

The background features large, stylized letters 'A' and 'C' in a light green color, set against a dark green background. The 'A' is on the left and the 'C' is on the right, both rendered in a bold, sans-serif font.

**ALGONQUIN**  
COLLEGE

Corporate Training

**LEARN AGILE IN 59  
MINUTES**

**with Peter Paul**

[training@algonquincollege.com](mailto:training@algonquincollege.com)

Version 1.0 Sep 2018

# Presentation Objectives

In 59 minutes, you will learn some:

- AGILE's basic principles
- AGILE artifacts
- AGILE rituals and ceremonies
- An AGILE variant called SCRUM
- Roles and responsibilities of team members and stakeholders

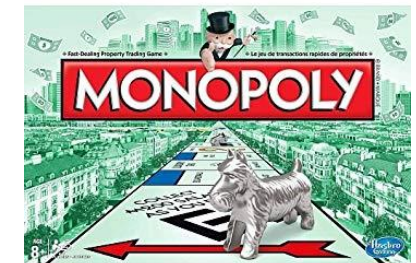
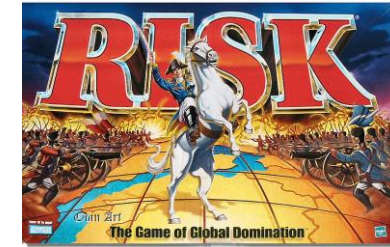
# Objectives



Develop a new game with all the necessary components that we can readily develop into a commercial product

Conditions:

- Working version of the new game is developed in an hour or less
- Game can be learned in 7 minutes or less
- Can only use the resources available within the room (assume no Internet access)
- No-one sits idle
- Only one voice of the customer – me!



## Exercise: organizing into teams

An opportunity to meet others and to sample Agile techniques in action!

- Pull a card from the deck
- Go to those with the same colour card (red or black)
- If there's more than 7 people divide by suit (diamonds, hearts, clubs, spades)
- If there's still more than 7 people divide into numbers  $\leq 7$ , 8 plus face cards

3 MINUTES MAX!

Self-organizing is  
one of the  
underlying principles  
of Agile

DID YOU KNOW?  
Frasier ran for 13 seasons  
because it never  
underestimated the  
intelligence of its audience  
(stakeholders)

## Round 0: Ideation

8 MINUTES MAX!

1. How about a game based on:
  - chance (card, dice)
  - skill (knowledge, trivia, draw, doodle, charade)
  - responses or votes (Cards Against Humanity)
2. Existing board game with a different theme e.g., Monopoly using BABOK v3
3. A whole new game?

You have 8 minutes to come up with a single viable idea to present back to me

No-one sits idle!

No bottlenecks!

No rubberneckerers!

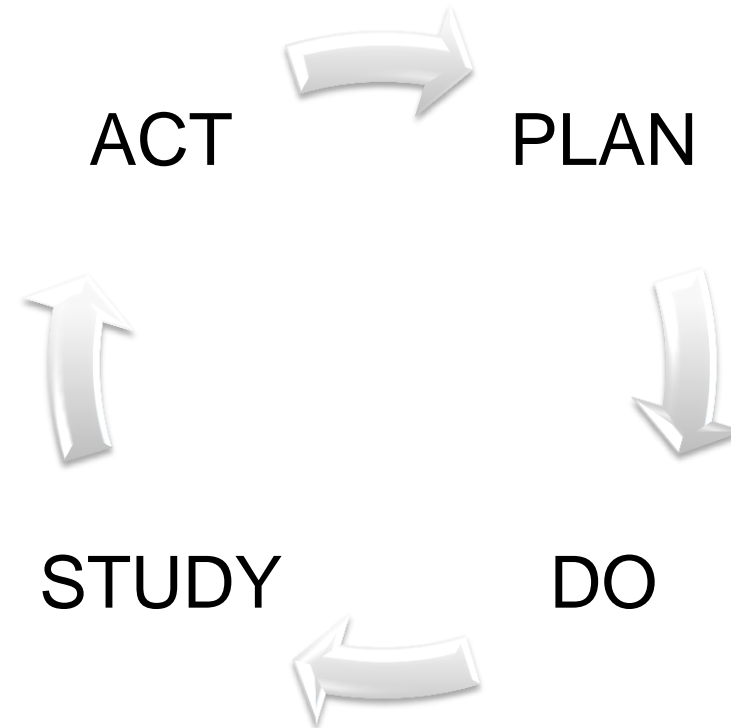
Lots of techniques at your disposal (brainstorm, clustering, NGT, multi-voting) to get the team to arrive at a single idea

# Retrospective

Can you comment on:

- 3 things we did well?
- 3 things we could do better?
- Is there a minimal shippable product?
- Was “value” (business, economic, personal, learning, social) demonstrated?
- Did new requirements emerge?

**4 MINUTES MAX!**



<https://deming.org/explore/p-d-s-a>

## Round 1: Making it work!

1. The goal for this 8 minutes is to demonstrate how this game will work!
2. What are the rules of the game?
3. What are the basic materials that you need to play the game?
4. Can it be learned by someone not on the team in 7 minutes or less?

8 MINUTES MAX!

No-one sits idle!

You'll be given 9 minutes – no more and no less!

Set aside a little time to:

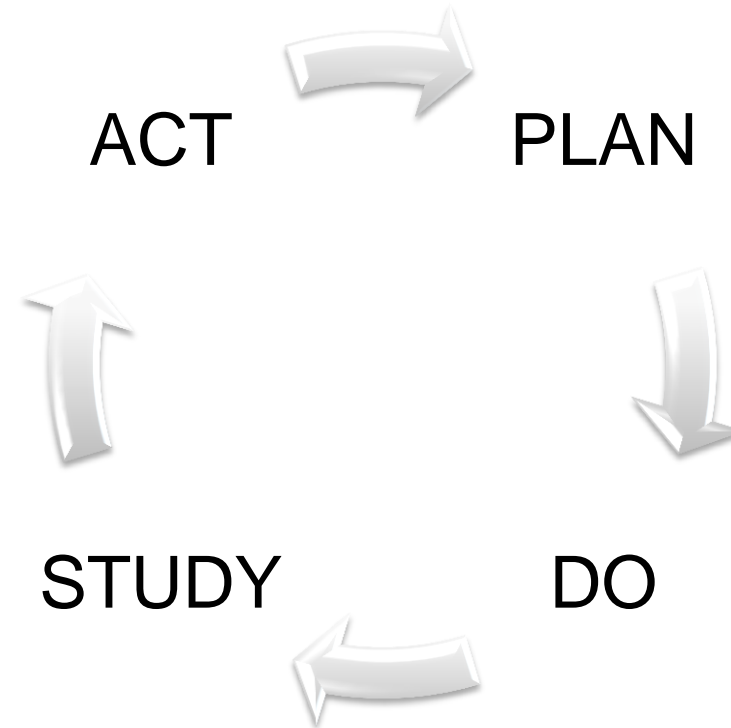
- Plan what you're going to do before doing it
- Get everything ready for the demonstration
- Be open and flexible to changes

