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 well known

• 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?

Money, praise, promotion

Maslow Hierarchy of Needs: see L3: page 31

• Proposed by Abraham Maslow in 1943 and still widely used today

• A key tool used by managers in how individuals are motivated

• Focuses on a 5 tier model of human needs

• Describes humans are motivated to achieve certain needs

• Needs take precedence over others and the basic needs must be more or less met before higher needs

• Individual behaviour is multi-motivated and stimulated by more than one need

Hertzberg Two Factor Theory: see L3: page 32

• Proposed by Fredrick Hertzberg in 1959 and still widely used today

• Asked people to describe situations when they felt really good and really bad about their jobs

• There are a set of factors in the workplace that cause satisfaction

• And a separate set of factors that cause dissatisfaction

• Remedying the causes of dissatisfaction will not create satisfaction

• Adding the factors of job satisfaction eliminate job dissatisfaction

Power:

Referent: based on followers' identification and liking for the leader. a schoolteacher who is adored by her students has referent power

Expert: based on followers' perceptions of the leader's competence. a tour guide who is knowledgeable about a foreign country has expert power

Legitimate: associated with having status or formal job authority. a judge who administers sentences in the courtroom exhibits legitimate power

Reward: derived from having the capacity to provide rewards to others. a supervisor who gives rewards to employees who work hard to using reward power.

Coercive: derived from having the capacity to penalize or punish others. a coach who sits players on the bench for being late to practice is using coercive power.

Influence:

1. Authority: Legitimate hierarchical.

2. Assignment: Perceived ability to influence future work assignments.

3. Budget: Perceived ability to authorise use of funds.

4. Promotion:Ability to improve workers position

5. Money:Ability to increase a workers pay & benefits

6. Penalty:Ability to cause punishment.

7. Work Challenges:Ability to assign work to individuals.

8. Expertise: Perceived special knowledge that others deem / think is important.

9. Friendship:Ability to establish friendly personal relationships.

Manage yourself

1. They Take Initiative. [“Be Proactive”]

2. They Focus on Goals. [“Begin with the End in Mind”]

3. They Set Priorities. [“Put First Things First”]

Lead others

4. They Only Win When Others Win. [“Think Win/Win”]

5. They Communicate. [“Seek First to Understand, Then to Be Understood”]

6. They Cooperate. [“Synergise”]

Unleash protential

7. They Reflect on and Repair Their Deficiencies. [“Sharpen The Saw”]

Why Teams

1.Few individuals possess all the knowledge, skills, and abilities needed to accomplish all tasks.

2.Complementary teamwork skills are one of the most commonly required skills in the work environment.

3.Substantial benefits to the organisation and to the team members.

4.Shared accountability increases likelihood of success.

Benefits

1. Enhanced Opportunities: Individuals & organisation.

2. Greater Productivity: Leverage the strengths and skills of the collective group.

3. Increased Ownership & Accountability: Multiple people collectively owning the activity and the outcome.

4. More Creativity and Innovation: Individuals build upon one another’s ideas with solutions going beyond one person’s vision of what’s possible.

5. Greater Joy and Satisfaction Among Team Members: A space for people to socialise, connect and be part of something bigger.

6. Broader Perspective: Ability to leverage the collective perspective of all team members.

7. Increased Representation: Involvement of multiple stakeholders groups and their input.

8. Increased Equality: Individuals across all levels can more freely offer their ideas, knowledge and concerns.

9. More Dialogue: Teams offer a site where people can voice their feelings, disagreements, opinions and ideas.

forming->storming->norming->performing->adjourning

Positive signs

• Clear communication

• Regular brainstorming with all members participating

• Consensus among team members

• Problem solving done by the group

• Commitment to the project outcomes and the other team members

• Regular team meetings are effective and inclusive

• Timely hand off from team members to others or early advise if this won’t happen

• Positive, supportive working relationships

Not so Positive signs

• Lack of communication

• No clear roles and responsibilities

• Work is “thrown over the wall”, with lack of concern for timelines or work quality

• Team members work alone, rarely sharing information and offering assistance

• Blame for what goes wrong, no one accepts responsibility

• Lack of support for others

• Frequently absent impacting time and creating additional work for other

Advantages

• Provides a larger pool of ideas – creative & collective problem solving

• Interaction enhances the knowledge of the whole team

• Individuals working together can stimulate performance, motivation and output

• Provides continuity across the tasks if people leave

• Increased ownership of the overall outcome & not just the individual component

Disadvantages

• It take time, effort and great skill to effectively manage

• Some individuals find it difficult and may become overshadowed / overwhelmed

• Unequal involvement - Some people may not pull their weight

• One person can demoralise the whole team

• Social loafing

• Group think

Internal Stakeholders External Stakeholders

Shareholders Customers

Employees Suppliers

Board Members Governments

Business Managers Unions

Project Manager Public [Local Communities or General Public]

Management Other Related Institutions [e.g. Financial]

