

WHOA!

NOT THE AGILE MANIFESTO!

Agile

And a Current Declarative Development Process Primer

Overview

1. Agile
2. Project Management Basics
3. Current Declara Development Process
4. Agile, Declara, and You!

Agile

Agile



Agile

And, it works!

Wait... really?

So... why Agile?

What is Agile?

- NOT a methodology or a set of proscribed processes
 - Scrum
 - Kanban
 - Extreme Programming
 - Lean Development
 - Crystal
 - Feature Driven Development

What is Agile?

- A set of values
- A conceptual framework
- A philosophy

Agile values

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

Agile values

- **Individuals** and **Interactions** over processes and tools

Agile values

- **Working software** over comprehensive documentation

Agile values

- **Customer collaboration** over contract negotiation

Agile values

- **Responding to change** over following a plan

Agile values

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

So... Why Agile?

(again)



What the Customer Described.



What got budgeted.



What the Engineer Designed.



How Manufacturing Installed it.



What Marketing Advertised.



What got documented.



What the Customer finally received.



What the customer was billed for.



What the customer actually wanted.

Why Agile?

Agile vs. Waterfall

Netpax and Netflix: a made-up case study

Discuss: Why Agile?

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

Project Management

Project Management

The Basics

Project Management

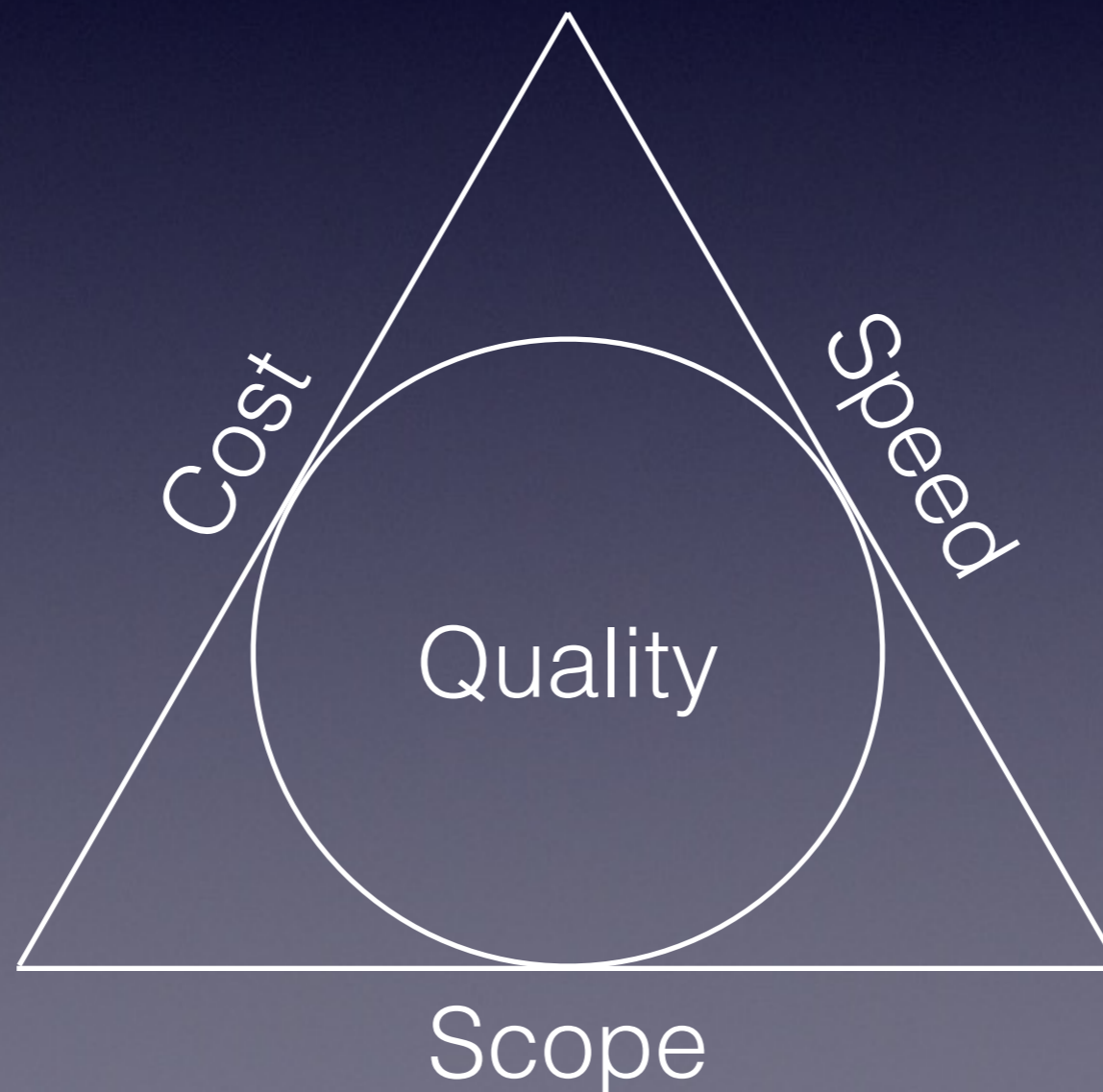
The Basics

Triple Constraint (aka the Triangle of Truth)

Project Management

The Basics

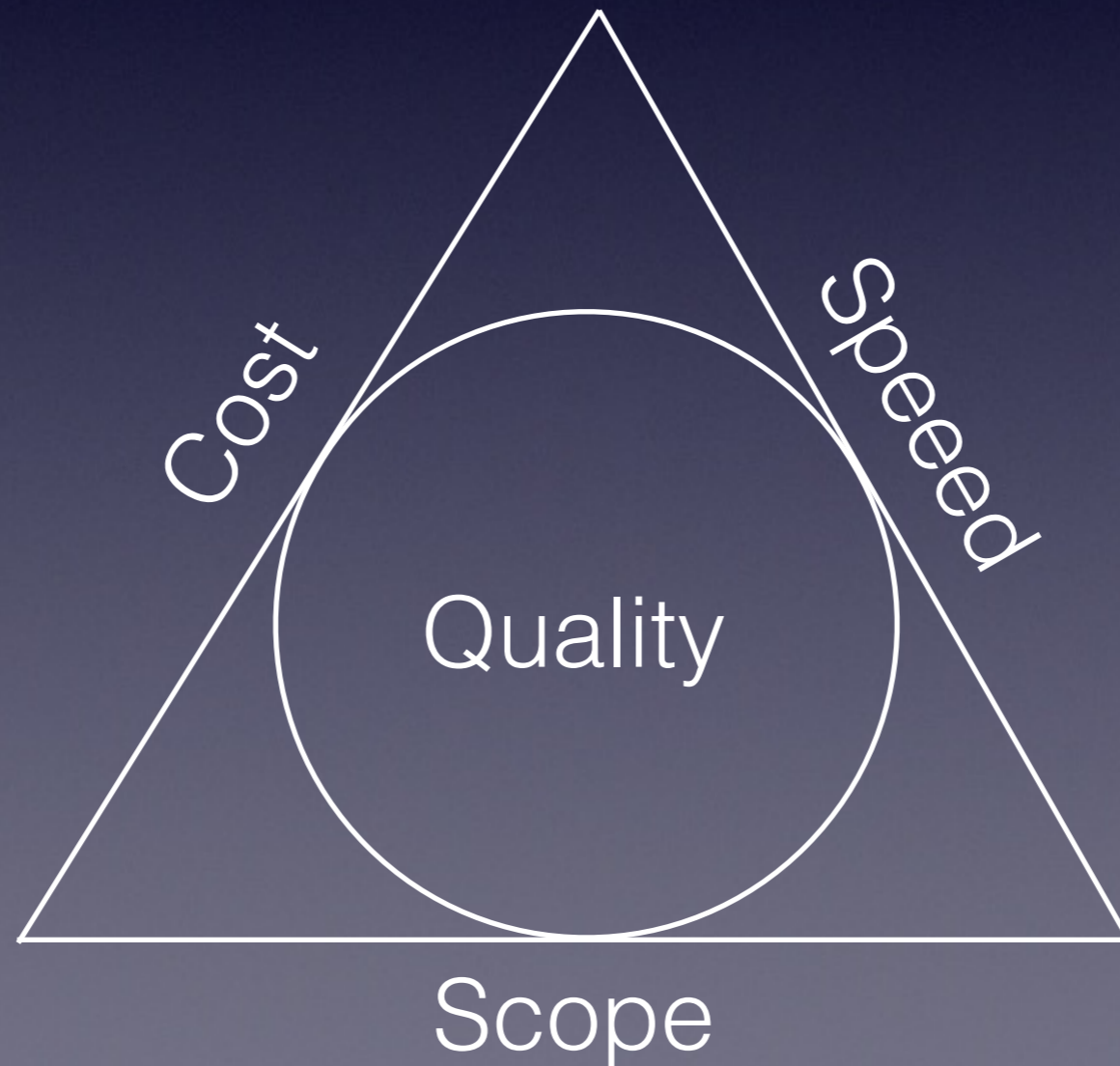
Triple Constraint (aka the Triangle of Truth)



