

## **WARNING**

This material has been reproduced and communicated to you by or on behalf of *Charles Darwin University* in accordance with section 113P of the *Copyright Act 1968 (Act)*.

The material in this communication may be subject to copyright under the Act.  
Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act.

**Do not remove this notice**



Family Name					
Given Name/s					
Student Number					
Teaching Period	Semester 1, 2019				

<b>PRT453 – Process Development Methodologies</b>	<b>DURATION</b>	
	Reading Time:	<b>10 minutes</b>
	Writing Time:	<b>180 minutes</b>
<b>INSTRUCTIONS TO CANDIDATES</b>		
<p>1.1 The examination has <b>1</b> section. There are 10 questions. You must answer all questions.</p>		
Suggested Time:	180 minutes Marks:100	
<p>All questions must be answered in the Answer Booklet provided. Please ensure that your name and student number are clearly indicated on your Answer Booklet and at the top of this examination paper.</p>		
<p>1.2 Note that questions <b>ARE NOT</b> of equal value.</p>		
<p>1.3 Read <b>ALL</b> questions carefully.</p>		
<b>EXAM CONDITIONS</b>		
<p><u>You may begin writing from the commencement of the examination session.</u> The reading time indicated above is provided as a guide only.</p>		
This is a RESTRICTED OPEN BOOK examination		
No calculators are permitted		
One A4 sheet of handwritten double-sided notes permitted		
No dictionaries are permitted		
<b>ADDITIONAL AUTHORISED MATERIALS</b>	<b>EXAMINATION MATERIALS TO BE SUPPLIED</b>	
No additional printed material is permitted	1 x 20 Page Book	

**THIS EXAMINATION IS PRINTED  
DOUBLE-SIDED.**

**THIS PAGE HAS BEEN INTENTIONALLY  
LEFT BLANK.**

## Section A

### Short Essay Questions

**Total No of Marks for this section: 100**

This section should be answered in the Answer Booklet provided.

Marks for each question are indicated. Suggested Time allocation for Section A: 180 mins

---

#### Question 1

- a) You are the team leader of a team of ten software engineers working on an important IT 1-year project that has heavy penalties for running late. It is now 8 months into the project and it is quite evident that the project will not reach its deadline. One obvious option is to add more staff to the project. With reference to the text Mythical Man-Month by Fred Brooks, explain if you will take up this option  
(Marks: 3)
- b) What other software process options should be considered?  
(Marks: 3)
- c) "The man-month as a unit for measuring the size of a job is a dangerous and deceptive myth"  
Discuss this myth with different types of tasks  
(Marks: 4)

#### Question 2

- a) Discuss the differences between "Predictable manufacturing" and "New product development"  
(Marks: 3)
- b) How to deal with change management (change of requirements) in an IID method?  
(Marks: 4)
- c) Agile projects come with a set of challenges and problems that are different from traditional methodology. List three reasons why agile projects fail  
(Marks: 3)

### Question 3

- a) Discuss Compare XP and SCRUM methods using
1. Cockburn scale
  2. Ceremony and Cycles scale
- (Marks: 4)
- b) Explain Kent Beck's metaphor of "two hats". (Short Sentence answers)
- (Marks: 3)
- c) In software engineering, CI/CD or CICD refers to the combined practices of continuous integration and continuous delivery. List three benefits of using CI/CD
- (Marks: 3)

### Question 4

- a) The word extreme in XP comes from Beck's conviction that to take known good practices and "turn the dial up to 10", or to extreme levels.  
In the context of above line, explain how XP takes good practices to extreme levels.
- (Marks: 5)
- b) Discuss how pair programming helps in knowledge sharing? Explain your answer with team size of C20 and L100?
- (Marks: 5)

### Question 5

- a) Duplicate Code is a pungent code smell. To fix this we could use refactoring like Extract Method or Extract Class.  
List and describe two composite Refactoring's that could be used? (Short answers)
- (Marks: 4)
- b) Refactoring is intended to improve non-functional attributes of the software. Refactoring during development might delay the project delivery.  
Explain when we should refactor? (Short answers)
- (Marks: 3)
- c) Discuss the difference between Test Driven Development (TDD) and Unit Testing?
- (Marks: 3)

### Question 6

Describe each of the following (Short answers. 2 marks each.)

