

連載 プロマネの現場から

第 37 回 『人月の神話』と『デザインのためのデザイン』との間

蒼海憲治 (大手 SI 企業・金融系プロジェクトマネージャ)

ソフトウェア構築における生産性や信頼性を一挙に高める「銀の弾はない」で有名なフレデリック・ブルックスさんが、『人月の神話』以来、実に 35 年ぶりの新作『デザインのためのデザイン』を出されています。

昨年 12 月、『デザインのためのデザイン』に合わせて、長らく絶版だった『人月の神話』も、従来の縦組みから横組みにして再刊されたことで、合わせて 2 冊を手にとることができるようになりました。

旧版の『人月の神話』の愛読者でもあったのですが、今回『デザインのためのデザイン』を手にとる上での最大の関心事は、『人月の神話』が書かれた時から 35 年の間に、システム構築において何が変わったのか？、そして、いま何が課題なのか？、ということでした。

35 年前を想像してみて、一番異なっているのは、システム構築に必要となる環境です。思いつくものを少し上げてみると・・・

パソコン、メール、グループウェア、インターネット、ウェブ、
パッケージソフト、リレーショナル・データベース、オブジェクト指向言語、
アプリケーションフレームワーク、仮想化技術、
BI ツール、ETL ツール・・・

静かできれいなオフィス、オフショア・ニアショア利用
などが、35 年前にはなく、またあっても実プロジェクトへの適用はこれからの段階でした。

システム構築を巡る環境、道具立ては大きく変わったのですが、『人月の神話』を再読してみて気づくのは、不思議なことに全く古く感じられないことです。

その理由には、大きく 2 つあります。

一つ目の理由は、『人月の神話』における議論の中心は、「人とチーム・組織に関する課題」であること。特に、大規模プロジェクトを扱っており、そこでの組織論、コミュニケーションのあり方は、プロジェクト活動をする以上、いまでも存在する古くて新しい課題です。

有名な「ブルックスの法則」は、「遅れているソフトウェアプロジェクトへの要員追加は、さらにプロジェクトを遅らせるだけである」といいます。

また、本のタイトルにもなっているように、「人月は、間違った危険な神話である。というのも、人月とは「人」と「月」が相互に交換可能だということを意味しているからだ」と指摘します。5名で6か月かかるプロジェクトを、30名で1か月では決してできません。世の中にそういう仕事もあるかもしれませんが、ソフトウェア構築は決してそういう仕事ではありません。「おいしい料理には時間がかかるのと同じように、結果が惨憺たるものにならないようにするためには、急かすことのできない仕事もある」ということを認識して、マネジメントする必要があります。

チームの組成は、「少数精鋭チームが最高である一人数はできるだけ少ない方がよい」。でも、「少数精鋭チームで非常に大きなシステムに取り組むと、開発が遅れすぎる」。

そのため、大人数のチームは避けられず、この矛盾をコントロールするプロマネが必須となります。

「プロジェクトマネージャの基本的仕事は、全員を同じ方向へ進ませることだ」とし、「プロジェクトマネージャの主要な日課は、意思決定ではなくてコミュニケーションを図ることである。そして、文書は計画と決定事項をチーム全体に伝達する」とことです。

「組織の目的は、必要となるコミュニケーションと調整作業の量を減らす」とあります。

「従来の木構造組織は、「二君に仕えず」という権限構造の原理を反映している」が、現在では、「組織におけるコミュニケーションの構造は、ネットワーク構造であって木構造ではない」とことを意識して、マネジメントする必要があると、マトリックス組織が求められていることを35年前の時点で指摘されています。

この人と組織の課題は、ハードウェア・ソフトウェアの課題とは異なる「ピープルウェア」の課題として、取り組むべきことであると考えています。

『人月の神話』が古く感じられない二つ目の理由は、システム構築における「本質」的な課題に焦点を当てていることにあります。システム構築にまつわる課題には、大きく「本質」的な課題と、「偶有」的な課題（副次的な課題）がありますが、この35年のIT技術や様々なプロダクトが解消したことの多くは、「偶有」的な課題に対してであり、ソフトウェアの持つ「本質」的な課題は依然残っているといいます。

「本質的な難しさとは、ソフトウェアの性質に固有な困難のことであり、

偶有的な難しさとは、目下の生産にはつきまとうが本来備わっているものではない困難さのことである。」

本質的な難しさには、複雑性・同調性・可変性・不可視性の4つの観点があります。

複雑性・・・大きくて複雑なことそのもの。また、複雑性は、ソフトウェアの大きさにしたがって、非線形に増大する。

同調性・・・多くの複雑性は、他のインタフェース（ハードウェア、ネットワーク、他システム等）にしたがうことにより発生する。

可変性・・・ソフトウェア製品は、アプリケーションの利用者、慣習および機器（媒体）といった文化的マトリックスにすっかりはめ込まれ、そしてそれらは絶えず変化し続ける。その変化は、ソフトウェア製品に容赦なく変更を強制する。

不可視性・・・ソフトウェアは目に見えない。《ソフトウェアの構造を制限したり単純化したりすることは進歩したにもかかわらず、その構造は本質的に視覚化できないままになっている。その欠落は1人の人間の頭の中のデザインプロセスを妨げるばかりでなく、複数の人間の間でのコミュニケーションもひどく妨害する。》

これらの「本質」的な課題は、「偶有」的な課題が減少した分、相対的に増え、今日では半分以上の割合を占めるに至っているといえます。また、大規模化、複雑化する中で、「本質」的な課題に対する難度は確実に上がっています。

