

Agile Samurai Principles

The Agile Samurai

How Agile Masters
Deliver
Great Software



Jonathan Rasmusson

Edited by Susannah Davidson Pfalzer

Agile Development

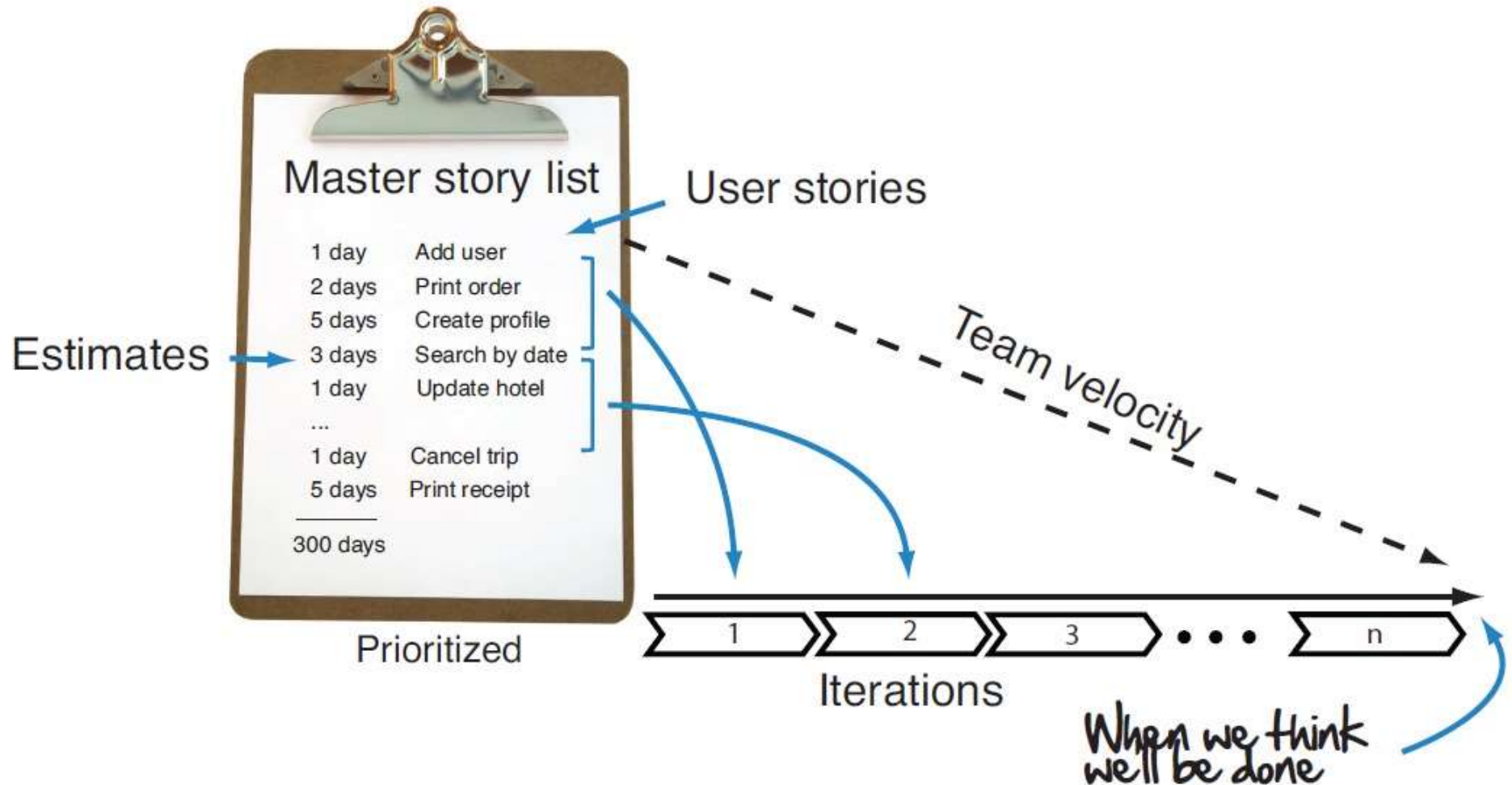
Three simple truths

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.

Deliver Value Every Iteration

- Break big problems into smaller ones
- Focus on most important issues
- Deliver something that works
- Lots of customer feedback
- Change course when necessary
- You are accountable

Agile Planning



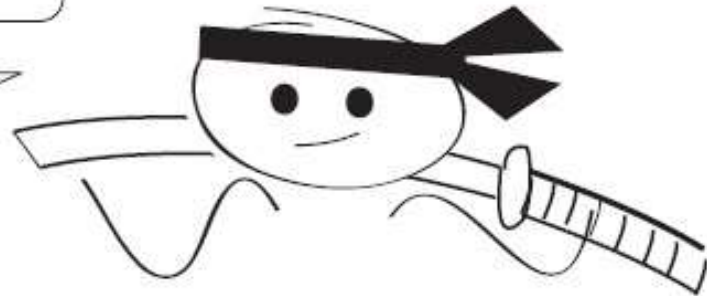
Agile principle

Working software is the primary measure of success.

