

# Aufwandsschätzung in IT-Großprojekten

## Function Point Methode

Florian Weigand

**Abstract:** Im Folgenden wird beschrieben, wie man den Aufwand eines IT-Großprojektes bestimmen kann. Zuerst werden die Faktoren behandelt, die die Aufwände eines Softwareprojektes beeinflussen. Ausgehend von diesen Einflussgrößen erfolgt das Erschließen von Methoden-Bausteinen, auf deren Basis eine grobe Schätzung möglich ist. Im letzten Kapitel wird die Function Point Methode vorgestellt. Diese kombiniert zwei der Methoden-Bausteine.

## 1 Einleitung

Eine genaue Aufwandsschätzung in einem IT-Projekt ist von großer Wichtigkeit, denn eine Aussage über die zu erwartenden Kosten wird schon im Vorfeld des geplanten Projektes benötigt. Dies ist im Falle von Standardsoftware wichtig, um einen Preis der Software festlegen zu können, weil der Preis von den im Unternehmen entstandenen Kosten abhängig ist. Bei Individualsoftware kann auf Grundlage einer Aufwandsschätzung ein Kostenvoranschlag abgegeben werden.

Laut [WM07] sind die größten Kostenfaktoren eines Softwareprojektes die Aufwände für den Lizenzeinkauf von Entwicklungssoftware (oder für das Projekt zugekaufte Softwarekomponenten) und die Kosten für den Personaleinsatz. 'Die Kosten für andere Dienstleistungen, Büromaterial, Druckkosten, [...], Reisekosten etc. sind im Verhältnis zu den Personalkosten bedeutungslos.'<sup>1</sup>

Um den Aufwand eines IT-Projektes zu schätzen, werden unterschiedliche Methoden herangezogen. Diese werden in Kapitel 4 näher beschrieben. Am Ende vieler Verfahren, wie zum Beispiel der Function Point Methode, auf die im Kapitel 5 eingegangen wird, wurde eine genau wirkende Dezimalzahl berechnet, die den geschätzten Aufwand des Projektes angeben soll. Es ist anzumerken, dass es sich bei dem Prozess der Aufwandsschätzung, wie der Name schon sagt, um eine Schätzung handelt und dass das Ergebnis der Aufwandsschätzung von erfahrenen Projektleitern in Relation zu bereits abgeschlossenen Projekten gesetzt werden sollte.

Neben den Methoden, die zur Messung des Aufwandes herangezogen werden, gibt es Faktoren, die den Aufwand eines Softwareprojektes maßgeblich beeinflussen. Das Ziel eines Verfahrens zur möglichst genauen Abschätzung muss auch sein, diese Einflussfaktoren mit einzubeziehen. Diese Faktoren werden im Kapitel 3 behandelt.

---

<sup>1</sup>Wieczorrek, H. W., & Mertens, P. (2007). Management von IT-Projekten. Von der Planung zur Realisierung (2nd.). Berlin, Germany Springer.

## 2 Aufwandsschätzung in IT-Großprojekten

In IT-Großprojekten spielt die Aufwandsschätzung eine wichtige Rolle. So hat sich in der Vergangenheit gezeigt, dass eine falsche oder nicht getätigte Aufwandsschätzung zu erheblichen Mehrkosten führen kann.

### 2.1 Warum betreibt man Aufwandsschätzung?

Um diese Frage zu beantworten, ist es hilfreich, gescheiterte Softwareprojekte herauszugreifen und diese zu analysieren. Dabei zeigt sich, dass Software die mehr Mann-Monate (MM) Arbeit benötigt als geplant oder erheblich später fertiggestellt wird, hohe Kosten verursacht. Klar ist, dass personeller Mehraufwand höhere Kosten verursacht.

