Waterfall

? Traditional approach used for over 40 years

? Requirements must be defined at the start

? Little / no alternations

? Sequential - Complete 1 task and then the next

? Used in large scale SW development where thorough planning and predictability is required

Pros

? Extensive planning, this thoroughness often results in more accurate timelines and budgets

Cons

? Difficult to apply changes or modify / correct previous steps (water can＊t run backwards), need to be proactive in anticipating problems

Agile

? Focuses on adapting to changing situations

? Depends on constant and regular feedback

? Focuses on iterative outcomes delivering value as quickly as possible & collaboratively

? Involvement & ownership across the team 每 self select the work

? Small manageable actions and activities

? Customer focus over formalised sign-offs

Pros

? Retains flexibility while continually producing outcomes 每 less rework ? Greater communication & engagement

Cons

? Difficult to do without an experienced Scrum Master ? Large projects co-location a problem ? Difficult to contract suppliers

Success Factors

1. Executive Sponsorship 15%

2. Emotional Maturity 15%

3. User Involvement 15%

4. Optimisation Statement of Requirements 15%

analyze Case Study (business needs) ->

analyze constraints (scope, time, cost) ->

develop Business Case (cost verses benefit) ->

develop Project Charter (stakeholder analysis)

Step 1: Screen candidate projects & seek approval

Step 2: Business Case: document organizational buy-in

Step 3: Manage Project

– Project Management Plan

Step 4: Build & Deploy via SDLC

Step 5: Deliver Benefits Final Project Report Lessons Learnt

A typical PMP consists of all / or most of the following categories.

• Project Governance

– Roles and Responsibilities

– Mandatory Project Planning / Key Additional Activities

o Schedule

o Risk Management

o Cost Estimation

o Quality Assurance

o Configuration Management (Change Management)

The PMP is a large multi-page document that takes time to prepare, review and complete. Multiple people (subject experts) are involved and prepare the specific details. The Project Manager coordinates all items and has ultimate accountability for the quality and final outcome.

Activities in SDLC:

• Requirements gathering

• Systems / Architectural Design

• Implementation / coding

• Integration

• Testing

• Delivery and Release - Deployment

• Maintenance

There are many SDLCs around with organisations typically favouring a blend of Formal and Agile approaches.

Formal Processes

• Waterfall

• Incremental

• V-Model

Agile Processes

• Extreme Programming

• Scrum

• Kanban

Waterfall

Requirements->Design->implementation->testing->deployment

Advantages

• Simple and easy to understand and use

• Easy to manage due to the rigidity of the model

• Phases are processed and completed one at a time

• Documentation available at the end of each phase

• Works well for projects where requirements are very well understood and remain stable

Disadvantages

• Difficult to accommodate change after the process in underway

• One phase must be completed before moving on to the next

• Unclear requirements lead to confusion

• Clients approval is in the final stage

• Difficult to integrate risk management due to uncertainty

Incremental

In incremental model the whole requirement is divided into various releases. Multiple cycles take place, making the life cycle a multi-waterfall cycle. Cycles are divided up into smaller, more easily managed modules.

Advantages

• Each release delivers an operational product

• Less costly to change the scope/requirements

• Customers can respond to each build

• Initial product delivery is faster

• Customers get important functionality early

• Easier to test and debug during smaller iterations

Disadvantages

• More resources may be required

• More management attention is required

• Defining / partitioning the increments is difficult and often not clear

• Each phase of an iteration is rigid with no overlaps

• Problems may occur at the time of final integration

V-model

The V-Model is an extension of the waterfall model and is based on the association of a testing phase for each corresponding development stage. This means that for every single phase in the development cycle, there is a directly associated testing phase. This is a highly-disciplined model and the next phase starts only after completion of the previous phase

Advantages

• Simple and easy to understand and use

• Each phase has specific deliverables and well defined objectives and goals

• High chance of success over waterfall model due to the development of test plans early on during life cycle

• Works well for small projects when requirements are easily understood

Disadvantages

• Very rigid process like the waterfall model

• Little flexibility and adjusting scope is difficult and expensive

• Software developed during implementation phase, no early prototypes

• No clear path for problems found during testing

• Changes in later stage cause test documentation across all stages to be changed

Formal Models

Characteristics where “Formal” Models make sense to use

• Projects where the customer has a very clear view of what they want

• Projects that will require little or no change to requirements

• Software requirements are clearly defined and documented

• Software development technologies and tools are wellknown

• Large scale applications and systems developments

Kanban

• Signboard / Billboard: Work items are visualised to provide participants a view of progress and process, from start to finish usually via a Kanban board

ToDO->Doing->Done

Scrum

•Scrum is an agile process that allows us to focus on delivering the highest business value in the shortest time.

•It allows us to rapidly and repeatedly inspect actual working software (every two to four weeks).

•The business sets the priorities. Teams self-organise to determine the best way to deliver the highest priority features.

•Every two to four weeks, you can see real working software and decide to release it as is or continue to enhance it for another sprint.

