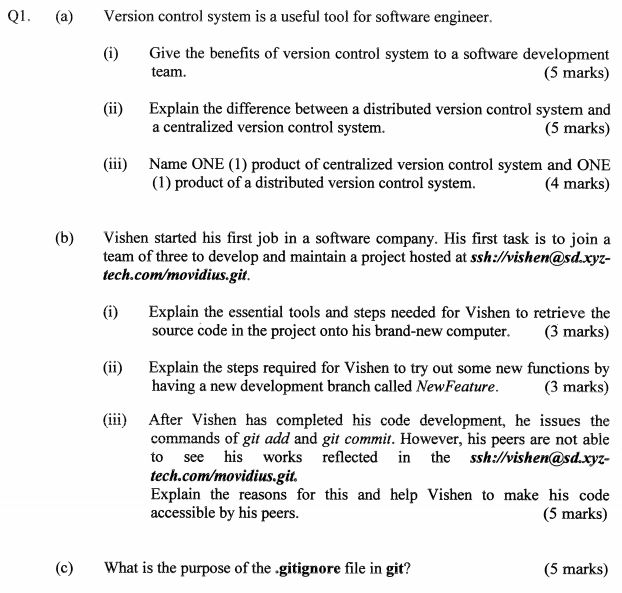
1)



A)

i) Benefits of VCS

- working independently with own backup in case of errors

-great support for diff version of same project

-simplifies concurrent works and merging changes when working with others

ii)Distributed VCS VS

Becoming the de facto way λ Better supports large open source efforts λ No central server/repository λ All developers have all code locally λ Consistency is maintained via network transfers

Centralized VCS

λ Client / server architecture λ Central server is THE code repository λ Developers run CVS clients, transfer files across network to local store for work λ Commit copies files back to central repository λ Detect conflicts for resolution

iii) Name one CVCS and DVCS

CVCS: Subversion ; DVCS: Git

B)

i) Essential tools and steps to retrieve source code

cd to project directory

git init (to create repository of the code)

ii) Steps to try out new functions by having a new development branch called NewFeature

mkdir NewFeature

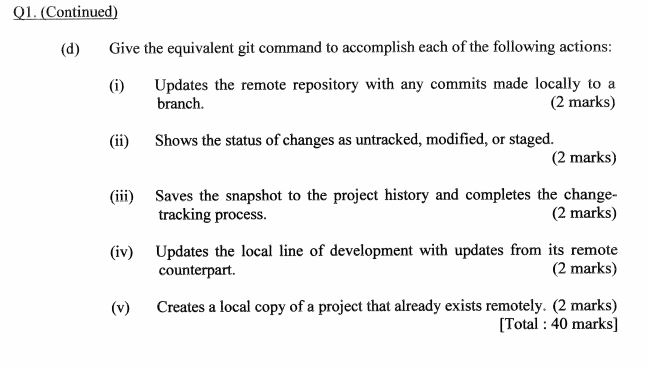
cd NewFeature

git init

git add

iii) Code not accessible, explain why

C) .gitignore file in git purpose: The **purpose** of **gitignore** files is to ensure that certain files not tracked by git remain untracked



D)

i)

git commit -a -m “new feature”

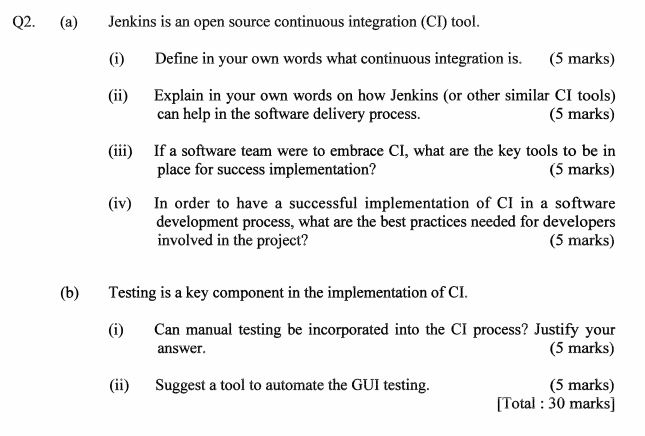
git push origin master

ii) git status

iii)git commit -m “new feature”

