Dashboard / My courses / Information Technology / IT303 - 26725 / General / Mid Semester Examination September 2020

Started on	Friday, 25 September 2020, 2:01 PM
State	Finished
Completed on	Friday, 25 September 2020, 4:08 PM
Time taken	2 hours 7 mins
Grade	Not yet graded

Question **1**
Complete
Marked out of 2.00

UML diagramming and design activities are a time to fully and accurately define designs and models in great detail, and of programming as a simple mechanical translation of these into code? Do you agree with this statement? State Reasons if you agree or not?

No

- The unified process is not like the traditional waterfall model where each phase has to be completed before moving to the next.
- In a waterfall model implemenatation comes after the desing phase and there is very little modification of the initial requirements.
- The unified process is divided into phases called inception, elaboration, construction and transition.
- Unified process is based on iterative form of development
- development is organized into iterations and the result of each iteration is an executable system which means in each iteration some coding is done.
- Only a little thinkin**.**g is done about the design using visual modeling using rought **UML diagrams,** the time taken for this depends on the size of the team and then is implemented.
- This happens in each of the iterations.

iterative development approach.jpg

Question **2**
Complete
Marked out of 2.00

If you adopt the UP, you take "long time" doing requirements or design work before programming starts, Why ?

No it is not true that it takes a "long" time doing requirements or design work before programming starts.

Unified process is not like waterfall model, where a thorough investigation is done before one begins programming.

In unified process much of the investigation about design etc happens in the elaboration phase. This includes refining the use-case model, and coming up with the design model etc.

But even in the **inception** phase which is kind of a feasibility phase where the order-of-magnitude costs and business modelling is done, **some programming is also done.** This might be done in order to create "proof of concept" prototypes and to clarify some requirements and do some programming experiments.

programming happens in every phase of the **Unified process** which includes inception, elaboration, construction and transititon.

Question **3**

Complete

Marked out of 40.00

Imagine you have used Unified Process in your SE project (which is the part of this course IT303),

a. Elaborate on the artefacts you generated during Inception Phase? 20 Marks

b. Elaborate on the artefacts you developed during the first Iteration of the Elaboration phase? 20 Marks

a) The artefacts generated using the design phase include

```
1. Vision and Business case
```

- The vision document will contain the idea of our project.
- What our project will do and how the end user will be able to use the service given to him/her by our project.
- For example, our project includes classification of an image as malicious or benign using machine learning techniques.
- So this would be our vision for the project.
- The business case will talk about if the resources put into the project which includes time and effort in our case , or money and investment in larger projects is worth it or not.
- Is there a business case to justify carrying on with the project.

In our case **Business case** would contain the fact that using machine learning techniques can help save money invested in antivirus software and hence is economically worth it.

```
2. Use case model
```

- This contains information about the actors in our machine learning based solution and the use cases present.
- Assuming **malJPEG** our projects name as the system , we can have one primary actor the user of the malJPEG solution who will interact with the system.
- This would also contain all the functional requirements described in the form of use-case diagrams and all the non-functional requirements as well.
- For example the non-functional requirements for our project would be a clear indication of the result of the malJPEG solution.

```
3. Supplementary specification
```

- This document includes all those requirements that do not come under functional or non-functional requirements.This can contain any legal permissions and other legal guidelines to be followed and adhered to.
- This will also contain other information about the performance, security and supportability of the system.
- Othere requirements that included operating system constraints etc are also mentioned in this document.
- Our project won't pose any security or performance issues to the end user.
- While creating the application we will have to train the app with thousands of malicious JPEG images and this might require some legal permission.

```
4. Glossary
```

- The glossary would contain the information related to the domain objects , terms that might be unclear to the reader are made clear in this document.
- It would contain a column containing term, its definition and type.
-

Term	Defintion	Type
User	The user who will input the image	Actor
.		
.		
.		

```
5. Iteration plan
```

- This plan will contain information about what to do in the first elaboration phase

5. Development Case.

- The unified process can be modified to suit the needs of the company or organization or in our case our mini project
- Not all documents would be necessary.
- So this document can include details such as
- 1.which artefacts will be dropped and which artefacts will be arrived at.
- 2. candidate tools that will be used in the project.
- In our case we might include the language used and other libraries used in developing the project.

b) a) The artefacts generated during the first iteration of the design phase include

1.Domain model

* Illustrates meaningful classes in the problem domain

* It is **not** a representation of software objects but of real world concepts.

* we can use UML diagrams to show the domain objects, association between objects and their attributes.

2.Design model

* Here we make use of use case diagrams to represent functional requirements.

3.Data model

* Visually represent how the data will flow inside the system.

* describe entities, attributes and relationships between entities.

* This migh include phases in the form of

1.Conceptual modelling

* The entities are indentified at this point.

2. logical modelling

* Identify the attributes of the entities.

3. Physical modelling

* design the databse structure.

* This might include information regarding databse operations, schemas etc.

4. Test model

5. Software architecture document.

* In the elaboaration phase the **core architecture** is arrived at by iterative development. This is recorded in the form of a software architecture document which will be refined in further elaboration iterations.

Question **4**
Complete
Marked out of 2.00

The purpose of elaboration is to thoroughly and carefully define models, which are translated into code during construction. Do you agree with this statement? State Reasons if you agree or not?

No
The purpose of elaboration is not to throughly and carefully define models.

That would be like imposing ***waterfall ideas*** on to to the ***iterative development*** approach of the **Unified Process**.

In the elaboration phase

    * majority of requirements are discovered and stabilized,

    * major risks are mitigated.

    * core architecture is iteratively implemented

In fact trying to carefully define the models goes against the idea of unified process where the system is incrementally and interatively improved.

Question **5**
Complete
Marked out of 4.00

Take your SE project as a case study and explain without ambiguity the Non Functional requirements in your project?

1. **Portability**: The software developed must be able to be run in any operating system. Operating system must favourably not be a limitation.

2. **Maintainability:** The code must be written in a way such that it can be developed and maintained by someone else too. good documentation has to be maintained.

3.**Usability:** The software must not be too hard to use.Maybe using a gui can give some help. For example: GUI for inserting an image to be classified as malicious or benign

4.**Reliability:** The software must do what it claims to do. For example our project claims to classify an image as benign or malicious. Then it has to do it almost all the time.

5.**Scalability:** If possible the software project should be scalable to include more features.

For example, our project the malJPEG solution takes 10 features, there must be the option to scale it to include more than 10 features.

◄ Lecture 11

Jump to...