Agenda

1. Project Execution Traditional Approach
2. Agile framework
3. Scrum
4. Roles
5. Ceremonies
6. Artifacts
Project Execution Traditional Approach

What comes to your mind???

- Waterfall
- Hierarchy
- Delegation
- Rigid Processes
- Work Break Down
- Phased Development
- Sign-offs, & Hand-offs
- Specialists (Team Members and Teams)
- Heavy Documentation
- Big Bang Release
Project Execution Traditional Approach

- Big Bang / Code & Fix
- Waterfall 1950 - 1970
- Spiral Model 1980 - 1988
  - Incremental
  - Iterative
- RAD (Rapid Application Development) 1990
- Agile 2001
Typical Waterfall

- Analyze
- Design
- Build & Development
- Testing
- Deployment
- Warranty Support
Agile framework & principles

Origin -

The concept of iterative and incremental approach to development started away back from 1950s.

- In 1950s – NASA and The United States Department of Defence (DoD) have used the incremental and iterative development (IID) approach.

- In 1960s – the concept of Evolutionary project management (Evo) recommended a one- to two-week iterations, delivery of product in each iteration.

- In 1986 – The “rugby approach” was discussed with dedicated, self-organizing teams. Here a team tries to go the distance as a unit, passing the ball back and forth—may better serve today’s competitive requirements.”
Agile framework & principles

Rugby Approach –

(Photo from www.zimbio.com – Northland club Rugby)
Agile framework & principles

What is Agile –

Agile software development is a group of software development methods based on

• **Iterative** and **Incremental** development, where requirements and solutions evolve through collaboration between **self-organizing, cross-functional teams**.

• It promotes **adaptive planning**, **evolutionary** development and delivery, a **time-boxed iterative** approach, and encourages rapid and flexible response to change.

• It is a conceptual framework that promotes foreseen interactions throughout the development cycle

Agile framework & principles

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

The above states that while there is a value in the items on the right, we value the items on the left more.
Agile Manifesto –

**Individuals and interactions over processes and tools** –

- Promotes empowered, self-managing, cross functional teams
- Advocates members to collaborate in person to solve a mutual problem
- Tools support cannot replace individual interactions
Agile framework & principles

Agile Manifesto –

**Working software over comprehensive documentation**

- Status / progress should be measured on actual working product and not by percentage completion of the functional milestones
- Agile encourages teams for face-to-face communication over documentation which is simpler, faster, and far more reliable
- Documents usually gets out-dated as system design keeps on changing
Agile framework & principles

Agile Manifesto –

Customer collaboration over contract negotiation

- Customers become an integral part of the development process and not only during contract negotiation and defining specification
- Better way to deliver value instead of a carefully worded contract
Agile framework & principles

Agile Manifesto –

Responding to change over following a plan

- Agile plans follow more of a rolling wave approach using top-down planning
- Change management is easier when the organization and the customer share a clear understanding of the project’s status
- In plan-driven environments, all requirements are specified up front, broken down to the task level and estimated which is not the case in Agile developments
- Business needs and priorities can shift in the months or even years it can take for a large system to be fully built
12 Agile Principles –

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

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

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

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

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

6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
12 Agile Principles –

7. Working software is the primary measure of progress.

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

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

10. Simplicity—the art of maximizing the amount of work not done—is essential

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

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly
Agile / Traditional Approach

- Analysis
- Design
- Code
- Test
- Deploy

Road Map
- Release
- Iteration
- Day
During an agile project, planning occurs at three levels:

- **✓ Release planning:** Release plans contain a release schedule for a specific set of features. The product owner creates a release plan at the start of each release.

- **✓ Iteration planning:** Team members gather at the beginning of the iteration (referred to as a *sprint* in the Scrum methodology) to identify the work to be done during that iteration. This is referred to as *self-organization*.

- **✓ Daily planning:** Development teams begin each day with standup meetings to plan the day. These meetings are generally 5 to 15 minutes long.
Agile Methods

Some of the Agile Methods are –

- Scrum
- Extreme programming (XP)
- Rational Unified Process (RUP)
- Clear Crystal
- Dynamic Systems Development Method (DSDM)
- Kanban
- Many more...
What is Scrum?

• Scrum is a simple yet incredibly powerful set of principles and practices that help teams deliver products in short cycles, enabling fast feedback, continual improvement, and rapid adaptation to change. As the leading Agile development framework, Scrum has predominantly been used for software development, but it is also proving to be effective in efforts far beyond -- **Source: Scrum Alliance**

• Scrum is an iterative and incremental agile software development framework for managing software projects and product or application development. Its focus is on "a flexible, holistic product development strategy where a development team works as a unit to reach a common goal" as opposed to a "traditional, sequential approach". – **Source: Wikipedia**

• In short:
  
  ✓ Scrum is an agile process that focuses on delivering the highest business value in the shortest time thereby rapidly and repeatedly inspect actual working software.
  
