

ソフトウェア開発における デジタル人材育成に関する一考察

小林 孝史

(受付：2025年9月6日 受理：2025年9月6日)

1 はじめに

各種「教育」におけるデジタル人材の育成について、「デジタル人材が不足している」と言われているが、デジタル人材が不足しているのではなくて、身の回りにあること・起きていることについて、その本質を残したまま、デジタルで扱いやすい形に変換したり、デジタル処理に落とし込む処理を行える人材が不足しているのではないかと考える。

本稿では、各事業体・立場におけるデジタル人材の育成に関して、ソフトウェア開発や情報システムの維持・管理のデジタル人材がどのようにあるのが望ましいのか、考察した。第2節では、各事業体や立場でどのような情報環境にあるか、事業に関する知識等との関係を挙げた。第3節では、情報システムやソフトウェア開発の初学者の周辺環境について述べた。第4節では、それらを受けて、デジタル人材の役割について、どういったことを考慮する必要があるのかを述べ、第5節では本稿全体のまとめを述べる。

2 身の回りにあること・起きていること

社会生活を営むにあたって、それぞれの人にとっての「身の回り」にはさまざまな種類がある。ある会社・団体に勤める人にとっては、その企業で行なっている事業の種類・業態・業務内容が身の回りにあること・起きていることであり、学生にとっては、講義の内容・課されるレポート・勉強・研究がそれに当たる。それぞれの立場で、身の回りにあるもの・起きていることが次々に情報化されていき、あるいは既に情報化された状態にあり、その都度、その環境への適応が求められてきたし、これからも適応し続けることが求められることになる。

関西大学 総合情報学部

2.1 一般企業に団体における業務

一般企業や団体であれば、必要な業務知識は研修や資格取得等の自己研鑽により蓄積していくことになる。かつて、あるドメイン知識を獲得するために、オントロジーを用いた知識獲得・知識蓄積の研究が盛んに行われたことがあった。狭義の目的は、エキスパートシステムの構築に必要な知識を獲得することである。エキスパートシステムとは、専門家の知識を内包した情報処理システムをいい、目指すところは、専門家から獲得した知識を利用して、その専門家が判断したかのように判断を行うエキスパートシステムを構築することである。ただし、エキスパートシステムを構築するために獲得した専門家の経験則・知識は、開発者の「暗黙のオントロジー」に基づいて構築されていることがあり、真の意味でのオントロジーを表現していない^[1]。

その業務知識を情報部門で情報システム化したり、従業員が使用しているコンピュータ等でデータ等を処理するエンドユーザーコンピューティング (EUC) を行ったりしてきた。最近では、ノーコードやローコード開発も流行しており、特にプログラミング教育を受けていなくても、簡単に業務改善のできるアプリケーション開発ができるようなシステムも全社的に使えるようになってきている。

2.2 地方公共団体における業務

地方公共団体、特に基礎自治体においては、市民向けをはじめとした、地域に所在する法人や団体向けの業務等さまざまな業務があり、それらを各部署に担当して行っている。

情報システムでの業務支援ももちろんあり、その際に、人手による業務の一部を情報システムで実施する・情報システムによる処理と人手による処理を併用する・情報システムのみを使用して業務を行う、などさまざまな導入パターンがある。調達方法も自治体によってさまざまで、情報処理統括部署で一括導入したり、各業務担当部署で個別

に導入したりするケースがある。一括導入する方が情報システム間の連携がしやすいとか、ひとつのパッケージシステムで多くの業務処理が実施できるなどのメリットはあるが、パッケージ（および担当事業者）の変更がしにくく、コストが高止まりするという問題点がある。

一方、各業務担当に調達する場合は、その業務システムに複数の提案が可能になるなど、競争原理が働いてコストは低く抑えることも可能になるとされる。どちらも、改善すべき点もあるが、なにより、そのシステムを導入したときの思惑が年々失われるようなこともある。

2.3 学生の「業務」

学んでいる立場の学生にとっての「業務」とは学業に他ならないが、その「業務」に対して情報技術を利活用することは、コロナ禍以降に急速に高まった。

ところが、同時期に生成系 AI が目覚ましく発展したため、生成系 AI によって作成されたレポートの問題も明らかになってきた。関西大学でも、生成系 AI ツールへの対応に関する考え方を作成し³⁾、学生に周知されているところであるが、易きに流れることは止められず、筆者の担当する情報処理関係科目等において生成系 AI のみで作成した課題プログラム等を提出する受講生や、講義資料や教科書は読まずに生成系 AI を利用した調査を行っている受講生が多く見られる状況もある。情報処理技術は日々進歩するものの、学生の学びが追いついておらず、その進歩に絶望や諦めが生じて、検索や生成系 AI に頼った学習になりつつあるかもしれない。

3 初学者に厳しい環境

ソフトウェア開発を行うにあたって、誰もが初学者だったはずであるが、30年前と現在では、環境が全く異なっている。インターネットはまだ一般的ではなく、情報源は教科書やプログラミング系の図書であり、各事業者の開発方法に沿ってソフトウェア開発を行ってきた。それが今や、インターネット上にはソフトウェア開発を行うための情報には困らないくらいに情報があり、個人でも組織でも開発しやすい環境は整っている。

