An Introduction to Agile/Scrum

Jeff Pulcini
Cyber Security Workshop
October 3, 2020
Today’s Goals

- Give you the big picture, concepts, and keywords of Agile/Scrum
- Contrast Agile with traditional waterfall methods
- Answer why we should care?
Ground Rules

I’m not going to be rigorous in history or definitions.

I’m likely to make extreme statements to make the point.

We will talk software development, but Agile can be used for anything.

INTERACTIVE
CREDITS

Base presentation was done by:
Mike Cohn (circa 2007)
mike@mountaingoatsoftware.com
www.mountaingoatsoftware.com
(720) 890-6110 (office)

Other portions are credited as appropriate.

Additional material by Jeff Pulcini
Problem Statement

Today, we need to...

- Produce products faster and for less
- Maximize the actual and perceived value we deliver
- Keep up with the pace of change in the market

Adapted from a presentation by Evan Campbell, Slide 32
http://agilecommons.org/files/4e9f355ad9/Agile_Fncl_Impact_Slides.pdf
So, what’s broken?

Answer:

Our philosophy
Waterfall Project Management Methodology

- A step-wise approach to product delivery

- It is a relay race of analysis, requirements definition, design, implementation (code and test) and then delivery and maintenance.

- Product delivery is a “big bang”/“all or nothing”
Downside of This Approach

- **Focus** is primarily on process, not people.
- **Formal communications**, Lots of words on paper.
- Rigid in execution. **Unresponsive** to changing markets (and thus users needs).
- Each step fosters a **them vs. us** mentality.
- **Plan based** - We measure if we are on track, not what we have done.
- **Long development cycles**
- **You know the least when you start**
Cone of Uncertainty

As described by Steve McConnel, *Software Estimation: Demystifying the Black Art*

Example: Boeing’s 787 Dreamliner was 5x more expensive and took twice as long to deliver than original estimates.
Downside of This Approach

- **Focus** is primarily on process, not people
- **Formal communications**, Lots of words on paper
- Rigid in execution. **Unresponsive** to changing markets (and thus users needs)
- Each step fosters a **them vs. us** mentality
- **Plan based** - We measure if we are on track, not what we have done
- **Long development cycles**
- **You know the least when you start**
Waterfall vs. Agile

THE WATERFALL PROCESS

‘This project has got so big, I’m not sure I’ll be able to deliver it!’

THE AGILE PROCESS

‘It’s so much better delivering this project in bite-sized sections’

From various sources from the internet
Process Comparison

• At a high level, both processes will deliver a product.

• The difference is in weight, execution, philosophy, and value delivered.

• Waterfall is Command and Control - Agile is about communications and personal responsibility.

• Agile and Scrum are Empirical Project Management.
  • Based on Statistical Process Control (Shewhart and Deming) and Lean Manufacturing.
  • Frequent Inspect and Adapt Cycles.
A Quick Poll
I like to work...

1. By myself with specific, written requirements and/or process and not bothered with other people
2. By myself, but with general requirements and with access to others to discuss requirements, and issues
3. In collaboration a partner with access to a team associated with the project
4. In a large room with many people so I can overhear conversations
When I work, I like to:

1. Be told exactly what to do with little latitude for interpretation.
2. Be given a goal specific enough that I know where I am going, but allowing me flexibility in achieving that goal.
I have worked on the following types of projects:

1. Small personal projects
2. Medium to large personal projects
3. Small projects in a business
4. Medium projects in a business
5. Large projects in a business
Agile

A Philosophy to Change Our Mindset
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 Process 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 on the items on the right, we value the items on the left more.”

Source: www.agilemanifesto.org
Leadership - Mission First, People Always?

In 2009, James Zenger published a fascinating survey of 60,000 employees [...] Two of the characteristics that Zenger examined were results focus and social skills. [...] if a leader was seen as being very strong on results focus, the chance of that leader being seen as a great leader was only 14%. [...] If a leader was strong on social skills, he or she was seen as a great leader even less of the time — a paltry 12%

However, for leaders who were strong in both results focus and in social skills, the likelihood of being seen as a great leader skyrocketed to 72%.

