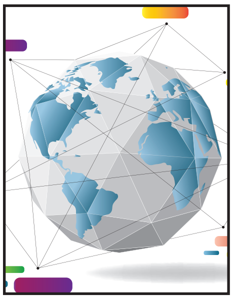

Engineering the Success of Software Development

Simon Liu, *US National Agricultural Library*

Software is constantly evolving, and its effect on business operations and daily life continues to grow. It's hard to imagine a day without using your cell phone, browsing the Internet, sending an email, watching TV, or using other IT products and services. Today, most human activities and organizational businesses depend heavily on software. Unfortunately, software failures happen across many industries—banking, retailing, marketing, the airline industry, and others.

Software project failure rates are considerably higher than failure rates for other types of engineering projects.[1] A recent study of 5,400 large-scale IT projects (those with initial budgets of more than US$15 million) revealed persistent IT project development problems,[2] with 17 percent of the projects going so badly that they threatened the company's existence. The study also revealed that 45 percent of the projects were over budget, 7 percent weren't on schedule, and 56 percent delivered less value than expected. Additionally, in November 2012, the US Air Force decided to scrap a major enterprise resource planning software project, called the Expeditionary Combat Support System, after it racked up $1 billion in expenses but failed to yield "any significant military capability."[3]

These are disheartening statistics for organizations that invest a great deal of time, money, and staff energy into IT projects. Why are software failure rates so high and the impact so large? What are the main reasons for software failure?

Software Engineering

Studies suggest various reasons for software project failures,[1] but a common reason cited is companies failing to follow sound software engineering practices. The software engineering discipline examines how to build and maintain software systems in a controlled, predictable way, with the aim of developing high-quality, reliable, maintainable, and cost-effective software in a timely fashion. It's a distinct and powerful tool for managing the development of large software projects.

Software development depends heavily on three major factors: the people, processes, and products. Quality products require good people following solid processes, which is why software development is a people-centric activity. Developers must have the proper skills to plan, manage, and develop software. The process provides a framework for establishing a solid and comprehensive plan for developing the software. The product is the solution or artifact delivered to the customer. It can include services, applications, source code, requirements and specification documents, the system design, and manuals.

Software engineering has become an important discipline of study, practice, and research along the people, process, and product dimensions. It's essential for developing high-quality and maintainable software that's delivered on time and within budget while satisfying the requirements.

In This Issue

The five articles in this issue provide a glimpse of software engineering, focusing primarily on the process and product dimensions. The first article, "Operational Business Intelligence: Processing Mixed Workloads," describes a mixed-workload environment in a software engineering process. Michael Seibold, Dean Jacobs, and Alfons Kemper propose an architecture for software-as-a-service business applications that can process these mixed workloads with a special-purpose main-memory database system, while meeting strict service-level objectives.

In "Leveraging Process Mining Techniques," Geetika T. Lakshmanan and Rania Khalaf define semistructured processes as data-driven, human-centric, flexible

Published by the IEEE Computer Society

processes whose execution between instances can vary dramatically. Lakshmanan and Khalaf identify major key characteristics of semistructured processes and the mining challenges they present and then discuss emerging mining approaches that aim to address these challenges.

In "Personal Medical Monitoring System: Addressing Interoperability," Yuan-Fa Lee proposes a new medical monitoring system that lets medical devices transmit vital sign data to an application host device (AHD) on a network. The medical monitoring system is the first dual-meter device providing both blood-glucose and blood-pressure measurements certified by the Continua Health Alliance. Using the medical monitoring system, users can measure their vital signs, blood pressure, and blood-glucose measurements, which are then transmitted to any Continua AHD using the interoperable platform.

In the next article, "A Smart Healthcare Systems Framework," Haluk Demirkan proposes a systematic framework for conceptualizing the data-driven and mobile- and cloud-enabled smart healthcare systems. With the adoption of smart healthcare systems, healthcare organizations can provide cost-effective quality services with fewer IT set-up costs and reduced risks.

Finally, in "The Market-Driven Software Ecosystem," Ligui Yu argues that a software product isn't used alone—it's integrated with supporting hardware and software that, together with developers, users, and other players, form a software ecosystem. Yu describes the concept of a market-driven software ecosystem in which the market acts as the energy source, exemplified in both open source and mobile-computing communities.

The nature and complexity of software has changed rapidly in the past decade, and software engineering offers a systematic approach for producing high-quality software. Over the years, software engineering has evolved, and IT professionals must continue to learn, improve, and develop the discipline to deliver quality software on time and within budget, satisfying user needs. I hope these articles inspire you to explore new software engineering processes and examine new software products. IT

References

1. R. Goatham, "The Story Behind the High Failure Rates in the IT Sector," Calleam Consulting, 2009; http://calleam.com/WTPF/wp-content/uploads/articles/Whatmakes.pdf.
2. M. Bloch, S. Blumberg, and J. Laartz, "Delivering Large-Scale IT Projects on Time, on Budget, and on Value," McKinsey & Co., Oct. 2012, www.mckinsey.com/insights/business_technology/delivering_large-scale_it_projects_on_time_on_budget_and_on_value.
3. C. Kanaracus, "Air Force Scraps Massive ERP Project after Racking Up $1 billion in Costs," *CIO Magazine*, 14 Nov. 2012; www.cio.com/article/721628/Air_Force_scraps_massive_ERP_project_after_racking_up_1_billion_in_costs.

***Simon Liu** is the director of the US National Agricultural Library. His research interests include IT architecture, cybersecurity, software engineering, and database and data mining. Liu has two doctoral degrees in computer science and higher education administration from George Washington University. Contact him at simonyliu@yahoo.com.*