



# Informatik an der OST

## Im Fokus: Software Engineering an der OST

Bulletin

2/2022

Hervorragende Ausbildung, die unter Arbeitgebenden in der Schweiz als eine der besten gilt

**Prof. Stefan Richter**  
Studiengangleiter  
Informatik



Software Engineering ist neben Network Engineering, Cybersecurity und Data Engineering eine der wichtigsten Disziplinen der Informatik. Tatsächlich suchen sehr viele Stellenanzeigen in erster Linie nach Software Engineers und weniger nach Informatikerinnen oder Informatikern. Während die Informatik ursprünglich die Wissenschaft der automatisierten Informationsverarbeitung ist, also stark vereinfacht die Wissenschaft effizienter Algorithmen und Datenstrukturen, geht es beim Software Engineering um die konkrete Implementierung komplexer Softwaresysteme. Das ist in etwa vergleichbar mit dem Unterschied zwischen Physik und Maschinenbau.

An der OST vermitteln wir Studierenden sowohl einen wissenschaftlichen Hintergrund in Informatik, Mathematik, Physik und weiteren Themen als auch praktische Kenntnisse moderner Technologien und Methoden des Software Engineerings. Am Institut für Software IFS und am Institut für Netzwerke und Sicherheit forschen unsere Professorinnen und Professoren gemeinsam mit ihren Mitarbeitenden an zahlreichen Themen des Software Engineerings und an anderen angewandten Disziplinen der Informatik. In Kooperation mit Industriepartnern entwickeln wir spannende Prototypen, die konkrete Probleme der echten Welt lösen. Wir wünschen Ihnen viel Spass bei der Lektüre dieses Newsletters.

An der OST gibt es im Studiengang Informatik seit vielen Jahren zwei Schwerpunkte im Bereich Software Engineering: Im Schwerpunkt Software Engineering werden die Studierenden zu Allroundern in der Softwareentwicklung ausgebildet, die später meist in den Forschungs- und Entwicklungsabteilungen von Softwareunternehmen arbeiten. Sie erhalten vertiefte Kenntnisse von Betriebssystemen, verteilten Systemen sowie paralleler Programmierung. Diese sind sowohl für Backends als auch für Frontends wichtig. Zusätzlich zu den Sprachen Java und Python, die alle Informatik-Studierenden an der OST erlernen, beschäftigen sich die Studierenden mit diesem Schwerpunkt mit den industriell bedeutenden Programmiersprachen C# und C++. Sie erhalten ausserdem eine weitergehende Ausbildung in der Softwarearchitektur, damit sie auf eine Rolle als technische Führungskräfte vorbereitet sind.

Der auf die Benutzenden ausgerichtete Schwerpunkt Frontend Engi-

neering erlaubt es den Studierenden, sich auf eine Karriere im Umfeld der User Experience vorzubereiten. So gibt es zunehmend Abteilungen und Unternehmen, die sich ganz auf die Erstellung perfekter, leicht zu bedienender Oberflächen spezialisieren. Dass eine gute Oberfläche entscheidend ist, weiss vermutlich jeder, der schon einmal eine Software benutzen musste, die keine benutzerfreundliche Oberfläche hatte. In diesem Schwerpunkt werden die Studierenden zum einen mit den Technologien für Oberflächen vertraut gemacht, insbesondere Mobile-, Web- und Cloudtechnologien. Zum anderen lernen sie, wie die Interaktion zwischen Mensch und Maschine optimal gestaltet werden kann. Ausserdem erhalten sie vertiefte Kenntnisse in Technologien künstlicher Intelligenz, die gerade in Oberflächen eine immer wichtigere Rolle spielt.

