Hochschule für Technik Stuttgart Leistungsnachweis im **Sommersemester 2013**		Page 1 of 8
Masterstudiengang **S**oftware **T**echnology		Name:
Fach:	Software Engineering 2 (Part 1)	Vorname:
Datum:	15. Juli 2013, 8.30 – 10.30 (Both Parts)	Semester:
Zeit:	120 min (Both Parts)	Matrikel-Nr.:
Prüfer:	Prof. Dr. Deininger	
Hilfsmittel:	One double sided sheet A4 handwritten for Part 1	
Anlagen:	-	

General Note: You may give all answers directly on these pages. If you remove the staples please add your name and matriculation number on **each** page. Overall you can reach 50 points in part 1 and 50 points in part 2.

Examination Question 1 "Configuration Management" (10 Points)

Explain shortly the following terms

Configuration Item Element under configuration control
Version Control Tracking and documentation of changes and configuration items
Repository Management of Configuration Items
Baseline Establishing a defined state in the repository set up on the subsequent activities
Release Product version for customers
Branch Development version

Explain and compare the two concepts “Lock – Modify – Unlock” and “Copy – Modify – Merge”

	Lock – Modify – Unlock	**Copy – Modify – Merge**
Main Ideas	Configuration item is locked while modification, unlocked when writing back	Configuration item is copied (but not locked) while modification,merged when writing back
Advantages / Disadvantages	Secure, too restrictive	May produce conflicts

Examination Question 2 “Advanced Testing” (12 Points)

Explain shortly the following different Test Doubles

Dummy Object SuT expects some object without using it / null or an object without functionality
Test Stub SuT expects some input from an object / provides input, sometimes not provided by regular objects → execution of irregular paths
Test Spy a test stub / additionally records and retrieves interactions executed
Mock Object a test spy + expected calls are configured at the beginning / the mock verifies, that these calls have happened
Fake Object working implementation which is able to substitute the actual object / usually “lighter”, e.g. in-memory-db vs. real db

Given is the following class, which should be used as a “Dummy Object” within your test environment:

```
package services;

public class Service {

     public Result calculate(Object[] objects){
          // TODO Auto-generated code
          return null;
     }
}
```

Explain shortly what extensions / changes you need (by using only standard Java) to make it a Test Stub, Test Spy, or a Mock Object? You may give coding examples or just plain text. About the expected behavior of the test double, you may make up any kind of scenario you want.

Test Stub

Return some real Result-object, maybe in dependency of incoming objects

Test Spy

Additionally a List<Objects → Results> which stores calls/results and can be retrieved

Mock Object

Additionally a List of expected Calls/Results and a function “verifiy” which compares the actual calls/results with the expected

Examination Question 3 "Build Management" (8 Points)

Explain shortly the following terms

Build automation scripting or automating the process of compiling computer source code into binary code ideally a one-step process for turning source code into a working system saves time and to reduces errors
Commanded Automation A user running a script on the command line Examples: Make, Ant, Maven
Continuous Integration Scheduled: a continuous integration server running a nightly build Triggered: a continuous integration server running a build on every commit to a version control system

Within ANT differentiate the terms "target" and "task". Name five examples of predefined tasks.

Target Group of tasks
Task Single ANT-Command
Five predefined tasks javac, junit, war, jar, ftp

Within Maven what is the POM and what is the function?

describes standard models with structure, artifacts, dependencies, project-management-details

defined in the Project Object Model (POM)

Examination Question 4 "Model Driven Software Development" (12 Points)

What are the five core / mandatory elements of MDSD; name and describe shortly

Modeling Language – the language for making the domain model
Metamodel – describes the rules/constraints for the models made by the Modeling Language
Model-Editor (based on the Metamodel) – tool for making the model
Generator/Interpreter – transforms the model into the target platform
Tools for creating Metamodel/Language, Editor, and Transformator

What are internal and external DSLs – explain shortly.

Internal DSL
embedded within the language that implements it / a program written in the host language and using the entire infrastructure of the host
External DSL
designed as an independent language / has its own syntax, semantics, and language infrastructure

Within the Xtext-Exercise you had two roles “Tool Builder” and “Tool User”. Describe shortly, what were the main tasks and outcomes in these two roles.

Tool Builder
Creation of the DSL and defining the Eclipse-Tool
Tool User
User of the before created tool, building elements with DSL

Which of the above roles do you have in AndroMDA – what are the main tasks in this role / these roles?

Only tool-user: creating UML-Diagrams which are used for generating code

Examination Question 5 "Large-Scale Software Systems with OSGi" (8 Points)

What is the difference of programmatic and descriptive OSGi-services?

Programmatic services Originally services had to be created, registered and looked up programmatically / this required a lot of boiler-plate code.
Declarative ~~Descriptive~~ services Declarative Services bypass this through xml-declarations. Additionally needed: Server: XML-(Interface)-Declaration / Client: XML-(Reference)-Declaration

How is Eclipse Equinox related to OSGi?

OSGi is the specification / Equinox is one implementation

A client class within a client bundle wants to access a service, provided by a service bundle. What is the best practice to make this service accessible within OSGi? Draw the expected elements and relationships and give explanations for these elements.