
accenture technology
Agile University: Delivery School
Agile at Accenture Overview
High performance. Delivered.
Strategy | Digital | Technology | Operations

Agenda

- Agile At Accenture
- Accenture Scrum Enhancements
- Initiation Overview
- Sprint 0 Overview

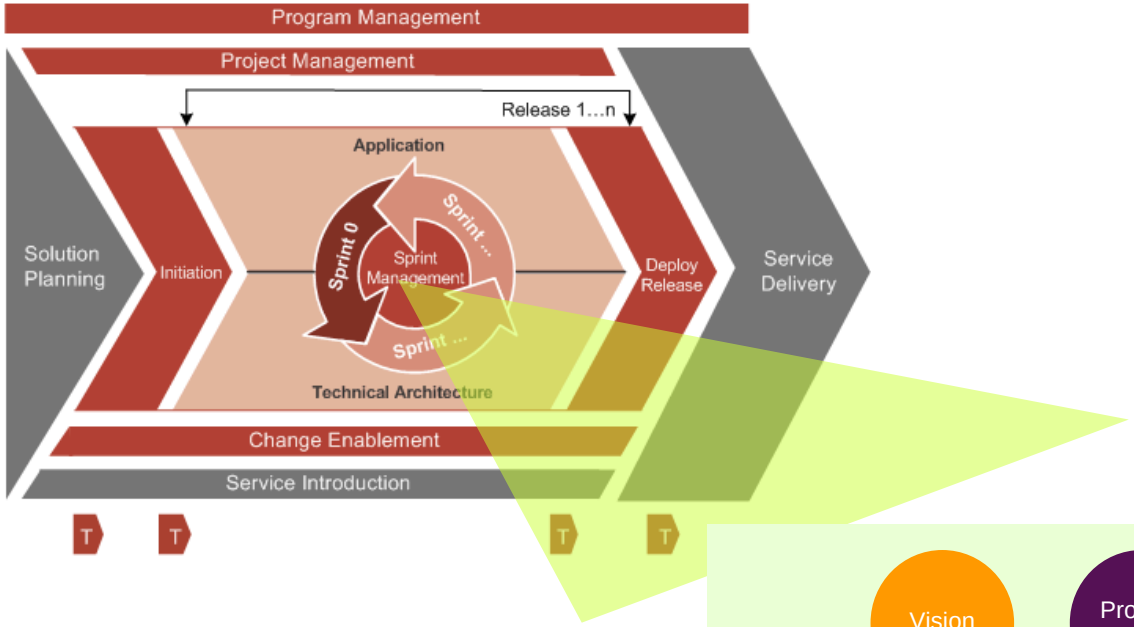
Why Agile at Accenture?

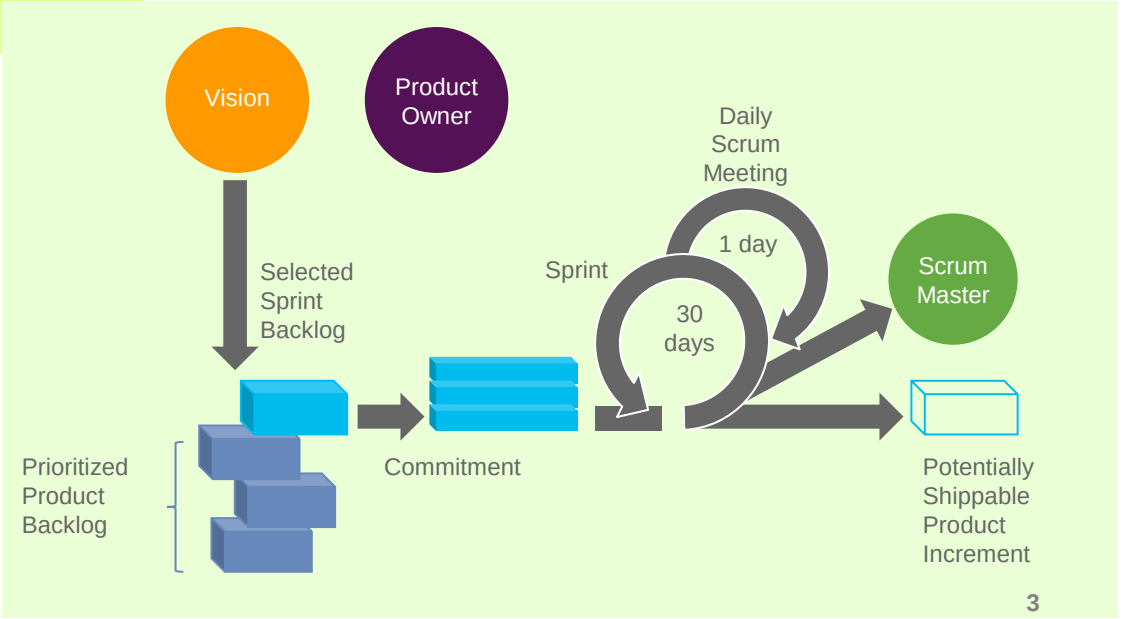
- Accenture has observed a strong interest from several clients towards adopting Agile approach to software development.

