広島修道大学大学院　学位審査論文

ソフトウェア技術者: プロの精神と職業倫理

～ソフトウェアの開発に関する諸問題と高等教育におけるソフトウェア技術者の育成

大場　充

2015/02/23

ソフトウェア技術者: プロの精神と職業倫理
～ソフトウェアの開発に関する諸問題と高等教育におけるソフトウェア技術者の育成

まえがき

「たくみ(匠)」とはその道の専門家のことであり、社会からその能力を認められた人々である。「こころ(心)」とはひととして何を大切にし、何に力を注ぐかを決めるための考え方や決断の能力を言う。「しごと(仕事)」とは単なる肉体労働を意味する経済的行為を言うのではなく、行為者の人生を意味のあるものにするための経済活動である。「みち(道)」とは、仕事をする人々が何を意識すべきか、何を忘れてはならないのかについて経験から学んだ知恵を整理した結果、確立した安定的な行動原理のことである。本論文では、これからの時代を生きるソフトウェア技術者が、現代の匠としてどう生きるべきかを、彼ら、彼女らが直面している根本的な問題を紐解きながら議論する。

本論文の内容は、有名なマックス・ヴェーバーの「プロテスタンティズムの倫理と資本主義の精神」を参考にしたものである。同書は、20 世紀の初頭に、社会学者マックス・ヴェーバーが、資本主義国家として著しい発展を遂げた、オランダ、イギリス、ドイツ、アメリカ合衆国の諸国が、例外なくプロテスタント(新教)国であったことから、資本主義とプロテスタンティズム(新教徒の規律)との間に、必然的な関係が見出されることを論じたものである。著者は同書に倣って、米国におけるソフトウェアの発展と、ソフトウェア技術者育成の成果が、技術者の倫理観と専門家教育で叩き込まれるプロとしてのストイックな生き方にあることを示すことを狙う。さらに、日本のソフトウェア技術者にも技術者としての社会的責任を明確に認識し、技術者として恥じることのない生き方を通すことで、日本のソフトウェア技術力の向上を願い、示唆したいと考える。

著者が、日本の次世代を担う後輩のソフトウェア技術者に対して、「匠の心と仕事人の道」を主題として、博士論文を書きたいと考え始めたのは、2005 年頃のことであった。当時、著者は我が国に導入することが検討されていたアクレディテーション制度の情報分野、特にソフトウェア分野の教育分野からの専門委員として委員会の議論に参画していた。アクレディテーションとは、米国などで実績のある、大学における工学系学部教育の質を保証する認定制度である。この制度によって、大学の学部を卒業した技術者が、社会から要求されている知識や実践能力を備えていることを保証するのである。

このソフトウェア技術者育成教育認定制度は、基本的にコンピュータ科学やソフトウェア工学の基本カリキュラム(案)に準じた教育が実施され、産業界も納得できる単位修得基準に基づいて単位認定が行われ、結果としてその学部学科を卒業した人材が、社会が要求する水準に達した人材であることを検証し、それが確認された場合、当該学部学科の認定を行うものである。著者はこの議論から、米国のソフトウェア技術者教育には、技術者倫理の教育が必須であり、その単位を修得していない学生は、技術者としての資格が認められないことを認識した。

数年後、著者が当時勤務していた大学の学部教育に技術者倫理の科目が導入され、偶然にも著者にその講義科目を担当することとなった。講義の準備に取り掛かると、いく

つかの新しい情報を得た。その一つは、米国における技術者倫理教育の一般的な内容が分かってきたことである。また、他大学でどのような分野を専門とした教員が、技術者倫理の教育を担当しているかも分かってきた。さらに、これからの日本で、どのような倫理的な問題が起こる可能性があるのかも、分かってきた。

まず、米国の大学における標準的な技術者倫理に関する教育の内容である。これについては、まず技術者の倫理憲章から説明するものが多かった。つぎに、ソフトウェアの場合、ソフトウェア特有の知的財産権に関する法律の解説、さらにソフトウェアの受発注に関する倫理的な問題の解説、そして技術者の社会的責任と企業の利益追求の姿勢が矛盾を生じる典型的な事例の議論とその対応についての解説などである。最近では、これに企業倫理憲章の説明や、内部統制、コンプライアンスなどについても解説されている。総じて、企業内研修のマニュアルに沿った教育内容と言える。

第二に、各大学の工学系学部で実際に技術者倫理を担当されている教員の専門分野が何であるかの問題があった。技術者倫理の講義を担当されている教員が、定期的に参加する研修に参加する機会があり、その時、周囲の教員に尋ねた結果である。大きく 2 種類に分類することができた。最初のグループは、特定の工学系科目を専門とする教員の方が、技術者倫理を学び、その教育を担当している例であった。著者自身もこのグループに属していた。もう一つのグループは、哲学を専門として研究されている教員が、技術者倫理の教育を担当されている例であった。著者の調べたところでは、この哲学を専門とされる方々が技術者倫理の講義を担当されている事例が、もう一方を少し上回っていた。

最後にこれからの日本で発生する問題として、学生達が知っておかなければならない事柄がある。それは、開発者としての技術者の社会的責任の拡大の問題、日本における専門家人材育成システムが機能不全に陥っている問題、そして最後の問題は、日本のビジネス実践法が経済のグローバル化に伴い、合理性を失いつつあることである。これらの問題は、我が国の企業と同様に、技術者達にも、これまでとは異なる新しいシステムの導入と対応を要求する。その新しいシステムに対応できない企業や個人は、産業市場や労働市場において淘汰される。

第一の問題は、ソフトウェアに関する技術上の問題で、専門家と非専門家との間の知識のギャップが特に大きいため、結果としてユーザが専門技術者の裁量に任せざるを得ない問題が多くなる。これは、全ての技術一般に言えることであるが、我々が五感で感じることのできないソフトウェアでは特に問題になる。第二の問題は、専門家育成の問題である。現在の日本におけるソフトウェア技術者の育成は、大学での専門に関係なく、大学卒業者を一括採用し、企業内研修で専門知識を教育し、OJT で専門家に育てるというシステムが前提になっている。このシステムは、経済のグローバル化の進展に伴って、有効性を失っている。最後は、日本国内における商習慣に基づいた様々なビジネス規範

が、経済のグローバル化の影響を受け、国際的には通用しないものになりつつあることである。

以上のような背景から、著者は、日本におけるソフトウェア技術者の技術者倫理教育では、受講者である学生達が、これからソフトウェア技術者としてどう生きなければならないのかを教育すべきであると考えた。すなわち、技術者の社会的責任、技術者にとっての仕事と人生、米国と日本における技術者人材の育成の違い、米国と日本におけるソフトウェア技術者の採用プロセスの違いとキャリア形成の違い、専門家として責任を持つべき仕事の質の概念の歴史的変遷である。さらに、これらのことを学ぶ基礎として、倫理とは何かを教える必要がある。

倫理に対応する**やまと**言葉はない。つまり、古代の日本人は、倫理に近い概念を持たなかったと言う結論になる。漢語には、徳と言う倫理に近い概念の言葉がある。ただ、漢語の徳は支配者に要求される精神の状態であり、全ての人間がもつべき普遍的な精神の状態ではない。倫理の語源は、ギリシャ語のエチカである。この語が英語の ethics に変ったのである。倫理と言う語は、明治に入ってから、日本で作られた和製の漢語である。倫理の概念に近いやまと言葉としては、「**いきかた**(生き方)」や「**ひと**の**みち**(人の道)」がある。しかしこれらの言葉には「人として善を為す」と言う意味が含まれていない。これは、日本人が基本的に問題を相対化して考える傾向が強いためであろう。つまり、倫理には、絶対的な善の追求と言う、もともとの日本人が知らなかった問題が含まれているのである。

本論文においては、第一にソフトウェア技術者として「常に善を為し」、「善い生を全うする」ためには、どう生きるべきかを考えられる人材を育てることを目標にしている。さらにその実践において重要な、「質の良い仕事をし」、「高い品質のソフトウェアを作る」ために、「高い品質とは何を意味するか」について考える。この「高い品質」の概念は、時代と社会によって少しずつ変わってきている。これからも少しずつ変わるであろう。

グローバル化する経済の中で、ソフトウェア技術者の生き方も、変わらざるを得ない。ただし、ソフトウェア技術者にとっての「善い生を全うし」、素晴らしい人生を全うすることがどのようなことであるかは、ギリシャ時代から現代まで変わっておらず、これからも変わらないであろう。

目次

第1章　序章

第１節　背景とねらい

1970 年代に始まったマイクロエレクトロニクス革命は、コンピュータの製造コストを著しく低減させた。このことによって、従来は複雑な機械装置や電気・電子装置としてしか実現できなかった様々な製品を、比較的単純な機械装置や電気・電子装置と、それらの外部装置から入力信号を受け取り、決められた計算手順に基づいて計算を実行し、その計算結果に基づいて外部装置へ出力信号を送り出して、それらの装置を意図通りに制御することで機能を実現する、新しい製品実現方法が確立された。

この、センサ、制御、アクチュエータの 3 つの独立した構成要素によって機能を実現する新しい方法の確立により、製品の設計と製造は、著しく簡単化された。すなわち、製品の製造コストは、大きく低減されたのである。また、このことによって製造プロセスにおいて、熟練工による作業はほとんど必要がなくなったため、製品の製造と設計を、地理的に全く異なる場所と人々で実施することが可能になった。すなわち、製造は労働コストの低い国で実施することが容易になった。

その一方で、単純な構成の機械装置や電気・電子装置から入力信号を受け、それらの装置に出力信号を送り出す制御装置は、LSI 技術で安価に実現できるようになったマイクロコンピュータと、その上で稼働するソフトウェアで構成される。マイクロコンピュータは、汎用的なものを開発することが可能であるが、個々の製品によって異なる動作を実現しなければならないソフトウェアは、製品ごとに異なるものを開発しなければならない。このような背景から、製品開発において最も大きな投資を必要とする部分は、従来のような装置の設計・製造から、ソフトウェアの開発に変った。

このような組込みソフトウェアの応用の初期においては、組込みソフトウェアの利用は限定的で、従来の機械装置や電気・電子装置で実現されていた機能と等価な機能を実現するものであった。しかし、1990 年代後半からの組込みソフトウェアの応用では、従来の機械装置や電気・電子装置では実現不可能であった機能を、ソフトウェアの利用によって実現するようになった。その典型が、学習機能の製品への組込みである。これによって、個々の製品の使い易さは、従来の製品と比較すると格段に向上した。

このようなソフトウェアによる機能の実現は、従来の実現方法にはなかった問題をもつ。それは、物理的な機構にはなかった「非決定性」の問題である。非決定性は、ソフトウェアがその計算過程で変数に記憶されている過去の値を参照できる能力があるために生じる。過去の計算結果の値を参照できることによって、上述した学習機能の実現が可能になる。しかし、そのような過去の計算結果の値を、計算過程で参照できることは、同じ入力に対して、異なる出力が生じることを意味する。

この非決定性の問題が製品開発に導入されたことによって、従来の製品開発や製品製造において、重要かつ有効な手段であったテストはほとんど機能しなくなった。テストは、

ある入力に対して、特定な出力が出されることを確認することで、実現されたソフトウェアが適切に稼働することを確認するものである。しかし、非決定性は、同じ入力に対して、ときとして同じ出力を生じることがないため、テストでは機能の実現が適切であったかどうかを確認できない。

このような技術的な背景から、これからの時代に生きるソフトウェア技術者には、「適切に機能するソフトウェアを作る」社会的責任が強く要請される。例えば、自動車の制動装置を制御するソフトウェアが、特定の条件で、従来の機械式の制動装置とは異なる動作をすることがあるとすると、そのような制動装置を搭載した自動車の運転者は、知らず知らずのうちに、事故のリスクを負っていることになる。そのような事故を未然に防止するためには、そのような組込みソフトウェアの開発を担当する技術者は、非決定性に起因した問題があることを明確に認識し、製品の利用者が想定する製品の動作と現実の製品の動作との乖離問題が生じないような対策を講じなければならない。

これからの時代のソフトウェア技術者に求められているのは、上述したような問題の本質を深く理解する知識の教育と、そのような認識に基づき、製品の設計や実現に最善を尽くそうとする技術者の倫理観である。そのような倫理観は、ごく自然に技術者達の身につくものではない。ギリシャ時代の哲学者たちが主張したように、そのような倫理観をもち、なおかつ必要な状況において、その倫理観に基づいた行動を実践できるようになるためには、十分な教育訓練が必要となる。

本論文においては、経済のグローバル化が進展する社会的な状況の中で、我が国の産業を担う技術者として生きるソフトウェア技術者をどう教育し、どう育成してゆくべきかについて、高等教育制度、企業における専門家雇用制度、産業界における人材育成制度、国家の技術開発政策の視点から考える。さらに、その議論に基づき、提言を行うことを目的とする。本論文におけるいくつかの提言が、これからの日本の産業の発展に寄与することを願う。

第 2 節　論文の構成

本論文は、大きく 2 つの部分から構成される。前半は、ソフトウェアの開発に関する諸問題に関する認識を明確にし、論点を整理する。特に、これからの社会においてソフトウェアが何をもたらし、どのような根源的な問題を内包しているかの認識を明確にする。さらに、その問題を解決するためには、その開発に関わる技術者は、社会から何を求められるかについて議論する。

本論文の後半においては、前半で議論した個々の問題に対する解決策を模索するため、我が国の社会やソフトウェア産業がどのような特徴をもつかについて、歴史的な視点からの分析を試みたうえで、対策について提言を行う。この時、単に我が国のソフトウェア産業だけを見るのではなく、経済史や世界史の流れの中で、世界がどのような状態に向かっているかについても分析する。そのようなトレンド分析から得られる歴史の文脈に基づき、我が国の行政、大学、そして技術者個人が、どのような変革に向かって歩むべきかについて論じる。

本論文の第 2 章においては、マイクロエレクトロニクス革命によって、プロセッサやメモリの価格が急激に低下した結果、様々な製品に「組込みソフトウェア」が応用され、製品の構造が単純化され、その製造も容易になったため、低価格で製品が生産できるようになった。その反面、従来の製品にはなかった非決定性と言う、ソフトウェアのみがもつ特性のため、従来の製品の開発においては、品質保証の重要な手段であったテストが、十分に機能しなくなった。このような背景から、ソフトウェアの設計者の製品品質に対する責任は、従来製品よりも重いものになりつつある現実について議論する。

ソフトウェアの特性と開発に関わる問題については、1970 年代の半ばにソフトウェア工学が確立され、研究が進められてきた。当初の研究対象は、メーンフレーム・コンピュータと呼ばれる大型計算機上で稼働する大規模ソフトウェアに関する問題を研究することが主眼であった。ソフトウェア工学の研究における成果の一つとして、1970 年代後半に提起された、レーマンによるソフトウェアタイプの理論がある。このソフトウェアタイプの理論によれば、長期の利用に供するソフトウェアは、進化型の e タイプ・ソフトウェアとして、保守性と移植性のための設計に関する特別な注意が必要となる。

1990 年代までの組込みソフトウェアは、そのような e タイプのソフトウェアではなく、仕様が変化しない s タイプのソフトウェアとして開発され、利用されていた。このタイプのソフトウェアは、ハードウェアの変更に弱く、一般的にハードウェア固定のソフトウェアとして開発され、ROM(Read Only Memory)に書込まれて利用される例が多かった。したがって、一般的に利用開始後に保守のためにソフトウェアを変更することは、全く仮定していない。小規模で、実現する機能仕様が単純である場合、ハードウェアが固定していれば、s タイプのソフトウェアとして開発することは可能であるし、合理的である。この場合、機能が単純なことから、非決定性が導入されているにも関わらず、その影響が限定的であ

り、テストもかなり有効な検証手段となる。

1990 年代に入って、マイクロプロセッサの価格性能比が格段に向上したことから、様々な製品に組込みソフトウェアが応用されるようになった。このことは、類似の機能が様々な製品で実現される状況を生み出した。したがって、開発企業側は、組込みソフトウェアの開発コストを低減する目的から、同じ機能を実現するために、先行製品のために開発されたソフトウェアの部分を、部品として他の製品のための組込みソフトウェアの開発にも流用するようになった。

このソフトウェアの流用は、非決定性の影響を大きくし、流用した組込みソフトウェアの設計によっては、予想できなかった問題を引き起こす原因となる事例も出現し始めた。特に、組込みソフトウェアの開発規模が、数十万ステップを超えるようになった 1990 年代後半には、複数の製品開発において、同時に類似の原因で性質の異なる問題を発生する例が出始めていた。これは、1970 年代のソフトウェア工学が、メーンフレーム・ソフトウェア開発の問題として、研究していた根問題と本質的に同じである。

第 3 章においては、これまでソフトウェアがどのように開発され、どのような仕組みで流通してきたのかについて、歴史的な変遷を振り返る。ソフトウェアは、最初、コンピュータ・ハードウェアを販売するための手段として開発され、附属物として供給されていた。しかし、米国政府が IBM 社を独占禁止法で訴えたため、IBM 社はソフトウェアをハードウェアからは独立した製品として販売する方式を採用した。アンバンドリングである。

アンバンドリング以後、ソフトウェアを独立した製品として、市場を通して供給することが一般化した。現在、市販のソフトウェアとして良く知られたマイクロソフト社のウィンドウズやオフィスは、そのような商品としてのソフトウェアの代表と言える。さらに、1980 年代の初頭に提案されたオープンなソフトウェアが、様々な経緯を経て、1980 年代の末にオープンソース・ソフトウェア(OSS)として、社会的に認知されるようになった。

1970 年代初頭に、コンピュータ市場をほぼ独占していた米国 IBM 社は、米国司法省による独占禁止法裁判の提訴により、企業分割を迫られていた。この企業分割は、同社が独占状態にあったメーンフレーム市場に製品を供給する大型機部門と、同社が市場開拓に着手していた将来の成長が期待された小型機部門を、別々の企業にするものであった。この分割案は、大型機部門で得た収入を、まだ採算を確保できていなかった小型機部門の主要製品の開発に投入できなくなると言う視点から、同社にとっては避けるべき企業分割であった。

IBM 社は、当時はコンピュータ・ハードウェアの付属物としてユーザに提供されていたソフトウェアを、ハードウェアから独立させた製品として販売する方針を採用することにより、IBM の強みの一つであったソフトウェアを独立した製品とすることで、他社のハードウェアで稼働することを可能にし、IBM の独占状態が永続的に続く産業構造の転換を提案した。

このアンバンドリング政策を実施するためには、当時、ハードウェアの引き渡しと同時にユーザに提供されていたソフトウェアのソースコードを、秘匿しなければならない問題を生み出した。IBM 社は、ソースコードを秘匿し、ソフトウェアの実行形式であるオブジェクトコードのみをユーザに有償で引き渡す方法を選択し、アンバンドリングを実現した。そして、著作権法によるソースコードの保護の戦略を採用した。しかし、これはあくまでソフトウェアの表現の保護であり、ソフトウェアの設計に関する基本的なアイデアを保護するものではない。このことから、ソフトウェアに関する知的財産権保護の問題が、この時生まれた。

1980 年代に入って、大学の研究者を中心に、ソフトウェアのソースコードを秘匿して、オブジェクトコードのコピーのみを製品として提供し、莫大な利潤を得ることが、倫理的に許される行為かどうかに関する議論が提起された。ソースコードを秘匿することで、そのソフトウェアがどれほど質の高いものかどうかは、第三者は知ることができない。このことは、ソフトウェアの質の低下を招く危険がある。開発を担当する技術者達の職業倫理にも影響される。

このような議論を背景に、1980 年代の末に、オープンソース・ソフトウェアと名付けられた、新しいソフトウェアの供給が提案された。これは、技術者達が自由に参画するグループでソフトウェアを開発し、開発したソフトウェアのソースコードを無償で公開する方法である。1960 年代のソフトウェアがそうであったように、ソースコードが公開されている場合、開発者ではなかった他の技術者達が、そのソースコードに何らかの問題を発見すると、自分の能力を使ってその問題を解決し、新しいソースコードを作り、他の人々に対して公開することができる。

これは、人類が文明の誕生から 1 万年以上にわたって実践してきた文明の進歩の方法をソフトウェアにも適用し、より良いソフトウェアを開発するための強力な方法となることが期待できる。そのため、ライセンスという概念が導入された。ライセンスは、従来の著作権のような排他的・独占的知的財産権保護とは異なり、他者による自由な配布、他者によるソースコードの改変、さらにそのソフトウェアを活用した新しいソフトウェアの開発などを許す。これによって、ソフトウェアの進化を支援し、利用者の評価による自然淘汰によって、より良いソフトウェアを社会が選択することを可能とするのである。

現在、ソフトウェアの供給には、前述したオブジェクトコードのコピー利用を販売する著作権を利用した流通方式と、ソースコードを公開し、自由な利用と変更を可能とするオープンソース・ソフトウェア方式が利用されている。この 2 つの方式には、それぞれ個別の問題があるが、特にこれからの情報化が進展した社会で、次世代を担う世代の育成を、可能な限り平等に実施しようとする場合など、オープンソース・ソフトウェアの社会的な役割は大きい。

第 4 章においては、倫理と職業倫理の歴史的変遷について議論する。倫理は、ギリシャ

哲学がその議論の主題とした重要なテーマの一つである。特にソクラテスは、ギリシャの市民としてギリシャの倫理観に沿った生き方をすることが、重要であることを訴えた。つまり、人間として「善を為す」ことの重要性を説いたのである。ソクラテスにとっての善は、ギリシャ人にとっての善であり、ギリシャ人として尊敬される人間が持つ、様々な特徴をもった行為である。ソクラテスは、そのような行為は、教えられなければ理解できず、長期の訓練が無ければ、非日常的な場面で実践することができないとした。

ソクラテスやプラトンにとって重要だったことは、ギリシャ人として、どのような場面においても自分がすべきと信じることを行うことであった。それは、自分の生死を懸けてでも、実践しなければならないことである。プラトンは、人間のイデアを考え、そのイデアが実践すると考えられる行為を実践できるとき、「人は善を為す」とした。

これに対して、マケドニア出身のアリストテレスは、人が生まれながらにして持っている自然な価値観に基づき、人が何を成すべきかの目的(テロス)に沿った手段としての行為を、実践の状況をよく認識したうえで、冷静に分析した結果、極端ではない行為を選択する(中庸)ことによって、人は善を為せるとした。ここで、ソクラテスやプラトンが、法的な問題を倫理観から除外したのに対して、アリストテレスは法的な問題や社会的な価値観も考慮したうえで目的を達成するうえで最も効果的な行為を実践することが善であるとした。

ヨーロッパ中世社会の最初の 1,000 年間は、ローマ教会の倫理観が全てであった。人は、神のために生きるのであって、人が生きるために神がいるのではなかった。人は、神に祈るために田畑を耕し、作物を作る。余剰な作物が収穫された場合は、それを教会に寄進する。もちろん、凶作で人が植えるような状況では、教会は蓄積した余剰の作物を人々に分け与える。この結果、1,000 年間、経済は発展しなかった。

16 世紀のヨーロッパで、宗教改革が起きた。ドイツのルターは、当時のドイツで開発された印刷技術を活用し、ドイツ語に翻訳された聖書を印刷して安価で庶民に配布した。これによって、庶民の知的水準は大幅に向上した。イギリスにおいても似たようなことが起きた。ルターは、聖書に人が富を蓄積してはならないとは書いてないことを説き、一生懸命に働くことが庶民にとっては神に仕える道のひとつであると説いた。特に、当時のドイツ語にあった「天職」と言う概念を使って、天職を全うすることが、庶民にとって神に感謝することであり、祈ることと同じように重要なことを説いた。イギリスでは、哲学者のロックが国王の統治のための権力の行使と、人民の守られるべき権利について議論した。特に、民主主義の原則として、人民の生命の保障、人民の私有財産の保全、そして人民の信教の自由の保障を基本的な人民の権利とした。

また、イギリスでは、スイスの宗教改革者であるカルバンの影響を強く受けた清教徒(ピューリタン)が、人間として一生懸命に働くこと自体が徳であるという教えを重視した。これら新教徒達の仕事に対する考え方は、ローマ教会が教えた人間の生き方とは大きく異なり、仕事をすることが良いことであるとする新しい考え方に基づいていた。ただし、新教徒達も、「浪費は悪徳である」と言う価値観では一致していた。この「一生懸命に働くこと

が善である」とする倫理観により、オランダ、イギリス、ドイツ、アメリカ合衆国などの新教国では、産業革命が進展し、経済の著しい発展を遂げたとしたのが、ドイツの社会学者であるマックス・ヴェーバーの説である。

20 世紀は、産業化社会の時代であった。就業人口の半分が工場で、製品の大量生産に従事した。その労働は、産業革命の時代とは異なり、分業化と標準化の進んだ、パターン化されたルーチン業務であった。工場の中には、肉体労働だけでなく、事務系の作業も多く、そのような定常業務(ルーチンワーク)に従事していた作業者も多かった。いずれにしても、人間が手を動かして機械を操作したり、計算機械を動かすような作業が主体であった。

そのような時代の労働者の多くは、ブルーカラーと呼ばれる肉体を使った仕事を主体とした労働に従事する人々であった。その数は、非常に多く、多いときには労働人口の半分以上を占めていた。この人々に要求された能力は、簡単な読み書き、計算ができることと、機械を操作するための四肢の運動が可能であることであった。学力であれば、中学校卒業程度で十分であった。この階層に属する人々が、中流階層を構成した。

中流階層の倫理観は、就業時間中は与えられた仕事を間違いなく、しっかりとこなし、就業時間を終われば、次の日の労働のために自分の時間をゆったりと過ごすことであった。独立した家庭を営むには、自分たちの経済基盤を確立しなければならない。そのために、中流階層に属する人々は、一生懸命に働いた。自分たちが働き、より多くの収入を得ることが、家族全体の繁栄のために重要だった。1900 年代の前半は、まだ、産業革命社会の名残が強く残っており、倫理観は宗教の影響を強く受けていた。特に、米国社会では、よく働くことが善であり、浪費は悪であった。

1900 年代の最後の 25 年間は、産業化社会からポスト資本主義社会への転換の時代であった。労働について言えば、先進国でもそれまでの肉体労働的な仕事が多く残っていた。ただし、ゆっくりではあったが、先進国においては少しずつ、知的労働の割合が増加しつつあった。また、コンピュータの導入によって、単純事務作業のコンピュータ処理が可能となり、工場などで働く事務作業者が激減した。

社会におけるコンピュータの普及に伴い、ソフトウェアの開発に従事する労働者の割合も増加した。ソフトウェア開発に従事する労働者は、それまでの肉体労働者とは違って、頭脳だけで仕事をすることができる。工場のような作業環境で、工場の労働者のように時間管理されて働く必要もない。このような背景から、ソフトウェア技術者の職業倫理観は、従来の中流階層の倫理観からは大きく変わった。

頭脳労働では、長時間働くことが多くのアウトプットを保証するものではない。逆に、極めて短い時間の労働で、普通の能力の人間がある程度の時間を要する仕事と同程度の成果を出せることもある。このように、就業時間と成果との因果関係が、単純ではなくなったことが、その労働の管理を難しくした。米国社会においては、専門家によって実施される仕事の成果の質をどう管理するかが問題にとなった。専門家の仕事の質は、専門家でなければ評価できない。しかし、仕事を依頼する、または仕事の成果を利用する者は、一般

の人々(非専門家)である。専門家が故意に手を抜き、質の悪い成果の仕事をしてもそれを見破ることは困難で、それが原因で損失を被るのは、一般市民である場合が多い。

このような背景から、米国社会では、専門家に対して厳しい規律を守る倫理観を要請すべきとする社会的な環境が醸成された。具体的には、専門家が受ける高等教育の内容に、職業倫理に関する教育プログラムを入れ、専門家が専門家の社会的役割と責任をしっかりと認識するように育成することを要請した。特に、専門家の場合、他の専門家が実施した作業の成果が、妥当なものであるかどうかを第三者の目で検証するレビューを、自分の社会的責任と自覚して、しっかりと実践することを訓練することなどが重視されるようになった。

第 5 章では、質の良い仕事の本質を議論する基礎として、品質概念の歴史的変遷について議論する。人類の歴史において質の概念を最初に議論したのは、ギリシャの哲学者プラトンであったと言われる。プラトンは、「どのようにある」を意味するギリシャ語の形容詞ポイオテースから、名詞ポイオンを作ったと伝えられている。このポイオンは、アリストテレスの形而上学において詳細に議論された。それは、物事の本質を議論するために、量の議論だけでは不十分だったからである。

アリストテレスの質は、今日の言葉で表現すれば、「物事の存在するあり様」と言う意味で、強いて翻訳すれば「様態」となる。アリストテレスは、ものごとを量の面から議論する場合と、質の面から議論する場合と、問題によって使い分けている。これは、それまでの哲学者になかった議論の方法である。そして、そして、リストテレスは、量が極端に増加することで、問題の質的側面が変化することについても議論している。

人類の歴史上で、アリストテレスの後に「質」の議論をしたのは、イギリスの哲学者で、経済学の父とされるアダム・スミスである。スミスは、財の供給量と市場で交換される貨幣の量について、主に議論した。しかし、財の流通を考える時、財(製品)の質を議論の枠外に放置することはできない。同じ財であっても、質が異なれば、同じ価格での交換はできない。良い質の財の方が高価で取引され、悪い質の財の方が低い価格で取引される。

アダム・スミスは、そのような財の質に関する議論のため、効用と言う概念を用いた。効用とは、財の購入者がその財から受ける便益のことである。しかし、この効用理論は、マルクスがその理論的な根拠としたにも拘らず、その実践的な問題から、経済の主要な問題ではなくなっていった。実践的な問題とは、効用を測定することが困難であると言う問題である。

19 世紀のドイツとオーストリアでは、商品学が生まれ、研究された。この学問は、商品が売り買いされる公正な値段を議論するための学問として始まった。商品学者たちが考えた方法は、商品(財)を構成する材料を詳細に分析し、それらがどのような物質かを特定し、さらにそれらの含有量を詳細に定量分析することである。この分析結果を統合し、それぞれの材料の相場価格を乗じて足し合わせれば、商品の適正な価格を求めることが可能であ

るとした。この理論は、物質的には成立するが、今日の経営学で言う付加価値をどう評価するかについては全く議論していなかった。そこで、商品学者たちは、上述したものを交換価値とし、実際の商品を利用することで消費者が得る便益を使用価値と名付けて議論した。この使用価値は、アダム・スミスの効用概念と基本的に同じものである。

20 世紀のアメリカ合衆国では、フレデリック・テイラーの科学的管理法によって、工場労働の標準化と標準時間の導入が進み、製品の大量生産に関するシステム的な考え方が進歩した。しかし、いかに製品を低コストで大量に生産できても、税産された製品の質が悪ければ、利益は生み出さない。このことに気付いたシューハートは、初等統計を利用することで、生産している製品が利益を生み出す可能性がある製品かどうかを判断できることを示した。これが統計的品質管理の始まりである。

シューハートは、製品の品質(quality)を、生産されている製品が製品の仕様に規定されている属性に適合する確率と定義した。これで、アダム・スミス以来、人類が課題としてきた財の質を科学的に計測する方法が確立したのである。さらにシューハートは、平均値と標準偏差を使うことで、生産ラインが正常な生産を行っているか、異常な状態にあるのかを判断するための管理図の基本的なアイデアを提案した。これによって、古典的な経験主義的品質管理の方法が確立した。

シューハートと共同研究を実施していた同じベル研究所のデミングは、晩年、シューハートが強い興味を抱いていたプラグマティズムの哲学の影響を受けた。それは、生産工程のデータを収集し、分析し、そのデータに変化が生じていればその原因を究明し、もし良品率を低下させるような原因が見つかれば、その原因を除去する方法を考え、その新しい生産方法を適用して生産を行うことで、生産プロセスを継続的に改善する方法であった。

このデミングの考え方は、従来の考え方が製品に焦点を当てていたのに対して、それを生み出す生産プロセスに焦点を当てたものであった。この視点の切り替えは、その後の米国における品質管理に大きな転機となった。もし、生産プロセスの途中で、最終工程の試験で不良と判断される可能性が高い製品であれば、そり以上生産ラインに乗せて生産を継続する必要はない。このことから、生産ラインの最終段階で、製品が良品であるか不良品であるかを判定するよりも、各工程の作業中に、不良と判断される可能性の高い半完成品は、生産ラインから除外する方が合理的である。品質保証の考え方は、そのようなプロセス視点から、生産中の製品の品質を予測することが可能であるとの理解に基づいて提案された。

これまでの品質の議論は、品質は生産中の製品がその仕様に適合する確率であるとの前提に立っている。しかし、20 世紀の後半に入って、産業化社会が成熟期に移行すると、そのような原始的な品質の定義では、消費者の意向を満足することが不可能になって来た。

この問題について、日本の品質管理は重要な問題提起をした。それは、従来の品質概念が、「当たり前の品質」に属するものであり、今後の時代において製品の売れ行きを左右するのは、「魅力的な品質」に属するものである考え方である。従来の品質では、製品の仕様

書に記述された内容が全てであった。しかし、ここで問題にしていることは、むしろ仕様書には記述されていないことでも、ユーザが重要視している項目があることであった。

20 世紀末から、世界経済は著しい速さでグローバル化を進展させている。この時代の経済学として特徴的なものは、自由な市場における競争を基本原則とした経済である。特に、フリードマンらによるリバタリアニズム経済学が有名である。その理論では、国家などの公的な権力は、可能な限り市場への介入を避けるべきであるとする。このような思想の影響を受けて、1990 年代から顧客満足度を理想的な品質の尺度と考える人々が増加した。単純化して言えば、顧客満足度の高い製品が市場で評価され、最も売れる商品になると言う考え方である。この顧客満足度を品質とする考え方は、一見、合理性が高いように見えるが、市場で最も数多く売れる製品が、最も品質の良い製品と言うわけではない。それは、市場の製品に対する評価が一過性であり、決して長期的な視点で製品を吟味した結果ではないためである。

第 6 章においては、技術者の雇用制度と育成のシステムについて議論する。現在の日本の雇用制度は、第 2 次世界大戦が始まる直前の昭和初期に、当時の政府の主導によって確立された終身雇用を基礎としている。それは、労働の流動性を限定し、安定した労働力の確保を可能とすることを目的としていた。この頃、日本の労働市場は、大量の労働力を必要とした肉体労働の従事者、特に熟練工と呼ばれる経験豊かな技能工が、特定の企業に定着せず、それ以前の親方制度の名残から企業から企業へと渡り歩く例が多かった。

この政府の施策は、企業における勤続年数によって給与が昇給する年功賃金に支えられて、徐々に定着した。さらに、日本企業では、ホワイトカラー職の従業員と、ブルーカラー職の従業員を明確に区別せず、同じような条件・待遇で雇用することを特色としていた。このことから、米国におけるホワイトカラーのエクゼンプション制度のような頭脳労働を主とした高等教育を受けた従業員の処遇制度は、官僚組織以外では導入されなかった。

多くの企業で、多数採用された高等学校卒業生と、少数の大学卒業者を、極端に区別しない雇用制度は、高校卒業で入社した従業員でも、能力に応じて社内教育を受けさせることで、大学卒業で入社した社員と同等のキャリアを歩むことを可能にしている。これは、社内の特定の業務に従事するための要件として、特定の高等教育を前提としない日本の雇用制度に特有なシステムである。

現代の日本企業における雇用制度は、新入社員一括採用、年功序列(年功賃金)、終身雇用を基本としている。新入社員一括採用は、全ての新規採用者を毎年 4 月 1 日に採用し、それ以外の時期には新入正社員の採用を行わないことを言う。ここで、新規採用の対象となるのは、高校や大学をその 3 月に卒業したばかりの人材に限られる。唯一の例外が、大学院の修了者達である。このため、一般的に他の企業での就業経験のある人材の採用は実施していない。

日本企業における年功序列制度は、従業員の処遇である昇進と給与の昇給において、入

社からの当該企業での就業経験の長さが、最も重要な要素であることを意味する。つまり、ライン管理職としての昇進の道を歩むとき、現場の作業者として業務に就いていなければならない期間は、ほぼ一定の期間が決っており、さらに係長としての勤務期間、課長としての勤務期間、さらに部長補佐としての勤務期間、部長としての勤務期間がある程度、決まっている。

この間、給与も職位と勤務年限に応じて一定の範囲内で支給される。これによって、昇進が遅れた社員においても、先に昇進した社員との給与の格差が顕著にならないように調整されている例が多い。また、ライン管理職としての道を歩まず、技術系専門職としての道を歩む従業員に対しても、給与の面での処遇はライン管理者としての道を歩んでいる従業員と著しい差が生じないようになっている。つまり、給与の面ではかなり厳格に、年功給与制度が適用されていると言える。

最後に日本企業の終身雇用は、新入社員として 4 月 1 日付けで企業に入社した人材は、最初に集合社内教育を受け、その後、長期の新入社員研修を受ける。この間、新入社員は、社内の様々な業務に必要な基礎知識を学び、さらに場合によっては現場の実践研修を経験する。製造系の企業の場合には、工場における作業実習を課されることもある。その場合の指導者は、将来、研修している新入社員の監督・指示を受けるかもしれない現場の労働者である。このような、社内の様々な現場での作業に従事することによって、現場の仕事の難しさや問題点を理解する。

新入社員研修を終了した人材は、人事部門が作成した人材育成計画に基づき、所属部門が決定され、その所属部門での仕事に従事する。大学を卒業した人材の場合、一定期間は配属された部門に留まるが、長期的には他の部門へ配置転換となる。そのような配置転換も、人事部門の人材育成計画に則って実施される。この配置転換は、官僚の人事異動と似ている。このようにして、同一企業内での移動を続けながら、少しずつ昇進を繰り返してゆく。

上述したような日本の雇用制度に対して、米国の雇用制度は、職務記述書に基づく採用、業務の遂行、そして人事評価が実施されるシステムである。採用に際しては、採用すべき人材が従事する仕事の内容の記述を基本に、従業員としての採用を希望している人材と企業との間に合意を形成する。この時、当該の仕事を遂行するために必要な資格や知識、そして職務経験なども職務記述書に定義される。この求める人材が持つべき資格、知識、職務経験から、給与の幅が決定される。

企業内においては、人材を必要としている部門から人事部門にそのような職務記述書と必要な人員数などが伝えられる。この情報に基づいて、人事部門ではどの労働市場から人材を獲得するべきかを判断し、最終的な給与の上限下限を確定する。その上で人事部門は、当該労働市場に求人情報を開示する。職を求めている人材は、公開された求人情報に基づいて、自分がその要件に該当するかどうかを判断し、適格だと判断すれば応募書類を提出する。

求人企業の担当者と応募者は、面接試験(interview)の機会を設定し、面接を実施する。この面接では、応募者の資質の評価などが実施されるとともに、雇用条件についての交渉も実施される。勤務時間、仕事の量、勤務場所、給与などが議論され、両者が合意できる点が探される。合意が成立しなければ、不採用になり、合意が成立すれば、応募者の経歴調査などに移行する。経歴調査は、応募者の履歴書と推薦書にもとづき、問い合わせをする対象者を決定し、応募者の過去の実績などについて直接、聴き取り調査を実施する。これらの情報を総合して、最終的な採否を決定する。

この職務記述書に基づく採用では、職務を遂行できる能力をもつ人材に限定して、採用を実施するので、採用後(入社後)の研修は必要としない。この仕事の遂行能力を定義しているものが、職務記述書の資格と知識に関する記述である。資格の代表的なものが、高等教育で得た知識とその知識の獲得を証明する文書(卒業証書など)である。また最近では、非営利の組織が実施している資格認定試験の合格を証明する文書なども資格を証明するもののひとつとなる。

知識については、当該の知識を得るために受けた教育の受講証明等が参照される。つまり、高等教育を終了したことを証明する文書と、高等教育機関等で何を学んだかを証明する文書が重要になる。このような人材採用システムが確立した米国社会では、高等教育機関においてどのような教育を実施しているか、さらにどのような条件に適合した受講者に単位を授与しているかが問題になる。このため、米国の大学、特に工学系の学部学科においては、標準カリキュラムに基づいた教育プログラムの実施が求められる。

さらに、各科目に対して、どのような条件に適合した受講者に単位を授与するかを明確に記述することが求められる。そのような単位授与基準を厳格に順守して、単位を授与し、一定の条件を満足した学生に対して卒業証書または学士の学位を授与していると認定された大学の学部学科に対しては、認証が与えられている。そのような認証システムをアクレディテーション(accreditation)と呼ぶ。この認証を取得していない大学の特定学部・学科を卒業した人材に対しては、上述した職務記述書の資格部分に記載された条件が満足されたとは言わない。

米国企業においては、職務記述書に記載された内容に基づいて、その内容に適した人材であるかどうかが判断される。そのようなプロセスを通して採用された専門家人材は、一般にエクゼンプトと呼ばれる。それは、定年制の無い終身雇用である。ただし、一生その企業で働くと言うことではない。なぜならば、職務記述書に基づいて審査され、採用された人材は、当該の職務を遂行する能力を認められただけで、社内の他の業務を遂行できると認められたわけではないからである。

同一の職務に従事している限り、基本的に給与は大きく変わらない。処遇も変わらない。本人がそれを望んでいる場合は別として、より高い給与求めている場合は、より難易度の高い業務に従事しなければならない。そのためには、その新しい業務の職務記述書に記載された資格や知識を獲得しなければならない。このため、米国企業に勤務する従業員は、

より高度な資格を得るために、企業を退職して、夜間の講義を受講するなどし、新しい資格や知識の獲得に努める。このような背景から、専門的な職務に従事する従業員の本人希望に基づく退職はかなりの頻度で発生する。

上述したように、日本企業における雇用制度と、米国企業における雇用制度には、大きな違いがある。さらに重要なことは、この社会制度上の違いを支えているのは、大学教育システムであるという現実である。単純に表面的なむエクゼンプション制度だけを導入しても、アクレディテーション制度が機能していないことや、終身雇用制度のために大学教育と入社後に従事する仕事が直接的に結びつかない社会では、雇用の流動性や、知的労働の生産性は改善できない。

第 7 章では、資本主義社会と仕事、そして職業倫理の関係について議論する。アダム・スミスが議論した資本主義社会においては、その社会の構成員である人々の高い倫理観が維持されることを前提としていた。市場における公平な競争は、外部の権力や、神の力によって維持されるわけではない。その市場に参加している人間に求められている高い倫理観の実践によって、市場の公平性が維持され、公平な市場が維持されるからこそ、競争が成立するのである。この市場に参画する人間の倫理観を無視して、競争だけに焦点を当ててしまうと、貪欲な人間たちによって戦われる市場の競争は、単なる弱肉強食でしかなくなる。

ここで、市場と言うのは、単に財やサービスと貨幣が交換される場を言っているわけではない。市場の概念には、財やサービスを企画する人々の活動、財やサービスの実現を考える人々の活動、財やサービスを現実に実現する人々の活動、さらに実現された財やサービスを販売する人々の活動も含まれる。また、市場の概念には、地域性や財やサービスを消費する人々が所属する社会階層なども含まれる。資本主義が高度に進展した現代社会では、財やサービスの市場への供給者としての企業の経営者に対して、単に当該企業の利益追求だけを考えるのではなく、より良い社会の実現を模索する倫理観が求められる。このことは、産業化社会から変わっていない。今後、重要になるのは、経営者層だけでなく、企業の第一線で仕事に従事する全ての専門家にも、高い倫理観が求められることである。

仕事と人間の関係を歴史的に概観し、人間にとっての仕事の意味について考察すると、仕事が一部の人間にとって重要な意味を持ち始めたのは、文明の発達によって人間が余剰生産物を蓄積できるようになってからのことである。余剰生産物を蓄積できるようになったことで、一部の人間が他の多くの人間を統率し、より効率的な社会を形成することが可能になった。社会的な分業の誕生と、支配階層の分化の始まりである。特に、支配階層の人間たちにとって、彼らが従事した仕事は、社会的な役割であったと同時に、それ以上に彼ら自身がその過程と成果から多くのものを得、人生の充実感を味わうものであった。

ギリシャ時代からローマ時代にかけての古代社会では、知的な労働に従事する人間、将軍として兵を統率する役割を担った人間、王や議員として国を支配する役割を担った人間

にとって、彼らの仕事は時として彼らを悩ませるものでもあったが、成功した場合には大きな達成感を得ることができる行為であった。多くの場合、そのような仕事は血統によって引き継がれていたが、そのような環境の中で、自分の才能を発揮できた者は、幸福感を味わったに違いない。

西ローマ帝国が滅亡した後に形成された中世ヨーロッパ時代、ローマ教会の聖職者として布教に従事した者、研究に従事した僧侶達、さらに各地を支配していた貴族や王たちにとって、彼らの社会的責任を全うすることは、彼らに与えられた役割であったと同時に、やはりその成功によって彼らが得たものは、一般の人々が日々の生活から得るものに比較して、はるかに大きなものであった。ただ、ギリシャ・ローマ時代の社会と、中世ヨーロッパ時代の社会の間に、本質的な違いはほとんどなかった。

16 世紀に北ヨーロッパに起こった宗教改革の波は、オランダ、ドイツ、イギリスなどの国々に社会的な改革を起こした。さらに、オランダは旧宗主国であるスペインを戦争で倒し、世界の大帝国に成長した。イギリスでは、国王と人民との関係を整理し、より合理的な政治体制を確立する努力が行われた。このような環境下で、宗教改革は少しずつ進んだ。宗教改革が終わり、オランダ、ドイツ、イギリスが新教国の道を歩み始めた頃、私有財産の保全が人々の権利として認められ、資本主義が生まれた。

資本主義が進んだオランダでは、先行投資によって完成した技術に対する所有権を保護するための制度として発明権を法的に認める動きが始まった。ドイツでは、グーテンベルグによる印刷技術の開発によって、大量のコピーを自動的に生産する手段が確立し、音楽や絵画、文学、報道の分野で技術革新が起こり始めていた。これに伴って、著作権を法的に保護する動きも始まった。人々の生活水準が著しく向上し始めたのである。

17 世紀に始まった知の革命によって、科学技術は大きく進歩した。これは、英国における経験主義哲学が、科学的事実とは何かを、明確に定義したことが基礎となっていた。実験的な科学の方法が確立したのである。しかし、人々の心は、宗教改革の時代とそれほど変わっていなかった。ドイツに急進的な考え方が生まれてはいたが、一般的には広い意味でのキリスト教の倫理観を基礎とした価値観が社会を動かしていた。アダム・スミスもそのような倫理観を基礎として、資本主義経済を議論した。

18 世紀になって、産業革命が進み、イギリスの経済は大きく発展した。特に、時計の大量生産は時計を人々のものとし、標準時間によって人々の行動を律する、資本主義的な傾向が一般化した。この時代の人々にとって、一生懸命に働き、富を得て、それを蓄積することで、より裕福になることは、良い生き方であるとする考え方が一般化した。ただし、中世以来の思想を引き継ぎ、浪費は悪徳とされた。そのような、歴史的な背景の中で、アメリカ合衆国が誕生した。

19 世紀は、アメリカ合衆国が経済的に大きく発展する時代であった。アメリカは、イギリスの植民地として開拓されたが、その主たる役割を担った人々は、カルバン派の流れをくみ、イギリスの清教徒革命以後、社会的に弾圧されていた清教徒たちであった。彼らは、

カルバン派の教えを守り、勤勉に働くことが人の道だとする強い倫理観を持っていた。また、浪費を悪徳として極端に嫌う人々でもあった。

このような「勤勉働く」ことを心情とする人々は、大量生産された時計によって時間による仕事の管理が容易になった社会で、合理的に仕事をする姿勢を確立した。すなわち、生産性を務めて高めるように働くことが善であるとする価値観である。このような人々を中心とした社会では、産業革命を通して、入植当初の農業国から、軽機械生産を中心とする産業化社会へと変身し、大きく経済を発展させた。特に、大量生産方式の確立において、米国産業社会の世界に対する貢献は大きかった。

20 世紀に入っても米国の産業化社会は、加速度的に進行した。特に、フレデリック・テイラーの科学的管理法の提唱によって、大量生産において分業化された作業を標準化し、さらにその標準時間を計測することによって、生産工程の生産性を厳密に管理することが可能になった。このことに対して、米国社会で生まれ、労働者の力を結集して企業経営者に立ち向かおうとした労働組合は、過度な労働を労働者に強制するものであるとして、反対運動を展開し始めた。しかし、長期的な視点で見れば、生産性の向上は、米国経済の力強さを支えるものであり、経済の発展に寄与するものであった。

20 世紀の後半に入った社会で、人間関係論に基づく理論が、重視されるようになった。このことは、工場における労働が、20 世紀の最初の半世紀と、20 世紀の後半では質的変化を遂げていた可能性を示している。すなわち、20 世紀前半の工場労働が、極めて単純化された作業の繰り返しを要求する肉体的労働であったのに対して、20 世紀後半に入った産業化社会における工場労働は、人間の判断能力も活用した、知的活動を含んだ肉体的労働に変っていたと言える。知的な作業を要求する限り、人間には「やる気」がなければ、正しい作業を実践し続けることはできない。

このことから、単に勤勉であることを要求する倫理観から、仕事が作業者にとって有意義なものであるべきとの認識に基づき、自分の役割に対する認識に対して、自分が適確な仕事をすることで社会的責任を全うしていると感じることが重要になっていた。ここに、仕事が単に収入を得るための経済的な活動だけではなく、自分が社会の一員として社会に貢献していることを実感するための手段となっていることを指摘できる。

知的労働の占める割合が増加するこれからの先進国のポスト資本主義社会では、仕事における肉体労働的側面はほとんどなくなると言える。労働コストの高い先進諸国の労働力を使って、肉体労働的側面の多い仕事を実施することは、経済合理性がないからである。そのような知的労働の実践においては、人類の歴史の初期に、特別な階層に属する人間だけが感じられた満足感や達成感を、個々の専門家が感じるように自分の人生をかけて仕事に従事することが要求されるようになる。

第 8 章では、資本主義社会において公正な企業間競争を促進するための国家の役割について議論する。特に、ソフトウェアに代表される純粋に論理的かつ知的な財の知的財産権

の保護の問題と、その保護のあるべき姿を中心に述べる。

従来から、ソフトウェアの知的財産権は、第一義的に著作権で保護すると言う考え方が世界的にも一般的である。しかし、著作権は 16 世紀の活版印刷技術の開発によって、知的生産物の大量のコピーとその配布が可能になったことと、その出版物を最初に作成するために投入される労力に比較して、その複製の製作と配布が極端に容易であるために起こる、複製品の生産・配布を防止するための法的制度である。

このため、著作権が本来保護する対象としているのは、著作物の表現を完全に真似た複製品の生産と配布・販売である。最近では、著作物の表現を完全にまねたものでなくても、人間に区別ができないような精度で真似た複製物の生産と販売も違法とされる例が一般的である。この著作権をソフトウェアの重要な要素であるプログラムに適用する時問題となるのは、類似性をどう判断するかの方法である。

仮に、プログラムを機械語に変換した結果のオブジェクトコードの表現を保護の対象とすれば、同じソースコードから 2 つの異なるコンパイラを利用して生成されるソースコードの表現は、異なる可能性が高い。人間はソースコードを書いている例が多いので、このような行為は盗作と言える行為であるにも拘らず、類似の製品ではないと判断される。

ソースコードに基づいて類似性を判定する時、オブジェクトコードのみが存在する状況においては、そもそもソースコードの類似性を判断すること自体が困難である。仮に、オブジェクトコードからソースコードを逆変換できたとしても、変数名やリテラル名など、ソースコードに特有の情報を復元することはできない。また、コンパイラによって最適化されたオブジェクトコードから、ソースコード上のステートメントの並びを完全に復元することも困難である。したがって、現状で 2 つのプログラムの類似性を判定することは容易でない。

さらに、同じソースコードを参照して作成した類似のプログラムであっても、何を計算したいのかを理解し、元々のプログラムとは異なる方法で計算を実行するように工夫すれば、一見、2 つのプログラムは、全く異なるもののように見える。このように、プログラムの設計を問題にするか、表現されたプログラムだけを問題にするかによって、どのような知的財産権が守れるかが変わる。

特許権は、著作権と異なり、表現上の類似性を問題にするのではなく、実現のアイデアの類似性を問題にする。このため、似たようなアイデアに基づいていても、用途が完全に異なる場合には、異なる発明と認識されることが多い。しかし、上述したような同じソースコードを参照して、その一部分を異なる計算方法で実現しても、基本的なアイデアは、変わらないので、特許権の侵害と判定される可能性は高い。このように、知的財産権の根源であるアイデアを保護すると言う目的のためには、特許権が著作権に勝っている。

しかし特許権は、元来物理的な性質を利用して特定の機能を実現する方法に関するアイデアを保護する目的で、オランダにおいて導入された法制度であり、産業革命後のイギリスで発展した制度である。このため、世界における特許の概念には、計算方法を主体とし

て構成される製品実現法は、含まれていなかった。特に、数学的な計算方法に関するアイデアや、その応用は発明とは見なされて来なかった。世界的には、米国の特許法が特殊な例で、計算方法に関する工夫なども特許で保護する対象としてきた。

日本の特許庁も、純粋なソフトウェアで構成される処理に関するアイデアは、特許として保護する対象の範囲外としている。ただし、従来は物理的な構築物として設計され、生産されていた製品の一部分に、マイクロプロセッサを導入し、組込みソフトウェアで周辺のハードウェアを制御する製品の実現方法に関するアイデアについては、プログラム特許と言う特別な範疇を設定し、特許権を認める政策を採用している。ヨーロッパ諸国においては、ソフトウェアで実現される機能については、プログラムによって実行される計算については、数学的な計算手順の範疇であるとして、いかなる発明も特許での保護対象としてこなかった。このような各国の法律とその解釈によって、対応が異なる現状には、国境をまたいだ貿易において、さまざまな法的係争を起こす可能性があるとともに、自由な商取引を阻害する要因になることが懸念されている。

以上のように従来から存在する法的枠組みを適用してソフトウェアの知的財産権を保護しようとする考え方に対して、新しい法制度を確立し保護すべきであるとする考え方がある。そのような例として、1970 年代後半に日本で議論されたソフトウェア権に関する法制度は興味深い。この法案は、特許庁を管轄する当時の通商産業省によって検討された法律で、ソフトウェアの知的財産権の保護と、ソフトウェア産業の発展を支援する目的で検討された。

この日本の法制度案は、当時の IBM が著作権でソフトウェアの保護が可能であるとしたこと、米国政府もその IBM の戦略に沿ってソフトウェアの法的保護の政策を準備していたことなどの背景もあり、新たな日米貿易摩擦の要因とされ、法制化は断念された。また、国内においても著作権の管理を管轄していた文部省の、プログラム著作権法の提案もあって、省間の対応の差を浮き彫りにし、結果的に法制化はされなかった。

1980 年代に入って、米国においても 2 章において議論したように、ソフトウェアの健全な発展のためには、ソースコードを公開し、その自由な変更と再配布を可能とするオープンなソフトウェアが望ましいとする考え方が、大学の研究者から提起された。その後、大学を中心としてそのような無償でソースコードを配布するソフトウェアが作成され、いくつかの成功例が示された。特に、基本ソフトウェアの Linux の成功は、多くの専門家にとって驚くべき現実であった。

この成功の裏には、新しいライセンスの概念があったことは、第 2 章の議論の通りである。このライセンスの概念を応用することで、特定個人や特定企業への著作権の帰属を排除し、ソフトウェアを社会的な公の資産として、皆で管理することが可能となった。ここでは、その作成や利用に関わる人々は相互に協力し合って、ソフトウェアの健全な進化に貢献することが期待されている。また、いかなるソフトウェアも人々も排除せず、他人のいかなる貢献も阻害したりしない倫理観が求められている。

近年の市場における競争を原則とした経済システムにおいては、個人や個々のグループは、自分や自分たちが市場で勝利者として生き残ることを第一の命題としている。そのような競争原理に基づく行動規範に対して、オープンソースに参画する技術者達の行動規範は、社会全体でより良いソフトウェアに進化させるよう個々人が考えて行動することに重点を置いている。

この 2 つの行動規範は、同じ時代に生きている技術者達が従うべきとする全く相反する思想に見える。しかし現実には、特定の企業の従業員としてソフトウェアの開発に従事しながら、同時にオープンソース・ソフトウェアの開発者として自分の時間を投入している数多くの技術者がいる。国家としては、そのような技術者の新しい生き方を阻害するような法規制や雇用制度については、改革を進めるべく対応しなければならない。例えば、終身雇用制度は従業員技術者のそのような社会的活動を阻害する要因として働く可能性が高い。

第 9 章では、第 2 章から第 8 章までの議論に基づき、高等教育においてソフトウェア技術者はどう育成されるべきかを議論する。最初に、明治以降の日本の高等教育の歴史を振り返る。特に、日本の大学教育は、ヨーロッパの政治や宗教の権力から独立した、自治組織としての大学とは異なり、日本の大学は政府が国家のニーズに基づいて設立した官製大学である。その重要な任務は、ヨーロッパの大学と同じように学位を授与することである。しかし、歴史的に見れば、日本の大学は政府で働く官僚を育成するフランスのグランゼコールのような教育機関であったとも言える。

そのような背景から生まれた日本の大学は、一定の知識を獲得し、社会において専門家として特定分野の活動を担当できる人材を社会に供給することは、第一義的な使命ではなかった。産業界においても、大学卒業者としての人材に求められたものは、将来、企業の経営層で働く人材を育成することであった。そのため、人材育成の大部分は卒業後の企業内に於けるオンザジョブ・トレーニングによって実施されていた。この基本モデルが第 2 次世界大戦後の終身雇用へとつながってゆく。

したがって、大学教育の重点は、個々の分野の知識を学ばせることよりも、将来、企業の経営幹部になるための資質を養成することにあったと言えよう。ただし、一部の専門的な仕事に従事する人材の育成に特化した高等教育機関(必ずしも大学ではない)も存在した。特に、商学系、工学系、医学系の高等教育機関は、特定の専門家人材の育成だけを目的としており、高等専門学校の範疇に分類されていた。

第 2 次世界大戦後、日本の大学教育は大きく改革され、ドイツ型の 6 年制教育から、アメリカ型の 4 年制教育に変更された。当然のことながら、2 年間短縮された教育期間に対応すべく、カリキュラムも変更された。ただし、米国の 4 年制大学が、一般教養教育に重点を置いた教育機関であるのに対して、日本の大学は、旧制の大学と同様に教養教育課程と専門教育課程に分割されていた。このことは、専門教育課程が 2 年間に短縮されたことを

意味する。

専門教育において教授しなければならない知識が限定されていた時代においては、2 年間の専門教育でも、何とか社会の要請に応えることができていた。しかし、1960 年代の半ばから、各専門分野において教授すべき知識の量は急激に増加した。その膨大な量の知識を 2 年間で教育することには無理が生じていた。このため、日本企業においては、大学の専門教育に社会での実践的な仕事の遂行能力育成を期待することができなくなり、企業内における研修で、知識を教授することが必要とされた。

20 世紀末からの経済のグローバル化により、日本企業は、世界の他の国々の企業との競争に巻き込まれるようになった。その時、最も重要な資源は人間であり、人間の能力である。すなわち、日本人従業員の専門家としての能力が、他国の専門家技術者の能力よりも優れていなければ、日本企業は競争に負けることとなる。1990 年以降、日本の産業界は日本の大学に対して、即戦力人材の育成と産業界への供給を強く要請するようになった。しかしながら、要求を出している産業界の雇用制度が、従来からの新入社員一括採用、年功序列、終身雇用から変化していないため、大学としては即戦力専門家人材を育成することが、大学の生き残りに役立つかどうかの確信が持てていない。大学が大学間の競争に勝って生き残るためには、大学を卒業した人材が大企業として社会的に認知されている企業に就職することが重要となる。

しかし、日本全体の発展を考えると、長期的視野に基づいた高等教育の改革と、産業界における雇用制度の改革が実施されなければ、日本経済は停滞する。さらに、そのような改革が実施されても、その成果が出るのは、10 年後、20 年後である。特に、新しい時代に適合した教育を実施できる教員人材を育成し、そのような人材を教育の第一線に立たせなければならない。そのような新しい時代の人材育成を担う教員人材の一部には、実社会で専門家として仕事に従事した人材の獲得も重要である。実際に業務遂行のためには、形式化された知識だけでは不十分で、実践経験に基づいた経験知の教育も重要になる。

そのような専門家教育は、これまでに日本の大学の医学部において実践されてきている。その意味で医学部教育モデルは、参考になる。特に、研修医の制度は、これからの複雑化する問題解決に従事できる専門家人材を育成するためには、極めて重要な教育方法となる。また、大学病院で臨床を担当した人材が、大学教育で学生の教育に当たるシステムも、実践的な人材育成の方法として参考になる部分が多い。さらに、これからの大学における専門家人材育成においては、専門知識を教授するだけでなく、専門家として生きて行ける人材を養成するため、その資質を養成し、さらに専門技術者としての倫理観をもった人材の育成をすることが重要になる。

第 3 節　提言の要約

本論文において、著者が提言するところは、以下のとおりである。

- テストが従来のような物理的に実現されている製品のように、品質保証の重要かつ有効な手段ではない組込みソフトウェアの開発では、ソフトウェア技術者は自分たちが開発したソフトウェアが適正な設計に基づき、正しく実現れたものであることを厳密に確認する技術的能力を持たなければならない。そして、その能力を確実に発揮する倫理観を持って働らかなければならない。

- 人間が実施するソフトウェアの開発活動では必ず誤りが混入するので、既存のソフトウェアを改良して進化させながら活用しようとする、オープンソース。ソフトウェアの考え方が重要である。ソフトウェア技術者はオープンソース・ソフトウェアの社会的意義とその根幹をなすライセンスの概念を利敢為しなければならない。

- 倫理の概念は、人間が生来の認識や一般的な文化から学ぶことができるものではなく、教育によって獲得すべき高度な概念である。ソフトウェア技術者にとって、どのように生きることがその職業倫理に適うものなのかは、技術者を育成する教育訓練を通して学ばなければならない。

- 品質の良い製品を開発することは技術者の社会的責任であるが、「品質の良い」製品がどのようなものであるべきかは、時代背景によってかわりうるものである。ソフトウェア技術者は、自分が開発を担当するせいひんについて、それがどのようなものであるとき、良い品質のものであると言えるかをつねに考えながら仕事をしなければ背ならない。

- 日本の雇用制度は、専門家としてのソフトウェア技術者を処遇する制度としても、また、ソフトウェア技術者が持つ能力を社会的に活用する制度としても、不完全であり、非効率なものである。有能な技術者を社会的に活用するためには、職務記述書に基づく雇用制度を基本とするジョブ型雇用に変革し、同一労働同一賃金を実現するようにしなければならない。

- 日本の技術者達、日本企業、そして産業は、製品開発への組込みソフトウェアの導入によって、製品の高度化や低価格化の実現が可能になるものの、その開発方式の変革によって、製品の品質保証が従来型の製品に比較して、はるかに困難になることを理解しなければならせない。企業間競争の激しい市場の中で生き残るためには、

そのような製品の開発における企業倫理や専門家の職業倫理について、基本的な綱領を確立し、実践しなければならない。

- 日本政府は、このような状況に置かれている技術者、企業、産業の経済活動を支援するため、知的財産権の保護に対する戦略、ソフトウェア技術開発とその現場への技術移転のための戦略、そして技術者の育成と雇用のための制度改革の戦略を立案しなければならない。特に、次世代の技術者育成のための教育改革は、20 年後の産業のあり方を描き、立案する必要がある。

