



Faculty of Science

Academy of Computer Science and Software Engineering

Final Summative Assessment *First Opportunity*

Module	IFM03A3 / IFM3A10 Informatics 3A – Introduction to Software Engineering
Campus	APK
Month	June 2020

Date	5 June 2020	Time	08:30
Assessor	Mr F F Blauw		
Moderator	Prof AJ van der Merwe (UP)		
Duration	180 minutes	Marks	150

MEMORANDUM

SpectreStable

Who you gonna call?

A couple of months ago Mr Fred Jones, Chairperson of SAGA, approached you in an attempt to reduce the administrative load of South African Ghosting Association (SAGA). While you did not believe in “those types of things” at the time, you found that your thoughts had changed over time. Today, you once again received a call from Mr Jones:

“We have completed the development and deployment of SpectreSpotter that you designed a couple of months ago. Our Paranormal Investigators, or Paratroopers, have seen a definite increase in their productivity.”

You smile at the thought that your system helped capture ghosts – something you did not even think ever existed!

“As you will undoubtedly remember, the SpectreSpotter was used by Paratroopers to investigate houses and capture the results of their investigations and manifestations in the SpectreSpotter system. Now we need to start with the next phase of our development. Investigating haunted houses is one thing, but now, once the ghosts are caught, we need to ensure that they are securely stored at the SpectreStable.

“The SpectreStable is where captured ghosts are brought in to be processed and hopefully released. However, the SpectreStable has been inundated with ghosts over the past couple of months and ghosts tend to get stranded in the process. Or worse, dangerous ghosts are accidentally released! The Curator of the SpectreStable is becoming furious! This is what generally happens at the SpectreStable:

“Paratroopers bring in ghosts to the SpectreStable and hand them over to the GhostGuard on duty – as appointed by the Curator. These ghosts were identified in the SpectreSpotter system. A GhostGuard will then take the ghost to The Vault.

“The Ghost ReAdjustment Division (GRAD), led by Professor P Geist, will examine each ghost placed in The Vault. Taking into considering the evidence provided by SpectreSpotter, the GRAD will determine whether a ghost can be readjusted and released, or whether it should be sent to The Beyond™.

“Ghosts selected for readjustment are given a regime they must follow. This regime involves a programme of Silent Spooking and Frolicking Phantasms to make the ghosts less frightening. Once they passed the Apparition Assessment, and the GRAD is satisfied, they will be allowed to go.

“On the other hand, should a ghost be selected to go to The Beyond™, it must be agreed upon by at least 66% of the GRAD. Since passage to The Beyond™ is extremely busy, ghosts are placed in a queue, with more dangerous ghosts placed at the front.”

You start wondering if there is something beyond the design of this system... Mr Jones interrupts your thoughts:

“We need you to design an information system that can effectively manage the process I just described to ensure that ghosts do not get lost and only the approved ones are released.”

QUESTION 1

- 1.1. Provide a definition for Software Engineering. (3)

The application of a
systematic, disciplined, quantifiable approach [1]
 to the
development, operation, and maintenance [1]
 of software; that is, the application of engineering to software.
 And the **study of approaches [1]** of such.

- 1.2. But, what is software? (3)

Instructions [1] that when executed provide desired features, function, and performance;
Data structures [1] that enable the programs to adequately manipulate information;
 And **description information [1]** in both hard copy and virtual forms that describes the operation and use of the programs.

- 1.3. Building software faces new challenges every day. Briefly describe one such a challenge and the best way you believe to overcome this challenge. (4)

Heterogeneity

Increasingly, systems are required to operate as distributed systems across networks that include different types of computer and mobile devices.

Business and social change

Business and society are changing incredibly quickly as emerging economies develop and new technologies become available. They need to be able to change their existing software and to rapidly develop new software.

Security and trust

As software is intertwined with all aspects of our lives, it is essential that we can trust that software.

DESCRIPTION OF CHALLENGE [2]

MEANS TO OVERCOME SUCH CHALLENGE [2]

[10]

QUESTION 2

One of the most difficult tasks software engineer faces, is understanding the requirements. The seven tasks of requirements engineering can be defined as follows:

1. Inception
2. Elicitation
3. Elaboration
4. Negotiation
5. Specification
6. Validation
7. Management

- 2.1. Which **requirements analysis model** would be the best for the project described by Mr Jones? Motivate your answer by referring to the steps described above. (7)

Scenario-based models
 Class-oriented models
 Behavioural and Pattern-based models
 Data Models
 Flow-oriented Models

MOTIVATION SHOULD MATCH PROCESS [3]

MOTIVATION SHOULD INCLUDE CASE STUDY [4]

- 2.2. Provide a concise **problem statement** for the SpectreStable. (4)

Relevance to Case Study [2]

Description of source Problem, not solution [2]

- 2.3. Provide a brief **proposed solution** for the SpectreStable. (4)

Relevance to Case Study [2]

Brevity/Cohesiveness of solution [2]

[15]

QUESTION 3

- 3.1. Name one **non-functional requirement** that specifically applies to the SpectreStable. Explain how you believe it can be applied and how it can then be measured. *NOTE: Do not use general non-functional requirements such as: security, reliability, availability, etc.* (5)

Relevance to Case Study [1]

Application to Solution/Functional Requirements [2]

Applicable Measurement [2]

- 3.2. List *all* the **functional requirements** that you can extract from Mr Jones's description of SpectreStable. *NOTE: The mark allocation does not indicate the number of functional requirements.* (6)

Phrased as Functional Requirements (Stories) and NOT Use Cases

Each use case should be relevant from the case study.

