# Agile Methodologies

# Winston

@winstonyw

2011 ~ 2013

# Pivotal Labs SG

## Agile Software Engineering

Test Driven Development • Pair Programming • Continuous Integration

2011 ~ 2013

# Neo Innovations

## Agile Software Engineering

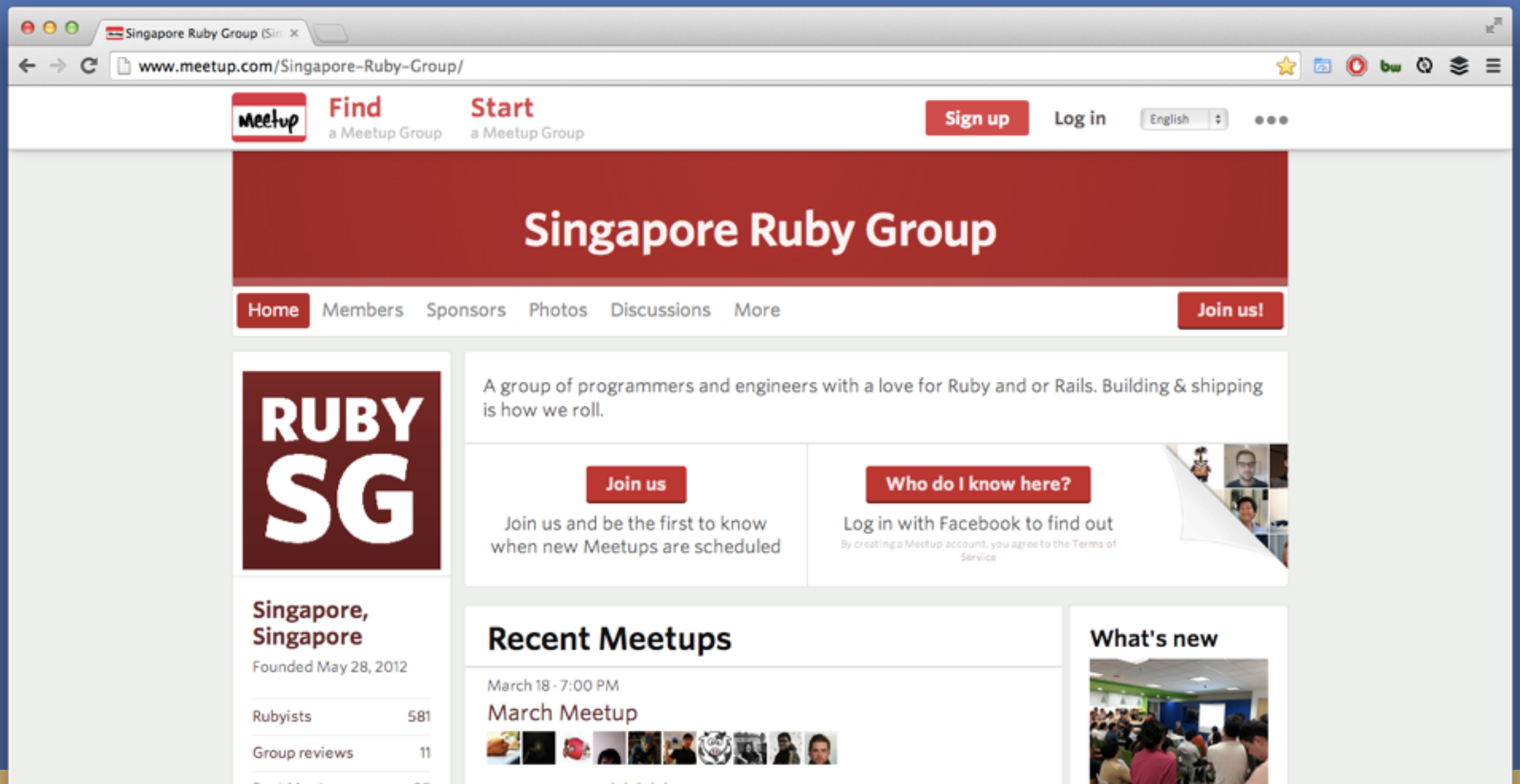
Test Driven Development • Pair Programming • Continuous Integration

