Agile

ITMC Presentation

April 18, 2016

History

Agile has its roots in the total quality movement. It allows for requirements instability and learning during the project.

- 1986 Takeuchi and Nonaka had a new approach to commercial product development allowing trial and error
- 1993 Scrum introduced
- 1996 Extreme programming with short cycles and frequent releases
- Other lean principles influenced agile such as the Toyota Production System, Kanban, lean manufacturing with emphasis on waste reduction
- 2010 – PMI adds Agile certification
- Today - Mixing and adapting of agile approaches is the norm. The purist approaches are abandoned

Key historical event

- 2001 - The Agile Manifesto

> “We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:
>
> **Individuals and interactions** over processes and tools
> **Working software** over comprehensive documentation
> **Customer collaboration** over contract negotiation
> **Responding to change** over following a plan
>
> That is, while there is value in the items on the right, we value the items on the left more.”

Roles in Agile

- Product Owner
  - Set and ensure common vision
  - Prioritize features/user stories
  - Negotiate iteration goals and backlog with the team
  - Decides on requirement questions and defines done
  - Ensures decision lead to building value in the product/project
- Team
  - Cross-functional, autonomous, self-organizing
  - Accountable for their commitments for each iteration
- Scrum Master/Agile Project Manager
  - Helps team self-organize, facilitates the process
  - Removes roadblocks and seeks opportunities for improvement

When to use Agile

Easy answer – it is a fit for any type of project!

….the big caveat is the people and process must be ready to use it

It is not a good fit

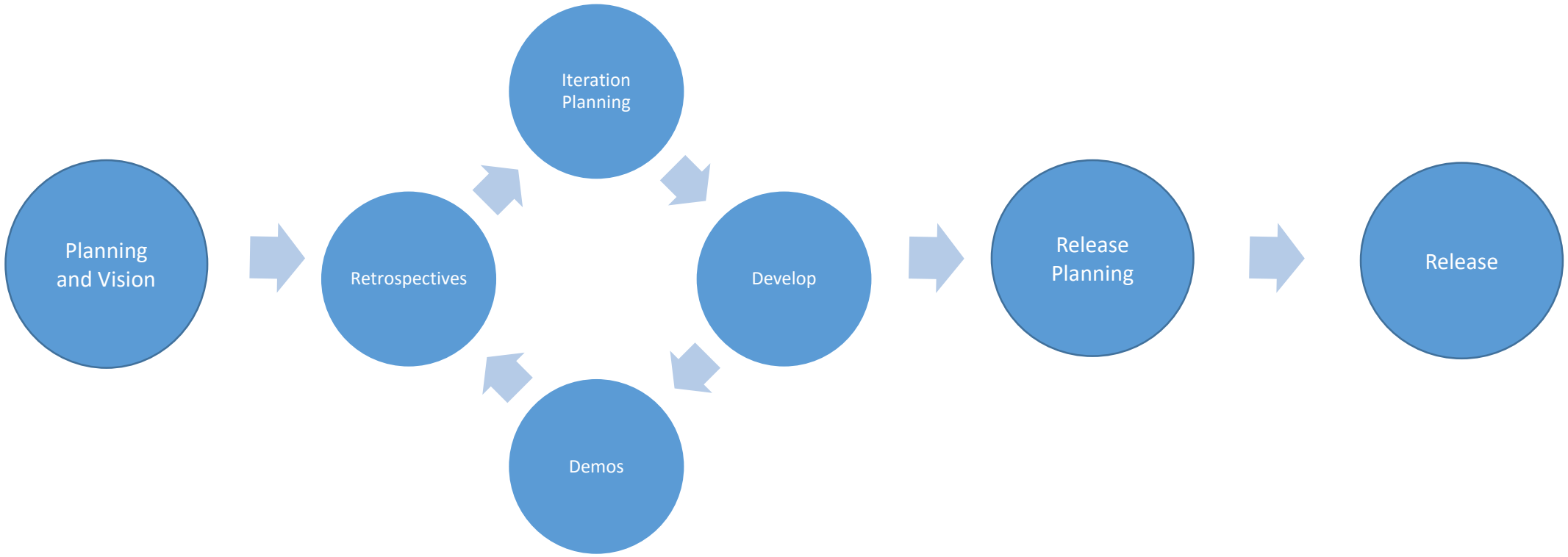
- Vendor fixed bid/fixed scope contracts
- Vendor is not experienced working with agile methodologies
- No product owner exists
- Organization not ready for flexibility and changes

