

ti&m expert series



HACK AN APP IN ONE WEEK

Das Kursbuch
zur IT-Projektwoche

EDITOREN: CHRISTOF MEILI
THOMAS WÜST



ti&m expert series



HACK AN APP IN ONE WEEK

Das Kursbuch
zur IT-Projektwoche

EDITOREN: CHRISTOF MEILI
THOMAS WÜST



**Editoren: Christof Meili
Thomas Wüst**

ti&m AG

Hack an app in one week

– DAS KURSBUCH ZUR IT-PROJEKTWOCHE –

ti&m expert series

Books on Demand

Ute Blewonska trug das Kapitel über „Frauen in der Informatik“ bei. Ute ist diplomierte Elektrotechnikerin und wickelt erfolgreich Software-Projekte und Beratungsmandate ab.

Peter Böhnlein verfasste neben der Einleitung die Geschichte der Informatik, ihre Grundlagen sowie den Ausblick und die Informationen zu den IT-Berufen. Ausserdem integrierte er die Einzelbeiträge zu diesem Buch. Peter arbeitet als Principal bei der ti&m AG und berät Kunden bei der Lösung unterschiedlichster Problemstellungen in der Informatikwelt.

Von **Stefan Kinigadner** stammt die Einführung zu den agilen Vorgehensweisen. Stefan ist diplomierter Informatiker und wendet begeistert und erfolgreich agile Vorgehensweisen in seinen Projekten an.

Die gestalterischen Aspekte der Informatik beschrieb **Thomas Kneubühl**. Er hat unter anderem einen Masterabschluss in Human Computer Interaction Design und arbeitet als User Experience-Architekt bei der ti&m AG.

Stephan Meier führt uns in die Themen OS X, Erarbeiten von Inhalt sowie digitale Fotografie und digitales Video ein. Unser Profifotograf ist ausgewiesener IT-Projektmanager mit Fokus auf Datenhaltung und Business Intelligence.

Aus der Feder von **Florian Reiser** stammt das Kapitel über Social Media. Florian ist Diplom-Betriebswirt und baut als Software-Entwickler, Datenbankdesigner und Projektleiter industrietaugliche Lösungen.

Luisa Sartori prüfte und verbesserte mit ihrem end-to-end Lektorat massgeblich die sprachliche Qualität. Luisa sorgt mit ihrer Ausbildung in den für die ti&m AG wichtigen Sprachen Deutsch und Englisch dafür, dass alle kundenrelevanten Dokumente auch korrekt ausgeliefert werden können.

Mit **Ingmar Stein** trug ein ausgewiesener Fachexperte sein Wissen im Bereich „mobile“ bei. Als promovierter Informatiker hat er neben seinem breiten Überblick über die wichtigen Entwicklungen im mobilen Umfeld auch die Fähigkeit, entsprechende Lösungen zum Nutzen unserer Kunden zu implementieren.

Fabian Uehlin erarbeitete das Kapitel mit seinen Beispielen zur Sprache HTML. Fabian als diplomierter Nachrichtentechnik-Ingenieur entwickelt Software selbst, berät Kunden bei schwierigen Problemen und leitet entsprechende Projekte.

Philipp Weber beschrieb das Beispielprojekt und stimmte dieses Buch mit der Kurswoche ab. Mit seinem Hintergrund als Primarlehrer, der sich dann auch noch zum diplomierten Informatik-Ingenieur weiterbildete, bringt er beste Voraussetzungen mit, um diese Kurswoche für eine Oberstufenklasse zu konzipieren.

Michael Wolski steuerte den Inhalt und die Code-Beispiele zur Gestaltung der grafischen Oberfläche bei. Der Wirtschaftsinformatiker deckt breite Teile der Software-Entwicklung wie Analyse und Design, Implementierung und Test ab. Darüber hinaus leitet er auch entsprechende Projekte.

Von **Tomas Zelenay** stammen die Angaben zu Windows. Tomas bringt als diplomierter Informatiker das Fachwissen mit, um Software-Lösungen

kundenspezifisch zu entwerfen und zu bauen.

Roland Ziegler hat sich die Beispiele und Übungen zur Programmiersprache Javascript ausgedacht. Roland als ausgewiesener Medieninformatiker entwickelt Software mit Schwerpunkt Content Management.

Impressum

Als Schweizer KMU „leisten“ wir uns zwei nicht-kommerzielle Projekte, nämlich art@work (www.ti8m.ch/art_at_work) und „Hack an app in one week“ (www.ti8m.ch/school). Unseren diesbezüglich eher beschränkten finanziellen Rahmen kompensieren wir durch Innovation, Ideenreichtum und der Freiheit, Aktionen, von denen wir überzeugt sind, einfach umzusetzen. Während wir mit art@work die Auseinandersetzung mit innovativer Kunst suchen, streben wir mit „Hack an app in one week“ kreative, nachhaltige und spannende Jugendförderung an. Beides gehört vermeintlich nicht in den Berufsalltag der klassischen Ingenieure, doch sind wir überzeugt, dass uns ohne freie, teils wilde und junge Kunst der Mut für Ideen abhandenkommt und ohne für Informatik begeisterte Tüftler unser Ingenieursnachwuchs ausbleiben wird. Beides brauchen wir, um langfristig und nachhaltig erfolgreich sein zu können. Die beiden Themen Kunst und Jugendförderung sind mittlerweile Teil unserer ti&m-Kultur und unseres Selbstverständnisses als kreative Ingenieure und Entwickler.

Die vorliegende Publikation „Hack an app in one week“ ist Teil unserer Produktionsreihe ti&m expert series. Sie richtet sich nicht wie gewohnt an Ingenieure oder Prozessverantwortliche, sondern an Jugendliche.

