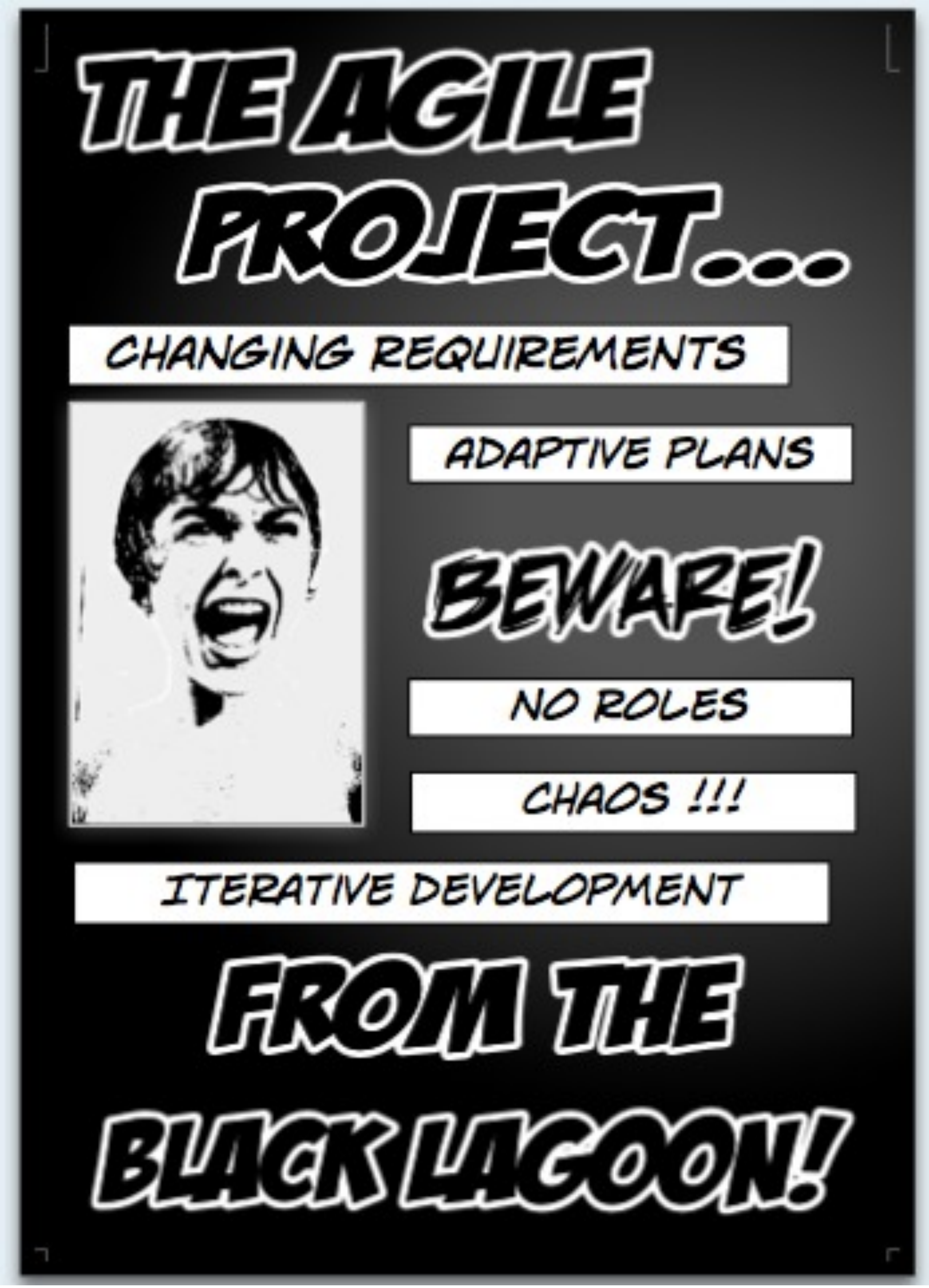
Agile In a Nutshell

Jonathan Rasmusson



THE AGILE PROJECT...

CHANGING REQUIREMENTS

ADAPTIVE PLANS

BEWARE!

NO ROLES

CHAOS !!!

ITERATIVE DEVELOPMENT

FROM THE BLACK LAGOON!

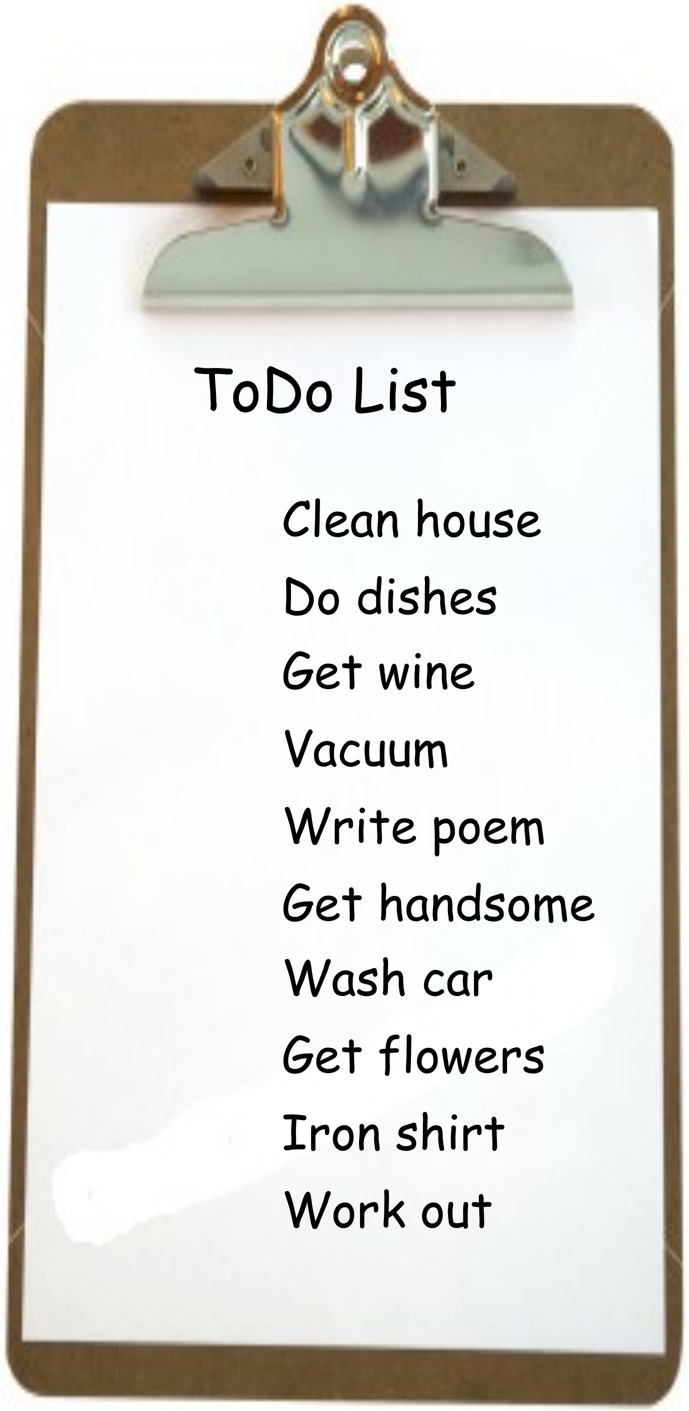
What we're going to cover

- How agile works
- What to expect
- What agile is
- Agile myths
- Review most popular methods
- Three steps towards agility

Too much to do, not enough time

Credit: http://www.flickr.com/photos/geneoh/

You make a list

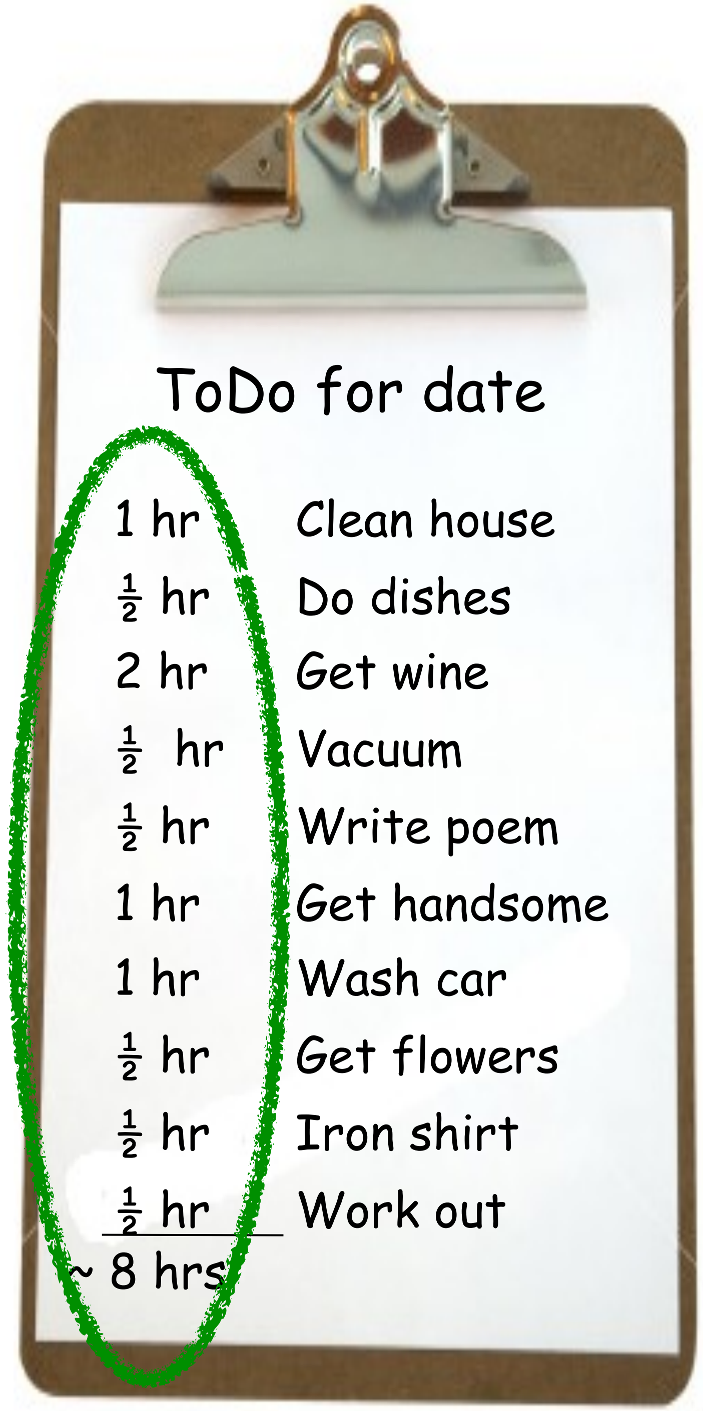


ToDo List

- Clean house
- Do dishes
- Get wine
- Vacuum
- Write poem
- Get handsome
- Wash car
- Get flowers
- Iron shirt
- Work out

