

ソフトウェア・エンジニアリング

今日、ソフトウェアは私たちの日常を取り巻くいたるところに存在しています。Web閲覧や電子メール、銀行オンライン、PC(Personal Computer)・携帯電話はもとより、自動車、自動販売機、電子レンジ、ビデオ・デッキと、生活のありとあらゆる場にソフトウェアがあふれ、もはやソフトウェアなしでの文明生活は想像すらできない状況であるといえます。見方を変えれば、これらのソフトウェアが正しく動作しない際の被害・不便さも社会にかなりの影響を及ぼします。ソフトウェアやシステムの不具合に関するさまざまなニュースが報道されたことも記憶に新しいところでしょう。

このような現状において、IT(Information Technology : 情報技術)に携わる私たちとしては何をすべきでしょうか。土木・建築の世界に目を転じると、その悠久の歴史とともに、土木工学・建築工学というものがしっかりと「実学」として根付いていることに驚かされます。もちろん土木・建築の世界とITの世界では、つくり上げる対象が異なりますし、技術変革の度合いや、成果の寿命など、さまざまに異なる点もあります。しかしながら、物事を構造的にとらえてそれらを組み上げていくという作業においてはまったく同じ目標を持っています。つまり、ソフトウェアのライフサイクルにおいても「実学」の重要性は変わらないということです。ソフトウェア・エンジニアリングとは、ソフトウェアをつくり、適用し、保守していくための具体的な実践方法を探究していく試みです。ソフトウェア・エンジニアリングの研究・実践こそ、ソフトウェアにかかわるITプロフェッショナルすべてが挑戦すべき事柄であり、向上へのためめぬ努力に値する重要な分野であることは言うまでもありません。

Perspectives in this Issue

Software Engineering

Today, software is associated with almost every aspect of our lives from web browsing, e-mail, online banking, personal computers and cell phones to less obvious forms such as automobiles, vending machines, microwave ovens and VTRs. In fact, life today is unimaginable without software. This implies that software will have grave effects on society in the event of a malfunction as evidenced by frequent news concerning software or system failure.

Under such circumstances, what is expected of those of us working in IT(Information Technology) industries? In the world of construction and architecture, we are amazed that, through their long histories, they have developed into practical sciences in the form of civil engineering and construction engineering, respectively. Of course, construction and IT differ in many ways such as end-products, the level of technical innovation and life cycles. However they are identical in terms of achieving goals by structurally grasping objects and assembling them. In other words, practical science is equally important to software life cycles.

Software engineering is a discipline which strives to find practical ways in which to apply and maintain developed software. Studying and practicing software engineering should be the universal goal for IT professionals associated with software and constant effort must be made to find ways for improvement.

今日、デジタル家電製品や携帯電話などの開発は、その熾烈な競争上、非常に多くの機能を盛り込んだ新製品を数カ月サイクルで市場に出すことを強いられています。これらの製品は多くの場合シリーズ開発であり、そのターゲットは特定の顧客層となります。ところが、こうしたシリーズ開発は開発ラインが独立するケースがほとんどで、共通のソフトウェア資産を大規模に再利用する仕組みを備えているケースはまれです。

そのため製品開発のリード・タイムは増大する傾向にあり、結果としてタイムリーなリリースができずに顧客満足度の低下につながってしまいます。

そこで、CMU(Carnegie Mellon University : カーネギーメロン大学)のSEI(Software Engineering Institute : ソフトウェア工学研究所)を中心に考案されたのがソフトウェア・プロダクトラインです。ソフトウェア・プロダクトラインは、次の三つの大きなアクティビティから成る開発プロセスの取り組みです。

- ・ コア・アセット開発(ドメイン・エンジニアリング)
- ・ プロダクト開発(アプリケーション・エンジニアリング)
- ・ マネジメント

従来のナレッジマネジメント、再利用技術、ドメイン・エンジニアリング、アーキテクチャー中心などの考え方が有機的に結合され、プラクティス・エリアとして体系付けられていると見なせます。

ソフトウェア・プロダクトラインの狙いは、最終的に開発の生産性を上げ、コストの低下を図ることにあります。その方法として、コア・アセットを開発し、共通のアーキテクチャーの下、作成すべきプロダクトのラインごとにそれらを組み合わせて全体としての開発を進めるというコンセプトが生まれました。ソフトウェア・プロダクトラインは、生み出されたきっかけこそ組み込み開発ですが、最近ではそのコンセプトが着目され、エンタープライズ系のITアプリケーションの開発にも転用する試みが増えつつあります。

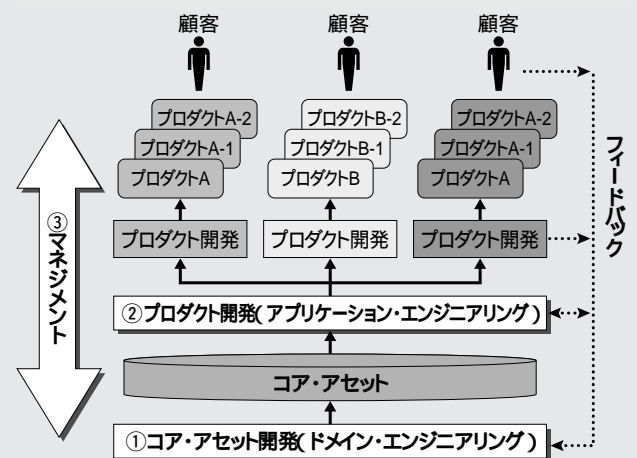


図. ソフトウェア・プロダクトラインのプラクティス・エリア概要