Wir betreiben vier sinnverwandte Formate, die moderne und innovative Technologien und Lösungsansätze auf unterschiedliche Art und Weise aufgreifen:

- *ti&m special*: Unser jährliches Journal zu aktuellen und relevanten IT-Themen, in welchem Praktiker und Visionäre in spannenden Artikeln kontrovers, provokant und kompetent diskutieren.

- *ti&m forum*: Unsere halbjährlich stattfindende Vortrags- und Diskussionsreihe, die den direkten Austausch mit Vordenkern aus Unternehmen und Forschung fördert.
- *ti&m seminare*: In unseren Seminaren und Kursen werden die interessantesten Themen und Trends in der IT von innovativen Köpfen vorgestellt – direkter Wissenstransfer auf höchstem Niveau.
- *ti&m expert series*: Detailliertes Technologie-Know-how von unseren Experten. Das Wissen und die Erfahrung, die unsere Mitarbeitenden in vielen Kundenprojekten erworben haben, bringt diese Reihe kompakt und präzise zu Papier: fundiertes Wissen vom Experten für den Experten.

Damit ist *ti&m – technology, innovation & management* nun in der spannenden Lage, innovative Technologien aus einer Vielzahl von Perspektiven zu beleuchten. Fundierte Fakten werden ergänzt durch eine kontroverse Diskussion und visionäre Betrachtung. Dabei setzt *ti&m* auf die Ausarbeitung von interessanten Schnittmengen zwischen Anwenderstrategien, Angeboten der Software-Hersteller, Beratungsansätzen sowie der angewandten Forschung. Der gemeinsame Nenner ist hierbei die Praxisrelevanz und das Erfolgspotenzial der behandelten Themen.

Die so geschaffene Plattform ergänzen wir durch einen „Domänensprung“, indem wir unserem Innovations- und Kunstprojekt *art@work* Raum für eine kreative oder auch provokante Sicht auf unser jeweiliges Leitthema einräumen. *art@work* ist eine Ausstellungsform, die moderne Kunst in die aktuellen Räume und Tätigkeiten unserer Ingenieure und Berater integriert.

Das enorme Potenzial der „IT-Werkstatt Schweiz“ stellt für uns als IT-Firma eine grosse Herausforderung dar. Im Spannungsfeld zwischen stetiger Innovation, erhöhten Anforderungen und dem Bedürfnis nach Stabilität und

Sicherheit gilt es für uns, gemeinsam mit unseren Kunden, Partnern und Mitbewerbern eine IT-Vision für die Zukunft zu entwickeln und umzusetzen.

Das Ziel von *ti&m expert series* - „Hack an app in one week“ ist ausnahmsweise nicht, Experten auf eine (noch) höhere Wissens Ebene zu bringen, sondern junge Menschen für unser spannendes und vielseitiges Berufsfeld zu begeistern. Die Inhalte leiten durch eine jugendgerechte Sprache und eine einfache Darstellung von relativ komplexen technischen Themen Jugendliche dazu an, ihre ersten Schritte in unserem Metier zu gehen, und die Gadgets, mit denen sie heute oft schon täglich umgehen, besser zu verstehen. Die Begeisterung entsteht dabei im Umgang mit den Geräten. Der Computer, das Mobiltelefon werden von Kommunikations- und Spielgeräten zu Werkzeugen, die es ermöglichen, etwas Kreatives zu schaffen.

Damit bringen wir den Funken, Technik nicht mehr nur benutzen, sondern auch beherrschen zu wollen, zum Überspringen.

Wir beschränken uns allerdings bewusst nicht nur auf technische Aspekte. In der modernen IT-Welt geht technisches Fachwissen Hand in Hand mit kommunikativen Fähigkeiten und dem Vermögen, sich in die Verhaltensweisen der Nutzer hineinzusetzen. Daher beschäftigt sich ein grosser Teil des vorliegenden Buches mit Themen wie Usability, dem Scrum-Vorgehensmodell und gestalterischen Fragen.

Eine derart ganzheitliche Darstellung für ein jugendliches Publikum ist auf dem heutigen Buchmarkt wohl einzigartig. Dahinter steht auf der einen Seite die Intention, die Kids nicht mit trockenem Stoff zu überfrachten. Auf der anderen Seite wollen wir ihnen ein möglichst vollständiges Bild unserer Branche vermitteln, die für ihre Innovationsleistung auf Nachwuchs angewiesen ist. Wir sehen Jugendförderung

daher als Zukunftsförderung. Als Technologieunternehmen, das die Innovation fest im Namen verdrahtet hat, stehen wir zu dieser Verantwortung.

Die Idee wie auch das Konzept dieses Buches resultieren aus unserem langjährigen Engagement und den Erfahrungen im Bereich der Nachwuchs- und Frauenförderung. Wir beteiligen uns seit Jahren am nationalen Zukunftstag, in dessen Rahmen Kinder mit erfahrenen Mobile-Ingenieuren Mini-Apps bauen oder mit Security-Architekten über Social Media diskutieren können. Daraus wiederum haben wir ein Unterrichtsmodul entwickelt, mit dem wir an Schulen gehen und Kurse auf Basis unserer *ti&m seminare* veranstalten.

Entscheidend für das Gelingen einer Idee ist der erste Schritt. Diesen haben uns Herr Tarzis Lüscher und Herr Emil Klaus von der Bezirksschule Lenzburg ermöglicht. Ohne ihre Bereitschaft „das Eis zu brechen“, hätten wir viel länger gebraucht, um dieses Projekt umzusetzen. Dafür, dass sie spontan und vertrauensvoll den ersten Pilotkurs mit uns durchgeführt haben, möchten wir uns sehr herzlich bedanken.

