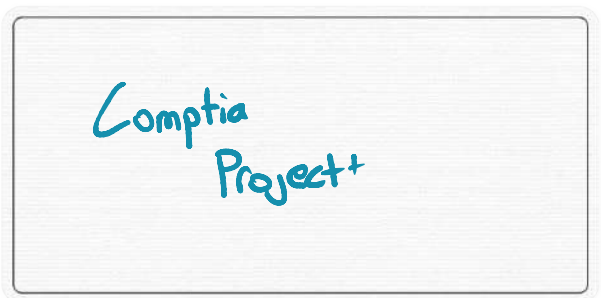
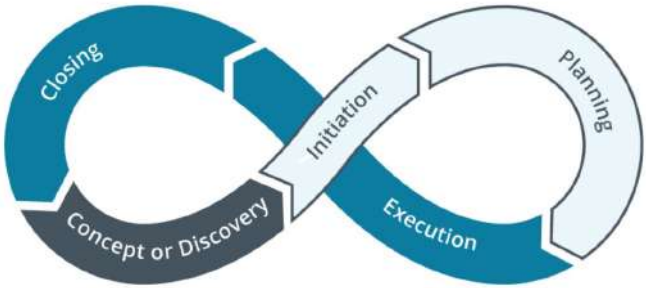

Comptia
Project+

Preparing for the Project

project life cycle a process that defines the five phases that a project goes through from beginning to end



Business Case Template

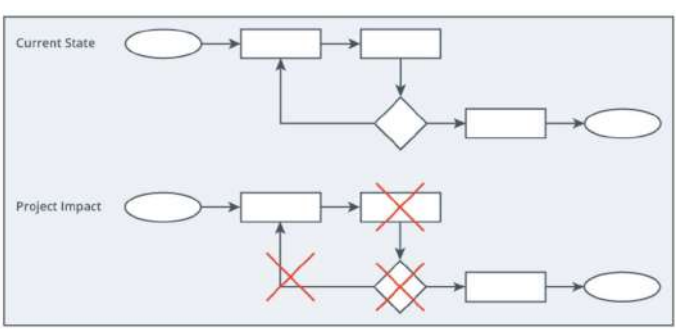
1. Executive Summary
   - Problem
   - Solution
   - Expected results
2. Problem Statement
   - Current situation
   - Business problem
3. Problem Analysis
   - Data
   - Supporting arguments
4. Options
   - Alternatives A, B, C
   - Impact of each alternative
5. Project Definition
   - Project scope
   - Timeline
   - Milestones
   - Needed resources
6. Financial Overview
   - Cost benefit analysis
   - Return on investment (ROI)
   - Risks
7. Recommendation
   - Recommended option from Options section
   - Next steps

business case a brief document that justifies the investments made for a project and describes how a particular investment is in accordance with the organization's policy

Current state describes what is happening now

Future state describes what the business will look like after implementing the project

process flowchart shows the sequence of events and the flow of inputs between elements in a process or system



Return on Investment (ROI)

ROI is often presented in the following three formulas (all will generate the same result).

$$\text{ROI} = \frac{\text{Net Profit}}{\text{Cost}} \times 100$$

$$\text{ROI} = \frac{\text{Revenue - Cost}}{\text{Cost}} \times 100$$

$$\text{ROI} = \frac{\text{Financial Value - Project Cost}}{\text{Project Cost}} \times 100$$

benefit measurement models a project selection decision model that analyzes the predicted value of the completed projects in different ways

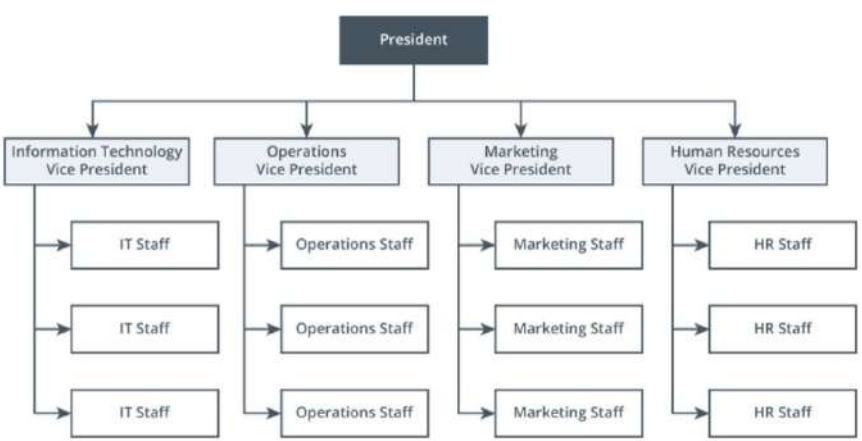
ESG factors environmental, social, and governance factors

Operational work is routine, predictable, and repetitive

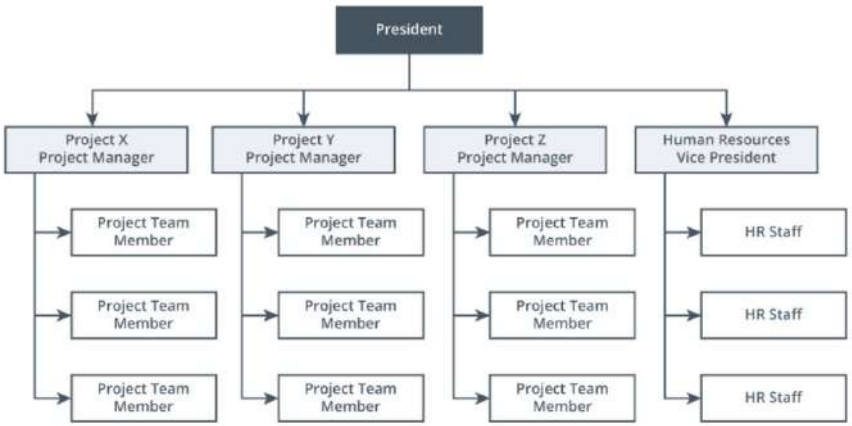
Project work accomplishes something new

project objectives criteria used to measure whether a project is successful or not

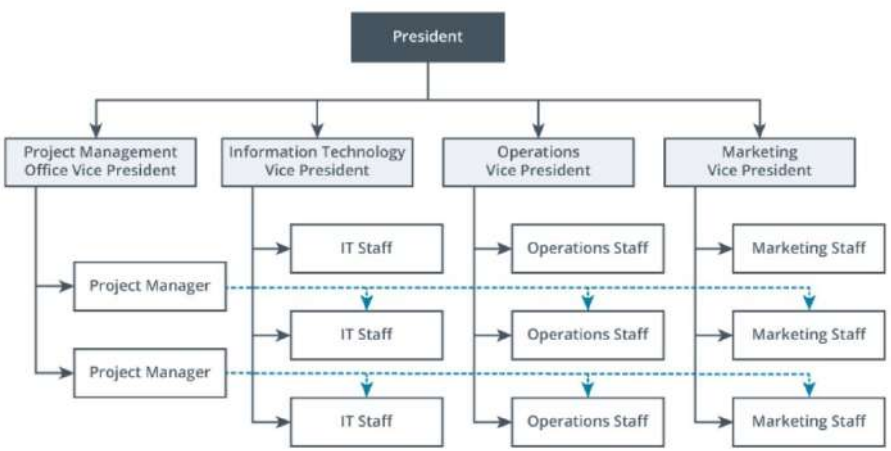
functional organizational structure



projectized organizational structure



matrix organizational structure



Matrix organizations can be further divided into 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.

If functional and project organizations represent the extreme ends of a spectrum, matrix organizations represent everything in between.

program a group of related projects that have a common objective

Project Management Office (PMO) is the functional department for all PMs in a company

The PMO's strength and authority will vary depending on its format. There are three PMO types.

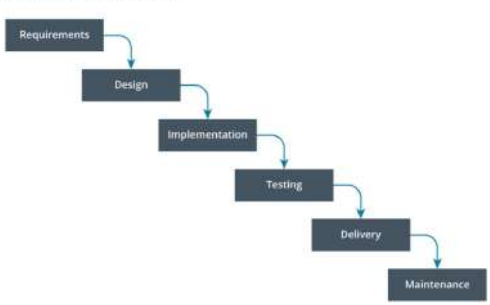
- A **supportive PMO** provides support when it's requested. In this model, the PMO is essentially a library of project information. It doesn't force adherence to standards and may not assign or evaluate projects. Organizations that favor decentralization may opt for a supportive PMO. In this model, business areas hire PMs and manage projects. They only request administrative support and occasional assistance from the PMO. PMs do not report to the head of PMO; instead, they report to somebody in a business area.
- A **controlling PMO** is 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.
- A **directive PMO** is the most controlling type of the three covered here, and it 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. PMs report to the head of the PMO. An organization that cannot afford missing documentation and half-finished procedures will opt for a directive PMO. For example, you will see this format in highly regulated industries.

portfolio a collection of projects, programs, and operational work to achieve the strategic business objectives of an organization

Selecting the Project Framework

Waterfall

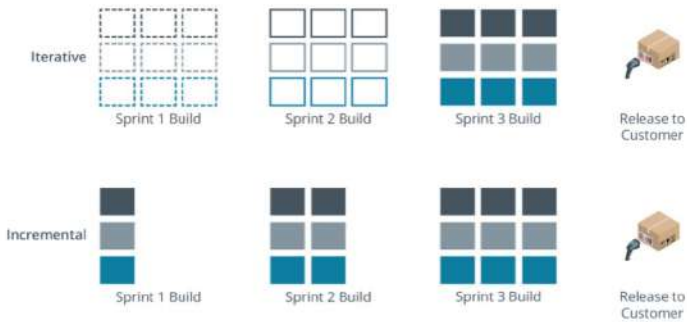
Many frameworks and methodologies are circular when graphed, but waterfall phases uniquely cascade from the first stage to the last.



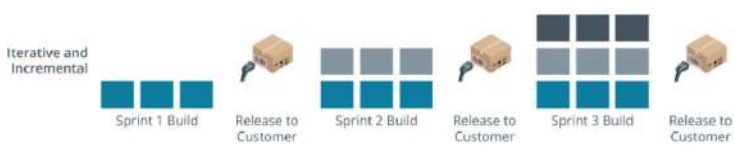
Agile

- **iterative development** develop small parts of features and improve them often

- **incremental development** build all of one of the features and release it



- **iterative and incremental development (IID)** build one feature at a time



Enterprise Resource Planning (ERP) software that enables an organization to manage services, personnel, and IT resources

Scrum is a lightweight, customer-centric framework

The Scrum process is easily described in four steps:

1. The product owner prioritizes work into a product backlog.
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 as long as the product backlog exists.

The framework also introduces four new roles:

- **Scrum team:** the group of people who work together to deliver increments of value. It includes one Scrum master, one product owner, and developers. The team holds all the capabilities to deliver a product increment each sprint. However, it is small enough to form cohesive relationships. Therefore, the cross-functional teams usually have more than three but fewer than ten members.
- **Product owner:** each team needs one product owner, but they can serve on multiple agile teams. The product owner creates, maintains, and owns the product backlog. It is up to the product owner to maximize value by optimizing the product backlog.
- Scrum master: the agile team's Scrum coach. They help the team, product owner, and organization improve how they apply Scrum.
- Developers: the remaining Scrum team members (not the product owner or Scrum master). They are not limited to software developers. Instead, view "developer" as a shorthand term for a Scrum team member. They can be any job title or function that helps the Scrum team produce value. For example, developers own the sprint backlog and work together to complete the backlog items.

product backlog a prioritized list of customer requirements

product goal a long-term goal of an agile product

Sprint a Scrum iteration and the container for all other Scrum events

- Sprint backlog is the Scrum team's plan for the upcoming sprint
- Sprint goal the primary value that a team plans to deliver in a given sprint

An increment is a complete body of work that meets the definition of done and moves toward the product goal

Cadence the predictable repetition of events

1. **Sprint Planning:** the product owner discusses the highest-priority items in the product backlog and how they relate to the product goal. The team creates a sprint backlog by breaking down PBIs into smaller pieces of work. The team is wholly responsible for deciding how to deliver work. Sprint planning lasts eight hours for one-month sprints and less for shorter sprints.
2. **Daily Scrum:** developers meet to review progress toward the sprint goal and plan the next 24 hours of work. Developers should self-organize and use this time to create plans, review issues, and make decisions. The daily Scrum lasts for 15 minutes every working day.

3. Sprint review: the agile team shares their progress with stakeholders. Demonstrating small pieces of work reveals valuable information. It uncovers issues early when they are small and easy to fix, reveals previously unknown development ideas, or confirms that the team is on track. Typically, the group shares their progress, and the stakeholders provide feedback and ask questions. All attendees discuss where to go next, and the product owner may adjust the product backlog based on the sprint review. A sprint review lasts four hours for one-month sprints and less for shorter sprints.

4. **Sprint Retrospective:** the team inspects how they worked in the past sprint and identifies improvement opportunities. The retrospective is the only event that isn't concentrated on the product. Instead, the retrospective focuses on the Scrum team and how they can improve how they work together. The event lasts three hours for one-month sprints and less for shorter sprints.

Day 1	Day 2	Day 3	Day 4	Day 5
9:00–9:15 – Daily Scrum 12:00–4:00 – Sprint Planning	9:00–9:15 – Daily Scrum	9:00–9:15 – Daily Scrum	9:00–9:15 – Daily Scrum	9:00–9:15 – Daily Scrum

Day 6	Day 7	Day 8	Day 9	Day 10
9:00–9:15 – Daily Scrum	9:00–9:15 – Daily Scrum	9:00–9:15 – Daily Scrum	9:00–9:15 – Daily Scrum	9:00–9:15 – Daily Scrum 10:00–12:00 – Sprint Review 2:00–4:00 – Sprint Retrospective

Kanban a highly visual agile development methodology that emphasizes controlling work in progress and visualizing work

Kanban's six core practices drive the rest of the methodology.

1. **Visualize work:** create a Kanban board that shows the specific phases that work moves through on your team. Then create the actual pieces of work that are moving through the stages.

2. **Limit WIP:** when people focus on too much work, they 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. This practice embraces Kanban's roots in lean and reduces waste in the system.

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's most recognizable feature is the Kanban board. It 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 teams follow a four-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.

Extreme programming (XP) an agile software development framework noted for its heavy emphasis on software engineering practices

Roles

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 looks for improvement opportunities. It is usually a part-time role for a team member.
3. **Coach:** mentors team members on how to use XP practices. It's usually somebody outside the team who has experience in XP.

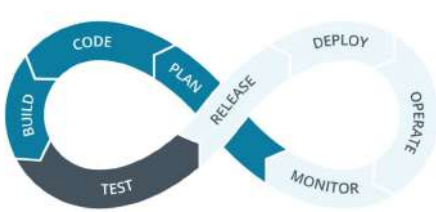
Below are commonly used XP practices; XP has an extensive library of practices extending beyond this course's scope. Because many agile concepts are broadly applicable, you will often see similarities across frameworks.

- **Pair programming:** two software engineers work side by side to create code at once. Each line of code has two people working on it; typically, one person writes code, while the second person watches and provides feedback. Pair programming is a perfect example of taking a good development practice (code review) to an extreme level (reviewing code continuously while it is written).
- **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.
- **Continuous integration:** merge local code onto the main repository regularly, at least a few times a day. Frequent integration reduces conflicts and allows automatic builds and tests to run.
- **Test-first programming:** traditional software development writes and passes tests after development, but XP takes the opposite approach. Also called "Test Driven Development" (TDD), test-first programming writes tests and then writes the code to pass the tests. This practice drives XP's code simplicity. Using TDD, developers write just enough code to pass the test, then move on to the next work item.
- **Incremental design:** XP adopts incremental design, like all other agile methodologies. Small, frequent improvements are more valuable than delayed releases.
- **Sit together:** teams often work in a shared, open space. Imagine a large room with no walls, cubicles, or dividers. Removing physical barriers removes communication barriers.
- **Whole team:** like other cross-functional teams, everybody needed for a project is on the team. For longer projects, the team members could change throughout the project's life cycle.
- **On-Site Customer:** "whole team" really does mean everybody; XP includes the customer on the team. The customer is even collocated with the team whenever feasible.

DevOps a combination of software development and system operations

CI/CD a software development method combining app and platform updates with code updates

Continuous integration a software development method in which code updates are tested and rapidly committed to a development or build server/code repository



Continuous delivery app and platform requirements are frequently tested and validated for immediate availability

Continuous deployment app and platform updates are rapidly committed to production

DevSecOps a combination of software development, security operations, and system operations

SAFe a popular agile-at-scale framwork incorporating multiple agile practices and frameworks

Key SAFe Terms

SAFe's framework introduces many new terms, but the concept of many terms is rooted in the agile frameworks you already know.

- Agile team: teams of 3–10 cross-functional people work together to deliver increments of value. SAFe genericizes the term to "agile team," regardless of the team's chosen framework. So Scrum and Kanban teams still practice their preferred planning and delivery methods. However, SAFe documentation refers to them all as agile teams.

- **Agile release train (ART):** in large companies, multiple agile teams work together to create large products. For example, one agile team of eight people probably wouldn't work alone to create a new operating system. Many teams contribute to the final product, and their shared interest means they need to communicate and collaborate. An ART groups the related teams to simplify communication, planning, and schedules. Teams on an ART plan and work together to support a value stream.

- **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. SAFe genericized the term to "iteration" for all teams. The purpose of the timebox doesn't change; teams still select enough work to deliver an increment of value in the set timebox. However, the timebox may be more structured. SAFe recommends 1–4 week iterations, just like Scrum (and compliant with XP). However, the iteration schedule needs to align with the Program Increment schedule.

- **Program Increment (PI):** similar to XP's Quarterly Cycle, a Program Increment is a longer-term planning cycle. The PI differs from the Quarterly Cycle in a couple of notable ways. A PI is a fixed-length increment that lasts 8–12 weeks instead of three months. A PI also involves more than one team. The Program Increment is a shared timebox for all the teams on an ART.

- **Program Increment Planning (PI planning):** the ART's planning event, similar to sprint planning for a single Scrum team. The agile teams on an ART come together to plan what they will deliver in the upcoming PI. The agile teams synchronize their goals, plan for any dependencies, and address risks in this event.

