
Go From *Doing* Agile to *Being* Agile

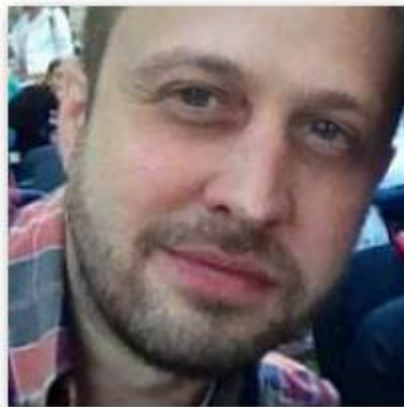


November 4, 2016

About Us

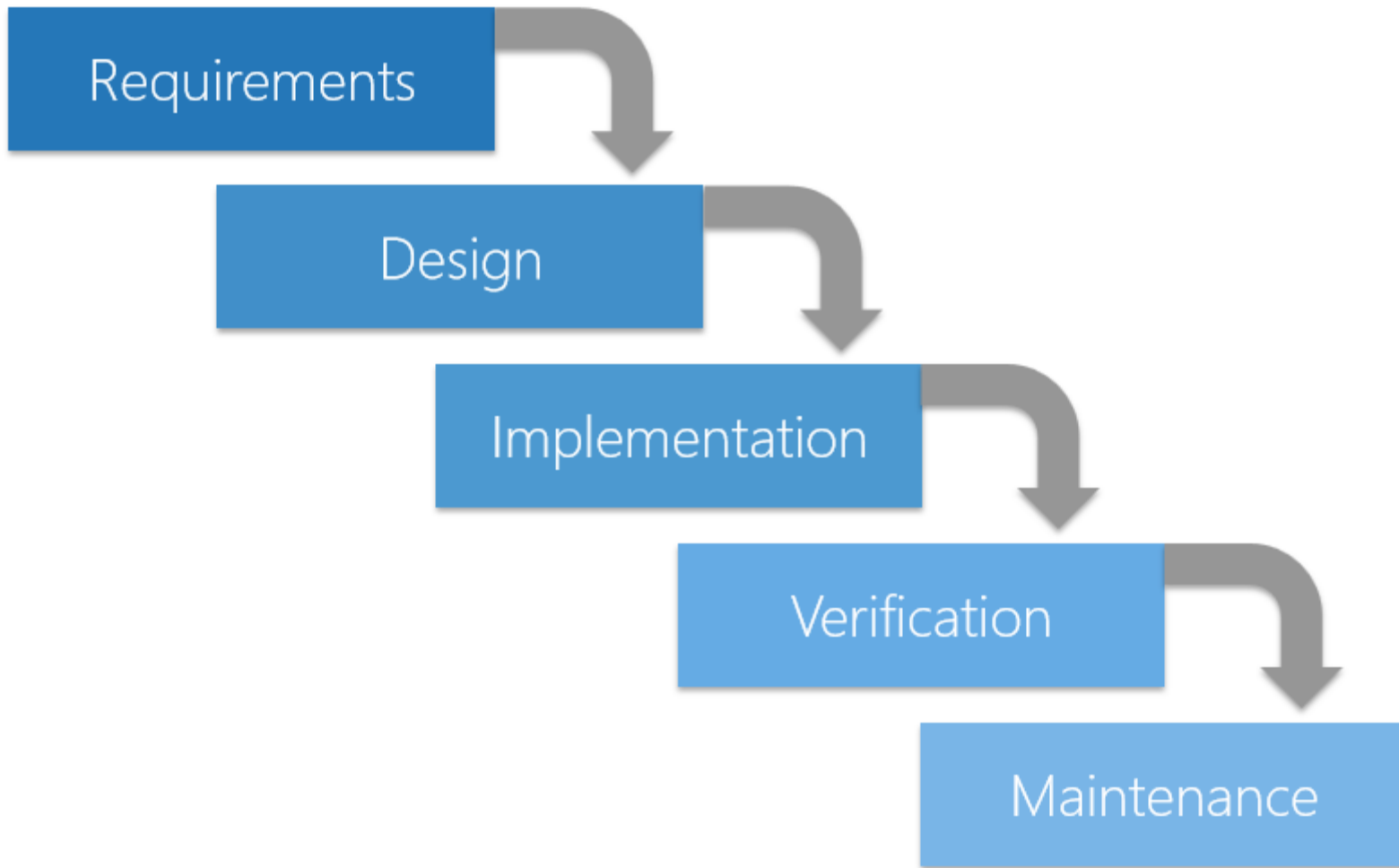


- ▶ Maria Hamidi, Associate Product Manager/Product Owner, Morningstar
maria.hamidi@morningstar.com