Project Management

The Basics

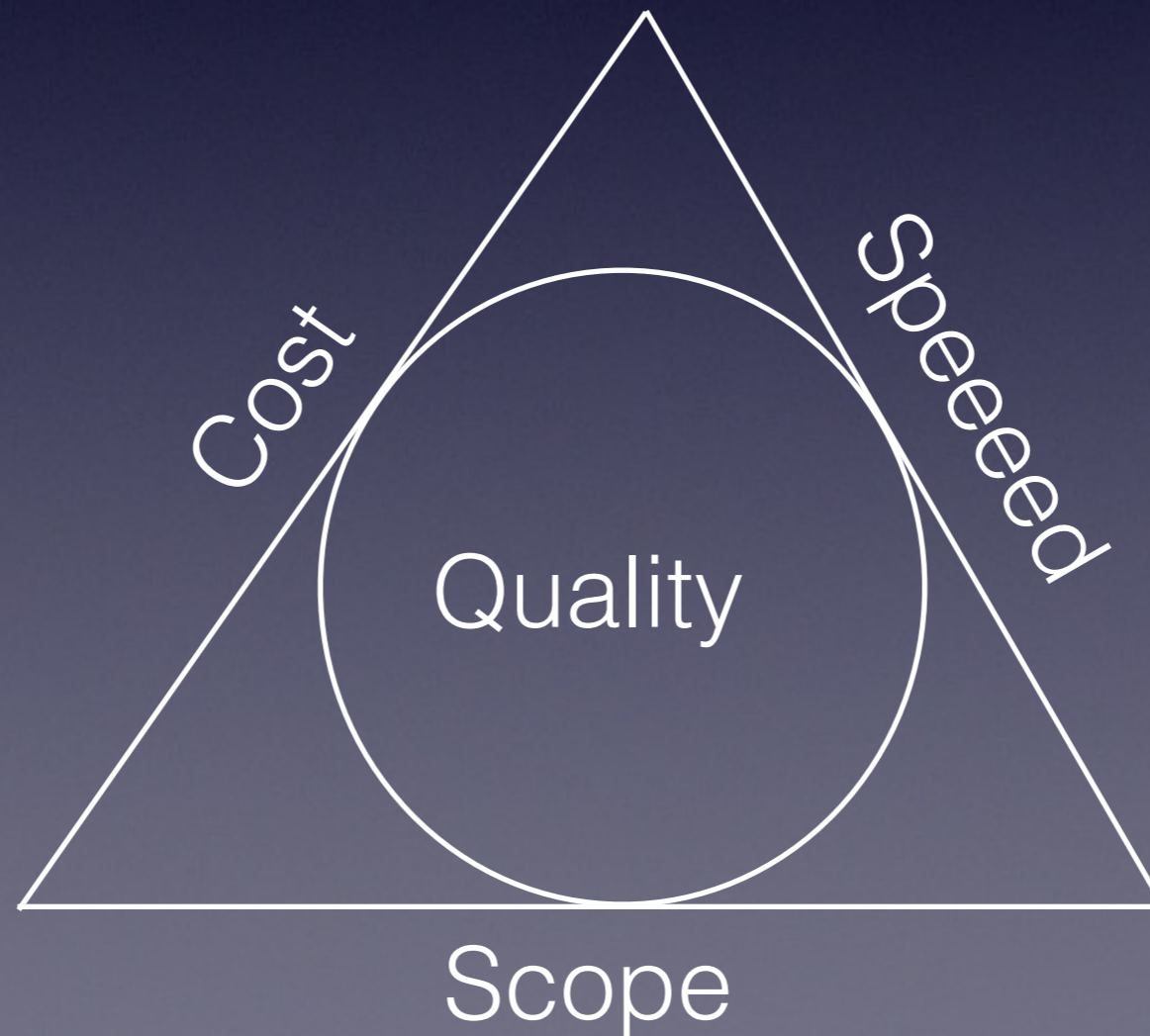
Triple Constraint (aka the Triangle of Truth)



Project Management

The Basics

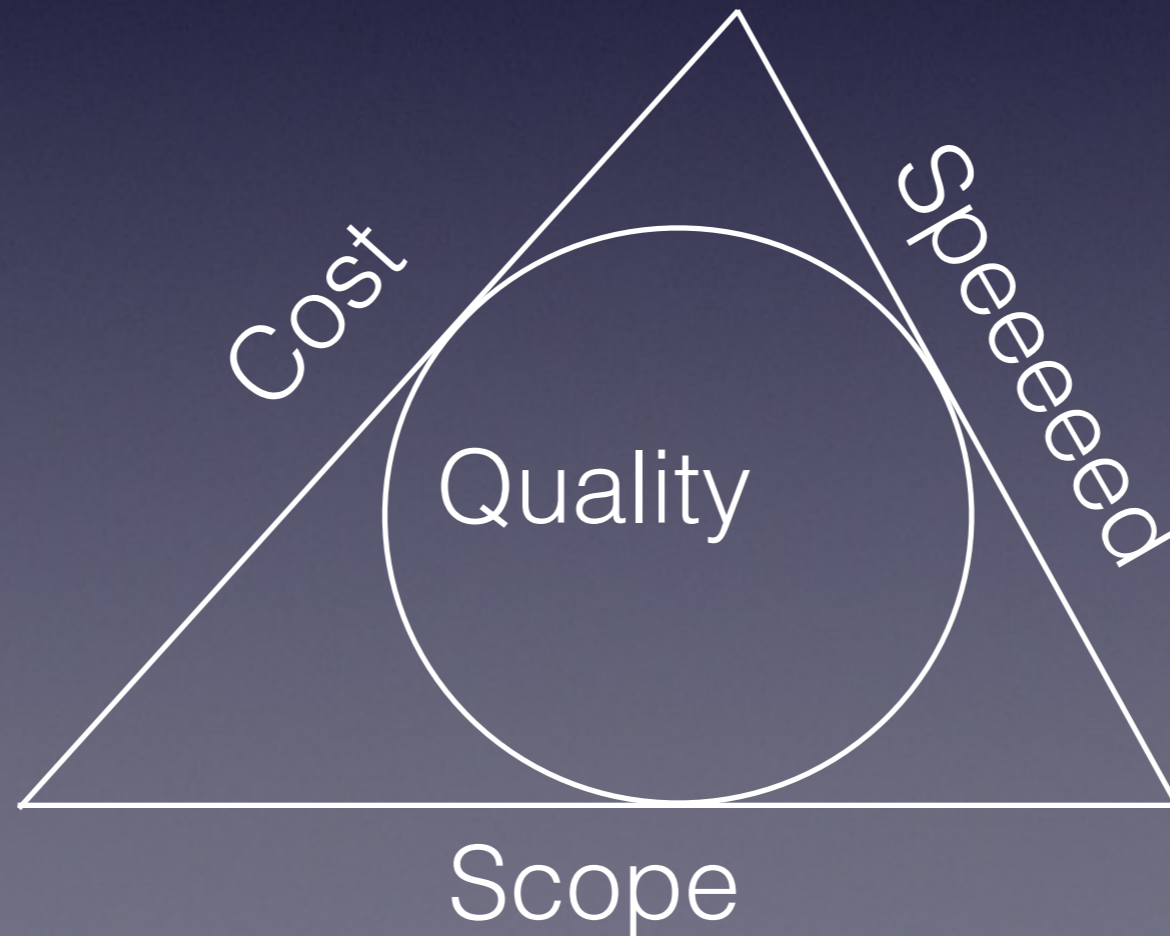
Triple Constraint (aka the Triangle of Truth)



Project Management

The Basics

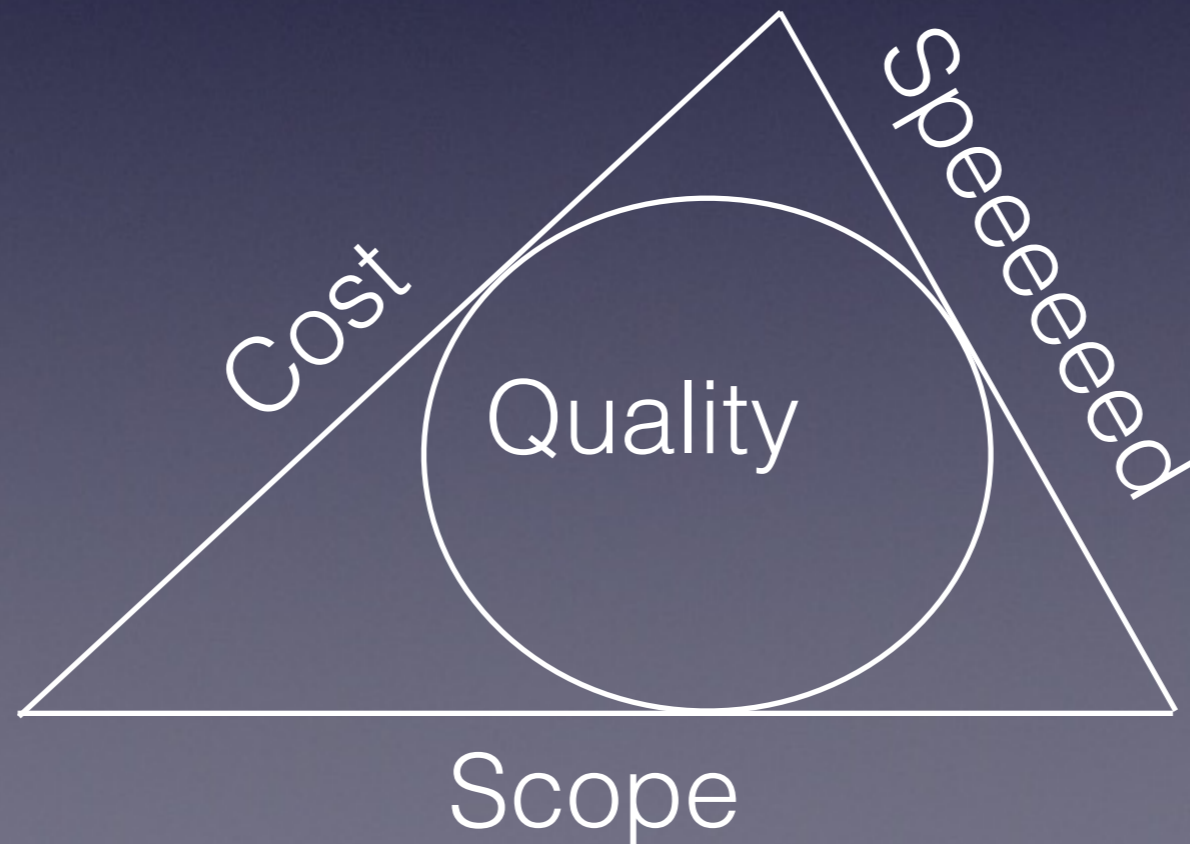
Triple Constraint (aka the Triangle of Truth)