# Retrospective

Can you comment on:

- 3 things we did well?
- 3 things we could do better?
- Is there a minimal shippable product?
- Was “value” (business, economic, personal, learning, social) demonstrated?
- Did new requirements emerge?

**4 MINUTES MAX!**



<https://deming.org/explore/p-d-s-a>



## Round 2: Make it real!

Agile Manifesto values  
working software over  
comprehensive  
documentation

Agile Manifesto  
values responding to  
change over  
following a plan

The goal for this 8 minutes is to incorporate any changes suggested by the **Product Owner** (Peter) into this **iteration** (round)

Consider any feedback regarding team performance that you could implement in this round

- Do you have a clear idea of what's being asked?
- Are there any “rules” on how to get this done?
- Are there any “rules” on how resources should be applied, requested, allocated, **shared**?

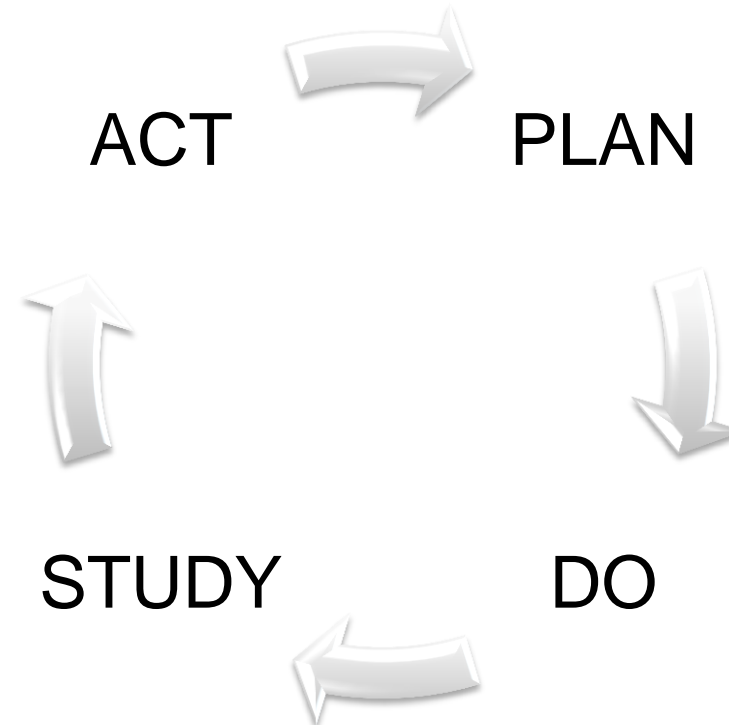
8 MINUTES MAX!

# Retrospective

Can you comment on:

- 3 things we did well?
- 3 things we could do better?
- Is there a minimal shippable product?
- Was “value” (business, economic, personal, learning, social) demonstrated?
- Did new requirements emerge?

**4 MINUTES MAX!**



<https://deming.org/explore/p-d-s-a>

## *(Optional) Round 3: Put your back into feedback!*

The goal for this 8 minutes (optional) is to play the game and obtain real feedback from others outside the development and delivery team

- Does the game meet the original objectives (***Minimum Viable Product***)?
- Is there any fine-tuning or adjustments that should be made to the game?
- Do all the components come together to present a real experience of the game (***vertical or horizontal slice prototyping***)?

Delivering incremental value across a time frame is possible!

Time-boxing is one valued characteristic of Scrum

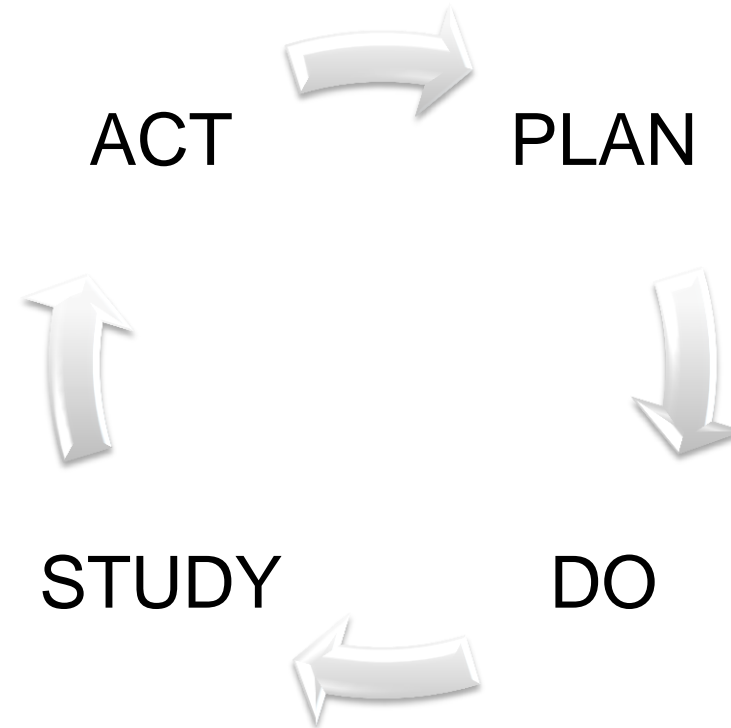
8 MINUTES MAX!

# Retrospective

Can you comment on:

- 3 things we did well?
- 3 things we could do better?
- Is there a minimal shippable product?
- Was “value” (business, economic, personal, learning, social) demonstrated?
- Did new requirements emerge?