  ✓ The business sets the priorities. Scrum teams self-organize to determine the best way to deliver the highest priority features. Time-boxed for two to four weeks, anyone can see real working software and decide to release it as is or continue to enhance it for another sprint.
Scrum focuses on the following values to lay a strong foundation to emphasise on teamwork and its continuous improvement:

**Focus:** Because we focus on only a few things at a time, we work well together and produce excellent work. We deliver valuable items sooner.

**Courage:** Because we are not alone, we feel supported and have more resources at our disposal. This gives us the courage to undertake greater challenges.

**Openness:** As we work together, we practice expressing how we're doing and what's in our way. We learn that it is good to express concerns so that they can be addressed.

**Commitment:** Because we have great control over our own destiny, we become more committed to success.

**Respect:** As we work together, sharing successes and failures, we come to respect each other and to help each other become worthy of respect.
The Agile: Scrum Framework at a glance

Inputs from Executives, Team, Stakeholders, Customers, Users

Product Owner

The Team

Sprint Backlog

Product Backlog

Sprint Planning Meeting

Task Breakout

Sprint Backlog

1-4 Week Sprint

Sprint end date and team deliverable do not change

Burndown/up Charts

Scrum Master

Every 24 Hours

Daily Scrum Meeting

Sprint Review

Finished Work

Sprint Retrospective

Picture source: www.agileforall.com
Scrum

Framework –

- With inputs from business stakeholders, the Product Owner creates a prioritized business wish list called a product backlog
- Based on priorities set by product owner, the Scrum Team takes up a small chunk from the top of the product backlog, in sprint backlog, and decides how to implement those pieces
- Scrum team has a time-boxed period — a Sprint (usually two to four weeks) — to complete its work, but it meets each day to assess its progress (daily Scrum)
- The ScrumMaster keeps the team focused on its goal throughout.
- At sprint end, the work outcome is potentially shippable: ready to be handed over to customer, put on a store shelf, or show to a stakeholder
- A sprint review and retrospective concludes a sprint
- The next sprint begins with another chunk of the product backlog
Scrum

Framework –

Roles
- Product Owner
- Scrum Master
- Team

Ceremonies
- Sprint Planning
- Daily Scrum Meeting
- Sprint Review
- Sprint Retrospective

Artifacts
- Product Backlog
- Sprint Backlog
- Sprint Burndown Chart
**Scrum**

Framework –

### Roles
- Product Owner
- Scrum Master
- Team

### Ceremonies
- Sprint Planning
- Daily Scrum Meeting
- Sprint Review
- Sprint Retrospective

### Artifacts
- Product Backlog
- Sprint Backlog
- Sprint Burndown Chart
Scrum Roles

Product Owner – Responsibilities

• Define the features of the product
• Work on a shared vision as he represents the business stakeholders
• Manage and Prioritize features according to market value in Product Backlog
• Accept / reject the work at the end of each iteration
• Managing the release plan
• Responsible for the profitability of the product (ROI)
• Decide on release date and content
Scrum Roles

Scrum Master – Responsibilities

- Empowers and shepherds the Scrum team
- Represents management to the project
- Keeps the process moving by enacting Scrum values and practices
- Removes impediments
- Ensure that the team is fully functional and productive
- Enable close cooperation across all roles and functions
- Shield the team from external interferences
- Socialising Scrum to the greater organisation
Scrum Roles

Team – Responsibilities

• Typically a team of 5-9 people
• They are usually cross-functional: Programmer, QA, UX designers, Analyst, etc.
• Estimate size of backlog items
• Committing to increments of deliverable software and delivering it
• Tracks own progress
• Is self-organising—but accountable to the Product Owner for delivering as promised
Scrum

Framework –

Roles
- Product Owner
- Scrum Master
- Team

Ceremonies
- Sprint Planning
- Daily Scrum Meeting
- Sprint Review
- Sprint Retrospective

Artifacts
- Product Backlog
- Sprint Backlog
- Sprint Burndown Chart
Scrum Artifacts

Artifact stages –
Product Backlog –

The product backlog is simply a list of work items that need to be done over time. Items may be added to the backlog by anyone, but only the Product Owner has the right to determine the order in which they will be executed by the team and he negotiates this with stakeholders and the team. The Product Backlog is a living document and requires constant grooming to keep it current and useful. Many new items will be added over time; existing items are disaggregated to multiple, smaller items; some items may be removed on realising that a desired feature is no longer needed.

