

# 企業の新入社員研修におけるプログラミング教育の実践例

塩崎正人<sup>†1</sup> 敦賀誠一<sup>†1</sup> 中川宙<sup>†1</sup> 古谷芳康<sup>†1</sup> 吉田典弘<sup>†2</sup>

**概要:** 本報告では、情報科学あるいは情報工学を専門としない学科の学生が IT 企業へ就職し、新入社員研修を実施したことで得られた知見から、産学が連携したプログラミング教育の在り方を提案する。特に、プログラミング言語に依存しないプログラミング教育を大学の授業で実践してもらうことで、企業での新入社員教育がやりやすくなることと、このような素養を持たせることで、これからの時代におけるイノベーション（技術革新）に対応できる人材育成ができることを提案する。また、フルタニ産業株式会社では、独自のカリキュラムで新入社員研修を実施してきたが、この研修を通して判明したことがいくつかある。まず、入社時点でのプログラミング経験の有無は、新入社員研修が始まれば能力の差としてはほとんど表れてこない。この原因は学生の学習に対する意識に問題があると考えられる。また、多くの学生は大学での情報教育に関する授業で、プログラミングに関して十分に学び切れていないことも要因と思われる。このようなことに対応するには、学生は低学年の段階から社会人体験をすることが重要であろう。これにより学生自身の学習意識が変わり、その後の大学における学び方に良い影響を与えると思われる。近年、大学の動向として、インターンシップ（就業前企業実習）に学生を参加させることや、大学の授業に企業からゲストスピーカーを招いて就職や仕事に関する話を学生に聞ける機会を多くなってきている。よって、プログラミング教育に関しても、このような産学連携ができるように、可能な限り多くの学ぶ機会を提供し、IT 業界を志望する学生のプログラミングスキルを向上させたいと考える。

**キーワード:** 新入社員研修, プログラミング教育, インターンシップ

## A Practice Example of Programming Education in New Employee Training of an Enterprise

MASATO SHIOZAKI<sup>†1</sup> SEIICHI TSURUGA<sup>†1</sup> HIROSHI NAKAGAWA<sup>†1</sup>  
YOSHIYASU FURUTANI<sup>†1</sup> NORIHIRO YOSHIDA<sup>†2</sup>

**Abstract:** The student of a department outside of information sciences and information engineering proposes improving programming education via industry and academia cooperating with an IT enterprise so that new employee training can be better implemented. It is further proposed that programming education outside of university sessions conducted by enterprises for new employee would facilitate lasting employee improvements and facilitate technical innovation. Further, it would be beneficial to implement new employee training using an original curriculum by a FURUTANI industrial corporation. This training has revealed several things. When new employees first begin training, those with no programming experience upon entry show improvement in their programming abilities. The cause may be a problem with awareness of student education. It may be a significant factor that many students cannot learn programming sufficiently via university sessions. It will be important for students to gain experience as a member of society from the early stages of lower grades. Students appear to improve how much they are able to learn at the university level when provided with greater early education. A trend in recent years that has encouraged students to participate in internships (enterprise training before the commencement of work) is the presence of guest speakers from enterprises in the university classroom discussing how to obtain employment. As far as it is possible, I would like to improve the programming skills of students who want to learn and wish to join the IT industry. Therefore, such industry-academia cooperation is being facilitated for programming education.

**Keywords:** New Employee Training of an Enterprise, Programming Education., Internship

### 1. はじめに

『IT 人材の最新動向と将来推計に関する調査結果（報告書概要版）』商務情報政策局情報処理振興課（平成 28 年 6 月 10 日）[1]によると、「大型の IT 関連投資の継続や情報セキュリティ等のニーズ増大により、IT 人材の不足が改め

て課題となっている。しかし、日本の人口減少に伴い労働人口（特に若年者）が減少することから、IT 需要の拡大にもかかわらず、国内の人材供給力が低下、IT 人材不足は今後より一層深刻化する可能性が高くなる。更に IT 産業は日本の経済成長を担う役割を期待されている」と述べられている。このような状況を踏まえると IT 人材の教育・育成がいかに重要な課題であるかが浮き彫りにされてくる。

また、文部科学省では 2020 年度からの小学校プログラミング教育の全面実施に向けて、平成 30 年 3 月に『小学校プログラミング教育の手引（第一版）』を作成した[2]。そ

<sup>†1</sup> フルタニ産業株式会社  
Furutani Sangyou Co., Ltd.

<sup>†2</sup> 羽衣国際大学  
HAGOROMO University of International Studies.