Should Leaders Focus on Results, or on People?, Dr. Matt Lieberman, Harvard Business Review, December 27, 2013
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 timescale.

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

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.**

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.

Source: http://agilemanifesto.org/principles.html
Sequential vs. overlapping development

Rather than doing all of one thing at a one time...

...Scrum teams do a little of everything all the time

A Framework - Scrum
Characteristics

• Requirements are captured as items in a list of “product backlog”

• Product progresses in a series of short “sprints”

• Self-organizing teams

• No specific engineering practices prescribed

• Uses generative rules to create an agile environment for delivering projects
Generative Rules

• Think of the word “general”. Dee Hock, former CEO of Visa said. “Simple, clear purpose and principles give rise to complex and intelligent behavior. Complex rules and regulations give rise to simple stupid behavior.”

• Jim Donehey, former CIO of Capital One used four rules to help ensure everyone was working toward the same shared goals:
  • Always align IT with the activities of the business
  • Use good economic judgment
  • Be flexible
  • Have empathy for the other in the organization
Take a Break
Scrum Details
Scrum Framework

Team Roles
• Product Owner
• Scrum Master
• Team

Ceremonies
• Sprint planning
• Sprint review
• Sprint retrospective
• Daily scrum meeting

Artifacts
• Product backlog
• Sprint backlog
• Burndown charts

Some now call them EVENTS!
Scrum Framework

Team Roles
- Product Owner
- Scrum Master
- Team

Ceremonies
- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts
- Product backlog
- Sprint backlog
- Burndown charts
Product Owner

- Define the features of the product
- Decide on release date and content
- Be responsible for the profitability of the product (ROI)
- Prioritize features according to market value
- Adjust features and priority every iteration, as needed
- Accept or reject work results
The Scrum Master

- Represents management to the project
- Responsible for enacting Scrum values and practices
- Removes impediments
- Ensure that the team is fully functional and productive
- Enable close cooperation across all roles and functions
- Shield the team from external interferences
- The ScrumMaster differs from a Project Manager in that he does not exercise command and control
The Team

• Typically 5-9 people

• Cross-functional:
  • Programmers, testers, user experience designers, etc.
  • Members should be full-time (no multi-tasking!)
  • May be exceptions (e.g., database administrator)

• Teams are self-organizing
  • Ideally, no titles but rarely a possibility

• Membership should change only between sprints
The Myth of Multitasking

from a 1990’s Harvard Study by Steven C.Wheelwright and Kim B.Clark
Testing Multitasking

1. Take paper and pen and prepare to write down the following in three columns HORIZONTALLY.

   A  a  1  THEN
   B  b  2
   C  c  3
   -------continue until Z, z and 26--------
   Z  z  26

2. Everybody turns on stop watch using their smart phone and begin

3. When done, record your time.

4. Reset stop watch and repeat the same exercise, BUT this time go VERTICALLY.
   Complete capital A to Z first, then go for a-z, and then 1-26.

5. When done, record your time.

   Was There a Difference?

From Jeff Sutherland’s book *Scrum: The Art of Doing Twice the Work in Half the Time*
Scrum Framework

**Team Roles**
- Product Owner
- Scrum Master
- Team

**Ceremonies**
- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

**Artifacts**
- Product backlog
- Sprint backlog
- Burndown charts
Sprint Planning

Sprint planning meeting

Sprint prioritization
- Analyze and evaluate product backlog
- Select sprint goal

Sprint planning
- Decide how to achieve sprint goal (design)
- Create sprint backlog (tasks) from product backlog items (user stories / features)
- Estimate sprint backlog in hours

Team capacity
Product backlog
Business conditions
Current product
Technology

Sprint goal
Sprint backlog
The Sprint Goal

- A short statement of what the work will be focused on during the sprint

**Database Application**

- Make the application run on SQL Server in addition to Oracle.

**Life Sciences**

- Support features necessary for population genetics studies.