Abschliessend wünsche ich Ihnen und Ihren Kindern viel Spass beim Lesen und kreative spannende Stunden bei der Anwendung der erworbenen Kenntnisse.

Ihr Thomas Wüst – CEO, ti&m AG

Dieses Buch widmen wir allen Schülerinnen und Schülern,
die mit ihrem Interesse, ihrer Neugier und ihrer Kreativität
helfen,
die Informatik auf immer wieder neuen Wegen
weiterzuentwickeln.

Inhaltsverzeichnis

- 1 Einleitung
 - 1.1 Was macht eigentlich ein Informatiker?
 - 1.2 Und was muss er dazu können?
 - 1.3 Was will dieses Buch erreichen und für wen ist es gedacht?
 - 1.4 Wie ist dieses Buch aufgebaut?

- 2 Was - wir machen ein IT-Projekt!?
 - 2.1 Wie ist denn die Aufgabenverteilung?
 - 2.2 Und was bauen wir?
 - 2.3 Und was hat das agile Vorgehensmodell Scrum damit zu tun?

- 3 Was sollte ich über Informatik im Allgemeinen wissen?
 - 3.1 Wie ist die Informatik eigentlich entstanden?
 - 3.2 Hat die Informatik wichtige Grundlagen?
 - 3.3 Was bedeutet in diesem Zusammenhang „mobile“?

- 4 Wie gehe ich mit einem Betriebssystem um?
 - 4.1 Was ist ein Betriebssystem?
 - 4.2 Wie ist ein Betriebssystem denn aufgebaut?
 - 4.3 Wie bediene ich das Betriebssystem OS X?
 - 4.4 Warum soll ich die Festplatte pflegen?

- 5 Content - was für Content?
 - 5.1 Was verstehen wir unter Content?
 - 5.2 Und wie erarbeite ich die verschiedenen Inhalte?

5.3 Wie baue ich meine Geschichten denn auf?

6 Haben digitale Fotografie und Videos etwas mit Informatik zu tun?

6.1 Wie kann ich besser fotografieren und filmen?

6.2 Und was bedeutet das für meine Arbeit am Computer?

6.3 Was sollte ich bei der Bildbearbeitung beachten?

6.4 ...und was bei der Videobearbeitung?

6.5 Kann ich mit Bildern Geschichten erzählen?

7 Hat die Informatik auch eine gestalterische Seite?

7.1 Wie gehe ich beim Gestalten einer Webseite vor?

7.2 Welche Rolle spielen Publikum, Aufgaben und Ziele?

7.3 Was sollte ich bei der Navigation beachten?

7.4 Was bedeutet Informationsdesign?

7.5 Was ist für das Layout wichtig?

7.6 Visuelles Design - was ist das?

8 Wie erstelle ich die Benutzeroberfläche?

8.1 Wieso sehen alte Programme so langweilig und kompliziert aus?

8.2 Wie baue ich denn für ein iPhone die Benutzeroberfläche?

8.3 Wie funktioniert der Interface Builder?

8.4 Wo finde ich den Programmcode?

9 Was gibt es Wichtiges zu HTML?

- 9.1 Was sind die Voraussetzungen, um eine Webseite zu erstellen?
- 9.2 Was ist HTML?
- 9.3 Meine erste Webseite!
- 9.4 Spezialfrage: Wie behandle ich Umlaute?
- 9.5 ...und wie Überschriften?
- 9.6 Wie formatiere ich meine Texte ansprechend?
- 9.7 Kann ich auch etwas Farbe ins Spiel bringen?
- 9.8 Wie setze ich Verlinkungen ein?
- 9.9 Ich möchte Bilder einsetzen - was nun?
- 9.10 Was kann ich denn nun alles?

10 Programmiersprachen - was hat es damit auf sich?

- 10.1 Was ist eigentlich eine Programmiersprache?
- 10.2 Kann ich mal selbst programmieren? Vielleicht mit JavaScript!?
- 10.3 Wie lade ich JavaScript-Funktionen?
- 10.4 Wie kann ich mit JavaScript HTML-Elemente verändern?
- 10.5 Wenn das Programm selbst entscheiden soll, was dann?
- 10.6 Was kann ich denn nun schon alles?

11 Präsentationstechniken und Informatik?

- 11.1 Wie bereite ich mich auf eine Präsentation vor?
- 11.2 Wie läuft eine gute Präsentation ab?
- 11.3 Welche Rolle spielt hier die Visualisierung?
- 11.4 Auf was muss ich bei der Präsentation achten?
- 11.5 Was war nochmal das Wichtigste?

12 Was bleibt sonst noch zu sagen?

Anhang A Gibt es sonst noch Interessantes zum Internet und zum Web?

- A.1 Was ist das Internet?
- A.2 Was ist der Browserkrieg?
- A.3 ...und was hat es mit dem Web 2.0 auf sich?
- A.4 Das Internet der Dinge: Was bringt es uns?
- A.5 Wie sieht das Internet von morgen aus?

Anhang B Social Media - schon wieder so ein Schlagwort!?

- B.1 Gibt es verschiedene Arten von Social Media?
- B.2 Kann mir Social Media etwas nützen?
- B.3 Welche Auswirkungen haben Social Media?
- B.4 Sagt der Gesetzgeber auch etwas dazu?

Anhang C Wäre Informatik etwas für mich?

- C.1 Was sind die wichtigsten Berufsbilder?
- C.2 Kann ich in der Informatik Karriere machen?
- C.3 Welche Chancen haben Frauen in der IT?

Anhang D Wie bediene ich das Betriebssystem Windows?