の中で「今日、コンピュータは人々の生活の様々な場面で活用されている。家電や自動車をはじめ身近なものも多くにもコンピュータが内蔵され、人々の生活を便利で豊かなものになっているのである。誰にとっても、職業生活をはじめ、学校での学習や生涯学習、家庭生活や余暇生活など、あらゆる活動において、コンピュータなどの情報機器やサービスとそれによってもたらされる情報とを適切に選択・活用して問題を解決していくことが不可欠な社会が到来しつつある」と記載されている。

そのような社会情勢の中、ひとりでも多くの良い IT エンジニアを育成したい。その思いは産学共通のものであると筆者は信じている。

本報告では、IT エンジニアの育成のためにフルタニ産業株式会社に取り組んでいる新入社員研修の実践例を紹介し、産学協力の元、より良い IT エンジニアをより多く輩出するための一助となる提案を行う。

## 2. フルタニ産業株式会社における新入社員研修について

フルタニ産業株式会社では、教育専門の部門を設立し、社内外の新入社員に対して独自のカリキュラムで研修を実施している。

本章ではフルタニ産業株式会社が実施してきた独自の教育カリキュラムについて紹介する。

### 2.1 背景

フルタニ産業株式会社では 2014 年からある程度まとまった人数（10 名前後）の新卒採用を実施している。それ以前も新卒を採用していたが、年度によって採用者がゼロの場合もあり、安定した採用活動は行われていなかった。

当時は新入社員研修として、新入社員にプログラミング関連の書籍を貸与し、基本的に自学自習で進めてもらっていた。その中でわからない部分を近くにいる先輩社員に聞くという研修とは名ばかりの放置状態であった。また、研修期間は人それぞれで、配置先が決定した新入社員から順に研修を切り上げ、実際の仕事に配置されていった。20 年前はこれが当たり前の時代であったし、今でも可能な限り早く実際の仕事に携わることが成長への近道であると考えている。そのためにも、必要な学習は自分で行い、自分に時間とお金をどれだけ投資できるかが重要になってくる。しかし、そのような考えは今となっては一昔前の古い考え方になっているのかもしれない。

2015 年の新入社員研修では、外部の研修機関を利用することを試みた。在学中にプログラミングを多少なりとも学んできたプログラミング経験者と全くの未経験者に分けて、プログラミング経験者に関しては配置先決定までの期間に自学自習でプログラミングを学習させ、配置先が決定した

者から順に研修を切り上げ、実際の仕事に配置されていった。これは前年までと同じやり方である。一方、プログラミング未経験の新入社員に対しては 1 ヶ月半の期間をかけて、外部研修を利用してプログラミングについて学ばせた。しかし、実際に外部研修を終えて戻ってきた新入社員の成長度合いを確認すると、仕事で使えるレベルには遠く及ばなかった。研修で使用したテキストに記載のある内容についてはプログラムを作成することが可能だが、そこからの応用が全く出来なかった。外部研修で実施したカリキュラムは、Java, Servlet, JSP を使った Web アプリケーションの開発であったが、Web アプリケーションの仕組みすら理解できていない者がほとんどであった。

2016 年の新入社員研修では、かけたコストの割に効果が全く感じられない外部研修の利用を中止し、プログラミングに関する書籍を与え自学自習のやり方を続けた。出来の良い者は客先の配置となり、客先で決まらなかった者は社内受託案件に配置された。社内受託案件に配置された新入社員は、最初に簡単なテスト作業等を実施した。そこから、教育の意味も込めて徐々にプログラミングの作業に移行していった。しかし、実際にプログラミングの作業が始まると、まともにプログラムを書けないことが顕著になった。そのため、該当社員は一旦実作業から切り離し、プログラミング学習の時間を設けた。そこでは、当時まだエンジニアとして社内受託案件に携わっていた筆者が講師となり、1 ヶ月程度をかけてプログラミングを最初から教え直すこととなった。

ここまでの 3 年間およびそれ以前は、新入社員研修というよりは、配置先決定までの間、自学自習でプログラミングの学習をさせて、配置先での仕事が始まってから、その仕事の中で必要な事を学習させるという考えが主流であった。しかし、仕事の中で学習すると言っても、仕事場に立つだけの能力が養われていない状態では適応することが困難である。結局は配置先の現場から「新入社員研修では一体何を教えてきたのだ」という声も上がってきていた。

自学自習では限界があり、新入社員は思ったように育たない。教育として十分な成果が得られないことがわかってきた。新入社員教育はフルタニ産業株式会社としても大きな課題となっていた。

そこで、フルタニ産業株式会社は本格的に教育に力を入れるべく、2017 年に教育専門の部門、人財育成推進部門を立ち上げ、選任講師を置いて新入社員教育を実施することとした。

### 2.2 人財育成推進部門の新入社員研修

人財育成推進部門が新入社員研修を実施するに当たり、育成目標として掲げたのが、新入社員が「現場で邪魔者扱いされない最低限の能力を身に付けること」、および「資質向上の土台を作ること」であった。

仕事を学ぶためには実際に仕事をするのが一番の近道であるという考えに変わりはない。よって、実際の仕事の現場で「仕事」を学んでいくために、OJTの中で先輩の足手まといとなり、経験できる仕事が少なくなならないよう、最低限「ジャマ！」と言われなだけの能力を新入社員研修で身に付けさせる必要があった。そのためにはプログラミングができるだけでは不十分である。

