

ソフトウェアエンジニアと コンピュータエシックス

Software Engineer and Computer Ethics

田中秀和

大同工業大学情報学部

Hidekazu TANAKA

School of Informatics, Daido Institute of Technology

【Key words】

1. ソフトウェアエンジニア (Software engineer)
2. ソフトウェアエンジニアリング (Software engineering)
3. プロフェッショナルエンジニア (Professional Engineer)
4. コンピュータエシックス (Computer ethics)
5. 倫理綱領 (Code of Ethics and Professional Practice)

【概要】

現代はコンピュータが人工物やサービスのいたるところに利用されている。ソフトウェアはその機能実現の要にありながら、ソフトウェアエンジニアは専門職として認められているとはいえない。ソフトウェアエンジニアが専門職として認められるための条件の1つにコンピュータエシックスがあり、それは情報倫理にとどまらず、広い意味での技術者倫理を包含する。ここでは、ソフトウェアエンジニアが備えるべき倫理について考える。

1. はじめに

ソフトウェアエンジニアが備えるべき技術者倫理は広い領域を抱えている。ソフトウェアと言えば連想される情報の倫理を含む、広い意味で技術の倫理の全体を考えなければならない。現代はあらゆるところでコンピュータが使われている。ノートパソコン、ゲーム機、携帯電話等明示的に使われているだけでなく、家庭電化製品や自動車の燃料制御等、システムの中の装置や部品の中にそれとなく組み込まれている。当然組み込まれたコンピュータだ

けでその機能や性能を実現することはなく、そこにはソフトウェアが重要な機能を果たしている。そのソフトウェアを設計、開発にあたる技術者がソフトウェアエンジニアである。

2. ソフトウェアエンジニアとソフトウェアエンジニアリング

IEEE(Institute for Electrical and Electronics Engineers)の用語集では、ソフトウェアエンジニアリングは、ソフトウェアの開発、運用及び保守に対して、系統的な、専門分野として確立された、定量的評価のできるアプリケーションと定義されている (IEEE Std 610.12-1990)。この用語は、1968年のNATO主催のソフトウェアエンジニアリングに関する国際会議（ガルミッシュ、ドイツ）において、はじめて提唱されたとされている。

ソフトウェアエンジニアリングの分野は、ソフトウェアの要求定義、ソフトウェアの設計、ソフトウェアの構築、ソフトウェアテスト、及びソフトウェアの保守を遂行するための知識、ツール及びメソッドを含む。いわゆるプログラミングとかデバッグは、ソフトウェアの構築、ソフトウェアのテストのステージの作業である。ソフトウェアエンジニアとプログラマという職種は混同されることが多いが、ソフトウェアの計画、開発から保守、運用に至るまでを系統的に扱うことができる職種をソフトウェアエンジニアと呼んでいる。

またこの分野に深く関連する分野としては、数学、コンピュータサイエンス、コンピュータエンジニアリング、マネジメント、プロジェクト管理、品質管理、ソフトウェア人間工学、及びシステムエンジニアリング等の分野があり、それらの知識を利用することが多い。

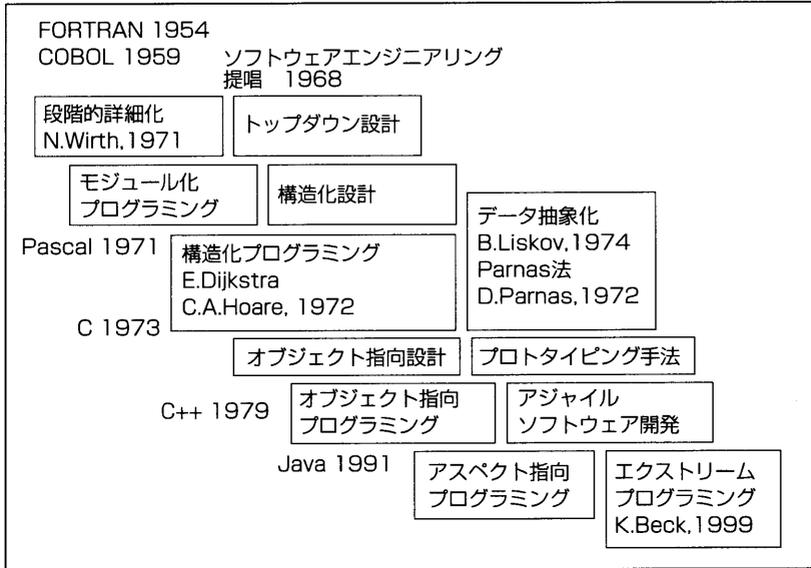
ソフトウェアエンジニアリングの分野は、エンジニアリングといえるほど十分に成熟しているかという議論がある。ソフトウェアエンジニアリングは、ソフトウェアを自ら構築する際の発想、創造、思考、実践の結果得られた普遍的な法則あるいは経験則から構築されるものが多い。例えば、数学の分野としてとり上げられるような普遍的なアルゴリズムとかデータ構造から、ダイクストラの相互排除問題、デッドロックの回避策、資源割り当て問題での銀行家のアルゴリズム等経験から生まれてきたと思われる法則まで数多く