- D.1 Wie ist Windows entstanden?
- D.2 Was sind die wichtigsten Windows-Grundlagen?
- D.3 Wie funktioniert das Dateisystem von Windows?
- D.4 Was können der Explorer und der Arbeitsplatz?
- D.5 Was hat es mit der Systemsteuerung auf sich?
- D.6 Wozu sind Shortcuts gut?
- D.7 Gibt es sonst noch Wichtiges zu Windows?
- D.8 Bist du bereit für ein paar Übungen?

Anhang E Checklisten zur Präsentation

- E.1 Checkliste zur Vorbereitung der Präsentation

[E.2 Checkliste zum Verhalten während der Präsentation](#)

[E.3 Checkliste bei Lampenfieber](#)

[Anhang F Glossar - Was bedeuten die vielen Begriffe und Abkürzungen?](#)

[Anhang G Verzeichnisse](#)

[G.1 Literaturverzeichnis - Wo kann ich noch mehr über Informatik erfahren?](#)

[G.2 Abbildungsverzeichnis](#)

[G.3 Tabellenverzeichnis](#)

[G.4 Verzeichnis der Code-Beispiele](#)

[Anhang H Index](#)

1 Einleitung

*"Niemand wird in zehn Jahren Software-Produkte haben. Jeder wird Produkte und Dienste haben."
Steve Ballmer, Microsoft, über das Softwaregeschäft im Internet-Zeitalter.*

In diesem Kapitel betrachten wir die Aufgaben, die Informatiker typischerweise erledigen, was sie dazu benötigen und was dich in diesem Buch sonst noch alles erwartet.

1.1 Was macht eigentlich ein Informatiker?

Das Tätigkeitsfeld eines Informatikers ist sehr breit. Vorab aber ein Wort, weshalb wir vom *Informatiker* sprechen - und nicht von der *Informatikerin*: Selbstverständlich sprechen wir in diesem Buch immer Vertreter beiderlei Geschlechts an. Es würde aber die Lesbarkeit des Textes nicht unbedingt erhöhen, wenn wir immer von „der Informatiker/die Informatikerin“ sprechen würden. Wir hoffen, das ist ok für dich.

Nun, was macht ein Informatiker so den ganzen Tag? Er kümmert sich darum, dass wir mit Computern arbeiten können. Das umfasst zum Beispiel

- den Bereich der Entwicklung: Hardware und Software müssen hergestellt werden.
- den Bereich des Betriebs - z.B. in einem Rechenzentrum oder
- die Unterstützung bei der Anwendung von Computern.

Selbstverständlich setzt der Informatiker in seiner täglichen Arbeit auch selbst Computer ein - er ist also selbst auch Anwender der Informatik.

Um seine Aufgaben gut erfüllen zu können, arbeitet er eng mit Mitarbeitern aus anderen Fachgebieten zusammen. Bei der Entwicklung von Hardware sind das z.B. Elektrotechniker, Elektroniker, Physiker und Werkstoffwissenschaftler – beim Betrieb können dies Experten für Arbeitsabläufe sein.

Seine Schwerpunkte liegen häufig in der Erstellung der Software. Dazu stimmt er sich mit den späteren Benutzern zu deren Anforderungen ab, liefert dem Projektmanager den erwarteten Aufwand und erstellt die Vorgaben für die Programmierer. Der Informatiker programmiert aber oft auch selbst, definiert dazu die geeigneten Datenstrukturen (um Informationen im Computer sinnvoll speichern zu können) und entwirft eine stabile Software-Architektur. Das Spannende an der Informatik liegt sicher darin, dass du echte Produkte herstellst – seien sie für den kommerziellen Einsatz oder „just for fun“ – und das in einem sehr schnellen und schnelllebigen Umfeld. Gerade die neuen mobilen Endgeräte zeichnen sich durch rasante Entwicklungen – Stichwort „Apps“ – aus.

Daneben gibt es noch einige weitere wichtige Aufgaben im Berufsbild eines Informatikers: Seine Lösungen müssen die *Sicherheitsanforderungen* der Gesetze und der Benutzer erfüllen, sie sollen *benutzerfreundlich* sein und auch in der Erstellung und im späteren Betrieb *nicht zu teuer* werden.

1.2 Und was muss er dazu können?

So vielfältig die Aufgaben eines Informatikers sind, so vielfältig sind auch die Anforderungen an ihn – bzw. so abwechslungsreich gestaltet sich sein Job.

Von zentraler Bedeutung sind sicher *Interesse an Computern* und *Interesse an Menschen*. Für ersteres

benötigt er ein gutes *Abstraktionsvermögen* – viele Dinge wie die Software kann man ja nicht anfassen – und auch die Fähigkeit, *präzise* zu arbeiten: Computer sind nun mal dumme, mathematische Maschinen, die ohne Verständnis Befehl für Befehl „blind“ ausführen.

Für den zweiten Punkt ist es sehr wichtig, gut und gerne zu *kommunizieren*. Der Informatiker arbeitet oft in einer komplexen, heterogenen Welt mit Kolleginnen und Kollegen aus unterschiedlichen Nationen, aus unterschiedlichen Kulturkreisen, mit unterschiedlichem Wissensstand und so weiter zusammen. Damit das funktionieren kann, muss er als Vermittler zwischen diesen Welten agieren können; bei ihm laufen schliesslich viele Fäden zusammen, da er ja die Anforderungen und Bedürfnisse aller Teilnehmer erfassen und unter einen Hut bringen muss. Bei grösseren Projekten bedingt das auch, dass er gut Englisch lesen, sprechen und schreiben kann: Englisch ist in internationalen Projekten die gemeinsame Sprache, und auch Fachliteratur ist häufig in Englisch geschrieben.

