



Wie man Wissen und Erfahrung aufbaut, ohne dabei alt zu werden

Die Besten sterben bekanntlich jung

Möglichkeit 1: Beim Alter schwindeln



Möglichkeit 2: Auf Schlaf verzichten



Weitere Möglichkeiten im Überblick

- Ausprobieren und Üben
- Lesen
- Kommunizieren
- Andere scheitern lassen
- Lernen lernen
- Cheaten
- Fokussieren

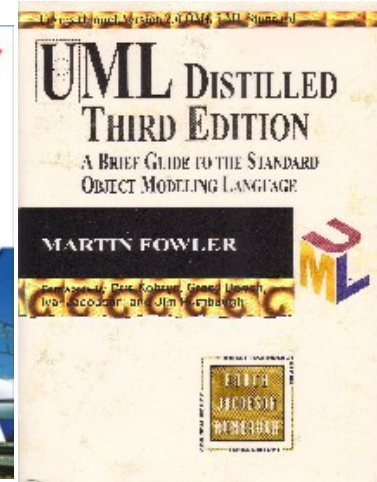
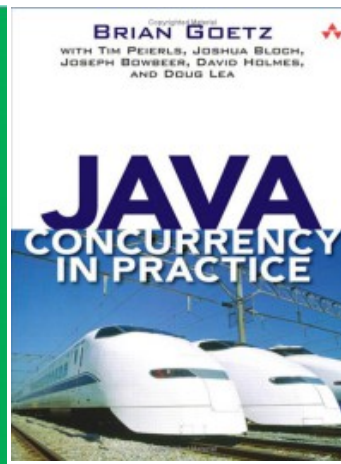
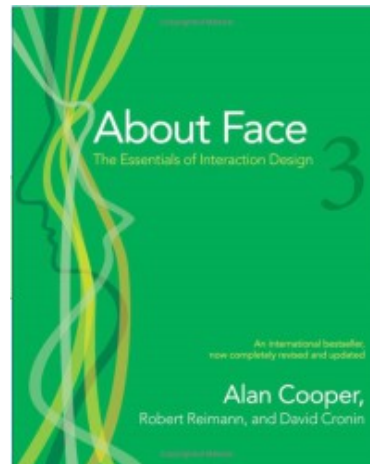
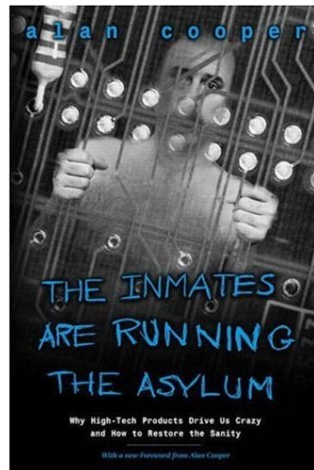
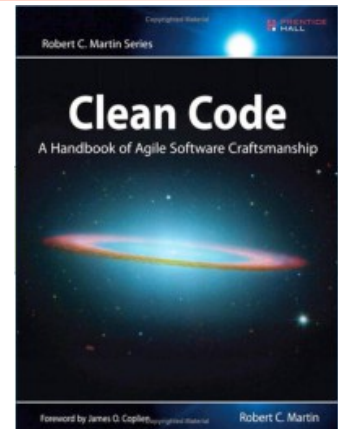
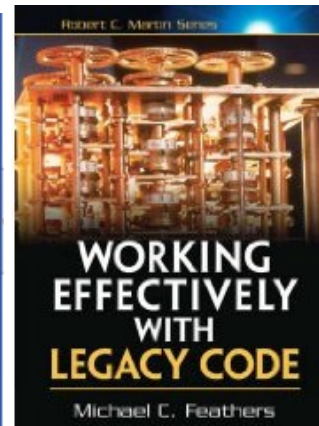
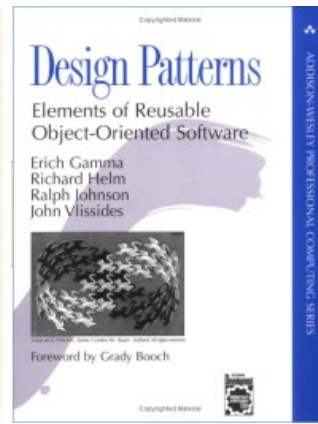
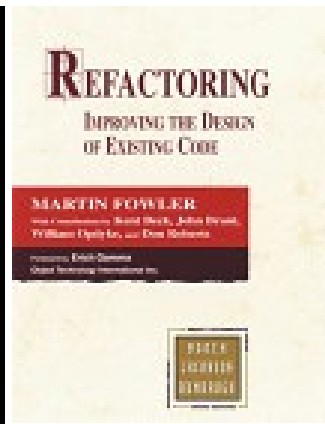
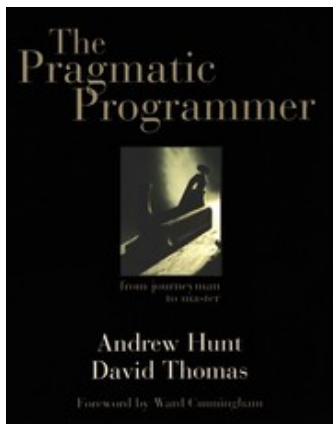
Möglichkeit 3: Ausprobieren und Üben