- 日本の高等教育においては、これまでの 20 世紀型の大学教育は、21 世紀の産業のニーズに適合できない。即戦力の専門家人材を育成するため、実践経験のある技術者人材も活用し、さらに、産業界の協力も得た、長期インターンシップ・プログラムの導入など、抜本的な改革が必要になる。さらに、企業側においても、従来型の専門性を問わない新入社員選抜ではなく、職務記述書に基づく専門家人材選抜に移項すべく改革すべきである。

第2章　組込みソフトウェア～その根源的問題と解決への道

第１節　マイクロエレクトロニクス革命[1]

LSI 製造技術の長足な進歩にともない、その価格性能比は、いわゆる「ムーアの法則に」(Moor's law)従い、18 カ月で 2 倍の速さで改善され続けている。歴史的には、メモリも CPU も、12 カ月から 24 カ月でほぼ 2 倍になるのが経験則である。

このため、CPU やメモリの低価格化は著しいものがあり、同じようなコンピュータであれば、ほぼ 15 年で 1,000 分の 1 の価格に低下する。このようなことから、1980 年、ローマクラブ(the club of Rome)は、「マイクロエレクトロニクス革命」(micro-electronics evolution)の到来を予測し、その社会への影響を報告書にまとめた。

その報告書によれば、LSI の低価格化によって、あらゆる製品にコンピュータを組込み、データを収集し、収集したデータを分析し、その分析結果にもとづいて、製品を動作させたりすることが容易になる。現実に 1985 年を境に、マイクロエレクトロニクス革命は、人々の身近で、さまざまな革新を巻き起こした。

そのような例の 1 つに、当時のカメラメーカのミノルタによって実現された、カメラにコンピュータを内蔵させ、焦点合わせを自動的に行う「オートフォーカス機能」があった。その後、オートフォーカス機能は、ほぼ全てのカメラ製品に導入され、その品質も年々改善され、今日に至っている。

オートフォーカス機能は、従来のカメラでは、人間がファインダの中の 2 つの像を手動で重ね合わせることで行っていた焦点合わせを、コンピュータに肩代わりさせたものである。初心者に多い「ピンボケ」映像の失敗確率を著しく低減させた。これによって、カメラを専門家のものから、皆のものへと大変身させた。

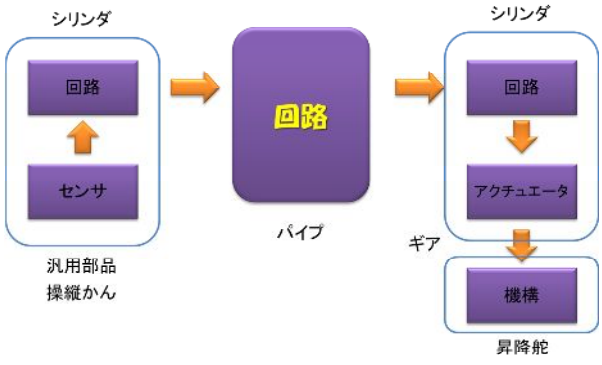

図 2.1 航空機の昇降舵の制御

今日の組込みシステムの多くがそうであるように、それは、センサの情報をコンピュー

[1] 拙著、組込みソフトウェア工学ハンドブック[21]、pp.23-25 参照

タに送込み、ソフトウェアで処理する。そのソフトウェアで最適な制御量を算出し、その制御量を出力として、コンピュータからアクチュエータへ伝達することで、高度な機能を実現する。

図 2.1 に示したのは、航空機の操縦かんが、水平尾翼に付属している昇降舵を動かす機構である。機体前方のコックピットにある操縦かんの下には、小さな油圧のピストンが接続されていて、その油圧の力で機体後方にある水平尾翼の根元におかれた大きな油圧のシリンダを動作させ、テコの力で昇降舵を上下させる。2 つのピストンの間にあるのは、油を通すための長い銅製のパイプだけである。

このような機構は、構造は単純であるが、製造には多大なコストが必要になる。特に、制御の精度を向上させるためには、個々の部品の加工精度を高めることと、質の良い材料を使うことが要求されるため、さらに生産コストが高くなる。

これに対して、機構の一部分をコンピュータに置き換え、ソフトウェアによって基本的な制御を実現することで、著しいコスト削減が可能になる。従来のように、コンピュータの価格が高かった時代には、コンピュータを使うことは、製造コストの極端な増大を招いたため、ロケットや宇宙船など、特殊なシステム以外の応用はなかった。

航空機の姿勢制御の例について言えば、エアバス社のエアバス A300 開発において、それまで宇宙船開発でしか応用されていなかったフライ・バイ・ワイヤ(fly-by-wire)方式が採用され、著しいコスト低減が実現された。これは、それまでの機械システムによる水平尾翼制御に対して、センサ=コンピュータ=アクチュエータを組合せた組込み系サブシステムによる制御で、図 2.2 に示すような構造となっている。

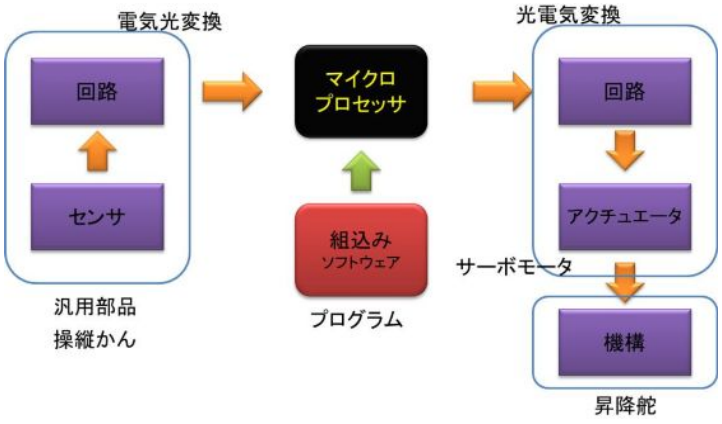

図 2.2 フライバイワイヤ制御による姿勢制御

マイクロエレクトロニクス革命によってコンピュータの価格が低減したため、操縦かん

の動きを直接、センサで電気信号に変換し、電気信号をノイズに強い光信号に変換してコンピュータへ送込み、コンピュータに搭載されたソフトウェアによって最適な制御量を計算して、再度、光信号として水平尾翼に直結した巨大なサーボモータの制御回路へ入力する。

これによって、高価な部品を全て取り除き、簡素な部品と機構による複雑な制御を可能にした。このような実現方法は、1990 年以降、多くの製品で導入され、生産コストの低減と高機能化に大きく寄与した。

第 2 節　ソフトウェア

ソフトウェアは、1950 年代末にコンピュータ製造メーカ内において、電気電子部品や機械部品から構成されるハードウェアの開発から、プログラムの作成を主体とするハードウェア製品の販売に必要不可欠な付属品の開発を、明確に分離して考える必要性があるとの認識が生じたことが起源である。この時、電気電子部品や機械部品など、物理的な要素から構成される部分に、「金物」(hardware)と言う、従来からの単語を当てはめ、それに相対する概念を「固い」(hard)の反対語である「柔らかい」(soft)を用いて、ソフトウェア(software)と言う造語をしたものである。

英語の ware(もの)は、ドイツ語でものを意味する waren から派生した語である。例えば、銀食器は、silverware と呼ばれる。また、銅製の食器は copperware と呼ばれる。このような背景から、主として金属製の生活品を総称して hardware と言う。米国の開拓時代、そのような hardware を売る店舗では、釘や金槌、のこぎりから、銃や弾丸までを売っていたとの記録がある。そのような「金属製のもの」と言う概念から、電気電子部品や機械部品から構成されるものを、コンピュータ・ハードウェアと呼ぶようになった。

ハードウェアの語源である hardware は、古くからある英語の単語であったのに対して、ソフトウェアは全く新しく考案された造語である。そしてその意味は、コンピュータ製造メーカが開発するプログラムのみを指すものであった。従って、コンピュータを利用するユーザが、その利用目的を達成するために作成するプログラム類は、ソフトウェアには含まれておらず、「プログラム」と呼ばれ、区別されていた。

ソフトウェアに含まれるプログラムとしては、例えば、オペレーティング・システム、コンパイラ、ユーティリティ・プログラム類などが含まれていた。後に、開発されたデータベース管理システムや、データ通信システムなども、ソフトウェアとして分類されていた。しかしこの分類は、整合性がないため、後に、オペレーティング・システム、ミドルウェア、アプリケーション・ソフトウェアと言う分類に変化してゆく。

ソフトウェアは、そのような時代の変化とともに、意味が少しずつ変化してゆき、現在では、「プログラムとそれに付随する文書類を総合的に指す概念」として理解されている。従って、ソフトウェアの最も重要な部分はプログラムと言うことになる。このプログラムは、アルゴリズムと呼ばれる計算(データ処理)手順と、その処理対象となるデータのもつ構造とから構成される。このことから、プログラムで最も重要な部分は、一般的には、アルゴリズムである。

新しいアリゴリズムを考案し、それに基づいてデータ構造を定義し、プログラムを作成することは、極めて高い知的な作業を要求することから、新しい電気電子部品や機械部品の開発作業に似ている。しかし、作業結果としての成果物は、電気電子部品や機械部品とは全く異なり、プログラムと言う、人工言語で記述されたデータ構造の定義と計算手順の記述に過ぎない。特に、プログラムの最も重要な要素であるアルゴリズムそのものは、自然言語で記述されることが多く、数学や物理の書物における記述と類似している。

企業でコンピュータを開発する場合、そのコンピュータの実現に応用された画期的なアイデアは、電気電子部品や機械部品として実現されるため、そのアイデアの盗用を防ぎ、開発に投入した資金の回収を可能にするため、特許権によってそのアイデアに関する知的財産権を法的に保護することができる。しかし、物理的な形式での実現を必要としない新しいアルゴリズムの開発は、同じ企業内で実施されている、似たような作業の成果であるにもかかわらず、自然の法則を応用していないため、特許権によってそのアイデアを法的に保護することはできなかった。

そのような現実は、ソフトウェアの(違法)コピーや勝手な改ざんを社会的に許す環境を生み出してしまった。1970 年代に入って、IBM はプログラムの記述そのものを著作権によって知的財産権を法的に保護することを提案して、それを企業戦略として導入したことから、他の企業もそれを追随する結果を生み出した。しかし、著作権が保護の対象としているのは、守るべきアイデアそのものではなく、アイデアの表現形式である。つまり、プログラムが似ているかどうかが問題になるのである。従って、同じアルゴリズムを全く別のプログラム言語で表現すれば、2 つのプログラムが同じアルゴリズムを基に作成されたものであるかどうかを問題にすることはできない。

また、人間の思考プロセスをプログラム言語で記述したプログラムは、人間が記述したと言う意味で、完全ではありえない。すなわち、誤りを含んでいるのである。この不完全なプログラムを著作権で法的に保護し、その記述の改善を作成者にしか許さないという著作権の制限は、そのプログラムを改善し、より完全なものにするという進化のプロセスを不可能とするため、プログラムは常に不完全なもののままで終わると言う問題が内在する。

現在の日本では、組込みソフトウェアを応用し、ハードウェアで実現すべき機能をソフトウェアで実現している場合に限り、その組込みソフトウェアの実現アルゴリズムに関する知的財産権を特許権で保護することを認めている。この場合、そのアルゴリズムの用途は、保護対象となっている製品の利用分野における用途に限定されることになる。これは、暫定的な解釈による特許法の実践であり、現実との妥協の産物であると言える。

米国では、1980 年代のプロパテント政策への変更によって、物理的な実現であるかどうかに関係なく、経済的な価値を生むアイデアであり、新規性があるアイデアには特許を認めると言う政策を採用している。このため、日本企業が開発した製品に応用されている組込みソフトウェアの知的財産権を法的に保護できなければ、日本企業が市場での競争に不利になることが、現在の日本の特許庁の考え方の基礎にある。

第 3 節　非決定性の導入[2]

コンピュータとその上で稼働するソフトウェアによる制御を導入することは、高度な制御を可能にする。例えば、学習機能を付加することは極めて容易になる。これは、従来の機械式機構や電気・電子回路による制御では、過去の動作結果に関する記憶に基づいて、将来の動作に対する制御を最適化することが困難だったためである。コンピュータの利用は、この過去の動作に関する記憶に基づいた制御を容易にする。それは、コンピュータにメモリが装備されていることによる。

コンピュータの数学的モデルであるチューリング機械がそうであるように、機械が記憶を持つことは、一般に計算に**副作用**を導入することになる。もちろん、副作用を利用しない関数型計算や論理型計算も可能である。

限定された規模のメモリで、効率よく計算を実行しようとすると、副作用を利用することが前提とならざるをえない。しかし、この副作用の利用は、理論的には複雑で解決困難な問題を導入することとなる。

副作用とは、記憶の上書き(更新)によって発生する、古い記憶の抹消が原因で発生する問題を言う。例えば、人間の記憶であれば、昨日まで記憶されていた事実に関する情報を、新しい情報で上書きしても、新しい情報と古い情報の両方が記憶に残る。これに対して、コンピュータの記憶では、古い情報は新しい情報で、完全に上書きされるため、古い情報が必要になったとしても、それを復元することは不可能である。

数学や物理の世界では、$y=f(x)$の計算は、同じ値の x に対して、常に同じ値の y を得る関数を計算することである。しかし、図 2.3 に示されるようなチューリング機械と同様な記憶をもつ機械では、このことは保証されない。

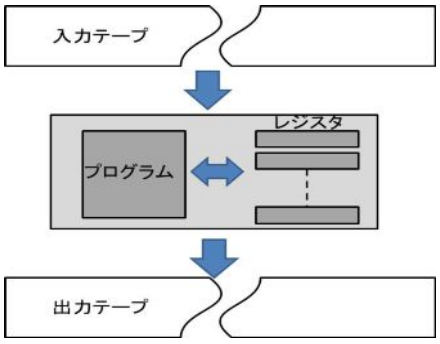

図 2.3 チューリング機械

すなわち、同じ y の値を得るためには、同じような独立変数 x の値の系列を与えなければならない。そのような独立変数 x の値の系列の長さがどれくらい必要かは、関数 f の実現で

[2]拙著、組込みソフトウェア工学ハンドブック[21]、pp.26-28 参照

x の過去の値がどこまで影響しているかで決まる。このように、同一の x に対して、常に同じ値の y が得られるかどうかが確定していない問題を、**非決定性**と言う。

つまり、変数 x も y も、時間 t を独立変数とした従属変数であり、$x(t), y(t)$ と書ける。さらに、$f(x(t))$は、$f(x(t))=(x(t)+ x(t-\triangle t))/2$ のように定義されることもある。これは、時間 t と変数 x が連続量であれば、線形な微分方程式で与えられる関係である。

この計算を、チューリング機械で実現すれば、副作用を利用した計算として実現され、非決定性をもった計算となる。つまり、現在の $f(x(t))$の値に対しては、過去 7 回分の計算で使われた変数 x の値が、$(1/2)^i$ $(i=1,2,..7)$だけ影響している。

一般的に、ソフトウェアで機能を実現することは、機械式の機構や電気・電子回路などの物理現象で機能を実現することとは異なり、そのような非決定性を導入することとなる。従来の物理現象を応用した制御系では、ある条件で動作を確認すれば、同一条件でのシステムの動作を保証できたのに対し、この非決定性が導入されることにより、さまざまな入力(条件)の組合せを確認しなければ、似たような条件下での動作でも、その保証が困難となる重大な問題に直面する。一般に、全ての条件の組合せを網羅することは時間的に困難であるため、完全な品質保証は不可能となる。

例えば、16 種類の論理変数(2 値)によって動作が決定されているソフトウェアの動作を確認するためには、最大で 65,536 通りの条件で動作を確認しなければならない。現実のソフトウェアでは、高々16 種類の論理変数で動作が決定されるような例はほとんど存在せず、その条件の組合せは数百万通り(論理変数の種類で 20 種類)を超えることが圧倒的に多い。すなわち、物理系によって実現されていたシステムでは、テストが有効に機能していたが、組込みソフトウェアを応用して実現したシステムでは非決定性が導入されたことで、テストがあまり意味を持たなくなったと言える。

第 4 節　機能安全[3]

非決定性の問題が明らかになり、さらに組込みソフトウェアを応用した製品が市場に増加したため、2000 年に安全の新しい概念である機能安全が提唱され、IEC61508 が規格として制定された。機能安全は、それまでの安全の概念とは異なり、人命や社会の脅威となる事象の発生を未然に防止するのではなく、そのような脅威が発生しても人命や社会の脅威となる事態に至る確率を、現実的な意味で十分に小さな確率(例えば、10 の 10 乗分の 1 以下)に限定しようとする考え方である。

組込みソフトウェアを応用した場合、そのソフトウェアが誤動作をする確率をゼロにすることは、理論的に不可能である。特に、そのような誤動作をする確率がゼロであることを証明することは現実的に困難な問題である。従って、そのような確率をゼロにするのではなく、そのようなリスクを管理できる範囲に閉じ込めようとする考え方が**機能安全**である。言いかえれば、実質的な安全を担保しようとする考え方と言える。

ここでは、発生が極めて稀な事象であっても、その事象が生起した場合の損失が重大であれば、そのような事象の発生を一定確率以下に抑えるべく、開発プロセスにおいて必要な対応を採るべきとしている。また、事象が生起した場合の損失はそれほど大きくなくても、発生の頻度が高ければ、発生した場合、損害の大きさが大きくなるので、同様にそのような事象の発生を一定確率以下に抑えるべきとしている。一般に、そのようなリスクは、発生頻度の確率と損失(額)の積で評価される。

IEC61508 では、実質的な安全性(機能安全)を保証するために、要求される機能安全のレベルを 4 段階に分類する。最も要求レベルの低い段階を SIL(Safety Integrity Level)1 とし、複数の人間の生命に影響を与えるような問題など、最も高い要求レベルを SIL4 とする。この 1 から 4 の SIL をシステム開発の初期に、システム概念設計段階で明確にし、システムやサブシステムに対して、各構成部分に要求される SIL を定義することが IEC61508 の実践においては要求されている。

システムの概念設計段階で設定された各サブシステム単位での SIL を達成するために、システム設計段階では、サブシステムを構成するコンポネント単位に要求される SIL を割り当て、最終的にサブシステムに統合されたときに、全体として要求される SIL を達成できるようにする。各コンポネントの実現段階では、目標とする SIL を達成するために、適切な設計が行われることや、適切な設計が行われたことを検証することが要請される。そのような検証手段ついては、SIL のレベルによって異なるが、SIL4 では、形式的(数学的)な検証の実施が推奨されている。

例えば、人命への影響から 1,000 万回に 1 回以下の確率でしか誤動作が許容できないサブシステムで、その組込みソフトウェアを応用したコンポネントの中に、1 回のサブシステムの動作に対して、平均 10 回の動作が期待されるものがあるとする。仮にこのコンポネントが当該サブシステムの機能の実現に重大な役割を担うものであるとすれば、そのコンポ

[3]拙著、組込みソフトウェア工学ハンドブック[21]、pp.28-29 参照

ネントに要求される信頼度は、平均的には 1 億回に 1 回以下の故障率でなければならない。実際問題としては、平均値の議論は、分布の 50 パーセントにしか対応しないため、分散を考慮しなければならず、1 億回に 1 回の故障確率では不十分になる。

仮に 1 億回に 1 回の故障確率であったとしても、そのようなコンポネントの故障率は、どのように保証できるであろうか。1 億回の試験を実施して、故障が起こらなければ良いと言うことにはならない。これは、試験回数を 10 億回にしてもあまり変わらない。試験を実施するコンポネントのサンプルを増やすことやテストの入力をランダムに生成することで、統計的な信頼限界を高めることはできるが、本質的な改善はできない。この水準を保証するためには、組込みソフトウェアの設計と実現に誤りがないことを示す以外に、有効な方法はない。

IEC61508 では、システムの概念設計段階で、サブシステムに対して要求される SIL から、各コンポネントに対して要求される SIL を適切に割り当てるため、FTA 分析や FMEA 分析を実施することが推奨されている。また、そのようにして決定された SIL の水準が満足されることを保証するために、設計や実現の作業完了時に、作業の成果である設計書やソースコードをレビューすることが要求されている。さらに形式手法については、特定の方法を決定しているわけではないが、Z 言語による仕様記述や、VDM による設計記述なども、有効な方法として推薦されている(Z 言語や VDM は、既に ISO/IEC 規格になっている)。

第 5 節　製品のネットワーク化とビザンチン問題

マイクロエレクトロニクス革命は、現在も進みつつある。単に、個々の製品が、センサとマイクロコンピュータ、それを動かす組込みソフトウェア、そしてその制御下で具体的な機能を実現するアクチュエータで構成された独立したシステムとして存在するのがこれまでの製品であった。カメラでも、薄型テレビでも、自動車でも、この状況は変わらなかった。しかし、今後、これらの製品に起こることは、これらの個別のシステムが、ネットワーク化され、相互に情報を交換し、個々の製品だけでは達成不可能な高度な動作を実現することである。これを製品のネットワーク化と呼ぶ。

例えば、現在、自動車のブレーキは、ABS と言う方式が採用されている。この方式のブレーキでは、運転者がブレーキペダルを踏むと、従来のブレーキシステムでは、油圧方式で、ブレーキペダルにかかった力が増幅されて、4 輪の車軸近くに取り付けられたキャリパのブレーキパッドを油圧ピストンの力で動かし、車軸に固定されているブレーキロータを挟み込む。このパッドとロータの摩擦によって車軸の回転を止めるのがブレーキの原理である。

このパッドによるロータの挟み込みによって、急ブレーキをかけると、車軸の回転が止まるため、タイヤの回転が止まり、タイヤが路面を滑る現象が発生する。これが、タイヤのロック現象である。タイヤがロックすると、タイヤは道路の路面を滑ってゆくため、十分な摩擦抵抗が得られず制動距離が長くなる。このため、急ブレーキ状態においても、パッドによるロータの挟み込みを一瞬緩め、もう一度締め付けるという動作を繰り返すことで、制動距離が短くなることが知られている。このパッドによるロータの締め付け動作をコンピュータで制御し、最適な条件で動作させ、制動距離を自動的に短くするのが ABS システムである。

この ABS システムは、高度なコンピュータ制御システムであるが、1 台の自動車の動きだけを制御する部品であり、周辺の自動車の制御には、全く影響を与えない。このことは、前後に縦走する 2 台の自動車を考えると、前を走る自動車が急ブレーキをかけた時、後続車の運転者がすぐにそれに気づいて急ブレーキ操作に入らなければ、後続車は前を走っている自動車に衝突することを意味する。両方の自動車の走行速度が速くなればなるほど、この衝突リスクは高くなる。

そのような状況でも事故を回避できるように、前を走る自動車に急ブレーキがかかった途端、前を走る自動車のコンピュータから周囲を走る自動車に対して「急制動中」の情報を通報すると、運転者が感知するよりもはるかに早く、後続車は前方走行車の制動を認知し、必要があれば急制動状態に遷移することが可能となる。このような自動車間における情報交換を可能にするためには、無線通信などを利用したコンピュータのネットワーク化が必要条件となる。現在、検討されている car2car は、そのような自動車対自動車の無線通信を利用した情報交換の技術である。

しかし、コンピュータがネットワーク化された場合、我々は分散処理の全く新しい問題

に直面せざるを得ない。その問題は、**ビザンチン(将軍)問題**と命名された問題である。ネットワーク化されたコンピュータによって構成される複合システムは、人間社会と同じ問題を持つ。コンピュータは、人間と同じように誤動作する。さらに、コンピュータは、人間と同じように故意に嘘の情報を流すこともある。いずれの場合にも、誤った情報がネットワーク上に注入される。このとき、ネットワーク化された複合システムは、ネットワーク化されていない個別のシステムの集合よりも、多大な被害を受ける可能性がある。

ビザンチン問題は、ネットワークに接続された数多くのコンピュータの１つ、または少数が、何らかの理由で正しくない情報をネットワーク上へ送信すると、その情報をネットワーク経由で受信したコンピュータは、そのコンピュータ自身は正常に稼働していたとしても、正常状態にある場合とは異なる動作を実行することになる。結果として、複合システム全体としては、正しくない動作を実行することとなり、複合システム全体としては故障状態に陥ることを言う。さらに、そのような正しくない情報を受信したコンピュータが、当該情報を他のコンピュータへも転送すると、複合システム全体は、さらに深刻な故障状態へと陥りかねない。

このビザンチン問題に対する理論的な解決策は、まだ見出されていない。基本的には、ネットワークに接続した多くのコンピュータに問い合わせを送信し、問題になっている情報、またはその情報を送信したコンピュータの信頼度を、それぞれのコンピュータがどう評価しているかを確認し、多数決によって真偽を決定する方法が合理的であるとされている。ただし、この場合でも、ネットワークに接続しているコンピュータの 3 分の 2 が、信用できるコンピュータでなければ、多数決による結論も十分に信用することはできない。

そのような新しい問題を克服するためには、マイクロコンピュータを制御する組込みソフトウェアの機能の実現は十分に信用できるものであり、誤動作の原因となる欠陥を含むものであってはならない。また、ネットワークに接続した他のマイクロコンピュータから発信された情報の真偽を適切に判断し、信頼できない情報は廃棄できるような高度な判断機能を装備し、誤った情報がネットワークを通じて広範囲に拡散することがないようにしなければならない。

第 6 節　ソフトウェア開発の本質的な問題[4]

1960 年代初頭、IBM 社で世界初の汎用オペレーティングシステムであった OS/360 開発の陣頭指揮を執ったＦ．ブルックスは、その著書「人月の神秘」（Mythical Man-Month）において、OS/360 開発プロジェクトでの経験を赤裸々に記録した彼の日誌に基づき、その開発過程で起きた様々な出来事を公表した。それは、ブルックスが IBM 社を去ってほぼ 10 年が過ぎた頃であった。ブルックスは、この失敗プロジェクトから何を学び、後輩たちに何を伝えようとしたのか。

この著書のタイトルが暗示しているのは、ソフトウェア開発の管理で重要な指標となる「人月」が、神秘的な、そして科学的には意味のない数字であることであった。ブルックスは、その神秘性が、一見正しそうであり、合理的であるかのように見えるが、実は全く不合理であることを、彼の実体験に基づき訴えている。彼は、電気電子工学を学び、南部ノースカロライナの名門デューク大学で、学士の学位を取得、ハーバード大学の大学院へ進学、エイケンの指導のもとでマークⅠの開発に参画した。

マークⅠの開発プロジェクトの完了の後、その開発に協力していた IBM 社へ入社した。後に、IBM のチーフサイエンティストに就任し、ハーバード大学の教授となったブランスカム博士の大学の同窓生でもある。そのような経歴のブルックスの目には、OS/360 のソフトウェア開発の現場で起こること全てが極めて非合理的で不条理に見えたようである。

このプロジェクトで、ブルックスは定期的に当時の社長、IBM 社の創業者の長男であるトーマス・ワトソン・ジュニアへの進捗報告を義務付けられていた。そのワトソンこそ、IBM を単なる事務機器会社からコンピュータ会社へ変身させ、世界企業にまで育てた中興の祖である。ワトソンは、ブルックスのプロジェクトが彼の会社の将来を決定付けることを直感的に理解していた。

ブルックスとトム・ワトソンとの会議はいつも暗い、重苦しい雰囲気のものであったと想像できる。ブルックスは、その会議でワトソンに対して、常にプロジェクトの遅れを報告しなければならなかった。当時、ワトソンは、決まって「何人月遅れたのか」と尋ねていたそうである。ブルックスも、プロジェクトの遅れを人月単位で、「20 人月です。」と言うように答えていたそうである。

このブルックスの回答に対してワトソンは、「それでは新たに君のプロジェクトに 30 人を投入しよう。１週間以内に任務に就けるように手配する。来月には遅れを取り戻して欲しい。」と、言っていたと伝えられている。しかし、プロジェクトに参画している技術者は増えても、遅れは取り戻せるどころか、プロジェクトはますます遅れて行ったのである。ブルックスにとっても、ワトソンにとっても、そのような経験は初めてのものであったと記録されている。

何が原因か。それは、ソフトウェアの開発がハードウェアの開発と全く違うために起こる問題であったと、後にブルックスは回想している。つまり、「20 人月の遅れ」に対して、

[4]拙著、組込みソフトウェア工学ハンドブック[21]、pp.7-13 参照

仮に、新たに 30 人が投入されても、その 30 人が、自分たちに課せられた仕事を理解し、もともと配属されていた 200 人と調和して仕事ができるようになるためには､3 か月以上を必要としたからである。

ハードウェアの開発では、各エンジニアの役割は明確であり、やらなければならないことも、仕様書の記述から明確である。これに対して、当時のソフトウェアの仕様書の書き方や設計時の決まりごと（変数の使い方や名前のつけ方などの規約）は、プロジェクトごとに決められており、標準化されていないため、新たに配属された技術者達がそれらを完全に理解するまでには時間を要する。それらは、既に成熟した技術であるハードウェア設計では、プロジェクトに関係なく、一般的な規約として定められ、教育されていた。

また、ブルックスが悩まされたもう一つの問題が、開発過程における機能仕様やその実現を規定する設計等の変更であった。仕様や設計の変更は、設計や仕様書だけにとどまらず、プログラム、テスト仕様書、保守マニュアル、そしてユーザ・マニュアルにまで影響する。さらに、同時並行して開発されている他のソフトウェアにも影響が及ぶことも少なくないのである。ハードウェアの開発では、設計が完了した時点以降の、仕様や設計の変更は、簡単に許されることがないのに対し、ソフトウェアの開発では容易に許される傾向があった。

特に､開発途中にあったハードウェア(S/360)の設計変更(Engineering Changes)への対応作業は、想像以上のものであった。ハードウェアの設計変更がソフトウェアに影響し、ソフトウェアの一部の変更が、さらにいくつかのソフトウェアの部分の変更を必要とするのである。これらの変更を矛盾なく、完全に実施するには、厳密な変更管理・構成管理の手順が必要であった。しかし、当時の IBM にあったのは、ハードウェアの技術変更のために定められた手続きのみで、ソフトウェアのそれには、不十分なものであった。

1986 年、ブルックスは、IFIP 国際会議で、「銀の弾丸はない」（No Silver Bullet*）というタイトルの基調講演を行い、ソフトウェアプロジェクトの問題として、以下の 4 点を指摘した[72]。

- **複雑性**（大規模なソフトウェアは、設計の見通しが利きにくいこと）
- **不可視性**（ソフトウェアは、物理的な性質がないこと）
- **変更容易性**（ソフトウェアは、変更が簡単にできるため、管理が難しいこと）
- **任意性**（ソフトウェアの実現には、従うべき絶対的な法則がないこと）

ここで複雑性は、大規模なソフトウェアだけに特有と言うことではなく、大規模なハードウェアにも当てはまる性質である。しかし､その他はハードウェアにはない性質であり、その分、ソフトウェアの開発はハードウェアに比較して、大きなリスクを伴うのである。

複雑性とは、開発対象の規模が増大するに従って、開発に関係する人間の理解や推論能力（因果関係を見通す力）の限界に近づくにつれて、人間には予想できない問題が発生す

る確率が増大することを言う。これは、ハードウェアや大規模な建造物を設計したり、実現したりするときにも経験することである。

模型の飛行機は確実に飛ぶものを製作することも容易であるが、人間が乗って操縦できる飛行機を設計し、実現するにはそれなりの経験が必要になる。そして、小型機でさえ開発し、生産することは困難を極める。さらに、数百人の乗客を乗せて安全に飛行する大型ジェット旅客機を開発し、生産・販売に成功するのは、世界的にも数社しかない。規模(スケール)の問題は、ものづくりにおいて、それほど根源的な問題である。

不可視性とは、文字通りの意味は、「実体が見えないこと」である。しかし、ここで言う「不可視」とは、単に見えない、または見えにくいと言うことではなく、むしろ**物理的な性質がない**ことである。物理的な性質、すなわち大きさや重さ、またはある種のエネルギ量等がないことは、科学的な測定が困難であることを意味している。

計測は、全ての工学の前提であり、計測に基づく管理こそが工学を成り立たせている基本原理である。その計測が困難になることで、開発や生産の途中段階において開発完了後のソフトウェアの性質を予測することは、ほぼ不可能になる。科学的な予想ができなければ、科学的な管理は不可能である。これによって、ソフトウェア開発のリスクは、増大する。

変更容易性とは、文字通り「いつでも簡単に変えられること」を意味する。一般的には、簡単に変えられることは、製品や商品の望ましい性質である。しかし、それはある限界内での変更のし易さを前提としている。つまり、「何とかすればいつでも変えることが可能である」と言う意味での変更可能性である。本当は、「どのような変更も可能である」ことを意味している訳ではない。

ハードウェアでは、一般に、一度設計が確定してしまうと、それを簡単に変えることは困難であるし、変えることのコストとリスクは多大である。変えたことによって、何が起きるかが容易には予測できないからである。ソフトウェアの場合の変更のし易さは、ある意味で、限界のない変更可能性である。つまり、ソフトウェアでは、完成後でもどのようにでも好きなように、ソフトウェアを変更することが可能であるかのようなことを意味している。

この変更容易性が、なぜリスク要因になるのか。それは、むしろ技術的な問題ではなく、管理的な問題である。変更を何の制約も無くできると言うことは、ソフトウェアを開発している技術者にとっては、「いつでも必要な修正ができる」ことを意味している。それは多くの場合、「実現が適切であるかどうかは、後で確認し、その時点で問題が判明した場合にのみ、修正すれば良い」ことを暗示する。

従って、「最初から適切・適正な設計を開発しなければならない（Do it first right!)」とする品質管理の原則を、ソフトウェアにおいては守らなくても良いとする解釈が成立してしまう。しかし実際には、複雑性と不可視性が絡み合うことにより、大規模なソフトウェアの場合、変更の影響を正確に予測・評価することは困難である例が多い。そのため、変

更は可能であっても、その悪影響を正確に予測できないことは、やはり最初から慎重に設計・実現しなければならず、その意味で変更容易性は、管理上のリスク要因と言える。

最後に、任意性とは、ソフトウェアの設計や実現において、技術者は矛盾を起こさない限り、「全てのことを自分の裁量で決定できる」ことを言う。すなわち、論理的な矛盾を引き起こさない限り、特定の問題をどのように解決するかは、その問題解決を担当する技術者が全ての決定権を握っている。ハードウェアの場合、いかに有能な技術者であっても、加工技術や加工精度の限界や、人間ではどうにもできない物理現象によって、どのような解決策を採用できるかは制限されている。

例えば、ハードウェア設計におけるグランドルール（基本規則）によって、最小線間距離や最小線幅等は、技術者個人が自由に決定できるものではない。また、機械式機構を設計するとき、材料の強度や摩擦の問題を無視して設計を考えることはできない。これに対して、ソフトウェアの設計においては、論理的な矛盾以外、自由な設計を阻害する要因はない。このことは、設計の自由度を増大させ、ソフトウェアの有利性を高める。

しかし任意性は、設計の自由度を増大させる反面、設計規範や設計規律の遵守を困難にする。特に数多くの技術者が開発に参画する大規模なソフトウェアの開発において、設計規律や設計規範の逸脱を許すことは、ソフトウェアの一部分が変更されたとき、その影響を予測することを困難にする原因となる。

それだけでなく、保守局面においてはそれまで正常に動作していた部分が、他の部分のわずかな変更によって、甚大な問題を発生させる誤動作を誘発する不安定な設計に変えてしまう例もある。大規模な組織によるソフトウェア開発においては、設計規範や設計規律を遵守することが、設計変更に強い安定したソフトウェアを開発する上で重要である。その意味で、任意性はソフトウェア開発のリスク要因になっていると言える。

以上のような、ブルックスが指摘した大規模ソフトウェアの 4 つの特徴は、その開発においてリスク要因として作用する。仮に、個々の技術者の能力が十分に高いとしても、多数の技術者の共同作業を要求する大規模ソフトウェア開発プロジェクトでは、技術者集団による規律を遵守した作業が、開発における失敗のリスクを低減させる。そのような技術者による規律の遵守を徹底させることを阻害する要因として、上述したソフトウェアの性質が影響すると、ブルックスは指摘したのである。

第 7 節　ソフトウェア品質とソフトウェアタイプ[5]

国際規格 ISO/IEC 9126(JIS X 0129)では、ソフトウェア品質をユーザの視点から次のように定義する。ソフトウェア品質とは、ソフトウェアがもつ性質のひとつで、ユーザが当該ソフトウェアを使用することによって、そのソースコードを見ることなしに、使用経験と提供された文書の記述のみに基づいて、認識しうるソフトウェアの良し悪しに関する属性の総合評価である。

このとき、評価の対象となるソフトウェアの属性は、以下の 6 つの**品質特性**である。

- **機能性**(合目的性、正確性、セキュリティなど)
- **信頼性**(成熟性、障害許容性、回復性)
- **使用性**(理解性、学習性、操作性)
- **効率性**(時間効率性、資源効率性)
- **保守性**(解析性、変更性、試験性、安定性)
- **移植性**(適応性、設置性、共存性、置換性)

すなわち、ソフトウェア品質は、これらの 6 つの品質特性を総合的に判断して評価されるソフトウェアの価値である。

この 6 つの品質特性の中で、機能性(functionality)、信頼性(reliability)、使用性(usability)、効率性(efficiency)の 4 つの品質特性は、当該ソフトウェアが評価時点でユーザの品質要求を満足しているかどうかを評価する属性である。これに対して、保守性(maintainability)と移植性(portability) の 2 つの品質特性は、当該ソフトウェアが評価時点でユーザの要求品質を満足しているかどうかを評価するのではなく、将来、ユーザが新たな要求を持ったときに当該ソフトウェアがその新たな要求に適確に対応できるかどうかを評価する属性である。

一般的に、前者と後者は、相対立する性質を持つことが多く、ソフトウェアの良し悪しは、これら 2 つのグループに属する品質特性のバランスによって大きく左右される。一般的には、ライフサイクルの長いソフトウェアにおいて、後者が重要となり、従来型の組込みソフトウェアなど、寿命の相対的に短いソフトウェアでは、前者が重要になる。

レーマン(M. M. Lehmann)は、ソフトウェアは大きく 2 つの種類に分割できることを指摘した。ひとつは、**s タイプ**(specification type)と呼ばれるソフトウェアであり、原子炉制御用組込みソフトウェアなどが典型とされる。もうひとつは、**e タイプ**(evolution type)と呼ばれるソフトウェアであり、一般的な情報システム用アプリケーションなどが典型とされる。

s タイプとは、開発の初期段階でソフトウェアが満足すべき機能仕様を完全に定義することが可能であり、ソフトウェアのライフサイクルにおいて動作環境が変わらないソフトウ

[5]拙著、組込みソフトウェア工学ハンドブック[21]、pp.79-83 参照

ェアを言う。逆に、e タイプとは、ソフトウェアが満足すべき機能仕様が時間とともに変化するため、開発の初期段階では完全な機能仕様を定義することが本質的に困難なものであり、ソフトウェアのライフサイクルが極端に長く、多くの場合、動作環境や入出力装置などが時間経過とともに変化するソフトウェアを言う。

一般に、**s タイプ**のソフトウェアにおいては、**機能性**、**信頼性**、**使用性**、**効率性**が、その品質を決定づける品質特性になるのに対して、**e タイプ**のソフトウェアにおいては、長期的には**保守性**と**移植性**がその品質を決定づける品質特性となる。すなわち、ソフトウェアのタイプによって、品質保証のポイントは、違ったものになる。

e タイプのソフトウェアは、一般に、時間とともにその規模が増大する傾向があるとともに、一定期間ごとに大改造が必要になることが多い。このため、そのような大改造が可能な構造であることが重要であり、そのことは保守性や移植性への要求に強く現れている。

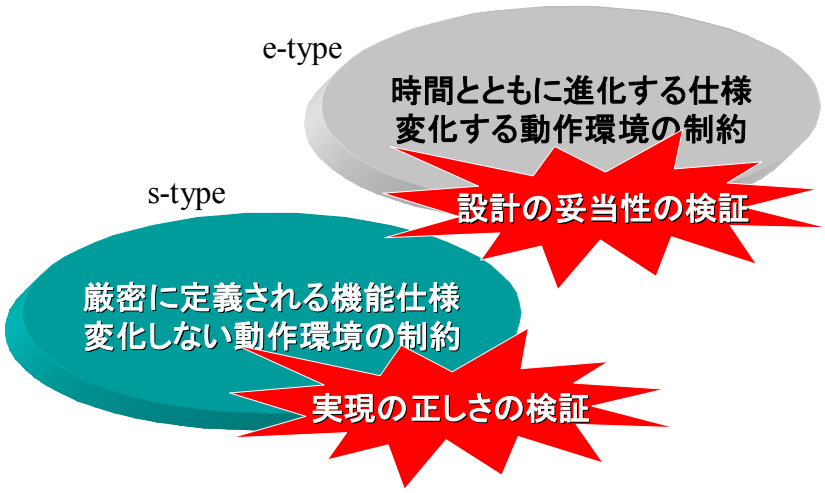

図 2.4　e タイプのソフトウェアと s タイプのソフトウェア

初期の組込みソフトウェアは、純粋に s タイプのソフトウェアであった。電話交換機用ソフトウェアや自動車の電子燃料噴射制御ソフトウェアなどが典型である。しかし、主としてその大規模化が原因で、組込みソフトウェアの多くが e タイプのソフトウェアに変質しつつある。このため、開発組織に蓄積されてきた「何が重要か」に関する知識の一部が陳腐化し、「誤った知識」になりつつある。このことが原因となって、問題を発生した事例は、この 10 年間増加の一途をたどっている。

以上のようなことから、s タイプのソフトウェアでは、実現されたソフトウェア(プログラム)の機能が、機能仕様書の記述に合致していて、要求仕様書に定義される品質要求にも合致しているかどうかが問題になる。これは、品質の保証においては、実現の正しさを確

認することが重要になると言える。これに対して、e タイプのソフトウェアでは、実現されたソフトウェアの機能や品質は、時間とともに変わってゆく性質があるため、完成時に仕様書に合致しているものであっても、それが良い品質のソフトウェアであるとは言えない。

e タイプのソフトウェアが良い品質のソフトウェアであるためには、開発完了後に発生する新たな要求に対応すべく、機能を修正したり、新しい機能を追加したり、新しい稼働環境での稼働を可能とするようにプログラムを修正することが可能であることが重要になる。そのためには、ソフトウェアの構造が単純で変更がし易いなど、「設計の良さ」(設計の妥当性)が重要になる。

我々の経験から学んだことは、現存し、良く利用されている多くのソフトウェアは、e タイプのソフトウェアであり、逆に、現存する s タイプのソフトウェアは、極めて限定されている。これは、ソフトウェアは、利用者が増加するに従って、多様な要求を反映するために規模が増大する傾向が強いからである。そして、その規模の増大に対して開発を効率的に実施するため、既存の機能を実現している部分は、可能な限り再利用しようとされる傾向がある。この再利用を可能にする設計の品質が、e タイプのソフトウェアの良し悪しを決定づけるのである。

現在、航空機に組込まれているソフトウェアや、車載組込みソフトウェア、携帯電話の組込みソフトウェアなど、数多くの分野の組込みソフトウェアが、100 万行を超える規模になっていると言われている。組込みソフトウェアも大規模化するに従って、オペレーティングシステムやデータベースシステムなどの汎用基本ソフトウェアと、仕様書に記述された具体的な機能を実現するためのアプリケーションソフトウェアに分化してきている。このような機能分化を進めることで、複雑な機能の実現を可能にしているのである。

このような開発形態を採用することは、製品開発の効率を改善するが、フレームワークと呼ばれる、製品横断的に応用される一群の基本ソフトウェアは、e タイプのソフトウェアとして、より洗練された設計が要求されるようになる。このことは、フレームワークの設計者に対する要求を高める結果になり、より高度な設計を技術者達に要求することになる。また、その品質を保証するためには、戦略的な品質保証体制の構築が必要になる。

第3章　ソフトウェアの供給メカニズム

第１節　黎明期におけるソフトウェアの供給[6]

歴史的に見れば、当初ソフトウェアは IBM 等のハードウェアメーカによって開発され、ハードウェアと一体で供給されていた。このハードウェアと一緒にソフトウェアも供給する制度をバンドリング制と呼ぶ。その典型が、IBM によるシステム 701 とモニタシステムの開発と市場への投入であった。それ以後、米国司法省によって対 IBM 独占禁止法裁判が提訴されるまで、ソフトウェアの供給は、ハードウェアとのバンドルで行われるのが商習慣として定着していた(図 3.1)。

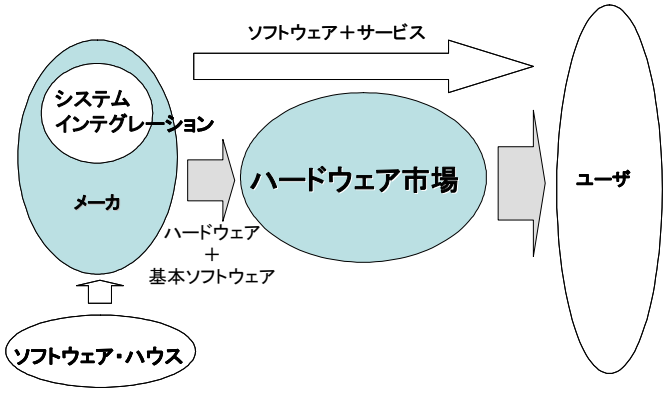

図 3.1 アンバンドリング以前のソフトウェア供給

同じような状況は、我が国においても違いはなかった。富士通や日立に代表される我が国のコンピュータハードウェアメーカは、ハードウェアと基本ソフトウェアと呼ばれる、オペレーティングシステム、コンパイラ、汎用のユーティリティプログラム類を、自社のハードウェアユーザに対して、無償で提供していた。それらのソフトウェアの中には、後に日米の貿易摩擦問題でも議論になった、IBM が開発してユーザに提供していたソフトウェアのソースコードを入手し、内容を詳細に分析・検討し、自社のハードウェア上で効率よく稼働するように改良したソフトウェアも含まれていたとされている。

この時代にも、コンピュータメーカ以外にソフトウェア開発を実施、それを業務とするソフトウェア企業が、少数であるが存在していた。米国のコンピュータ・サイエンス社(CSC)もそのような企業の一つとされている。当時、そのような企業は、メーカが人材の不足から開発を断念せざるを得なかったユーティリティプログラムやコンパイラの開発を、メーカからの発注を受けて開発を行い、納品すると言う業務形態を採用していた。

これらのソフトウェア企業の多くは、それ以前、コンピュータハードウェアをメーカか

[6]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.1-3 参照

ら購入し、そのハードウェアを時間貸しし、利用料を徴収するデータセンタサービスを主たる業務としていた企業が多かった。そのようなデータセンタサービス提供業務を経験していた企業の一部は、一部の顧客がソフトウェア開発の委託先を求めていたことを知り、プログラム開発要員を雇用し、それらの需要に応えるサービスとして事業展開を始めたことがきっかけであった。

この時代、ユーザがそれぞれの目的に合ったデータ処理の実現のために必要な応用プログラム(アプリケーション)は、基本的にユーザ企業が自分の責任で開発しなければならなかった。数多くのユーザの中には、そのようなプログラム要員を雇用する余裕のない企業も存在したことから、データセンタサービスを提供している企業が、個々のユーザの要求に応える応用プログラムを開発し、自社のコンピュータでデータ処理を行い、計算結果をユーザに納入することは、経済合理性があったと言える。

米国政府などの大規模ユーザの場合は、組織内に大規模な応用プログラム開発グループを抱えている例が多かったが、そのような大規模組織であっても、大規模な応用システムの開発となれば、大きなリスクを伴うことから、大規模ソフトウェア開発経験のあるコンピュータメーカに、開発委託を行う例があった。そのような需要に応えるため、コンピュータメーカ内には、大規模ユーザのための応用ソフトウェア開発を受注し、有料サービスとして実施する組織も存在していた。

第 2 節　OS/360 の出荷と IBM による市場の独占[7]

1960 年、IBM はそれまでのコンピュータとは異なり、規模の異なるコンピュータハードウェアであるにも拘らず、同一の命令セットで動作する一連のコンピュータシステムとしてシステム 360 シリーズのコンピュータ群を発表し、その開発に着手した。この新しい考え方に基づく一群のコンピュータは、コンピュータの機械語命令を実行するのに、従来のような回路による命令の解釈実行方式ではなく、マイクロコードと呼ばれる詳細な動きを記述するプログラムを応用し、同じ機械語命令を異なる設計のハードウェアで実行することができるようにした。

このような同じシリーズのコンピュータハードウェアであれば、同じ機械語命令で書かれたプログラムが稼働すると言うことは、ユーザが相対的に小型のハードウェアから、より大型のハードウェアに移行したとしても、応用ソフトウェアは、全く変更なく稼働させることが可能になると言う、非常に有利な側面を持っていた。このことは、企業がその事業展開の必要性から、データ処理量が増加しても、それに合わせてハードウェアを上位機種に入れ替えるだけで、継続的にデータ処理を実施できることになる。

また、システム 360 の開発とともに、IBM はこの新しいコンピュータをユーザが容易に稼働できるように、従来の 701 シリーズのコンピュータに導入したモニタシステムを大幅に拡張したオペレーティングシステムとして OS/360 を開発する計画も発表した。このオペレーティングシステムは、コンピュータを利用したデータ処理の効率を高めるため、複数のプログラムの連続処理や、複数の同時並行的実行、異なる入出力装置をあたかも同一の装置のようにプログラムから制御する機能、そして主記憶装置の割り当てをプログラムの実行時に動的に実行する機能など、さまざまな新しい機能を提供するそれまでにはなかったソフトウェアであった。

この OS/360 の開発は、当初、システム 360 の開発チームの設計者の一人であった、ブルックス(F. Brooks)が主幹技術者に任命され、開発が始まった。この大規模でまったく新しいソフトウェア開発の試みは、それまでのコンピュータシステムの開発経験では全く経験することがなかった様々な問題に直面せざるを得なかった[1]。このため、ソフトウェアの開発は困難を極め、当初の予定であった 1964 年中には開発を完了することができなかったi。

1964 年にシステム 360 の 1 号機を市場に投入するとき、IBM は急きょ OS/360 の簡易版のオペレーティングシステムを複数開発し、ユーザに提供せざるを得ない状況におかれた。1965 年に OS/360 の開発が完了し、ユーザへの提供が可能になったが、その時、すでにいくつかの簡易オペレーティングシステムが、ユーザのシステムでは稼働していた。IBM は、ユーザに対して OS/360 への移行を勧めたが、一部のユーザは現状に不満はないとして、OS/360 への移行を拒否した。

とは言え、この全く新しいソフトウェアの市場への投入は、その後のコンピュータの歴

[7]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.3-5 参照

史だけでなく、ソフトウェアの歴史を大きく変えることとなった。ユーザは、この新しいソフトウェアを利用することで、ハードウェアを著しく効率よく利用できることを知ったのである。それは、オペレーティングシステムの基本機能の 1 つであるジョブ管理機能と、タスク管理機能によるところが大きかった。

ジョブ管理機能は、ユーザがシステムに投入するデータ処理作業を、そのデータ処理に必要となる各種のコンピュータ資源(利用する主記憶域の大きさ、必要なる計算時間、利用する入出力装置の種類と数)に関する情報を事前評価し、どのデータ処理をいつ実行することが、コンピュータの稼働率を向上させるために有利かを事前評価し、データ処理の実行順序を人間に代わって決定し、コンピュータの制御を実施するものである[ii]。これによって、オペレータと呼ばれるコンピュータ操作員の作業は大きく軽減された。

タスク管理機能は、コンピュータが実行すべき 1 つのデータ処理を、可能な限り短時間に、可能な限り大量に実行するため、コンピュータの最も重要な要素である CPU(中央演算処理装置)の遊び時間を最小にすべく、複数のプログラムを同時に矛盾なく実行させるための仕掛けを提供する。そこでは、1 つのプログラムが別の複数のプログラムを起動し、それぞれのプログラムが入出力装置からのデータを受け取って処理を実行し、その結果を同時に稼働している他のプログラムへ手渡し、流れ作業で実行できるようにする。銀行のオンライン処理などでは、このような並列処理をいかに効率的に実行できるかが問題になる。

このようなオペレーティングシステムによる新しい機能の提供によって、IBM のシステム 360 は、この時代に最も多く利用されるコンピュータとなり、IBM を名実ともに最も巨大なコンピュータメーカに成長させる原動力となった。その結果、世界中で稼働するコンピュータの約 80 パーセントが、IBM 製のコンピュータと言う現実を生み出すこととなった。文字通りの市場独占である。

第 3 節　米国政府による独占禁止法裁判とソフトウェア市場の確立[8]

1970 年、米国商務省は IBM を独占禁止法に抵触したとして裁判所に提訴した。商務省と IBM との間では、それ以前からこの問題に対して継続的な議論が行われてきていた。当時､IBM 製品(コンピュータ)の市場占有率は米国市場のほぼ 80 パーセントと言われていた。

このような状態は、IBM の事業形態等とは関係なく、客観的に「独占状態にある」と認定できる。しかしながら、IBM の視点から見れば、IBM に市場を独占する意図はなく、かつ市場を独占すべく採用している経営戦略はなかったと言う主張が成立していた。

1971 年、裁判所は米国司法省の訴えを認め、IBM の分割を主張する司法省が勝訴する結果となった。この時、IBM は IBM の市場独占状態を生じさせている主たる原因が、IBM が開発した一連のソフトウェアにあると言う分析結果に基づき、対応策を検討していた。その結果、IBM のハードウェアとソフトウェアをそれぞれ独立な製品として市場へ供給することで、ユーザに選択の自由度を与え、全体として IBM 製品の市場占有率を低下できる戦略を考案し、実施に移すことを計画していた。

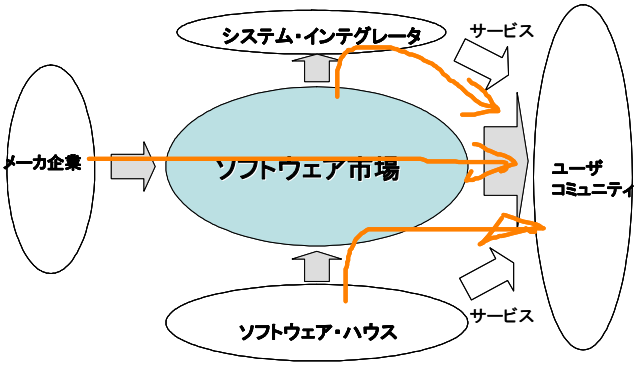

図 3.2 アンバンドリング後のソフトウェア供給

独占禁止法裁判で敗訴したことによって、IBM はその企業分割を回避するためには、自社が開発したソフトウェア製品をハードウェアから独立した、採算性のある製品として市場へ投入するアンバンドリング制度を導入せざるをえない状況に追い込まれた(図 2.2)[iii]。これによって、採算性を度外視したソフトウェア製品の開発と販売や、ハードウェア製品と同じように品質保証に関する責任が発生し、実現性の低い開発計画の発表や、無償のソフトウェア提供に歯止めがかけられ、市場における公正な競争が求められた。また、それまでの慣習であったソースコードの配布は、このことをきっかけとして停止された。

IBM のアンバンドリング制度導入により、米国市場においては、ハードウェア市場からは完全に独立した、ソフトウェア市場が形成され、コンパイラ製品の開発やライブラリ管

[8]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.5-7 参照

理システム等の開発に特化したソフトウェア企業の誕生と市場参入を促進した。さらに、1980 年代に入ると、パーソナルコンピュータ市場が誕生し、そのソフトウェア市場も誕生し、マイクロソフトなどのソフトウェア企業が大きく成長するきっかけとなった。

翻って日本の市場を見ると、現在でもソフトウェア市場は依然として成熟化していない。このことは、我が国の情報サービス産業が依然として、パッケージ開発販売よりも、個別システムの受託開発業務に大きく依存している事実に現れている[iv]。

これは、パッケージが存在しないのではなく、日本では米国 IBM の ACS やドイツ SAP の ERP パッケージ、そして米ロータスのノーツのように、汎用性の高いパッケージを独自開発して市場へ投入することが一般化しなかったことを意味する。特に日本の大手ベンダにおいては準汎用パッケージを開発し、それを人手で改造する受託ソフトウェア開発サービスを提供することが、地方公共団体を含め、日本のユーザの個々のニーズに適合していたからであろう。

そのような準汎用パッケージのカストマイゼーションを効率的に実施する組織として、日本のソフトウェア工場は誕生し、成熟化した[2]。国内人件費が相対的に安価であった 1980 年代末までは､そのようなアプリケーションソフトウェア開発は､経済合理性が高く、高い品質のソフトウェアを、比較的効率的に開発するための方式として、世界的にも高く評価された[v]。しかし、国内人件費が世界で最も高い国内で、当時と同様な方法を採用することは、高コスト構造を生み出す原因となり、合理的ではなくなった。

第 4 節　米国におけるソフトウェア市場の発展と PC の普及[9]

米国では、1971 年にインテルが世界最初の汎用マイクロプロセッサ 4004 を開発した。このマイクロプロセッサの開発は、当時、日本で開発されていた電子卓上計算機(電卓)で使われる集積回路の数を大幅に減らすため、汎用のマイクロプロセツサを必要としていたことが影響していたと言われている。

インテルが 4004 を開発すると、そのチップを使ったパーソナルコンピュータが開発され、販売されるようになった。1976 年、アップルは汎用マイクロプロセッサチップを使い、APPLE I と呼ばれるパーソナルコンピュータを発表し、販売し始めた。

1977 年、アップルは一般的な利用を想定した家庭用のパーソナルコンピュータ APPLE II を発表し、1978 年にはその APPLE II の外部記憶装置として利用できるフロッピーディスクドライブも開発した。そして、その APPLE II 上で稼働するソフトウェアとして、1979 年に表計算ソフトウェアの VisiCalc が開発され、販売された。

このパーソナルコンピュータ APPLE II と表計算ソフトウェア VisiCalc の組合せは、米国市民の数多くを悩ませていた確定申告の労力を劇的に減少させた。そのため、APPLE II の出荷台数は、1980 年に約 78,000 台、1982 年に約 30 万台と、爆発的に増加していった。

このアップルによる APPLE II の急激な普及を見て、IBM は 1981 年に、後に IBM PC と呼ばれるパーソナルコンピュータ IBM5150(IBM PC/XT)を開発し、販売を開始した。これ以前に IBM は、1975 年にも個人利用を前提とした小型のデスクトップコンピュータ IBM5100 を開発し、販売していた。

このデスクトップコンピュータは基本的なプログラム言語として、アイバーソン(K. Iverson)が開発し提唱した APL 言語を採用していたこともあり、販売台数は極めて限定されていた[vi]。これに対して、IBM PC は、CPU にインテルの汎用マイクロプロセッサである 8088 を採用し、ビル・ゲイツ(W. Gates)らが開発したマイクロソフトの MS-DOS をオペレーティングシステムとして採用したため、他社による互換機の開発も容易であった。このことも関係し、IBM PC とその互換機は、またたく間にパーソナルコンピュータ市場を席巻することができた。

1984 年に IBM は IBM5150 の後継機として IBM5170 を開発、販売を開始した。この IBM PC は、IBM PC/AT と呼ばれ、CPU にインテルの最新型マイクロプロセッサの 80286 を搭載した。この IBM PC/AT も XT と同様に、オペレーティングシステムとしてマイクロソフトの MS-DOS を採用していた。

さらに、表計算ソフトウェアとして VisiCalc に対抗して、米国ロータスは Lotus 1-2-3 を開発し、発売した。この Lotus 1-2-3 は、IBM PC/AT の性能を活かす工夫が随所に散りばめられており、計算速度で他の表計算ソフトウェアを凌駕していた。この時、マイクロソフトは、IBM からの MS-DOS の権利譲渡請求を拒み、自社ブランドでの互換機メーカへの供給や単独販売戦略を採用したと言われている。

---

[9]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.7-11 参照

1980 年代は、コンピュータの歴史上、パーソナルコンピュータがコンピュータ産業の動向を左右した時期であったと言える。アップルは、APPLE II の成功の後、IBM PC の普及に押され、市場における影響力を少しずつ失っていった。この間、IBM PC とその互換機の市場は、急激に拡大し続けていた。しかし、その市場でも、IBM PC の市場占有率は、徐々に低下してゆき、逆に互換機メーカであるコンパックの製品などが市場占有率を上昇させていった。

この間、ソフトウェアは、オペレーティングシステムはマイクロソフトの MS-DOS がほぼ市場を独占していた。表計算ソフトウェアでは Lotus 1-2-3 が、最も普及した表計算ソフトウェアであったが、数多くの競合企業も存在し、しのぎを削っていた。また、この頃から、表計算ソフトウェアだけでなく、ワードプロセッサソフトウェア、電子メールソフトウェア、さらにプレゼンテーション資料作成ソフトウェアなども開発され、販売され始めていた。

1990 年代に入って、マイクロソフトは、オペレーティングシステムを大幅に改訂し、ウィンドシステムを中心とした、マルチタスク型オペレーティングシステムに書き換える努力をしていた。その最初のシステムが MS-DOS 上で稼働する Windows3.1 であった。

その後、マイクロソフトは、Windows 系のオペレーティングシステムを次々と開発し、市場へと投入して行った。また、汎用アプリケーションソフトウェアの開発にも力を入れ、Office と呼ばれる Word、Excel、PowerPoint などを開発し、市場に投入していった。

1990 年代に入って、米国社会全体が電子情報社会化し、電子メールによる情報交換、ワードプロセッサによる文書の作成が一般化し始めていた。そのため、小学校を含めた学校教育でも、コンピュータの利用に関するリテラシ教育が注目されるようになりつつあった[vii]。

特に小学校では、不景気で教育税が削減される状況の中、学校内にパーソナルコンピュータを導入するための募金活動等が活発に実施されていた。一部の米国企業には、マッチングファンド制度が導入されており、社員が社会貢献で一定額を学校等に寄付した場合、それと同じ額をその従業員を雇用している企業も寄付をしなければならない制度ある。この制度を利用した、小学校のコンピュータ導入は、比較的裕福な家庭の多い地域では、頻繁に行われていた。

より広い視野で見ると、特に貧しい家庭の多い地域の小学校におけるコンピュータ利用のリテラシ教育のためのパーソナルコンピュータ不足は深刻であった。大企業からの寄付も、その供給量は限定されていたからである。

この問題を解決するため、様々な地域で、コンピュータ関連技術者が連携して、中古のコンピュータを企業から回収し、小学校の体育館に集約して分解し、再利用できる部品と利用できない部品に分類して、再利用できる部品を組み合わせて新しいパーソナルコンピュータを作り出すボランティア活動が盛んに行われた[viii]。そのようなボランティア技術者を雇用している企業は、本人が申し出た場合、その技術者がその活動に参加する期間、その社員は仕事に従事したものとみなし、給与を支払わなければならなかった。

パーソナルコンピュータのハードウェアは、そのようにして調達することも可能であるが、それを動かすために利用するソフトウェアについては、それほど簡単ではない。マイクロソフトの MS-DOS であっても、ロータスの Lotus 1-2-3 であっても、教育割引が適用されても、かなりの額を支払わなければ入手することはできない。

必要なソフトウェアを入手し、コンピュータに導入しなければ、コンピュータはただのゴミでしかない。ソフトウェアの技術者達は、購入したソフトウェアの導入はできても、ソフトウェアそのものの入手を助けることはできない。

このような米国社会の現実は、米国社会における貧富の格差を広げる要因としてのみ働く[3]。つまり、米国社会の活性度を維持するためには、子供たちが生まれ、育った地域に関係なく、将来、彼らが必要とするスキル(ジェネリックスキル)を獲得できる環境が整っていなければならない[ix]。

ソフトウェア技術者として、そのような問題の解決に貢献できるのは、自分達が無給で、ボランティアとして自分達が望ましいと思うソフトウェアの開発に従事し、完成したソフトウェアを無償で、そのような小学校や家庭に提供することであった。