- ▶ Jason Stipp, Director, U.S. Individual Investor Products, Morningstar
jason.stipp@morningstar.com

Build: Journey to Agile Development



Build: Journey to Agile Development

Where we came from...

- ▶ Long, multi-month projects developed with waterfall methodology
- ▶ Siloed disciplines and hand-offs of requirements from product to design, design to development, and development to QA (and often back again)
- ▶ Large product launches after months of coding and refinement
- ▶ Inflexible development schedule
- ▶ Little upfront testing with customers
- ▶ Products driven by internal stakeholders

Our first steps toward Agile...

- ▶ Shorter development sprints with testing and iteration
- ▶ Building cross-functional squads
- ▶ More frequent, but smaller, launches
- ▶ Greater flexibility to shift priorities if needed
- ▶ More testing with customers and product optimization
- ▶ Products driven by customer needs, guided and prioritized by internal stakeholders

We Started *Doing* Agile (kind of...)

- ▶ We built quasi cross-functional teams of front-end and back-end developers
- ▶ Had both U.S. and Shenzhen Scrum Teams, but they were not fully cross-functional
- ▶ Design, QA, project managers, and business analysts supported the Scrum Teams but were not technically part of them
- ▶ We started a Wiki for requirements and story development, facilitating review and feedback between product, business analysts, and technical subject matter experts
- ▶ We started using Jira for project tracking, story development, and defect management
- ▶ We implemented sprint planning and daily stand-ups
- ▶ Held sprint retrospectives, but they were perfunctory
- ▶ We released every two weeks