これまでフルタニ産業株式会社ではプログラミング言語の学習を新入社員研修の中心に位置付けていた。しかし、ただ「プログラミング」ができるように育成するだけではなく、システム開発の「仕事」ができるように育成する必要があった。それだけのことを学習するためには1ヶ月程度の研修期間では不十分であった。そこで、まずは研修期間を3ヶ月間とし、専任の講師を用意し、研修を途中で切り上げて実際の仕事に配置することも禁止した。

また、研修中はどのように学習すればよいかを講師が示してくれるが、研修終了後は仕事の中で、自分自身で学び続けていく必要が出てくる。学習する目的を設定し、学習する方法を考え、実行に移していかななくてはならない。少し前まで学生であった新入社員には、自ら考え、自ら動くということは、筆者が考える以上に難しいことであるように見える。それができるようになるために研修の中では基本的に自分たちから発信しないと何も得ることができない環境を作り出している。一方的に講師が講義をするような形式はとらない。自身で学習を進め、わからないところが出てきた場合、自分から質問するように仕向ける。質問してきたことに対しては相手が理解するまで何度でも説明する。中にはわからないことが増えてくると「全部がわからない」という錯覚に陥る者が少なくない。「全部がわからない」という者は「何がわからないかがわからない」状態にあるため、どこまでわかっている、本当にわからないことは何であるのかを一緒に確認していく。「何がわからないかがわからない」という者は、その時点で自ら考えることを放棄してしまっている。そんな状態では自分自身でいくことなど不可能である。残念ながら、これらの「全部がわからない」や「何がわからないかがわからない」というセリフを研修の初期段階でよく聞く。できる限り早い段階でその思考から脱却させる必要がある。それが出来れば、どんな問題に遭遇しても途中で諦めることなく、解決方法を模索することができるであろう。それが学びにつながっていく。

### 2.3 フルタニ産業カリキュラム 2017

人材育成推進部門が実施する新入社員研修では、システム開発の仕事遂行のための力を身に付けさせるため、研修の前半でプログラミング言語研修を実施し、後半ではチーム演習で模擬システム開発を実施する。

まず、プログラミング言語研修として、開発ツールであ

る Visual Studio を使用し、C#の学習を進める。プログラミング言語として C#を選択したのは、当時フルタニ産業株式会社が扱っている仕事として C#の需要が増えてきていたためである。その後、模擬システム開発に必要な基本的なデータベース・SQLについて学習する。

その後、チーム演習として模擬システム開発を実施する。3~4名でチームを構成し、チームリーダーを決め、チームメンバーにも役割を持たせて、チームとして最大の力が発揮できるように工夫して進めてもらう。このチーム演習では要件定義から納品までを一通り経験する。講師を顧客に見立て、ヒアリングを実施し、要件を決めていく。開発する対象が決まれば、設計、プログラミングへと作業を進める。最後にテストを実施して、問題なければ納品という形になる。

開発する模擬システムは社員情報や経歴情報をデータベースで管理する「社員情報管理システム」である。

各チームの進捗状況を確認しながら、遅れがちなチームにはフォローを入れ、場合によっては仕様を削ることもある。逆に進んでいるチームには、機能追加や仕様変更といった実際に起こり得る問題を与えることもある。

表 1 フルタニ産業 2017 年新入社員研修カリキュラム

Table 1 New Employee Training in 2017

日程	研修内容
1 日目	研修内容の説明 開発環境構築
2~3 日目	社会人マナー研修 (外部研修)
4~9 日目	プログラミング言語研修 (C#) プログラミング基本構文
10~16 日目	プログラミング言語研修 (C#) オブジェクト指向プログラミング
17~18 日目	プログラミング言語研修 (C#) デバッグ
19~28 日目	プログラミング言語研修 (C#) GUI プログラミング
29~32 日目	データベース研修 SQL
33~35 日目	データベース研修 プログラム連携
36~37 日目	作業効率化テクニック マクロ・バッチ・正規表現・ショートカット
38~58 日目	チーム演習 (模擬システム開発) 社員管理システム
59 日目	成果発表

### 2.4 フルタニ産業カリキュラム 2018

2017年の新入社員研修を振り返ってみると、プログラミングの基本的な部分を理解していない社員が何人かいた。そんな状態でもチーム演習の課題を作成し、それなりに動くものを作成できていた。しかし、ソースコードレビューを実施した際に、自分自身が作ったプログラムをうまく説明できない者が複数名いた。原因を探っていく、ひとつの仮説に辿り着いた。