- a) Risk driven planning
- b) Client driven planning
- c) Evolutionary and Adaptive development
- d) Abstract factory over Factory method
- e) Composition over Inheritance

(Marks: 10)

### Question 7

- a) Draw a burn down chart for the following sprint

User Stories		Days (Points Completed)										Remaining	
Sprint Backlog Items	Beginning Balance	1	2	3	4	5	6	7	8	9	10	Task Balance	Percentage Completed
Feature 1	25	2	2		1	3						17	32.00%
Feature 2	22	3	2	1			3					13	40.91%
Feature 3	40	4	3	1	1	1	3					27	32.50%
Feature 4	23	3	1				2					17	26.09%
Feature 5	10		1				3					6	40.00%
Feature 6	32	3										29	9.38%
Feature 7	4	4			4		2					-6	250.00%
Feature 8	9	3	2	2	1							1	88.89%
Feature 9	8	2					1					5	37.50%
Feature 10	21	1	1	2	2	1						14	33.33%

(Marks: 5)

- b) Product owner asked to add one high priority story with critical functionality to deliver in the above sprint. Discuss your action points as Scrum team?

(Marks: 5)

## Question 8

- a) Consider the following narrative:

Tom: Sure, we don't apply the waterfall methodology – everybody knows it doesn't work. We've adopted SCRUM for our project. We have already completed listing all the requirements for the first two iterations. After we have meet with the client and he has reviewed our story cards we'll start programing. Comment on this. (Short paragraph).

(Marks: 4)

- b) Scrum ceremonies provide the framework to get the work done. List and describe scrum ceremonies

(Marks: 3)

- c) User stories are most widely used agile technique to capture product functionality. Give one example of groomed user story

(Marks: 3)

## Question 9

- a) Examine the code and describe how you would refactor this code to remove the smell.

(Marks: 5)

- b) Identify any design patterns [GOF] you would use

(Marks: 5)

```
public enum EmployeeType
{
    Worker,
    Supervisor,
    Manager
}

public class Employee
{
    private float salary;
    private float bonusPercentage;
    private EmployeeType employeeType;

    public Employee(float salary, float bonusPercentage, EmployeeType employeeType)
    {
        this.salary = salary;
        this.bonusPercentage = bonusPercentage;
        this.employeeType = employeeType;
    }

    public float CalculateSalary()
    {
        switch (employeeType)
        {
            case EmployeeType.Worker:
```

```

        return salary;
    case EmployeeType.Supervisor:
        return salary + (bonusPercentage * 0.5F);
    case EmployeeType.Manager:
        return salary + (bonusPercentage * 0.7F);
    }

    return 0.0F;
}

public float CalculateYearBonus()
{
    switch (employeeType)
    {
        case EmployeeType.Worker:
            return 0;
        case EmployeeType.Supervisor:
            return salary + salary * 0.7F;
        case EmployeeType.Manager:
            return salary + salary * 1.0F;
    }

    return 0.0F;
}
}

```

## Question 10

- a) Examine the code and identify at least 3 code smells in this code (Marks: 5)
- b) For each code smell describe how you would refactor this code to remove the smell (Marks: 5)

```

public class Loan
{
    public int taxrate = 9;

    public Loan(float notional, float outstanding, int rating, Date expiry)
    {
        this.strategy = new TermROC();
        this.notional = notional;
        this.outstanding = outstanding;
        this.rating = rating;
        this.expiry = expiry;
    }

    public Loan(float notional, float outstanding, int rating, Date expiry, Date maturity)
    {
        this.strategy = new RevolvingTermROC();
        this.notional = notional;
        this.outstanding = outstanding;
        this.rating = rating;
    }
}

```



```

        this.expiry = expiry;
        this.maturity = maturity;
    }

    public Loan(CapitalStrategy strategy, float notional, float outstanding,
        int rating, Date expiry, Date maturity)
    {
        this.strategy = strategy;
        this.notional = notional;
        this.outstanding = outstanding;
        this.rating = rating;
        this.expiry = expiry;
        this.maturity = maturity;
    }

    public static Loan newAdvisor(double notional, Date start, Date maturity, int rating)
    {
        return new Loan(notional, start, null, maturity, rating, new TermLoanCapital());
    }

    public static Loan newLetterOfCredit(double notional, Date start, Date maturity, int rating)
    {
        return new Loan(notional, start, null, maturity, rating, new TermLoanCapital());
    }

    public static Loan newRCTL(double notional, Date start, Date expiry, Date maturity, int rating)
    {
        return new Loan(notional, start, expiry, maturity, rating, new RCTLCapital());
    }

    public static Loan newRevolver(double notional, Date start, Date expiry, int rating)
    {
        return new Loan(notional, start, expiry, null, rating, new RevolverCapital());
    }

    public static Loan newSPLC(double notional, Date start, Date maturity, int rating)
    {
        return new Loan(notional, start, null, maturity, rating, new TermLoanCapital());
    }

    public static Loan newTermLoan(double notional, Date start, Date maturity, int rating)
    {
        return new Loan(notional, start, null, maturity, rating, new TermLoanCapital());
    }

    public float CalculateRepayment()
    {
        return this.oustanding * 9.83;
    }

    public double calcCapital()
    {
        return capitalStrategy.calc(this);
    }

    // more functions
}

```