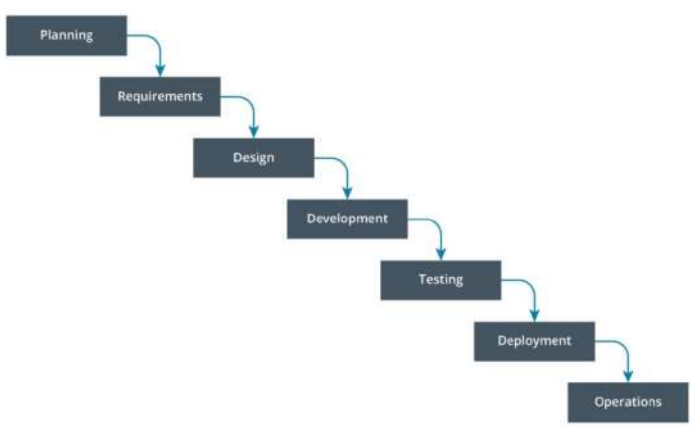
Software development life cycle methodology (SDLM) aims to produce high-quality, low-cost, and thoroughly tested software

Software development life cycle (SDLC) the processes of planning, analysis, design, implementation, and maintenance that often govern software and systems development

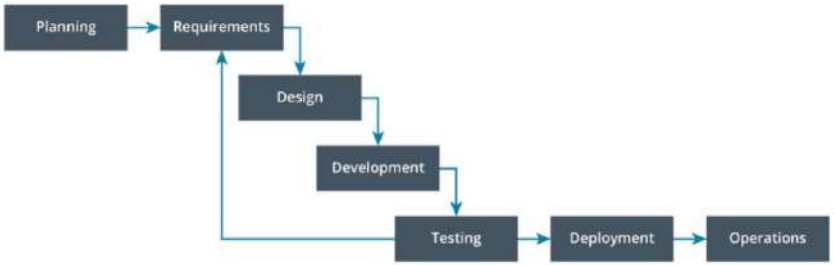
Common SDLC Phases

1. **Planning:** outline the project and define the software's scope and purpose.
2. **Requirements:** define the resources needed for the project. Determine what the software needs to do.
3. **Design and prototyping:** define how the software needs to work, such as which programming language to use, which security protocols to use, and how the user interface behaves. Create a low-function prototype to elaborate on the design. Initiate a risk management plan.
4. **Development:** create the software.
5. **Testing.** run tests to ensure the software performs as expected.
6. **Analyze:** assess risks, issues, and changes of the software development lifecycle.
7. **Operations and Maintenance:** support the software as long as it is in production. Launch new development cycles to fix bugs and create new enhancements for the product.

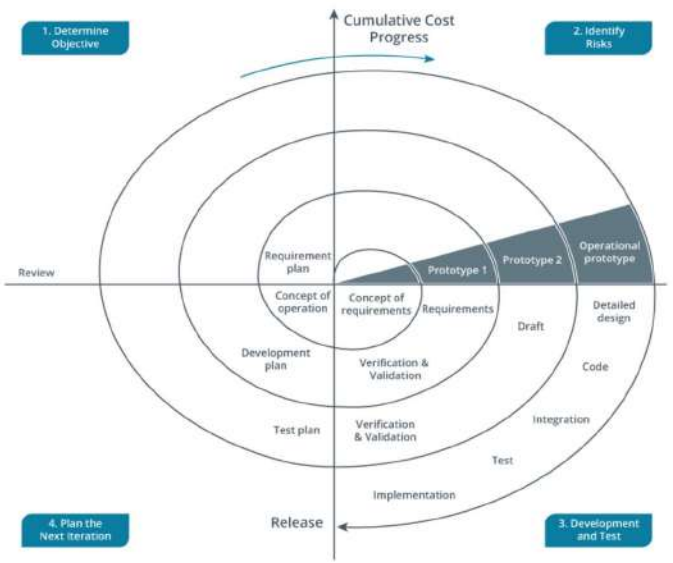
•Waterfall SDLC Model



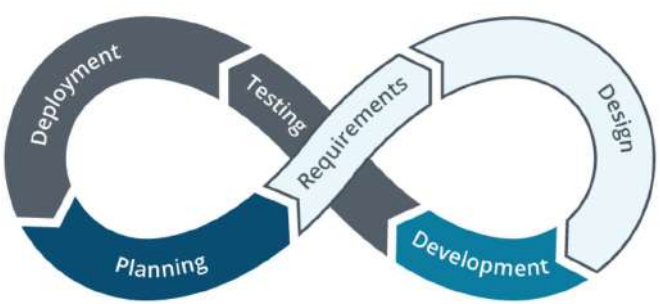
•Iterative SDLC Model



•Spiral SDLC Model



•Agile SDLC Model



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

tailoring the act of determining which processes are appropriate for any given project

Roles and Responsibilities
PRINCE2 introduces several new roles.

- **Team manager:** on large project teams, a team manager joins the project management team. They help the PM by supervising the teams and managing the quality of outputs.
- **Project board:** 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.
- **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.
- **Senior user:** represents the customer's perspective. A project board will have one or more senior users.
- **Senior supplier:** represents the supplier or implementation partner's perspective. One or more senior suppliers sit on the project board.

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.

One of the methodology's strengths is the customization; you can use this framework for projects of any size by adding or eliminating certain elements. But these principles are the only part of PRINCE2 that cannot be customized; you must apply all seven principles, or the project is not PRINCE2.

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. A project is continually evaluated to ensure that it is still a good decision to pursue it.
2. **Learn from experience:** this lessons learned or continual improvement process helps teams learn from their own experience and other projects. 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 has a shared understanding of 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. For example, the project board reviews the project's progress and plans to decide if the project should continue. If the board approves, the PM receives funding and authorization to complete the next phase. If the board doesn't approve, the PM closes the project.

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. Tolerances clarify the PM's authority by providing flexibility to address minor performance issues instead of escalating every overage. The PM handles decisions as long as the project stays within the predefined tolerances. An exception occurs when the project exceeds a tolerance, and the PM escalates exceptions to the project board.

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.

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.

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 if it should advance to the next stage or not. The project team captures lessons learned.

7. **Closing:** the PM completes closing materials, turns over completed work, and closes the project.

Roles

- **Product Owner:** Agile frameworks introduced the product owner role. Product owners 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.
- **Project Manager:** PMs lead projects in any framework. They form a team for a short duration to achieve a specific objective. The PM is responsible for a project only as long as it is active.
- **Product Manager:** works in any framework. Product managers serve a more strategic function than product owners or PMs. Product managers oversee products and therefore exist for a product's entire life cycle. They define the product strategy and create roadmaps to show how it will improve and change over time. The PM identifies projects and product changes. They often generate ideas that lead to new work for both PMs and product owners.

Responsibilities

Relatively speaking, the PM's position in a project is more short-lived than a product owner's (though we know projects can last for years). PMs and product owners have shared interests with both unique and overlapping responsibilities.

The product owner's responsibilities include the following:

- 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.

The PM's responsibilities include the following:

- Managing project plans, including the schedule, initiation, and closing documents.
- Measuring the project's progress.
- Managing project risks.
- Working within a fixed time (project schedule), fixed cost (project budget), and fixed scope (requirements).

Both PMs and product owners are responsible for the following:

- Communicating directly with stakeholders.
- Leading their team (The product owner shares guidance about the product's purpose and vision. The PM builds support for the project and project approach.).
- Maintaining work lists. (The product owner maintains a product backlog, and the PM uses a work breakdown structure (WBS) or project backlog. However, the product backlog represents the product owner's wish list, and the WBS represents the customer's requirements.).

Requirements discrete descriptions of how a project or product needs to look, behave, or operate

requirements management plan a document that describes how project requirements will be managed

progressive elaboration additional layers of detail are defined over the course of a project

Initiating the Project

Project Manager

The PM owns day-to-day project management responsibilities. The project sponsor holds ultimate accountability for the project, and they essentially delegate project operations to the PM. Therefore, the PM is accountable to the project sponsor.

Their responsibilities include the following:

- Build the project team and secure necessary resources.
- Build the project charter and define the project scope.
- Establish the required project logs and processes, including the issue log, change log, and risk register. These documents record events that may affect the project's scope, budget, and timeline. The PM sets the process for recording and responding to each event.
- Ensure the project deliverables are met. Their work includes working with the team to define the tasks, verifying that the team members have the necessary skills and resources, and tracking activity completion. They also manage the activity list and follow up on issues.
- Prepare and deliver project status updates.
- Track the project's progress and status, including performance to schedule and performance to budget.
- Manage vendor performance and relationships.

Business Analyst

The business analyst (BA) improves the final product. BAs hold a unique role between business and technical teams. They deeply understand business direction and company priorities and are familiar with the software environment. As a result, they operate as a translator between business and IT.

Their responsibilities include the following:

- 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

Architect is an umbrella term for many designer roles in an organization. For example, enterprise architects, software architects, and solution architects support IT projects and operations. Architects design solutions; they look across an organization at how different systems work together and ensure projects adhere to solution requirements.

Their responsibilities include the following:

- Contribute to solution design.
- Build system blueprints.
- Evaluate systems against organizational standards, such as information security.

Developer or Engineer

Developers and engineers represent multiple roles that create the final product. Each software engineer or developer possesses specialized expertise in their field. For example, software developers create the front-end and back-end, write the code, and create the products.

Their responsibilities include the following:

- 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 or Quality Assurance

Test and quality assurance specialists help ensure code quality. They work closely with developers throughout to build in quality, and they help prevent errors from escaping into production.

Their responsibilities include the following:

- Write tests against requirements.
- Run tests on completed code to find defects and bugs.

Subject Matter Expert

SME is a general label for anybody who is a respected expert in their field. The SME is the person you turn to when you need somebody to explain a complex topic simply or solve advanced issues.

SME is not an explicit role on the team, but many of your team members are SMEs in their areas. However, you can recruit SMEs on a short-term basis as well. For example, suppose your project team decides that they would like to create a survey to gather information. However, nobody on the team has created a survey before, and they would like guidance. In this situation, you could contact somebody who designs surveys and ask them to give a quick 60-minute overview of how to create surveys and use the software. They wouldn't be a team member, but they are a valuable SME.

As you are adding team members, capture the following data about each member as well:

- Position
- Core/Extended
- Availability
- Subteam or Grouping
- Project Responsibilities
- Preferred Contact Methods
- Manager
- Organization

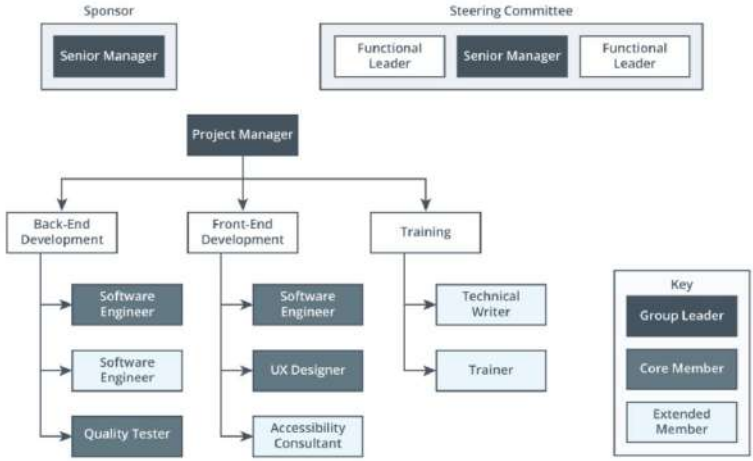
	Robin	Casey	Taylor
Position	Software Engineer	Technical Writer	Web Accessibility Consultant
Core or Extended	Core	Extended	Extended
Availability	100% for 2 years	25% (10 hours per week)	As needed, up to 50 hours per month
Subteam	Back-end Development	Training	Front-end Development
Project Responsibilities	Python & Java Development	Technical Documentation & Training	Accessibility Design Consulting
Preferred Contact Method	Anything is fine	Chat	Email
Manager	Blake	Lee	Colleen
Organization	Thinking App	Thinking App	Sam L. Consulting

project organizational chart a visual representation of the project's organizational structure

project interfaces the various reporting relationships that occur within a project

Sections of a Project Organizational Chart

- **Sponsor:** list the sponsor or sponsors of the project.
- **Steering Committee:** list the members of the steering committee or advisory group.
- **Project Manager:** at the top of the main chart is the PM.
- **Logical Groupings:** group the team members in a way that makes sense for your project. For example, a large project might have phases with many subgroups. A small project may use functional expertise. You can use color-coding to add more clarity, such as internal versus external members or core versus extended members.
- **Solid and Dotted Lines:** use solid lines to show a reporting relationship and dotted lines to show an indirect management relationship.



Responsibility Assignment Matrix (RAM) clarifies who is responsible for completing and managing the activities

RACI charts helps detect the level of responsibility for each project team member

- **Responsible:** completes the work.
- **Accountable:** ultimately accountable for the work. Work isn't complete until they approve it. It's best to have only one accountable person for each activity because it creates clarity of authority. For example, suppose people are accountable for a task. If one person signs off and the other person refuses, the team doesn't have a clear path forward.
- **Consulted:** acts as a consultant. They are an SME who can provide advice or direction. They may also review and quality check work for the responsible group.
- **Informed:** receives updates but doesn't interact with the tasks.

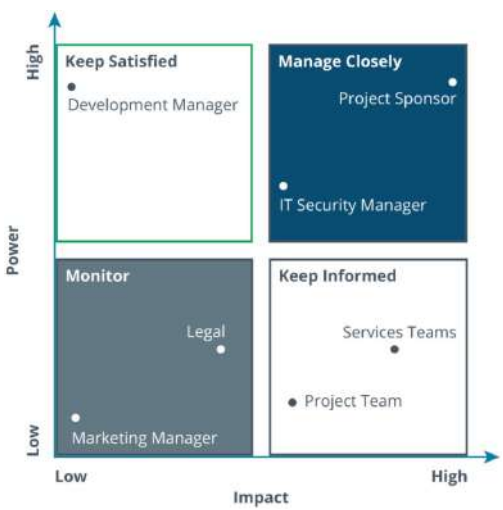
Task \ Name And Role	Bailey Lead Software Engineer	Robin Software Engineer	Casey Technical Writer	Ryan UX Designer	Taylor Web Accessibility Consultant
Activity 1	A	R			
Activity 2	A	R	I	C	C
Activity 3	I	R	A/R		C
Activity 4	R		I	A	C
Activity 5	A	I		R	R
Activity 6			A/C		R

Key	R – Responsible	A – Accountable	C – Consulted	I – Informed

Stakeholder engagement plan a management strategy that is created to ensure increase in support and minimize the negative impacts of stakeholders throughout the entire project life cycle

The grid quadrants in order of priority, highest to lowest, are as follows:

- **High Power/High Impact:** have the most to gain and the most power. It's important to work with them to build support and ensure they are satisfied with the project's progress. Their lack of support could derail a project.
- **High Power/Low Impact:** have enough influence that they could derail a project if they wanted. However, the project has little impact on them, so they might have lower interest. Engage with them enough to ensure they are satisfied with the project.
- **Low Power/High Impact:** have little power to change a project, so they are less likely to disrupt it. However, their work will change significantly because of the project. Keep them informed so they feel heard and included.
- **Low Power/Low Impact:** have little power or impact on the project. You are likely to find the lowest level of interest in this quadrant, but the situation could change. Monitor stakeholders in this quadrant in case they move to another grid. This low-priority group will require little engagement, if any.



Stakeholder register a document that identifies stakeholders of a project with information that includes their identification, assessment, and stakeholder classification

Lagging indicators are reflective measures

Leading indicators are predictive measures

SMART Objective

The SMART format is a simple way to create a goal. SMART is an acronym that stands for specific, measurable, achievable, relevant, and time-bound.

- **Specific:** should explain exactly what needs to change.
- **Measurable:** needs to include a quantifiable metric, so that you can know when you've met the goal.
- **Achievable:** meant to be realistic. Therefore, ensure the goal is within the realm of possibility.
- **Relevant:** needs to have a purpose. In an organization, it should align with business strategy or business priorities.
- **Time-Bound:** needs to have a target end date.

Objective and Key Results (OKRs)

Organizations apply objectives and key results (OKRs) to achieve tremendous results quickly. Since OKRs are meant to be aspirational and slightly out of reach, they drive teams to deliver value that they didn't think was possible.

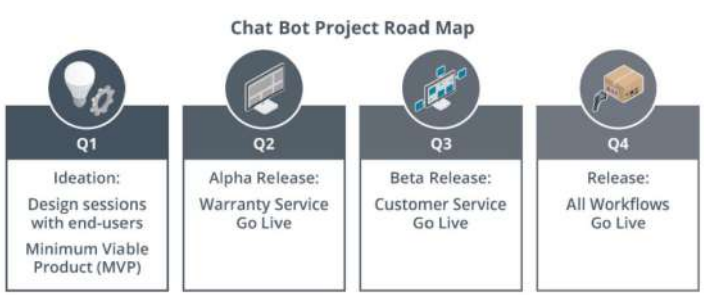
OKRs create a two-part goal statement composed of objectives and key results. Each OKR includes one objective and up to five key results.

- The **objective** is a qualitative goal statement. It doesn't need to be as measurable as KPIs or SMART objectives. The Objective portion of the OKR tells the story about what the goal is for and why it matters.
- **Key Results** are quantitative measures of success. The key results will look similar to SMART objectives. They describe what you need to do to meet the goal.

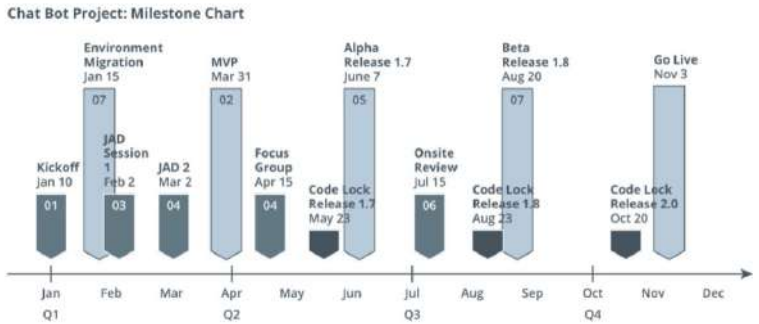
You can write an individual OKR for a project, but the OKR practice creates an entire goal-setting framework that helps organizations align to the top business priority. OKRs take a less is more approach. Organizations using OKR want teams to focus on the critical few changes that support the organizational need. The framework requires organizations to identify the top priority. Next, individuals and teams create OKRs that align loosely with the corporate priority.

Timeline Charts

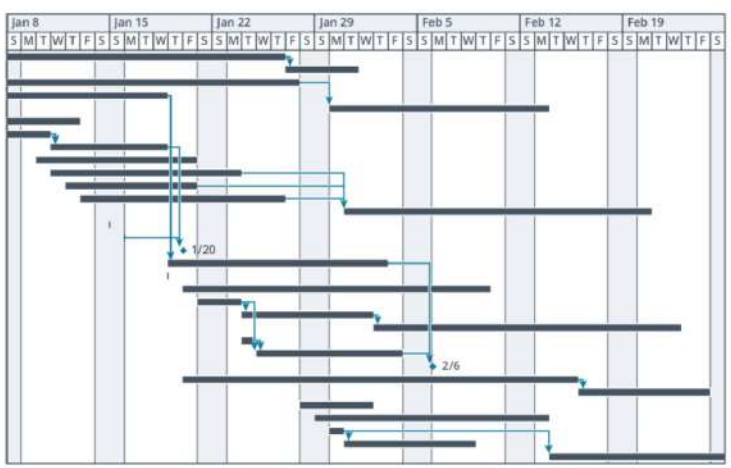
Project road maps are high-level documents that share major progress points, such as new feature rollouts. A road map is highly visual and resembles the format and level of detail that you might read in a press release. This visual works well for executive summaries.



