

Software Processes & Project Management Notes

<i>Introduction to Projects and Project Management</i>	<i>1</i>
<i>Project Management Plan & SDLCs</i>	<i>3</i>
<i>Individuals, Motivation and Teams</i>	<i>7</i>
<i>Communication & Stakeholder Management</i>	<i>9</i>
<i>Project Planning and Scheduling</i>	<i>13</i>
<i>Cost Estimation.....</i>	<i>17</i>
<i>Risk Management</i>	<i>21</i>
<i>Quality Management</i>	<i>24</i>
<i>Ethics, Procurement, Outsourcing & Contracts.....</i>	<i>28</i>
<i>Configuration Management.....</i>	<i>31</i>

Introduction to Projects and Project Management

1. What are the characteristics of project?
 - Temporary
 - Creates a unique product, service or outcome
 - Introduce change
 - Has defined beginning and end
 - Cross-functional
 - Deal with unknown
 - Unique
 - Vary in size, people, time and money
2. Why do organisations use projects?
 - Providing strategic alignment
 - Prioritising activities
 - Delivering change formally
 - Managing limited resources
 - Establishing ownership and accountability
 - Clarifying what will be done, when, who, why
3. Where does the value of project management lie in?
 - Organising and structuring scarce resources
 - Managing risks
 - Identifying and clearing issues
 - Managing and implementing change
 - Retaining and re-using knowledge
 - Organisational wide learning from past success and failure
4. What are the responsibilities of project managers?
 - Project planning: scope, management plan, schedule, policies and procedures
 - Organising: team structure, roles, services, staff positions
 - Leading: setting direction, coordinating activities, motivating, assigning work
 - Controlling: defining project baseline, tracking progress, reporting status, taking actions
5. Project management methodologies:
 - Waterfall: sequential, complete 1 task and then the next
 - Agile: adapting to changes, iterative outcomes
 - Prince2: starting & initiating – planning & directing – stage control – stage management – closing
 - PMBOK: initiating – planning – monitoring & controlling – executing – closing

6. What are the key drivers of why projects fail/succeed?
 - Executive sponsorship (15%)
 - Emotional maturity (15%)
 - User Involvement (15%)
 - Optimisation – statement of requirements (15%)
 - Skilled resources
 - Standard architecture
 - Agile process
 - Modest execution
 - Project management expertise
 - Clear business objectives
7. What is project screening?
 - A mechanism for organisations to undertake an initial review and filtering of all projects wishing to be undertaken
 - Required as there are always more projects to be completed than funds to allocate
8. What is the process of project initialization?
 - Analyse case study (business needs)
 - Analyse project constraints (scope, time, cost) – person day
 - Develop business case
 - Develop project charter (stakeholder analysis)
9. How to analyse business needs?
 - The goal of the project
 - Challenge: what is known to exist; requiring resources
 - Risk: May not happen; causing failure to reach the goal; needs mitigation strategy
10. What is Business Case?
 - Supporting decision making in initial and continued investment
 - Demonstrating how the project adds value to the organisation
 - A living document throughout the project
 - Containing executive summary, why it is required, business options, expected benefits, expected dis-benefits, timescale, costs, investment appraisal and major risks
11. Who is responsible for building Business Case?
 - Corporate: provides mandate; holds senior users; post project benefits validation
 - Executive/Sponsor: own the business case; reviewing the benefits
 - Senior users: accepting benefits and delivering; ensuring quality standard; actual/forecasted benefit realisation

- Project manager: preparing business case; risk assessing and impact analysing; assessing and updating business case
 - Project Assurance/QA: assists in developing business case; ensuring value for money and risks; monitoring changes and validating it
 - Project support: capturing data and preparing management reports; supporting for all stakeholders (schedules, cost analysis, minutes, actions, supplier liaison)
12. What investment techniques can be used to analyse financial benefit?
- Return on investment (ROI, income/investment, higher is better, required 11% to 14%)
 - Net present value (NPV, most often used, discounting all future costs and benefits)
 - Payback period (how long it takes to recover initial cost, tracking net cash flow, <1 year)
 - Rough order to magnitude (ROM, based on initial or incomplete data)
13. What is project charter and how is it used?
- A summary project proposal to secure approval for the project goals and terms
 - Useful as part of Business Case
 - Primary Use: communicate, engage, gain buy-in and obtain approvals
 - It contains project description, project team, milestones, target date, costs and gains

Project Management Plan & SDLCs

1. What is a process?
- A series of progressive and interdependent steps by which an end is attained
 - Project management is a process that defines a series of tasks to deliver an outcome
 - System development lifecycle describes a process for planning, creating, testing and deploying and information system
2. What is Project Management Plan and when it should be used?
- Defining how the project is executed, monitored and controlled
 - Formal approved document that owned and controlled by project manager
 - Used as part of mobilisation and on-going management of the project
 - Primary Use: establish and manage the project; defines all key items need to consider
3. What are typical PMP components (Formal)?
- Project information: executive summary, financial authority to proceed, key stakeholders, scope, delivery approach/SDLC, resources/people, key milestones, budget, business value (financial & non-financial), lessons learned applied to this project, constraints

- Project governance: roles and responsibilities; schedule, risk management, cost estimation, quality assurance, configuration management (change management)
4. What activities are included in SDLC model?
 - Requirements gathering
 - Systems / architectural design
 - Implementation / coding
 - Integration
 - Testing
 - Delivery and Release – deployment
 - Maintenance
 5. What are examples of formal processes and agile processes?
 - Formal: Waterfall; Incremental; V-Model
 - Agile: Extreme programming; Scrum; Kanban
 6. When to choose Formal models?
 - The customer has a clear view of what they want
 - Requiring little or no change to requirements
 - Requirements are clearly defined and documented
 - Technologies and tools are well-known
 - Large scale systems and applications
 7. What are the advantages and disadvantages of Waterfall model?
 - Advantages: simple and easy to manage and monitor; phases are processed and completed one at a time; documentation available at the end of each phase; requirements stable and precise
 - Disadvantages: the following phase cannot start until the previous phase has finished; difficult to respond to changing customer requirements; client feedback and bad news known late in process; difficult to integrate risk management due to uncertainty
 8. What are the advantages and disadvantages of Incremental model?
 - Advantages: cheaper and easier to change the scope/requirements; customer can get important functions at early stage; easier to get customer feedbacks; easier to test and debug
 - Disadvantages: more management attention is required; more resources may be required; hard to define and partition the increments; integration risk; each phase of iteration is rigid with no overlaps
 9. What are the advantages and disadvantages of V-model?

