

ソフトウェア開発を理解してもらうためには

漆畑 晶



© 2019 URUSHIBATA Akira (Licensed under CC BY 4.0)

本作品は CC BY 4.0 ライセンスによって許諾されています。ライセンスの詳細な内容は <https://creativecommons.org/licenses/by/4.0/deed.ja> でご確認ください。

ソフトウェア開発を理解してもらうには

情報技術者は開発の仕事が実際にどういうものであるのか、一般人に伝える努力をあまりして来ませんでした。その結果、人々は自分達が日常利用しているソフトウェアについてよく理解できないでいます。本稿では言語、中国古典、経済思想に触れながら情報技術の具体像を知らない人向けにどのように語るべきかについて論じます。

無料のソフトウェアに支えられるアプリケーション開発

オープンソースはタダだと思われて悩んでいませんか？

- オープンソースカンファレンス名古屋で掲示された来場者からの質問（2019年7月）

オープンソースソフトウェアは無料で入手できます。この「無料」というのはたいへん大きな問題をはらんでいます。私達技術者は日頃、無料でダウンロードできる道具や部品を使って応用ソフトを作っています。その品質はどうなるでしょう？専門技術者と一般人では捉え方が大きく異なります。技術者は信頼できる物を選んで使うようにしていますが、実情を知らない一般人は極端に安価な道具部品は当然品質劣悪、それを用いて作った応用製品も品質が悪いはずと思うものです。一般人にそう思われたくないの、多くの開発者は自分達が仕事に何を使っているか、詳しく語らないようにしています。コンピュータが行き渡り、生活のあらゆる面で使われている当今、その技術の基底に何があるのか、多くの人が誤ったイメージを抱えています。これは危うい社会状況です。

価値が高い物は値段が高く、価値が乏しい物は値段が低いと思われがちですが、そうとは限りません。まずこのことを理解しておく必要があります。

無料で良い物が手に入ることを日本語で「有り難い」と表現します。「ありがとう」は「お礼」と呼ばれます。東アジアでは「礼」を社会の安定の基礎とする考えが古くからあります。特に教育において礼は重要だとされています。

孟子の有名な言葉に「辞讓の心、礼の端（はじまり）なり」があります。開発者が敢えて知的所有権の主張を控えることがあります。これは他者との協力と教育に対する配慮から行われるものです。「自分は先人の技術を自由に使わせてもらっている、自分が開発したものを後輩が使うのを制限するのはおかしい」と言う人がいます。

近年の情報技術の急速な発展の背景にはこの孟子の教えに通ずる思想がありますが、これを理解している人は少数です。

ダイヤモンドと水： 価値のパラドクス

価値のあるものが必ずしも高価とは限らないというのは東アジアの伝統思想だけでなく、西欧近代の経済学の中でも指摘されている事実です。

近代経済学の父とされるアダム・スミスの名著「国富論」に「ダイヤモンドは何にも役立たないけれども多くの物と交換できる。水は命に欠かせないが水と交換でほとんど何も手に入らない」と指摘しています。これは経済学で重要な論点で、後の学者は「価値のパラドクス」 *paradox of value* と呼んでいます。

現実の経済学ではお金で価値を測定できる現象だけを対象とする研究が多いものです。その種の研究は価値のパラドクスを視野に入れていません。しかし社会の難問には価値のパラドクスが関係している場合があるとされています。（* 1）

プログラミング言語の役割

プログラミングの経験の無い人はその作業がどのようなものか、想像しにくいものです。「わらいものはわからない」と謙虚に認める人もいます。一方で知らないことについて勝手な想像をする人もいます。最近ではスマートフォンが普及し、幼い子も操作します。これを見て、コンピュータを思い通りに動かすとは機械を少しいじる程度と勘違いしている人がいます。東京オリンピックの暑さ対策として夏時間を導入しようという案が発表され「日本人は几帳面だからコンピュータの設定変更も問題なく為し遂げるだろう」と発言した政治家がいました。切り替えの日に家庭の時計を調整する程度の作業であらゆるコンピュータの夏時間対応ができると思っ込んでいたのです。一般人にはこの程度の理解の人が少なからずいることを意識する必要があります。