Milestone charts have more detail than a roadmap but are not exhaustive enough to form a plan. Milestone charts describe the milestone events, which are project checkpoints. They look like a timeline and plot deadlines, check-ins, and critical dates over the course of the project. This chart is a versatile tool for stakeholders; it provides enough detail to help people understand the project sequence but isn't so comprehensive that it's hard to read. You will use the milestone chart in the project charter and kickoff materials during the initiation phase. In addition, it's helpful to use for any presentation where you need to share an overview of the project; you may use a version of the milestone chart in every project phase.



Gantt charts are highly detailed project plans. They show tasks and time in a grid to create a visual schedule view in a unique bar chart that shows time across the x-axis and a list of tasks or activities on the y-axis. The bars represent the time that each activity will occur. Additionally, Gantt charts show relationships and dependencies between tasks. Where roadmaps and milestones show events, Gantt charts show the tasks needed to reach the events. The team will use the Gantt chart during the planning, execution, and closing phases because it helps them plan and deliver the work.



project charter a brief, formal document created in the project initiation phase that outlines the project parameters

General Project Information

1. **Project Name:** give the project a short title. This name will appear on subsequent reports and dashboards.
2. **Description:** write a short sentence that expands on the project name to describe the project. This field will also appear on reports and dashboards, so you want something straightforward and understandable.
3. **Version Number and Date:** track the revision history according to your company's requirements.
4. **Project Manager:** list just the PM's name here. You will provide additional information later in the charter.
5. **Project Sponsor:** list just the sponsor's name here. You will provide additional information later in the charter.

Project Vision

6. **Objectives:** identify the specific milestones or targets. The objectives should be clear, observable, and measurable; in other words, you should know when they have been met.
7. **Purpose:** the business case for the project. List organizational priorities that this project will affect, such as the strategic plans or enterprise-level objectives.
8. **Preliminary Scope:** set the project boundaries, and clearly state what is in and out of scope. List only high-level deliverables in this section because you will refine the content in the scope statement
9. **Success Criteria:** identify the measures of success for the project. Usually, you will list the time, cost, scope, and quality constraints. However, the success criteria can include any relevant factors. For example, a PM could measure team happiness or stakeholder satisfaction scores as success criteria if they wanted to increase focus on relationship building. A company promoting social initiatives might encourage teams to measure community impacts, such as jobs created.
10. **Expected Benefits:** describe how the business will change as a result of completing the project.

Organization Overview

11. **Stakeholders:** the main stakeholders are the people who can make or break the project. The stakeholder engagement plan will list more people and groups. At this stage, look for those who will have the most influence or experience the most impact from this project.
12. **Customers and End-Users:** describe who will either use or purchase the project's outputs. This section doesn't include names; instead, use a general description, such as groups, demographics, or divisions. Note if they are internal or external to the organization as well.
13. **Project Roles:** list the PM and project sponsor. List the project board, steering committee, or other governing bodies that will oversee this project. Include high-level responsibilities as well.
14. **Team Members:** list each member's name, role, and responsibilities on the project.
15. **Project Organization:** create a project organization chart to show the project team and reporting structure.

Implementation Overview

16. **Timeline and Milestones:** capture a general timeline and critical milestones. Later, you will create a detailed project schedule to show all activities and milestones.
17. **Resource Requirements:** summarize the resources you will need for the project. Identify the labor, equipment, materials, and capital required.
18. **Risks:** list the risks that stakeholders and sponsors will be most interested in. The risks listed here will not replace the risk register. Usually, this list will include high-impact and high-probability risks.

19. **Assumptions and Constraints:** list the assumptions and constraints that the project will operate within, because it helps you manage expectations with the project sponsor and stakeholders. Assumptions and constraints are similar; they are both factors that will affect the project plan. They differ because constraints are known truths, and assumptions are assumed to be true. We operate with constraints and assumptions all the time. For example, your alarm might go off at 6:00 a.m.; this is a constraint because you know you will not sleep undisturbed past 6:00 a.m. You assume you will wake up at 6:00 a.m. because you believe you won't use the snooze button; this is an assumption. The 6:00 a.m. alarm is nearly guaranteed to occur, but the snooze button might or might not happen.

20. **Budget:** share anticipated project costs. The charter doesn't need to include an entire budget. However, the total figure should be accurate enough that the project sponsor and stakeholders can make an informed decision.

project scope statement defines the project and what it does and does not need to accomplish

Elements of a Preliminary Scope Statement

- **Project Scope Description:** summarize the project's intent and purpose. This section summarizes the project vision contents of the project charter.
- **Project Acceptance Criteria:** list the success criteria of the project. Acceptance criteria are the measurable outcomes of success. This section expands on the success criteria element of the project charter. Where the success criteria may have listed three key measures, you might list six or more acceptance criteria on this scope statement.
- **Project Deliverables (In Scope):** set the project boundaries and identify what the team will deliver. The items listed here are in scope, and your project plan will need to ensure they are completed. Listing items in scope is essential because anything not in scope is assumed to be out of scope.
- **Project Exclusions (Out of Scope):** set the project boundaries, and identify what the team will not deliver. There are more things out of scope than in scope. However, this shouldn't be a laundry list of all the things the team won't do. Instead, use this section as a clarification point, and identify the items most likely to cause scope creep throughout the project. For example, list the systems nearly adjacent to the project deliverables and historically contested processes in your organization.
- **Assumptions:** list the project team's collective assumptions about the project. This list will expand on the assumptions shared in the project charter. Treat this as a living document, and update the list as assumptions are proven true or false and new assumptions arise.
- **Constraints:** list the known constraints and maintain the list throughout the project. Items might move between the assumptions and constraints section. Proven true assumptions become constraints, and unproven constraints start as assumptions. For this reason, templates often combine assumptions and constraints into a single field.

records management the process of controlling data throughout its lifecycle

Types of Records

You will incorporate many types of artifacts into the records management plan. The following are common records you will need to manage.

- **Project Management Artifacts:** the artifacts used to manage the project, such as the project budget and stakeholder engagement plan.
- **Project Artifacts:** the documents created to move the project along, such as the requirements documentation and blueprints.
- **Legal Documents:** the project will generate project contracts and other legally binding agreements.
- **Communications:** meeting notes, status reports, and written communications generate communications records.

Types of Data

Project information is a company asset, and some of the information provides a competitive advantage against competitors. Furthermore, the company has a legal obligation to protect certain data types. For example, in the United States, the Health Insurance Portability and Accountability Act (HIPAA) requires companies to protect people's personally identifiable health information with adequate security measures. Therefore, it's essential to understand the data classification level of every project artifact.

- **Public data** has no restrictions. Anybody can access public data; it is the information found on the company website and in public records.
- **Internal data** is available to anybody within the company. It isn't necessarily secret, and exposing this data wouldn't violate laws or trust. Examples of internal data include work procedures and project plans.
- **Confidential data** has limited access. The information is only available to authorized users either on a role or case-by-case basis. Personnel records are confidential data.
- **Restricted data** is top secret, and access is severely restricted. This is the highest-security information. The company is likely to bring criminal charges against unauthorized users or face charges if it inadvertently leaks the information. Examples of critical data include research, top-secret projects, and user's passwords.

Requirements of a Records Management Plan

- **The plan needs to comply with organizational policy.** Most organizations build policies for retention, storage, records recovery, naming conventions, standard headers, or formats. Find the corporate policy, and use it as the starting point for your records management plan. Additionally, your PMO may have standards specific to project documentation, such as storage locations or retention periods.
- **Records need to be organized and easy to find.** If the artifacts are hard to find, nobody will look at them. Therefore, look for the most convenient way to store and retrieve the files.
- **The plan needs to make information accessible.** Team members need to have access to the file locations. They also need to have the ability to open the file formats or use the applications. Common challenges are external parties who cannot use internal channels and specialized software with uncommon file types.
- **Information needs to be inaccessible to unauthorized users.** The plan needs to ensure that records are only available to people who need access. Your organization's records and information security policies will establish standards as a starting point for you. Expect to see standards requiring encrypted file transmission, data classification on records, and limitations on record storage locations by data classification level.
- **The plan needs to establish accepted communication channels.** Summarizing the requirements above, building the records management plan lays the foundation for project communications. Through this plan, you will set expectations for how teams share information and, consequently, how they will not do so. Your communication channels need to accommodate all the requirements above and create open channels with your team members and vendors. This requirement seems harder to meet for external customers because internal communication channels already exist. However, even internal teams are cross functional, and members will have different ways of operating. A good records management plan creates a single point of reference for everybody.

Elements of a Records Management Plan

Every record flows through a three-stage life cycle, and the records management plan should include standards for each stage. This approach makes a simple plan for users to follow.

1. **Creation:** define naming conventions, file owners, and who can create files. Include data classification standards and confidentiality expectations in this step.
2. **Storage and Retrieval:** identify where the team will store files and the file organizational structure. Address the process for adding new versions and managing old versions. Establish standards for retrieving and sharing the records. For example, note if the team should email files or only share links to a single shared file. Set restrictions on downloading or sharing files outside the project team.
3. **Archival and Disposition:** define how to handle duplicate files and outdated files. Decide how long records will be archived after the project closes.

project kickoff marks the end of the initiation phase and the start of the project work

Facilitating Effective Meetings

Synchronous communication is real time communication

Asynchronous communication occurs when information exchanges include a delay

The meeting **facilitator** guides the meeting's progress

The **scribe** records the meeting for the meeting minutes, action items, and follow-ups

Informative meetings distribute specific information to increase an organization's knowledge about a subject

Examples of Informative Meetings

- **Demonstrations and presentations:** in demonstrations and presentations, the presenter shares product features or offers training. They might demonstrate functionality with the actual product, a video, or a slide show. By the end of the meeting, the audience should be more knowledgeable about the topic. For example, they should be able to make an informed decision about supporting a project or be able to apply the training they received. Agile teams use demonstrations to gather feedback from stakeholders. At the end of each sprint, the group invites all stakeholders to a product demo, where they present the latest product functionality. During the meeting, stakeholders ask questions and suggest changes. The team and product owner use the stakeholders' feedback to stay the course or adjust the backlog. As a result, demonstrations help a team's outcomes align with the stakeholder's expectations.

- **Status:** project status meetings give an update on progress at a specific point in time. These meetings vary in frequency and agenda, depending on the audience. For instance, internal project meetings occur more often than meetings with the customer. Status meetings allow the audience to quickly assess a project's state across multiple criteria. Since status meetings present summarized data and key points, you can cover many topics quickly. Status meetings help the audience stay updated on the project without investing too much time in the minutiae. At the same time, the speed of information can create an overwhelming experience for the audience, where they can't process all the information.

- **Stand-ups:** daily stand-up meetings were popularized in Scrum as the Daily Scrum meeting, but the format works so well that many teams adopt them. Stand-up meetings are short—typically 15 minutes or less. They received their name because participants often stand to attend the meeting, which keeps it focused and brief. For instance, the team might hold the meeting at a Kanban board instead of in a conference room. Since people get tired of standing at a board, the meeting won't drag out. Stand-ups happen at the same time and place, usually daily. Teams use this time to report on the status of their work. This meeting format ensures the entire team understands the WIP, priorities, and issues. The group addresses quick concerns in this meeting, for example, problems that can be corrected in two minutes or less. However, team members can also arrange to meet after for brainstorming and problem-solving sessions.

Decisive meetings one or more attendees are expected to make decisions

Collaborative meetings interactive sessions where attendees work together on a shared goal

Workshop an umbrella term for many collaborative group meetings

Joint Application Development (JAD) the team and customer meet and design a product together in a working session

Focus groups trained moderator-guided interactive discussions that include Stakeholders and Subject Matter Experts

Moderators use several techniques to facilitate focus groups.

- A **single-moderator focus group:** uses one moderator, who facilitates the session and monitors interactions. However, they may recruit a helper to take notes or record the session.
- A **dual-moderator focus group**: employs two moderators. One is the lead facilitator and keeps the interactions flowing. The other serves as a quality check; they track the topics and ensure the group covers all the content.
- A **dueling-moderator focus group:** also uses two moderators. However, both moderators are actively engaged in facilitation and take opposite views. As a result, this technique generally draws out the most viewpoints.
- A **two-way focus group:** allows one group to observe another focus group's session. This technique gives the focus group more perspective and can create more complex responses.
- A **respondent-moderator focus group**: uses participants as moderators, creating dynamic energy with many new ideas. This format can rotate moderators or select a single moderator. Rotating also minimizes a single moderator's impact.

Brainstorming many people contribute ideas to help reach a goal or solve a problem

There are many brainstorming techniques suited for different group sizes. Most methods work well in virtual or in-person formats, and many brainstorming applications are available as well.

- **Rapid brainstorming:** gives team members a short time limit to write down 3–5 ideas. The time limit should be brief enough to create a sense of urgency but long enough for everybody to capture their thoughts. After writing everything down, the team members take turns sharing their ideas. This method allows everybody to participate, and it generates many ideas. In addition, the group tends to refine the list as they go, so it can quickly create thoughtful complex ideas. This method works particularly well for senior executives. When senior management engages in brainstorming, the crowd tends to follow them, but this approach gathers ideas before anybody speaks.
- **Reverse brainstorming:** occurs when you look at a problem from different angles. For example, suppose you are brainstorming how to solve a problem. With reverse brainstorming, you would brainstorm ideas for why you could never solve the problem. This technique can reveal remarkable ideas. By thinking about why you couldn't solve a problem, you might learn it is more solvable than you realized.
- **Stop-and-go brainstorming:** combats the quantity-over-quality issue. First, the team brainstorms for a short period, perhaps ten minutes, and then they stop brainstorming to review and refine the results. Next, they start brainstorming and repeat the cycle. This method gives the team time to generate ideas. However, the periodic evaluations keep the content on track and offer more frequent feedback than a single brainstorming activity.
- **Round-robin brainstorming:** gives every participant an equal voice. Each person offers one idea on the topic. The group critiques the ideas after everybody contributes once, then the process repeats. This method works well if some attendees dominate the conversation.

POWER Start Agenda Technique

One method for creating a solid meeting agenda is to use the POWER start technique developed by the Agile Coaching Institute. POWER is an acronym representing five elements of a productive meeting. POWER statements have two outstanding benefits:

1. They help you create engaging, productive meetings. Thinking through the POWER elements will help you describe a meeting from multiple perspectives. You also think about advanced facilitation techniques, such as making a meeting more interactive for attendees.

2. They help you create an easy-to-read agenda. For example, you can list the POWER Statement at the start of the meeting, followed by the topics and schedule to create a short yet comprehensive meeting agenda.

The POWER statement elements are the following:

- **Purpose:** Describe the reason for the meeting, including the general topic and why a meeting is the best approach.
- **Outcomes.** List the expected results or deliverables that the team will create.
- **What's in It for Me (WIIFM):** Explain how the meeting will benefit the attendees. Meetings can seem like a one-sided arrangement, where the attendees help the organizer but get nothing in return. Use this line to clarify why attendees should want to be there.
- **Engagement:** Describe how attendees can or should participate in the meeting. You can create engaging exchanges by creating activities and setting expectations with the team.
- **Roles and Responsibilities:** Explain what responsibilities various attendees will have. For example, list the facilitator, explain that SMEs will create diagrams, and everybody will brainstorm ideas.

POWER Statement Elements		Examples
P	Purpose	Officially kick off the Chat Bot Project (P457).
O	Outcomes	At the end of this meeting, all team members and stakeholders should understand the project's expectations, objectives, and deliverables.
W	WiiFM	Meet your teammates and learn more about the project plan.
E	Engagement	Review the project charter and scope statement. Share any concerns and ask any questions related to the scope.
R	Roles & Responsibilities:	▪ Facilitator: Jo K. ▪ Scribe: Lena L. ▪ Ask Questions: Everybody

timebox an agreed-upon, fixed length of time allocated to a specific topic or activity

Action items are tasks that somebody will complete after the meeting

team life cycle a sequence of stages that describe a working team's interactions, productivity, and needs at varying maturity levels

forming marked by low productivity and friendly introductions

storming individuals are competing for position and learning about each other

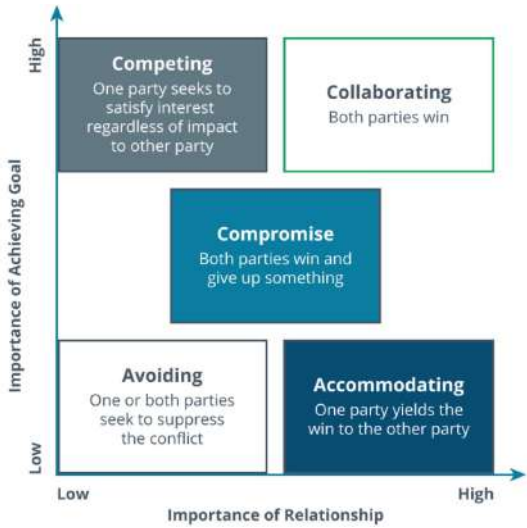
norming marked by increased productivity and reduced conflict as teams learn how to work together

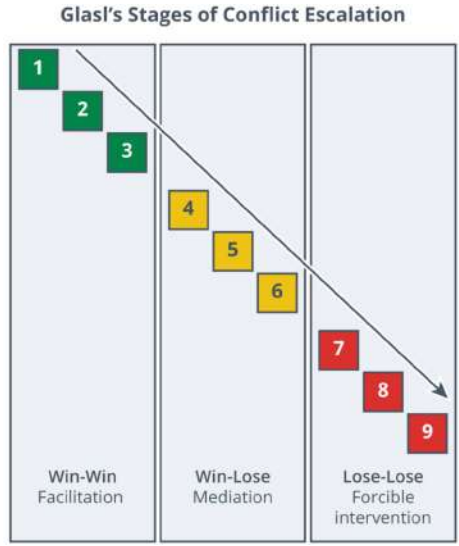
performing marked by cohesion and focus on the work

adjourning concluding work and preparing to disband

substantive conflict a task-based conflict

emotional conflict conflicts around relationships and feelings





Total Cost of Ownership (TCO) the associated costs of an asset

Operational Expense (OpEx) the spending of business funds for ongoing business costs, such as utilities, payroll, etc...