This alone makes you feel good

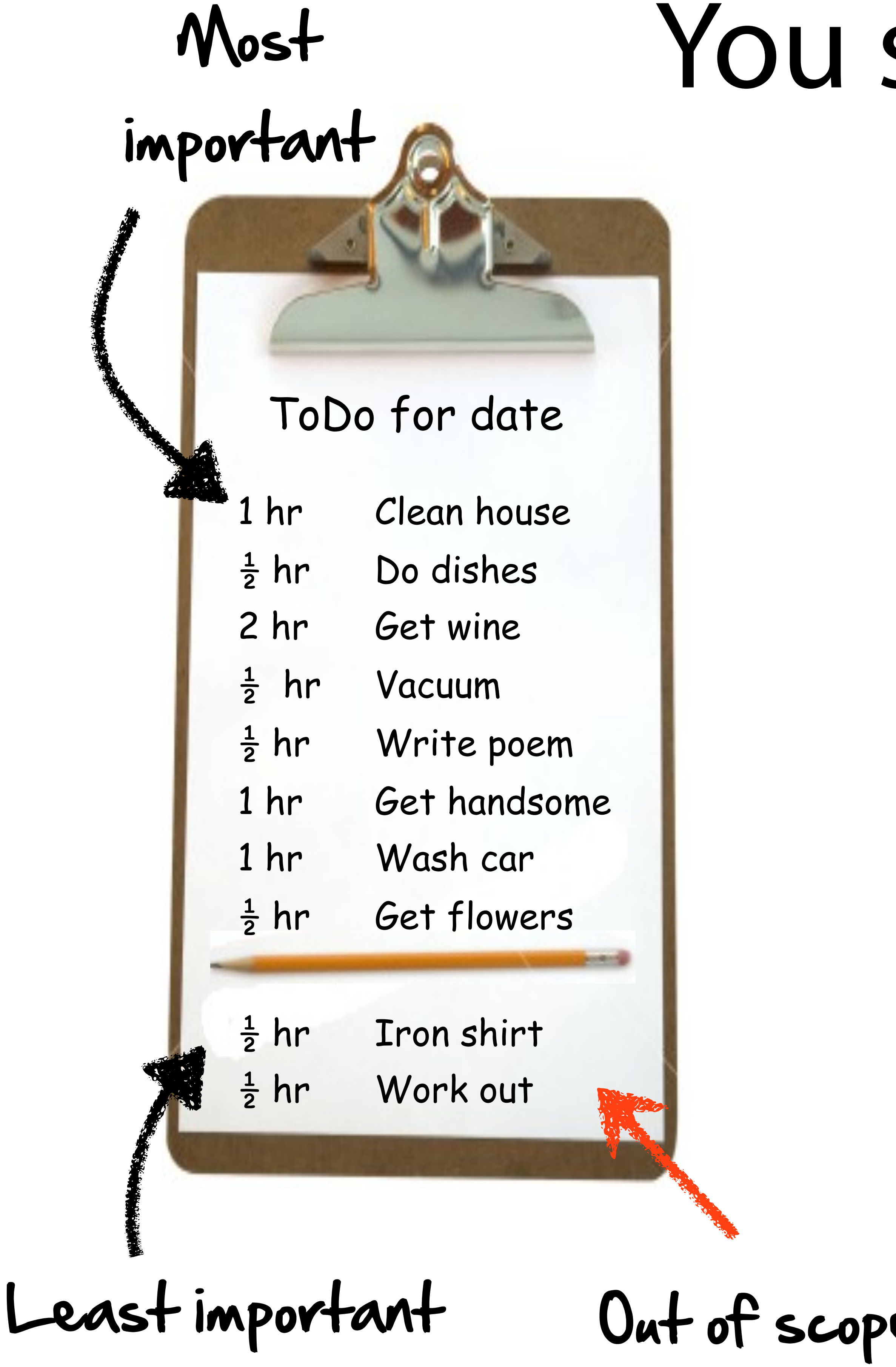
You size things up



ToDo for date

1 hr	Clean house
½ hr	Do dishes
2 hr	Get wine
½ hr	Vacuum
½ hr	Write poem
1 hr	Get handsome
1 hr	Wash car
½ hr	Get flowers
½ hr	Iron shirt
½ hr	Work out
~ 8 hrs	

You set some priorities



Most important

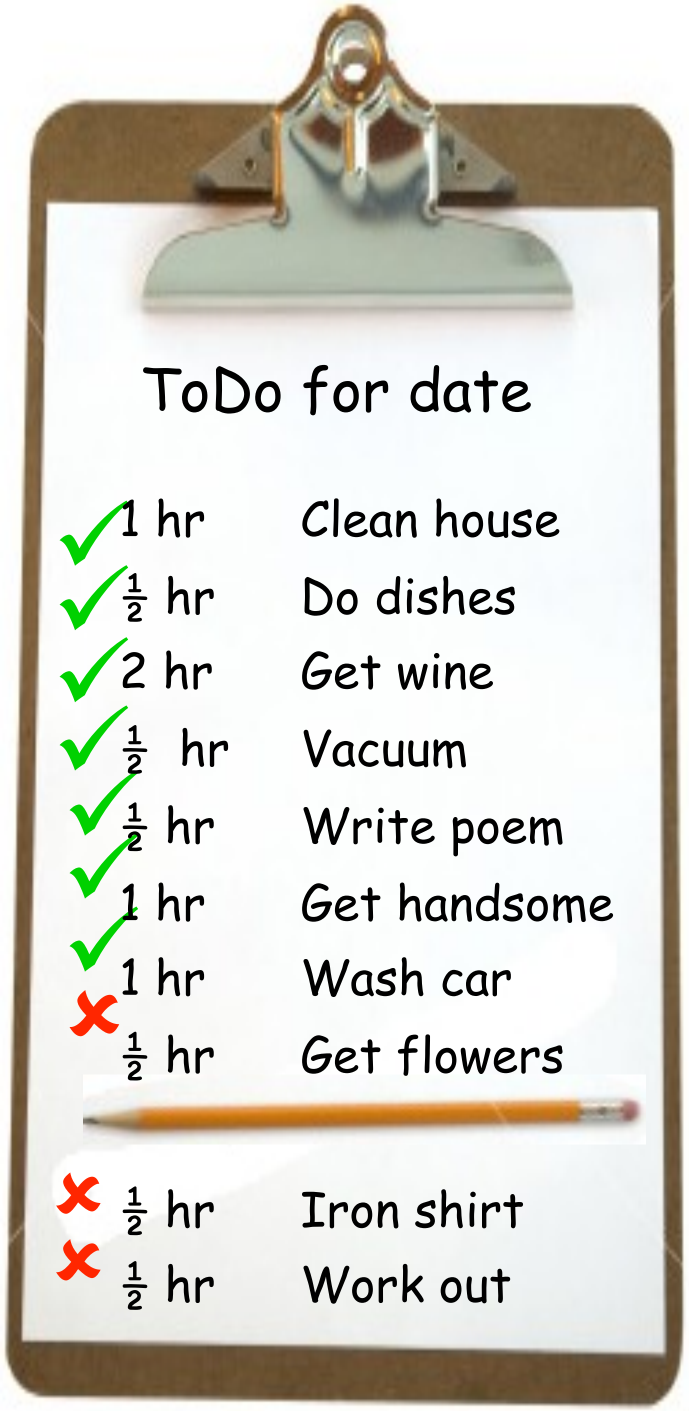
ToDo for date

1 hr	Clean house
½ hr	Do dishes
2 hr	Get wine
½ hr	Vacuum
½ hr	Write poem
1 hr	Get handsome
1 hr	Wash car
½ hr	Get flowers
½ hr	Iron shirt
½ hr	Work out

Least important

Out of scope

Start executing



ToDo for date

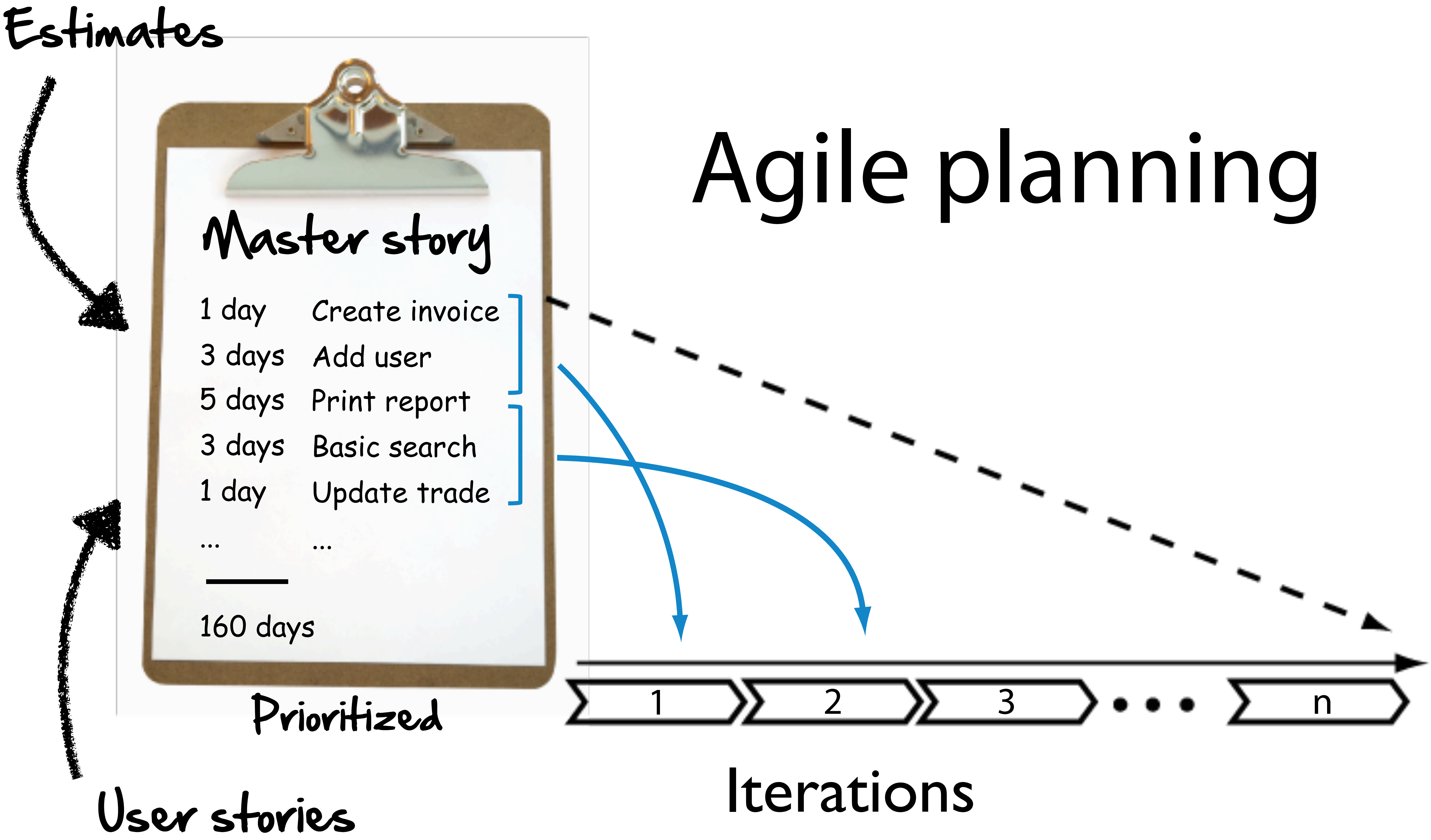
✓	1 hr	Clean house
✓	½ hr	Do dishes
✓	2 hr	Get wine
✓	½ hr	Vacuum
✓	½ hr	Write poem
✓	1 hr	Get handsome
✓	1 hr	Wash car
✗	½ hr	Get flowers
✗	½ hr	Iron shirt
✗	½ hr	Work out

This is what most people do ...