「プログラミングの基本をわかっていなくても便利なツールが補完してくれるため、プログラミングを理解した気になっているのではないか。」

この点を改善すべく、2018年の新入社員研修ではプログラミングと開発ツールの使い方を分けて学習するように研修カリキュラムを変更した。

2018年のカリキュラムでも研修の前半でプログラミング言語研修を実施し、後半でチーム演習を実施するという基本路線は2017年を踏襲している。

2017年のカリキュラムと異なる点は以下の通り。

- プログラミング学習の言語としてJavaを使用
- プログラミング学習時点で便利な開発ツールの不使用
- 途中でプログラミング言語をJavaからC#に変更
- 最後に配置先への準備、研修の総復習の時間を確保
- 人前で話をするのが苦手という者が多くいたので、1分間スピーチを実施
- チーム演習の期間の拡張

まず、プログラミング言語研修としてJavaを使用し、1ヶ月間程度で市販の書籍「スッキリわかるJava入門」[3]にて、プログラミングの考え方を学ぶ。基本的に翌日までに書籍の進めてもらいたい範囲を発表し、各自自宅等で学習してきてもらう。翌日の午前中に学習してきた内容の質問を受け付け、午後には練習問題をこなす。この際、Eclipse等の便利な開発ツールを使用しない。コンパイルには「javac」コマンドを、実行には「java」コマンドを使用する。Eclipse等の便利な開発ツールを使用しない理由は、3章で詳しく述べる。Javaを一通り学び終えた後に別のプログラミング言語C#を2,3日間で学ぶ。基本的には既にJavaでプログラミングの考え方を学んでいるので、JavaとC#との相違点を確認する程度で十分である。C#では便利な開発ツールのVisual Studioを使用して、GUIアプリケーションの作り方を学ぶ。その後、データベースの基本的な考え方と模擬システム開発に必要なSQLを学び、チーム演習で実施する模擬システム開発に入っていき準備が整う。

次にチーム演習に入るが、開発する模擬システムの内容は2017年と同じ「社員情報管理システム」である。

新入社員研修の最後には、実際の仕事の配置先に向けた準備や新入社員研修全体を通しての総復習の時間を設けている。ここは新入社員それぞれで自分に必要な内容を学習してもらう。たとえば、配置先が組込開発の仕事で、CやC++でのプログラミングが必要であれば、新入社員研修で学んだことを応用して学習していく。異なるプログラミング言語への対応は既に経験済みであるため、大きな抵抗はないはずである。

表2 フルタニ産業2018年新入社員研修カリキュラム

Table 2 New Employee Training in 2018

日程	研修内容
1日目	研修内容の説明 開発環境構築 (Java)
2~9日目	プログラミング言語研修 (Java) 書籍による学習 社会人基礎研修
10~17日目	プログラミング言語研修 (Java) 練習問題
18~19日目	プログラミング言語研修 (C#) 環境構築 Javaとの相違点を確認
20日目	プログラミング言語研修 (C#) 練習問題
21~23日目	データベース研修 環境構築 SQL プログラム連携
24~51日目	チーム演習 社員管理システム
52日目	成果発表
53~62日目	総復習 配置先に向けた準備

## 2.5 フルタニ産業カリキュラム2019

2018年のカリキュラムでは、最後のチーム演習で顧客へのヒアリング不足により、自分勝手なシステムを作り出してしまう失敗を犯し、要件を満たしたシステムを完成させることができなかった。全チームがほぼ同じ失敗を犯していたので、2019年のカリキュラムでは、チーム演習を2回実施することとした。

2019年のカリキュラムは、基本的には2018年の研修を踏襲するが、2018年と異なる点は以下の通りである。

- プログラミング言語研修で独自のテキストを使用、2018年に使用した書籍は参考図書として使用
- チーム演習を2回実施
- 1分間スピーチを廃止
- 1日に先輩社員3名と話をするノルマ
- 総復習の時間を縮小
- バージョン管理ツールの導入

まず、プログラミング言語研修としてJavaを使用しプログラミングの考え方を1ヶ月間程度で独自のテキストを使用して学ぶ。やはりここでも、Eclipse等の便利な開発ツールは使用しない。Javaを一通り学び終えたら、C#を2,3日間で学ぶ。このときは開発ツールのVisual Studioを使用して、GUIアプリケーションの作り方を学ぶ。その後、データベースの基本的な考え方と模擬システム開発に必要なSQLを学ぶ。ここは2018年のやり方と同様である。

次にチーム演習に入るが、今回は模擬システム開発を2回実施する。1回目の模擬システム開発では1週間程度で終わる簡単な開発を経験し、いろいろな失敗を犯してもらう。その失敗から学んだことを活かして2回目の模擬システム開発を実施する。同じミスを犯さなければ、プロジェ

クトが成功する確率は格段に上がる。

1 回目の課題は実施するインターンシップとその参加者を管理する「インターンシップ管理システム」で、データベースは使用せず、データはテキストファイルで管理する。

2 回目の課題は、2017 年、2018 年と同じ「社員情報管理システム」である。

研修の最後には、配置先に向けた準備や研修全体を通しての総復習の時間を設けているが、チーム演習を 2 回実施したことより、総復習の時間は 2018 年の研修より減少した。

