

**Software Project Management Exam Personal-Study Material**

This material provides valuable insights to aid in your exam preparation. While it offers useful guidance, it is not meant to serve as your sole preparation resource. In addition to this material, it's essential to incorporate all the topics covered in both lectures and tutorials into your exam preparation. The intention behind this resource is to assist you in navigating through the lecture slides efficiently, preventing the need to review a large volume of slides aimlessly. Also, remember that you have access to short support videos, which can be beneficial. We trust that you will find this material helpful.

Included below are a set of questions and exercises designed to assist you as you prepare for the exam. However, please note that this is just a guide. As mentioned earlier, we encourage proactive studying and recommend using the slides provided during lectures and tutorials as a fundamental part of your preparation process."

# General

* EXPLAIN WHY the objectives of a software project should be SMART? What does SMART stand for?

**S – specific**

**M – measurable**

**A – achievable**

**R – relevant**

**T – time-constrained**

Should be smart as it allows you to track project progress, ensure alignment with goals and that the project is completed within the set deadline.

* EXPLAIN WHY is software project management important? Why is it needed?

**Software project management is essential because it provides structure, planning, and oversight to ensure that software projects are completed on time, within budget, and with the desired quality.**

EXPLAIN WHY is software project management different from other project management disciplines? PROVIDE three reasons.

* **Requirements are often changing, so a suitable methodology is required (eg. Agile)**
* **Products are intangible therefore harder to inspect quality/ progress.**
* **Managers need to account for technological aspects and ensure their team is trained in those fields.**

* EXPLAIN the advantages and disadvantages of off-the-shelf software.

|  |  |
| --- | --- |
| **Buying Software: Pros** | **Cons** |
| No down time due to dev | Ownership and customisation |
| Errors spotted by other users and resolved | Competitive Edge |
| Cheaper as cost spread out between buyers | Not dependent on a supplier |

* EXPLAIN the advantages and disadvantages of prototyping.

Advantages:

* **Can check if all requirements have been included and met**
* **Check quality, has iso 9126 standards been adhered to (functionality, reliability, usability, efficiency, maintainability, and portability)**
* **Used to gather user feedback**

**Disadvantages of Prototyping:**

1. **Time-Consuming.**
2. **Over emphasis on aesthetics**
3. **Scope Creep.**

* EXPLAIN WHAT are the reasons for prototyping?

1. **Learning by doing**
2. **Improved communication**
3. **Improved user involvement**
4. **A feedback loop is established.**

* EXPLAIN the advantages and disadvantages of incremental development process.

1. **feedback from early stages used in developing later stages**
2. **easier to cope with changing requirements**
3. **user gets some benefits earlier**
4. **some costs will be repeated**
5. **software breakage – later increments may change earlier** **increments.**

* EXPLAIN WHAT are incremental development process and agile methods?

**Agile: Emphasis on speed of delivery rather than documentation • RAD – Rapid application development emphasized use of quickly developed prototypes.**

**Incremental: The application to be delivered is broken down into a number of components called lots that will be actually used by customers • Each components is developed as a separate increment**

* EXPLAIN WHAT are the differences between incremental delivery and prototype?
* **Incremental delivery focuses on gradually building a complete and functional system over time through successive iterations, while**
* **Prototyping is primarily used for quickly creating a preliminary model to clarify and refine requirements, often with less concern for the final product's completeness**
* EXPLAIN WHY apply agile methods?

**Agile methods are chosen in software development for their ability to adapt to changing requirements, foster collaboration, and deliver incremental value.**

EXPLAIN WHAT are the recommendable ‘rules of thumb’ to decide which development process should be used (How to choose a model)?

**• If uncertainty is high: – Agile or prototyping**

**• If complexity is high but uncertainty is not: – Incremental delivery**

**• If uncertainty and complexity are both low: – Waterfall**

**• If schedule is tight: – Evolutionary prototype or agile**

* WHAT is Project size estimation?

**Project size estimation in project management refers to the process of predicting the most realistic amount of effort, resources, and time required to complete a project.**

NAME and EXPLAIN three techniques to improve estimation in Software Project Management. Name advantages and disadvantages of each one.