A little secret

Only instead of ...	We use fancy names
ToDo lists	Master story lists
Tasks	User stories
Guesses	Estimates

Agile planning



Estimates

Master story

1 day	Create invoice
3 days	Add user
5 days	Print report
3 days	Basic search
1 day	Update trade
...	...

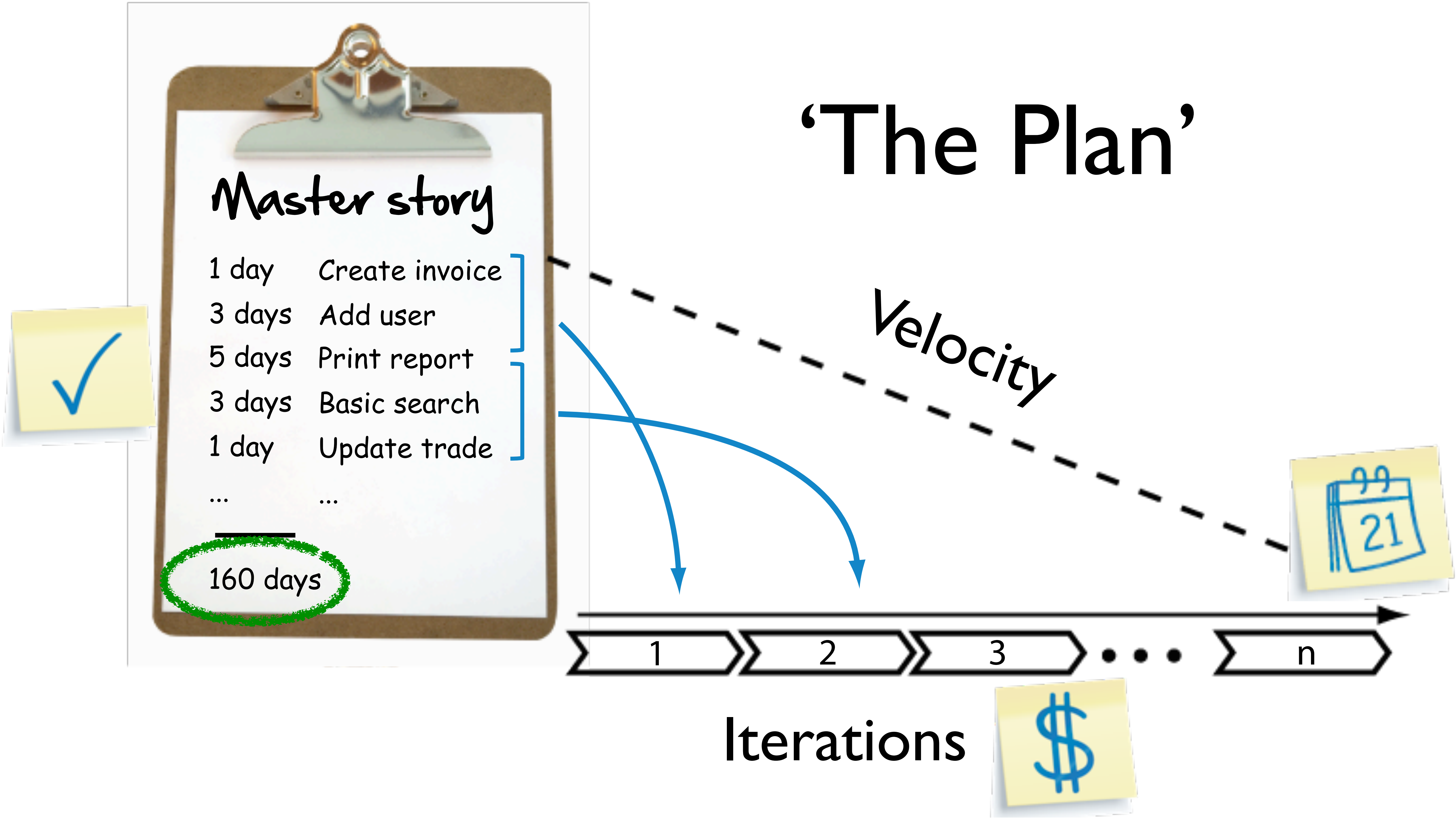
160 days

Prioritized

User stories

Iterations

‘The Plan’



Master story

1 day	Create invoice
3 days	Add user
5 days	Print report
3 days	Basic search
1 day	Update trade
...	...

160 days

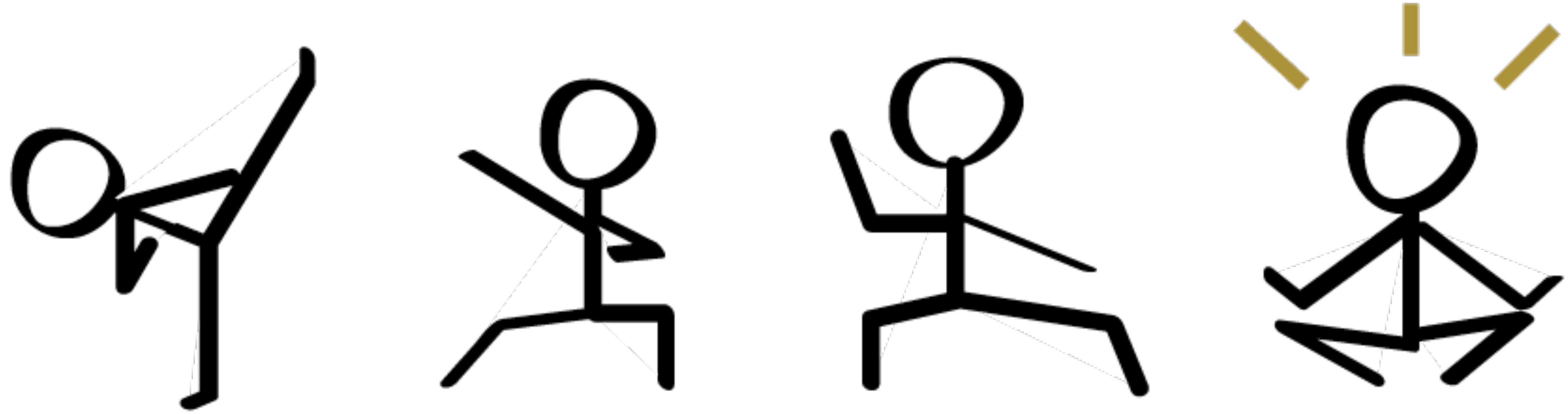
Velocity

Iterations

What we’re going to cover

- How it works
- What to expect
- What agile is
- Agile myths
- Review most popular methods
- Three steps towards agility

We flex on scope



Time Budget Quality Scope

We have the same definition of ‘done’

Credit: http://www.sxc.hu/photo/422930

A plan?

A design?

A report?

Working software

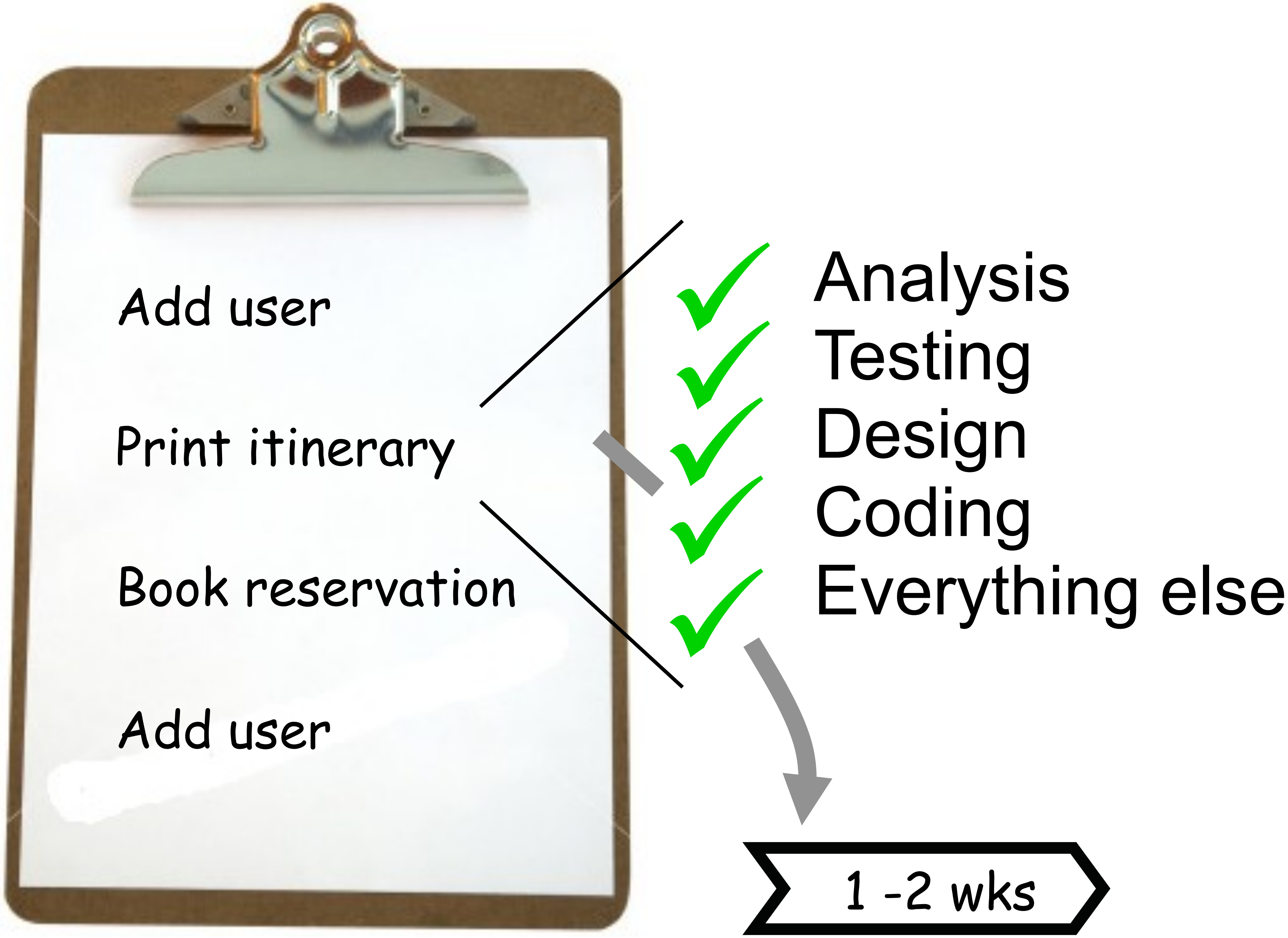
is the primary measure of success

Project plans
Test plans
Requirements docs
Architectural diagrams
Analysis models
Security reports
Deployment plans

… are of no value
to the customer.

