scrum_cheat_sheet.md

Scrum Cheat Sheet

Created by: Brendan Herger Created: 2017-02-13 Updated: 2017-02-14

tl;dr: Scrum is an approach (interface) to allow developers to quickly and efficiently build useful products. It is defined by 3 roles, 4 meetings, and 3 artifacts. Implementing this interface has helped many groups to run more efficiently.

Intro

Scrum is a common project management paradigm that allows software engineers to efficiently build software. Similar to building and using a CI/CD framework or micro-service framework, it allows software engineers to pay a bit more upfront to reduce friction and reduce overall work.

Scrum as an interface

It is an interface, in the Java sense of interface, which defines certain roles, meetings and artifacts which a team promises to implement. This interface has been optimized to:

- Minimize meeting time
- Regularize workload
- Keep projects tightly tied to customer need

Unfortunately, most engineers experience with scrum has been ... lackluster (including my own). This is likely due to shoddy or half-assed implementations of the scrum interface, rather than the interface itself.

Time management

A core principle of scrum is to quickly build a product, identify feedback, and iterate on the product. This avoids building needless features (clippy), or getting to attached to an inefficient approach.

The atomic unit of time in scrum is a sprint (usually 1, 2, 3 or 4 weeks). During each sprint a team will:

- Produce an atomic, documented and integrate-able unit of work
- Hold each of the planning / recap meetings once

People

Role	Brief Description	Primarily works with
Scrum Master	Team coach	Product Owner, Developers
Product Owner	Work filter	Customers
Developer	Doer	Scrum Master

Scrum Master

Someone familiar with the scrum interface, and responsible for implementing it. Key responsibilities include:

- Identifying and removing blockers

- Coordinating other scrum roles
- Coaching team in scrum interface
- Acting as buffer between developers and external roles

Product Owner

Someone responsible for making sure the product being built is valuable to the customer, and that the extraneous feature requests are removed from the backlog. Key responsibilities include:

- Owns customer relationship
- Owns, filters and prioritizes product backlog
- Accepts backlog items from customer, developers
- Coordinates with scrum master to provide project time lines

Developer

People who are able to build the product, and willing to partake in scrum process. Key responsibilities include:

- Estimates effort necessary to complete backlog items
- Accepts backlog item(s) to work on for sprint
- Provides feedback on scrum process
- Provides new backlog items for future sprints

Meetings

Meeting	Frequency	Max duration	Brief Description	Input artifacts	Output artifacts
Sprint Planning	1 x sprint		Accept work to complete this sprint	Product Backlog	Sprint Backlog
Sprint Review	1 x sprint		Review (demo) work completed this sprint		Product Feedback, updated Product Backlog
Retrospective	1 x sprint		Review what went well, what didn't go well, and what can be done differently next sprint		Items to change during next sprint
Daily Standup	1 x day	15 minutes	Identify blockers, individual status updates		

Sprint Planning

An opportunity for the team to choose which features can and will be completed by the end of sprint

- Review the filtered and prioritized product backlog
- Accept work that can be completed during the sprint
- Create sprint backlog with features to be completed this sprint

Sprint Review

An opportunity to demo work that has been completed, and solicit feedback

- Demo work that has been completed during the sprint

- Gather feedback on current implementation of product
- Update produce backlog w/ feedback

Retrospective

An opportunity to iterate on the scrum implementation

- Review scrum implementation for previous sprint
- Identify what went well, what went poorly, and what should be changed
- Identify one item to change, plan to change it in the next sprint

Daily Standup

- Identify blockers
- Individual's status updates

Artifacts

Artifacts act as working documents, concrete interactions between roles, and archival records.

Artifact	Update Frequency	Owner	Brief Description	Relevant Meetings
Product Backlog	Constant	Product Owner	Prioritized list of features that will be implemented	Sprint review, Sprint planning
Sprint Backlog	1 x sprint	Scrum Master	List of features that will be implemented by the end of the sprint	Sprint planning
Product Increment	1 x sprint	Team	Self contained, deployable product including features on sprint backlog	Sprint review

Product Backlog

A constantly evolving list of features that might be worked on.

- Filtered to remove unrealistic / unnecessary features
- Prioritized, to guide work that is accepted into the Sprint Backlog
- Maintained by the Product Owner, who makes sure that all features bring value to the customer

Sprint Backlog

A list of features that has been accepted for the current sprint.

- Work is taken from the product backlog
- Work can be completed by the end of the sprint
- Work results in self contained, deployable product increment

Product Increment

A self contained, deployable product, that could be released to the customer.

Vocab

Sprint: Atomic unit of time, usually 1, 2, 3 or 4 weeks. **Demo:** Present atomic unit of work, created over one sprint, to an audience including technical competent audience members and product customers. **Feature:** Atomic unit of work, generally can be completed by one team member during one sprint (or less). These are features of the product that represent add value to the customer. **Velocity:** (Output) / (Unit time)

Buzzwords / Catch Phrases

Use these words, and people will think you're a pro scrum master.

- **Build one to throw away:** Quickly build a proof of concept to understand the problem space and data, then start from scratch to avoid technical debt
- **Let's put that in the parking lot:** Your question is not valuable to the group, or is distracting to the current conversation. Let's save it for after the meeting.
- **Inspect and adapt:** Evaluate current iteration, use evaluation to inform next iteration
- **Have we heard from everyone?:** Technique to get everyone to speak / implicit group aporval
- **Must have, should have, could have, won't have:** Backlog prioritization tool
- **Weighted shortest job first:** (cost of delay) / (job duration)
- **Collective ownership:** Everyone on dev team can interact with a resourge (e.g. everyone can modify a database, removing the bottleneck around a database guru)
- **Scrum is silent about ...:** Because scrum is an interface, it does not care about implementation