

ソフトウェア開発業務と文系学生との親和性について

中 島 豊四郎
松 山 智恵子
岩 田 員 典

1. はじめに

今日の高度に情報化された社会において情報システムや情報機器等は非常に重要な社会基盤である。これらのシステムや機器はソフトウェアなしには、まったく機能しない。その意味においてソフトウェアは非常に重要である。このような中、ソフトウェア開発に関する研究は、その生産性や品質、プロセスや管理に関するもの、教育や人材育成に関するもの¹⁻¹⁶等がなされている。また、ソフトウェア開発は人が行うため、人的アプローチ^{17,18}からの研究もなされている。しかし、ソフトウェア開発に必要な能力については、その上流工程を主として担当するSE (System Engineer) を対象にした問題解決力とコミュニケーション力を向上させる方策^{14,15}についてのものはあるが、ソフトウェア開発全般わたって必要な能力についてのものは、ほとんどない。また、ソフトウェア開発と文系の学生との親和性について言及したものもない。

そこで、ここではソフトウェア開発の各プロセスの中でなされる開発業務を通して、ソフトウェア開発に従事する人に必要とされる能力について検討した後、ソフトウェア開発と文系の学生との親和性について述べる。さらに、その

親和性の一つであるプログラミング力をつけるために文系学部で実践したプログラミング演習とその効果について述べる。

2. ソフトウェア開発のプロセス

ソフトウェア開発の手法としてはプロトタイプング、ウォーターフォール、スパイラル、アジャイルモデル等^{22,25,27,29}あるが、最もよく利用されているのが、ウォーターフォールモデル²²である。このモデルは各企業や組織によって、また、開発対象のソフトウェアの種別により適宜定義され運用^{23,24}されているが、ここでは一般によく知られている簡略化モデルのプロセスの概要（業務内容）²²を表1に示す。表1に示すように、このモデルの最初の工程の要件定義・開発要件定義では、顧客の要望をドキュメントやヒアリング等を通して、客先要求仕様（要件定義）として定める。また、その要件定義（What）を基にそれをいかに（How）実現するかを開発要件にまとめる。その後、外部設計、内部設計、プログラミング（モジュール設計、プログラムの作成）、単体テスト、結合テスト、システムテスト、運用テスト、システム移行を経て運用・保守までのプロセスがある。

このように一つのソフトウェアが完成までには、要件定義・開発要件定義から運用・保守に至るまで多岐にわたる業務が必要であり、その業務に従事する人々には種々の能力^{12-14,19-21}が求められることがわかる。また、プログラム作成は1工程の1部分にすぎない。運用・保守のプロセスは開発したソフトウェアが搭載されている情報システムや情報機器が使用され続ける限り長期間にわたってソフトウェアは利用され続けることになるので大事なプロセスであることもわかる。

しかしながら、このような多岐にわたるソフトウェア開発業務をこなす人的能力について検討したものはほとんど見当たらない。あるものとしては、その

ソフトウェア開発業務と文系学生との親和性について

表1 ソフトウェア開発の工程の概要

プロセス名	業務内容
要件定義・ 開発要件定義	客先の要望をヒアリングして実装すべき機能や満たすべき性能などを明確にし要求を定義する。また、それを基に開発要件としてまとめる。
外部設計	開発要件定義の内容をもとに画面や帳票などのユーザーインターフェースを設計する。
内部設計	外部設計の内容をもとに開発するシステムを大まかな機能ごとに分割し、それらのコンポーネント間をつなぐインターフェースの仕様などを設計する。
プログラミング (モジュール設計、 プログラム作成)	内部設計をもとにモジュールを定義しモジュール仕様書を作成する。
	モジュール仕様書を基にプログラム言語を用いてプログラムを作成する。
単体テスト	作成したモジュールが仕様書で要求された機能を満たしているかどうかを検証する。
結合テスト	複数のモジュールを組み合わせて検証する。主にモジュール間のインターフェースがうまく機能するかを確認する。単体テストが完了したあとに実施する。
システムテスト	システムが全体として要求された機能や性能を満たしているかどうかを検証する。結合テストが完了したあとに実施する。
運用テスト	実際の業務の流れに沿って利用してみて問題なく動作するかを検証する。また、業務担当者がシステムの操作や運用に慣れるための工程でもある。
システム移行	実際の稼働環境へシステムを移す。ある時点で旧システムから新システムへ一気に切り替える一斉移行や、システムの機能単位ごとに新システムへ切り替えていく順次移行などの方法がある。
運用・保守	「運用」はマシンの起動や停止、現行のシステムを日々動かしていく作業である。運転状況の監視や、CPU やメモリの利用状況などシステム資源の監視も行う。「保守」とはシステムを改善・変更する作業である。主にシステムの障害や改善要望に伴うプログラムやデータの改修を行う。また周辺機器のリプレースやアップデートなども行う。