Credit: http://www.flickr.com/photos/chris-hunt/3248785666/

That means …



Add user

Print itinerary

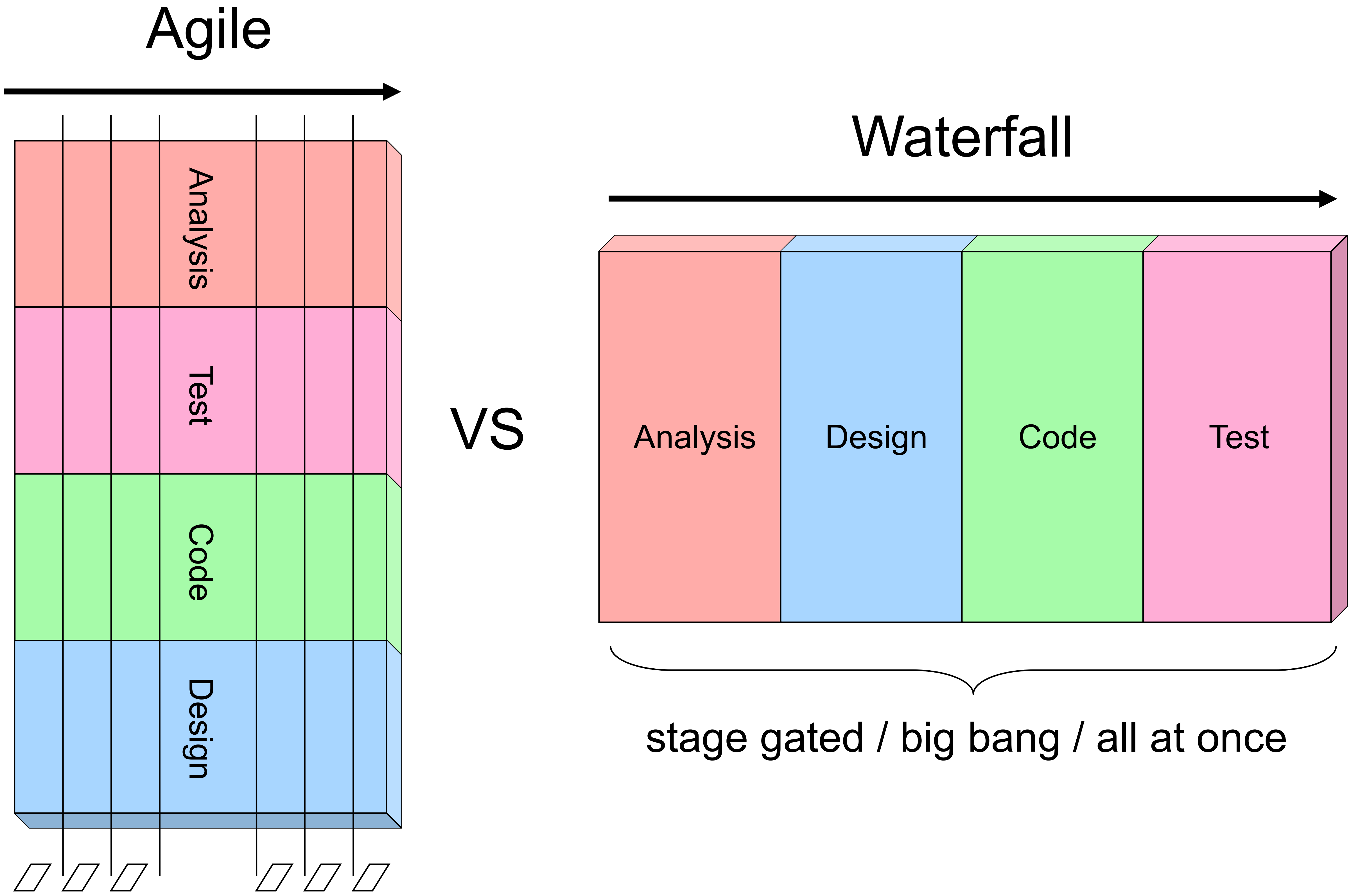
Book reservation

Add user

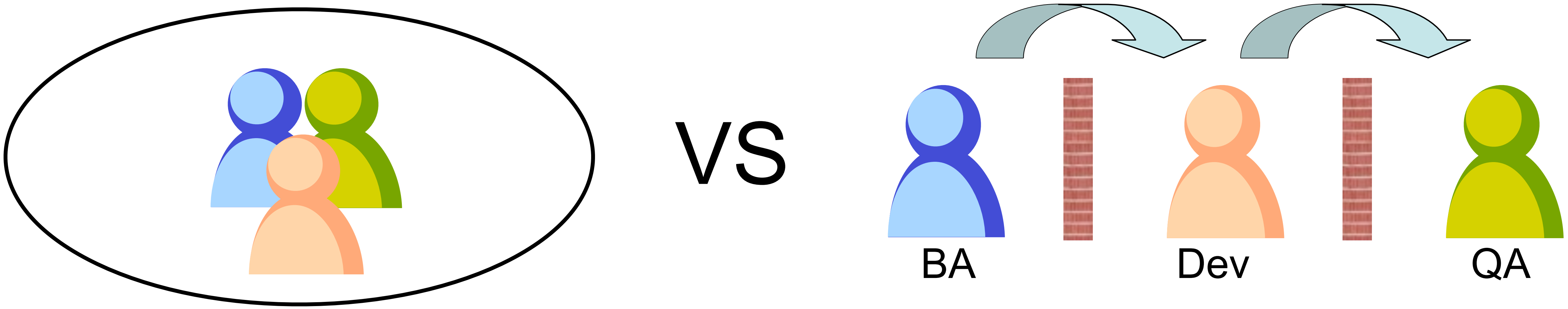
- ✓ Analysis
- ✓ Testing
- ✓ Design
- ✓ Coding
- ✓ Everything else

1 -2 wks

Analysis, design, testing, and coding are continuous activities



And we have to work as one team



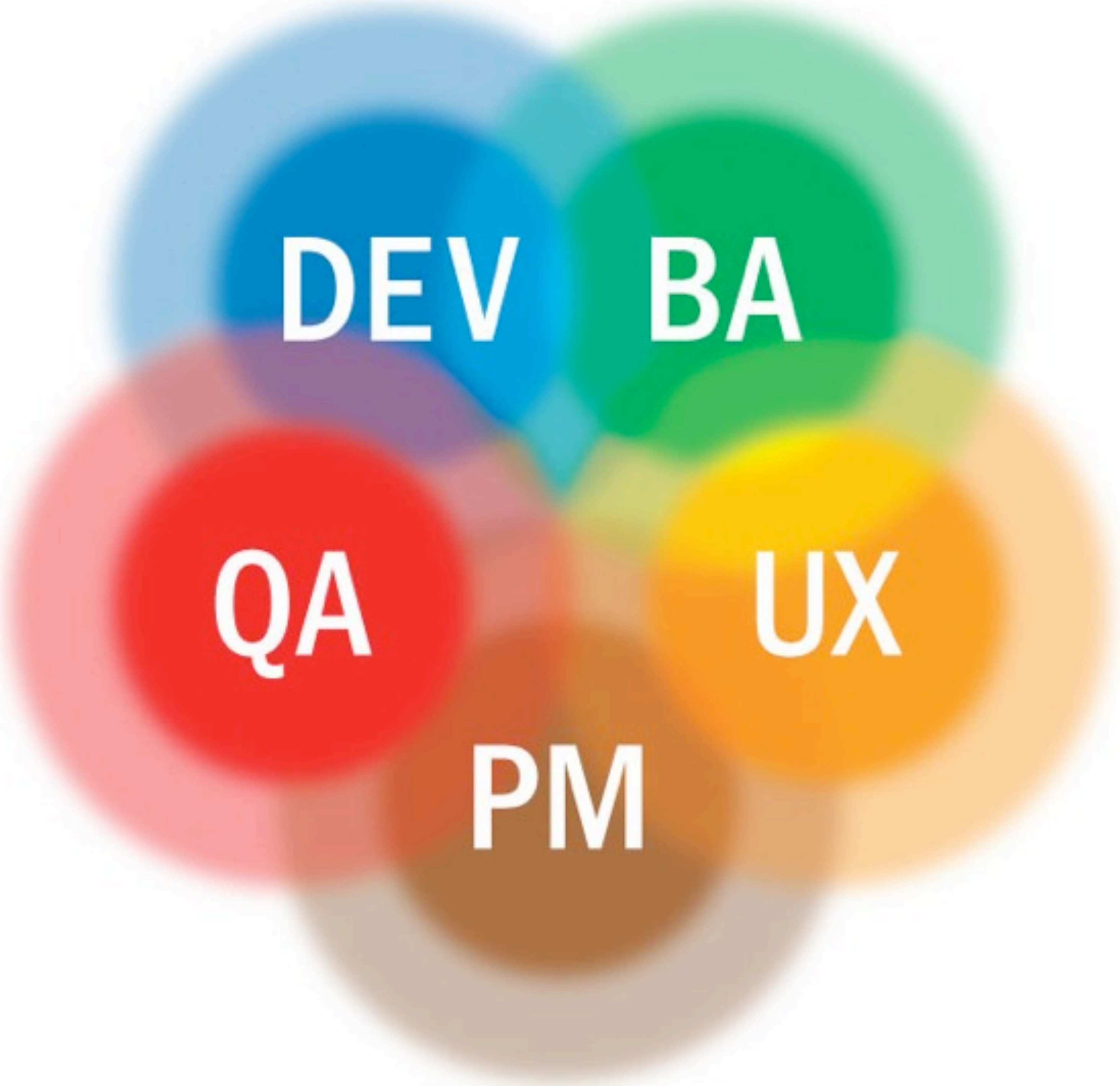
VS

BA

Dev

QA

More overlap between roles



What we’re going to cover

- How it works
- What to expect
- What agile is
- Agile myths
- Review most popular methods
- Three steps towards agility

