

Objektyp: **Issue**

Zeitschrift: **Visionen : Magazin des Vereins der Informatik Studierenden an der
ETH Zürich**

Band (Jahr): - **(2013)**

Heft 6

PDF erstellt am: **28.05.2024**

Nutzungsbedingungen

Die ETH-Bibliothek ist Anbieterin der digitalisierten Zeitschriften. Sie besitzt keine Urheberrechte an den Inhalten der Zeitschriften. Die Rechte liegen in der Regel bei den Herausgebern.

Die auf der Plattform e-periodica veröffentlichten Dokumente stehen für nicht-kommerzielle Zwecke in Lehre und Forschung sowie für die private Nutzung frei zur Verfügung. Einzelne Dateien oder Ausdrucke aus diesem Angebot können zusammen mit diesen Nutzungsbedingungen und den korrekten Herkunftsbezeichnungen weitergegeben werden.

Das Veröffentlichen von Bildern in Print- und Online-Publikationen ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. Die systematische Speicherung von Teilen des elektronischen Angebots auf anderen Servern bedarf ebenfalls des schriftlichen Einverständnisses der Rechteinhaber.

Haftungsausschluss

Alle Angaben erfolgen ohne Gewähr für Vollständigkeit oder Richtigkeit. Es wird keine Haftung übernommen für Schäden durch die Verwendung von Informationen aus diesem Online-Angebot oder durch das Fehlen von Informationen. Dies gilt auch für Inhalte Dritter, die über dieses Angebot zugänglich sind.

Ein Dienst der *ETH-Bibliothek*
ETH Zürich, Rämistrasse 101, 8092 Zürich, Schweiz, www.library.ethz.ch

<http://www.e-periodica.ch>

717 826



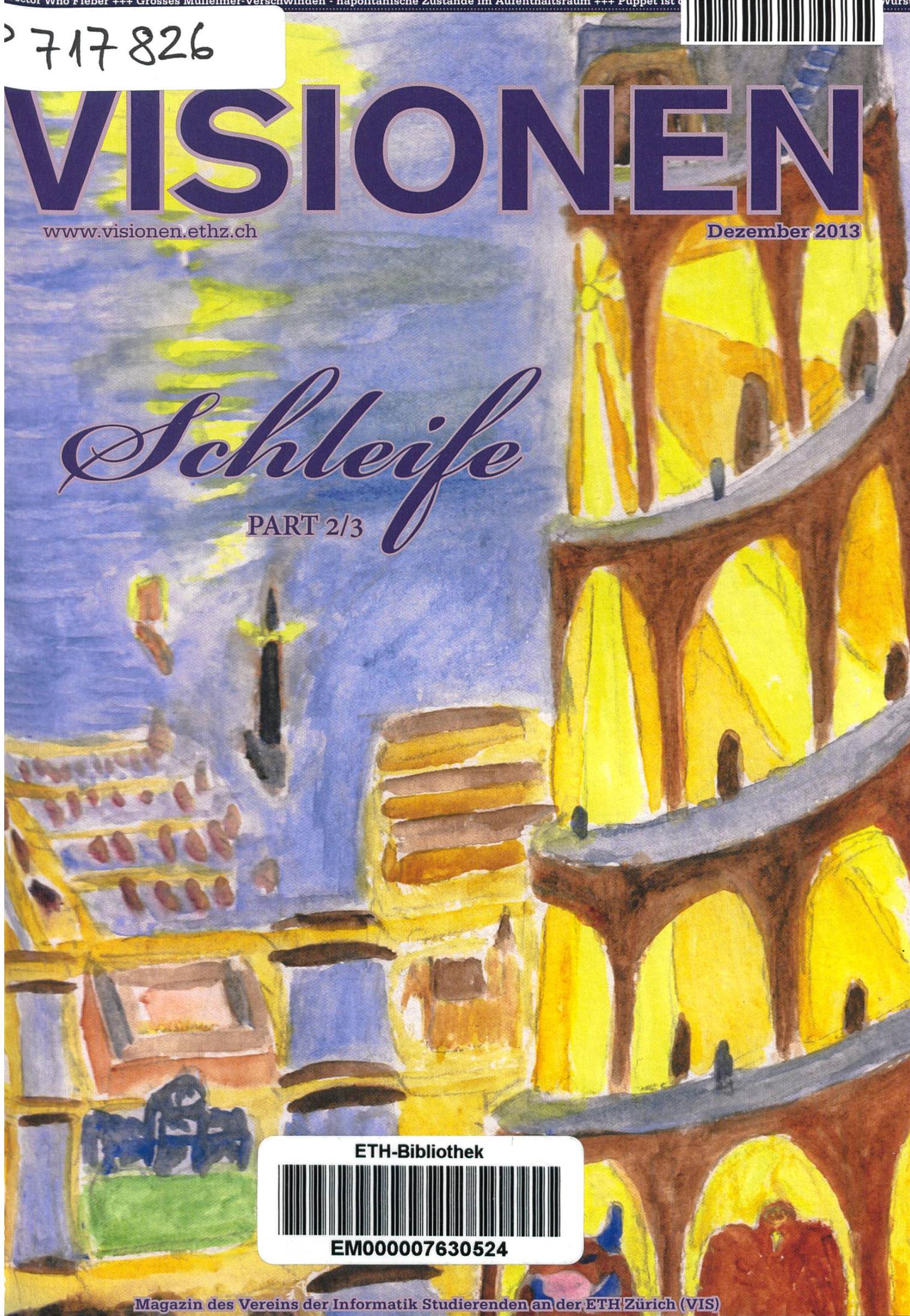
VISIONEN

www.visionen.ethz.ch

Dezember 2013

Schleife

PART 2/3



ETH-Bibliothek

EM000007630524



Open Systems gehört mit seinen Mission Control Security Services im Bereich IT-Sicherheit zu den europaweit anerkannten Anbietern. Wir arbeiten von Zürich und Sydney aus in einem dynamischen Umfeld in über 175 Ländern. Bei uns kannst Du Dein Wissen in einem jungen Team in die Praxis umsetzen und rasch Verantwortung übernehmen. Infos über Einstiegs- und Karrieremöglichkeiten sowie Videos findest Du auf unserer Website. www.open.ch

ETH-ZÜRICH

06. Jan. 2014

BIBLIOTHEK

Editorial

Geehrte Leserinnen und Leser,

Seit einigen Wochen beschäftige ich mich ab und an mit der Programmiersprache Scala^[1]. Dazu habe ich mir natürlich ein Buch gekauft. Ein schweres Buch, geschrieben vom Erfinder der Sprache, Professor Martin Odersky. Manchmal seufze ich beim Hervorkramen dieses dicken Schunkens^[2] – gibt es wirklich noch soviel neu zu erfinden an einer Programmiersprache? Oder anders gefragt: Wie viel gibt es denn noch zu lernen, bis man die Kunst der Programmierung in ihrer Vollkommenheit beherrscht?

Programmiersprachen gibt es unterdessen wie Sand am Meer. Und jede neue Programmiersprache eröffnet einen neuen Blickwinkel auf ein und dasselbe. Scheinbar simple Konzepte, wie das einer Schleife, werden stets neu erfunden. Und jeder Advokat weiss, warum man Schleifen in der Art schreiben sollte, wie es seine Sprache ihm vorschreibt. Und so weiss uns auch Martin Odersky anhand einiger Beispiele zu erklären, warum die Schleifen in seiner Programmiersprache schöner sind – und unbestrittenermassen sind sie das.

Die Vielfalt ist aber auch ein Graus – vor allem für den Ideologen. Denn für letzteren zählt nur die ultima ratio, und diese zu finden treibt ihn mitunter in den Wahnsinn. Wie wollten wir die Idealität eine Sprache messen? Was ist das Mass aller Dinge? Programmlänge? – Dann würden alle in Haskell programmieren ... Compiler-Komplexität? Maschinennähe?

Daran erkenne ich zweierlei: Innovation ist eine Form der Freiheit. Wer stets die Frage nach alten Idealen stellt und die ultima ratio sucht, wird sich augenblicklich im Korsett der Idealmasse wiederfinden – und im Zweifelsfalle beim Alten bleiben. Zweitens, vielleicht ist das Konzept einer Programmiersprache in gewisser Hinsicht auch inhärent unzulänglich, weil es prinzipiell keinen Formalismus gibt, der das Vollkommene fassen kann.

So bleibt uns wohl auch in Zukunft nichts anderes übrig, als uns weiterhin durch dicke Schunken zu quälen und uns der Schönheit neuer Schleifenkonstrukte zu erfreuen.

Ich wünsche ein spannendes Leseerlebnis!

Euer Chefredaktor,



Stefan Dietiker

Bemerkungen

[1] <http://www.scala-lang.org>

[2] Ok, ok, es gibt Schlimmeres. So denken die meisten beim Stichwort «dicker Schunken» wohl an ein anderes Werk, mit dem sie im ersten Semester Bekanntschaft machten.

Inhalt

Schleife

| | |
|---|----|
| Stellensuche und Bewerbungsschreiben | 6 |
| Around The World – New Zealand | 12 |
| Die Bachelorarbeits-Trilogie: 2. Teil – while (!Abgabetermin) fixBug(); | 16 |
| An Introduction to Puppet | 20 |
| Spielen, Sterben, Savegame laden | 28 |

VIS

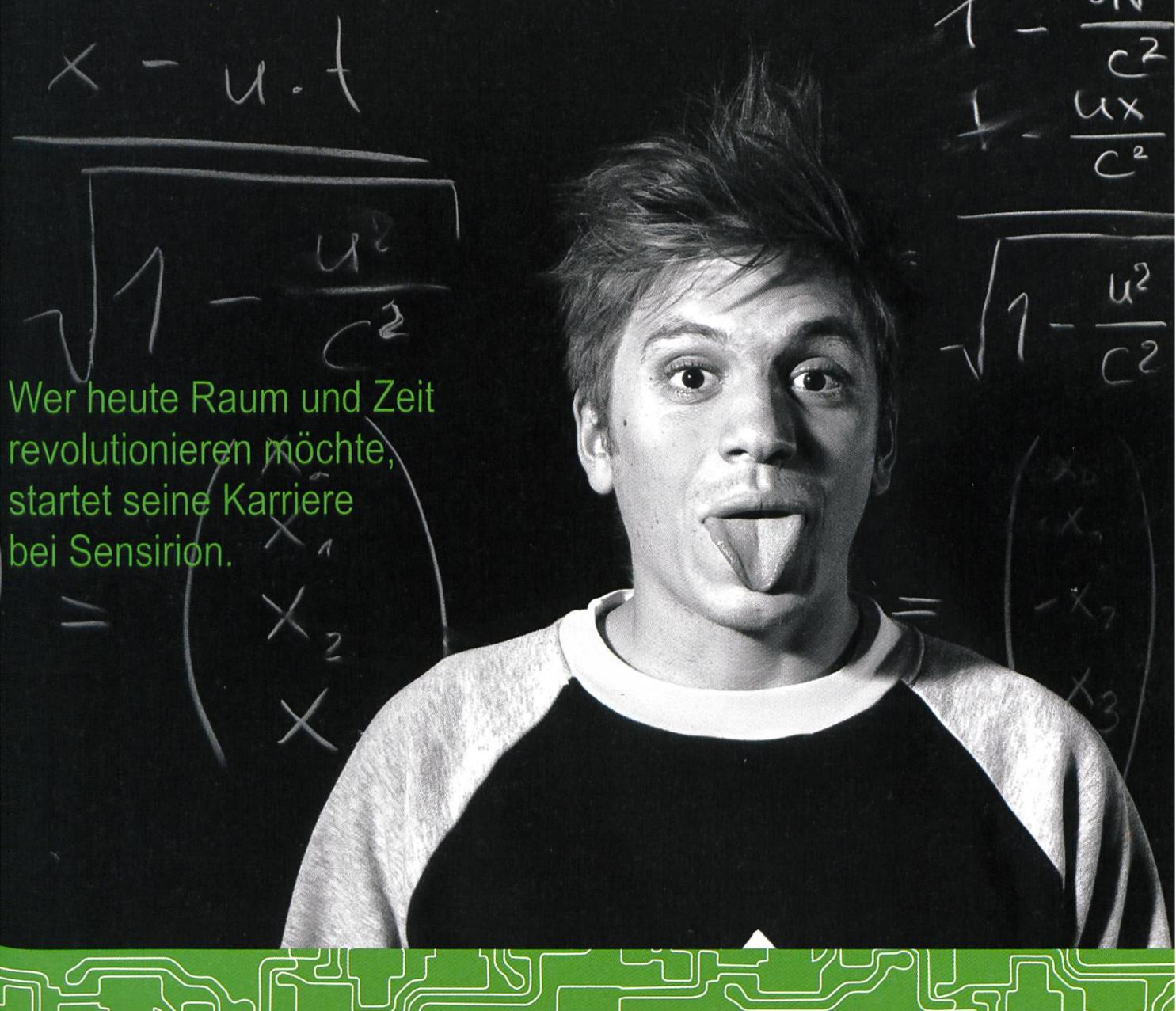
| | |
|----------------------------------|----|
| Notenstatistik | 32 |
| ACM ICPC SWERC13 in Valencia | 38 |
| VISky Degu HS13 | 43 |
| Ein «Danke» kann Berge versetzen | 44 |
| FIGUGEGL | 46 |

Berichte & Meinungen

| | |
|--|----|
| Klimaskeptizismus | 48 |
| A Short Story on Learning Russian | 50 |
| IAETH Portrait Letter 019: Georg Schätti | 54 |
| Parkour | 56 |

Spass

| | |
|----------------------------------|----|
| Beni Koller: Leidensgemeinschaft | 62 |
|----------------------------------|----|



Wer heute Raum und Zeit revolutionieren möchte, startet seine Karriere bei Sensirion.

Und wird Teil der Sensirion-Story: Sie freuen sich auf Herausforderungen, bei denen Sie Ihr ganzes Wissen und Ihre ganze Persönlichkeit einbringen können. Dann heißen wir Sie herzlich willkommen bei Sensirion.

Sensirion ist das weltweit führende und mehrfach preisgekrönte Hightech-Unternehmen auf dem Gebiet der Feuchtesensoren und Durchflusssensoren – mit Niederlassungen in Übersee und im Fernen Osten. Dank unserer einzigartigen CMOSens® Tech-

nologie vereinen wir das Sensorelement mit der digitalen Auswerteelektronik auf einem winzigen Siliziumchip. Damit verschieben wir die Grenzen des Messbaren ins schier Unermessliche.

Schreiben Sie Ihre eigenen Kapitel der Sensirion-Erfolgsgeschichte und übernehmen Sie Verantwortung in internationalen Projekten. Schicken Sie uns Ihre Bewerbungsunterlagen und stimmen Sie sich auf www.sensirion.com/jobs auf eine vielversprechende Zukunft ein.



SENSIRION
THE SENSOR COMPANY

Stellensuche und Bewerbungsschreiben

GREGOR WEGBERG – GET EMPLOYED

Dein CV mag ja gut und recht sein, doch wie bringst du jemanden dazu, es sich überhaupt anzusehen? Wie kommt man am ersten Filter der HR-Abteilung vorbei?

Bevor wir das Thema der Stellensuche bestreiten und uns ausführlicher mit dem Bewerbung- bzw. Motivationsschreiben befassen, die obligatorische Erinnerung zu dieser Visionen Artikelreihe. Die nachfolgenden Empfehlungen beziehen sich primär auf Bewerbungen für eine Arbeitsstelle in der Schweizer Privatwirtschaft. Vieles mag allgemeingültig sein, doch es ist, wie beim letzten Mal, Vorsicht geboten.

Stellensuche: Vorbereitung

Beim Schreiben deines Lebenslaufs lohnt es sich stets im Kopf zu behalten, bei welchen Unternehmen du dich überhaupt bewerben möchtest. Doch wie findet man solche Unternehmen? Wie findet man überhaupt eine Arbeitsstelle?

Hierzu solltest du dich zuerst fragen, wie du dir deinen Arbeitgeber und deine Arbeitsstelle vorstellst. Dabei geht es vor allem um folgende Fragen:

- Welche Unternehmensgrösse strebst du an? Mögliche Antworten können vom Spin-off/Start-up bis zum international tätigen Konzern reichen.
- Welche Geschäftsfelder bzw. Branchen interessieren dich? Hier lohnt es sich auch zu überlegen, welche Geschäftsfelder du meiden möchtest (z. B. aus ideologischen Gründen).
- Wie stellst du dir die Unternehmenskultur vor? Möchtest du mit deinen Kollegen freundschaftlich umgehen und Abends ein Bier trinken, oder trennst du Arbeit und Vergnügen strikt voneinander? Hier geht es aber auch um die Arbeitsweise. Möchtest du in einem sehr dynamischen und innovativen Unternehmen arbeiten, oder doch lieber in einem traditionellen?
- Wie sicher soll dein Arbeitsplatz sein? Bist du bereit, für ein Unternehmen zu arbeiten, das es möglicherweise morgen nicht mehr gibt?
- Welche Einkommensvorstellungen hast du? Wie viele Ferientage sind für dich das Minimum?
- Wo sollte sich der Arbeitsort befinden? Bist du bereit, in einem anderen Land zu arbeiten?

Viele dieser Fragen sind voneinander abhängig. So beisst es sich in den meisten Fällen, wenn du für ein Start-up arbeiten möchtest, gleichzeitig aber auch einen sehr sicheren Arbeitsplatz suchst. Es lohnt sich offensichtlich, sich Gedanken zu machen, wie der zukünftige Arbeitgeber/-platz aussehen sollte.

Vielen fällt aber gerade dieser sehr schwer. Wie soll man den überhaupt wissen, wo man später arbeiten möchte? Hier kann ich vor allem drei Wege empfehlen: (1) Möglichst viele verschiedene Unternehmen besuchen. Zum Beispiel bietet der VIS immer wieder Firmenexkursionen^[2] an, auf denen du unterschiedlichste Unternehmen kennen lernen kannst und einen Blick in die Arbeitswelt erhältst. (2) Eine weitere Möglichkeit sind Jobmessen, an denen du ins Gespräch mit Mitarbeitern verschiedener Unternehmen kommen kannst. Hier kannst du in Erfahrung bringen, wie es ist, für ein Unternehmen zu arbeiten und wie der Alltag aussieht. Auch die Firmenkultur kommt in solchen Gesprächen meistens sehr gut hervor. Allem voran muss hier natürlich die «Kontaktparty»^[3] des VIS am 08.03.2014 erwähnt werden. (3) Meine letzte Empfehlung, und persönlicher Favorit, ist das Gespräch mit Freunden und Kollegen. Es ist sehr wahrscheinlich, dass du bereits viele Personen kennst, die in verschiedenen Unternehmen und verschiedenen Positionen gearbeitet haben. Rede mit ihnen über ihren Arbeitsalltag und wie sie diesen erleben.

Je mehr du in Erfahrung bringst, desto eher wirst du herausfinden, was du suchst.

Deine Erfahrung

Hast du Gutes oder Schlechtes erlebt beim Bewerben? Bist du nicht gleicher Meinung und möchtest eine andere Sichtweise auf die Angelegenheit aufzeigen? Dann schreib mir doch einen Leserbrief oder gar einen ganzen Artikel und schick ihn an visionen@vis.ethz.ch. Wir, und im Besonderen auch ich, würden uns sehr über Feedback und Kommentare freuen!

Stellensuche: Die Suche

Zur Stellensuche selbst gibt es nicht all zu viel zu sagen. Hier geht es vor allem darum, alle Wege, die es gibt, auszunutzen. Du möchtest ja nicht deine Traumstelle verpassen, nur weil du zu faul warst, eine Webseite mehr besucht zu haben.

Die typischen Optionen, eine Stelle zu finden, sind:

- Online-Stellenportale^[1],
- Webseite des Unternehmens, für das du gerne arbeiten möchtest,
- Freunde und Verwandte,
- Jobmessen,
- Ehemalige Arbeitgeber bzw. ehemalige Arbeitskollegen,
- Zeitungen und Inserate in Fach- und Vereinszeitschriften,
- Blindbewerbungen und vieles mehr.

Falls du eine Stelle auf einem Stellenportal findest, empfehle ich dir, auch einen Blick auf die Firmenwebseite zu werfen. Manchmal sind die Ausschreibungen auf Stellenportalen nicht mehr aktuell oder du findest eine interessantere Stelle direkt auf der Webseite. Ein Blick lohnt sich sicher, du musst die Webseite ja sowieso kurz besuchen, um einen Eindruck vom Unternehmen zu erhalten.

Blindbewerbungen lohnen sich vor allem, wenn du besonders motiviert bist, für ein spezifisches Unternehmen zu arbeiten. In einem solchen Fall ist es besonders wichtig, deine Motivation hervorzuheben in der Bewerbung! →

Bewerbungsschreiben 1.0 (Postweg)

Eine Bewerbung besteht meistens aus den folgenden Teilen:

1. Bewerbungsschreiben,
2. Lebenslauf,
3. und Kopien relevanter (nicht aller) Diplome, Zertifikate, Zeugnisse, etc.

Dabei ist es wichtig anzumerken, dass dies auch die typische Reihenfolge ist, wie die Unterlagen in einem Umschlag verschickt werden.

Es folgt sofort, dass das Bewerbungsschreiben das erste Dokument ist, welches durch einen zukünftigen Arbeitgeber gesehen und gelesen wird. Dabei möchte ich im Besonderen betonen, dass es gesehen und gelesen wird. Eine dreiseitige Abhandlung, wieso du der Grösste bist, wird sehr wahrscheinlich gar nicht erst gelesen werden. Hier ist also Kürze und hoher Informationsgehalt wichtig. Das Bewerbungsschreiben wird auch gerne als «Bewerbungsbrief» bezeichnet. Im Wort «Brief»

liegt bereits die nächste Empfehlung; das Erscheinungsbild des Bewerbungsschreibens sollte einem Brief entsprechen. Dazu gehört die Anschrift des Empfängers und

Absenders, Ort und Datum, eine Betreffzeile, Anrede, der eigentliche Inhalt, abschliessender Gruss, Unterschrift (von Hand unterschrieben!) und die Liste der Beilagen.

Beim Inhalt gibt es vier zentrale Abschnitte, die du behandeln solltest.

