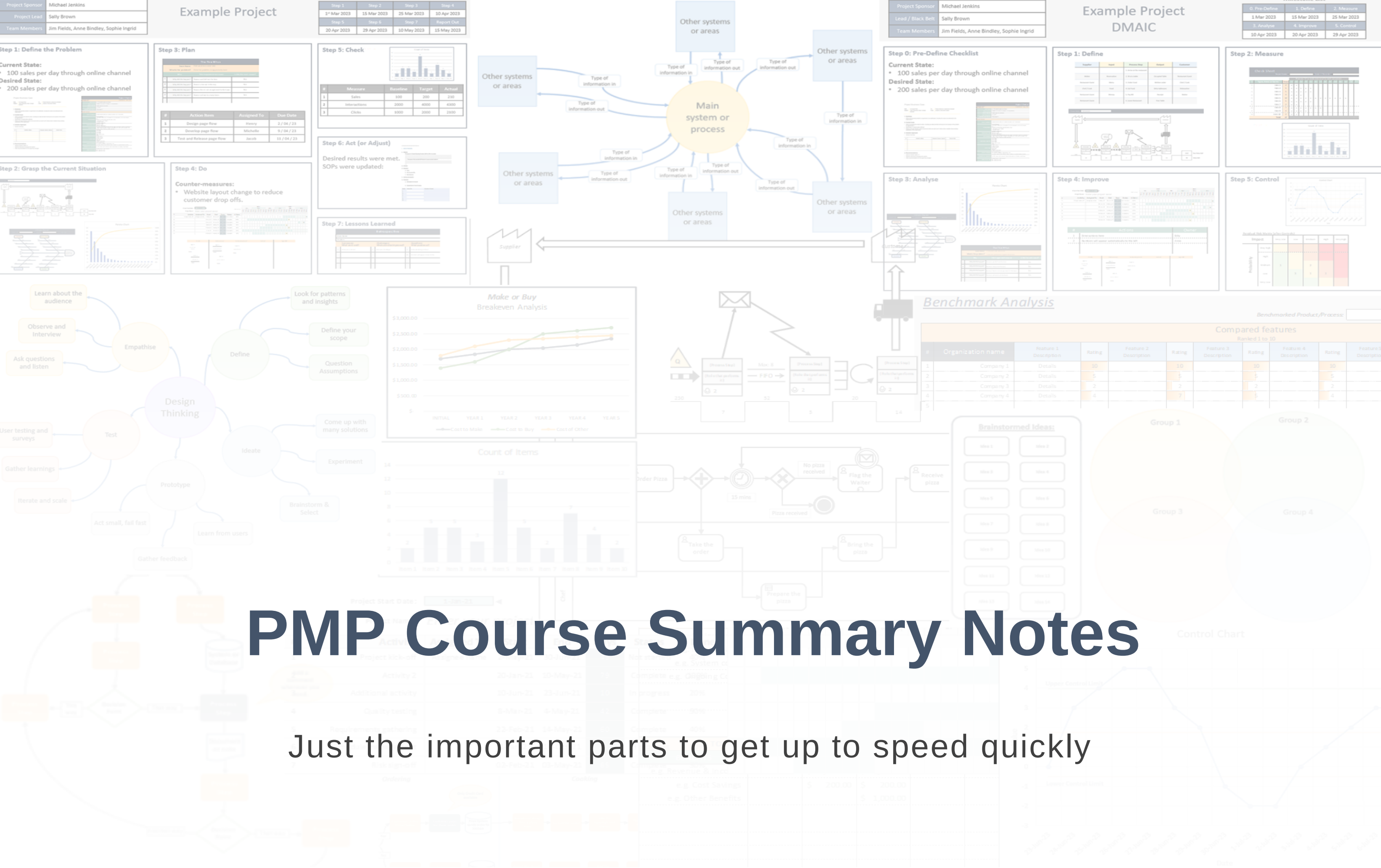


PMP Course Summary Notes

Just the important parts to get up to speed quickly

Collated by David McLachlan

Contents

Introduction

Welcome to the PMP Course Summary Notes. I am so glad you are a part of this project management education, and I hope it helps you achieve everything you want.

This course covers all the essential, current Project Management Topics from the Project Management Body of Knowledge (7th Edition), the Process Groups Practice Guide, and the Agile Practice Guide.

These **course summary notes** summarise all that information for you so you can easily review it as you go through the course, and again before your exam.

It also adds **Project Manager Mindset** summaries to help you pass your exam, for both Traditional and Agile projects.

When you combine these notes with the Video Content and 500 Practice Questions to really help you embed the knowledge, and you practice a little bit every day, I truly believe you can pass your PMP and gain the prestige and career recognition you deserve.

I look forward to hearing about your success!

Kind regards,

David McLachlan

PMP Exam Mindset

You may have heard about the Project Manager's mindset. The PM mindset idea was pioneered by Andrew Ramdayal, but has always been a part of being a good Project Manager and passing the PMP exam. The items below also make great general tips for the exam.

- When answering Exam questions, always assume the information you need to answer the question is included in the question.

- There will often be two or more possible answers: **Prioritise the answers from best/most likely to worst/least likely** based on the information provided.

- **Avoid extreme actions in your answers**, such as closing a project, escalating to the Sponsor or other external party, not helping the customer, or answers that say you "must", or "only".

- Aim for answers that have inclusiveness and collaboration, or an action by the Project Manager.

- Don't rush into action in your answers – first:
    1. Assess and analyse the situation (i.e. review the plan or brainstorm with your team) to understand the Root Cause of the issue.
    2. Review your plan
    3. Meet with stakeholders to work out a solution
    4. Take Action

- Don't skip project steps, processes or documents to accelerate project delivery or closure.

- Don't pass the problem to someone else, or escalate to management unless it's a last resort.

- The Project Manager is always proactive, puts the team first as a servant leader and is a good steward for the business / customer's goals.

- There may be different types of scenarios, that ask:
    - **What should you do?** Normally an action.
    - **What should you do first / do next?** Normally assessing the situation or reviewing a document.
    - **What should have been done?** Look for what could have been done to prevent it.
    - **What should you not do?** Look for the worst option.

- Dealing with Resource Requirements: work with the **Functional Manager** of that area.

- **Risks** are things that might occur in the future, **Issues** have already occurred.
    - When a Risk becomes an Issue, manage it with the response already outlined in your Risk Register.

- When guidance is needed, review the process in the relevant plan (i.e. Scope Management Plan). Overall guidance is in the Project Management Plan.

- Determine the source of any conflict before acting. Discuss issues with your team, using their **Expert Judgement**.

- Figure out what framework you are in: Predictive (Waterfall), Adaptive (Agile), Incremental, Iterative or Hybrid, then answer accordingly. Sometimes it says it explicitly, other times you have to guess (**it might include keywords** such as Iterations or Sprints, Product Owner, Retrospective, etc.

Traditional Project Mindset

- The Project Manager assigns the work, based on the skills of the team and the work required to be completed.

- **Stakeholder identification** and analysis is done throughout the project, not just at the beginning. This includes analysing, agreeing and tailoring **communication needs**. Engage and communicate often (or as outlined in your Stakeholder or Communication Plan).

- **Virtual Teams** will use video conferencing, accommodating time zones and cultural differences. The preference is for **co-location** where possible.

- Any stakeholder who wants to make a change to a baselined item (i.e. the approved Scope, Schedule or Budget) will need to submit a **change request**.

- Change requests are analysed for their impact to Scope, Schedule, Cost or Quality and brought to the **Change Control Board (CCB)** for approval. This process (including members of the CCB) is decided in your **Change Management Plan**.

- If you are unsure, ensure your decisions ultimately **deliver business value** and benefit the objectives of the project. This includes any decision, such as conflicts between team members. Always revert to business value.

- For any problem or conflict, find and understand the **root cause** before taking action.

- **Your job is to integrate** the many parts of a project (Scope, Schedule, Cost, Resources, Quality). Do not focus on one aspect at the expense of others.

- **Your team are the best people to break down the work**, estimate on its effort or time, and determine when an activity should happen.

- **Your project customer is the best person (or group of people) to determine Quality requirements**, and ensure your product meets those requirements.

- Identify and **manage risk** throughout your project, not just at the beginning. A negative risk is a Threat, a positive risk is an Opportunity (identify both).

- **Risks** are items that could happen in the future, **Issues** are items that have happened and we manage them now.

- When **selecting a contract** for a vendor, ensure it benefits both the seller, the buyer and the objectives of the project.

Agile Project Mindset

- The team **self-organises** and self-assigns work.

- The Project Manager should be an **advocate** for Agile principles and methods.

- Always use the **Servant Leadership** style to grow and develop your team, and remove obstacles or blockers. Collaborate, don't dictate.

- The **Product Owner** represents the business or the customer, and **prioritises the features in the Product Backlog** based on the highest business or customer value. Any changes in scope go through them.

- **Co-locate** your team to ensure fast communication and learning from conversations around them (osmosis).

- Always prefer **face-to-face communication**, as it is the richest form of communication (with body language, facial cues, voice tonality, along with the words themselves).

- Use an **Information Radiator** in your team area to showcase project information openly to anyone who needs it. This can be charts on walls (physical or virtual) including the Kanban Board, Burndown or Burnup charts, Remaining features, Releases, Risks etc.

- When problems occur, **problem solve** with the team, coach and support them to a resolution.

- Ensure **psychological safety** – where your team can fail or experiment without being punished, raise concerns or ideas and discuss things openly.

- Reduce or **remove multi-tasking** where possible to promote focused work instead, by **limiting WIP (Work in Progress)** and solving bottlenecks. You can see these on your **Kanban Board** where User Stories start to pile up.

- Create a **Team Charter** with input from your team to decide ways of working, ensure everyone understands the mission and what success looks like.

- Fast feedback loops are important. Use **Retrospectives** to raise and solve issues in the team's way of working.

- Use Demonstrations or the **Sprint Review** to gather feedback on the product from the customer.

Process Groups Practice Guide (Previously PMBOK Guide – 6th Edition)

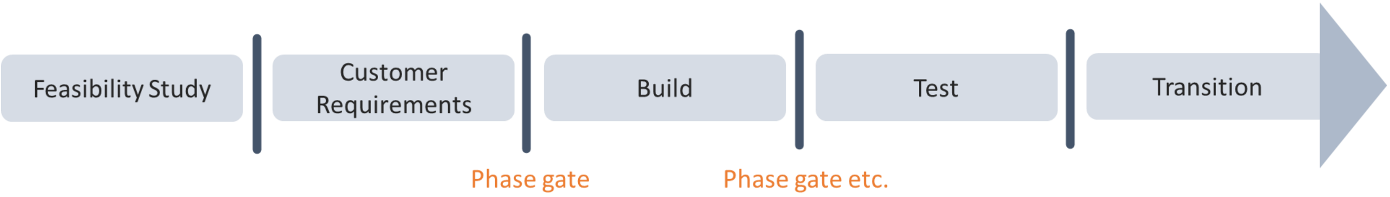
The Process Groups Practice Guide is a wonderful in-depth, step-by-step process for managing projects.

Project Foundations

- A **Project** is **temporary** (has a beginning and end) and **delivers change**, in the form of business or customer value.

Types of Lifecycles:

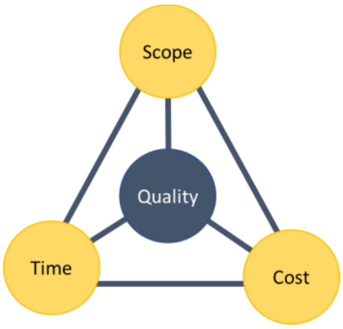
- Your **Project Lifecycle** is the **phases** of your project i.e.



- These may have **phase gates**, where we review performance against the project objective and either proceed, change or stop the project.
- Your **Development Lifecycle** is how you are delivering (Predictive, Iterative, Incremental, Agile, Hybrid)

Development Lifecycles:

Approach	Requirements	Activities	Delivery	Goal
Predictive	Fixed	Performed once for the entire project	Single delivery	Manage cost
Iterative	Dynamic	Repeated until correct	Single delivery	Correctness of solution
Incremental	Dynamic	Performed once for a given increment	Frequent smaller deliveries	Speed
Agile	Dynamic	Repeated until correct	Frequent small deliveries	Customer value via frequent deliveries and feedback



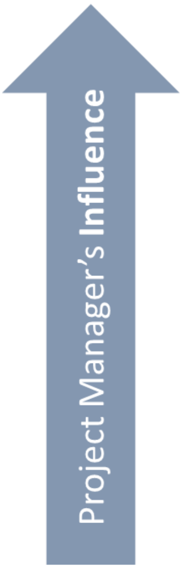
The project “**Triple Constraint**” includes Scope, Cost, and Time (and ultimately Quality).

Changes to one of these will affect all the others, so be aware when making changes.

Types of Managers:

- **Project Manager:** Assigned by the performing organisation to lead the team responsible for delivering business value.
- **Functional Manager:** Manages and provides resources from a Functional or Business unit.
- **Operations Manager:** Responsible for ensuring business operations are efficient.

Project Manager’s Authority in different structures (starts high and gets lower):



Structure Type	Resource Availability	Budget Managed by
PMO	High to Total	Project Manager
Project-oriented	High to Total	Project Manager
Matrix – strong	Moderate to High	Project Manager
Matrix – balanced	Low to Moderate	Mixed
Hybrid	Mixed	Mixed
Virtual	Low to moderate	Mixed
Matrix – weak	Low	Functional Manager
Multi-divisional within a project	Low	Functional Manager
Functional (centralized)	Low	Functional Manager
Organic or Simple	Low	Owner or Operator

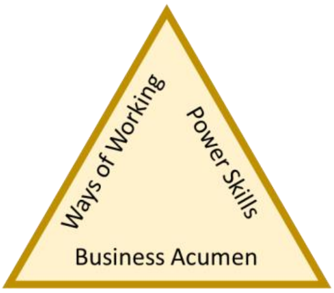
Types of Project Management Office (PMO):

PMO Type	Authority and Control
Directive	• Takes control of the projects by directly managing the projects. • Project managers are assigned by and report to the PMO. • **High** degree of control.
Controlling	• Provides support and compliance through adoption and conformance to methods or frameworks, and the use of specific templates, forms and tools. • **Moderate** degree of control.

Supportive	• Provides a consultative role to projects. Serves as a project repository. • Supplies templates, best practices, training, and information from other projects. • **Low** degree of control.

Project Manager Competences (PMI Triangle):

Ways of working	Understanding the many methods available to get the job done (Agile, Waterfall etc), tailoring your method to suit the environment.
Power Skills	Soft skills, collaborative leadership, effective communication.
Business Acumen	Understanding priority (cost versus benefit), making good business decisions, continuous learning.



Project Process Groups and Knowledge Areas:

Knowledge Areas	Initiating	Planning	Executing	Monitoring & Controlling	Closing
Project Integration	1. Develop Project Charter	2. Develop Project Management Plan	3. Direct and Manage Project Work 4. Manage Project Knowledge	5. Monitor and Control Project Work 6. Perform Integrated Change Control	7. Close Project or Phase
Stakeholders	1. Identify Stakeholders	2. Plan Stakeholder Engagement	3. Manage Stakeholder Engagement	4. Monitor Stakeholder Engagement	
Scope		1. Plan Scope Management 2. Collect Requirements 3. Define Scope 4. Create WBS		5. Validate Scope 6. Control Scope	
Schedule		1. Plan Schedule Management 2. Define Activities 3. Sequence Activities 4. Estimate Activity Durations 5. Develop Schedule		6. Control Schedule	
Cost		1. Plan Cost Management 2. Estimate Costs 3. Determine Budget		4. Control Costs	
Quality		1. Plan Quality Management	2. Manage Quality	3. Control Quality	
Resources		1. Plan Resource Management 2. Estimate Activity Resources	3. Acquire Resources 4. Develop Team 5. Manage Team	6. Control Resources	
Communications		1. Plan Communications Management	2. Manage Communications	3. Monitor Communications	
Risk Management		1. Plan Risk Management 2. Identify Risks 3. Perform Qualitative Risk Analysis 4. Perform Quantitative Risk Analysis 5. Plan Risk Responses	6. Implement Risk Responses	7. Monitor Risks	
Procurements		1. Plan Procurement Management	2. Conduct Procurements	3. Control Procurements	

Project Integration

#	Process	Process Group	Main Output
1	Develop Project Charter	Initiate	Project Charter
2	Develop Project Management Plan	Plan	Project Management Plan
3	Direct and Manage Project Work	Execute	Project Deliverables, Work Performance Data
4	Manage Project Knowledge	Execute	Lessons Learned Register
5	Monitor and Control Project Work	Monitor & Control	Work Performance Reports
6	Perform Integrated Change Control	Monitor & Control	Approved Change Requests, Updated Project Plan
7	Close Project or Phase	Close	Final Report

The key benefits of Project Integration are to ensure we deliver business value that is aligned with the organisation's strategic objectives.

