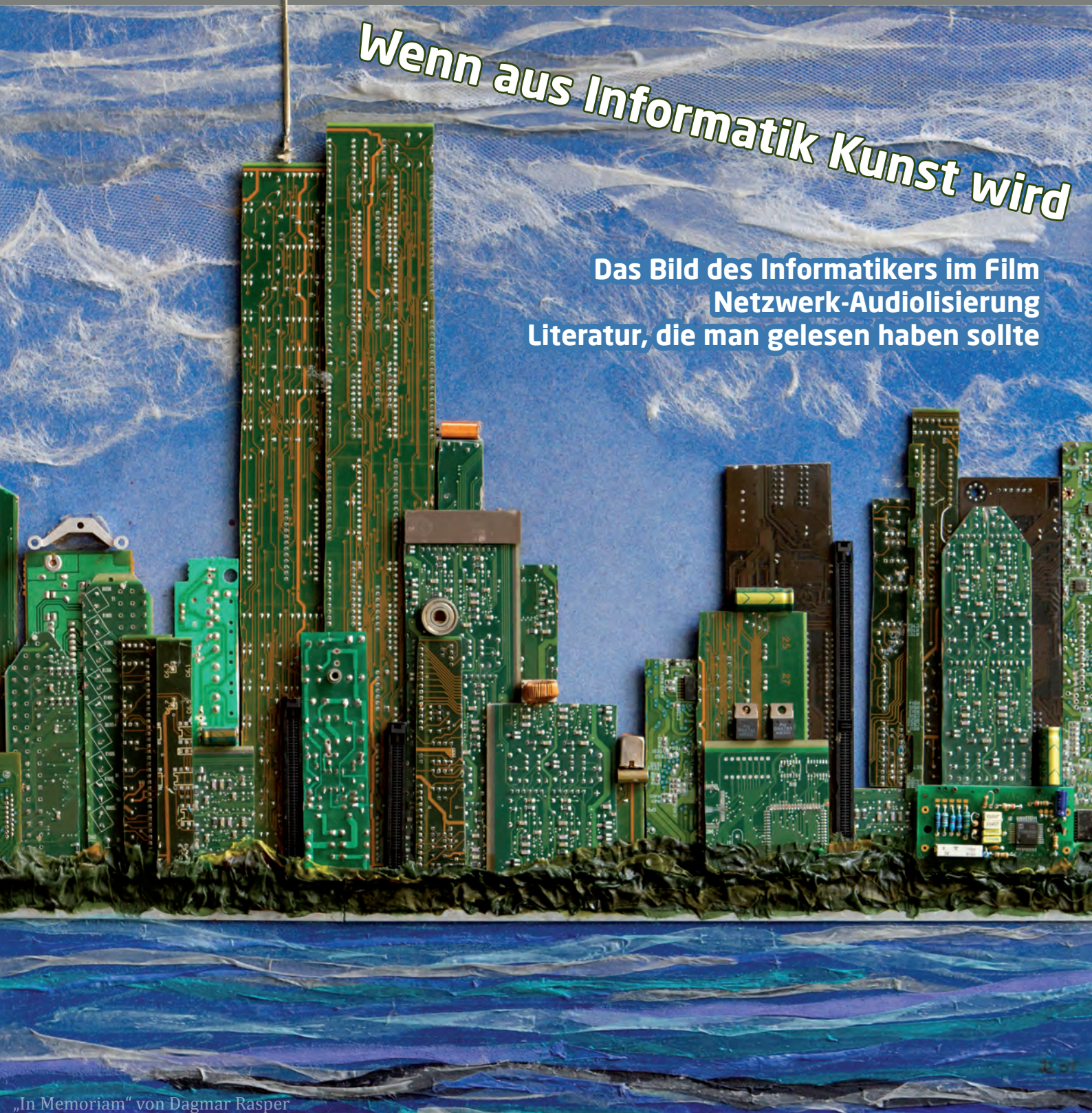
HPImgzn

Hasso-Plattner-Institut

Ausgabe 12 – Sommer 2012



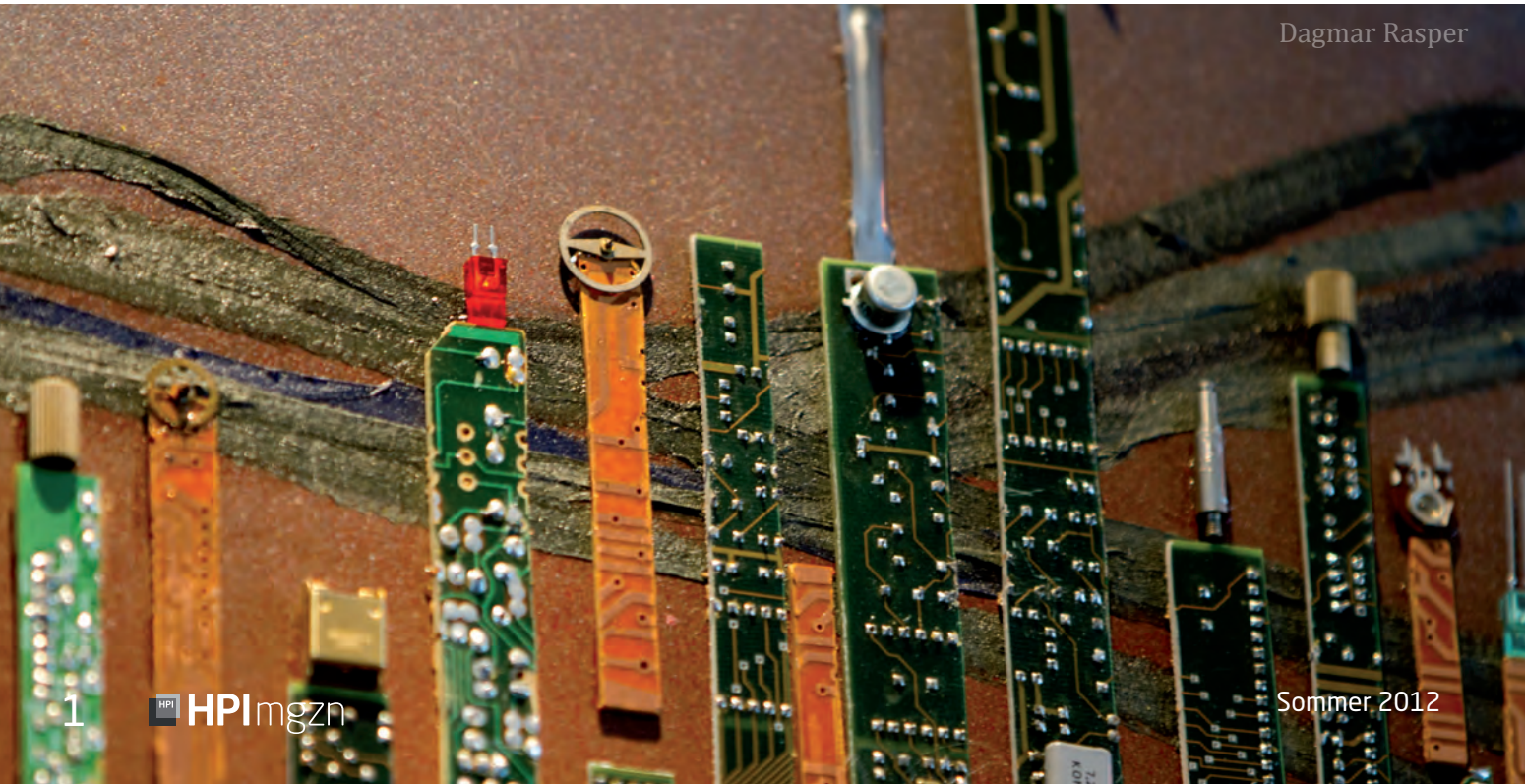
Wenn aus Informatik Kunst wird

Das Bild des Informatikers im Film
Netzwerk-Audiolisierung
Literatur, die man gelesen haben sollte

„In Memoriam“ von Dagmar Rasper

Wenn aus Informatik Kunst wird

Schlaues



Dagmar Rasper

Editorial „Wenn aus Informatik Kunst wird“

Platinen auf Acryl und Stoff

Hoch in den Himmel ragen zwei Türme. Arrogant übertreffen sie die Skyline der New Yorker Küste. Der blaue Himmel zeigt sich voller Wolkenfäden, unter ihm die wogende See, die an der Böschung schäumt. Ein Mahnmal für 9/11. Nicht aus Stein. Auf Dagmar Raspers Collage „In Memoriam“ kleben Platinen auf Acryl und Stoff. Die Künstlerin verwischt die Grenzen zwischen Informatik und Kunst und zeigt, dass Informatik viel mehr sein kann als schnödes Hacken.

Die Lösung eines algorithmischen Problems oder das Basteln eines hübschen Workarounds verlangen neben den theoretischen Kenntnissen Kreativität und Ideen, die über den Tellerrand hinausgehen. Ohne Kunst im Kopf ist Informatik kaum möglich. Wenn sie vielseitig sein soll. Anders. Neu.

Student Jakob Zwiener stellt in diesem HPImgzn seinen Network-Audioliser vor. Ein Programm, das Netzwerk-Übertragungen hörbar macht. Als die Idee in seinem Kopf entstand, lauschte er den Klängen des Solistenensembles Kaleidoskop. Ob und was HPI-Studenten dagegen beim Programmieren hören, zeigt eine Umfrage, die darauf zielt, welche Musik das Programmieren unterstützt.

Doch nicht nur Musik steht beim Titelthema „Wenn aus Informatik Kunst wird“ im Mittelpunkt. Das Medium Film beleuchtet die Gattung der Informatiker in vielseitiger Weise. Sind wir Informatiker Hightech-Helden oder heimlich im Keller sitzende Hacker? Mit einer spritzigen Glosse fühlt das HPImgzn den Informatiker-Klischees in Hollywood-Produktionen auf den Zahn. Wer findet sich darin wieder?

Klischee Nr. 1: Angeblich halten sich übermüdete Informatiker mit Club Mate wach. Ob das Trendgetränk möglicherweise gesundheitliche Risiken birgt, steht auf Seite 29. Wer wach bleibt, sollte neben dem Arbeiten vor dem Bildschirm auch mal zu einem Buch greifen. Wir stellen bedrucktes Papier vor, das uns Informatikern unterhaltsam Utopien und Visionen liefern will. Dazu Fachliteratur, die wir wärmstens empfehlen.

Letztlich sollen studentische Aktivitäten natürlich nicht zu kurz kommen. In der 12. Ausgabe des Magazins setzen wir unsere Reihe über Open-Source-Projekte am Institut fort und präsentieren den ersten Teil des Erlebnisberichts zum „Google Summer of Code“.

Wir wünschen nun viel Freude beim Lesen der zwölften Ausgabe des HPImgzn.

— Carolin Fiedler und Franz Liedke

Helden, Playboys und Nerds

Das Bild des Informatikers im Film

Aus Sicht des gemeinen Menschen ist der Informatiker – im Volksmund auch als Nerd bezeichnet – eine anstrengende Spezies. Das liegt zum Teil an seiner nicht unerheblichen Intelligenz. Die macht es schwer, mit ihm Gespräche zu führen, ohne Gefahr zu laufen, früher oder später auf dem Schlauch zu stehen. Allzu leicht ist man auch verwirrt vom unkonventionellen Kleidungsstil des Informatikers, seinem aufdringlichen Geruch oder der Angewohnheit, geduckt zu laufen. Der Nerd ist eine Lebensform, die ihrer Welt entwachsen ist. Allzu gewöhnliche Dinge wie die Wahl der Kleidung oder die Körperhygiene überträgt er Mutti oder lässt sie einfach ganz weg. Er trägt eine dickrandige Brille – aus Notwendigkeit, weil das Sitzen vor dem Computer die Augen schädigt – und eine Zahnspange. Der Nerd ist sozial unfähig. Er unterhält sich auf klingonisch, weil er zu wenig Sex bekommt. Und er trinkt immer nur Cola, nie Bier. Ein Nerd verträgt keinen Alkohol.

Intelligenz hat ihren Preis: der typische Nerd

Alles Lüge? Wer das HPI regelmäßig besucht, weiß das. Jeder mag den einen oder anderen „echten" Nerd kennen, aber man kann nicht behaupten, dass wir alle so wären. Also woher kommt dieses falsche Bild? Wer oder was treibt die Leute dazu, vorbehaltlos Dinge zu glauben, die dann zum Vorurteil über eine ganze Berufsgruppe werden? Na klar – die Medien.

Die Frage, inwiefern die Medien den Ruf der Informatiker bleibend schädigen, sollte uns alle interessieren. Wer den Feind nicht kennt, kann ihn nicht bekämpfen. Insbesondere die Darstellung von Informatikern in Spielfilmen ist interessant. Diese erfreuen sich einer gesteigerten Beliebtheit, seit sie in 3D auf der Leinwand erscheinen. Ein Film kann Menschen verändern, ihnen angeblich sogar einen neuen Lebensinhalt geben. Außerdem ist Heimkino eine erfreuliche Alternative zur üblichen redaktionellen Recherchearbeit. Im Folgenden wird anhand einiger der bekannteren Filme über Informatiker ein fundierter Beitrag zur Aufklärung *skandalöser Medienverbrechen* geleistet. Es ist mit Spoilern zu rechnen.

Wenn sich Filme mit Informatik beschäftigen, dann geht es meistens um Hacker oder um aggressive künstliche Intelligenz. Das scheinen die beiden Disziplinen zu sein, von denen die Regisseure sich den größten Ertrag erhoffen. Sie sind schön reißerisch: Ein Informatiker ist entweder damit beschäftigt, dein Bankkonto zu knacken, oder den Untergang der Menschheit herbeizuprogrammieren. Das klingt spannend und alles andere als nerdy. Das manische Lachen passt einfach nicht zum Hänfling mit der Zahnspange.

Informatiker retten die Welt vor den Maschinen…

Informatik hat Potenzial. Informatik ist wichtig für den technologischen Fortschritt der Menschheit. Dass das auch nach hinten losgehen kann, ist kein Geheimnis. Die Angst vor der künstlichen Intelligenz, die außer Kontrolle gerät, kennen wir spätestens seit Terminator. Wer kann helfen, wenn es

zum Worst Case kommt? Ganz klar: derjenige, der den Mist verzapft hat – oder seinesgleichen. Wir als Informatiker werden also auch in Zukunft gefragte Arbeitskräfte sein.

Zwei der bekanntesten Filme, in denen Informatiker die Hauptrolle spielen, befassen sich mit dieser Problematik. Gemeint sind *Tron* (1983) und *Matrix* (1999). Die Protagonisten, Flynn in Tron und Neo in Matrix, sind beide studierte Informatiker. Der eine hat seinen Beruf aufgrund eines inkompetenten Vorgesetzten verloren. Der andere arbeitet bei einer Software-Firma, wird dort aber offenbar nur schlecht bezahlt und muss sich nachts als Hacker etwas dazuverdienen. Das ist so ungefähr das Schicksal, das einen gewöhnlichen Informatiker auch im wirklichen Leben einmal treffen kann. Ein Leben für den Beruf, unverstanden von den Chefs. Für Freizeitaktivitäten bleibt nur die Nacht, sofern man nicht mit dem Kopf auf der Tastatur einschläft. In beiden Filmen ist es mit dem normalen Leben allerdings zu Ende, als der Protagonist in eine Parallelwelt eintritt, die von Maschinen regiert wird. Gekonnt werden dort Armeen von bösartigen Programmen niedergemacht und damit die Realität gerettet. Beide Figuren sind jedoch grundverschieden. Flynn ist ein von sich selbst überzeugter Weiberheld, Neo ein eher ruhiger Typ, der erst lernen muss, an sich selbst zu glauben. Doch keiner von ihnen ist ein Klischeenerd. Beide sind attraktiv, kommen ohne Sehhilfe aus, können problemlos mit Frauen reden und haben auch sonst keine sozialen Defizite. Keiner von ihnen ist unsportlich – im Gegenteil. Wie viele Informatiker sind schon Großmeister des Martial Arts? Speziell Matrix gelingt es ausgezeichnet, den Nerd mit einem Actionhelden zu verwechseln. Kung-Fu macht einfach mehr her als stundenlange Hack-Sessions vor dem PC – vor allem mit Ledermantel und Sonnenbrille.

Der Informatiker, der die Welt vor der bösen künstlichen Intelligenz rettet, findet sich auch im Film *WarGames – Kriegsspiele* (1983). Es handelt sich diesmal um den technikbegeisterten Schüler David. Dieser Junge bricht, ohne es zu wollen, in einen intelligenten Militärrechner ein, für den Atomkrieg und Spiel dasselbe sind. David ist naiv genug, die Maschine zu provozieren und ist den Rest des Films damit beschäftigt, die drohende Eskalation des Kalten Krieges aufzuhalten. Beschrieben wird der Protagonist als intelligent aber dennoch schlecht in der Schule. Er verbringt seine Freizeit lieber in der Spielhalle und mit seinem Computer als mit Schulbüchern und Hausaufgaben. Auch hat er nicht viele Freunde. Nichtsdestotrotz hat er eine feste Freundin und sein Auftreten ist das eines ganz normalen Jungen. Also wieder kein echter Nerd. Schade, dabei hatte er so gute Chancen. Im Gegensatz zu Tron und Matrix ist WarGames darum bemüht, auf dem Boden der Tatsachen zu bleiben. Das gelingt ihm nicht ganz. Speziell die Inkompetenz der IT-Fachleute des Verteidigungsministeriums ist alles andere als glaubwürdig. Denen gelingt es schließlich nicht einmal, ein Passwort zu ändern, ohne das System dabei funktionsunfähig zu machen. Man kann den Regisseur verstehen. Immerhin fallen die Einspielergebnisse offenbar umso

Wer die blaue Pille nimmt, ist ein Schlappschwanz.

magerer aus, je näher sich die Handlung am Boden der Tatsachen bewegt. Das Szenario von WarGames ist dennoch nichts im Vergleich zu aufwendig computeranimierten alternativen Realitäten, was wohl der Grund für den geringen Bekanntheitsgrad des Films sein wird.