プログラミングとはどのような作業か、一般の人にイメージしてもらうのには「数学の証明に似ている」と言うのが有効です。数学の問題には内容は平易でも証明が難しいものもあります。また証明は緻密でなければなりません。一見小さな狂いでも結果に大きな影響が出てしまうものです。数学で分母がゼロのケースを考慮するのを忘れてしまいますが、プログラムでもこの類はしばしばあることです。たまにその例外的条件が生じた時にプログラムは異常作動します。こうした際はウイルスに付け込まれる可能性があります。

一般人向けのソフトウェアはアイコンをクリックするなど、絵を操作するものが主流ですが、プログラミングは文字で行うのが普通です。ワープロで文章を作成し、必要に応じて訂正していく作業に似ています。マウスを使う場合もありますが、作業効率の観点からはなるべくマウス操作が生じないようなやり方が良いと経験の長い人は言います。マウス操作は以前の操作を再現できないという問題もあります。

コンピュータの正確な動作には正確なプログラムが必要です。コンピュータは文系と理系の別では理系とされていますが、プログラムはコンピュータ言語で記述されます。このコンピュータ言語で書かれたプログラムをコンピュータの理解できる実行形式（見方によってはまた別のコンピュータ言語）に変換する道具はソフトウェア開発においてたいへん重要なものです。（*2）コンピュータ言語は言語の側面があり、その考察には文系の発想が必要です。

コンピュータ言語は人造言語です。英語や日本語などは人造言語に対して自然言語と呼ばれます。まず日本で使われている文字、漢字と仮名を考えてみましょう。漢字は真名（まな）とも呼ばれます。仮名は表音文字ですが、漢字は表意文字です。両者を比べると、仮名の方が幼稚な感じを与えます。文章では漢字表記が推奨されますが、それは表音文字より表意文字の方が意味の誤解が起きにくく、正確な意思疎通ができるからです。

英語を始めとするヨーロッパの言語は表意文字であるアルファベットで記述されています。アルファベットに加えて、数字や算術記号も使われます。これらは実は表意記号です。3は英語では *three*、フランス語では *trois*、ドイツ語では *drei*、ハンガリー語では *három* と発音は違って意味するところは同じです。これは表意文字の特徴です。数学では様々な記号の操作が行われますが、それぞれの定義は厳密です。プログラムは主に数と数学記号で書かれていますが、やはり厳密性があります。

一般にシステム開発のプロジェクトは多数の人が協力して作業をします。人が機械に命令するだけでなく、人と人が意思疎通を図りながら進めます。システム全体を構成する部品ごとに、さらにその部品の部分ごとに動作を明記するドキュメントが作成され、参照されます。ドキュメントは日本語や英語など、人の言葉で書かれます。

プロジェクトが混乱してくるとプログラムとドキュメントとの不一致が生じてきます。混乱が深刻化するとドキュメントが全く当てにならなくなります。ドキュメントが当てにならない状況では、プログラムが多様な条件下どう動作するのかいちいち試しながら作業を進めることとなります。これは効率の悪いやり方です。またドキュメントが不完全な部分はどう動作するのが正常かが決まっておらずテストはできません。ドキュメントに不備のあるプロジェクトでは、テスト不足による不具合が多くなります。

このように一定水準の品質のソフトウェアを作り出すのにはプログラム、ドキュメント双方が正確に記述される必要があります。よい技術者になるには物事を正確明解に記述する表現力が必要です。

文章による表現力は学校教育で長い年月をかけて養成されるものです。名作文学を熟読する、文法を意識する、さらには字源語源に興味を持つなどの方法が有効とされ、広く行われています。日本では外国語として英語を学びますが、表現力の育成が重視されていないとの批判が聞かれます。オープンソースでは国際的な開発プロジェクトが多数ありますが、日本のプログラマは余り参加していません。英語が壁となっているのが一因です。日本で教育される外国語は英語だけではありません。漢文は古代中国語、外国語とも言えます。漢字の正しい使い方を知るには中国の古典を学ぶのが古来良い方法とされてきました。英語の教育と違ったところに重点が置かれています。しかし理系を志す学生には漢文は自分の専門とは無関係なものと思って目を向けず、学ぶ意義を理解せずに大学に進学している者が多いものです。

