What do your Toyota and your cool new app have in common?

They may both be the products of Lean Thinking

The principles of Lean Production transformed the manufacturing sector in the 1980s and made Japan a serious contender in the automobile industry. By scrapping the traditional mass-production approach to manufacturing and adopting a simplified approach that focused on rapid development and customer satisfaction, Japan proved that efficient processes could lead not only to better products, but happier customers. Since then Lean Thinking has been applied to many fields, including software development, with undeniable success.

"Lean development principles mirror lean manufacturing," says Dan Perron, director of organizational effectiveness at Protegra. "Lean development focuses on value and attacks waste at its roots by implementing efficient iterative and incremental processes supported by team-client communication and rapid feedback mechanisms."

The traditional model for software development follows a rigorous plan over a long timeframe with a lot of requirements gathering effort and design early in the project, then development, then testing and debugging, and finally implementation and maintenance.

"The problem with the traditional model is that it can take a very long time before the business end-user actually gets some say in the process," says Perron. "And by that time the developers have used a lot of time building in features that may not provide any value at all to the end-user. It's actually quite risky. Problems don't show themselves until late in the process when it can be difficult and very costly to go back and make corrections. The customer can end up with a piece of software that doesn't meet his needs."

When Lean Thinking is applied to software development the model changes dramatically. Often called Agile Software Development, developers and end-users work together to identify immediate needs the project must address. The development team (including the customer/user) focuses on producing a working piece of software with limited functionality within a very short time frame. In each iteration of the software, new functionality may be added, changes incorporated, and problems fixed.

"You can deliver quick solutions to complex problems that the user has identified," says Perron. "There's immediate value in it for the business and end-users, and the developers don't waste time working on features that may no longer be needed."

"One of the identifying features of Lean Software Development is its change tolerance and flexibility. The team doesn't shy away from last minute requests for change if these are valuable," says Perron. "That's what drives the whole process. When the customer or user identifies a need and the development

Protegra Solutions

Protegra is a Winnipeg-based, employee-owned IT and business consulting company. Protegra's offering includes business alignment and process improvement, IT architecture and IT strategic planning, solution design and development, and outsourced support and maintenance services.

We call our development method Lifecycle™; it has been in place for over 5 years and benefits from rigorous continuous improvement and effective feedback loops. Its founding principles are very much aligned with "Lean Thinking"; its practical application delivers results that could not be achieved otherwise.

"Lean Thinking" -- making waste visible and driving it out of our processes, high customer focus and short feedback loops – is a mindset. There are no silver bullets: it's about people, teamwork, learning and leadership.

For more information:
www.protegra.com

Who fills the skills gap when the Baby Boomers go?
read article »

What do your Toyota and your cool new app have in common?

team addresses it quickly, you immediately provide value for the customer."

Users of popular software packages from large software vendors may be forgiven for wondering where Lean Thinking fits into the picture. "Traditional models of development are still predominant," says Perron. "Lean Software Development has been criticized for being a bit too loose, a bit too focused on face-to-face conversation with the customer and end-user. You end up with less documentation in the Lean process, simply because the ongoing communication is intense, high bandwidth: everybody has been involved in the process from the start."

That said, the benefits of Lean Thinking applied to software development promise benefits that can't be ignored. The recent proliferation of Web 2.0 applications, delivered over the Internet, is a clear example.

"The development teams behind many online applications listen to users every day," says Perron. "User feedback is built into the development cycle. Google's Gmail, for example, began with limited functionality, but new features and fixes are being added all the time."

In other words, Lean Thinking in software development is here to stay.

read article »

Chicago mission opens doors for Winnipeg ICT companies
read article »

Primmer Outlines 2006/2007 ICT Priorities and Strategies
read article »

@-IT Systems Group Inc.
read article »

MOSKAL Electric Ltd
read article »

Advertise with ICTAM
read article »

Principles behind Lean Software Development

The Agile Manifesto

We follow these principles:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximizing the amount of work not done--is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Source: agilemanifesto.org