CSE583: Mid-Sem Exam

Answers for grading – document not to be shared with students – I will discuss these in class.

Grading Policy: Be liberal – the answers given here are not the only answers and other possibilities exist. As long as the answer is reasonable, give marks. When in doubt – err towards giving marks. Goal of the exam was to make them study – so we need to just ensure that the answers are reasonable.

1. (3) List the key differences (<=3) between industrial strength software and student software.

Student software is poor quality – bugs, poor reliability, poor UI, ...
No external user depends on it – has implications
It is short lasting – till the demo time
Often done alone
Developed in an ad hoc manner

Rubric: If three of these (or other such attribute) mentioned – give 3; if two reasonable points mentioned, give 2; if one mentioned – give 1.

2. (3) What are the most common defects (top 3 categories) in a requirements specification. Suggest one effective approach for identifying requirement defects, and v. briefly describe it.

Omission – forgot some requirements
Ambiguity – requirements are not precise / clear / have different interpretations / cannot be verified
Inconsistency – different requirements are not consistent with each other
Missing fact – some important facts are missing

Rubric: Any three, give full. Any two – give 2....

3. (3) What are the top-level strategies for improving productivity in a project for developing an application. Briefly describe how it improves productivity, and what will its impact be on quality.

Using good processes and tools – by having good processes rework can be avoided and bugs can be detected early which can lead to high P and Q

Reuse libraries and frameworks and components – reuse can reduce the work considerably and so improve P; as these components are well tested and are of high quality, quality improves

Use generative AI – to generate code or any other artifact (test cases, design, etc). This will clearly improve P; it can also improve Q as many LLMs have been trained on very good examples

Rubric: 1 point for each

4. (3) List the main components of a good SRS for developing an application and a brief description of what each of them specifies.

Overview – which gives user categories, goals of the application, etc
Functional requirements – what the system has to do – the functionality
Performance requirements
Design constraints
External interfaces

(Sometimes the last three can be combined into Non-functional requirements – like Performance, reliability, security, UI, constraints, scalability etc)

Rubric: If they mention functional requirements, and some non-functional requirements (performance, design constraints, ...) full marks.

5. (3) What are the three most important software quality characteristics – give one line description of each. How can quality of a software.be quantified.

Functionality – does it provide all the key functions as desired
Reliability – how often does the software application fail to perform its task; goal is to have a long mean time to failure for high reliability
Maintainability – as maintenance cost is very high, this helps reduce the cost over the full life of sw
Usability – how good is the user interface
Performance – is the performance acceptable / good

Interoperability – can it work on different platforms and connect with other applications
Scalability
Security and privacy

Rubric: 2 marks if they mention any two of these

Sw quality can be quantified as defects / KLOC, i.e. number of defects per KLOC of delivered code.

Rubric: 1 mark if they specify this

6. (2) What are the two main software license types for source code. Briefly describe each.

Copy left / restrictive: In this if any software is built using the licensed software, it must also be put as open source with same license.

Permissive / MIT: In this the source code can be used by anyone in any manner, including commercial goals; *only the copyright has to be acknowledged*

Rubric: 1 mark for each

7. (3) What is the relationship between the waterfall model and the iterative development model. Which are the main shortcomings of waterfall that iterative tries to address. What is the main drawback of iterative.

Iterative development is essentially a series of waterfalls. Using this approach it counters the main drawbacks of waterfall: (1) all or nothing approach – in iterative a working application is delivered after each iteration, (2) all requirements must be known/frozen at the start – new requirements can be added after each iteration

Rubric: Two marks for mentioning something like these two

Main drawback of iterative – some older decisions may need to be revised and software may need to change to accommodate new requirements, which can cause extra rework.

Rubric: 1 mark for this.

8. (5) A new project suggestion has come to a company in the business of developing applications. It was estimated that this project will require 20KLOC of java code to be written by the team. Estimate the total effort this project may take, the time the project will take, and the team size needed. Before doing the estimation, clearly list all the assumptions you are making (which must be reasonable).

Assumption: (i) Productivity of the team is 2KLOC per person month. (ii) total duration of the project can be estimated as square root of effort. (They may use an equation E = a S**b - in that case a is 1/productivity and can be around 1/ 1-3 KLOC/PM, and b should be around 1.0)

Effort. The project will require 10 Person-months (20KLOC / 2KLOC per PM)

Duration and team size. Using the square root rule, a team size of 3 will be good for this project. This team will take 10/3 = 3.3 months to complete. (A team size of 4 is also OK – this will take 2.5 months)

Rubric: 2 marks for Assumptions, 1 mark for effort, 2 marks for Duration/Team-size

9. (5) You are designated as a team leader for a project which has a team of 4 people. Write a concise team charter with most important values and practices (<= 3 for each). For each item mention the most relevant teamworking principle that it supports.

Team working principles are: Humility, Respect, Trust

Values:

1. We will join each meeting on time (Respect)
2. No blame game – team takes the ownership (Humility, Respect)
3. All of us will be honest and transparent about our work (Humility, Trust)
4. We will take everyone's input for all major decisions (Respect)

Practices:

1. Document the work done (Respect)
2. Complete the work within deadline (Respect, Trust)
3. Respect everyone's opinions (Respect)
4. Meet regularly (Trust)
5. Do not be protective about your code (Humility)
6. Accept others suggestions on your output (Humility, Respect)

Rubric: Give 2 marks for mentioning at least two reasonable values and 2 or practices. 1 mark for mentioning the Team working principles (give mark whichever of the three they mention)

10. (5) You have a team of 2 people (P1 – you, P2) to execute one complete iteration for an application development. In this iteration, there are two main FE components to be added – FE1 and FE2, each taking 1 person days, two main BE components to be added – BE1 and BE2 taking 2 and 4 person days respectively. All the FE / BE modules have no dependencies between them and can be developed in parallel. Make a plan for executing this iteration and delivering the application. The plan should be given as a table listing the tasks (<=6), and for each task give the starting day, the end day, who it is assigned to, and effort. (Assume that first day of project as day 1; a task that takes 1 person-day can have a start day of 1 (say) and end day will also be 1 – for a total effort of 1 person-day).

	Task	Assigned to	Start Day	End Day	Effort (p days_
1	Dev FE1	P1	1	1	1
2	Dev FE2	P1	2	2	2
3	Dev BE2	P2	1	4	4
4	Dev BE1	P1	3	4	2
5	Testing	P1, P2	5	6	4 (2*2)
6	Deployment	P1, P2	6	6	2 (1*2)

Rubric: 2 marks for listing the first 4, 2 marks for scheduling it properly, 1 for mentioning the other activities – testing and/or Deployment.

11. (5x3) An application has to be built for students in IIIT-D to do online registration in courses. A professor will specify a course, enrollment limit for it, and pre-requisites for the course. There will be a deadline for registration in a course. Students can register for a course.

    (a) List the use cases in this application in a table specifying the use case name, the actor, brief description, and the entry condition.

    (b) Write the use case for a student registering for a course.