**Software Engineering**

**Exam Answers**

SET08103

Matriculation: 40538519

Submission date: 02/05/2023

1. 1) Scrum –
   * Description:

Is a Team work and Project management process. Teams meet regularly in daily standup meetings and working in cycles called sprints. Work is checked and adapted as required regularly. The process is as follows:

* 1. Produce the backlog by defining each task to be completed - prioritise the tasks
  2. Estimate for each task compared to the others in backlog
  3. Decide which tasks are to be completed in sprint
  4. Start work on the sprint
  5. Meet with the team (Daily Scrum) to check the progress of the sprint
  6. Once sprint is complete, Review and reflect (can be 2 items in process list), what could have been done better, what went well? Learn lessons and adapt for next sprint.
  7. Check velocity of sprint (how much work was completed in sprint time)
  8. Plan next sprint (item 3)
* Value in DevOps:
  + 1st way - Maximizes flow of work by limiting work in progress as each team member only takes on as much as they believe they can complete. Reduces batch sizes by breaking work into small tasks to work on one at a time. Handoffs are also reduced.
  + 2nd way – Provides feedback by having regular standup meetings along with reviewing and reflecting after/before each sprint.
  + 3rd way – Continual learning and experimentation by enabling a learning and safety culture. Daily work should also include scheduled time to fix any bugs or issues.

2) Version Control –

* + Description:

Is about management of change. Developers can share code and work on the same project from different locations.

Tracks commits a change was made and by whom, making it easier to track and fix issues (rewind). Allows for development of multiple versions at the same time by means of branching. Once complete, these can be merged together.

* Value in DevOps:
  + 1st way – Maximizes flow by making work visible by allowing team members to see what has been done in the past and the current state of work.
  + 2nd way – principles of feedback, by swarm and solve. When there is an issue or bug, a branch can be created and everyone can work on that without affecting the main project. Once fixed this can be merged back into main/release/develop.
  + 3rd way – principles of continual learning and experimentation – Because version control allows for easy branching and easy rewinding to previous commits, it encourages experimentation on a new branch. This will not affect the original project if the experiment doesn’t work or gets abandoned.

3) Kanban –

* + Description:

Kanban is a task board to organize and view tasks for a project and allows everyone to quickly and easily know the current state of the task. It is a pull based system - Tasks/products are pulled through the system as required on demand. Limits work in progress Kanban allows responsiveness with Just-in-time manufacturing. Makes work visible

* Value in DevOps:
  + 1st way - Kanban boards can show the flow of work, left to right through development stages making work visible. It also limits work in progress by showing what is already in progress.
  + 2nd way – Kanban provides feedback by means of showing what state a current task is in so everyone can get a clearer picture of the project. Comments can also be added to tasks to provide additional feedback.
  + 3rd way - Institutionalizing the improvement of daily work by allowing easy scheduling of time to fix bugs/ defects and improve what the daily routine looks like.

4) UML –

* + Description:

UML is Unified Modelling Language and is a collection of General purpose diagrams for visualizing design of a system. It is split into two categories, Behavioral diagrams which show the functionality of the system, defines what happens and how objects interact, and Structural diagrams which show relationships between components or objects in a system and models the structure and architecture of a system.

* Value in DevOps:
  + 1st way – Work is made visible by diagrams showing how elements /components relate to each other.
  + 2nd way – UML diagrams provide feedback. Depending on the type of diagram this could be feedback used to optimize for downstream work centres. They could also help identify problems in the software which can then be swarmed and solved by the team.
  + 3rd way – Continual learning and experimentation – Ideas can be tested out in diagrammatic form and identifying any problems before moving further into development.

5) User Stories –

* + Description:

User stories are NOT requirements (something software will do), but provide similar capability. They define what software should do for the user / customer.

Consists of 3 parts:

1. User role (Who the story is about, detailed enough to talk about)
2. Activity (task user wants or requires)
3. Business Value (Why task is important)

Should also have a general definition of done, along with conditions of satisfaction - questions that need to be answered and conditions met to satisfy product owner. Should be broken down into tasks - pieces of work that can be completed by an individual.

Their main purpose is to view how a software feature will provide value to the customer by describing what the feature does from the users point of view.

* Value in DevOps:
  + 1st way – Helps to maximize flow by helping to reduce batch sizes. User stories help to break a project down into small components showing how the user interacts with a system. This can then be used to identify how to prioritize which components to develop and in which order. It can also help identify and elevate any constraints.
  + 2nd way – Principles of feedback. User stories can help identify how a user will interact and any components which will require particular attention.
  + 3rd way – Transform local discoveries into global improvements, User stories helps to learn how a user will interact and what the user wants. This can then be communicated throughout teams or the organization and then the knowledge applied to other areas.

1. a)
   * Confidentiality – Similar to privacy. Sensitive information / data should only be accessed / viewed by authorized personnel.
   * Integrity – Data should be trustworthy. This means data should be consistent and accurate. It should not get changed or modified unless this is intentional.
   * Availability – Authorized personnel should have access to data / information /resources at all times.

b)

* Disclosure – When information is viewed against the owners wish (attack against confidentiality)
* Alteration – When data is changed by an unauthorized person (attack against Integrity)
* Denial – When a users access to a system or service is hindered or blocked.

c)

* Asset – something valuable owned by a person or company such as passwords, servers, or documentation.
* Threat – A potential event which could cause loss of value such as an attacker taking control of an asset, or damaging / changing a file or password.
* Vulnerability – A weakness which potentially could allow a threat to happen. This could be by not verifying user data opening up the potential of an SQL injection attack.
* Risk – The combination of a threat and a vulnerability, leading to the increased chance of an attack if the risk is not mitigated. As the vulnerability exits such as missing authorization and the potential threat of an attacker gaining access to the system.