We do this with a well-researched **business case**, initiated with an approved **Project Charter**, and managed with a well-planned **Project Management Plan** to ensure our stakeholders, scope, schedule, cost and quality work together to deliver business value.

Documents Comparison:

Business Case	Benefits Management Plan	Project Charter	Project Management Plan
A feasibility study – asks do we need a project? Includes: • The business or customer need, • Analysis of the situation, • Solution options, • Recommendation.	The Benefits Management Plan describes how and when the project benefits will be delivered, and how we will measure those benefits.	**Formally authorizes** the existence of a project. Provides the project manager with the authority to **apply organisational resources** to the project. Issued by the project initiator or **project sponsor**.	Our plan for managing the project. Includes all smaller plans (Scope, Schedule, Cost, Quality etc). Includes project baselines (original plans).

Project Logs Comparison:

Change Log	Issue Log	Assumption Log
The **Change Log** notes changes requested to baselined items, and their status.	The **Issue Log** notes issues affecting the project, their status and who is responsible for resolving them.	The **Assumption Log** notes any assumptions or dependencies we've made – especially when estimating.

Other Plans Comparison:

Configuration Management Plan	Change Management Plan
The **Configuration Management Plan** outlines which parts of your project will be **baselined** (i.e. Scope, Schedule, Budget etc), and their current versions.	The **Change Management Plan** outlines your **process** for raising, managing and approving **Changes** to any baselined items in your project.

- Our **project baseline** is the approved version of our project plan, most often centred around the Triple Constraints:
  - Scope baseline
  - Schedule Baseline
  - Cost Baseline
  - Performance Measurement Baseline (what we measure project performance against)

- **Changes to baselined items require a Change Request**, and **can be requested by any Stakeholder**. The steps are:
  - ➢ Raise a change request
  - ➢ Analyse the impacts to Scope, Schedule, Cost
  - ➢ Note it in the Change Log
  - ➢ Go to the Change Control Board for approval, deferral or rejection
  - ➢ The outcome is communicated back to the stakeholder and updated in the Change Log.

- Avoid **Scope Creep,** which is when we or our stakeholders add Scope or features that were not planned for and not approved. This impacts our Cost, Schedule.

- Avoid **Gold Plating**, which is when we deliver additional features that the customer did not need or ask for.

- **Work Performance:**
  - **Work Performance Data** is the raw project activities, i.e. Number of defects or Change Requests, Actual Cost.

  - **Work Performance Information** is the comparison of two or more bits of data to make it meaningful i.e. number of Work Packages completed each week, Schedule Variance.

  - **Work Performance Reports** is the physical or electronic representation of Work Performance Information, intended to raise actions or awareness (i.e. Memos, status reports, electronic dashboards).

- To close your project or phase:
  - All procurement documentation is collected and filed, all bills are paid
  - Ensure lessons learned are gathered, archived and project Resources are released.
  - Archive all project information for future use, into the Organisational Process Assets.
  - **The Final Report** provides a summary of the project performance, and whether the objectives were met.

Stakeholder Management

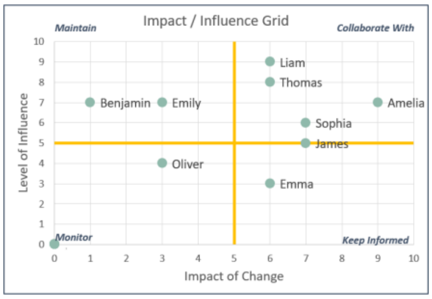
#	Process	Process Group	Main Output
1	Identify Stakeholders	Initiate	Stakeholder Register
2	Plan Stakeholder Engagement	Plan	Stakeholder Management Plan
3	Manage Stakeholder Engagement	Execute	Change Requests
4	Monitor Stakeholder Engagement	Monitor & Control	Work Performance Information Change Requests & Project Document updates

- Identify and analyse the engagement of your Stakeholders early and throughout your project. You can use an **Organisational Breakdown Structure** of the organisation to find initial stakeholders.

- Use a **Stakeholder Register** to keep your Stakeholders':
  - Names
  - Role on the project
  - Power / Interest / Influence
  - Expectations or needs
  - **Communication** Requirements

- Use a **Stakeholder Engagement Assessment** to determine your Stakeholders' level of engagement with the project with **C** for Current and **D** for Desired:
  - Unaware
  - Resistant
  - Neutral
  - Supportive
  - Leading

- Use **Stakeholder Mapping** to determine your Stakeholders' Influence on the project, and Impact by the project. You can also use:
    - Power / Interest
    - Interest / Influence
    - **Salience Model:** Power / Urgency / Legitimacy



- Any model with three inputs (like the Salience model) is a 3D or Cube matrix.

- **Directions of Influence** from stakeholders on your project include:

⬆	Upward:	Senior management, Sponsor, Steering committee.
⬇	Downward:	Project team, or specialists providing knowledge or skills to the project.
⬅	Sideward:	Peers of the Project Manager, other middle managers in charge of resources.
↗	Outward:	Suppliers, users, government departments.

- **Communication Channels** Calculation (e.g. for a team with **10** people):
    - N x (N – 1) / 2
    - 10 x 9 / 2
    - 90 / 2 = 45

Scope

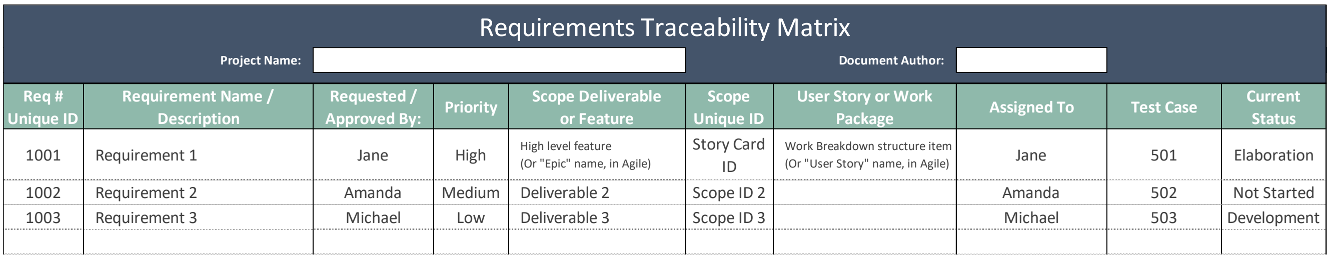
#	Process	Process Group	Main Output
1	Plan Scope Management	Plan	Scope Management Plan, Requirements Management Plan
2	Collect Requirements	Plan	Requirements Traceability Matrix
3	Define Scope	Plan	**Project Scope Statement:** • Scope Description • List of Deliverables • Acceptance Criteria • Specific Exclusions
4	Create WBS	Plan	**Scope Baseline:** • Scope Statement • WBS • WBS Dictionary • List of Work Packages
5	Validate Scope	Monitor & Control	Accepted Deliverables, Work Performance Information, Change Requests
6	Control Scope	Monitor & Control	Work Performance Information, Change Requests

- **For Predictive projects**: Scope is planned at the beginning and managed for changes with Change Requests throughout delivery.

- **For Agile projects:** The Product Owner determines and prioritises features in the Product Backlog and this can (but doesn’t always) change each Sprint or Iteration (usually of 2 – 4 weeks).

- The **Scope Management Plan** is our **process** for how we will gather, manage and control the scope.
- The **Requirements Management Plan** (or Business Analysis Plan) is our defined **process** for how we will gather, manage and control our Requirements.

- Use a **Requirements Traceability Matrix** to trace the customer requirements to the Project Scope, approvals or sign-offs, acceptance criteria and tests, and ultimately completed deliverables.



Requirements Traceability Matrix

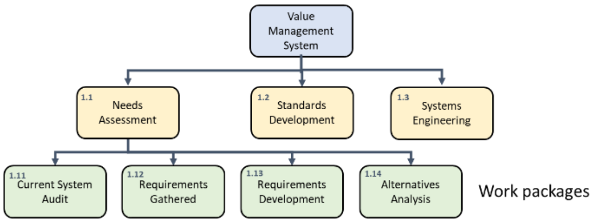
Project Name: Document Author:

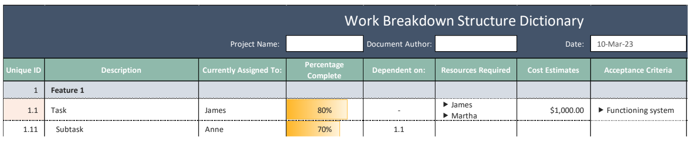
Req # Unique ID	Requirement Name / Description	Requested / Approved By:	Priority	Scope Deliverable or Feature	Scope Unique ID	User Story or Work Package	Assigned To	Test Case	Current Status
1001	Requirement 1	Jane	High	High level feature (Or "Epic" name, in Agile)	Story Card ID	Work Breakdown structure item (Or "User Story" name, in Agile)	Jane	501	Elaboration
1002	Requirement 2	Amanda	Medium	Deliverable 2	Scope ID 2		Amanda	502	Not Started
1003	Requirement 3	Michael	Low	Deliverable 3	Scope ID 3		Michael	503	Development

- Use **Decomposition** to break down the work (with your team) from Deliverables or Features into the smallest possible unit – usually Work Packages.

- A **Work Breakdown Structure** is an example of Decomposition and is usually visual.
- A **Work Breakdown Structure Dictionary** contains any additional Scope information for each work package, and is usually a matrix.

- A **Work Package** is a piece of work or functionality small enough to be assigned to a person or team. It goes from: **Scope Statement > WBS > Work Packages > Activity List (**for our schedule)**.**


Value Management System
1.1 Needs Assessment
1.2 Standards Development
1.3 Systems Engineering
1.11 Current System Audit
1.12 Requirements Gathered
1.13 Requirements Development
1.14 Alternatives Analysis
Work packages



Work Breakdown Structure Dictionary

Project Name: Document Author: Date: 10-Mar-23

Unique ID	Description	Currently Assigned To:	Percentage Complete	Dependent on:	Resources Required	Cost Estimates	Acceptance Criteria
1	**Feature 1**						
1.1	Task	James	80%	-	▶ James ▶ Martha	$1,000.00	▶ Functioning system
1.11	Subtask	Anne	70%	1.1			

Different Scope Processes:

Validate Scope	Control Scope	Control Quality
The Project Sponsor or Customer signs off on the completed deliverables for final acceptance or “Validation”.	Managing changes to the Scope baseline.	Checking the testing results to ensure the product meets its requirements.

Schedule

#	Process	Process Group	Main Output
1	Plan Schedule Management	Plan	Schedule Management Plan
2	Define Activities	Plan	List of Activities, List of Milestones
3	Sequence Activities	Plan	Schedule Network Diagrams
4	Estimate Activity Durations	Plan	Duration Estimates & Basis of Estimates
5	Develop Schedule	Plan	Approved Project Schedule (Schedule Baseline)
6	Control Schedule	Monitor & Control	Work Performance Information, Schedule Forecasts

Predictive (Waterfall) Scheduling: Activities, broken down from Work Packages, are noted in their necessary order on a planned schedule, with changes needing a change request once the schedule is approved and baselined.

Adaptive (Agile) Scheduling: The Product Owner owns the Product Backlog and Product Roadmap, showing at least the order of when features are to be delivered.

Kanban or Pull Systems: The team "pulls" each piece of work when they are ready, respecting "Work in Progress" (WIP) limits to keep multi-tasking to a minimum.

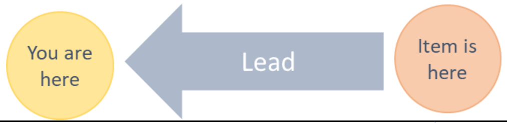
Schedule Definitions:

- The **Critical Path** is the sequence of activities which make up the shortest possible project duration.
- **Free Float** is the amount of time an activity can be delayed.
- **Total Float** is the amount of time the entire project can be delayed.

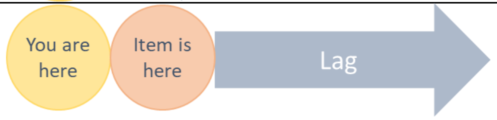
- **Milestones** are dates when major items will occur or be delivered.
- **Rolling Wave Planning** plans near-term items in detail, while further away items are kept at a high level.
- **Project Calendar** shows us the days and shifts that are available for scheduled activities, removing holidays or other days.

Leads and Lags:

- **Lead time** is the amount of time you can bring an item forward. You are *leading* it forward.



- **Lag time** is the amount of time you can delay an item. It is *lagging* behind.

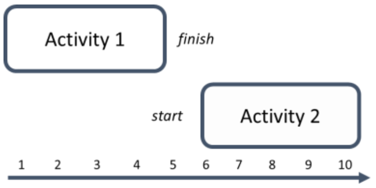


Precedence Diagramming Method:

Shows how activities are linked and the sequence in which they need to be performed.

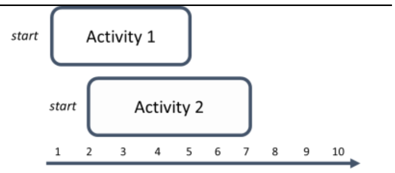
Finish to Start

The next activity cannot **start** until the previous activity has **finished**.



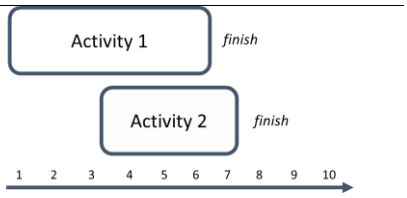
Start to Start

The next activity cannot **start** until the previous activity has **started**.



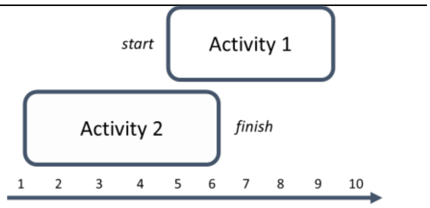
Finish to Finish

The next activity cannot **finish** until the previous activity has **finished**.



Start to Finish

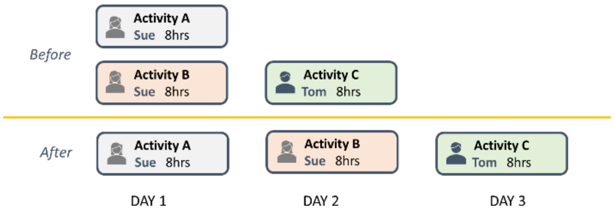
The next activity cannot **finish** until the previous activity has **started**.



Resource Optimisation:

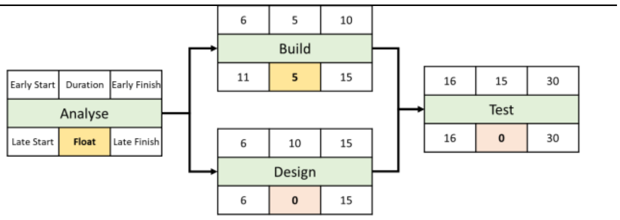
Resource levelling

When shared resources are over-allocated, or a resource has been assigned to two or more activities during the same period.