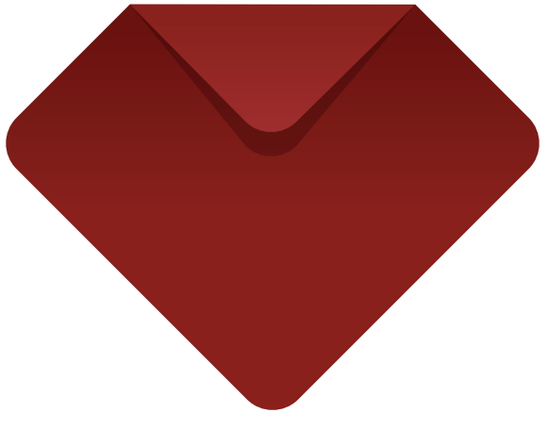
*Established*

*Since Nov 2013*

# SG Ruby Group

**http://www.meetup.com/Singapore-Ruby-Group/**





# REDDOT

RUBY CONFERENCE

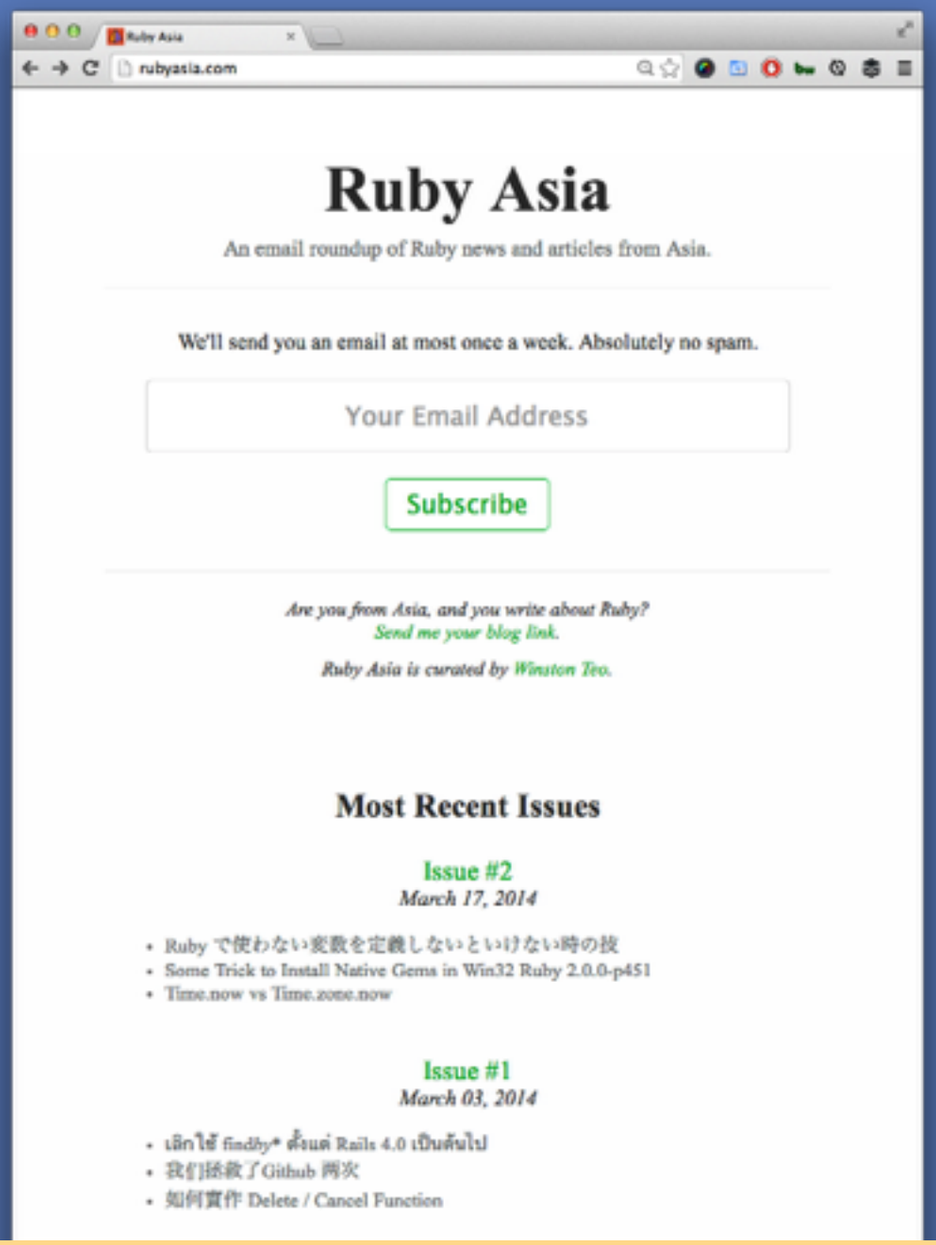
June 26-27, 2014

Singapore

http://www.reddotrubyconf.com

# RubyAsia.com

## A fortnightly newsletter for Ruby news and articles from Asia

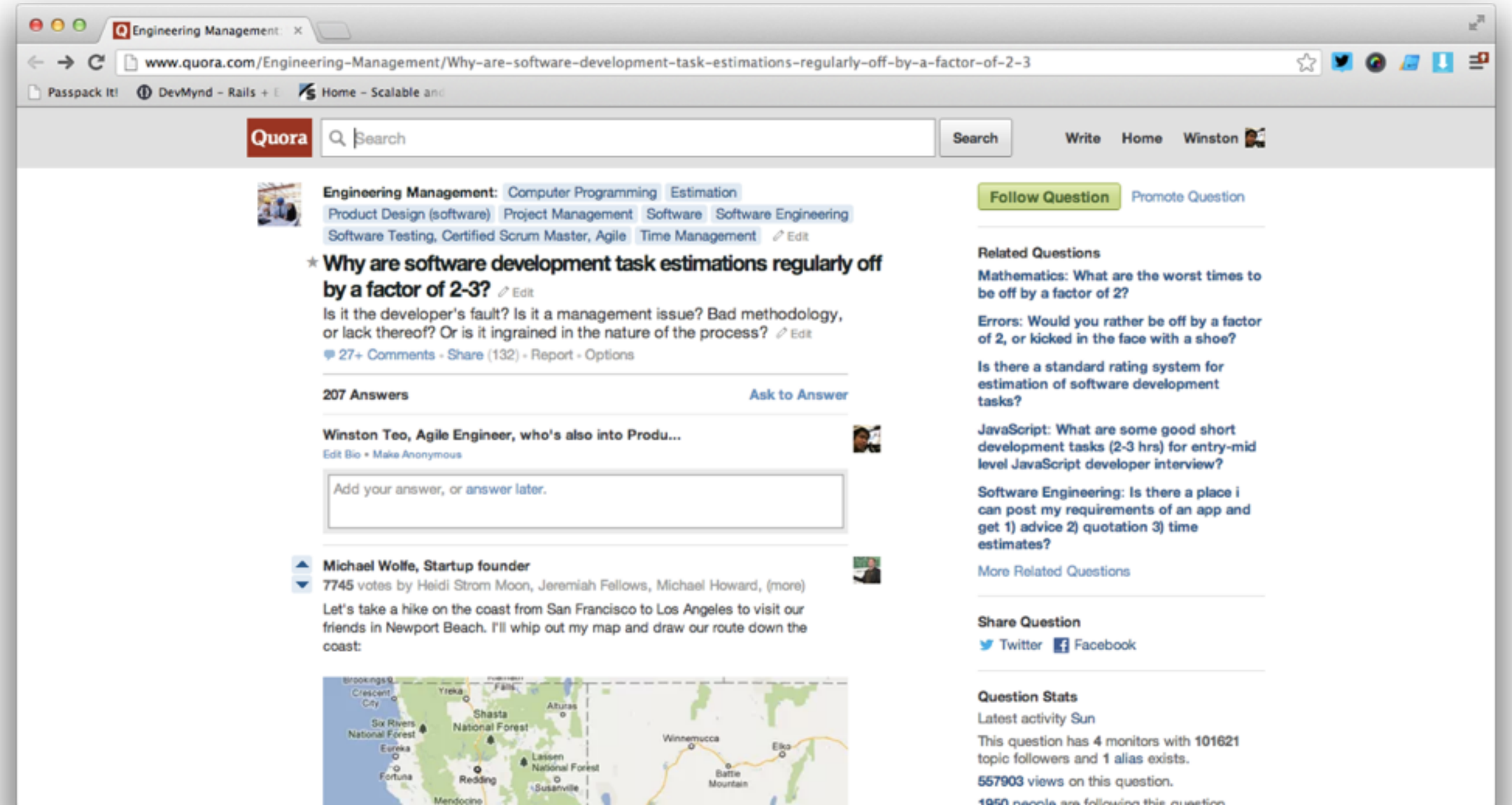


# Why do Software Projects Fail?

# PLANNING AND ESTIMATING IS DIFFICULT

# Why Are Software Development Task Estimations Regularly Off By A Factor of 2-3

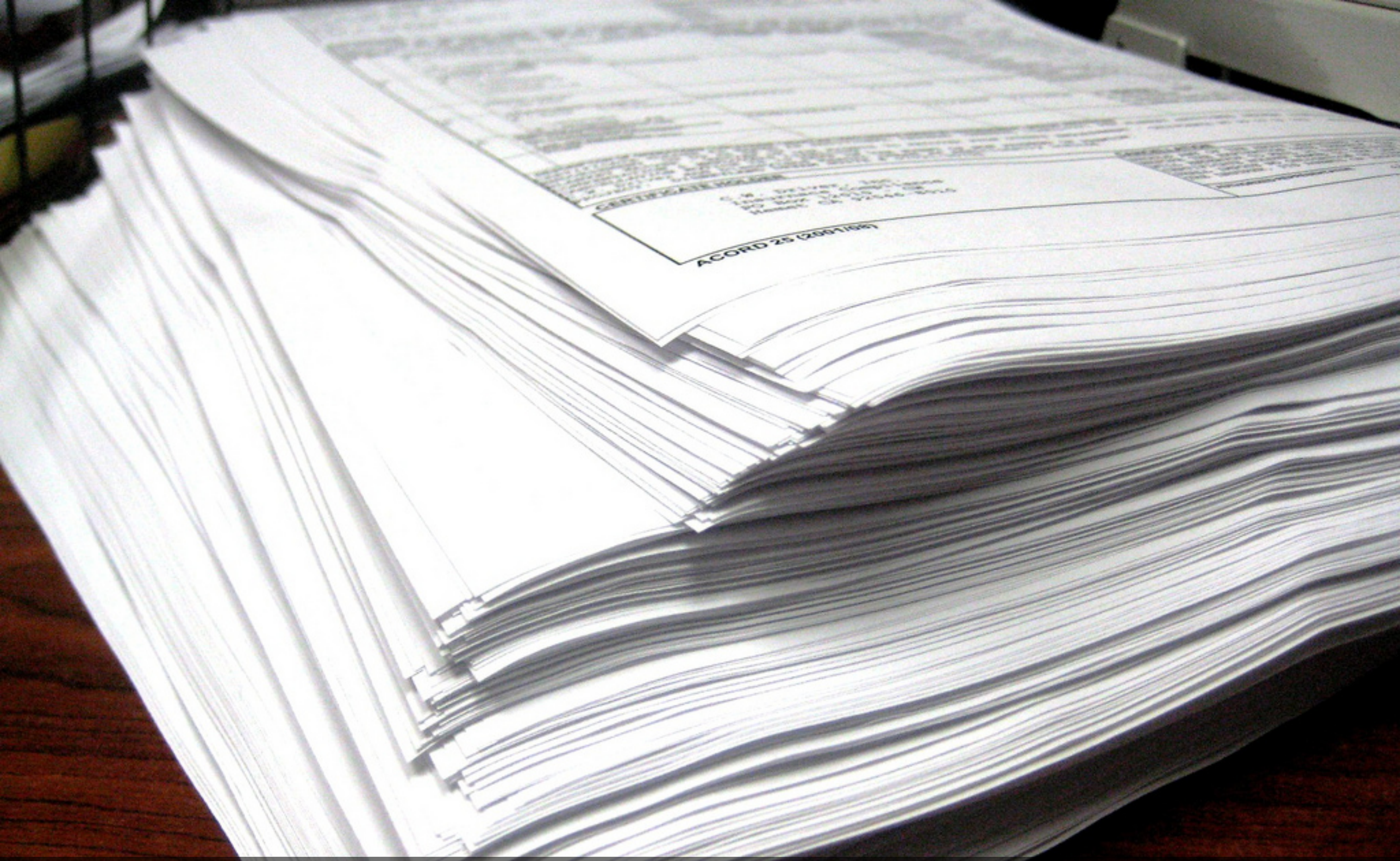
http://www.quora.com/Engineering-Management/Why-are-software-development-task-estimations-regularly-off-by-a-factor-of-2-3



# Waterfall

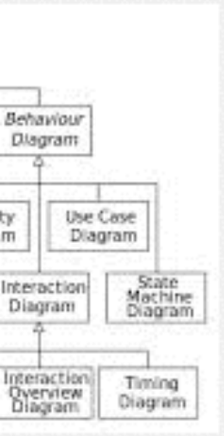
http://www.flickr.com/photos/mcbridejc/8885815845/

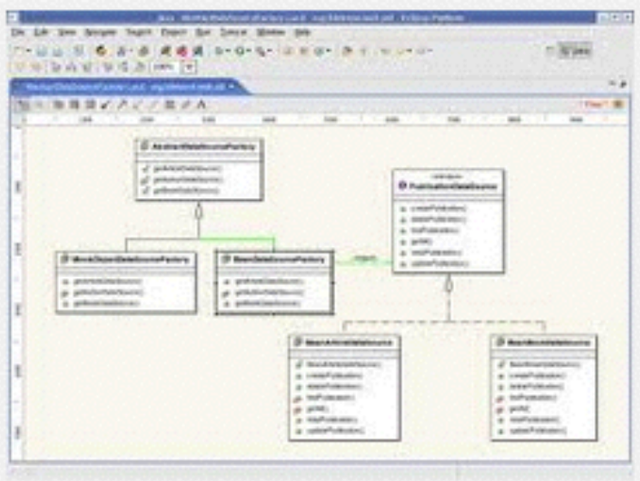
# Problems with Traditional Planning & Estimation

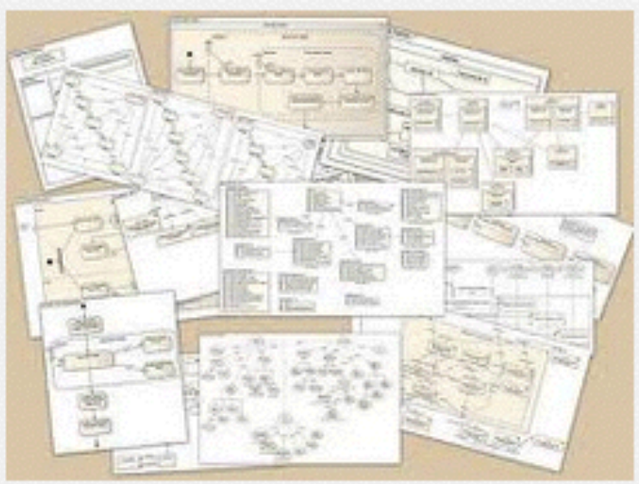


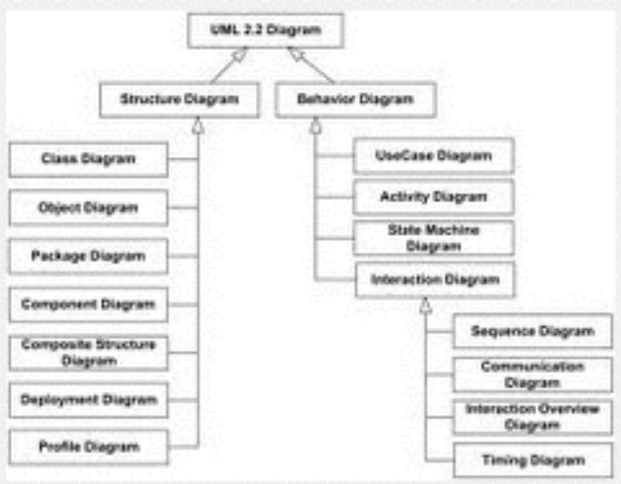

# Almighty Documentation

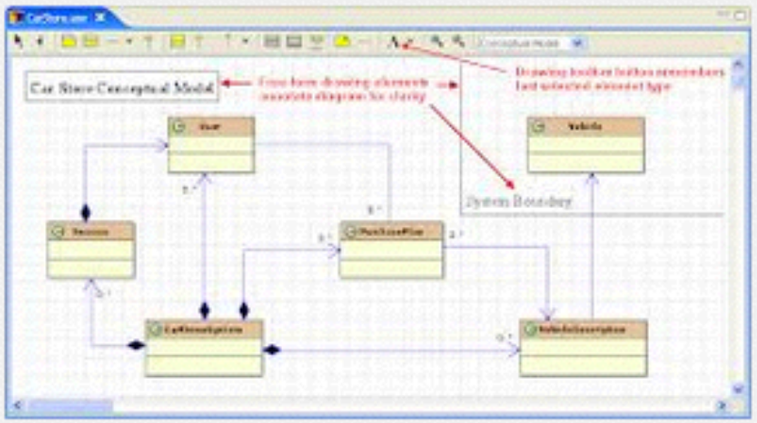
http://www.flickr.com/photos/sonrisaelectrica/3695743740/

