ay_of_the_web_tester) (/about)

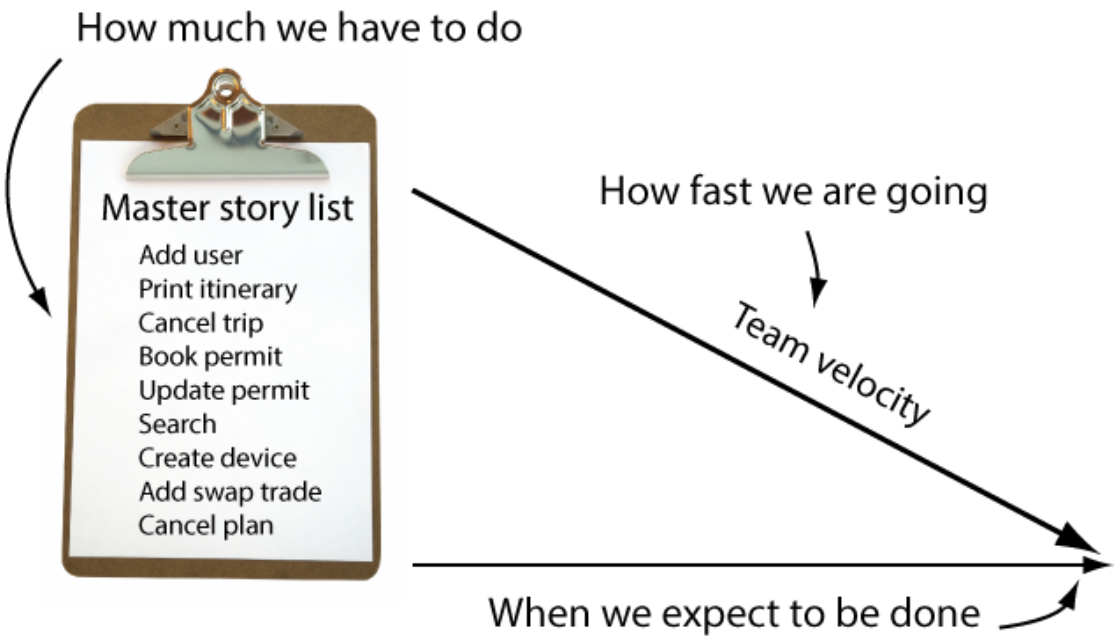
Getting Started (/) Videos (/videos) Courses (/bootcamp)

Learn More

Planning

The fine art of expectation setting

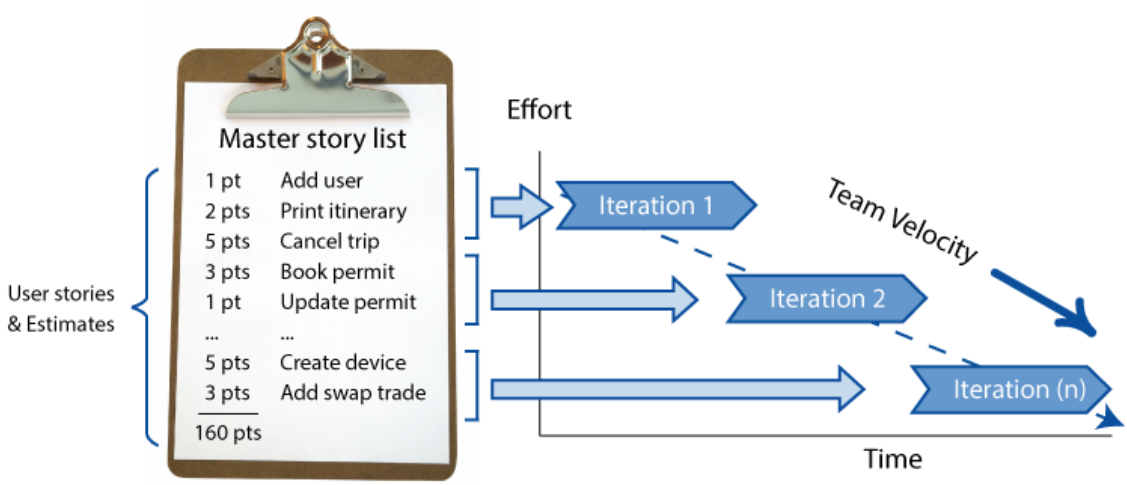
In its simplest form, agile planning is nothing more than measuring the speed a team can turn user stories into working, production-ready software and then using that to figure out when they'll be done.



Our to-do list on an agile project is called the master story list. It contains a list of all the features our customers would like to see in their software.

The speed at which we turn user stories into working software is called the team velocity. It's what we use for measuring our team's productivity and for setting expectations about delivery dates in the future.

The engine for getting things done is the agile iteration - one to two week sprints of work where we turn user stories into working, production-ready software.