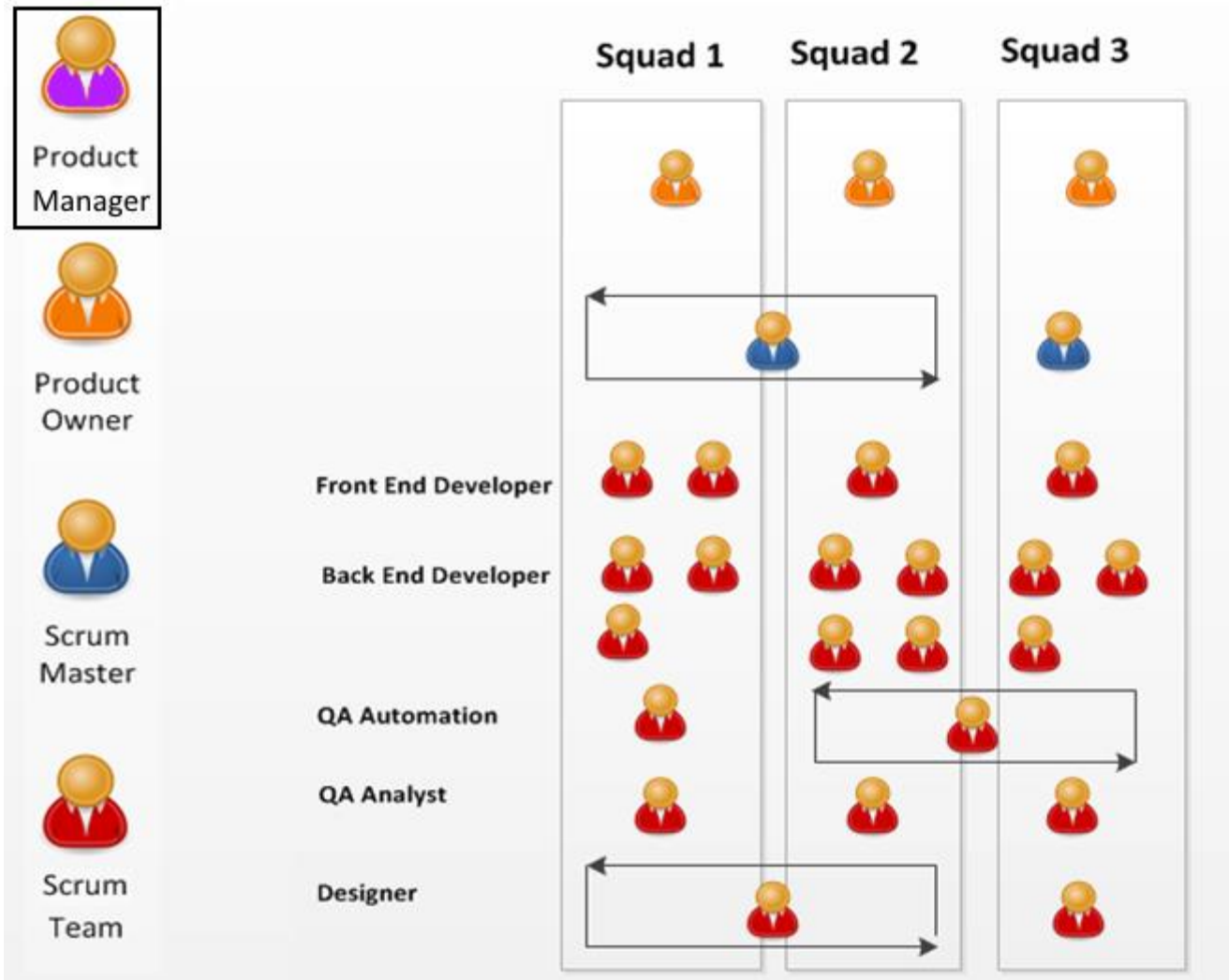
But...We Were Doing “Scrummer-Fall”

- ▶ Scrum Team members were passive participants in sprint ceremonies
- ▶ Project Managers ran the stand-ups with team members “reporting in” to PMs around a TV monitor
- ▶ No “Product Owner” at the scrum level; priorities were communicated from Product Managers to Scrum Team via Project Managers
- ▶ Product Managers and other stakeholders could “meddle” with the sprint
- ▶ Scrum Team members were assigned stories, with each focusing on his/her own work
- ▶ Lots of stories were allowed to “carry over” from sprint to sprint when they did not meet acceptance criteria or finish on time

Now We're Working on *Being* Agile

- ▶ New process and roles are focused on achieving 3 key criteria for *being* agile:
 - ▶ Empowerment
 - ▶ Collaboration
 - ▶ Accountability/Ownership
- ▶ Scrum Team is self managed, commits to stories and divides the work as a group, and is collectively accountable for all story delivery and defining a strategy to meet their commitment
- ▶ Scrum Team members encouraged to be more collaborative: hands-on versus hand-offs
- ▶ New "Product Owner" role manages backlog, writes stories, and owns prioritization for Scrum Team
- ▶ Product Managers work with Product Owner on changing/new priorities for backlog prioritization, but cannot directly assign work to the team
- ▶ Stories that don't meet requirements are rejected by Product Owner; Scrum Team doesn't get credit for rejected work

Our New Scrum Team Structure



Role: Product Owner



Product Owner is empowered to make product decisions and is responsible for:

- ▶ Owning the plan and prioritization for executing short-term (sprint-level) product roadmap objectives (next 3-6 months)
- ▶ Creating executable stories (functional and nonfunctional requirements) to form a product backlog; defining “what” needs to be delivered
- ▶ Continuous backlog grooming and prioritization of stories and new priorities
- ▶ Working with stakeholders to balance competing priorities and manage scope creep
- ▶ Managing and communicating with stakeholders to keep them informed
- ▶ Accepting or rejecting sprint work
- ▶ Project/sprint reporting
- ▶ Milestone, release and launch planning/management
- ▶ Medium-term dependency and risk management