Alle Schwerpunkte im Informatikstudium der OST können sowohl in Teilzeit mit drei Tagen Präsenzunterricht als auch in Vollzeit studiert



werden. Die Regelstudienzeit beträgt 8 bzw. 6 Semester, kann aber bei Bedarf bis auf 14 Semester ausgedehnt werden. Da der Unterricht vollständig modular – ohne feste Klassenzuteilung – erfolgt, kann das persönliche Pensum in jedem Semester neu bestimmt werden. Ein Wechsel zwischen Vollzeit und Teilzeit oder andersherum ist problemlos machbar. Es ist ausserdem möglich, ohne Schwerpunkt zu studieren und sich für die Module einzuschreiben, die am interessantesten schei-

nen. So kann sich eine Studentin beispielsweise zum Teil in Frontend Engineering und zum Teil in Cybersecurity vertiefen und damit ihr Karriereziel als Security Expertin im Webumfeld verfolgen. Oder ein Student vertieft sich in Data Engineering & Machine Intelligence und in Software Engineering, um in der Entwicklung von AI-Software zu arbeiten.

Für welchen Schwerpunkt sich Informatikstudierende an der OST auch immer entscheiden: Alle erhalten

eine hervorragende Ausbildung, die unter Arbeitgebenden in der Schweiz als eine der besten gilt. Neben den herausragenden Dozierenden, topaktuellen Lehrinhalten und der einmaligen Lage direkt am Obersee ist dies einer der Hauptgründe für die höchsten Zufriedenheitswerte unter unseren Absolventinnen und Absolventen.

Weitere Informationen zum Studium finden Sie unter: [ost.ch/i](https://ost.ch/i). (RIST)

## OST tool CodePanorama presented at the ICPC 2022 conference

OST student Marc Etter presented the CodePanorama tool at the

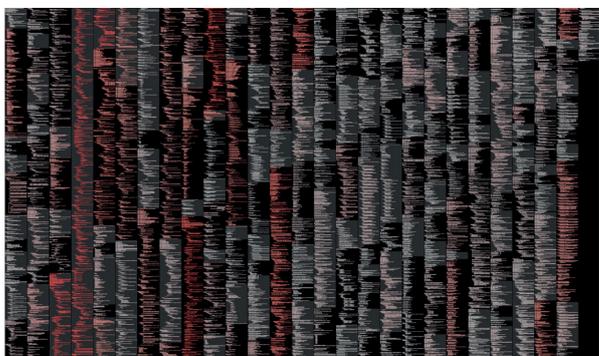
International Conference on Program Comprehension (ICPC) 2022 conference.

The CodePanorama tool, developed at the OST under the supervision of Prof. Dr. Farhad Mehta, allows developers to efficiently spot anomalies in the source code of large computer programs by generating zoomed-out images of the codebase with a selection of useful colour overlays.

The presentation and peer-reviewed conference paper «CodePanorama: a language agnostic tool for visual code inspection» is also available free of charge online at the following link <https://conf.researchr.org/details/icpc-2022/icpc-2022-tool-demonstration/1/CodePanorama-a-language-agnostic-tool-for-visual-code-inspection>

(MEFA)

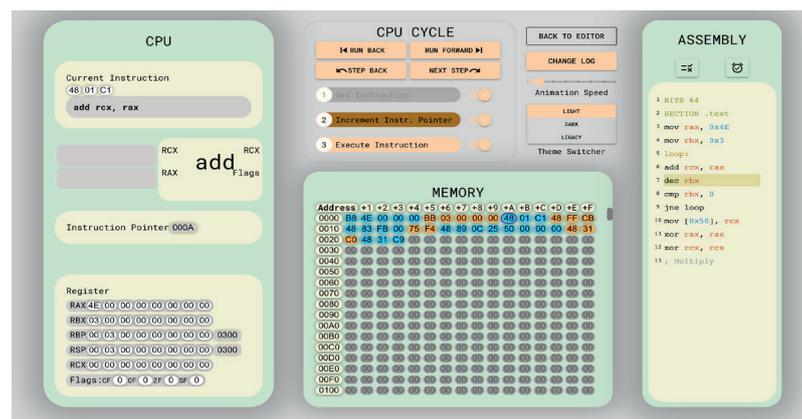
The tool is free to use online at <https://codepanorama.io>



