Reports from the Field

Using Extreme Programming and Other Experiences

Wolfgang Strigel, Software Productivity Center

Learning from the successes and failures of others is a quick way to learn and enlarge our horizon. Our own experience can only cover a narrow path though the wealth of existing knowledge. Last June, the IEEE Software Editorial Board decided to make more room for experience reports and give our readers a forum to share their own learning experiences with others. If you are interested in submitting an experience report, please refer to www.computer.org/software/genres.htm for author guidelines.
By lucky coincidence, we had a large backlog of experience reports and were able to include six of them in this issue. On an ongoing basis, we hope to publish two or three shorter experience reports per issue. I think you'll enjoy these interesting stories that are typical of the challenges we all face in this industry. Even if you were to pick only one gem from the experience of others, it might help you, your project, and your company.

The first four articles address the topic of Extreme Programming; the final two address a different set of experiences from the field.

Extreme Programming in the real world

Many methodologies have come and gone. Only time will tell if one of the more recent methodology innovations, Extreme Programming, will have a lasting impact on our way to build software systems. Like other methodologies, XP is not the ultimate silver bullet that offers an answer to all development problems. But it has gained significant momentum and an increasing number of software teams are ready to give it a try. Our first article is not really an experience report but an interesting comparison of XP with the more established Capability Maturity Model. As one of the foremost experts on CMM, Mark Paulk offers an opinion on XP as a lightweight methodology from the perspective of the heavyweight CMM. From my perspective, the difference is not so much the “weight” of the methodology than the way they are introduced in an organization. XP tends to be a grassroots methodology. Developers and development teams typically drive its introduction. This becomes quite clear from reading the subsequent experience reports. CMM, on the other hand, is typically introduced at the corporate level and then deployed to development teams. As in past “methodology wars,” there are heated debates about the pros and cons of the respective approaches. I agree with Paulk that CMM and XP can be considered complementary. To establish lasting success, methodologies need buy-in from management as well as from the developers.

Martin Fowler offers a few links to further information about XP and agile methods in the “Web Resources” sidebar.

Two more reports

The last two articles in the set cover dissimilar experiences, but they have one thing in common: an account of our continuous struggle to make software development more efficient. The first article presents a typical example of survival struggles in a rapidly growing company and its attempts to use process to get development activities under control. The second article describes a technique, called defect logging and defect data analysis, that aims to decrease programmers’ repetitive errors. The author picked one element of the Personal Software Process and made it easier to apply.

About the Author

Wolfgang Strigel is the founder and president of Software Productivity Center, a consulting and products company, and of QA Labs, a contract testing company. His interests include collaborative software development, process improvement, project estimation, testing, and software engineering economics. He has a BSc in mathematics from the Technical University, Munich, Germany, an MSc in computer science from McGill University, and an MBA from Simon Fraser University. Contact him at strigel@spc.ca.