**Good Product Backlog contains:**

Good product backlogs should be **DEEP** (Coined by Roman Pichler and Mike)

- Detailed appropriately
- Emergent
- Estimated
- Prioritized
## Scrum Artifacts

### Contents of Product Backlog –

<table>
<thead>
<tr>
<th>Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature</td>
<td>As a recruiter I want to search candidates using keywords so that I can find the suitable job seeker</td>
</tr>
<tr>
<td>Change</td>
<td>As a job seeker I want default ordering job search results to be by freshness rather than location so that its easier to see latest jobs first</td>
</tr>
<tr>
<td>Defects</td>
<td>Please fix defect #148 so that space characters in search wont crash the system</td>
</tr>
<tr>
<td>Technical Improvement</td>
<td>Upgrade to the latest version of Internet Information Server</td>
</tr>
<tr>
<td>Knowledge Acquisition</td>
<td>Create integration prototype using MSSQL Server and Salesforce.com and run performance test to check real time integration speed and reports.</td>
</tr>
</tbody>
</table>
Sprint Backlog –

The Sprint backlog is more of like a task board, which is the physical representation of the list of work that the Scrum team have committed to do during the current sprint. It tells the whole team and anyone else what work they have planned or the sprint and their current status.

**In short:**

- The Sprint Backlog defines the work team will perform to turn selected Product Backlog items into a “Done” Increment
- The list emerges during the Sprint.
- Every team member takes responsibility / ownership of each task in progress task
- Each Tasks has information about estimated amount of work remaining on the task on any given day during the Sprint
Scrum Artifacts

Sprint Backlog Management –

- Individuals take up work of their own choosing. Work is never assigned
- Estimated work remaining is updated on daily basis
- Any team member can add, delete or change the sprint backlog
- If work is unclear, define a sprint backlog item with a larger amount of time and break it down later
- Update work remaining as more becomes known

<table>
<thead>
<tr>
<th>User Story</th>
<th>Tasks</th>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>Day 4</th>
<th>Day 5</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>As a member, I can read profiles of other members so that I can find someone to date.</td>
<td>Code the ...</td>
<td>8</td>
<td>4</td>
<td>8</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Design the ...</td>
<td>16</td>
<td>12</td>
<td>10</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Meet with Mary about ...</td>
<td>8</td>
<td>16</td>
<td>16</td>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Design the UI</td>
<td>12</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Automate tests ...</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Code the other ...</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>As a member, I can update my billing information.</td>
<td>Update security tests</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Design a solution to ...</td>
<td>12</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Write test plan</td>
<td>8</td>
<td>8</td>
<td>4</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Automate tests ...</td>
<td>12</td>
<td>12</td>
<td>10</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Code the ...</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Picture source: www.mountaingoatsoftware.com
Scrum – Done

Definition of Done –

- Everyone must understand what done means
- This varies significantly per Scrum Team, members must have a shared understanding of what it means for work to be complete, to ensure transparency
- This guides Development Team in knowing how many Product Backlog items it can select during a Sprint Planning Meeting.
- The purpose of each Sprint is to deliver Increments of potentially releasable functionality that adhere to Definition of “Done.”
- Definition of Done may change during the project
Sprint Burndown Chart –

The sprint burndown chart helps the Scrum team to monitor its progress and works as an indicator of whether it will meet its commitment at the end of the sprint. It requires teams to estimate the duration of each task in hours and on a daily basis to chart the total remaining hours for all uncompleted tasks.

**Helps to:**

- Estimate new tasks and re-estimating in-progress tasks requires effort
- Re-estimate inaccurate tasks
- Completion of tasks delivers no value; only completed stories (features) deliver value
Scrum Artifacts

Sprint Burndown Chart Sample –

User Story

Definition –

A user story describes functionality that will be valuable to either a user or purchaser of a system or software.

A User Story Is

- An agreements between customers and team members to discuss detailed requirements during an iteration
- Emphasize verbal rather than written communication.
- Right size for planning

A User Story Is Not

- A Requirement document, requirement needs to be captured by discussion, if the discussion is not possible then should be augmented by documentation
User Story

• Your user story needs to have, at a minimum, the following parts

✓ **Title**: *<a name for the user story>*
✓ **As a** *<user or persona>*
✓ **I want to** *<take this action>*
✓ **So that** *<I get this benefit>*

• The story should also include validation steps. That step is worded as follows:

✓ **When I** *<take this action>, this happens* *<description of action>*

<table>
<thead>
<tr>
<th>Title</th>
<th>Value</th>
<th>Author</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer money between accounts</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title</th>
<th>Value</th>
<th>Author</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carol,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I want to review fund levels in my accounts and transfer funds between accounts so that I can complete the transfer and see the new balances in the relevant accounts.</td>
<td></td>
<td>Jennifer</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title</th>
<th>Value</th>
<th>Author</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jennifer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;personal/user&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I want to &lt;action&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>so that &lt;benefit&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Scrum

Framework –

Roles
- Product Owner
- Scrum Master
- Team

Ceremonies
- Sprint Planning
- Daily Scrum Meeting
- Sprint Review
- Sprint Retrospective

Artifacts
- Product Backlog
- Sprint Backlog
- Sprint Burndown Chart
Scrum Ceremonies

Sprint Planning –

• The Team and the Product Owner collaborate to help the Team determine how much Product Backlog it can turn into functionality during the upcoming Sprint.