Capital Expense (CapEx) spending business funds to buy or maintain fixed business assets, such as data centers, servers, and buildings

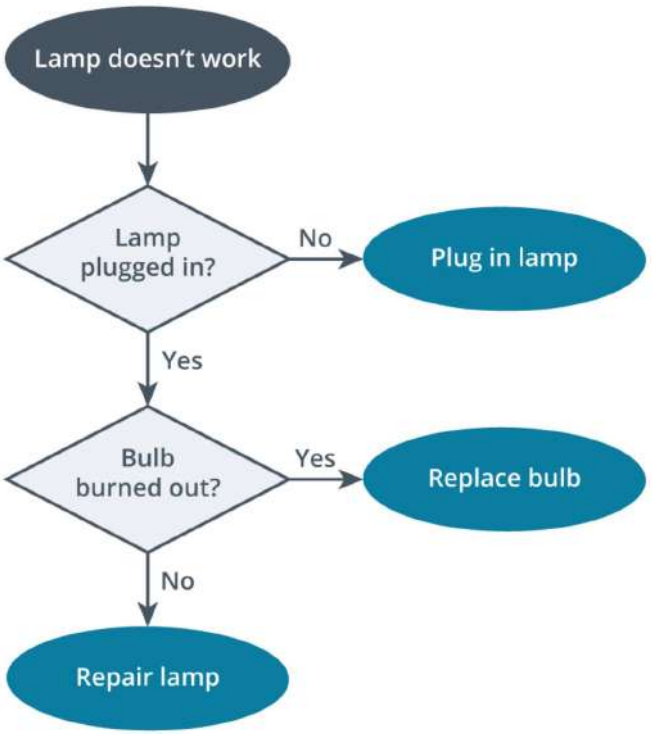
Project management software a software application that generates and organizes resource information

Gantt Chart

The Gantt chart might be the most familiar chart for PMs. A Gantt chart is an advanced scheduling chart. It's a particular type of bar chart that shows time across the horizontal axis and a list of tasks or activities on the vertical axis. The bars represent the time that each activity will occur. You can also show milestones, deadlines, and dependencies between activities. A Gantt chart creates a visual project schedule. Some software options include built-in functionality to automate resource and time calculations. You can customize Gantt charts to show varying levels of detail as well. For example, you can create a project phase chart or a task chart for different audiences.

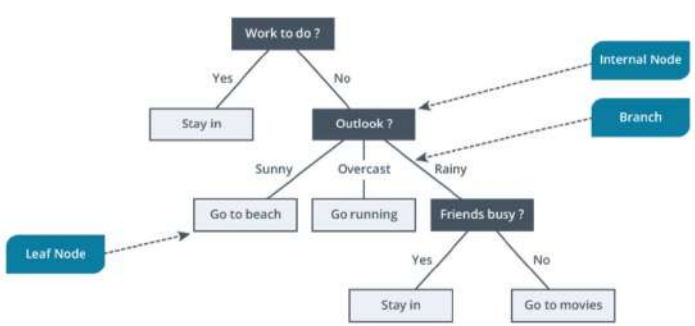
Flowchart

A flowchart is also referred to as a "process flow" or "process diagram." This chart visualizes a process, and it can compress complicated work instructions into a few charts. Project teams often create multiple flowcharts to show how a project affects processes. They are also commonly used in process analysis. For example, current state charts describe a process in the as-is state. Future state charts show the process after implementing project changes. Flowcharts are great additions to slideshows because they are easy to read and understand; removing or changing steps is incredibly easy to demonstrate with a flowchart.



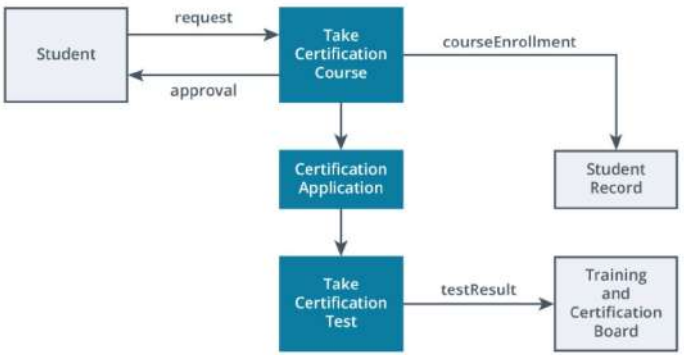
Decision Tree

Visually, a decision tree looks a lot like a flowchart. However, a flowchart describes a process that could include many decisions. A decision tree represents a single decision that could have many outcomes. Use a decision tree when a single decision requires consideration of many factors.



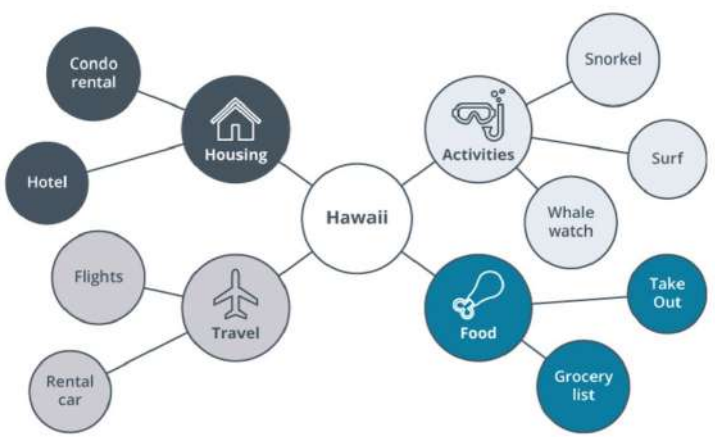
Data Flow Diagram

A data flow diagram (DFD) shows how information flows through a process or system. This diagram type works particularly well for audiences that include people with technical and nontechnical expertise. Information management is a traditionally technical topic, but DFDs present it using clear language and simple visuals.



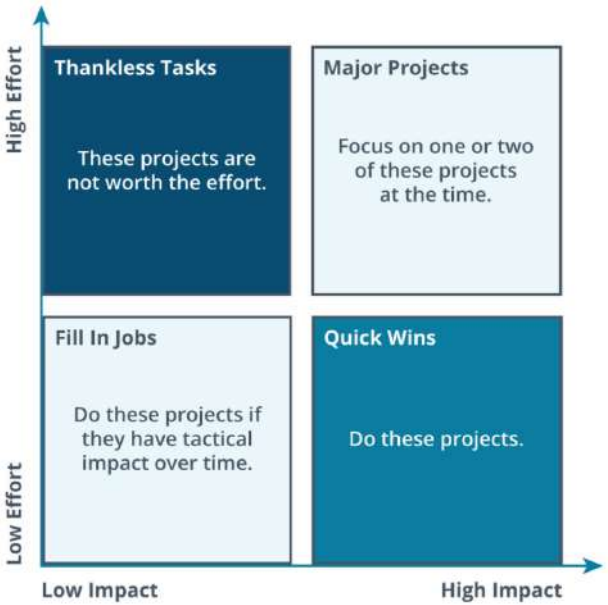
Mind Map

A mind map visualizes categories of information around a single theme. Mind maps are excellent for facilitating brainstorming activities, thinking through problems, or consolidating information.



Quadrant Diagram

This helpful diagram type includes many implementations that use the same concept. This diagram creates a four-quadrant grid. One factor is placed on each axis, and each quadrant represents a high-low combination of the two factors. Quadrant diagrams are an excellent tool for quickly assigning values or showing the relative priority of multiple options.



Communication plan a document that describes the project team's approach to communicating project information

Communication requirements analysis an investigation that leads to a clear articulation of the stakeholders' communication needs

escalation plan outlines when an event should be escalated to the next level of management

An escalation plan typically includes the following four fields at a minimum. However, some plans may have more granularity.

1. **Category:** describes the type of escalation. For example, it might include the type of work to be done or the decision that needs to be made.
2. **Level:** indicates the escalation order. The first level is the first person you would escalate to, and the escalation levels continue sequentially.
3. **Escalation owner:** each escalation level will have one owner or role. The escalation owner is the person you will contact when you reach this level.
4. **Trigger:** describes the conditions that initiate an escalation. A trigger can be an observable event or a timed event. For instance, a staffing incident might need to be escalated when a team member leaves the company. However, a software environment event might need to be escalated when the environment is down for more than 60 minutes.

Implementing Solution Design

Solution design aims to optimize the product in the given environment

technical feasibility the hardware, software, facilities, and databases needed for a proposed project

Solution architect a professional who designs computer and networking systems to optimize efficiency

Functional requirements project requirements that detail the desired functionality, capacity, or capability expected from the project

Nonfunctional Requirements (NFRs) necessary system attributes that do not present as functionality but instead describe how a system operates - many end in "-ility"

Examples of Nonfunctional Requirements

- **Usability:** refers to how user friendly the solution is.
- **Maintainability:** defines how easy it is to change the solution later. For instance, you might hear systems or code referred to as "brittle." Brittle code has low maintainability because it breaks easily.
- **Scalability:** describes how easy it is to grow and shrink the solution as needs change.
- **Availability** or **reliability:** refers to how long the system is running in a given time.
- **Extensibility:** is how easy it is to add new features without changing what is already in place.
- **Security:** refers to how the solution and the company's assets are protected. Authentication methods, data classification, user access, and administration procedures are part of the security requirements.
- **Portability** or **compatibility:** evaluates how well the solution can work within the existing environment. The solution components need to communicate with each other and other platforms in the company, including data exchanges, operating systems, software, and hardware.
- **Compliance:** refers to the need for the solution to comply with applicable regulations.

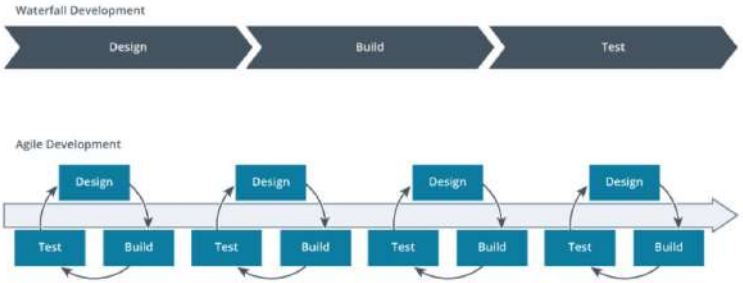
enterprise environmental factors internal or external factors that can have a positive or negative influence on the project outcome

alternatives analysis the process of analyzing the different methods of accomplishing activities

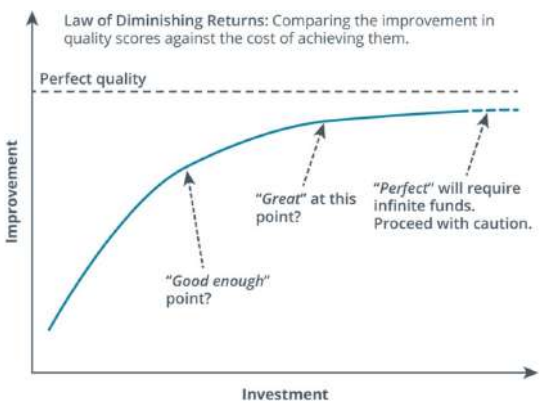
Sections of a Solution Design Document

1. **Overview:** provide a summary of the project or solution designed in the document.
2. **Audience:** clearly describe who will use this document.
3. **Purpose**: describe how the audience will use this document.
4. **References:** name any other documents that the audience will want to be familiar with, such as the project charter or scope statement.
5. **Glossary:** this document is technical and probably uses terms, abbreviations, initialisms, and acronyms unfamiliar to a casual audience. Include a glossary of terms to clarify the document without dragging out the text sections.
6. **Summary of Existing Functionality:** describe the current system performance or how people are working today.
7. **Functional Requirements Details:** clearly describe business requirements. The team will use these notes to create tests later, so they need to represent the stakeholders' expectations accurately. In an agile project, this section would include user stories and epics.
8. **NFR Details:** NFRs are the requirements the project needs to meet before it goes before the stakeholders. Describe the organization's expectations of system performance, and the team will create tests against them. This section would include tests or requirements for all user stories and epics in an agile project.
9. **Assumptions and Prerequisites:** capture the operating conditions the team needs and that they assume to be in place. During the review, the stakeholders should pay special attention to this section. False assumptions can lead to significant slowdowns and impediments.
10. **HLD:** create visuals, such as process flowcharts and data flow diagrams, to describe the systems and integrations.
11. **LLD:** provide thorough details about all changes. This section is the main part of the LLD document. The LLD includes the technical plan for the project; an excellent LLD means the team can work on the project even if the architect is unavailable. The architect captures software-related implementation details, including products used, configuration requirements, and product specifications. They also capture hardware-related implementation requirements, including equipment needed, license and storage method, and known configurations.
12. **Impact Analysis:** evaluate how this solution will affect the existing environment. For example, suppose a new application requires updating all browsers to the latest version. Identify if other current systems aren't compatible with updated browsers or if additional testing is needed.
13. **Out-of-Scope**: clearly describe the project's boundaries and what is not in scope for this solution. Stakeholders should also pay special attention to this section because it adds to the project's boundaries.
14. **Risks and Mitigation:** document any known solutions risks and the current mitigation strategy. This section can also cover critical tests or known unknowns.
15. **Appendix:** the appendix is optional. Include an appendix if you have valuable but not integral information that the team might appreciate.

Solution Design in an Agile Project

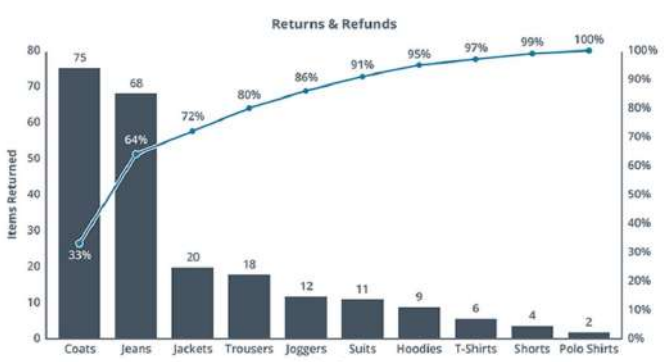


Law of Diminishing Returns



Pareto principle asserts that you can solve 80% of a problem by addressing 20% of causes

A Pareto chart visualizes the Pareto principle using lines and columns. Each column represents the counts of a category, and the columns appear in descending order by total count. The line on the chart measures the total percentage of the count; the line will slope upward.



Multitiered architecture is a modular design, which divides software into layers, or tiers

A **data warehouse** is an extensive database used for reporting and analytics

Enterprise Resource Planning (ERP) software that enables an organization to manage services, personnel, and IT resources

Customer Relationship Management (CRM) software helps companies track and manage customer interactions

Electronic Document Management Systems (EDMSs) also called **Records Management Systems (RMS)** is software that allows the user to create, track, and share business documents

Content Management Systems (CMSs) are user-friendly systems that allow more users to complete historically complex activities

Managing Resources

human resource plan a document that provides guidance on how the human resources required for a project should be defined, staffed, managed, controlled, and released

procurement plan a document that outlines the specifications for procuring work from outside sources

The procurement plan may include the following information:

- An overview of the needed goods or services
- Delivery and implementation dates
- Critical dates
- Analysis of the impact of procuring these resources, such as financial and risk impacts
- Contract types
- Evaluation methods
- Vendor performance metrics
- Existing vendor options
- Detailed financial analysis

Calculating Shared Resource Capacity

An employee is working on two projects, Project Lemon and Project Melon. They split their time evenly across two teams. They work 40 hours per week, and they reserve eight hours for administration and development.

1. Identify the total hours and the overhead hours.

$$\text{Total Working Hours} = 40$$
$$\text{Overhead Hours} = 8$$

2. Subtract the overhead time from the total time to calculate their available time. The available time is the number of hours they can spend on projects.

$$\text{Available Time} = 40 - 8 = 32 \text{ Hours}$$

3. Divide their available time by the number of projects to determine how many hours are available for each project. This employee is working on two projects.

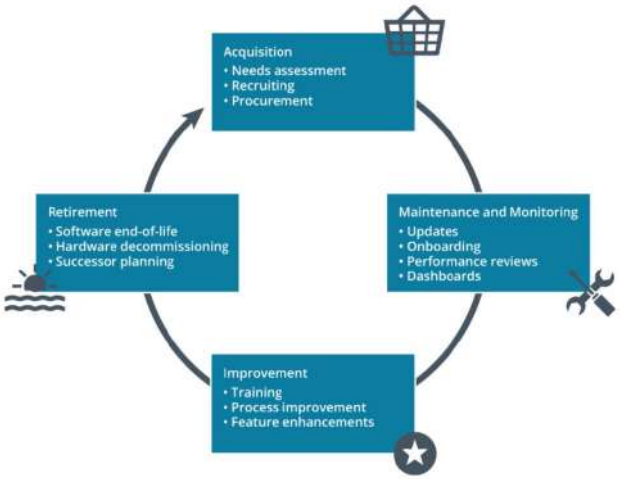
$$\text{Time Per Project} = \frac{32}{2}$$
$$\text{Time Per Project} = 16 \text{ Hours}$$

4. Allocate time to each project.

$$\text{Project Lemon Allocation} = 16 \text{ hours per week}$$
$$\text{Project Melon Allocation} = 16 \text{ hours per week}$$

You can schedule 16 hours of work to this employee each week. If you exceed that number, you would be taking capacity away from the other project or forcing the employee to work overtime. Both situations cause stress for the employee and damage relationships, so it's critical to treat allocations as the upper limit.

Resource Life Cycle



Succession planning is a proactive practice where companies create a plan for who would replace critical roles in a company if they were suddenly vacated

The Needs Assessment Process

1. **Understand the project's requirements:** identify the project's objective, success criteria, milestones, and deliverables. Use this information to start defining how the project will flow, and then identify the resources needed to make it run efficiently. Finally, identify any other resources or processes you would like; you can always pare down the list later, so dream big now. Turn these wish lists into a cohesive summary or map, and this is your future state.

2. **Identify the current resources:** look across your resources to determine what you already have. The resource pool is broad; it could be specific team members, software, or equipment. Also, take note of the resource's availability. For example, it's good to know you have a cloud migration expert on staff. On the other hand, it's better to know you have a cloud migration expert who is already working on multiple projects for the next six months. Turn this resource list into a cohesive summary or map, and this is your current state.

3. **Identify any resource gaps:** also known as "gap identification." Review the sections where the current and future states differ, and these are the gaps. Document each gap.

4. **Address the gaps:** create a plan to close each gap identified in the previous step. Next, put the plan into action. This step is the purpose of the needs assessment. A needs assessment identifies unmet needs early in the project so that you can proactively correct them.

Gap analysis compares the current state and future state and describes the differences

Utilization Gaps

Resource utilization is a project KPI that measures the effectiveness of project resources. Utilization is expressed as a percentage, where 100% represents a fully utilized resource. It measures how much a resource's available time is already allocated to tasks. Anything under 100% is underutilized, and anything over 100% is overallocated.

In project management, resource utilization uses two values:

- **Time allocated** is the total amount of time that a resource is scheduled to work on any task in the project.
- **Total time available** is the total amount of time a resource is available to work on the project.

Resource utilization is calculated by dividing the allocated time by the available time.

$$\text{Resource Utilization} = \frac{\text{Time Allocated}}{\text{Time Available}}$$

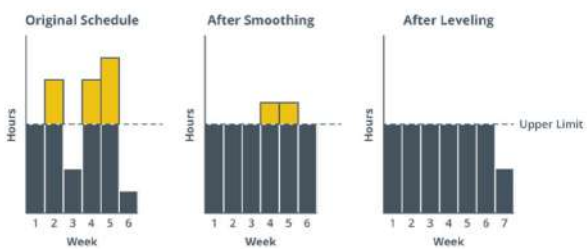
The project team is using a 3-D printer. The printer is available for 400 hours, and the project schedule includes 500 hours of 3-D printing time.

