# Scrum Foundations

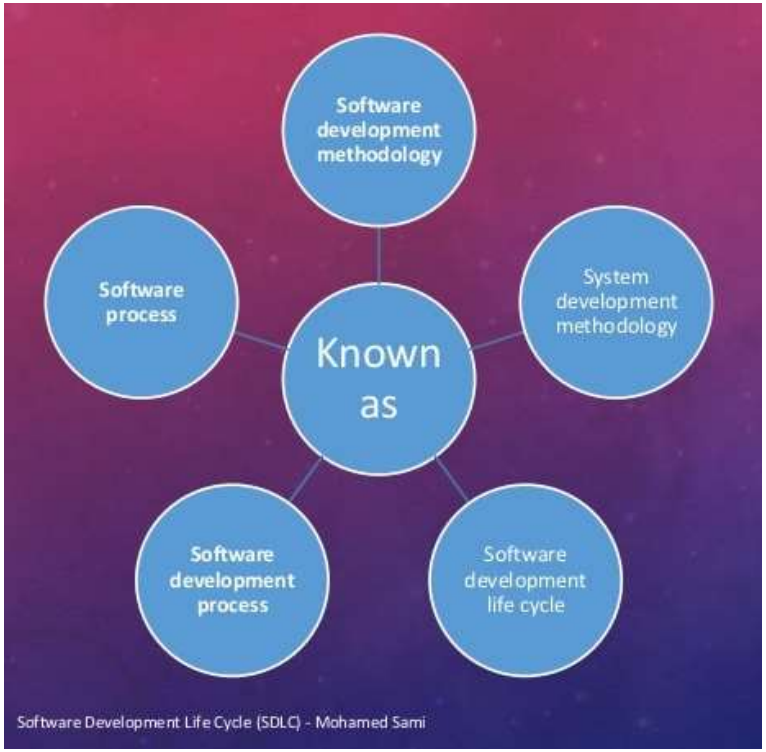
# Software development processes

Formalizes
- who participates in a project
- doing what, when and how

to develop a software product

Common framework
- Phases (or stages)
- Activities
- Methods
- Tools
- Roles

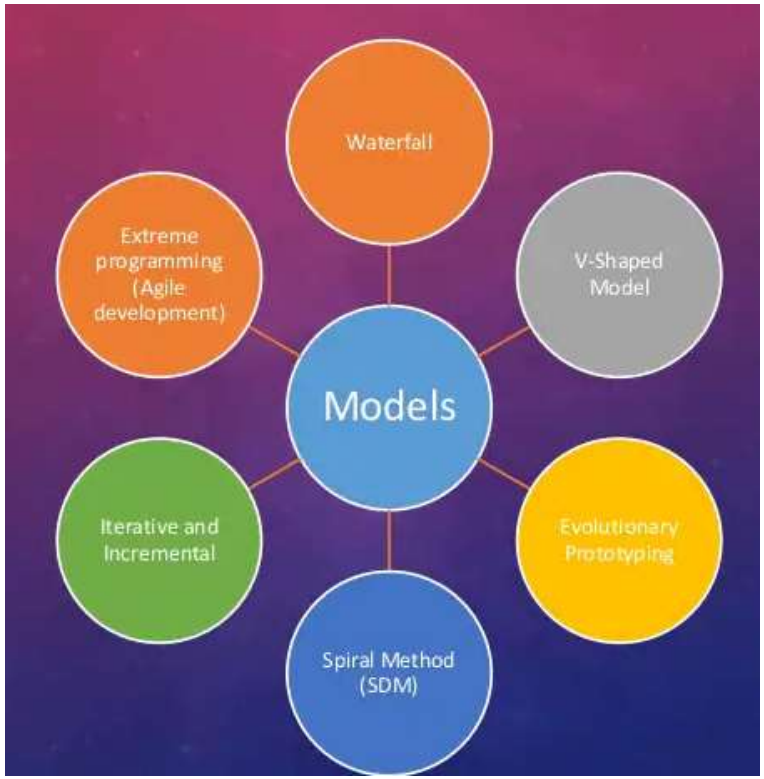
https://melsatar.blog/2012/03/15/software-development-life-cycle-models-and-methodologies/



# What is software development processes

- SDP (SDLC) is a splitting of software development work into distinct **phases** (or **stages**) containing activities with the intent of better planning and management
- Each process model follows series of steps unique to its type, in order to ensure success in process of software development.
- The SDP aims to produce a high-quality software that meets or exceeds customer expectations, reaches  completion within times and cost estimates
- SDP models have been created by software development experts, universities, and standards organizations to solve some repeated issue or to enhance other models.

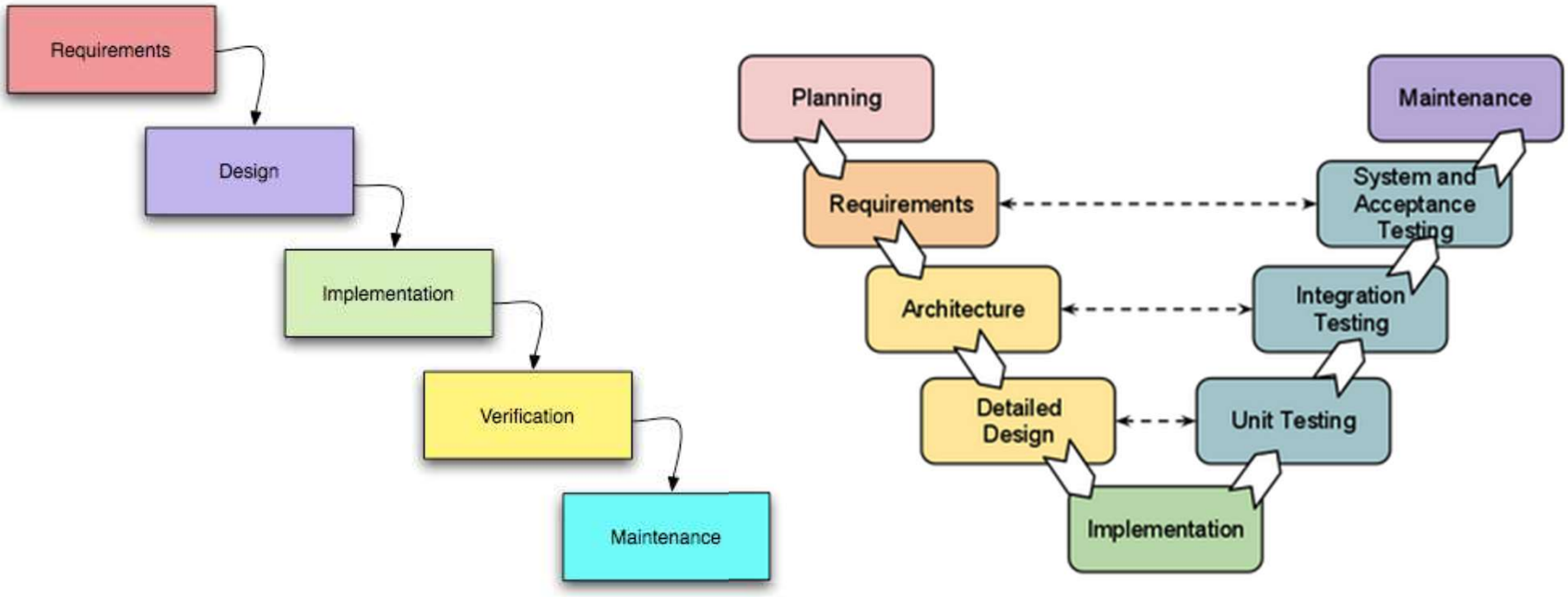
# Main software development processes



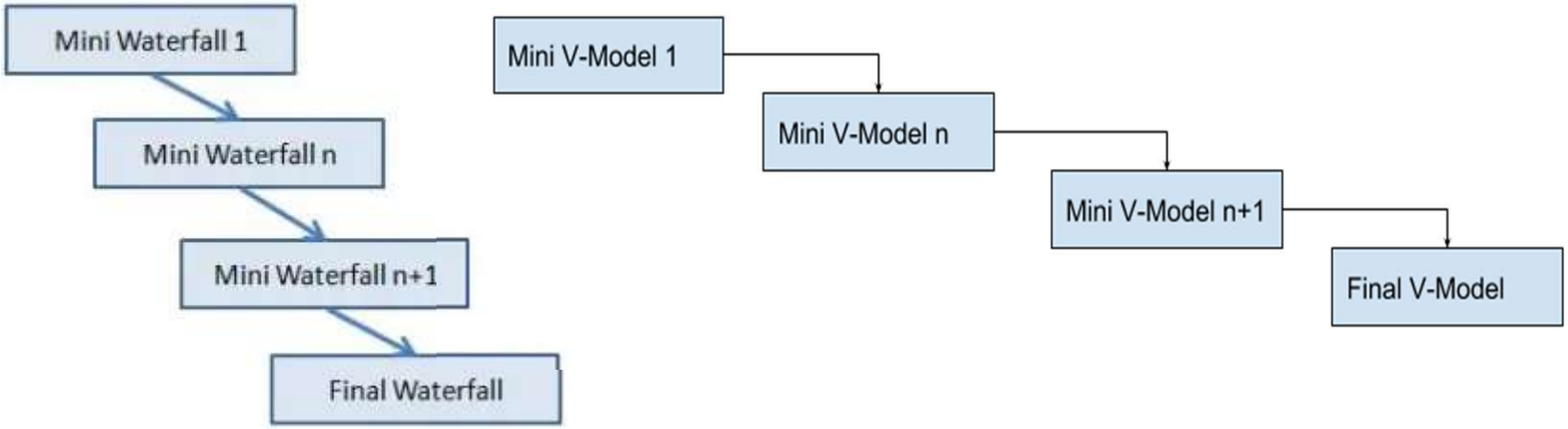

- Methodologies (Life Cycle Models)
  - Waterfall
  - V-Shaped Model
  - Evolutionary Prototyping Model
  - Iterative and incremental development
  - Spiral model
  - Agile

