



W7

Class

6/10/2009 10:00:00 AM

# "In Defense of Waterfall: Deconstructing the Agile Manifesto"

Presented by:

Kenneth Katz  
DST Output

Presented at:

Better Software Conference & EXPO 2009  
Las Vegas, Nevada



330 Corporate Way, Suite 300, Orange Park, FL 32073  
888-268-8770 · 904-278-0524 · [sqeinfo@sqe.com](mailto:sqeinfo@sqe.com) · [www.sqe.com](http://www.sqe.com)

# Ken Katz

Educated as an aerospace engineer at MIT and the University of Michigan, Ken Katz spent the first part of his career in the US Air Force and the aerospace industry. With the end of the Cold War and the downturn in the aerospace industry, he considered new careers as a circus clown and a country music superstar before settling on project management. For the past eight years, Ken has been employed by DST Output, managing projects in software development and information technology. Ken has earned the PMP certification from the Project Management Institute and has taught project management courses at the graduate level.

# **In Defense of Waterfall**

Deconstructing the Agile Manifesto

---

Kenneth P. Katz

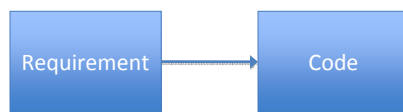
Better Software Conference and Expo  
June 2009

## Topics

- How did we get to agile?
- Critical analysis of the Agile Manifesto.
- Improving waterfall, using agile techniques.

## How Did We Get To Agile?

- Once upon a time, software development processes were ad hoc (“cowboy coding”).



## Why Cowboy Coding Worked

- Simple.
- No overhead.
- Business user (product owner) communicated directly with the developer.



## Why Cowboy Coding Failed

- No economy of scale.
- No coordination between requirements.
- Overwhelmed by complexity.
- Requirements were not correctly understood by the programmer.
- Devolved into code-and-fix.

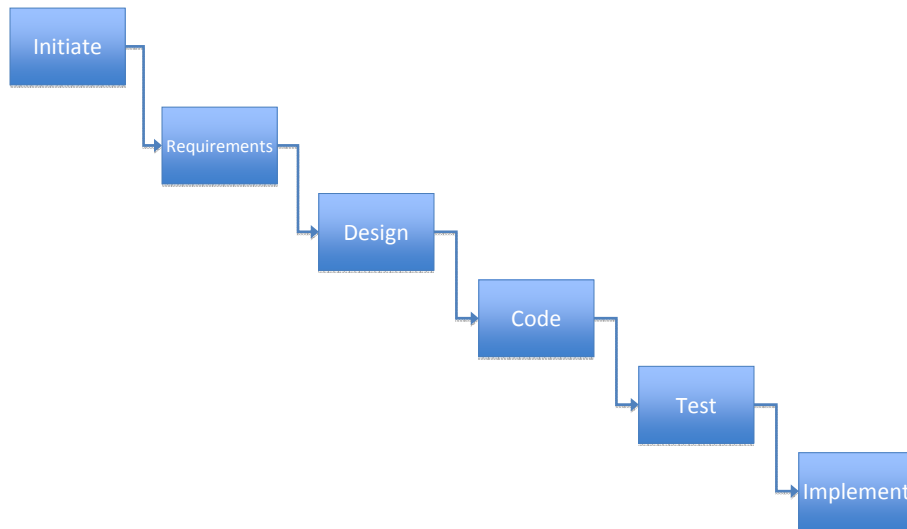


## Waterfall

- Structured and logical.
- A proven process for engineering projects such as construction that were thought to be analogous to software development.
- Promised predictability and control.
- Used the division of labor and specialization.



## The Classic Waterfall Process



## Waterfall: What Went Wrong

- The use of waterfall led to a series of expensive software development project failures.
- Several thought leaders in the field of software development identified fundamental flaws with waterfall.



## Ken Schwaber

- Software development is too complex for *defined process control*.
- It requires *empirical process control* based on visibility, inspection and adaption.



## Johanna Rothman

- It is impossible to forecast everything you need to know about the future.
- Serial life cycles predict the future, without having sufficient data to check that the future can be accomplished based on current work.



## Mike Cohn

- Uncertainty is inherent and inevitable in software projects and processes.
- Planning by activity doesn't work well.
  - Activities don't finish early.
  - Lateness is passed down the schedule.
  - Activities are not independent.