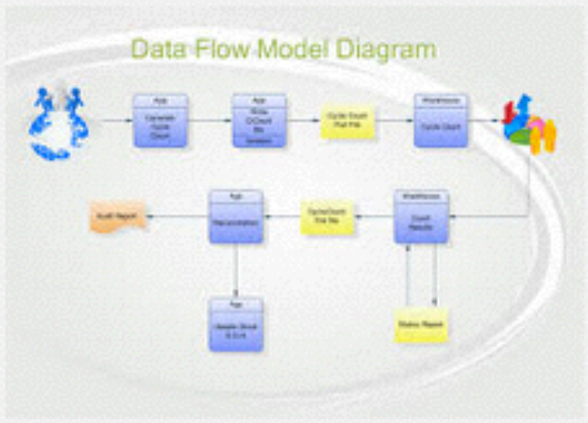


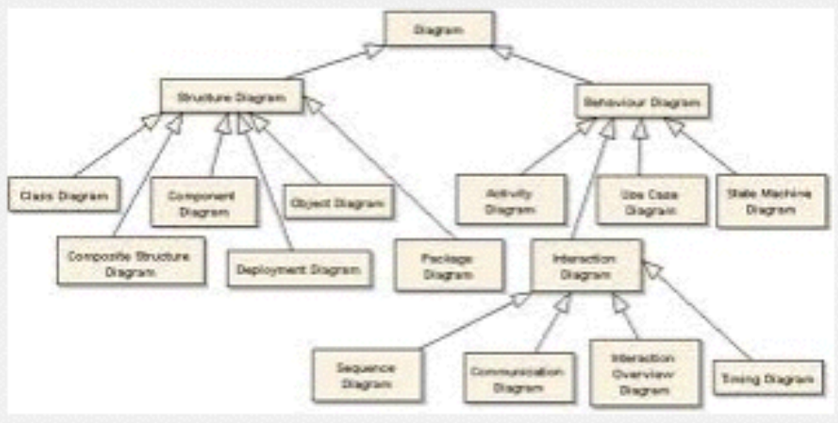



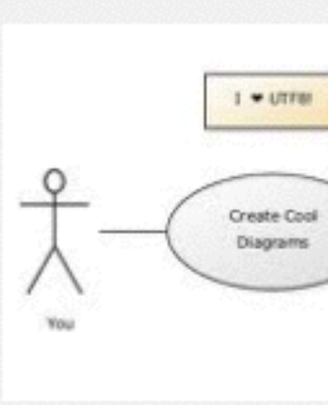


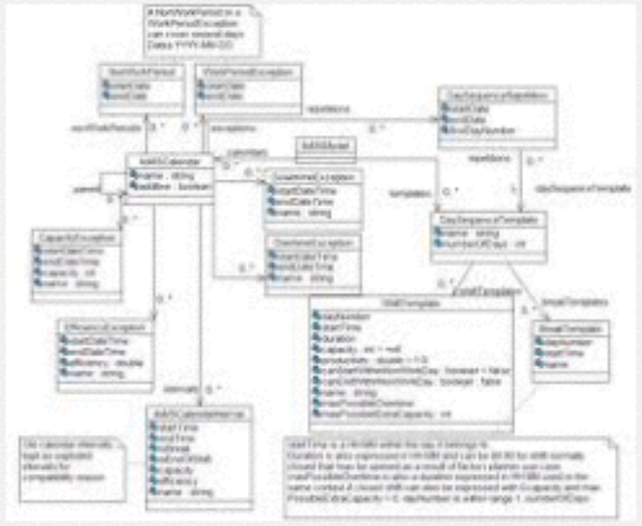


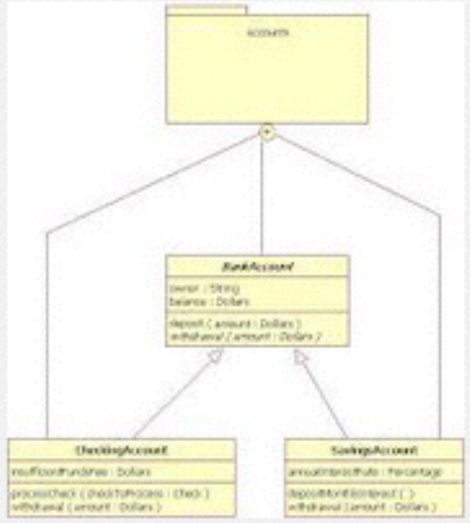


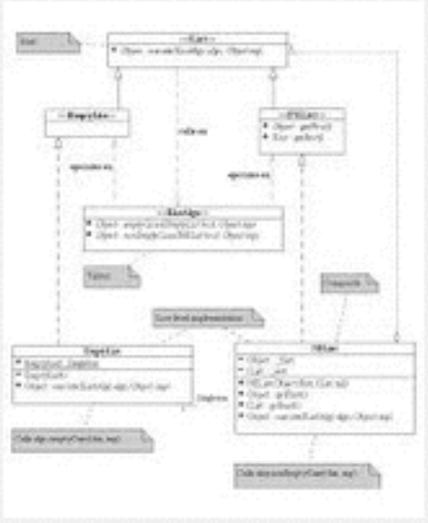



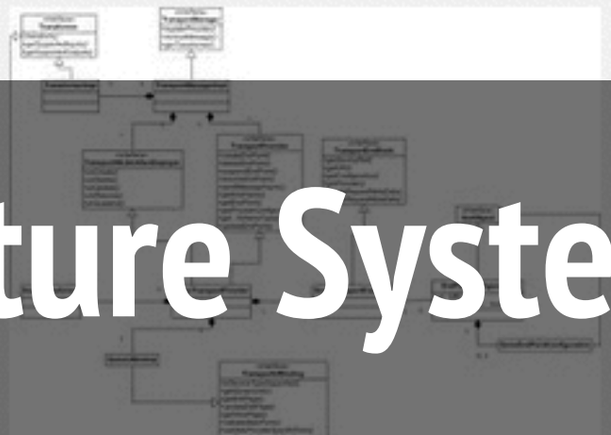



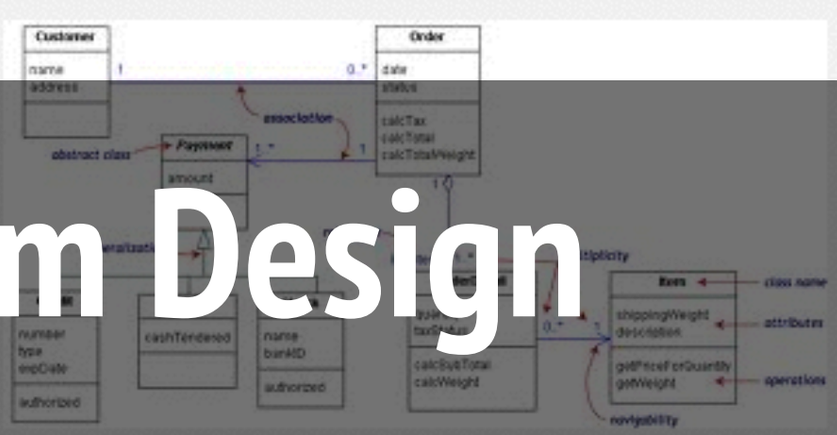



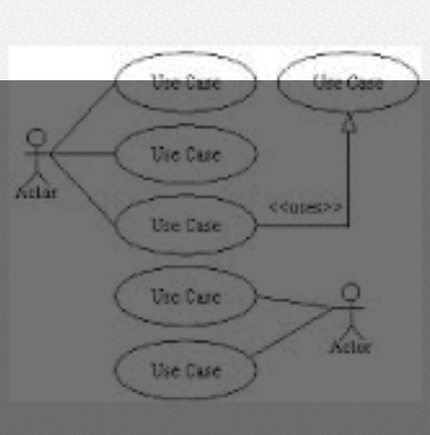






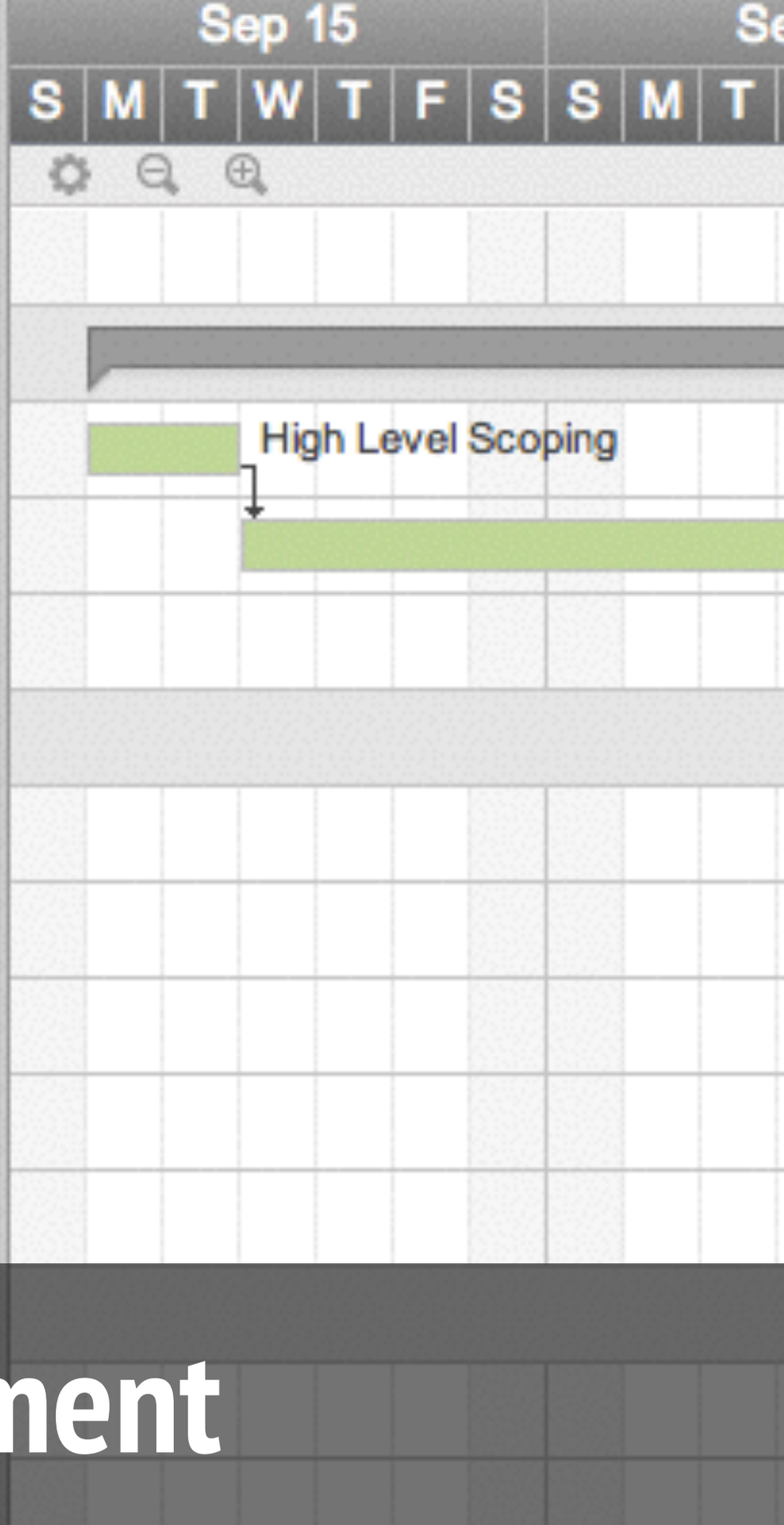




# Premature System Design

| Task Name | Start Date | End Date |
|---|---|---|
| **Requirements Gathering** | **09/16/13** | **10/04/13** |
| High Level Scoping | 09/16/13 | 09/17/13 |
| Design | 09/18/13 | 09/27/13 |
| Wireframe | 09/30/13 | 10/04/13 |
| **Build** | **10/07/13** | **11/08/13** |
| Home Page | 10/07/13 | 10/11/13 |
| Contact Us | 10/14/13 | 10/18/13 |
| Blog | 10/21/13 | 10/25/13 |
| Products Page | 10/28/13 | 11/01/13 |
| SEO | 11/04/13 | 11/08/13 |
| **Testing & Deployment** | **11/11/13** | **11/18/13** |
| QA | 11/11/13 | 11/13/13 |
| Deployment | 11/14/13 | 11/18/13 |




# Activity-Based Management

# Wrong Perception of Estimation

VS.

