Certified ScrumMaster® (CSM)

Rick Waters, CST

Certified ScrumMaster® Workshop

Certified Scrum Trainer®
Rick Waters, CST®

Rick has been practicing Agile software development since the late 1990's. His move into Agile leadership roles was made out the necessity to retain his sanity, in a world of command and control managers. Discovering that he gravitated to processes built on Empiricism, Rick found that Scrum and Kanban resonated with his sensibilities.

Rick has been teaching Agile Principles and Values, and how they are used in Scrum and Kanban, professionally since 2012. When Rick is not training, he is coaching Agile teams and former students. Since 2010, Rick has coached more than 100 Scrum, Kanban and XP teams, participated in over 2000 Sprints, and helped teams move from deploying once per year, to twice per week.

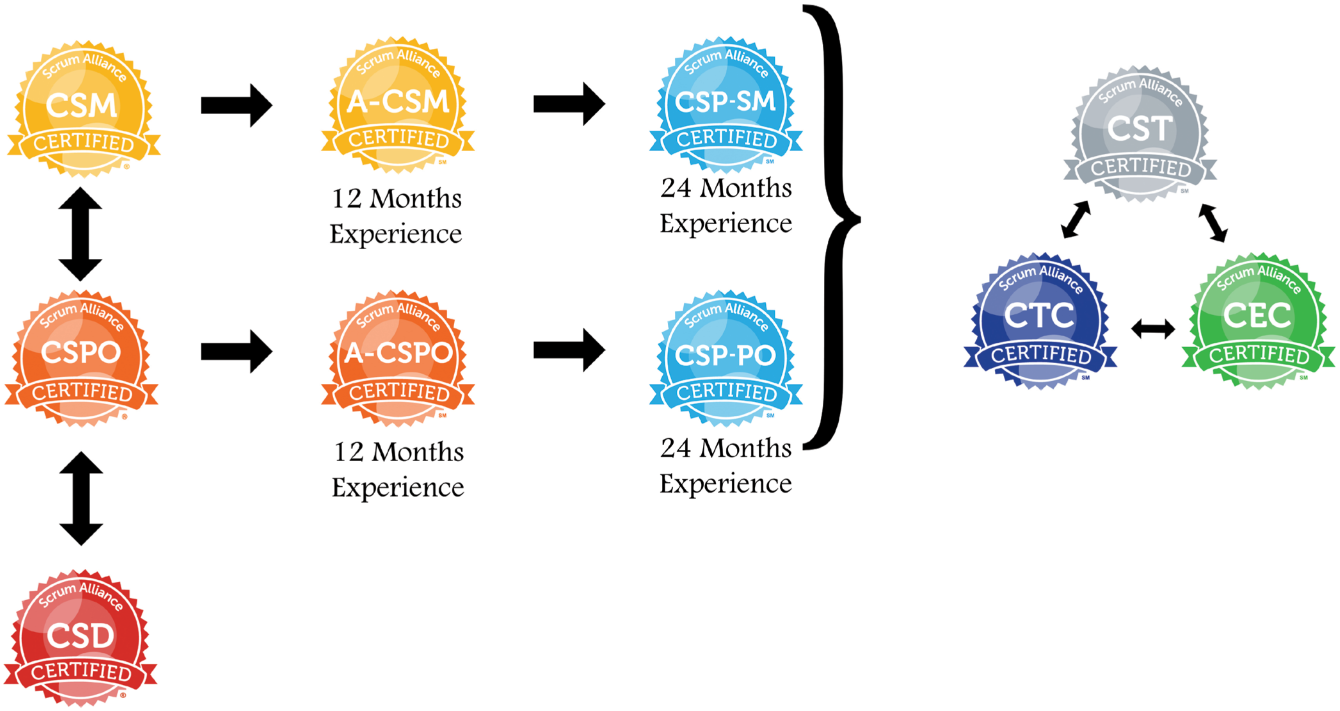
This workshop will focus on single-team Scrum, as defined by the Scrum Guide, authored by the inventors of Scrum—Ken Schwaber & Jeff Sutherland. This class is not focused on scaling Scrum, scaled Agility, or what electronic tools you should use to sub-optimize your Agility efforts. If you have questions pertaining to tools or scaling, please seek your trainer's attention during breaks or after class.

If you have Scrum-related questions, don't wait for a Q&A session, we won't have one. Ask your question when it's still fresh on your mind. It's possible you aren't the only person with the same question.

Rick's Email: wisdom@wisecrum.com
Rick's Twitter Handle: @AgileWaters
Rick's LinkedIn Profile: https://www.linkedin.com/in/agilerick/
Wisecrum Website: http://wisecrum.com/
For Veterans Needing Training: http//Scrum.vet

CSM Workbook Table of Contents

Certification Paths



Where you go from here is completely up to you. Consider your true career goals, and focus your efforts there. If further and continued certification is for you, consider staying on the path that defines your role, before seeking certifications outside your role.

However, since you are a Scrum Master, and one of the three services a Scrum Master provides is to coach Product Owners into understanding their job...you might want to consider getting some training as a Product Owner too.

Be advised, the need for bona-fide Agile Coaches is growing daily, and the path to certification as a Scrum Alliance Certified Coach (CEC & CTC) is your best option in terms of proving your experience and acceptance in the industry.

Class Structure

This class is obviously being delivered virtually, but that is no reason to not participate. Your participation will determine whether you end up fully understanding the materials, or not.

