von uwe vigenschow und christian weiss

mehr zum thema:
www.oose.de

DAS ESSENZSCHRITT-VERFAHREN: AUFWANDSSCHÄTZUNGEN AUF DER BASIS VON USE-CASES

Wie kann man auf der Basis von Anwendungsfällen möglichst frühzeitig eine möglichst gute Aufwandsschätzung abgeben? Der Beitrag stellt das Essenzschrittverfahren vor, das dies prinzipiell leisten kann. Es wird erläutert, welche Voraussetzungen, Konventionen und Randbedingungen zu beachten sind und welche praktischen Erfahrungen hierzu vorliegen. Da das Essenzschrittverfahren Anwendungsfälle mit einer speziellen Granularität und Abstraktion benötigt, beschreibt dieser Beitrag nebenbei auch, welche Tücken und Probleme Anwendungsfälle in der Praxis bieten und wie diese erfolgreich vermieden werden können.

Die Problematik frühzeitiger Aufwandsabschätzungen

Auf Grundlage von Erfahrungswerten und über eine Metrik geeigneter Messgrößen versucht man seit den 70er Jahren die zu erwartenden Aufwände eines Software-Entwicklungsprojekts zu einem möglichst frühen Zeitpunkt mit möglichst hoher Wahrscheinlichkeit abzuschätzen. Eines der bekanntesten Verfahren ist *COCOMO*, das *Constructive Cost Model* von Barry Boehm (vgl. [Boe81]), das 1981 veröffentlicht wurde und mittlerweile in einer neuen überarbeiteten Form als COCOMO II vorliegt (vgl. [Boe00]).

An den COCOMO-Modellen lässt sich die grundsätzliche Problematik sehr gut darstellen. Jedes Modell ist an bestimmte Annahmen gebunden. Durch die Einführung von Komplexitätsparametern werden die resultierenden Formeln und Tabellen an die existierende Breite realer Projekte angepasst. Dem Kern der jeweiligen Aufwandsformeln liegen harte Messgrößen zu Grunde. Bei COCOMO ist die zentrale Messgröße die Anzahl der Codezeilen (*LOC – Lines of Code*), bei COCOMO II bilden *Object-Points* die Basis. Die aus statistischen Berechnungen abgeleiteten Tabellen in COCOMO beruhen auf COBOL- und Assembler-Programmen, für COCOMO II wurden im Wesentlichen objektorientierte Projekte berücksichtigt.

Bei der Weiterentwicklung von COCOMO zu COCOMO II erfolgte eine Verschiebung von der Messgröße LOC, die erst der Codierung entspringt, hin zu *Object-Points*, wie Masken, Reports und 3GL-Sprachmodulen, die innerhalb der Anwendung programmiert werden sollen bzw. wurden. Letztere sind bereits – zumindest teilweise – aus Analyseergebnissen ableitbar. Damit wird das Problem angegangen, dass häufig bereits vor dem Beginn der Codierung Aussagen zu den Kosten für ein Projekt benötigt werden.

Das Problem entzerrt sich ein wenig bei agilen Vorgehensweisen wie dem *eXtreme Programming (XP)*, bei denen bereits sehr früh mit dem Codieren begonnen wird, verschärft sich aber deutlich im Umfeld modellgetriebener Architekturen (MDA), wo ein Großteil des Codes aus Analyse- und Designergebnissen generiert wird. Generell versuchen alle *Function-Point*-basierten Verfahren (vgl. Kasten 1), zu denen auch die *Object-Points* zählen, aus Maskenentwürfen und anderen Analyseergebnissen auf den Gesamtaufwand zu schließen (vgl. [Bun00]).

Häufig stehen wir aber vor dem Problem, noch früher einigermaßen abgesicherte Aussagen zu Aufwänden machen zu müssen. Weder *Object-Points* und erst recht kein Code stehen uns so früh zur Verfügung. In diesem Artikel beschreiben wir ein Verfahren zur Aufwandsschätzung, das auf noch früheren Analyseergebnissen beruht und somit bereits zu Beginn eines Projekts gut einsetzbar ist. Unsere Grundlage ist die *essenzielle Beschreibung von Use-Cases*, also von *Anwendungsfällen*.