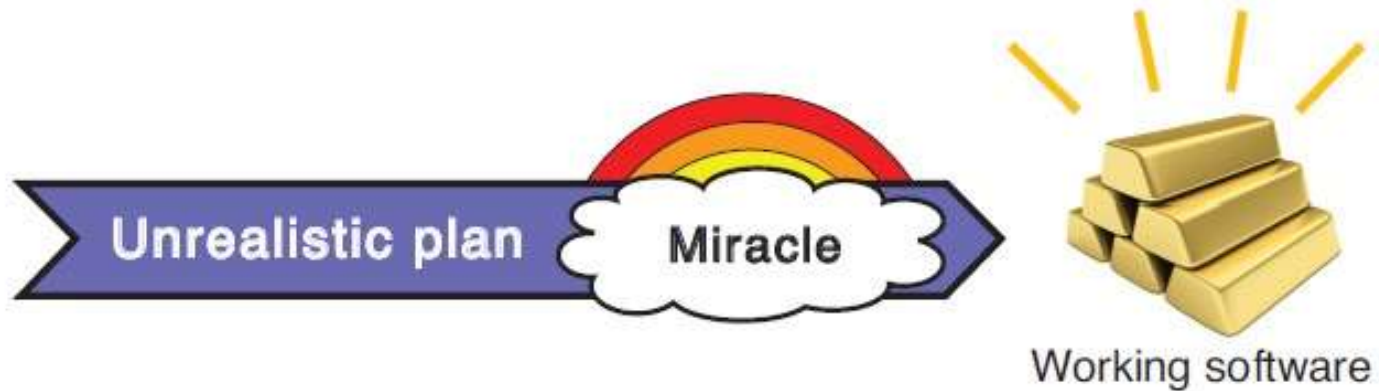
Agile Planning

*HOLD ON ... YOU ARE MAKING
THIS SOUND WAY TOO EASY.*

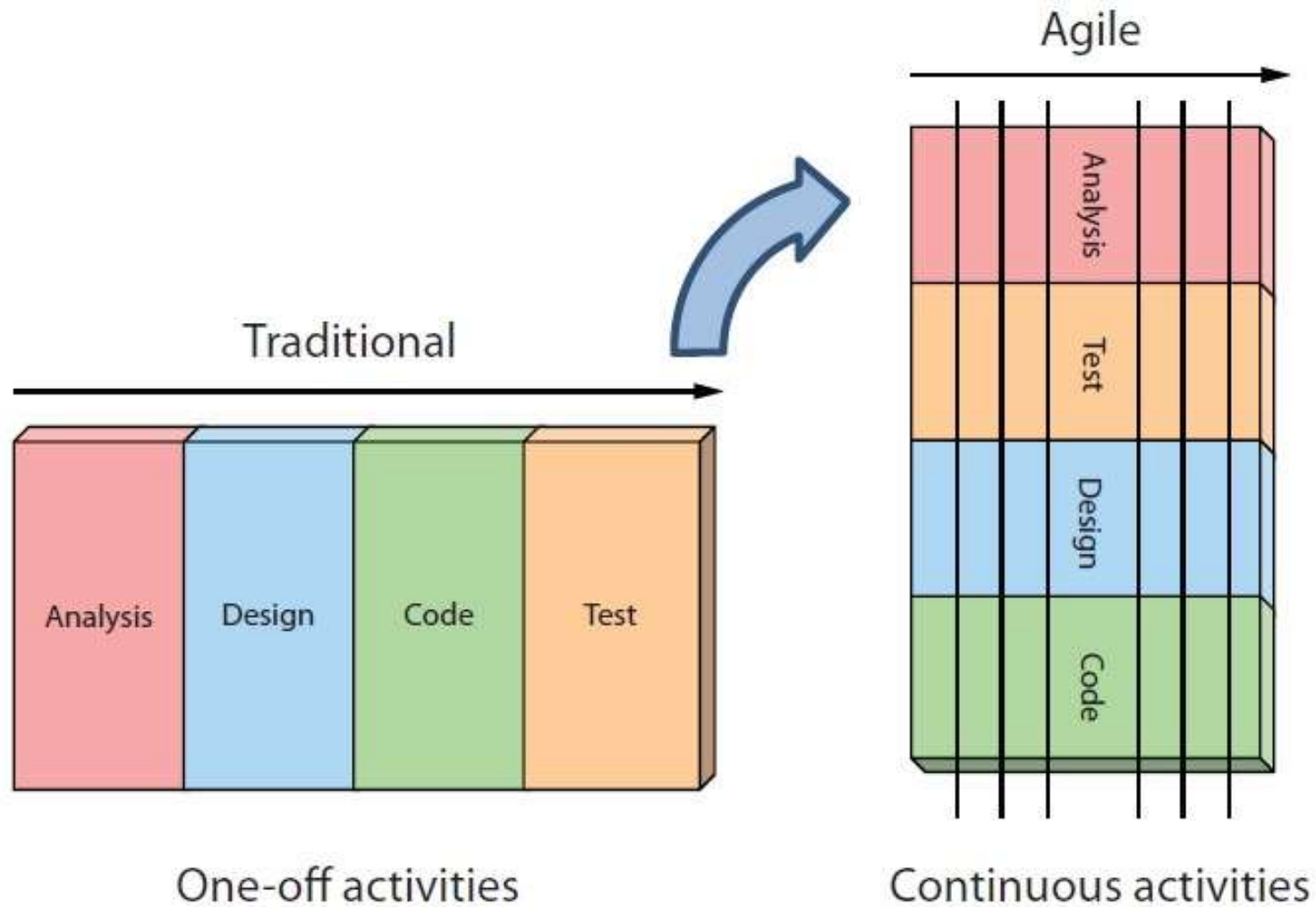
*WHAT IF IT'S A FIXED BID CONTRACT
AND EVERYTHING HAS TO BE DONE
OR WE ARE ALL GOING TO DIE ???*



Agile Planning



Agile Lifecycle



Agile Team

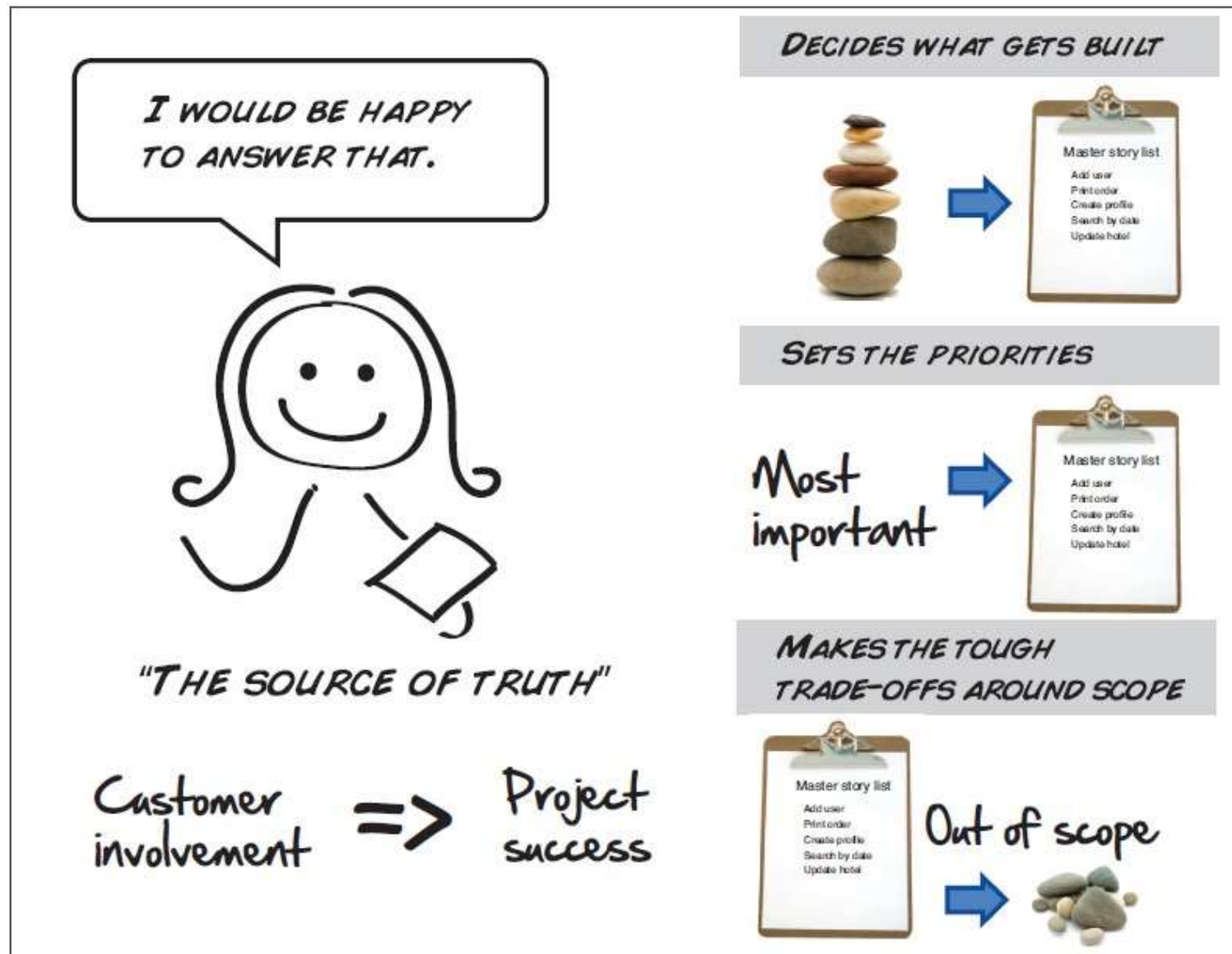
- Blurred instead of fixed roles
- Characteristics of successful teams
 - Co-located, at least for initial meetings
 - Engaged customer
 - Self-organizing instead of top-down
 - Accountable and empowered
 - Cross-functional