iv)git pull origin master

v) git clone remote\_repo

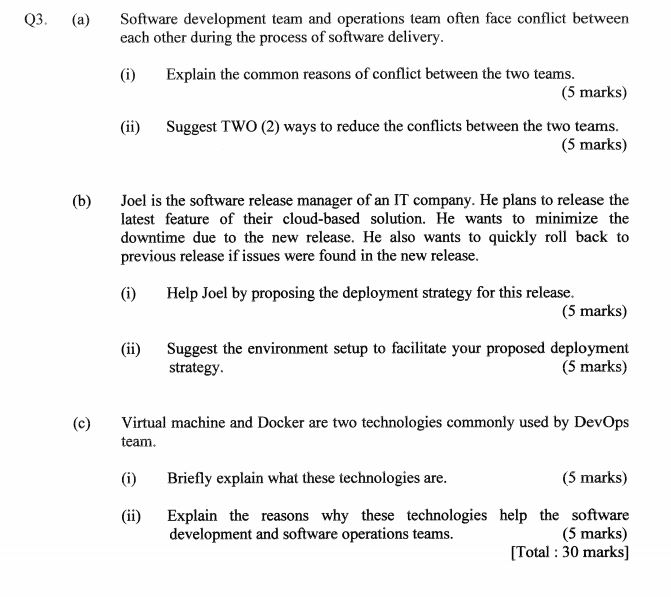
  
2) a) i) Continuous integration (CI) is a practice in software engineering of merging all developer working copies with a shared mainline frequently. Its main aim is to prevent integration problems.

ii) CI allows early detection of errors in the codes λ CI removes integration sessions λ CI minimizes the integration bugs λ If you build and test your software once an hour, no problem is more than an hour old. λ CI improve team works λ CI delivers latest best build products λ CI reduces overall development cost λ making it easier to find and fix problems λ provides valuable and timely information, letting the development be managed more tightly

iii) Single source code repository 2)Automated build scripts 3)Automated tests 4)Developers’ disciplines λ Synchronize often λ Don’t break the build λ When you break the build, fix it

iv) Don’t check in on a broken build λ Always run all commit tests locally before committing, or get your CI server to do it for you λ Wait for Commit tests to pass before moving on λ Never go home on a broken build

B)i)

  
3) A) i) Dev Team: Need for change λ Development produces changes, λ e.g.: new features λ bug fixes λ work based on change requests, etc. They want their changes rolled out to production.

Ops Team: Fear of change λ Once the software is delivered, the operations department wants to avoid making changes to the software to ensure stable conditions for the production systems.

ii) A mix of patterns intended to improve collaboration between development and operations. λ Addresses shared goals and incentives as well as shared processes and tools. λ Respects specific cultures, that people are more important than processes, which, in turn, are more important than tools. λ Accepts the inevitability of conflicts between development and operations

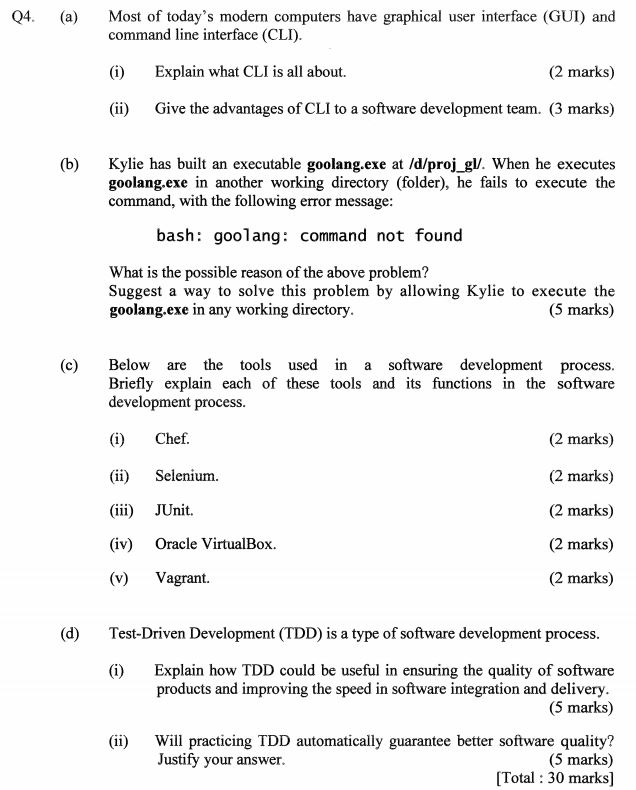
B) i)

ii)

C)

i)

ii)



4) A) i) Command Line Interface

ii) <https://www.teach-ict.com/gcse_new/computer%20systems/user_interface/miniweb/pg3.htm>

b) <https://devconnected.com/command-not-found-in-bash-fixed/#:~:text=Most%20of%20the%20time%2C%20you,in%20your%20PATH%20environment%20variable.&text=As%20you%20can%20see%20here,in%20my%20PATH%20environment%20variable>.