**Financial services**

- Support more technical indicators than company ABC with real-time, streaming data.
User Stories

Feature #1 - Publish Trainings

As a manager, I want to publish trainings in order to make them available to customers.
What is a User Story

A user story describes functionality that will be valuable to either a user or purchaser of a system or software*

*Mike Cohn, User Stories Applied
Requirements Collection / User Stories

- “Big Stories” to Small Stories

Release 1
- Limit Search to one field
- Search by Keyword
- Limit Search to fields
- Move Emails
- Create and send basic email
- Open basic email
- View list of appts
- Create basic appt
- Update contents / location
- View Appt
- Create basic contact
- Update contact
- Delete Contact

Release 2
- Send HTML e-mail
- Open HTML e-mail
- Empty Deleted Items
- View Daily Format
- Create HTML appt
- Propose new time
- Add address data
- Update Address Info
- Delete Contact

Release 3
- Search attachments
- Search sub folders
- Get address from contacts
- View Weekly Formats
- Get address from contacts
- View Attachments
- Import Contacts
- Export Contacts
- Search Calendar
- Add Attachments
Sprint Planning

- Team selects User Stories from the product backlog they can **commit to completing**
- Sprint backlog is created
  - Tasks are identified and each is estimated (1-16 hours)
  - Done Collaboratively, not alone by the Scrum Master/Product Owner
- High-level design is considered

User Story

As a vacation planner, I want to see photos of the hotels.

(The who, what and why of what is to be done)

Code the middle tier (8 hours)
Code the user interface (4)
Write test fixtures (4)
Code the foo class (6)
Update performance tests (4)
Sprints

• Scrum projects make progress in a series of “sprints”

• Typical duration is 2–4 weeks or a calendar month at most

• A constant duration leads to a better rhythm

• Product is designed, coded, and tested during the sprint
No changes during a sprint

- Plan sprint durations around how long you can commit to keeping change out of the sprint
The Daily Scrum

- Parameters
  - Daily
  - 15-minutes
  - Stand-up
- Not for problem solving
  - Whole world is invited
  - Only team members, ScrumMaster, product owner, can talk
- Helps avoid other unnecessary meetings
Everyone Answers Three Questions

1. What did you do yesterday?
2. What will you do today?
3. Is anything in your way?

- These are *not* status for the ScrumMaster
- They are commitments in front of peers
- Transparency is key!
The Big Picture

- Product Backlog
- Sprint Backlog
- Daily Scrum Meeting
- Sprint
- 24 Hours
- 2-4 Weeks
- Potentially Shippable Product Increment

Image available at www.mountaingoatsoftware.com/scrum
The Sprint Review

- Invite the world
- Whole team participates
- Informal
  - 2-hour prep time rule
  - No slides
- Team presents what it accomplished during the sprint
- Typically takes the form of a demo of new features or underlying architecture
Sprint Retrospective

• Periodically take a look at what is and is not working - Inspect and adapt!
• Typically 15–30 minutes
• Done after every sprint
• Whole team participates
  • ScrumMaster, Product owner, Team
  • Possibly customers and others
Start / Stop / Continue

- Whole team gathers and discusses what they’d like to:

- Start doing
- Stop doing
- Continue doing

This is just one of many ways to do a sprint retrospective.
Scrum Framework

Team Roles
- Product Owner
- Scrum Master
- Team

Ceremonies
- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts
- Product backlog
- Sprint backlog
- Burndown charts
**Product Backlog**

- The requirements
- A list of all desired work on the project
- Ideally expressed such that each item has value to the users or customers of the product
- Prioritized by the product owner
- Reprioritized at the start of each sprint
# A Sample Product Backlog

<table>
<thead>
<tr>
<th>Backlog item</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allow a guest to make a reservation</td>
<td>3</td>
</tr>
<tr>
<td>As a guest, I want to cancel a reservation.</td>
<td>5</td>
</tr>
<tr>
<td>As a guest, I want to change the dates of a reservation.</td>
<td>3</td>
</tr>
<tr>
<td>As a hotel employee, I can run RevPAR reports (revenue-per-available-room)</td>
<td>8</td>
</tr>
<tr>
<td>Improve exception handling</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>34</td>
</tr>
</tbody>
</table>
Sprint Backlog

• One or more Product Backlog items make up the Sprint Backlog.