What if I don't have an engaged customer?

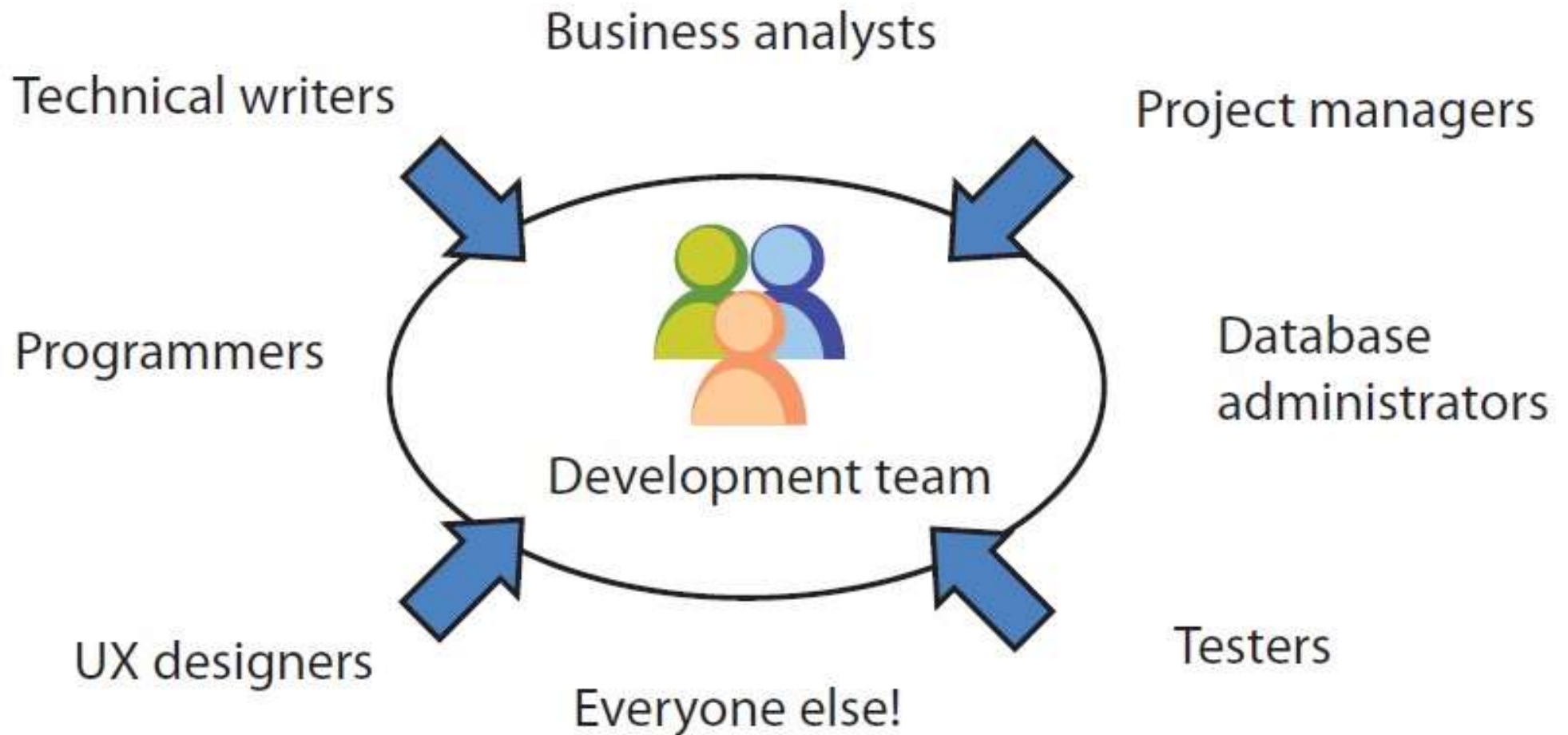
- Build credibility
 - Find a problem and make it go away
 - Show you are a fierce executor that will get things done and can help them
 - Might take a few iterations but they will see your value



The Agile Customer



Agile Development Team



Agile Analyst

*YOU CAN COUNT ON ME TO DO
OUR HOMEWORK, FOR EACH AND
EVERY ITERATION!*



"I SWEAT THE DETAILS."