Project Management

The Basics

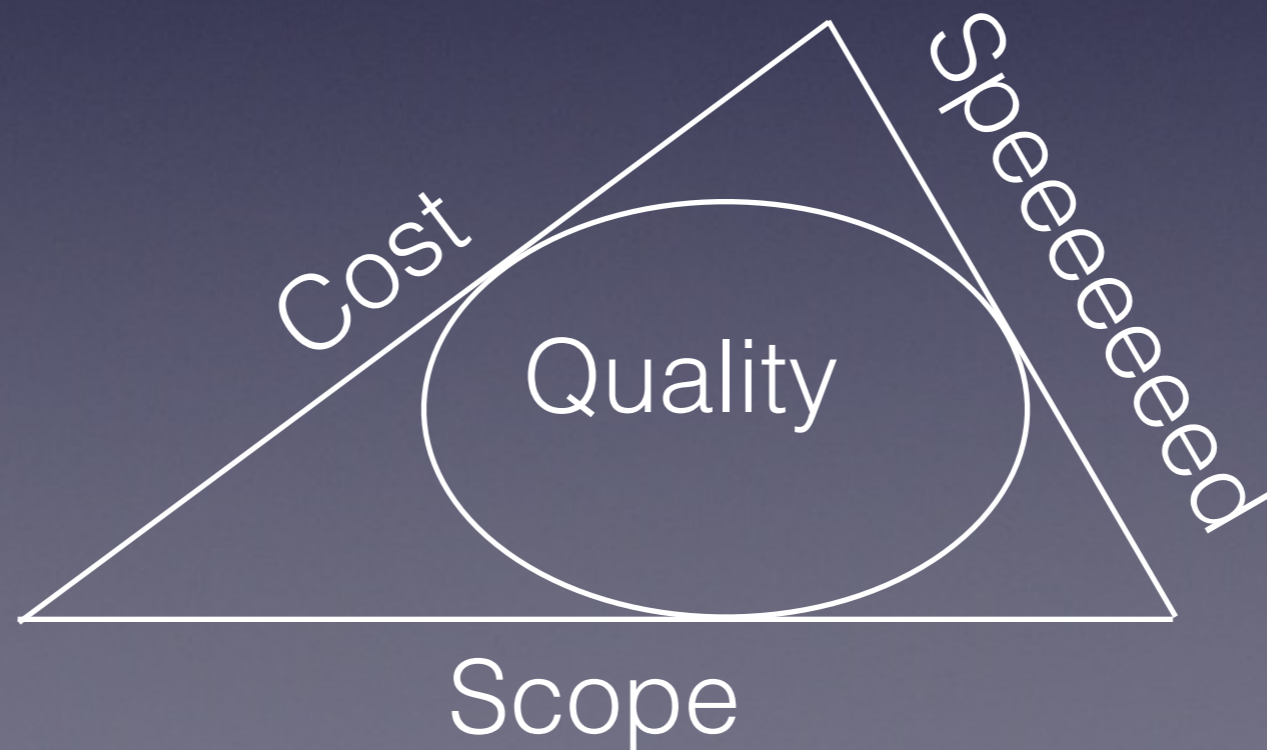
Triple Constraint (aka the Triangle of Truth)



Project Management

The Basics

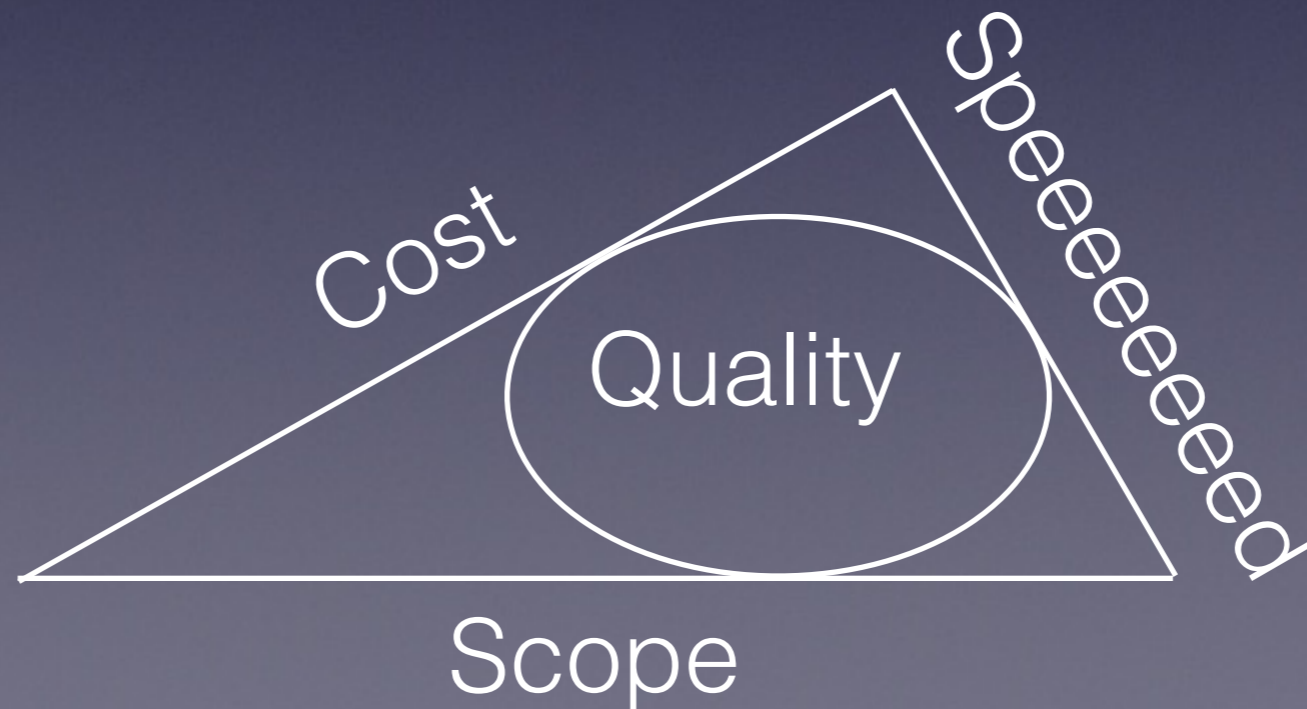
Triple Constraint (aka the Triangle of Truth)



Project Management

The Basics

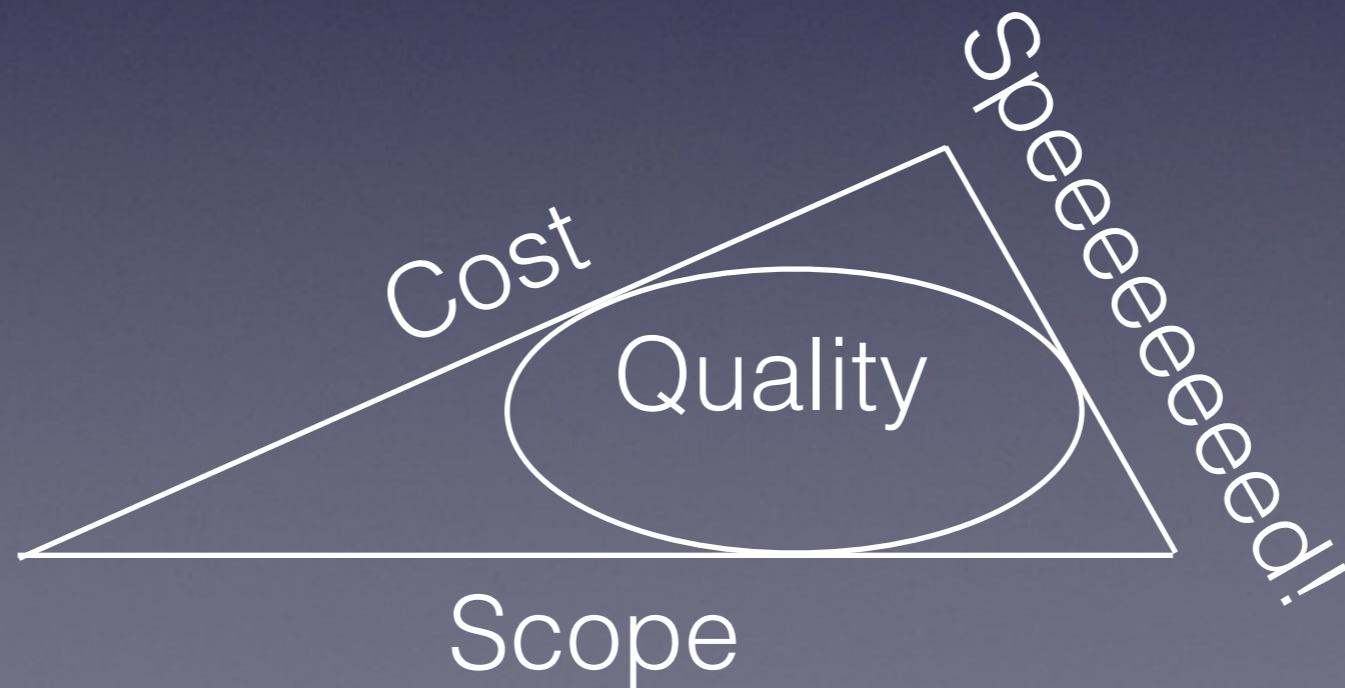
Triple Constraint (aka the Triangle of Truth)