2018 年に取り入れた 1 分間スピーチは、途中から当初の目的からずれて行っていたため、2019 年では実施しなかった。その代わりに 1 日に最低 3 名の先輩社員と話をするノルマを課した。先輩と話をすることをノルマ化することにより、新入社員にとっての大義名分、話しかけるきっかけを用意した。その甲斐あって、比較的研修の早い段階で、多くの先輩社員と関わることができていた。

表 3 フルタニ産業 2019 年新入社員研修カリキュラム  
 Table 3 New Employee Training in 2019

日程	研修内容
1 日目	研修内容の説明 開発環境構築 (Java)
2~20 日目	プログラミング言語研修 (Java) 独自テキストによる学習 社会人基礎研修
21~22 日目	プログラミング言語研修 (C#) Java との相違点を確認
23~24 日目	データベース研修 SQL プログラム連携
25~31 日目	チーム演習① インターンシップ管理システム
32 日目	チーム演習①振り返り
33~54 日目	チーム演習② 社員管理システム
55 日目	成果発表
56~60 日目	総復習 配置先に向けた準備

### 3. 新入社員研修を通してわかったこと

この章では、2017 年から 2019 年までの 3 年間の新入社員研修でのトピックスを挙げながら研修担当者としてわかったことを説明する。

#### 3.1 プログラミングの考え方・開発ツールの使い方

「プログラミングの考え方」と「開発ツールの使い方」の学習を一緒にしてはいけない。

プログラミング入門の書籍の中には、まず便利な開発ツールをインストールし、それを使用してプログラムを作成していく方法を紹介しているものが多く存在する。便利な開発ツールはシステム開発を効率よく進める上で不可欠なものである。よって開発ツールの使い方も当然習得する必

要があるが、プログラミングを学ぶ際には、便利な開発ツールは使うべきではないと考える。開発ツールは必要な作業を勝手に実行してくれることがある。下手すると裏で自動的にコンパイルが実行されていることすら認識せずに、開発ツールの実行ボタンを押下すれば、プログラムが動くと思いきや、コンパイルエラーを解消するために意味はわかっていないが、とりあえずソースコード上に表示された赤い波線を消せばよいと考えている。筆者はそんなプログラマを実際の開発の現場でも数多く見てきた。

「プログラミングの考え方」と「開発ツールの使い方」の学習を一緒に実施すると表面上だけ理解した気になって、一応動くものは作成可能だが、内容をきっちりと理解していないプログラマが生まれる。当然そのようなプログラマには、何か問題が起こった時に原因を追究し、解決する能力は備わっていない。

そのため、フルタニ産業株式会社では最初のプログラミング言語研修では便利な開発ツールは使わず、プログラミングを理解した状態になってから開発ツールを使うようにカリキュラムを組んでいる。

便利な開発ツールは、プログラミングができるプログラマが開発効率を上げるために使用するものである。未熟なプログラマの無能を補完するためのものではない。

#### 3.2 研修の評価・仕事の評価

研修での評価と仕事での評価は必ずしも一致しない。2017 年入社社員 A はプログラミング言語研修でテキストに従って学習を進めていたが、周りの学習スピードに遅れないためにサンプルソースコードを見ながらプログラミングし動かすことで「できた」ことにして次から次へと進んでいった。2 週間程経った頃、これまでしっかりと理解せずに進めてきた影響が表面化してきた。様子がおかしかったので詳しく聞いてみたところ、わからなくてテキストが進められないとのことであった。すでに「何がわからないかがわからない」状態に陥っていた。社員 A がどこまで理解できているかを明確にするために、しばらく筆者が横で確認しながらテキストを最初からやり直させた。理解できていると判断できたところは軽く流し、理解できていない部分を本人が理解できるまで徹底的に説明した。その甲斐あって、チーム演習が始まるまでに何とかプログラミングができるレベルにまで達することができた。社員 A の研修中の評価は研修を受けていたメンバの中でも下の方であった。しかし、現場に配置された後も業務終了後に会社に帰ってきてプログラミングの学習を続けた結果、今となっては現場での評価も上々で活躍している。

2018 年入社社員 B は研修開始当初からやる気が感じられず、何事に対してもネガティブなとらえ方をしていた。プログラミング言語研修でもチーム演習でも評価は低かった。やる気のなさもそうだが、チームメンバに対する言動、

態度にも問題があり、チーム内の雰囲気も悪くなっていた。あまりにも目に余るものがあったので、個別に呼び出して指導することにした。元々やる気があまり感じられないメンバであったため、強く指導すると退職するかもしれないという懸念もあった。しかし、このままでは配置先に出すこともできないと考えたため、「辞める」と言い出したとしても、それはそれで構わないと腹をくくって指導を行った。ある意味ギャンブルであった。言い方を一つ間違えれば社員 B は過去の人になっていたかもしれない。結果的には指導を行った後から社員 B の言動が多少前向きになり、研修に対する取り組み方も良くなってきた。仕事の現場に配置されてからは周囲の先輩たちに恵まれたこともあってか、それなりにうまくやれていて、現場リーダからの評価も悪くない。最近では業務終了後に会社に戻ってきて資格取得の勉強を続けている。研修開始当初と比べるとまるで別人のようだ。