HELPS WRITE USER STORIES

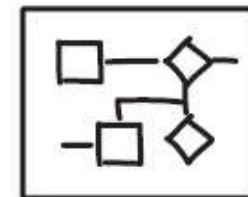


*I KNOW WHAT
I WANT, BUT HOW
DO I DESCRIBE IT !?*

DOES THE DETAILED ANALYSIS



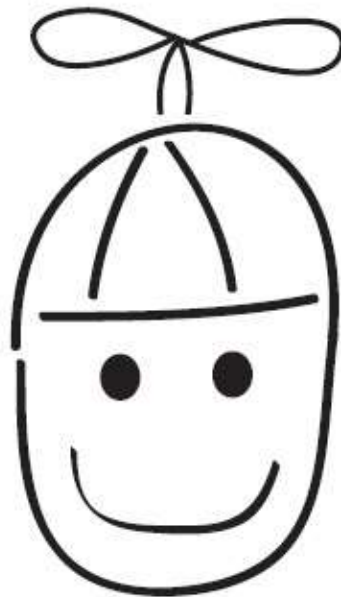
*MAKES SURE WE'VE DONE
OUR HOMEWORK*



Analysis artifacts

Agile Programmer

*THINK OF ME AS A CUSTOMER
WITH A KEYBOARD !*



*"BECAUSE THE RUBBER HITS
THE ROAD WHEN YOU CODE."*

*URNS USER STORIES INTO
WORKING SOFTWARE*

Create permit



```
if X  
then Y;
```

ESTIMATES (WITH REST OF TEAM)

Basic search

1 pt
3 pts
5 pts

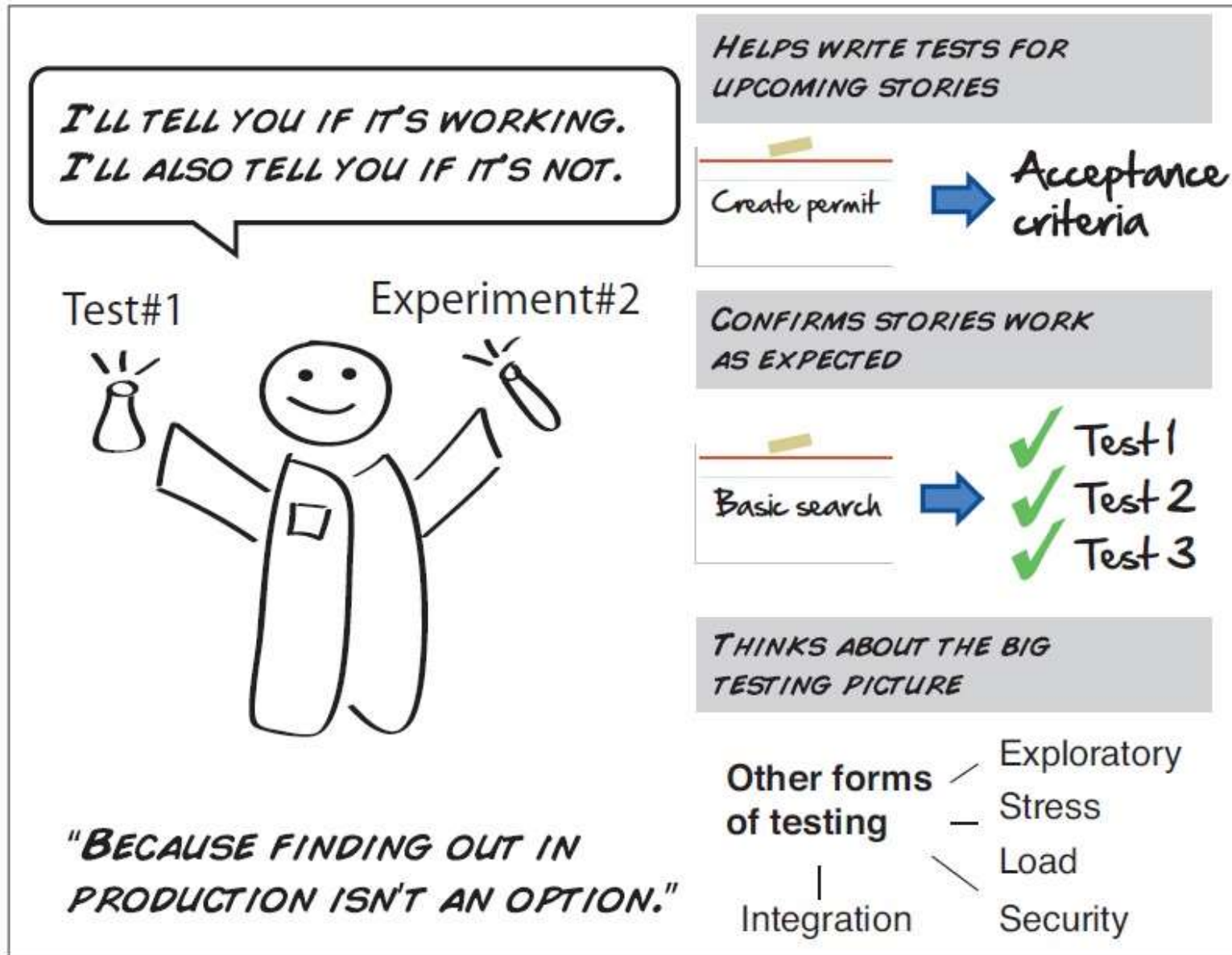
*MAKES THE TECHNICAL
DECISIONS*

TOOLS/ARCHITECTURE

DESIGN

DEVELOPMENT PRACTICES

Agile Tester



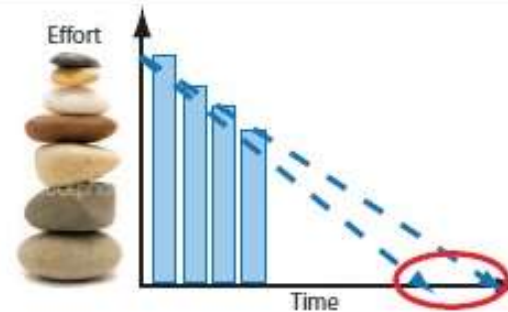
Agile Manager

HI! WHAT CAN I DO
FOR YA?



"WATCHING THE BOTTOM LINE."

TRACKS HOW WE ARE DOING



**COMMUNICATES THE STATE
OF THE PROJECT**



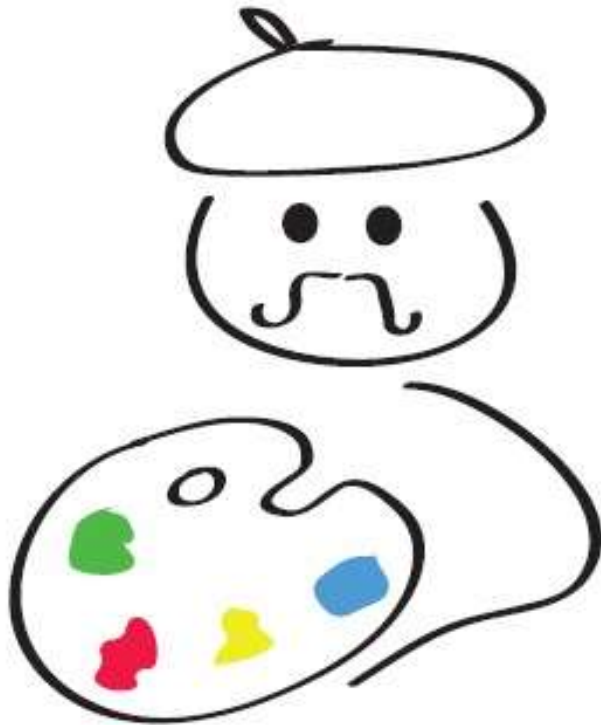
**REMOVES ROADBLOCKS
STANDING IN THE TEAM'S WAY**



**CO-LOCATED WORKSPACE
FAST COMPUTERS
SOFTWARE LICENSES
FOOT RUBS/BACK MESSAGES**

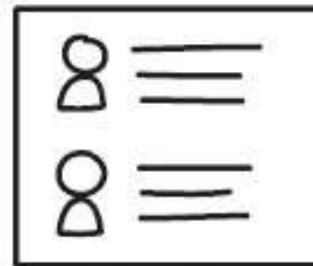
Agile Usability Designer

I ♥ CUSTOMERS



"BECAUSE IT'S COOL TO THINK ABOUT THE CUSTOMER."

