

The Four Myths of Agile Development

(Why Agile ain't for everyone)



by John Langlois, PMP®, SCPM®
projectEZ Consulting

“*Don't force Agile on every project.*”

Software development teams are exploring new ways to create valuable programs. An Agile philosophy promotes a flexible approach with close client collaboration.

There are many reasonable ways you can execute an Agile philosophy in your organization. You are not obligated to follow a particular method, be it SCRUM, Agile Unified Process (AUP), or any other approach. The very essence of Agile is flexibility and, consequently, teams should be able to customize their processes within a general framework.

This special report will expose four common misconceptions about Agile:

Myth 1: There's only one way to do Agile

Myth 2: You don't need to know where you are going

Myth 3: One process fits all

Myth 4: Agile can save a crummy idea

What is Agile Software Development?

At the very end of the 20th century, programmers realized that the business world was changing faster than their ability to execute projects. Planning and executing monolithic legacy applications over a multi year time frame no longer met the needs of clients.

An Agile philosophy crystalized to meet the reality of this new world order characterized by constant change. This philosophy centers on these principles:

- Frequent communication with clients
- Frequent delivery of products
- Frequent process iterations

There is nothing particularly inflammatory in these philosophical principles. My trouble with Agile is when people try to translate these guidelines into rigid processes and methods. The architects of the Agile manifesto recognized that process rigidity can get you in trouble when they published these principles in 2001¹:



- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

That is, while we value all of the items in this list, we value the items in bold more. [paraphrased]

The architects were careful not to endorse any single method. The very first principle warns against holding to inflexible processes and tools. And this principle can certainly be applied to any Agile method.

¹ Agile Manifesto, <http://agilemanifesto.org/> (Dec. 22, 2008).

Also notice that the architects of the manifesto “value” certain things more than others. This careful choice of wording means that the Agile playlist is eclectic with a wide variety of songs. For example, every now and then, it may be in your best interest to follow a plan or to produce comprehensive documentation. That may not be your favorite or most popular approach, but it may very well suit a particular project circumstance. In other words, you are welcome to access a large library of methods that appeal to the unique mood of each project.

Myth 1: There’s only one way to do Agile

Project Managers value a definable, repeatable process. Perhaps that is why they want to turn Agile into a well documented method like Agile Unified Process (AUP), Feature Driven Development (FDD) or SCRUM². There are several accepted methods for Agile Development. Use whatever works for you. And permit self directed teams leeway to find the fastest way to complete the deliverable.

The principle of continuous process improvement demands that Agile methods evolve as we learn better ways to do things. It’s almost anti-Agile to think that there is only one “best way” to execute Agile principles.



² For more information about agile development methods see “Agile Software Development,” Wikipedia, Dec. 22, 2208 (http://en.wikipedia.org/wiki/Agile_development).

Myth 2: You don't need to know where you are going

Agile leverages frequent iterations to respond to change. At the front end of each development cycle, the team generates the plan for the next run. With Agile, you don't plan what you don't know so these sessions tend to be short, sometimes less than 4 hours.

For a minor release, this level of planning may be sufficient. Customer collaboration will prompt course corrections after each inspection of the deliverables. However, it is still prudent to set goals and directions for the program. Agile does not preclude you from laying out a high level release plan. This portfolio and program planning should be done outside of a development sprint.

Some Agile teams wander about aimlessly. They are like the little girl named Alice who was walking down a forest path when she came to a branch in the road.³ ... she was a little startled by seeing the Cheshire-Cat sitting on a bough of a tree a few yards off.

"Would you tell me, please, which way I ought to go from here?"

"That depends a good deal on where you want to get to," said the Cat.

"I don't much care where ----" said Alice.

"Then it doesn't matter which way you go," said the Cat.

³ Carroll, Lewis, *Alice's Adventures In Wonderland* (The MacMillan Company, New York, 1898), 56.

