Most up-to-date version is on another page. Click on the link to go to the updated study guide. If you are unable to get to it then you can comment here and let me know: CompTIA Project+ Study Guide.docx

Terms

- **Assumptions** – Factors you believe to be true. Ex: "Revenues will grow at 5% per year"
- **Change Request** - Requests to modify a project's activities or resources, which would lead to a change in the project's scope, budget, or timeline.
- **CI/CD** - A software development method combining app and platform updates (rapidly committed to production) with code updates (rapidly committed to a code repository or build server).
- **Cloud Computing** - Computing architecture provides on-demand resources with high availability, scalability, and elasticity attributes billed to customers based on metered utilization.
- **Cloud Service Providers** - Organizations providing infrastructure, application, and/or storage services via an "as a service" subscription-based, cloud-centric offering.
- **Enterprise Resource Planning (ERP)** - Software that enables an organization to manage services, personnel, and IT resources.
- **Feasibility** - A feasibility study to analyze the hardware, software, facilities, and databases needed for a proposed project.
- **Functional Manager** - Individuals who are part of management on the administrative or functional side, such as human resources, finances, accounting, or even procurement of the business in the organization. They sometimes act as subject matter experts or may provide services needed for the project.
- **Gantt Chart -** A bar chart showing a project schedule, including start and finish dates for project elements.
- **Incremental Development** - An approach that breaks the software development process down into small, manageable portions known as "increments." Each increment builds on the previous version so that improvements are made step by step.
- **Issue** – Something that has already happened and is impacting the project. Risks turn into issues.
- **Iterative and Incremental Development** - A method that builds an incremental model developed in multiple cycles of iterations. A product may begin with relatively small components or steps, and incremental milestones are made during each cycle of the iterations until the final product is achieved.

- **Iterative Development** - A process by which product development, such as software development, is broken down from a large application into smaller chunks. A product or component is designed, developed, and tested repeatedly.

- **Portfolio** – Collection of programs, sub-portfolios, projects, and even operations.

- **Portfolio Management** – Selecting the right projects to align with the organization's strategic objectives.

- **Processes** - Activities that underlie the effective practice of project management; they include all the phases of concept/discovery, initiation, planning, execution, and closing a project.

- **Product Scope –** The specific features and functions that characterize our creation.

- **Program** – Group of related projects.

- **Program Management** – Focuses on interdependencies between projects.

- **Project** – Has a single objective that clearly explains what it intends to accomplish.

- **Project Budget** - The total financial sum available to pay for a project's expenses, including the cost estimates and additional reserves to cover issues.

- **Project Scope –** The work to deliver the product with the specified features and functions.

- **Project Team** - Consists of the project manager, the project management team, and other team members. The project team contains people from different groups who possess knowledge on specific subjects or have unique skill sets to carry out project work.

- **Requirements Management Plan -** A document that describes how project requirements will be analyzed, documented, and managed throughout the project life cycle.

- **Risks** – Events that you think could happen. It is uncertain if it will happen.

- **Schedule Baseline –** The approved version of the schedule that can be changed using formal change control procedures and is used as the basis for comparison to actual results.

- **Scope** – The definition of a project's objectives and deliverables. Defines what will be done to complete the project successfully.

- **Scope Baseline –** The approved version of the scope statement, WBS, and its associated WBS dictionary that can be changed using formal change control procedures.

- **Scope Creep -** Uncontrolled changes or continuous growth in a project's scope.

- **Scope Statement** – Officially declares what's going to be done. A description of all the major deliverables and any exclusions.

- **Stakeholder Engagement -** Engaging individuals or groups with a vested interest in the project.

- **Team Logistics** - The practice of providing materials and facilities needed by the team to accomplish their tasks.

- **The Kickoff Meeting** - The first official meeting between the project team and the sponsor.

    - The goals of the meeting are to make introductions, communicate the project purpose/goals, set expectations, clarify roles and responsibilities, and ask questions.

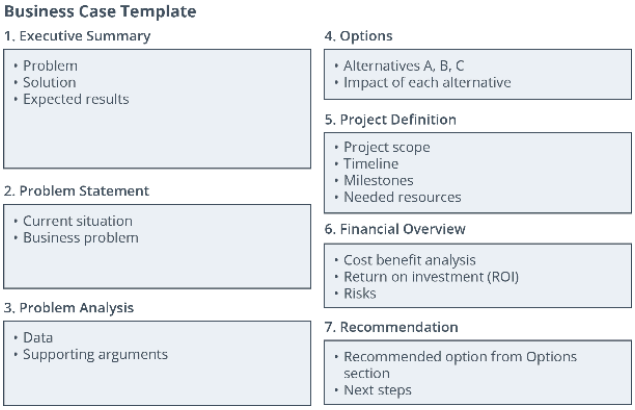
- **Triple Constraints –** Scope (features, functionality), cost (resources, budget), and time (schedule). Sometimes referred to as the Iron Triangle.

- **Work Breakdown Structure (WBS) -** A hierarchical decomposition of the total scope of work to accomplish project objectives.

Business Case

AKA **business objective** or **business justification –** A value proposition for a proposed project that may include financial and non-financial benefits.

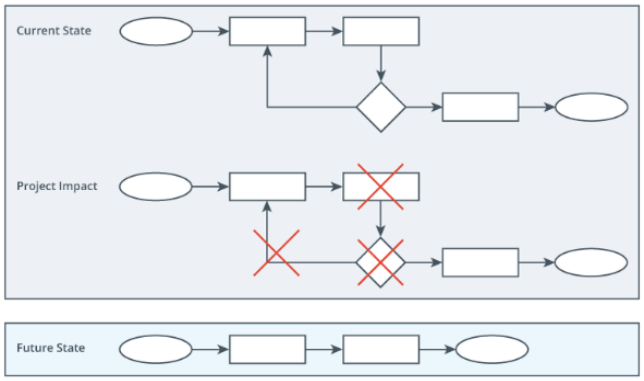
Summarizes information about the project and serves as the first project proposal. You will share the business case with leaders to obtain approval to start a project and secure funding and resources. A business case presents an objective analysis of a business problem, potential solutions, and financial impact. The goal of your BC is to have business leaders agree to allocate resources to your project.



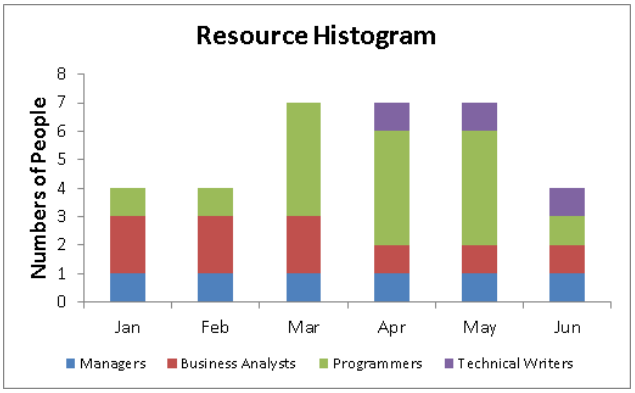
1. **Executive Summary** – Brief synopsis of the rest of the business case.
2. **Problem Statement** – Shares more details about the problem and should be concise and shorter than a paragraph.
3. **Problem Analysis** – Provides more context about why the project should be prioritized.
4. **Options** – Describes several approaches to solving the problem.
5. **Project Definition** – Includes additional relevant information about the proposed project.
6. **Financial Overview** – Covers the relevant economic impacts of the project. Ex: Costs, money source, benefits of project.
7. **Recommendation** – Narrow the options to the best solution and justify why this approach is the best business case.

Charts

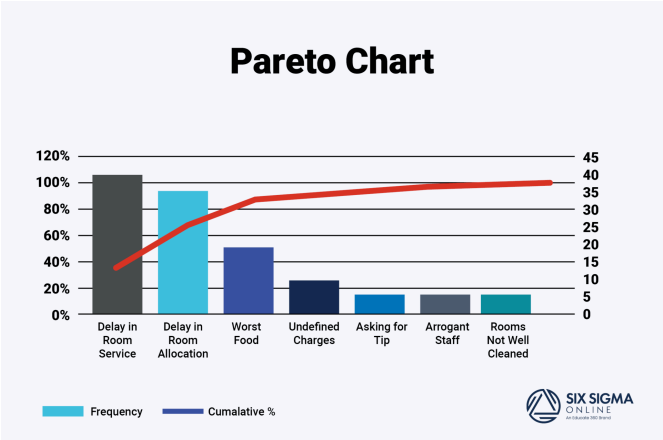
Process Flowchart – Shows the sequence of events and the flow of inputs and outputs between elements in a process or system.



Histogram Chart – This can visualize the number of times something occurred or the amount of something.



Pareto Chart – A type of histogram with the data sorted.



Responsibility Assignment Matrix (RAM) – Provides clarity and structure about the roles of key stakeholders. Helps you establish stakeholder engagement and buy-in early in the project—**initiation** phase.

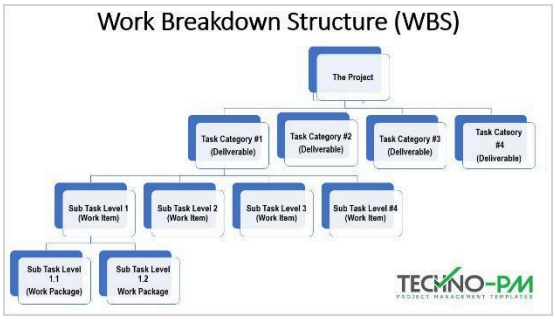
Deliverable	Stakeholder Responsible
System Design	Albest
Backend Development	Charlie
Data Migration	Sowatee
Desktop Apps	Mason
Mobile Apps	Junior
User Training	Chris

RACI Chart – A type of RAM. Responsible, Accountable, Consulted, Informed.

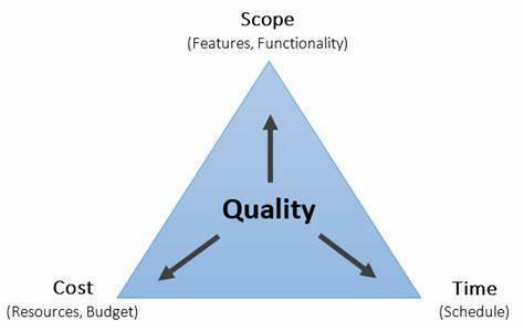
1. **Responsible** – The person **doing the work** or the management responsible.
2. **Accountable** – The person who **approves** or signs off the work.
3. **Consulted** – Someone who **provides input or information** for the work. It can be multiple people.
4. **Informed** – Someone who needs to be **kept informed** about the work. It can also be multiple people.

Work Breakdown Structure (WBS) – Provides a framework of deliverables. Breaks down projects (decomposes) into work packages (deliverables). **WBS Dictionary** – Supporting document providing more detail. Benefits of the WBS include:

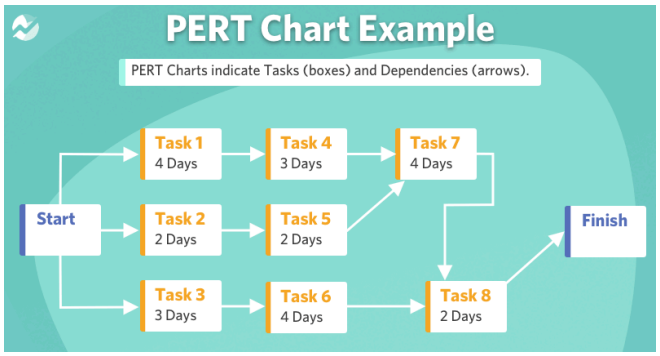
- Provides a clear picture of what needs to be done.
- Acts as a communication tool.
- Reduces the risk of missing out on important work.
- Encourages transparency and accountability.
- Helps with tracking the impact of issues.
- Provide input to the project schedule.