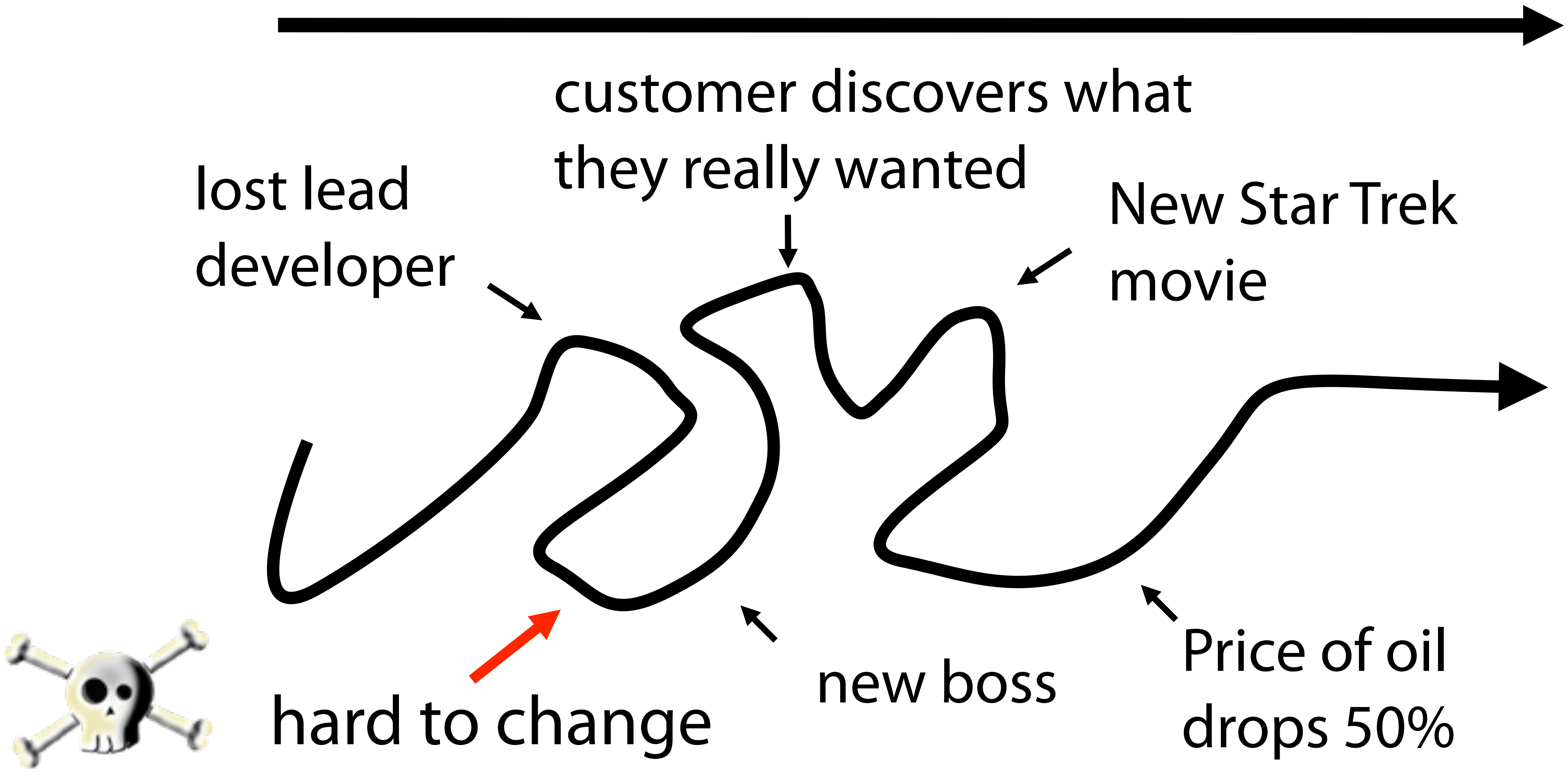
Iterative

Iterative

Adaptive

Adaptive

Adaptive



customer discovers what they really wanted

lost lead developer

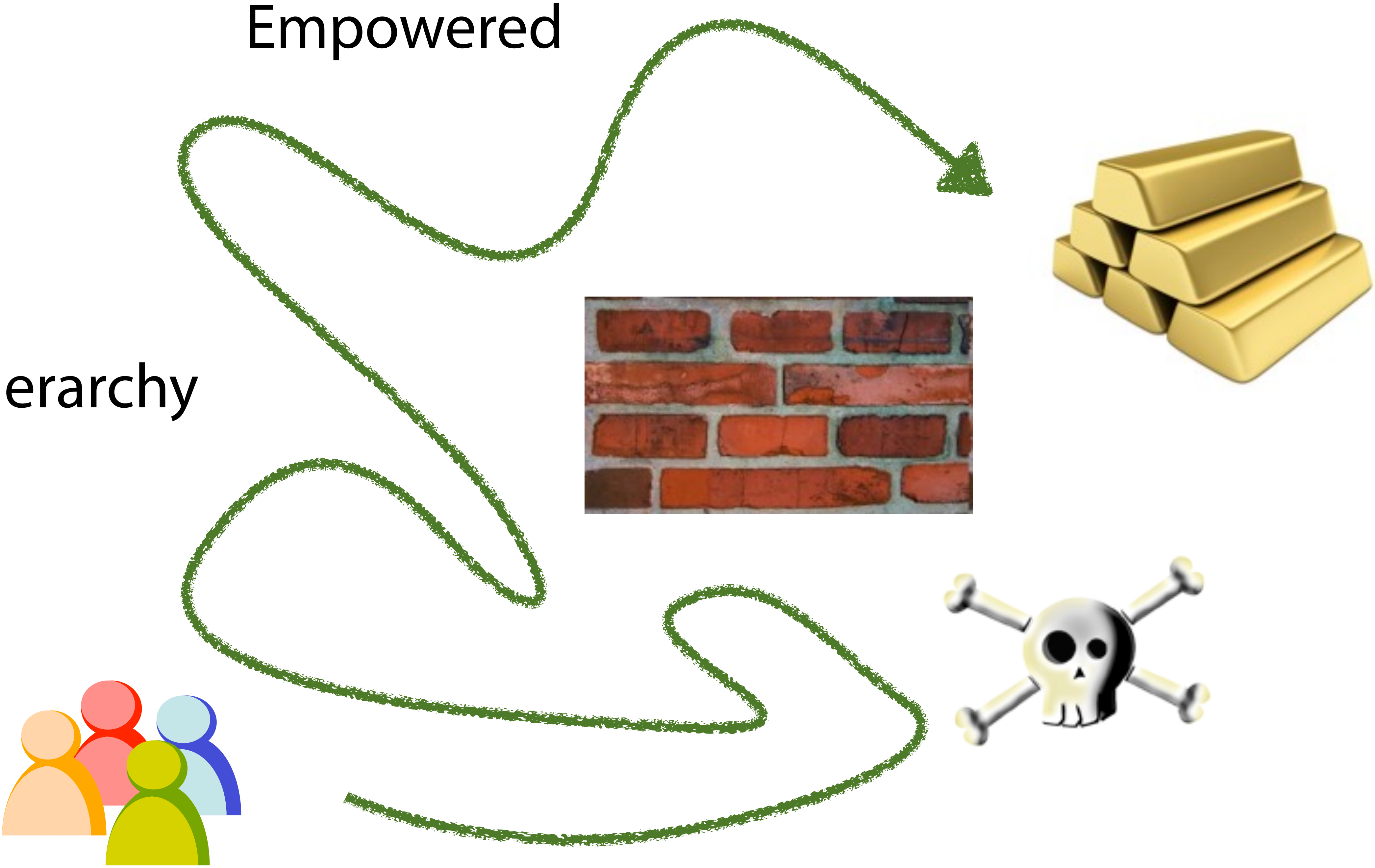
New Star Trek movie

hard to change

new boss

Price of oil drops 50%

Self directed



Empowered

Flat hierarchy

Adaptive

Self-organizing

Smart,

Talented,

Motivated People

Like working this way.

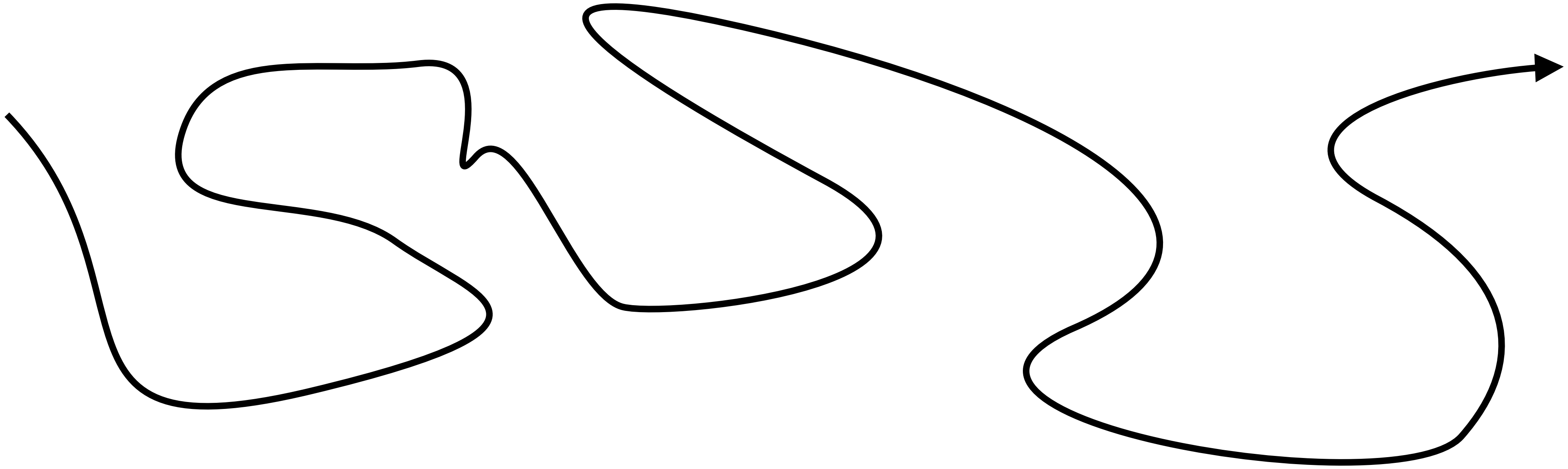
Agile is

A common sense way
of building software ...

That reminds us that ...

