A.A's Note

(Please read first, Exam Takers!)

This guide might seem comprehensive and even a little overwhelming at first, but don't worry, **there's no need to memorize.** Focus on understanding the concepts and the logic behind PMI's mindset. Sometimes, you'll come across new terms on the exam that aren't even in Study Hall, so understanding how everything connects will help you handle any question with confidence.

If you're completing your 35 PDUs, take time to really **understand the flow of the knowledge areas and the 49 processes (take notes).** Once you grasp that, it becomes easier to see where each question fits within the project process during the exam.

Please remember, just understand and do your best. **Consistency matters.** I've noticed that when I take time off, like a vacation, it feels like starting over when I return. So if you plan to be away, refocus right after.

If you're studying for 1–2 months (or more of course), a weekday schedule (Monday–Friday) works well, around 20–25 hours a week. I usually study after work and reserve Saturday mornings (8 am–12 pm) for mock exams to simulate the real PMP test setting. During the real exam, they provide noise-cancelling headphones, which help create a focused, distraction-free environment. Try to simulate that feeling during your practice mocks as well.

Book your exam ahead for accountability. You can reschedule for free if it's more than **30 days out**, but less than that costs **$70 USD**. If it's **less than 48 hours**, you can't reschedule at all, so plan carefully!

Sometimes I read how others finish their prep in just a few weeks, and honestly, wow, I'm always impressed. This r/PMP Reddit group has been such a supportive space; it really helped me feel more confident.

Mindset is everything. Once you start thinking "I'm not ready," you'll believe it, but the truth is, **no one ever feels completely ready**. Be **graceful and kind to yourself**, because this isn't easy. Even with Study Hall, I personally found the real exam felt different at times, and some questions just didn't make sense.

All of this is just my **personal experience**; everyone's journey is different. Some people get easy questions, some don't make sense. Also, **retaking the exam is costly and a pain**, so do your best and stay committed. I'll keep **updating this guide as I go** and will mark anything that actually **showed up on my exam** (from what I remember). If you see **green**, it indicates that it appeared on my exam.

Check your mistakes, **just understand**, read the scenario, grasp the context, and it will start to click. Once you dedicate time to study, you'll find you can answer questions **even without reading them in full**.

Of course, the **exam is situational**, but understanding **keywords and how they connect** will make the logic just click.

Good luck, stay confident, stay consistent, and trust yourself. You've got this.

-A.A.

REDDIT Gems:
Thank you, users! MuckyDuckyPuppy
PMP GIFT by Ahmed
STEPS FOR ME IN ORDER:

- 35 PDUs
- Application
- Understand Exam Outline: Start with the PMP Exam Content Outline (ECO). Review the three domains — People, Process, and Business Environment. Copy each task (for example, *Manage Conflict* under People) and search it in the Study Hall to find related "Practice Questions". Work through them in order: People, then Process, then Business Environment.
  Take notes or use an Excel sheet to separate each domain, list key points you want to remember, and mark your weak areas. Review those weak spots regularly and redo practice questions to strengthen your understanding.
- **Again, start Study Hall (basic: 2 Mocks)/Plus:** has more Expert Q's Mock 4 and 5 total 5 Mocks), Practice Questions and Exam Q's, then full mock on weekends/whatever's convenient for you.

To strengthen my understanding further, watched the videos below:

- DM 150 PMBOK
- ANDREW 200 ULTRA
- DM 200 Agile Q's
- DM Drag and Drop
- ANDREW 50 MINDSET PRINCIPLES
- RV PMBOK Guide
- Play ITTO's/Drag and Drop's Interactive, Practice Q's (week of exam as refresher)
- ThirdRock Notes Exam prep: a week before the exam
- ThirdRock Notes Cheat Sheet: day before exam for light review
- Before the exam starts, refresher MR video and review my memorized formula to put in the note pad provided

---

EXAM DAY
They gave me a notepad, first thing I wrote down:

- **ART: Assess → Review → Take Action**
- If I see words like **"always, NEVER, immediately, ONLY" → eliminate right away.**
- **Formulas I memorized and wrote down:**
  - SPI (Schedule Performance Index)
  - CPI (Cost Performance Index)
  - Communication Channels Formula
  - PERT, Triangular Distribution *(Only SPI came up — sometimes a few may appear, so still memorize to be prepare! Find info below or Ctrl+F.)*

During the Exam: Approach & Strategy
Approach:

- **Read each question fully** — don't rush.
- **If unsure → flag it and move on**. Keep a steady pace.
- **Eliminate obviously wrong answers first:**
  - I always **strike out two choices immediately**.
  - Then **carefully analyze the remaining two options** using logic and PMI principles.
- **Mindset:** Stay calm, confident, and focused.
- **Review flagged questions** at the end with a clear head and fresh perspective.

Pacing

- Aim for **steady focus** — about **1.2 minutes per question**.
- Use **highlighting** to mark key words. Ask yourself:
  - **Agile, Hybrid, or Predictive**? (learn keywords)
  - Which **process group**: Initiating (IN), Planning (PL), Monitoring & Controlling (M&C), Closing (CLO)?
  - Which **knowledge area**: Scope, Procurement, Cost, etc.?
- **Understand the flow** — no need to memorize everything, focus on reasoning.

Breaks

- **1st Break (7 mins):**
  - Jumping jacks, stretch, washroom, splash cold water, breathe.
  - Keep telling yourself: *"You can do this."*
  - Eat a **banana, fruit, or peanut butter sandwich** for energy.
  - Hydrate lightly; use the washroom before the exam starts.
- **2nd Break (5 mins):**
  - Repeat light stretching, jumping jacks, and breathing exercises.

Self-Talk:

"I'm calm. I've got this. I've done the work."

Let's start!

1. Motivation & Personality
Leadership Styles and Needs:
Theory X/Y/Z -A model explaining why people are motivated at work.
-**Theory X:** Assumes people dislike work and need control, direction, or threats to perform.
-**Theory Y:** Assumes people enjoy work, seek responsibility, and are self-motivated.
-Theory Z-long-term job security
People work harder when they feel capable, see results from their effort, and are rewarded fairly.
-**Herzberg: Motivators (intrinsic):** Meaningful work makes you happy → achievement, growth, recognition
Hygiene (extrinsic): Salary, benefits, and work conditions stop unhappiness, but don't motivate. ("I have to do it")
-**Daniel Pink:** True motivation comes from **Autonomy, Mastery, Purpose**
Money or rewards **don't motivate** for creative or meaningful work
-**MBTI / DISC – Personality Assessment Tools**

Tools used to understand how people think, behave, communicate, and fit into teams or leadership roles.

-**MBTI (Myers-Briggs Type Indicator):**

1. Extraversion (E) vs Introversion (I)

- **Extraversion (E):** Energized by people, activity, and talking things out.
- **Introversion (I):** Energized by alone time, reflection, and thinking before speaking.

2. Sensing (S) vs Intuition (N)

- **Sensing (S):** Focus on facts, details, and the present moment. Practical, realistic, likes step-by-step instructions.
- **Intuition (N):** Focus on patterns, ideas, and possibilities. Imaginative, future-oriented, likes innovation and brainstorming.

3. Thinking (T) vs Feeling (F)

- **Thinking (T):** Makes decisions based on logic, facts, and objective analysis—values fairness and consistency.
- **Feeling (F):** Makes decisions based on values, emotions, and impact on others. Values harmony and empathy.

4. Judging (J) vs Perceiving (P)

- **Judging (J):** Prefers structure, planning, and organized schedules. Likes closure and decisions.
- **Perceiving (P):** Prefers flexibility, spontaneity, and keeping options open. Likes to adapt and explore.

Quick Memory Tip:

- **E/I → energy source**
- **S/N → how you take in info**
- **T/F → how you make decisions**
- **J/P → how you live/organize your life**

MBTI helps understand how a person prefers to work, make decisions, and interact with others.