Warum aber auch eine spätere Fertigstellung der Software mit hohen Kosten verbunden ist möchte ich am Beispiel der InPol-Neu Software darlegen. InPol-Neu, das 'Neue Polizeiliche Informationssystem', ist eine Software, die den Datenaustausch zwischen der Polizei in den einzelnen Bundesländern erleichtern und eine Software aus den 70er Jahren ablösen soll. Es war geplant, dass InPol-Neu im Jahr 2001 fertig gestellt wird. Auf Grund einer Fehlplanung konnte das Softwareprojekt erst 2003, also mit zwei Jahren Verspätung an den Start gebracht werden. Die Mehrkosten beziffern sich auf 50 Millionen Euro. <sup>2</sup>

### 2.2 Aufwandsschätzung ist ein kontinuierlicher Prozess

Je weniger Informationen man über das zu entwickelnde Softwaresystem hat, desto ungenauer ist eine Aufwandsschätzung. Es gibt viele Unbekannte, die schlecht abgeschätzt werden können, oder man hat eine falsche Vorstellung einer Lastenheftfunktion. All dies sind wichtige Erkenntnisse in der Aufwandsschätzung. Dabei gilt nach [WM07]: 'Eine Schätzung ist um so ungenauer, je früher sie im Verlauf des Softwareprojektes eingesetzt wird.'

Aber wie in Kapitel 1 gesehen, ist es wichtig, eine möglichst frühe Schätzung der Kosten zu haben, um beispielsweise einen Kostenvoranschlag machen zu können. Nun widersprechen sich beide Punkte. Eine Aufwandsschätzung ist im frühen Stadium des Projektverlaufes ungenau aber nötig, im späteren Verlauf aber genauer.

Das Verfahren wird nicht zu einem bestimmten Zeitpunkt eingesetzt, sondern kontinuierlich. Wichtig ist es zu erkennen, dass die Aufwandsschätzung ein iterativer Prozess ist, der um so genauer ist, je weiter der Projektablauf voranschreitet.

---

<sup>2</sup>Ministerium des Innern und für Sport des Landes Rheinland-Pfalz, Peter Henrich, LKA

### **3 Einflussfaktoren auf die Aufwandsschätzung**

Nicht nur die zu entwickelnde Software selbst hat Einfluss auf den Gesamtaufwand eines Projektes, sondern auch die unternehmensinternen Abläufe beeinflussen den Gesamtaufwand mehr oder weniger stark. Wenn ein ähnliches Softwareprojekt im Unternehmen bereits abgeschlossen wurde, kann man auf Grund von geänderten Einflussfaktoren wie zum Beispiel mehr Erfahrung der Mitarbeiter, auf einen geänderten Aufwand im nächsten Projekt schließen.

Einflussfaktoren lassen sich dabei in zwei große Gruppen einteilen: Einerseits die 'ergebnisbezogenen Einflussfaktoren', also Einflussgrößen, die sich direkt durch das Lastenheft beziehungsweise das ausgearbeitete Pflichtenheft ableiten lassen.

Zum anderen sind es 'abwicklungsbezogene Einflussfaktoren', also Einflussgrößen, bei denen nicht das Ergebnis als Einfluss gewertet wird, sondern äußere Einflüsse wie die Zeit, die zur Verfügung steht oder unternehmensinterne Abläufe wie zum Beispiel die eingesetzten Entwicklungsumgebungen.

Um dies zu verdeutlichen, möchte ich den Einflussfaktor 'Erfahrung' zuordnen: Erfahrung der Entwickler ist unabhängig von den Ergebnissen, die im Softwareprojekt erzielt werden sollen. Somit ist 'Erfahrung' ein abwicklungsbezogener Einflussfaktor.

#### **3.1 Ergebnisbezogene Einflussfaktoren**

Faktoren, die sich auf Grund von Festlegung von Ergebniszielen ändern, werden als ergebnisbezogen bezeichnet. Sie lassen sich am besten aus Lasten- und Pflichtenheft des Auftraggebers, beziehungsweise des Auftragnehmers ermitteln. Dabei kann man folgende Größen ablesen.

##### **3.1.1 Quantität**

Unter die Quantität fallen alle mengenmäßig beschreibbaren Anforderungen an das Softwareprojekt. Neben der bereits erwähnten Anzahl an Funktionen, die umgesetzt werden müssen, ist auch der Datenumfang von entscheidender Rolle. Von Vorteil ist hier, dass der Funktionen- und Datenumfang unabhängig von der Programmiersprache gemessen werden kann.

