**SSW567 Homework 02**

Docker and Configuration Management

Ziyu Zhang

1. Integration (when developers integrate code from multiple sources)

Description: Multiple sources means the code might from different physical environment, software environment. Maybe the different code files from different sources have different file system. This situation forces developers to build up a new environment every time to adjust to those code.

Docker: Docker offers a container image allow developers. Assuming every developer uses a image with same environment, even code might from different people, all files can run in the same container. Furthermore, it is far more easier for a developer to build up a new container with different parameters than build up a new environment.

1. Testing

Description: The environment between testing and developing might different.

Docker: Docker ensures consistent environments from development to production. That is to say, we can use a same container with the same environment from development to testing.

1. Deployment

Description: The environment might change during product’s release cycle.

Docker: Developers, maintainers can easily change the necessary to containers, test them, and implement the same changes to existing containers.

1. Customer support

Description: The customers might want to add new features into the application.

Docker: With docker, developers are able to ensure to pass the same container from build to test process all the way.

1. Internal and external hackers who might attack the system

Description: hackers might want to find some loopholes (bugs, vulnerabilities) to manipulate local memories in order to damaging our computer.

Docker: Docker ensures that applications that are running on containers are completely segregated and isolated from each other, granting developers complete control.