SE423 — Software Project Management

Complete Exam Study Guide · Prepared for Aljowharah (Juju)

Covers: Introduction · Project Performance Domains · Development Approach · Risk Management · Software Engineering · Tailoring · Team

🖨 Table of Contents

1 · Introduction to Project Management

Core Definitions (memorize these word-for-word)

Project: A *temporary* endeavor undertaken to create a *unique* product, service, or result. "Temporary" means it has a beginning AND an end.

Project Management: The application of **knowledge, skills, tools, and techniques** to project activities to meet project requirements.

Program: Related projects, subsidiary programs, and program activities managed in a coordinated manner to obtain benefits not available from managing them individually.

Portfolio: Projects, programs, subsidiary portfolios, and operations managed as a group to achieve strategic objectives.

Product: An artifact that is produced, is quantifiable, and can be either an end item itself or a component item.

Outcome: An end result or consequence of a process or project — broader than outputs, focuses on benefits and value.

Value: The worth, importance, or usefulness of something. Perceived differently by different stakeholders.

🧠 **Memory tip:** Project → Program → Portfolio is the size ladder. One project = one temporary effort. Program = multiple related projects. Portfolio = everything managed together for strategy.

History & Profession

- Some argue Egyptian pyramids / Great Wall were projects.
- Most consider the **Manhattan Project** the first to use "modern" project management (3 years, $2 billion in 1946 dollars, separate PM + technical manager).
- In the **1990s**, companies began creating **PMOs (Project Management Offices)**.
- **PMI** (Project Management Institute) founded in **1969** — international professional society, 652,000+ members by late 2020.
- PMI offers **PMP®** and students can earn **CAPM**.

PMI's Code of Ethics — 4 Core Values

🗝 **Key fact:** Responsibility, Respect, Fairness, Honesty → **RRFH**

Why Project Management Matters

- By 2027, PMI estimates **87.7 million** PM-oriented workers needed.
- Organizations waste **$97 million for every $1 billion** spent on projects.
- Saudi Vision 2030: AI market projected to reach **$135.2 billion by 2030** (12.4% of GDP).
- Top 3 skills employers want: teamwork, problem-solving, verbal communication.

Project Manager — Role & Skills

Main responsibilities: Schedule, Budget, Quality, Delivery, Locking in resources.

Key Competencies

- People skills · Leadership · Listening
- Integrity & ethical behavior (consistent)
- Strong at **building trust** and **building teams**

- Verbal communication · Conflict resolution & management
- Critical thinking & problem solving
- Negotiating · Influencing · Mentoring
- Process and technical expertise

> 🔑 **Key fact:** 97% of successful projects were led by *experienced* project managers.

Advantages of Formal Project Management

Better control of resources · improved customer relations · shorter development times · lower costs + higher productivity · higher quality & reliability · higher profit margins · better coordination · meeting strategic goals · higher morale · less death marches · less overworked personnel.

Self-Check 📝

Project vs program?

→ Project = ONE temporary unique effort. Program = group of RELATED projects managed together for benefits not achievable individually.

Name the 4 PMI ethics values.

→ Responsibility, Respect, Fairness, Honesty (RRFH).

First modern PM project?

→ The Manhattan Project (3 years, $2 billion in 1946 dollars).

2 · PMBOK — 12 Principles & System for Value Delivery

The 12 Principles of Project Management (PMBOK v7)

These are **guidance, not rules**. They overlap but don't contradict each other.

> 🧠 **Memory tip:** Mnemonic — "**STSV · SLTQ · CRAC**": Stewardship · Team · Stakeholders · Value · Systems thinking · Leadership · Tailoring · Quality · Complexity · Risk · Adaptability & resiliency · Change

#	Principle Statement
1	Be a diligent, respectful, and caring **steward**
2	Create a collaborative **team** environment
3	Effectively engage with **stakeholders**
4	Focus on **value**
5	Recognize, evaluate & respond to **system interactions**
6	Demonstrate **leadership** behaviors
7	**Tailor** based on context
8	Build **quality** into processes and deliverables
9	Navigate **complexity**
10	Optimize **risk** responses
11	Embrace **adaptability and resiliency**
12	Enable **change** to achieve the envisioned future state

System for Value Delivery

Projects don't exist in isolation — they live inside a bigger "system for value delivery." Flow:

```
Deliverables → Outcomes → Benefits → Value
```

The 5 parts that define the system

1. **Creating Value** — how projects produce value
2. **Organizational Governance Systems** — supports value delivery, oversight
3. **Functions Associated with Projects** — roles people play
4. **Project Environment** — internal + external factors
5. **Product Management Considerations**

Project Environment: Internal vs External Factors

Internal

Process assets · governance docs · data assets · knowledge assets · security & safety · infrastructure · IT software · resource availability · employee capability

External

Marketplace conditions · social/cultural · regulatory · commercial databases · academic research · industry standards · financial · physical environment

Functions Associated with Projects (8 functions)

1. Provide oversight and coordination
2. Present objectives and feedback
3. Facilitate and support
4. Perform work and contribute insights
5. Apply expertise
6. Provide business direction and insight
7. Provide resources and direction
8. Maintain governance

3 · The 8 Project Performance Domains

Project Performance Domain: A group of related activities critical for the effective delivery of project outcomes. They are **interactive, interrelated, interdependent** — they run **concurrently** throughout the project.