Role: Scrum Team



- ▶ Cross functional team of 5-7 people--developers, designers, QA--is responsible for delivering sprint story commitments, including:
 - ▶ Estimating backlog stories
 - ▶ Committing to sprint work
 - ▶ Defining (and possibly revising) a strategy to ensure the team stays on target
 - ▶ Helping each other when needed to meet sprint commitments
 - ▶ Ensuring stories are completed while meeting the “Definition of Done”
- ▶ Scrum Team (with help of Scrum Master) handles short-term dependency and risk management during development sprint
- ▶ Whole team actively participates in all Scrum sprint ceremonies

Role: Scrum Master



Responsible for:

- ▶ Ensuring unbiased execution of all scrum ceremonies and scrum process
- ▶ Impediment (roadblock) removal
- ▶ Shielding the team from external interferences
- ▶ Coaching Scrum Team members and Product Owner on the scrum process

Supporting Role: Product Manager



May observe but does not directly participate in sprint ceremonies. Responsible for:

- ▶ Understanding user needs/jobs (customer/market knowledge)
- ▶ Driving/defining market and user research
- ▶ Creating product vision, value proposition, and roadmap to meet long-term business objectives
- ▶ Managing costs, budgets, pricing in order to meet revenue/profit expectations
- ▶ Owning product lifecycle management
- ▶ Defining and reporting on product KPIs/metrics
- ▶ Handling longer-term dependency and risk management
- ▶ Being a subject matter expert for Scrum Team
- ▶ Resource for PO in sprint prioritization

Supporting Role: Tech Lead



Attends but does not directly participate in sprint ceremonies.

- ▶ Not responsible for completing any of the work assigned to the Scrum Team, nor for any sprint planning or strategizing/organizing how work gets done
- ▶ Does not assign/prioritize work for the Scrum Team, needs to direct it all through Product Owner
- ▶ Acts as a technical advisor to the Scrum Team, may do some code review or sign-off
- ▶ Can participate in scrum ceremonies but as a guest or observer
- ▶ In rare cases, can assume the role of a Scrum Team member if taking on actual development in a sprint

Individual Investor: Squads



Scrum Process

Input from Executives,
Team, Stakeholders,
Customers, Users



Product Owner

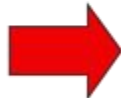


The Team



Product Backlog

Ranked list of what is required:
features, stories...



Team selects, starting
at top, as much as it
can commit to deliver
by end of sprint.

Sprint Planning
Meeting



Sprint Backlog

Sprint end date and team
deliverable do not change



Burn-down/up
Charts



Scrum
Master



Daily Scrum
Meeting

Every
24 hours



2-week
Sprint



Sprint Review



Finished Work



Sprint Retrospective

Scrum Artifacts

Product Backlog

- ▶ Owned by the Product Owner
- ▶ Prioritized set of requirements (user stories)
- ▶ Estimated with story points
- ▶ Needs to be frequently re-prioritized (grooming); can remove and add requirements over time as more information is available