- Advantages: simple and easy to understand and use; easy to manage; each phase has specific deliverables; high change of success due to early testing; works well with small projects when requirements are easily understood
- Disadvantages: rigid process; little flexibility and hard to adjust scope; no early prototypes; no clear path for problems found; changes in later change cause all stages to be changed; client feedback known late in process

10. What are the product management stages for Formal and Agile models?

- Formal: initiation – plan – execute – monitor & control – close
- Agile: initiation – sprint plan – sprint – review & retrospective - release

11. When should you choose Agile?

- Requirements continue to emerge
- Customer gives time to project
- Change is welcome
- Small or medium-sized product

12. What are principles of Agile?

- Customer involvement
- Incremental delivery
- People not process
- Embrace change
- Maintain simplicity

13. What are the advantages and disadvantages of Agile?

- Advantages: customer satisfaction by rapid deliverables; adapting to changing requirements; focus more on people and interactions
- Disadvantages: difficult to assess the effort at the start; can be intense for the team; requires experience; requires teamwork skills

14. What are the key characteristics of Scrum?

- Self-organising teams
- Product progresses in series of sprints
- Requirements are captured as items in product backlog
- Time frame is contained to manageable size (weeks or months)

15. What are the roles in a Scrum team?

- Product Owner: defines the features, release date and content; responsible for the benefits; prioritise and adjust features; accepts or rejects work results (internal stakeholder, a person in the organisation)

- Scrum Master: represents management to the project; enacting Scrum values and practices; enables close cooperation (a member of the Scrum Team)
- Development Team: 5-9 people; cross-functional; co-located

16. What are Scrum Ceremonies?

- Sprint planning: design (how to achieve sprint goal); create sprint backlog (user stories) from product backlog; estimate sprint backlog in team velocity and story points; product owner sets priority; release plan is created; high-level design is considered
- Sprint review: informal; demo of new feature; a showcase for a big audience
- Sprint retrospective: 30 minutes; Scrum Master, Product Owner and team attend
- Daily stand-ups: 15 minutes; only team member, Scrum Master and product owner talk

17. What are User Story and Story Points?

- Scrum express the product features by user stories (epic, feature, low level)
- User story is a requirement expressed from the perspective of customer
- User story template: As a ..., I want... so that...
- Story points: a relative measure of the size of a user story
- The raw value of story points is unimportant, what matters are relative values
- SME/Product Owner has a “Just-In-Time” conversation with the developer to add the detailed requirements

18. What are Scrum Artefacts?

- Product backlog: requirements; reprioritized by product owner at the start of sprint; scope is expressed as a list of features; release milestones annotated to list
- Sprint backlog: feature selected for this iteration; visual Kanban board
- Burndown charts: a graphical representation of work left to do versus time; sprint burndown chart (x: days within the sprint; y: effort-hours remaining/story points remaining) VS release burndown chart (x: sprints within the release; y: story points remaining)

19. How does Scrum envisage the cost, date and features the product will deliver?

- Groom user stories in product backlog
- Select high business value user stories into a release
- Add quick and cheap story point estimation to these user stories
- Use team velocity to estimate when done
- Actual chart updated after each sprint by the Scrum Master

20. How do you monitor the productivity of the Sprint?

For user stories in the sprint backlog:

- Decompose each and re-estimate more carefully
- Will this number of story points fit in Sprint?

- Monitor “done” user stories on Sprint Burndown chart
- Update actual velocity at end of Sprint

Individuals, Motivation and Teams

1. What is motivation and how people are motivated?
 - Motivation is the driving force within individuals that propels them into action
 - Organisations are groups of individuals that are structured and managed to meet a need or to pursue collective goals
 - Maslow Hierarchy of Needs: physiological – safety – social – esteem – self-realization
 - Hertzberg Two factor Theory: motivator factors (achievement, recognition, responsibility, work itself, advancement, personal growth); hygiene factors (working conditions, co-worker relations, policies and rules, supervisor quality, base wage)
2. How leaders lead and managers manage?
 - Leadership is the ability to influence and direct people to achieve a common goal
 - Leaders inspire and motivate people to meet the goal
 - Management is the process where resources are used, and decisions made to achieve the goal
 - Managers set objectives and decide how to achieve them
 - Project managers need to be both leaders and managers
 - Power: referent, expert, legitimate, reward, coercive
 - Influence: authority, assignment, budget, promotion, money, penalty, work challenges, expertise, friendship
3. What are habits of highly effective people by Stephen Covey?
 - Take initiative – be proactive
 - Focus on goals – begin with the end in mind
 - Set priorities – put first things first
 - Only win when others win – think win/win
 - Communicate – seek first the understand, then to be understood
 - Cooperate – synergise
 - Reflect on and repair deficiencies – sharpen the saw
4. What is the difference between group and team?
 - A group is a collection of people who do not necessarily work collectively toward the same goal
 - A team is two or more individuals consciously working together to achieve a common objective

- A group becomes a team when members demonstrate a commitment to each other and to the end goal
5. Why do we need teams?
 - Teamwork creates human synergy
 - Complementary teamwork skills are one of the most commonly required skills in the work environment
 - Few individuals possess all skills required to accomplish tasks
 - Shared accountability increases likelihood of success
 - Substantial benefits: enhanced opportunities, greater productivity, increased ownership and accountability, more creativity and innovation, greater joy and satisfaction, broader perspective, increased representation, increased equality, more dialogue
 6. How does team form and perform? (Tuckman's Team Development Model)
 - Forming: establishing ground rules and preserving formalities; high dependency on the leader (team experience initial decline in performance after forming stage)
 - Storming: members communicate, but maintain strict individuality; leader coaches and support
 - Norming: team bonding and higher acceptance of perspectives; leader moved to facilitator and enabler
 - Performing: less emphasis on hierarchy and more on flexibility; leader delegate and oversees
 - Adjourning: yearly assessment and plan for acknowledging individual contributions; leader acknowledge, recognises and directs
 7. How to judge whether a team is effective or not?
 - Positive signs: communication, brainstorming, consensus, problem solving by the group, commitment, effective and inclusive meetings, positive relationship
 - Negative signs: lack communication, no clear roles, lack of work quality, work alone, no one accepts responsibility, lack of support for others, frequently absent
 8. What are different team structures?
 - Controlled centralised: leader coordinates tasks and directs work; communication and control are vertical; sub-teams with leader to direct and guide sub-groups; there is very little or no involvement of groups in decision making
 - Controlled decentralised: leader coordinate tasks; sub-teams with leaders; communication in group will be horizontal and between groups, communication will be vertical; problem solving is team task
 - Democratic decentralised: There is no group leader as such. A task coordinator is chosen for short duration based on skill or experience level.
 - SWAT team: no permanent leader; highly specialised team; particular tasks

- Scrum team: product owner, scrum master, scrum team
9. What are the advantages and disadvantages of teams?
- Advantages: creative and collective problem solving; more knowledge from interaction; stimulate performance, motivation and output; if people leave, others can continue the task; increased ownership of the overall outcome
 - Disadvantages: takes time, effort and skill to manage; some individuals may become overshadowed/overwhelmed; unequal involvement; social loafing; group think
10. What are different roles in a team?
- Initiator: offer ideas, solutions, brainstorm, lateral thinker
 - Information seeker: wants facts
 - Information giver: describes own experience, offer facts, clarification
 - Coordinator: combine contribution of others
 - Evaluator: assess quality of contributions
 - Encourager: praising, accepting, cohesion and warmth
 - Harmonizer: build consensus, humour to neutralise anger
 - Standard setter: focus on goals and standards
 - Follower: agreeable
 - Group observer: provides feedback

