

第 43 章 ピープルウェア

「ピープルウェア」とは何か

「ピープルウェア」とは、あるいは聞き慣れない言葉かもしれない。

情報システムを構成する要素には、ハードウェアとソフトウェアがある。しかしこの 2 つに加えて、人間の要素も無視するわけにはいかない。その 3 つ目の人間の要素を真剣に考えてみようではないか。そういう意図で、トム・デマルコ (Tom DeMarco) とティモシー・リスター (Timothy Lister) が作り出した「造語」である。聞き慣れない人がいても、不思議はない。

彼ら 2 人が 1987 年に、この聞き慣れない言葉を書名にして 1 冊の本を書いた。私は、この本はこの領域での名著の 1 つと評価している。これ以前にソフトウェア開発に関わる人の問題を論じた本は、多分ジェラルド・ワインバーグの「プログラミングの心理学」[WEI71]だけだったような気がする。この本も私は名著と評価しているが、トム・デマルコとティモシー・リスターはワインバーグよりもっと広くて高い観点から、ソフトウェア開発に関わる人の問題を取り上げた。そして彼らは、1999 年に米国でこの本の第 2 版を、2013 年に第 3 版を、それぞれ出版した (第 3 版の日本語の翻訳版は、やはり 2013 年に出版された) [DEM87]。

たいへんおもしろいことにこの第 2 版と第 3 版では、彼らは初版の部分にはほとんど手を入れずにそのまま採録し、さらに一部の原稿を付け加えて新しい版にすることをしている。このことは、初版の内容がたいへんに充実していて、素晴らしいものであった証拠と私は受けとめている。

この章ではこの本 (第 3 版) をベースにして、私自身の経験も交えてソフトウェア開発に関わる人の問題を考えてみたい。

ピープルウェアの 4 つの話題

初版のピープルウェアでは、4 つの話題が議論されていた [DEM87]。それは以下の項目である。

- ① ソフトウェア技術者の管理に関わるテーマ
- ② ソフトウェア技術者の執務環境と開発の生産性に関わるテーマ
- ③ 優秀な人材を集めることについて
- ④ チーム作りについて

以下で、それぞれのテーマについて彼らの議論を紹介してみたい。

ソフトウェア技術者の管理について

オープン・ソース・ソフトウェアについて議論した章¹で、「人はなぜ働くのか」という問題を考えた。そこでは、「人は雇用関係や金銭的な契約に基づく『強制力』によって動く」という人の労働の動機についての昔からの「原理」が、ソフトウェア技術者の世界では成立しなくなったことを見た [UME07]。

つまりオープン・ソース・ソフトウェアの開発に参画している時、若くて優秀なソフトウェア技術者たちは作業の成果を他の人たちに認めてもらい、高い評価を得ることを目的に働いている。当然無償である。彼らにとってこのような形で働くことは決して苦痛ではなく、自己を表現する絶好の機会であり、大きな楽しみですらある。このような時、このソフトウェア技術

¹ オープン・ソース・ソフトウェアについての議論は、第 29 章で行った。

者を「管理」する必要は全くない。彼／彼女は自発的、自律的に働き、十分な成果を上げる。

しかしこれは何も、オープン・ソース・ソフトウェアの開発に参画している時だけの話ではない。日常の仕事としてのソフトウェア開発でも、ソフトウェア技術者はそのような楽しみを多く見出し、仕事をするを楽しんでいる。ここに「仕事」というものについて、少なくとも一般的な肉体労働とソフトウェア開発との間には、大きな開きがある。

私も現役のプログラマだった頃、このように仕事をしてきた。プログラミングという仕事をする事は、楽しみだった。もちろんストレスや苦勞がなかったとはいえないけれど、苦痛などは全くなかった。楽しみながら給料がもらえて、こんなに素晴らしいことはないと考えていた。だから大学の教員になった後、就職活動をしていた学生には、「好きで、お金を払ってでもやりたいことは何かを自分でよく考えてみなさい。そしてそれができる仕事に就きなさい。長い人生を楽しみながら仕事をするのができたら、こんなに素晴らしいことはない」と話していた。