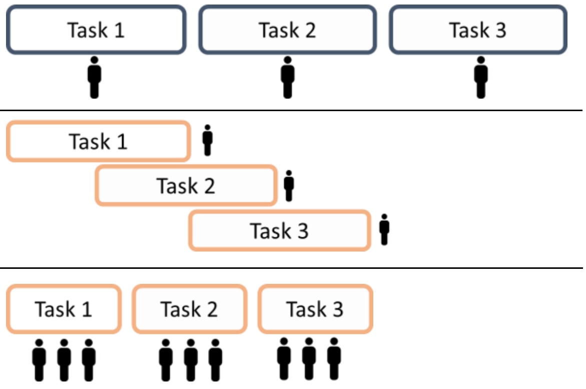
Resource smoothing

Adjusts the activities within their free and total Float **(the Critical Path is not changed** when doing this**)**



Schedule Compression Techniques:

Normal



- **Schedule Fast Tracking** is when activities on the Critical Path are done in parallel (overlapped) to shorten the project duration.

- **Schedule crashing** is approving overtime, adding resources, or paying to expedite delivery of activities on the critical path.

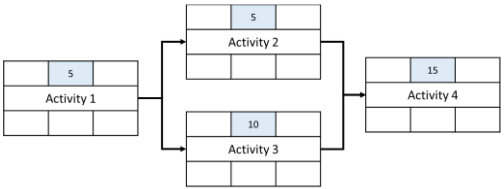
- Remember the **law of diminishing returns**, where adding more resources will only improve your schedule up to a point - the effect reduces the more you add.

Critical Path Method and the Forward and Backward Pass:

Early Start	Duration	Early Finish
Activity Name		
Late Start	**Float**	Late Finish

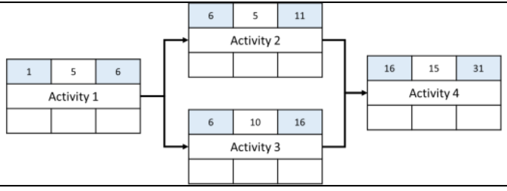
To calculate the Critical Path of our project, we can use the Forward and Backward Pass to determine the amount of free Float (wiggle room) available.

- Enter activity **Durations**



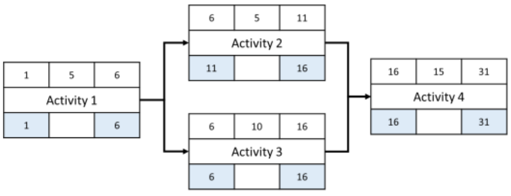
Forward Pass:

- *Calculate **Early Start** and **Early Finish** times:*
- ***EF** = ES + Duration*
- ***ES** = (highest) previous EF*



Backward pass:

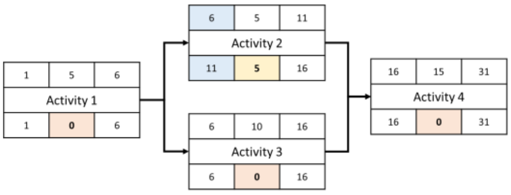
- *Calculate **Late Start** and **Late Finish** times:*
- *Enter highest EF in last box*
- *LS = LF – Duration*
- *LF = (lower) LS*



Calculate Float:

- ***Late Start – Early Start***

- **The critical path is the path with no available Float.**



Estimating Types:

Use **Expert Judgement** (from your team or the people doing the work) to estimate things.

- **Analogous**: Using something **similar** (an analogy) to estimate, like a similar product, project or system. **Faster but lower accuracy**.

- **Parametric**: Using a **parameter** to estimate, like $55 a meter or $100 an hour. **Medium effort, medium accuracy**.

- **Bottom-up**: Adding together the smallest pieces to get an overall estimate (i.e. cost of each work package combined for the project budget). **High effort, high accuracy**.

- **3-point:** An average of three estimates: Optimistic, Most Likely and Pessimistic (O + M + P) / 3. Useful when there are differing opinions.
  PERT (Program Evaluation and Review) or **Beta Distribution** estimation is similar but gives weight to the Most Likely one: (O + (**4 x M**) + P) / 6

- **Wideband Delphi**: Also "**Planning poker**" in Agile – the people doing the work estimate on its effort (or cost). The high and low estimates discuss their reasons, then re-estimate until a consensus is reached. Useful in complex situations.

Cost

#	Process	Process Group	Main Output
1	Plan Cost Management	Plan	Cost Management Plan
2	Estimate Costs	Plan	Cost Estimates & Basis of Estimates
3	Determine Budget	Plan	Cost Baseline, Project Funding Requirements
4	Control Costs	Monitor & Control	Work Performance Information, Cost Forecasts

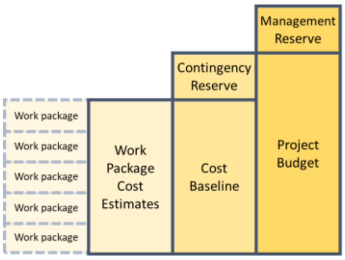
For Predictive Projects, the total budget is created from the total cost estimates of each Work Package in a bottom-up approach, Contingency Reserves are added for Risks, Management Reserves are added for unforeseen scope, and once approved the budget can only be changed with a Change Request.

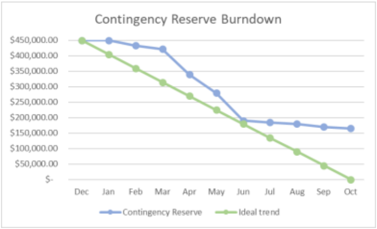
For Agile Projects, The team is fully funded for a set amount of time. That means **Cost and time are fixed**, and Scope is the only thing that changes (via Prioritisation of features by the Product Owner).

The Cost Baseline is made up of all the Work Packages costs (Bottom Up estimating), plus the Contingency Reserve.

Contingency Reserves: Set aside to deal with planned risks, should they occur.

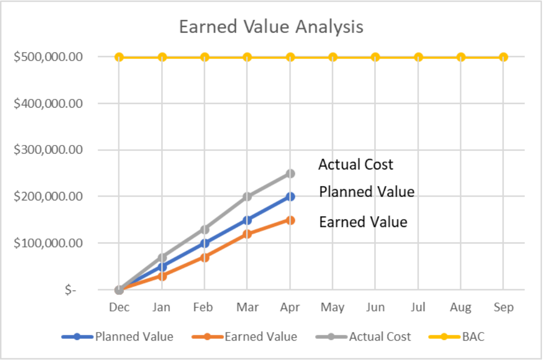
Management Reserves: Set aside for unexpected activities related to in-scope work.
Reserve Analysis: Shows the amount of Risk versus the amount of Contingency remaining.





Earned Value Analysis:

- Budget at Completion: (**BAC**) Is the total planned budget.
- Planned Value: (**PV**) Shows us the work that should have been completed by that time.
- Earned Value (**EV**) Is what we have actually completed (earned) at a given point in time.
- Actual Cost: (**AC**) Is what we have actually spent at that point in time.



Variance Analysis:
Cost Performance Index: (**CPI**) – More than 1 is good, less than 1 is bad. EV / **AC**.
Schedule Performance Index: (**SPI**) – More than 1 is good, less than 1 is bad. EV / **PV**.

Cost Variance: (**CV**) = EV – AC. Positive is good, negative is bad.
Schedule Variance: (**SV**) = EV – PV. Positive is good, negative is bad.

- **Benefits Management Plan**: Use this for the proposed project benefit (usually in the form of money - Revenue)
- **Resource Management Plan**: Use this for resource costs, necessary to estimate the budget.

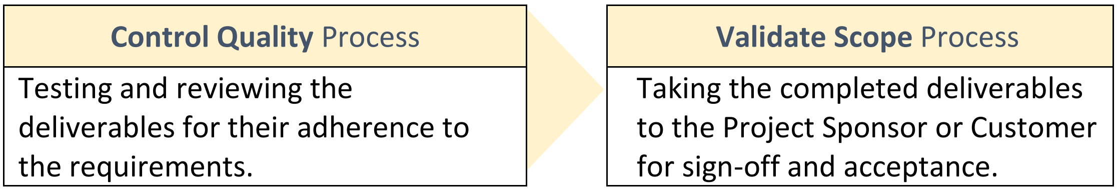
Quality

#	Process	Process Group	Main Output
1	Plan Quality Management	Plan	Quality Management Plan, Quality Metrics
2	Manage Quality	Execute	Quality Reports, Test and evaluation documents
3	Control Quality	Monitor & Control	Verified Deliverables

Quality is how well the deliverables meet the customer requirements.

For Predictive Projects, the product is developed and then tested in sequential order, usually in large batches.

For Agile Projects quality is everyone's responsibility, from writing correct requirements and acceptance criteria, to simple design, regular refactoring (streamlining) of code, test-first approaches (writing the test first, then writing the code that will pass the test), to the team demonstrating a small deliverable in the Sprint Review.



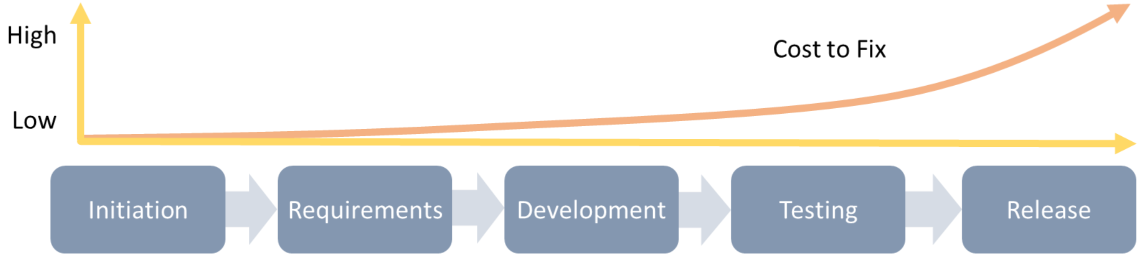
Control Quality Process

Testing and reviewing the deliverables for their adherence to the requirements.

Validate Scope Process

Taking the completed deliverables to the Project Sponsor or Customer for sign-off and acceptance.

Quality is more expensive to fix, the closer it gets to our customer:



The Cost of Quality:

- **Preventative Costs**: The cost to **prevent** future defects or errors in the product (e.g. training, creating prototypes, refactoring / streamlining code).
- **Appraisal Costs**: The cost to **find** defects or errors in the product (e.g. testing, quality audits or inspections, the Sprint Review)
- **Internal Failure**: When the product fails internally, often during testing or review (e.g. defects)

- **External Failure**: When the non-conformance reaches the customer (defects, quality issues, customer complaints and returns).

Quality Methods:

There are existing problem solving methods to use with your team:

Deming or Shewhart Cycle (PDCA):

- **Plan**: Review real data or go and see the problem directly. Find the **root cause**, brainstorm and prioritise solutions with your team.
- **Do**: Put your plan into action as a "pilot" or test in a small area first.
- **Check**: Check the results.
- **Act**: **Or adjust** – respond to the results.

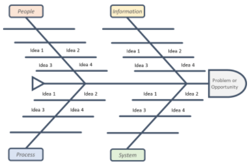
PLAN
DO
PDCA
CYCLE
1
2
3
4
ACT
CHECK

Six Sigma (DMAIC):

- **Define**: Define the problem clearly, as a gap from where you are to where you want to be.
- **Measure**: Measure the current problem with real data.
- **Analyse**: Analyse the data, figure out the root cause, brainstorm and prioritise solutions with your team.
- **Improve**: Put your plan into action and measure the results.
- **Control**: Ensure your new process is under control (not too much variation or defects).

Root Cause Analysis (RCA):

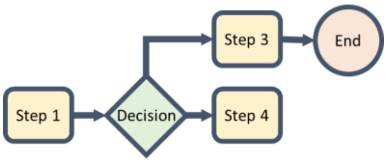
- **Ishikawa Diagrams**, also **Fishbone** diagrams or "**Cause and Effec**t" diagrams. The problem or opportunity is noted at the "head" and we brainstorm causes in predefined buckets (PIPS): People, Information, Process, Systems.



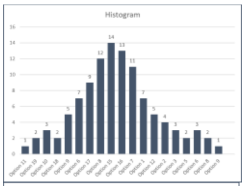
- **The Five Whys**, helps us get to the root cause by asking "why" something happens until we can't go any further (usually the first "why" is just surface level or the symptom).

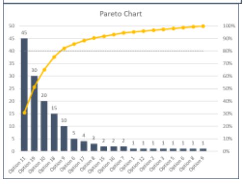
Other Quality Tools:

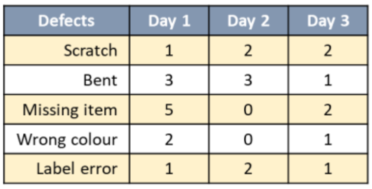
- **Flowcharts** are used to show a process flow, including decision points. Useful for prototyping.



- **Histograms** show us a count of a large amount of data (e.g. defect types, product sales).
- **Pareto analysis** is similar, but sorts them from most to least, and focuses on the few that have the biggest impact.





Checklists	Check Sheets
A list of items, actions or points to be considered, often used as a **process reminder**.	Also “Tally sheets”, used to **count** the number of times something occurs (e.g. defect types). 

Defects	Day 1	Day 2	Day 3
Scratch	1	2	2
Bent	3	3	1
Missing item	5	0	2
Wrong colour	2	0	1
Label error	1	2	1

Grade	Quality
The number of desirable features.	The degree which those features meet the customer’s requirements.

Resources

#	Process	Process Group	Output
1	Plan Resource Management	Plan	Resource Management Plan, Team Charter
2	Estimate Activity Resources	Plan	Resource Requirements & Basis of Estimates, Resource Breakdown Structure
3	Acquire Resources	Execute	Physical resource assignments, Project team assignments, Resource calendars
4	Develop Team	Execute	Team performance assessments
5	Manage Team	Execute	Change Requests, Project Document updates.
6	Control Resources	Monitor & Control	Work Performance Information & Change Requests

Agile teams prefer to be co-located so you can learn by osmosis (surrounding conversations), benefit from face-to-face communication, and with a visible Information Radiator of project information in the team area.

- **Team Charter -** create this with your team – it establishes the team values, agreements, and operating guidelines.

- **Resource Breakdown Structure (RBS)** – Is an organisational breakdown chart for the Project Team, showing where everyone fits in the Project. Can also be for physical resources.

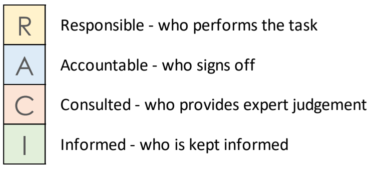
- **Resource Calendars** show the availability of resources on the project (including holidays or days off).

Tuckman's Ladder

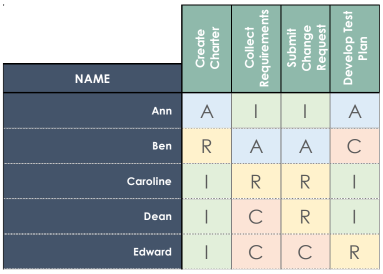
Focuses on how the team works together, from formation until project closure.

- Forming: Team members come together, are still independent and individual.
- Storming: Conflicts may arise as different personalities and working styles come together.
- Norming: The team know each others' roles and find a regular rhythm.
- Performing: The team can rely on each other, have high trust and know each other well.
- Adjourning: The project finishes and the team move on.

- **Responsibility Assignment Matrix** - Shows which team members are responsible for which work packages. The most common is a RACI:



R	Responsible - who performs the task
A	Accountable - who signs off
C	Consulted - who provides expert judgement
I	Informed - who is kept informed

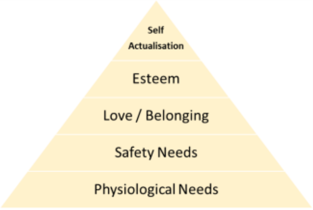


NAME	Create Charter	Collect Requirements	Submit Change Request	Develop Test Plan
Ann	A	I	I	A
Ben	R	A	A	C
Caroline	I	R	R	I
Dean	I	C	R	I
Edward	I	C	C	R

Conflict Management

- Collaborate / Problem Solve: Usually **Win / Win**. Incorporating multiple viewpoints and differing perspectives. Useful in most situations.
- Force / Direct: Usually **Win / Lose**. Forcing your viewpoint at the expense of others. Useful in urgent or crisis situations. Usually needs Positional power (i.e. a manager title).
- Compromise / Reconcile: Usually **Lose / Lose**. Searching for solutions that satisfy all parties. Everyone gives up a little bit.
- Smooth / Accommodate: Usually **Lose / Win**. Conceding your position to maintain harmony. Useful if you need to placate or keep the other person on side (i.e. a Sponsor).
- Withdraw / Avoid: Retreating from a conflict situation to be better prepared or resolved by others.

