SWEN 6301: Assignment #2

Revision 1

Due on October 24, 2020 at 2:00 PM

40 Points (10% Overall)

Problem 1

(10 points)

1. *(2 points)* Briefly discuss a non-functional requirement that can be implemented or designed without having detailed knowledge about the target software product.

2. *(2 points)* Your client has decided to choose an open-source software. Is this a functional or non-functional requirement? How early in the life-cycle model can this requirement be handled? Explain your answer.

3. *(2 points)* Your client has decided that all documentation has to be written in both English and Arabic. Is this a functional or non-functional requirement? How early in the life-cycle model can this requirement be handled? Explain your answer.

4. *(2 points)* Discuss a possible solution on how an engineer responsible for drawing up a system requirements specification might keep track of the relationships between functional and non-functional requirements.

5. *(2 points)* Your technical manager has decided to use the developed rough prototype as a basis for developing your client's software. List at least 2 possible issues for using and evolving the rough prototype into a software product.

Problem 2

(10 points)

Read more about **requirements traceability**[1] using online resources and answer the following questions:

1. *(2 points)* Define in simple terms what is requirements traceability.

2. *(2 points)* List at least 2 benefits of requirements traceability?

3. *(2 points)* List at least 2 available tools to support requirements traceability?

4. *(4 points)* Explain how requirements traceability is important when: (a) planning a change in requirement, and (b) performing acceptance testing by the client or quality assurance team.

Problem 3

(10 points)

The US Defense Advanced Research Projects Agency (DARPA) has recently posted a Broad Agency Announcement (BAA) for a new program called Intent-Defined Adaptive Software (IDAS). The program is about developing specific techniques to manage complexity in engineering software. Read more about the IDAS problem online[2] and watch the Youtube video[3], then answer the following questions:

1. *(5 points)* In few sentences, briefly describe the central problem the IDAS program is concerned with.

---

[1]https://en.wikipedia.org/wiki/Requirements_traceability
[2]https://www.darpa.mil/program/intent-defined-adaptive-software
[3]https://www.youtube.com/watch?v=QbfB9ECmYYA

2. *(5 points)* One possible solution to the IDAS problem is to define a domain-specific language (DSL)[4] that defines programming constructs that are specific and optimized to the problem domain. For example, a DSL can allow the developer to type "sort array ascendingly" instead of writing her own sorting algorithm. DSL would allow the developer to focus more on the problem (intentions) her trying to solve instead of the specificity and complexity of implementing the required algorithms.

    (a) *(3 points)* Briefly describe what is a DSL and give two DSL examples.
    (b) *(2 points)* Discuss the relation between meta-programming[5] and DSLs?

Problem 4

(10 points)

Draw a **sequence diagram**[6] associated with the use case of an interactive order on Internet where someone orders an item on the Internet, but when the overall total, including sales tax and delivery charges, is displayed, the buyer decides that the price is too high and cancels the order.

The **sequence diagram** should depict the dynamic creation and subsequent dynamic destruction of order with the following requirements:

- *(2.5 points)* The diagram should include an actor called `Buyer`, a `User Interface` class, `Assemble Order` class, `Order` class, and the `Price` Class.

- *(7.5 points)* It should include the following activities:

    (a) *(1 point)* `Buyer` gives order details.
    (b) *(1 point)* Details are conveyed to the `Assemble Order` class.
    (c) *(1 point)* A new order is created.
    (d) *(1 point)* The newly created order is conveyed to the `Assemble Order` class.
    (e) *(1 point)* The `Assemble Order` class consults the `Price` class and determines the price of the order.
    (f) *(1.5 points)* The price is conveyed to the `Buyer` through the `User Interface` class.
    (g) *(1 point)* The `Buyer` destroys the order.

---

[4]`https://en.wikipedia.org/wiki/Domain-specific_language`
[5]`https://en.wikipedia.org/wiki/Metaprogramming`
[6]You can use `https://www.draw.io/` to draw/export your UML diagrams