The Agile journey is less like this exchange and more like the trip Dorothy took to Oz. She didn't know what the city looked like. She didn't exactly know how to get there. But off she headed toward her goal in a certain direction. Likewise, you should have some idea where you want to go.

Myth 3: One process fits all

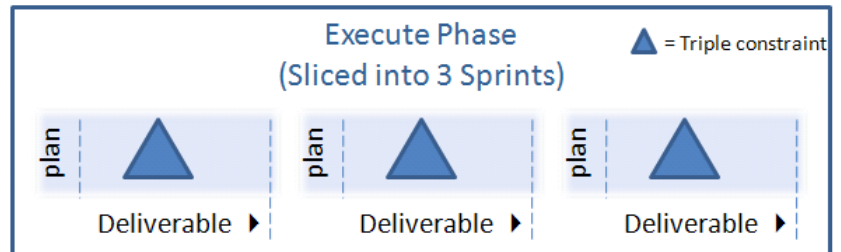
Why limit your process options? Agile Development, Extreme Programming, Waterfall, Spiral, Seat-of-Your-Pants, Rapid Application Development all have merit. Use whatever works best for the project at hand.

In fact, you could apply different processes during different phases of the same project. To illustrate, let's break down project delivery into four sequential phases: Concept, Develop, Execute and Finish. This follows the convenient mnemonic C, D, E, and F.

In the Concept phase, our goal is to determine if the proposal has merit. We're looking at market research. We are sketching order of magnitude estimates for the triple constraint of scope, time and cost. We are assessing our organization's core capability to deliver the project. We have not yet written a single line of code.

A waterfall style detailed work break down structure that outlines the steps to complete a robust concept decision check point package (with templates) might fit that phase perfectly well. Where's the need for Agile?

Moving along through the phase gates, we might slice the Execute phase into a series of sprints⁴ as shown below.



Each sprint begins with a short planning exercise that produces the prioritized list of requirements and the triple constraint for that specific run. If the triple constraint for our first sprint fixes all elements of scope, time and cost, then Agile may not be the best approach for that run. If the final two sprints permit some flexibility with scope and time, then Agile might be the best fit.

It is conceivable that you might apply different processes to each phase and to each sprint. Now that's flexibility!

You could employ a different process for each sprint.

⁴ sprint: short development cycle in the SCRUM software engineering method.

Many projects are doomed for failure the moment they leave the station.

Beware of applying Agile to some projects

Ernie Nielsen, Managing Director, Enterprise Project Management at Brigham Young University, recommends that you do not apply Agile to projects with these characteristics⁵:

1. The introduction to a unpredictable new technology
2. A project with no scope flexibility
3. A project with no process flexibility

There is also healthy debate in the literature as to applying Agile to large scale projects with a large number of developers. In any case, these debates indicate that Agile isn't a "one size fits all" approach.

Myth 4: Agile Can Save a Crummy Idea

Many projects are doomed for failure the moment they leave the boarding station. No amount of Agile can fix that.

Here's a few common causes of troubled projects⁶:

1. Executives place irrationally large bets on exceedingly risky projects.
2. Stakeholders can't agree on the final destination.
3. Project sponsors are unwilling or unable to move large obstacles on the tracks.

5 Ernie, Nielson, "Agile—Philosophy, Methodology, or Practice?," Lecture at Project World®, Orlando, FL, Nov. 20, 2008.

6 Langlois, John, Project Wreck, (JoRoJim Publishers, 2007), 3.

***Agile cannot save
a crummy idea.***

4. Employees throughout the organization lack the courage to cancel a “stupid” project.
5. Offerings are burdened with unnecessary scope.
6. Projects are not properly provisioned with the necessary resources to reach their final destination.

If you perform a rigorous Concept phase at the front end of your project, many of these triggers for trouble can be defused. But if your portfolio planning process is dysfunctional, don't expect Agile or a high performing team to save a crummy idea. Don't charter. Don't pass Go. Don't Agile a pig.

