

1 List two key reasons that have caused both software project successes and failures. Explain why.

Two of the most important reasons for project success are clear requirement statements and proper planning. If there are clear requirement statements, individuals will know what is needed from each one of them and they will deliver while meeting expectations. If there is a clear plan from the beginning, individuals will know when each requirement should be submitted, which, in turn, will make following deadlines and following the schedule much easier.

Two of the most important reasons for project failure are lack of user input and incomplete requirements and specifications. When there is a lack of user input, the engineers will not know what should be modified how because the user will be no feedback system. When there is an incomplete requirements and specifications list, the developers will have a vague idea about the functionalities, causing every individual to produce something that may not work together when integrated or something that is not what is required (needed)

2. From the definition of software engineering given by the textbook authors, list three areas that software engineering must touch on. Prioritize the three areas. Explain your prioritization.

Software engineering: A broad field that touches upon all aspects of developing and supporting a software system:

1. Technical and business processes
2. Specific methodologies and techniques
3. People, skills, and teamwork

In the beginning finances need to be set up. Necessary calculations are made to ensure that everyone is getting paid. After that, what needs to be known is how should we do what needs to be done, as a team, while not exceeding the budget set in the first area. After that, people who possess the necessary skills should be gathered and implemented as a team do what needs to be done the way it should be done, while staying within the budget limits.

3. List two of the three strategies cited by the 2004 US General Accounting Office report as key to ensuring delivery of successful software. Explain why.

- 1- Attention needs to be more focused on the software development environments
- 2- Disciplined development process

Since the DOD is an acquisition office, its acquisition managers need to be able to recognize signs of successful software organizations from which they source their software. To do that, they must be able to differentiate those that are practicing good software engineering from those that are not

4. Which of the possible interpretations of term “quality” in Davis’s first principle do you think is most important? Explain your answer.

Correct functionalities. I believe that if the program does not do what it is supposed to do (has incorrect functionalities) then the program not only low-quality, but also useless. Therefore, having a program that functions properly, does what it is supposed to do and has the right functionalities, then we have created a high-quality program.

5. Trace the three sets of software engineering principles of Davis, Royce, and Wasserman and discuss the following software engineering topics in terms of relevance and emphasis placed by them

Requirements:

Davis: Quality is the number 1, high-quality software is possible, give products to customers early, determine the problem before writing the requirements

Royce: base the process on an architecture-first approach

Wasserman: Abstraction

Architecture and design:

Davis: Evaluate design alternatives

Royce: Establish an iterative process that addresses risks early in the process, Emphasize component-based development to reduce the coding effort.

Wasserman: Analysis and design methods and notation

Process:

Davis: Use an appropriate process model

Royce: Use model-based and machine-processable notation to capture design

Wasserman: Life cycle and process

Tools:

Davis: Put technique before tools, Use different languages for different phases, Good management is more important than good technology, People are the key to success,

Royce: use a demonstration-based approach where intermediate artifacts are transitioned to executable demonstration of the user scenario so that these artifacts can be assessed earlier

Wasserman: Tools and integrated environment

Metrics and measurement

Davis: Inspect code

Royse: High-quality software is possible, Give products to customers early

Wasserman: Metrics

Testing and quality

Davis:

Royse: Plan to have incremental releases, each composed of a group of usage scenarios, with evolving levels of detail.

Wasserman: none

6. Discuss one advantage and one disadvantage of the waterfall process

Advantage: it provides a clear way to track the stages of development for the project managers.

Disadvantage: it provides very little task overlapping. It is sequential and a single iteration.

7. What is the goal of a software process model?

The goal of a software process model is to provide guidance for systematically coordinating and controlling the tasks that must be performed in order to achieve the end product and the project objectives.