- Guidance on how to apply Agile in a dispersed team consolidated with the existing Accenture methodologies and past experience is a key requirement from our Clients and strategic for positioning Accenture ahead of our competitors in this area.

ADM for Distributed Agile is a flexible and scalable methodology for single and multi-site projects covering both sales and delivery stages, focusing on end to end delivery. It supports both Custom and Package development, implements Scrum, extreme programming practices and some Lean principles.

Agile at Accenture - ADM for DA





Agenda

Agile At Accenture

Accenture Scrum Enhancements

Initiation Overview

Sprint 0 Overview

Accenture Scrum Enhancements

- **Solution Planning**, **Project initiation**, and **Sprint 0** phases focus on providing appropriate and advanced Governance, Communication, Collaboration and Tools structures.
- Complex **Sprint Management** is achieved through the use of a suite of Delivery Models and Scalability Practices, such as, Scrum of Scrum, Product Owner and Product Backlog patterns; tailored to fit various project contexts.
- **Sprint Management** is further bolstered through the use of Distributed Delivery Practices focused on Tools, Infrastructure, Practices, People and Processes.
- A **Project Manager** role has been added outside of the SCRUM to perform upward stakeholder reporting and maintain an oversight of the overall project engagement, including large scale activities and/ or distributed teams.
- A **Deploy Release** phase at the tail-end of a release allows for the convergence of multiple distributed and parallel inter-project 'sprint-releases' to be aligned for effective and comprehensive **Test** phase (including full-scale Integration and Performance Testing) and **Deploy** phase.

Agenda

Agile At Accenture

Accenture Scrum Enhancements

Initiation Overview

Sprint 0 Overview

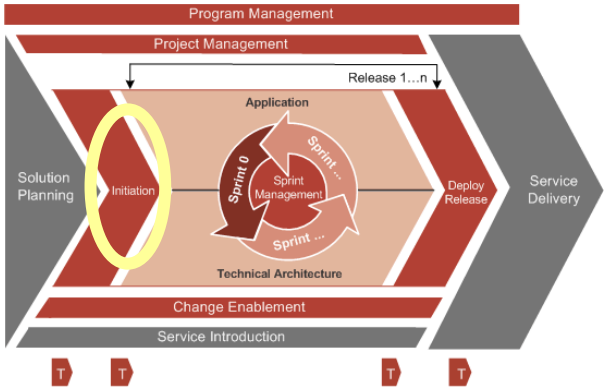
Deploy Release Overview

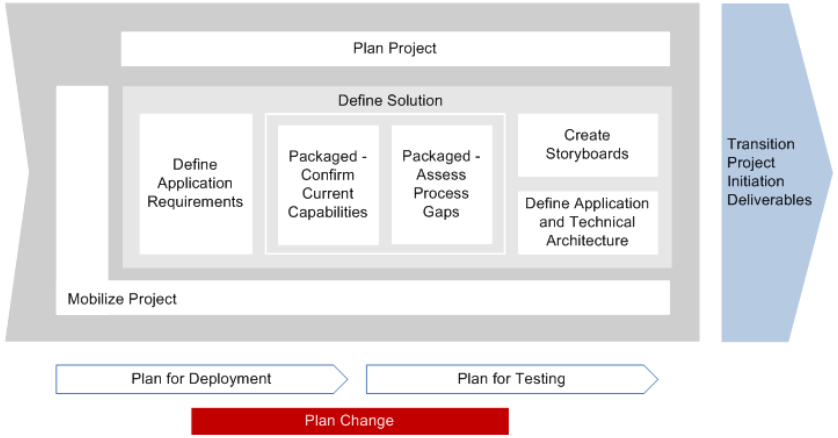
Initiation Phase

Why do we need a Initiation phase for Agile projects?

