



PROJECT MANAGEMENT  
CENTER FOR EXCELLENCE

A.J. CLARK SCHOOL OF ENGINEERING  
Civil & Environmental Engineering Department



# AGILE PROJECT MANAGEMENT

*Course 1. Applied Scrum for Agile Project Management*

*John Johnson, PMP CSM SPC*

*A. James Clark School of Engineering, University of Maryland, College Park*

*Copyright 2018, All rights reserved*



# Week 1. Agenda

- 1. Agile Basics**
- 2. Proof Agile Works**
- 3. Evolution of Agile**
- 4. Case Study 1 – Netflix**
- 5. Case Study 2 – 18 F**



# Valuable Sprints & Dispelled Myths

# Agile Basics



## Intro to Agile: the Manifesto

Agile was codified in 2001 at the Snowbird Resort by 17 practitioners of Iterative Development. The Agile Manifesto was written by XP, DSDM, and Scrum practitioners stating

<http://logicboost.com/agilemanifesto.html>



① **CUSTOMER COLLABORATION**  
over contract negotiation

② **INDIVIDUALS AND INTERACTIONS**  
over processes and tools

③ **RESPONDING TO CHANGE**  
over following a plan

④ **WORKING SOFTWARE**  
over full documentation



## Intro to Agile: the Manifesto

*“...while there is value in items on the right, we value items on the left more...”*

- Individuals and Interactions over *processes and tools*
- Working Software over *comprehensive documentation*
- Customer Collaboration over *contract negotiation*
- Responding to Change over *following a plan*

# Sprint Basics

## Sprint Planning

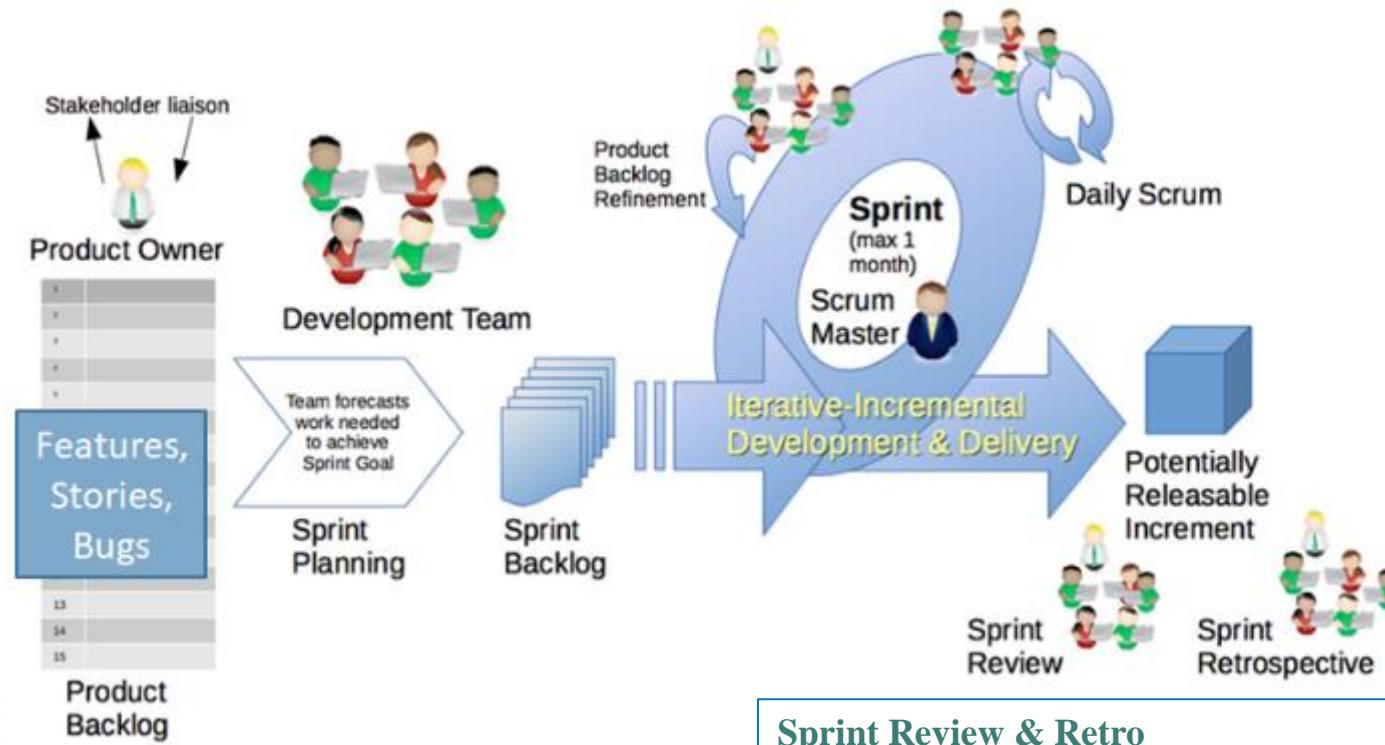
- Team & Product Owner select work
- Team commits to complete work inside the Sprint
- All work is stated as a “User Story” with a clear “who, what, why” and acceptance criteria
- Scrum Master facilitates and guides

1

## Sprint Development

- Team meets daily to decompose & assign work
- Team self-organizes based on skills
- No client can interrupt or change their work
- Product Owner liaisons with end users
- Product Owner builds and prioritizes backlog
- Scrum Master facilitates and tracks

2



[https://en.wikipedia.org/wiki/Scrum\\_\(software\\_development\)#/media/File:Scrum\\_Framework.png](https://en.wikipedia.org/wiki/Scrum_(software_development)#/media/File:Scrum_Framework.png)

## Sprint Review & Retro

- Team presents completed work to customer
- Team reviews work performed
- Team performs retrospective to improve itself
- Scrum Master facilitates and guides

3

# Comparing Agile, Traditional, and Lean

Scope

Schedule



Total Cost  
(Not Quality)

Budget

	Agile	Traditional	Lean
Adjust	Scope	Budget	Schedule
Requires	Trust	Efficiency	Expertise
Goal	Speed	Predictability	Innovation

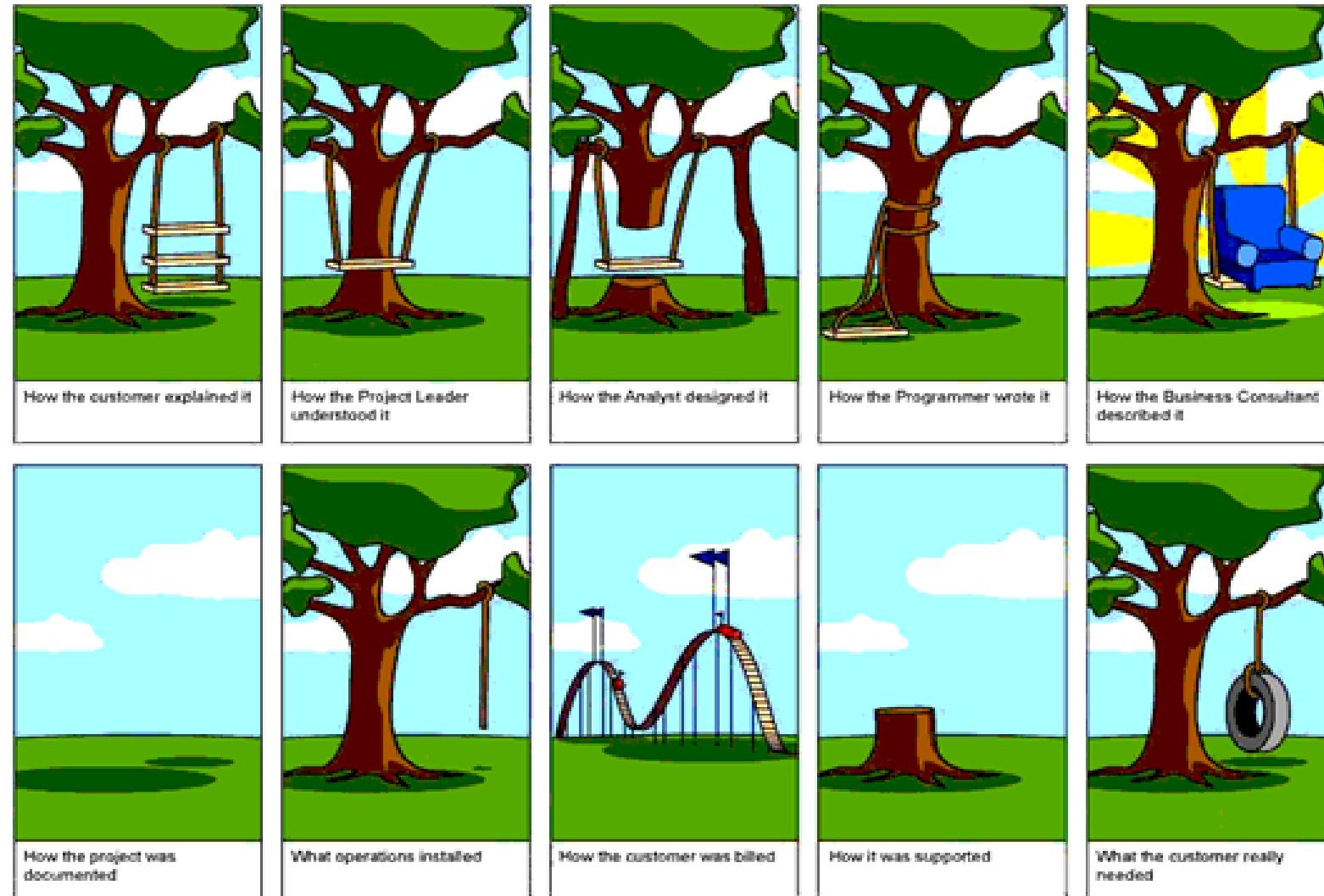


## False Comparisons

The following is true for not only Traditional, but also Lean, and Agile projects. Each methodology has:

- *A Charter*
- *A Plan*
- *Documentation*
- *Design*
- *Testing*

# The Story





## Wait, so What Is Agile Again?

Agile is a Project Management ***Methodology*** we can define as having:

- Shared Vision Robust to Change (can vary tech scope)
- Whole Teams (customer + a cross-functional team)
- Incremental Delivery (learn by doing using small “Sprints”)
- Continuous Integration & Testing (teams test increments early often)

***Scrum, SAFe, Disciplined, Kanban*** - these are ***Frameworks***  
which offer a structure for conducting Agile projects.



# From Spacecraft to Supercomputers Proof That Agile Works

## Proof Agile Works: Skunkworks

- Clarence Leonard “Kelly” Johnson, Lockheed Martin Engineer in WWII.
- In 1943, tasked with extending range of fighter jets
- He and his team colocated in a tent because they needed the space...
- Program was called “Skunk Works”



*Designed and built the first jet-fighter, “P-80 Shooting Star,”  
in just 143 days*



# Keys to Skunkworks Success

Kelly Johnson's Skunkworks Program had 14 Rules of Management, which roughly translate to:

- Small, Strong, Self-Directed Cross-functional Teams
- Owners and Vendors must Collaborate and Trust
- Manage and Respond to Change
- Minimize Reports, But Record Important Work
- Incremental Development with Self-Testing Teams

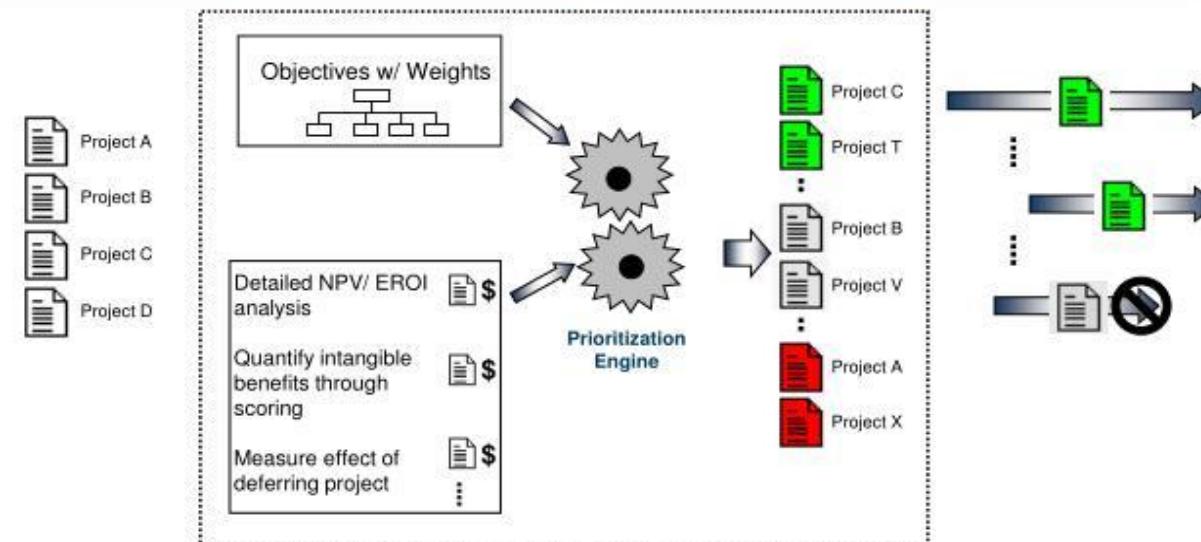
Remember! Agile is a Project Management *Methodology* we can define as:

- Shared Vision Robust to Change (can vary tech scope)
- Whole Teams (customer + a cross-functional team)
- Incremental Delivery (learn by doing using small “Sprints”)
- Continuous Integration & Testing (teams test increments early often)

# Proof Agile Works: Navy Energy Program



## eROI Project Selection Process



eROI creates a Navy-wide, optimized portfolio of energy projects and investments which position the Navy to achieve its energy goals with efficient use of resources



# Proof Agile Works: Navy Energy Program

## Navy Shore Energy Program, Energy Return on Investment (eROI) Support

*Booz Allen Hamilton (BAH)*

**Scope:** Build decision support systems to identify, evaluate, and select \$500M/yr. in shore energy projects

**Total Cost:** \$5M over 4 years (T&M)

- 2 Fully Cross-Functional Teams
- BAH Personnel: 8 (1 PM, 3 Devs, 4 BA/Testers)
- Navy Personnel: 5 (1 PgM, 3 Officers, 1 Analyst)

**Output: *Project ROI: 50***

- QA/QC avoided \$20M/yr. in net-loss projects
- Improved selection by \$30M/yr. annualized returns
- Modeled investments with 95% accuracy by year 3
- BAH sole sourced the \$10M/yr Renewables Program

# Proof Agile Works: Spacecraft to Supercomputers



*Condor Cluster  
Supercomputer,  
Air Force 2010*

*NASA led the Faster,  
Better, Cheaper (FBC)  
initiative in the 1990s*

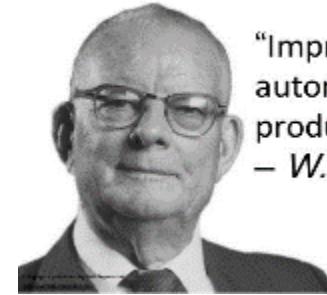


*Image Source: <http://www.zdnet.com/article/what-the-dods-playstation-powered-condor-cluster-means-for-the-future-of-supercomputing/>*



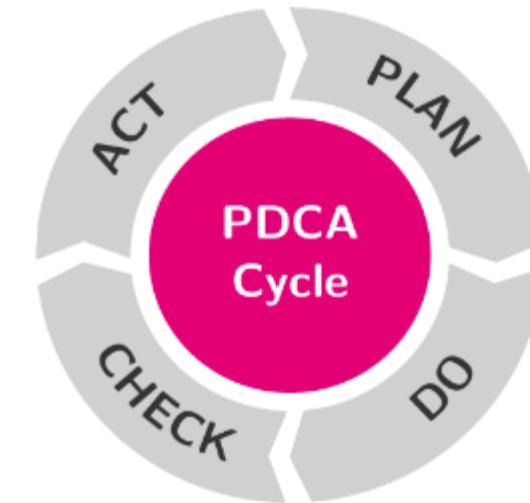
# Spiraling Away from Waterfall: a Total Quality Revolution Evolution of Agile

# Total Quality Management (TQM)



"Improve quality, you automatically improve productivity."  
– W. Edwards Deming

<https://i.pinimg.com/736x/98/bc/1f/98bc1f7c7ce266dd7e2fe796be001285--teacher-w-edwards-deming.jpg>



- Improving quality decreases costs
- Must continuously improve (systems and people)
- Key is pride of workmanship, cross-functional teams, and trust
- Plan – Do – Check – Act (PDCA)

*Proof it works: turned around Ford Motors in 1986 from \$B losses to first profits in years*

# Toyota Production System (TPS)



[https://c1.staticflickr.com/9/8110/8472007819\\_485415e875\\_b.jpg](https://c1.staticflickr.com/9/8110/8472007819_485415e875_b.jpg)



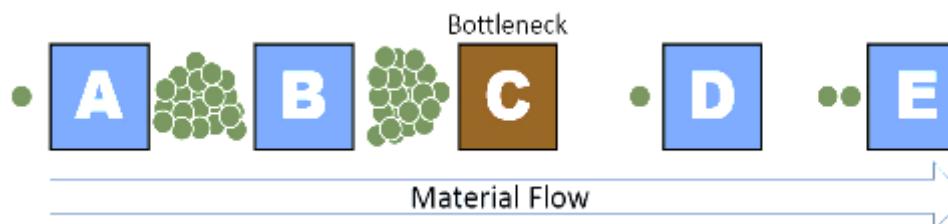
<https://lifehacker.com/productivity-101-how-to-use-personal-kanban-to-visualize-1687948640>

## Toyota Production System (TPS) – Taichii Ohno and Lean (1980s - Present)

- Eliminate 7 Wastes - Movement, Inventory, Motion, Waiting, Overproduction, Over-processing, Defects
- Small Batches – addresses most of the waste – *Kanban!*
- Continuous Improvement w/ Fixed Reporting Schedules & Metrics (KPIs)

*Proof it works: Toyota's a Top 3 Car Manufacturer with 70% employee satisfaction  
....Note that employee satisfaction is only 30% avg. Nationally*

# Theory of Constraints (TOC)



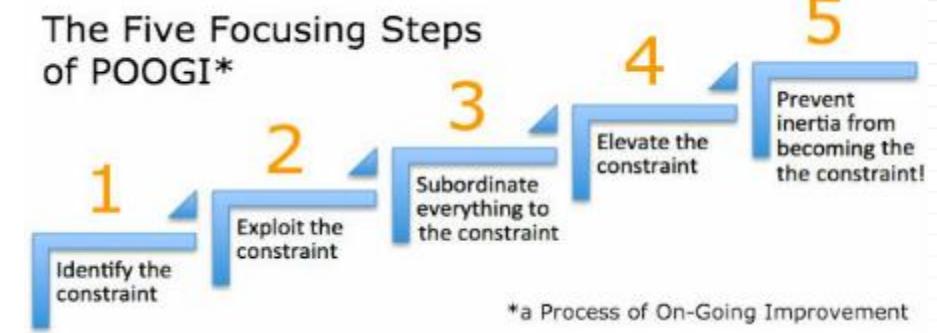
[https://en.wikipedia.org/wiki/Bottleneck\\_\(production\)](https://en.wikipedia.org/wiki/Bottleneck_(production))

[https://en.wikipedia.org/wiki/Eliyahu\\_M.\\_Goldratt](https://en.wikipedia.org/wiki/Eliyahu_M._Goldratt)

## Theory of Constraints (TOC) – Eli Goldratt (1980s to Present)

- Optimize “System Throughput” not “Cost Centers” towards a **Goal**
- Five Focusing Steps to Exploit System Constraints (Physical, Paradigm, Policy, Market)

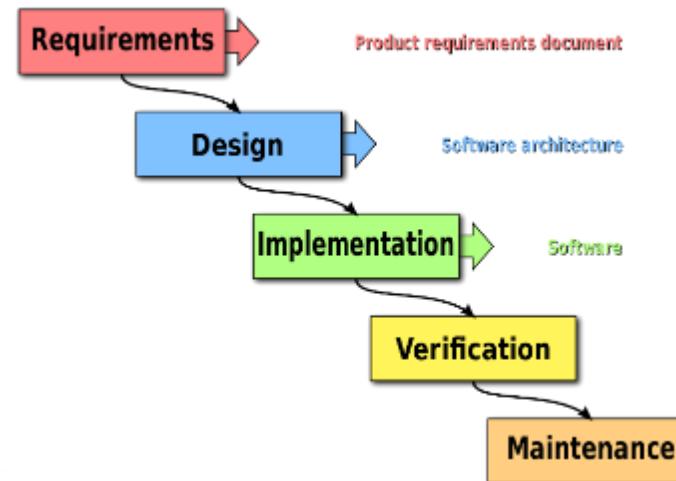
*Proof it works: BP used TOC to save \$200M and rapidly clean 10,000 boats after Gulf Oil Spill*



\*a Process of On-Going Improvement

<https://www.tocinstitute.org/five-focusing-steps.html>

# The Waterfall Mistake



*The average Software Project had a 10% success rate in the 1970s*

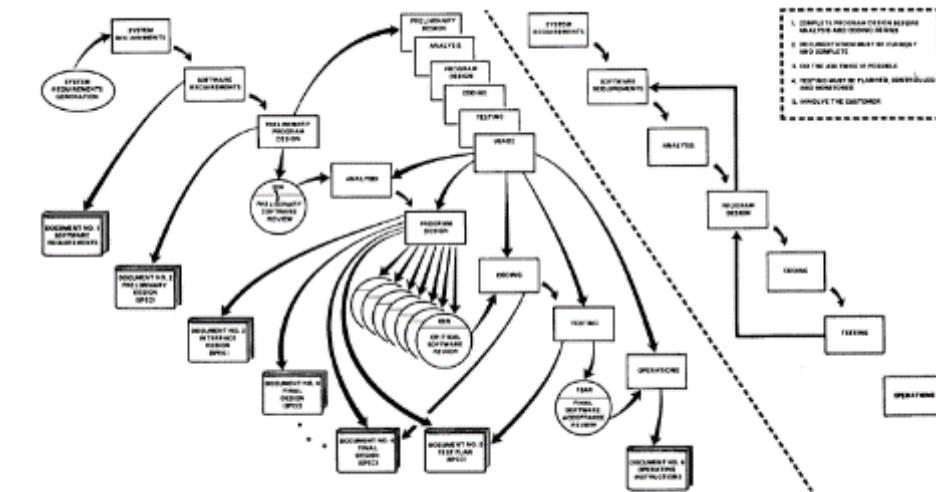


Figure 10. Summary

*Waterfall model probably the most costly mistake in the world*

<http://valueatwork.se/waterfall-model-probably-the-most-costly-mistake-in-the-world/?lang=en>

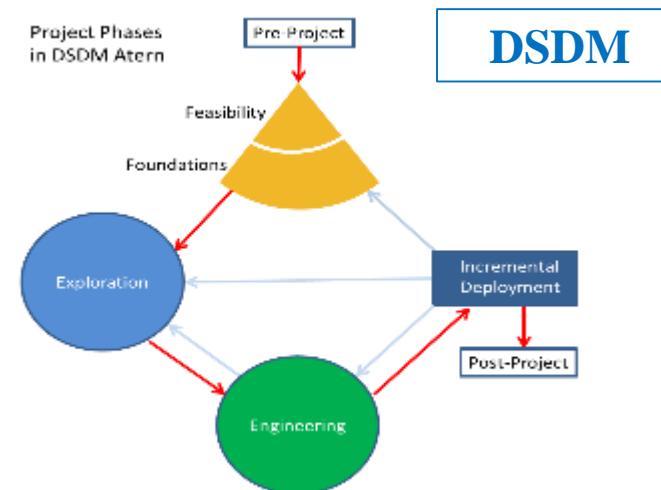
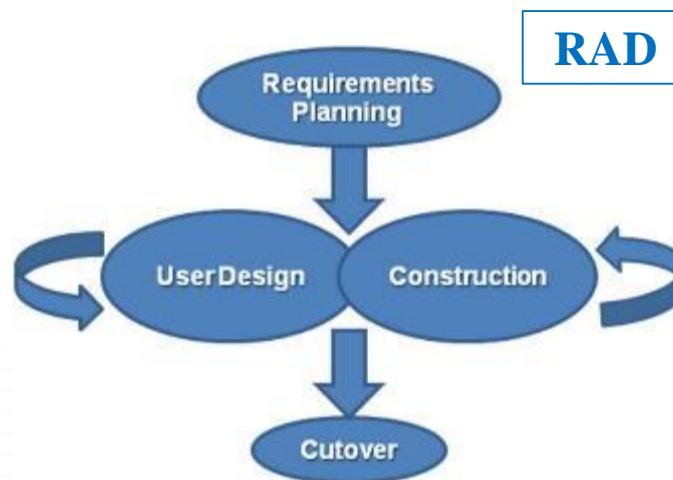
By the 1980s “Waterfall” was the predominant methodology, but it was a poor fit for the immaturity of the software development world (although embraced by DoD until 1996)

.....*Tom Cargill of Bell Labs said it all with his “Ninety-Ninety” Rule said it all:*

The first 90 percent of the code accounts for the first 90 percent of the development time.