-DISC:

Categorizes behaviour into **4 types:**

D (Dominance): Results-oriented, assertive, likes control.

I (Influence): Social, persuasive, enjoys collaboration.

S (Steadiness): Cooperative, patient, loyal, supports stability.

C (Conscientiousness): Analytical, detail-oriented, follows rules.

-helps predict how someone behaves in a team or work environment.

Tuckman Ladder (Team Development Stages): Forming → Storming → Norming → Performing → Adjourning

-F (Direct), S (Coach), N (Support), P (Delegate), A (Release) – How teams evolve and improve over time.

-Business Organization Matrix:

Functional

People grouped by specialty (IT, HR, Finance).

PM power is low, mostly coordinates.

Communication is vertical.

PM tip: Follow department rules, coordinate tasks.

Projectized

Team built around a project.

PM power is high—controls budget, schedule, resources.

Communication is horizontal.

PM tip: PM runs the project, makes decisions.

Matrix (Hybrid)

People report to **both the functional manager and PM**.

Weak Matrix: PM coordinates, functional manager decides → PM has low power.

Balanced Matrix: PM and functional manager share authority → PM has power medium.

Strong Matrix: PM controls the project like a projectized → PM has high power.

Organic/Composite: Flexible, can change depending on project → PM power flexible.

Quick tip:

- PM full control → projectized or strong matrix
- PM just coordinates → functional or weak matrix
- Power shared → balanced matrix
- Flexible roles → organic/composite

Conflict Management

-**Confronting / Collaborating:** Solve issues together (win-win).
-**Compromising / Reconciling:** Give & take; moderate approach.
-**Accommodating:** Yield to others' needs.
-**Forcing / Directing:** Assert own solution.
-**Withdrawal / Avoiding:** Low influence; ignore minor conflicts.

Leadership Styles

-**Servant:** Focus on people, purpose, process.
-**Interactional:** Mix styles based on context (transactional, transformational, charismatic).
-**Transformational:** Inspire & motivate change.
-**Laissez-faire/free rein:** Minimal interference; trust the team.
-**Transactional:** Reward & punishment approach.
-**Charismatic:** Lead through personality & vision.
-**Authentic:** Lead with honesty & integrity.
-**Democratic:** Team involved in decisions.
-**Autocratic:** Make decisions alone.
-**Situational:** Adapt style to team/situation.
-**Affiliative:** Emphasize harmony & relationships.
-**Visionary:** Inspire long-term direction.

Project Delivery Approaches

-**Iterative:** Undefined scope, single delivery, continuously refined.
-**Incremental:** Defined scope, small usable deliverables in steps.
-**Continuous Delivery / Continuous Integration (XP / Agile / CI):** includes practices like **TDD**, **pair programming**, **continuous integration**, and **small releases**.

💡 *Goal:* Deliver high-quality software that adapts to change.-**TDD (Test Driven Development):** Write tests before code to reduce defects. **Write test → Write code → Run test → Refactor.**
-**BDD (Behaviour Driven Development):** Business-readable tests ensure correct behaviour. Built by user behaviour.
-**FDD (Feature Driven Development):** A method that delivers software in **small, functional features**, best suited for **large and complex projects** needing clear structure and frequent progress.

Planning & Scope
-Scope Statement: Defines entire scope, exclusions, deliverables, acceptance criteria.
-**Scope Baseline:** Approved scope statement + WBS + WBS dictionary. Approved via CCB procedures.
-**WBS (Work Breakdown Structure):** Break project into manageable tasks.
-**Rolling Wave Planning:** Near-term work detailed; future work high-level.
-**Scope Creep:** Uncontrolled changes without approval.
-**Milestone / Gantt Charts: High-level** (can present executives) vs detailed project timelines.
-**Project Scope vs Product Scope:**
Project scope: All work needed to deliver the result.
Product scope: Features of product or service.
-**Validate Scope:** Acceptance of deliverables by customer or PO.
-**Control Quality:** Performed before Validate Scope; ensures correctness & meets quality requirements.

Scheduling & Dependencies
Four Types of Task Dependencies:
1. FS – Finish to Start
Most common type.
Task B **cannot start** until Task A **finishes**.
Example: You cannot start painting a wall (Task B) until the wall is built (Task A).
2. **SF – Start to Finish**
Less common.
Task B **cannot finish** until Task A **starts**.
Example: A night shift (Task B) cannot end until the day shift (Task A) begins.
3. **FF – Finish to Finish**
Task B **cannot finish** until Task A **finishes**.
Example: Editing a document (Task B) cannot finish until writing the document (Task A) is done.
4. **SS – Start to Start**
Task B **cannot start** until Task A **starts**.
Example: You cannot start installing software (Task B) until the server setup begins (Task A).
-Dependencies show **which tasks rely on others**, helping schedule and avoid delays.

-**Mandatory/Hard:** Must occur in order.
-**Discretionary/Soft:** Preferred; best practice.
-**External:** Outside team control.
-**Internal:** Within project control.
-**Crashing:** Add resources to shorten the schedule; higher cost.
-**Fast-Tracking:** Do tasks in parallel; higher risk.
-**Dropped-baton:** Delay from the task not being ready.
-Student Syndrome / Parkinson's Law: Delay due to procrastination or filling available time.

Estimation & Cost
-Analogous (Top-down): Use historical data/expert opinion.
-**Bottom-up:** Break tasks into detail for accuracy.
Top-down = faster, less detailed; Bottom-up = more accurate, detailed.
-**Parametric:** Statistical relationship with historical/unit cost.
-**Triangular / PERT:** (P+M+O)/3; captures uncertainty.
-**Deterministic:** No uncertainty; low accuracy.
-**Agile Velocity:** Team productivity measurement (story points).
-**Sunk Cost:** Cannot recover; ignore in decision making.
-**Gold Plating:** Extra features without client request; avoid.

Risk & Quality Management
Risk Terms
-Threats – Negative risks that could harm the project:
Avoid – Change plan to eliminate risk.
Accept – Acknowledge risk and do nothing unless it happens.
Mitigate – Reduce the likelihood or impact of risk.
Transfer – Shift risk to another party (e.g., insurance, vendor).
Escalate – Pass responsibility to a higher authority if out of PM control.
-Opportunities – Positive risks that could benefit the project:
Exploit: ensure the opportunity **definitely happens** (e.g., assign best resources to guarantee it).
Enhance: increase the probability or impact of the opportunity, but it's not guaranteed.
Accept – Take advantage if it occurs, but don't actively pursue.
Share – Partner with another party to maximize opportunity.
Escalate – Pass responsibility to a higher authority for handling.
Emerging Risk Terms
-Event Risk – Future risk that is already identified.
-**Non-Event Risk** – Uncertain potential events that might happen.
-**Variability Risk** – Risk arises from unpredictable variations; can use Monte Carlo simulation.
-**Ambiguity Risk** – Risk due to lack of knowledge or unclear information.
-**Emergent Risk** – Risk that appears unexpectedly, often from blind spots or unforeseen changes.

-**Residual Risk:** Remaining risk after mitigation.
-**Secondary Risk:** Arises due to implementing a response.

Types of Risks

1. Technical Risk

- Related to the **project scope, requirements, or technical processes**.
- Example: Missing Definition of Done (DoD), incomplete feature, software defect, or integration challenges.

2. External Risk

- Originates **outside the organization**.
- Example: Changes in legislation, regulatory shifts, competition, environmental factors, or supplier failure.

3. Management Risk

- Related to **project management and organizational processes**.
- Example: Poor resource allocation, ineffective communication, weak leadership, and scheduling issues.

4. Commercial Risk

- Related to **transactions, contracts, or partnerships**.
- Example: Vendor delays, procurement problems, subcontractor failures, fixed-price contract issues.