• The work is decomposed into tasks and hours.

• If work is unclear, define a sprint backlog item with a larger amount of time and break it down later.

• Individuals sign up for work - work is never assigned.

• Estimated work remaining is updated daily as more becomes known.

• Any team member can add, delete or change the sprint backlog.
### A Sprint Backlog

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Mon</th>
<th>Tues</th>
<th>Wed</th>
<th>Thur</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code the user interface</td>
<td>8</td>
<td>4</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code the middle tier</td>
<td>16</td>
<td>12</td>
<td>10</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Test the middle tier</td>
<td>8</td>
<td>16</td>
<td>16</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>Write online help</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Write the foo class</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Add error logging</td>
<td></td>
<td></td>
<td>8</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>
## Burndown Example

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Mon</th>
<th>Tues</th>
<th>Wed</th>
<th>Thur</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code the user interface</td>
<td>8</td>
<td>4</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code the middle tier</td>
<td>16</td>
<td>12</td>
<td>10</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Test the middle tier</td>
<td>8</td>
<td>16</td>
<td>16</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>Write online help</td>
<td>12</td>
<td></td>
<td>44</td>
<td>32</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>18</td>
<td>18</td>
<td>8</td>
</tr>
</tbody>
</table>

![Ideal line chart with burndown example data]
Review / Take Aways
Review / Take Aways

- Agile is a *philosophy* - a way of BEING Agile
- Scrum is a framework, method or process - a way of DOING Agile
- We defer to BEING Agile when we implement a framework, process or method, process
Review / Take Aways

• A leader is strong in both results focus and in social skills - Mission First, People Always

• Leaders like Generative Rules

• Remember to define ready and done

• Leader/friends don’t let friends “multi-task”
Review / Take Aways

• Process is not bad, it just must be subservient to Agile Principles

• Scrum teams do a little of everything all the time

• We deliver functionality, not products - Bite size chunks vs. a Big Bang
Review / Take Aways

- The Product Owner is the ultimate authority in form, function and delivery

- The Scrum Master is assures we are BEING Agile and helps the team move forward

- The team is responsible and self organizing - we all may have roles, but in the end, we are a team of leaders
Review / Take Aways

• User Stories define functionality and promote discussion

• It is utter arrogance to try to predict the future - Cone of Uncertainty

• We first T-Shirt size and only define tasks and time at Sprint Planning
Review / Take Aways

• No Heroics! Sprints provide a sustainable cadence for EVERYONE.

• Transparency is paramount (Remember George Bernard Shaw)

• Inspect and Adapt!
Jeff Pulcini
Jeff.Pulcini@me.com
248-462-9794

2020 Pittsburgh PMI - PittAgile Gathering

PittAgileDay.com

Sign Up at:
https://pittsburghpmi.org/calendar?eventId=5939
Why do this?
Reasons for Adopting Agile Success Rates

![Diagram showing success rates for Agile vs Waterfall methods.](https://vitalitychicago.com/blog/agile-projects-are-more-successful-traditional-projects/)

https://vitalitychicago.com/blog/agile-projects-are-more-successful-traditional-projects/
Bottom Line

- Agile projects are 2X more likely to succeed
- Agile projects are 1/3 less likely to fail than waterfall projects

The Standish Group has conducted surveys of IT project success and failure rates every 2 years since 1994.

Source: Standish Group Chaos Study 2018
Reasons for Adopting Agile
Shifts in Industry Attitudes

Changes from 2018 to 2019

• Less about increasing productivity (51% compared to 55%)
• More about improving team morale (34% compared to 28%)
• Less about reducing project risk (28% compared to 37%)
• More about reducing project costs (41% compared to 24%)

