UNIVERSITY OF COLOMBO, SRI LANKA

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY (EXTERNAL)

Academic Year 2018 – 2nd Year Examination – Semester 4

IT4305: Rapid Software Development

Part 2: Structured Question Paper

30th September 2018

(ONE HOUR)

To be completed by the candidate

BIT Examination Index No: ________________

Important Instructions:

- The duration of the paper is **1 (one) hour**.
- The medium of instruction and questions is English.
- This paper has **3 questions** in **11 pages**.
- **Answer all questions.** The first question carries 40 marks. Second and third questions carry 30 marks each.
- **Write your answers** in English using the space provided **in this question paper**.
- Do not tear off any part of this answer book.
- Under no circumstances may this book, used or unused, be removed from the examination hall by a candidate.
- Note that questions appear on both sides of the paper.
  If a page is not printed, please inform the supervisor immediately.

Questions Answered

Indicate by a cross (×), (e.g. ☒) the numbers of the questions answered.

To be completed by the candidate by marking a cross (×).	**1**	**2**	**3**	
To be completed by the examiners:				

All right reserved.

1. The following question is related to *Xorg*, which is a construction company, who are planning to transform their largely manual information processing activities in to an IT based work environment, using custom developed software solutions. The background of this project is as follows.

‘ABC (Pvt) Ltd’ manages multiple construction projects at the same time, where most of the activities (Purchasing, Budgeting and scheduling etc.) occurs in a paper based setting. The set of activities involved in each of their projects differs from one project to another. Further, they work with a number of business partners in their construction activities. Although the top management of *‘ABC (Pvt) Ltd’* intends to implement a software solution, there is resistance at the ground level due to an aging work force who have worked in a paper based setting for a long time period. Further, the workforce is finding it difficult to clearly define their requirements for the software due to the inexperience in IT solutions and the complexity of the activities. Further, the management expects the software to be implemented in a fixed and limited time period, as they are hoping to use the solution for their next set of projects, which are to be initiated soon.

a.

i. List **four** (4) challenges faced by the development team when providing a solution for the ABC (PVT) Ltd.

(4*2=8 marks)

ANSWER IN THIS BOX

1. Unclear requirements from the client and/or Difficulty for the workforce to clearly define the requirements for the software

2. Complexity of the project (Multiple projects and the same time, each with a different/varying set of activities)

3. Resistance at the ground level from an aging work force for an IT solution **(**Resistance for changing their existing paper based practices)

4. Limited timeline to develop a working solution

ii. For your answers in **a.(i)** above, explain why choosing an *agile method* for this project would suite, as compared to the traditional waterfall method.

(4*3=12 marks)

ANSWER IN THIS BOX

1. High potential for requirements to change in the current project.

Incremental development with regular releases which is a feature in agile methods, would therefore be suitable. Agile, welcome changing requirements, even late in development.

(Other accepted answers: High responsiveness for change in agile methods, Flexibility, Ability to scale as features of agile suited for this project)

2. Iterative and incremental development (any related advantage),

Collaboration with the user throughout the project cycle, would aid in dealing with the complexity of the project.

3. The need to engage with the workers in all stages of the project is a requirement of agile methods. Better awareness of the benefits, capturing the actual user requirement, visibility of project details etc. would aid to minimize resistance of the workforce.

(High level of collaboration with the user in agile methods)

4. Working software over comprehensive documentation favoured in agile methods, resulting in faster releases.

(or other relevant answers)

b. *Extreme Programing (XP)* is one the Agile Methodologies. Write down **five** (5) other Agile methods (methodologies and frameworks) which can be used in the above project.

(5*1=5 marks)

ANSWER IN THIS BOX

1. Scrum

2. Feature-driven development (FDD)

3. Rapid application development (RAD)

4. Agile unified process (AUP)

5. Adaptive software development (ASD) or any other correct framework

c. *'Personal success'* is one of the "three types of successes" which can be gained from an agile project. Briefly explain how the other **two** (2) successes would be beneficial in this specific project.

(2*3=6 marks)

ANSWER IN THIS BOX

1. Agile methods achieve **organizational successes** by focusing on delivering value and decreasing costs. This directly translates to increased return on investment. Agile methods also set expectations early in the project, so if your project won't be an organizational success, you'll find out early enough to cancel it before your organization has spent much money.

Agile Projects release their most valuable features first and release new versions frequently

When business needs change or when new information is discovered, agile teams change direction to match.

An experienced agile team will actually seek out unexpected opportunities to improve its plans.

Any other relevant answer.

2. If Extreme Programming is selected the below technical successes can be achieved

–XP programmers work together, which helps them keep track of the nitpicky details

–At least two people review every piece of code.