Der dritte wichtige Aspekt sind *Kreativität* und ein Gefühl für *Ästhetik*. Das Design von Benutzeroberflächen wird stark von den diesbezüglichen Fähigkeiten der Entwickler geprägt. Aber auch eine Software-Architektur kann ästhetisch sein. Das Gute daran ist, dass eine solche Architektur dann meist auch gut im Sinne der Informatik ist. Ein Analogon aus der Medizin: Gesunde Organe sind immer auch schön anzusehen, von Krebs befallene sehen hingegen fürchterlich aus.

In Kapitel Anhang C „Wäre Informatik etwas für mich?“ gehen wir auf diesen Themenkomplex genauer ein.

1.3 Was will dieses Buch erreichen und für wen ist es gedacht?

Dieses Buch ergänzt und unterstützt die IT-Projektwoche der ti&m AG. Es enthält einen Teil der in der Projektwoche vermittelten Inhalte, aber auch Zusatzinformationen, die während der Workshops nicht (direkt) behandelt werden.

Das Buch richtet sich an die Teilnehmer der Projektwoche – also an dich. Informatiker gehen miteinander locker und informell um. Daher sprechen wir dich in diesem Buch in der Du-Form an – wir hoffen, das ist ok!

Bitte setze dieses Buch als Arbeitsmittel ein: Du solltest dir also Notizen reinschreiben, die darin enthaltenen Aufgaben durcharbeiten und es auch als Nachschlagewerk verwenden.

1.4 Wie ist dieses Buch aufgebaut?

In dieser Einleitung erfährst du, wozu dieses Buch dient und was dich darin erwartet. Der Rest dieses Buches ist in die folgenden drei Hauptbereiche unterteilt:

- Grundlagenteil
- Software-Entwicklungsteil
- Ausblick und Abschluss

Im Grundlagenteil gehen wir auf das Projekt dieser Woche ein ([Kapitel 2](#)), geben dir einen Überblick über die Informatik ([Kapitel 3](#)) und zeigen dir, wie du mit einem Betriebssystem umgehst ([Kapitel 4](#)). Ausserdem erarbeiten wir mit dir, wie du zu sinnvollem Inhalt für dein Projekt kommst ([Kapitel 5](#)), fordern deine kreative Ader unter anderem mit einem eigenen Kapitel zu digitaler Fotografie und digitalem Video heraus ([Kapitel 6](#)) und überlegen uns, was bei der Gestaltung der Benutzeroberfläche zu beachten ist ([Kapitel 7](#)).

Der Software-Entwicklungsteil führt dich in [Kapitel 8](#) in die Erstellung einer Benutzeroberfläche ein: Was ist dabei zu beachten und wie mache ich das? [Kapitel 9](#) und 10 tauchen dann endlich in die Tiefen der Programmierung ab. Mit HTML und JavaScript kannst du professionelle Apps und Webapplikationen selbst entwickeln! Im Ausblick und Abschluss zeigen wir dir, wie du deine Leistungen gut präsentieren kannst ([Kapitel 11](#)) und welche Entwicklungen sich in der Informatik für die Zukunft abzeichnen ([Kapitel 12](#)).

Mit einem Verzeichnis wichtiger Begriffe der Informatik und der verwendeten Quellen bieten wir dir eine einfache Möglichkeit, unklare Begriffe nachzuschlagen und bei Interesse auf weitere Informationen über die Informationstechnik zuzugreifen.

Fünf „Bonustracks“ behandeln ausserdem noch für dich hoffentlich interessante Themen, die uns auch wichtig erscheinen, in der Projektwoche nicht oder nur teilweise angesprochen werden (können).

Du findest im folgenden Text immer wieder in eckigen Klammern geschriebene Zahlen, wie z.B. [\[10\]](#). Diese Zahlen verweisen auf die Quelle, aus der wir die entsprechenden Informationen haben. Du findest diese Quellen in Anhang F. Dies gilt sowohl für Texte als auch für Bilder und Grafiken. Ist keine Quelle bei den Bildern und Grafiken angegeben, so hat sie die ti&m AG erstellt.

Entscheidend für das Gelingen einer Idee ist der erste Schritt. Diesen haben uns Herr Tarzis Lüscher und Herr Emil Klaus von der Bezirksschule Lenzburg ermöglicht. Ohne ihre Bereitschaft „das Eis zu brechen“, hätten wir viel länger gebraucht, um dieses Projekt umzusetzen. Dafür, dass sie spontan und vertrauensvoll den ersten Pilotkurs mit uns durchgeführt haben, möchten wir uns sehr herzlich bedanken.

Dieses Buch in der nun vorliegenden Form wurde durch Mitarbeiterinnen und Mitarbeiter der ti&m AG ermöglicht. Wir möchten uns bei ihnen dafür herzlich bedanken.

Und nun geht's wirklich los: Wie ist die Informatik eigentlich entstanden?

Aber halt: Zu deinem Vergnügen haben wir auch Lustiges und Witziges in dieses Buch gestreut - und: Da ist ja bereits etwas davon...

☺ *Just smile!*

Sind Computer männlich oder weiblich?

Wie man weiss, werden Schiffe seit langem als weiblich betrachtet. Kürzlich gab eine Gruppe (ausschliesslich männlicher) Informatiker bekannt, dass auch Computer als *weiblich* anzusehen seien. Folgende Gründe haben sie zu dieser Schlussfolgerung gebracht:

1. Ausser ihrem Schöpfer versteht niemand ihre interne Logik.
2. Die Muttersprache, in der sie sich mit anderen Computern verständigen, ist für alle anderen unbegreiflich.
3. Die Meldung „Falscher Befehl oder Dateiname“ ist etwa so informativ wie: „Wenn du schon nicht weisst, warum ich wütend auf dich bin, dann verrate ich es dir bestimmt nicht.“
4. Selbst kleinste Fehler, die man macht, werden im Langzeitspeicher abgelegt und später wieder abgerufen.
5. Sobald man sich fest mit einer eingelassen hat, gibt man sein halbes Einkommen für Zubehör aus.