Project Management

The Basics

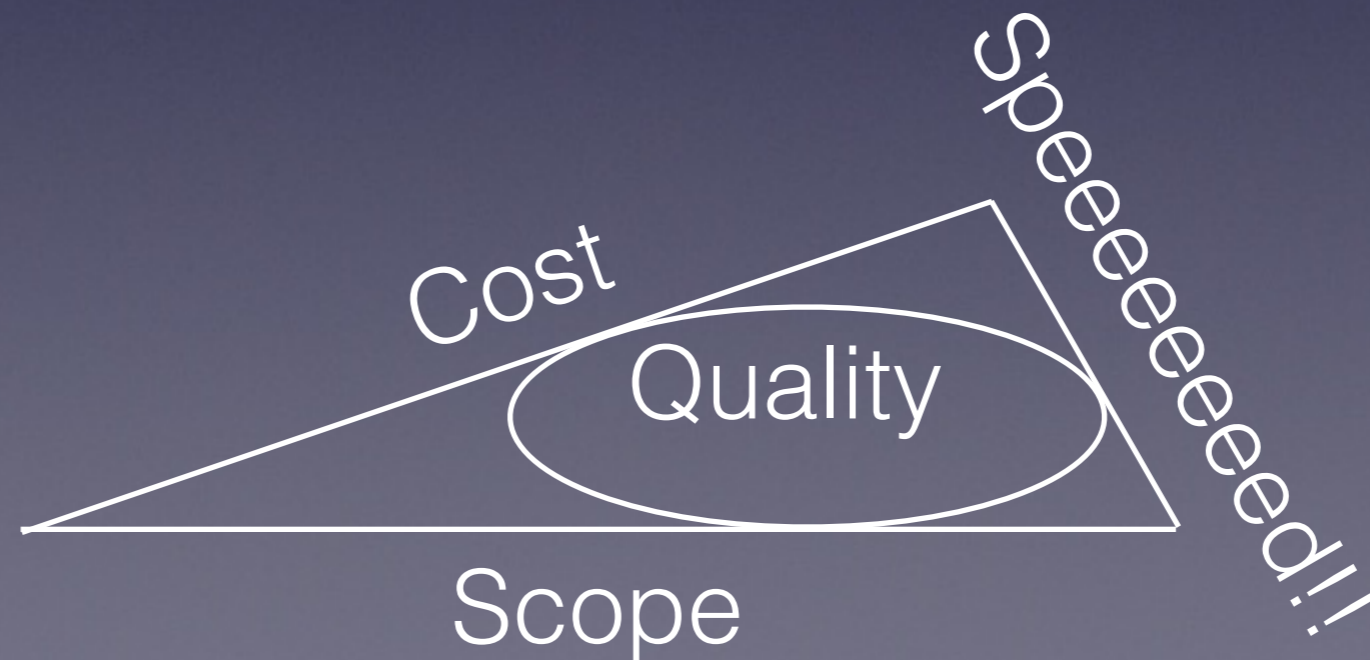
Triple Constraint (aka the Triangle of Truth)



Project Management

The Basics

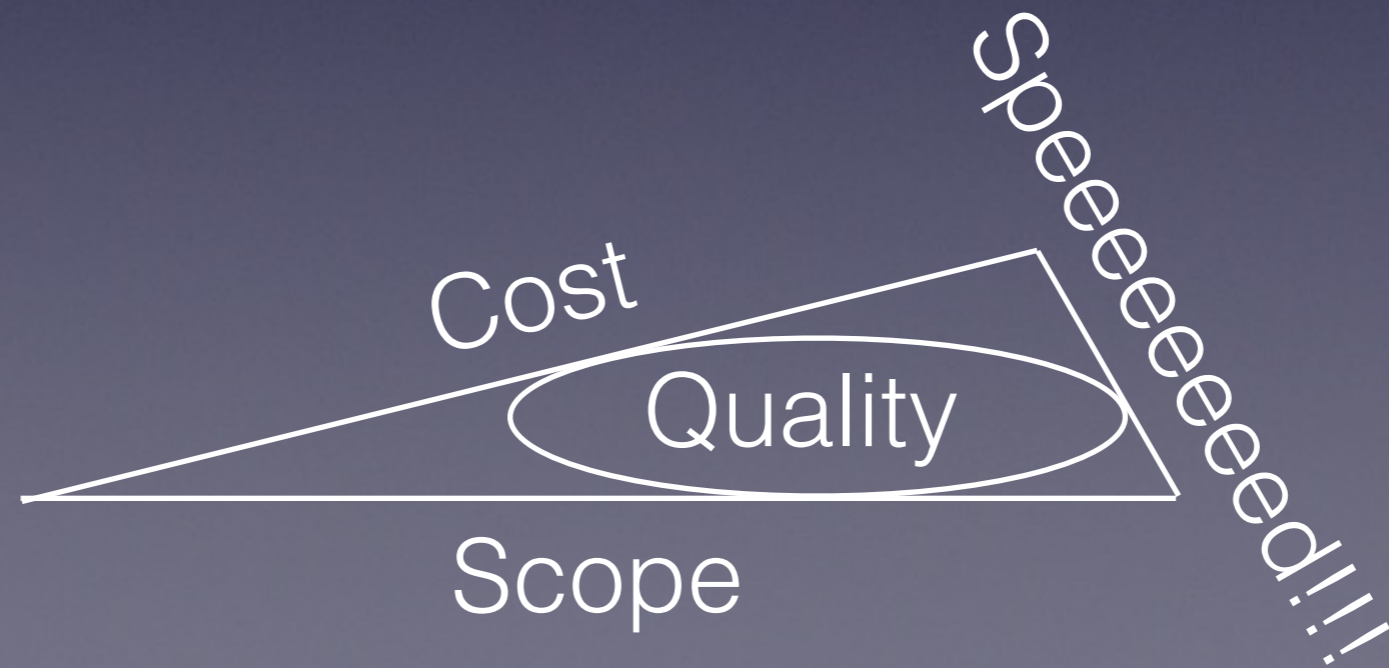
Triple Constraint (aka the Triangle of Truth)



Project Management

The Basics

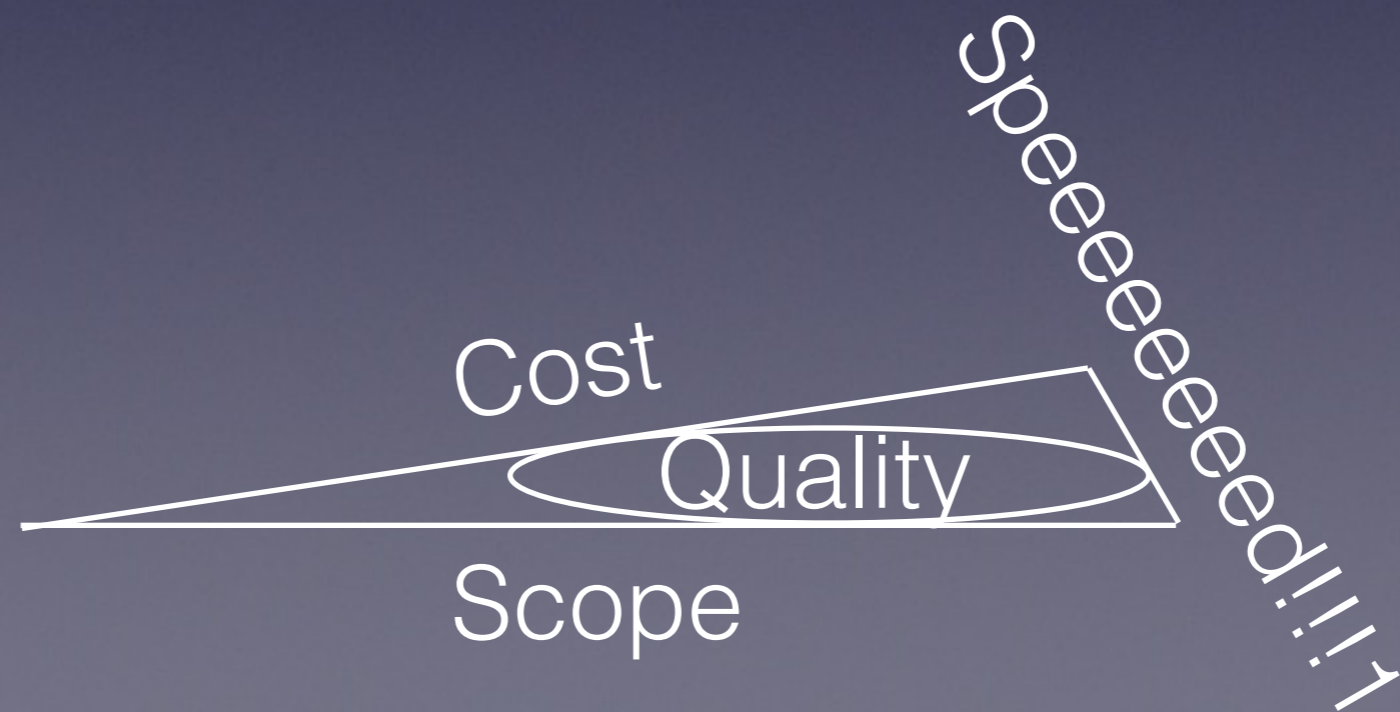
Triple Constraint (aka the Triangle of Truth)