Sprint Backlog

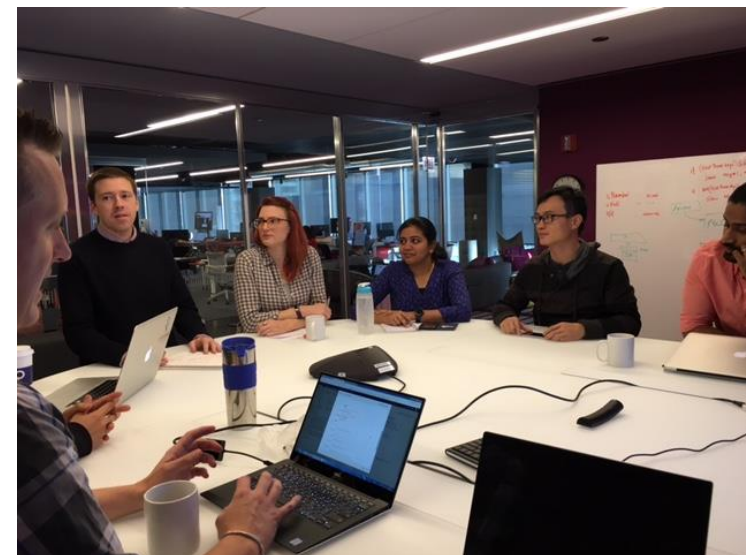
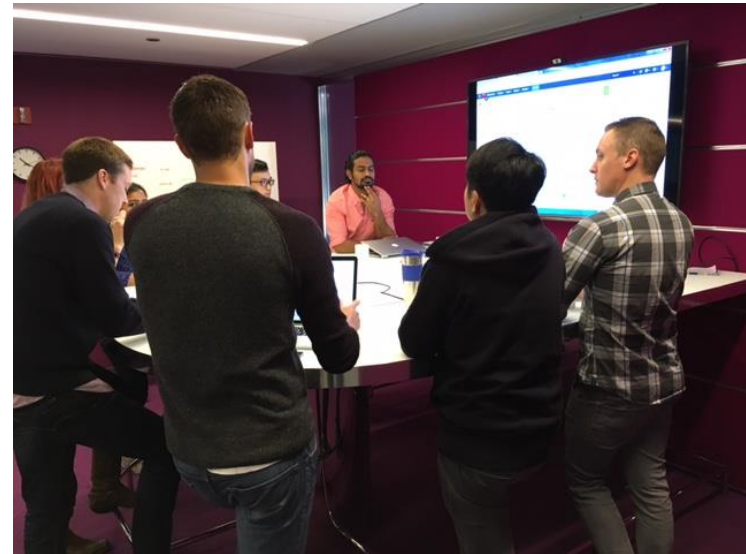
- ▶ Set of requirements (user stories) for a specific sprint
- ▶ Details and work needed for each user story

Product Increment

- ▶ Working software: Sum of all product backlog items completed during the sprint, along with what was previously completed.
- ▶ Increment delivered must meet the “Definition of Done” as agreed upon by Product Owner and team members.

Ceremonies: Backlog Grooming

- ▶ Before ceremony, Product Owner updates prioritized backlog of stories, incorporating stakeholder feedback and accounting for any new items
- ▶ Scrum Team meets to review un-estimated stories
- ▶ Scrum Team members ask questions about and discuss story details and potential dependencies
- ▶ Scrum Team members individually estimate points for each story: 1, 2, 3, 5, 8, 13
- ▶ Point estimate differences are discussed and the team collectively decides on an estimate
- ▶ After estimates are given, Product Owner may confer with stakeholders and reprioritize backlog



