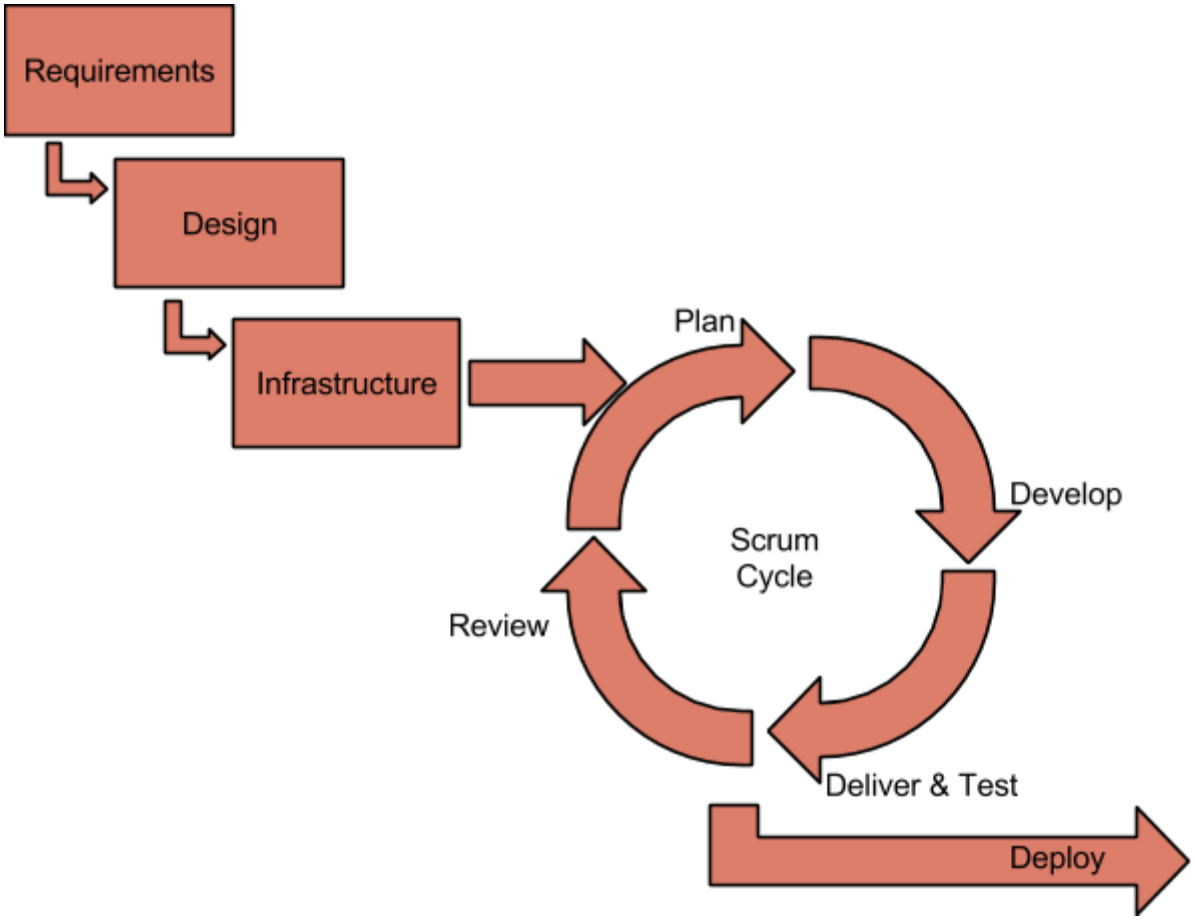
Team 4YourHealth
Trillium Health
Senior Project
2015

Process Description

“Scrumfall” High Level Process Flow



Stages

Requirements: Phase of development focused on understanding the problem

For our process, The Requirements phase had additional aspects not encountered in some software development. Our team was new to both each other and the customer, as well as much of the domain knowledge (Healthcare). Therefore, in addition to gathering requirements, our team also had to quite a bit of research into the domain. Learning to communicate within the team and externally also took place during this stage. The so-called "forming, storming, norming, and performing" process largely took place during requirements gathering.

Most of the work in this phase went towards producing the SRS, a comprehensive document outlining what the customer wants in detail. It is this document that much of our subsequent work was based upon.