Project Team Competitors

Levels of Stakeholder Engagement

• Unaware: Unaware of the project and its potential impacts on them

• Resistant:Aware of the project yet resistant to change

• Neutral:Aware of the project yet neither supportive nor resistant

• Supportive:Aware of the project and supportive of change

• Champion / Leading:Aware of the project and drives change

The stakeholder management plan can include:

• Current and desired engagement levels

• Interrelationships between stakeholders

• Communication requirements

• Potential management strategies for each stakeholder

• Methods for updating the stakeholder management plan

Stakeholder Analysis includes:

• Names and Organisations of Key Stakeholders

• Their Role on the Project

• Unique Facts about Each Stakeholder

• Level of Interest in the Project

• Influence on the Project

• Suggestions and Strategies for Managing Relationships with each Stakeholder

Types of Outsourcing:

1. Onshoring:

• Relocating activities inside national borders to access targeted benefits.

2. Nearshoring:

• Activities relocated to another country with close proximity e.g. New Zealand, Indonesia.

3. Offshoring:

• Activities relocated to another country irrelevant of geographical location and time zones.

Accessing a broader skills base at a lower cost

Pros

• Reduces costs

• Access to difficult to find capabilities & skills

• Time savings – 24/7 based activities

• Freeing scares internal resources to focus on core business activities

• Leverage best practice

• Access to better Technology

• Lower training costs in high turn over jobs

• Flexibility – Ramp up and down

• Increased Accountability - Contracts

• Risk mitigation – Access established and proven approaches e.g. Agile, Project Management etc

Cons

• Loss of control

• Process / supply chain fragmentation

• Security issues

• Employees feel threatened

• Additional effort and cost to engage and manage

• Lower quality work / work to contract

• Time zone, cultural & language challenges

• Location stability - Political, Economic, Religious

• Ethical standards - environment, slave / child labour

• Difficult to change

• Damages to the local job markets

• Loss of Relationship building opportunity with key stakeholders

If there is no need to buy (outsource) any products or services from outside the organisation, then there is no need to perform any of the procurement management processes.

However you will find that most (if not all) projects will contain some sort of outsourcing.

The Procurement Management Process consists of 3 broad stages:

1. Plan.

2. Source.

3. Manage.

• Project Schedule:

– One of the important artefacts generated during the project planning phase

– Is used and maintained throughout the project to monitor and track project progress

- is a living document

1. Breakdown the task into small chunks you can deal with – Work Breakdown Structure (WBS)

2. Identify the interdependencies between the broken down tasks and develop a task network

3. Estimate the effort and the time allocation for each task

4. Allocate resources for tasks and validate effort

5. Develop the project schedule

• Planning and executing large tasks is challenging:

– Estimating the time and resources

– Identifying interim goals and deliverable

– Progress monitoring

• Solution is to break the task down to manageable units:

– Each task should have a specific outcome or a deliverable

– Results in a Work Breakdown Structure (WBS)

Develop a project schedule

• Project Schedule will answer two important questions not answered so far:

– How long will the system take to develop?

– How much will it cost?

• Two widely used graphical notations to represent the Project Schedule

– Gantt charts

• A bar chart that shows the schedule against a calendar

– PERT (Program Evaluation and Review Technique) charts

• An activity network that shows the dependencies among tasks and the critical path

• Milestones

– Mark specific points along a project timeline

– These points may signal anchors such as:

• a project start and end date

• a need for external review

• start and end of a phase

• a completion of a deliverable

• Deliverable

– Specific artefacts that are of interest

– Examples of deliverables include:

• Project documents such as the Project Management Plan, Requirements Specification, Design Document, Test Plan etc.

• Prototypes

• Final application

• How do software projects fall behind schedule? One day at a time – Fred Brooks, the well-known author of the seminal article Mythical Man-Months

• Project scheduling is important, but tracking and controlling are even more important!

• How to track and control project progress?

– Periodic meetings where team members report progress

– Evaluating the results of reviews and audits conducted as part of the software engineering process

– Tracking formal project milestones

– Comparing actual start dates with scheduled start dates

– Meeting engineers and having informal discussions

– Using a formal method like earned value analysis

1. Expert judgement

– Several experts on the proposed software development technique and the application domain estimate project cost. These are then discussed, compared and adjusted until consensus is reached

– Some expert judgement techniques involve polling each expert independently, in some cases for three estimates, pessimistic estimate (p), optimistic estimate (o) and the most likely estimate (m), and the expert’s estimate is computed as the:

𝑒 = Τ (𝑝 + 4𝑚 + 𝑜) 6

– Delphi technique: asks several experts to make an individual judgement of the effort using any method they wish. Then, the average effort is calculated, and presented to all of the experts. Each expert is then given a chance to revise their estimate, in some cases after a discussion between all experts. This continues until no expert wishes to revise their estimate.