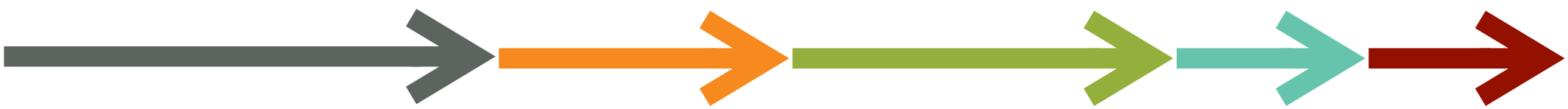
# No Customer Collaboration

**“This project will be completed by Christmas.**

**- Project Manager,**
**based on Traditional Planning**

# Unrealistic Timeline

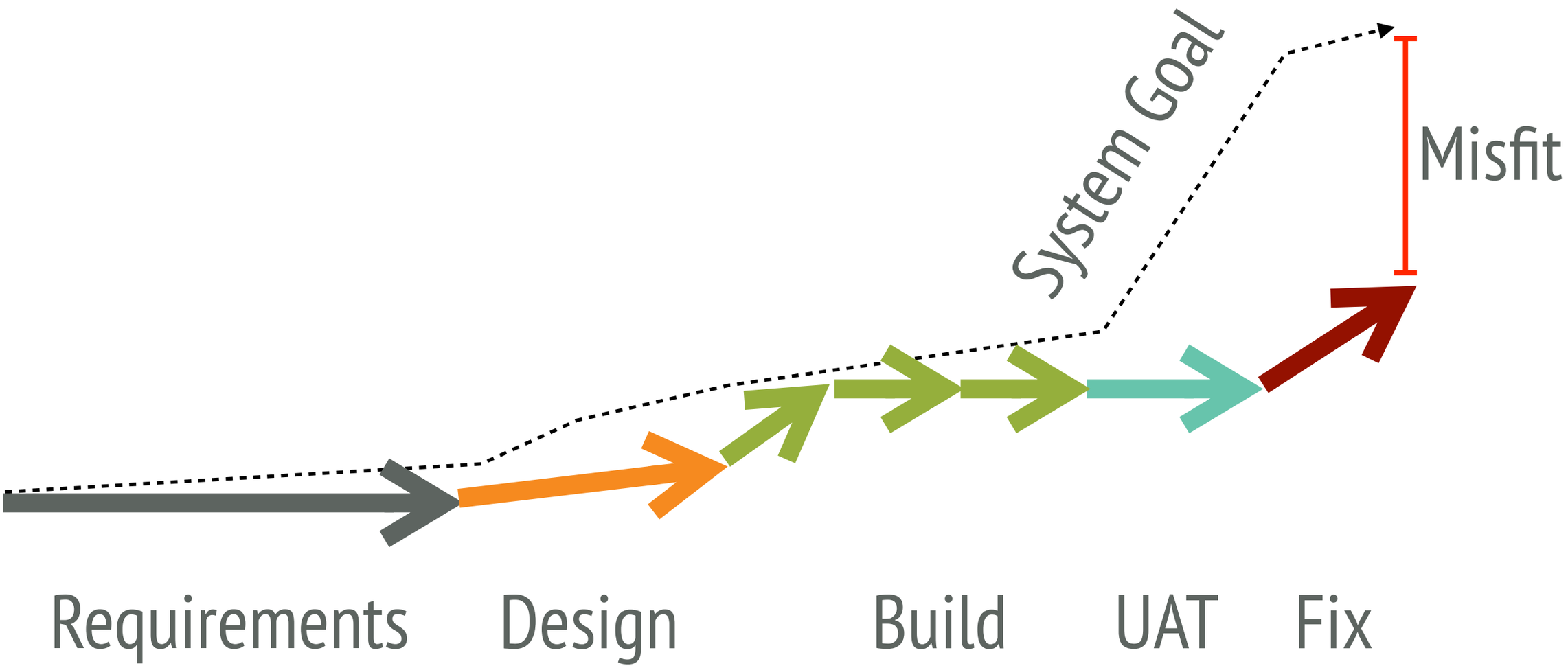
# Waterfall



Requirements Design Build UAT Fix

*Time to Ship*

# Waterfall Product/Market Fit



SOFTWARE SERVES

# BUSINESS NEEDS

# SOFTWARE IS IMPORTANT!

WHAT'S A BETTER WAY TO PLAN AND ESTIMATE..

Features

Resources

Duration

Cost

# AGILE

# Manifesto

*agilemanifesto.org*

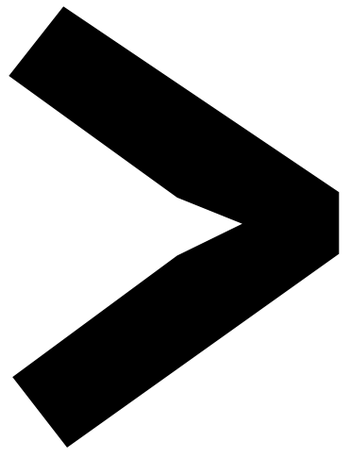
# Manifesto

**Individuals and Interactions**

**Working Software**

**Customer Collaboration**

**Responding to Change**



**Processes and Tools**

**Comprehensive Documentation**

**Contract Negotiation**

**Following a Plan**

# Benefits of Agile

# HIGHER VISIBILITY.

# HIGHER ADAPTABILITY.

# REDUCED RISK AND UNCERTAINTY.

# GREATER BUSINESS VALUE.

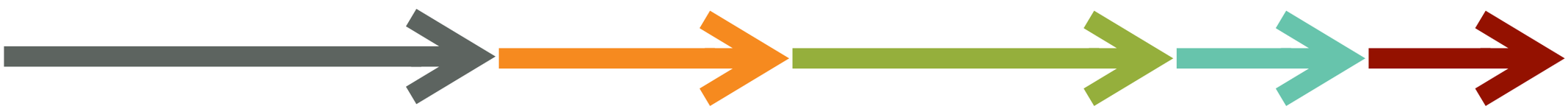
[for the engineers]