Communication & Stakeholder Management

1. What is the difference between hearing and listening?
- Hearing: the act of perceiving sound by ear
 - Listening: requires concentration and mentally organising what you hear to make sense
 - Listening demands real effort and is an essential life skill
 - Truly effective listeners are rare, and few people practice listening or have been trained
 - The process of listening: predicting – receiving – assigning meaning – assess/validate – remembering
2. Why do we listen?
- Promotes problem-solving abilities
 - Demonstrates acceptance of others
 - Builds and retains trust in relationships
 - Increase speaker's receptiveness to thoughts and ideas of others
 - Increases self-esteem of the speaker helps you evaluate messages
 - Help you understand and retain information
 - Allows you to help others

3. What is the importance of active listening?
 - Shows the speaker you are concerned or interested
 - Leads to getting better information
 - Encourages further communication
 - Has the potential to enhance relationships
 - Can calm down someone who is upset
 - Invites others to listen to you
 - Leads to better cooperation and problem solving
4. What are different types of listening?
 - Passive listening: take in information without processing or reacting
 - Active/Empathetic listening: show interest, ask questions, avoid distractions, direct eye contact, do not interrupt, read both verbal and nonverbal messages
 - Critical listening: determine speaker's motive, argue ideas, distinguish fact from opinion, recognise own bias, evaluate the message
5. What are challenges of listening?
 - Physiological limitations
 - Inadequate background information
 - Selective memory or expectations
 - Fear of being influenced or persuaded
 - Bias and being judgemental
 - Boredom or interference from emotions
 - Partial listening and distractions – mobile phones/background noise
 - Physical barriers including environment, lighting, uncomfortable seating
 - Cultural differences
 - Past experiences
 - Jargons and acronyms
6. What are communication challenges?
 - Individual: semantics (meaning), perception (interpretation), communication channel, feedback, anxiety, culture
 - Team: status, silos, information overload, lack of communication, protocol (rules)
7. What are key communication skills in project management?
 - Verbal communication including listening
 - Delivering presentations / updates
 - Recognising, defining and solving problems
 - Motivating and influencing others
 - Delegating

- Setting goals and articulating vision
- Self-awareness
- Managing conflict
- Networking

8. Why is it important for a Project Manager to communicate well?

Because they must have the ability to:

- Read/understand the client
- Run a meeting
- Communicate written and orally thoughts accurately
- Manage the team
- Influence your environment
- Communicate, ensure alignment and buy-in purpose

9. What is communication plan and how is it used?

- It is used to assist managing and coordinating key communication messages
- Good project communication plan ensures communications to be effective and efficient; allows the project manager to be proactive; sets common understanding of what will be done and when; clarifies who is responsible for key item, what will be delivered and by who
- It defines what information will be communicated; communication channel (meetings, email, telephone); when information will be distributed; who is responsible; communication needs of stakeholders; resources for communication; how sensitive and confidential information is communicated and authorised; communication flow; constraints; templates and documents; escalation process to resolve conflicts or issues
- It is underpinned by a Communication Matrix (stakeholder, communication objective, format, frequency, owner and importance), a Communication Flowchart and a Communication Escalation Plan (priority, definition, decision authority, timeframe)

10. What are key communication considerations?

- Face to face meetings: 58% of communication is through body language
- Rarely does the receiver interpret a message exactly as the sender intended
- Geographical location and cultural background affect the complexity (working hours, language, cultural norms)
- Communication helps manage conflict effectively
- Spend time developing communication skills – practice & feedback
- Choose the right channel (hard copy, telephone call, voice mail, email, meeting, web)

11. How to run effective meetings?

- Determine if meeting can be avoided
- Define the purpose and intended outcome

- Determine who should attend the meeting
- Provide an Agenda before the meeting
- Prepare and handouts or visual aids
- Control the meeting – focus on agenda topics
- Only one person at a time talking – ground rules
- Summarize actions, dates, owners and next steps via Action Register
- Follow up formally with scheduled activities to ensure success

12. What are common conflicts across the project life cycle?

- Definition/mobilisation phase: relative importance between projects; which SDLC should be used; personnel to lead and drive; costs, benefits, timelines and milestones
- Planning phase: final schedule, costs, timeline, deliverables; availability of resources and when required; spending funds before projects approved; decision rights; technology
- Execution phase: schedule delay, technical problems, quality and staff; missed milestones; when resources required compared to planned due to delays; third party supplier management
- Delivery phase: schedule, quality and costs; scope – what was expected; missed milestones; release windows; staff burnout

13. What are differences of internal and external stakeholders?

- Internal stakeholders are those who are involved in the company's internal operations or who are directly affected by management decisions. E.g. shareholders, employees, board members, sponsor, project manager, team members
- External stakeholders are people who have a stake (interest) in some aspects of the project but are not in the software development organizations. E.g. customers, suppliers, governments/unions, general public, competitors

14. What is the process of project stakeholder management?

- Identifying stakeholders: Stakeholder Register (name, position, location, role and contact information)
- Planning stakeholder management: current and desired engagement levels (unaware – resistant – neutral – supportive – champion/leading); stakeholder interrelationship; communication requirements; management strategy for each; methods for updating stakeholder management plan
- Managing stakeholder engagement

15. What are included in stakeholder analysis?

- Name and organisations
- Role and unique facts
- Power/influence on the project
- Current engagement level

- Suggestions and strategies for managing relationships with each stakeholder

Project Planning and Scheduling

1. Why some projects are challenged? (successful – challenged – failed)
 - Lack of a scope document: making a detailed scope document that highlights all requirements is imperative
 - Inconsistent communication: 57% projects failed due to poor communication
 - Unrealistic expectations and deadlines: 60% of failed projects have deadline <1 year
 - Incompetent project manager and team: 80% successful projects managed by certified project managers
 - Lack of cohesion between team members: must have same goals
 - Poor monitoring and risk management
 - Poor planning: 40% projects fail due to poor planning and lack of resources
2. What are included in Project Planning?
 - Project scheduling
 - Cost estimation
 - Risk management
 - Quality management
 - Configuration management (quality management)
 - Resource management
 - Communication management
3. What is Project Schedule?
 - A living document used and maintained throughout the project to monitor and track project progress
 - It contains duration and dependencies for each task; people and physical resources required by each task; milestones and deliverables; project timeline
 - The project scheduling is based on plan-driven Formal SDLC model
4. What are the steps to develop Project Schedule?
 - Breakdown the task into small chunks – work breakdown structure (WBS)
 - Identify interdependencies between tasks and develop a task network: a successor task depends on a predecessor task (constrained and unconstrained)
 - Estimate effort and time allocation for each task
 - Allocate resources for tasks and validate effort
 - Develop project schedule

5. What are different types of task dependencies?
 - Finish-to-start: predecessor must finish before successor can start (most common)
 - Start-to-start: predecessor must start before successor can start
 - Finish-to-finish: predecessor must finish before successor can finish
 - Start-to finish predecessor must start before the successor can finish

6. How to estimate the effort and time allocation for each task?
 - Effort estimation: man-months / person-months: the time in months for a single person working full time to complete the task
 - Putnam-Norden-Rayleigh (PNR) curve shows time (X axis) against effort and cost (Y axis). Important points: lowest-cost delivery time; optimal delivery time
 - Time estimation: $T_E = (O + 4M + P) / 6$ Expected/Optimistic/Normal/Pessimistic time