* **Three-Point Estimation: calculating best case, likely case and worst case scenario**
* **Risk Exposure: assigning a monetary value to risks to estimate the cost impact**
* **PERT: stats used to estimate project duration using delphi average and variance**

# Software Project Planning

* DEFINE the objective of the PLANNING activity and STATE what it enables. GIVE an EXAMPLE per each item.

**Project planning ensures the goals and requirements of the project is met, with sufficient quality, in a timely manner. It enables:**

**Functionality: establishes core project capabilities e.g project scope**

**Reliability: ensure fault free errors- how long it takes to resolve issues e.g test cases**

**Usability: how well users understand/ learn the system e.g usability testing**

**Efficiency: the resources and time taken e.g resource allocation**

**Maintainability: stability e.g clear documentation**

* NAME THREE ways in which the ACTIVITY network helps the Software Project Manager.
* Allows manager to monitor activity dependencies to ensure smooth work flow.
* Allows manager to monitor critical path, and allocate resources to time sensitive tasks
* Allows manager to estimate project duration, and feed that back to stakeholders.
* DESCRIBE all the approaches that can be used to identify activities?
* Top down approach: making general estimates using a previous project
* Bottom up approach: break the project into smaller components ensuring a focused approach
* Expert judge approach: having experts come to a consensus, allows for qualitative answers
* **Activity-Based Approach: This is like creating a to-do list for your project. List every task from beginning to end, ensuring each has a clear start, finish, and everything it needs to get done.**
* **Product-Based Approach: Focus on what you want to end up with – the final products. From there, work out what activities are needed to create these deliverables.**
* **Hybrid Approach: Combine the first two approaches. Think about both the tasks you need to do and the end products you want to achieve, ensuring a comprehensive plan.**
* CRITICALLY COMPARE PRINCE (i.e. PRINCE 2) and Step Wise.
* **Scope and Application: PRINCE2 is suitable for a wide range of projects, not limited to software development, whereas Step Wise is more tailored to software project management.**
* **Complexity and Detail: PRINCE2 is more detailed and can be seen as more complex due to its comprehensive nature. Step Wise offers simplicity and is easier to implement in smaller projects.**
* **Documentation and Process Rigor: PRINCE2 requires more in-depth documentation and adherence to its defined processes, which can be advantageous for larger, more complex projects. Step Wise, with its lean approach, is more adaptable but might lack the rigor required for larger-scale projects.**
* **Flexibility and Tailoring: Both methodologies allow tailoring to specific project needs, but Step Wise inherently offers more flexibility due to its less prescriptive nature.**
* **Recognition and Certification: PRINCE2 is internationally recognized with widely acknowledged certification, making it a preferred choice for many organizations and project management professionals. Step Wise, being less formalized and recognized, might not carry the same weight in the professional community.**
* Critical path: EXPLAIN WHAT is the critical path of a plan? How is it used? EXPLAIN WHAT is it useful for? Can there be more than one critical path? Can there be no critical path at all?
* **Definition: The longest path through network, any delay on the critical path will delay project deadline.**
* **Use: Focuses on key tasks affecting the project deadline**
* **Utility: Essential for prioritizing critical tasks, managing risks, and ensuring on-time project completion.**
* **Multiple Paths: Yes, there can be more than one critical path, increasing project complexity.**
* **Absence of a Critical Path: Rare, but possible in extremely simple or undefined projects; usually, at least one exists in well-planned projects.**
* EXPLAIN what a sub-critical path is and WHY it needs to be monitored. WHY is a sub-critical path important?

**Definition: A sequence of tasks in a project that is not the longest (critical) path but is close in duration and can become critical if delayed.**

* **Monitoring Need: Essential to monitor as any delays can turn it into the new critical path, impacting the project timeline.**
* **Importance: Tracking sub-critical paths helps in proactive risk management and maintaining schedule flexibility, ensuring overall project robustness against unexpected changes.**
* EXPLAIN what objective-based software projects and product-based software projects are and their differences.

**Objective-Based Projects:**

* **Focus:** Achieving specific goals or solving problems. Defined by client or business needs

**Product-Based Projects:**

* **Focus:** Developing a marketable software product. Driven by market demands and features.
* **Differences…Objective-Based:** Goal-driven, client-specific, flexible to needs.
* **Product-Based:** Market-oriented, feature-stable, focuses on broad audience.
* WHAT is estimation and WHY is done in Software Project Management?
* **Predicting a value for an attribute with a currently unknown value.**
* **It is done to plan and budget and resource prediction.**
* EXPLAIN HOW is the estimation by analogy approach to obtaining effort estimates done?

