



OST

Ostschweizer
Fachhochschule

Master of Advanced Studies in Software Engineering

MAS-SE

Studienführer

Prof. Dr. Luc Bläser

7. April 2021

Departement Informatik

MAS in Software Engineering

Summary

Zielgruppe	Im Umfeld der Softwareentwicklung tätig
Ziel	Professionelle Anwendung von aktuellem Software Engineering
Umfang	4 Semester berufsbegleitend (Dauer 2,5 Jahre)
Aufbau	3 Semester Zertifikatskurse mit Abendunterricht jeweils am Dienstag und Donnerstag und ca. 10 h Selbststudium pro Woche 1 Semester Masterarbeit
Durchführungsort	OST Ostschweizer Fachhochschule, Campus Rapperswil-Jona
Aufnahmebedingungen	Hochschulabschluss mit nachgewiesener Programmiererfahrung oder "Sur Dossier" mit entsprechender Praxisnähe u. Programmiererfahrung
Kosten	CHF 23'000,-
Start	Nächster Start April 2022 (genauer Termin wird noch bekannt gegeben)
Anmeldung	Bis Mitte Januar (ohne Vorkurse bis Anfang März)
Abschluss	Master of Advanced Studies FHO
Titel	MAS in Software Engineering

Zielgruppe

Personen der Zielgruppe des MAS Software Engineering sind/haben

- Tätigkeit im Umfeld der Softwareentwicklung
- Kein Informatikstudium, sondern eher in die Softwareentwicklung "hineingerutscht"
- einen Hochschulabschluss mit nachgewiesener Programmiererfahrung oder ausgewiesene, mehrjährige Praxiserfahrung mit Programmierkenntnissen, meist mit HF- Abschluss
- Programmiererfahrung: Je mehr desto besser, Unterstützung durch Vorkurse für Interessierte mit wenig Vorkenntnissen (Vorkurse alleine reichen jedoch nicht!)
- das Ziel, das Know-how zu professionalisieren

Zielsetzung

- Der Master of Advanced Studies in Software Engineering kann Softwareprojekte über alle Phasen kompetent bearbeiten
- Das Studium gibt den Teilnehmerinnen und Teilnehmern die modernsten Methoden, Prozesse und Technologien des Software Engineering in die Hand.
- Ziel ist das "Anwendenkönnen".

History

- 1989 Erste Durchführung NDS Software Engineering
- Seit 1989 jährliche Durchführung
- Im Jahr 2000 Umbau zu FH-Nachdiplomstudium und Ausbau auf 3 Semester und Prüfung
- 2006 ... Umsetzung Bolognaform in Schweizer Hochschulen und damit wurde aus FH-NDS ein Master of Advanced Studies (MAS)
- 2007,08,09 Master Upgrade Programm für Nachdiplom-Absolventen
- Seit 2008 jährliche Durchführung als MAS-SE
 - Beruht auf Bewährtem und wird jährlich aktualisiert / optimiert

Bologna Reform an Fachhochschulen

- Fachhochschulausbildung früher:
 - Grundstudium: Diplom FH
 - Weiterbildung: Nachdiplom FH – möglich bis 2009
- (Fach)hochschulausbildung heute gemäss Bologna Reform:
 - Verstärktes Gewicht auf Selbststudium
 - Grundstudium: Bachelor / Master of Sciences
 - Weiterbildung: MAS – Master of Advanced Studies – gilt für alle Hochschulen: ETH, Uni, FH

Anforderungen MAS

- Leistung von 60 ECTS Punkten
- 1 ECTS entspricht ca. 25 – 30 Arbeitsstunden
- Aufwand: 4 Semester mit 18 - 22 Stunden pro Woche

