

## ESE Kongress 2015

### Vortragsskript:

# Mist – Schon wieder zu spät! Aufwandsabschätzung von Projekten

Frank Listing, MicroConsult GmbH

**Oft ist am Ende der Zeit noch so viel Projekt übrig. Die Abschätzung des Software-Entwicklungsaufwands ist in vielen Projekten noch ein Mysterium. Die einen schätzen zu viel, die anderen zu wenig.**

Heutzutage ist schon fast Standard, dass Projekte nicht zum Termin fertig werden. Das betrifft nicht nur Softwareprojekte (siehe Flughafen BER). Allerdings scheint es bei der Zeitabschätzung von Softwareprojekten besonders viele Probleme zu geben. Die Schätzung über den Daumen des Entwicklers landet in den meisten Fällen doch auf einem Datum, dass am Ende (weit) überschritten wird.

Die Erfahrung zeigt, dass Entwickler gerne zu optimistisch schätzen; der Aufwand für Test und Fehlerbehebung wird dabei vergessen. Das liegt unter anderem an der Funktionsweise unseres Gehirns. Besonders bei langen und schwierigen Aufgaben tendieren Menschen dazu, die Zeit, die sie dafür benötigen, zu unterschätzen. Unser Kopf versucht automatisch, abstrakte Dinge zu vereinfachen. Je kleiner die zu lösende Aufgabe, desto weniger unterliegt man dieser Selbsttäuschung.

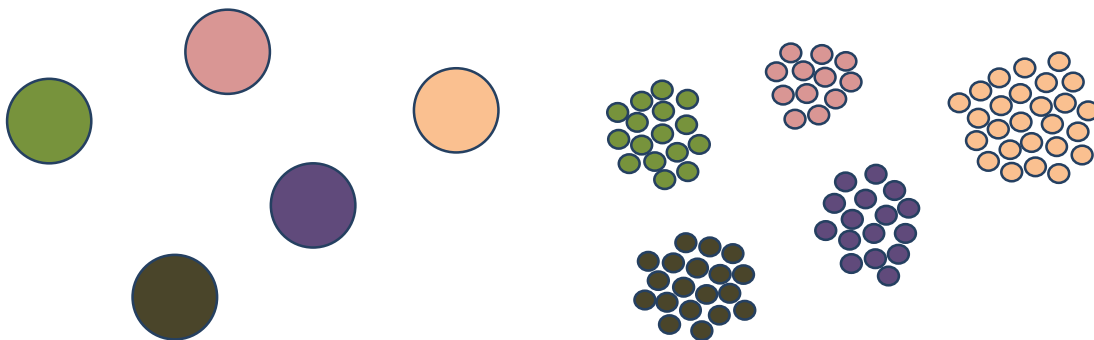


Abb. 1: Sinnestäuschung

Ein Beispiel, wie unser Gehirn uns täuscht, zeigt Abbildung 1: Schaut man flüchtig auf die Bilder, sieht man jeweils fünf Elemente. Nur bei genauerem Hinsehen können wir erkennen, dass auf der rechten Seite fünf Gruppen von kleinen Elementen zu sehen sind. Dass dort in der Realität 89 Elemente zu sehen sind, lässt sich nur sehr mühsam ermitteln.

Es gibt aber Spezialisten, die der Meinung sind, dass die Fehlerrate bei der Zeitabschätzung sehr genau vorhersagbar ist. Damit kann eine Formel zur Korrektur abgeleitet werden. Anders Abel (Autor des Blogs „Passion for Coding“ [1]) sagt: Man nehme die Zeit, die ein Entwickler abschätzt, multipliziere sie mit der Zahl Pi und transformiere das Ergebnis in die nächsthöhere Zeiteinheit. Das heißt, wenn ein Entwickler einen Tag schätzt, muss von einer realen Dauer von 3,14 Wochen ausgegangen werden ( $1\text{Tag} * 3,14 = 3,14\text{Tage} \rightarrow$  ergibt mit der nächsthöheren Einheit 3,14 Wochen).

Abel meint aber auch, dass richtiges Schätzen erlernbar ist. Es gibt eine bestimmte Zeitspanne, die ein Entwickler realistisch abschätzen kann. Das bewegt sich bei erfahrenen Entwicklern etwa von einer halben Stunde bis hin zu 25 Stunden. Um das Schätzvermögen zu trainieren, schätzt man vor jeder Aufgabe, wie lange man für sie braucht. Nach Abschluss der Aufgabe wird notiert, wie viel Zeit tatsächlich in Anspruch genommen wurde. So lernt man mit der Zeit, realistisch zu schätzen.

