UNIVERSITI TUNKU ABDUL RAHMAN

ACADEMIC YEAR 2020/2021

MAY 2020 TRIMESTER

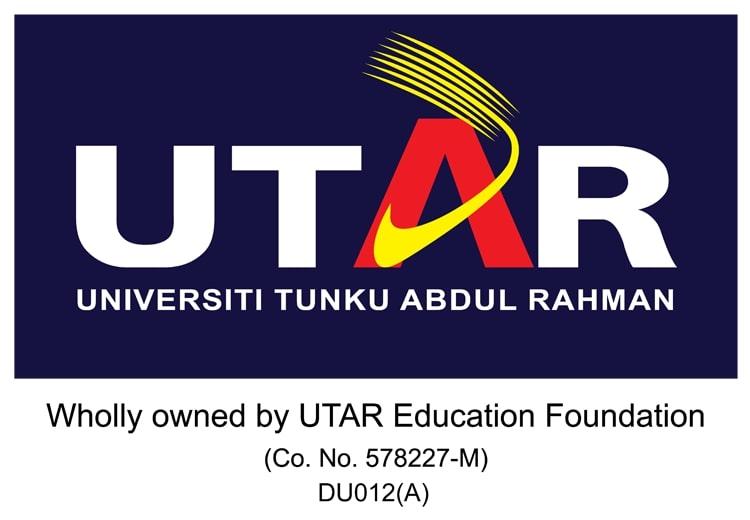
FINAL ASSESSMENT

**ANSWER SCRIPT**

**Candidate is required to fill in ALL the information below:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name : (as stated in Student Identity Card) | Tan Ying Yao 1703648 | | |
| Faculty /Institute/ Centre: | LKC FES | Programme : | SE |
| Index No. (in numbers) : | U00463EBSEF | Index No. (in words) : | UZEROZEROFOURSIXTHREEEBSEF |
| Course Code : | UECS2363 | Course Description : | Software Construction and Configuration |
| Submission Date : | 1/10/2020 | Time : | 0830 |

|  |  |  |
| --- | --- | --- |
| **QUESTION NUMBER** | **FOR EXAMINER’S USE ONLY** | |
| **MARKS** | |
| **Internal** | **External** |
| **1** |  |  |
| **2** |  |  |
| **3** |  |  |
| **4** |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| **TOTAL MARKS** |  |  |



**DECLARATION STATEMENT**

I, Tan Ying Yao (Name), Student ID No. 1703648 hereby solemnly and sincerely declare and confirm that I have read, understood and shall abide and comply with all laws, rules, regulations, guidelines and lawful instruction of the University and its staff in relation to the commencement of any assessment / examination during my programme of study in Universiti Tunku Abdul Rahman.

I hereby declare that my submission for all assessment / examination during my programme of study in the University shall be based on my original work, not plagiarised from any source(s) except for citations and quotations which have been duly acknowledged. I am fully aware that students who are suspected of violating this pledge are liable to be referred to the Examination Disciplinary Committee of the University.

Tan Ying Yao

Name: Tan Ying Yao

Student’s I.C. / Passport No.: 971210-14-5673

Date:1/10/2020

Index Number (in figure): U00463EBSEF Course Code UECS2363 Page: 1

1)

a)

The main advantage of distributed version control system as compared to centralized version control system is that it better supports large open source efforts. There are no central server or repository and all developers have their own code locally. This allows users to work with maximum productivity when not connected to a network. Consistency is maintained via network transfers and there is no need to communicate changes as there is no need to link to a central server. Distributed version control system can also allow for various development branches to be created and worked on.

b)

Each member is in different regions and it is physically impossible for developers to meet and share work together. DevOps can help improve collaboration between developers and centralise work process. We will be using tools that helps facilitate the DevOps process with application builder such as GitHub. GitHub serves as a distributed version control and source code repository for use to work around with each other. The planning of the project is done via Google Docs and communications through Microsoft Teams. Google Docs and Teams can help solve the issue of miscommunication between members to ensure the work process is unified and explained well. The Continuous Integration tools used is Jenkins which serves as an open source automation server to build and test the software. Next, we will deploy the software using Amazon Web Services as our cloud computing services in order to publish it to the web. The operational tools used will be AppDynamics as it can be used to manage the performance and availability of our applications across cloud computing environments. Finally, our continuous feedback tool will be Sauce Labs as it can be used for automated testing platform. In order to push out the application quickly and effectively, we will be adopting the agile methodology. Agile can be used to quickly deliver the software and involves the customers and developers. Late changes in requirements can be fit and fixed into the software if necessary.

