

Software Engineering, ESOF 322, Fall 2019
Exam 1, Sept. 30

Multiple Choice

1. “When a valid report request is submitted, the report shall be displayed within 3 seconds” is most likely to be categorized as what type of requirement?
(4 pts.)
 - a. Functional requirement
 - b. Project requirement
 - c. Design constraint
 - d. **Quality attribute**
 - e. All of the above

2. “The system shall be web-based using a RESTful architecture” is most likely to be categorized as what type of requirement?
(4 pts.)
 - a. Functional requirement
 - b. Project requirement
 - c. **Design constraint**
 - d. Quality attribute
 - e. All of the above

3. What is the most likely response when asked about an early process model, that was once the most popular agile process model, and formed the foundation for many of the more recent agile processes?
(4 pts.)
 - a. waterfall
 - b. incremental
 - c. **XP**
 - d. Kaban
 - e. Scrum

4. WordPress allows users to build websites, create a blog, a business site, a portfolio, or an online store. WordPress is most likely to be categorized as what type of service?
(4 pts.)
- a. Infrastructure as a service
 - b. Platform as a service
 - c. Cloud application services
 - d. None of the above
 - e. All of the above
5. Unit testing is most likely to be categorized as what type of testing?
(4 pts.)
- a. Whitebox
 - b. Blackbox
 - c. Neither
 - d. Both
6. The goal of CMM is mostly likely to be described as which of the following?
(4 pts.)
- a. Helps a software organization define its software development process
 - b. Helps a software organization promote continuous improvement
 - c. Helps a software organization assess how well they are adhering to a process model
 - d. Can be used as a marketing tool for software organizations
 - e. All of the above

Acronyms

7. Expand the acronym CMM. (4pts.)

Capability Maturity Model

8. Expand the acronym REST. (4pts.)

REpresentational State Transfer

9. Expand the acronym XP. (4 pts.)

eXtreme Programming

10. Expand the acronym RUP. (4 pts.)

Rational Unified Process

Short answer

11. Describe the role of a moderator in the code inspection process that we used in class, and plan to use during Sprint 1. Include the activities of the moderator before the inspection meeting, during the inspection meeting, and after the inspection meeting.

(10 pts.)

The moderator is to facilitate and oversee the inspection.

Before the meeting:

- Talk to the author to determine if the work product is far enough along to warrant an inspection. Particularly, has adequate attention has been given to the SRS, SDD, test suite, database and code.
- Sets a time and date for the inspection and invites people to it at least 24 hours in advance.

During the meeting:

- Is in charge of the meeting
- Facilitates the meeting
- Determines if enough preparation time has occurred to move forward with the meeting
- Makes sure that all defects are clearly recorded

After the meeting:

- Verifies the author's rework and, if ok, closes the merge request and removes the branch

Essays

12. List the four Agile principles. For each principle, discuss how this differs from non-agile process models in addressing the software crisis. (20 pts.)

Clarity, grammar, spelling, organization	(5 pts.)
Agile principles	(10 pts.)
Discussion of how the principles differ from non-agile processes	(5 pts.)

Individuals and interactions over processes and tools

Before agile defining an excellent process and purchasing or developing tools was considered a good way to address the software crisis. However, the developers of agile feel that clients, users, system architects, designers, developers, testers working together, in frequent and efficient interactions is the best way to address the software crisis.

Working software over comprehensive documentation

Before agile documenting requirements, architectures, designs, code, bugs and bug fixes was considered a good way to address the software crisis. However, the developers of agile feel that releasing working software to users and clients is the best way to address the software crisis.

Customer collaboration over contract negotiation

Before agile a complete, unambiguous, consistent Software Requirements Specification (SRS) was considered a good way to address the software crisis. However, the developers of agile feel frequent interaction with the clients and users, determining the most useful system is the best way to address the software crisis.

Responding to change over following a plan

This is similar to the first and third principle. Before agile a comprehensive plan was considered a good way to address the software crisis. However, the developers of agile feel that frequent interaction with the clients and users, responding to their concerns and changes in the environment is the best way to address the software crisis.

13. Describe what is meant by loose coupling and high cohesion and relate these concepts to Tom Killalea's article "Velocity in Software Engineering". (20 pts.)

Clarity, grammar, spelling, organization	(5 pts.)
Definitions of loose coupling and high cohesion	(10 pts.)
Accuracy and completeness of Tom Killalea's article	(10 pts.)
Relation of the two topics	(5 pts.)

Loose coupling refers to lowering the interdependency of software components (objects, modules, components), typically by adhering to a strict interfaces. High cohesion refers to making components focus on a single thing, only performing highly related tasks.

Tom Killalea's "Velocity in Software Engineering" suggests:

Autonomy – have maximum autonomy for teams with a **high degree of cohesion** around a specific responsibility. Teams that have maximum autonomy could be considered loosely coupled to other teams and highly cohesive within the same team.

Speed – have maximum autonomy for teams and the services operated by those teams, adopt REST-style interfaces between highly decoupled components, platform standardization, removal of roadblocks and continuous deployment of isolated components. As above, the maximum autonomy for teams and services could be described as teams and services that are loosely coupled and highly cohesive.

Culture – assume that people are talented, aligned with the mission and want to work at high velocity. I see these people are cohesive with the mission but free (loosely coupled) to work at high velocity.

Focus – avoid gatekeepers, cloud computing is an enabler for permissionless innovation, access controls and compliance assertions are programmatically enforced. Work being loosely coupled, avoids gatekeepers.

Other traits:

Direction – clear identification of who the customers are, then working backward from their needs to a product definition that will meet those needs, pay attention to the voice of the customer, be very good at course correcting.

Agility – balance between course correcting and optimizing for speed. Backlog may change constantly, but the latest version is used for spring planning and the team explicitly commits to a set of tasks which don't change for the sprint.

Enablement – invest in systems that enable engineers to work at speed and to maximum the percentage of their time spent working on their area of unique responsibility. Promotes metric collection and propagation, monitoring, alarming and issue tracking

Experimentation – reduce the cost of experimentation so more can be run. This is important with course correcting.

Extra Credit

Define the term “software engineering” and relate your definition to what is needed for startups and next-generation technology companies. (5 pts.)

There are many definitions of software engineering. I’ll use:

Software engineering – engineering discipline whose focus is on the cost-effective development of high-quality software system.

The article “Software Engineering Grads Lack the Skills Startups Need” by Kathy Pretz says that engineers at startups may have to wear many hats doing everything from designing the system architecture to sales. Since time to market is critical for startups, rather than defining a system architecture, companies may rely on cloud computing so students need a practical understanding of infrastructure architecture design patterns and cloud platform services like compute instances, object storage, and queuing services. Rather than static testing, testing distributed systems, web services and infrastructure resiliency is important. Startups need clean, reusable, scalable code and developers who can conduct and undergo code reviews.

Much of these values seem consist with making software development an engineering discipline. For instance, utilizing cloud services and creating clean, reusable, scalable code. Testing and code reviews is consistent.

Wearing many hats may be counter the idea of making an engineering discipline.