We will be using Mural and Zoom to communicate and collaborate.

The class has a set agenda, but we can stray from that agenda, slightly, if students want to explore a certain topic more deeply. What this means, though, is that we may not be able to focus much on other topics. Your trainer is very experienced in making these types of adjustments, so please don't worry.

Here is how the class usually breaks down…

Class Ground Rules & Introductions
About 20 to 30 mins

Agile Values
About 1 hour

Scrum Deep Dive - A Drawing
About 4 hours

Definition of Done
About 45 mins

Many Hats of a Scrum Master
About 60-90 mins

Scrum Teach Back (optional)
About 1 hour

Burning Questions
About 15 to 30 mins

Agile Principles
About 1 hour

Scrum Values
About 20 mins

Scrum Team Bill of Rights
About 1 hour

Retrospective Facilitation
About 60 mins

Course Wrap-Up
About 10-30 mins

I know the times don't look like they add up to 16 hours of class, but they do. Inevitably, we will have lengthy conversations/discussions about some of these topics that make the times stretch. This is what makes this class so interesting!

Certified ScrumMaster® Workshop

Ground Rules

Participation

Scrum Alliance rules dictate that all students need to participate for at least 15 of the 16 scheduled instruction hours of class in order to graduate and take the certification exam.

Breaks

We will take 10-minute breaks after each hour of class. We will take a 45-minute lunch in the middle of each day. Luch break will begin sometime between 11:45am and 12:15pm. Please do not expect us to wait for you if you are late coming back from break - we will start without you.

Harrassment & Bullying

Rick has a zero-tolerance policy for bullying/harrassment. Please allow everyone in class the opportunity to share their own opinions, experiences and ideas without criticizing them. Rick will swiftly remove anyone from class who violates this policy.

Fun

This class can be EXTREMELY boring if you aren’t willing to speak up and share during discussions. REALLY boring!
Please treat this class less like a lecture, and more like a clinic or workshop.
Rick is going to tell a lot of really bad dad jokes - be prepared.

Burning Questions

Agile Manifesto

At the Snowbird Ski Resort in Utah, in February of 2001, 17 gentlemen from the Software Development, Organizational Design and Management Consulting industries gathered together to discuss the current and foreseen future states of their combined industry. They concluded that what they were engaged in was all intertwined so much, that a collective understanding of the future was called for. They created the...

Manifesto for Agile Software Development

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**.

Exercise
Take 10 minutes and discuss what each of the Values actually mean.

The 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.

The Agile Principles

Draw lines connecting the Agile Principles to their meanings. These are not 1-to-1 matches.

1

2

3

4

5

6

7

8

9

10

11

12

Echoes our connection to Lean

Eliminate Waste

Skill Mastery

Our Purpose

Autonomous Teams

Shows our focus on Customers

Promotes a collaborative and cohesive team

Helps prioritize lists of work

Sustainable Pace

Shows our focus on Customers

If we make our sprints as short as possible, we give ourselves more opportunities to improve

Shows our focus on Customers

Comittment to Quality Work

What does Progress really mean?

Eliminate asynchronous communication

Our team maturity goal

Agile Manifesto

Things to commit to memory

- The exact wording of the 4 Agile Values
- A Scrum Team is implementing Agile Principle #10 when they remove unnecessary work froom the Product Backlog, or Sprint Backlog, because this allows them to prioritize their work easier
- The exact wording of Agile Principle #11

Scrum Overview

What is the iteration called in Scrum? What is it's maximum duration?

Who states the 'initial goal' during Sprint Planning?

What is Sprint Planning Part 1 called? Explain the point of Part 1

What is Sprint Planning Part 2 called? Explain the goal of Part 2

What is Sprint Planning Part 3 called? What might happen if we don't do Part 3?

What are the two questions the Scrum Master needs to ask the Developers during Sprint Planning?

Who is supposed to facilitate the Daily Scrum?

What is the format of the Daily Scrum? What's the maximum duration?

When Developers are done with a task, what do they do with it?

When Developers are done with all of the tasks for a PBI, what do they do next?

What happens if a PBI isn't finished during the planned Sprint?

Scrum Overview

Who compares all of the ‘Done’ work from the Sprint to the DoD?

What happens to the work that doesn’t pass the comparison test with the DoD?

What happens to the work that passes the comparison test with the DoD?

What 3 words describe the Increment?

What are the 3 Scrum Artifacts?

Who is the audience for the Sprint Review? What do we want to get from them? Why?

What is the one mandatory thing we need to do in the Sprint Review? What else can we do?

What’s so important about the Sprint Retrospective?

What’s so important about the Product Backlog Refinement? List some things to do.

What’s the intent of Product Backlog Refinement? What’s a side effect?

Who is the only person on the Scrum Team that can cancel a Sprint? Why?

Scrum Overview

What is the Product Backlog?

What is the Sprint Backlog?

What is the Increment?

What is the Sprint Goal?