There are so many ways to use Agile - it does fit with many projects

- New software implementation – minimum viable product
- Managing existing software backlogs – iteration cycle approach
- Help desk student training development – iteration cycle for development

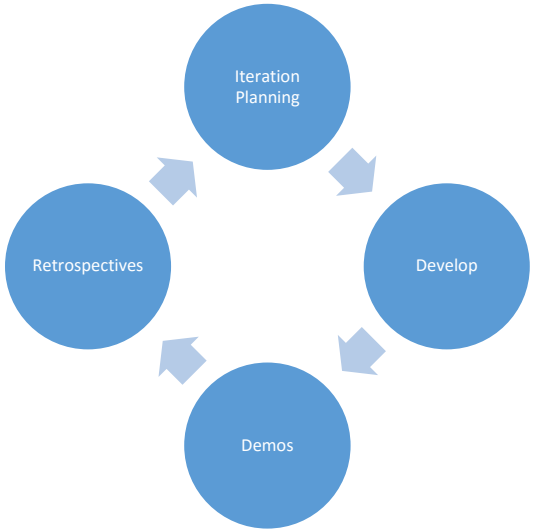
Agile Process Model

The process model for all agile methodologies is similar



Planning and vision

If you think Agile is this…..



Then you are missing a very important part of the picture.

- Product vision
- User stories
- Product backlog
- and other deliverables such as business case, team planning, release planning….

Product vision

- What are you going to make?
- Why?
- Who wants or needs it?
- How do you know?

Product Vision Statement

Statement that describes the product highlighting key attributes.

The product is a durable, light-weight tablet for kids. It will be pre-loaded with educational and fun applications targeted at kids aged 4 to 12. The exterior will be robust with a large frame offered for easy handling for small hands and offered in a variety of colors. The touchscreen will be approximately 6” x 8” and shatter/water proof. An additional feature will include the ability to easily identify the owner of the tablet. The auto lock screens will have the name of the user logged into the tablet for easy identification. The target battery life is six hours to allow uncharged operation to cover a plane flight or long road trip. It will be fully featured with Android OS to allow the child to access a full range of age appropriate applications. A cloud-based e-store will be established to hold prepackaged bundles of content specifically targeted for smaller age groups between 4 and 12. Purchasing the tablet will include a download of one content package. As the child-user grows, new packages can be downloaded allowing the ADAPTablet to grow along with them. The administration of the tablet will allow for non-tech savvy parents to setup the account and parental controls while allowing the child to customize and explore different applications and features of the tablet. The price will be competitive with other tablets in the market.