USES A COLLECTION OF TOOLS AND TECHNIQUES TO HELP CREATE A COMPELLING USER EXPERIENCE

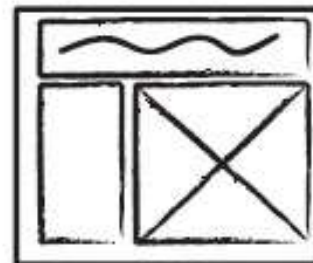


Personas

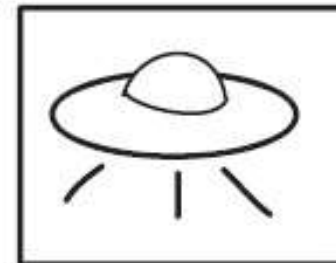


Storyboards

OVERLAPS WITH ANALYSIS



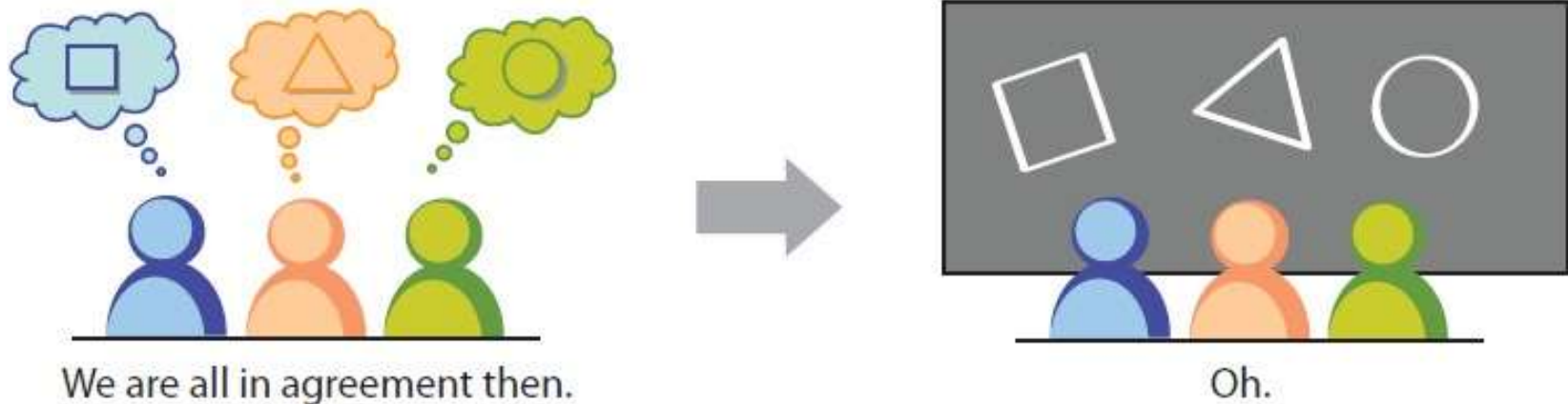
Paper prototypes



Concept designs

Kicking off a project

- The Inception Deck
 - Ten questions you'd be crazy not to ask before starting any software project
 - Gets everyone pointing in the same direction
 - Shared goals, vision, context



Inception Deck

- Collectively fill out a slide on to get a pretty good idea about what the project is, what it isn't, and what it's going to take to deliver
- Need to get customer/stakeholders involved
- It's a living document

<Your project name>

<Your sponsors>

Why are we here?

- Important reason #1
- Important reason #2
- Important reason #3



<#1 reason for doing this project>

The elevator pitch

- For [target customer]
- who [statement of need or opportunity]
- the [project name]
- is a [product category]
- that [key benefit, compelling reason to buy].
- Unlike [primary competitive alternative]
- our project [statement of primary differentiation].