の知恵の蓄積を利用する。そこには統一原理と見なせるような骨太のものは見つからない。

ソフトウェアにおいては、感覚的、情緒的要素を含む多様な要求を求める利用者からの要求を完全に定義することが不可能である、たとえ定義されたとしてもその要求は頻繁に変更されることが多い、また一般に実行時に生成される状態数が極めて大きく、1つの結果をもたらす条件の数とその組み合わせの数が膨大になり、ハードウェアを含めての全正当性の検証はほとんど不可能であり、部分正当性の検証すら現実には困難とされている。それが工業製品としての機能検証、品質保証が完全に網羅する形で形成されていない。すなわちソフトウェアはその正当性や信頼性を実用的な水準で検証することは不可能である。さらに想定していない条件がそこに入り込む余地はあるため、想定外の事象があらかじめプログラムに組み込まれていないこともバグということにすると、バグがないということを保証することが現実には不可能である。このことからソフトウェアエンジニアリングはまだエンジニアリングと呼ばれるにふさわしい水準に到達していないという批判を受け入れざるを得ない状態である。しかし優れたソフトウェアエンジニアは直感、あるいは発想ともよばれるトレース不能な能力によって正当性の高いソフトウェアを構築することができる。従って、直感、発想、経験、管理、規律等、エンジニア及びエンジニアチームの経験や知恵に頼って、品質、機能を確認するヒューマンファクタが特に重視される。

ソフトウェアエンジニアリングの歴史を図1に示す。最近の30年で、モジュール化プログラミング、構造化プログラミング、オブジェクト指向設計、オブジェクト指向プログラミング、あるいはアジャイルソフトウェア開発等いくつかのソフトウェアの品質の向上、生産性の向上を目指した革新的なアプローチがソフトウェア開発の方法論として提案され、ソフトウェア開発の主流に取り込まれてきている。この分野は、エンジニアリングの専門分野としては確立しているとはいえないのではないかという印象を与えるほどに、急速に変化している。ただ一方でこのような革新的なアプローチの変遷は、基盤を揺るがすような変化というより、新しいツールがどんどん入ってきているだけであると主張する人もいる。

図1 ソフトウェアエンジニアリングの歴史



3. ソフトウェアエンジニアとプロフェッショナルエンジニア

米国にはソフトウェアエンジニアに対する資格認定に関しては2つの枠組みがある。1つは任意認定制度(Certification)であり、これはマイクロソフト、オラクル、シスコ等のITベンダーが自社製品に的を絞った自主的に行っている能力認定や、ベンダー中立のCompTIA(Computer Technology Industry Association)が認証するもの等がある。もう1つは免許制度(Licensing)である。免許制度は州政府等の機関が主体となり、法で定められ、関連する業務を行う個人に対して免許を与え、必ず保有を義務化するものである。

一般的なエンジニアに対しては、米国にはプロフェッショナルエンジニア(PE)を規定する法があり、1907年(ワイオミング州)から1947年(モンタナ州)にかけて全州に制定されている。そして米国には現在約50万人のPEがいるといわれている。エンジニアという職務タイトルを名刺、電話帳、論文、履歴書等に掲載する権利と、PEとしてのエンジニアリングサービスを提供する権利が与えられている。特に公共性の高い業務に関してはPEの免許を持つ

ていないと技術業に従事することができない、いわゆる業務独占がある。また国際的な仕事を行う場合にはPEの免許が重要な意味を持つ。

テキサス州におけるソフトウェアエンジニアに対するPE制度導入

テキサス州は、1998年にソフトウェアエンジニアリングをエンジニアリングと認めPE制度に導入することを決めた。テキサス州がソフトウェアエンジニアにPE制度を適用した背景には、人気が高まっているITベンダーの試験、例えば"Microsoft Certified System Engineer"や"Certified Novell Engineer"等に、エンジニアというタイトルが使用され、これまでPE制度で確立してきたエンジニアに対する概念が揺らいでしまうという懸念があったことである。1990年代の半ばにNSPE(the National Society of Professional Engineers)は米国の全ての州に対して、名称としてあるいは雇用に関する呼称としてソフトウェアエンジニアという用語の使用を禁止する訴訟を起こした。その結果大半の州で勝訴した。しかしながらその時点でも、また現在でも、ことソフトウェアエンジニアリングに関してはPEの対象となるエンジニアリングの分類に一般的には入っていない。そこで、テキサス州のPE組織であるTBPE(Texas Board of Professional Engineers)や、全米のPEの制度や試験を横断的に支援する組織であるNCEES(National Council of Examiner for Engineering and Surveying)は対応策を講じた。そして、ソフトウェアエンジニアリングのPE制度を検討した結果、テキサス州においては、ソフトウェアエンジニアを対象とするPE免許制度が認められた。1998年以来2002年まで44人に適用されているとのことである。また、同様の制度はカナダにも存在する。その免許を受けるためには、定められた要件の下で試験を受けなければならない。ただこのソフトウェアエンジニアの免許制度も建設分野のような高度に差別化されたPEになったとは現時点ではいえない。テキサス州の免許の目的は、ソフトウェア開発の契約の際に、PEとして適性や資質があるかどうかを判断する基準を提供することにある。なお、一般のPE認定には実務経験や試験が必要であるが、この段階ではソフトウェアのエンジニアを認定するための実質的な試験が作成されておらず、試験免除制度が適用されているようである。

テキサス州の動きに対するACMやIEEE-CSの対応

しかしながら、テキサス州のソフトウェアエンジニアに対するPE制度に関しては反対する意見が関連業界や学協会に根強く存在する。中心となったのは、ACM(Association for Computer Machinery)である。ACMは、情報処理の分野に対して非常に大きな影響力を持っている学協会である。

TBPEは、1998年ソフトウェアエンジニアがPEの要件を満足するかどうか、ACMとIEEE-CS (IEEE - Computer Society) に支援を求めた。ACM及びIEEE-CSの両協会はソフトウェアエンジニアリングを専門職として確立することを検討するための合同検討委員会JCESEP(Joint Steering Committee for the Establishment of Software Engineering as Profession)を、1993年に既に設置していたが、それが発展的に解消され1998年にSWECC(Software Engineering Coordinating Committee)となる。

