Obligatory Assignment # 5

Continuous Build System

IMT 3501

Joakim Harbitz (120924), Jan Samuelsen Lindemann (120926), Alexander Gausdal (120927), Anders Storsveen (120928)

November 5, 2014

Abstract

This document contains the design of a continuous build system with extra emphasis on security.

Contents

Chapter 1

Introduction

Obligatory assigment # 5:

- Sketch a continuous build system using a private/hybrid cloud for HiG/IMT. Consider the following functional requirements: a) Students submit code to a common repository, b) Build system compiles source to binaries, c) Teachers retrieve source code+binaries, d) Deployment to web servers and app stores, e) Automated functional testing

- What are the legal, organizational, and technical security requirements in software integration?, a) Requirements of students?, b) Requirements of teachers?, c) Requirements of system administrators?, d) Requirements of the institution? You work in groups of up to 6 people. Partition the tasks and submit a common report. Specify in the report who is responsible for which section.

Chapter 2

Design

Anders Storsveen & Alexander Gausdal

The use case diagram in figure 2.1 is a representastion of a users interaction with the system. Build system, Test and web server/app store are external systems which we will not describe in detail.

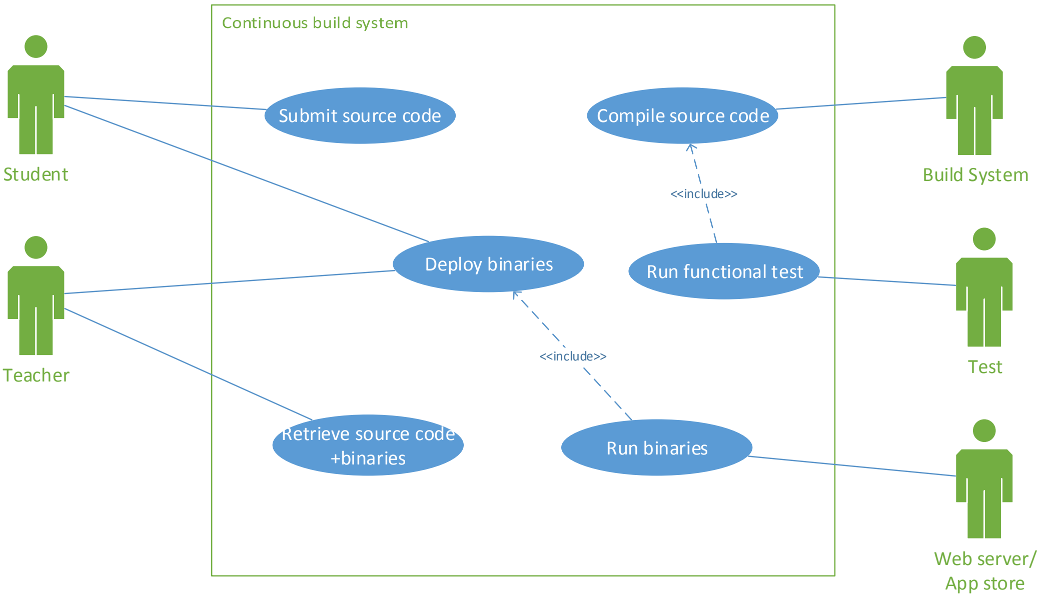

Figure 2.1: Use Case Diagram

2.1 High level use case

Use Case:	Submit source code
Actors:	Student
Goals:	Submit source code to common repository.
Description	Student uploads his/her code to a cloud based continuous build server.

Use Case:	Deploy binaries
Actors:	Student, teacher
Goals:	Deployment to web servers and app stores
Dependencies	Requires compiled source code
Description	The student/teacher takes the compiled code and deploys it to the web server/app store.

Use Case:	Compile source code
Actors:	Build System
Goals:	Compile source to binaries
Description	Send source code to build system, and get binary in return.

Use Case:	Run functional test
Actors:	Test
Goals:	Automate functional testing
Description	Runs functional tests on the compiled code before deployment.

Use Case:	Run binaries
Actors:	Web server/App store
Goals:	Make the application available through the web server/app store
Description	The web server/app store publishes the binary, and makes it available for downloads.

Use Case:	Retrieve source code+binaries
Actors:	Teacher
Goals:	Teacher can inspect the students code
Description	Lets the teacher read and comment the published code