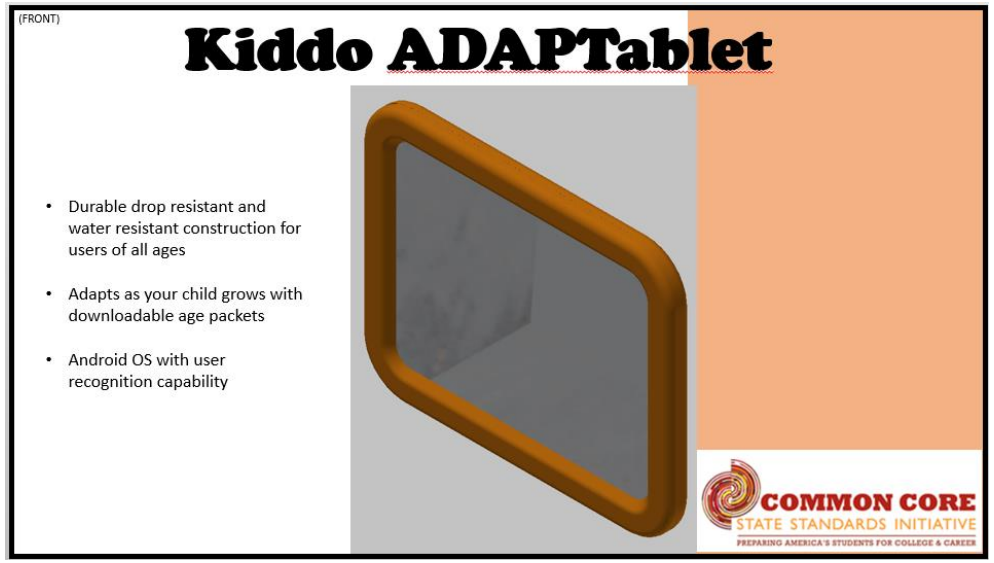
Elevator Test Statement

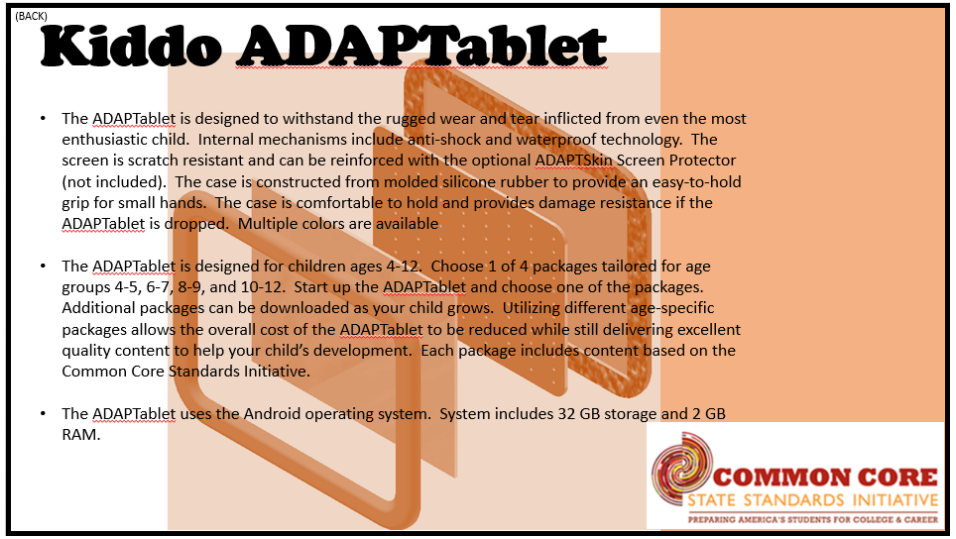
Structured statement that defines the product, target customer and value

- ***For*** children ages 4 – 12
- ***Who*** have parents looking to purchase a durable, affordable, secure tablet that provides access to both educational and fun applications and software
- ***The*** Kiddo ADAPTablet ***is a*** tablet device
- ***That*** is simple enough to be used by a child but functional enough to be used by a parent
- ***Unlike*** other tablets marketed for children which use a limited, proprietary operating system and pre-loaded software
- ***Our Product*** utilizes a modern, flexible Android OS combined with an ever-growing online application store to support changing installed software as your child grows and learns.

Product Vision Box

Visual representation of the product box to align on goals and customer value





Product Data Sheet

- Provides project/product background in one view.
- Similar to traditional project management approaches combining aspects of the charter, scope, requirements, etc
- Driven by the product owner with input from project team and customers

Project Data Sheet

Project Name: Tablet	Project Manager: Ben
Project Start Date: February 15	Product Manager: College of Education
Clients:	Executive Sponsor: Dean
Children ages 4 - 12	Client Benefits:
Adults who purchase for children age 4 - 12	Durable full featured tablet that is flexible to meet the needs of
	children ages 4 - 12 as they grow with easy to use parental controls
	that allow controlled exploration and learning
Project Objective Statement:	Performance Attributes:
Produce a tablet for children to be available to ship by	Rugged and durable
October 1. The product should sell at $350 or less pric	Full featured operating system
point.	Easy to use parental controls
	Colorful, attractive to kids
	Easy to identify
	Preloaded educational applications for 4 - 12 year olds