A **risk register** is a document that records details of all identified individual risks for a project.

-**Identified risks** – What could go wrong or impact the project.
-**Risk owners** – The person responsible for managing each risk.
-**Potential risk responses** – Planned actions to mitigate, avoid, transfer, or accept each risk.

-Risk Attitudes

- **Risk-Averse** → Avoids risk, plays safe
- **Risk-Seeking / Risk-Taker** → Likes risk, goes for big reward
- **Risk-Neutral** → Doesn't care about risk, decides by expected outcome

-RBS (Risk Breakdown Structure): Categorize risks from high to low level.
-**Qualitative Risk (do this first, as quantitative is costly):** Likelihood & impact assessment.
-**Quantitative Risk:** Math-based analysis; costly & time-consuming.
-**Risk Tools: Brainstorming** – Group discussion to identify all possible risks; encourages creative ideas.
Checklists – Predefined lists of common risks to make sure nothing is missed.
Interviews – Talk to experts or stakeholders to uncover risks based on experience or insight.
Decision Trees – A Visual diagram to evaluate decisions, possible outcomes, and their risks.
Monte Carlo Simulation – Uses random scenarios and probability to predict the impact and likelihood of risks.

-**Risk Register:** Document with all identified risks, updates, and tracking.
-**Risk Report:** Overall project risk summary.
-**Issue Log:** Tracks issues from risk events (missed deadlines, system failures, team changes).
-**EVM:** SPI = EV/PV, CPI = EV/AC; measures schedule & cost performance.
-**Control Quality / Manage Quality:** QA audits, testing deliverables, and reducing defects.
-**COQ (Cost of Quality)** measures the total cost to ensure quality and the cost when quality isn't met:
Cost of Conformance (CoC): Costs to prevent defects and ensure quality, e.g., training, audits, testing. Think: "doing it right the first time."
-**Cost of Non-Conformance (CoNC):** Costs when things go wrong, e.g., rework, warranty claims, scrap, customer complaints. Think: "cost of mistakes."
Data Representation Techniques:
-Control Chart: Tracks process performance over time; identifies variations and ensures stability.
-**Affinity Diagram:** Groups ideas or data into related categories; helps organize brainstorming results.
-**Ishikawa / Fishbone Diagram:** Identifies root causes of problems; visually shows cause-and-effect.
-**Flowchart:** Diagram of steps in a process; helps understand workflow or decision paths.
-**Histogram:** Bar chart showing frequency distribution of data; highlights patterns or trends.
-**Matrix Diagram:** Shows relationships between two or more sets of data, e.g., responsibilities vs. tasks.
-**Scatter Diagram:** Plots two variables to identify correlation or patterns.

Resource & Team Management
-RBS (Resource Breakdown Structure): Hierarchy of team & physical resources.
-**Resource Calendar:** Tracks availability.
-**Team Charter:** Social agreement among team members.
-**Maslow's Hierarchy (teams need to perform better):** Rewards and recognition should address different levels of human needs:
(bottom) Physiological: Basic needs like salary or perks.
Safety: Job security, stable environment.
Social: Team inclusion, belonging.
Esteem: Appreciation, acknowledgment, promotions.
(top) **Self-actualization:** Opportunities for growth, creativity, and purpose.
Hybrid Feedback: Combines formal (performance reviews, bonuses) and informal feedback (praise, recognition in team meetings) to motivate and encourage continuous improvement.

-**McClelland's Three Needs Theory:** People are motivated by **three main needs**:
Achievement: Desire to accomplish goals.
Power: Desire to influence or control others.
Affiliation: Desire for social relationships and belonging.
-**Virginia Satir's Change Model:** Shows how **individuals experience and manage change**, including stress, resistance, and adaptation stages.
-**ADKAR Model:** A **5-step framework for change management**:
Awareness – Recognize the need for change.
Desire – Willingness to participate and support change.
Knowledge – Understand how to change.
Ability – Implement required skills and behaviours.
Reinforcement – Sustain the change over time.
-**Expectancy Theory:** People are motivated when they believe their **effort leads to performance**, and performance leads to **desired outcomes/rewards**.
-**Halo Effect:** A **cognitive bias** where your **overall impression** of someone influences how you judge their specific traits.
-**Three Dimensions of Complexity:** Evaluates project complexity based on **structural, technological, and human factors** (scope, technology, stakeholders).
-**Make-or-Buy Decisions:** Decide whether to **produce internally or outsource**, based on cost, resources, and project needs.
-**DB (Design-Build) vs. DBB (Design-Bid-Build):**
Design-Build: Single contract for both design and construction; faster, integrated.
Design-Bid-Build: Separate contracts for design and construction; traditional, more control, but longer timeline.
Conflict Management: Collaborate, Force, Compromise, Accommodate, Withdraw.
-**RACI Matrix:** Responsible, Accountable, Consulted, Informed.
-**High-Performance Teams:** Personal drive, challenge, stimulating environment.
-**T-Shaped Team:** Combination of generalist + specialist.
-**Functional Manager:** Controls specialist availability.

Procurement Management
-**Procurement Management Plan:** Defines how procurement will be planned, executed, and controlled throughout the project. Includes roles, responsibilities, and procedures.
-**Procurement Strategy:** High-level approach for acquiring goods or services, e.g., make vs. buy, selecting vendors, timelines.
-**SOW (Statement of Work):** Detailed description of the work to be done, deliverables, and requirements for the procurement.
-**Bid Documents:** Documents sent to vendors to request proposals or quotes. Includes RFI (Request for Information), RFP (Request for Proposal), and RFQ (Request for Quotation).
-**Conduct Procurement:** Obtain responses from sellers, bidder conferences, and external resources.
Contract Types:

-Fixed Price (FP): Total price agreed upfront; vendor bears cost risk.
FFP (Firm Fixed Price): Price is fixed; no adjustments.
FPIF (Fixed Price Incentive Fee): Fixed price with incentive for meeting/exceeding targets.
FPEPA (Fixed Price with Economic Price Adjustment): Fixed price but allows adjustments for economic factors (e.g., inflation).
-**Cost Reimbursable:** Buyer reimburses actual costs plus fee; useful when scope is uncertain.
CPFF (Cost Plus Fixed Fee): Reimburse costs + fixed fee.
CPIF (Cost Plus Incentive Fee): Reimburse costs + fee based on performance.
CPAF (Cost Plus Award Fee): Reimburse costs + award fee based on satisfaction/performance.
CPF/CPPC (Cost Plus Percentage of Cost): Reimburse costs + fee based on percentage (rare; risky for buyer).
T&M (Time & Material): Pay for time & materials used; used when scope is not fully defined.
-**Bid Document Types:**
RFI (Request for Information): Gather information from potential vendors.
RFP (Request for Proposal): Ask vendors to propose a solution with cost & timeline.
RFQ (Request for Quotation): Ask vendors for a price quote when requirements are clear.
-**Control Procurement:** Monitor contracts & compliance; ensure delivery meets agreements.

Stakeholder Management
-Stakeholder Types: Internal, Connected, External.
-**Stakeholder Register:** Contact info, role, influence, expectations, communication needs, effect on project.
-**Stakeholder Analysis:** Align goals & actions; engagement assessment matrix.
-**Engagement Levels:** Unaware → Resistant → Neutral → Supportive → Leading.
-**Power/Interest, Power/Influence, Impact/Influence:** Models to prioritize stakeholders.
-**Directions of Influence:**
Upward: Influence toward senior management or sponsors.
Downward: Influence toward your team or subordinates.
Outward: Influence toward peers, other teams, or **external stakeholders (government)**
Sideward: Often used interchangeably with outward; influence across the same level/peers of PM
-**Salience Model:** Helps prioritize stakeholders based on three attributes:
Power: Ability to influence the project.
Legitimacy: Stakeholders' rightful involvement in the project.
Urgency: How critical their needs or issues are.
Stakeholder Priority (based on attributes):

-Definitive: Has all three attributes → Manage closely.
-**Dominant:** Power + legitimacy → Keep satisfied.
-**Dangerous:** Power + urgency → Keep informed and monitor.
-**Dormant:** Power only → Minimal effort; just monitor.
-**Discretionary:** Legitimacy only → Monitor lightly; low priority.
-Stakeholder Engagement Plan: Define actions to manage relationships & expectations.