1. **Look for a past project (or unit) Q that is ‘very similar’ to the new project P**
2. **2. Obtain recorded effort value\* E for Q: Q.E**
3. **3. Identify orthogonal factors accounting for significant differences between Q and P & expected to impact P’s effort**
4. **4. Adjust Q.E to take into account factors identified in P.E**

* NAME and DESCRIBE the limitations of measuring and/or estimating software size in SLOC? WHY do we use it anyway? What are the possible alternatives?
* **Language Dependency: SLOC measurements depend on the programming language used, making it inaccurate for comparing projects in different languages.**
* **Code Quality: SLOC doesn't account for variations in code complexity and quality, making it an unreliable measure of software effort.**
* **Copy-Paste Code: Duplicate or copied code artificially inflates SLOC count without reflecting actual development effort.**
* **Comments and Whitespace: SLOC includes comments and whitespace, which can significantly inflate the size without adding functionality**.

**Why use SLOC anyway:**

SLOC is still used because it provides a simple and widely understood metric that can be helpful for high-level project planning, cost estimation, and historical data analysis.

Alternatives.. function points, use case points, story points (agile) and Cocomo (constructive cost model)

WHAT units can be used to measure the Size of Software? (NAME four).

**function points, use case points, story points (agile) and Cocomo (constructive cost model)**

* CRITICALLY COMPARE the function-point analysis of software with the analysis done using SLOC.
* **Function-Point Analysis vs. SLOC Analysis:**
* **Focus:**
* **Function-Point: Functionality and user interactions.**
* **SLOC: Physical lines of code.**
* **Language Independence:**
* **Function-Point: More language-independent.**
* **SLOC: Language-dependent.**
* **Complexity:**
* **Function-Point: Considers functionality and complexity.**
* **SLOC: Primarily measures code quantity.**
* **Accuracy:**
* **Function-Point: More accurate for sizing and effort estimation.**
* **SLOC: Can be less accurate due to code variations.**
* **Use Cases:**
* **Function-Point: Requirements analysis and user-centric sizing.**
* **SLOC: Simple metrics and project planning.**
* **Agility:**
* **Function-Point: Adaptable to changing requirements and suitable for Agile.**
* **SLOC: Less adaptable to changes.**
* EXPLAIN the difference between effort and elapsed time **Effort:**
* Amount of work or labor hours.
* Measured in hours or person-days/months.
* Focuses on work put in task without considering when.
* **Example:** If it takes a developer 40 hours to complete a software module, the effort expended is 40 hours regardless of how long it takes to finish.
* **Elapsed Time (Duration):**
* Total calendar time from start to finish.
* Measured in hours, days, weeks.
* Considers total time, including delays.
* **Example:** If a software module is started on Monday and completed on Thursday, the elapsed time is four days, even if the developer worked only 40 hours during that period.

# Risk Management

* EXPLAIN WHAT are the top 10 software development risks according to Barry Boehm? Also, explain the risk reduction techniques used for each of the 10 risk categories.
* NAME the benefits of risk planning.

**Risk Identification: Recognizing potential risks that could hinder project success.**

**Risk Analysis and Prioritization: Evaluating and ranking the identified risks based on their potential impact and likelihood.**

**Risk Planning: Developing plans to mitigate or eliminate the identified risks. This step involves creating strategies to reduce the threats posed by these risks.**

**Risk Monitoring: Continuously overseeing the risks throughout the project lifecycle.**

* EXPLAIN WHY is risk planning not widely used?

**• Lack of awareness of the approach**

**• Unwillingness to spend additional time and resources on risk management**

**• Development managers may want projects to go ahead and do not want project sponsors to be deterred by consideration of possible failure**

**• If successful, you might not experience any tangible benefit, in spite of there being a cost associated with its use (which is actually a risk)**

* Briefly DESCRIBE 4 mitigation strategies to get a given software project back on track when the project has experienced delays due to unexpected events.