Scrum Overview

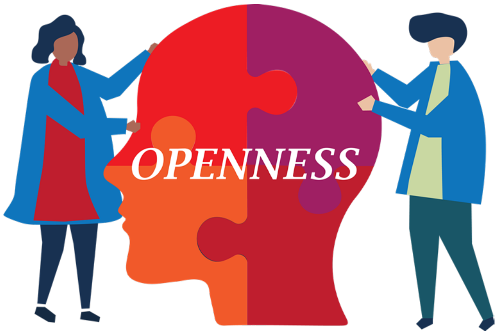
Things to commit to memory

- The maximum duration of the Sprint
- When Sprint Planning starts, the Sprint starts
- Sprint Planning has 3 distinct parts with distinct goals
- The Product Owner owns the Product Backlog
- The Product Owner reprioritizes the Product Backlog (daily) based on Value
- Velocity is used to remind the team how much work they are good at accomplishing in an average Sprint.
- The Daily Scrum is held daily
- The mandatory Daily Scrum participants
- The true purpose of the Daily Scrum
- The Increment is considered done, usable and releasable
- The 3 Scrum Artifacts
- The 3 Scrum Roles
- The audience for the Sprint Review
- The expected outcome of a Sprint Review
- The Sprint Review and Sprint Retrospective are seperate events
- The Sprint Retrospective is the only event in Scrum that is solely dedicated to the three pillars of Empiricism (Transparency, Inspection and Adaptation)
- The expected outcome of the Sprint Retrospective
- The Product OWner is the only person on a Scrum Team that can cancel a Sprint
- The only legitimate reason to cancel a Sprint
- The intent behind Product Backlog Refinement is to bring greater transparency to the Product Backlog
- Part of Product Backlog Refinement is to give the Developers an opportunity to size/ estimate the results of Refinement
- The maximum duration of the Daily Scrum
- The Sprint is one of the 5 Scrum events
- The maximum duration of the Sprint
- Sprint Planning is one of the 5 Scrum Events
- The maximum duration of the Sprint Planning
- The Daily Scrum is one of the 5 Scrum Events
- The maximum duration of the Daily Scrum
- The Sprint Review is one of the 5 Scrum Events
- The maximum duration of the Sprint Review
- The Sprint Retrospective is one of the 5 Scrum Events
- The maximum duration of the Sprint Retrospective

The Scrum Values

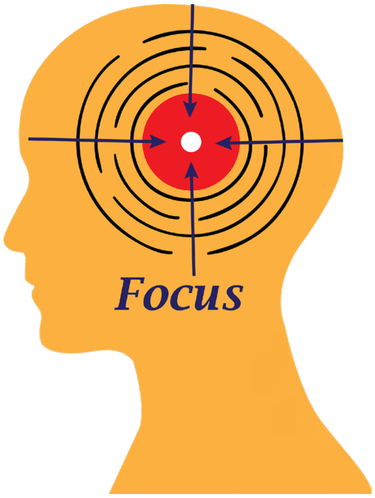
Courage

Openness



Respect

The Scrum Values



Focus

Committment

Scrum Values - Commit to Memory

Acceptance Criteria

Given the situation described in class, write a list of Acceptance Criteria for each of the tasks needing to be done by the worker.

Remember, Acceptance Criteria are a checklist list of sorts - imagine a supervisor testing their work to make sure it is satisfactory. Acceptance Criteria are that short list of tests.

Definition of Done (DoD)

Given the situation described in class, write a Definition of Done for the company that the worker works for.

Remember, Definitions of Done cover ALL of the work that a team will EVER engage in. DoD’s are not specific to the job we are doing right now. They are descriptions of the ‘acceptible level of quality’ that all of our customers expect from us - every single Sprint. DoD’s are NOT rewritten each Sprint. Generally, DoD’s last for long periods of time (sometimes years) before they are modified.

Scrum Team Bill of Rights

Show which Scrum Role is responsible/accountable for the following
(this may be different than who is involved)
Draw an ‘X’ in the appropriate box

SM	Dev	PO	Produce a “Done,” usable, releasable Product Increment each and every Sprint
SM	Dev	PO	Help those outside the Scrum Team understand which of their interactions with the Scrum Team are helpful and which aren’t
SM	Dev	PO	Promoting and supporting Scrum as defined in the Scrum Guide
SM	Dev	PO	Maximize the value of the product resulting from the work of the Development Team
SM	Dev	PO	Define scope for the Developers (Making sure scope has been defined)
SM	Dev	PO	Produce quality work
SM	Dev	PO	Sign up for work, rather than being assigned work
SM	Dev	PO	Improve their development practices
SM	Dev	PO	Implement Action Items
SM	Dev	PO	Removing barriers between stakeholders and the Scrum Team
SM	Dev	PO	Decide when to release
SM	Dev	PO	Ensure the psychological safety of the team
SM	Dev	PO	Estimate Work
SM	Dev	PO	Budgeting - Owns the budget of the Scrum Team
SM	Dev	PO	Experiment with new ideas (multiple answers)
SM	Dev	PO	Assist with Impediments (multiple answers)
SM	Dev	PO	Ongoing visioning
SM	Dev	PO	Ordering the Product Backlog
SM	Dev	PO	Independant product authority
SM	Dev	PO	Causing the removal of impediments (Making sure all impediments are eventually removed)
SM	Dev	PO	Help with Product Backlog Management techniques - including Product Goal definition
SM	Dev	PO	Helping Scrum Team understands the need for clear & concise Product Backlog Items (PBI’s)
SM	Dev	PO	Lead, train and coach the organization in its Scrum Adoption
SM	Dev	PO	Creating a plan for the Sprint (creating a fully useful Sprint Backlog)
SM	Dev	PO	Creating PBI’s and making them transparent to everyone
SM	Dev	PO	Help establish empirical product planning for a complex environment
SM	Dev	PO	Ensuring the Product Backlog is transparent at all times
SM	Dev	PO	Creating and explicitly communicating the Product Goal
SM	Dev	PO	Facilitating stakeholder collaboration as requested or needed
SM	Dev	PO	Instilling quality by adhering to the Definition of Done
SM	Dev	PO	Helping the Scrum Team focus on creating high-value increments that meet the Definition of Done
SM	Dev	PO	Adapting their plan each day toward the Sprint Goal
SM	Dev	PO	Coaching the team members in self-management and cross-functionality
SM	Dev	PO	Holding each other accountable as professionals (Holding themselves accountable)

Many Hats of the Scrum Master

Doctor

Focused on preventing anti-patterns from becoming extreme problems. Prescribes solutions.