## Grafischer Prozessor-Simulator

Um den Informatikstudierenden die Funktionsweise des Prozessors näherzubringen, entwickelten Yves Boillat und Eliane Schmidli in ihrer Studien- und Bachelorarbeit einen Prozessorsimulator, der die Ausführung von Assembly Code durch den Prozessor schrittweise anzeigt. Dafür musste ein stark abstrahiertes und dem Lehrinhalt des Moduls Betriebssysteme 1 entsprechendes Design entwickelt werden. Die Benutzbarkeit und Verständlichkeit der Applikation wurden mithilfe von Usability Tests mit den betreffenden Studierenden sichergestellt.

Michael Schneider und Tobias Peter entwickelten den Simulator in ihrer Studien- und Bachelorarbeit weiter. Unter anderem ergänzten sie ihn um die Möglichkeit, einzelne Instruktionen in einem Programm zu inspizieren, ohne den gesamten Simulator Schritt für Schritt durchlaufen zu müssen. Nun können die Studierenden die Ausführung einer Inst-



ruktion wiederholen, oder mit Hilfe von bedingten Break- und Watchpoints an eine bestimmte Stelle springen. Durch die Bedingungen wird es beispielsweise möglich, den Simulator bei der x-ten Ausführung eines Loops anzuhalten. Die zwei Studenten führten die Applikation durch diese Änderungen immer näher an eine «IDE für Assembly» heran, ohne dabei den Fokus für den pädagogischen Charakter zu verlieren.

Durch die Arbeit der vier Studierenden entstand ein Produkt, das Zustandsänderungen ansprechend und nachvollziehbar anzeigt und sich somit von anderen Simulatoren abhebt. Ab diesem Herbst wird der Simulator im Modul Betriebssysteme 1 eingesetzt.

(SELI)

Simulator: <http://cpusimulator.pages.gitlab.ost.ch/cpusim-frontend/>

## Buchprojekt: Patterns for API Design

Software ist aus der heutigen Welt nicht mehr wegzudenken: Als Innovationstreiber bei der Entwicklung neuer Produkte und Dienstleistungen aller Couleur. Aber auch, um den Kontakt zwischen Freunden und Familien während eines Lockdowns aufrechtzuerhalten oder die Arbeit verteilter Teams im Homeoffice zu ermöglichen. Moderne Softwaresysteme bestehen aus vielen Teilen, die sich an unterschiedlichen Orten befinden: als Webapplikationen im Browser, als Apps für unterschiedliche mobile Geräte, in einer Cloud und in Firmenrechenzentren. Als Schnittstelle für die Kommunikation und den Datenaustausch zwischen den Systemen dienen Application Programming Interfaces (APIs). Sie sind gewissermassen die Noppen auf den Anwendungs-Legosteinen. Dabei sind APIs häufig nicht nur interner Bestandteil eines Systems, sondern werden als Software-as-a-Service-Angebote auch Dritten zur Verfügung gestellt: So bieten Cloud-Provider wie Amazon und Google via API Zugang zu ihren Produkten an. Dasselbe trifft auf das freie Geodatenprojekt OpenStreetMap zu.