そのSWECC委員会では、Joint Task Forceを3つ構成して作業を進めている。それらは、学問としてソフトウェアエンジニアリングの知識体系を検討するSWEBOK(Software Engineering Body of Knowledge, SEBOKと略すこともある)の作成と、ソフトウェアエンジニアに対する倫理標準の制定及びソフトウェアエンジニアの教育カリキュラムの整備についてである。

SWEBOKプロジェクトは、SWECC設立当初からテキサス州のPE制度とは関係なく、ソフトウェアエンジニアリングの発展そのものに寄与するために作業を進めてきており、そのプロジェクトがPE制度との関係性が強調されるにつれてACMはプロジェクトの本来の目的に疑問を呈し始めていく。また、その流れの中でACMとIEEE-CSのJoint Task Forceは、ソフトウェアエンジニアのための倫理綱領 (Version 5.2)を策定し、1997年に公開した。これについては後述する。

ACMは1999年3月に独自にソフトウェアエンジニアリングと免許制度に関する検討を開始した。その結果1999年5月ACMは同委員会としてソフトウェアエンジニアの免許制度を支持することはできないとの結論に達した。その主な理由は、ソフトウェアエンジニアリングの知識や業務は未だに成熟していないこと、免許制度の実施を正当化することができないということ、免許制度では、ソフトウェアの品質や信頼性の保証を提供できるとは考えられないということである。引き続き、ACMは2000年5月にPE免許制度のフレーム

ワークとソフトウェアエンジニアリングの業界の実情と照らし合わせた結果、ソフトウェアエンジニアにPE免許制度を導入するとエンジニアの能力について誤った保証を与えてしまうという懸念や既に能力が認められたソフトウェアエンジニアが免許を得られないという危険性を指摘した。そしてACMはSWECCから脱退し、テキサス州の試みに対して明確に反対の立場を表明した。

その一方、IEEE-CSは10万人の会員を擁する巨大な電気・電子技術者の学協会の中の、コンピュータ関連のソサイエティであり、ACMとは異なり、任意認定制度の必要性を認めており、SWEBOKをベースにして独自にCSDP(Certified Software Development Profession)と呼ばれる任意認定制度を2002年から実施している。2007年5月にWebで確認したところ、認定された625名の名前がCSDPのホームページ上に掲載されている（日本に住所を持つ人も数名いた）。

現段階では、ソフトウェアエンジニアの免許というものは一般的だとはいえない。その必要性については議論が必要であり、さらに実際に実行する際には大きなハードルがある。

4. ソフトウェアエンジニアとコンピュータエシックス

これまで見てきたとおり、ソフトウェアエンジニアのPE免許というものは一般的だとはいえない。その必要性については議論が必要であり、さらに実際に実行する際には大きなハードルがある。しかしながらそのエンジニアの作り出すソフトウェアの社会や環境に及ぼす影響を考えれば、PEとしての認定はどうあれ、倫理的な振る舞いをしないでよいという論理は成り立たない。ここではソフトウェアエンジニアの具備すべき倫理について考える。

情報の倫理とは、情報を扱う上で必要とされる倫理である。情報社会において注意すべき点、例えば誹謗・中傷等をしないといった、情報モラル、情報マナーを意味することもある。情報社会においては、著作権等の知的財産権、個人情報の保護やプライバシー権等に注意する必要がある。インターネット等の情報通信を利用する際のネットワークエチケット（ネチケット）等も含まれる。情報を発信する際の倫理、情報を中継する際の倫理、情報を利用する際の倫理の至るところで必要となる倫理である。ソフトウェアエン

エンジニアはその情報を取り扱う中心に位置しており、情報倫理を具備する必要性は当然ながら存在する。

技術の倫理は、技術者が開発、設計する人工物やサービスを介した倫理である。技術の倫理は、技術者が顧客、雇用者、及び公衆に対して責務として守る倫理であり、技術者としての倫理だけでなく、組織・企業の倫理や社会の倫理に対して深く関与する。ソフトウェアエンジニアの倫理は、ソフトウェアは情報そのものを扱っている。情報の倫理が必要であることは既に述べた。情報の発信、中継、利用のすべてに関わるツールを介してソフトウェアが介在する。サービスを介在した倫理が必要である。私が過去に設計開発に携わった新交通システム、シミュレータ、トレーナあるいは診断システムは、ソフトウェアが組み込まれた人工物でありシステムである。ということから容易に分かるように人工物を介在した倫理、技術の倫理が必要である。ソフトウェアエンジニアには情報の倫理及び技術の倫理の両方が重要な存在となる。

これまでみてきたように、ソフトウェアエンジニアには、現在まだ社会全体との契約であるライセンスが与えられるほど専門職としての社会的地位はない。医者や弁護士等、世間で専門職と認められている職業は、独占的に商売できるライセンスと引き替えに、守るべき厳しい倫理規定が課せられている。ライセンスがないソフトウェアエンジニアは、どのような倫理のもとに仕事をすべきか、個人が考えればいいのか、あるいは、免許がないのだから義務もなく、さらに倫理は必要ないのか、答えは、ソフトウェアエンジニアである前に守らなければならない人間としての倫理、社会の一員としての倫理、組織の一員としての倫理は当然のこと、PE免許はなくてもその産み出す人工物やサービスが社会に及ぼす影響を考えれば、情報の倫理、技術の倫理を踏まえたPEが持つべき専門職倫理と同等の倫理が必要である。

そこで、ACM/IEEE-CSのソフトウェアエンジニアリングに対する倫理綱領について紹介しよう。ACMとIEEE-CSのJoint Task Forceが策定し1997年に公開したソフトウェアエンジニアのための倫理綱領がある。表1にその抜粋を紹介する。なおフルバージョンには前文と8つの原則毎にまとめられたソフトウェアエンジニアの行動指針が含まれており大変参考になる。興味のある人はWebで公開されているのでそれを参照していただきたい。