The remaining 10 percent of the code accounts for the other 90 percent of the development time.

# Spiral to Scrum



In response to failure rates as high as 90%, “iterative development” was born:

- Rapid Application Development (RAD)
- Dynamic System Development Methodology (DSDM)
- Extreme Programming (XP)

1970s - 1980s

1980s - 1990s

1990s - 2000s



# Spiral to Scrum

## Key Tenants of Iterative Development:

- ***Consolidated Up-Front Planning*** - single “Systems Design” phase with Stakeholders
- ***Iterative Development*** – Users Propose and Test Product Throughout Development
- ***Timeboxes*** - Emphasizes On-Time Delivery
- ***User Stories*** - Emphasizes Business Needs, Not Tech Specs
- ***Test-Driven Development*** - Incorporation of “best practices”

## 2013 Cross-Industry Study

173 Industry Respondents

<http://www.ambisoft.com/surveys/success2013.html>

Agile	Traditional
64% Successful	49% Successful
28% Challenged	33% Challenged
8% Failed	18% Failed

<https://clearcode.cc/blog/agile-vs-waterfall-method/>



PROJECT MANAGEMENT  
CENTER FOR EXCELLENCE

A.J. CLARK SCHOOL OF ENGINEERING  
Civil & Environmental Engineering Department

# How Netflix Wins! Commercial Case Study

# Netflix Case Study

## SPEED WINS!

Keynote: Velocity and Volume  
(or Speed Wins) by Adrian Cockcroft

Adrian Cockcroft names four (4) things required to turn Netflix from a manufacturing company into an web-centric large-scale business:

1. Culture of Innovation
  2. Data and Analytics
  3. Decentralized Decisions
  4. Agile and Self-Service Deploy
- *respond to opportunities*
  - *compare alternatives*
  - *assign resources quickly*
  - *freedom & responsibility culture*



*Watch the speech here:*

<https://www.youtube.com/watch?v=wyWI3gLpB8o>



# 18F to the Rescue! Government Case Study



# 18F Case Study

## *General Services Administration (GSA) supports CA Social Services...*

In 2015, the State of California began a process to modernize their child welfare services case management system.

- Used by more than 20,000 social workers
- Track and manage the more than 500,000 cases of child abuse and neglect annually

18F worked with California's Department of Social Services and Office of Systems Integration to add:

- modular contracting,
- agile development,
- user-centered design, and
- open source practices

This project is still in the early stages, but this change in strategy has started to produce greater vendor competition, cost savings, a vastly improved end product, and a better contracting experience.



### Agile development processes

Learn how to use Agile development methodologies for your project.

[Read more](#)

### Embracing DevOps

Learn about DevOps practices and tools for delivering high quality software

[Read more](#)

### Managing modular contracts

Learn how to break your project into smaller, less-risky modules.

[Read more](#)



### Open source software

Learn how leveraging open source can help your next procurement.



### Building prototypes

Learn how to use prototypes to reduce risk and ambiguity before issuing an RFP



### Using COTS solutions

Demystify COTS and learn whether these solutions are right for your project.



<https://www.youtube.com/watch?v=lNSmF7-xisU>



You've just completed Module 1 of  
*Applied Scrum for Agile Project  
Management*

Thank You!

[CLICK here to go to the EdX Course Page](#)





# Week 2. Agenda

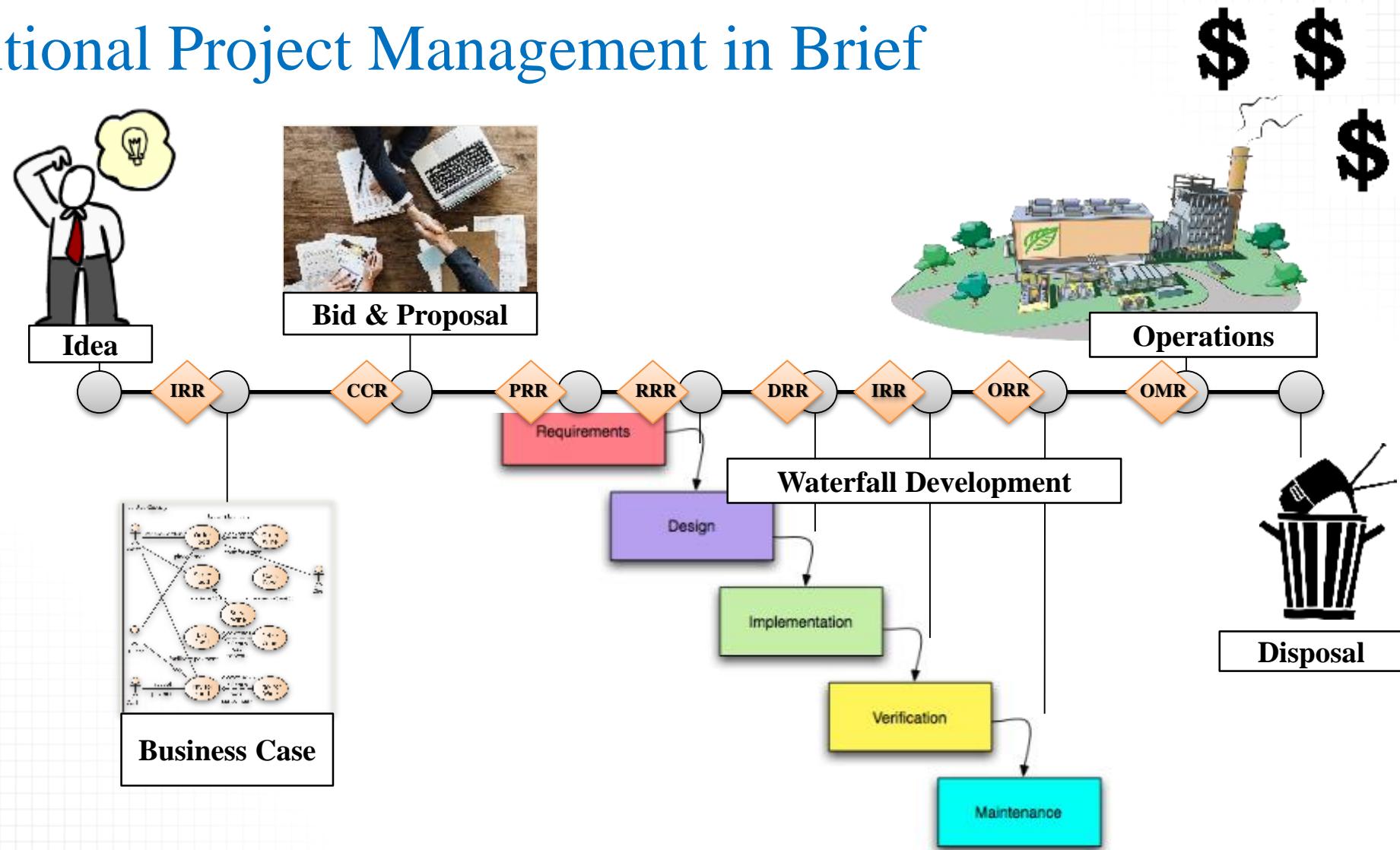
- 1. Simple PM Methods**
- 2. Approaching the Triple Cost Constraints**
- 3. Comparing Methods – Industries**
- 4. Comparing Methods – Customers**
- 5. Comparing Methods – Engineering**



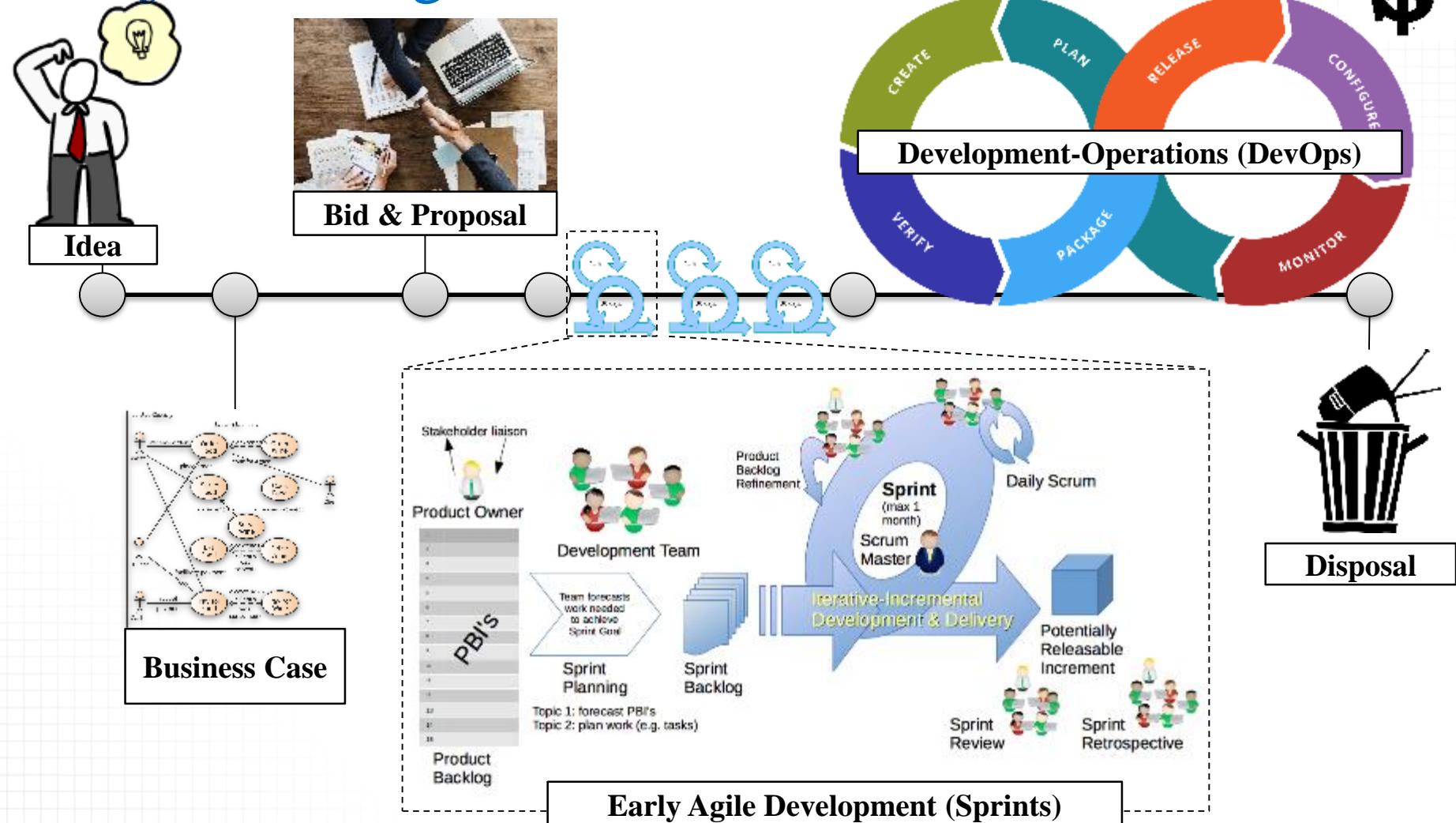
# Why There Are Three Costs and What to Do about It!

## Simple PM Methods

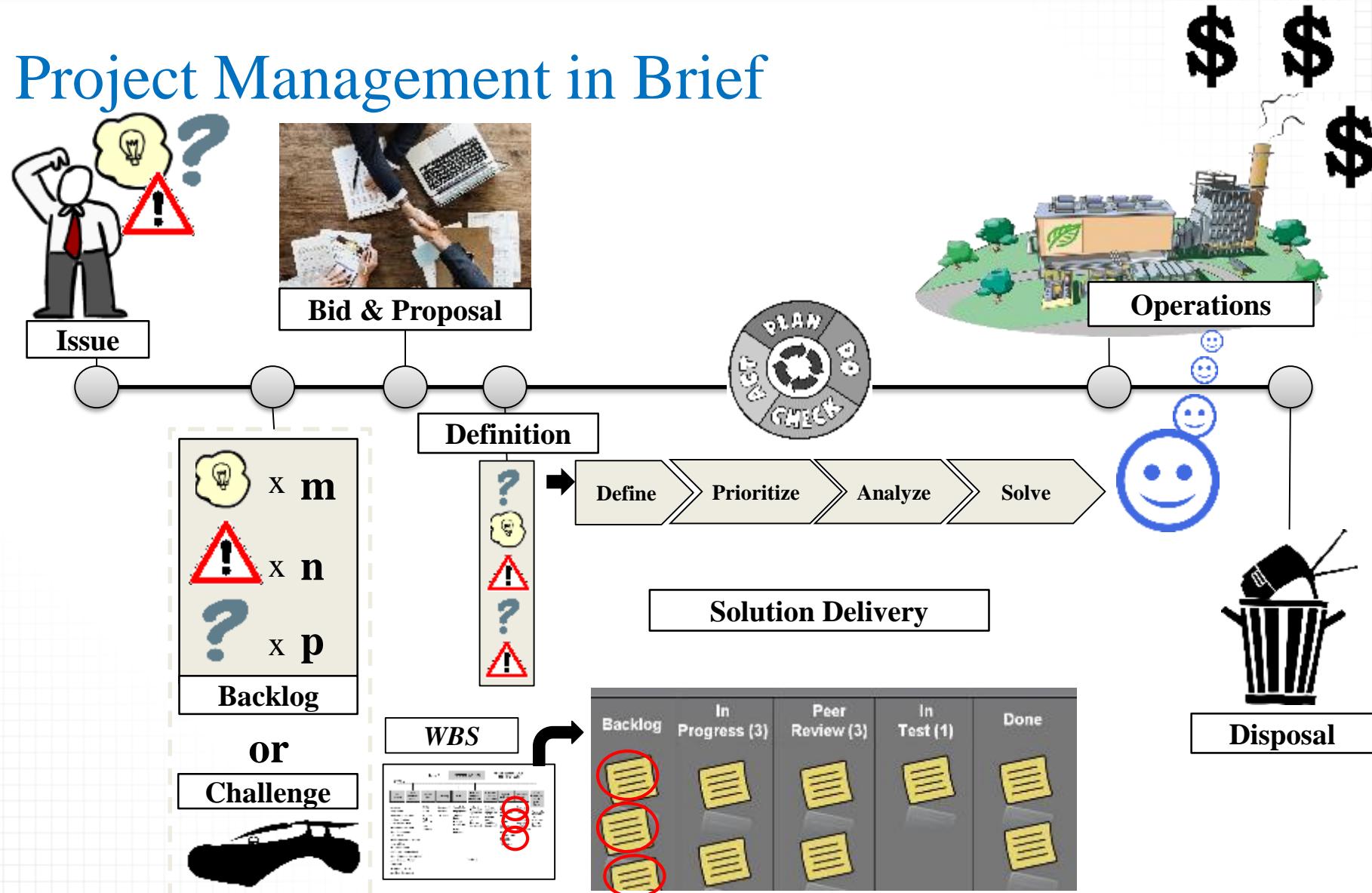
# Traditional Project Management in Brief



# Agile Project Management in Brief



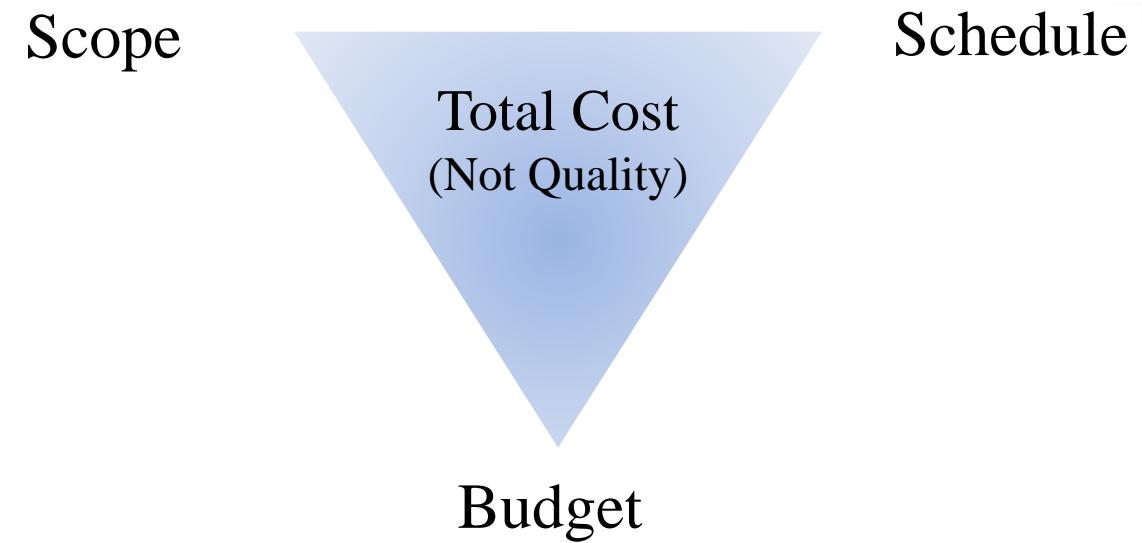
# Lean Project Management in Brief





# Why There Are Three Costs and What to Do about It! Approaching the Triple Cost Constraint

# Exploring the Triple Cost Constraint

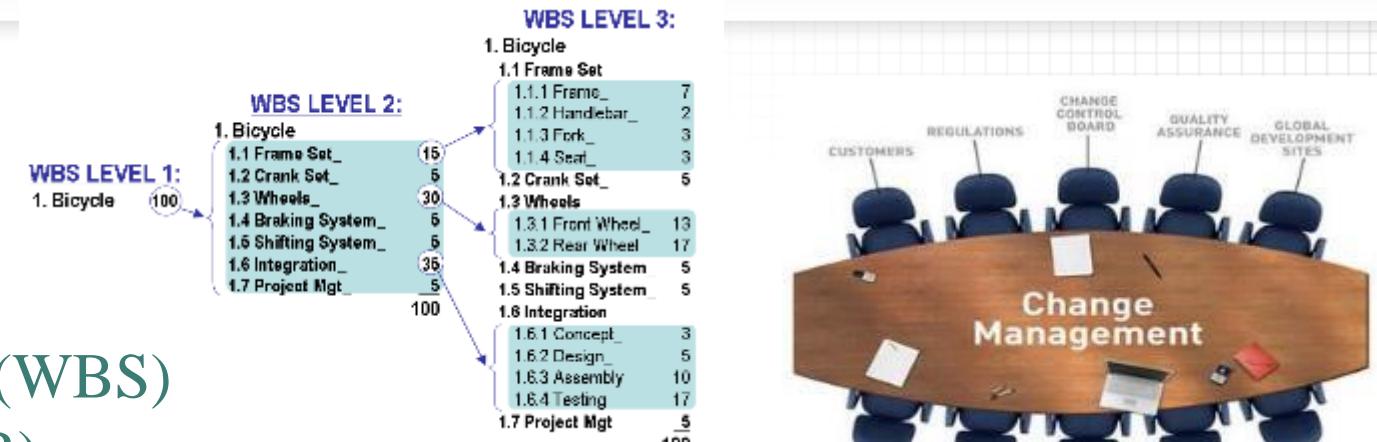


	Agile	Traditional	Lean
Adjust	Scope	Budget	Schedule
Requires	Trust	Efficiency	Expertise
Goal	Speed	Predictability	Innovation

# Controlling Scope

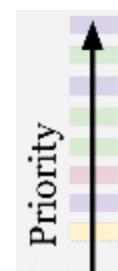
## Traditional

- Work Breakdown Structure (WBS)
- Change Control Board (CCB)

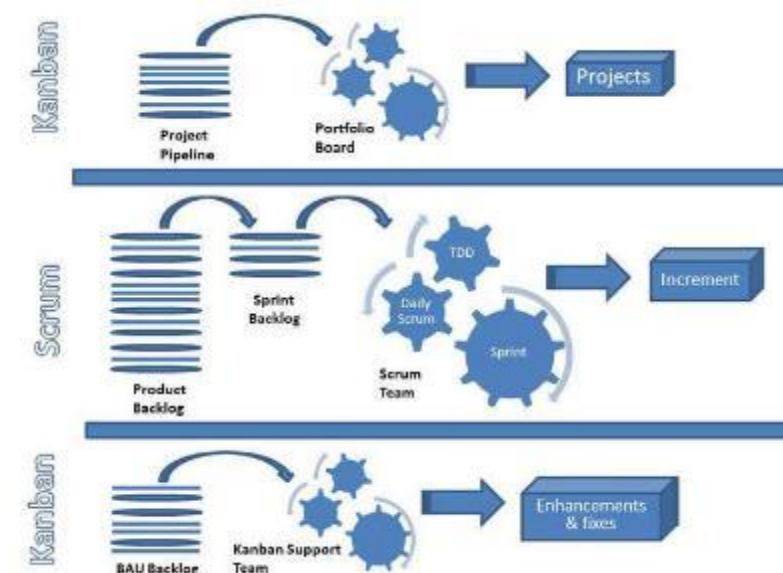


## Lean

- Tickets
- Requests



OR



# Controlling Schedule

## Traditional

- Duration estimates and Schedules
- Program Evaluation & Review
- Technique (PERT)
- Critical Path Method (CPM)

