



Quality, Performance and Success in 1-2 person teams

A story of Small Scale Scrum

Dr. Leigh Griffin
Engineering Manager

Brendan O'Farrell
Engineering Manager

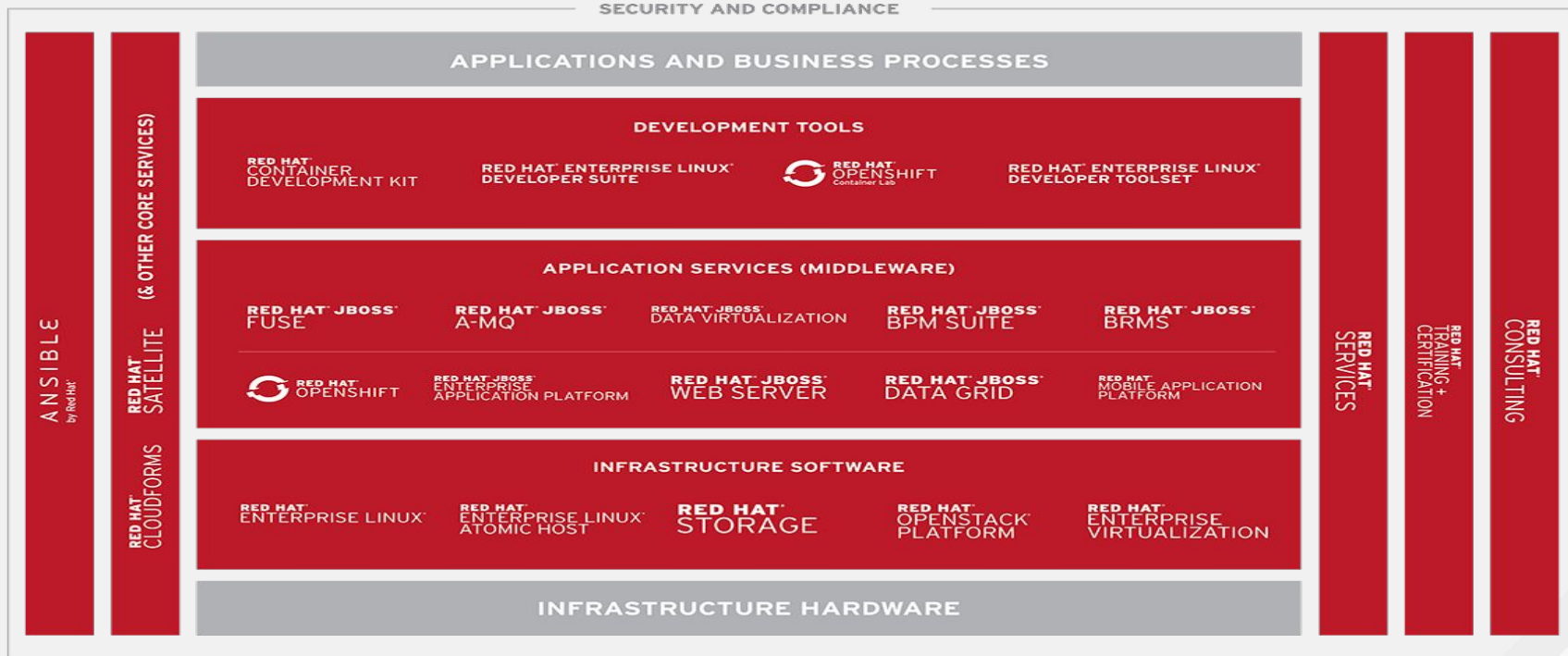
Agile Cambridge
28-SEP-2017

Leigh Griffin

Engineering Manager

- 10 years of industry experience
- PhD Graduate through TSSG / WIT 2012
- Worked in several local companies as Senior Engineer, Scrum Master, Development Lead and Manager
- Joined Red Hat in December 2015 tasked with an Agile Transformation
- Promoted to Engineering Manager since June 2016
- Responsible for a distributed team of 17 Engineers as part of Red Hat Mobile
 - Co-manager for 34 brave souls

Red Hat: Aren't you guys the Linux company?