Organisational Management

• **Maslow's Hierarchy of Needs** is where team members meet these needs from bottom to top – if the previous level is not met, they cannot achieve the next level.	 Self Actualisation Esteem Love / Belonging Safety Needs Physiological Needs
• **Herzberg's Theory of Motivation:** Meet the Hygiene factors first, then move to Motivation.	**Hygiene Factors:** Job security, fairness of salary, work conditions and status. **Motivational Factors:** Recognition, Acheivement, doing something meaningful.
• **MacGregor's Theory X and Y:**	**Theory X:** Where the team is lazy and avoids work – needs a more hands on, authoritarian management style. **Theory Y:** Where the team takes pride in their work and sees it as a challenge – use inclusive and participative management, trusting the team.

- **Parkinson's Law** states that work expands to fill the time allotted to it (so give things short timeboxes).

- **Student Syndrome** states that we leave things until the last moment to complete them (like a school project).

Types of Leadership

- **Servant Leader** focuses on **others' growth**, learning, development, autonomy, and well-being.
- **Laissez Faire** takes a **hands-off approach** and allows the team to make their own decisions.
- **Charismatic** is high energy, self-confident, holds strong convictions.
- **Transactional** helps those who help them, manages by exception (where things are going wrong).
- **Transformational** gives inspirational motivation.

Communication

#	Process	Process Group	Output
1	Plan Communications Management	Plan	Communications Management Plan
2	Manage Communications	Execute	Project communications
3	Monitor Communications	Monitor & Control	Work Performance Information & Change Requests

For Predictive Projects, stakeholder communication needs are captured in the **Communications Management Plan**, including type and frequency of communication. This is closely tied to stakeholder engagement.

Agile Projects prefer face-to-face communication, close in person, place and time to when things happen. The team is co-located so they can get answers quickly. They pair up to work so they can learn by osmosis. Project information is transparent – placed on the walls in the team area. They demonstrate the product directly to the customer, instead of creating presentations.

Richness of communication is:

- Able to handle multiple information cues simultaneously (i.e. body language and tone),
- Get rapid feedback,
- It is personal, and
- Uses natural language.

Communication Skills

• **Listening Actively**	Stay engaged with the speaker, show your interest, summarise conversations or repeat their message back to check your understanding.
• **Feedback**	Listen to feedback - written, verbal or non-verbal, to help ensure that yours and others' messages are heard correctly.
• **Non-verbal**	Be aware how you are expressing yourself, with appropriate body language and gestures, tone of voice, and facial expressions.
• **Presentations**	A presentation is the formal delivery of information, including progress reports, updates to stakeholders.
• **Aware of cultural or personal differences**	Be aware of differences in habits, what is important to them, their celebrations, and ways of working.
• **Managing expectations**	Identify and set stakeholder expectations early, use negotiation to manage them ongoing.
• **Enhancement of skills**	Enhance your skills in persuasion, motivating people, coaching people, negotiating and resolving conflict.

The Five Cs of Communication

Be aware of these general rules for good communication:

- **Correct** grammar and spelling,
- **Concise** expression and elimination of excess words,
- **Clear** purpose and expression directed towards the needs of the reader,
- **Coherent**, logical flow of ideas,
- **Controlling** your flow of words and ideas.

Push and Pull Communication

Push Communication	Pull Communication
Sent to people who need to receive it. We are *pushing* it to them. e.g. emails, reports, phone calls.	Where people can access information at their own discretion. e.g. Web portals, databases, self-paced e-learning.

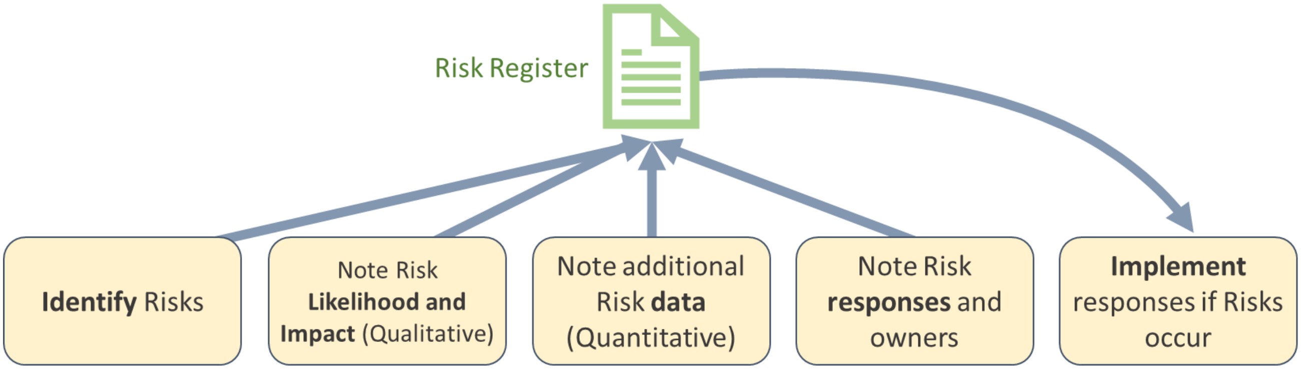
Risk Management

#	Process	Process Group	Output
1	Plan Risk Management	Plan	Risk Management Plan
2	Identify Risks	Plan	Risk Register, Risk Report
3	Perform Qualitative Risk Analysis	Plan	Updated Risk Register
4	Perform Quantitative Risk Analysis	Plan	Updated Risk Register
5	Plan Risk Responses	Plan	Updated Project documents
6	Implement Risk Responses	Execute	Change Requests, Project Document updates
7	Monitor Risks	Monitor & Control	Work Performance Information & Change Requests

Predictive Projects note and manage risk as the project starts and throughout the project.

Agile projects can use a Risk-Adjusted Backlog, where project risks are a User Story and prioritised in the Backlog against other value adding cards. They might also be shown on the team's wall or Information Radiator.

Project Risks, similar to Quality, are often **cheaper to fix earlier** in the project.



Risk Register

Identify Risks

Note Risk **Likelihood and Impact** (Qualitative)

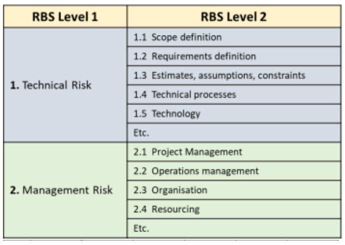
Note additional Risk **data** (Quantitative)

Note Risk **responses** and owners

Implement responses if Risks occur

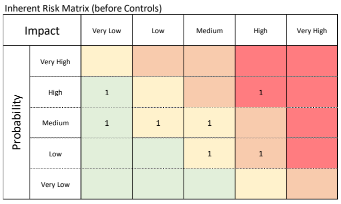
- The **Risk Register** is our list of Risks, including its title, category, impact, current status, owners and responses.

- The **Risk Breakdown Structure** is our list of Risk categories, broken down from high level (i.e. External Risk) to lower level (i.e. our competition). We can use it as a Checklist for Risk ideas, when brainstorming.



RBS Level 1	RBS Level 2
1. Technical Risk	1.1 Scope definition
	1.2 Requirements definition
	1.3 Estimates, assumptions, constraints
	1.4 Technical processes
	1.5 Technology
	Etc.
2. Management Risk	2.1 Project Management
	2.2 Operations management
	2.3 Organisation
	2.4 Resourcing
	Etc.

- The **Risk Probability and Impact Matrix** shows where our Risks fit in relation to their probability and impact scores.



Inherent Risk Matrix (before Controls)

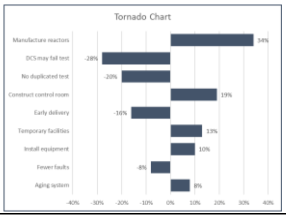
Impact		Very Low	Low	Medium	High	Very High
Probability	Very High					
	High	1			1	
	Medium	1	1	1		
	Low			1	1	
	Very Low					

Other Risk Factors we can include:

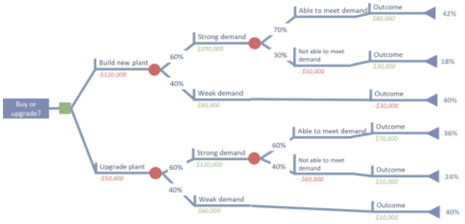
- **Urgency**: How soon we need a response, in order to be effective – a short period indicates high urgency.
- **Proximity**: How close the risk is to occurring. Short period is low proximity.
- **Dormancy**: The period of time between the risk occurring and when we might discover it. Short period is low dormancy.
- **Manageability**: How easy the risk is to manage if it occurs. Easy management is high manageability.
- **Controllability**: How easy it is to control the risk outcome. High controllability means easy to control.
- **Strategic Impact**: How the risk affects the organisation's goals.
- **Connectivity**: How related the risk is to other project risks.
- **Propinquity**: The degree to which stakeholders care about the risk. If the risk is perceived to be significant, propinquity is high.

Quantitative Risk Tools:

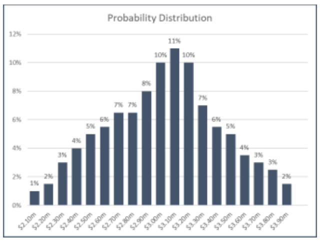
- **Sensitivity Analysis:** also "Tornado Charts", shows positive and negative impacts on an outcome.



- **Decision Trees** show us the decisions and impacts (costs or benefits) for a path, along with their probability. This usually gives us an **Expected Monetary Value** for each branch outcome.



- **Histograms** show us how often different items occur. Useful for **Monte Carlo Simulation**, where we analyse many different variations of a scenario.



Risk Responses

Threats		Opportunities	
Escalate	When it is outside the Project Manager's responsibility	**Escalate**	When it is outside the scope of the project
Avoid	When we change the project objective or the project plan to eliminate the threat.	**Exploit**	We ensure the opportunity happens (i.e. assigning resources to it, prioritise it)
Transfer	Shifting ownership of the threat to a third party, often for a fee (i.e. insurance).	**Share**	Transfer ownership of it to another party to benefit.
Mitigate	An action to reduce the probability or impact of the threat.	**Enhance**	We increase the probability of it occurring.
Accept	Acknowledges the threat, but takes no action.	**Accept**	We acknowledge it is there but don't take any action.

Procurement

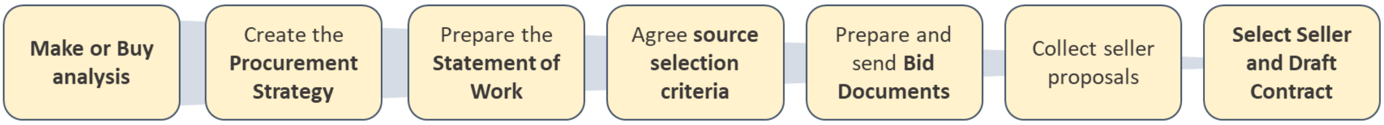
#	Process	Process Group	Output
1	Plan Procurement Management	Plan	Procurement Management Plan
2	Conduct procurements	Execute	Selected sellers, Agreements
3	Control procurements	Monitor & Control	Closed Procurements, Procurement document updates

Procurement is used for any resources external to your company – it could be a vendor, people and consultants or any other resource.

You can have an agreement without a contract (e.g. Project Charter agrees the use of internal resources), but you cannot have a contract without an agreement.

- **Claims** occur when the buyer and seller cannot agree on a change, or that a change has occurred. If they still cannot agree after a Claim, a third party may be used in Alternative Dispute Resolution (ADR).
  - Both your Claims process and ADR organisation are outlined in your Contract.
  - Like Change Requests, Claims can happen throughout your project.

Procurement process:



Make or Buy analysis	Do we really need a vendor?
Create the **Procurement Strategy**	Delivery methods, Procurement phases.
Prepare the **Statement of Work**	Details on the work to be done.
Agree **source selection criteria**	How we will select our seller.
Prepare and send **Bid Documents**	RFI RFP RFQ
Collect seller proposals	
Select Seller and Draft Contract	Fixed Price, Cost + Incentive fee, Time & Materials.

- **Advertising** in specialty trade publications or websites can invite more companies to bid and help increase our pre-qualified sellers list.

- **Bidder Conferences** (also vendor conferences or pre-bid conferences) are meetings between the buyer and all the prospective sellers prior to proposal submittal, to ensure they all have a common understanding.

Bid Documents include:

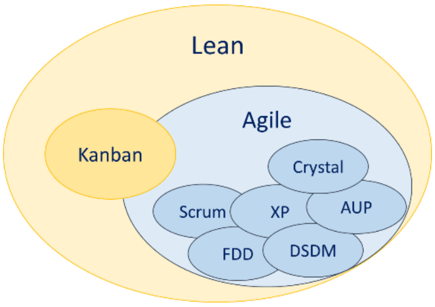
• **Request for Information (RFI)**	When we need more information on what the seller can provide.
• **Request for Proposal (RFP)**	When there is a problem and the solution is not easy to find (needs someone to **propose** a solution).
• **Request for Quote (RFQ)**	When we just need the cost from the sellers, matched to how they will satisfy the requirements.

Contract Types include:

• **Fixed Price Contracts**	Including Firm Fixed Price, Fixed Price + Incentive Fee, Fixed Price with Economic Adjustments (for currency or inflation). Good when the scope is straight forward and requirements are well defined.
• **Cost Reimbursable Contracts**	Including Cost plus Fixed Fee, Cost plus Incentive Fee, Cost plus Award Fee. Good when you want to incentivise the seller, or if work is expected to change.
• **Time and Materials Contracts**	Useful when a precise statement of work is not available. Also used for acquisition of staff, experts or outside support.

Agile Practice Guide

Agile is both **Incremental** (small, regular releases or increments of value) and **Iterative** (reflecting and improving regularly). Agile as we know it today came from many different methodologies:



Agile Domains and Tasks

Below are PMI's Agile Certified Practitioner Domains and Tasks – use these as a guide to your answers as well. **These have become a natural part of the PMP Exam since 2021**:

Agile Principles and Mindset		
Advocate for Agile Principles	Contribute to a Safe and Trustful team environment	Encourage Emergent Leadership
Support organisational change towards Agile	Encourage Knowledge Sharing	Practice Servant Leadership
Value Driven Delivery		
Define Positive Value	Manage Risk	Prioritise Value
Develop Incrementally		
Agile Stakeholder Engagement		
Understand Stakeholder Needs	Ensure Stakeholder Involvement	Manage Stakeholder Expectations
Agile Team Performance		
Form Your Agile Team	Empower Your Agile Team	Ensure Collaboration and Commitment
Adaptive Planning		
Plan at multiple levels	Adapt your plan	Use Agile Sizing and Estimation
Finding and Solving Problems		
Ensure Psychological Safety	Make Problems Visible	
Continuous Improvement		
Tailor your Process	Improve Team Processes	Seek Product Feedback

Agile Manifesto and Overview

In 2001 a group of 17 individuals representing the most widely used lightweight software development methodologies, agreed on a common set of values and principles which became known as the Agile Manifesto:

We value:

Individuals and Interactions	over	Processes and Tools
Working Software	over	Comprehensive Documentation
Customer Collaboration	over	Contract Negotiation
Responding to change	over	Following a Plan

The 12 Clarifying Principles

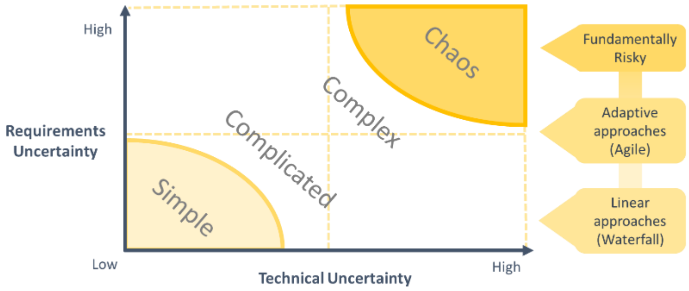
Keep these principles in mind – they are a good mindset for the PMP Exam.