Specific development process

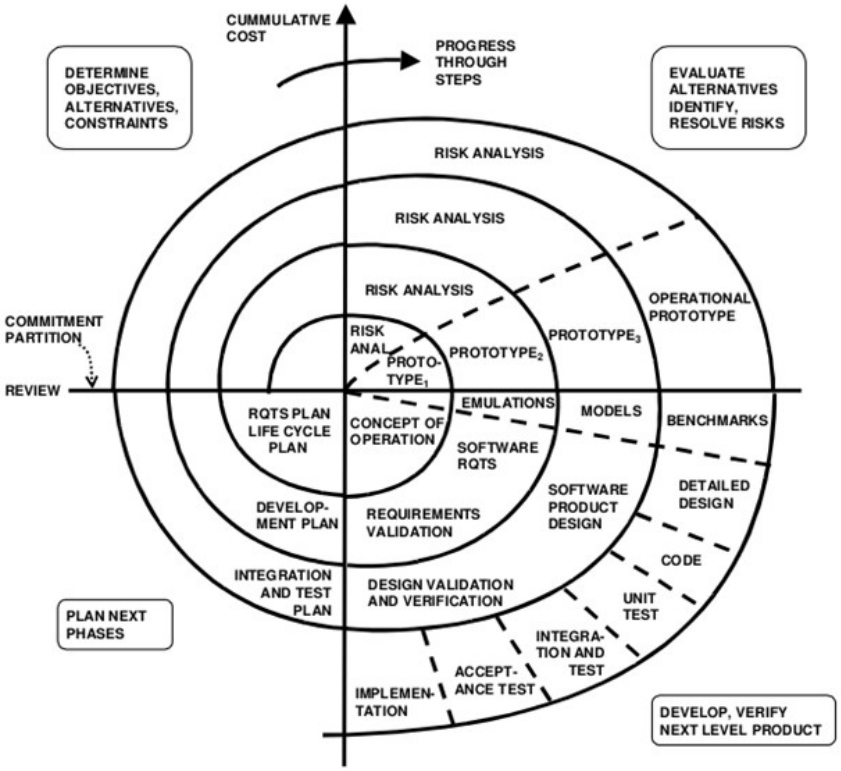
# Sequential Models: Waterfall, V-Shaped



# Iterative and Incremental Model

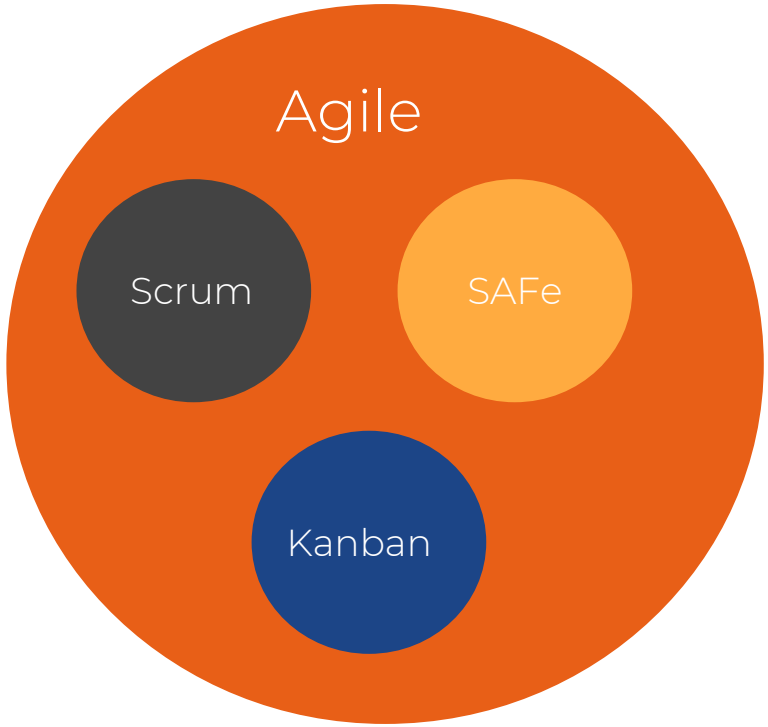


# Sequential & Iterative Models: Spiral Model



## Better Ways Of Working

- Scrum
- Extreme Programming
- DSDM
- Adaptive Software Development
- Crystal
- Feature Driven Development
- Pragmatic Programming




# The Agile Manifesto (March 2001)

**Kent Beck**
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
**Martin Fowler**
James Grenning
Jim Highsmith
Andrew Hunt

Ron Jeffries
Jon Kern
Brian Marick
**Robert C. Martin**
Steve Mellor
**Ken Schwaber**
**Jeff Sutherland**
Dave Thomas

# The values of Agile Manifesto

http://agilemanifesto.org/

**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 Manifesto (1)

http://agilemanifesto.org/

Our highest priority is to **satisfy the customer** through early and continuous delivery of valuable software.

Welcome **changing requirements**, even late in development. Agile processes harness change for the customer's competitive advantage.

**Deliver working software frequently**, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must **work together** daily throughout the project.

# Agile Manifesto (2)

http://agilemanifesto.org/

Build projects around **motivated individuals**. Give them the environment and support they need and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is **face-to-faceconversation**.

**Working software** is the primary measure of progress.

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

# Agile Manifesto (3)

http://agilemanifesto.org/

Continuous attention to **technical excellence** and good design enhances agility.

**Simplicity-**-the art of maximizing the amount of work not done--is essential.

The best architectures, requirements, and designs emerge from **self-organizing teams**.

At regular intervals, the team reflects on how to **become more effective**, then tunes and adjusts its behavior accordingly.

# Agile is not prescriptive

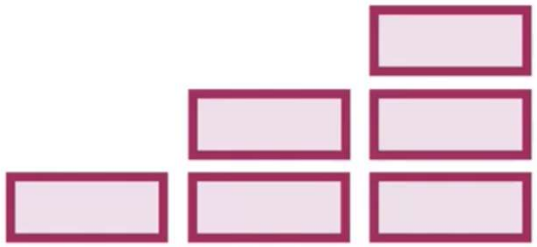
# Agile

is a philosophy

# Scrum

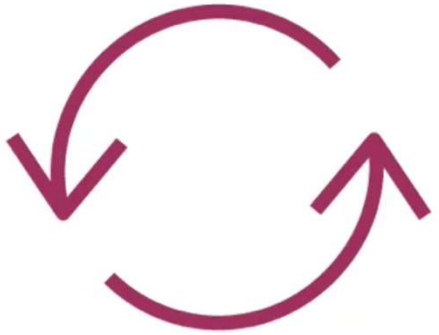
is a framework

# Incremental Versus Iterative



**Incremental**

Small increments of a project are delivered piece by piece until the entire project is complete



**Iterative**

Clear feedback is elicited after each increment and incorporated into the next increment

# Incremental and Iterative Delivery

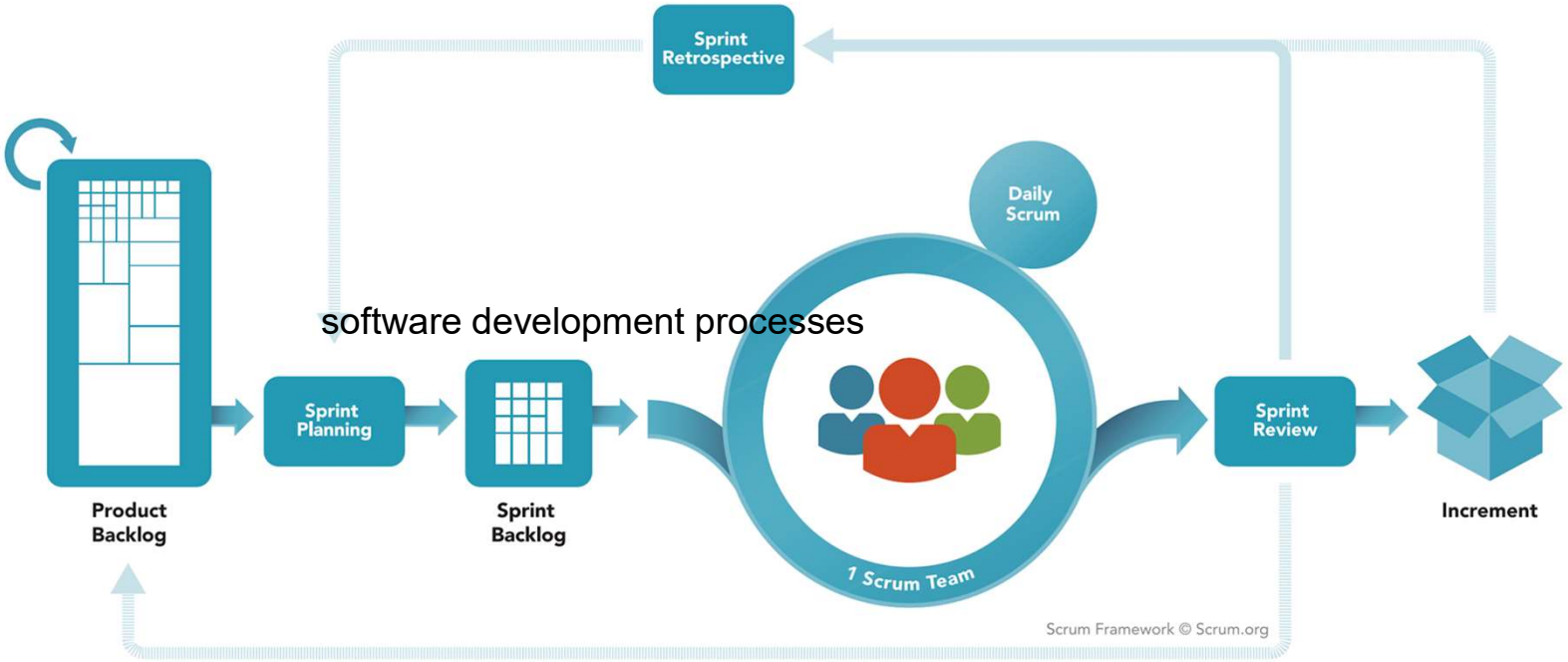
Breaks a larger project into smaller deliveries