Red Hat Mobile: A Large(ish) Scrum(ish) Team

- Red Hat Mobile Application Platform (RHMAP) has ~50 team members
 - ~30 Engineers
 - ~10 DevOps
 - ~6 QE
 - ~4 Others {Documentation, Security, Internationalisation}
- Cross functional teams with cross skilled individuals
- Typically 5-7 Agile teams active
 - Most following a modified Scrum principles
 - Some following Kanban
- Large Scale Scrum principles applied for coordination and planning
- A successful Agile Conversion team ~24 months into their journey

Red Hat Motivation

- Red Hat are heavily engaged in three areas where 1-2 person teams are the norm
- Professional Services or Consultancy projects typically charge per person day
 - Industry norm ; tends to be small as a result of costing it like this
 - Many hats worn by a single person
 - Projects running for typically 20 days or less is the most popular
- Student completing a Final Year Project (FYP)
 - Capstone project, of around 24 weeks duration
 - Individual sport for assessment reasons
 - Red Hat mentoring of Student Projects is a community outreach initiative
- Paid for Commercial Research Projects
 - Specific problem, 6-24 month duration, 2 research engineers

Combined Background Motivation

- Brendan and Leigh came from a Research Background
 - Small Teams are the norm here as well
- Agnieszka working towards her MSc in conjunction with Red Hat
 - Huge focus on quality in her work and role as a Consultant
- College Research Projects
 - MSc or PhD students are running a long lived single person project
 - 5 years for my own PhD which is average for IT
 - 2 years for an MSc
- Changing Research Model of funding
 - Innovation Vouchers, 2 weeks of work, 1 person project, state sponsored
 - Innovation Collaborations, 1-12 months, 1-2 person project, state & company funded

Consistency and Quality

- RHMAP Team wanted a consistent way of communicating
 - Our day to day work demanded it
 - We would parachute into Consultancy projects to provide assistance
 - Frustrating to have differences in expectancy with respect to Quality & Success
- RHMAP Team working closely with our local college
 - Work placement students
 - FYP sponsoring
 - Our main hub for recruitment, 68% of our team were Alumni
 - A smoother onboarding for potential new staff
- The concept of Small Scale Scrum started to form
 - Had to mirror how we worked as much as possible
 - Had to have a huge emphasis on quality

Small Scale Scrum

Small Scale Scrum: A Definition

Small Scale Scrum is a people first framework defined by and for small (1-2 person) teams which supports planning, developing and delivering production quality software solutions

Agnieszka Gancarczyk

What Small Scale Scrum means to us

- Communication and Collaboration
 - Tightly defining expectations
 - Defining collaboration boundaries with customers or external entities (e.g. college lecturers)
- Engagement and Motivation
 - Proactively involving the customer in our day to day life
 - Our research showed lone developers suffer at times
 - Motivation to innovate and communicate with peers is low
 - Feeling of being overwhelmed, too much work, too little time, underappreciated
 - Swimming in the sea alone ← turnover becomes a problem
- Performance
 - Internal efficiency and ultimately happier customers
- Quality
 - Of the utmost importance to our beliefs and what we represent as a company

Small Scale Scrum in Action

- Use cases we successfully applied this to:
 - Single person 4th year (Student) BSc Hons project, sponsored by Red Hat.
 - x6 students, remotely working with us
 - DHS 18 month project researching state of the art security
 - Professional Services through our Consultancy Team
 - 1-2 people, ~20 days average work, ~5 projects completed using our framework
 - State funded enterprise voucher, allows a company to explore a business opportunity or problem. ~50 voucher projects completed

The background of the slide is a photograph of a large cable-stayed bridge, likely the Akashi Kaikyo Bridge, spanning a body of water. The bridge's massive concrete piers and the network of stay cables are visible. The right half of the image is covered by a solid red overlay, which serves as a backdrop for the title text.