Die **Einleitung** soll den Leser informieren, für welche Stelle du dich bewirbst und wie du auf diese gekommen bist. Hier ist darauf zu

achten, dass der Anfang nicht zu formell und trocken daher kommt. Es darf dem Leser bereits etwas Honig um den Mund geschmiert werden.

Im zweiten Abschnitt geht es um deine **fachliche Qualifikation**. Hier beschreibst du in Kürze deine aktuelle Arbeitssituation und wieso du dich für die ausgeschriebene Stelle qualifizierst.

Im dritten Abschnitt gehst du auf deine **Motivation** ein. Wieso soll das Unternehmen dich einstellen und nicht jemand anderes?

Als letztes folgt der **Abschluss**, in welchem du dich kurz und bündig dazu äusserst, dass du dich gerne zu einem persönlichen Gespräch mit dem Leser treffen würdest.

Für praktische Beispiele und Tipps empfehle ich dir wärmstens, einen Blick in den Bewerbungsratgeber des ETH Careercenter^[4] zu werfen. Die darin enthaltenen Empfehlungen zum Bewerbungsschreiben decken sich komplett mit den meinen.

Trotzdem möchte ich noch einige zusätzliche Anmerkungen machen. Bei der Stellensuche

solltest du Ausschau halten, ob bestimmte Vorgaben gemacht werden, die den Bewerbungsbrief betreffen. Manche Stellenausschreibungen enthalten Identifikationsnummern. Diese

gehören in den meisten Fällen in die Betreffzeile oder in den ersten Abschnitt des Briefes. Oft wird auch eine bestimmte Person als Empfänger für Bewerbungen angegeben. In einem solchen Fall sollte dein Bewerbungsbrief entsprechend adressiert werden, und die Anrede ist ebenfalls anzupassen. Achte darauf, das Schreiben so ausulegen, dass es dem Empfänger erklärt, wieso sie mit dir die richtige Wahl treffen würden. Es

**Eine dreiseitige
Abhandlung, wieso du der
Grösste bist, wird sehr
wahrscheinlich gar nicht
erst gelesen werden.**

soll im Schreiben nicht darum gehen, wie toll du bist, sondern wie genial die Entscheidung wäre, dich für die ausgeschriebene Stelle einzustellen. Das Bewerbungsschreiben soll den Leser dazu bringen, deinen Lebenslauf zu lesen, sich also mit dir weiter zu befassen.

Lege besonders viel Wert auf Rechtschreibung! Briefe mit Tippfehlern oder gar Rechtschreibfehler machen sehr schnell einen sehr negativen Eindruck beim Leser. Gib also dein Schreiben anderen Personen zur Korrektur!

Bewerbungsschreiben 2.0 (E-Mail)

Hach, die moderne Welt. Bewerbungen per E-Mail gehören nicht gerade zu meinen Favoriten. Sie sind aber Realität und ich habe das Gefühl, dass sie immer beliebter werden. Vor allem gibt es gefühlt immer mehr Stelleninserate, die einem die Bewerbung per E-Mail aufzwingen.

Grundsätzlich entspricht eine Bewerbung per E-Mail einer Bewerbung auf dem Postweg. Natürlich gibt es aber auch spezifische Details, auf die man zu achten hat.

Alle Unterlagen (Bewerbungsschreiben, CV, Beilagen) müssen in elektronischer Form vorliegen. Dabei ist vor allem darauf zu achten, dass all diese Unterlagen in der Summe nicht mehr als 2 MB einnehmen. Dies hängt, wie es dir als Informatiker bekannt sein sollte, mit der Limite für E-Mail Grössen zusammen. Viele Unternehmen haben auch heute noch eine 2-MB-Limite. Gleichzeitig sollten die Unterlagen in genug guter Qualität vorliegen, dass sie in gedruckter Form weiterhin einen anschaulichen Eindruck machen.

Gib den Dateien sinnvolle Namen. Dabei empfiehlt es sich, im Dateinamen auch deinen Familiennamen einzufügen. Beispielsweise: CV_GWegberg.pdf, Begleitschreiben_GWegberg.pdf, usw.

Der Text in der E-Mail sollte kurz und formell sein. In diesem Text begrüsst du den Empfänger, machst eine Referenz auf die Stellenausschreibung und verweist auf die angehängten Dateien. Sei vorsichtig, der Text in der E-Mail sollte nicht bereits dein Bewerbungsschreiben sein.

Auch hier ist auf Rechtschreibung zu achten. Nur weil es sich um eine E-Mail handelt, ist es nicht sinnvoll, den Text mal so schnell nieder zu tippen. Betrachte die E-Mail als einen Brief. Am besten schreibst du sie auch im Texteditor deiner Wahl und gibst auch diesen sehr kurzen Text zur Korrektur an Freunde und Verwandte.

Bewerbungsschreiben 3.0 (Onlineformular)

Auch Bewerbungen sind im Web 3.0 angekommen. Was auch immer dies nun heissen mag. Gerade grosse Unternehmen haben eigene Stellenportale mit Formularen, über welche man sich direkt für eine Stelle bewerben kann.

Hier gelten alle bisher aufgelisteten Empfehlungen und Warnungen. Zusätzlich solltest du dich mit dem Formular auseinandersetzen. In den meisten Fällen erwarten sie eine feste Anzahl an PDF Dokumenten, die wiederum bestimmte vordefinierte Inhalte zu enthalten haben. Bereite zuerst all diese Unterlagen vor und gib sie gegebenenfalls zur Korrektur.

Erachte alle Textfelder, die du ausfüllst, als Bewerbungsschreiben: Peer Review ist das A und O!

Peer Review ist das A und O!



Behalte den Überblick

Eine der grossen Herausforderungen für mich ist es jeweils, den Überblick über alles zu behalten. Verschiedene Versionen von Lebensläufen, eingescannte Unterlagen in verschiedenen Auflösungen und Kombinationen, Bewerbungsbriefe für verschiedene Unternehmen und Stellen etc. pp.

Es empfiehlt sich, von Anfang an eine gewisse Ordnung einzuhalten. Dies hat den Vorteil, dass die so entstehenden Unterlagen und Vorlagen auch noch in ein paar Jahren von grossem Nutzen sein werden, wenn man eine neue Stelle sucht.

Ich persönlich organisiere meine Unterlagen wie folgt. Als Erstes habe ich einen Ordner mit allen Originaldokumenten. Dazu gehört der Lebenslauf in seiner ungekürzten Version und alle eingescannten Diplome, Zertifikate, Auszeichnungen und so weiter in möglichst grosser Auflösung und als einzelne PDF-Dokumente. Der Inhalt dieses Ordners stellt also in gewissem Masse die Bausteine dar, aus denen ich meine Bewerbungen zusammenbaue.

Als Nächstes folgt ein Ordner, in dem ich Lebensläufe und zusammengesetzte PDF-Dokumente der verschiedenen Unterlagen aufbewahre. Diese Dokumente helfen vor allem dabei,

Arbeit zu sparen und nicht bei jeder Bewerbung alles von Neuem zusammensetzen zu müssen. Dabei passieren ja doch gerne mal Fehler.

Nun kommen wir zum wichtigsten Ordner für mich. Dieser enthält für jede Bewerbung einen Unterordner, der mit dem Firmennamen und Stellenbezeichnung beschrieben ist. Ein Plaintext-Dokument hilft mir zusätzlich, den Überblick zu behalten, welche Bewerbungen in welchem Zustand sind («in Vorbereitung», «in Korrektur», «bereits zum Verschicken», «verschickt», «abgelehnt» oder «Bewerbungsgespräch»). In diesen stellenspezifischen Ordnern liegen alle Dokumente, die ich an das Unternehmen zu verschicken gedenke bzw. verschickt habe. Dazu gehören auch Ausdrucke von verschickten und empfangenen E-Mails und Onlineformularen. Zusätzlich enthalten bei mir diese Ordner Plaintextdateien mit Informationen zum Unternehmen, die ich im Laufe der Bewerbung gesammelt habe. Diese Informationen sind besonders nützlich beim Schreiben des Bewerbungsbriefes und im Falle eines Bewerbungsgesprächs. So wird es dir auch nicht passieren, dass du dich für eine Stelle zwei Mal bewirbst. Auch wirst du dankbar sein, diese Dokumente zu haben, wenn du dich das nächste Mal auf Stellensuche machst! 

Referenzen

[1] In keiner bestimmten Reihenfolge:

- <http://www.jobs.ch> bzw. direkt <http://ictcareer.ch>
- <http://www.jobscout24.ch>
- <http://www.eth-gethired.ch/>
- <http://jobs.acm.org/>
- <http://careers.ieee.org/>
- <http://careers.stackoverflow.com/>
- ... und viele mehr

[2] http://www.vis.ethz.ch/de/current/excursions_seminars/list

[3] <http://www.kontaktparty.ch>

[4] <http://www.careercenter.ethz.ch/students/applicationguide>

Hast du **VISIONEN?**

Hier ist noch Platz für deinen Artikel.

Hast du etwas, das du deinen Mitstudenten mitteilen willst? Warst du im Ausland und willst von deinen grossartigen Erfahrungen berichten? Hast du ein Praktikum gemacht und möchtest erzählen, was du erlebt hast? Dann nutze die VISIONEN als Plattform und schick uns deinen Artikel! Alle Infos unter <http://www.vis.ethz.ch/de/visionen/articles>.

Oder möchtest du aktiv mithelfen? Dann werde Mitglied der VISIONEN-Kommission. Einfach E-Mail an visionen@vis.ethz.ch.

Around The World – New Zealand

LUKAS LIMACHER – COUNTED MANY SHEEP, STILL NOT TIRED

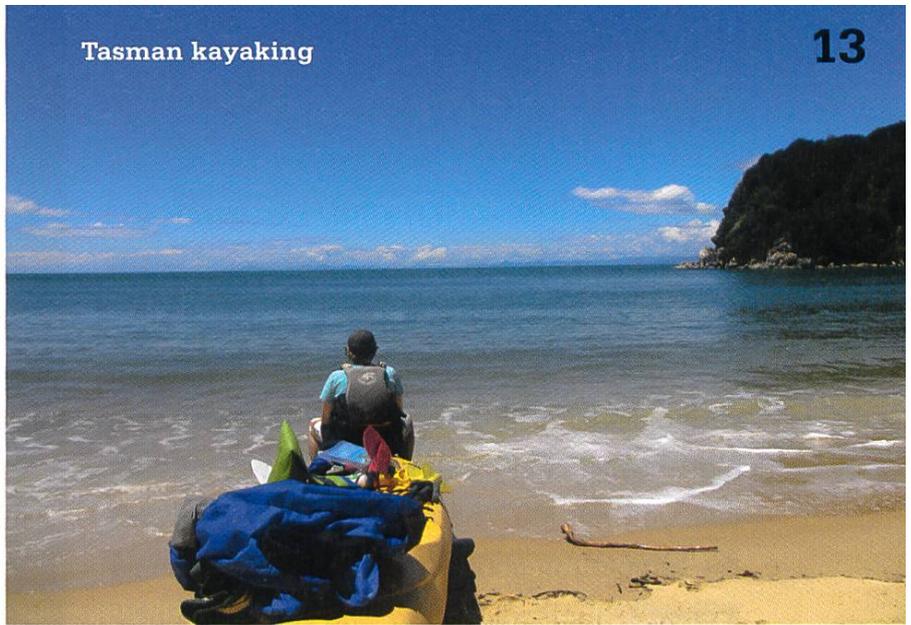
Have you ever visited a language school? It's a great way to learn English and to get to know different cultures and people all at the same time! So my friend and I decided to visit one in Auckland, New Zealand, and then travel around. It was our next destination after Hawaii. So today, I will tell you about the land of the Māori, sheep, Hobbits, and Kiwis.

Honestly, I was not tired of travelling at all and addicted to surfing (from Hawaii). Anyway, It felt great to visit a language school and to study English again. Especially the differences between British, American, and New Zealand English. In addition, it wasn't too bad to be at one place for a while, and I got to know a Kiwi family. Did you know that Kiwi has three meanings? One for the people in New Zealand, the fruit, and the well-known little bird. Unfortunately, the little, flightless birds endemic to New Zealand are endangered. Their main threat are invasive predators such as opossums. Everybody travelling to New Zealand will realise: they care about their native nature very much. Every foreign organic item is carefully controlled. For instance, dirty shoes are cleaned and it is forbidden to import most food on your own. Indeed, when entering New Zealand, my friend had to show his hiking boots, which the border control cleaned. Not only the Kiwis, but also the native plants are threatened by invasive competitors. A remark worth mentioning is: Although being fussy, the New Zealand border control seemed not only correct but a thousand times more friendly than the evil ones from the USA—sorry USA!



Auckland

So I went to school again—for 6 weeks. Fun times! The school was in the middle of Auckland, which has everything in its centre and yet it is not too big. It is a truly multicultural city: From Kiwis over Māori (the original inhabitants) to various people from the Pacific regions. During the whole time in New Zealand, I had the feeling that the Māori culture is respected well and also explained a lot. New Zealand is proud of it.



However, the influence of the “modern time” is clearly present. I became used to my host family quickly. By the way, the mother of the family I lived with worked as a prison warden and did talk to “Kim Dotcom” while he was in prison there. In addition, I received a number of insights into typical Kiwi life, such as some specialities regarding food. For example, Vegemite is really popular in New Zealand. It is made out of yeast extract, which is a leftover produced when brewing. If you are curious: try tasting Cenovis in Switzerland, it’s pretty much the same thing. It looks like chocolate, but tastes very bitter. As you might think, the Kiwi fruit is widely available and eaten. Tasty!

Things changed: I wanted to live close to the city, so I moved to the centre of Auckland after a while. During our time in Auckland, we made trips to nearby national parks and regions. These trips showed us the great variety of New Zealand: from dream beaches over jungle to desert—you can see everything! We did not miss any chances and made lots of trips on weekends. Let me tell you about some of our adventures.

Hobbiton was on our must-do list. It is a whole Hobbit village used for all “The Lord of The Rings” movies, as well as for “The Hobbit”, built in reality! It’s crazy that a whole little village has been built to that purpose! On the one hand, most houses are fake inside, but on the other hand, the huge tree you might remember from the movies has been moved there from

far away. In addition, the tree on the huge hill (Bilbo’s house) is fake: it’s made out of carbon. On the way to Hobbiton we saw beautiful landscapes and a lot of sheep. Honestly, there are sheep everywhere.

Another time, we went east of Auckland: to Coromandel, featuring breathtaking jungle scenery. I took the chance and rented a kayak to explore some of the rivers in the Hawaii-like scenery. If you go to the north of Auckland, you get to the 90 mile beach. We made a trip there, too, as it’s a highlight on the northern part of the island. As the name suggests, it is an enormously long beach in the north of New Zealand. Despite its name, it’s only 55 miles long and classified as a highway: we drove along it with our bus to reach beautiful remote beaches. On the way back, we visited a small desert to cover us in sand. Sand surfing! Definitely an interesting activity, but wave surfing is still better.

After some time, Christmas came near, which was strange as in December the summer really begins in New Zealand. It’s the perfect time there to enjoy swimming, sun bathing, and writing back to Switzerland to make other people jealous. All of Auckland got decorated with the standard Christmas decoration. It really



felt strange to me, as if it was not making much sense. We finished school before Christmas, but not without a big event: going west to a beach with black sand. It is very rough: huge waves in high frequency and strong currents which could take you out into the ocean. I could not resist the temptation and surfed for a while. We had a great school “graduation” and chilled in Auckland, getting to know some cool local, whom we could spend Christmas with. They had a huge, shared flat—awesome mate! Afterwards, we headed south for some more neat town visits, and finally flew to Queenstown shortly before New Year’s Eve. As a small hint for you, we mainly travelled using „Nakedbus” (“stripping the cost of travelling”).

Queenstown

Queenstown was amazing. So many people, so many bars and so many people we knew: from our school, from Auckland, or from other hostels. That’s one great preciousity about travelling in New Zealand: wherever you travel, you will see some people you have seen before. Queenstown is known as the adrenaline junkie’s place: we couldn’t resist and tried sky diving. It gives a unique feeling of freedom and excitement! In addition, I did a bungee jump into the river—what a refreshment! Last but not least, I did a rope swing: free fall which then transitions



Bungeejumping in Queenstown

to a swing through a cliff. Among other things, Milford Sound is definitely worth mentioning. It consists of steep mountains covered by jungle trees and waterfalls. Truly a special place. Later in New Zealand, an experienced kayaker would tell us his story about a two week expedition at the very most remote regions in Milford Sound.

The adventure continues

After Queenstown, we visited, among other places, Dunedin, the students town in New Zealand, and went to the steepest street of the world, which was built there. Looking for an opportunity to visit the cool beaches and nature in the south of the southern island, we ended



The mountains around Queenstown

up hitch-hiking with two girls. That was really impressive, seeing how they took to saving money while travelling to the extreme. After this fancy travelling variant, we ended up in Nelson, having seen some opossums and glow-worms along the way. Once there, we experienced the most exciting of our adventures, a three day trip in two days: kayaking, hiking and again hiking. We kayaked with seals and touched jellyfish. Later, we hiked to Bark Bay. We slept on a little camp site, next to this remote beach. Here we met the previously mentioned, experienced kayaker: he told us about the plague of the opossums, threatening the local birds by eating their eggs. He also told us about an uncommon, but in New Zealand very real, threat in the jungle: wild calves or cows. They hide in the jungle and overrun you. Clearly, this doesn't sound as scary as the the animals in Australia can pretty much all kill you. He also told us about his expedition to Milford Sound. Impressively, he and his friends were dependent on catching fish for food. After thirteen days in truly untouched nature, they were picked up by the navy. The next day, we hiked back to the starting point and were rewarded with wonderful views. Sometimes, we

even hiked barefoot: the track partly follows the beach (coast track) and is only dry if the tide is low. As luck had it, the tide kicked in and we had to hurry up.

Finally, we ended up in Christchurch. Sad but true: most of its down-town has been destroyed by an earthquake and is now closed. We went back to Auckland to say goodbye to some of our friends still in school. Sad to leave awesome

New Zealand behind, we went off for a new adventure: the mega city of Hong Kong. ⌘



Die Bachelorarbeits-Trilogie

2. Teil – while (!Abgabetermin) fixBug();

MANUEL BRAUNSCHWEILER – BEKÄMPFT TERMITEN

Blöde Frage: Wer von euch hätte gedacht, dass der schwierigste Teil der Bachelorarbeit mit dem Unterschreiben eines unscheinbaren Zettelchens beginnt – eines Zettelchens, das den bedeutungsschweren Titel «Registrierung zur Bachelor-Arbeit» trägt? Wohl alle! Ebenso, wie vermutlich alle den dafür vorgeschlagenen 300 Stunden nur ein müdes Lächeln schenken und diese gleich als masslose Untertreibung abstempeln. Doch was mir während diesen 300++ Stunden alles geschehen ist (und auch euch geschehen kann), das lest ihr hier.

Der eigentliche Anfang meiner Bachelorarbeit war wie meine ersten paar Vorlesungen in Numerik I. Soll heissen: ich verstand überhaupt nichts. Die freundliche Atmosphäre des mit Disney-Bildern gespickten Zimmers half mir da wenig. Konzentriert folgte ich den Ausführungen meines Betreuers, und obwohl ich dabei immer wieder nickte und vorgab, ich würde den grundlegenden Gedanken verstehen, so arbeitete mein Gehirn auf Hochtouren daran, das verwirrende Muster, das sich in meinen Gedanken dazu bildete, zu lösen. Leider ohne Erfolge – ich verstand noch zu wenig von der gesamten Thematik. Das Beunruhigende dabei war, dass das Resultat meiner Arbeit nicht «nur» meine Note bestimmen würde, sondern indirekt auch die Grundlage für die Masterarbeit eines anderen Studenten bildete. Soll heissen, dieser Student wollte meine Resultate in seinem Projekt verwenden. Sollte ich also scheitern, könnte dies den anderen Student in arge Probleme bringen ... bloss kein zusätzlicher Druck und so.

Fremde im Haus

Deswegen fand bereits das zweite Treffen nicht nur mit meinen beiden Betreuern, sondern auch mit dem Masterstudent und dessen Betreuer statt. Nachdem alle den NDA-Vertrag unterschrieben hatten, damit auch ja niemand die Idee klauen konnte, wurde der Ansatz erneut besprochen. Es war schon Mai und die Räume von Disney Research Zürich waren von der Sonne und deren drückenden Wärme erfüllt. Für mich war es nun schon das zweite Mal, dass ich der grundlegenden Idee des Algorithmus lauschen konnte, doch musste ich feststellen, dass ich seit dem letzten Mal kaum schlauer geworden war. Nun gut, ich hatte ja auch noch Vorlesungen und Übungsserien, um welche ich mich in der Zwischenzeit hatte kümmern müssen, doch ärgerte es mich dennoch. Zweifelnd und beunruhigt zugleich sass ich im stoffbezogenen Sessel des Büros und beobachtete, wie sowohl der Masterstudent als auch dessen Betreuer (beide nicht von der ETH) danach nickten und meinten, dass dies eine gute Idee sei. Hatten die das wirklich alles verstanden?! Wie sich erst Monate später herausstellen würde, war

dem nicht wirklich so ... aber dass Menschen einfach mal nickten, obwohl sie nicht verstanden haben, worum es ging, war ja nichts Neues – sei es im Ausgang beim Flirten oder eben bei einem Fachgespräch über allerlei mathematische Formeln (ja, ich schaue beschuldigend auf mich). Erst als mir mein Betreuer im Anschluss an dieses Treffen eine Mail schrieb, wo sein Ansatz in acht Punkten geschildert stand, begann sich der Knoten in meinem Kopf zu lösen, und ich konnte wirklich mit der Arbeit beginnen.