Eine weitere Möglichkeit wäre, den quantitativen Aufwand in 'Lines of Code' (LOC) zu messen. Dabei ist es schwierig den Begriff der Line, also der Zeile, zu definieren. Es ist fraglich, ob beispielsweise ein Kommentar mitgezählt werden darf oder ob ein Programmierer, der gut strukturiert mit Leerzeilen arbeitet mehr Lines programmiert. Insbesondere, wie oben angedeutet, ist eine Unabhängigkeit von der Programmiersprache nicht mehr gewährleistet.

### **3.1.2 Qualität**

Qualität ist ein subjektiver Begriff. Es gibt nicht die Qualität, sondern viele verschiedene Arten von Qualität. Darunter fällt zum Beispiel die möglichst geringe Wartungsanfälligkeit oder eine leichte modulare Erweiterbarkeit.

Qualität an sich lässt sich nicht messen. Man kann aber Metriken heranziehen, die bestimmte, vorher wohl definierte, Qualitätsanforderungen überprüfen können.

Je höhere Anforderungen an die Qualität gestellt werden, desto höher ist der Aufwand.

### **3.1.3 Komplexität**

Die Komplexität eines Softwareprojektes wird von dem verwendeten Hardwaresystem oder der verwendeten Programmiersprache, mit der entwickelt wird, beeinflusst. Dabei kann man ähnlich wie bei der Definition der Softwarequalität Metriken festlegen, die auch die Komplexität des zu entwickelnden Programms bewerten.

## **3.2 Abwicklungsbezogene Einflussfaktoren**

Einflussfaktoren, die sich auf Grund von Änderungen interner Abläufe wie z.B. der verwendeten Entwicklungsumgebung und äußeren Faktoren wie z.B. der zur Verfügung stehenden Zeit ändern, nennt man abwicklungsbezogen.

### **3.2.1 Erfahrung**

Wenn ein Entwicklerteam ein ähnliches Projekt wie das anstehende bereits durchgeführt hat, kann man auf Grund der Erfahrungen, die das Team in dem alten Projekt gesammelt hat, davon ausgehen, dass sich diese Erfahrung positiv auf das Ausführen des nächsten Projektes auswirken wird. Außerdem hilft Erfahrung zu einem besseren Einsatz der unternehmensspezifischen Entwicklungstools.

Erfahrung wirkt sich also positiv auf dem Aufwand aus.

### **3.2.2 Entwicklungstools**

Entwicklungsumgebungen stellen einen weiteren Einflussfaktor auf den Aufwand eines Projektes dar. Umfangreiche Bibliotheken helfen, doppeltes Programmieren von bereits früher implementierten Methoden zu vermeiden. Setzt ein Unternehmen aber eigens entwickelte Entwicklungstools ein, müssen neue Mitarbeiter erst geschult werden und brauchen Zeit, bis sie am Softwareprojekt mitarbeiten können.

### 3.2.3 Programmiersprache

Auch die eingesetzte Programmiersprache beeinflusst den Aufwand. Man denke an eine hardwarenahe Programmierung in Assembler im Vergleich dazu an eine objektorientierte Umsetzung in Java.

## 4 Methoden der Aufwandsschätzung

Um den Aufwand eines Softwareprojektes schätzen zu können, gibt es grundlegende Methoden, mit der eine Aufwandsschätzung betrieben werden kann. Diese Methoden sind alleine genommen nicht ausreichend, um einen aussagekräftigen Schätzwert zu bestimmen. Man kann sie aber als eine Art Baukasten betrachten. Erst eine Kombination aus mehreren Methoden kann wie zum Beispiel in der Function Point Methode als aussagekräftiger Indikator herangezogen werden.

### 4.1 Vergleichsmethoden

Methoden, die auf direktem Vergleichen mit Aufwänden bisheriger Softwareprojekte basieren, nennt man Vergleichsmethoden. Man sucht ein Projekt, welches sich in den Einflussfaktoren mit denen des jetzigen Projektes ähnelt, und schließt so auf die Gesamtkosten des zu realisierenden Softwareprojektes. Wie man dabei vorgehen kann, möchte ich nun an zwei Methoden, die auf Vergleichen basieren, darstellen.