- Standard industry **Agile** methodologies do not focus on the initial planning activities.
- For any project, it is important to understand the **big picture** and form a rough **road map**.
- Define **high-level requirements** and create a high-level design or blueprint for each area of the solution.
- It is key that both the **Client and Accenture** are aware of the overall Plan, costs and **resource requirements**.
- **Upfront work** is required before getting into the Sprint Phase.

Initiation Overview





Initiation Phase

- Define stakeholders **Goals and Expectations**
- Define the **Project Plan –** Major Milestones, Key Release and testing cycles and Creating the Process Model
- Define **high-level requirements -** Requirements are gathered either as User Stories or Use Cases. User Stories/Use Cases are decomposed further during Sprint 0
- Confirm **Current Capabilities** and Assess **Process Gaps** **
- Create a **blueprint** of the solution
- Estimate overall **cost** and **schedule** for the project
- Create the **test approaches** and early identification of the **deployment requirements**
- Re-usable prototype(s) for Technical Architecture is built

https://methodology.accenture.com/dist_agile/#publish.dist_agile/disciplines/Initiation_516BFFD1.html

Key Roles

- **Project Manager**
  - Coordinates with all the stakeholders
  - Creates the governance structure
  - Creates the project plan
  - Get the resources on-board
- **Product Owner**
  - Often a **Client** resource: ensures that the Project Manager gets all the required help in establishing the project
  - Creates the initial set of User Stories and prioritizes them
  - Clarify the requirements during storyboarding
- **Additional Roles**
  - **User Experience Architect** in Storyboarding
  - **Architects** (Data, Business, Solution and Technical): clarify the requirements and creates the initial solution
  - **Integration Solution Architect** : confirms Current Capabilities and Assess Process Gaps **

Key Work Products

- **MG103 Governance Plan** – Hierarchical structure to provide direction and decision making for the program and the projects under it.
- **MG110 Management Plan –** The five core management plans are Project, Risk, Quality, Measurement and Configuration Management plan.
- **PG134 Tailor Waiver Request** – Documents the request for approval of the methodology tailor and waiver from the operating group quality leadership.
- **IE219 Product Backlog** – The Product Backlog is a very critical deliverable which consists of a prioritized list of functional and non-functional requirements (epics and user stories).
- **AP246 Storyboard** – A sequence of user interactions with the application or web site through graphics and narration.
- **PL146 Prototype** – A prototype is built to verify the feasibility of new, complex, or high-risk technical considerations.
- **AP215 Packaged - Fit/Gap Analysis** – Document where high level gaps for key business processes are documented. **

Differences with Traditional Waterfall Methods

Waterfall	Agile
Plan is an one-time activity	Planning in different levels -Initial up-front planning at the start -'Just-In-Time' planning in Sprint 0 and later Sprints
Project Manager plans for the team	Team is empowered and participates in planning
Detailed requirements up-front	High-Level requirements only
Significant requirements documentation	Requirements are captured in the form of User Stories or Use Cases
Limited client collaboration after requirements definition	Collaborative analysis with client throughout the project lifecycle

Product Backlog

- **Prioritized (MoSCoW) list** of **requirements/features** that provide business value for the customer

  MoSCoW stands for:

  - Must have (or Minimum Usable Subset)
  - Should have
  - Could have
  - Won't have (but Would like in future)

- Usually a combination of Epics and User stories
- **Product Owner owns and prioritizes** the product backlog
- Items can be **added** to the product backlog **at any time** during the project but outside the Sprints
- The **Product Backlog** consists of three types of items:
  - Product functionality
  - Non-functional requirements
  - Environmental requirements

User Story	Priority	Size in Story Points
User Story 1	Must Have	5 points
User Story 2	Should Have	8 points
User Story 3	Could Have	13 points
User Story 4	Should Have	5 points
User Story 5	Should Have	8 points
User Story 6	Could Have	3 points
User Story 7	Should Have	1 points
User Story 8	Must Have	5 points
User Story 9	Must Have	8 points
User Story 10	Could Have	13 points
User Story 11	Should Have	5 points
User Story 12	Must Have	8 points
User Story 13	Could Have	3 points
User Story 14	Must Have	13 points