表1 ACM/IEEE-CSソフトウェアエンジニアリングの倫理綱領

ACM/IEEE-CS倫理綱領 (Version 5. 2)

ショートバージョン 前文

このショートバージョンは、この綱領が何を希求しているのかを極めて簡潔に要約したものである。フルバージョンにはソフトウェアエンジニアリングの専門職としての行動のあり方を、これらの希求に基づいて実践する際どのように変化させるべきか、ということについての具体的な例と説明が述べられている。希求が明確にされなければ、フルバージョンの各項目はその記述をただ表面的に守ることを要求するものとなり、愚直で退屈なものになってしまう。その一方で具体的な説明がなければ、希求はそれがいかに高尚なものであったとしても、空虚なものになってしまう。希求と具体的な説明の両方があってはじめて1つの完結した綱領となる、

ソフトウェアエンジニアは、ソフトウェアの要求分析、仕様の決定、設計、開発、テスト、及び保守を有益で尊敬されるものとするのに関与する。その業務が公衆の健康、安全、及び福祉を損なわないために、ソフトウェアエンジニアは以下の8原則を固く守る:

ソフトウェアエンジニアの守るべき8つの原則

- 1.公共性：ソフトウェアエンジニアは、一貫して公益を守る行動をとる。
- 2.顧客と雇用者：ソフトウェアエンジニアは、公益と一致しているという前提で、顧客と雇用者の最大の利益を得るために行動する。
- 3.製品：ソフトウェアエンジニアは、製品とその変更が可能な限り最も高い水準を満たすことを保証する。
- 4.判断：ソフトウェアエンジニアは、専門的な判断の際には、統合性と独立性を維持する。
- 5.管理：管理者あるいは指導者としてのソフトウェアエンジニアは、ソフトウェア開発と保守の管理に対して、倫理的アプローチに依拠し、さらにそれを促進する。
- 6.専門職：ソフトウェアエンジニアは、公益と一致する、専門職としての統合性と評判を得ることを推進する。

7.仕事仲間：ソフトウェアエンジニアは、仕事仲間に対して公平で、仕事仲間から支持される存在となる。

8.自身：ソフトウェアエンジニアは、専門性の実践のために生涯にわたる研鑽に努め、専門性を実践する際に倫理的なアプローチを促進する。

フルバージョン 前文

コンピュータは、商取引、産業、行政、医療、教育、娯楽、社会全体で中心的な役割を果たしており、その役割はますます拡大している。ソフトウェアエンジニアはソフトウェアシステムの分析、仕様決定、設計、開発、認証、保守、及びテストに直接的に参加あるいはそれらの教育に携わることで貢献している。ソフトウェアエンジニアはソフトウェアシステムの開発に対して果たす役割ゆえに、良いことも悪いこともさせる重大な影響力を有する。このような潜在力をできるだけ良い方向に導くためには、ソフトウェアエンジニア自身が、ソフトウェアの設計や開発という仕事を有益でかつ尊敬に値するものにすべく各自で最大限の努力する必要がある。その責務を負うためには以下に示す倫理綱領を遵守しなければならない。

この綱領には、実務者、教育者、管理者、監督者、政策立案者といった専門職のソフトウェアエンジニアと、さらにはこの専門職の訓練を受けている者、あるいは学生のための行動と決定に関する8つの原則が述べられている。これらの原則は、個人、グループ、及び組織が倫理的に責任を負うべきどのような関係性の中におかれているのかを示し、そうした関係性における主要な義務を定めている。それぞれの原則の中で列挙されている項目は、これらの関係性における義務のいくつかを説明するものである。こうした義務はソフトウェアエンジニアの人間性と、ソフトウェアエンジニアの仕事に影響を受ける人々に対して払うべき特別な配慮、ならびにソフトウェアエンジニアリングの実務に特有の要因に基づいている。この綱領は、ここに書かれた義務を、自らをソフトウェアエンジニアと考える者、あるいはソフトウェアエンジニアになりたいと考える者すべてにとっての義務として規定する。 (中略)

倫理的な対立状況では、これらの詳細な規則を盲目的に信頼するよりもむしろ、基本的な原則について深く考えることによって、最もよく取

り組むことが可能となる。これらの原則は、ソフトウェアエンジニアに次のようなことを促すものである。誰がソフトウェアエンジニアの仕事の影響を受けるのかについて、広い視点で考えること。ソフトウェアエンジニアとその仲間が他の人々に、誠意を持って接しているかどうかについて考えること、社会の人々が自分たちソフトウェアエンジニアの決定をもしその内容がきちんと知らされるならば、どのように思うのかについて考えること、弱者と呼ばれる人々が、ソフトウェアエンジニアの決定によってどのような影響を受けるのかについて検討すること、ソフトウェアエンジニアの行動がソフトウェアエンジニアとしての理想的な専門職として働いていることを判断できるかどうかを考えることである。これらのすべての判断において、社会の人々の健康、安全、福利への関心・配慮が最も重要である。すなわち公共の利益こそがこの綱領の中心に存在するものである。（中略）

この倫理綱領は、ソフトウェアエンジニアリングという分野に参加し活動しようとする人々に求められる事柄を示す教育的な役割を果たす。この倫理綱領は、倫理的な諸問題に関する専門職の一致した見解を書き表したものであり、すべてのソフトウェアエンジニアの責務、倫理的義務について判断する際の参考になり、社会の人々とこれからその職業に就こうとする人々の双方を教育、啓蒙する手段として活用できる。