7. How to allocate resources for tasks?
 - Number of personnel: $N = \frac{Effort}{T}$
 - Project manager has to consider the expertise of people, and the availability of them for tasks, which might require validation and adjustment of the schedule

8. What are the definitions of the following terms?
 - Activity (Task): part of a project that requires resources and time
 - Milestone: completion of an activity that provides evidence of a deliverable completion
 - Free float/slack: the amount of time that a task can be delayed without causing a delay to subsequent tasks
 - Total float/slack: the amount of time that a task can be delayed without delaying project completion
 - Critical path: the longest possible continuous path taken from the initial event to the terminal event
 - Critical activity: an activity that has total float equal to zero

9. What are milestones and deliverables?
 - Milestones: mark specific points along a project timeline, which may signal anchors such as project start and end date, need for external review, start and end of a phase, and a completion of a deliverable.
 - Deliverable: specific artefacts that are of interest. E.g. project documents such as project management plan, requirements specification, design document, test plan; prototypes; final application

10. How to use graphical notations to represent project schedule?
 - Gantt chart: a horizontal bar chart showing tasks against a timeline; useful tool for monitoring project progress; linked Gantt charts contain lines indicating dependencies; progress Gantt charts give visual presentations of the progress by shading tasks

- PERT chart: showing the dependencies along with time related information and the critical path; help understand the characteristics of the project for managers to do scheduling trade-offs; help perform critical path analysis; monitor project progress
11. What are the key steps in critical path method?
 - Specify each activity
 - Establish dependencies
 - Draw the network diagram
 - Calculating earliest start time (ES), latest start time (LS), earliest finish time (EF), latest finish time (LF) and slack time
 - Identify the critical path: path with the longest duration; activities have a total free slack of 0; a delay in any of activities in critical path will case the project to delay
 12. What are the advantages of critical path?
 - clearly presents the time required to complete tasks
 - makes dependencies visible between the project activities
 - enables the calculation of the float (slack) of each activity
 - encourages the Project Manager to reduce the project duration by optimizing the critical path, which can be achieved by removing dependencies or shortening the duration of activities in the critical path
 13. How to use project schedule to monitor and track project progress?
 - Periodic meetings where team members report progress
 - Evaluating the results of reviews and audits
 - Tracking formal project milestones
 - Comparing actual start dates with scheduled start dates
 - Meeting engineers and having informal discussions
 - Using formal methods like earned value analysis
 14. What is earned value analysis (EVA)?
 - A project management tool that is used to measure project progress
 - It compares the actual work completed at any time to the original budget and schedule
 - It can be used to report current/past project performance
 - It predicts future project performance based on current/past performance
 - Results can be expressed in dollars or percentage
 15. What are the steps to perform EVA?
 - Determine the percent complete of each task
 - Determine Planned Value (PV): the amount of task that is supposed to have been completed; in monetary terms as a portion of the task budget
 - Determine Earned Value (EV): the amount of task that is actually completed

- Obtain Actual Cost (AC): the total costs incurred in a given period
- Calculate Schedule Variance (SV): $SV = EV - PV$
- Calculate Cost Variance (CV): $CV = EV - AC$
- Calculate Other Status Indicators: $SPI = EV / PV$, $CPI = EV / AC$
- Interpret Results: SV negative – behind schedule; CV negative – over budget

16. What are the principles in Agile planning?

- Detailed planning is deferred until the start of the iteration in order to handle change
- Planning is based on light weight lists. Gantt and PERT are considered less useful.
- Plan short iterations (an iteration includes requirements, design and test)
- Deliver working software
- Use “Just in time (JIT) planning” – next iteration
- Use the team

17. What are different types of Release Planning?

- Fixed-date release planning: fixed date and budget and have the scope flexible; used when date is more important
- Fixed-scope release planning: fixed scope and have the date and budget flexible; used when scope is more important

18. What are the steps of Fixed-date release planning?

- Determine how many sprints are in the release
- Groom the product backlog by creating, estimating the size of, and prioritizing product backlog items
- Measure or estimate the team’s velocity as a range
- Create a will have line by multiplying the slowest number in the velocity range by the number of sprints
- Create a might have line by multiplying the largest number in the velocity range by the number of sprints

19. What are the steps of Fixed-scope release planning?

- Groom the product backlog to by creating, estimating the size of, and prioritizing and identify the must-have stories
- Determine the total number of must-have story points
- Measure or estimate the team’s velocity as a range
- Divide the number of must-have story points by the fastest velocity in the range
- Divide the number of must-have story points by the slowest velocity in the range
- Show the minimum and maximum number of sprints on burndown chart

20. What are different levels of planning in Scrum?

- Strategy

- Portfolio: a year or more; stakeholder and product owners
- Product (envisioning): months; stakeholder and product owners; roadmap and vision
- Release: 3-9 months; Scrum team, stakeholders; balance customer value and quality against constraints of scope, schedule and budget
- Sprint: every iteration (one week to one month); Scrum team; what features to deliver
- Daily: Scrum Master, development team; how to complete committed features

Cost Estimation

1. What is software cost estimation and what are the challenges?
 - Estimation of how much money, effort, resources and time would be required to build a specific software-based system or product
 - Highly-ranked among failure reasons: Inaccurate cost & task time estimation (28%, 26%)
 - There is no exact science for cost estimation; it will not be considered as all accurate
 - No person can reasonably predict what can go wrong in the project
 - Most estimation methods assume things will proceed as expected and simply adds some slack to account for what can go wrong
 - Possible solutions: delay estimation but less useful; base estimation on data from previous projects; estimate for small parts of the system; use experience-based method
2. What techniques can be used for cost estimation?
 - Expert judgment: several experts discuss, compare and adjust until consensus is reached; three estimates: $e = (p + 4m + o)/6$; Delphi technique: individual judgment & revise
 - Estimation by analogy: based on similar projects in the same domain
 - Parkinson's law: cost is determined by available resources rather than objective assessment; work expands so as to fill the time available for its completion
 - Pricing to win: the cost depends on the budget of customer not on the functionality
 - Algorithmic cost modelling: a model is developed using historical cost information based on some software metric (usually its size) to the project cost
3. What is the formula for Algorithmic cost modelling?
 - $Effort = A * Size^B * M$
 - A: a constant factor that depends on the organizational practices
 - Size: estimated by source lines of code, function of point, or use case points
 - B: a value between 1 and 1.5; increases with the size and complexity of the system
 - M: a multiplier combining process, product and development attributes, such as stability of requirements and experience of the team
4. What are the basic steps in Algorithmic cost estimation?