-**Renegotiate the deadline**

**– Shorten critical path**

**– Reconsider activity dependencies**

* **Reduce scope**
* EXPLAIN the steps to follow while planning for risk? What does each step cover?
* **Risk Identification: Recognizing potential risks that could hinder project success.**
* **Risk Analysis and Prioritization: Evaluating and ranking the identified risks based on their potential impact and likelihood.**
* **Risk Planning: Developing plans to mitigate or eliminate the identified risks. This step involves creating strategies to reduce the threats posed by these risks.**
* **Risk Monitoring: Continuously overseeing the risks throughout the project lifecycle.**
* EXPLAIN WHAT is risk exposure? What is risk reduction leverage?

**Risk Exposure**: **Risk exposure refers to the potential impact of a risk on a project. risk exposure = (potential damage) X (probability of occurrence).**

**Risk Reduction Leverage (RRL): Risk reduction leverage is a metric used to assess the cost-effectiveness of risk reduction actions. It is calculated using the formula: RRL=RE before−RE aftercost of risk reductionRRL=cost of risk reduction**

**RE before−RE after​, where 'RE before' is the risk exposure before risk reduction actions are taken, and 'RE after' is the risk exposure after the actions. An RRL value above 1.00 indicates that the benefits of the risk reduction (in terms of reduced risk exposure) outweigh the costs involved in implementing the risk reduction measures.**

* DEFINE risk and risk management in the context of software projects.
* **PRINCE2, another project management standard, defines risk as "the chance of exposure to the adverse consequences of future events.**
* **Risk management: assessing and prioritizing risks and drawing up plans for addressing**

**those risks before they become problems**

WHAT are the categories of risk?

**Actors, Structure, Technology and Tasks**

* WHAT are the approaches to identify risks?
* **Brainstorming: This is a group discussion technique where team members generate ideas and potential risks. It encourages open and creative thinking and can uncover risks that might not be immediately apparent.**
* **Checklists: Using checklists based on past projects or industry standards can help identify common or recurring risks. These checklists are often developed from historical data and lessons learned from previous projects.**
* **Interviews: Conducting interviews with experienced project members, stakeholders, or experts can reveal risks based on their past experiences and insights.**
* **Delphi Technique: A structured communication technique, often used as a way to reach a consensus among a panel of experts. Experts answer questionnaires in multiple rounds, and after each round, a facilitator provides an anonymous summary of the experts' forecasts and reasons. This process continues until a consensus is reached.**
* DESCRIBE the FOUR categories of risk in software projects

**Actors, Structure, Technology and tasks.**

* IDENTIFY FOUR risks associated with a final-year undergraduate project whose objective is to develop a web site for a cricket club. Assess the likely probability level and impact level of EACH of these risks AND draw up a probability-impact grid showing the risks that you identified.

**Technical Challenges: Difficulty in implementing certain web functionalities or integrating third-party services.**

**Probability: Medium (Depends on technical proficiency)**

**Impact: High (Could significantly delay or degrade the project)**

**Scope Creep: The project's requirements may change or expand over time, especially if the club's needs are not well-defined initially.**

**Probability: High (Common in student projects)**

**Impact: Medium (Can lead to increased workload and time)**

**Time Management Issues: Balancing project work with other academic responsibilities can be challenging.**

**Probability: High (Common in academic settings)**

**Impact: Medium (Could lead to rushed work or missed deadlines)**

**Data Security Risks: Ensuring the security of the website, especially if it handles member information or online transactions.**

**Probability: Medium (Depends on the complexity of the website)**

**Impact: High (Security breaches can have serious consequences)**

* **IDENTIFY TWO risks in your final year project. How have you deal with them? (THREE lines max per risk).**

# Project Monitoring and Control

* WHY are project monitoring and control important?

**To ensure that the project progresses according to the plan – monitor what is happening and compare actual achievement against the schedule**

DESCRIBE 5 strategies for getting a given software project back on track when the project has experienced delays due to unexpected events.

