

Master of Advanced Studies in

# SOFTWARE ENGINEERING

MAS-SE

## STUDIENFÜHRER

Prof. Dr. Luc Bläser

Studienleiter MAS Software Engineering

Ausgabe vom 07.08.2019



# 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<br>▶ Abendunterricht Dienstag 5 Lektionen, Donnerstag 5 Lektionen und Selbststudium ca. 10h/Woche<br>1 Semester Masterarbeit |
| Durchführungsort    | HSR Hochschule für Technik Rapperswil  |
| Aufnahmebedingungen | Hochschulabschluss oder entsprechende Praxisnachweise<br>Programmiererfahrung  |
| Kosten              | CHF 23 000.-   |
| Anmeldung           | Mit Anmeldeossier bis zum 30. Januar 2020 (ohne Vorkurse bis 28. Februar 2020) Berücksichtigung in der Reihenfolge des Eingangs                          |
| Start               | Nächster Start 21. April 2020  |
| Abschluss           | Master of Advanced Studies der Fachhochschule Ostschweiz   |
| Titel               | MAS FHO in Software Engineering  |
| Warum an der HSR?   | ▶ Langjährige Erfahrung mit Software Engineering Weiterbildung<br>▶ Dozenten mit praxisbezogenem Know-how<br>▶ Gute Akzeptanz in Wirtschaft              |

# Zielgruppe und Zielsetzung

## Personen der Zielgruppe des MAS Software Engineering sind/haben

- im Umfeld der Softwareentwicklung tätig
- meist nicht Informatik studiert, sondern ist in Softwareentwicklung „hineingerutscht“
- einen Hochschulabschluss oder ausgewiesene, mehrjährige Praxiserfahrung, meist mit HF-Abschluss.
- Programmiererfahrung: Je mehr desto besser, Unterstützung durch Vorkurse für Interessierte mit wenig Vorkenntnissen (Vorkurse alleine reichen 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**
  - seither jedes Jahr durchgeführt!
  
- **2000 Umbau zu FH-Nachdiplomstudium**
  - Ausbau auf 3 Semester, Prüfungen
  
- **2006 ... Umsetzung Bolognaform in CH-Hochschulen**
  - FH-NDS → Master of Advanced Studies (MAS)
  
- **2007,08,09 Master Upgrade Programm für Nachdiplom-Absolventen**
  
- **Seit 2008 jährliche Durchführung als MAS-SE**
  - Beruht auf Bewährtem
  - Wird jährlich aktualisiert
  - 2019 Start mit 29 Teilnehmenden

# Bologna Reform an Fachhochschulen (FHs)

- **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*
    - gleich für alle Hochschulen: ETH, Unis, FHs

# Anforderungen MAS

- **Leistung von 60 ECTS-Punkten**
- **1 ECTS = 25-30 Arbeitsstunden für Studierende**
- **Aufwand:**
  - Mind. 4 Semester mit 18-22 Std./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

## **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

**Alle drei Semester beinhalten zu einem wesentlichen Teil praktische Übungen**



## 4. Semester: Master-Arbeit

- Dauer 20 Wochen
- Umfang ca. 400 Stunden
- 15 ECTS
- Selbständige Arbeit, vorzugsweise in Team
- Betreut durch Dozent



# 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 und Donnerstag**  
**je 5 Lektionen: 17:15 –21:50 Uhr**
- **Abgesetzte Prüfungstermine für mehrere Kursblöcke**
  - Pro Modul genügender Durchschnitt erforderlich

- **Die Studierende bringen Thema ein. Die Bewilligung der Arbeit erfolgt durch einen Antrag bei der Studienleitung.**
  - Auch Firmenarbeiten möglich
- **Die Studierende wählen einen Coach aus dem Kreis der Dozenten**
- **Durchführung:**
  - In der Regel in Teams
    - höherer Lerneffekt, macht mehr Spass
  - Ausnahmsweise auch Einzelarbeiten
- **Selbstständige Durchführung**
  - Wenige Meilensteine von Studienleitung vorgegeben
- **Schlusspräsentation, auch für nächste Generation der Studierenden**

# 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

## Arbeiten in Teams

- Selbstmanagement
- Kommunikation und Kommunikationsmodelle
- 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 Modellierungspraxis

- Grundlagen der Domainmodellierung
- UML für Domainmodellierung
- Modellierungstraining an mehreren Fallstudien

## 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, Changelogmanagement
- 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“)

## Projektmanagement

- Projektmanagement und -organisation
- Projektziele, Akteure und Rollen
- Phasen- und Entwicklungsmodelle, Scrum
- Dokumentation, Fortschrittskontrolle, Berichtswesen, Projektabschluss, Debriefing
- Change und Software-Konfigurationsmanagement

## 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)

- Grundprinzipien Objektorientiertes Design
- UML als Designnotation
- Responsibility Driven Design
- Design-Verifikation
- Design Pattern
- Control Style und Software Architektur