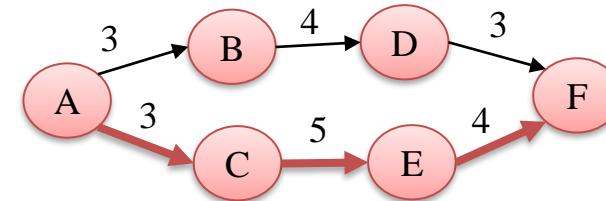
## Lean

- Kanban & Queues
- Service Agreements

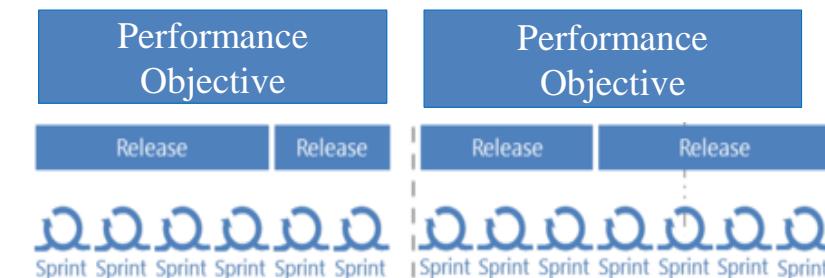
## Agile

- Timeboxes
- Releases and Roadmaps

Task	Duration	Requires
A	3	
B	4	A
C	5	A
D	3	B
E	4	C
F	0	D and E



Priority	Business Impact	Response
Critical	Shutdown	Hours
Major	Costly Delays	Days
Minor	Frustration	Weeks



# Controlling Budget

## Traditional

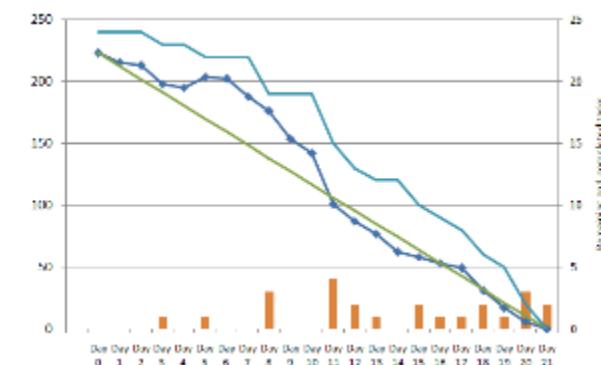
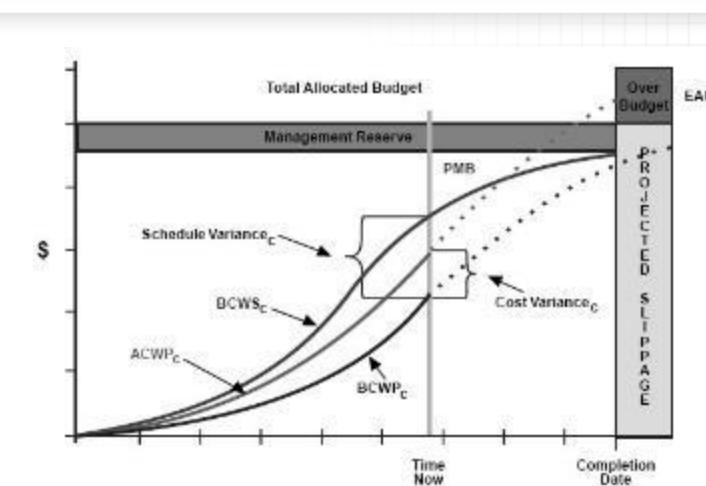
- Earned Value Management (EVM)
- Cost Centers

## Lean

- Service and Severity Levels
- Key Performance Indicators (KPIs)

## Agile

- Return on Investment (ROI)
- Burndown Charts





# Matching Methods with Size (for Some) Comparing Methods: Industries



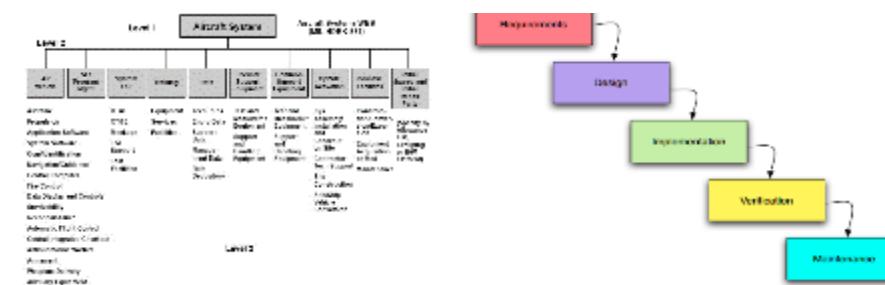
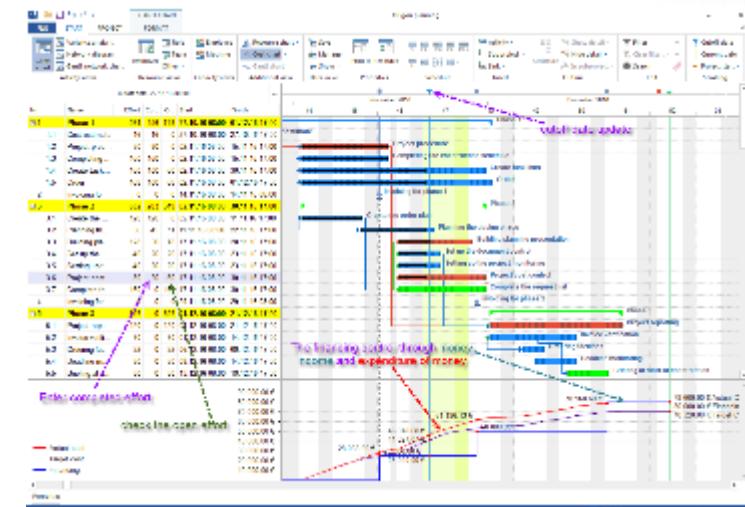
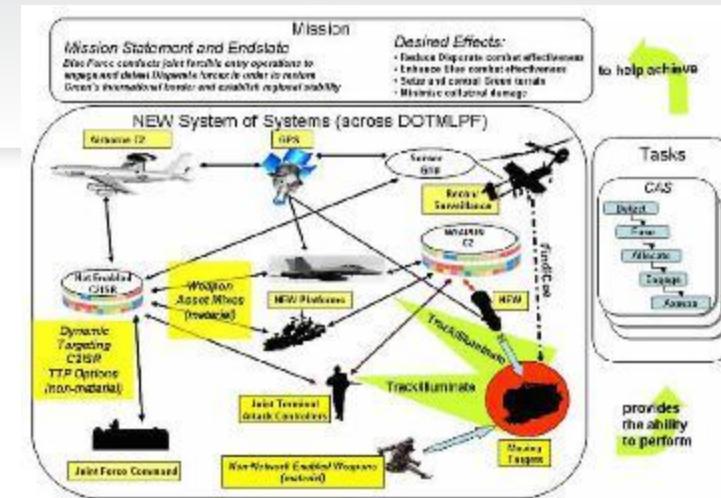
# Industry Insights: What Works....

	<b>Traditional</b>	<b>Agile</b>	<b>Lean</b>
<b>Project Size</b>	Large	Medium	Small
<b>Industries</b>	Construction Military Government / Policy Relocation	Information Technology Product Development Consulting Operations	Sales Customer Support Legal Research & Development
<b>Planning*</b>	Master Schedules	Releases	Backlogs (Prioritized Lists)
<b>Sourcing</b>	Efficiency	Trust	Expertise
<b>Goals</b>	Predictable (Low Cost)	Speed (Maximize ROI)	Innovation (Problem Solve)



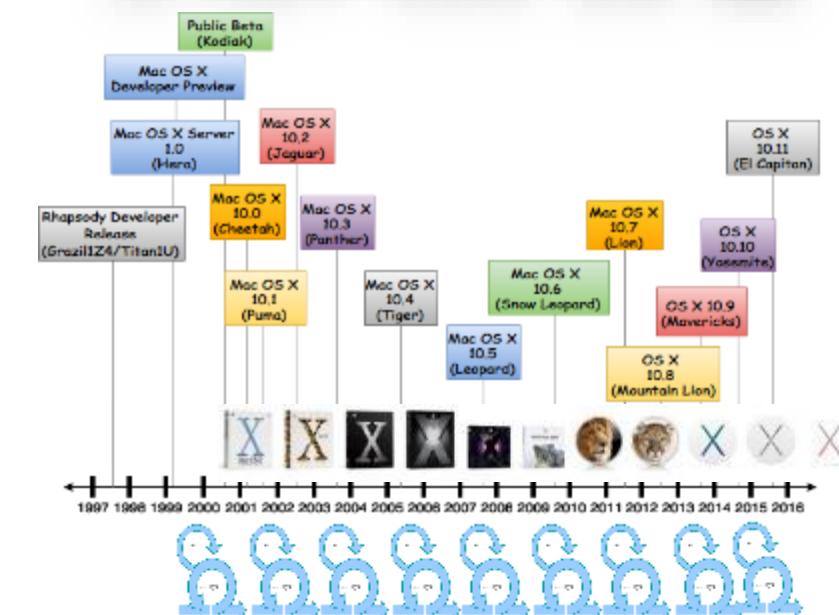
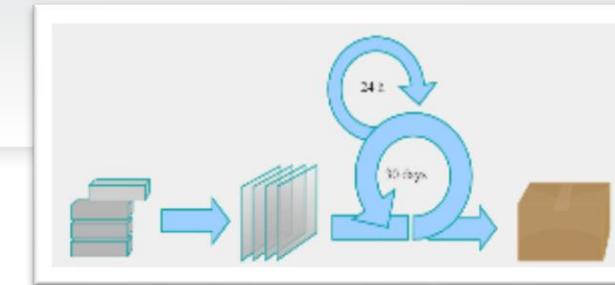
# Traditional Projects

Traditional	
<b>Project Size</b>	Large
<b>Industries</b>	Construction Military Government / Policy Relocation
<b>Planning*</b>	Master Schedules
<b>Sourcing</b>	Efficiency
<b>Goals</b>	Predictable (Low Cost)



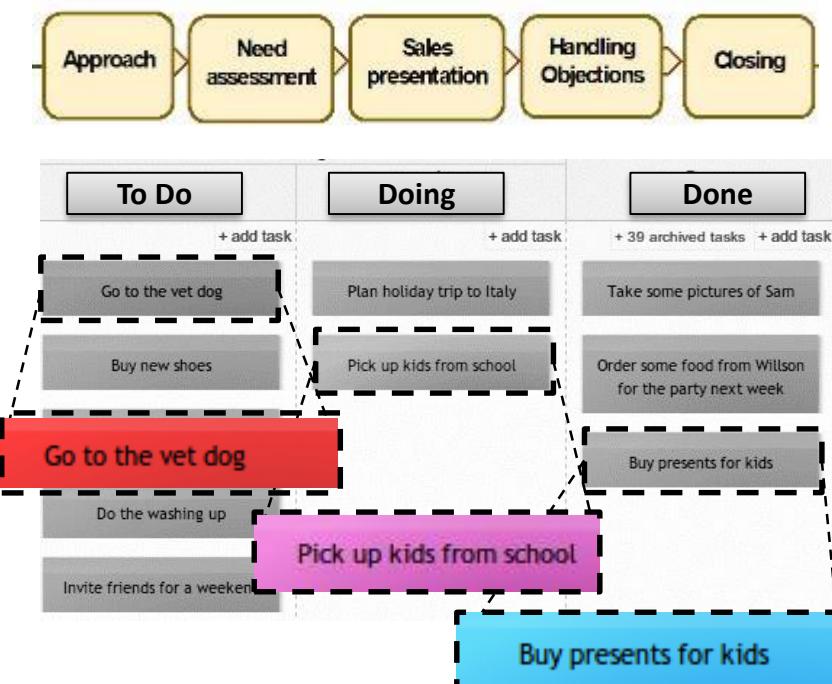
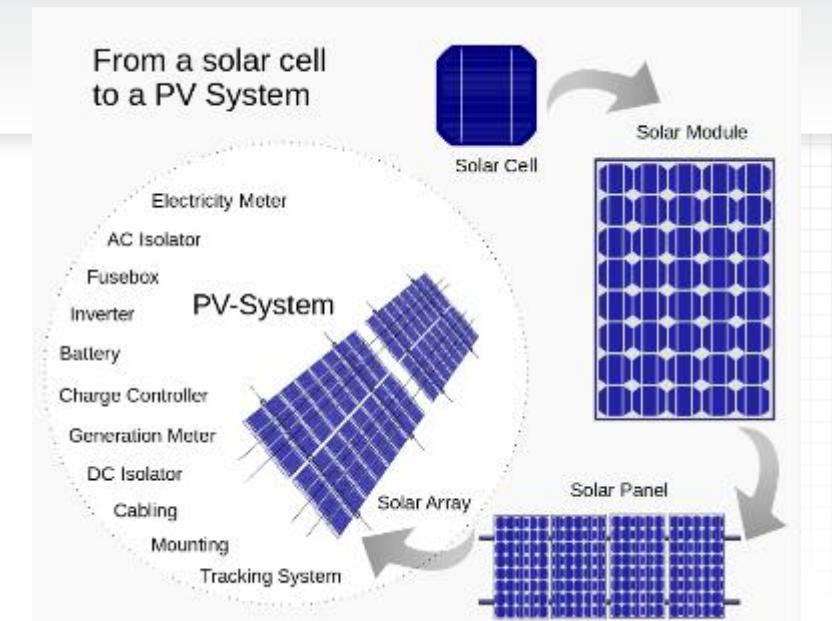
# Agile Projects

	<b>Agile</b>
<b>Project Size</b>	Medium
<b>Industries</b>	Information Technology Product Development Consulting Operations
<b>Planning*</b>	Releases
<b>Sourcing</b>	Trust
<b>Goals</b>	Speed (Maximize ROI)



# Lean Projects

	<b>Lean</b>
<b>Project Size</b>	Small
<b>Industries</b>	Sales Customer Support Legal Research & Development
<b>Planning*</b>	Backlogs (Prioritized Lists)
<b>Sourcing</b>	Expertise
<b>Goals</b>	Innovation (Problem Solve)



# Industry Insights: What Works....

	<b>Traditional</b>	<b>Agile</b>	<b>Lean</b>
<b>Project Size</b>	Large	Medium	Small
<b>Industries</b>	Construction Military Government / Policy Relocation	Information Technology Product Development Consulting Operations	Sales Customer Support Legal Research & Development
<b>Planning*</b>	Master Schedules	Releases	Backlogs (Prioritized Lists)
<b>Sourcing</b>	Efficiency	Trust	Expertise
<b>Goals</b>	Predictable (Low Cost)	Speed (Maximize ROI)	Innovation (Problem Solve)

**Why?**      **Why?**      **Why?**



Say What? It is NOT about Product?  
**Comparing Methods: Customers**



# Industry Customer Characteristics

	<b>Traditional</b>	<b>Agile</b>	<b>Lean</b>
<b>Project Size</b>	Large	Medium	Small
<b>Industries</b>	Construction Military Government / Policy Relocation	Information Technology Product Development Management Consulting Operations	Sales Customer Support Legal R&D
<b>Customer Size</b>	>250 participants	Up to 250 participants	Up to 10
<b>Customer Communication</b>	Representatives Large, Facilitated Meetings	Part of the Team Small Meetings	On-Call Ticketing / Request
<b>Payment Method</b>	Firm Fixed Price / Custom Pricing (Quote)	Time & Materials / Retail Purchase (Paid)	Cost-Plus / Subscription (SLA)

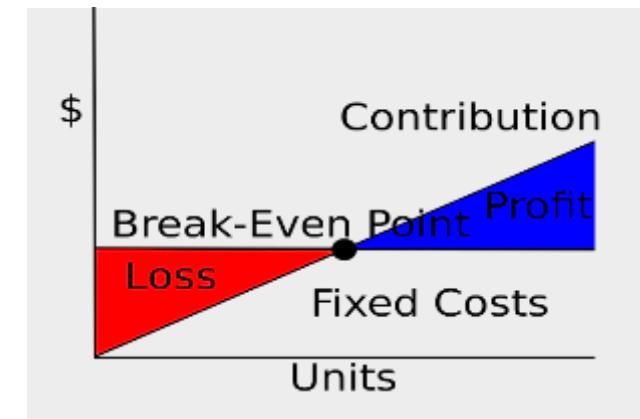
# Traditional Customer Management

	<b>Traditional</b>
<b>Project Size</b>	Large
<b>Industries</b>	Construction Military Government / Policy Relocation
<b>Customer Size</b>	>250 participants
<b>Customer Communication</b>	Representatives Large, Facilitated Meetings
<b>Payment Method</b>	Firm Fixed Price / Custom Pricing (Quote)



# Agile Customer Management

	<b>Agile</b>
<b>Project Size</b>	Medium
<b>Industries</b>	Information Technology Product Development Management Consulting Operations
<b>Customer Size</b>	Up to 250 participants
<b>Customer Communication</b>	Part of the Team Small Meetings
<b>Payment Method</b>	Time & Materials / Retail Purchase (Paid)



# Lean Customer Management

	<b>Lean</b>
<b>Project Size</b>	Small
<b>Industries</b>	Sales Customer Support Legal R&D
<b>Customer Size</b>	Up to 10
<b>Customer Communication</b>	On-Call Ticketing / Request
<b>Payment Method</b>	Cost-Plus / Subscription (SLA)



*Sure I can help you...  
Now...how did it happen?*





# Industry Customer Characteristics

	<b>Traditional</b>	<b>Agile</b>	<b>Lean</b>
<b>Project Size</b>	Large	Medium	Small
<b>Industries</b>	Construction Military Government / Policy Relocation	Information Technology Product Development Management Consulting Operations	Sales Customer Support Legal R&D
<b>Customer Size</b>	>250 participants	Up to 250 participants	Up to 10
<b>Customer Communication</b>	Representatives Large, Facilitated Meetings	Part of the Team Small Meetings	On-Call Ticketing / Request
<b>Payment Method</b>	Firm Fixed Price / Custom Pricing (Quote)	Time & Materials / Retail Purchase (Paid)	Cost-Plus / Subscription (SLA)



# Matching Method with Size Comparing Methods: Engineering

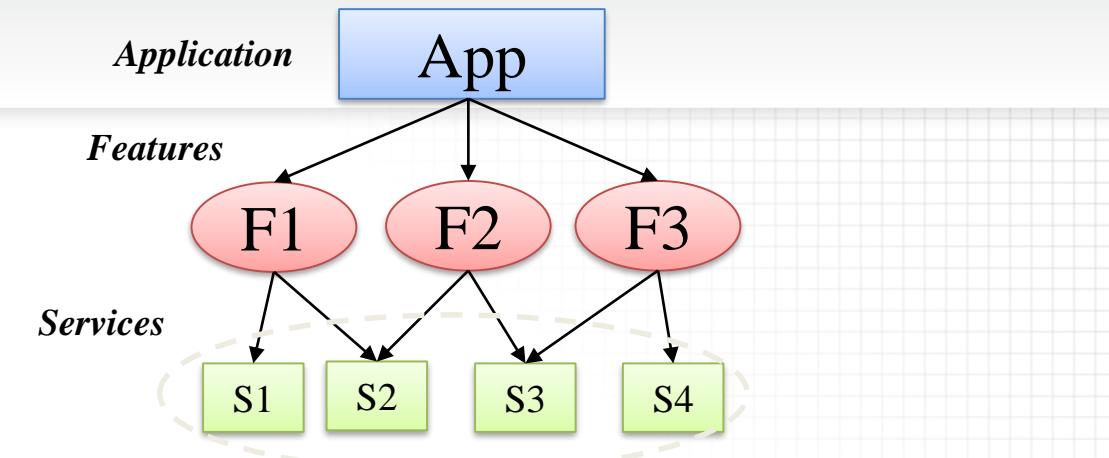


# Industry Engineering Characteristics

	<b>Traditional</b>	<b>Agile</b>	<b>Lean</b>
<b>Project Size</b>	Large	Medium	Small
<b>Industries</b>	Construction Military Government / Policy Relocation	Information Technology Product Development Management Consulting Operations	Sales Customer Support Legal R&D
<b>Design</b>	Dependent / Coupled	Independent / Decoupled	Constrained / Evolutionary
<b>Teams</b>	Departmental	Matrixed / Projectized	Emergent (Ad Hoc)
<b>Development</b>	Linear	Iterative	Incremental
<b>Integration / Testing</b>	End Phase	Continuous	When Possible
<b>Closing</b>	3rd Party Acceptance	Team Acceptance	Customer Acceptance

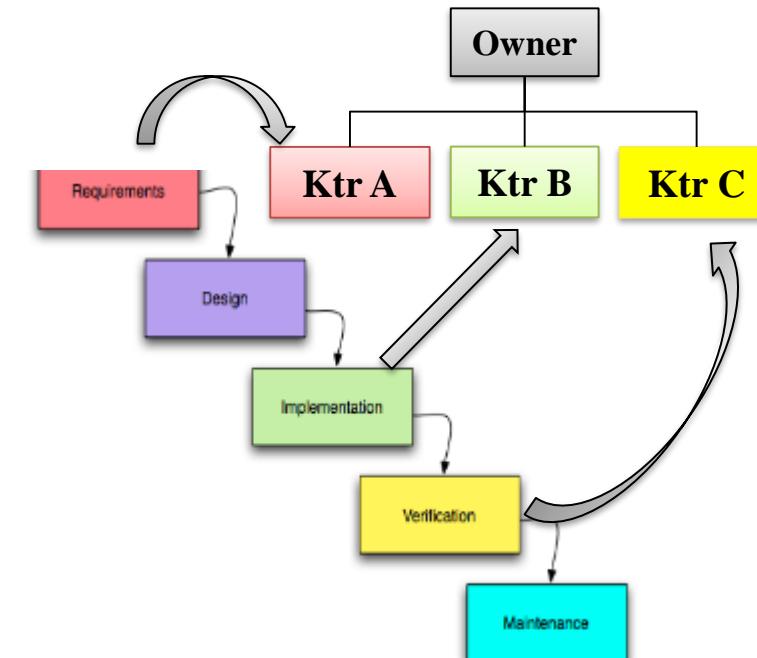
# Traditional Engineering

	<b>Traditional</b>
<b>Project Size</b>	Large
<b>Industries</b>	Construction Military Government / Policy Relocation
<b>Design</b>	Dependent / Coupled
<b>Teams</b>	Departmental
<b>Development</b>	Linear
<b>Integration / Testing</b>	End Phase
<b>Closing</b>	3rd Party Acceptance



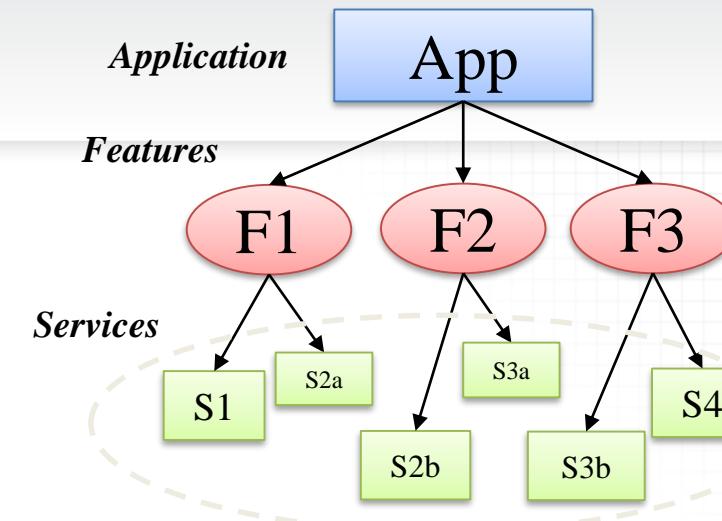
*High reuse of components means a “Tightly Coupled” design.*