* **Reduce Scope: Scaling back the project's scope to focus on essential features can help realign the project with available resources and time.**
* **Shorten the Critical Path: Analyze the project's critical path and identify tasks that can be expedited or performed in parallel to reduce the overall project duration.**
* **Reconsider Project Deadlines: Extending the project timeline can provide additional time to complete tasks without compromising quality.**
* WHAT is exception planning?
* **Exception planning… project managers are allowed to change major aspects of a project plan that they deem is necessary as long as outcome is done on time and still within budget.**
* **This can affect users e.g reduction in project scope and business case.. e.g increased costs which will reduce the potential profit of software product.**
* **Hence we need an EXCEPTION REPORT by the project manager where they justify ways we need to deviate from the original plan of the project.**
* EXPLAIN the role of checkpoints and milestones in a software project. Agile Methodologies: Scrum and Kanban
* **Point in project where progress is checked. Checkpoints ensure that intermediate products are compatible. Milestone: When a particular achievement has occurred at some point the project lifecycle. In Agile, checkpoints often take the form of regular meetings like daily stand-ups (in Scrum) or continuous monitoring of the Kanban board.**
* **In Scrum, milestones might coincide with the end of sprints, where a potentially shippable product increment is delivered. In Kanban, milestones may be less rigid but associated with the completion of significant features..**
* COMPARE “big-design-up-front” methodologies against “agile” methodologies.
* **Rigidity (BDUF) vs. Flexibility (Agile): BDUF (e.g., Waterfall) is rigid and follows a strict sequence, while Agile is flexible and adapts to changes quickly.**
* **Controlled (BDUF) vs. Adaptive (Agile): BDUF (e.g., Change Control, Predictive) is controlled and planned, whereas Agile is adaptive, responding to evolving requirements and customer feedback.**
* **Thorough (BDUF) vs. Simplified (Agile): BDUF emphasizes extensive documentation and comprehensive planning, while Agile simplifies processes, focusing on the essentials like user stories.**
* **Predictable (BDUF) vs. Dynamic (Agile): BDUF aims for predictability in project outcomes through upfront planning, while Agile is dynamic, embracing changes and continuous adaptation.**
* EXPLAIN the core values and principles of the Agile Manifesto.

A purple rectangular boxes with text

Description automatically generated

* DEFINE the high-level structure of the Scrum Framework.

A white text on a white background

Description automatically generated

* EXPLAIN the pillars and values of Scrum.

1.Commitment to achieving the goals and supporting each other

2. Focus on the work of the Sprint to make the best possible progress toward the sprint goal

3. Openness about the work and the challenges you run into

4. Respect each other to be capable, independent people

5. Courage to do the right thing and work on tough problems

A scrum with text on it

Description automatically generated

* EXPLAIN the structure of the Scrum Team
* **Product owner – Possibly a Product Manager / Project Sponsor / Key end-user – Decides features, release date, prioritisation, budget**

**• Scrum Master – Typically a Project Manager or Team Leader – Responsible for enacting Scrum values and practices – Remove impediments / politics, keeps everyone productive**

**• Project Team – 3-10 members; Teams are self-organising – Cross-functional: QA, Programmers, UI Designers, etc. – Membership should change only between sprints**

* EXPLAIN the Scrum events (sprint planning, daily standups, sprint reviews, and sprint retrospectives).
* **Sprint Planning: This is where the team decides what to work on during the sprint. The Product Owner presents the top items on the product backlog, and the team selects tasks they can complete during the sprint, turning them into a sprint backlog.**
* **Daily Standups: These are short, daily meetings (usually 15 minutes) where each team member discusses what they did the day before, what they will do today, and any blockers they are facing. It's a quick way to coordinate efforts and identify issues.**
* **Sprint Review: At the end of each sprint, the team presents what they have accomplished during the sprint. This is an informal meeting, with the key stakeholders, where the team demonstrates the new features and the Product Owner reviews the sprint’s progress against the goals.**
* **Sprint Retrospective: This occurs after the sprint review and before the next sprint planning. The team discusses what went well, what didn't, and how they can improve in the next sprint. It's a moment for reflection and continuous process improvement.**
* COMPARE Product and Sprint Backlogs.