The Student Projects

Brendan O'Farrell

Engineering Manager

- 25 years of process management and improvements
- Transitioned late in life into an IT role as a Quality Engineer, Project Manager & Software Manager.
- Joined Red Hat in February 2017 as Engineering Manager
- Responsible for a distributed team of 8 Engineers in Red Hat Mobile
 - Co-manager for 34 brave souls

The student project

- Single person 4th year (Student) BSc Hons project, sponsored by Red Hat. Duration two semesters, nine months
- 2 person team project, Sponsored by Red Hat. Duration 1 semester, 12 weeks.
- Sponsored student given access to tools and domain knowledge
- Process established: Two week sprints, 3 of the 4 Scrum ceremonies adhered to
 - Scrum by osmosis as we only had a short time each sprint with them
- Definition of Done (DoD) established
 - High emphasis on quality (for us) and documentation (for the college)
- Definition of Ready (DoR) agreed upon
 - Ensured clarity and expectations set all around

Starting Point: Setting the expectations

- Student projects driven by a loose requirement spec
 - Primarily from a Red Hat staff member
 - Driven by the students desire to skill into an area for future employment
 - Driven by the colleges desire to skill into an area for future teaching
- Spec deliberately over-scoped
 - We encourage hard decisions because they mirror real world projects
 - Allows for interaction and customer conversations
- Lightweight Product Requirement Document (PRD) established with the student
 - This becomes the customer contract
 - Gave them a guiding voice and finish line
 - Allowed them put an emphasis on Quality for the end output Vs a rough feature

Quality Contracts: DoD

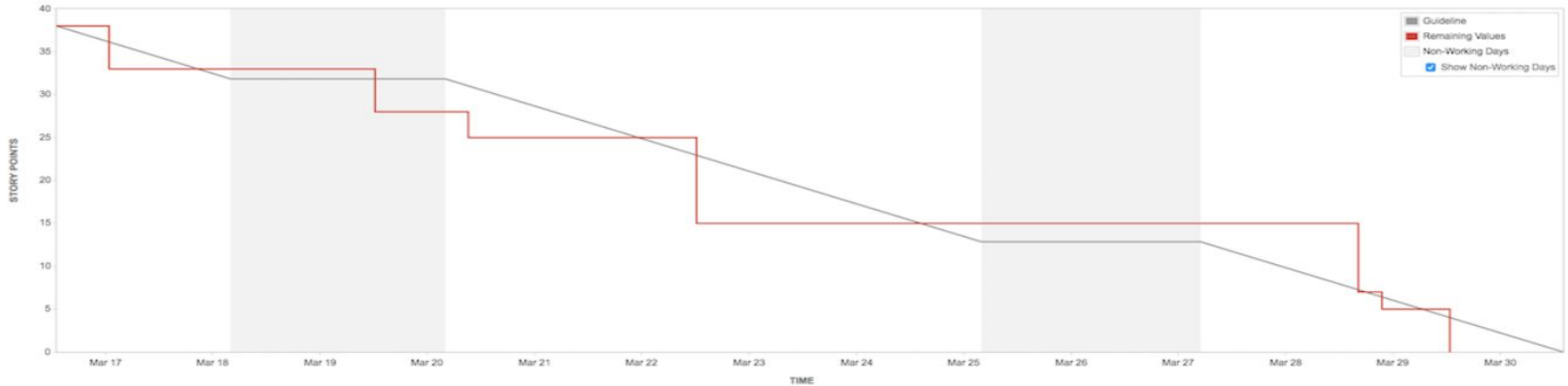
- Definition of Done was hugely important
- Designed to be loose and designed for the student to enforce
 - Acceptable Unit & Integration Test Coverage
 - Acceptance Tests were mandatory
 - CI/CD build systems passing
 - Work documented to an adequate level
- Became a guiding principle for the student
 - Pushed back on priorities and challenged us on what was next
 - Forced them into hard decisions between efficiency, ease of life and features