# EMPOWERED.

# PURPOSE.

# Comparison

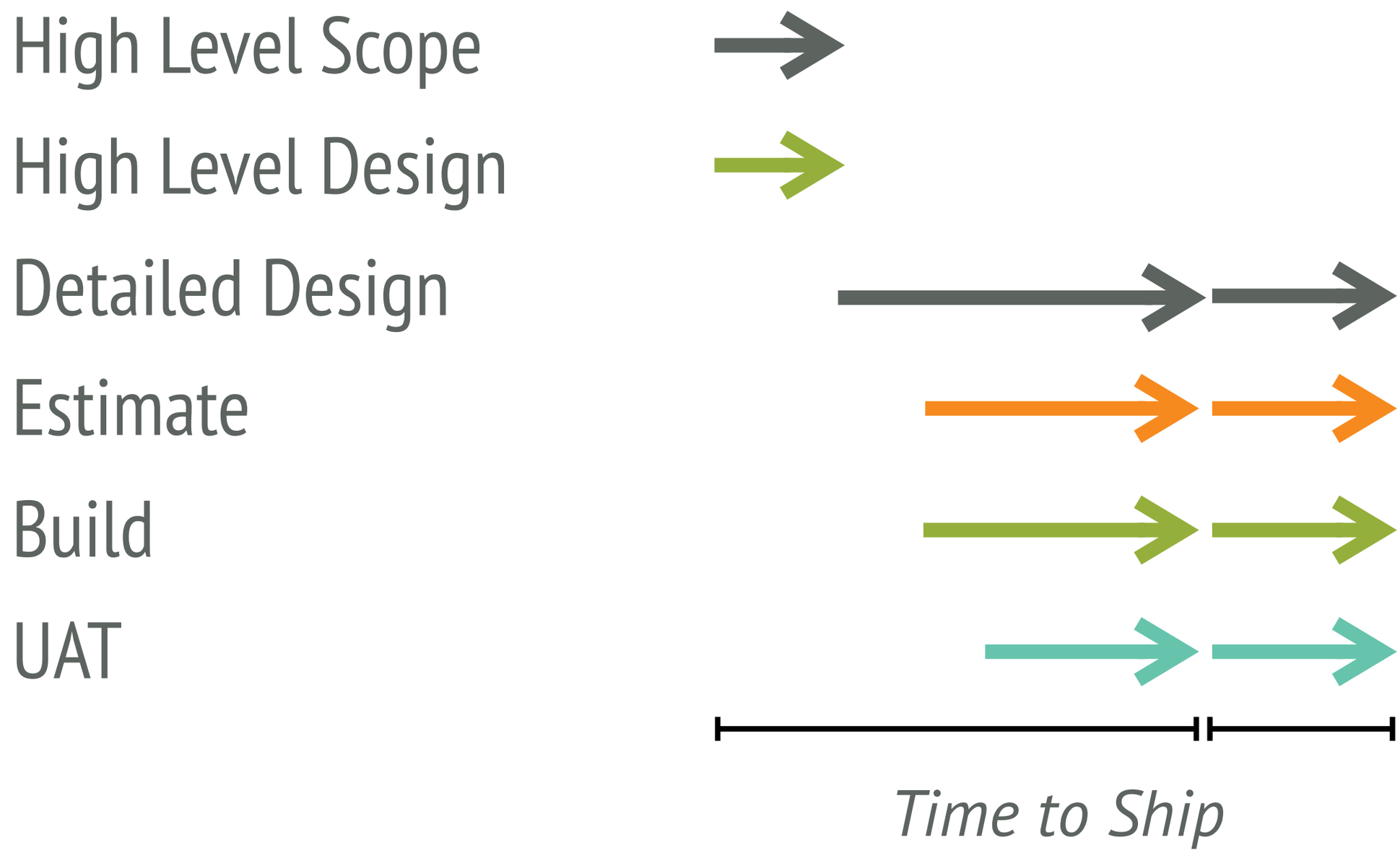
# Waterfall



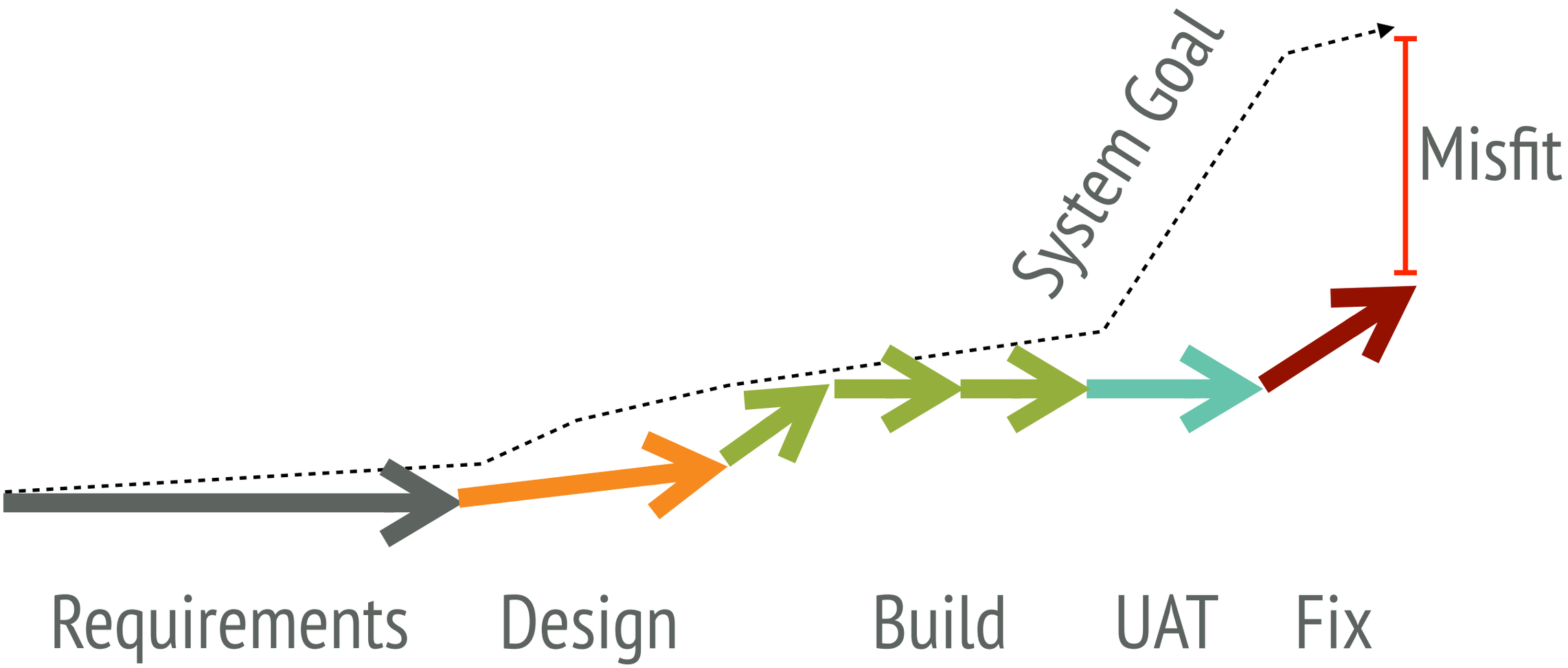
Requirements Design Build UAT Fix

*Time to Ship*

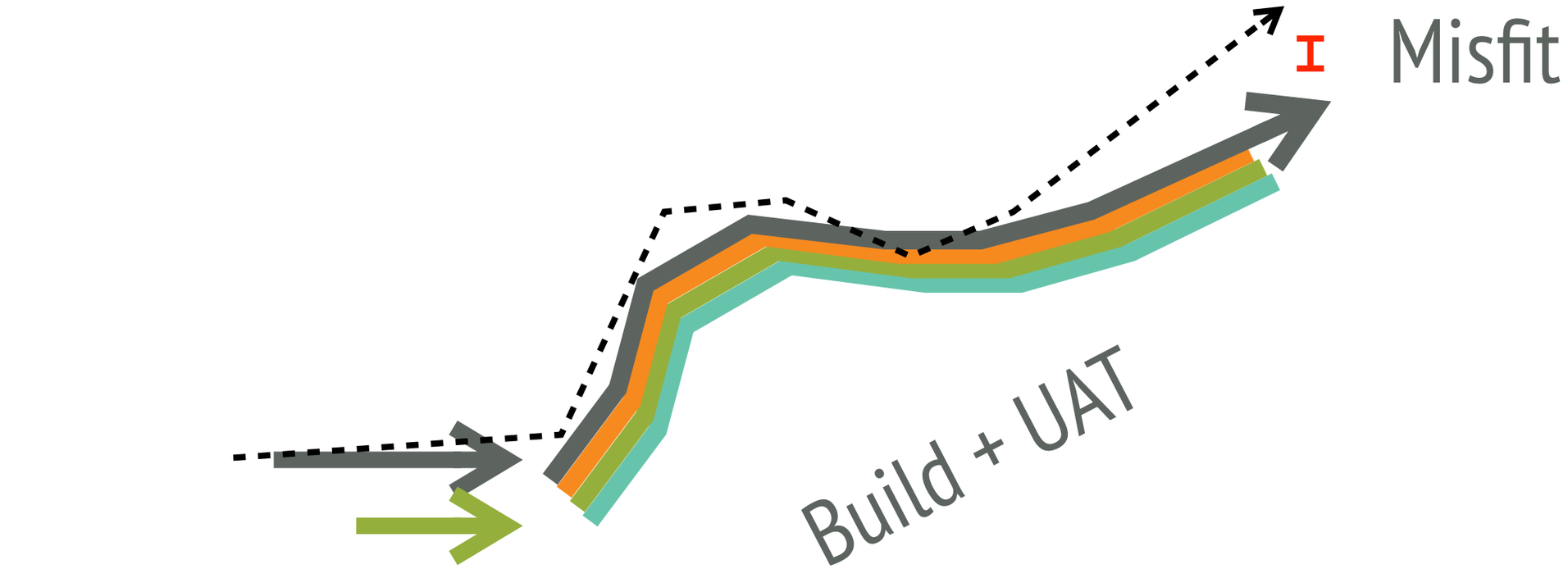
# Agile


High Level Scope
High Level Design
Detailed Design
Estimate
Build
UAT
Time to Ship

# Waterfall Product/Market Fit



# Agile Product/Market Fit



# Agile Implementations

**Kanban**
**Scrum**
**Extreme Programming**
**etc**

# Roles in an Agile Project

Project Manager

**Customers** **Product Owner** **Engineers**

Icons: http://www.iconarchive.com/show/sleek-xp-basic-icons-by-deleket

# Customers

- Use the system/service
- Provides feedback on the system/service

# Product Owner

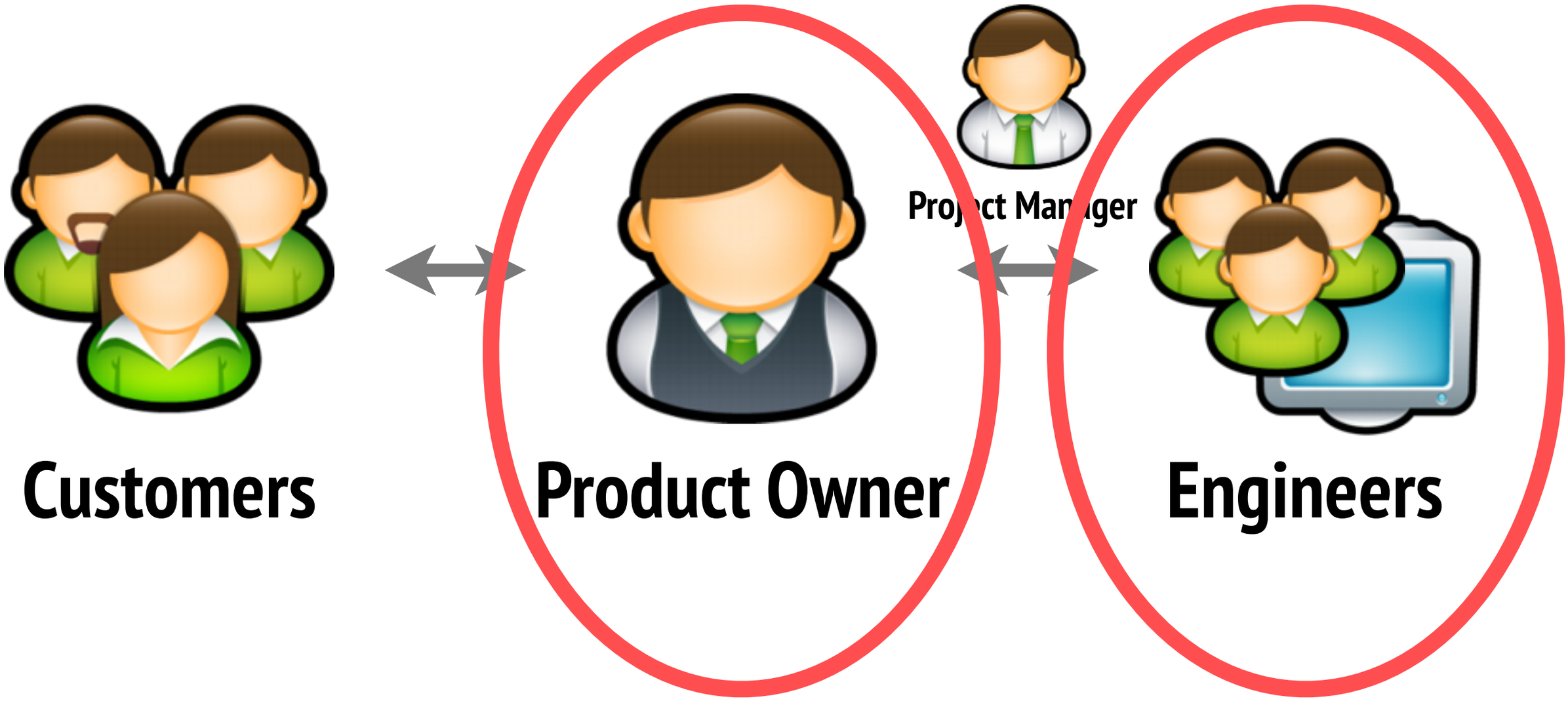
- Represents the stakeholders
- Represents customers and subject matter experts
- Manages product backlog by ranking and prioritizing user stories
- Ensures that the Team delivers value to the business

# Project Manager

- Bridge between the Product Owner and Engineers
- Could be the Lead Developer too

# Engineers

- Write the code
- Estimate the stories effort
- Provide expertise on simplest solution first
- Pivotal to the success or failure of a project




Icons: http://www.iconarchive.com/show/sleek-xp-basic-icons-by-deleket

# Management Practices

# Engineering Practices

# Management Practices

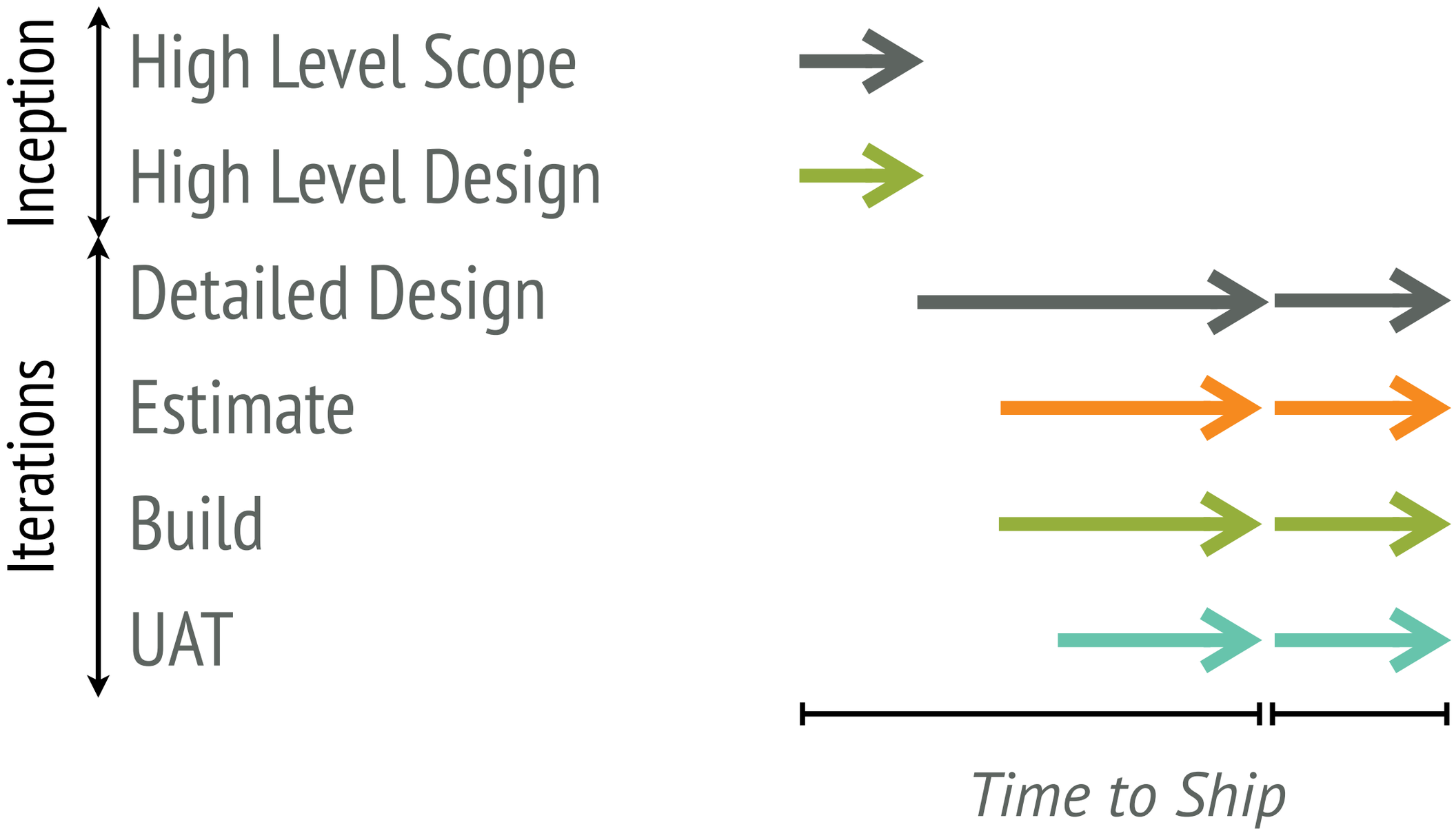
- Inception
- Iteration Planning
- Release Planning
- Retrospectives