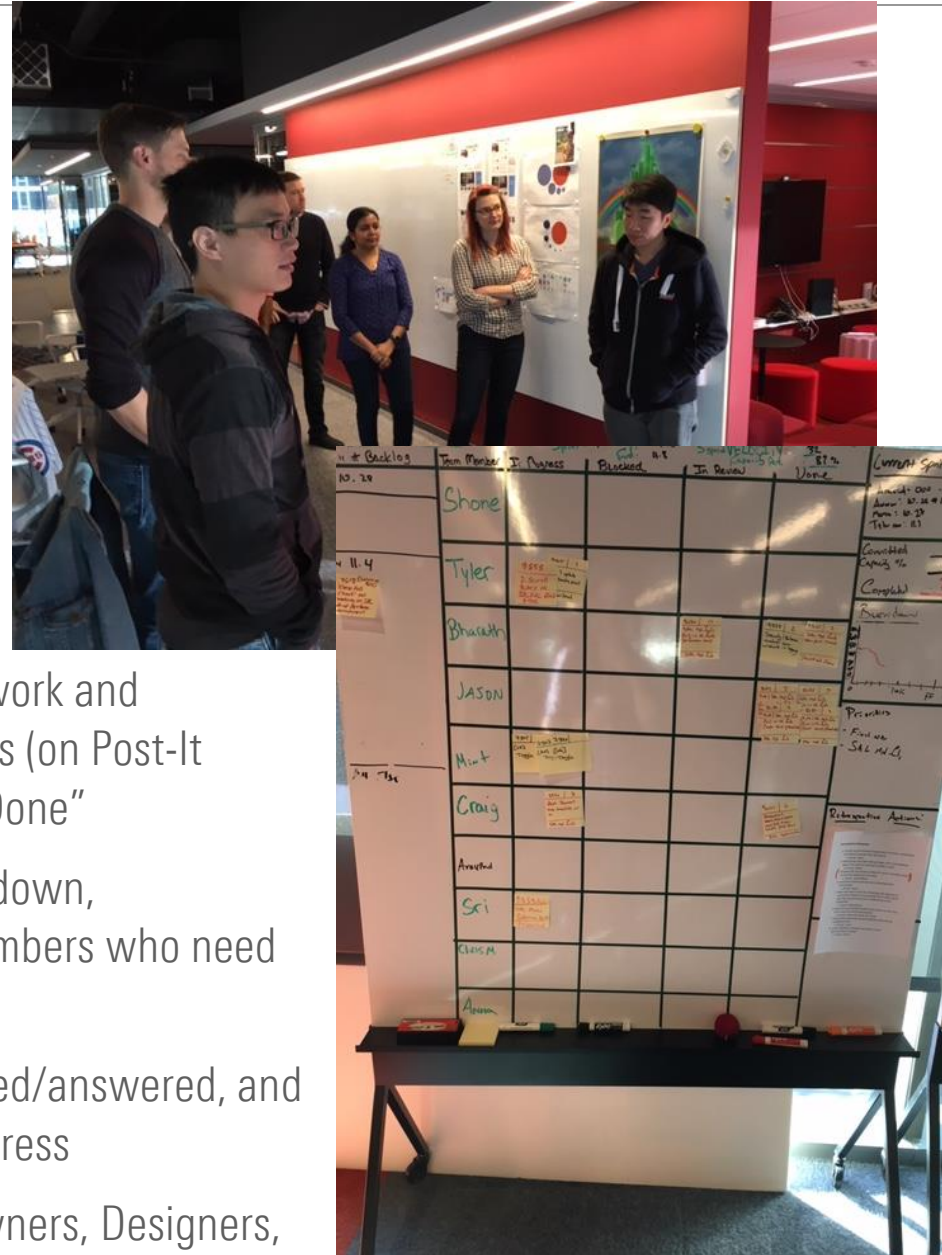
Ceremonies: Sprint Planning

- ▶ Scrum Team meets to plan for upcoming sprint. Vacations, training, and other known absences are accounted for, and any functional resource constraints are discussed (e.g., if QA will be shorthanded during a sprint).
- ▶ Team capacity is calculated for the sprint.
- ▶ Team reviews prioritized backlog stories (highest, second highest, third highest...) and commits to a certain number of them based on their capacity and resources available in the sprint and negotiation with Product Owner.
- ▶ Team will not commit to work on stories that cannot be completed (meeting our “Definition of Done”) in the sprint.
- ▶ Committed stories are pulled into the sprint.
- ▶ Scrum Team meets separately to finalize strategy among themselves for competing committed work.



Ceremonies: Daily Standup

- ▶ Scrum Team meets daily in the morning for 15 minutes
- ▶ Scrum Master facilitates and Product Owner listens, but neither is actively involved in the standup
- ▶ Scrum Team members update each other on work and update Visual Board, moving committed stories (on Post-It Notes) from “In-Progress” to “In Review” to “Done”
- ▶ Visual Board also shows blocked stories, burndown, committed vs completed points, and team members who need help
- ▶ Blockers are surfaced, story questions are asked/answered, and team confirms sprint strategy and checks progress
- ▶ Team may plan “desk checks” for Products Owners, Designers, and stakeholders to review work in progress



Ceremonies: Sprint Review

- ▶ Scrum Team gathers for final review of completed stories
- ▶ Product Owner reviews each story against the acceptance criteria (ACs) and Definition of Done
- ▶ Product Owner accepts or rejects stories
- ▶ Only accepted stories go into production. Team does not get credit for rejected stories. (Scrum Team may negotiate with PO on rejected stories, but PO has the final word.)
- ▶ Rejected stories are reprioritized in the backlog.
- ▶ No carryover work is allowed; however, if an AC was misunderstood or a late bug is discovered, PO may choose to accept a story as-is, letting it go to production, but create and prioritize a related follow-up story for a future sprint.