Divide the allocated time by the available time to calculate the utilization rate.

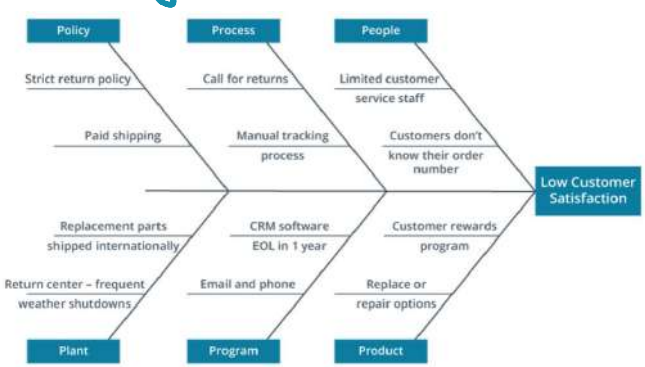
$$\text{Utilization Rate} = \frac{500}{400}$$
$$\text{Utilization Rate} = 125\%$$

Resource leveling is an optimization technique that reduces variation and allows the project timeline to extend

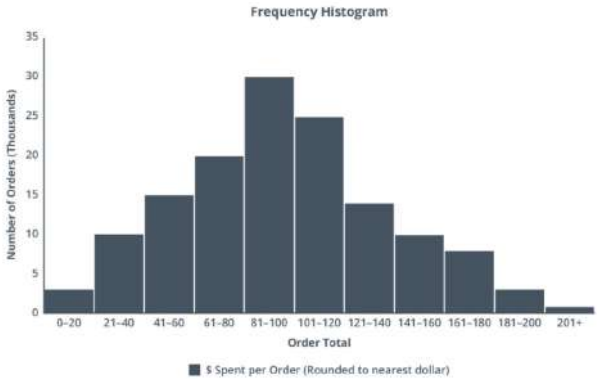
The resource leveling process moves tasks until no resources are overallocated. It will add additional time to a project if necessary.



fishbone diagram cause-and-effect diagram



Histogram a chart used in statistical analysis



Managing Risk

A **risk** is something that you think will happen that you don't control

A **issue** is something that is already happening that you don't control

issue management documents issues after they arise and recovers from them

Risk Management Overview

1. **Identify the Risk:** Create a list of all potential risks. Risks come from all areas, including previous experience, the current environment, and other projects. You've probably been thinking about risks since you first heard about the project. Now is the time to write everything down. Risk identification will require cleanup and refinement of the list. You will need to manage all the risks throughout the project, so take time now to remove duplicates and clarify the language. Finally, record the refined list in a risk register, which you will maintain throughout the project.

2. **Analyze the Risk:** Determine how likely a risk event will happen; this is the probability. If the risk were to happen, identify how it would impact this project, other projects, and the organization. Use this information to assign priority levels to the risks. This process is known as effect-based risk classification.

3. **Treat the Risk:** Create a risk management strategy for every risk. That assigns an owner and response plan for every risk in the register. The response plan describes what the risk owner will do to prepare for a risk's occurrence.

4. **Monitor the Risk:** Create a plan to monitor the risks, including progress on response plans and the occurrence of risk events. Maintain the risk register throughout the project by adding new risks and updating existing risks. Report on high-priority risks as their probability or response plan changes.

risk owner develops a response strategy for an individual risk

risk manager administers the risk register

risk analysis process for qualifying or quantifying the likelihood and impact of a factor

Qualitative risk analysis is subjective; it's based on how people perceive and interpret the risk

Quantitative risk analysis is objective; it uses verifiable data to assign scores to project risks

Interconnectivity describes how risks interact with each other and create ripple effects across a project

quality audit an independent evaluation, inspection, or review of a project's quality assurance systems

quality escape a deliverable containing an unacceptable deviation or defect that was not detected by the producer or quality team

- **Notice of Escape** a formal notification informing the supplier of a defect

cost of quality the total cost of effort to achieve an acceptable level of quality in the project's product or service

probability and impact risk rating matrix the assignment of risk rating to risks or conditions

Failure Mode and Effect Analysis (FMEA) is a structured risk analysis tool used frequently in manufacturing operations and product design

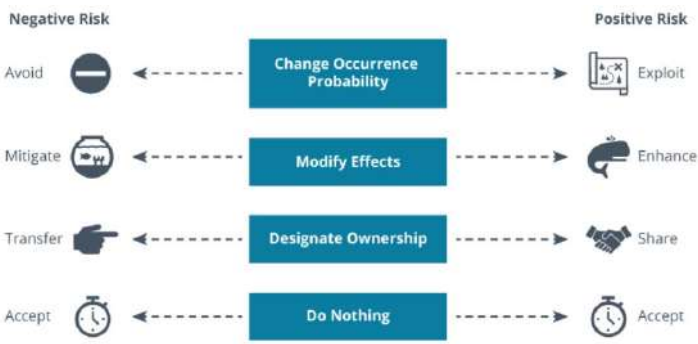
Scenario analysis where you generate potential events and evaluate the impact

Negative risk strategies

- Avoid
- Mitigate
- Transfer
- Accept

Positive risk strategies

- Exploit
- Enhance
- Share
- Accept



contingent response strategy meant to be used in the event that identified risks become reality

Contingency plan implemented if the initial response strategy is ineffective in responding to the risk event

risk register a document highlighting the results of risk assessments in an easily comprehensible format

The risk register includes at least the following fields:

- Risk ID
- Risk description
- Risk analysis summary
- Impact, priority, or other scoring criteria
- Risk priority or ranking
- Risk owner
- Risk response or treatment

risk report a periodic summary of relevant information about a project's risks

risk management plan a document that describes the team's approach to manage risks

Risk Management Plan Elements

- **Approach:** Describe how the process will work and which analysis tools will be used. Include any specific terminology, such as if you are choosing to use "threat" and "opportunity" instead of "negative risk" and "positive risk."
- **Risk Identification Plan:** Include a plan for how the team will identify risks. The risk identification plan can include multiple activities and sources, but it should have a clear focus. Don't list every possible source of risks; instead, list the sources you will strategically explore.
- **Risk Register:** Identify where you will store and manage risks.
- Risk Breakdown Structure (RBS): Identify which categories you will use to classify the nature of the risks in an RBS.

Project Risk Breakdown Structure Example Categories				
Technical	**Management**	**Organizational**	**Commercial**	**External Risk**
Scope and Objective	Project Management	Culture	Vendor Stability	Geopolitical Forces
Technical Processes	Project Resources	Sponsorship	Financing	Regulatory
Technology Scaling	Communication	Support	Contract Terms	Legislative
Overall Technical Performance		Policies and Processes	Legal	Natural Disasters
User Interfaces				External Groups

- **Risk Assessment Strategy:** Specify how the team will score risks, including criteria and scales. Include definitions for any qualitative scales.
- **Risk Response Strategy:** Summarize the risk response options used to manage risks.
- **Roles and Responsibilities:** Name the roles and responsibilities for the entire risk management process. Include clear responsibilities for the risk owners and the risk manager. Describe who will add to or modify the risk register and when.
- **Funding:** List any financial factors, such as funding availability, funding needs, or resource requirements.
- **Risk Monitoring:** Describe how you and the risk owners will identify and track risks. Outline what you are looking for in the monitoring process and how you will react. (In other words, explain why you monitor risks and what you will do if you notice something unusual.)
- **Schedule:** Outline the timing of all risk management activities. Include the initial risk assessment meeting, ongoing risk reporting, and other recurring activities.

product transition a formal hand-off of a project's outcome to its recipients

transition plan describes the transition to the permanent owners

Transition Plan Elements

You can create a transition plan as a checklist or a detailed document. Any clear, concise format with all necessary information will meet the goal.

- **Roles and responsibilities:** Include who owns the transition plan, who will complete transition activities, and who will own various maintenance activities. Ensure everybody who is part of the transition plan understands their role.
- **Transition schedule:** Define the schedule of transition activities, including when the project team will officially end work on the project. Always include the go-live date because transitioning responsibility begins soon after that.
- **Maintenance schedule:** Identify the ongoing maintenance requirements for the project. This section defines the schedule of activities that the operations team will need to build into their work.
- **Training:** Identify the operational training requirements and delivery channels, such as SMEs or computer-based training. Operational training helps the operations team acquire the skills and certifications needed to maintain the system. For example, it teaches the operations team how to complete specific, routine tasks, such as error handling and configuration management. Training also includes certifications or certificates. Such credentials may be necessary by law, such as in food safety and medical care. Certifications can also be a self-imposed necessity because they produce better business results or improve the work environment. For example, many vendors offer training tailored to their products. You may define these sessions as part of the transition plan to ensure that the operations team learns about a specific product version. Training can also include knowledge transfer through SMEs. For instance, they often host product training events to impart knowledge quickly.
- **Activities:** List all the other tasks required to complete the handoff. For example, activities might include access setup, asset turnover, and review sessions.
- **Tools and techniques:** List any specific tools, techniques, or processes the operations team will need longterm. Then, incorporate any access requirements into the activities list.
- **Product documentation:** The transition plan should include product-specific information, such as code documentation, product specifications, configurations, and access and credentials. The plan might embed the materials, such as with a playbook, or it can direct the maintenance team to a permanent document location.
- **Asset turnover plan:** Create a strategy for how and when assets will move from the project team to the maintenance team. Also, include a strategy for how the team will distribute assets when both the project team and maintenance team need to use the system. The turnover plan is surprisingly crucial if the project uses high-cost, low quantity physical assets.

impact analysis the process of evaluating the impact of an event on the project

go-live when software moves from a test environment into production

operational handoff where responsibility moves from the project team to the operations team

Creating a Project Schedule

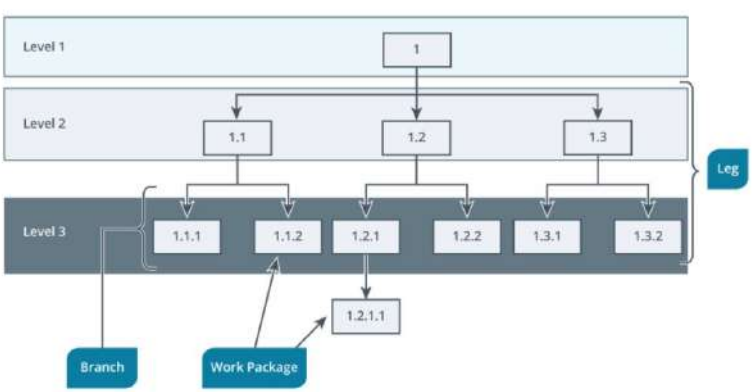
Project **milestones** represent critical checkpoints in a project

Decomposition a technique for creating the Work Breakdown Structure by subdividing project deliverables to the work package level

Breaking a project into manageable pieces is integral to the project management process. It feeds into other project documents, such as the project schedule. Therefore, the WBS needs to be reliable and accurate.

Key Terms

- **Task:** the action completed to deliver project work. Tasks are not part of the WBS, but you will encounter them throughout the project. Tasks include verbs and are things you can do, such as "install tile" and "install kitchen wiring." Later in the planning process, the project team will use the WBS or backlog to create tasks.
- **Activity:** in practice, activity can mean any work related to the project. However, this course uses the term "activity" to describe the items you would include in a project schedule. A project schedule will include a mix of WBS elements, user stories, project tasks, project management tasks, and milestones. For improved readability, this course refers to these items as "activities" unless it is discussing a specific work item type.
- **Work:** the result of doing something, not the act of doing something. In a WBS, "work" is a noun, not a verb. In other words, work is the value of the project. For example, if you are building a house, installing a tile floor might seem like the work. In project management, installing the tile is an activity, and the finished floor is the work.
- **Project schedule:** includes the WBS deliverables, timing, and resource needs. The WBS is not a project schedule, but it is the starting point. The WBS describes what needs to be delivered, and the project schedule adds on who will work on it and when it will be delivered.
- **Deliverables:** the outcomes of the project. They are observable products, artifacts, and benefits that move the project to completion. Every element in the WBS creates a deliverable.



Sections of a Work Breakdown Structure

- **Element:** the deliverables in a WBS. It is each box in a chart and each row in a table. The WBS element is short, at just two or three words. The remaining details about the elements are stored in a separate document called a "WBS Dictionary."
  - The elements of a WBS include nouns only—no verbs. For example, a home construction WBS will have the element "Electrical," not "Install Electrical." It seems like a small change at first, but it changes the document's tone.
- Work package: the lowest element in each branch or leg; an element without a child element is a work package. It's also called a "terminal element." The work package is the only level where cost, resources, and timing are estimated. All other levels are a rollup of the work packages. Like all other elements, the work package completes a deliverable. Even though it is the lowest level of the WBS, it is still an outcome-based element of work and not a task. The project team will use the work package to create all the project activities throughout the project.
- **Level:** the layer of the WBS where an element resides. The levels are numbered sequentially, usually starting at level 1, which has only one element, the project. Level 2 has the large deliverables needed to complete the project.
- **Numbering Scheme:** each element is assigned a number according to its place in the hierarchy; it is also called the code of accounts. The WBS uses a sequential numbering scheme: each level is separated by a decimal point and restarts at 1.
  - For example, let's assume our project element ID is 1. The project will have several deliverables assigned to it, which reside at level 2. To add another level to the ID, add a decimal point and start again at 1. So the level 2 elements in our example are 1.1, 1.2, and 1.3. You can tell they are level 2 elements because they have two numbers and one decimal point.
  - The process repeats for each level, all the way to the last work package. So, 1.3.1 is a level 3 element because it has three numbers and two decimal places, and it is a child of 1.3.
- **Leg:** A leg is a group of elements with a shared ancestor. The lowest elements in a leg will roll up to equal everything in the highest level of the leg.
- **Branch:** A branch is one or more elements that do not represent the whole top-level element in the branch.

Mutually exclusive, collectively exhaustive (MECE) each element exists without overlapping anything else

Visualizing the mutually exclusive and collectively exhaustive concepts.

The WBS is MECE. Level 1 is the whole project, so it's MECE automatically. When you create the level 2 elements, everything in the project's scope must fit within the level 2 elements. None of the deliverables can overlap with each other.

When you break 1.1 into level 3 elements, everything in 1.1's scope must be represented by 1.1.1, 1.1.2, 1.1.3... None of the elements at any level can overlap each other.

100% Rule

The 100% rule means the WBS includes 100% of the project work. It sounds similar to MECE, but it's slightly different. This rule means the WBS includes exactly 100% of the project work, nothing more, nothing less. The project should not include anything beyond the project's deliverables and project management work. You would include your deliverables related to project management, and you would not include anything out of scope.

WPS Creation Process

1. **Determine Scope**

Review the project scope on the project charter, scope statement, statement of work (SOW), and terms of reference.

Lesson 10 covers statements of work in greater detail.

2. **Identify the Deliverables**

The deliverables need to capture the full scope of the project. These will turn into WBS elements.

3. **Identify the Team Members**

As you identify new deliverables and work packages, add people with the necessary skills to your team.

4. **Build Level 2**

Define the level 2 elements. This level sets the grouping or construction method, so take time to create a structure that fits this project.

The grouping method or construction method for each project is your choice. However, it should make the project easier to understand, plan, and manage. Therefore, each project you lead might have a different WBS. For instance, you probably wouldn't need project phases for a two-week project, but they might work well for a two-year project.

The two most common construction methods are deliverable based and phase based. A deliverable-based method organizes the structure around deliverables and scope. In this method, the deliverables you identified in the previous step would appear on level 2.

A phase-based WBS organizes elements around the project phases. Activities that need to span multiple phases are split into multiple elements because elements need to align fully with their assigned leg. So in a phase-based construction, level 2 would be the project phases. The deliverables you identified in the previous step would need to be split and assigned to level 3.

5. **Build Remaining Levels**

Use the design guidelines to continue breaking work down into new levels. When you reach a point where you don't need to break it down again, you've reached the work package.

6. **Build the WBS Dictionary**

The WBS contents are intentionally brief, but we need more information to manage a project. The WBS dictionary includes WBS details that don't fit on a WBS diagram. While the diagram needs to be short and easy to read, the dictionary contains more specific information. It also includes information you can reuse on the schedule, budget, and reports.

A WBS dictionary is customizable, and it should include the following elements at a minimum:

- Work package ID
- Work package name
- Work package description
- Assigned to
- Date of assignment
- Due date

The backlog is the body of work that a team will complete

WPS Versus Project Backlog

The WBS and project backlog share several similarities. Neither creates activities; both create deliverables. Both strive to leave the execution details to the team. Finally, both have a specific hierarchy where the large project breaks into progressively smaller pieces of work.

The two methods differ in several ways. A WBS and project schedule will balance scope, time, and cost. Meanwhile, agile embraces flexible scope but fixes time and cost through agile ceremonies. WBS details are stored in a WBS dictionary, and product backlog details are stored within the backlog items.

The WBS has more structure than a backlog in its design. The WBS has specific numbering conventions and design rules, including MECE and the 100% rule. Backlogs do not have these requirements in their design, but you will probably see them in implementation. For instance, most agile product management software includes IDs and hierarchies to help companies organize their work. So while a WBS seems more structured in the original drafts, they have similar levels of structure in today's software.

A WBS assumes that scope, time, and cost are fixed and works to control them. A project backlog uses agile ceremonies to fix time and cost. However, the scope of work remains flexible; a backlog-based team will deliver as much of the scope as they can fit into an iteration. This model still generates adequate results because agile teams assume responsibility for increasing their productivity.

Executing a project backlog will differ from a WBS. A WBS-based project schedule assumes that all scope items will be delivered. A backlog-based schedule assumes that some items won't be delivered.

Therefore, a backlog-based project schedule means the project team would work on one deliverable set at a time. The team wouldn't invest in breakdown until the backlog item is closer to the top of the list, when they have more confidence that they will have time to work on it.

Project backlogs do not assume value. Instead, each deliverable is potentially valuable, and delivering the work tests this benefit hypothesis. Therefore, a project backlog may complete a deliverable and discard it if it doesn't provide value. When a hypothesis is proven false, the team will need to try something else, which is the equivalent of a change to the project. You can introduce change in either a WBS or project backlog. The difference between the two is that a WBS builds everything at the start and incorporates changes later. The project backlog assumes that changes are coming and therefore doesn't build work too far ahead.