Studienkonzepte

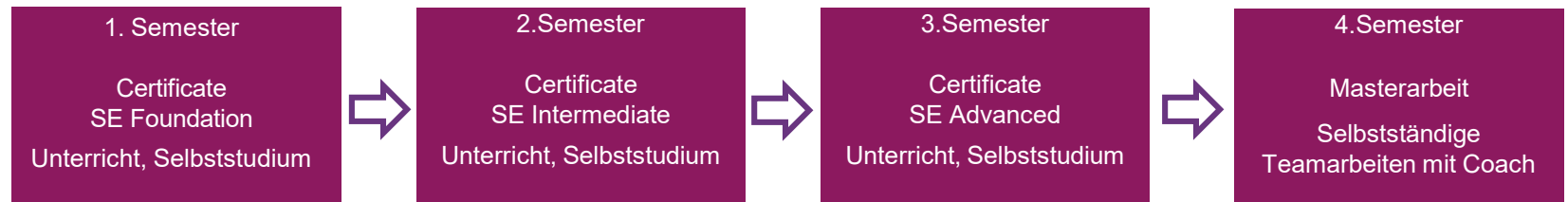
- Iterative Behandlung der Studieninhalte
 - Beispiel OOA ↔ OOD ↔ OOP
- Konzepte sind wichtiger als konkrete Technologien,
 - aber Konzepte kann man nur anhand konkreter Technologien lernen
 - Hands on experience ist wichtig!
- Wenn schon konkrete Technologien, dann aktuelle
- Java und Java-Technologien als Schulsprache
 - Frei verfügbar, viele Open Source Technologien

Aufbau MAS SE

- 1. Semester - Software Engineering Foundation
 - Dauer 20 Wochen
 - 10 Lektionen pro Woche (DI und DO 17:15 bis 21:50) und Selbststudium
 - Kurszertifikat mit 15 ECTS
- 2. Semester - Software Engineering Intermediate
 - Dauer 20 Wochen
 - 10 Lektionen pro Woche (DI und DO 17:15 bis 21:50) und Selbststudium
 - Kurszertifikat mit 15 ECTS
- 3. Semester - Software Engineering Advanced
 - Dauer 20 Wochen
 - 10 Lektionen pro Woche (DI und DO 17:15 bis 21:50) und Selbststudium
 - Kurszertifikat mit 15 ECTS

Aufbau MAS SE

- Semester - Masterarbeit
 - Dauer 20 Wochen
 - Umfang ca. 400 Stunden
 - 15 ECTS
 - Selbständige Arbeit, vorzugsweise im Team
 - Betreut durch Dozierende



Zertifikatskurse 1. – 3. Semester

- Ca. 200 Lektionen Unterricht pro Semester
 - Vorlesungen, Übungen, Gruppenarbeiten
- Ca. 200 Stunden Selbststudium pro Semester
- Aufbau aus drei Modulen
 - gegliedert in mehrere Kursblöcke
- Hoher Übungsanteil (in den meisten Kursblöcken 50%)
- Unterrichtszeiten Dienstag- u. Donnerstagabend je 5 Lektionen
- Abgesetzte Prüfungstermine für mehrere Kursblöcke
 - Pro Modul genügender Durchschnitt erforderlich

Masterarbeit

- Die Studierende bringen Thema ein. Die Bewilligung der Arbeit erfolgt durch einen Antrag bei der Studienleitung.
 - Es sind auch Firmenarbeiten möglich
- Die Studierende wählen einen Coach aus dem Kreis der Dozierenden
- Durchführung in der Regel in Teams
- Grundsätzlich Selbstständige Durchführung mit wenigen Meilensteinen der Studienleitung (z. B. Reviews)
- Schlusspräsentation der Arbeit - für die Öffentlichkeit zugänglich

Inhalte 1. Semester – SE Foundation 1/2

Einführung Objektorientierte Softwareentwicklung

- Grundbegriffe: Klassen, Vererbung, Polymorphismus
- Einführung in UML
- Objektorientierte Analyse (OOA) und Objektorientiertes Design (OOD) mit UML
- Umsetzung des Objektorientierten Designs in Programmiersprache Java

Teambuildingsabend

- Teambildung und Teamdesign
- Teamdiagnose und Teamentwicklung

Grundlagen Betriebssysteme

- Programmausführung und Hardware
- Systemprogrammierung
- Prozesse, Threads und Interprozesskommunikation
- Ein- /Ausgabe