Let's be real. Some ideas are epically bad. Say, for instance, that you are a chef at a Japanese restaurant and the owner suggests that you add something bold to the menu:
Goldfish sushi!



In the long run, your interests and the business interests are both better served if you stand up and resist the proposal. Likewise, in a software development world, some proposals should be rejected. It would be unrealistic to think that Agile can transform every project proposal into a break out business success.

Get Agile off on the right foot

The truth is that you are hurting your Agile agenda if you apply it to a doomed project. You need to register some visible successes with Agile in an organization before you roll it out more broadly.

Be very judicious about picking your first pilot Agile project. Here's a few tips for ensuring the success of your first Agile project:

- Say "no" to bad ideas that you know in your heart stand zero chance of success.
- Select an "A-Team" of experienced programmers. Don't assign a gaggle of new employees to an Agile project or they'll wander around dazed and disoriented.
- Appoint a project leader who has a "serve and facilitate" personality. Do not appoint a leader with a "command and control" personality.
- Ensure that you have an executive sponsor who is willing and able to move large obstacles out of your path. If your sponsor is a junior executive weasel with his own pet project, stand on the tracks!
- Agree on a basic set of project status documentation before you start. That list may include things like requirements backlog, burn down rate, a list of obstacles, legal documentation and so forth. In the rush to the sprint finish line, you don't want to forget these things!
- Make sure that each deliverable passes a predefined quality checklist before you move to the next sprint.

Agile does not mean zero project documentation.

***Filtered communication
is worthless.***

Of course, you can't always pick your projects or the resources assigned to them, but there is one thing that you can always do. It's really a little thing that generates tremendous results. You can tell your executives the truth all of the time.

In my mind, the core of an Agile philosophy is this:

Frequent, honest, unfiltered, straight up, no pounding sunshine up your arse communication.

Communication that is filtered by a team member or by an executive is worthless no matter how frequent.

Most executives will value open and honest communication. They want to help you succeed because it is in their best interest to do so. The invisible hand of self interest works in your favor. Furthermore, many executives appreciate the courage required to reject a bad project proposals.

Take the risk to be open and honest and you will see rewards.

Summary

You should not apply Agile to every project.

Agile may not be the best choice to pilot projects with well defined scope or government sponsored projects that must be executed a certain way. Agile may not fit cookie cutter projects that have a rich history of success using a different approach. After all, if an approach is already working well, why Agile it?

If you decide to use Agile on a project, then remember that there are a variety of processes and methods that fall under the Agile umbrella. Don't believe that there is only one right way to do it. In fact, you and your team may contribute to the Agile body of knowledge by lending your experiences, tips and tricks to the process toolkit.

The good news is that Agile doesn't require a wholesale and instantaneous shift in your organization. Pilot a project or two using the Agile philosophy. And remember that teams will be reluctant to use a new approach when it is forced upon them. So be flexible. That's the Agile philosophy at a molecular level.

About

John Langlois

John Langlois is a visionary project manager who exercises wisdom to break down process barriers and cultural resistance to change.

John led the team that planned and delivered one of the most successful programs in the history of notebook computing - ThinkPad® T-series. His projects received numerous Editor Choice product awards while enjoying the highest levels of customer satisfaction.

John earned IBM's most prestigious corporate award for outstanding project performance. His work has been featured at ProjectWorld® & World Congress for Business Analysts® and PMLessonsLearned. His workshop was rated as a top 10 value at IBM's Technical Leadership exchange.

Contact us at www.projectEZ.com to see how we can help your organization improve project delivery.



Photo Credits

Page 3: Agile iPod playlist created by John Langlois.

Page 4, 5, and 9: Goldfish photos were purchased from istockphoto.com

All rights reserved. No part of this paper may be used or reproduced in any manner whatsoever without written permission except in the case of brief quotations embodied in critical articles and reviews.