行動原則

原則1：公共性

ソフトウェアエンジニアは公共の利益にかなうように、特に専門職としてふさわしい行動をとるべきである。

- 1.01. 自分自身の業務に対する全責任を負わなければならない。
- 1.02. 公共の利益と、ソフトウェアエンジニアの利益、雇用者の利益、顧客の利益、そしてユーザの利益とを加減する。
- 1.03. 安全性、仕様準拠、適切なテストの合格が保証され、生活の質の低下、プライバシーの侵害、環境への危害がないことを十分確信できるときのみ、そのソフトウェアを承認する。その業務の究極の効果は公益にあるとすべきである。
- 1.04. ソフトウェア及び関連文書は、ソフトウェアエンジニアの道理にかなう判断において、ユーザ、社会の人々、環境に対して現実にまたは潜

的に存在する危険性を、適切な人あるいは権威ある人や組織に公表しなければならない。

1.05. ソフトウェア、そのインストール、保守、サポートあるいは文書化によって生じる重大な社会的懸念に対する取組みに協力する。

1.06. 以下は省略、続けて原則2：顧客と雇用者、原則3：製品、原則4：判断、原則5:管理、原則6:専門職、原則7：仕事仲間、および原則8：自身まで、具体的で詳細な行動指針が書かれている。

5. 日本におけるソフトウェアエンジニアリング

専門職技術者の特徴として、Johnsonは、(1) 深遠な知識体系を修得し、継続研鑽に努める、(2) 自律性をもつ、(3) 公式に認められた組織に所属する、(4) 倫理綱領、行動指針をもつ、及び(5) 公共の安全や福祉を守る、の5つをあげている。この条件を満足するためにはエンジニアリングの教育課程を卒業することが重要である。まず、その教育課程や教育カリキュラムの議論や整備について考察を加える。ソフトウェアエンジニアリングと専門的学理について、深遠な専門的学理があるかという議論が根強くあることは既に紹介した。学理というより方法論いわゆるハウツー知識が主導ではないかとか、ソフトウェアエンジニアリングの提唱以来40年経過するも、その方法論もいまだに定まらないではないかといった議論が活発である。また、ソフトウェアの信頼性（品質）の向上と生産性の向上は私が技術職として在籍した会社における昇級論文のテーマであったし、現在でもQCD（Quality, Cost, Delivery）としてエンジニアリングの課題となっている。ソフトウェアの信頼性や品質の向上、開発コストの低減や納期の保証や工程の短縮は難問であり、まだまだ若い専門分野であることは確かであるが、英知を結集してその研究や教育体制の整備が進められている。

バブルがはじける前の日本においては、ソフトウェア人材が非常に逼迫した状態で、ソフトウェアあるいはコンピュータ関連のエンジニアリング課程

を卒業した者は非常に稀で、そういった課程を持つ大学そのものが日本には僅かしかなかった。大学卒でない者とか、大学卒であっても、エンジニアリング課程ではない理学部、法学部、経済学部、商学部等卒の者は、一般的にJohnsonの条件を満たさない。そういった自称エンジニアが山ほど参入して、現実の業務をこなしている状況であった。

エンジニアリングの教育課程やカリキュラムは現在でもその整備の最中にある。パブルがはじけ、エンジニアを養成する大学がそのエンジニアリング課程で日本技術者教育認定制度 (JABEE, Japan Accreditation Board for Engineering Education) 対応や教育重視を叫んで、本格的にカリキュラム整備をして、教育内容、教育体制に工夫をしている最中である。しかしながら、こうした黎明期における混乱した歴史的な流れから、専門職の条件をクリアするにはまだまだ長い時間が必要となる。

ソフトウェアエンジニア教育カリキュラム:米国

コンピューティングへの教育を本格的に取り組み始めたのは、米国ではACMとIEEE-CS両協会のJoint Curriculum Task Forceが作成したコンピューティングに関する教育カリキュラムCC91 (Computing Curricula 1991)がもとなっている。それ以前はACM及びIEEE-CSが別個にモデルカリキュラムを策定していた。ACMは主としてサイエンス的視点で、カリキュラム68を策定し、それを78年、88年と改訂してきた。一方IEEE-CSはサイエンス及びエンジニアリングの視点からモデルカリキュラムを77年に策定し、83年に改訂してきた。

ソフトウェアエンジニアリングに関する教育カリキュラムが本格的に検討され始めたのが、その次の世代であるCC2001となる。CC2001は1998年から検討を開始している。そこでは急速に進展する情報化産業に起因するソフトウェアエンジニアリングビジネスへのニーズの拡大、ソフトウェアエンジニアリングのプロフェッショナルエンジニアとしての確立への要求に刺激されて、既に紹介したようにSWECCが設立された。その中で、コンピュータサイエンス (CS)、コンピュータエンジニアリング (CE)、ソフトウェアエンジニアリング (SE)、及び情報システム (IS) の4つの領域について検討を加え、おのおのCS2001、CE2004、IS2002を作成した。ソフトウェアエンジニアリン

グについてはSE2004 が作成されている。なおSE2004はJpn1として日本におけるソフトウェアエンジニアリングの知識体系のもとになっている。その後CC2004になって、情報技術（IT）という領域を追加し（IT2006が作成された）現在に至っている。

ソフトウェアエンジニア教育カリキュラム:日本

日本におけるソフトウェアエンジニア教育カリキュラムはその米国の後追いついてきている。1970年代に情報関連の学科として情報工学科、計算機科学科、情報科学科が誕生し、その後情報処理技術者の大量不足時代を迎え多くの大学が続々と学科を設置してきた。1971年の文部省資料によると、情報科学は広範な学問であり、専門学科を画一化することは好ましくないとして、性格の異なる、基本的に4つのタイプの学科を想定し、それぞれに平均的なカリキュラムを提案し、各大学の事情に則して弾力的に運用するべきであるとされた。基礎理論を中心としたもの（CSに相当）、ハードウェアを中心としたもの（CEに相当）、ソフトウェアに中心としたもの（SEに相当）及び情報システムに中心としたもの（ISに相当）である。その後20年ほど経過し、情報関連学科のアイデンティティを明確にするためにも、共通の尺度が必要ということで、コンピュータサイエンス教育カリキュラムJ90と名付けられたカリキュラムが検討された。専門学科ならばこれだけは習得しておくべき科目を示し、卒業生の専門能力について共通の尺度を明らかにすべきであるとされた。現在の全国の情報関連学科のカリキュラムの骨子を形成しているといつてよい。その後情報分野の広がりや踏まえて見直しがかけてられて、J97が発表されている。その頃情報処理学会にアクレディテーション活動が始まり、JABEEへの取組みもその頃から始まるようになった。その1つの動きとしてソフトウェアエンジニアリング領域が明確に意識されるようになり、ACM/IEEE-CSのSE2004の策定に参加し、その結果、日本向けモデルJpn1を作成した。

