とDXの本格的な展開」。https://www.meti.go.jp/shingikai/mono_info_service/digital_transformation/2018_0907_report.html(2022年12月7日参照)
 小池和男(1991)「営業分野の人材開発」小池和男編『大卒ホワイトカラーの人材開発』東洋経済新報社、pp.61-102。

- (2005)『仕事の経済学 第3版』東洋経済新報社。
 ——(2015)『なぜ日本企業は強みを捨てるのか——長期の競争 vs. 短期の競争』日本経済新聞社。
 河野英子(2009)『ゲストエンジニア——企業間ネットワーク・人材形成・組織能力の連鎖』白桃書房。
 雇用職業総合研究所(1987)『情報処理技術者の能力開発とキャリア形成——ソフトウェア技術者を中心に』。
 東京都立労働研究所(1987)『技術革新下における労働者の生活と健康(その1)——ソフトウェア技術者を中心に』。
 戸塚秀夫・中村圭介・梅澤隆(1990)『日本のソフトウェア産業——経営と技術者』東京大学出版会。
 玉井哲雄(2004)『ソフトウェア工学の基礎』岩波書店。
 三輪卓己(2001)『ソフトウェア技術者のキャリア・デベロップメント——成長プロセスの学習と行動』中央経済社。
 古谷真介(2007)『パソコン・ブーム下におけるパッケージ・ソフトウェア開発の作業組織研究——K3社における開発過程、開発管理、人事管理制度』東京大学社会科学研究所研究シリーズ No. 23, 東京大学社会科学研究所。
 ——(2008)「中国における日本向けソフトウェア開発における管理体制の類型——設計情報伝達の仕組み」田島俊雄・古谷真介編著『中国のソフトウェア産業とオフショア開発・人材派遣・職業教育』現代中国研究拠点 研究シリーズ No. 2 東京大学社会科学研究所 pp.39-76。
 労働政策研究・研修機構(2022)「第3章 IoT サービス事業における外部人材活用」『「サービス化」の下での人材マネジメント——企業ヒアリング調査から』JILPT 資料シリーズ No. 257。
 Stolterman, Erik and Anna Croon Fors(2004)“Information Technology and the Good Life.” *Information Systems Research*, pp.687-692。
 Williamson, Oliver E.(1975) *Markets and Hierarchies*, Free Press.(=1980, 浅沼万里・岩崎見沢(1980)『市場と企業組織』日本評論社)

聞き取り一覧

- TSB社[1]1999年10月27日, TSB社O事業所, K氏, T氏, H氏からの聞き取り:プロジェクト編成, 保守プロジェクトにおける開発工程の編成とその作業内容。
 TSB社[2]2000年2月8日, TSB社O事業所, K氏からの聞き取り:部門における技術者の技能形成。マネージャになるまでの職歴, 人事評価制度の運用。
 TSB社[3]2000年2月9日, TSB社O事業所, T氏からの聞き取り:開発・保守しているシステムの特徴とその作業内容, プロジェクトにおける技術者の技能形成。マネージャになるまでの職歴, 人事評価制度の運用。
 TSB社[4]2000年2月23日, TSB社O事業所, H氏からの聞き取り:開発・保守しているシステムの特徴とその作業内容, プロジェクトにおける技術者の技能形成。マネージャになるまでの職歴, 人事評価制度の運用。

ふるや・しんすけ 大阪産業大学経済学部経済学科准教授。最近の主な論文に「IoT サービス事業における外部人材活用」『「サービス化」の下での人材マネジメント——企業ヒアリング調査から』JILPT 資料シリーズ No. 257 (西村純氏との共著, 2022年)。人的資源管理論, 労働経済論専攻。