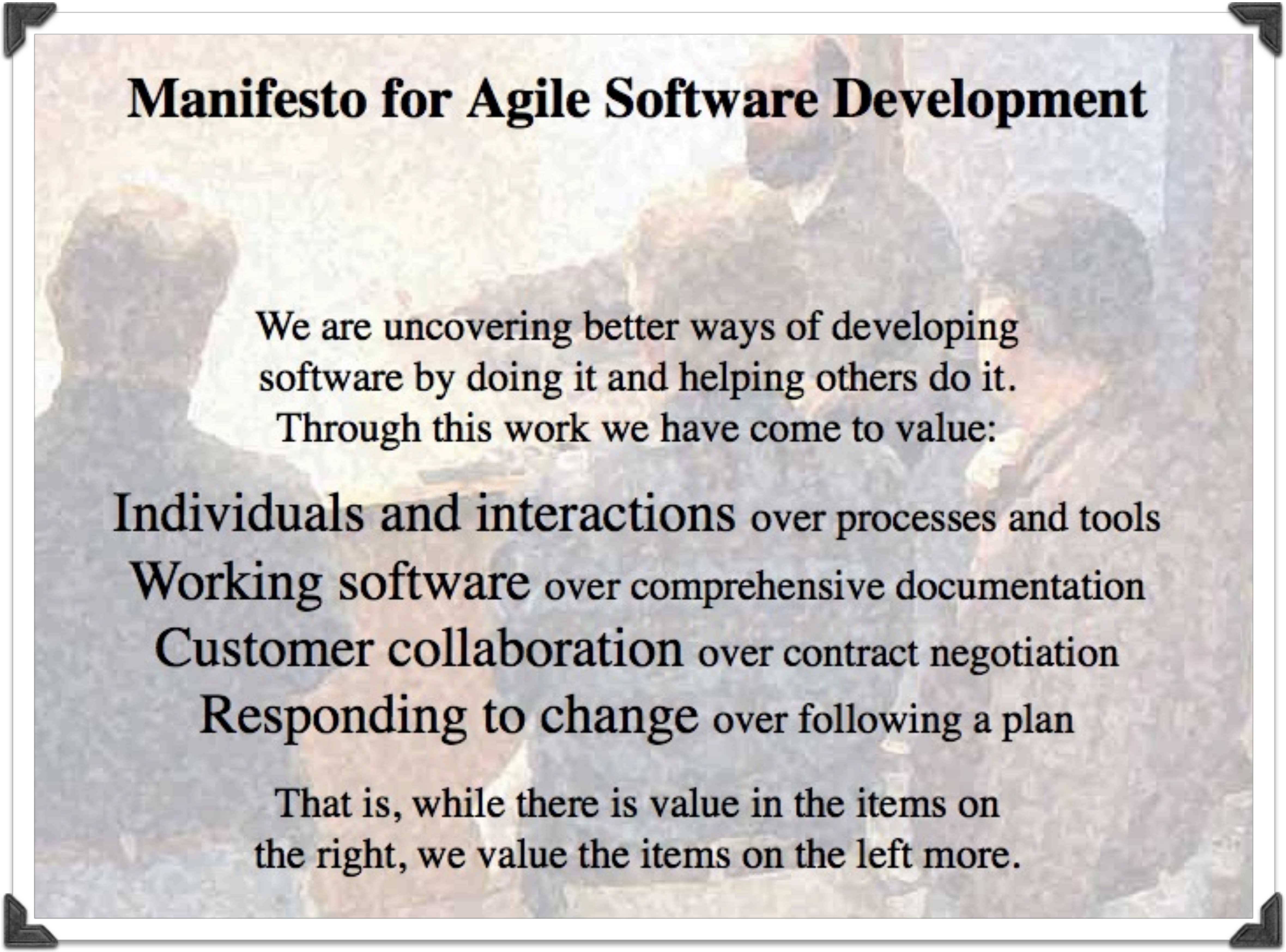
software is about people.

It’s messy.
It isn’t perfect.
It doesn’t follow a Gantt chart.



It accepts the **inherent complexity** and **uncertainty** that comes with software delivery.

It accepts the **inherent complexity** and **uncertainty** that comes with software delivery.



Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.

12 Principles

Face-to-face

Continuous delivery

Working together

Simplicity

Self organization

Working software

Sustainable pace

Motivated individuals

Welcoming change

Technical excellence

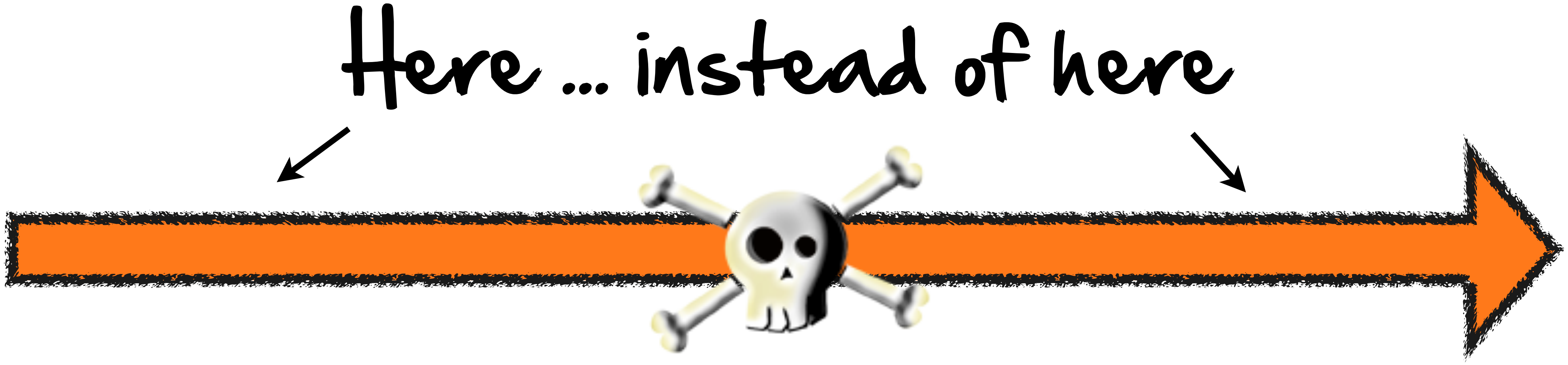
Frequent delivery

Continuous improvement

Agile myths

Agile is a silver bullet

- You can fail just as spectacularly on an agile project as any other.
- You just might do it sooner.



Here ... instead of here

Agile teams don't do documentation

- More accurate to say agile teams don't write any 'unnecessary documentation'.
- Treat documentation like any other deliverable:
  - Estimated and prioritized.
- Prefer face-to-face communication.

Agile is anti-planning

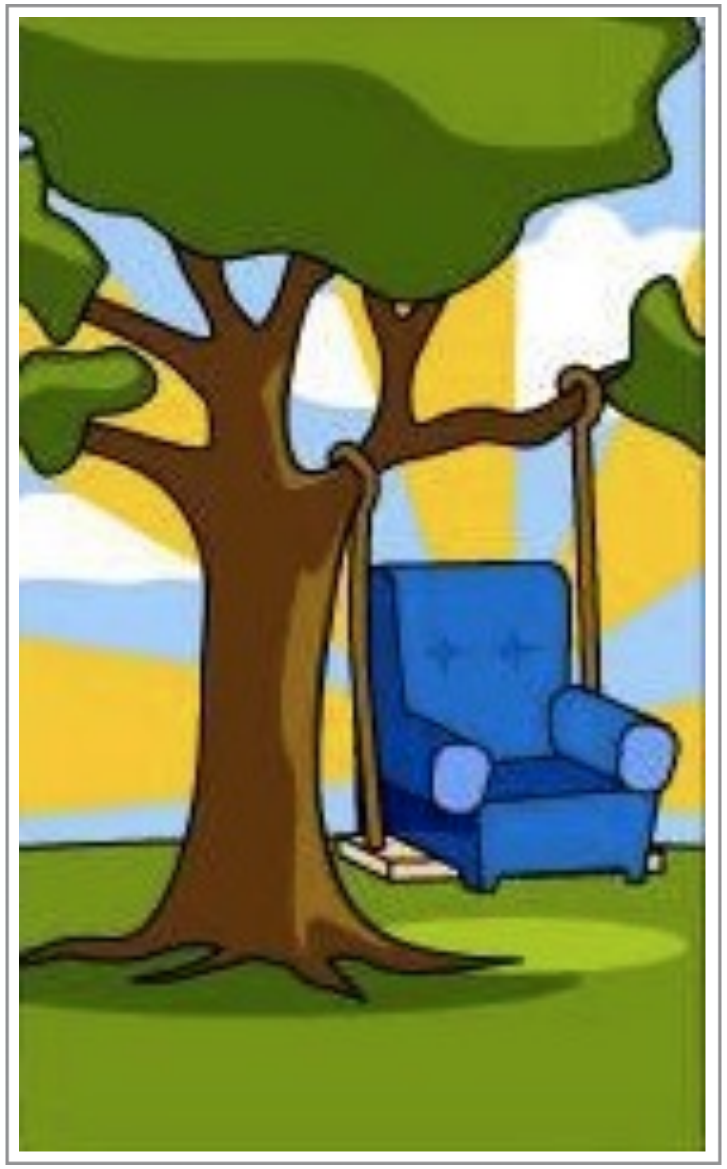
- Agile teams plan extensively
  - –Every quarter (release)
  - –Every couple weeks (iterations)
  - –Every day (daily stand-ups)
- Uses different tools
  - –Burn downs vs Gantt charts
- Planning is very visible
  - –Stakeholders know early if there is a problem

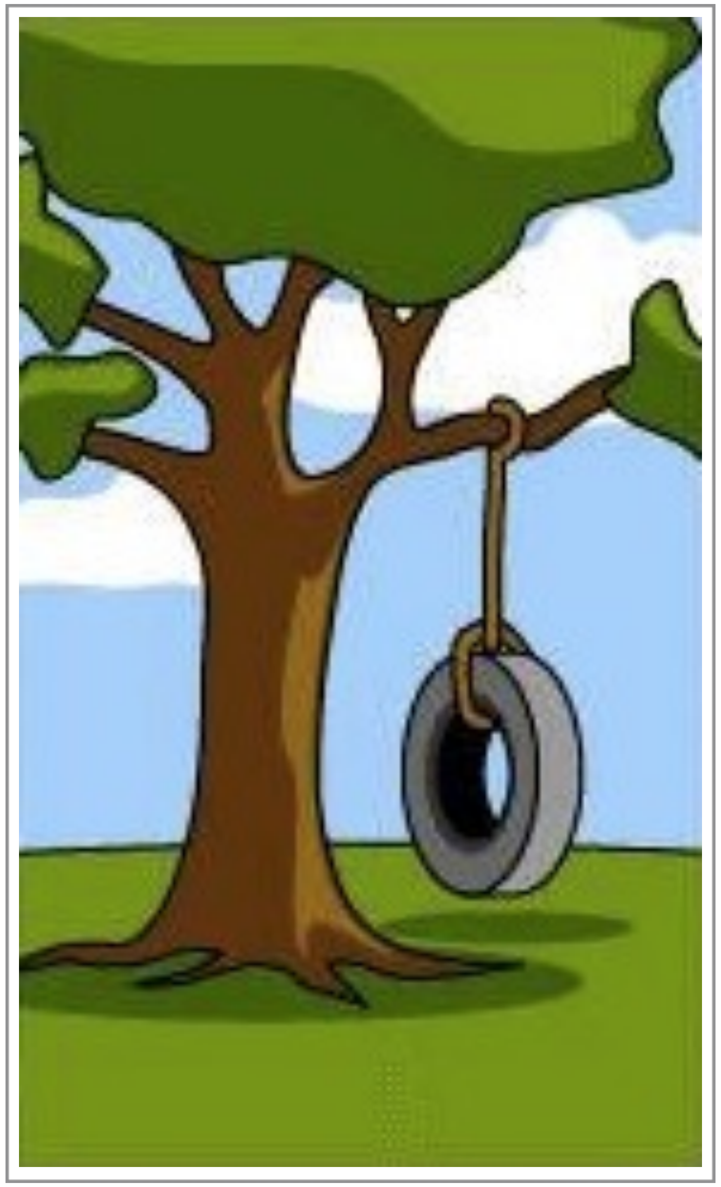
Agile is undisciplined

- Truth is agile is very disciplined
  - You have to test.
  - You have to get feedback.
  - You have to regularly ship.
  - You have to update the plan.