Can reduce the overall effort

Reduces the impact of costly mistakes

Incorporates feedback along the way

# SCRUM FRAMEWORK




software development processes

Scrum.org

# The Roots of Scrum

## Empirical Process Control

Prevalent in the manufacturing space

## Toyota Production System

Early Scrum was heavily influenced by the TPS

## Predates the Agile Manifesto

Scrum appears in the 1990

# The New Product Development Game (1986)

Resaltan las ventajas competitivas aparecidos en la industria japonesa gracias a nuevos modelos de desarrollo de producto basados en la innovación y la rapidez a la hora de lanzar sus productos al mercado. En estos casos se observaba una mayor interacción de equipos multidisciplinarios de elite que trabajaban en proyectos de inicio a fin

utilizan como metáfora el juego del rugby, haciendo referencia a la melé (scrum), donde un equipo unido de quince jugadores tiene como objetivo común hacer avanzar el balón hacia el campo contrario

Un enfoque de 'carrera de relevos' en el desarrollo de productos … puede entrar en conflicto con los objetivos de máxima velocidad y flexibilidad. En su lugar, un enfoque holístico o estilo 'rugby' – donde un equipo intenta ir a la distancia como una unidad, pasando la pelota hacia adelante y hacia atrás -pueden servir mejor a los actuales requisitos competitivos".

Hirotaka **Takeuchi** y Ikujiro **Nonaka**, *Harvard Business Review*, Enero 1986.

# The Roots of Scrum

## Empirical Process Control

Prevalent in the manufacturing space

## Toyota Production System

Early Scrum was heavily influenced by the TPS

## Predates the Agile Manifesto

The manifesto was written in 2001, but Scrum appears in the 1990

# Scrum Is a Framework, Not a Process

## Process

Step-by-step guidance

Repeatable from project to project

## Framework

Definition of key tasks and routine

Non-prescriptive

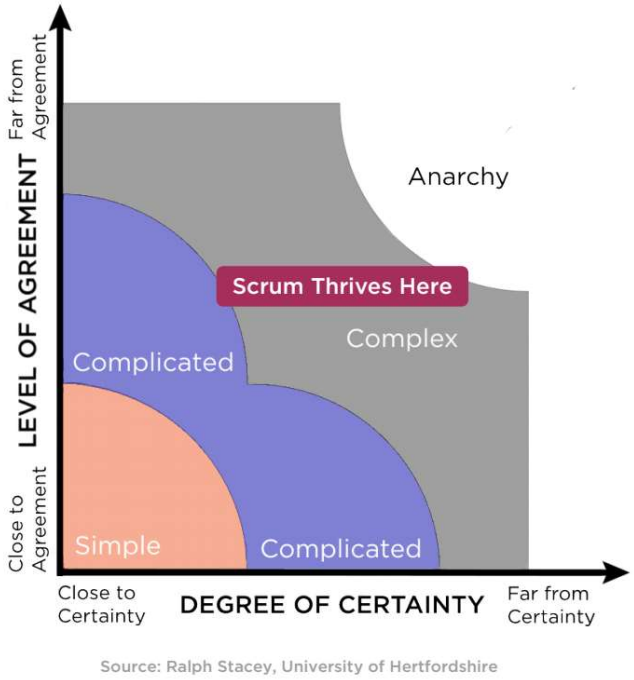
# Your Scrum is not my Scrum

Scrum Framework  Your Environment  Your Process

Scrum does not solve your problems

Scrum enables *you* to solve your problems

# When Should We Use Scrum?




Source: Ralph Stacey, University of Hertfordshire

- The Stacey Matrix is a decision aid
- Zone indicates where Scrum helps
- Scrum thrives in the Complex zone
- Complexity needs leadership

# Ideal Environment for Scrum

## Well Suited

High amount of complexity

Aggressive deadlines

High level of risk

## Not Well Suited

Priorities change daily

Team composition changes regularly

Transparency is not encouraged

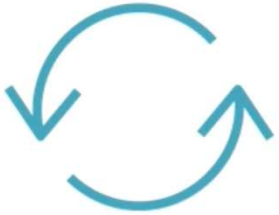
# Empiricism

Knowledge comes only, or primarily, from experience.

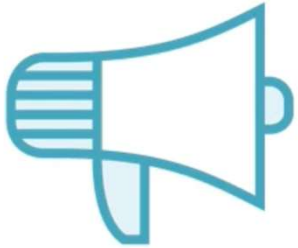
*- Wikipedia (Empiricism)*

# Three Pillars of Empiricism

Inspect

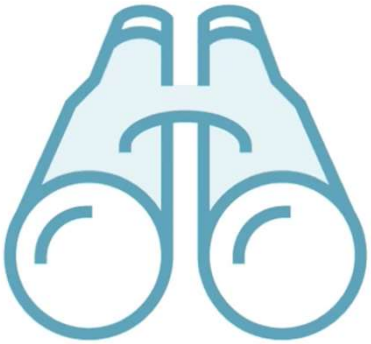


Adapt



Transparency

# Inspect



Inspect both the project and the process

Look for variances

Strike the right balance

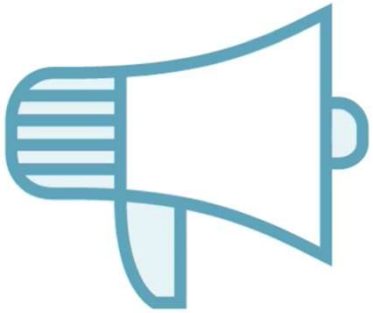
# Adapt

Reduce what isn’t working, increase what is

Introduce changes at the right time

Adapt the process to serve the project

# Transparency



Make your pace and progress visual

Encourage openness across the team

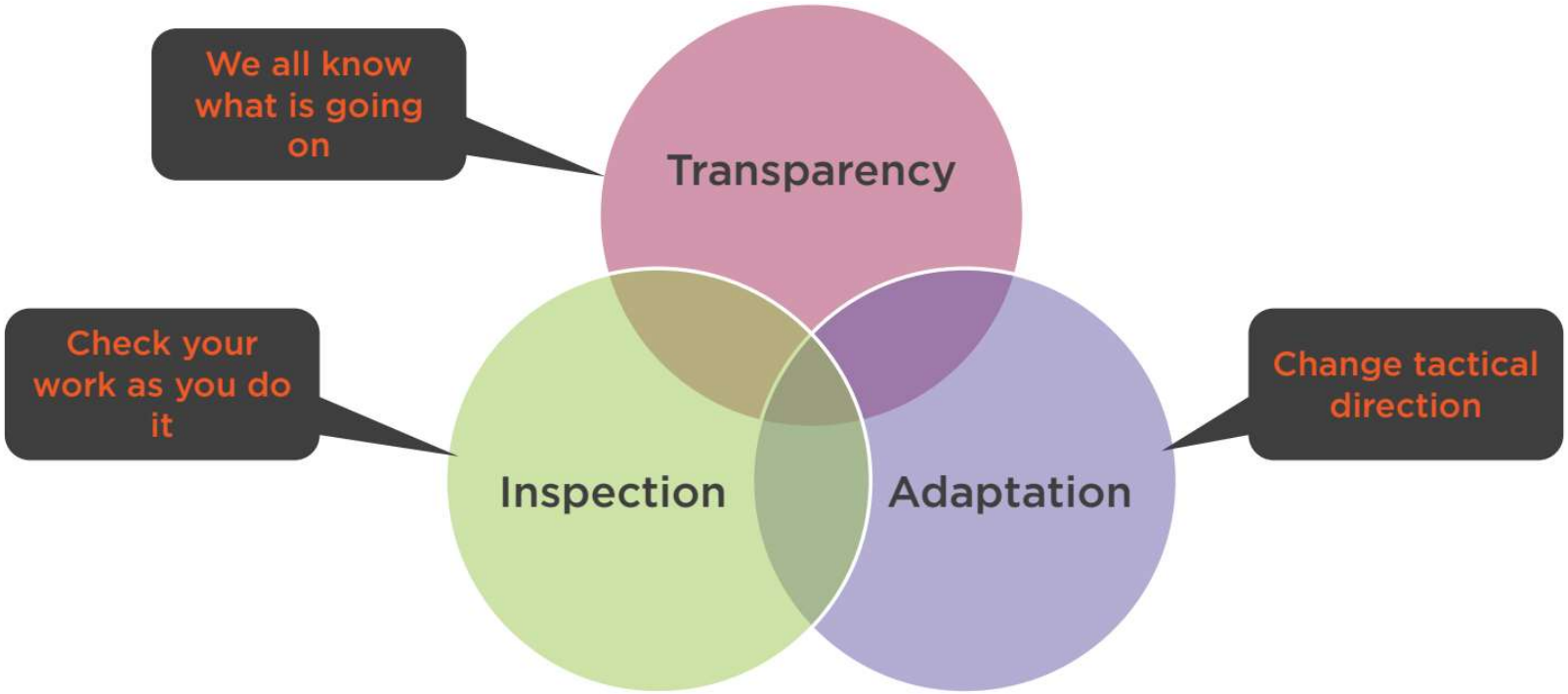
Share clear status across team boundaries