## Craig Larman

- We want the requirements to be stable, but they aren't.
- The waterfall aggravates complexity overload and analysis paralysis.
- Predictive planning is only suitable for low change, low complexity projects.





## Michele Sliger and Stacia Broderick

- The tasks involved with building software, day to day, cannot be predicted and certainly are not repeatable.
- Attempting to predict task-level detail in such mayhem is waste, and formulating a date off of this flawed prediction is setting up a project team for failure.



## Agile: The Solution

- A consensus emerged that waterfall methods for software development were fundamentally flawed.
- Agile (incremental, iterative) methods emerged.
- The Agile Manifesto brought together many leading thinkers and advocates.

## The Agile Manifesto

- We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:
  - 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 there is value in the items on the right, we value the items on the left more.

## Reality Didn't Match Theory

- I managed a series of projects using waterfall that were successful.
  - The notable failure had an ill-conceived business case and unrelated to methodology.
- I managed a project using agile and it was not particularly successful.
- What could explain these results?

## Let Me Ask You

- In your experience, what are the root causes of successes and failures:
  - Using cowboy coding?
  - Using waterfall methodologies?
  - Using agile methodologies?



## The First Explanation

- I'm an old school project manager too tied to my PMI/PMBOK®/PMP® view of the world.
- I don't think so.
  - I was and remain an advocate for the adoption of agile in my organization.
  - Michele Sliger and Stacia Broderick have shown that there is no contradiction between PMI/PMBOK®/PMP® and agile.
  - This would explain the failure of the agile project but not the success of the waterfall projects.

## Agile Manifesto vs. Reality

- Principle behind the Agile Manifesto:  
Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Reality
  - Commitments often are made without the input of the project team.
  - Changing requirements mean that commitments cannot be kept.

## Agile Manifesto vs. Reality

- Principle behind the Agile Manifesto: Business people and developers must work together daily throughout the project.
- Reality
  - The top priority of business people is their “day jobs”.
  - Their contact with the project team is sporadic.

## Agile Manifesto vs. Reality

- Principle behind the Agile Manifesto: Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- Reality:
  - Most project team members are multi-tasked (sad but true!).
  - Shared services have long lead times to service requests.

## Agile Manifesto vs. Reality

- Principle behind the Agile Manifesto: The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Reality
  - Virtual teams preclude routine face-to-face conversation.
  - Technology does not provide the full benefits of face-to-face communications.

## Agile Manifesto vs. Reality

- InfoWorld columnist Bob Lewis claims that agile methods and offshore development are incompatible.
  - Sensory limitations.
  - Interface distractions.
  - Time zones.

## Agile Manifesto vs. Reality

- Principle behind the Agile Manifesto: The best architectures, requirements, and designs emerge from self-organizing teams.
- Reality
  - Project team members are assigned, not chosen by the project manager.
  - Project teams must work with who they have, not who they want.
  - Some people can't or won't self-organize.

## The Waterfall Manifesto 😊

- No software development methodology is an panacea.
- All methodologies are based on certain underlying premises that are not necessarily correct in all circumstances.
- Waterfall is appropriate for certain projects, and can be improved with concepts from agile.



## When May Waterfall Be Appropriate?

- Firm commitments exist to deliver specified scope by a specified date.
- The requirements are well-defined and stable.
- The solution is well-defined and has low risk.
- Organizational issues:
  - 50% of people are below average.
  - Geographically-distributed project team.
  - Multi-tasked (time-sliced) project team members.
  - Extensive external dependencies.

## The Benefits of Waterfall

- The project team calculates what it needs to meet imposed commitments.
- Business people know when they must participate in the project and what is the effect if they do not participate when scheduled.
- Multi-tasked team members can plan their participation in ways that allow them to meet their multiple commitments.
- Reduced dependence on face-to-face communications.
- It gives specific tasks that must be completed by specific dates to people who are not self-starters.

## Devising a Better Waterfall

- That the principles underlying the Agile Manifesto are not always valid does not mean that the critiques of waterfall are invalid.
- The tools of agile can be selectively applied to improve waterfall.





## Develop in Iterations

- Break projects into smaller iterations (preferably 1-3 months duration).
- The sum of the iterations is the total scope of the project.
- Smaller iterations are easier to estimate, plan and understand.
- Smaller iterations reduce risk.
- Each of the smaller iterations can be a waterfall.



