

# Agile in the Trenches

---

A Technologist's View of  
Agile in the Real World

# Who am I?

- Senior Technical Lead at LL Bean
- 20 years professional software development experience
- 10+ years working on teams at LL Bean following Agile methodologies to varying degrees
- Rock Star (obviously)



# What is Agile?

- Continuous delivery of small pieces of working software
- Work comes to a standing team; teams are not assembled for particular projects
- Project plan broken into short time periods called sprints
- Work is broken down into chunks: Epics, Stories, and Sub-tasks



# Manifesto for Agile Software Development

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.



# Unpacking the Manifesto

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





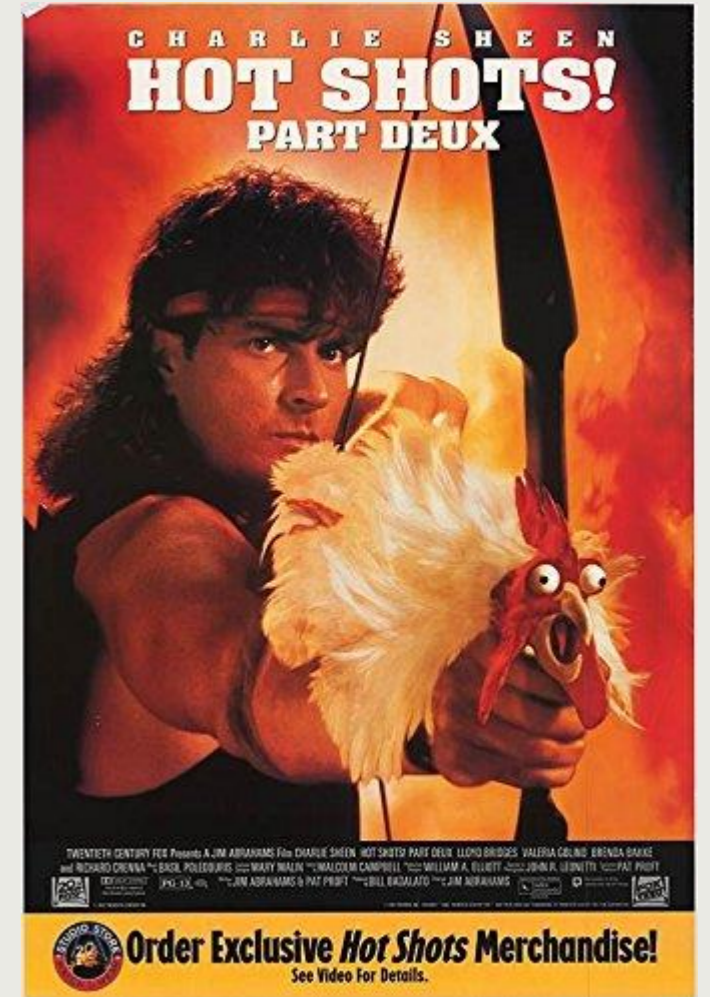
# Principles behind the Agile Manifesto, Part 1

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.



# Principles behind the Agile Manifesto, Part Deux

- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.



# Principles behind the Agile Manifesto, Part 3

- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.