Nurse

Promotes team health and prevention; cares for the ill, advocates for a safe and supportive environment. Helps team maintain and improve their health.

Psychologist

Uses interviews and asking questions to identify psychological safety of the team. Connects personally with individuals; honors doctor/patient confidentiality.

Guard Dog

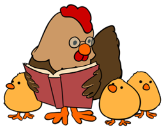
When needed they are the protector of the team; protects someone or something from outside or harmful influences

Mr. Miyagi

Agility is intuitive and woven into their being. Immense amount of research and application experience.

Bloodhound

Listens for spoken and unspoken impediments. Asks questions to seek more information, not to interrogate.

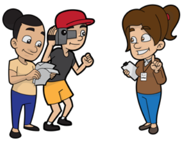


Mother Hen

Doing whatever is needed to protect the team from any harm (Helicopter-Mom)

Agile Junkie

Can't get enough of this 'Agile' stuff. High energy and seeks new experiences.



Guide

Provides assistance, points the way to cultural, technical and historical information. Provides little real information.

Statistician

Produces trustworthy data and analyzes data to make meaning clear; draws practical conclusions; reports data outward from team

Referee

Resides from a neutral point of view, make on the fly decisions and enforces rules

Scientist

Researcher; experimenter; creatively makes hypotheses and tests them out to found out what works (for the team).

Servant Leader

Devotes self to serving the needs of others, focusing on meeting those needs; the model for future leaders

Concierge

Goes the extra mile to make experience happy and memorable for team members

Translator

Communicates the meaning of language to create common understanding

Mentor

Provides examples of past experiences; guides professional development

Coach

Listens, asks focused questions, reflects back, clarifies goals, anticipates obstacles, gives experience-based advice.

Air Traffic Controller

Maintains a safe and orderly flow of work traffic; prevents collisions

Teacher

Educates everyone on Scrum, Agile, and many related areas that help people understand the 'what', 'why', and 'how' of managing work.

Facilitator

Here to make it easier to reach objectives.

Interrogator

Aggressively questions others in search of information about impediments.

Scrum Master as a Servant Leader

Servant Leader

Devotes self to serving the needs of others, focusing on meeting those needs; the model for future leaders

"Servant leadership is a philosophy and set of practices that enriches the lives of individuals, builds better organizations and ultimately creates a more just and caring world."
Please explore more about Servant Leadership, research Robert Greenleaf's work.

You may act as a servant leader in your organization in many ways. Whenever you truly put the needs of others, before your own needs, you are acting as a servant leader.

The Scrum Master serves the Product Owner by coaching them on new/better Product Backlog maintenance techniques. This is a form of servant leadership.

The Scrum Master serves the Development Team by training them on the use of Scrum, getting them proper training in their engineering practices (if needed), and assisting them in removing their impediments. These are all forms of servant leadership.

The Scrum Master serves the entire company by helping other areas determine where Agility can thrive, where Scrum (or other techniques) could thrive; by helping set up Communities of Practice that will bring other employees together around common interests. These are forms of servant leadership.

Scrum Master as a Mentor

Mentor

Provides examples of
past experiences; guides
professional development

Mentoring, as a Scrum Master, only extends to the mentee's career. It's has more to do with helping other Scrum Masters with their career path and career developement than anything else. Often enough Mentoring may include helping a Product Owner or a Developer in their respective career paths, but only when the mentoring Scrum Master has experience in those roles.

Discussions with a Mentor, when what you really need is a Coach, may result in a different kind of advice than you are expecting (see the next page). The advice that a mentor will give will resemble the relating of specific success stories. Mentors are not expected to help anyone navigate their way through difficult challenges of day-to-day work.

Each company/organization with many Scrum Teams should have at least 1 Mentor Scrum Master for every 10 - 15 Scrum Teams. Not everyone needs, or even wants, a mentor.

Scrum Master as a Coach

Coach

Listens, asks focused questions, reflects back, clarifies goals, anticipates obstacles, gives experience-based advice.

Coaching, like Servant leadership, has been part of the definition of Scrum Master for several decades. In the description above, it mentions experience-based advice. Brand-new Scrum Masters may have trouble with this, since they have no experience. While experienced Scrum Masters have an easier time with this, inexperienced people can pull from the experiences outside of Scrum Mastery as well.

While mentoring is about the careers of others, coaching is about the Scrum Master helping others through the trouble they are currently having. Like a good sports coach, a coaching Scrum Master should know when not to jump in, when to tell people what the next right thing to do is, and when to present options.

When pressed for an explanation about their advice, coaching Scrum Masters should be able to give examples from their own experience where they've used certain tactics (of should have used them) to similar problems.