8. What are the four quadrants of a spiral model? Trace the requirements set of activities through each quadrant.

- 1- Plan next phase
- 2- Determine objectives, alternatives, constraints
- 3- Evaluate alternatives, identify, resolve risks.
- 4- Develop, verify next-level product
- 5- The requirement plan is done during the “plan next” and “determine objectives, alternatives, constraints” phases. The requirements prototype is done during the “evaluate alternatives, identify, resolve risks” phase. The requirements specification is done during the “develop, verify next-level product” phase.

9. What are the entry and exit criteria of a process? Give two entry criteria examples and discuss their importance. Give two exit criteria examples and discuss their importance.

Entry criteria: describes the conditions, that must be met, before initiating the activity.

An example of entry criteria is a list of the required people. The people required to perform the task must be specified. They must be available before the beginning of the task. It is important because if those people are not available to do the task, then no one else would do it at the moment.

Another example of entry criteria is the definition (clear explanation) of the task itself. It is important because if there is no clear and consistent understanding among the individuals on what needs to be done, then there would be an extremely high chance that every individual would perform the task differently, causing unpredictable, irregular, and false results.

Exit criteria: describe the conditions, that must be met, before terminating an activity (for the activity to be considered complete).

An example of exit criteria is for all the artifacts to be review. This is important because the exit criteria are supposed to describe the artifacts that must be available for the next activity. If the artifacts are not reviewed, then false artifacts might be what is used for the next activity.

Another example of the exit criteria is all the prespecified percentage of errors are corrected. This is important because the if the prespecified percentage of errors are not corrected, then those errors will be included in the next activity and will most likely cause more errors.

10. What motivated software engineers to move from the waterfall model to the incremental or spiral model?

There was a need to develop multiple components, overlap the development, the need to reduce risks as much as possible. All these needs led to moving from the waterfall model to the incremental model.

11. What are the major concepts that drove the RUP framework? What is the relationship of the four phases to the development activities such as requirements analysis, design, and testing?

The RUP is driven by 3 major concepts: architecture centric, use-case and requirement driven, iterative and incremental.

The 4 phases: inception, elaboration, construction, and transition

The RUP phases are not named after activities such as coding, design, or testing because an iteration may include numerous activities in various degrees.

12. List all of the key processes addressed by SEI's CMM model. Which ones are required for maturity level 2?

There are 5 levels of the original Capability Maturity Model (CMM): initial, repeatable, defined, managed, and optimizing.

The key processes addressed by SEI's CMM model:

Organizational Process Performance

Quantitative Project Management

Organizational Innovation and Deployment
Causal Analysis and Resolution
Requirements Management
Project Planning
Project Monitoring and Control
Supplier Agreement Management
Measurement and Analysis
Process and Product Quality Assurance
Configuration Management
Requirements Development
Technical Solution
Product Integration
Verification
Validation
Organizational Process Focus
Organizational Process Definition
Organizational Training
Integrated Project Mgmt (with IPPD extras)
Risk Management
Decision Analysis and Resolution
Integrated Teaming (IPPD only)
Org. Environment for Integration (IPPD only)
Integrated Supplier Management (SS only)
Please find below Key Process Area for maturity level-2

Requirements Management
Project Planning
Project Monitoring and Control

Supplier Agreement Management

Measurement and Analysis

Process and Product Quality Assurance

Configuration Management

The ones required for maturity level 2 are:

Requirements management

Project planning

Project monitoring and control

Supplier agreement management

Measurement and analysis

Process and product quality assurance

Configuration management

13. What is the difference between the multiple component incremental model and the multiple release incremental model?

Multiple component incremental model: increment the functionality of a particular model. After testing, when a product is released by an organization and if a new functionality is needed to be introduced after the release, the new functionality is then implemented and added in a newer (updated) version of the release. After complete testing, the next version (newer model) is then released.

In multiple release incremental model: as we release the product then we again add functionality and bug fixing (if any) of previous release. After complete testing we release particular product and so on... that is called multiple release incremental model as release increases as well as functionality implemented as a new component.