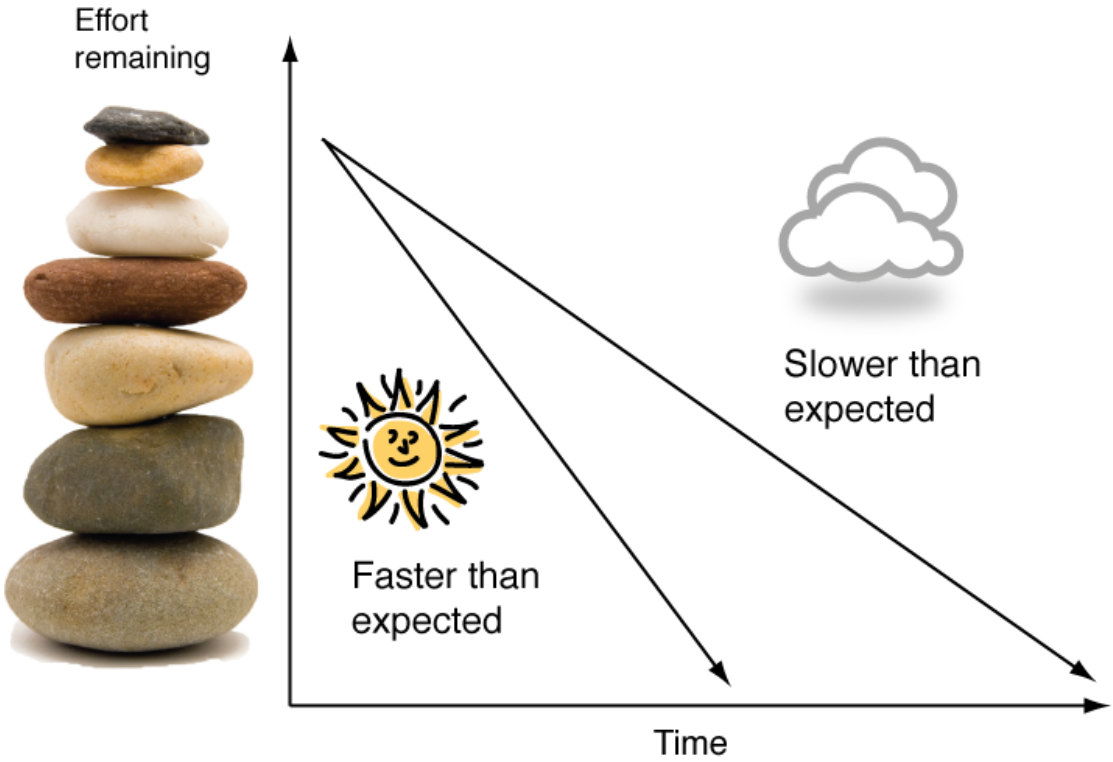
To give us a rough idea about delivery dates, we take the total effort for the project, divide it by our estimated team velocity, and calculate how many iterations we think we'll require to deliver our project. This becomes our project plan.

iterations = total effort / estimated team velocity

For example:

iterations = 100 pts / 10 pts per iteration = 10 iterations

Now, as we start delivering, one of two things is going to happen. We are going to discover that a) we are going faster than expected or b) we are going slower than we originally thought.



Faster than expected means you and your team are ahead of schedule. Slower than expected (more the norm) means you have too much to do and not enough time.

When faced with too much to do, agile teams will do less (kind of like what you and I do when faced with a really busy weekend). They will keep the most important stories, and drop the least important. This is called *adaptive planning* and it's how Agile teams work within their budgets and keep their projects real.

Watch this video on estimation (http://www.agilenutshell.com/episodes/3-estimation) to learn more about Agile Planning.

← Previous (/iterations)

Next → (/unit_testing)

IN A NUTSHELL

What is Agile? (/what_is_agile)

How does it work? (/how_does_it_work)

How is it different? (/how_is_it_different)

Agile Myths (/agile_myths)

Agile vs Waterfall (/agile_vs_waterfall)

FUNDAMENTALS

User Stories (/user_stories)

Estimation (/estimation)

Iterations (/iterations)

Planning (/planning)

ENGINEERING

Unit Testing (/unit_testing)

Refactoring (/refactoring)

Continuous Integration (/continuous_integration)

Test Driven Development (/test_driven_development)

CONCEPTS

Burndown Charts (/burndown)

Cone of Uncertainty (/cone_of_uncertainty)

Management by Miracle (/management_by_miracle)

Three Simple Truths (/three_simple_truths)

XPISMS

Bill of Rights (/bill_of_rights)

YAGNI (/yagni)

Yesterday's Weather (/yesterdays_weather)

The Simplest Thing (/simplest_thing)

Testing (/could_possibly_break)

Production Readiness (/production_readiness)

© Jonathan Rasmusson - Privacy Policy (/privacy)