## Typische Fehler bei der Aufwandsabschätzung

Steve McConnell hat in seiner Präsentation „10 Deadly Sins of Software Estimation“ [2] typische Fehlerquellen aufgezählt. Einige davon sollen hier erwähnt werden:

- Die Zeitabschätzung wird gemacht, bevor die Anforderungen bekannt sind.
- Es wird angenommen, dass die besten Abschätzungen von den Leuten mit der lautesten Stimme kommen.
- Die Abschätzung wird anhand eines vergleichbaren Projektes gemacht, welches seinen Zeitplan überschritten hat. Die Abschätzung orientiert sich aber nicht an der benötigten Zeit, sondern an der ursprünglichen Abschätzung.
- Es wird angenommen, dass der Vertrieb den Aufwand wesentlich besser schätzen kann als die Entwicklung.
- Abschätzungen werden unter der Annahme gemacht, dass keiner ...
  - ein Training besucht
  - an Meetings teilnimmt
  - kurz einem anderen Projekt helfen muss
  - einen wichtigen Kunden unterstützt
  - Urlaub nimmt
  - krank wird
  - ...
- Es wird davon ausgegangen, dass spezielle Software für die Abschätzung bei weitem nicht die Rechenpower hat wie ein Kugelschreiber und ein Bierdeckel.

## Was kann verbessert werden?

- Klar unterscheiden zwischen Zielen (der Wunsch, wann das Projekt fertig sein soll) und Abschätzungen (wie lange brauchen wir wirklich?).
- Keine Abschätzung von außen (z B Marketing) aufdrängen lassen.
- Nicht zu früh abschätzen. Erst wenn die meisten Anforderungen und Randbedingungen bekannt sind, ist eine sinnvolle Abschätzung möglich.
- Keine Wunder von neuen Tools und Technologien erwarten. Im Gegenteil, sie kosten im Allgemeinen erst einmal mehr Aufwand.
- Verschiedene Methoden für die Abschätzung nutzen, um Fehler zu minimieren.
- Spezielle Software für die Aufwandsabschätzung einsetzen.
- Risiken in die Aufwandsabschätzung mit einbeziehen
- Die Abschätzung von großen Projekten in kleinere, überschaubarere Einheiten zerlegen, um die Genauigkeit zu erhöhen.

## Agile Methoden

Innerhalb agiler Prozesse ist es wesentlich einfacher, den Vorgang der Zeitabschätzung zu üben. Durch die regelmäßige Sprintplanung und Retrospektive wird der Prozess der Abschätzung und Auswertung der real benötigten Zeit innerhalb kurzer Zeiträume regelmäßig wiederholt. Dadurch werden Fehler, die in der Abschätzung gemacht werden, schnell erkannt, und der Lerneffekt setzt schneller ein.

## Erschwerende Faktoren

Erschwert wird eine realistische Abschätzung durch unternehmenspolitische Randbedingungen. Oft wird eine reale Zeitabschätzung vom Management nicht Akzeptiert. Termine sind das Ergebnis von Wunschenken. Auch die (unrealistischen) Erwartungen des Kunden können zu falschen Abschätzungen führen. Es ist erst einmal wichtig, den Auftrag zu bekommen.

Ein weitere Punkt ist das Verwechseln von Manntagen und Zeittagen (ein Manntag, ca. 0,5 Zeittage). Kein Mitarbeiter kann zu 100% an seinem Projekt arbeiten. Es gibt immer andere Arbeit, die auch erledigt werden muss (z.B. Besprechungen, Pflege älterer Projekte, ...).

### **Aufwand abschätzen mit Methode – Expertenschätzung**

Bei der Expertenschätzung schätzen mehrere Experten den Aufwand unabhängig voneinander ab. Basis sind persönliche Erfahrungen und Daten aus bereits abgeschlossenen Projekten. Das Ergebnis wird gemittelt. Dadurch werden bessere Ergebnisse als nur bei einer Schätzung erzielt. Allerdings hängt dieses Verfahren sehr von der Erfahrung der Experten ab. Probleme entstehen vor allem, wenn versucht wird, Erfahrungen aus einem völlig andersartigen Projekt auf das aktuelle Projekt zu übertragen.

### **Aufwand abschätzen mit Methode – Delphi-Methode**

Bei der Delphi-Methode schätzen mehrere Experten den Aufwand unabhängig voneinander ab. Die Abschätzung wird begründet. Jeder Schätzer erhält alle Ergebnisse der Schätzung und schätzt auf Basis der Zeiten und Begründungen der Kollegen erneut ab. Dabei sind Abweichungen vom Mittelwert der vorherigen Schätzung zu begründen. Die Schätzungen werden solange wiederholt, bis sie nahe beieinander liegen. Die Delphi-Methode ist genauer als die Experten-Schätzung, kostet aber wesentlich mehr Aufwand.

