

Softwaretechnologie

Ankündigungen

1

Prof. Dr. rer. nat. Uwe Aßmann
Lehrstuhl Softwaretechnologie
Fakultät für Informatik
TU Dresden
Version 15-1.0, 13.04.15

- Zugehörig zu Modul INF-B-310, INF-D-240
- Kenntnisse sind Voraussetzung zur Teilnahme für Modul INF-B-320



▶ Vorlesungen:

- Prof. Dr. Uwe Aßmann, Nöthnitzer Str. 46, 2. OG, Raum 2087
- Katrin Heber, Sekretärin. 0351 463 38 463
- Sprechstunde Do, 11:00-13:00. Bitte bei Frau Heber anmelden.
- Email katrin.heber@tu-dresden.de. Bitte auch über Frau Heber kontakten, da emails an Prof. Aßmann oft nur verzögert beantwortet werden können

▶ Übungsleitung:

- Dr. Birgit Demuth, Nöthnitzer Str. 46, 2. OG, Raum 2085

▶ Wichtigste Informationsquelle:

- <http://st.inf.tu-dresden.de/> -> Teaching -> Softwaretechnologie
- <http://st.inf.tu-dresden.de/teaching/swt>

Navigation im Web:

Fakultät Informatik

Institut Software- und Multimediatechnik

Professur Softwaretechnologie

Teaching

Softwaretechnologie

Vorlesung und Übungen

3

- ▶ **Vorlesung "Softwaretechnologie":** Konzepte, Überblickswissen zu:
 - Objektorientiertes Programmieren (OOP, aber *keine vollständige* Einführung in Java)
 - Objektorientierter Modellierung (OOM)
 - Objektorientierte Analyse (OOA) + Objektorientiertes Design (OOD)
- ▶ **Übungen "Softwaretechnologie":**
 - Praktische Anwendung von Modellierungstechniken und Java
 - Grundlage für Praktikum "Softwaretechnologie" im 3. Semester
 - Achtung: Ohne regelmässigen Besuch der Übungen ist der Erfolg bei Klausur und Praktikum unwahrscheinlich!
- ▶ **Leistungsnachweis:**
 - **Klausur** (120 Minuten) zu Semesterende (**Prüfung** für INF, MEDINF, WINF, **Schein** für IST)

Voraussetzungen des Praktikums INF-B-320

4

- ▶ Die Kenntnisse, die in INF-B-310 erworben werden, sind, siehe Modulhandbuch, Voraussetzung zur Teilnahme am Praktikum “Softwaretechnologie” INF-B-320 im 3. Semester (Bachelor und Diplom INF, Bachelor MEDINF)
 - Die erfolgreiche Teilnahme an INF-B-320 ohne die vollen Kenntnisse von INF-B-310 ist sehr unwahrscheinlich, da ein kompletter, praktischer, anspruchsvoller Softwareentwicklungsprozess in der Gruppe durchgeführt wird
 - Ein Teilnehmer mit unzureichenden Kenntnissen in Java oder UML schädigt seine Gruppe durch mangelnde Leistungen
 - Muss ein Teilnehmer aus dem Gruppenpraktikum INF-B-320 wegen mangelnder Leistungen ausscheiden, schädigt er seine Gruppe
- ▶ **Vorsicht:** im Praktikum scheidet man aus, wenn man die Projektarbeit nicht absolvieren kann
- ▶ **Vorsicht:** Das Praktikum kann nicht jedes Semester durchgeführt und absolviert werden!
 - Es sind nicht genügend Ressourcen vorhanden, das Praktikum semesterlich durchzuführen
 - Die Klausur Softwaretechnologie kann nach jedem Semester geschrieben werden; das Praktikum nicht!

