Responding to Change

There is nothing permanent except change.
-Heraclitus

How does it feel to read: *There is nothing permanent except change*? Do you get the thrill of open-ended possibilities? Or the sinking feeling that there is no way to catch up?

Suppose you just completed writing a 300-page requirements document, and worked with the latest design notation to document a 100-page design document. How would you feel about a change in the business needs? A little motion sick perhaps?

> Making Change the enemy is a losing strategy

A recent ad in InfoWorld magazine for Rational's source control system listed the following supposed software developer *enemies*: "Bugs, Changes, Meetings, Changes, Chaos, Changes". It seems clear that in today's environment making Change the enemy is a losing strategy. Even Heraclitus in the year 510 BCE would agree. He said you could never step in the same stream twice, because it is always changing.

Change has many forms in a software project: requirements, technologies, the team members, even the words we use to describe the problem and the solution all of these change simultaneously. There are two main approaches to managing change in software project management. One approach is to predict & prepare, the other is to become Agile & adapt.

The traditional approach attempts to predict, plan, and prepare for foreseeable changes. This approach attempts to anticipate changes through the phases of development and be ready as they occur. Martin Fowler refers to this as the *Engineering* approach to software. It works quite well, when changes are foreseeable and the problem domain is familiar. Once the potential changes are

understood the team leadership can address them. This then also requires leadership that has experience guiding a similar project. (Warning: You can never step in the same stream twice.)

Plan perfectly or Adapt gracefully

Engineering processes handle foreseeable changes with a plan involving design documents and special notations. Resources are applied to creating and maintaining documents, diagrams, and schedules in order to better organize the team toward eventually producing program code. Un-foreseeable changes are addressed as they occur without a formal process. When Engineering is *done right* we will not have un-foreseen changes. Engineering approaches work well as long as Change remains minimal, contained, and under control.

The Agile approach is gaining increasing attention by changing our relationship to change. In a move reminiscent of the ancient jujitsu masters, we convert the dynamics of change to our own advantage. The Agile motto is: *Embrace Change*. Change is harnessed to empower the development team. Maximum resource is applied to solving the problem at hand by completing the program code in a sustainable way. Well crafted, test, running code total focus. Un-foreseen changes are handled exactly like foreseeable changes without breaking the stride of the programming team.

An Agile team does not expect frozen requirements or everlasting technology

Responding quickly is a major part of project success. Agile processes take that to the logical conclusion. An Agile team does not expect frozen requirements or everlasting technology. This mindset also allows a team to deliver software faster. We do not need to schedule long meetings to search for eternal truths and the absolute end-point of the software being developed. We focus tightly on delivering a top-quality small release of the feature with the highest business value today. We have techniques, processes, and confidence to ensure that the tomorrow's high-value feature will be addressed in the same way while flattening the cost curve of adding new features in the future.

The Engineering approach to project management attempts to predict and control change. The Agile approach takes advantage of change to the benefit of the software project.
There is nothing permanent except change. *-Heraclitus*
Embrace Change!

Please send comments or questions to: mailto:info@objectwind.com

Mark Windholtz
ObjectWind Software Ltd.
www.ObjectWind.com