**2 MINUTES MAX!**



<https://deming.org/explore/p-d-s-a>

# Debrief

- What did we learn?
- How did it feel?
- What can we take away from this and apply right away?

# Objectives

Develop a new game with all the necessary components that we can readily develop into a commercial product

Conditions:

- Working version of the new game is developed in an hour or less
- Game can be learned in 7 minutes or less
- Can only use the resources available within the room (assume no Internet access)
- No-one sits idle
- Only one voice of the customer – me!

**Minimum Viable Product (MVP)**

**Prototype**

**Product Owner**

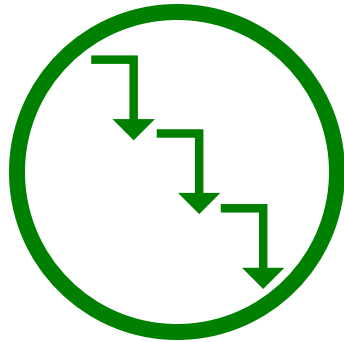
## Round 0: Ideation Debrief

- **Seeding:** planting ideas and watching them grow.
- **Brainstorm:** gather ideas regardless of how crazy or out there they are. Quantity versus quality.
- **Clustering or Nominal Group Technique:** gather and display ideas, look for commonalities, duplicates, and tease out other ideas from the collaboration. Look to name the ideas if possible.
- **Multi-voting:** gaining group consensus on ideas that will go forward. This would only be an option if the Product Owner doesn't outright make a decision; otherwise the PO is the FINAL word and decision-maker.
- **Product Evaluation or Sprint Demo:** you need to show what you've been doing; time to prepare for a demo should be allotted within the sprint.

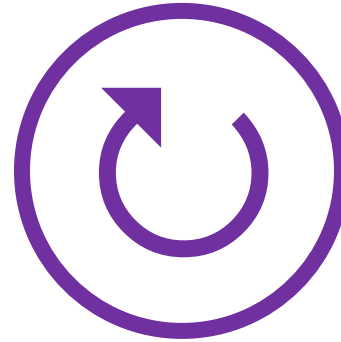


# Solution Development Evolution

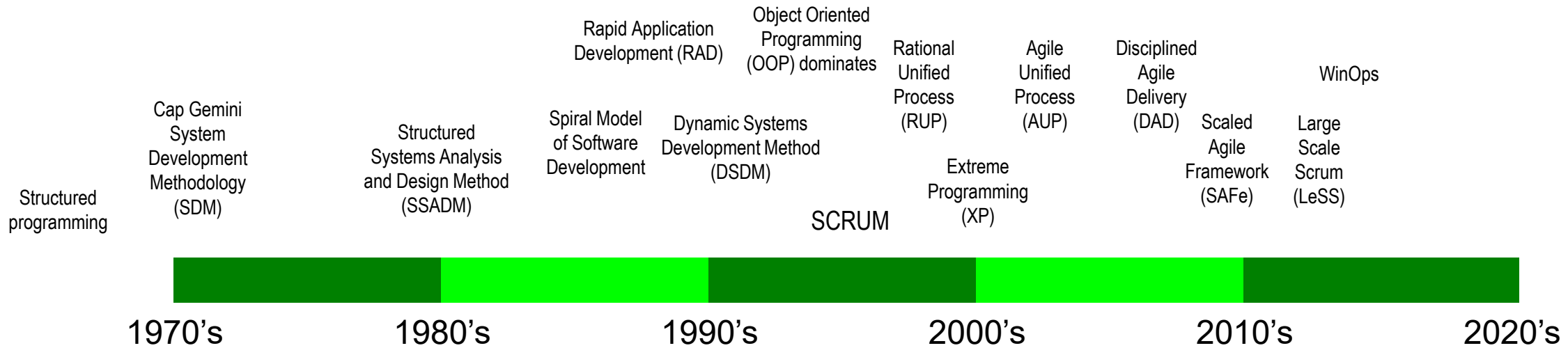
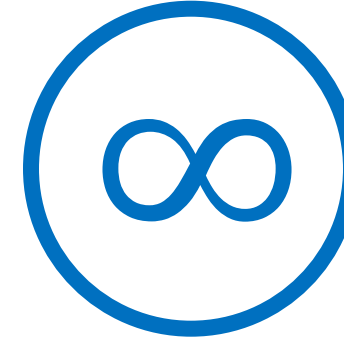
## Waterfall



## Agile



## DevOps





# Agile Manifesto

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

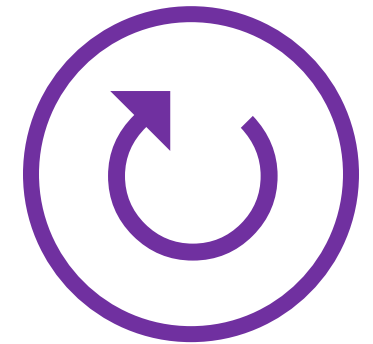
While there is value in the items on the right,  
we value the items on the left more.

February 2001

# Principles behind the Agile Manifesto

## The Principles behind the Agile Manifesto:

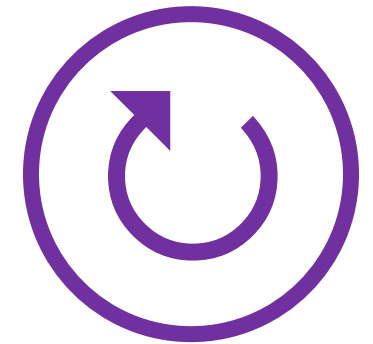
- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- 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-face conversation.



