

4 October 2007

The Roots of Agile, Part 2

by [Preston G. Smith](#)

Last month (see "[The Roots of Agile, Part 1](#)," 6 September 2007), we broadened the concept of agile software development by considering the underlying factors that make any development process more agile, whether it is a software development process that can take advantage of certain special characteristics of the software medium or the development of a non-software product for which conventional agile software techniques would not apply. In this concluding installment, I cover five additional basic factors that will make your product development of any kind more agile.

Plan Piecemeal

If your project is subject to constant change, you can waste a lot of time constantly replanning. The alternative is not to quit planning but to plan in small chunks and use these plans before they decay. Clearly, this is a balance between the waste implied in not having a plan (too little planning) and the waste in plans that become obsolete (too much planning).

There are two ways to plan piecemeal. One is rolling-wave planning, where you plan near-term activities in detail and leave the longer-term plans -- the ones most subject to change -- at a high level. The other is loose-tight planning, where you alternate periods of careful planning with periods where the project is relatively open to change. Agile software development does this with its iteration planning, but Boeing also used this technique in developing its 777 airliner.

Involve Customers

The source of many changes during development is customers: they change their minds or are uncertain of what they actually want. Thus, carefully selected customers can be your bellwether for likely areas of change. The trick is to pick customers who can articulate their dissatisfaction with the current product and to find clever ways of keeping in touch with them. For instance, when I wrote *Flexible Product Development*, I assembled a group of 27 product developers worldwide who were facing great change in their projects and knew that existing methods were not working well enough. They became my "customer council" to guide me and suggest areas that were especially subject to change and uncertainty.

Organize Around Iteration

The earmark of a flexible development process is iteration. Activities are not executed sequentially but by designing a little and trying it out, then

More in this series

The Roots of Agile

[Part 1](#)[Part 2](#)

Welcome to the **Agile E-Mail Advisor**, a weekly electronic briefing from Cutter Consortium's [Agile Product & Project Management](#) Advisory Service.

Recently published:

- [Principles of Planning, Part 3: What and Why?](#) by David N. Rasmussen
- [Understanding Change in a Broader Context Than Just IT Investment Assessment](#) by John Berry
- The Cutter Blog: [Watson and Rosen on Agile and EA](#)
- [Single or Multi-Sited Teams](#) by Jutta Eckstein
- [See more ...](#)

XML [Get the Agile Feed](#)

What is Open Innovation?

We'd like to know if and how your organization is embracing this concept. Let us know by participating in our survey on [Open Innovation](#).

18 October 2007**11:30 pm EDT / 8:30 am PDT / 15:30 pm UTC/GMT**

[Looking For and Developing Successful Agile Management Behaviors](#)

Spend an hour with [David Spann](#) and discover what behaviors are present in successful agile managers, how to identify who possesses those behaviors, and how to cultivate them in others so they can successfully manage and lead agile initiatives, and -- if that individual is you -- what behaviors you need to focus on to improve your effectiveness. [Learn more ...](#)

proceeding with yet another loop of design-try. As mentioned before, if you knew at the outset exactly what you wanted and knew that it would not change, iteration not only would be unnecessary but probably wasteful.

However, to the extent that you are not sure what is needed or if needs are likely to change, iteration becomes a powerful tool to actually see where you stand in satisfying the customer. That is, iteration provides feedback on where you are, which is invaluable in a changing world. All flexible methods use iteration intensively -- they cannot afford not to.

Preserve Flexibility in Upper Layers

Product development processes present a dilemma regarding change: too much freedom to change and you have chaos, but too little and you become locked into poor choices. An interesting fact we have learned about development processes is that process quality generally comes from doing more basic activities consistently, but flexibility comes from the freedom to rearrange these activities to meet current needs.

Consequently, standardize your basic activities, such as how you run a test and how you analyze its results or how you space components on a printed circuit board. Then purposely leave developers the freedom to rearrange these basic activities as they proceed. Observe that many development processes do poorly here in that they attempt to specify activities in the upper layers, for example, by insisting that certain activities be done in certain project phases.

Maintain Reserve Capacity

When change happens in a development project, you will probably have to redesign something and retest it. Consequently, you will need to repeat activities like design and testing frequently on short notice. This may seem obvious, but many companies do very poorly in keeping such capability available for quick response. Instead, they run "lean" by overcommitting all resources: labor, equipment, facilities, and management attention. When the virtually certain change demands resources, the change must go into a long queue, which destroys responsiveness and thus flexibility.

The solution is straightforward but not always inexpensive. Simply provide reserve resources in known bottleneck areas that affect flexibility. You can find these bottlenecks and see how detrimental they are by proposing a change and watching how long the change requires and where it hangs up.

In Conclusion

Whether you are developing software using an established agile methodology such as Extreme Programming or Scrum or, alternatively, developing semiconductors or footwear using a proprietary methodology, you can make your process more accepting of change by applying the principles I have described. Remember, if you aren't changing, you aren't innovating.

I welcome your comments on this *Advisor* and encourage you to send your insights on agile project management in general to me at comments@cutter.com.

Cutter Consortium

-- [Preston G. Smith](#)

Collaboration and Documentation

To update your e-mail address with Cutter Consortium, reply to this message with your old and new address. Or phone +1 781 648 8700.

If you do not wish to receive this email newsletter, [unsubscribe here](#).

Did a colleague forward this *Advisor* to you? [Sign up for your own free 4 week trial](#).

Cutter Consortium | 37 Broadway, Suite 1, Arlington, MA 02474, USA. | Tel: +1 781 648 8700 | Fax: 781 648 1950 | www.cutter.com