12 Clarifying Principles		
1. Our highest priority is to satisfy the customer through **early and continuous delivery** of valuable software.	**5.** **Build projects around motivated individuals.** Give them the environment and support they need, and trust them to get the job done.	**9.** Continuous attention to technical excellence and **good design** enhances quality.
2. **Welcome changing requirements**, even late in development. Agile processes harness change for the customer's competitive advantage.	**6.** The most efficient and effective method of conveying information to and within a development team is **face to face conversation**.	**10.** **Simplicity** – the art of maximising the amount of work not done – is essential.
3. **Deliver working software frequently**, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	**7.** **Working software** is the primary measure of progress.	**11.** The best architectures, requirements, and designs emerge from **self-organising teams**.
4. **Business people and developers must work together daily** throughout the project.	**8.** Agile processes promote sustainable development. The sponsors, developers, and users should be able to **maintain a constant pace** indefinitely.	**12.** At regular intervals, **the team reflects** on how to become more effective, then tunes and adjusts its behaviour accordingly.

The Difference Between Project Approaches

Definable work:	(operations, “Business as Usual” or well-defined projects)
High-uncertainty work:	(new designs, exploratory work, high rates of change or complexity).

Agile is suited towards **high-uncertainty** work.



Agile is a combination of Incremental and Iterative approaches:

Approach	Requirements	Activities	Delivery	Goal
Predictive	Fixed	Performed once for the entire project	Single delivery	Manage cost
Iterative	Dynamic	Repeated until correct	Single delivery	Correctness of solution
Incremental	Dynamic	Performed once for a given increment	Frequent smaller deliveries	Speed
Agile	Dynamic	Repeated until correct	Frequent small deliveries	Customer value via frequent deliveries and feedback

A guide to solving Agile issues on your PMP Exam:

Issue	Tailoring idea
Demand pattern is **sporadic**	Build in a cadence (a sprint or regular timebox) to help the team demo, retrospect and take in new work.
A new team who needs **process improvement**	Retrospect more often and select improvements
Flow of work is often delayed or interrupted	Make the work visible using Kanban boards, and limit the work in progress to avoid multi-tasking.
Quality of deliverables is poor	Consider using Test Driven Development practices (test-first), finding the root cause through retrospectives and error-proofing.
More than one team is needed to build a product	Use Agile scaling frameworks, starting with Scrum of Scrums.
Team is new to Agile approaches	Start by training team members in Fundamentals of the Agile mindset and principles. Workshop a specific approach (i.e. Scrum) once chosen.

Building an Agile Team

Agile teams are:

- Small (range in size from three to 12 members)
- Co-located in the same space (to pair up and learn by osmosis – the environment around them)
- Generalising specialists (with one deep specialty and many general skills)
- 100% dedicated to the team (to avoid multitasking)

Team areas have an **Information Radiator**, with team Burndown charts, Kanban Boards and other information available at a glance.

- Pairing: Work in pairs to complete work, check and learn together.
- Swarming: Multiple team members "swarm" around a problem to solve it quickly.

Agile Team Roles:

- **Cross functional team member:** Our generalising specialists or "T-shaped" people.
- **Product Owner**: Represents the customer or the business, and prioritises the Product Backlog (features to be completed)
- **Team Facilitator:** Also the Scrum Master, Agile Lead, Team Coach, Servant Leader. Facilitates problem solving, leads team ceremonies (i.e. Stand-up), grows the team and removes blockers or impediments.

Servant Leadership

We lead through service to the team by:		
Listening.	Helping people grow.	Making it safe to make mistakes and raise concerns.
Coaching versus Controlling.	Promoting the energy and intelligence of others.	Promoting psychological safety, respect and trust.

Servant Leaders work on the team:	
Purpose	The team's "Why", their goal or reason for being there.
People	Ensuring an environment where everyone can contribute and succeed.
Process	Base your process on results: If a **cross-functional team** is **delivering finished value** often, and **reflecting and improving** their product and process, the team is **Agile**.

Servant Leaders also:

1. Facilitate: Help the team problem solve, elicit the information needed.
2. Remove impediments
3. Grow the team: In Agile, and their role capability.
4. Pave the way for others' contribution: It's not about "me", it's all about "we".

Agile Core Foundations

Early and Frequent Feedback	Feedback (good and bad) is valued. We get feedback through: • **Retrospectives** (checking on and improving the team process), • **Sprint Reviews** (by demonstrating the product to the customer), • **Information Radiator** (by having all needed information visible. • **Stand-ups** (by raising blockers), • **Pairing** (by working and checking together).
The Whole Team Approach	Anyone needed to create our "Product" in **brought into the team**.
Rolling Wave Planning	**Work near-term is planned in detail, while work far away is planned at a high level only**. Example: • Product Backlog and Product Roadmap have a high-level view of the features coming up, then; • The team plans User Stories to create a part of one of these features during a Sprint during Sprint Planning.
Visual Management	All necessary information is visible in the team area, e.g. • Kanban Board to show work in the Sprint, • Burndown chart to show how much work is completed.
Sticky / Stable Teams	Teams are **fully funded** for the necessary time. Teams are more engaged to stay by helping our team grow and removing blockers.
The Three Amigos *(creating User Stories)*	A "triad" of **Business Person / Customer, Developer and Tester** come together to create User Stories – providing requirements, a solution and test criteria.
Story Card Sizing	The team doing the work **estimate on the effort** to complete each User Story (before the sprint begins). The smallest User Story is 1 point, and other Stories are sized relative to the smallest one, in the Fibonacci number sequence: 1,2,3,5,8,13,21 etc. Could also be T-shirt sizes of S, M, L, XL etc.
Build in Quality	By improving our process (through **retrospectives**) and reviewing the product regularly (in the **Sprint Review**), the whole team focuses on Quality.

Servant Leadership	**Leading through service** to the team.
Continuous Integration	Integrating the new features into the whole system (test environment) and running an overall test ("regression test") regularly, we ensure our work hasn't broken anything.

Agile Core Practices

Charter the Project and the Team	**Project Charter**: What the team is delivering, and its benefits. **Team Charter**: Team ways of working, ceremonies, team values.
Sprint Planning	The team meet before the sprint begins and agree on the User Stories that will go into the sprint. This depends on: • **Team Velocity**: how much the team can complete in each sprint. • **Priority**: What the Product Owner says is the highest priority. • Any **technical limitations.**
Daily Stand-ups	The team meet each day, update their User Stories on the Kanban Board, and answer: • What did I complete yesterday? • What am I working on today? • Is there anything blocking me?
Backlog Refinement	The Product Owner continually prioritises the Product Features, the team breaks Features down into User Stories, with Acceptance Criteria, and sized for effort.
Sprint Review	The team demonstrates a usable (increment) or feature to the customer (or Product Owner) at the end of each sprint, for additional feedback.
Retrospectives	The team meet at the end of each sprint to reflect on their process and improve, asking: • What went well? • What didn't go well / what challenges do we face? • What did we learn? • What still puzzles us?

Other Agile Terms

- **Product Backlog:** The **prioritised list of features** to be delivered (owner by the Product Owner) that will deliver customer value.
- **Sprint Backlog:** The **User Stories** the team decide they can complete in a sprint (or iteration), based on their **velocity**.
- **Technical Debt:** Degraded code quality over time (from implementing not-well-thought-out solutions).
- **FDD:** **Feature Driven Development**, where we design by feature, build by feature, test and release by feature, and have dedicated feature teams.
- **DoR:** **Definition of Ready** – criteria the team agree on for when an item is ready to be worked on.

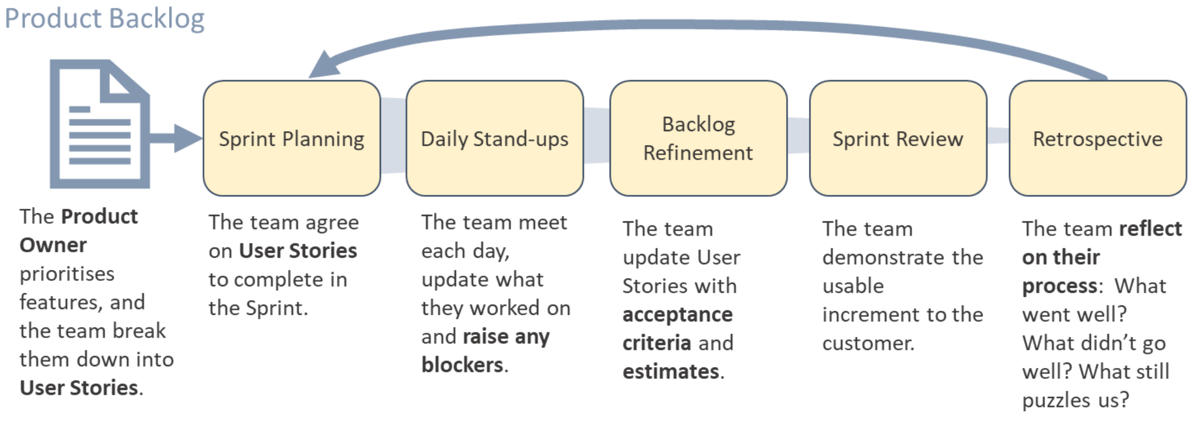
- **DoD:** **Definition of Done** – the criteria for when an item can be called "complete" (e.g. developed, tested, and signed off)
- **User Story:** A User Story is a part of a feature that is small enough to complete in a sprint. It has acceptance criteria and an estimate on its effort (as "Story Points").
- **INVEST:** User Stories should be: **I**ndependent, **N**egotiable, **V**aluable, **E**stimable, **S**mall and **T**estable.
- **TDD:** **Test Driven Development**, where we write the test first, then the code that will allow the test to pass.
- **BDD:** **Behaviour Driven Development**, where we can write our automated tests in user-centric language: Given (a scenario), When (something happens), Then (I want this to happen).

Scrum Overview

Scrum is one of the post popular Agile frameworks. It is also fairly simple.

Scrum Roles	
Product Owner	Prioritises the features that will deliver customer value, provides business input.
Scrum Master	Facilitates ceremonies and problem solves with the team.
Team	Who do the work and deliver value.

Typical Scrum Sprint:



The **Product Owner** prioritises features, and the team break them down into **User Stories**.

The team agree on **User Stories** to complete in the Sprint.

The team meet each day, update what they worked on and **raise any blockers**.

The team update User Stories with **acceptance criteria** and **estimates**.

The team demonstrate the usable increment to the customer.

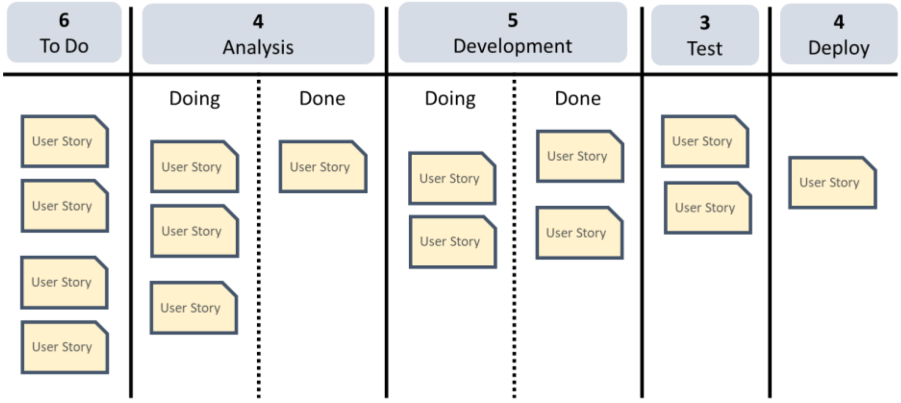
The team **reflect on their process**: What went well? What didn't go well? What still puzzles us?

Kanban Overview

Kanban translates to "Visual sign" or "card" from Japanese. It was used in the Toyota Product System to signal when a previous (upstream) process needed to begin work or create something. In Agile, we mostly use it as a **Kanban Board**.

User Story cards move from the backlog (or To Do), to In Progress, to Done. We can clearly see where the bottlenecks are (when there are too many cards in a column).

Kanban aims to limit "Work in Progress" (WIP) to reduce multi-tasking and increase focus. We can have WIP limits on each Kanban Board column. Each team is different, so adjust your board to suit!



eXtreme Programming (XP)

You might also hear about XP. It is similar in nature to other frameworks and the Core Practices and Agile Foundations.

XP Core Values

Communication, Simplicity, Feedback, Respect, Courage.

XP Core Activities

Coding, Testing, Listening, Designing.

XP Core Practices

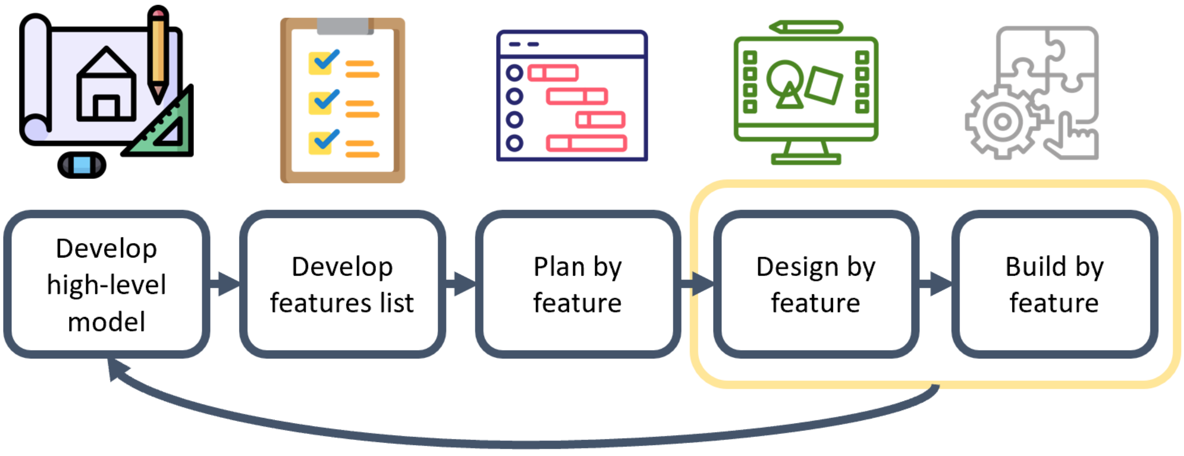
- **Whole Team:** Everyone needed to do the work is within the team.
- **Planning Game:** Using **Planning Poker** to estimate on User Stories (the team estimate, high and low estimators explain their reasons, and the team re-estimate until a consensus is reached).
- **Customer Tests:** Writing tests first, from the customer's point of view.
- **Small Releases**
- **Simple Design**
- **Pair programming:** Developers pair up to code and check together.
- **Test Driven Development:** The team write tests first, then the code to make the test pass.
- **Refactoring:** Cleaning up the code and removing duplication.
- **Continuous Integration:** Merging new features into the main system test environment (often daily) to check that nothing is broken.
- **Sustainable Pace**
- **Collective Code Ownership:** Anyone can change any part of the code.

- **Coding Standard:** A standard way of coding, commenting etc to make it easier for anyone to change any part of the code.
- **Metaphor**: The feature is described as a metaphor, so everyone understands.

Feature Driven Development (FDD)

A feature is a small piece of usable capability that brings business value.

The five activities of FDD are performed iteratively.



FDD Engineering Best Practices are:

- **Feature Teams** Teams dedicated to a feature.
- **Develop by Feature** Developing a usable feature at a time.
- **Individual Class Ownership** One owner of a group of code.
- **Domain Object Modelling** Sequence or Context Diagrams of how the code interacts.
- **Inspections** Of the code and the result.
- **Configuration management** Maintaining versions of changes.
- **Regular builds** Merging the code to the main (test) system regularly.
- **Visibility of Progress and Results**

Crystal

Crystal advocates that:

- Technologies change techniques.
- Cultures change norms.
- Distances change communication.