Der Irre und die Frau - gefährliche Hacker aus ‚Hackers'

... und vor machtbesessenen Schurken

Wir wissen aus Filmen, dass die meisten Informatiker Hacker sein müssen. Diese Disziplin scheint *total* überlaufen zu sein. Hacker sind gut darin, verrückt gewordene Maschinen aufzuhalten. Aber sie können es jederzeit auch mit gemeinen menschlichen Widersachern aufnehmen. Es bedarf nur einer Handvoll von diesen Leuten, um mächtige Monopole zu Fall zu bringen. Und dabei sehen sie so unverschämt gut aus! Wer möchte da nicht gern ein Hacker sein?

Hackers – Im Netz des FBI (1995) ist das Paradebeispiel eines unauthentischen Films über Informatiker. Das Thema: Gute Hacker bekämpfen bösen Hacker und umgekehrt. Bei den guten Hackern handelt es sich um eine Horde Studenten, alle computerbegeistert. Einer hat beispielsweise bereits mit elf Jahren einen Killerwurm geschrieben. Sie hacken aus Spaß einen Supercomputer und legen sich dabei mit dem Sicherheitschef eines Mineralölkonzerns an. Der übernimmt die Rolle des bösen Hackers. Er hat einen fiesen Virus programmiert und will damit ein paar Schiffe versenken, um reich zu werden. Einen Nerd findet man hier nicht. Der Hauptcharakter gibt zwar zu, mit 18 noch Jungfrau zu sein, aber das liegt wohl daran, dass er nie dazu kam, sich mit Frauen zu beschäftigen. Das ändert sich, denn ausgerechnet der beste Hacker in der Gruppe ist eine Frau, gespielt von Angelina Jolie. Ein Hacker mit vollen Lippen – wen würde das nicht begeistern? Alle Beteiligten sind natürlich Genies, gewissermaßen Übermenschen. Sie können Großrechner, Ampelsteueranlagen oder Telefonverbindungen des Secret Service innerhalb von Minuten knacken. Bei einigen von ihnen lassen sich allerdings ernste geistige Störungen identifizieren. Das trifft vor allem auf den bösen Sicherheitschef zu, der keine Gelegenheit auslässt, kindische Sprüche zu reißen. Er fährt auch auf einem Skateboard zu Erpressungen und geheimen Treffen. Sportlich sind die Hacker ohnehin alle. Das bevorzugte Fortbewegungsmittel sind Inliner.

Ein weiterer verzerrter Film über Computerkriminelle ist *Passwort: Swordfish* (2001). Er lehrt uns vor allem eines: Hacker sind cool. Dieser Film trägt sogar noch etwas dicker auf als Hackers. Ironischerweise beschwert sich John Travolta noch im Anfangsmonolog darüber, wie wenig Authentizität in Hollywoods Machwerken steckt. Doch wie authentisch kann ein Film sein, in dem Hugh Jackman die Rolle des gefährlichsten Hackers Amerikas einnimmt? Der Mann, der in X-Men Wolverine verkörpert hat, soll nun für eine Terroristenvereinigung das System einer Bank knacken. Der Zuschauer bekommt, was er erwartet: ein Superhirn mit krassen Bauchmuskeln, einen Playboy, der sich öfter auf Partys als mit dem Computer vergnügt. Es beginnt damit, dass unser Hacker eine FBI-Datenbank knackt, während er gerade Sex und eine Waffe am Kopf hat. Er programmiert einen Superwurm, während er flaschenweise Wein trinkt, sinnfreie Gespräche mit sich selbst führt und ab und an lustvoll stöhnt. Statt Codezeilen zu schreiben, setzt er dabei blaue 3D-Gebilde zusammen. Schließlich ist er noch in etliche Verfolgungsjagden involviert – Bürojobs sind was für Versager. Dieser Film ist göttlich, zumindest für jeden, der über gewisse Informatikkenntnisse

Wolverine als Golf spielender Informatiker

verfügt. Auf die Frage, wie er die FBI-Datenbank geknackt hätte, antwortet Hugh Jackman beispielsweise mit: „Used a logic bomb, dropped it through the trapdoor.“ Sein Ton klingt dabei sehr ernst. Seine Konsole trägt übrigens den Titel ‚Compiler‘.

Es geht auch anders

Aber es gibt authentische Filme über Informatik. Die sind selbstverständlich nicht so unfreiwillig komisch, was nicht heißen soll, dass sie schlecht sind. Zumindest bekommt man einmal nicht den Eindruck, dass die Macher sich statt eines IT-Beraters lieber ihrer blühenden Fantasie und eines Bullshit-Bingo-Heftes bedient haben.

Startup (2001), der im Englischen den sehr viel passenderen Titel „Antitrust“ trägt, schlägt hier eine Brücke. Es sind keine gravierenden sachlichen Fehler zu bemerken (von der falschen Benutzung privater IP-Adressbereiche einmal abgesehen), und doch klingt das gesamte Szenario ein wenig nach modernen Verschwörungstheorien. Es geht um einen Konzern, der Telekommunikationssoftware entwickelt und sich dabei hinterhältig der Ideen der Open-Source-Gemeinde bedient. Dazu wird jeder der Entwickler rund um die Uhr kameraüberwacht und, wenn nötig, umgelegt. Milo, ein genialer junger Programmierer wird von diesem Konzern engagiert, um bei der Fertigstellung eines Projektes zu helfen. Als kurz darauf sein bester Freund Opfer der brutalen Geschäftsmethoden wird, beginnt sein Kampf gegen den eigenen Arbeitgeber. Der Protagonist wirkt wie das Klischee eines Jura-Studenten, was Aussehen, Auftreten und Freundin betrifft. Er ist jedoch auch jemand, der die Nächte vor dem Rechner verbringt, um Bugs im eigenen Code zu beheben. Das macht ihn glaubwürdig. Ungefähr so sieht der perfekte Informatiker aus: Jung, gut aussehend, intelligent und sowohl privat als auch beruflich erfolgreich. Leider hat Milo einen Hang zum Übermenschlichen. Er braucht nur fünf Sekunden auf eine Seite Quellcode zu starren, um diesen sofort als genial beurteilen zu können. So schnell kann niemand lesen. Doch wir wollen ihm das verzeihen. Der Film wäre schnell langweilig geworden, hätte man versucht, solche Verstehensprozesse in Echtzeit darzustellen.

Wer jedoch einen realistischen Film über Hacker sehen will, dem sei *23 – Nichts ist so wie es scheint* (1998) empfohlen. Er wurde in Deutschland produziert und befasst sich mit dem KGB-Hack, der gegen Ende des Kalten Krieges Aufmerksamkeit erregte. Erzählt wird aus Sicht des beteiligten Hackers Karl Koch. Der ist nach dem Tod seines Vaters besessen von den Illuminaten und findet Gleichgesinnte im Chaos Computer Club. Seine politische Einstellung und später auch seine Drogensucht veranlassen ihn dazu, für die Russen in IT-Systeme der westlichen Wirtschaft einzubrechen. Er wird zum Vollzeithacker und findet schließlich ein tragisches Ende. Im Gegensatz zu den bisher betrachteten Protagonisten ist Karl niemand, in dessen Haut man gern stecken möchte. Es fällt schwer, diesen Jungen zu mögen (was zum Großteil auch am Gesichtsausdruck des Schauspielers liegt). Er ist sozial unangepasst und kann nicht wirklich mit Frauen umgehen. Seine große Liebe vergrault er, ihm bleibt ein einziger Freund. Dennoch ist auch er nicht der typische Nerd, schließlich ist für ihn das Hacken nur Mittel zum Zweck. Er gibt sich der Computerkriminalität hin, um seine politischen Überzeugungen durchzusetzen und seinen Drogenkonsum zu finanzieren. Wirklich intelligent ist er ebenfalls nicht. Stattdessen profitiert er von seinem Einfallsreichtum und einem Haufen schlecht gesicherter Unternehmensnetzwerke.

Der eine Filmnerd: Jesse Eisenberg als Mark Zuckerberg

Das Beste, was Hollywood in den letzten Jahren an Filmen über Informatik zustande gebracht hat, ist *The Social Network* (2010) – der Facebook-Film. Es geht natürlich um Mark Zuckerberg und die Entstehung unserer Lieblingswebsite. Inwiefern die Handlung auf Tatsachen basiert, lässt sich schwer feststellen. Doch der Umstand, dass der Film von Prof. Polze in seiner Betriebssystemvorlesung als ergänzendes Lehrmaterial herangezogen wird, sollte einige Vorbehalte wegen mangelnder Authentizität beseitigen. Der Film ist unterhaltsam, nicht übertrieben und man kann sogar noch etwas Fachliches lernen. Das Beste ist: Hier haben wir einen Nerd.

***The IT-Crowd*. Ein echter Nerd ist auch in schlechter Auflösung unverkennbar**

Herr Zuckerberg ist hochintelligent, gleichgültig, was sein Äußeres betrifft, und hat Probleme damit, mit Anderen zu kommunizieren, ohne sie zu beleidigen. Seine Freundin wird er daher gleich zu Beginn des Films los. Er bemüht sich zwar redlich, dies wieder geradezubiegen, doch das Weibsvolk hat lediglich Interesse an seinem Ruhm und seinem Reichtum. Ähnlich ergeht es ihm mit seinen Freunden. Der Protagonist leidet unter seiner eigenen rationalen Art, die ihn daran hindert, glücklich zu werden.

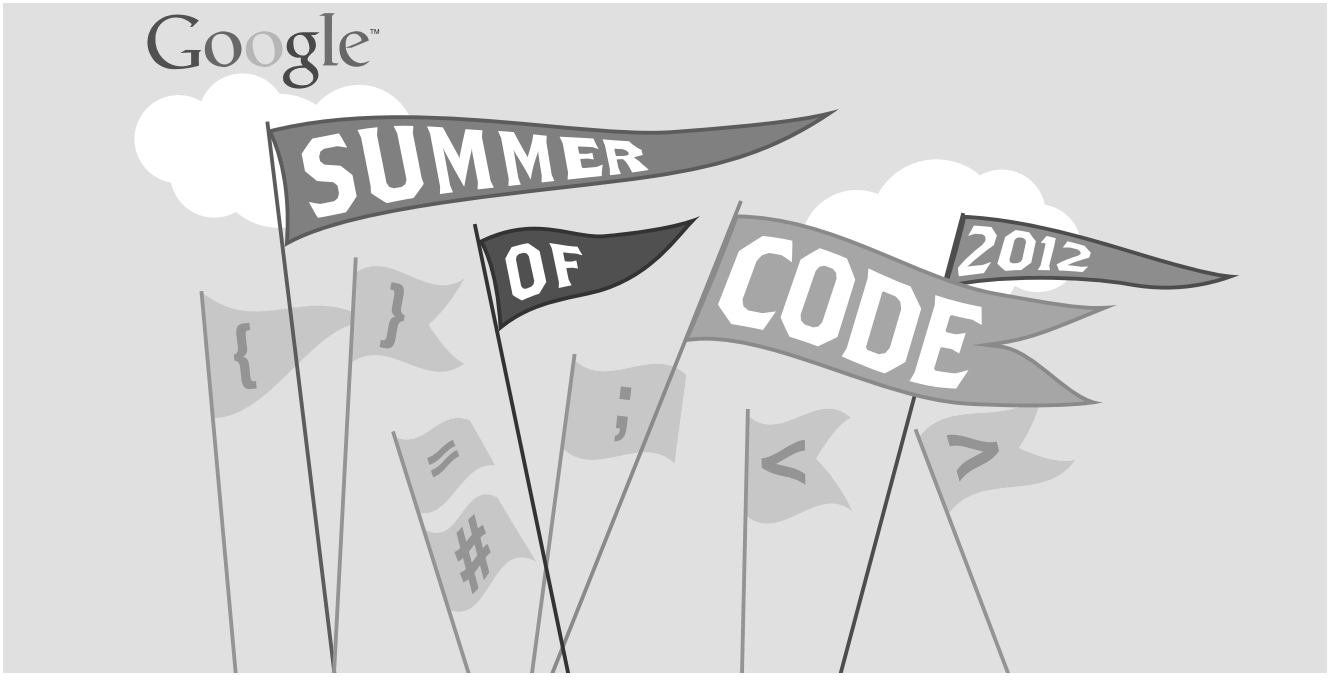
Auch, wenn wir schließlich doch noch einen Film gefunden haben, dessen Hauptrolle ein waschechter Nerd ist, so steht doch fest, dass dieses Bild in Spielfilmen alles andere als üblich ist. Hollywoods Informatiker sind fast immer Helden, Sexsymbole, Übermenschen oder (im Fall des bösen Informatikers) Psychopathen. Sie sind der Traum jedes Mädchens und das große Vorbild jedes Jungen. Die wenigen Ausnahmen spielen meist Nebenrollen und bleiben nicht lange in Erinnerung. Das Kino ist nicht schuld am Klischee des Nerds. Wer dann?

Es fällt nicht schwer, einen anderen Sündenbock zu finden: das Fernsehen. Serien wie *The Big Bang Theory* und *The IT-Crowd* oder zweifelhafte Dokumentationen über die GamesCom tun ihr Bestes, um das zuvor beschriebene Bild des wahren Nerds unter das Volk zu bringen. Der Fernseher wird ungleich stärker frequentiert als das Kino, weil das Vergnügen preiswerter ist und er direkt im Wohnzimmer steht. Wer sich nach einem langen Arbeitstag seine Lieblingssendung ansieht, ist oft zu erschöpft, um noch über das Gesehene nachzudenken. Er nimmt es einfach hin. Und was er sieht, frisst sich in sein Gehirn, nistet sich dort ein und geht nicht mehr raus. Wer Informatik studiert, ist ein Nerd. Wer Informatik studiert, ist bemitleidenswert. So etwas macht sich auch gut am Stammtisch. Da fühlen sich alle besser, das hebt die Stimmung. Also gibt es keinen Grund, zu hinterfragen.

Ich danke Hollywood daher für ein verzerrtes Bild des Informatikers. Im Film sind wir Helden und retten die Menschheit. Das bringt das Weltbild des unwissenden Volkes zum Wanken und wir haben etwas zu lachen.

— *Jan Teske*

Sommercode



Oft hört man von ihm: Dem Google Summer of Code (GSOC). Seit mittlerweile sieben Jahren lädt Google jedes Jahr Studenten aus aller Welt dazu ein, ihre Freizeit im Sommer für einen guten Zweck in ein Open-Source Projekt zu investieren. Was genau verbirgt sich hinter dem Summer of Code, warum lohnt es sich daran teilzunehmen und welche Schritte sind für eine Teilnahme nötig?

Ob Python, Hadoop oder JavaScript: Jeder, der sich schon einmal genauer mit diesen von Google tagtäglich eingesetzten Technologien auseinander gesetzt hat, wird festgestellt haben, dass diese größtenteils in Open-Source-Communitys entwickelt werden. Sogar Android, Googles Smartphone-Betriebssystem, wird mit offenen Quellen ausgeliefert. Schon lange unterhält Google ein sehr enges Verhältnis zur Open-Source-Gemeinde und fördert diese entsprechend intensiv. Ein Eckpfeiler dieses Bestrebens von Google ist der Summer of Code, der im Jahre 2005 zum ersten Mal ins Leben gerufen wurde. Die Kernidee ist dabei, Studenten dazu zu bewegen, ihre über den Sommer (Ende Mai bis August) oft freiliegende Zeit für Beiträge in der Entwicklung von Open-Source-Projekten zu nutzen. Dabei wird jeder Student von einem Mitglied aus dem jeweiligen Open-Source-Projekt begleitet, was auch als „mentoring“ bezeichnet wird. Zudem stellt Google für jeden Teilnehmer ein Stipendium in einer Höhe von 5 000 USD aus, sodass sich jeder Student vollständig auf das jeweilige Projekt konzentrieren kann. Insgesamt unterstützt Google so alle aufgenommen Projekte pro Jahr mit ca. sechs Millionen Dollar, womit von allen Beteiligten zusammen etwa drei Millionen Zeilen Code innerhalb von drei Monaten geschrieben werden. Damit stellt der GSOC eine der effizientesten (zwei Dollar pro Codezeile) Produktionsstätten von Code in der Softwareindustrie dar.

Neben dem Stipendium winken einem teilnehmenden Studenten aber noch einige weitere Vorteile: So hat eine nachgewiesene Teilnahme am GSOC auch in einer Bewerbung mittlerweile immer mehr Gewicht. Google verwendet beispielweise die Resultate des GSOC jedes Jahr, um potenzielle Mitarbeiter oder Praktikanten ausfindig zu machen. Aber auch bei vielen anderen Firmen berichten Mitarbeiter, dass sie allein durch eine Teilnahme beim GSOC an den jeweiligen Job gekommen sind. Abgesehen von besseren Karrieremöglichkeiten gibt es natürlich auch den heiß begehrten Summer of Code-Notizblock, ein offizielles Teilnahmezertifikat und nicht zu vergessen: die berühmten T-Shirts,

die jedem HPI-Studenten dank eines gewissen Dozenten gut bekannt sein sollten.

Wie sollte man sich als angehender Teilnehmer also an den GSOC heranwagen? Es gibt mehrere Phasen des GSOC: Die erste ist die Kickoff-Phase, bei der nur die Bekanntmachung von Google herausgegeben wird, dass der GSOC nun offiziell begonnen hat. Anschließend folgt über den März der Zeitraum, in dem sich Open-Source-Projekte selbst bei Google bewerben, um in das Programm aufgenommen zu werden.