Project Management

The Basics

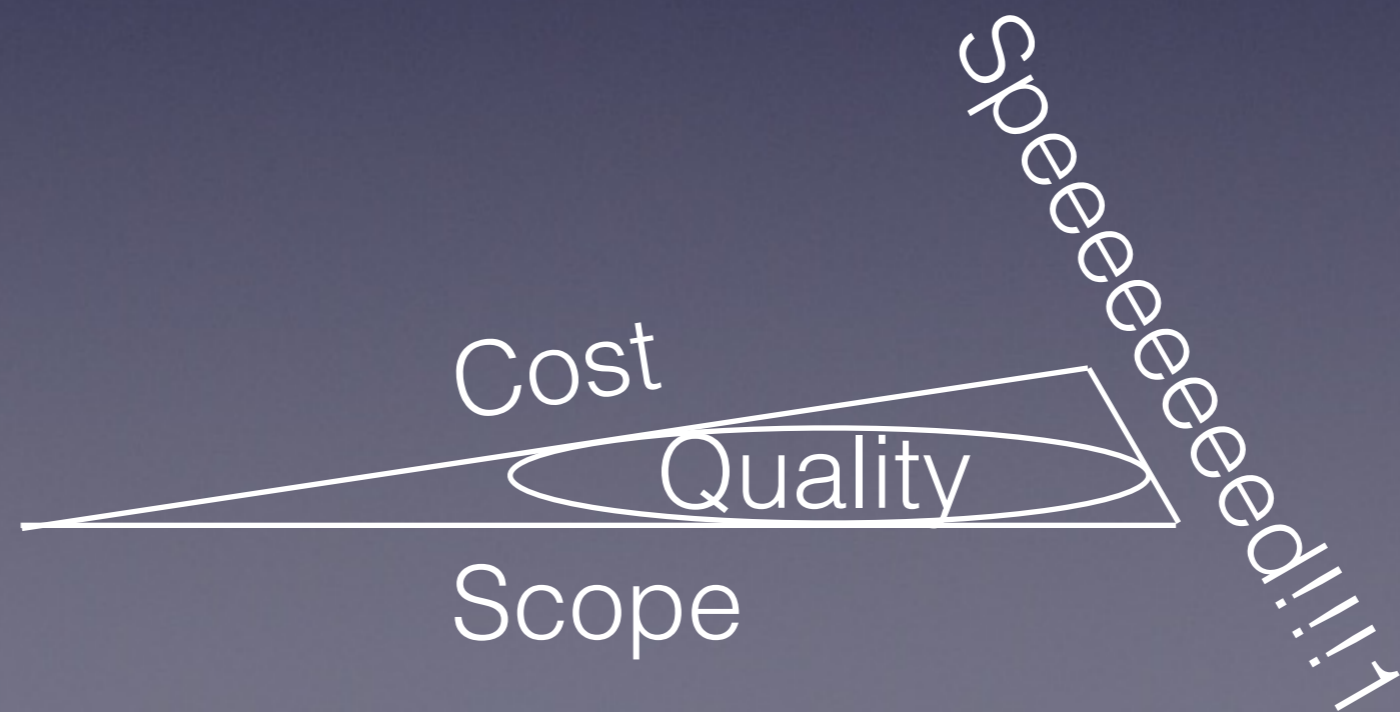
Triple Constraint (aka the Triangle of Truth)



Project Management

The Basics

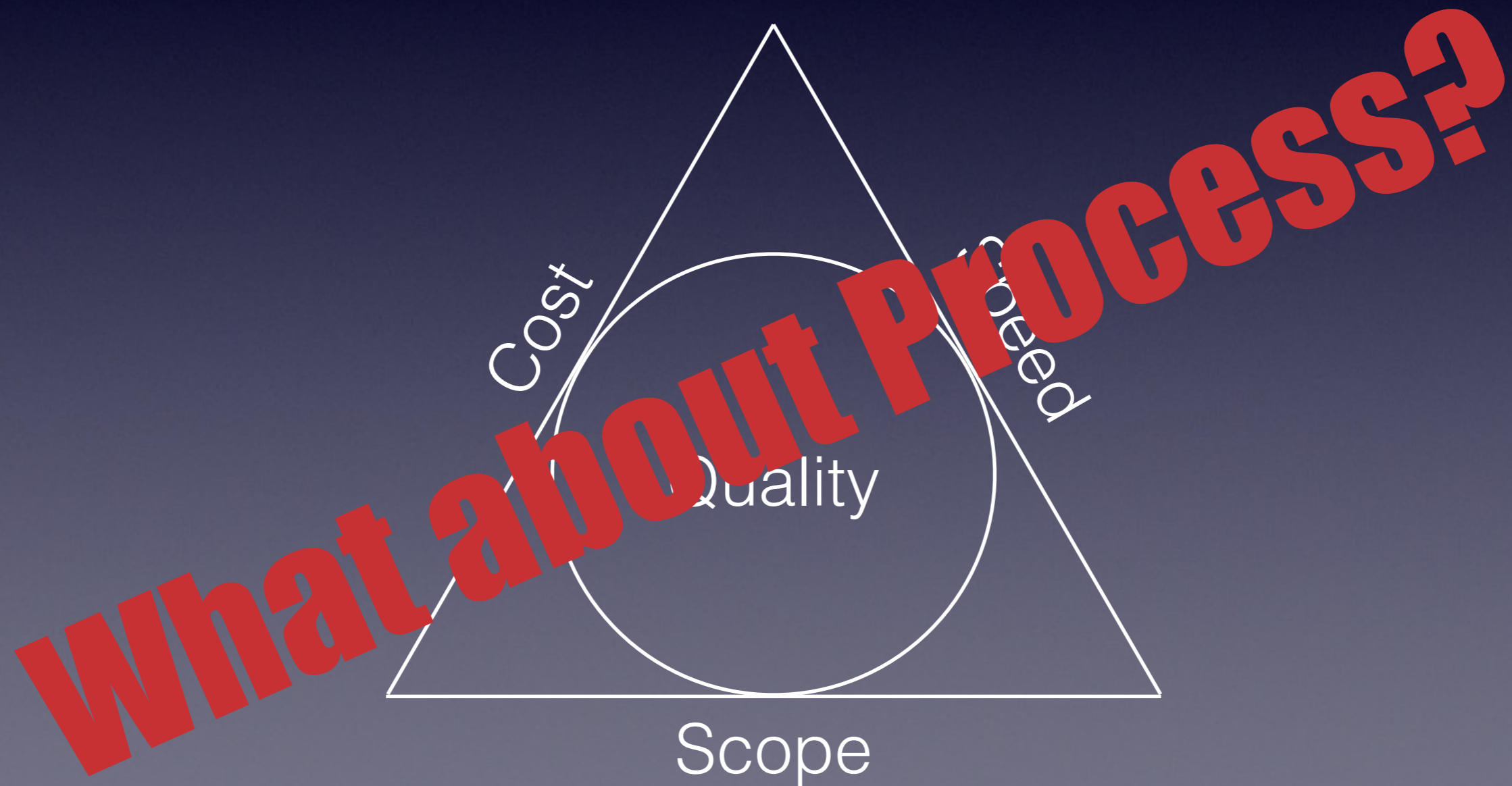
Triple Constraint (aka the Triangle of Truth)



Project Management

The Basics

Triple Constraint (aka the Triangle of Truth)