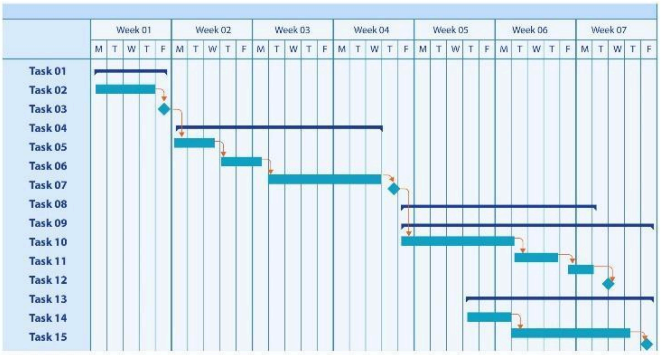
Iron Triangle – Lists the 3 main constraints. Will have to sacrifice one for the other two.



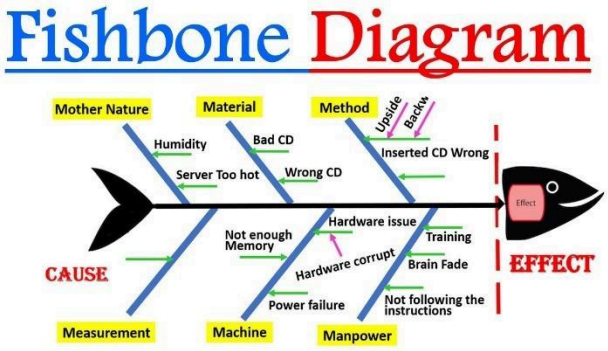
PERT Chart – Program Evaluation and Review Technique. Used to visualize the project schedule tasks and dependencies.



Gantt Chart – Another visualization that shows the timeline of the project.

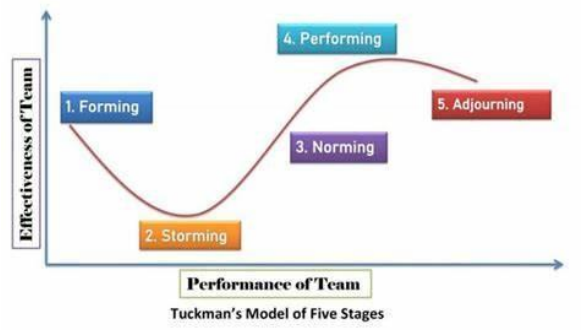


Ishikawa – Cause-and-Effect diagram. AKA Fishbone diagram.



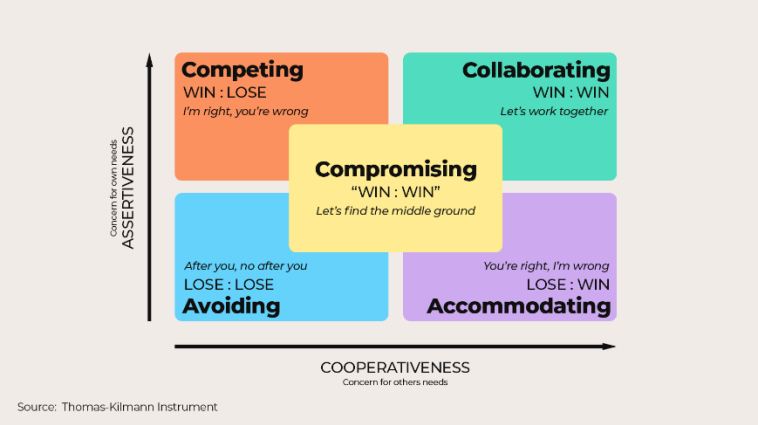
The Tuckman Ladder – A 5-step process for team development.

1. **Forming** – Bringing together a group who coordinate their individual efforts. This is where a team objective needs to be set.
2. **Storming** – Team members clash, and performance can dip.
3. **Norming** – Connecting with coworkers. Performance starts to creep up.
4. **Performing** – Team members become committed to the team goal.
5. **Adjourning** – At the end of the project, team members will go their separate ways.

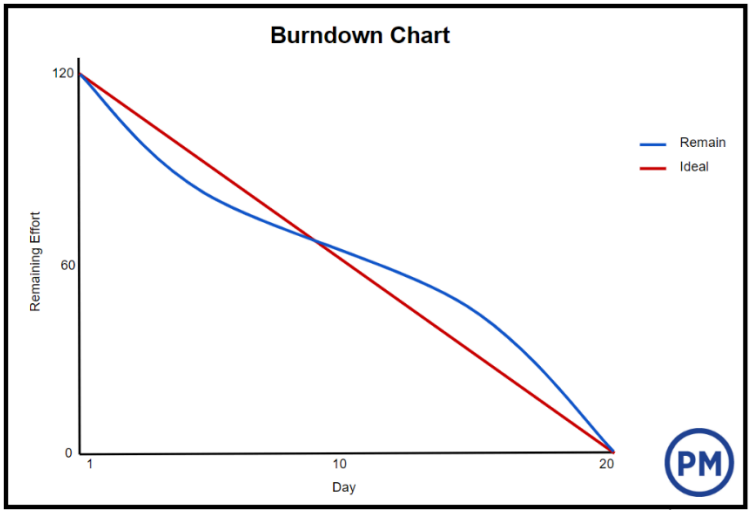


The Thomas-Kilman Model – Managing conflict resolution. 5 styles:

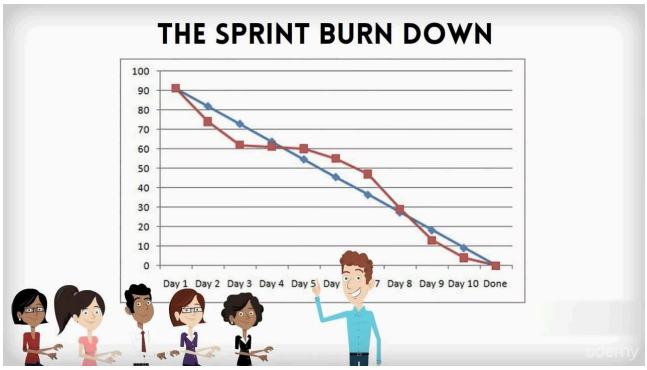
1. **Avoiding** – Avoiding conflict can be useful if the atmosphere is emotionally charged or if it isn't the right time.
2. **Forcing** – When someone forces another to do something. It is only used as a last resort when a quick resolution is needed for a high-priority issue. This could lead to more conflicts.
3. **Compromising** – The two individuals compromise on a solution. Neither party gets what they want.
4. **Accommodating/Smoothing** – One person gives in. It is best when there are extenuating circumstances or when preserving the relationship is more important than the goal.
5. **Collaborating** – Working together to find a solution to the problem.



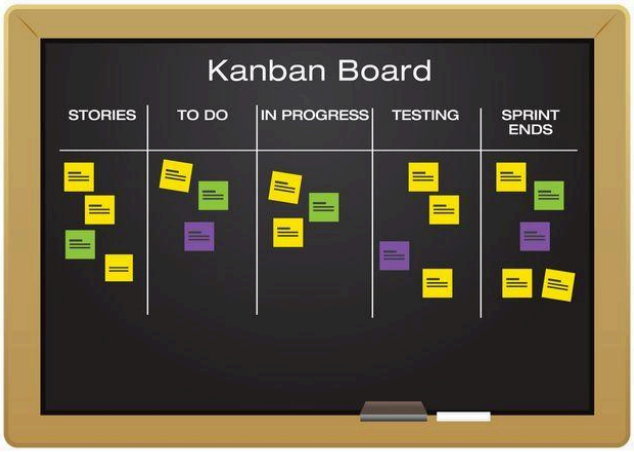
Budget Burndown Chart – Displays the expected rate of money to be spent vs the money spent.



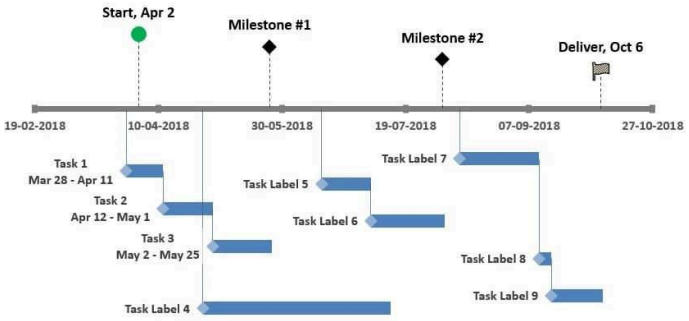
Sprint Burndown Chart – Displays the amount of work left to perform vs the expected amount.



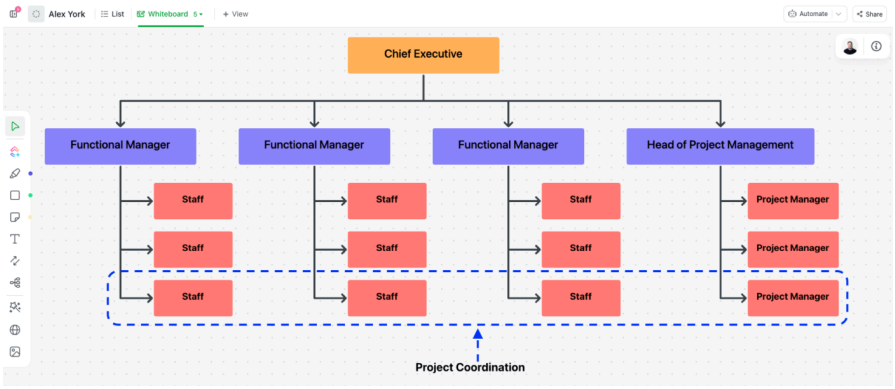
Kanban Board (Task Board) – Visualizes the status of tasks.



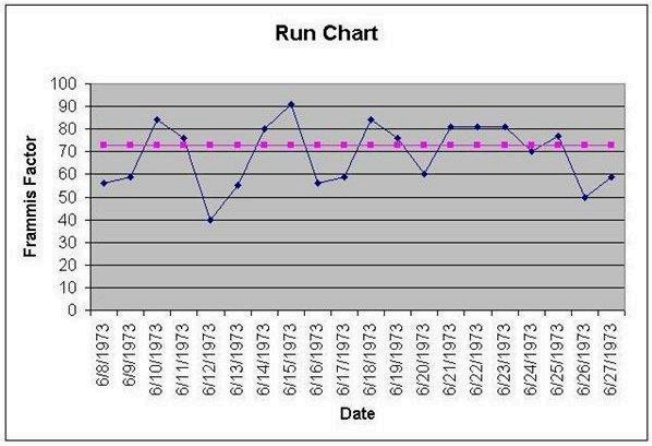
Milestone Chart – Great for providing an overview for key stakeholders. Marks significant points in the schedule.



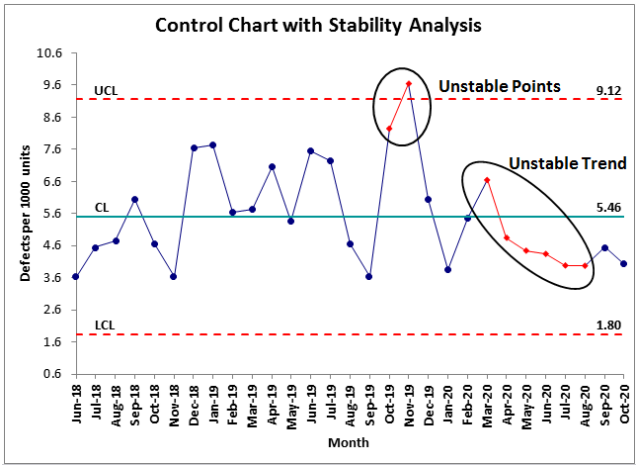
Project Organization Chart – Shows the roles and who is reporting to whom.



