

Midterm Exam Review

Format: Short answer, fill-in-the-blank, problems

Point distribution:

Domain model/use case/requirements writing – 45

Readings – 15

Software process – 20

Software/Architectural Design - 20

Topics

The Scope of Software Engineering and Software Processes

Do you believe that software engineering is “real” engineering? Why or why not.

One widely-believed software development myth is that the only deliverable work product is the working program. Why is this belief untrue?

What would you say to someone who tells you that software engineering creates voluminous and unnecessary documentation and will invariably slow down the project.

Explain the difference between a waterfall process and the Unified Process for software development.

What is the cause of the biggest problems with a waterfall approach?

Discuss the differences and similarities between the waterfall process model and the Unified Process. (Refer to your class notes. You should be able to answer this question in about 3 sentences.)

Name the phases of the Unified Process and briefly describe the purpose of each.

In your own words, describe the workflows and sequence of activities that are common to every software development process model. (You might need a paragraph or two to answer this one.)

What are the main ideas behind the Agile Manifesto?

The Agile values of are:

Individuals and interactions valued over _____
Working software valued over _____
Customer collaboration valued over _____
Responding to change valued over _____.

Provide an example of a catastrophic software failure. How do you think the catastrophe might have been avoided?

The famous Standish CHAOS Report states that somewhere around 1/3 of software projects fail. Do you think this is a fair statement? Explain.

What are the names of the CMMI maturity levels? How does an organization move from one level to the next?

If an organization meets 50% of Level 5 goals, 80% of Level 4 goals, 90% of Level 3 goals, and 95% of level 2 goals, at which CMMI level is the organization's software development process?

What are the advantages and disadvantages to instilling CMMI best practices in an organization?

What is the cost-benefit ratio for process improvement?

How might CMMI maturity levels be misused?

The Requirements Discipline

Why does a software engineer care about the business context in which the system under development will be used?

From a use case modeling perspective, explain the difference between a user of the system (or participant in a business process) and an actor.

Describe the activities that take place within these key components of requirements engineering:

Elicitation	Specification	Management
Validation	Analysis	

What are SMART requirements?

Of structured interviews and direct observation which will provide you with a better idea of what your software needs to do? Why?

What are the risks of using prototyping to elicit requirements?

What mistakes do people commonly make when they start the requirements elicitation activity?

The following statement was given as a functional requirement. What is wrong with it?

```
The system shall provide a robust security mechanism to
ensure controls over what information can be seen by whom.
```

Non-functional requirements are sometimes called system qualities, why do you think this is? Can you think of a non-functional requirement that is not a system quality?

Name and define three types of non-functional requirements.

Why has performance (or execution efficiency) become such an issue for cloud computing?

Which system qualities does cloud computing provide good support for? Why?

What are some non-functional requirements that may conflict with each other? Discuss how you would try to resolve these conflicts.

Under what standard non-functional requirement does Section 508 compliance fall? On what part of a system does it have the most impact?

We say that non-functional requirements drive the architecture of the system. Why?

How can you use a good set of functional requirements to manage the project schedule?

What is requirements traceability? What is its benefit to a development project?

What is the difference between requirements verification and requirements validation?

The Design Discipline

What are the components of Krutchen's 4+1 view?

How does Kriutchen's model differ from Rozanski and Wood?

Name ten different architectural styles and the uses of each. The Model-View-Controller (MVC), Pipe and Filter, Layered System are particularly important.

What is cohesion?

What is coupling?

List seven types of module coupling and describe their characteristics.

List five different types of coupling and describe their characteristics.

What system qualities are enhanced through modules exhibiting low coupling and high functional cohesion? (Think about the Boehm, McCall, or ISO 9150 models)

If you have trouble coming up with a name for your module, it is an indication that what design attribute might be lacking?

What are the SOLID design principles?

How do the ideas of coupling, cohesion, and SOLID design principles address the ISO 9150 software qualities (non-functional requirements) of maintainability? Do you think they might work against efficiency and portability?

Does the Interface Segregation Principle have more to do with reducing coupling or maximizing functional cohesion? Explain.

What are design patterns? Provide examples of creational, structural, and behavioral design patterns. (You are not responsible for knowing any patterns that were not discussed in class.)

What is an anti-pattern? (You are not responsible for knowing any anti-patterns that were not discussed in class.)

Supplemental Readings

What is technical debt? Is technical debt always paid at some point? If not, why not?

Is it wise to sometimes knowingly take on technical debt? When would you do this?

In Brook's "No Silver Bullet" paper, he speaks of the "essence" and "accidents" of software engineering complexity. What is the difference between these two? What "accidental" complexity are we dealing with today? Has the essential complexity changed at all since the article was written?

Regarding Wiegers's Requirements Trap #1: Confusion Over "Requirements" What are the different types of requirements that Wiegers describes? Are they all necessary for you to effectively build the system the user wants?

Explain Wiegers's Requirements Trap #2: Inadequate Customer Involvement. Why do you think this is a common problem? What can you do about it if you have no administrative control over the customer?

Wiegers provides some good advice for guarding against vague and ambiguous requirements. What are they?

Based on the required reading and discussions in class, what do you think is the biggest obstacle to creating a "perfect" set of requirements? What do you think is the biggest problem in requirements engineering? What can be done about it? Explain.