UNIVERSITY OF JOHANNESBURG

FACULTY OF SCIENCE
ACADEMY OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING

MODULE	**IFM03B3 & IFM3B10** INFORMATICS 3B: ADVANCED SOFTWARE ENGINEERING
CAMPUS	APK
FINAL SUMMATIVE ASSESSMENT	JANUARY 2021 (SSA)

DATE		2021-01-19	DOWNLOAD ALLOWANCE	15:00–15:10
			WRITING TIME	15:10–18:10
			UPLOAD ALLOWANCE	18:10–19:00
ASSESSORS			PROF W.S. LEUNG DR F.F. BLAUW	
EXTERNAL MODERATOR			PROF A. VAN DER MERWE *(UNIVERSITY OF PRETORIA)*	
DURATION	*DOWNLOAD TIME*	*10 minutes*	MARKS	150
	WRITING	*3 Hours*		
	UPLOAD TIME	*50 minutes*		

MEMO

Question 1: Quality Concepts [15]

1.1. Correctness. (3)
1.2. Usability. (3)
1.3. Integrity. (3)
1.4. Testability. (3)
1.5. Reusability. (3)

Marking Notes:

- For the above, students must describe what would constitute as a indication of the five characteristics with regard to features that would typically be available to contract customers – this would include registration, viewing current data balance, and donate data.
- 3 marks are given for each quality characteristic.
- For reference:
  - **Correctness:** accuracy and completeness of the process
  - **Usability:** the system behave approrpriately, with users finding it easy to work with.
  - **Integrity:** The system operates in a consistent and complete manner.
  - **Testability:** It is possible to test the product.
  - **Reusability:** In the event that the product is to be "translated" or "moved", the components that make up the system is able to be reused in the new system without reinventing the wheel.

0	Provides the interpretation of the characteristic (e.g. Effectiveness is described as being comfortable)
1	Correct interpretation, but really just a definition, no actual application.
2	Correct interpretation, with application, but is either a shallow reference or only relates to a limited portion of the characteristic (e.g. satisfaction: users find the experience of installing the app with ease)
3	Correct interpretation, comprehensive application (e.g. context coverage: users are able to install the app, indicate their consent, and have the option of granting permissions according to several options)

Question 2: Conducting Technical Reviews [10]

2.1. Calculate the total review effort. Show your full working out. (5)

Marking Notes:

- Marks allocated based on: correct formula used, correct values substituted, correctly calculated
- $E_{review} = E_p + E_a + E_r$
- E_p = Sum of person hours in table above (12 + 2.5 + 2 + 1.5 + 3) = 21
- E_a = Five team members, meeting up for 4 hours = 5 × 4 = 20
- E_r = Total time to fix all errors (7 × 8) + (26 × 3) = 134
- $E_{review} = E_p + E_a + E_r = 21 + 20 + 134 = 175$ person hours

2.2. Comment on the effectiveness of the technical review. Advise how you would have it run the next time. (5)

Marking Notes:

- The team waited until the end (closed Beta testing) to do the technical reviews.
- The team had to review the entire work product (should be smaller, more manageabke component)
- Testing should take place a lot sooner and more frequently
- The software is likely riddled with more problems than identified
- Number of people involves = 5 (which is good)
- Time spent on the preparation, time spent on the actual technical review: too long
- Use the error density from this time to advise on weaknesses the next project

QUESTION 3: SOFTWARE QUALITY ASSURANCE [15]

3.1. Explain why statistical quality assurance is seen to be a strategic approach to ironing out errors and defects in software. (5)

Marking Notes:

- Resources are not infinite, do not have time to check EVERYTHING
- Pareto Principle states that 80% of defects can be traced to 20 percent of all possible causes. Point is then to isolate that 20%
- A statistical approach would help us to isolate that 20% (instead of guessing and hoping for the best).

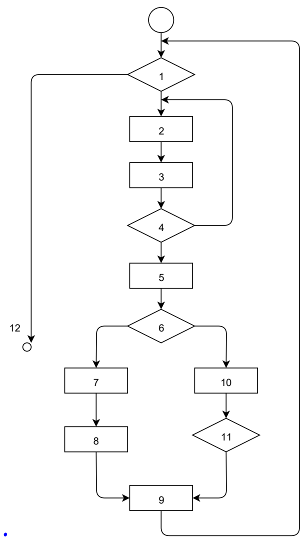
3.2. Identifying a potential (and reasonable) defect that may be present in `DataDonor`, describe the first four steps you would undertake if following the Six Sigma strategy reducing costs while improving quality. (10)

Marking Notes:

- Student to identify one potential defect relating to DataDonor (2 marks)
- The last 8 marks, 2 for each step is for the student to describe how they would eliminate the defect through Six Sigma: define, measure, analyse, improve
- The steps MUST relate to the problem they have identified.