Index Number (in figure): U00463EBSEF Course Code UECS2363 Page: 2

2)

a) Refactoring typically involves making structural changes to the source code without changing the external behaviours of the software. Having unit testing allows a developer to refactor without breaking the overall structure. Test Driven Development should be practiced as it uses automated unit testing to decouple the software. This results in a comprehensive suite of unit testing that can be run at any time to provide proper feedback on the software compatibility. Automated testing refers to the usage of a software to control the execution of tests. Automated testing usually needs to be run repeatedly in order to help perform “compatibility testing” on different configurations and platforms. Long term costs are reduced and it is possible to run regressions on codes that is continuously changing and in shorter time. However, not all test can be automated and some tests have to be done manually. There are a few types of automated testing such as monkey testing, capture and unit testing. Monkey testing randomly selects input from a large range of values and monitor exceptions. Capture testing captures user inputs and stores it into a script to test later. This can ensure repeated testing quickly but is difficult to determine specifics of bugs. Unit testing are individual unit of source code used to determine if they are usable. This can help find bugs in the early stages but not effective for integrated system testing.

b)

//using Junit as test

public class TestCalculator{

@Test

public void testAddTwoNumbers() {

Calculator calc = new Calculator();

int txtResult = calc.add(button8, button4);

assertEquals(12, txtResult);

}

}

Index Number (in figure): U00463EBSEF Course Code UECS2363 Page: 3

3)

a)

Container should be used if the biggest priority is maximizing the number of applications running on a minimal number of servers. Containers are used in situations that requires lightweight and native performance. It can help reduce IT management resources and faster application development. Containers are used when all containers share the host operating system. It also requires less memory space but is less secure. Containers are quick, require less memory and relatively lightweight but requires all containers to share the host operating system and less secure. Virtual machines should be used for applications that require all of the operating system’s resources such as running multiple application on a server. Virtual machines are used in situations that require heavyweight and has limited performance. Each virtual machine runs in its own operating system and require heavy resources. Virtual machines should be used in situation where security is the most important requirement. Virtual machines are slow, require more memory and relatively heavyweight but is fully isolated and much more secure.

b)

i) Travis CI is used to build and test software projects hosted. It can be used to reduce overall development cost by making it easier to find and fix issues. It also provides valuable and timely information by letting the development be managed more tightly. Travis has a myriad of features such as automatic integration with GitHub along with repository access to pull requests. It comes with pre-installed build & test tools and offer multiple services such as deployment to multiple cloud services.

ii) Vagrant is an open-source software used for building and maintaining portable virtual software development environments such as VirtualBox. Vagrant can help in sharing setups between team members by using the same work environment. It can be used to test how deployment will work before pushing changes to other environments. It provides a simple solution for testing an environment on a local machine

iii) Git is a distributed version-control system for tracking changes in source code during software development. GitHub is used as a distributed version control and source code management with the functionality of Git. Both of them helps improve teamwork among multiple developers and allow early detection of errors in code.

Index Number (in figure): U00463EBSEF Course Code UECS2363 Page: 4

4)

a) Dark Launching refers to the strategy of deploying the first version of functionality into production before releasing the functionality to all users. It can deploy new versions of the software continuously while not releasing all the features to old users. It is used to find bugs more easily before releasing a buggy build to the public. It can provide an approach to remediate in a low-risk solution. Problems of an early version can be avoided as only a few users may experience the issue. The incident can be reverted by just switching off the feature if it is necessary.

b) One example of software that should adopt “Dark Launching” is popular social media application such as Instagram. A social media application such as Instagram can utilise the dark launching deployment in order to test the first version of functions to a specific few. This can be used to find bugs more easily before releasing a buggy build to the public. This provides a low risk approach to gauge the audience interest before fully pushing the feature fully. Bugs and errors can be avoided as not all users will experience the issue. This is achieved by feature toggling in which the feature can be removed by “turning” it off. Another example of software that can benefit from Dark Launching is communication applications such as WhatsApp. As the functionality of the application is the utmost importance, new functions had to be monitored closely in order to not affect the user’s experience. This can avoid too many bugs from emerging as the function is only available to the selected few. The popularity of the new function can be tested by observing the opinion of the public in order to see if it will make it to the final build. Finally, one software that can use the Dark Launching method is media streaming application such as YouTube. YouTube has faced plenty criticism in its inability to listen to its audience and push out new redesign that had been met with negative reception. Dark launching a new feature can help avoid this outcome by understanding what the users want and need by first testing itself to a few. This can help avoid public backlash as the features are tested more and implemented to measure its interest to users. This low risk solution can prevent users from leaving negative reviews and impression of the application.

**References**

Docker vs Virtual Machine - Understanding the Differences - Geekflare. (2020). Retrieved 1 October 2020, from https://geekflare.com/docker-vs-virtual-machine/