Change Management

- **Kotter's 8 Steps:**
    1. Urgency
    2. Guiding coalition
    3. Vision
    4. Volunteer army
    5. Remove barriers
    6. Short-term wins
    7. Sustain momentum
    8. Institutionalize change

-Change Management: Manage resistance, educate, and engage stakeholders.
-**Servant Leadership:** Focus on people, purpose, process.

Performance & Monitoring
-Work Performance Data → Info → Reports
Data: Raw project facts (dates, costs, hours, defects).
Information: Analyzed data showing insights (variances, trends, earned value).
Reports: Communicated info to stakeholders (status updates, dashboards).
Flow: Collect → Analyze → Share.
Measure, understand, and communicate project performance.
-**SPI / CPI:** Schedule & cost performance indices.
-**CV / SV / EV / PV / AC:** Track cost & schedule health.
-**Point-in-time measurement:** Snapshot of project status.
-**Run Chart / Bubble Chart / Scatter Diagram / Flowchart:** Visualize data & dependencies.
-**Milestone / Gantt Charts:** High-level vs detailed planning.

Other Key Concepts
-Opportunity Cost: Value of the next best alternative.
-**MVP (Minimum Viable Product):** Deliver essential value first.
-**Phase Gate:** End-of-phase review.
-**Escalation Path:** Escalate external threats to the sponsor.
-**Parking Lot:** Postpone non-urgent issues.
-**User Personas:** Represent target audience.
-**Monte Carlo Simulation:** Uncertainty analysis using random scenarios.
-**Value Engineering:** A systematic approach to find the **most cost-effective way** to meet project requirements without sacrificing quality or function.

-**5 Whys:** A method for **root cause analysis**—ask “Why?” repeatedly (usually 5 times) to uncover the underlying cause of a problem rather than just addressing symptoms.

-**PMI Code of Ethics:** Four core values guiding project managers’ behavior:
Responsibility: Own your decisions and actions.
Respect: Value people, culture, and opinions.
Fairness: Make impartial decisions, avoid bias.
Honesty: Be truthful and transparent in all dealings.
-**12 Guiding Principles (PMI):** High-level principles for project management success, such as **stakeholder engagement, leadership, governance, risk management, value delivery**, and continuous improvement.
-**Project Performance Domain:** Focuses on **how the team works and how project performance is measured**:
-**ST. Team Development & Lifecycle:** Team formation, evolution, and work cadence.
-**Planning & Project Work Performance:** Organizing tasks, executing work, tracking progress.
-**Delivery & Measurement:** Use data, metrics, and KPIs to assess performance.
Examples:
RTC (Real-Time Completion): Track work done on schedule.
EAC (Estimate at Completion): Forecast project cost at completion.
The Hawthorne Effect: People perform differently when they know they’re observed.
Vanity Metric: Data that looks good but doesn’t measure meaningful outcomes.
Correlation vs. Causation: Don’t assume one thing causes another just because they’re related.
-**Uncertainty:** Focuses on **Risks (threats)** and **Opportunities** that can affect the project.

What is Agile? Agile is **an umbrella term for flexible, iterative approaches to project delivery**, emphasizing collaboration, customer feedback, and continuous improvement rather than following a single rigid methodology.
Scrum (Common Agile Framework): A popular Agile framework focused on **short iterations (sprints), cross-functional teams, and delivering incremental value**.
Lean-Kanban-Agile: Focuses on **visualizing work, limiting work in progress, and improving flow**.
Other Scaled Agile Frameworks:
-SAFe: Lean + Agile + DevOps for enterprise-wide coordination.
-LeSS: Scrum for multiple teams on the same product.
-Enterprise Scrum: Flexible, goal-driven, adapts Agile to context.
-Disciplined Agile (DA): Toolkit to scale Agile and choose the best approach.
-Crystal Methods: Lightweight, focuses on team interactions.
-Lean: Minimize waste, maximize value.
-DSDM: Time-boxed, iterative framework focused on business needs.
-AUP (Agile Unified Process): Simplified RUP, Agile software development.
-Scrumban: Hybrid of Scrum + Kanban for workflow flexibility.

Agile Manifesto:
-**Individuals and interactions over processes and tools** → People and communication are more important than rigid processes.
-**Working software over comprehensive documentation** → Deliver usable results rather than just paperwork.
-**Customer collaboration over contract negotiation** → Engage customers continuously to meet their needs.
-**Responding to change over following a plan** → Adapt quickly rather than rigidly sticking to the original plan.

12 Agile Principles: (simplified meanings)

1. **Customer satisfaction through early and continuous delivery** → Deliver value fast and often.
2. **Welcome changing requirements** → Adapt to business needs anytime.
3. **Deliver working software frequently** → Short iterations, quick feedback.
4. **Business and developers work together daily** → Close collaboration for best results.
5. **Build projects around motivated individuals** → Trust, empower, and support the team.
6. **Face-to-face conversation is most effective** → Communication clarity improves outcomes.
7. **Working software is primary measure of progress** → Tangible results matter most.
8. **Sustainable development** → Maintain pace without burnout.
9. **Continuous attention to technical excellence and good design** → Quality is built-in, not added later.
10. **Simplicity** → Maximize the amount of work not done; avoid unnecessary complexity.
11. **Self-organizing teams** → Teams decide how to achieve goals.
12. **Regular reflection and adaptation** → Improve processes continuously.

Agile Project Charter:

- **Purpose:** Provides an essential, clear vision for the project. Unlike traditional charters, it may not exist initially in Agile, so the **team collaborates with the sponsor** to create it.
- **Key Questions to Answer:**
  -**Why are we doing this?** → Defines the **project vision**.
  -**Who benefits and how?** → Explains **stakeholders and value**; may tie into project vision/purpose.
  -**What does "done" mean?** → Specifies **release criteria**, acceptance, and quality expectations.
  -**How will we work together?** → Defines the **intended flow of work**, team interactions, and collaboration approach.

Project Methodologies:
1. Predictive (Waterfall): Detailed planning upfront; scope, schedule, and budget are defined early.

2. Incremental:
-Deliverables are built in **small, usable increments**.
Scope is partly defined, with clear and well-defined parts.
-**Feedback occurs after each increment**, allowing adjustments.
-Reduces waste by focusing on **functions that meet specific requirements**.
3. **Iterative:**
-Work is completed in **repeated cycles** to refine deliverables.
-**Feedback loops occur throughout**, improving the solution continuously.
-Ideal for **high uncertainty** projects.
-Emphasizes **ongoing refinement and repeated iterations** for continuous improvement.
4. Agile: Flexible, adaptive; emphasizes **collaboration, customer feedback, and responding to change** over rigid plans.
Visualize Work – Show all tasks on a board so the team can see progress and bottlenecks.
Six Principles of Kanban
-**Limit Work in Progress (WIP)** – Set maximum tasks per stage to avoid overloading and improve focus.
-**Manage Flow** – Track and optimize how work moves through the process to ensure smooth delivery.
-**Make Process Policies Explicit** – Clearly define how tasks move and rules for team members.
-**Implement Feedback Loops** – Use regular reviews and standups to improve processes continuously.
-**Improve Collaboratively, Evolve Experimentally** – Encourage small experiments and teamwork to refine processes over time.
Situational Leadership adapts style based on team competence and commitment:
-**Directing** – High guidance, low support; tell team exactly what to do (used for inexperienced team members).
-**Coaching** – High guidance, high support; explain, encourage, and involve team in decisions (used for learning team members).
-**Supporting** – Low guidance, high support; facilitate, listen, and encourage autonomy (used for competent but hesitant team members).
-**Delegating** – Low guidance, low support; empower team to make decisions and work independently (used for highly competent and motivated team members).
-The **OSCAR Model** is a coaching/feedback framework to guide structured conversations, often used in leadership or performance discussions. It stands for:

1. **O – Outcome**: Define the goal or result you want to achieve. *"What do you want to happen?"*
2. **S – Situation**: Describe the current situation or context. *"What is happening now?"*
3. **C – Choices**: Explore possible options or actions. *"What could you do?"*
4. **A – Actions**: Decide on specific steps to take. *"What will you do?"*
5. **R – Review**: Reflect on progress and outcomes after action. *"How will you check success?"*

Helps structure problem-solving or coaching conversations from clarity to action and review.

