January 2010

With a traditional process we schedule our activities one after another until the project end date. Design always starts when analysis is finished. Coding starts when the design is done, etc. We make a plan to get everything finished on time. At the end of the schedule we demonstrate our new software and see if the customer likes it.
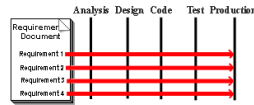

We can visualize our requirements remaining constant throughout the project. Requirements flow through the different activities like a production line. We finish each activity on all the requirements at the same time and deliver a fully functioning program at the end. This approach offers many cost and effort efficiencies.

We can have people who specialize in one specific activity and change our team as each phase is finished. Unfortunately most projects

can't use that efficiency because requirements get changed before the project ends, usually right from the start.

Agile processes like Extreme Programming (XP) and Scrum accept that requirements will change and create opportunities for improvement and competitive advantage. We just can't count on requirements being a constant. Instead we will represent each feature the customer wants as a user story. Story cards are a low cost mechanism for scoping out a project without spending our entire analysis budget up front.
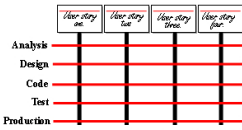
Now we can flip the axis of the previous chart. We visualize the activities, our process itself as the constant. Our process is applied to our user stories in sequence. The activities are on going and the user stories get what ever they need.

Don't panic. With professional software engineers on our project we can relax knowing that the team will do what is needed to get the job done. Any activity needed with any combination of people will just get done without any further scheduling or ceremony. This is the spirit and benefit of a self organizing team.

So now when we create our schedule we pencil in the features in order of importance. This offers us a different kind of efficiency; we can change our minds about what we want without huge cost over runs. We can also stop at any time and have the most important features done.

An Agile process takes the traditional process and turns it 90 degrees on its side. This allows managers to get an estimated cost per feature instead of per activity. Customers can make the difficult decisions about what can be left undone in an intelligent way. If you find yourself running short on money you will not have wasted your money on analysis and design for features that will never be coded and tested. Costs make more sense this way since customers can't pick and choose which activities should be done. This gives us a "shopping list" style of requirements and planning.

One of the costs of managing this way is to always be ready for new features. We must keep code and design quality up throughout the entire project. NEXT PAGE

January 2010

Agile-Process.org home | Quality is Important | About the Author