現在は、J07としてカリキュラムの整備作業がされている。この段階では、CS、CE、SE、IS、ITの5つの領域の検討がなされており、米国とはほぼ歩調が揃っている。なお、JABEEでは情報及び情報関連分野の領域分けにはCS、CE、SE、ISの4つが例示されており、IT領域は現段階では例示されていない。

い。ソフトウェアエンジニアリングの領域は、J07-SEとして、Jpn1のサブセットとして位置づけ、より発展形を検討可能にした形で現在精力的に整備されている。

安全性を最優先するシステム

安全性を最優先するシステム(safety critical system)において、ソフトウェアは安全を保証できるかという問題がある。ソフトウェアビジネスではQCDという課題があることを示したが、安全性を最優先するシステム例えば、公共システムや医療用アプリケーション等ではQ(quality)の尺度すなわちソフトウェアの信頼性が最も重要な尺度である。そのため、信頼性の高いオペレーティングシステムの選定からはじめてソフトウェアの全開発工程において慎重に設計を進める。しかしながら想定していない事象に対する対応策をあらかじめプログラムをすることはできないのは道理である。そのことから、100%安全であるソフトウェアをすることはできない。ソフトウェアに関しては、いわゆる設計上のミスや欠陥をバグといているが、その想定していない事象に対する対応策が講じられていないこともバグといている。ひっくり返して、バグであり、そのため、使用許諾のパスを設定している。しかしながら、使用許諾のパスは現実には、ほぼ形式的であり悪い言い方をすれば、ミスや欠陥があっても、製品として主張できることになっている。またソフトウェアの動作誤りに対する違法性の概念は一般的にはない。そのため一般にリコールはなく、その代わりにバージョンアップ文化をもつ。そのバージョンアップにも対価を要求される。他のエンジニアリングの成果と比べてソフトウェアエンジニアリングに対して甘い評価がなされている。そのあたりもエンジニアリングとして見なされない遠因ともなっている。蛇足であるが、ソフトウェアエンジニアは業務上過失の訴訟の対象になっていない。

ソフトウェアエンジニアと専門職団体

ソフトウェアエンジニアリングに関連の深い専門職団体としては、日本においては学会と技術者団体とが明確に分かれている。情報処理学会、電子情報通信学会、ソフトウェア科学会、人工知能学会、計測自動制御学会等は基本的に米国の技術者協会に相当する学協会ではなく、工学の研究活動の支援

を指向した学会である。日本工学教育協会はそういった学問の横断的な観点、特に工学教育を中心に据えた学協会といわゆる専門分野の学会とは一線を画しているが、とはいえ技術者の団体ではない。技術者団体の色彩の濃いのは唯一日本技術士会が挙げられる。日本においては平成12年の技術士法の改正以降、技術士がプロフェッショナルエンジニア（PE）という名称を使っているため、日本技術士会が専門職団体にあたるはずだが、米国と違って技術士の名称独占はあっても業務独占はほとんどない。建設部門の一部の業務には独占に近い形態が存在するようであるが、その他の部門では一般的ではない。ソフトウェアエンジニアの専門職団体としてはそういった意味ではぴったりする団体は存在しないといってもよい。

ソフトウェアエンジニアと資格、試験、スキル標準

ソフトウェアエンジニアに対する資格や認定については、日本においてもいろいろな資格、試験、スキル標準が乱立している。まず、技術士（情報工学部門）がある。ただし米国のPEのような免許制度ではなく、技術士の名称独占だけがあり業務独占は一般にない。

また任意認定制度による試験としては、経済産業省の支配下にある情報処理技術者試験センターの情報処理技術者試験（基本情報技術者、初級システムアドミニストレータ、ソフトウェア開発技術者、アプリケーションエンジニア、テクニカルエンジニア（エンベデッドシステム）ほか、13種を数える）がある。また同じくITベンダー系試験（マイクロソフト、シスコ）やITベンダー中立のCompTIAの試験も盛んに行われている。いずれもエンジニアの知識や能力の程度を認定する任意認定に終わっている。

2007年3月から「情報システムに係る政府調達の基本指針」では、一定の金額を超える調達に際して情報処理技術者試験や技術士（情報工学部門）の資格を持つ人材の有無が入札時に審査されることになった（総務省）というニュースが飛び込んできたが、冷静にみれば業務独占とは程遠いものである。

また認定試験ではないが、ITスキル標準（ITSS、11の職種36分野のスキル）や組込みスキル標準（ETSS）といったスキル標準を設定して、業種毎にスキルレベルを競わせるといった利用の仕方をしている。

日本のソフトウェアエンジニアと倫理綱領

ソフトウェアエンジニアが所属する日本における学協会には主として情報処理学会と日本技術士会があり、いずれも倫理綱領を持っている。専門学会としての位置づけの情報処理学会の倫理綱領は基本原則で終わり、具体的な行動指針を提供するものではない。