Memory tip: Mnemonic: "**STDPPDMU**" — Stakeholders · Team · Development approach & life cycle · Planning · Project work · Delivery · Measurement · Uncertainty

3.1 Stakeholder Performance Domain

Stakeholder: Individual, group, or organization that may *affect, be affected by, or perceive itself to be affected by* a decision, activity, or outcome.

Stakeholder Analysis: Systematically gathering & analyzing quantitative and qualitative information to determine whose interests should be taken into account.

3.2 Team Performance Domain

Two key skill dimensions:

- **Management** = focused on *means* — processes, planning, coordinating, measuring, monitoring
- **Leadership** = focused on *people* — influencing, motivating, listening, enabling

Centralized vs Distributed

- **Centralized:** One person (PM) accountable. Project charter authorizes them.
- **Distributed:** Shared responsibility, sometimes self-organizing with a facilitator.

Project Team Culture — behaviors the PM models

Transparency · Integrity · Respect · Positive discourse · Support · Courage · Celebrating success

Factors of High-Performing Teams

Open communication · shared understanding · shared ownership · trust · collaboration · adaptability · resilience · empowerment · recognition

3.3 Development Approach & Life Cycle Performance Domain

Deliverable: Unique & verifiable product, result or capability required to complete a process, phase, or project.

Development Approach: Method used to create/evolve the product — predictive, iterative, incremental, adaptive, or hybrid.

Cadence: Rhythm of activities throughout the project.

Project Phase: Collection of logically related activities culminating in deliverables.

Project Life Cycle: Series of phases from start to completion.

Delivery Cadence Types

1. **Single delivery** — one at the end
2. **Multiple deliveries** — different components at different times
3. **Periodic deliveries** — fixed schedule (monthly, bimonthly)
4. **Continuous delivery** — feature increments delivered immediately (DevOps, digital products)

Example Life Cycle Phases

Feasibility → Design → Build → Test → Deploy → Close. A **phase gate** (stage gate) = review at end of phase to check exit criteria before moving on.

3.4 Planning Performance Domain

Essential Definitions

Estimate: Quantitative assessment of likely amount/outcome (cost, resources, effort, duration).

Accuracy: Assessment of correctness. **Precision:** Assessment of exactness. *Not the same!*

Crashing: Schedule compression — shorten duration for the *least incremental cost* by adding resources.

Fast Tracking: Schedule compression — activities normally done *in sequence* done *in parallel* for at least part of their duration.

⚠ **Watch out:** Accuracy ≠ Precision. You can be precise but inaccurate (a tight estimate that's wrong).

Predictive Schedule Planning — 5 Steps

1. Decompose scope into specific activities
2. Sequence related activities

3. Estimate effort, duration, people, resources
4. Allocate people and resources based on availability
5. Adjust until agreed schedule is achieved

Four Types of Dependencies

Type	Meaning	Modifiable?
Mandatory	Contractually required or inherent to the work	Usually NO
Discretionary	Based on best practices or preferences	YES (may be)
External	Between project & non-project activity	Usually NO
Internal	Between two project activities	YES (may be)

Budget Reserves

- **Contingency reserves** = for *identified* risks (known unknowns).
- **Management reserves** = for *unknown*, unplanned in-scope work (unknown unknowns). Managed by sponsor/PMO.

> **Memory tip:** Contingency = "I know this might happen, budgeted for it." Management = "Surprise! Unplanned but in-scope."

Scope: Product vs Project

- **Product scope** = features and functions
- **Project scope** = the *work performed* to deliver them

3.5 Project Work Performance Domain

Bid Documents: Documents used to solicit info, quotations, proposals from prospective sellers.

Bidder Conference: Meetings with prospective sellers before bid prep. Also: contractor/vendor/pre-bid conferences.

Explicit Knowledge: Can be codified with symbols (words, numbers, pictures).

Tacit Knowledge: Personal — hard to articulate (beliefs, experience, insights).

Three Bid Document Types

1. **Request for Information (RFI)**
2. **Request for Proposal (RFP)**
3. **Request for Quote (RFQ)**

Physical Resource Objectives (4)

1. Reduce/eliminate material handling & storage on site
2. Eliminate wait times for materials
3. Minimize scrap and waste
4. Facilitate a safe work environment

3.6 Delivery Performance Domain

Requirement: Condition/capability necessary in a product to satisfy a business need.

Work Breakdown Structure (WBS): Hierarchical decomposition of total scope of work.

Definition of Done (DoD): Checklist of criteria for a deliverable to be ready for customer use.

Quality: Degree to which inherent characteristics fulfill requirements.

Cost of Quality (COQ): All costs for: prevention, appraisal, and failure (internal + external).

Cost of Quality — 4 Components

1. **Prevention** — stop defects (training, good processes)
2. **Appraisal** — find defects (testing, inspection)
3. **Internal Failure** — defects found before delivery (rework)
4. **External Failure** — defects found after delivery — *most expensive!*

3.7 Measurement Performance Domain

Metric: Description of a project/product attribute and how to measure it.

Baseline: Approved version of a work product used as comparison basis.

Dashboard: Set of charts & graphs showing progress vs important measures.