Die Grundidee der Aufwandsschätzung über Use-Cases

Die These, die auf einer Idee von Bernd Oestereich beruht, lautet, dass es eine Korrelation zwischen der Anzahl der Essenzschritte eines Anwendungsfalls und seinem Realisierungsaufwand gibt (was wir unter essenziellen Anwendungsfällen verstehen, wird im übernächsten Abschnitt erläutert). Anders formuliert, besteht eine Beziehung zwischen der Anzahl der Essenzschritte und der Komplexität des Anwendungsfalls und damit seiner Umsetzung. Die Messgröße ist die *Schrittzahl n*. Mathematisch ausgedrückt heißt das:

$$\text{Aufwand} = a_1 \cdot n$$
$$\text{Komplexität} = a_2 \cdot n$$

wobei a_1 und a_2 die entsprechenden Proportionalitätsfaktoren sind.

Welche Art des Aufwands in Korrelation zur Schrittzahl gebracht wird, ist recht frei. Wir können sowohl

- Analyseaufwand,
- Realisierungsaufwand inklusive Tests oder den
- Gesamtaufwand

die autoren

Uwe Vigenschow (E-Mail: uwe.vigenschow@oose.de) ist Berater und Trainer bei der oose.de GmbH in Hamburg. Er beschäftigt sich seit 1991 mit Objektorientierung und seit 1997 mit Qualitätsmanagement.

Christian Weiss (E-Mail: christian.weiss@oose.de) ist ebenfalls Berater und Trainer bei der oose.de GmbH. Seine Schwerpunkte liegen im Bereich objektorientierter Analyse und Geschäftsprozessmodellierung.

Function-Points sind ein von der Programmiersprache und damit auch von den LOC unabhängiges Maß zur Bestimmung der Programmgröße. Dabei sind ausschließlich solche Programmfunktionen zu berücksichtigen, die der Benutzer eines Systems sehen kann. Die Function-Points können also nicht so einfach gezählt werden, sie liefern ein indirektes Maß. Erschwerend kommt hinzu, dass die Komplexitätsunterschiede von Funktionen durch Gewichtungen Berücksichtigung finden. Um trotzdem eine gewisse Reproduzierbarkeit und damit Vergleichbarkeit zu erlangen, hat die International Function Point Users Group strikte Regeln zur Gewichtung erlassen.

Als *Widget* wird ein einzelnes GUI-Element bezeichnet, das jeweils vom Programmierer spezifiziert werden muss. Da sich Komplexitäten innerhalb eines Programms zumeist auch in der Benutzerschnittstelle widerspiegeln, kann auch die Summe der einzelnen Widgets als Größenmaß genommen werden. Widgets sind fein-granularer als Function-Points und können auch ungewichtet gut verwendet werden. Es gibt einen empirisch ermittelten Bezug zwischen Widget und Function-Points:

```
1 Widget-Point = 2,28 +/-
   0,34 Function-Point
```

Beide Verfahren sind sinnvoll nur bei GUI-lastigen Programmen einsetzbar. Eine algorithmische Komplexität kann so nur erfasst werden, wenn sie sich direkt in der GUI bzw. in Benutzerfunktionen widerspiegelt (vgl. [Oes01-b]).

Kasten 1: Function-Points und Widget-Points

betrachten. Aus den beschriebenen Anwendungsfällen wird eine Projekt-Aufwandsformel abgeleitet, die für jeden Anwendungsfall anhand seiner Schrittzahl den Aufwand berechnet.

Unsere praktischen Erfahrungen aus einem Großprojekt mit vorgeschaltetem Pilotprojekt stützen diese These. Bevor wir auf die Einzelheiten und Lösungen für die praktischen Probleme des Verfahrens kommen, müssen wir jedoch einen gemeinsamen, einheitlichen Blick auf das UML-Modellelement des *Anwendungsfalls* werfen.

Der Use-Case in der Praxis

Die UML lässt uns in der praktischen Arbeit mit Anwendungsfällen etwas im Regen stehen und definiert den Use-Case wie folgt: „The use case construct is used to define the behavior of a system or other semantic entity without revealing the entity's internal structure. Each use case specifies a sequence of actions, including variants, that the entity can perform, interacting with actors of the entity" (aus [OMG01]).

Diese Definition lässt reichlich Interpretationsspielraum. Was für Systeme sind gemeint? Müssen alle Varianten berücksichtigt werden? Es erfolgt also eine Beschreibung einer Abfolge von Schritten oder Aktivitäten. Können es nur Sequenzen sein? Nach welchen Kriterien finden wir Anfang und Ende von Anwendungsfällen? Und wie werden sie strukturiert? Und so weiter, und so fort.

