

Lecture 2: Process, Estimation, and Agile

Michael Hilton
SE for Startups
1/17/2018

Administrivia

Fill out the schedule quiz, linked from Canvas, so we can assign teams.

Apologies for the out-of-cycle deadline; we need to assign the teams by Tuesday, so we need the info before then.

Have you confirmed Canvas access yet?

Software Process

“The set of activities and associated results that produce a software product”

STARTUP-SPECIFIC PROCESS CONCERNS?

“Agile” Software Development Is ...

Both:

a set of software engineering principles and best practices (allowing for rapid delivery of high quality software)

a business approach (aligning development with customer needs and goals)

Agile in a nutshell

A project management approach that seeks to respond to change and unpredictability, primarily using incremental, iterative work sequences (often called “sprints”).

Also: a collection of practices to facilitate that approach.

All predicated on the principles outlined in “The Manifesto for Agile Software Development.”

Agile principles are well-suited for startup context.

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Agile principles are well-suited for startup context.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Agile principles are well-suited for startup context.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Agile principles are well-suited for startup context.

Simplicity—the art of maximizing the amount of work not done—is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

PRINCIPLES ARE NICE, BUT A BIT HIGH-LEVEL...

Agile Practices

- **Backlogs (Product and Sprint)**
- Behavior-driven development (BDD)
- **Cross-functional team**
- **Continuous integration (CI)**
- Domain-driven design (DDD)
- Information radiators (Kanban board, Task board, Burndown chart)
- **Acceptance test-driven development (ATDD)**
- Iterative and incremental development (IID)
- Pair programming
- Planning poker
- Refactoring
- **Scrum meetings (Sprint planning, Daily scrum, Sprint review and retrospective)**
- Simple design

PROCESS INTERVENTIONS ARE USEFUL INASMUCH AS THEY PREVENT PROBLEMS.

Collective Ownership

Every programmer improves any code anywhere in the system at any time if they see the opportunity.

Test-driven development

Programmers write unit tests minute by minute. These tests are collected and they must all run correctly. Customers write functional tests for the stories in an iteration.

Short development cycle

The software development process is organized in a way in which the full software development cycle—from design phase to implementation phase to test and deployment phase—is performed within a short timespan, usually several months or even weeks.

Continuous Integration (CI)

New code is integrated with the current system after no more than a few hours. When integrating, the system is built from scratch and all tests must pass or the changes are discarded.

Small Releases

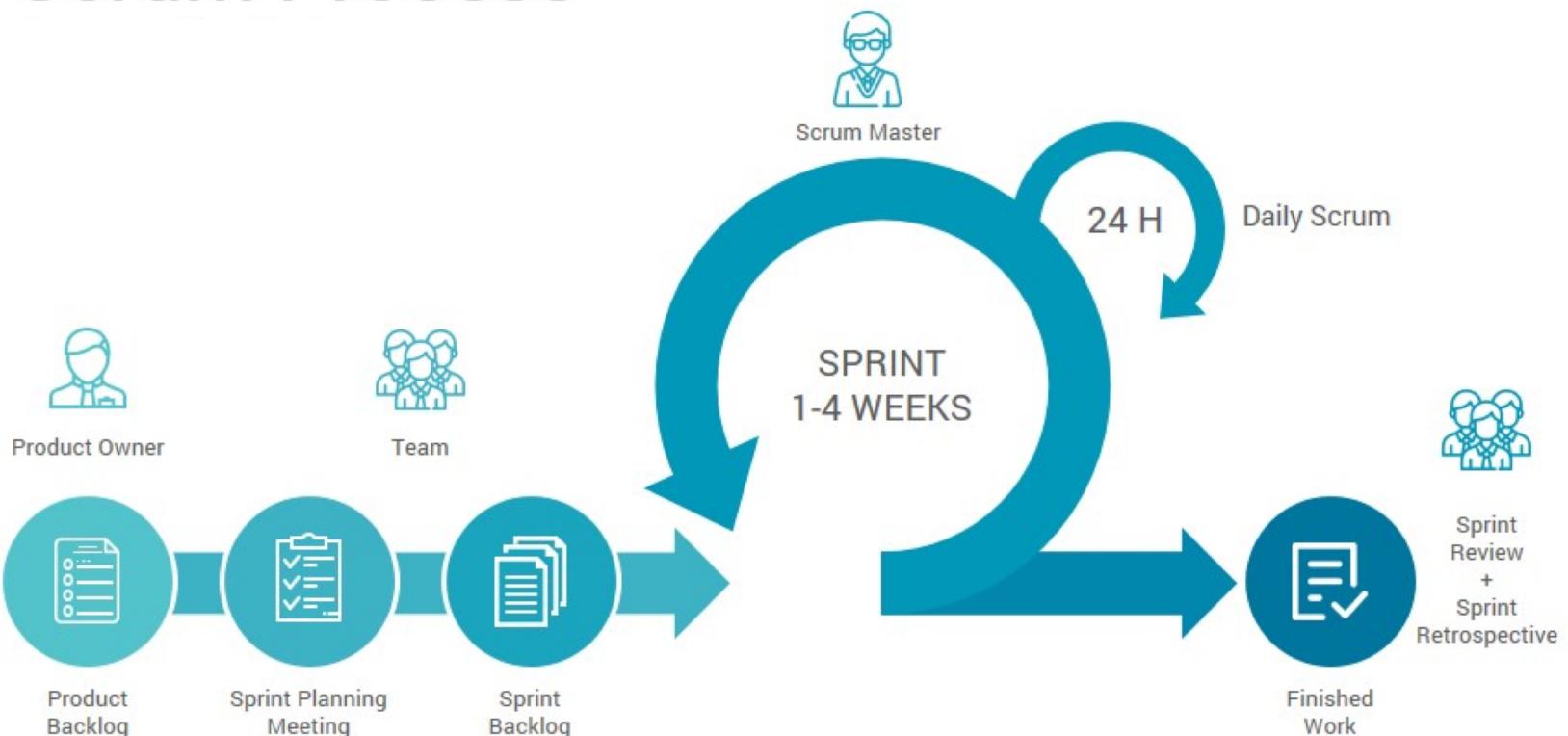
The system is put into production in a few months, before solving the whole problem. New releases are made often—anywhere from daily to monthly.