上流工程を主として担当する SE を対象にした問題解決力とコミュニケーション力を向上させる方策^{14,15}についてのものがある。SE がソフトウェア開発業務全般にわたって担当するのは費用の面から現実的とは言えないが、SE 求められる能力はソフトウェア開発業務に必要な能力であるので、ここでは先ず SE 求められる能力について検討する。主に開発の上流工程（最下流工程の運用テスト等の立会を含む場合もある）や開発全体を統括する SE に求められる能力としては、問題発見・解決力、システム構築力、コミュニケーション力、プロジェクトの遂行力があげられている¹²⁻¹⁴。

問題発見・解決力とは、問題を発見しそれを解決する能力であるが、そのためには論理的思考力、抽象化力や分析力等、種々の能力が必要である。そこで、ここでは問題発見・解決力の中で一番重要な論理的思考力と言い換えることにする。

システム構築力とは、システムをどのようなサブシステム（ソフトウェアの大きな集合体）で構成し、順にその集合体をモジュール（あるまとまった小さなプログラムの集まり）まで詳細化しコード化する能力である。これには、論理的思考力、抽象化力やプログラミング力を求められるが、論理的思考力は、既にあげているので、ここでは抽象化力とプログラミング力に言い換えることにする（ただし、SE が担当するのは、サブシステムの構成くらいまででモジュールまで担当することは少ない）。

コミュニケーション力は、他者とコミュニケーションを上手に取ることができる能力である。プロジェクトの遂行力は、開発するソフトウェアが決めた予算と期間内に目的の品質で出来上がるように管理する能力である。

また、ソフトウェア開発においては、開発の成果物として多くドキュメントを作成する必要がある²⁴のでドキュメンテーション力（SE も含む）もあげられる。

このように、ソフトウェア開発に求められる能力として論理的思考力、抽象化力、プログラミング力、コミュニケーション力、プロジェクトの遂行力、ド

コミュニケーション力があげられる（ただし、これらの能力は、開発の各工程と個々に対応するものではなく複合的に求められるものである）。

次に、開発業務と関連付けて個別の能力について検討する。

3.1 論理的思考力

ソフトウェアは「筋道に矛盾がない」論理から成り立つ表現物であるため、ソフトウェア開発において、特に論理的思考が必要である。この力は、ソフトウェア開発の全行程で必要であり、また、後述するすべての能力の前提となる²⁰。

3.2 抽象化力

抽象化とは、具体的な事実から一般論を導くことである。ソフトウェア開発では、この能力も重要である。複数の事象・事実をよく観察・分析し、それらを一般化して段階的に詳細化して最終的にプログラムという形に組み上げる業務そのものが、ソフトウェア開発の根幹であるからである。特に、ソフトウェア開発の上流工程、特に要件定義や開発要件定義では、このような思考の積み重ねで行われる。この抽象化力はゴードン法²⁸として知られている創造性開発手法の一つとして用いられている。

3.3 プログラミング力

プログラムがないと、いろいろな情報システム・情報機器や情報家電も何の機能も果たせない。この点に関し、プログラミング力は「無から有を生む」ものでソフトウェア開発に必須である。どのようなアルゴリズム（処理手順）も逐次、選択、繰り返し構造の組み合わせで表現³⁰⁻³²されており、プログラムはそれらを最適な言語でコード化（記述する）ことに他ならない。プログラミング力は、論理的思考ができる人ならば、数ヶ月でマスターできると言われている。また、後述するように文系の学生も身に付けることができる。マイクロソ