- 3.3. Draw a **use case diagram** illustrating the use cases derived from the requirements identified in **Question 3.12**. (20)

Use Cases MUST match Functional Requirements from previous question

	Weak	Fair	Accomplished
Applicable Actors	0-1	2-3	4
Applicable Subsystems	0	1	2-3
Applicable Use Cases	0-3	4-6	7-9
Diagram Correctness	0-1	2-3	4

Applicable refers to Case Study

- 3.4. If you could **alter** the functional requirements for SpectreStable, what would you add or change, and why? (4)

Student's Answer. Should NOT changes business processes!

[35]

QUESTION 4

Draw an **activity diagram** for a use case entitled: “*Place Ghost in Queue for The Beyond*”.

[15]

If it does not contain INFORMATION SYSTEM subsystems/swimlanes = 0

	Weak	Fair	Accomplished
Basic Activity Elements (Start, End, Arrows)	0	1	2
Swimlanes	0	1	2-3
Appropriate Actions	0-1	2	3-5
Logical Flow	0-1	2	3-5

QUESTION 5

Consider the following C# code. Draw an **Interaction Sequence Diagram** to model the code. You may assume that a GRAD member clicked on **ClickSendGhostToBeyond** on the GUI.

```

01: using System;
02:
03: public class GUIBeyondManager
04: {
05:     public void ClickSendGhostToBeyond()
06:     {
07:         Ghost InvestigateGhost;
08:         InvestigateGhost = new Ghost(/* read score from GUI */, /* read votes from GUI */);
09:         bool PriorityGhost = false;
10:         if (!InvestigateGhost.HasRehabilitated())
11:         {
12:             Console.WriteLine(TheBeyond.SendToBeyond(InvestigateGhost, PriorityGhost));
13:         }
14:         else
15:         {
16:             Console.WriteLine("Ghost released.");
17:         }
18:     }
19: }
20:
21: public class Ghost
22: {
23:     public int Score { get; set; }
24:     public int Votes { get; set; }
25:     public Ghost (int S, int V)
26:     {
27:         this.Score = S;
28:         this.Votes = V;
29:     }
30:     public bool HasRehabilitated()
31:     {
32:         return (this.Score > 50 && Votes > 66);
33:     }
34: }
35:
36: public static class TheBeyond
37: {
38:     public static String SendToBeyond(Ghost sendGhost, bool Priority)
39:     {
40:         if (sendGhost.Score > 10) return "Sent to the The Beyond™.";
41:         return "Unable to send. Ghost too naughty.";
42:     }
43: }

```

[20]

	Weak	Fair	Accomplished
Classes from Code	0-1	2	3-4
Appropriate Messages/ Method Calls	0-1	2-4	5-6
Logical Flow	0-2	3-5	5-8
Diagram Correctness	0	1	2

QUESTION 6

- 6.1. Which **software development process** do you think will be best for the development of SpectreStable? Motivate your decision based on SpectreStable. (3)

Waterfall

Evolutionary

Iterative

Any Agile-based process

MOTIVATION SHOULD MATCH PROCESS [3]

- 6.2. Provide a brief description of the **operation** of the software development process you selected in Question 6.1. You may use a diagram to aid you. (7)

Description of give an overview

	Weak	Fair	Accomplished
Diagram	0	1	2-3
Description	0-1	2-3	3-4
Description*	0-1	2-4	5-7

* If no diagram, full [7] marks can be awarded to description

[10]

QUESTION 7

7.1. What is a **software architecture**? (3)

The software architecture of a program or computing system is the **structure or structures of the system [1]**, which comprise **software components [1]**, the **externally visible properties of those components [1]**, and the **relationships [1]** among them

7.2. Why is it important to decide on a software architecture for a particular project? (3)

Analyze effectiveness
 Consider architectural alternatives
 Reduce the risks
 Communication among all stakeholders
 Early design decisions
 Model of how the system is structured

7.3. When starting an architectural design, you are faced with many possible architectural styles. What are the two factors you need to consider when choosing an appropriate style? Briefly describe each. (2)

CONTROL

How is control managed? How do components transfer control?

DATA

How is data communicated? Is the flow continuous? What is the mode of transfer?
 How do functional components interact with data components?

7.4. Considering SpectreStable, which **generic architectural style** will you base your design on? Motivate your decision. (2)

Layered
 Data Centred
 Object Oriented
 NOT: Data flow, Main/Subsystem
 Motivation must make sense in terms of the Case Study

7.5. Use a diagram to describe the architectural style you selected in **Question 7.4** while referring to SpectreStable in the diagram. (10)

Description of give an overview

	Weak	Fair	Accomplished
Diagram*	0-1	2	3-4
Description*	0-1	2-3	3-6

* If diagram does not contain components from case study, WEAK only, full ACCOMPLISHED can still be awarded to description

[20]

QUESTION 8

- 8.1. Choose any one (1) design pattern that could be applied to **SpectreStable**. Describe this design pattern and the reason you chose it. (5)

	Weak	Fair	Accomplished
Description	0	1	2
Motivation	0	1	2-3

*

- 8.2. Draw a conceptual class diagram of the **business domain** and **process components** for **SpectreStable**. (15)

	Weak	Fair	Accomplished
Applicable Business Domain Components	0-1	2-4	5-6
Applicable Process Components	0-1	2-3	4-5
Logical Connection between Classes	0-1	2-3	4

*

[20]**QUESTION 9**

Considering that **SpectreStable** deals with highly sensitive data, discuss measures you will put in place to secure this data from unauthorised access.

[10]