Benefits of Adopting Agile

<table>
<thead>
<tr>
<th>Benefit</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ability to manage changing priorities</td>
<td>69%</td>
</tr>
<tr>
<td>Project visibility</td>
<td>65%</td>
</tr>
<tr>
<td>Business/IT alignment</td>
<td>64%</td>
</tr>
<tr>
<td>Team morale</td>
<td>64%</td>
</tr>
<tr>
<td>Delivery speed/time to market</td>
<td>63%</td>
</tr>
<tr>
<td>Increased team productivity</td>
<td>61%</td>
</tr>
<tr>
<td>Project predictability</td>
<td>52%</td>
</tr>
<tr>
<td>Project risk reduction</td>
<td>50%</td>
</tr>
<tr>
<td>Software quality</td>
<td>47%</td>
</tr>
<tr>
<td>Engineering discipline</td>
<td>42%</td>
</tr>
<tr>
<td>Managing distributed teams</td>
<td>39%</td>
</tr>
<tr>
<td>Software maintainability</td>
<td>34%</td>
</tr>
<tr>
<td>Project cost reduction</td>
<td>28%</td>
</tr>
</tbody>
</table>

Scrum has been used for:

- Commercial software
- In-house development
- Contract development
- Fixed-price projects
- Financial applications
- ISO 9001-certified applications
- Embedded systems
- 24x7 systems with 99.999% uptime requirements
- The Joint Strike Fighter
- HR improvement projects
- Sales and Marketing projects
- Training and Education
- Video game development
- FDA-approved, life-critical systems
- Satellite-control software
- Websites
- Mobile phones
- Network switching applications
- ISV applications
- Some of the largest applications in use
Scrum Has Been Used By

- Microsoft
- Yahoo
- Google
- Electronic Arts
- High Moon Studios
- Lockheed Martin
- Philips
- Siemens
- Capital One
- BBC
- Intuit
- Nielsen Media
- First American Real Estate
- BMC Software
- Ipswitch
- John Deere
- Nokia
- Lexis Nexis
- Sabre
- Salesforce.com
- Time Warner
- Turner Broadcasting
- Oce
Agile Principles in Action

The Troubled HH60W Program

“The team had a moment whether we decided we’re all in, and we’re going to do whatever it takes,” Roper tells Aviation Week. “We’re going to follow the rules, but we’re going to slim them down to the minimum set necessary to keep us focused on delivering on time.”

Will Roper, Air Force acquisition executive

The “old school” methods shaved 4 months from the deployment test program.
Bonus Slides
Stories emphasize the user’s goals not the system’s attributes.

What are we building?
1. The product shall have a gas engine.
2. The product shall have four wheels.
   2.1. The product shall have a rubber tire mounted to each wheel.
3. The product shall have a steering wheel.
4. The product shall have a steel body.

Source: Adapted from The Inmates are Running the Asylum by Alan Cooper (1999).
Example Task Board

<table>
<thead>
<tr>
<th>Story</th>
<th>To Do</th>
<th>In Process</th>
<th>To Verify</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>As a user, I... 8 points</td>
<td>Code the... 9</td>
<td>Code the... 4</td>
<td>Test the... 8</td>
<td>Code the...</td>
</tr>
<tr>
<td></td>
<td>Code the... 2</td>
<td>Test the... SC 8</td>
<td>Test the... SC 8</td>
<td>Test the...</td>
</tr>
<tr>
<td></td>
<td>Test the... 8</td>
<td>Test the... SC 4</td>
<td>Test the... SC 6</td>
<td>Test the...</td>
</tr>
</tbody>
</table>

As a user, I... 5 points

<table>
<thead>
<tr>
<th>Story</th>
<th>To Do</th>
<th>In Process</th>
<th>To Verify</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Code the... 8</td>
<td>Code the... 8</td>
<td>Test the... 8</td>
<td>Code the...</td>
</tr>
<tr>
<td></td>
<td>Code the... 4</td>
<td>Test the... SC 8</td>
<td>Test the... SC 8</td>
<td>Test the...</td>
</tr>
<tr>
<td></td>
<td>Test the... SC 6</td>
<td>Test the... SC 6</td>
<td>Test the... SC 6</td>
<td>Test the...</td>
</tr>
</tbody>
</table>

Reproduced from [http://www.mountaingoatsoftware.com/agile/scrum/task-boards](http://www.mountaingoatsoftware.com/agile/scrum/task-boards)
Why the Neanderthals became extinct