Verhältnis von ST-Vorlesung und dem Praktikum im Wintersemester

5

- ▶ ST-Vorlesung gibt einen Überblick, aber bereitet nicht speziell für das Praktikum vor
 - Das Praktikum enthält einen kompletten Durchgang durch einen Entwicklungszyklus
 - Semi-realistisch bis realistisch (oft industrielle Kunden)
- ▶ Es lohnt, beides intensiv zu betreiben. **Programmieren heißt Realisieren**
 - Wer sich das Programmieren sparen will, wird große Lücken in seiner beruflichen Praxis haben und seine Ideen nicht wirklich realisieren können
 - es bei Bewerbungen schwer haben, denn Programmierkenntnisse werden vorausgesetzt und Realisierer werden gesucht
- ▶ Wer aber mitprogrammiert, hat viel Gewinn
- ▶ **Parallel: Vorlesung “Programmierung”**
 - http://www.inf.tu-dresden.de/index.php?node_id=3628&ln=de
 - Programmieren *innerhalb von Operationen (Methoden)*
 - Hier: programmieren *außerhalb von Operationen (Methoden)*

Die BaFöG-Falle im 5. Semester

6

- ▶ **Beachte:** Wer im 3. Semester das Praktikum INF-B-320 nicht erfolgreich abschließen kann, kann die 6 Leistungspunkte nicht beim BaFöG am Ende des 4. Semesters geltend machen.
- ▶ Ein Bachelor-Student muss am Ende des 4. Semester erstmals seinen Studienfortschritt dokumentieren, um weiterhin BaFöG zu erhalten.
- ▶ Nachzuweisen sind 100 von 120 LP (für 4 Semester).
- ▶ Achtung: Wird das Praktikum nicht im 3. Semester bestanden, kann es erst im 5. wiederholt werden

Übungen

7

- ▶ Ab erster Woche
- ▶ Bitte dringend noch in JExam in Übungsgruppen eintragen!
- ▶ Übungswoche läuft jeweils von Mo bis Fr (in Synchronisation mit der Vorlesung)

- ▶ Beispiel aus einem der vorigen Jahre
 - 616 Studenten, 544 nehmen an erster Klausur im WiSe teil
 - 39% bestanden
- ▶ Wiederholungsklausur im SoSe
 - 101 Studenten, 1% bestanden
- ▶ Hauptproblem: Viele Studenten können nicht mehr programmieren. Gängige Vorurteile:
 - *“Ich bin Medieninformatiker – ich brauche nicht programmieren”*
 - Fehler: die meisten Medienanwendungen (Websites, alle Spiele) sind komplexe Programme
 - *“Ich werde Softwarearchitekt oder Manager – ich brauche nicht programmieren”*
 - Fehler: Architekten, die nicht mauern können, taugen nichts
 - [Beispiel: Microsoft bestellt keinen zum Manager, der nicht die technischen Vorkenntnisse mitbringt]



Es sind substantielle Java-Programmierkenntnisse nötig,
um die Klausur zu bestehen.

Selbsttests mit dem Praktomaten

9

- ▶ Im Laufe des Kurses werden wir Informationen zum **Praktomaten** veröffentlichen
 - Webbasiertes Selbstlern-System,
 - in das Java-Programme eingetippt werden können
 - das Stil und Übersetzbarkeit prüft
 - und automatisch Tests mit Testdatensätzen anwendet
- ▶ Frühes Feedback über Ihre Programmierfähigkeiten möglich!
 - Die Erfahrung der letzten Jahre zeigt, dass fleissige Benutzung des Praktomaten das Bestehen der Klausur erleichtert.
 - Der Praktomat ist eine Chance für Sie, nutzen Sie sie!
- ▶ Zur Anmeldung Ihren ZIH-Account nutzen
- ▶ Bei Problemen bitte über “Auditorium” melden
 - <https://auditorium.inf.tu-dresden.de/courses/2154394>