- Vielfältige Möglichkeiten
 - Projekt- und Studienarbeit, Thesis
 - Training on the Job
 - Werkstudent, Praktikum
 - Eigene Projekte, Open Source
- Übung ist sinnvoll und notwendig
- Allerdings alleine nicht effizient
- Markt wandelt sich zu schnell

Möglichkeit 4: Lesen

- Der durchschnittliche Software-Entwickler liest weniger als ein Fachbuch pro Jahr
- Dabei sind alle Fachbücher inbegriffen, auch die schlechten
 - „The Ultimate XYZ“
 - „XYZ in N Tagen“
- Es gibt zahllose gute Fachbücher
 - Alle Ebenen: Codierung bis Management

Lesen – (m)eine Top 9



Lesen light

- Für viele Themenbereiche gibt es Fachzeitschriften
 - Besonders an Bahnhofs- und Flughafen-Buchhandlungen
- Jede Programmiersprache
 - z.B. Java-Magazin, Java-Spektrum
- Übergeordnete Themen
 - Objekt-Spektrum, Der Entwickler, iX, c't
- Zeitschriften sind Fast-Food-Bücher

Lesen in Echtzeit

- Nachrichten-Webseiten für Entwickler
 - www.heise.de
 - www.devx.com
 - www.theserverside.com
- Aufsatzsammlungen von „Vordenkern“
 - <http://www.joelonsoftware.com/>
 - <http://martinfowler.com/articles.html>
- Blogs

Möglichkeit 5: Kommunizieren

- Das Internet sorgt für globalen Austausch
 - Entwickler-Foren
 - Chats (z.B. IRC)
 - Mail-Listen
- Diskussionsseiten
 - www.stackoverflow.com
 - www.heise.de
 - www.slashdot.org

Kommunizieren durch Publikation

- Blogs lesen und schreiben
 - Kommentare
 - Antworten auf eigenem Blog
- Aktuell bleiben
 - Aggregator-Seiten
 - www.javablogs.com
 - www.groovyblogs.org
 - Gibt es für fast jedes Thema
 - RSS-Reader

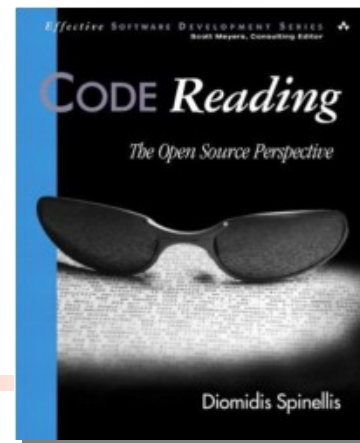
Kommunizieren in echt

- Konferenzen mit Diskussions-Anteil
 - Entwicklertage (Open Space)
 - Barcamps
 - Vortragsreihen und Workshops
- Lokale User Groups
 - z.B. JUG-KA, Java User Group Karlsruhe
- Technik-affine Vereine
 - Chaos Computer Club
 - Lokal: Entropia e.V.

Möglichkeit 6: Andere scheitern lassen

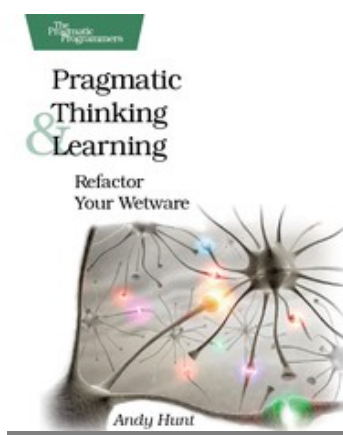
- Projektberichte lesen bzw. anhören
 - z.B. auf infoq.com
 - gamasutra.com -> Postmortems
- Sensationsberichte lesen
 - www.worsethanfailure.com
- Entwickler-Konferenzen besuchen
 - Dort die „uncoolen“ Vorträge anhören