If an inexperienced Scrum Master is trying to coach by giving advice that is NOT based on their own experience, the right thing to do is to tell those being coached where the advice is coming from. Ex: "I read that in a book once", "I was at a conference and this is what they said", "I saw this on LinkedIn."

Scrum Master as a Teacher

Teacher

Educates everyone on Scrum, Agile, and many related areas that help people understand the ‘what’, ‘why’, and ‘how’ of managing work.

The Scrum Master is required to be the one person on a Scrum Team who people can go to and be taught almost anything they need to know about the world of Agility, Scrum, and anything adjacent to Scrum.

We are all educators to some extent. We teach our children how to grow up and be respectible adults. We teach co-workers tricks we’ve learned to navigate the system. We teach our elders how to use current technologies they aren’t comfortable with.

Similarly, the Scrum Master is expected to keep everyone on the Scrum Team knowledgable about how Scrum is supposed to work, and how to become more Agile while using Scrum.

Keep in mind: though you are required to teach, some people will not want to learn.

Scrum Master as a Facilitator

Facilitator

Here to make it easier to reach objectives.

Scrum Masters often misunderstand Facilitation.

Facilitation is not Management.
Facilitation is not Directing.
Facilitation is not Controlling.

“To Facilitate” means “to make easy”. Those who tell you different do not fully understand how to facilitate. Often enough, those same people can be found controlling or managing evets that they are expected to facilitate.

If your role is to facilitate, you will first need to understand why the people you are facilitating are gathered, and what they are expecting to accomplish. Second, you will need to plan out how to facilitate the event. This, in itself may be difficult.

Facilitation does not come natuarlly to everyone. Practice seems to help most people. If you shy away from facilitation opportunities, you may never become very good at this. Introverts and extroverts both do well as facilitators, given they both practice it.

If you would like to learn more about facilitation techniques, you can spend hundreds (and even thousands) of dollars taking classes, or you can read books like Sam Kaner’s “The Participatory Guide to Decision-Making”, and Lyssa Adkins “Coaching Agile Teams”. Both are great resources.

Scrum Team Relationships

What is the majority of the communication between the Product Owner and the Developers about?

What kind of relationship are Scrum Masters supposed to have with Developers?

Describe the relationship between the Product Owner and Scrum Master

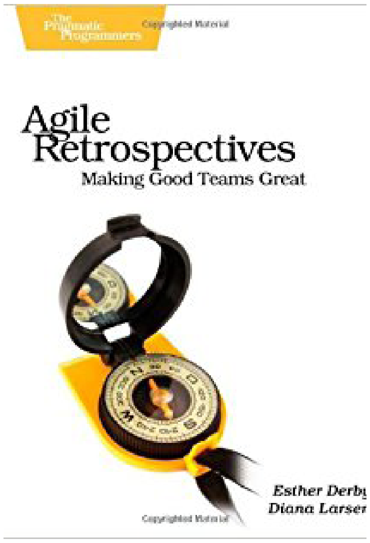
Scrum Team ‘Commit to Memory’

What does ‘Cross-Functional Team’ mean?

What is the ‘right size’ of a Scrum Team?

What are the Scrum Team’s maturity goals? (there are two)

Facilitating Sprint Retrospectives



Start here

Since most people are not going to go out and buy the most important book a Scrum Master could ever read...

Here's a lecture on how to facilitate good Retrospectives.

5 Stage Retrospective

- Setting the Stage
- Gather Data
- Generate Insights
- Decide what to do
- Close

Before the Retrospective

- Start planning the Retro on the first day of the Sprint
- Spend at least as much time planning the Retro as you expect to spend IN the Retro
- Collect data throughout the Sprint, to share during the Retro
- Plan to use engaging activities throughout the Retro - at least for stages 2, 3, and 4
- Plan out what you will say and do - plan out your facilitation

Note: The remainder of this lecture is how Rick facilitates.
Your style will likely be different.

Setting the Stage (5 mins max)

- Welcome everyone
- Orient them to what you are doing here
- Get everyone speking in the first 5 minutes of the meeting
- Get them laughing, if you can

Gather Data (12-20 mins)

- Use a "Context Collector"
- Collect as many ideas, for topics to talk about, as possible
- Group all 'like topics' together (Affinity Grouping)
- Vote on the importance of the topics

Generate Insights (35 mins)

- Use an activity to draw out the information below
- Identify the impact - What is the risk, if this is something that has not yet occurred
- Identify where this risk came from (root cause) - 5 Why's; Fishbone
- Identify all known information about the topic
- Identify if this topic is within the Team's control to resolve, or if they would need help
- Do not solution the problem yet!!!
- Keep these discussions short, so you can have several of them
- Collect all topical information from each small group before starting another discussion
- BE VERSATILE!