Danach folgen die Phasen, die für uns Studenten von Belang sind: Die Bewerbungsphase, die „Community bonding phase“ und der Programmierzeitraum, welcher sich in zwei Teile aufteilt.

In der Bewerbungsphase sollten Studenten heraussuchen, bei welchen Projekten sie partizipieren wollen. Man kann nur zu einem Projekt zugelassen werden, aber durchaus Bewerbungen für mehrere Projekte einreichen. Zudem sollte man sich überlegen, ob man einen eigenen Vorschlag für eine Feature-Erweiterung einbringen möchte oder lieber einen von den jeweiligen Projekten präsentierten Vorschlag aufgreift. Bei ersterem ist sehr viel mehr Planung und Erfahrung seitens des Studenten notwendig, wohingegen man bei letzterem schnell über Mitglieder in der jeweiligen Community nähere Informationen erlangen kann und zugleich mögliche Mentoren kennenlernt.

Es ohnehin sinnvoll, sich bereits in der Community bekannt zu machen, da dies die Chance auf eine Annahme verbessern kann. Anschließend muss der Student (unabhängig davon, ob er einen eigenen Vorschlag oder ein angebotenes Projekt implementieren möchte) ein Proposal verfassen. Dessen Struktur ist sehr frei, verschiedene Projekte machen aufgrund der Vielzahl von Bewerbern aber formale Vorgaben, um einen einfachen Zulassungsprozess zu ermöglichen.

Das Proposal sollte zudem ausschließlich vom Studenten selbst verfasst werden, da es auch dazu verwendet wird, die technische Kompetenz des Bewerbers einzuschätzen und festzustellen, ob er überhaupt in der Lage ist, das Projekt zu bewältigen.

Der Zulassungsprozess selbst wird von jedem Open-Source-Projekt eigenständig abgewickelt. Dabei werden alle eingesendeten Proposals gelesen und entsprechend einer Vielzahl von Parametern priorisiert. Zudem hat jedes Projekt eine von Google vergebene Kapazität, also eine maximale Anzahl von Studenten, die akzeptiert werden dürfen. Diese korreliert meistens auch mit der Größe des jeweiligen Open-Source-Projektes. Letztlich werden der Kapazität entsprechend die obersten Proposals aus der Liste entnommen und die Studenten zugelassen.

Nachdem man bis Mitte April das Proposal eingesendet hat, muss bis Ende April gewartet werden, bis man direkt von Google die Bestätigung zur Aufnahme in das Projekt bekommt. Auch bei einer Ablehnung geht die Welt natürlich nicht unter, denn es werden jedes Jahr nur rund 20% der eingereichten Bewerbungen angenommen. Viele erfolgreiche Teilnehmer des GSOC haben es teilweise erst beim dritten Anlauf geschafft, in das Programm aufgenommen zu werden.

Sollte man sich also nach dem Bewerbungsprozess in der von Google herausgegebenen Liste wiederfinden, sind alle Hürden für den GSOC genommen. Was nun folgt, ist die sogenannte Community bonding phase: Hier lernen die Studenten ihre Mentoren kennen – sofern dies nicht schon geschehen ist – und sie arbeiten sich in die Prozesse ein, mit denen innerhalb der jeweiligen Community an Projekten gearbeitet wird. Zudem muss von den Studenten eine Studienbescheinigung und eine spezielle Tax-Form, die von Google ausgestellt wird, unterschrieben werden.

Ende Mai beginnt die Programmierphase und Google gibt die erste Auszahlung in Höhe von 500 USD an die Studenten aus. Dies geschieht bisher, indem Google jedem Student eine Kreditkarte zusendet und diese mit den entsprechenden Beträgen auflädt. Zu bedenken ist dabei, dass das Konto in Amerika liegt und beim Nutzen des Geldes internationale Überweisungskosten anfallen. Diese belaufen sich auf etwa 3 % des Überweisungsbetrages inklusive 3,50 USD Fixkosten. (Stand 2012) Google zahlt aus diesem Grund meist noch etwas mehr (5-10 $) auf die herausgegebenen Beträge, um diese Nebenkosten auszugleichen.

Der erste Teil der Programmierphase läuft bis Anfang Juli, bis die Midterm Evaluation stattfindet. Dabei bewerten die Mentoren den bisherigen Fortschritt des Studenten beim Projekt und teilen

Google anschließend mit, ob sie eine Weiterführung des Projektes für sinnvoll halten. Dabei sei Folgendes erwähnt: Für HPI-Studenten und auch viele andere internationale Teilnehmer des GSOC liegt die erste Hälfte der Programmierphase recht ungünstig, also Juni bis Juli, wenn in der Regel noch Vorlesungen gehalten, aber auch sehr bald Klausuren bzw. die Bachelorarbeit geschrieben werden. Die Wahl des Zeitraumes orientiert sich dabei an den Studienzeiten von US-Amerikanischen Studenten, deren Studienzeit in Quarters unterteilt ist. Während im Fall, Winter und Spring Quarter studiert wird, hat im Summer Quarter der Student meist komplett frei und Zeit zur Absolvierung von Praktika oder anderen Aktivitäten. Obwohl Google dieses Problem schon seit langem bekannt ist, hat man sich nicht dazu entschlossen, den Programmierzeitraum umzulegen. Dies liegt offensichtlich daran, dass man die unzähligen verschiedenen terminlichen Bedürfnisse von internationalen Studenten ohnehin unmöglich befriedigen könnte. Es bietet sich daher für HPI-Studenten an, einen Teil der Programmieraufgaben schon in der Community bonding phase durchzuführen, sodass man sich über die Klausurzeit einen Puffer vorhalten kann. Entsprechend haben die meisten Mentoren auch Verständnis für die terminliche Problematik, man sollte ihre Geduld aber auch nicht überstrapazieren.

Ist die Midterm Evaluation erfolgreich durchlaufen, werden weitere 2 250 USD des Stipendiums ausgezahlt und der verbleibende Rest am Ende der zweiten Programmierphase. Diese Auszahlung ist allerdings auch an das Bestehen der Final Evaluation gebunden. Gegen Ende des Projektes hat Google eine sehr strikte „Pencils Down“ Deadline, die unbedingt einzuhalten ist. Aus diesem Grund wird auch empfohlen, das Projekt – wenn möglich – bereits eine Woche davor abgeschlossen zu haben. Man sollte die Zeit also entsprechend gut durchgeplant haben (was allerdings auch schon während des Schreibens des Proposals in der Vorbereitungszeit geschehen sein sollte).

Zusammengefasst ist der GSOC eine hervorragende Gelegenheit, Erfahrung in professioneller Softwareentwicklung zu gewinnen und Ratschläge von einer Vielzahl sehr begabter Programmierer zu erhalten. Zumal die Interaktion mit einer Open-Source-Community auch eine gute Schärfung der am HPI viel besagten Soft-Skills ist, da man stets um eine strukturierte und höfliche Umgangsform bedacht sein muss.

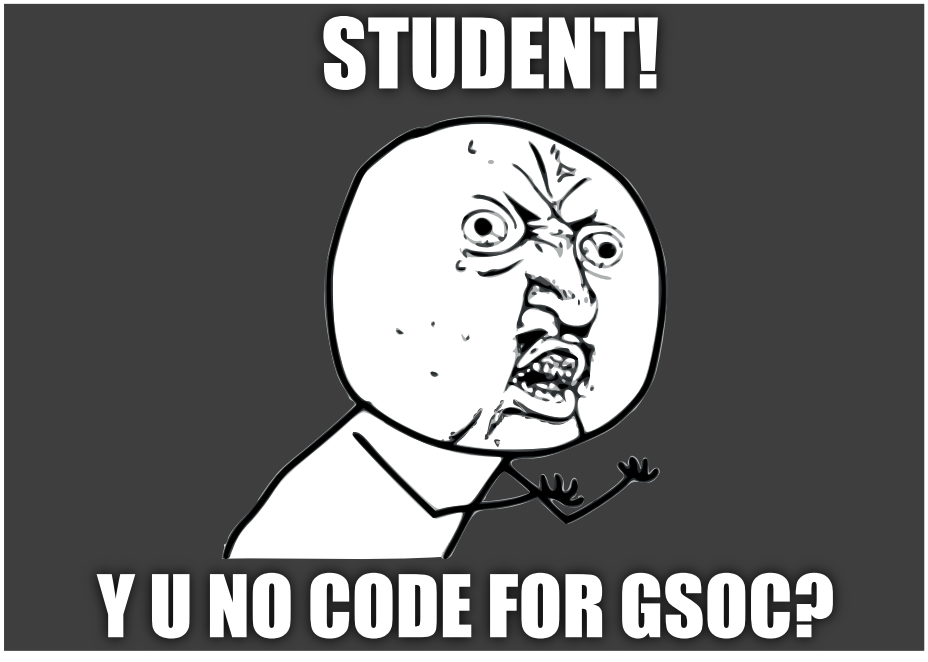
Studenten, die mehr über den GSOC erfahren möchten, können im Internet unter den unten aufgeführten Adressen weitere Informationen finden.

— *Robin Schreiber*

Informationen

Aktuelle GSOC Webseite:
http://www.google-melange.com/gsoc/

Inspirierender Bericht eines ehemaligen GSOC Studenten:
http://google-opensource.blogspot.de/2012/02/road-not-taken-adventures-of-post.html



Netzwerk-Audiolisierung

Eine akustische Repräsentation von Netzwerkdaten

Die Idee für den Network-Audioliser habe ich in einem Konzert des Ensembles Kaleidoskop im Kammermusiksaal der Philharmonie bekommen. Das Ensemble hat Musik von Iannis Xenakis gespielt, darunter auch einige Werke, die der generativen oder stochastischen Musik zugeordnet werden können. In der generativen Musik werden Algorithmen genutzt, um Musikstücke zu erzeugen. Meist werden diese Algorithmen oder Reihen zu dem Zweck der Generierung von Musik geschrieben. Computer wurden dazu von Xenakis mindestens seit den 1960er Jahren eingesetzt.

Wenn es Algorithmen gibt, die Musik generieren, dann kann man sich vielleicht auch bestehende Algorithmen anhören. Diesen Ansatz verfolgt der Network-Audioliser, ein Programm, das Netzwerkübertragungen hörbar macht. Hierzu werden für jedes Paket Midi-Events erzeugt. Für große Pakete werden tiefe Töne und für kleine Pakete hohe Töne generiert.

Schon diese sehr einfache Abbildung ermöglicht es Hörern der Audiolisierung, gewisse Aussagen über die Netzwerkübertragung zu treffen. ARP-Pakete führen zum Beispiel zu sehr hohen Tönen. Die Anzahl dieser Töne pro Sekunde lässt somit bestimmte Rückschlüsse auf die Anzahl der Rechner im Netzwerk zu.

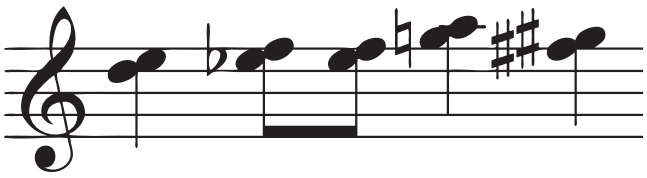


Abbildung 1: Manche Netzwerkprotokolle erzeugen charakteristische Akkordfolgen. Gut herauszuhören ist z.B. SSDP.

Manche Protokolle lassen sich sogar an charakteristischen Motiven erkennen. So erzeugt zum Beispiel SSDP regelmäßig die obenstehende Akkordfolge, die auch in stark frequentierten Netzwerken leicht herauszuhören ist.

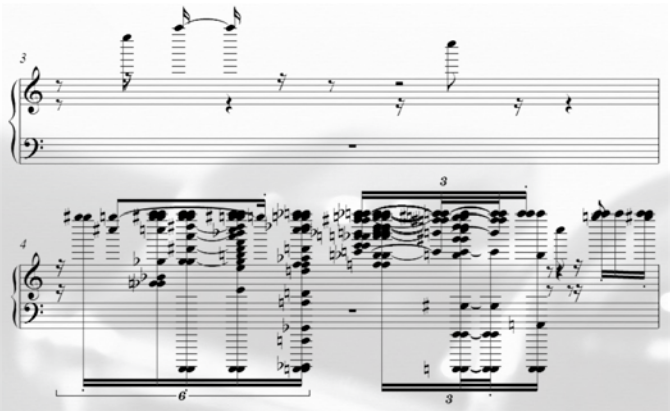


Abbildung 2: Der Aufruf einer Webseite ist sehr deutlich in der Audiolisierung zu erkennen.

Auch komplexere Situationen können durch Hören erkannt werden. Abbildung zwei zeigt den Ausschnitt einer Audiolisierung, bei deren Aufzeichnung eine Webseite geladen wurde. Der Aufruf der Webseite tritt deutlich aus dem ARP-Hintergrund hervor – das Verhältnis von hohen und tiefen Tönen sowie die Länge des gesamten Ereignisses sind charakteristisch. Dadurch ist es möglich, dass man das Laden einer Webseite nur durch Hören von anderen Aktivitäten im Netzwerk unterscheiden kann.

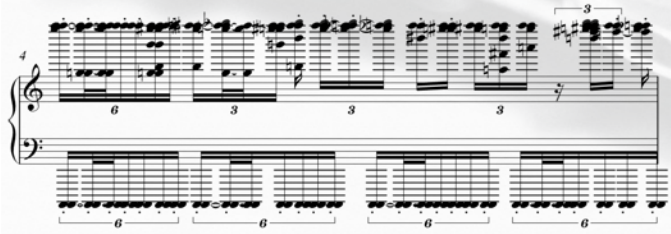


Abbildung 3: Ein längerer Download kann leicht durch das Hören der Audiolisierung erkannt werden.

In der dritten Abbildung sieht man die Audiolisierung eines Downloads einer größeren Datei. Dabei entstehen viele tiefe Töne, denn es werden große Pakete empfangen. Diese werden jeweils mit einem ACK quittiert, wodurch wiederum hohe Töne generiert werden. Die Audiolisierung des Downloads kann leicht von dem Aufruf einer Webseite unterschieden werden.

Die meisten Parameter einer Note werden in der aktuellen Version des Network-Audiolisers nicht genutzt. Dabei könnten verschiedene Informationen gleichzeitig repräsentiert werden. Es wäre zum Beispiel denkbar, verschiedene Instrumente für verschiedene Protokolle zu nutzen, die Tonlänge und Lautstärke zu variieren oder bestimmten IP-Adressen bestimmte Akkorde zuzuordnen und damit weitere Informationen über den Datenstrom zu kommunizieren. Die verwendeten Parameter der Note (Tonhöhe, Instrument, Länge, Lautstärke und Harmonie) könnten dabei gleichzeitig vom Hörer wahrgenommen werden.

Der Begriff Audiolisierung wurde schon 1990 von Blattner, Greenberg und Kamegai für die akustische Repräsentation von Datenmengen verwendet [1]. Der Ansatz, Informationen durch computergenerierte Klänge zu repräsentieren, scheint auf Bly [2] zurückzugehen, die 1982 zu dem Thema promovierte. Ihre Arbeit hat unter anderem ergeben, dass Testpersonen sechs Dimensionen einer Audiolisierung gleichzeitig wahrnehmen und den zugrundeliegenden Datensatz einschätzen können. Spätere Arbeiten im Bereich der Audiolisierung haben unter anderem vorgeschlagen, Computerprogramme beim Debuggen zu audiolisieren [3]. Dabei werden zum Beispiel Schleifendurchläufe durch ansteigende Töne dargestellt.

Neben der Nutzbarkeit der Audiolisierung zur Darstellung von Datenströmen sind die generierten Musikstücke aber auch interessant anzuhören.

Ein paar kurze Audiolisierungen von verschiedenen Access-Points im HPI finden sich auf meinem HPI-Webhome.

http://myhpi.de/~jakob.zwiener/NetworkAudioliser/

— *Jakob Zwiener*

[1] Blattner, M. M., Greenberg, R. M. & Kamegai, M. (1990). Listening to turbulence: An example of scientific audiolization. ACM SIGCHI'90 workshop on multimedia and multimodal interface design.

[2] Bly, S. (1981). Presenting Information in Sound. Proc. CHI'82 on Human Factors in Computer Systems, 371-375.

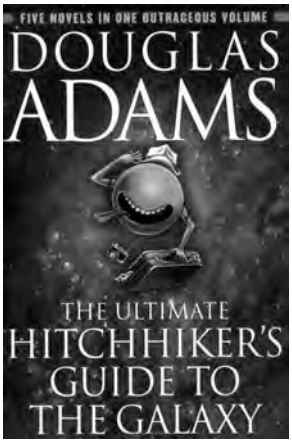
[3] Alty, J. L., Rigas, D. I., & Vickers, P. (1997). Using music as a communication medium. CHI Extended Abstracts, 30–31.

Das etwas andere Literaturverzeichnis

Vor wenigen Monaten beschwerte sich der Fantasie-Autor Neal Stephenson, dass der modernen Literatur die Utopien fehlen, um Wissenschaftler wie Ingenieure mit den nötigen Visionen auszustatten. Grund genug für das HPIMgzn sich zu fragen, was eigentlich die Bücher sind, die Visionen für die Informatik geliefert haben. Nachfolgend wollen wir euch kurz die Romane vorstellen, die unserer Meinung nach eine wichtige Rolle im kulturellen Selbstverständnis der Informatiker spielen, oder die Begriffe und Anschauungen der Informationstechnologie geprägt haben. Die Erkenntnis, dass dabei vornehmlich Dystopien zusammengekommen sind, überlassen wir dem geneigten Leser zur eigenständigen Interpretation.

