ay_of_the_web_tester) (/about)

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

Learn More ▾

How does it work?

At its core, Agile does the same thing you and I do when faced with too much to do and not enough time.

You make a list

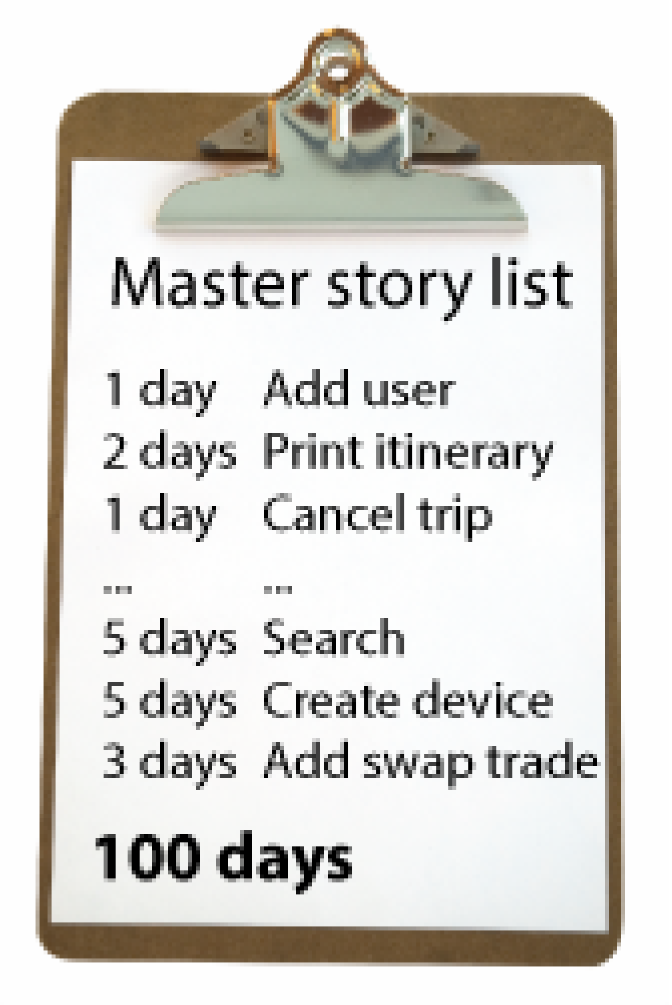
Sitting down with your customer you make a list of features they would like to see in their software. We call these things user stories (/user_stories) and they become the To Do list for your project.

Master story list

Add user
Print itinerary
Cancel trip
...
Search
Create device
Add swap trade

You size things up

Then, using Agile estimation (/estimation) techniques, you size your stories relatively to each other, coming up with a guess as to how long you think each user story will take.



Master story list

1 day	Add user
2 days	Print itinerary
1 day	Cancel trip
...	...
5 days	Search
5 days	Create device
3 days	Add swap trade

100 days

You set some priorities

Like most lists, there always seems to be more to do than time allows. So you ask your customer to prioritize their list so you get the most important stuff done first, and save the least important for last.

Master story list

1 day	Add user
1 day	Cancel trip
2 days	Print itinerary
...	...
5 days	Search
3 days	Add swap trade
5 days	Create device

You start executing

Then you start delivering some value. You start at the top. Work your way to the bottom. Building, iterating, and getting feedback from your customer as you go.

Master story list

1 day	Add user
1 day	Cancel trip
2 days	Print itinerary
...	...
5 days	Search
3 days	Add swap trade
5 days	Create device

100 days

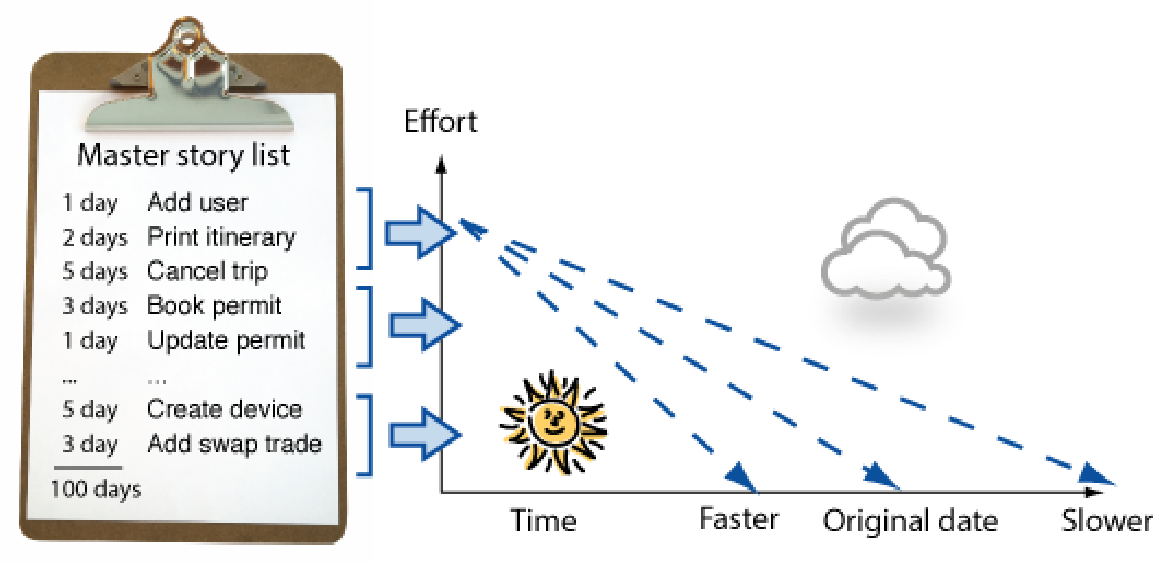
You update the plan as you go.

Then, as you and your customer starting delivering, one of two things is going to happen. You'll discover:

a. You're going fast enough. All is good. Or,

b. You have too much to do and not enough time.

At this point you have two choices. You can either a) do less and cut scope (recommended). Or you can b) push out the date and ask for more money.



← Previous (/what_is_agile)

Next → (/how_is_it_different)

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)