- *Small changes have big costs*
- *Lower costs if no changes*



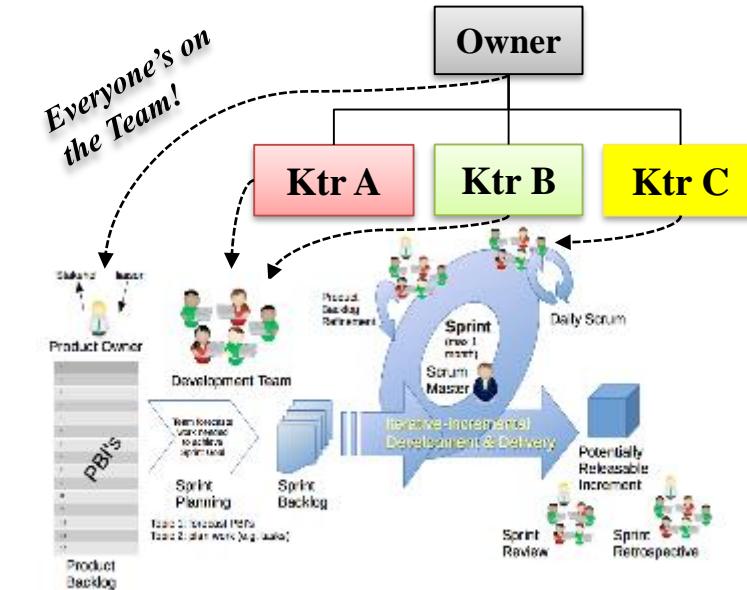
# Agile Engineering

	<b>Agile</b>
<b>Project Size</b>	Medium
<b>Industries</b>	Information Technology Product Development Management Consulting Operations
<b>Design</b>	Independent / Decoupled
<b>Teams</b>	Matrixed / Projectized
<b>Development</b>	Iterative
<b>Integration / Testing</b>	Continuous
<b>Closing</b>	Team Acceptance



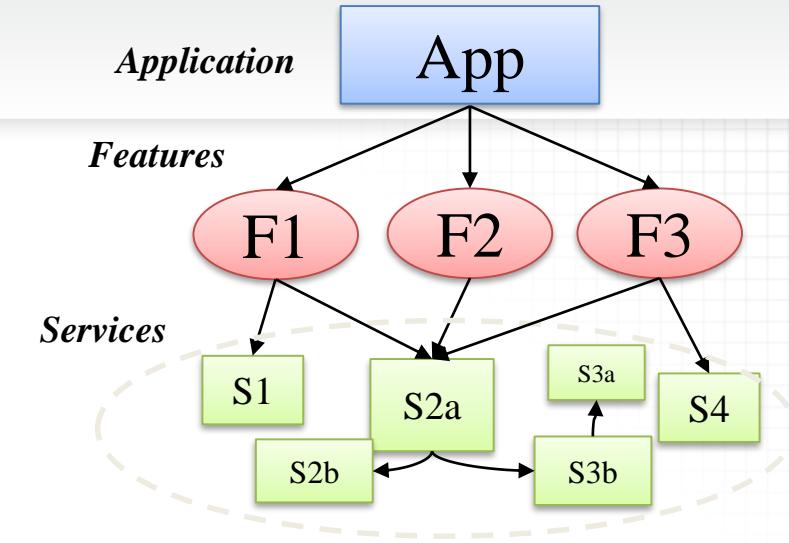
*Low or no reuse of components means a “Decoupled” design.*

- *Small changes have small costs*
- *Can release single Features quickly*
- *Higher cost if no expected changes*



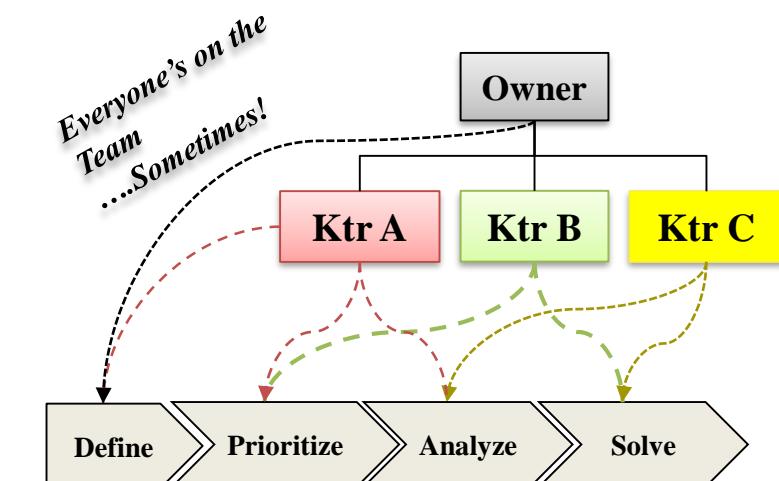
# Lean Engineering

	<b>Lean</b>
<b>Project Size</b>	Small
<b>Industries</b>	Sales Customer Support Legal R&D
<b>Design</b>	Constrained / Evolutionary
<b>Teams</b>	Emergent (Ad Hoc)
<b>Development</b>	Incremental
<b>Integration / Testing</b>	When Possible
<b>Closing</b>	Customer Acceptance



*Lean designs “Evolve” as each feature is developed*

- *Uses “just enough” feature by feature*
- *Reuses as possible to lower costs*
- *Often results in complex final product*

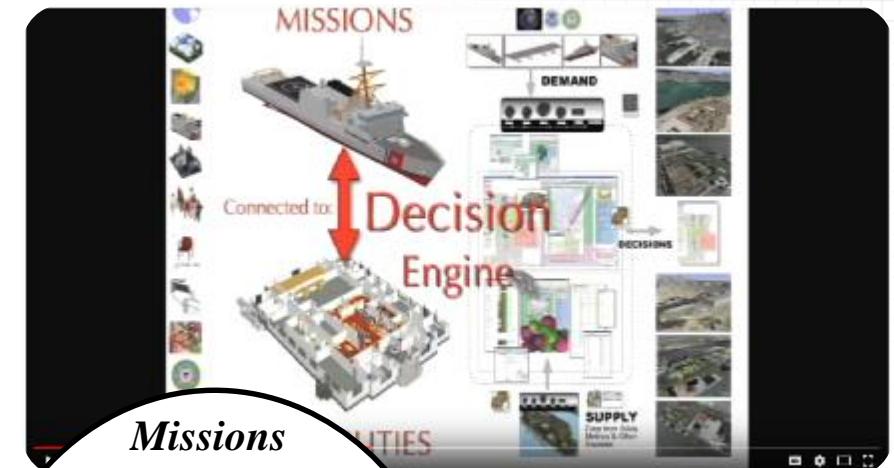


## How Industries are Changing Now...

IT is turning traditional and lean industries into Agile ones...

- Building Information Modeling (BIM)
- Modular Acquisitions
- Online Legal Products
- “Everything as a Service”

*Onuma Planning System introduced  
“BIMStorm”  
Back in 2008!*



*Missions  
drive demand for  
Buildings...*

We can simulate  
buildings down to the  
nuts and bolts...

*Buildings can be  
dynamic, too!*

*Watch and Learn More:  
<http://www.onuma.com/services/BimStorm.php>*





# Industry Engineering Characteristics

	<b>Traditional</b>	<b>Agile</b>	<b>Lean</b>
<b>Project Size</b>	Large	Medium	Small
<b>Industries</b>	Construction Military Government / Policy Relocation	Information Technology Product Development Management Consulting Operations	Sales Customer Support Legal R&D
<b>Design</b>	Dependent / Coupled	Independent / Decoupled	Constrained / Evolutionary
<b>Teams</b>	Departmental	Matrixed / Projectized	Emergent (Ad Hoc)
<b>Development</b>	Linear	Iterative	Incremental
<b>Integration / Testing</b>	End Phase	Continuous	When Possible
<b>Closing</b>	3rd Party Acceptance	Team Acceptance	Customer Acceptance



You've just completed **Module 2 of**  
*Applied Scrum for Agile Project  
Management*

Thank You!

[CLICK here to go to the EdX Course Page](#)





# Week 3. Agenda

- 1. Scrum Team Formation**
- 2. Three-Part User Story**
- 3. Sprint Planning**
- 4. Sprint Development**
- 5. Sprint Retro & Review**



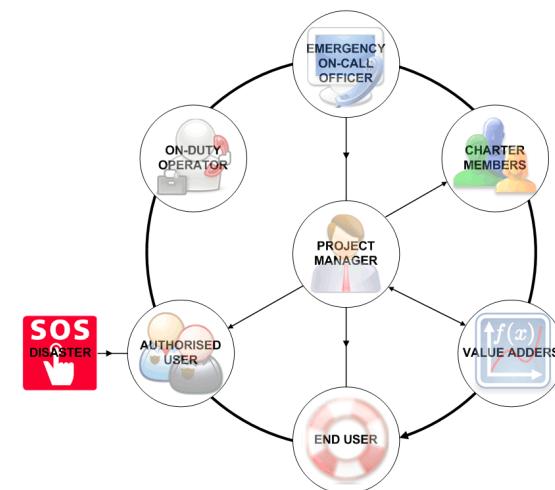
# Chartering the Right Course from the Beginning! **Scrum Team Formation**

# Project Charter

- **Project Objectives** - what the sponsors and/or customers expect from this project
- **Stakeholders** - who “has a stake” from sponsors to customers and why
- **Constraints** - what must the project also do or not do to achieve the objective
- **Risks** - what are major risks: internal vs. external, business vs, technical
- **Definition of Done** - the agreement on how work is closed



<https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTG6a-TR4eYNjDQm3f91JgEXuDZoklCGei3CMiTJv3Buis29Vx6IA>



[https://upload.wikimedia.org/wikipedia/commons/9/99/Charter\\_Activation\\_Scheme.png](https://upload.wikimedia.org/wikipedia/commons/9/99/Charter_Activation_Scheme.png)

## Scrum Team Members

- **Product Owner** - person responsible for managing the Product Backlog
- **Scrum Master** - person responsible for facilitating
- **Development Team Member** - person on the team building the product



<https://pxhere.com/en/photo/868229>



[https://c1.staticflickr.com/8/7193/7008611307\\_8da73afaff\\_b.jpg](https://c1.staticflickr.com/8/7193/7008611307_8da73afaff_b.jpg)



[https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQ-KcFqr\\_DgOYfbx3X\\_au6Z3iqOHyM-KV9FAkqoVy8rUQ3AneQ5iw](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQ-KcFqr_DgOYfbx3X_au6Z3iqOHyM-KV9FAkqoVy8rUQ3AneQ5iw)

# Product Owner Variations

*Business Rep*



[https://cdn.pixabay.com/photo/2016/04/15/18/05/computer-1331579\\_960\\_720.png](https://cdn.pixabay.com/photo/2016/04/15/18/05/computer-1331579_960_720.png)

*Architect*



<https://www.maxpixel.net/Business-Man-Architect-Work-Businessman-Architektin-1816217>

*Owner*



<https://pixabay.com/en/avatar-icon-document-entrepreneur-1789663/>

# Scrum Master Variations

## *Project Manager*



<https://pixabay.com/en/agenda-comic-characters-gantt-2025762/>

## *Junior PM*



<https://pxhere.com/en/photo/1444293>

## *Business Analyst*



<https://pixabay.com/en/cartoon-character-idea-business-2948029/>

# Development Team Member Variations

## Architect



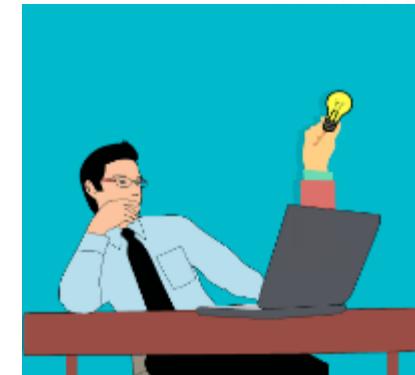
<https://www.maxpixel.net/Business-Man-Architect-Work-Businessman-Architektin-1816217>

## Tech Writer



<https://pxhere.com/en/photo/1199795>

## Business Analyst



<https://pixabay.com/en/cartoon-character-idea-business-2948029/>

## Support



<https://www.flickr.com/photos/rosenfeldmedia/11496789864>

## Developer



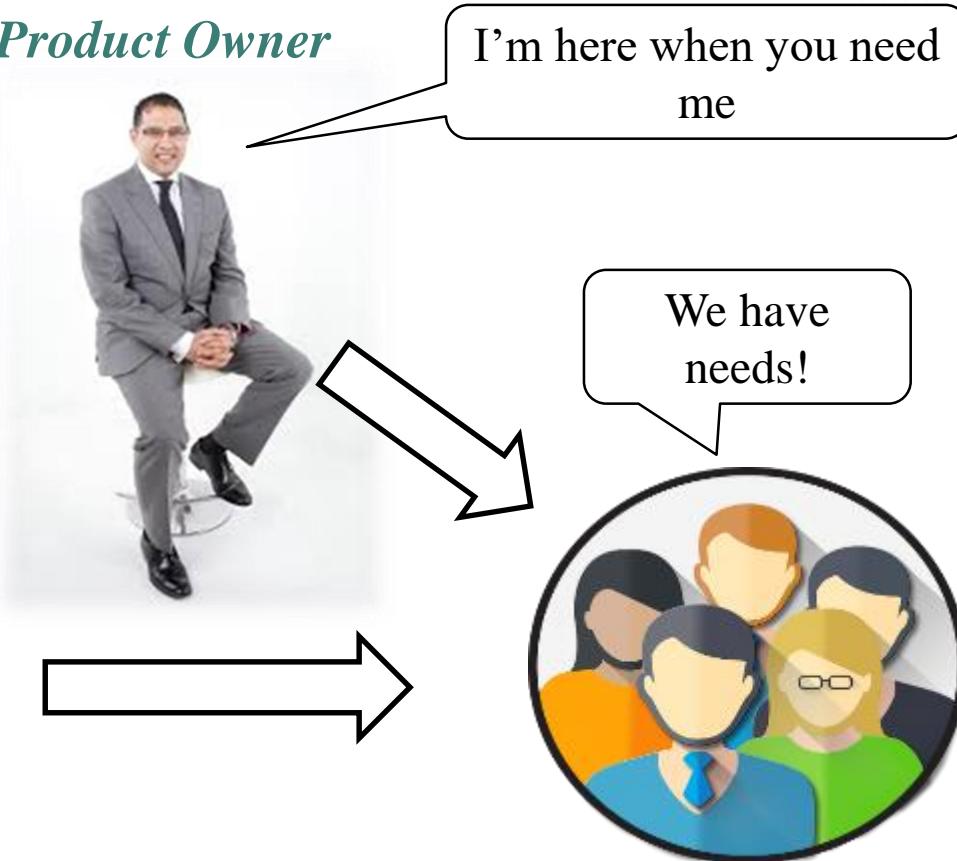
[https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQ-KcFqr\\_DgOYfbx3X\\_au6Z3iqOHyM-KV9FAkqoVy8rUQ3AneQ5iw](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQ-KcFqr_DgOYfbx3X_au6Z3iqOHyM-KV9FAkqoVy8rUQ3AneQ5iw)

# Benefits of a Dedicated Team



*Scrum Master*

*Product Owner*



*Stakeholders*



# Defining Work Meaningfully Three-Part User Story



# Part of a User Story

- There are three parts to a User Story:
  - 1) Value Statement
  - 2) Assumptions
  - 3) Acceptance Criteria
- The proper means of creating a Value Statement:

<i>As a...</i>	[Who]...
<i>I want to....</i>	[What Functionality]...
<i>in order to...</i>	[Why It's Important]...

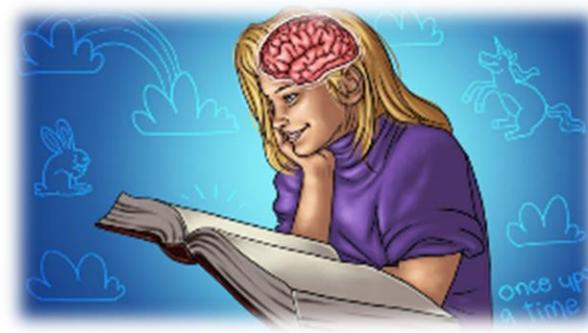
# Part of a User Story

## *Wrong Wants and Right Wants:*

- *I want to Login Using My User Name and Password -- WRONG!*
- *I want to access my account -- RIGHT!*

## *What's in a Why?*

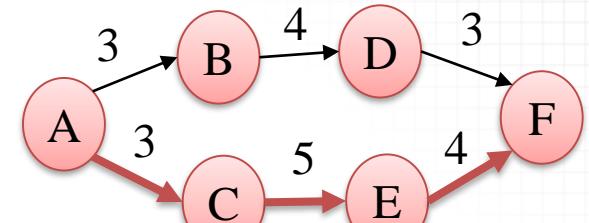
- To check notifications
- To assign work
- To check the rankings of my fantasy football team



*Image Source: <https://lifehacker.com/5965703/the-science-of-storytelling-why-telling-a-story-is-the-most-powerful-way-to-activate-our-brain>*

# Understanding Assumptions

- Captures less important value created by the User Story
- Captures detailing of Why the user story is important
- Identifies constraints from preceding or proceeding tasks, work, components, etc.
- Identifies all the standards, influences, reference architectures, etc.
- Captures other reasons “Why” this story might be important
- Can limit the Acceptance Criteria and the Value Statement - not just the Value Statement



[https://commons.wikimedia.org/wiki  
/File:Google\\_2015\\_logo.svg](https://commons.wikimedia.org/wiki/File:Google_2015_logo.svg)

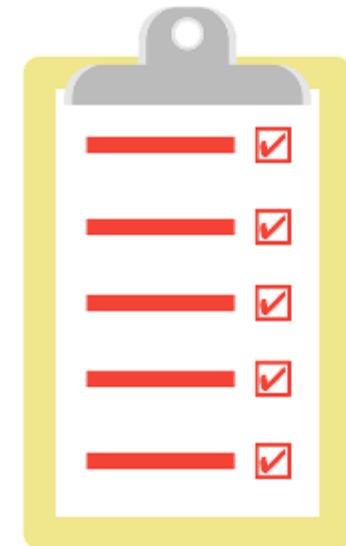
Procedural information that is in the ***Definition of Done*** does not have to be captured in Assumptions – it’s always repeated!

# Addressing the Acceptance Criteria

- Acceptance Criteria are NOT restatements of the Value Statement
- Should clearly define the primary use cases for testing
- Must specify any performance or loading that the product increment should meet
- Must be comprehensive in detailing all tests that will be run to close the story



[http://www.publicdomainfiles.com/show\\_file.php?id=13962379226067](http://www.publicdomainfiles.com/show_file.php?id=13962379226067)



<https://pixabay.com/en/photos/checkbox/>

# Definition of Done - One for All!

- Standard approvals
- Reviews by stakeholders
- Prototyping (if required)
- Documentation
- Design constraints



<https://www.maxpixel.net/Office-Map-Document-Empty-Folder-Full-Icon-2813518>



<https://www.flickr.com/photos/limelightleads/36422453096>



[https://commons.wikimedia.org/wiki/File:Corporate\\_Woman\\_Researching\\_GIF\\_Animation\\_Loop.gif](https://commons.wikimedia.org/wiki/File:Corporate_Woman_Researching_GIF_Animation_Loop.gif)



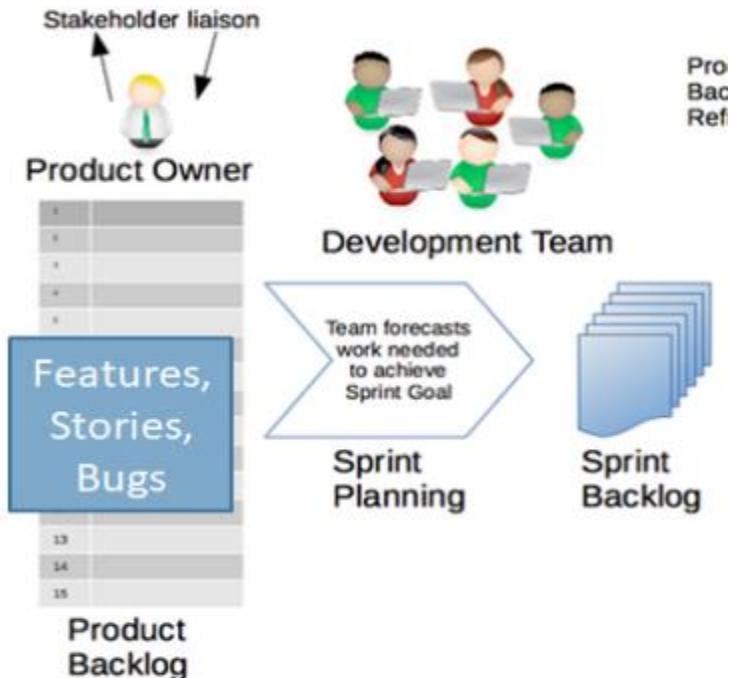
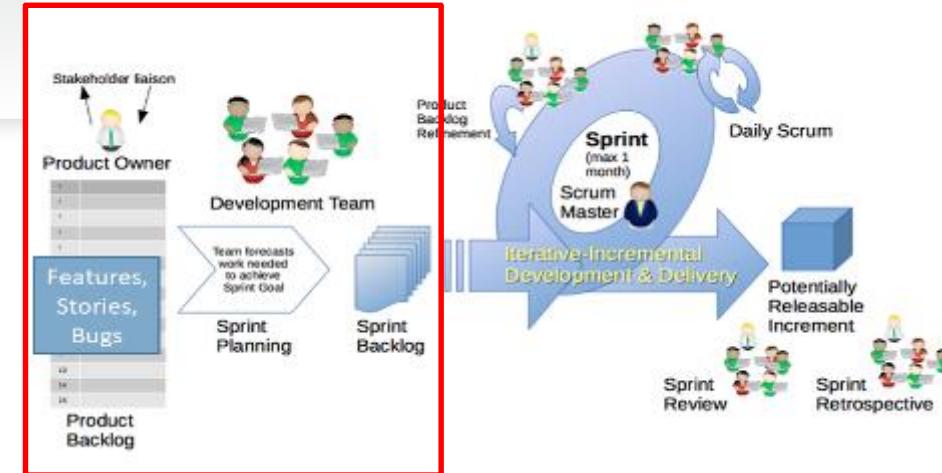
[https://en.wikipedia.org/wiki/Open-source\\_software](https://en.wikipedia.org/wiki/Open-source_software)



# Getting the Right Start on Time! **Sprint Planning**

# Sprint Planning

- Product Owner *presents the updated Product Backlog*
- Development Team *selects and refines User Stories*
- Development Team is able to *commit to the Sprint Backlog*