Per Anhalter durch die Galaxis

Der universelle Reiseführer „Anhalter durch die Galaxis“ rät: „Ein Handtuch ist so ungefähr das Nützlichste, was der interstellare Anhalter besitzen kann. […] Der Strag (Nicht-Anhalter) denkt natürlich, dass ein Mann, der kreuz und quer durch die Galaxis trampt, ein hartes Leben führt, in die dreckigsten Winkel kommt, gegen schreckliche Übermächte kämpft, sich schließlich an sein Ziel durchschlägt und trotzdem noch weiß, wo sein Handtuch ist, eben ein Mann sein muss, auf den man sich verlassen kann.“ Die wichtigste Lehre der fünf Bände rund um den „Anhalter“ ist damit bereits wunderbar auf den Punkt gebracht. Eigentlich geht es aber um die verworrenen Reisen des Arthur Dent, dessen Planet Erde eines Morgens gerade dann zerstört wird, als er im Bademantel vor die Tür geht, um sein Haus vor dem Abriss zu bewahren. Ursprünglich von Douglas Adams als kleines Hörspiel der BBC veröffentlicht, entwickelt sich Arthurs Abenteuer zu einer Odyssee durch alle Ecken des Universums. Sein Kumpel Ford Prefect führt ihn in die Tradition der Weltraumanhalter ein und gemeinsam durchreisen sie die Galaxis. Dabei entdecken sie, warum die 42 keine Antwort, sondern vielmehr ein Problem darstellt, dass die Menschheit nur die drittschlauste Spezies auf der Erde ist und das man fliegen kann, wenn man es nur schafft beim Fallen einfach nicht aufzuprallen. Streckenweise werden sie immer wieder von Marvin dem paranoiden Roboter begleitet, der meistens alles weiß, den aber keiner gefragt hat. Wer die Kultur der Informatik verstehen möchte, kommt an diesem Buch nur schwer vorbei. Die Verfilmung von 2005 ist übrigens nur ein kleiner Ausschnitt. Sie umfasst lediglich einzelne Teile der ersten drei Bände und folgt einer etwas anderen Handlung. Die englische Ausgabe enthält zudem viele Sprachwitze, die in der deutschen Ausgabe fehlen. Ist aber genau darum auch schwerer zu lesen. So oder so: Keine Panik!



Titel
Per Anhalter durch die Galaxis

Autor
Douglas Adams

ISBN
978-3453209619 (dt)
978-0307291813 (en)

Alice im Wunderland

42 ist ein Zitat aus dem „Anhalter“, das weiß jedes belesene Kind. Dass Adams nicht der erste Phantastik-Autor war, der dieser Zahl eine besondere Bedeutung zuwies, wissen dagegen die Wenigsten. Charles Dodgson, besser bekannt unter seinem Pseudonym Lewis Caroll, machte schon im 19. Jh. in seinen Werken umfassend von dieser Zufallszahl Gebrauch. Unter anderem versucht in „Alice im Wunderland“ der Herzkönig Alice mithilfe der Regel 42 aus dem Gericht zu verbannen, und in der „Jagd nach dem Schnark“ besagt die Regel 42, dass niemand mit dem Steuermann zu reden habe. Auch der Fakt, dass der Anhalter in seiner Radiofassung, genauso wie Carolls Nonsensballade über den Schnark, in „Fits“ unterteilt wurde, sollte Grund genug sein, sich Adams Inspirationen einmal selber zu Gemüte zu führen.

Dass „Alice“ ursprünglich als Kindergeschichte geplant war, sollte niemanden davon abhalten es zu lesen, denn es hält auch heute noch locker mit den schlimmsten Drogenbüchern der Literaturszene mit. Im Verlauf der Geschichte folgt Alice einem paranoiden weißen Hasen mit goldener Taschenuhr in seinen Bau. Dort wechselt sie durch den Genuss diverser Substanzen mehrfach ihre Größe, ehe eine Shisha-rauchende Raupe ihr endlich einen Pilz verabreicht, der sie auf Normalgröße bringt. Im Anschluss besucht das junge Mädchen eine Teeparty, die nie endet, weil die Teilnehmer glauben, sie hätten bei einer Gesangseinlage die Zeit ermordet. Der Trip endet schließlich damit, dass Alice einen Schauprozess gegen den Herzbuben sprengt, der beschuldigt wird, der Herzkönigin die Törtchen geklaut zu haben. Das alles entstand übrigens als eine Improvisation auf einer Bootsfahrt, als Dodgson der echten Alice eine Geschichte erzählte, vermutlich um ihr ein Bild für seine Sammlung von Fotos junger Mädchen abzuluchsen. Die letzten Worte in diesem Abschnitt sollen übrigens ein Rat von Morpheus aus „The Matrix“ sein: „Follow the White Rabbit“.

1984

Gerne redet der gebildete Informatiker von Big Brother, wenn er in einem beliebigen Kontext Unterdrückung und Überwachungsstaat anprangern will. Diese Anspielung stammt aus George Orwells Roman „1984“, der kurz nach Ende des zweiten Weltkriegs entstand und eine dystopische Gesellschaft im Jahre 1984 beschreibt. In Orwells Großbritannien werden die gebildeten Schichten rund um die Uhr durch einen Fernseher-ähnlichen Teleschirm überwacht und mit Propaganda indoktriniert. Um subversive Gedankengänge zu verhindern, arbeiten die Machthabenden daran, die Sprache zu vereinfachen und stattdessen ein primitives, funktionales „Neusprech“ einzuführen, frei nach Wittgensteins These, dass man nur über Sachen denken kann, die sich auch in Worte fassen lassen. Solange dies nicht geschehen ist, werden „Gedankenverbrechen“ verfolgt, um Widerstand gegen die Partei undenkbar zu machen. Das Kernstück des Romans ist die These, dass es keine objektive Vergangenheit geben kann. „Wer die Macht über die Geschichte hat, hat auch Macht über Gegenwart und Zukunft“ lautet die Aussage des Buchs. Ein ganzes Ministerium („für Wahrheit“) ist damit beschäftigt alte Zeitungsauflagen und Do-

Titel
Alice im Wunderland

Autor
Lewis Carroll

ISBN
978-3458317425 (dt)

Titel
1984

Autor
George Orwell

ISBN
978-3453164215 (dt)

kumente neu zu drucken, sodass die Aussagen dort stets der Wahrheit entsprechen. Obwohl Jahrzehnte vor dem Entstehen des Internets geschrieben, beschreibt das Buch überspitzt Problematiken, die gerade heute Regierungen und Verfassungsgerichte in ganz Europa beschäftigen. Ganz nebenbei enthält es auch noch die schlimmste Folterszene, die ich jemals gelesen habe, sodass auch Freunde von härterer Literatur hier auf ihre Kosten kommen.

Schockwellenreiter

Wir schreiben das Jahr 1975. Computer sind so groß wie Schreibtische und die modernen Netzwerkprotokolle entwickelten sich gerade erst. In dieser Zeit schreibt John Brunner den zukunftsweisenden dystopischen Roman „Schockwellenreiter“. Alles dreht sich um die Flucht des Hackers Nick Haflinger Anfang des 21. Jahrhunderts. Er muss allerdings nicht nur physikalisch flüchten, sondern auch seine digitalen Spuren verwischen. Denn in Brunners Zukunft wird die Gesellschaft von einem riesigen Datennetz kontrolliert. Privatsphäre existiert nur für privilegierte Bürger. Das Leben aller anderen wird genau aufgezeichnet. Gleichzeitig hat ein gieriger Kapitalismus Kultur, Bildung und Gesellschaftsordnung zerstört. „Schockwellenreiter“ gilt als der erste Cyberpunk Roman, lange bevor das Genre Mitte der 80er Jahre populär wurde. Neben dem globalen Datennetz nahm Brunner auch bereits die Idee der „Würmer“, als Unterart von Computerviren, vorweg. Ein wichtiges Thema ist die Wahrnehmung von Menschen. Geben die Daten über einen Menschen ein besseres Bild über ihn ab als die Person selbst? Nick muss auf seiner Flucht immer wieder neue Persönlichkeiten im Netz erschaffen, die ihm als Deckung dienen. Nur die Tochter einer privilegierten Bürgerin sieht seine wahre Persönlichkeit. Und letztendlich sind Nicks Tarnungen nicht makellos und so bringt er den letzten Zufluchtsort gebildeter Menschen in Gefahr.

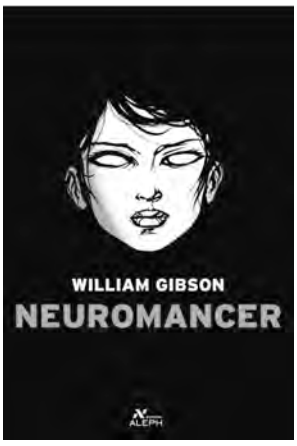
Neuromancer

1984, neun Jahre nach Brunners „Schockwellenreiter“ etablierte „Neuromancer“ das Genre des Cyberpunks endgültig. Wieder sind die Hauptcharaktere Hacker. Das globale Computernetz heißt hier zum ersten Mal „Cyberspace“. Damit prägte der Autor William Gibson den Begriff, der noch bis in die 90er als Synonym für das Internet verwendet wurde. Henry, die Hauptfigur, ist ein drogensüchtiger Programmierer. Seine Gehirn-Computer Schnittstelle wurde blockiert. Damit ist er aus dem Cyberspace verbannt. Ein zwielichter Agent verspricht ihm die Schnittstelle zu reaktivieren, wenn er im Gegenzug etwas für ihn erledige. Was Henry nicht weiß ist, dass im Hintergrund eine künstliche Intelligenz die Fäden in der Hand hält. Ihr Ziel ist es, sich mit ihrer zweiten Hälfte im Cyberspace zu vereinigen, obwohl Turing Schlösser sie davon abhalten. Henry und seine Mitstreiter sollen diese öffnen. Mächtige Kräfte in der realen Welt und im Cyberspace versuchen das allerdings zu verhindern. Dabei schrecken sie selbst davor nicht zurück, die Persönlichkeit toter Freunde des Teams im Cyberspace wieder auferstehen zu lassen. Denn der Cyberspace ist nicht mehr einfach nur eine virtuelle Welt. Mittlerweile haben sich in ihm bereits eigene Arten von Bewusstsein entwickelt.

Titel
Schockwellenreiter

Autor
John Brunner

ISBN
3453305841 (dt)



Titel
Neuromancer

Autor
William Gibson

ISBN
0441569595 (dt)

Schismatrix

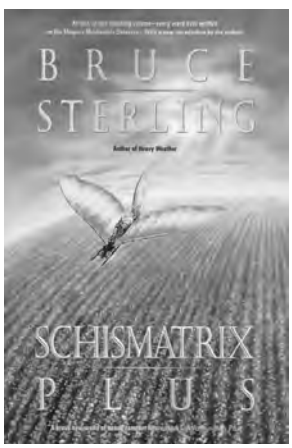
Welche Folgen kann die rasante Entwicklung von Technologie und Gentechnik auf eine menschliche Gesellschaft haben? In seinem Sci-Fi-Roman Schismatrix beschreibt Bruce Sterling über mehrere Hundert Jahre hinweg die Entwicklung der Menschheit zum „Transhumanismus“. Der Begriff Mensch scheint in dieser Zeit nur noch eine abstrakte Beschreibung für eine Ansammlung von Wesen zu sein, die sich lediglich in ihrer Abstammung ähneln. Mitten im Kampf der genetisch veränderten „Shaper“ gegen die kybernetisch aufgerüsteten „Mechanisten“ befindet sich der Diplomat Lindsay, ein Ausgestoßener, der durch geschickte Manipulation ganze Philosophien und Städte entstehen lässt und wieder zu Fall bringt.

Während die technologischen Neuerungen in dem Roman eher kurz kommen, beschreibt Sterling sehr ausgiebig und durchdacht, ganze Reihen von Weltanschauungen. Er schafft es erstaunlich gut, die charakterliche Entwicklung seiner Protagonisten über mehrere Jahrzehnte hin zu beschreiben. Ein Buch für alle, die auch gerne mal über die Folgen technologischen Fortschritts nachdenken.

Just For Fun

„Just For Fun“ zeigt, dass auch scheinbar revolutionäre Dinge, oft einfach aus einem persönlichen Bedürfnis eines einzelnen Menschen heraus entstehen. Auf leicht zu lesenden 250 Seiten berichtet Linus Torvald, der Erfinder von Linux, von seinem Lebensweg: Wie er mit Computern in Kontakt kam. Wie er seinen ersten Computer auf Kredit kaufte, als Einsiedler an dem neuen Betriebssystem arbeitete, wie die Community auf seine Arbeit reagierte und er schließlich sogar halb unfreiwillig eine Menge Geld dafür bekam. Auch persönliche Details lässt er nicht aus. So widmet er zum Beispiel ein ganzes Kapitel seinen Schwierigkeiten mit dem ersten Vortrag vor einem großen Publikum. In einem weiteren berichtet er, wie seine zukünftige Frau Torve ihn zu ihrem ersten Date per Mail einlud. Dank der technischen Details lernt man, mit etwas Hintergrundwissen aus BS I, auch einiges über den damaligen Konflikt zwischen Mikrokernel und monolithischen Systemen. Im Mittelpunkt steht hier ein abgedruckter Schlagabtausch via Email zwischen Andrew Tanenbaum, dem Verfechter von Mikrokernel-Systemen, und Linus Torvald, der seinen Entwurf in Linux verteidigt

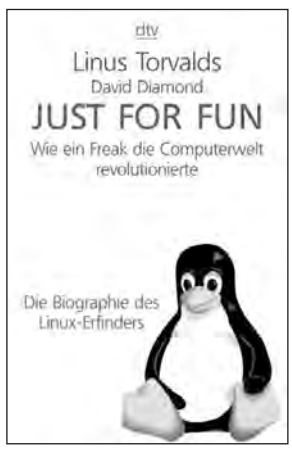
— Leonid Berov und Patrick Rein



Titel
Schismatrix

Autor
Bruce Sterling

ISBN
978-0441003709 (dt)



Titel
Just For Fun

Autor
Linus Torvald
David Diamond

ISBN
978-3423362993(dt)

Das gemeine Literaturverzeichnis

Im zweiten Teil unserer Abhandlung über Literatur nähern wir uns wieder dem Boden der Tatsachen und betrachten, welche Fachbücher die moderne Informatik am meisten geprägt haben. Natürlich ist es klar, dass jede Generation, vielleicht sogar jeder einzelne Informatiker, anderen Werken den Vorzug geben wird. Deswegen darf die nachfolgende Auflistung gerne als subjektive Auswahl der Autoren, und damit mit entsprechender Vorsicht, genossen werden. Nichtsdestotrotz hoffen wir eine möglichst repräsentative Auswahl getroffen zu haben, die Überblick gibt über Bücher zur Projektplanung, Geisteshaltung und zum Design von Software.

The Pragmatic Programmer

Wie wird man eigentlich ein „Entwickler"? Reicht es, drei Jahre studiert zu haben? Oder sollte man eher 50.000 Zeilen Code geschrieben haben?

Wie auch immer sie es geworden sind, Hunt und Thomas, die Autoren des „Pragmatic Programmers", sind Entwickler. Was diese Rolle für sie ausmacht, beschreiben sie in ihrem Buch.

Vieles davon kennt man aus Vorlesungen: Ruthless Testing, Refactoring und Ubiquitous Automation sind Kernthemen von Softwaretechnologie und –Architektur. Aber es geht den beiden Autoren auch um mehr! Welche Geisteshaltung prägt einen pragmatischen Entwickler? Wie sehen die Werkzeuge aus, die er nutzt? All das versuchen sie anschaulich, wenn auch zeitweise etwas länglich, zu vermitteln. Natürlich geht es auch ums Programmieren. Wozu nutzt man Exceptions, und wozu nicht? Wie schafft man es, komplexe Systeme zu synchronisieren ohne sich zu Tode zu „locken"? Wann fängt man mit dem Entwurf der Software an, und wann hört man damit am besten wieder auf?

Dieses Buch nutzt einem am meisten, wenn man schon ein oder zwei größere Projekte programmiert hat. Vieles, was darin beschrieben wird, wird man dann nachvollziehen können, einiges anzweifeln. Der eigentliche Wert ist aber der Gesamtkontext, in den diese Erfahrungen gestellt werden.

The Mythical Man Month

„Adding manpower to a late software project makes it later". Frederick Brooks beschrieb und erklärte 1975 diesen Effekt in seinem Buch „The Mythical Man Month". Seitdem ist dieser Satz auch als Brook's Law bekannt. Das Buch gehört zu den zeitlosen Standardwerken der Softwareentwicklung. Es beschreibt allgemeine Probleme der Entwicklung in großem Stil und deren Lösung - alles direkt aus der Praxis. Denn Brooks hatte vorher die Entwicklung von IBMs OS/360 geleitet, die sich mehrfach verzögerte. Unter anderem auch deshalb, weil er weitere Programmierer hinzuzog, als das Projekt bereits im Verzug lag. Die mussten sich erst einarbeiten und anschließend mit noch mehr Entwicklern absprechen. Der zugrundeliegende Trugschluss liegt, laut Brooks, in der Berechnung des Aufwandes. Wenn ein Programmierer zehn Monate braucht, um ein System zu bauen, dann braucht das System zehn Mann-Monate. Nimmt man also zehn Programmierer, müsste man nach einem Monat fertig sein, was selten funktioniert. Brooks Frustration über Softwareentwicklung gipfelte in dem Aufsatz „No Silver Bullet". Darin vertritt er die Meinung, dass es kein universelles endgültiges Wundermittel gibt und geben wird, um Softwareentwicklung termingerecht zu planen. „The Mythical Man Month" ist 1995 zudem in einer neuen Auflage erschienen, in der Brooks bewertet, was sich in den bis dahin vergangenen 20 Jahren verbessert hat. Aus seiner Sicht war der größte Schritt in dieser Zeit das Aufkommen von Büroanwendungen, insbesondere Tabellenanwendungen. Damit konnten normale Angestellte plötzlich Probleme lösen, die vorher von Programmierern gelöst werden mussten. Alles in allem sah er 1995 allerdings kaum Verbesserungen und zweifelte weiter an der Existenz einer „Silver Bullet".