Project Management

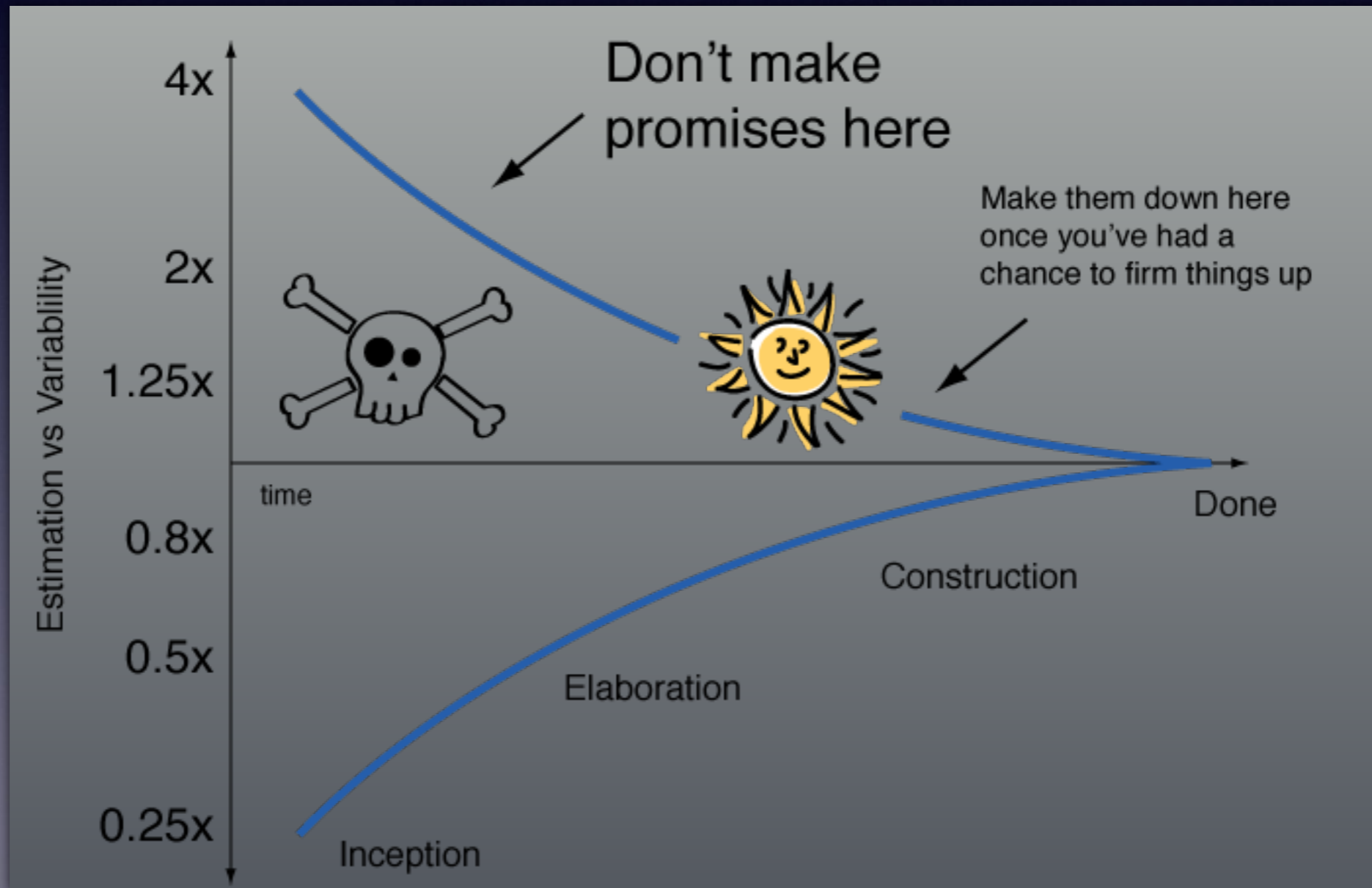
The Basics

Cone of Uncertainty

Project Management

The Basics

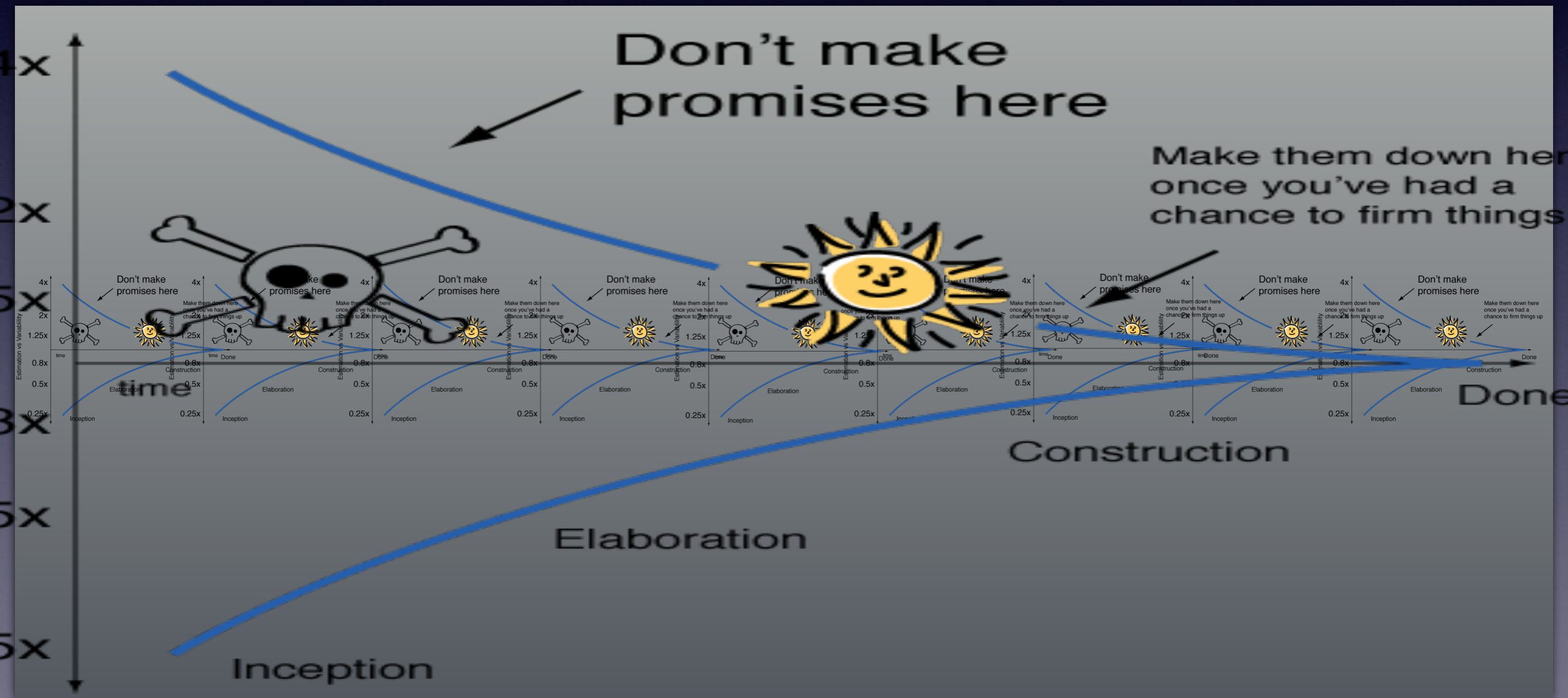
Cone of Uncertainty



Project Management

The Basics

Cone of Uncertainty



Project Management

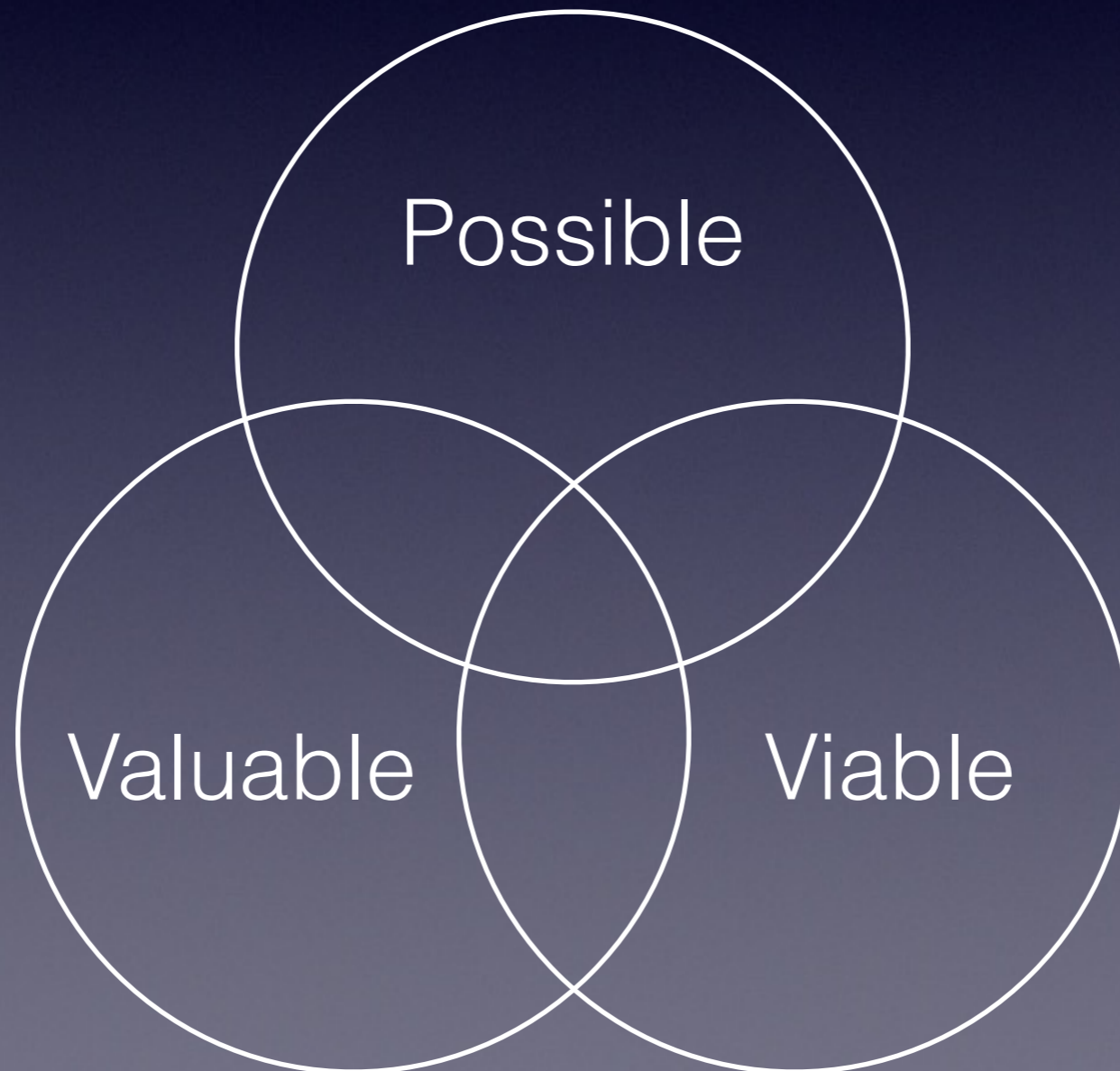
The Basics

Possible, Valuable, Viable

Project Management

The Basics

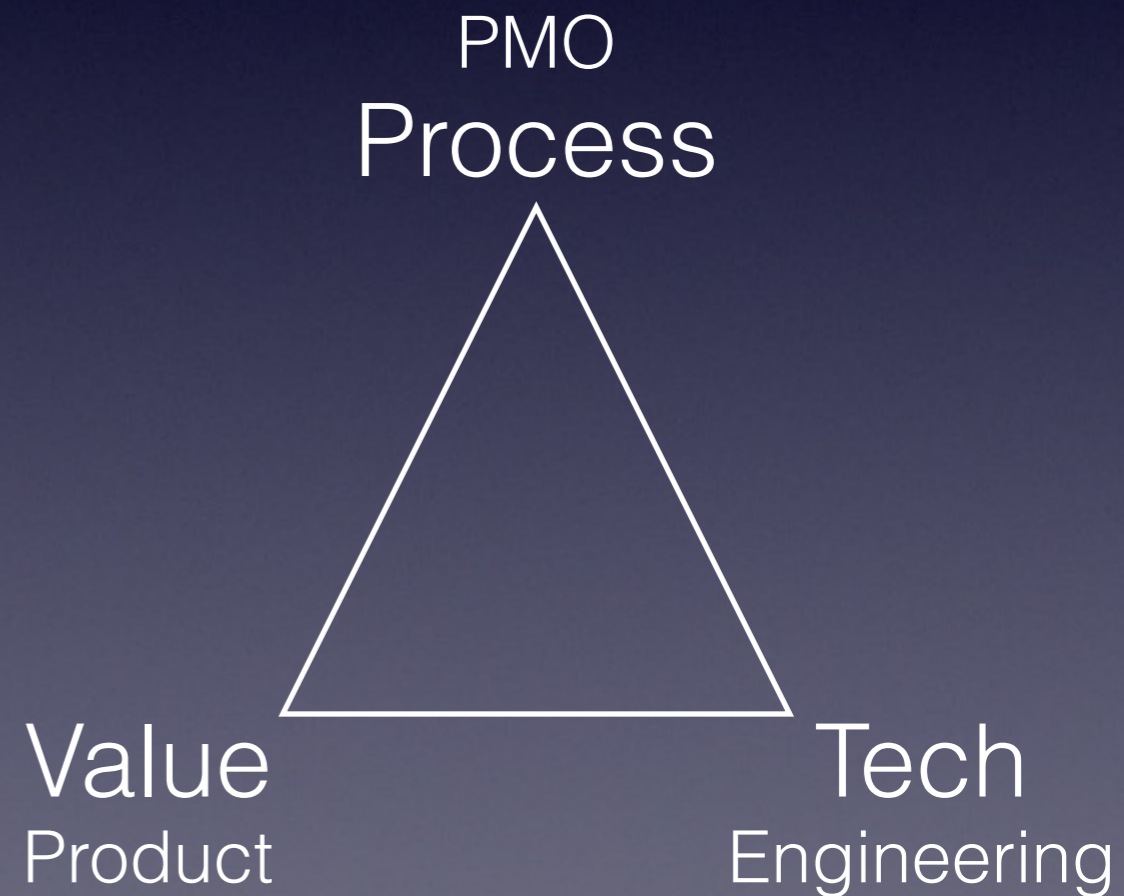