Continuous Delivery (CD)

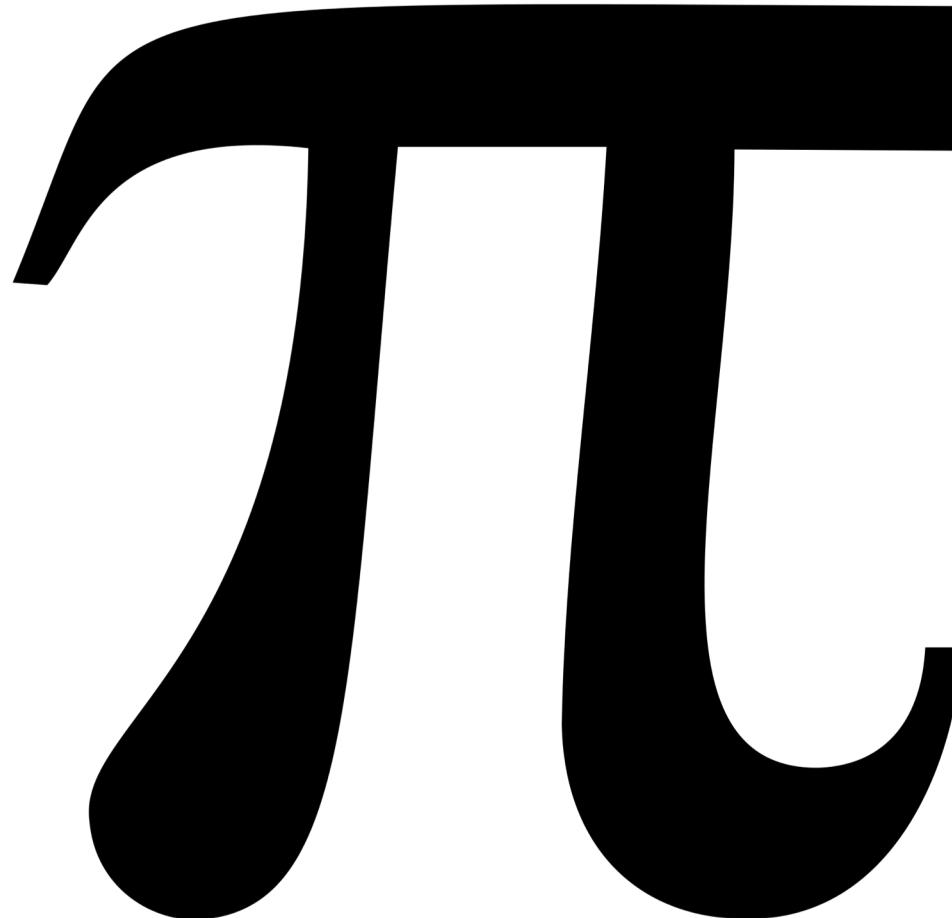
New releases are deployed automatically (not necessarily to production, but in a production-like environment) for early adopters to use and assess.