<http://praktomat.inf.tu-dresden.de/>

Für Wirtschaftsinformatiker

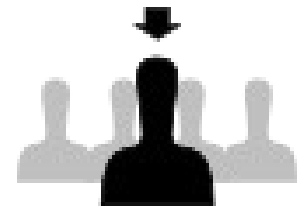
10

- ▶ Anfertigung eines Übungsbeleg
- ▶ 3 selbst gewählte Aufgaben aus der Rubrik "Previous Exams" des Praktomaten
 - <http://praktomat.inf.tu-dresden.de/>
 - Das sind Aufgaben aus alten Klausuren, die Ihre Java-Programmierfähigkeiten und Ihr Verständnis für Design Pattern prüfen
- ▶ Achtung: Sie sollten sich an diese Aufgaben erst ab Übung U07 wagen
- ▶ Zuvor empfehlen wir Ihnen, die einfachen Praktomatsaufgaben zu lösen, auf die in den Übungen hingewiesen wird

Ziel: Die Universität bildet Problemlöser aus

11

- ▶ Die Universität ist keine Schule, sondern eine **Bildungsanstalt**:
 - Sie setzt *selbständige Aktivität* voraus und lehrt *Problemlösen*
 - ◆ Probleme von Menschen erkennen und präzise definieren
 - ◆ selbstständig Wege zur Lösung eines Problems finden
 - Selbstständiges Lernen wird gelernt
 - ◆ kein Standardstoff: Sie bekommen kein Buch vorgelesen, und das war's
 - ◆ selbstständige Literaturerarbeitung von den Folien aus
- ▶ Sie will *Lernliebhaber* und *Literaturfresser* ausbilden
 - Beachten Sie die Lese-Anweisungen, die angegeben werden. Es werden pro Woche 2-4 Kapitel zu lesen sein
 - Steigern Sie also Ihr persönliches Lesetempo
 - Leseleistung: Im Laufe des Studiums sollten Sie lernen, 8 Stunden am Tag zu lernen



Sehr empfohlen für die Technik des wiss. Arbeitens

12

- ▶ **Stickel-Wolf, Wolf. Wissenschaftliches Arbeiten und Lerntechniken. Gabler.** Sehr gutes Überblicksbuch für Anfänger.
- ▶ Stary, Kretschmer: Umgang mit wissenschaftlicher Literatur. Cornelsen. Sehr gutes Buch zum Thema “Lesen”.
- ▶ **Kurs “Academic Skills in Computer Science (ASiCS)”**, mit Teil **“Vorbereitung von Abschlussarbeiten/Forschungskolleg Softwaretechnologie”**
 - Sommersemester, Beginn 14.4.
 - <http://st.inf.tu-dresden.de/teaching/asics>
 - Dienstag, 13:00-15:30, APB-E10
 - Donnerstag, 16:40-18:10 APB-E01

Wie man die Lehrveranstaltung erfolgreich absolviert

13

- ▶ Starte mit der Vorlesung
 - Höre einfach zu.
 - Schreibe auf einem leeren Blatt mit, um das Gehörte in eigenen Worten auszudrücken.
 - Falls du dich nicht recht konzentrieren kannst, versuche, auf ausgedruckten Folien Anmerkungen zu machen.
 - Zeichne Mindmap
- ▶ Zuhause nach der Vorlesung
 - Gleiche deine Notizen mit den ausgedruckten Folien ab.
 - Erweitere die Folien um Anmerkungen.
 - Suche die Buchkapitel, die empfohlen wurden
 - Versuche herauszufinden, was aus der Vorlesung im Buch behandelt wird und was nicht (selektives Lesen von Kapiteln).
 - Schreibe eine Liste von Fragen auf (wiki, blog, Papier)

Während des Semesters:

- ▶ Erstes Lesen (nur das nötigste)
 - Beantworte Fragen, soweit als möglich
- ▶ Rede mit Freund
 - Diskutiere Fragen.
- ▶ Löse alle Übungsaufgaben
- ▶ Löse die Praktomat-Aufgaben
- ▶ Zweites Lesen, auf Klausur vorbereitend (erschöpfendes Lesen)



