Agile, Fragile, Tragile

Smells I have smelt &
Some ideas on fumigating your environment

Shane Hastie
Chief Knowledge Engineer & Agile Practice Lead

Hello

• Passionate agilist since before agile was born
• Building software this way since about 1985
• Found my way to XP in 2000 and discovered my community
• Educator and coach since 2002
• Agile Alliance board member since 2011
• Look after the Manifesto Translation Program
A Map for Our Journey

• Do Your Agile Our Way
• Process Über Alles
• We have Customers?
• Are we a Team?
• It’s not Work Unless we’re Writing Code
• Testing is Easy – Anyone can do it
• We’re Agile, so we . . .
• Your Walls Tells Stories
• Faster – Better – Cheaper?

“Your mileage will vary”

Do Your Agile Our Way

• KPI’s
• Silo structures
• Temporary teams
• People are “resources” or “assets”
• Don’t let them slack off – 100% load
Challenge the Status Quo

• Leverage allies
• Show results
• Push back
• Build slack into the process & innovation emerges
• Subvert (e.g.: reform the same teams)

Process Über Alles

• Velocity as a target and lever
• “Ready for” status
• % Complete reporting
• “The book says . . .”
• All those meetings
• Standups or sitdowns?
Shu-Ha-Ri

- Clarify what velocity means
- Don’t accept the pressure
- Identify the bottlenecks and fix them
- Knife-edged sharp Definition of DONE
- Workshops – not meetings, ruthlessly reject waste!
- Adapt to the realities

We have Customers?

- Stakeholders “too busy” to collaborate
- “You know what I want – just give it to me”
- “Everything is critical”
- Estimation done to the team, not by the team
- Product Owner owns nothing
Make Collaboration Easy

• Smother them with kindness
• Constant and rapid feedback
• Use the language of risk
• Find ways to help
• “Never attribute to malice that which can be explained by incompetence”
• Stop calling our customers “users”!

Are we a Team?

• It’s not my job
• We’re so polite to each other
• “That’s just her way – she doesn’t mean it”
• We don’t have permission to do that
• All hail the mighty scrummaster
• We don’t need facilitation
• Retrospective Groundhog day
Make Teamwork a Discipline

- Everybody is responsible for getting to DONE
- Practice servant leadership
- Tackle the tough interpersonal issues
- Ask hard questions
- Take ownership – forgiveness vs. permission
- Get help from outside if needed (even just another team)
- Retrospectives must result in concrete changes starting tomorrow

It’s not Work Unless we’re Writing Code

- Rush to code
- Testing squeezed into the end of the iteration
- Paperwork left behind
- “Works on my machine”
Setup for Success

- Explicit definition of DONE that includes all the work needed to produce a production-ready, fully tested, documented, compliant valuable product increment
- Value is not earned unless it is DONE
- Everyone must take ownership for getting to DONE, not for just doing their own tasks

Testing is Easy – Anyone can do it

- We don’t need “real” testers
- We can automate everything
- We’re doing TDD - unit tests are enough
- Why waste time just bashing away at the keyboard?
- The users are too busy – we’ll do it for them
Testing Needs Skills

- Testing is a creative (structured) activity
- Understand risk and quality goals
- Need to understand test approaches—boundary values, equivalence partitions, flow, black- & white-box, exploratory, checklists, heuristics
- Multiple testing levels
- Automation at all levels needs careful design
- Engineer for testability
- Lots of tests <> good tests

We’re Agile, so we . . .

- Don’t do no stinkin’ requirements
- Don’t need any documents
- Let the architecture “emerge”
- Know what the users want
- Just need to worry about the code and refactor
Craftsmanship Matters!

• Building good products is hard work!
• Standards and quality targets included in Definition of DONE
• Peer review or pair programming
• Create an expectation of craftsmanship

Your Walls Tell Stories

• Stories, stories everywhere
• No backbone
• Everything is fully elaborated before any “work” begins
• Bottlenecks in the flow
• No WIP limits
• Bleeding with bugs not fixed
Learn to Read the Patterns

- Map your backbone/ know your epics
- Use other models too – simple process models provide the end to end picture
- Zero-tolerance for defects – when a bug is found fix it now
- Measure your flow, and bring in WIP limits
- Ensure all work is visible, and all aspects of the work are included.
- Definition of DONE

Agile = Faster + Better + Cheaper?

- It sure is possible, but it ain’t easy!

- Requires a whole team commitment to the disciplined application of Agile values and principles supported by good social, cultural and technical practices
Thanks, and keep in touch

shaneh@softed.com
blog.softed.com
@shanehastie