–Programmers continuously integrate their code

–The whole team focuses on finishing each feature completely before starting the next

–Create simple, ever-evolving designs that are easy to modify when plans change

d. Assume that it is proposed to use Agile methodologies in developing a solution for ABC (PVT) Ltd. List down **three** (3) limitations of using such methods in this project.

(3*3=9 marks)

ANSWER IN THIS BOX

1. Products with many external dependencies: In Agile/Scrum, dividing product development into short sprints requires careful planning; external dependencies, such as deliveries of software from other teams, can lead to delays and the failure of individual sprints

2. Teams whose members have very specialized skills: In Agile/Scrum, developers should be able to work on any task or pick up work that another developer has started. This can be managed by good leadership. While team members with very specific skills can and do contribute well, they should be encouraged to learn more about and collaborate with other disciplines.

Or

Teams whose members are geographically dispersed or part-time: In Scrum, developers should have close and ongoing interaction, ideally working together in the same space most of the time. While recent improvements in technology have reduced the impact of these barriers (e.g., being able to collaborate on a digital whiteboard), the Agile manifesto asserts that the best communication is face to face

3. Products that are mature or legacy or with regulated quality control: In Scrum, product increments should be fully developed and tested in a single sprint; products that need large amounts of regression testing or safety testing (e.g., medical devices or vehicle control) for each release are less suited to short sprints than to longer waterfall releases.

2. The following questions are related to the Scrum framework.

a. Briefly write down **three** (3) differences between the roles played by the *Scrum master* and a traditional *Project Manager*.

(3*2=6 Marks)

ANSWER IN THIS BOX

The scrum master is not a traditional team lead or project manager but acts as a buffer between the team and any distracting influences. The scrum master ensures that the Scrum framework is followed. The scrum master helps to ensure the team follows the agreed processes in the Scrum framework, often facilitates key sessions, and encourages the team to improve. <u>The role has also been referred to as a team facilitator or servant-leader to reinforce these dual perspectives, as opposed to the more authoritative project manager or team lead</u>.

Authority for a scrum master is not the same type of authority that a functional manager or project manager would have. For example, <u>the ScrumMaster doesn't hire and fire and cannot dictate to the team what tasks it should do or how to do them</u>. <u>The ScrumMaster also is not responsible for making sure the work gets done (Which usually a traditional project manager overlooks)</u>. <u>Instead, the ScrumMaster helps the team define and adhere to its own process for making sure the work gets done.</u>

Any 3 answers which reflect the above content.

b. Briefly explain the following **two** (2) artifacts of a Scrum project;

(2*4=8 Marks)

ANSWER IN THIS BOX

i. Product Backlog:

-The product owner is responsible for determining and managing the sequence of the work in scrum.
-This prioritized (or ordered) list is known as the product backlog
-The development teams break down each targeted feature in the product backlog into a set of tasks.
-This, forms a second backlog called the sprint backlog
The format of product backlog items varies, common formats include user stories, use cases, or any other requirements format the team finds useful.

ii. Sprint Backlog:

The sprint backlog is the list of work the development team must address during the next sprint

The list is derived by the scrum team progressively selecting product backlog items in priority order from the top of the product backlog until they feel they have enough work to fill the sprint.

Once a sprint backlog is committed, no additional work can be added to the sprint backlog except by the team

c. In Scrum, *"Don't Plan, Let's start the sprint and figure out what to do next!!!"* Write down **two** (2) points (justifications) on what the above phrase means by comparing it with the practices of a traditional (e.g. waterfall based) approach.

(3*2=6 Marks)

ANSWER IN THIS BOX

-Traditional Projects vs SCRUM Projects

-Traditional projects: Creates a detailed plan up front before development work begin.

-Get it right at the beginning so that rest can follow in an orderly fashion

SCRUM: The Scrum approach to planning is true to its empirical roots of inspection and adaptation. When developing using Scrum, we don't believe we can get it right up front, so we don't try to produce all of the planning artifacts up front. (We do, however, produce some of the planning artifacts early on in order to achieve a good balance between up-front and just-in-time planning.) Therefore we save the planning that is best performed in a just-in-time fashion for a time when we have much better information (e.g. may be after a couple of sprints are done)

d. Briefly explain the Envisioning (product planning) phase of a scrum framework, using the breakdown below;

ANSWER IN THIS BOX

i. The main goal and output of the phase (4 Marks)

-Used to create the initial product backlog- a product vision is needed to generate an initial product backlog

-The goal is to work upon an idea, describing the essence of the potential product and creating a rough plan for how to approach its creation.

-Can decide whether to fund the next level of more detailed development

ii. Three (3) participants for this phase (3 Marks)

The product owner
Internal stakeholders
Specialists in various areas
Ideally, the Scrum Master and the development team should participate

iii. Three (3) guidelines for economically sensible envisioning (3 Marks)

-Target a realistic confidence threshold
-Focus on a short horizon
-Act quickly
-Pay for validated learning
-Use incremental/provisional funding
-Learn fast and pivot

3. a. *'Cross functional teams'* are used in Extreme Programing (XP). Briefly describe **two** (02) reasons for the need to have people with skills of **software design** and **architecture** in such cross functional teams.