Projektarbeit Objektorientierte Softwareentwicklung

- Objektorientierte Analyse, Objektorientiertes Design und Objektorientierte Programmierung eines Warenautomates in Java
- Arbeit wird in kleinen Teams ausgeführt

Objektorientierte Modellierung

- Vertiefung Objektorientierte Analyse und Design
- Domainmodellierung mit UML
- Entwurf von Klassen
- Modellierung mit UML Zustands-, Use Case- und Sequenzdiagrammen

Programmieren Java

- Datentypen, Operatoren, Anweisungen und Kontrollstrukturen
- Programmstruktur (Blöcke, Methoden, Klassen, innere Klassen, Anwendung von Generics, Packages)
- Schnittstellen, Vererbung und Polymorphismus,
- Exception Handling
- Ein- /Ausgabe und Streams
- Architektur der Java Virtual Machine

Einführung in Requirements Engineering

- Übersicht Requirements Engineering Methoden
- Use Cases für Funktionale Anforderungen
- Qualitätsmodell für Nichtfunktionale Anforderungen
- Requirements Management: Verwalten, Priorisieren, Changemanagement
- Quellen und Erhebungstechniken für Requirements

Inhalte 1. Semester – SE Foundation 2/2

Windows Betriebssysteme

- Einführung: Entstehung, Versionen und Architektur
- Windows Registry und Services
- Zugriffsrechte
- Prozesse, Threads und Scheduling, Interprozesskommunikation
- Windows Graphical User Interface
- Dateisysteme: FAT, NTFS und Festplattenpartitionierung

Software Prozesse

- Software-Vorgehensmodelle
- Wasserfall- und V-Modell
- Iterative Modelle (Unified Process)
- Agile Prozesse (Scrum, XP)
- Kombination von Prozessen („balanced approach“)

Web Engineering 1

- Grundlagen HTML 5
- Grundlagen CSS
- Grundlagen JavaScript/EcmaScript 6
- DOM und jQuery
- Web App ohne Serververbindung

Programmieren in C++

- Datentypen, Operatoren, Anweisungen, Kontrollstrukturen
- Blöcke, Funktionen, Klassen, friend Klassen, Templates, Namespaces
- Vererbung, Virtuelle Funktionen, Mehrfachvererbung
- Dynamische Speicherverwaltung
- Schnittstelle zu C, Runtime-Type-Information RTTI
- Exception Handling, Input- und Output in C++
- ANSI/ISO C++ und Klassenbibliotheken

Unix Betriebssysteme

- Einführung: Entstehung, Derivate, Normierung
- UNIX-Philosophie und Architektur (Kern und GUI-Aufsätze)
- Unix Shell und Shell Scripts, Unix Programmierung
- Parallelverarbeitung und Interprozess-Kommunikation, Prozesszustände, Scheduling
- POSIX Threads, Interprozesskommunikation (IPC): Unix Signale, Unix Pipes

Algorithmen und Datenstrukturen

- Rekursion
- Analyse von Algorithmen: O-Notation, Arithmetische Progression
- Sortierung: Selection-, Insertion-, Merge-, Quick-Sort
- Collections: Vector, ArrayList, Stack, Queue, Linked List, Hashtable, Map und Dictionary, Bäume, Graphen

Inhalte 2. Semester – SE Intermediate 1/2

Programmieren Java advanced

- Generics und Vererbung
- Annotations
- Java-Reflection API
- Lambdas
- Garbage-Collection und Schwache Referenzen (Strong-, Weak-, Soft- und Phantom-Referenzen)
- Java Native Interface (JNI)
- Aspekt-Orientierte Prog. (AOP) mit AspectJ
- Design-by-Contract mit Assertions und JML
- Java-Internationalization, Logging-Framework

Objektorientiertes Design (OOD)

- Aufteilung eines Software Systems nach Domain Driven Design (DDD)
- Zuweisung von Verantwortlichkeiten nach GRASP
- Design Principles und Practices für objektorientierte Software-Systeme
- Grundlegende und fortgeschrittene Design Patterns
- Code Smells / Refactoring
- Test Driven Development (TDD) Workshop