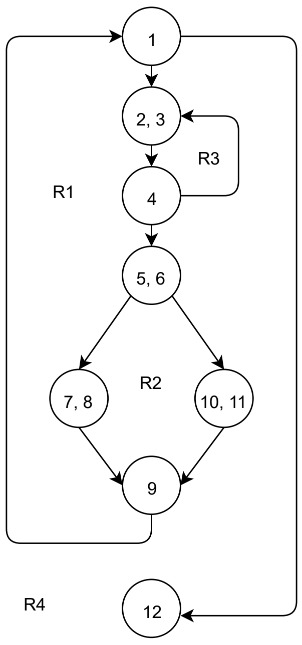
Question 4: Software Testing [40]

4.1. Consider the following flowchart. Draw the corresponding flow graph to determine how many regions (and hence independent test paths) exist. (10)



Marking Notes:

- Student MUST draw below. Edges and nodes may differ slightly
- Regions must be considered



4.2. Discuss how you would set up a unit-test environment to test the units that make up `DataDonor`. (10)

Marking Notes:

- Discussion of scaffolding
- Description of stubs (receives input from the unit being tested, performs calculation, and “returns” output to the unit being tested, and drivers (passes the data to the unit)
- Description of different types of test cases to achieve this, including interface, local data structures, boundary conditions, independent paths, and error-handling paths

4.3. Assuming that DataDonor comprises a number of classes that include `Donor`, `DonorAccount`, `Donation`, and `Student`, use the information provided previously to discuss how integration testing will take place for `DataDonor`. (10)

Marking Notes:

- Student to discuss how DataDonor can be subject to fault-based test-case design as well as scenario-based test-case design
- Both of complementary of each other
- Fault-based test case design should consider the available interactions between Donor, DonorAccount, Donation, and Student to come up with random test cases that may be triggered during any operation (e.g. when user wants to donate their data, the system will need to check the user's account for available data)
- Fault-based test case design is system-based (what the system does)
- Scenario-based covers the other end, which is when human uses mess up

4.4. Discuss the concept of security testing of `DataDonor` and suggest how this may be done properly. (10)

Marking Notes:

- Based on the nature of data stored at by the app, highly confidential data must be protected
- This includes biographic information of the users (includes donor and students)
- Security testing verifies whether the protection mechanisms in place are working and protect against attempts to gain the data (hacking)
- Tests probe the possibility of gaining unauthorised access (simulation of hackers)
- Bugs may be introduced based on dependencies in the existing code libraries
- Design of security should not be an afterthought
- For data of extremely high sensitivity, it is best to get outside expert help

QUESTION 5: SOFTWARE CONFIGURATION MANAGEMENT [10]

5.1. Discuss how you will make use of e-Change Control to effect any change management for `DataDonor` effectively. (10)

Marking Notes:

- Category each change request that has been approved into one of four classes
- Based on the classes, the chagne is processed
- Depending on the degree of change, the change project may be subject to longer time to fix
- Small changes can be done informally
- The more complex the changes, the more formal the management of the change must go.
- Changes classified as Class 3 and 4 require documentation

Question 6: Software Metrics and Analytics [15]

Using Chidamber and Kemerer's OO software metrics, describe how the following design metrics will be used to assess DataDonor

6.1. Weighted methods per class. (3)
6.2. Depth of the inheritance tree. (3)
6.3. Number of children. (3)
6.4. Coupling between object classes. (3)
6.5. Lack of cohesion in methods. (3)

Marking Notes:

- As with Question 1, student will need to describe how the above will determine the overall complexity of DataDonor
- Discussion in terms of whether DataDonor should realistically have reason to have lower or higher of the values below...
- Weighted methods per class: complexity of each of the classes in the program. Higher value means lower reusability
- Depth of the inheritance tree: larger DIT values = lower level classes have much inheritance in action - larger DIT means more many methods may be reused
- Number of children - reuse will increase with larger number of children but it may also be indication of child classes inheriting methods that it doesn't actually need to reuse. Testing can increase due to the numerous available children classes
- Coupling between object calsses: relates to collaboration between classes. High coupling means complicated modifications (decreased reusability) and more testing
- Lack of cohesion in methods: methods that access one or more of the same attributes in the class. In this case, higher cohesion amongst the methods, which complicate the design. Ideally should be kept low.

QUESTION 7: RISK MANAGEMENT [20]

7.1. Identify and briefly describe two potential risks relating to `DataDonor` that are relevant to two different categories of risk. (6)

7.2. For each risk identified in 7.1, assign a risk probability (as a percentage), and describe the perceived risk impact *(in rand value)* to calculate a risk exposure for each. Provide motivations for each of the values you assign. (6)

7.3. Choosing one of the risks you have identified, describe the mitigation, monitoring and management plan for that risk. (8)

Marking Notes:

- Mark according to student input

7.1:

- Potential risks should be relevant to DataDonor
- Risk categories available (they must choose two):
    - Product size
    - Business impact
    - Stakeholder characteristics
    - Process definition
    - Development environment
    - Technology to be built
    - Staff size and experience

7.2:

- Student to allocate values as appropriate (appropriateness will be determined by the student's motivation

7.3:

- Mitigation, Monitoring, and Management should be realistic and logical

QUESTION 8: LEGAL ELEMENTS OF SOFTWARE ENGINEERING [20]

8.1. Is the above an example of an Opt-in or Opt-out form? Motivate your answer. (4)

Marking Notes:

- Example of opt-out – user has to make move to be taken OFF the marketing distribution by default as the mark is checked by default

8.2. For each of the eight principles of the POPI Act, describe how you would ensure that they are correctly implemented for `DataDonor`. (16)

Marking Notes:

- Elements (in particular design) of DataDonor must be referenced for each of the eight principles of POPIA
  - Accountability
  - Processing Limitation
  - Purpose Specification
  - Further Processing Limitation
  - Information Quality
  - Openness
  - Security Safeguards
  - Data Subject Participation
- Student cannot just list the above. Explaining what they are is not sufficient.
- For example information quality: DataDonor must provide users with the option to update their personal information should it change.

Question 9: A Strategy for Software Support [5]

9.1. Describe how you would make use of social media to proactively support `DataDonor`. (5)

Marking Notes:

- The app store allows for users to provide feedback
- It would be helpful to monitor the feedback from the app stores
- It is possible to automate the analysis using AI (not always possible to monitor each and every comment)