# How Scrum is Driven by Empiricism

**Empirically Driven Decisions**
Make decisions based only on what has occurred

**Lagging Indicators**
Use indicators driven from historical metrics

# Scrum Implements the Empirical Process



# Scrum Elements

## Roles

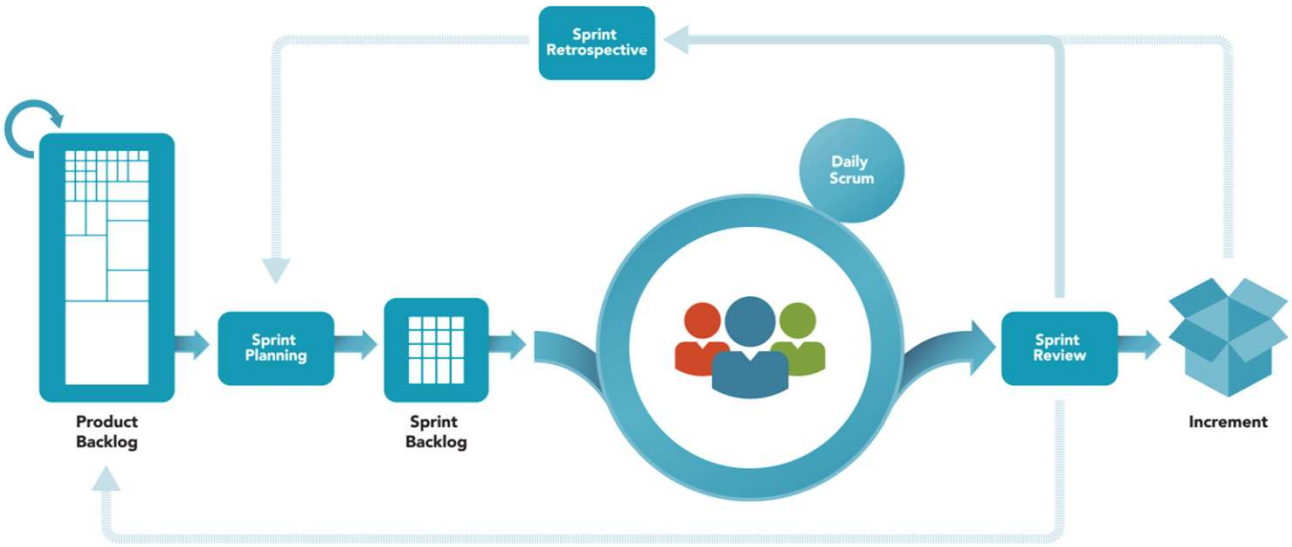
- Product Owner
- Development Team
- Scrum Master

## Artifacts

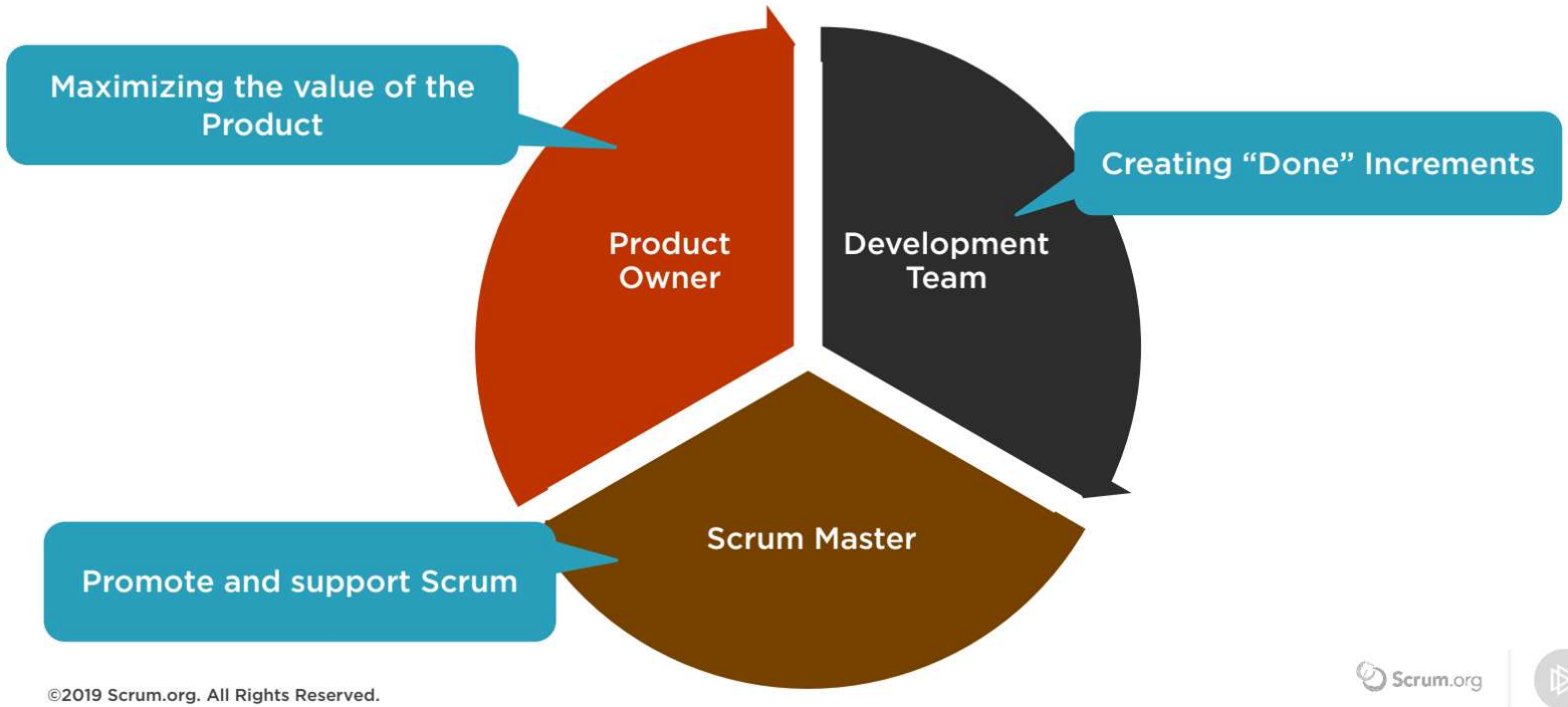
- Product Backlog
- Sprint Backlog
- Increment

## Events

- Sprint
- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective



# Roles: Each One Has a Specific Responsibility






Scrum.org

# Events: Each One Has a Specific Purpose

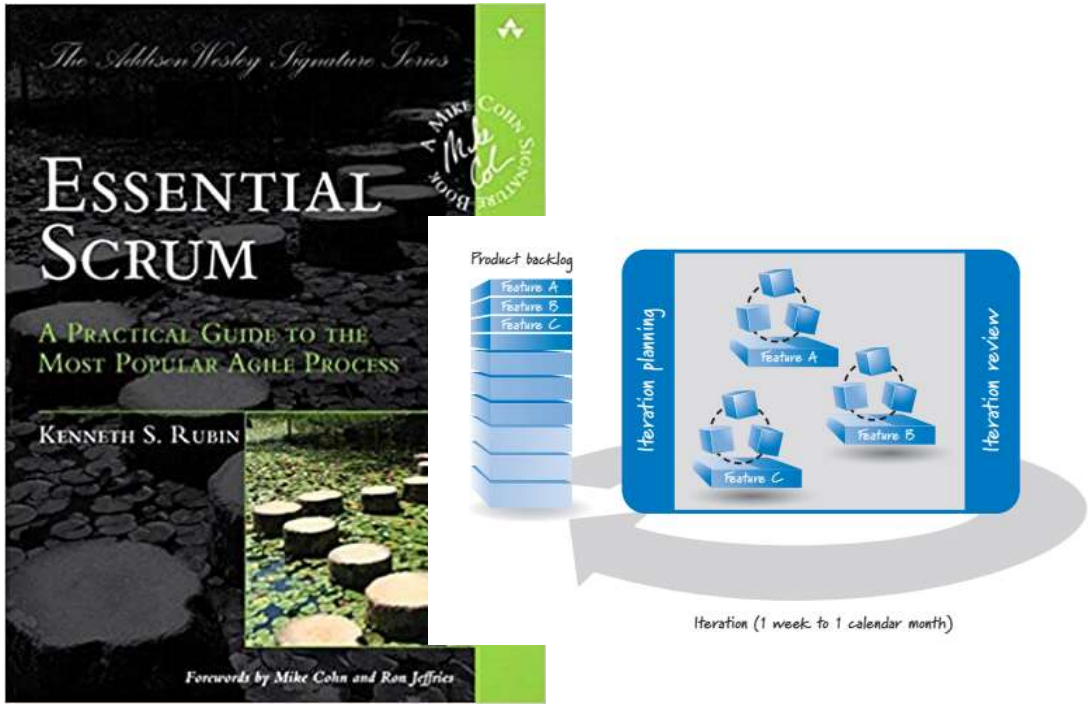
| | |
|---|---|
| Sprint Planning | • From: Product Backlog<br>• To: Sprint Goal, Sprint Backlog |
| Daily Scrum | • From: Daily Progress, Sprint Backlog<br>• To: Updated Daily Plan |
| Sprint Review | • From: Sprint, Increment<br>• To: Updated Product Backlog |
| Sprint Retrospective | • From: Past Sprint<br>• Improvements For Next Sprint |
| Sprint | • Container Event<br>• One month, or less, in duration |