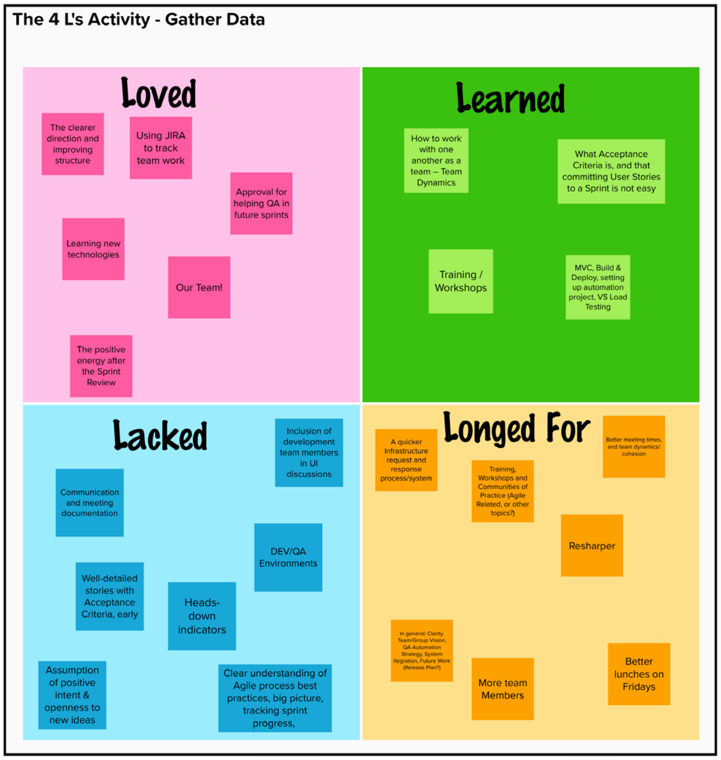
Facilitating Sprint Retrospectives

Decide what to Do (35 mins)

- Use an activity to develop a plan of action
- The goal of this stage is for the team to develop a plan to fix at least 1 single problem
- By the end of this stage, the entire team needs to come to a consensus on the plan
- Fixing 1 problem is best for immature and stable teams; fixing more than 1 problem per sprint is for more advanced teams

Close (5 mins)

- Thank everyone for their participation and for being an awesome team
- Summarize everything the Scrum Team accomplished during the Retro
- Tell the team how much being part of the team means to you
- ‘Appreciations’ is an activity best saved for teams that don’t have egos at all



The 4 L's Activity - Gather Data

Loved

The clearer direction and improving structure

Using JIRA to track team work

Approval for helping QA in future sprints

Learning new technologies

Our Team!

The positive energy after the Sprint Review

Learned

How to work with one another as a team – Team Dynamics

What Acceptance Criteria is, and that committing User Stories to a Sprint is not easy

Training / Workshops

MVC, Build & Deploy, setting up automation project, VS Load Testing

Lacked

Inclusion of development team members in UI discussions

Communication and meeting documentation

DEV/QA Environments

Well-detailed stories with Acceptance Criteria, early

Heads-down indicators

Assumption of positive intent & openness to new ideas

Clear understanding of Agile process best practices, big picture, tracking sprint progress,

Longed For

A quicker Infrastructure request and response process/system

Better meeting times, and team dynamics/ cohesion

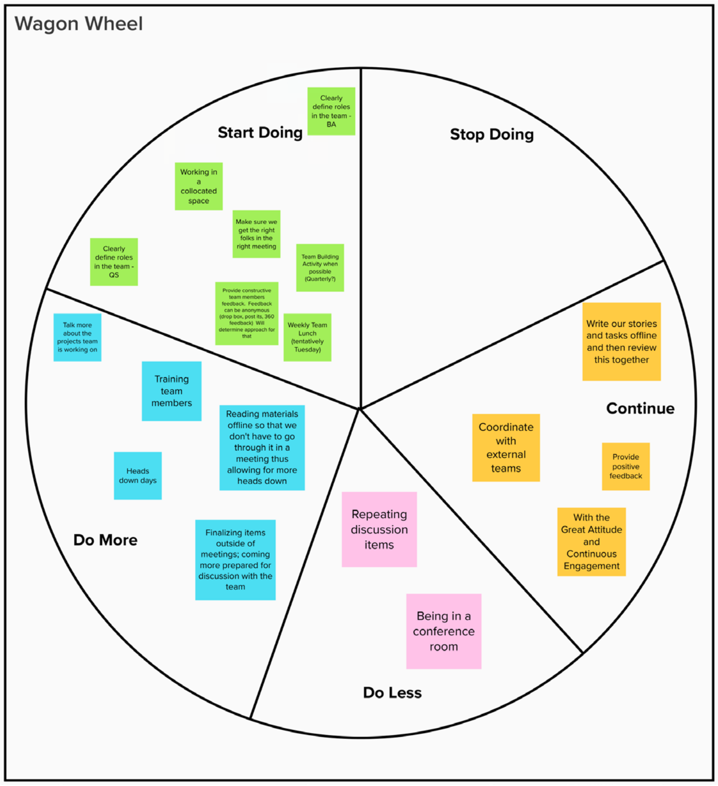
Training, Workshops and Communities of Practice (Agile Related, or other topics?)

Resharper

In general: Clarity. Team/Group Vision, QA Automation Strategy, System Itegration, Future Work (Release Plan?)

More team Members

Better lunches on Fridays



Wagon Wheel

Start Doing

Clearly define roles in the team - BA

Working in a collocated space

Make sure we get the right folks in the right meeting

Clearly define roles in the team - QS

Team Building Activity when possible (Quarterly?)