フトのビル・ゲイツは、人の作成したコードを読んでプログラミング力を付けたと言っている³³。

3.4 コミュニケーション力

ソフトウェア開発は一人で行うものではなく、どのようなシステムを構築するのかについて客先（自社の他の部署である場合もある）との度重なる折衝やプレゼンテーション、また、開発を進めて行く上でチームの多くのメンバーと情報を共有し、協働して作り上げていく。従って、開発関係者との意思疎通やチームにおけるコミュニケーションが非常に重要である。また、コミュニケーションをとる場合には、自分の考えていることを正確に伝えることや相手の考えていることを正確に受け取ることができることが求められる。このようなことが正確に行われないと際限ない仕様変更に伴うプログラムの変更、バグ対応のための無駄な労力につながる。このため、このコミュニケーション力がソフトウェア開発において最も求められる能力としてあげられている所以である^{15,19}。

3.5 プロジェクトの遂行力

ソフトウェア開発における予算や工程の仕事量等の見積、ソフトウェア開発の計画・運用を行う能力であり、いわゆるマネジメント力である。この能力は、ある程度の実務経験を有する必要がある。

3.6 ドキュメンテーション力

ソフトウェアは目に見えないため、各プロセスにおいて成果物として各種のドキュメントを作成する必要がある。また、これらのドキュメントは各工程の終了時に次工程に進むためのレビュー等の対象となる。図1にウォーターフォールモデルを基にしたソフトウェア開発の工程と各工程で作成されるドキュメントとレビューの関係を示す²⁴。これらのドキュメントは、客先はもちろん当該

ソフトウェア開発業務と文系学生との親和性について

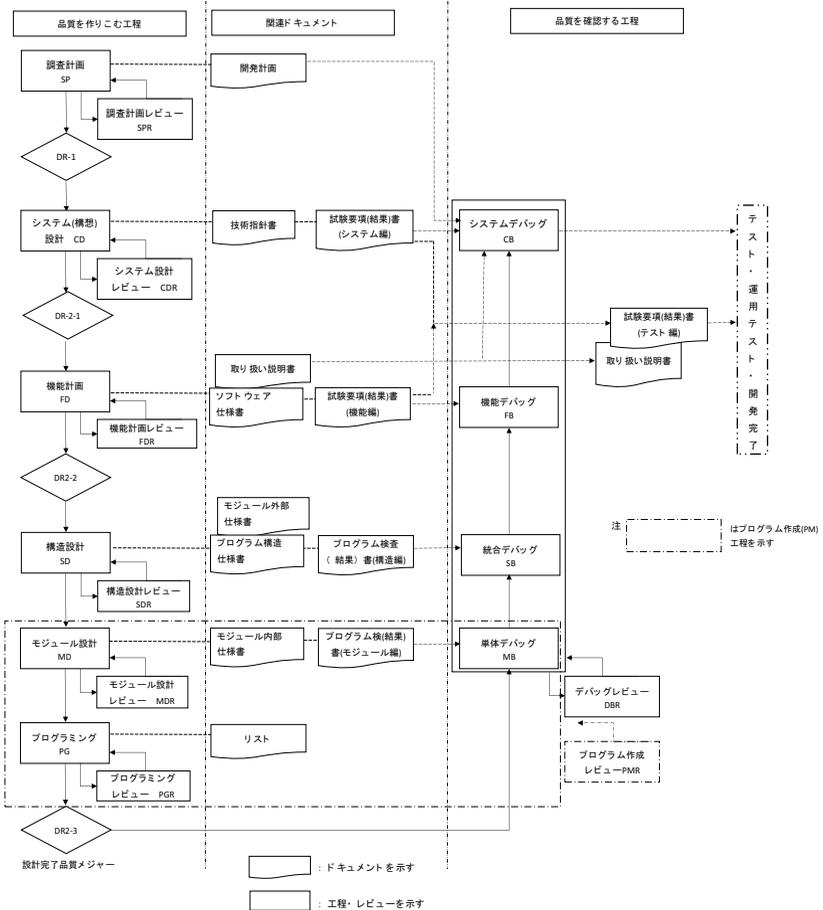


図1 開発工程と作成ドキュメント、レビューの関係

のソフトウェア開発に関係のある人達が理解できるように記述することが求められる。また、ソフトウェアは、メンテナンスがつきものである、メンテナンスする人が理解できるように記述することも必要である。この能力の一部はコミュニケーション力に含められることもあるが、ここでは、ドキュメントはソフトウェア開発における可視化した成果物である、あえて個別の能力

としている。

また、ソフトウェア開発においては、これまで述べたようにプログラミング以外の業務も多数存在する。全工程にわたって必要なコミュニケーションやドキュメンテーションにおいては、PowerPoint や Word、Excel のソフトを利用するケースが多いと推測される。