これが一般の管理者にはなかなか理解できないようで、そこから「ピープルウェア」に書かれている『「管理』とは人の尻を蹴飛ばすこと』にまつわる、いろんな悲喜劇が生まれる。

「ピープルウェア」の中には、順調に開発が進んで、予定の期日にカットオーバーできるという報告を受けた管理者がその期日の繰り上げを命じたという話がある。「管理者の仕事とは、場合によればそれが無意味なものであったとしても、常に部下に圧力をかけ続けること」と、その管理者は考えていたのだろうか。

個人的な話で恐縮だが、私にも似たような経験がある。

私は会社勤めをしていた一時期、あるアプリケーション・システムの仕様を決め、テストでその結果を確認する課の課長だった時があった。コンピュータ・センターは本社から離れた場所にあり、私の課員がスムーズに仕事をするためには、その全員がコンピュータ・センターに執務場所を移した方が都合が良かった。それで私はコンピュータ・センターの一部を我々の執務場所として借りるように手配し、部長にその承認を求めた。部長は、反対した。彼の目の前から彼が管理すべき集団の一部がいなくなることを、彼は拒んだのである。彼にとっては仕事の能率より、彼の目の前で全ての部下が仕事をしているという事実の方が大切だったのである。あるいは彼は、自分の見えるところ以外の場所では、「全ての部下は遊んでいて仕事をしないもの」と考えていたのかもしれない。

「自分の見えるところ以外の場所では、全ての部下は遊んでいて仕事をしないもの」と考えていたと思わせるような別の話もある。他のプロジェクトの立ち上げ時点で、キックオフ・ミーティングと JAD²（ジョイント・アプリケーション・ディベロップメント）を兼ねて、箱根にあった会社の保養所に行って 1 週間仕事をしたいと申し出た時も、彼から反対された。彼は、箱根ではみんな朝から温泉に入り、ゴルフをし、場合によれば朝酒を飲み、全く仕事をしないものと考えていたに違いない。

私がまだ現役のプログラマだった時は、常に「朝の 9 時から夕方 5 時まで自分の席に座っていること」を期待されていた。仕事をしている／していないは、全く問われなかった。とにかく、居れば良かった。居ることが、仕事だった。デバッグのための徹夜の後でも、それを求められた。そうすることで疲労の回復が遅れ、後の仕事に支障をきたすといった配慮は全くなかった。

私が管理者になった時私は、自分の部下に「9 時から 5 時まで単に席にいること」を求めず、

² JAD については、第 21 章で論じた。

その代わりに効率よく仕事を進めることを求めた。会社のルール違反を指摘されたら、全ての責任を私が被るつもりだった。しかし幸いなことに、そのような摘発はなかった。

デマルコなどが「ピープルウェア」の最初で取り上げているように、私の周りでもソフトウェア技術者の管理にはいろいろと問題があった。今は、少しは改善されてきているのだろうか。そうであってくれることを、期待している。

ソフトウェア技術者の執務環境と開発の生産性について

「ソフトウェア技術者に良い執務環境を与えると、開発の生産性が良くなる」という法則を見つけ、公表したのも、彼らのこの本が最初だった。

彼らは「プログラミング・コンテスト」という、組織単位でのプログラミングの生産性を競うコンテストを継続的に実施している時に、ある組織から来たプログラマは常に他の組織からのプログラマよりも生産性が高いことに気付いた。その理由を調べた結果、プログラミングの生産性が高い組織では、プログラマの執務環境が優れていることを発見した。

単にプログラミングの時だけでなく、ソフトウェア技術者が仕事をしている時、最も重要なものは「集中した状態を維持する」ことである。プログラムのアルゴリズムを組み立てている時、デバッグをしている時などに、大きな集中力が要求される。良い執務環境では、ソフトウェア技術者は集中することと、その状態を維持することがより容易になる。