## Rolling Wave Planning

- Also known as progressive elaboration.
- Near-term project activities are planned in great detail.
- Farther out in time, tasks are only defined in at high level.
- Rolling wave planning is a natural complement to developing in short iterations.



## 15 Minute Daily Meeting

- Identical to the Daily Scrum.
- Agenda:
  - What have you done since last meeting?
  - What do you plan on doing between now and the next meeting?
  - What impediments to progress are you facing?
- Conducted as a conference call for geographically distributed teams.
- Particularly useful for geographically distributed teams which cannot benefit from informal “over-the-cubicle” communications throughout the work day.



## Customer Involvement

- Full-time involvement by what Scrum calls a Product Owner is desirable but often is not going to happen.
- Even so, the more customer involvement the better.
- Develop requirements with customers in a face-to-face meeting that has both customers and the project team.
- User Stories in combination with user interface mockups are a much better way to obtain customer input to requirements than traditional “the system/application shall ...”.
- Set up an extended project team that includes customers. Keep them in the loop and consult them when required.



## Face-To-Face Communications

- For distributed teams, face-to-face communications implies costly (in time and expenses) travel.
- Face-to-face meetings are still the best way to conduct highly interactive communications.
- Have one face-to-face meeting per iteration, focusing on requirements, conceptual design and project planning.



## Collaborative Planning

- Top-down “command and control” project management is not a characteristic of waterfall, it is a characteristic of bad project management.
- PMBOK® Guide, 4<sup>th</sup> Edition (2008)
  - ... the involvement of all team members in project planning and decision making can be beneficial.
  - Early involvement and participation of team members adds their expertise during the planning process and strengthens their commitment to the project.

## Collaborative Planning

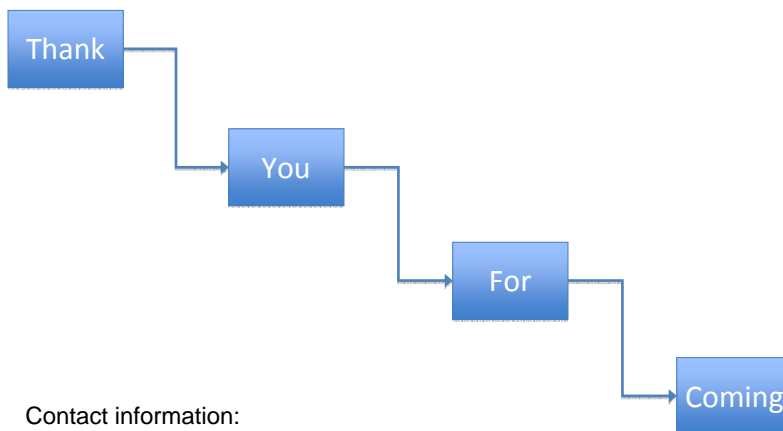
- When planning a project, a project manager's primary roles are facilitator and integrator.
- A project plan created without the input (and therefore the expertise and buy-in) of the people who will execute the plan is likely to fail.

## What Works for You?

- Are there any other techniques that you have successfully used to improve waterfall?
- Place stickers on the chart to vote for the techniques that you think are most important.
  - 5 stickers per participant.
  - Place 0-5 on each technique (total of five).
  - After the stickers are placed, they will be counted.

## Pick The Right Tool for the Job

- The problems with waterfall are real. For most projects, agile is a better methodology.
- Agile is valuable but not a panacea. The necessary conditions for the application of agile are not always present.
- Waterfall can be a viable alternative to agile in some circumstances.
- Waterfall can be improved by the use of certain techniques usually associated with agile.



Contact information:  
kpkatz@dstoutput  
(860) 290-7181

## Resources

Cohn, Mike. *Agile Estimating and Planning*. Prentice Hall, 2006.

Larman, Craig. *Agile & Iterative Development: A Manager's Guide*. Addison-Wesley, 2004.

Rothman, Johanna. *Manage It! Your Guide to Modern, Pragmatic Project Management*. The Pragmatic Bookshelf, 2007.

Schwaber, Ken. *Agile Project Management with Scrum*. Microsoft Press, 2004.

Sliger, Michele and Stacia Broderick. *The Software Project Manager's Bridge to Agility*. Addison-Wesley, 2008.