4. ソフトウェア開発と文系学生との親和性

このように、ソフトウェア開発に必要な能力は、プログラミング力とプロジェクトの遂行力を除けば、どのような分野の業務にも必要である。これらの能力は、大学におけるレポートや卒業論文で一定の評価が得られるレベルにある学生であれば、必然的に身につけている。なぜならレポート・卒業論文の作成は、視点を決めてテーマを掘り下げ、問題点を明らかにし（問題提起）、データ等を収集・分析すると共に問題の原因の解明、対応策の提案^{34,35}を行うからである。また、レポート・卒業論文の作成は問題解決能力、すなわち論理的思考や抽象化力そのものと言えるからである。同時にレポート・卒業論文の作成にはドキュメンテーション力も必要であるし、それらの完成までには、学生間また、教員を含めた議論やプレゼンテーション^{34,35}等も必要である。従って、これらの能力を有する学生は、ソフトウェア開発業務と親和性があると言える。このことは、富士通³⁶やソフトウェア開発を主な社業としている多くの企業³⁷が採用条件に文系理系を問わないとしていることからわかる。さらに、後述するような方法等でプログラミング力を身に付ければ、ソフトウェア開発業務との親和性をさらに高めることができる。

5. プログラミング力をつけるための方策（演習の実践とその効果）³⁸⁻⁴¹

次に、筆者が文系学部で実施したプログラミング力をつけるための演習とそ

ソフトウェア開発業務と文系学生との親和性について

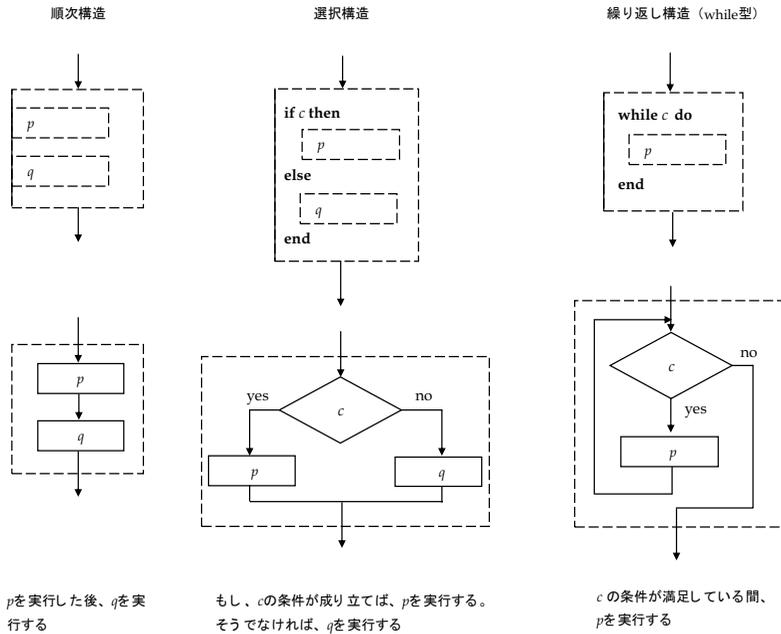


図2 逐次、選択、繰り返し構造

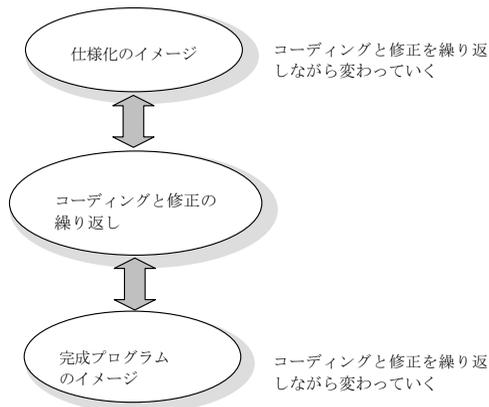


図3 システム進化 (開発) のプロセス

の効果について述べる。本演習では、プログラミングに興味を持たせることを第一に考え、プログラミング教育で一般に用いられている理系的題材に代えて、動画（アニメーション）制作を題材とし、課題として作成する動画の仕様を履修者自身に定義させ、それをプログラム（C言語）で作成することにした。また、プログラムは、図2に示す逐次、選択、繰り返し構造の組み合わせで表現できる³⁰⁻³²ことも説明した。図2において上部は各構造の疑似コード、下部はフローチャートを示す。