-In **Agile**, instead of using a traditional **WBS**, work is organized hierarchically as:

- **Theme** → Large strategic focus or goal for the product/project.
- **Epic** → Big features or capabilities that contribute to a theme.
- **User Story** → Small, user-focused requirement: *"As a [user], I want [function] so that [benefit]."*
- **Task** → Specific actionable items to implement a user story.

-Risk-driven prioritization: If a **risk item or story** is identified as **high risk**, it is **prioritized early in the sprint** to reduce uncertainty and potential impact.
This approach allows the team to address the riskiest or most critical parts of the product **first**, aligning with Agile's focus on early delivery of value and iterative learning.
Agile replaces WBS with **themes** → **epics** → **user stories** → **tasks**, and high-risk items get done early to reduce uncertainty.

Product Owner (PO)

- **Writes User Stories**: Captures what the user needs and why.
- **Prioritizes Stories**: Decides which stories go into the backlog and in what order.

Development Team

- **Responsible for Product Increments**: Implements user stories into working product increments.
- **Slicing Stories**: Breaks large epics or stories into smaller, manageable stories that deliver value independently.

-3 C's of User Stories (Card, Conversation, Confirmation)

1. **Card** – The written description of the story (short summary on a card or tool).
2. **Conversation** – Collaborative discussion between PO and team to clarify requirements.
3. **Confirmation** – Acceptance criteria or tests that confirm the story is done.
   PO writes and prioritizes stories, Dev team builds increments, breaks big stories into smaller pieces, and uses **3 C's** to ensure clarity, discussion, and verification.

INVEST is a guideline to ensure high-quality, actionable user stories:

I – Independent: The story can be developed and delivered separately from other stories.
Meaning: No dependencies, can be implemented alone.
N – Negotiable
The story is flexible and can be discussed, refined, or changed.
Meaning: Not a contract; allows collaboration between PO and dev team.
V – Valuable
The story delivers value to the user or customer.
Meaning: Always focused on business or user benefit.
E – Estimable
The team can estimate the effort to complete the story.
Meaning: Clear enough to gauge size and complexity.
S – Small
The story is small enough to be completed in a single sprint.
Meaning: Avoid huge stories (epics); break them down.
T – Testable
-The story has clear acceptance criteria that allow verification.
Meaning: You can confirm the story is done correctly.
A good user story is **Independent, Negotiable, Valuable, Estimable, Small, and Testable**—ready to be prioritized and delivered.
-Story Mapping: A visual technique to **organize and prioritize user stories** along a timeline or workflow.
Helps see the **big picture**: what the user does, the sequence of actions, and which features deliver value first.

- Usually arranged by **themes → epics → user stories → tasks**.
  Simple meaning: Shows **what the user does, in what order, and what features matter most**.

User Persona: A **fictional representation of a target user**.
Includes: **Demographics:** Age, gender, location, role, **Behaviours:** How they use the product or service, **Motivations:** Why they do what they do, what problems they want solved
-**Acceptance Criteria:** What success looks like from their perspective
Helps the team **design features for real users, not abstract ideas**.

Definition of Ready (DOR) vs. Definition of Done (DOD)
-DOR (Ready): Conditions a user story must meet **before the team starts working on it**.
Example: Story is clear, prioritized, has acceptance criteria,and dependencies identified.
Meaning: “Ready to start work.”
-**DOD (Done):** Conditions a story must meet **before it’s considered complete**.
Example: Code written, tested, reviewed, documented, integrated, meets acceptance criteria.
“Work is complete and usable.”

- **DOR = start line** ✅

- **DOD = finish line** 🏁

-I-Shaped Team
Members have **deep expertise in a single area**.
Strength: Highly specialized skills.
Limitation: Harder to collaborate across disciplines.
“Expert in one thing.”
T-Shaped Team
Members have **deep expertise in one area (the vertical bar of T)** and **broad knowledge across other areas (the horizontal bar)**.
Strength: Can collaborate effectively, fill in gaps, and is flexible.
Specialist who can help others.”
Broken Comb – Team members work in smaller, semi-independent groups on different parts of the product simultaneously.
Strengths: Allows parallel work, faster progress, and flexible allocation of resources.
Weaknesses: Can create integration challenges and require strong coordination to avoid conflicts.
-Swarming
Multiple team members **focus together on a single high-priority story or problem**.
Goal: Resolve bottlenecks or high-risk work quickly.
“Whole team attacks one problem.”
-Mobbing: The Entire team works **together on the same story or task at the same time** with one computer (or main output).
Stronger than swarming; every team member contributes simultaneously.
“Team all-in on one thing at once.”
-Daily Standup (Daily Scrum)
Purpose: Quick team sync to share progress, plans, and impediments.
Timebox: Usually 15 minutes, but can extend if needed for onboarding or clarifications.
Focus Questions:

1. What have I completed since the last standup?
2. What am I planning to complete today?
3. Any impediments/blockers?

New Team Members: If someone asks to define roles or clarify responsibilities, **do not disrupt the standup**—capture it in the **parking lot** to handle in the next appropriate meeting (e.g., backlog refinement or sprint planning).
Note: Standup is **not a retrospective**—avoid problem-solving or deep discussions during this time.
-Sprint Review / Demo
Purpose: Show completed work to stakeholders, gather feedback, and adapt future work.
Frequency: Typically every 2 weeks (or at the end of an iteration).
Focus: Demonstrate the increment, inspect the product, collect input, and discuss next steps.

-Iteration-Based (Scrum): Work is planned in fixed-length sprints/iterations (e.g., 2 weeks). Deliverables are completed by sprint end.
-Flow-Based (Kanban): Continuous delivery; work flows through the process as capacity allows. No fixed iteration; focus on limiting work-in-progress (WIP) and maintaining steady throughput.
How Teams Deliver Value in Agile/XP
-Continuous Integration (CI)

- **What:** Frequently merge code into a shared repository.
- **Value:** Detects integration issues early, ensures working software is always available.
- **Example:** Daily builds and automated tests run on new code. **Kaizen / Continuous Improvement**
- **What:** Incremental improvement mindset; small, frequent enhancements. **Value:** Optimizes processes, reduces waste, and increases team efficiency.