しかし私が現役のプログラマだった時、こんな「法則」はまだ認知されてなくて、たいへん悪い執務環境の中で我々は仕事をしていた。私の周りで人が歩き回り、人の話し声がし、電話のベルが常に鳴っていた。「ピープルウェア」の中に、「プログラムは夜できる」という話がある。「夜は職場が静かで、集中できるから」というのが、その理由である。この本のその部分は、私にはまさに身につまされる話だった。

この本を契機に、日本でも多くのソフトウェア会社はソフトウェア技術者の執務環境を良くすることに力を入れ始めた。アメリカのように個室を用意したり、あるいは一部屋に2人程度を割り当てるということは日本ではまだ無理だが、ソフトウェア技術者の席の間に間仕切りを立てて、周囲の状況が目に入らないようにする形が今では多く採用されている。日本だけでなく、私が最近仕事でよく来ているミャンマーでも、経営者の意識の高いソフトウェア会社ではこの間仕切りを導入している。個室に比べると効果は落ちるだろうが、それでも私の頃と比較すると大きな前進といえる。

しかし私の経験からすると、人はどんな環境でも集中することができるようにはなる。私は若い頃、劣悪な執務環境の中でも集中する術を学んだ。これはそれ以降、たいへん役に立った。今でも役に立っている。しかしそうはいっても、執務環境は良いに越したことはない。私が会社勤めから大学の教員に替わって何が一番うれしかったかといえば、大学では「研究室」と称する個室がもたらされたことだった。

どんな環境でも集中できるようにはなるが、環境が悪くないと集中するまでに時間がかかり、デマルコなども書いているように覚醒させられるのも早い。そして一度覚醒すると、再度集中するまでに時間がかかる。やはりソフトウェア技術者にとって、執務環境が良いことは必要である。

まだ私が現役のプログラマだった時、私の上司で「ニコボン（何も用事がないのに肩をポンとたたいて、顔を見てニコニコして、一言二言話しかけて通り過ぎること）」をして回る人がいた。私はこの「ニコボン」で、何度も集中状態から覚醒させられた。私だけでなく、私の同僚

の多くのプログラマが、彼のニコボンで被害を受けていた。ある時私は、彼にそれを止めてくれるよう頼んだことがある。この話を聞いて、彼はたいへん驚いたようだった。ソフトウェアのバックグラウンドを全く持っていなかった彼にとって、「ニコボン」がソフトウェア技術者である部下との間で良い関係を保つ彼一流の、あるいは唯一の方法だったのだろう。それ以降彼は我々に「ニコボン」することを止めたが、ここにも肉体労働者とソフトウェア技術者との管理の仕方についての違いの 1 つがある。

優秀な人材を集めることについて

仕事でもスポーツでも、優秀な人を集めて十分に教育／訓練をし、さらにそれらの人たちを素晴らしいチームに仕立て上げると、その仕事やスポーツでの成功の可能性が高くなる。これは自明の理であり、ソフトウェア開発でも同じことがいえる。「ピープルウェア」には、このソフトウェア開発で優秀な人を集める方法についての彼らの提案が書かれている。

彼らが提案している方法の 1 つは、プログラマを採用する時、その人とチームを組んで一緒に仕事をするようになる、既にその組織で働いているプログラマを面接に参加させて、「一緒に仕事をできる人かどうか」を判断させて、その上で採用するのがよい、というものである。私は、この方法をそのまま適用したことはない。

しかし私が管理者を務めていた時に、私が働いていた会社の人事部はソフトウェア技術者の採用に当たって、私をよく面接の担当に割り当てた。私は新卒者の面接は苦手だったので、既卒者の採用に重点的に割り当てて貰うよう人事部に依頼していた。その既卒の応募者と面接する時の私の基準の 1 つは、「私の部下として、私は彼／彼女と一緒に仕事ができるか」というものだった。

その結果、たいへん素晴らしい人を何人も採用することができた。その人たちの中には、今でも当時私が働いていた会社にそのまま留まり、役員になってまだ大活躍してくれている人がいる。その人たちの、さらなる活躍を期待したい。