Quality Contracts: DoR

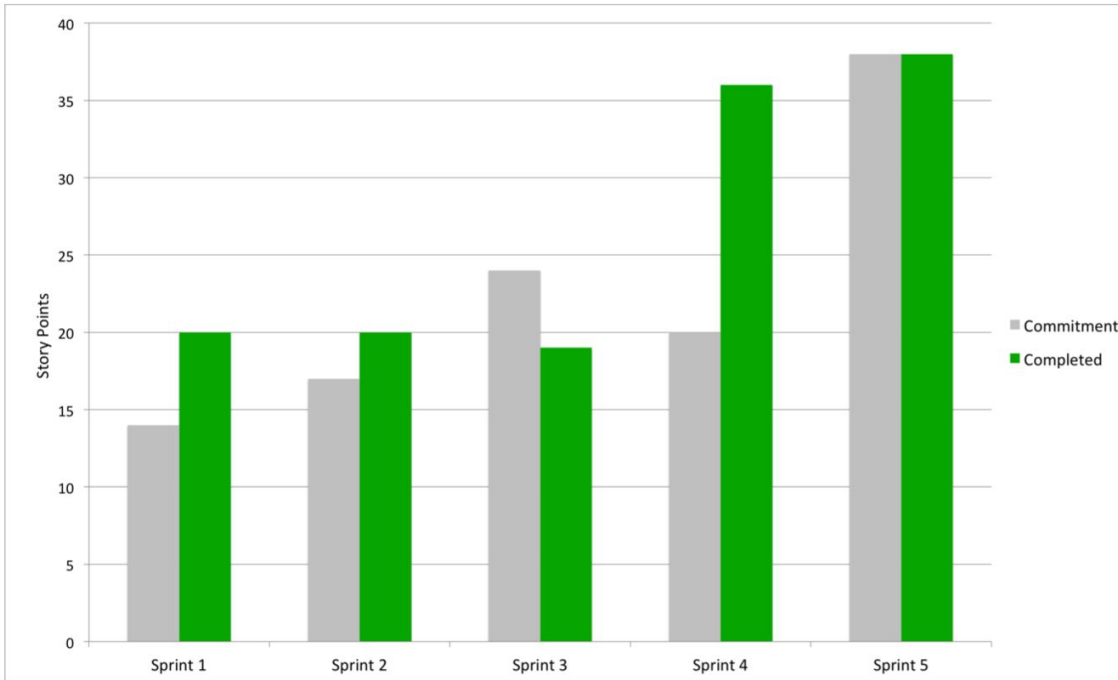
- Definition of Ready helped to control the pace of work
- You are Ready to start on this Task / Feature when:
 - Test strategy is in place
 - Exit strategy is in place and agreed upon
 - User Stories are generated and agreed upon
 - Rich JIRA tickets are in place with sub tasks adhering to the DoD
- If all is in place.....you are ready to bring it into your sprint
- This created a very granular backlog and focused on the high value items
 - Allowed us act as a Product Owner and guide the students work

Sprint Planning

- Encouraged the students to define complexity for their work
 - Gravitated naturally to Story Points
- Allowed the student pace their own work and know their commitment



Sprint Planning



The Focus on Quality: Build early, build often

Builds per day

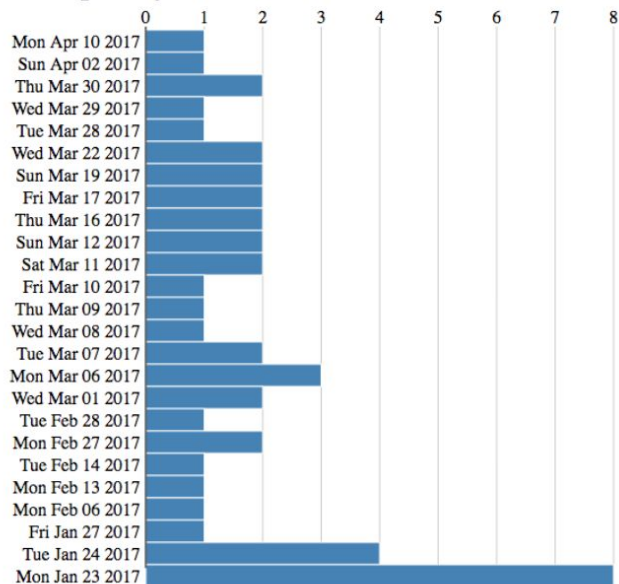


Figure 24: *Project Builds per Day*

Number of Builds	84
Failures	12
Nightly Deploys	42
Successful Pull Requests	18
Releases	12

Self Reflection: The 1 person Retrospective

- Encouraged the student to keep a diary as a daily standup kind of log
- Meeting every 2nd week to run a formal retrospective
- Keeping it engaging is key, drawing an experience from the person
 - The Good, The Bad, The Ugly
 - Starfish Retro
 - 3 Little Pigs (House of Straw, Wood and Stone)
 - The Sailboat
- Maturity emerged within 2 - 3 retrospectives
 - Very self critical with respect to the variety of the roles
 - Helped shape and redefine the DoR and DoD

The Focus on Quality: Code Coverage

In Figure 27 below we can see that the code coverage for this project was consistently maintained at 90%.