# Principle 1

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

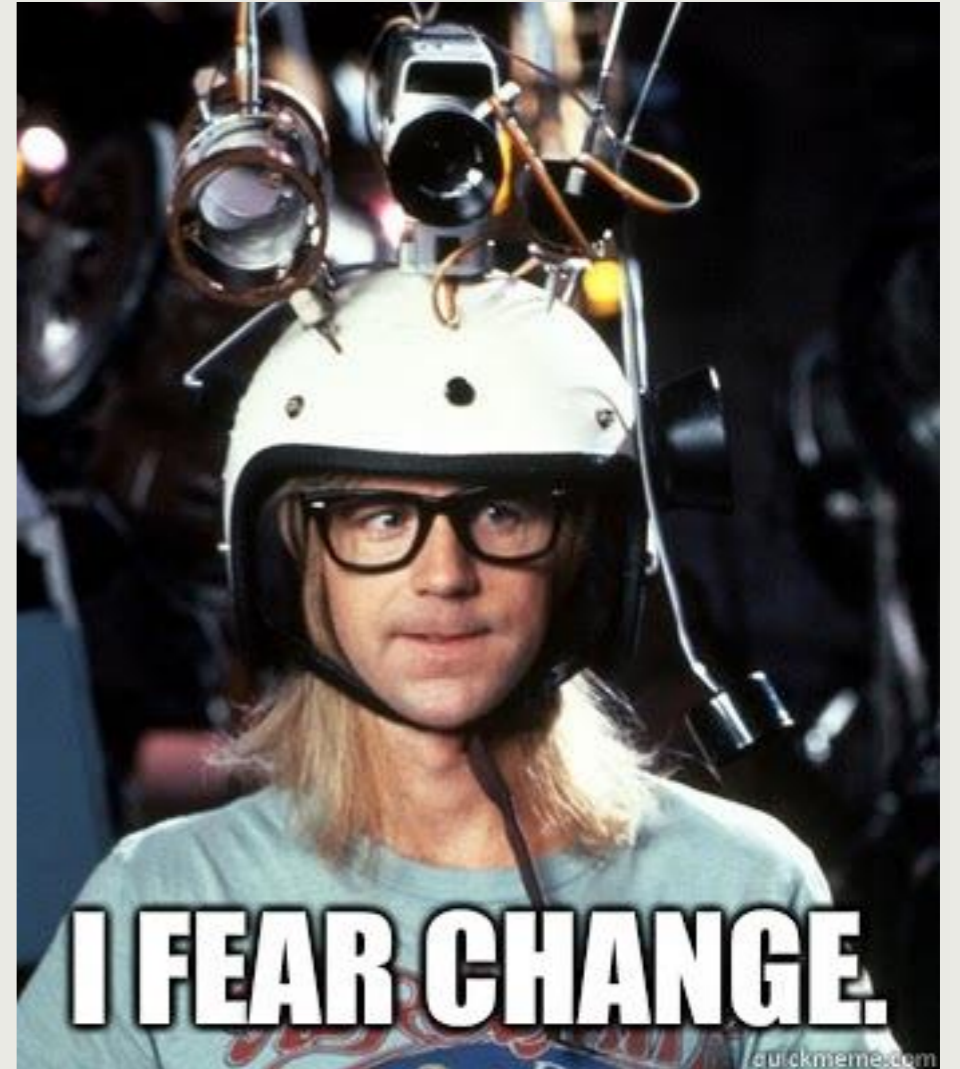
- Make sure you have in-house expertise in devops, or have management buy-in to hire and/or develop, or else this will be a painful process



## Principle 2

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

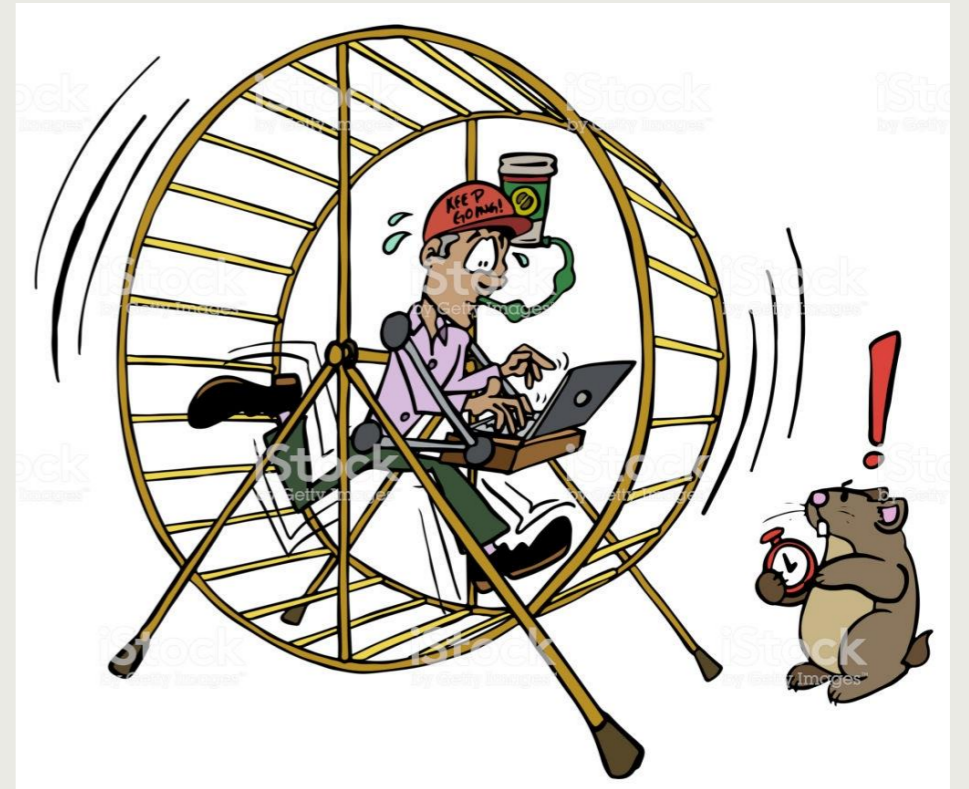
- Shift from getting all requirements nailed down up front to only defining enough to get started
- Understand the concept of Minimal Viable Product (MVP) and start with that



# Principle 3

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

- Shift from getting all requirements nailed down up front to only defining enough to get started
- Don't try to go here all at once. Work towards it incrementally -- go from once or twice a year to quarterly first.