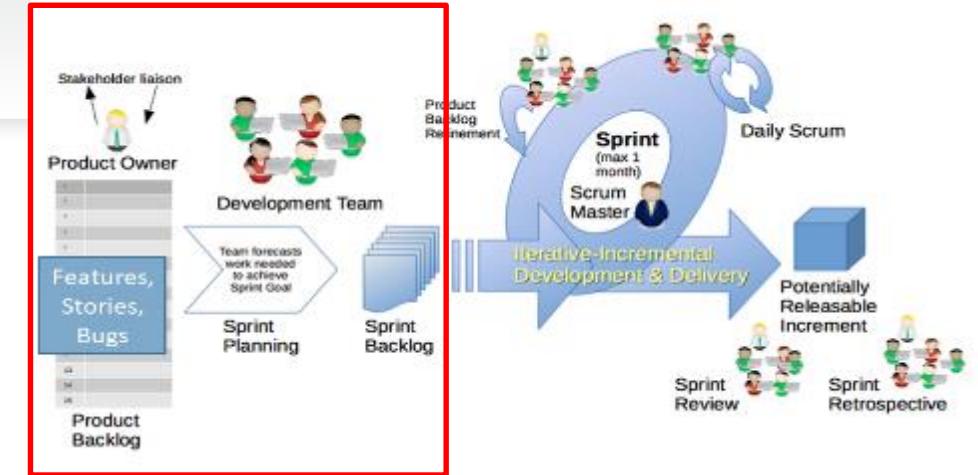
# Sprint Planning Guidelines

## Dos:

- All voices are heard
- Review and elaborate all User Stories in the Sprint
- Size and select those stories before end of Sprint Planning

## Don'ts:

- Discuss stories to death
- Argue or create conflicts, it should be objective
- Forget to prepare important story details (if possible)



# Sprint Planning Techniques

## Keys to Planning Success:

1. Great Users Stories
2. Use Planning Poker

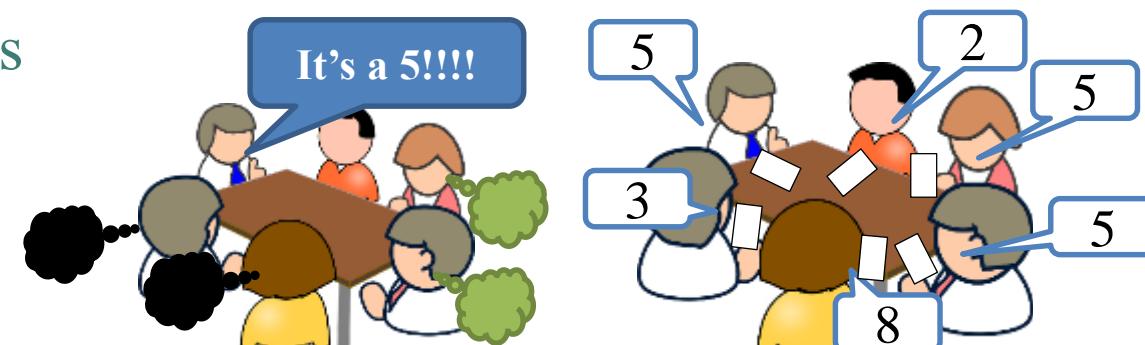
## No Open Discussions

- Loudest voice wins
- No timebox
- Can't prove consensus
- No way to improve



[https://commons.wikimedia.org/wiki/File:The\\_Layers\\_of\\_Login.png](https://commons.wikimedia.org/wiki/File:The_Layers_of_Login.png)

<https://pixabay.com/en/photos/clipboard/>





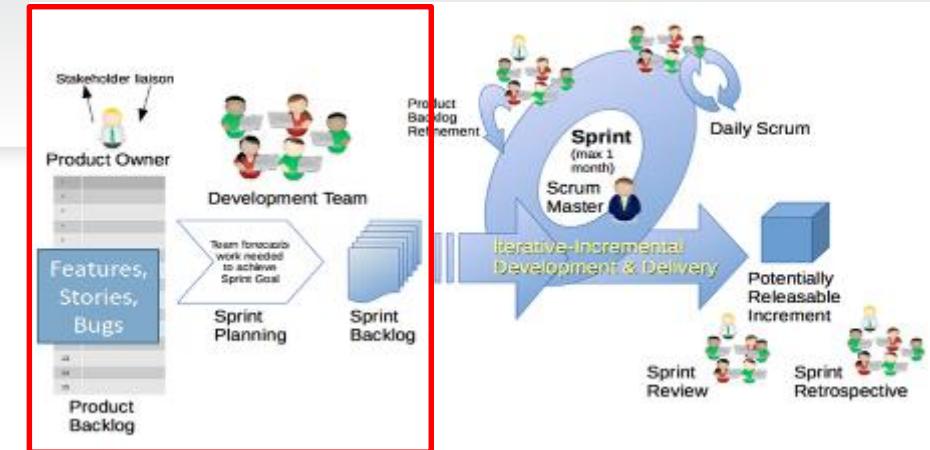
# Planning Poker

## Planning Poker Procedure:

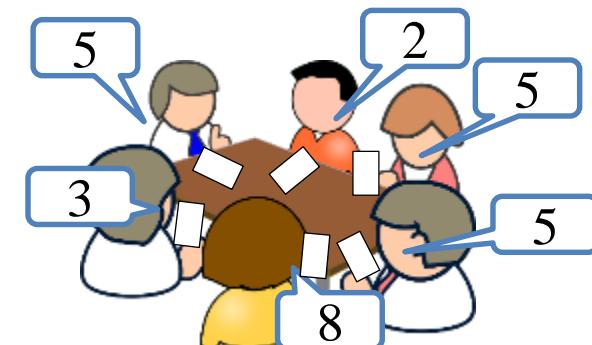
1. Agree on a point scale
2. Team briefly discusses the User Story
3. Everyone picks a card in silence
4. Team members reveal the card
5. If an outlier exists (more than one step from the mode), then discuss
6. (Optional) After two rounds take an average, or “yes/no” team vote

## Important notes about Point Scales

- Relative measure of size
- More than effort (size, risk, complexity)
- Remains same size over time, but teams may speed up!



<https://commons.wikimedia.org/wiki/File:CrispPlanningPokerDeck.jpg>



# Sprint Planning to Close

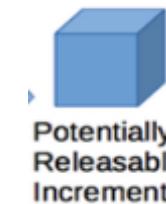
How to develop a great plan on-time:

- Work Iteratively through stories
- Plan for an Increment
- Work one-by-one
- Get a vote of confidence
- Write new stories on the spot
- Simple tooling
- Prioritize the stories together



*Get a Flow Going!*

*Plan to Release!*



[https://www.flickr.com/photos/kanban\\_tool/15817131058](https://www.flickr.com/photos/kanban_tool/15817131058)

*Confidence Vote!*



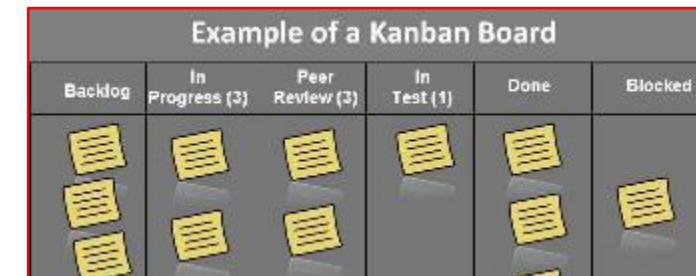
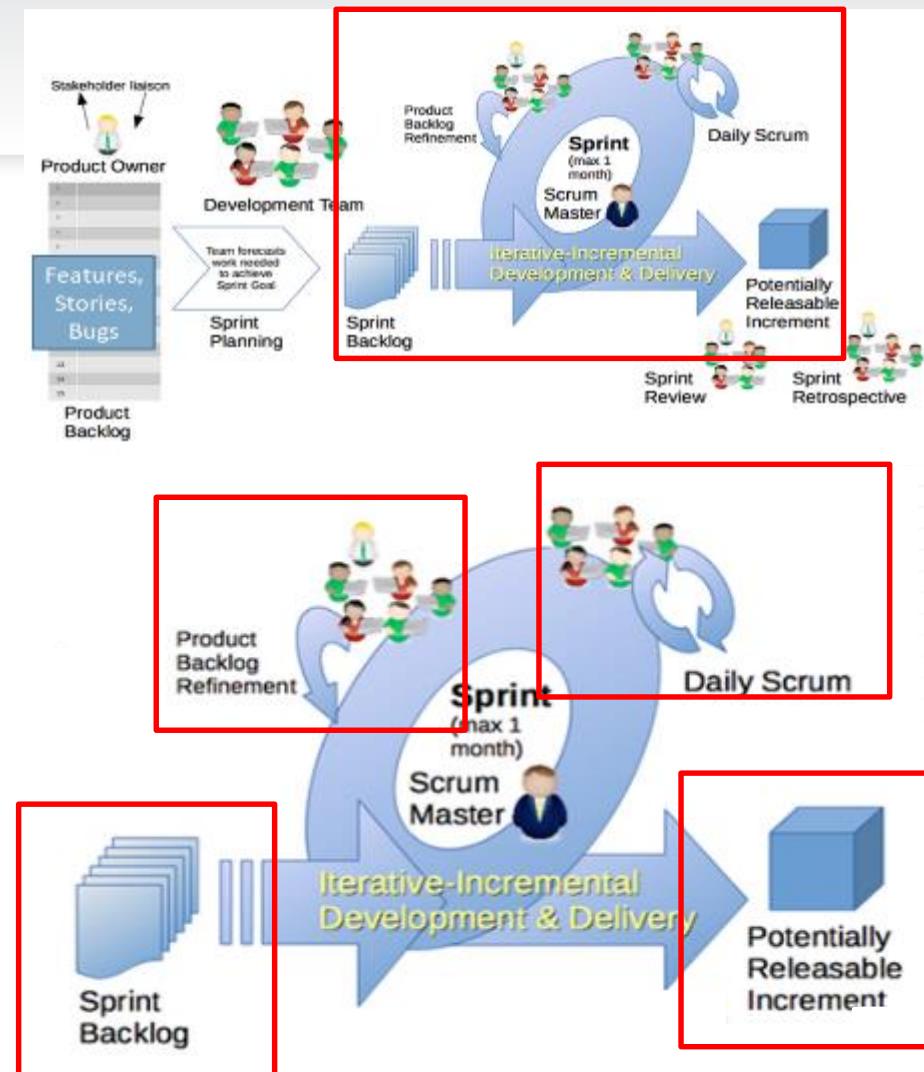


# Joyful Flow that Gets Work Done

## Sprint Development

# Sprint Development

- **Daily Stand Ups** - daily face to face communication
- **Whole Teams** - both executes and plans work together
- **Team Ownership** - Multiple team members work on a User Story
- **Limit WIP** - limit the Work-In-Progress (WIP) to achieve flow



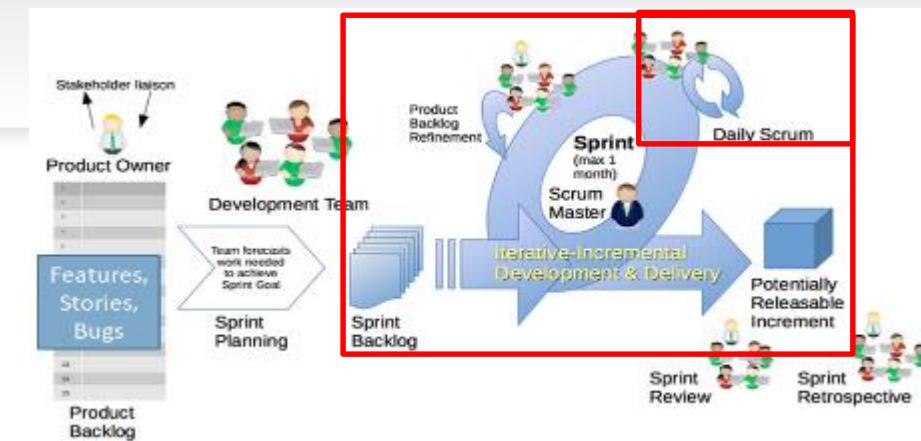


# Sprint Development

## Daily Standup

Two types of daily standups:

1. “Scrum Standup” – each person self-reports
2. “Forward-Looking Standup” - facilitator guides team to volunteer to help



<https://pixabay.com/en/photos/feedback/>



[https://commons.wikimedia.org/wiki/  
File:Pictofigo-Scrum.png](https://commons.wikimedia.org/wiki/File:Pictofigo-Scrum.png)

# Sprint Development

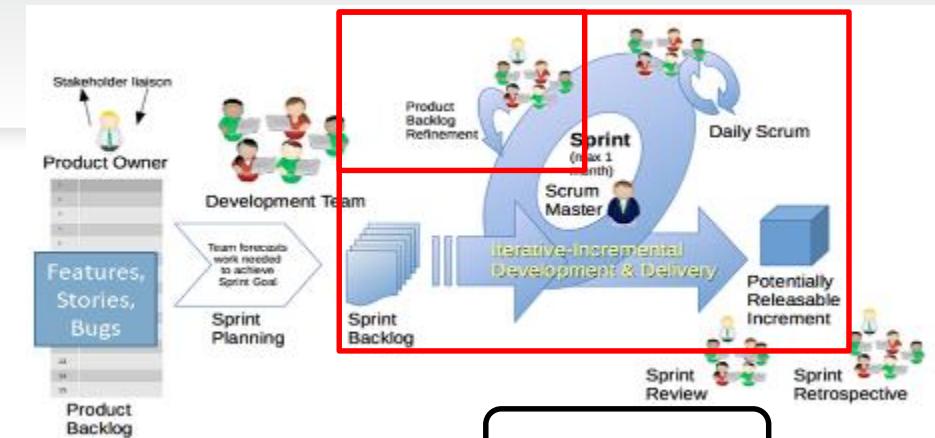
## Whole Teams

### User Story Execution

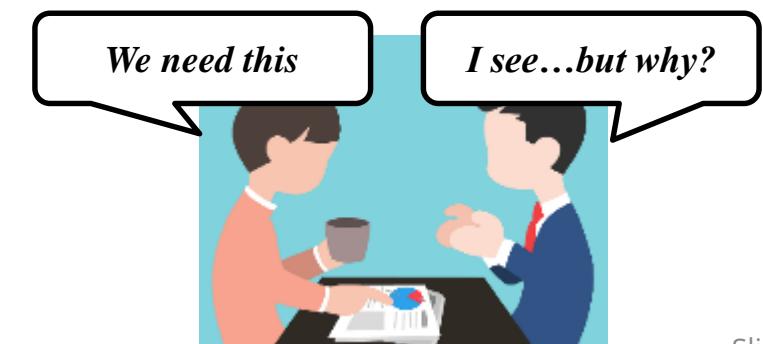
- **Product Owner** is available to provide input and is available to close
- **Scrum Master** facilitates meetings, workshops, and ensures quality through good User Stories, Story Closing, etc.
- **Development Team** completes work

### Continuous Planning

- **Product Owner** consults with stakeholders as product takes shape
- **Scrum Master** facilitates meetings, workshops, and trains teams as needed
- **Development Team** helps writing new stories or in meetings as needed



<https://media.defense.gov/2016/Aug/11/2001602138/-1/-1/160721-F-RI777-001.JPG>



<https://pixabay.com/en/meeting-conference-sales-business-1184892/>

# Sprint Development

## *Team Ownership*

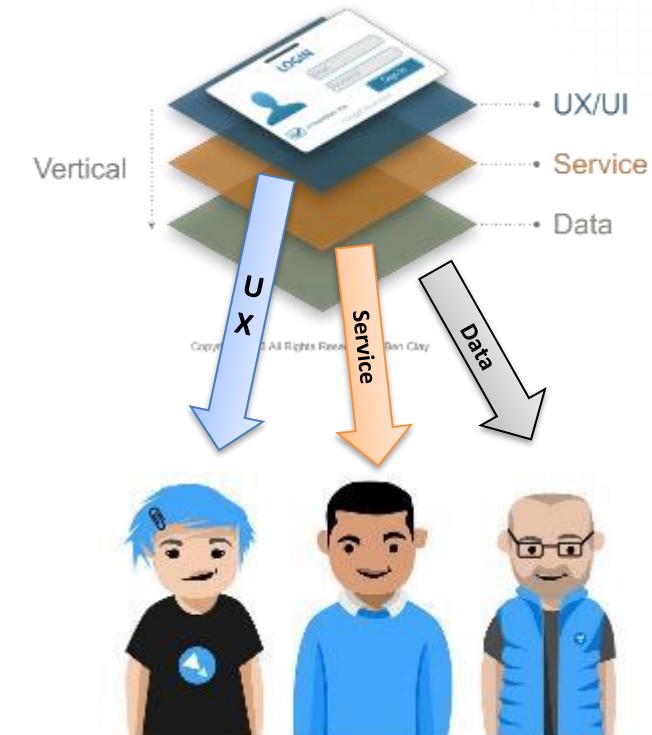
The Team works together to complete the Sprint Backlog:

- Team members work together to close a single User Story
- Work or Tasks are Split Up based on availability and skill
- If needed, the team will “swarm” or go into a “war room” to complete one story
- No “pride of authorship,” stories are clear in terms of tasks needed to get done

Team Ownership ensures highest priority work is accomplished first!



[https://commons.wikimedia.org/wiki/  
File:The\\_Layers\\_of\\_Login.png](https://commons.wikimedia.org/wiki/File:The_Layers_of_Login.png)



<https://www.goodfreephotos.com/vector-images/three-developers-character-set-vector-clipart.png.php>

# Sprint Development

## *Limit WIP*

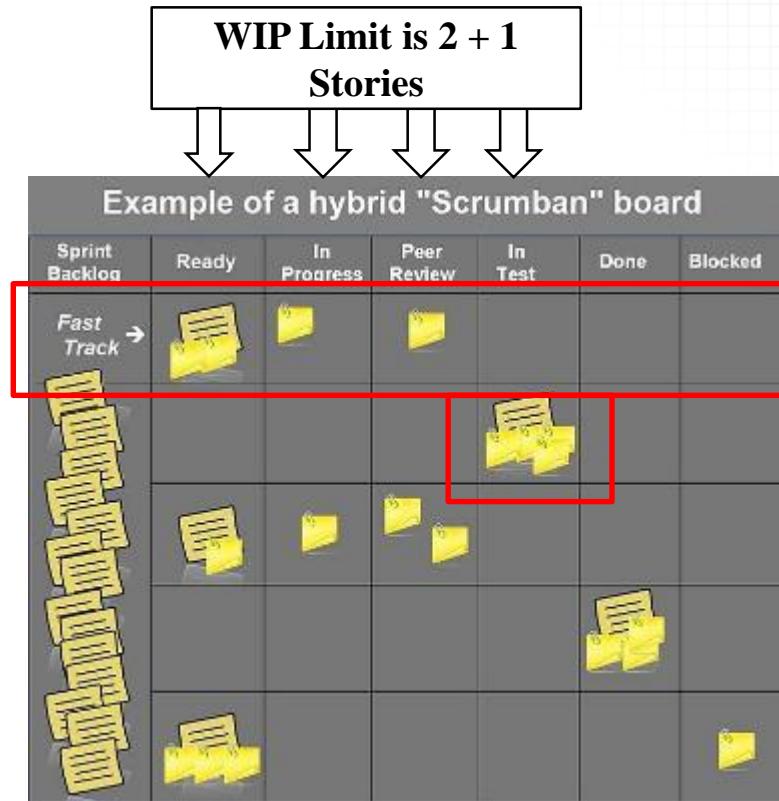
### Limiting Work-In-Progress (WIP)

- Ensures the Whole Teams mentality
- Reduces coordination meetings
- More quickly delivers working product

### Kanban and Scrumban board facilitate

- Kanban manages stories
- Scrumban manages tasks
- Use “Fast Lanes” when in continuous development-operations model

Limiting WIP is a Lean idea, but in Agile the team (not process) controls the WIP



[https://commons.wikimedia.org/wiki/File:Scrumban\\_task\\_board\\_example.jpg](https://commons.wikimedia.org/wiki/File:Scrumban_task_board_example.jpg)

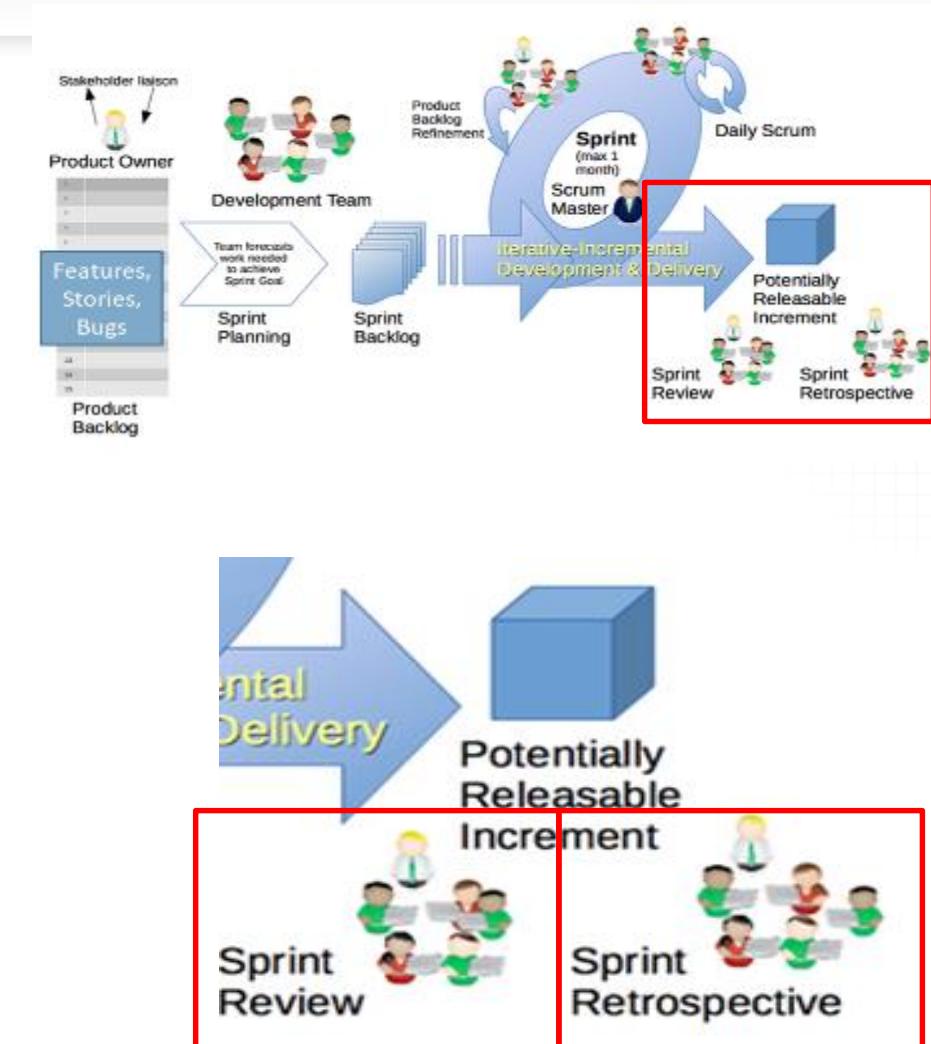


Learn Fast, Change Fast, Win Fast  
**Sprint Retro & Review**

# Sprint Retro & Review

## Closing a Sprint

- **Sprint Review:** the Product Owner presents the completed, potentially shippable increment to the stakeholders.
- **Sprint Retro:** the Sprint Team collaboratively inspects the sprint and looks for ways to build on or change for the better.