Die meisten APIs sind keine fixfertigen Softwarebibliotheken, sondern

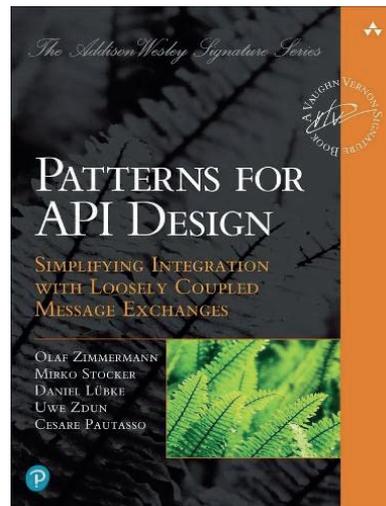
werden in Projekten neu entwickelt und auf individuelle Anforderungen ihrer Nutzer zugeschnitten. Dabei ist es wichtig, das Rad nicht in jedem Projekt neu zu erfinden, sondern von der kollektiven Erfahrung der API-Community zu profitieren. Eine beliebte Form, solches Communitywissen zu strukturieren, sind Design-Patterns, die bewährte Lösungen mit Vor- und Nachteilen auf einer konzeptionellen Ebene technologieübergreifend beschreiben. In der Literatur finden sich bereits Ausführungen zu Patterns, die den API-Entwurf be-

schleunigen und risikoärmer machen können. Der Datenaustausch zwischen API-Nutzerprogramm und API-Implementierung wurde aber noch nicht umfassend betrachtet.

### Unter der Leitung von OST-Informatikprofessoren

Dieser Herausforderung stellte sich ein Autorenteam aus der Industrie und von den Universitäten Wien und Lugano unter der Leitung der OST-Informatikprofessoren Olaf Zimmermann und Mirko Stocker. Ihr für Dezember 2022 im Addison-Wesley Professional Verlag angekündigtes Buch «Patterns for API Design: Simplifying Integration with Loosely Coupled Message Exchanges» stellt 44 Patterns zu unterschiedlichen API-Designaspekten vor, die in verschiedenen Entwicklungsphasen einer API nützlich sind. Einige Beispiele sind API Key (Zugriffskontrolle und -schutz), Pagination (Blättern in Ergebnismengen) und Two in Production (duale Bereitstellung von Versionen).

Die Website [www.api-patterns.org](http://www.api-patterns.org) bietet bereits einen Überblick über die Patterns und deren Anwendung. (STM7/ZIOL)



## IDE-Plugin zur Prüfung von Clean-Code-Regeln

Guter Code sollte so geschrieben sein, dass er auch für Programmierinnen und Programmierer, die noch nicht mit ihm vertraut sind, verständlich ist. Dies ermöglicht es, dass Code in Teams effizient weiterentwickelt werden kann. Regeln und Standards unterstützen Coderinnen und Coder, dieses Ziel zu erreichen. Diese Regeln unterstützen zum Beispiel bei der Auswahl geeigneter Bezeichner für Variablen und Funktionen und werden oft als Clean Code bezeichnet. Bezeichner sollten aus

ganzen und aussprechbaren Wörtern bestehen, die den Zweck einer Variablen, Klasse oder Funktion beschreiben.

Verstöße gegen Clean Code, zum Beispiel schlecht gewählte Bezeichner, werden häufig erst im Nachhinein in Code Reviews korrigiert. Ein Mechanismus, der direkt bei der Programmierung auf die Verletzung dieser Regeln hinweist, wäre für die Programmierausbildung und die Praxis wünschenswert. Dadurch würden Probleme gar nicht erst ent-

stehen. In einer Bachelorarbeit entwickelte ein Student ein Plugin für Visual Studio Code und implementierte dieses. Es weist schon während der Programmierung darauf hin, dass ein bestimmter Bezeichner kein vollständiges Wort ist. So wird die Entwicklerin oder der Entwickler bei der Programmierung motiviert, einen Bezeichner wie «idx» durch das vollständige und aussprechbare Wort «index» zu ersetzen.

Das Plugin wurde mit Nutzerinnen und Nutzern mit unterschiedlicher Programmiererfahrung getestet. Es zeigte sich ein Mehrwert für alle Erfahrungsstufen, sowohl für Personen in der Ausbildung als auch für solche in der Berufspraxis. Das Plugin steht unter freier Lizenz im Visual Studio Code Marketplace zur Verfügung. Der Quellcode kann auf Github eingesehen und bearbeitet werden.