3.1 初学者には複雑に見える開発モデル

開発の流れとしては、従来、多くの組織がウォーターフォールモデルで開発を行っていて、それはまだ継続しているが、新しくはアジャイル型開発が使われるようになり、さらに CI/CD (Continuous Integration and Delivery)

開発によって、アプリケーションやサービスの開発を加速し、継続的にソフトウェア部品を統合し、テストの自動化によって問題の発見を早期化して修復する、一連のサイクルを継続的に行っていく。

そういった最新の開発手法については、ソフトウェア開発の初学者にとっては、一見、厳しいものに見えてしまう。情報処理関係の教育を受けていなくても、情報処理技術の仕事に就く場合も多くあり、プログラミングの初心者であると同時に、開発の初心者であり、その仕事に就いた段階で周囲の開発者との能力差もある。その経験的な差や能力的な差をできるだけ埋めるために研修プログラム等を用意して、長期的に埋めていく必要はある。

一方で、情報処理技術の教養的なものを学んでいなくても、最新の開発手法等について興味を持っている人材が存在していることも確かであるので、両者の満足度または個々の満足度を上げる教育プログラムの確立も重要と考える。

3.2 できあがってしまっている情報通信技術

現代の学生や初学者にとって、情報技術は利用の面でもほとんど不満がなく、ほぼ完成された形で見せられている。彼らは、筆者の世代（1990年代初頭からインターネットをはじめとした情報通信技術の発展とともに知識を蓄積してきた世代）のような知識と経験を徐々に蓄積してきたのではなく、いきなり現代の情報技術を詰め込まれる教育を受けていて、物心ついたときには既に情報通信技術は社会基盤となっており、いい意味でも悪い意味でも、その基本的原理や動作を知らなくても、コンピュータすら不要で、利用することはできてしまっている。それらを超えたところに最新技術があるが、年々、新しい技術が開発されていっているの、「できあがってしまっている情報通信技術」の上のさらに「上の階」が積み重なってしまう、「ゴール」がさらに遠のいてしまうような感覚にもなるだろう。それを感じさせない教育プログラムも必要ではないだろうか。それを超えた先に、新しい技術開発が待っているし、デジタル人材も生まれてくるのではないだろうか。

3.3 溢れすぎた開発情報

プログラムの作成方法は複雑になってきており、プログラムライブラリのモジュール化が進みつつも、同じような機能を持ったモジュールがいくつも存在しており、同じことを実現するにもいくつかの方法が存在している状態である。オンラインのバージョン管理プラットフォームである Github には、それこそ何千万ものプロジェクト、リポジトリが存在しているが、その品質は、個人の趣味的なもの

から製品レベルに達しているものまでである。問題解決のシステムを構築するために、いちから開発しなくても、既存のソフトウェアやパッケージ等を組み合わせて問題解決のシステムを構築することも可能²⁾で、それが実用レベルに達していることも多くある。

そのようなプログラム開発の環境において、初学者が迷うのは「どのモジュールを使用すべきか」ということである。大半の初学者は、最初に学んだ教科書や図書、ウェブサイトに書かれていることをそのまま利用することになると思われるが、教科書等の種類が非常に多いことがあって、最初の「教科書」的な情報源を得にくいのが現状である。安易に検索エンジンに頼れば、企業広告に引き寄せられ、授業であれば、まさに教科書はあるものの、実践的ではないものも多く、実践的なものでは初学者には敷居が高く、途方にくれる者も少なくない。

通常の社会活動の中でも周囲の評判等のいわゆる「口コミ情報」を参考にすることが多く、溢れすぎた情報の中から一つを選ぶために評判を参考にすることもある。先に紹介した Github 内のリポジトリには評価指標があり、そのモジュール等を選ぶ際の参考になる。

4 デジタル人材の役割

情報システムやプログラムはその時点で最適な動作や業務を行うに足る機能を持っている。実際に使用していくと、徐々に、不具合や実は不都合な操作になっている部分が明らかになっていく。その箇所については、開発者による改修が必要になるが、システム等の方針や、影響範囲等の解析の都合上すぐには対応できない場合には、問題の解消が相当遅れる場合もあったり、問題が解決しない場合もある。そのような場合には、エンドユーザーがその箇所の回避行動をとったり、その処理を使わないようにする緩和策が必要であるが、根本的な解決にはならない。そこで、エンドユーザーだけでなく、デジタル人材も共に問題点を把握しつつも、業務を行うための方策を考えていくことも必要である。ここでは、デジタル処理を中心にした業務フローの見直し、情報システムの GUI 操作の問題、情報技術の正しい理解、開発コミュニティの活用を考えてみる。

4.1 デジタル処理を中心にした業務フローの見直し

デジタル人材としては、処理内容・作業内容の全体を把握した上で、デジタル至上主義でもなく、単なる情報システム化でもなく、コンピュータ処理にして運用するコストも考慮に入れて、コンピュータ処理を行う部分、人的処理