KPIs — Leading vs Lagging

- **Leading indicators:** *Predict* changes/trends — look forward
- **Lagging indicators:** Measure deliverables/events — info *after the fact*

SMART Criteria for Effective Metrics

Specific · **M**eaningful · **A**chievable · **R**elevant · **T**imely

Categories of Metrics

Category	Examples
Deliverable	Errors/defects, performance measures, technical performance
Delivery	Work in progress, lead time, cycle time, process efficiency
Baseline Performance	Start/finish dates, effort & duration, schedule variance
Resources	Planned vs actual utilization; planned vs actual cost
Business Value	Cost-benefit ratio, ROI, NPV
Stakeholders	NPS (Net Promoter Score), Mood chart, Morale, Turnover
Forecasts	ETC, EAC, VAC

Measurement Pitfalls (6)

1. **Hawthorne effect** — people behave differently when measured
2. **Vanity metric** — looks good but doesn't inform decisions
3. **Demoralization** — bad metrics hurt morale
4. **Misusing the metrics**
5. **Confirmation bias**
6. **Correlation vs causation**

Presenting Information — 3 methods

Dashboards · **Information Radiators** (visible displays in team area) · **Visual Controls** (task boards, burn charts)

3.8 Uncertainty Performance Domain

Uncertainty: Lack of understanding/awareness of issues, events, paths, or solutions.

Ambiguity: State of being unclear — difficulty identifying causes or multiple options.

Complexity: Program/project characteristic difficult to manage due to human behavior, system behavior, or ambiguity.

Volatility: Possibility for rapid and unpredictable change.

Risk: Uncertain event/condition that, if it occurs, has *positive or negative* effect on objectives.

Two Categories of Ambiguity

- **Conceptual ambiguity** — lack of shared understanding; same words used differently
- **Situational ambiguity** — multiple possible outcomes or options

Working with Complexity — 3 approaches

Approach	Techniques
Systems-based	Decoupling; Simulation
Reframing	Diversity (brainstorming, Delphi); Balance (mix leading & lagging)
Process-based	Iterate; Engage (stakeholders); Fail safe (redundancy)

4 · Development Approach & Life Cycles

Methodology Concepts — Hierarchy

1. **Process** — collection of activities/tasks creating a work product
2. **Process Model / Life Cycle** — abstract description of a process
3. **Methodology** — instantiation of a process model; prescriptive. "How we work around here."

> 🔑 **Key fact:** McConnell's **Visibility**: "*The ease & accuracy with which it is possible to assess the status of a project's cost, schedule, functionality, or other characteristic.*"

Elements of a Methodology (Cockburn) — 10 items

Teams · Roles · Skills · Techniques · Activities · Process · Work Products · Milestones · Standards · Quality

> 🧠 **Memory tip:** Milestones mark an **instant in time** — either fully met or not (binary).

IT Project = Two Concurrent Life Cycles

- **PLC (Project Life Cycle)** — aimed at *project requirements*
- **SDLC** — aimed at *product requirements*

> ⚠ **Watch out:** Neither alone is enough — need **both integrated** for success.

Software Development Process — 3 Categories

Category	Approaches
Ad hoc	Code and Fix, Rapid Prototyping
Prescriptive (Plan-Driven)	Linear/sequential (Classic, Waterfall); Evolutionary (Iterative/Incremental, Spiral); Unified Process
Adaptive	Lean and Agile methods

Waterfall SDLC

Coined by **Winston Royce in 1970**. Ironically, Royce presented it as an *ill-conceived, risk-prone* approach and advocated iterative feedback! Managers adopted the anti-form *without* feedback loops.

Waterfall Phases

Requirements → Analysis → Design → Construction → Quality Assurance (Testing) → Deployment

Waterfall Failure Symptoms

Protracted integration & late design breakage · Late risk resolution · Requirements-driven functional decomposition · Adversarial stakeholder relationships · Focus on documents & review meetings

When Sequential/Waterfall IS Suitable

Clear/unambiguous/stable requirements · Familiar, proven technology · Low complexity · Adequate time & stable schedule

Evolutionary / Iterative Approach

Top 5 principles:

1. Architecture first — central design element
2. Iterative life-cycle process — risk management
3. Component-based development — technology element
4. Change management environment — control element
5. Round-trip engineering — automation element

Four Life-Cycle Phases (Iterative SDLC)

1. **Inception** — scope, feasibility, architecture synthesis, business case
2. **Elaboration** (most critical!) — detail vision, process, architecture
3. **Construction** — useful versions (alpha, beta), components
4. **Transition** — deployment, assess against vision, plan next iteration

> 🧠 **Memory tip:** I-E-C-T. **Elaboration** is most critical — baseline architecture quickly.

Comparative Expenditure — Waterfall vs Iterative

Activity	Waterfall	Iterative
Management	5%	10%
Requirements	5%	10%
Design	10%	15%
Code & Unit Test / Implementation	30%	25%
Integration & Test / Assessment	40%	25%
Deployment	5%	5%
Environment	5%	10%

Three Development Approaches (PMBOK)

Approach	When to Use
Predictive	Requirements defined up-front; stable scope/schedule/cost (= Waterfall)
Hybrid	Combination. Useful when uncertainty in requirements OR modular deliverables
Adaptive	High uncertainty & volatility; iterative + incremental; agile

Factors Influencing Approach Choice

Category	Factors
Product/Deliverable	Innovation, requirements certainty, scope stability, ease of change, delivery, risk, safety, regulations
Project	Stakeholders, schedule constraints, funding availability
Organization	Structure, culture, capability, team size & location

5 · Agile, Scrum, XP & DevOps

Agile Manifesto (2001) — 4 Values

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

> 🔑 **Key fact:** Items on the right still have value — but items on the LEFT are valued MORE.