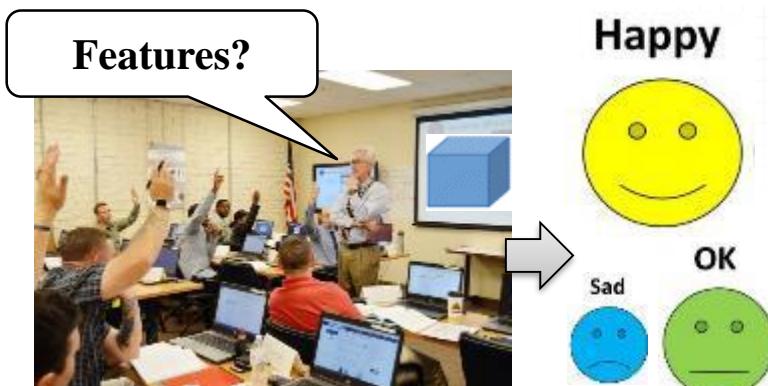
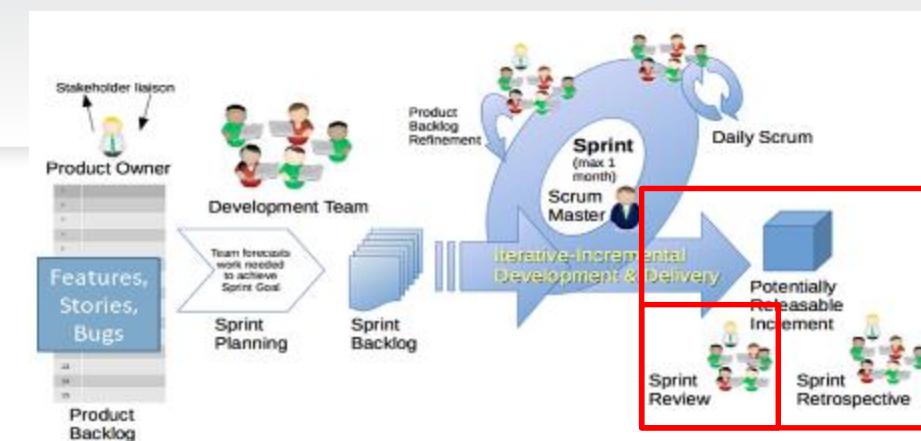
# Sprint Review

## Goals of Sprint Review

- Validate product is something users want
- Discuss what the next features should be
- Build stakeholder buy-in
- Force a shippable product to be ready\*

## How to Run a Great Sprint Review

- Focus on demoing the product
- Keep the presentations simple and small
- Avoid PowerPoint
- Show planned work in the planning tool
- Prepare for the event, but not too much
- Get the stakeholders involved!



[https://www.army.mil/article/200578/sj\\_l\\_tap\\_director\\_meets\\_with\\_stakeholders\\_in\\_hawaii](https://www.army.mil/article/200578/sj_l_tap_director_meets_with_stakeholders_in_hawaii)

<https://www.psypost.org/2017/02/emoticons-help-gauge-school-happiness-level-young-children-47328>



<https://www.seymourjohnson.af.mil/News/Article-Display/Article/307585/strike-eagle-demo-team-showcases-capabilities-says-thanks/>



# Sprint Retro

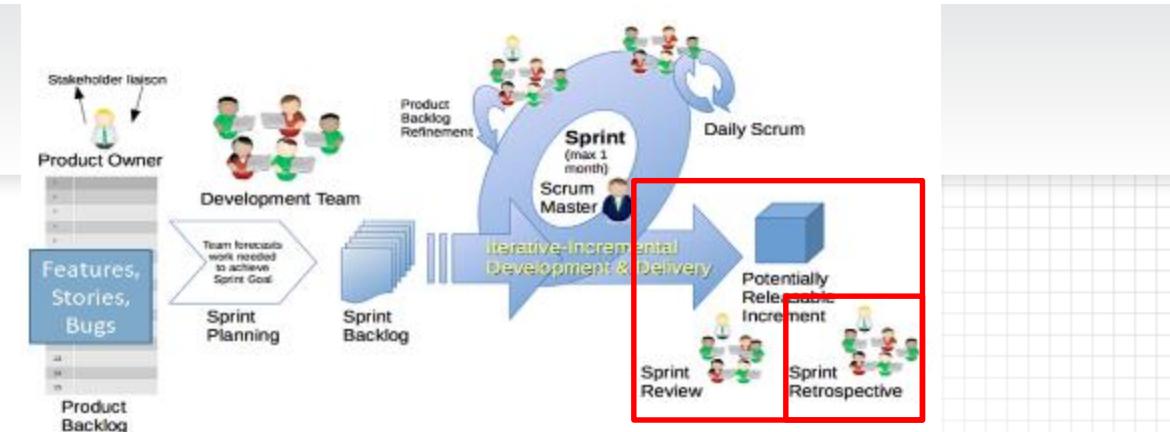
## Goals of Sprint Retro

- Capture what went right during the sprint
- Capture what went wrong during the sprint
- Capture what the team can do to improve

## How to Run a Great Sprint Retro

- Identify and plan work to improve team
- Stop doing the bad stuff early
- Use “lessons learned” on YOUR project (imagine it!)
- Dedicate time to hear from everyone, and gain team buy-in

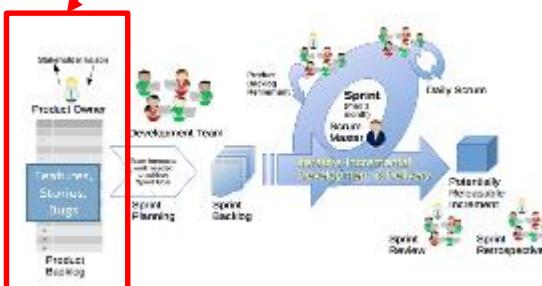
Now let's play the Retro Game!



Well	Not Well	Can Do
■ ■ ■ ■ ■	■ ■ ■ ■ ■	■ ■ ■ ■ ■



<https://pixabay.com/en/meeting-conference-people-table-152506/>



# Sprint Retro Game

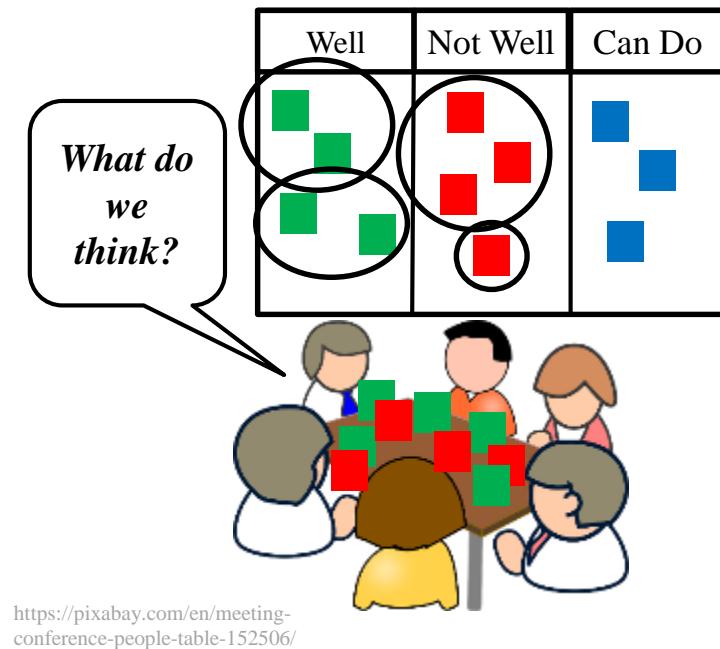
Two rounds to the Retro Game

- Evaluate what went right and wrong
- Evaluate what the team can do

In Both Rounds you do the Following

- Each person writes down their thoughts
- Everyone puts their thoughts on the board
- The whole team together labels the notes
- Scrum master facilitates discussion
- At the end the Can Do items are captured

Scrum Master or Team Member records!



<https://pixabay.com/en/meeting-conference-people-table-152506/>



# Sprint Celebration

- Go be social with your team!
- Remember:

*Individuals and Interactions  
OVER  
Processes and Tools*

- You just had a big win, now Celebrate!



<http://www.adventurejay.com/blog/index.php?m=03&y=15&d=14&entry=entry150314-151622&category=4>



You've just completed **Module 3 of**  
*Applied Scrum for Agile Project  
Management*

Thank You!

[CLICK here to go to the EdX Course Page](#)





# Week 4. Agenda

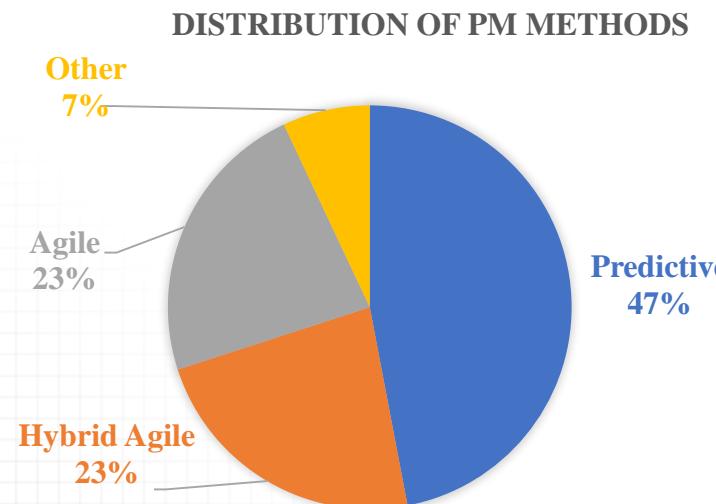
- 1. Scrum in the World of Agile**
- 2. Exploring the Scaled Agile Framework (SAFe)**
- 3. Exploring Disciplined Agile Delivery (DAD)**
- 4. Exploring Large-Scale Scrum (LeSS)**
- 5. Pitfalls and Benefits of Agile at Scale**



# The World of Agile is Here! **Scrum in the World of Agile**

# Pulse of the Profession by PMI: *Agile is Taking Over*

- Only 47% of projects use Predictive / Traditional approaches
- About half of all projects are now Agile



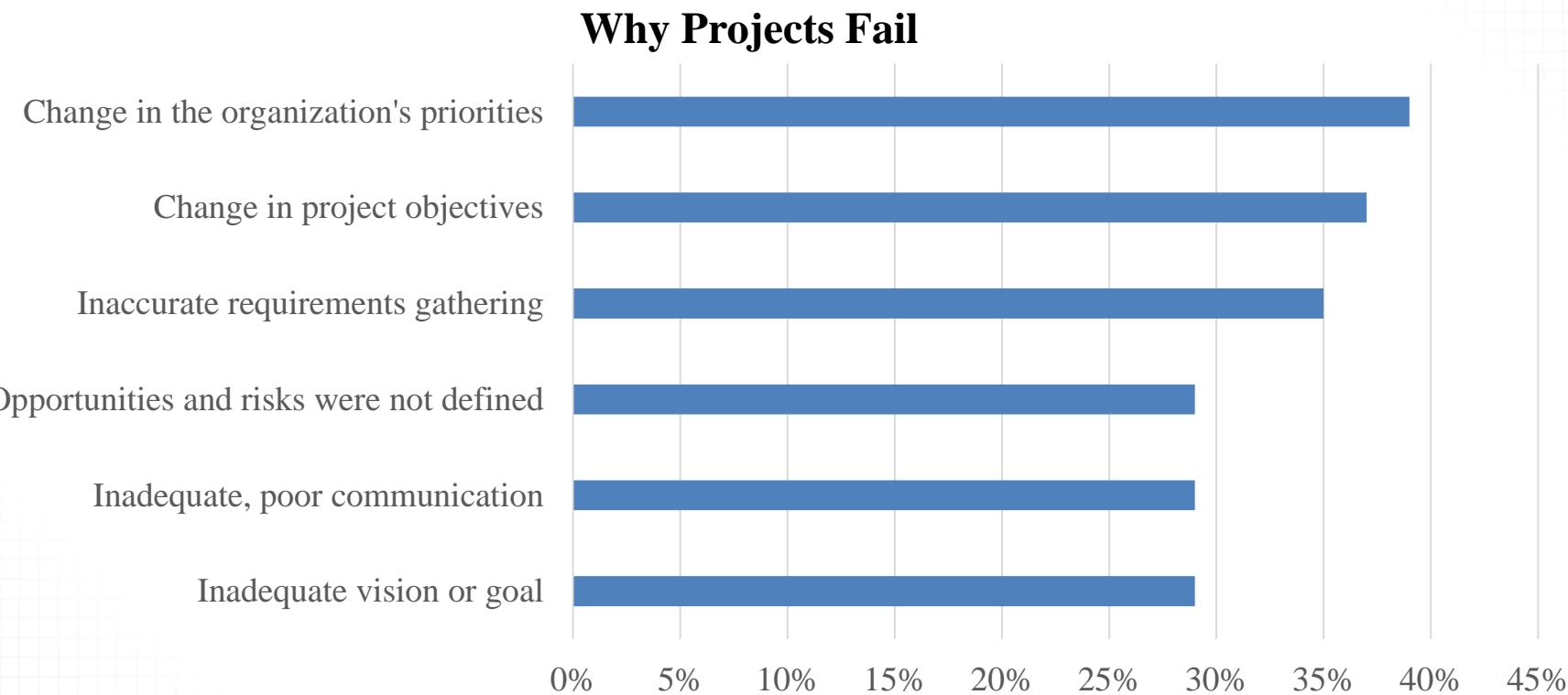
## Expanding Emphasis on Organizational Agility

71% of organizations report greater agility over the last five years. More and more organizations are recognizing that agility—the capability to quickly sense and adapt to external and internal changes to deliver relevant results in a productive and cost-effective manner—is helping them stay competitive.

71%

<https://www.youtube.com/watch?v=ICeVs6dKf74>

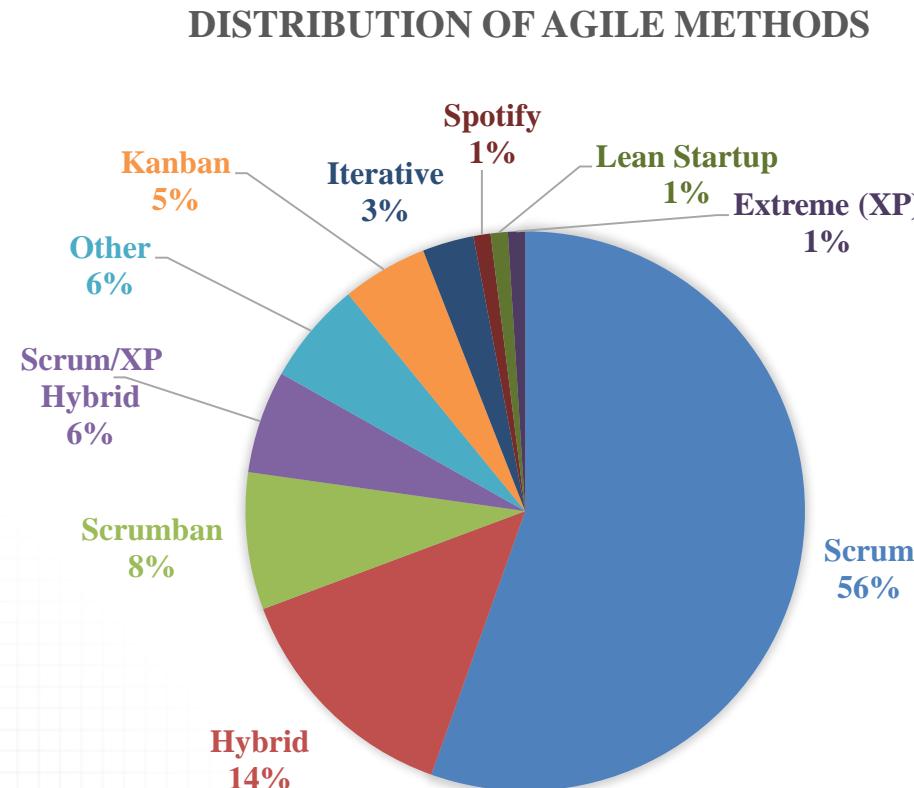
# Pulse of the Profession by PMI: *Why Projects Fail*



*Doesn't this look like a problem Agile could solve?*

# 12<sup>th</sup> Annual State of Agile by VerizonOne

## *Scrum Dominates Teams*

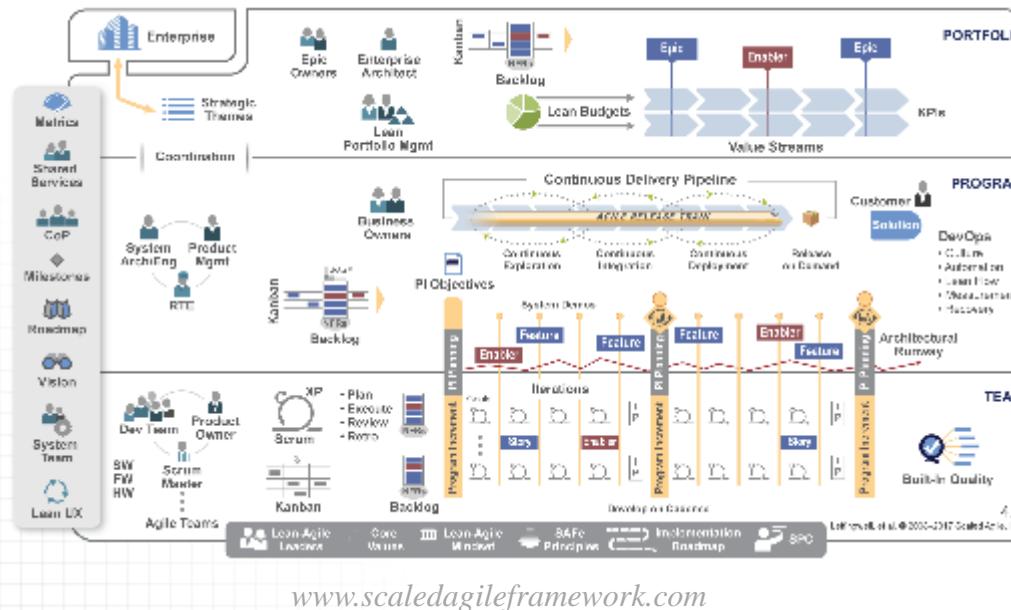


*Scrum, Scrumban,  
and Scrum/XP Hybrid  
account for 70%*

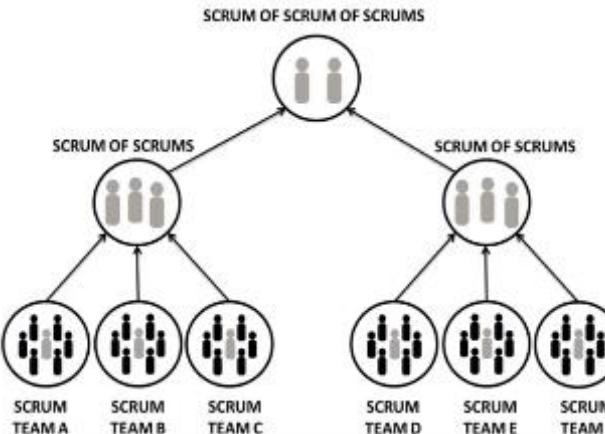
# 12<sup>th</sup> Annual State of Agile by VerizonOne

## *SAFe Dominates Scale*

*Scaled Agile Framework 29%*



*Scrum of Scrums 19%*



<https://www.youtube.com/watch?v=xH-o9iTcWNQ>

Agile Framework	Share
Internally Created (Hybrid)	10%
Disciplined Agile Delivery (DAD)	5%
Large-Scale Scrum (LeSS)	5%

# Scaling Frame Breakdown

Each Scaling Framework has:

- Scrum
- Scrum Roles

Agile Scaling Frameworks Differs on:

- How to manage “Support Teams”
- How to make an organization Agile.

Simplest Scaling Frameworks:

- Scrum of Scrums
- Hybrid Methodology



Scrum Master



<https://pxhere.com/en/photo/1444293>

Owner



<https://pixabay.com/en/avatar-icon-document-entrepreneur-1789663/>

Development Team Member



<https://pixabay.com/en/cartoon-character-idea-business-2948029/>

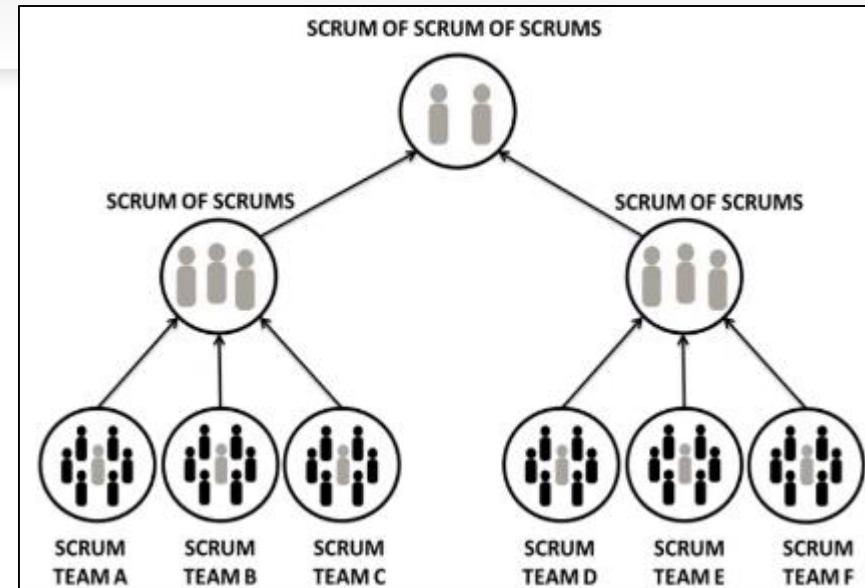
# Scrum of Scrums

## How does it work?

- Teams send representatives or leaders to meet and “Scrum”
- Like a Daily Stand Up, can be longer and less frequent
- Coordination among teams

## Why do it?

- Address team dependencies
- Address shared resources
- Coordinate product launches
- Share insights on shared goals



<https://www.youtube.com/watch?v=xH-o9iTcWNQ>

Originally this was proposed by Jeff Sutherland, "Agile Can Scale: Inventing and reinventing SCRUM in Five Companies" -  
Jeff Sutherland, Cutter IT Journal, 2001



**Saab Gripen Fighter**

[https://en.wikipedia.org/wiki/Saab\\_JAS\\_39\\_Gripen](https://en.wikipedia.org/wiki/Saab_JAS_39_Gripen)

*Teams on the Gripen report out over 500 people in an organization in less than 90 minutes!*

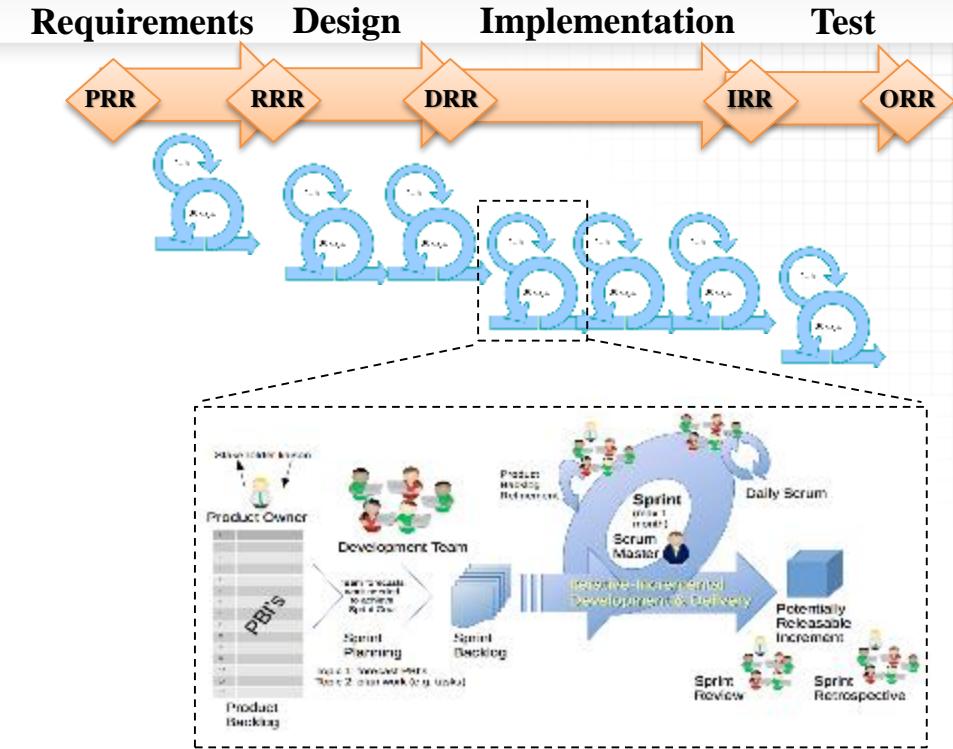
# Hybrid Model

How does it work?