How to run successful projects: The Silver Bullet

Frederick Brooks behauptet seit langer Zeit, es gäbe keine „Silver Bullet" mit der die Entwicklung von

Software endlich pünktlich und günstig gesehen könnte. Fergus O'Connell hält 2001 mit „How to run successful projects: The Silver Bullet" klar dagegen und behauptet: Es geht doch. Es braucht allerdings eine ganze Menge Disziplin. Das Buch beschreibt in einzelnen Schritten, wie man es schafft ein Softwareprojekt erfolgreich und pünktlich durchzuziehen. Dabei geht er immer von realen Fallbeispielen aus, entwickelt eine Regel und überträgt sie anschließend auf die Softwareentwicklung. Manche Dinge scheinen auf den ersten Blick trivial, wie zum Beispiel: „Stelle dir das Ziel vollständig vor". Anhand des Beispiels von Ernest Shackleton wird aber klar, dass dieser Schritt essentiell ist. Shackleton reiste bereits 1909 zum Südpol. Damit war er Amundsen um fast zwei Jahre voraus. Doch 97 Meilen vor dem Ziel bemerkten er und sein Team, dass sie vergessen hatten, den Rückweg mit einzurechnen und mussten umkehren. Ihre Vorstellung des Ziels war nicht klar genug gewesen, denn sie hatten nicht bedacht lebend zurückzukehren. Von zentraler Bedeutung im Buch ist die Tatsache, dass sich Funktionalität, Qualität und Aufwand immer gegenseitig beeinflussen. Will man mehr Funktionalität muss man entweder an Qualität sparen, oder aber mehr Aufwand betreiben. Das ist nicht zu ändern. Wie man trotzdem erfolgreiche Projekte durchführt, zeigt dann der Rest des Buches.

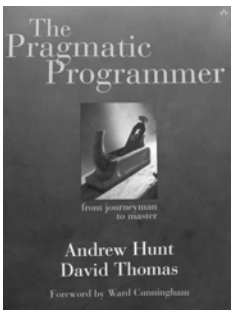
GoF Book

Entwurfsmuster. Elemente wiederverwendbarer objektorientierter Software, von Erich Gamma, Richard Helm, Ralph Johnson und John Vlissides ist ein ziemlich langer Name für ein Buch. Deswegen nennt man die Autoren die Gang of Four, und ihr Buch das GoF Buch. In der Softwarearchitektur Vorlesung werden die dort vorgestellten Muster ausführlich behandelt, das sollte aber kein Grund sein, das Buch nicht zu lesen.

Entwurfsmuster sind Lösungen für häufig auftretende Probleme, die nicht entwickelt, sondern entdeckt wurden. Entdeckt, weil unterschiedliche Programmierer an unterschiedlichen Projekten sie genutzt haben, ohne voneinander zu wissen, im Prinzip, weil sie sich geradezu anbieten, um dieses oder jenes Problem zu lösen.

Für alle 23 Muster erläutern die Autoren den Zweck, geben ein motivierendes Beispiel, diskutieren die Anwendberkeit sowie ihre Konsequenzen und veranschaulichen die Struktur des Musters anhand von Diagrammen und einer Beispielimplementierung. Alles davon lässt sich wohl nicht merken, und darum geht es eigentlich auch nicht. Was man anhand dieses Buches lernen kann, ist, wie man Probleme mit Hilfe von Objektorientierung elegant lösen kann. Man bekommt ein Gefühl für gute Architektur-Entscheidungen und kann sich die Kommunikation mit anderen Entwicklern vereinfachen, die auch mit den GoF-Mustern vertraut sind.

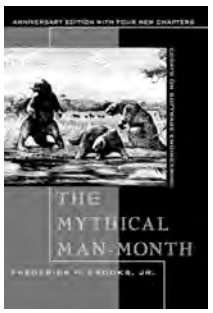
— *Leonid Berov und Patrick Rein*



Titel
The Pragmatic Programmer

Autor
A. Hunt, D. Thomas und W. Cunningham

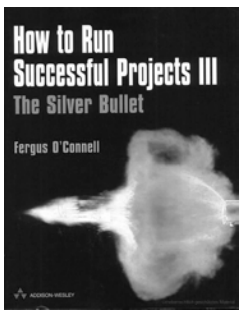
ISBN
978-0201616224 (en)



Titel
The Mythical Man-Month

Autor
Frederick P. Brooks

ISBN
978-0201835953 (en)



Titel
How to Run Successful Projects III: The Silver Bullet

Autor
Connell Fergus O

ISBN
978-0201748062 (en)

Titel
Design Patterns.

Autor
E. Gamma, R. Helm, R. Johnson, J. Vlissides

ISBN
978-0201633610 (en)

Ergonomie geistiger Arbeit

Geistige Arbeit stellt andere Ansprüche an den Menschen als körperliche Arbeit. Diese sollten Wissensarbeitern bewusst sein, damit sie frei von unnötigen Belastungen arbeiten können. Unter anderem beeinflussen drei Aspekte unsere geistige Leistungsfähigkeit: Motivation, Techniken und Arbeitsbedingungen.

Geistige Arbeit wird in der Psychologie grob über drei Schritte definiert: Man nimmt Informationen auf, dann verarbeitet man sie und gibt sie anschließend wieder. In der Verarbeitungsphase entsteht das tatsächliche Inkrement. Im Inneren unseres Kopfes beteiligt sich unser Verstand daran wiederum mit drei Tätigkeiten: Zum einen nimmt er Informationen über die Sinne wahr. Zum anderen kann er aber auch Informationen oder Eindrücke durch Phantasie entstehen lassen. Die tatsächliche Arbeit findet dann beim Denken statt. Dabei werden aufgenommene Informationen kombiniert und Schlussfolgerungen gezogen.

„Das einzig Weise im Leben ist die Konzentration, das einzige Übel die Zerstreutheit."

Prof. Alexis Carrel, Nobelpreisträger

Die grobe Theorie ist also recht übersichtlich. Die Praxis sieht oft anders aus, denn damit der Prozess des Denkens problemlos ablaufen kann, müssen einige Rahmenbedingungen erfüllt sein. Bildlich kann man sich geistige Arbeit wie Bogenschießen vorstellen. Das Problem ist die Zielscheibe, unser Verstand ist der Bogen. Wir versuchen mit unseren Gedanken genau die Lösung des Problems zu treffen.

Die Motivation ist in diesem Bild die Hand an der Sehne. Sie gibt den Gedanken Kraft und hält sie grundlegend in Richtung des Ziels. Wenn man von einer Sache sehr angetan ist, dann fällt einem die Arbeit daran sehr leicht. Das liegt daran, dass die Sache im Idealfall an sich motiviert. Allerdings können Menschen auch noch auf andere Weisen motiviert werden.

Arbeitstechniken sind wie die Wahl des richtigen Bogens für ein konkretes Ziel, also das Problem. Wenn man über einen konkreten Sachverhalt nachdenkt, ist Denken komplexer als reine Kombination und Schlussfolgerung. Zum einen braucht man Wissen, um über etwas nachdenken zu können. Zum anderen braucht man aber auch Denkstrategien. Viele solcher Lösungswege haben wir implizit gelernt. Für manche Probleme braucht es aber auch spezielle explizite Strategien, beispielsweise Techniken zum kreativen Denken. Die d-school versucht zum Beispiel, solche Techniken zu erfassen und zu bewerten.

Zu guter Letzt braucht man Ruhe, um sich konzentrieren zu können, und wenig Wind, damit der Pfeil ungestört fliegen kann. Denn selbst, wenn man motiviert sein sollte und die richtigen geistigen Werkzeuge an der Hand hat, wird man nicht richtig vorwärts kommen, wenn die Arbeitsumgebung nicht stimmt. Zu den Arbeitsbedingungen gehören zum einen direkte Einflüsse aus der Umgebung, wie zum Beispiel ein aufgeräumter Schreibtisch oder Stille. Weiter zählt dazu ein passender Lebensstil, mit guter Ernährung und ausreichend Schlaf.

Sind Rahmenbedingungen nicht erfüllt, so wird die Arbeit belastender. Die Beanspruchung entspringt vielen verschiedenen Einflüssen, was sie

schwer zu fassen macht. Beispielsweise strengt es an, wenn man regelmäßig unterbrochen wird und sich neu konzentrieren muss. Es ist aber ebenso anstrengend, wenn man eine Aufgabe erledigen soll, die zu schwer oder auch zu leicht ist. Allgemein entsteht die Belastung immer aus einer individuellen Auseinandersetzung mit einer Aufgabe. Neben externen Einflüssen sind dabei die Hauptquellen für Belastung über- und unterfordernde Aufgaben. All diese Belastungen sollte man vermeiden und darum möglichst optimale Rahmenbedingungen schaffen.

Möchte man mit einem Bogen ein Ziel treffen, so muss man den Pfeil ruhig halten und die Sehne spannen. Die Motivation hat diese Wirkung auf unsere Gedanken. Wenn man motiviert ist, ist es kein Problem, seine Gedanken anzustrengen und auf das Ziel auszurichten. Ist man nicht motiviert, schweift man gerne ab und verliert sich in Kleinigkeiten.

Für die Wissensarbeit ist es wichtig, dass man sich vor allem der tatsächlich zu Grunde liegenden Motivation bewusst ist. Warum gibt ein Student seine Übungen pünktlich ab? Zum einen ist er direkt dadurch motiviert, dass er keine Anerkennung bekommt, wenn er sie zu spät abgibt. Zum anderen ist er indirekt motiviert, da er mit den Übungspunkten zur Klausur zugelassen wird, die ihm irgendwann einen Abschluss ermöglicht, mit dem er seine Lebensziele erreichen kann. Der Pädagoge Prof. Ernst Ott unterscheidet zwischen zwei grundlegenden Arten der Motivation: positive und negative. Menschen, die positiv motiviert sind, erledigen etwas, um einen Erfolg zu erringen. Sie stellen sich etwas Positives vor, das sie erlangen, wenn sie ihr gestecktes Ziel erreichen. Zum Beispiel übt der Student für Klausuren, um gute Noten zu erzielen, die ihm ermöglichen, seinen Traumberuf zu erreichen. Negativ motivierte Menschen hingegen versuchen durch ihre Tätigkeiten, Misserfolge zu vermeiden. Der Student versucht vielleicht mit der pünktlich abgegebenen Übung, eine Blamage vor dem Übungsleiter zu vermeiden.

Beide Motivationsarten sind extern, da sie auf dem Ergebnis der Arbeit basieren. Die beste Motivation ist aber die Primärmotivation. Menschen, die für eine Aufgabe primärmotiviert sind, tun etwas, ohne einen positiven Effekt zu erwarten oder einen negativen vermeiden zu wollen. Die konkrete Aufgabe motiviert sie von selbst. In dem Fall ist man automatisch voll konzentriert und die Arbeit geht wie von selbst. Um eine gute Grundlage für die Arbeit zu haben, sollte man zu einer Aufgabe positiv motiviert sein. Dazu muss man sich erst einmal bewusst werden, was für einen selbst als Motivation dient. Dann kann man Stück für Stück den Schritt von der negativen Motivation zur positiven machen. Bewusst den Schritt zur primären Motivation zu machen ist schwer. Oft erlangt man sie erst durch regelmäßiges Tun.

Auch der kräftigste Arm hilft nichts, wenn man den falschen Bogen gewählt hat. Ein weit entferntes still stehendes Ziel braucht einen anderen Bogen als ein bewegtes Ziel. So hilft einem auch die beste Motivation nichts ohne passende Werkzeuge.

Der Werkzeugkasten jedes Wissensarbeiters sollte zwei Teile haben: Auf der einen Seite liegen fachliches Wissen und Lösungsstrategien, wie zum Beispiel ein mathematischer Beweis oder ein juristisches Gutachten. Auf der anderen Seite liegen allgemeine logische und kreative Denkstrategien und Wege, die eigene Arbeit zu organisieren.

Logische Strategien werden kaum mehr explizit formuliert. Dabei nutzt man sie in der geistigen Arbeit regelmäßig, wie zum Beispiel das Abgrenzen eines Themas oder die Bestimmung der Prämissen einer Aussage. Das freie Buch „Thinking like a Knowledge Worker“ formuliert zum Beispiel solche allgemeinen Strategien, von denen alle Zweige der Wissensarbeit profitieren können. Kein Zweig kommt zum Beispiel an Argumentationen vorbei. Dann kann es helfen, ein Verfahren zu haben, um Grundannahmen der eigenen Argumente zu finden.

Beim Lösen von vagen Problemen helfen kreative Techniken. Diese reichen vom einfachen Skizzieren über Brainstorming bis zum projektübergreifenden Ansatz des Design-Thinking. Kreative Techniken findet man in vielen Quellen. Die Kunst ist es vielmehr zu wissen, wann man eine Technik anzuwenden hat. Man muss sich klar werden, dass man gerade an einem vagen Problem sitzt, das sich nicht direkt mit einer fachlichen Strategie lösen lässt.

Beide Arten von allgemeinen Techniken können einem die Arbeit erleichtern. Da man sie in vielen Situationen anwenden kann, kann man vormals un-

klare und damit vielleicht auch überfordernde Aufgaben systematisch angehen und lösen.

Organisationstechniken helfen, da Wissensarbeit oft mit einem hohen Maß an Selbstverantwortung für die eigene Arbeit verbunden ist. Je nach Beschäftigung muss man seine Aufgaben mehr oder weniger selbst verwalten. Verliert man den Überblick, so entsteht wieder Überforderung, da man gar nicht weiß, wie viel Arbeit eigentlich noch zu tun ist. Das eigene System zur Organisation ist dabei nebensächlich, solange man sich dabei wohl fühlt. Man kann versuchen, ein reglementiertes System wie „Getting Things Done“ umzusetzen oder beispielsweise einfach Post-Its an einer Wand verwenden. Wichtig ist, dass man dem System zutraut, dass es alle Aufgaben erfasst. Ansonsten denkt man während einer Aufgabe wieder an alle noch zu erledigen Dinge, die eigentlich durch das System erfasst werden sollten.

Lernen ist eine der grundlegenden Techniken – genauer gesagt das Lernen tatsächlicher Zusammenhänge. Oftmals lernt man nur auswendig. Selbst wenn man lernt wie etwas funktioniert, kann das in Wirklichkeit reines Auswendiglernen sein, wenn man das dahinter stehende Wirkprinzip nicht begreift. Bewusstes Lernen ist schwerer, allerdings kann man dann aus allen Situationen lernen. Abwechslung kann helfen, die Umgebung bewusster wahrzunehmen. Sich aus seinem Komfortbereich zu bewegen ist immer lehrreich.

Der Bogen ist perfekt gewählt, die Hand hält Bogen und Sehne ruhig. Das Ziel ist anvisiert und der Winkel passt perfekt zur Entfernung. Doch der Wind macht einen sauberen Schuss unmöglich. Die Bedingungen müssen stimmen, selbst wenn man motiviert ist und sich der Aufgabe gewachsen fühlt. Das gilt sowohl für die Umwelt, in der man arbeitet, als auch den eigenen Zustand.

In einem Merkblatt von 1932 ist bereits der ideale Arbeitsplatz beschrieben. Der Arbeitsplatz sollte genug Platz bieten, aufgeräumt und gut gelüftet sein. Die Temperatur sollte angenehm und auf keinen Fall zu warm sein. Stuhl und Tisch sollten ergonomisch sein und eine aufrechte Haltung unterstützen. Außerdem sollte man sich für dezidierte Arbeiten in Stille zurück ziehen können. Ob Stille nun Musik einschließt oder nicht, darüber kann man sich streiten (s. Umfrage Seite 25). Auf jeden Fall sollte man nicht durch beiläufige Gespräche abgelenkt werden. Denn jede dieser Ablenkungen stört unsere Konzentration, die wir anschließend wieder komplett neu aufbauen müssen. Das kostet viel Energie und verlangsamt die Arbeit, was zusätzlich anstrengt. Bei agilen Softwareprojekten müssen die Entwickler hier zum Beispiel die Balance finden zwischen reger Kommunikation und individueller Stillarbeit.

Neben den externen Bedingungen sollte man auf sich selbst achten, sowohl während der Arbeit, als auch im allgemeinen Lebensstil. Auf lange Sicht lohnt es sich nicht, sich andauernd zu überlasten. Am besten sind zwei bis drei Stunden konzentrierte Arbeit und anschließend eine halbe Stunde Pause. Während der Pause sollte man Abstand von der Arbeit halten, sich bewegen und sich geistig zerstreuen. Wann man arbeitet, hängt von den eigenen Vorlieben ab. Ob früh am Morgen oder spät abends ist egal, solange man wach und aufmerksam ist. Wenn man doch mal müde sein sollte, ist Koffein nur für den Notfall. Am besten hilft immer noch Schlaf, frische Luft und Sport.

Jeder dieser drei Aspekte – Motivation, Techniken und Arbeitsbedingungen – kann einem die Arbeit erleichtern. Bei allen drei Punkten sollte man sich immer zuerst bewusst werden, wo man etwas verbessern kann. Dann sollte man nach entsprechenden Verbesserungen suchen und sie so umsetzen, dass sie einem in Fleisch und Blut übergehen. Von selbst wird sich die Arbeit zwar nie erledigen, aber vielleicht so leicht von der Hand gehen, dass sie schlussendlich Spaß macht.