-**Testing Practices**
All Levels: Unit, integration, system, and acceptance tests.
Smoke Testing: Quick check to ensure the basic functionality works after a build.
Test-Driven Development (TDD): Write tests before code to ensure correctness.
Acceptance Test Driven Development (ATDD): Collaborate with stakeholders to define acceptance tests before development.
-**Pair Programming / XP Practices**
What: Two developers work together on the same code.
Value: Knowledge sharing, fewer defects, improved design, immediate feedback.
-**Measuring Results**
Agile Metrics: Velocity, burndown/burnup charts, cumulative flow diagrams, cycle time.
Value: Helps track progress, forecast delivery, and identify bottlenecks.
-**Delivering Value Continuously**
Incremental Delivery: Work is delivered in small, usable increments (user stories, features).
Flow-Based (Kanban): Focus on continuous flow, WIP limits, and cycle time improvement.
Feedback Loops: Demo/review sessions with stakeholders to adjust priorities and refine product.
Spike (Agile):

Purpose: Investigate or research something **uncertain or risky** before committing to work.
Why: Used when there's **unknowns**—technical, functional, or dependency-related.
Types:
-Technical spike: Explore technical solutions, new tools, integrations.
-**Risk-based spike:** Investigate risks that could block progress.
-**Functional spike:** Clarify requirements or business logic.
Example: The PO is unsure how a new API will behave. The team runs a **spike** to test and learn, so they can plan the story accurately.
Timebox (Agile):Purpose: Limit the **time spent on an activity** to avoid overwork or over-analysis.
Why: Encourages focus, efficiency, and keeps the team from spending too long on a single task.
Examples:
Daily standups: 15 minutes max.
Spike: 1–2 days to research a problem.
Retrospectives: 1 hour per sprint.
-A spike is usually **timeboxed**, meaning it has a fixed duration to learn something and produce output without delaying the sprint.
-**Team Velocity:** Measure of how much work (story points) a team completes per sprint.
Establishing Velocity:
Track **3–4 sprints** to find a stable average.
Work **one story point at a time** to get an accurate measurement.
-**Calculation:**
Sum all story points completed in a sprint → team velocity for that sprint.
Average over multiple sprints → stable velocity.
-During the sprint, **PM should avoid changing the team's work**, as this can disrupt velocity.
-If a story is missed, PM should **meet the team** to identify why and plan adjustments for future sprints.
-Velocity tracking helps in **forecasting delivery dates** and assessing if Agile methodologies or training are needed.
-Burn Charts
Burndown Chart – Shows **work remaining** over time. Helps track sprint progress.
Burnup Chart – Shows **work completed** vs total scope. Useful for visualizing scope changes.
Cumulative Flow Diagram (CFD) – Shows **work in different stages** of the process (e.g., Analyze, Test, Done).
-**Bottlenecks:** Appear when **work accumulates in a column**, indicating too much workload or process issues.

Bottlenecks

-Short-term: Temporary pile-ups are slowing work in a stage.
-**Long-term:** Chronic inefficiencies blocking flow.
Causes:
-Unequal workload distribution.
-Dependencies not resolved.
-Cycle time (time to complete a task) is longer than lead time (time the task sits in the workflow).
Agile metrics like **velocity, burn charts, and CFD** help teams identify bottlenecks, balance workload, and improve delivery predictability.
Agile Definitions
-Planning Poker: Team estimates story points using cards to reach consensus.
-**Traffic Light Status Reporting:** Visual project status: **Green** = on track, **Yellow** = at risk, **Red** = off track.
-**Team Workspace:**
    - **Fishbowl:** Video conferencing links for team collaboration and visibility.
    - **Remote Pairing:** Virtual tools to share screens and collaborate on tasks.
-**Follow the Sun:** Global team handoff across time zones to enable continuous work.
-Parking Lot: Place to capture questions or issues not addressed immediately; revisit later.
-MBTI: Personality assessment (16 types) to understand communication, team fit, and leadership.
-**Shu Ha Ri:** Agile learning model:
Shu: Follow the rules.
Ha: Move away / break free from rules.
Ri: Innovate independently; go beyond existing practices.
-Information Radiator: Visual display of project status, metrics, and progress for transparency.
-**Project Tweet:** Short, concise project update (~140 characters).
-**Elevator Statement:** Quick explanation of project purpose/value to stakeholders (30 sec–2 mins).
-**Kill Point:** Project review point; project can be stopped if not viable.
-Little's Law: WIP (Work In Progress) = Throughput × Cycle Time; measures workflow efficiency.
-**Escaped Defects:** Defects found **after production release**.

- **Blockers:** Issues preventing a team member or task from progressing; the Scrum Master helps identify during standups.
- **Iterations:** Short, time-boxed cycles (e.g., 2 weeks) to deliver incremental product features.

Project Management Terms

- **Parkinson's Law:** Work expands to fill available time.
- **Student Syndrome:** Delay in starting a task until the deadline approaches.
- **Imposter Syndrome:** A Feeling of not deserving success or doubting abilities.
- **Dropping Baton:** Delays caused when work isn't ready for the next stage.
- **Resource Bottlenecks:** Limited resources slow project progress.
- **Sandbagging:** Intentionally underestimating capacity or progress.

- **Self-Preservation:** Avoiding risk or blame, influencing decision-making.
- **Monochronic vs Polychronic:**
  - **Monochronic:** Focus on one task at a time.
  - **Polychronic:** Handle multiple tasks simultaneously.
- **Gulf of Evaluation:** Delay or difficulty in assessing progress or system feedback.
- **Project Buffer / Feeder Buffer / Resource Buffer:**
  - **Project Buffer:** Extra time in the critical path to absorb delays.
  - **Feeder Buffer:** Extra time on non-critical paths feeding the critical chain.
  - **Resource Buffer:** Additional resource availability to prevent bottlenecks.

-Sashimi in Scrum: Assessment of each phase
-**Thin-Slicing:** Breaking work into smaller, manageable pieces for faster delivery.
-**Discrete Effort:** Work estimated or tracked in clear, countable units contributing project milestone
-**Dangle / Stale Sandwich:** Tasks or work items that are incomplete or stalled, waiting for attention. Stale: use of outdated and inappropriate feedback mechanisms
Procurement & Contract Terms
-Incremental Release Planning: Deliverables are released in small, usable increments, which allows feedback and continuous value delivery.
-**Multi-Tiered Contract Structure:** Contract divided into layers or phases, often aligning with releases or milestones; each tier can have its own terms.
-**Emphasizing Value Delivered:** Focus on contracting for results or functionality rather than just hours or materials.
-Fixed-Price Increments: Set price for each release or increment; helps control costs while delivering value in steps.
-**Not-to-Exceed Time & Materials (T&M):** Maximum cost cap for T&M contracts to limit risk while maintaining flexibility.
-**Graduated Time & Materials:** T&M rate changes over time or based on performance, providing incentive alignment.
-**Early Cancellation Option:** Allows client to stop contract early if deliverables or priorities change; reduces wasted effort.
-**Dynamic Scope Option:** Scope can adjust during contract execution, accommodating evolving requirements.
-**Team Augmentation:** Supplier provides additional skilled resources to the client team for capacity or expertise.
-**Favouring Full-Service Suppliers:** Prefer suppliers who can handle design, development, testing, and delivery end-to-end for simplicity and accountability.

-Types of Agile Methodologies (Key Points)

- **Scrum:** Iterative, incremental; sprints, roles (PO, SM, Dev), ceremonies (standups, retrospectives).
- **Extreme Programming (XP): Technical focus; TDD, pair programming, continuous integration, short iterations.**
- **Lean: Eliminates waste, continuous improvement, maximizes customer value.**
- **Kanban: Visual workflow, WIP limits, continuous flow rather than time-boxed.**

- **Crystal: Lightweight, adaptable to team size and project criticality.**
- **Scrumban: Hybrid of Scrum and Kanban; Scrum events with Kanban boards.**
- **Agile Unified Process (AUP/AgileUP): Simplified iterative process with disciplined Agile practices.**
- **DSDM: Iterative, delivers high-priority features first, and constant stakeholder involvement.**
- **Feature-Driven Development (FDD): Frequent delivery of small, client-valued features.**

Agile Scaling Frameworks (Key Points)

- **Scrum of Scrums (SoS):** Coordinates multiple Scrum teams. A short meeting where **representatives from multiple Scrum teams** coordinate on **dependencies, blockers, and progress**.
- **SAFe:** Scales Agile across team, program, and portfolio levels in large enterprises.
- **LeSS:** Extends Scrum principles to multiple teams on one product.
- **Enterprise Scrum:** Applies Scrum across the whole enterprise for alignment and visibility.
- **Disciplined Agile (DA):** A Hybrid of Agile, Lean, and traditional practices for enterprise-scale adoption.