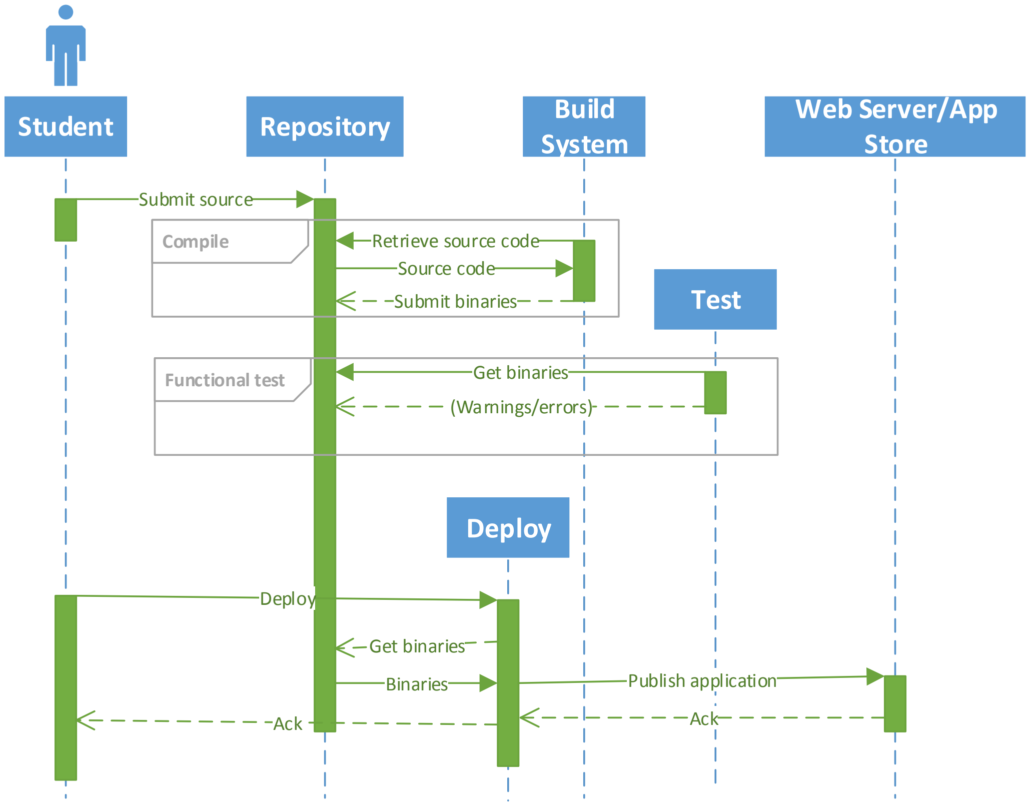

Figure 2.2: Sequence Diagram

2.2 Sequence Diagram

The sequence diagram in figure 2.2 is an interaction diagram that shows how processes operate with one another and in what order.

2.3 Class diagram

The class diagram in figure 2.3 is a static structure diagram that shows the system's classes and the relationships between them. Test, Web server/App store and Build system is not represented in the diagram because they're external components that's not a part of our internal system.

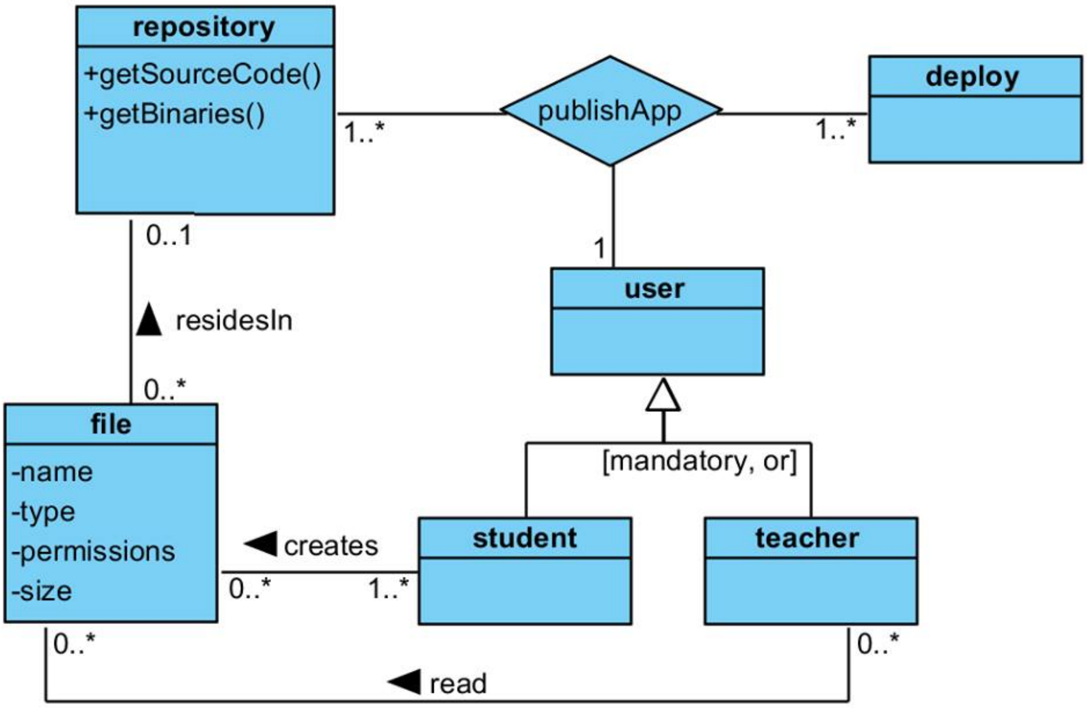

Figure 2.3: Class Diagram

Chapter 3

Requirements

3.1 Legal

Joakim Harbitz

The system has to comply with existing laws and regulations that already exists. This includes the personal data act [6]. The purpose of this law is to protect the privacy of the individual using the system by ensuring that the personal data and source code submitted to the system will not be exposed to the public. The system may have its own rules that the user has to comply to. For our system, this means that the user have to accept that their private source code may be inspected and analyzed to ensure that it does not include malicious code that can harm the system or their users downloading the binaries from the web or app store

3.2 Organizational

Jan Samuelsen Lindemann

The HiG information-security policy [1] and principles [2] covers several requirements in regards to the proposed system we are describing, the following ones are those that seemed relevant in the system.