Product box

<product name>

fun picture

<slogan>

<benefit #1>

<benefit #2>

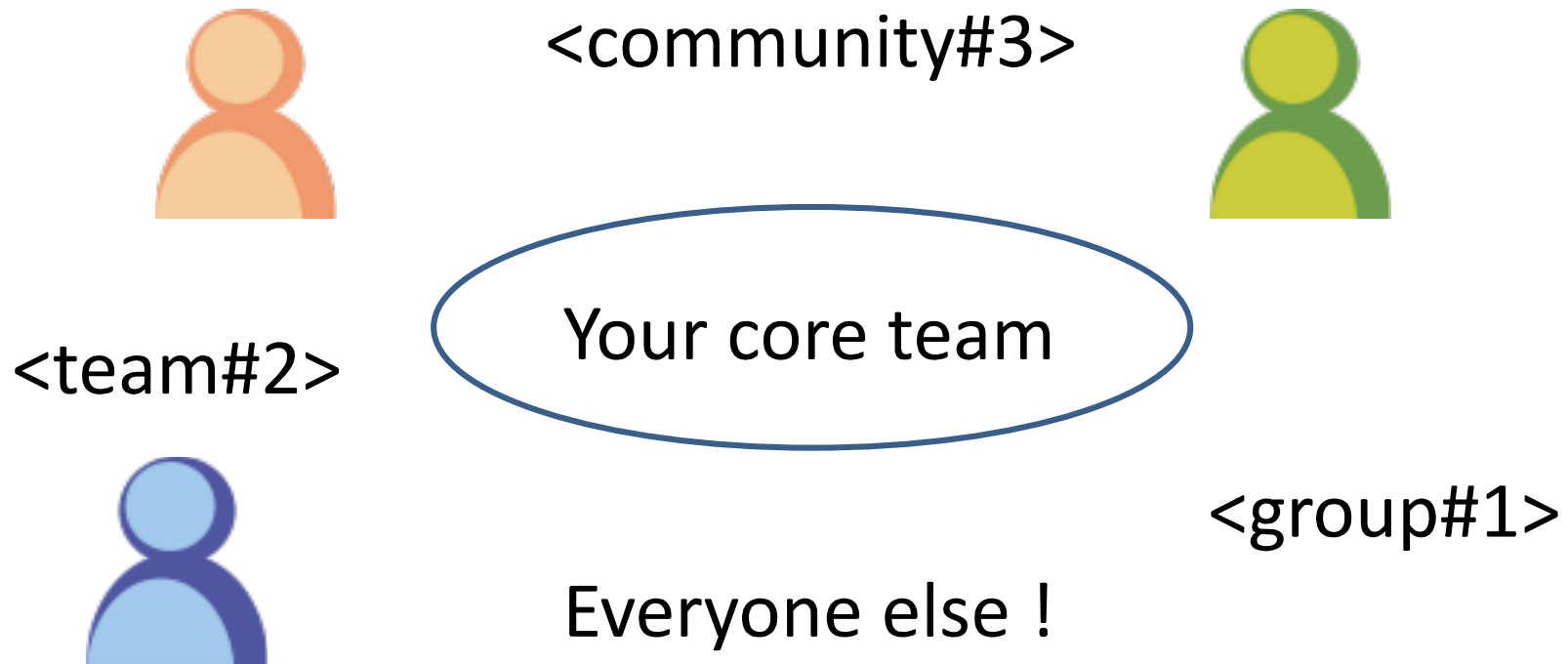
<benefit #3>

The NOT list

IN	OUT

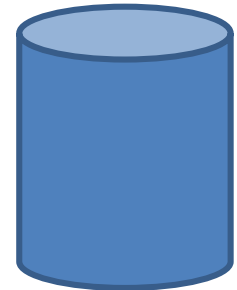
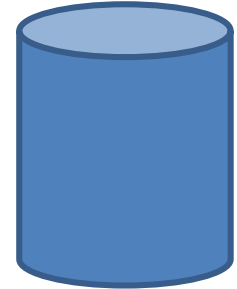
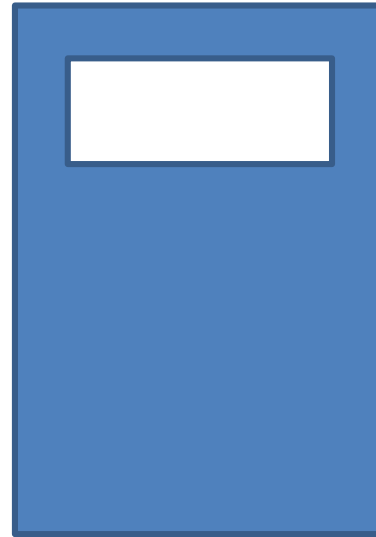
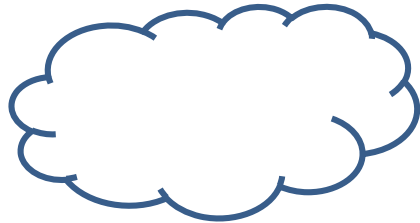
UNRESOLVED

Your project community



... is always bigger than you think!

Technical solution



Technologies:

- <language>
- <libraries>
- <tools>
- <technology>

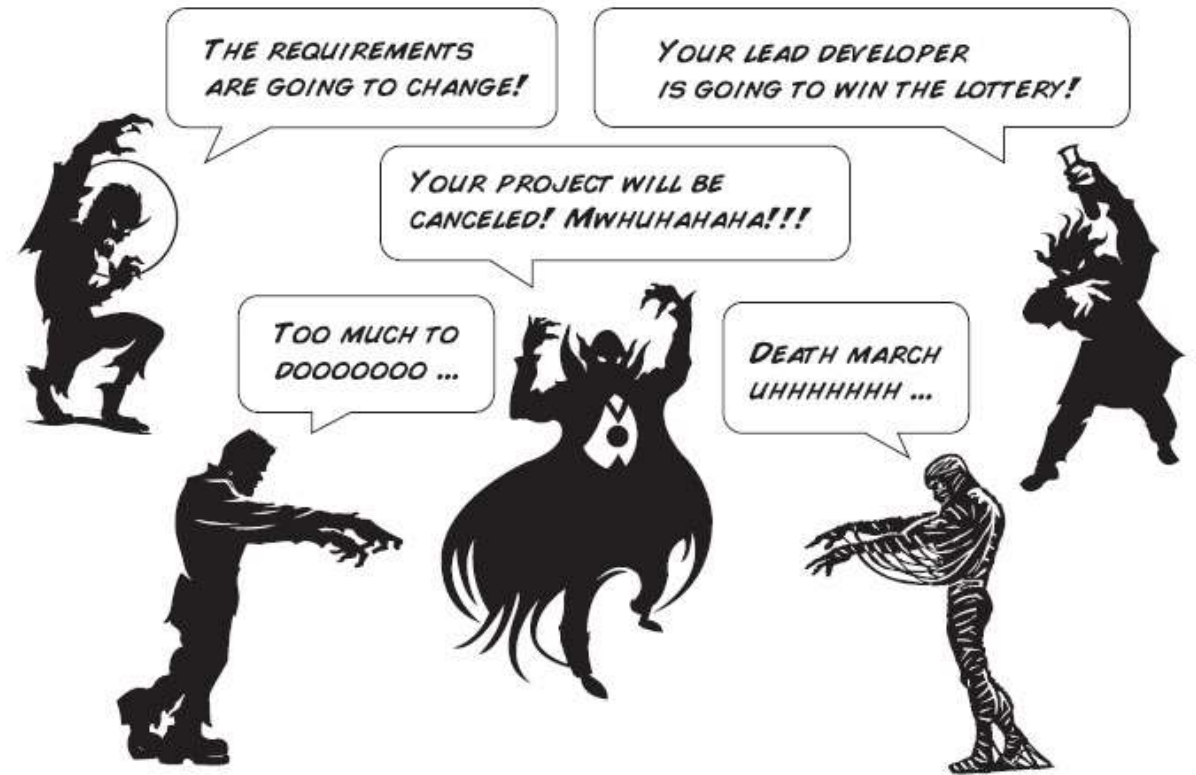


Danger!

