

**ATHLONE INSTITUTE OF TECHNOLOGY**

**SCHOOL OF ENGINEERING**

**SEMESTER 2 EXAMINATIONS 2017**

**May Session**



**BACHELOR OF ENGINEERING IN SOFTWARE ENGINEERING**

**YEAR 4**

**AGILE METHODOLOGIES 4.2**

**External Examiner(s):      Dr Norah Power**  
**Mr Padraig Brennan**

**Internal Examiner(s):      Mr Michael P. Russell**

**Instructions to candidates:**

Read all questions carefully.

All questions carry equal marks.

Answer **ALL (Two)** questions.

***Time Allowed: 2 Hrs***

**No. of pages including cover sheet: 4**

Q.1. Consider the following specification for a code module:

*“The module is passed three integer values: revenue, costs and target. The module calculates the profit (revenue-costs). The value of the dividend is dependent upon the amount of profit. The dividend value is 5% of the profit unless the target is met or exceeded, in which case the dividend value is 10% of the profit. A dividend is only paid if the profit is greater than zero. The dividend value is calculated to 2 decimal places.”*

This specification can be represented by the following pseudo-code:

READ revenue
READ costs
READ target
dividend = 0
profit = revenue – costs
IF profit > 0 THEN
IF profit >= target THEN
dividend = profit * 10%
ELSE
dividend = profit * 5%
ENDIF
ENDIF
PRINT dividend

(a) What is the minimum number of test cases required to achieve 100% **statement coverage**?

2

(3 marks)

(b) What is the minimum number of test cases required to achieve 100% **branch coverage**?

3

(3 marks)

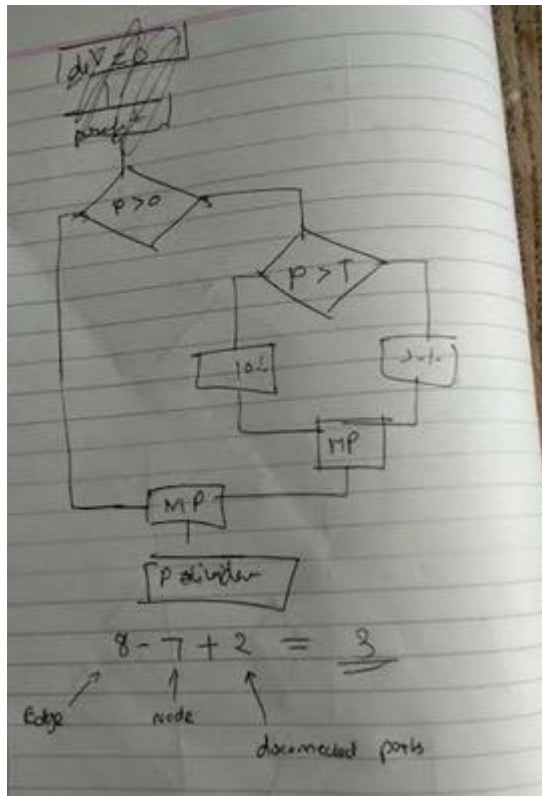
(c) What is the Cyclomatic Complexity? Explain how you obtained your answer.

Count of number of decisions in your source code. The higher the count, the higher the complexity. Can be used to limit code complexity, determine no of test cases.

The number of independent paths through a program. Cyclomatic complexity (V) is defined as:  $V = N + 2P$ , where  $N$  = the n

umber of edges/links in a graph - N = the number of nodes in a graph - P = the number of disconnected parts of the graph (e.g. a called method or function))

**CYC comp = decisions + 1**



(4 marks)

(d) How many additional test cases (if any) need to be added to this test set to achieve 100% statement coverage?

	Input			Expected Outcome
Test 1	revenue = 5	costs = 3	target = 10	dividend = 0.10
Test 2	revenue = 500	costs = 70	target = 150	dividend = 43.00
Test 3	revenue = 1000	costs = 500	target = 700	dividend = 25.00

(3 marks)

Test 1 and Test 3 cover same statement so only two test cases are enough to cover all statements. Test 1 and 2 satisfy all statement coverage.

(e) How many additional test cases (if any) need to be added to this test set to achieve 100% **branch coverage**?

	Input			Expected Outcome
Test 1	revenue = 500	costs = 600	target = 100	dividend = 0
Test 2	revenue = 700	costs = 300	target = 450	dividend = 20
Test 3	revenue = 3000	costs = 2000	target = 1100	dividend = 50

Test 2 and Test 3 cover same branch. We need one more test case which covers profit > target

(3 marks)

(f) Calculate the branch coverage of the following test set:

	Input			Expected Outcome
Test 1	revenue = 30	costs = 100	target = 70	dividend = 0
Test 2	revenue = 650	costs = 660	target = 300	dividend = 0
Test 3	revenue = 1000	costs = 500	target = 200	dividend = 50
Test 4	revenue = 3000	costs = 200	target = 1000	dividend = 280

66.6 % branch coverage  
2 out of 3 branches covered

(4 marks)

**[20 marks]**

Q.2. (a) Using examples, explain why configuration management/version control is important when a team of people are developing a software product.

