**Software Engineering**

Activity on Version Control System

1. Create a github account
2. Search for a repository with the name Software Engineering\_Level\_300
3. Fork the repository to your own github account
4. Clone the repository to your own local machine.
5. Identify your class in the forked project from the groups folder.
6. Create a new file with your surname, index number and the .txt file extension eg: Surname\_100100.txt.
7. Create a new feature branch to work on.
8. Provide solution to the questions found in the “Qusetions.txt” file in the feature branch.
9. Commit your changes and push your new feature repository on Github.
10. Create a pull request to submit your change for review by the project owner.

Please make sure to follow every bit of instructions given instructions will be at your own detriment.

Please make sure to read every bit of instruction in the "instruction.txt" file

before you start answering these questions.

**Questions**

1. What is a software process model and why is it important in software development?

**Answer:**

A software process model is the mechanism of dividing software development work into distinct phases to improve design, product management and project management. It can also be said to be all the various stages involved in designing or building a new software.

Software Process Model is very important in software development because:

1. Software Process Model provides a distinct set of activities, actions, tasks and milestones (deliverables) required to engineer high quality software.
2. Software Process Model provides stability, control and organization to a process that if not managed can easily get out of control.
3. Software Process Model creates a form of roadmap for engineering work.
4. They help to adapt or adjust to meet the needs of software engineers and managers for a specific project.
5. Software Process Model create the avenue for engineers to work systematically in building or developing their softwares.

**2.** Compare and contrast the Waterfall and Agile software process models.

What are the key differences between the two models and

what are the benefits and drawbacks of each?

**Answer:**

Waterfall software process model is a sequential plan driven process where you must plan and schedule all your activities before starting the project itself whiles Agile software process model is developed over an iteration and each iteration is designed to be small and manageable so it can be completed within a few week.

Waterfall model is represented by a separate phase arranged in a linear order and produces one or more documents that need to be approved before the next phase, while Agile model involves customers in the development process and requires less documentation by using informal communication.

**Advantages of waterfall model**

1. **Clarity**

The strict method of development enables the developer to announce the deliverables and timelines of the project well in advance. It helps the managers to create an environment of clear communication with the clients and the development team.

1. **Documentation**

This methodology demands extensive documentation of each of the stages. This rigorous documentation eliminates or reduces the chances of misunderstandings or errors.

1. **Efficient Work**

If you follow this methodology, you would not be required to employ every one of your development team to complete one task. The method helps you to explain the tasks to the workers clearly, allowing each of them to concentrate on their job.

1. **Hands-off approach:**

The waterfall method moves the software development process in a hands-off customer approach. That means, after the planning and designing are over, the developers would not feel any presence of the customers till it is review time.

**Disadvantages of the waterfall model**

1. **Lack of Involvement of The Customers:**

While the hands-off approach can help the developers to work in relative peace, it can also create a negative impact. There are some projects where the involvement of the customer is necessary. If the customers do not like the hands-off approach, then the waterfall method can create significant issues.

1. **Difficulty in Making Changes:**

As this method follows strict steps, it is sometimes difficult to trace back to the previous stages to make certain changes. It can become an issue if the developer faces roadblocks. As you should not expect that the customers would understand the software right from the beginning, you will need to be flexible.

1. **Last Minute Testing**

Developing software is a time-bound process. In the waterfall method, if the coding procedure takes up more than the allocated time, the development team is compelled to compromise on the testing phase of the software to complete the project within the stipulated time frame.

**Advantages of agile model**

1. **Client Involvement**

One of the biggest advantages of using this method is the involvement of the clients. It makes sure that the end product always fulfils the requirements of the clients.

1. **Flexibility Regarding deliverables**

In this method, you can arrange the deliverables according to the importance of the tasks. For example, it allows you to release the basic version of the software even when the development process is not fully done.

1. **Adaptability**

Adaptability is one of the advantages of using the agile method. It helps you understand which portion of the software development is complete and where the developers still need to work. It also helps the developers to plan the sprints without facing many hassles.

1. **Customer-friendly**

It involves the customers in every phase of development.

**Disadvantages of agile model**

1. **High Commitment Levels**

It requires the full development team to commit to a single project. That can pose a huge challenge for the development team if they have to work on multiple projects.