Major Tasks in the Requirements Phase:

- *Reviewed Problem statement*
- *Discussions with users and product owner*
- *Requirements gathering*
- *Production of SRS*
- *Use Case Descriptions*
- *Role Assignment*

Design: Phase of development focused on creating a solution to the problem

We separated the work of our Design stage into the three distinct components of our proposed solution. First, the Database design, a schema which describes the tables and fields we would need in our database. Second, the Clinician Portal, a user interface for the clinic employees to view and edit patient, session, and document information. Finally, the Session Application, a GUI design for how the users sign into and out of sessions.

There were many choices to be made in the design phase, mostly concerning what technologies we should use. We made these choices with easy deployment foremost in our minds. Given our short project lifespan, it was important to choose tools that our team was comfortable with but also easy to set up for the customer.

Major Tasks in the Design Phase:

- *Technology choices*
- *Database Schema*
- *GUI concepts*
- *Backend API*

Infrastructure: Phase of development dedicated to preparing for implementation

The infrastructure stage is not one that usually exists in process models. However we added it to our workflow to describe the work done between design and our iterative design

stages. Tasks in this stage included creating our build process, setting up the virtual machine as a development platform, and creating our repository.

This “infrastructure” stage is distinct from its neighbors because the deliverables are not created for the customer, and therefore, it does not make sense to include them into the iterative development cycles. Work was still tracked by tasks, but it is difficult to present the finished work to the customer. However, the work was still essential to our process, as its completion was a prerequisite for efficient and successful implementation to come.

Major Tasks in the Infrastructure Phase:

- *Build process*
- *VM setup*
- *Database setup*
- *Repository setup*

Scrum Cycles: Phases of development dedicated to implementation

During implementation we fully embrace the iterative scrum process. Now that we are able to deliver working user stories to the customer and project owner every sprint, the team can rapidly react to feedback using the four stages below.

- **Plan -** Decisions are made on what to focus on during the current sprint, workload is divided amongst team members. If changes to the project plan are necessary they are made during this phase.
- **Develop -** User stories are fully realized in code and implementation. New changes are pushed through the build process.
- **Deliver & Test -** Working user stories are shown to the customer and project owner.
- **Review -** Feedback from the customers, project owner, and team is assessed. Decisions on whether or not to refactor a user story are made.

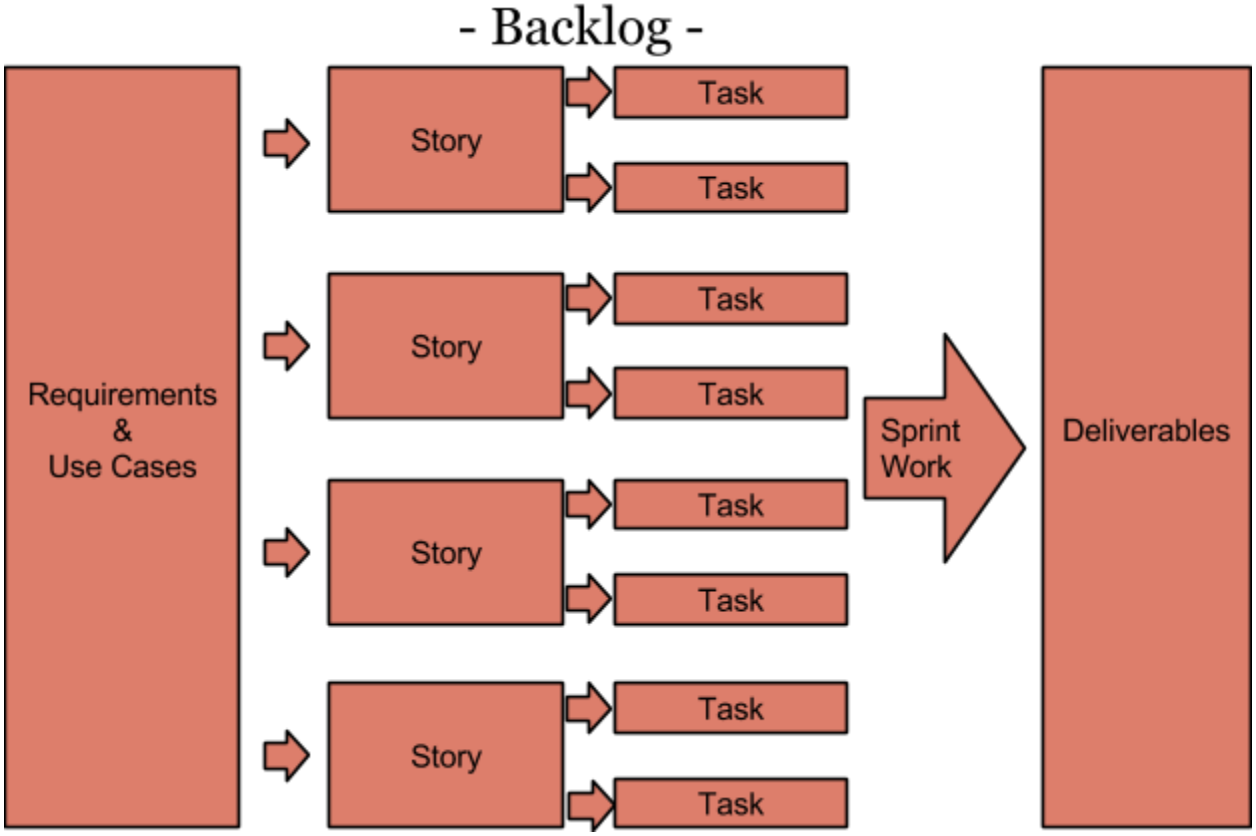
Deploy: Phase of development dedicated to the use of the completed system