## Software Architektur

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

## Datenbanken Grundlagen

- Grundlagen Datenbanken
- Datenmodellierung, Relationales Datenmodell, Normalisierung
- SQL als DDL und DML

## Funktionale Programmierung

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

## Parallele Programmierung

- Multi-Threading mit Java, Java Memory Modell
- Synchronisation in Java (Kritische Abschnitte, Monitor Prinzip)
- Zustandssynchronisation u. gegenseitiger Ausschluss
- Semaphoren und Locks (Condition Variable)
- Deadlock-Problematik, Klassische Probleme der Synchronisation

## Project Automation

- Build Automation: Einführung, Ant
- Scheduled Builds mit Cruise Control
- Testing, Packaging und Deployment mit Ant
- Release Automation mit Ant und Maven

# Inhalte 2. Semester: SE intermediate 2/2

## Grundlagen Internettechnologien

- Architektur von Internetapplikationen, Übersicht Internettechnologien
- Java-Programmierung auf dem Webserver: Servlets, JSF
- Programmierung auf dem Web-Client: JavaScript, JQuery, AJAX

## Kommunikation in verteilten Systemen

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

## C# und .net

- Grundlagen des .NET Frameworks
- Grundkonzepte der Programmiersprache C#
- Datenbankzugriff mit ADO.NET
- Web-Applikationen mit ASP.NET

## Human Computer Interaction & GUIs in Java

- Grundlagen Userinterfaces in Java: Model View Controller (MVC) Architektur, Composite Pattern in GUIs, Ereignisverarbeitung
- Übersicht der Java Bibliotheken für Graphical User Interfaces: awt, Swing, SWT
- Usage Centered Design (Constantine u.a.)
- Techniken des HCI-Designs
- Experimentelle Userinterfaces

## 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

# 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

## Software Engineering für Mobile Plattformen

- 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)

## Rechtliche Aspekte

- Überblick über das Immaterialgüterrecht
- Software: Schutz und Überlassung (Übertragung und Lizenzierung)
- Domain-Streitigkeiten
- E-Commerce: Rechtliche, insbesondere Datenschutz und
- persönlichkeitsrechtliche Aspekte im Zusammenhang mit einer Internetplattform

## 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

## Qualitätsmanagement

- Grundlagen der Software-Prüfung
- Messbare Merkmale von Software
- Reviews
- Qualitätsmanagementsysteme (ISO 9001 und 90003)
- Qualitätssicherung
- Prozessbewertung und -verbesserung (CMMI)
- Prozessmetriken

## Performante Teams, Process Communication Model

- Moderationsformen von Teams
- Konstruktive Kommunikation als Schlüssel zur Teamarbeit
- Gruppenregeln zur Effizienzsteigerung in Gruppenarbeiten
- Kommunikationsmodell der Process Communication: grundlegende Persönlichkeitsprofile, Dynamik der zwischenmenschlichen Kommunikation



# Inhalte 3. Semester: SE advanced 2/2

## 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

## Informationssicherheit

- Information Security Management (Werte, Bedrohung, Massnahmen, Verletzlichkeit)
- Massnahmen zur Verbesserung der Sicherheit (Organisation, Technik, Personal)
- Kryptologie Grundlagen (Verschlüsselung, Signatur, Zertifikate, Authentisierung)
- Anwendungssicherheit (OWASP)

## Internet of Things Grundlagen

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

## Web Engineering Advanced

- Einführung in die Server-Programmierung
- Weiterführende Themen in der Server-Programmierung
- Single Page Applikation (SPA)

## 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

- **Notwendige Bücher werden abgegeben.**  
Die Kosten sind in den Studiengebühren inbegriffen.
- **Die Unterlagen werden für jeden Unterrichtsabend in Papierform bereitgestellt.**
- **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).
- **HSR Login**
  - Zugriff auf HSR-Infrastruktur (auf Campus WLAN, VPN)
  - Kursplattform Moodle für Ablage der Unterlagen

# Dozierende

- **Ca. 29 Dozierende**
  - Ca. 9 Professoren von Fachhochschulen
  - Ca. 20 aus Wirtschaft
- **→ Langjährige Erfahrung**
- **→ Hoher Praxisbezug**
- **Für Übungen am Computer zusätzliche Übungsbetreuer**