— *Patrick Rein*

Literatur

Thinking like a Knowledge Woker
von William P. Sheridan
– http://unpan1.un.org/intradoc/groups/public/documents/unpan/unpan031277.pdf

Getting Things Done von David Allen
Das Konzentrationsprogramm von Ernst Ott
– ISBN: 978-3499170997

Merkblatt von 1932 zum Thema geistige Arbeit
– http://library.fes.de/pdf-files/arbeitersport/a80-10456.pdf

Geistige Arbeit braucht Form

Jährlich nimmt der Anteil der Menschen zu, die mit der Aufnahme, Verarbeitung und Weitergabe von Informationen ihren Lebensunterhalt verdienen. Dazu zählen auch Wissenschaftler und natürlich Ingenieure.
Gleichzeitig mehren sich die Berichte über Stress im Beruf. Neben steigendem Druck durch den Arbeitsmarkt könnte die Haltung von Unternehmen und Angestellten gegenüber Wissensarbeit ein Problem sein, denn geistige Arbeit funktioniert anders als körperliche. Das Wissen um dieses Wesen der Wissensarbeit und die Methodik scheint verloren zu gehen. Das bildet aber das Fundament und die Werkzeuge jedes geistigen Arbeiters und darum wird es Zeit, dieses Wissen wieder zu beleben und zu erneuern.

Mit den richtigen geistigen Werkzeugen gehen viele Arbeiten leichter von der Hand.

Bei körperlicher Arbeit ist klar, wie man effektiv und nachhaltig arbeitet. Übertrieben vereinfacht arbeitet man möglichst regelmäßig, dafür aber kürzer. Man macht ausreichend Pausen und wechselt gelegentlich die Tätigkeit, damit keine Monotonie entsteht. Für zahlreiche Berufsgruppen sind die häufigsten Berufskrankheiten bekannt, sodass man diesen zu einem großen Teil vorbeugen kann.

Auch auf dem Gebiet der Wissensarbeit ist viel über „richtiges“ Arbeiten bekannt. Warum arbeiten wir dann aber dennoch bis spät in die Nacht? Warum packen moderne Software-Firmen, wie z. B. AppNexus oder Klout, weiterhin ganze Säle mit Programmierern voll? Warum halten wir den Rindsbraten mit Soße für ein gutes Mittagessen kurz vor dem Neuentwurf von Kernkomponenten? Die Gründe für die einzelnen Schritte sind vermutlich individuell verschieden. Aber alle diese Phänomene haben ein Problem, dass das alte Wort Arbeitshygiene beschreibt. Alle führen zu schlechten Arbeitsbedingungen, durch die unsere Aufmerksamkeit abnimmt und wir damit geistig weniger leisten können.

Neben diesen Arbeitsbedingungen haben Arbeitstechniken einen wichtigen Einfluss auf die tatsächliche Belastung. Eine wichtige Technik ist zum Beispiel die Art, wie wir unsere Aufgaben organisieren. Bei physikalischer Arbeit sind die anstehenden Aufgaben oft sichtbar, wie zum Beispiel ein ungestrichener Zaun. Geistige Arbeiten hingegen sind nur selten greifbar – sie haben keine Grenzen und sind damit vorerst sehr abstrakt. Es ist schwer auszumachen, was wirklich getan werden muss. Die Art, die Aufgaben zu notieren und zu organisieren, kann hier den Unterschied machen; zwischen dem ruhigen Gefühl, alles unter Kontrolle zu haben und dem Gefühl, von all der Arbeit überwältigt zu werden. Dabei ist das tatsächliche System nebensächlich, solange man selbst das Gefühl hat, den Überblick zu bewahren. Solche Techniken sollte man bereits bewusst in der Ausbildungszeit lernen oder besser noch vorgestellt und gelehrt bekommen. Fest steht, dass sie zum Werkzeugkasten eines bewussten Wissensarbeiters gehören müssen.

Stimmen die Arbeitsbedingungen nicht oder fehlen einem grundlegende Arbeitstechniken, so ist die Arbeit belastender, als sie sein müsste. Gerade wenn die Arbeit einen sonst ausfüllt, kann das einem das Leben unnötig schwer machen. Darum sollten wir uns unnötiger Belastungen bei geistiger Arbeit bewusst werden und ihnen passende Techniken und gute Arbeitsbedingungen entgegensetzen.

— *Patrick Rein*

Open Source am HPI

Teil II: Eigeninitiative und Fachschaftsrat

In loser Folge stellen wir an dieser Stelle Open-Source-Aktivitäten am HPI vor. Dabei möchten wir möglichst viele verschiedene Facetten dieser Arbeit darstellen. So auch diesmal: Da ist auf der einen Seite das Engagement eines Studenten, der eine Lücke für sich entdeckt und daraufhin eine passende Bibliothek veröffentlich. Auf der anderen Seite soll auch die Arbeit unseres Fachschaftsrates nicht unerwähnt bleiben, der sich von der Veröffentlichung seines Codes wertvolle Hilfe erhofft.

Philipp Giese

https://github.com/pxlplnt

Bei seinem Praktikum bei SAP in Südafrika stellte Philipp Giese fest, dass Entwickler, „sobald sie mit Datenbanken in Berührung kommen, gleich wieder versuchen, doch nicht damit in Berührung zu kommen."

In diesem Zusammenhang war der Umgang mit dem Python-Framework Django für ihn ein echter Genuss. Dessen Datenbank-Tools vereinfachen die Arbeit mit Datenbank-Abfragen enorm. Bei weiteren Projekten im mobilen Bereich stellte er dann jedoch ernüchtert fest, dass ähnliche Werkzeuge bei der Java-Entwicklung für die Android-Plattform fehlen.

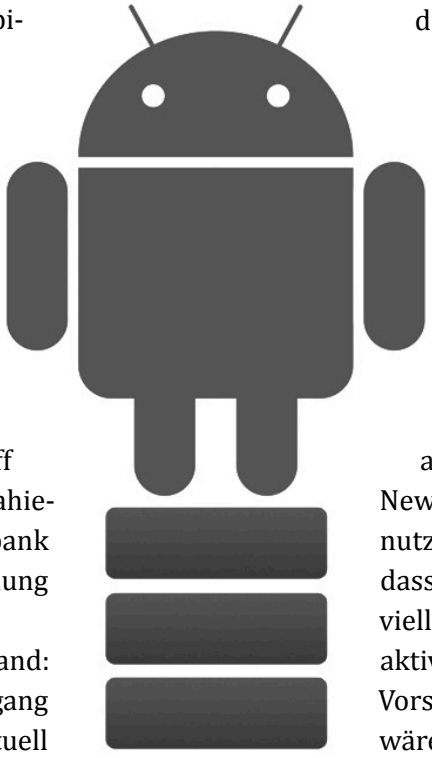
Und so entwickelte Philipp kurzentschlossen androrm, ein – der Name ist Programm – ORM (Object-Relational-Mapper) für die Entwicklung für Android. ORMs versuchen, den Zugriff auf die Datenbank so weit zu abstrahieren, dass einer Entität in der Datenbank immer ein Objekt in der Anwendung entspricht.

Die Vorteile liegen auf der Hand: man erspart sich den lästigen Umgang mit SQL-Abfragen sowie eventuell nötige Escape-Mechanismen für Daten vom Benutzer und typische Workflows wie „Auslesen, verändern, speichern" sind mit wenigen Zeilen Code getan. Natürlich ist die Abstraktion nicht immer komplett und so wird auch normales SQL von ORMs unterstützt. Für die üblichen Anwendungsfälle benötigt man als Entwickler jedoch, genau genommen, wenig bis keine Ahnung von Datenbanken, so stark ist das Lesen und das Schreiben der Daten gekapselt. Alles objektorientiert – „der Entwickler braucht nie heraus aus seiner Java-Welt", ergänzt Philipp.

Motivation, die eigene Arbeit als Open-Source anzubieten, hat Philipp genug: Zum einen hat er selbst schon viel offene Software benutzt, zum anderen erhofft er sich auch, von der Community zu profitieren. So erwartet er neue Ideen und auch „mehr Qualität, da ja mehr Leute draufgucken".

Überrascht hat ihn die Intensität des Supports über die verschiedenen Kanäle: Er benutzt die Ticket-Funktion auf GitHub, bis vor Kurzem noch ein eigenes Ticket-System („Fabricator" von Facebook) und auch noch die klassische Variante über Email. Interessanterweise hatte er vergessen, die entsprechende Email-Adresse auf der Projekt-Webseite anzugeben. Nicht nur in solchen Fällen ist es natürlich wichtig, Vorschläge der Community ernst zu nehmen. Auch schnelles Antworten „auf allen Kanälen ist wichtig", erzählt er.

Konkurrenz ist Philipp nicht bekannt. Stattdessen wächst die Community bereits langsam aber stetig – selbst eine Firma in New York hat Pläne, die Library zu benutzen. Für die Zukunft wünscht er sich, dass diese Entwicklung anhält und sich vielleicht noch beschleunigt. Besonders aktive Entwickler, die ihre Ideen und Vorschläge in das Projekt einbringen, wären da natürlich eine große Hilfe.



androrm bringt Android auf Datenbanken

Fachschaftsrat ITSE

https://github.com/fsr-itse

Dass auch Organisationen von Open-Source-Projekten profitieren können, beweist seit Neuestem der Fachschaftsrat des Studiengangs IT-Systems Engineering.

Für die Verwaltung der am Ende jedes Semester stattfindenden Veranstaltungs-Evaluierungen wurde seit Jahren die eigens für diesen Zweck entwickelte Software evaJ benutzt. Diese erwies sich mitunter als zu kompliziert und die Wartung wurde nach und nach immer schwieriger.

So reifte der Entschluss, evaJ komplett zu erneuern. Herausgekommen ist dabei evaP, nach Auftrag des Fachschaftsrates entwickelt von Michael Grünwald und Stefan Richter. Die neue Software präsentiert sich nicht nur optisch in neuem Gewand, auch hinter den Kulissen hat sich einiges getan. Geschrieben in Python, ist nun auch der Code wieder wartbar und aufgeräumter als zuvor. Denn trotz des einfach scheinenden Prozesses – muss man nicht bloß Stimmen einsammeln und die Ergebnisse aggregieren? – steckt hinter den Kulissen der Evaluierungsplattform so einiges an Logik.

Jetzt wird evaP eröffentlicht. Und was bietet sich dafür besser an als GitHub; eine Plattform, die es wie wohl keine andere schafft, Open-Source-Prozesse zu vereinfachen und beschleunigen, ja Beteiligung überhaupt erst anzuregen. Schon bei der alten Software war nie wirklich klar, wer daran weiterarbeiten konnte. Zwar zeichnete der Fachschaftsrat verantwortlich für die zweimal im Jahr stattfindenden Evaluierungen – zum Programmieren verpflichten konnte man ihn deshalb noch lange nicht.

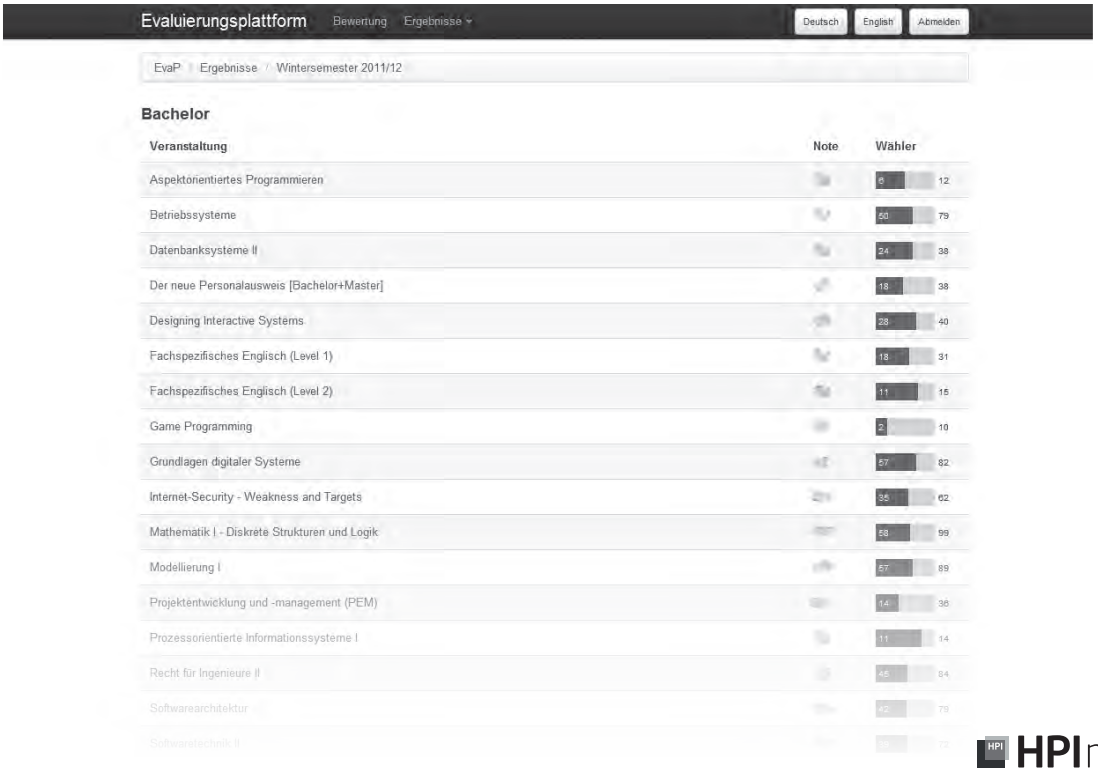
Das Veröffentlichen der Software auf GitHub ist somit ein Versuch, die Studenten besser in die Arbeit des Fachschaftsrates einzubinden.

„Wenn alle sich den Code angucken können“, so Matthias Jacob für den Fachschaftsrat, kann auch jeder Probleme beheben. Zwar ist die Software derzeit sehr auf die Anforderungen am HPI zugeschnitten. Es bestehe jedoch auch die Hoffnung, dass das Tool sich als hilfreich für andere Einrichtungen mit ähnlichen Prozessen erweisen kann, die dann ihrerseits wiederum neue Ideen und Bugfixes einbringen können – eine klassische Win-win-Situation.

Und nicht zuletzt ist es vielleicht gar nicht so schlecht, den Quellcode eines Programms, das Teil der studentischen Mitgestaltung ist, frei verfügbar zu machen. So kann jeder sich selbst ein Bild davon machen, wie die Ergebnisse gerecht und unbeeinflusst ermittelt werden.

— *Franz Liedke*

Dein Name fehlt hier? Du bist doch nicht so berühmt, wie du es verdient hast? Wir möchten gern auch dein Projekt an dieser Stelle vorstellen. Melde dich bei franz.liedke@student.hpi.uni-potsdam.de.



Musik macht müde

Warum hörst du Musik beim Programmieren?

„Weil Programmieren sich dann mehr wie Freizeit und weniger wie Arbeit anfühlt.“

„Geht nicht ohne!“

„Guter Metal bläst eben alles Unwichtige aus dem Hirn.“

„Weil der Kopfhörer eine Kommunikationsbarriere ist. Man spricht seltener andere an und wird seltener angesprochen.“

„Um in einen gleichmäßigen ›Flow‹ zu kommen. Es geht vielmehr um Gleichartigkeit als um Konzentration.“

„Weil die Musik mich daran erinnert, dass es trotz den endlosen, dunklen Tiefen des Strudels der Informatik auch noch schöne Dinge auf der Welt gibt.“

„Weil ich gerade sowieso nicht ordentlich arbeite und Lust auf bestimmte Musik habe. Wenn der ›Flow‹ dann kommt, dann vergesse ich die Musik sowieso ganz schnell.“

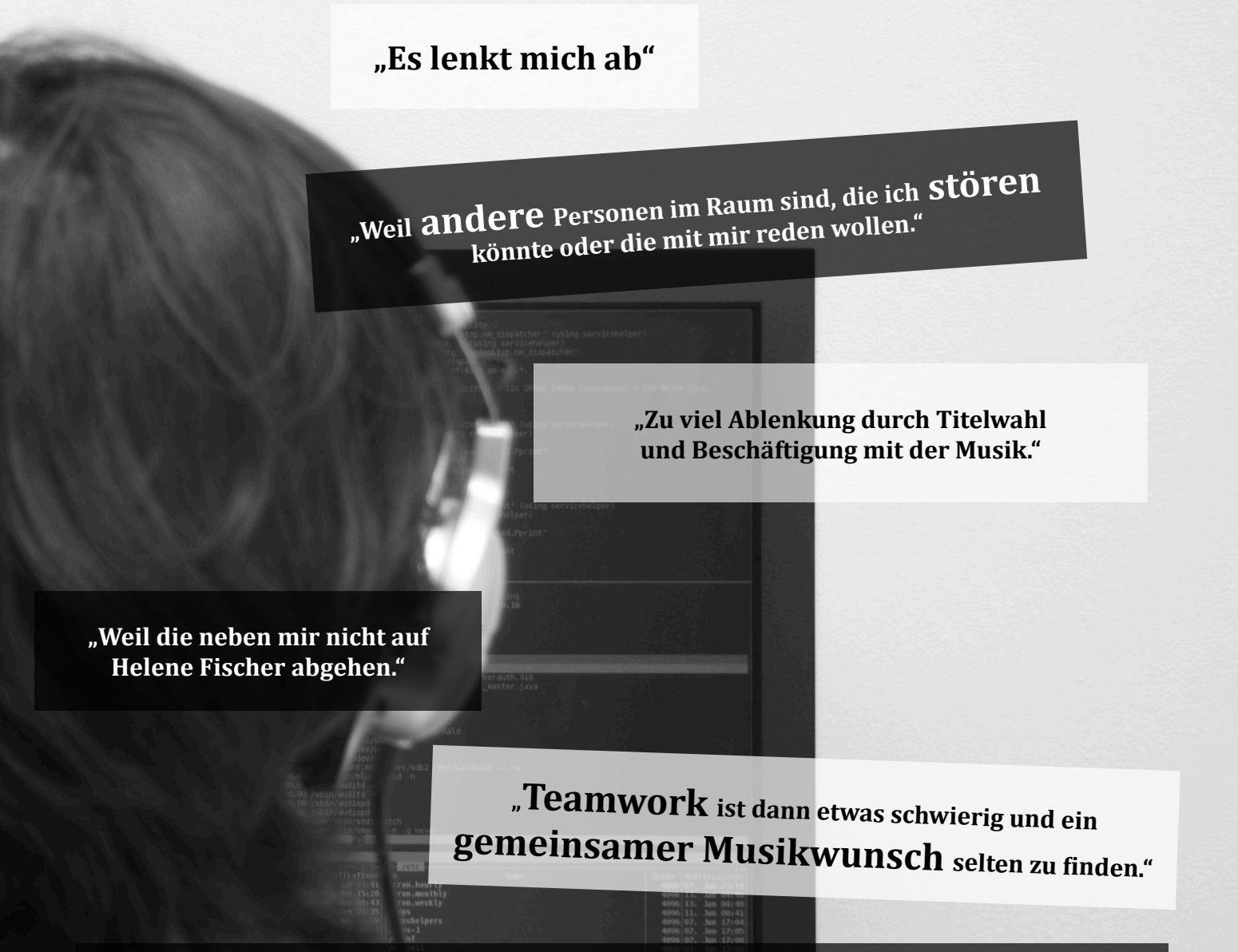
Musik lockert viele Arbeiten auf. Küchenarbeit geht leichter von der Hand mit Musik, das Fahrrad ist schneller wieder zusammengebaut und auch die Vorlesungsunterlagen des letzten Jahres sind schneller sortiert. So scheint es auf jeden Fall, denn viele Menschen hören während solcher Arbeiten Musik. Auch beim Programmieren hören viele Musik. Allerdings stellt sich hier die Frage, ob sie das bewusst zur Leistungssteigerung einsetzen oder einfach nur gerne Musik hören. Einer Studie zufolge geben Kühe zum Beispiel mehr Milch, wenn man sie mit der richtigen Musik beschallt. Liegt es da so fern anzunehmen, dass auch Programmierer munter werden und mehr und hoffentlich besseren Code schreiben, wenn sie Musik hören?