# Engineering Practices

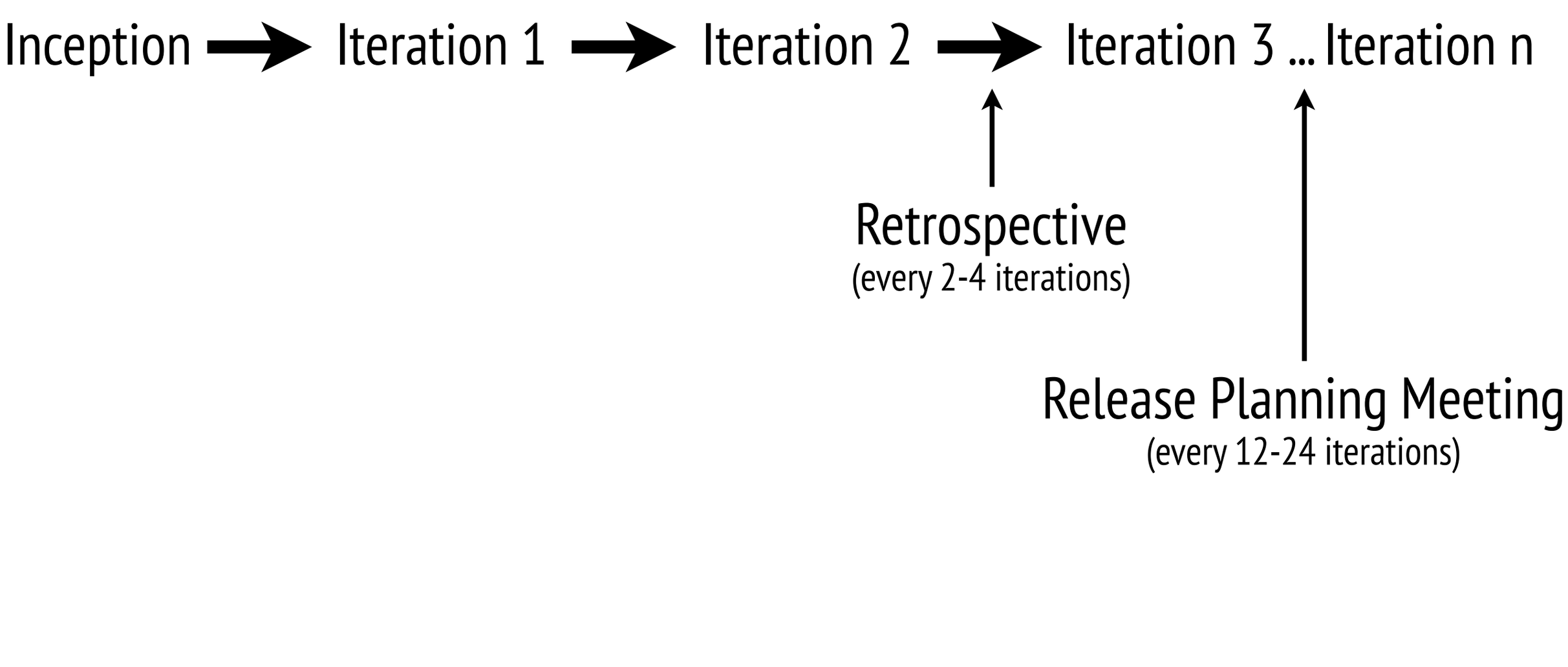
- Daily Standups
- Pair Programming
- Test Driven Development
- Continuous Integration/Deployment

# Agile Project Life Cycle

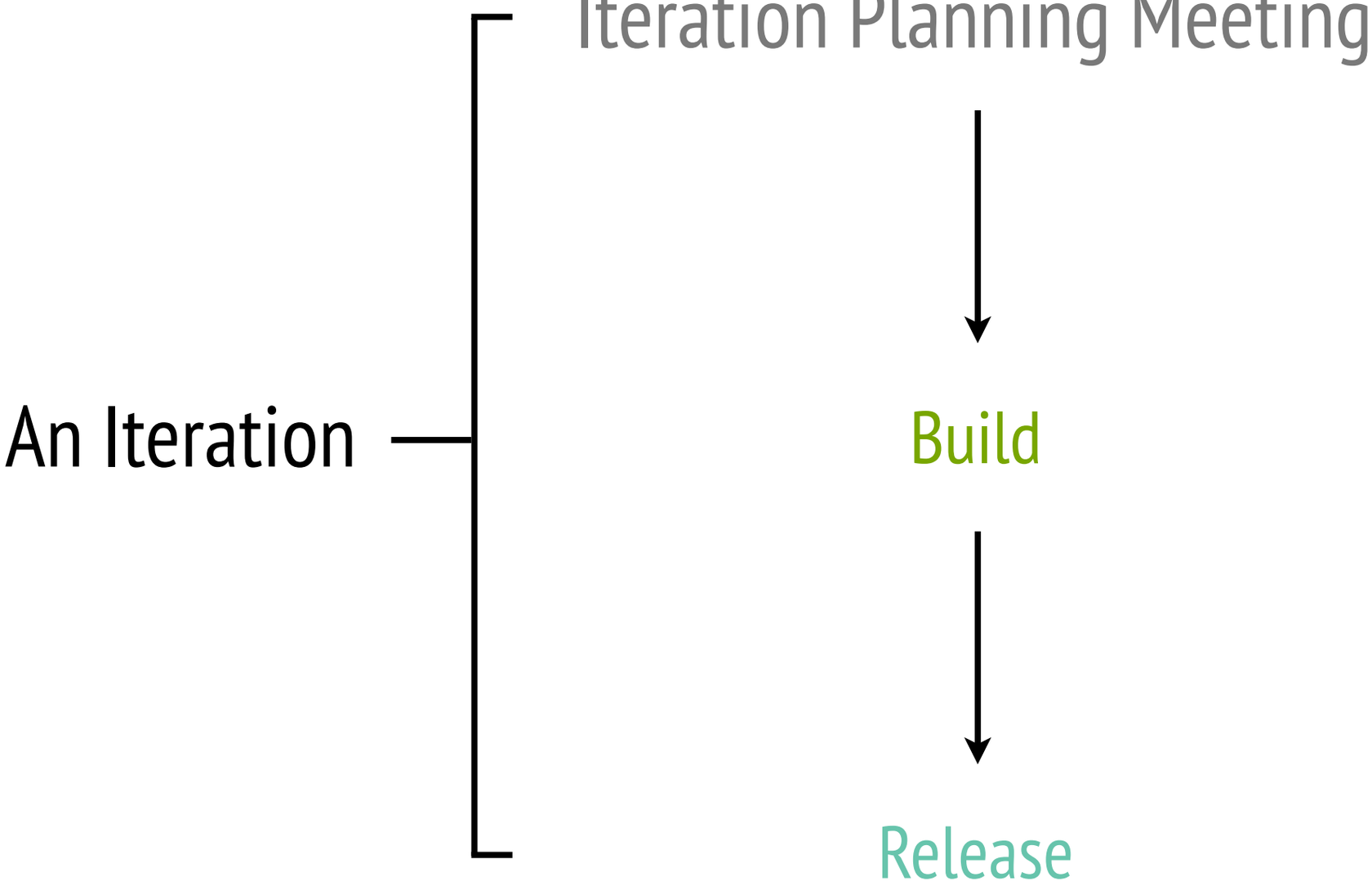
# Agile


Inception
High Level Scope
High Level Design
Detailed Design
Iterations
Estimate
Build
UAT
Time to Ship

# Agile


Inception
Iteration 1
Iteration 2
Iteration 3 ... Iteration n
Retrospective
(every 2-4 iterations)
Release Planning Meeting
(every 12-24 iterations)

# Iteration



# Iteration

An Iteration

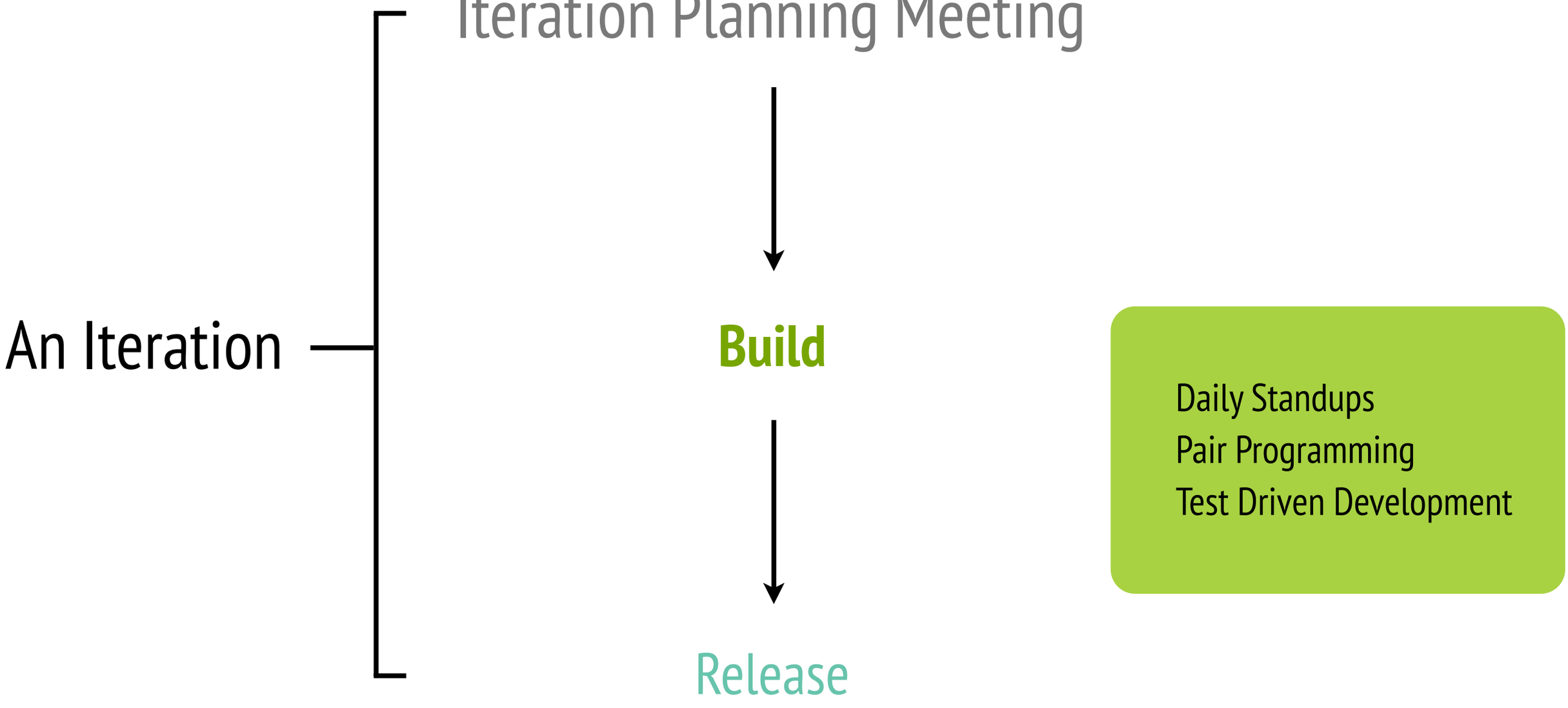
## Iteration Planning Meeting

- Stories Generation
- Stories Clarification
- Stories Estimation
- Stories Prioritization