c)i) <https://en.wikipedia.org/wiki/Chef_(software)>

ii) <https://www.selenium.dev/>

iii) <https://junit.org/junit4/>

iv) <https://www.virtualbox.org/wiki/VirtualBox>

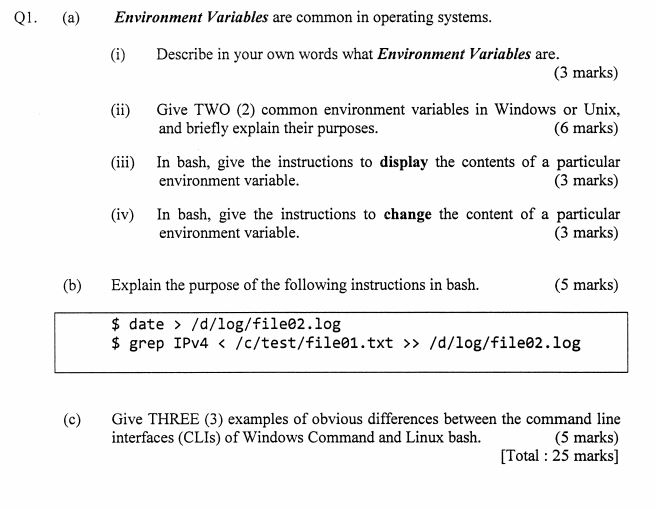
v) <https://www.vagrantup.com/>

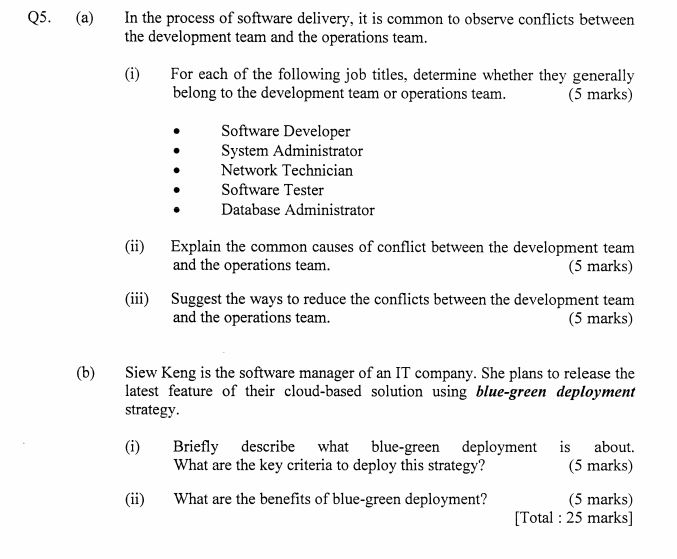
d)i) TDD is an advanced technique of using automated unit tests to drive the design of software and force decoupling of dependencies. λ The result of using this practice is a comprehensive suite of unit tests that can be run at any time to provide feedback that the software is still working.

ii) The suite of unit tests provides constant feedback that each component is still working. λ The unit tests act as documentation that cannot go outof-date, unlike separate documentation, which can and frequently does. λ When the test passes and the production code is refactored to remove duplication, it is clear that the code is finished, and the developer can move on to a new test.

λ Test-driven development forces critical analysis and design because the developer cannot create the production code without truly understanding what the desired result should be and how to test it. λ The software tends to be better designed, that is, loosely coupled and easily maintainable, because the developer is free to make design decisions and refactor at any time with confidence that the software is still working. This confidence is gained by running the tests. The need for a design pattern may emerge, and the code can be changed at that time.

The test suite acts as a regression safety net on bugs: If a bug is found, the developer should create a test to reveal the bug and then modify the production code so that the bug goes away and all other tests still pass. On each successive test run, all previous bug fixes are verified. λ Reduced debugging time!



  
5) A) i)

DEV: ST, SD

OPS: SA, NT, DA

b)i) This strategy deploy the new version of the application side by side with the old version. λ To switch over to the new version or roll back to the old version, back and forth, we merely have to change a load balancer or router setting. λ Blue-green deployment ensures that you have two production environments that are as similar as possible.

At any one time, one of them (e.g., the green environment) is live. While bringing a new release of your software to production, you conduct the final steps of testing in the blue environment. λ Once the software is working in the blue environment as expected, configurations are done, and smoke tests are run successfully, we switch the router to redirect all incoming requests to go to the blue environment.

Afterward, the green environment is out of production and can be used to prepare the next release.