- ▶ Hauptseite, wichtig:
 - http://www.inf.tu-dresden.de/index.php?node_id=2717&ln=de#s1
- ▶ Lade die Studienordnung herunter (Bachelor)
 - <http://www.inf.tu-dresden.de/content/study/regulations/download/ba-inf/2009/study.de.pdf>
- ▶ Module Bachelor (ähnlich für MedInf-Bachelor und Diplom)
 - <http://www.inf.tu-dresden.de/content/study/regulations/download/ba-minf/2010/study.app.1.pdf>
- ▶ Studienablaufplan Bachelor
 - <http://www.inf.tu-dresden.de/content/study/regulations/download/ba-minf/2009/study.app.2.de.pdf>
 - http://www.inf.tu-dresden.de/index.php?node_id=2774&ln=de
- ▶ Master INF
 - <http://www.inf.tu-dresden.de/content/study/regulations/download/ma-inf/2010/study.app.2.de.pdf>

Anleitung zum Unglücklichsein

15

- ▶ Besuche Übung nur unregelmässig
- ▶ Surfe während Vorlesung
- ▶ Probiere Java-System erst im Juni aus
- ▶ Ignoriere den Praktomat
- ▶ Leihe kein Buch aus, lese nichts
- ▶ Konzentriere dich auf andere Kurse, die schwerer erscheinen
- ▶ Warte mit Lernen bis 2 Wochen vor der Klausur (ST ist ja so einfach...)
 - Achtung: es gibt nur zwei Wiederholungsklausuren (sächs. Hochschulgesetz)
- ▶ Verschiebe die Klausur auf WS
 - Teilnahme am Praktikum erst ein Jahr später möglich





0.1. Softwaretechnologie Ziele und Inhalt

16



Warum ist Softwaretechnologie wichtig?

17

- ▶ Softwaretechnologie ist eine Schlüsselindustrie, da eine *Rationalisierungsindustrie*
 - Die Wohlfahrt eines Landes hängt von der Produktivität ab
 - Nach wie vor entstehen völlig neue Anwendungen in unvorhergesehenen Märkten
 - Google, Google Earth, Video Google
 - Ebay, Amazon
 - Bioinformatik
 - Bauinformatik
 - Maschineninformatik (Virtual Engineering)
 - Digital Pen and Paper
 - Warum so wenig Europäer aktiv in neuen Anwendungen?

Konsumgüter

Investitionsgüter

Rationalisierungsleistung

Software ist die größte gesellschaftsverändernde Kraft heute.
(Anonymous)

Software is eating the world. (Marc Andreessen)

<http://online.wsj.com/news/articles/SB10001424053111903480904576512250915629460>

Warum sind *gute* Softwaretechnologe so wichtig?

18

- ▶ Als Rationalisierungsindustrie ist die IT besonders den Schweinezyklen ausgesetzt:
 - Tal 1993/94, Boom 1997-2000, Tal 2001-03, Boom 2007-heute
 - Viele Firmen in DD suchen momentan gute Softwareingenieure!
- ▶ Einstiegsgehalt pro Jahr brutto [Quelle IX 1/2005]
 - Obere 10%: 50592 Euro
 - Median: 48629 Euro
 - untere 10%: 42900 Euro
 - Projektleiter: 80000 Euro
- ▶ Arbeitsplätze wird es auf lange Sicht in Europa hauptsächlich für den Software-Architekten und Projektleiter geben
 - Programmieren, Testen, ... wird nach Indien oder China ausgelagert
 - Wollen Sie mit 45 arbeitslos sein?
- ▶ Daher muß der Software-Werker ein *guter Softwaretechnologe* werden, dessen Produktivität höher liegt als die der Konkurrenz

Zwei Gruppen von Kenntnissen

19

- ▶ "Software engineering" beinhaltet Wissen über:
 - Softwaretechnologie (Software-Techniken)
 - ◆ Systemanalyse
 - ◆ Systementwurf
 - ◆ Systemimplementierung
 - ◆ Systemwartung
 - Software-Prozesse
 - ◆ Entwicklungszyklus
 - ◆ Lebenszyklen
 - ◆ Projektmanagement
 - ◆ Konfigurationsmanagement
 - ◆ Qualitätsmanagement