Abbildung:  
Darstellung von  
Regelverstößen  
in der Entwick-  
lungsumgebung

```
C: > Users > friederloch > tester.java
1 public class calculate {
2
3 }
4
```

calculate is not a noun  
class names should start with a noun. Learn more at  
<https://gitlab.com/cleancode2/cleancode-plugin/-/blob/master/docs/wiki/english/namingrules.md>  
View Problem No quick fixes available



## Faszination Wissen: Es darf etwas mehr sein!

Die Herausforderung, aufgrund von komplexen Sachverhalten fundierte Entscheidungen zu treffen sowie komplexe Systeme zu verstehen und zu optimieren, fasziniert mich schon länger.



**Thomas Kehl**  
Softwareentwickler

Nachdem ich vor einigen Jahren an der damaligen HSR meinen Master mit Bestnoten erlangte, möchte ich nun meine Arbeit auf diesem Feld fortführen und habe mich für eine Promotion entschieden. Diese werde ich über die nächsten Jahre in Zusammenarbeit mit der UniBw in München sowie mit meinem Doktorvater Andreas Rinkel, der mich bereits durch meinen Master begleitete, absolvieren.

Zu Beginn geht es darum, das Forschungsthema festzulegen. Nachdem ich in meiner Masterarbeit den Fokus auf die Konzeption, Implementierung und Konfiguration von agentenbasierten Simulationsmo-

dellen legte, könnte ich in meiner Doktorarbeit dieses Feld in Richtung künstliche Intelligenz erweitern. Festzulegen, wie genau künstliche Intelligenz genutzt werden kann, wo die heutigen Problemfelder liegen und was der Stand der Forschung ist, wird der Beginn meiner wissenschaftlichen Arbeit sein. Aus den gewonnenen Erkenntnissen möchte ich anschließend neue Konzepte entwickeln und diese mit einer Software den Menschen einfach zugänglich machen – es darf halt doch etwas mehr sein.

Ich freue mich sehr auf diese spannende Herausforderung!

(KETH)

## Frontend Best Practices 2022: Herbstevent zum Thema Accessibility

Am Dienstag, 11. September 2022 organisierte Prof. Dr. Markus Stolze den neunten Event Frontend Best Practices, der seit 2021 zum fünften Mal in Folge online stattfand.

Nach den Events mit den Themen Design Systeme – Infrastruktur, Vue App Entwicklung als Teamsport, Lightweight Design Systems und Code Reviews – Effective & Sustainable war das diesjährige Thema Accessibility. Umrahmt von einer Einführung und einem Abschluss von Markus Stolze beleuchteten drei eingeladene Referenten das Thema von unterschiedlichen Seiten. Miro Dietiker (MD Systems GmbH), erklärte warum Agenturen das von ihm mitentwickelte Accessibility Manifest unterschreiben sollten. So können sie als Consultant, Agentur oder Softwareanbieter öffentlich sichtbar werden und Accessibility nach bestem Wissen und Gewissen in Produkten und Dienstleistungen umsetzen. Josua Muheim zeigte, wie HTML Form Controls genutzt werden können, um komplexe Controls wie Multi Select Dropdown Controls zu implementieren und warum dies vorteilhaft ist.

Den letzten Vortrag hielt Stefan Barac (mimacom). Er zeigte beispielhaft, dass Accessibility nicht nur eine Sache der richtigen Verwendung von HTML, CSS und JavaScript ist, sondern dass Kontext, konkrete Hardware-Gestaltung und die Mon-



tage von Apparaten im öffentlichen Raum ebenfalls einen signifikanten Einfluss darauf haben, ob diese zugänglich sind oder nicht.

Es war erfreulich, dass sich mehr als 30 Personen, vielfach Alumni des Bachelors Informatik, des CAS Frontend Engineering oder des MAS Human-Computer Interaction Design in den Lifestream einwählten und einige den Mut fassten, zur Diskussion beizutragen.

