

21世紀の組み込みソフト技術者に求められる開発パラダイム

本質的な課題から目を背けず、部分最適化から全体最適化へ

ここでは、今後、組み込みソフトウェア開発者が直面するであろう課題と、そのような課題に対処するための開発手法を紹介する。従来は、ソフトウェア開発の技術的な部分(例えば要求分析や設計など)に焦点を当てた手法が主流だった。一方、21世紀は企業活動全体を視野に入れた包括的な手法が主流となる。ソフトウェア開発が企業の経営戦略や製品戦略と強く結び付いていく。(編集部)



久しく、ソフトウェアの肥大化とそれに伴う開発の難易度の上昇がクローズアップされています。ソフトウェアを開発する企業や研究機関、大学など、それぞれの立場から、この問題に対する多くの提案がなされてきました。しかしこれまでのところ、ソフトウェア開発の多様な問題に対して、十分に対応できているとは言い難いのが現状です。特に組み込みソフトウェアの開発は、この傾向が強いようです。

本稿では、組み込みソフトウェアの課題を洗い出し、私たちがどの方向へ向かえばいいのか、ソフトウェア開発のどのような新しい考え方に着目すればいいのかについて、筆者の思うところを述べたいと思います。

1. 前世紀の方法論の限界と目指すべきゴール

実際のところ、ソフトウェアを開発してきたシステム開発の最前線では、これまで、いったいどのような課題に直面してきたのでしょうか？

● 既存の手法・方法論が抱える二つの弱点

過去を振り返ってみると、20世紀後半のソフトウェア業界では、肥大化・複雑化するソフトウェアをいかに作成するかということを取り扱う「手法」や「方法論」に注目が集まっていました。こうした手法や方法論は、扱う範囲や内容に違いはあるものの、その多くは「設計手法」と呼ばれ、主に技術的な議論に焦点が当たっていました。

「設計」の話題は、電子システム、ハードウェア、ソフトウェアを含めたものづくりの最重要部分と言えます。効果

的な手法や方法論を注意深く学習し、それらを開発現場が(自分たちのやり方に合わせて)効果的に取り入れることは、非常に重要なことです。

その一方で、こうした手法や方法論には、共通する以下の二つの課題が存在します。

● 対象としているのは限定された分野や製品

一つ目の問題は、それぞれの手法・方法論が対象としている分野や製品が、ある程度限定されているという点です。中には「汎用的である」と主張している手法・方法論もありますが、すべての分野、すべての製品にそのままストレートに適用できる手法・方法論は存在しないと考えておいた方が間違いないと思います。

これは、現実には、それぞれの手法・方法論が論じている技術的な考え方に違いがあるからです。そのため、利用者側が手法・方法論の持つ前提条件、および特徴をきちんと理解し、自分たちの製品に適しているかどうかを判断する必要があります。

● 開発作業の一部分にしか言及していない

二つ目の問題は、こうした手法・方法論の多くがソフトウェア開発の作業の一部分(例えば要件分析や設計の工程)にしか言及していないことです(図1)。企業や組織で製品開発を行う際には、さまざまな活動が存在します。単純に一部の開発作業の技術論だけで開発全体を議論することはできません。

開発作業はさまざまな活動の集大成であり、多くの利害関係を持つ部署や関係者(ステーク・ホルダ)がかかわっています。技術以外のことが製造や開発作業に大きな影響を与え、最終的に生産性や品質、コストを左右します。この