1. a)

* Obligation to society – All people have an interest in computing in some form. Computing professionals should do their best to contribute to society, having regard for public health, privacy, security and well being of others. The key idea is ‘do no harm’
* Reporting Misconduct – Is another person is known to be doing harm, then a computing professional has the obligation to report this as soon as possible so no harm comes to others.
* General Conduct – This is about professionalism and respect for others (other employees, clients, customers etc.).

b)

* Intellectual property are specific rights over conceptual creations given to individuals or companies.
* It includes copyright which is how others can use the work such as the right to print or reproduce a piece of work .
* It also includes patents which is the rights on what to exclude others from doing with the work. For example with Microsofts chatbox, the patent would exclude others from copying their code.

c)

* Asset – something valuable owned by a person or company such as passwords, servers, or documentation.
* Threat – A potential event which could cause loss of value such as an attacker taking control of an asset, or damaging / changing a file or password.
* Vulnerability – A weakness which potentially could allow a threat to happen. This could be by not verifying user data opening up the potential of an SQL injection attack.
* Risk – The combination of a threat and a vulnerability, leading to the increased chance of an attack if the risk is not mitigated. As the vulnerability exits such as missing authorization and the potential threat of an attacker gaining access to the system.

1. 1) Scrum –
   * Description:

Is a Team work and Project management process. Teams meet regularly in daily standup meetings and working in cycles called sprints. Work is checked and adapted as required regularly. The process is as follows:

1. Produce the backlog by defining each task to be completed - prioritize the tasks
2. Estimate for each task compared to the others in backlog
3. Decide which tasks are to be completed in sprint
4. Start work on the sprint
5. Meet with the team (Daily Scrum) to check the progress of the sprint
6. Once sprint is complete, Review and reflect (can be 2 items in process list), what could have been done better, what went well? Learn lessons and adapt for next sprint.
7. Check velocity of sprint (how much work was completed in sprint time)
8. Plan next sprint (item 3)

* Value in Lean:
  + - Reduces waste by not overcommitting as tasks can be seen on the task board and discussed at each scrum meeting. Also reduces the need for context switching.
    - Amplifies learning by increased feedback from regular meetings. If there are any issues, team members can easily swarm and solve. The task board and meetings also provide regular feedback as to the progress of the project tasks.
    - Allows decisions to be made as late as possible as the progress of tasks can be see on the task board as well as discussed continually at regular daily standups.
    - Delivery can be made as fast as possible with sprints being adapted as required (agile).
    - Meetings and the Scrum Master can also empower the team producing a happy and healthy environment and a more productive team.
    - The task board also allows being able to see the whole.

2) Version Control –

* + Description:

Is about management of change. Developers can share code and work on the same project from different locations.

Tracks commits a change was made and by whom, making it easier to track and fix issues (rewind). Allows for development of multiple versions at the same time by means of branching. Once complete, these can be merged together.

* Value in Lean:
  + - Eliminates waste by providing branching to fix defects. These can then be merged so everyone is working on the latest version.
    - Amplify learning by allowing team members to see what has happened in the past and what has lead to the current software state.
    - Decide as late as possible by having branches. These can be merged as and when required.
    - Deliver as fast as possible – by allowing multiple people to work on the same thing at the same time.
    - Version control can be tested regularly and versions can be rewound if needed allowing integrity to be built in.

3) Kanban –

* + Description:

Kanban is a task board to organize and view tasks for a project and allows everyone to quickly and easily know the current state of the task. It is a pull based system - Tasks/products are pulled through the system as required on demand. Limits work in progress Kanban allows responsiveness with Just-in-time manufacturing. Makes work visible

* Value in Lean:
  + - Helps eliminate waste by avoiding task switching as team members will only be assigned what they believe they can complete.
    - Avoids waiting as all tasks and their current states can be viewed easily.
    - Amplifies learning by providing feedback of a tasks status.
    - Decide as late as possible, as tasks can be monitored and discussed at each standup meeting.
    - Empower the team by giving the team purpose
    - See the whole as all tasks and their statues can be viewed on the kanban board.

4) Continuous Integration and Delivery - A process whereby everyone integrates work frequently and is tested and verified by an automated build and test process. This means errors can be identified and rectified sooner.

Values:

* + Risk is reduced
  + Manual repetitive tasks are reduced by automating build tasks
  + Software can be deployed at any point
  + Increased confidence in development of product

Process:

* + Commit changes
  + CI server receives changes
  + CI server builds software
  + Automated tests are run
  + Build and tests either pass or fail
  + Team notified of pass or fail

Continuous Delivery extends Continuous Integration and makes sure software can be released at any point. Releases are still manual which is different to continuous deployment which extends continuous delivery by automating releases.

* Value in Lean:
  + - Helps eliminate waste by automating tasks so avoiding extra processes.
    - Software is delivered faster allowing for faster feedback from the customer or client. This means it supports amplified learning.
    - Integrity built in – As tests are automated, they provided added trust in the software.
    - See the whole – Build light indicators allow the status of the project to be known.