Eine andere Gruppe (ausschliesslich weiblicher) Informatikerinnen ist jedoch der Meinung, Computer seien als *männlich* zu betrachten. Ihre Gründe lauten:

1. Sie speichern viele Daten, haben aber trotzdem keine Ahnung.
 2. Eigentlich sind sie dazu da, Probleme zu lösen, aber die Hälfte der Zeit sind sie selbst das Problem.
 3. Sobald man sich für einen entscheidet, merkt man, dass man kurze Zeit später ein besseres Modell hätte kriegen können.
 4. Um ihre Aufmerksamkeit zu wecken, muss man sie anmachen.
 5. Grosse Energiespitzen setzen sie für den Rest des Abends ausser Gefecht.
-

2 Was - wir machen ein IT-Projekt!?

*"Ich verstehe die Technik nicht und werde es auch nie. Aber man muss auch nicht.
Es reicht, wenn man versteht, was sie für einen tun kann."
Rupert Murdoch, Medienmogul.*

In diesem Kapitel stellen wir dir das IT-Projekt vor, das wir während unserer Projektwoche bearbeiten werden. Dazu betrachten wir die anstehenden Aufgaben und die Aufgabenverteilung und befassen uns mit dem agilen Vorgehensmodell Scrum.

2.1 Wie ist denn die Aufgabenverteilung?

In einem IT-Projekt sind verschiedenste Menschen involviert. Jeder übernimmt dabei Aufgaben, die seiner Ausbildung und seinen Fähigkeiten entsprechen; wir sprechen dann auch davon, dass jemand bestimmte (Projekt-) Rollen übernimmt. Typischerweise bestehen IT-Projekte aus etwa fünf bis zehn Personen. Im Extremfall kann auch nur eine Person an einem Projekt arbeiten. Dann sprechen wir von einer „One-Man-Show“.

Die folgenden Rollen gibt es üblicherweise in einem IT-Projekt:

- Projektleiter – der Chef
- Business-Analyst – der Kommunikative
- Programmierer – der Techniker
- Designer – der Kreative
- Tester – der Genaue

Der Projektleiter hat die Verantwortung für das Projekt. Das bedeutet vor allem, dass das Projekt in der geplanten Zeit, in der geforderten Qualität und mit dem gesprochenen Geld erfolgreich abgeschlossen wird (siehe auch [Abbildung 1](#)).



[Abbildung 1: Das „Magische Dreieck“ \(Quelle: \[26\]\)](#)

Zentrale Aufgaben sind *Teamführung*, *Koordination* und *Planung* der Projektarbeiten und die *Kommunikation* mit den sogenannten Stakeholdern. Stakeholder sind Personen, die ein berechtigtes Interesse am Verlauf und Ergebnis des IT-Projektes haben.

Ein *Business-Analyst* ist eine Person, die die Anforderungen und Wünsche der Auftraggeber - des Business - so aufbereitet, dass die Programmierer verstehen, was sie entwickeln sollen. Im Business arbeiten in der Regel Personen, die von Software-Entwicklung nicht unbedingt viel verstehen, z.B. Buchhalter, Banker, KV-Angestellte oder auch Personen aus dem Gewerbe oder der Industrie. Sie brauchen eine solche Ansprechperson.

Das Ergebnis der Arbeit der Business-Analysten ist ein Anforderungskatalog oder auch Pflichtenheft. Der *Scrum-Prozess* verwendet dafür den Begriff *Product Backlog*.

Programmierer erstellen basierend auf den Anforderungen des Business die Computerprogramme. Dazu muss ein Programmierer die *Programmiersprache* beherrschen, mit der er mit einem Computer kommunizieren kann. Es gibt mittlerweile unzählige Programmiersprachen, die sich je nach Anwendungsgebiet unterscheiden. Bekannte Programmiersprachen sind Java, C++, HTML oder SQL. Mit Java und C++ werden Programme oder Spiele programmiert, während HTML für die Gestaltung von Webseiten verwendet wird.

SQL bedeutet „Structured Query Language“, was so viel bedeutet wie „strukturierte Abfragesprache“. Sie verwenden wir für die Abfrage von Datenbanken.

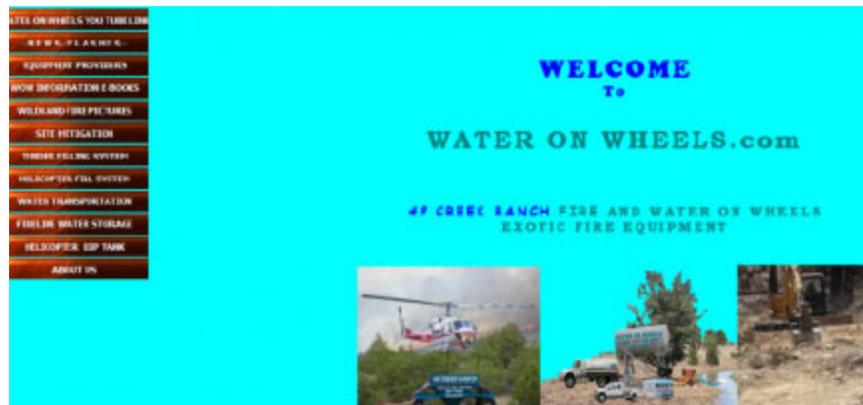
[Tabelle 1](#) zeigt dir ein Beispiel in SQL. Weitere Informationen zu den Programmiersprachen findest du in [Kapitel 10](#), „Programmiersprachen - was hat es damit auf sich?“.

