BDSA Assignment 3

Gustav Espersen gues@itu.dk Silke Bonnén ssbo@itu.dk
Viggo Starcke vist@itu.dk

September 29, 2022

GitHub Repository

https://github.com/viggostarcke/BDSA-Assignment-3

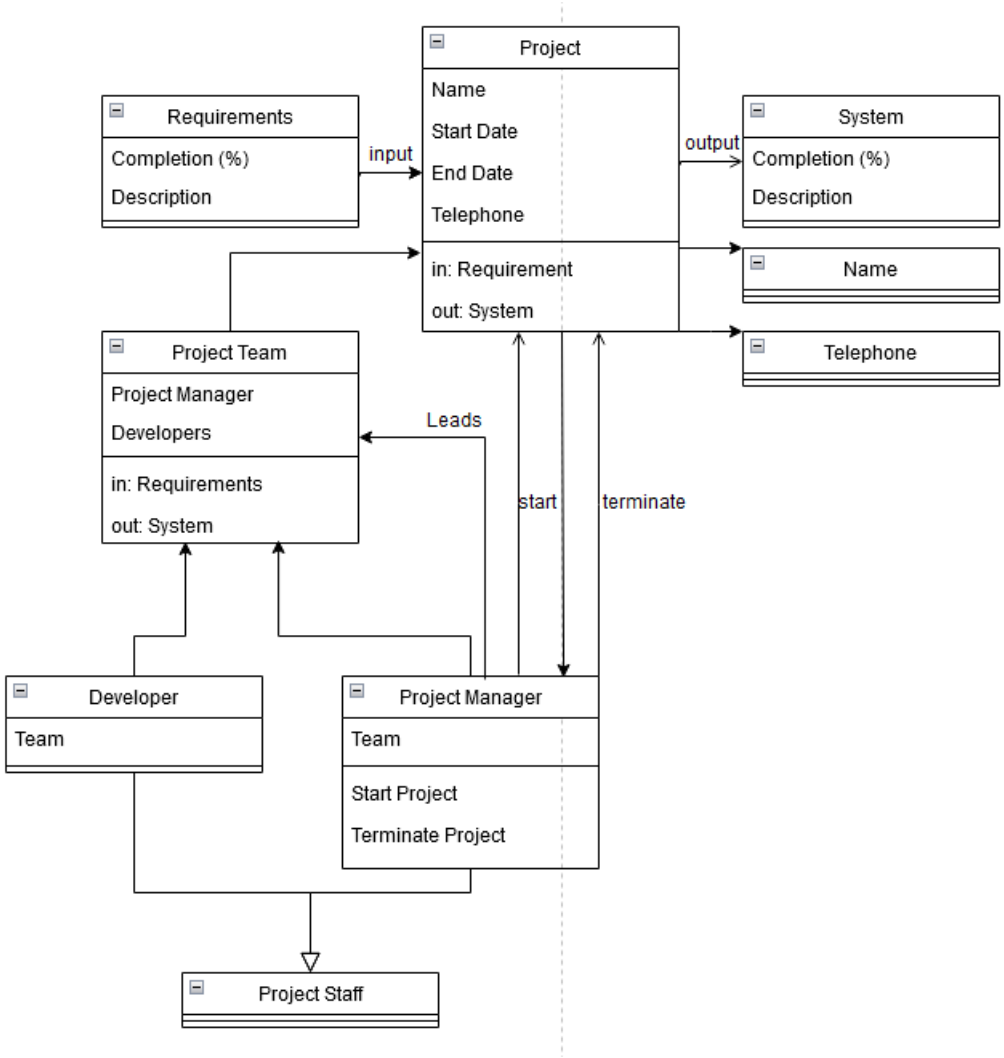
Software Engineering

Exercise 1

What level of detail should UML models have? There are different types of UML diagrams which represent the different levels of detail. One could think these models are to give an easily comprehensible insight of the system, which is correct. But not at the cost of lackfulness when it comes to information. Generally you want a high enough level of detail to ensure nothing is misunderstood.