の部分に切り分けて、業務フローの見直しにつながるような提案のできる人材であることも重要であると考えられる。

パッケージソフトウェアやシステムが存在していても、既存の業務にシステムを適合させるための「カスタマイズ」も多くなればなるほどコスト増につながるので、業務をシステムに合わせることも考えていく必要はある。

もちろん、システムに対する改善提案も必要である。情報システムであれば利用ログや操作ログが残されているはずなので、そこから利用率や個々の操作の使用率なども数字として挙げることで、普段は目にする事のない「情報を可視化」し、カスタマイズの重要度を測って次期システムの仕様を決めることもデジタル人材としては必要な業務であろう。

人手をかけて処理している業務を省力化するために、RPA (Robotics Process Automation) を採用することも多くなっていると思うが、それは一時的な対応として、将来的には「その処理を経ることなく、情報システムへデータが流れていくための方法」を考えることも重要である。RPA による処理が残ったままでは、いつまでもその「変換処理」に頼ることになり、完全なデジタル処理には移行できないままになっていることになる。そのような部分についても設計・提案できるような人材も必要である。

4.2 ウェブ型情報システムの GUI 操作の問題

情報システムの主流がクライアントサーバ型からウェブ型に移行して久しいが、大量処理に向かないことは周知のとおりである。ブラウザで操作していく際に、データを入力したり、ファイルを選択したり、登録ボタンを押したりする GUI による操作が必要で、少数であれば手作業でも可能であっても、数十件を超えてくると GUI を使った単純作業の繰り返しは、ミスを誘発する原因ともなる。

RPA による自動化もその解決方法の一つではあるが、一部の情報システムに備わっている、バッチ処理やデータの一括登録のための準備に必要な GUI 操作がやはり問題である。

GUI 操作なしで多くのデータを入力できるようにするために、目的に適合した言語、得意な言語やシステム等を使って個人またはチーム体制で開発する。その選択は、開発者に委ねられるが、将来にわたって維持できる言語等を選択し、かつ、定期的な見直しにより、適したものを選択し続ける必要はあると考えている。そうした「アンテナ」を張って、定期的な情報収集を行うこともデジタル人材の役割の一つであろうと思う。

4.3 情報技術や制度の正しい理解

情報技術や制度は年々新しいものが発表され、実際に使われていっている。新しい技術や制度を誤って理解していると、新しい技術を使わない方向に向かってしまったり、新しい制度を実現するために技術を使いこなすこともできなくなる。

どの技術や制度が正しく理解されていないか、ということを上申上げるつもりはないが、報道やSNS等でも間違った理解が広まっていたり、誤った内容で議論が進んでいることは認識しているので、細々でも、確実に正しい理解が広められるような情報技術の使い方等にも目を向ける必要もあると考えている。

4.4 開発コミュニティの活用

バージョン管理プラットフォームにもコミュニティ機能があり、それに参加することによる情報の入手は有用と考えている。インターネット黎明期にはNetNewsがあり、ソフトウェア開発だけではなく、さまざまなトピックに関するコミュニティ機能を担っていて、ソフトウェアのリリース情報やバグ報告、新機能の議論などに活用されていた。それが、ウェブ技術の発展とともに「掲示板」へ移行し、そしてまた、バージョン管理プラットフォームにコミュニティ機能が搭載され、それぞれのプロジェクトやリポジトリ等で議論が進んでいる。2025年7月現在、「/community」という文字列を含んだリポジトリ数は228,000件を超えており、非常に多くのコミュニティが存在していることが窺える。

これらのコミュニティに参加することで、より多くの情報を入手することもでき、そのプロジェクトやリポジトリについてより深い理解を得ることができるようになる。

5 まとめ・イノベーションへの期待

前節で、デジタル人材にもコミュニケーションが必要であることを説明した。実社会に既にある問題に目を向け、その問題の本質を捉えて、それを解決するためにデジタル技術を活用する。問題を発見するために、デジタル技術を使って可視化する。技術を獲得するにも情報収集のためのコミュニケーションは必要で、問題を解決する方法、それがデジタル技術なのか、そうでないかに関わらず、それを採用してもらうためには、意思決定をする人物を説得できるコミュニケーションも必要である。

坂村の著書^[2]の第1章の締めくくりにこうある：『イノベーションに大事なものは、技術力でもマーケティング力でもない。やり方を変える「勇気」だ。そして、現代の情報通信技術がもたらしたのは、そういう勇気を助ける「自由」なのである。』

デジタル技術で実社会に存在する仕組みを変革する、その活動に勇気を持って自由に取り組んでくれる人材が育ってくれることを期待したい。

参考文献

- [1] 溝口, “オントロジー研究の基礎と応用”, 人工知能学会誌, 14, pp.45-56.
- [2] 坂村, “DX とは何か”, KADOKAWA, 2021.
- [3] 関西大学教育推進部, “定期試験・レポート作成等における生成系 AI ツールの使用について”, https://www.kansai-u.ac.jp/ja/about/pr/news/2023/07/post_72537.html, 2025/07/31.