12 Agile Principles (condensed)

1. Early & continuous delivery of valuable software
2. Welcome **changing requirements**, even late
3. Deliver working software **frequently** (weeks)
4. Business & developers work **together daily**
5. Build projects around **motivated individuals**
6. **Face-to-face** conversation is most efficient
7. **Working software** = primary measure of progress
8. **Sustainable pace** indefinitely
9. Continuous attention to **technical excellence**
10. **Simplicity** — maximize work NOT done
11. Best designs emerge from **self-organizing teams**
12. **Reflect & adjust** regularly

5 Agile Development Principles

1. Involve the customer
2. Embrace change
3. Develop and deliver incrementally
4. Maintain simplicity
5. Focus on people, not things

Agile Methods Family

Agile is a **blanket term**: Kanban, Crystal, ScrumBan, Scrum, XP, Lean, FDD, DSDM, AUP.

Incremental Development — 5-step Cycle

1. Choose features for increment
2. Refine feature descriptions
3. Implement and test
4. Integrate feature and test
5. Deliver system increment

Extreme Programming (XP)

Coined by **Kent Beck in 1998**. 12 techniques pushing good practices to "extreme" levels.

10 Widely Adopted XP Practices

1. **Incremental planning / user stories** — prioritized, no grand plan
2. **Small releases** — minimal useful functionality first
3. **Test-driven development (TDD)** — write tests FIRST
4. **Continuous integration** — integrated as soon as task complete
5. **Refactoring** — improve structure/readability/security
6. **Pair programming**
7. **Collective ownership**
8. **Simple design**
9. **On-site customer**
10. **Sustainable pace**

Scrum Terminology — EXAM FAVORITE ☆

Term	Definition
Scrum (meeting)	DAILY team meeting reviewing progress & day's work
Sprint	Short period (**2–4 weeks**) developing a product increment
ScrumMaster	Team *coach* (NOT a traditional PM)
Product Owner	Identifies features & attributes; reviews work & tests
Product Backlog	To-do list of bugs/features/improvements
Development Team	Self-organizing **5–8 people**
Potentially Shippable Product Increment	Sprint output — high enough quality to deploy
Velocity	Estimate of work team can do in single sprint

Key Scrum Practices (3)

1. **Product backlog** — to-do list reviewed/updated before each sprint
2. **Timeboxed sprints** — fixed 2–4 week periods
3. **Self-organizing teams** — make own decisions by consensus

Product Backlog Item (PBI) States

1. **Ready for consideration** — tentative, may change
2. **Ready for refinement** — agreed important, needs detail
3. **Ready for implementation** — enough detail to estimate & build

External Interactions Management

- **Team-focused** external interactions → ScrumMaster
- **Product-focused** external interactions → Product Owner

DevOps

DevOps: Extends agile by integrating Development + Operations into one team. Fewer defects because operating requirements considered from start.

DevOps Stats

2021 market: $7,398 million · Projected 2030: $37,227 million (20% CAGR)

6 Best Practices of DevOps Project Management

1. **Develop a collaborative culture**
2. **Embrace the MVP mindset** (Minimum Viable Product)
3. **Focus on both on-time AND quality deliveries**
4. **Emphasize dependencies**
5. **Use tools to do the heavy lifting** (automation)
6. **Track the right metrics** (lead time, mean time to detect, severity, unit cost)

DevOps PM Role — 2 Key Shifts

1. **Treat Dev & Ops as a single organization** — prevent silos; value stream maps extend Gantt Chart planning
2. **Effectively embrace change** — MVP, agile tools, eliminate silos, reduce handoffs, real-time visibility, less overhead

Self-Check

Scrum meeting vs Sprint?
→ Scrum = DAILY meeting. Sprint = 2–4 week DEVELOPMENT period.

Development team size?
→ 5 to 8 people, self-organizing.

ScrumMaster vs Product Owner?
→ ScrumMaster = team coach (team-focused externals). Product Owner = product features/attributes (product-focused externals).

Who coined XP and when?
→ Kent Beck, 1998.

6 · Risk Management

What is a Risk?

> **PMBOK:** "Project risk is an uncertain event or condition that, if it occurs, has a *positive or negative effect* on one or more project objectives."

A risk = a problem that **hasn't happened yet**.

Characteristics of a Risk

- **Uncertainty** (0 < probability < 1)
- **An associated loss** (money, life, reputation)
- **Manageable** — some action can control it
- **Proactive vs. Reactive** — active risk mgmt = mature org

Four Questions Defining a Risk

1. What can go wrong?
2. What is the likelihood?
3. What will the damage be?
4. What can we do about it?

Risk Categories by Predictability — 4 Types

Type	Description	Example
Known	Uncovered after evaluation; estimable from history	Vendor delay, key staff leaves, systems down
Predictable	Extrapolated from past — probable but impact uncertain	Staff turnover, poor communication
Unpredictable	"Joker" risks — hard to predict	—
Unknowable	Outside historical models; needs disaster recovery	Disasters, terrorism, war