履修者は、各自動画の仕様を考え、それをコード化し、デバッグをし、動画を完成させる。完成までの過程においては、仕様で考えたことが用意されたグラフィックス関数では実現できないことや、もっと出来映えをよくしたい等の理由から、図3に示すような仕様の見直し、設計変更、デバッグが何回も繰り返し行われており、システムの進化⁴²が観察された。

そして、演習の最終回に、各自が制作した作品を発表すると共にアンケート調査を行った。アンケート結果を表2、表3に示す。表2より「動画制作は楽しかった」と回答した履修者は82.5%、否定的な履修者は0%、「動画を制作してプログラムを作成する力がついた」は65%、力がつかなかった履修者は5%、「動画制作は難しかった」は90%、難しくなかった履修者は2.5%であった。また、「プログラミングに興味を持った」履修者は70%、否定的な履修者は30%であった。このように大半の履修者は、動画制作は難しいと思ったものの、興味を持ち楽しみながら取り組み、かつ、プログラムを作成する力もついたと認識していることがわかった。

さらに、表3よりプログラミングとは仕様を特定の言語（例えば、C言語）で表現するものということを57.5%の履修者が理解でき、ソフトウェア開発のプロセスを理解できた履修者は82.5%であった。一方、この演習を履修した6年間、249名のプログラミング力を、生産性の尺度（Lines of Code^{43,44} また、Function Point⁴⁵⁻⁴⁸ による評価方法）で調査したところ、前者では約40倍の差、後者では約20倍とその能力には大きな差があることが分かった。こ

ソフトウェア開発業務と文系学生との親和性について

表2 アンケート調査結果1

項目	選択肢			
	はい	どちらとも いえない	いいえ	無回答
グラフィックス（動画制作）は楽しかったですか	82.5%	17.5%	0.0%	-
グラフィックスで動画を制作してプログラムを作成する力がついたと思いますか	65.0%	30.0%	5.0%	-
グラフィックス（動画作成）は難しかったですか	90.0%	5.0%	2.5%	2.5%
プログラミングに興味を持ちましたか	70.0%	30.0%	0.0%	-

表3 アンケート調査結果2

項目	選択肢					無回答
	5 わかつた	4 どちらかといえ ば	3 言えない どちらとも	2 わからなかつた	1 わからなかつた	
Q1	30.0%	27.5%	30.0%	7.5%	5.0%	-
Q2	37.5%	45.0%	15.0%	0.0%	0.0%	2.5%

Q1：プログラミングとはアルゴリズムを特定の言語（たとえば、C言語）で表現するものということが理解できましたか

Q2：実行できるプログラムができるまでの作成過程がわかりましたか

のことは、ブルックス Jr の著した「ソフトウェアの神話」⁴⁹ と一致している。

このように本プログラミング演習は、履修者がプログラミング力を付ける効果があると共にソフトウェア開発の何たるかを知ることができる点で評価できる。

6. おわりに

第3次 AI (人工知能 : Artificial Intelligence) ブームと言われる今日の高度に情報化された社会において、ソフトウェアは不可欠である。社会基盤の大きな要素である。また、ソフトウェアの知識なしでは、すべてがブラックボックス化してしまう。この点に関し文系の学生は、授業 (演習を含む) 等である程度はカバーされていると思われるが。ここでは、ソフトウェア開発業務に要求される能力について、また、その能力を基にソフトウェア開発業務と文系の学生と親和性について検討し、ソフトウェア開発業務と文系の学生との間には親和性があることについて述べた。さらに、プログラミング演習等の情報関連の授業科目を有している文系学部においては、さらに親和性を高めることができる一つの方策を示した。

ここで述べたソフトウェア開発業務と文系の学生との親和性があることは、富士通株式会社³⁶をはじめ多くの企業³⁷がソフトウェアの業務に従事する人材として、文系理系を問わないとしていることから推測できる。今後は、ソフトウェア開発業務に要求される他の能力や性格、この分野における文系学部の卒業生の状況等について検討していきたい。