Die Umfrage zeigt, dass viele Musik vor allem als störend empfinden, wenn sie sich konzentrieren müssen. Unter den Gründen für Musik beim Programmieren hat es „Bessere Konzentration“ nur auf Platz vier geschafft. Ein paar meinen sogar, dass das Hören von Musik die Teamarbeit behindere, selbst wenn man Kopfhörer nutzt. Zumindest subjektiv

Die betrachteten Musikstile sind weder vollständig noch differenziert. Sie basieren auf einer groben Bestimmung durch Google und Wikipedia.

Programmierer munter

Warum hörst du beim Programmieren keine Musik?



„Es lenkt mich ab“

„Weil andere Personen im Raum sind, die ich stören könnte oder die mit mir reden wollen.“

„Zu viel Ablenkung durch Titelwahl und Beschäftigung mit der Musik.“

„Weil die neben mir nicht auf Helene Fischer abgehen.“

„Teamwork ist dann etwas schwierig und ein gemeinsamer Musikwunsch selten zu finden.“

scheinen wir beim Programmieren also nicht das Gefühl zu haben, dass Musik unsere Produktivität verbessert.

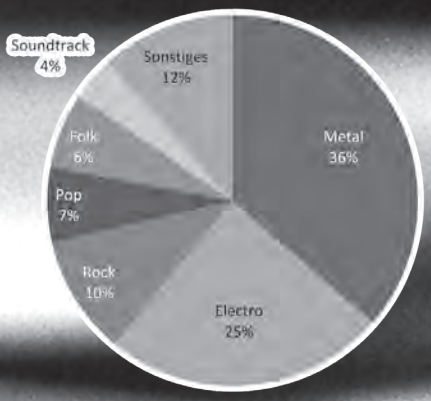
Warum aber hören dann so viele wenigstens gelegentlich Musik beim Programmieren? Die Erklärung ist beinahe banal: Viele hören einfach gerne gute Musik. Für manche gehen die Gründe noch weiter. Sie brauchen akustische Untermalung, um die Stille während der Arbeit zu übertönen. So oder so ist Musik mehr Unterhaltung als akustisches Doping für den Verstand.

Auf einem anderen Weg macht uns Musik aber vielleicht doch produktiver. Denn Programmieren bereitet mit Musik mehr Vergnügen. Empfindet man eine Tätigkeit als Arbeit, so muss man sich überwinden. Was sich allerdings wie Freizeit anfühlt und uns Spaß macht, geht auch leichter von der Hand und laugt uns weniger aus. Ein Umfrageteilnehmer bringt es auf den Punkt: „Programmieren [fühlt] sich dann mehr wie Freizeit und weniger wie Arbeit [an].“ Also merket liebe Programmierer: Man macht sich munter mit Musik!

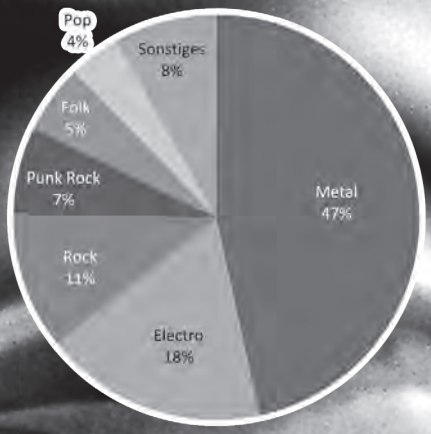
– Patrick Rein

Warum hörst du Musik beim Programmieren?

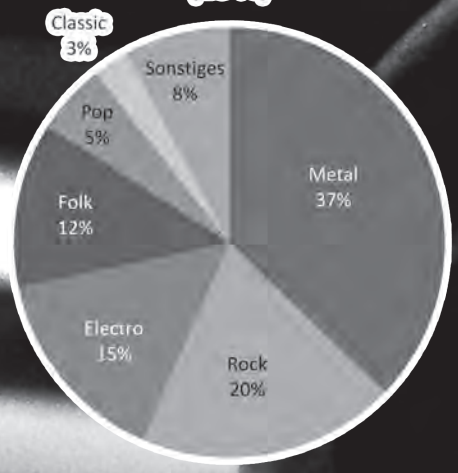
... um weniger von der Umgebung mitzubekommen (24 %)



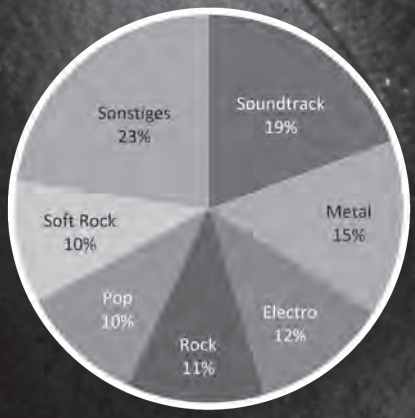
... um mich aufzurütteln (7 %)



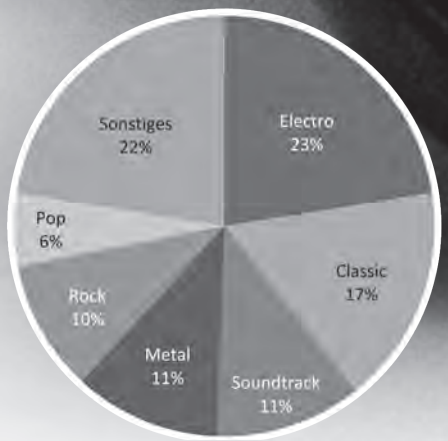
... einfach weil ich gerne etwas höre (28 %)



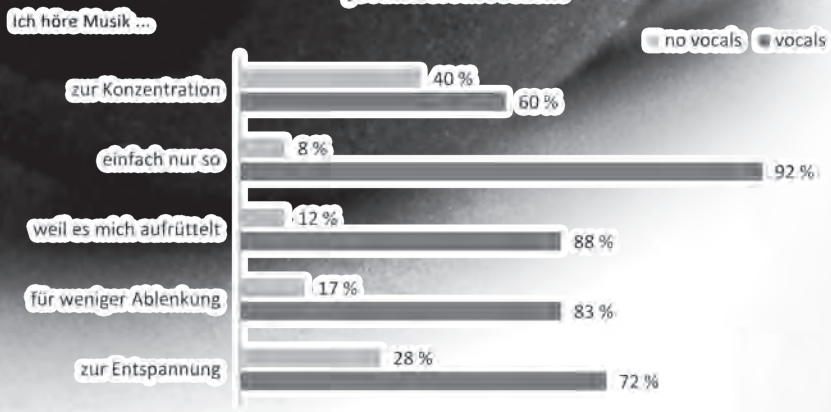
... um eine entspannte Atmosphäre zu erzeugen (26 %)



... um mich besser konzentrieren zu können (16 %)



Anteil der Musik mit Gesang je Antwort in Prozent



befragt wurden etwa 160 HPI-Studenten im Mai 2012

Die Umfrage führten Leonid Berov und Patrick Rein

Schlaues

Koffein, oh du mein bester Freund!

Von Cola, Mate und deren Risiken

Dass Informatiker auf Kaffee und Cola zurückgreifen, um konzentrierter zu arbeiten und auch mal ein bisschen länger wach zu bleiben, ist mehr als nur ein Gerücht. Das zeigt sich auch am HPI: Ulfs Kaffee ist hoch frequentiert und der Automat im Keller des Vorlesungsgebäudes scheint mit seinen Cola-Flaschen guten Umsatz zu machen.

Doch jedes Wundermittel hat auch seine Schattenseiten: Krebserregend, knochenzersetzend, kariesfördernd. Gesundheitliche Bedenken gibt es viele. Wie viel dahinter steckt, ist meist nicht ganz klar.

Der Aufsteiger

In den letzten Jahren erfreut sich neben der klassischen Cola ein ganz anderes koffeinhaltiges Getränk wachsender Beliebtheit. Die trendigen Glasflaschen mit dem gelb-blauen Etikett sieht man in jeder Vorlesung auf den Pulten stehen und in Bachelorprojekt-Räumen stapeln sich die gelben Kästen nahezu bis an die Decke. Die Rede ist natürlich von Club Mate. Ein Getränk, das ohne Werbung auskommt und allein von Mundpropaganda und dem menschlichen Drang Neues auszuprobieren lebt. So exklusiv, dass man schon einmal zwei oder drei Läden abklappern muss, bevor man eine Flasche ergattert. Club Mate ist so cool, dass sie noch nicht einmal schmecken muss. „Ich kenne niemanden, dem es auf Anhieb geschmeckt hat“, sagt der Chef der Mate-Brauerei selbst und der Geschmack beschreibt sich am besten durch den markeneigenen Slogan „Man gewöhnt sich dran“. Und doch trinken es alle.

Mate als Krebsrisiko?

Club Mate besteht, wie jeder weiß, der schon einmal das Etikett gelesen hat, aus der Mate-Pflanze. Diese wächst in Südamerika und wird dort auf Grund ihrer anregenden Wirkung schon seit Jahrhunderten zu Tee verarbeitet. Mate heißt das Getränk übrigens deswegen, weil der Tee traditionell aus einem ausgehöhlten Flaschenkürbis getrunken wurde und der heißt auf Spanisch „mate“.

Was schon seit Jahrhunderten getrunken wird, kann ja eigentlich nicht so schädlich sein, denkt man sich da. Doch gilt Mate-Tee im Allgemeinen als risikoerhöhender Faktor für Speiseröhrenkrebs. Das National Cancer Institute in Maryland hat in einer Studie im Jahr 2008 die Menge an polyzyklisch aromatischen Kohlenwasserstoffen (PAK) im Mate-Tee untersucht. Diese Kohlenstoff-Verbindungen sind schuld daran, dass angekohltes Grillfleisch als krebserregend gilt und eben diese Verbindungen wurden in sehr hohen Konzentrationen im Mate-Tee gemessen. Aber heißt das, dass auch Club Mate karzinogen wirkt?

Club Mate enthält, ausgehend von den Höchstgehalten der amerikanischen Studie, heruntergerechnet auf die in Club Mate enthaltene Matemenge, PAK-Werte von unter 0,5 Mikrogramm je Liter Getränk. Die Höchstgrenzen der EU für PAK in Lebensmitteln liegen zwischen 10 und 1 Mikrogramm je kg. Dabei sind die PAK-Werte für Säuglingsanfangsnahrung auf 1 Mikrogramm/kg beschränkt. Club Mate liegt also deutlich darunter. „Gesundheitliche Bedenken beim Genuss von Club Mate in Bezug auf PAKs würden wir deshalb verneinen“, kommentierte Marcus Loscher, Geschäftsführer der Mate-Brauereien, diese Werte. Und wenn man sich auf die PAK-Werte beschränkt, ist Club Mate laut EU folglich sogar als Muttermilch-Ersatz geeignet.

Cola

Mit krebserregenden Inhaltsstoffen hatte Coca Cola auch zu tun. So wurde kürzlich der von Cola verwendete Farbstoff in den USA in die Liste der krebserregenden Stoffe aufgenommen. Um einen Aufdruck „Achtung, krebserregend“ auf dem Etikett zu vermeiden, hat man den Farbstoff in den USA gewechselt. In Deutschland allerdings nicht, denn

Zu viel Cola ist schädlich.

Zu viel Wasser auch, sagt Coca Cola.

hier gilt der Farbstoff nicht als krebserregend. Auch Coca Cola weist darauf hin, dass die in den USA nachgewiesene krebserregende Wirkung durch die in Cola enthaltenen Mengen unmöglich ausgelöst werden könne.

Doch scheint der Farbstoff in der Cola ohnehin eins der geringeren Probleme zu sein. Von dem hohen Zuckergehalt abgesehen, will eine Studie der American Society for Clinical Nutrition im Jahr 2006 einen Zusammenhang zwischen dem vermehrten Konsum von Cola und einer geringeren Knochendichte bei Frauen festgestellt haben. Die in Cola enthaltene Phosphorsäure erschwere die Aufnahme von Calcium und führe so zu Osteoporose. Warum das nur bei Frauen der Fall zu sein scheint, ist unbekannt. Coca Cola weist den Vorwurf zurück und beruft sich auf die International Osteoporosis Foundation (IOF), die angibt, dass es keine überzeugenden Beweise für einen schädlichen Einfluss von Cola-Getränken auf die Knochengesundheit gibt. Die IOF vermutet, dass, wer viel Cola trinkt, dafür das Milch-Trinken vernachlässigt. Inwiefern sich diese Vermutung jedoch auf andere Milchprodukte übertragen lässt, ist nicht wirklich klar. Scheibchen gefrosteter Cola auf dem Brot statt Butter und Käse? Auflauf mit Cola überbacken? Coca Cola versichert: „Alle Produkte aus dem Hause Coca-Cola können im Rahmen einer abwechslungsreichen und ausgewogenen Ernährung ihren Platz haben."

Im Februar 2012 zeigte sich in Neuseeland, wie gefährlich eine unausgewogene Ernährung wirklich sein kann. Eine 30-jährige Frau starb an einem Herzinfarkt, nachdem sie täglich bis zu zehn Liter Cola getrunken hatte. Schlecht ging es ihr wohl schon länger, aber dass es an der Cola liegen könnte, hat schlichtweg keiner vermutet. Geraucht hat sie natürlich auch und wenig gegessen. Coca Cola wies darauf hin, dass so ziemlich alles in größeren Mengen gefährlich sei, auch das Trinken von zu viel Wasser könne das Gleichgewicht des Körpers schädigen und ihm wichtige Nährstoffe entziehen.

Club Mate: hübsch, trendy und nicht krebserregend.

Dehydrierung mindert Leistungsfähigkeit

Als Beweis für die Schädlichkeit von Cola kann das Beispiel der Neuseeländerin freilich nicht dienen. Denn zehn Liter Club Mate wären auf Dauer auch nicht gesundheitsfördernd. Absolut sicher können wir uns über die gesundheitlichen Auswirkungen von Lebensmitteln nie sein. Es empfiehlt sich wahrscheinlich, zu seinem Glas Cola mal eine Käse-Stulle oder zu seiner Cola-Stulle mal ein Glas Milch zu sich zu nehmen. Denn solange man in Maßen konsumiert und sich ausgewogen ernährt, kann einem nicht viel passieren.

Vielleicht ist es trotzdem eine gute Idee, ab und zu einfach auf ein Glas Wasser zurückzugreifen. Das ist mit hoher Wahrscheinlichkeit nicht krebserregend und hat auch nichts mit Osteoporose zu tun. Und wenn es auch nicht besonders viel Koffein enthält, so ist Wasser auch für seine erfrischende und belebende Wirkung bekannt. Denn Dehydrierung vermindert erwiesenermaßen die mentale Leistungsfähigkeit. Und das schon in einem Stadium bevor wir überhaupt Durst verspüren. Wenn man sich also nicht mehr erinnern kann, wie eine Komposition in ein Klassendiagramm eingezeichnet wird, und die Entscheidung für einen Platz im Vorlesungssaal schwer fällt, dann könnten das Zeichen einer leichten Dehydrierung sein.

Also mehr trinken und auch mal an Calcium denken. Und wenn es dann doch etwas später wird, kann man ruhig mal auf etwas Koffeinhaltiges zurückgreifen.