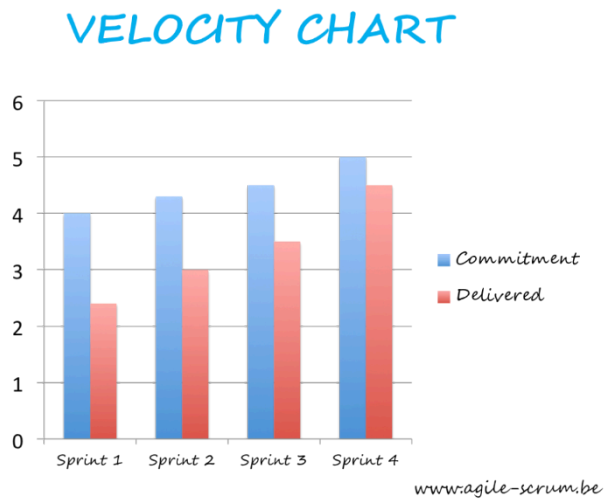
Run Chart – A line chart plotting data over time. It can be used to plot any information you want.



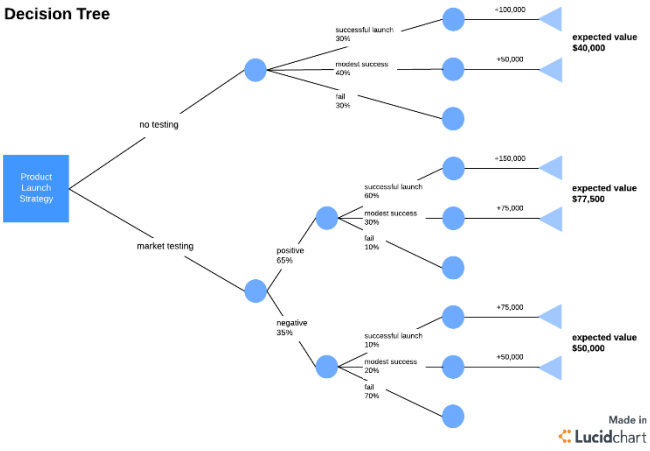
Control Chart – A run chart with the upper and lower control limits added.



Velocity Chart – An agile term. It shows how much work the team can complete in a sprint.



Decision Tree – A map of the possible outcomes of a series of related choices.



Software Tools

Can take advantage of local applications or cloud-based applications. Use whatever is right for the situation.

Word processing: Word/Google Docs

Spreadsheet: Excel/Google Sheets

Presentation Program: PowerPoint/Google Slides

Charting/Diagramming: MS Office or Google documents

Project Management Software: MS Project

Time Tracking: Spreadsheet or special tools

Ticketing/Case Management: Many options

Collaboration Tools: Google Drive, whiteboard

Communication Tools

Synchronous – Real-time communication.

Asynchronous – Delay in responses.

	Synchronous	Asynchronous
Email		X
SMS/Chat	X	X
Telephone	X	
Face-to-Face Meetings	X	
Video Meetings	X	
Enterprise social media		X

Meeting Types/Tools

Types

- **Collaborative**:
    - **Workshops** – Collaborative meetings where attendees are asked to participate in exercises, brainstorming sessions, and so on for creating new processes, problem-solving, identifying risk, and more.
    - **Focus Groups** - Focus groups are often used for market research to understand the attendees' responses and reactions to the topic. Attendees are carefully chosen ahead of time to participate in the focus group. There are numerous factors the researchers will use to choose these attendees, which are based on the subject matter they are researching
    - **Reviews** -
    - **Brainstorming** - This is a meeting where the participants are asked to generate ideas or solutions related to the topic. The facilitator will typically address one topic at a time and ask participants to provide succinct responses, usually one word or one sentence. These ideas or responses should be generated quickly, without much deep thinking. The facilitator will group the responses into common themes and then move on to the next question.
    - **Joint Application Development (JAD)** – Often used on agile development projects. The meeting focuses on developing the requirements for the system or application. It is held to focus

on the business need or the problem they are trying to solve and to create a deep understanding of the requirements among all team members.

- **Informative:**
    - **Demos/Presentations** – Informative meetings where the audience can view a demonstration of how a product works, examine a prototype or participate in how-to topics.
    - **Stand-up** - They are used on agile projects and are typically performed daily
    - **Status Updates -**
- **Decisive**
    - **Backlog Refinement** -
    - **Task Setting** - If you are working in a waterfall methodology, tasks are usually assigned by the team's supervisor performing the work. Or perhaps the tasks are assigned during a team meeting. This occurs in sprint planning meetings for agile projects. You'll recall that the sprint planning meeting occurs at the beginning of the sprint when team members pick the items they'll work on during the upcoming sprint.
    - **Committee Meetings** - The project sponsor, key project stakeholders, and the project manager typically comprise the steering committee. The steering committee is an oversight committee that ensures the project stays on track and helps resolve major issues. It is also an escalation point for the project team. When there are risks, decisions, budget items, or changes that may have a major impact on the project, they should be escalated to the steering committee for decision. This is also an escalation point for the project team when they cannot agree on a decision or action. The criteria for escalation should be outlined in the communications plan. The steering committee meeting is a decisive meeting type.

Tools

Real-time surveys/polls: This is especially useful for encouraging engagement.

Calendaring tools: For attendee availability.

Conferencing platforms: Online meetings.

Methods for Selecting a Project

Mathematical Modeling: Linear and nonlinear programming – known as **Constrained Optimization**.

Benefits Comparisons: E.g. ROI or scoring model.

Scoring Model: Identify relevant criteria, weigh them according to their importance, and then score each project accordingly. E.g.:

Criteria:	Weight	Project A	Project B
Profitability	5	3x5 = 15	4x5 = 20
Time to Market	3	5x3 = 15	2x3 = 6
Ease of Integration	1		

Opportunity Cost – What's the cost of the other opportunities given up by investigating the chosen project.

Formulas

- **Net Present Value (NPV)** – Uses an economic theory based on the assumption that the value of a sum of money today is more than it will be in the future. More bang for your buck now than later.
- **Return on Investment** – A % that shows how much money you will make by investing in something. Compares the financial benefit of the project to the cost. A positive number represents a profitable project; a negative number means the project lost money. ROI has 3 main values:
  1. **Benefit** – Money generated or explicitly saved due to the project. "**Financial Value**" or "**Revenues**".
  2. **Cost** – Represents all the money spent on the project. "**Project Cost**" or "**Project Expenses**".
  3. **Net Profit** – The difference between the benefit and the cost.

  ROI = (Net Profit / Cost) × 100

  ROI = (Revenue – Cost) / Cost × 100

  ROI = (Financial Value - Project Cost) / Project Cost × 100

- **Internal Rate of Return (IRR)** – A % number that shows if the investment will have a high or low rate of return.
- **Benefit-Cost Ratio (BCR)** – How much $ will you get back for every $ you invest. E.g., $150 for every 100 dollars spent.
- **Payback Period (PBP)** – How long it will take to pay back what you invested in the project.
- **Earned Value Management (EVM)**:

  1. **Planned Value (PV)**: The approved budget assigned to scheduled work.
  2. **Earned Value (EV)**: The value of work performed, measured by progress.
  3. **Actual Cost (AC)**: The cost incurred for the work performed during a specific period.
  4. **Cost Performance Index (CPI)**: The ratio of earned value to actual costs.
     Formula: **CPI = EV / AC**
     - If CPI > 1, the project is under budget; if CPI < 1, the project is over budget.
  5. **Schedule Performance Index (SPI)**: The ratio of earned value to planned value.
     Formula: **SPI = EV / PV**
     - If SPI > 1, the project is ahead of schedule; if SPI < 1, the project is behind schedule.

Environmental, Social, And Governance Factors

A project manager won't directly deal with these issues but rather steer someone else in the right direction to deal with them.

- **Environmental** – How the organization impacts the natural world. Considerations include landfill consumption, waste generation and disposal methods, pollution, habitat protection or destruction, and resource consumption. Ex: Combatting climate change and reducing emissions.
    - **Cultural Factors**:
        - The work culture of an agile team requires strong commitment, trust, and collaboration to achieve the value of agile. Agile adds complexity compared to waterfall but produces better products faster. Agile team members work cross-functionally, so they must learn new skills and continually collaborate.
        - Waterfall favors deep knowledge and role specialization so team members will move on and off the project as their expertise is needed.
    - **Developmental Factors**:
        - Waterfall's development process is linear and sequential: code is written, tested, and delivered. This means that developers, testers, and operations tend to work on the project at different times and do not collaborate as often.
        - Agile development is iterative and incremental, meaning they repeatedly create small functionality until a project is complete.
    - **Industry Standards**:
        - Companies that lean on documentation often choose waterfall or layer documentation requirements into an agile approach. Industry norms, such as the pace of change, also influence how organizations approach projects.
- **Social** – How the company develops relationships and treats people, including employees, people in the communities, and groups impacted by the company's products and services. Considerations include employee well-being, volunteering in the community, working conditions, employee pay, and community involvement. Ex: engaging in salary transparency, volunteering paid time, and creating safe working conditions.
- **Governance** – How the company operates, including its policies, transparency, and structure. Ex: A board's diversity, lobbying activities, political donations, and management structure.

Environmental	Social	Governance
Renewable fuels	Health and safety	Ethical standards
Greenhouse gas (GHG) emissions	Working conditions	Board diversity and governance
Energy efficiency	Employee benefits	Stakeholder engagement
Climate risk	Diversity and inclusion	Shareholder rights
Water management	Human rights	Pay for performance
Recycling processes	Impact on local communities	
Emergency preparedness		

- **Corporate Identity** – The company's vision, mission statement, values, and brand. It can incorporate its position on multiple **ESG** factors. A company cannot directly control its brand identity but can cultivate it by increasing or decreasing specific activities. For example, a company that wants a reputation for community involvement could sponsor local events, engage in the community, and advertise its participation.
- **Company Brand** – Describes how people perceive the company; describes what they remember, feel, and believe to be true when they think about a company. Ex: Tangible items – symbols, color schemes, memorable slogan/spokesperson. Intangibles – a general sense of how they feel about the company, general characteristics or traits, and a company's societal influence.

Work Types

- **Operational** – Routine, predictable, and repetitive.
- **Project** – Accomplishes something new. Every project must meet 3 criteria: uniqueness, specific reason/purpose, and temporary.
  1. **Unique** – Each project aims to complete a brand-new objective. It can be similar to other work but will differ in specific requirements, targets, or activities.
  2. **Reason/Purpose** – Projects exist for a specific reason.
  3. **Temporary** – Projects must have a beginning and end date. It can last for days or even decades. In comparison, operational work continues to repeat until something forces it to stop.