Datenbanken Grundlagen

- Einführung: Aufbau eines DBMS
- Datenmodellierung
- SQL als Datendefinitionssprache
- SQL als Datenmanipulationssprache
- Transaktionen
- Implementierung und Abfragen einer Datenbank in PostgreSQL

Parallele Programmierung

- Multi-Threading mit Java
- Synchronisation: Monitor Konzept, Semaphore, Lock&Condition, RW-Locks, Latches & Barrieren
- Korrektheit und Vermeidung von Data Races, Deadlocks, Starvation
- Thread Pools
- Java Memory Model, lock-freie Programmierung

Software Architektur

- Grundprinzipien guter Applikationsarchitekturen
- Rolle des Architekten
- Architekturtypen: Schichten, Pipes und Filters, Interaktive und Verteilte Systeme
- Sichten auf Architekturen
- Architektur-Patterns

Inhalte 2. Semester – SE Intermediate 2/2

Web Engineering 2

- REST Service
- Ajax
- Single Page Application
- node.js

Kommunikation in verteilten Systemen

- Grundlagen: Basisarchitekturen, Interprozesskommunikation, Service-Architekturen
- Middleware: Synchrone Kommunikation (RMI, REST, Webservices), Asynchrone Kommunikation (JMS)
- Enterprise Service Bus

.Net und C#

- Grundlagen .NET Framework und C#
- Unterschiede C# und Java
- .NET Konzepte wie Delegates, Events und LINQ
- User Interfaces mit WPF und XAML
- Entity Framework und LINQ to SQL

Agile Software Development

- Grundkonzepte, Agile Manifesto
- Agile Prozesse: XP, Crystal Clear, Adaptive Software Development (ASD), Feature Driven Development (FDD), Scrum
- Agile Unified Process
- Vergleich und Einordnung Agiler Prozesse

Project Automation

- Verständnis der Grundprinzipien der Projektautomatisierung
- Sicherstellung der Qualität durch Automatisierung und Continuous Integration
- Einsatz moderner Werkzeuge zur Automatisierung von Build-, Release- und Deploymentprozessen und deren Anwendung im Projektalltag.
- Kennenlernen von Continuous Deployment und Betriebsszenarien auf Basis von Containerization-Technologien wie Docker

Inhalte 3. Semester – SE Advanced 1/2

Requirements Engineering Advanced

- Requirements Engineering Prozess
- Qualitätsmodell für Anwendungssysteme
- Essenzielle objektorientierte Analyse der funktionalen Anforderungen
- Systemdesign der funktionalen und nicht-funktionalen Anforderungen

Datenschutz

- Datenschutz Grundlagen – was gilt es zu schützen?
- Was verlangen Datenschutzgesetze? (Schweiz, EU, international)
- Um welche Daten geht es? (Personendaten, Profile, Tracking, Gesichtserkennung, ...)
- Worauf haben Entwickler von Anwendungen zu achten? (Privacy by Default, Privacy by Design)
- Welche Risiken gehen wir ein im Bereich Datenschutz? (als betroffene Person, als Firma, als Staat)

Qualitätsmanagement

- Übersicht Qualitätsmanagement
- ISO 9000 und ISO 9001
- Qualitätssicherungsmassnahmen
- Reviewtechniken
- Risikomanagement

Architekturen verteilter Systeme

- Herkunft und Basisarchitekturen
- Grundlagen und Vorgehen
- Bekannte Architektur-Strukturen und –Plattformen (SOA)
- Service Architekturen
- Anatomie verteilter Systeme, OMA, SOA, EDA Architekturen
- Design, Implementation und Integration: Modularisierung, Schnittstellen-Design, Design eines Services

Mobile App Engineering

- Einführung in das Software Engineering für Android
- Übersicht über die grundlegenden Challenges bei der Entwicklung für Mobile Plattformen
- Entwickeln einer App für Android

Datenbanken Advanced

- DB-Security
- Stored Procedures and Triggers
- DB-Programmierung (JDBC, OR-Mapper JPA)