# Artifacts: Each One Contains Specific Information

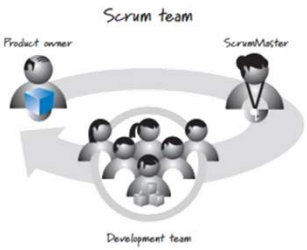
| | |
|---|---|
| Product Backlog | • Holds the requirements for the product<br>• Managed by the Product Owner |
| Sprint Backlog | • Holds all work for the Sprint Goal<br>• Managed by the Development Team |
| Increment | • Working addition to the product<br>• Potentially releasable |

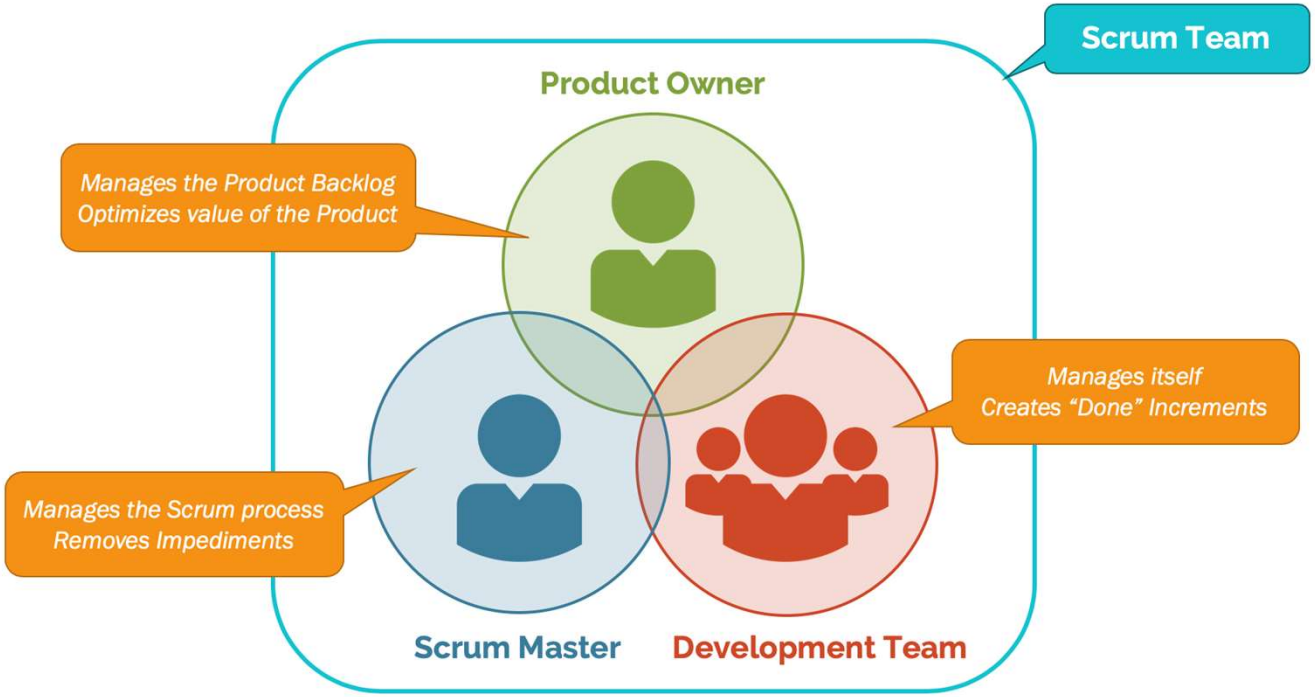
# Essential Scrum (2012)

Kenneth S. Rubin
Essential Scrum. A Practical Guide to the Most Popular Agile Process (2012, Addison-Wesley Professional)

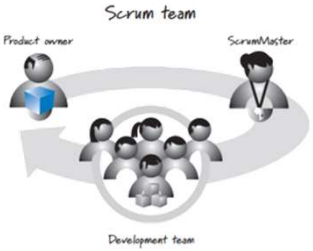


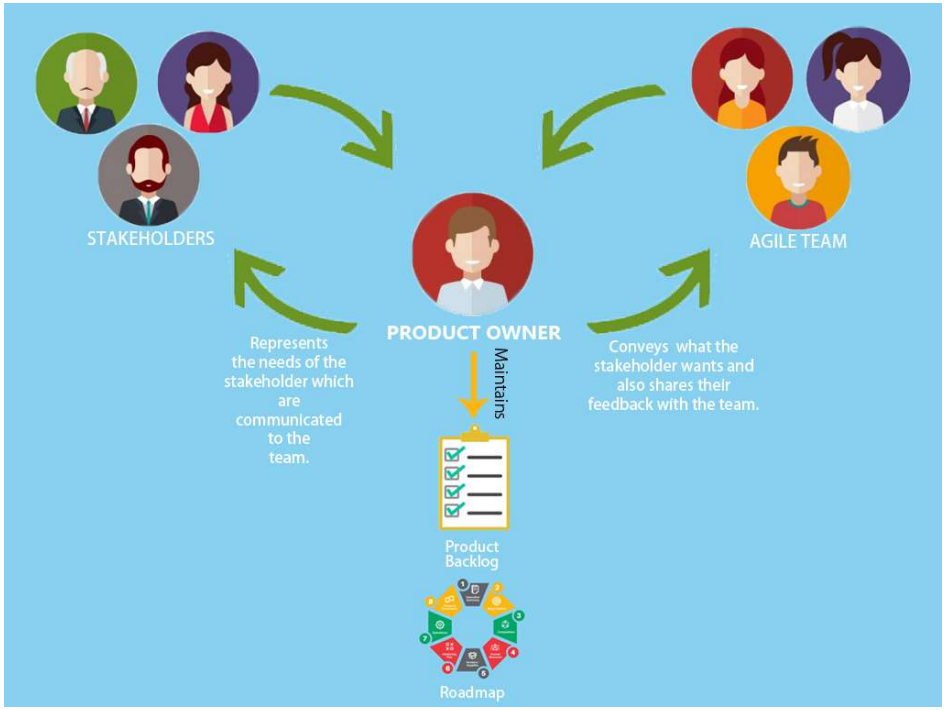
# Roles


Scrum team
Product owner
ScrumMaster
Development team


Scrum Team
Product Owner
Manages the Product Backlog
Optimizes value of the Product
Manages itself
Creates "Done" Increments
Manages the Scrum process
Removes Impediments
Scrum Master
Development Team

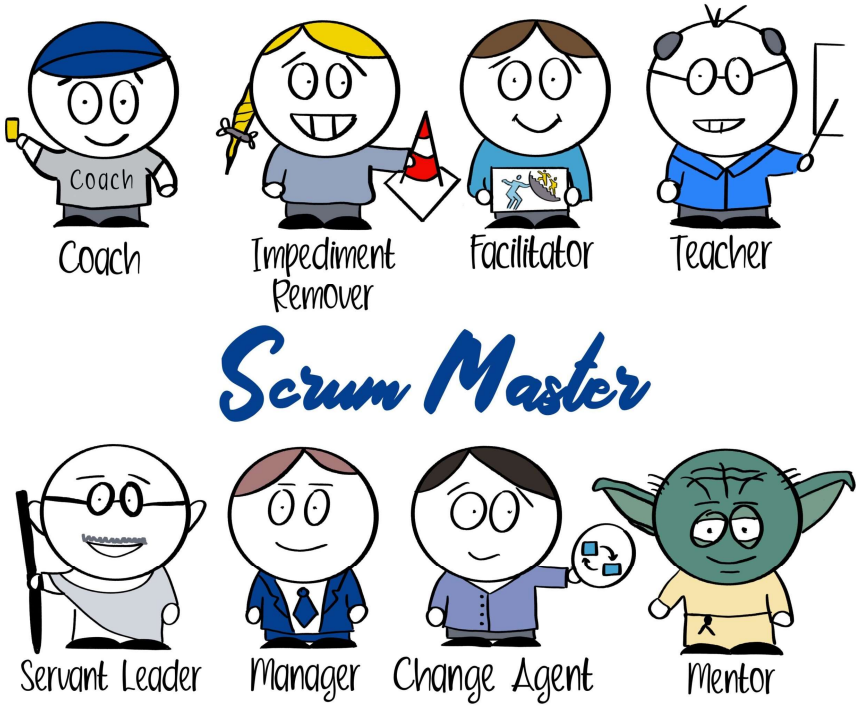
# Product Owner

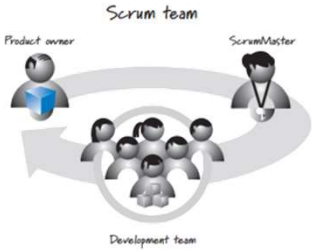


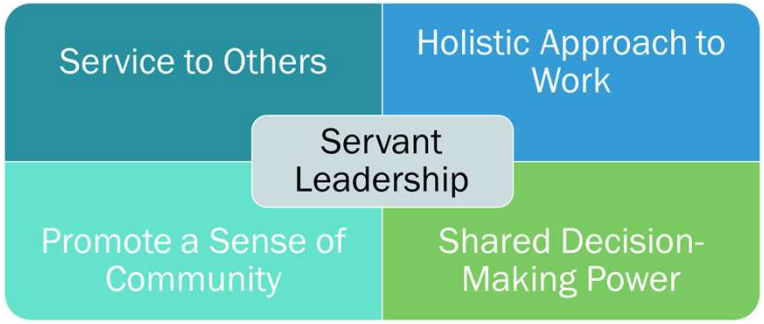




- Responsable de la visión de producto y la gestión económica de su desarrollo
- Nexo de conexión entre equipo de desarrollo y stakeholders, clientes y usuarios
- Decide que características y funcionalidades ha de tener el sistema en desarrollo y el orden en que deben ser implementadas (Product Backlog) y las valida
- Participa activamente en el equipo Scrum