Phasen und Meilensteine der Vorlesung

20

- ▶ **Objektorientiertes Programmieren (OOP)**
- ▶ Teil I: bis Ende April: Java I
 - Grundlegende Kenntnisse in Java und jUML
 - Objekte, Klassen, Vererbung, Polymorphie, CRC-Karten
 - Java starten, APIs lesen können, Tests durchführen können
- ▶ Teil II: bis Ende Mai: Java II – Das Objektnetz
 - Generics, Collections, GUI
 - Entwurfsmuster, Frameworks
- ▶ **Objektorientiertes Modellieren (OOM)**
- ▶ III: bis Juni: Objektorientierte Analyse (OOA)
 - Balzert-Methodik, UML
 - Dynamische Modellierung mit Zustandsmaschinen
- ▶ Teil IV: bis Ende Juli: Objektorientiertes Design (OOD) und Projektmanagement
 - Software-Architektur
 - Projektmanagement (PM)

OOP-I: Objekte

OOP-II: Das Netz

OOA

OOD und PM

Warum problemlösende Softwareingenieure gut für die Gesellschaft sind..

21

- ▶ [Charles M. Horton. Opportunities in Engineering, 1920, by Harper & Brothers. Free Gutenberg Book. <http://www.gutenberg.org/ebooks/24681>]
- ▶ Lawyers and politicians have successfully dominated our government from its beginning, with a single beautiful exception in George Washington at one end and another admirable exception in Woodrow Wilson at the other. Washington was a civil engineer, and Wilson, while trained as a lawyer, was an educator. In between these two men there may have fallen a scattering of others who were not lawyers or politicians; the writer is not sure. Of one thing he is sure, however, and that is that engineers in the future will dominate politics to the betterment of the nation as a whole. For engineers are idealists--otherwise they would never have entered upon an engineering career--and idealism has come, as it were, into its own again. The man of vision of a wholesome aspect, the man who can so completely forget himself in his work of service as to engage in tasks whose merits nobody save himself and those pursuing like tasks can or will understand--which is pre-eminently the engineer--is the one man best fitted to administrate in public affairs. More important still than this statement is the fact that the world at large is beginning to realize the truth of it. Engineers as a body stand poised upon the rim of big things. Nor will they as a body stoop to the petty in politics, once they are fairly well launched in active participation of civic affairs. ..

Warum problemlösende Softwareingenieure gut für die Gesellschaft sind..

22

- ▶ Neither their training nor their outlook, based upon their training, will permit it. **For engineers, more than any other group of professional men, are given to "see true." And seeing true, being, as it is, the essence of a full life, is what is needed in our public administrators.**
- ▶ **Engineers belong in civic affairs. The world of humanity needs men of their stamp in high places.** Humanity needs men in control of state and national affairs who would hold the interests of humanity sacred. Engineers are such men. Not that engineers more than any other professional men are sprouting wings--not that. But engineers do see things in their true light--cannot see them in any other light than the one imposed by the law of mathematics, which is that two and two make four, never five or three--and this involuntarily would admit of decisions and grant graces from the point of view of absolute truth, which is, of course, the point of view of humanity--the greatest good for the greatest number. With such men occupying high places in the nation's affairs, the world of men and mankind would leap forward ethically and spiritually at a pace in keeping with the pace at which civilization has progressed under the impetus of engineering thought since the days of Watt. Nobody can deny that progress. Nobody could well deny the fact that ethical progress under engineering guidance would be equally great.