## Dozentenliste – Momentaufnahme 1/2

|   |  |
|---|--|
| Baeriswyl Bruno, Dr. iur.                                     | Datenschutzbeauftragter des Kantons Zürich |
| Bauer Manuel, Dipl. Inf. Ing. FH / Executive MBA              | Devinit GmbH                               |
| Bianchi Bernhard, Dipl. Ing.                                  | Bianchi & Partner GmbH                     |
| Bläser Luc, Prof. Dr.   | HSR Hochschule für Technik Rapperswil      |
| Briner Thomas, Dipl. Inf. Ing. ETH                            | Ergon Informatik AG                        |
| Bruderer Rolf, Dipl. Inf. Ing. ETH                            | Zühlke Engineering AG                      |
| Corbat Thomas, M. Sc. In Engineeringn FHO                     | HSR Hochschule für Technik Rapperswil      |
| Demarmels Mischa, M. Sc. HCI                                  | Zühlke Engineering AG                      |
| Frühauf Karol, Dipl. Ing.                                     | INFOGEM AG                                 |
| Gehrig Silvan, B. Sc. Informatik FH                           | HSR Hochschule für Technik Rapperswil      |
| Gfeller Michael, B. Sc. Informatik FH                         | HSR Hochschule für Technik Rapperswil      |
| Glatz Eduard, Prof. Dr.                                       | HSR Hochschule für Technik Rapperswil      |
| Hänggi Esther, Dr. sc. ETH                                    | Ergon Informatik AG                        |
| Hauri Christian, lic. phil. Psychologe                        | Hauri Ergonomie & Coaching                 |
| Höltzsch Thomas, Dipl. El. Ing. HTL, NDS Software Engineering | Samotech Engineering GmbH                  |
| Huser Hansjörg, Prof.   | HSR Hochschule für Technik Rapperswil      |
| Koch Andres, Dipl. Ing., M. Math                              | Object Engineering GmbH                    |



## Dozentenliste – Momentaufnahme 2/2

|  |  |
|--|--|
| Kolb Robert, Dipl. Ing.                        | Noser Young Professionals AG           |
| Konopka Ute-Christine, Dr.                     | thinksafe                              |
| Letsch Thomas, Dipl. Ing.                      | Letsch Informatik                      |
| Mattmann Rudolf, Dr. sc. techn. ETH            | Mettler-Toledo AG                      |
| Mehta Farhad D., Prof. Dr.                     | HSR Hochschule für Technik             |
| Memmel Thomas, Dr., M.Sc. Comp. Science        | Zühlke Engineering AG                  |
| Merkli Jean-Daniel, Dipl. Inf. Ing. ETH        | Zühlke Engineering AG                  |
| Müller Stefan, MSc EEIT ETH                    | Digitec Galaxus AG                     |
| Qvortrup Michael, Dipl. Ing. ETH               | Zühlke Engineering AG                  |
| Ruggli Sandro, lic. iur., Rechtsanwalt, LL. M. | GRP GLOOR RUGGLI PARTNER               |
| Schöb Stefan, B.Sc. FH                         | Zühlke Engineering AG                  |
| Steimle Toni, M. Sc. Uni ZH, UX Director       | Crealogix AG                           |
| Stocker Mirko, M. Sc. In Engineering FHO       | HSR Hochschule für Technnik Rapperswil |
| Stolze Markus, Prof. Dr.                       | HSR Hochschule für Technik Rapperswil  |
| Tobler Daniel, Dipl. Ing.                      | CSS Versicherung                       |
| Zimmermann Martin, Prof. Dr.                   | Hochschule Luzern                      |
| Zimmermann Olaf, Prof. Dr.                     | HSR Hochschule für Technik Rapperswil  |

# 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.

## **Nächste Infoabende:**

Datum: 10. Oktober 2019

Dauer: 18:15 – 19:15 Uhr

Anmeldung unter: [marion.hug@hsr.ch](mailto:marion.hug@hsr.ch)

Die Informationsveranstaltung wird durchgeführt wenn Anmeldungen vorliegen.

Daten und Anmeldung unter <http://www.hsr.ch/mas-se>

# Wie können Sie sich weiter informieren?

## **Allgemeine Auskünfte und Anmeldung:**

HSR Hochschule für Technik Rapperswil  
Weiterbildung  
Marion Hug  
Oberseestrasse 10  
CH-8640 Rapperswil

E-Mail [marion.hug@hsr.ch](mailto:marion.hug@hsr.ch)  
Telefon +41 (0)55 222 49 22

Aktuelle Daten und Kosten entnehmen Sie bitte dem Anmeldedossier:  
Download der Broschüre und des Anmeldedossiers MAS Software Engineering unter:  
<http://www.hsr.ch/mas-se>

# Wie können Sie sich weiter informieren?

## **Persönliches Beratungsgespräch:**

### **Fachliche Fragen MAS und Ihre berufliche Entwicklung**

Prof. Dr. Luc Bläser  
Studienleiter MAS-SE

E-Mail [luc.blaeser@hsr.ch](mailto:luc.blaeser@hsr.ch)  
Telefon +41 (0)55 222 46 22

### **Organisatorische, administrative Fragen MAS und Ihre berufliche Entwicklung**

Peter Nedic  
Leiter Weiterbildung

E-Mail [peter.nedic@hsr.ch](mailto:peter.nedic@hsr.ch)  
Telefon +41 (0)55 222 49 21