Der nächste Event Frontend Best Practices findet am Mittwoch, 1. März 2023 statt. Die Aula ist schon reserviert: Nach langer Pause gibt es also wieder einen Event vor Ort – mit anschließendem Apéro.

Wer den Link zur Aufzeichnung des vergangenen Talks und die Ankündigung des nächsten Events nicht ver-

passen will, meldet sich am besten bei der Meetup Gruppe an:

[www.meetup.com/Rapperswil-Frontend-Best-Practices-Meetup-Group](https://www.meetup.com/Rapperswil-Frontend-Best-Practices-Meetup-Group).

Zum Thema passt, dass am 25. Oktober der etablierte Weiterbildungskurs (CAS) Frontend Engineering Advanced beginnt. Dieser richtet sich an angehende Frontend Teamleader. Drei Alumni des Bachelors Informatik geben ihr Wissen weiter: Peter Manser, Robert Vogt und Christoph Bühler. Mehr dazu und zu anderen Weiterbildungs-Angeboten, z. B. im Bereich Machine Learning, Cybersecurity und User Experience, finden Sie hier: [www.ost.ch/de/weiterbildung/weiterbildungsangebot/informatik](https://www.ost.ch/de/weiterbildung/weiterbildungsangebot/informatik).

(SMAR)

## ZuriHac 2022 at the OST-RJ

After two years as a virtual event, ZuriHac 2022 returned as a physical event between Saturday, June 11 and Monday, June 13, 2022 at the Rapperswil campus of the OST.

ZuriHac (<https://zurihac.info>) is the world's largest practitioners' conference centred on the functional programming language Haskell. It features guest speakers including leading experts from industry and academia, and hands-on beginners and advanced tracks. Aside from the talks and tracks, attendees are encouraged to work together on all sorts of Haskell-related open-source projects and socialise.

About 400 people took part in this year's ZuriHac. Around 75% of the attendees travelled from outside Switzerland to attend, and 80% were working engineers. This year's guest speakers focused on some of the more practical aspects of programming in Haskell. Here are a few



highlights: Gabrielle Keller, Professor of Software Technology at Utrecht University, showed how Haskell can be used to develop software that targets massively parallel architectures including multicore CPUs and GPUs; Simon Peyton Jones, who recently moved from Microsoft Research to his new role as Engineering Fellow at Epic Games, talked about how the Haskell compiler is able to automatically optimise pro-

grams based on mathematically proven laws. Simon Peyton Jones is one of the key designers – and the compiler – of the Haskell language. A recording of all guest speakers and advanced session talks is available free of charge online ([https://youtube.com/playlist?list=PLOvRW\\_utVPV-nqp24VsFowilRa-m9oWrox](https://youtube.com/playlist?list=PLOvRW_utVPV-nqp24VsFowilRa-m9oWrox)).

Some of the projects that participants represented and worked on during this year's ZuriHac include the Haskell GHC compiler, related tools and integrated development environments, pandoc (a widely used universal document converter) and xmonad (a commonly used X11 window manager). You can find a complete list of all projects on the ZuriHac web page (<https://zurihac.info>).

All in-person ZuriHacs since 2017 have taken place at the OST campus in Rapperswil. ZuriHac 2023 is scheduled for June 10-12, 2023 at the OST. (MEFA)



### Impressum

Redaktion  
Studiengang Informatik, Claudia Furrer (FUCL), [ost.ch/i](mailto:ost.ch/i), T +41 58 257 49 09

Autoren  
Prof. Stefan Richter (RIST) | Thomas Keel (KETH) | Prof. Stefan Keller (KEST) | Prof. Dr. Frieder Loch (LOFR) | Prof. Dr. Farhad Mehta (MEFA) | Eliane Schmidli (SELI) | Prof. Mirko Stocker (STM7) | Prof. Dr. Markus Stolze (SMAR) | Prof. Dr. Olaf Zimmermann (ZIOL)