Altlasten

An dieser Stelle möchte ich kurz klarstellen, dass ich einer jener Informatikstudenten bin, die vor ihrem Studium nie wirklich programmiert hatten und ihre Freizeit auch nicht damit verbringen, Webseiten oder komplizierte Fluid-dynamik-Simulationen zu programmieren. Die Bachelorarbeit war für mich also mein erstes grosses Informatikprojekt und somit war ich auch froh darüber, dass einer der nimmermüden Betreuer einen simplen OpenGL-Viewer mit ein paar GUI-Elementen und dem Grundgerüst für den benötigten Solver zusammenschusterte. Eine derart gute Starthilfe hatte ich wirklich nicht erwartet. Von da an war es mir selbst überlassen, ob ich dieses Template erweitern und umschreiben oder ob ich es von Grund auf neu programmieren möchte.

Von Entscheidungen ...

Mein Wissensrückstand in Sachen Programmieren führte schliesslich auch dazu, dass ich einige für andere Informatiker wohl unverständliche Wege einschlug, was mir zwar bewusst,

aber herzlich egal war (und immer noch ist). Ein Beispiel gefällig: Garantiert fehlerfreie und leistungsoptimierte third-party-Library nutzen oder den Solver selbst schreiben? Natürlich selber schreiben! Nicht weil ich es besser könnte, sondern weil ich die Funktionen der Library und deren Parameter schlichtweg nicht verstand. Obwohl ich diesen Entscheid im Nachhinein durchaus komisch finde, so hat mir das selber Schreiben mehr geholfen, als ihr vielleicht denkt. Auf diese Weise hatte ich mich mit der Materie viel mehr auseinandersetzen müssen und wusste schlussendlich auch, welche Parameter gebraucht wurden und wofür. Als mich mein Betreuer nach einer Woche fragte, warum ich den Solver selber

Garantiert fehlerfreie und leistungsoptimierte third-party-Library nutzen oder den Solver selbst schreiben? Natürlich selber schreiben!

schreibe und nicht den benutze, den sie mir zur Verfügung gestellt hatten, antwortete ich genau das. Damit ich verstehe, was

dahinter vor sich geht. Von da an habe ich mich aber wieder dem Solver der Library gewidmet (insbesondere da meine Implementation extrem langsam und vielleicht ein kleines Bisschen fehlerhaft war). Und siehe da: Waren vor eine Woche viele der Funktionen noch dicke Fragezeichen für mich gewesen, so machte alles auf einmal Sinn! Insofern hat mir mein selbst geschriebenes Code-Stück jede Menge beigebracht.

... und Termitten

Dennoch funktionierte lange Zeit überhaupt nichts. Das Gleichungssystem schien zu stimmen, die Berechnung der Ableitungen ebenfalls, doch irgendwie war das Ganze noch



«verbuggt». Vermutlich nicht nur vereinzelt, sondern geradezu termitenhügelkolonienartig. Überall wuselten sie herum, und ich musste sie alle finden und vernichten. Gerade mal einen Monat nach Beginn der Arbeit reisten meine beiden Betreuer für ein Internship 3 Monate lang in die USA. Erst dachte ich, dass der fleissige Mail-Kontakt dadurch nicht leiden würde, doch scheinen solche Internships arbeitsintensiver zu sein, als ich zu Anfang noch dachte (d. h. meine Mails wurden meistens erst dann beantwortet, wenn sich das Problem für mich sowieso schon erledigt hatte). In weiser Voraussicht dessen wurde ich dann aber nicht einfach alleine zurückgelassen, sondern es wurden mir zwei neue Betreuer zugeteilt, die ebenfalls bei Disney Research arbeiteten und von grosser Hilfe waren. Zusätzlich zu den wöchentlichen Meetings gab es nun auch jeden Monat eine Videokonferenz, um die Leute in den USA über die neusten Entwicklungen zu unterrichten. Hatte ich in der ersten Konferenz noch nichts vorzuweisen (was einem schon auch ein mulmiges Gefühl beschere kann), so schaffte ich in der Zeit zum nächsten Meeting endlich den Durchbruch – der wohl grossartigste Moment in jeder Bachelorarbeit! Ich traute meinen Augen kaum, als das Programm nach einer scheinbar belanglosen Kompilation plötzlich das ausspuckte, was ich wollte. Zugegeben – die Resultate sahen noch arg fehlerbehaftet aus, doch das gewünschte Ergebnis war schon ansatzweise zu erkennen. Dies versetzte mich auch derart in Euphorie, dass ich Maus und Tastatur zur Seite legte und mir für den restlichen Tag wohlverdient frei nahm.

Ein ETH-Sommer

Als sich die erste Euphorie wieder gelegt hatte, mussten natürlich die verbliebenen Fehler

gefunden werden, was schon sehr bald zu matter Ernüchterung führte, da schon das nächste Treffen zeigte, dass es einiger nicht unwesentlicher Änderungen bedurfte (welche hinter dem Rücken gesagt meist wieder zu neuen Problemen führten). Doch musste ich mich zu diesem Zeitpunkt sowieso von der Arbeit trennen, da eine viel drängendere Angelegenheit im Anmarsch war: die Prüfungssession! Meine Befürchtung, dass ich während der Prüfungsvorbereitungen einen Grossteil des Bachelorarbeit-relevanten Wissens wieder vergessen würde, war jedoch unbegründet. Wenn das Lernen zu langweilig wurde (und das war häufig der Fall), verbrachte ich immer mal wieder einen Tag oder auch nur ein paar Stunden damit, den Code zu verbessern oder die Ableitungsformeln zum gefühlt hundertsten Mal zu korrigieren und immer noch Fehler darin zu finden. Grosse Fortschritte machte ich während dieser Zeit jedoch nicht. Schliesslich musste neben dem Lernen auch dem Sich-am-See-sonnen gebräunt werden, um die uns nachgesagte blassbleiche Hautfarbe loszuwerden und um zwischendurch auch einfach mal abschalten zu können.

Das letzte Kapitel bricht an

Nach überstandenen Prüfungen dann gleich wieder in die Vollen zu treten brauchte aber durchaus etwas Überwindung. Normalerweise wäre dies schliesslich die Zeit, in der man die verbliebenen drei Wochen geniesst und das Hirn wieder von all dem befreit, was es sich für die Prüfungen so sorgsam eingepägt hat. Doch das Treffen am Abend nach der letzten Prüfung zeigte mir schnell wieder, was es zu tun galt: Ich musste noch immer dieselben Probleme lösen wie schon vor sechzig Tagen ... und dafür blieb mir nur noch ein einziger Monat. ⌘



«Dürfen wir uns bei dir bewerben?»

Christian A. Rusche, CEO bei BSI

BSI Business Systems Integration AG
www.bsiag.com/zukunft

 **bsi**
meet your clients

An Introduction to Puppet

PASCAL SPÖRRI & MANUEL STOCKER – HAVE PLAYED AROUND WITH SERVERS A LOT

This article attempts to give you some insight into server infrastructure deployments and to give future VIS sysadmins a handy manual on how the VIS infrastructure works. Additionally, it gives you insights on how to choose future management systems.

Why Configuration Management?

Usually, every deployment of “computing resources” starts off small: you probably have a few servers that all take care of their specific functions, say a mail server, a database machine, and a web application host. As soon as you start getting bigger, you need to be able to rapidly deploy new machines and keep their configuration (which includes installed software packages!) in sync. And this is probably the first time you start to think about needing something that takes over the boring tasks of managing these aspects.

But configuration management is not only limited in its usefulness in deployments of lots of machines with similar settings. Even if you just have a few machines, all with different setups, keeping their configuration in something like Puppet lets you rapidly replace these machines in the event of a failure: just hook up the new machine and let the configuration management do the rest.

What is puppet?

Puppet is a tool that can manage configurations on Linux, OS X, and Windows systems. A configuration is defined using a declarative and model language; meaning that you describe

the state which you want your system to be in, instead of scripting the actions that have to be run. Typically, the configuration is stored on a central host called the Puppet master. The configuration is parsed by the Puppet master, and simple actions are then sent to the client respecting their dependencies. As an example, if you define that you want to have Apache installed and the Apache service running, latter obviously depends on the Apache package being installed on the system.

The clients themselves authenticate via a public key infrastructure providing mutual authentication. Thus communication between Puppet master and client is both encrypted as well as authenticated. The client downloads its configuration and optionally additional files from the Puppet master, and adjusts the state of the system to match the configuration.

The installation on Debian-based systems is fairly simple. On the Puppet master host, install the Puppet master package with `apt-get install puppetmaster`, and on clients with `apt-get install puppet`. The client find the Puppet master by looking for `puppet.domain.tld`.

Configuration

Configuration starts in `/etc/puppet/manifests/site.pp` and can be spread out to more files. You define the configuration each host has. A basic example could look like this:

```
class base_config {
  file {'/etc/puppet/puppet.conf':
    source => 'puppet:///files/puppet.conf'
  }
  package {'fail2ban':
    ensure => present,
  }
}

class server_basic inherits base_config {
  package ['less', 'screen', 'vim-nox']:
    ensure => present,
  }
}

node 'www.vis.ethz.ch' inherits server_basic {
  include apache # Install apache as well
}
```

We define a base class to be used for all nodes. It fetches a file from the Puppet master and installs fail2ban. For servers we have some additional packages we'd like to install, which can be achieved by defining a new class that inherits the basic class. Furthermore, we define a node by its hostname inheriting the server config as well as some additional things, in this case installing Apache since it's the webserver.

The definitions we see in Puppet are classes, defined types, and resources. Classes are named blocks of Puppet code, typically stored in modules (more on this later), which contain Puppet resource definitions. Defined types are like classes, but can be declared multiple times. Puppet resource definitions are the basic build-

ing blocks of Puppet. A definition, such as "file" or "package", defines the state the machine should be in.

The client periodically gets the config from the Puppet master and tries to adjust its actual state to reflect the desired state. In the example above, the machine www.vis.ethz.ch will at some point:

- Download the configuration file stored in `puppet:///files/puppet.conf` to `/etc/puppet/puppet.conf`.
- Install several packages on the server
- And do something with Apache (whatever is specified in the included definition). →

| Example | Description | In JavaScript |
|--|---|--|
| <code>\$foo = true</code> | Variable declaration with initializer | <code>var foo = true;</code> |
| <code>\$bar = "Hello \${bar}"</code> | String with interpolation | <code>var bar = "Hello " + foo;</code> |
| <code>\$bar = 'Hello \${bar}'</code> | String without interpolation | <code>var bar = "Hello \${bar}";</code> |
| <code>\$bar = foo</code> | Single word string | <code>var bar = "foo";</code> |
| <code># awesome comment</code> | Comment | <code>// awesome comment</code> |
| <code>\$a = [1, 2, 3, 4]</code> | Array declaration with initializer | <code>var a = [1, 2, 3, 4];</code> |
| <code>\$b = \$a[0]</code> | Array indexing | <code>var b = a[0];</code> |
| <code>\$h = {"foo" => \$a}</code> <code>\$h = {foo => \$a}</code> | Hash, also called dictionary in other languages | <code>var h = {"foo": a};</code> <code>var h = {foo: a};</code> |
| <code>\$h["foo"][2]</code> | Dereferencing | <code>h["foo"][2]</code> or <code>h.foo[2]</code> |

Language Primer

Refer to table above to get a quick overview of Puppet's configuration language. So that you can relate to something you might know already, we list the JavaScript equivalent for your convenience.

Conditionals

Puppet supports if/then/else similar to what you see in other languages^[A] which can be written alternatively as a case block (also called select/switch in other languages)^[B].

The selector statement lets you do this inline (like the ternary `?:` operator other languages know)^[C].

Conditionals are influenced by Ruby or Perl in that they can directly use regular expressions^[D].

```
if $operatingsystem == "Debian" {
  # ...
} elsif $operatingsystem == "Ubuntu" {
  # ...
} else {
  # We are somewhere else, maybe on Windows :)
}
```

```
case $operatingsystem {
  'Debian': { # do something here }
  'Ubuntu': { # do something else }
  /^(Redhat/Centos)$/: { # uuuh a YUM based OS
  default: { # default action }
}
```

[C]

```
$rootgrp = $::osfamily ? {
  'Solaris' => 'wheel',
  /(Darwin/FreeBSD)/ => 'wheel',
  default => 'root'
}
```

```
if $host =~ /^www(\d+)\./
{
  # ...
}
```

Resource Types

Resource types are by far the most important feature of Puppet. Resource types are provided by Puppet itself and most of them are platform agnostic. Every resource has a **type**, a **title**, and a set of **attributes**^[E]. If the "name" attribute is not declared, it will get the value of the "title" attribute^[F]. The title is an identification string and **must be unique** per resource type. The most useful resource types are:

- **Package**, which installs a package on the system:

```
package { "apache":
  ensure => present,
}
```

It's also possible to install multiple packages with a single declaration:

```
package { ["nginx",
  "mysql-server", "python3"]:
  ensure => present,
}
```

or you can give a package a special name:

```
package { "Install webserver":
  name    => "apache",
  ensure => present,
}
```

- **File**, which will deploy a file on the server:

```
file { "/etc/hosts":
  source => "puppet:///
  modules/hosts/hosts.conf"
}
```

```
resourcetype { "title":
  attribute0 => value
  attribute1 => $bar
}
```

[E]

```
resourcetype { "A title":
  name => "value",
  attribute0 => value
  attribute1 => $bar
}
```

[F]

using templates:

```
file { "/etc/nginx/nginx.conf":
  content => template('nginx/
  nginx.conf.erb')
}
```

- **Service**, manages a system service:

```
service { "nginx":
  ensure => running,
}
```

Services can be notified to restart upon notification changes. Say we have a file declaration here:

```
file { "Nginx Configuration":
  name => "/etc/nginx/nginx.conf":
  content => template('nginx/
  nginx.conf.erb')
  notify => Service['nginx']
}
```

then puppet will make sure that nginx is restarted as soon as the nginx configuration changes.



- **Exec**, which executes a command on a the client machine:

```
exec { "logrotate":
  path    => ['/usr/bin', '/usr/sbin', '/bin'],
  onlyif => "test `du /var/log/messages | cut -f1` -gt 100000"
}
```

which implements a poor man's logrotate. Please note that it's generally a bad idea to depend on Puppet to execute regular jobs; use cron instead, e.g. via Puppet's native cron support.

- **Notify**, which sends an arbitrary message to the agent run-time log:

```
notify { "Log to syslog": }
```

For additional types and configuration please refer to the Puppet type reference^[1].

It's also possible to create custom types directly in Puppet. Before we can introduce these types, let us first introduce the creation of classes.

Classes

Classes are named blocks of Puppet code. To make reuse easier, they are usually stored in separate modules (covered later).

In contrast to resource types, they can only be instantiated once. A class declaration looks like [G]. To instantiate a class, one can either use the include keyword:

```
include apache
```

use the class declaration:

```
class { 'apache':
  config_file => "/etc/apache2/
  apache2.conf",
}
```

or, alternatively, use the require keyword.

```
# Because of feature X we need
# to include apache here
require apache
```

In contrast to the class declaration, the require and include keywords can be used multiple times. The require keyword will, however, create a dependency on said resource on the DAG. This can introduce some problems in the resolution process. The general recommendation is: Don't use require. We will introduce alternatives later on.

Please note that include and require can be mixed with the class declaration. The output of this action is however undefined. As a general rule, only use include/require when there is no explicit resource ordering needed or when the classes do not provide any parameters.

[G]

```

class apache($config_file = "/etc/httpd/httpd.conf")
{
  # Puppet code
  package { "apache":
    ensure => installed,
  }
  class { "apache::config":
    config_file => $config_file
  }
}

```

Defined Types

Defined types are custom defined resource types that can be declared multiple times. These types are especially useful for creating databases, Apache virtual hosts, DNS entries, etc.

They can be defined in the following way:

```

define apache::vhost(
  $port,
  $www_root,
  $template = 'apache/vhost.conf.erb',
  ...
) {
  file { "/etc/apache/sites-available/${name}.conf":
    template => content($template),
    require => Class['apache']
  }
  file {$www_root:
    ensure => directory,
    mode => 755,
    owner => "www-data",
    group => "www-data",
  }
  # Automate backup etc here..
}

```

and can be instantiated in the following way:

```

apache::vhost { 'www.vis.ch':
  ...
}
apache::vhost { 'www.vis.ethz.ch':
  ...
}

```



Modules

Modules are a collection of classes, templates, defined types, and files. Typically, these modules are stored in `/etc/puppet/modules/<nameofth module>`. These modules then define a scope with the name of the module. The manifests folder in the module contains the Puppet classes and defined types. Puppet will first look for the file `init.pp`, which stores the module base class. We have reproduced such a module right beside.

The files stored in the files folder can be accessed by directly referencing the module:

```
/etc/puppet/modules/apache
  /manifests
    /init.pp
      # Contains the base class
      class apache { ...
        include apache::config
      }
    /config.pp
      class apache::config {...}
    /vhost.pp
      define apache::vhost {...}
  /files
    /static.conf
  /templates
    /apache.conf.erb
    /vhost.conf.erb
```

```
file { ["/etc/apache/conf.d/static.conf":
  source => "puppet:///modules/apache/static.conf"
}
```

which uses the Puppet fileserver to distribute the files. Templates can be accessed in the following way:

```
file { ["/etc/apache/apache.conf":
  content => template('apache/apache.conf')
}
```

Puppet templates are fairly easy to use. Due to space constraints, we however have to reference to the excellent Puppet documentation^[2]. Since Puppet is widely used, Puppet modules can be found all over the web. A good starting place is puppetforge.com or github.com. A module from puppetforge can easily be installed with `puppet module install puppetlabs-apache`.

We generally recommend to look at and test the Puppet code before using it in a production environment.

Tips

- The Puppet documentation is a good starting point to get started on Puppet.^[3]
- We recommend the use of Puppet 3.x over 2.7
- It's also possible to use Puppet on Windows.^[4]
- Future version of Puppet may or may not include loops or map/reduce functions.^[5]

Alternatives to Puppet

As you might expect, there exist quite a few opinions on how configuration management should be done. As such, there are other tools with other philosophies or technology stacks that might suit your needs better than Puppet. To name a few examples:

- Salt^[6]: Written in Python and following a more master-centric workflow. Minions have to be brought to the newest state explicitly.
- Opscode Chef^[7]
- CFengine^[8]



Links

- [1] <http://docs.puppetlabs.com/references/latest/type.html>
- [2] <http://docs.puppetlabs.com/learning/templates.html>
- [3] <http://docs.puppetlabs.com/>
- [4] <http://puppetlabs.com/presentations/managing-windows-puppet>
- [5] http://docs.puppetlabs.com/puppet/3/reference/lang_experimental_3_2.html
- [6] <http://saltstack.org>
- [7] <http://www.opscode.com/chef/>
- [8] <http://www.opscode.com/chef/>

ANZEIGE

Ich
erzeuge
Energie.



Von Turbinenhalle bis Proberaum: Als Mitarbeitende/r der BKW-Gruppe fließt Ihre Energie an vielen Orten. Und mit klimafreundlichem Strom aus Wasser, Wind, Sonne und Kernkraft lassen Sie täglich mehr als eine Million Menschen daran teilhaben – unterstützt von 3'000 kompetenten Kolleginnen und Kollegen.

Die BKW-Gruppe entwickelt und realisiert die Energieinfrastruktur von heute und morgen. Realisieren Sie bisher ungenutztes Energiepotenzial und steigern Sie die Effizienz unserer Anlagen, zum Beispiel bei unseren Projekten in der Wasserkraft. Für Ingenieurinnen und Ingenieure gibt es bei uns viel zu tun! Bewerben Sie sich jetzt – Informationen und aktuelle Stellenangebote finden Sie auf der zentralen Stellenbörse unserer Webseite:

Spielen, Sterben, Savegame laden

MATTHIAS GEEL – IST SCHON EIN PAAR MAL GESTORBEN

Unsere Alltagserfahrung lehrt uns, dass der Tod etwas Permanentes ist, etwas Endgültiges, von dem es kein Zurück mehr gibt. Deswegen auch der etwas zynische Spruch: «Man stirbt nur einmal.» In den virtuellen Welten der (heutigen) digitalen Spiele wird dieses Konzept jedoch meistens durch eine beliebig wiederholbare Schleife aus *Spielen, Sterben, Speicherpunkt laden* ersetzt. Durch jederzeitiges Speichern oder grosszügig verteilte Speicherpunkte hat der virtuelle Tod schon seit längerem seinen Schrecken verloren. Doch wie kam es überhaupt dazu?

«You Died», «Wasted», «You are Dead». Der digitale Tod ist auch in modernen Spielen allgegenwärtig und hat viele Gesichter. Doch nur in den wenigsten Fällen hat er noch wirklich ernsthafte Konsequenzen. Meistens ist der Tod innerhalb weniger Sekunden (bei Konsolen häufig etwas länger) überwunden, und unser virtuelles Alter Ego erscheint quietschlebig beim letzten Speicherpunkt oder wird aus einem (fiktiven) Krankenhaus entlassen. Das klassische «Game Over» kennen viele jüngere Spieler nur noch vom Hörensagen.

Dies war jedoch nicht immer so. Wir schreiben das Jahr 1978, und *Space Invaders* tritt gerade seinen Siegeszug durch die japanischen Spielhallen an. Das Spielprinzip ist so einfach wie genial. Der Spieler steuert eine Laserkanone am unteren Rand des Bildschirms und versucht die sich von oben nach unten bewegenden Alienreihen abzuschiessen, bevor diese den Spieler erreichen oder die Kanone abschiessen können. *Space Invaders* war das erste Videospiel, welches dem Spieler mehrere *Leben* gewährte (nämlich 3). Sobald alle Leben aufgebraucht wurden, war das Spiel vorbei (*Game Over*) und

man musste von Neuem beginnen. Dieses Konzept der begrenzten Leben fand sich daraufhin in zahlreichen anderen Spielen zur damaligen Zeit wieder, so auch im zwei Jahre später erschienen, ebenfalls extrem erfolgreichen *Pac Man*.

Dabei darf jedoch nicht vergessen werden, dass Arcade-Spiele vor allem aus finanziellen Gründen auf Spielkonzepte setzten, die den Automatenbesitzern einen konstanten Fluss von Münzeinwürfen garantierten. Ebenfalls nicht ganz ohne monetäre Hintergedanken war es dann bei späteren Arcade-Spielen durchaus üblich, dass man durch den Einwurf einer weiteren Münze das digitale Ableben noch etwas hinauszögern konnte. *Continue?* lässt grüssen.