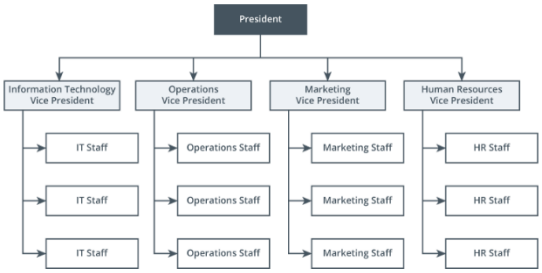
Organization Type	Operational Work	Project Work
Coffee Shop	• serve customers • Reorder supplies	• host a fundraiser • Develop a new menu
Event Planning	• process payroll • Hire employees	• sponsor a conference • Expand to a new city
Clothing Manufacturer	• Mass produce products • Ship products	• design a seasonal line • Launch an accessories product
Software Company	• Run security checks • Set up user accounts	• implement identity management • Establish single sign-on
Computer Repair	• Take help desk calls • Repair hardware	• Launch a marketing campaign • Increase supply chain efficiency

---

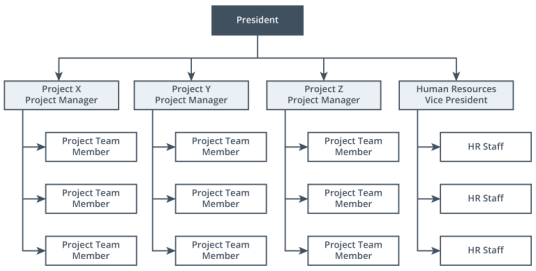
Organizational Structure Types

The compositional makeup of an organization that dictates how the various groups and individuals within the organization interrelate. There are 3 main types of organizational structures, each with its own use case.

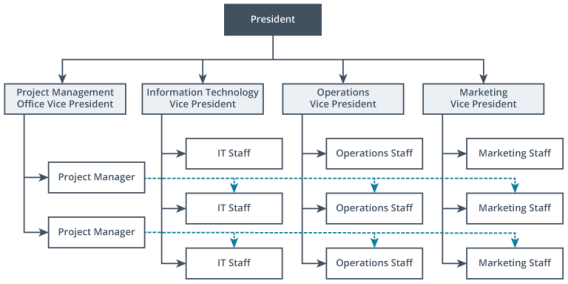
- **Functional Organization** – Reporting is hierarchical, with each reporting to a single manager. More suitable for small businesses.

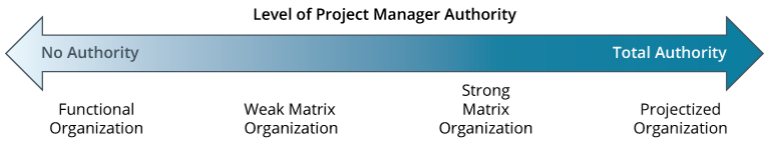


- **Projectized Organization** – AKA Project-Oriented. Pool resources around projects. A PM leads a group of people as long as a project exists. When the project ends, the team disbands and reforms around new projects. The structure is similar to a functional organization. However, it will have PMs instead of specialized managers. The PM has more control over this structure than a functional one.



- **Matrix Organization** – Most organizations are matrix organizations, which include functional leaders and specialized roles. However, a matrix organization also hires PMs and sponsors projects. Structured visually like a grid. Matrix organizations have two subtypes:
    1. **Weak Matrix** – The functional manager retains all budget and staff management responsibilities. In this sense, the PM has less control over the project.
    2. **Strong Matrix** – A PM has substantial control over the project. They may have management responsibilities for the budget and staff.





Roles

	Product Owner	Project Manager
Approach/Methodology	Agile	Waterfall
Accountability	Project success/management	Backlog management
Strategy	Develops the project's goals and ensures that maximum value is delivered to the end customer.	Focuses on building strategic roadmaps that make a business' vision and goals a reality.
Team Interaction	Provides support to team members on project-related tools and activities.	Acts as the main figure who guides the rest of the product team on agile processes, activities, and deliverables.
Relationship with stakeholders	Does not communicate directly with stakeholders	Has to implement a strategy that makes it easy for product team members and stakeholders to communicate with each other.
Risk Management	Adopts a long-term approach to project management that emphasizes minimal iterations throughout the project's life cycle.	Focuses on solving challenges throughout the various sprints within product development

- **Product Owner** – Work within agile teams. They identify how to improve a product and are accountable for creating the most valuable product possible. A product owner tends to own a product for the entire product's life cycle. Responsibilities include:
    - Creating a valuable product.
    - Work within a fixed time (iteration), fixed cost (development team cost), and flexible scope (whatever the team accepts into an iteration).
    - Providing product direction and vision.
- **Project Manager**: - Leads projects in any framework. They plan, organize, and keep teams on track throughout the project life cycle. Responsibilities include:
    - Build the project team and secure necessary resources.
    - Builds the project charter and defines the project scope.
    - Establish the required project logs and processes, including the issue log, change log, and risk register.
    - Ensure project deliverables are met.
    - Prepare and deliver project status updates.
    - Track the project's progress and status, including performance to schedule and performance to budget.

- **Business Analyst** – A project team member who closely understands business direction and company priorities and is familiar with the software environment, operating as a translator between business and IT.
    - Improves the final product.
    - Help define the project.
    - Gather business and technical requirements.
    - Ensure requirements stay aligned with project scope and business needs.
    - Interpret business requirements as technical requirements and vice versa.
    - Verify project deliverables against requirements.
    - Assist with testing and validation.
- **Architect** - A professional designer of various solutions. Organizations hire specialized architects who design and oversee different solution elements. For example, value stream architects arrange value streams, and solution architects design computer and networking systems.
    - Contribute to solution design.
    - Build system blueprints.
    - Evaluate systems against organization standards, such as information security.
- **Developer/Engineer** – Represent multiple roles that create the final product.
    - Write code to expected standards.
    - Build products according to the blueprints, project plan, or sprint backlog.
    - Report on development progress to the PM or product owner.
- **Tester/Quality Assurance** – Help ensure code quality.
    - Write tests against requirements.
    - Run tests on completed code to find defects and bugs.
- **Quality Assurance Specialist** - Professionals who improve an organization's quality standards through inspection and process improvement.
- **Product Manager**: A professional responsible for developing and improving a company's offerings; experts in customers' problems who develop product strategies and road maps to meet their needs. They serve a more strategic function than product owners or PMs.
- **Program Manager** – Leads programs. They keep the program aligned with organizational strategy and ensure it is within budget and on time. Program managers are like PMs of programs, though with a more

strategic slant. When your project is in a program, the program manager is an invaluable partner who will help you stay connected with other projects and align to broader strategies and timelines.

- **Project Management Office (PMO)** – Functional department for all PMs in a company. Evaluate new projects and assign them to a PM. Provides administrative support, such as maintaining archives, best practices, and project management tools. There are 3 PMO types:
  1. *Supportive* – Provides support when it's requested. In this model, the PMO is essentially a library of project information. It doesn't force adherence to standards and may not assign or evaluate projects. Do not report to the head of PMO.
  2. *Controlling* – More involved than a supportive PMO. It actively monitors project performance. It coordinates resource selection and allocates PMs to projects. It also coordinates communications and sets some project standards. However, this PMO does not have full authority, and its influence is limited by functional management. Matrix organizations often use this format.
  3. *Directive* – Most controlling type. Sets the rules for everybody in the company. This PMO has full authority over projects, standards, and procedures. It also prioritizes projects and allocates PMs for all projects in the company. A directive PMO focuses on organizational strategy and will start, cancel, or adjust projects to ensure strategic alignment. Reports to the head of the PMO.
- **Stakeholder** – Anybody with a vested interest in a project. People or groups affected by projects. Some stakeholders are more important than others when assessing their needs during a stakeholder analysis.
  - Understand the project's purpose and how it will affect them
  - Be transparent about their support or lack thereof for the project.
  - Provide timely feedback about the project's progress and direction.
  - *Internal*:
    - Sponsor, PMO, Program Manager, Team Members.
  - *External*:
    - Customers, End users, Suppliers, Shareholders, Competitors.
  - *Assessing Stakeholders*:
    - **Classification Grids**:
      - Power/Influence, Power/interest, Impact/Influence.

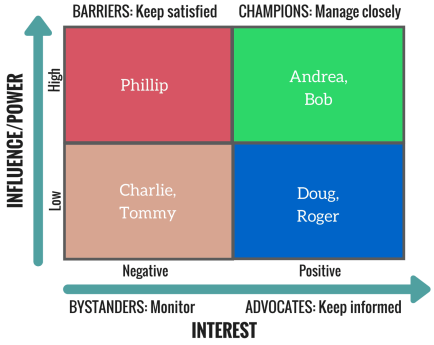
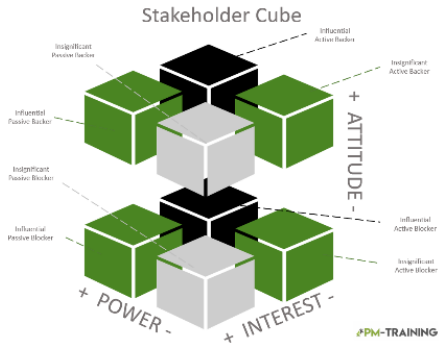
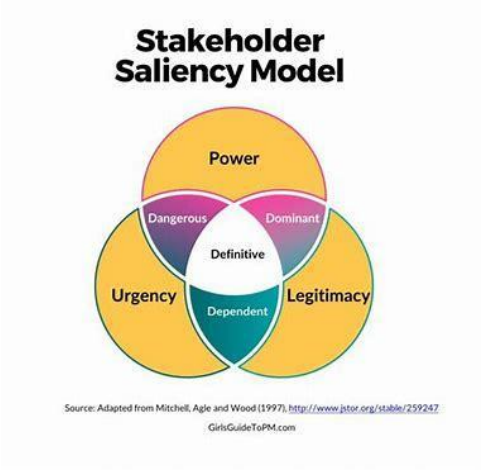


FIGURE 1. STAKEHOLDER INFLUENCE/INTEREST GRID

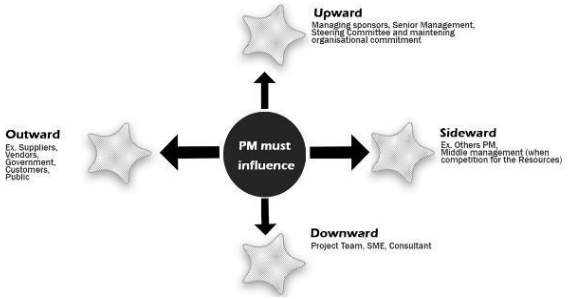
    - **Stakeholder Cube**:
      - 3D version for 3 factors – Power/Influence/Impact

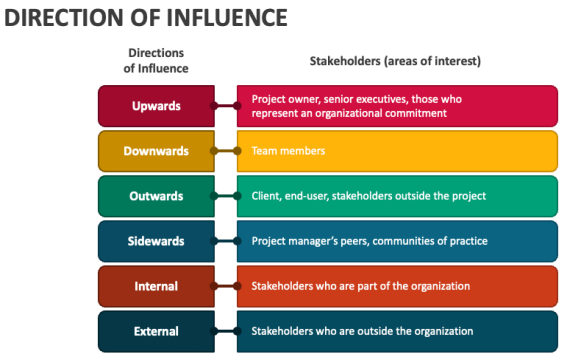


- **Salience Model**:



- **Directions of Influence**: The stakeholder's name and their direction of influence.





- **Senior Management** – Refers to the highest leadership level in an organization.

- **Subject Matter Expert (SME)** – A general label for anybody who is a respected expert in their field. The person you turn to when you need help.
- **Project Sponsor** – The person, group, or organization paying for the project. The sponsor is accountable for a project and is often a single senior management member but could also be a customer. Reviews and validates the initial business case. They are more familiar with the project than any other senior management member. Approves the Project Charter.
- **Customer** – Those who receive the benefits from the projects. They can be internal or external to the company.
- **End-User** – The person interacting with the project's final output.
- **Vendor** – Often referred to as "suppliers" or "business partners". External parties who enter into a contractual agreement with the organization and provide components or services needed for the project. Seller, contractor, and supplier are also used when referring to vendors.
    - Provide the work outlined in the contract. For instance, the deliverables could include delivery and setup, configuration, technical support, or advisory services.
    - Maintain the service-level agreements (SLAs) outlined in the contract.
    - Maintain clear lines of communication and productive relationships.