Agile is anti-architecture

Don’t build this ... if all you need is this.





Agile doesn't scale

- Agile scales like any other software process

– Not that great –

- Instead of looking at how to scale up, try to imagine ways you could scale down.

What we're going to cover

- How it works
- What to expect
- What agile is
- Agile myths
- Review most popular methods
- Three steps towards agility

Agile comes in many flavours

Crystal

Scrum

Lean

DSDM

FDD

Extreme Programming (XP)

Lean Toyota’s ultra-lean manufacturing process.

I would like to buy a Toyota Camry please.

Eliminating waste

Lean

+

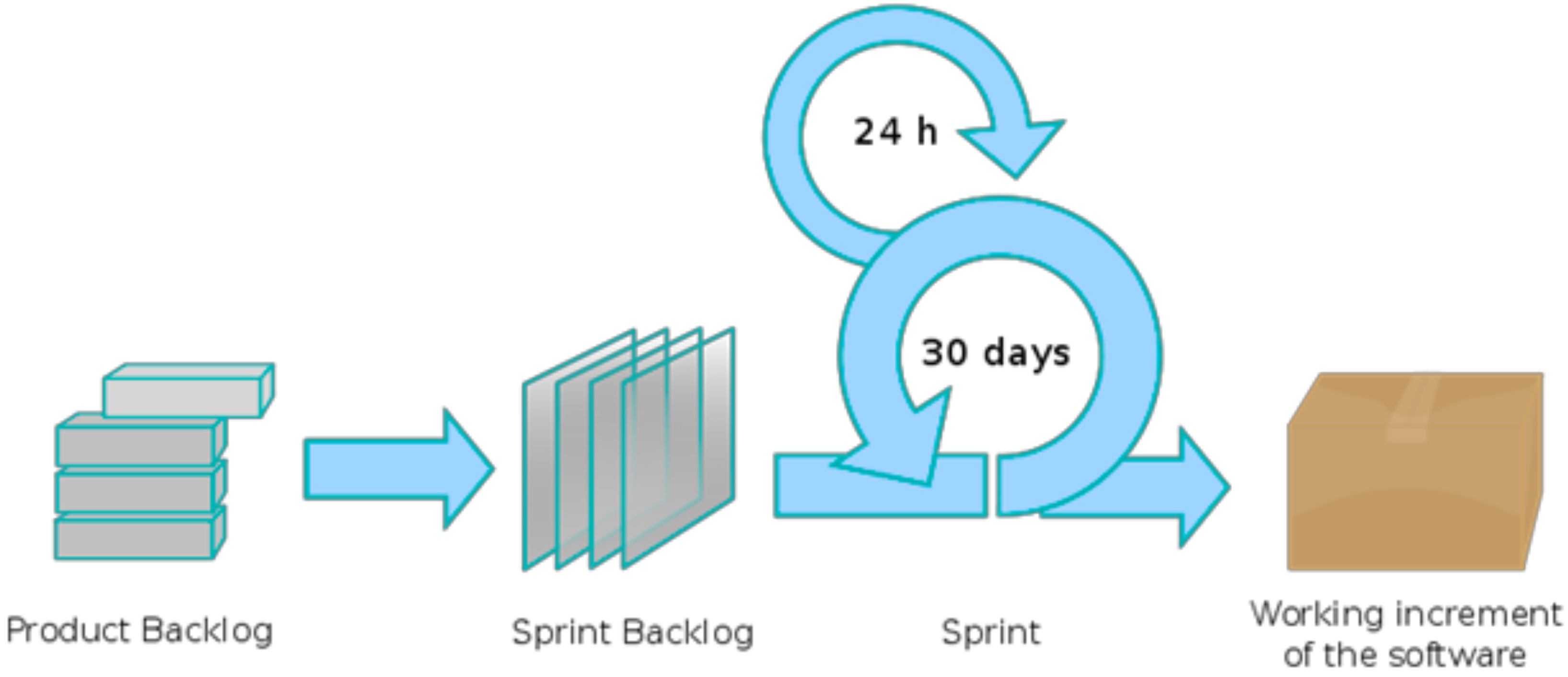
- Very good high-level principles and practices
- Addresses system and organizational inefficiencies

−

- Not IT specific
- Can be harder to implement

Lean has a lot to offer. Very worth studying!

Scrum



- A project management wrapper for incremental delivery of projects, independent of technology or business vertical.
- Can be used in non-IT projects.

Scrum

➕

- Easy to understand /pickup
- Low barrier of entry
- Speaks well to project managers
- Non-threatening
- Very popular

➖

- Silent on engineering
- Easy to do the easy stuff while skipping on the hard

Easy to adapt - most non-threatening.

Extreme Programming

- Popularized software engineering practices necessary for agile development
- Emphasizes
  - upfront testing
  - automation
  - evolutionary design
  - continuous integration

User story

Extreme Programming

➕

- Sound engineering practices
- Strong in development community
- First real popular agile method

➖

- Very developer focused
- Hard for other disciplines relate
- Sometimes characterized by zealots

Essential engineering practices

- unit testing - refactoring - continuous integration - test-Driven Design (TDD)

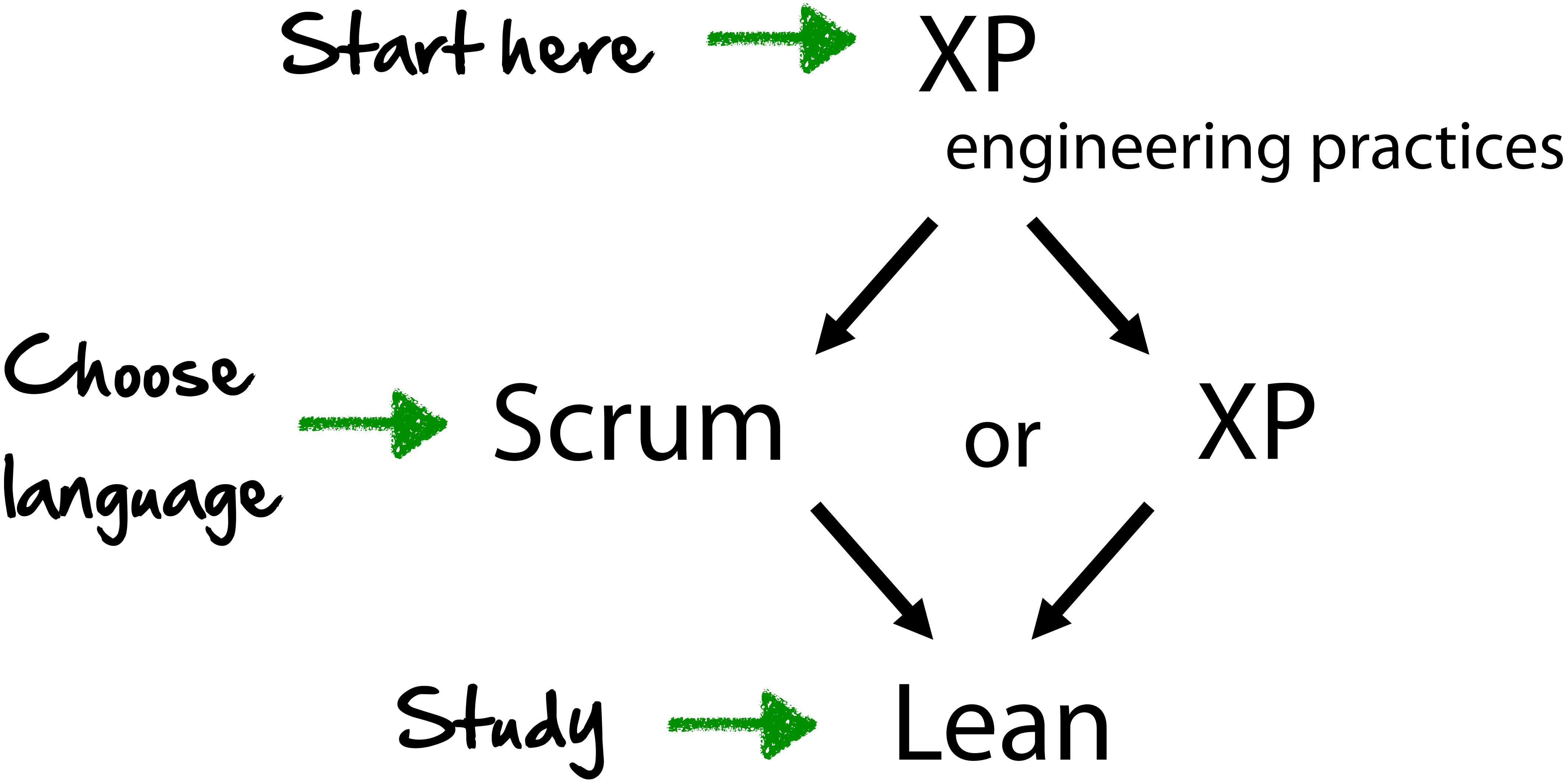
Options for adopting

XP
full on

Scrum + XP
engineering practices

Scrum + XP + Lean
engineering practices spirit

What I recommend



Start here → XP

engineering practices

Choose language → Scrum or XP

Study → Lean

There is no one way

Extreme Programming (XP)

Your own!