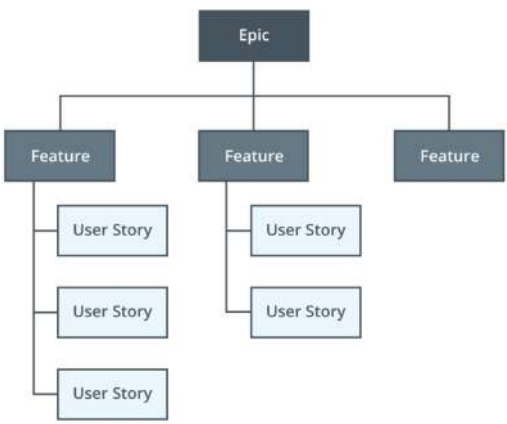
	Work Breakdown Structure	Project Backlog	Product Backlog
Duration	Length of the project	Length of the project	As long as the product is viable
Scope	Fixed	Flexible	Flexible
Resources	Fixed	Fixed	Fixed
Time	Fixed	Fixed	Fixed
Deliverable or Activity	Deliverable	Deliverable	Deliverable
Hierarchy	Numbered Levels	Epic > Feature > User Story	Epic > Feature > User Story
When Created	Planning Phase	All Phases	Entire Life Cycle

Agile Project Backlogs

epic is a large deliverable often triggered by a business case - compared to a WPS, an epic is equivalent to the project at level 1

feature is smaller than an epic but still sizeable - compared to a WPS, a feature is the equivalent of the level 2 elements

user story is the smallest agile deliverable - equivalent to a work package in both size and spirit



Project Tasks

The project breakdown process will create the lowest-level deliverables. A WBS generates work packages, and a backlog generates user stories.

Next, you can break the work packages and user stories into tasks.

In project management, a task is smaller than a work package or user story. Tasks are the actions you take to complete deliverables. Tasks describe specific steps to take. A project team will complete countless tasks while delivering a project, but they don't have to write down all tasks.

Breaking out tasks requires expert judgment to identify an appropriate level of detail. Neither a WBS nor a user story requires tasks, and as a PM, you won't visualize tasks at the WBS or user story level. Tasks are too specific and change too often to be useful on these documents. If you were to try to track every small task for a project, you would spend most of your time updating the changes. A task is small enough that a changed task will not affect the outcome; it only shows how somebody changed their approach.

A **dependency** describes a relationship between activities

The activity that starts first is called the **predecessor**

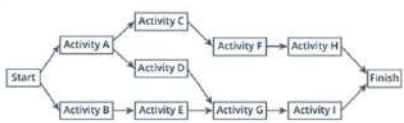
The activity the starts second is the **successor**

Precedence relationship describes the sequence in which they should be carried out

The team will identify sequencing requirements for each activity in the project. This dependency determination process can occur during a work breakdown or a separate planning session.

You can visualize dependencies in multiple ways. A project schedule lists the predecessors for each dependency. Gantt charts and project network diagrams show arrows connecting dependent activities.

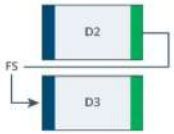
	Task Name	Duration	Start	Finish	Predecessors
0	My Project	16 days	Mon 8:00	Mon 17:00	
1	Phase 1	200 days	Mon 8:00	Wed 17:00	
2	Activity A Initiation	70 days	Mon 8:00	Mon 17:00	
3	Activity C Planning	65 days	Tue 8:00	Thu 17:00	2
4	Activity F Initiation	60 days	Fri 8:00	Tue 17:00	3
5	Activity H Planning	5 days	Wed 8:00	Wed 17:00	4
6	Phase 1 Completed	0 day	Wed 17:00	Wed 17:00	5
7	Phase 2	180 days	Thu 8:00	Mon 17:00	
8	Activity B Initiation	180 days	Thu 8:00	Thu 17:00	
9	Activity E Planning	140 days	Fri 8:00	Tue 8:00	8



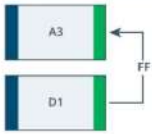
Four Dependency Types

- **Finish-to-Start (FS)** means that Activity 2 starts after Activity 1 finishes
- **Start-to-Start (SS)** means that Activity 2 starts after Activity 1 begins
- **Finish-to-Finish (FF)** means that Activity 2 finishes after Activity 1 finishes
- **Start-to-Finish (SF)** means that Activity 2 finishes only after Activity 1 has started

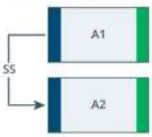
Finish-to-Start
When cloud service configuration (D2) is finished, server installation (D3) can begin.



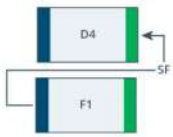
Finish-to-Finish
The technical product owner will finish approving the designs (A3) at the conclusion of the demo (D1).



Start-to-Start
The procurement office should be engaged (A2) as soon as project activities start (A1).



Start-to-Finish
Data integrity audits (F1) must be started before the cloud services turnover (D4) can be completed.



Dependency Logic Types

A team has multiple reasons for creating a dependency. This logic describes why it exists or the nature of the dependency. We use four categories to classify dependencies.

Mandatory Dependency
Also referred to as a "hard logic dependency," a mandatory dependency is unavoidable, often because of legal or contractual reasons. Mandatory dependencies also arise because of the natural order. For example, a user cannot create a password until they access the site.

Discretionary Dependency
Also referred to as a "soft logic dependency," a discretionary dependency is not mandatory. Instead, it is a preference of the project team. Discretionary dependencies often represent a strategy for optimizing efficiency, lowering costs, or improving quality. They reflect experience, best practices, or organizational processes.

External Dependency
An external dependency is outside the control of the organization. For example, waiting for the internet provider to upgrade the network is an external dependency.

Internal Dependency
The organization controls an internal dependency. For instance, if a project needs to wait for the company's legal team to review the contracts, this would be a dependency internal to the organization.

Dependency logics combine to clarify the nature of dependencies. Each dependency is either mandatory or discretionary and external or internal.

	Mandatory	Discretionary
Internal	The legal department must review contracts before they are accepted and signed.	To avoid pursuing ineligible products, the security team should complete a review before a product demo.
External	The project team cannot implement cloud file storage until the internet provider completes fiber upgrades to the building.	If the company waits until the upcoming technology conference to select a product, the project team could attend the conference and view additional product demos.

Estimation Techniques

Top-Down Estimating
Top-down estimating creates an estimate of the entire project or key deliverables. Then estimates are cascaded down to the activities. Top-down estimating is less accurate than other techniques. Nevertheless, it works well when you don't have more information during the discovery or initiation phases.

Bottom-Up Estimating
Bottom-up estimating creates estimates at the lowest-level activities, work packages, and user stories. Then, the individual estimates are added together to create a project estimate. A bottom-up estimate is more time consuming but more accurate than a top-down estimate. This course teaches bottom-up estimating.

Analogous Estimating
Analogous estimating is a top-down approach that uses historical data. For example, suppose you are estimating a marketing campaign project. If your previous similar projects lasted for six weeks, you could reasonably estimate that this project will last six weeks.

Parametric Model Estimating
Parametric estimating augments either bottom-up or top-down estimates with historical data. It uses known variables, such as cost per hour and the number of hours, to create a custom estimate for a project. For instance, suppose you need to create four new advertisements for an ad campaign. Usually, you can complete a single advertisement in two hours. Using a parametric model, you can estimate that it will take eight hours to complete the four campaigns.

Three-Point Estimating
You can use three-point estimating in combination with any other estimating technique. It factors in multiple scenarios and creates an average estimate.

For each estimate, you collect three values: optimistic, pessimistic, and most likely.

- **Optimistic (O):** The estimate if everything happens perfectly.
- **Most likely (M):** The estimate of the most likely outcome. It will be somewhere between optimistic and pessimistic.
- **Pessimistic (P):** The estimate if everything goes wrong.

The formula for calculating the three-point estimate (E) is:

$$E = \frac{(O + M + P)}{3}$$

Duration refers to the time between when a work package starts and when it ends

resource loading the process of assigning work to a person until all of their available capacity is consumed

Precendence Diagram Method (PDM)

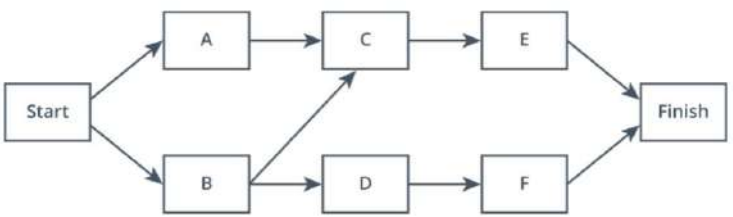
The precedence diagram method (PDM) is a popular project diagram style. PDM diagrams use rectangles and arrows to show the dependencies between activities. A PDM is useful for visualizing activities and dependencies.

The diagram layout describes the flow of work. A "Start" rectangle on the left side of the diagram represents the start of a project, and a "Finish" rectangle on the right side represents the end. The project activities are arranged in a sequential flow from left to right. The left side of the activity rectangle is the start of the activity, and the right is the finish.

An arrow connects the two activity rectangles if activities depend on one another. The dependency type is depicted in one of two ways.

1. You can use the connection points to describe the relationships. For example, an SF dependency would draw an arrow from the left side of the first activity to the right side of the second activity. This method visually shows the connection points. However, this model looks complex quickly.
2. An alternative approach is to connect the shapes from end to end and label each arrow with the appropriate relationship type (FS, FF, SS, SF). This method works well for diagrams with many activities or many dependencies.

A precedence diagram is a good starting point for the other project network diagrams covered in this course. It is a simple diagram, and the information and style are all reusable for PDM and CPM diagrams. Sketch a PDM chart to focus on the activities, sequencing, and dependencies. Then, you can layer in duration and calculations using another diagram technique.



Program Evaluation and Review Technique (PERT)

A project diagram that uses the Program Evaluation and Review Technique (PERT) represents duration, activities, and dependencies. A PERT chart uses the same diagramming style as PDM, with rectangles for activities and arrows for dependencies. It also documents a duration estimate for each activity, known as the "PERT estimate." The PERT estimate is a weighted factor. The PERT estimate requires three duration estimates for each activity:

- **Optimistic (O)** is the duration if everything goes right.
- **Pessimistic (P)** is the duration if everything goes wrong.
- **Most Likely (M)** is the duration that will probably happen.

The PERT estimate weights the most likely value by four against the other values. The formula for the PERT estimate is:

$$\text{PERT Estimate} = \frac{O + 4M + P}{6}$$

For example, suppose Activity A generated the following estimates:

$$\text{Optimistic} = 2 \text{ Days}$$

$$\text{Most Likely} = 4 \text{ Days}$$

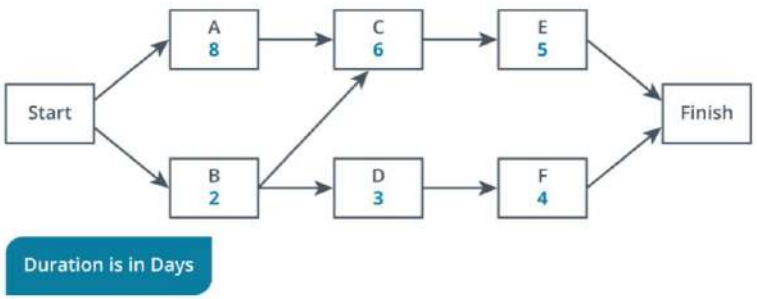
$$\text{Pessimistic} = 12 \text{ Days}$$

$$\text{PERT Estimate} = \frac{2+(4\times4)+12}{6}$$

$$\text{PERT Estimate} = \frac{30}{6}$$

$$\text{PERT Estimate} = 5 \text{ Days}$$

The three-point estimation and PERT estimation processes are completed for every activity on the diagram. A PERT estimate is helpful for calculating realistic timing estimates. It represents a typical schedule with a slight adjustment for worst-case and best-case scenarios. A PERT chart is often created in anticipation of completing critical path analysis.



Critical Path Method (CPM)

A project network diagram that uses the critical path method (CPM) incorporates duration, activities, dependencies, PERT estimates, and additional calculations. CPM analyzes a project schedule to identify the dependent activity sequence with the longest duration. This sequence represents the shortest total project duration, known as the critical path. For this reason, CPM is also called "critical path analysis." Critical path analysis is practical when a project has a fixed timeline.

To calculate the critical path, identify the longest chain of dependent activities. They aren't necessarily the most important activities from a value delivery perspective. Instead, they represent the areas where the project schedule is the least flexible. Items not on a critical path can start or end late without affecting the project's timeline.

The CPM analysis process consists of seven steps. The process is introduced here, and the following pages will define each step. The CPM analysis process introduces several terms and calculations. The next several pages will introduce the information alongside an example.

CPM Analysis Process

1. Draw the project activities and dependencies.
2. Calculate the PERT estimate duration for each activity.
3. Complete a forward pass to calculate the early start and early finish.
4. Complete a backward pass to calculate the late start and late finish.
5. Calculate the total float (TF).
6. Calculate the free float (FF).
7. Identify the critical path.

Minimum Viable Product (MVP) is an early version of a product

Cost aggregation technique used to calculate the cost of a whole component by finding the aggregate of the cost of the constituent parts of the whole component

A **contingency reserve** is a calculated buffer of time and cost that covers documented risks

Management reserve is an additional sum of time or money that covers "unknown unknowns"

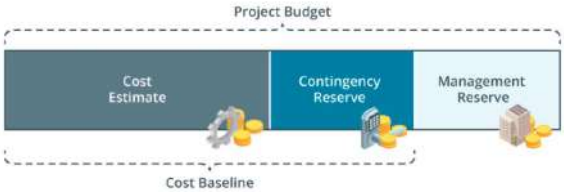
Reserve analysis accommodates multiple methods for calculating contingency reserves, and each has benefits and drawbacks

- **Monte Carlo analysis** is a popular simulation technique
- **Expected Monetary Value (EMV)** requires each risk's probability and impact assessment
- **Decision tree analysis** is a quantitative decision-making tool

A **project baseline** consists of three parts:

- **scope baseline** is the original scope from the detailed scope statement
- **schedule baseline** is the total project duration captured from the project schedule
- **cost baseline** is the expected costs plus the contingency reserves

change control process the process by which the need for change is recorded and approved



Quality Assurance (QA) policies, procedures, and tools designed to ensure defect-free development and delivery

QA Plan Contents

- **Quality Standards:** document any industry or regulatory standards that apply to the project, such as health codes.
- **Quality Objectives:** create a list of measurable quality targets. The project will have multiple quality metrics, such as on-time delivery and the number of defects.
- **Roles and Responsibilities:** identify the people responsible for managing the program, adhering to standards, and evaluating products against the standards.
- **Deliverables and Approach:** identify all deliverables and processes subject to QA activities. Then, list the necessary QC activities, standards, and when they will occur for each deliverable or process.

A **unit test** measures the smallest piece of functionality in software

A **regression test** confirms that the code change did not affect any existing functionality

The **smoke test** runs after the code is compiled into a build and before it is deployed and released

User Acceptance Testing (UAT) evaluates the final outputs against specifications

Performance tests measure the system's performance

Stress testing evaluates how software performs under extreme load

A **test plan** is a reference document that describes how a project will verify that the project's outputs meet the expected quality

Test Plan Elements

- **Scope.** The scope section is a compressed form of the project scope with a few additional details. It should include the project objectives and exclusions, and it lists the user scenarios used in the project test. A user scenario includes the role and expected activity. For example, an administrator user scenario is "an administrator can create a new user account."
- **Types.** The test types section identifies how the team will test the deliverables. It lists the types of tests used in the project, what they will test for, and when they are used.
- **Schedule.** The schedule includes when testing will start and end. It should include dates for each test category, as well.
- **Resources.** The resources section includes the systems and people needed to complete testing and all activities in the test plan. Assign appropriate resources to each test. This section covers both resource needs and roles and responsibilities.
- **Environment.** Software needs a safe space to run, known as a "test environment." A test environment mimics the production environment, but failures in the test environment do not impact the product's end-users. Outline the necessary configurations, hardware, and software needed for a test environment. Note if the environment is already available. If a new environment needs stood up, include setup activities on the project schedule.
- **Tools.** The testing process requires applications for building and running tests. The team also needs to know where to report test results, record defects, and share progress. List all the software requirements under tools and a brief reason for use.
- **Defect Management.** A team will find software defects, and they need to know how to address them. Include clear expectations for defect handling. For example, specify which system to use for recording, when to record a defect, and which information to capture. If the team needs to include specific information, such as which test it was found with or environment details, list them in this section.
- **Risk Management.** Risk management within the test plan summarizes those risks relevant to the testing process. It includes risks likely to occur during testing and any new risks uncovered while executing the test plan. Finally, reinforce why testing is vital to this project by detailing which risks are resolved by this test plan.
- **Exit Parameters.** The exit parameters section explains what the end of testing looks like. It describes the acceptable state of the project deliverables. When the team writes tests, each test should meet these requirements. Therefore, it sets the standard that the QA team measures against and the development team writes against.

verified deliverables project products or results that are completed and verified for their correctness while performing quality control

project management plan a document that details how a project will be executed to achieve the specified objectives

Project Management Plan Contents

- Detailed Scope Statement
- WBS or Backlog
- Resource Plan
- Project Schedule
- Quality Management Plan
- Risk Management Plan
- Communication Plan
- Stakeholder Management Plan
- Project Baselines
- Project Budget
- Project Plan Approval

Critical Path Method Formulas

Term	Name	Formula	Meaning
PERT	Program Evaluation and Review Technique Estimate (PERT Estimate)	PERT Estimate = (O + 4M + P)/6	The most realistic project timing estimate, with adjustments for worst- and best-case scenarios
ES	Early Start (ES)	ES = maximum EF of all direct predecessors	The soonest an activity can start because of how long it will take to complete predecessor activities
EF	Early Finish (EF)	EF = ES + t *where t = activity duration [i.e., time]*	The earliest an activity can finish because of how long it takes to complete it
LS	Late Start (LS)	LS = LF - t *where t = activity duration*	The latest an activity can start and still allow the project to finish on time
LF	Late Finish (LF)	LF = minimum LS of all successors	The latest an activity can finish and still allow the project to finish on time
TF	Total Float (TF)	TF = LS - ES TF = LF - EF	The amount of time an activity can wait to start without impacting the total project timeline
FF	Free Float (FF)	FF = minimum ES of all direct successors - EF	The amount of time an activity can delay without encroaching on the early start of any subsequent activities
CP	Critical Path (CP)	Critical Path = List of all activities where (TF = 0) *If TF = 0, add that activity to the Critical Path*	The sequence of events with zero TF that cannot afford any delays at all

Procuring Solutions

Procurement management acquiring the necessary products and services from outside the project team

Procurement Process Steps
PMs are expected to review and provide input on these key elements in the procurement process:

- **Determine needs:** the PM defines the procurement requirements needed to meet the project's objective.
- **Submit a purchase requisition:** the PM formally requests help from the procurement team to acquire resources. The managers who would provide the budget for the requisition review the request for approval. Rejected requests are sent back to the PM, who is free to adjust the requisition and resubmit it when appropriate. Approved requisitions move to the next step.
- **Complete the solicitation process:** the procurement specialist prepares a request to solicit quotes, bids, or proposals from vendors. First, the specialist identifies vendors who could fulfill the project's requirements. Then, they contact the vendors and invite them to respond to the request with their strategy and prices.
- **Evaluate and select vendors:** an evaluation committee will review and score the vendor proposals on predetermined, objective criteria. All vendors are scored against the same criteria, and committee members do not discuss scoring with anybody, including other committee members. Finally, the procurement specialist finalizes contract terms and executes the contracts.
- **Manage orders:** the procurement specialist or PM generates purchase orders (POs), and the vendor begins work in response. The vendor delivers the work requested in the PO and within the timeline and conditions outlined in the contract. The PM inspects incoming work and monitors vendor performance work. They also work with the vendor to resolve problems.
- **Manage records and payments:** the accounting, purchasing, and procurement teams review order records. They identify discrepancies between the orders, payments, and deliveries. They also maintain the records for auditing and accounting purposes and pay for work completed.