Product Backlog for Course - Prioritized

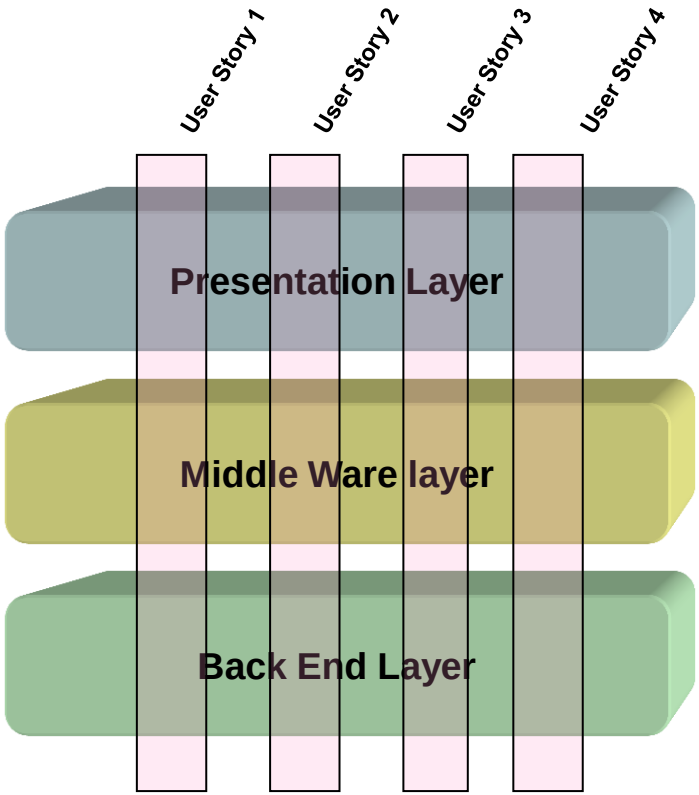
Story Title	Description	Priority
Sprint Application & Management	As a trainee, I want to understand various activities and roles performed during Sprint application & Management so that I can contribute more to my project.	1
Product Backlog Refinement	As a trainee, I want to understand how Product Backlog Items will be refined so that I can have more detailed requirements.	2
PM & Deploy Release in ADM for DA	As a Project Manager, I want to understand my role and deployment considerations in an agile project so that I can effectively manage upcoming agile projects.	3
Estimation in Agile Projects	As a team member, I want to understand how story points can be mapped to effort so that I can estimate better during Release/sprint planning.	4
Agile Tools	As a trainee, I should understand various Agile industry tools used so that I can use them for better project management, tracking project's progress etc.	5
Change Control & Defect Management	As an agile team, we should understand how to manage changes in agile projects so that we can effectively follow the practice of "sustainable pace".	6
Agile Metrics	As a trainee, I want to understand various agile metrics that can be measured to track the health of my project.	7