Andere scheitern lassen



- Code anderer Entwickler lesen
 - Schwachstellen finden
- Open Source Code verbessern
 - Verbesserungen einbringen
 - Schwachstellen der Verbesserung finden lassen
- Kommerzielle Produkte untersuchen
 - Programmierfehler
 - Interaktionsdesign-Missgriffe
 - Bekannte Komponenten

Möglichkeit 7: Lernen lernen



- Arbeit ist im Zustand des „Flow“ am effektivsten
 - Ähnlich ist es mit Lernen
- Lern-Beratung an jeder (Hoch-)Schule
 - Was ist im Berufsalltag?
- Wissen um eigene Stärken und Schwächen
 - Wissen um eigenen Fähigkeiten-Stand
- Wissen um wirksame Lerntechniken

Berufliche Weiterbildung

- Meistens durch kostenintensive, aber zeitsparende „Crash-Kurse“
 - Sheep-Dip Training
 - Dort gelerntes muss jahrelang „reichen“
- Bücher werden angeschafft
 - Keine Zeit zum Lesen eingeplant
 - Ein Buch für ein Zehn-Personen-Team
- Firmen-interne Seminare
 - Lokale Weitergabe von Wissen

Lernen: Das Dreyfus-Modell der Fertigkeits-Aneignung

- Je nach Ausprägung fünf bis sieben verschiedene „Stufen“
 - Jede Stufe hat eigene Bedürfnisse und Fähigkeiten
 - Übergänge sind fließend (aber deutlich!)
- Nicht global zu sehen
 - Für jedes einzelne Thema ist jeder auf anderer Stufe
 - Ein sehr guter C++-Programmierer ist nicht automatisch auch sehr gut in Java

Fünf Stufen nach Dreyfus

- Anfänger („Novice“)
 - Folgt festen Regeln, braucht diese
 - Bereits geringe Abweichungen verunsichern
 - Benötigt Feedback, um besser zu werden
 - Kann sich meistens nicht selbst beobachten
- Fortgeschrittener („Advanced Beginner“)
 - Adaptiert die Regeln in Details
 - Entscheidet sich für passendere Regel
 - Lernt aus diesen Entscheidungen

Fünf Stufen nach Dreyfus

- Sachkundiger („Competent“)
 - Weiß um sein Unwissen
 - Fängt an, Prinzipien und Muster zu finden
 - Beachtet den Kontext mehr als die Regel
 - Fällt Entscheidungen und übernimmt Verantwortung für das Ergebnis

Fünf Stufen nach Dreyfus

- Erfahrener („Proficient“)
 - Arbeitet mit Erfahrung, Instinkt und Intuition
 - „Berechnung und rationale Analyse beginnt unwichtig zu werden“
 - Weiß oft in neuen Situation, was zu tun ist
- Experte („Expert“)
 - Löst Probleme ohne bewußte Entscheidung
 - Kann Lösungsschritte nicht mehr erklären
 - Daumenregel: 10-15 Jahre Erfahrung in einem Gebiet notwendig

Einschätzung unseres Berufsstandes

- Selbsteinschätzung ist sehr schwierig
 - niedrige Level „überschätzen“ sich häufig
 - hohe Level „unterschätzen“ sich häufig
- Untersuchung:
 - Durchschnitt: Knapp über Level 2
 - Das war der „Advanced Beginner“
- Viele Entwickler erreichen Level 3 nicht
 - Selbständige, verantwortliche Entscheidungen

Möglichkeit 8: Cheaten

- „Aufhübschen“ des Lebenslaufs
 - Ist nur auf kurze Sicht eventuell erfolgreich
 - Frage: Wer möchte von „aufgehübschtem“ Arzt operiert werden?
- Aneignen fremder Leistungen
 - Siehe „Aufhübschen“ und addiere noch die geschädigte Person mit auf
- Fähigkeits-„erweiternde“ Substanzen
 - Sind meistens illegal und/oder ungesund

Möglichkeit 9: Fokussieren

- Genaue Marktbeobachtung
 - Eigene Interessen wiederfinden
- Intensives, ausschließliches Studium des gewählten Themas
 - Bündelung der Fähigkeiten
 - Spezialisierung
- Hat schwerwiegende Nachteile:
 - Marktbewegung
 - Fehlende Anregung aus anderen Gebieten

FAIL