# Principle 4

Business people and developers must work together daily throughout the project.

- Have daily check-ins that involve whole team.
- Keep them brief. 15 min preferably, 30 min max.
- Each person answers three standard questions
- Vary format based on where you are in the sprint



# Principle 5

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

- Team makeup is important; you want a good balance of developers, QA testers, and analysts.
- If you have too few QA testers, QA can quickly become a bottleneck. This is a common problem
- Scrum master can be either a Project Manager or a Technical Lead. Either can work, but technical understanding is a valuable attribute for a scrum master to have.

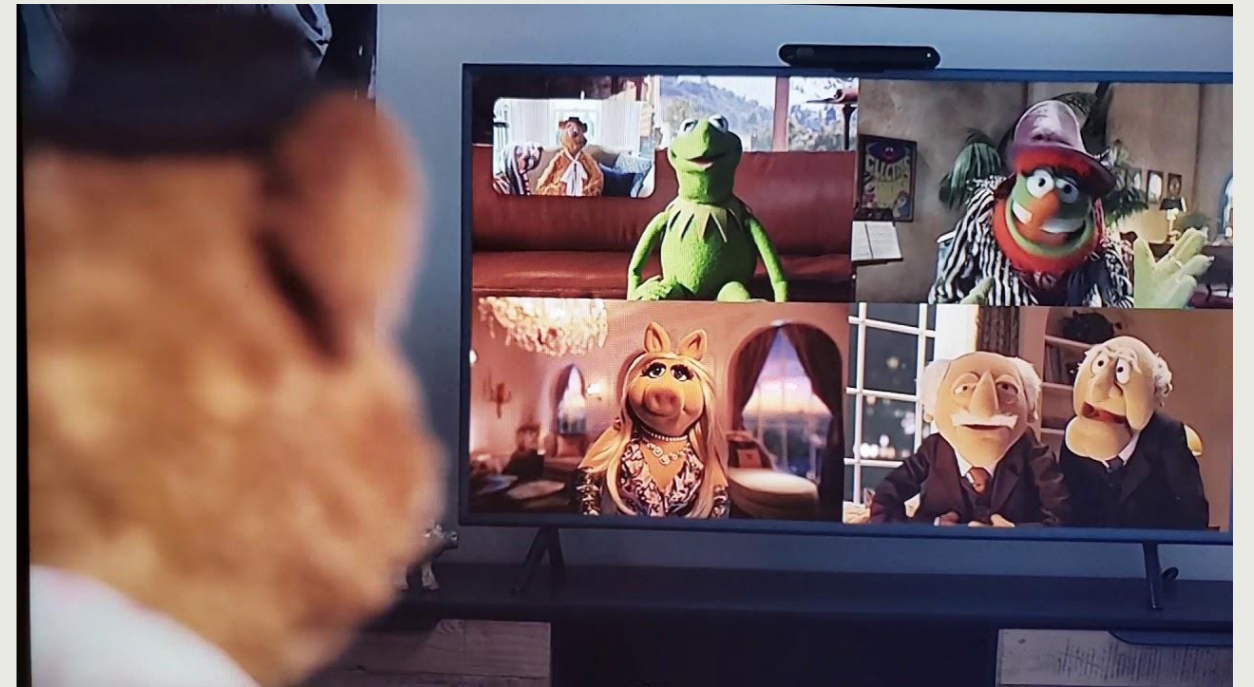




# Principle 6

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

- This is not always practical or possible. I currently have team members in Maine, Costa Rica, India, California, Washington, and Oregon, for example.
- Technology to facilitate on-line collaboration is essential
- Where there is critical mass, try to have team members sit next to or at least near each other



# Principle 7

Working software is the primary measure of progress.