- Estimate the size of the development product
 - Estimate the effort in person-months or person hours
 - Estimate the schedule in calendar months
 - Estimate the project cost in agreed currency
5. What are the techniques for estimating software size?
 - Source Lines of Code (SLOC): based on code; physical SLOC counts the number of lines excluding comments and blank lines; logical SLOC measure the number of executable statements, but specific definitions are tied to computer languages
 - Function Points (FP): based on software requirements specification (SRS); used to express the amount of functionality in a system; can be used to predict the number of errors and components, and to measure productivity
 - Use-case Points (UCP): based on use cases; concept is similar to FPs
 6. What is Software Requirements Specification (SRS)?
 - A document that specifies what is expected of a software system
 - Referred to as the requirements of the system
 - It contains functional requirements that specify the required functions, and non-functional requirements such as performance, reliability, scalability (quality)
 7. What are the advantages and disadvantages of SLOC?
 - Advantages: easy to count and can be automated using a tool; an intuitive metric because it can be seen, and the effect can be visualized
 - Disadvantages: variable depends on programmer experience, programming language, framework support and reuse; hard to estimate in analysis and design phase; lack of universally accepted definition for a line of code
 8. What are the advantages and disadvantages of Function Points?
 - Advantages: measure the size of solution instead of problem; only need requirements; can be estimated early in analysis and design; independent of technology and programming languages
 - Disadvantages: well-defined requirements specifications are necessary; hard to gain proficiency; time-consuming thus costly
 9. What are the advantages and disadvantages of Use Case Points?
 - Advantages: can be measured very early in the project life cycle; found to be close to actuals when estimation is performed by experienced people; easy to use and no additional analysis; use case are used vastly as a method of describing requirements
 - Disadvantages: can only be used when requirements are written in the form of use cases; dependent on goal-oriented and well-written use cases; technical and environmental factors have high impacts on UCP; not as well established as FPs

10. What is the meaning of the following acronyms?

- Internal Logical File (ILF): grouping of data that the system maintains and modified by external inputs. E.g. tables in relational databases, files containing user setting
- External Interface File (EIF): grouping of data maintained external to the system and may be used by the system. E.g. data hosted on third-party servers, data structures holding information about system state
- External Input (EI): inputs to the system from users or other applications; used to control the flow of the system or provide data; modify internal logic files. E.g. data fields from users, program source code to a compiler, files feed from external applications
- External Outputs (EO): outputs to users that provides information about the state of the system. E.g. screens, error messages, reports shown to users
- External Queries (EQ): inputs to the system that used to query internal logic file and provide outputs; the output is retrieved directly with no derived data. E.g. reading a user setting, reading a record from a database table
- Data Element Type (DET): a unique user recognizable, non-repetitive field
- Record Element Type (RET): a user recognizable subgroup of data elements in ILF & EIF
- File Type Reference (FTR): a file (ILF, EIF) referenced by a transaction

11. How to compute Function Points (FP)?

- Categorize requirements as data functions (ILF, EIF) & transaction functions (EI, EO, EQ)
- Estimate a complexity value for each category (simple, average, complex): data functions (ILF, EIF) are broken down into DETs and RETs; transactional functions (EI, EO, EQ) are broken down into DETs and FTRs; determine the complexity according to tables:

Complexity table for Data Functions

	DETs		
RETs	1-19	20-50	51+
1	Simple	Simple	Average
2-5	Simple	Average	Complex
6+	Average	Complex	Complex

Complexity table for Transaction Functions

	DETs		
FTRs	1-5	6-19	20+
1	Simple	Simple	Average
2-3	Simple	Average	Complex
4+	Average	Complex	Complex

- Compute count total from complexity: $unadjusted\ FP = \sum (functions * complexity\ value)$
- Estimate value adjustment factors (VAF): rank each characteristic on a scale of 0-5 (not important-essential); characteristics include data communications, distributed data processing, performance, heavily used configuration, transaction rate, online data entry, end-user efficiency, online update, complex processing, reusability, installation ease, operational ease, multiple sites, facilitate change
- Compute adjusted FP: multiply business function by VAF; $Adjusted\ FP = unadjusted\ FP * (0.65 + 0.01 * VAF)$

12. How to compute Use Case Points (UCP)?

- Compute unadjusted use case weight (UUCW): count the number of transactions to classify use cases (simple:1-3; average:4-7; complex:8+); $UUCW = N_s * 5 + N_A * 10 + N_C * 15$
- Compute unadjusted actor weight (UAW): classify actors by simple (using well defined API, average (using standard protocol like HTTP/TCP), complex (using GUI); $UAW = N_s * 1 + N_A * 2 + N_C * 3$
- Compute technical complexity factor (TCF): score each factor between 0-5 (irrelevant-essential); factors include distributed system (2.0), response time (1.0), end-user efficiency (1.0), internal processing complexity (1.0), code reusability (1.0), easy to install (0.5), easy to use (0.5), portability to other platforms (2.0), system maintenance (1.0), concurrent/parallel processing (1.0), security features (1.0), access for third parties and end user training(1.0); $TF = \sum_{i=1}^{13} S_i * W_i$; $TCF = 0.6 + TF/100$
- Compute environmental complexity factor (ECF): score each factor between 0-5 (irrelevant-essential); factors include familiarity with development process used (1.5), application experience (0.5), object-oriented experience of team (1.0), lead analyst capability (0.5), motivation of the team (1.0); stability of requirements (2.0), part-time staff (-1.0) and difficult programming language (-1.0); $EF = \sum_{i=1}^8 S_i * W_i$; $ECF = 1.4 + (-0.03 * EF)$
- Compute the final size estimate: $UCP = (UUCW + UAW) * TCF * ECF$

13. What are the principles of COCOMO II model?

- An empirical model derived by collecting data from a large number of software projects. These data were analysed to discover the formulae that were the best to fit the observations
- It has been widely used and evaluated; well documented, available in the public domain and well supported by tools
- It embeds sub models that produce increasingly detailed estimates: application composition model based on number of application points, used for prototyping projects that developed by composing existing components; early design model based on number of function points, used during early stages after requirements are established; post-architecture model based on source code lines number, used once initial architectural designs for the system is available

14. How to use early design model in COCOMO II?

- Based on the formula $Effort = A * Size^B * M$
- $A = 2.94$, estimated empirically
- Size is estimated based on logical lines of code, can be computed from FPs

- B vary from 1.01 to 1.26; $B = 1.0 + 0.01 * \sum_{i=1}^5 W_i$; W_i is a scaling factor value ranging from 0-5 depending on familiarity of the application domain, level of flexibility, architecture completed and risks eliminated, team cohesion, process maturity
- M is computed based on seven project and process attributes, including product reliability and complexity (RCPX), reuse required (RUSE), platform difficulty (PDIF), personnel capability (PERS), personnel experience (PREX), schedule (SCED), and support facilities (FCIL). $M = PERS * RCPX * RUSE * PDIF * PREX * SCED * FCIL$
- Estimating nominal delivery time (T) and number of personnel (N): $N = Effort/T$, $T = 2.5 * Effort^{(0.33+0.2(B-1.01))}$