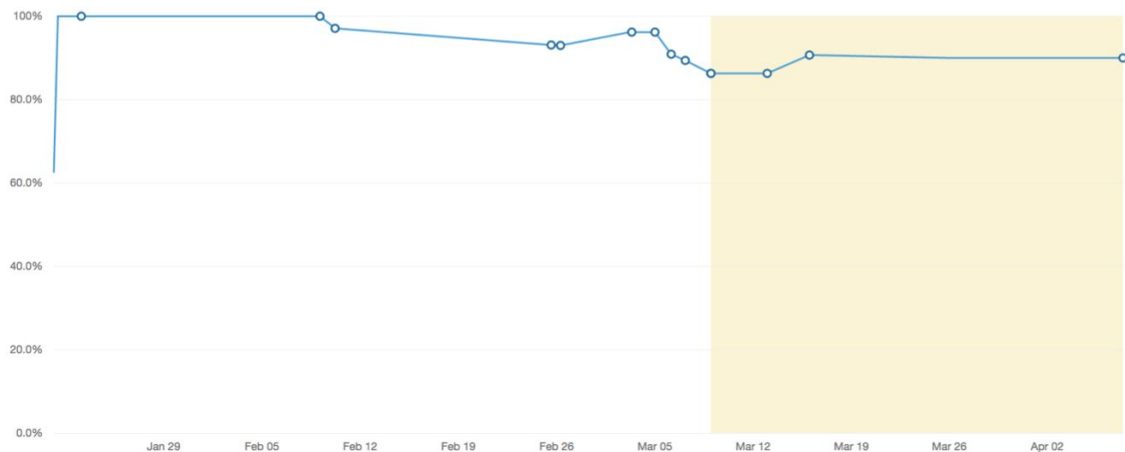
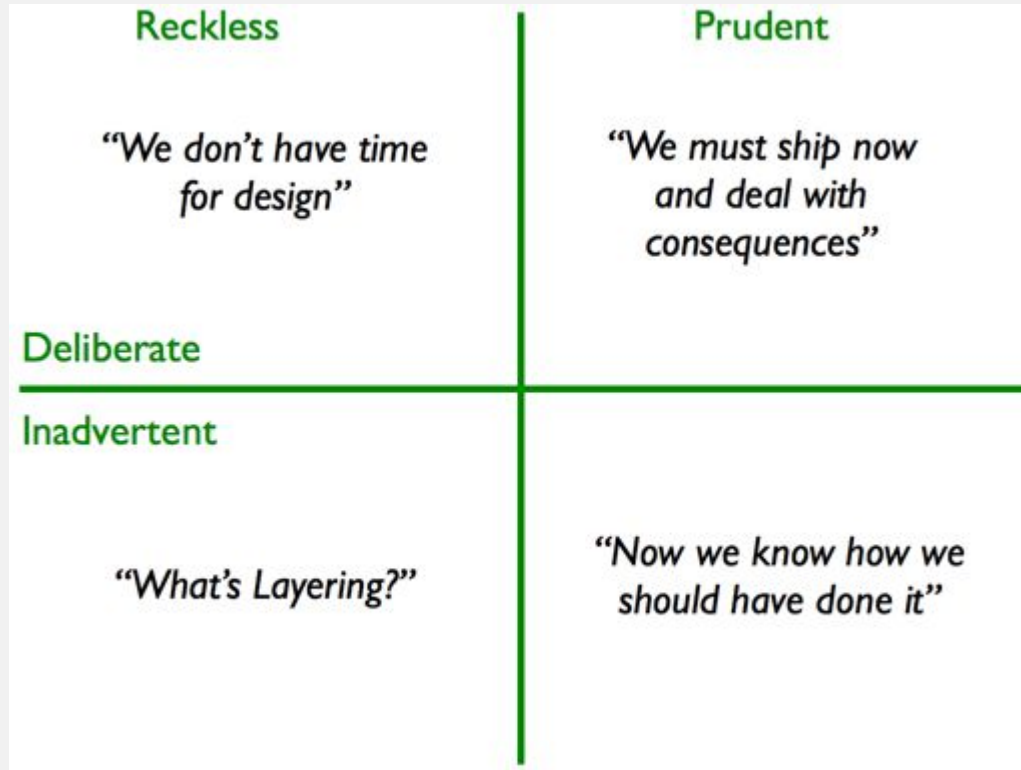


Figure 27: *Code Coverage*

The Focus on Quality: Technical Debt



Project Outcomes: How Successful were we?