Product Backlog for Course – Prioritized

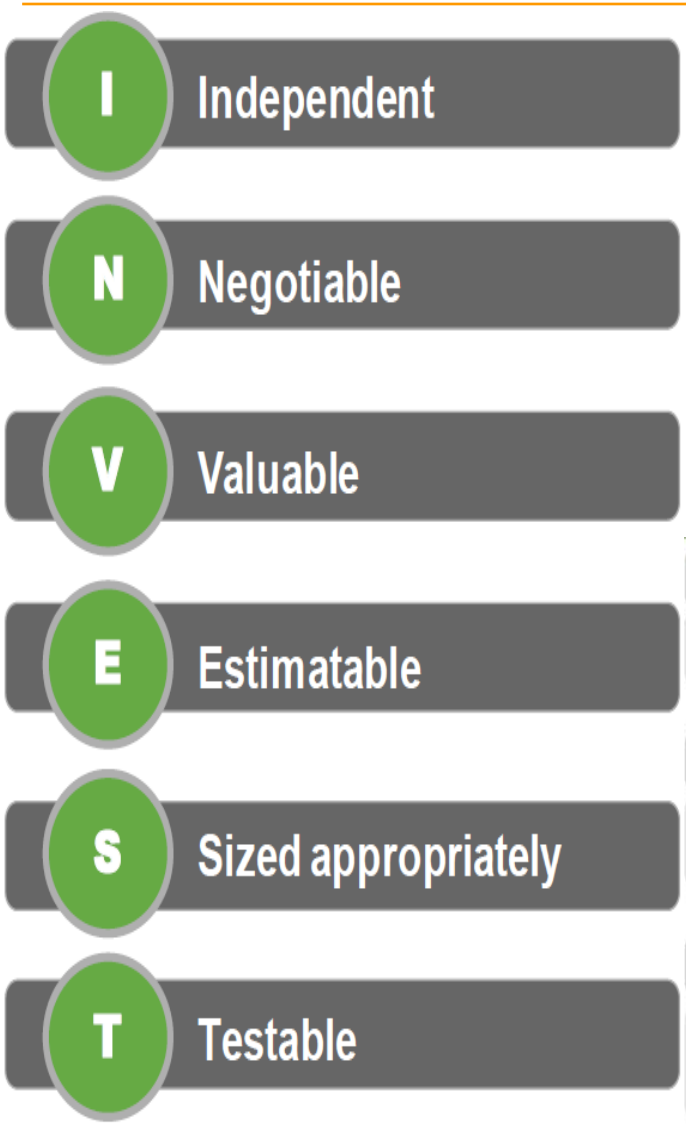
Story Title	Description	Priority
Scaling & Distributed Delivery	As project manager, I want to understand the key considerations & distributed delivery models available in scaling up agile to larger projects so that I can better manage them while dealing with such larger projects.	8
Project Experience / Case Study	As a trainee, I want to see the execution details of a real time project in Accenture to strengthen my agile concepts.	9
Agile in Maintenance Projects	As a trainee, I want to understand the usage of Kanban Board to effectively visualize the progress of the project.	10
Introduction to DevOps	As an Ops team member, I want to understand how development and operations team collaborate along with other teams to optimize the whole IT process.	11
Agile at Accenture Links	As a trainee, I should know various agile links available so that I can visit them to get information on trainings, certifications, tools and participate in blogs/discussions.	12

What is a User Story?

- User Story is a description of a feature or a requirement that is valuable to the user of the system
- Should be small enough so that team can implement it within a single Sprint
- User Story Template :
  - ***As a <role>, I want this <goal>, so that <reason>***
  - ***Acceptance Criteria :***
  - *Priority :*
  - *Estimate :*



What Makes a Good Story? - INVEST



I Independent

- Dependencies lead to problems estimating and prioritizing
- Can ideally select a Story to work on without pulling in 18 other stories

N Negotiable

- Stories are not contracts
- Leave or imply some flexibility

V Valuable

- To users or customers, not developers
- Re-write developer stories to reflect value to users or customers

E Estimatable

Because plans are based on User Stories, we need to be able to estimate them

S Sized Appropriately

- Small enough to complete in one sprint if you're about to work on it
- Bigger if further off on the horizon

T Testable

- **Testable so that you have an easy, binary way of knowing whether a story is finished – "Definition of Done"**
- Done or not done: no "partially finished" or "done except"

User Story Template – Front of the Card

Story Title :

Release & Sprint # :

As a <**User of the Application**>, I want to < **Required Functionality** > so that, <**Reason for the functionality**>.

- **Priority** : Must Have/Should Have/Could Have/Wont Have
- **Estimate** : X Story Points
- **Business Value (Optional)** :
- **Owner :**

Acceptance Criteria:

……..

……..

User Story Template – Back of the Card

Dependent User Stories:

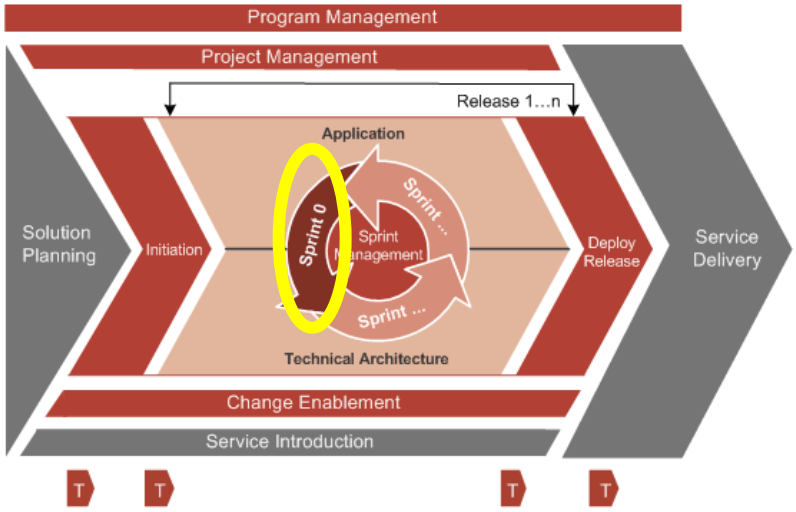
Technical Risks :

User Story Test / Design Considerations:

Activity 2 – Writing User Stories

- The aim of this activity is to:
  - Understand how to effectively write User Stories for an Agile project

Sprint 0 - Overview



- A ‘warm-up’ or preparatory Sprint in a release to start up the project effectively
- Performed at the beginning of a release cycle
- Common to both Application and Technical architecture work stream

Activities in Sprint 0

- Create Release plan
- Analyze the requirements
- Create, update, estimate and prioritize the requirements
- Create Design Framework

https://methodology.accenture.com/dist%5Fagile/#publish.dist_agile/disciplines/Sprint%200_9BD330A1.html

Roles

Product Owner

- Prioritizes the requirements/User Stories based on business value and risks
- Identifies the User Stories/Use Cases and the Sprints they will be built in
- Defines the goals of the Release and Sprints

Scrum Master & Team – Assists the Product Owner

- Plans for each release and identifies the scope of each release
- Creates detailed estimates for each release
- Performs detailed elaboration for application and TA requirements for the release and performs database design for the release

Key Work Products

- **IE219 Product Backlog**
  - User Stories/Use Cases for the release are identified from the Product Backlog.
  - The Sprint in which the User Story/Use Case will be built is identified.
- **MG110 Management Plan** - Consolidated Management Plan, Risk Management Plan, Quality Management Plan, Measurement Plan and Configuration Management Plan
- **TE580 Test Plan** - Composite of Test Scenarios, Test Conditions and Expected results and a Test Cycle Control sheet
- **IE212 Sprint Estimate** - Generates estimate efforts for technology specific inventory items. To be used for estimation for the release
- **PL101 Requirements Traceability** - Bi-directional traceability between the requirements and the various analysis, build and test components
- **AP215 Packaged - Fit/Gap Analysis** - Documents all potential gaps

Story Points

- **Story Points** are a unit of measure for expressing the overall **size** of a User Story, feature or other piece of work
- The number of Story Points associated with a User Story represents the overall size of the story
- When estimating with Story points, we assign a point value to each item
- We are estimating the **SIZE** of each item on the list

  SIZE = Effort x Complexity x Uncertainty

  - Effort = How long will it take to do
  - Complexity = How difficult will it be to do
  - Uncertainty = How unsure are we of the requirements or implementation

Planning Poker

Steps

Each team member gets a deck of cards	Customer/ Product Owner reads a story	Team members select cards	Cards are turned over	Discuss differences	Re-estimate
Each team member is given a deck of cards. Each card has a valid estimate written on it based on modified Fibonacci series i.e. ½, 1, 2, 3, 5, 8, 13, 20, 40, 60 100 ?	Customer/ Product Owner reads a story and it's discussed briefly	Each developer selects a card that's his or her estimate	Cards are turned over so all can see them	Discuss differences (especially outliers)	Re-estimate until estimates converge