# Principles behind the Agile Manifesto

## The Principles behind the Agile Manifesto:

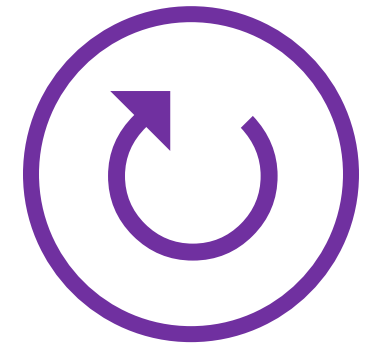
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity – the art 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 a methodology – it's a mindset

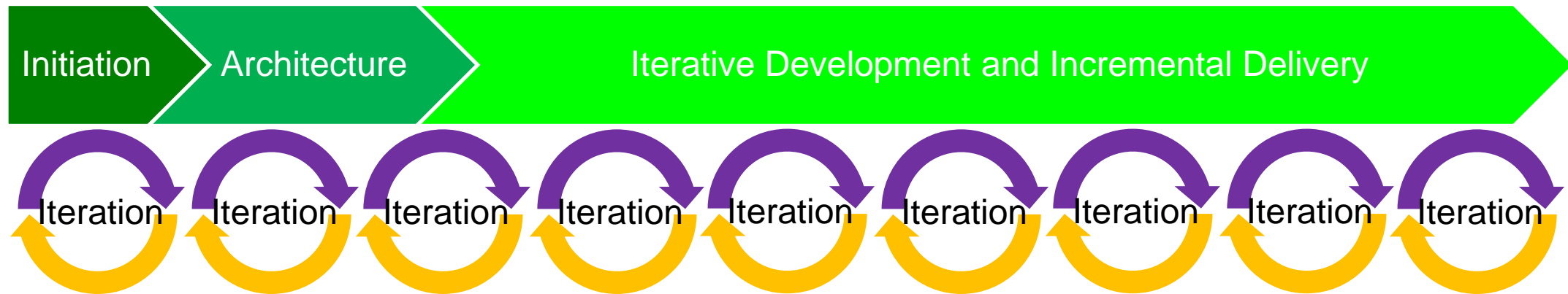
Although Agile variants may advocate an *approach* to maximize effectiveness:

- Agile doesn't embrace any specific processes (individuals and interactions over processes and tools)
- An Agile mindset embraces the flexibility of any approach at one's disposal to maximize value and minimize waste
- Agile variants often *borrow* from each other when gaining traction / adoption. For instance:
  - User stories came from XP
  - Scrum boards came from Kanban

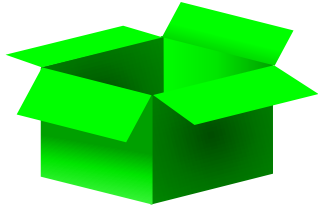


# Agile Solution Development Life Cycle

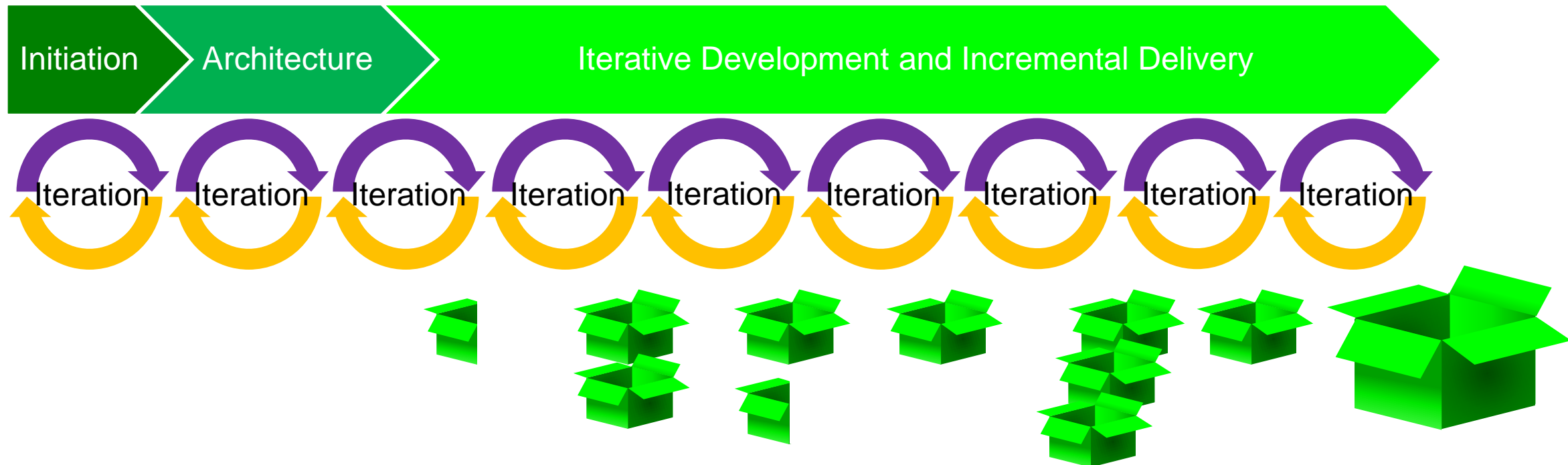
- Iterative effort is a key characteristic of an Agile project
- Iterations are normally sized equally in duration e.g., 6 sprints of 4 weeks each
- In *Scrum*, iterations are referred to as *sprints*



# Agile Solution Development Life Cycle

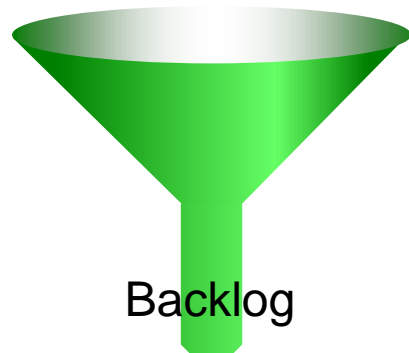


- Iterations focus on delivering value or benefit sooner rather than later
- Value or Benefit is realized through *Potentially Useful Product (XP)*, *Minimal Shippable Product* or *Minimal Viable Product (Scrum)*, or *Minimal Marketable Features (Kanban)*.