第 5 節　フリーソフトウェア運動と OSS の誕生[10]

米国においては 1970 年代後半から、大学を中心として汎用オペレーティングシステム Unix の利用が一般化し、さらに NSF の支援でカリフォルニア大学バークレー校による改良版である BSD 版が無償配布されるに至って、さらに普及した。このとき、アテネプロジェクトで開発されたウィンドシステムや emacs なども組み込まれて配布された。

政府から BSD への支援が停止されて以降も、GNU による無償ソフトウェアの配布は、今日まで継続されている。この GNU によるソフトウェアの配布に際して導入された新しい概念が GPL のライセンスであり、従来のメーカやソフトウェア会社による有償かつオブジェクトコードのみの配布とは、正反対のあり方が存在しうることを示した。

現在、一般的にオープンソースソフトウェア(OSS)と称されるソフトウェアの多くは、この GNU のライセンスの概念を一般化した OSI のライセンス定義に基づいて、多くの場合無償で配布されている、オペレーティングシステムやデータベース管理システム等に代表される基本ソフトウェアである。

これは、特定組織に固有の問題解決を目的としたエンタープライズ系アプリケーションソフトウェアの開発と異なり、工学的に設計・開発方法が確立したソフトウェアであり、開発や保守において開発者間および開発者とユーザ間の密なコミュニケーションを絶対的な条件としないソフトウェアである。このような基本ソフトウェアの分野においては、それを必要しているユーザが多く、また、提供すべき基本的な機能がほぼ固定化しているため、OSS は特に基本ソフトウェアの分野では、今後一般化すると予想される(図 3.3)。

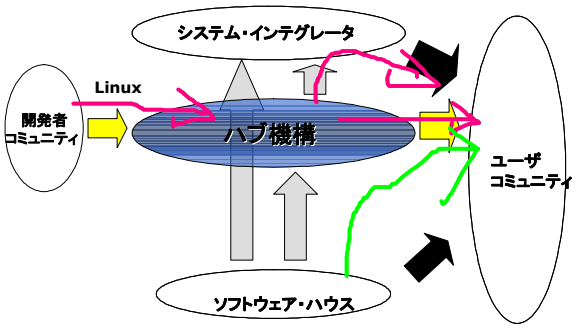

図 3.3 OSS のソフトウェア供給

このような OSS の起源は、1982 年頃に始まったフリーソフトウェア運動に始まる[4]。1982 年、マサチューセッツ工科大学のメディアラボに所属していたストールマン(R. Stallman)は、当時、ソフトウェアの供給方法として一般的であったオブジェクトコードのみの提供が、ソフトウェアの自然な発展を阻害することから、全てのソフトウェアのソー

[10]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.11-15 参照

スコードを公開し、誰でもそのソースコードを変更する自由を与え、その変更したソースコードも無償で公開することによって、ソフトウェアの自然な発展を助けるべきとする提言を、電子メールを使って広く世界に配信した[5]。

このフリーソフトウェア運動の提言は、多くの専門家の共感を受け、ソースコードの公開と無償提供を前提としたソフトウェアの開発が進められることとなった。そのような、新しい考え方に影響された基本ソフトウェアの 1 つに、フィンランドの大学院生が開発した Unix 互換のオペレーティングシステムである Linux がある。

Linux は、GNU によってソースコードが無償配布され、世界中の大学等の研究機関だけでなく、私企業、さらに公的機関等での利用が広まっている。この Linux の開発には、その後、多くのマイクロソフトの Windows 開発者達が参加するようになった。

マイクロソフトの技術者たちがボランティアとして、OSS の開発者コミュニティに積極的に参加する動機は、彼らの収入はマイクロソフトからの給与で保証されているが、彼らの技術者としての創造性は、製品としての Windows の開発では十分に発揮されることがないことである。ここに、労働としての仕事と天職としての仕事の矛盾がある。

彼らは、その矛盾を解決するため、社会貢献として自分の才能を発揮して自分が最良と思うソフトウェア設計を実践する。彼らは、数多くのユーザから認められることを自分が投入した労力への見返りとし、生きがいを感じるのである。

そのような開発者コミュニティに参加する技術者たちの期待に応えるため、OSS では開発技術者達の自尊心や名誉欲を損なわないよう、彼らがソフトウェアに対して行った貢献を明確にし、その痕跡が永続的に保持されるよう仕組みを導入している。それが、上述したライセンスである[6]。

OSS ではソフトウェアを人類が共有すべき資産として、改良を継続的に実施することを可能にするため、ソースコードの公開と、その自由な変更、さらに変更したソースコードの公開を原則としなければならない。ソースコードを公開し、その修正を可能とすることは、従来の著作権では開発者のソフトウェアに対する貢献を明確にし、その権利を守ることは不可能である[7]。

著作権は、企業が巨大な資本を投入してソフトウェアを開発し、そのソフトウェアに対する全ての独占的権利を維持することを容易にする。しかし、それは資本主義の市場における競争原理を前提とした枠組みのみに適合する仕組みである。

OSS の発展と普及のためには、従来の著作権によるソフトウェアの知的財産権の保護ではなく、全く新しい仕掛けを必要としていた。そのため、ライセンスの枠組みでは、ソースコードに全ての貢献者の痕跡を明示的に残す仕組みを導入する必要があった。これは、ソースコードが公開されているために可能なことである。

ソースコード上で、変更前のソースコードを残すことは容易である。従前のソースコードをコメントとしてソースコードに残し、新しい変更を正規のソースコードとして追加することで可能となる。このことによって、ソフトウェアの修正過程も詳細に残されること

になる。

このソフトウェアの**オープン化**のために開発された枠組みは、その対象をソフトウェアだけに限定するものではなく、知的な生産物全てに適用可能な知的財産権の保護と、その普及・流通を容易にする方法であり、知的生産物の流通を促進する。このため、インターネットを利用した電子的な流通が可能な、電子テキスト情報、電子画像、電子映像、電子化されたキャラクタ、電子的に公開される音楽などのコンテンツに対しても、ライセンス方式を適用する事例が出現している。

OSS 方式で配布される基本ソフトウェアやオフィススイーツ(マイクロソフトが販売しているマイクロソフト Office の互換ソフトウェア)の誕生によって、個人所得水準が先進諸国に比較して著しく低い低開発国の子供たちにも、100 米ドルのコストでパーソナルコンピュータと必要なソフトウェアを配布し、先進国の子供たちとの教育格差を縮めることが可能となった[x]。このことは、21 世紀に人口が爆発的に増加し、それに伴って、顕著な経済発展が期待されるアフリカ大陸の諸国における子供たちの教育に多大な影響を与えることと期待できる。

この場合、OSS の枠組みが適用される対象の範囲は、基本ソフトウェアだけでなく、オフィススイーツソフトウェア、教育コンテンツ再生ソフトウェアを含め、教育コンテンツそのものも含むようになると予想される。そのような教育コンテンツを OSS 化することで、教育コンテンツに残存しやすい軽微な誤り(誤字・脱字など)や、教育コンテンツの使いにくさ(アニメーションの起動方法など)、教育コンテンツの誤り(説明内容の間違いなど)を、それらを発見した者が修正を行い、その後でその教育コンテンツを利用する者が、より良い教育コンテンツを利用することが可能になる。

現在 OSS では、その配布に当たり、配布を受ける全てのユーザに対して知的財産権の共有に関する取り決めである「ライセンス規約」に同意し、それに準拠した行動をとることを要求する。その内容は、受取ったソースコードを活用して新しいソフトウェアを開発した場合の、新しいソフトウェアのソースコードの取り扱いに関する規定、受取ったソースコードに変更を加えてより品質の良いソフトウェアに完了した場合の、新しいソフトウェアのソースコードの取り扱いに関する規定などである。

GNU の GPL と呼ばれるライセンスの定義では、そのようにして派生したソフトウェアを開発した場合でも、その派生したソフトウェアに対して GPL のライセンスが適用されることを要求する。しかし、他のライセンス規定では、派生したソフトウェアに対しては、大元のソフトウェアに対して適用されているライセンスの適用を任意としているライセンスも存在する。

第4章　倫理と技術者倫理の歴史的変遷

第１節　ギリシャ哲学における倫理[11]

人類史上、最初に倫理と言う言葉を使い、「倫理的とはどのようなことか」を議論したのは、ギリシャの哲学者たちであった。特にソクラテスは、同時代のギリシャ市民に対して、「倫理的に生きること」の重要性を説いた[8]。それは、基本的に平等と考えられていたギリシャ市民の間で、戦いが始まれば、敵との戦いに勝つために市民たちは命を懸けて、勇敢に戦わなければならかったからである。

都市国家間の組織的な戦いでは、将軍を頂点とした軍組織をつくり、作戦に基づいて軍隊を有機的に動かすことが重要となる。将軍となった市民は、たとえ幼馴染の友人であっても、敵の目を欺くことが必要であれば、おとりの部隊を組織し、死を覚悟した行動を命令しなければならなかった。

このとき、命じられた者(兵士)は、命令に従わなければならず、命じる者(将軍)は、命令を納得させなければならない。したがって、将軍には尊敬されるべき徳が要求され、兵士には国家のために命令に従う義務が要求される。

すべてのギリシャ市民にとって戦争は、重要な仕事であった。戦争に負ければ、市民は殺されるか、自由をはく奪された奴隷になるかしかなかった。自分たちの国はなくなるのである。従って、ギリシャ市民の平時における重要な仕事は、身体を鍛えることであり、精神を磨くことであった。この精神を磨く目的で、人々はソクラテスのような哲学者に教えを乞うたのである。

ソクラテスの思想は、極端にギリシャ的であったと言われている。つまり、ソクラテスの説く倫理観は、狭い意味でのギリシャ人にのみ当てはまる、普遍性のない倫理観であったと言える。しかし、そのことはソクラテスの倫理観が、現代に生きる人々に全く無意味なものであると言うわけではない。今日においてもリーダーに求められる倫理観は、ソクラテス的な倫理観である。

ソクラテスは、徳の本質を理解することは、「正義とは何か」「真の勇気とは何か」に答えることであり、それを自分の身をもって実践することであるとした。そのためには、何が正義であり、何が真の勇気かを知らなければならない。したがって、全ての徳の基礎は知であると考えた。しかし、知っているだけでは、非日常的な状況で適切な行動がとれるわけではない。そのためには、教育と訓練の積み重ねが重要である。

自分の生死が懸かった時、人は自らの死を犠牲にした行動を選択することができない。ソクラテスは、教育と訓練によって、そのような非日常的な状況下でも、自らの死を犠牲にした行動が適切なものであれば、躊躇なく選択できるようになることを説いたのである。ソクラテスは、そのような行動の選択がなされているとき、人は「善をなしている」と言うことができ、その状態を徳と表現した(図 4.1 参照)。

---

[11]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.19-24 参照

図 4.1 ソクラテスの倫理観

ソクラテスは、そのような行動の選択における目的の適切さを問題にした。つまり「善をなすこと」を目的とした行動の選択を重視したのである。また、行動の選択においては、状況の冷静な分析に基づき、いくつかの選択可能な行動の中から最適なものを選択すべきとした。この選択が適切でなければ、善をなすことはできないのである。

人が川でおぼれかけているのを見た時、何も考えずに川に飛び込むのは、ソクラテスから言えば正しい行動ではない。仮に、自分が泳げない場合は、その行為は無意味かも知れないからである。

しかし、おぼれかけている人が子供で、川の水深が十分に浅ければ、自分でも助けられるかもしれない。そのような状況で、川に入らず、いたずらに時間を費やすことは、正しい行動とは言えない。

この例のように、ソクラテスにとって徳を実践することは、目的をよく意識し、状況を冷静に把握し、最も適切と考えられる行動を選択することであった。このような考え方は、今日においても倫理的な行動の範疇に入るものである。

しかし、ソクラテスの場合、現代的な表現を使えば、「リスクを取る」勇気を持つことも徳の実践に重要な要素となっている。この点が、ギリシャ的と言われる理由である。

倫理と言う日本語は、英語の ethics を翻訳するために明治時代に作られた和製漢語である。すなわち、もともとの日本語である「やまと言葉」には、それに近い言葉はなかったのである。さらに、中国から入ってきた漢語にも、近い意味の言葉はなかったのである。このことから、井上哲次郎が考え出した言葉であると言われている。

英語の ethics の語源は、ギリシャ語のエチカと言われている。学問の分野でもあることから、英語では複数形の形式で表現されている。現代の倫理の概念と、ギリシャ時代の倫理の概念には、若干の違いがある。ギリシャ時代の倫理は、ギリシャ市民の生き方を言うものであり、奴隷には適用されなかった。

ギリシャ語のエチカを強いてやまと言葉で表現すれば、「あるべき生き方」が最も近い表

現であろう。ただし、やまと言葉の表現は、少し抽象度が高く、その分あいまいな感じがする。

旧来の漢語で近い表現は、「徳」であろう。ただし、徳は人々の上に立つ支配者が持つべき行動の特性を言うものであり、市井の人々をも含んだ「あるべき行動の姿」を言うものではない。

現代の日本語では、「善をなす」と言う表現がある。この善を実践することを可能にする、実践的な知を倫理と考えることが最も妥当な解釈であろう。

この場合もソクラテスと同じように、「実践」が問題になっている。すなわち、単に「何をすべきか」を知っていることだけでは、倫理的とは言えない。その行為を実際に実行できるとき、「倫理的である」と言うのである。

ソクラテスに続いてプラトンは、より普遍的な哲学を追求し、イデア論を確立した。このイデア論は今日に至るまで我々の思考過程に大きな影響を与えている[xi]。

プラトンは、人間が認識する対象を個々の事物ではなく、そこから抽象されるイデア(理想像)であるとした。この思考方法はプラトンが教えを受けたピタゴラスの影響と言われる。

数学者は、我々が現実に見ている具体的な個々の点や線、三角形や円を議論するのではなく、現実世界のものを抽象して得られる面積のない点や直線、それを結合してつくることができる三角形や、点と点とを結ぶ直線の長さを極限まで小さくし、最初の点と最後の点を一つの点に集約することで得られる円などを考える。これらは、現実世界に存在するものではなく、そこから抽象することで得られた概念である。

我々が「人間」を議論するときに想定しているのは、現実に生きている個々の人間ではなく、数多くの人間の集合を考え、その人間の集合の要素に共通した性質の組合せを抽出し、それらの性質を満足する架空の存在を想定しているのである。これは、ソフトウェア工学の言葉で説明すれば、オブジェクト指向のクラス概念に置き換えることができる。我々は、インスタンスを見ているが、思考の対象は個々のインスタンスではなく、それらから抽象されるクラスである。

プラトンは、人間のイデアを考え、そのイデアとしての人間がある状況でどのような振る舞いをするかを議論したとき、そのイデアとしての人間の理想的な振る舞いを「善」とし、それを「倫理的」であるとする。このイデア上の人間の行動は、それを考える個々の人間(哲学者)によって、少しずつ違う可能性がある。従って、真の人間のイデアは、多くの哲学者の考える人間のイデアに共通する性質をもつものでなければならない。

そのような個々の哲学者が想定する人間のイデアから、多くの人々が受け入れることのできる人間のイデア像を導き出すために、プラトンは弁証法を適用する。弁証法は、ある人が考えたイデア像に対して、別の人が考えるイデア像を対比させ、この 2 つのイデア像の差異を議論することで、より理想的な新たなイデア像を作り出す方法である。この方法も、議論の方法として現代に生きる我々に大きな影響を与えている。

プラトンもソクラテスと同じように、倫理的な行動の実践は教育と訓練なしには達成で

きるものではないと考えていた。つまり善をなす人間のイデアは、自然に獲得できるものではなく、教育と訓練によって身につくものであるとした。

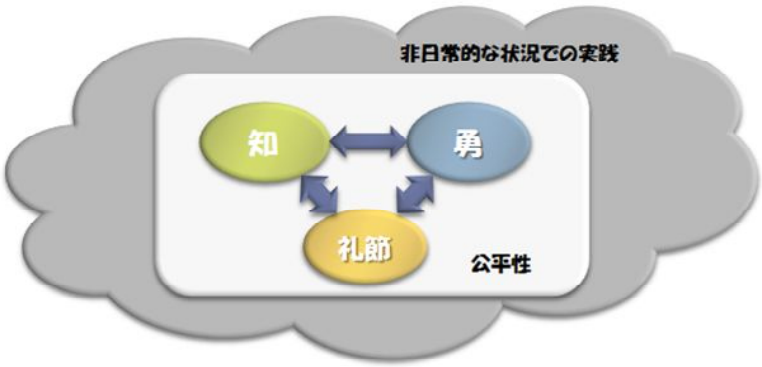

図 4.2 プラトンの倫理観

特に、非日常的な状況においても、善をなす人間のイデアと同じように行動できる人間になるためには、実践的な訓練と経験が必要であるとされた。その意味で、ソクラテスとプラトンの倫理観には、著しい違いはなかった。

ギリシャの哲学者アリストテレスは、ソクラテスが説き、プラトンが理論化した倫理を、より普遍性を持たせた一般的な人間の倫理観として議論した[9]。ここで、ソクラテスが実践において重視したのが「中庸」であった。

中庸は、極端を避けると言う思想である。極端に自己犠牲的な行動は、決して「倫理的とは言えない」とするのがアリストテレスの立場であった。現代的な表現を用いれば、バランスのとれた考え方に基づいた行為の選択が、アリストテレスによれば倫理的な行動の重要な規範であった。

このことからアリストテレスは、法や社会の行動規範に準じた行動を選択することも、倫理的な行動に重要な要件であるとした。ソクラテスが、社会の常識がどうであっても、自分が正しいと信じることを、勇気をもって実践することが重要であるとしたのに対して、アリストテレスは、社会常識に照らして極端にはならない行動を選択するという実践的な考え方を重視している（図 4.3 参照)。

アリストテレスの哲学は、キリスト教を介して、特に現代西洋社会に大きな思想的な影響を与えている。その意味で、現代のアメリカ社会を見ても、アリストテレス的な倫理観を正統的な倫理観とする傾向があると言える。

中庸を重視する思想は、時としてベンサムの功利主義的な実践論と結びつき易い[xii][10]。このことは、多数が受け入れやすい意見を、正しい意見とする考え方になることが多い。

図 4.3　アリストテレスの倫理観

アリストテレスの倫理観とソクラテスの倫理観のもう一つの違いは、アリストテレスが善を為す姿勢は、人間の自然な発達によって備わるとしたのに対して、ソクラテスは、人間は教えられ、訓練を受けなければ善をなすことができないとした点である。非日常的な状況における行為の選択を問題にしても、この二人の哲学者は全く異なる立場に立っている。

これは、ソクラテスが絶対的な善の存在を前提として議論しているのに対して、アリストテレスが倫理的行為については絶対的な善は存在せず、状況によって何が正しいかは変化するものであることを前提に議論していることによる。ソクラテスよりもアリストテレスの方が、よりプラグマティック(実践論)的な立場に立っていたと言える[11]。

同じギリシャの時代に、医師ヒポクラテスとその弟子たちは、医者が従うべき倫理観について考え、次のような「ヒポクラテスの誓」として知られた行動規範をまとめた。それは、

- 医学を病める人々のためにつかい、人を病めさせるためにつかわない。
- 命を危うくする薬は、たとえ頼まれても他人に手渡さない。
- 私の生涯と医術の全てを、医の徳に捧げる。
- 患者の最善のために患者を診る。金や他の欲望のために患者を診ることはしない。
- 仕事で知りえたことは、他人には漏らさない。

である。このヒポクラテスの誓が、現代の専門家の倫理規定の原型になっている。

第 2 節　プロテスタンティズムの倫理と資本主義の精神[12]

20 世紀初頭、ドイツの社会学者マックス・ヴェーバーは、米国の急速な経済発展の原因として、**プロテスタンティズム**の倫理観と資本主義の本質に親和性があるとした。セントルイス博覧会での講演を基に書かれた名著「プロテスタンティズムの倫理と資本主義の精神」である[12]。

この書物の分析が科学的に正しいかどうかについては異論もある。とは言え、その主張には説得力がある。それは、オランダ、イギリス、米国、ドイツなど 18 世紀以降の世界で経済的に発展できた国々の全てが、新教国(プロテスタント教国)であったからである。特に米国は、イギリスを追われた清教徒たちが建国した国であり、米国の企業家たちの経済活動に対する徹底した合理主義的傾向は、イギリスやドイツにも見られなかった特徴があったからである。

ヴェーバーは、米国に根付いた清教徒的な生活姿勢が資本主義に適合したものであるとして、米国の経済発展が起こるべくして起きたものであると述べた。特に、米国の企業家たちが、寸暇を惜しんで働く姿は、ヨーロッパ諸国では見られないものであるとし、カルビニズムの影響の強さを指摘した[xiii]。

また、ヴェーバーは、ルターが指摘したドイツ語の「天職をまっとうすること」の重要性と、勤勉に働くことが神への帰依に矛盾しないことが、ドイツの経済的な発展に寄与していることも指摘している。イギリスの産業革命とその後の経済発展についても、労働の倫理観におけるカルビニズムの影響と、一部の宗派における勤労の奨励と私有財産の蓄積の容認が、重要な役割を担ったとしている。

私有財産保全の容認に関する社会的な認知は、イギリスの哲学者、ジョン・ロックによる基本的人権に関する議論が始まりとされる[10]。ロックは、市民と国家権力を代表する国王の間には、基本的人権の維持に関する共通の理解が必要であるとした。基本的人権とは、国民の生命の保全、国民の私有財産の保全、そして国民の信教の自由の権利である。これらについては、国家と言えどもそれを犯すことはできないとした。

この基本的人権の 1 つである「私有財産の保障」に関する権利は、資本主義の根幹を支える要素の 1 つである。この権利の保障がなければ、人々は富の蓄積を目指すことができない。

したがって、勤勉に働くことの目的は、自分自身のためではなく、何か別の理由が必要になる。これは、医者であれば、患者の命を救うと言う医者の使命のためとなるが、そのような論理は、一般の労働者には適用できない。

また、イギリスや米国で、資本主義経済が大きく発展した背景には、資本主義が市場における競争をその重要な特徴としていることがある(図 4.4 参照)。このとき、競争はいくつかの基本的ルールに則って実施されなければならない。

それらのルールの中には、独占の禁止のように明文化されるものも多いが、特に公正な

[12]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.24-30 参照

競争の維持のための規範の多くは、明文化されないルールも多い。そのような明文化されないルールの中に、資本主義の発展を支援するものがあったと言える。

18 世紀の産業革命の時代のイギリスで、「諸国民の富」(国富論)を著したグラスゴー大学の教授、アダム・スミスは「道徳感情論」の著者でもある[13]。彼が人間の倫理的な行為が円滑な社会の運営を支えているとし、倫理観の重要性を述べたことは、注目に値する[xiv]。

すなわち、何が公平な競争であるかは、社会の構成員である個人個人の倫理観に依存して決まるのである。その意味で、19 世紀のイギリスや米国の国民にとっての常識が、それらの国々の資本主義の発展にとって、適したものであったと言える。

図 4.4 アダム・スミスの市場と競争

特に、市場における競争が、国家や政府の影響からは独立したものでなければならないことは、重要な前提である。また、基本はより低価格で財やサービスを売る供給者が市場の勝者になることが前提だが、その価格は供給者が原価を無視して、意図的に低く設定したものであってはならないとする規範も重要である。ここでは、供給者の「意図」と言う、行為者である人間の倫理的な側面が重視されている。

19 世紀の世界で同じヨーロッパでも、カトリック教国において経済発展が遅れた理由として、中世以来、ローマ教会が私有財産の蓄積を悪として、経済的な余剰価値の全てを教会へ寄進することを教えとしていたことが、庶民の勤労に対する態度を中世的なものにしていたことを指摘している。私有財産の蓄積を認めなければ、特に労働者階層における勤労・勤勉の意欲は生まれないからである。

19 世紀のドイツの哲学者カントは、ベンサムが提唱した功利論を原則とした判断は、倫理的には誤りだとした[10]。カントは、何が倫理的に正しいかは、理性的な思考の結果で、条件に依存せずに決まるとした。カントは、倫理の実践的な側面は認めながらも、それは

理性的な思考の実践であるとした[14]。

そして、倫理的であるかどうかは、「善をなそうとする」意志があり、その目的を達成するための方法を吟味する純粋に理性的な思考によって決まるとした。ただし、その思考の結果を実践する意志と勇気がなければ、善をなすことはできない。カントの倫理観は、絶対的な倫理の存在を仮定している点で、ソクラテスやプラトンの倫理観に近い。

カントは労働の倫理について直接言及しなかったのに対して、同じドイツの哲学者であるヘーゲルは、労働そのものと人間との関係について言及している[15]。このことは、ヘーゲルが提示した主人と召使の比喩に明確に述べられている。

ヘーゲルは、召使に為すべき問題を提示するのは、主人の役割であることは認めている。しかし、その問題を解決するために、自分の知識を総合し、その時点で最善と思われる解決策を考え、それを実践することで、主人の要求に応えようとする召使には、問題を解決する仕事の喜びがあるとしたxv。

ヘーゲルのこの比喩は、産業革命以後の社会で、資本家と労働者との関係を見た時、資本家は自分の資産を増加させる目的で労働者を雇い、労働につかせる。労働者には、与えられた仕事を拒否する自由はない。

しかし、その自由を認められない労働者には、自分の能力を活かして、与えられた仕事を実施し、何かの成果を生み出すと言う、現代的な表現を使えば、自己実現の喜びがあるのである。ヘーゲルは、明らかに新しい時代の労働の意味を見出していた。このことは、ルターの天職をまっとうすると言う思想にも関係している。

19 世紀から 20 世紀前半の世界では、個人における私有財産の蓄積は、一般の労働者が勤労に励む姿勢を積極的に支援する動機として機能していたのである。人々は、より豊かになることを目的として、勤勉に働き、富を蓄積し、結果として豊かになったのである。

このことは、19 世紀の後半から資本主義体制を取り、経済発展の道を歩み始めた日本についても言えることである。ただし、日本は新教国ではない。ただ、個人の私的財産の蓄積に対して、宗教的な問題も持っていなかった。

多数の労働者が働く工場における財(製品)の大量生産においては、労働者の大半がより多くの収入を得ることを目標として勤勉に働くことが重要になる。その労働に対する姿勢と、その社会における一般的な倫理観に矛盾が存在すると、労働者の労働に対する姿勢は消極的なものにならざるを得ない。つまり、生きるために必要なものを買うために必要な収入を得る分だけ働き、残りの時間は自分たちの生活のために有効に使った方が良いことになる。

社会全体が豊かになること、すなわち経済成長を長期間にわたって維持し続けることは、その社会の構成員である全ての労働者への分配も増加し、豊かになることを意味する。1800 年から 1950 年まで、米国の経済は、毎年ほぼ 2 パーセントの成長を続けたのである[16]。

これによって、国民の生活水準は、大きく改善され、豊かになったのである。それを支えたのも、国民の労働に対するピューリタン的勤勉さであった。

この時代、工場での労働はほとんど全てが単純な作業であった。従って、労働者の技能が問題になることはなかった。つまり、誰でも働く意欲さえあれば、働くことができたのである。

また、工場が機械化されていなかった時代であり、単純労働で製品を作るためには、個々の作業を単純な作業に分解し、分業によって効率的な生産を実現せざるを得なかったのである。このことが、極端な不況期を除いて、多くの人々が労働に従事できる機会を提供できる背景となっていた。

米国では、この間、新しい学問分野として経営工学(Industrial Engineering)が生まれた。テイラー(F. Tylor)による科学的管理法の提案に始まり、分業作業の標準化と時間研究が進んで、工程のボトルネックを簡単に見出し、それを平準化して取り除く方法が確立された[17]。工場には、生産管理と言う部門が作られ、専門的な教育を受けた技術者達が作業の時間研究に従事した。

さらに、1930 年代に入って、米国内では、統計的品質管理の基本的な理論の構築が行われた。特に、現在、管理図として知られている方法がシューハート(W. Shewhart)によって提案され、生産工程の状態を統計的に管理する方法論が確立された。

この統計的品質管理法の導入によって、工場での大量生産によって生産される製品の品質が、管理されるようになった。さらに、生産工程における不良品の混入を、低く抑えることが可能となった。

特に、汎用的な部品の生産においては、仕様書に記述された仕様を満足する部品を生産することは、その部品を使って実現される最終製品を生産する工場において、不良品の生産を可能な限り少なくするために重要なこととなる。このように、製品の構造が複雑になるにつれて、個々の部品の品質をしっかりと管理することが重要な条件となる。このことは、工場内に品質管理のための統計理論の基礎を学んだ品質管理技術者が雇用される状況を生み出した。

このような生産技術者や品質管理技術者が工場内の職場に進出することにより、工場内には単純作業を担う工場労働者と高度な知識を学んでその知識を応用して仕事をする技術者が共存するようになった。この両者が勤勉に働くことによって、工場の生産が適切に実施され、品質の良い製品が生産されることになる。この場合も、従業員の勤労意欲は品質の良い製品を大量に生産するための原動力になっていた。

19 世紀末から 20 世紀初頭の社会、特に米国の社会においては、勤勉に働き富を蓄積することが善であり、倫理的な生き方であった。ただし、蓄積した富を浪費することを悪徳とする考え方も一般的であった。

これは、宗教改革で新教徒達が信じていた神の教えに適合したものである。特に、カルバン派の思想に近い。カルバン派の思想の強い影響を受けたイギリスの清教徒が中心となって建国したアメリカ合衆国においては、この思想が色濃く残ったと言える[12]。

解放された市場における競争を前提とした資本主義経済においては、公正な競争環境が

維持されなければ、資本主義経済は崩壊する。その市場における公正な競争環境の維持を可能としていたのが、個人及び企業組織の倫理であったことも、重大な事実である。第 2 次世界大戦が始まるまでの米国社会においては、労働市場においても、製品市場においても、株式市場においても、暗黙の規律が厳格に守られていた。

特に商品を売り買いする市場における独占の禁止に関する様々な規律については、法的な禁止事項を含めて、様々な事柄が合意事項として了解されており、それを逸脱することは社会的な制裁を受けることを意味していた。

たとえば、企業間の関係で、特定の企業がある企業に何かを発注する場合、直接的な資本関係がなければ、相手先企業の総売り上げの 3 分の 1 以上を占める発注をしないことが暗黙の了解になっていた。これは、発注量によって実質的に相手先企業の経営を支配する可能性があるからである。

このような暗黙の了解事項は、米国以外の国で生まれ育った人々の常識からは想像できない規律である可能性もある。しかし、米国の市場が国内に閉じていた時代、このような暗黙の規律は、円滑な経済の運営に大きく寄与していたと言える。

このように、資本主義は、それを支える社会の倫理観を前提としなければ、成立しないシステムである。極端な言い方をすれば、市場における「公正な競争」の意味も、その市場の背景にある社会の倫理観によって、変化する可能性がある。

第 3 節　産業化社会と中産階級の仕事と倫理[13]

20 世紀中ごろ、世界は第二次世界大戦の時代に突入する。そこでは、兵器の大量生産を支援する技術が大きく発展した。経営工学や品質管理を応用した工場の設計と管理が進んだ。

また、工場で生産される兵器もそれまでの兵器のように簡単な構造のものから、高度な科学技術が応用された複雑な構造のものに変化していた。連射を可能にする大砲の製造、標的を自動追尾する照準装置、レーダや無線機などである。

米国では、大型の B29 のような爆撃機も、工場で大量生産できるようになっていた。第二次世界大戦中の米国の国民(労働者)の倫理観は、生産する兵器の構造は変わっていたが、「勤勉に働くことが良いことである」とする考え方には、変わりはなかった。ただし、多くの男性の若者が兵士として戦場に出て行ったため、工場内の労働力に占める女性労働者の割合は著しく増加した。

このことは、アメリカ合衆国内の社会に 2 つの変化を起こした。その第一は、工場における労働の質的変化である。それまでの工場の労働は、筋力の必要な肉体的労働が主であった。それに対して、第二次大戦中から、工場の労働においては、筋力よりも人間の判断や動作の柔軟性を活用するものが増加した。筋力を必要としていた部分には、機械が導入され、少ない力で物資を動かすようになっていた。

このような労働の変化に伴って、工場内では工場における筋肉労働から、主たる労働を人間の判断力を利用する仕事に転換するための専門的で知的な仕事が徐々に増加していた。20 世紀初頭では、経営工学を応用できる人々だけであったのが、品質管理の知識を応用できる人々、物資の運搬方法に関する問題を考える人々、資材の作業台への固定方法を考える人々、資材や半完成品に新しい部品を取り付けるための工具を設計・製作する人々、部品の注文や納品管理を専門にする人々などである。

これらの人々は、一般の工場労働者とは違って、高等教育を受けていることを前提としていた。これらの人々は一定期間の専門的な教育訓練を受け、専門家として業務に従事した。

さらに、これらの人々の活動を支援するため、比較的単純な計算業務や情報の収集・整理を実施する事務系の職員が必要となる。コンピュータが存在しなかった時代、これらの計算業務は全て人手で実施されていたのである。

このような背景から、工場における労働は、肉体労働、事務処理作業、専門的な仕事の 3 種類に分化してきた。数の視点では、肉体労働と事務処理作業に従事する人々が圧倒的多数であった。そして、これらの人々においては、それまでの時代と同じように「勤勉に働くこと」が、倫理的であった。働くことができるにも関わらず、働かないことは倫理的でなかった。また、一生懸命働かないことも、倫理的ではなかった。

工場内で専門的な仕事に従事していた人々の中にも、主として高等教育で学んだ知識を

[13]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.30-35 参照

応用して仕事をする人々と、仕事の実践から学んだ技能を活かして仕事をする人々がいた。どちらの人々も、自分の能力を活かした労働に従事しており、知的な仕事と言える。

これらの人々にとっては、「勤勉に働く」ことが自分の仕事の成果を保証するわけではない。このことから、資本主義社会の中でも、これらの仕事に従事している人々の間では「勤勉に働く」という倫理観は、普遍的な倫理観ではなくなりつつあった。

似たようなことは、日本の社会においても成立していた。それは、日本企業のモデルが当時の米国企業であったことが強く影響していると言える。しかし、それだけではない。多くの日本人にとって、「勤勉に働く」ことは、徳をまっとうするための条件として重要なことであった。

当時の米国の国民と同じように、「勤勉に働かない」ことや、「浪費にうつつを抜かす」ことは、悪徳とされていたのである。第二次世界大戦後の日本経済の発展に、そのような日本人の倫理観が大きく寄与していたことは、否定できない。

特に、米国企業で働く、高等教育で獲得した知識を応用して仕事をしている専門家にとっては、学んだ知識を適切に活用して問題を適確に解くことが重要になる[xvi]。このとき重要になるのは、何が問題なのかを探り当てる力であり、問題が明らかになった時にそれを解決する方法を考える力であり、考えた解決方法を実験し、場合によってはその方法を改善して問題解決に結びつける力である。

これらの能力は、主として高等教育で獲得するものであるが、その教育の質に大きく影響される。つまり、良い高等教育を受けた人々にとっては、獲得が容易なものであり、そうでない人々にとっては、獲得の困難なものである。

当時の米国の高等教育が、既に細かな専門分野に細分化されていた理由がここにある。ただし、重要なことはそれだけではない。むしろもっと大切なことは、「自分が専門家として問題を解決しなければならない」と自覚することである。これは、高等教育以前の姿勢の問題である。

すなわち、専門家としての社会的責任を明確に認識しているかどうかの問題である。そのような認識をもたなければ、どんなに良い教育を受けたとしても、「この問題を解決しなければならない」と言う認識にはならないからである。知的な労働に従事している人々においては、そのような認識をしっかりともつことの方が、単に勤勉に働くよりも重要になるのである。

このことは、高等教育で獲得した知識を使って仕事をする人々だけでなく、自分が仕事の経験から得た体験的な知識を活用して、自分の手で他の人々が必要とするものを作り出す技能的な仕事に従事している人々に対しても言えることである。他の人々が、何を必要としているかを察知し、どのようなものを作り出せばよいかを考え、それを試験的に実現し、利用者の声を反映して改良できなければ、仕事は完結しないからである。

それを実践するための技能は、訓練や実践経験から得られるのであるが、それだけでは仕事はできない。ここでも、やはり自分の社会的な役割や、組織における役割を認識し、

問題を解決しようとする強い動機がなければ、良い仕事はできない。良い仕事をしなければ、良い成果は生まれてこない。その意味でも、このような仕事に従事する人々には、自分に与えられた社会的責任や使命の理解と認識が重要となる。

20 世紀中頃から 20 世紀末に至る産業化社会が進展した時代、社会的には 2 つの階層の分化が起きた[xvii]。つまり、広義の知的労働に従事する階層と、従来と同じような主として肉体的な労働(単に筋肉を使った労働と言う意味ではない)に従事する階層である。

その社会においては、それぞれの階層に適合した 2 つの倫理観が存在した。すなわち、知的労働の階層における「社会的責任の自覚に基づいた仕事の仕方を追求する」生き方と、肉体的な労働に従事する階層における「勤勉に働く」生き方である。

20 世紀末の産業化社会においては、勤勉に働く倫理観を持つ社会階層の人々の中から、労働によって得た富を蓄積し、それを投資に使うことによってさらなる富を得ることに生き甲斐を見出す人々が世界的に増加した。これがその時代の倫理観であったと言えるが、そのような生き方が真に倫理的であったどうかは疑問である。そのような社会的な傾向が貪欲な資本主義を生み出す基盤となり、資本主義の健全な発展を阻害したことも事実である。

同じように、知的労働に従事していた、「自らの社会的責任を全うすべき」とする倫理観をもつ階層の人々の間でも、自分が所属しているより大きな社会に対する責任と、より限定された自分が所属する組織に対する責任のどちらを優先すべきかと言う問題がしばしば提起された。特に、社会の利益と組織の利益が対立する状況においては、労働者個人はその両者の板挟みになる。

このような問題が発生した場合、専門家が採るべき行動には、「社会に対して警鐘を鳴らす」意味での内部告発と、「組織の安定的な存続を重視する」意味での隠ぺいの 2 つがある。どちらの場合にも、当事者としての専門家には、大きな経済的負担のリスクや、社会的負担のリスクが問題になる。

内部告発に踏み切った場合、組織からは裏切り者として除外され、仕事を失うと言う犠牲が発生する例が多い。逆に、問題を隠ぺいした場合、内部告発のような問題は発生しないものの、問題が社会的に明白になれば、組織の存続が困難になるだけでなく、自分自身の専門家としての社会的信用も失墜する。

先進諸国において産業化社会の時代が終わろうとしている 20 世紀末、世界は個人的なレベルでは技術者倫理や職業倫理を問題にし、組織のレベルでは企業倫理や企業統治などを問題にするようになった。技術者倫理の教育が、高等教育における必須の項目として要求されるようになったのは、米国社会においても 1990 年以降であり、世界的には 2000 年以降と言える。さらに、企業倫理や企業統治が議論されるようになったのは、米国社会においても 2000 年以降であり、世界的には 2005 年以降である。

経済のグローバル化が急速に進展する背景の中で、専門家の職業倫理も企業組織の倫理や統治の問題も、個々人や個々の企業の自助努力に任せることはできなくなっている。そ

れでは、世界的な社会の秩序の維持ができなくなっているのである。すなわち、倫理は人が「善をなすため」の問題ではなく、社会の秩序を維持するための問題になりつつある。

第 4 節　ポスト資本主義社会の仕事と職業倫理[14]

フランスの現代哲学者コントスポンビル(A. Cont-Sponville)は、人間の思索には 4 つの基本軸(コントスポンビルは秩序と呼んでいる)があるとする[18]。第一の軸は理性と合理性の(科学)軸である。第二の軸は社会的秩序と法の(法律)軸である。第三の軸は善と人間性の(倫理)軸である。第四の軸は奉仕と自己犠牲の(宗教)軸である。

これらの基本軸は、互いに関係がなく、独立である。第一の軸は、科学や合理性(論理性)を追求する思考に関係する。第二の軸は、円滑な社会生活のために法律や規律で個人の自由を規制する思想に関係する。第三の軸は、善を追求する倫理的な生き方の実践に関係する。第四の軸は、宗教的信条や信念と生き方に関係する。

このような基本的な思考から、コントスポンビルは第一の軸に関係した人間の経済的な行動と、第三の軸に関係した人間の倫理的な行動の間に、調和的な関係を見出そうとすることは誤りであると主張する。どちらを優先するかは、個人の信念の問題であり、自由であるべきであるとする。

同様にして、ある人にとって倫理的な生き方が、その人が属している社会にとっては違法な生き方である例もありえる。だからと言って、その人に法に従うことを強制することはできないとしている。

このコントスポンビルの思想は、アリストテレスの普遍的な倫理性はないとする思想に対立する。しかし、18 世紀末のドイツの哲学者、カントは、倫理的であることは無条件に倫理的であり、周囲の状況によって変わることはないとした[14]。

これは絶対的な倫理性を前提とした議論であり、ソクラテスの議論を踏襲している。現代の多くの哲学者は、大多数の人間が共感できる絶対的な「善」の存在を前提にしている。その意味で、コントスポンビルもそのような思想の持ち主である。

20 世紀の産業化社会が終わり、新しい社会の時代が始まろうとしている。それがどのような時代であるのかについて、多くの思想家が共感する理論的枠組はまだ存在しない。ただ、新しい社会のいくつかの特徴は、わかっている。

その重要なひとつが、労働の知識集約化である。20 世紀の主たる労働が、人間の肉体を使った労働であったのに対して、21 世紀のそれは、人間の頭脳活動を利用した認識や推論、連想を使った創造的活動を主とした労働であると考えられている。

それは、時としてサービスをキーワードとしたり、知識集約をキーワードとしたり、問題解決をキーワードとして説明される。いずれにしろ、依頼者の問題を明確に認識して、その問題を解決するような解決策を模索し、最終的に依頼者の望んでいる状態を実現する。

それは、人間の思考活動を主体とした労働である。そのような労働においては、労働に投入した労働時間が問題になるのではなく、労働の成果が依頼者の要求を満足するかどうかが問題になる。

そのようなことから、問題解決型知識集約労働に従事し、成功するためには、問題の分

[14]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.35-39 参照

析を適確に実施する能力、解決策を考えその適切さを適確に評価する能力、さらに選択した解決策を適切に適用し依頼者が満足する状態を作り出す能力を要求される。人をしてそれらを適確に実践できる専門家として育成するためには、理論を理解させるだけでなく、理論の適用に関する実践的手順や方法を理解させ、実践できるような訓練をさせる必要がある。そのため、専門家育成に必要な期間は長くなる。

単に長期間の教育を受けたからと言って、その分野の専門家として生きてゆけるわけではない。実践を通して、専門家として社会から認知されて初めて、専門家として生きることができる。そのためには、仕事をする機会に恵まれなければならない。専門家は、仕事の経験から多くを学んでゆく。従って、仕事の機会に恵まれれば恵まれるほど、有能な専門家として社会に認知される可能性が高くなる。

そのような特性をもつ仕事に従事し、専門家として社会に認知されるためには、20 世紀を代表する肉体労働型労働者の倫理観であった「勤勉にまじめに働く」だけでは不十分であろう。何よりも「常に最新の知識を獲得し続けること」、「失敗を恐れずに正しい知識を適切に適用すること」、「依頼者に対して、依頼者が納得できる仕事にするため、しっかりと説明し納得してもらうこと」、「常に仕事の経験から学ぶ姿勢を維持すること」などが求められる。

「創造的労働者階層の台頭」を提起したフロリダ(R. Florida)によれば、米国社会においてはそのような創造的な労働に従事している人々は、全労働人口の 30 パーセントを超えている[19]。それらの人々の全てが、自分の職業からの収入で、自分の生活を維持しているわけではない。

かなりの割合の人々が、本来の仕事の他に、自分の生活を維持するための肉体的な労働にも従事している。すなわち、専門家としての経験が十分ではないため、自分の本来の仕事からは十分な収入が得られていないのである。

しかし、これらの人々の多くが、機会さえ与えられれば、その仕事に従事し、その経験から学び、将来、自分の仕事からの収入で生活できるようになることを目指している。とは言え、そのような状態に到達できる人間の数は極めて限られている。競争はそれだけ激しいのである。その競争に勝ち残るため、次世代の専門家を目指す人々が日々の修練に耐え、苦しい生活を続けているのである。

そのようなことから考えると、これからの時代の専門家に必要な職業倫理観は、「自分の選んだ仕事を天職として信じること」、「自分の仕事の社会的意義を認識し、それに打ち込むこと」、「生活のためではなく、その仕事のために必要であれば自分の生活も犠牲にできること」、「どのような状況でも諦めず、努力を継続できること」であろう。

そのような新しい時代の専門家にとって、仕事は生活を維持する目的(収入を得るため)のものではなく、仕事に成功すること自体が生きる目的になるであろう。別の表現をすれば、仕事はその人にとって自己実現の手段であると言える[xviii]。

単に、組織における地位を獲得するためでもなく、いわんや富を蓄積する手段でもない。

組織に所属する理由は、所属している組織が自分に必要な仕事の機会を与えてくれるからである。

すなわち、これからの社会においては、仕事の目的と個人の富の蓄積が直接的に結びつかないという結論になる。このことは、仕事と富の蓄積が直接関係しないため、勤勉に働くと言う態度が仕事に対する態度として求められなくなることも意味している。仕事に対する取り組み方はどうあれ、周囲から期待されている成果を達成できれば良いのである。成果主義の考え方は、このような論理からの結論である。

経営学者であり、マネジメントと言う言葉の発案者でもあるドラッカー(P. Drucker)は、これからの時代を「ポスト資本主義」の時代と呼び、その社会では、合目的的に構成される組織が重要な役割を担うとした[20]。すなわち、ある目的を達成するために様々な分野の専門家が集まり、効果的に協調して仕事をするための枠組みとしての組織が重要になると説いた。

組織は、産業化社会の企業のように労働者の生活を安定させるコミュニティ的な側面を消し去り、特定の業務目的を達成するために作られ、その目的が達成されるまでの期間、存続するものになるとしたのである。そのような組織においては、個々の専門分野を仕事とする専門家と、専門家を組織してその力を活用し、組織の目的を達成することを目的として仕事をする専門分野をもたない専門家である管理者が必要になる。その意味で、マネジメント人材も広い意味では専門家人材であり、その育成には他の従来型の専門家よりも長期の教育と訓練が必要となる。

ポスト資本主義社会の時代に生きる専門家にとっては、高等教育で学んだ知識だけで一生働くことはできない。専門的な知識は時間とともに陳腐化する。従って、専門家は常に新しい理論で古い理論を置き換えてゆかなければならない。

そうしなければ、依頼者の期待に応えられる仕事をすることはできない。新しい理論は、革新的に新しい理論の場合もあるが、古い理論に新しい説明を付けたような理論もある。いずれにしろ、内容を理解できない依頼者に対して、適切な説明をすることで、正しい理解をしてもらうことが必要になる。

このようなことから、これからの専門家は、常に学ぶ態度が必要になる。学ぶことを止めた時、その人は専門家ではなくなる。新しく学ぶ知識には、作業の手順に関する知識もあるが、問題とその解決策に関する理論的知識もある。専門家の会議で議論されるそれらの新しい知識を学び、依頼者に対して最も適切と思われる解決策を提示し、実践しなければならない。

さらに専門家には、自分が見出した新しい方法や理論に関する知識を、同じ分野の専門家に対して公表し、説明し、議論する責任もある。それがなければ、専門的な分野の知識は進歩しないからである。論理的に議論を積み上げることによって、人間は一歩ずつ真理に近づいてゆくのである。

専門家は、人間社会全体として、真理に近づいてゆくように自分の行動を律しなければ

ならない。富や名声を得るために、自分が獲得した新しい知識を秘匿することは、専門家としての倫理観に反する行為である。

第５節　ソフトウェア技術者の社会的役割と育成の課題[15]

1960 年代の米国のソフトウェア技術者は、大学で何を専攻したかに関係なく、主として就職した企業によってその適性が審査され、プログラマとしての教育訓練を受けて仕事に従事していた。当時、大学にはハードウェア開発に必要な知識を教える電気電子系専門学部学科は存在していた。

しかし、ソフトウェア開発に関する専門的な教育プログラムはなかった。電気電子系専門学科の卒業生を中心として、数学科や物理学科のような理学系学科の卒業生、また機械工学や経営工学などの工学系学科の卒業生、さらに言語学科、心理学科、経済・経営学科などの文系学科の卒業生なども、プログラマの候補者として適性検査が実施されていた。

当時、米国 IBM で実施されていた適性検査は、どちらかと言えば論理的思考能力を評価することに重点を置いたものであった。米国内で実施されていた大学院入試のための共通試験である GRE に似た内容と、知能の高さを測定する知能テストの内容に似た問題から構成されていた。

採点方法は、GRE と同じように加点・減点法で、選択した問題に対して正解であれば加点、間違った答えを選択した場合は減点するものであった。ただし、正解を選択した場合の加点よりも、不正解を選択した場合の減点の方が大きく、選択に確信をもてなければその問題に解答しない方が良い評価が得られるものであった。

知能の高さを測る問題としては、あたかもランダムに並んでいるかのように見えるアルファベットの文字列から、そこに隠れた法則性を探し出して最後の文字の次に来る文字を答えの中から選ぶものが多かった。問題の中には、法則性が単純で直感的にも分かり易いものから、文字を自然数に置き換えて、数列として表現し直してから、隠れた法則性を見出さなければならない、暗号解読のような問題まで、様々な問題が混在していた。

また、論理的な思考能力を問う問題では、示された文章を読み、そこから推論できる内容を 5 つの解答群から選択する問題や、与えられた文章から線形な連立１次方程式を導き出して、それを解かなければ正しい答えを選択することができない問題など、高校までの基礎学力を応用して問題を解く力を問うような問題が出題されていた。

1970 年代の末まで、IBM ではこのような試験問題を世界共通試験問題として準備し、世界中の IBM でほぼ同一の試験を実施し、プログラマとして採用できるかどうかを判定していた。特に、このテストで問われる能力は、解答者が日常使っている言語における、言語能力に依存しない論理的な能力が測定できるため、世界のどの国においても同じ基準で対象となる人材のプログラマとしての適性を評価する方法として利用された。

しかしながら、日本の学生についてのみ考えると、日頃から加点・減点法の試験を受ける経験がないため、明確に答えが分からない場合でも、それらしいと判断できる答えを解答群の中から選択し、確信が無いにも拘らず解答する傾向が著しい。このため、日本人の学生の場合、日本人同士を比較する場合には問題は発生しないが、他国の学生と日本人の

[15]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.39-47 参照

学生を比較すると、一般に日本人の学生の得点が、外国人の学生の得点に比較して著しく低くなっていた。すなわち、必ずしも日本人学生の真の知的能力を評価できているとは言えないと言う意見もあった。

いずれにしても、1960 年代から 1970 年代末までの 20 年間は、大学を卒業したばかりの人材については、主として論理的な思考能力に基づいてソフトウェアの技術者を選別し、採用を決定していた。米国内においては、IBM が採用していた適性検査が、人間の持つ能力のごく一部だけを取り出して過大に評価する傾向があることが議論されていた。

それは、その試験が左脳優位な試験であり、言語を使った形式的思考や、論理的な思考のみを重視し、人間の思考を制御するもう一方の右脳の能力が十分に評価できていないと言う指摘であった。左脳には、言語野もあるため、その意味では言語を使った論理的思考の能力だけが評価されていたと言える。特に、左利きの人間が多い米国社会では、左利きの人間にとって不利な試験の内容であるとの指摘が多かった[xix]。

1980 年代に入ると、米国社会では大学にコンピュータ科学科と名付けられた、主としてソフトウェアの開発技術者を育成するカリキュラムを主体とした教育を実施する学科を工学部に設置する例が急増した。このことによって、ソフトウェア技術者の採用は、コンピュータ科学科の卒業生またはそれと同等な教育プログラムをもつ大学の、アクレディテーション認定を受けた学科の卒業生を採用すればよいこととなった。当然のことながら、コンピュータメーカでは、そのような学科に在学している学生を対象として、夏季休業期間中に就業研修のために研修生として受け入れるインターンシップ制が普及した。

1980 年代の末になって、OSS が徐々に普及し始めた頃、ソフトウェア技術者として自分が歩む道を選んだ学生たちが、その進路として最も注目していた企業は、マイクロソフトであった。特に、西海岸の有名大学のコンピュータ科学科では、マイクロソフトへの就職を希望する大学生が増え始めていた。

それは、マイクロソフトが最初の Windows である Windows3.1 を開発していた時期でもあった。多くの学生がマイクロソフトでのインターンシップに参加し、そこで Windows3.1 の開発に参画した。その時、インターンシップに参加した学生達の多くは、新しいオペレーティングシステムを開発する喜びを味わったのである。

ちょうど同じころ、米国社会では、高度化する製品とその開発技術に対して、一般市民からの不信感が出始めていた。それは、巨大なソフトウェアの場合、その全体を確実に理解できる開発責任者がいないことから、何か問題が発生したとしても開発企業の技術者達だけではしっかりとした対応ができなくなっているのではないかと言う懸念が背景になっていたと言える。

そのような社会的な不安を背景にして、大学教育の中に、技術者達が持つべき倫理観の教育を導入する必要性があるとの認識が一般化しつつあった。これを受けて、アクレディテーション制度を運営している組織である ABET と、米国の学界(IEEE)のコンピュータ科学教育カリキュラム検討委員会は、コンピュータ科学の標準カリキュラムに「倫理」の講

義を必修科目として導入することを提言した。

この社会における懸念は、第 1 章で議論したマイクロコンピュータの普及に伴い、多くの製品に組込みソフトウェアが利用される傾向が明確になってきたことが原因であった[21]。一般市民の間にも、組込みソフトウェアの利用の利点と、そのソフトウェアに残存する誤りによって生じる問題の脅威が身近に感じられるようになっていたのである。

つまり、社会は組込みソフトウェアを開発するソフトウェア技術者たちの良心と責任感、そして倫理観に全てを託すしかなかったのである。例えば、社会的基盤として利用される交通システムや通信システムの開発に携わるソフトウェア技術者が実施した仕事の質は、それらのシステムの完成後にシステム内に残存する誤り等の不具合の量に直接的な影響を与える。

もちろん、ソフトウェアに残存する誤り等の不具合を完全になくすことは、人間であるソフトウェア技術者が作業する限り不可能である。しかし、その残存量は、ソフトウェア技術者達の仕事の質に大きく依存して変化する。完成したシステムに未発見の問題が残存していれば、それが社会的に重大な事故を発生させる原因になるリスクとなる。

組込みソフトウェアの利用が進むことは、一方で社会的な利点も大きい。従来は、製品の実現や製造に多大なコストがかかっていたのに対して、実現が容易になったこと、製品の構造を簡素にできることなどから、開発コストや製造コストを著しく低減できる。機器の製造においては、組み立て工程でも熟練工を必要としないため、労働コストの低い開発途上国などで製品を製造しても、製品の品質に大きな影響を与えなくなったからである[22]。

この利点と、組込みソフトウェアの利用が増加することによって拡大する、ソフトウェア特有な問題である非決定性によって生じる社会的リスクによる脅威とのバランスをとることが、現実に社会的な課題となった[23]。その課題を解決する 1 つの解決策が、技術者の倫理観に対する認識の徹底であった。そのためにも、高等教育における将来の技術者に対する倫理教育が重視され始めた。

さらに、ソフトウェアを主体とした製品の開発では、ハードウェア技術に対する依存度が低下するため、類似製品の開発は従来のハードウェアを主体とした製品に比較して、はるかに容易になる。したがって、開発期間は短縮できる。

そのような背景から、アジャイル開発が注目される。アジャイル開発では、新製品の要素を独立な機能別に分割し、同時並行して開発を進める。これは、ハードウェアが主体となる製品では容易ではない。

また、そのようにして開発する類似製品への資本の投資は、その類似製品の元となるオリジナル製品の開発に必要な投資に比較して、著しく小さなものになる。それは、アジャイル開発で最も時間を必要とする、製品にとって最も重要な機能の試作と選別のための開発作業の繰り返しを必要としないからである。

このことは、オリジナル製品の開発者よりも、類似製品の開発者の方が相対的に有利であることを示す。それでは、オリジナル製品の開発者は、その製品への投資を回収するこ

とが困難になる。これは、資本主義の原則である私有財産(知的財産権)の保全が成立しないことを意味する。

そのような後発メーカの優位性から、先発メーカの投資を守り、その開発投資の回収を保証することが重要になる。すなわち、オリジナル製品を開発した先発メーカに対して、その権利の優先権を保護するための法的な制度の整備が必要となる。製品の実現方法に対する知的財産権を保護する制度としての特許権と、その最も重要な部分である組込みソフトウェアに対する知的財産権を保護する制度としての著作権は、企業の知的財産権を守る手段として、今日の世界では重要である。

これまでの市場における競争を基盤とした資本主義社会の枠組みの中で、これまでの工業製品に対しては、これらの法的枠組みは機能していた。それはこれまでの工業製品が、ソフトウェアを利用せず、物理的な機構として実現されていたため、大量生産のためには工場の生産設備をその製品設計に合わせて整備する必要があったためである。組込みソフトウェアを利用した製品の場合、物理的に実現される部品は標準化され、製品の組み立て工程も簡単に整備できる。

このことは、コピー製品の開発と生産を著しく容易にする。そのため、新製品の開発者の権利を保護する法的な枠組みを整えることが必要となる。ただし、このことで問題が解決するわけではない。重要なことは、知的な生産物については、他者が考案したものをコピーすると言う姿勢が、社会全体にとっても危険な姿勢であり、技術者としてやるべきではない行為であることを、技術者達がしっかりと認識することである。

OSS が一般化する新しい社会では、この知的財産権保護の枠組みの位置づけは大きく変化する。そこでは、新しい製品の開発にも公的な資産でもある OSS が利用される可能性は高い。その場合、OSS それ自体の変更、または OSS を利用するソフトウェアの開発によって、オリジナル製品は開発される。

この場合、一般的なライセンス規約では、OSS それ自体に変更を加えた場合には、変更部分を含めて改良した OSS 全体を新しい OSS として公開することが要求される例がある。また、OSS を利用するソフトウェアを新たに開発した場合には、開発したソフトウェアは公開が要求される対象とはならないが、その部分の開発は多くの場合、容易であるため知的財産権の保護は困難である。

ここで重要なことは、コピー製品の開発を行おうとするソフトウェア技術者にとってはオリジナル製品の機能を知ることで、その実現のかなりの情報を得ることになり、類似製品を開発することに、大きな困難はない。従って、類似製品を開発するかどうかは、その開発に携わるソフトウェア技術者の判断に依存する部分が大きいと言える。

すなわち、開発技術者がオリジナル製品に類似した製品を自らが開発することに対して技術者としての良心の呵責を覚えなければ、類似製品が開発される結果となり、オリジナル製品に投入された資本は回収できない。

また、オリジナル製品の開発のために OSS それ自体に変更を加えたとしても、その部分

の開発を担当したソフトウェア技術者がその事実に即して、自分が変更を加えたOSSのソースコードをOSSとして公開することを選択することも、それを秘匿することも可能である。

その変更した部分を新たなOSSとして公開せずに秘匿すれば、開発した企業の利益を守ることはできるが、そのOSSの健全な進歩を阻害する結果となる。そのOSSの健全な進歩の阻害によって、将来数多くのユーザが損害を被る結果を招く可能性もある。

このような倫理的な問題を避けるために、時として企業はOSSの利用を回避する例も生じる。OSSを利用せずに、OSSに類似したソフトウェアを独自のソフトウェアとして開発する方法を選択するのである。このこと自体は、企業経営の視点から見れば短期的には合理的な選択であると言える。

しかし、社会全体の発展、すなわちそのソフトウェア(もとになったOSS)の健全な発展(進化)の視点から見れば、その健全な発展を阻害し、社会全体の発展にとって阻害要因になるため、望ましい選択とは言えない。つまり、局所的には最良の選択であるものが、大局的には最悪の選択になる例と言える。このような状況において、ソフトウェア技術者としてどちらを選択すべきかの問題は、極めて高度に倫理的な問題と言える。

どのような専門分野（discipline）においても、ある専門家の実施した作業の成果を、他の専門家（ピア）が専門家としてレビューし、責任をもってその成果の妥当性を保証することが求められている。もちろん、それでも見逃される問題は残っているはずである。

しかし、これまで報告されている事件、事故の原因は、そのような人知の及ばない部分が原因で発生しているのではなく、十分に予防可能な問題を見逃したことが原因である。そのような意味で、企業組織も、そこで働く専門家にも、最善を尽くすことが求められている。

組込みソフトウェア開発において、良い設計を考えることは、設計者として当然のことである。そのことと同様に、他の設計者が「良い設計」と考えたものを、精査し、その設計に隠れた問題点を洗い出し、設計者と議論し、さらに最初の設計案より良い設計にするべく最善の努力をすることも、専門家としての重要な責務である。

現代社会の専門家達は、自分たちの失敗が、自分たちに権限を付託した一般の個人や消費者に、どのような悪影響を与える可能性があるかを十分に認識し、失敗を未然に防止できるよう、最善を尽くすことが必要である。ここに、職業人としての倫理観が問われる原因がある[xx]。

ピアレビューが事故の防止策として機能するためには、上述したような資本主義的倫理観が企業組織において、職業倫理観が専門家個人個人において確立していることが前提となっている。しかしながら、日本の企業を見ても、また日本の技術者を見ても、そのような状況にはない。

21世紀の社会においてソフトウェア開発に従事する技術者は、上述したような問題に直面せざるをえない。それは、20世紀の社会に生きたソフトウェア技術者が直面しなかった

問題である。そのような問題に対して、常に大局的な視点から合理的な選択をすることができる人材を育成することは、21 世紀の社会にとって重要な課題である。

また、企業の一員としてソフトウェアの開発業務に携わり、その職務を全うすべく、常に自らを律して働くことができるソフトウェア技術者人材を育成することも、社会の健全な発展のために重要な課題である。

これからのグローバルな社会の発展のためには、特に教育環境としてのコンピュータ利用が多くの人々に可能になることが重要である。そのことが、国際社会の持続的発展のために必須の条件となるためである。そのようなソフトウェアの開発や教育コンテンツの開発、そしてそれらの持続的な発展のためには、ライセンスに基づくソフトウェアやコンテンツのソースコードの公開と、その自由な改良を許す OSS の枠組みがますます重要になる。

したがって、それらの OSS の開発や改良に積極的に参加する人材の育成が世界的な課題になる。現在、Ruby の開発を除けば、日本の技術者が OSS の開発者グループにあまり積極的に参画していない現実がある[xxi]。このことは、日本の高等教育における技術者育成プログラムに根本的な問題が内在していることを意味している。

第5章　品質の概念と品質論の発展

第 1 節 ものの質に関する考察の起源[16]

歴史上、「品質」に相当する概念を表す言葉が最初に使われたのは、ギリシャの哲学者、プラトンが書いた書物においてであるとされる[xxii][35]。すなわち、プラトンは、「いかなる」と言う意味のギリシャ語の形容詞を名詞化し、**poiotes** という語を造語した。ローマ時代のキケロは、この言葉をラテン語に翻訳するため、「どのような」を意味するラテン語の形容詞 qualis から qualitas と言う名詞を作り出した。

キケロの qualitas は今日の英語の quality(質)に相当するラテン語の名詞である。洋の東西を問わず、哲学者たちは物質の根源を、いくつかの性質の異なる火、水、土などの組合せとして考えていた。水が温度によって、固体(氷)、液体(水)、気体(水蒸気)に変化することから、存在の状態を区別する必要があることを知っていた。このことは、「質」の概念自体がプラトンの発案と言うものではなく、もともと人類が共通に持っていた思考方法に共通したものであったと言える。