Three Types of Software Risk — EXAM FAVORITE ⭐

Type	Threatens	Examples
Project Risks	Project plan (schedule, cost)	Budget, schedule, personnel, resources, customers, requirements, complexity, hardware, environment
Technical Risks	Quality & timeliness of software	Design, implementation, interfacing, verification, cutover, maintenance, security
Business Risks	Viability of the product	Market, strategic, sales, management, budget, contracts, political, legal

5 Sub-types of Business Risk

1. **Market risk** — great product nobody wants
2. **Strategic risk** — doesn't fit company strategy
3. **Sales risk** — sales force doesn't know how to sell it
4. **Management risk** — loss of senior management support
5. **Budget risk** — loss of budget/personnel commitment

Risk Identification — Tools & Techniques

1. **Documentation reviews**
2. **Information-gathering:**
   - **Brainstorming** — self-regenerating
   - **Delphi** — facilitator distributes questionnaire; participants DO NOT interact directly
   - **Interviews**
3. **Root cause analysis** — "Five Whys?" (Hall, 1998)
4. **Checklist analysis** — based on historical info; never exhaustive
5. **Assumptions analysis** — false assumptions become risks
6. **Diagramming:**
   - **Cause-and-effect (Ishikawa / fishbone)**
   - **System/process flowcharts**
   - **Influence diagrams**

Risk Item Checklist — 7 Sub-categories

1. **Product size** — risk is *directly proportional* to size
2. **Business impact**
3. **Customer characteristics**
4. **Process definition**
5. **Development environment**
6. **Technology to be built**

7. **Staff size and experience**

US Air Force Risk Components (4)

1. **Performance risk** — product fit for purpose
2. **Cost risk** — budget maintained
3. **Support risk** — easy to correct/adapt/enhance
4. **Schedule risk** — on-time delivery

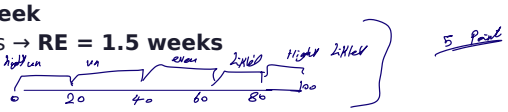
Risk Analysis — Calculations ☆

```
Risk Exposure (RE) = Probability × Size of Loss
```

Examples

- "Facilities not ready on time" — P=25%, size=4 weeks → **RE = 1 week**
- "Inadequate design — redesign required" — P=15%, size=10 weeks → **RE = 1.5 weeks**

Sum all REs = expected overrun.



Probability Scales

5-point Likert: Highly unlikely (<20%) · Unlikely (20–40%) · About even (40–60%) · Likely (60–80%) · Highly likely (>80%)
3-point: Low (<35%) · Moderate (35–75%) · High (>75%)

Reactive vs Proactive Risk Management

Reactive (bad)	Proactive (good)
React when risks occur	Formal risk analysis
Plan for fire-fighting resources	Correct root causes
Fix on failure	TQM & statistical SQA
Crisis management	Skill to manage change

Risk Response Planning — Threats (5 strategies) ☆

Strategy	Meaning
Avoid	Eliminate the cause / change plan so it can't happen
Escalate	Pass up to management (outside project authority)
Transfer	Move to third party (insurance, contract)
Mitigate	Reduce probability or impact
Accept	Passive (document) or Active (set up reserves)

Risk Response Planning — Opportunities (5 strategies) ☆

Strategy	Meaning
Exploit	Ensure the opportunity happens
Escalate	Pass up to management
Share	Partner with third party better positioned
Enhance	Increase probability or impact
Accept	Take the opportunity if it comes

Memory tip: THREATS: **A·E·T·M·A** · OPPORTUNITIES: **E·E·S·E·A**. Escalate and Accept are in BOTH lists.

Contracting — Two Types

- **Fixed-price** — Contractor raises price to cover accepted risk
- **Cost reimbursable** — Contractor paid for extra costs; *majority of risk stays with buyer*

Residual vs Secondary Risks

- **Secondary risks** = *caused BY* implementing a risk response
- **Residual risks** = leftover, can't be fully dealt with; handled by contingency reserves

Seven Principles of Risk Management

1. Maintain a global perspective
2. Take a forward-looking view
3. Encourage open communication
4. Integrate
5. Emphasize a continuous process

6. Develop a shared product vision
7. Encourage teamwork

Risk Categories for Planning

Category	Examples
Technical/Quality/Performance	Unproven tech, tech changes, unrealistic goals
Project Management	Improper schedule/resource planning, poor planning
Organizational	Resource conflicts, unrealistic objectives, lack of funding, politics
External	Laws, labor, weather, ownership changes, catastrophic

Risk Breakdown Structure (RBS) = hierarchical decomposition of risk categories — analogous to WBS.

Mitigation, Monitoring, Management

- **Mitigation** — how to avoid the risk
- **Monitoring** — track if risk is becoming more/less likely
- **Management** — contingency plans if risk becomes reality

Self-Check — Risk

RE for P=30%, loss=20 weeks?
→ 0.30 × 20 = 6 weeks.

5 threat responses?
→ Avoid, Escalate, Transfer, Mitigate, Accept.

5 opportunity responses?
→ Exploit, Escalate, Share, Enhance, Accept.

Residual vs secondary risk?
→ Secondary = caused BY response. Residual = leftover AFTER response.

3 types of software risk?
→ Project, Technical, Business.

7 · Software Engineering — Product vs Project

Software Engineering: Engineering discipline concerned with ALL aspects of software production from specification to post-deployment maintenance.