Trade-Off Matrix:					Age based interface
	Fixed	Flexible	Accept	Target	
Scope		x			Product Architecture Guidelines:
Schedule	x			Ship by Oct 1	Android OS
Resource			x		Easy to maintain/update user interface
Stability			x		Fast response time, fast load time for software
					Easy to connect to wifi
					Obvious buttons and controls

Feature breakdown structure:	Issues and Risks:
Hardware:	Procurement lead time
Screen - water and shatter resistant, 6 - 8"	Assembly capacity
Case - grippable, shock resistant, colorful	Ability to meet price point
Expandable storage, flash storage at 32GB	Quality of age based educational software bundles
Camera	Competitor products
Biometric identification	
Long battery life	Major Project Milestones:
Software:	Project Start February 15, 2015
OS - Android	Planning March 15, 2015
Interface - Age based, subscription	Iteration 1 May 15, 2015
Easy to use parental controls	Iteration 2 June 15, 2015
Screen lock with name	Iteration 3 July 15, 2015
	Iteration 4 August 15, 2015
	Final product September 15, 2015
	Ship product October 1, 2015
	Project End October 1, 2015

Exercise 1 – Create a product vision

Each person on your team should select a role for the exercise (product owner, agile PM or team member), then complete the deliverables for your product/project

- Product vision statement
- Elevator test statement
- Product vision box
- Product data sheet

User Stories

- The product vision is further defined through user stories
- User stories define the customer wants and needs
  - A request for something of value
  - Turns a feature into a story
  - Key artifact for determining the work in the iterations
  - Used for estimating and velocity through story points
  - User stories are a great tool for analysts to define requirements with the customer or product owner
- Defines “done”
- “As a _____ I want to _____ so that _____”

Intro to Scrum Video

- 8 minute video - https://www.youtube.com/watch?v=9TycLR0TqFA

User stories

Use invest to make sure your user stories are sized right

- I – Independent – self contained, movable
- N – Negotiable – changeable until committed
- V – Valuable – delivers value to customer/user
- E – Estimable – must be able to estimate the story
- S – Sized/Small – Not too big to plan (no epics)
- T – Testable – must be able to test done and good

Exercise 2 - User stories

Define what a customers wants or needs the product to do

- Defines “done”
- “As a _____ I want to _____ so that _____”

In your team, review the features and customers you identified from the product vision exercise. Create your user stories to define the feature.

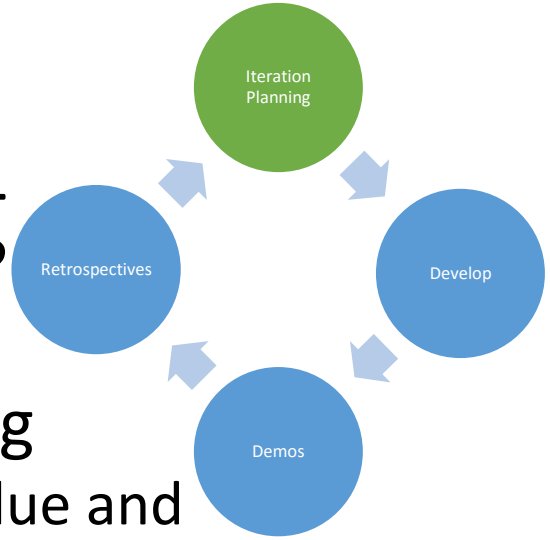
Story points

- Provides estimates for sizing the user stories
- Factors in the development, testing and risk
- Uses a scale, typically the Fibonacci sequence (1,2,3,5,8,13,20,40,100)
- 4 minute video https://www.youtube.com/watch?v=qE5h6Q2lMsI&feature=related

