Product Management - Agile or Waterfall Development

Contributed by John Mansour
07.31.2007
Last Updated 09.15.2008

Every so often a new concept sweeps the industry, promising to be a panacea for whatever ails us . . . think open source. More recently, Agile development has gained momentum as such a panacea. When development cycles are slow and the product misses its mark, it’s appealing to think a new methodology will be the Holy Grail for delivering products faster with more features.

5 Common Sense Rules to Get the Best of Both

Every so often a new concept sweeps the industry, promising to be a panacea for whatever ails us . . . think open source. More recently, Agile development has gained momentum as such a panacea. When development cycles are slow and the product misses its mark, it’s appealing to think a new methodology will be the Holy Grail for delivering products faster with more features.

But the debate over agile versus waterfall development isn’t really about the product development methodology. It’s about delivering top quality products in a reasonable time that solve high impact problems for a broad market.

In the end, technology companies just want to capture market share and generate healthy revenue streams as quickly as possible. And customers just want products that work and solve complex problems in a simple fashion.

If the agile versus waterfall winds are swirling around your organization like a cyclone, read on. You don’t need to go to one extreme or the other. Take concepts of both methodologies and apply some common sense. You’ll be delighted with the results.

Key Tenets of Each Methodology

- Agile development typically calls for shorter planning cycles, light documentation and many short iterations of each concept to arrive at an end solution, whereas Waterfall development calls for longer planning cycles, complete documentation and few if any iterations to create the end solution.

- Taken to the extremes, Agile is best suited for early stage products where cross functional teams are together during the entire development cycle, whereas Waterfall is best suited for more complex enterprise products where documentation is critical to cross functional communication and traceability.

Reality

The reality is that many technology products today fall somewhere in the middle of early stage and enterprise, so applying one methodology or the other to the extreme will only get you right back where you started – an inability to deliver high quality revenue generating products in a reasonable time.

5 Common Sense Rules to Get the Best of Both Worlds

- Define situations & problems. Define business situations and problems at their most fundamental level (they have nothing to do with your product) and validate the accuracy of your problem definitions with target customers who are screaming for a solution. A well defined problem is half solved.

- Prioritize. Let the impact of the problem determine the development priority. Deliver the highest impact solutions first. Follow rule #1 to make this a simple exercise.

- Iterate and validate. Just because you iterate doesn’t mean you get it right. Involve your target users. In most cases, 2-5 iterations are sufficient to accurately nail the solution. The key to getting value from each iteration is doing it with people outside the building…your target users.

- Document. Product quality problems begin when new features are built on top of existing features without any reference to earlier designs. Pay now or pay later. It’s 50-200 times more expensive to fix it later. And one more thing…documented problems, solutions, designs and test cases completed during the development cycle improve the quality and speed of the product rollout by serving as the main vehicles of knowledge transfer required to market, sell, deliver and support a product.

- Deliver with maximum impact. There is no law that says every solution must be launched when development is complete. Determine both the market impact of the solutions as well as your customers’ ability to absorb them before launching. Launch when it benefits you most.