The completed system will be fully deployed to the customer with enough time for the team to perform early maintenance.

Workflow - Tasks and Stories

Throughout development, even during the stages borrowed from traditional waterfall, our team's work has been organized into tasks and stories. This system presented many advantages to us. We were able to spread work out amongst a large team, and in many cases adjust workloads week to week based on how availible various team members were.

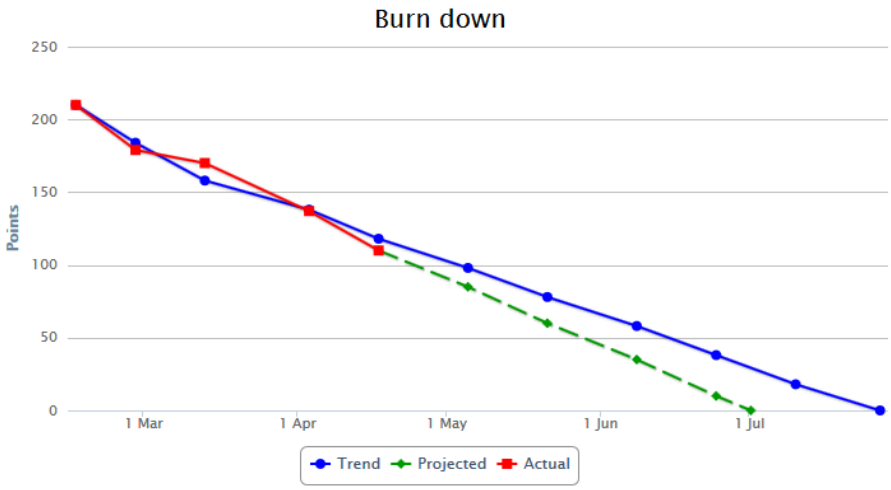
The project backlog is an essential resource for our team's process. It is this backlog that has guided our process sprint-to-sprint and is a crucial bridge between the project timeline and our requirements (mostly the use cases). Early on though, there were a few issues with this approach. Before the team had found a good pace for its work, estimates for task completion were often inaccurate. Additionally, tasks that did not produce customer deliverables had to be self-verified on occasion. But on the whole, using a backlog has improved our process vision and task efficiency.