Der Aufstieg des Savegame

Mit dem Einzug der Heimcomputern in die privaten Haushalte Anfangs der 80er-Jahre fiel dieser finanzielle Druck jedoch weg. Zusätzlich sorgte die rasante technische Entwicklung dieser Systeme dafür, dass Computerspiele nicht nur komplexer, sondern auch immer umfangreicher wurden. So ist es nicht verwunderlich, dass



Diablo III - Hardcore Modè

bereits Spiele auf Floppy-Disk eine Speicherfunktion besaßen.

Etwas anders sah die Situation bei den Videospielekonsolen aus. Gerade den jüngeren Semestern unter uns dürfte kaum noch bewusst sein, dass die meisten Konsolen- und Handheld-Spiele der 1980er-Jahre keine Speicherfunktion besaßen. Dies einerseits, weil viele der früheren Spiele noch Portierungen von Arcade-Spielen waren, und andererseits fehlte den meisten Konsolen zu dieser Zeit schlichtweg ein beschreibbares Speichermedium. So mussten z. B. die früheren Mario-Titel wie *Super Mario Bros.* auf dem NES noch an einem Stück durchgespielt werden. Erst mit *The Legend of Zelda* (NES) wurde 1987 zum ersten Mal ein Konsolenspiel mit einer Speicherfunktion veröffentlicht. Technisch wurde dies mit einem im Steckmodul fix eingebauten RAM-Chip erreicht, welcher durch eine interne Batterie mit Strom versorgt wurde.^[1] Von da an waren Speicherstände auch bei Konsolenspielen immer häufiger anzutreffen, und spätestens mit dem Erscheinen der PlayStation 1995 etablierten sich auch externe Speichermedien wie Memory Cards.

Durch die unterschiedliche Ausrichtung der beiden Plattformen (Konsolen und PCs) fand jedoch eine gegensätzliche Entwicklung in Bezug auf Savegames statt. Während bei den häufig

Level-basierten Konsolenspielen sich die Methode der festgelegten Speicherpunkte allmählich durchsetzte (z. B. zu Beginn eines Levels oder an in der Spielwelt platzierten Speicherstationen), wurde es bei PC-Spielen bereits sehr früh zu einer Selbstverständlichkeit, jederzeit und überall abspeichern zu können. Das hatte natürlich ganz unterschiedliche Spielweisen zur Folge und wirkte sich erheblich auf das jeweilige Spielerlebnis aus. So konnte auf dem PC durch gewissenhaftes, regelmässiges Abspeichern des aktuellen Spielstandes in Strategie- oder Rollenspielen die Konsequenzen des digitalen Scheiterns auf ein Minimum reduziert werden. Den (vorläufigen) Höhepunkt dieser Entwicklung wurde mit der Einführung der Quicksave-Funktion erreicht. Dabei konnte mittels spezifischer Tasten (meistens den F-Tasten) das Spiel ohne Unterbruch zu jedem x-beliebigen Zeitpunkt gespeichert und mit einer anderen Taste wieder geladen werden. Gerade bei handlungsorientierten Ego-Shootern wie *Half-Life* war diese Methode auf dem PC sehr verbreitet. Die digitale Wiederauferstehung war also nur noch einen Tastendruck entfernt.

Heutzutage sind jedoch viele Spieledesigner wieder davon abgekommen (sehr zum Missfallen zahlreicher PC-Spieler) und setzen stattdessen vermehrt auf *Autosaves*. Autosaves sind



festgelegte Speicherpunkte, die z. B. beim Betreten eines bestimmten Gebietes automatisch ausgelöst werden. Dadurch wird dem Spieler faktisch die Kontrolle entzogen, unter welchen Bedingungen er wieder auferstehen kann. Die Diskussion darüber, welche Speichermethode dem Spielspass am zuträglichsten ist, könnte noch endlos fortgeführt werden und wurde auch bereits von zahlreichen Spielmagazinen behandelt.

Alternative Mechanismen

Stattdessen möchten wir uns nun mit Spielen auseinandersetzen, die bewusst auf andere Techniken und Spielmechanismen setzen, um dem digitalen Tod (wieder) mehr Tragweite zu geben und diese ewige Schleife aus Tod und Wiederauferstehung zu durchbrechen.

Einer solcher als *Permadeath* bekanntgewordener Spielmechanismus findet sich vor allem in Games mit Rollenspiel-ähnlichen Elementen wieder. Nicht unbedingt der erste, aber sicher einer der bekanntesten Vertreter dieses Genres ist *Diablo II* (und später auch *Diablo III*). Und zwar lassen sich in diesen Spielen Charaktere erstellen, die nur ein Leben besitzen, genauso wie im richtigen Leben (no pun intended). Stirbt der Charakter, so geht auch der komplette Spielfortschritt verloren. Dieser sogenannte *Hardcore-Modus* findet man auch in anderen Spielen wie z. B. *Torchlight*, *Minecraft* und *Terraria*. Bei Triple-A-Spielen ist solch ein Modus aber nach wie vor eher selten, auch wenn es immer wieder Ausnahmen gibt wie das erst im Oktober dieses Jahres erschienene *Batman: Arkham Origins*.

Eine etwas abgeschwächte Variante von *Permadeath* erlaubt zumindest das vereinzelte Speichern des Spielfortschrittes. In *Dead Space 2*, einem Survival-Horror-Shooter, darf der Spie-

ler im *Hardcore-Modus* genau dreimal speichern und wird bei einem eventuellen Ableben an den letzten Speicherpunkt zurückgesetzt.

Was die bisher vorgestellten Spiele jedoch gemeinsam haben ist, dass der permanente Tod lediglich ein optionales Feature ist und somit einfach einen höheren Schwierigkeitsgrad darstellt. Jedoch gibt es auch Spiele, bei denen dieser Mechanismus von Anfang an in das Spieldesign einbezogen wurde. Im letzten Jahr veröffentlichten und über Kickstarter finanzierten Indie-Titel *FTL: Faster Than Light*, einem top-down-Strategiespiel, bei welchem man eine Raumschiffcrew steuert, ist der *Permadeath* fester Bestandteil des Spielprinzips. Stirbt die Crew oder wird das Raumschiff zerstört, muss das Spiel von Neuem gestartet werden.

Noch radikalere Spielmechanismen finden sich in zwei kostenlosen Spielen, die man aber eher als Experimente verstehen darf: *One Chance* und *One Single Life*. *One Chance* ist ein kleines Browserspiel, in welchem man genau eine



FTL: Faster Than Light

Chance kriegt, um die Menschheit innerhalb von 6 Tagen vor einer globalen Katastrophe zu retten. Das ebenso tragische wie interessante Spiel ist als interaktive Geschichte aufgebaut, wobei jede Entscheidung endgültig und unwiderruflich ist. Das für iOS veröffentlichte *One*

Single Life basiert zwar auf einem anderen Spielprinzip, kann jedoch auch nur einmal gespielt werden. Am Ende wird in einer Statistik angezeigt, wie man sich im Verhältnis zu allen Spielern geschlagen hat.

Faszination und Frust

Es gibt einen guten Grund, warum vieler dieser Mechanismen in Mainstream-Spielen nicht zu finden sind: Spiele sind zu einem Unterhaltungsmedium für die Masse geworden. So geht es vielen Gelegenheitsspielern vor allem darum, gut unterhalten zu werden, eine Geschichte zu erleben oder gar sich einfach zu entspannen. Der sportliche Aspekt des Spielens und die damit verbundene Herausforderung, die ein Spiel bieten muss, geraten dadurch in den Hintergrund. Hinzu kommt, dass viele Spiele natürlich eine möglichst grosse Anzahl an potentiellen Käufern ansprechen wollen. Insbesondere die beliebte Zielgruppe der «Casual-Gamer» rückt immer stärker in den Fokus. Dies hat naturgemäss zur Folge, dass Spiele derart gestaltet werden, dass die Frustrationstoleranz der breiten Masse ja nicht überschritten wird.

Denn genau diese wird bei den hier diskutierten Spielen ganz schön strapaziert. Und zwar in einer Weise, wie dies bei den ersten Videospiele in dieser Form gar nicht möglich war. Denn auch wenn man in diesen frühen Spielen bereits seinen eigenen Avatar steuerte, so fand dabei keine wirkliche Charakterentwicklung statt. Dadurch hielt sich die emotionale Bindung mit der Spielfigur in Grenzen, und ein Verlust derselbigen wurde auch nicht wirklich als tragisch erachtet. Im Gegensatz dazu tut es wohl jedem Spielerherzen weh, wenn man seine nach einiger Zeit liebgewonnene Raumschiffcrew (in FTL) oder seinen während Tagen hochgele-

velten Paladin (in Diablo II) bloss wegen einer kleinen Unachtsamkeit oder (noch tragischer) Selbstüberschätzung unwiderruflich an das digitale Nirwana verliert. Noch schlimmer (und fast schon ironisch) sind externe, *reale* Faktoren, die einen *virtuellen* Tod herbeiführen können, wie z. B. ein Einbruch der Framerate, Stromunterbruch oder dem simplen «Ausrutschen» mit der Maus.

Dem gegenüber steht ein viel intensiveres Spielerlebnis. Entscheidungen müssen im Voraus geplant und wohl überlegt sein. Auch die Befriedigung, ein Spiel «besiegt» zu haben, ist um ein Vielfaches grösser, und das Gefühl, dem virtuellen Tod die Stirn geboten zu haben, entschädigt für so manche zuvor erlebte Niederlage. Zusätzlich erzwingen solche Spielmodi häufig ein Umdenken in der gewohnten Spielweise. Es kann eben nicht mehr blindlings um jede Ecke gerannt werden.

Die Permadeath-Modi sind gewiss nicht jedermanns Sache. Dennoch möchte der Autor den geschätzten Leser ermutigen, für einmal seine Komfortzone zu verlassen und beim nächsten Spiel auf die gewohnten «Sicherheitsnetze» zu verzichten. Wer weiss, welche neue Spielerfahrung er oder sie dadurch machen wird. ⌘

Anmerkung der Redaktion

[1] Auch einige Gameboy-Module speichern die Spielstände so, etwa Pokémon Silber. Sobald die Batterie leer ist, sind das auch die Speicherstände. Die Batterie kann ersetzt werden, wer keinen passenden Schraubenzieher für das Modul hat, schmilzt sich einen selber aus einem Stabilo.



Notenstatistik

GREGOR WEGBERG – DANKT DEN ENTWICKLERN DER «XL TOOLBOX»

Look at the graphs, graphs are important!

Basisprüfung

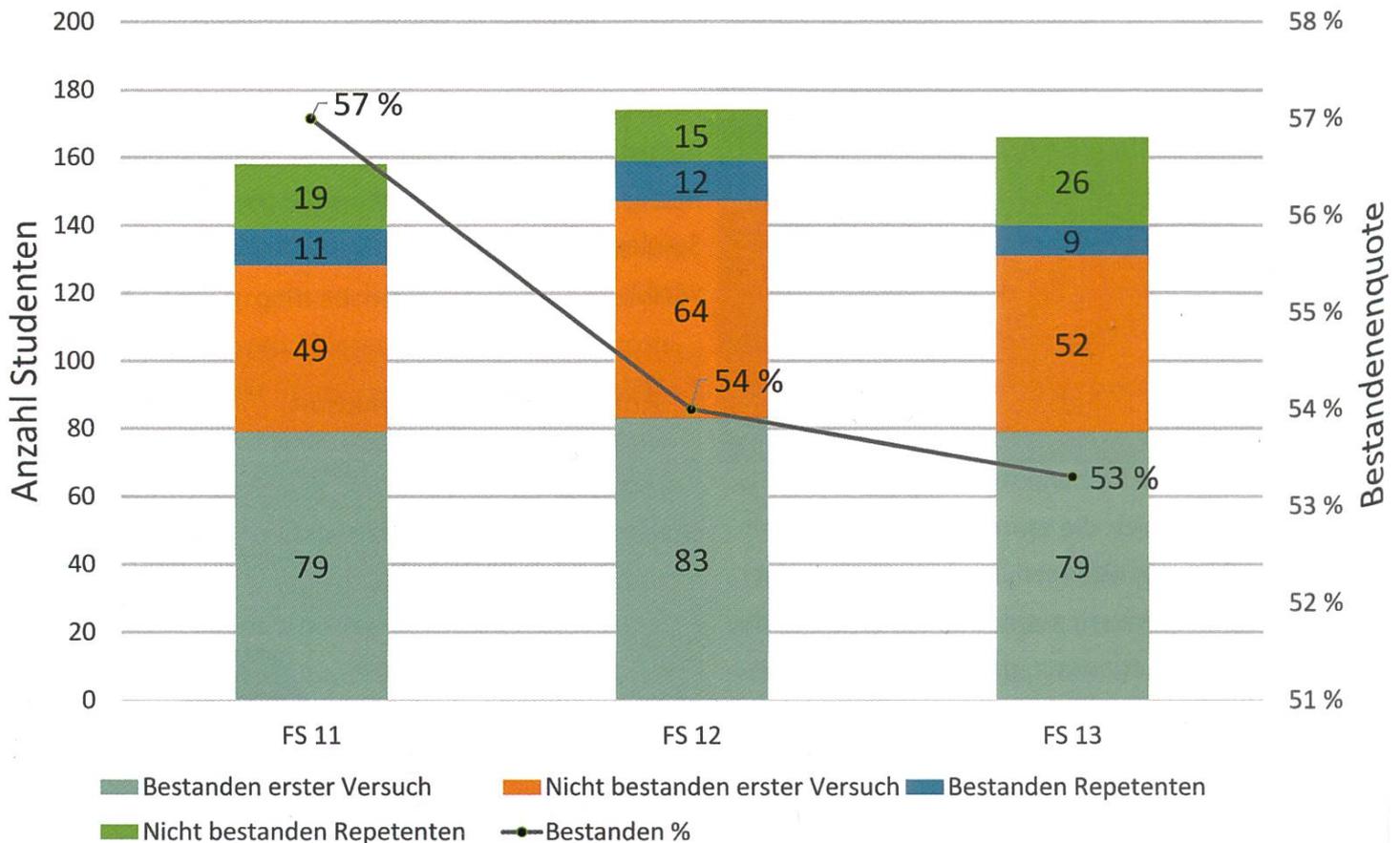
Die Durchschnittsnote der Basisprüfung legte beinahe eine Punktlandung hin. Gerade einmal um -0.01 liegt sie unter dem letztjährigen Durchschnitt.

Letzten Sommer konnte sich vor allem «Datenstrukturen & Algorithmen» massiv verbessern mit einem Plus von 0.51 und ist wieder im Bereich von FS 11 und FS 10. Es ist auch schön zu sehen, dass Analysis I & II sich nicht weiter verschlechtert und im Vergleich zum Vorjahr ein Plus von 0.11 hinlegt.

Dagegen hat sich Physik wieder stark (-0.63) verschlechtert, was mit dem Professorenwechsel schnell erklärt ist (vergleiche FS 10 mit FS 11).

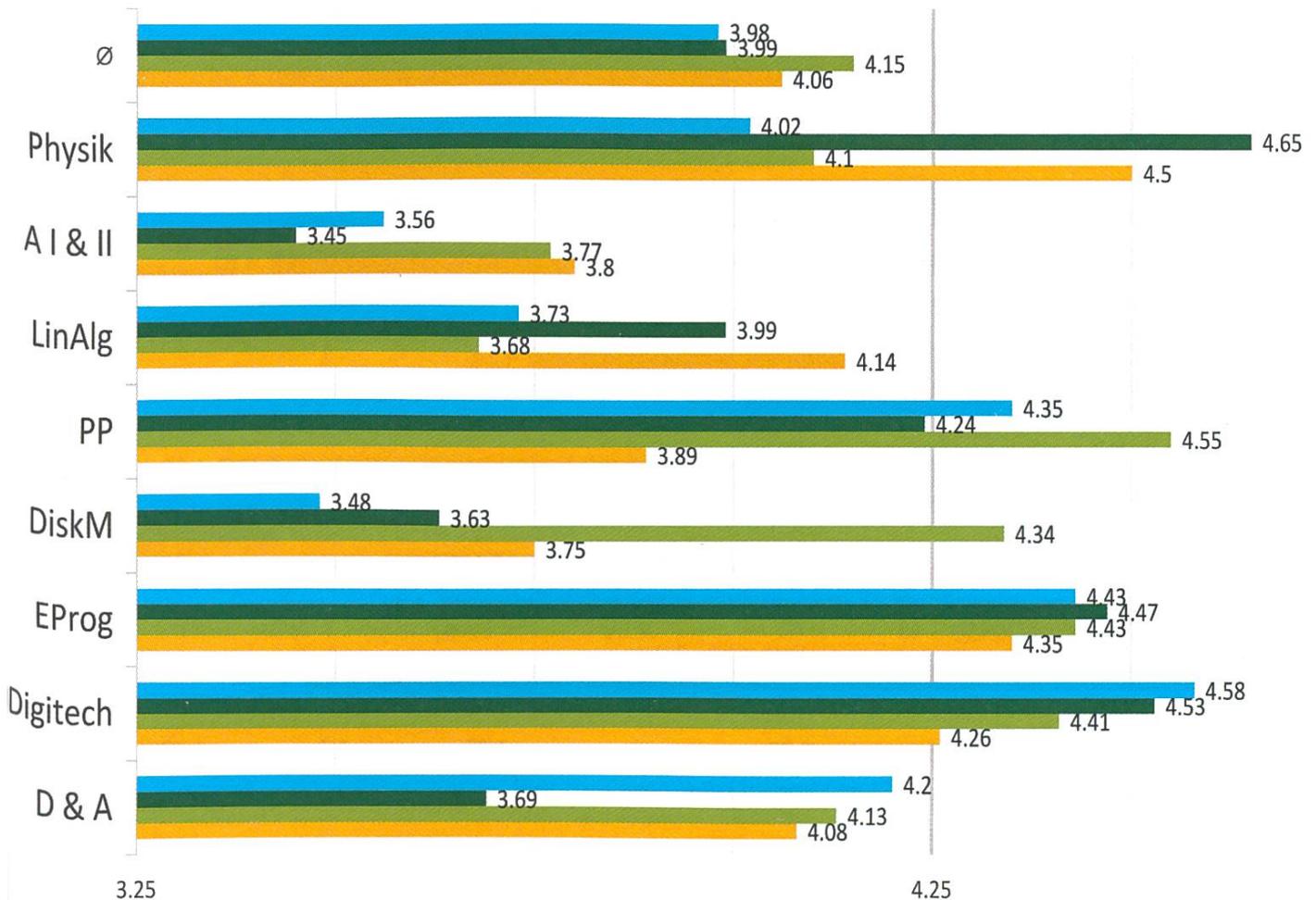
Vergleicht man die letzten drei ordentlichen Basisprüfungen (FS 11, FS 12 und FS 13), so zeichnet sich bezüglich der Bestanden-Quote kein besonders gutes Bild ab. Von 57 % fällt es auf heutige 53.3 %. Interessanterweise bleibt die absolute Anzahl der Studenten mit erfolgreicher Basisprüfung stabil.

Anzahl Studenten an der Basisprüfung



Notenschnitt Basisprüfung

■ FS 13 ■ FS 12 ■ FS 11 ■ FS 10



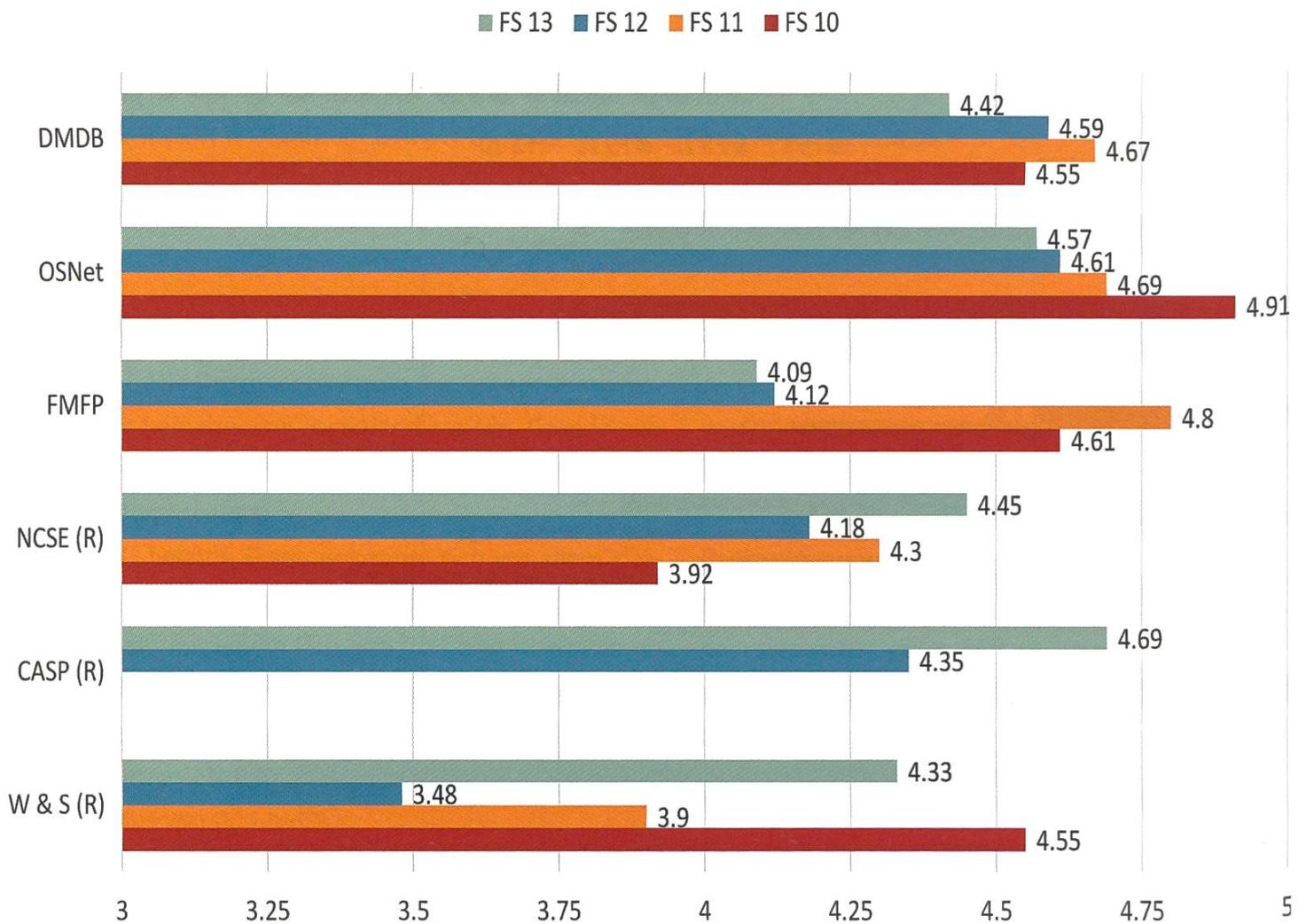
Obligatorische Fächer

«Operating Systems and Networks» kann den Abwärtstrend bisher nicht brechen und landet im FS 13 bei einer 4.57. Besonders positiv fällt die Wiederholungsprüfung von «Wahrscheinlichkeit und Statistik» auf, die von einer 3.48 auf eine 4.33 springt und sich so wieder dem Stand von FS 10 annähert. Auch «Systems Programming and Computer Architecture» macht eine gute Figur mit einem Plus von 0.34 bei der Repetitionsprüfung (R).