Softwaretechnologie Literatur

23



Das Vorlesungsbuch von Pearson

- 24 ▶ Das Anschaffen von Büchern lohnt sich ▶ für die Softwaretechnik, weil
- das Gebiet sehr breit ist und man immer auf Bücher als Nachschlagewerke zurückgreifen muss. Das Lernen von Folien alleine genügt nicht



- Softwaretechnologie für Einsteiger. Vorlesungsunterlage für die Veranstaltungen an der TU Dresden. Pearson Studium, 2014.
- 20 Exemplare ausleihbar in der Lehrbuchsammlung sowie 1 Präsenz-Exemplar im DrePunct. Jeweils unter ST 230 Z96 S68(2)
- ▶ Enthält ausgewählte Kapitel aus:
- UML: Harald Störrle. UML für Studenten. Pearson 2005. Kompakte Einführung in UML 2.0.
 - Softwaretechnologie allgemein: W. Zuser, T. Grechenig, M. Köhle. Software Engineering mit UML und dem Unified Process. Pearson.
 - Bernd Brügge, Alan H. Dutoit. Objektorientierte Softwaretechnik mit UML, Entwurfsmustern und Java. Pearson Studium/Prentice Hall.
 - Erhältlich bei Thalia (Nürnberger Platz), Thalias (Haus des Buches)

Weiterführende Literatur zum Programmieren

25

- ▶ Java-Bücher:
 - Helmut Balzert. Objektorientierte Programmierung mit Java 5. Elsevier, . www.w3l.de
 - D. Ratz et al: Grundkurs Programmieren in Java. Hanser-Verlag, 2006
 - ◆ Band 1: Der Einstieg in die Programmierung und Objektorientierung,
 - ◆ Band 2: Einführung in die Programmierung kommerzieller Systeme.
 - C. Heinisch, F. Müller, J. Goll: Java als erste Programmiersprache. Teubner.
 - D. Ratz, J. Scheffler, D. Seese, J. Wiesenberger. Grundkurs Programmieren in Java. Hanser-Verlag
 - Hans-Peter Habelitz. Programmieren lernen mit Java: Aktuell zu Java 7 - Keine Vorkenntnisse erforderlich. Galileo Computing
- ▶ Wenn Sie noch mehr programmieren können/möchten:
 - Imperatives Programmieren, Rekursion:
 - D. Boles. Programmieren spielend gelernt mit dem Java-Hamster-Modell. Teubner.
 - Wenn Sie schon imperativ programmieren können:
 - D. Boles, C. Boles: Objekt-orientierte Programmierung spielend gelernt mit dem dem Java-Hamster-Modell. Teubner.

Weitere Literatur

26

- ▶ Eclipse Intro: <http://www.eclipse.org/whitepapers/eclipse-overview.pdf>
- ▶ Andrew Hunt, David Thomas. The pragmatic programmer. Addison-Wesley
 - Deutsch: Der Pragmatische Programmierer. Hanser-Verlag.
 - Sehr schönes Buch mit “Gesetzen des Programmierens”.

Weiterführende Literatur zu UML und OO-Modellierung

27

- ▶ Bernd Oestereich. Die UML-Kurzreferenz 2.3 für die Praxis. 5., überarbeitete Auflage 2009. I, 186 S., broschiert, Oldenbourg, ISBN 978-3-486-59051-7
- ▶ Martin Hitz, Gerti Kappel: UML@Work, dpunkt-Verlag
- ▶ Online-Dokumentation bei der OMG (kostenlos) www.omg.org/uml
- ▶ Dan Pilone, Neil Pitman. UML 2.0 in a nutshell. Free ebook download <http://it-ebooks.info/book/154/>. O'Reilly Media, ISBN: 978-0-596-00795-9, 2005
- ▶ G. Booch, J. Rumbaugh, I. Jacobson: The Unified Modeling Language User Guide, Addison-Wesley 1999.
- ▶ Bernhard Lahres, Gregor Rayman. Praxisbuch Objektorientierung- Von den Grundlagen zur Umsetzung. Galileo Computing. Schönes Buch über OO, nicht auf Java fixiert, breit angelegt.
- ▶ Bernd Oestereich: Objektorientierte Softwareentwicklung mit der Unified Modeling Language, Oldenbourg-Verlag
- ▶ Ken Lunn. Software development with UML. Palgrave-Macmillan. Viele realistische Fallstudien