Project Life Cycle Phases

Series of phases a project goes through from start to finish

DIPEC

Phase	Activities
Discovery/Concept Preparation	▪ Return on investment (ROI) analysis ▪ Preparing the business case ▪ Seeing what vendors or contracts we already have in place that we could utilize ▪ Looking at high-level budgets
Initiation	<u>The formal authorization for the project to begin</u> ▪ Identifying stakeholders (those affected by our project) ▪ Reviewing existing artifacts (docs, templates) ▪ Developing the project charter (Formally authorizes the existence of the project) ▪ Develop a Responsibility Assignment Matrix (RAM). ▪ Develop a records management plan and define access requirements. ▪ Project kickoff meetings.
Planning	▪ Develop a project management plan ▪ Develop a detailed scope statement ▪ Define units or work ▪ Determine the budget

	▪ Assigning resources ▪ Developing a schedule
Execution	<u>Where the actual work takes place</u> ▪ Tracking and reporting on project work/budget ▪ Managing changes ▪ Coordinate resources ▪ Updating plan if changes are made ▪ Managing conflict
Closing	▪ Final sign-off of the project work ▪ Handing over to operations ▪ Closing contracts ▪ Document lessons learned

Discovery/Concept Phase

A **business case** is created regarding the **business needs**. The business need could be a system upgrade, legal requirements, or a customer request.

Analysis will be performed to investigate more information and risks involved.

A **feasibility study** will determine whether the project is doable or worth doing. This may include mathematical modeling and benefits comparisons.

Procurement Methods:

· **Build/Make:** Created "in-house" by your team.

· **Buy:** Outsource or buy a solution. A Make-or-Buy analysis could be performed to determine if it is more profitable to build/make or buy the solution.

· **Lease:** Some companies prefer to lease.

· **Subscription:** E.g. Cloud services.

Engaging Vendors – This is where processes will be performed to find the best vendors. E.g. RFI, RFP, RFQ, and RFB.

Initiation Phase

Initiation is the formal authorization for a new project to begin or for an existing project to continue into the next phase. Key activities according to CompTIA Project+ objectives are:

· Identify the preliminary scope statement

· Create the project charter

· Identify and assess stakeholders

- Develop a responsibility assignment matrix (RAM)
- Establish accepted communication channels
- Develop a records management plan
- Define access requirements
- Review existing artifacts
- Determine solution design
- Conduct the project kickoff

The **Project Charter** is the primary purpose of the initiation phase. This document issued by the project sponsor formally authorizes the project's existence. The sponsor is the person who provides resources and support for the project.

Project Charter	
Project purpose and high-level requirements, objectives, etc.…:	Will become more detailed once you start planning – **Product Vision**
Overall project risk:	Quantifiable and measurable
Summary schedule and budget:	High-level milestones – **Product Roadmap**
Key stakeholders:	
Success criteria and approval:	Criteria for whether the project is a success and therefore approved.
Exit criteria:	Under what conditions might the project need to be canceled.
Project manager name and level of authority	Official authorization to start detailed planning.

Identifying stakeholders in this initiation phase using models such as a classification grid is important. You will create a **Stakeholder Register** to identify the stakeholder, their requirements, and their classification.

Planning Phase

In the Planning phase, the project goals, objectives, and deliverables are refined into manageable work units. Project managers create time and cost estimates and determine resource requirements for each activity. Planning involves several other critical areas of project management, such as communication, risk, human resources, quality, and procurement. Some of the key activities in the Planning process group are as follows:

- Develop a detailed project scope statement
- Develop a project schedule
- Determine budget considerations
- Develop a quality assurance plan
- Develop a communication plan

- Assess and assign project resources
- Perform an initial risk assessment
- Assess the resource pool and assign project resources
- Develop a project management plan
- Train project team members
- Define units of work
- Develop a transition plan/release plan

Execution Phase

The execution phase is where the project's work is performed and monitored for adherence to the project management plan. Key activities in this phase include:

- Producing/verifying deliverables
- Implementing change management
- Managing vendors
- Tracking and reporting project results
- Updating project elements such as budget, risk, and timelines
- Managing conflict
- Monitoring the risks and issues log
- Performing QA activities such as phase gate reviews
- Monitoring the budget
- Project meetings and updates

Closing Phase

The primary purpose of the Closing phase is to validate deliverables and document the formal acceptance of the project work. Once that’s complete, a handoff occurs whereby the completed product or result of the project is turned over to the organization for ongoing maintenance and support. The Closing phase includes validating deliverables, signing off on the project, archiving project documents, handing off the product to the organization, releasing project team members, and reviewing lessons learned.

The key activities in the Closing process are as follows:

- Validating deliverables

- Closing contracts
- Removing access
- Releasing resources
- Holding the project closure meeting
- Writing the project closeout report
- Obtaining feedback and lessons learned
- Project sign-off
- Handing off the product to the organization
- Evaluating the project
- Archiving project documents
- Rewards and celebration

Project Termination Types:

1. **Starvation** – Resources needed to finish the project are cut off. The project died due to being deprived of its needed resources.

2. **Extinction** – The project was completed and handed over to the client. All objectives have been met, and all project activities have ended. OR… An unsuccessful project due to the project not passing regulations.

3. **Addition** – A successful project that turns into day-to-day operations. Will operate as an addition to the organization.

4. **Integration** – A successful project that turns into day-to-day operations. The project outcome becomes integrated throughout the organization.

Transition Plan – The goal is to identify what we will do to ensure the new owner has everything they need. This will include:

- The delivery date – Expected "go live" date.
- Knowledge transfer – Provide the knowledge that the owner will need to know.
- Documentation – Provide documentation for the project.
- Training – Another type of knowledge transfer.
- Assets transfer
- Maintenance requirements

- Warranty period
- Post-implementation support
- Ongoing costs

Release Planning – A term used in Agile environments to plan when to release features to end users.

Verification – The evaluation of whether a product complies with a regulation, requirement, specification, or imposed condition. E.g. Is the product correct? Verification is achieved through **Quality Control Processes**.

Validation – This is when we present the verified deliverables to the customer. Occurs AFTER verification. This is where the customer formally accepts the completed deliverables. The "seal of approval" that the deliverables align with the scope.

Lessons Learned – The knowledge gained during a project that shows how project events were addressed (or should be addressed in the future) to improve future performance.

Phase Gate Review – A review at the end of a phase, in which a decision is made to continue to the next phase.

Sprint Retrospective – Occurs in Agile. The team regularly reflects on becoming more effective and then tunes and adjusts its behavior accordingly.

Archive Documentation – Important for future reference.

Project Closeout Report – Summary level description, scope objectives, quality objectives, cost objectives, risks and issues, schedule performance, and summary of how the business needs were met.

Documents, Artifacts, and Chart Phases

A document is any piece of written information. It can be edited. It may be saved for a long or short period of time.

A record is a piece of evidence that is final and will be kept for a specific period.

Document Name	Document Description	Phase
Business Case	A value proposition for a proposed project. Justification for the project. Business needs, feasibility study.	Developed in Discovery/Concept but used throughout the entire project
Request for Information (RFI)	This document is often used to gather preliminary information from vendors before creating a formal **Request for Proposal (RFP)**. Does the vendor have the skills/capacity? Narrows down vendor selection.	Discovery/Engaging Vendors

Request for Proposal (RFP)	Detailed and complex design approach, timelines, and costs.	Discovery/Engaging Vendors
Request for Quote (RFQ)	Costs for specific items/services.	Discovery/Engaging Vendors
Request for Bid (RFB)	Bids are submitted	Discovery/Engaging Vendors
Project Charter	Authorizes the project to begin. The document issued by the project sponsor that formally authorizes the existence of a project. Contains high-level information about the project. Identify overall project risk.	Initiation
Preliminary Scope Statement	Project objectives are defined; the business problem the project will address is stated	Initiation
Stakeholder Register	List of stakeholders and their interests, expectations, and impact on the project.	Initiation
Responsibility Assignment Matrix (RAM)	Defines stakeholder responsibilities. Incorporates the use of the RACI chart	Initiation
Communication Plan	Documents the types of information needs the stakeholders have, when the information should be distributed, and how the information will be delivered	Planning
Project Schedule	Determines the start and finish dates for project activities and assigns resources to the activities	Planning
Scope Statement	Documents the product description, key deliverables, success and acceptance criteria, key performance indicators, exclusions, assumptions, and constraints	Planning
Product Backlog	An agile document. A prioritized list of required product functionality that provides a central view of what needs to be done and the order in which to do it. Higher-priority stories are at the top of the document. Items that go into the list include features (user stories), bug fixes, technical updates, knowledge gathering (spiking), and more.	Planning
Project Management Plan	Defines how the project is executed, monitored, and controlled. Consists of all the project planning documents, such as the charter, scope statement, schedule, and more.	Planning

Quality Management Plan	Describes how the project will achieve quality objectives.	Planning
Procurement Management Plan	Outlines how procurement will be managed in the project. Guides the acquisition of goods and services from external sources.	Planning
Risk Register	A list of all identified risks, planned response, priority, and owner. It includes risk descriptions, probability, impact, response strategies, and risk owners.	Created in the **Planning** phase but modified throughout the **Executing** phase
Issue Log	Tracks the resolution of issues and reassures stakeholders they are being dealt with. It will include the issue owner, priority, planned resolution, target resolution date, and status.	Executing
Change Log/Change Control Processes	It describes change requests and their disposition for the project	Executing
Status Report	A report to stakeholders on the status of the project deliverables, schedule, risks, issues, and more	Executing
Dashboard Information	An electronic reporting tool that lets users choose elements of the project to monitor project health and status	Executing
Meeting Agenda/Meeting Minutes	Meetings agendas describe the items to be discussed and addressed at upcoming meetings, and minutes recap what was discussed and the decisions made at the meeting	Executing
Project Closeout Report	Reports on the final closeout of all the phases of the project	Closing

- **Business Requirements**: The document defining a project's scope, success factors, constraints, and other information to achieve project goals.
- **Service-Level Agreements (SLAs)**: Outlined in the contract. Agreement that sets the service requirements and expectations between a consumer and a provider.
- **Vendor Documents**:
    - **Statement of Work (SOW)**: The details of the services or goods we want to purchase so the vendor knows what we ask of them.
    - **Terms of Reference (TOR)**: The description of the work requirements, including the scope, timeline, and resources.
    - **NDA** – Agreement to not disclose information.
    - **Cease and Desist Letter**: Stop someone from committing a violation and prevent them from doing it again.
    - **Memorandum of Understanding**: Useful for outlining terms but is not legally enforceable.
    - **Letter of Intent**: Describes the intention of what the buyer and vendor will do, usually before they enter a contract. It is not always legally binding but can carry weight.
    - **Master Service Agreement**: An umbrella agreement to set general terms between parties.

- **Service Level Agreement (SLA)**: Defines expectations of the vendor's service level.
- **Maintenance Agreement**: Describes what the vendor is expected to do to keep the goods or services running.
- **Warranty**: Vendor's guarantee of the product's performance.
- **Purchase Order**: Official document issued by a buyer committing to pay the vendor for the sale of the product/service.

Risk Responses

Negative:

- **Avoid** – To avoid the risk altogether. Taking action to eliminate the threat or remove its impact. Appropriate for threats with a high chance of occurring and a large impact.
- **Mitigate** – Reduce the impact or reduce the chance of it happening. It is more effective than repairing damage after the threat has occurred.
- **Transfer** – Move ownership of the risk to a third party. E.g. Insurance.
- **Accept** – No proactive action is taken. Appropriate for low-priority risks and may be an acceptable response.

Positive:

- **Exploit** – We ensure we take full advantage of the opportunity or take action to ensure it happens.
- **Enhance** – Increasing the impact of increasing the chance of it happening.
- **Share** – A joint venture with a third party to share the benefit of the opportunity.
- **Accept** – Can be appropriate for positive or negative risks.