- There are multiple people working on software which is continually updating
- It may be a case where multiple version, branches, authors are involved in a software project, and the team is geographically distributed and works concurrently
- Changes in user requirement, policy, budget, schedule need to be accommodated.
- Software should be able to run on various machines and Operating Systems
- Helps to develop coordination among stakeholders
- SCM process is also beneficial to control the costs involved in making changes to a system

Storing Versions (Properly)

Restoring Previous Versions

Understanding What Happened

Backup

(b) Imagine a situation where 2 developers are simultaneously modifying 3 different software components. What difficulties might arise when they try to merge the changes that they have made?

- ▶ There is the usual problem where developers each make changes to the same component and these changes are, in some way, incompatible.
- ▶ However, where several components are being changed at the same time, the problems are exacerbated because there may be dependencies between the components that are affected by the changes.
- ▶ For example, say developer A checks out components X and Y and decides to implement a change by changing Y, which depends on a particular feature of X. Developer A checks X and Y back in with no changes recorded as being made to X. Developer B also is working on X and Y and changes both X and Y. However, the changes made to X mean that the assumptions made by Developer A no longer hold. However, incompatibility is not detected as there has only been a single change made to component X. With more than 2 components, the problem becomes even worse because of the chains of dependencies that can be introduced. These can be very difficult or impossible to detect automatically.

(4 marks)

(c) What problems do you think might arise in extreme programming teams where many management decisions are devolved to the team members?

While the notion of devolving management decisions to the team is attractive in terms of motivation, there are two types of problem that can arise:

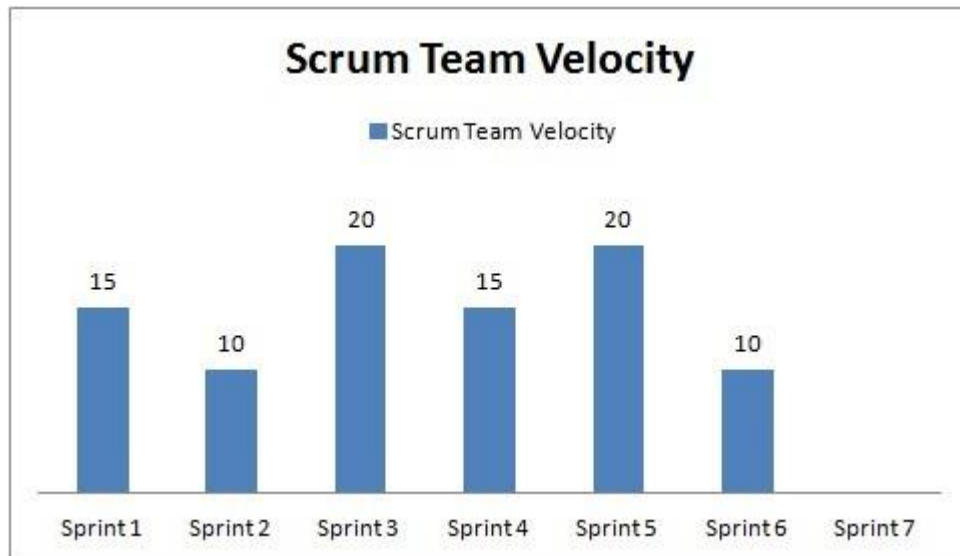
1. Decisions are liable to be primarily influenced by technical considerations rather than business decisions. This is natural given the type of people on an XP team – it is difficult for them to take a business perspective.
2. Because of the focus on rapid iteration, management decisions tend to be short-term and pay insufficient attention to long-term issues. While this is in keeping with the XP philosophy, there is sometimes a need for a more detached, longer-term perspective which can be taken by a manager.

I assume here that management decisions on e.g. the performance of team members are not taken by the team. Given the close knit nature of XP teams, it is difficult for the team to take decisions that censure individual team members.

(4 marks)

(4

(d) A scrum team's Product Backlog and velocity for Sprints 1 to 6 are summarized in the diagrams below. Assuming that none of the team members have taken holidays during Sprints 1 to 6 and no team member plans to take holidays during Sprint 7, identify the User Stories from the Product Backlog that the team will commit to in Sprint 7? Justify your answer.



Scrum Team Product Backlog Summary	
User Story Number	Allocated Number of Story Points
1	3
2	1
3	3
4	5
5	8
6	3
7	1
8	1
9	5
10	1

(8 marks)  
[20 marks]