プラトンの後、アリストテレスは形而上学を提唱し、自然界を支配する法則について議論した形而学(physics, 現在の物理学)に対して、人間が世界をどう捉え、どう考えたらよいかの思考・分析方法を議論する理論(meta-physics, メタ物理学)を考えた。その中で、アリストテレスは、物質（実体、存在するもの（substance)）とは何かを考え、その量的側面と質的側面を分けて考えようとした。量的な側面を表現する言葉として現在の英語の quantity(現在、日本語では質量と訳されることが多い)に相当するギリシャ語を用い、その対立する概念である質的な側面を表現するためにプラトンの発案した poiotes を使ったのである[xxiii]。

アリストテレスの哲学における「質」は、今日の日本語で言えば「様態」に近い概念であったと思われる。それは「ものが存在している様子」を意味している。例えば、水と氷で言えば、「水が低温で固体化したものが氷」である。この物質の「液体状」と「固体状」の様態(状態)の変化を、明確に分類したいとの意図から「質」の概念と、それを意味する言葉を用いたと言える。今日的な言葉で表現すれば、物質の物理的な量的側面と化学的な質的側面を分離して考えようとしたのであろう[73]。

現在の「品質」の概念は、20 世紀に入った米国で、製品の「質の良さ」を意識して議論するために、プラトンやアリストテレスの「質」の概念を、少し定義を変えて転用したものであると言える。米国においてテイラーにより科学的管理法が誕生し、生産の量的管理が進み、次に質的側面を問題にせざるをえなくなっていた。そのための理論として 1931 年に品質の管理(control of quality)が、シューハート(W. A. Shewhart)によって、提唱されるに至り、プラトンやアリストテレスの質概念の延長線上に品質概念が定義づけられたのである。

---

[16]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.51-52 参照

その後、デミング(W. E. Deming)は、科学測定の誤差に関する研究においてシューハートの業績を参照したことがきっかけとなり、共同研究に着手した。デミングは、シューハートの生産に関する実践的理論を、品質管理(quality control)と呼んだ。シューハートは、プラグマティズムの哲学者、ルイス(C. I. Lewis)の思想的な影響を強く受けたと言われている。シューハートが考案したとされる管理限界の考え方には、プラグマティズム的要素が感じられる。

第 2 節　近代から現代への品質概念の発展[17]

(1) 経済学における効用理論

シューハートやデミングによって、品質管理が提唱されるはるか以前の 18 世紀から、製品の品質に近い概念は理論的に議論されてきた。経済学は、基本的に財の量的側面（需要や供給）と貨幣の量（価格）だけを議論する学問であるが、貨幣と財との交換を深く考察すれば、必ず「質の問題」に直面する。ある店で売られているバターと、別の店で売られているバターの価格が異なるとき、その 2 つを単に価格と量(重さ)だけで議論することはできない。

なぜ A 店で売られているバターが、他店との比較で、量の少なさに比較して価格が高いにも関わらず需要が多いかを説明しようとすれば、それは「消費者がバターの量と価格だけで、どの店でバターを買うべきかを決めているのではない」ことが明らかだからである。そのような基本的な問題は、経済学が誕生してすぐにアダム・スミスらによって認識された。そして、そのような問題を説明するため「効用」(utility)と言う概念が経済学に導入された。

この効用と言う一般性の高い概念は、スミスやベンサムに先立って、英国の思想家ロックによって提唱された概念であると言われる。産業革命後の思想家であるベンサムもスミスも、このロックによって提案された人間から見た物事の質や望ましさの状態を議論する概念を政治哲学や経済学に応用したのである。

ロックは、人々が労働によって生産するものの価値を効用として捉え、労働の価値をその生産物の効用で評価することも考えた。ベンサムは、このロックが提案した効用を、民主主義の原理を説明する方法として応用し、人々の幸福(効用・便益)と人々の苦痛(損失・費用)の総和の差で政治的な意思決定を導き出すことを提案した。

スミスと同時代に生きた数学者であるバッベージは、このベンサムの考察を発展させ、数学的に労働とそれによって生産される(生産物の)効用の総量を議論する労働価値説の枠組みを経済学において構築した。それは、労働の量を、その財の生産に投入した労働時間 T で測定し、生産された財(効用)の総量を Q とするとき、T/Q は生産方法が変わらなければ一定の値であり、それが生産された財の根源的な価値であるとする理論である。この労働価値説は、スミスによって、分業生産に関する経済学の重要な概念として取り入れられた。

効用概念と労働価値説は、マルクスの資本論においても重要な理論的根拠を形成する概念として論じられている。その効用とは、財を購入した消費者にとっての当該財の価値であり、「ありがたさ」である。消費者は財を購入するとき、つねに支払う貨幣の量に対する効用の総和を最大化するように行動するはずであると考える経済学者が多かった。

また、効用は所有する財が増加するのにつれて低減するという性質にも経済学者は気づいた。つまり、同じものでも、最初に得たときの方が、2 回目に得たときより、その財の購入者にとっての「ありがたみ」が大きいという（限界効用逓減）法則が成り立つ[xxiv]。

---

[17]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.52-57 参照

効用と労働価値(労働時間の価値)という 2 つの純粋に理論的な概念を使えば、財の生産と消費は、スミスが「神の手」によって合理性をもって動作するとした市場の概念と、その市場において財と交換される貨幣概念を使うことなしに説明できる。市場と貨幣が存在しなければ、財の価格を媒介として売買を考える必要もなくなる。

つまり、国家は、国民一人一人が生産に投入した労働価値を計算するとともに、特定の財の生産に必要な労働価値を計算し、国民が生産に供した労働価値に相当する財を、国民の要望に従って分配することが可能になる。また、国家として生産すべき特定の財の量は、その財の効用から算出することが理論の世界では可能である。

しかし、このような財の効用は、現実世界での計測が困難である。このことが原因で多くの経済学者を悩まし、さらには、「実証不可能な観念的な理論」であるとする経済学者が増加した。

確かに、効用は購買者の価値観に基づくものである。従って、そのような効用は個人個人の価値観の反映であり、量的な議論や分析に適合しない。このような分析の限界から、効用理論は、徐々に経済学の中心的なテーマではなくなっていった。

(2) 商品学と使用価値

19 世紀のドイツやオーストリアは、イギリスやフランスに比較して、近代国家としての発達が遅れたために、工業経済が思うように発展せず、全体として経済後進国の立場に陥っていた。このため、大量の工業生産品が、ヨーロッパ最大の市場であるドイツ・オーストリア経済圏へ流入する結果となった。この巨大市場に流入する外国製品の中には、粗悪品も多く、そのため経済効率が著しく悪化するという事態をまねく結果ともなっていた。

そのような背景から、ドイツやオーストリアでは、商品の質と価格の問題を論ずる「商品学」が誕生し、大きな進歩を遂げる。この商品学(Waren Kunde(独), study of ware)は、その後同様な立場にあった日本にも紹介され、当時の一橋大学を中心に研究が発展する。

商品学では、経済学とは異なり、財の量的側面ではなく、主として質的側面に焦点を当てた研究が行われた。すなわち、「この商品がこの値段で輸入され販売されることは、合理的かどうか」を分析し、議論することである。その方法は、当時、発展途上にあった化学分析である。

明治政府が鉄道を輸入したとき、機関車、車両はもちろんのこと、レール、枕木、ジャリ石まで高額な価格で輸入された。枕木やジャリ石は、国内でも調達でき、輸入するよりもはるかに安価である。さらに、国内で調達した枕木やジャリ石が、本質的に輸入品に劣るとは考えられない。

国内で入手困難な性質の石や木ならば別である。このような分析をするためには、当該商品(財)を形成する物質の性質を厳密に比較しなければならない。その方法を論じるのが商品学であった。

その基本的な方法は、商品を構成する部分(部品)の素材を分析し、その素材別の重量を割

り出すことである。個々の素材は、市場における相場価格を決めることができるので、そのような細分化された材料の素材価格を積算すれば、商品(財)のおおよその価格を計算できる。

このような、分析的な方法を組合せることで、もの(財)の価格の妥当性を分析できるとするのである。ただしこの場合、近年の経営学で議論される付加価値については、全く考慮されない[51]。

当然、商品を議論しても、「極めて似たような 2 つの商品を、消費者が著しく異なる価格で購入する」という問題に直面する。そこで、商品学者たちは、様々な価値を議論し始めた。

そして、「交換価値」、「使用価値」、「所有価値」という 3 つの独立した価値概念を定義した。交換価値は、需要量と供給量に基づいた「相場(市場)価格」である。この価格ならば、他人に売れるという価値(価格)を言う。使用価値とは、その商品を利用することによって所有者が得られる価値(すなわち効用)を言う。「切れる包丁と、切れない包丁では、使用価値が異なる」とする考え方である。所有価値とは、所有者が商品を所有することで得られる価値を言い、所有者の価値観に大きく依存する。

例えば土地は、農業、物流業、不動産開発業にとっては重要な生産財であり、使用価値があるが、消費者を含め多くの場合、直接的な使用価値はない。しかし、多くの消費者はできるだけ広い土地を獲得したいと願う例が多い。

これは、広大な土地を占有できることが、一種の社会的権威を暗示しているからである。このことによって、我が国において土地は、使用価値が低い場合でも、所有価値は高く、交換価値も高いとする価値観が根強く残っている。

交換価値は、従来の経済学上の財の価格概念と同じである。使用価値や所有価値は、経済学者が効用概念を導入したきっかけとなった、消費者の価値観に大きく依存する財の評価を言っている。使用価値は、今日、我々が品質として議論しているものに近い。上述した、枕木やジャリ石の例は、まさに使用価値が同じであるにも関わらず、不当に高価な輸入品を買わされた事例である。

(3) 品質管理と良品率

米国の品質管理における品質概念は、当初、大量生産される製品が良品と不良品に選別されることから、この良品率を議論する必要性から始まった。どんなに検査を徹底しても、製品が消費者の手に渡ったとき、全部の製品が良品であることは保証できない。

しかし、製品が消費者の手に渡ったとき、その製品が良品であるか不良品であるかの確率は、統計的にある範囲内に限定(管理)することができる。そのような消費者が入手した製品が良品である確率をもって、製品の品質としたのが始まりである(シューハートの製品品質)。

この品質管理における良品とは、製品がその仕様を満足している状態を言い、そうでな

いとき不良と言う。生産した製品の数(量)を分母として、そのうちで仕様を満足する良品の数(量)を分子とした比率を、良品率とする。どのような生産ラインでも、良品率を 100 パーセントにすることは不可能である。

したがって、生産者が採用すべき合理的な対応は、消費者が必要としている良品の数(量)に適合した数(量)の製品を生産することとなる。シューハートは、そのような経済合理性の高い製品の生産を提唱した。

第 3 節　功利主義に基づく統計的品質管理論の確立[18]

第 2 次世界大戦が始まる前の米国では、テイラーによる科学的管理法の提案以来、大量生産のための技術として、工場における分業方式を科学的に検討する実践的な考え方が確立していた。テイラーの考え方の根源は、18 世紀の産業革命時代に英国の数学者バッベージによって創案され、スミスによって経済学の重要な理論として紹介された労働価値説である。

テイラーの科学的管理法では、バッベージが、「生産方法が変わらなければ、1 単位の財を生産するために必要な労働の総量は変わらない」としたことを基本としている。したがって、「1 単位の財を生産するために必要な労働の総量(労働時間)を減少させる生産方法は、それ以前の生産方法をよりも優れている」と考えた[17]。

そのため、テイラーは、作業者が実施する作業の標準化と、標準作業時間の計測に重点を置いた。このことが、米国の一般の労働者の目には、労働強化に結び付くと映った。

テイラーの科学的管理法とそれに続く、大量生産の方法に関する議論は、その意味でスミスの古典的経済学を企業経営に応用したものであった。特に、労働価値説、効用理論、労働の分業を基礎とした経営の合理化に関する理論は、スミスがその著書『諸国民の富』(wealth of nations)で詳しく議論した理論であり、米国における大量生産・大量消費の経済と、それを支える産業の合理化理論は、その古典的な理論を基礎としていた。この時代、経済学はケインズによる近代経済学の時代に入ろうとしていた。

科学的管理法によって、米国における大量生産技術は大きく進歩し、米国は世界の工場としての基盤を築くこととなった。しかし、財をできるだけ小さなコストで生産する方法は確立できても、生産される財の価値が低ければ、その財は自由競争市場では競争力をもつ財ではありえない。

フォードによって自動車の大量生産方式の確立がなされ、アメリカの中産階級が購入できる自動車が生産されるようになっても、購入した自動車がすぐに故障しては意味がない。やはり、高価なイギリス製やドイツ製の自動車が市場競争力をもつ現実は変わらないことになる。

米国の物理学者であったシューハートとデミングは、そのような状況下で、大量生産している財の「質の良さ」を計測し、その計測結果に基づいて生産ラインを管理する方法を考案し、産業界へ提案した。その基本は、財の質の良さ、「品質」を「製品の仕様に対する生産された個々の製品の適合の度合い」と定義したことによる。すなわち、製品の仕様を明確に定義し、生産過程において生産された製品がその仕様に適合したものであるかどうかを判定し、適合しない製品を不良品とし、適合する製品を良品とした。

シューハートとデミングは、そのようにして判定された良品と不良品について、個々の製品の判定に利用された計測値を統計的に処理することで、生産工程が正常に機能しているかどうかを管理図によって判定することが可能であることを示した。この判定には、通

[18]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.57-60 参照

常、ガウス分布が応用された。すなわち、長期間の観測から得られる、生産工程が正常に機能している場合の測定値の分布(平均値と標準偏差)を、生産ロット単位での測定値の分布と比較することによって行われる。

生産ロットの平均値と、生産ラインが正常に機能している場合の平均値の差が統計的に有意な差である（大きくずれている）場合、生産ラインに異常が発生しており、生産を継続すれば製品ロットの不良率が高くなることを予測できる。したがって、生産を一時中止させて、生産工程で何が起きているのかを分析し、その分析結果に基づいて、適切な対応を採ることが重要になる。

このような考え方は、イギリスの思想家ロックによって提唱された経験主義的な考え方に基づくものであった。経験主義においては、現実に生起している現象の観測が全ての基礎となる。

その現実こそが世界の真の姿(現実)であるとする。この考え方を発展させ、現実に生起している現象を、観察、測定して、そこから得られたデータを統計的に処理することで、現実の真の姿を数値化することが可能となる。

この正しい現実の姿に対して、今、観察している現象が、それと大きく外れているとすれば、それは何かが正しく機能していないことを示している。このことを客観的に判定するため、基礎的な統計理論の検定法を応用したのである[24]。

このシューハートとデミングによる基礎統計学を応用した品質管理論の誕生によって、古典的な経済学では精細な議論が不可能であった財の質に関する議論、効用理論の科学的・客観的議論が部分的にも可能となった。もちろん、この統計的品質管理論において、理論化された財の質に関する議論は、問題となっている財の仕様と生産された財(製品)において観測された特性の差が、許容可能な範囲を逸脱しているかどうかだけを言うものであり、財そのものの質全体を総合的に把握するものではなかった。

また、生産ラインの正常時における測定値の平均値と、観測している現在の生産ラインにおける製品の測定値の平均値との差が有意かどうかは、設定された信頼度によって決まる。すなわち、差が大きい場合には高い信頼度(例えば 99 パーセント)で有意となり、差が相対的に小さい場合には、低い信頼度(例えば 90 パーセント)で有意となる。

これは、「差がある」とする結論が誤りであるリスク(危険度)の違いを意味する。その差が小さければ、「工程に異常が発生している」とする判断の誤りリスクは高くなる。製品の品質を高めようとすれば、低い信頼限界を採用して不良かどうかの判定を厳しくすることになる。

そのことは、「良品であるかも知れない製品を不良品と判定する」リスクを高め、結果として生産ラインにおける歩留まりを悪くするため、生産コストを高める。生産コストが高くなれば、製品の販売価格を上げなければ、採算性が悪化する。

つまり、品質を高く維持しようとすれば、結果として、製品の販売価格を高めざるをえないことになる。そのことから、可能な限り多くの購買者が満足できる品質の製品を、多

くの購買者が許容できる範囲内の価格で生産するためには、良品と不良品を判定する合理的な基準を設定することが重要となる。すなわち、功利主義的な品質管理である。

このような古典的な経験主義思想に基づく、功利主義的な統計的品質管理法は、米国社会における大量生産技術の発展に大きく貢献し、第二次世界大戦における連合国の日独伊枢軸国に対する戦略的優位性を保つ要因となった。すなわち、兵器の稼働率の大きな差に表れてきたのである。

兵器そのものの量で言えば、連合国側と枢軸国側に極端な差が見られなかったものの、実際に稼働し、利用できる兵器の量には、統計的品質管理の効果によって、大きな差があったと言われている。特に、精密機械である軍艦や戦闘用の飛行機(戦闘機、爆撃機、偵察機など)、戦車や自動車、無線機やレーダ等、高度な工業製品でもある兵器の稼働率には、著しい差があったと言われている。

第４節　デミングのプラグマティズム的品質管理論と品質保証[19]

1970 年頃から米国企業に浸透し始めたプラグマティズム的品質論は、古典的な経験主義思想に基づく功利主義的統計的品質管理へのアンチテーゼ(反論)として提案されたものである。そこでの議論の焦点は、従来のような品質管理の対象としての製品（プロダクト）ではなく、それを生み出しているプロセスにある。「良いプロセスが実践されているからこそ、良い品質の製品が生み出される」という仮定が、このプラグマティズム的品質論の根幹をなす仮説であり、重要な主張である。

そのプラグマティズムの根底には、デューイ(J. Dewey)の思想に、根強くあるように「何が正しいか」「何が真実か」について、人類は確定的な答えを得ることはできないとする立場がある[11]。従って、良い品質が何であるかを知って、製品を開発し、生産することはできないとなる。

我々人間にできることは、市場において消費者や利用者が良い品質であると評価する製品を開発し、生産しているプロセスを記述し、そのプロセスを記述に基づいて再現することである。それによって結果的に、品質が高いと言われる製品の生産を実現し、供給することができるとする。

デミングのプラグマティズム的品質論の思想的萌芽は、経験主義的色彩の濃い功利主義的な統計的品質管理における工程管理のための管理限界の設定と、それに基づく生産工程の管理にあるxxv。管理限界を逸脱した特性をもつ製品が継続的に生産されている生産工程では、過去の経験に基づいて設定された統計的な計測値の範囲を超えている[24]。

したがって、「同じ生産工程から生産されている」とは言えない製品が継続的に生産されていることになる。そのような「悪い質」を示す特性をもつ製品生産現象を認識した場合には、直ちに生産ラインを止めて、その「悪い質」の製品が継続的に生産されている原因を究明し、その原因を取り除かなければならない。

このような生産されている財(製品)の質を計測して、そのデータを収集し、統計処理することで生産ラインの状態を判定し、その情報に基づいて生産ラインの問題を早期に発見し、対策をとるべきとする理論をデミングは提唱した。この考え方を拡張すると、同じ不良率(欠陥混入率またはMTTF)の製品を実現していても、市場における評価がそれまでとは全く異なって、低い評価を与えられる例は存在する。

そのような経験から、我々は単純な統計的品質管理の限界を知った。それは、生産システムの内部の問題ではなく、外部に生じた変化が原因である。古典的な統計的品質管理にはなかった、「開発・生産プロセス全体を見ることで、(実際のプロダクトを見ることなしに）品質の重要な部分を管理・保証できる」とする思想は、その意味で全く新しく、画期的なものであった。

これに先立って、経済学の分野では、ケインズのマクロ経済学から、高度な数学を応用してより詳細に経済を分析する、新しいミクロ経済学が発展しつつあった。特に、米国に

[19]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.60-65 参照

おいてはレオンチエフ(W. Leontief)によって、産業連関分析の方法が提案された。

この産業連関分析は、国家経済をいくつかの構成要素に分割し、各構成分野とそれ以外の構成分野との間に観測されるマネーフロー(資金の流れ)の関係を線形連立微分方程式の係数行列として見ることで、均衡状態における各構成分野の最終的なストックがどのようになるのかを、係数行列の線形連立方程式の解として解く方法である。

ほぼ同時期に、フォレスター(J. Forester)が「インダストリアルダイナミクス」(Industrial Dynamics)を発表した。それは、マネーフローだけでなく、資源、情報などの経営資源も視野に入れた、微分方程式による現実のモデル化と、コンピュータを利用したシミュレーションによる分析の提案であった。

レオンチエフの産業連関分析もフォレスターのインダストリアルダイナミクスも、現実の経済活動を静的な方程式ではなく、動的な微分方程式でモデル化し、その解を求めることでその本質を見極めようとしたものである。これらの新しい理論の根底には、微分方程式の理論を応用した制御工学の発展と普及があったと言える。

この新しい思想の影響を受けて、米国における品質管理は品質保証へと大きく、方向転換を成し遂げた。つまり、「工程の最終段階でテストによって欠陥を発見・除去し、その情報に基づいてプロセスの状態を知り、対策をとる」というコストがかかる、短期的な効果に着目したフィードバック方式から、「プロセスの現状を適確に把握し、その情報に基づいて開発・生産されるプロダクトの品質を適切に管理する」というコストが低減され、長期的な効果が期待できるフィードフォアワード方式の予測管理に転換したと言える。

経験主義的な色彩の濃い古典的な品質管理論から、新しいプラグマティズム的品質論への移行の過度期にあって、ソフトウェアの分野においては、品質保証のための技術として米国 IBM のフェーガン(M. Fagan)によって提案されたのが、ソフトウェアインスペクション法(software inspection)であった[25]。その理論には、古典的なテストに基づく品質管理に近い「欠陥除去」を優先する思想と、新しいプロセスの実態に基づいて品質を保証しようとする葛藤がにじみ出ている。このフェーガンの試みは、後にミルズのクリーンルームプロセスの提案に統合され、ソフトウェア開発プロセスの進歩に多大な貢献を残した。

この大転換によって、経営理論的には、多大なコストが必要不可欠なテスト工程を簡素化し、最終的なプロダクトの品質(統計的品質管理と同様に残存欠陥率を前提としている)を低下させることなく、コスト低減と品質維持・改善を実現できるはずであった。しかし、そのような品質保証を底辺で支える新技術の開発が遅れたため、このプラグマティズムに基づく方法は十分な成果を達成できなかった。

このような背景から、米国企業では、1980 年代の後半から、プロセスをよりマクロな視点で俯瞰し、総合的な対応を考えることが重視され始めた。それまでの局所プロセスに焦点を当てた品質管理や品質保証ではなく、企業の様々なプロセスを統合的に見ることで、局所的なプロセスの改善では不可能であった、マクロ的または大局的なプロセス改善が可能となり、経営上の成功をもたらした。これが、ビジネスプロセス再構築 (BPR)の理論で

ある。

1991年センジ(P. Senge)により提案された「組織学習の理論と第5の組織原理」(The fifth discipline)は、プラグマティズム的組織論の代表である。センジは、プロセスの改善という方法で、組織の学習プロセスを管理することで、組織の目的に適合した成果を、効果的に達成できるとした[26]。

これは、1980年代の日本企業の品質管理の実践を分析した結果に基づく主張であった。組織目標を品質改善として組織学習を推進すれば、「プロセス改善によって品質向上を成し遂げる」という品質論に帰結するのである。

企業における新製品の開発を例にとれば、要求プロセス、開発プロセス、検証プロセス、品質保証プロセス、生産プロセス、販売プロセス、保守プロセスなどをそれぞれ独立したプロセスとしてとらえ、それぞれを個別に改善することが従来の考え方であった。

これに対して、全体を、それら全てを統合したビジネスプロセスと見ることで、全体最適化(最も弱点となっているプロセスを分析し、改善すること)を図ることが重視され、その成果が注目されるようになった。

これらの思想の影響を受けたものの代表に、1988年にハンフリー(W. Humphrey)らによって発表されたソフトウェア開発プロセス成熟度評価モデルがある[27]。このモデルは、ビジネスプロセスまでを含んで見ることはしていないが、「ソフトウェア開発組織全体を見て、そのプロセスを最適化すべき」との思想に基づいている。ただし、1980年代の前半にIBM社内で開発されたCMMの原型となったモデルは、後述するようにプラグマティックな立場ではなく、観念論的な立場に立っていたと言える。

フェーガンのインスペクション法を新しいプロセスとして見た場合の強固な反論も、ソフトウェア開発プロセス成熟度評価モデルへの強力な反論も、基本的には同じ疑問から出発している。それは、「なぜ良いプロセスが実践されていることによって、悪い品質のソフトウェアが開発されるリスクを低減させることができるのか」と言う経験主義者から提示される疑問である。

また、「決して良いプロセスを実施しているとは言えない組織でも、品質が高いとされる製品が開発され、生産されている例がある。これをどう説明するのか」と言う疑問も同じである。

仮に、市場では「ユーザが良い品質であると評価する製品が、最も受け入れられる（売れる）製品である」とする前提が正しいとすれば、プラグマティズム的品質論に根差したプロセス評価で良い評価を受けた組織によって開発される製品は、広く利用され、売れる製品でなければならない(ベラディ(L. Belady)のハンフリーへの反論[xxvi])。しかし、我々が現実に目にしているのは、必ずしも良い品質とは言えない製品が、市場をほぼ独占する例である。

例えば、市場で評価され、世界的に広く利用されているパーソナルコンピュータ用OSであるマイクロソフト社の製品は、CMMのモデルで評価した場合、レベル5の組織で開発され

ているのであろうか。また、フェーガンプロセスと呼ばれるインスペクションを忠実に実施している組織で開発されているソフトウェアの品質は、ユーザにとっていつも満足のゆくものであろうか。どちらの問いに対しても、回答は肯定的ではない。

第 5 節　日本における観念論的品質論の発展[20]

米国で発展したプラグマティズム的品質論は、現実を反映させて、個々の理論の不備を修正しながら時間とともに発展し続けている。その意味で、プラグマティズム的品質論の枠組みは、強固であると言える。何よりも、現実的であることが強みである。これに対して、古くから日本の技術者達を魅了してきた理論が、観念論的品質論である。

古くは、1960 年代の ZD(Zero Defect)運動は、生産現場における不良撲滅運動であり、「不良率の低減が製品品質の向上、そして企業の発展に直結する」とする考え方に基づき、「不良を完全になくすことが、究極的な品質の実現である」とする考え方であった。これは、古典的な統計的品質管理を極端に抽象化し、その一面だけをとらえて議論している観念論的な品質論である。そこでは、「どんなにテストや検査を強化しても、不良がゼロになることはない。」と言う現実的な理論が無視されている。

ところで、上述したハンフリーらのプロセス成熟度評価モデルも、1980 年代の前半、IBM 社の社内におけるソフトウェア開発組織のベンチマーク手法として開発された時点では、極めて観念論的プロセス論の影響を受けた考え方であった。品質改善はもちろんのこと、生産性向上も含めて、理想的なソフトウェアは、「理想的なウォーターフォール型プロセスを実践した結果として達成される」とする先験的な認識がその基本となっていた[xxvii]。

これは、当時のソフトウェア開発現場の実情から考えられた、現実の弁証法的逆説である。つまりそれは、ソフトウェア開発において標準プロセスが遵守されていないことが原因で、ソフトウェア開発に失敗する例が多かったからであった。

1980 年代後半から普及したシックスシグマ運動は、古典的な統計的品質管理と ZD 運動の精神を強く引き継いでいるものの、プラグマティズム的解釈が根底にある。第一に、シックスシグマでは、「品質向上イコール不良撲滅」という単純化した論理は採用していない。

また、サービスも対象とするため、より一般化された品質論を展開する。品質が何かを議論するのではなく、ある品質概念が定義された時、その品質をどこまで追求すべきかの相対的な目標設定の重要性を問題にする。その意味で、プラグマティックな立場を重視している。

観念論的品質論では、まず普遍的で先験的な品質概念の存在を前提とする。その品質は、市場におけるユーザの評価と無関係ではないにしても、ユーザの評価そのものではない。

例えば、マッコールによるソフトウェア品質特性の議論の影響を受けた、かつての ISO/IEC 9126 のソフトウェア品質の議論では、ソフトウェア全体について一般的（普遍的）に適用可能な品質概念が、先験的(アプリオリ)に存在すると仮定している[28]。

ISO/IEC 9126 で言えば、

- 機能の実現に関すること(機能性)、
- 機能上の欠陥の存在がユーザに与える影響に関すること(信頼性)、
- ソフトウェアがユーザにとって使い易いかどうかに関すること(使用性)、

[20]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.65-70 参照

- 設計に無駄がなく実行効率が良いか悪いかに関すること(効率性)、
- 問題が明確になったとき修正や機能追加が容易かどうかに関すること(保守性)、
- 稼働環境を変更しようとするときに簡単に変更できるかどうかに関すること(移植性)

などが品質の各側面(品質特性)として定義されている。これらの品質特性が望ましい状態にあるソフトウェアが、その利用者にとって品質が良く、高く評価されるソフトウェアであるとする考え方は、観念論的であると言える。

このソフトウェア品質モデルの原案を提案したマッコールは、それまで研究者個々人がそれぞれの立場と思想に基づいた定義で議論していたソフトウェア品質が、より普遍的な11の基本概念に集約可能であり、それらを厳密に定義できるとした[29]。ISO/IEC 9126は、そのマッコールが提案した11の品質特性を、ユーザの視点から分かり易く整理し、6つの基本的な特性に再分類したものである。

日本的品質管理を支える代表的な概念の例として、世界的に認知され、広く応用されている概念に、「**当たり前の品質**」と「**魅力的な品質**」の2つの対立概念がある。これらの概念は、表面的には極めて経験主義的な品質論に根差しているように見える。しかし、その本質は経験から得られた観察や考察を、何段階も抽象することでのみ獲得できる。

それは、経験からは直接的に得ることが不可能な、高度に先験的で観念論的な概念である。製品の設計者にとって、信頼性は重要な品質特性である。それは、設計者にとっては一般的に、直接的な使い易さよりも重要な特性である。必要なときに製品が機能しなければ、使うことさえできないからである。

しかし、その製品の利用者であるユーザにとっては、使い易さはしばしば信頼性に優先する。この矛盾を説明する理論として、「当たり前の品質と魅力的な品質」の理論は、我々の直観に鋭く訴えかけてくる。

「当たり前の品質」と「魅力的な品質」がそれぞれ、具体的にどのようなものであるかは、当然のことながら時代の要請によって変化する。しかし、品質概念を構成する製品の品質(特性)の中には、明らかに、ユーザから見れば最低限の保証は必要としながら、一定以上の水準を達成していれば、全く意識されなくなる「当たり前の品質」に属するものがある。

逆に、ユーザは一定水準に達するまでは全く意識しないが、その一線を超えると製品の競争力を大きく左右する要因となる「魅力的な品質」に属するものもある。今日のソフトウェアにおいては、当たり前の品質の代表例が、信頼性であり、魅力的な品質の代表例が、使い易さである。

1990年前後における米国IBMの社内調査部門によるIBM社製オペレーティングシステムに対するユーザ評価の調査結果によれば、信頼性は、「当該製品の残存欠陥密度が一定水準を超えて高くなった時、ユーザによる負の評価との相関が高く」なる(当たり前の品質)。また、使い易さや導入のし易さは、「従来製品に比較して、ユーザが明確に認知できるほど

改善されると、ユーザによる正の評価との相関が高く」なる(魅力的な品質)。当たり前の品質と魅力的な品質の概念と理論は、こうした我々の実践経験によく適合している。

観念論的品質論では、普遍的で純粋な品質概念の存在を仮定する。純粋品質概念としては、モノ(財)やサービスの効用が一般的なものであると言える。効用は、品質論の文脈では、製品を利用することで利用者が得る便益から感じられる幸福感の大きさ言う。

スミスの効用概念は、まさにロックの効用概念の延長線上にある。この効用概念は、経験論に根付いているものの、極めて観念論的な先験的概念と言わざるをえない。

1900 年頃、ドイツやオーストリアで著しく発展した観念論哲学の影響を強く受けた商品学における商品の使用価値も、このスミスの効用概念と類似の外延(デノーテーション)をもつ。さらには、1930 年代に登場した統計的品質管理における不良率に基づく品質概念も、成熟過程にある製品の効用(「当たり前の品質」に近い概念)に関連した概念である。

良品率に基づく品質概念は、スミスの効用概念の内包(コノーテーション)に関係している。この考察から言えることは、効用概念の外延は、極めて観念論的な意味を含んでいるものの、その内包は経験論的な意味をもつことである。このことは、哲学における「善」の議論に似ている[xxviii]。

いずれにしろ、普遍的な純粋品質概念の根幹にはスミスの効用概念がある。そして、その外延として、ソフトウェアに関して言えば、上述した ISO/IEC 9126 で定義される品質特性(機能性、信頼性、使用性、効率性、保守性、移植性)やマッコールのソフトウェア品質特性の概念などがある。

そのような品質特性が内包する概念には、多くの製品やサービスに共通する基本的なものがある。さらに、特定の製品(ソフトウェア)にのみ適用可能な個別的なものもある。

日本的品質管理に特徴的なもう一つの方法である品質機能展開は、特定の製品やサービスを対象として純粋品質概念を詳細な内包概念に分解する形式的な手法である。そこでは、ユーザ視点での効用を適確に説明する(財やサービスの質の良さを表現する)形容詞と、開発者・生産者視点での品質評価指標(測定項目)との関係を統合的に分析・整理する。

この形容詞を出現頻度順に並べ、それと関係の深い品質評価指標との関係をマトリクスで表現することにより、ユーザ視点での品質と設計者視点での品質との整合性を検討することができる。これは、純粋品質概念を個別の製品やサービスで、どのようにして明確に定義できるかを示した体系的な方法である。大きな意味で、日本的品質管理の原点は、観念論的品質論を基本的枠組みとして発展してきたと言える。

このことは、日本における品質論が、茶道や武道、そして陶芸などのような工芸などにおいて、その作用主体である人間に「私利私欲を捨て、その対象の完全性を追求する」ことを理想像としてきた文化的伝統が影響している。その主体としての人間が、「その存在のすべてをかけて打ち込む」ことで、完成度の高い結果が得られると信じられているからである。このことは、カントが「人が善を為す」とは、「心の奥底から、それが自分のすべきことである」と信じて実践した場合の結果であるとしたことに似ている。

この日本的で観念論的な品質の考え方と、1990 年頃の米国において一般的であったプラグマティズム的な品質の考え方は、相互に相容れない考え方である。しかし、観念論的な品質論の思想は、米国社会に対して大きな影響を与えたと言える。

それは、プラグマティズム的な立場と観念論的な立場の議論が、倫理の問題において常にあい対立する立場であり、どちらが正しいとは言えない議論が展開可能なためである。観念論的な品質論からの批判を受けて、プラグマティズム的な品質論は、経済のグローバル化と市場における自由競争を中心とした思想の時代的背景もあり、最終的にリバタリアニズム的な品質論に変質した。

第 6 節　市場における自由競争とリバタリアニズム的品質論の問題[21]

自由競争と政府による規制の緩和を前提とした**リバタリアニズム**の思想は、経済が激しい勢いでグローバル化する環境にあって、個々の企業がゴーイングコンサーンとして、その存続をかけて活動を継続するための重要な指針を与えている。経済学の分野において、フリードマン(M. Friedman)は、経済のグローバル化が進展する環境下で、国家は企業活動に対して可能な限りの自由を保証することが重要であるとした。

品質論に関して言えば、ユーザ視点なしに品質論は語れない。この時代にリバタリアニズム思想の影響を強く受けた品質論の代表例が、1980 年代末から普及した CS 運動である。

CS(Customer Satisfaction)運動では、顧客(ユーザ)の製品やサービスに対する満足度こそが、真の効用であり、品質(価値)の根源的な状態を示すものであるとする。その**顧客満足度**と呼ばれる顧客視点での効用を最大化するように、製品やサービスを企画・設計し、実現して、タイムリーに市場に投入することが、供給側の企業にとって最も重要だとする考え方である。顧客満足度に基づく効用概念と、規制の少ない開放された市場における自由競争の結果としての販売実績を統合することで、新しい品質論が構築された。

この新しい品質論は、グローバル化経済社会においては、究極の品質論であるかのように見える。そのような理由から、多くの企業が自社製品や提供サービスに関する CS 調査を実施するようになった。また、客観的な立場にある第三者機関(企業)が、複数企業が市場に供給している類似の製品やサービスを対象として調査を実施する例も増加している。

個々の企業にとっては究極の品質論のようにも見えるが、ソフトウェアのような知的な労働の成果を基本とする場合、後述するような新しいオープンソースパラダイムに関する強力な反論に対して、その矛盾を合理的に説明できる理論ではない。すなわち、重要な理論ではあると言えるが、まだまだ不完全な理論であると言わざるを得ない。

市場を完全に独占した企業は、市場にその企業が開発した製品だけを存続させる戦略を主体的に採用することができる。その条件下では、市場から財やサービスを購入するユーザには選択肢がなく、その市場で入手可能な財やサービスの品質の善し悪しに関係なく、全てのユーザが特定の財やサービスを購入し、利用することになる。

社会的正義の視点で、そのような事態(市場独占)を国家および社会が許容することはできない。それこそが、1970 年代においてほぼ市場の独占状態にあった米国 IBM を永年にわたって悩ませた、米国における独占禁止法の存在意義である。

以上の議論から、市場で最も成功している企業が開発し、生産している財やサービスと、その財やサービスの品質との間には、本質的に、論理的な因果関係が成立しているわけではないことがわかる。市場によっては、そのような意味での論理的な因果関係が成立する傾向が強くみられる地域の市場と、そうでない地域の市場が存在する。

それは、あくまで結果論であって、経験的には厳密に言って「良いプロセスが実践されている」ことが、財やサービスの良い製品品質を実現する直接的原因であるわけではない

[21]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.70-73 参照

のと同じである。また、競争のある、特定の市場で勝ち残った財やサービスの品質が、市場に供給されている他の製品に比較して優れているとは言えない。

ここに、新しい品質論の必要性がある。市場におけるユーザの財やサービスに対する評価は、財やサービスの効用に対する認識を検討する上で重要な情報を提供するものではあるが、そのような主観的な評価だけで財やサービスの品質を議論することはできない。

ここで重要になることは、ある程度多くの人々が納得でき、多くの観点から検討して矛盾をきたさない理論の構築である。そのような理論は、スミスの提示した客観的な質の問題も踏まえたものでなければならないはずである。しかし、スミスの効用理論には本質的な問題が潜んでいた。

すでに述べたように、財やサービスの効用は、供給量が増加するにつれて、その変化量(増分)に対する効用は減少する(限界効用逓減の法則)。これは、供給量の微小変化に対する効用の変化量(効用の消費量微分)が常に負であることを示している。

例えば、家族にとって、1 台目の自家用車は、2 台目の自家用車に比較すると、はるかに大きな効用をもたらすことを意味している。さらに 3 台目の自動車の効用は、2 台目の自動車に比較してもはるかに小さいものとなる。

このことからもわかるように、効用は市場や消費者の様々な状態の影響を受けて変化する。従って、製品やサービスを特定しても、普遍的な効用の定義と計測はできない。確かに、リバタリアンが言う「市場で選択された製品やサービスは、利用者のニーズに適合しているはずであり、選択したユーザにとって効用の高い製品やサービス」であるとする主張は、真理のある一面を述べている。

つまり、結果論として効用は分析可能であり、定義可能であるとしても、それは観察対象の製品やサービスの、観察期間内における効用を説明するにすぎない。特定家族の 1 台目の自動車に対する効用の評価データと、別の家族における 2 台目の自動車に対する効用の評価データを層別せずに比較することになりかねない。

効用理論を基礎とする限り、品質を、時間軸を捨象することで、普遍的な概念にまで昇華することは不可能である。または、「何が良い品質かは、市場とその時代の消費者に依存する」としか言えない。これが品質の本質であろうか。それでも、我々は「どのような製品やサービスが、良い品質であるか」を議論する。

それは、品質論はスミス的効用概念だけでは議論できないからである。品質論は製品やサービスの開発者が、「どのような製品やサービスが、良い製品やサービスであるとするか」を市場に提案する理論でもある。

ここにもう一つの興味深い理論があるマッキンタイア(A. MacIntyre)は、著書『美徳なき時代』で、「物語る存在」という概念を提起した[30]。それは、我々は家族や地域や国家の歴史(物語)の中で、それらの影響を受けながら生きている存在であり、これらのコミュニティを抜きにして自分自身のアイデンティティは説明できないとする立場である。

このことを製品やサービスに当てはめれば、「市場は、特定の企業がその社会と組織の歴

史の中で、どのように企業を発展させてきたかを理解した上で、その企業が市場に投入する製品やサービスを評価する」と解釈できる。

すなわち、似たような製品や商品であっても、それを市場に提案し、投入する企業が異なれば、市場の評価も変わる可能性が高い。優良企業が「そこそこの品質」の製品を市場に投入すれば、その期待値が高い可能性も高く、一定以上の効用が実現されていなければ、「品質の悪い製品」と評価される可能性がある。

逆に、無名のベンチャー企業が似たような製品を開発し、市場に投入すると、その期待値が低い可能性が高く、同じような効用の実現度であっても、「良い品質の製品」と評価される可能性は高い。その意味でも、純粋な効用だけでは、品質を語ることはできない。

それが、**ブランドイメージ**と呼ばれる概念の本質であろう。従って、市場の評価を、対象である製品やサービスの純粋な品質評価と見なすことはできない。

第 7 節　ソフトウェア品質論の基本的な問題[22]

プラグマティズム的品質論において既に議論したように、プロダクトとその生産過程であるプロセスとの関係は、経験的に明確なものではなく、実証された事実でもない。我々の直観にはよく適合しているようには思えても、あくまで仮説である。

確かに現実を見ていると、半分は当てはまるものの、半分は当てはまっていない。それを規律と言う説明で、教義的に押し付けることは、科学的ではない。「良いプロセスが良い品質の製品やサービスを生み出す」とする仮説は、実証的な研究を必要としている問題である。

ハンフリーに代表される観念論的プロセス論(品質論ではない)の立場からは、いかなる製品やサービスも、それを産出するプロセスに入力を与えることで生産される。従って、出力である製品やサービスの品質は、本質的にその産出過程であるプロセスの影響を受けざるを得ないとする先験的な知識が前提となっている。

これが観念論的プロセス論者の主張の根拠である。現実を抽象化した世界(現実ではない)では、この先験的な関係は論理的に成立するはずである。

上述した基本的な命題は、その観念論的な議論においても問題になる。つまり、観念論的プロセス論と観念論的品質論の統合である。このプロセスと品質の結びつきは、観念論的なアプローチを採用するとき、プラグマティズム的アプローチを採用する場合よりも、直接的かつ強固になる。

例えば、観念論的品質論の立場に立つミルズ(H. Mills)は「良いプロセス(例えば、クリーンルームプロセス)が実践されなければ、良い品質(欠陥の少ない)のソフトウェアが定常的に開発されることはない」とする信念をもっていた。これに対して、現在の日本の全社的品質管理論では、失敗の経験を詳細に検討し、その失敗事例の原因となったプロセスの不備を改善することで、品質を向上させられると言う立場をとる。

一見合理的な観念論的解釈ではあるが、現実を注意深く観察すると、適合しない例が存在する。それは、プロセス定義の粒度とプロセスへの入力が、現実の成果の品質に与える影響が大きいからであるとも考えられる。

仮に良いプロセスであっても、プロセスへの入力としての要求定義が不完全であったり、プロセスを実施する技術者・管理者の能力が十分でなければ、十分な品質のソフトウェアの開発は困難である。特に、完全な要求仕様をまとめることが容易でないソフトウェアでは、入力が不完全である場合が一般的である。

実際に良い実践を検証してみると、「良いプロセスが良い品質のソフトウェアを生み出す」例は多い。しかし、そうでない例も存在する。特にこの矛盾が、プラグマティズム的品質論者を悩ませ続けている。従って、プラグマティズム的に解釈すれば、「良いプロセスを実践することは、良い品質のソフトウェアを開発することに失敗するリスクを低減する」という解釈にならざるをえない。

---

[22]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.73-76 参照

ISO9000 の品質マネジメントシステムでも、「良いプロセスを実践することができる組織は、良い品質の製品やサービスを開発・提供することに失敗するリスクが低い」とする魅力的な仮説を前提としている[31]。その意味では、基本的にハンフリーらと同じ立場である。前述したように、この仮説は前述した、経験論的傾向の強い現場主義者による強力な反論に十分対抗できていない。

経済学者のフリードマンら、(規制のない)市場における自由競争を資本主義の最も重要な原理とみなすリバタリアニズムの信奉者によれば、「最も良い品質のソフトウェアを開発している組織は、市場で最も成功している企業」である。しかし、それらの企業が実践しているソフトウェア開発プロセスの多くは、プラグマティズム論者が言う意味での「良いプロセス」を実践しているわけではない。この矛盾を合理的に説明する理論はない。

このリバタリアンからの反論に対しては、全く反対の立場に立つリベラリズム論者から、強力な別の反論が提示されている。「市場で最も成功している企業が開発しているソフトウェアが、最も品質の良いソフトウェアだとすると、市場を経由しないで交換(提供)されているオープンソースソフトウェアがこれだけ多くのユーザに支持される現実は、どう説明されるのか。」

さらに、そのようなソフトウェアの開発に関わっている開発者のコミュニティが実践しているプロセスは、プラグマティズム論者から見ても、観念論者から見ても、市場で最も成功している企業で実践されているプロセスよりも、はるかに良いプロセスである。

オープンソースソフトウェアの出現で、ソフトウェア品質の議論は、ますます複雑になっている。ただし、ソフトウェアが純粋に人間による知的活動の成果物であると考えれば、知的活動の成果物であるソフトウェアも、古典的書物が、印刷技術を利用して、質の良い知識が低コストで大量に複製され、多くの人々の思考や人生に影響を与えているのと同じように、高い品質の機能を低コストで多くの人々に利用可能とされるべきである。

第 8 節　品質問題へのマイクロエレクトロニクス革命の影響[23]

1990 年代に入って、急速に進んだ**マイクロエレクトロニクス革命**によって、多くの製品にマイクロプロセッサが組込まれ、そのマイクロプロセッサ上で動く組込みソフトウェアによって、製品の重要な機能が実現されるようになった[22]。このことによって製品の機能を高度化するとともに、その製品の構造を簡素化し、製造工程を簡素化することで、生産現場における自動化や機械化が容易となった。その結果、製品の大幅なコスト低減が可能となった。

その反面、組込みソフトウェアによる制御は、部品や機器の動作に**非決定性**を導入する結果となり、開発プロセスにおけるテストや、生産現場におけるテストでは、重要な欠陥を発見することが不可能になってきた。開発プロセスにおいては、何よりも欠陥が残り難いプロセスを構築することが重要となり、また、欠陥が残存していても、それが利用時に問題を生じさせることがないような設計を心掛ける**機能安全**を保証する設計が重要になってきた[32]。

このようなテストによる品質の管理や保証が困難になった理由は、非決定性をもったソフトウェア機能の実現度を保証するためには、入力の連鎖に対して、起こり得る全ての組合せをテストすることが必要になることが原因である。しかしながら、ソフトウェアがその内部に包含している条件判定の組合せは、小さいものでも数千、大きいものでは数百万に及ぶ。このことは、小さいソフトウェアでも 2 の数十乗の場合を網羅するテストが必要であることを意味する。

テストによる品質の管理や保証が困難であることから、組込みソフトウェアの機能面での品質を保証するためには、ソフトウェアの設計に重大な誤りがないことを立証しなければならない。このことを厳密に実施するためには、ソフトウェアの設計が正しいことを数学的に証明しなければならない。

これを**形式的手法**と呼ぶ。形式的手法では、数学や論理学を応用して、ソフトウェアの設計に誤りや矛盾がないことを証明する[xxix]。しかし、そのためには、ソフトウェアが実現すべき機能を数学や論理学の方法を用いて記述しなければならない[33][34]。したがって、数学や論理学の専門的な知識が必要不可欠になる。

産業界でソフトウェア開発の実践に携わっている技術者の多くが、そのような数学や論理学の応用に関する知識を持っているとは言えない。そのような現状から来る問題と、ソフトウェアを利用する社会からの要請のギャップを埋めるために、1990 年以降のソフトウェア産業界では、**ピアレビュー**の実践を次善の解決策としてきた。

ピアレビューとは、設計者ではない専門家による設計の妥当性の検証を言う。通常、1 名または 2 名程度の専門家が選ばれ、設計者が作成した設計文書を徹底的に精査して、その設計が妥当であると言えるかどうかを査定する。レビューを行った専門家が「妥当でない」と判定した設計は、ソフトウェアの実現に適用されることはない。

---

[23]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.76-78 参照

機能安全を提案した国際規格である **IEC61508** においても、「形式手法による証明が推奨される」としながらも、現実的な方法として「厳格なレビューを実施すること」を、開発プロセスにおける機能安全保証の最低条件としている[32]。

ただし、レビューの質を確実に管理することは容易ではなく、レビューを担当する専門家の能力や使命感に、結果が大きく左右されることは否定できない。すなわち、ピアレビューにおいて「妥当である」と結論付けられた設計に、重大な問題が残ることを完全に防止することは不可能である。

以上の議論から、マイクロエレクトロニクス革命によって引き起こされた、重要な製品機能を組込みソフトウェアで実現すると言う技術革新は、多くの**非機能的品質特性**については、それらの大幅な改善を低コストで実現する道を切り開いたと言える。

しかし、同時に、それ以前にはあまり問題にならなかった**機能的品質特性**については、テストによる確認が非現実的な手段となった。そのため、形式手法に基づく証明以外に本質的な解決策がなく、機能安全などの品質面で大きな課題を生み出した。

第6章　ソフトウェア技術者と企業の関係

第１節　資本主義における雇用と労働提供契約[24]

資本主義社会において、市場へ製品を供給する役割を担うのは、主として企業である。その企業は、複数の人間(社員や従業員)から構成される組織体である。企業は消費者が必要としている製品を開発し、生産し、市場を通じて消費者へと、その製品を供給する。この製品の開発、生産、販売、さらに販売後の保守サービス等に従事する従業員を、企業は雇用し、組織化して労働に従事させることで、消費者へ販売した製品の代金を売上として計上する。売上から製品の開発、生産、販売・保守、さらに組織の運営に必要な費用を差し引くと、最終的な利益となる。

企業は、そのような利益を極大化するため、一般的に製品の販売量を最大化する戦略を採用する。しかし、製品によっては生産工場への投資額が一定規模を超えると、極端に大きくなる製品もあるため、企業によっては生産量を一定限度内に制限し、得られる利益を極大化する戦略を採用する場合もある。そのような戦略の策定も、企業組織の一員である社員や従業員によって検討され、決定されている。

開発・生産する製品の一部または全てがソフトウェアである場合、そのような製品の開発には、ソフトウェア技術者が必要となる。専門的なソフトウェア技術者が社内にいなければ、ソフトウェアの開発を外部の企業に発注するか、社内の他の専門家がソフトウェアを開発しなければならない。この後者の場合、ソフトウェア開発を担当する技術者の教育を受けた専門は、ソフトウェアの設計やプログラミングではないが、ソフトウェア技術者であるとみなすこともできる。

特に、被雇用者の採用に当たって、その対象者の教育上の専門性を問わない、日本の雇用慣行においては、従業員である技術者は、自分の仕事の遂行に必要な知識を、必要に応じて自ら学び、仕事を遂行する義務を負っていると看做されている。つまり、日本企業においては、ソフトウェアを開発している従業員がソフトウェア技術者であり、ソフトウェア技術者としての高等教育を受け、その知識が認定されている技術者であるとは限らない。

そのような企業(雇用者)と雇用契約を締結し、企業から自分の労働に対する対価を受取る被雇用者は、資本主義社会においては、一般的にその企業との間に自分の労働提供役務に関して契約を締結しなければならない。そのような契約においては、被雇用者がどのような労働に従事すべきかが、明確に記述されていることが前提となる。

そのような被雇用者が提供すべき労働の内容に関する詳細な記述を、職務記述(書)と呼ぶ[xxx]。すなわち、企業と労働者とが締結する契約においては、職務記述が重要な内容となる。その職務記述に記述されている業務を遂行する能力が被雇用者になければ、雇用者から見て被雇用者は業務を遂行できないため、債務不履行となり、契約違反で解雇となる。これが、近代ヨーロッパで確立された労働法の基本的な考え方である。

---

[24]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.81-82 参照

第 2 節　ジョブ型雇用契約とメンバーシップ型雇用契約[25]

上述したような、職務記述を明確に規定して、被雇用者との間で契約を締結する近代ヨーロッパ型の雇用契約を、労働法に詳しい濱口桂一郎は、「ジョブ型雇用契約」と呼んでいる[75]。これは、企業側が必要とする人材が保持すべき技能や能力を職務記述に規定し、その要求に適合した人材を選別し、雇用契約を締結して採用すると言うやり方である。この場合、企業の経済環境等によって、その職務記述に規定されている業務を遂行する必要がなくなれば、その従業員は解雇の対象となることを意味している。つまり、被雇用者の仕事内容を限定した採用となっている。

これに対して、日本の雇用慣行では、職務記述を明確に示さず、どのような仕事に従事させるかは、採用後、企業側の判断で決定できる自由度を残し、雇用契約を締結することが一般的である。この場合、被雇用者である従業員は、どのような専門的な知識があるか、どのような資格を保持しているか、どのような専門教育(高等教育)を受けているかは、ほとんど問題にされない。被雇用者にどのような能力があるかは、採用後、企業側が判定を行い、必要な教育を実施して、実務を担当させた上で評価を決定する。

このような職務を限定せずに雇用契約を締結する日本型雇用契約を、濱口は、「メンバーシップ型雇用契約」と呼ぶ。このメンバーシップ型雇用契約では、被雇用者が従事すべき仕事や労働が明記されないため、被雇用者から見れば、仕事を選択してその企業に入社するのではなく、その企業の一員になることを選択していることになる。つまり、その企業社会(コミュニティ)の一員になることが目的で、雇用契約を締結することになる。このタイプの契約では、職務記述が規定されないので、他の雇用条件も厳密には規定されない。

メンバーシップ型雇用契約においては、雇用条件も企業側が決定する仕事の内容に応じて変化する可能性が高く、契約書で確定することが困難である場合が多い。このため、日本の企業においては、雇用契約に変わって「就業規則」を詳細に規定し、その就業規則の記述に従った条件での労働の提供を、被雇用者に要求することが実践として実施されている。この就業規則には、給与等の条件についても規定されている。

一般的にジョブ型雇用契約においては、職務記述が限定的に記述されているため、給与は、雇用契約に明示されることになる。さらに、仕事の内容(量や質)に変化がなければ、基本的に給与にも変化はない。つまり、同一労働同一賃金に近い原則に基づいて、給与額が決定される。これに対して、メンバーシップ型雇用契約では、職務が限定されていないため、事前に給与額を限定することも困難である。従って、給与そのものは、いわゆる年功賃金体系に基づいて決定される。その年功賃金体系も、基本的には就業規則によって決まる[xxxi]。

日本の司法においては、原則においてはジョブ型雇用契約を前提とした法律に基づき判断が実施されることとなっているが、メンバーシップ型雇用契約の慣行を考慮に入れ、就

[25]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.82-84 参照

業規則を雇用契約の一部とみなす考え方を採用している。これは、明治時代に制定された労働法が、ヨーロッパ諸国の労働法を手本としてまとめられたためであると分析されている。

日本の司法は、理論的にはジョブ型雇用契約を前提としているため、本来であれば雇用契約はその労働が必要な限り有効であるが、経済や技術の変化でその労働を雇用者が必要としなくなったとき、その被雇用者を解雇できる。しかし、現実の雇用形態や慣習はジョブ型雇用契約ではなく、メンバーシップ型雇用契約であり、その被雇用者は職務記述に規定された職務を限定して雇用契約を締結しているわけではない。このため、その被雇用者が現在従事している職務が、必要でなくなった場合でも、その被雇用者を解雇できないとする考え方を日本の司法は慣例的に採用している。

第 3 節　米国企業におけるソフトウェア技術者のキャリア形成[26]

米国社会において企業は、従業員との間に職務記述に基づいたジョブ型雇用契約を締結し、被雇用者による労働提供に対して対価(給与)を支払う債務を負う。ソフトウェア開発に関する労働に対する職務記述としては、要求の分析と定義を行う職務、要求を満たす機能仕様を定義し記述する職務、機能仕様を満足するソフトウェアの設計を実施する職務、設計に従ってプログラムを作成する職務、作成されたプログラムに誤りが残存していないことを確認する職務、開発されたソフトウェアが機能仕様に合致していることを検証する職務、完成したソフトウェアがユーザの要求に適合していることを検証する職務、完成したソフトウェアに問題が発見された時、それを修正・改変する職務等が考えられる。

米国の大学におけるソフトウェア技術者の育成は、主として工学系の学科であるコンピュータ科学関係学科、システム工学系の学科、そして電気・電子工学系学科等において実施されている。それらの学科においては、CSC と呼ばれる標準カリキュラム基準に沿った教育課程が組まれ、実施されている。そのような専門家を育成する大学では、ABET と呼ばれる非営利組織の団体が実施するアクレディテーションと呼ばれる教育課程の認証評価を受け、その認証を受けた学科の卒業生は、社会的に専門技術者としての資格を認定される。

そのような ABET による認証(アクレディテーション)審査においては、産業界の代表と教育・研究界(大学)の代表から構成される評価委員によって、評価を受ける学科のカリキュラム、教育内容、教員の経歴、単位認定基準、単位認定試験の結果、実施されている講義の評価、その学科の卒業生の産業界における評価等が総合的に審査され、評価が決定される。米国社会においては、有名大学のほとんどが、各専門分野別にこのアクレディテーション審査を受審し、その結果を公表している。

アクレディテーション審査に合格していない大学の学科の卒業生は、その学科の教育内容が、例えコンピュータ科学であったとしても、実社会において正式にはソフトウェア技術者としては認められず、ソフトウェア技術者としての職務記述が規定されている職業に就くことはできない。つまり、経済学部で経済を学んだ人が、ソフトウェア技術者として専門的な職に就くことはできない[xxxii]。

このような社会的背景から、米国の若者は、「自分は将来、とのような職に就きたいか」と言う問いを考え、進学する大学の候補を決定する。仮に、「自分は将来、ソフトウェアの専門家としてソフトウェア開発に従事したい」と考えれば、工学系の学部で、アクレディテーション制度で既にコンピュータ科学やソフトウェア工学等の認定を受けている学科を選んで受験することとなる。受験は、SAT と呼ばれる高校卒業程度の学力を測る全国共通テストを受け、各大学の学部・学科別に公開されている受験基準(最低点)と自分の SAT の点を比較し、合格しそうな大学を選ぶ。

大学の受験においては、SAT の成績表以外に、大学が指定する数の推薦状を集め、また

[26]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.84-90 参照

自分がその大学のその学部・学科を志望する理由、将来、就きたいと思っている職業やその理由、自分の夢やこれまでの自分の歩みをまとめた文書、そして与えられたテーマに関する小論文などを大学のアドミッションオフィスへ送付し、審査を受ける。アドミッションオフィスでは、送られてきた書類と、その学部・学科のアドミッションポリシーに基づいて、入学を許可する学生を選抜し、決定する。

この選抜においては、アドミッションポリシーが厳格に適用される。例えば、アドミッションポリシーに、「社会の多様な人種、多様な階層の人々、多様な才能の持ち主」であることを重視して、そのバランスの取れた組合せを重視した教育を実践できるようなポリシーが標榜されているとする。その場合、男女の比率、社会における各人種の人々の構成比、各階層に属する人々の構成比、受験者の特技や特別な才能(例えば、プログラミングコンテストの優勝経験など)を考慮した選抜が実施される。すなわち、必ずしも SAT の成績上位者から合格が決まるわけではない。近年では、特にマイノリティと呼ばれる社会の少数派(何らかの障害をもつ人々や、社会の構成比がきわめて少ない民族の出身者など)を優先的に受け入れる大学が増加している傾向にある。

大学への入学後の教育は、基本的に大学生は大学寮に入って生活しながら勉強するため、生活全体が大学での寮生活を基本とする。このため、大学での教育は、教員からの説明を聴いて学ぶと言う、講義形式の勉学もあるが、特に、同じ大学の同じ学部で学び、一緒に生活している同じ寮の学生から学ぶことが多いと言われている。特に、教員による講義は、主として学生間の議論や、教員による基本的なことに関する説明が主であり、ほとんどのことは、教科書を自分で読み、考え、学んでゆかなければならない。この時、同じ寮で生活を共にしている学生から、学び、獲得する知識の量は多い。

期末試験では、単位取得のための基準線が決まっているため、学生達はその基準を満足するように計画的に学ぶことが要求される。また、将来の仕事を考えると、その仕事に強く関係する科目については、可能な限り良い成績をとっておくことも重要である。従って、学生は受講する講義科目をある程度限定せざるを得ない。また、単位を取得するためには、一定回数以上、講義に出席しなければならないので、講義を休むことは実質的に困難になっている。講義に出席するためには、予習が必要になるため、その意味でも、それほど多くの講義を履修することはできない。

大学生の場合､6 月から 9 月までのほぼ 3 カ月間近くの年度末休暇がある｡多くの学生は、この間、将来の自分の仕事について学ぶため、インターンシップに行く。そのためには、その半年前ぐらいからインターンシップ先の企業を選び、数か月前にはその企業に依頼状を送付して、受け入れを打診しなければならない。各企業は、寄せられてくる依頼状の中から、受け入れる学生を選び、受け入れの準備をしなければならない。

インターンシップ期間中に学生に与えるべき仕事を決めるのである。インターンシップ期間は 1 カ月半から 2 カ月半程度なので、その間に完了することが可能な仕事でなければならない。

学生によって異なるものの、3 年間、場合によっては 4 年間、同じインターンシップ先で仕事を経験する者も少なくない。多くの場合、そのような学生は、その企業のその部門に就職する。つまり、企業側から見ても、インターンシップを受け入れることは、ある学生を長期間にわたって観察し、その能力を評価することができると言う利点がある。

学生側から見れば、将来自分が就職するかもしれない企業の現場で仕事をすることで、その企業でどのような仕事を、どのような環境で、誰とするのかを予め経験できる良い機会である。インターンシップ中、仕事はしても研修と言う名目であるため、給料は出ない。ただし、多くの企業はその間の生活費を提供する。

学生は、就職の時期が迫るまでには、どのような企業で働きたいかを決めている。そして、そのような目指す企業に対して、推薦状や就職の希望を述べた書面、さらに経歴書(学業やインターンシップの経験なども記入されている)等を送付する。企業側では、インターンシップ時に、これはと思う学生には、「将来、ここへ来て働かないか」と打診している例も多い。そのような学生も含めて、受取った書類から採用したい人材を選別する。

このような選抜過程を経て、就職した人材は、当初、「訓練生」として、メンターと呼ばれる先輩指導者の下で様々な仕事を経験させられる[xxxiii]。ソフトウェア技術者の場合、この訓練生の期間は、半年から 1 年程度である。

この訓練生の期間を終え、一人前の技術者として仕事を任させると、その技術者は「准技術者」と言う職位を与えられ、管理者が与える仕事を一人でこなすことになる[xxxiv]。この場合、命令を受けるのも自分一人であるが、管理者に報告を行うのも自分一人である。

准技術者として仕事をする数年間は、仕事を覚える期間でもあり、その意味で実践的な教育を受けている教育機関でもある。教育を受けながらも、仕事をし、成果を上げることが要求されている。

この准技術者としてのキャリアを成功裏に終えることができると、その技術者は、(正)技術者に認定される[xxxv]。そのような(正)技術者は数名の准技術者を指揮監督する任務が与えられ、自分の仕事の成果だけでなく、自分が指導している准技術者達の仕事の成果に対しても責任を負うことになる。