- Traditional controls
- Stage-gates are kept in place
- Rapidly iterate between gates
- Requirements are Use Stories

Why do it?

- Traditional Management
- Development gets “Agile”
- Can still be Iterative
- 3<sup>rd</sup> Party Verification



Reason Agile Fails	Share
Organizational culture at odds with agile values	53%
General organizational resistance to change	46%
Inadequate management support	42%
Lack of skills/experience with agile methods	41%



# 12<sup>th</sup> Annual State of Agile by VerizonOne: Why Go Agile?

## Reasons for Adopting Agile

- Accelerate Software Delivery - 75%
- Manage changing priorities - 64%
- Increase productivity - 55%
- Better alignment of business and IT - 49%
- Increased software quality - 46%

## Benefits of adopting Agile

- Better management of priorities - 71%
- Project visibility - 66%
- Alignment of business and IT - 65%
- Delivery speed / time to market - 62%
- Team productivity - 61%

***Which  
Framework  
will you  
choose?***

***How will you  
Benefit?***

***How will you  
Manage?***



The Most Popular Scaling Method!  
**Exploring the Scaled Agile  
Framework (SAFe)**

# The Scaled Agile Framework (SAFe): Expands on the Roles

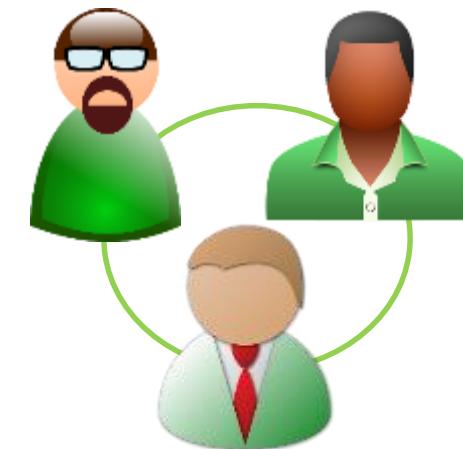
- **System Teams** - those that manage delivery and integration of products produced by individual Scrum teams
- **Architecture Teams** - manages and promotes the shared architecture framework across teams
- **Product Manager** - leads the Product Owners as the primary person in charge of targeting features and EPICs
- **Release Train Engineer** - leads the Scrum Masters on each of the Scrum teams, and conducts the large team or team ceremonies



<https://pixabay.com/en/photos/%E0%B8%A0%E0%B8%B2%E0%B8%9E%E0%B8%95%E0%B8%B1%E0%B8%94%E0%B8%9B%E0%B8%80%E0%B8%A0%E0%B8%9E0%B8%9E/?cat=industry>

<https://pixabay.com/en/avatar-messenger-person-user-beard-153139/>

[http://icongal.com/gallery/icon/59447/256/male\\_man\\_boy\\_guy\\_user\\_person\\_customizer\\_face](http://icongal.com/gallery/icon/59447/256/male_man_boy_guy_user_person_customizer_face)



<https://pixabay.com/en/photos/user%20avatar/>

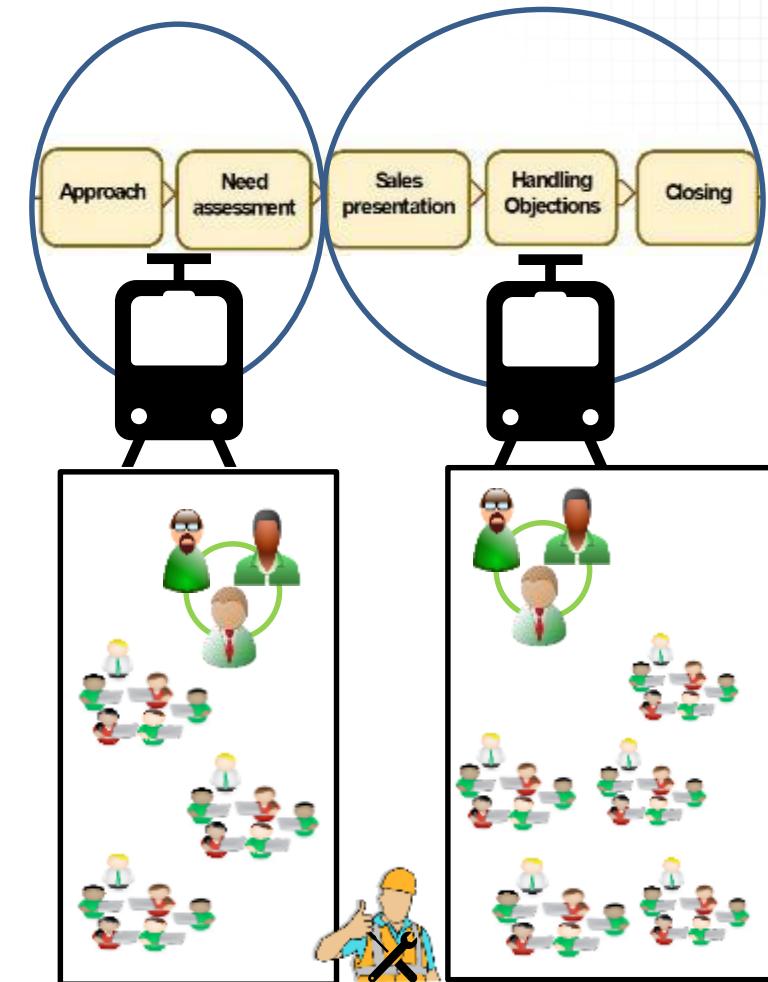
# Value Streams and Agile Release Trains (ARTs)

## Value Streams:

- Model the business using Lean principles
- Manages work in small batches & PDCA

## Agile Release Train (ARTs)

- Align to one or more similar parts of the Business Value Stream
- Limited to up to 120 people (Dunbar's number)
- Shared ceremonies at the Release boundaries (Planning, I&A)



# Big Room Planning

All Agile Teams get in a room

- Release Train Engineer organizes and coordinates Planning Day
- Product Manager provides a shared vision and features

Teams then conduct faster Scrum-like Planning

- Points are considered absolute ***one per person for one day***
- Teams commit to complete PI Objectives
- PI Objectives given business value points by Business Owners

Teams identify their dependencies

- ***Program Board work and dependencies across teams***
- “ROAM” risks: Resolve, Owned, Accepted, or Mitigated

Everyone gives a “vote of confidence”

- Determines whether they can meet the objectives
- Keeps going until the whole team puts up “5 out of 5.”



<https://commons.wikimedia.org/wiki/File:AmsterdamHackathon-20130524-2629.JPG>



[https://vi.wikipedia.org/wiki/Scrum\\_\(m%C3%B4t%C3%A4nh\\_ph%C3%A1t\\_tr%E1%BB%83n\\_ph%E1%BA%A7n\\_m%E1%BB%81m](https://vi.wikipedia.org/wiki/Scrum_(m%C3%B4t%C3%A4nh_ph%C3%A1t_tr%E1%BB%83n_ph%E1%BA%A7n_m%E1%BB%81m)

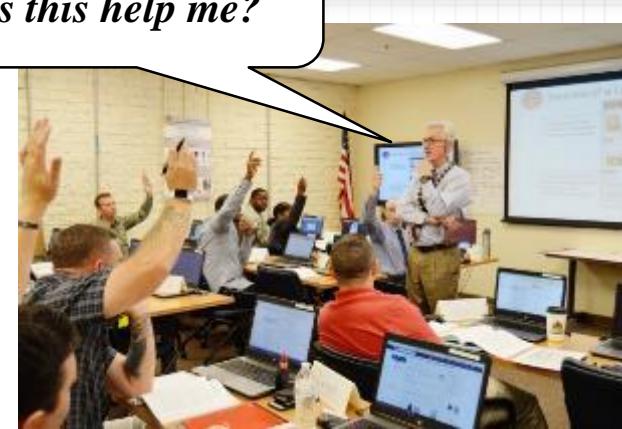


<https://pixabay.com/en/hands-raised-raised-hands-arms-up-1768845/>

# Program Inspect and Adapt (IA)

- System demo is performed across all teams
  - Often includes the Project Sponsors
  - Humanizes management
- Business Owners give feedback on achievement of business value points
- Retrospectives are run briefly to identify the most important problem to solve
- Problems are then addressed using workshops that include Business Owners, with clear outcomes and support by leadership

*As a customer, why does this help me?*

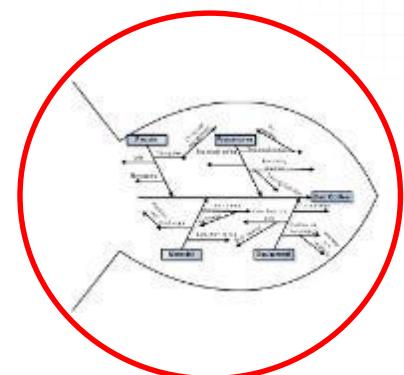


[https://www.army.mil/article/200578/sfl\\_tap\\_director\\_meets\\_with\\_stakeholders\\_in\\_hawaii](https://www.army.mil/article/200578/sfl_tap_director_meets_with_stakeholders_in_hawaii)

Well	Not Well	Can Do
■ ■ ■ ■ ■	■ ■ ■ ■ ■	■ ■ ■ ■ ■



<https://pixabay.com/en/meeting-conference-people-table-152506/>



[https://commons.wikimedia.org/wiki/File:Fishbone\\_BadCoffee\\_Example.jpg](https://commons.wikimedia.org/wiki/File:Fishbone_BadCoffee_Example.jpg)

# SAFe Principles and Configs

## SAFe Principles that are unique

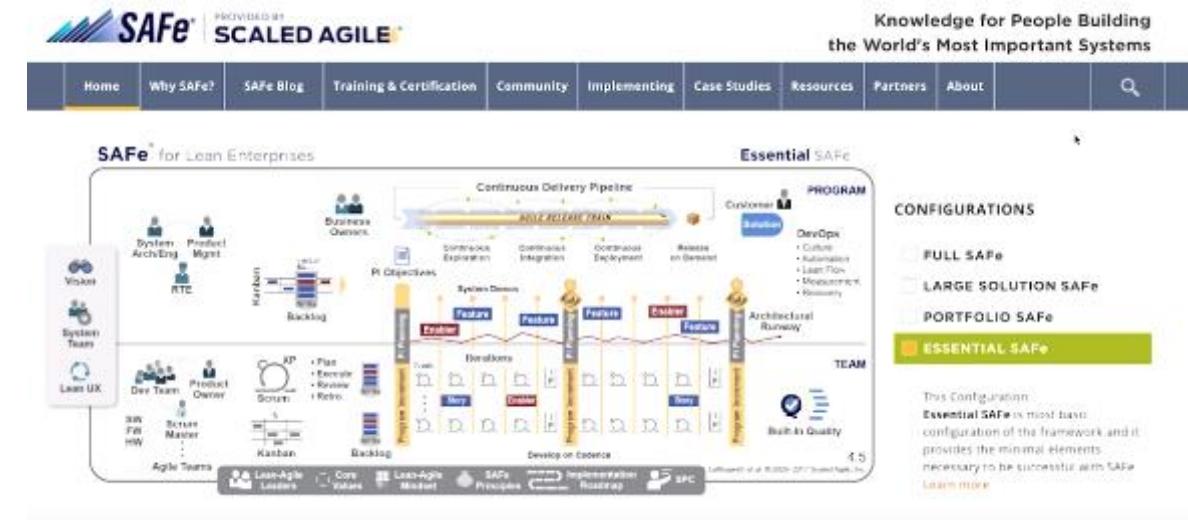
- Take an economic view
- Plan on cadence, release on demand
- Base milestones on working systems
- Visualize and limit WIP
  - Reduce batch sizes
  - Manage queue lengths

## SAFe Configurations

- Essential SAFe
- Portfolio SAFe
- Large-Solution SAFe
- Full SAFe

WSJF =   
**Cost of delay**  
**Duration**

Learn about Cost of Delay (CoD) and Weight Shortest Job First (WSJF):  
<https://www.youtube.com/watch?v=FuHcVpPgGNg>



Learn about SAFe 4.5: [https://www.youtube.com/watch?time\\_continue=1&v=qTG4I6jUbj4](https://www.youtube.com/watch?time_continue=1&v=qTG4I6jUbj4)

[www.scaledagileframework.com](http://www.scaledagileframework.com)



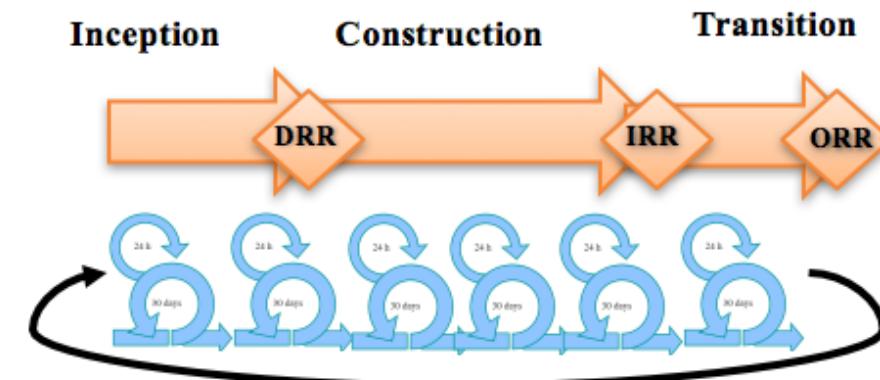
An Original Framework to Scaling Agile!  
**Exploring the Disciplined Agile  
Delivery (DAD)**

# Characteristics of Disciplined Agile

## Principles of Disciplined Agile

- People first
- Learning oriented
- Agile
- Hybrid
- Goal-driven
- Delivery focused
- Enterprise aware
- Risk and value driven
- Scalable

## Disciplined Agile Delivery (DAD)



# Disciplined Agile Delivery (DAD): *Primary Roles*

## Primary Roles:

- **Stakeholder** - these are the same as in Scrum - anyone who is impacted by the solution
- **Team Member** - focuses on producing solutions for stakeholders
- **Team Lead** - servant leader that coaches, often considered an “Agile Project Manager”
- **Product Owner** - provides the “voice of the customer”
- **Architecture Owner** - can be simple as the “senior developer” or an architect; with the goal of reducing technical debt risk at scale



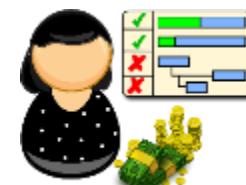
**Stakeholders**

[https://cdn.pixabay.com/photo/2016/04/18/05/computer-1331579\\_960\\_720.png](https://cdn.pixabay.com/photo/2016/04/18/05/computer-1331579_960_720.png)



**Team Member**

<https://pxhere.com/en/photo/1199795>



**Team Lead**

<https://pixabay.com/en/agenda-comic-characters-gantt-2025762/>



**Product Owner**

<https://pixabay.com/en/atar-icon-document-entrepreneur-1789663/>



**Architecture Owner**

<https://www.maxpixel.net/Business-Man-Architect-Work-Businessman-Architektin-1816217>

# Disciplined Agile Delivery (DAD): *Secondary Roles*

## Secondary Roles:

- **Specialist** - may be the specialist in a certain technology or tool that's used in the solution
- **Domain Expert** - provides detailed domain expertise on critical topics for complex solutions
- **Technical Expert** - can be experts in key non-functional areas (UX, Security, etc.)
- **Independent Tester** - can be required for complex solution or regulatory requirements (e.g. government)
- **Integrator** - can be a separate role for integration and delivery mechanisms in complex solutions (DevOps)

**Domain expert**  
**Technical expert**



<https://www.flickr.com/photos/rosenfeldmedia/11496789864>



**Specialist**

<https://pxhere.com/en/photo/868229>

**Independent Tester**



**Integrator**



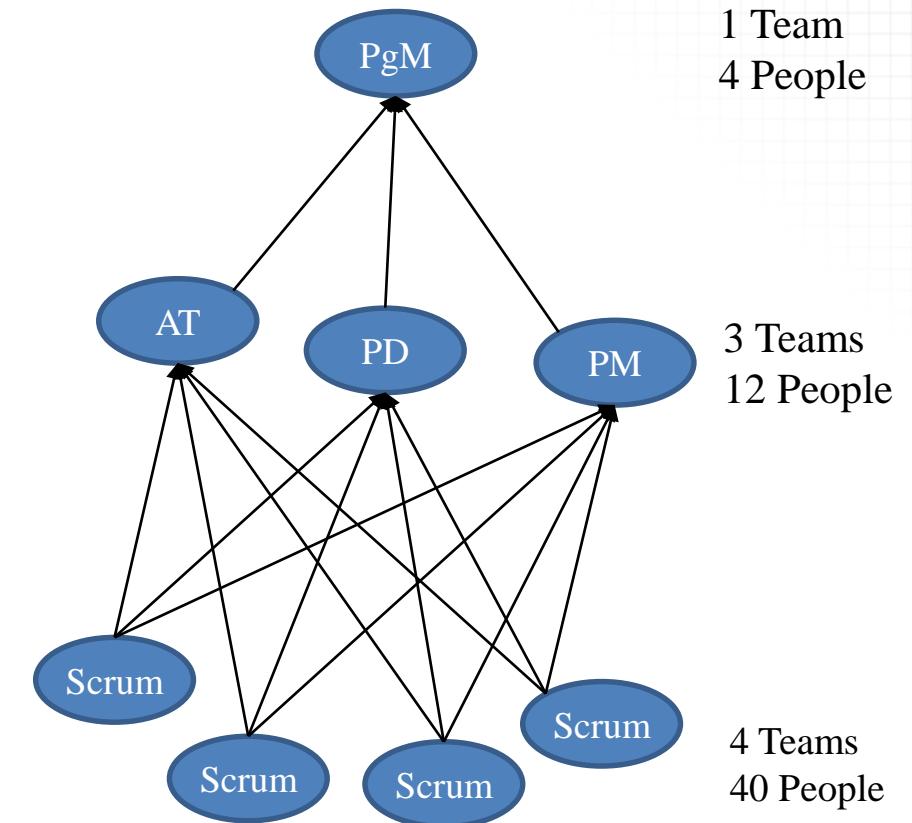
<https://pixabay.com/en/photos/%E0%B8%A0%E0%B8%B2%E0%B8%9E%E0%B8%95%E0%B8%B1%E0%B8%94%E0%B8%9B%E0%B8%B0%E0%B8%A0%E0%B8%B2%E0%B8%9E/?cat=industry>

# Disciplined Agile Delivery (DAD): *Hybrid Scaling*

## How DAD Scales Hybrid-Like:

- Agile teams meet in **Daily Standups**
- Team Leads meet separately to coordinate delivery, as the **Product Delivery Team**
- Architecture Leads meet to coordinate architecture and remove dependencies, as the **Architecture Team**
- Product Owners meet separately to coordinate planning, as the **Product Management Team**
- The **Program Manager** coordinates work across all three teams to reduce the accumulation of debt.