Ceremonies: Sprint Retro

- ▶ Prior to sprint end, the Scrum team takes the Sprint Retrospective survey to gather feedback.
- ▶ Survey questions consist of questions regarding team sentiment, what worked well, and what didn't work well.
- ▶ Scrum Team uses the Sprint Retro session to review the results of the survey, add any additional feedback, and define action items for improvement for the next sprint.
- ▶ Owners are assigned for action items in the next sprint and progress against those action items is reviewed during each daily stand up.
- ▶ The Sprint Dashboard is also reviewed which consists of various KPIs to track team progression.

Scrum Tools: Definition of Done (DoD)

What it is

- ▶ A list of activities (writing code, coding comments, unit testing, integration testing, release notes, design documents, etc.) that add verifiable/demonstrable value to the product that must be completed before code is released
- ▶ Applies to all functions – Dev, QA, Design, Product

Why it's important

- ▶ Shared understanding of expectations that software must meet in order to be high quality and releasable into production; everyone must have the same definition of “done”
- ▶ Defined activities at two levels – Story & Release

Process for creating/enforcing

- ▶ Management team consisting of members from each function (Dev, QA, Product, Design) works together to define the activities to be included within DoD
- ▶ Training session conducted for all team members to introduce and review DoD
- ▶ All team members equipped with tools, skills, knowledge necessary to execute on it
- ▶ DoD will be validated during story review; story will be accepted or rejected
- ▶ DoD reviewed and update on an ongoing basis

Tools: Definition of Done

Definition of Done – Story Level Checklist

Item	Dev	Design	QA	Product
Code Complete				
Unit Tests Pass – New & Existing	X			
Functional Tests Pass	X		X	X
Selenium Tests Pass	X		X	
Defined Standards Followed	X	X	X	X
Browser & Device Support Checked	X	X	X	X
Code Reviewed	X		X	
Performance Impact Assessed	X		X	X
Used automation identifier	X		X	
Added helper text in author edit dialogues	X		X	X
Accessibility	X	X	X	X
Deployment configured				
Environment differences should be configured	X			
Data source approval documented	X			X
Non-AEM changes noted in wiki				X
Operational readiness documented				
Documentation noted in wiki	X			
Monitoring & Logging				
Created page(s) for major configs	X			
Followed logging standards	X			

Definition of Done – Release Level Checklist

Item	Dev	SM	QA	Product
Performance & Load Testing				
Define Site Speed & Load Requirements	X			X
Run & Review Site Speed Testing	X		X	X
Run & Review Load Testing	X		X	X
Security				
Security Scan Tests	X			
Operational Readiness				
Regression Tests Passed	X		X	
Ops will attend release go/no go meetings	X			
Train Customer Support				X
Dependent Partner Signoff	X			X
Release & Communication Plan Completed		X		X
Project, Launch & Communication Plan Completed		X		X
CAB Approval		X		

Scrum Tools: Working Agreement

What is it?