この(正)技術者としての役割を問題なく果たし、技術的にも優れたものを身につけている人は、専門家として専門職の道を歩む例が出てくる。そのような人の場合、(正)技術者の職位から、「スタッフ」技術者へ昇進することとなる。米国のエクゼンプト制度の中では、このスタッフ技術者は、管理職と同等の待遇と考えられることが多い。職場よっては、このスタッフ技術者は、専門技術を指定してはいるものの、その専門には関係なく、数人の(正)技術者を直属の部下として実質的に管理する立場にしている場合もある。逆に、スタッフ技術者は、特定の専門を限定した高度専門職であるとして、部下を付けないが、特別な専門に関する業務だけに専念させる体制を採用している職場もある。この場合、待遇は管理職と同等であっても、仕事のやり方は、(正)技術者と大きく変わるものではない。このスタッフ技術者がさらに専門分野に関する知識を蓄積し、その組織の技術的な発展に大きく貢

献するようになると、「上級技術者」へと昇進する[xxxvi]。

多くの(正)技術者は、ある程度の経験を蓄積すると、プロジェクトの計画・実施に責任を持つ、「ライン管理者」に昇進する[xxxvii]。ライン管理者は、自分でプロジェクトを計画し、提案する。提案が認められれば、その実施に必要な技術者である部下の採用を行い、プロジェクトの実施を準備する。さらに、それらの部下を指揮してプロジェクトの実施管理を担当する。そのような管理は、階層構造を構成する例が多く、末端のライン管理者から始まって、少しずつ階層構造の上位の仕事を担当するようになって行く。

米国の技術者は、大きなプロジェクトが終了すると、別の企業へ移籍する例が多い。これは、「有名なプロジェクトに参加した」と言う履歴が、自分をより高いレベルへ引き上げる効果を持つからである。

つまり、自分の昇進や評価を考えると、より重大なプロジェクトに参加することが、より有利であると言う現実的な判断からである。このことは、誰にとっても同じ条件であるため、時として大きなプロジェクトを完了したチーム全体が、その全体を指揮したプロジェクトマネージャとともに、ある企業から別の企業へ移籍すると言う例もある。

第 4 節　日本企業におけるソフトウェア技術者のキャリア形成[27]

本章において、これまでも議論してきたように、日本における雇用契約はジョブ型ではなく、メンバーシップ型である。すなわち、技術者であるかないかに関わらず、被雇用者の雇用を決定する時点で、雇用者と被雇用者の間には、職務に対する限定がほとんどない。

このことから、被雇用者は、雇用者が実施しているいかなる業務・作業であっても、雇用者の命令があれば、その業務・作業に従事しなければならない。また、雇用者は被雇用者が指定された業務・作業の遂行に必要となる特別な知識や技能があれば、それを教育する義務を負うことになる。

このような雇用慣行の中で、企業側は技術者の採用に当たって、特別な理由がなければ、職務の遂行に特別な学部学科の卒業であることは要求されない。さらに特別な資格(国家試験)を必要としない限り、採用の対象となる人材の出身学部学科を問題にしない。職務の遂行に必要な知識や技能は、入社後の社員研修で身につけさせることができるからである。

さらに、技術者の視点から見れば、日本の社会では昭和に入って作られた様々な制度が、企業を頻繁に移動しないことを前提に作られているため、企業を変わることで不利益を被る結果となる例が多い。このため、従業員(技術者を含む)は、一度、特定の企業に籍を置くと、簡単にはその企業を退職して、他の企業に移動する選択をしない習慣が根強く残っている。

そのような雇用環境の問題もあり、企業は従業員に対して自社特有の業務遂行形態に適合させた知識や技能の修得を要求する。これは、従業員にとっても、長期的に安定的な雇用を保証する上で有利に働くことから、それを積極的に受け入れることになる。

このことから、技術者であっても、特定の企業の業務形態に合わせた、極めて局所的に最適化された知識の集合を学ぶ結果となる。従って、技術者の能力は、その企業内に留まる限りは、有効に働くものの、一般的な知識ではないため、他社に移動した場合にも、その人材の能力が活きるかどうかが全く予想できないこととなる。

日本の技術者は、そのような雇用慣行の中で、どちらかと言えば、特定企業の業務内容に特化した知識を蓄積しながら、見習い期間、技術者助手、准技術者として、徐々に知識を蓄積してゆく。そのようにして 5 年程度の育成期間を過ぎると、各技術者は、それぞれの配置された部門において一人前の正技術者として、仕事をするようになる。正技術者になると、数人の助手や准技術者を指導する役割も担うようになり、それらの部下の人材育成にも責任を持たされるのが一般的である。

職務記述が限定されていない日本の雇用制度では、被雇用者は、一定の期間を経過すれば、管理職として部下の監督指導に当たるようになる。管理職でも、その地位が低ければ、部下の監督指導も、正技術者として実施してきたことと大きな違いはなく、育成しなければならない部下の数が、数名から 10 名程度になるだけである。つまり、数名の正技術者も自分の部下として監督・指導することが要求される。

---

[27]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.90-93 参照

そのようなことから、管理職への昇進は、単に正技術者としての能力だけでなく、部下の指導能力も問われるため、一般的に一定の年齢を超えていることが条件とされる。つまり、部下よりも若い上司は、ほとんどいない。

ところが、管理職を長く経験すると、部下の指導・監督の能力と、技術者個人としての能力に大きな違いが生じるため、より高いレベルの管理者としての仕事を指示される人材と、そうでない人材に分かれてくる。この段階になると、部下を指導・監督する能力は、多くの場合、その人が生まれ育った環境の要因が強く働くので、その人の年齢よりも性格や適正で能力が決まるようになる。

従って、相対的に若い部長の下に、年上の課長が部下として勤める例も稀ではなくなる。指導・監督の能力には、一般的にコミュニケーション能力と呼ばれる、他人との意思疎通を効果的に図れるかどうかの能力が大きく影響すると言われる。その能力と物事を迅速に判断する能力によって、それぞれの管理者のキャリアは、異なってくる。

日本企業の給与体系は、基本的に一定年齢以上の人材は、管理職に昇進することが前提となっているため、技術者の職位に留まる限り、給与は頭打ちになる。管理職の給与は、管理職のレベルに相応した給与体系になるため、より上位の管理職に就くことがより高い給与を得ることを意味する。

ところが、管理職のレベルが高くなればなるほど、その地位に着ける人材の数は減少する。つまり、ポストの数が少なくなるのである。このため、一定の年齢までに、特定のレベルのポストに昇進できない人材は、その組織内でのそれ以上の昇進を断念することになる。

そのように昇進を断念した人材は、日本企業の場合、その企業のその組織が直接関係する関係会社へ移籍することになる例が多い。この場合、移籍対象になる企業の規模は、従来勤務していた企業の規模に比較して小規模なことが多く、このため、移籍後のポストは、一般的に移籍前のポストよりも高くなる傾向がある。

また、従来であれば、ポストが高くなる分、その人材が得る給与水準も高くなることが多かった。この傾向は、1990 年代の後半から、徐々に変わりつつあり、最近では移籍前の給与水準を下回る例も出ている。

このように見ると、日本企業における技術者は、受けた専門教育には関係なく、入社当初は技術者としての仕事をする。そして、10 年から 15 年で管理職となり、その後の 15 年から 25 年間は管理職としてのキャリアを歩む例が多い。これは、一般に専門職として生きる米国の技術者のキャリアとは、大きく異なっていると言える。

また、日本企業では、採用において出身学部学科を問わないため、専門分野の基礎的知識が不十分であるにもかかわらず、専門的な業務・作業に従事している人材の数も多い傾向がある。このことは、仕事の成果の質が、知識不足によって米国企業の場合に比較して低くなるリスクが高いことを意味している。

第 5 節　日米企業におけるソフトウェア技術者選抜の違い[28]

日米の企業におけるソフトウェア技術者の選抜プロセスは、2 国間の雇用制度の違いと人材活用制度の基本的な違いによる影響で、大きく違っている。両国における大学入試選抜方式と同様に、米国では比較的長期にわたる選抜プロセスを経た技術者の選抜が実施されているのに対して、日本では短期間で技術者人材としての学生を選別する方式が採用されている。これは、日本社会における雇用慣行が、メンバーシップ型雇用契約に基づいていることによると言える。

前節で議論したように、日本の企業ではメンバーシップ型雇用契約方式を採用しているため、採用対象者の出身学部学科を問わずに人材募集をする例が多い。特に、日本のソフトウェア産業におけるソフトウェア技術者(一般的には SE 職と呼ばれている)の採用においては、学生の所属する学部学科を限定する例は少ない。

ただし、採用試験においては、適性検査と称する知能テストと一般教養を問う試験、さらに回答者の性格や精神構造を分析する心理テスト等を課す例も多い。これらのテストは、専門的知識を問うものではなく、ソフトウェア技術に関する知識があるかないかは、ほとんど問題にならない。

このような適性検査で、企業側が入社希望者の何を評価しようとしているかを考えると、その目的はその企業の社員として働くための資格や資質を備えているかどうかを判定するためと言える。例えば、グループでの仕事が多く、協調性を重視する企業では、知的な水準が一定範囲内にあり、似たような一般知識の蓄積があり(コミュニケーションがとり易い)、他人との協調性に富む性格の人材を望む傾向が強い。知能のレベルが他の候補者に比較してかなり高くても、協調性がなく、一人で黙々と仕事をする傾向が強く、自分と異なる意見を持つ者との柔軟な意見交換ができない人材は、敬遠される傾向が強い。

このような傾向は、仕事の責任を明確にして、仕事の成果を明確な指標で評価し、その評価結果に基づいて給与や昇進を決定する米国の職場では、ほとんど見られない。日本の職場では、個々人の仕事の責任は、職務規定が明確に規定されていないことから、明確にできない例が多い。仕事の責任は、それを実施しているグループ全員によって分担されているとする認識が強い。

従って、グループのメンバーの一人が何かの問題を解決できずに仕事が停滞していると、他のグループ・メンバーで余裕のある者が、その仕事の遂行を援助すると言う例が一般的である。この例のように、日本の企業ではチームワークによる仕事の効果的な遂行を重視しているため、協調性の高い人材を採用したがる傾向にある。

このように日本企業での仕事の仕方が、米国企業での仕事の仕方と大きく異なっているため、ソフトウェア技術者の採用プロセスも大きく異なっている。つまり、採用候補者が大学で何を学び、どのような知識を修得したかはあまり問題にならない。

これに対して、採用候補者がどのような大学で学んだかは、採用・不採用に大きな影響

[28]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.93-97 参照

を与える要因となる。このことは、候補者が就学した大学が、どの程度の知的水準を持った学生を受け入れている大学であるかが、その候補者の基礎的学力を知る上で重要な手掛かりとなるからである。一般的に、入学が難しい大学に入学するためには、入学試験において他の高校生と比較して高い得点を得なければならず、そのためには高い基礎学力が必要になる。

高い基礎学力は、一般に、社会で言うジェネリックスキルの高さと正の相関がある。つまり、高いジェネリックスキルを持った学生を採用したいとすれば、より入学が困難な大学で学んでいる学生を選抜することが、企業にとってリスクの小さな戦略になる。

高いジェネリックスキルを持った人材は、一般に学習能力が高く、企業へ入社した後も、新しい専門知識を身に着け、適切に仕事をこなしてゆく能力が高い可能がある。ただし、このことは、それらの人材が就く職務が、高い専門性を要求する特殊な仕事ではないことが前提である。

日本の企業は、採用者決定プロセスにおいて、適性検査、数回にわたる面接・口頭試問を実施し、採用者を決定する。このとき、適性検査は採用候補者の絞り込みの手段であり、採用者を決定するためのものではない。

採用者の決定は、あくまで対面式の面接を重視したものである。面接では、志望動機や、入社後に就きたい仕事の希望、入社後に何を達成したいかの夢、自分の性格や特徴、大学時代をどのような考えで過ごし・学んできたか、等々について言葉で、どこまで表現できるかを評価する。

自分自身を適切に表現できた者は、コミュニケーション能力が高いと評価され、そうでない者はコミュニケーション能力が低いと評価される。また、候補者の答えの中に、その企業の文化や思想に適合するキーワードが入っているか、またはその逆はないかなどを検討することで、その候補者が企業風土に適合した人材であるかどうかを事前評価する。

そのような個人思想的背景が、企業の文化や思想的背景に適合するかどうかを判定することは、かなり困難である。そのため、面接を重ねることで候補者を淘汰しながら選考を進める方法が採用されている。とは言え、この選考プロセスは、1 カ月から 2 カ月程度で終了する。

これに対して、米国企業における大学生人材の選考プロセスは、長期にわたる例が多い。一般に、短くても 1 年間、長い場合には 3 年間程度をかけている。逆に、既卒で、他社での就業経験のある専門家の場合、選考プロセスは一般に短く、簡単であることが多い。

そのプロセスの違いは、候補者に実践経験があるか、ないかの違いによる。実践経験のない、または少ない学生候補者の場合、その学生がどの大学のどの学部学科で何を学んだかを調査し、それが職務記述に書かれている知識に適合していれば、次に、その学生がどのような企業でインターンシップを経験していたかが問題にされる。インターンシップ先が、自分達と同じ産業分野である場合、当該企業の指導責任者にその学生の評価を問い合わせることから始める。

大学の指導教員の評価、インターンシップ先の企業の指導担当者の評価、大学で一緒に学んでいる他の学生による評価などを総合して、その学生の専門家としての能力評価が確定する。その能力評価が、職務記述に書かれた仕事の遂行に十分なものであると判断されれば、学生は採用となる。

逆に、情報が不完全で、その学生の専門技術者としての能力が未知であり、職務記述に書かれた仕事が遂行できるかどうか分からないとき、その学生は不採用になる。この過程で、最も重視される意見は、企業内に在籍する知人の推薦である。企業内に推薦者が居ない場合、その学生の採用は極めて難しい。その意味でも、その企業におけるインターンシップの実績は、学生にとって極めて重要である。

既卒の社会人専門技術者の場合は、その能力の評価は比較的容易である。特に、前職で上司をしていた管理者たちの候補者に対する評価を聴くことで、その技術者の能力はかなり正確に判定できる。

ソフトウェア技術者の場合は、その候補者がこれまでに経験してきた仕事が書かれた経歴書に基づき、経験した仕事がどのようなものであり、候補者が何を達成してきたか、そこでどんな問題に遭遇したかなどを調査する。そのような関係者へのヒアリングを実施することで、候補者の全体像を把握することができる。

そのようにして得られた情報に基づき、職務記述に書かれた仕事に従事し、それを全うできる能力があるかどうかを判定することで、候補者の採用、不採用を決定することができる。一般的に募集時に公表された資料に、候補者が受けていなければならない教育、知っていなければならない知識、望ましい実践経験等が記入されているため、候補者は自分自身がその基準を満足できているかどうかは、ほぼ判定可能である。

ただし、その人材募集に応募する人材候補は自分一人ではないことから、競合する候補者がいることも想定して応募している。このため、自分の過去の実務経験の内、何に焦点を当てて職務経歴を記述するかは重要な問題になる。

このような社会人で実務経験のある専門技術者の場合であっても、人材を募集している企業内の推薦者の意見は、極めて重視される。米国社会では、専門技術者達は、専門家達が構成する団体を作り、そのメンバーになることで、専門家であることを客観的に示そうとする。

そのような専門家団体に所属するメンバーのうち、人材を募集している企業に勤務している者が、候補者の推薦人になることは一般的である。そのような社内の推薦人による、候補者の評価に関する意見は、その候補者の能力の判定や採否の決定において、非常に重要な役割を担うことが多い。

第 6 節　米国企業におけるエクゼンプション制度[29]

専門技術者の雇用において、日米の企業間での違いの中で、最も著しいものがエクゼンプション(exemption)制度であると言える。大学以上の高等教育機関で専門を学んだ技術者等の専門家が米国企業に入社すると、エクゼンプトとしての待遇を保証される。エクゼンプトは、終身雇用を保証されている社員である。

そのため、基本的にレイオフ制度は適用されない[xxxviii]。エクゼンプトの場合、終身雇用であるため、本人が退職を申し出るまでその企業に留まり、働くことができる。米国企業に非常に高齢の社員がいるのは、この制度のためである。ただし、年功賃金体系は採用していないため、給与は職務内容と業績評価によって決まる。

エクゼンプトの場合、毎年の仕事の内容は、雇用者である直属の管理者と本人が話し合いを行い、その合意で決定する。この話し合いでは、前年の仕事の総括が行われ、それに基づいて成果の評価が決定される[xxxix]。この前年度の仕事の成果に対する評価と、次年度の仕事の難しさ(目標の達成がどの程度困難か)によって、給与額が決定される。

前年度の仕事の成果に対する評価が高ければ、次年度の仕事の内容はより難しいものとなることが一般的で、その分、給与も高くなる。逆に、前年度の仕事の成果に対する評価が低ければ、次年度の仕事の内容は、より目標の達成を容易にするため、より簡単なものになり、その分、給与も低くなる。

仕事の目標は、達成度の評価を客観的に実施することができるようにするため、数値目標にすべく努力される。ただし、それは一般的な傾向であって、全ての仕事の目標を数値化できるわけではない。その場合でも、達成されたか、達成されなかったかの判定をかのうなかぎり客観的にできるようにすべく、工夫がされる。

このような評価と目標設定を繰り返すことで、毎年の仕事を決めてゆくのがエクゼンプト達の仕事の仕方である。このため、人生の階段のどこにいるかによって、必要な生活費が違っているため、その人生の段階に合わせて、仕事を計画し、それに合わせて給与も決定できるようになっている。

ただし、エクゼンプト社員の場合、仕事は計画時に綿密に考えられ、記述されるため、その目標達成のためにどの程度の負荷が必要になるかもある程度予想できる。負荷が大きくなれば、労働時間も長くなる例が多い。給与の決定には、そのような負荷の予想も踏まえて給与額を確定する。

従って、エクゼンプト社員に対しては、時間外労働の手当は支払われない。仮に、順調に進んで、全く時間外労働をせずに仕事を終わらせることができても、逆に、週 20 時間の時間外労働を連続的に実施しなければ仕事を終わらせることができなかった場合でも、その社員に支払われる給与は、事前の協議で決定された額である。

ただし、労働の負荷に対する見積りが悪く、目標達成のために多大な時間外労働が必要になっていた場合には、評価時にそのことを考慮に入れて、高い評価とする例も多い。評

[29]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.97-100 参照

価が高ければ、次年度の給与は必然的に高くなる。

米国企業のエクゼンプト社員は、日本企業の正社員に似たような部分もある。それは、管理職への昇進である。例え技術者として企業へ入社したとしても、管理職としての能力が高ければ、企業はその社員を管理職に昇進させ、部下の管理をさせる道を模索する。

米国企業の場合、全てのエクゼンプト社員に対して、将来、管理職としての道を歩む希望があるかどうかを事前に調査している。このため、管理職に昇進する社員は、最初からある程度絞り込まれている。

そのようなエクゼンプト社員の中でも、仕事に恵まれ、能力も高く、やる気がある社員は、管理職へと昇進してゆく。この点は、日本企業における正社員の昇進に似ている。

企業による違いもあるが、一般に米国企業の場合には、専門家であるエクゼンプト社員には定年制がないが、エクゼンプト社員であったとしても管理者に昇進すると、定年制の適用を受ける例が多い[xi]。米国企業の経営者(CEO)の年齢が、日本企業の社長や会長の年齢に比較して若い傾向があるのは、この定年制のためである。

特例はあるものの、CEO の定年を 60 歳前後に設定している企業は多い。さらに、CEO として成果を出すためには、8 年から 10 年程度の期間が必要である。このため、遅くとも CEO に就任する 10 年前には、経営会議のメンバーになっている必要がある。

このように計算すると、人生のかなり早い時点で、管理職に昇進しなければ、経営者になることはできない。明文化された定年はなくても、各マネジメントのレベルで実質的な定年が決まっているとも言える。

一般的な技術者の場合は、エクゼンプトとしてキャリアを歩み始め、人生のある時点で管理職に昇進しても、一定年齢に達した時点で管理職を辞し、再び、専門家として生きる道を選ぶ社員も少なくない。この場合、勤めていた企業を退職し、新しい企業へ専門技術者として入社し直す例もあるが、その企業に残り、専門技術者として第二の人生を歩む例も少ないわけではない。

特に、技術者としての能力が秀でている場合、企業はその従業員をより高い地位の技術者として処遇し、他社へ移籍する選択をさせないようにすることもできる。フェロー制度などがそのような例である。

フェロー制度は、通常のエクゼンプト社員が、新しい仕事の提案を企業へ提出し、管理者の承認を受けて仕事を予算化し、その予算の範囲で仕事をしなければならないのに対して、企業側へ仕事の提案をせずに、一定の予算の範囲内で、自由に仕事を実施することができる身分を保証している。このため、自由に仕事を選び、好きな仕事に没頭できると言う利点がある。また、他の社員からも、フェローとして認知されるため、本人の名誉欲も満足できる場合が多い。

近年、日本企業においてもエクゼンプト制度や、フェロー制度の導入を検討する企業が出て来ていると言われている。ただし、日本の雇用慣行の中で、このような制度を導入しても、意味のある結果が出るかどうかは、分からない。

特に、エクゼンプト制は、実質的に定年制と年功制が機能している日本の雇用慣行の中では、定年制をなくすことで企業側の労働コストが極端に膨張する結果になる可能性も高く、現実的な対応とは言えない。また、職務記述が曖昧な日本企業の雇用慣行の中では、エクゼンプト制における評価と次年度の仕事の計画策定、そして給与額の決定は、極めて困難な問題をはらんでいる。

第 7 節　日米におけるソフトウェア技術者の知識獲得プロセス[30]

米国社会においてソフトウェア技術者として仕事に就くためには、4.3 節に述べたアクレディテーション制度で、ソフトウェア技術者を育成できると認定された大学の学部・学科を卒業することが大前提となる。そのような大学の学部・学科を正式に卒業していない者が、ソフトウェア技術者の募集に応募することは、社会的には経歴詐称になる。

そのようなアクレディテーション認定を受けた大学の学部・学科で学ぶことは、専門技術者として働く上で重要である。すなわち、ソフトウェア技術者に求められる基礎的な知識を、標準カリキュラムに沿ったカリキュラムを通して、その教育を担当できると評価された教員による教育・訓練を得ていることを意味しているからである。

そのような大学の学部・学科のカリキュラムでは、基礎的な知識として、離散数学の基礎、確率統計学の基礎、論理学の基礎、倫理学の基礎、などの基礎的な科目のほかに、プログラミング言語とプログラミング、データ構造とアルゴリズムの基礎を学ぶ。さらに専門的な知識として、コンピュータハードウェア、オペレーティングシステム、ソフトウェア工学、データベース、コンピュータネットワーク、人工知能などについて学ぶことが要求される。

さらに、実践的なスキルを身に着けるため、プログラム開発演習、ソフトウェア設計演習、ソフトウェア開発プロジェクトの演習(実験)、そして一部の大学の学部学科では、卒業前の最終的な演習(cap-stone project)と呼ばれる長期の実践的ソフトウェア開発演習を課される。米国の大学生は、これらの演習科目を通して、現実の問題に近い実践的なテーマに取り組み、社会に出た後の仕事の仕方を学ばなければならない。特に、卒業前の最終的な演習では、企業から提供される現実の問題に取り組まなければならない例も少なくない。

このような高等教育課程での学びを行うと同時に、多くの学生は、年度末の各休業期間中に、企業において実際の仕事に従事する研修に参加する。この長期にわたる研修(internship)では、研修先の企業が決める現場指導者の指導の下で、その企業の社員と同じ作業に従事し、ソフトウェアの設計、プログラミング、そしてテストなど、実践的な作業に従事する。

その結果、自分達が大学で学んでいる知識が現場でどのように役立つか、また、大学で学ぶことのできない知識にどのようなものがあるのかを学んでゆく。企業側は、そのようなインターン制を利用した学生の研修を活用して、個々の学生の適性や能力を判定する。

大学教育を終了した者は、ソフトウェア技術者の見習いとして、企業において実際の仕事に就く。多くの学生は、インターン制での研修によって、ある程度就業経験のある職場で、ある程度経験のある仕事に従事する。

従って、そのような新入社員に基礎的な教育はほとんど必要がない。情報の管理規定のような規則であっても、インターン時の研修で、説明を受けており、内容の再確認は必要があるかも知れないが、基本的には熟知している内容である。このようなことから、新入

[30]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.100-106 参照

社員であっても、最初から一般の技術者と同様に、仕事が与えられる。

見習い期間中は、先輩社員であるソフトウェア技術者が、指導員(メンター)として新入社員の指導に当たる。インターンシップの研修である程度慣れている仕事・作業であっても、研修学生として担当する場合と、社員として担当する場合では、責任の重さが大きく異なっており、仕事の仕方も自ずから違っている。この点を、メンターである先輩社員が必要に応じて確認することで、適切な仕事のやり方を学んでゆくのが、この期間である。

見習い期間を終えた技術者は、正式に准ソフトウェア技術者に任命され、全てを自己管理できる立場になる。仕事は、管理職から与えられるが、その仕事をどのような手順で実施してゆくかは、自分で決め、その計画に基づいて作業を進めることができる。必要なことは、管理職への報告義務だけである。

管理者は、特に問題が発生していないと判断すれば、報告を聴くだけである。管理者が、問題が発生していると判断すれば、それを確認するために質問をし、自分の意見を述べることもある。また、他のメンバーへの進捗報告で、質問を受け、他のメンバーから自分の意見とは異なる意見を聴くこともある。

このような管理者や同僚とのコミュニケーションを通して、ソフトウェア技術者は、実践的な知識を少しずつ身に着けてゆく。上司や同僚からの質問に適切な答えを述べるためには、常に、何が問われそうかを考え、その問に対する基本的な回答を考えておく必要がある。

様々な場面で、説得性のある回答を準備するためには、予め確認しておくべきことも多々存在する。そのような下調べのプロセスを経て、ソフトウェア技術者は学び、新しい知識を獲得してゆくのである。

そのような状況に対応するため、米国の技術者達は細かな業務日誌(ログと呼ばれる)を、毎日、記入することが習慣づけられている。自分が、今日、何を考え、何をして、どのような結果を得たかを、事細かに記述するのである。

このような業務日誌は、上司や同僚から、進捗会議で質問を受けた時に、事実に即して回答するために極めて重要であるとともに、自分の犯した過ちを見なおし、将来の作業における同様な失敗を未然に防ぐための仕事の手順を考える上で、極めて重要な材料を提供してくれる。さらに、上司との仕事の成果のレビューを実施するとき、失敗の原因が自分の間違いによるものなのかどうかを確認する資料としても大変有効である。

業務日誌の内容においては、定量的に記述できる情報は、可能な限り定量的に記述するようにしている者が多い。仕様書や設計書などの場合は、文書の行数やページ数を記載しておくなどである。

プログラミングに関する記述の場合には、プログラムの行数を記載することを習慣としている技術者が多い。さらに、テストに関係した記述においては、作業時間やテストで確認すべき内容の項目数とテストで発見した誤りの数を記述する例が多い。このような定量的なデータを記しておくことで、後日の分析を容易にする工夫がなされている。

プロジェクトに関係した全ての作業が終了すると、各技術者はある時点で、自分が記載した業務日誌を読み直し、何が悪かったか、何が良かったかを、詳細に見なおしてゆく。これによって、次の仕事では、どのようなことに注意して作業を実施すべきかを検討するのである。

このような検討を、プロジェクトの度に実施してゆくので、技術者達はプロジェクトの経験を重ねれば重ねるほど、経験知を蓄積し、有能な技術者に成長してゆく。米国企業で、実践経験の豊富な技術者を高く評価する傾向があるのは、このような背景があるからであると言える。専門家は、仕事を通じて学び、成長してゆくのである。

特に、米国企業では職務記述がある程度明確に書かれるため、仕事の成果が明確になった時、自分が担当した仕事が成功したかどうかは、多くの場合、明確に分かることが多い。仮に、仕事が失敗であったことが分かれば、その失敗の原因が自分の実践能力の不足、自分の理論的な知識の不足、自分の注意不足など、何が原因で失敗が起こったのかを上司と分析しなければならない。

そして、自分に問題がないことが確認されれば、組織または他のプロジェクトメンバーの仕事に問題があったこととなる。仮に、仕事が失敗であるとされても、その責任の所在が自分になければ、次の仕事においても給与水準を維持または改善することができる。

このように仕事が失敗に終わったかどうか、その失敗の原因が自分にあったかどうかを詳細に検討することで、技術者達は自分の実践的な能力を少しずつ高めてゆくのである。従って、高度な仕事を経験した者ほど、学ぶものが多く、学んだものの多いものほど、能力が高いと評価され、より有利になる。その意味でも、どのような仕事に就けるかは、技術者にとって極めて重要に問題であり、良い仕事に就くためには、現在の仕事で高い評価を受けることが前提条件になる。

このような米国のソフトウェア技術者と比較すると、日本のソフトウェア技術者の成長には、明確なプロセスが示されていない。日本のソフトウェア技術者が、大学で何を学んだかを問われていないことは、技術者としての基礎知識を問われていないことを意味している。

企業は、特定の技術者候補をその基礎的な学力(主としてジェネリックスキル)によって評価し、その評価の高いものを優先的に採用する。特に、メンバーシップ型雇用契約を採用している日本企業では、評価対象の候補者を企業内で教育し、成長させ、企業メンバーの一人として育て上げられるかどうかが大きな関心事になる。

そのため、日本企業のソフトウェア技術者は、入社時から退職するまで、継続的にソフトウェア技術者であることを前提としてはいない。むしろ、その企業の従業員として、一生をその企業のメンバーとして捧げることが可能かどうかが問題である。

つまり、一時的にはソフトウェア技術者としての仕事に従事したとしても、何年か後には、別の仕事に就いている可能性を否定できない。また、同じソフトウェア関係の技術者と言っても、基本ソフトウェアからアプリケーションまで、様々な専門分野に分かれてい

る。

一般に日本の企業では、従業員は新入社員の時代には、特定分野の専門家として仕事をしても、経験が増えるに従って、技術的な要素が徐々に減ってゆき、管理的な要素が徐々に増加する傾向が著しい。さらに、最近では、技術に特化した仕事は、専門家した派遣社員や系列企業の社員に任せる例も増加している。

つまり、日本企業の従業員の場合、特定の専門技術に特化した、技術者として成長することは余り期待されていない。むしろ管理者として技術者などを管理する能力を身につけることが期待されている。

このような状況では、技術者個人としては、実践的な仕事に精通し、与えられた仕事をしっかりとこなすための知識を身につけなければならない必要性は低い。従って、日本のソフトウェア技術者は、一般に米国のソフトウェア技術者のように業務日誌をつける者が少ない。

それは、似たような仕事は、2 度としないからであろう。来年は、今年よりも、より管理的な色彩の濃い仕事になるはずであり、現在の技術的な仕事は、その管理的な仕事を行うための予備的な知識を獲得するためのプロセスであると考えられるからである。

業務日誌をつけないことは、仕事に失敗をしても、その原因がどこにあったかを分析することは不可能であることを意味する。日本企業では、そのような詳細さで、仕事の結果を分析することは必要ないのである。

もちろん、仕事が成功したか失敗したかは、仕事が終われば分かることである。しかし、終わった仕事が成功であったのか、失敗に終わったのかの判定に拘泥しなければ、多くの仕事は成功でもなく、失敗でもないグレーゾーンに落ちる。そして、著しい成功例と失敗例だけが、成功、失敗の評価を受けることになる。

日本では、失敗を明確にして、そこから何かを学ぶと言う、プラグティズム的方法が根付いていない。このため、失敗は一般に隠される傾向にある。すなわち、その仕事に参画した人々の間では失敗であったとの認識が共有されていても、組織としては失敗と認識されないように、詳細が隠されるのである。

逆に、成功例の場合は、皆が学ぶべき手本としてその詳細が分析され、多くの人々に周知されることが多い。つまり、成功例だけが残されるのである。

一般に、人間は成功例からは何も学べない。成功には、幸運の影響が強いからである。しかし、失敗には、幸運とは無関係に必然的な理由がある。それを見つけ出して、同じ行為を繰り返さないようにすれば、同じ失敗を繰り返すことはない。

だからこそ、米国の技術者達は業務日誌を詳細に書き、分析するのである。しかし、日本企業には、その必要性がないのである。

このような実践が行われていることは、日本では技術者個人個人のレベルでは、仕事の改善が実践され、知識の蓄積が行われている例があるとしても、組織的な観点から仕事の改善が取組まれ、組織的な知識の蓄積が実践されているとは言えない。このことは、技術

者個人をとっても、技術者が仕事から知識をしっかりと獲得してゆくプロセスが実践されているとは言い難い。

特に、ソフトウェアのように専門性が高く、高度な知識の蓄積を要求する分野においては、個々の技術者における知識の獲得プロセスを確立し、それを仕事の場面でも実践できるように教育・訓練することは、社会の知識レベルを高度化するためにも重要である。これはある意味、ソフトウェア技術者達に課せられた社会的責任であると言っても過言ではない。

そのような実践が根付いていない社会は、技術の集積が困難である。そして、長期的には他の知識の集積がし易い社会との競争に負け、淘汰される運命にあると言える。

第 8 節　日米におけるソフトウェア技術者の働き方の違い[31]

米国社会においてソフトウェア技術者は、ジョブ型雇用契約に基づき、その採用と解雇が決定されていることは既に述べた。専門職であるソフトウェア技術者の場合、一般的に企業側からの一方的な解雇通告によって解雇される例は少ない。

多くの場合、ソフトウェア技術者達は、自分が関わったプロジェクトの終了とともに、社内で次のプロジェクトを探す。社内に適切なプロジェクトが見つからなければ、その企業を退職し、他社で企画されている別のプロジェクトに参画するため、退職する例が多い。

この時、ソフトウェア技術者は専門職として専門家団体に所属しているため、その団体に対して提出された企業からの人材募集情報を参考に次の勤務先を決定する例が多い。また、専門家は、専門家達の人的ネットワークを形成していることが多く、その人的ネットワーク内で求人情報を得て、その求人に応募する例も多い。

また、そのような人的ネットワークに参加している特定のメンバーが、主要な役割を担っているプロジェクトの企画がある場合もある。そのようなメンバーは、自分が参加している人的ネットワークを利用して、必要な人材を集める例も多い。

米国の専門家としてのソフトウェア技術者は、そのような専門家団体の斡旋や、自分が関係している専門家グループの人的ネットワークを活用した求人情報の収集が主要な方法となっている。特に、後者の場合、その人的ネットワークを基盤としたメンバー間の絆は強く、時として主従の関係に近いものがある。

つまり、人的ネットワークに参加している技術者の中に、何人かの中心的人物、高い技術的知識や経験を持った技術者がおり、それらの人々がプロジェクトを企画することも少なくない。そして、企業に自分のアイデアを売り込み、そのプロジェクト案が採用になった場合は、自分達の仲間を組織してそのプロジェクトの実施に当たるのである。

このような背景があるため、米国企業のソフトウェア技術者グループは、企業組織と言うよりも、そのような固定した人的グループを中心として構成される場合が多い。そのような人的グループの結束は強く、特に、そのリーダー的存在の技術者に対する他のメンバーの忠誠心は極めて強い。

プロジェクトそのものは、一定期間で終わっても、彼らのグループとしての活動は、長期にわたって継続するからである。メンバー個人にとっても、そのようなグループに所属していれば、良い仕事に就け、自分の技術を磨くことができ、より高い収入を得られる可能性が高い。

そのようなグループを中心とした仕事では、各メンバーは、自分の信念に基づいた仕事の内容についての意見や主張がある。しかし、グループ全体の意見をまとめなければならない場合には、議論は尽くすものの、最終的な決定はリーダー的存在の技術者に一任する。

それは、その人が実質的なリーダーだからであり、グループとしては全員が唯一の方針を決め、それに向かって活動しなければ、プロジェクトは実効しないからである。また、

[31]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.106-111 参照

そのようなリーダー的存在の技術者からの仕事の命令については、ほとんどの場合、無条件にその命令に従う。

一般に企業に対する忠誠心は、日本人従業員は米国人の従業員に比較して極めて高いと言われている。しかし、それは従業員達が所属している企業組織に対する忠誠心に焦点を当てた議論であって、忠誠心そのもののことではない。

米国の技術者、ソフトウェア技術者を含めて、のリーダーに対する忠誠心は、日本人の忠誠心とは比較にならないほど強固であり、命令に対する服従の精神もはるかに強い。このことの背景には、宗教的な違いや、社会的な倫理観の違いが影響している可能性はある。日本人の場合、組織と言う抽象的な概念に対する忠誠心はあっても、具体的な命令に対する服従と言う意味では、表面的には組織の命令に従っているように振舞っても、その活動の徹底ぶりは不十分であったりすることも多く、不完全だと言える。

日本人のソフトウェア技術者は、既に議論したように、ソフトウェア技術者として大学教育を受けているとは限らない。様々な専門を学んだ人材が、特定の企業に就職し、その企業においてソフトウェア関連の業務を専門とする部門に配属されたために、一時的または一定期間従事する仕事してソフトウェアに関する技術的業務に携わっているのである。

企業とそのソフトウェア技術者との関係には、継続性があっても、管理者や技術的リーダーとの関係は、一時的または短期的である。管理者やリーダーは、職場のローテーションでかなり頻繁に仕事を変えてゆくからである。

また、プロジェクトに参加している他のメンバー達との関係も、一般的には短期的と言える。自分自身を含めて、プロジェクトに参加しているメンバーも、ジョブローテーションによって担当業務が変わってゆくのがメンバーシップ型雇用の特徴である。

日本人の技術者は、一時的には技術者的な仕事をしているものの、本質的にはその企業の管理者に育ってゆくべき人材であり、管理者としての態度や考え方を重視されている。特に業績評価において、単なる仕事の成果そのものよりも、人間関係の維持や会社に対する貢献を高く評価するのはそのためである。

日本の組織においては、ソフトウェア技術者個人個人の仕事を厳密に規定しないだけでなく、その責任を明確に規定することもしない。仕事の成果は、個人の仕事の成果ではなく、その技術者が所属しているグループ全体が達成した成果であることが多く、失敗の場合も、技術者個人の失敗ではなく、グループ全体の失敗であり、個人的な責任を問われることは稀である。

このことから、技術者(従業員)のグループに対する依存度は大きく、そのためグループ内における人間関係の維持は、重要な問題であることが一般的である。その最も重要なものが、企業全体の組織である。

そのようなことから、日本人のソフトウェア技術者は、その企業組織のメンバーとして、管理職的な視点からの組織への忠誠心を高く維持している。つまり、最も重要なことは、自分達が所属している社会でもなく、自分達が関わっているソフトウェアでもなく、自分

達がメンバーとして参画している企業組織である。

何が正しく、何が正しくないかは、この企業組織の視点で決定される。時として、日本企業における社内の不正が、なかなか公にならない理由は、そのような日本企業とそこで働く日本人技術者との関係性が原因であろう。

例えば、特定の製品における設計の一文に関する議論で、それが良い設計であるかどうかの議論の場合、日本人ソフトウェア技術者の視点では、それが技術的に妥当な設計かどうかよりも、それが「我々の企業」が開発する製品の設計として有利なものであるかどうかが問題になる。技術者の社会的責任や、技術者から見た設計の妥当性が問題にされるのではなく、企業にとっての収益性や売れるかどうか、会社のイメージに合うかどうかが議論される。

このような日本人技術者のもつ思考の偏りは、時として重要な製品の問題を引き起こすことがある。そのような問題が、表面化した場合における日本人技術者のよくある対応に、問題を隠すと言う傾向がある。

本来、技術者個人とその技術者個人が勤務する企業組織とは別のものであり、技術者個人の価値観と、企業組織の価値観が違うことに不思議はない。にも拘らず、日本人技術者の場合、自分が勤務する企業の存続を第一に考え、場合によっては問題を隠ぺいしたり、問題を小さく見せるような努力をしたりする[xli]。

このような日本人技術者、もっと広く言えば日本人従業員の姿勢は、近年、一般化しつつある監査制度の実施においても重要な問題を引き起こす可能性がある。すなわち、監査の実施において、現場の担当者が組織防衛のために互いに協調して嘘をつくのである。

監査の原則は、監査でヒアリングの対象になった組織のメンバー個人は、「嘘をつかない」ことである。これは、キリスト教に根差した倫理観に基礎を置いたものである。

実際に、米国の組織で監査をすると、組織のメンバー個人は、組織を守ろうとはするが、組織を守るために嘘をつくことはない[xlii]。それは、自分が嘘をついて(宗教的な罪を犯して)まで、組織を守る義務はないと考えるからである。

監査を担当する者が、個別の事実を指摘し、それらの事実と組織が表明している原則とが適合しないことを確認する時、米国企業では、ほとんどの技術者は組織の問題を簡単に認める。そして、その理由に関する自分なりの説明をするのである。

これに対して、日本の技術者や日本企業の従業員は、とのような事実を突き付けられても、企業組織を守るために嘘をつき通す傾向がある[xliii]。これは、そのような行為が組織を防衛するために必要であると認識し、さらに組織を防衛するために自分が嘘をつく(罪を犯す)ことは、他の多くの組織メンバーを救う結果になるため、許されることだとする考え方が根底にあるからである。これでは監査は機能しない。

ガット・ウルグアイ・ラウンドにおける多国間交渉において、貿易の自由化が議論され、その新しい枠組みについての合意が成立して以降、外部監査に基づく認証によって、企業等の組織の信用を評価し、担保する仕組みが次々と制定されている。強制力を伴う認証規

格としては、品質保証システムに関する国際規格としての ISO9000、環境保全のための管理システムに関する国際規格としての ISO14000、情報セキュリティ管理システムに関する国際規格としての ISO27000 などが、その例である。

さらに強制力を伴う認証規格ではないが、ISO/IEC15504 のようなソフトウェア開発組織の評価に関する国際規格や、IEC61508 のような組込みソフトウェアを応用したシステムの安全性確保に関する国際規格なども制定されている。

これらの国際規格は、基本的に対象となっている企業組織が、その製品やサービスを市場に提供する組織として、十分に整備された管理システムを導入しており、十分に信用できるものであることを、監査等によって保証するものである。従って、監査システムが十分に機能しない社会においては、それらの制度自体が意味を持たないことになる。

アジアの国々の中には、その意味で日本と似たような問題を内包している国も少なくなく、今後、大きな問題になる可能性もある。

第7章　技術者と企業の倫理

第１節　資本主義における倫理の問題

21 世紀に入り、日本国内では企業による虚偽報告や隠ぺい、世論操作、情報の改ざん、インサイダー取引、偽装など、多くの事例が問題になっている。これらの問題の根源にあるものは、日本企業における倫理規範の崩壊である。また、企業内で働く専門家達の職業人としての倫理観の欠如の現れであるとも言わざるをえない。

自由主義市場経済においては、企業は市場における熾烈な競争を勝ち抜いてゆかなければ、その存続を維持できない。しかし、競争だからと言って、「勝つために何をしても良い」わけではない。企業は、社会によって決められたルールの範囲の中で、競争しなければならない。

経済がグローバル化すれは、そのルールもグローバル化するので、ルール自体は時代とともに変化するものである。例えば、昭和 40 年代までは、日本国内における習慣として広く認められていた、中元や歳暮の習慣は、現在もその形式は存続しているが、高額の金品の贈り物は贈収賄と判断されるようになってきている。

競争のルールが時代とともに変化するとは言え、自由主義市場経済における競争原理とそれを底辺で支える倫理的価値観は、普遍的なものである。つまり、それら普遍的な価値観が変わっているわけではない。

ただ、何がどこまで許され、何がどこから許されないかの行動規範が、社会の進歩に従って法規制として変わっていっているのである。人類の倫理的価値観が変わっているわけではない。

産業革命以後の自由主義世界は、ほぼ毎年２パーセントの成長を維持してきた。この成長を維持するために機能していた自由主義社会の行動規範の根底にあるのが、「個人による社会への貢献を目的とした金儲けを是とする」倫理観である。

マックス・ウェーバーは、それを「資本主義の精神」と呼び、イギリス系プロテスタントの倫理観との強い関連性を説いた。それは、産業革命に成功して、国力を大きく発展させたイギリス(大英帝国)やアメリカ合衆国と、それらの社会に共通的に見られる傾向を分析し、議論したものである。

元来、「倫理観」（ethics）なる概念そのものは、ギリシャ的な概念である。人間の普遍的な行動規範である「規律」（discipline 社会的約束事）は、文化による影響をあまり受けない。

しかし、それを抽象化した倫理観は、ソクラテス、プラトン、アリストテレス、ロック、ベンサム、カントらによって意識的に議論され、個別的な規律から、それらを生み出す根源となる思想（または哲学）にまで昇華した。この思想は、個別的な規律や法律とは異なり、宗教や基本的な価値観の影響を大きく受ける。

ギリシャ哲学における倫理は、ソクラテスの議論に強く現れているように、「ギリシャ

の市民として、どう生き、どう行動すべきか」を決めるための指針を示す思想である。ソクラテス達にとって、ギリシャ市民とは、ギリシャの都市国家の政治に参画する者達であり、その都市国家の戦士たちである。

彼らは、平時においては政治を行い、戦時においては兵士として戦う。どのような状況においても、周囲の人々(市民)からの尊敬を受けられなければ、自分の能力を発揮することはできない。そのためには、周囲の人々から尊敬されるように振舞い、生きなければならないのである。

戦場において、平等な市民の一人として、他の兵士に対して、例えその兵士達には死を意味する命令であっても、国家を守るためにはその命令を下す必要がある。命令を下された兵士が、その命令に従うのも、その命令に従わなければ国家の存続はなく、自分の家族全員が敵国の奴隷にならざるを得ないからである。

下された命令が、自分達の国を救うために必要であることを理解し、その命令を下した市民を尊敬できる人であると考えるから、命令に従う。そのような、危機的な状況において、何が最善かを判断し、その策を実行できる人にこそ、徳がある。徳を為すこと、為せることが、倫理的である。

17 世紀のイギリスの思想家、ロックの倫理も、ソクラテスの倫理の延長線上にあるが、ロックにとっては、社会を円滑に機能させるために、「何が重要か」が問題となっていた。それは、当時の社会において万能の権力を握っていた国王でさえも、侵すことができない市民の基本的な権利を規定することであった。

それらは、国家でさえ侵害できない個人の生命と財産、さらに信教の自由であった。18 世紀に入って、イギリスのベンサムは、功利主義の原則を唱え、「最大多数の最大幸福」と言う原理に基づき、政治が行われる(法律で個人行動を制約する)べきであるとした。

これに対して、同じ 18 世紀のドイツの哲学者カントは、善を為すとは、「その人がその心の奥底からこれが正しいと信じることを実践することである」とした。さらに、そのような本当に良い行為は、人によって変わらない普遍的なものであるとした。

カントは、他人から見て同一の行為であったとしても、そのきっかけが純粋なものであるか、不純なものであるかによって、善かどうかが決まると言う立場を取った。すなわち、偽善と善を区別したのである。また、カントは、「本当の善は、時と場所によって異なることはない」とした。つまり、本当の善は、条件や状況によって変わるものではなく、普遍的に正しいものであるとした。

資本主義社会において、上述した企業倫理や職業倫理は、これらの先人たちによる倫理に関する考察の文脈の上にあり、背景となる哲学の影響を色濃く受けていると言わざるをえない。例えば、よく知られているように「自らが食べ、着、住まうために必要な金銭を稼ぐことは許されても、それ以上の金銭を稼ぎ、蓄積することは悪」とする古典的カトリック教徒の倫理観と、「金銭をかせぎ、蓄積することは善」とする自由主義市場経済や資本主義の倫理観は両立しない。

この「可能な限りの金銭を、個人の全精力を注いで稼ぐことを善とする」プロテスタント的な倫理観でも、そのことが「社会全体に対して貢献するものでなければ悪とする」ことが前提となっている。つまり、個人であれ企業組織であれ、社会に対する貢献を目的とする限りにおいてのみ自由な競争が許されている。

ここでは、経済活動を通じての社会貢献が、重要な価値基準となっている。従って、結果として社会に悪い影響を与えるような行為や、単に私腹を肥やすためだけの行為は、そのような倫理観から見れば、いかに競争状態であっても許されない。

第 2 節　現代の資本主義社会における倫理の必要性

現代の資本主義社会において、企業や専門家が、その存在環境である社会から暗黙のうちに負託されている権限と責任は、社会の構成員である一般の個人とは比較にならないほど大きい。それは、企業組織やそこに所属する専門家が、一般の個人には不可能なほど複雑で高度かつ巨大な成果を生み出し、社会全体がそれに依存せざるをえないからである。

例えば、金融機関としての銀行や、その銀行で働く融資担当の行員は、一般市民から見れば巨大な資金を動かし、特定企業の活動を支援したり、逆に特定企業の存続を不可能にさせたりする権力を握っている。そのような行員が私利私欲から、特定企業への財政支援を決定し、実施することは社会を崩壊させる可能性がある。

また、建築設計事務所の専門家(一級建築士)による耐震偽装問題は、国家資格を与えられた専門家が高度な専門知識を悪用し、本来は適法でない建築物の構造設計を、あたかも適法であるかのように見せかけ（詐欺行為)、経済的な利益をえたものである。その建築物を建築した建設会社も、そのような不十分な耐震構造の建築物を建設することで、建築コストを著しく低減させて、利益を得ていた。

この詐欺行為自体は、倫理上の問題ではなく、建築家としての規律(法律)遵守の問題である。この問題に関して、その耐震偽装設計をしっかりとレビューせずに、形式的に承認を与えた公的機関の専門家も、法的にはどうあれ、倫理的責任を問われるべきであろう。これは、我が国において、専門家による技術的レビューが実質的に機能しないことを示す事例である。

本来、耐震偽装設計のような事象は、成果である設計書や構造計算書を精査すれば、検出可能なものである。実際に、問題の設計を検証した経験のある他の専門家は、以前から耐震偽装の可能性を指摘していた。

ではなぜ、検出可能な偽装を検査機関は見逃したのか。それは、担当者が専門家として、社会から負託された責任を全うしようと、真剣に対応しなかったことが第一の原因である。すなわち、専門家としての知識はあっても、それを活用して問題を発見しようと努力しなかった結果である。

どのような専門分野（discipline）においても、ある専門家の実施した作業の成果を、他の専門家（ピア）が専門家としてレビューし、責任をもってその成果の妥当性を保証することが求められている。もちろん、それでも見逃される問題は残っているはずである。

しかし、これまで報告されている事件、事故の原因は、そのような人知の及ばない部分が原因で発生しているのではなく、十分に予防可能な問題を見逃したことが原因である。そのような意味で、企業組織も、そこで働く専門家にも、最善を尽くすことが求められている。

組込みソフトウェア開発において、良い設計を考えることは、設計者として当然のこ

とである。そのことと同様に、他の設計者が「良い設計」と考えたものを、精査し、その設計に隠れた問題点を洗い出し、設計者と議論し、さらに最初の設計案より良い設計にするべく最善の努力をすることも、専門家としての重要な責務である。この第二の視点、すなわち他人の仕事の成果が妥当なものであるかどうかを客観的に評価する能力は、これからの社会で活動する技術者として極めて重要な能力である。

現代社会の専門家達は、自分たちの失敗が、自分たちに権限を付託した一般の個人や消費者に、どのような悪影響を与える可能性があるかを十分に認識し、失敗を未然に防止できるよう、最善を尽くすことが必要である。ここに、職業人としての倫理観が問われる原因がある。

規律を守ること以上に、社会に対して負っている責任を認識し、自分たちがどう行動するかを考え、自分たちの行動を律しなければならない。そして、その結果に責任を持つことが重要になっている。

三菱自動車は同社製トラックの開発において、車輪を車軸に固定するハブの設計に問題があることを知りながら、その設計を改善せずに発売に踏み切った。当該トラックが利用されるようになり、日本のいくつかの場所でトラックの車輪が、トラックの車体から外れる、または外れそうになった問題が報告された。

しかし、同社は当時の管轄官庁であった運輸省に報告をせず、リコールをしなかった。品質保証部が問題の対応に当たってはいたものの、販売した全てのトラックの改修は不可能であったと考えたのである。

そのような状況の中で、横浜市内に住む女性と子供がトラックから外れ、転がってきた巨大な車輪にひかれて死亡した。当初、同社は設計上の欠陥を否定したが、最終的には設計上の欠陥を認めざるを得ない状況に追い込まれた。これは、企業の技術者の倫理観が問われるべき問題であった。

ピアレビューが事故の防止策として機能するためには、上述したような資本主義的倫理観が企業組織において、職業倫理観が専門家個人個人において確立していることが前提となっている。しかしながら、日本の企業を見ても、また技術者を見ても、そのような状況にはない。

技術は日々進歩し、製品は高機能化、複雑化を続けている中で、企業も専門家個人も競争を勝ち抜くことを義務付けられている。しかし、どのような環境にあっても、我々は、専門家として自らの使命を全うしなければならない。レビューは、現代のものづくりの実践において、その最初の第一歩であり、最後の一歩でもある。

第 3 節　米国におけるトヨタ車の急加速事件[xliv]

以下の 2010 年 2 月 8 日付ウォール・ストリート・ジャーナル掲載のジェフ・キングストン氏によるコラム記事は、我々にとって示唆に富む警鐘である。以下にそれを示す。

「臭いものにふた」が、リコール問題に対するトヨタ自動車のアプローチのようだ――当初は利かないブレーキや自らの意思を持つアクセルペダルの問題を否定し、問題を最小限まで矮小化しようとした。豊田章男社長が 5 日の記者会見まで 2 週間にわたり姿を見せないなど、重大な安全問題に対する準備が足りないために世界の顧客からの信頼を失いかねない状況にある。

品質や信頼性の代名詞であるトヨタにとって、ブランドイメージへの打撃は計り知れない。今回の危機管理はこれ以上ないほどお粗末である。国内生産分も含め、リコール拡大は確実だろう。同社に対する訴訟の準備が進められており、多額の費用が予想される。その上、遊休生産施設や空のショールームの負担もある。

トヨタの反応が鈍く的はずれなのは意外ではない。日本では危機管理がひどく遅れているのだ。過去 20 年を振り返っても、日本企業が危機管理に成功した例は思い当たらない。どの問題でもパターンはお決まりで、当初の対応は通常遅く、問題を最小限に見せようとし、製品リコールを先延ばしにし、問題についての対外的なコミュニケーションが不足し、製品から悪影響を受けた消費者へのいたわりや配慮がなさすぎる。火を噴くテレビであれ、汚染粉乳あるいは産地偽装であれ、企業はどのケースでも証拠が積み重なり言い逃れができなくなってようやく公表に踏み切り責任を認めるという形で消費者をごまかしてきた。製造物責任法（ＰＬ法）に基づく訴訟の賠償額がほとんどの場合恐ろしく少ないか存在しない日本では、そうした怠慢によるコストは低い。

日本では生産側の利益が消費者の安全に勝るのが普通だ。日本企業は、事実を隠したりごまかしたりすることがよくあり、広報担当が業務遂行に必要な情報を持っていないことも多い。経営トップに正確な情報を迅速に知らせる体制がないため、正確で十分な対応ができない。そのため、経営陣はメディアからの質問に対処する準備が整っておらず、「協力を渋っている」、「無関心である」という印象を与える。

この危機対応の誤りには文化的要素もある。技能と品質に対する強迫観念のある国で製品の欠陥を告白することには不名誉や当惑が伴うため、公表や責任を負うことに対する基準が高いのだ。トヨタのような社会的地位の高い企業であれば、会社の顔が危機にさらされることになり、失うものも多い。

日本企業では「敬い」の文化のために、目上の社員に質問したり問題を知らせたりすることが難しい。コンセンサスやグループを重視することはチームワークを醸成する上ではプラスだが、決まったことや計画されていることに異論を唱えるのが難しい場合もある。

今回の危機は、トヨタが企業文化を刷新し、品質保証を改善するチャンスだ。そのために、顧客重視姿勢を強める方法もあれば、情報の伝達とフィードバックという双方向の流

れを使う方法もある。独立社外取締役を指名しコーポレートガバナンスを改善する、あるいはリスク管理を事後処理以上のものにする手もある。まだ状況の立て直しには間に合う。ただ、これは、古臭い企業文化による制約を取り払い、リコールとそれ以上のアフターサービスを提供して顧客の歓心を買うことを意味する。しかし、当初の兆候をみると、トヨタは半世紀にわたり世界を席巻してきた機動的な企業ではないようだ。

米国では、ゼネラル・モーターズ（GM）のためになることは国のためになり、国のためになることはGMのためになると言ったものだ。トヨタが米国に多くの工場、従業員、サプライヤー、ディーラーを擁するようになった今、同社が安全問題に対処し軌道に戻ろうとする際に、同社と米国の相互利益が危うくなっていることを思い起こす価値はある。

日本のメディアはこの問題の報道を最小限に抑えようとしている。日本でのトヨタは、米でよりずっとニュースの管理に成功しており、メディアも政府も一段と慎重だ。ただし5日には前原誠司国土交通相が、トヨタが問題のあることを否定しており、顧客の視点が欠けていると苦言を呈した。しかし、米の当局とは違い、同相は安全面の欠陥調査を認めていない。

米でアクセルペダルに関連したリコールが発表されてから2週間たったこの日、豊田社長はようやく記者会見を開いた。会見では、世界中の顧客に迷惑をかけたと謝罪し、窮地脱出を試みた。同社はブレーキの問題について、アンチロック・ブレーキ・システム（ＡＢＳ）に対するドライバーの感覚が車両の動きとずれているためだとした上、問題は2009年型のみだとした。同社は1月、ＡＢＳの反応を速めるためソフトウェアを修正している。

今回の会見では、顧客を安心させ、10日に米で予定されている公聴会の影響を弱めようとしたが、失敗に終わった。明らかに、トヨタは日本での正式なリコールを避けようとしており、イメージダウンもコストも少なくすむ自主回収を認めるよう政府に働きかけている。（ブレーキに）欠陥はなく、問題は単にソフトウェアの不具合だとする主張はむなしく響き、信頼を取り戻し、信用を回復するにはほど遠い。プリウスはトヨタの売り上げをけん引する重要な車種であり、ブレーキシステムその他に関する疑問は後を引く。

当初、安全面の欠陥はメイド・イン・アメリカとされていたが、設計の欠陥が本国を襲い、有名なトヨタのＱＣ（品質管理）サークルに対して新たな疑問を投げかけている。この話が米国発でなければ、日本でのリコールは検討されたかすら疑わしい。しかし今では、品質問題に関する海外報道の拡大に背中を押された国内メディアが、それほど辛らつでないにしても、繰り返し同じ質問をし、同じ問題を取り上げそうだ。

トヨタの信頼回復は、同社にとどまらず、日本にとっても極めて重要性が高い。ここ数年、日本製品が世界や国内の消費者から期待される高い品質基準を満たせないケースが驚くほど多い。これは、漂流し滑りながら衰退する国のバロメーターという見方もできる。

日本には、品質に関して自己満足にひたり、生産性の停滞を放置する余裕はない。人口構成の時限爆弾があることを考えるとなおさらだ。高齢化と人口減少が同時に訪れるため、少ない労働力でより多くを生産する必要がある。拡大する高齢者層、年金、医療ニーズを

支えるには、付加価値や 1 人当たりの生産量を高めなくてはならない。そしてこれは、日本がつまずいたときにいつでも取って代われる態勢にある、韓国のような競合国に後れをとらないことを意味する。トヨタ復活というシナリオは、打ちひしがれた国の心理に大きな意味を持ち、品質に疑いではなく賞賛の目が向けられる生産大国の評判を日本が取り戻す可能性を意味する。そうなれば、トヨタが立ち直り、再び勢いづき、日本が必要とするインスピレーションを与えることにつながるだろう。

また、以下の 2010 年 2 月 10 日付ウォール・ストリート・ジャーナル掲載の記事は、トヨタ車リコール問題の本質を組込みソフトウェアの導入にあると分析している。

『トヨタ自動車が米国でアクセルペダルの不具合でリコールを届け出た問題の背景には、自動車の最重要機能を電子システムに頼る傾向が強まっているという業界の流れがある。トヨタを 8 車種の販売中止に追い込んだ、車が突然加速する問題の原因究明で、従来の管や油圧用作動油に代わって半導体や電動センサを使う、こうした電子システムの複雑性が焦点となっている。2000 年代に、トヨタは新型の電子式アクセルを使い始めた。これまでアクセルペダルは、加速用のエンジンスロットルを開けるケーブルに繋がっていた。だがトヨタが使い始めた全電動型の車両には、運転者のアクセルペダル踏み込みの強度と速さを感知するセンサがついており、エンジンのコンピュータに加速か減速かの指示信号を送る。電子式ペダルは現在普及しており、スリップ防止のためのブレーキ制御を助け、車両のハンドル操作をより正確にするため電子システムを備えた車も多い。ハンドルは物理的にタイヤに繋がっており、油圧システムがブレーキペダルからの力をブレーキに伝えるといった具合にブレーキとハンドルのシステムの多くは依然、機械部品によって制御されている。リコールに関するトヨタの説明会に参加したミシガン大学のジェフリー・ライカー生産工学部教授は「電子式システムのほうがずっと優秀で安全。制御と情報伝達の機能を持つマイクロチップがあり、このチップが一つでも狂うと表示が出て停止する」と語る。トヨタは今回のリコール問題の原因はヒータの結露がアクセルペダルの摩擦を増し、踏み込んだペダルが戻らなくなるケースもあったとしており、電子システムではなく機械的な問題だと主張している。だがトヨタのシステムがなぜアクセルとブレーキが同時に踏まれた際にアクセルが緩んで車両を止める「ブレーキ・オーバーライド・システム」を備えていなかったかを疑問視する専門家もいる。自動車業界調査会社セイフティ・リサーチ・アンド・ストラテジーズのショーン・ケイン社長は、自動車メーカは「設計力が検証力を上回っている状況。車両の電子システムは増えているが、問題発生のペースはそれを上回っている」と指摘する。 』

第 4 節　歴史的に見た労働の意義

1. 類人猿から人類への進化

世界最初の人類であるナンシーは、子供を育てること、自分および子供たちの生命を守ること、食物を手際よく獲得することが最大の関心事であったはずである。それは、ナンシーの大脳が小さかったからだけではなく、人類が単なる新種の類人猿だったに過ぎないからである。極端に言えば、ナンシーには、ものごとを考える力はなかったのである。彼女には、他の類人猿と同様に、与えられた本能のままに生きることしかできなかったはずである。

ナンシーの死後、その何万世代以上もの子孫である人間の中に、言葉を発明した子孫がいた。言葉を使って周囲の人間たちと、情報を交換し、新しい知識を獲得した。例えば、食べられる植物と、食べると死んでしまう植物の判別の仕方など、生きるための知識が交換された。また、どこへ行けば食べられるものが豊富にあるかなどの情報も交換したに違いない。しかし、それでも彼らに抽象的な概念を使って、ものごとを考える力はなかったと言える。

この世代の人類は、働くと言う概念を意味する言葉すら知らなかったであろう。または持っていなかったと言うべきかもしれない。それは、彼らには他の動物と同じく、生きること自体が当初から与えられた目的であり、それ以外の目的はなかったからである。外敵から身を守り、食物を獲得し、寒さから身を守り、子孫を増やすための伴侶を見つけることだけが関心事であったはずである。

2. 文明の発祥と身分制度

さらに世代を経て、古代バビロニアの時代になると、人類は文字を発明し、数(自然数)の概念を獲得する。この段階では、人間は自分の周囲に存在するものに名前を付けることだけでなく、実際には存在を確認できない抽象的な対象にも名前を付けて考えるようになっていた。例えば、ものが一つ存在することを抽象した概念としての「1」に、名前をつけたり、アルファベットの各文字に名前を付けたりした。