1. **High Cost and Late Deadlines**

While the sprints allow the developers to plan the development process, it also enhances the chance of pushing the deliverable deadlines. If you need to concoct additional sprints for a project, then the cost of the project creeps up significantly.

1. **Communication**

As this method involves the customers in every phase of development, the managers need to keep in constant touch with them. That can hinder the workflow significantly.

All in all, both of these factors have respective advantages and disadvantages. Which among these two methods is suitable for your project entirely depends on the type of the project. You can consult with the adept professionals of Matrix Media Solutions to come to the final solution. We will not only help you choose the right methodology, we will also make sure that your project becomes successful.

**3.** What is Rational Unified Process (RUP), and

how does it differ from other software development process models?

What are the key phases and activities involved in RUP, and

what are the benefits and drawbacks of using this approach in software development projects?

**Answer:**

The rational unified process (RUP) is a software engineering and development process focused on using the unified modeling language (UML) to design and build software. Using the RUP process allows you to operate business analysis, design, testing and implementation throughout the software development process and its unique stages, helping you create a customized product.

**Inception**

In the inception stage of RUP, you communicate and plan the software concept or idea, evaluating what resources you need for the project and determining if it's viable. You use case modeling to identify the project scope, costs and time required to build it. If there are specific customer needs or requests for the software, you consider how to incorporate them effectively within the design plan.

Elements often included in the inception stage are:

* Risk assessments and project plans
* Vision or mission statements
* Financial projections and business models
* Prototype development

**Elaboration**

During the elaboration phase, you further evaluate the resources and costs needed for the project's full development, creating actionable and executable baseline architecture of the software. This detailed stage aims to diminish cost totals and risk and produce a revised use case model. You compare the software projections against the established milestones and project criteria. If there are discrepancies,

* Elements often included in the elaboration stage are:
* Use case model
* Viable software architecture
* Risk reduction plans
* Use manual

**Transition**

The transition stage releases the project to the user, whether that's the public or internal users like employees. A transition phase is rarely perfect and often includes making system adjustments based on practical and daily usage. Ensuring a smooth transition and rectifying software issues timely can help make this stage a success.

Elements often involved in the transition period include:

* Beta testing
* Education and training
* Deployment and data analytics
* Collection of user feedback

**Production**

This last phase of the RUP process includes software deployment, intending to gain user acceptance. You maintain and update the software accordingly, often based on feedback from people who use the software, app, program or platform.

This last stage usually includes:

* Packaging, distribution and installation
* User help and assistance platform availability
* Data migration
* Continued user acceptance initiatives

**Advantages of RUP**

RUP can provide software development or design teams an array of advantages, including:

1. Offering thorough documentation: The RUP process involves carefully documenting each step, which can be highly beneficial for collaborative projects.
2. Enhancing risk management RUP can help software individuals proactively respond to potential software challenges. This can improve risk management and troubleshooting efforts.
3. Giving regular feedback to stakeholders: A vital part of the RUP process is giving consistent updates to project stakeholders. These stakeholders may range from other software individuals involved in the project to company leaders or vendors.
4. Reducing total project time: RUP may allow the software development team to lower their time in both the development and integration stages.
5. Determining working elements early on in the project: With RUP, project stakeholders may notice potential software issues earlier on during the design or development processes. This can make mitigating or solving challenges easier before they become more complex.
6. Potential drawbacks of RUP
7. Using RUP can come with some possible disadvantages. If you or your team plans to use RUP, it's important to prepare for these potential issues so you can proactively navigate challenges.
8. Following are some of the potential drawbacks of RUP, plus ideas about how to overcome them:
9. Complexity of process: Since RUP is a complicated procedure, successfully performing it requires software team members with great expertise. If some of the individuals on your software team are new to the field, it might be easier to choose a different software development process.
10. Cost and time: The amount of documentation required for RUP can be time-consuming and expensive. Software teams with smaller budgets might benefit from choosing a more cost-efficient approach for their project.
11. Challenge of using it for projects with multiple development streams: RUP may cause confusion during the testing stage for larger projects involving multiple components and software teams. Because of its emphasis on ongoing integration, those working on projects with multiple development streams may want to either slow down the RUP process or look for another development procedure.