| Befehle in SQL | Sinngemässe Übersetzung |
|--|---|
| <pre>SELECT A.A_NR, A.A_NAME, A.A_PREIS FROM ARTIKEL As A WHERE A.A_PREIS > 200 ORDER BY A.A_PREIS</pre> | <p>Hole (Artikel-)Nummer, (Artikel-)Name und (Artikel-)Preis aus der Artikel-Tabelle für alle Artikel mit einem Preis höher als 200 sortiert nach dem Preis</p> |

[Tabelle 1: Ein SQL-Beispiel](#)

Der *Designer* hat die Verantwortung für das Aussehen, das „Design“ einer Webseite oder eines Computerprogrammes. Es ist eine sehr kreative Aufgabe, die nicht unbedingt Computerfachkenntnisse voraussetzt. Der Designer gestaltet beispielsweise einen Dialog, setzt Farben und Schriftarten richtig ein und gibt dem Programm das „gewisse Etwas“.

\$\$\$!



[Abbildung 2: Ein Beispiel für schlechtes Design](#)

Die Benutzer von Programmen nehmen als erstes das Design wahr, welches sie auch beurteilen. Gefällt ihnen ein Design nicht, auch wenn das Programm noch so gut programmiert ist, fällt ihre Beurteilung schlecht aus (siehe [Abbildung 2](#) und [Abbildung 3](#)).

Weitere Informationen zum Thema *Design* findest du im [Kapitel 7](#), „Hat die Informatik auch eine gestalterische Seite?“.



[Abbildung 3: Ein Beispiel für gutes Design](#)

Eine wichtige Rolle in jedem IT-Projekt spielt der *Tester*. Er überprüft mit manuellen und automatisierten Tests die Funktionalität eines Computerprogrammes.

Oft schleichen sich Fehler (Bugs) ein, welche der Programmierer übersehen hat. Ein guter Tester findet viele Fehler und meldet diese dem Programmierer, damit er die Fehler korrigieren kann (Bug fixing). Je früher Fehler gefunden werden, desto kostengünstiger können sie behoben werden. Deshalb ist es wichtig, Software möglichst frühzeitig intensiv zu testen.

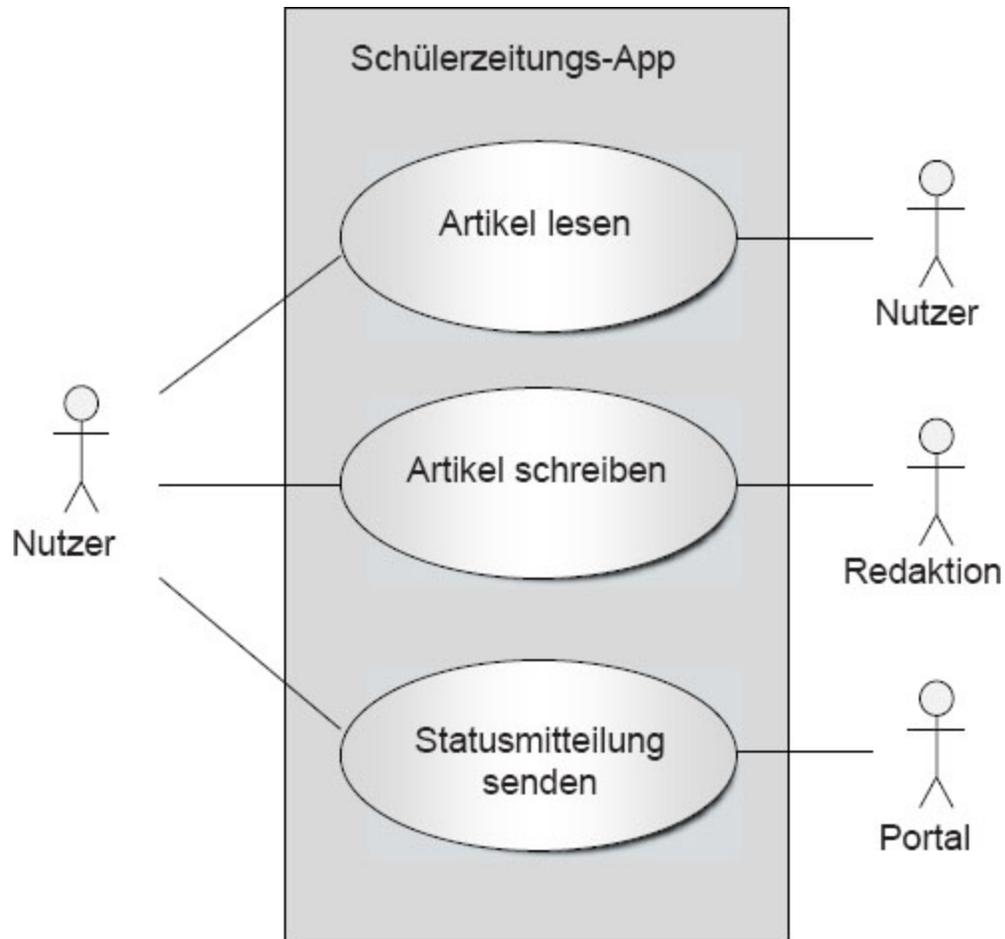
Hier ein Beispiel für einen Fehler, der dank Simulationstests erkannt und behoben werden konnte ([36]): „Beim Kampfflugzeug F-16 brachte der Autopilot das Flugzeug in Rückenlage, sobald der Äquator überflogen wurde. Dies kam daher, dass keine negativen Werte für die Breitengrade als Eingabedaten berücksichtigt waren. Dieser Fehler wurde (leider erst) sehr spät in der Entwicklung des

Kampfflugzeuges F-16 anhand eines Simulators entdeckt und beseitigt.“

Das nächste Kapitel geht nun auf das Thema unseres IT-Projektes ein.