本稿が文系の学生にソフトウェア開発業務に就いてみたいと思ってもらえる契機になることを期待したい。

(本稿の一部は、^{いとぎく} 梶山女学園「糸菊 2017」に掲載されている。また、第 47 回 愛知大学教養セミナー「文系の学生と職種としての IT (Information Technology) との適応性について」 可能性へのチャレンジ! (2017 年 10 月 愛知大学名古屋キャンパス) で講演したものである。)

参考文献等

- 1 松澤芳昭、大岩元「産学共同の Project-based Learning によるソフトウェア技術者教育の試みと成果」、情報処理学会論文誌 48(8), pp. 2767-2780, 2007
- 2 鈴木直義、堀口貴光、渋沢良太、旗持静香ほか「民産官学協働ソフトウェア開発による大学低学年教育の試み ソフト・イノベーションの視点から」、情報教育シンポジウム 2006 論文集, pp. 45-52, 2006
- 3 大西淳「要求工学 高品質なソフトウェア開発を目指して」、情報処理、2000
- 4 鈴木直義、森下真衣、湯瀬裕昭、渋沢良太「ソフトウェア開発教育と地域情報化」、情報処理学会研究報告コンピュータと教育 (CE)、2006-CE-087, pp. 9-16, 2006
- 5 中所武司「業務の知識を有するエンドユーザ主導のアプリケーション開発技法：フレームワーク・ドメインモデル・サービス連携」、電子情報通信学会技術研究報告. KBSE, 知能ソフトウェア工学 107 (331), pp. 19-24, 2007
- 6 福山峻一、高木英雄、田中僚史、渡辺道広「ソフトウェアプロセスの持続的な改善を誘導するチェックリストの実装手順」、情報処理学会論文誌 42 (3), pp. 529-541, 2001
- 7 松澤芳昭、武田林太郎、大岩元「学生主体のプロジェクトベース・ソフトウェア開発実践教育」、情報教育シンポジウム 2005 論文集 2005 (8), pp. 37-42, 2005
- 8 松澤芳昭、大岩元「産学が共に学ぶ情報システム構築 PBL (Project-Based Learning) の試み」、情報処理学会研究報告情報システムと社会環境 (IS) 2007 (25 (2007-IS-099)), 57-62, 2007
- 9 樋山淳雄、長田圭史、宮寺庸造 [他]、横山節雄「業務ソフトウェア設計・開発教育支援システム」、教育システム情報学会誌 17 (3), pp. 251-262, 2000
- 10 相磯正司、湯浅耕季、鈴木圭一「ソフトウェア開発における統計的プロジェクト管理手法の導入と実践」、富士フィルム研究報告 (56), pp. 31-34, 2011
- 11 岩田貞典、阿南佳之、中島豊四郎「組み込みソフトウェア開発プロジェクトに関するプロジェクト管理モデルの検討」、情報処理学会論文誌 46 (5), pp. 1137-1144, 2005
- 12 竹田尚彦、大岩元「プログラム開発体験に基づくソフトウェア技術者育成カリキュラム」、情報処理学会論文誌 33 (7), pp. 944-954, 1992
- 13 橋本千恵子「プログラム開発体験に基づく上流工程 SE 育成カリキュラムの開発と実施」、情報処理学会研究報告コンピュータと教育 (CE) 1995 (9 (1994-CE-035)), pp. 1-12, 1995
- 14 下郡啓夫、大場みち子、伊藤恵「問題解決能力育成のためのプログラム作成に向けた学習法の提案」、日本科学教育学会研究会研究報告 29 (4), pp. 73-76, 2014
- 15 斎藤祐一郎、久野靖「コミュニケーションスキル獲得を促すソフトウェア技術者教育の試行」、情報処理学会論文誌、教育とコンピュータ (TCE) 1 (2), pp. 7-18, 2015
- 16 坂本憲昭、深瀬光聡、峯恒憲、他「大規模な産学連携による高度 ICT 人材育成に向けての取り組み」、情報処理学会論文誌 49 (8), pp. 2830-2842, 2008
- 17 河野善彌、陳慧「人の設計知識構造と定量評価 (1/2)」、電子情報通信学会技術研究報告. KBSE, 知能ソフトウェア工学 103 (709), pp. 67-72, 2004
- 18 山田茂、影山高章、木村光宏 [他]、高橋宗雄「コードレビューにおける人的エラーと