- 3x 1st Class Honour results
 - 2x 95+ marks, highest marks in the college in several years
- All projects were completed to minimum spec and 75% average completion on stretch goals
 - Projects reshaped with more goals added throughout
 - Students over achieved with respect to customer expectations
- Projects completed on average 2 weeks ahead of school schedule
 - Huge pressure relief on the students
 - Happy customer!
- 4 students hired directly into Red Hat and walked into our processes with no onboarding
 - One student serving as a ScrumMaster now and moving his career in that direction

Project Outcomes

- An overall retrospective was driven from the successful college projects
- Fed in with knowledge from our Consultancy projects
- Model being adopted by our local college Waterford Institute of Technology
- Running the same approach again this year with 5x sponsored projects
 - Self running projects
 - Little to no overhead on us as staff
 - The handbook and expectations are there

Our principles realised

- Communication and Collaboration
 - Formed a close relationship between the customers (college, Red Hat)
 - Solved the communication problem that exists in modern students
- Engagement and Motivation
 - Engaged with Red Hat staff and our community
 - Blogged, tweeted, emailed and interacted
 - Motivation for a happy customer (and a job!)
- Performance
 - Production grade projects produced
 - Perfect mix of Engineering, Documentation and Quality tasks consumed
- Quality
 - 2 applications integrated into our production stack seamlessly
 - Full UAT carried out and signed off as complete



State Sponsored Mini Projects

Method to our Madness

WHY



- 7 Internal units, 130 researchers, 15-20 Innovation Vouchers every 6 months
- Work on Innovation Vouchers was completed using different tools, word, Google Doc, AsciiDoc, Markdown, Etc
 - This led to a non uniform approach
 - With no apparent structure, they invariably ran over the allotted time
 - No natural flow when the client moved from voucher to partnership



Basecamp



Confluence



TAIGA



asana

Atlassian



Bitbucket



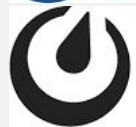
HipChat



JIRA



GitLab



Mattermost



slack



OpenProject

FOUNDATION

PROJECT Hub



Microsoft Office Project

Google docs



REDMINE

flexible project management



teamwork.com

Method to our Madness

- The research centre had moved from the Waterfall model to Agile a number of years ago, but!
 - Sprint Retrospectives some do some don't, action points few and far between
 - Story Points were done on an hoc basis
 - They didn't really do Reviews
 - Stand ups, once the time was divisible by 15, that was ok
 - Some or all unfinished tickets could be removed before the Sprint finishes
 - Sprint goals were not always identified
 - ScrumMaster role was often covered by QE
- This is not unique, quite often we see this scenario when moving from one process to another
 - It is at this stage that we had to reset
 - The small scrum approach was one avenue to consider

State Sponsored Mini Projects

- Single person. Innovation voucher - explore a business opportunity or problem. Duration 10 days.
- 2 person team. Matched voucher - new product/process development. Duration 30 days
- No software development takes place in either option at this stage
- Process established: Two week sprints, Scrum ceremonies adhered to, timing of these can vary
- Definition of Done (DoD) established with client
- Definition of Ready (DoR) agreed upon
 - Ensured clarity and expectations set all around

Initial Steps

- Agree Definition of Done with client
 - We could project our principles but ultimately they had an outcome to achieve
- Designed to be explicit
 - Limited finance
 - Limited scope
 - Limited time
- The DoD becomes the researcher's guide
 - It controls the client's expectations
 - It lays the foundation for any future work

Initial Steps

- Agree Definition of Ready with client
- DoR in the case of state sponsored research requires input from outside the research department
 - Information form completed for state body
 - Invoice generated by research centre
 - Invoice paid
 - Contract signed, good to go
- Only when the above is completed can work begin
 - This mirrors our Consultancy projects
- As no development is done at this stage, no test plans are required, however
 - User Stories are generated and agreed upon with the clients input
 - Rich JIRA tickets are in place with sub tasks adhering to the DoD