Exercise 3 – Estimate story points

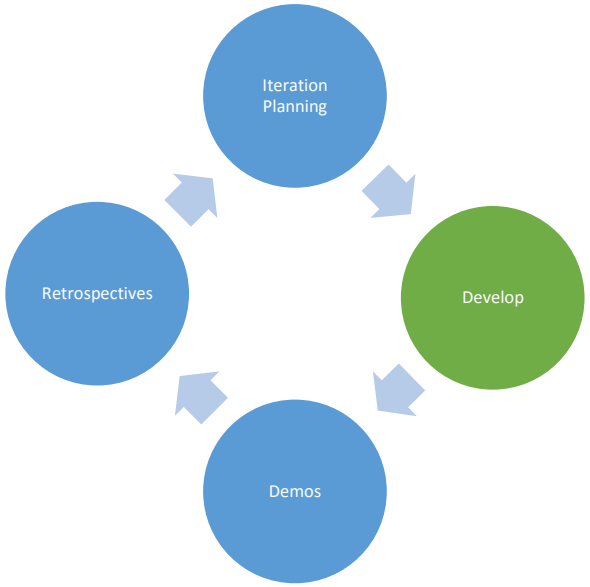
- Assign story points to your user stories
- Use the Fibonacci sequence (1,2,3,5,8,13,20,40,100)
- If you have an epic, break it down into smaller stories

Product Backlog and Iteration Planning



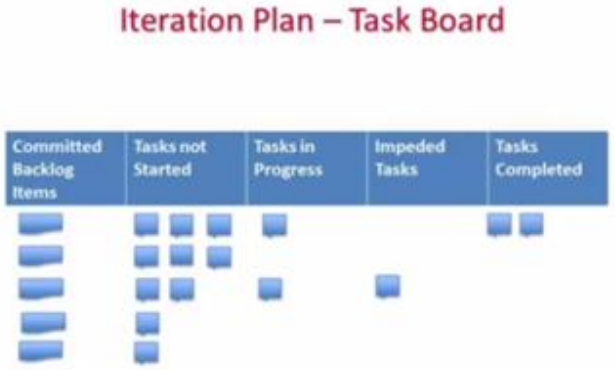
- The user stories along with issues form the product backlog
  - Product owner is responsible for removing items no longer of value and reprioritizing as needed
  - Prioritized at each iteration
  - Product owner is responsible for ensuring there is value from the user story
  - Higher priority items are more refined/detailed than lower priority items – some items may be at a feature level for a low priority feature, high priority items should be at a user story level
- Product backlog can be a list of enhancements or issues – it is not only for new projects, the concept can apply to your existing software or projects
  - Examples of CRM development, 25Live configuration using governance teams

Organizing the iteration

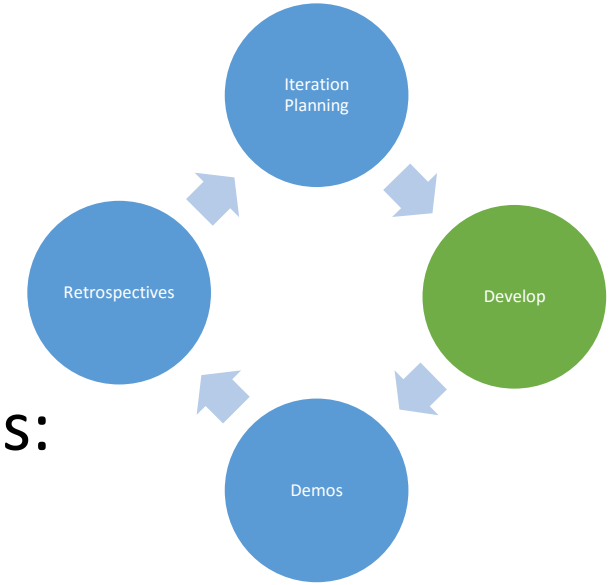


There are a number of ways to organize the information related to the iteration

- Information should be visible – see the task board example
- The scope is fixed for the iteration with the user stories defining done
- There is flexibility in who does the work and when it is started/finished within the iteration timeframe
- Regular scrum or stand-up meetings allow the team to share information on progress



