

Is extreme programming the solution to today's dynamic project development needs?



Taking it to the extreme

RIGOUR FOR ROGUES?

According to Ward Cunningham, an XP lecturer and writer, extreme programming has its fair share of both supporters and detractors. "Extreme programming (XP) has been described as combining all the worst practices of rogue developers in one methodology. At the same time, it has been called a method of high discipline – which isn't necessarily contradictory. In fact, XP is an original configuration of practices that are known to be useful in small quantities, only in XP they are carried to the extreme... Iteration to the extreme. Unit testing to the extreme. Pair programming to the extreme. Re-factoring to the extreme."

The economics of application development have changed radically in the past decade. On any given project, the cost of hiring developers now typically outstrips the cost of the chosen hardware platform by a factor of 20. Understandably, then, organisations are constantly looking for ways to maximise programming efficiency and minimise cost.

In the early 1990s, programmers Kent Beck and Ward Cunningham were thinking about better ways to develop software. They concluded that to improve communication and simplicity, programmers need to get constant feedback on how well the programming effort is going; and they always need to "proceed with courage".

In March of 1996, Beck started a project at DaimlerChrysler using his new concepts in software development. The result was Extreme Programming (XP), a methodology that is said to complement the new dynamic nature of application development, and the concept has caught fire in many programming and IT project management circles.

XP sounds almost like the holy grail of programming. According to Beck and Cunningham, it "delivers the software the customer needs when it is needed" and enables developers to respond to changing customer requirements, even late in the lifecycle. "Your customers may not have a firm idea of what the system should do," says J. Donovan Wells, the author of *Extreme Programming: A Gentle Introduction*. "You may have a system whose functionality is expected to change every few months. In many software environments, dynamically changing requirements is the only constant. This is when XP will succeed while other methodologies do not."

Beck and Cunningham, who designed XP for groups of between two and 10 programmers, outline 12 key

practices in XP. (See box, *Extreme principles*, for a summary of the main points.)

TEAM SPIRIT

As the key practices suggest, the emphasis of XP is on teamwork. Managers, customers and developers are all part of a group "dedicated to delivering quality software". Programmers move between projects so that knowledge is shared, leading to cross-training.

It is easy to see the benefits of these principles. However, there is an aspect of XP that is rather more controversial. For its creators, the most important facet of the XP approach is that all code should be written by two people working together at a single computer. "It's counter-intuitive," says Wells, "but two people working at a single computer will add as much functionality as two working separately, except that [the end result] will be much higher in quality." Equally controversial, the XP rationale suggests that overtime should be banned as it sucks the motivation out of the team.

Such suggestions will draw scorn from many IT managers, but XP works, say some of its early adopters. The approach is being used successfully by companies such as Bayerische Landesbank, Credit Swiss Life, Ford, UBS and OK Chicken, which provides programmers to investment banks in London. John Giblin, senior vice president of engineering at Dublin, Ireland-based software company Iona, turned to XP last summer to reduce software delivery times. "Sometimes, because of the length of development cycles, by the time the product is developed, the original set of requirements is only partially relevant," Giblin says. But, since using XP for Iona's application server product, the company has seen strong results. "We're getting products to the market much faster," he says.

There are more cautious voices, however. Java guru David Megginson, head of Megginson Technologies, says that XP has "a lot of very good ideas, but the package as a whole may lead to fanaticism." While his company uses several XP principles in its work – including starting with the minimum that can possibly work, eliminating duplication, and constant re-factoring – Megginson does not feel the need to radically change his own working practices. "I feel no sudden urge to do all my coding paired up with another developer in front of the same terminal."

Extreme principles

Extreme programming (XP) is a method of software project development and management that emphasises teamwork over coding skills. The main practices are:

- Programmers work in teams of two using just one computer
- The customer defines the business value of desired features, and uses cost estimates provided by the programmers to choose the aspects of the design that need to be coded
- Simple systems are put into production early, and updated frequently
- Programmers 're-factor' to improve the design of the system throughout the development process
- Customers are on hand to determine requirements, set priorities and answer the programmers' questions.

CONTACT

Rob Buckley

rbuckley@infoeconomy.com