#### 4.1.1 Analogiemethode

In der Analogiemethode sucht man sich ein Softwareprojekt, das dem zu entwickelnden ähnlich ist. Stimmen Funktionen fast überein, kann der Aufwand des bereits entwickelten Funktion übernommen werden, wobei man die Faustregel nach Gredy heranziehen kann: 'Software-Entwicklungen, die Code wieder verwenden, benötigen nur ca.  $\frac{1}{4}$  der Zeit und der Ressourcen'<sup>3</sup>. Segmente, die völlig überarbeitet werden müssen, können mit dem gleichen Zeitaufwand, der für die erste Implementierung benötigt wurde, eingerechnet werden. Neue Softwarekomponenten, die eine höhere Komplexität zeigen, werden mit einem Komplexitätszuschlag von 1,5 mit in die Aufwandsschätzung eingerechnet.

#### 4.1.2 Relationenmethode

Die Relationenmethode ist im Gegensatz zur Analogiemethode formalisiert. Während bei der Analogiemethode Faustregeln und Richtwert wie der Komplexitätszuschlag mit 1,5

---

<sup>3</sup>'Faustregel' nach Grady R.B., Practical Software Metrics for Management and Process Improvement, 1992

herangezogen wurden, wird bei der Relationenmethode auf eine Faktorenliste zurück gegriffen. Nach [BA00] gilt:

Programmiersprache	Programmiererfahrung	Dateiorganisation
PL/1 = 100 COBOL = 120 Assembler = 140	5 Jahre = 80 3 Jahre = 100 1 Jahr = 140	sequentiell = 80 indexsequentiell = 120

Diese Faktorenliste stammt aus dem Jahre 1978. Sie ist deshalb nicht mehr aktuell. Die zur Zeit verwendeten Faktorenlisten sind käuflich zu erwerben und werden beispielsweise von Unternehmensberatungen im Bereich der Softwareentwicklung erstellt, da diese Unternehmen eine Übersicht über viele Softwareprojekte haben.

## 4.2 Algorithmische Methoden

Bei algorithmischen Methoden wird der Aufwand durch eine geschlossene Formel berechnet. Das hat den Vorteil, dass Ergebnisse besser vergleichbar sind als bei einer Aufwandsschätzung, die auch von einer Expertenmeinung beeinflusst werden kann. So basiert die Vergleichsmethode, die weiter oben vorgestellt wurde, nur auf einer Abschätzung durch einen Experten und der Hochrechnung auf den erwarteten Aufwand.

### 4.2.1 Multiplikatorenmethode

Die Multiplikatorenmethode ordnet jeder Softwarekomponente einen Aufwandsfaktor, also einen Multiplikator zu. Mit der über die Softwarekomponente geschätzten Anzahl von LOC kommt man durch Multiplikation von ungewichteten LOC, mit dem Aufwandsfaktor multipliziert, zu gewichteten LOC. Mit den gewichteten LOC kann eine genauere Aussage über den erwarteten Aufwand getroffen werden.

*In folgendem Beispiel nach [BA00] sieht man wie eine Multiplikatorenmethode eingesetzt wird.*

Kategorie	Teilprodukte	Summe LOC	Faktor	LOC bewertet
Steuerprogramm	1 x 500	500	1,8	900
E/A-Programme	1 x 700 + 2 x 500	1700	1,5	2550
Datenverwaltung	1 x 800 + 2 x 250	1300	1,0	1300
Algorithmen	1 x 300 + 5 x 100	800	2,0	1600

Dabei werden Teilprodukte mit einem Faktor multipliziert. Dieser basiert auf dem tatsächlichen Aufwand, den ein Teilprodukt in Vergangenheit verursacht hat. So ist in diesem Beispiel der Aufwandsfaktor für die Datenverwaltung mit 1,0 angegeben. Bei Algorithmen wurde der Faktor 2,0 angegeben. Die Zeit die für die Implementierung eines Algorithmus benötigt wird, wird also wesentlich höher eingeschätzt, als die Dauer für die Implementierung einer Datenverwaltung.