研修で良い評価を得ている者は実際の仕事でも大抵活躍している。研修での評価が悪かった者の中には、良い意味で予想を大きく裏切り、実際の仕事では活躍している者が結構な数存在する。

### 3.3 プログラミング言語の違いは問題ではない

新入社員研修で異なるプログラミング言語を短時間で学ぶことによって、プログラミング言語が変わることに対する抵抗は少なくなっている。

過去に筆者と一緒に仕事をしたプログラマの中に「自分は VB しかやったことがないので、他の言語はできません」と平気で発言する者がいた。基本的にプログラミングの考え方を身に付けていけば、言語の違いなんて書き方の違いだけで大した問題にはならない。フルタニ産業株式会社の新入社員研修では「自分はこの言語しかできない」と言うような人材に育てたくはなかった。

2018 年入社の社員 C はプログラミング未経験で入社し、新入社員研修で Java と C# を学んだ。研修中の評価は高くなかったが、研修が終わってからも自分で勉強を続け、徐々に力を付けていった。社員 C が最初に配置された仕事は C# での開発だった。しばらくして、C# での開発の仕事が終了し、次に VB.NET での開発の仕事に配置された。もちろん社員 C にとって VB.NET は初めて扱うプログラミング言語であったが、社員 C は経験がない言語だからといって臆することなく仕事に向き合い、今では期待通りの成果を上げてくれている。

新入社員研修の中で異なる言語への対応の仕方を学んでおり、実際の仕事でもそれが活かされた良い例である。

### 3.4 積極性の欠如

積極的に自分から動こうとしないおとなしい学生が年々増えている。

先輩から話を聞きたいのに自分から声をかけない。研修内容でわからないことがあるのに講師に聞こうとしない。チーム演習ではリーダが一番多くのことを学べることを説明しても、リーダに立候補しない。新入社員の研修の初期段階ではそんな人がほとんどである。

研修後のアンケートで「もっと積極的に先輩社員と話をすればよかった」と書いている新入社員が結構な数存在する。時間はたっぷりあったにもかかわらず、しかも、自分では話がしたいと思っているにもかかわらず行動に移すことができない者が多い。

フルタニ産業株式会社の新入社員研修では学校と違って、他人に答えを聞いてもよい、他人の答えをカンニングしてもよい、他人の答えを写してもよいと最初に伝えている。ただし、写したソースコードも含めて自分自身で書いたソースコードは責任を持って説明できるようになる必要があるとも指導している。しかし、実際には何か問題が起きると、まず自分で調査して何とかしようとする。そのこと自体は素晴らしいことだが、期限がある作業において、調査を続けていたので間に合わなかったでは本末転倒である。次に人に聞きに行くのだが、行き先はまず同期の中でできる新入社員。一番正解に近い答えを出せる講師のところに聞きに来るのは最終手段となっている。

リーダをやりたい人を募ると、しばしの遠慮の間があって、何度か「どう？」と聞いた後で、ひとりふたりと手を上げだす状態。2019 年の新入社員研修に至っては、誰ひとり手を上げなかった。しかし、こちらから指名すれば受け入れてリーダとして動こうとしてくれる。やれば誰でもある程度はできるはずなのに、その一歩を踏み出すことができない。

フルタニ産業株式会社の新入社員研修では、目的意識を持つこと、結果にこだわること、何があっても諦めないことを叩き込んでいる。研修が終了する頃にはほとんどの新入社員が学生から社会人への脱皮を果たしている。

### 3.5 プログラミングの経験・未経験大差なし

大学や専門学校でプログラミングを学んできた新入社員も、文系でプログラミングを全くやったことがない新入社員も大した差はない。

2017 年入社の社員 D は専門学校でプログラミングについてずっと学んできたはずだが、新入社員研修での様子を見ると、基本的な文法を少々知っている程度で、何もない状態から自分で作り上げることはできなかった。プログラミング言語研修の最初こそ文法を知っている程度のアドバンテージはあったが、1 週間もしないうちに同期入社の人と同じくらいでプログラミング未経験のメンバに抜かれていった。研修中の社員 D の評価は最低クラスであった。

2018 年入社の社員 E は在学中にプログラミングを学んできたということで少し期待していた。未経験で入社した同

期メンバに対してプログラミングのことを教えていたが、チーム演習が始まるころには文系出身でプログラミング未経験のメンバの方がよくできるようになっていた。

自分は大学でプログラミングのことを勉強してきて、最初の段階で他の同期メンバより優位に立っていると油断してしまったら、成長のスピードが遅くなり、結果的に未経験で頑張っているメンバに抜かれることになる。