# Scrum Master

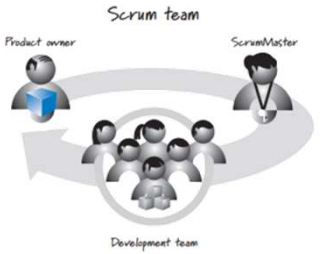



https://www.scrum.org/resources/what-is-a-scrum-master

# Development Team

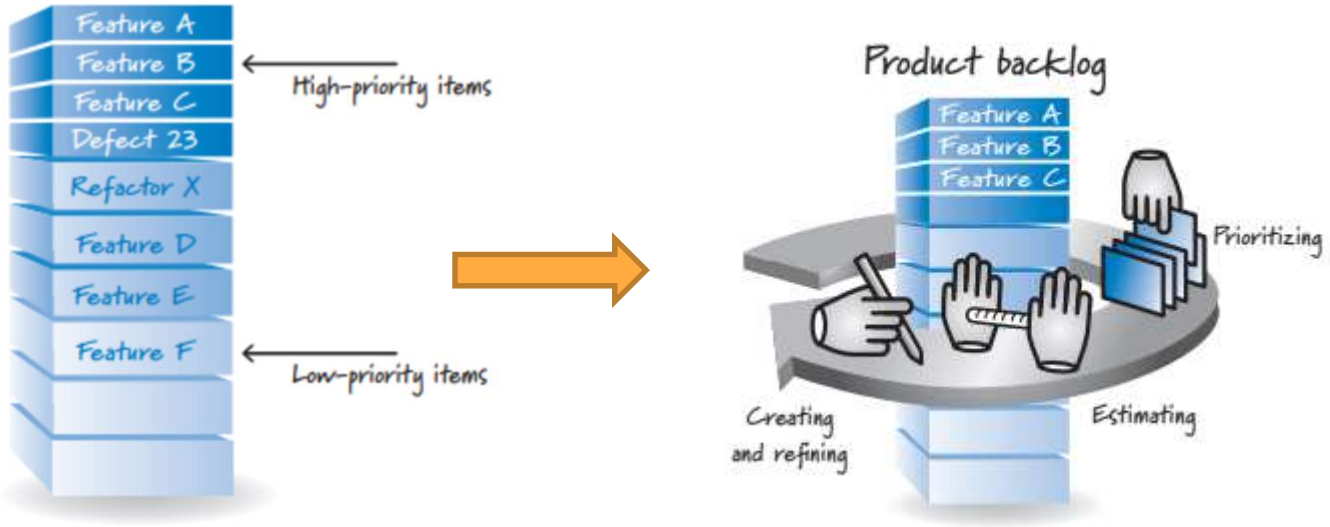




5-9 members

# Events & Artifacts

# Product Backlog




Product backlog grooming

# Sprints

**Fixed timebox in which to work**

Often defined in 1 to 4 weeks

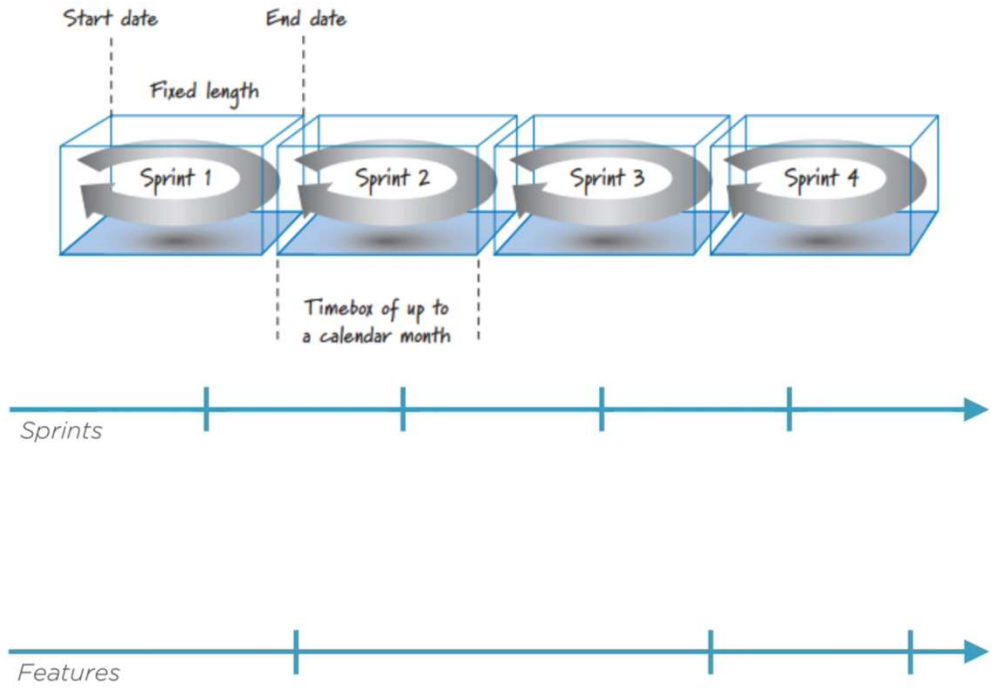
**Preselected batch of work**

Team attempts to complete all work

**Review at completion**

Incorporate feedback into the next sprint

# Feature Increments Versus Sprint Increments



# Feature Increments Versus Sprint Increments

## Feature Increments

Features not sized consistently

Feedback comes unevenly

May not receive feedback on all features

## Sprint Increments

Feedback occurs more regularly