チーム作りについて

チームとは、単に何人かの人が集まっただけのものではない。何人かのそれぞれの人の力を集めたもの以上のものが、チームである。

チームでは、何らかの方法でそれぞれの人のチーム内での役割が明確になり、みんなが協力し合い、弱いところを補い合い、力を合わせて目標に立ち向かうという土壌が形成されなければならない。それを「チームワーク」と呼ぶ。チームワークが形成されていないチームは、単なる人の集合であって、「チーム」とは呼べない。

それでは、どうすればチームワークが形成されるようになるのだろうか。「ピープルウェア」の中で彼らは、ある女性管理者に指名された人の話を紹介している。彼女は最初、部下になることに決まった人をディナーに招待する。しかし彼女は、飲み物を用意するだけで後は何もしていない。人は集まって、彼女を中心にして雑談を始める。そのうちに彼女は、皆で何かを作って食べようと提案し、ディナーの材料を買いに全員で近くのスーパーに買い物に出かける。スパゲッティを作ることにし、そのための材料をそろえるが、彼女が主導的に動くわけではない。みんなの自由にさせている。帰ってきて料理を作る際も、後片付けも同様である。その結果集まった人たちは、最初はいささか居心地の悪さを感じるかもしれないが、そのうちに全体の中での自分の持ち場を見つけて、全員で夕食を作って、全員でそれを楽しむという話である。こ

のやり方は、素晴らしい方法と私も考える。彼女の人柄／人との接し方とこのディナーの結果で、彼女のチームは最初から素晴らしいチームになっただろう。

「ピープルウェア」の中には、著者の 2 人が「結束の強いチーム (jelled team)」と呼ぶ、もっと素晴らしいチームの話が書かれている。これがどのようなチームなのかを、文章で説明することは難しい。しかしこのようなチームの一員になって仕事をすると、その時はたいへん幸せであり、結果として非常に良い結果が出る。そして、生涯忘れられない思い出が残る。私もこれまでのソフトウェア開発で、このようなチーム作りに参画し、そのチームの中で仕事をした経験を 2 回持っている。

どうすればこのようなチームができるのかは、私には分からない。この本の著者も、それは書いていない。多分いくつかの要素が複合的に影響し合い、その結果として自然発生的に、徐々に生まれてくるものなのだろう。1 ついえることは、目標の達成がむずかしいほどチームの結束力は強くなる、ということがいえるらしい。軍隊で命をかけて一緒に敵と戦う戦友同士が、こういうチームを作ることが多いと聞いたことがある。

逆に、既にできている良いチームを壊すのは簡単だと、著者は述べている。それには、次のようなことを行えばよいらしい。

- ① チームに対して自己防衛的な管理をする。
- ② 官僚主義に徹した管理をする。
- ③ チームの作業場所を分散させる。
- ④ メンバーにいくつもの仕事を兼務させて、作業時間を分断する。
- ⑤ 品質の悪い製品しか開発できないような状況に追い込む。
- ⑥ 最初から「さば」を読んだ納期を押しつける。
- ⑦ チームを解体させる方針を最初から持っており、それをメンバーに感じさせる。

これらを実践することは、簡単である。良いチームというものは、作るのは難しいけれど、残念なことに壊すのはいとも簡単なものらしい。

ICTTI

手前味噌になって恐縮だが、私自身が経験したチーム作りの成功例について 1 つ話を追加しておきたい。

私は 2006 年 (平成 18 年) 11 月から 2009 年 (平成 21 年) 3 月まで、国際協力機構 (JICA) の専門家として、ミャンマーでのソフトウェア技術者とネットワーク技術者の育成に参画した。

具体的には、JICA とヤンゴン・コンピュータ大学 (University of Computer Studies, Yangon: UCSY) が合同で ICTTI (Information and Communication Technology Training Institute) という組織を作り、ミャンマー側が ICTTI の教員候補を用意し、私を含む日本の専門家が彼女たち³に最初の 1 年間 ICTTI の教員として必要な教育を行って、後はミャンマー側が自主的にこの組織を運営するという形のものだった。