死者の霊魂や神や悪魔の概念は、我々が見ているものに付けた名前ではなく、我々の思考が生み出した概念に、名前を付けたものである。このような時代になると、人間は考えることを仕事とする特別な人々の集団を生み出した。数学者や自然科学者である。また、人々を統治することを仕事とする人間(王)や、その仕事を助けることを仕事とする人々(政治家)や、宗教行事を行う司祭・僧侶などの専門家を生み出した。

これらの専門家は、一般の人間と比較すれば、極めて少数であり、当時の社会では特殊な存在であったに違いない。しかしながら、そのような天文学などの自然科学や政治の専門知識に精通した人々は、すでに自ら得た知識を利用して、一般の人々よりもはるかに豊かな生活を営んでいた。それらの人々にとって働くことは、他の一般の人々のように自分の肉体を使って活動する単なる肉体労働ではなかった。

もちろん、その時代においては、大多数の人々は農耕などの肉体を駆使した生産活動に従事していた。また、他の人間たちの集団と戦い、自らが属する集団を守る役割を担う、自分の生命を懸けた仕事に従事する階層(兵士)も出現していた。

一般的に言えば、封建社会の確立である。封建社会では、各階層に属する人々は、それぞれの階層に与えられた役割を担うことが期待され、それを専門の仕事として実践していた。

では、仕事とは何か、労働とは何か。人間が生命を維持し、健康で安全な生活をおくれるように、人間自身が行うべき実践的な活動の総称を仕事・労働と言う。特に、労働の概念は、それらの実践的な活動が社会的な枠組みの中で、組織的に実施され、その結果として、労働を提供した人々に対して、提供された労働の対価として何らかの価値のある物質が提供される場合に当てはまる。

自分の生命を懸けて国のために戦う戦士に対しては、戦いに勝てば、戦利品が分け与えられる。戦利品には、時として奴隷のような人間が与えられることもあった。

3. ギリシャ・ローマ時代

ギリシャ、そしてそれに続く古代ローマ帝国時代を経て、人々の労働に関する概念は大きく変わってゆく。ギリシャの市民は、生産のための、経済的価値を生み出す労働に参加してはいなかった。

市民は、単にものを消費するだけの人々であった。生産活動は主として、奴隷(戦勝国が敗戦国の市民を強制的に連れてきて、役務に従事させていた人々)による労働によって支えられていたのである。

ギリシャ市民の仕事は、平時においては政治を行うことであり、戦時においては敵と戦うことであった。ギリシャの市民は基本的に平等であったため、誰が行政を行うかは、選挙で決められていた。

行政能力の高いものが政治を行うことが正しいと信じられていた。また、戦争で誰が戦略を立て、誰が将軍として軍を指揮するのかも、市民相互の選挙で決められていた。

敵味方が複雑に入り組んで戦っている戦場では、適確かつ断固とした判断に基づいた命令が、勝敗の行方を決することも多い。一部の局面では負けても、戦い全般では勝つことも多い。

将軍となった市民は、部下の兵士となった市民に向かって、負け(死ぬこと)が明白な場合でも、その戦場に出向くことを命令しなければならない。そして、部下となった兵士の市民はそれに従うことが義務であり、その義務を全うすることが美徳であった。

なぜ、平等が原則であるギリシャ市民の世界で、将軍に選ばれた市民の命令に、部下の兵士となった市民が従うのか。それは、戦争に負ければ全ての市民は奴隷にならざるを得ず、勝つためには誰かがその命令に従わなければならず、将軍が最も適切な任務遂行者として自分を選定したからである。

将軍の判断が誤りでなければ、自分が行くのが義務である。これが、ギリシャ市民の倫理観であった。このことは、特にソクラテスの倫理観の基礎となっている。

そのようなギリシャの市民にとっては、平時に倫理観や哲学を学び、人としての徳を磨くことが重要であり、それが主たる仕事であった。また、戦士でもある多くの市民にとっては、身体を鍛え、武術の技を磨くことも、平時の重要な仕事の一つであった。ギリシャ市民にとって、スポーツは余暇のためのものではなく、身体を鍛えると言う意味で、戦争のための準備でもあった。

ギリシャ時代においては、市民だけでなく、奴隷も一部において専門化していた。例えば、数学の計算だけを専門とする奴隷もいたと言われている。

奴隷たちは、いかに専門的な活動に関わっていたとしても、何をするかを自分で決定する自由はなく、主人から与えられた命令に従って、専門的な活動(役務)に従事しなければならない。しかし、広い意味では奴隷も専門的で高度な仕事や労働に従事していた例も存在したのである。

ギリシャ文明の影響を色濃く引き継いだ古代ローマ帝国の市民も、キリシャ市民と似たライフスタイルを踏襲したが、ローマ帝国が巨大化するにしたがって、ギリシャにはなかった階層化や、皇帝の世襲制が導入され、軍人も専門化する。特に、カルタゴとの長期にわたる戦争の後、ローマ軍は大きく変容したと言われている。

同時代に繁栄したカルタゴが地中海貿易を中核とした商業国家であったことから、それを真似てローマの商業が発展した。商品の輸出入を商売とする人々が生まれ、富をなすに至った。

首都ローマには、世界中から商品が集まり、世界中に分散していった。また、ローマ人は戦い、広大な地中海世界やヨーロッパ大陸を支配するための手段として、基幹道路網の計画・整備の専門家(現代の都市工学の技術者)を多く育て、各地に供給した。

以上のことを総合すると、ギリシャに比較して、古代ローマ帝国の社会は、はるかに複雑化しており、それを矛盾なく統治・統合するための仕事は、階層化され、専門化されていたことが予想される。その意味で、商人や物づくりを生業とした人々も含み、古代ローマ帝国市民の多くにとっては、専門知識を磨き、それを活かして軍事・政治活動や経済活動に従事することが仕事であった。

歴史を振り返れば、古代ローマ帝国の市民がそのような実態を理解し、適切に対応できなかったことは明らかである。古代ローマ帝国の絶頂期を過ぎると、どんなに立派な皇帝が統治をしても、努力を嫌うようになったローマ市民から自発的に兵士になる者は減り、未開の地で幹線道路や上水道の建設に赴くものもいなくなり、危険なシルクロード交易に出るものもいなくなった。

歴史家のトインビーは、文明のミメシスという概念を用いて、古代ローマ帝国の滅亡を説明しようと試みている。ミメシスとは、文明を支えている人々の「生きる力」であり、その力が失われることが原因で、文明は衰退すると説く。

文明は発展期において、大きなミメシスをもち、絶頂期においてミメシスも頂点に達する。そして、各文明のミメシスの積分値は一定である。

ミメシスは、別の解釈をすれば、文明の発展を担っている市民の勤労意欲、活力、職業倫理観の総和であると言える。個々人の力には、どんなに立派な指導者であっても限界があるが、市民全体の力を統合できれば、文明は一定のベクトルをもって大きく発展する。しかし、それが無限に続くことはない。市民の勤労意欲、活力、職業倫理観の減退・停滞とともに、大文明は必ず滅亡する。

古代ローマ人の中には、工芸、文芸、歴史学、天文学(暦)、航海術、土木・建築、政治学、宗教において功績を遺した専門家もいた。歴史的には評価されてはいないが、カトリック教会が誕生したのが古代ローマ帝国の絶頂期をすぎたころであり、帝国とは目的は違っていても、その文化的遺産、そして政治的遺産が引き継がれた。

このことが、キリスト教を発展の軌道に乗せた。後に、キリスト教に改宗した皇帝達にとっては、教会と帝国の区別はほとんどなかったのではないか。これが逆に、古代ローマ帝国の滅亡につながったと言える。

4. 中世ヨーロッパの時代

西ローマ帝国を滅亡に追い込んだゲルマン人は、フランク王国を建設するが、当時のゲルマン人には古代ローマ人のような政治的能力が不足していた。フランク王国の王も、キリスト教に帰依していたため、古代ローマ帝国後の西ヨーロッパ世界は、実質的にキリスト教会が支配することになる。国王といえども、ローマ教会の承諾なしに即位はできず、教会によって戴冠式を実施することが必要となった。中世ヨーロッパ時代の成立である。

この中世ヨーロッパにおいて、知的な活動のほとんどは、ローマ教会の管理下にあった修道院で行われた。後に、ヨーロッパに教会とは独立した大学が誕生するまで、教会の知的独占が続くのである。

では、現代社会で言う一般市民は、何を仕事としていたのか。古代ローマ帝国と同様に、ローマ市民と似た環境にあった貴族(騎士)たちは、戦うことが仕事であった。古代ローマ帝国によって作られた各地の都市では、商業や鍛冶屋・靴屋・服屋などの手工業を仕事とするものも出現していた。しかし、ほとんどの人々は貴族が所有する土地を借りて耕作をする農民であった。

教会は、農民や都市の手工業の従事者、商業の従事者に対して、勤勉に働くことは教えたが、物の所有や蓄財を教えなかった。一生懸命に働くのは神のためであり、自らが働いて得たものは、自らが生きるための分を残し、全てを教会へ寄進することを説いた。

貴族は、農民に税を課し、その一部を教会へ寄進した。驚くべきことに、ヨーロッパの人々は、千年以上にわたり、蓄財(資本蓄積)機構が貴族と教会にしかない社会を維持したのである。

個人による財産の保有が認められない社会において、人々が積極的に働く意味はあまり

ない。毎日、必要なだけの労働をし、あとは家族との生活を楽しむことが重要になる。しかし、少数ではあるが、僧侶(神父)や貴族は違っていた。

これらの特権的な人々は、自らが天から与えられた才能を発揮し、より良い成果を生み出すことによって、より良い生活が保証されたからである。しかし、大多数の農民にとっては、労働は「やらなくてはならない役務」であり、仕方なしにやっている活動であった。

教会であれば、ローマ教会組織の巨大なピラミッドの頂点へ向かって、一歩ずつ階段を上ってゆくことができる。ピラミッドの頂点に近いほど、多くの民衆からの尊敬を受け、美しい衣装をまとい、数多くの側近に囲まれる身分になる。油断をすると、それまでの身分をはく奪され、自給自足に近い修道院の生活が待っている。

貴族であっても、一地方の領主に始まって、国王に至るまでの階段を一歩ずつ登ることができる。失敗すれば、家族全員の命を奪われるか、幸運に恵まれても森の中に逃げ込み、自給自足の生活に耐えなければならない。

中世ヨーロッパの特権階級にとっては、勤勉に働くことは、まず自分や周囲の関係者の身分の保全を保証するための必要条件であった。さらに、天から才能を与えられた特権階級の人々にとっては、自分の才能を試し、自己実現を達成する手段であった。

十字軍に参画した騎士団の貴族たちは、神への忠誠心を表現するとともに、自らの戦闘能力を誇示するためにキリスト誕生の地へ向かったのであろう。それは、自分たちの使命を全うすると言うよりも、自分たちの才能を顕示するためであったのであろう。

ローマ教会の絶対的権力に陰りが見えてきたイタリアでは、メディチ家のような領主が、その地域で巨大な富と絶対的な権力を持つようになる。それは、ローマ教会さえもしのぐほどだったようである。

自らの権力と神への忠誠心の証として、そのような領主たちは、立派な教会の建立に力を注いだ。教会の建設には、建築家、土木工学の専門家だけでなく、画家、オルガン作家など多種類の専門家を雇い、それらの能力を統合しなければならない。

そのような時代には、工房で必要な知識を身に着けた、天賦の才能に恵まれた人材が必要になる。ビンチのレオナルド(レオナルドダビンチ)やミケランジェロのような天才が排出される背景が整ったのである。

彼らは、天賦の才能には恵まれていたが、決して特権階級の家系に生まれたわけではない。工房で学び、技法を自ら研究し、それまでに人々が見たことのない絵画や彫刻を遺したのである。芸術家だけでなく、ガリレオのような科学者も、領主の庇護のもとで研究を進めていた。

レオナルドもミケランジェロも、そしてガリレオも、自らの才能をパトロンに売り込み、雇われなければならなかった。レオナルドがあるパトロンに送った経歴書には、様々な兵器の開発構想が書かれていて、今日、彼を有名にしている絵画についてはほとんど書かれていなかった。

それが、当時の社会における一般的なニーズであったと思われる。彼らにとって、働く

とは正に、自分の才能を世に示し、自分の作品を後世に残すための、自己実現の手段であったと言える。それは、天が彼らに与えた類稀な才能を活かし、この世界に彼らが生きた足跡を残すことであった。

イタリアで生まれたこのパトロン制度は、東ローマ帝国の首都、コンスタンチノープルがイスラム軍によって攻撃を受けたため、当時の最先端技術を持った織物職人、革職人、ガラス職人、宝石加工職人、科学者、数学者などが、フィレンツェを中心としたイタリアの港町に移住したことに始まるとされている。そのような渡来人を受け入れ、庇護したのがパトロンとなった領主たちであった。

このパトロン制度によって、従来は僧侶と特権階級の出身者にしか与えられていなかった、天賦の才能を活かして働くと言う選択が、一般市民(ほとんどは農民)の出身者にも可能になった。また、社会に受け入れられる才能も、科学技術、美術、音楽、演劇、工芸など、多分野に拡大してゆく。

そのような背景から、これらの分野の人材を育成する制度も、少しずつ形成されてゆく。ルネッサンス期イタリアに始まる工房制度や、中世ドイツ社会におけるマイスター(徒弟)制度などがその典型である。ただし、哲学、法学、医学の専門家については、この時期に全く異なる人材育成制度が確立する。

ヨーロッパの大学の起源は、中世ヨーロッパに始まる。大学は、領主や国王の行政制度から独立した機関であり、教会からも独立した機関として誕生した。大学は、高度な知識を教え、生み出す組織である。

誰を入学させるかは、大学自身が決めることができる。また、誰に学位を与えるかも大学に決定権がある。誕生間際の大学は、それまでの修道院に似たものであったかもしれない。ただし、修道院における教育と研究の中心が、神学であったのに対して、大学の教育と研究の中心は、哲学、法学、そして医学であった。

中世の大学の門戸が、どこまで一般の人々に開かれていたかは分からない。しかし、大学で学ぶ者の多くが貴族などの裕福な階級の出身者であったことは間違いない。特に、法学においてはその傾向が強かったと言える。

自然科学や数学は、哲学の一部とみなされていた。哲学の分野では、貴族出身者だけでなく、修道院の僧侶も活躍していた。それだけ、修道院における人材育成は進んでおり、また知識の水準も高かったと言える。大学を卒業し、学位を授与された者の中には、教育者として大学に残るものも多かった。

人間機械論を著したため、ローマ教会によって弾圧され、火あぶりの刑に処せられたフランスのラ・メトリは、僧侶ではない専門の医者である。人間が精巧に造られた機械であるとすれば、人間は人間を作り出せる。

しかし、人間を創ったのは全知全能の神であり、神以外には人間は創れないとする教会の説に矛盾する。このことから、ローマ教会はラ・メトリとその著書を放置することが、脅威になると考えたのである。

ガリレオが、動いているのは天空であるとするアリストテレスの天文学の矛盾を発見し、大地である地球が回っているという仮説を提示したとき、アリストテレスの天文学を教義としていたローマ教会が、ガリレオを弾圧したのも、ラ・メトリを火あぶりにした理由と同じである。デカルトも、ラ・メトリと同様に心臓がポンプであり、大脳がコンピュータであるとの仮説をもち、人間は精巧な機械と考えていた。

中世大学の教員や、工房で働く職人達、特にマイスターのような親方たちにとって、働くとはどのようなことであっただろうか。それは、少し前の時代のレオナルドやミケランジェロと大差がなかったであろう。

成功したかどうかは別として、皆、後世に残る作品、後世に残る著作を生み出すため、時としては自らの生命を懸けていた。やはり、自己実現のために仕事をしていたとしか言いようがない。南ドイツのニュルンベルグに育った芸術家のデューラーは、数多くの精細な木版画を残したが、彼は、芸術家でもあったが、マイスターのようでもあった。

5. 近代ヨーロッパと産業革命の時代

長い中世の時代が終わりに向かっていた頃、大きな変化がヨーロッパ世界に起きた。宗教改革である。宗教改革、特にドイツにおける宗教改革の背景には、16 世紀中ごろのグーテンベルグによる活版印刷機の開発があった。

中世の修道院では、全ての僧侶たちにとって、ラテン語を学ぶことと、ラテン語で書かれた聖書を書き写すことが、最も重要な仕事であった。新約聖書と旧約聖書を完全に書き写すためには、5 年以上の月日がかかったに違いない。文字を書くための羊皮紙の製造から考えると、グーテンベルグの印刷機は、その仕事を不要なものにした。これは、ローマ教会においても一大技術革新であったはずである。

一方、イギリスやドイツでは、ラテン語で書かれた聖書を、現地語に翻訳する試みがなされていた。それは、神父として、聖書の内容をラテン語で読み、それを現地語で解説する作業を省略し、生の聖書の言葉を全て、最初から現地語で説いた方が良いとする思想に基づいていた。現実に、ラテン語を理解できる人々は、僧侶と貴族、大学教育を受けた一部の人々に過ぎなかったからである。しかし、現地語聖書と言えども、書き写されていた。

ドイツのルターは、現地語訳された聖書を印刷するという、宗教的な技術革新を起こそうとした唯一の宗教家であった。彼の努力と執念によって、一般の市民・農民の手にも、それでも高価ではあったろうが、聖書は入手可能なものとなった。ルターは、聖書の教えこそが真実であり、ローマ教会の教義そのものは、キリストの直接の教えではないとする立場をとった。ルターは、聖書の記述をあるがままに解釈しようとしたのである。

(ローマ)教会の神父は、「人はパンのために働くものではなく、働くためにパンを求める」と言う意味のラテン語の文を引用し、人々に「神のために真面目に働きなさい。働けば何がしかの報酬が得られるはずです。あなたは、その報酬の一部を、自らの生活のために取ることができます。そして、残りを神への感謝の徴として、教会へ寄進しなさい。決して

欲を張ってはいけません。」と説くことが多かった。これでは、人々は経済的な余剰を生み出そうとしなくなる。つまり、働くことのインセンティブを感じることはなかった。

ルターの説は全く違っていた。「神は皆さん一人一人に、別々の才能と使命を与えています。皆さんの仕事は、牧師と同じ『天職』です。一生懸命働くことが、神への感謝の徴となるのです。その結果、あなたたちが得る名声や報酬は、天からの授かりものでもあります。しかし、強欲は罪です。」ルターの用いた天職というドイツ語は、英語の profession に相当する語である。ルターは、当時のドイツ人に身近な天職という概念を用いて、一生懸命働くことを説いたのである。これは、広い意味での自己実現を説いていると考えられる。

基本的な思想は違ったが、イギリスのプロテスタントの一部にも、「働くことが、祈ることと同様に、神への奉仕につながり、救いとなる」とした人々がいた。その中でも極端な思想を持った集団があった。カルバン派に属する清教徒、ピューリタンである。彼らは、イギリスにおいて清教徒革命を起こし、独善的で理想主義的な政治を行おうとしたが、成功しなかった。その過程で、多くの国民や貴族を殺したため、後に国内では、逆に迫害や差別の対象とされ、後に新天地をもとめて、アメリカ大陸へ移住する。ピルグリムファーザーズによるアメリカ合衆国の建設である。

この資本主義の誕生期に、イギリスでは労働がものの価値の源泉であるとする思想が生まれた。その最初は、ロックである。ロックは、国王と言えども、市民の生命や財産を犯すことはできないとする思想を確立した。その過程で、ロックは人間の社会活動によって生み出される価値の源泉は、その人が所有する財産(資本)ではなく、それを活用して生産活動にたずさわった人々の労働であると主張した。土地は人々が耕さなければ、何も生み出さない。人々が労働を投入し，耕し、作物を育てることで、富を生み出すのである。

このロックの思想を継承し、理論化したのがアダム・スミスである。彼は、その著書『諸国民の富』において、労働価値説を提唱し、その概念に基づいて分業の合理性を説明した。この理論が、産業革命中のイギリス社会に大きな影響を与えた。ところで、この労働価値説の誕生には、コンピュータの父とも言える数学者のチャールズ・バッベージが大きく関わっている。バッベージは、数学を使って、労働の成果の量と、その成果を生み出すために投入された労働の量(労働時間)から、単位労働量当たりの生産量を定義し、議論している。これは、今日の生産性の議論の根源でもある。

アダム・スミスは、労働の成果である商品の価値は、その商品の生産に投入された労働の総量によって決定されるとした。すなわち、商品の価格は、その商品の生産に投入された労働の量によって自然に決まるはずであるという理論である。

また、今日的な言葉を用いれば、同じ商品を生産するために投入する労働の総量を低下させる新しい生産手段の採用は、商品の価格を低減させるが、その生産手段を採用した資本(家)を、市場での競争で優位にさせるため、その資本家における資本の蓄積を助けるとした。つまり、20 世紀のシュンペーターによるイノベーションである。

アダム・スミスは、産業革命を経験していたが、産業革命で大きく発展した資本主義社

会の変化は経験していない。資本主義経済においては、資本家は生産手段である生産設備に投資をし、生産に従事する労働者を労働市場から調達して生産活動を行う。生産活動が軌道に乗り始めると、生産施設は利潤を生みだすが、生産施設は時間とともに老朽化する。

老朽化した生産設備は、徐々に生産性を低下させ、利潤を生みださなくなる。このような生産設備のライフサイクルの変化によって、景気循環が発生する。資本家に渡る利潤が少なくなれば、資本家は生産規模を縮小し、労働者の雇用も減少させる。または、給与を減らす。

収入が少なくなった労働者は、消費にまわす手元の資金が少なくなるため、消費を控えるようになる。このような連鎖が、市場における需要を減退させるため、ますます供給過剰状態を作り出し、資本家はさらに生産を減退させる。景気循環である。

6. 帝国主義の時代

産業革命によって、大きく発展したイギリスを始めとした国々の経済は、このようなメカニズムによって発生する景気循環に悩まされることになる。特に、経済的に大きく発展した中産階級(労働者階級)への経済的打撃は大きかった。

定期的にやってくる不景気によって、失業する労働者は、収入を確保することができなくなり、生活に困窮することになる。この状況を目の当たりにしたドイツの経済学者マルクスは、アダム・スミスの労働価値説の不備を改善し、新しい労働価値説に基づいた『資本論』を執筆した。

マルクスは、産業革命時代のアダム・スミスほど、楽観的ではなく、景気循環による労働力のひっ迫と過剰状態を、理論的に説明しようとした。アダム・スミスの労働価値説では、チャールズ・バッベージの労働価値説も、ジョン・ロックの思想も同じであるが、労働と労働市場で売買される労働力を明確に区別していなかった。

マルクスは、労働価値説の詳細な検討から、市場で売買される労働力を一般的な労働概念とは分離して議論する方法を採用した。すなわち、資本家は労働力を労働者から買い、労働者は市場で労働力を売る。労働者には、労働力を売る自由もあるが、資本家には労働者の労働を買う自由がある。この 2 者間の関係も労働市場における需給関係によって価格が決定されるとの考え方である。

このようにして、労働力を資本家に売った労働者は、生産施設を利用した労働に従事し、商品の生産にたずさわるのである。生産された商品は資本家が所有し、市場において販売される。販売されているのは、労働者(達)の労働の成果としての商品である。

生産したのは労働者であるが、労働力を資本家に売っている労働者には、労働の成果である商品に対する所有権はない。この商品の市場における価値が、労働者(達)が資本家に売った労働力の総和の価値よりも著しく大きければ、資本家は労働者(達)による労働の成果を搾取していることになる。これが、資本主義の根源的問題であるとするのが、マルクスの立場であった。

ところで、興味深いことに、産業革命を境にして、その後、経済的に大きく発展した国々は、日本を除いて、全てプロテスタントの国々である。このことは、マックス・ヴェーバーが今世紀初頭に、彼の著書で指摘した事実である。ヴェーバーは、明らかに資本主義における経済発展とプロテスタンティズムを基礎とした労働倫理との間に、論理的な関係があることを確信していた。

その真偽は別として、ヴェーバーが主張したように、「建国以来、アメリカの人々は懸命に働き、得た富を再び社会へ還元すべきであると確信し、1 秒たりとも怠ることのないように働こうとする」傾向が顕著であったことは現実であった。そしてアメリカ合衆国は世界一の経済大国に成長した。

第 5 節　現代社会における労働の意義

ヴェーバーは、仕事に対する倫理観に焦点を当て、そこにルター的天職観との類似点や、イギリスにおける清教徒の「最後の審判における神の救いを信じて生きる」倫理観との共通点を見出していた。しかし、米国人でもある心理学者のマズローは、全く別の視点から、アメリカ人の仕事や労働への情熱を、心理学的な視点から説明しようとした。それが、マズローの欲求の段階説である。

コメント [U1]:

コメント [U2]:

マズローは、人間の欲求を動物的・生物的欲求の保証段階、動物的・生物的欲求の満足段階、人間の社会的欲求の保証段階、社会的欲求の満足段階、そして自己の存在目的を実現しようと欲求する段階に分割した。最初の 2 つは、全ての動物に見られるものである。

最初は、生命の保全に関する欲求である。第二は、食欲を満たす、安心して暮らせる環境を確保するなど、動物としての安定な状態の保全に関する欲求である。これは、ロックが基本的人権として最低限、守られるべきとしたものである。

マズローの欲求の段階において、第三以上のものは、人間に特有とも言える欲求である。第三は、特定の社会の構成員として認知されることを求める欲求である。第四は、特定の社会の中で、他の構成員から尊敬されたい、愛されたいとする欲求である。最後の第五は、それまでのような相対的な欲求ではなく、自分自身が自分自身らしくありたいとする欲求で、絶対的な価値に基づく。

人間は、その人格の成長とともにその欲求の階段を一歩一歩登ってゆくとするのが、マズローの仮説である。赤子のとき、人は母親にその生死の全てを依存する。母親が、見えなくなると赤子が泣くのは、そのためである。また、お腹が空くと赤子は泣くが、授乳が終わると静かになる。これは、満腹感によって満たされているからである。貧しい人々の多い国では、赤子はいつも泣いているか、死んでいるかである。

人間は、ある程度成長すると、共通点の多い者同士が集団を形成し、そのような集団に帰属することを望むようになる。それは、子供の社会だけでなく、どの世代にも共通して見られる現象である。同じ大学に進学した者同士、同じ企業に就職した者同士などである。同じ年齢の子供を持つ親同士のように、間接的な関係で結ばれた集団も存在する。

人間は、そのような集団の一員として、認知されることを望む。それは、特定の集団に参画したいと欲することも含まれる。就職試験は、そのための門である。日本社会において、内定を得るとは、学生が希望する企業の従業員集団組織への参画(メンバーシップ)が認められた証と考えられる。

ある集団に属している人間は、その集団内または、より広い社会の中で自分が認められたいと感じるようになる。企業内で、昇進を争うのはそのためである。早めの昇進は、現実には運の方が強く影響するが、少なくとも本人にとっては、他者よりも自分の能力が高いことの証明と認識でき、より高い満足感を得ることができる。

また、同じ大学の同期卒業生の中で、最も高い年収を得た者は、それが自分に対する社会的評価であるとして、それを誇りに思う。逆に、そうでなかった者たちは、社会の評価

が妥当でないと感じることもある。

しかし、毎日の仕事に忙殺され、短時間の睡眠、長時間の通勤、上司からの叱咤、部下からの苦情に悩まされ続け、これが自分の仕事の成果だと主張できるもののない管理職の仕事は、賽の河原で石を積むがごとくである。名刺の肩書は立派であり、大きな家に住み、有名ブランドのスーツで身を固めていても、真の満足感は得られない。

周囲の者から見れば、文句の言いようのない人生のようであっても、本人にとっては、入社直後の数年間の充実した毎日の方が、貧しかったとは言え、はるかに有意義な生活であったと思う人もいる。それは、管理職としての仕事が、その人の天職ではないからである。

ドイツの哲学者ヘーゲルは、弁証法による労働の哲学的解明に挑戦した。ヘーゲルは、その著書である『精神現象学』の中で、「主人と奴隷」をテーマとした逸話を紹介し、労働の哲学的な意味を説明している。主人は、奴隷を所有し、奴隷が仕事をするための道具や材料などすべてを所有している。他方、奴隷は何をするかの自由さえ所有していない。

主人は、奴隷に対してあるものを作ることを指示する。奴隷はその指示に従ってものを作るため、どの材料を使い、との道具を利用すれば主人が喜ぶものが作れるかを考える。奴隷は、その目的を達成するため、試行錯誤を繰り返すが、最終的に主人が欲していたものを作り出すことができる。

この逸話に出てくる奴隷には、全く自由がないが、奴隷は、主人が満足するものを作り出そうと努力する過程で、その全人格を懸けて、ものづくりに没頭する。その実践過程こそが、仕事であり労働である。この労働を通して、奴隷はものを作る喜びを感じるのである。

つまり、労働は人間が人間になるための重要な過程であると言える。これに対して、奴隷の主人は、全てのものを所有しているにもかかわらず、できることは奴隷が作ったものを消費するだけである。そこから、人間的な歓びは得られない。

この逸話をマルクスの労働価値説に当てはめて考えると、奴隷は主人に対して労働力を提供している。主人は奴隷が生きるために必要なもの(衣食住)を提供し、生産手段を提供する。奴隷は、労働力を提供して生産活動を行うが、その労働から造り出される成果の所有権はない。その所有権は主人に帰属し、主人はその成果を消費する。

しかし、奴隷は自分の労働力を提供して実施した労働によって、主人が味わうことができなかった歓びを感じることができる。これが、労働のパラドックスである。マルクスは、資本家による労働の搾取にのみ焦点を当てて議論した。しかし、ヘーゲルは、実践過程としての労働に焦点を当てて議論している。

ある日突然、有名企業を退職し、リスクの大きなベンチャー事業に人生を懸ける人がいる。それは、自分の天職を全うしたいと願うからであろう。そのような人は、企業に残り、天職を全うできると考えれば、企業人として働き続ける。また、他社から、好条件を提示され、移籍を勧められても、それまでの企業に残るであろう。お金では、仕事は買えない。

自分を活かし、育てるのは仕事である。これが、ヘーゲルが「主人と奴隷」の逸話で表現した問題である。特に、知的問題解決の本質が強くなればなるほど、労働者は仕事の経験から学び、育って行く。良い仕事がなければ、「主人と奴隷」の主人のように、人は育たない。

リチャード・フロリダは、その著書の中で、現在の米国社会においては、30 パーセントを超える人々が、『創造的階層』に属すると述べた[19]。フロリダは、現代の米国社会では、従来のような資産による上流、中流、下流と言う社会階層の分類は、意味を失っていると説く。現代社会では、社会階層は創造的階層と、非創造的階層の 2 極に分化し、必ずしも所得では特徴づけられない。従来型社会階層は、産業革命後の産業時代の名残である。

創造的階層とは、人間の持つ創造性を活用して仕事をする人々が構成する社会階層である。そのような人々には、医者、弁護士、教員、専門技術者、コンサルタントのような高度専門教育・訓練を受けた人々だけでなく、美術家、音楽家、作家、役者、芸人、キャスター、美容師、料理人など、長期の訓練と修行を重ねた人々と、その養成課程にある人々が含まれる。

これらの人々の仕事に共通することは、顧客の問題解決が仕事であることである。顧客は、特定の問題を持っている。それを解決するために、これらの人々を訪れ、その問題の解決を依頼するのである。

フロリダによれば、これらの人々が構成する社会構造には、共通する特徴がある。異分野の人々が特定の地域に集中し、互いに影響しあう。古典的で、制度化され、形式化された固定的な社会組織よりも、斬新で、自由で、決まりきったパターンや決まりごとの少ない柔軟な社会組織が好まれる。

また、地域社会の中心に、以前から、ある程度の規模の大学が存在している。そのような例に、シリコンバレー近辺のバークレー、パロアルト、サンタクララや、テキサス州オースチン、ノースカロライナ州ラーレー、ダーラム、チャペルヒルなどがある。

創造的階層に属する人々は、自分自身に意欲的に投資をする。時間があれば、新しいことを学ぶことに費やす。収入よりも仕事の内容で、仕事を選ぶ。時として、生活をするための主たる収入(生活のために労働力を提供する仕事)が、自分本来の仕事(天職)ではないこともある。

特に、この階層において養成課程にある人々の収入は、本来の仕事からは得られないことが多く、パートタイムの仕事で、生活をやりくりしている人も多い。この階層に属している人々の中で、実際に成功できるのはごく一部である。そのリスクを認識しても、自分の才能に懸けるのが、この階層の人々である。

このフロリダの創造的階層論は、マズローが人間の欲望を生物学的・心理学的視野から分析したのに対して、職種という労働経済学的視野から現代社会の仕事と人間の関係を分析したものである。その意味で、フロリダの理論は、ルターの天職理論に近い。

ルターとフロリダとの違いは、時代による社会の経済発展状態の違いによる。ルターの

時代には、ほとんど意識されることのなかった知識と創造性という新しい視点が、フロリダの議論には追加されている。ところで、知識は明らかに後天的なものであり、一部の創造性は先天的と言ってよいものである。

それでは、自己実現は天賦の才を与えられた、ごく一部の幸運な人々だけのものなのだろうか。少なくともルターは、そう考えてはいなかった。人には、それぞれ農夫として、鍛冶屋として、靴職人として神から与えられた才能と、この世での使命がある。それこそが天職である。

与えられた才能を十分に発揮しようと努力しないことは、怠惰であり、悪であると考えた。この部分に関しては、イギリスの清教徒たちも同じであった。ただし、清教徒たちが仕事の内容を狭く限定せずに、他人に役立つ仕事と広く考えたのに対し、ルターは天から与えられた仕事と限定している。ルターは、中世的で世襲的な職業観から抜け出せなかった。

創造的階層論を主張するフロリダの視点は少し複雑である。創造的階層に属する人々やそれを望んでいる人々は、成功するかどうかは別として、自己実現を目的として生きている。自己実現のために教育・訓練の機会を選び、住む場所を選び、周囲の人々との交流の仕方を選び、仕事を選ぶのである。

この階層に属している人々の数は、この 30 年間に急激に増加したが、現実社会では、この階層には属していない人々の方が多数である。ここに暗示されている事実は、米国社会では、自己実現を求めることすら知らない人々が多いことである。つまり、社会的格差の問題が、背景に存在している。これは、中世の実態とは大きく違うものの、社会の構成員の一部だけに門戸が開かれていると言う意味では、本質的に同じ問題である。

第 6 節　自己実現と現代社会における階層格差問題

山田昌一は、その著書において、日本が希望格差社会になっていることを指摘し、警鐘を鳴らした[64]。今や日本社会は、中流が消失し、一部の上流階層(「勝ち組」と表現される)と、大多数の下流階層(「負け組」と表現される)に 2 極分化したとしている。

そして、山田はその社会階層が世代を超えて固定化しつつあることの問題を提起した。すなわち、上流階級の家に生まれた者が上流階級の構成員となり、下流階級に生まれた者は、下流階級から脱出しようとする希望まで失っているとの主張である。

最近の日本では、偏差値の高い大学には、裕福な家庭の出身者だけが集まる傾向が著しく、大学の偏差値の順位と、学生の出身家庭の平均収入額に強い相関がみられるようになっているとの指摘がある。

また、偏差値の高い大学の卒業生が、一流企業に入社する傾向は強く。これらの人々が、将来、高収入を得る人々に育ってゆく。裕福な家庭ほど、子供の教育費を捻出し易いため、結果としてその階層に属する家庭の子供たちの偏差値も高くなってゆく構造が確立した。

フロリダも山田も、現代社会が 2 極化し、自己実現を求めて仕事を選べる人々と、そうでない人々に分化しているという認識では、一致している。大学教育が一般化した米国や日本の社会では、もはや大学を卒業すること、大学院を修了すること自体は、何も意味しない。

また、日本の大学における専門(職業)教育は、米国のそれに比較すると、著しく貧弱である。また、社会の中でいかに働くかを教えると言う意味での教員の質も劣っている。従って、大学卒業時の学生の仕事の遂行能力から見ると、日本の学生の能力は、米国などの学生の能力に比較して劣っている。

20 世紀末まで、日本の社会では、大学を卒業することは、特に工学系学部の場合、ほぼ就職と同義語であった。就職を希望しているにも関わらず、就職ができないという例はほとんどなかった。企業側も、大学内での推薦プロセスを信用し、大学側が推薦した学生を就職試験で採用しないとすることは、極めてまれであった。

この就職環境は、21 世紀に入って、大きく変わった。大学の推薦は意味を持たなくなり、自由応募による、自由競争の時代に移行すると同時に、いわゆる大学新卒一括採用の枠を、企業は縮小した。これは、日本企業の終身雇用制度と大学新卒一括採用制度が、機能しなくなりつつあることを示唆している。

従来であれば、企業は、大学が推薦する学生が、大学で何を学び、どのように大学生活を過ごしたかにはほとんど興味を示さなかった。多くの企業は学生の出身大学だけに興味を持っていた。しかし、自由応募で入社試験を受けに来る学生に対して、そのようないい加減な選抜は企業の命取りになる。

企業人の目で個々の学生を吟味し、採用に値するかどうかを、企業の自己責任で決定しなければならない。その選択の誤りリスクは大きく、企業側は慎重にならざるを得ない。当然、企業は、学生がどのような性向をもち、学生が大学で何を学び、大学生活をどう過

ごしてきたか、入社後企業内で何をしようとしているか、何ができそうかを問題にせざるをえない。

このような現実の変化は、さらに社会的な階層格差を増幅し、固定化する。裕福な家庭の出身者は、ヨーロッパの上流(貴族)階級出身者と同様、学生時代に、外国への遊学・留学、一流企業における長期のインターンシップ、海外でのボランティア活動などに参加する経済的余裕があり、有利である。

経済的な余裕の乏しい家庭の出身者は、例え大学の成績は同程度であったとしても、アルバイトで働かなければならない時間のため、十分に学ぶことも、その他の学びにも参加できる機会や、将来の社会活動に役立つ経験を積む機会なども限定されている。

東京証券取引所一部上場企業への就職の門戸は、誰にも開かれていると言う時代ではない。このことが、大学における学びと、天職との乖離をさらに大きくしている。周囲の適確な指導で、自らの適性に合った適切な大学に入学し、良い教員にめぐり合う幸運に恵まれ、適切な企業への就職にたどりつける人々は、天職を見出し、自己実現へ向けてまい進できる。

しかし、偏差値だけを頼りに志望大学・学部を決定するという無責任な指導を受け、自らの適性とは無関係な大学・学部に進学した人々には、進学した大学の偏差値がいかに高くても、自己実現への道は遠のく。

現代に生きる人間にとって、仕事は自らの人生の質を左右する唯一のものであると言える。一流企業へ入社することが、自己実現ではない。その企業で、何を行い、何を為すかが問われている。

一流企業への入社は、世間体と言う言葉を使えば、自らの社会的認知度を高めるだけである。数年すれば、その企業でどう生きているかが問題にされる。年収や職位は、客観的で共通的な基準でもあるので、すぐにそのような見かけの数字や名前が気になってくる。

しかし、その企業を退職するとき、課長であるか、部長であるか、取締役であるかにどのような意味や違いがあるのか。さらに言えば、人生の終末を目前にして、自らの人生を振り返ったとき、年収や職位を考える人はいない。

それ以上に、自分は何を学び、どのような仕事に就き、どのような成果を残し、どのように成長し、生きてきたかを振り返るだろう。場合によっては、その仕事は、就職した企業での仕事ではなく、無償で務めた非営利団体の仕事かも知れない。その仕事こそが、天職であると言えよう。

天職を見出し、その仕事に人生をかけ、何かを成し遂げたと認識できるとき、自己実現は現実になる。また、時としては何も成し遂げることができなかったとしても、一生懸命仕事に打ち込んだこと自体が、心に残ることもある。人によっては、その時は永遠に来ないかもしれない。

それは、その人の努力が足りないからではなく、その人の要求する水準が高いからである。周囲の人間から見れば、どちらも変わりはない。自己実現を成し遂げた、幸福な人な

のである。その意味で、自己実現は相対的な概念ではなく、絶対的な概念である。

第8章　資本主義社会における企業間競争と国家の役割

第１節　ソフトウェアの知的財産権保護問題[32]

ソフトウェアは、資本主義社会が新しく手にした無形財である。それまでの人類の社会は、コンピュータソフトウェアのような、産業的に作り出せる、大規模な無形財を手にしたことがなかった。

「産業的に作り出す」ということは、資本集約的であり、かつ労働集約的な面もあることを意味する。そのようなことから、完成したソフトウェアの知的財産権を法的に保護することは、資本主義社会においては重大な問題となる。

さらに、現代のソフトウェア開発は、極めて知識集約的な側面も併せ持つようになってきている。特に、組込みソフトウェアの分野では、これまでの技術では実現不可能であった機能を、組込みソフトウェアによって実現するため、その実現には新しい理論の応用など、様々な分野の知識を統合して問題を解決し、1 つの製品として統合してゆくことが求められる。このため、長期にわたる実験や研究の成果が基礎になっている例も多くなっている。

このように、ソフトウェアは、資本集約的、労働集約的、かつ知識集約的な産業活動であるソフトウェア開発の成果ではある。しかし、その成果物は無形財であり、簡単に類似物を開発できる性質を持っている。それが単なる複製であっても、一旦コピーされれば、どちらがオリジナルの成果物かは判定不能である。

このような理由で、ソフトウェアの知的財産権を法的に保護することは、企業における多大な投資によって開発されるソフトウェアに対する様々な権利を、第三者による侵害から守ることが必要な、資本主義の国々では重大な関心事とならざるをえない[xlv]。

現在の時点で、ソフトウェアの知的財産権を法的に保護する方法として、著作権法による保護、特許法による保護、商取引上の秘密事項(トレードシークレット)としての保護の 3 種類の保護が一般的である。しかしながら、法律での保護は、各国の法体系による差異があり、国家間をまたがる問題については、懸案事項も多いのが現実である。

さらに第 2 節でも議論したように、ソフトウェアに関する知的財産権の保護問題は、歴史的には 1970 年代以降の社会での問題である。それまでのソフトウェアは、コンピュータメーカが作成したものであっても、基本的にオープンソースであり、ソースコードを公開し、しかも無償で配布されていた。

これは、物理的に実現された様々な製品に付随して配布される説明書と同じ扱いであったと言える。そのような説明書でも、その内容は著作権で保護される対象である。

1970 年頃から、ソフトウェアの法的保護が問題になった理由は、IBM が独占禁止法裁判で敗訴したため、開発したソフトウェアをハードウェアとは独立させた製品として販売する必要が生じたことによる。販売するためには、一般の書籍と同様に、その内容で

[32]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.125-145 参照

あるソースコードを、無断での複製から守る必要があったためである。

このことが、IBM において開発される全てのソフトウェアに著作権を主張するためのコピーライトノーティス(copyright notice)が記載されるようになった理由である[xlvi]。さらに、IBM では著作権の移転に関する米国の法制度を考慮して、ソフトウェア開発業務に従事する全ての従業員に対して、従業員が業務上で作成した全てのソフトウェアに関する著作権の権利を、本人から IBM へ移転することに同意していることを証明する契約書を提出させた。

当時の米国社会でも、ソフトウェアを実現するための基本的アイデアであるアルゴリズムを特許法で保護すべきであるとする意見があった。これに対して、米国特許庁は、ソフトウェアのアルゴリズムは、自然法則の応用ではなく、純粋な数学的思考の結果であり、特許の保護対象である発明には該当しないとの理由から、ソフトウェアまたはそのアルゴリズムを特許で保護することは認められないとする立場を表明した。

著作権や特許権などの既存の知的財産権保護の枠組みを適用して、ソフトウェアを保護することが妥当であるかどうかは、自明の問題ではない。特に、ソフトウェアが知的な資産であることを考慮すれば、それを多くの人が確認し、必要があれば改善できる仕組みを整備することで、ソフトウェアを人類全体の資産として、多くの人々に利用可能とし、さらによりよいものに進化させてゆくことができる。そのような目的で、オープンソースソフトウェアのライセンスは提案された。

このライセンスの概念は、ソフトウェアの利用と進化のための改善を可能とするため、従来は著作権で保護してきたソフトウェアのソースコードのコピーを、全く自由にさせると言う新しい方法を採用する。しかしながら、ソフトウェアを作成した人々の貢献を認め、その痕跡を確実に残すために、ソースコードを改変した場合の手続きを決め、それを順守することを利用者に要求することで、著作者であるソフトウェア技術者(または、それを開発した企業)の権利を守ろうとする。

ただし、オープンソースソフトウェアのライセンスでは、ソフトウェアのソースコードを無断コピーから守ることはできない。従って、有償で配布される市販のソフトウェアについては、ライセンスによってその知的財産権を保護することは不可能である。

ただし、著作権を利用して保護することが可能なソフトウェアの知的財産権は、ソフトウェアの著作物としての側面のみである。つまり、ソフトウェアを無断複製から守ることは可能であっても、その基本的なアイデア自身を守ることはできない。

ソフトウェアの新しい実現や新しい機能に関する斬新なアイデアを模倣から守ろうとするとき、それを可能とする現在の法的手段は、特許権である。ただし、特許権が発明として本質的に認められるためには、そのアイデアが自然の法則の新しい応用による、経済的に価値を生むものの実現であり、そのアイデアが従来の類似品に応用されている既存のアイデアからは容易に導き出せない新規性を持つことを条件としている。この国際的な発明の定義のため、一般に数学的な計算過程として実現されているソフトウェア

は、法的保護の対象にならない例がある。

特に、ソフトウェアの価値の源泉となっている例の多いアルゴリズムについては、事前の法則の応用と言う側面が弱く、逆に数学的な計算過程としての側面が強い。このことから、ソフトウェアの価値を生み出しているアルゴリズムが、その製品の価値を生み出す源泉となっている場合でも、そのアルゴリズムを発明として特許権で守ることがしばしば困難である例がある。

このソフトウェアに対する特許権の適用については、1970 年代までは多くの国々で類似の対応が採られていた。しかし、1980 年代に入って、ソフトウェアの経済的価値が注目され始めた時から、各国での対応が少しずつ違ってきた。

特に米国では、特許による知的財産権の保護が、自国の製品に対する知的財産権保護の重要な手段になるとの考え方に基づき、プロパテント政策の採用を決定した[xlvii]。この政策の採用により、米国特許庁もソフトウェアに対する特許権による知的財産権保護を積極的に認めようとする方針に転換した。

我が国の特許庁においても、米国の特許庁における動向を捉え、1980 年代の末から、「プログラム特許」の概念を打ち出し、製品の実現において組込みソフトウェアを応用している場合、そのソフトウェアの実現に関するアイデア(主としてアルゴリズム)に対する特許申請を必要に応じて認める方針を決めた。しかし、純粋なソフトウェアの実現アルゴリズムについては、今日でもその対象外とされている。

例えば、米国では認められたアマゾンの「ワンクリック特許」は、日本の特許庁においては、紆余曲折を経たが、最終的に認められなかった。ワンクリック特許は、アマゾンで本や商品の発注をしたユーザの決済情報などをデータベースに格納し、次の注文の発注時も、ユーザのログイン情報に基づき、注文情報の一部である決済情報をデータベースから自動的に読みだすことで、ユーザがいちいち入力する手間を省けるようにした、純粋にソフトウェアでの処理を特許化しようとしたものである。

1.ソフトウェアの著作権

著作権法に基づくソフトウェアの知的財産権の保護は、一般的には、ソフトウェアの違法コピーを防止するための手段として考えられている。ソフトウェアのソースコードは、人間が特別な言語(プログラミング言語)で記述したものであり、著作物としての性質をもつ。

著作権法は、15 世紀のグーテンベルグによる活版印刷機の発明によって、著作物の大量コピーが可能となり、それによって経済活動(出版)が可能になったことから、主として著者と出版社の権利を保護する目的で導入された。著作権法によって、第三者による大量の(違法)コピーの制作と販売が法的に禁止され、著者や出版社の権利が守られる。

しかしながら、小説などの一人または少人数の著者によって執筆される著作物の権利を保護する目的で制定された著作権法を、多数の技術者集団によって産業的に開発され

るソフトウェアに適用することには、本質的な問題も存在する[xlviii]。例えば、日本の著作権法には、著作者の著作物に対する絶対的な権利を保護する目的で、「著作人格権」なる権利が認められている。

この権利は、「著作物の内容を、著作者の了解なしに、出版社が変更することはできない」とするものである。これは、例え誤字であったとしても、著作者の了解なしに、誤字を修正することはできないことを意味する。著作権法のこの部分をソフトウェアに適用することには、重大な問題がある。

例えば、プログラムの一部に少し複雑な誤りが潜在していたとする。これは、設計上の間違いではなく、プログラミング上の誤りであるとする。A 社は、そのソフトウェアの設計・実現を B 社に依頼していたとする。すなわち、その誤ったプログラムを記述したのは、B 社の社員である。

A 社は納品されたそのソフトウェアを利用していたが、ある日、そのソフトウェアに誤りが潜在していることに気付いたとする。仮に、そのプログラム上の誤りを特定し、修正方法が分かったとしても、A 社はその誤りを修正できない。それは、A 社に著作権はなく、その部分の著作人格権は B 社の社員に帰属しているからである。

この問題に対して、日本の著作権法では、著作物の中にプログラム(ソースコード)を含め、それに対して一部特別な取り扱いを許している(プログラム著作権)。このプログラム著作権に関する法律では、プログラムに明らかな瑕疵と判断される誤りがある時、プログラムの利用者はそのプログラムを開発したプログラムマの承諾を得なくても、その誤りがある部分の修正が可能であるとしている。

これは、ソフトウェアが一般の書籍と違って、コンピュータで実行されるため、誤りを放置すると、ユーザの要求に沿ったデータの処理ができない問題が発生するからである。この問題を、現実的な問題として捉えて、回避しようとした法的対応であると言える。

また、著作権法が適用されるのは、ソフトウェアの設計ではなく、表現としてのプログラムであり、保護できるのは表現としてのプログラムに限定される。例えば、全く同じ設計のプログラムを、別のプログラミング言語で記述しなおした場合、表現上は全く異なるものとなり、違法コピーとは言えなくなるからである。ただし、同じプログラミング言語での記述の場合、記述の順序がある規則によって変更されている場合などについては、表現上は同一でなくても、違法コピーとみなされる。

著作権法によるソフトウェアの知的財産権の保護に対して、一部の専門家からは、単なる表現上の類似性を問題にする違法コピーの概念では、ソフトウェアに関わる本質的な知的財産権は保護することが困難であるとする考え方が根強くある。これは、上述したように、同じ技術的なアイデアに基づいたソフトウェアの設計であっても、そのプログラムを記述するプログラミング言語が異なれば、同じアイデアに基づいて作成された 2 つのプログラムが、本質的に同じものであることを判定することが困難になるからであ

る。

さらに、米国の法律家の中には、著作権によるソフトウェアの知的財産権の保護は、ソフトウェアを作成したプログラマが直接書いたソースコードに限定され、そのソースコードをコンパイル処理して生成されるコンピュータの実行形式であるオブジェクトコードや、それを読み取り専用メモリチップ ROM(Read Only Memory)に焼き付けたものは、人間が読める形式ではないので保護の対象にはならないとする意見もあった[78]。

実際の米国における裁判での判例でも、そのような法的解釈に基づくものもあった。ただし、米国の法律家の中には、著作権法の規定では、オブジェクトコードやそれを ROM に焼き付けたものは、保護の対象とはしていないものの、著作権法の精神で保護すべきとしている違法なコピーの範疇に入るとする意見もあった[xlix]。

1980 年代に IBM と富士通の間で争われた、IBM が開発したオペレーティングシステムを富士通が違法に複製して利用しているとした著作権侵害の裁判では、IBM は極めて巧妙な方法で富士通のオペレーティグシステムが IBM のオペレーティグシステムのコピーであることを示した。

この時、富士通のオペレーティングシステムのソースコードは、IBM のオペレーティングシステムのソースコードと簡単には、1 対 1 の対応が取れなかった。つまり、ソースコード上では、部分的には似たようなコードの列が存在していたが、全体的に見るとそれらが現れている場所に共通的な特徴がなく、コピーとは言えなかったのである。

このような状況下で IBM は、IBM のオペレーティングシステムのソースコードに残存していたいくつかの誤りに注目した。プログラマは、ソースコードを作成する時、偶然に誤りを犯す。そのような誤りの中には、ユーザが利用しているオペレーティングシステムの機能の実行に全く影響を与えないものがある。しかし、その一連のソースコードの意味を注意深く分析すると、プログラマが意図していたことと、ソースコードの表現の間に矛盾が含まれるものがある。それが誤りである。

巨大なオペレーティングシステムのソースコードは、100 万行を超える巨大な文書である。その文書の中に、そのような誤りはある頻度で存在している。人間は、どんなに優秀なプログラマであつても、ある確率で誤りを犯す。

多くの場合、そのような誤りはテストで発見され、修正されるので最終製品に残存する例はまれである。しかし、数十回に 1 回程度、そのような誤りが見過ごされることがある。特に、ユーザが利用する機能に無関係な部分の誤りの場合、発見されずに残存する確率は高くなる。

プログラマが誤りを犯すと言っても、2 人の全く異なった人間が同じ仕様からプログラムを作成したとしても、全く同じ誤りをする確率は極めて低い。仮にそのようなことが起こる確率を千回に 1 回とする。そのような誤りが、問題となっている 2 つのソフトウェアに 3 か所見つかったとする。

前述した確率から計算すると、そのような 3 つの誤りが同時に存在する確率は、10 億

回に 1 回である。つまり、一般的には起こることはないほど小さい確率の事象が発生したと言える。これは、理論的に不自然である。従って、2 つのソフトウェアのうちの一方が、他方のコピーであるとする方が合理的な考えである。

このような方法は、今日、あるソフトウェアが他のソフトウェアのコピーであるかないかを検証する手段として、故意に誤りを埋め込んでおき、それが存在することでコピーであることの証明とするために利用されている。しかし、この方法もコピーを人手で注意深く実施すれば、すり抜けることも可能である。その意味で、完璧な方法ではない。

上述した IBM と富士通のオペレーティングシステムの無断複製に関する裁判の結果、両社は以下のような妥協案に基づく合意に達し、2 社間での契約を締結した。その内容は、富士通は同社のオペレーティングシステムのソースコードの一部に IBM 社のソースコードがコピーされて利用されていることを認め、損害賠償金を支払う。また、IBM は富士通に対して、その後に開発するオペレーティングシステムのソースコードを全て開示する。富士通は、IBM が開示するオペレーティングシステムのソースコードを、「クリーンルーム」と呼ぶ特別な施設内でその社員に読ませると言うものであった[1]。

富士通が IBM とのこのような妥協に賛同した理由の一つには、それまでに、同じ国産コンピュータメーカである日立製作所が、1982 年に米国内において実施されたおとり捜査によって、日立製作所社員 2 名が IBM の秘密文書のコピーを入手しようとした疑いで、米国捜査当局に検挙されたことかに始まる、いわゆる「IBM 産業スパイ事件」に関係して、同様の内容の契約を締結していた背景もあった[1i]。

日立製作所は、その合意において、IBM に対して損害賠償をすること、IBM の機密情報を変換すること、1988 年までの間、日立製作所が開発コンピュータと基本ソフトウェアについては IBM による事前検査を受け入れることなどを認めていた。IBM も、IBM が開発した基本ソフトウェアのソースコードを日立製作所に対して開示し、日立製作所の社員による閲覧を認めることを承諾していた。

クリーンルームとは、富士通の社員が手書きのためのノートやパーソナルコンピュータなどを持ち込まずに、人間だけが入室し、IBM が提供したソースコードと機能仕様書を読んで理解するための特別な部屋を指す。富士通の社員は、そこで理解した機能仕様とソースコードの一部に基づいて、自分の作業する場所へ戻り、IBM のオペレーティングシステムと類似した機能をもつソースコードを作成するのである。この施設の存在によって、富士通は IBM のオペレーティングシステムに類似したオペレーティングシステムは開発できるが、それは IBM のものの複製ではない。

ここで問題になっているのは、著作権の侵害であって、機能そのものの必要性に関する認識や、そのような機能をどのように実現すべきかのアルゴリズムについての基本的なアイデアを保護しようとする姿勢ではない。IBM の視点に立てば、そのような新しい機能は、製品が世に出れば、いずれ富士通の技術者にも知れ渡るという認識になる。

そうなれば、その実現方法を考えることは、それほど困難なことではなく、ソースコ

ードを見なくても実現を考えることは可能だとする認識である。従って、ソースコードを隠すことの意義はそれほど高くないのである。

2.ソフトウェアの特許権

ソフトウェアの機能をどのような計算で実現するかを決める計算方法のことを、アルゴリズムと呼ぶ。これは、9 世紀のアラブの大数学者(代数学が専門)、アル・フワリズミの名前に由来するものである。

与えられた一連の変数の値から、要求される変数の値を算出するための計算手順を言う。計算に利用するアルゴリズムの良さは、計算の回数や、計算過程で利用する中間変数の数に大きな影響を与えるため、結果として計算速度や、計算に必要な記憶装置の量に影響する。記憶装置の容量が少なくてすみ、計算回数の少ないアルゴリズムほど効率が良いとされる。

一般的に、ソフトウェアの品質の良さは、このアルゴリズムの良さ(計算効率)によって決まる。従って、ソフトウェアの開発においては、効率の良いアルゴリズムを考えだすことが重要となる。

この効率の良いアルゴリズムを考えだすことは、極めて知的な作業であり、機械の設計や電気・電子回路の設計に似た活動である。その活動の成果が、新しいアルゴリズムであるとすれば、良いアルゴリズムを考案することは、新しい機械などの発明と似ていることになる。

1960 年代から 1980 年代の前半まで、米国特許庁は、上述したような特徴をもったソフトウェアとそのアルゴリズムに対しては、それらが純粋な数学的思考の産物であり、厳密な意味で自然法則を応用したものではないと言う理由から、それらを特許権で保護することはできないとする立場を採ってきた。しかし、1980 年代に入って、電話会社ベルの研究組織であるベル研究所による特許申請によって、その基本方針を転換した。

それは、同研究所の研究員が、巨大な多元連立 1 次不等式を高速に解くアルゴリズムを開発し、その特許を申請したことがきっかけになった。高度な数学を応用したアイデアでも、経済的な価値の高いものは、特許法でその知的財産権を保護する必要があると言うことを理由に、特許で守るべきアイデアの対象を拡大する政策に転換したのである。

この経済的価値の高いアイデアを特許権で保護すると言う考え方についても、従来から米国社会では特許法でその知的財産の独占的な権利を保護するべきではないとする法律家が多かった。これは、そのような独占的な権利を認めることが、科学技術の進歩を阻害する危険性が高いと考えたからである。

そのような新しいアイデアは、学術的な価値があるもので、論文を発表することで公知のアイデアとすることが自然であると考えられる。すなわち、公知のアイデアとすることで、人類の科学・技術の発展に貢献させることの方が健全であるとする考え方が一般的であった。

1980 年代になって米国特許庁がソフトウェアに対する特許法による知的財産権の保護を認めるまで、IBM などの企業では、社内で開発された新しいソフトウェアの技術を、ハードウェアと密接に関係したアイデアの場合はハードウェアの特許として申請していた。そして、純粋なソフトウェアの実現に関するアイデアの場合は秘密情報として管理していた。

秘密情報とする経済的な価値のないものについては公知のアイデアとするために公開技術情報として出版した。さらに、学術的な意味の大きなものについては IBM が出版する論文誌に掲載すると言う多様な対応を採っていた。

特に、公開技術情報としてまとめて出版し配布する方法は、IBM が特許として申請し、その権利を確保するまでのことはないが、他社が類似のアイデアを特許として申請し権利を確保することを積極的に防止する方法として重要であった。ソフトウェアに関する新しいアイデアの場合、それが IBM で開発する製品に直接関係し、利益を生み出す可能性が高い場合を除き、IBM は積極的にそのような情報を開示して、競合他社がそれを特許化して知的財産とすることを、未然に防ぐ必要があったからである。

1980 年代に米国政府がプロパテント政策を採用して以後、米国特許庁は、アルゴリズムが発明として申請された場合、それを審査し、基準を満足していると判断されれば、その考案を特許として認可する体制をとっている。このことは、ソフトウェアの開発が産業的に実施されている現実を考えれば、合理的な判断であると言えよう。

ソフトウェアを開発しようとしている企業は、その活動に対して多大な人的資源を投入し、投資を行っている。この投資を、知的財産権で保護しようとすれば、その結果を記述したプログラムを保護することだけでは不十分であり、その根底にある、新しく考案されたアルゴリズムを保護することが重要になるからである。

しかし、従来の発明の概念には、特に数学的な問題の解き方に関する新しい方法の考案は含まれていない。16 世紀にオランダで特許の概念が創案されて以降、特許権は発明に対してのみ適用されるもので、物理法則そのものの発見や、数学における問題の解法は、発明に属さないとされてきたからである。数学における問題の解法(計算方法)が発明とされない理由は、それが自然の法則を応用したものではないからである。

例えば、今までのどの方法よりも早く、与えられた正の整数の根の値を求める計算法は、それがどんなに有用なものであっても、数学上の問題に対する解法であり、発明とは言えない。このような解法は、数学的には価値があっても、20 世紀までの社会においては、一般的に経済的な価値はなかったからである。

この発明に関する厳密な解釈を適用するのが、ヨーロッパ諸国の特許法の解釈である。従って、ヨーロッパ諸国においては、ソフトウェア開発において考案された新しいアルゴリズムは、それがどんなに経済的な価値のあるアイデアであっても、特許としては認められない。

このことは、米国における特許法の運用と大きく異なっている。ちなみに、日本にお

ける特許法の運用では、基本的に「アルゴリズムは発明として認められない」とする立場を取りながらも、「それが特定の装置の実現に関与するものであれば、発明として認めることもある」と言う立場を取っている[lii]。

特にヨーロッパ諸国政府の立場と、米国政府の立場が著しく異なるのは、米国が先発明主義など独自の考えに基づく発明権の定義に基づいて、特許法を運用していることに起因していると言われている。先発明主義とは、2 つのアイデアのどちらが特許権よってその知的財産権を保護する対象になるかを判断する時、それぞれの発明がいつなされたのかを最優先する考え方である。

このため、それぞれの発明の発明者にとっては、自分がいつ、その発明を考え付いたのかを明確にできる証拠を残すことが重要となる。つまり、特許の申請は遅くても、自分の発明が他の者の発明よりも早ければ、その権利は自分のものであることを主張できる。米国以外の国々の政府は、パリ条約に基づいた「自然の法則を利用したものを発明とする」という基本的な発明の定義から、大きく逸脱したものを発明として認めることはできない状況にあるとともに、特許としての申請の時点が、特許権を認める場合に重要となる制度を採用している。

このように、ソフトウェアの開発に応用された新しいアイデアに対する法的な保護策が、地域によって異なることは、ソフトウェアの健全な進歩やソフトウェアに関係した産業の健全な発展を考えると、それを阻害する要因になりかねない。つまり、A 国ではその知的財産権が特許法によって保護されているのに対して、B 国では同じ知的財産権が特許法で保護されないと言う事態が発生する。

6.1 節において述べた、アマゾンの「ワンクリック特許」は、まさにそのようなアイデアの例である[liii]。確かに、その実現は容易であり、専門家であればどうすればそれが実現できるか、容易に予想できる。その意味では、従来の特許法での新規性の要件を満足してはいないと言える。