# Agile Solution Development Life Cycle

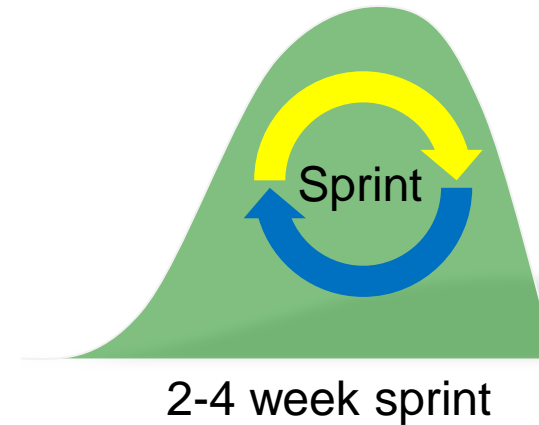
Product Owner  
prioritizes list of  
items to be done



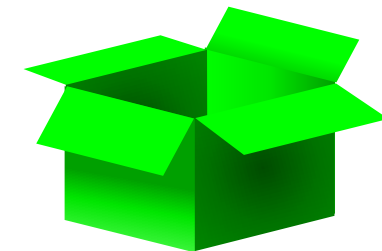
Product Owner and  
delivery team select  
items for sprint



Delivery team  
implements  
selected items



Delivery team  
demonstrates  
implementation; product  
owner gives feedback



# Agile Iterations

- Iterations are used
- Each is planned and filled with sustainable and valuable effort

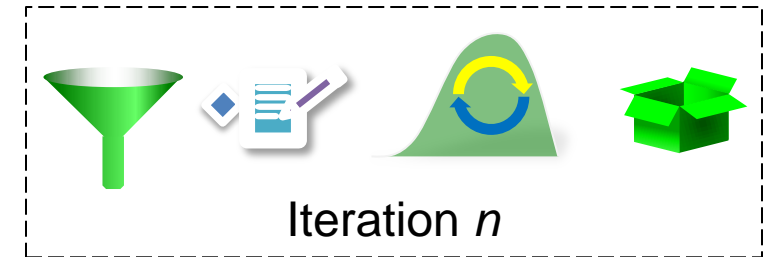




# Agile Language

Agile avoids using process and documentation-driven language of traditional / plan-based projects

For instance, the word meeting often carries a negative connotation



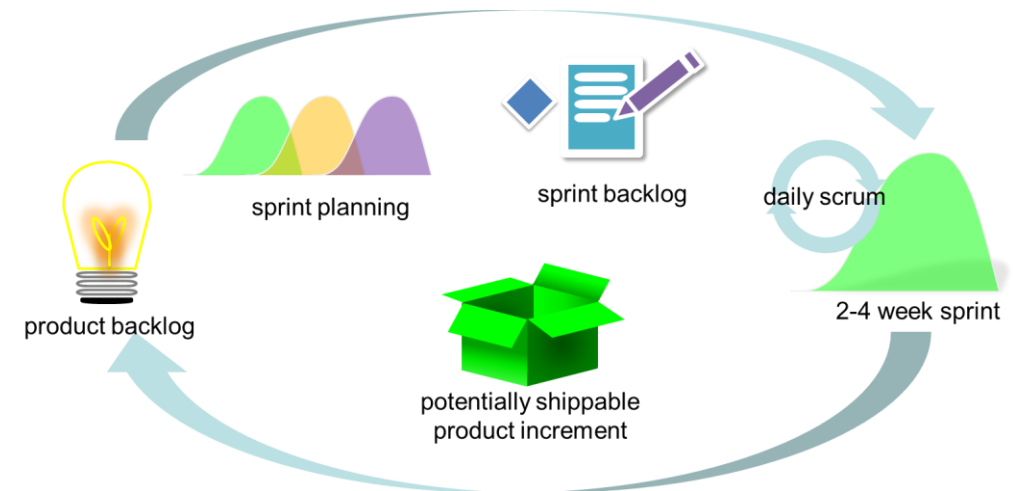
- **Artifacts** to manage key information in a valuable and least wasteful manner
- **Rituals** are face-to-face gatherings that take people away from the work they're doing to have targeted communication with each other about the context of that work. (Communication over documentation)
- **Ceremonies** represent important or key events that maximize opportunities for collaboration, understanding, and empowerment
- **Impediments** generally refer to what traditional or plan-based projects would call *issues*, *problems*, *risks*, *assumptions*, and *constraints*

# What Is Scrum?

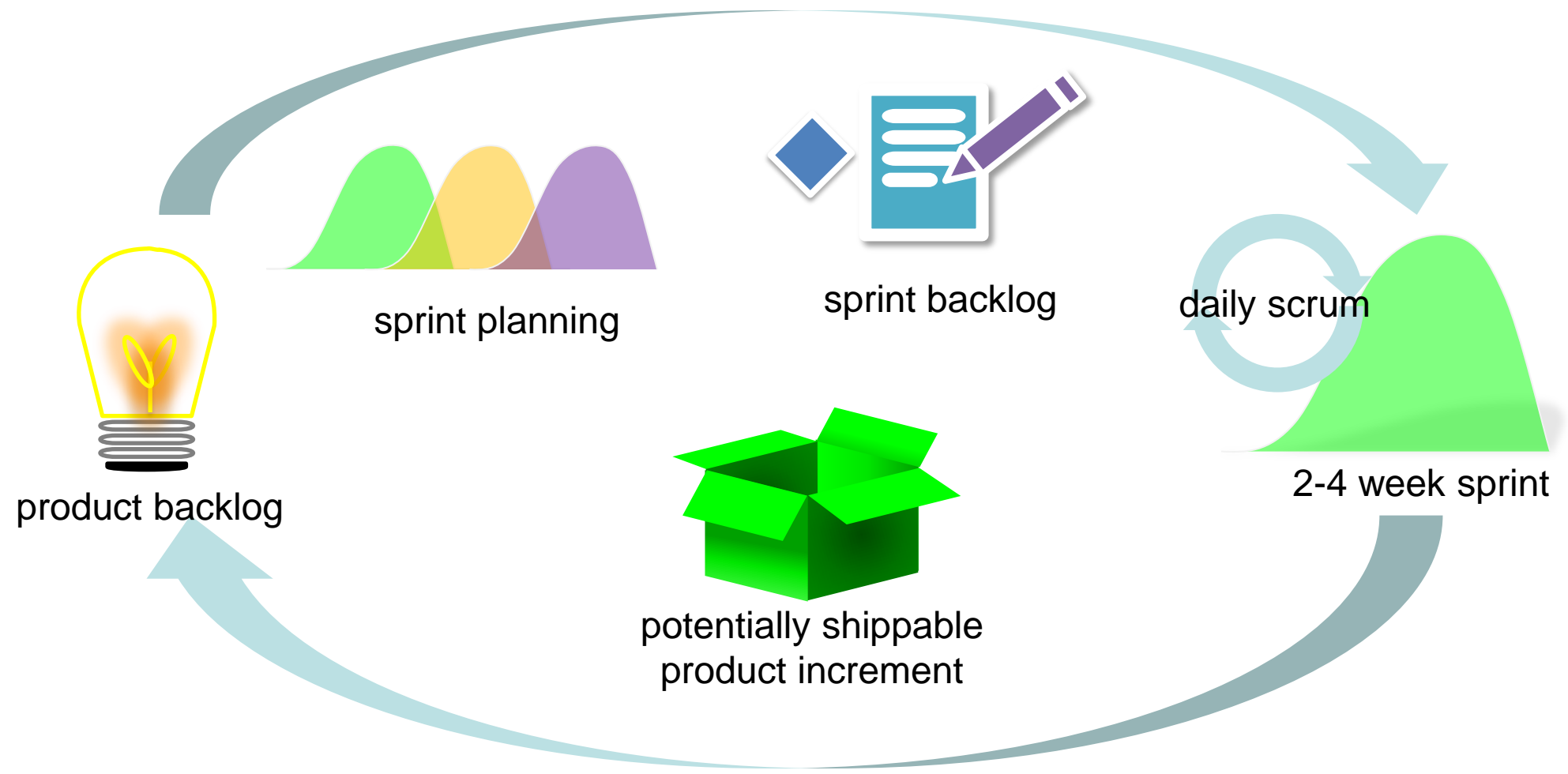
“Scrum” is an iterative Agile framework that is designed to deliver working software (or value) frequently