Zur praktischen Nutzbarkeit von Anwendungsfällen sollten diese durch Ergänzungen der Definition weiter konkretisiert werden. Als sinnvoll haben sich die folgenden Zusätze erwiesen (vgl. [Oes01-a]):

- Ein Anwendungsfall hat keine fachlich vorgesehene Unterbrechung.
- Ein Anwendungsfall hat genau einen fachlichen Auslöser.
- Am Ende eines Anwendungsfalls steht mindestens ein Ergebnis von fachlichem Wert.

So kann beispielsweise eine klare Abgrenzung des Use-Cases zum Geschäftsprozess erfolgen. Ein einzelner Schritt innerhalb eines Anwendungsfalls ist eine kohärente Beschreibung genau eines fachlichen Zwecks. Sobald es zu einer Verzweigung oder Variante kommen kann – z.B. einem fachlichen Fehler – haben wir das Ende eines Schritts erreicht.

Konkrete Anwendungsfälle haben trotz dieser Erweiterungen das Problem, dass sie schnell so unterschiedlich werden, dass wir bei der zu erwartenden hohen Anzahl von Anwendungsfällen kaum noch Nutzen daraus ziehen können. Sie unterscheiden sich z. B. nach

- Abstraktionsgrad,
- Granularität,
- Blickwinkel,
- Sprachstil,
- Verbindlichkeit und
- Umfang.

Eine Lösung dieses Dilemmas kann über die *essenziellen Anwendungsfälle* erfolgen.

Die essenzielle Beschreibung von Use-Cases

Die essenzielle Beschreibung von Anwendungsfällen geht zurück auf Ideen von Larry Constantine (vgl. [Con99]) und wurde von Bernd Oestereich verfeinert (vgl. [Oes01-a]). Ein Anwendungsfall wird dabei auf seine fachliche Intention reduziert und besteht nur noch aus wenigen aufgelisteten Sätzen, die abstrakt und frei von jeder konkreten Technologie oder Implementierung formuliert sind. Ein essenzieller Anwendungsfall ist

- vereinfacht,
- generalisiert,
- abstrahiert,
- technologieunabhängig und
- implementierungsunabhängig.

Es erfolgt also eine Reduktion auf die geschäftlichen Intentionen. Am einfachsten erkennen wir den Wert anhand eines Beispiels. Schauen wir uns dazu einen *Geldautomaten* näher an. Die essenzielle Beschreibung könnte folgendermaßen aussehen:

Beispiel: Geld am Automaten holen

1. Verfügungsberechtigten identifizieren
2. Betrag bestimmen
3. Verfügbarkeit prüfen
4. Betrag buchen
5. Geld übertragen

Durch den hohen Abstraktionsgrad ist diese Beschreibung sehr langlebig und kompakt. Sie kann auch als hervorragender Ausgangspunkt für eine detaillierte Beschreibung dienen und gibt uns dabei eine erste Struktur vor.

Schätzungen über Use-Cases

Nachdem wir ein gemeinsames Verständnis von essenziellen Anwendungsfällen haben, können wir uns der Berechnungsformel widmen. Die Bestimmung der projektbezogenen Formel für das Essenzschritt-Verfahren läuft folgendermaßen ab.

1. Möglichst viele Anwendungsfälle müssen in vergleichbarer Granularität essenziell beschrieben werden. Das hört sich einfacher an, als es ist – doch dazu später mehr.
2. Exemplarisch werden essenzielle Anwendungsfälle ausgewählt, die das bisherige Spektrum der Schrittzahlen abdecken und aufgrund der bisherigen Analyse gut erfasst sind.

Die einfachste Schätztechnik ist, mit mehreren Leuten zu schätzen und aus diesen Werten einen Mittelwert als Schätzwert zu nehmen. Die Gruppe sollte dazu möglichst heterogen sein, also Menschen mit unterschiedlichen Erfahrungshorizonten beinhalten. Da kleinere Strukturen einfacher zu schätzen sind, sollten die zu schätzenden Blöcke möglichst fein untergliedert sein, z. B. in Analyse, Realisierung, Integrationen und Tests sowie Auslieferung für jeden erkannten fachlichen Aufgabenblock. Auch sollten Sie sich für jede Einzelschätzung ausreichend Zeit lassen. Schnelle Ad-hoc-Schätzungen sind in der Regel deutlich ungenauer.

Der nächste Schritt zur Verbesserung ist es, nach der Ermittlung und Bekanntgabe der Mittelwerte im selben Team erneut zu schätzen. Unter Kenntnis des ersten Ergebnisses werden sich diverse Teilschätzungen verändern. Unserer Erfahrung nach tun sie dies meist in die richtige Richtung, sodass die jetzt zu berechnenden zweiten Mittelwerte deutlich näher an der Realität liegen als die der ersten Runde.