Encourages stakeholders to give more frequent feedback

Easier to coordinate

# How Long Should a Sprint Be?

**As short as 1 week in duration**

The Scrum Guide™ places no lower limit on sprint lengths

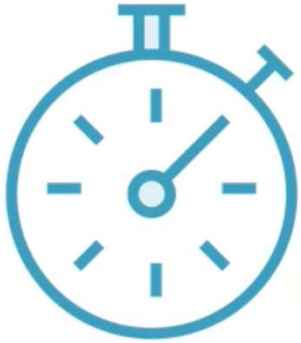
**2 weeks in duration**

The most common sprint duration for Scrum teams

**As long as 30 days in duration**

The Scrum Guide™ defines this as the upper limit

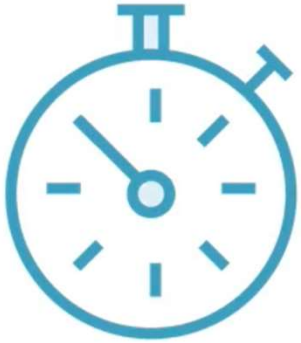
# Shorter Sprints Versus Longer Sprints



**Shorter Sprints**

Easier but more frequent planning

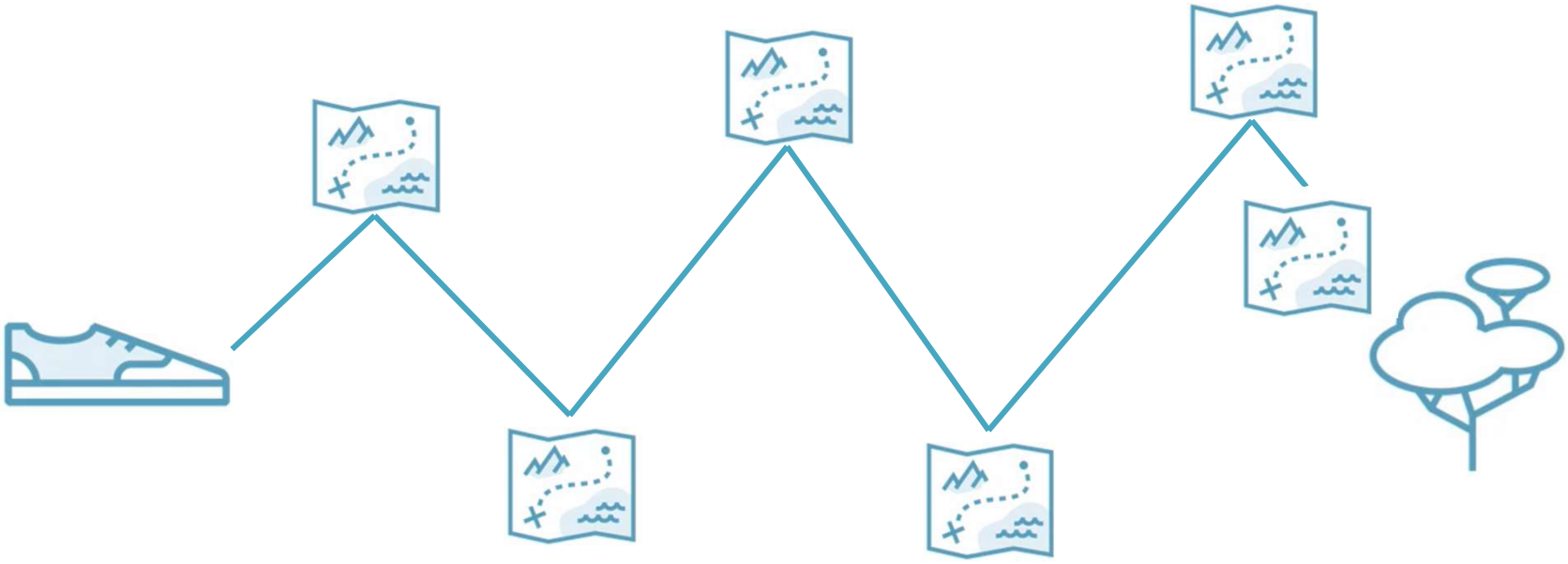
Better ability to reduce risk

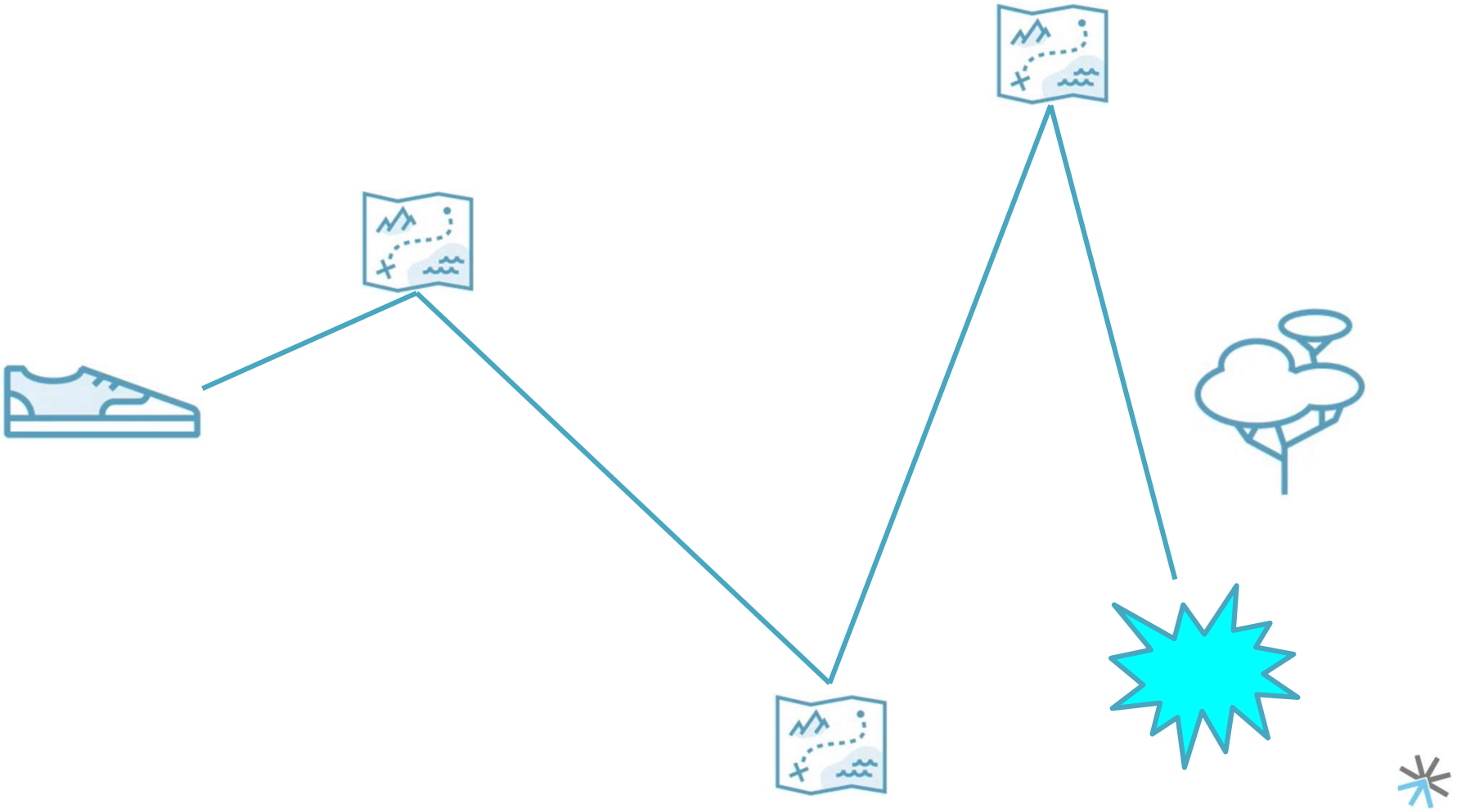


**Longer Sprints**

More difficult but infrequent planning

Will incur more risk





# Skylab Bootcamp Sprints


node.js

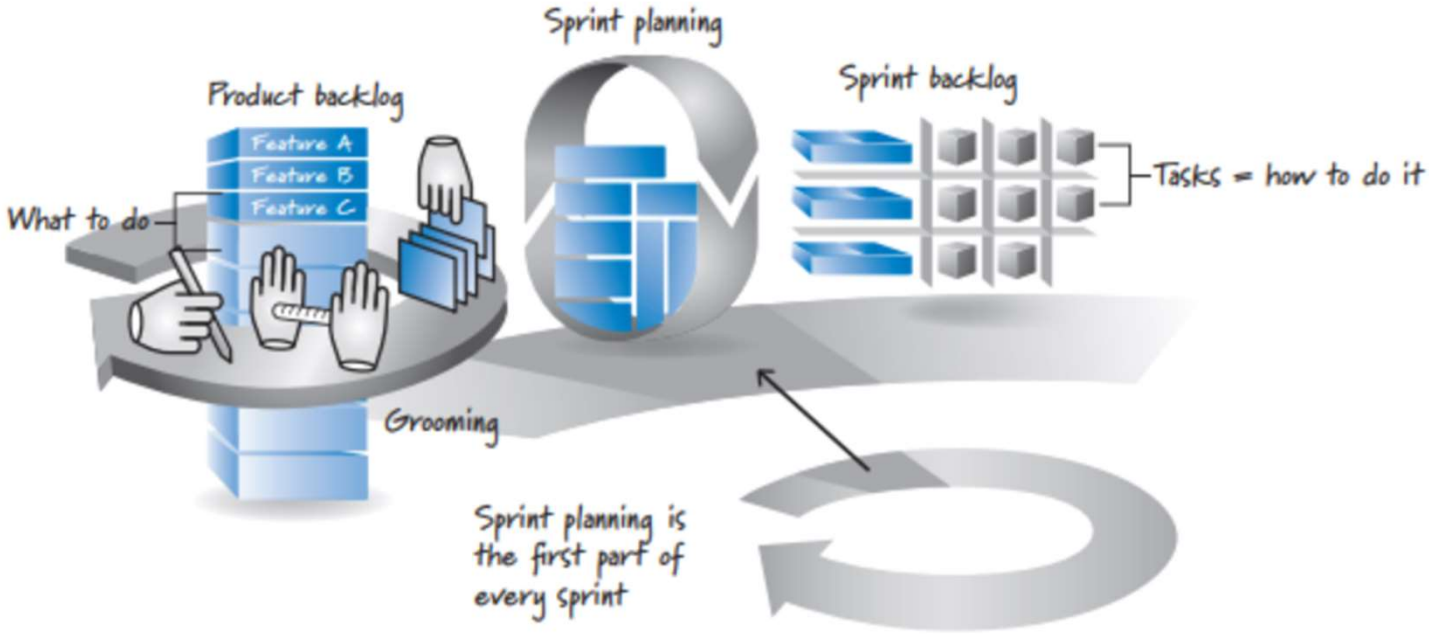
express

mongoDB.