#### **4.2.2 Gewichtungsmethode**

In der Gewichtungsmethode wird ein berechneter Gesamtaufwand in einen *bewerteten* Gesamtaufwand überführt. Im Gegensatz zur Multiplikatorenmethode, die von den einzelnen Softwarekomponenten ausgeht, betrachtet man hier den Gesamtaufwand. Dabei findet die Umrechnung über eine geschlossene Formel statt. Diese Formel berücksichtigt die Einflussfaktoren. Größen die herangezogen werden, können entweder abwicklungsbezogen oder ergebnisbezogen sein. Dadurch können unternehmensspezifische Eigenschaften, wie das Personal gewertet und in den Gesamtaufwand mit einbezogen werden. So ist eine genauere Abschätzung möglich. Ein Beispiel, auch für die geschlossene Formel, findet sich im nächsten Kapitel. Die Function Point Methode verwendet die Gewichtungsmethode, um von den ungewichteten Function Points (FPs) zu gewichteten FPs zu kommen. Die gewichteten FPs sind dabei viel aussagekräftiger, da sie Einflussfaktoren mit in den Gesamtaufwand einbeziehen.

### **5 Function Point Methode**

Die Function Point Methode kombiniert zwei der vorgestellten Methoden:

- Relationenmethode
- Gewichtungsmethode

#### **5.1 Historischer Hintergrund**

Die Function Point Methode (FPM) wurde im Jahre 1979 von Allen J. Albrecht entwickelt. Als Elektroingenieur mit dem Schwerpunkt Elektronik, arbeitet er bei IBM. Dabei war er mit der Entwicklung der ersten Mikrocomputer beschäftigt und arbeitet zwischen 1955 und 1985 im Bereich des Software-Managements. Die in diesen Bereichen gesammelten Erfahrungen flossen in die Entwicklung der FPM mit ein.

#### **5.2 Vorgehen**

Bei der FPM ermittelt man den erwarteten Projektaufwand anhand des Lasten- und später des Pflichtenheftes.

##### **5.2.1 Erster Schritt**

Dabei kategorisiert man zu erst die Produkthanforderung. Jede Anforderung wird dabei einer von fünf Anforderungskategorien zugeordnet:

- Eingabedaten
- Abfragen
- Ausgabedaten
- Datenbestände
- Referenzdaten

### 5.2.2 Zweiter Schritt

Ist diese Einteilung vorgenommen, werden die Anforderungen unter Zuhilfenahme von Richtlinien in Komplexitätsklassen eingeordnet:

- einfach
- mittel
- komplex

*Da wir in unserem Beispiel nur das Lastenheft betrachten, werten wir alle Produkthanforderungen als komplex, da eine im Lastenheft definierte Funktion auf jeden Fall implementiert werden muss und sie in der Regel weiter ausgebaut und entwickelt wird.*

### 5.2.3 Dritter Schritt

Die ermittelten Werte werden in das Berechnungsformular eingetragen und durch Multiplikation mit der Gewichtung wird die Anzahl der ungewichteten Function Points errechnet. Nun sind alle zu implementierenden Funktionen, Datenbestände etc. notiert und mit einem Faktor gewichtet. Aktuelle Faktorenlisten kann man bei der 'International Function Point Users Group' (IFPUG) <sup>4</sup> erwerben.

### 5.2.4 Vierter Schritt

Nun folgt die Bewertung der Einflussfaktoren. Die in Kapitel 3 behandelten Einflussfaktoren werden zur Hand genommen und fließen in die FPM ein. Dabei werden die aktuell verwendeten Einflussfaktoren von der IFPUG zur Verfügung gestellt und laufend ergänzt. Allen Einflussfaktoren wird hierbei ein Punktwert zugeordnet, je nach ihrer Wichtigkeit. Insgesamt können 60 Punkte vergeben werden.

---

<sup>4</sup>[http:// www.ifpug.org](http://www.ifpug.org)



### 5.2.5 Fünfter Schritt

Im fünften Schritt werden aus den im Schritt drei berechneten ungewichteten Function Points durch folgende Formel:

$$FP_{bew.} = FP_{unbew.} \times \left( \frac{\text{Einflussfaktor}}{100} + 0,7 \right)$$

gewichtete FP berechnet. Für die Einflussfaktoren können zwischen insgesamt 0 und 60 Wertungspunkten vergeben werden. Die maximale Abweichung beträgt somit  $\pm 30\%$ .