ただし、そのような「ワンクリック特許」で提供しようとした機能そのものは、それまでに存在しておらず、また、それを考えた出すためにはかなりの問題分析力を必要とするとも言える。その意味では、日本のプログラム特許が認める特許の範囲には該当しないが、知的財産としての価値は高いものである。

また、機能の実現が容易であることは、それを法的に保護する価値が高いとも言える。米国においては、この特許はしばしば「ビジネスモデル特許」の範疇に入るものとして議論される。このことも問題を複雑化させている要因の一つである。つまり、米国の特許法では、そのような特許申請も、特許として認めることができるのである。そして、その経済的価値は大きい。

技術者として維持すべき倫理的な枠組みから言えば、他者の創造的な活動の成果として、提案された新しい(それまでに存在しなかった)機能や、新しい実現方法(計算のためのアルゴリズム)に類似したアイデアを実装したソフトウェアを作成することは、アイデ

アの複製(真似)と言う意味から、いずれにしろ行うべきではない行為である。また、その経済的な価値を考慮すれば、資本主義社会においては、それらに対する独占的な権利は、侵害されてはならない権利である。その意味で、知的財産権を守るための法的な整備は、社会的に必須の条件である。

3.ソフトウェア権とその法的保護

このような状況を反映して、1980 年代の日本においては、米国 IBM 社と複数の日本のコンピュータ製造メーカ間に発生したオペレーティングシステムの不正コピー問題をきっかけとして、当時の通商産業省を中心として、中山信弘らの提案による、ソフトウェアを対象とした新しい知的財産権に関する法律の制定を指向する動きがあった[liv]。これは、1972 年の通産省重工業局ソフトウェア法的保護調査委員会中間報告において、ソフトウェア開発投資の推進、ソフトウェアの活発な流通を通じた効率的な情報化投資の実現を目指した対応策の提言が基礎になっている。

これに対して、当時、ソフトウェア開発の分野で世界をリードしていた IBM は、「ソフトウェアの知的財産権は、著作権法で守るべきである」との立場を取り、この通商産業省の考え方に反対した。当時の IBM の主張の根底には、著作権であれば、申請も認可も必要なく、プログラムが書かれると同時に著作権が発行するため、柔軟な運用ができるとの理解があった。また、1973 年文化庁著作権審議会第 2 小委員会も、プログラムは既存の著作権法に若干の修正を加えることで、著作権法によって保護しうるとするとの報告を行った。

プログラム権法の基本的な考え方は、ソフトウェアの開発に必要な投下資本の回収を可能にする方法を確保することで、企業によるソフトウェア開発へのインセンティブを増すこと、類似のソフトウェア開発への重複投資を防止し、新しいソフトウェア開発へ積極的に取り組める環境を整備すること、そしてソフトウェア流通促進のための情報公開や争議の調停制度を確立することなどにより、日本国内におけるソフトウェア関連産業の活性化を図ることであった。

ここでは、1972 年のソフトウェア法的保護調査委員会の中間報告で提起された、ソフトウェア関連産業の活性化のためには、ソフトウェアの不正な使用と盗用を禁止することで十分であるとの認識が根底にあった。つまり、資本を投入して類似のソフトウェアを開発することを禁止することまでの必要はないとする考え方が根底にあったと言える。

具体的には、著作権には考慮されていないソフトウェアの使用権概念を導入し、逆に著作権にある著作人格権を除外して、改変権、複製権、貸与権などを設定することとした。さらに著作権が、著作物の作成時に自動的に権利が発生するとしているのに対して、特許権と同様に権利の発生時期を明確にするため、登録機関を設置し、登録制とすることが提案された。また、紛争解決のために特許法と　似たような裁定制度を導入し、その審査委員としてプログラムの専門家を任命し、斡旋、調停、仲裁など裁判外での紛争

解決を可能にすることも提案された。

この「プログラム権法」は、米国でも議論になっていた著作権や特許権によるソフトウェアに対する知的財産権の保護問題について、新しい法律を整備して、著作権で問題になっていたオブジェクトコードなどを含むソフトウェア実体の保護や、特許権で問題になっていた機能実現のための新しいアイデアの保護に焦点を置いていた。、機能の実行で利益を得るソフトウェアユーザがソフトウェア使用権を購入したと考えることで、その正当な権利を保護すると言う視点から守ることを中心に考えられたものである。

類似する機能をもつ 2 つのソフトウェアに対して、著しく異なる使用料をユーザに負担させることのないようにすることが重要だと考えられていた。そのため、ソフトウェアを登録制にして、類似性を審査することで、多重な開発投資を防止するとともに、利用者の経済的負担を軽減しようとするものであった。

この日本の通産省による「プログラム権法」の考え方に対しては、IBM の米国の研究者の一部にも賛同するものがいたが、最終的には IBM の弁護士達の意見と、それに押された米国政府の強い意向で、「プログラム権法」案は法制化されずに終わった。文化庁は、1984 年に著作権第 6 審議会の中間報告を出し、法人著作の概念を導入した著作権法の改正を提案した。このとき、IBM と通商産業省との間に立ち、我が国の著作権法の改訂によってソフトウェアの知的財産権を保護することを主張したのが、当時、日本 IBM に籍を置いていた高石義一らの法律家(弁護士)たちであった[lv][77]。

この「プログラム権法」の考え方は、特許権による知的財産権保護の対象になっていなかった一般のソフトウェアの知的財産権をどう守るべきかに関する議論が国際的に行われていたにもかかわらず、日本のソフトウェア市場と言う極めて限定された視野で検討されたものであった。問題の本質には、それまでの特許権や著作権では対応できない問題があり、世界中の法律家やソフトウェア技術者を巻き込んで議論することで、より普遍的な知的財産権保護に関する法的整備を考えることが可能な環境にあったにも関わらず、それを逃したことは大変残念であった。

法律の原案として見た場合、IBM が主張したように、「プログラム権法」と従来の著作権法によるソフトウェアの知的財産権の保護には、著しい違いはない。特に、米国における著作権の適用においては、企業組織に所属して、業務としてソフトウェアを開発している技術者がその著作権を、技術者を雇用する企業に移転することは可能であった。

また、著作権の場合、著作が作成されると同時に権利が発生し、その登録は必要ない。この事務的な処理の簡単さは、法的保護を厳密に実施しようとする場合には、問題を生じさせる可能性があるものの、企業にとっての利点も多かった。

当時、既に、後述するストールマンのフリーソフトウェア運動が始まっており、全く新しい知的財産権保護の方法も模索されていた。ユーザのソフトウェア使用権に注目し、さらにソフトウェアの発展のための改変を許すべきとする、現在のオープンソースソフトウェアのライセンス概念に似た法的枠組みも盛り込まれており、広く意見交換を行え

ば、様々な改善案を集約することで、より普遍的な知的財産権保護の枠組みを提案できた可能性はあったと言えよう。

この議論の過程では、ソフトウェアの実現に関する知的財産権を、ソフトウェア開発を実施した企業または個人と、そのソフトウェアを利用する契約を締結したユーザ(企業または個人)との間における商取引上の契約事項として、秘密情報の第三者への開示を禁止する条項(トレードシークレット条項)を記述することで、知的財産権を保護することも可能であるとの主張もあった。これは、今日でも一般的に利用されている方法であり、特殊なソフトウェア開発においては有効な法的手段であると言える。

4. OSS のライセンス概念

OSS におけるライセンスの概念は、ソースコードを公開して配布する OSS において、その作成者の著作権を守り、また、他者による内容の吟味・検討、さらに修正や改善を可能とし、さらに改善されたソースコードをもともとのソースコードと同様に広く配布することを可能にするための法的な根拠を与えている[lvi]。ただし、ライセンス規定の詳細は、個々の OSS によって異なっている部分がある。

このライセンス規定によって、修正または改変された OSS のソースコードは、OSS として配布しなければならない制約をユーザや関係者に課している。このため、OSS に商用ソフトウェアの一部分を統合し、新しい商用ソフトウェアとして販売することはできない。また、特定のユーザだけを限定して配布することもできない。

個別の問題解決に供するアプリケーションソフトウェアの開発に、同様な OSS の枠組みも適用可能であり、それが有効に機能するかどうかは、ソフトウェア工学において未知の課題である。仮に、問題そのものが多くの組織において共通であっても、アプリケーションソフトウェアの開発においては、個別のユーザの個別のニーズを反映するカストマイゼーションの実施機構が重要となる。

そのような機構の整備には多額の研究資金の投入が必要であり、その資金の回収においては個別のソフトウェアの有償配布が必要になる。さらに、その機構によって提供されるソフトウェアや導入・保守サービス等を排他的な知的所有権で保護し、持続的な成長を支援することが市場経済の原則に適合する。

このような現実的制約にもかかわらず、アプリケーションソフトウェアの OSS 化は、社会的な意義が高い。特に、アプリケーションソフトウェアの中でも、より一般性の高い、オフィスワーク用ソフトウェアスイートや、情報共有のためのソフトウェア類、情報セキュリティ保護のためのソフトウェア類等では、適用分野が広く、かつシーズとなる技術が成熟化しつつある。

また、多くのユーザによる利用が期待できることから、開発者グループが多くのユーザグループからのフィードバックを受け、ユーザとの協力を基盤としたプロセスによって当該ソフトウェアを成熟化させるバザールモデルが有効に働く[lvii]。それによって、ソフトウェ

アの調達コストを劇的に低減させることも可能となる。

一般的に類似の機能を提供する同種類のソフトウェアは、特定のソフトウェア技術者(集団)が、ユーザの要求や希望に基づいて、実現すべき機能を機能仕様書にまとめあげ、それを実現するソフトウェアを開発することから始まる。その開発の過程で、ソフトウェア技術者たちはいくつかの重要な実現のためのアイデアを提案し、そのアイデアに基づいてソフトウェアを実現してゆく。

一旦、そのようなソフトウェアが発表され、市場に供給されると、類似の機能を実現した別のソフトウェアを、別のソフトウェア技術者集団も開発できる。この時、新しく開発に取り組む技術者たちは、より新しいアイデアを盛り込むことで、より品質の良いソフトウェアを開発しようとする。このような複数の開発者集団の競争によって、それらのソフトウェアは少しずつ進化してゆき、最終的には機能そのものは標準化されてゆく。

ソフトウェアは、機能に関する標準が決まると、その実現方法も固定化するので、実現技術も少しずつ標準化されるようになる。そのようにして標準化された実現技術は、時として大学等の専門教育で教えられるようになる。このソフトウェアとして成熟化した段階に至ると、類似のソフトウェアは、大学の学生にも実現可能なものとなり、必ずしも利益を追求する企業の活動として開発する理由も薄れてゆく。

このようにして、企業活動で実施してゆく意味の薄れたソフトウェアの中には、多くのボランティア技術者達が参加して開発を行うオープンソースソフトウェアが適しているものも存在する。特に、多くの一般ユーザが必要とするオペレーティングシステムなどは、OSS としての配布が適したソフトウェアと言える。

それは、その信頼性の要件や、情報セキュリティ保護機能に対する要件が厳しく、ソフトウェアに誤りを残しておくことには問題があるからである。そのため、多くの人々がこのソフトウェアの開発に関係し、そのソースコードを点検することで、信頼性や情報セキュリティ機能の実現度を向上させてゆくことが必要だからである。

また、多くのユーザがそれを必要とすることから、限りなく安価に、それを提供することが望ましい。ソフトウェアの獲得に一定以上のコストが必要になると、それを獲得できる人々が限定され、相対的に裕福な人々だけがそのメリットを享受するようになる。

そのような社会的な背景もあって、ソフトウェアのソースコードを公開し、誰でもが入手でき、誰もが修正可能にすることは、ソフトウェアの健全な進歩(進化)の必要条件であると言える。さらにそのようにして開発され、維持されているソフトウェアは、それを利用する全ての人々とそれを開発する全ての人々が、そのソフトウェアの問題点を共有し、その解決について相互に協力してゆく必要がある。そのための制度として、オープンソースソフトウェアでは、ライセンスの概念を規定し、それに基づく利用や修正、そして配布を実施してゆく。

このライセンスの考え方は、独占的な知的所有権の保護を目的とした従来の著作権や特許権とは全く異なり、ソフトウェアそのもの(ソースコードやそこから生成されるオブジェ

クトコード)と、その実現に応用されたアルゴリズムなどの基本的なアイデアを、人類が共有すべき公共の財産とすることを前提とする。このことは、かつて米国の法律家たちが主張していた、重要なアイデアは公知にすることが重要であり、特定個人や特定組織が独占的な知的財産としてそれに関する全ての権利を主張すべきものではないとした思想に近いものであるとも言える。

ソフトウェアには、企業間での経済競争を有利に実施してゆくために、特定の組織が自らの利益の追求のために、自らの投資で開発し、独占的に利用することで、競争に打ち勝つための手段としてのものがある。その一方に、そのような企業間競争には関係なく、地球上のすべての人間がコンピュータを有効に利用して、人々の様々な活動を効果的に実施するための手段として利用することが必要な、万人のためのソフトウェアもある。

後者のソフトウェアは、高い品質のものを低コストで提供することが、社会全体の進歩のために必要である。そのようなソフトウェアの開発と保守、そして配布のための仕組みとして、オープンソースソフトウェアの考え方は、これからの社会に極めて重要である。

第 2 節　ソフトウェア技術開発に関する産業施策[33]

ソフトウェアは、資本主義社会が新しく手にした無形財である。また、それは一人の天才によって作成されるものではなく、数多くの専門技術者の参画によって実現される資本集約的であり、労働集約的な側面をもったものである。さらに、それはコピーによる複製が、他の工業製品に比較してはるかに容易な性質を保持している。

そのため、その知的財産権を保護することには困難が伴う。しかし、多くの人々が必要とするソフトウェアは、その開発に多額の資金の投入が必要だとしても、開発に成功した企業にとっては巨大な富を生む無形財である。

このことは、国家の視点から見ても、重要な産業であるとの位置づけとなる傾向があることを示唆している。特に、ソフトウェアの開発に必要な高度な知識をもった技術者を数多く教育できる教育システムが整備されている国々においては、原材料を必要とせずに開発・生産が可能であることから、関連産業が育成されていなくても発展が可能な特殊な産業であると言える。このことから、開発途上国であっても、人的資源が豊富でさえあれば、すぐに着手可能な開発・生産であり、育成の容易な産業であると言える。

実際に 1990 年以降、特にインドや中国においては、それらの国々おける豊富な人的資源に着目し、ソフトウェア技術者の育成を目的とした高等教育機関が設立され、海外の先進諸国から、最先端の技術が導入され、高度な知識をもつ人材の育成が国家戦略として重点的に実施されてきた。これ以前にも、シンガポールにおいては、世界中から優秀な人材を集め、プロジェクトを組んで高度なソフトウェアを開発しようと言う試みが行われていた。

日本においても、1980 年代に実施された「第 5 世代コンピュータプロジェクト」[lviii]や「シグマプロジェクト」など、新しい時代のプログラミング言語である Prolog を活用した人工知能技術の応用や、新しいソフトウェア開発環境として柔軟なオペレーティングシステムである Unix と通信回線を活用した分散プログラム開発環境などをテーマとした国家プロジェクトなどが実施された。

ただし、これらの国家プロジェクトでは、日本の国家予算の利用が、ハードウェアを主体とした古い枠組みであった。そのため、どちらのプロジェクトも、ハードウェア開発に焦点を合わせた国家プロジェクトの形式を踏襲したものになっていた。

その後、日本においては 1990 年代に入って、インターネットを活用したソフトウェア開発を推進することを目的とした、ソフトウェア CALS プロジェクトが、他の産業におけるインターネットの活用に着目した CALS プロジェクトの一部として実施された[lix]。この CALS プロジェクトでは、プラント開発などの効率化を目指したシステム CALS、自動車開発の効率化を目指した自動車 CALS、建設工事などの効率化を目指した建設 CALS、ソフトウェアの受託開発などの効率化を目指したソフトウェア CALS など、産業分野別のプロジェクトが組まれていた[lx]。

---

[33]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.145-151 参照

さらに 2000 年代に入ると、それまでは製品として売られることが一般的であったソフトウェアを、普及したインターネット回線を利用して低コストで提供するオープンソースソフトウェアが出現した。これは、ソフトウェアを開発し、それを個人に販売することで、莫大な利益を得る企業が出現し、そのような高価なソフトウェアを購入できない階層の人々と、購入が可能な階層の人々との分断を進めたことを良しとしない人々のフリーソフトウェア運動が契機となって、無償で汎用ソフトウェアを配布することを目的として始められた流れの中で、誕生したものである。

オープンソースソフトウェアは、従来のソフトウェアのように営利組織である企業によって開発されるのではなく、一般的には技術者が個人個人として参加している開発者コミュニティによって計画され、開発される。この開発者コミュニティに参画する技術者達は、個人として参画し、給与等の報酬を受け取らない。

従って、ソフトウェアの開発において最も資金が必要となる人的資源に、ほとんど資金を投入する必要がない。技術者達は、大学や企業に所属する人間で、給与は所属する組織から得られている。彼らは、彼らの自由な時間を使って、自分が担当している活動(プログラムの作成など)を実施するのである。

このようにして、参加する技術者達の自由意志で開発されたソフトウェアは、時として巨大な資金を投入して、企業によって開発されるソフトウェアよりも、ユーザにとって質の良いものであることがある。これは、開発者達が納得して開発したプログラムであるからと言える。

彼らが、企業内で開発するソフトウェアは、企業の存在目的や、経営目的の制約の中で、それらに合う製品として開発されなければならない。このことは、技術者個人の価値観から言えば、ソフトウェアの質を落としても、企業の目的に合うものを開発しなければならないことを強いる結果になることがあるからでもある。

上述した理由により、ソフトウェア技術者として最良と思える設計や実現方式を応用して、ソフトウェアを開発したいと考える技術者は少なくない。そのような技術者の技術者としての信念に基づく、率直な感情から、たとえ無償であっても自分が最良と思える設計や技術でソフトウェアを実現できる新しい仕掛けが、オープンソースソフトウェアである。

また、オープンソースソフトウェアでは、開発したソフトウェアのソースコード(開発者が記述した可読可能なプログラム記述)を公開し、誰でもそれを変更して改善することを可能としているライセンス規定がある。このような新しい仕掛けによって、ソフトウェアの開発者は、自分が当初考えた設計や実現方法よりも、より良い設計や実現方法があったことを知ることができ、自分の技術を磨くことも可能である。

以上のような理由から、オープンソースソフトウェアは、時として企業が巨大な資金を投入して開発し、大々的に販売する市販のソフトウェアよりも高い品質のソフトウェアであると評価される例は、少なくない。例えば、オペレーティングシステムの一つと

して知られている Linux や、インターネットを活用してホームページを閲覧するためのプラウザとして知られている FireFox などは、オープンソースソフトウェアである。特に、FireFox は、オープンソースソフトウェアとして開発され、一般に配布されている Mozilla と名付けられたプラウザを基にして開発されている。オープンソースソフトウェアは、このようなことから、ソフトウェア産業のあり方を大きく変えるきっかけになる可能性がある。

第 3 節　資本主義社会における市場競争と社会規制[34]

資本主義において、企業は自由市場における競争を通して、他の企業と利益、売上げ、市場占有率、社員数、資本規模、知的財産の蓄積、など様々な指標において競い合っている。競争に勝つためには、企業は手段を選ばないが、選択しうる手段においては、自ずから社会的な通念に基づいた制約がかかっている。その制約には、慣習、標準規格、法制度、企業のポリシーなどがある。

近年の市場における競争を原則とした経済システムにおいては、個人や個々のグループは、自分や自分たちが市場で勝利者として生き残ることを第一の命題としている。その市場における競争において重要なことは、製品としてのソフトウェアに限らず、消費者や利用者から評価される意味での、高い効用を提供できる品質の良い製品を開発、提供するだけでなく、製品に問題が発見された場合には、その問題を速やかに解決し、利用者・消費者における効用を維持するための保守サービスを提供できることである。

そのような視点で第一義的に重要なことは、消費者・利用者に提供した製品に対する顧客満足度を高めることである。その意味では、リバタリアニズム的な品質論の立場に立って、良い品質の製品を企画し、提供することが望まれる。このことは、市販品でない製品であるオープンソースソフトウェアにおいても、重要な問題であることには変わりない。

さらに、消費者・利用者がその製品を長期に利用する場合、時間経過とともに様々な問題が表面化する。そのような問題に対して、適切な対応を行い、問題を解決しながら製品の保守を継続的に実施できる、よい設計の製品であることが必要となる。これは、品質特性で言えばソフトウェアの場合、保守性と移植性が重要になることを意味する。すなわち、e タイプのソフトウェアである。

保守性は、製品に問題が発見された時、その問題の解決法を容易に特定でき、その解決方法を実際に適用でき、さらにそのような保守作業によって新たな問題が発生しにくい設計になっていることを言う。移植性とは、ソフトウェアの稼働環境であるコンピュータハードウェアに変更があったとしても、大きなソフトウェアの変更なしに容易に新しい稼働環境に適応させられる設計になっていることを言う。

この 2 つの品質特性が重要になるソフトウェアは、一般的に e タイプのソフトウェアと言われている。e タイプのソフトウェアは、時間の経過とともに機能仕様の内容が変化するソフトウェアを言う。このような製品は、従来型のハードウェアを基本として物理的なシステムとして設計され、生産されるこれまでのシステムにはなかった製品である。

そのような市場での競争に勝てる製品を開発するためには、その開発に参画する技術者は、単に機能の良いソフトウェアや処理効率の良いソフトウェア、残存欠陥が少

[34]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.117-125 参照

なく故障頻度の低いソフトウェア、利用者にとって使い易いソフトウェアを設計し実現することにのみ努力を集中するだけでなく、高い保守性や移植性を達成する設計の実現を目標として、開発活動に従事しなければならない。

特に、そのような競争の激しい市場に投入する製品の開発に参画している技術者においては、単に自分の責任で実施した設計の部分に対して細心の注意を払うだけでなく、周囲の技術者が担当して実施した設計に対しても、第三者の専門家の視点から、同僚技術者が実施した設計の内容が、専門家として妥当なものであることを検証し、証言するピアレビューにも細心の注意を払うことを心掛けなければならない。

このようなソフトウェア技術者の行動に関する規律や、それを順守しなければならないと考える倫理観は、法的に強制されているものでもなれば、国際的な認証規格のような半強制的なものでもない。自由な市場における競争と言えども、そのような行動規範を守ることがその時代の前提条件とされている。このような行動規範を守ることのできない技術者を多く雇用している組織は、長期にわたる市場競争の過程で淘汰されてゆくはずである。

第 4 節　企業間競争と国家の役割～立法・司法・行政の役割

現代国家においては、企業間や個人間の競争が、社会の慣習や通念上の制約を逸脱しないように、また、逸脱した場合には不当な行為を止めさせるための機構が整備されている。ここでは、そのような視点から立法、司法、行政における役割について述べる。

三権の分立を前提とする近代国家においては、国家の立法権を行使する制度として国会がある。これは、多くの国々が間接民主主義のシステムを取り入れているからである。この制度は、17 世紀のイギリスやオランダで発祥したものである。情報通信技術が未成熟で、その基盤も整備されていなかった当時、国民全員の意見を集約する直接民主主義のシステムは、非現実的であった。

間接民主主義システムを採用する国家においては、国会が立法を担当する。ただし、イギリスやアメリカ合衆国の場合、法律は判例主義を取っているので、国会における新法制定活動は、どちらかと言えば現実の後追いになる傾向が強い。これに対して、ドイツやフランスなど、ヨーロッパ大陸諸国においては、大陸法と呼ばれる、法に規定されていない行為や活動は、基本的に合法的な行為や活動とみなされる。したがって、国会における新法制定活動は、現実の先取りが前提となる。

判例法を採っている国であれ、大陸法を採っている国であれ、社会的な問題を生じる可能性のある行為や活動に対して、それが社会の秩序維持を阻害する可能性があると認識されれば、そのような行為や活動は、社会的規制の対象になり、そのための法律が準備される。つまり、資本主義諸国においては、企業間や個人間の競争において、社会通念から見て公正でないと認識される行為や活動が考えられるとき、そのような行為や活動を禁止する法律を制定することが必要となる。

例えば、製造業において、実際に製品を生産する工場では、様々な材料が利用される。また、そのような材料を加工するための高度な処理機械が導入される。また、そのようなプロセスから生産された製品の部品の中には、長期的な視点から見ると人々の健康を阻害する効果をもつ部品もありうる。このため、近年、各国において工場の生産工程で排出される廃棄物処理等に関して、周辺住民の健康への影響を実質的になくすための工業廃棄物処理等に関する法律が整備されている。

このような法律に対する順守義務を企業に課すことは、工場等における生産工程の変更、特に材料の変更や生産機械の変更を義務付ける結果となり、生産コストを押し上げる結果となる。生産コストが上昇すれば、販売価格を上げるか、利益を下げるかのどちらかの対応を、企業は採らざるを得ない。いずれにしろ、企業の競争力を低減させる。

特に、グローバル経済における競争では、法規制の異なる 2 つ以上の国々にある工場で生産が実施されるとき、他の条件が全く同じであっても、法的規制の違いによって、製品の生産コストに差が生じる。このことは、法規制の強い国から、法規制の弱い国への工場の移転が生じる可能性を意味する。

国家にとって、その国の産業を守り、発展させることは、国家経済発展のために重要で

ある。したがって、国内の企業に過度に厳しい規制を強いることは、産業政策上問題が多い。しかしながら、工場周辺に住む住民の健康に対する影響を考えると、住民に被害が及ぶ水準の環境汚染は、国民の生命と財産を守らなければならない国家として、看過することはできない。このことから、国会としては、産業競争力と国民の健康被害への影響のバランスを考慮した法制化を考えざるを得ない。

この事例でも問題になるように、国家としてどのような法整備をしなければならないかについては、グローバル経済に対する規制の影響を考えざるを得ない。例えば、自国の企業や工場に対して、福祉的な視点で理想的な法制度の制度化を目指した場合、短期的には国内の企業に対する負担を過重にする。

しかし、それが生産した製品そのものにも関係する規制である場合、その法律の施行によって、国内企業に課される負担も大きくなる。その一方で、将来、その国の市場に参入する他国企業にとっては、市場への非関税参入障壁となるため、間接的に自国の作業を保護する効果がある。

近年、ドイツで導入された自動車部品のリサイクルに関する法律は、自動車の生産に利用された部品の材料をリサイクルし、不要な材料の消費を低減する目的で制定されたものである。この法規制のため、各自動車メーカは、部品や自動車の設計を大きく変更する必要が発生し、一時的には多大なコスト増を招いた。

しかしながら、この法律自身は、ドイツ市場に参入する全ての自動車メーカに対する規制であり、結果として早期に新しい法規制に対応したドイツ車メーカは、市場の競争で優位に立った。最近の EU における自動車の二酸化炭素排出量規制は、類似の傾向をもった法規制である。

このように、各国の法律は、自国で財やサービスを供給する企業を主たる対象とするものであるが、経済がグローバル化した今日、それは自国市場に参入しようとする全ての国の企業に対しても適用されるため、戦略的に導入されると、長期的な視点から見れば自国産業の保護につながる性質を持つ。

近代国家においては、特定の個人または法人が行った行為や行動が、その国の法律にのっとったものであり、その国の一般通念上許される範囲のものであるかどうかは、最終的に司法の判断となる。2者間の問題で係争が発生した場合、その決着は2者間の協議で行い、合意にいたならければ片方が他方に対して報復的な対応に出ることは、法治国家の秩序を維持する方法として許されるものではない。

グローバル経済が進み、市場における企業間競争がし烈化すると、2つ以上の国々を本拠とする企業間で問題が生じる可能性がある。そのような問題の典型的な例が、調達または獲得段階における業者選定の問題である。

これは、ある国の組織(企業や公益的な法人)が、特定の目的で財またはサービスの購入を目的として、その仕様を公開し、供給企業からの応募を募集する。その募集には、当該国の企業だけでなく、第三国の企業からも応募する可能性がある。この時、調達実施者が、

自国企業を優先して業者の選定を実施することは、一般的に公正な競争の視点から問題があるとされる[lxi]。

特に入札案件の規模が大きい場合、第三国の企業は、調達を実施した組織が本拠を置く国の政府と自国政府の両者が世界貿易機構に加盟している場合、自国政府を通して世界貿易機構(WTO)に提訴することもできる。これは、他国の企業を入札において不当に差別したことが理由となる。

世界貿易機構が設立される以前、日本の大手通信会社の調達が、基本的に日本国内の数社に限定されていたことを、米国政府が問題にし、日米間の政治問題になっていた時期があった。このような、商取引の実践が、ある国の中では慣習上許される範囲のものであっても、文化的背景の異なる他国の人々の目から見ると、許容できる範囲を逸脱している例は、決して少なくない。

上述したことからも明らかなように、商取引の実践が、ある国の中では慣習上許される範囲のものであっても、文化的背景の異なる他国の人々の目から見ると、許容できる範囲を逸脱している例は、決して少なくない。この場合、当事者はどの国の裁判所に提訴するのが妥当と言えるのであろうか。基本的には、訴訟を提起する企業や個人が本拠を置く国の裁判所へ訴えることが、訴えを起こす当事者にとっては有利である。

逆に、訴えられた当事者が本拠を置く国の裁判所で裁判をすれば、訴えられた側が有利になる。本来であれば、提訴する場所によって、判断が変わってはならない。しかしながら、社会的通念の違い、法律の違い、過去の判例の違いから、訴訟が行われる場所と最終的な判決の間には、強い相関がみられる。

その意味で、各国の司法担当者、特に裁判官は、自国の法律と判例だけを学ぶのではなく、他国における法制度や判例を学び、グローバルな視点から判決を出すことが重要になる。このことを支援するため、グローバルな案件だけを取り扱う裁判所と専門裁判官の養成や、裁判官に助言を与える専門家の認定と育成の制度などが重要になる。さらに、米国における商取引上の係争を解決する機構としての商工会議所の役割を見習い、日本の制度を見直すことなども、重要な対応策である。

近代国家の行政機関である各国政府は、自国の発展と国民を守ることを目的として、様々な公的サービスを提供する。その中には、政府の機能を維持するために外部から様々な財やサービスを購入する。また、自国産業の国際競争力を向上させるための支援の計画と実施、自国内の社会基盤の整備と維持の計画と実施、自国民の将来の労働力としての国際競争力強化のための教育訓練の計画と実施、自国内の市場における公正な取引と競争環境維持のための施策の計画と実施などもある。

ここでは、特にソフトウェア開発に関係した問題として、知的財産権の保護、ベンチャー企業の育成、高等教育と雇用制度の改革について議論する。知的財産権については、現在、ソフトウェアの知的財産権は、著作権、特許権、オープンソースソフトウェアのライセンスによって守ることが、制度として実質的に確立している。既に議論したように、著

作権についても、特許権についても、その発生時点での状況と、ソフトウェアが出現してからの状況が違い過ぎるため、重大な問題をはらんでいる。

とはいえ、これらの制度によって権利を守られる技術者においては、共通の倫理観を確立し、それを教育プログラムに組込むことによって、問題の発生を予防または減少させることができる。このような対応が望ましいかどうかは別として、どのような時代の、どのような社会であっても、法的な制度だけで社会的秩序を維持することは不可能であった。したがって、グローバルに技術者達が持つべき倫理観を確立し、教育しなければならないことは、絶対的な必要条件である。

例えば、ソフトウェアに限らず、「他人のアイデアを真似しない」と言うイギリス的倫理観は、著作権や特許権の基本でもあり、知的財産権の保護において、基本的な役割を担っている[lxii]。しかし、特に、東洋の国々には、そのような倫理観は一般的でなく、コピーは人々の権利であるかのように認識されてきた。日本においても、数十年前まで、製品、設計、仕様、デザイン等のコピーは、技術者達が抵抗感なく実施していた。

その意味で、「他人と同じことをしない」ように生きると言うイギリス的価値観は、ある意味、これからの技術者達にとっては、基本的な姿勢としなくてはならない。ただし、「同じことを別の方法でする」ことは、コピーとは言われない。この方法の改良に独自のアイデアが注入される可能性が大きいからである。行政としては、そのような幼少期からの教育の問題として、知的財産権を考えることも重要である。

次に、産業政策としてのベンチャー企業育成政策を議論する。新しい製品や新しいサービスは、新しい企業によって生み出される例は多い。最近では、米国企業のフェースブックが提供しているサービスがその例である。また、少し古くなるが、インターネット書店のアマゾンや、インターネットオークションのイー・ベイなどもそのような例である。さらに古くは、アップルコンピュータやマイクロソフトもベンチャー企業として生まれた。

それに対して日本では、この 20 年間、ベンチャー企業がほとんど誕生していない。実は、誕生してはいるものの、育っていないのである。インターネットサービスを提供する IIJ は、1990 年代初頭にベンチャー企業として発足したが、現在は NTT の資本参加を受けており、ベンチャーとは言えない形態になっている。イー・モバイルもソフトバンク傘下に入り、同様の道を歩んでいる。

日本の最近の傾向に、ベンチャー企業の特性を活用していない事業に参入する例が少なくない。このような事業では、本来的に既存の大企業が有利であり、長期的な競争に勝ち残るのは困難である。もう一つ、日本のベンチャー企業に共通して見られる傾向に、特にインターネット系のサービスでは、米国で開発されたサービスの日本版サービスを提供する楽天市場のような形態も多い。すなわち、独自色に乏しいものが多い。

スマートフォンの機能を活用したラインのサービス提供などは、日本独自のものと言えるが、インターネットを利用した電話機能・個人間通信機能と考えれば、米国に類似のものが存在しなかったわけではない。この場合も、やはり独自性に乏しいと言わざるを得な

い。フェースブックの独自性は、本来匿名性の高い、公的な情報交換の道具であるインターネットに、既知の人に限定したクローズドな情報交換網を仮想的に作り出す機能を提供している点にある。

このような独創性の高いサービスや製品を供給しようとする新しいベンチャー企業が、相当数生まれてこなければ、日本の産業は長期的に衰退する。大企業の新製品・新サービス開発のスピードは、ベンチャー企業のそれに比較にならないくらい遅い。この時間的競争力の差が、これからの時代のグローバル経済においては、極めて重要な競争力の源泉となる。

各国の労働行政は、過重な労働からの労働者の健康の保護、極端に低賃金での雇用と若年労働者の雇用に対する規制、非正規社員と外国人労働者の雇用に関する指導と規制などについて、法制度の整備と法の順守を徹底するための施策を担当している。日本の労働行政において現時点で最も重大な問題は、雇用制度としての終身雇用、年功序列、新入社員一括採用が相互に絡み合い、同一労働同一賃金への意向が困難な状況をどう打開するかである[lxiii]。

経済のグローバル化が進む中で、先進諸国の企業は、生産性の向上、技術革新の進展、提供する製品やサービスの品質を高める努力などの継続を要求されている。これらのことは、それらの企業で働く人材による知的な労働の成果として達成される性質のものである。したがって、グローバルな競争に勝つためには、各企業は、可能な限り良質な人材を多く雇用することが重要になる。

このことは、国家のレベルで考えれば、そのような企業のニーズに適合する質の人材を、産業が必要とする量だけ供給できるかどうかが問題となる[lxiv]。ここに、教育行政が重要になる一つの理由がある。さらに、労働力の最適配分を行うため、人材を必要とする企業と、その企業のニーズに適合した人材のマッチングを効果的に実施するサービスが必要となる。そのような人材斡旋サービスは、日本においては厚生労働省の認可事業のひとつとなる。

大学等の高等教育機関においても、卒業生を社会人として企業に送り出すため、人材斡旋サービスを実施している。しかし、日本の大学は本来教育機関であるため、人材の斡旋と言う意味では、人材を募集している企業からの求人票を学生に公開し、就職試験を受けるように調整するだけである。就職環境が悪化している今日においては、大学によってはこの就職関連サービスを充実させ、効果的なマッチングを実施している大学も出始めている。

とは言え、新入社員一括採用と終身雇用のシステムが機能しているため、求人が極端に少ない年の卒業生は、採用試験に合格せず、結果的に非正規雇用の仕事に就かなければならない状況に陥ったりする。逆に、社会的に好景気で企業からの求人が多い年の卒業生には、複数の企業の採用試験に合格する学生が発生する。しかし、余剰な労働力が存在しても、その年以前に大学を卒業している人材に対しては、基本的に門戸は閉じられている。

この不健全で非効率なシステムによって、日本の労働市場は、産業が必要としている人

材に余剰があるにも関わらず、企業側では人材不足現象が発生している。つまり、良い人材が、活用できていない現実がある。この硬直化した労働市場の問題を解決するためには、年功賃金体系の雇用制度から、同一労働同一賃金の雇用制度に移行し、正規雇用の社員であれ、非正規雇用の人材であれ、同じ労働に従事する人材の給与を同一にすることが必要である。

このことは、終身雇用で年功序列的な雇用制度を維持したままでは不可能である。終身雇用で、企業側の都合で従業員が従事する仕事が変わった場合、当該の従業員の給与額が変化し、生活が不安定になるからである。年功賃金体系は、このような問題を解決する方法としても機能してきたわけである。

ただし、そのような前提が成立するのは、従業員が従事する仕事の知識集約度が低く、特別な専門知識を必要としない場合である。経済が著しく進展した今日、知識集約労働のほとんどは専門的知識を必要とするため、特別な教育が必要となる。

最後に、高等教育であるが、日本の高等教育は後述するように、産業界で活躍する専門家人材を育成することを目的としていない。むしろ、国家的教育システムとして閉じたシステムになっている。

このことは、外部からのフィードバックや、外界の変化に対応しなくてもある程度の期間は生き残れるようになっていることを意味する。この産業の現場と高等教育の断絶が、今後の日本の発展に大きな影響をもたらす。

前述したように米国社会においては、アクレディテーション制度によって、大学における教育プログラムと産業界における仕事が密接に関係している。このため、大学は産業界における人材の要請に適合した人材の育成を心掛けなければならない。これに失敗すれば、アクレディテーション認証を受けられず、卒業生たちは専門家として職に就けなくなるからである。

日本においても、そのようなアクレディテーション制度を導入しようとする動きがあり、すでに認証制度は確立している。しかし、人材を受け入れる日本企業の側にその準備が不足しており、認証取得が学生の職の選択に影響を与えることはない。このことは、多大な労力を投入して認証取得の準備をする大学側のやる気をそぐ結果となっている。

日本の高等教育のもう一つの問題点は、大学が授与する学位が、学位を授与された人材が従事できる仕事とほとんど関係がないことがある。例えば、教員としての職に就く場合、大学教育で学んだ専門と、教職課程で学んだ知識によって、特別に教員免許状が与えられる。医学関係の職に就く場合にも、厚生労働省が認めた高等教育機関が授与した学位記または卒業証書と、別途、厚生労働省が課す国家試験に合格することが求められる。

法曹関係の職に就く場合、公認会計士や税理士そして弁理士の職に就く場合、建築関係の職に就く場合、無線機を取り扱う職に就く場合などには、国家試験に合格することが前提条件になる。それ以外の職業については、内容は専門的であっても、国家試験のような資格試験、認可された大学等の高等教育機関での教育歴が前提となることはない。そのよ

うな一般的な職業に、ソフトウェア技術者が従事するソフトウェア開発や保守がある。

既に議論したように、米国の社会では、ソフトウェア技術者として仕事に従事する時、国家試験のような資格は前提とされないが、アクレディテーション認証を受けた大学の学部学科を卒業し、学位が授与されていることが前提となる。コンピュータ科学科の卒業生と言っても、アクレディテーション認証を受けていない大学の卒業生の場合、ソフトウェウ技術者としての仕事に就くことはできない。

これまでも議論してきたように、今日の製品開発においてソフトウェアの担う役割は重大である。これまでの製品開発において、機械の技術者や電気電子系の技術者が担ってきた役割以上のものを、ソフトウェア技術者は担っているのである。自動車が停止しなくなる原因は、過去においては高温になって沸騰したオイルの問題、摩耗したブレーキパッドやロータの問題、オイルが漏れているパイプなど、単純な原因のみであった。

今日の自動車では、衝突防止ブレーキが付加されていない場合であっても、ブレーキペダルセンサの不具合、マイクロプロセッサの誤動作、組込みソフトウェアに隠れた誤り、ブレーキ作動制御機構の故障や不具合など、これまでの自動車に比較すると著しく複雑化している。しかし、製造現場では、これまでのような複雑なオイルパイプの取り付けなどが無くなり、製造過程での作業は大幅に簡素化されている。

自動車のブレーキの例で言えば、ブレーキシステムが正常に作動しなければ、自動車は停止しない。したがって、それが緊急時であれば、人命が失われる可能性は高い。この例のように、組込みソフトウェアを設計し、実現する作業自体はそれほど高度な知識を必要としないが、実現したシステムがいかなる状況でも適切に作動し、事故に至らないことを保証するためには、高度に専門的な知識の教育が必要になる。そのような、専門的人材の質を保証するシステムの確立が重要である。

日本の高等教育機関で実施されている教育の内容と産業界での実践的な仕事の乖離は、非常に大きい。それは、大学の教員の多くが産業界での現実の仕事を知らないことに帰する部分が大きい。

ただし、問題はそれだけではない。幼少時の教育から高校卒業までの日本の教育は、より有名な大学の入学試験に合格することを目的とした教育である。ここで重要なことは、大学が実施する筆記試験や大学入試センター試験で、可能な限り高い点を取ることである。

大学入試センター試験の場合、マークシート方式の選択問題である。したがって、選択肢の中に必ず 1 つだけの正解がある。それを効率よく見つけて、解答用紙の適切な欄にマークすることが重要である。

このような標準化された問題の場合、正解を出すための知識が無い場合でも、正しい選択肢を選ぶことができる。そのような技能を学ぶことによって、現実に高得点を得ることは可能である。

問題は、そのような試験で高得点を得た高校生を、得点順に並べ、上位から合格者を機械的に選ぶ現在の大学入試制度は、専門家人材の育成を目的とした大学において、入学さ

せる高校生の選抜方式として適切な方法とは言えないことである。この方法は、誰がこの処理を実施しても結果が変わらないと言う意味において、公平性が保たれていると言える。しかし、それが真の意味での公平性と言えるかどうかの保証はない。

女子学生の入学希望者が少ない大学では、女子生徒の受験者を優先して合格させることが、長期的な視点から大学として採るべき道である可能性は高い。また、障害をもつ生徒の入学がない大学では、そのような受験生の入学を推進することが重要かもしれない。同様なことは、外国人受験生についても言える。つまり、何が公平かは人間が決めることであり、機械的に決めることではない。

米国の大学では、入学事務室において、入学希望者の標準学力テスト結果を参照して、要求学力レベル以上の学生を選び出し、それらの学生から提出された書類、特に課題小論文などを精査し、人種、性別、親の所得階層、国籍などを総合的に評価して、入学者を選定する。この過程で投入される労力は大変なものである。このため、標準学力テストの結果が、入学した学生よりも高かった受験者が不合格になる例は多い。小論文の評価も重要であるが、それ以外の要素で合否が変わることもある。

このことは、大学が何のために存在するかを突き詰めて定義した大学のポリシーと、入学者選抜の方針として公開されているアドミッションポリシーに明確に述べられている。多様な人々が集まり、議論しあうことで、科学技術の発展と社会の発展に貢献できる人材を育成しようとする大学では、本質的に多様な学生を必要としている。

このため、適切な人種、性別、親の所得階層、国籍などのバランスを達成すべく入学者の選定が行われるからである。日本においても、画一的な入学試験制度を脱して、専門的な人材の育成に適した学生を選ぶ入試制度の確立が重要である。

以上のような教育改革を実施するためには、日本の高等教育が産業界の現場と直結し、大学と企業間における人材の流れが双方向になると同時に、企業の雇用制度を変革し、専門家人材の採用と育成のシステムを改革しなければ、大学の教育プログラムも変わることはない。ソフトウェア技術者に限定して言えば、企業がソフトウェア技術者の採用基準として、情報処理技術者試験の基本情報処理技術者の試験に合格していることを条件にするだけでも、大学の専門家教育プログラムは大きく変化する。

それは、大学の卒業生の就職に大きく影響を与えるからである。現段階では、大学生受験者の合格率が約 25 パーセントであるが、数年後には 70 パーセントを超えるようになるであろう[lxv]。このために、各大学はソフトウェア技術者としての就職活動を考えている学生に対して、入社試験の合格率を上げるような特別な努力をするはずである。

ただし、ソフトウェア技術者として働くためには、基本情報処理技術者の試験に合格しているだけでは、知識として不十分であり、より高度な試験への合格を課すことも重要であろう。

第9章　提言: 高等教育の役割とソフトウェア技術者の育成

第１節　我が国の近代化と高等教育の果たした役割

明治以降、1880 年頃から 1970 年代末までの日本の高等教育の在り方と、日本の近代化に与えた貢献ついて、歴史的に振り返ってみる。日本の高等教育は、明治政府が欧米と類似の人材育成システムを日本にも確立することを目的として、文部省を中心に取り組んできたものである。とは言え、初期においては、政府の各部門がそれぞれのニーズを充足させる目的で、様々な専門家を育成する教育機関を設立した。

大学について言えば、日本に最初に設立された大学は、東京大学であり、かなりの期間にわたり、東京大学は日本の唯一の大学として存在した[66]。設立当初の東京大学には、医学部、法学部、工学部があった。それらの中には、旧江戸幕府が設立した高等教育機関を改変して設立したものもあった。3 つの学部は、明治政府の必要性に応じて設立されたと言える。

医学部は、江戸時代には特に公的な資格を必要としなかった医師の養成を、国家的に実施して、不足していた医師を育成する必要があったからである。法学部は、行政と司法を分割し、行政から独立した裁判所を設立し、裁判をする必要があったからである。また、明治政府において、必要な法律を制定するために法律の知識を有した人材も必要であった。その一部は、裁判官、検事となり、また官僚として政府につかえた。工学部は、日本が近代国家の道を歩むため、工業の発展が重要であったことによる。工学部の人材も、その多くが官僚として政府で働いた。

東京大学で学生の教育に当たる教員の多くは、当初、諸外国から採用された比較的若い教員であった。それらの教員の中には、産業革命を終えたばかりのイギリスで、新しい教育方法と新しいカリキュラムを学んだ工学部の教員もいた。これらの人材が、新しい東京大学のカリキュラムをつくり、それに基づいた学生の教育に従事した。

東京大学を卒業した学生の中には、当時の先進諸国へ留学し、最先端の知識を学び、帰国して東京大学の教員に就任する者もいた。このようにして、時間とともに、東京大学の卒業生が、東京大学の教員に就任すると言う人材育成システムが確立した。東京大学以外の官立の高等教育機関を卒業し、その後、先進諸国へ留学して成果を出し、東京大学の教員に就任する者もあった。

文部省が帝国大学令を制定し、東京大学は東京帝国大学となった。さらに、第 2 の帝国大学として京都帝国大学が設立された。その後、東北、九州、と言うように、7 つの帝国大学が次々と設立され、専門家の育成を行うようになった。

この頃、高等専門学校制度が作られ、工業系高等専門学校、医学系高等専門学校、商業系高等専門学校など、産業化社会の形成期にあった日本の発展を支えた人材の育成を担う高等教育システムが確立されていった。この頃から、私立大学の設立が認められるようになった。

この時期の日本の高等教育には、高等学校を経て専門を 4 年制の大学で学ぶ道と、高等専門学校を経て大学へ入学する道があった。特に、高等学校で一般的な基礎科目を学び、大学に進む道が標準的な道として推奨されていた。後に、この高校教育の 2 年間と大学教育の 4 年間を統合して、4 年間の大学教育プログラムが作り出される。

この時、大学の最初の 2 年間を教養教育としたのは、高等学校の名残であったと同時に、高等学校教員の雇用を確保することが目的であった。4 年制の大学教育は、日本社会における高度人材不足の問題を解消するためのものであった。

第 2 次世界大戦後、GHQ が主導して実施した教育改革によって、それまでの高等専門学校等の多くが大学となって、戦後の日本の復興に必要な人材の育成に携わった。当時の日本の産業社会は、大学を卒業したごく一部の幹部への昇進を約束された従業員と、中学や高校を卒業して入社した一般従業員に区別されていた。大卒の社員がホワイトカラー層を形成していた。

産業の発展が急速に進み、日本の社会は深刻な人材不足状態が続いた。大学も数多く新設され、学生数は急増した。とは言え、人口全体に占める大学卒業者の割合は、高々10 パーセントであった。労働者の大多数は、義務教育の中学を卒業して入社した者たちであった。そのため、企業では中学を卒業して入社した若い社員を、自社内の教育訓練施設で教育する制度を持つ例が多かった。

この間、日本の社会は経済の目覚ましい発展に支えられて、徐々に豊かになっていた。これによって、義務教育終了後に高校へ進学する者が急激に増加した。しかし、それでも大学まで進学する者の割合は、高々20 パーセントであった。

この時代においても、大学を卒業した人材が、ホワイトカラー層の大部分を占めた。しかし、高校卒業後、コンピュータの無かった時代に、工場の事務員として、専門家の大学卒業者の仕事を支援する業務に従事する者も多かった。彼ら、彼女らもホワイトカラー層と言えた。

このようなホワイトカラー層に属し、知的な労働に携わっていた高校卒業人材の中には、能力も高く、経験を積むことによって大卒従業員と同等の仕事に従事できる人材もいた。日本企業は、そのような人材も活用し、単純に学歴だけで就業できる仕事を決定することはなかった。これは、終身雇用が良好に機能した例のひとつである。

日本企業の人材育成は、第 2 次大戦後の大学改革後は、基本的に卒業した大学や、その大学で何を学んだかに関係なく、将来はその組織の幹部に昇進することを前提に、労働に従事させた。命じられた労働に就くために必要な基本的な知識は、社内の教育で教授する方法がとられた。また、高度な専門知識はオンザジョブで学ぶことが要請された。

このようなことから、従業員を雇用する企業にとって最も痛手が大きいのは、育成した人材が企業を去ることであった。幹部人材の育成には、15 年から 20 年の歳月が必要である。その間、企業側は従業員の労働から直接的に得られるものはほとんどない。このため、従業員を囲い込み、他社に転職する道を狭めなければならなかった。終身雇用や年功序列は、

その意味で企業における人材流出の防護壁として機能していた。

第 2 節　我が国の高等教育

ここでは、特に専門家育成のための工学系高等教育の内容を中心に日本の高等教育の特徴について論じる。大学の工学部においては、卒業生の多くが企業に入社し、専門的な労働に従事することが期待されていた。

しかし、教育プログラムを見ると、卒業後に従事する実践的な仕事には関係の薄い、どちらかと言えば基礎的な知識を教授するカリキュラムが中心になっていた。例えば、製品の設計や部品の選択などを教える教科はほとんど存在しなかった。

大学教育で特に重視されたことは、専門分野の基礎的・理論的な知識をしっかり学び、基礎的な問題を題材にした卒業研究を行うことであった。これは、どちらかと言えば、大学教員を育成するためのカリキュラムである。逆に、企業における実際の仕事をするために重要な、部品の選択や、コストの計算、故障率の計算などの実践的な知識を教授する内容の科目はほとんど存在しない。

もう一つ、日本の大学の工学部教育の特徴として、学生が卒業までに修得しなければならない単位数が非常に多いことがある。つまり、学生は朝から夕方まで大学で学び、自習する時間はほとんどない。

講義で学ぶべき知識は、そのほとんどを教員が解説し、学生自身が考えたり、調べたりする部分がない。これは、受動的な学習態度を育成・助長し、高等教育のひとつの目的である議論する態度を育成することが困難になることを意味する。

これに対して、米国の大学における工学系教育は、その内容が標準カリキュラムによって指定されている。標準カリキュラムには、各単元で何を学ばせるかまで指定されている。

また、米国の学部教育は、1 つの科目を複数人の教員の講義で進める例が多いため、それぞれの単元で何を話すべきかが、各大学で決っている例が多い。このことから、学部教育においては、教員の個性が出にくいと言える。

さらに、米国の大学の工学系学科の教育では、実社会での仕事に直結した問題や、その問題への対処法を教える実践的な科目も多い。特に、設計法の講義や、ハンドブックを利用した部品の選択に関する講義、製品のコスト計算法に関する講義などがその例である。さらに、夏季休業期間中の長期インターンシップなど、企業の現場での実習なども含めると、企業の現場でもすぐに仕事ができる人材を育成する教育が実施されている。

この日米の大学の工学部における教育内容の違いの原因は、主として教員人材の能力の違いによると言える。日本の大学の教員は、学部卒業後、大学院へ進学し、学位を取得してそのまま大学教員になる例が多い。つまり、企業などでの就業経験の無い人材が多い。

これに対して米国の大学には、企業での実践経験のある教員が相当数存在する。そのことは、アクレディテーション認証を受けるためにも重要な要件であり、大学として対応が社会的に要請されている。

第 3 節　グローバル化の進展と現代の高等教育で教えるべきこと[35]

1990 年代に始まる世界経済の急激なグローバル化は、世界中の企業で構造的な変化を起こした。特に先進国の製造業系企業では、それまで、製品の企画・開発、生産、販売、関連サービスまでを総合的に実施していた例が多かった。

しかし、先進国における労働コストが、それ以外における労働コストに比較して極端に高いことから、先進国内の工場での生産はコスト高の原因になり、経済合理性を全く失った。また、マイクロエレクトロニクス革命の結果、組込みソフトウェアでハードウェアを制御して機能を実現する例が多くなり、工場労働者の熟練度は製品の品質にほとんど影響しなくなった。

このような背景から、先進国における専門家の労働は、ますます知的労働にシフトしたのである[20]。すなわち、企画、設計開発、試験、販売などに関する知的な労働だけが先進国の作業現場に残り、それ以外の設計の実現やテスト、工場における生産などは労働コストが低い開発途上国で実施されるようになった。これは、世界的な規模での垂直分業によるコストの最適化である。

このような経済のグローバル化が進展した世界で働く技術者に求められることは、異なる国々の専門家と労働者が協業して 1 つの製品を開発・生産し、市場投入後も必要な保守体制を維持し、製品の顧客が満足する製品と付帯サービスを提供するために、チームで働くことができる人材であることである。その第一に、コミュニケーション能力があげられる。これは、相手が言いたいことを聴き、自分が伝えなければならないことを正確に伝える能力である。多くの場合、多国間でのコミュニケーションの場合、英語が用いられる。

上手な英語が使えることではなく、自分が伝えたいことを正確に伝えるための技術や能力を持つことが重要である。いくら英語に堪能であっても、伝えたいことが無ければ無意味である。また、相手の言いたいことを理解できなければ無意味である。相手が言いたいことを理解するためには、言語だけでなく、心理を理解すること、相手の文化や宗教を理解することも重要である。

明治以来、日本の高等教育では、全ての専門知識を日本語で教授できるように、全ての概念を翻訳してきた。進歩が速い今日でも、翻訳が間に合わない場合には、カタカナ語で表現するようになっている。

文法は、日本語の文法である。しかし、現在、世界中のほとんどの国で、専門に関する知識はほとんど全て英語で教育されている。それは、教科書を翻訳する時間的な余裕がないことなどが主たる原因である。

現在でも、日本の大学では、専門的な講義であっても日本語で講義が実施されている。これは、受講者である学生にとって最もストレスが無い方法であるが、新しい理論が形成されている段階で、その内容を講義するためには、関連するテーマのキーワードを全て和訳する必要があり、場合によってはかなりの負担になる。日本の学生は、比較的簡単な教

[35]拙著、ソフトウェア技術者： プロの精神と職業倫理[80]、pp.165-168 参照

科書の内容も、英語では読むことができない。

そのような教育を受けた学生が、卒業後、仕事の現場で日本語訳の無い仕様や説明書を読みながら問題を解決することは困難である。また、外国の同僚と議論する時、特定の技術用語の日本語は知っていても英語での表現を知らなければ、議論の内容を説明するのにかなりの時間と労力が要求される。その意味でも、教育の内容と方法をグローバル化することが重要になる。

第 4 節　大学の教員に求められる資質とその育成

ここでは高度な専門家の育成を目的とした高等教育の在り方と教員人材に求められる資質について議論する。これまで議論してきたように、経済のグローバル化が進む世界で、かつマイクロエレクトロニクス革命によって組込みソフトウェアが製品の最も重要な部品になる時代に生きる技術者を育成するためには、組込みソフトウェアの特性やその危険性に関する知識をしっかりと理解させることが重要である。

そのような高等教育に従事する人材の一部には、産業界において実際の製品設計・開発の経験をもつ人材が必要である。実践経験の無い教員には、何が重要であり、何があまり重要でないかの価値判断ができない。

そのため、講義内容が網羅的になる傾向がある。限られた時間の中で、効果的に教育を実施するためには、何を講義し、何を講義しないかのメリハリを明確につけることが重要になる。

特に、設計における失敗例の分析等に関する講義においては、実践で失敗を経験した社会人技術者に講義を依頼することも重要であろう。そのためには、大学における講師招聘の基準を見直すことも必要である。つまり、実践的な内容の講義を実施するためには、実社会での実践を経験している人材を教員として採用・活用することが必要である。

技術者の倫理観に関する講義については、哲学的な議論の歴史に関する知識を解説することも重要であり、そのような専門を学んだ教員に講義を担当させることも一案と言える。ただし、哲学者の講義は抽象的で、技術者を目指す若者達には内容が分かりにくい場合も多い。

重要なことは、抽象的な議論と、それの具体例に関する議論を交互に行うことである。このためには、哲学者であれば、問題分野の専門家から話を聴き、その内容からたとえ話を作るなどの対応が必要である。ここでも、実務家と哲学者の協業が重要になる。

いずれにしろ、これからの大学で技術者教育を担当する教員には、高い倫理観が求められる。また、講義すべきテーマの内容について具体的な技術の内容を掘り下げて理解する能力が求められる。

表面的な技術の説明では、しっかりとした大学教育はできない。特定の技術を説明する場合でも、その技術の歴史的な意味や、時代背景など、なぜその技術が生み出されたのかについての理論的な背景の解説がなければ、学生の身に着く教育はできない。

また、教員人材には、社会において実践の現場でどのようなことが日々、問題として議論されているのか、何が原因で問題が発生しているのかなどについて、実務家と議論し、理解を深め、技術を学ぶ学生に、何が最も重要なことであるのかを教えなければならないのかを継続的に探究することが求められる。そのような内容は、時代や時間とともに変わってゆく例が多い。したがって、1 回の調査から得た結果で、その内容を決定することはできない。実務家との定期的な交流を継続することが必要である。

高等教育における専門家人材育成は、知識の教授であると同時に、全人的な教育である

ともいえる。知識の教授であれば、教育者である教員が居なくても、教科書などの教材がそろっていれば、知識の伝達はできる。しかしながら、専門家の教育は、基本的に対面教育でなければならない。

前述したアクレディテーション認証制度においても、学生と教員との接触については、接触時間が指定されている。現在の日本の大学の教育では、その接触時間の基準を満足することはできない。それほど、専門家教育において対面教育は重要なものと言える。

第 5 節　ソフトウェア技術者育成に必要な高等教育

経済のグローバル化が進展し、先進諸国における労働は知識集約型に変化しつつある。そのような経済環境の中で、マイクロエレクトロニクス革命により製品の構成方法が変わり、組込みソフトウェアによって様々なハードウェアの動作を制御することで製品機能を実現する方法が一般化している。このことは、ソフトウェア技術者の社会的責任を、従来と比較して格段に重くしている。

このような時代環境の中で、専門家として仕事に従事するソフトウェア技術者に求められる能力は、従来の技術者に求められたようなプログラミング能力や、アルゴリズムに関する知識だけではない。特に、プログラミングに関しては、その作業に労働集約的な傾向があり、先進国でプログラミングを実施するよりも、労働コストの低い開発途上国で実施する方が経済合理性が高い。このことから、先進国のソフトウェア技術者に求められるのは、製品の企画、仕様の確定、設計と実現完了後の試験までである。

また、そのような先進国の技術者に要求される社会的役割の中には、質のよいソフトウェアを開発すべく努力を惜しまないこと、製品の実現に重大な問題が発見された場合はその問題の解決に努力を惜しまないことなどが含まれる。これらは、技術者倫理の範疇の問題である。

前者の問題に対しては、自分の責任で実施する設計の部分対して、保守性や移植性を含めた意味での品質を確保することはもちろんである。さらに、自分の同僚たちが実施した設計についても、それを第三者の視点から専門家として妥当な設計であるかどうか、真剣に検証する姿勢が要求される。

このレビューについては、高等教育における現状のカリキュラムにはそれを盛り込んだものがない。これからのソフトウェア技術者にとっては、専門家として他人の設計作業の成果を、第三者の目で評価する能力は、極めて重要な能力である。そのような重要な能力を育成するためのカリキュラムが存在しないことは、構造的に極めて問題がある。

上述した問題の後者に関しては、専門家として自分が開発に関わっている製品に重大な品質上の問題が発見された場合、どのような対応をすべきかの問題である。日本人の場合、自分が所属する相対的に小さな組織の利益を優先し、より大きな視点から見た組織や社会の利益を軽視する傾向がある。

これは、日本人社会の文化的な特性であり、日本人として育てられたときに、自然に獲得する習性である。しかしながら、この習性は、専門家としてそのままで済むものではない。この習性を、専門家として働いてる場合にも放置すると大きな問題がある。

組込みソフトウェアで制御される構造の製品では、ソフトウェアが重要な役割を担う。そのソフトウェアに誤りが潜在していても、一般の消費者には全く認知できない。このことから、一般の利用者は製品を開発した企業やその従業員である技術者に全幅の信頼を寄せ、その製品を購入し、利用する。

もし、その製品の組込みソフトウェアに重大な欠陥が潜在していれば、その製品を開発

した企業と技術者は、その利用者の信用を裏切ったことになる。資本主義の社会では、取引においてその当事者たちは、最善を尽くすことが前提とされているが、この場合それをしなかったと言うことになる。

このようなことから、重大な問題が発見された時、専門家としての技術者には、取るべき行動の規範がある。その規範に基づいた行動を実践できるように教育することが、社会の秩序を維持するためには重要である。

その第一は、発見した問題を報告し、皆で共有して、解決を模索することである。企業活動として実施している製品開発の場合、この解決を模索する行為には、時間的にも経済的にも限界があることが多い。このため、組織内の決定としては、問題を認知しながらも製品を世に出すことを決定する例がある。

この時、組織の決定は、問題が原因で事故が発生する確率は十分に小さいとの認識で、製品を世に出し、利益を確保することとなる。しかしながら、事故等が発生する確率が十分に小さいと判断されたとしても、そのような事故が発生しないわけではなく、いつかは発生するのである。

ここに、技術者の社会的責任と組織の一員として行動することの矛盾が発生する。一般的には、そのような場合、その問題の存在を組織の上層部に知らせ、しかるべき対応を採るべく助言することが求められる。

この場合でも、組織の上層部は、組織として利潤追求を優先して製品を世に出すことを決定する可能性がある。そのような決定がなされた場合、問題が人命にかかわるような事故の原因になるような性質のものであれば、技術者としては社会に対して警鐘を鳴らす意味で、内部告発に踏み切る選択肢がある。一般的には、このような問題は、内部告発をしてでも販売を差し止めるべきである。

しかし、内部告発は、その制度によって事故の犠牲になる人々が発生する事態は防げるが、内部告発を実施した技術者本人には、多くの場合、組織的な制裁が行われる例がある。仮に、そのような場合に、その企業を解雇され、なおかつ、他の企業への転職ができなかったとすれば、当該技術者の技術者としての生命は断たれたと言える。そのような報復に対する恐れから、多くの技術者は内部告発の選択を行わない。

このような所属する組織の利益を優先する価値観では、これからの時代の技術者としては、その社会的責任を全うできない。その意味で、技術者の社会的責任と、そのあるべき姿をしっかりと教育する内容の教育プログラムを確立すべきである。