(2*2=4 Marks)

ANSWER IN THIS BOX

Everybody do coding in XP, Hence,

To maintain a **solid foundation and an architecture design** in the overall project

To help team members see ways of **simplifying complex designs**

Similar explanations to above mentioned answers. 2 marks for one complete answer.

b. List down **two** (02) main responsibilities of *on-site customers* in Extreme Programing (XP).

(2*2=4 Marks)

ANSWER IN THIS BOX

On site customers are responsible for defining the software the team builds

Release planning.

Evangelize the project's vision

Identify features and stories

Determine how to group features into small frequent releases

Create an achievable plan by coordinating with programmers

Manage risks

Similar explanations to above mentioned answers. 2 marks for one complete answer.

c.

i. Briefly describe *'The Last Responsible Moment'* in Extreme Programing (XP)?

(2 Marks)

ANSWER IN THIS BOX

The last responsible moment is the last moment at which you can **responsibly make a decision**, If commitments are delayed **beyond the last responsible moment, then decisions are made by default.**

Similar explanations to above mentioned answer. 2 marks for the complete answer.

ii. List down **two** (02) advantages that can be achieved by delaying the decision making process until the last responsible moment

(2*2=4 Marks)

ANSWER IN THIS BOX

Increase the accuracy of decisions

Decrease workload

Decrease the impact of changes

Similar explanations to above mentioned answers. 2 marks for one complete answer.

d. *Management support* is one of the pre-conditions (prerequisites) to adopt Extreme Programing (XP) in an organization. Briefly describe **three** (03) more such pre-conditions (prerequisites).

(3*2 = 6 Marks)

ANSWER IN THIS BOX

Team Agreement:
If team members don't want to use XP, it's not likely to work. XP assumes good faith on the part of team members there's no way to force the process on somebody who's resisting it.

A Colocated Team:
XP relies on fast, high-bandwidth communication for many of its practices. In order to achieve that communication, your team members needs to sit together in the same room

On-Site Customers:
They, led by the product manager, determine which features the team will develop. In other words, their decisions determine the value of the software.
Of all the on-site customers, the product manager is likely the most important.

The Right Team Size:
recommend 4 to 6 programmers and no more than 12 people on the team
even number of programmers
Teams with fewer than four programmers are less likely to have the intellectual diversity they need.

Similar explanations to above mentioned answers. 2 marks for one complete answer.

e. *Mismatched skills* is one of the challenges faced in *'Pair Programing'*. Briefly describe **two** (02) more such challenges faced in pair programing.

(2*2 = 4 Marks)

ANSWER IN THIS BOX

Comfort: Pairing is no fun if you're uncomfortable. Should consider; Positions, Equipment, Physical spaces, personal spaces, personal hygiene

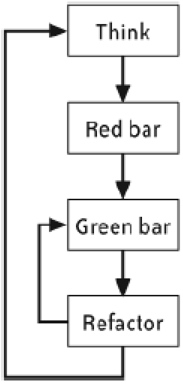
Communication style: New drivers sometimes have difficulty involving their partners; they can take over the keyboard and shut down communication. Too little communication and too much communication.

Tools and key bindings: Sometimes co-workers' tool preferences are annoying. Standardize on a particular toolset. Different coding standards.

Similar explanations to above mentioned answers. 2 marks for one complete answer.

f. The following figure shows the life cycle of 'Test Driven Development (TDD)'. Select **any three** (03) life cycle stages of TDD from the following diagram and describe their key activities.

(3*2 = 6 Marks)



ANSWER IN THIS BOX

Think: Imagine what behaviour you want your code to have, then think of a small increment that will require fewer than five lines of code. Next, think of a test—also a few lines of code—that will fail unless that behaviour is present.

Red Bar: Now write the test. Write only enough code for the current increment of behaviour (typically fewer than five lines of code). Code in terms of the class' behaviour and its public interface, not how you think you will implement the internals of the class. Respect encapsulation.

Green Bar: Write just enough production code to get the test to pass. You should usually need less than five lines of code. Just do what you need to do to make the test pass. Run your tests again, and watch all the tests pass. This will result in a green progress bar.

Refactor: With all your tests passing again, you can now refactor without worrying about breaking anything. Review the code and look for possible improvements. Ask your navigator if he's made any notes. For each problem you see, refactor the code to fix it. Work in a series of very small refactoring. Refactor as many times as you like. Make your design as good as you can, but limit it to the code's existing behaviour. Don't anticipate future needs, and certainly don't add new behaviour

Similar explanations to above mentioned answers. 2 marks for one complete answer.

****