Jede der beiden Techniken lässt sich durch Schätzungen nach der Drei-Punkt-Methode verbessern. Diese Methode kommt aus *PERT*, der *Programming Evaluation and Review Technique*. Dabei werden für jeden Schätzwert drei Einzelwerte ermittelt, der optimistisch-minimale, der am häufigsten eintretende modale und der pessimistisch-maximale Aufwandsschätzwert. Da die Verteilung dieser und aller Zwischenwerte nicht der Normalverteilung entspricht, sondern zumeist in Richtung kleinerer Werte verschoben ist, entspricht der modale Wert nicht dem, der die Fläche unter unserer Kurve in zwei gleiche Hälften teilt, also dem Mittelwert. Den gesuchten Mittelwert, für den die Wahrscheinlichkeit, dass es besser oder schlechter läuft jeweils 50% beträgt, können wir über die folgende Näherungsformel ermitteln:

$$Aufw_{mittel} = (Aufw_{optimist} + 4 \cdot Aufw_{modal} + Aufw_{pessimist}) / 6$$

Die Bedeutung der Werte ist **Abbildung 1** zu entnehmen.

Wenn wir komplexere Schätzkurven handhaben wollen, kann das Schätzpunkt-Verfahren hilfreich sein. Hierbei wird für jede mögliche Projektdauer deren Wahrscheinlichkeit geschätzt und der Mittelwert gebildet. Die Mittelwerte über alle möglichen Zeiten bilden eine Schätzkurve, die mit optischen oder mathematischen Verfahren so aufgeteilt wird, dass die Fläche unterhalb der Kurve in zwei gleichgroße Teile zerfällt. Der Wert an dieser Grenzlinie ist unser gesuchter Schätzwert. In **Abbildung 2** ist ein Beispiel für drei Schätzpersonen zu finden.

Kasten 2: Einfache Schätztechniken im Überblick

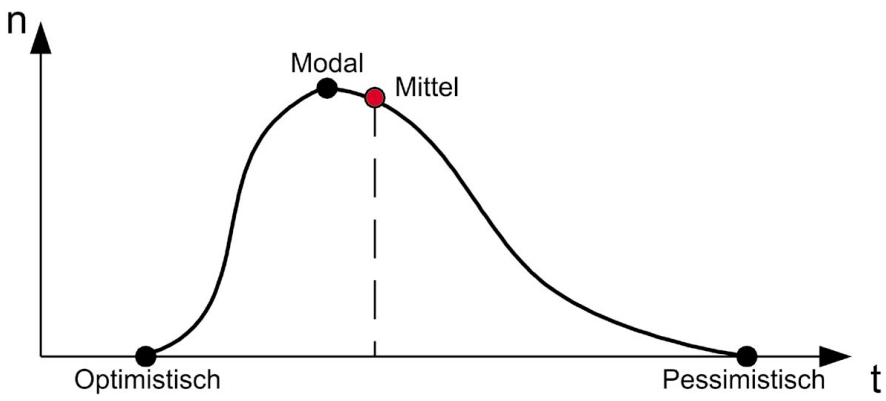

Abb. 1: Die Parameter der Drei-Punkt-Schätzung. Die Kurve entspricht einer β-Verteilung. Der Mittelwert kann näherungsweise über die in Kasten 2 angegeben Formel bestimmt werden.

3. Die ausgewählten Fälle werden von einer möglichst heterogenen Gruppe bzgl. der zu betrachtenden Art des Aufwands geschätzt. Dabei können die verschiedensten Verfahren und Techniken zum Einsatz kommen. Anregungen dazu finden Sie in **Kasten 2**.
4. Aus bereits abgewickelten Projekten wird der Bezug zwischen der Anzahl der Essenzschritte und dem zu betrachtenden Aufwand (Analyse, Realisierung, Gesamtaufwand usw.) ermittelt. Je mehr vergleichbare Projekte als Grundlage dienen, um so besser. Ein Pilotprojekt im Rahmen eines Großprojekts ist aber bereits ausreichend, um mit der Methode anfangen zu können. Dazu müssen nur die Essenzschritte der Anwendungsfälle gezählt und in Bezug zum benötigten Aufwand gebracht werden. Voraussetzung dafür ist, dass der benötigte Aufwand in einer Granularität vorliegt, die eine Zuordnung auf einzelne Anwendungsfälle zulässt. Das hört sich einfach an, aber hier treten in der Praxis reichlich Probleme auf. Doch auch dazu später mehr.
5. Die einzelnen Essenzschritte werden mit einem Komplexitätsfaktor versehen. Wie die Komplexität bestimmt werden könnte, wird im Folgenden näher betrachtet.
6. Die Schätzungen und Messwerte werden zueinander in Bezug gesetzt. Aus Regressionskurven kann eine Formel abgeleitet werden, die als Variable nur die Anzahl der Schritte kennt und den Aufwand berechnet.