フルタイム産業株式会社では採用の際に「文理問わず」「プログラミング未経験歓迎」と謳っているが、それはこれまでの経験から、入社時点でプログラミングについて知っているかどうか、プログラミングができるかどうかは大した問題ではないと考えているからである。

#### 4. 企業での研修担当者からの大学の情報教育（プログラミング教育）への要望

大学におけるプログラミングの学習と仕事としてのプログラミングとの違いを考えた場合、一番大きな違いは自分の好きなように作れるかどうかである。仕事としてのプログラミングの場合は、顧客の要望、費やせる予算、限られた納期、プロジェクトで決められた書き方、その他いろいろな制約の下、最大の成果を上げなくてはならない。

本章ではこれらの違いを埋めるために大学のプログラミング教育に期待することを説明する。

##### 4.1 仕事としてのプログラミングのポイント

大学におけるプログラミング教育の中で次のようなことを学生が身に付けておくと、プログラミングという点では新入社員研修、その後の仕事を進めやすくなる。

- 基本的な文法、プログラムの書き方
- オブジェクト指向の考え方
- API ドキュメントの見方
- コーディングルールに則ったプログラムの書き方
- クラス、メソッド、変数等の名称の付け方
- コメントの書き方

ソースコードを書くための基本的な文法や考え方を押さえることはもちろんのこと、それに加えて、他人が読みやすい、わかりやすいソースコードの書き方を身に付けてもらいたい。

システム（プロジェクト）で定められたコーディングルールに従ってソースコードを書くことも、わかりやすい名称を付けることも、コメントを充実させることも、すべては自分自身のためだけではなく、他人にも読みやすい、わかりやすいソースコードを書くために必要なことである。

ほとんどのシステム開発はチームで進められる。そして、運用開始後のシステムであったとしても仕様変更等により、随分前に書かれたプログラムに手を加えることもある。つまり、仕事としてプログラミングする場合は、必ず自分以

外の誰かが自分自身が書いたソースコードを修正するということを前提にプログラミングしなくてはならない。

実現可能か試すためにソースコードを書く際に、変数名に「a」等、変数名を見ただけで何を表しているのかわからない名称を付ける者が多い。本番のソースコードとして記述する際には正式な名称に変更しようと思っていたのかもしれないが、大体的場合、最初に付けた名称がそのまま使われる。

英語が得意ではないせいか、綴りミスに気付かない者もいる。削除処理が書かれたメソッドを「delete」と命名したかったのだろうが、「delele」と最初に間違っただけで宣言してしまっただけで、ずっと間違っただけでコーディングが進んで行った。途中で綴りが間違っていることに気付いた者は何人もいただろうが、提供されている共通処理であったため、使う側からは修正することもできなかった。

プログラミング言語研修でも、コメントを一切書かない者が多い。いや、書けない者が多い。自分が書いたソースコード、正確には拾ってきたソースコードを正確に理解していないため、コメントを書けないのではないかと考える。チーム開発をする際のコメントの重要性和合わせてコメントの書き方が身に付いているとよい。

大学で勉強として作るプログラムについては、動けばよい、自分がわかればよいという感覚でソースコードを書く者は多く存在するのではない。他人が読みやすいソースコードを書くことは、プログラムを動かすということだけを考えては不要であるし、適切な変数名を考えること、または、適切なコメントを書くことは非常に面倒くさいと感じる者も多いだろう。それ故に、プログラミングを学び始めた最初の段階でその習慣を身に付けておかないと、後々矯正することが難しくなる。大学におけるプログラミング教育の中で他人が理解しやすいプログラミングを心がける習慣を身に付けてもらえると非常にありがたい。仕事としてのシステム開発はチームで行われ、自分の書いたソースコードは必ず自分以外の誰かが修正するというのを頭に入れておくだけでもより良いプログラミングが可能になるのではなからうか。

##### 4.2 プログラミング言語に依存しないプログラミングの考え方

システムは顧客の要望を実現するためや、課題を解決するために開発される。従って、顧客ごとに、プロジェクトごとに、そのシステムがどんな構成で、どんな機能が必要で、どんなプログラミング言語で開発するのかわかり、その時々で最適なものを選択する必要がある。システム開発を続けていると、急に経験のないプログラミング言語での開発を余儀なくされることもある。「私、VBしかできません」では、仕事はできない。少なくとも仕事の幅は広がらない。

プログラミングの学習は、例えば Java といったプログラミング言語を使って、プログラミングの考え方を学ぶことが重要である。ある処理を実現するために、その処理を小さく分解して、どの部分をメソッド化して、どの順序で組み立てて、どの部分を共通化して等、プログラミング言語には依存しない考え方を身に付けておけば、どんなプログラミング言語であれ、書き方や表現が違うだけで対応可能である。