Crystal also understands that projects need more governance the larger and riskier they are. It measures by Team Size, and Project Criticality. Agile favours small teams, so Crystal Clear and Crystal Yellow.

Sizing framework:

Criticality	Life (L)	L3	L10	L30	L80	L200
	Essential Money (E)	E3	E10	E30	E80	E200
	Discretionary money (D)	D3	D10	D30	D80	D200
	Comfort (C)	C3	C10	C30	C80	C200
	Number of **people** involved +/- 20%	1 - 6	7-20	21-40	41-100	100-200
		Crystal clear	Crystal yellow	Crystal orange	Crystal red	Crystal Maroon

Crystal Core Properties

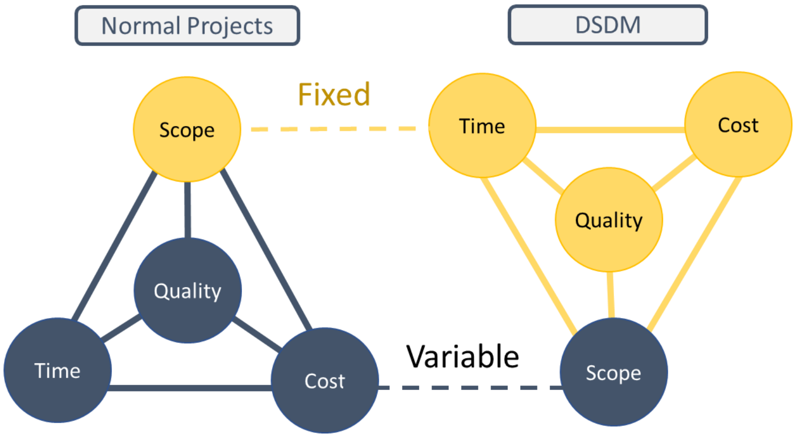
- Frequent Delivery
- Reflective Improvement
- Close Communication
- Personal Safety
- Focus
- Easy access to expert users
- Frequent Integration
- Automated tests
- Configuration Management

It also uses similar techniques, as we've seen before in other frameworks and our Core Practices.

- **Methodology shaping** To tailor our project and team to the situation.
- **Reflection workshops** Retrospectives, anyone?
- **Blitz Planning** Similar to the Planning Game in XP, where we plan and write User Stories.
- **Wideband Delphi** Estimating as a group, the high and low estimates explain when we estimate again.
- **Daily stand-up** Sound familiar?
- **Process miniature** Running a shortened version of the way of work, to become familiar.
- **Side-by-Side programming** Like Pair programming, two people work and learn together.

DSDM

Dynamic Systems Delivery Method is where the alternate "Constraints Triangle" comes from, where Time and Cost are fixed, and Scope is variable.



DSDM Core Principles:

- Focus on the Business Need
- Collaborate
- Build incrementally from firm foundations
- Communicate continuously and clearly
- Deliver of time
- Never compromise quality
- Develop iteratively
- Demonstrate control & use appropriate techniques

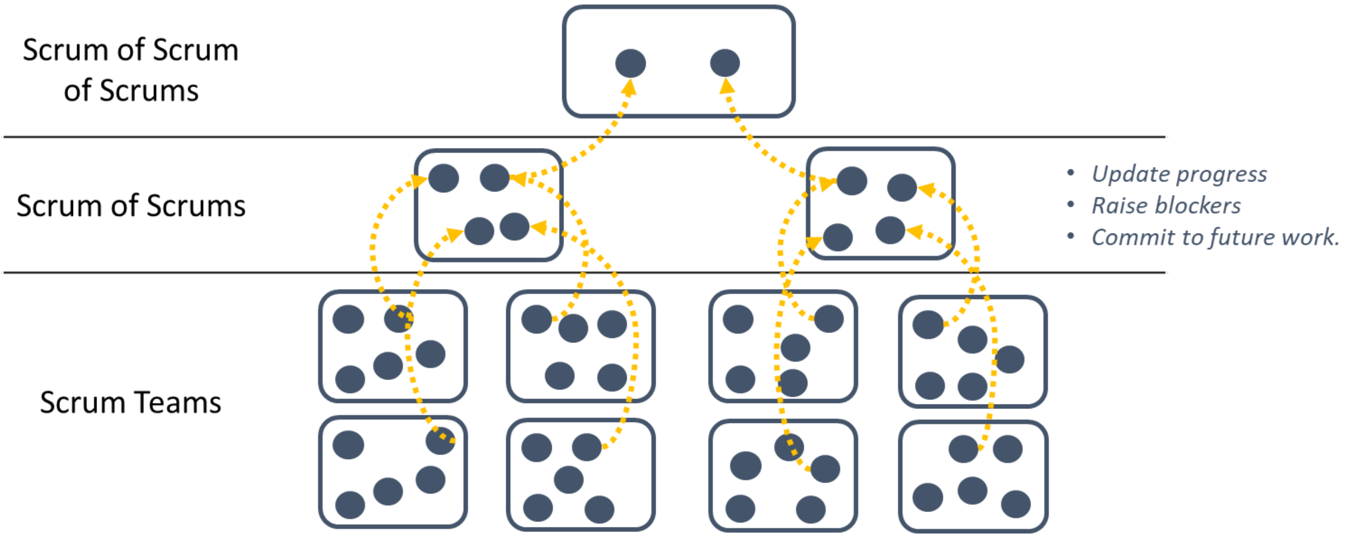
DSDM Core Techniques:

- **Timeboxing** i.e. short iterations, or timeboxing "spikes" for research or problem solving.
- **MoSCoW** Prioritising by Must have, Should Have, Could Have, Won't have.
- **Prototyping**
- **Testing**
- **Workshopping** Swarming around problems.
- **Modelling** Building diagrams of the code, systems or processes.
- **Configuration Management** To track changes in the system and code.

Scaling Agile Frameworks

Scrum of Scrums

Where a **representative** from each Scrum team attends a **Program**, and ultimately **Portfolio** Scrums, to report on high-level work completed, and any blockers across teams.



Large Scale Scrum (LeSS)

LeSS applies Scrum ways of working to Programs and Portfolios, only when necessary. It believes:

- **Large Scale Scrum is Scrum**: Applying it to larger areas and treating them as one team.
- **Empirical Process Control**: Updating and improving the process as we go.
- **Transparency**: Managing through transparency (Kanban, Information Radiator).
- **More with LeSS**: More ownership with less roles, more learning with less processes.
- **Whole Product Focus**: Focus on the whole with Systems Thinking.
- **Customer Centric:** Always being connected to paying customers.
- **Continuous Improvement:** From Lean: Going to where the work is done, Kaizen (small improvements), removing Waste, improving Flow, Respect for people).
- **Systems Thinking**: Being aware of how each part affects the rest.
- **Lean Thinking**: Organising around customer value, understanding the value stream.
- **Queuing Theory**: A process is only as fast as its slowest part.

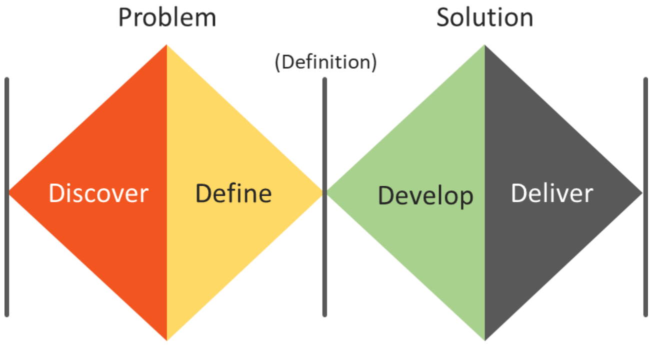
Scaled Agile Framework (SAFe)

SAFe focuses on organising project teams around "Value Streams" – i.e. products or services.

Agile "Release Trains" are teams of teams, adding anyone needed from around the organisation to deliver the value (i.e. the Whole Team Approach).

- Each value stream has **OKRs** - **Objectives** (a written goal) and **Key Results** (the business measures we meet by delivering value).

- It uses the Human Centred Design (**Double Diamond**) to solve problems and create features.



- It uses a portfolio Backlog and Kanban,
- Features are ranked by Desirability, Feasibility and Viability,
- It uses Lean Budgeting (to fund Value Streams / Product teams, not projects),

Other Agile Frameworks

There are many other frameworks that helped make up the original Agile Manifesto, but the **Core Practices** and **Foundations** cover them all.

Agile Scenarios

You will come across tricky Agile scenarios in the exam. Here are some examples and what to do:

Agile Scenarios	
Unclear purpose or mission for the team:	• Workshop the **Agile Team Charter** to align the team Vision, Mission and high level Features.
Unclear working agreements for the team:	• Workshop the **Agile Team Charter** for alignment between Values, **Definition of Ready & Done** and working agreements.
Unclear team context:	• Workshop the **Agile Team Charter** for context, with boundaries, committed people and systems, and high level features.
Unclear requirements:	• Help sponsors and stakeholders craft a product vision, help the Product Owner build a **product backlog** of features. • Ensure the "**Three Amigos**" get together regularly to create User Stories.
Poor user experience:	• **Involve actual users** of the system early, for requirements and user story collaboration. • **Showcase** to customers in the Sprint Review.
Inaccurate estimation:	• **Reduce story size** by splitting stories. • Estimate with the people doing the work. • Provide guidelines on user story sizing.

Unclear work assignments or progress:	- Use a **Kanban board** to see the work. - Use daily stand-ups to report blockers and micro-commit.
Team struggles with obstacles:	- Ensure the servant leader / scrum master helps clear these obstacles.
Technical debt:	- Build in "**Slack**", a 5-point card to refactor code, - **Refactor** regularly (tidy up the code or product), - Ensure developers are part of user story solution collaboration.
Defects:	- Consider technical practices like **Pair work**, shoulder checking, - **Test Driven Development** and Automated Testing, - Unit and System testing, a robust **Definition of Done**.
Work is not complete:	- Reduce User Story size, define the definition of done.
Work delays/overruns:	- Ensure User Stories are small (8 points or under), - Ensure the Product Owner is present in the team, - Ensure the “Triad” of Business representative/Developer and Tester meet regularly.
Too much product complexity:	- Encourage “the simplest thing that could work”, - Use Prototypes or mock-ups.
No improvement in Team Process:	- Hold retrospectives regularly.
Too much up-front planning leading to rework:	- See what the options are to deliver value in usable increments, instead of plans or designs.
False starts, wasted efforts:	- Ensure the Product Owner is 100% part of the team.
Inefficiently ordered product backlog items:	- Rank features (then stories) by value or benefit, adjusting for effort or cost (ensure the reward meets the cost).
Rush then wait - uneven flow of work:	- Plan each Sprint to the team's velocity, and not more. - Reduce User Story size, reduce Work in Progress. - Swarm around blockers quickly.
Unexpected or unforeseen delays:	- Use stand-ups to raise blockers, - Ensure psychological safety so the team raises issues.
Siloed teams, instead of cross-functional teams:	- Work with managers of external resources to dedicate them to the team

Hybrid Projects

You will come across questions about a **Hybrid** (**part Waterfall, part Agile**) way of work. Here are some general tips:

- **Predictive** (Waterfall) is best for creating things with:
  - Stable or **well-defined** requirements,

    - When we can't release small pieces,
    - If there is **high risk** or high quality-checking needs,
    - If we're in a highly regulatory environment.
- **Adaptive** (Agile) is best for creating things when:
    - We need to work through uncertain or **complex** environments,
    - We need to (and can) release small pieces along the way,
    - We have a small team (3-12 people), a close customer, and control over the product.

If you have to choose, or split your team, use those as a guide. Scenarios might include:

- A building planned up front (**predictive**) where we are experimenting with new roofing materials (**Agile**),
- Working with a third-party vendor where:
    - They are creating their part of the product in one piece (**predictive**),
    - We are creating small increments along the way, and use our last sprint (**Agile**) to integrate their product.
- A linear product (**predictive**) where we use Stand-ups, a Kanban board and Showcases (**Agile**)

Moving to an Agile Way of Work

If the organisation is moving to an Agile way of work, we can **use Agile to move to Agile**. We can:

- Ensure there is a **Product Owner** representing each area for the change,
- Have a **backlog** of items to try, one at a time (i.e. forming small, self-sufficient teams with Product Owners, starting stand-ups, using Visual Management and Kanban boards, forming and Prioritising Backlogs etc),
- Use a **Kanban board** to show progress of your changes towards Agile,
- Meet with your executives or change team in a **daily stand-up** to report progress and blockers,
- **Demonstrate** changes at the end of each Sprint (i.e. two weeks).

PMBOK Guide – 7th Edition

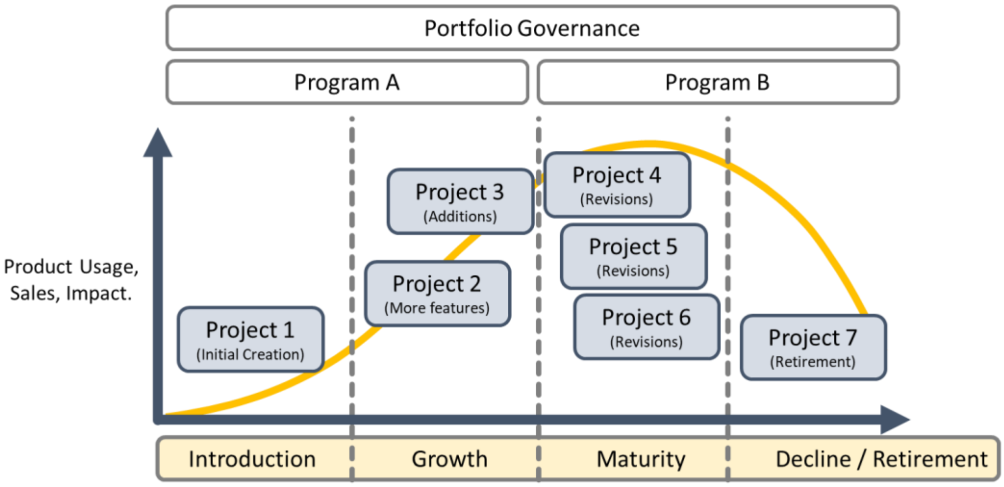
Twelve Principles of Project Management

These are a good guideline for the right mindset when answering questions on the PMP Exam.

Be a diligent, respectful and caring steward.	Recognise, evaluate and respond to system interactions.	Navigate complexity.
Create a collaborative team environment.	Demonstrate leadership behaviours.	Optimise risk responses.
Effectively engage with stakeholders.	Tailor based on content.	Embrace adaptability and resiliency.
Focus on value.	Build quality into processes and deliverables.	Enable change to achieve the envisioned future state.

Product Management

Product Management is becoming more popular, and involves **managing a portfolio of projects** for a Product across its lifecycle.



Project Performance Domains

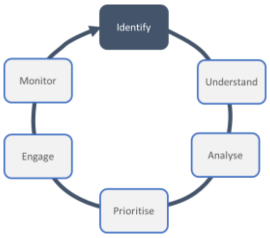
Stakeholder	Team	Development Approach and Lifecycle	Project Planning
Project Work	Project Delivery	Project Measurement	Uncertainty and Risk

Stakeholder

A **stakeholder** is anyone affected (or who thinks they are affected) by our project.

Outcome: We want productive relationships with our stakeholders, to ensure they support project objectives.

We need to Identify, Analyse, Prioritise and Engage our Stakeholders, then Monitor their engagement.



Team

Includes the Project Management Team, the Project Manager, and the Project Team.

Outcome: Shared ownership and leadership displayed by all members.