Provide constructive team members feedback. Feedback can be anonymous (drop box, post its, 360 feedback) Will determine approach for that

Weekly Team Lunch (tentatively Tuesday)

Stop Doing

Continue

Write our stories and tasks offline and then review this together

Coordinate with external teams

Provide positive feedback

With the Great Attitude and Continuous Engagement

Do Less

Repeating discussion items

Being in a conference room

Do More

Talk more about the projects team is working on

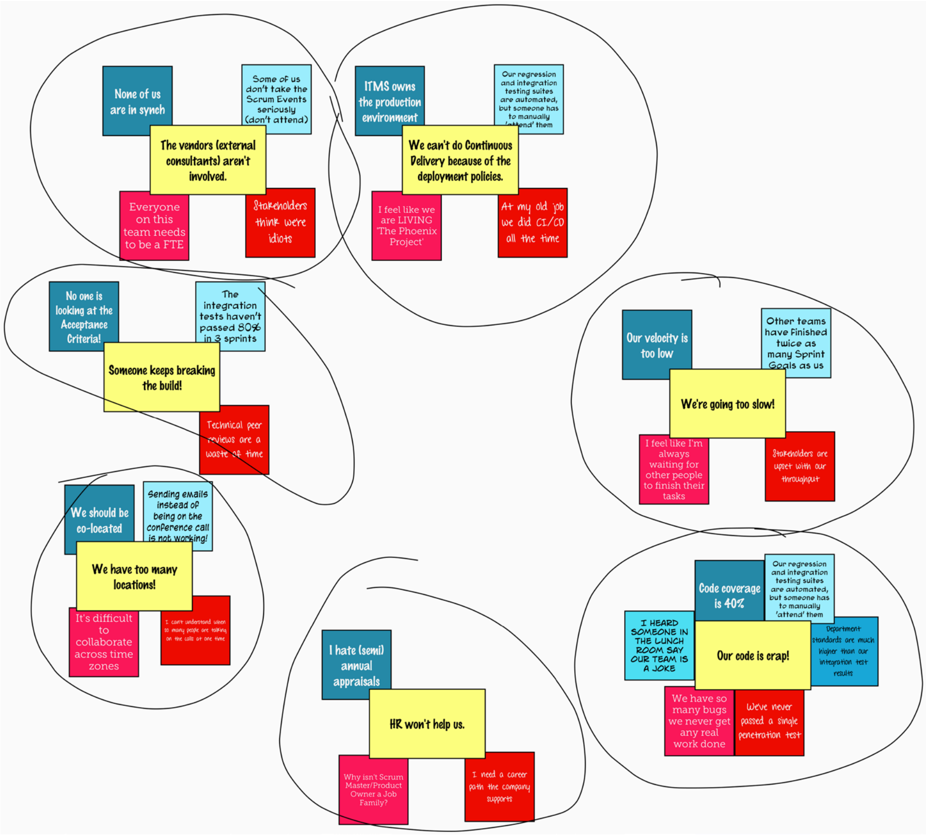
Training team members

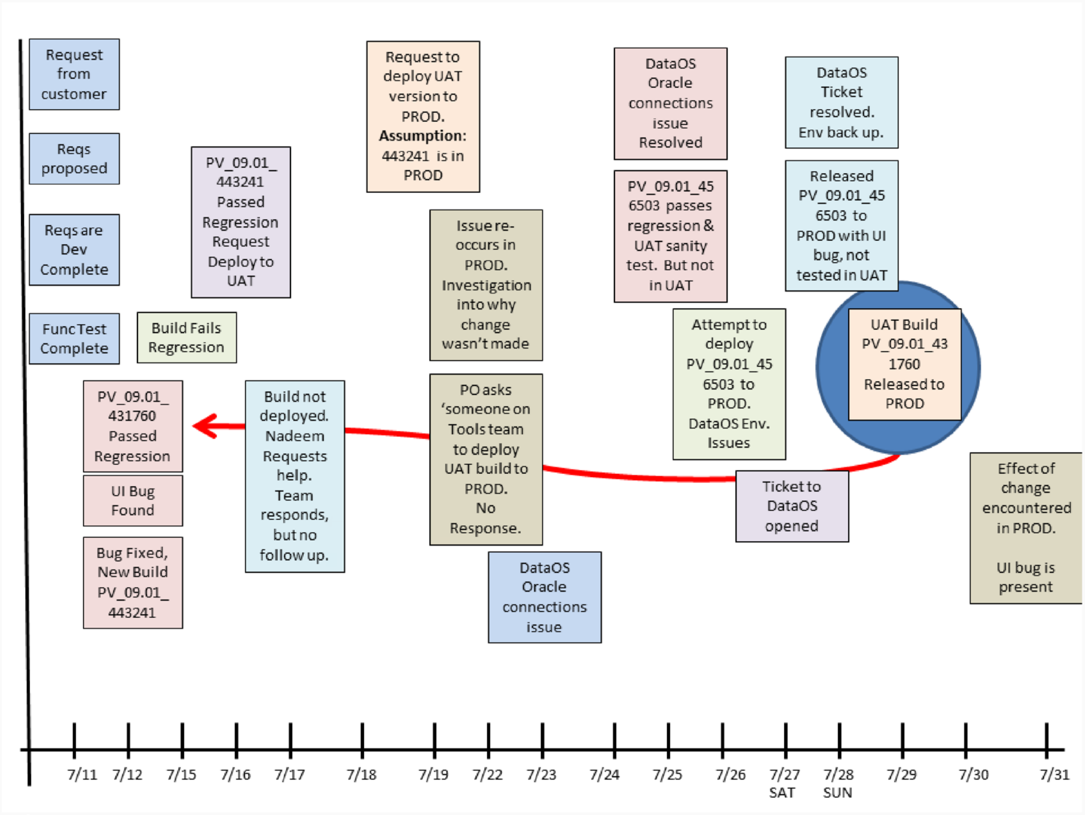
Reading materials offline so that we don't have to go through it in a meeting thus allowing for more heads down