言葉を保持する手段としての詩

情報産業の振興には知的著作権保護が重要だと言われますが、その中で全く論じられないのが詩です。

詩とは何でしょう。字義から考えます。左の「言」は自明ですが、右の「寺」は何でしょう。宗教に関係ありそうですが、実はそうではありません。太古「寺」は「持つ」という意味でした。漢の時代「物事が執行される所」ということで「寺」は「官庁」「庁舎」の意味で使われます。そのお役所の一つ、外務省の建物の迎賓館に外国から来た仏教の僧侶が滞在した縁で「寺」は今のように「仏教寺院」を意味するようになります。「寺」が本来の意味から離れているので「もつ」は手偏をつけた漢字「持」で表現します。

詩とはまさに「言葉」を「持つ」手段なのです。字が発明されるまでは暦、工芸、医療、歴史などあらゆる知識は言葉の暗記と口伝によって継承されていました。覚えやすくするために、余計な言葉を徹底的に取り払い、韻をつけるなど調子を整える工夫をしました。（*3）

活字印刷が普及して口伝が減るとともに詩の重要性は下がって行きました。

現代において詩は感情の表現の手段とされています。これはなぜでしょうか。感情を表現するのは難しいものです。人は感情に押されると、言葉がもつれたりして、物事を上手に伝えられなくなるものです。心の内を整理して五七五七七等、一定の形式で表現するとうまく伝えられるものと昔から知られています。詩で感情を表現するとは、心と向き合い、自分の抱えている問題を整理して解決の緒を探る作業なのです。心の中を整理して初めて、他人の協力を請うことができるようになるのです。

子曰く「詩に興り、礼に立ち、楽に成る」これは「論語」の一節です。学ぶ順番を説いていると言われます。孔子は学問は最初に詩を学ぶべきだとの立場です。

品質の高いプログラムを作成するには無駄な物を省き、簡潔に表現する才能が不可欠です。これは詩を作る才能と同じです。理論の流れに無駄があってもプログラムは一応動作します。記憶素子が安価な今日、肥大化したプログラムは気になりません。それどころか、大きいと機能が豊富で良さそうだと消費者は思ってくれるものです。迂遠で無駄な動作を指示されてもコンピュータは不平を言いません。しかし整理されていないプログラムはエネルギーを浪費します。そして整理されていない、もつれた部分が保守や機能拡張を難しいものにします。

優秀なプログラムを求めるなら詩的才能に着目することです。「これこれの機能を持つプログラムを書いた。こんな機能を追加した」と自慢するのは並のプログラマです。優秀なプログラムは余計な物を除去し、論理を整理し、プログラムを小さくした実績を持っているものです。

合成の誤謬

「合成の誤謬」は経済学で重要な概念です。経済政策論（マクロ経済理論）で特に重要です。英語では *fallacy of composition* と言います。深刻な不況を最近ではデフレーション（デフレ）と呼びますが、その打開の鍵となると言われています。

リチャード・ストールマン博士は1984年に誰でも自由に利用でき、誰でもプログラムを閲覧できる基本ソフトOSの開発を始めました。（*4）その成果は広く利用されています。彼は「自由ソフトウェアの運動を推進するにあたって、何が一番困難でしたか？」と問われて「企業はマイクロソフトのソフトウェアを使う。採用する社員が学生時代それを学んでいるから、と称して。学校はマイクロソフトのソフトウェアを教える。企業でそれを使うからと称して。双方が消極的になっていて、中々打開できない」と答えています。企業も学校も単独なら合理的と言える行為ですが、社会全体としてみると、合理的とは言えません。こうした状況を経済学では「合成の誤謬」と呼びます。