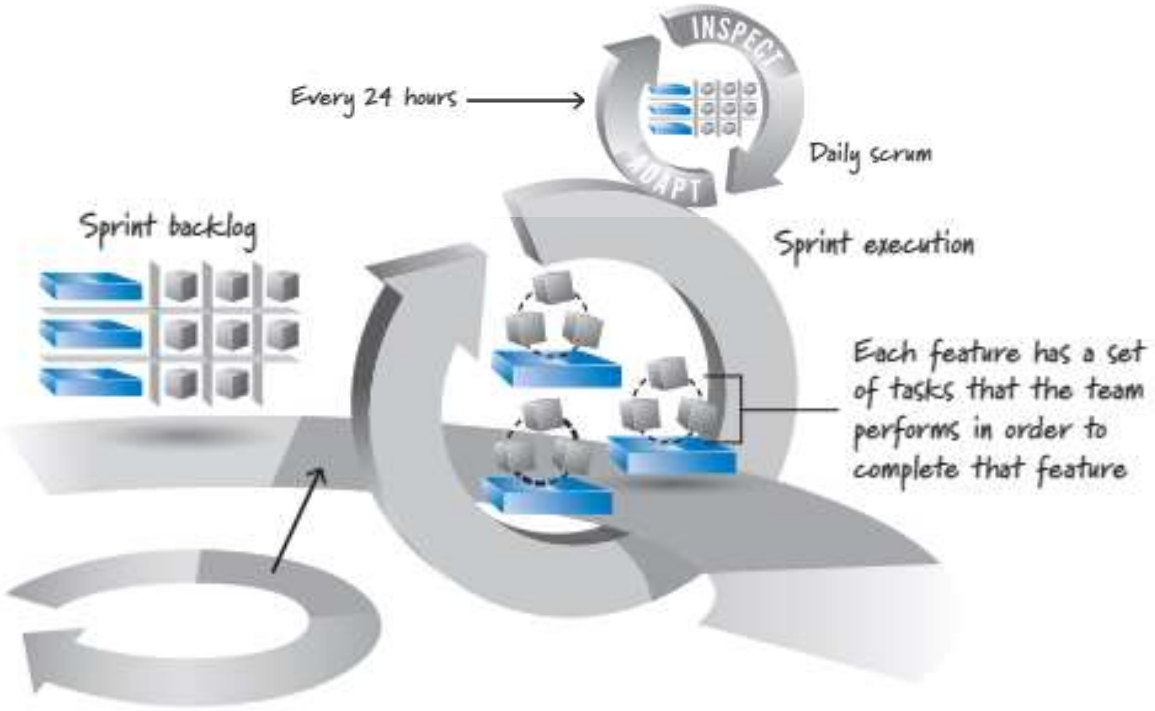
SKYLAB CODERS ACADEMY

💬final projects 🤖👧💻

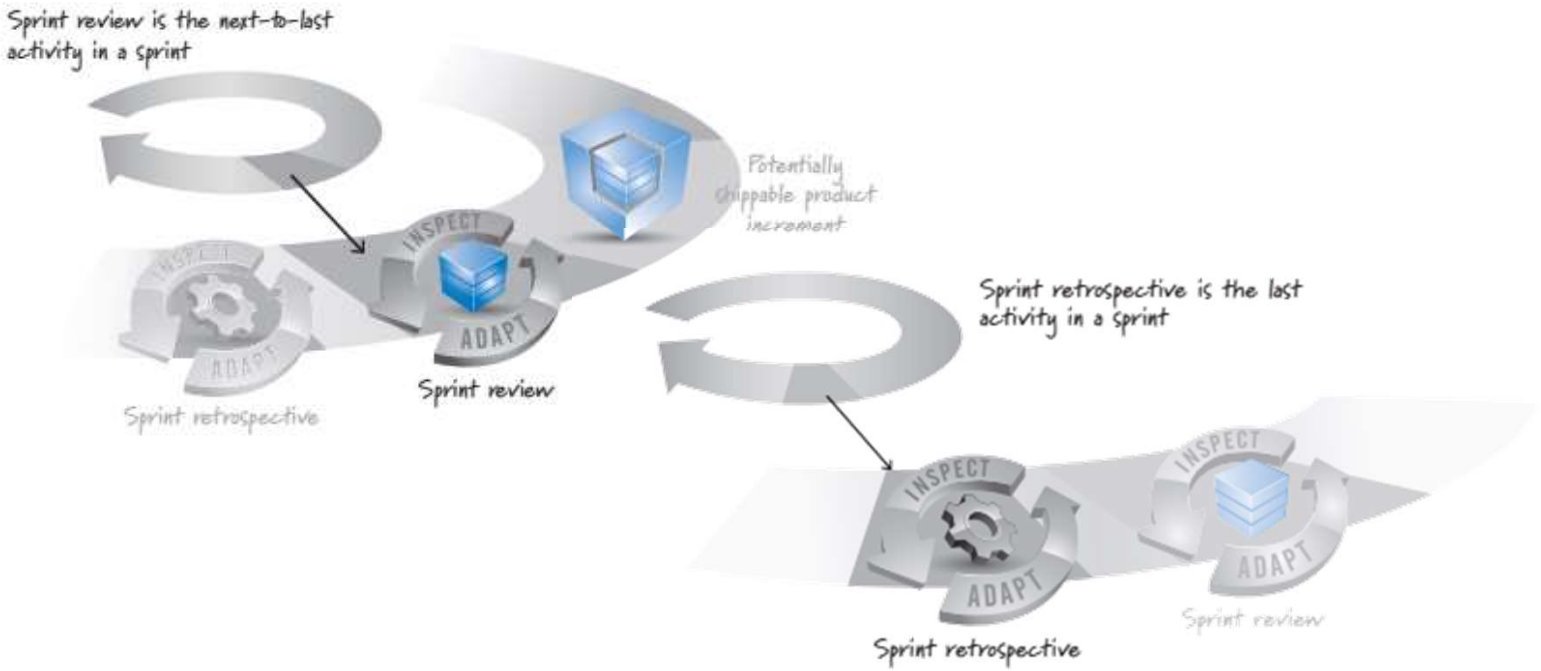
# Sprint Planning



# Daily Scrum


Every 24 hours
INSPECT
ADAPT
Daily scrum
Sprint backlog
Sprint execution
Each feature has a set of tasks that the team performs in order to complete that feature

# Sprint review & retrospective



# Share Understanding of Done

# Creating a Definition of Done

**Attainable**

Realistically represents the capabilities of your team

**Collaborative**

Created by both stakeholders and the development team working together

**Flexible**

Subject to change as the team finds better ways to work

# ISDI Coders Bootcamp DoD

**Attainable**

All the code uploaded to GitHub

**Collaborative**

At least 2 reviewers on each pull request

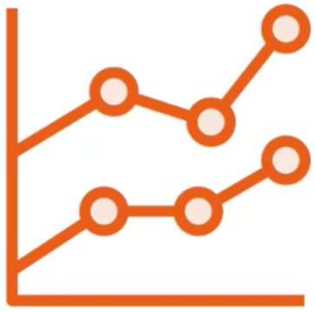
**Flexible**

DoD will evolve with the bootcamp

Potentially shippable
product increment

Always leave the product in a releasable state at the end of each sprint.

# Advantages of Potentially Shippable



**Better business value**

Recoup cost sooner or lessen overall investment

**Reduced risk**

Create a product that is ready to ship at any moment

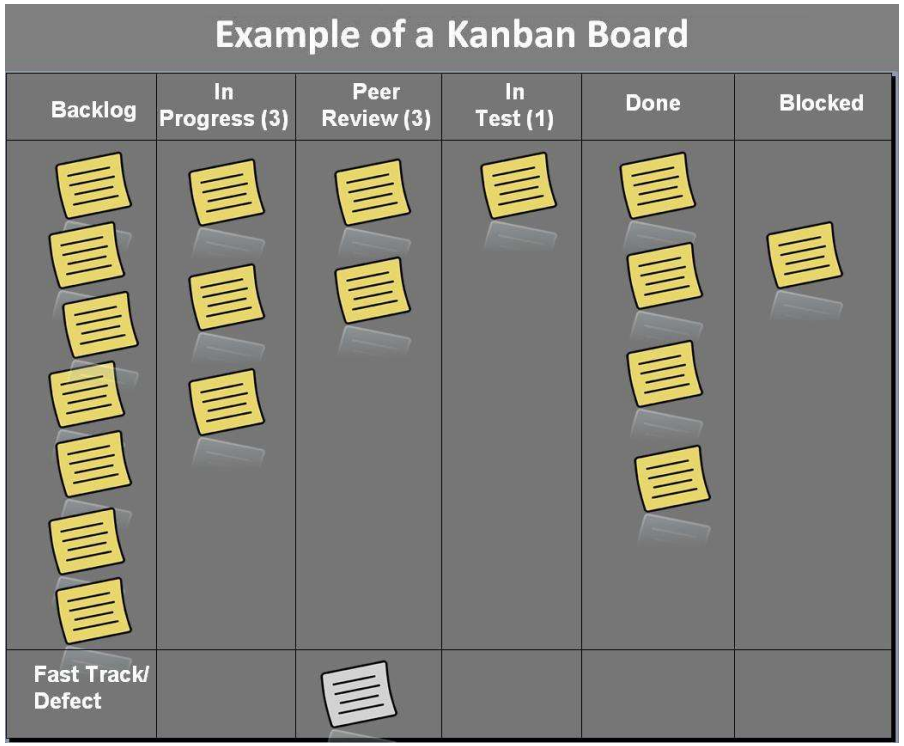
**More transparency**

Convey the team's true status to project stakeholders

# Kanban -> Scrumban

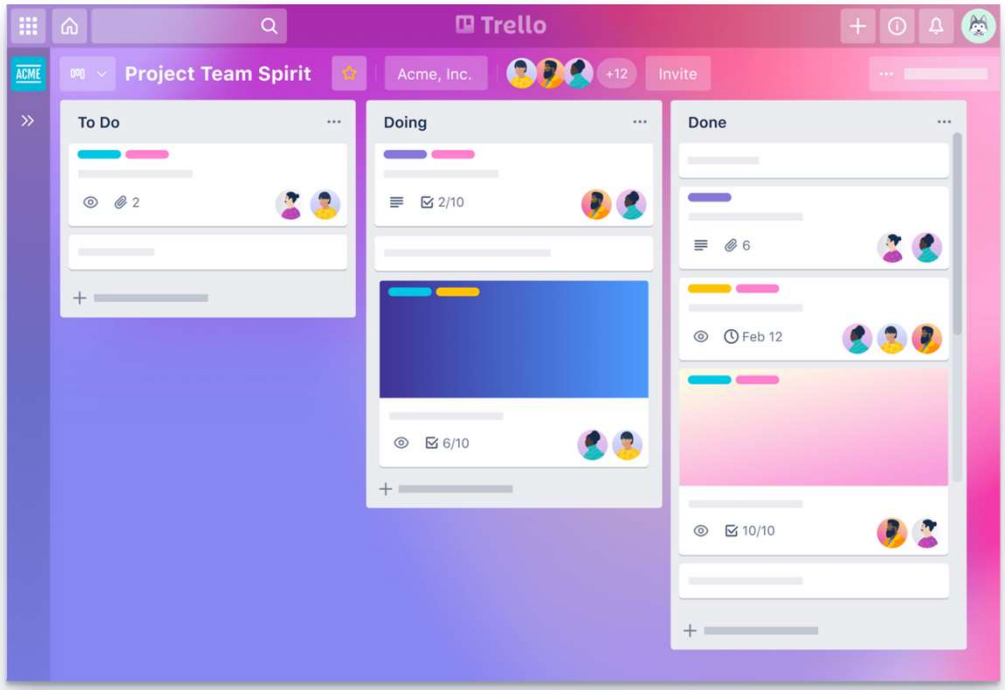
Kanban (Japanese: 看板, meaning signboard or billboard) is a lean method to manage and improve work across human systems.

Originated in lean manufacturing, which was inspired by the Toyota Production System.



Trello

Iniciar sesión Registrarse



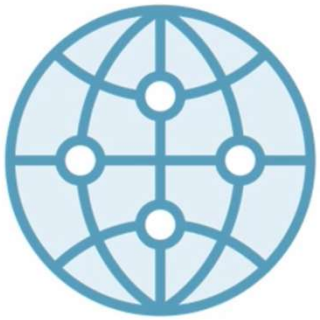

https://trello.com/

Únete a más de 1 millón de equipos de todo el mundo que utilizan Trello para obtener más y mejores resultados.

Google Fender SQUARESPACE COSTCO WHOLESALE

# The Five Values of the Scrum Framework

# Learning More



Agile Manifesto
www.AgileManifesto.org

Scrum Guide
www.ScrumGuides.org