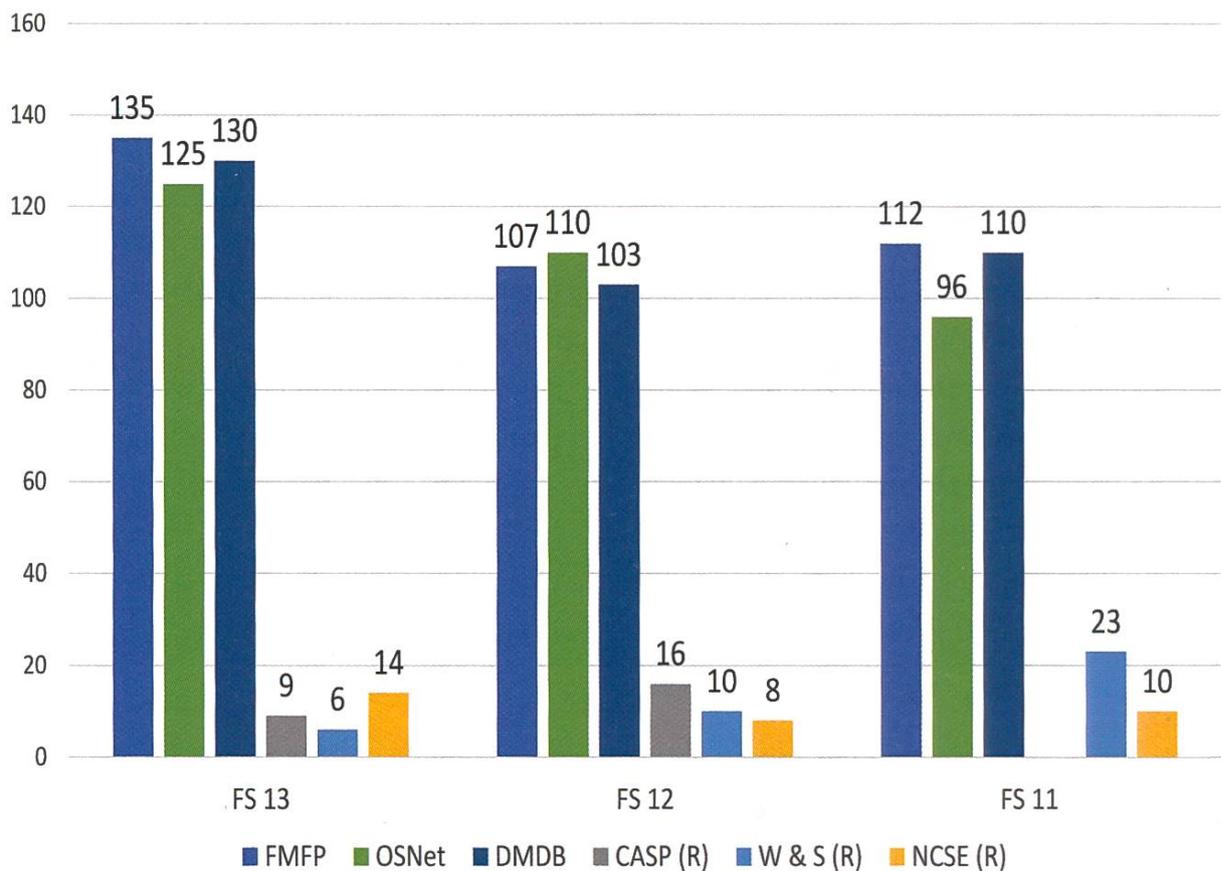
Betrachtet man die Teilnehmerzahlen der einzelnen Prüfungen, so ist «Wahrscheinlichkeit und Statistik» in einem Abwärtstrend, was natürlich sehr zu begrüßen ist bei einer Repetitionsprüfung.

Bei der Bestanden-Quote macht vor allem «Formal Methods and Functional Programming» gar keine gute Figur. Nach einem Topresultat im FS 11 (91.9 %) liegt es heute bei nur noch 66 %. Dies ist zwar eine Verbesserung zu FS 12, aber immer noch weit weg vom FS-11-Resultat. →

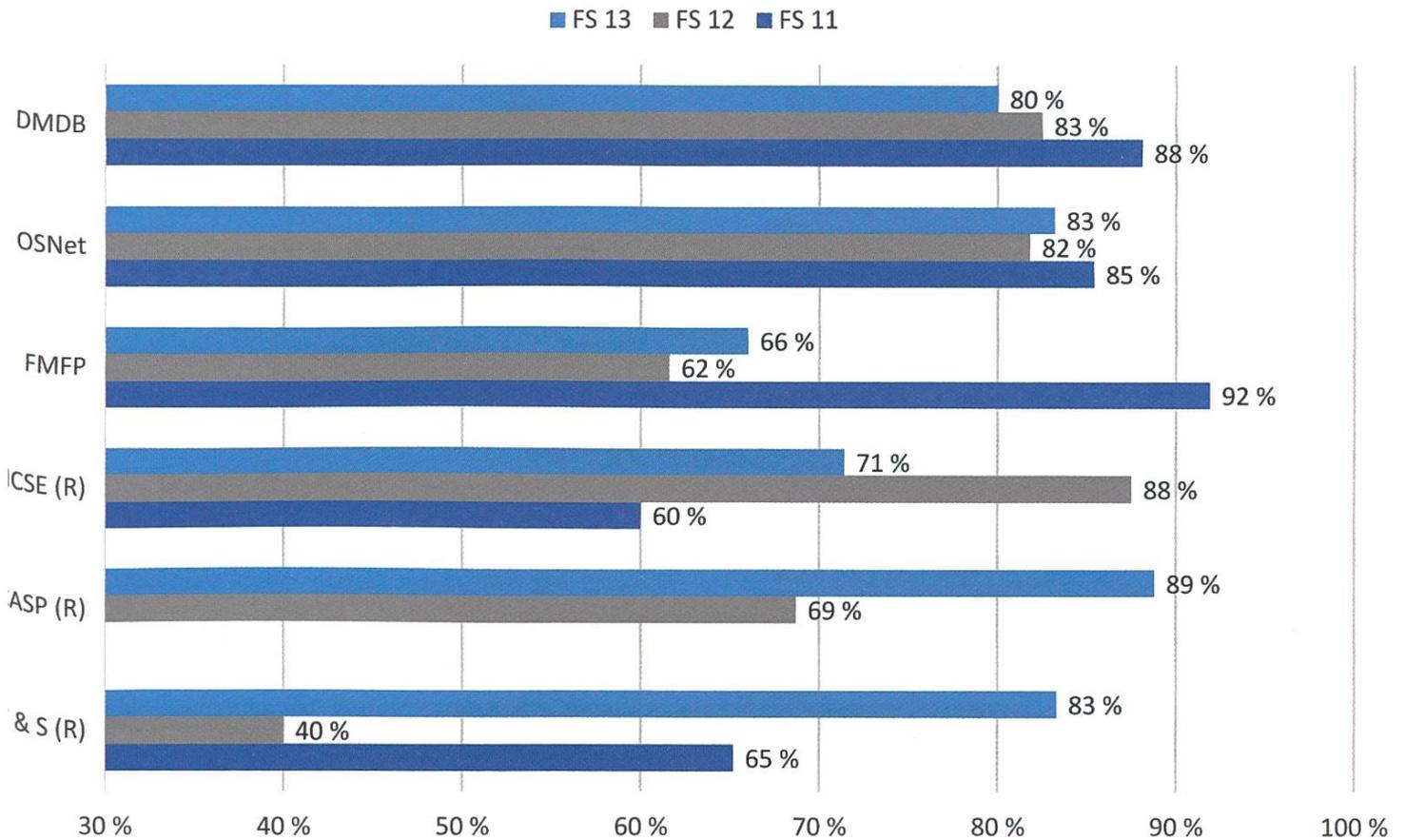
Obligatorische Fächer FS



Teilnehmer Obligatorische Fächer



Bestandenquote Obligatorische Fächer

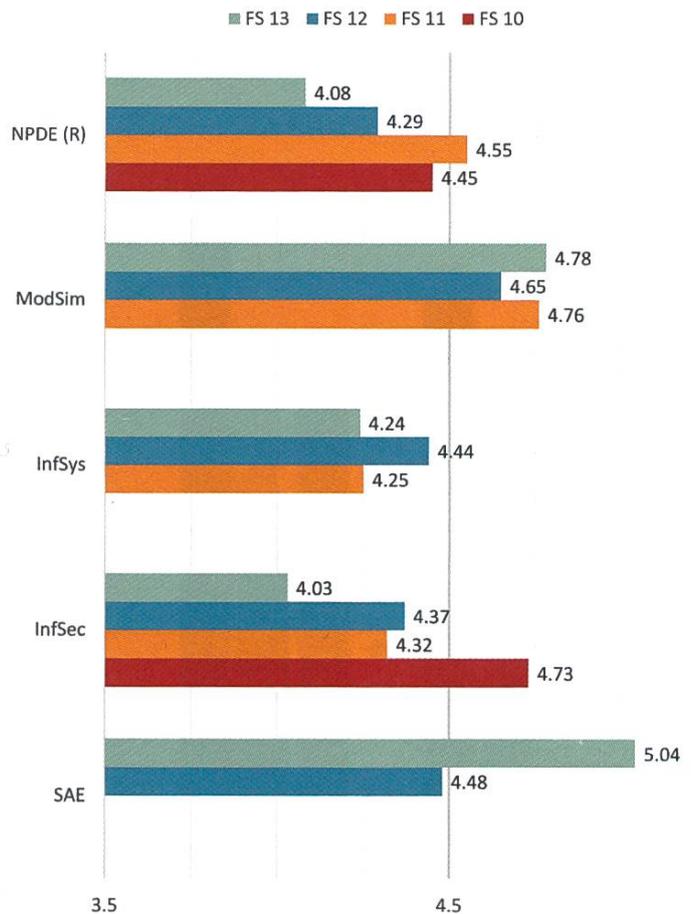


Obligatorische Fächer der Vertiefung

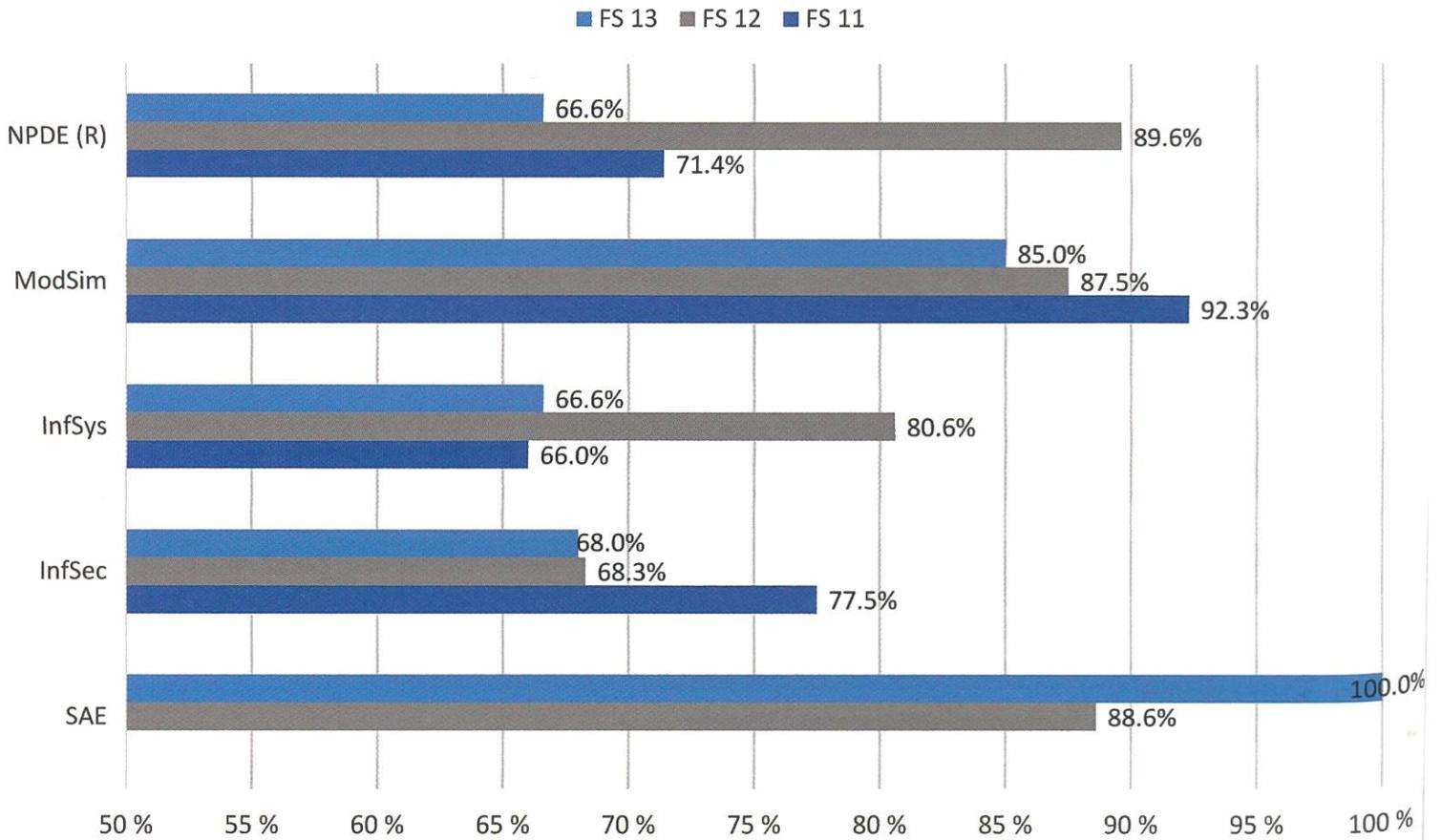
Dieses Semester fällt im Besonderen «Information Security» mit einem Minus von 0.34 auf. Gleichzeitig erlebt das Fach einen wahren Sprung in der Zahl der Studenten.

Auf der positiven Seite steht «Software Architecture and Engineering» mit einem Plus von 0.56 und einer Bestanden-Quote von 100 %, was wir bisher nur von «Compiler Design» kennen.

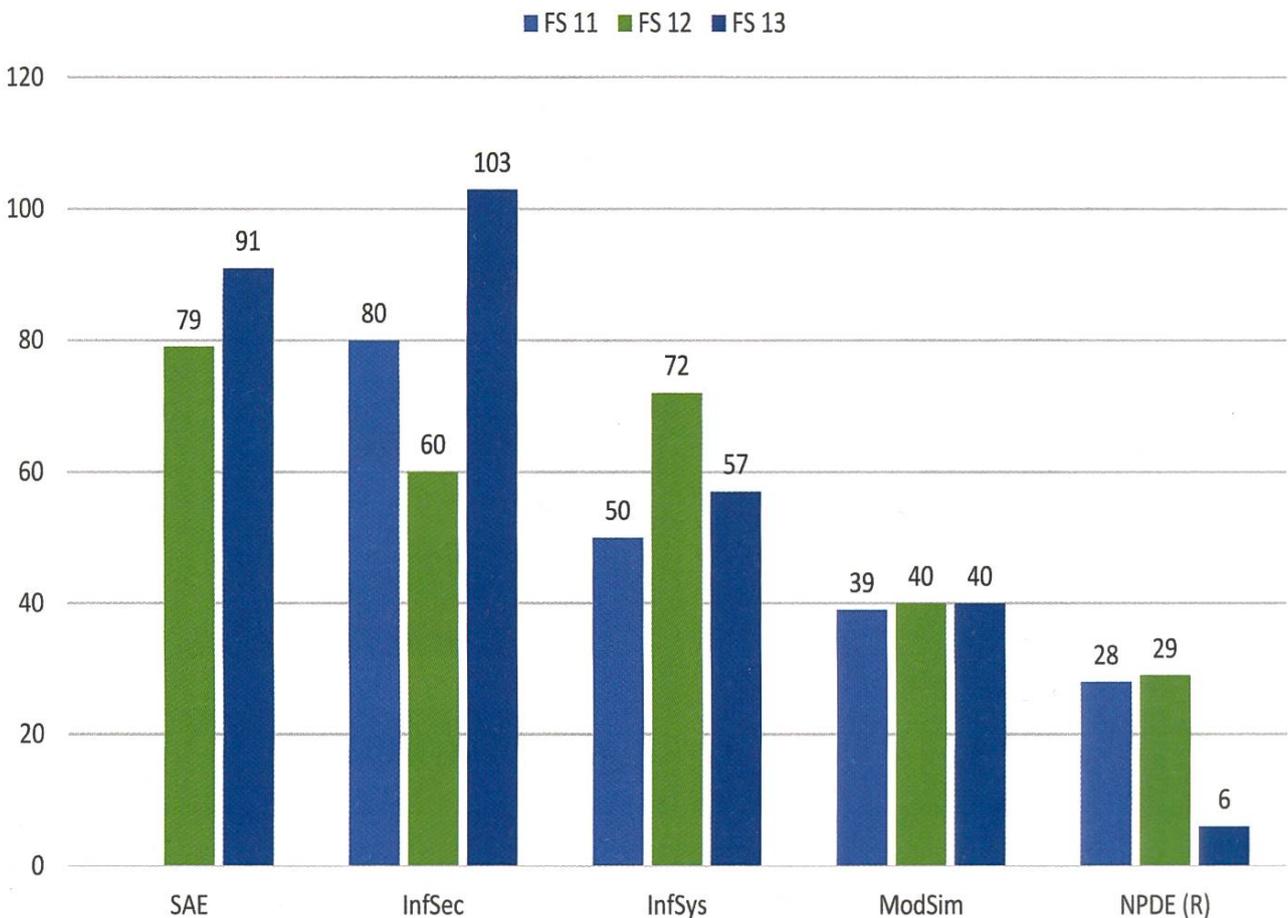
Obligatorische Fächer der Vertiefung FS



Bestandenquoten Obligatorische Fächer der Vertiefung



Teilnehmer Obligatorische Fächer der Vertiefung



Labs

Im Frühlingssemester fand bisher nur das «Computational Intelligence Lab» statt, welches sich im üblichen Notenrahmen bewegt. Es nahmen 92 Studenten an der CIL-Prüfung statt.

Als Neuerung gab es das erste Mal eine Repetitionsprüfung fürs «Advanced Systems Lab», welches alle drei teilnehmenden Studenten bestanden haben.

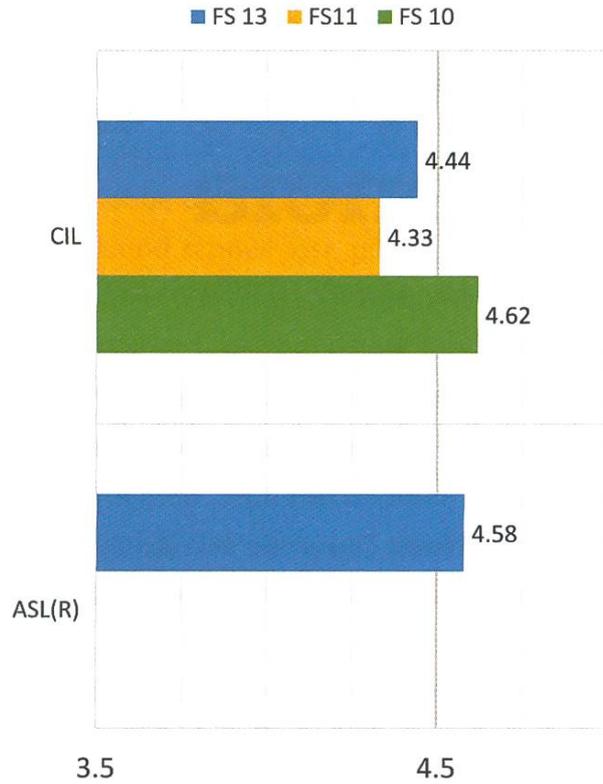
Masterabschlüsse

An dieser Stelle möchten wir allen Masterabsolventen gratulieren und wünschen ihnen viel Erfolg für die Zukunft!