2. Estimation by Analogy

– The cost of a new project is estimated based on similar projects in the same application domain

3. Parkinson’s Law

– This law states that the work will expand to fill the time available

– The cost is determined by available resources rather than by objective assessment

– For example, if the software is to be delivered in 12 months, and 3 people are available, the effort is 36 person months

4. Pricing to win

– The cost is estimated to be whatever the customer has available to spend on the project - cost depends on the budget not on the software functionality

5. Algorithmic cost modelling

– A model is developed using historical cost information based on some software metric (usually its size) to the project cost

– When a project effort needs to be estimated, an estimate of the metric is computed

– Using the model, the effort is predicted

– The most general form of an algorithm cost estimate is given by:

𝐸𝑓𝑓𝑜𝑟𝑡 = 𝐴 × 𝑆𝑖𝑧𝑒^𝐵 × 𝑀

𝐴 - a constant factor that depends on the organizational practices 𝑆𝑖𝑧𝑒 -

size of the software estimated in a metric of choice (e.g. lines of code, function point, use case points)

𝐵 - a value between 1 and 1.5 derived experimentally

𝑀 − a multiplier made by combining process, product and developmet attributes such as stability of requirement, experience of the team

Basic steps in algorithmic cost estimation

1. Estimate the size of the development product

2. Estimate the effort in person-months or person-hours

3. Estimate the schedule in calendar months

4. Estimate the project cost in agreed currency

• Plan

– How to approach and plan risk management activities?

• Identify

– Identify the possible risks

• Analyse and Assess (Qualitative and Quantitative):

– Identify the relative priorities of the identified risks

• Respond (Action):

– How can we reduce the likelihood or impact of risks?

• Monitor and Control:

– How can we detect the ongoing status of our risks? How can we control them effectively and efficiently?

• Generic Risks:

– Threats or opportunities common to every software project (e.g. staff turnover, budget and schedule pressures)

• Product-specific Risks:

– Threats or opportunities specific to the product, and can only be identified by people who have a clear understanding of the product and technology

• Project risks – Affect the planning of the project e.g. Budget, Schedule, Scope, Personnel, etc.

• Product risks – Affect the quality or performance of the outcome being developed e.g. Design problems, implementation problems, interface problems, maintenance problems, verification problems

• Business risks – Affect the economic success of the project e.g. No demand for product, loss of management support, loss of external funding for the project etc.

• Risk identification

– Deals with using a systematic approach for identifying and creating a list of threats and opportunities that may impact the project’s goals

• Rick identification techniques

– Pondering

– Interviewing

– Brainstorming

– Checklists

– Delphi Technique

– SWOT Analysis

• Tools for monitoring and controlling:

– Risk Audits:

• external team looks at comprehensiveness of the identification process and ensuring other procedures and processes are in place

– Risk Reviews:

• internal reviews of risks periodically that result in status reports generated for PM and those who need-to-know

– Risk status meetings:

• risks must be reviewed and discussed in project status meetings, which are periodically held in projects (e.g. weekly meetings)

Quality Assurance -> Quality Planning -> Quality Control

1. Quality assurance:

The establishment of a framework of organizational procedures and standards that lead to high-quality software

2. Quality planning:

The selection of appropriate procedures and standards from the framework, adopted for the specific project

3. Quality control:

Ensuring that the software development team has followed the project quality procedures and standards

• Advantages of technical reviews:

– Can be performed on any software artefact, whereas many “hard” methods of quality assurance, such as testing and measurement, can be performed only on executable artefacts.

– Earlier detection of problems in software artefacts leads to lower costs of resolution.

– Studies show that roughly 30-70% of all programming faults found in a project were located using source code reviews, and up to 80% according to studies performed by IBM. Some studies demonstrated that review techniques found several types of faults that testing failed to find, and vice-versa.

– Reviews find the actual faults in source code, in contrast to testing, which merely indicates that there is a fault somewhere in the program. After a fault is detected with testing, it must then be located.

– Due to internal pressure of getting software releases out the door, programmers make more mistakes when correcting faults that were found during testing than they do correcting faults during the review phase

• Disadvantages of technical reviews:

– Could be time and resource consuming

– Should be carefully planned and executed to get the desired outcomes

• Types of technical reviews

– Informal Reviews

– Formal Reviews

– Walk throughs

– Code inspections

– Audits

QC and Management

• Involves monitoring the software development process to ensure that the quality assurance procedures and standards specified in the SQP are being followed

1.Understand the role of configuration management

• Software projects generate a large number of different types of artefacts

• The aim of configuration management is to manage change properly without losing overall consistency through:

– establishing processes;

– setting up repositories; and

– using other appropriate tools and techniques

• Configuration Management (CM) addresses the following:

– How do we manage requests for change?

– What and where are the software components?

– What is the status of each software component?

– How does a change to one component affect others?

– How do we resolve conflicting changes?

– How do we maintain multiple versions?

– How do we keep the system up to date?

CM Aims:

1. To identify all items that collectively will make up the configuration

2. To manage changes to one or more of these items so that the collection remains consistent

3. To manage different versions of the product

4. To assure software quality as the configuration evolves over time

• Identification

– the configuration items necessary for the project are identified

• Version control

– processes and tools are chosen to manage the different versions of configuration items as they are developed

• Change control

– changes that affect more than just one configuration item are managed

• Configuration auditing

– the consistency of the configuration is checked

• Configuration reporting

– the status of configuration items is reported