• The Team create plans (Sprint Backlog) by identifying tasks for converting selected Product Backlog into functionality

• Team selects items from the product backlog that they can commit to complete within a Sprint

• Sprint backlog is created
  ✓ Tasks are identified and each is estimated (1-16 hours)
  ✓ Collaboratively, not done alone by the Scrum Master

• High-level design is considered
**Scrum Ceremonies**

**Sprint Planning** –

**Inputs**
- Product Backlog
- Team Capacity
- Business Conditions
- Current Product
- Technology

**Sprint Planning Meeting**

**Sprint prioritization**
- Analyse and evaluate product backlog
- Select sprint goal

**Sprint planning**
- Decide how to achieve sprint goal (design)
- Create sprint backlog (tasks) from product backlog items (user stories / features)
- Estimate sprint backlog in hours

**Output**
- Sprint Goal
- Sprint Backlog
Scrum Ceremonies

Sprint Planning –

Prioritizes Product Backlog

Develops Sprint Backlog

Facilitates
Daily Scrum Meeting –

The team meets daily to communicate and synchronise its work. Since the team is collaborating, it’s essential to ensure that continued project progress and avoiding work blockages. The team continuously assess its own progress towards achieving its sprint goal and NOT report progress to the ScrumMaster or Product Owner or anyone else.

<table>
<thead>
<tr>
<th>Helps to:</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Fine-grain coordination</td>
<td>• Daily 15-minutes Stand-up</td>
</tr>
<tr>
<td>• Daily commitment</td>
<td>• Not for problem solving</td>
</tr>
<tr>
<td>• Raising impediments</td>
<td>✓ Only team members, Scrum Master,</td>
</tr>
<tr>
<td>• Peer pressure</td>
<td>✓ Product Owner can talk</td>
</tr>
<tr>
<td>• Access Progress towards sprint goal</td>
<td>Comprises of:</td>
</tr>
<tr>
<td>• Helps avoid other unnecessary meetings</td>
<td>✓ Story</td>
</tr>
<tr>
<td></td>
<td>✓ To Do</td>
</tr>
<tr>
<td></td>
<td>✓ WIP</td>
</tr>
<tr>
<td></td>
<td>✓ Done</td>
</tr>
<tr>
<td></td>
<td>✓ Impediment</td>
</tr>
</tbody>
</table>
Scrum Ceremonies

Daily Scrum Meeting –

1. What have you done yesterday?
2. What are you going to do today?
3. Are there any impediments in your way?
Scrum Ceremonies

Scrum Board
IF you add a task(s) after screen, put here.
Sprint Review –

The focus of the sprint review is the product the team is building. While it does include a demonstration of the new features the team has completed during the sprint, its primary purpose is to inspect what the team has delivered and gather feedback from the attendees to adapt the plan for the succeeding sprint.

**Helps to:**

- Team presents to product owner, stakeholders and scrum master what it accomplished during the sprint
- Typically takes the form of a demo of new features
- It’s usually Informal
  - 2-hour prep time rule
  - No slides
- Whole team participates
- Audience – Invite the world!!
Scrum Ceremonies

Sprint Review –

Provides feedback

Presents the new development

Facilitates
Sprint Retrospective –

The sprint retrospective is the final meeting of the sprint. It follows immediately after the sprint review and should never be omitted. The retrospective is focused on the process—the way in which the Scrum team is working together, including their technical skills and the software development practices and tools they are using.

**Parameters:**

- Periodically take a look at what is and is not working
- Done after every sprint
- Participants
  - Scrum Master
  - Team
  - Product owner (Optional)
Scrum Ceremonies

Sprint Retrospective –

1. Start doing
2. Continue doing
3. Stop doing
Sprints –

• A series of “Sprints” depicts a Scrum project progress

• It holds the same meaning to that of Extreme Programming iterations

• Sprints are usually time-boxed two to four calendar weeks

• Since time-boxed, the duration is constant which leads to a better rhythm

• Within a Sprint – Product is designed, coded and tested

Picture source: www.kaeru.se
Questions?

Feel Free to Contact: pritennayak@gmail.com