A **purchase order (PO)** is a customer's request for specific goods or services

A **procurement contract** is a mutually binding agreement that details the obligations of the buyer and vendor

procurement documents documents submitted to prospective vendors or service providers to solicit their proposals for the work needed

Request For Information (RFI) a solicitation document used when a business knows the solution it needs but requires information on how vendors will provide the solution

Request For Proposal (RFP) a formal business document that announces a project, describes it, and solicits bids from qualified contractors to complete it

Request For Bid (RFB) commonly used when deliverables are commodities with clear specifications and when price will be the primary determining factor

Request For Quote (RFQ) a formal solicitation for goods or services in which a company invites vendors to submit price quotes and bid on the job

procurement SOW a formal document that outlines the work a vendor needs to deliver

There are three types of SOWs:

- A **design SOW** is process prescriptive. It outlines how to deliver the work, and it sets resource and performance requirements at each stage. Companies with detailed design specifications would create a design SOW.
- A **functional SOW** is outcome focused. It lists the requirements of a finished product only. How to deliver the work is left to the vendor, which can propose alternative approaches or scope adjustments.
- A **performance SOW** describes the expected outcomes along with minimum performance standards. It is more fixed than a functional SOW, but it is still outcome focused and flexible for vendors.

Terms Of Reference (TOR) a formal document that defines a body of work's purpose, structure, boundaries, and expectations

A TOR is a formal document that creates a shared understanding of a group's scope of work or expectations. It often defines how a group will work together and may include strategies to manage issues and risks. For example, you can use a TOR for defining a project or project governance team. You can even use it to define a recurring book club meeting. In procurement, a TOR is included with an RFP to define the expectations of the selected vendor. It covers the project's purpose and scope and defines the technical requirements the vendor would need to meet. A TOR is created early in the project and refined until it goes out with the RFP.

Competitive analysis a vendor analysis process where the organization evaluates competitors within the same market

predetermined client the model user or preferred configuration of a software product

Cost-Benefit Analysis (CBA) compares the cost of a project against the expected financial benefits it will deliver

A CBA needs to include comprehensive data. Costs should include direct costs, such as consulting fees, vendor expenses, and capital costs. It also includes indirect costs, such as training, costs incurred downstream and upstream, and changes in overhead costs. The monetary value of benefits needs to be thorough as well. For example, consider the financial gains that could come from streamlining workflows, increasing outputs, and reducing labor. Include these potential savings in the benefits total.

CBA is used to compare options. It works well when the details vary because it turns everything into a simple sum. You can compare projects against each other to decide which one you should work on. For example, if Project A would generate $50,000 in benefits and Project B would generate $5,000, you can invest in Project A to achieve the most benefit. You can also use CBA to compare vendors against each other to decide which vendor will deliver the best value for your project.

vendor viability a vendor that has a viable and in-demand product and the financial means to remain in business on an ongoing basis

Fixed Price Contracts

fixed-price contract agrees to a set price for a fixed scope of work and places most of the contract risk on the vendor

firm fixed-price (FFP) the price for products or services is set at the outset and not subject to change unless the scope of work changes

fixed-price incentive fee (FPIF) is flexible in that it allows for deviation from performance - an incentive is an additional payment for meeting specific targets

fixed-price with economic price adjustment (FP-EPA) contract that factors in inflation costs

Cost-Plus Contracts

cost-plus contract pays for all resource costs needed to complete the work plus a fee to the vendor

cost-plus fixed fee (CPFF) the vendor receives a fixed fee payment based on the initial estimated project costs

cost-plus incentive fee (CPIF) the vendor receives a predetermined target fee and an incentive fee

cost-plus award fee (CPAF) pays costs plus a fee adjusted for performance

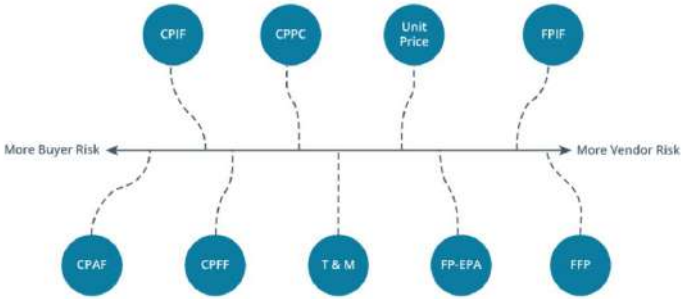
cost-plus percentage of cost (CPPC) the vendor receives a fee equal to a percentage of all costs

Hybrid Contracts

unit price contract ensures that the vendor is paid a fixed price per unit of work delivered

term contract engages the vendor to deliver a set amount of service

time and materials (T&M) contract the buyer pays the vendor a negotiated hourly rate and full reimbursement for materials used to complete the project



warranty is the vendor's commitment to resolving defects

vendor management policies and procedures to identify vulnerabilities and ensure security of the supply chain

rules of engagement working agreements that set the operating norms for a project team

Managing Project Execution

project momentum energy that keeps the project focused and moving forward at an appropriate pace

Requirements Traceability Matrix (RTM) a document that is created by associating the project's deliverables with the requirements for creating each deliverable

version control they track and manage code changes by automatically tracking what changed, when, and by whom

An **issue log** tracks and manages issues

Common issue log fields include the following:

- Issue Number
- Description
- Priority
- Impact
- Raised By (the person who reported the issue)
- Owner (the person who will fix the issue)
- Status
- Date Found
- Date Closed
- Comments

defect log a document containing a list of defects for a project or product

Common Defect Log fields include the following:

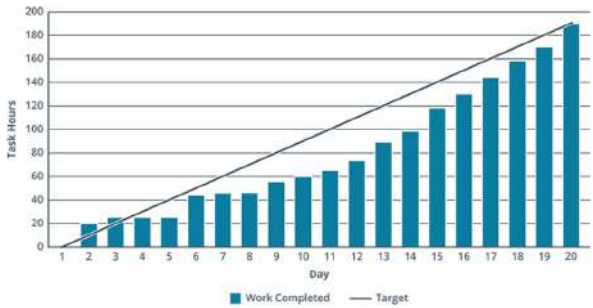
- Defect Number
- Description
- Failed Test(s)
- How to Reproduce
- Priority
- Impact
- Found By
- Owner (the person who will resolve the defect)
- Status
- Date Found
- Date Fixed
- Comments

change log a document used to maintain a list of formal project change requests and their status

Some of the more common change log fields are the following:

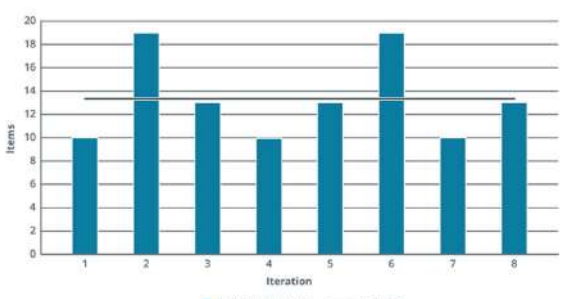
- Change Number
- Description
- Request Type
- Priority
- Impact
- Requested By
- Owner
- Status
- Date Requested or Identified
- Date Completed
- Comments

A **burnup chart** measures the total amount of work completed each day

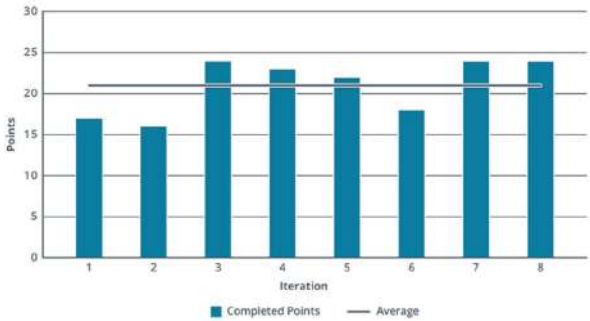


The **burndown chart** measures how much work is remaining each day

A **throughput chart** measures the number of items completed per iteration



A **velocity chart** measures how many story points are completed per iteration



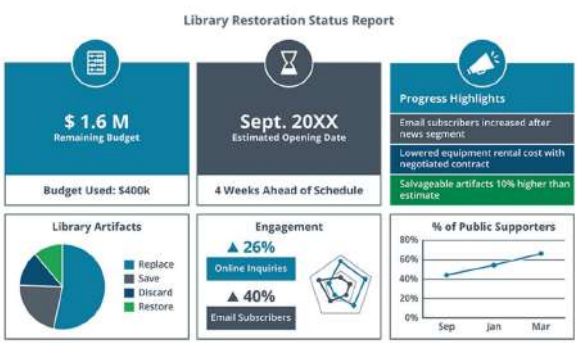
backlog prioritization the process of rearranging a backlog's contents to ensure that the most valuable work is at the top of the list

Stakeholder Meetings

The agenda for an external stakeholder meeting should include the following:

- High-level project status
- Project roadmap or upcoming releases
- Major milestones and deadlines
- Next steps
- Feedback solicitation

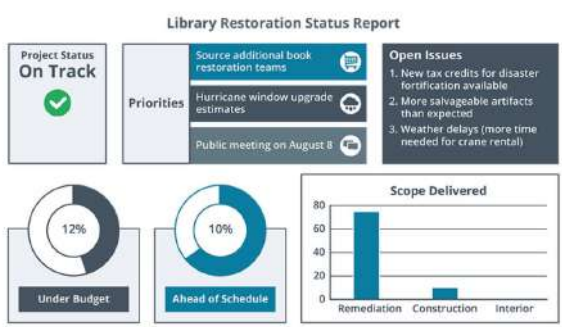
An external project status report contains summarized information. It will look more like an advertisement than other status reports; for instance, it might show recent successes, summarized benefits, or a road map of upcoming features.



The internal stakeholder meeting agenda would include items such as the following:

- High-level project status
- Critical roadblocks, risks, and issues
- Major milestones and deadlines
- Scope, time, and cost performance
- Next steps
- Feedback solicitation

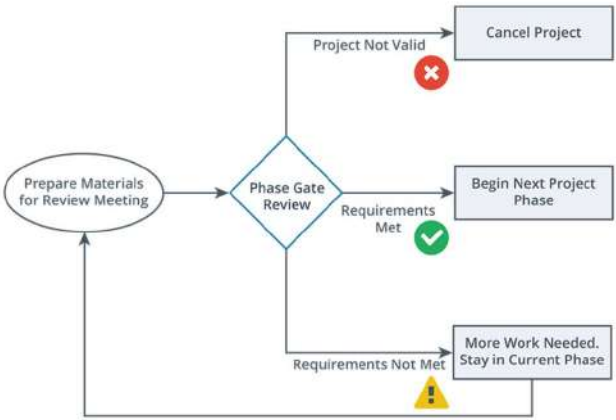
An internal project status report is a proper dashboard compared to the external and team status reports. It shows summarized information that stakeholders should use to make decisions. For example, it might list the top risks, and stakeholders should review them and weigh in on the mitigation strategy or prioritization level. A project behind schedule and over budget should prompt the stakeholders to discuss scope changes.



phase gate review a checkpoint review of deliverables and performance at the end of each phase - a management review or sign-off may be required

Phase gate meetings end with one of three decisions:

1. All phase criteria are met, and the project can move to the next phase (or close in the final stage).
2. The project hasn't met all phase criteria and needs to stay in the current phase.
3. The project is no longer viable and is canceled.



Change Control Board (CCB) a formally chartered group responsible for reviewing, evaluating, approving, delaying, or rejecting changes to the project

ad hoc report progress summaries that are generated as needed

Feedback Process

A feedback loop is a process cycle where you inspect and improve a system.

1. **Gather Feedback.** Gather work performance information from multiple objective and subjective sources.
2. **Analyze Feedback.** Review and summarize the information, and clarify details whenever needed. Develop proposals for expectations.
3. **Act on Feedback.** Share the information with the affected parties. Identify what will change, if anything. Set a change plan.
4. **Follow Up.** Watch to see how the feedback and changes affect the process. Verify if the change had the intended affect. Then, follow up with the individuals to praise successes and review setbacks.

Feedback Sources

- **Performance Metrics.** You will generate project performance reports, and you can glean helpful feedback. Analyze the data to identify specific successes and failures.
- **Stakeholder Comments.** Stakeholders will provide feedback throughout the project. You might hear this information in meetings, emails, and informal conversations. Part of your responsibility is ensuring that the team receives this information, though you may need to summarize it.
- **Observation.** You will monitor the project's operations and inevitably see people deviate from a process or improve a system. Look for how each person is contributing to the product, and take note.
- **Teammate Feedback.** Teammates will know how a person is affecting the project. Solicit comments from teammates through formal feedback systems or one-on-one meetings. Observe how teammates interact and look for opportunities to amplify positive feedback. Summarize this information, and share it anonymously.
- **Lessons Learned.** The lessons learned or retrospective sessions generate an extensive amount of feedback. Fortunately, the entire team is present and receives the feedback together. However, you can monitor the feedback and look for trends that support other feedback you've gathered.

Managing Issues and Changes

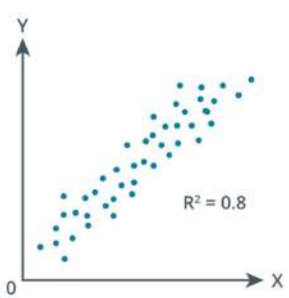
Issue Management Process

1. **Identify the issue.** Find and record issues in an issue log.
2. **Analyze the issue.** Determine the issue's cause and impact and prioritize all issues.
3. **Resolve the issue.** Decide what to do about the issue and do it.
4. **Monitor the issue.** Confirm the issue is resolved and close it.

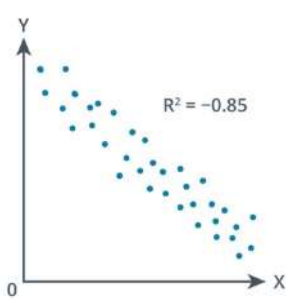
Root cause analysis used to determine the true cause of the problem

Scatter diagram shows the level of correlation between two variables

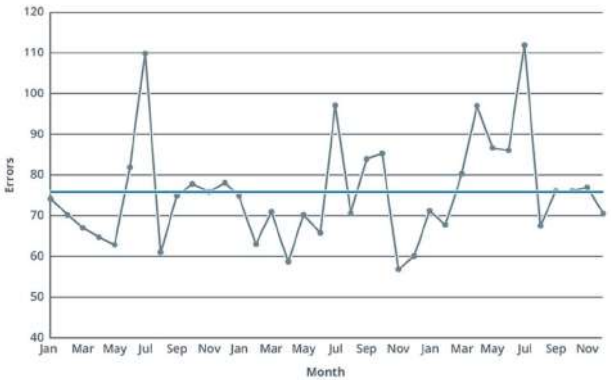
Positive Correlation



Negative Correlation



A run chart is a time-based line graph



Five-Why analysis is a simple root cause analysis tool that looks beyond the surface issue to uncover hidden root causes

repeat the process until you reach what you believe is a solvable root cause. It's called "5-Why analysis" for simplicity, but you can ask "why" as many times as you need to.

Define the Problem: Deliverables are turned in late.

1. Why is this problem happening? *We underestimated the time requirements for the tasks.*
2. Why is that? *We underestimated the complexity of the work.*
3. Why? *We didn't understand how large the deliverables were.*
4. Why? *We didn't break the project backlog down as a team.*
5. Why not? *We couldn't find a time when all the team members were available.*
6. Why is that? *The team members were still working on their previous projects.*
7. Why is that? *The PMO resource management process reassigns team members to new projects before a project is fully closed.*

Issue Prioritization

The prioritization factors often use a descriptive scale to assist with the initial evaluation. However, each descriptor is paired with a numerical value, which is the actual score for each factor. Scores are usually arranged on a 1–9 scale, where 9 is the highest or most critical.

It's a common practice to spread the scores out so that high-priority issues are more separated from lower-priority issues. For instance, a 1-3-9 scale is ideal for scoring three options, such as low, medium, and high. It forces decisions by pushing high-priority items to the top. For example, if you measured low, medium, and high using 1-2-3, items would score too close together, and most items would have a relatively medium priority. However, with a 1-3-9 scale, each item is defined as either truly high priority or relatively low priority. The table below shows an example of a spread scale.

Numeric Scale	Impact on Project	Impact on Organization	Issue Escalation
9	High	High	High
3	Medium	Medium	Medium
1	Low	Low	Low

You can create the same effect with four values by using a 1-3-7-9 scale. This scale has the same effect; 1 and 3 are close to the bottom, and 7 and 9 are close to the top. You can see the logic by comparing the score to the descriptor. Recall that the top two severity measures, critical and major, require unscheduled patches. Meanwhile, the two lowest descriptors, minor and low, can wait for a regularly scheduled release. Therefore, you want critical and major to stand out as high-priority items.

Numeric Scale	Severity	Urgency
9	Critical	Critical
7	Major	Major
3	Minor	Medium
1	Low	Minor

The issue prioritization process is relatively straightforward. Often, you will store the issue priority directly in the issue log.

1. Identify the factors relevant to your prioritization process. You should use at least two factors and as many as needed to drive sound decisions.
2. Add each factor as a new column in your issue log.
3. When issues arise, simply score them on all listed factors. It's important to prioritize the problems against the same criteria every time. For instance, if you measure severity, urgency, and issue escalation, every issue must be scored on all three items. Consistent ratings allow you to make like-for-like comparisons.
4. Calculate the relative issue prioritization score. The simplest scoring method sums the scores of each factor to create a priority score. For example, in the table below, you can see that issue IS-001 has a priority score of 13, which is the sum of its urgency, severity, and issue escalation scores.

The prioritization score shows how an issue's priority compares to other issues on the list. The issues with the highest number are the highest priority. You will have multiple issues with the same priority score because issues are measured against the factors, not against each other. The numerical value of the score isn't relevant. The score only tells you which issues need to be resolved now and which ones can wait until later.