— *Susanne Bülow*

Was tun, wenn das Silicon Valley zu weit weg ist?

Ein erfolgreiches Studium am HPI eröffnet einem Absolventen viele Tore: Die Einen suchen sich ihren Weg in die Selbstständigkeit, die Anderen gehen in die Forschung. Wieder andere versuchen sich in der Wirtschaft als IT-Consultant oder Berater. Zu guter Letzt gibt es noch jene, die sich schon von vornherein um eine Einstellung in einem bereits etablierten IT-Unternehmen bemühen. Die Größe und die Dichte der IT-Unternehmen in einem Gebiet spielen bei der Wahl des Arbeitsplatzes oft eine große Rolle. Je größer ein Unternehmen ist, desto größer ist meist der Bedarf an Informatikern. Je höher die Dichte an IT-Unternehmen, desto höher ist nicht nur die Auswahl an möglichen Arbeitsplätzen für den Absolventen, sondern auch der Komfort der Mitarbeiter. Denn wo viele etablierte Unternehmen sitzen und um die besten Absolventen konkurrieren, sind Recruiter oft dazu geneigt, potenzielle Mitarbeiter mit vielen kleinen Extras zu locken. Und wo findet man wohl mehr IT-Giganten als im Silicon Valley?

Der Begriff des Silicon Valley, erstmals 1971 in einem Artikel der Electronic News erwähnt, stellt eine Komposition der Worte Silizium (engl. Silicon), ein in der Computer- und vor allem Halbleiterindustrie essenzieller Rohstoff, und Valley, dem Santa-Clara-Valley, dar. Doch bereits lange vor der Entstehung des Wortes wurden die ersten Grundsteine für die Geburt des Silicon Valley gelegt. Als in den 50er Jahren die Stanford University bei Palo Alto mit finanziellen Problemen zu kämpfen hatte, versuchte Fred Terman, Professor der Universität, diese durch die Gründung des Stanford Industrial Parks zu lösen. Die Universität besaß über 3.000 Hektar ungenutztes Land, jedoch kein Geld um mit dem Wachstum der Nachkriegszeit mithalten zu können. Ein Verkauf dieses Landes kam jedoch aufgrund der Bestimmungen über den Nachlass von Leland Stanford, dem Gründer der Universität, nicht infrage. Bereits einige Jahre zuvor hatte Terman seine Studenten William Hewlett und David Packard bei der Gründung ihres IT-Unternehmens Hewlett-Packard, dem ersten Start-up des Silicon Valley, unterstützt. Durch die positiven Erfahrungen motiviert, beschloss er nun die ungenutzten Flächen der Stanford Universität günstig an Tech-Unternehmen und Start-ups zu verpachten. Aufgrund der günstigen Bedingungen siedelten sich bald viele Unternehmen entlang des Freeway 101 zwischen San Mateo und San Jose an. Diese hatten nicht nur die Möglichkeit Kooperationen mit den naheliegenden Universitäten und Forschungseinrichtungen wie etwa Stanford University, Berkeley (University of California) oder Xerox PARC einzugehen, sondern konnten auch die besten Absolventen und Wissenschaftler direkt von den Universitäten abwerben.

Heute findet sich im Silicon Valley nahezu alles, was in der IT-Branche Rang und Namen hat. Neben IBM, Oracle, Google, Microsoft, Adobe und Facebook haben auch deutsche Firmen wie SAP, Siemens und die Deutsche Telekom Laboratories dort ihre Niederlassungen. Wer eine Anstellung im Valley ergattern

möchte, muss eine Koryphäe seines Spezialgebiets sein. Denn die Jobs sind beliebt und der Andrang ist groß, nicht nur in den USA. Jeder zweite Angestellte kommt aus dem Ausland, nahezu jeder dritte Ingenieur und Wissenschaftler im Silicon Valley ist Inder. Hat man es aber geschafft sich gegen all die Konkurrenz durchzusetzen, so kann man wohl beruhigt von sich behaupten, es „geschafft“ zu haben. Denn so schwer es auch sein mag im Valley überhaupt aufgenommen zu werden, so schwer wird es einem fallen, dieses wieder zu verlassen. Von den Arbeitsbedingungen, die dort in manch einem IT-Unternehmen herrschen, kann man nur träumen. Während man in Unternehmen anderswo verzweifelt nach einer Kaffeemaschine sucht, sind dort Fitnesscenter mit Personal-Coachs, Massagezentren, Arztpraxen, Schlafbereiche, Teeküchen mit Snacks und Knabbereien sowie Cafeterien mit selbst angebauten, organischen Lebensmitteln oder ein Barbecue in der Mittagspause nicht selten zu finden – selbstverständlich zur kostenlosen Nutzung. Doch nicht nur die gutenArbeitsbedingungen fesseln die Ingenieure und Softwareentwickler ans Silicon Valley, sondern auch die Gehälter.

Ein Universitätsabsolvent ohne Berufserfahrungen kann ein Einstiegsgehalt von circa 80 000 bis 90 000 Dollar erwarten, nach einigen Jahren Erfahrung nahezu 150 000 Dollar. Hinzu kommt der ständig wachsende Kampf der Unternehmen um IT-Talente. Sogenannte „Headhunters“ sind stets auf der Suche nach neuen Gesichtern und Talenten für ihr Unternehmen. Die Suche findet dabei hauptsächlich in den Reihen der Konkurrenz statt. Als talentierter Mitarbeiter eines namenhaften Unternehmens im Valley liegt einem nicht selten ein Geschenkkorb an der Türschwelle oder ein seriöses, gut dotiertes Jobangebot im (elektronischen) Briefkasten. Mit etwas Geschick kann man sich dann Aktienanteile, höhere Gehälter und einen Sign-on- bzw. Bleibebonus erhandeln.

Kaum verwunderlich also, dass das Silicon Valley ein begehrter Arbeitsplatz ist. Alles schön und gut, aber muss man sich als Softwareentwickler wirklich ans andere Ende der Erde begeben, um solche Arbeitskonditionen genießen zu dürfen? Gibt es ähnliche Strukturen und Bedingungen wie im Silicon Valley nicht auch in Europa, vielleicht sogar in Deutschland?

Und tatsächlich gibt es auch in Europa Versuche einer Imitation des Silicon Valley. Vom Erfolg des Originals inspiriert, hoffen viele kleinere europäische, aber vor allem auch deutsche Bezirke durch den Aufbau eines IT-Clusters von der Erfolgswelle mitgerissen zu werden. Solche Imitatoren sind das Silicon Valley North in Kanada, das indische Silicon Valley in Banglore, das Silicon Oasis in Dubai, das Silicon Alps in Österreich, das IT-Cluster Rhein-Main-Neckar um Frankfurt, Karlsruhe und Heidelberg oder auch das Isar Valley in München, um einige zu nennen. Jedes von ihnen hat seine Eigenheiten und alle sind aufgrund unterschiedlicher Kontexte und aus unterschiedlichen Voraussetzungen heraus entstanden. Zwei dieser Imitatoren sind vor allem für Softwareentwickler sehr interessant.

Das in Europa wohl am schnellsten an Bedeutung gewinnende IT-Cluster ist das Silicon Roundabout an der Old Street in Shoreditchim im östlichen London. Im Gegensatz zum Silicon Valley ist es ein Cluster von Tech-Start-ups und kleinen bis mittelgroßen Unternehmen.

Blueseed – Hacken vor der Küste

Auch wenn ein großer Teil der Mitarbeiter in etablierten IT-Unternehmen aus dem Ausland kommt, ist die Situation von nicht-US-Bürgern im Silicon Valley sehr schwer. So erhalten viele nach dem Studium trotz Qualifikation und Jobangebot oft keine Arbeitserlaubnis. Abhilfe soll dabei das Blueseed-Projekt leisten. Dabei wird 1000 ITlern und deren Startups auf einem gigantischen Schiff 22 km vor der US-Küste Platz geboten. Dort werden sie mit Gleichgesinnten wohnen und arbeiten, auf engstem Raum. Sobald das Unternehmen Fuß fasst und die Bedingungen der US-Behörde erfüllt, können die Geeks eine Arbeitserlaubnis beantragen und ans Land ziehen. Das Projekt soll auf legalem Weg und in Übereinkommen mit den Grenzbehörden umgesetzt werden. Noch fehlen jedoch die Investoren für das 25-Millionen-Dollar-Projekt.

Alles begann mit einem weniger ernst gemeinten Tweet des damaligen Chief Technical Officers der Travel-Site „Dopplr“ im Jahr 2008:„'Silicon Roundabout': the ever-growing community of fun start-ups in London's Old Street area“. Damals waren es gerade einmal 15 Start-ups an der Old Street, heute sind des weitmehr als 200 (Stand: 2011), wobei etwa 90% der Unternehmen eine Mitarbeiterzahl von 10 nicht überschreiten. Die meisten dieser Unternehmen sind hochgradig spezialisiert und füllen mit ihren innovativen Ideen die neu entstehenden Marktlücken der dotcom-Gesellschaft. Während das Cambridge Cluster „Silicon Fen“ Jahrzehnte benötigte, um erste größere Erfolge verzeichnen zu können, hat sich in Shoreditch innerhalb der letzten vier Jahre die Zahl der Start-upsmehr als verzehnfacht. All das ohne staatliche Unterstützungoder Verbindung zu Universitäten. Zu den größten Erfolgsgeschichten des Silicon Roundabout zählen unter anderem last.fm, eine Musik Webseite, die für 280 Millionen Dollar an CBS Interactive verkauft wurde, die Reiseseite Dopplr, Moo.com und der ursprünglich österreichische Mode-Startup Lookk.

Ein Problem, dem sich nahezu alle Startups an der Old Street stellen müssen, ist der Mangel an Investoren. Im Gegensatz zu den amerikanischen Investoren, die nicht nur Geld, sondern auch über Erfahrung verfügen, bildet sich in Großbritannien erst nach und nach eine technologiebezogene Venture-Capital-Industrie. In den letzten Jahren haben immer mehr IT-Giganten wie Facebook und Google die britischen Start-ups wahrgenommen und investieren in diese. Auch die britische Regierung erkennt die Bedeutsamkeit des Silicon Roundaboutfür ihre Wirtschaft. Durch Anpassungen des „Entrepreneur Visa“, die Zurverfügungstellung einer High-Speed-Breitband-Übertragung von über 100 Mbps und das Hinzuziehen der Management Expertise der McKinsey & Company, versucht die Regierung die „East London Tech City“, wie sie das Silicon Roundabout gerne nennt, zu unterstützen.

Eine interessante Innovation Shoreditches ist das Unternehmen Tech Hub. Tech Hub bietet Coworking-Space für kleine Tech-Startups und jene, die es werden wollen. Jeder mit einer Idee und mit wenig Geld kann sich flexibel monatsweise (oder für langfristige Projekte auch jahresweise) einen Schreibtisch samt Besprechungsraum mieten. Doch es geht weniger nur um den „Workspace“, als vielmehr um das Networking und die Gemeinschaft. Anstatt im Keller sitzend konzentriert auf den Monitor zu starren und auf der Tastatur herum zu hämmern, kann man hier mit Gleichgesinnten in lichtdurchfluteten Großraumbüros zusammenarbeiten. Das Wissen, nicht der Einzige zu sein, der eine gut bezahlte, solide Festanstellung in einem etablierten Unternehmen für eine verrückte Idee aufgegeben hat, muntert auf und gibt einem Kraft. Schließlich sind die anderen am Nachbartisch mindestens genauso verrückt wie man selbst. „Alchemie“, so nennt es die CEO des Tech Hubs, Elizabeth Varley.Es ist ein Platz für kreative Zusammenarbeit, ein Paradies für Nerds, an dem man seine Ideen störungsfrei entwickeln kann oder, wie Google-Chainman Eric Schmidt sagte, ein „Richtfeuer“ für die Tech-Start-up-Community.

Im Gegensatz zum Silicon Valley gehen 90% der IT-Studenten in London nach dem Abschluss lieber in eine Bank oder als Beraterin die Wirtschaft anstatt in ein Tech-Unternehmen. Um dem entgegenzuwirken, arbeiten die Firmen im Silicon Roundabout vielgleisig. Zum einen veranstalten sie regelmäßig Recruitingevents wie das „Silicon Milkroundabout“ - eine Konteraktion gegen die „Milkaround“-Roadshows an Universitäten, bei denen Banken und Beratungsunternehmen die besten Absolventen noch bevor diese auf den Arbeitsmarkt kommen anwerben. Zudem werden fette Fische, wie etwa der technische Leiter bei Google als CTO abgeworben. Diese ziehen wiederum die talentierten und ehrgeizigen Absolventen automatisch an. Wer will nicht bei jemandem lernen, der maßgeblich an Googles Suchanwendungen mitentwickelt hat? Ein weiteres Mittel um IT-Talente anzulocken, sind die gemeinsamen Aktivitäten der Roundaboutler. Natürlich kann es sich kein Unternehmen in Shoreditch leisten, den Mitarbeitern Essen aus aller Herren Länder in insgesamt 18 verschiedenen Cafés anzubieten oder umweltfreundliche Elektroautos für eine

kleine Spritztour während der Arbeitszeit zur Verfügung zu stellen, wie es etwa Google tut. Kleinere Gemeinschaftsaktivitäten wie das Fußballturnier ‚Silicon Kickabout', gemeinsame Roulette-Abende, Opensource-Stickkreise oder auch wöchentliche Trinkabende, die ‚Silicon Drinkabouts' sind dagegen gang und gäbe.

Sticken mit offenen Quellen

Der Open-Source-Stickkreis ist ein Kunstprojekt.
Einmal die Woche setzten sich die ITler zusammen und besticken bunte Stofffetzen mit Definitionen aus einem digitalen Lexikon – Buchstabe für Buchstabe, Wort für Wort.

Trotz alledem ist Shoreditch und vor allem die Old Street nicht unbedingt der Traumarbeitsplatz eines jeden und vor allem nicht der Ort, an dem man eigentlich sein Start-up aufbauen möchte. Klar, die Mieten sind günstig. Jedoch sind die Gebäude teilweise heruntergekommen, manche Ecken im Vergleich zu dem modernen High-Tech-Silicon Valley eher alt und schäbig. Wieso also gehen viele Start-ups nach wie vor zum Silicon Roundabout? Wieso hat sich das Silicon Roundaboutausgerechnet dieses Eck Londons ausgesucht?

Trotz seines etwas heruntergekommenen Äußeren haben Shoreditch und die Old Street ihre Vorzüge. Shoreditch war im 18.Jahrhundert ein Weberviertel. Dort, wo heute die Start-ups sitzen, standen damals die Werkstätten der Weber. Nach und nach folgten ihnen zunächst die Designer, später auch die Verleger. Kreativität und Einfallsreichtum gehörten also von Beginn an in dieses Viertel. Während man einerseits mit London, einer Weltmetropole, einen global günstigen Standort gefunden hat, profitiert man andererseits an der Old Street von den günstigen Mieten und dem unkonventionellen Lifestyle des Londoner Ostens – optimal für junge, kreative Teams. Der Lärm, die Lebendigkeit der Großstadt und die sogenannte „underground-attitude" der Old Streetfördern zudem die Kreativität und das Andersdenken. So bricht beispielsweise an einer der Wände des Büros von Unruly-Media, ein Tech-Unternehmen an der Old Street, ein weißer Hirsch durch die Wand, während sich an einer anderen Wand die Tischtennisschläger-Kollektion der Mitarbeiter mit kleinen Comicporträts der Besitzer befindet. Shoreditch hat im Gegensatz zum Silicon Valley den Coolness-Faktor. Ein Viertel voller „trendy geeks" statt verklemmter, sozial inkompetenter Nerds.

Wem London allerdings immer noch zu weit weg ist, bietet sich in Zukunft auch bei uns eine Alternative: das Silicon Sanssouci bei Berlin-Potsdam. Ein Cluster, das sich noch in seiner Geburtsphase befindet. Nicht nur die Nähe zur Hauptstadt und der Zugang zu deren Infrastrukturen und den dort angesiedelten Unternehmen, sondern auch die über 100 Forschungsinstitutionen in dem Gebiet Berlin-Potsdam begünstigen die Entstehung des Clusters. Vor allem Start-ups der Softwarebranche haben hier gute Aussichten und werden vielfach durch Capital Venture Unternehmen wie Hasso-Plattner-Ventures finanziert und unterstützt. Das kreative und innovative Denkvermögen wird wiederum an der HPI-Academy geschult und optimiert. Nun soll durch die Eröffnung des SAP-Innovationszentrums zusätzlich eine Brücke zum Silicon Valley geschlagen werden, die Kooperationen mit dem MIT, der Berkeley Universität und der Standford Universität pflegt. Allein seit 2008 zählte die Industrie- und Handelskammer weit mehr als 1.300 Startups in der Web- und IT-Branche. Laut kalifornischen Investmentbankern hat das Cluster hohes Potenzial und gilt als eines der wenigen, die dem Silicon Valley die Stirn bieten können und „die Welt verändern" werden. Beste Voraussetzungen also für das Silicon Sanssouci eventuell auch bald etwas von dem Erfolg, wie ihn das Silicon Valley schon lange genießt und das Silicon Roundabout beginnt zu genießen, abzubekommen.

Auch wenn das Silicon Valley einem IT-Absolventen wohl viele Perspektiven bietet und ihm das Tor zum Erfolg öffnen kann, geht man dort schnell in der Masse der Mitarbeiter eines solchen IT-Giganten unter. Lange ist das kalifornische Valley nicht mehr das einzige erfolgreiche Cluster. Gerade für junge ITler mit Talent und innovativen Ideen bieten oft schon die kleineren aufblühenden IT-Cluster wie das Silicon Roundabout und das Silicon Sanssouci gute Möglichkeiten.

— *Suhanyaa Nitkunanantharjah*