Scrum

DSDM

Lean

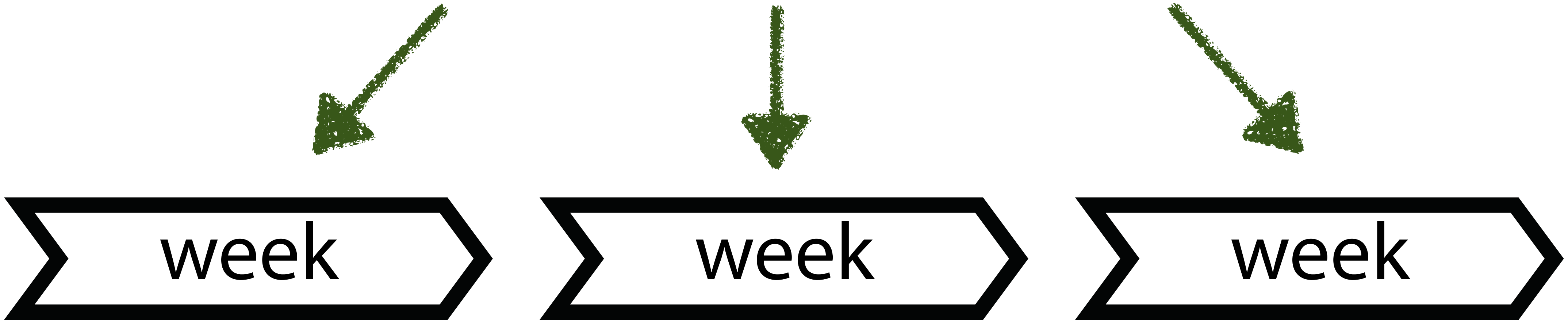
FDD

Do what ever works for you

What we're going to cover

- How it works
- What to expect
- What agile is
- Agile myths
- Review most popular methods
- Three steps towards agility

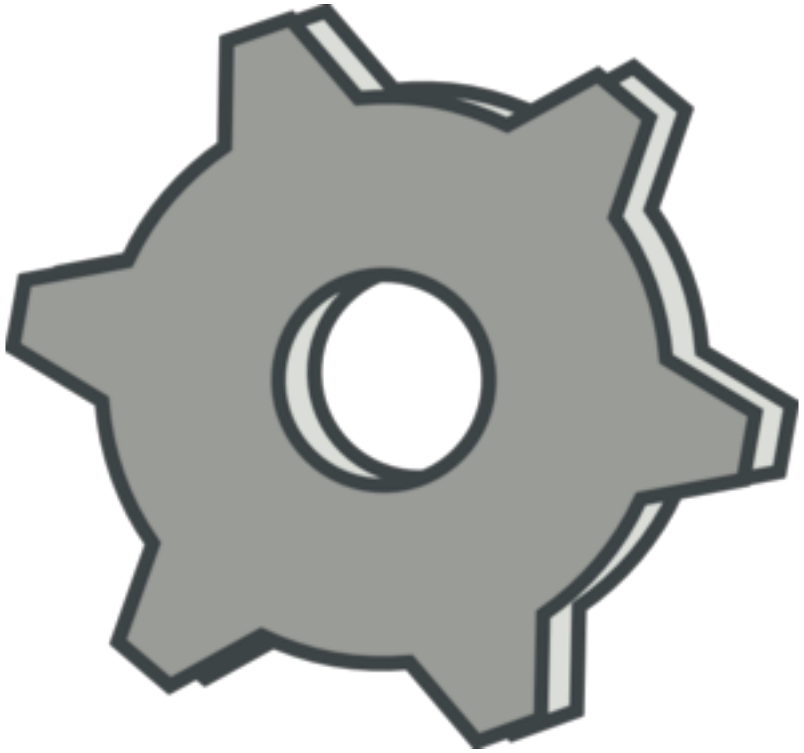
Step 1: Deliver something of value every week



1. Big problems down into smaller ones.
2. Important stuff first.
3. Forces you to test.
4. You’ll want feedback.
5. Change course when necessary.

Step 2: Start doing these 4 software engineering practices today!

1. Unit testing
2. Refactoring
3. Test Driven-Development
4. Continuous Integration



Accept three simple truths

1. It is impossible to gather all the requirements at the beginning of a project.
2. Whatever requirements you do gather are guaranteed to change.
3. There will always be more to do than time and money will allow.

Final words

Do I think everyone will one day be doing agile?

No

For the same reason people
don’t eat right and exercise.

Agile is hard

It’s tough!

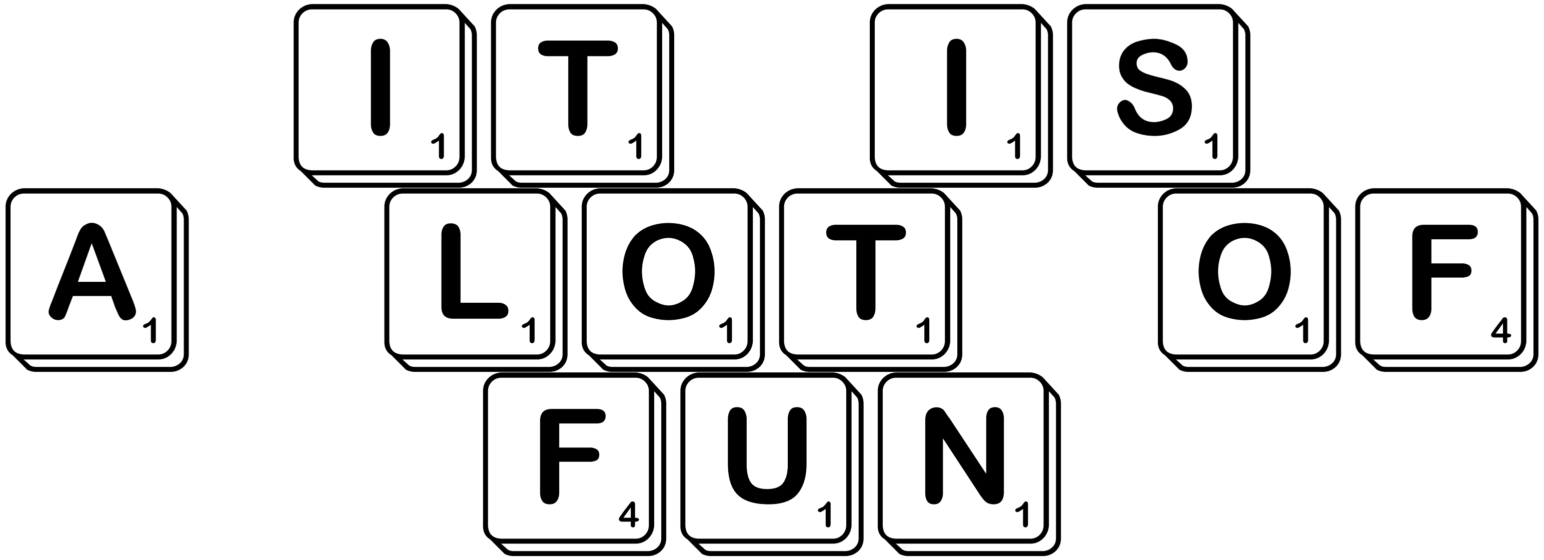
Requires a lot of discipline

And thinking

Not everyone’s into this kind of stuff.

But it is a very natural way to work

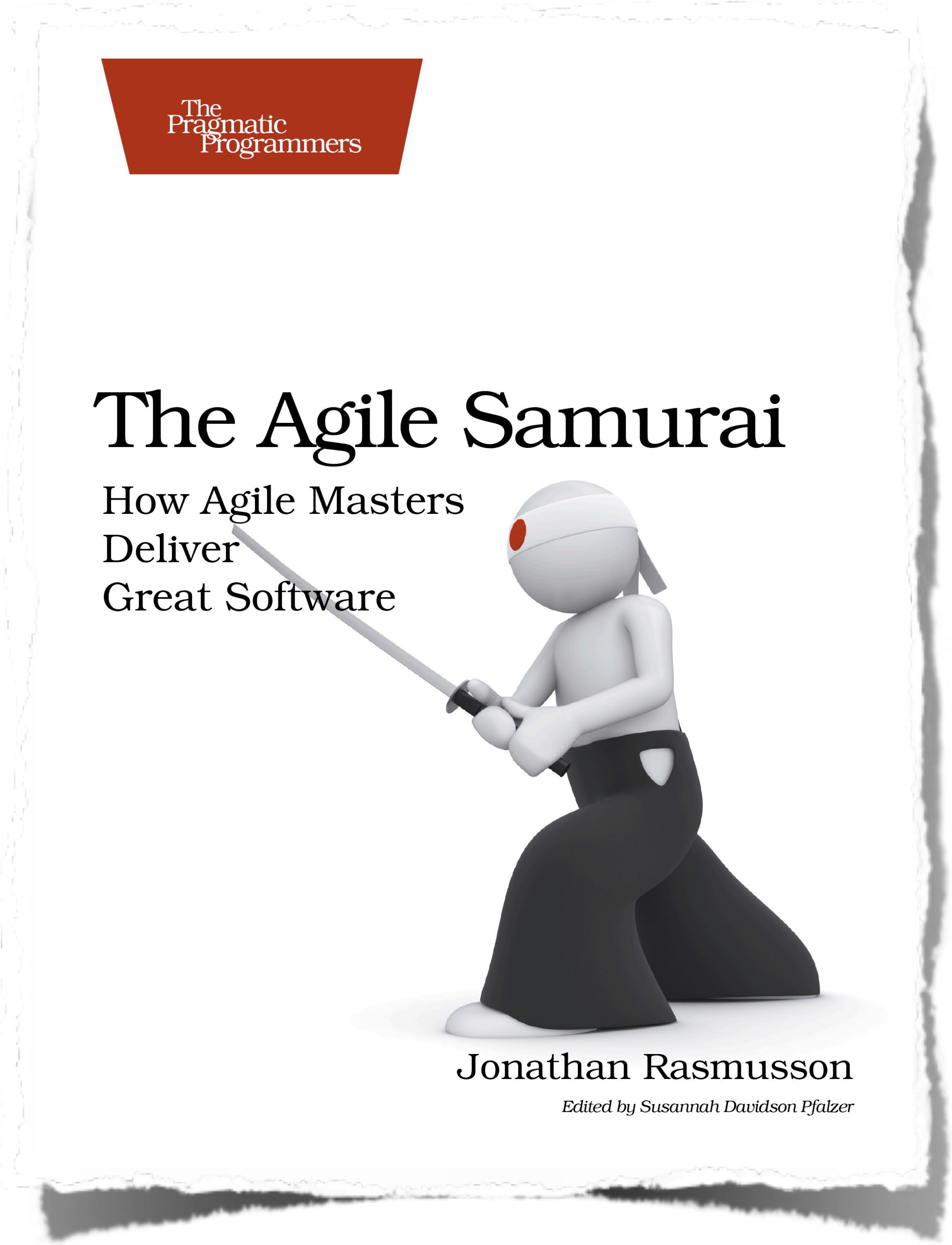
A lot of people enjoy it



Learn more

- Blog
  - agilewarrior.wordpress.com
- Twitter
  - @jrasmusson
- Email
  - jr@agilewarrior.com

40% off this week only
Code: AGILEWEEK



The Pragmatic Programmers

The Agile Samurai

How Agile Masters
Deliver
Great Software

Jonathan Rasmusson

Edited by Susannah Davidson Pfalzer