Note: page 131 (ThirdRock Exam Notes) for Agile ***pain points and troubleshooting tips,*** such as common team challenges, bottlenecks, and scaling issues.

Leadership Styles (PMP)

1. **Visionary:** Shows direction; inspires the team toward a goal.
2. **Coaching:** Helps the team grow skills and achieve long-term goals.
3. **Affiliative:** Focuses on team harmony and relationships.
4. **Democratic:** Encourages team participation in decision-making.
5. **Pace-Setting:** Sets high standards; expects quick results.
   Commanding: Gives orders; used in crises or urgent situations.

PDCA (Deming Cycle)

-Plan → Do → Check → Act

-Continuous improvement loop for processes and products.

-PDM (Precedence Diagram Method): A Visual way to show **task dependencies** in scheduling:

FS (Finish to Start): Task B can start **after** Task A finishes.
SS (Start to Start): Task B can start **when** Task A starts.
FF (Finish to Finish): Task B can finish **when** Task A finishes.
SF (Start to Finish): Task B can finish **after** Task A starts.

-**Cost of Delay-**How much **value is lost** by delaying a feature or project.

-**Kano Analysis**

Categorizes features by **customer satisfaction**:
Must-be, One-dimensional, Attractive, Indifferent, Reverse
-**WSJF (Weighted Shortest Job First)**
Prioritization formula in Agile: **Value / Job Size** → higher score = do first.
-**ICE Scoring**

- Agile prioritization: **Impact × Confidence × Effort**

Agile Decision-Making / Tools

- **HIPPO:** "Highest Paid Person's Opinion" – avoid relying solely on
- **Product Box:** Imagine your product as a product box to prioritize features.
- **Affinity/T-shirt Sizing:** Grouping or sizing tasks by relative effort.
- **Pareto Chart:** 80/20 principle – visualize the main causes of problems.
- **Ishikawa/Fishbone:** Root cause analysis diagram.
- **Deming PDCA:** Continuous improvement framework.
- **Remembered the Future:** Imagine a perfect future outcome to guide decisions.
- **Ideation 2.0:** Structured idea generation.
- **Decision Tree Analysis:** Map decisions and outcomes to choose best path.

Value Stream / Metrics

- **Value Stream Mapping & Analysis:** Track steps adding value; remove waste.
- **Histogram:** Distribution of data; shows frequency of events.

Agile Principles & Practices

- **Four Value-Driven Principles:** Deliver value frequently, prioritize, collaborate, and adapt.
- **Cost Constructive Model:** Budgeting/cost estimation method.
- **Cost of Changes:** Early changes are cheaper; later changes cost more.
- **Refactoring:** Remove redundancy, discard unused features, update designs.
- **Technical Debt:** Accumulated shortcuts or outdated code/decisions.
- **Quality Assessment:** Continuous checks to ensure Agile deliverables meet standards.
- **Expert Judgement & Checklists:** Use experienced input to guide decisions.
- **MVP (Minimum Viable Product):** Build smallest thing to deliver value and get feedback.
- **Product Roadmap:** High-level plan of releases and features.

Agile Collaboration / Work Methods

- **Pairing / Swarming / Mobbing:** Different ways for team members to work together.
- **Speedboat:** Exercise to identify what's slowing the team down.
- **Pre-Mortem:** Imagine project failure to identify risks upfront.
- **Workshops / Prioritization:** Collaborative way to decide what to work on next.
- **Unclear Requirements:** Use iterative clarification.
- **Asynchronous Communication:** Communicate without requiring everyone online at the same time.

3 Scrum Roles

1. **Product Owner:** Defines what needs to be built; prioritizes the backlog.
2. **Scrum Master:** Coaches the team, removes blockers, ensures Scrum is followed.

3. **Development Team:** Builds the product; self-organizing and cross-functional.

5 Scrum Activities (Events)

1. **Sprint Planning:** Decide what to do in the sprint.
2. **Daily Scrum (Stand-up):** Quick daily check-in; what was done, what's next, blockers.
3. **Sprint Review:** Demo the work to stakeholders; gather feedback.
4. **Sprint Retrospective:** Reflect on the process; improve next sprint.
5. **The Sprint:** Time-boxed development cycle (usually 2–4 weeks).

3 Scrum Artifacts

1. **Product Backlog:** All desired work/features for the product.
2. **Sprint Backlog:** Selected items from the backlog for the current sprint.
3. **Increment:** Working, potentially shippable product at the end of the sprint.

Product Backlog (PO Responsibility)

- **Creation:**
  - Product Owner (PO) creates the **Product Backlog**, which is the single source of truth for all features, enhancements, fixes, and technical work needed for the product.
  - The team **contributes ideas** to the backlog, but **only the PO prioritizes**.
- **Grooming / Refinement:**
  - Regular process to **review, clarify, and reprioritize backlog items**.
  - Ensures backlog items are **well-defined, actionable, and ready for upcoming sprints**.
- **Management (PM role):**
  - Record backlog items, track progress, **communicate priorities**, and make sure the backlog reflects stakeholder needs.
- **Notes:**
  - There is **only one product backlog**.
  - **PO has the authority to cancel a sprint if business conditions require it.**

Sprint Lifecycle

A. Sprint Planning (Start of Sprint)

- PO presents prioritized backlog items.
- Team **selects items** for the sprint.
- Sprint goal is defined.
- Output: **Sprint Backlog** (items the team commits to deliver in this sprint).

B. Sprint Execution

- Team develops the items from the Sprint Backlog.
- Daily activities include:
  - **Daily Scrum (stand-up):** 15-minute sync-up to discuss:
    1. What was done yesterday
    2. What will be done today
    3. Blockers/issues

- Collaboration: **Pairing, swarming, or mobbing** may be used for problem-solving.

C. Sprint Review (End of Sprint)

- Team **demonstrates completed work** to stakeholders.
- PO collects feedback.
- Determines if backlog priorities need adjustment.
- **Focus:** Value delivery, not detailed reporting.

D. Sprint Retrospective

- Team reflects on **process, collaboration, tools, and improvements**.
- Typically uses **5-stage process**, around **2 hours** for a standard sprint.
- **ESVP Framework:** Tool to gauge team engagement during retrospectives:
  - **Explorer:** Curious, wants to learn more
  - **Shopper:** Looking for value, may take what they like
  - **Vacationer:** Present physically but not engaged
  - **Prisoner:** Engaged out of obligation, not enthusiasm
- Output: Actionable items to **improve next sprint**.

After Multiple Sprints

- The product vision is **gradually realized**.
- Each sprint produces a **potentially shippable product increment**.
- Iterative development ensures feedback loops and continuous value delivery.

Scrum Framework Components

1. **Product Backlog:** PO-prioritized list of all work.
2. **Sprint:** Time-boxed iteration (usually 2–4 weeks).
3. **Sprint Planning:** Decide which backlog items to tackle in the sprint.
4. **Sprint Backlog:** Selected items for execution.
5. **Sprint Execution:** Development of backlog items.
6. **Daily Scrum:** Team sync-up to monitor progress.
7. **Done / Increment:** Working, tested product ready to deliver.
8. **Sprint Review:** Demo to stakeholders; gather feedback.
9. **Sprint Retrospective:** Reflect and improve team process.

Key Roles & Responsibilities

- **Product Owner (PO):**
  - Owns backlog, prioritizes, can cancel sprint.
  - Ensures product vision is clear and backlog items deliver value.
- **Scrum Master:**
  - Facilitates Scrum, removes blockers, ensures adherence to process.
- **Development Team:**
  - Self-organizing, cross-functional, delivers the work.