- 人的要因に関する考察」、電子情報通信学会論文誌. A, 基礎・境界 00081 (00009), pp. 1238-1246, 1998
- 19 南條優「SE の資質と能力」、東京家政学院筑波女子大学紀要 9, pp. 61-66, 2005
- 20 細川泰秀：解説「べた語義：情報システムの源流で求められる人材 今こそ日本の発展に役立つ情報システムを」、情報処理 55 (1), pp. 94-97, 2013
- 21 初田賢司、後藤協子、小林義和「特集 誰もが認める本物の SE 合格ライン」、日経システム (276)、日経 BP 社、pp. 50-57, 2016.4
- 22 富士通エフ・オー・エム「よくわかるマスター 基本情報技術者試験 対策テキスト平成 29-30 年度版」、FOM 出版 2016
- 23 室中健司、原直朗「システム構築の標準プロセス体系：SDEM」、Fujitsu 63 (2), pp. 193-199, 2012
- 24 中島豊四郎、岡部明、田中敏文「ソフトウェア管理システム構築」、OMRON TECHNICS Vol. 30 No. 2 (通巻 94 号) pp. 114-128, オムロン株式会社 1990
- 25 JAVA PRESS 編集部、JAVA+DB PRESS 編集部「Software People」、Vol. 1 技術評論社、2002
- 26 JAVA PRESS 編集部、JAVA+DB PRESS 編集部「Software People」、Vol. 2 技術評論社、2002
- 27 JAVA PRESS 編集部、JAVA+DB PRESS 編集部「Software People」、Vol. 3 技術評論社、2002
- 28 矢野研究所 未来企画室「アイデア発想法」CCC メデヤハウス 2018
- 29 Ali Behforooz, Frederick J. Hudson: "Software Engineering Fundamentals", Oxford University Press, New York, 1996.
- 30 Brian W. Kernighan 著、木村泉訳「ソフトウェア作法」、共立出版社、1991
- 31 宇都宮公訓「コンピュータ入門」、共立出版社、1990
- 32 阿部圭一「ソフトウェア入門 (第 2 版)」、共立出版社、1994
- 33 ビル・ゲイツ「あなたのアンコール "日本の若者へのメッセージ"」、NHK、2000 年 10 月 1 日放映
- 34 石坂春秋「レポート・論文・プレゼン スキルズ」、くろしお出版、2008
- 35 慶応義塾大学通信教育部編「卒業論文の手引き」、慶応義塾大学出版会、2003
- 36 <http://www.fujitsu.com/jp/microsite/employment/workstyle/index.html>
- 37 <https://job.rikunabi.com/2019/>
- 38 松山智恵子、中島豊四郎、石井直宏「プログラミング初心者における動画製作の試みと評価」、電気学会論文誌 C, Vol. 124, No. 12, pp. 2482-2488, 2004
- 39 松山智恵子、中島豊四郎、石井直宏「プログラミング初心者のソースプログラムの分析と傾向」、電気学会論文誌 C, Vol. 125, No. 12, pp. 1914-1915, 2005
- 40 中島豊四郎、松山智恵子「プログラム作成力の評価における LOC とファンクションポイントの相関についての検討」、椋山学園大学 文化情報学部紀要、第 10 巻、pp. 71-81, 2010
- 41 TOYOSHIRO NAKASHIMA, CHIEKO MATSUYAMA, NAOHIRO ISHII: "Development of Teaching Materials for Students to Tackle Programming with Interest and

ソフトウェア開発業務と文系学生との親和性について

- its Effectiveness", "Java in Academia and Research", pp. 37-60, iConcept Press Ltd., 2012.
- 42 Mint (経営情報研究会) 「図解でわかるソフトウェア開発のすべて」、日本実業出版社 (2000)
- 43 R. S. Pressman: "Software Engineering", McGraw-Hill, New York, 1992.
- 44 Ali Behforooz, Frederick J. Hudson: "Software Engineering Fundamentals", Oxford University Press, New York, 1996.
- 45 Capers Jones 著、鶴保証城、富野壽訳 「ソフトウェア開発の定量化手法」、共立出版、1993
- 46 児玉公信 「実践ファンクションポイント法」、日本能率協会マネジメントセンター、1999
- 47 (株)アレア 「失敗しないファンクションポイント法」、日経 BP 社、2002
- 48 河村一樹 「改訂新版ソフトウェア工学入門」、近代科学社、2003
- 49 Frederick P. Brooks, Jr 著、山内正彌訳 「ソフトウェア開発の神話」、企画センター、1991