Product vs Project — The Core Distinction

	Project-Based	Product-Based
Starting point	Client requirements	Business opportunity identified by developer
Who decides features	Customer	Software company
Who pays	External client	Many customers buy
Who changes requirements	Customer (any time)	Developer
Lifetime	Custom, often 10+ years	Rapid delivery to capture market

Software Execution Models

1. **Stand-alone:** Entirely on customer's computers
2. **Hybrid:** Part customer, part vendor servers
3. **Software as a Service (SaaS):** All on vendor servers; browser/mobile access

3 Software Process Models

1. **Waterfall** — separate phases
2. **Incremental development** — interleaves spec, dev, validation; versions
3. **Integration and configuration** — reusable components

Capability Maturity Model (CMM) — 5 Levels

1 Initial · 2 Managed · 3 Defined · 4 Quantitatively Defined · 5 Optimizing

The 4 Life Cycles Summary

Approach	Requirements	Activities	Delivery	Goal
Predictive	Fixed	Once for whole project	Single	Manage cost
Iterative	Dynamic	Repeated until correct	Single	Correctness
Incremental	Dynamic	Once per increment	Frequent smaller	Speed
Agile	Dynamic	Repeated until correct	Frequent smaller	Customer value via frequent deliveries + feedback

Stacey Complexity Model

Two axes: **Requirements uncertainty** × **Technical uncertainty**

- Low both → **Simple** (linear works)
- One high → **Complicated**
- Both high → **Complex** (adaptive works)
- Very high both → **Chaos** (fundamentally risky)

7 Software Product Quality Attributes

Reliability · Usability · Maintainability · Security · Responsiveness · Resilience · Availability

Software Product Line & Platform

- **Product line:** products sharing a common core with customer-specific adaptations
- **Platform:** product allowing new apps to be built on it (e.g., Facebook + Facebook apps)

The Product Vision — 3 Questions

1. **What** is the product?
2. **Who** are the target customers/users?
3. **Why** should customers buy it?

Moore's Vision Template ⭐

```
FOR (target customer) · WHO (statement of need/opportunity)
The (PRODUCT NAME) is a (product category)
THAT (key benefit, compelling reason to buy)
UNLIKE (primary competitive alternative)
OUR PRODUCT (statement of primary differentiation)
```

4 Information Sources for a Product Vision

1. **Domain experience**
2. **Product experience**

3. **Customer experience**
4. **Prototyping and playing around**

Product Manager — 3 Concerns

1. **Business needs** — meet company goals
2. **Technology constraints**
3. **Customer experience**

Product Manager — 6 Technical Interactions

1. Product vision management (prevent "vision drift")
2. Product roadmap development
3. User story and scenario development
4. Product backlog creation and management
5. Acceptance testing
6. Customer testing
7. User interface design

Product Prototyping — Two Stages

1. **Feasibility demonstration**
2. **Customer demonstration**

> ⚠ **Watch out:** Always plan to **throw away** the prototype and re-implement for security & reliability.

Customer Value

```
Customer Perceived Value = Total Benefit – Total Cost
```

> **Total Customer Benefit**
> Product · Services · Personnel · Image
>
> **Total Customer Cost**
> Monetary · Time · Energy · Psychological

Business Model Canvas — 9 Blocks

Key Partners · Key Activities · Key Resources · Value Proposition · Customer Relationships · Channels · Customer Segments · Cost Structure · Revenue Streams

8 · Tailoring, Models, Methods & Artifacts

Tailoring: The *deliberate adaptation* of project management approach, governance, and processes to make them more suitable for the given environment and work.

Why Tailor? — 3 Benefits

1. **More commitment** from team members who helped tailor
2. **Customer-oriented focus**
3. **More efficient** use of resources

What Can Be Tailored? — 5 Aspects

1. **Life cycle and development approach** selection
2. **Processes**
3. **Engagement**
4. **Tools**
5. **Methods and artifacts**

Process Tailoring — Options

Add · Modify · Remove · Blend · Align

Engagement Tailoring — 3 Considerations

People · Empowerment · Integration

The 4-Step Tailoring Process ☆

1. **Select the initial development approach**
2. **Tailor for the organization**
3. **Tailor for the project**
4. **Implement ongoing improvement**

Tailoring for the Project — 3 Attribute Categories

Category	Attributes
Product/Deliverable	Compliance/criticality, type, industry, technology, time frame, stability, security
Project Team	Size, geography, organizational distribution, experience, customer access
Culture	Buy-in, trust, empowerment, organizational culture

Key Definitions — Models, Methods & Artifacts

Model: A *thinking strategy* to explain a process, framework, or phenomenon.

Method: The *means* for achieving an outcome, output, result, or deliverable.

Artifact: A *template, document, output, or deliverable*.

🧠 **Memory tip:** Model = "how to think." Method = "how to do." Artifact = "a thing you produce."