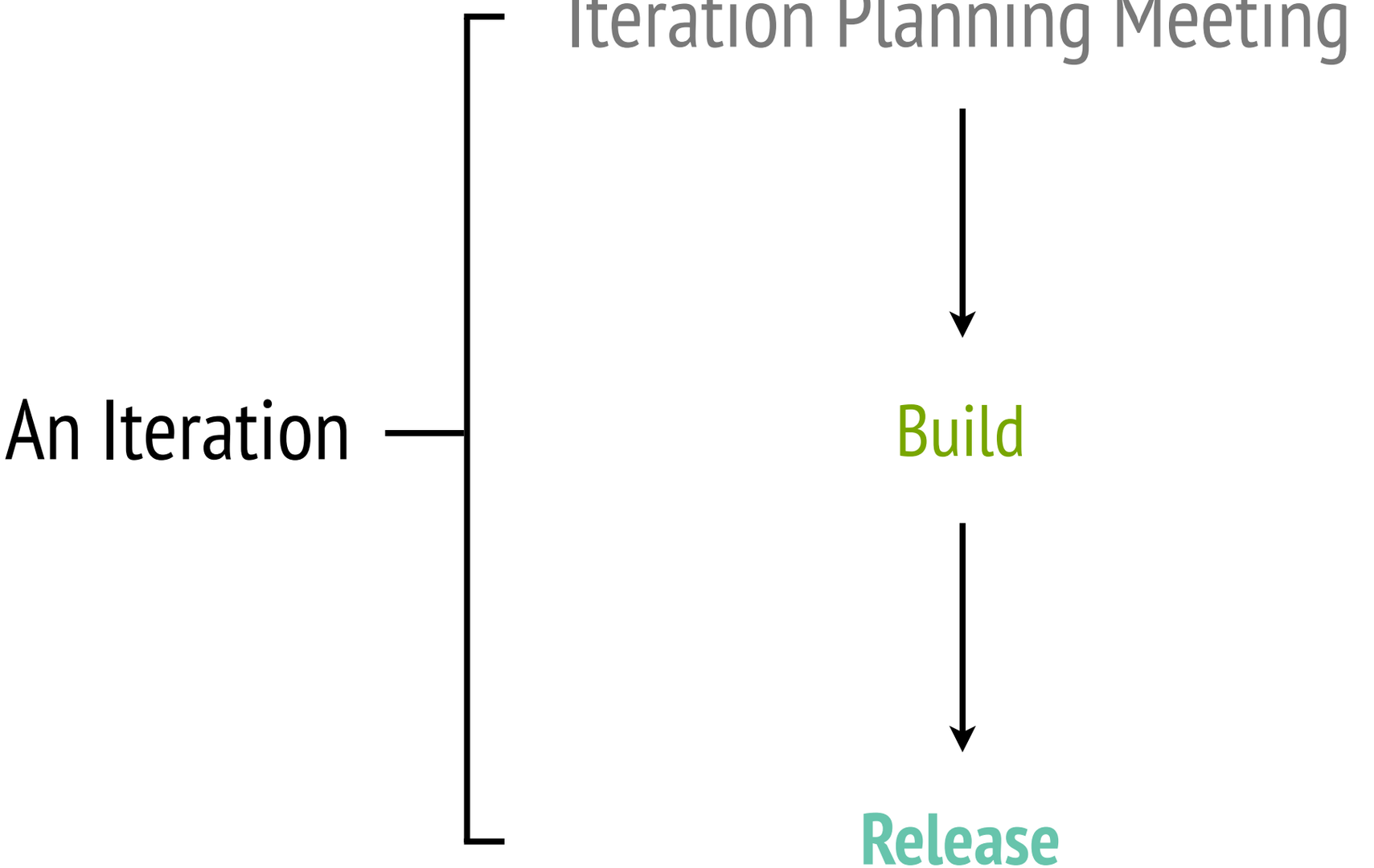
Build

Release

# Iteration


Iteration Planning Meeting
An Iteration
Build
Release
Daily Standups
Pair Programming
Test Driven Development

# Iteration




Continuous Integration
Continuous Deployment

# Iteration

An Iteration

**Iteration Planning Meeting**

Stories Generation
Stories Clarification
Stories Estimation
Stories Prioritization

**Build**

Daily Standups
Pair Programming
Test Driven Development

**Release**

Continuous Integration
Continuous Deployment

# What is..

# Inception

[Management Practice]

- Alignment and Expectation Setting
- Days to Week depending on size of project
- Agenda:
  - Goals (Business, Product, Non-Goals)
  - Risks
  - High Level Stories Estimation and Prioritization
  - Release Planning

# Release Planning Meetings

[Management Practice]

**.. is used to create a release plan, which lays out the overall project. The release plan is then used to create iteration plans for each individual iteration.**

**-http://www.extremeprogramming.org/rules/planninggame.html**

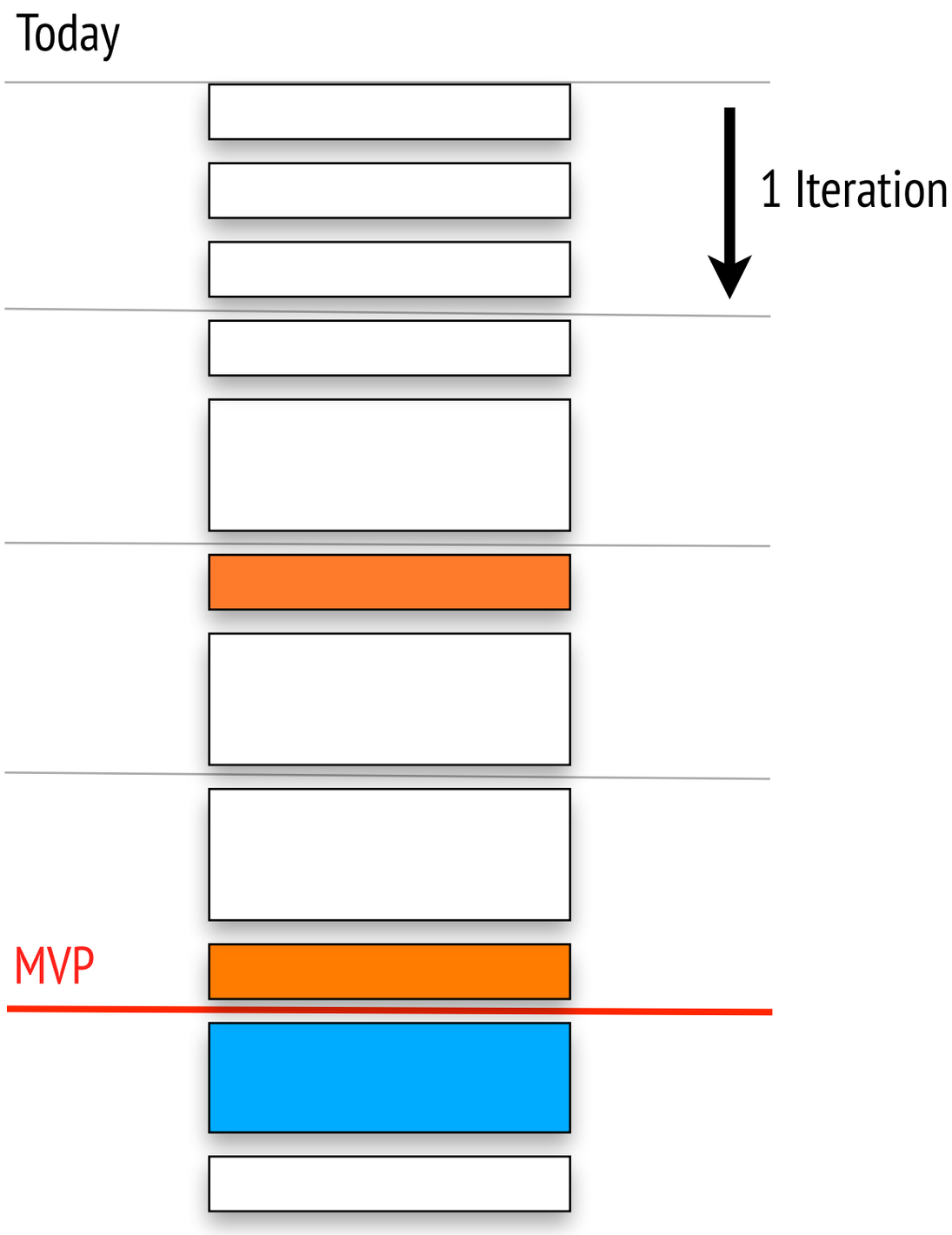
# Release Planning Meetings

[Management Practice]

- Each release is 3 - 6 months
- Business to make decision on MVP, and its features
- Release is planned by Time or Scope
- Project can be quantified by:
  - Scope, Time, Resources and Quality

# Release Planning Meetings

[Management Practice]