15. What is the process of effort estimation in Agile?

- Develop user stories for the system
- Estimate the number of story points for each story, basing the estimate on the number of story points on previous stories (estimate by analogy, decompose a story into tasks, use the right units and pattern-based scale, use group-based estimations)
- Use the team's velocity from previous experience to estimate the delivery time of the project: $V = SP/T_i$, SP is the number of completed story points, T_i is time period for completing the story points; $T = \frac{\sum_{i=1}^n SP_i}{V}$, T is estimated delivery time, V is velocity, SP_i is the number of story points in the i-th user story, n is the total number of user stories
- In the case of fixed-scope release planning, develop a release burn-down chart
- During development, measure the actual velocity of the team
- Using the velocity to re-estimate the time it will take to deliver the product

16. What techniques can be used for Agile estimation?

- Planning poker: customer reads story – team estimates – discusses – estimates again
- Bucket System: randomly choose an item and place it in “8” bucket; choose another random item and discusses its relative position; divide and conquer; final review
- Relative mass valuation: pick one story and estimate it as large, medium or small; assign points values starting from easiest one as 1, and story with twice difficulty as 2
- T-shirt sizes: estimate whether a story is XS, S, M, L, XL or XXL
- Affinity estimation: silent relative sizing – edit on the wall – place in size bucket – product owner challenge – store the data
- Dot voting: vote preferences by placing a coloured dot on items with higher priority than other items

Risk Management

1. What is the relationship between risk and uncertainty?

- A risk is an uncertain event or condition that has a positive or negative effect on the project objectives (PMBOK)
 - Uncertainty means the lack of complete certainty about an event or outcome; the event or outcome has a probability of less than 1. E.g. outcome of a sporting event
 - Risk is uncertainty that has an impact; it is a result of uncertainty, but not every uncertainty is a risk
2. What is project risk management?
 - Projects have many possible risks that could have significant impact on the outcomes
 - A planned risk management process is essential
 - The goal of project risk management is to minimize the impact of potential negative risks while maximising the impact of potential positive risks
 3. What are the steps of risk management process?
 - Plan risk management activities: the output is a Risk Management Plan (RMP) that documents the procedures for managing risks; it contains methodology, roles and responsibilities, budget and schedule, risk categories, risk probability and impact, tracking, risk documentation, contingency plans and fall-back plans
 - Identify possible risks by systematic approaches: generic risks & product-specific risks
 - Analyse the probability and impact of the risks, and assess the relative priorities of identified risks so that effective risk strategy can be formulated; approaches include qualitative (based on experience/intuition) & quantitative (mathematical and statistical)
 - Respond (action): requiring resources so that appropriate response strategies should be selected; can be documented as part of Risk Register containing Risk ID, Trigger (flags the risk has occurred), Owner, Response and Resources required
 - Monitor and control: triggers must be monitored to keep track of project risks; new threats and opportunities must be identified, analysed and responded
 4. What are different kinds of risk?
 - Project risk: affecting the planning of the project. E.g. budget, schedule, scope, personnel
 - Product risk: affecting the quality or performance of the outcome. E.g. design problems, implementation problems, interface problems, maintenance problems, verification problems
 - Business risk: affecting the economic success of the project. E.g. no demand for product, loss of management support, loss of external funding for the project
 5. What are techniques for risk identification?
 - Pondering: sitting and thinking about possible risks; common initial risk assessment task
 - Interviewing/questionnaires: getting knowledge from stakeholders & domain experts

- Brainstorming: using risk framework or work breakdown structure to identify threats and opportunities; encourage contributions from everyone; discuss and evaluate
 - Checklists: collated from experience; used as triggers for experts to think
 - Delphi technique: experts update their response based on others' responses
 - SWOT analysis: strengths, weaknesses, opportunities and threats
6. What are the steps of risk analysis?
- Estimating the risk probability (P): usually based on expert judgement; 0-1 inclusive
 - Estimating the risk impact (I): usually measured in a scale of 1-5, representing no impact, minimal impact, moderate impact, severe impact and catastrophic impact
 - Computing the risk exposure (P*I score): $Risk\ exposure = P * I$
 - Identifying the root cause of all risks
7. What are the techniques for risk assessment?
- Risk matrix is used to define the level of risk by considering the probability or likelihood consequence severity; it is a mechanism to increase visibility of risks and assist management decision making
 - Quantitative approaches that are based on modelling a particular risk situation; probability distributions of risks are the main consideration; common techniques include decision tree analysis, simulation and sensitivity analysis
8. What are common strategies to handle threats?
- Accept or ignore: the risk is of an acceptable exposure; or exposure is less than the cost of other techniques to avoid, mitigate or transfer it
 - Avoid: completely prevent the risky event from occurring by ensuring its probability or impact is 0
 - Mitigate: employing techniques to reduce the probability or impact of the risk; the residual risk should be analysed
 - Transfer: transferring the burden of the risk to another party. E.g. insurance, outsourcing a proportion of the work
9. What are common strategies to handle opportunities?
- Exploit: add work or change the project to make sure opportunity occurs
 - Enhance: increase the probability and positive impact of risk events
 - Share: allocate ownership of opportunity to a third-party
 - Accept: the cost to exploit or enhance is not justifiable so do nothing about it
10. What are the tools for monitoring and controlling risks?
- 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 that periodically held in projects. E.g. weekly meetings
11. What is the risk management process for Agile?
 - Build small piece of working software with minimal features
 - Showcase the product chunk to the stakeholders early
 - Fail fast and as cheaply as possible, and get timely feedback
 - Capture the risk item in the Product Backlog: the format can vary, optionally use feature driven development (FDD) syntax when the role is not obvious (action-result-business objective)
 - The product owner sets the priority of the risk item
 12. How does the key differences between SDLCs influence risk management?
 - The formal incremental SDLC has explicit architecture, explicit UX design (end user consideration) and explicit configuration, so the risk management should be planned ahead
 - Agile iterative SDLC features productivity increase, and it is responsive to feedback for client satisfaction, and building working software, so the risk management should be planned Just-In-Time

Quality Management

1. What is software quality?
 - End user's perspective (external quality characteristics): a system has quality if it is fit for purpose, reliable, has reasonable performance, easy to learn and use, helps the users in achieving their goals; if function is hard to learn but extremely important, users will judge the system to have high quality
 - Developer's perspective (internal quality characteristics): number of faults, ease of modifying the system, ease of testing, ease of understanding the system design, reusability of components, conformance to requirements, resource usage and performance
 - Software quality attributes: safety, understandability, portability, security, testability, usability, reliability, adaptability, reusability, resilience, modularity, efficiency, robustness, complexity and learnability (trade-off is necessary)
 - The quality of the process influences the quality of the product. Ensuring product quality typically involves measuring and assessing the product and processes