Weiterführende Literatur zum Gebiet Softwaretechnologie

28

- ▶ *Weiterführende Literatur zum Gebiet Softwaretechnologie. Können Sie anschaffen, wenn Sie ST-II hören wollen*
- ▶ Helmut Balzert: Lehrbuch der Software-Technik, 2 Bände, Spektrum Akademischer Verlag 2000 und 1998. Umfassendes Kompendium.
- ▶ Ghezzi, Jazayeri, Mandrioli. Fundamentals of Software Engineering. Prentice Hall. Sehr gutes, fundamentales, weiterführendes Buch. Klar. Starke Kost.
- ▶ S. Pfleeger: Software Engineering – Theory and Practice. Prentice-Hall. Gutes Buch, breit angelegt.
- ▶ Bernd Brügge, Alan H. Dutoit. Objektorientierte Softwaretechnik mit UML, Entwurfsmustern und Java. Pearson Studium/Prentice Hall.
- ▶ Leszek A. Maciaszek. Requirements Analysis and System Design – Developing Information Systems with UML. Addison-Wesley.

- ▶ Geistesgeschichte Deutschlands und Preußens 1700-1930:
 - Herbert Meschkowski. Jeder nach seiner Facon. Berliner Geistesleben 1700-1800. Piper-Verlag.
 - Herbert Meschkowski. Von Humboldt bis Einstein. Berlin als Weltzentrum der exakten Wissenschaften. Piper-Verlag. Deckt 1820-1930 ab

Appendix

Fähigkeiten des *guten* Softwareingenieurs

31

- ▶ Gute Softwareingenieure **wissen, wie man lernt** (lernen zu lernen)
 - Und das lebenslang
 - Gute Softwareingenieure kennen ihre Lern-Grenzen, -Stärken und Schwächen:
 - Was kann ich wie schnell lernen? [Komplexprüfungen]
 - Wie gut kann ich schätzen?
 - Wie gut kann ich in Abstraktionen denken?
- ▶ Gute Softwareingenieure **gewinnen Erfahrung**
 - Lernen jedes Jahr eine neue Modellier- und Programmiersprache
 - Lerne Projekte kennen (Prozess- und Produktmanagement)
 - Lerne so viele Ideen kennen als möglich
- ▶ Gute Softwareingenieure sind **teamfähig**
 - Die meiste Software wird in Teams erstellt
 - Daher wird das Softwaretechnologie-Projekt in Teams erstellt



Wie erreicht man Bildung?

32

A violin can sing a melody better than the piano can,
and melody is the soul of music.
[Max Bruch, in Fifiend]

Bildung und Meisterschaft

33

- ▶ Christopher Fifiield. Max Bruch: His life and works.
 - Alle, die Melodien lieben, sollten dieses Buch lesen, um zu erfahren, wie man Meisterschaft in dem erreichen kann, wofür man Interesse und Leidenschaft hat. Die Geschichte von Max Bruch und seiner Meisterschaft für Melodien ist eng verknüpft mit dem Schicksal der Deutschen im 19. und 20. Jahrhundert, mit der Entwicklung von Kunst und Wissenschaft in der Zeit um den Ersten Weltkrieg. Vielleicht das allergrößtes deutsche Werk für ein Solo-Instrument ist sein 3. Violinkonzert, Beethoven und Beatles hin oder her. Bruch ist ein Genuss und wird noch in 500 Jahren einer der größten deutschen Komponisten sein! ...Lesen, hören, bilden, um Sonnenuntergänge intensiver zu sehen und Vögel zwitschern zu hören, ...
 - Tipp: von der Oma zu Weihnachten mit dem 3. VK schenken lassen (amazon) und dann abwechselnd mit ihr hören.

Bruch is a music architect. He architects a piece, much in the sense you should be able to do writing, speaking, or programming. He THINKS about his works. He can create tension over a full violin concert, e.g., VC III or Serenade. Prepare yourself to do the same.