

AVOIDING SPEED BUMPS ON THE ROAD TO AGILE ADOPTION

If you don't like change, you're going to like irrelevance even less.

Gen. Eric Shinseki

ADP East - Nov 2010

Don Gray - Agile Anthropologist



Agenda

- Gather and group speed bumps
- Review Assumptions
- Review speed bumps and what to do
- Wrap-up



Gather Speed Bumps

- Speed bump - a ridge set in a road surface, typically at intervals, to control the speed of vehicles.
- Pick a partner and share your experiences where you hit a speed bump in your agile adoption. Make notes one per post-it.



The Agile Manifesto (2001)

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.

www.agilemanifesto.org



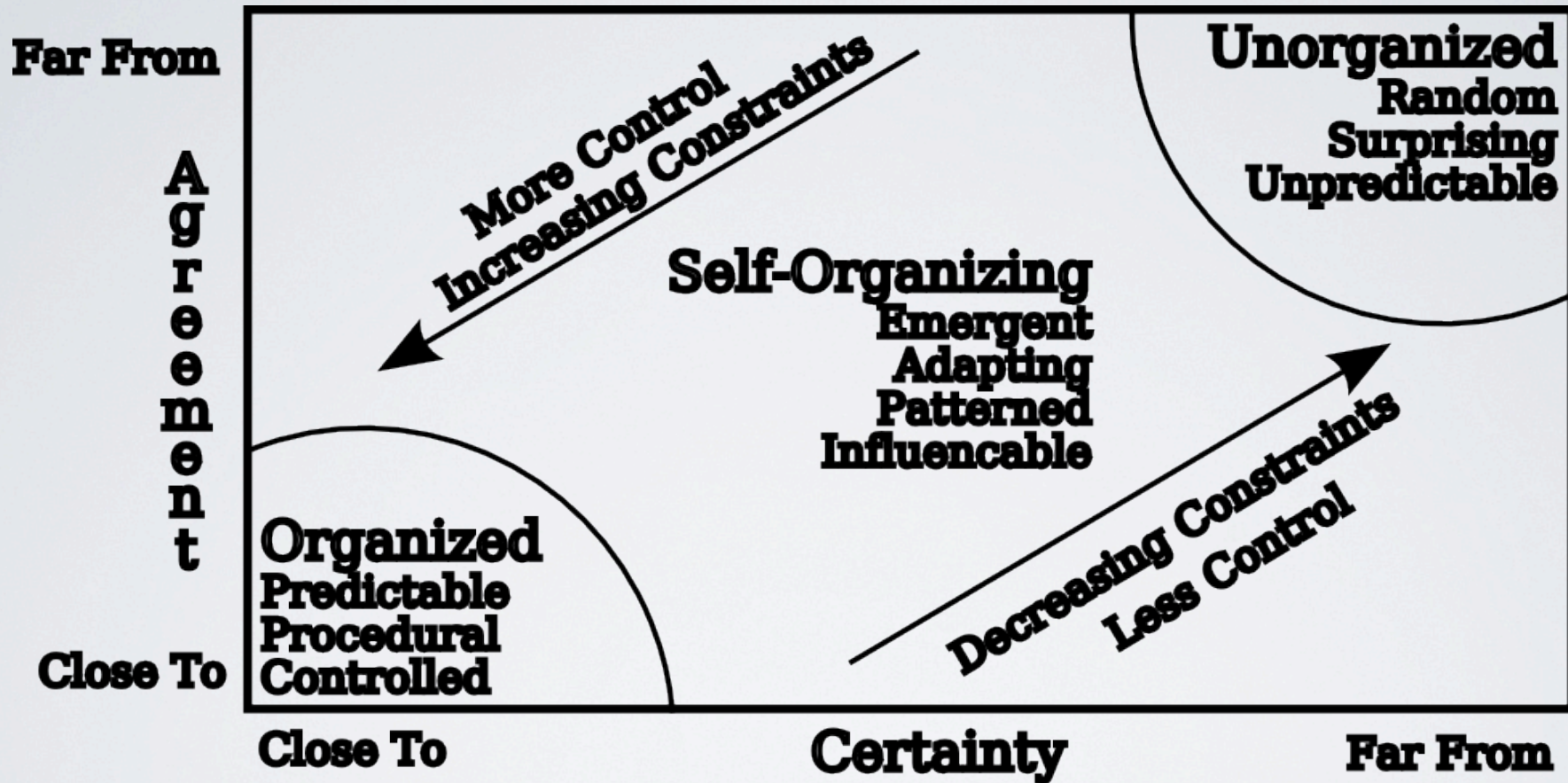
Process Assumptions

- Ziv's law - specifications will never be fully understood
- Humphrey's law - the user will never know what they want until after the system is in production (maybe not even then)
- Wegner's lemma - an interactive system can never be fully specified nor can it ever be fully tested.
- Langdon's lemma - software evolves more rapidly as it approaches chaotic regions (taking care not to spill over into chaos)

Jeff Sutherland



COMPLEX SYSTEMS



Influencing Patterns for Change



People Assumptions

- People are communicating beings, doing best face-to-face, in person, with real-time question and answer.
- People have trouble acting consistently over time.
- People are highly variable, varying from day to day and place to place.
- People generally want to be good citizens, are good at looking around, taking initiative, and doing “whatever is needed” to get the project to work.

Alistair Cockburn



Systems Principles

- Increase Bandwidth
- Match impedance
- Decrease Delays
- Reduce Friction
- Single Responsibility
- Clarify Target



Speed Bumps

- Book-ended Agile
- Team composition / location
- Combined roles (Manager AND ScrumMaster)
- Repurposed resources
- Bottom up/Top down implementation
- Self-governing teams



More Speed Bumps

- Change control process for user stories in the product backlog
- Changing sprint contents
- Changing sprint lengths
- Gaming the Product Owner
- Release Planning



12 Agile Principles

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



12 Agile Principles

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



12 Agile Principles

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



Half Way There

Manifesto for Half-Arsed Agile Software Development

We have heard about new ways of developing software by paying consultants and reading Gartner reports. Through this we have been told to value:

Individuals and interactions over processes and tools
and we have mandatory processes and tools to control how those individuals (we prefer the term 'resources') interact

Working software over comprehensive documentation
as long as that software is comprehensively documented

Customer collaboration over contract negotiation
within the boundaries of strict contracts, of course, and subject to rigorous change control

Responding to change over following a plan
provided a detailed plan is in place to respond to the change, and it is followed precisely

That is, while the items on the left sound nice in theory, we're an enterprise company, and there's no way we're letting go of the items on the right.

Cobbled together one Saturday morning before breakfast by [Kerry Buckley \(@kerryb\)](#), following [an article](#) by Ron Jeffries and [this suggestion](#) from Eastmad.

<http://www.halfarsedagilemanifesto.org/>



For More Information

- Influencing Patterns for Change, ©2008, Hollady/Quade, ISBN 1440415927
- Fearless Change: Patterns for Introducing New Ideas, Mary Lynn Manns, Ph.D., Linda Rising, Ph. D. ©2005 Pearson Publishing ISBN 0-201-74157-1
- Systems thinking, teams, communication, and models:
 - Articles: <http://www.donaldegray.com/category/article/>
 - Blog posts: <http://www.donaldegray.com/category/blog/>