Heads down days

Finalizing items outside of meetings; coming more prepared for discussion with the team

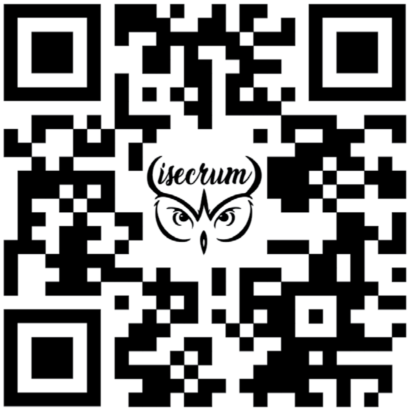
Facilitating Sprint Retrospectives


None of us are in synch
Some of us don't take the Scrum Events seriously (don't attend)
The vendors (external consultants) aren't involved.
Everyone on this team needs to be a FTE
Stakeholders think we're idiots
ITMS owns the production environment
Our regression and integration testing suites are automated, but someone has to manually 'attend' them
We can't do Continuous Delivery because of the deployment policies.
I feel like we are LIVING 'The Phoenix Project'
At my old job we did CI/CD all the time
No one is looking at the Acceptance Criteria!
The integration tests haven't passed 80% in 3 sprints
Someone keeps breaking the build!
Technical peer reviews are a waste of time
Our velocity is too low
Other teams have Finished twice as many Sprint Goals as us
We're going too slow!
I feel like I'm always waiting for other people to finish their tasks
Stakeholders are upset with our throughput
We should be co-located
Sending emails instead of being on the conference call is not working!
We have too many locations!
It's difficult to collaborate across time zones
I can't understand when so many people are talking on the calls at one time
I hate (semi) annual appraisals
HR won't help us.
Why isn't Scrum Master/Product Owner a Job Family?
I need a career path the company supports
Code coverage is 40%
Our regression and integration testing suites are automated, but someone has to manually 'attend' them
I HEARD SOMEONE IN THE LUNCH ROOM SAY OUR TEAM IS A JOKE
Our code is crap!
Department standards are much higher than our integration test results
We have so many bugs we never get any real work done
We've never passed a single penetration test


Request from customer
Reqs proposed
Reqs are Dev Complete
Func Test Complete
Build Fails Regression
PV_09.01_ 431760 Passed Regression
UI Bug Found
Bug Fixed, New Build PV_09.01_ 443241
PV_09.01_ 443241 Passed Regression Request Deploy to UAT
Build not deployed. Nadeem Requests help. Team responds, but no follow up.
Request to deploy UAT version to PROD. Assumption: 443241 is in PROD
Issue re-occurs in PROD. Investigation into why change wasn't made
PO asks 'someone on Tools team to deploy UAT build to PROD. No Response.
DataOS Oracle connections issue
DataOS Oracle connections issue Resolved
PV_09.01_45 6503 passes regression & UAT sanity test. But not in UAT
Attempt to deploy PV_09.01_45 6503 to PROD. DataOS Env. Issues
Ticket to DataOS opened
DataOS Ticket resolved. Env back up.
Released PV_09.01_45 6503 to PROD with UI bug, not tested in UAT
UAT Build PV_09.01_43 1760 Released to PROD
Effect of change encountered in PROD. UI bug is present
7/11
7/12
7/15
7/16
7/17
7/18
7/19
7/22
7/23
7/24
7/25
7/26
7/27 SAT
7/28 SUN
7/29
7/30
7/31

Reference QR Codes

Scrum Guide (Website)



Agile Manifesto (Website)

CSM Resources (Website)

Office Hours (Website)

Mural Board (Website)

Wisecrum Website

Iterative & Incremental (Video)

Sprint Goals (Video)

Reference Materials

Books
Agile Software Development with Scrum - Schwaber & Beedle
Agile Retrospectives - Derby & Larsen
The Age of Agile - Denning
Turn the Ship Around - Marquet
Coaching Agile Teams - Adkins
The Nature of Software Development - Jeffries
The Skilled Facilitator - Schwarz
Unlocking Agility - Hesselberg
Software for Your Head - McCarthy & McCarthy
Drive - Pink
The Phoenix Project - Kim, Behr, Spafford
Amelia Bedelia - Parish
Orbiting the Giant Hairball - MacKenzie
Core Scrum - Jeffries, Hendrickson
The 5 Dysfunctions of a Team - Patrick Lencioni
The New New Product Development Game

Websites
http://agilemanifesto.org
http://scrumguides.org
http://wisecrum.com
http://formulaink.com
http://mypersonalagility.org
http://scrumalliance.org
http://retromat.org
http://ronjeffries.com
http:/AgileGames.ca
http://poppendieck.com
http://objectmentor.com
http://martinfowler.com
http://www.scrumplop.org/
http://scrumguides.org/scrum-guide.html
https://pragprog.com/titles/dlret/agile-retrospectives
http://www.romanpichler.com/blog/one-page-product-owner/
http://scrummasterchecklist.org

Videos
On YouTube: @AgileWaters
Sh*t Bad ScrumMasters Say
The Cynefin Framework
Silicon Valley Season 1, Episode 5
Scrum Master - The Power of Scrum
Fix the Plumbing - by Erin Perry
Office Space - I Have People Skills
The Lone Nut and the First Follower