Perspectives on Agile from a non-agile person

How I went from hating agile to embracing it.

Kevin Finkenbinder, MSU Libraries

Presentation to the MSU Research Software Developers Group, September 25, 2024

Agenda

- Agile / Scrum conceptual introduction
- Why I resisted Agile
- Experiences in transitioning to Agile/Scrum
- Benefits of agile/scrum
- Detriments of agile/scrum
- Key to success with agile/scrum.

Side note:

- While Agile and Scrum are most commonly viewed in connection with software development, they can be applied to many other realms where it is important to create deliverables on a timely basis.
  - e.g. Agile Architecture, Agile Manufacturing, Space X
  - See: https://www.geeksforgeeks.org/can-agile-be-used-outside-of-software-development/, https://dcmlearning.ie/scrum-resources/top-8-industries-that-are-adopting-scrum-other-than-software.html
  - Other potential uses – space planning, product development, research projects

Scrum

- First described in 1986 Harvard Business Review paper: *The New New Product Development Game* comparing product development to a rugby team.
- Refined by Schwaber and Sutherland in 1995.

Scrum Principles

- Break work into small goals
- Accomplish in sprints
- Daily micro-meetings (standups)
- Iterative Development
- Continuous Feedback

Scrum Roles

- Stakeholders
  - People who will eventually use or be affected by the product
- Product Owner*
  - Business/end user needs focused. Liaison with stakeholders
  - Conveys RAID to team and to stakeholders
    - Risks, Assumptions, Impediments, Dependencies

Scrum Roles (2)

- Scrum Master*
  - Educates on scrum practice/theory, communication facilitation specialist within the team, planning, objective setting, backlog management.
- Developers
  - Cross functional team!!! And often Cross unit.
    - Designers, Coders, Systems and any others needed
  - Determine what needs done in what order to achieve goals set by PO/SM

Related Items/Terms

- Kanban – A methodology focused on visualizing your work in progress as an efficient flow.
- Backlog – List of tasks that need to eventually be accomplished
  - PO should prioritize tasks (low, normal, high, critical, etc.)
  - Team does not have to follow priority in choosing tasks, just consider it
- AC – Acceptance Criteria
  - What needs to be done before this ticket is complete?
- Sprint Board – Tasks currently being worked on and their progress state (often a Kanban board)

Scrum Workflow

- Sprint Planning
  - Collaboratively determine next development goals for sprint out of the backlog
- Sprint
  - Set length of 2-4 weeks depending on team preferences.
  - Goal of a functional deliverable at end of sprint

Scrum Workflow (2)

- Sprint (2)
  - Daily standup
    - No more than 15 minutes
    - Can be done asynchronously, but this NEVER works as well.
    - Progress reports
    - Issues faced

Scrum Workflow (3)

- Sprint (3)
  - Team Time/Breakout Sessions*
    - Collaboration/Paired Programming
- Sprint Review
  - Time to brag AND get feedback.
  - “Public” is invited including any known stakeholders and other interested parties.

Scrum Workflow (4)

- Sprint Retrospective
  - Sprint strengths and challenges
  - Future improvement areas
  - Specific plans to try.
- {BREAK}
  - PO – Backlog Refinement (what's on the list, AC, etc.)

Agile

- Developed in 2001 in *Manifesto for Agile Software Development*
- Values*
  - **Individuals and interactions** over processes and tools
  - **Working software** over comprehensive documentation
  - **Customer collaboration** over contract negotiation
  - **Responding to change** over following a plan
- Draws on: Rapid Application Development (1991), Dynamic Systems Development Method (1994), Scrum (1986), Extreme Programming (1996), Feature Driven Development (1997)

Agile Principles

- Iterate, Increment and Evolve
- Efficient face to face communication
  - Centralized, easily viewed, board of tasks and progress
- Short feedback/adaptation cycle
- Quality Focus
  - Continuous integration
  - Unit testing/test driven development*
  - Pair programming*
- Adaptive, not predictive*

Scrum vs. Agile

- Agile can be done without scrum.
  - https://medium.com/@sudharshantony/top-5-agile-and-non-agile-methodologies-767a1ab40510
    - Extreme Programming Loop
    - Rapid Application Development model
    - Kanban
    - Lean Software Development (LSD???)
- Scrum can be done without being agile.