Identify Risks

SWOT Analysis - Strengths, weaknesses, opportunities, and threats.

- **Strengths** – What positive factors might impact our project? For example, a team has a history of completing similar projects.
- **Weaknesses** – Negative factors that might impact your project. E.g. being too reliant on one team member,
- **Opportunities** – Positive factors that might impact your project. For example, labor law changes are making hiring experts easier.

Threats – Negative factors that might impact your project. E.g. changes to legislation that add extra processes to your plan.

Managing Issues

1. Record the issue in the issue log and keep it updated.
2. Notify appropriate stakeholders.
3. Put together a resolution plan.
    a. **Prioritization** – Issue severity, impact on the project, impact on the organization, urgency.

b. **Root Cause Analysis** – Try to understand the issue using a cause-and-effect diagram (Ishikawa or Fishbone Diagram).
c. **Workaround** – The response to an unidentified risk.

4. Document outcomes and update other project documents.

Risk Management Plan (Document)

The how-to guide for how to manage risk in the project. Part of the project plan. Will include:

- Risk Strategy
- Definitions
- Risk Categories
- Roles/Responsibilities
- Escalation Paths
- Stakeholder Risk Appetite
- Reporting Formats
- Tracking

Scope Change Control Processes

Waterfall

Steps:	Explanation:
1. **C**reate or receive a change request	It can be submitted by anyone associated with the project. This should be done in writing and explain/justify the reasons for change.
2. **L**og the request	Using the Change Request Log.
3. **P**reliminary review	Discuss with the project team and SMEs to get an idea of whether the change is feasible.
4. **A**ssess the impact	Figure out what other areas of the project or product would be impacted by the change.
5. **R**ecommendation is documented	Recording your findings from step 4.
6. **D**ecision makers determined	Who will approve or deny the change?
7. **E**scalate to the Change Control Board (CCB)	The CCB is a panel of people who review changes and have the power to approve/deny the request.
8. **S**tatus is documented and communicated	Communicate the results of step 7.
9. **U**pdate the project plan	Updating the plan, such as the schedule and scope statement.
10. **I**mplement the change	Perform the change.

11. **V**alidate change implementation	Has it met the requirements of the original change request?
12. **C**ommunicate change deployment	Discuss the change at a staff meeting.

CR.L.P.A.R.D. – CRaig Loves Pampering And Riding Donkeys

E.S.U.I.V.C. – Esther Sometimes Understands It's Very Cold

Project Schedule

Estimate how much work activities will require using the WBS. Break down the work packages into activities and estimate the time for each activity.

Cost Estimates:

- **Analogous Estimating**: Using historical data from a similar project. It is useful when you don't have much detail. Quickest and easiest.
- **Parametric Estimating**: Performing simple calculations to determine the time needed.
- **Three-Point Estimating**: Formulas using 3 values to get a more accurate estimate. The most likely estimate, the pessimistic estimate, and the optimistic estimate.
- **Bottom-Up Estimating**: Every single component is estimated and then totaled. Most time-consuming but most accurate.

Effort – The amount of work required to complete the activity. E.g. 112 hours.

Duration – Considers the resources allocated to the task. E.g. 56 hours = 7 days.

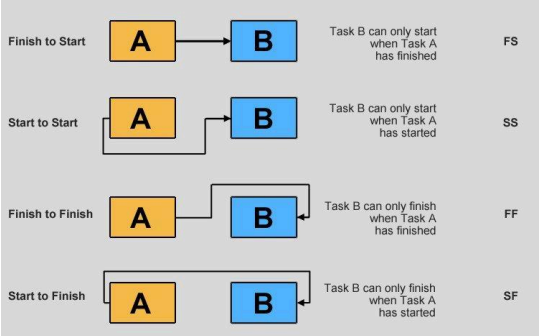
Elapsed Time – Considers the amount of work that can be done daily and weekly. E.g. 9 days.

Contingency Reserves – AKA **Buffers** – Adding extra time to our activity estimates in case things don't go as planned.

Dependencies – Influence the relationship between activities. 4 main types:

- **Mandatory** – Hard logic or hard dependency. Non-negotiable.
- **Discretionary** – Preferred logic. Order is negotiable.
- **External** – Something outside the project team's control.
- **Internal** – Something which is under the team's control.

Successor/Predecessor Activity Relationships



Developing the Schedule

A PERT Chart, an example of PDM, can be used to visualize activities and their relationships. Using a Gantt Chart to visualize the duration of tasks.

The Critical Path – The sequence of tasks that take the longest to complete. Any delays to the critical path will delay the project.

Assigning Resources

Resource Loading – Filling team members' available time with project tasks.

Resource Leveling – Adjusting the schedule according to resource availability. This will likely affect the critical path.

Resource Smoothing – Delays activities according to resource limits only within the available float.

Float – The time a task can be delayed without delaying the whole project. The critical path has zero float.

Schedule Compression – Techniques that shorten or accelerate the schedule duration without reducing scope. Examples:

- **Fast Tracking**: Overlap tasks to shorten the total duration.
- **Crashing**: Adding more resources to tasks on the critical path to reduce the total duration. This will add cost.

Resource Types

Physical Resources:

- **Capital Resource** – Physical resources used more than once to help us produce goods or services. E.g. Machinery, buildings, vehicles.

Human Resources:

- **Dedicated Resource** – Dedicated 100% of their time to the project.
- **Shared Resources** – Splits their time between the project and other work.
- **Overallocated Resources** – Too much work has been allocated to a resource.
- **Benched Resources** – Available resources that are not being used.
- **Internal Resources** – Already employed by the organization.
- **External Resources** – Acquired from outside the organization.
- **Remote Resource** – A resource located elsewhere. They could be internal or external.
- **Operational/Core** – On the project from start to finish.

- **Functional/Extended** – Will pop in and out for specialized project parts.

Budget

A combination of cost estimates and the WBS is used to get the total likely cost of the project, resulting in the cost baseline.

Cost Baseline – The approved version of the time-phased project budget excludes any management reserves, which can only be changed through formal change control procedures.

Contingency Reserves – When dealing with the budget, this refers to the money allocated towards known risks and included in the cost baseline—known unknowns.

Management Reserves – The amount of the project budget held outside the baseline reserved for unforeseen work within the project's scope—unknown unknowns.

Project Records Management

Records are like a document. A record is a piece of evidence that is final and will be kept for a specified period.

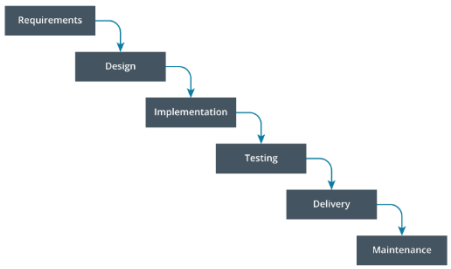
Contracts

- **Fixed-Price Contracts**: A total fixed price for the product or service is set. Appropriate for well-defined scopes where no significant changes are expected. There are subtypes:
  - **Firm Fixed Price (FPP)**: Price fixed and won't change unless the scope changes.
  - **Fixed Price Incentive Fee (FPIF)**: There are financial incentives if targets are met. E.g. Delivered ahead of schedule.
  - **Fixed Price with Economic Price Adjustments (FPEPA)**: Adjustments to cover exchange rates, interest rates, or cost changes of commodities.
- **Cost Plus Contracts**: The buyer pays the vendor for the actual costs plus a fee representing profit for the seller. Appropriate when the scope is expecting significant change. Subtypes:
  - **Cost Plus Fixed Fee (CPFF)**: all costs paid by the buyer, and then a **fixed fee paid** on completion.
  - **Cost Plus Incentive Fee (CPIF)**: Costs paid by the buyer, and then a **bonus if targets are met**.
  - **Cost Plus Award Fee (CPAF)**: All costs are reimbursed, and an extra fee is awarded based on the buyer's **judgment** of the seller's performance.
- **Time and Materials Contracts**: Contain aspects of fixed-price and cost-reimbursable contracts. Paying for time + fixed cost.

Frameworks/Methodologies

"The processes, tasks, roles, and guiding principles that form the structure of completing work."

- **WATERFALL** – AKA "**Predictive**". One of the oldest software development methodologies, predating software. Big Design Up Front (**BDUF**) approach. They are usually large teams, but they can be any size. A waterfall team assembles specialists to complete the work. The PM decides how many job types and skill sets are needed and fills them. There are 6 phases in a waterfall project:
    1. **Gather Requirements**: Define the project requirements. BDUF.
        a. **Project Requirements** - A statement that defines why a project is being undertaken, the functionality that a project is designed to accommodate, or how the functionality will be achieved and satisfied by the solution.
    2. **Design**: Convert the requirements into design specifications.
    3. **Implementation**: Complete the necessary work, such as writing code or constructing a building.
    4. **Testing**: Verify that your work meets the requirements and quality specifications, such as performing tests on your code.
    5. **Delivery**: Referred to as deployment in a software project. Moving the product into production.
    6. **Maintenance**: Maintain and solve issues.



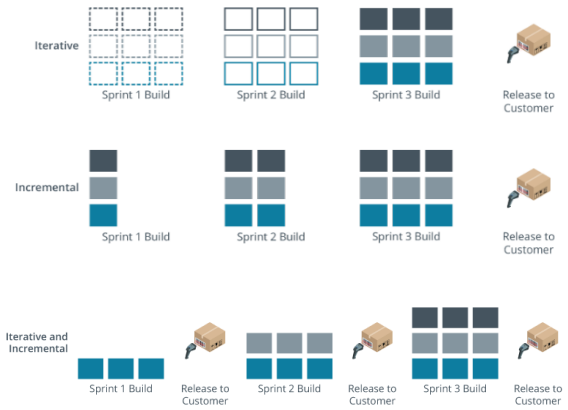
Pros: Simple and easy to learn. It can be tuned to any industry or project type. Includes thorough documentation.

Cons: Not flexible. Change management in waterfall is one of the more expensive elements.

Project Example: Designing and manufacturing a new vehicle.

Stakeholder Communication: The stakeholders provide comprehensive requirements at the start of the project. Then, they review the finished product at the end of the project. However, the team doesn't interact with the stakeholders during development. As a result, project risk increases because development could easily stray from the customer's needs, or those needs could change entirely.

- **AGILE** – AKA "**Adaptive**". A software development model focusing on iterative and incremental development to account for evolving requirements and expectations. Iterative and incremental design. Teams are a cross-functional group that stays with the project the entire life cycle, small by design, with just 3-10 members.



4 Key Values:

1. **Individuals and interactions** over processes and tools
2. **Working software** over comprehensive documentation
3. **Customer collaboration** over contract negotiations
4. **Responding to change** over following a plan - adaptive

12 Principles:

1. The highest priority is to satisfy the customer.
2. Welcome changing requirements.
3. Deliver working software frequently.
4. Businesspeople and developers must work together daily.
5. Build projects around motivated individuals.
6. Face-to-face conversations are the most efficient and effective communication method.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity – Essential to keep it simple and not worry about the "bells and whistles".
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective.

Pros: Great for uncertain environments with expected change. Typically produces higher quality results.

Cons: Changing requirements can sometimes lead to scope creep and missed deadlines. It can lead to less documentation.

Use Case: Projects where requirements will change. Stable, skilled teams capable of self-organization.

Project Example: Launching a new cloud-based personal budgeting application would use an agile approach. The team can quickly and affordably release small features to the customers. While the team has a good sense of valuable features, the customers will have ideas, requests, and issues that the team can incorporate into the project.