Scrum upholds 4 core principles over other Agile variants:

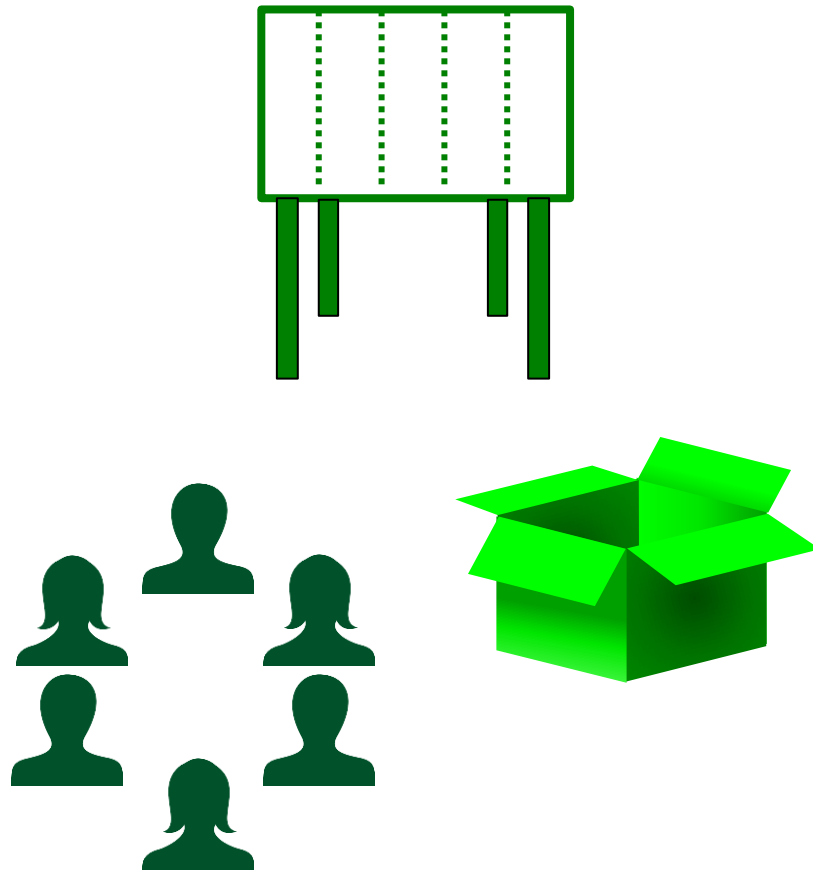
- **Time boxing:** time awareness of everything (rituals, ceremonies, artifacts)
- **Self-organization:** scrum teams organize themselves around the work (not the job titles)
- **Pull principle:** scrum teams “pull” work that they need to do (instead of being assigned)
- **Iterative Value Delivery:** build fast, learn fast (fail fast)



# Scrum Framework



# Scrum Artifacts



Scrum Artifacts include:

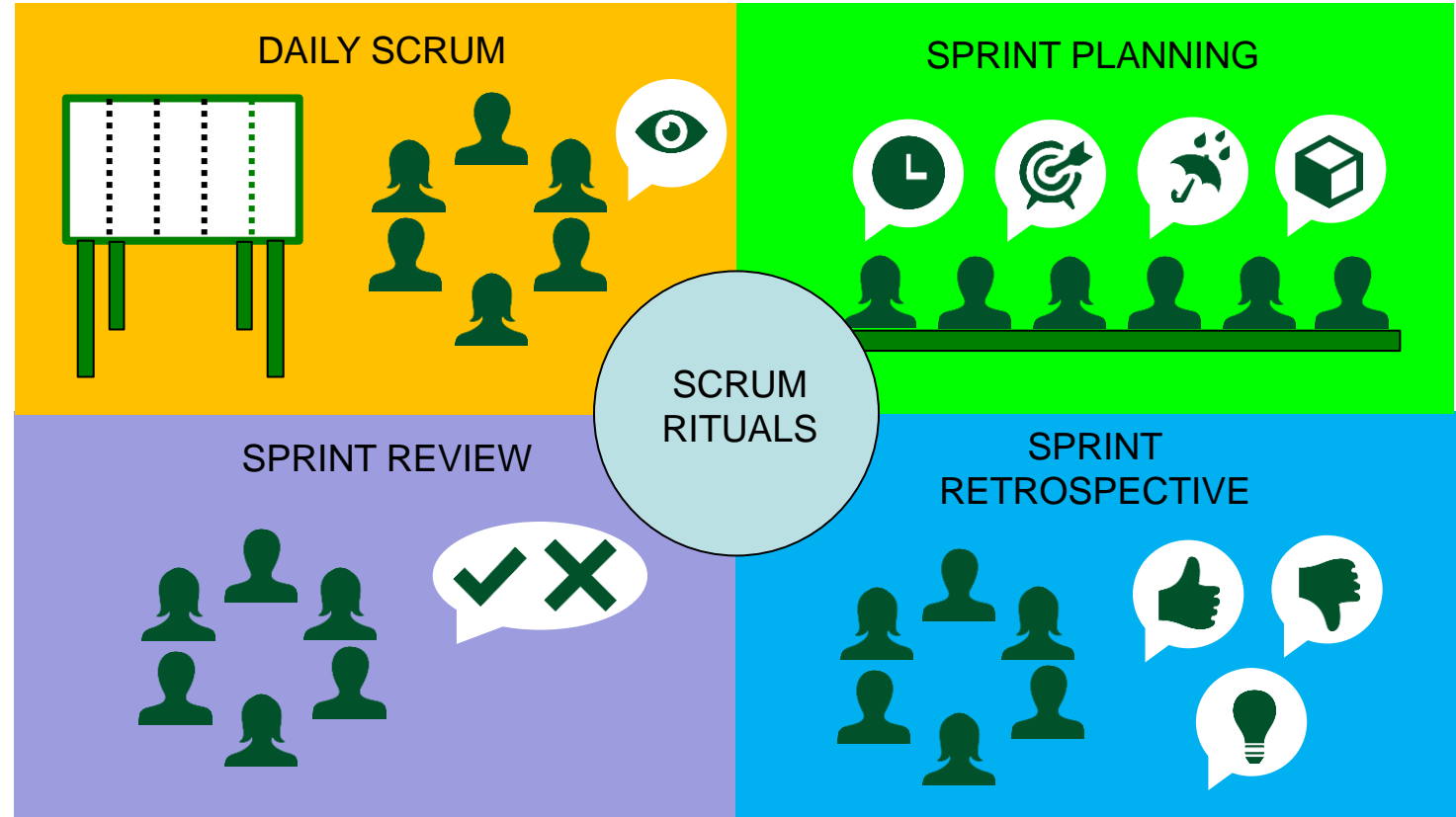
- **Product Vision box** articulates essential solution features, capabilities, and benefits
- **Product backlog (PBL)** to manage (collect, prioritize, plan) elicited requirements
- **Scrum board** to visually capture and track tasks and efforts including pending, in-progress and completed work
- **User stories** to articulate and maximize communication between non-technical (user) and technical team member (solution enabler)
- **Burn-down charts** to evaluate Scrum team performance and effectiveness

Artifacts can serve as ***information radiators*** when they are made highly visible, informative, and accessible

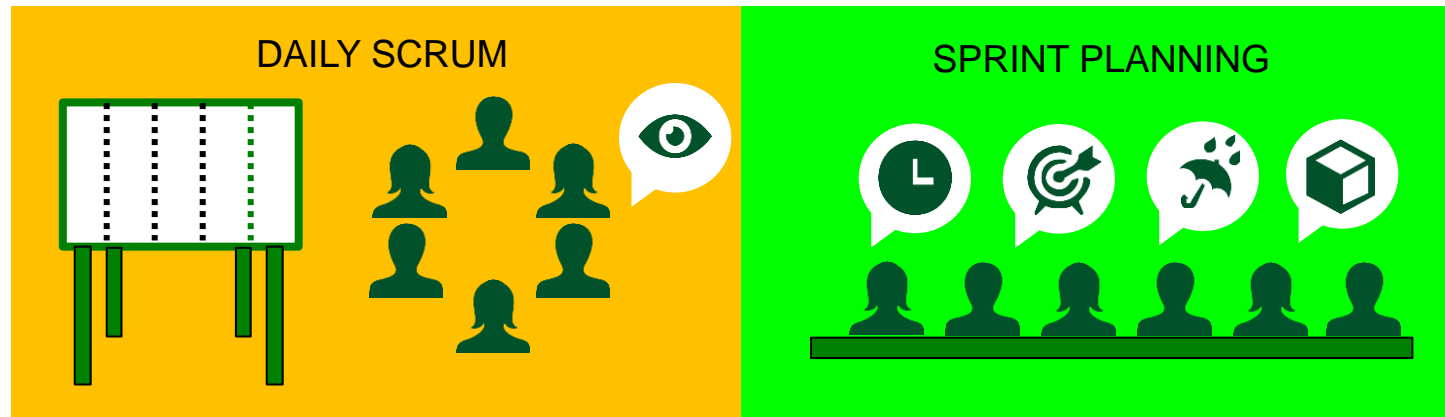
# Scrum Rituals

**Rituals** are face-to-face gatherings that take people away from the work they're doing to have targeted communication with each other about the context of that work (communication over documentation)

**Time boxing** is an important consideration for all rituals – demonstrating respect for each other's time, reminding everyone to get to the point, and remaining focused on the ritual's objectives



## Scrum Rituals (cont.)



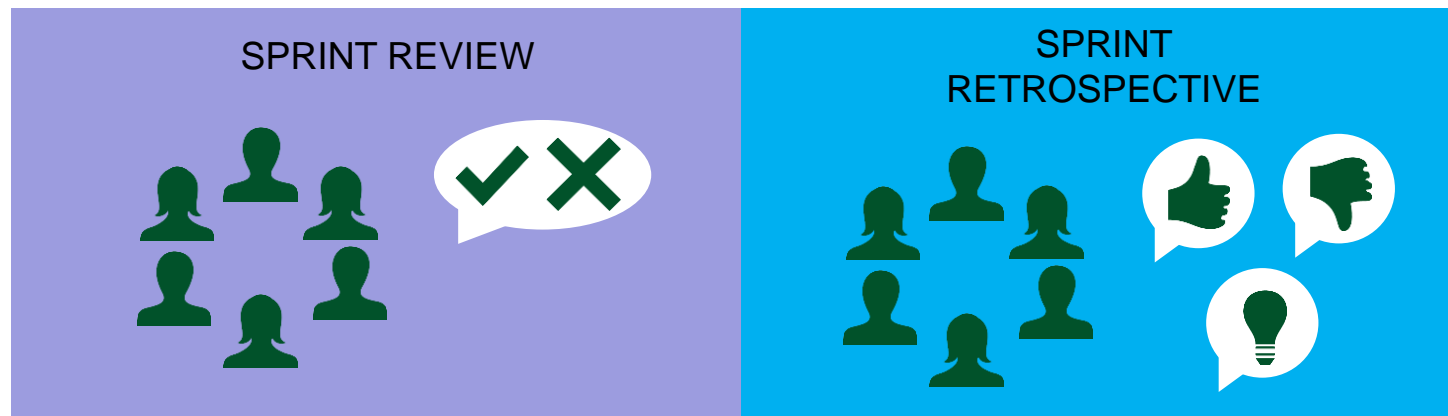
**Daily Scrum (also called Stand-Up)** is used to check in with the team frequently and manage *impediments*