日本技術士会は日本における技術専門職団体であるため、その技術士倫理要綱を定めている。表2に技術士倫理要綱を示す。さらに2007年1月に宣言された技術士プロフェッション宣言等を通じて専門職であることの意識高揚とその行動指針を示し、現実の倫理問題に具体的に対応できるように整備しようとしている。表3に技術士プロフェッション宣言を示す。しかしながら、先進の米国の専門職団体の示す倫理綱領や技術者倫理への取組みと比較においてその差が歴然とあり、特に現実的な倫理問題への対応や、具体的な行動指針を示しきれていないため、現在はその途上であると考えられる。

技術士は技術士法に基づいており、単なる名称独占は明示されているもののいわゆる業務独占がなく、その意味で医師や弁護士のような専門職ではない。しかしながら技術士の業務の重要性から考えて技術者倫理の遵守は必須でありより現実的な、きめの細かい倫理的対応が望まれる。

ソフトウェアエンジニアに関しては米国においても業務独占がなく、その意味で日本の技術士と同様な立場である。ソフトウェアエンジニアが専門職として確固たる存在となるためにも、ACM/IEEE-CSの制定したソフトウェアエンジニアリングの倫理綱領は参考にすべき例として学ぶべきである。

表2 技術士倫理要綱

<p>技術士倫理要綱</p> <p style="text-align: right;">昭和36年3月14日理事会制定 平成11年3月 9日理事会改訂</p> <p>技術士は、公衆の安全、健康及び福利の最優先を念頭に置き、その使命、社会的地位、及び職責を自覚し、日頃から専門技術の研鑽に励</p>

み、つねに中立・公正を心掛け、選ばれた専門技術者としての自負を持ち、本要綱の実践に努め行動する。

- (品位の保持) 1. 技術士は、つねに品位の保持に努め、強い責任感をもって、職務完遂を期する。
- (専門技術の権威) 2. 技術士は、つねに専門技術の向上に努め、技術的良心に基づいて行動する。また、自己の専門外の業務あるいは確信のない業務にはたずさわらない。
- (中立公正の堅持) 3. 技術士は、その業務を行うについて、中立公正を堅持する。
- (業務の報酬) 4. 技術士は、その業務に対する報酬以外に、利害関係のある第三者から、不当な手数料、贈与、その他これらに類するものを受け取らない。
- (明確な契約) 5. 技術士は、業務を受けるにあたり、事前に相手方に自己の立場、業務の範囲等を明確に表明して契約を締結し、当該業務遂行上両者間で紛争が生じないように努める。
- (秘密の保持) 6. 技術士は、つねにその業務にかかる正当な利益を擁護する立場を堅持し、業務上知り得た秘密を他に漏らしたり、または盗用しない。
- (公正、自由な競争) 7. 技術士は、公正かつ自由な競争の維持に努める。
- (相互の信頼) 8. 技術士は、相互に信頼し合い、相手の立場を尊重し、いやしくも他の技術士の名誉を傷つけ、あるいは業務を妨げるようなことはしない。
- (広告の制限) 9. 技術士は、自己の専門範囲以外にわたる事項を表示したり、誇大な広告はしない。
- (他の専門家等との協力) 10. 技術士は、その業務に役立つときは、進んで他の専門家、あるいは特殊技術者と協力することに努める。

表3 技術士プロフェッション宣言

技術士プロフェッション宣言

われわれ技術士は、国家資格を有するプロフェッションにふさわしい者として、一人ひとりがここに定めた行動原則を守るとともに、社団法人日本技術士会に所属し、互いに協力して資質の保持・向上を図り、自律的な規範に従う。これにより、社会からの信頼を高め、産業の健全な発展ならびに人々の幸せな生活の実現のために、貢献することを宣言する。

技術士の行動原則

1. 高度な専門技術者にふさわしい知識と能力を持ち、技術進歩に応じたえずこれを向上させ、自らの技術に対して責任を持つ。
2. 顧客の業務内容、品質等に関する要求内容について、課せられた守秘義務を順守しつつ、業務に誠実に取り組み、顧客に対して責任を持つ。
3. 業務遂行にあたりそれが社会や環境に与える影響を十分に考慮し、これに適切に対処し、人々の安全、福祉等の公益をそこなうことのないよう、社会に対して責任を持つ。

平成19年1月1日
社団法人 日本技術士会

技術士プロフェッション宣言においてプロフェッションの概念は、

- 1) 教育と経験により培われた高度の専門知識及びその応用能力を持つ
- 2) 厳格な職業倫理を備える
- 3) 広い視野で公益を確保する
- 4) 職業資格を持ち、その職能を発揮できる専門職団体に所属するとある。

日本のソフトウェア産業の行方

最近のソフトウェアの傾向として、ソフトウェアの巨大化、複雑化が挙げられる。ソフトウェア開発にあたるエンジニアも膨れあがり、その品質やコストあるいは納期といったビジネスとしての価値を満足させる技術が確立していません。とはいえ教育での問題点も多くソフトウェアエンジニアが市民権を得るまでにはまだまだ幾多の波を越えなければならない。

ソフトウェアの巨大化、複雑化の波を受けて、問題が顕在化してきたことにブラックボックス化する情報基盤がある。コンピュータというハードウェアの上に、まず基礎となるシステムソフトウェアがあり、さらに幾層ものソフトウェアが積み上げられアプリケーションソフトウェアとして全体を構成するわけであるが、その基礎基盤にあたるソフトウェアが膨大なものになり、そこには高信頼性が要求されるのであるが、想定していない状況に対するソフトウェアをあらかじめ組んでおくことはできないと述べたが、その基盤にあたるソフトウェアがブラックボックス化する傾向が強くなることである。下層にあたるソフトウェアがブラックボックス化すればその上に積み重ねたソフトウェアが不具合を起こしても開発した上層のソフトウェアに責任があるのか下層のソフトウェアの想定外なのかの判断がつかなくなる。悩ましい問題である。