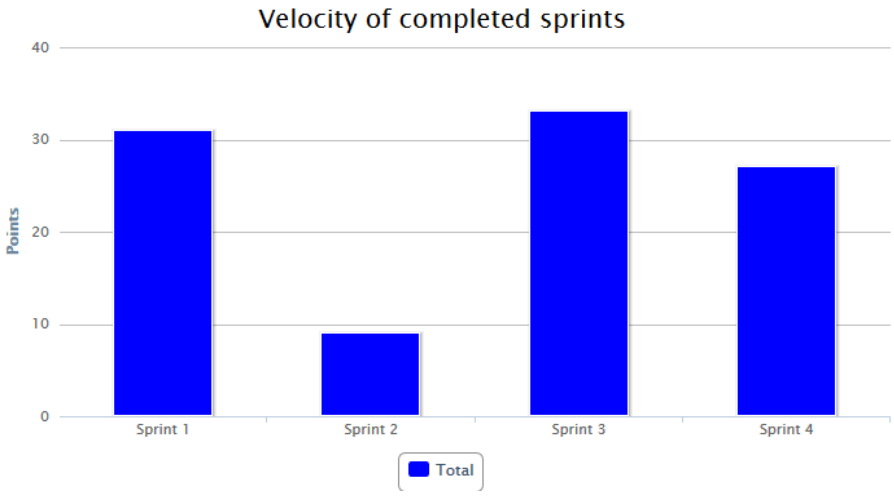
Metrics

We used the following metrics to measure our teams efficiency and success both with following our process model and delivering timely deliverables. Study of the metrics from early on in development has allowed us to make changes to improve our process. Task estimation has become more accurate, and hours spent on the project are more effective when dedicated to more specific tasks.

Story Points:
Use story points to track what amount of time was spent in each phase.

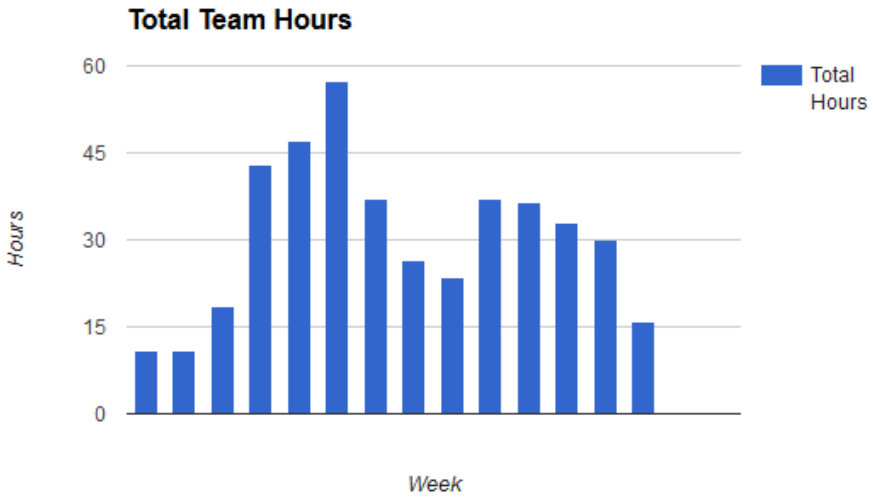


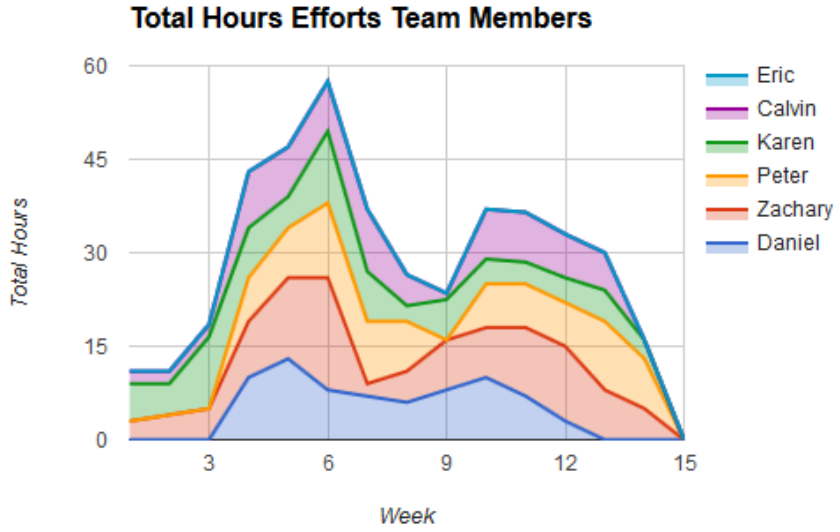
Here you can see the burndown of our backlog, in keeping with our project plan, main tasks are projected to complete in July, leaving the team time to deploy to the customer



Here you can points completed in each sprint. Sprint 1 was the Requirements Stage, Sprints 2 and 3 were Design, and Sprint 4 was Infrastructure.

Hours:
We used work hours to measure the accuracy of task estimations and to determine if team members had a balanced workload.





*NOTE, this is a living document and hours for the most recent 2 weeks do not reflect actual work, as they have yet to be completed.

Issues:
Once we have begun to deliver code for customer feedback and testing, our team will be tracking reported issues as a metric of code quality and testing effectiveness.