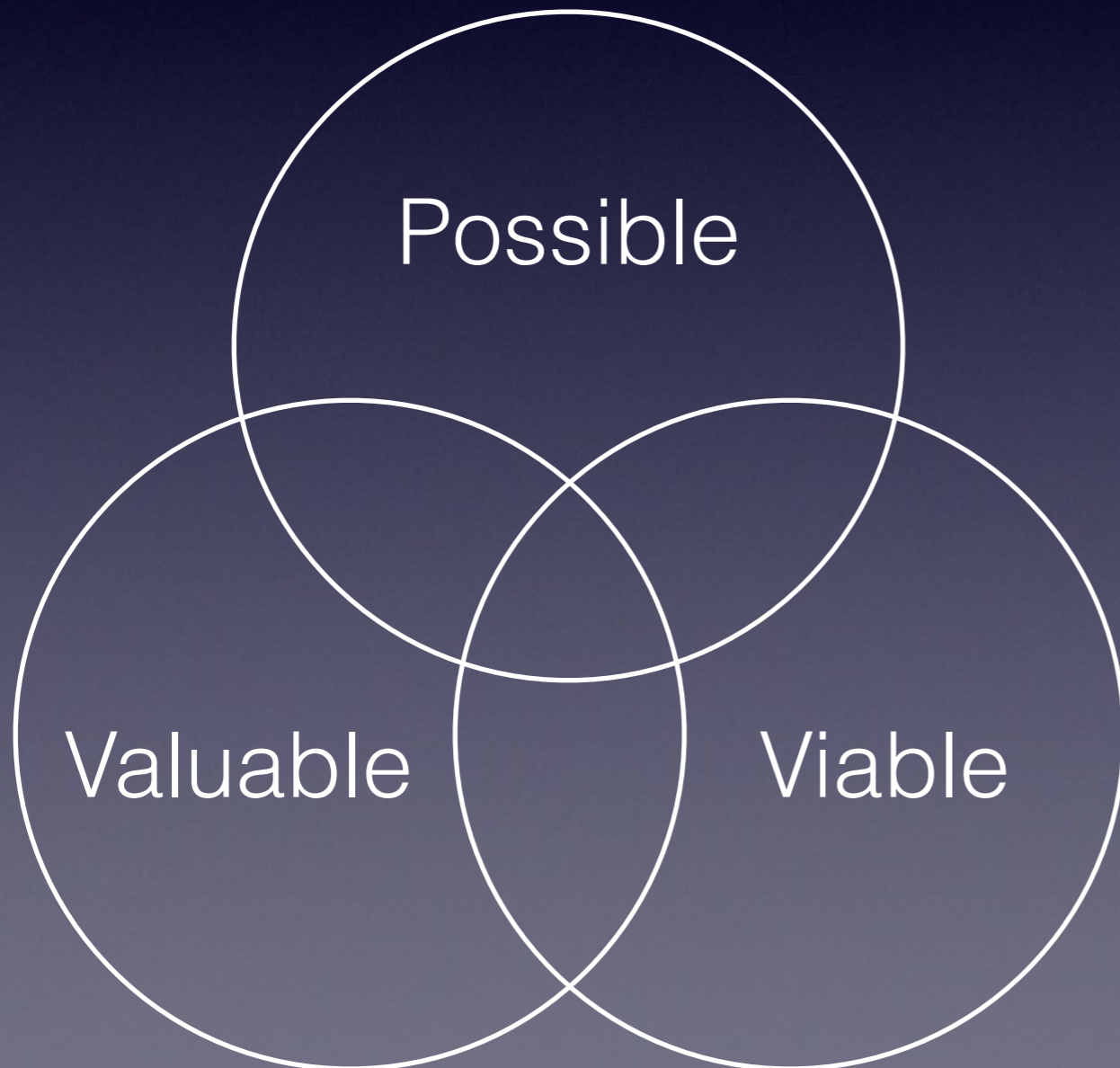
Possible, Valuable, Viable



Project Management

The Basics

Possible, Valuable, Viable



Declara Development Process

Declara Development Process



Declara Development Process



Declara Development Process



Declarara Development Process



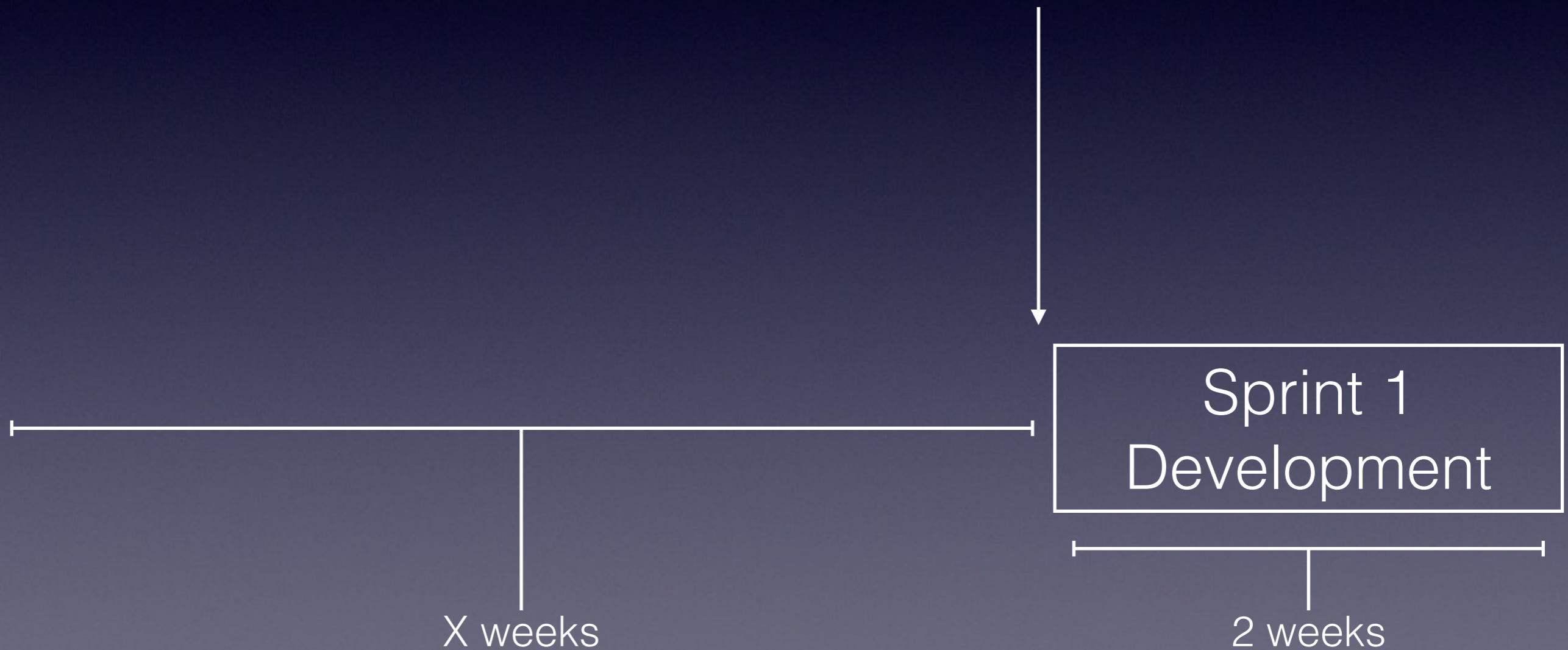
Agile promotes sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Declara Development Process