3.2.1 IS-prinsipper 1.4.1

> *1.4.1 Informasjon, infrastruktur og rutiner skal tilpasses og klassifiseres i henhold til nødvendig sikkerhetsnivå og behov for tilgang.*

This means that the system owner and/or the system adminisrtation might have to administer different privilege levels, depending on what sort of access the different users might need. According to Saltzer/Schroeder [3], the principle of "Least privilege" means that:

> *Every program and every user of the system should operate using the least set of privileges necessary to complete the job.*

3.2.2 IS-prinsipper 1.4.7

> 1.4.7 Informasjon skal klassifiseres i tre kategorier
> **ÅPEN**: Informasjon som ikke inneholder intern eller sensitive opplysninger.
> **BEGRENSET**: Informasjon som regnes som intern og kan være skadende for Høgskolen i Gjøviks omdømme eller ikke er passende for en tredjepart. Systemeier avgjør behandling og lagring.
> **KONFIDENSIELL**: Sensitiv informasjon hvor uautorisert tilgang kan medføre betydelig skade for enkeltpersoner, høgskolen eller deres interesser. Konfidensiell er synonymt med forvaltningslovens ”Unntatt offentlighet”.

In most cases the information stored on the system would probably fall under the first category, OPEN, however, there might be situations where student-work submitted to the system has been done on behalf of GUC/IT-service(IT-tjenesten) and this might be categorised as LIMITED or even CONFIDENTIAL.

3.2.3 IS-prinsipper 1.5.2

> 1.5.2 Høgskolen i Gjøvik skal ha adekvate retningslinjer for behandlig, lagring og transportering av studentprodusert data.

On this point i was unable to locate any such document online on the HIG website, however I am a bit unsure if it in this case is aimed towards research-data "Akademiske data" and if this might not cover the use-case scenarios of our system.

3.2.4 IS-prinsipper 1.8

> 1.8.1 Informasjonssystemer skal ha retningslinjer for tilgangskontroll med adekvat loggføring.
> 1.8.2 Autentisering skal, hvis ikke avvik er nødvendig, utføres via sentrale fellessystemer for administrasjon, autentisering og autorisasjon av brukere og tjenester.
> 1.8.3 Systemeiere har ansvar for at systemet konfigureres med adekvat tilgangskontroll og loggføring etter retningslinjer godkjent av rektor.

This covers number 3 on the list of design principles for secure software by Saltzer/Schroeder [3], "Complete mediation",

3.2.5 IS-prinsipper 1.10

> Kravspesifikasjoner på systemer og tjenester skal inneholde krav til informasjonssikkerhet der det tas høyde for risikoer ved implementasjon og utvikling.
> 1.10.2 Det skal finnes retningslinjer for loggføring og endringshåndtering av alle systemer i produksjon.
> 1.10.3 Utvikling, test og vedlikehold/drift skal separeres for å forhindre uønskede feilsituasjoner.

The system is supposed to functions a deployment platform for HIG/IMT and will therefor probably have to deal with a huge ammount of varying types of applications, from the repository to the testing. Logging and testing of the system itself is very important to remediate any possible incidents.

3.3 Technical Security

Jan Samuelsen Lindemann

It's advicable that the developers, during the development, following industry approvied guidlines/best practices. Examples of resources for the current best practice in development:

- SALTZER/SCHROEDER: Design principles [3]
- OWASP top 10 [4]
- CERT Top 10 Secure Coding Practices [5]

While OWASP is specifically targeted towards web applications, the principles can be applied to software-development in general.

Bibliography

[1] Christoffer Hallstensen, HiG, *Informasjonssikkerhetspolicy*. 29.05.2012 .
hig.no/content/download/35317/431879/file/ispolicy.pdf

[2] Christoffer Hallstensen, HiG, *Informasjonssikkerhetsprinsipper*. 29.05.2012.
http://hig.no/content/download/35318/431882/file/isprinsipper.pdf

[3] JEROME H. SALTZER/MICHAEL D. SCHROEDER, IEEE, *The Protection of Information in Computer Systems*. 1975.
http://www.acsac.org/secshelf/papers/protection_information.pdf

[4] OWASP, *Top 10 2013-Top 10*.
https://www.owasp.org/index.php?title=Top_10_2013-Top_10&oldid=181201

[5] CERT, *Top 10 Secure Coding Practices*.
https://www.securecoding.cert.org/confluence/display/seccode/Top+10+Secure+Coding+Practices

[6] *Personopplysningsloven*.
https://lovdata.no/dokument/NL/lov/2000-04-14-31#KAPITTEL_1