また、ソフトウェア開発の特色はチームを組んで開発にあたることが多いのであるが、その開発人員の構成が必要の都度集められ開発の当初と開発のピーク時とは大きく体制が異なる。ピークに近い時点では設計思想や設計方針といった基本的な設計の考え方がチームの全員に浸透するかどうかが開発のキーポイントになる。その際に問題点となるのはチームのメンバーはコストの関係から、アウトソーシングとなることである。機密保護の関係もあって、アウトソーシングの場合敢えて設計方針や設計の考え方といったものを伝えないで、仕様とか仕様の詳細説明のみを伝えて設計に当たらせることが多い。オフショアアウトソーシングという言葉があるように、そのアウトソーシングもオフショアになるとその仕様そのものもどこまで伝わるのか疑問や懸念を持ってばかりがない。ソフトウェアエンジニアがおかれたビジネス現場では、そういった現状であることも大きな課題を抱えている。

6. おわりに

ソフトウェアは便利で有用な存在であり、そのライフサイクル全体にわたってその存在価値を、持ち続けることができ、その信頼性や生産性が学問的な議論の俎上にのり、工業製品として、いわゆるQCDの要請、特にコストダウンや納期の保証を議論することができ、公衆の健康、安全あるいは福祉に対して責任を持つことができることが求められている。

ソフトウェアエンジニアは、便利で有用なソフトウェアを開発し、その製品やシステムのライフサイクル全般にわたり、高い品質と社会的な責任を持ち、厳しい倫理綱領を持ち、公衆の健康、安全あるいは福祉に対して責任を持ち、公平で統合的な判断ができ、社会からの高い評価と職務からの高い利益を得、日本における最高の職といわれるようなエンジニアでありたい。

日本においては、エンジニアは米国のPEと同等な専門職としては認められていない。ソフトウェアエンジニアも同様である。日本に、米国のようなPE制度が本格的に導入されるには業務独占をはじめとして、技術政策、産業界からの要請等、多くの関門がある。ソフトウェアエンジニアはその米国でも、テキサス州を除いてPEとして認められていない。しかしながら、ソフトウェアエンジニアは、PEとしての適性や資質をもち、継続研鑽に努め、公衆の健康、安全あるいは福祉への責任を持つこと、適正な倫理綱領や行動原則を順守し、自律的な行動をとることが求められている。

[文献]

- ・ Johnson, Deborah G.(2001), "Computer Ethics,3rd Eds", Pearson Education International, pp.59-61 (水谷雅彦ほか監訳,「コンピュータ倫理学」, オーム社, pp.88-90(2002).)
- ・ McConnell, S., Tripp, L.(1999), "Professional Software Engineering: Fact or Fiction?", IEEE Software, Vol.16, No.6, pp.13-18.
- ・ Tripp, L.(2002), "Benefits of Certification", in Software Certification Debate, IEEE Computer, Vol.35, No.6, pp.31-33.
- ・ Kolawa, A.(2002), "Certification Will Do More Harm than Good", in Software Certification Debate, IEEE Computer, Vol.35, No.6, pp.34-35.
- ・ Knight,J.C., Levenson,N.G.(2002), "Should Software Engineers Be Licensed?", CACM, Vol.45, No.11, pp.87-90,(青山幹雄訳, 情報処理, 第45巻, 第2号, pp.177-

180(2004)

- Software Engineering-Wikipedia
http://en.wikipedia.org/wiki/Software_Engineer
- Software engineering professionalism-Wikipedia
http://en.wikipedia.org/wiki/Software_engineer_professionalism
- Debates within software engineering-Wikipedia
http://en.wikipedia.org/wiki/Debates_within_software_engineering
- Money Magazine のザベストジョブ(2006)
<http://money.cnn.com/magazines/moneymag/bestjobs/2006/>
- 渡辺弘美, 「米国の I T 人材市場について」, コンピュータソフトウェア協会
http://www.csaj.jp/info/04/20050303_us_it_softmarket.pdf
- 大学の情報系専門学科のための情報システム教育カリキュラムISJ2001
<http://www.ipsj.or.jp/12kyoiku/ISJ2001.html>
- ACM/IEEE-CSソフトウェアエンジニアリングの倫理綱領
<http://www.acm.org/serving/se/code.htm>
(あるいは<http://www.computer.org/tab/seprof/code.htm>も同じ内容)
- History: Joint IEEE Computer Society and ACM Steering Committee for the Establishment of Software Engineering as a Profession
<http://www.acm.org/serving/se/History.htm>
- Annual Report-1998: Joint IEEE Computer Society and ACM Steering Committee for the Establishment of Software Engineering as a Profession
<http://www.acm.org/serving/se/Rpt9811.htm>
- ACM Panel on Professional Licensing in Software Engineering Report to Council (May 15,1999)
http://www.acm.org/serving/se_policy/report.html
- A Summary of the ACM Position on Software Engineering as a Licensed Engineering Profession Final Version(July 17,2000)
http://www.acm.org/serving/se_policy/selep_main.html
- An Assessment of Software Engineering Body of Knowledge Efforts
http://www.acm.org/serving/se_policy/bok_assesment.pdf
- ACM Task Force on Licensing of Software Engineers Working on Safety-Critical Software
http://www.acm.org/serving/se_policy/safety_critical.pdf
- IEEE:CSDPの認定制度及び試験の案内 <http://www.computer.org/certification/>
- 情報処理学会の倫理綱領 <http://www.ipsj.or.jp/03somu/ipsjcode/ipsjcode.html>
- 日本技術士会の技術士倫理要綱 <http://www.engineer.or.jp/gijutsusi/rinri.html>

※URLの確認は2007年5月30日.