Declarara Development Process

T - 1 day: Sprint Planning

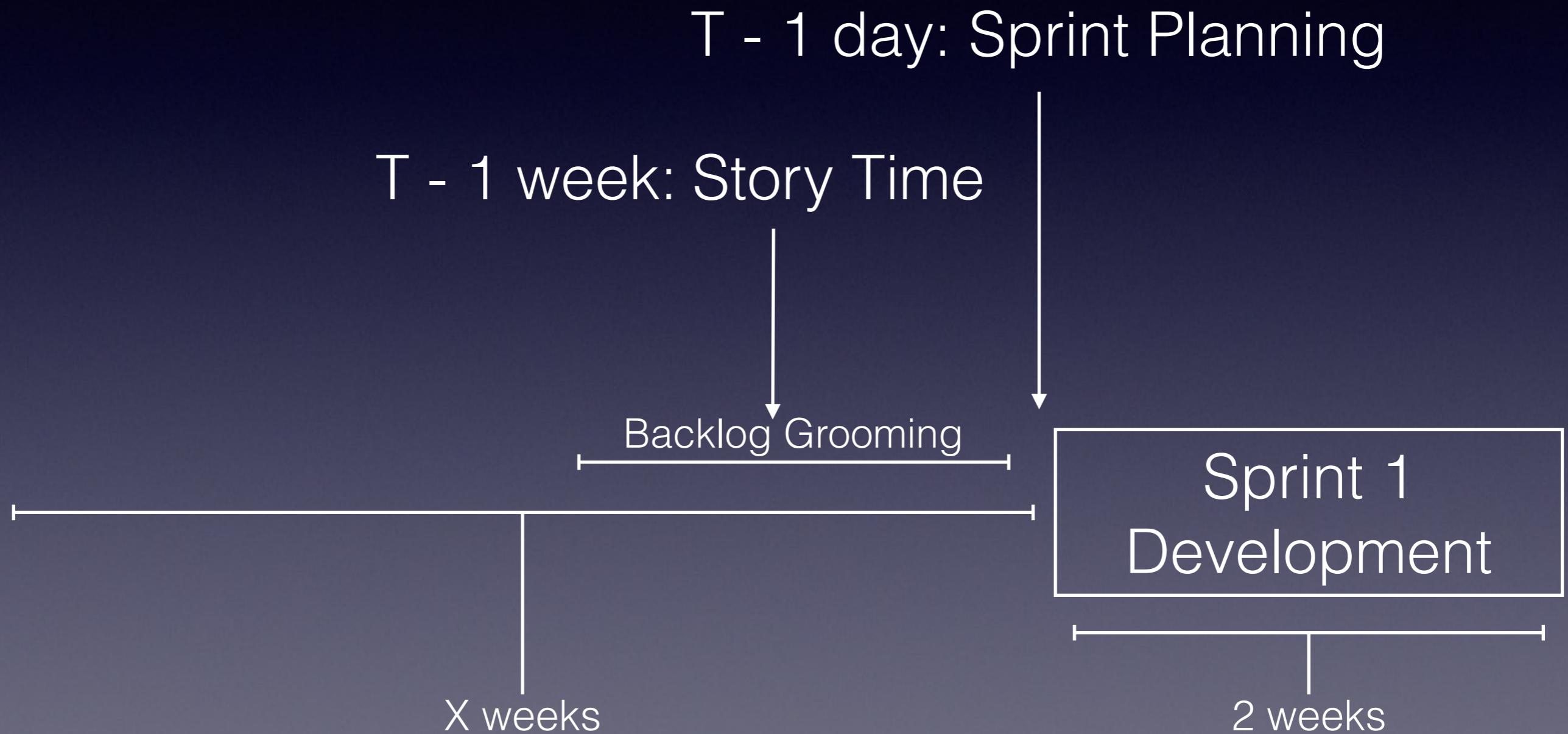


Declarative Development Process

Sprint Planning

- Teams size any remaining unsized stories
- Teams commit to stories that have been prioritized
 - This is based on past velocities and team agreement
- The result is a list of work that the team believes they can complete in one sprint, barring any unforeseen events

Declara Development Process



Declarative Development Process

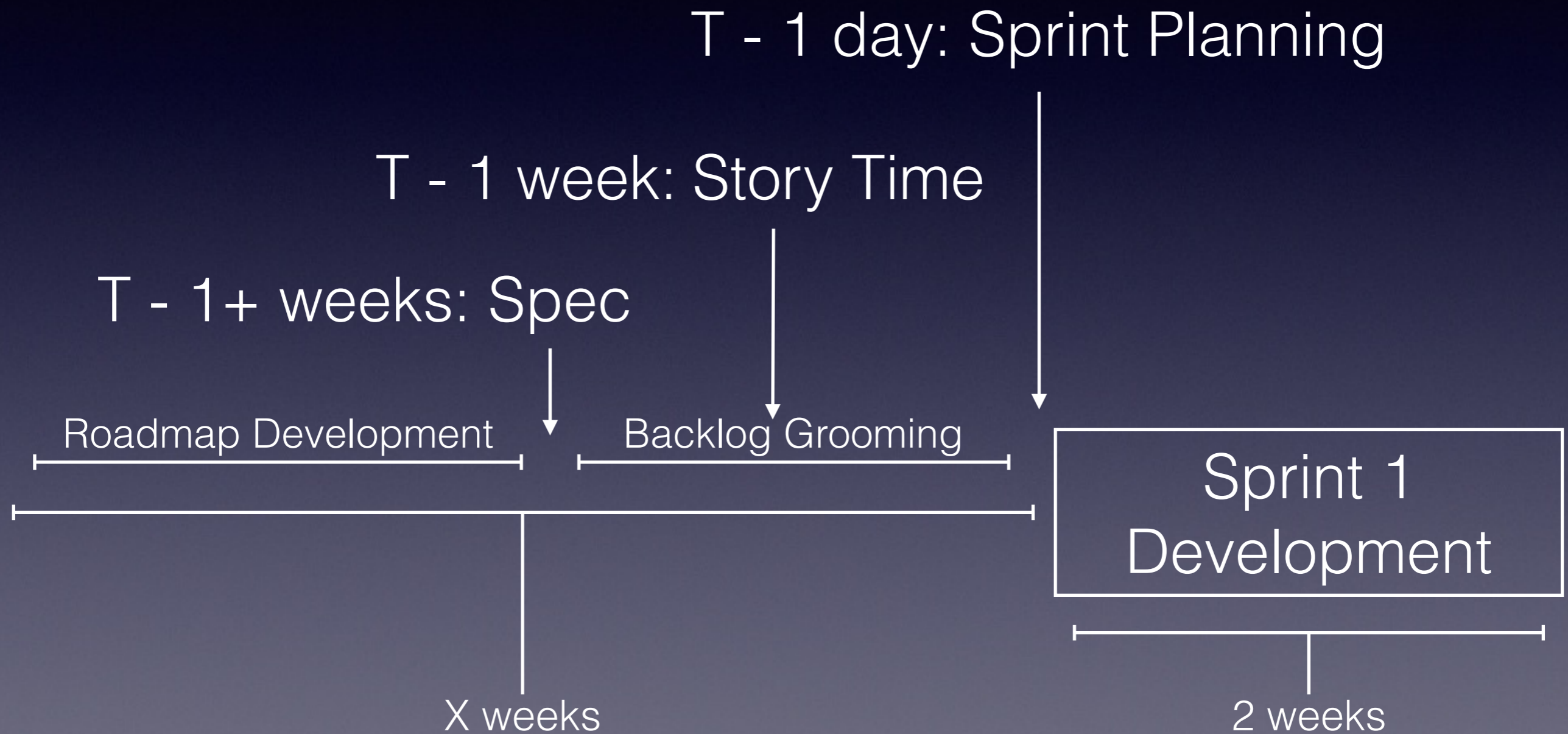
Backlog Grooming

- More-or-less constant process after requirements have been gathered
- Consists of prioritization of all desired features
- Items on the backlog will probably (but not necessarily) get built

Story Time

- A meeting that happens once each sprint
- This is the time for Product to share what they would like the platform to do, and for Engineering to ask questions, give feedback, and quickly size each story (in terms of complexity)

Declara Development Process



Declara Development Process

Roadmap

- A very high level list of desired deliverables
- Inputs: Sales (potential customers), Client Ops (current customers), Product, Other
- New items go through Product
- Items may get into the backlog (but also may not)

Spec

- A document that Product creates
- Contains user stories (as a... I want... so that...)
- These are the documents that are used for story time
- JIRA tickets are created from these specs

Declara Development Process

Review

T - 1 day: Sprint Planning

T - 1 week: Story Time

T - 1+ weeks: Spec

Roadmap Development

Backlog Grooming

Sprint 1
Development

X weeks

2 weeks

Now you know!



What's the other half?

Support Declara in being more Agile!

Individuals and interactions over processes and tools

- Pertinent Agile Principles:
 - Business people and developers must work together daily throughout the project.
 - The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Human-centric philosophy:
 - Work WITH people; don't just occupy the space next to them. Talk to someone on the ground regularly!
 - People are cool! (BTW, this includes QA, sales, dev-ops, client ops, and EVEN engineers!) Don't just sit at your desk; go out and interact with Declararians!

Working software over comprehensive documentation

- Pertinent Agile Principles:
 - Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
 - Working software is the primary measure of progress.
- What you can expect from engineering:
 - You should be able to interact with functional software at the end of (almost) every sprint.
 - Hold engineering to this. If working software is not coming out of most sprints, something is wrong.

Customer collaboration over contract negotiation

- Pertinent Agile Principles:
 - Agile processes harness change for the customer's competitive advantage.
 - Simplicity—the art of maximizing the amount of work not done—is essential.
- The business side of Agile:
 - Building a product to the letter of a contract is not collaboration, it's confrontation; contracts can be used as both swords and shields.
 - In software development, everything is possible. It is crucial to understand not just whether or not something is possible, but also how difficult it is.

Responding to change over following a plan

- Pertinent Agile Principles:
 - Welcome changing requirements, even late in development.
 - Continuous attention to technical excellence and good design enhances agility.
- Change happens:
 - Customers don't always know what they want. Getting too committed to a plan can be dangerous, both for them and for us!
 - This is a balance. On the one hand, sticking to the plan even when it no longer makes sense can be disastrous. On the other, chasing every opportunity that presents itself can result in a disjointed product.

Questions?

The PMO has a representative dedicated to your team.
Please talk to the following people:

- Sales:
- Engineering:
- QA:
- Operations:
- Product:
- Client Ops:
- HR:
- Legal:
- C-level:

Thank you!

Thank you!