Scrum: Structure and Communication

Scrum Process

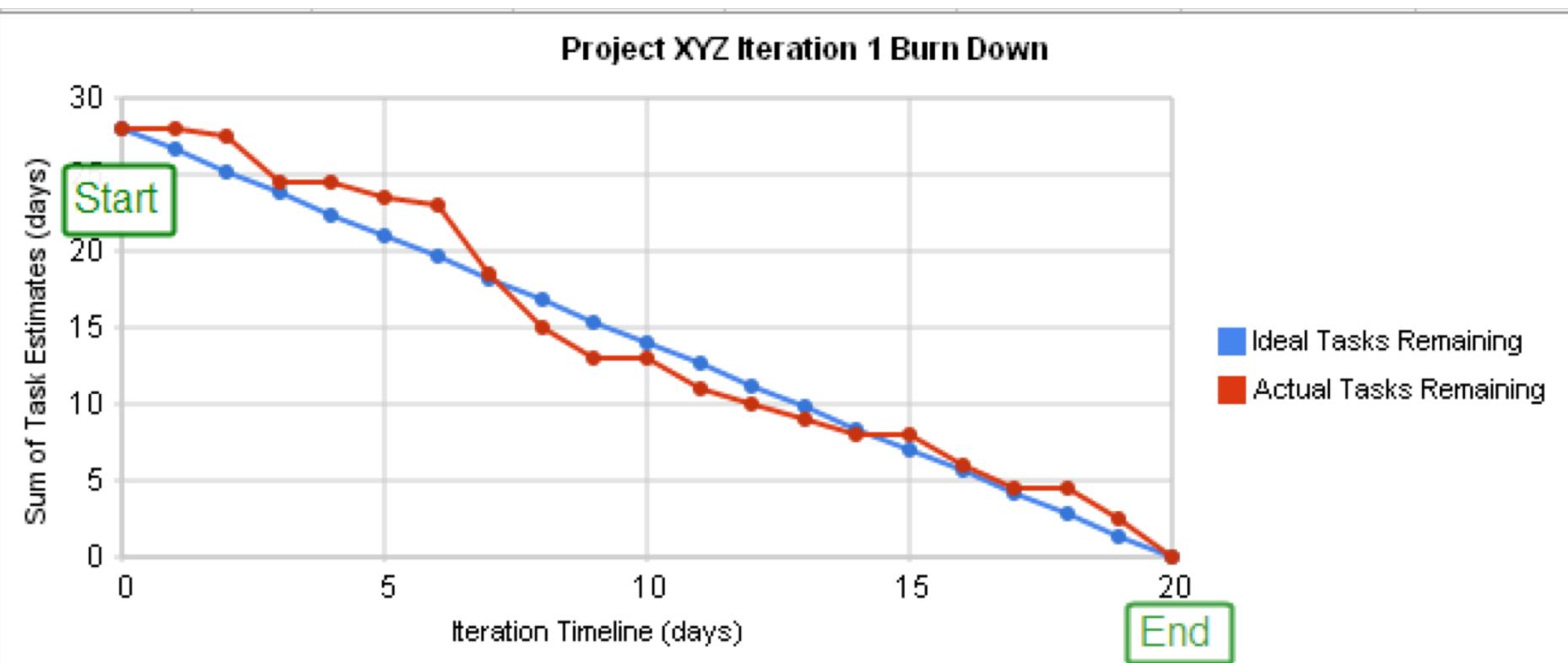


Sprint planning: estimation



Question: Why is this so hard?