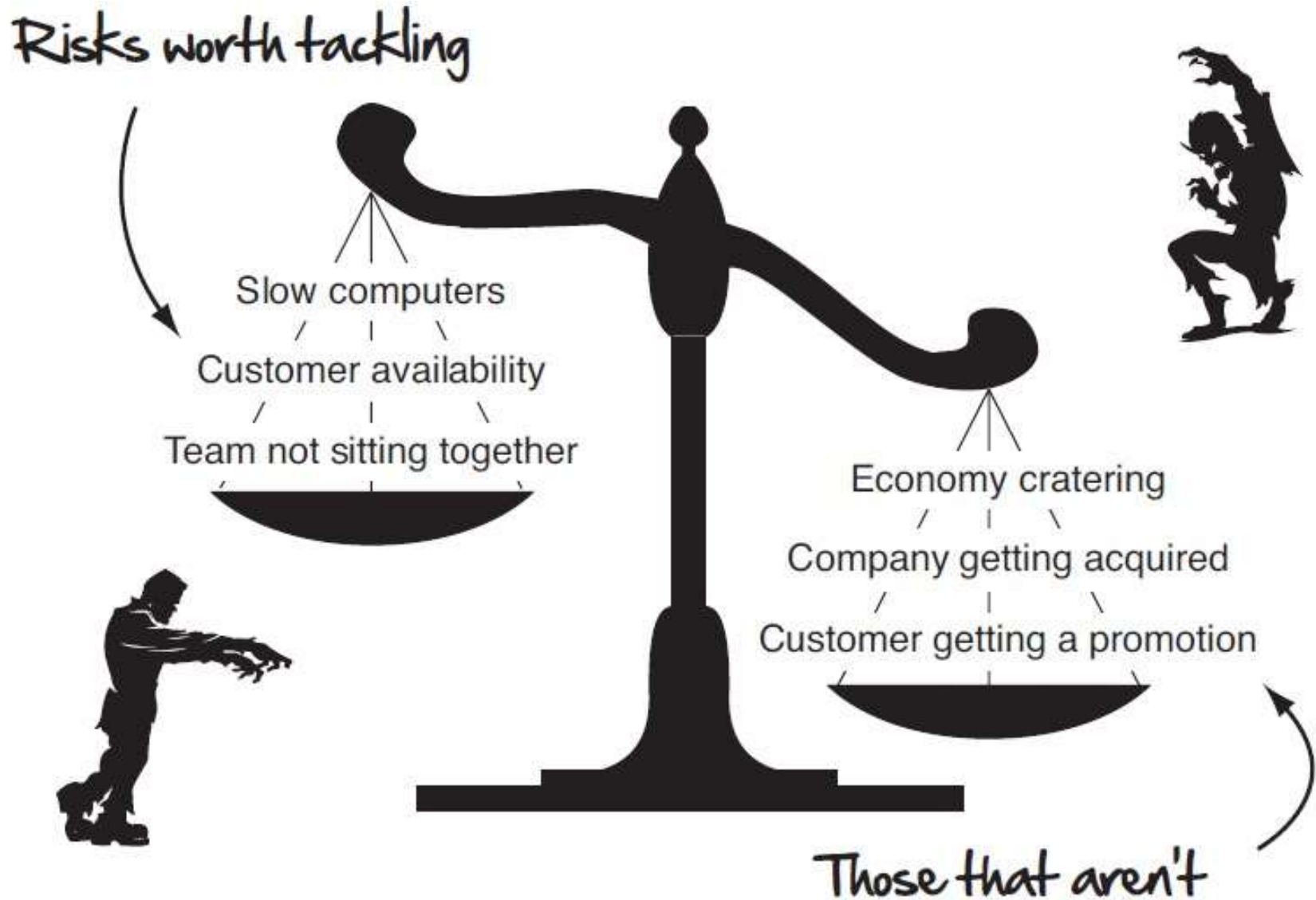
Out of
scope

What keeps us up at night

- <scary thing #1>
- <scary thing #2>
- <scary thing #3>



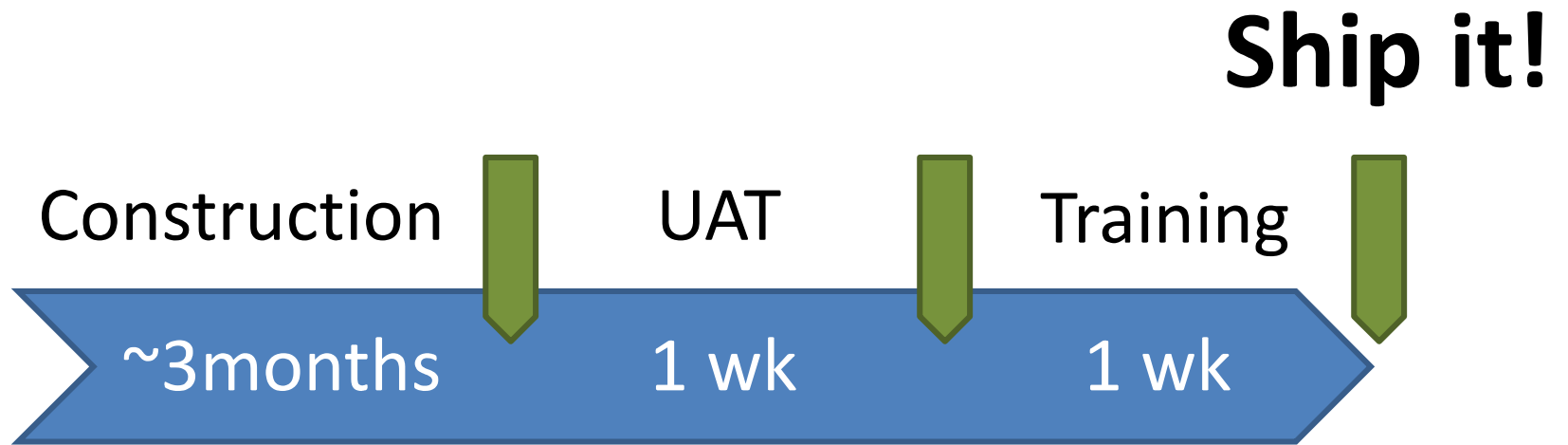
Don't overdo it



The A-Team

#	Role	Competencies/Expectations
1	Analyst	Comfortable with just-in-time analysis. Likes to test. Comfortable with rapid iterative development.
2	Developers	C#, MVC.NET, jQuery, SQL Unit testing, refactoring, TDD, continuous integration
0.5	Project manager	Responsible for outward facing communication Status reports, scope, budget, and reporting upwards