オープンソースソフトウェアでは合成の誤謬に注意する必要があります。「フリーソフトへのタダ乗り」は合成の誤謬の典型的な例と言えます。使うだけ使ってお金は払わない、骨の折れる貢献は一切しないというのが個人としては合理的な行為となります。しかし皆がそうしたら全体が成り立ちません。

自分達が日頃使っている開発に使っているソフトウェアは実は無料で入手したものだとして人に告げないのも、個人のレベルでは合理的であっても、積み重なると社会がソフトウェア開発の現実を理解してくれないという望ましくない事態が現出します。これも合成の誤謬です。

渋沢栄一の「論語と算盤」

渋沢栄一は多くの企業を興した実業家で「日本の資本主義の父」とも呼ばれています。数年後には1万円紙幣に彼の肖像が使われる予定です。

渋沢栄一は企業を創立しただけでなく、近代的営利企業がどうあるべきかの思想を唱えました。この業績の方が重要と言えます。江戸時代には「士農工商」の身分制度がありました。工業商業は卑しいとの見方が人々、特にかつて武士だった者に根強くありました。欧米の先進諸国の強さは工業商業の盛況に支えられていると見た渋沢はこれを改めようと「土魂商才」を唱えます。彼の努力のお陰で商工業に対する悪いイメージは払拭されるようになります。

渋沢は「論語」を愛読し人生の標としていました。漢文に明るく、正確な文書を書くのが上手で、道徳を重んずる人が実業界に入り、企業の規模拡大と経営の安定に寄与して行きました。

渋沢が晩年に著した「論語と算盤」は上述の合成の誤謬についても論及があります。「論語」を読み、理解する人は社会を客観視し、合成の誤謬が起きていたらそれを認識し、対策を唱えて指導力を発揮すると彼は考えたのでしょう。

一方で渋沢には批判もあります。営利事業を発展させたいとの強い心情から「論語」の思想からは遠い人を要職に登用したこともあったようです。新しい1万円札の肖像の話がニュースになるや、韓国から苦情が来しました。日本では知られていませんでしたが、朝鮮半島では利益重視でモラルを軽視する者の活躍が目立ち、渋沢はそのリーダー格と想われて来たようです。

渋沢の負の面としてはもう一つ、明治維新、士族解体とともに終わったはずの身分制度を形を変えて復活させてしまったことが挙げられます。大会社の管理職、官僚、学者が江戸時代の武士のような身分意識を持つようになりました。

この身分意識は知的集約産業である情報産業において深刻な問題をもたらします。「学びて時に之を習う」に始まる「論語」には学問は人格を育てるとの思想があります。さて、プログラマーになるには多く学ぶことが必要ですから、プログラマーは人格者ということになりそうですが、実際にはそのように思われていないと思います。それはなぜでしょうか？一方には情報産業には技術情報の秘匿を通じて実際以上に物事を難しく装う人たちがいて、また一方には情報産業の外の地位のある人たちの中に身分への思い入れが強く「論語」本来の思想を軽視し、身分が人格を作るような考えに傾倒し、あわせて地位の低い他者の学びは自分のして来た学びと違い努力を要しない、人格形成に寄与しないものであると思いつく人がいるからです。簡単に言うと前者は「専門用語で人を煙に巻く不屈者」、後者は「人の貴賤を学びに投影する差別主義者」です。後者の態度はプログラミング、そこで行われる

協力や教育の理解を阻害するものです。しかも厄介なことにこの予断はなかなか打開しにくいものです。

一般社会にどう働きかけるか

ここまでお話ししてきた内容は従来語られたことのない、新しいことが多く人に伝えるのは難しいと感じるでしょう。人に語る場合の方針としては、自分たち情報技術の専門家の事情を一般人に理解してもらおうとするより、人の仕事を助けようとするのが有効になると思います。