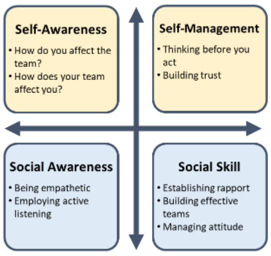
Develop your team with:

- Team **vision and objectives**
- Known **Roles and Responsibilities**
- Clear **Team Operations**
- Ongoing **Guidance and Growth**

Ensure transparency, a positive discourse, and celebrate success with your team.

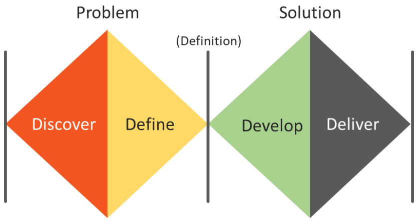
Emotional Intelligence is key:

- Self-Awareness: How your actions affect others.
- Self-Management: Thinking before you act.
- Social Awareness: Being empathetic to those around you.
- Social Skill: Establishing rapport and managing your own attitude.


Self-Awareness
• How do you affect the team?
• How does your team affect you?
Self-Management
• Thinking before you act
• Building trust
Social Awareness
• Being empathetic
• Employing active listening
Social Skill
• Establishing rapport
• Building effective teams
• Managing attitude

Make Decisions with:

- Voting
- Brainstorming
- Multi-criteria Decision Analysis
- Diverge / Converge, Problem / Solution



Manage Conflict by:

1. Keeping communication open and respectful,
2. Focus on the issue not the person,
3. Focus on the present not the past,
4. Search for alternatives together.

Development Approach and Lifecycle

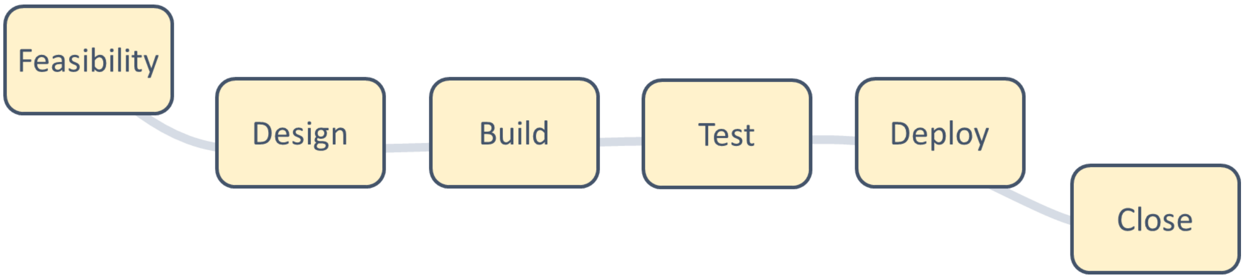
Outcome: We want a Development approach that matches our deliverables and environment.

We can choose between **Predictive, Incremental, Iterative, Adaptive,** and **Hybrid**.

How do we select our Development Approach?

Product	• What is the degree of innovation needed? • How certain are the requirements? • How easy is it to change? • Is there high risk or regulations?
Project	• Is there a Product Owner available? • Do we have access to our customer? • Can we deliver small pieces, early?
Organisation	• Is the Org Structure flat, or bureaucratic? • What is the capability of the organisation? • What are the project team sizes, and locations?

Common Lifecycle Phase definitions:



Predictive (Waterfall) performs all these once, while **Adaptive** (Agile) perform them once for each product increment.

Project Planning

Outcomes: Ensure the project progresses in a deliberate manner, and evolving information is used to adjust.

Planning is impacted by our:

- Development approach (Agile or Waterfall)
- Deliverables (small or large)
- Market Conditions and;
- Regulatory requirements.

We'll need to estimate our plan at first.

- Adjust estimates for uncertainty (**give a range** i.e. -10% to +25%)
- Use simulation or prototypes and **build in reserves** for uncertainty.

Planning our:

Schedule	1. Decompose project scope into activities, 2. Sequence those activities, 3. Estimate effort, duration and resources required, 4. Allocate people based on availability. Remember Crashing, Fast Tracking, Leads and Lags.
Budget	1. Estimate the work package costs. 2. Add a Contingency Reserve (these become the Cost Baseline) 3. Add Management Reserves (for unforeseen project scope)
Project Team	• People might be Internal or External to our organisation • Consider their **Cost, Expertise** and **Location.**
Communication	• **Who** needs the information? • **What** information do they need? • **Why** do we need to share this information? • **How** do we provide the information? • **When** and how often do we share? • **Where** do we get the information from?
Physical Resources	• Consider the lead time for delivery, transportation time and storage of materials.
Procurement	• Know your make or buy analysis
Changes	Could be: • A formal change control process • Reprioritising a backlog • Re-baselining a project artefact Might come from: • Customer requests • Evolving product scope • Environment or technical changes

Project Work

Outcomes: We want efficient use of resources, effective project performance, and appropriate communication with stakeholders.

The Project Manager establishes the project process – ensure you learn and adapt as a team.

Balance competing constraints of **Scope, Time, Cost** and **Quality**.

Project Delivery

Outcomes: Our project contributes to the business objectives (and strategy), our team is clear on their requirements and our stakeholders accept the deliverables.

We create deliverables, based on Customer Requirements:

- Use **prototypes** or mock-ups to discover complex requirements,
- Ensure requirements are traceable, clear, verifiable.
- Define them as **Acceptance Criteria**
    - Include any technical performance measures.
- **Decompose** project scope into deliverable pieces.

Project Measurement

Outcomes: We'll have a reliable understanding of the status of the project – with data we can **make decisions** from.

We can use:

Leading Indicators:	• Size of a project, or deliverables • Lack or processes, • Stakeholders not available.
Lagging Indicators:	• Deliverables completed, • Resources consumed, • Schedule or cost variance.

We can measure our delivery, business value, budget and schedule performance, Stakeholder satisfaction, errors or defects, Lead time and Cycle time.

Beware of:

Hawthorne Effect	• What we measure influences behaviour.
Vanity Metrics	• Measuring things that aren't **actionable**.
Demoralisation	• If it isn't achievable.
Misusing the metrics	• Using metrics as a beating stick, instead of helping remove blockers.
Confirmation Bias	• Using metrics to confirm our own opinions.
Correlation versus causation	• If two things happen at the same time (correlated) it doesn't always mean one has caused the other.

Uncertainty and Risk

Outcomes: Proactively exploring and responding to uncertainty – anticipate threats and deliver on time.

Responding to Uncertainty:

- Gather information
- Prepare for multiple outcomes
- Use prototyping or modelling to work through complexity
- Build resilience into the process – with checks, error proofing.

Types:

- Ambiguity: Lack of understanding or where more than one outcome is possible.
  Use progressive elaboration (Agile), experiments and prototypes.

- Complexity: When there are many interconnected influences.
  Decouple parts, use simulation, balance viewpoints and error-proof.

- Volatility: Subject to rapid or unpredictable change.
  Use reserves and analyse multiple alternatives.

Tailoring your Project

Basically, we can (and should) tailor anything on our project to ensure it meets the project outcomes and delivers business value. Including:

- Our Process or Project approach,
- Artifacts and tools we use,
- Methods.

Tailoring approach:

1. Select the initial Development Approach,
2. Tailor for the Organisation,
3. Tailor for the Project
   a. Based on Product type, Culture, Size and Criticality
4. Implement ongoing improvement.

Models, Methods and Artifacts

- **Models:** Helps explain how something works in the real world (like a framework),
- **Methods:** A means for achieving an outcome (like a process),
- **Artifacts:** A template or document.

Leadership and Communication Models:

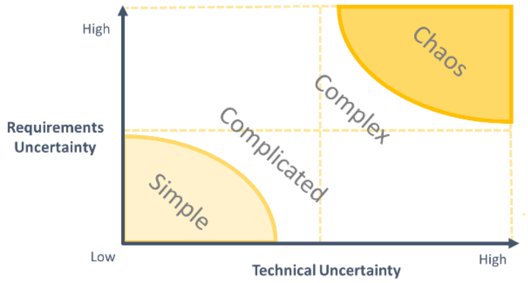
Situational Leadership II	From Ken Blanchard, measures a person by their: • **Skill** – how well they can do the job. • **Will** – how much they want to do the job.
OSCAR Coaching Model	Uses: **O**utcome (goals) **S**ituation **C**hoices **A**ctions **R**eview

Cross Cultural Communication	Be aware of the sender and receiver's: • Knowledge, Experience, Language, Thinking and Communication Style.
Effectiveness of Communication	Measured by Effectiveness and Richness (face to face is best)
Gulf of Execution and Evaluation	**Execution:** Does it match what we expect it to do? **Evaluation:** Does it support the user to discover how to interact with it?

Change Models:

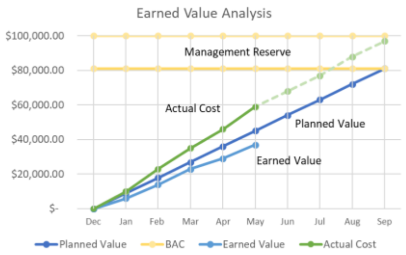
PMI Managing Change	1. Formulate the change 2. Plan the change 3. Implement the change 4. Sustain the change **5.** Manage the transition
ADKAR	1. Build **A**wareness 2. Build **D**esire 3. Build the **K**nowledge 4. Build the **A**bility 5. **R**einforce and Monitor the change.
8 Steps to Change	From John Kotter: 1. Create urgency 2. Form a powerful coalition 3. Create a vision for the change 4. Communicate the vision 5. Remove obstacles 6. Create short term wins 7. Build on the change 8. Anchor the changes in corporate culture
Satir Change Model	From Virginia Satir: 1. Late status quo 2. A foreign element introduces change (i.e. a competitor) 3. There is chaos 4. We have a transforming idea 5. Practice and integration 6. New Status Quo
Bridges Transition Model	William Bridges believes people experience loss as they go through change. 1. Communicate the reason for the change, 2. Understand the impact from those affected, 3. Check the organisation's readiness for change, 4. Educate leaders on how the change will affect their people, 5. Monitor progress as people go through the stages of transition, 6. Help people understand how they can positively contribute.

Complexity Models:

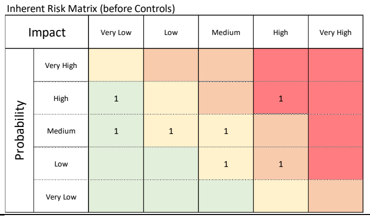
Stacey Matrix	Measures by uncertainty of the deliverable, and uncertainty of the technology to create it.  High Requirements Uncertainty Low Technical Uncertainty High Chaos Complex Complicated Simple
Cynefin Framework	• If the cause and effect is **obvious**: Use best practices • If complicated relationships or **known unknowns**: Use good practices. • If complex relationships or **unknown unknowns**: Probe the environment (iterate forward). • If **chaotic**, stabilise the situation first. • If disordered, break into smaller parts and assess from there.

Commonly Used Methods

Ensure you are familiar with these methods:

• **Alternatives analysis**	Finding the best of multiple alternatives.
• **Assumption and Constraint analysis**	Analysing constraints.
• **Benchmarking**	Comparing your process against the process of another organisation to find improvements.
• **Business Justification Analysis**	NPV, IRR, Benefit to Cost ratio. Higher is better.
• **Check Sheet**	For counting or tallying
• **Cost of Quality**	Preventative, Appraisal, Internal Failure, External failure.
• **Earned Value Analysis**	BAC, PV, EV, AC. 
• **Expected Monetary Value**	Probability x Benefit (or Cost)
• **Forecasting**	Regression analysis, Variance analysis, simulations.
• **Make or Buy analysis**	Is it better to make in-house or buy from a vendor? (e.g. which is better Benefit to Cost)

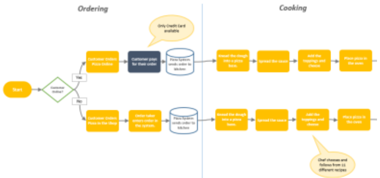
- **Probability and Impact Matrix** A risk's Probability and Impact.



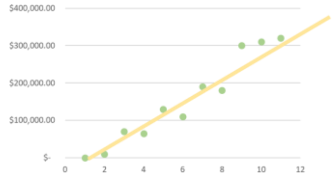
Inherent Risk Matrix (before Controls)

Impact		Very Low	Low	Medium	High	Very High
Probability	Very High					
	High	1			1	
	Medium	1	1	1		
	Low			1	1	
	Very Low					

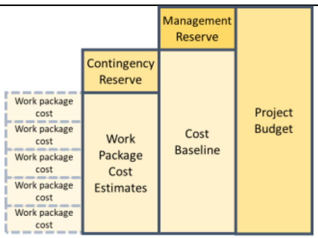
- **Process Analysis** Flow charts,
  Process steps.


Ordering
Cooking
Only Credit Card available
Start
Customer Online?
Yes
No
Customer Orders Pizza Online
Customer pays for their order
Pizza System sends order to kitchen
Knead the dough into a pizza base.
Spread the sauce
Add the toppings and cheese
Place pizza in the oven
Customer Orders Pizza in the Shop
Order taker enters order in the system.
Pizza System sends order to kitchen
Knead the dough into a pizza base.
Spread the sauce
Add the toppings and cheese
Place pizza in the oven
Chef chooses and follows from 11 different recipes

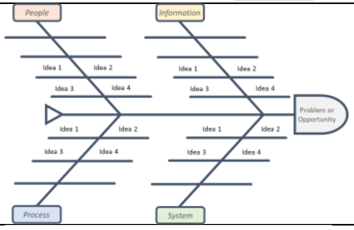
- **Regression Analysis** Statistical tool to find trends.



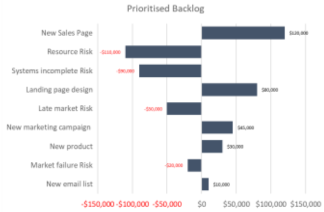
- **Reserve Analysis** How much Contingency and Management reserves do we have left?



- **Root Cause Analysis** Helps us find the real cause of an issue.
  Fishbone diagram brainstorms causes from *People, Information, Process* and *Systems.*

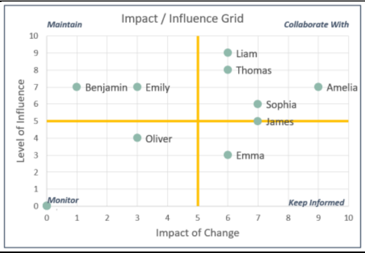


- **Sensitivity Analysis** Or Tornado Charts – shows the impact (positive and negative) on an item.



- **Simulations** Monte Carlo Simulation runs many different scenarios with slightly different inputs to get a range of possible outcomes.

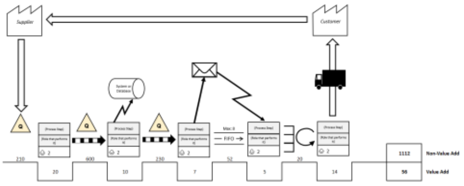
- **Stakeholder Analysis** Prioritising our stakeholders by their Influence and Impact.



- **SWOT Analysis** Analysing our Strengths, Weaknesses, Opportunities and Threats.

- **Value Stream Mapping**

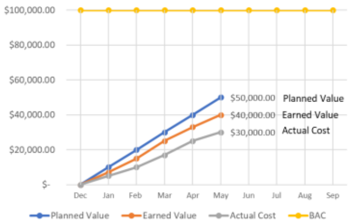
Mapping the Customer journey from their order to delivery, and removing “non-value add” activities.



- **Variance Analysis**

Cost Variance: **EV - AC**
Schedule Variance: **EV – PV**

Cost Performance Index: **EV / AC**
Schedule Performance Index: **EV / PV**
More than 1 = good.