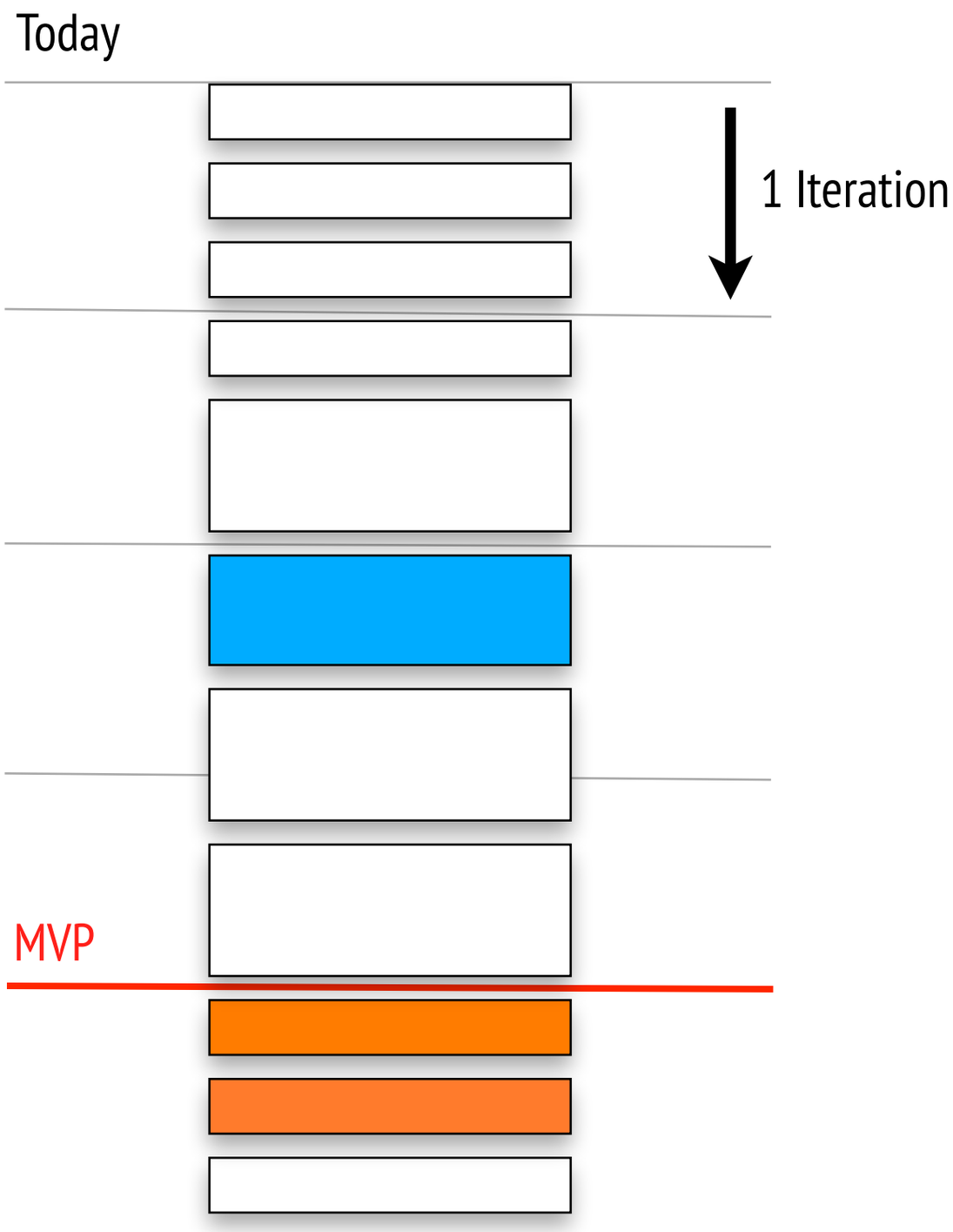



**Estimation**
by Engineers

**Prioritization**
by Product Owner

# Release Planning Meetings

[Management Practice]




**Estimation**
by Engineers

**Prioritization**
by Product Owner

# Iteration Planning Meetings

[Management Practice]

**.. is called at the beginning of each iteration. User stories are chosen for this iteration by the customer in order of the most valuable to the customer first.**

**-http://www.extremeprogramming.org/rules/iterationplanning.html**

# Iteration Planning Meetings

[Management Practice]

- Each iteration is short, 1 - 3 weeks
- Features are prioritized, based on:
  - Business value (usually in financial sense)
  - Effort required
  - Amount and significance of new knowledge gained from feature
  - Risk added / removed

# Retrospective

[Management Practice]

**.. play a very important role in iterative and incremental development. At the end of every iteration a retrospective is held to look for ways to improve the process for the next one.**

**- http://en.wikipedia.org/wiki/Retrospective**

# Retrospective

[Management Practice]

- Reflect and Adapt methods and to help whole team learning
- Benefits of Retrospectives:
  - Productivity
  - Capability
  - Quality
  - Capacity

# Daily Standups

# Daily Standups

[Engineering Practice]

- General project updates
- Which story did you work on yesterday?
- Which story are you going to work on today?
- Blockages?

# Pair Programming

# Pair Programming

[Engineering Practice]

- Benefits of Pair Programming:
  - Focus
  - Knowledge Exchange
  - Collective Code Ownership
  - High Code Quality
  - Engineering Happiness!

http://winstonyw.com/2012/12/02/pair_programming/

# Test Driven Development

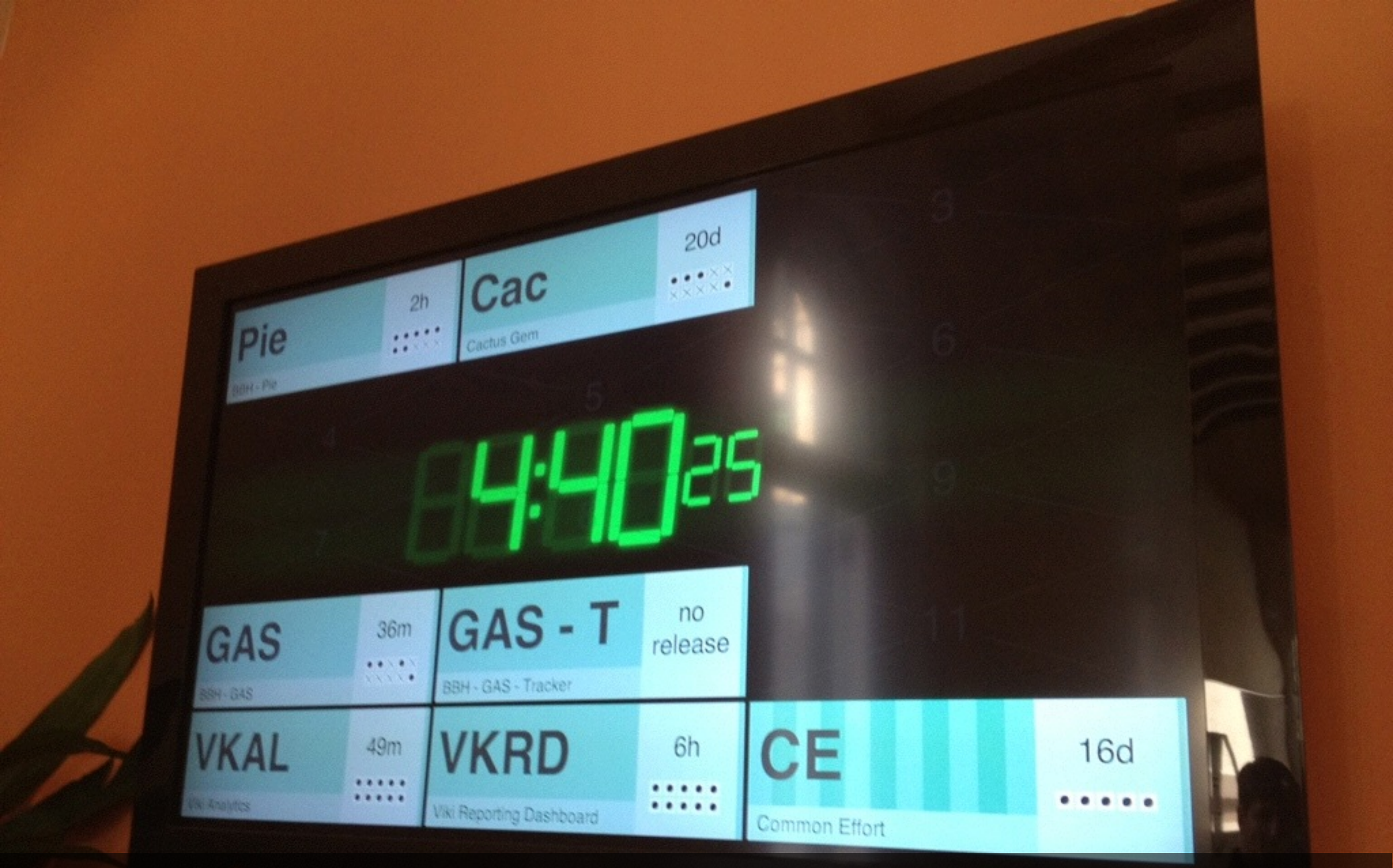
[Engineering Practice]

> **.. when you create your tests first, before the code, you will find it much easier and faster to create your code.**
>
> **- http://www.extremeprogramming.org/rules/testfirst.html**

# Test Driven Development

[Engineering Practice]

- Red, Green, Refactor
- Benefits of Test Driven Development:
  - Quality
  - Accountability
  - Maintainability
  - Live Documentation




# Continuous Integration/Deployment

# Continuous Integration/Deployment

[Engineering Practice]

**.. often avoids diverging or fragmented development efforts, where developers are not communicating with each other about what can be re-used, or what could be shared.**

**- http://www.extremeprogramming.org/rules/integrateoften.html**

# WHAT SETS AGILE APART FROM WATERFALL

# High Level Stories Estimation and Prioritization

# What is a User Story?

**.. one or more sentences in the everyday or business language of the end user or user of a system that captures what a user does or needs to do as part of his or her job function.**

**- http://en.wikipedia.org/wiki/User_story**

# Structure of a User Story (1)

As a
**<user role>**,
I want
**<capability>**,
so that
**<business value>**.

# Structure of a User Story (2)

**Acceptance Criteria:**

1. <do this>
2. <do that>

...

n. <expected result>

# Example of a User Story (1)

As a
**Facebook User**,
I want
**to sign up with my FaceBook account**,
so that
**I can start using the service instantaneously**.

# Example of a User Story (2)

**Acceptance Criteria:**

1. Go to homepage
2. Click on “Login with Facebook”
3. Wait to be redirected to Facebook Login page
4. Login with your Facebook credentials
5. You will be redirected to homepage
6. You should see “Welcome John” in header

# HOW TO

# ESTIMATE

# STORIES?

# Techniques

# ANALOGY.

# EXPERT ADVICE.

# DIVIDE AND CONQUER.

# TECHNIQUES

# NOT

# ENOUGH

$$\frac{\text{TOTAL STORY POINTS}}{\text{VELOCITY}} = \text{DURATION}$$

# Relative Sizing

# Relativity

# Let's Estimate

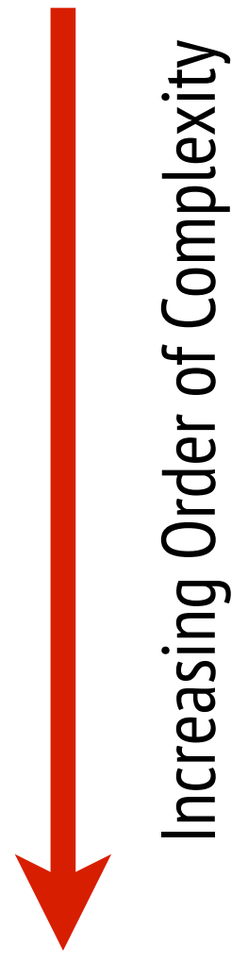
**Estimate the size of these balls. Pick from S, M, L or XL.**

| Type | Size |
|---|---|
| Golf Ball | S |
| Basketball | XL |
| Soccer Ball | L |
| Ping Pong Ball | S |
| Tennis Ball | M |

# Size Chart and Story Points

# Size Chart

| Size | Points | Points |
|---|---|---|
| S | 1 | 1 |
| M | 2 | 2 |
| L | 4 | 3 |
| XL | 8 | 5 |
| XXL | 16 | 8 |



$$\frac{\text{TOTAL STORY POINTS}}{\text{VELOCITY}} = \text{DURATION}$$

# Velocity

# Velocity

- Story Points per Iteration (Throughput)
- Derived based on:
  - Estimation
  - Historical Data
- Depends on:
  - Team Members (skills)
  - Team Size

$$\frac{\text{TOTAL STORY POINTS}}{\text{VELOCITY}} = \text{DURATION}$$

# ESTIMATE WITH

# PLANNING

# POKER

ENGINEERS

# ESTIMATE

Product Owner

# PRIORITIZE

AGILE IS

# CONSTANT

# FEEDBACK

# Thank You