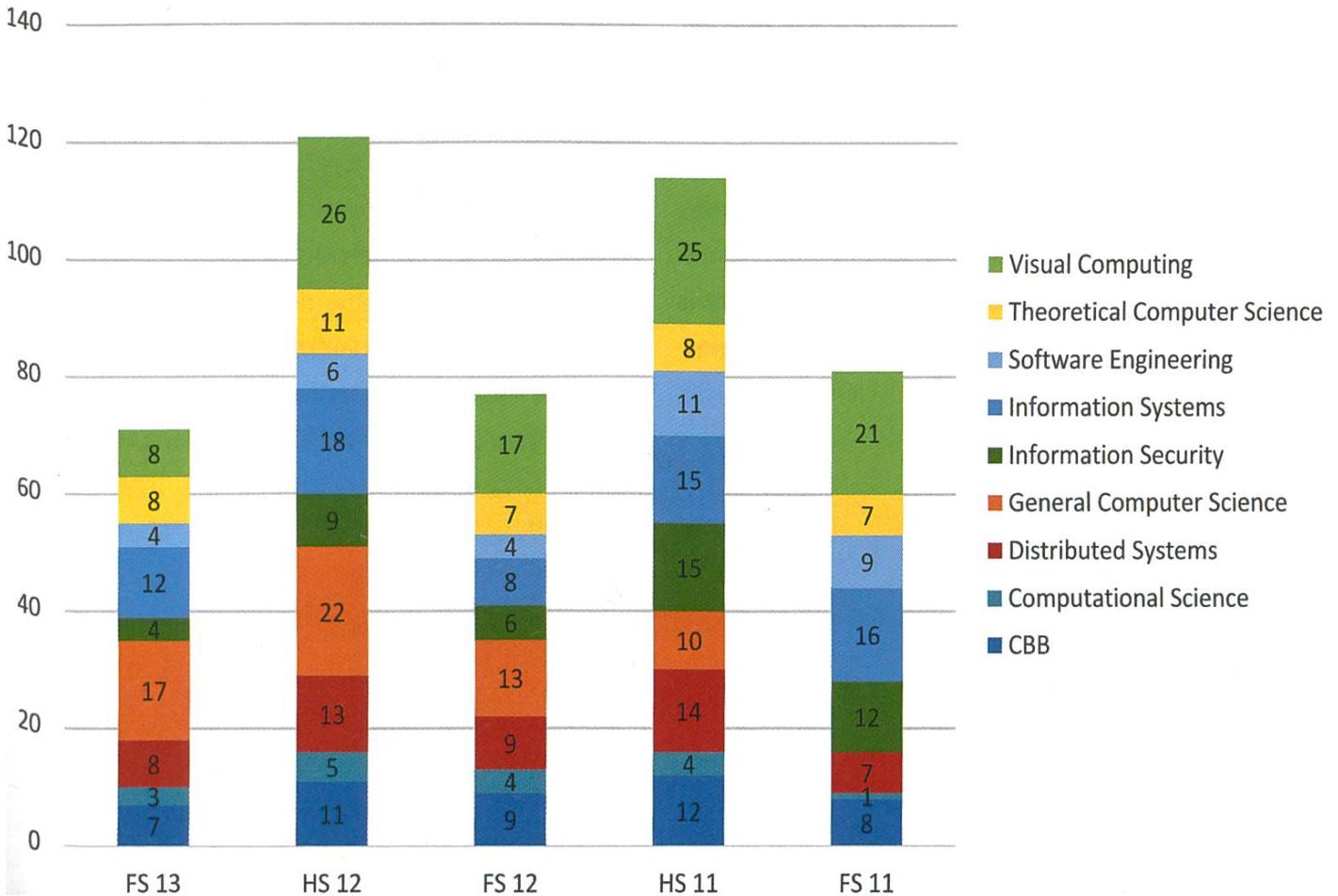
Für mich ganz persönlich ist die grosse Überraschung die Vertiefung «Visual Computing» welches nun auf dem Niveau von Theoretischer Informatik liegt! «General Computer Science»

bringt das erste Mal mehr Masterabsolventen hervor als Visual Computing und bringt am meisten Masterstudenten hervor. ⌘

Labs FS



Masterabschlüsse



ACM ICPC SWERC13 in Valencia

DANIEL GRAF – COMPETITIVE HACKER

Die ACM (Association for Computing Machinery) kürt neben dem Turing-Award-Gewinner jedes Jahr auch das Team der besten Informatikstudierenden im algorithmischen Programmieren. Dies im Rahmen des ICPC, dem International Collegiate Programming Contest⁽¹⁾. Jährlich nehmen weltweit Tausende von Dreierteams an regionalen Wettbewerben teil und kämpfen um einen der begehrten Plätze am Weltfinale.

Programmieren als Wettbewerb?

Die ACM-Kommission des VIS organisiert jeden Herbst den lokalen Ausscheidungswettbewerb an der ETH zusammen mit der EPFL. So trafen sich am späten Samstagmorgen des 12. Oktober eine Studentin und 20 Studenten, um den Tag mit Knobeln und Coden zu verbringen. Wer die Vorlesungen ACM-Lab oder Algolab besucht hat, kennt das Format: In fünf Stunden gilt es, möglichst viele Programmerrätsel zu lösen und zu implementieren. Jeder für sich, ohne Hilfsmittel oder Internet. Als Vorwissen reicht aus, was in «Datenstrukturen & Algorithmen» behandelt wird. Viele Probleme erfordern einen Standard-Algorithmus (Breitensuche, Dijkstras Kürzester-Pfad-Algorithmus, minimaler Spann-

baum, ...), jeweils kombiniert mit einem cleveren Twist. Programmiert wird in C/C++ oder Java mit den jeweiligen Standard-Libraries als Starthilfe.

Mir lief es gut, die ersten fünf Probleme wurden auf Anhieb accepted. Doch bei der DP-Aufgabe (Dynamische Programmierung) schlich sich ein kleiner Fehler ein: Ziel war es, zu zählen, wie viele Spielverläufe ein Fussballspiel mit einem Endresultat von X:Y bei Z Führungswechseln haben kann. Schliesslich fand ich den Bug, und es reichte mir für den zweiten Platz, sodass ich mich mit den beiden Mathematik-Studenten Nikola Djokic und Johannes Kapfhammer für das erste ETH-Team qualifizierte. Für das zweite Team qualifizierten sich Kieran Nirkko (D-INFK), Jan Wilken Dörrie (D-INFK) und Marco Keller (D-ITET).

Team-Training: Kondition und Technik

Nun galt es, aus Einzelkämpfern ein Team zu formen. Fortan stand uns zu dritt nur noch ein Computer zur Verfügung, sodass wir uns gut absprechen mussten, wer wann was codet und debuggt, ohne dass sich sechs Hände um die Tastatur streiten.

Während einer Woche übten wir täglich von 10 bis 15 Uhr. Die Trainings wurden organisiert von der VIS-ACM-Kommission in Zusammenarbeit mit Maxim Buzdalov und Fedor Tsarev (ihres Zeichens ACM ICPC World Champions von 2009 respektive 2008) von der University of Information Technologies, Mechanics and Optics (ITMO) in St. Petersburg. Die ITMO hat in den letzten paar Jahren die ACM-ICPC-Szene dominiert und



ETH-Team während dem Probe-Contest

die jährlichen Trainings mit ihnen sind für die ETH Teams immer sehr fördernd und fordernd zugleich. Sie haben einen grossen Teil zu den regionalen Erfolgen der letzten Jahre beigetragen. Während der Trainings sahen wir jeweils die Einsendungen der russischen Teams, wie sie an den Contests geschahen, und konnten uns so virtuell mit ihnen messen.

An manchen Tagen lief es super – an anderen weniger. Das binäre Grading ist da gnadenlos. Solange die Einsendung nicht alle Testfälle korrekt und schnell löst, gibt es keinen Punkt. Nicht jeder hat eine so einfache Strategie wie das russische Top-Team letztes Jahr: «First, each of us solved three tasks. Then we solved the last one together.»

¡Hola Valencia!

Zuversichtlich und auch ein wenig aufgeregt bestiegen wir das Flugzeug und freuten uns darauf, für ein letztes Wochenende dem Züri-Nebel nach Valencia zu entfliehen.

Nach der Ankunft im Hotel erkundeten wir die Altstadt Valencias. Vorbei an der Stierkampfarena und der Kathedrale begaben wir uns auf die Suche nach Churros, einem spanischen frittierten Gebäck, das man in heisse Schokolade tunkt. Zum Abendessen trafen wir uns mit den beiden Teams der EPFL, um eine grosse Auswahl an Tapas zu teilen.

Der Samstag war dann der Vorbereitung auf den kommenden Wettbewerb gewidmet. Austragungsort des Southwestern Europe Regional Contest (kurz SWERC^[2]) war die Universität Politecnica de Valencia. 44 Teams aus Spanien, Frankreich, Portugal, Italien, Israel und der Schweiz waren angereist. Nach der Registrierung vertrieben wir uns die Zeit mit Karten spielen und erhöhten so die Anzahl D-INFK-Doktoranden, die Jassen und Tichu beherrschen. Nach den Eröffnungsansprachen folgten die beiden obligaten Vorträge: Ein Professor des dortigen Instituts erklärte uns den Unterschied zwischen SaaS (Software as a Service) und PaaS (Platform as a Service), bevor uns eine Facebook-Ingeni-





Die ETH-Delegation am Mittelmeer: v. l. n. r.: Daniel Graf, Kieran Nirrko, Akaki Mamageishvili, Jan Wilken Dörrie, Nikola Djokic, Johannes Kapfhammer, Marco Keller, Jan Hązła

eurin ihr Code-Review-System erläuterte. Am Nachmittag folgte ein Testlauf, damit wir das Wettbewerbs-System auf Herz und Nieren prüfen konnten. Und um am nächsten Tag richtig fit zu sein, liess unser Team gar das noble Contest-Dinner aus, um richtig ausgeschlafen zu sein.

Der Contest

Jetzt galt es ernst. Wir jogten zur Uni, um richtig wach zu werden, und nutzen die letzten zehn Minuten vor dem Start, um alles aufzusetzen. Nikola codete schon mal den Maximum Flow auf Vorrat, denn der kommt doch sowieso immer dran – er sollte Recht behalten.

Die zehn Aufgaben waren spannend und herausfordernd. Wir berechneten die aktuellsten Themen aus der Blogosphäre, die Parität der Anzahl Nullen von $n!$, einen optimalen Stundenplan und vieles mehr. Nach vier von fünf Stunden schwebten über unserem Monitor sechs Ballone und wir lagen in Führung. Nur das erste Team der Universität Politècnica de Catalunya (kurz UPC^[3]) hatte bis dahin ebensoviele Aufgaben gelöst.

Während der letzten Stunde blieb die Rangliste eingefroren. Wir sahen nicht mehr, woran die anderen Teams noch arbeiteten, und somit blieb es spannend bis am Schluss. Wir lösten kurz darauf ein siebtes Problem, doch am achten bissen wir uns die Zähne aus. Es galt, einen planaren Graphen mit vier Farben zu färben^[4]. Ein simples Backtracking war zu langsam, und so versuchten wir es auch noch randomisiert. Glücklicherweise hatte ich in der Vorlesung über randomisierte Algorithmen von Prof. Angelika Steger gerade kürzlich von einer Abwandlung von Schönings 3-SAT-Algorithmus auf Färbeprobleme gehört. Wir kombinierten beides, op-



Die Teams der ETH und EPFL im Felsenkell



JUNIOR SOFTWARE ENGINEER (ZH/BE)

Haben Sie einen Hochschulabschluss in Informatik auf Master-Level und Java/Java-EE-Kenntnisse? Interessieren Sie sich für Applikationsentwicklung und Integration? Steigen Sie bei uns als Integrator oder Application Developer ein!

WERKSTUDIUM (TEILPENSUM) (ZH/BE)

Machen Sie Ihren Master an der ETH? Möchten Sie Ihre Kompetenzen bereits während des Studiums einsetzen? Bei uns stehen praxisnahe Aufgaben bereit, die gelöst werden wollen! AdNovum bietet auch einen reichen Fundus an Themen für Masterarbeiten.

Für weitere Informationen besuchen Sie uns auf www.adnovum.ch/jobs.
Unser HR-Team steht Ihnen unter Telefon +41 44 272 6111 gerne zur Verfügung.

AdNovum-Standorte: Zürich – Bern – Singapur – Budapest



timierten und schickten mit diversen Random Seeds ein, bis 24 Sekunden vor Schluss die 16. Einsendung akzeptiert wurde.

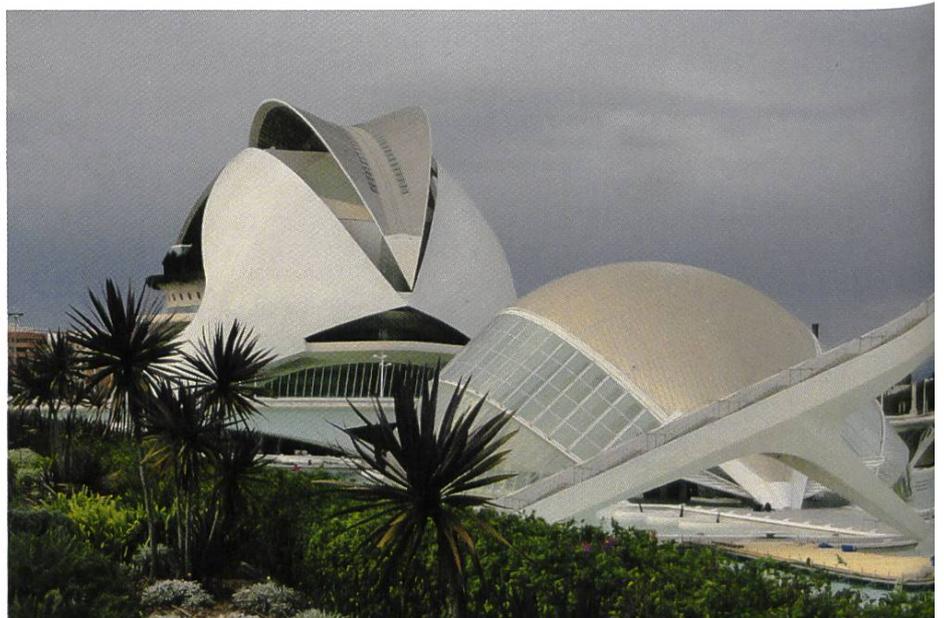
Auch das UPC-Team hatte insgesamt acht Probleme gelöst, sodass wir auf dem zweiten Platz der Schlussrangliste zu liegen kamen. Damit haben wir uns, zusammen mit dem UPC-Team, für das Weltfinale qualifiziert. So dürfen wir nächsten Sommer ins russische Jekaterinburg reisen, um dort die ETH und damit Südwesteuropa zu vertreten.

Insgesamt fiel die Schweizer Bilanz äusserst erfreulich aus: Drei der insgesamt zwölf Medailensätze gingen in die Schweiz. Die ETH holte Gold und Bronze auf den Plätzen 2 und 12, und die EPFL gewann eine Silbermedaille mit Platz 5 und verpasste Bronze mit Platz 13 nur haarscharf. Grund genug für eine feines Essen in der Stadt: In einem Felsenkeller wurde uns beste spanische Ente serviert.

Meer und Architektur

Bis zum Rückflug blieben uns noch ein paar Stunden. Für einen Spaziergang an den Strand reichte es allemal. Zum Baden war das Mittelmeer schon etwas zu kalt, aber es war ideal zum Jassen im Sand. Auf dem Weg zum Flughafen machten wir noch einen kurzen Abstecher zur Ciudad de las Artes y de las Ciencias. Dieser moderne Museumskomplex steht im Flussbett des in den 1990er Jahren trocken gelegten Río Turia. Entworfen wurden die extravaganten

Palau de les Arts Reina Sofía und L'Hemisfèric, zwei der Gebäude der Ciudad de las Artes y de las Ciencias



ten Gebäude vom valencianischen Architekten Santiago Calatrava, welcher übrigens an der ETH Zürich Bauingenieurwesen studierte und promovierte. Besonders bekannt sind seine skelettartigen Tragwerke und andere der Natur entlehnte Formen.

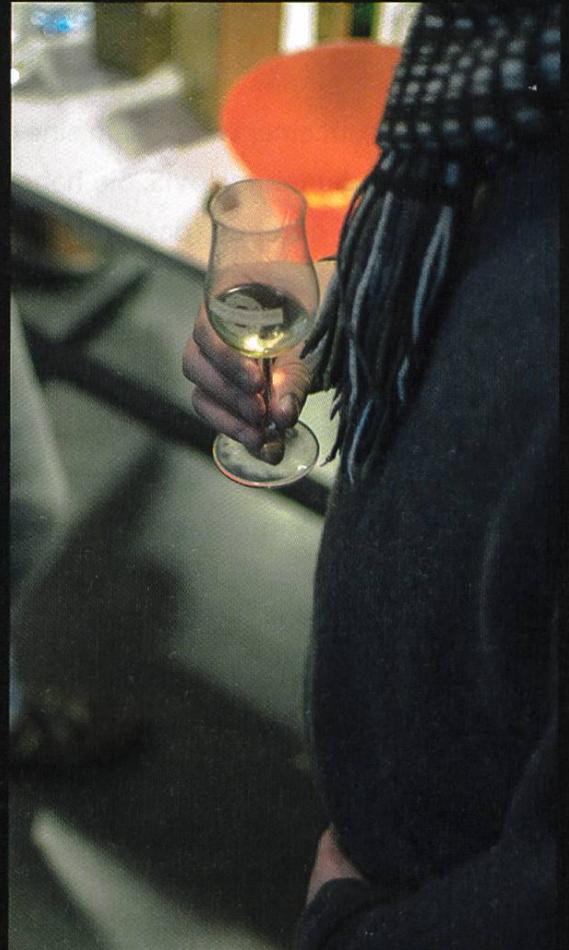
Fazit

Wir hatten ein tolles Wochenende, viel Spass und erfreuliche Resultate. An dieser Stelle möchten wir uns ganz herzlich bei unseren Coaches Jan Hązła und Akaki Mamageishvili und bei allen weiteren fleissigen Mitglieder der ACM-VIS-Kommission^[5] bedanken, welche unsere Teilnahme erst ermöglichten.

Wenn du interessiert bist, nächstes Jahr selbst mitzumachen oder mitzuhelfen, kontaktierst du am besten das ACM-Komitee unter acm@vis.ethz.ch. Es lockt eine Reise an den SWERC 2014 in Porto! ⌘

Links

- [1] <http://icpc.baylor.edu>
- [2] <http://swerc.eu>
- [3] <http://www.upc.edu>
- [4] http://en.wikipedia.org/wiki/Four_color_theorem
- [5] <http://www.vis.ethz.ch/en/about/committees/acm>



VISky Degu HS13

Ein «Danke» kann Berge versetzen

GREGOR WEGBERG – MACHT SICH GEDANKEN ZUR EHRENAMTLICHEN ARBEIT IM VIS

Vor gut zwei Wochen kamen zwei Studenten ins VIS Büro («oVIS») herein. Sie beschwerten sich, in einem nicht besonders freundlichen Ton, über die gerade leere Kaffeemaschine im Aufenthaltsraum. Immerhin haben sie noch Kaffeekapseln für den Tag zu gute und möchten diese jetzt beziehen. Wir informierten, dass wir die Kaffeemaschine auffüllen, sobald wir mit der restlichen Arbeit fertig sind. Mit unzufriedenem Gesichtsausdruck verliessen sie das oVIS.

Vor gut einer Woche kam eine ganze Gruppe von Studenten ins oVIS. Sie haben auf der VIS-Webseite gesehen, dass ihnen ein gratis Bier zugute steht, und wollten dieses jetzt beziehen. Wieder war der Tonfall sehr fordernd. Wieder erklärten wir, so gut es ging, wieso der Bierautomat noch kein gratis Bier an VIS-Mitglieder herauslässt. Auch informierten wir, dass bereits ein Codeli-Mitglied sich der Sache angenommen hat. Es ist leider nicht ganz so einfach und es reicht nicht aus, «einfach die VIS-Mitglieder an die Liste der AMIV-Mitglieder anzufügen» (frei zitiert). Auch dieses Mal verliessen die Studenten unser Büro unzufrieden und waren, zumindest hat es mir den Anschein gemacht, nicht ganz von unserer Begründung überzeugt. So schwer kann es ja nicht sein, dem Bierautomat beizubringen, den VIS-Mitgliedern gratis Bier herauszulassen. Leider ist es aber genau das.

Wöchentlich erlebe ich solche Situationen im oVIS oder per E-Mail. Es gibt VIS-Mitglieder, die davon überzeugt sind, dass sie mit den 10 CHF VSETH-Mitgliedschaftsgebühr sich ein Anrecht auf gewisse Dienstleistungen gekauft haben

und wir, die aktiven VIS Mitglieder, diese gefälligst zu erbringen haben. An dieser Stelle möchte ich ganz klar sagen: Nein. Das habt ihr nicht. Die zwei Kapseln Kaffee können wir unseren Mitgliedern nur offerieren dank der unglaublichen Arbeit der Kontaktparty-Kommission, des Visionen-Teams und des VIS-Vorstands. 10 CHF würden gerade einmal für gut 18 Kaffeekapseln pro Semester reichen. Bei 2 Kaffeekapseln pro Tag wärst du nach 9 Tagen auf dem trockenen. Darin habe ich jetzt noch nicht einmal die Betriebs- und Anschaffungskosten eingerechnet. Geschweige der Finanzierung aller VIS-Events, die wir unter dem Semester anbieten.

Versteh mich nicht falsch. Wir alle machen unsere Arbeit gerne. Doch immer wieder werden wir durch einzelne in einer Weise behandelt, dass uns jegliche Lust an der Arbeit zu vergehen droht. Hier und da ein «Danke» zu hören, würde vieles verändern.

Es gibt natürlich auch die (wenigen) Studenten, die uns tatkräftig unterstützen, ohne aktiv im VIS zu sein. So war erst letztens jemand im oVIS und hat sich einen Besen bei uns besorgt, um den Aufenthaltsraum zu säubern. Wow!

Bei all den VIS-Mitgliedern, die sich unserer Arbeit im Klaren sind und uns dabei unterstützen, möchte ich mich deshalb bedanken! Eure Dankbarkeit und Unterstützung ist es, welche mich persönlich antreibt, weiterhin im VIS aktiv zu sein und mein Bestes für euch zu geben. Durch euch werden unsere Events zu unterhaltsamen Abenden für alle und durch euch erhält unsere Arbeit erst einen Sinn.

Falls auch du uns bei unserer Arbeit unterstützen möchtest, komm doch im oVIS vorbei, und wir zeigen dir, was es so zu tun gibt. Vielleicht finden wir ja etwas, an dem du gerne mitarbeiten würdest!

⌘

So war erst letztens jemand im oVIS und hat sich einen Besen bei uns besorgt, um den Aufenthaltsraum zu säubern. Wow!