Meetings and Events

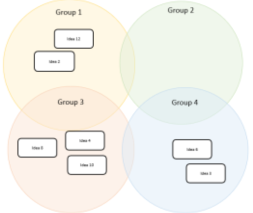
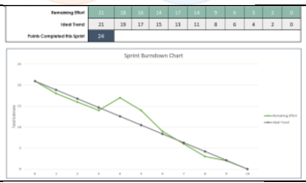
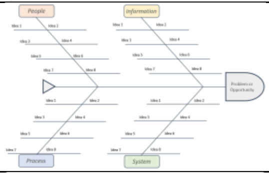
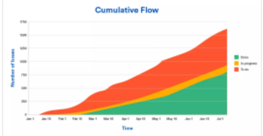
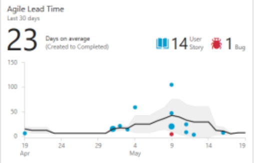
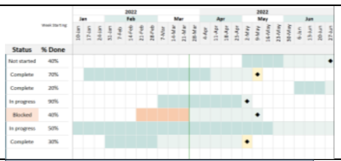
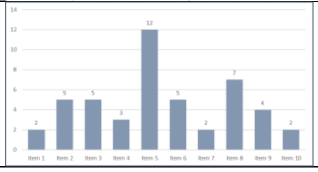
Agile	Common	Other
Iteration (sprint) planning	Bidder Conferences	Kick-off meeting
Backlog refinement	Lessons Learned	Steering Committee
Daily stand-ups	Planning Meetings	Change Control Board
Iteration (sprint) review	Project closeouts	Project Review
Retrospectives	Release Planning	Status Meeting
	Risk Reviews	

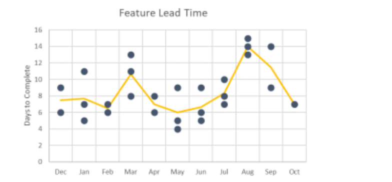
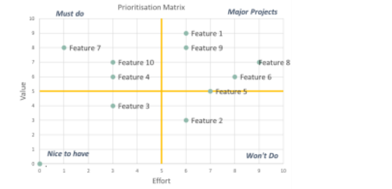
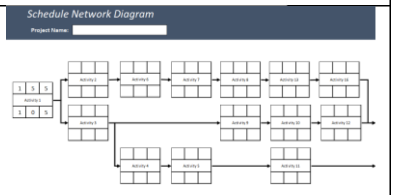
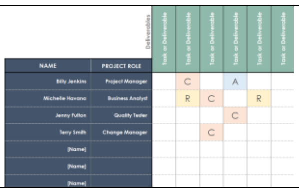
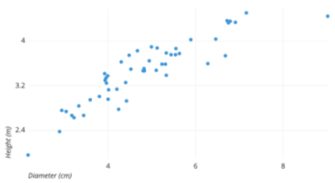
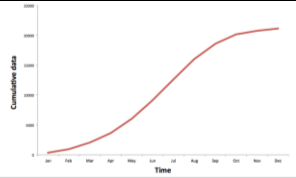
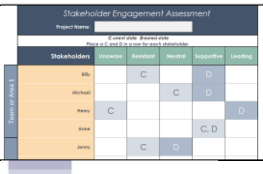
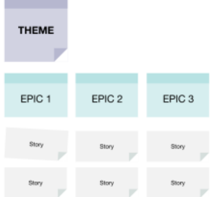
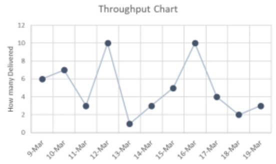
Commonly Used Artifacts

Use **Plans** when you need to see your process or what to do next,
Logs and Registrars to take an action and move forward,
Strategy documents to report on your project and understand business value.

Plans	Logs and Registrars	Strategy
Change Control Plan	Assumption log	Business case
Communications management plan	Backlog	Business model canvas
Cost Management Plan	Risk adjusted backlog	Project brief
Iteration (sprint) plan	Change log	Project charter
Procurement Management Plan	Issue Log	Project vision statement
Project Management Plan	Lessons learned register	Roadmap
Quality Management Plan	Risk register	
Release Plan	Stakeholder register	
Requirements Management plan		
Resource Management Plan		
Risk Management Plan		
Scope Management Plan		
Schedule Management Plan		
Stakeholder Engagement Plan		
Test plan		

Visual Data and Information Artifacts

• **Affinity Diagram**	To group ideas in similar categories.	
• **Burnup/Burndown Chart**	To show the planned trend of work, versus actual trend of work.	
• **Cause and Effect Diagram**	To find the root cause of a problem.	
• **Cumulative Flow Diagram**	To show features in various stages over time (e.g. in analysis, development, sign off)	
• **Cycle Time Chart**	Shows the average cycle time of items completed (time to complete a task or smaller piece of the feature)	
• **Dashboards**	Shows the progress or performance of the project.	
• **Flowchart**	Shows process or information flows.	
• **Gantt Chart**	A project schedule showing activities on a calendar.	
• **Histogram**	Shows number of different items.	
• **Information Radiator**	Project information in the team area.	

- **Lead Time Chart**	The time of features from customer order to delivery.	
- **Prioritisation Matrix**	Where features are prioritised by Effort and Value.	
- **Schedule Network Diagram**	Shows the relationships between project activities.	
- **Requirements Traceability Matrix**	Traces the customer requirements to their deliverables.	
- **Responsibility Assignment Matrix**	Shows who is assigned to each work package or activity.	
- **Scatter Diagram**	Shows relationships between two variables.	
- **S-Curve**	Shows the cumulative probability of something occurring (until it reaches 100%)	
- **Stakeholder Engagement Assessment Matrix**	Shows our Stakeholders as Unaware, Resistant, Neutral, Supportive or Leading.	
- **Story Map**	Agile method of breaking down the features into User Stories, by customer functionality.	 THEME EPIC 1 EPIC 2 EPIC 3 Story Story Story Story Story Story
- **Throughput Chart**	Shows the number of items completed each day.	

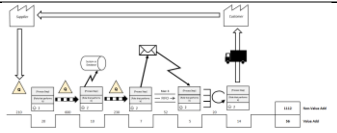
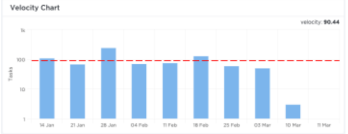
NAME	PROJECT ROLE	Task or Deliverable	Task or Deliverable	Task or Deliverable	Task or Deliverable	Task or Deliverable	Task or Deliverable
Billy Jenkins	Project Manager		C		A		
Michelle Havana	Business Analyst		R	C		R	
Jenny Fulton	Quality Tester				C		
Terry Smith	Change Manager			C			
[Name]							
[Name]							
[Name]							

Stakeholder Engagement Assessment

Project Name:

Current state Desired state
Place a C and D in a row for each stakeholder

	Stakeholders	Unaware	Resistant	Neutral	Supportive	Leading
Team or Area 1	Billy		C		D	
	Michael			C	D	
	Henry	C				D
	Anne				C, D	
	Jenny		C	D		

• **Use Case**	Describes how a user interacts with a system to achieve their goal.	Actor System Actions . . . Goal
• **Value Stream Map**	Shows the customer journey from order to delivery, noting non-value add activities so we can improve.	
• **Velocity Chart**	Shows the team Velocity in Story Points – how much the team completes each sprint.	

Reports

Our team will create reports to **share information and progress**, and facilitate decisions with project Stakeholders.

- Quality Report: To show defects found, their resolution and progress of quality activities.
- Risk Report: For overall project risk status and information on major risks.
- Status Report: For overall project status.

Agreements and Contracts

- **Fixed Price Contracts**
  - **Firm Fixed Price**
  - **Fixed Price Plus Incentive Fee**
- **Cost Reimbursable Contracts**
  - **CPAF**
  - **CPFF**
  - **CPIF**

A Final Note

Congratulations! You have made it to the end of the PMP Prep Course Summary Notes. I hope you enjoy reviewing these notes, and you enjoy this course, videos and practice questions.

If you practice and study every day, I truly believe that you can pass the PMP exam. Thousands of people have already done so using these materials, and I know you can do it!

Remember:

- The exam is a marathon, not a sprint.
    - Do at least two full practice exams (of at least 200 questions) in the two weeks leading up to your exam (limit four hours each).
- Make sure you answer every question. Do not leave any question blank before you finish the exam!
    - If you are not sure of an answer, flag the question and return to it at the end. You will likely get some clues or jog your memory from other questions as you proceed.
- On your exam day, make sure you have had a healthy breakfast (with protein if possible) and have had enough water. Both are good for your focus and your brain health.
    - Also, go to the bathroom before you begin – the test is long!

Please be sure to drop me a message on Udemy, LinkedIn and/or YouTube when you have passed your exam. You are doing the right things, and I believe in you!

Best wishes,

David McLachlan

Keep in mind the Exam Content Outline (ECO)

In-depth content for managing projects is found in the **PMBOK and Process Group Guides**, but on the exam they fall within the below Exam Content Outline (ECO). **Answer your exam questions with these in mind:**

People	
Task 1	**Manage Conflict** • Interpret the source and stage of the conflict • Analyse the context for the conflict • Evaluate/recommend/reconcile the appropriate conflict resolution solution
Task 2	**Lead your Team** • Set a clear vision and mission • Support diversity and inclusion (e.g., behaviour types, thought process) • Value servant leadership (e.g., relate the tenets of servant leadership to the team) • Determine an appropriate leadership style (e.g., directive, collaborative) • Inspire, motivate, and influence team members/stakeholders (e.g., team contract, social contract, reward system) • Analyse team members and stakeholders' influence • Distinguish various options to lead various team members and stakeholders
Task 3	**Support Team Performance** • Appraise team member performance against key performance indicators • Support and recognize team member growth and development • Determine appropriate feedback approach • Verify performance improvements
Task 4	**Empower your Team and Stakeholders** • Organize around team strengths • Support team task accountability • Evaluate demonstration of task accountability • Determine and bestow level(s) of decision-making authority
Task 5	**Ensure Team Members are Adequately Trained** • Determine required competencies and elements of training • Determine training options based on training needs • Allocate resources for training • Measure training outcomes

Task 6	**Build a Team** • Appraise stakeholder skills • Deduce project resource requirements • Continuously assess and refresh team skills to meet project needs • Maintain team and knowledge transfer
Task 7	**Address and remove blockers for the team** • Determine critical impediments, obstacles, and blockers for the team • Prioritize critical impediments, obstacles, and blockers for the team • Use network to implement solutions to remove impediments, obstacles, and blockers for the team • Re-assess continually to ensure impediments, obstacles, and blockers for the team are being addressed
Task 8	**Negotiate Project Agreements** • Analyse the bounds of the negotiations for agreement • Assess priorities and determine ultimate objective(s) • Verify objective(s) of the project agreement is met • Participate in agreement negotiations • Determine a negotiation strategy
Task 9	**Collaborate with Stakeholders** • Evaluate engagement needs for stakeholders • Optimize alignment between stakeholder needs, expectations, and project objectives • Build trust and influence stakeholders to accomplish project objectives
Task 10	**Build a Shared Understanding** • Break down situation to identify the root cause of a misunderstanding • Survey all necessary parties to reach consensus • Support outcome of parties' agreement • Investigate potential misunderstandings
Task 11	**Engage Virtual Teams** • Examine virtual team member needs (e.g., environment, geography, culture, global, etc.) • Investigate alternatives (e.g., communication tools, colocation) for virtual team member engagement • Implement options for virtual team member engagement • Continually evaluate effectiveness of virtual team member engagement

Task 12	**Define Team Ground Rules** • Communicate organizational principles with team and external stakeholders • Establish an environment that fosters adherence to the ground rules • Manage and rectify ground rule violations
Task 13	**Mentor your Team and Stakeholders** • Allocate the time to mentoring • Recognise and act on mentoring opportunities
Task 14	**Promote Team Performance with Emotional Intelligence** • Assess behaviour through the use of personality indicators • Analyse personality indicators and adjust to the emotional needs of key project stakeholders
Processes	
Task 1	**Execute your project with the urgency required to deliver Business Value** • Assess opportunities to deliver value incrementally • Examine the business value throughout the project • Support the team to subdivide project tasks as necessary to find the minimum viable product
Task 2	**Manage Communications well** • Analyse communication needs of all stakeholders • Determine communication methods, channels, frequency, and level of detail for all stakeholders • Communicate project information and updates effectively • Confirm communication is understood and feedback is received
Task 3	**Assess and Manage Risk** • Determine risk management options • Iteratively assess and prioritize risks
Task 4	**Engage your Stakeholders** • Analyse stakeholders (e.g., power interest grid, influence, impact) • Categorize stakeholders • Engage stakeholders by category • Develop, execute, and validate a strategy for stakeholder engagement

Task 5	**Plan and Manage your Budget and Resources** • Estimate budgetary needs based on the scope of the project and lessons learned from past projects • Anticipate future budget challenges • Monitor budget variations and work with governance process to adjust as necessary • Plan and manage resources
Task 6	**Plan and Manage your Schedule** • Estimate project tasks (milestones, dependencies, story points) • Utilize benchmarks and historical data • Prepare your schedule based on your chosen methodology • Measure ongoing progress based on methodology • Modify schedule, as needed, based on methodology • Coordinate with other projects and other operations
Task 7	**Plan and manage quality of products/deliverables** • Determine quality standard required for project deliverables • Recommend options for improvement based on quality gaps • Continually survey project deliverable quality
Task 8	**Plan and manage scope** • Determine and prioritize requirements • Break down scope (e.g.,WBS, backlog) • Monitor and validate scope
Task 9	**Integrate project planning activities** • Consolidate the project/phase plans • Assess consolidated project plans for dependencies, gaps, and continued business value • Analyse the data collected • Collect and analyse data to make informed project decisions • Determine critical information requirements
Task 10	**Manage project changes** • Anticipate and embrace the need for change (e.g., follow change management practices) • Determine strategy to handle change • Execute change management strategy according to the methodology • Determine a change response to move the project forward

Task 11	**Plan and manage procurement** • Define resource requirements and needs • Communicate resource requirements • Manage suppliers/contracts • Plan and manage procurement strategy • Develop a delivery solution
Task 12	**Manage project artifacts** • Determine the requirements (what, when, where, who, etc.) for managing the project artifacts • Validate that the project information is kept up to date (i.e., version control) and accessible to all stakeholders • Continually assess the effectiveness of the management of the project artifacts
Task 13	**Determine appropriate project methodology/methods and practices** • Assess project needs, complexity, and magnitude • Recommend project execution strategy (e.g., contracting, finance) • Recommend a project methodology/approach (i.e., predictive, agile, hybrid) • Use iterative, incremental practices throughout the project life cycle (e.g., lessons learned, stakeholder engagement, risk)
Task 14	**Establish project governance structure** • Determine appropriate governance for a project (e.g., replicate organisational governance) • Define escalation paths and thresholds
Task 15	**Manage project issues** • Recognise when a risk becomes an issue • Attack the issue with the optimal action to achieve project success • Collaborate with relevant stakeholders on the approach to resolve the issues
Task 16	**Ensure knowledge transfer for project continuity** • Discuss project responsibilities within team • Outline expectations for working environment • Confirm approach for knowledge transfers

Task 17	**Plan and manage project/phase closure or transitions** • Determine criteria to successfully close the project or phase • Validate readiness for transition (e.g., to operations team or next phase) • Conclude activities to close out project or phase (e.g., final lessons learned, retrospective, procurement, financials, resources)
Business Environment	
Task 1	**Manage Project Compliance** • Confirm project compliance requirements (e.g., security, health and safety, regulatory compliance) • Classify compliance categories • Determine potential threats to compliance • Use methods to support compliance • Analyse the consequences of noncompliance • Determine necessary approach and action to address compliance needs (e.g., risk, legal) • Measure the extent to which the project is in compliance
Task 2	**Deliver Project Benefits and Value** • Investigate that benefits are identified • Document agreement on ownership for ongoing benefit realization • Verify measurement system is in place to track benefits • Evaluate delivery options to demonstrate value • Appraise stakeholders of value gain progress
Task 3	**Evaluate and Address Business Environment Changes for Impact on Scope** • Survey changes to external business environment (e.g., regulations, technology, geopolitical, market) • Assess and prioritize impact on project scope/backlog based on changes in external business environment • Recommend options for scope/backlog changes (e.g., schedule, cost changes) • Continually review external business environment for impacts on project scope/backlog
Task 4	**Support Organisational Change** • Assess organisational culture • Evaluate impact of organizational change to project and determine required actions • Evaluate impact of the project to the organization and determine required actions