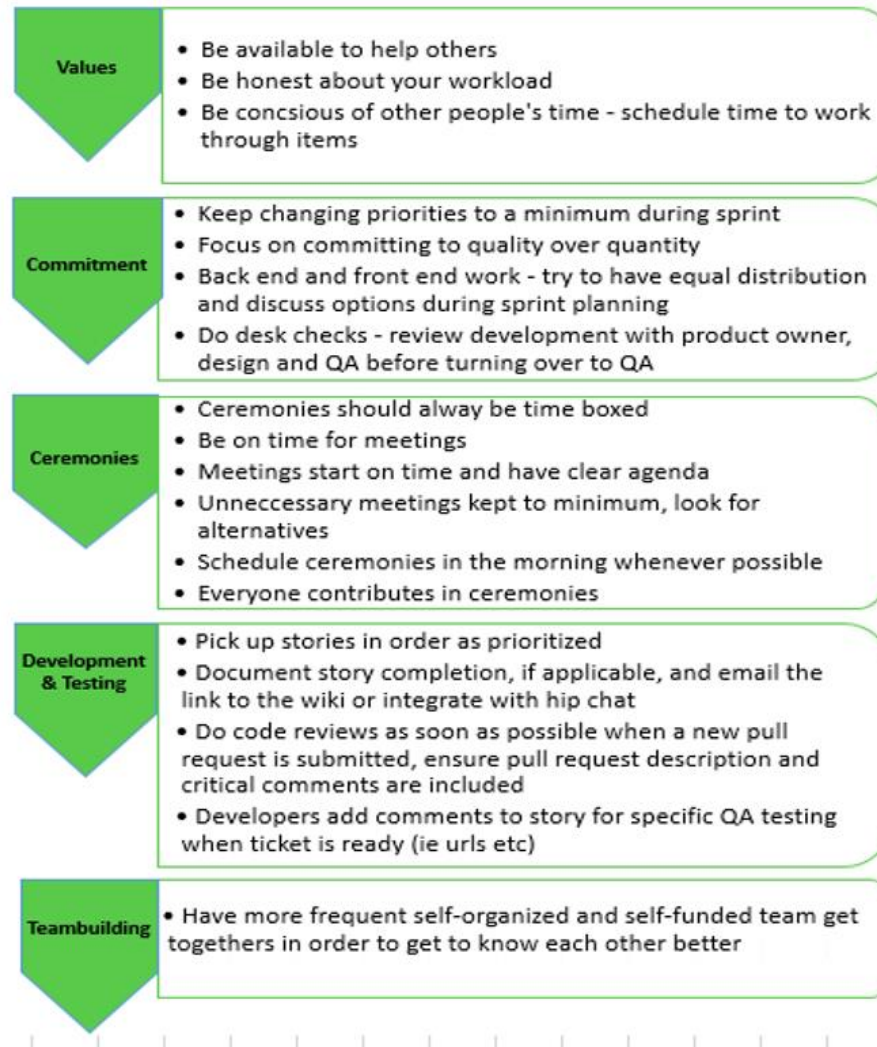
- ▶ Set of values, behaviors, ground rules or guidelines the Scrum Team itself defines and agrees to follow to make themselves more efficient and have a positive, productive process
 - ▷ E.g., “Update the visual board and Jira regularly” and “Be on time for scrum meetings”
- ▶ Describes positive behaviors that are basic but not often automatically demonstrated
- ▶ Develops a sense of shared responsibility on the team
- ▶ Increases members’ awareness of their own behaviors and helps avoid conflicts

Once completed:

- ▶ Team members reinforce the work agreement with each other if someone is not adhering
- ▶ Should be displayed in the team’s working area, revisit with team during retrospective, etc.
- ▶ Scrum Master is the custodian of the agreement, regularly reviewing at sprint retro.
- ▶ Agreement is revisited for transparency and continuous improvement to determine if updates are needed

Sample Working Agreement

EMERALD SQUAD WORKING AGREEMENT



Agile Transformation Learnings

- ▶ *Doing* the agile ceremonies doesn't necessarily mean you're getting the benefit of the agile process.
 - ▶ Rather, the ceremonies promote and reinforce a culture of ownership and collaboration that must be present on the Scrum Team, as well as alignment of product/business priority and scrum work. *Agile is as much a mind shift as a work-process shift.*
 - ▶ In order for teams to be accountable, they must be *empowered* and *engaged*.
 - ▶ **Agile won't make dev teams work faster:** You will *not* get 12 developers' worth of work from 6 developers during a sprint by using agile.
 - ▶ What agile can do:
 - ▶ Ensure your product-development team is aligned and working on the most important things (as defined by Product Managers and Product Owners) at any given time
 - ▶ Breaks work into completable units, leading to realistic commitments and providing visibility into true velocity
 - ▶ Ensures the team is working *collaboratively* versus handing off work from one person to another
 - ▶ Brings a sense of ownership and *accountability* to a team
 - ▶ Allow for necessarily pivots to happen sooner
- ...All of which **boost sentiment** and **reduce rework, waste, risk and (in the long run) time.**