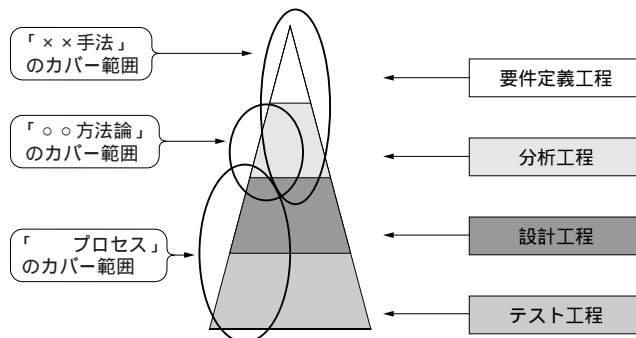


図1 開発手法・方法論の特徴

一口に手法・方法論と言っても、開発ライフサイクルの中でカバーしている範囲が異なる。手法・方法論の持つ個性が必ずしもすべての対応ツールにあるとは限らない。

ことについては、後ほど詳しく説明します。

● 現状打開の方策は所属する組織ごとに異なる

では、どのような取り組みを行えば、現在置かれている状況を打開できるのでしょうか？

この答えは単純ではありませんし、絶対的な回答も存在しません。なぜなら皆さんが所属する組織ごとに抱えている課題や置かれている環境が異なり、それに応じて対策が変わるからです。

答えが単純ではないもう一つの理由は、改善のための活動がソフトウェア開発の活動全般に及び、組織を挙げての取り組みになることです。市場要求の調査・企画から設計開発、保守までの作業内容を考慮し、それぞれの業務に適した改善活動を実施しながら、なおかつ企業として一貫性のある取り組みとする必要があります。つまり、それぞれの業務は、それぞれに適した改善活動を実施しながらも、ほかの部署との業務連携や改善活動との調整を図りながら、事を進めていかなければなりません。

幸い、改善活動には、古今東西の企業や組織がこれまでチャレンジしてきた結果を集大成した「ベスト・プラクティス」と呼ばれる定番の事例研究が存在します。今後は、ベスト・プラクティスを重視した手法・方法論が重要になってくると思います(図2)。本稿の後半では、プロセス改善を通じて組織を変革させるベスト・プラクティスの枠組みを紹介し、実際の開発現場の活動の指針を示していきます。

● 今世紀に入って、ソフトウェアの開発戦略が変化

1990年代までは、要件定義や設計の技法を扱う手法・方法論の競争は、現在ほどし烈ではなく、グローバルゼー

エンジニアリング研究成果を重視した手法・方法論

ベスト・プラクティスを重視した手法・方法論

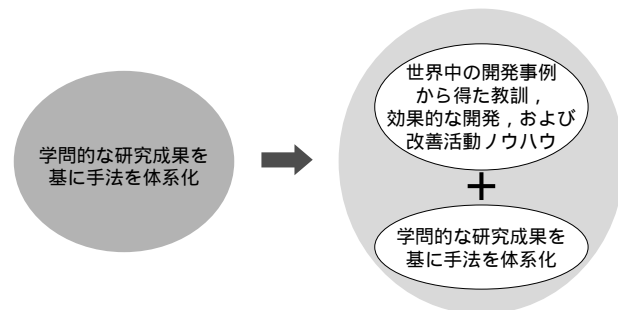


図2 ベスト・プラクティス重視にシフト

最近では、世界中で効果があった活動のベスト・プラクティスが従来の手法・方法論に取り込まれている。

ション(国際化)もそれほど進んでいませんでした。そのため、アジア諸国の安価な労働力の脅威は、主にハードウェアの世界にとどまっていた。そして、ソフトウェア開発の手法・方法論を現場にうまく取り入れることができれば、相応の効果が現れる時代でもありました。

しかし、近年はかなり様相が変わってきています。特に最近のソフトウェア開発では、「少しでも早くソフトウェアを完成させればよい」という発想とそのための手法だけでは解決できない課題が増えています。状況は日に日に複雑になっており、国内だけではなく世界規模の競争の時代に突入しています。

誤解していただきたいのは、ここで言っている「競争」が、単に製品の価格競争を指しているわけではないということです。この競争には、品質や性能といった製品の魅力、そして開発プロセス、国際標準化などの動向も絡み合っています。企業単位の競争から国家単位の競争へと発展してきている点も要注意でしょう。

これはどういうことなのでしょう？ 今やソフトウェア開発を含む製造業の競争力の源泉は、製品に備わる機能だけでは済まなくなっています。家電や自動車、情報端末などに見られるように、短期間で新製品を市場に投入することが重要になっています。特に自動車業界は、現在、転換点を迎えています。搭載する電子装置やソフトウェアが開発の生産性と自動車の性能を左右するようになっているからです。

このような背景のもと、21世紀に入って、ソフトウェア開発に対する戦略の考え方が変化しています。「ソフトウェ