Burndown charts: now you've seen one

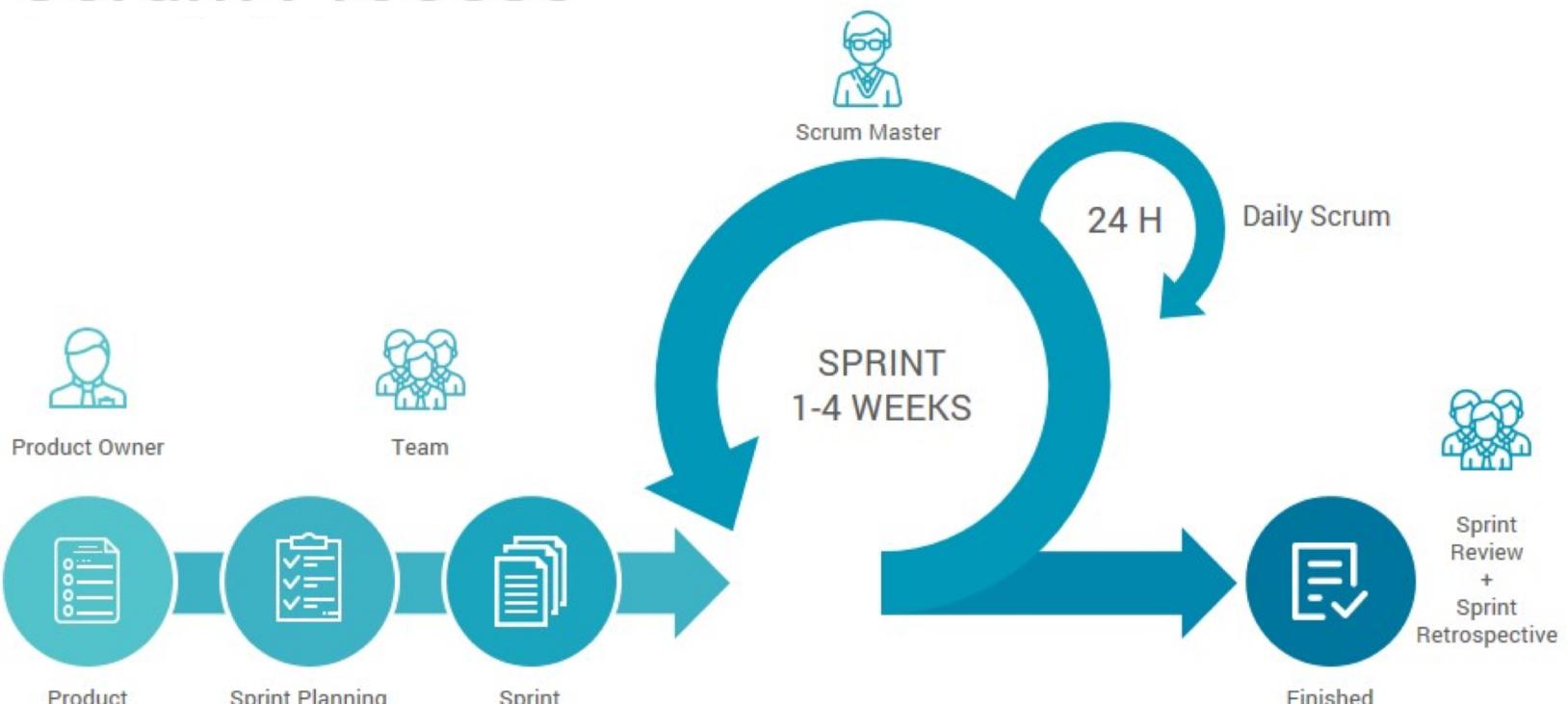


More realistic...



Roles: product owner, team, scrum master

Scrum Process



Sprint planning

Product Backlog

List of user stories for the product

All entries should add value

It is a living document (should be groomed)

No low level tasks

Backlog is ordered

Sprint planning

Sprint Backlog

Tasks to be done that sprint.

Features (user stories, broken into features and tasks)

Bugs

Technical work

Knowledge acquisition

Activity

 bill peduto 
@billpeduto

 Following ▾

Exactly. We will bid it out this year. A “route smart” system will not only create micro-districts for drivers to be accountable, it will also determine salt domes and fuel stations to keep them in the road. Today, we operate on clipboards & paper & expect them to succeed.

Daniel Winne @thedigitalpit
Replying to @billpeduto

I wonder if @Uber or @googlemaps could give the city an algorithm for where the trucks should drive most efficiently.

8:00 PM - 16 Jan 2018

12 Retweets 71 Likes

9 12 71 

Sprint review and retrospective

Sprint Review: End of sprint meeting: inspect increment, adapt the backlog. Based on changes to backlog, attendees decide what could be done next to optimize value.

- Discuss what has been done, effect on backlog; team demonstrates what has been done.
- Discusses what went well and what problems were solved, and how.
- Discussion of what to do next (including potential changes in the product use case or market).

Sprint retrospective: team inspects itself and creates a plan for self-improvement for the next Sprint.

- Occurs after the review, before planning.
- Team discusses:
 - a. What went well in the Sprint
 - b. What could be improved
 - c. What will we commit to improve in the next Sprint

Scrum meetings

Daily “stand-ups”, strictly time-boxed to 15 minutes, where each team member answers three questions:

1. What did you do yesterday?
2. What will you do today?
3. Are there any impediments in your way?

Question: Why? And how can you simulate that in your teams?