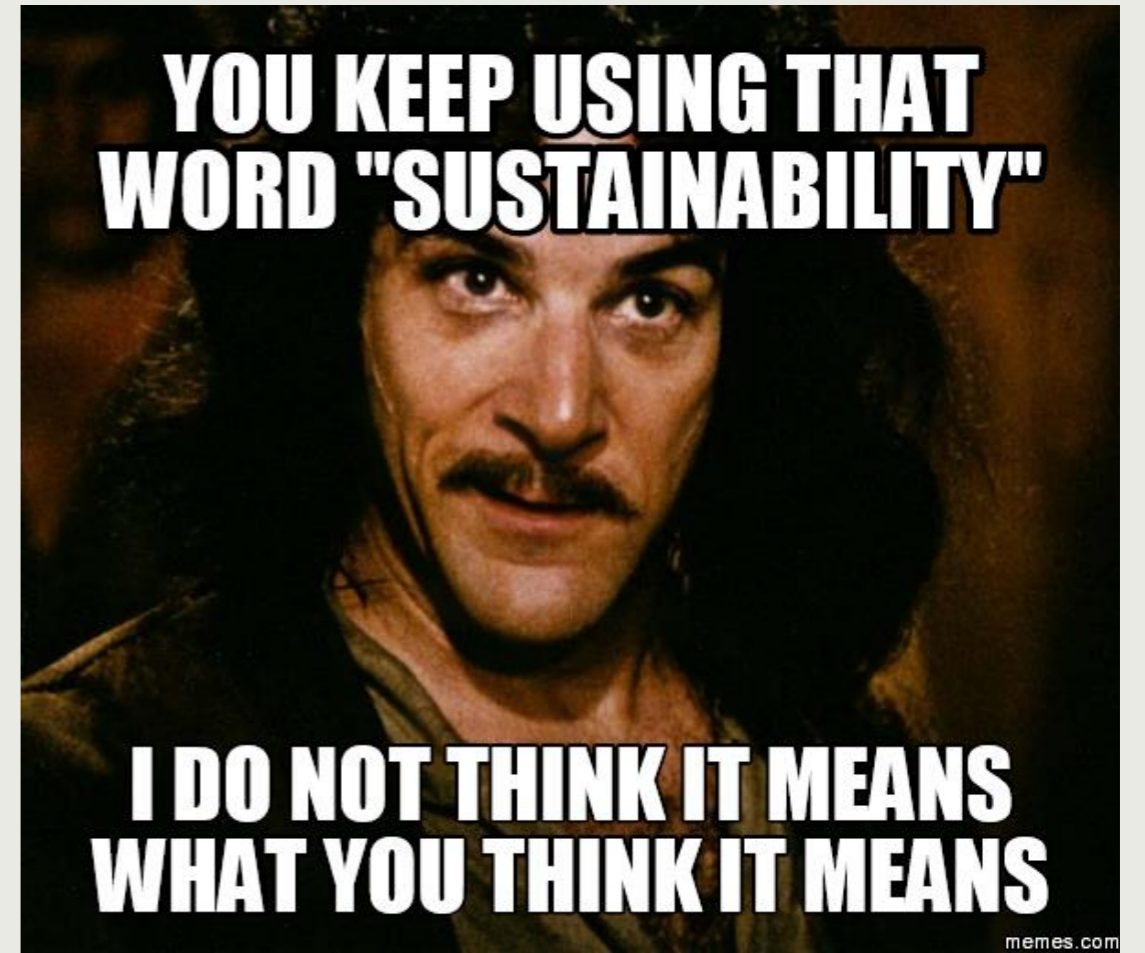
- At the end of each sprint there should be something working and deployed to a shared environment, even if it's just a test/development environment
- Working software should be demoed at a Sprint Demo meeting, which should be scheduled at the end of each sprint and should include business stakeholders and interested parties outside the team
- Software to be deployed to production should be reviewed in a prep meeting near the end of the sprint to ensure that the deployment plan is solid



# Principle 8

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

- Develop and maintain good practices around estimating and planning
- Have a process to ensure the right things are in the backlog in the right order
- [www.pointingpoker.com](http://www.pointingpoker.com) is a useful tool
- Track productivity over time





# Principle 9

Continuous attention to technical excellence and good design enhances agility.

- Schedule regular meetings to review technical designs and code -- "Code Safari"
- Ensure that each development story has a design subtask and a subtask for the team to review the design



# Principle 10

Simplicity--the art of maximizing the amount of work not done--is essential.

- Stories should be groomed regularly by business analysts/owner and senior members of the technical team to "weed out" unnecessary work and re-evaluate priorities





# Principle 11

The best architectures, requirements, and designs emerge from self-organizing teams.

- The team should be given the latitude to do so, with only high level direction from outside the team
- The team still needs to adhere to organizational architectural directions.