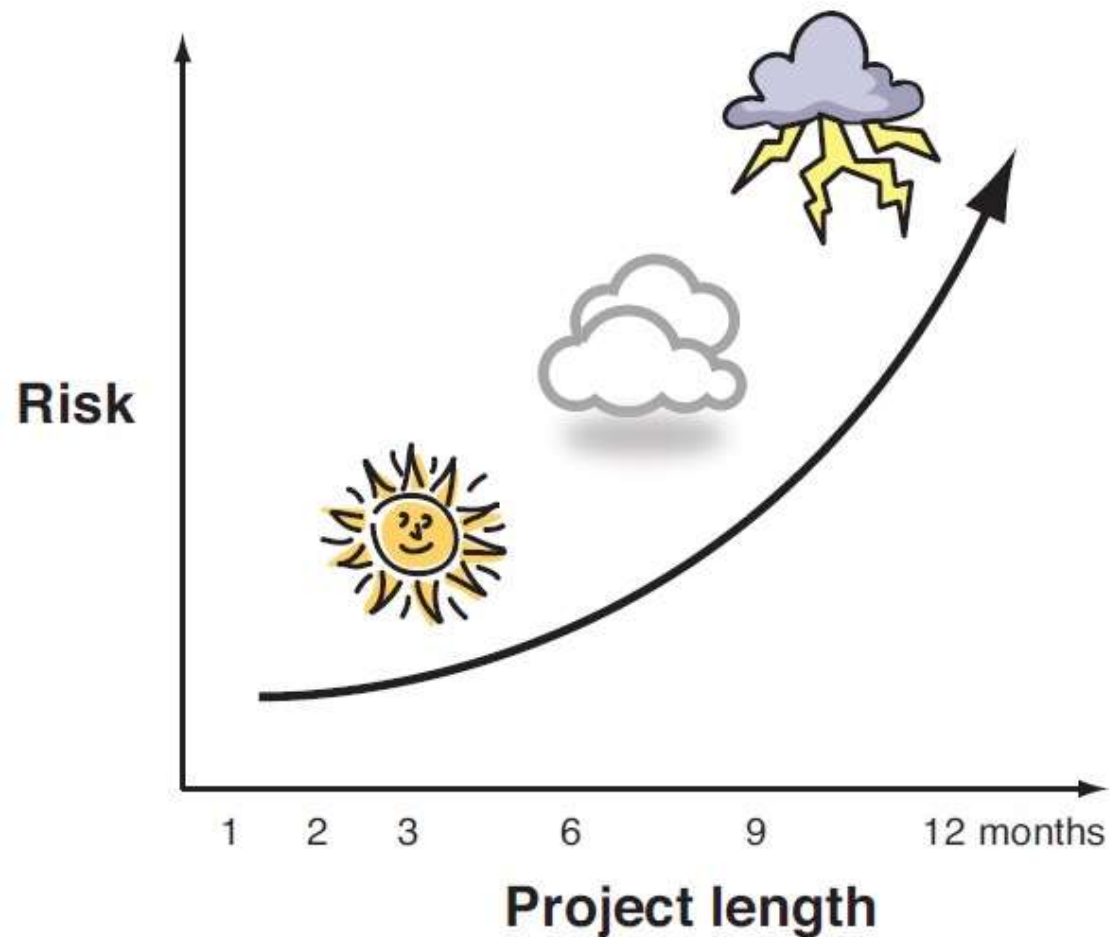
How big is this thing?



This is a guess. Not a commitment.

A hand-drawn squiggly line in black ink, starting from the end of the timeline and extending downwards and to the right.

Risk vs. Time



The risk of project failure increases over time – think small

The Test

*MASTER. CAN YOU TEACH
ME HOW TO DEAL WITH THE
FORCES I WILL FACE ON MY
SOFTWARE PROJECT ?*



*THIS I CAN DO.
BUT FIRST YOU MUST PASS
MY TEST ...*



*OR YOU WILL
SURELY DIE.*

The Test

1. Which of these forces is most precious to a software project?

- a) Quality.
- b) Time.
- c) Scope.
- d) Budget.

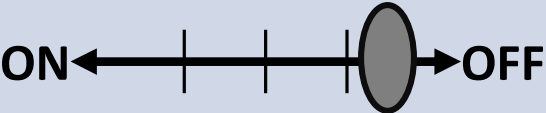
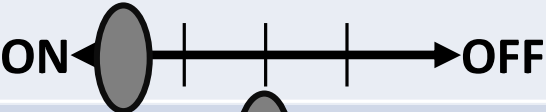

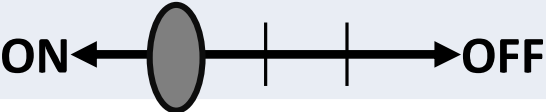
2. When faced with too much to do and not enough time, is it better to do the following:

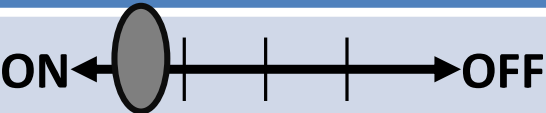
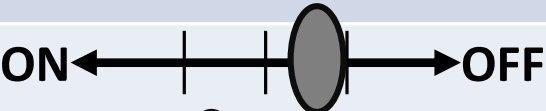


- a) Cut scope
- b) Add more people to the project
- c) Push out the release date
- d) Sacrifice quality

3. Which is most painful?

- a) Walking on fire
- b) Chewing broken glass
- c) Doing the Macarena
- d) Asking your sponsor for more money

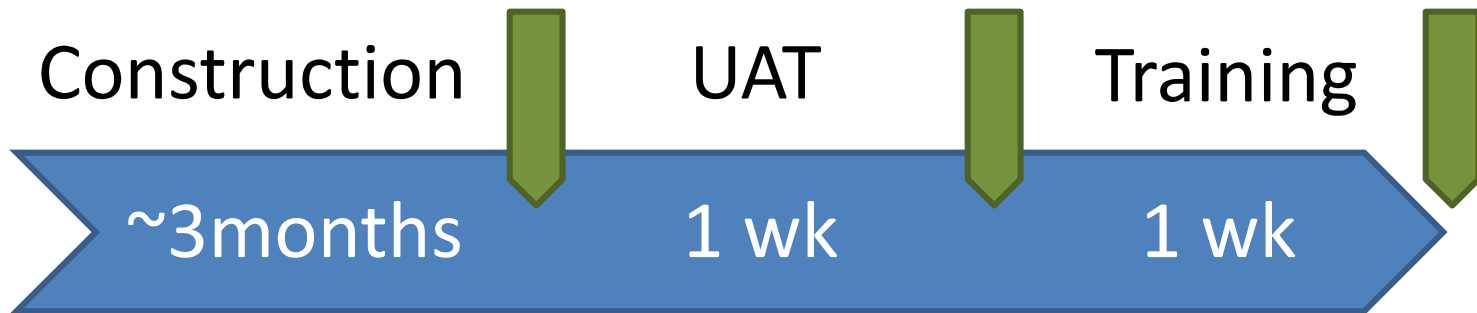
Trade-off sliders

	The classic four
	Feature completeness (scope)
	Stay within budget (budget)
	Deliver project on time (time)
	High quality, low defects (quality)

	Other important things
	Ease of use
	Community of users
	Detailed audits (log everything)
	<insert yours>

The first release

Ship it!



3 people, 3 ½ months, \$250K