Ongoing Steps

Scrum Events

- Sprint planning, Innovation voucher project only has one sprint, the matched voucher has 2 sprints
 - Attended by client, researcher and Unit Manager (acts as Scrum Master)
- Daily standup
 - No standup takes place for innovation voucher project
 - Daily standup takes place on the matched voucher project
- Sprint Review
 - Attended by client, researcher and Unit Manager (acts as Scrum Master)
- Retrospective
 - Attended by client, researcher and Unit Manager (acts as Scrum Master)

A Focus on Quality

- Metrics for non code based projects (all voucher projects)
 - Written in AsciiDoc
 - Output & Extensible
- Title page
 - Abstract
 - Introduction
 - Procedure
 - Results
 - Discussion
 - References
 - Appendix (or Appendices if more than one)
- Creating a structured approach, increased quality and customer satisfaction
 - Reduced the number of quality issues that had previously been identified by the state board

What this approach provided

- Scrum Training
 - Agile revision/reset on an individual level
- Consistency
 - A consistent project approach across all projects and units
 - This was to play into a wider Research Center conversion to Agile
 - That's another presentation for another day :)
- More valuable output for the client
 - Time limited, maximum value with the focus needing to be on success
- More identifiable approach to process
- Excellent foundation when the client moves through the state funded options

Our principles realised

- Communication and Collaboration
 - Strong outputs for the customer
 - Branch points to allow them progress their work further
- Engagement and Motivation
 - Consistency across 60+ vouchers and 20+ researchers
 - Formed the basis for a wider understanding and execution of Agile
- Performance
 - Research group had the highest conversion rate from Vouchers to Collaborations
 - Greater chance to involve new researchers, expanded our base
- Quality
 - All vouchers completed 100% successfully, historically ~10% failure rate
 - All vouchers completed on time, historically ~25% ran over at our own cost

Lessons Learned

Our Retrospective!

22-09-17

SMALL SCALE RETROSPECTIVE

Good

- COMBS,
 - ASYNC
 - CUSTOMER
 - SELF + TEAM
- AGILE SANDBOX
- RAPID TEAM DEVELOPMENT
- FORMAL PROJECT PLANNING
- TOOLING
- DEMO FEEDBACK

BAD

- LACK OF GUIDING VOICE

UGLY

- PROCESS FLOW TOOLS
IE GIT, JIRA
- PRD
- BACKLOG SPIRING
- UPSKILLING

Good Scrum Principles are not Size Dependent

- Well defined business requirements are a must
 - Establishes expectations
 - Derives your strategy
 - Exit criteria and knowing when you are done is crucial
- User Stories help give a potentially disconnected person a sense of ownership
 - Longer running teams feed off of this
 - A short lived project will mean very little to an overloaded person
- Enforcing the ceremonies
 - Daily standups became a diary or a short blog, we are exploring this
 - Helped them think through scenarios and problems
 - Helped put structure on their day
 - Afforded stakeholders a view into the process

Quality above Quantity

- Quality focused items were primary entities in the plan
 - Given equal consideration and helped dictate milestones
 - Came at the “cost” of features
 - Forced honest conversations with customers
 - The Iron Triangle of Software Dev where Resources are iron
- Code freeze was a huge help to the team
 - Defined a boundary between when they needed to change hat
 - Gave appropriate time for UAT with the customer
 - Gave appropriate time for end of lifecycle tasks such as final user documentation



A Potential Training Ground

- Allowed our larger Scrum House run a small dynamic team of 1-2 people successfully
 - Bringing a more research focus to our teams
 - Ensuring that the work done was not hacky or proof of concept
 - Much more trust and empowerment in the team
 - Fear of small teams evaporated
- Fed into a larger Agile conversion mentality
 - In Red Hat it helped reinforce our Agile direction
 - In our old research house it gave a unified view of Agile
- Either way we learned something, and I hope you did as well :)

Student Blogs

[Gantry—A Lifecycle Management Application for Docker](#)

Stephen Coady

[Aerodoc: Monolithic to Microservices](#)

Colum Foskin

[Kujira - An Intern college project with Red Hat](#)

Austin Cunningham

[Scrum for two?](#)

Damien Murphy

- Students published 4x Blogs about their journey
 - Read about it on twitter & Slack we just sent them out on the #agilecam



redhat.

bofarrel@redhat.com
lgriffin@redhat.com



plus.google.com/+RedHat



facebook.com/redhatinc



linkedin.com/company/red-hat



twitter.com/RedHatNews



youtube.com/user/RedHatVideos