Characteristics

• Self-organising teams

• Product progresses in a series of focused sprints

• Requirements are captured as items in a list of product backlog

• Scrum is one of the agile processes – the one most widely used, discussed and debated

• Time frame is contained to a manageable size (weeks or months)

see L2:page 48

Sprints

Requirements->Design->Code->Test

Roles:

Product owner, ScrumMaster, Teams

Ceremonies

Sprint planning, Sprint review, Sprint retrospective, Daily stand-ups

Artifacts

Product backlog, print backlog, burndown charts

Product Owner

• Defines the features of the product

• Decides on release date and content

• Is responsible for the Benefits / Profitability of the product (ROI)

• Prioritises features according to market value

• Adjusts features and priority every iteration, as needed

• Accepts or reject work results

Scrum Master

• Represents management to the project

• Responsible for enacting Scrum values and practices

• Removes impediments / road blocks

• Ensures that the team is fully functional and productive

• Enables close cooperation across all roles

• Shields the team from external interferences

• Is a member & active participant of the Scrum Team

The Team

• Typically 5 - 9 people

• Cross-functional: – Programmers, testers, user experience designers, business representatives etc.

• Members should be full-time – some exceptions

• Co-located (physically or virtually)

Sprint Planning

• Defines how to achieve sprint goal (design)

• Create sprint backlog (User Stories) from product backlog

• Estimate sprint backlog in team velocity and Story Points

• Product Owner priority guides the work

• Release Plan is created

• High-level design is considered

Daily Stand-up

• Parameters o Daily o 15-minutes o Stand-up

• Not for problem solving / Not a status meeting o Whole world is invited o Only team members, ScrumMaster, product owner, can talk

• Helps avoid other unnecessary meetings

• 3 key questions asked: 1. What did I do yesterday. 2. What will I do today. 3. What is in my way to get my work completed.

Sprint Reviews - Showcase

• Team presents what it accomplished during the sprint

• Typically takes the form of a demo of new features or underlying architecture

• Informal

• 2-hour prep time rule

• No slides

• Whole team participates

• Invite the world

Sprint Retrospective

• Periodically look at what is and isn’t working

• Typically 30 minutes

• Done after every sprint

• Whole team participates: o ScrumMaster and Team

• Possibly Product Owner, customers and others

• Discuss what to: o Start Doing, Stop Doing and Continue Doing

User Stories

• AUser Story is a requirement expressed from the perspective of an end-user / customer of the system

• User stories shift the focus from writing about requirements to talking about them

• User stories are short, simple descriptions of a feature told from the perspective of the customer who wants the new capability of the system. They follow a simple template:

– As a < type of user >, I want < some goal > so that < some reason >

• User stories are written at varying levels of detail.

• They can cover a large amounts of functionality such as this example from a desktop backup product:

– As a site visitor, I need to access all news on line

• Because this level of detail is too large for an agile team to complete in one iteration, it is sometimes split into smaller user stories before it is worked on

Story Points

• Story points are a unit of measure for expressing an estimate of the overall effort that will be required to fully implement a product backlog item or any other piece of work

• Story points help estimate how much work can be done in a sprint

• When estimating with story points, a value is assigned to each item. The raw values are unimportant, what matters are the relative values

• A story that is assigned a 2 should be twice as much as a story that is assigned a 1. It should also be two-thirds that is estimated as a 3 story point.

• Instead of assigning 1, 2 and 3, that team could assign 100, 200 and 300. Or 1 million, 2 million and 3 million. It is the ratios that matter, not the actual numbers

Product Backlog

• The requirements

• A list of all desired work on the project

• Ideally expressed such that each item has value to the users or customers of the product

• Product Backlog Feature level User Stories are selected for a Sprint by Product Owner

• Reprioritised at the start of each sprint

Sprint Backlog / User Story

• Scrum team decompose User Stories to a Low level User Stories during Sprint Planning

• The User Stories are used for a conversation between the SME and developer. Developer updates the User Stories with the tasks and hours estimates, "Just-In-Time“

• Remaining estimated items are updated daily

• Sprint Backlog is seldom altered

• User stories in the sprint are either completed 100% or not done

Burn Down Chart see L2: page 70

• A burn down chart is a graphical representation of work left to do versus time.

• The outstanding work (or backlog of user stories) is often on the vertical axis, with time along the horizontal.

• It is used to predict when all of the work will be completed

8.Understand advantages / disadvantages of Agile

Advantages

• Customer satisfaction by rapid, continuous delivery of usable software

• People and interactions are emphasised rather than process and tools

• Continuous attention to technical excellence, good design and quality

• Regular adaptation to changing circumstances

Disadvantages

• Difficult to assess the effort required at the beginning

• Can be very demanding (from traditional approaches) on users time

• Harder for new starters to integrate into the team

• Agile is a very different approach – It can be intense for the team

• Requires experienced resources (which are limited in today’s market)

9.Understand key questions that will help select which approach to use and some examples

There is no one right answer. The following questions can assist deciding:

• How Stable Are the Requirements?

• Do the end users need to collaborate?

• Is the Time Line Aggressive or Conservative

• What Is the Size of the Project

• Where Are the Project Teams Located

• What Are the Critical Resources?