Waterfall vs Agile

Waterfall	Agile
Requirements can be analyzed in detail upfront.	Requirements are expected to evolve.
Changes are not welcome, and any required are subject to tight control.	Changes are embraced.
The working product isn't available until the end of the project.	A working iteration of the product is regularly available for feedback.

- **SCRUM** – An agile framework that focuses on iterative and incremental delivery of products; owes its popularity to a simple approach, high productivity, and scope for applicability to multiple areas. You must use all parts of SCRUM, or it isn't considered SCRUM. There are 5 SCRUM values:
    - Commitment
    - Focus
    - Openness
    - Respect
    - Courage

There are 3 SCRUM pillars:

1. Transparency (Sharing information improves decisions).
2. Inspection (Frequent reviews disclose problems and needed changes).
3. Adaptation (Incorporating results of inspection).

The SCRUM process has 4 steps:

1. The product owner prioritizes work into a product log
2. The Scrum team selects the top items in the backlog. The top items create a sprint backlog, which will deliver a working product that helps solve the top problem in the product backlog.
3. The Scrum team and stakeholders review the sprint results and adapt the product and the team's approach for the next sprint.
4. The steps repeat if the product backlog exists.

SCRUM framework includes 4 roles:

1. **Scrum Team** – The group of people who work together to deliver increments of values. Includes one Scrum Master, one product owner, and developers.
2. **Product Owner** – Owns the vision of the project. A Scrum team member who maximizes outcomes and selects and prioritizes the work that the agile team will complete. Creates, maintains, and owns the **product backlog**.
3. **Scrum Master** – The agile team's Scrum coach. A role in the Scrum framework that enables the team to apply the framework and offers coaching, training, and guidance. Promotes SCRUM values.
4. **Developers** – The remaining Scrum team members. It can be any job title, not just software developers. There must be fewer than 10 people in total—cross-functional and self-organizing.

Key SCRUM Artifacts:

1. **Product Backlog** - A prioritized list of customer requirements. It is the first step of Scrum.
2. **Sprint Goal** – The long-term goal of an agile product.
3. **Sprint Backlog** - A list of user stories selected from the product backlog that the Scrum team chooses and commits to complete in that sprint cycle.

4. **Increment** - A complete body of work that meets the definition of done and moves toward the product goal.

SCRUM Events:

1. **Sprint** – A Scrum iteration and the container for all other Scrum events. They are fixed lengths, usually one week to one month. Often referred to as the heartbeat of Scrum.
2. **Sprint Planning** - A session where the scrum team selects enough work to build a sprint backlog for the upcoming sprint. Lasts 8 hours for 1-month sprints and less for shorter sprints.
3. **Daily Scrum** - A meeting where the complete team gets together for a quick status update. These are short, 15-minute meetings conducted by standing in a circle. Lasts for 15 minutes every workday.
4. **Sprint Review** - An informal, end-of-sprint meeting of the agile team and product stakeholders to review the product's newest functionality in a working session, collaboratively adjust the backlog, and decide what to work on next.
5. **Sprint Retrospective** - A process improvement session where an agile team reflects on the previous sprint and identifies ways to improve how they work together. Lasts for 3 hours for one-month sprints and less for shorter sprints.

Pros: Appealing for both teams and customers. Flexible across any industry, and its short cycles mean less time between planning and the finished product.

Cons: Designed for a single, high-performing, cross-function team, less-experienced teams may struggle, and scaling to larger groups may be difficult. The daily meetings can be overwhelming for some.

Use Case: Larger projects, complex products with unpredictable workflows, and adaptable projects or products with expected unknown requirements.

Project Example: Launching a new cloud-based personal budgeting application. A simple version can be released to the market and improved through multiple increments. The team can iteratively develop and share portions of a feature with the stakeholders. The stakeholders' feedback will help the product owner and developers deliver the most valuable features.

Stakeholder Communication: Agile projects engage the stakeholders throughout the entire project. At the project's start, the stakeholders provide enough information to design the project. Throughout development, the team reviews progress with the stakeholders and gathers feedback that refines the requirements. If there's a problem, the team finds out early and can fix it at a much lower cost. Agile projects require stakeholders who have the time and motivation to collaborate. In exchange, stakeholders receive a product that meets their needs.

- KANBAN – A highly visual agile development methodology that emphasizes controlling work in progress and visualizing work. Used to control work-in-progress (WIP) levels and manage **workflows**. Kanban has 6 core practices:
    1. **Visualize Work** - Create a Kanban board that shows the specific phases that work moves through on your team. Then, the actual pieces of work that are moving through the stages will be created.
    2. **Limit WIP** - People who focus on too much work get overwhelmed. Context switching lowers productivity and degrades the work environment. Limiting WIP allows team members to focus and increases throughput.
    3. **Make Policies Explicit** - Clearly define how work gets completed and moves from one phase to another. Clear policies create a shared understanding of the work and the workflow.
    4. **Manage Flow** – Look for and eliminate bottlenecks in your workflow.

5. **Implement Feedback Loops** - Build feedback into the system to stay informed about the system's performance. Create and measure metrics and the flow of work across the Kanban board; these signals can help you identify bottlenecks and changes in your system.
6. **Improve Collaboratively, Evolve Experimentally** - Use the scientific method to test ideas and measure the results in the Kanban signals. Kanban's goal is to start now and keep improving. This practice reinforces the idea that whatever the organization is doing now is a good starting point because the team will keep improving the workflow.

Kanban Board – This can be a physical board with paper cards or completely virtual. The board visualizes all work, WIP limits, and work policies. Work moves through three basic phases: To Do, In Progress, and Done. However, teams can customize and adapt the phases or add phases, as in the example below, to match their workflow.

Kanban 4-part Workflow:

1. The product owner prioritizes and orders the backlog.
2. A team member selects the top-ordered item in the backlog and pulls it into progress.
3. The team moves the work item through the workflow, keeping within the WIP limits.
4. When the team is ready to start new work, they select the next item in the backlog.

Pros: It is the simplest agile framework to learn, and it can be applied to any industry.

Cons: Works best in a relatively stable work environment and can suffer if teams are oversized or have high turnover. Inaccuracies or changes between work items can disrupt the team's focus, and cycle times may stretch out without set iterations if performance is not properly managed.

Use Case: Where the backlog or worklist changes often, teams that work well with less structure, or in relatively stable workflows.

Project Example - A project to create and staff a new development team would use a Kanban approach. Team creation and recruiting are relatively predictable workflows. However, the time for each phase can vary; for instance, recruiting will take more or less time, depending on the number and quality of applicants. In addition, the team would like to keep the entire project visible, and they need to know if something is taking too long. Kanban will allow the team to visualize the work and move it through the system as fast as the specific task allows.

- LEAN – Focuses on removing anything that doesn't add value. Overlaps often with agile.

- **EXTREME PROGRAMMING (XP)** - An agile software development framework noted for its heavy emphasis on software engineering practices. XP gives special duties to three roles:

1. **Customer -** Decides which features are needed eventually and which are needed next. When the "customer" is thousands of users, teams select one or more customer proxies to represent the voice of the customer.
2. **Tracker** – Captures metrics, measures progress, and seeks improvement opportunities—usually a part-time role.
3. **Coach** – Mentors team members on how to use XP practices. It's usually somebody outside the team who has experience in XP.

Common XP Practices:

1. **Pair Programming -** Two software engineers work side by side to create code simultaneously. Each line of code has two people working on it; typically, one person writes code while the second person watches and provides feedback.
2. **Ten-Minute Build** – It should take ten minutes or less to build, test, and deploy the entire system. If it takes longer, developers are less likely to run the tests. It also drives other good practices, such as writing clean tests and automating testing and integrations.
3. **Continuous Integration** – Merge local code onto the main repository regularly, at least a few times daily. Frequent integration reduces conflicts and allows automatic builds and tests to run.
4. **Test-First Programming** – Write tests first, then create code to pass the tests.
5. **Incremental Design** – Small, frequent improvements are more valuable than delayed releases.
6. **Sit Together** – Teams often work in a shared, open space.
7. **Whole Team** – Everyone needed for the project is on the team.
8. **On-Site Customer** – Customers are also part of the team on-site.
9. **Energized Work** – XP believes people are at their best when they aren't exhausted. Employees should work no more than 40 hours. If somebody has to work overtime one week, they cannot do so the following week.
10. **Stories** – Also called "user stories," this is an effective format for communicating requirements. The user's needs are written in a short, descriptive format that conveys what the user wants and how it would help them.
11. **Weekly Cycle** – The heartbeat or pulse of the team, similar to a sprint cadence. Each week, the team selects stories and plans work for the week.
12. **Quarterly Cycle** – A planning and reflection event. The team inspects and adapts its processes, similar to Scrum's retrospective. They also identify the major themes for the project and create stories to deliver on them. In this context, themes serve the same purpose as Scrum's product goal.
13. **Slack** – A team's weekly plan should not be filled to capacity with nonnegotiable, high-priority work. It creates too much pressure and takes away flexibility. Instead, build slack into the team's weekly plan by including optional tasks that the team can reschedule if they don't have time to complete.

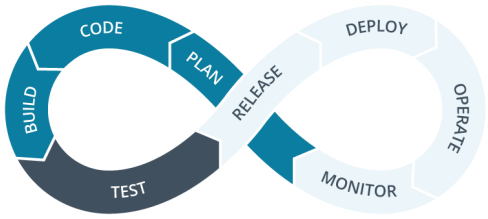
Pros: XP is highly efficient, builds the most essential increments first, and does not waste time on low-priority features. Continuous integration, automation, and a high adaptability rate are built into the process out of necessity. Teams using XP come together quickly due to the team-focused practices and experience less project failure because the customer is a team member.

Cons: XP focuses on developing good software; the framework has limited use outside that scope. Code-centric practices around software development can result in a lackluster design, which may drive away customers. The rapid pace of the work can cause high levels of stress on the developers, and the minimal documentation can hinder new team members' understanding. Additionally, the XP framework relies on face-to-face interaction between team members and customers.

Use Case: Small, collocated teams who must manage changing and unknown requirements, projects with tight deadlines, and teams with test automation capabilities.

Project Example: A project to launch a new cloud-based personal budgeting application could be a good use case for XP. This use case assumes the team is collocated and can include the customer. A cloud-based platform accommodates frequent updates. The project will create a software product to solve current users' problems.

- **DevOps** - A combination of software development and systems operations refers to integrating one discipline with the other. CI/CD is a core DevOps practice.



The constant flow of work within a CI/CD methodology

Continuous Delivery – The process after continuous integration that prepares the code for production release. Continuous delivery automates all processes except release to deployment. A company that implements continuous delivery can deploy at any time but chooses to delay deployment strategically.

Continuous Deployment – The fully automated alternative to continuous delivery. After integration, the continuous deployment process prepares the code for production release and deploys it to production.

Pros: Superior operational support, reducing team friction, and improving cross-functional collaboration. This framework's emphasis on automation increases the frequency of builds, tests, and deployments, allowing teams to deliver good code faster.

Cons: DevOps teams require significant expertise on each team, and groups may struggle to find enough team members with the necessary skills. The DevOps process is so different from other frameworks that traditional specialist teams may need more time to adapt. Additionally, large or legacy systems with minimal automation will find it difficult to switch to DevOps quickly.

Use Case: Teams that want to implement CI/CD and release more often, companies working to address friction, and companies with the resources.

Project Example: DevOps is not a project management methodology, but it does control the software engineering process. So, as a PM, you are unlikely to adopt DevOps for a single project, but you will need to know how to work within a DevOps team.

- **DevSecOps** - A combination of software development, security operations, and systems operations; refers to integrating each discipline with the others. DevSecOps creates a shared responsibility where everyone considers and owns the product's security.