Scrum meeting

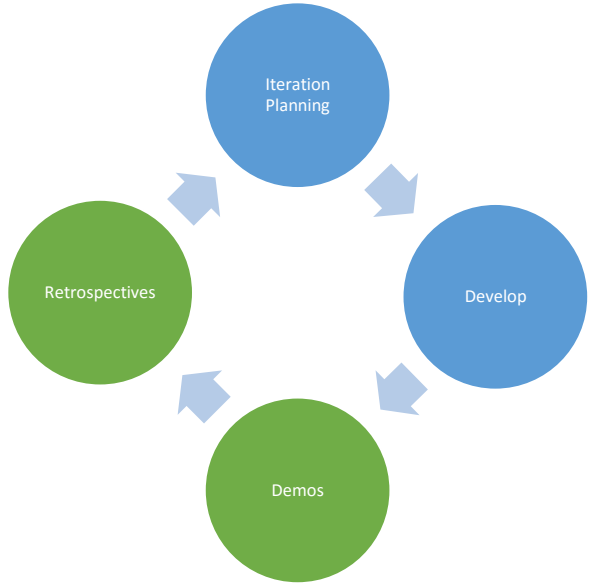


Each person in the team answers the following questions:

- What did we get done since the last meeting?
- What are the plans until the next meeting?
- Any roadblocks?

Adapt this format to fit your needs. This may be a weekly status meeting, email status or other type of information sharing that works for your environment.

Demo and Retrospective

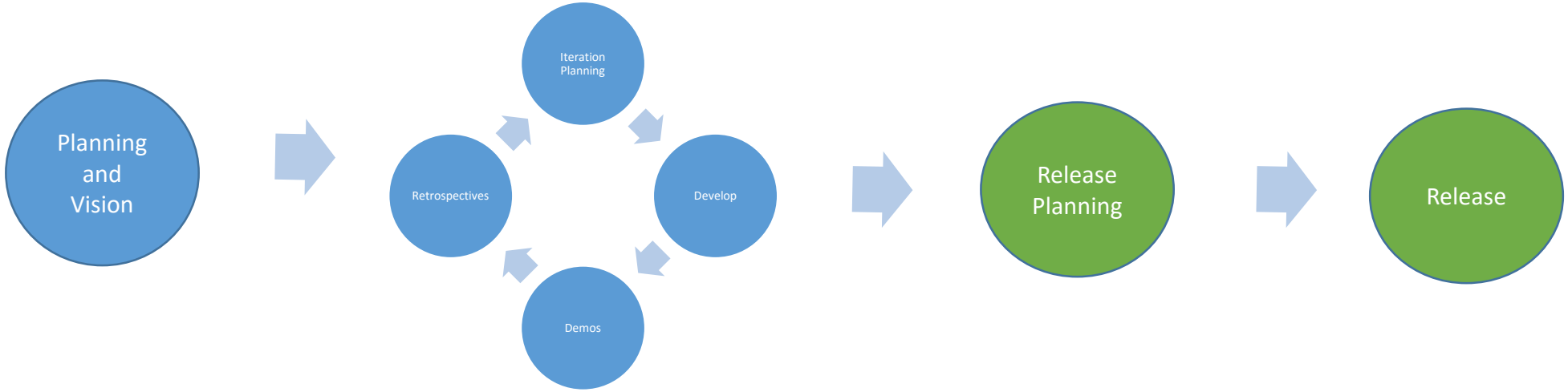


At the end of each iteration..

- Demo the iteration for the product owner
  - Owner accepts or small issues addressed in this iteration
  - Allows for changes in direction or new requirements since the product owner can add new items (user stories or issues) to the backlog based on the demo
  - New items are prioritized with all other backlog items into the next iteration
- Retrospective is similar to lessons learned in the traditional project management approaches
  - What went well?
  - What didn’t go well?
  - Do better next time

Release planning

- The release plan tells us when we will release something of value to production
  - High level plan spanning multiple iterations or one iteration
  - Used for product owner and team planning
  - Provides a timeline and milestones for the team
  - Similar to the traditional project plan with start, finish dates



Feedback

- What parts of Agile can you use in your organization?
- What are the biggest roadblocks?
- Are there any Agile concepts you are already using?
- What do you plan to do next with this information?