**Sprint Planning** is used to determine how best to utilize the Scrum team's time and effort in delivering shippable product

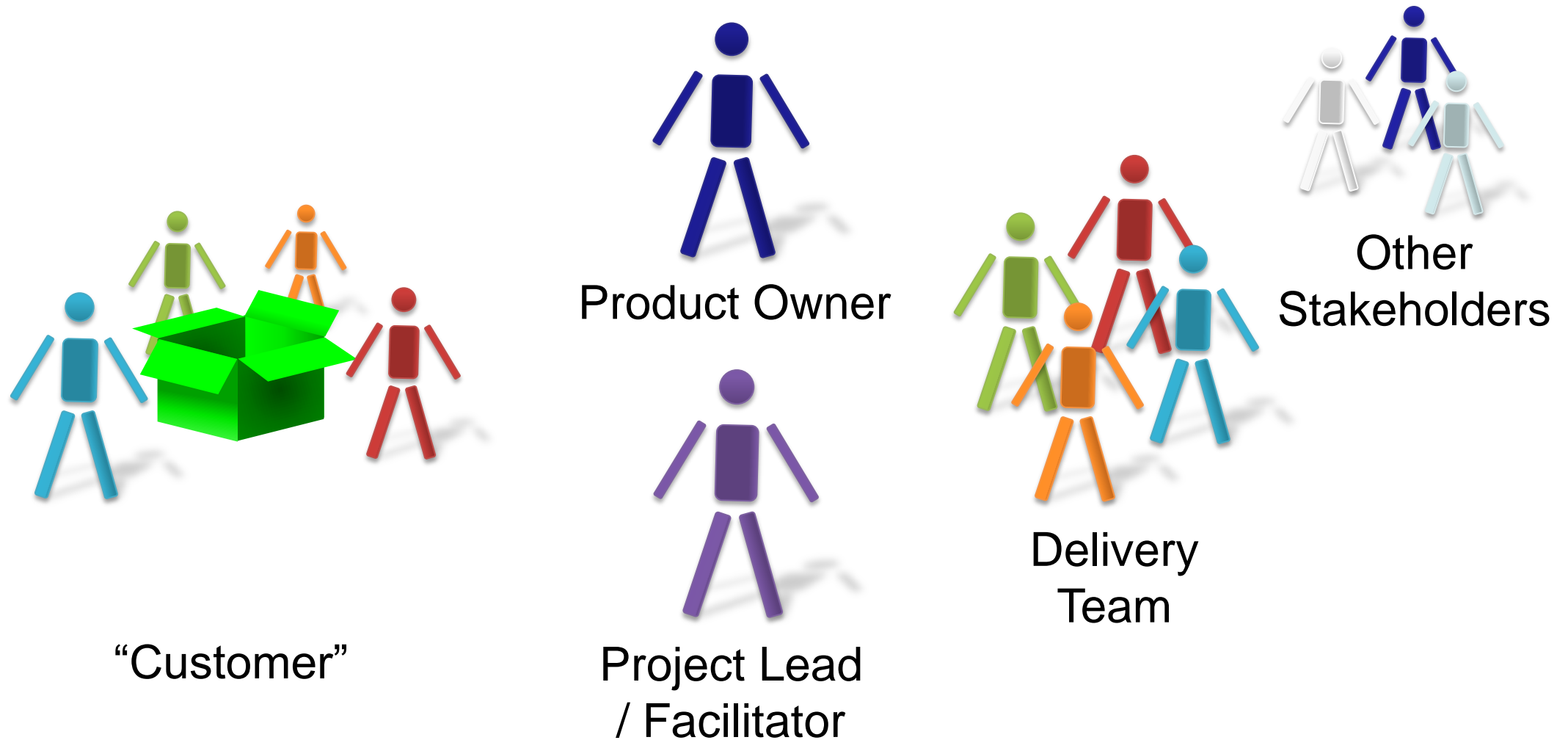
# Scrum Rituals (cont.)

**Sprint Review (also called Evaluation or Demo)** is used to gain real and meaningful feedback on a product's value or usefulness

**Sprint Retrospective** is finding and applying real lessons learned by the team to improve the next sprint



# Agile requires continuous interaction and communication





# Product Owner



**Product Owner** is responsible for maximizing the business value delivered by the team

- ONE person responsible for the backlog
- Accepts or rejects work
- Sets clear expectations for acceptance
- Knowledgeable, empowered, engaged
- Co-located with team as much as feasible
- Manages stakeholder and sponsor expectations
- Motivates team, celebrates success!

# Scrum Master



**Scrum Master** is responsible for facilitating the Scrum process and ensuring the team is delivering value

- Process facilitator
- Helps build self-organizing teams
- Removes impediments, escalates when needed
- Helps team inspect and adapt process
- Empowers the team through servant leadership
- Helps create visible information radiators
- Protects the team from disturbances

# Team

**The Team** is responsible for turning the product backlog items into increments of value each Sprint.

- Cross-functional, 7 +/- 2
- Dedicated
- Generalizing Specialists
- Deliver value in small chunks
- Focused on customer
- Build in quality



# Other team members: Business Analyst



- **Business Analyst** focuses on both uncovering / clarifying requirements (Discovery framework) and helping the team deliver within each iteration (Delivery framework)
- May also serve as **Product Owner Proxy**
- Gathers details one or two iterations ahead
- Helps business identify what it wants and breaks requirements down into valuable user stories
- Schedules and facilitates story review sessions, ensures right team members and SMEs are involved
- Works with team during sprints to clarify requirements

# Learning for Performance™

**Algonquin College**  
**Corporate Training Centre**  
340 Albert Street, 11<sup>th</sup> Floor  
Constitution Square  
Ottawa, Ontario K1R 7Y6

**Telephone: (613) 727-7729**  
[training@algonquincollege.com](mailto:training@algonquincollege.com)  
[www.algonquincollege.com/corporate](http://www.algonquincollege.com/corporate)