

Question One [total: 25 marks] -MANDATORY

(a) Describe the attributes of good software. List three key-challenges facing software engineering in large or complex software development?

[7 marks]

(b) Describe the common fundamental activities in all software processes.

[4 marks]

Component
System
Acceptance

(c) The testing phase of a software life cycle include a number of test types or activities; describe three of these test types or activities. Order these three types in the likely order of their occurrence in development life cycle.

[6 marks]

(d) Distinguish between requirements elicitation and analysis. Explain a technique that could be used to achieve requirement elicitation.

[8 marks]

Question Two [total: 25 marks]

(b) Explain the shortcomings of the waterfall model. Under what circumstances the waterfall model is recommended for use? Give an example of a domain where the waterfall model is useful.

[6 marks]

(c) Explain what you understand by completeness and consistency of requirements. Give an example of each.

[6 marks]

(d) In system modelling, distinguish between generalisation and composition. Give an example, using UML notations, to illustrate your answer.

[6 marks]

(e) Distinguish between functional and non-functional requirements. Give an example of each, making sure that the non-functional requirement is measurable/verifiable.

[7 marks]

Question Three [Question Total: 20 marks]

Consider the following (part of a) system description.

A user email system includes a module that allows registered users to enter the system. It includes the following activities: 1- to read emails from the registered student mailbox; 2- to write new messages; 3- to send the written new messages to other email addresses; 4- forward received messages to other email addresses.

(a) Given the above requirements, ~~define~~ draw the use cases for the registered user.

draw

diagram

[7 marks]

(b) Given the above activities, draw a class diagram(s) for the registered user.

[7 marks]

(c) Given the above use cases defined in (a), draw a sequence diagram for the "send new messages" use case.

[5 marks]

(d) In use case modelling, distinguish between <<include>> and <<extend>> use case extensions? Illustrate these in your defined use cases.

[6 marks]

END OF QUESTIONS

Final Page

Question 1:-

(A) A good software must deliver the functionality ~~and~~ that is required from the software, in addition to the good performance. Also, a good software must be dependable, maintainable and usable. ~~in~~ Also, it must be reliable and secure.
Deserhu!!

The challenges facing software engineering ~~is~~ are:-

- ① The programs becomes greater with thousands of lines that can be understood by one programmer ~~and so more~~ complexity. In other words: Complexity increases.
 - ② The changes that is occurring in the technologies, ~~and the changes~~ and environments. **!?**
 - ③ Changes in users requirements. **!!**
- 5/7**

(b) ~~All~~ software processes activities are:-

- ① Software Specification: Defining what the program will do, also defining the constraints.
- ② Software Development: In this activity the software is designed and implemented.
- ③ Software Validation: In this activity the software is checked to ensure that it meets with the user ~~req~~ requirements and so checking the validity.
- ④ Software Evolution: In this activity the software can be maintained and upgraded, in addition to meet the new requirements. **4/4**

(c) The testing phase includes:-

① Component - testing:-

Each component of the system is tested independently from the whole system, and a component can be operation, instance, object... etc.

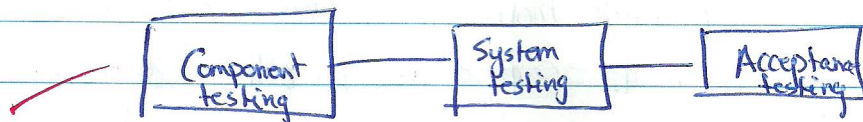
② System Testing:-

The system is tested as a whole, and so ensuring that

System is working well. And so here all components are tested as whole system.

③ Acceptance Testing:-

In this activity, the software will be tested by the ~~end~~ user/client, and to see whether the software meets with the needed requirements or not.



6/6

d) Requirements elicitation is ~~required~~ to get and collect the requirements ~~in order to understand the sys~~ from the clients by meetings, asking them and so on. Whereas, the Analysis means understanding these requirements ~~and~~ ~~and~~ in a good way. Since Analysis in general is understanding the problem and dividing it into pieces to solve them.

One way can be used for requirements elicitation is using scenarios, and a scenario is an actual series of ~~events~~ ^{events} ~~actions~~ describing single feature used by single user. Each scenario has a name in addition to entry condition, exit condition and flow of events. Also there will be any other special Requirements. Group of scenarios is a Use Case model that can be used to describe the functional behavior of the system. Scenarios have different type such as visionary, AS-is, ~~evaluative~~ scenarios.

Also, Requirements elicitation can be done visiting ~~clients~~ user on their working environment, Also by studying the field ~~in~~ that the software Related with.