2.2 Und was bauen wir?

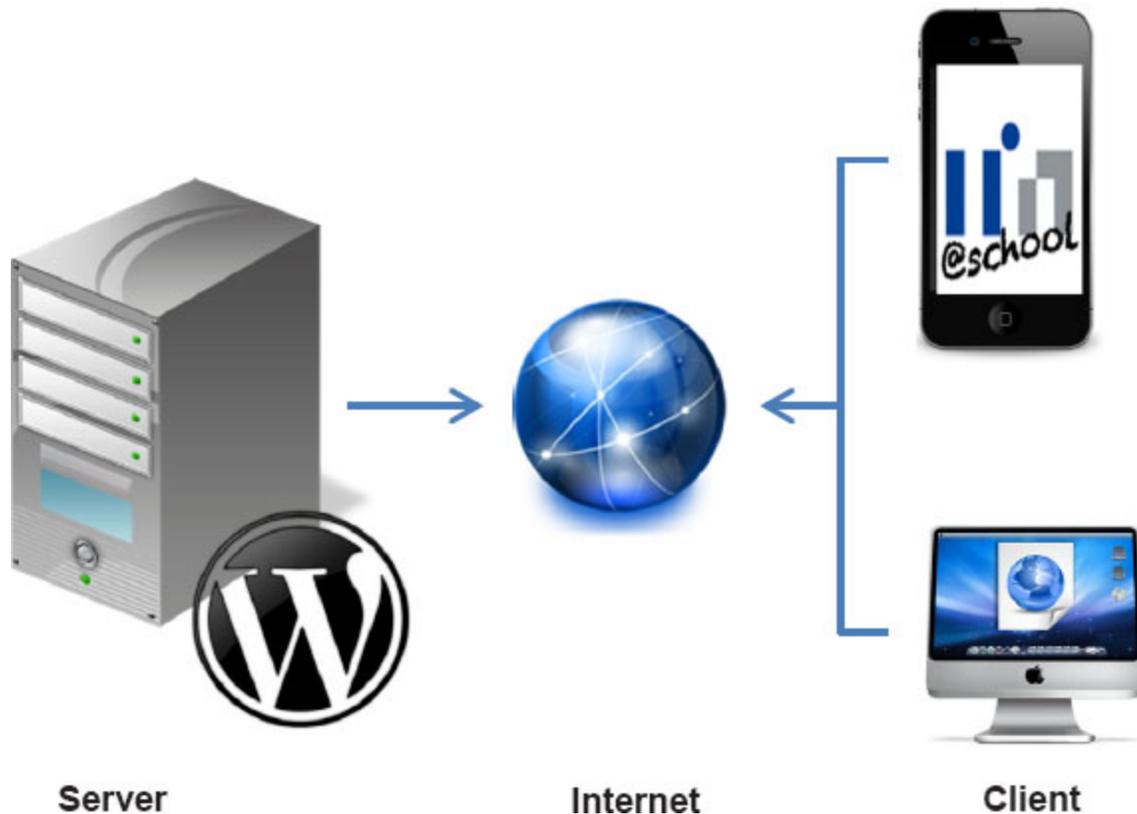
Wir wollen eine Schülerzeitung programmieren und dabei möglichst viele Aspekte der Informatik behandeln. Die Schülerzeitung wird sowohl mit einem Internet Browser als auch mit mobilen Geräten – in unserem Falle dem iPhone – benutzbar sein. Das *Use Case-Diagramm* ([Abbildung 4](#)) gibt dir einen ersten Eindruck, was unser Programm für die Schülerzeitung alles anbieten soll. Neben den *Anforderungen* müssen wir uns auch überlegen, welche Struktur die Lösung – deine Software – haben soll. Dazu gehört auch, dass wir die beteiligten Komponenten kennen: Welche Geräte und Netzwerke müssen wir einsetzen, um die Anforderungen aus den Use Cases überhaupt erfüllen zu können? Einen ersten Überblick dazu gibt dir der *Architekturplan* in [Abbildung 5](#).



[Abbildung 4: Use Case-Diagramm](#)

Für diese beteiligten Komponenten müssen wir nun die entsprechende Software erstellen. Bevor wir damit beginnen, muss noch die *Entwicklungsumgebung* mit der Programmiersprache bestimmt werden.

Client-Server-Architektur



[Abbildung 5: Ein erster Entwurf eines Architekturplans](#)

Auf der *Server-Seite* entscheiden wir uns für Wordpress. Es ist eines der grössten und beliebtesten Blog-Portale im Internet – und steht uns kostenlos zur Verfügung.

Auf den *Client-Seiten* setzen wir einerseits einen normalen Internet Browser ein, um auf die Wordpress-Seite zuzugreifen, und andererseits eine mobile App für das iPhone.

Deine Aufgaben, um die Software zu erstellen, gliedern wir in der Projektwoche wie folgt:

Am ersten Tag – nach einer Einführung in die Informatik und des Internets, deren Geschichte und Berufe – erfährst du viel über die Grundlagen der Gestaltung. „Wie werden Farben optimal eingesetzt?“, „Wie fotografiere und filme ich am effektivsten, um eine Geschichte treffend vermitteln zu können?“, sind nur einige der Fragen, die wir beantworten.