2. What is the process of quality management?
 - Quality assurance: establishment of a framework of organisational procedures and standards that lead to high-quality software; standards can be defined as a set of rules for ensuring quality
 - Quality planning: selection of appropriate procedures and standards from the framework, adopted for the specific project; the outcome is a software quality plan (SQP), sometimes called software quality assurance plan (SQAP)
 - Quality control and monitoring: ensuring the software development team has followed the project quality procedures and standards
3. What standards should be documented and why is it important?
 - Documents are the tangible manifestation of the software
 - Document process standards: how documents should be developed, validated and maintained
 - Document standards concerned with document identification, structure, presentation, changes highlighting
 - Document interchange standards: how documents are stored and interchanged between different documentation systems; XML is an emerging standard
4. What are different types of standards?
 - Product standards: apply to the product being developed; including design review from template, requirements document structure, documentation standards, coding standards to follow, project plan format and change request from template
 - Process standards: define the processes that should be followed during software development, including design review conduct, design validation process, version release process, project plan approval process, change control process and test recording process
5. What are the advantages and problems of standards?
 - Advantages: provide a framework around which the quality assurance process may be implemented; provide encapsulation of best or most appropriate practice; customers sometimes require a particular quality standard when choosing vendor
 - Problems: not seen as relevant and up-to-date by software engineers; involve too much bureaucratic form filling; unsupported by software tools so tedious manual work is involved to maintain standards
6. What are examples of software standards and systems?
 - ISO 9000: it sets out general quality principles, describes quality processes in general, and lays out the organisational standard and procedures that should be defined; must be documented in an organisational quality manual; most companies decide to go for ISO 9000 certification because their customers demand it

- Capability maturity model (CMM): it describes the key elements of an effective software development process; describes an approach for software companies to move from an ad-hoc and immature process to a mature developed process; organisations are characterised being at a level from 1-5 based on the processes they follow
- CMM and ISO 9000 share the common concern with quality and process management
- ISO 9001 compliant organisation would not necessarily satisfy all level 2 key process area, and it would satisfy most of level 2 and many level 3 goals
- It is possible for level 1 organisation to receive 9001 registration
- A level 3 organisation would have little difficulty in obtaining ISO 9001 certification, and a level 2 organisation would have significant advantages in obtaining certification

7. What are the five levels of CMM?

- Initial: The software process is characterized as ad hoc and occasionally even chaotic. Few processes are defined, and success depends on individual effort
- Repeatable: Basic project management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications
- Defined: The software process for both management and engineering activities is documented, standardized, and integrated into a standard software process for the organization. Projects use an approved, tailored version of the organization's standard software process for developing and maintaining software
- Managed: Detailed measures of the software process and product quality are collected, analysed, and used to control the process. Both the software process and products are quantitatively understood and controlled
- Optimizing: Continual process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies

8. What are the components of a software quality assurance plan?

- Product overview: description of product, intended market, quality expectation
- Product plan: critical release dates and responsibilities
- Quality goals: identification and justification of critical product quality attributes
- Process description: quality assurance processes such as reviews and audits
- Document and coding standards
- Risks and risk management: could provide a link to risks in Risk Management Plan

9. What is verification, validation and review in quality assurance?

- Verification: an attempt to ensuring you are building the system in the right way; whether the output products of an activity meet the specifications imposed on them in previous activities; involves two artefacts: requirement specification vs design; design vs code

- Validation: an attempt to ensure the right system is built to fulfil intended purpose and meet stakeholder needs; involves going back to stakeholders to check if the product meets their requirements; normally involves something or someone external
- Review: a common technique used for verification and validation; artefacts produced during development process are reviewed to identify problems and improve them early

10. What are common types of review?

- Technical reviews: review of artefacts is performed by peers in the development team, but authors are involved; aims to uncovering problems and improve them; considered as a “soft” method because nothing is executed
- Business reviews: ensuring the IT solution provides functions specified in the scope and requirements document; can include all project deliverables to ensure that it is complete, providing information needed to move to next phase or process, and meets the standards
- Management reviews: compares the project’s actual progress against a baseline project plan; project manager is responsible for presenting project progress and current status; may involve reviewing scope, schedule, budget and quality objectives; issues need to be resolved. E.g. resources reallocated as needed, change to the project course if needed

11. What are the advantages and disadvantages of technical reviews?

- Advantages: can be performed on any software artefact, while testing and measurement can only be performed on executable artefacts; earlier detection of problems leads to lower costs for resolution; actual programming faults can be found during source code reviews; more mistakes during testing due to release pressure
- Disadvantages: time and resource consuming; should be carefully planned and executed to get desired outcomes

12. What are the types of technical reviews?

- Informal reviews: simple desk check or casual meeting; no formal guidelines or procedures; using checklists is more effective; less time and resource consuming
- Formal reviews: meeting with multiple stakeholders including developers, testers and client; can bring out different perspectives; 3-5 well-chosen members, including review leader, author, reviewers and recorder; lasts no longer than 90 minutes; the results can include accept without further changes, accept with proposed changes and rejection
- Walkthroughs for code or document: the author leads a group of reviewers; reviewers do not need preparation; possible solutions are discussed when defects are found
- Code inspections: similar to formal reviews except that the focus is on the code
- Audits: including product audits and process audits; used to determine whether a particular product or process conforms to standards; the authors are not involved in; typically performed by a team that is external to the organisation

13. What is the QA process in Agile?

In Agile, there are separate testings done by the Agile team in a number of sprints after the development. Sprint review QA evaluation steps include:

- Build small piece of working software with minimal features
- Showcase the product chunk to the stakeholders early
- Fail fast and as cheaply as possible, and get timely feedback
- Capture the technical debt item in the Product Backlog: the format can vary, optionally in feature driven development (FDD) format
- The product owner sets the priority of the technical debt item

Ethics, Procurement, Outsourcing & Contracts

1. What is ethics and why is it important?

- Organisational ethics express the values of an organisation to its employees and other entities irrespective of governmental or regulatory laws
- Individual ethics are principles and values used to govern actions and decisions
- Being fair, honest and ethical satisfies basic human needs
- It creates credibility: an organisation driven by moral values is respected
- It is the common thread that links people and leadership
- It sets the basis for decision making
- Organisations guided by ethics and values last profitable in the long run

2. What are included in Australian Computer Society Code of Ethics?

- Place public interests above personal, business and sectional interests
- Enhance the quality of life of those affected by the work
- Be honest in the presentation of skills, knowledge, services and products
- Work competently and diligently for stakeholders
- Enhance own professional development
- Enhance the integrity of ACS and respect other members

3. What are the stages of procurement management process?

- Plan: analyse business needs – analyse & engage market – finalise procurement strategy
- Source: approach the market – select – negotiate & award
- Manage: implement arrangement – manage arrangement - renew

4. What is RFx in procurement process?

- RFx (request for x) encompasses the entire formal request process and can include any of the following: request for bid (RFB); request for information (RFI); request for proposal (RFP); request for quotation (RFQ); request for tender (RFT)
- The RFx document is prepared by the buyer and has specific information depending on what it is (RFI, RFP, RFT/Q). It will typically include purpose of RFx, organisation's background, basic requirements, hardware and software environments, description of RFx processes and evaluation, statement of work and scheduled information, appendices (current systems overview, system requirements, volume and size data, required contents of vendor's response to RFx, sample contract)
- A key component of the RFx document is a detailed statement of work (SOW) established by analysing business needs. It is a description of the required work, gives bidders an understanding of buyer's expectations. Key items include scope of work to be completed; location of the work to be completed; measurement and performance criteria; deliverable, milestones and schedule; applicable standards and acceptable criteria; any special requirements

5. What are the steps of sourcing procurements?

Approach the market, select, negotiate and award:

- Deciding whom to ask and potentially do the work
- Sending appropriate documentation to potential sellers/bidders
- Obtaining proposals/bids
- Evaluating responses and selecting a preferred supplier: evaluation team review of RFx response and evaluate against predetermined criteria; schedule shortlisted vendor presentations; check vendor preferences; shortlisted vendor presentations; evaluation team site visits to shortlisted vendors/references; evaluation team finalises evaluation and selects shortlisted firms; best and final offer (BAFO) with shortlisted firms; conduct final negotiation with preferred supplier
- Negotiating the contract
- Awarding a contract

6. What are the steps of managing procurements?

Implement, manage and renew:

- Implement the agreement and services according to contract and SOW
- Manage the agreement to ensure the seller's performance meets contractual requirements
- Review and control all changes: project managers and team members should watch for constructive change orders. If change is requested, contractor can legally bill the buyer for additional work
- Renew/closing procurements involves completing, setting contracts and resolving issues; the contract should include requirements for formal acceptance and closure; the project team should determine whether all work was completed correctly and

satisfactorily, resolve any issues or outstanding items, archive information and update record to capture all lessons and knowledge learned

7. What are the principles of change control?
 - Changes in any part of the project need to be reviewed, approved and documented
 - Evaluation of any change should include an impact analysis including time, cost, scope and quality
 - Changes must be documented and signed off by appropriate delegates (business, procurement, finance)
8. What is outsourcing and why is it used?
 - Outsourcing is the practice of engaging an external party to perform services or create goods that are traditionally performed in-house by company's own employees
 - Various activities are suited including architecture, change management, project management, business analysis, design, software development testing and operational support (application & infrastructure)
 - It is used for accessing a broader skills base at a lower cost
9. What are different types of outsourcing?
 - Onshoring: relocating activities inside national borders to access targeted benefits
 - Nearshoring: activities relocated to another country with close proximity
 - Offshoring: activities relocated to another country irrelevant of geographical location and time zones
10. What are the advantages and disadvantages of outsourcing?
 - Advantages: reduces costs; difficult capabilities and skills can be accessed; saves time; freeing scarce internal resources for core business activities; leverage best practice; access to better technology; lower training costs in high turn-over jobs; more flexible; increases accountability by contracts; access established and proven risk mitigation approaches such as Agile, project management
 - Disadvantages: loss of control; process and supply chain is fragmented; security issues; employees feel threatened; additional effort and cost to engage and manage; lower quality work; time zone, cultural and language challenges; location stability; ethical standards such as environment and child labour; difficult to change; damages to local job markets; loss of opportunity for building relationship with key stakeholders
11. What is a contract?
 - A mutual binding agreement that obligates the seller to provide specified products or services, and obligates the buyer to pay for them
 - A document that clarifies responsibilities and sharpens focus on key requirements such as deliverables, quality and timeframes

- A legally binding document that increases accountability for delivering the specific work as stated in the contract
- A document that must be detailed accurately as they are used as the final position
- Rarely used or relied on and seen as a last point of call
- One source of truth for all activities that are to be delivered by external parties

12. What are different types of contracts?

- Fixed price: involve a fixed total price for a well-defined product or service; requires a high level of accuracy in the scope; quality may be compromised; new changes lead to price negotiations
- Time & material: involve payment to the seller for actual time spent and any materials used in providing the service; deep involvement required to effectively and efficiently achieve the outcomes
- Cost plus: involve payment to the seller for direct and indirect costs; usually an additional payment (10%-25% profit/incentive) is also paid upon final delivery; suppliers may become complacent and not efficient

13. What should be included in contracts?

- Intellectual property ownership and indemnities
- Milestones and deliverables
- Quality criteria / performance and acceptance testing
- Variations to scope, time and costs – defined change request process
- Non-performance/termination – convenience, breach
- Disengagement and transition
- Liquidated damages
- Fees and penalties
- warranties

14. What is the change control process in Agile?

- Build small piece of working software quickly with minimal features
- Showcase the product chunk to the stakeholders early
- Fail fast and as cheaply as possible, and get timely feedback
- Capture the fix of the failed expectations as a new user story in product backlog
- The product owner sets the priority of the fix

Configuration Management

1. What is software configuration management and why is it useful?

- Configuration management (CM) is concerned with the policies, processes, and tools for managing changing software systems
 - The aims of CM include identifying all items that collectively will make up the configuration; managing changes to one or more of these items so that the collection remains consistent; managing different versions of the product; assuring software quality as the configuration evolves over time
 - If no effective configuration management is place, changes to artefacts may leave the configuration in an inconsistent state. E.g. One module may be changed without changing its dependent modules, leaving the configuration inconsistent
2. What are the tasks of configuration management?
 - Identification the items necessary for the projects (basic – aggregate – derived)
 - Version control: processes and tools are chosen to manage the different versions of configuration items as they are developed; a common method of tracking versions in a repository is version numbering (reviewed version vs un-reviewed version)
 - Change control: change management plan is used to control the changes. Steps include initiating the change, evaluating the change and making the change
 - Configuration auditing: assuring what is in the repository is actually consistent and all changes have been made properly
 - Configuration reporting: reviewing the configuration objects for consistency with other objects, to find any omissions or potential side effects; most common aim is to report on the status of the configuration items of interest and baselines that have been achieved
 3. What are typical configuration items?
 - Requirements specifications and sections, requirement models, individual requirements
 - Use cases, user stories
 - Design models, design documents, design elements, and class designs
 - Source code modules
 - Object code modules
 - Release modules
 - Software modules
 - Test drivers and stubs, and test scripts
 - Documents or sections of documents associated with the project
 4. What should be included in a change record?
 - The change made
 - Who made the change?
 - The rationale for the change
 - When it was implemented?
 5. What are the requirements for a version control system?

- A repository/configuration database for storing configuration items. It includes version (different functions), variant (same function but other differences) and release (distributed to users outside of development team)
- A version management function that allows software engineers to create and track versions and roll the system back to previous versions if necessary. E.g. git, svn, cvs
- A make-like facility that allows engineers to collect all of the configuration objects for a particular target together and to build that target. E.g. Apache Maven, Apache Ant, make (Unix, Linux)

6. What is a baseline?

- An artefact that is stable
- It has been formally reviewed and agreed upon, that is now ready for future development
- It can only be changed through a formal change management procedure

7. What are typical questions for a configuration audit?

- Have the changes requested and approved been made?
- Have any additional changes other than required by request been made?
- Did the configuration objects that were changed pass their quality assurance tests?
- Do the objects in the configuration meet the required external standards?
- Do the attributes of the configuration item match the change?
- Does every configuration item have appropriate change logs?

8. What is Git and how it works?

- Git is a commonly used version control system made by Linus Torvalds
- It is distributed, so everyone has their own code repository local to them; open source
- General workflow: git init (initializes local repository called master) – git add (track changes made to the content) – git commit (save the file in local repository) – git push (save the collection)
- Git clone: initialises an empty local repository and fetches code from remote repository into your local repository
- Git status: shows the status of the local repository, and files to be added, modified and untracked files
- Git add/rm: adds or removes a file from workspace to local repository; files in local repository are called staged; stages files have their contents monitored
- Git log: shows the history of commits to the system