フルタニ産業株式会社の新入社員研修で実施しているように、一つのプログラミング言語を学んだ後に、異なるプログラミング言語で課題に取り組みさせるような試みを大学のプログラミング教育の中でも実施してみたいかだろうか。

#### 4.3 コミュニケーション能力

システム開発の仕事、特にフルタニ産業株式会社においてシステム開発の仕事を遂行するにあたっては、大学の情報を専門とする学科で学ぶような高度な技術はそこまで必要ではない。3ヶ月の新入社員研修だけでも新人としてのシステム開発の仕事ができるレベルにまで成長させることは可能である。特にプログラミングはシステム開発の中の工程のごく一部にすぎない。

プログラミング能力よりもより重要なのがコミュニケーション能力である。仕事でシステム開発を行う場合、必ず自分以外の誰かがそこに存在する。顧客、会社の上司、プロジェクトリーダー、チームメンバといったステークホルダに対する報連相、ヒアリング、確認は必須である。

仕様書、設計書、質問表、レビュー表、ソースコード中のコメント、メール等、自分自身の考えを読み手に誤解を与えずに文書に表現するスキルも重要である。

良好な人間関係を作るためのコミュニケーションスキルを在学中に身に付けることが出来れば、社会に出てからも可愛がられ、目をかけてもらえ、質問しやすくなり、より仕事がしやすい環境を構築することができる。

#### 4.4 システム開発の仕事体験

大学におけるプログラミング教育をより効率的に仕事に直結させるために、できるだけ早い段階でシステム開発の仕事に触れ、システム開発の仕事をするためにはどのような点に注意すべきかを知ることが重要である。

学生の早い段階で実際の仕事について知る、または体験する機会があれば、そこからの学生生活の過ごし方が変わるかもしれない。目標を見つけることができ、それに向かって学習する密度の濃い学生生活を送れるだろう。

そこで、大学に入学後1年生からでも1、2週間程度の期間で仕事に近い経験ができるインターンシップに参加し、在学中に学生と社会人との違いを知っておくと意識もだいぶ変わるだろう。実際フルタニ産業株式会社で実施してい

るインターンシップに参加した学生は2週間というインターンシップ期間中に別人のように変わる者もいる。

もしくは授業の一環として、臨時講師として大学に企業からゲストスピーカーを招いて、就職にまつわる話や、実際の仕事の内容を学生に向けて話すような機会を増やしていくと良いのではないか。

大学がどんなに良いカリキュラムを用意したとしても、大変残念なことではあるのだが、結局は学生側の意識の問題で十分に成長できていないのではないかと考える。在学中に学生自身の意識をできるだけ社会人に近づけることが可能なカリキュラムを早い段階で用意することにより、学生にとっても大学で受ける授業がより有意義なものになるであろう。

#### 4.5 より濃い新入社員研修のために

以上のことを在学中に習得できていると、新入社員研修でプログラミング言語基礎の部分や社会人研修の一部を省略することができ、より仕事をするために必要な能力を身に付けるための内容の濃い研修を実施することができる。在学中に上記のようなことを学び、身に付けることが可能な授業を大学側で用意していただくと、新入社員を短時間で戦力化することが可能であると考えられる。

### 5. おわりに

一企業の研修担当者として、大学におけるプログラミング教育に関して、是非、実施してもらいたいと考えることは次の2点である。一つ目は、大学にプログラミングの授業において、プログラミング言語に依存しない教育、つまり、どのプログラミング言語（できればビジュアルプログラミング言語以外）でも構わないので、プログラミングの基本概念である構造化プログラミングの理解と、配列や引数のある関数の利用などを本や Web サイトを参考にして良いので、これらを利用し、自分だけでプログラムを書けるようにしてもらいたいと考えている。

二つ目は、低学年からのインターンシップ参加への参加である。低学年からのインターンシップ参加によって、学生自身の意識改革が期待できる。特に仕事をする上では、何を身に付けるべきかを早い段階で明確にすることができる。社会に出てからやることが明確になれば、学生の意識が大きく変わり、しっかりとした目的を持って大学の授業を受講するようになると思う。このようにすることで、大学で学んできたことが社会に出てから、より活かすことができるはずである。

上記の2つは、10年後のイノベーション（技術革新）がどのようなになるかは誰も予測できないが、AIの活用など、これからの時代でも必要とされる、実社会で通用するプログラミングスキルの素養だと考えているからである。



**謝辞** 本報告の作成にご協力頂いたフルタニ産業株式会社  
の社員をはじめとする皆様に、謹んで感謝の意を表す  
る。

### 参考文献

- [1]経済産業省 商務情報政策局 情報処理振興課. IT人材の最新動  
向と将来推計に関する調査結果～報告書概要版～.
- [2]文部科学省, 小学校プログラミング教育の手引 (第一版)
- [3]中山清喬, 国本大悟. スッキリわかる Java 入門 第2版. イン  
プレス