22/25

7/8

Question 2:-

(A) The waterfall model is the oldest model, but the problem with this model is ~~is~~ that this model can't afford the changes that happens. In other words, the waterfall model is a linear model and so ~~the~~ any phase that is ~~is~~ finished and we went to the next phase we can't return to it back. Furthermore, this model deals with well defined requirements that are understandable. Also, it can be costly because if one phase is done in a wrong way, this affects the whole next phases and so there is a need to ~~redo~~ redo them.

Customer centric

Also, the waterfall model needs time and so it is slow.

The waterfall model is used with large software projects and so it is used by large companies. It can be used in any domain that has large huge software projects. For example, can be used as a model for making manufacturing e.g.

5/6

(B) Completeness: means that the software we have ~~is~~ has full requirements and ~~has~~ contains all possible scenarios that can happen, and so the software is said to be complete. For example, ~~a calculator software~~, ~~with~~ ~~it in it the~~ ~~must be able to do all requirements such as adding, subtracting~~

Consistency: means that all functional and non-functional requirements ~~is~~ ~~don't~~ don't contradict each others and so they are in

implicit & explicit !!

a consistent situation. For example, ~~we need our system to be secure and~~ ~~and we need it to send~~ In any message system, we need it to be secure, Also we need it to send ~~the~~ be able to send a message ??

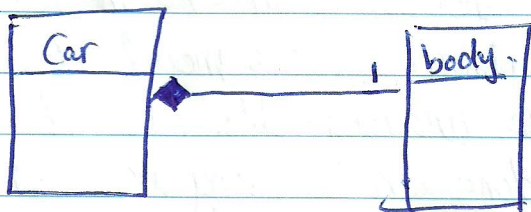
As an example of Completeness, in a message system, sending and receiving message must be done correctly, and solve any exception that will happen and so these different scenarios must be in mind.

completing?

?

5/6

- © A composition relation is a strong ~~self~~ aggregation relationship; which means that ~~relat~~ the ~~object~~ that is aggregated with another entity can't exist, if the ^{entity} original entity is removed and it is also define the (is a part of) in UML

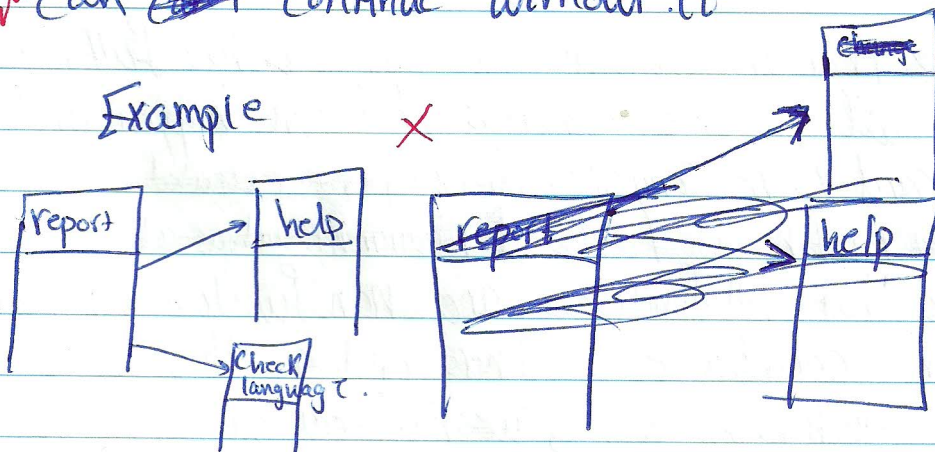


There is nobody without a car.

Generalization: it is used to reduce the flow of events by using external use cases, for example if we have ~~entity~~ called Report, we will ~~not~~ have another use case ^{as a use case} that is called help, this is used for additional information, and system can ~~not~~ continue without it.

Inheritance

Example



3/6

- (d) Functional requirements means the services that the software should provide whereas ~~the~~ nonfunctional requirements are the constraints on the software that comes from the environments and so it is constraints. Non-functional requirements are ~~not~~ usually done as a whole not individually, also in functional requirements it is decided how the system will react with particular inputs and how will it behave in particular situations also, things that the system should n't do can be defined here.

For an example of functional requirement, ~~Ritaj must be able to send messages~~ In Ritaj System, The teacher can send messages to students and vice-versa. ✓

For an example on non-Functional Requirements, A user on Ritaj System which is the website used by Birzeit's University employees and visitors must ~~be able to~~ complete the sending message operation in short time which is 2 minutes with 5% error.

7/7

20/25