	Prioritization Factor			
Issue Number	Urgency	Severity	Issue Escalation	Priority
IS-001	1	3	9	13
IS-004	3	0	3	6
IS-005	0	1	1	2

issue resolution plan a strategy for minimizing an issue's impact and reducing or eliminating its recurrence

control charts are statistical charts that track the variation in a process

- **Common cause variation** is what you expect to see
- **Special cause variation** is what you want to detect

A control chart looks like a run chart with additional data. It is also a time-based line graph. It uses all the data you would use in a run chart and more. Time or sequence is plotted on the horizontal axis, and the corresponding output is measured on the vertical axis. However, a control chart contains three additional lines:

1. The center line (CL) is the mean of the actual values graphed.
2. The upper control limit (UCL) shows the highest expected value for the process, assuming everything is under control.
3. The lower control limit (LCL) shows the lowest expected value for the process, assuming everything is under control.

A process can still produce values above the UCL or below the LCL. However, 99% of the results should occur within the control limits. If a process returns a value outside the control limits, you know something unusual happened.

change control the process by which the need for change is recorded and approved

change request requests to modify a project's activities or resources

change request form a simple document used to submit a new request for a change to a project

change log a document used to maintain a list of formal project change requests and their status

Software change control the process by which the need for a software change is recorded and approved

- **Technical Approval:** subject matter expert review and input. Source code and configuration changes can place entire systems at risk. As a result, technical staff familiar with the systems must review and approve the changes. This peer-review process can occur through technical review, code review, or pair programming.
- **Product Owner Approval:** confirm product meets all requirements. Software products often have several owners, including a business-oriented owner representing the end-users and a technology-oriented owner who understands the technical aspects. The owners will review and approve changes either in place of a CCB or alongside them.
- **Architectural Approval:** confirm product positively affects the system. The organization's architects design the information flows in an organization and will need to approve any changes. Software is integral to business operations. The company's information flows through multiple software systems, and the software can create interrelated dependencies over time. Additionally, organizations use the data in software to drive business intelligence via reporting. Therefore, any piece of software in your architecture could be a source of truth for another system. For example, a data warehouse could be pulling the data and using it to generate multiple reports. Therefore, any software change could have a ripple effect on the organization. It's essential to evaluate these impacts before changing the system.

IT Infrastructure Library (ITIL) classifies changes into three categories

- **Standard changes** are recurring, routine changes with set policies and procedures. These are the tasks that don't require additional approval or communication. For example, updating the contact list on a website and replacing a printer are standard changes. You would not need to notify customers of these changes, and as a customer, you receive these changes regularly without notification.

- **Normal changes** are modifications that do not have a set process and do not happen often. However, they are also not emergency changes. These are planned in advance and often require customer notifications to varying degrees. You can opt to use advance notice notifications and post-release notes for changes in this category, though it isn't always warranted. For example, if you are releasing subtle graphics enhancements to a website, you wouldn't need to provide a warning. However, you could mention it in the release notes if warranted.

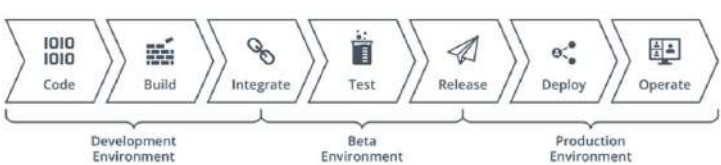
- **Emergency changes** are unplanned, recovery events. You rarely have time to give advance notice, but you can notify customers during and after the event. Emergency changes might warrant using different communication channels. For example, suppose an IaaS provider usually provides release notes and maintenance notices on its customer dashboard, but their site is down. In this situation, the provider might decide to send notifications via email or SMS. It would also use the same channels to inform customers that the site is operational afterward.

Tiered Software Architecture

1. The software is developed in the **development environment**; it is available locally to a single user. This isolated environment allows the developer to change, write, and adjust code. Many engineers can work in separate development environments at once. However, environments consume storage space, so the number of environments is usually limited by the IT infrastructure. Cloud computing models allow for more of these.

2. The beta stage is a **test environment**. When satisfied with the changes, the developer will push code from their development environment to the beta environment., which is a copy of the live product that is not connected to live data. As a result, any issues in this environment will not affect end-users. Multiple developers can push code from their development environment into the beta environment at the same time. The team uses this environment to test the release and resolve issues. Teams also demonstrate finished work in this environment.

3. When the release works in beta, it's ready to be deployed to the **production environment**. The production stage is the live environment that end-users can see. When you open your favorite website or app, you are looking at its production environment. The team will test the product again in this stage. If all goes well, the new software release is available. If the release fails, the team will deploy the rollback plan and undo the release. The software changes are still in the beta and development environments after a rollback, so the team can continue to troubleshoot without affecting the end-user.



budget burndown chart a time series chart that plots how much of the project's budget has been used and how much remains

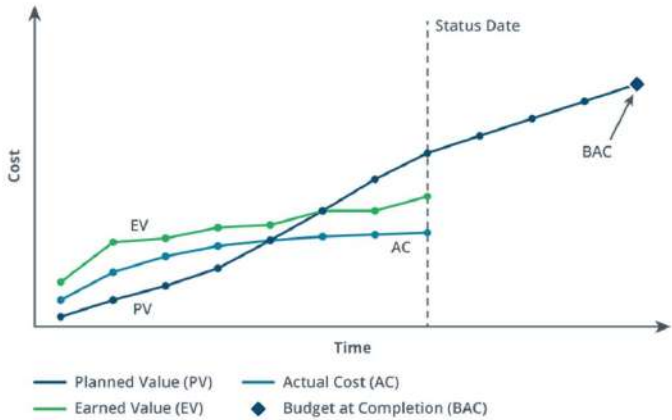
burn rate the rate at which you are using up your budget

budget forcast the estimated budget required to complete the remaining project work

causes of variance the sources or reasons for deviations from the expected standard in a process or item

Earned Value Management (EVM) is another method for monitoring a project's performance

- **Planned Value (PV)** is the target cost and schedule
- **Earned Value (EV)** is the actual schedule
- **Actual Cost (AC)** is the actual cost



Budget at Completion (BAC) is the total budgeted cost of the project at completion

Planned value (PV) is the cost allocated to a specific amount of work. For example, PV estimates that 10% of the project's work should use 10% of the project budget.

$$PV = (\text{Planned \% Complete}) \times (BAC)$$

To calculate the percent complete, you must define how you will measure progress for the project. For example, a project that requires steady resource requirements each week could use time as a measure. In this scenario, if a project is ten months long, every month would equate to 10% completion. An agile team could use story points, and any project could use activities or activity duration as a unit of measure. The PM should define the value measure within the project management plan.

Earned value (EV) is how much of the project's work was delivered within a specific period. It is expressed as a percentage of the project's cost. It represents how much of the project's scheduled work is actually completed. For example, suppose you are halfway through a project, meaning 50% of the project timeline has passed. However, if you've completed 60% of the deliverables, the EV is 60% because you have delivered more work than expected.

$$EV = (\text{Actual \% Complete}) \times (BAC)$$

Actual cost (AC) equals the actual project cost up to a particular time. There's no separate formula to calculate AC within EVM. Instead, the project costs accrue as you update the project's progress within your project management tool. You may also have access to organizational reports that provide additional cost information.

Cost Variance (CV) is the difference between the EV and the AC incurred to complete that work

$$CV = EV - AC$$

- **When CV > $0:** The project is spending less money than expected.
- **When CV = $0:** The project is spending exactly as much as expected.
- **When CV < $0:** The project is spending more money than expected.

Here is a simple example:

$$EV = 100$$
$$AC = 50$$
$$CV = 100 - 50 = 50$$

Conclusion: The project has spent $50 less than expected so far. The project is on track to finish under budget.

Cost Performance Index (CPI) determines whether the project is over or under budget

$$CPI = \frac{EV}{AC}$$

- **When CPI > 1:** The project is spending less money than expected.
- **When CPI = 1:** The project is spending exactly as much as expected.
- **When CPI < 1:** The project is spending more money than expected.

Here is a simple CPI example:

$$EV = 100$$
$$AC = 50$$
$$CPI = \frac{100}{50}$$
$$CPI = 2$$

Conclusion: For every $1 the project spends, it receives $2 in value. The project is well under budget.

Schedule Variance (SV) measures the difference between the actual completion of an activity and the planned or scheduled completion of an activity

$$SV = EV - PV$$

- **When SV > $0:** The project is ahead of schedule.
- **When SV = $0:** The project is progressing exactly as scheduled.
- **When SV < $0:** The project is behind schedule.

Here is a simple example:

$$EV=100$$
$$PV=200$$
$$SV=100-200$$
$$SV=-100$$

Conclusion: The project is far behind schedule. So far, it has failed to deliver $100 worth of its deliverables.

Schedule Performance Index (SPI) the ratio of work performed to work scheduled

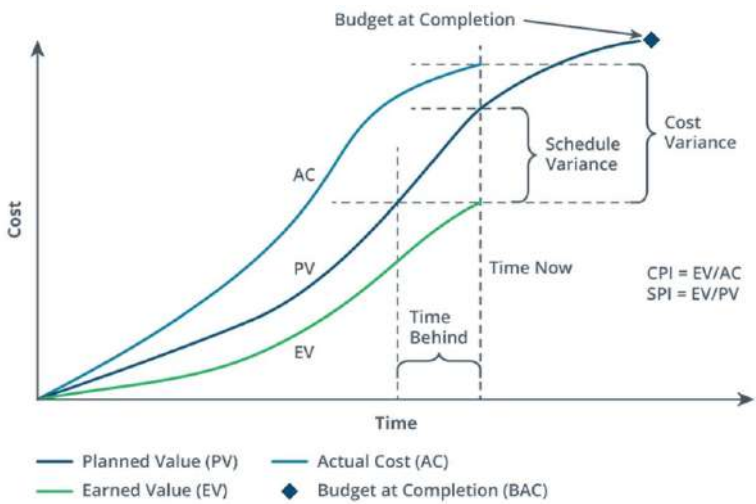
$$SPI = \frac{EV}{PV}$$

- **When SPI > 1:** The project is ahead of schedule.
- **When SPI = 1:** The project is progressing exactly as scheduled.
- **When SPI < 1:** The project is behind schedule.

Let's look at a simple example:

$$EV = 100$$
$$PV = 200$$
$$SPI = \frac{100}{200}$$
$$SPI = 0.5$$

Conclusion: The project team works two days to complete one day's worth of work. The project is far behind schedule.



Estimate at Conclusion (EAC) a forecast of total costs needed to complete the project, given the project's previous performance

$$EAC = AC + ETC$$

Estimate to Complete (ETC) equals the cost needed to finish the remaining work

There are four standard methods for calculating ETC. Each method uses different inputs and provides a different result. The methods are suitable for specific use cases, and the PM must use their expert judgment to select a method for their projects.

1. **Bottom-Up.** You can recalculate the bottom-up estimate for all remaining work.

This method repeats the process you used for estimating the project baseline, but you would only estimate incomplete activities and remaining risks. This method is more time consuming than the other approaches. On the other hand, it could provide the most detailed estimate. You would use this method when the previous estimates are wrong or the project dramatically changes.

$$EAC = AC + ETC$$
where ETC = sum of all remaining activities

2. **Budgeted Rate.** ETC at the budgeted rate uses the previously established baselines to calculate the updated estimate.

This method assumes that the project will experience no additional variances. You would use this method if the variance source were eliminated.

$$EAC = AC + (BAC - EV)$$
because $ETC = BAC - EV$

For example:

$$AC = 100$$
$$EV = 50$$
$$BAC = 200$$
$$EAC = 100 + 200 - 50 = 250$$

Conclusion: If the project experiences no additional variance, it will finish at a cost of $250, or $50 over budget.

3. **CPI.** You can use the CPI to modify the BAC based on previous cost performance.

This method assumes that the project will continue to experience a similar variance in cost. This method works well if you expect the project to have minimal SV but ongoing CV.

$$EAC = \frac{BAC}{CPI}$$
because it applies the current CPI to the entire project

For example:

Given that:
$$BAC = 200$$
$$CPI = 0.5$$
Then:
$$EAC = \frac{200}{0.5}$$
$$EAC = 400$$

Conclusion: If the current performance continues, the project will cost $400, or $200 more than budgeted. The cost performance is likely to double the project's budget.

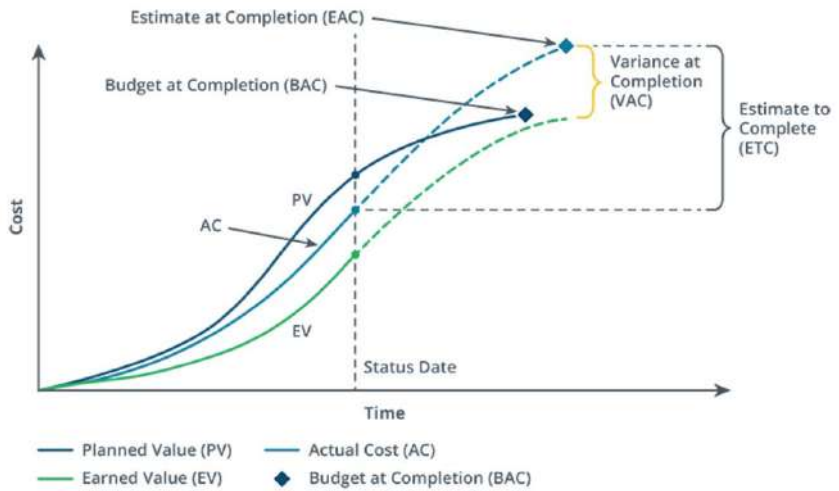
Variance at Completion (VAC) measures the total variance the project is expected to experience

VAC = BAC - EAC
where EAC = any method described in this course

For example:

BAC = 200
EAC = 400
VAC = 200 - 400
VAC = -200

Conclusion: Assuming the updated EAC is accurate, the project will cost $200 more than budgeted. The project will overrun the budget if we don't improve.



rebaseline an adjustment to project baselines to remove variances that aren't related to a change request

revised baseline an adjustment to project baselines resulting from project change request

Examples
Rebaseline: The project is behind by two weeks and over budget by $4,000. Upon investigation, we learned that the development team hadn't factored in lower productivity while learning how to use the new system. Therefore, we would like to rebaseline the project to reflect the increased time, cost, and scope.

Revised baseline: The project added another webpage to the scope; therefore, the baselines increased by $1,000 and five days.

Both situations lead to the same outcome: adjusted baselines that better reflect the reality of the project. However, revised baselines are a proactive, forward-looking approach. Rebaselines are reactive.

Project Dashboards

Executive Project Dashboard
An executive-level dashboard will have extremely compressed information. An executive dashboard may be reviewed monthly or less, so it doesn't need day-to-day information.

Dashboards provide quick answers to critical questions. The person using the dashboard should be able to quickly review the contents and decide if things are running as expected. If not, they notice right away and research the issue further. For example, the executive dashboard group paid for the project because they thought it would benefit the organization. Therefore, the dashboard should convey whether the project is returning value and performing as expected.

An executive dashboard could answer one or more of the following questions:

- Why does this project matter?

The dashboard should display a relevant business value measure, such as Revenue Generated.

- Is this project on time, on budget, and in scope?

The dashboard will have simple graphics showing schedule, budget, and scope snapshots.

- Was this project a worthwhile investment?

The dashboard might display ROI.

- Will this project put the organization at risk?

It could display high-impact risks or issues.

Team Project Dashboard
A team-level dashboard is tactical and more detailed than an executive dashboard, but the team still uses it to make decisions. A team's dashboard centers around the work.

A team dashboard could answer one or more of the following questions:

- What do we need to do next?

It should display a list of upcoming tasks.

- Is anybody overutilized or underutilized?

It could show resource utilization for the week or iteration.

- What are our deadlines?

It might list activity deadlines and milestones.

- Are we on track to finish on time?

It could summarize the schedule deficit or surplus.

Wrapping Up the Project

Organizational Change Management (OCM) is a structured approach to building change plans

change adoption is a performance metric measuring how many people use a new offering

Knowledge Base (KB) searchable database of FAQs, advice, and known troubleshooting issues

Validate all Deliverables

All stakeholders need to validate that the project has met all deliverables. Obtain documented approval from each identified stakeholder and include it within your project files. A stakeholder's approval is their confirmation that they agree that the project has met the deliverables. This process also signals to stakeholders that you are not accepting new change requests and that the project team will not deliver more work.

By approving the deliverables, the stakeholder agrees to several conditions:

- The project deliverables are completed to satisfaction. For canceled and suspended projects, the deliverables are completed as described.
- You are not accepting new change requests.
- The project team is not delivering new work.
- The project is approved to close. Note that this step does not close the project. Instead, the stakeholder agrees that the PM should begin closing the project.

budget reconciliation verification that each financial transaction in the project budget is matched with a corresponding payment or activity

administrative closure the project management process of verifying and documenting project results to formalize project or phase completion

closure meeting is a final send-off of the project and its last working meeting

Agenda

Project Overview: List the goals, objectives, and success criteria. Describe if the team completed the project and to what extent. If a project is canceled or suspended, outline how much of the project was completed.

Project Highlights: Outline the project team's significant accomplishments. Use this section to share the team's successes beyond completing the deliverables. For example, share how they improved processes and which delivery methods worked well for them.

Project Challenges: Challenges are an excellent learning opportunity for your project team and other teams who may face similar issues. Therefore, transparency benefits everybody. Identify the project's challenges and how the team met them. For example, describe process issues, technical issues, and underperforming metrics. Share any takeaways that other PMs should consider.

Future Considerations: A project generally initiates a new product, which is handed over to an operations team for maintenance. Usually, the project team identifies other enhancement ideas that could be implemented but were not part of the project scope. Use this section to share some of the future enhancement ideas; these could turn into future projects. You can also list any open action items or follow-up work required, along with the owners and due dates.

Lessons Learned Workshop: The lessons learned section is the largest agenda item and will require about half of the scheduled meeting time. Begin by summarizing the lessons learned throughout the project, which the team gathered at every retrospective or phase gate. Next, hold an end-of-project lessons learned session in this meeting. This project closure activity is also called a "post-mortem," "project retrospective," or "project review." It aims to identify ways to improve future projects.

project closeout report a final summary of the project's performance, outcomes, and lessons learned

project sign-off documented communication from project stakeholders that they agree to close the project

Closeout Report Contents

- **Project Overview:** List the project's purpose, goals, objectives, and success criteria.
- **Project Highlights:** Summarize the team's major accomplishments and successes.
- **Performance Analysis:** Summarize the project's scope, schedule, and budget performance.
- **Project Challenges:** List challenges, issues, and their impact.
- **Future Considerations:** Name one or two ideas for potential future projects.
- **Lessons Learned Summary:** Summarize the lessons learned from every phase, retrospective, and the project closure meeting. This lessons learned report is your best resource for improving future projects.
- **Financial Summary:** Summarize the project's final performance against the cost baseline, and summarize reserves usage. You can also attach a copy of the final reconciled budget to the report.
- **Transition Summary:** Summarize how the project affected ongoing operations and who the operation owners are.
- **Closure Approval Request:** Formally request project closure within the report. When the PM, project sponsor, and stakeholders sign off, the project is formally closed.

reward and recognition system a formal system used to reinforce behaviors or performance