

# SE489 DevOps Engineering

## Assignment 2



### Assignment 2: Version Control Systems

1. What is Version Control Systems (VCS)?
2. What is the role of Version Control Systems (VCS) in DevOps?
3. In order to minimize risks to the business within DevOps, what is a main goal of version control?
  - a) Ensure the ability to alert when configuration changes from the desired state
  - b) Ensure the ability to re-create the previous state of the test environment
  - c) Ensure the ability to re-create the production environment and build processes
  - d) Ensure the ability to share the source code between different developers teams
4. Please consider the following elements:
  1. Change requests
  2. Deployment pipeline tools
  3. Compiled program executables
  4. Tutorials and standards

Which two elements are typically stored in a single shared source code repository?

- a) 1 and 2
- b) 1 and 4
- c) 2 and 3
- d) 2 and 4

5. Consider the anatomy of a basic Deployment Pipeline. Which stage asserts that the system works at the functional and non-functional level?
- a) Automated acceptance test
  - b) Build and unit test
  - c) Manual acceptance test
  - d) Version Control
6. Even the smallest applications will have a dependency on other components or libraries. Therefore, understanding and managing dependencies is a key activity within Continuous Deployment in order to keep flow within the Deployment Pipeline. You have built an application that uses two libraries. Each of these libraries rely on a third, underlying library, although they refer to different versions. This creates a specific dependency. What is the best solution to solve or prevent this dependency?
- a) Assemble all libraries into a single library, so that you can refer to the library directly and prevent the problem
  - b) Manage the libraries by using Version Control, so that you see it directly if you create this type of dependency
  - c) Keep a visual overview of all your dependencies on sticky notes on a big board, so you can track the flow
  - d) Only check in small parts of your toolchain, so that you can easily debug problems that may occur upon check-in