Total of 45 minutes to report on 40 people



# Disciplined Agile Enterprise

Scaling in Disciplined starts with DevOps:

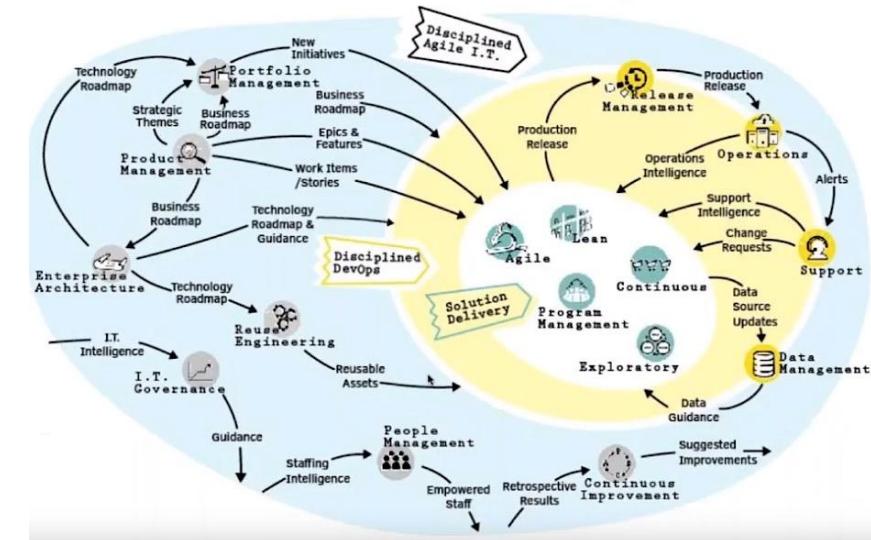
- IT Operations
- Customer Support
- Security
- Data Management
- Release Management

Scaling continues to Product Management, considered “IT”

- Enterprise Architecture
- Human resources
- Portfolio Management

Expands to Sales, Marketing, Finance, etc. as a Disciplined Agile Enterprise

Workflow of Disciplined Agile IT (DAIT)



<https://www.youtube.com/watch?v=OCDCKDScmHg&t=847s>

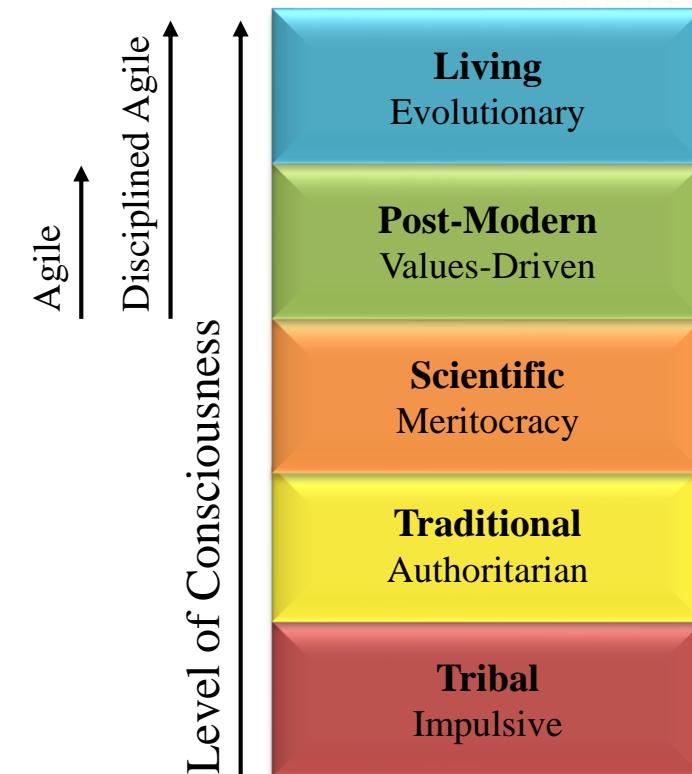
Workflow of a Disciplined Agile Enterprise (DAE)



# The Adaptive Learning Organization

Adaptive Learning Organization:

- **Tribal** - impulsive, and driven by urgency; management “preys” on its employees
- **Traditional** - Authoritarian, driven by protocols and formal roles and hierarchies
- **Scientific** - Profit or growth-oriented, driven by innovation and meritocracies of ideas
- **Post-Modern** - Consensus driven, with values-based decision making
- **Living** - Cellular models of management with an evolutionary purpose



*Based on diagram from:  
<http://www.disciplinedagiledelivery.com/dae/>*



# Large Solutions that De-Scale the Enterprise! Exploring Large-Scale Scrum (LeSS)

# The Basis of LeSS De-Scaling

Small Solutions as “Agile Parts” of an organization are a trap:

- We create “independent” teams with internal and external markets
- For larger products, “independent Teams” need a portfolio manager
- Portfolio Management leads to integration and dependencies
- Integration and dependencies drive the need for rules (our pills)
- These rules re-introduce the Program Management Office (PMO)
- Rules remove ownership and power from the Scrum teams, leads to complex teams

[http://www.publicdomainfiles.com/show\\_file.php?id=13540051816086](http://www.publicdomainfiles.com/show_file.php?id=13540051816086)



<https://pixabay.com/en/photos/torn/>

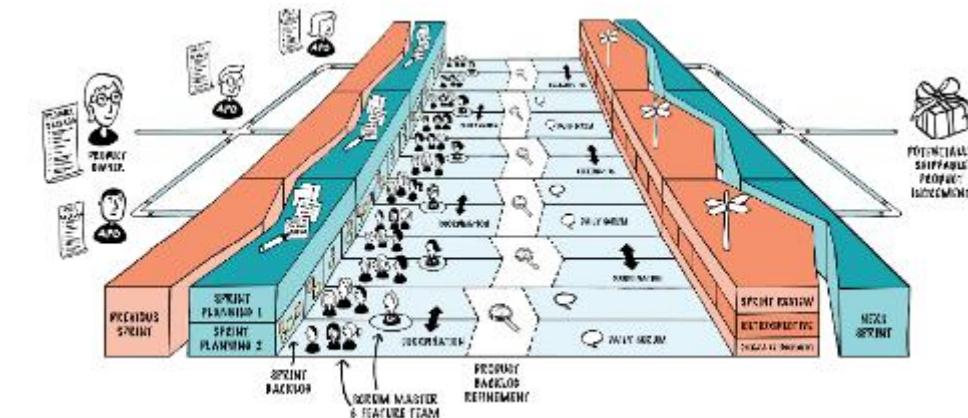
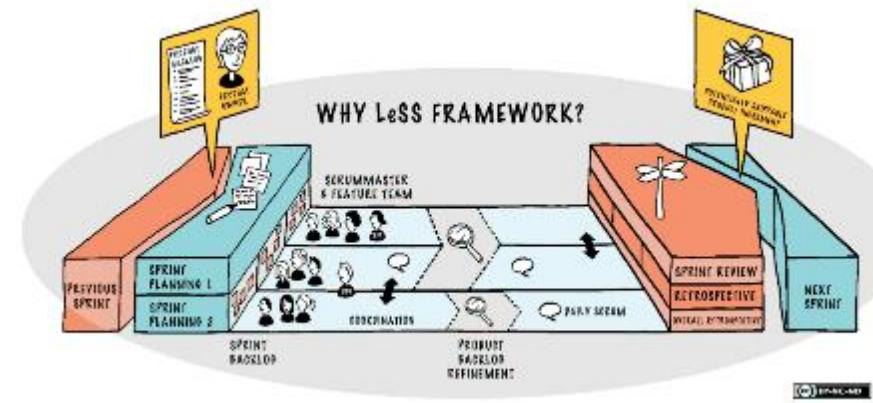
# Large-Scale Scrum (LeSS) Frameworks

Less has two frameworks:

- LeSS - used for up to eight (8) teams of eight
- LeSS Huge - used for up to thousands of teams delivering a product together

Still Scrum:

- Product Owner
- Scrum Master
- Development Team



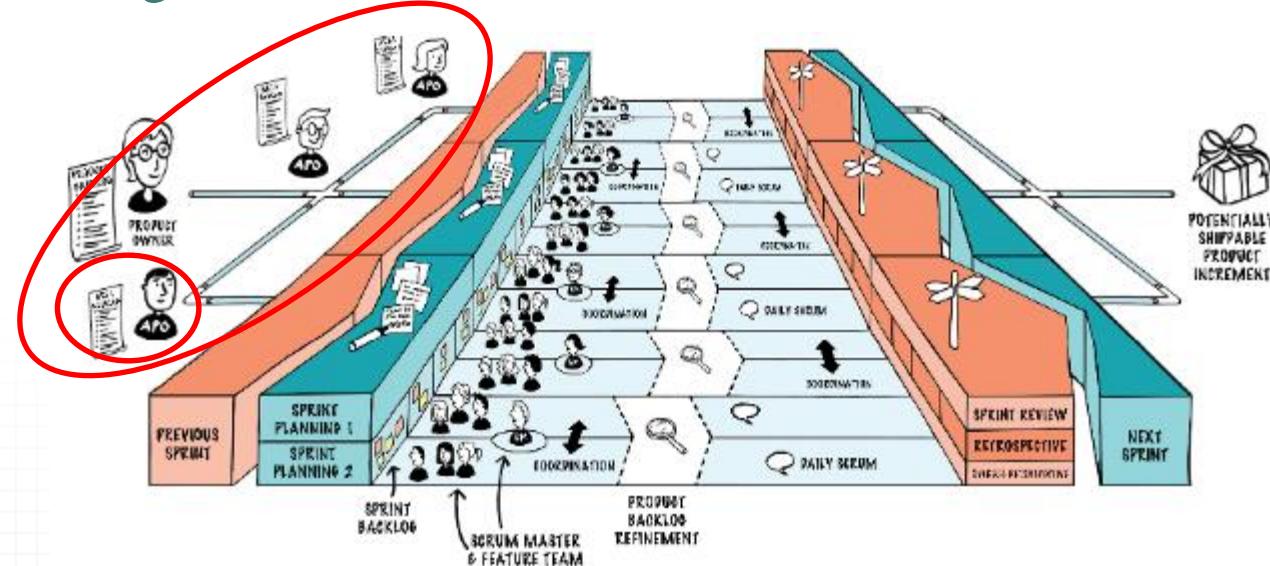
<https://less.works/resources/training-resources/index.html>

# LeSS Huge Adds a Single Role

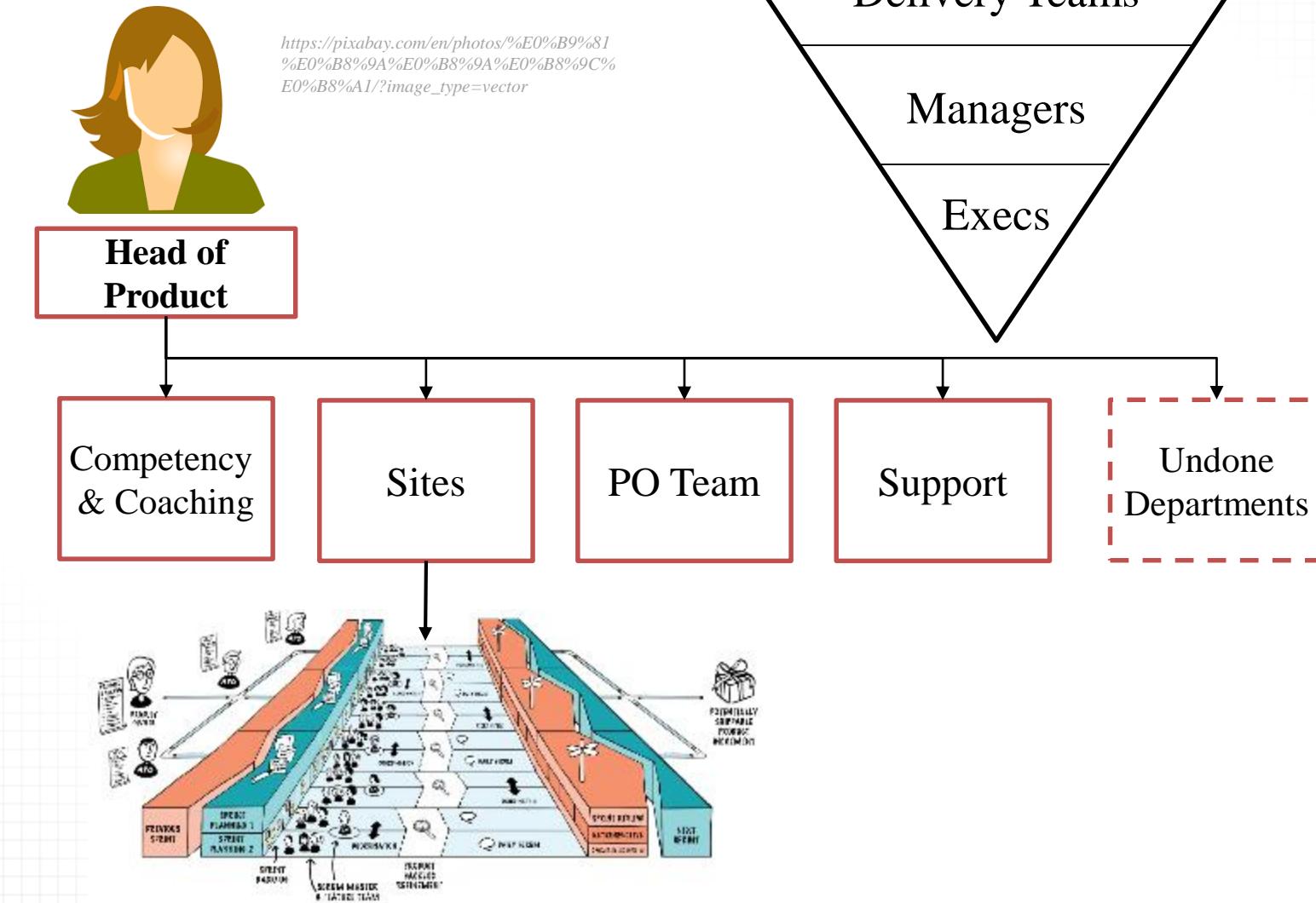
Area Product Owner (APO) - provides a buffer of work definition

- Can manage up to 8 teams
- Cannot override the prioritization backlog items (*this still belongs to the PO!*)

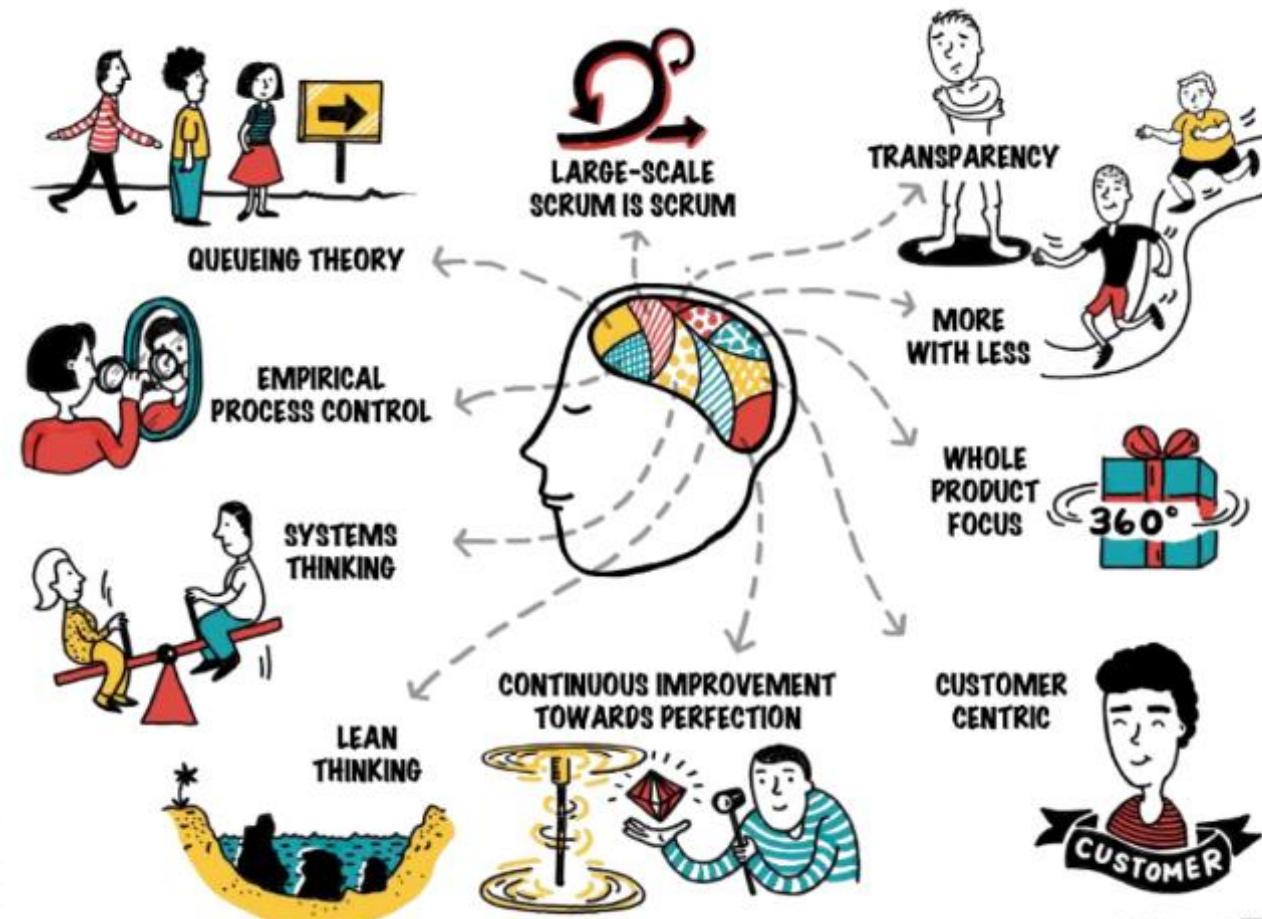
Product Owner (PO) can have up to ten (10) APOs working with them to manage requirements, forming the *Product Owner Team*



# LeSS: Company Model



# Goal is to Live by LeSS Principles



<http://less.works>

<https://less.works/resources/LeSS-complete-picture.pdf>



# Know before You Go! Pitfalls & Benefits of Agile



# Buyer Be Cautious and Know the Solutions

- **Lack of overall product design** – most common failure point
- **Adding stories to an iteration in progress** – very easy to do with poor plans or collaboration
- **Lack of sponsor support** – teams need access to end users, autonomy, and freedom
- **Insufficient training** – results in rework, or simply bad planning. Expect technical debt.
- **Product owner role is not properly filled** – organization is not “Agile.” Slows down decisions
- **Teams are not focused** – this happens when you have a wishy-washy product owner
- **Excessive preparation/planning** – often happens when there’s a lack of trust or experience.
- **Problem-solving in the daily standup** – what a drain! Team leads beware.
- **Assigning tasks** – Scrum Master isn’t facilitating. Need for a good story planning.
- **Scrum master as a contributor** – hard to avoid, but needed. Scrum Masters is a full-time job.
- **Lack of test automation** – needed on most projects. Testing automation is essential speed.
- **Allowing technical debt to build up** – address using Agile Engineering and “Hardening”
- **Attempting to take on too much in an iteration** – prioritization adds speed and innovation.
- **Fixed scope and quality** – Organizations must “embrace agile.”

# Benefits Realization in Agile: *Speed*



## Key Benefits:

- First Mover Advantages
- Higher ROI
- Lower Cost of Delays
- Faster Learning Cycles

## #1 Reason for Adopting Agile

### Techniques

- Scope Management
- Advanced Kanban board practices to limit work-in-progress (WIP)
- Time boxing activities to eliminate delays and gain schedule advances
- Pareto Principle
- Rolling Wave Planning

# Benefits Realization in Agile: *Innovation*



## Key Benefits:

- Target Solutioning
- Learning Cycles
- Constraints for Innovation
- Problem Solving Methods

## #1 Benefit Realized by Adopting Agile

### Techniques

- Business Value Mapping
- Agile Requirements Gathering
- Constraint-driven solutions
- Test-Driven Development (TDD)
- Competing Objectives Solving

# Benefits Realization in Agile: *Leadership*



## Key Benefits:

- Motivation
- Greater Team Creativity
- Better Decisions
- Reduced Conflicts
- More time

## #1 Reason that Teams Love Agile

### Techniques

- Building self-organizing teams
- Facilitating leadership (power of play)
- Decision science
- Negotiation styles and techniques
- Managing bias through mindfulness and emotional intelligence (EQ)

# Benefits Realization in Agile: *Control*



## #1 Reason that Agile Works

### Key Benefits:

- Transparency
- Stakeholder Agreement
- Meaningful Milestones
- Authentic Insights
- Continuous Improvement

### Techniques

- Agile Systems Engineering
- Controlling through measurement
- 3Ps - People, Process, Product
- Scaled Decision Science that Works
- Enterprise alignment of Agile Teams



# Questions?

# Thank You!





PROJECT MANAGEMENT  
CENTER FOR EXCELLENCE

A.J. CLARK SCHOOL OF ENGINEERING  
Civil & Environmental Engineering Department

Congratulations....you have now completed  
*Applied Scrum Agile for Project Management!*



Continue!



You are now ready to complete the next course....[Sprint Planning for Faster Agile Team Delivery!](#)

*Complete all five to earn your Mastering Agile Project Management Professional Certificate.*

[Register Today!](#)

WE HAVE THE  
**MASTERS LEVEL EDUCATION OPTIONS**  
YOU NEED.

Technical undergraduate degree

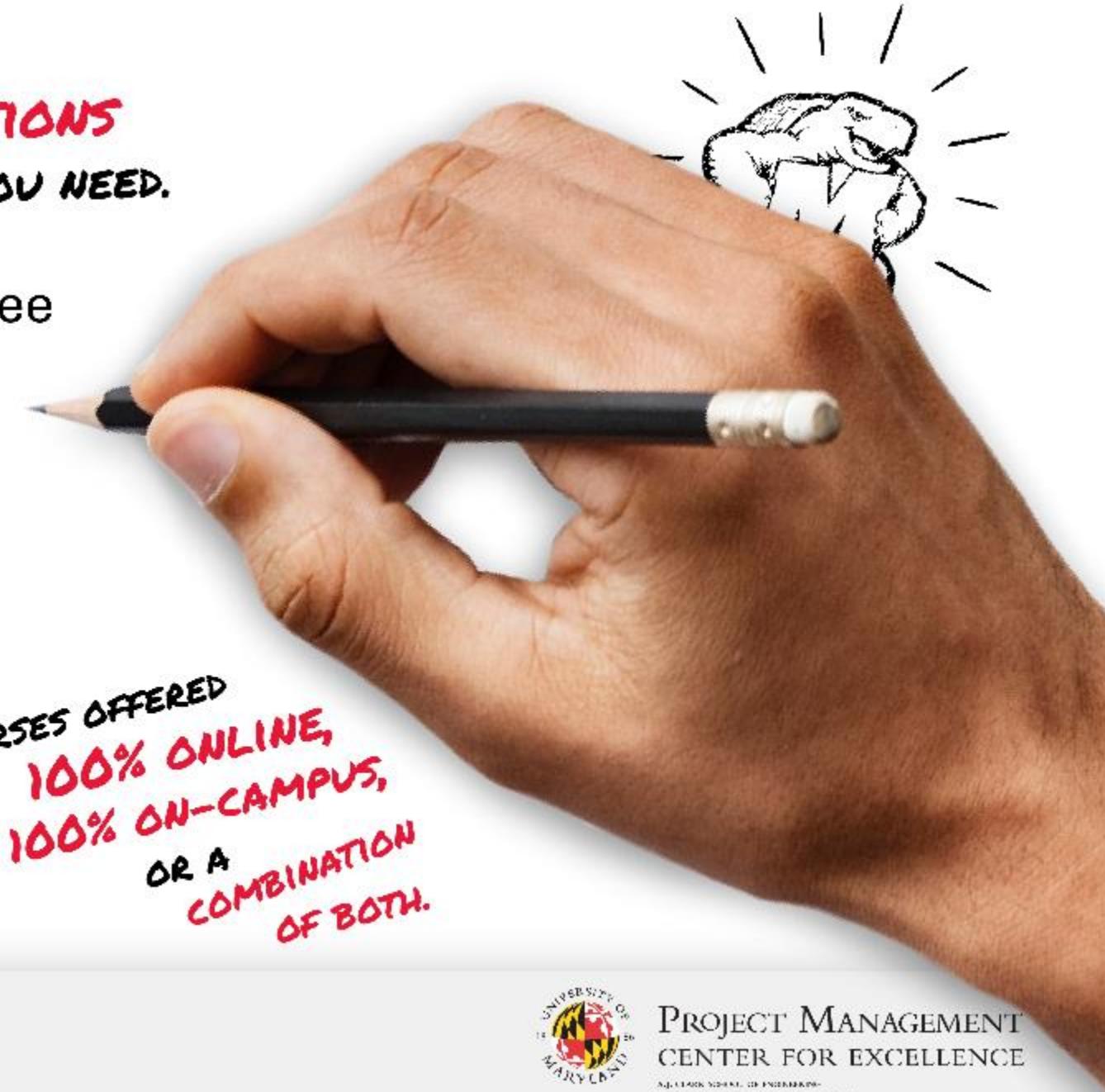
Project Management skills

- Master of Science
- Master of Engineering in Project Management
- Graduate Certificate

COURSES OFFERED  
100% ONLINE,  
100% ON-CAMPUS,  
OR A  
COMBINATION  
OF BOTH.



MORE  
INFORMATION  
AT  
**PM.UMD.EDU**



PROJECT MANAGEMENT  
CENTER FOR EXCELLENCE  
COLLEGE OF ENGINEERING  
COLLEGE OF COMPUTER & INFORMATION SCIENCE