Commonly Used Models — By Category

Category	Examples
Situational leadership	Situational Leadership® II · OSCAR
Communication	Cross-Cultural · Effectiveness of Channels · Gulf of Execution & Evaluation
Motivational	Hygiene & Motivational · Intrinsic vs Extrinsic · Theory of Needs · Theory X/Y/Z
Change	Managing Change in Orgs · ADKAR® · 8-Step Leading Change · Virginia Satir · Transition
Complexity	Cynefin Framework · Stacey Matrix
Team development	Tuckman Ladder · Drexler/Sibbet
Other	Conflict · Negotiation · Planning · Process Groups · Salience

Commonly Used Methods — 4 Categories

Data gathering & analysis · Estimating · Meetings & events · Other methods

Commonly Used Artifacts — 8 Categories

Strategy · Logs & registers · Plans · Hierarchy charts · Baselines · Visual data · Reports · Agreements & contracts · Other

Things to Avoid in Choosing Models/Methods/Artifacts

- Duplicates or unnecessary effort
- Not useful to team or stakeholders
- Incorrect or misleading information
- Caters to individual needs vs team's

9 · Team — Leadership, Trust, Conflict, People

Five Types of Influence (Power) ☆

1. **Formal (Legitimate)** — formal position/authority
2. **Reward** — rewards/incentives (bonuses, praise)
3. **Coercive** — threats, punishment; fear-based
4. **Expert** — knowledge, expertise, skills
5. **Referent** — charisma, admiration, identification

> **Memory tip: F·R·C·E·R** — Formal, Reward, Coercive, Expert, Referent.

Leadership Styles

- **Directing (Autocratic)** — unilateral; emergencies
- **Delegating** — autonomy; experienced teams
- **Laissez-Faire** — hands-off; self-motivated teams
- **Facilitating** — promotes collaboration & problem-solving
- **Consultative (Analytical)** — seeks input before deciding
- **Coaching** — mentorship, growth
- **Consultative-Autocratic** — asks input BUT leader decides
- **Charismatic** — vision + personal qualities
- **Driver** — high expectations, pushes performance
- **Supporting** — nurturing, well-being focused
- **Consensus** — collective decisions
- **Democratic / Participative** — team input
- **Influencing** — persuasion, negotiation, alliances

Tuckman's Model — Team Development ☆

1. **Forming** — group comes together
2. **Storming** — chaos, vying for leadership
3. **Norming** — agreement on how to operate
4. **Performing** — effective at meeting objectives
5. **Dissolving / Adjourning**

Trust — Foundation of Teamwork

- Trust is about **vulnerability** — difficult for most.
- Takes time; needs maintenance.
- Techniques: Behavioral profiling (Myers-Briggs).

How to Establish Trust

Earn it (not granted) · Open communication · Be fallible (own mistakes) · Lead by example, not mandate · Complete commitments on time · Don't confuse friendship with leadership

Managing Conflict — 6 Resolution Techniques

1. **Confronting (Problem Solving)**
2. **Compromising**
3. **Withdrawal (Avoidance)**
4. **Smoothing (Accommodating)**
5. **Collaborating**
6. **Forcing**

Problem-Solving Steps

1. Define the real/root problem
2. Analyze the problem
3. Identify solutions
4. Pick a solution
5. Implement
6. Review & confirm it worked

> **Key fact:** Disagreement ≠ Conflict. Conflict involves **hardening of position** and intractability.

Accountability & RACI

RACI = Responsible, Accountable, Consulted, Informed. **RAM** (Responsibility Assignment Matrix) shows "who does what" — can be for planning OR tracking.

Skills Matrix

Resources on one axis, skills on other. Cells = X's or numeric (level, years experience).

Team Models

Model	Key Features
Business Team (most common)	Tech lead + equal-status team; hierarchical with one principal contact
Democratic Team	Whole team decides (Weinberg's "egoless programming")
Chief-Programmer Team (IBM 70s, Brooks)	Superstar + backup, admin, tool-smith, "language lawyer"
Skunk Works	Isolated creatives; high ownership, low visibility; exploratory
SWAT Team	Highly skilled, skills match goal (security, performance)

Team Size Math

> 🔑 **Key fact:** Communication paths grow with the **square** of team size. 50 programmers = 1,200 paths!

- Optimal team: **4–6 developers**
- Always < 10 (**seven plus or minus one**)
- Large teams → hierarchy, formalize communication, smaller units

Maslow's Hierarchy of Needs

1. Physiological (food, sleep)
2. Safety
3. Social (belonging)
4. Esteem (respect, self-esteem)
5. Self-realization / Self-actualization

Lower needs must be satisfied before higher. If lower needs unmet → person re-prioritizes DOWN.

Need Satisfaction in PM Context

- **Social** — communal facilities, informal comms
- **Esteem** — recognition, appropriate rewards
- **Self-realization** — training, responsibility

3 Personality Types (Motivation)

1. **Task-oriented** — motivated by work itself
2. **Self-oriented** — work as means to end
3. **Interaction-oriented** — motivated by co-workers

Effective team = **balance of all three**. Most engineers are task-oriented.

Staff Selection Factors (8)

Application domain experience · Platform experience · Programming language · Educational background · Communication ability · Adaptability · Attitude · Personality

Time Distribution of a Software Engineer

20% non-productive · **30%** working alone · **50%** interaction

Group Cohesiveness — Benefits

Group quality standards develop · members work closely, inhibitions reduced · learn from each other · egoless programming possible

> ⚠ **Watch out:** **Groupthink:** Preserving the group despite technical/organizational concerns. Force *external involvement* to prevent it.

Technical vs Adaptive Leadership

	Technical	Adaptive
Solves problems by	Known means	Unknown means
Requires learning	Just execution	YES — leader + group
Uses	Vision (personal)	Sense of purpose (big-picture)
Motivates through	Personality	Progress

Most IT projects lean toward the *unknown* → need adaptive leadership.