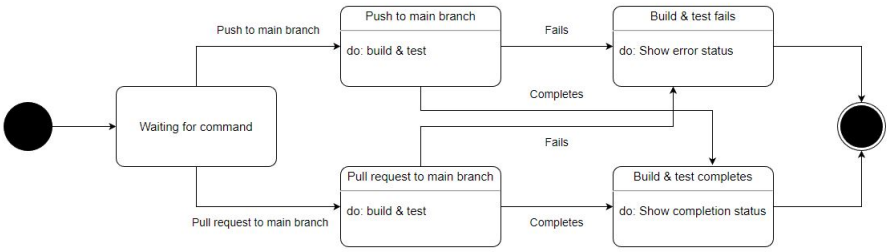
What is the difference between structure diagrams and behavior diagrams in UML? The main difference between these is their static or dynamic approach to the problem. UML structural diagrams depict the static elements of a system and their relations. A typical structural UML diagram is a class diagram. Much like the noun/verb class analysis which could be used in the preliminary rounds of a software engineering process the elements in a structural diagram can be described as nouns and their relations defined by the verbs.
UML behavioral diagrams depict the dynamic behavior of a system. In these it is shown how the system behaves in relation to itself and its own components and also external relations like a user or another system. It is typical to show how data moves through the system and how objects communicate between each other as well as what external behavior changes internal structure of the system.

Exercise 2



Exercise 3

Draw a UML state diagram that models your GitHub action configuration. Include all triggers that you have defined.

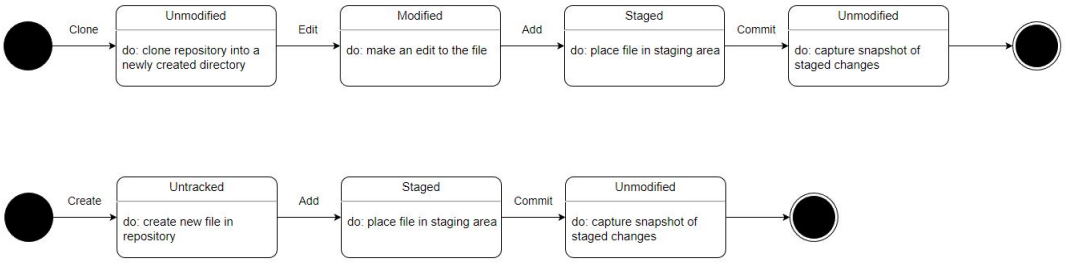


Exercise 4

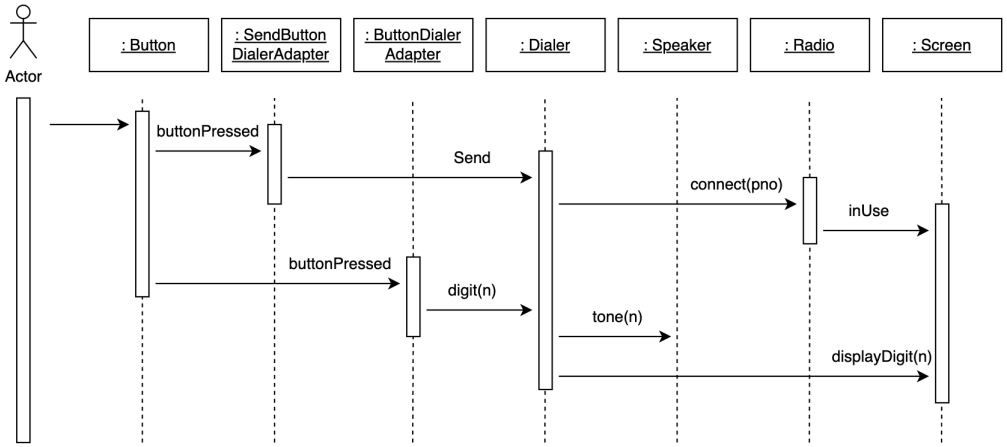
Files that are under version control with Git (or that should be) are in one of the four states: Untracked, Unmodified, Modified, or Staged, see the respective chapter in the Pro Git book. In that book, the authors provide a sequence diagram instead of a state diagram to illustrate states and state changes of a file.

Draw two UML state diagrams that illustrate the states of a single file that is version controlled with Git. Let the first state diagram start with cloning a remote repository containing a file that is then edited. The second state diagram has to illustrate the states of a file that is newly created in a Git repository.

Use the git commands clone, add and commit together with file edits as events that trigger state changes.



Exercise 5



Exercise 6