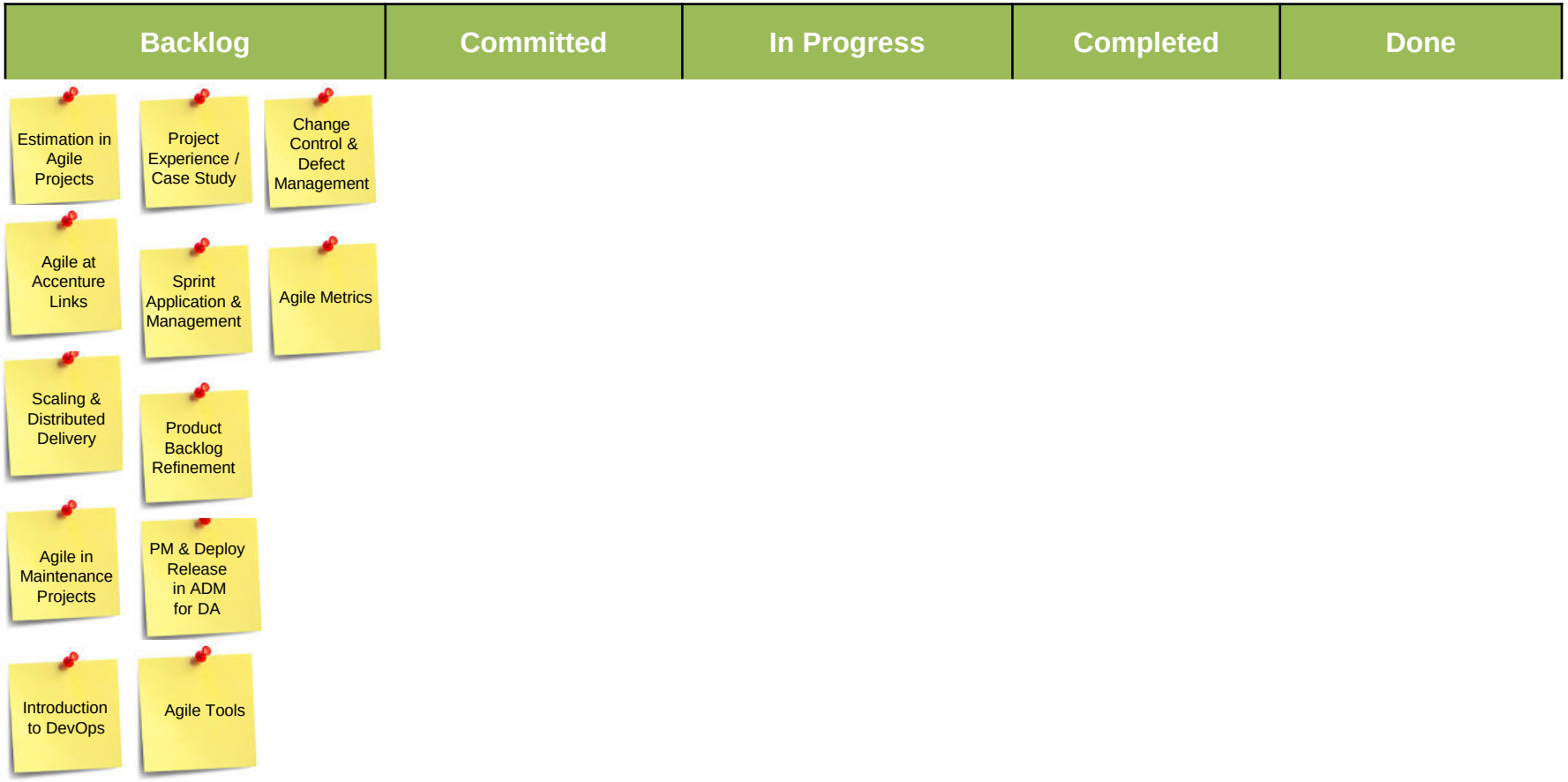
Activity 3 – Planning Poker

- The aim of this activity is to:
  - Understand the importance of using a relative scale for planning poker estimation

Release Plan for Course

Release Plan (Sprint 0)	
Day 1 – Sprint 1 (3 Hours 55 Min)	**Day 2 – Sprint 2 (3 Hours 55 Min)**
Sprint Planning	Sprint Planning
Sprint Application & Management	Agile Metrics
Product Backlog Refinement	Scaling & Distributed Delivery
PM & Deploy Release in ADM for DA	Daily Stand Up
Daily Stand Up	Project Experience / Case Study
Agile Tools	Agile in Maintenance Projects
Estimation in Agile Projects	Introduction to DevOps
Change Control & Defect Management	Agile at Accenture Links
Sprint Review (Quiz)	
Retrospective	Sprint Review (Quiz)
	Retrospective
Deploy Release / Go Live	
Agile Lifecycle Events	
RTC Tool Usage	
School Close	

Story Wall



Questions & Answers