**Product Backlog:**

* **Long-term view: Lists all desired features, enhancements, and fixes for the product.**
* **Managed by Product Owner: Reflects the product vision and prioritizes items based on business needs.**
* **Dynamic and evolving: Continuously updated based on feedback, market changes, and stakeholder input.**
* **Broad scope: Covers a wide range of potential updates and improvements for the entire product lifecycle.**

**Sprint Backlog:**

* **Short-term focus: Contains items selected for completion in the upcoming sprint.**
* **Managed by Development Team: Focuses on what is needed to meet sprint goals.**
* **More fixed during sprint: Once the sprint starts, changes are minimized to maintain focus.**
* **Specific scope: Only includes tasks and user stories that are planned for the current sprint.**
* COMPARE Definition of Ready and Definition of Done in Scrum

1. **Scrum's definition of "Done" refers to a shared understanding within the Scrum Team of what it means for work to be complete on a product increment. This definition is crucial as it ensures transparency and quality in the deliverables.**
2. **In Scrum, "Ready" refers to a state where a product backlog item (usually a user story) is well-defined, clarified, and prepared for inclusion in a sprint.**

* COMPARE epics, stories, and tasks in Scrum
* **Epics**
* **High-level requirements**
* **Large, complex pieces of work**
* **Represent significant portions of a project**

1. **Stories**

* **Detailed, user-centric requirements**
* **Smaller, manageable units of work**
* **Typically represent a single feature or functionality**

1. **Tasks**

* **Sub-divisions of user stories**
* **Break down user stories into specific actions**
* **Assigned to team members for implementation**
* EXPLAIN how to monitor Scrum sprints.

1. **Daily standups.**
2. **Sprint backlog.**
3. **Regular feedback**
4. **retrospectives.**
5. **Sprint review meeting.**

* EXPLAIN the four high-level steps of Kanban.
* EXPLAIN Disadvantages and Advantages of Scrum.
* **Facilitates rapid development of cutting-edge projects.**
* **Adapts easily to changes with short sprints and feedback.**
* **Boosts team productivity with daily meetings and minimal overhead costs.**
* **Team member departure can disrupt projects.**
* **Requires experienced and committed team members for success.**
* **Lack of commitment can lead to project failure or non-completion**.
* EXPLAIN Disadvantages and Advantages of Waterfall.

1. **Simple and easy to understand and use**
2. **Easy to manage due to rigidity**
3. **Process and results are well documented.**
4. **No working software is produced until late during the life cycle**
5. **High amounts of risk and uncertainty**
6. **Not a good model for complex and object-oriented projects**

* EXPLAIN when to use Scrum, Kanban, or StepWise/PRINCE2.
* **Scrum:**

**Use Scrum for dynamic projects with evolving requirements.**

* **Kanban:**

**Adopt Kanban for projects with continuous, variable workloads**

* **StepWise/PRINCE2:**

**Choose StepWise/PRINCE2 for large, complex projects with a structured approach.**

* **CRITICALLY COMPARE Scrum vs Kanban vs StepWise/PRINCE2.**

**Scrum:**

1. **Best suited for projects with evolving requirements and a need for adaptability.**
2. **Promotes frequent collaboration and iterative development but may lack formal structure and documentation.**

**Kanban:**

1. **Ideal for projects with continuous, variable workloads, focusing on workflow optimization.**
2. **Offers flexibility but may lack defined timeframes and sprint-based planning compared to Scrum.**

**StepWise/PRINCE2:**

1. **Suited for large, complex projects that require detailed planning, formal processes, and governance.**
2. **Provides a structured approach but can be rigid and less agile compared to Scrum and Kanban.**

Top of Form

* PROVIDE AN EXAMPLE of applying Scrum/Kanban to your FYP SMART goals and objectives
* WRITE a short paragraph (MAX 6-8 lines) describing the goal and context of your Final Year Project. Then, DESCRIBE at least THREE SMART objectives of this project, being careful to explain each of the 5 properties denoted by the SMART acronym but paying particular attention to the property denoted by “M” (in other words how will you check or know that you have succeeded with that objective?)

Note: In addition to the questions above, do carefully prepare all the tutorials (remember examples of solution have been published on blackboard). Specifically, take special care while preparing exercise problems that cover the following:

* + Unit 3: Software project planning:
    - Effort estimation: Tutorials 1, 2 and 3. You should know the formulas effort estimation given in class, such as effort = system\_size / productivity.
    - Activity planning: how to construct and use activity networks.
    - Critical path: how to identify it and make analysis based on it (Tutorial 4: planning)
  + Unit 4: Risk management:
    - Risk analysis and prioritization techniques
    - Risk identification
    - Calculation of the risk exposure factor
    - Application of the Program Evaluation and Review Technique (PERT), including the use of the table with z values.
    - Tutorial 5 and 6: Risk management.
  + Unit 7 : Agile methodologies:
    - Tutorial 7