Web Engineering 3

- Typescript
- React als Frontend-Framework
- Responsive Web Design

Inhalte 3. Semester – SE Advanced 2/2

Application Security

- Information Security Management, Sicherheitsmassnahmen (präventiv, korrektiv)
- Kryptologie Grundlagen (Verschlüsselung, Signatur, Zertifikate, Authentisierung)
- Schwachstellen (Injections, Fehlkonfigurationen, Schutz sensible Daten)
- Best Practices (Datenbanken, Webanwendungen, Software Development, Authentisierung und Autorisierung)

Internet of Things Grundlagen

- Hardwarenahe Programmierung
- SE Methoden im IoT Umfeld
- Internet Anbindung, Data/Event Hub
- Arduino Entwicklung

Funktionale Programmierung

- Eigenschaften und Anwendung Funktionaler Sprachen
- Funktionale Programmiersprache Haskell:
- Typen und Klassen, Generatoren
- Rekursive Funktionen, Funktionen höherer Ordnung
- Interaktive Programme
- Input/Output, Monaden

Software Testing

- Checking und Testing
- Exploratory Testing und Session Based Test Management
- Quality Models und Test Strategie
- Embedded Testing in agilen Projekten
- Domain Testing und Test Case Design
- Unit Testing, Mocking und TDD
- Regressionstests und Systemtest Automatisierung

Cloud Computing

- Grundlegende Eigenschaften: Self Service, Pay-per-Use, Elastizität, Resource Pooling, CAP-Theorem
- Cloud Deployment Types (z.B. Private Cloud, Public Cloud)
- Service Models: SaaS, PaaS, IaaS
- Service Level Agreements (Security, Billing) und rechtliche Rahmenbedingungen
- Cloud Architecture Patterns für Processing, Storage, Communication und Application Architecture
- Technische Umsetzung der Patterns in aktuellen Cloud Offerings

Interaction Design

- User-Centered Design
- Goal-Driven Design
- Personas, Szenarien
- User Interface Prototypen
- Usability Tests

Unterlagen und Infrastruktur

- Notwendige Bücher werden abgegeben.
 - Die Kosten sind in den Studiengebühren inbegriffen
- Sie bringen Ihr eigenes Notebook für die Übungen mit.
 - Das Betriebssystem ist beliebig wählbar.
 - Ein Player für Virtual Images ist erforderlich (z.T. werden Virtual Machines mit vorbereiteter Übungsumgebung abgegeben).
- OST- Login
 - Zugriff auf OST Infrastruktur (auf Campus WLAN, VPN)
 - Kursplattform Moodle als "zusätzliches virtuelles Klassenzimmer"

Dozierende

- Ca. 31 Dozierende
 - Ca. 7 Professor*innen von Fachhochschulen
 - Ca. 25 aus Wirtschaft
 - Langjährige Erfahrung
 - Hoher Praxisbezug
- Für Übungen am Computer zusätzliche Übungsbetreuer

Wie können Sie sich weiter informieren?

- Informationsanlässe:

An den Informationsveranstaltungen erhalten Interessentinnen und Interessenten eine Übersicht des Weiterbildungsstudiums, haben die Möglichkeit die Studienleitung kennen zu lernen und Fragen zum Programm zu stellen. Die Termine der Infoanlässe erhalten Sie unter: www.ost.ch/mas-se

Ansprechpartner MAS Software Engineering

- Allgemeine Auskünfte zu MAS-SE und Anmeldung:

Marion Hug
Studienkoordination MAS-SE
Telefon +41 (0)58 257 46 32
E-Mail marion.hug@ost.ch

- Fachliche Fragen zum MAS Software Engineering

Prof. Dr. Luc Bläser
Studienleiter MAS-SE
Telefon +41 (0)58 257 46 22
E-Mail: luc.blaeser@ost.ch

- Studienberatung Weiterbildung ICT-Bereich

Peter Nedic
Co-Leiter Weiterbildung, Campus Rapperswil
Telefon +41 (0)58 257 49 21
E-Mail: peter.nedic@ost.ch