Consensus-Building Steps

1. Pose the problem/decision
2. Solicit input on options
3. Discuss pros and cons
4. Work toward best option
5. Persuade opponents to buy in

> ⚠ **Watch out:** Beware passive/aggressive agreement — this is just subsumed conflict.

"Thought Workers" — Common Management Errors

Squeeze out error · Hard line on goofing off · Workers as interchangeable parts · Optimize steady state · Standardize procedure (run by book) · Eliminate experimentation

Personal Software Process (PSP)

Tracks daily time: Reading · Writing · Meeting · Training · Requirements · Design · Coding · Testing · Deployment · Email · Phone. Helps manage quality, meet commitments, improve estimates, reduce defects.

Self-Check — Team

Tuckman's 5 stages?

→ Forming, Storming, Norming, Performing, Adjourning.

5 types of influence?

→ Formal, Reward, Coercive, Expert, Referent.

Optimal team size?

→ 4–6 developers; always < 10 (seven ± one).

Communication paths for 20 people?

→ n(n-1)/2 = 20×19/2 = 190.

Disagreement vs conflict?

→ Disagreement = differing views. Conflict = hardening of positions + intractability.

⚡ 10 · Last-Minute Quick Reference Sheet

The 12 PMBOK Principles

Stewardship · Team · Stakeholders · Value · Systems thinking · Leadership · Tailoring · Quality · Complexity · Risk · Adaptability & Resiliency · Change

The 8 Performance Domains

Stakeholders · Team · Development Approach & Life Cycle · Planning · Project Work · Delivery · Measurement · Uncertainty

4 PMI Ethics Values

Responsibility · Respect · Fairness · Honesty

4 Agile Values

1. Individuals & interactions > processes & tools
2. Working software > comprehensive documentation
3. Customer collaboration > contract negotiation
4. Responding to change > following a plan

Key Formulas

```
Risk Exposure = Probability × Loss Size
```

```
Customer Perceived Value = Total Benefit − Total Cost
```

```
Communication paths = n(n-1)/2
```

Risk Responses

Threats (A·E·T·M·A): Avoid · Escalate · Transfer · Mitigate · Accept

Opportunities (E·E·S·E·A): Exploit · Escalate · Share · Enhance · Accept

5 Types of Influence

Formal · Reward · Coercive · Expert · Referent

Tuckman's Stages

Forming · Storming · Norming · Performing · Adjourning

SMART Metrics

Specific · Meaningful · Achievable · Relevant · Timely

CMM Levels

1 Initial · 2 Managed · 3 Defined · 4 Quantitatively Defined · 5 Optimizing

Maslow's Hierarchy (bottom → top)

Physiological → Safety → Social → Esteem → Self-realization

Iterative SDLC Phases

Inception → Elaboration (⭐ most critical) → Construction → Transition

Waterfall Phases

Requirements → Analysis → Design → Construction → QA/Testing → Deployment

Scrum Facts

- Sprint = 2–4 weeks · Dev team = 5–8 people · Daily Scrum = daily meeting
- ScrumMaster = coach, NOT traditional PM
- Product Owner = defines features, prioritizes backlog
- Output = "Potentially Shippable Product Increment"
- Velocity = estimate of work per sprint

4 Dependency Types

Mandatory	Contractual/inherent	Usually NOT modifiable
Discretionary	Best practices/preference	May be modifiable
External	Project ↔ non-project	Usually NOT modifiable
Internal	Between project activities	May be modifiable

3 Bid Document Types

RFI (Info) · RFP (Proposal) · RFQ (Quote)

Schedule Compression — 2 Methods

Crashing — add resources for least cost increase · **Fast tracking** — parallel work

3 Types of Software Risk

Project · Technical · Business

6 Measurement Pitfalls

Hawthorne · Vanity · Demoralization · Misuse · Confirmation bias · Correlation vs causation

4-Step Tailoring Process

1. Select initial approach · 2. Tailor for organization · 3. Tailor for project · 4. Implement ongoing improvement

Model vs Method vs Artifact

Model = thinking strategy · **Method** = means to achieve · **Artifact** = template/doc/output

7 Principles of Risk Management

Global perspective · Forward-looking · Open communication · Integrate · Continuous process · Shared product vision · Encourage teamwork

Contingency vs Management Reserves

Contingency = known risks (known unknowns), managed by project · **Management** = unknown in-scope work, managed by sponsor/PMO

Final 60-Second Scan Before the Exam 🎯

1. Project = *temporary* + *unique*
2. PMBOK = 12 principles + 8 performance domains
3. Risk = threats + opportunities (5 responses each)
4. RE = P × Loss
5. Agile = 4 values + 12 principles
6. Scrum: 2–4 weeks, 5–8 people, PO + SM roles
7. XP: Kent Beck 1998 (TDD, refactoring, pair programming, CI)
8. Waterfall: Royce 1970 — presented as a BAD example
9. Iterative SDLC: I-E-C-T (Elaboration ⭐)
10. Tuckman: Forming/Storming/Norming/Performing/Adjourning
11. Tailoring: 4 steps; tailor 5 aspects
12. 5 influences: F·R·C·E·R
13. Optimal team: < 10; 4–6 devs
14. Accuracy ≠ Precision
15. Secondary risk = caused BY response; Residual = left over

Good luck on your exam tomorrow, Juju! You've got this. 💪