The state of the beer vending machine concerning VIS members

Some of you may remember the time VIS offered its members one free beer per day in the student lounge. Maybe some of you even remember that the machine broke down and AMIV had to organize a new one and modify it to work with ETH Legi cards. The new machine had some troubles communicating with our servers, consequently we couldn't provide free beer to our members any more. The hard work of a Codeli commission member gave us the power to provide our students with free beer again. Well, at least technically speaking. The machine again knows how to talk to our servers. However, we cannot yet give away beer as the budget for provisions is exhausted by the increased demand for coffee. Therefore we will all have to wait until the next general assembly and the approval of a new budget that accommodates free beer for VIS members. But don't forget that you can already get beer from the machine for only CHF 1.20. That's pretty reasonable.

FIGUGEGL

MANUEL BRAUNSCHWEILER – DREHT BRÖTCHEN IM KÄSE



Alle Jahre wieder lädt der VIS zum fröhlichen Fondue-Plausch. Auch dieses Jahr fanden sich die VISler, Kollegen und Kolleginnen, Lebensabschnittspartner und Affären zahlreich im StuZ² ein, um sich diese traditionell schweizerische Köstlichkeit durch den Magen gehen zu lassen. Käse wurde zentnerweise angeschleppt, Knoblauch in ungeahnten Massen zu kleinen Stücken zerhackt und Brot so hoch aufgetürmt, dass dem Prime Tower als höchstes Gebilde Zürichs schon Angst und Bange wurde.

Bereits Stunden zuvor fand sich ein kleines Grüppchen VISler im beängstigend leeren StuZ² ein und schwur sich, den Saal derart mit schweren Bänken und Tischen vollzustopfen, dass auch wirklich niemand wegen Platzmangels in die dunklen Katakomben des ABBsolut abgeschoben werden müsste, um dort sein einsames Brötchen im Käse zu drehen. Und so ist es auch geschehen. Den Helfern war dabei der süsse Schmerz von Muskelkater am nächsten Tag Lohn genug – vorausgesetzt nicht eine andere Form von Kater gewann die Überhand. Schliesslich wurde auch der sagemuwobene

Kühlschrank tief unter dem CAB um etliche Flaschen Weisswein und Bier erleichtert, deren Inhalt kurze Zeit später in die gierigen Kehlen der Gäste floss und die winterliche Kälte von dort vertrieb; sie gesellig und munter machte.

Während die Schlange hungriger Studenten vor dem CAB immer kürzer wurde, verteilten im Innern des CAB die Helfer Caquelons und Käse an die gebildeten Vierergruppen, die dann emsig damit begannen, ihren Käse zu schmelzen und den sich dabei verbreitenden charakteristischen Duft in ihre Nase (und Haare und Kleider und alles andere auch) sogen (aber das würden sie erst



später erfahren). Dass sich dabei unterschiedliche Rührtechniken herauskristallisierten, war bei der geballten Ladung IQ und Erfindergeist im Raum ohnehin nur noch eine Frage der Zeit. Doch sobald der Hunger zu gross wurde und der Käse so weit geschmolzen war, dass er in weichen Fäden von der Gabel tropfte, war es vorbei mit taktischem Rühren und die annuelle Käsevernichtung nahm ihren Lauf. Als die erste Käseladung vertilgt war, waren erst die wenig-

sten satt, und der Ruf nach Nachschub wurde laut – welcher mit noch mehr Käse auch gerne beantwortet wurde. Es war perfekt. Sogar der lang gescholtene und berüchtigte Frauenanteil war am FIGUGEGL auf einem Allzeithoch. So waren alle glücklich und stopften sich mit Käse voll, bis sie nicht mehr mochten.

Während sich der Saal allmählich leerte, wanderten viele noch in den Aufenthaltsraum, um sich bei einer Partie Billard oder Tischfussball zu messen. Die Stimmung war gut, und erst gegen 2 Uhr morgens verliessen auch die letzten Bewohner das CAB, strömten an die kühle Nachtluft hinaus und liessen das schon wieder aufgeräumte StuZ² mit seinem milden Käseduft in der Dunkelheit zurück.

8



Klimaskeptizismus

STEFAN DIETIKER – SKEPTISCH OB DER VIELEN SKEPSIS

Die Klimaforschung erfährt immer mehr Argwohn. Kritiker schicken sich an, die Resultate der Klimaforschung in Zweifel zu ziehen. Ein bisschen mehr Ehrlichkeit, gefälligst.

Nehmen wir an, alle führenden Physiker dieser Welt würden gemeinsam ein tausendseitiges Schriftwerk verfassen, mit dem Ziel, auch den letzten Zweifler davon zu überzeugen, dass die Gravitation nicht nur «eine Theorie», sondern nach aktuellem Kenntnisstand eine schlichte Realität ist. Liesse sich mit Zweifel an der Gravitation Geld verdienen, gäbe es garantiert Schlaumeier, die in diesem Weltgravitationsbericht Fehler suchten, um die Menschen vom Gegenteil zu überzeugen – und da jedes von Menschenhand gemachte Werk von diesem Umfang mit an Sicherheit grenzender Wahrscheinlichkeit Fehler enthält, würden sie auch fündig.

Und wäre es auch noch bequemer und intuitiver, im Zweifelsfalle die Existenz der Gravitationskraft in Abrede zu stellen – wir hätten es tagtäglich mit einer Armada von Gravitations-skeptikern zu tun, die ununterbrochen darauf hinwiesen, dass Heliumballone ja steigen und damit der endgültige Beweis erbracht sei, dass die Gravitation ein Hirngespinnst gewissenloser Wissenschaftler ist, die sich gegen die Menschheit verschworen hätten.

Noch schlimmer käme es aber, wenn sich auch politische Parteien in die Diskussion einschalten würden, sich die einen für, andere gegen die Existenz aussprechen würden. Für einen Laien läge es dann auf der Hand, dass die Frage

nach der Existenz der Gravitation nicht durch systematische Forschung, sondern mittels einer politischen Debatte zu klären sei.

Debatte unter Amateuren

Bevor nun die «Skeptiker» damit beginnen, mich mit ihren «Argumenten» einzudecken, möchte ich gleich vorwegschicken, dass ich ein verschwindend kleines Interesse habe, mich mit Laien über den Wahrheitsgehalt des Weltklimaberichts oder ähnlichem auseinanderzusetzen.

Denn ich bin selbst Laie und masse mir deshalb nicht an, über die nötigen Kenntnisse zu verfügen, um für die eine oder andere Seite Partei zu ergreifen. Ich weiss lediglich, dass es in der Informatik Jahre braucht, bis man mit gutem Gewissen von sich behaupten kann, ein Gebiet wirklich zu meistern – und ich gehe davon aus, dass es sich in der Klimatologie nicht anders verhält. Ich beobachte aber, dass es vergleichbare Debatten gibt, die nach ähnlichen Mustern ablaufen.

In Teilen der USA wird zum Beispiel seit geraumer Zeit herzhaft darüber debattiert, ob nun die Evolutionstheorie tatsächlich als wissenschaftlich anerkannte Theorie an den Schulen unterrichtet werden soll. Und auch in dieser Debatte gibt es unzählige Personen, die sich gegen die Evolutionstheorie aussprechen – Personen, die zwar mitunter prominent, aber allesamt

eben auch Laien sind. Und auch hier werden handbüchene «Beweise» für die Unzulänglichkeit der bestrittenen Theorie angeführt. Sämtliche derartige «Beweise» stellen isolierte, an sich unbestrittene Fakten in einem falschen Kontext dar, um den Betrachter intuitiv in die Irre zu führen. Aber mit Erfolg, denn ein nennenswerter Teil der amerikanischen Bevölkerung hat heute Zweifel an der Gültigkeit der Evolutionstheorie.

Etwas mehr Ehrlichkeit

Sollten die führenden Klimawissenschaftler letztlich unrecht haben, es wäre nicht das erste Mal, dass sich die Wissenschaft mit derart gross angelegten Vorhersagen irrte. Die Apologeten des Klimaskeptizismus hingegen haben keine handfesten, wissenschaftliche Gründe, an der Richtigkeit eines Weltklimaberichts zu zweifeln. Die Überzeugung, mit der sie voranschreiten, fusst nicht auf kritischer Vernunft, sondern im Gegenteil auf der blinden Hoffnung, dass schon alles nicht so schlimm sein wird. Andererseits bestehen handfeste Partikularinteressen, den

Menschen beispielsweise glaubhaft zu machen, die Verbrennung fossiler Energieträger sei nicht weiter tragisch. Aber das ist nicht weiter verwunderlich, denn wo Hoffnungen sind, werden diese auch bewirtschaftet – das gilt freilich ebenso für die Apologeten der Energiewende.

Manch einem mag der Vergleich zur Gravitationstheorie absurd erscheinen. Unzählige historische Beispiele belegen aber, dass wir durchaus die Fähigkeit haben, den Status quo aus Eigeninteressen durch alle Böden zu verteidigen. So wurde auf diesem Kontinent die Ansicht, dass die Erde um die Sonne kreist und nicht umgekehrt, lange Zeit bekämpft, bzw. sie wurde toleriert, solange sie als rein «mathematische Hypothese» gehandelt wurde und die Verkündung der offiziellen Wahrheit dem Klerus überlassen blieb. Ich befürchte leider, dass die Wissenschaft auch heute noch mehr als «Hypothesen» zu bieten hat. ⌘

Errata «Beginn»

Men of the Bats

Im Artikel «Men of the Bats» in der letzten Ausgabe haben sich (mind.) zwei kleine Fehler eingeschlichen. Zum Ersten wurden im Fliesstext die Abbildungen 2 und 3 verwechselt: Abb. 2 zeigt einen schmalbandigen Ruf, während Abb. 3 eine Rufsequenz zeigt. Zudem ist die Beschriftung zum Foto auf Seite 39 biologisch nicht ganz korrekt: Ein «Nest» ist per definitionem eine von ihren Bewohner erbaute Behausung. Fledermäuse bauen keine Nester, sondern verstecken sich bspw. in Ritzen und Spalten von Häusern.

Bildnachweise

Cover © Nicola Maracci Rossi <http://nmr.io/>

S. 12-15: © Lukas Limacher

S. 29: © Blizzard Entertainment®

S. 30: © Subset Games

S. 31: © Simon A. Eugster

S. 32-37, 43, 46-47: © VIS

S. 50: Original by Morwen, edited by Nightstallion & VIS, via Wikimedia Commons (CC BY-SA 3.0)

S. 56-60: © Simon A. Eugster

A Short Story on Learning Russian

CLEMENS LUTZ – NJET? DA!

Six words forming a short sentence can have a large impact on your life. Your daily routine changes. Walking the streets in any larger city, you notice people and symbols you were blind to before. Your thought process becomes different in many everyday situations. All because of a desire expressed in six words. For me these words were: «I want to learn another language.»

A difficult choice

Once that decision had settled in, the big question was: «A language, yes, but which one?» That one led to me think for some time, but in the end it was a question about interests, emotions, and practicality.

The language centre offerings, although numerous and well-considered, are select and finite. Being a student of ETH, there are criteria to which languages at a given level can be attributed as ECTS points. While this shouldn't hold anyone back from following an inner instinct, it can help to narrow down the range of choices. A language takes time, years even, to learn and thus the utmost imperative is intrinsic motivation. Friends speaking that particular language or a connection to the culture are beneficial for making the final decision and for staying motivated to study.

For me, the choice came as a combination of the above. As a rational but undecided ETH



student, receiving ECTS is definitely a big plus for any language. I have always been interested in human history and politics, as such the influence of Russia and, before it, the Soviet Union, on the course of history in past centuries and today cannot be understated. However, one can always argue that the same goes for other nations as well. In the end, it came down to a primitive feeling. Strange as it may sound, I believed that I can feel at home in Russian culture, even though I am not and will never be Russian. And with time that affinity has grown.

From intention to realisation

Having decided on a language the issue became how to get started on it. A logical first step was to register for a language course, e.g. at the language centre. ETH students must register themselves both in «myStudies» and the language centre itself. Take note that there is only a short time window, usually around 2 or 3 weeks before start of term, when registration is open. The seats are limited and tend to be booked out quickly, thus try to register early. Set yourself a calendar reminder, get reliable friends to text you or program an Arduino to pour cold water over your head—whatever it takes so that you're not too late. A proper language course cannot be substituted by anything else, as the teacher guides the class in a structured approach step by step as to not overwhelm, giving many hints and insights along the way. Thus far I only have made positive experiences with the teachers and the institution of the language centre. They are very competent, patient and supportive, of their pupils.

There are many additional learning opportunities at your disposal outside of classes. The self-access centre provides books, audio CDs, and films for autonomous studying. Additionally, they have weekly help desk sessions supervised by tutors, who are often native speakers. The internet is rich in language resources as well. News websites, blogs, online dictionaries and encyclopaedias, cartoons, and images are just a few examples that come to mind. Next to browsing to a Wikipedia page and then switch-

ing to the foreign language, one of the more creative ideas I've come across is to type in new vocabulary on Google's image search for pictorial memorization. Undeniably, the best and most motivating resource to have are friends who happen to be native speakers. After all, languages exist for communicating with people. If you find a partner interested in learning your own native language among them, regular "Tandem" sessions are definitely worth closer consideration. It should go without mentioning that visiting a country full of native speakers is an excellent way to leave desk and pen behind for a few weeks.

Ultimately, being serious about learning a language requires sweat, boring repetition and, occasionally, overcoming frustration. Seeing it as a hobby instead of a course earning me a meagre 2 ECTS helps, as in busy phases during term, lectures with no fixed deadlines and the least credits tend to have lower priority and get de-scheduled. Hobbies, on the other hand, are well-suited as leisure time after a long day. A one-hour-per-day routine, regardless of the exam session and semester break, has worked well so far. The amount isn't the decisive factor—the repetition is.

Repetition, however, brings visible results only with time, making progress hard to see. Speaking from experience, that's where frustration can set in. For example, to systematically re-visit vocabulary, I have established a flash card methodology for myself. One card in particular came up again and again, and each time I simply couldn't remember the word in Russian. Finally, after over half a year, it stuck. The word was «повторить» («povtorit» in Latin characters), which means «to repeat».



Easily Hard

Diving into unknown waters headfirst is always cold initially. But the more you swim, the warmer it becomes. The Cyrillic alphabet was the foremost hurdle to overcome. While texts written in Russian may look cryptic to an English or German speaker, the good news is that in practice, one becomes accustomed to the type face within half a term. The bad news is that alphabet's complexity swiftly pales when compared to the grammar or vocabulary aspects. My observations made while studying these are interesting, however.

There are a large number of similarities between Russian and the languages I already know or have touched previously. For starters, many words are loanwords from other languages, making them rather simple to memorize. Declensions of nouns, adjectives and conjugations of verbs are, as in Latin, constructed with postfixes. If there is anything certain in Russian writing, it is that all words are inflected. Thus even «пицца» («pizza») becomes «пиццу» («pizzu») in its accusative form. At my current knowledge level, German appears to have many grammatical similarities to Russian, more so than English does. As I progress, however, there appear more and more diverging concepts. Continuous and progressive verb aspects are one of these. The closest example to me is in English: «Я делал»–«Я сделал» («la delal»–«la sdelal»), which is equivalent to «I was doing»–«I did».

Differences are numerous as well, starting, as previously mentioned, with the alphabet. There is no strict word order in sentences (think: subject–verb–object), although words can be formed into groups inside a sentence, e.g. with prepositions. This often leads me to order them either shortest-word-first or analogously to Ger-

man, both of which presumably sound awkward to native speakers. Conversely, native speakers often have difficulty with articles in languages foreign to them. For instance, «a» and «the» are frequently either confused or completely forgotten in their English. The reason is that Russian knows no articles, which in reverse eliminates a factor of complexity for me as a student.

Russian is classified as an East Slavic language, and in a broader sense an Indo-European language. Even when coming from as far as Germanic languages, it is clearly recognizable that there has been lingual exchange between them over the centuries. Nonetheless, while learning Russian is not as difficult as it may appear, it is hardly easy.

Reflections

Currently, my personal goal is two-fold: To speak fluently without accent, and to understand classical literature. The former is unreachable, but possible to approximate closely. The latter is regularly ridiculed by Russians themselves, when they say no one understands Tolstoy's «War and Peace» anyway. Then again, which native German speaker claims to truly comprehend Goethe's «Faust»? After three terms of studying the language, I've concluded that these ambitions will likely take me a decade to reach satisfactorily.

Hither, my path in learning Russian has acquainted me with several excellent friends, has given me insight into another culture, and led me to see with my own eyes a part of the world virtually inaccessible to Joe Bloggs, citizen of Western Europe, at the time of my birth. Even if the ambitions listed above prove to be unachievable as stated, the path thither is worth every step and stumble.



Apply for 2014 technical internships and full-time roles at Google now:
www.google.com/jobs/students

Full-time Software Engineering Roles

If you are graduating soon, you can apply to join us as a full-time Software Engineer. We have open roles in Zurich and other offices in Europe and globally. A large number of positions are available in our headquarters in Mountain View, California. Wouldn't it be great to start your career at Google's headquarters in the Silicon Valley after graduation?

Who can apply: BSc, MSc or PhD in Computer Science or a related technical field graduating in 2013/2014.

How to apply: Apply online at www.google.com/jobs/students for Software Engineering full-time roles in our offices globally.

I am interested in starting my career in Google's headquarters in Mountain View, California, in 2014: If you are interested in a full-time Software Engineering position in Mountain View we recommend you to apply online as soon as possible to allow your application to be processed in time. To be eligible for a Software Engineering New Grad position in Mountain View, you need to have 4 years of university education in Computer Science or a related technical field completed by March 2014.

What happens after you apply: Regardless of the location you apply for, a recruiter will reach out to you if they are interested in taking your application forward. If we move forward you will have technical phone interviews followed by several onsite interviews with engineers. If you are successful and pass different stages of reviews you will receive an offer.

2014 Software Engineering Internships

Technical internships at Google are full time paid positions, take place all year-round and have a minimum duration of 3 months. If you are interested in an internship we recommend you to apply online at www.google.com/jobs/students as soon as possible. We expect a majority of positions to be filled until January 2014. We are now looking for interns for all of 2014.

Text & Gestaltung: Felix Würsten

Dass ein Sprachstudium für ihn nicht in Frage kam, das war Georg Schätti schon früh klar, interessierte er sich doch im Gymnasium in erster Linie für die naturwissenschaftlich-technischen Fächer. Biologie und Informatik blieben schliesslich als mögliche Studienrichtungen übrig. Die Wahl fiel auf das Zweite, fand er doch die Aussicht, später einmal in einem grossen Pharmaunternehmen oder als Lehrer zu arbeiten, nicht besonders reizvoll. Die Berufsaussichten in der Informatik hingegen schienen vielseitiger zu sein, zumal die Informatik damals auf dem Höhepunkt der Dotcom-Blase gerade einen enormen Boom erlebte. Die damalige Euphorie hat sich zwar inzwischen wieder etwas gelegt. Dennoch war die Studienwahl offensichtlich richtig gewesen: Georg Schätti hat heute als Senior Consultant

beim Beratungsunternehmen AWK in Zürich eine Stelle gefunden, bei der er sein technisches Flair mit der Freude am Kundenkontakt auf ideale Weise miteinander kombinieren kann.

Die Zeit an der ETH Zürich hat Georg Schätti in guter Erinnerung, auch wenn ihm der Einstieg in das Studium nicht ganz leicht gefallen war. «Nach drei Wochen realisierte ich, dass das Tempo an der ETH Zürich doch relevant höher ist als am Gymnasium», meint er lachend. So richtig wohl fühlte er sich dann im Hauptstudium, wo er sich endlich in die Fachthematik vertiefen konnte und ihm «so richtig der Knopf aufging», wie er es heute formuliert. Seine Diplomarbeit schrieb er in einem sehr hardwarenahen Gebiet: Seine Aufgabe war es, die Möglichkeiten der «Java Virtual Machine» (JVM) von Sun auszuloten. «Zu dieser Zeit wurden erstmals auch auf günstigeren Rechnern Mehrkernprozessoren eingesetzt», erzählt er. «Die in der Diplomarbeit angestrebte JVM sollte es ermöglichen, auch bei Programmen, die noch nicht auf die neue Chiparchitektur ausgerichtet waren, die höhere Rechenleistung auszuschöpfen.» Es war eine zähe Arbeit, die sich Georg Schätti ausgesucht hatte, musste er sich doch tief in den komplexen Programmcode hineinknien. Doch er lernte dabei vieles, das ihm auch heute noch in der Arbeit zugute kommt.

Noch bevor er sich an die Masterarbeit machte, absolvierte er im Silicon Valley ein Internship bei der Firma Xilinx in San Jose. Den Kontakt zu diesem Unternehmen fand er über eine Anfrage an die ETH-Aussenstelle in San Francisco. «Ich konnte in Kalifornien ein kleines Projekt bearbeiten, das mir sehr entsprach», erzählt er. Ähnlich wie bei der Masterarbeit war auch dieses Projekt in einem hardwarenahen Bereich angesiedelt. Georg Schätti setzte sich intensiv mit «Field Programmable Gate Arrays» (FPGA) auseinander. Diese Prozessoren werden bevorzugt dort eingesetzt, wo eine hohe Performanz gefragt ist, aber keine spezifisch entwickelten Chips zur Verfügung stehen.