2019年7月初めにセブンイレブン社がキャッシュレス支払いサービス7Payを導入したところ、認証手順の不備を突いて数千万円の不正利用が発生、導入4日目にて利用停止となりました。主催したセブンイレブン社のみならず、キャッシュレス支払いに対する人々の信用期待を粉砕する社会的影響の大きい事件でした。7Payを開発した7Pay社の社長が会見で「二段階認証」が何かを知らず困惑する場面が度々テレビで放映され、経営陣の技術知識不足が問題の背景にあったとの見方が広まりました。このニュースを見た様々な会社の情報担当の管理職には不安になった人たちがいるはずです。新しい情報技術がどんどん導入され、技術の概略も理解できずにいる中、いずれ自分が責任を問われる領域で事故が起きるのではないかと懸念が生じていることでしょう。この立場の人を見かけたらソフトウェアとその開発の実際を話してあげようと提起することが可能です。情報技術は専門外の人には理解困難とされますが、7Payの不祥事が契機となって腰を据えて調査してみようとの機運が生じていると思います。これまで広く一般向けに語られた事柄から大きく乖離することでも聞いてもらえる状況ではないでしょうか。

学校でプログラミングを教えるようになっていっていますが、やはり経験のないことから不安に感じている教師がいるはず。そういう人にプログラミングの実際を伝えてあげたら、喜ばれるはず。教師の知り合いがいたら、数学や英語、国語を学ぶことにどんな意義があるのか議論すると良いでしょう。

1万円札に渋沢栄一の肖像が使われるということで「論語と算盤」を読む人が増えています。「論語と算盤」を数頁読むと「論語」も読むべきだとの認識に至ります。現在のソフトウェアの開発における協力の思想にはこの「論語」に代表される儒教の教義と類似が見られます。渋沢に関心を持つ人たちに、彼が尊重した孔子孟子の思想を現実の世の中に応用したようなものとしてソフトウェア開発の話をするというのも良いと思います。

注

* 1： 公害や魚の乱獲がその例です。

* 2： マイクロソフト社の創業者ゲイツ氏はここに目をつけていて、同社の最初の製品はBASICという入門者向け言語を解釈して機械が実行できるようにするプログラムでした。

* 3： 現在でも本質的には詩であるのに、そうとは呼ばれないものはいくつかあります。コマーシャルのキャッチ・コピーがその一例です。

* 4： ストールマン博士は4つの自由：「ソフトウェアを実行する自由、研究して改変する自由、再配布する自由、改変した物を再配布する自由」が不可欠と説きました。ストールマン博士の主張はオープンソースの思想の多くの出発点ですが、彼は「オープンソース」という呼称を「自由が十分尊重されていない」として拒絶し続けています。思想の創始者が誰であるのかははっきりしない状態は思想から人を遠ざける作用があります。思想が軽視されるなか、技術の成果だけ享受しようという人は増える一方です。

2019年8月 漆畑晶

著者紹介

漆畑 晶

プロフィール

- 画像処理ソフトウェアユーティリティ群 Netpbm の主要開発者
- FSF 創設者のリチャード・ストールマン博士と親交あり
- 中国の大学（上海復旦大学、南京大学、天津大学、中国科学院大学等）で講演多数

信州大学経済学部在学中に「国富論」やアメリカ産業史の本に加えて、「論語」「老子」などの古典を読んで引用する力を磨く一方で、R 言語による多変量回帰分析プログラムや、駅すぱーとの核心部である最短距離経路探索のプログラムを自作していました。

オープンソース・ソフトウェアについて

なにか良いものをタダで貰ったり使わせてもらったときは、必ずお礼を言うものです。お礼を欠いていると、おかしいことになってしまいます。誰でも自由に利用できるオープンソース・ソフトウェアも同じです。感謝を欠かしてはならないのです。

ソフトウェア開発を理解してもらうためには

著 者：漆畑 晶

発行者：（特非）オープンソースソフトウェア協会

<http://www.ossaj.org/>

info@ossaj.org

発行日：2024年2月1日



© 2019 URUSHIBATA Akira (Licensed under CC BY 4.0)

本作品は CC BY 4.0 ライセンスによって許諾されています。ライセンスの詳細な内容は <https://creativecommons.org/licenses/by/4.0/deed.ja> でご確認ください。