Abuses of Scrum: DARK SCRUM

- Power holders decide development plans (what order, how to implement, etc.) instead of explaining what needs to be accomplished and then letting the team make the plans/decisions.
  - Adding a standup meeting to existing development processes is not scrum, it must lead to team self-development
  - Power holders usually don't know what is involved in the development, nor the current state of existing structures
  - Developers burn out from being forced to work under the same requirements but at a faster pace.

https://ronjeffries.com/articles/016-09ff/defense/

Abuses of Scrum: DARK SCRUM (2)

- Using Sprint Review to remind team of what was not accomplished.
  - Team should deal with this in Retrospective without product owner in order to adjust their own priorities.
- Power holders participating in retrospective and use it as a time of pronouncing doom.
  - Consequences if results don't improve are shared.
  - This tends to squash creativity and ideas that would lead to improvement as the developers bow to the power holder's demands

https://ronjeffries.com/articles/016-09ff/defense/

Abuses of Scrum: DARK SCRUM (3)

- Power holder holding developers responsible for unclear requirements.
  - The product owner should be the one getting proper requirements from the user, if they are not, it is their failure.
  - Even if they bring developers to help gather requirements, it is still the PO responsibility.
- Antidote to Abuses of Scrum:
  - Dispel Fear of Failure
    - Make progress crystal clear
    - Make existing state of system (e.g. need for refactoring) clear BEFORE Sprint
  - Power of QUALITY Incremental Change

https://ronjeffries.com/articles/016-09ff/defense/

Why I resisted - Initial fears

- Project Management Courses in College: Poor quality comes from increased velocity and lack of a unified architecture
- Fear of micromanagement and higher stress
- Imposter Syndrome
- Past experiences of poor collaboration
- Past jobs in HIGH RISK environments (NORAD)
- Confusing terminology and JIRA
- MORE MEETINGS

Experience Transitioning

- WebSvcs – Initial attempts and failures
  - Tried to ease into it while also undergoing MSU IT reorg.
- Library Reorganization → WebUX
  - the force known as Robin
- Initial failures
  - Ticket scope
  - Ticket creep
  - Ticket Acceptance Criteria (AC) definition
  - Time management

Experience Transitioning (2)

- Eventual successes
  - Writing AC is now second nature.
  - When new tasks are needed, add a ticket for the next sprint (even if it is one point).
  - Split tickets before the sprint starts and don't do all of them.
  - Add more tickets if time allows.
- Current challenges
  - Ticket documentation
    - Ticket Review (what do we do with the closed tickets)

Benefits of Agile/Scrum

- Quick response to emerging issues.
- Team review of objectives, strategies and outcomes.
- Easy access to consultation without fear.
- All members of team are exposed to and learn basics of different disciplines.

Detriments of Agile/Scrum

- Tendency to focus on immediate needs while ignoring long term viability (code rot, tech debt, etc.).
- Easy to slip into a mindset of speed over quality.
- Generalists can be favored over specialists…but sometimes you need specialists.
- Dunning-Kruger effect - people with limited competence in a particular domain overestimate their abilities.
- Frustration in working with outside resources on a different schedule.

Keys to success.

- Jump in with both feet and move slowly.
  - Be committed to the change but don't expect to get it right the first time (nor the 100th time).
- Be agile with your implementation of agile/scrum
  - Start with your best understanding and change what doesn't work.
- Expect a time of lost productivity before seeing the benefits of later productivity.
- Don't expect a miracle solution.

Questions?

You can also email me at finkenb2@msu.edu