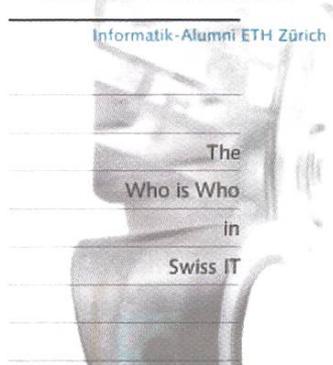
Der Aufenthalt in Kalifornien gefiel Georg Schätti derart gut, dass er sich nach dem Masterabschluss entschloss, erneut ins Ausland zu gehen – diesmal jedoch nach Osten. Anhand eines klaren Kriterienkatalogs kristallisierte sich nach und nach Singapur als ideale Destination für diesen Aufenthalt heraus. Er buchte kurzerhand einen Flug nach Asien und machte sich vor Ort auf Stellensuche. Nach zwei Wochen fand er einen Job beim Startup-Unternehmen Fonemesh, das einen Mitarbeiter mit breiten

Portrait Letter 019 Georg Schätti



[I][A][E][T][H]

Informatik-Alumni ETH Zürich



Informatikkenntnissen suchte. Die Firma hatte vor, eine Software zu entwickeln, mit der mobile Ad-hoc-Netzwerke aufgebaut werden können. Diese Netzwerke hätten es dann ermöglicht, den Kundinnen und Kunden neuartige Dienstleistungen anzubieten. Bald einmal zeigte sich jedoch, dass diese Idee zwar überzeugend tönt, sich kommerziell jedoch nicht so leicht umsetzen lässt, braucht es doch eine kritische Masse an Nutzern, damit das Konzept funktioniert. Die Firma richtete sich deshalb strategisch neu aus und begann, Apps für Mobiltelefone zu entwickeln. «Nach anderthalb Jahren sah ich keine befriedigende berufliche Perspektive mehr», berichtet Schätti. Zudem wurde ihm klar, dass er längerfristig nicht in Singapur leben wollte. Das starre politische Umfeld, die beschränkten Freizeitmöglichkeiten und die immer deutlicher spürbare Distanz zu Familie und Freunden in der Schweiz erlebte er zunehmend als Belastung.

Aus seiner Zeit in Asien brachte er jedoch eine wichtige Erfahrung zurück: In Singapur wurde ihm bewusst, dass ihm der Kontakt mit Kunden sehr entspricht. Deshalb hielt er nach seiner Rückkehr in die Schweiz nach einer Stelle bei einer Consultingfirma Ausschau. Auch hier ging er wiederum sehr systematisch vor: «Ich erstellte eine Liste von Kriterien, die mein künftiger Arbeitgeber erfüllen sollte», erzählt er. Nach den teilweise etwas ermüdenden Erfahrungen bei einer Startup-Firma wollte er nun bei einem Unternehmen arbeiten, bei dem die grundlegenden internen Prozesse gut organisiert sind. Gleichzeitig sollte die Firma aber auch nicht zu gross sein, damit er als Mitarbeiter nicht den Überblick verliert. Dass die Wahl am Ende auf AWK fiel, hatte auch damit zu tun, dass bei diesem Unternehmen Bewerbungen sehr professionell abgewickelt werden. «Ich bekam auf meine erste Anfrage rasch eine Antwort und konnte kurz darauf mit der verantwortlichen Fachperson sprechen.»

Auch als Berater hat Georg Schätti heute häufig mit Projekten zu tun, die relativ hardwarenah sind und bei denen es darum geht, technische Systeme erfolgreich zu implementieren. «Es sind vielseitige Herausforderungen die ich lösen muss», berichtet er. «Von der IT-Architektur über Datensicherheit bis hin organisatorischen Fragen bearbeite ich ein breites Spektrum.» Viele Aufträge stammen von Behörden und Infrastrukturanbietern. Für ihn hat gerade dies einen besonderen Reiz: «Ich sehe, wie die Schweiz funktioniert», meint er lachend. «Wir sind ein kleines Land, und so lernt man schnell die richtigen Personen kennen, die man kontaktieren muss, wenn man eine bestimmte Information benötigt.»

Dass er als Berater nun nicht mehr direkt in die Softwareentwicklung involviert ist, habe Vor- und Nachteile: «Einerseits würde ich manchmal gerne wieder direkt an der Entwicklung von Lösungen mitwirken. Andererseits erinnere ich mich noch sehr gut an die Projekte im Studium, wo ich manchmal stundenlang in mühsamer Kleinarbeit irgendwelchen Fehlern nachgehen musste.» Sein Ziel sei es heute, das technische Wissen mit nicht-technischen Kenntnissen zu verbinden. «An der ETH habe ich ein sehr breites und solides Informatik-Know-how erhalten», meint er. «Als Berater lerne ich nun, was es sonst noch alles braucht, damit eine technische Lösung im praktischen Alltag dann auch wirklich so funktioniert, wie sie sollte.»



Biographie

Georg Schätti hat an der ETH Zürich Informatik studiert und das Studium 2008 mit dem Diplom abgeschlossen. Noch während seiner Studienzeit absolvierte er von 2006 bis 2007 ein Internship bei der Firma Xilinx in San Jose (USA). Nach dem Diplomabschluss arbeitet er von Mitte 2008 bis Ende 2009 als Entwickler bei der Startup-Firma Fonemesh in Singapur. Seit 2010 arbeitet er nun als Senior Consultant bei der AWK Group. Diese ist mit über 140 Mitarbeitenden eines der grössten Schweizer Beratungsunternehmen für Informationstechnologie. Die AWK Group ist schweizweit tätig mit Standorten in Zürich, Bern und Basel und bietet Consulting, Engineering und Projektmanagement an. Kontakt: Georg.Schaetti@awk.ch

Parkour

SIMON EUGSTER – HAS FUN WITH OBSTACLES

You certainly have walked down the ramps on railway stations several times already. Depending on where you exit the train, the shortest path is clearly over the railing and not around it, especially on the first meters, where ground levels are still around equal. If you know the desire of taking the shortest path in such situations, Parkour might be for you.

The basic goal of Parkour is just this: taking the shortest path from point A to B. If there is a railing in-between, jump over it. If there is a wall, it is time to wall run. Obstacles become opportunities for getting to B more efficiently—and with more fun, too.

Fitness

But Parkour is much more than just that. The first obvious benefit is fitness. The diversity of moves used in Parkour leads to numerous muscles being trained, from your fingertips to the tips of your toes, which is a great alternative to always training with the same devices in an indoors gym. Strength training is also a firm part of trainings; usually left for last to tickle out the final reserves. Strength is not only important for certain moves, which sometimes require muscles that don't get trained in the daily life of a Computer Science student (i.e. the muscles $\notin \emptyset$)—strength is also a key factor to avoid injuries. For example, to go easy on joints when landing, we land on the forefeet and bend the knees, feathering with the muscles instead of blocking with the bones.

Did you know that not only muscles become stronger with training, but also bones? They build new stiffeners inside when stressed. I find it cool how alive even bones are.

Body control

Body control and balance is another aspect that is trained, and hereby improved, with Parkour. Once again, let's compare a CS student to a Biology student when typing. The biologist expends all his attention to locate the next key on the keyboard, and then, with his right index finger, pushes down the key—or maybe the key next to it. The CS student knows the position of all keys including special characters by heart, therefore types blindly while talking and eating, with a hit rate of 99.999 %.

Likewise in Parkour. When jumping on a ledge with a precision jump, we land on its edge with our forefeet. Precisely. After a while you feel where to put your feet, how much momentum is required, what distance the jump is—without thinking. Fortunately, this can be practised. That's why, occasionally, you can observe biologists typing blindly while talking and eating, too.

Some moves also need you to overcome fears—what if you do not catch the ledge?—and mentally focus on precise motion instead. This does not mean that we do lots of dangerous stuff: sometimes you just know that physically you can do the move, but your brain wants to convince you that it's not going to work; In those situations, mental focus helps to turn

away those false fears. Interestingly, kids often have less such problems with new moves. It's not unlikely that we have unlearned how to be in motion and to trust our bodies, since in town we now use buses and escalators for locomotion, and walk around obstacles as they are thought to be.

Repeat until

Naturally, this gift does not come for free; it is earned with hard—but rewarding—work. Every motion sequence has to be repeated hundreds, if not thousands, of times until it sits perfectly. The same jump can be trained on larger distances, over higher or longer obstacles, to higher or lower levels of elevation, or combined with other moves like a roll, which typically follows a jump to a lower level.

But those variations are already the second step: first, it is important that the move is conducted cleanly and efficiently. There should not be any hits on the joints; they are a sign of wrong balance or posture. It should also feel comfortable and not too hard; chances are that a small tweak can make it much easier if you suspect that too much force is required.

For practising, the usual techniques work. Repetition is important for internalising motions. Some vaults^[1] can be split into smaller parts and can then be trained separately and combined afterwards when both work well. Mental training is great, as it enables re-playing vaults in slow-motion, paying attention to (key) details, and then improving them in reality.

Let's get to some moves in order to examine concrete examples.

Vaults

Speed step A low-energy vault for higher obstacles, also safe on railings. The traceur (the person doing Parkour) keeps on running and just tilts for a short moment to one side, outer leg and hand touching the obstacle. Tweaks are:

- Jump height; as low as possible for efficiency and reduced stress on joints.
- Tilt; the higher the obstacle, the more horizontal. Rotation happens in-flight.
- Arms: they need to push the body back so that the traceur lands in an upright position. →



Speed step



Kong

Kong/Cat One of the most common moves, ideal for obstacles of low to medium height, well-suited for longer distances. For a short time, the traceur flies horizontally over the obstacle, then uses the arms to straighten up and land on the feet again. When jumping, the legs are bent and the arms behind the back in order to gather momentum. Tweaks are:

- Hip height; the higher, the more room for the feet when they are pulled through between the arms.
- Hands position; putting them closer to the rear end of the obstacle helps as well.
- Bent knees before jumping; if they are too stiff on the last two steps, joints up to the hip are strained and less momentum can be gained as ground contact is shorter.

Wallrun An impressive and useful move for reaching the top of a wall; impressive because of the vertical distance that can be overcome. As the name says, the traceur runs on the wall, converting kinetic energy into friction on the wall and vertical acceleration. Tweaks:

- Foot position on the wall; if the wall is slippery, the foot is positioned higher.
- Hands; they do not touch the wall since the muscles are a lot weaker than the leg muscles.
- Speed; more speed means more energy can be converted due to increased pressure against the wall. (Geckos have a workaround for this.)



Wallrun

Shoes

Speaking of geckos: shoes make a difference as well. Harder soles tend to have better grip than very soft ones. Grip is important—but much more important is to feel the ground. Hiking shoes stay flat no matter where you tread. A shoe used for Parkour, on the other hand, should bend when you stand on a railing and clearly tell if you landed on the railing cleanly with the forefeet—or rather with your heels. If the shoe is too hard, Parkour is like telling the difference of how different fabrics feel while wearing a winter glove. Also, size is at least as important. The closer they fit, the less the feet slip around. Otherwise, it feels a bit like running on a slippery carpet.

Currently, OLLO is the only shoe explicitly developed for Parkour, but almost anything you like works. That's a cool feature—no special equipment is required, the shoes are ordinary and even the playgrounds are to be found everywhere. I like this because it is another freedom, compared to so many sports that you can forget about without equipment. Let's nevertheless take a quick look at my shoe shelf.

Running shoe The feathered high heels are useless in Parkour as they make the shoe stiff, at least in the rear half, and just feel unnatural there. Remember: our heels do not have natural padding either. Balancing on railings is harder due to the lacking flexibility. Other moves work fine.



Nike Free Extremely soft shoe that provides good feedback about the ground. You seem to feel every ratchel and bumpiness. I like it for Parkour and usually wear it in the gym. Its drawback is the sole, which, due to its softness, does not grip too well on walls. The harder 3.0 is better than 5.0 which is too soft.



Reebok RealFlex Like the Nike Free, but more stable. Grip is better, feeling on the balls of foot absolutely excellent. Its drawback is the sole that will not survive more than a year of wall runs on rough walls. Ideal summer shoes, the toes hardly sweat due to the net.



OLLO Designed by traceurs and originally funded via Kickstarter. It is harder than the Nike Frees, but the sole has much better grip and should last longer. It is around 1 cm thick and hardly wider than the foot. Sits tight, feels good. For me, it is the only shoe on this list that “just fits”. And they are half as expensive as the other shoes here: \$ 55 only on ollopk.com.



Waldläufer About Reebok’s opposite. An extremely durable shoe—about 10 times older than the photographed Reebok, but in better condition. Not so well-suited for balancing or feathering hard landings due to their stiff soles, but with still better grip than OLLO. I used to lace it tighter for Parkour so that the feet do not slip around in it.



Community

But back to training. As a student, you can learn Parkour via ASVZ with great trainers, three times a week. But what about the summer holidays—sorry: lecture free time? The gym is closed then, but fortunately that is no problem. Parkour does not even need a gym. So we can just train outside in public places. We regularly meet in Zurich and Winterthur to train together and to share tips for improving moves. We train in order to get better, not to be better.

Occasionally, we also train non-related things, like baking cookies together.

For me, Parkour was an ideal balance to studying—sitting the whole day while hacking alone versus training together. Was? Yes, I’m done with studying! Harr :D (I won’t give up Parkour though.) Most likely this is my last article, which ends with the following words: enjoy it (once the first semesters are over) and benefit from ASVZ’s broad range of sports courses. Find the one for you!

✂

Remark

[1] A vault is a way to cross an obstacle. Parkour therefore mostly consists of vaults.

Impressum

VISIONEN

Magazin des Vereins der Informatik Studierenden an der ETH Zürich (VIS)

Ausgabe Dezember 2013

Periodizität 6x jährlich
Auflage 1500

Chefredaktion
Stefan Dietiker
visionen@vis.ethz.ch

Cover
Nicola Maracci Rossi
<http://nmr.io/>

Layout
Judith Meisterhans
Lukas Humbel
Lukas Gisi (Trainee)
Ming Zheng
layout@vis.ethz.ch

Inserate
Frédéric Vogel
inserate@vis.ethz.ch

Anschrift Redaktion & Verlag
Verein Informatik Studierender (VIS)
CAB E31
Universitätsstr. 6
ETH Zentrum
CH-8092 Zürich

Inserate (4-farbig)

| | |
|------------------------------|------------|
| ½ Seite | CHF 850.– |
| ¼ Seite | CHF 1500.– |
| ¼ Seite, Umschlagsseite (U2) | CHF 2500.– |
| ½ Seite, Rückumschlag (U4) | CHF 2500.– |

Andere Formate auf Anfrage.

Lektorat
Clemens Lutz
Simon Eugster

Redaktion
Clemens Lutz
Gregor Wegberg
Jascha Grübel
Judith Meisterhans
Manuel Braunschweiler
Martin Müller
Matthias Geel

und freie Mitarbeiterinnen und Mitarbeiter

Druck
Binkert Druck AG
5080 Laufenburg
<http://www.binkert.ch/>

Copyright
Kein Teil dieser Publikation darf ohne ausdrückliche schriftliche Genehmigung des VIS in irgendeiner Form reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden. Offizielle Mitteilungen des VIS oder des Departements für Informatik sind als solche gekennzeichnet.

© Copyright 1989–2013 VIS. Alle Rechte vorbehalten.

Die Visionen werden klimaneutral gedruckt.



ClimatePartner^o
klimaneutral

Druck | ID: 53213-1312-1005



VERBAND DER STUDIERENDEN AN DER ETH

Der VIS ist Teil des Verbandes der Studierenden an der ETH (VSETH).

Die Welt gemäss Beni Koller

Leidensgemeinschaft

MICHAEL GROSSNIKLAUS – EIN MONUMENT

Carl Miller, der Inspektor, der die Untersuchung des Einbruchs in Benis Zimmer im Bungalow-Hotel leitet, hatte Beni Koller am Tag zuvor am Telefon zu einer Bestandsaufnahme in sein Büro im Hauptquartier der Polizei in der Innenstadt eingeladen. Nachdem ihn Elias in die Stadt gefahren und dort abgesetzt hat, erscheint Beni pünktlich zum ausgemachten Termin. «Es tut mir leid, dass ihre Ferien nun auch noch von mir unterbrochen werden», entschuldigt sich Inspektor Miller. Beni winkt ab und der Inspektor fährt fort: «Es wäre eine grosse Hilfe, wenn Sie uns genau erzählen könnten, was passiert ist.» Carl Miller erklärt Beni, dass es in letzter Zeit immer wieder zu Überfällen auf Hotelzimmer gekommen sei und man es deshalb vermutlich mit einer organisierten Bande zu tun habe. Als der Inspektor mit seinen Ausführungen zu Ende ist, erwidert ihm Beni: «Ich bezweifle, dass diese Theorie auf den Vorfall zutrifft, in dem ich verwickelt war.» Er erzählt Inspektor Miller, wie er Julia im Flugzeug kennen gelernt hat, wie sie nach der Ankunft die Nacht zusammen verbracht haben und wie er am nächsten Morgen unsanft von zwei Schlägertypen über Julias Verbleiben ausgefragt wurde. «Das macht die Sache natürlich komplizierter», seufzt Carl Miller. Die nächsten drei Stunden verbringt Beni Koller damit, dem Inspektor jedes Detail, an das er sich noch erinnern kann, der kurzen gemeinsamen Zeit mit Julia zu schildern. Als dieser endlich zufrieden ist, bittet er Beni: «Ich kann Sie hier nicht festhalten, aber Sie würden mir und der Untersuchung einen grossen Dienst erweisen, wenn Sie vorläufig im Land blieben.» Beni antwortet, dass das kein Problem sei, da er ja eben erst angekommen und sein Rückflug erst in zweieinhalb Wochen sei.

Kurz darauf kann Beni die Polizeiwache verlassen. Als erstes nimmt er sein Mobiltelefon aus dem Plastikbeutel mit seinen Wertsachen und Dokumenten, die er von Inspektor Miller zurück erhalten hat. Zu seiner Erleichterung hat das Telefon noch Akku, so dass er Elias anrufen kann, der ihm seine Nummer vor der Fahrt in die Stadt auf einem Papierzettel zugesteckt hatte. Da Elias beim ersten Mal nicht abnimmt, setzt sich Beni auf eine Bank im kleinen Park vor dem Hauptquartier der Polizei. Die Mittagssonne steht hoch am Himmel und wärmt Benis Körper, während er die Stille im Park genießt, die nur durch das Plätschern eines gusseisernen Springbrunnens durchbrochen wird. Eine Weile lang schaut Beni den Vögeln zu, die aus dem Brunnen trinken und darin baden. Seine Augenlider werden schwer und er ist kurz davor einzunicken, als er eine grummelnde Stimme nicht unweit von ihm vernimmt: «Nun schau sich einer diesen faulen Hund an, er arbeitet nicht und ist dennoch müde!» Beni öffnet die Augen wieder und sieht sich um. Auf der benachbarten Bank hat ein älteres Ehepaar Platz genommen. Da er ausser den beiden der einzige Besucher im

Park ist, muss Beni annehmen, dass die Beschimpfung des Mannes an ihn gerichtet ist. Wütend beginnt Beni aufzustehen, um dem Kerl seine Meinung zu sagen. Doch als er ins Gesicht der Frau blickt, sieht er, wie diese ihn flehend anschaut und den Kopf schüttelt. Zur sichtbaren Erleichterung der Frau hält Beni inne und lässt sich wieder zurück auf die Parkbank fallen.

Wenig später geht eine Gruppe junger Männer durch den Park, die vermutlich auf dem Weg zu ihrer Mittagspause sind. Als sie an der Bank des Ehepaares vorbei laufen, brummelt der ältere Herr zu seiner Frau: «Kannst du riechen, wie die stinken?» Obwohl er sich nicht wirklich Mühe gibt zu flüstern, scheint ihn die Gruppe Männer entweder nicht zu hören oder sie ignorieren ihn bewusst. «Füttere lieber die Vögel und lass die anderen Leute in Ruhe», beschwichtigt ihn seine Frau und fährt fort: «Hier hast du nochmals eine Tüte altes Brot!» Doch das Ablenkungsmanöver verfehlt seine Wirkung. An jedem, der durch den Park geht, hat er etwas zu meckern, und mehr als einmal erntet er dafür einen zornigen Blick. Dass es bei diesen Blicken bleibt und nicht zu Streitereien oder gar Handgreiflichkeiten kommt, kann sich Beni nur mit dem schönen Wetter, dem fortgeschrittenen Alter des Herren und dem gequälten Ausdruck seiner Frau erklären. Als längere Zeit niemand mehr durch den Park gekommen ist, beginnt der ältere Herr plötzlich ein atonales Spottlied anzustimmen, das die bisherigen Ziele seiner verbalen Attacken weiter verunglimpft.

Beni merkt, wie erneut die Wut in ihm hochsteigt. Doch dieses Mal ist er nicht wütend, weil ihn der ältere Herr erneut als Taugenichts beschimpft. Dieses Mal ist er wütend, dass sich dieser Mann überhaupt nicht darum kümmert, was er seiner Frau mit seinem rücksichtslosen Verhalten antut. Noch wütender macht Beni, dass er nicht verstehen kann, welchen Grund es geben könnte, dass sich seine Frau noch immer mit ihm abgibt und ihn nicht längst schon verlassen hat. Beni hat nun grosse Lust, sich in das Leben dieses Ehepaares einzumischen und zu versuchen, der Frau zu helfen, sich von ihrem Mann zu befreien. Allerdings muss er frustriert feststellen, dass er nicht beurteilen kann, ob sie das wirklich will und wie das allenfalls zu bewerkstelligen wäre. Im Minimum, so glaubt Beni, müsste es doch eine gute Idee sein, die Situation weiter eskalieren zu lassen. Damit würde die Frau in eine Lage gebracht, in der sie handeln müsste. Bevor er seinen Plan allerdings in die Tat umsetzen kann, klingelt sein Mobiltelefon. Es ist Elias, der fragt, wo Beni sei und vorschlägt, sich in der Innenstadt zum Mittagessen zu treffen. Nachdem er aufgelegt hat, macht sich Beni auf den Weg zum vereinbarten Restaurant. Beim Vorbeigehen an der Parkbank des älteren Ehepaares wirft er dem Mann einen wütenden Blick zu. In Richtung der Frau ruft er nur: «Herzliches Beileid!»
Fortsetzung folgt ...

ETH Zürich
ETH Bibliothek
Zeitschriften
Rämistrasse 101
8092 Zürich

935
62

AZB
PP/Journal
CH – 8092 Zürich

Falls unzustellbar, bitte zurück an:
Verein Informatik Studierender
CAB E31
Universitätsstr. 6
ETH Zentrum
CH-8092 Zürich

zühlke
empowering ideas

Problem?

Kein Problem: Zühlke löst gerne komplexe Businessprobleme – in den Bereichen Produkt- und Software-Engineering, Managementberatung und Start-up-Finanzierung. Deshalb suchen wir Talente, die lieber den Weg der besten Lösung als den des geringsten Widerstands gehen. Kein Problem für dich? Wir freuen uns auf deine Bewerbung.

zuehlke.com/jobs