このような問題は、技術者の倫理観を確立する教育のテーマでもあるが、より広く考えれば、コンピテンシの問題でもある[lxvi][79]。コンピテンシは、高等教育で養成できる行動の特性ではなく、幼児期の育児に始まる長い教育・訓練によって育まれるものである。

ただし、上述したような典型的な社会正義の問題について言えば、コンピテンシに関係なく、一般論として議論することは可能である。しかし、その実践においては、しかるべきコンピテンシが備わっていなければ、勇気ある内部告発の実践はできない。

第10章 終章

第１節　まとめ

本論文においては、組込みソフトウェアの応用が著しい勢いで普及しつつある現代の製品開発においては、従来のような機械や電気電子の技術にまして、ソフトウェア技術の適切な利用が重要になりつつあるとの認識を述べた。そのような問題意識は、IEC 61508 の機能安全規格に代表されるように、社会的な問題になりつつある。

その根源には、物理的な理論を直接応用する機械や電気電子の理論に基づく製品のように、因果関係が明確で、利用条件によって動作が変わらない決定性が、成立していない問題がある。非決定性が導入されたことにより、製品の動作は複雑になった。このため、従来の製品開発で品質保証の重要な手段であったテストが、有効な手段とは言えなくなったのである。

マイクロプロセッサが安価に利用できる時代が到来するまで、ソフトウェアは、高価なコンピュータを機能させるための手段として、多大な労力を投入し開発されていた。初期のソフトウェアは、基本ソフトウェアがコンピュータメーカによって開発され、個別の目的で計算を実行するためのアプリケーションは、コンピュータを利用しているユーザによって開発されていた。

従って、市場に供給されていたのは、コンピュータと基本ソフトウェアで、それらはバンドルされて供給されていた。これが変わり、ソフトウェアが独立の製品として市場に供給されるようになって、ソフトウェア市場が成立した。

しかし、ほとんど無償での配布も可能なソフトウェアを高い価格で販売し、膨大な利益を得ることに倫理的な問題を感じた人々は、インターネットを利用して、有志が作成したソフトウェアを無償で配布するオープンソースソフトウェアを考案した。これによって、ソフトウェア産業は大きく変化した。ソフトウェアの開発は、一部は経済活動であり、一部が公的な活動となった。オープンソースソフトウェアは、組込みソフトウェアを利用した製品開発にも大きな影響を与えている。

オープンソースソフトウェアの誕生以前から、本質的にコピーの容易なソフトウェアの知的財産権保護の問題は、専門家や法律家の間で問題になっていた。その究極の問題は、他人が考え出したアイデアを真似ることに対する倫理観の問題である。他人が考え出したアイデアを真似ることに対して、何らの問題意識を持たない社会もあれば、そのような行為自身を嫌う社会もある。

それは、それらの社会で共有されている倫理観の問題であると言える。倫理に対する哲学的議論は、ギリシャ時代から行われており、職業倫理として知られているものに、医学におけるヒポクラテスの誓がある。

ギリシャ人の哲学では、労働の意義と倫理に関する深い議論はなかった。この傾向は、中世ヨーロッパが終わるまで続く。宗教改革において、ルターやカルバンは、初めて労働

の重要性について言及した。とくに、真剣に働くことの重要性について説いた。

さらにロックの基本的人権の思想が提唱され、人々の間で、財産の保有が公に認められてから、新教国において資本主義が著しく発展する。真剣に働き、富を得ることは善を為すことだとする倫理観が一般化したのである。

この労働の倫理観は、その後、20 世紀前半まで、産業化時代が終わりを迎えるまで、維持される。しかし、ポスト資本主義の時代に移行すると、労働の内容が知的なものに変化し、人間と労働との関係が変わった。

人間にとっての労働は、自己実現の手段になったのである。このとき、どのような自己実現を目指すかの倫理観は、個々の人間にとって重要な問題になる。

ところで、非決定性を活用し、高度な機能を比較的低コストで実現できる組込みソフトウェアの利用は、産業的な製品の生産において大きな技術革新を起こした。それは、テストが製品の品質を確認するための有効な手段ではなくなったため、新しい方法が必要となったことである。

と同時に、良い製品とは、どのようなものであるかの品質に関する今日的概念も少しずつ変化しつつある。物の質の概念は、プラトンの質の概念が起源であると言われている。

その概念を、アリストテレスが厳密に定義し、その延長線上で、18 世紀になってアダム・スミスは財の効用を議論した。さらに、20 世紀に入って、米国のシューハートは、今日、大量生産品の品質と呼ばれている概念を、製品が仕様に適合している確率であると定義した。

この統計的品質管理の考え方が、20 世紀を通じて品質の基本的な概念となった。しかし、その概念だけでは消費者や製品のユーザから見た製品の価値を適確に表現したものとして捉えることはできなくなった。

1970 年代に入ると、製品のライフサイクルを通して製品の利用者にとっての価値を問題にした稼働率を保証する製品保証の考え方が提案され、一般化した。さらに日本では、製品の品質が本質的に異なる 2 つの概念に分類できることが認識され、「当たり前の品質」と「魅力的な品質」に分割された。当たり前の品質が従来の品質の概念に相当し、不良率や稼働率を問題にし、魅力的な品質気はユーザ視点での品質の本質的な価値を高める性質に関する特性を問題にした。

1990 年代に入ると、市場のグローバル化が始まり、グローバルな市場における製品の競争力を問題にして、顧客満足度が問題にされるようになった。このように品質の概念は、時代とともに変遷している。

つまり、唯一絶対の品質概念は現時点では存在していない。このことから、製品を開発する技術者は、自らの価値観に基づいて、長期的な視点で消費者やユーザの満足する製品を企画、設計、実現する責任を負う。

組込みソフトウェアを利用した製品の設計や開発を実際に担当する技術者と、そのような技術者を雇用する企業との関係には、日米で大きな違いがある。それは、日米における

労働観を反映したものである。

日本の社会における労働は、技術者が雇用されている組織の根底にある他の人々と構成しているコミュニティへの帰属関係を最優先し、そのコミュニティのために仕事として自分が何をすべきかを考える。これに対して米国社会における労働は、技術者のもつ専門性を、その専門性を必要としている組織が活用し、組織の目的を達成することに重点が置かれている。

従って、技術者に求められていることは、何よりも与えられた任務、責任を全うすることである。このような背景から、米国では職務記述書に基づく雇用が原則になっている。これに対して日本においては、被雇用者である技術者の仕事は、雇用する企業が置かれた状況によって変化する可能性があり、専門外の内容を主とした労働であっても、任命されれば技術者はその労働に従事することが求められる。

つまり、日本企業における雇用は、企業社会のメンバーとして、参画していることに重点を置いたものになっている。このことは、専門家としての技術者の視点から見れば、自分の専門家としての成長に支障をきたすキャリアを歩まなければならないこともあることを意味する。

ただし、専門家としての技術者と言えども、人間としての成長があり、専門性を優先した仕事から、より広い知見に立って、部下を指導する立場に立った仕事の遂行に意義を見出す人は少なくない。このような、技術者のライフサイクルに適合した仕事に従事させることができる日本の雇用制度は柔軟性が高く、効率的である例も少なくない。

製品の実現に組込みソフトウェアを活用した事例が増加したため、製品設計や製品の品質保証は、従来の製品開発に比較してはるかに難しくなってきている。特に、前述した非決定性が導入されたことにより、テストが設計妥当性検証の手段として有効に機能しなくなったことで、市場に製品を供給する企業や、その企業内で製品開発を担当する技術者達に対して、従来とは異なる使命感をもって仕事に従事することを社会は要請するようになった。

市場に供給されている製品に不具合が発見された場合の、社会の製品供給企業に対する制裁は、従来の製品にはなかったほどの強いものになりつつある。米国市場におけるトヨタ車の異常加速問題は、トヨタにとって多大な負担を負わなければならない結果となったことはその典型である。

この問題で明らかになったように、重要なことは、結果的に企業における製品開発に瑕疵があったかどうかではなく、消費者が納得できる事実やデータを企業側が社会に提示できるかどうかである。つまり、企業の社会的責任を全うできるかどうか、技術者は専門家として自分の責任を全うしていることを社会に説明できるかどうかである。

そのためには、企業もそこで働く技術者達も、日常から倫理規範に即した行動を求められるようになっている。特に、技術者にとっての仕事は、21 世紀の社会においては、自己実現の手段となっている。このことをしっかりと理解した仕事への姿勢が、技術者個人個

人に求められている。

そのような社会・経済環境において、行政が担うべき基本的役割は、以下の 3 つである。第一は、新しい社会のための社会秩序の構築である。特に、ソフトウェアに関して言えば、知的財産権の侵害が容易な実体であるソフトウェアに対して、どのような法的制度を整備して、先行開発者の投資から得られている利益を保護するかである。

過去の時代のような、著作権を主としたソフトウェアの保護政策は、限界に達している。特に、オープンソースソフトウェアが普及している現代社会では、著作権は万能ではなくなっている。また、各国政府にとって、ソフトウェア産業の振興策は重要な政治課題である。ここでは、新技術の開発と、産業の育成・高度化が問題になる。

第二は、人材の育成である。ソフトウェアの開発において最も重要な資源は、人材である。他国との競争に勝てる人材を、他国よりも多く育成することが、国家の基本戦略となる。

ここでは、教育の高度化と専門家人材育成が重要な課題となる。特に、高等教育における専門家人材育成については、教育プログラムの開発と、その実践に当たる教員人材の育成が問題になる。

第三は、専門家人材の雇用である。雇用者である企業組織と、被雇用者である技術者との間に締結される雇用契約の形態として、どのような雇用契約にすべきかは、長期的に重要な問題である。日本においては、従来型の雇用形態は、専門家であるソフトウェア技術者に適合しない。

特に、専門家としての人材を育成するためには、非効率である。専門家を育てるためには、高等教育で専門的な知識を獲得している人材を活用すべきである。さらに、実践的な知識を付けさせるためには、経験を積ませることが重要になる。

そのためには、雇用者である企業を限定せずに、高度な実践経験を積ませることが重要になる。このことに関して、現在の日本の雇用形態の基本となっている終身雇用、年功序列、新入社員一括採用の制度には問題がある。

最後に、組込みソフトウェアの開発を担当する専門家人材の育成における高等教育について考えた。日本の高等教育は、ヨーロッパの大学制度を真似、明治政府主導によって作り上げられたものである。明治政府の課題は、近代国家の早急な建設であったと言える。

そのために、行政組織を作り、その運営を担う人材が必要であった。また、日本の工業化を促進するための、海外からの技術導入を担う人材を必要としていた。そのような、人材を育成するための制度として大学が設置された。

1945 年に日本は第 2 次世界大戦に敗北し、民主主義国として再出発をせざるをえなかった。この時、従来の軍事産業を中心とした工業国ではなく、民事産業を中心とした工業先進国の建設が課題とされた。

大学制度もこれに伴って大きく変わり、国家が直接運営に関わる国立大学は、国家の財政的支援を受けて運営される新制大学として再設立された。その結果、現在の日本の若者

全体の約半分を教育するのに十分な大学が設立され、現在、運営されている。

この戦後の大学政策は、新設される大学の教員人材の需要を生み出し、各大学における大学院進学者も、順調に伸びた。しかし、1990 年代に入り、日本経済成長が鈍化し始め、若者人口の伸びも鈍化し、大学への進学者が激減し始めた。これによって、各大学は生き残りのための学生獲得競争に参入しなければならない状況に追い込まれた。

このことによって、各大学は入学者選抜の方法を変え、教育カリキュラムにおいても、就職率を向上させる方法を採用して、高校からの進学者を確保することが重大な問題になった。

しかし、この間、日本の企業が置かれている経済環境は大きく変化し、企業間競争とすさまじい勢いでのグローバル化進展が浸透しつつある。そのような企業間競争を勝ち抜くためにも、各企業は、従来以上に有能な人材を必要としている。特に、製品を開発し世界市場に供給することを業務としている企業においては、他国の企業との競争に勝つためには、多国の企業の従業員の能力を上回る能力を持った人材の獲得が急務になってきている。

このような社会的な要請に応えるためには、これまで日本の大学が講じてきた対策のような、近視眼的かつ対症療法的対応では不十分である。特に、企業内において、専門家としての労働を求められている組込みソフトウェア開発に従事する技術者には、国際的な水準でも他国の技術者の知識を上回り、さらに他国の技術者に負けない職業倫理観を持った人材が必要とされている。

そのような人材を育成できる大学の教育プログラムの開発と、そのような人材を育成できる教員人材の育成が重要になっている。

第 2 節　提言の要約

本論文では、マイクロプロセッサの長足な進歩により、製品開発に組込みソフトウェアを活用することで、高度な機能を実現し、製造コストを極端に低減させることが可能になった社会において、組込みソフトウェア開発を担当する技術者達には、これまでの製品開発を担当してきた技術者達よりも重大な責務が課せられていると言う認識に基づき、その育成をどうすべきかについて、議論してきた。

特に、良い製品を実現すると言う目標に立ち向かう技術者に必要とされる倫理観について議論した。ここに、その提言をまとめる。

組込みソフトウェアを利用して製品を開発する方法を導入したため、低コストで高機能な製品を実現できるようになったが、非決定性の導入によって、これまでの製品にはなかった、テストで製品の品質を確認し、保証することが不可能になった。

この問題を克服するためには、設計や開発のプロセスを厳密に定義し、そのフロセス定義に従って作業を実施し、さらに、作業結果を丁寧に確認することが重要になる。この作業結果の確認には、可能な場合、数学的な方法や論理学的な方法の適用が必要になる。

このことは、技術者に対して、これまでの技術者達以上に、真剣な態度で作業をしなければならないことを要求する。実際に、数学的な証明法の適用は、部分的に実践され、実験されている。この証明が倫理的に必要であることを技術者達に理解させなければならない。

倫理的な行動とは、必ずしも遵法的な行動規範に則って行動することを言うわけではない。ソクラテスは、「徳を実践すること」を倫理的と言った。つまり、どのような環境にあっても、技術者が「正しいと信じていることを実践する」ことが重要である。

これは、「これをしてはならない」と言う遵法的な精神を言うのでなく、「人としてこれをしなければならない」と言う徳の実践の精神を言うものである。具体的に、設計について言えば、「この設計でならなければならない」と信じる設計をすることである。「この設計でも良い」でも「この設計も可能である」ではない、正しい設計を考え、選択する能力と、倫理観の醸成が必要である。

「良い設計」や「良い製品」の概念は、時代とともに変化する。それは、品質の概念が直感的な概念ではなく、かつ先験的な概念であることによる。1970 年頃まで、「良い品質」とは、購入した製品が仕様に適合する確率であった。

ソフトウェアの場合、ソフトウェアに残存する誤りの数の多さや、それによって誤動作が発生する確率を意味していた。1980 年代に入って、「良い品質」は「仕様通りに実現されている確率」だけを意味するものではなく、利用者にとっての価値を示す様々な属性の総合的な判断であるとされた。

今も、ソフトウェアの品質の定義は、変わりつつある。そのような状況において、「品質の良いソフトウェアを開発する」ためには、開発を担当する設計者において、「どのようなソフトウェアが良いソフトウェアであるか」についての理解が存在しなければならない。

それは、普遍的なソフトウェア品質の定義でないとしても、技術者にとってはその時点で絶対的なものでなければならせない。そのような品質の概念について深い理解を持つことが、専門家としての技術者に求められている。

日本の雇用形態は、現在、新入社員一括採用、年功序列、終身雇用を原則としている。ここでは、新入社員が大学で専門として何を学んだかは、ほとんど問題にされない。この雇用形態は、若年層における労働コストを低減させ、大学教育に依存しない人材の選択を可能とし、企業組織に適した人材の育成を可能とする、極めて柔軟な雇用形態である。

しかし、競争の激しいグローバルな市場で、外国企業との競争にさらされている現代の日本企業の視点から見ると、競争力のある専門家を育成する必要のある企業組織にとっては、非効率な雇用形態である。雇用した技術者を即戦力として仕事に従事させることが可能な職務記述書に基づいた雇用形態は効率が良い。

また、技術者の企業間の移動を可能にするため、社会全体での人材活用効率は向上する。そして、同一労働同一賃金制度を導入しやすくなるため、外国人技術者の雇用など、多様性を要求するこれからの社会には、より適合しやすくなる。

組込みソフトウェアの活用が進むこれからの社会においては、企業においても倫理綱領の公開や、その倫理綱領に基づいた企業経営の実践が強く求められるようになる。さらに、そのような企業で働くソフトウェア技術者などの専門家に対しても、専門家としての倫理綱領に則った活動の実践が求められる。

専門家には、雇用者である企業の倫理綱領に則って活動する義務もあるが、技術者としての倫理綱領に則り、社会に対する責任を全うすることが求められる。企業の利益と社会の利益との間に、齟齬が生じた場合、専門家としては、専門家の倫理綱領に基づく対応が迫られる。

それは、その個人にとっては不利益が発生することが予想される場合であっても、である。技術者の育成においては、そのような専門家としての社会的責任を全うできるような人格の育成を心がけねばならない。

各国の行政は、新しい社会の知的財産権保護に関する新しい制度の確立に努力しなければならない。ソフトウェアなどの知的生産物は、工業生産品と異なり、マネや複製の作成が極めて容易である。

そのため、先行投資を行った個人や組織の利益を保護することは容易でない。従来の著作権や発明権は、印刷物や工業製品などの知的財産権保護を念頭に置いて制定されており、必ずしもソフトウェアのような知的生産物に適合するものではない。

知的生産物の財産権保護は、基本的にはその開発を担当する技術者達の倫理観によって、自然に達成されるべきものであろう。しかし、社会的秩序の維持のために、倫理規範に順じた行為を実践することを、社会的に規定する必要はある。

さらに、日本においては、専門的労働を実践する人々を対象として、職務記述書に基づいた新しい雇用形態の確立、導入に努力すべきである。これによって、経験を積んだ既卒

社会人の採用や、社内の有能な人材の活用を可能とする同一労働・同一賃金の実践が容易になるからである。

大学教育制度についても、各国の行政機関は、改革に努力する必要がある。特に、技術者に対する社会の要請が変化しつつあるため、その社会的な変化に対応した教育の改革が必要になる。

その一つは、教育する知識の内容に関するものであり、もう一つは、専門家と一般の人々との知識の差が拡大する問題に対応するものである。前者については、特に、理論的な知識の教育だけでなく、仕事の実践に必要不可欠な経験的知識をしっかりと教育することの重要性を認識し、教育プログラムを改善することが必要になる。

後者については、専門家としての技術者が、技術を知らない一般の人々を考慮し、製品を設計する態度を身に付けさせる教育を実践する必要がある。これも技術者の倫理的活動の実践の問題である。

第 3 節　謝辞

本論文を作成するに当たり多くの方々とご議論を頂いた。特に、産業界の専門家、大学に籍を置く教員各位、さらに現場で行政に携わっておられる方々より、それぞれの立場、視点からのご意見を頂いた。また、2009 年からの数年間にわたり、著者が勤務していた大学で技術者倫理の講義を担当した。その講義を担当し、受講者である学生達からも様々な、新しい視点からの問題提起を頂いた。これらの全ての関係者に心から感謝の意を表する次第である。

特に、2014 年に「ソフトウェア技術者: プロの精神と職業倫理」を出版するに当たり、企画段階より日科技連出版の担当者より、様々な建設的ご意見を頂いた。頂いたご意見に対しては、可能な限り適切に対応するよう努力したつもりであるが、著者の能力の限界から、必ずしもその意に沿った対応ができなかった部分もあったに違いない。ここに記して、謝意を表する。

また、学位論文として最終稿を作成するに当たり、広島修道大学大学院経済科学研究科の諸先生方より、丁寧なご批判を頂いた。特に、主査を担当して頂いた廣光清次郎教授には、多大なご協力とご支援を頂いた。論文の精査と問題点のご指摘には、心より感謝の意を表する次第である。ご多忙中にもかかわらず、副査の任をおとり頂いた西田友是、海生直人の両先生方にも、感謝の意を表する次第である。

最後に、著者の IBM 社勤務時代の研究所の直接の上司であり、ソフトウェア技術と産業、特に技術移転の問題、技術者教育と 21 世紀の社会の課題について、幅広い視点から米国社会における問題点についてご教示をいただいた L. A. Belady 氏に心よりの謝意を表す次第である。Belady 氏の故郷であるハンガリーのブタペストにある母校、ブタペスト工科大学から、彼が名誉博士の称号を授与され、彼の存命中に著者自身が本学位論文を執筆できたことは、著者自身、大変な幸運に授かったと認識する次第である。

思い返せば今から 30 年ほど前、日本 IBM の東京に設立されたサイエンスインスティチュート(現東京基礎研究所)のオフィスの Belady 氏の執務室で、「君は、絶対に米国に渡り、米国で仕事をすべきだ。この会社で働く限り、それは絶対に必要なことだ」と諭された。そして、徹底的な英語の訓練をされ、彼とともに、米国 IBM の研究組織と米国で開催されたソフトウェア関連の学会に参加し、著名な研究者やビジネスマンと知り合うことができた。このようなきっかけがなければ、著者が本論文を執筆することはなかったであろう。

著者が米国 IBM での勤務を終えて、日本 IBM の SE 研究所に籍を置き、日米のソフトウェア技術の違いについて分析を始めた頃、NTT のソフトウェア研究所で部長をされていた長野宏宣氏と、日本のソフトウェア産業の改革を議論し、その結果を当時の通商産業省で説明し、若手の官僚たちと議論をした。

その結果、後の CALS プロジェクトの一部となったソフトウェア CALS が誕生する結果になった。このプロジェクトを遂行して、日本のソフトウェア業界が抱えている根深い問題について、若手官僚たちと議論する機会があった。彼らも、著者や長野氏と共通の認識

を持っていた。その数年間にわたって繰り返された議論がなければ、著者が本論文の執筆を完結できることはなかったであろう。

さらに、2010 年以降、日本科学技術連盟が主催しているソフトウェア品質関係のシンポジウムにおいて、複数回にわたり、日本のソフトウェア技術と技術者の問題について、発表を行う場を与えられ、大野侑郎氏、飯塚功氏、板倉稔氏、金子龍三氏などと数回にわたり、品質と倫理や技術者の倫理の問題について議論を重ねてきた。

これらの議論が、本論文の骨子になっていることをここに記して、ここに諸氏に対する謝意を表する。これらの人々との議論がなければ、本論文を執筆することはできなかった。

参考文献

[1] Brooks, F., The Mythical Man-Months, Addison-Wesley (1975)
[2] クスマノ、日本のソフトウェア戦略、三田出版会(1993)
[3] ファーロング、カートメル、若者と社会変容、大月書店(2009)
[4] オープンソースソフトウェア委員会、報告書、日本規格協会 (2008)
[5] ストールマン、フリーソフトウェアと自由な社会、アスキー (2003)
[6] 可知　豊、ソフトウェアライセンスの基礎知識、ソフトバンククリエイティブ (2008)
[7] 半田正夫、著作権、教育社 (1979)
[8] シュベーグラー、西洋哲学史、岩波文庫 (1966)
[9] アリストテレス、ニコマコス倫理学、岩波文庫 (2008)
[10] サンデル、これからの「正義」の話をしよう、ハヤカワ・ノンフィクション文庫(2011)
[11] デューイ、哲学の改造、岩浪文庫(1968)
[12] ヴェーバー、プロテスタントの倫理と資本主義の精神、岩波文庫(1989)
[13] フィリップソン、アダム・スミスとその時代、白水社 (2014)
[14] カント、実践理性批判・人倫の形而上学の基礎づけ、岩波書店(2000)
[15] 清水正徳、働くことの意味、岩波新書(1982)
[16] バーンスタイン、豊かさの誕生、日本経済新聞社 (2006)
[17] テイラー、科学的管理法、ダイアモンド社 (2009)
[18] コントスポンビル、資本主義に徳はあるか、紀伊国屋書店(2006)
[19] フロリダ、クリエィティブ資本論、ダイヤモンド社 (2008)
[20] ドラッカー、ポスト資本主義社会、ダイヤモンド社 (1993)
[21] 大場　充、組込みソフトウェア工学ハンドブック、日科技連出版社 (2014)
[22] Friedrichs, G., Microelectronics and Society, Pergamon Press (1982)
[23] 足立暁生、計算基礎論、オーム社 (1986)
[24] ウィルクス、初等統計学、東京大学出版会 (2000)
[25] Fagan, M.," Design and Code Inspections to Reduce Errors in Program Development," IBM Systems Journal 15(3), pp.182-211, 1976
[26] センゲ、最強組織の法則、徳間書店 (1995)
[27] Humphrey, W., "Characterizing the Software Process: A Maturity Framework," IEEE Software, Vol 5(2), pp.73-79, 1988
[28] 東基衛編、ソフトウェア品質評価ガイドブック、日本規格協会 (1995)
[29] McCall, J., et al, "Factors in Software Quality," National Technology Information Service, Vol. 1-3, 1977
[30] マッキンタイア、美徳なき時代、みすず書房 (1993)
[31] ISO 発行 ISO9000:2005 (Quality Management Systems)

[32] IEC 発行 IEC61508 (Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems)
[33] ISO/IEC JTC1 発行 ISO/IEC 13568-2002 (Z)
[34] ISO/IEC JTC1 発行 ISO/IEC 13817-1 (VDM-SL)
[35] 中村　元、日本人の思惟方法、春秋社 (1989)
[36] 小倉和夫、日米経済摩擦、日本経済新聞社 (1982)
[37] 経済産業省、GATT ウルグアイ・ラウンド報告 10 章、http://www.meti.go.jp/committee/summary/0004532/pdf/2012_02_10.pdf
[38] ドラッカー、ネクスト・ソサエティ、ダイヤモンド社 (2002)
[39] サロー、富のピラミッド、TBS ブリタニカ(1999)
[40] 特許庁、プロパテント政策の一層の深化に向けて、http://www.jpo.go.jp/shiryou/toushin/toushintou/ki6_1.htm
[41[中山　茂、科学技術の国際競争力、朝日新聞出版 (2006)
[42] 情報処理推進機構、組込みシステムの安全性向上の勧め、オーム社 (2008)
[43] ブルックス、人月の神話、丸善出版 (2014)
[44] カント、実践理性批判、岩浪文庫(1979)
[45] 竹田青嗣、人間の未来 – ヘーゲル哲学と現代資本主義、ちくま新書 (2009)
[46] Sandel, M., The Case against Perfection, Belknap Press (2009)
[47] Drucker, P., “The New Society of Organizations,” Harvard Business Review, September-October, 1992
[48] 野中郁次郎、「ナレッジ・クリエイティング・カンパニー」、ハーバード・ビジネス・レビュー1992 年 2-3 月号
[49] キッザ、IT 社会の情報倫理、日本経済評論社 (2001)
[50] アダム・スミス、国富論、岩波文庫 (2000)
[51] 吉田富義、賞品学 - 歴史と本質、国元書房 (1978)
[52] 石川　馨、日本的品質管理、日科技連出版 (1984)
[53] 飯塚悦功監修、「ソフトウェアの品質保証　ISO9000-3 対訳と解説」、日本規格協会 (1992)
[54] 大場　充、「学習する組織」、情報処理、第 38 巻 5 号 (1997)
[55] Ohba, M. “The Impact of Organizational Knowledge on Quality,” American Programmer, June 1993
[56] 大場　充他、ソフトウェアプロセス改善と組織学習、ソフトリサーチセンター (2003)
[57] 大場　充、ソフトウェア・プロジェクトの実績データ収集・分析技法、ソフトリサーチセンター(1993)

[58] Senge, P., The Fifth Discipline: the Art of Organizational Learning, Doubleday (1990)
[59] コールバーグ、道徳性の発達と道徳教育、麗澤大学出版会(1987)
[60] 濱口桂一郎、新しい労働社会、岩波新書 (2009)
[61] 島田晴雄、日本の雇用、ちくま新書 (1994)
[62] ドーア、働くということ、中公新書 (2005)
[63] リンカーン、それでも日本は変われない、日本評論社(2004)
[64] 山田昌弘、希望格差社会、ちくま文庫 (2007)
[65] ウィルキンソン、格差社会の衝撃、書籍工房早山(2009)
[66] 天野郁夫、大学の誕生、岩波新書 (2009)
[67] 吉見俊哉、大学とは何か、岩波新書 (2011)
[48] 三好信浩、明治のエンジニア教育、中公新書 (1983)
[69] 村井　実、教育の再興、講談社 (1975)
[70] 石黒一憲、国際摩擦と法、ちくま新書 (1994)
[71] 奥井智之、日本問題、中公新書 (1994)
[72] Brooks, F., "No Silver Bullet," *IEEE Computer*, April 1987.
[73] アリストテレス、形而上学(上)、岩波文庫 (1959)
[74] 中山信弘, 「ソフトウェアの法的保護をめぐる日米対立」, 現代経済 60 号, 現代経済研究会, 1984 年冬季.
[75] 濱口桂一郎、日本の雇用と労働法、日経文庫 (2011)
[76] メトリ、人間機械論、岩浪文庫 (1957)
[77] 高石義一、「ソフトウェアの法的保護と今後の課題」、コンピュートピア 1983 年 3 月号
[78] スターン、「オブジェクト・コードを著作権で保護できるか」、日経エレクトロニクス、1983 年 2 月 28 日号
[79] スペンサー他、コンピテンシー・マネジメントの展開、生産性出版 (2001)
[80] 大場　充、ソフトウェア技術者: プロの精神と職業倫理、日科技連出版社 (2014)

注

[i] このIBMにおけるOS/360の開発作業の実態については、ブルックスによる「人月の神秘」において、ブルックスの経験談として詳細に記録されている。その内容には、今日でも我々がソフトウェア開発で経験する普遍的な問題が多く記述されており、参考になる。

[ii] OS/360などの初期のオペレーティング・システムのジョブ管理では、このようなジョブと呼ばれるデータ処理の実行順序までを計画し管理する機能は実装されていなかった。ここでは、ジョブ管理サービスの説明のため、後継のオペレーティングッ・システムに実装されたジョブ管理サービスを含めて説明した。

[iii] それまでのようにソフトウェアを、ハードウェアを購入・レンタルしたユーザ企業に対して無償で提供する方式を、ハードウェアとソフトウェアが一体として提供されていることから、「バンドリング」制と呼ぶ。これに対して、ソフトウェアをハードウェアから独立させ、別個の製品として販売する方式を「アンバンドリング」制と呼ぶ。

[iv] 1995年11月2日付け日本経済新聞に掲載された「電算ソフトは大幅入超」の見出しの記事によると、日本のソフトウェア産業によるソフトウェアの輸出額は、その輸入額の20分の1であるとの報道があった。これは、ゲームソフトの輸出を除いた数字である。1994年の輸出額は約135億円に対して輸入額は約2,595億円であった。この記事では、日本のソフトウェア産業が、受託開発業務に特化していることを指摘していた。

[v] 日本のソフトウェア工場におけるソフトウェア開発と、米国のソフトウェア開発組織におけるソフトウェア開発の生産性と品質の差については、1980年代末から専門家によって研究されてきた。現在でも、差があるとする見方と、差がないとする意見が対立している。多くの場合、差があるとする結論は、両国の産業界における生産性と品質の計測の基礎となるソフトウェア開発規模の計測方法の違いを考慮せずに、数字を比較した場合のものが多い。逆に差がないとする結論は、投入された労力の計測(総労働時間の計測)の誤差を十分に考慮していないものが多い。人間が実施する作業なので、単位時間当たりのコード生産量にしろ、単位生産コード量当たりの欠陥混入率にしろ、大きな違いがあること自体が不自然である。したがって、両国における実践の比較は、微小な差をバラツキの大きなデータ間で比較することになる。その平均値の微小な差を有意とみるかどうかである。著者は、データの性質から、そのような細かな粒度の議論は不可能であるとの立場に立つ。したがって、両国の実践の間に大きな違いはないと考えている。ただし、相対的には、日本の実践の方が最終的なソフトウェアに残存する欠陥数が少ないが、生産性は米国の実践の方が高いと言える。これは、両国の国民性の違いもあるが、組織がどのように運営されているかの影響が大きい。米国の組織は、プロフィットセンタとして運営されており、日本のソフトウェア開発組織の場合は、コストセンタとして運営されている例が多い。この違いが、最終的なソフトウェアの品質を改善するために、どこまでテストに時間をかけるかに影響していると分析している。

[vi] アイバーソン言語(APL)は、アイバーソンによって、コンピュータアーキテクチャ記述用言語として提案されたと言われている。実際に、IBMはS/360のアーキテクチャをAPLで記述したと言われている。APL言語は、関数型言語と言われ、極めて多くの特殊記号を使う言語として有名であった。また、式の右辺の計算はカッコを使わず、左から右へ計算を進める方法が採用されていた。例えば、(A+B)*Cは、A+B*Cとなる。
これは、コンピュータにとっては、処理の容易な方法である。

[vii] 現在、日本の中学および高校で実施されている情報教育は、情報リテラシ教育と呼ばれる教育の範疇に入るものである。そこでは、パーソナルコンピュータの基本的な構造と操作、電子メールとブラウザの機能、文書作成ソフトウェアや表計算ソフトウェア、さらに発表資料作成ソフトウェアの使用方法について学ぶ。

[viii] このようなボランティア活動は、一般に「コンピュータデイ(computer day)」と呼ばれ、PTA の呼びかけで様々な企業で働く技術者達が、同じ日に小学校などの体育館に集合して、中古コンピュータの解体作業等を行った。このことには、後述するように、米国社会が市場における自由な競争を原則としながらも、次世代を担う人々になるべく公平な競争が可能となるような環境を整えるべきだとする個人個人の倫理観に強く支えられていることを見ることができる。

[ix] ジェネリックスキル(generic skill)とは、社会で仕事をするために必ず必要になる能力の中で、特に仕事の専門性や分野に関係なく要求される能力を言う。例えば、コミュニケーション能力は、他人の話を聴き、理解したり、他人が書いた記述を読み、理解したりするときに必要な、基礎的な言語能力を超えた、文脈を理解する力などを言う。さらに自分が自分の意見や考えを他人に伝えるための話を組み立て、表現する能力や、文章に表現する能力を言う。それ以外にも、問題を分析する能力、解決策を作り出す能力、解決策を実践する作業を計画・実行する能力、議論する能力、論理的に物事を考える能力などが含まれる。OECD が実施を準備している高等教育成果の評価試験である AHELO のテストでは、専門分野の知識に関する問題もあるが、ジェネリックスキルを問う問題も多い。

[x] アラン・ケイとネグロポンティらが提唱したプロジェクトで、アフリカの子供たちに教育目的のコンピュータとソフトウェアを無償貸与することを目的として実施された。TED のアラン・ケイの講演とネグロポンティの講演において提案の詳細が説明されている。この二人のカリスマ性の高い研究者の講演は、次世代を担う子供たちの教育のために、我々は今、何をすべきかについて、明確なメッセージを伝えてくる。

[xi] 例えば、システム工学における「理想システム設計」の設計法は、ある問題に対して、それを解決するための理想的なシステムを考え、次に現実に存在するシステムの特徴を整理して、理想的なシステムと現実のシステムとを比較し、その中間に開発すべきシステムを定義する。これは、プラトンのイデア論と弁証法をシステム設計に取り入れたものである。

[xii] ベンサムは、ロックが提唱した民主主義の原則を実践するための原理として、多数決原理の基礎となる功利論を提案した。その根本になる「効用」の概念は、ロックによって提案されたものである。ベンサムは、個々人の効用の総和を最大にすることが、民主主義を実践する方法として最も合理的であるとした。それを応用したのが、「最大多数の最大幸福」を実現すると言われた多数決原理である。アリストテレスの中庸を実践しようとすると、様々な人々の正の効用の総和を計算し、次に負の効用の総和を計算し、それらが釣り合う点を結論として探すことが最も合理的な手順になる。これが、中庸と功利主義的な実践が一致する理由である。

[xiii] カルビニズムの労働倫理は、カルバンの「予定教理」に基づいた「禁欲的労働倫理」と言える。予定教理は、神によって行われる、絶対的な正義に基づいた最後の審判で、「救われる人々」と「救われずに滅ぶ人々」に判別されるので、救われるためには社会全体を神の摂理に基づいて動かさなければならないとする教えである。すなわち、真剣に働くこと

が、救いの道であると考えたのである。この「禁欲的労働倫理」については、清水正徳著『働くことの意味』（岩波新書）に詳細な解説がある。

[xiv] アダム・スミスの伝記作者フィリップソンによれば、スミスは、『諸国民の富』(国富論)と同様に『道徳感情論』の執筆に、長い期間を費やし、出版後も改訂を継続したとのことである。スミスは、人間の経済活動もその他の社会的な活動も、突き詰めれば人間の本性に基づいた社会的な行動であり、その根源をなすものが普遍的な道徳観であると考えていたとのことである。

[xv] この比喩については、人間と労働との関係について議論した清水正徳著『働くことの意味』(岩波新書)の三章「主人と奴隷の弁証法」に詳細な解説がある。

[xvi] 本論文では「高等教育」は、多くの場合大学において実施される教育を指すが、日本の短期大学や高等専門学校、さらに各種専門学校等、高校教育を卒業した人たちが学ぶ教育機関での教育も含むとする。現在の日本では、そのような高等教育機関で学ぶ若者は、全体のほぼ半数と言われている。米国の制度においては、大学とカレッジ、工科系専門大学(Institute of Technology と Polytechnic)などを指す。

[xvii] 1995 年 11 月 2 日付日本経済新聞の連載記事「日本の雇用」⑭によると、日本の就業人口に占める専門・技術職の割合は、1980 年には 10 パーセント以下であったが、1994 年に10パーセントを上回り、2010年には20パーセント近くまでに増加すると予測されていた。この間、ホワイトカラー職の労働者が増加し、ブルーカラー職の労働者が減少することも予測されていた。

[xviii] 米国の心理学者コールバーグは、1971 年人間の道徳性の発展段階が、つぎのような 6 段階に分かれることを示した。すなわち、(1)罰と服従を原則とする段階、(2)(功利主義的)相対主義を原則とする段階、(3)「良い子」原理に基づく人間関係主義を原則とする段階、(4)規則と命令の行動原理を原則とする段階、(5)社会契約と法の原理を原則とする段階、(6)普遍的倫理原理を原則とする段階、の 6 段階である。この 6 段階目は、人間が自分自身を倫理的であるべきとし、その原理に基づいて行動しようとする段階であり、「自己実現」と類似の意味をもつ。

[xix] これは、右利きの人の場合、左脳をよく使うので、左脳が発達すると言う理論に基づいた解釈である。逆に、左利きの人は右脳を使うので、パターン認識など、論理性よりも直感を使う思考に適していると言われている。レオナルド・ダ・ビンチなど有名な画家の多くが左利きであったことは、よく知られている。ミルズの報告によれば、米国の大学に在学していたコンピュータ科学科の学生を対象に調査した結果によると、学生がどのような表現を用いた時に、もっともよく内容を理解し、記憶できるかを実験した場合、話し言葉による表現の復元率が最も高く、次に書き言葉による表現、最後にグラフや図などを利用した表現になった。コンピュータ科学系の学生は、左脳優位な思考方法を持つ人が多いそうである。

[xx]三菱自動車が同社製トラックの開発において、車輪を車軸に固定するハブの設計に問題があることを知りながら、その設計を改善せずに発売に踏み切った。当該トラックが利用されるようになり、日本のいくつかの場所でトラックの車輪が、トラックの車体から外れる、または外れそうになった問題が報告された。しかし、同社は当時の管轄官庁であった運輸省に報告をせず、リコールをしなかった。品質保証部が問題の対応に当たってはいたもの

の、販売した全てのトラックの改修は不可能であった。そのような状況の中で、横浜市内に住む女性がトラックから外れ、転がってきた車輪にひかれて死亡した。当初、同社は設計上の欠陥を否定したが、最終的には設計上の欠陥を認めざるを得ない状況に追い込まれた。

[xxi]日本の技術者の中にも、オープンソース・ソフトウェアの開発に貢献している人材が無いわけではない。例えば、アプリケーション開発言語である Ruby の提案者であり、開発者の一人でもある松本行弘氏(島根県松江市のネットワーク応用通信研究所勤務)がいる。同氏は、汎用性の高い医療系の応用アプリケーション・ソフトウェアの開発者としても著名である。

[xxii] 中村元は、「日本人の思惟方法」(春秋社)の第 4 節非合理主義的傾向(1)非論理的傾向の記述において、日本語のやまと言葉には感情的な精神作用を表現する語彙が豊富であるのに対して、理性的・推理的な能動的思考を表現する語彙が少ないと説明している。特に日本語の特徴として、形容詞から抽象名詞を作り出す方法が確立されなかったことに着目している。その逆の例として、プラトンが、「いかなる」を意味する形容詞ポイオンから、「どのようにあるかと言うこと」を意味する名詞のポイオテースを作り出したと述べている。

[xxiii] アリストテレスは、「形而上学」第 5 巻第 14 章において、ポイオンとは、(1)実体の差別相がどのようなものかを意味し、ポイオテースは実体の差別相のことであり、(2)数学的には数が直線的に並んでいるか、平面的に並んでいるか、さらに立体的に並んでいるかのように構成因子の次元数によって区別される、(3)状態が変化する実体の場合は、実体の変化する属性を言い、(4)徳や悪についても言う、と述べている。

[xxiv] 限界効用とは、数学的な表現を使えば、効用を供給する財の量で微分した値を言う。つまり、財の量が 1 単位増加したときに、効用がどれだけ変化するかを言う。逓減とは、その値が徐々に小さくなることを言う。ここでは、効用を財の量で微分すると、それが負の値になることを言う。つまり、財の量が増加すると、効用は徐々に小さくなる。

[xxv] 管理限界とは、生産されている個々の製品のある特性を測定し、その測定値の平均と分散を計算する。長期間にわたってこのデータ収集を継続すると、工程が正常に稼働している場合の平均値と標準偏差が求まる。この工程が正常に稼働しているときの平均値と、現在の工程で生産されている製品の平均値を比較して、その差が標準偏差の 1.64 倍以上ある時、現在稼働している工程の状態は、普通とは言えない(異常である)。このように、正常な状態にある工程の平均値とその上下に標準偏差の 1.64 倍離れたところに設定できる基準値の上側を、管理限界の上限、下側を管理限界の下限と言う。

[xxvi] これは、1993 年にボルチモアで開催された ICSE(ソフトウェア工学国際会議)のパネル討論における、ベラディからハンフリーへの質問である。

[xxvii] 先験的とは、カントによれば、人間が構築する理論には、経験を蓄積し、それを整理することで得られる知識と、どんなに経験を蓄積し、それを分析しても、そこからは得ることのできない知識がある。その後者の知識を「先験的知識」と呼ぶ。例えば、数学的な空間の概念は、特に無限の広がりを持つ抽象的な空間の概念は、我々が経験から知ることができるひとつひとつの空間(例えば箱)を理解できても、到達できない先験的な概念である。それは、人間が経験上知ることができる広がりは、常に閉じた空間だからである。つまり、人間は経験的に閉じた空間を理解する以前から、空間そのものの概念を知っていたと考えるのが合理的であるとする。そのような先験的な知識や理解を前提とした考え方を観念論

と呼ぶ。

[xxviii] 内包(コノーテーション)と外延(デノーテーション)は、1 つの概念がもつ 2 つの意味の側面を言う。内包は、ある概念によって想起される、その外延から連想可能な意味を言う。外延は、ある概念が直接示している意味である。例えば、「安い」という言葉の外延は、「対象となっているものの価格が低いこと」を意味する。そしてその内包は、「質が悪い」ことを意味する。人間と人間のコミュニケーションで交換される言葉は、この外延と内包を適切に利用しているとき、情報や知識の共有ができる。

[xxix] 形式手法の有名なものに、VDM や Z がある。どちらも、形式的に定義された言語を用いて、システムの仕様や設計を記述する方法である。これによって、厳密なシステム仕様の検証などが可能となる。

[xxx]職務記述書は、英語では job description と呼ばれる。文字通り、仕事がどのようなものであり、どのような人材がその仕事を担当でき、どの程度の給与が支払われるべきかが書かれている。米国の企業では、この職務記述書を最初に作成し、それに基づいて採用活動が実施され、採用後の人事評価も実施される。

[xxxi] 1995 年 10 月 24 日付日本経済新聞の連載記事「日本の雇用」⑥によると、日本企業における従来の賃金体系である年功賃金は、労働に対する対価と言うよりも、従業員の生活を支えるためのライフサイクルに応じた支出に対応するシステムである。このため、労働者の貢献と賃金との間の関係性が弱く、結果として貢献度の低い社員に対する賃金が高くなると言う問題があることを指摘している。特に、平均年収で見ると 20 歳から 24 歳の男子大卒従業員の収入を 100 とした場合、45 歳から 49 歳の従業員の収入は、1984 年で 306、1994 年で 269 であった。年齢の高い従業員の労働コストが著しく高いことが分かる。

[xxxii] これは、ヨーロッパ中世からのギルド制を踏襲したシステムであると言える。西洋社会では、全ての専門的な職業に就くための高等教育は、職業に従事している人々が所属する団体(ギルド)との強い関係を維持している。日本でも、医学部と医師会は同じような関係にある。

[xxxiii]訓練生と訳した英語は、trainee である。例えば、プログラミングを主たる職務とする人々の場合、採用直後の身分は、programmer trainee や software engineer trainee となる。この身分は、見習いであり、完全な意味でのイクゼンプトではない。仮雇用である。訓練生には、指導員(mentor)と呼ばれる先輩社員の監督がつく。

[xxxiv]准技術者と訳した英語は、associate engineer である。例えば、プログラミングを主たる職務とする人々の場合、その身分の呼称は、associate programming engineer や associate software engineer となっていることが多い。また、programming engineer associate や software engineer associate のような呼称もある。

[xxxv]技術者と訳した英語は、engineer である。例えば、プログラミングを主たる職務とする人々の場合、その身分の呼称は、programming engineer や software engineer となっていることが多い。日本企業で言えば、「主任」などの肩書をもつ職位に相当し、プロジェクトではリーダとしての役割を担うことが多い。

[xxxvi]上級技術者と訳した英語は、senior engineer である。例えば、プログラミングを主たる職務とする人々の場合、その身分の呼称は、senior programming engineer や senior

software engineer となっていることが多い。日本企業で言えば、「主幹」などの肩書をもつ職位に相当し、千人の技術系従業員に対して 1 名程度の割合で存在する。

[xxxvii]米国企業のライン管理者の場合、その呼称は担当する部門(グループ)の名前の前後に、その管理者であることを示す manager が付く。例えば、プログラマを多数抱えている部門のライン管理職の場合、manager of program development や manager of software development となる例が多い。

[xxxviii]米国のレイオフ制度は、企業(雇用主)が景気の動向に順応して従業員であるブルーカラー労働者の数を適切に調整する制度である。企業は、好景気になると生産を増強するために採用を増加させる。しかし、不景気になると生産を減少させるために雇用している従業員を一時帰休(レイオフ)させ、生産能力を落とす。この時、レイオフする従業員の対象者は、雇用している従業員の中で最後に採用した従業員からレイオフする。景気の好転で、再度、生産能力の増強が必要になった場合は、最後にレイオフした元従業員から職場復帰の意思を確認し、再雇用する。その意味では、厳密に言えばレイオフは解雇ではない。この 40 年間を見ると、産業構造の変化から、企業構造の再構築(リストラクチャリング)が進展し、一度レイオフした元従業員の仕事が消失し、再雇用の必要性が極端に減少したため、特に最近では、レイオフとリストラ解雇が実質的に同じ意味になっているのが実情である。

[xxxix] これはプロの野球選手やサッカー選手の例を考えると理解しやすい。選手の仕事の内容が職務記述書に書かれており、その仕事が必要とされる期間、選手は所属チームの選手であり続ける。毎年の年俸は、前年度のシーズンの選手の活躍ぶりやチームへの貢献度によって、決定される。選手は、チームの監督が自分をどのように見、評価しているかを日々の試合で理解し、それによって他チームへの移籍を考える。米国企業で働く専門家たちは、それと似たような待遇を受け、似たような考え方で職場を移動している。

[xl]米国社会では、年齢によって従業員の処遇を変えることは、「年齢による差別」と認識される。このため、企業内における従業員の記録には、その従業員の年齢に関する記述を残さないようにしている。つまり、会社は従業員の年齢を知ることができないし、知らないことになっている。これは、あくまで表面的な次元での問題である。実際に、管理者たちは自分の部下たちの実年齢を知っていることが多い。ただ、そのことによって部下の処遇を変えてはならないのである。定年は、その意味では年齢に基づく差別あり、米国社会では正当化できない。そのような社会でも、管理職には定年制が認められている。これは、管理職の仕事が非常に肉体的に過酷なものであると言う認識があり、「若くなくては務まらない」と言う認識が一般的であることによる。個人差はあっても、「60 歳を過ぎて、あの激務に耐えられる人はいない」と考えられている。

[xli]米国では、2009 年にカリフォルニア州で発生したトヨタ車の急加速問題で、日本企業と日本人従業員の労働倫理に以下のような問題があるとする米国誌の報告があった。それを要約すると、(1)日本企業は「臭いものにふた」をする傾向がある、(2)日本企業は危機管理が遅れており事件への対応が鈍く的外れなことが多い、(3)日本では生産者側の利益が消費者の安全よりも優先される傾向がある、(4)日本企業には、品質に関して自己満足に浸っている傾向がある(2010 年 2 月 8 日付ウォール・ストリート・ジャーナル掲載のキングストン氏によるコラム記事)、などとなる。分析内容については、個々に検証する必要があるが、一般の米国市民から見た日本企業の行動パターンにこのような傾向があると認識されている事実について、我々は認識を深めなければならない。結果として、トヨタはこの事故に関係した米国での訴訟の対応に、多額の資金を投入する結果となった。

[xlii]米国に多い敬虔なキリスト教徒の場合、「嘘をつくことは罪である」という認識が強い傾向にある。この場合の罪は、宗教上の罪で、カルビニズムの予定教理により、最後の審判を受ける時、嘘をついた事実があることは、「救われることがない」ことを意味するからである。「すくわれる」ためには、「嘘をついてはならない」のである。

[xliii]日本には、「嘘も方便」と言う考え方もある。良い目的のために嘘をつくことは、方法のひとつであり、適切な仕方であれば、良い結果をもたらす、「良い方法」であり、許されるとする思想がある。特に、日本人にとっては、周囲の人々との人間関係を維持することは、重要な問題であるため、組織の安泰や存続と言う目的のために嘘をつき、組織を守ることは許されるのである。

[xliv]米国で提起されたトヨタの急加速問題は、日本車の設計が米国市場で問題になった典型的な例である。この問題の真の原因が何であったのかは、本論文を執筆している現在でも明らかになってはいない。その原因が自動車の電子スロットルにあったか、それとも他の外的な要因であったかは別として、いずれにせよこの問題が企業としてのトヨタに与えた影響は甚大であった。特に、「日本企業は問題を隠す傾向がある」と言う風評は、このような社会的問題を引き起こす原因となる。

[xlv]経済学者サローは、その著書『富のピラミッド』において、21 世紀の経済発展において、知識の創造が極めて重要な要素であることを指摘し、知的所有権保護法の制度確立が知識主義経済の大前提であることを述べた。

[xlvi]米国の著作権法では、著作権が成立するための条件として、その著作物が保護対象のものであることを明示するために、コピーライトノーティス(copyright notice)を著作物に組込むことが要求されている。それには、それを明示するための宣言を示す©記号、著作権者である個人または企業を識別する文字または記号、そして著作権が発生した年月日などである。

[xlvii] 1985 年に米国大統領に報告されたヤング・レポートの提言に基づき、米国経済の復興のためには、知的財産権の保護強化が重要であるとの視点から、米国特許庁の人員増加、特許裁判所の新設と特許関連問題の法的処理の迅速な処理、通商法や関税法の改正による特許権を侵害する諸外国に対する制裁等の体制整備、等の政策を実施した。この結果、米国の技術輸出は黒字に転換したと報告されている。当時、特許の年間出願数では日本が米国を上回っていたが、最近では米国が日本を上回るようになっている。

[xlviii] 1983 年 7 月 28 日付朝日新聞の社説では、「コピー時代に対応する著作権」との見出しで、その時点での著作権が直面していた問題を解説している。そこでは、貸しレコードや貸しソフトなど、製品を販売する企業が想定していなかった利用を有料サービスとして提供するレンタル業務が出現し、従来の著作権では範疇としていなかった問題が生じていることが述べられていた。それまでの著作権では、そのような一時利用に対して、図書館のような非営利組織によるサービスは存在していた。

[xlix] 1983 年 2 月 28 日付日経エレクトロニクス誌に掲載された米国の弁護士スターンによる「オブジェクト・コードを著作権法で保護できるか」において、同氏はいくつかの事例を解説しながら、プログラマが記述したソースコードを機械的に翻訳して生成されるオブジェクトコードを著作権で保護できるとする立場に対して、反論を述べている。同氏は、ソ

ースコードが、プログラマの思考をコンピュータによって実行可能なオブジェクトコードを生成可能な方法で表現したものであるのに対して、オブジェクトコードは、コンピュータがソースコードを適切な規則に従ってコンピュータが実行できる命令の列に翻訳したものであり、人間が理解することを前提としていないことを問題にしている。このため、2種類のオブジェクトコードが存在する時、それらが全く同じソースコードから生成されたものかどうかは決定できないとする立場である。したがって、違法コピーとは言え、ソースコードが存在しない場合、オブジェクトコードのみからそれを判定することは困難であるとした。この問題については、1983 年 4 月 26 日付日本工業新聞の「ソフトと著作権」の見出し記事において、佐藤丈夫氏が問題の内容を解説している。佐藤氏によれば、米国内では「オブジェクトコードも著作権で保護できる」とする賛成派が多いとしている。

[l] クリーンルーム(clean room)とは、プログラミング業務担当者などが通常の業務実施に当たって、作業を実施している場所とは違い、完全に独立した場所を設置し、その場所でのみ IBM が作成したソースコードや仕様書を閲覧するようにする。この時、閲覧者はノートやパソコンなど、読んだものを記録するための道具を所持して入室してはならない。また、閲覧者の入室に関しては、詳細な入退出時間の記録を取ることが義務付けられている。

[li] 1983 年 10 月 8 日付朝日新聞の社説では、「IBM 事件の厳しい教訓」との見出しで、米国 IBM と日立製作所との間で締結された和解について、企業として情報の収集に努力することは当然であるとしながらも、技術や商品を生み出した人や企業の利益と権利は十分に尊重すべきだとする論説を載せた。その中で、国産メーカが国内市場で過半数の占有率を確保しているのは日本だけであるが、それは IBM が多年にわたって開発・蓄積してきた利用技術をそのまま使う IBM 互換機メーカが大半であるともしている。他人のものを拝借する手っ取り早さをとる日本の風潮に問題があるとしている。その前日の「日立、民事も和解」の見出しの記事で、朝日新聞は、日立製作所が IBM のソフトウェアに関する著作権に直接触れることなしに、事実上著作権を容認したとの報道があった。

[lii] 日本の現在の特許法では、ソフトウェアが従来、物理的に実現されていた機能の実現に利用されている場合、その「機器」の利用目的を限定することで、特許権を認める方針を採用している。このようなソフトウェアに関する特許を「プログラム特許」と呼ぶ。したがって、A の製品で利用されたソフトウェアよりも、効率が良いソフトウェアを考案し、B の製品の実現に利用した場合、そのソフトウェアの実現方法であるアルゴリズムは、日本においては特許による保護の対象になりうる。

[liii] 米国では認められたアマゾンの「ワンクリック特許」は、日本の特許庁においては、紆余曲折を経たが、最終的に認められなかった。ワンクリック特許は、アマゾンで本や商品の発注をしたユーザの決済情報などをデータベースに格納し、次の注文の発注時も、ユーザのログイン情報に基づき、注文情報の一部である決済情報をデータベースから自動的に読みだすことで、ユーザがいちいち入力する手間を省けるようにした、純粋にソフトウェアでの処理を特許化しようとしたものである。

[liv] 1983 年 6 月 12 日付けの朝日新聞では「ソフト保護へ新立法」の見出しで、経済産業省が、特許制度に似た登録制によって、ソフトウェアの使用権、改変権、複製権、配布権を保護するための法案を通常国会に提出する方針を固めたとの報道があった。これによって、著作権法の虚をつく形式で営業されている貸しソフト問題を解決できるとしている。この記事では、文化庁も著作権法の枠内で類似の権利保護を検討しているとの報道もしていた。1983 年 11 月 23 日付けの朝日新聞は、「ソフト保護へ新立法」の見出し記事において、産

業構造審議会情報産業部会のソフトウェア基盤整備小委員会がソフトウェア権法に関する答申案をまとめたことを報じた。その骨子として、開発者の権利保護のため使用権を創設すること、ユーザ保護のため開発したプログラムの内容表示を義務化すること、公的なプログラム登録機関を創設すること、などが解説されていた。同記事では、通商産業省が「ソフトウェアの権利の確立に関する法律」を国会に提出するための立法作業に着手することが報じられていた。

[lv] 高石氏は、1983 年 3 月号のコンピュートピア誌に掲載された「ソフトウェアの法的保護と今後の課題」と題された論文において、世界的に見たソフトウェアの法的保護に関する流れを概観し、当時の著作権を改正し著作権によって法的保護を行うこと提案した。

[lvi] 米国の法律家の中には、「ライセンスは著作権を放棄したことの宣言である」と考えることは正しくないと解釈している専門家も多い。これは、米国マイクロソフトが Linux を Windows の特許を侵害しているとして提訴することを検討していたときに、GNU がマイクロソフトを著作権侵害で逆提訴することを検討していたとの報道を受けての解説であった。マイクロソフトの社員が Linux の開発にも関係していたことから、このような事態が発生したが、最終的にマイクロソフトは、Linux に対しても、またその開発に従事した同社の社員に対しても、譲歩した結果となった。その後ゲーツ氏は、「マイクロソフトのオペレーティング・システムとオープンソース・ソフトウェアは、対立するものではない」とのコメントを発表した。

[lvii] バザールモデルとは、OSS の開発において、開発者コミュニティに参加する様々な記述者達が、それぞれ最善と思うアイデアに基づいたソースコードを作成し、それを組込んだ新しい版のソフトウェアを実現する。他の開発者コミュニティの参加者たちは、それらの提案を評価したり、さらにそれらを改善したソースコードを作成したりする。このような動的なプロセスによって、より良いソフトウェアを実現する枠組みを言う。

[lviii] 第 5 世代コンピュータプロジェクトは、1982 年に通商産業省が開始したプロジェクトで、従来のノイマン型コンピュータから脱却し、人間に近い知的な推論もできる高速コンピュータを開発することを目標として、日本企業 8 社が参加して実施したものである。1983 年 7 月 23 日付の日経産業新聞の「第 5 世代コンピュータ開発計画見直し機運」の見出し記事では、プロジェクト開始 1 年後に、このプロジェクトを推進していた産官学の代表の間に意見の不一致が見られ、産業界から理論重視との批判が出ていることが報じられていた。さらに、1983 年 9 月 9 日付の日刊工業新聞では、「日英共同開発が始動」の見出しで、通商産業省が日英両国において国際研究開発協力が動き出す見通しとなったことが報じられた。また、1983 年 10 月 3 日付けの The Japan Times に掲載された U.S., Japan Race for Artificial Intelligence Computer との見出しの記事で、同年の夏に TBS ブリタニカから出版されたファイゲンバウム著「第 5 世代コンピュータ」がベストセラーになったことを報じている。その記事によると、この国家プロジェクトは 1981 年に 10 年プロジェクトとして開始され、総額で 230 億円を投入し、新設された組織の ICOT を中心に高速な人工知能コンピュータの開発を目指した。

[lix] 1995 年 10 月 27 日付の日本経済新聞に掲載された「CALS」の見出しの記事では、1995 年の 5 月に通産省主導で CALS 技術研究組合と CALS 推進協議会が発足し、コンピュータメーカなどを中心に検討が始まった。このプロジェクトでは、受発注の電子化のための EDI、仕様書などの標準化のための SGML、図面情報の電子化とネットワーク経由での交換のための STEP が研究される予定であった。

[lx] 1998 年 3 月 17 日付けの日本経済新聞の社説では、「企業の競争は規模から知恵に」との見出しで、仮想企業の重要性を述べていた。その記事の中で、CALS の枠組みがネットワーク上で複数の企業を結びつけ、バーチャルコーポレーションを成り立たせる枠組みになると解説した。

[lxi] ドラッカーは、その著書『ネクスト・ソサエティ』において、自由市場が現実には、3 重の構造から構成されていることを指摘し、それらの間に本質的な矛盾があるため、単一のグローバル市場に統合することはできないと述べている。その 3 つの市場とは、国内市場、(複数国の市場を統合する)地域市場、世界的な規模のグローバル市場である。

[lxii]例えば、イギリスの幼稚園では、お絵かきの時間に、ある子供が別の子供と似たようなテーマの絵を描いていると、指導する教員がその子供に、別の子が既に同じテーマの絵を描いているので、別のテーマを考えるように指導するそうである。日本では、同じ教室にいる生徒全員が同じテーマの絵を描くことが普通である。そのようにして、絵を描く技術を競争させていると言ってよい。イギリスでは、むしろ別の絵を描かせ、敢えて競合しないようにして、絵を描く技術の競争にならないように指導していると聞いた。このような幼児からの教育の態度によって、知らず知らずに物事に対する価値観や考え方が育てられていると言える。

[lxiii] 同一労働同一賃金制度への移行は、労働関係の多くの専門家も指摘しているところである。メンバーシップ型効用制度の問題を指摘している濱口氏、『日本の雇用』の著者である島田氏などである。両氏とも、同一労働同一賃金制(島田氏は能力賃金制と呼ぶ)への移行は、年功制や終身雇用制からの同時移行が必要であるとする点で、ほぼ同じ意見である。

[lxiv] ドッカーは、その著書『ネクスト・ソサエティ』において、ポスト資本主義社会における人材の重要性について述べている。そこでは、知識労働に携わる労働者が競争力の源泉であり、その人材をいかに効果的に活用できるかが問題になることが指摘されている。

[lxv] 情報処理技術者試験を実施している独立行政法人情報処理推進機構の報告によれば、平成 26 年度(2014 年度)春の試験における基本編情報処理技術者試験の合格率は、理工学系学部に所属する大学生の合格率は、29.5 パーセントであり、文科系学部に所属する大学生の合格率は、27.3 パーセントである。これに対して、情報系の専門学科に所属する学生の合格率は、理工系学部で 25.3 パーセント、文系学部で 22.6 パーセントと、若干、低くなっている。情報系専門の学科に所属する学生では、受験時に学生が所属する年次がやや低いことが原因であろう。

[lxvi] コンピテンシ(competency)とは、人間が職務に就き、その職務を全うしようとしているとき、その成果に最も影響する要因は何かについて研究した結果、最も重要なものが、業務に関する知識ではなく、自分に与えられた責任に関する認識、相手の立場に立って物事を考えられる態度、自分の価値観と相手の価値観の違いを認識しその違いを克服できる解決策を探そうと努力する姿勢など、一般的な知識よりも生きる姿勢や価値観であることが理解された。これらの要素を総称して、コンピテンシと呼ぶ。コンピテンシは、高等教育で教えられる知識ではなく、人生の早い時点で獲得される行動のパターンであると言われる。日本語では、「行動特性」と訳される。最近の日本企業では、このコンピテンシとジェネリックスキルが混同されて、コンピテンシとして理解されている例も多い。ジェネリックスキルは、スキル(何かをするための技能)であり、高等教育で訓練することも可能である。

これに対して、コンピテンシは、高等教育では教育したり訓練しても身に着かないことが多い。高等教育以前の全人格的な教育や訓練によって身に着くものである。