この ICTTI のソフトウェア関係のカリキュラムは私が原案を作り、当時の JICA の職員の下承を得て決定した。具体的には、16 週間 (約 4 ヶ月) の研修期間を 2 つに分け、前半の 8 週間では Java の研修、オブジェクト指向技法による情報システムの作り方、データベースの設計法などの基本的な技術を、実習を交えながら教育し、後半の 8 週間 (約 2 ヶ月) を「ワークショップ」と称して、5 名程度を 1 チームにして、設計からテストまで 8 人月程度かかる情報シ

³ 教員候補として選ばれた 20 人の中、19 人が女性だった。

システムの開発に取り組ませた。

この目的は、ミャンマーの人たちが自主的にチームを作り、協力して情報システムの開発に取り組む訓練をすることだった。完成までに 8 人月見当必要な情報システムでは、いくら優秀な人でも 1 人・2 ヶ月で完成させるのは難しい。必然的にチームを編成し、誰かがリーダーシップを取り、誰かが全体の構想を固め、さらにそれを細分化して、全体を手分けし、分担して開発に取り組まなければ完成はおぼつかない。開発の途中でも、頻繁な打ち合わせや意見交換が必要になる。

最初は、なかなかうまくゆかなかった。教員候補は UCSY の現職の教員だったが、そのたいへん優秀な教員候補たちはチームを作ることができず、情報システムを完成できなかった。完成への見通しの甘さもあって、彼女たちは後で私から厳しい叱責を受けた。しかしその中に教員はコツをつかんで、適切に指導できるようになって、今ではほとんどのチームが課題の情報システムを完成できるようになったと聞いている。

当初のスタートからほぼ 10 年たつが、ICTTI は今でも健在である。卒業生は、ICTTI で（プログラムではなく）情報システムの作り方を教えてもらったとのコメントを残している。また卒業生はヤンゴンのソフトウェア会社に即戦力として高く評価されており、引く手あまたの状態である。ソフトウェア会社は採用後に、彼／彼女らに改めて社内でソフトウェア作りについて教育する必要がほとんどないという評価をしている。

ICTTI のカリキュラムが当初のものからほとんど変更されていないことから、JICA の元同僚はカリキュラムの古さを心配している。多分、それは正しいだろう。しかし自分たちでチームを作って、それを核にみんなで協力して情報システムを開発するという部分は、いつまでも古くはならないと私は考えている。

チームで仕事ができるということは、それだけでも素晴らしいことである。

キーワード

ピープルウェア、結束の強いチーム

略語

UCSY : University of Computer Studies, Yangon

ICTTI : Information and Communication Technology Training Institute

人名

トム・デマルコ (Tom DeMarco)、ティモシー・リスター (Timothy Lister)

参考文献とリンク先

[DEM87] トム・デマルコ／ティモシー・リスター著、松原友夫／山浦恒央／長尾高弘訳、「ピープルウェア 第 3 版 ヤル気こそプロジェクト成功の鍵」、日経 BP 社、2013 年。

前記の書籍はこの本の第 3 版であるが、この第 3 版には初版の内容がそっくりそのままの形で収録されている。

この本の原書は、以下のものである。

Tom DeMarco, Timothy Lister, “Peopleware 3rd Edition Productive Projects and Teams,” Dorset House, 2013.

[UME07] 梅田望夫著、「ウェブ時代をゆくーいかに働き、いかに学ぶか」、ちくま新書 687、筑摩書房、2007 年。

[WEI71] ジェラルド・M・ワインバーグ著、木村泉他訳、「プログラミングの心理学 または、ハイテクノロジーの人間学」、技術評論社、平成6年。

この本の原書は、次のものである。

Gerald M. Weinberg, “The Psychology of Computer Programming,” Van Nostrand Reinhold Co., 1971.

(2007 年 (平成 19 年) 12 月 5 日 初稿作成)

(2016 年 (平成 28 年) 9 月 13 日 一部修正)