- **Scaled Agile Framework (SAFe)** – A popular agile-at-scale framework incorporating multiple agile practices and frameworks. Not to be confused with Scaled Agile.

Scaled Agile – Generic term describes any methodology that expands agile tools to multiple teams. Since agile teams are small, you can have multiple groups of agile teams (10 groups of 6, for example) to achieve a scaled agile team.

Key SAFe Terms:

1. **Agile Team** – Teams of 3-10 cross-functional people who work together to deliver value increments.
2. **Agile Release Train (ART)** – Groups together related agile teams to simplify communication, planning, and schedules.
3. **Iteration** – Agile teams still iteratively develop in fixed-time increments. Scrum teams refer to this as a Sprint, and XP teams refer to this as a Weekly cycle.
4. **Timebox** - An agreed-upon, fixed time allocated to a specific topic or project activity.
5. **Program Increment (PI)** – Similar to XP's Quarterly Cycle, a Program Increment is a longer-term planning cycle.
6. **Program Increment Planning (PI Planning)** – The ART's planning event is similar to sprint planning for a single Scrum team.

Pros: Useful for bringing agile benefits to a larger group and scaling to any needed size. Ideal for building connections between business- and technology-focused teams to align with the larger business goals.

Cons: Adds multiple new terms, overhead layers, and staff, which can conflict with Agile's "ground-up" approach. SAFe's structure can also limit the team's freedom.

Use Case: Companies with projects spread across many teams and companies with many small teams must communicate and coordinate.

Project Example: SAFe is an organizational undertaking, so you wouldn't adopt SAFe for a single project. However, the organization's structure and framework will affect your role as a PM.

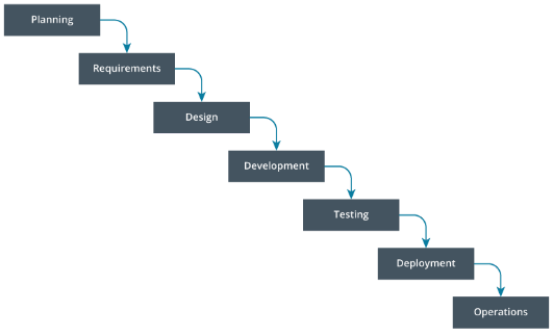
- **Software Development Life Cycle Methodology (SDLCM)** – A flexible software development framework that aims to produce high-quality, low-cost, and thoroughly tested software. This structure improves the software development process through continual improvement. Often have 5-8 phases, common phases include:
    1. **Planning** – Outline the project and define the software's scope and purpose.
    2. **Requirements** – Define the resources needed for the project and what the software should do.
    3. **Design and Prototyping** – Define how the software needs to work, which security protocols to use, and how the user interface behaves.
    4. **Development** – Create software.
    5. **Testing** – Run tests to ensure the software performs as expected without any bugs.
    6. **Analyze** – Assess risks, issues, and changes in the software development lifecycle.
    7. **Operations and Maintenance** – Support the software by enhancing features and fixing bugs.

Pros: A comprehensive framework that strives to include every necessary task and the required documentation. This highly detailed methodology reduces risk, cost, and time.

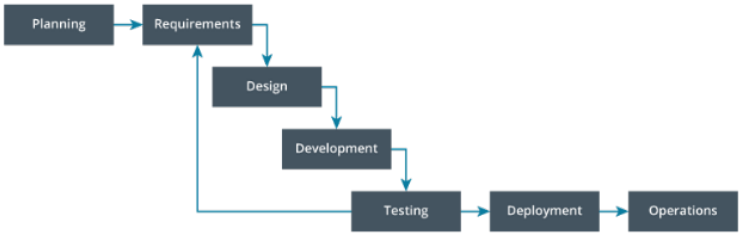
Cons: It depends on time spent on thorough advance planning, which might not be possible for some teams. Agile-focused team members may find the highly structured processes overly constraining.

- **SDLC Models**:

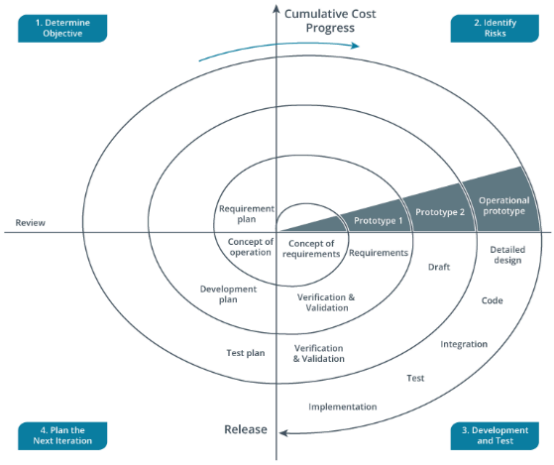
- **Waterfall SDLC Model**: The original SDLC model which replicates waterfall methodologies. Waterfall SDLC is suitable for predetermined technology stacks and short projects. It does not work well for long-term, complex projects.



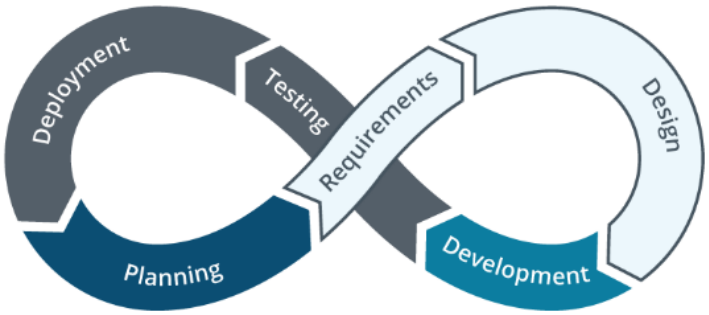
- **Iterative SDLC Model**: In this model, the beginning and ending phases occur only once, while the middle stages iterate as often as needed.



- **Spiral SDLC Model**: A risk-centric iterative model that is the most complex and resource-intensive model. The project cycles through all SDLC phases, and each cycle assesses risk. This model works well for large projects with urgent, unknown risks. It does not work well for small projects with limited resources.



- **Agile SDLC Model**: An iterative, customer-centric model. Agile SDLC repeats all phases through every iteration like Spiral SDLC, but the customer views the results after every iteration. This model works well for long-term projects where customer feedback is needed. Companies seeking a fast time-to-market also use this model. Agile SDLC does not work well for projects with critical budget, time, or scope limitations.



- **Projects in Controlled Environments (PRINCE2):** A process-based project management methodology that aims to control the project management process by predefining clear project phases, roles, and tasks. A PRINCE2 framework says a project should have an organized, controlled start, middle, and end. Flexible to be used with waterfall or agile. A process model for project organization and control.

PRINCE2 Roles:

1. **Team Manager**: They help the PM by supervising the teams and managing the quality of outputs.
2. **Project Board**: Those who are accountable for a project. The board authorizes resources and funding, supports the PM, and is ultimately responsible for the project. The project board includes three roles: executive, senior user, and senior supplier.
    a. **Executive**: A member of upper management and represents a business perspective. The executive is also the deciding member of the project board, and they own the project's business case. Each project board has one executive.
    b. **Senior User**: Represents the customer's perspective. A project board will have one or more senior users.
    c. **Senior Supplier**: Represents the supplier or implementation partner's perspective. One or more senior suppliers sit on the project board.

PRINCE2 7 Principles:

The seven principles represent the philosophy and best practices of PRINCE2. When you approach a PRINCE2 project, use these principles to guide your decision-making, and you will likely stay on track with the PRINCE2 framework.

1. **Continued Business Justification**: Where other methodologies create a business case to start the project, PRINCE2 projects update the business case throughout the project's life.
2. **Learn From Experience**: The team maintains a lessons log and reports on lessons learned throughout a project. They can also view the lesson reports from other projects.
3. **Defined Roles and Responsibilities**: PRINCE2 creates clarity for team members. This process leads to faster decision-making and reduced friction because the team understands who does what.
4. **Manage by Stages**: This process breaks projects into two or more phases, and the boundaries between phases serve as checkpoints for the project board. While PMs handle the day-to-day management duties, the project board reviews a project when it is ready to move from one stage to the next.
5. **Manage by Exception**: This phase sets performance tolerances for the standard project performance metrics: time, cost, quality, scope, benefits, and risk. A tolerance defines the acceptable range of performance. For example, a goal is 10 days, but 10 days +/-1 day is a tolerance.
6. **Focus on Products**: The team understands and agrees on the product's definition and quality requirements.

7. **Tailor to Suit the Product**: PMs adapt PRINCE2 elements to fit a specific project's size, scope, and requirements. Organizations adapt the PRINCE2 framework to meet scaling requirements, industry standards, and company needs.

PRINCE2 7 Themes:

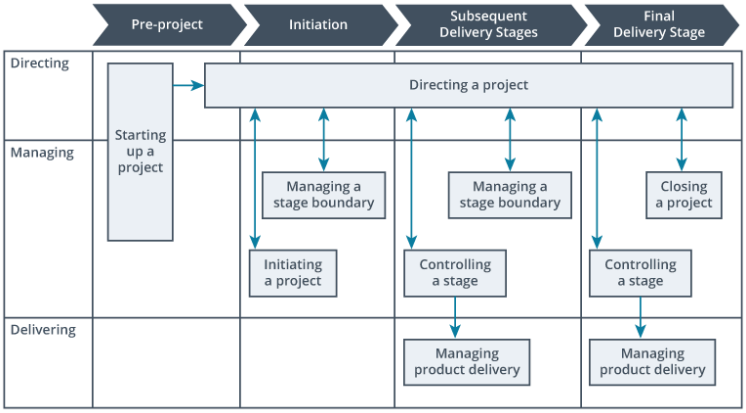
The seven themes represent the PM's knowledge areas. If the principles are ideals, the themes are what you do with them.

1. **Business Case**: Use the business case to provide continued business justification.
2. **Organization**: Define and document the roles and responsibilities.
3. **Quality**: Define product quality to help the team focus on the product.
4. **Plans**: Maintain a project plan and measure project performance.
5. **Risk**: Document and manage risks throughout the project.
6. **Change**: Track changes and obtain approval before adding them to the plan.
7. **Progress**: Regularly track and measure the project's progress so you can keep the project on track and address exceptions immediately.

PRINCE2 7 Processes:

The seven processes represent the activities and their purpose throughout a project's life cycle.

1. **Starting up**: Initiate the business and determine a project's viability.
2. **Initiating**: Define the project's aspects: scope, costs, time scales, risk, quality, and benefits.
3. **Directing**: The product board provides oversight, including approving the project at stage boundaries. The project board decides when a project is ready to close.
4. **Controlling a Stage**: This stage includes the PM's day-to-day activities. The PM breaks the project into tasks and oversees progress. They respond to and escalate issues when they arise
5. **Managing Delivery**: The PM measures the project's performance. They also verify quality and obtain approvals of completed work.
6. **Managing a Stage Boundary**: The PM prepares project updates and updates the project plan. The project board reviews the project data and decides whether it should advance to the next stage. The project team captures lessons learned.
7. **Closing**: The PM completes closing materials, turns over completed work, and closes the project.



Pros: Adaptable for any sized project, easy to learn the processes. The methodology is built on continuous improvement principles and project results reflect constant growth. Finally, PRINCE2 requires high levels of communication, so stakeholders are rarely surprised by the project's progress.

Cons: The extensive documentation required in the model can cause some teams to become bogged down. The structure also requires organizational buy-in to support certain elements, such as the project control board.

Use Case: Good for process-focused organizations that rely on documentation or executives looking to keep projects aligned and maintain an understanding of projects.

Project Example: PRINCE2 is an organizational undertaking, so you wouldn't adopt it for a single project. However, a company's industry, structure, and culture can indicate if PRINCE2 should be adopted.