# Principle 12

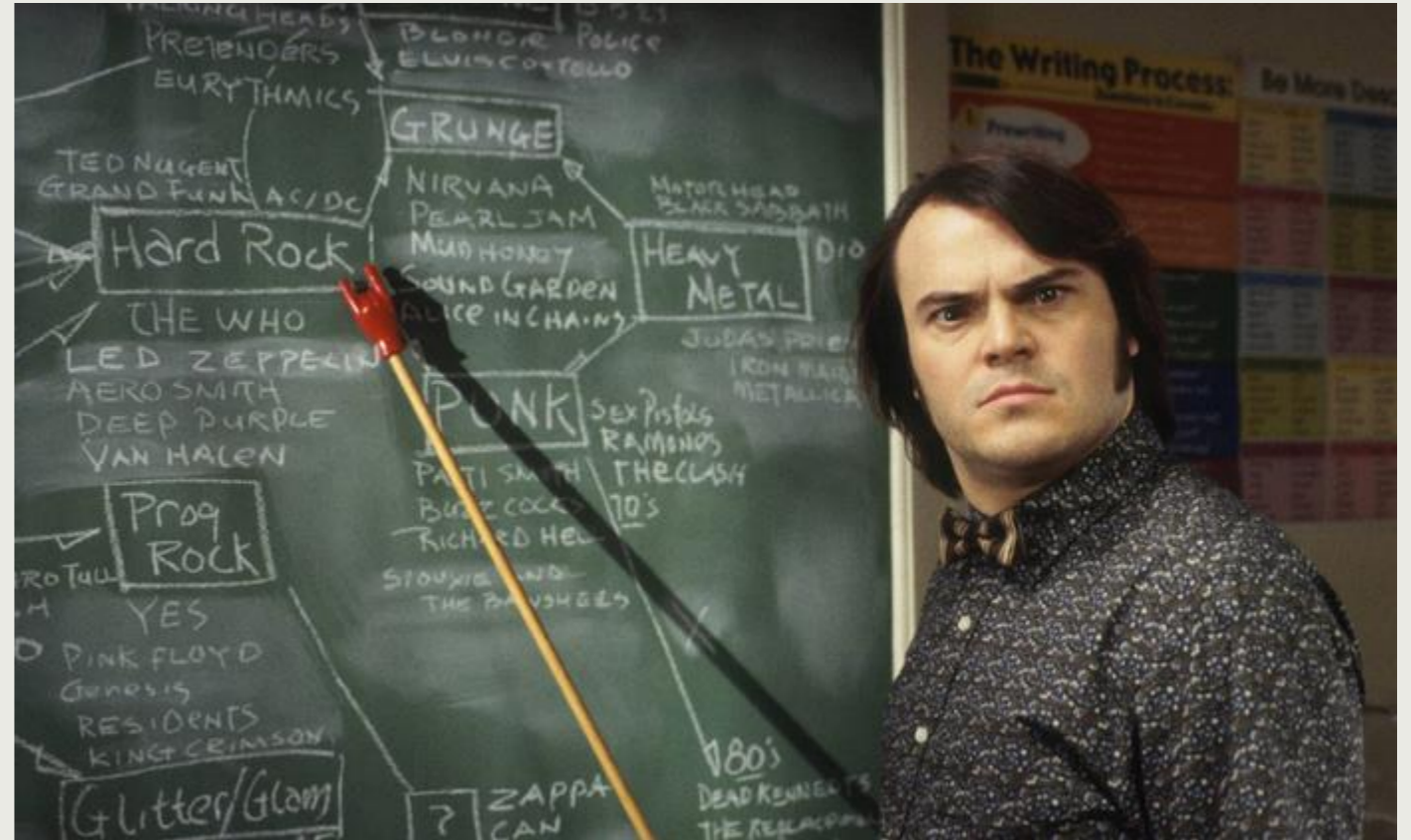
At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

- Retrospective after every sprint!
- Less focused on the actual work that was done than on how that work was done
- Identify action items for three different categories: Start Doing, Stop Doing, Keep Doing



# Lessons Learned

- A transition to Agile can't happen with a flip of a switch.
- Start small and get bigger
- Be flexible, be patient
- Don't be afraid to fail
- Be a good human



# Want to Learn More?

- Agile Maine is an organization dedicated to the promotion of Agile practices in the state
- Monthly meetups around the area
- Agile Maine Day annual conference each spring at USM
- FMI: [www.agilemaine.com](http://www.agilemaine.com)



● Questions?