### **Aufwand abschätzen mit Methode – Function-Point-Methode**

Die Function-Point-Methode wurde von Alan Albrecht (IBM) Ende der 70er-Jahre entwickelt[3] und ist heute als ISO-Norm standardisiert (ISO/IEC 20926:2009 [2009])[4].

Grundidee: Ein Vorhaben wird in verschiedene Grundfunktionen zerlegt, z.B. Eingaben, Ausgaben, Abfragen, Schnittstellen, logische Datenbestände. Dann wird für jede Grundfunktion Art, Umfang (z.B. LOC) und Komplexität (z.B. leicht, mittel, schwer) festgelegt. Abhängig von einer firmenspezifischen Function-Point-Funktion  $f(\text{Art}, \text{Umfang}, \text{Komplexität})$  werden jeder Grundfunktion so genannte Function Points zugeordnet. Aus der Summe der Function Points lässt sich der Gesamtaufwand abschätzen. Voraussetzung ist die Kenntnis über die Produktivität (resultiert in der Function-Point-Funktion) Diese Informationen müssen in Vorgängerprojekten gesammelt werden.

### **Aufwand abschätzen mit Methode – COCOMO**

COCOMO (Constructive Cost Model) wurde 1981 von Barry W Boehm (Softwareingenieur bei Boeing) entwickelt[5].

Ausgangspunkt ist die Schätzung der Produktgröße in KDSI (Kilo lines of delivered source instructions). Es werden nur die ausgelieferten source instructions betrachtet! Aus diesem Grundwert werden der Entwicklungsaufwand und die Projektdauer berechnet. Es gibt Formeln für

- einfache Anwendungen,
- Programmsysteme,
- Embedded-Systeme

Die erhaltenen Werte können durch die Multiplikation mit Kostenfaktoren (z.B. Produktkomplexität, Erfahrung der Entwickler, ...) verfeinert werden. COCOMO wurde mittlerweile von COCOMO II abgelöst[6] und beinhaltet Anpassungen an aktuelle Software-Entwicklungsmodelle.

### **Fazit**

Die Qualität der Schätzverfahren hängt sehr stark von den vorhandenen Daten anderer Projekte (derselben Firma) ab. Die Erfahrung der am Schätzprozess beteiligten Experten steigt mit jedem abgeschlossenen Projekt. Algorithmische Verfahren wie COCOMO oder Function Point können eine Aufwandsabschätzung deutlich verbessern, sind aber nur sinnvoll anwendbar, wenn sie für die eigene Firma kalibriert wurden.



Embedded Software Engineering Kongress

## Literatur- und Quellenverzeichnis

- [1] <https://coding.abel.nu/2012/06/programmer-time-translation-table/>
- [2] [http://www.construx.com/Resources/Presentation/10\\_Deadly\\_Sins\\_of\\_Software\\_Estimation/](http://www.construx.com/Resources/Presentation/10_Deadly_Sins_of_Software_Estimation/)
- [3] A. J. Albrecht [1979]: Measuring application development productivity, Proceedings of the IBM Applications Development Symposium, Seiten 83-92
- [4] ISO/IEC 20926:2009 [2009]: Software and systems engineering – Software measurement – IFPUG functional size measurement method, International Organization for Standardization, Genf
- [5] Barry W Boehm: Software Engineering Economics, Englewood Cliffs, NJ, Prentice-Hall, 1981 ISBN 0-13-822122-7
- [6] Barry Boehm, et al Software cost estimation with COCOMO II (with CD-ROM) Englewood Cliffs, NJ:Prentice-Hall, 2000, ISBN 0-13-026692-2

## Autor

Dipl.-Ing. Frank Listing ist seit 2002 Trainer und Projektcoach bei der MicroConsult GmbH mit dem Schwerpunkt Microsoft-Plattformen, objektorientierte Programmierung und Testen von Embedded-Systemen und u.a. fachlich für das Thema NET verantwortlich. Sein Wissen gibt er immer wieder auch in Publikationen und Fachvorträgen weiter.

## Kontakt

Internet: [www.microconsult.de](http://www.microconsult.de)  
E-Mail: [f.listing@microconsult.de](mailto:f.listing@microconsult.de)



**MicroConsult - Ihr Partner für Embedded Systems Engineering :  
professionelle Beratung, Projektunterstützung und Schulungen.**