インプリメンテーションにおけるソースコードの自動生成、プロダクトや各種ツールによるサポートは確かに充実してきています。でも、35年前の時点でも、インプリメンテーションにかかる工期は、プロジェクト全体工期の6分の1であり、現在だと10分の1程度になっているケースもあります。つまり、残りの9割の生産性や品質を、格段に向上させる方法が求められています。

また、不良の根本原因の大半は、インプリメンテーションではなく、設計やそれ以前の要件定義にあります。様々なモデリング技法はありますが、表記法のレベルではなく、表記する内容そのものに対する理解・把握は、エンジニアのスキルや知見に大きく依存しています。

そこで、35年ぶりの『デザインのためのデザイン』では、ずばり、この要件検討や要件定義を含むデザインプロセスに焦点が当てられています。

結論からいうと、このデザインプロセスを、事前に定義して整齐と行うことはできない。だから、前もって、やり直しや段階的にデザインを充実していくようにすることを織り込んで計画を立てるべきだ、ということを提言されています。

そもそもデザインプロセスを、系統的かつ段階的に行われるプロセスとしてモデル化されるべきものであるとする考え方は、ドイツの機械工学界で最初に発展したといいま

す。デザインプロセスを系統化するという取り組みは、いきなりコーディングすることに比べれば、格段の進歩でした。しかし、デザインプロセスの定義は、漠然と思っているよりはるかに難しい。

その理由は、《高度な技術分野のデザインでは、その領域の基本的な決定木を描けるほどの十分な知識を持ったデザイナーはほとんどいない》からです。

デザインプロセスにおける「決定木」にしたがって、デザインを整齊と進めていけばよいのですが、そもそも「決定木」が事前に決まりません。デザインを進める過程で、「決定木」が決まってきます。

なぜなら、

《デザインの最も困難な部分は、何をデザインするか決めることにある》し、
だからこそ、

《デザイナーが提供する主なサービスは、顧客が何をデザインしてほしいのかを見つけ出す手助けをすることにある》のが実態だからです。

また、要件とその重みづけは、デザインのプロセスを通じて変更し続けます。

《要するに、トレードオフをじっくり考えていくと、複雑に絡み合って相互作用する要因として、デザインの問題全体が新しく理解されてくるのである。同時に、要件の重みづけも変わってくる》からです。

つまり、顧客自身が、自分の要求が明確になるにつれ、優先順位づけも替わります。

そのため、大規模プロジェクトであったとしても、古典的なウォーターフォールモデルの見直しが迫られています。顧客と繰り返し対話しながら、要件を確定する方法として、プロトタイピングや漸増的（インクリメンタル）開発を取り込み、小さく生んで大きく育てる発想を考慮する必要があります。

さらに、このデザインプロセスを担うのは、最終的にプロジェクトの人員規模が数百名、数千名になろうとも、システム全体のアーキテクチャを構想するシステムアーキテクトは、1人か同じ意見を持つ少数のメンバーであるべきだといえます。

古今東西、素晴らしい創造物は、1人または2人によって作りだされている。さまざまな建築物だけでなく、絵画や音楽、文学作品・・・はもちろんだし、ブレインストーミングも、アイデアを収拾するところまでは衆知を集めることが有効なのでしょうが、そ

の深さや独創性は、じっくり考えた個人に勝てない。

その理由は、コンセプトの一貫性にあります。システムデザインにおいて、コンセプトの一貫性こそ、最も大切なことであると指摘します。

「偉大なデザインは、偉大なデザイナーから生まれる。ソフトウェア構築は創造的プロセスだ。」

ところで、『デザインのためのデザイン』の特徴的なところは、後半3分の1は、デザインの実践例になっているところです。ブルックスさん自身が実際にデザインした別荘や自宅の事例や、プロマネをされたIBMのシステム360やOS360、本の執筆におけるプロセスの事例が紹介されています。

その事例紹介の最後にある、デザインから得られた教訓が、とても興味深いものになっています。

ビーチハウスのデザインでの教訓としては、ユーザーやマネージャの立場でこう振り返ります。

「1. プロの建築家のデザインでも、細心の注意を払って確認し、その根拠を尋ねること。」

正直で、有能で、誠実な建築家であっても、間違いを犯すことはある。

2. 工事中に、頻繁かつ徹底的に点検すること。

正直で、有能で、誠実な施工業者であっても、間違いを犯すことはある。

3. メンテナンスのあらゆる側面についてじっくり考えること。

うまくいった設計デザインは、長期間メンテナンスすることになる。」

IBMのシステム360のデザインでの教訓としては、

「1. デザインには十分なプロジェクトの時間を割くこと。」

それにより製品は遥かに優れた、長い間役に立つものとなり、作り直しの作業が減ることで、より速く出荷することさえ可能になるだろう。」

IBMのOS360のデザインでの教訓としては、

「システムアーキテクトに、デザインに関する全権を与えること。」

《どれほどスケジュールの圧力があろうとも、デザインとプロトタイピングを十分に行えるだけの時間を取ることを。

そのように投資した時間によって、プロジェクトは遅れるどころか、早く完成するだろう。》

これらのデザイン上の教訓は、これからのシステム構築にとっても、大いに参考になります。ただし、これらの教訓を鵜呑みにするのではなく、適用に当たっては自身が行っているプロジェクトに合わせて見直す必要があることと、プロジェクト完了後の振り返り、レッスンズ・ラーンドを教訓として残していくことの大切さを再認識したのです。

フレデリック・P・ブルックス Jr. 『デザインのためのデザイン』 松田晃一・小沼千絵訳、ピアソン桐原、2010年刊

フレデリック・P・ブルックス Jr. 『人月の神話』 滝沢徹・牧野祐子・富澤昇訳 ピアソン桐原、2010年刊