- **Project Manager (if involved):**
    - Supports PO, tracks progress, ensures communication, publishes priorities.

Extra Notes / Agile Best Practices

- **Everyone can add items to the backlog**, but prioritization is PO's responsibility.
- **Grooming/Refinement** keeps backlog actionable.
- **Pre-mortem / Retrospective** improves risk management and process.
- **Daily Scrum** is for the team; **review** involves stakeholders.
- **Incremental delivery** ensures continuous value and feedback.
- **The ESVP framework** helps identify engagement in retrospectives.

Think of EVM like a progress report for your project (just tried to simplify the idea)
Imagine your project is like **baking a cake**. You have:
-**Planned ingredients (money & time)** → what you thought you'd use
-**Ingredients actually used** → what you really spent
-**Cake done so far** → how much of the cake is actually baked
EVM helps you see if:
-You are **on schedule**
-You are **on budget**

Core Concepts to Memorize
Earned Value (EV)
→ How much "cake" you've actually baked.
→ More = good, less = behind.
Planned Value (PV)
→ How much cake you *planned* to bake by now.
→ Used for schedule comparison.
Actual Cost (AC)
→ How much money/ingredients you *actually* spent.
→ Less = good, more = bad.

Key PMP Logic: "Compare the Right Things"
Schedule (Are we on time?)
→ Compare **EV vs PV**
→ EV < PV → behind schedule
→ EV > PV → ahead of schedule
Cost (Are we on budget?)
→ Compare **EV vs AC**
→ EV < AC → over budget (bad)
→ EV > AC → under budget (good)
✅ Memory trick:
"Earned vs Planned = schedule"
"Earned vs Actual = cost"

Formulas
S = Schedule → Are we on time?
C = Cost → Are we on budget?
Positive = good, Negative = bad
1 = good, <1 = bad (for SPI/CPI ratios)
SPI = EV / PV
EV (Earned Value) = Value of work *actually completed*
PV (Planned Value) = Value of work *planned to be completed* by now
CPI = EV / AC

EV (Earned Value) = Value of work done
AC (Actual Cost) = Cost *actually spent*

SPI > 1 → ahead schedule
CPI < 1 → over budget
-Communication channel formula:
n (n-1) / 2
n= stakeholder/group/teams

"Cake Story" Visualization
You planned to bake 50% → PV
You actually baked 40% → EV
You spent $30 (planned $50) → AC
→ EV < PV → behind schedule
→ AC < planned → under budget → good news

Situational Question Strategy (PMP Exam Mindset, also in intro)

1. **Read the last line first** – Find out what the question is *really* asking (e.g., "What should the PM do NEXT?").
2. **Identify the framework** – Agile? Hybrid? Predictive? Eliminate options that don't fit that environment.
3. **Identify the process group –** Initiating, Planning, Executing, Monitoring & Controlling, or Closing.
    - Helps you eliminate actions that are "too early" or "too late."

Thinking Order for Action Questions
If the question says:

- "What should you do FIRST?" → Assess/review before action.
- "What should you do NEXT?" → Usually meet, discuss, then act.
- "What should have been done?" → Look for something missed earlier (reactive).
- "What should NOT be done?" → Eliminate options that violate PMP principles (never bypass process, never skip approval).

Sequence:

1. Assess/analyze problem
2. Review plan/documents
3. Meet with team or stakeholders

4. Take appropriate action

Process Group Clues (Quick Recall)

- **Initiating:** Charter, stakeholder identification.
- **Planning:** Plans, baselines, risk register, scope statement.
- **Executing:** Direct work, manage team, manage communication.
- **M&C:** Monitor work, control changes, measure performance.
- **Closing:** Final acceptance, lessons learned, archive documents.

Risk Mindset (Always Assess Before Acting)

- **"May," "might," "could happen" → Risk (update risk register).**
- **"Will," "should," "already happening" → Issue (act or escalate).**
- New regulation or law → Assess impact, update risk register.
- **Confirmed impact → Seek guidance (SME/expert)**, raise a change request if needed.

Team & Stakeholder Situations

- Conflict → Coach or meet face-to-face.
- Stakeholder confusion → Educate, communicate, update stakeholder plan.
- Vendor issue → Meet and **brainstorm with the team** before escalating.
  Payment/contract problem → Involve procurement or finance department, not the sponsor.
- Team underperforming → Check training, motivation, and resource plan (not firing).

Common "PMP Logic" Rules

- Never skip change control. Always **assess → approve → implement**.
- **Never make scope or cost changes without approval.**
- **Never delegate your responsibility to sponsor or PMO.**
- Always prioritize **face-to-face communication** (unless distributed teams).
- For quality issues → Perform quality control or audit.
- Bottom-up estimating → Best for cost accuracy.
- Prototype → Best for demonstrating product concept.
- Deliverables → Must have pre-defined acceptance criteria.
- Project closure → Always formal, documented, and includes lessons learned.

Agile vs Predictive Clues

- **Agile:** Adaptive, collaborative, iterative. Use servant leadership.
- **Predictive (Waterfall):** Sequential, plan-driven, strong documentation.
- **Hybrid:** Combines both. Predictive for governance, Agile for execution.

If a stakeholder doesn't understand Agile → **Coach them.**

Mental Attitude to Keep During Exam

- Think like a **servant leader**, not a boss.
- Never ask others to do your job.
- Always communicate early and often.

- Train and support your team.
- Don't panic about math — but know what CPI/SPI *mean*.
- Always choose the answer that **prevents problems early**, not one that reacts late.

PRACTICE EXAM:

- **XP (Extreme Programming):** Best for **small to medium teams**; practices like pair programming and continuous integration are harder to scale beyond ~20–30 people without heavy coordination.
- **Crystal:** Designed with **scalability in mind**; it has different "flavours" (Clear, Yellow, Orange, Red) for team size and project criticality, so it's **more adaptable to larger teams**.
  Crystal is more naturally scalable than XP.
- **"Sponsor = Signs off & says 'Project Done'"** ✅
- **Compliance issue** → When project work violates laws, regulations, or standards. Logged in the **Issue Log** for tracking and resolution.
  If it needs correction, → raises a Change Request → is handled in Perform Integrated **Change Control (Monitoring & Controlling).**

In Agile Sprint cancellation Product Owner can decide.

- **Project/product cancel:** Sponsor/Business Owner decides.
- **Market research spikes** help the team gather info, validate ideas, and plan features before committing to development.
- Think: **"Beyond my control → escalate."**
- **Cycle Time:** the time it takes a team to **complete a task or work item** from start to finish.
- **Lead Time:** the time from when a **request is made** until the **task is delivered** to the customer.

Cost of Quality (CoQ):

- **Prevention:** cost to **avoid defects** (training, process improvement, quality planning)
- **Appraisal:** cost to **detect defects** (inspections, testing, audits)
- **Internal Failure:** cost of defects **found before delivery** (rework, scrap)
- **External Failure:** cost of defects **found after delivery** (warranty, returns, recalls)

Key: Prevention → avoid, Appraisal → detect, Internal → fix before customer, External → fix after customer.

Cost Baseline vs Project Budget:

-Contingency Reserve: Money or time set aside for **known risks.** The project manager can use it as needed.

-Management Reserve: Money or time set aside for **unknown risks.** You need approval from the sponsor or higher management to use it.

- **Cost Baseline: Approved, time-phased plan** for project costs; used to **measure and control cost performance**.
- **Project Budget: Total authorized funding** for the project, includes **cost baseline + contingency reserves + management reserves**.

Support this Guide

If you find this guide useful and would like to support the effort that went into creating it, you're welcome to buy me a coffee:

https://buymeacoffee.com/pmpsupport