*Beispiel: wir kommen somit auf folgendes Ergebnis:*

$$FP_{bew.} = FP_{unbew.} \times \left( \frac{\text{Einflussfaktor}}{100} + 0,7 \right) = 244 \times \left( \frac{25,5}{100} + 0,7 \right) = 244 \times 0,955 \approx 233$$

### 5.2.6 Sechster Schritt: Ablesen des Aufwandes in der Aufwandstabelle

Den bewerteten FP ist ein Aufwand zugeordnet. Dies kann in der Tabelle nachgelesen werden. Dabei kann die original IBM Tabelle verwendet oder eine neuere Tabelle herangezogen werden. Diese ist über die IFPUG zu beziehen. Große Unternehmen und Unternehmensberater können ihre eigenen Aufwandstabellen anlegen (siehe dazu Schritt 7).

*Beispiel: wir lesen das Ergebnis aus der original IBM Tabelle ab.*

FP	IBM-MM	FP	IBM-MM
50	5	700	52
100	8	750	56
150	11	800	60
200	14	850	64
250	17	900	68
300	20	950	72
350	24	1000	76
400	28	1100	85
450	32	1200	94
500	36	1300	103
550	40	1400	112
600	44	1500	122
650	48	1600	132

Laut IBM Tabelle benötigt man zur Realisierung eines Projektes mit 233 Function Points 16 MM Personalaufwand. Mit der Formel für die optimale Projektdauer nach [BA00] gilt:

$$\text{optimale Entwicklungsdauer} = 2,5 \times (\text{Aufwand in MM})^s [\text{Monate}]$$

wobei  $s = 0,38$  für Stapel-Systeme,  $s = 0,35$  für Dialog-Systeme oder  $s = 0,32$  für Echtzeit-Systeme.

Für das Beispiel gilt:

$$\begin{aligned} \text{optimale Entwicklungsdauer} &= 2,5 \times (\text{Aufwand in MM})^s [\text{Monate}] = \\ &= 2,5 \times (16MM)^{0,35} \approx 6,6M \end{aligned}$$

und somit für Ideale Anzahl an Mitarbeitern:

$$16MM/6,6M = 2,42M \approx 3$$

### 5.2.7 Siebter Schritt

Wenn das Projekt fertig gestellt ist, wird die Aufwandstabelle aktualisiert und mit dem in der Function Point Berechnung ermittelten Ergebnis und dem tatsächlichen Aufwand in die Tabelle geschrieben. Dabei 'lernt' die FPM also mit jeder Analyse dazu, da nach jedem Projektabschluss der tatsächlich benötigte Personalaufwand in die Tabelle eingefügt wird.

## 6 Zusammenfassung

Die FPM wurde im Jahre 1979 entwickelt. Seit dieser Zeit hat sich die Informatik stark gewandelt. Die Function Point Methode ist auch deshalb nicht als Perfekt anzusehen, sondern eher als eine gute Möglichkeit zu einer Schätzung zu gelangen, die aber von einem erfahrenen Projektleiter in Relation gesetzt werden muss. Auch der genau wirkende berechnete Aufwand, der nach der Durchführung als Resultat zurück geliefert wird, ist kritisch zu betrachten. Wenn ungenaue Zahlen oder gar falsche Eingaben gemacht werden, kann das beste Verfahren zu keiner guten Aufwandsschätzung gelangen. Die Frage, die letztlich zu beantworten bleibt ist die, wie man in der FPM eine gute Einteilung in die einzelnen Kategorie zustande bringt. Diese Einteilung ist leider immer noch nicht einfach und ist auch in großen Unternehmen Experten mit viel Erfahrung vorbehalten. Doch nur so lässt sich die FPM gut einsetzen und ein Aufwand errechnen, der eine gute Grundlage zur Kalkulation darstellt.

## 7 Literaturverzeichnis

### Literatur

- [WM07] Wieczorrek, H. W., & Mertens, P.: *Management von IT-Projekten. Von der Planung zur Realisierung (2nd.)*, Springer Berlin, Germany (1996)
- [BA00] Balzert, H.: *Lehrbuch der Software-Technik, zweite Auflage. 83 - 96*, Spektrum Akademischer Verlag Heidelberg, Berlin (2000)