Zusammenfassend müssen also die folgenden Voraussetzungen erfüllt sein, um das Essenzschritt-Verfahren einsetzen zu können.

- Die Analyse muss als Ergebnisse mindestens essenziell beschriebene Anwendungsfälle ergeben.
- Es muss mindestens ein vergleichbares Projekt oder Teilprojekt existieren, für das essenzielle Anwendungsfälle erstellt wurden und für die aus dem Projekt-Controlling heraus die geleisteten Aufwände für Analyse und Realisierung ermittelt werden können.

Bei größeren Projekten reicht bereits ein Pilotprojekt aus, um das Verfahren einzusetzen. Mit jedem weiteren Projekt wird das Verfahren über die Verbreiterung der Datenbasis immer sicherer.

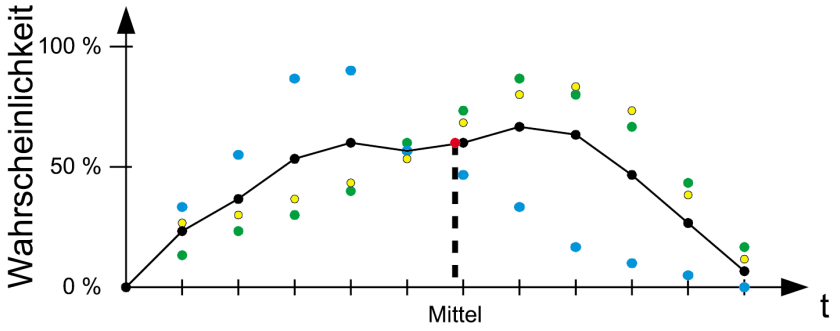

Abb. 2: Für komplexere Schätzkurven bietet sich das folgende Verfahren an: Für jede mögliche Dauer wird deren Wahrscheinlichkeit geschätzt, wobei aus praktischen Erwägungen Bereiche zusammengefasst werden. Aus den Mittelwerten der Einzelschätzungen für jede Dauer wird die Gesamtschätzkurve gebildet. Durch optische oder mathematische Verfahren wird der Wert ermittelt, der die Fläche unterhalb der Schätzkurve in zwei gleiche Hälften teilt. Dieser Wert ist unser gemeinschaftlicher Schätzwert (siehe Kasten 2).

Abb. 3: Die möglichen Formen der Regressionskurven im schematischen Überblick. Ist die Tiefe der Essenzschritte einheitlich, wird die Regressionskurve eine Gerade sein (links). Haben wir insbesondere bei den komplexen Anwendungsfällen eine uneinheitliche Tiefe, so wird die Kurve überproportional steigen und die Genauigkeit des Verfahrens abnehmen (rechts).

einheitlich sein. Je komplexer ein Anwendungsfall ist, desto eher werden grobgranulare Schritte beschrieben. Trotzdem lässt sich das Verfahren unter Einbußen der Genauigkeit anwenden. Die Regressionskurven sind dann keine Geraden mehr, sondern steigen mit zunehmender Schrittzahl überproportional an. Der Sachverhalt ist schematisch in **Abbildung 3** dargestellt.

Die Bestimmung der Formel erfolgt empirisch bei der Betrachtung verschiedener Regressionskurven durch das Datenmaterial. Dazu sind keine aufwändigen Statistikprogramme erforderlich, obwohl diese dabei durchaus hilfreich sein können. Aber mit einer Tabellenkalkulation wie Excel geht das bereits in ausreichender Form. Ein brauchbares Gefühl für das Verfahren stellt sich schnell ein, wenn gemessene und geschätzte Aufwände einander in einer Grafik gegenüber gestellt wie in **Abbildung 4**.

Bei einfacheren, kleineren Projekten, wie z. B. einem vorgeschalteten Pilotprojekt, kann der Bezug zwischen der Anzahl der Essenzschritte und dem benötigten Aufwand noch einfach ermittelt werden. Unsere Erfahrungen bestätigen dafür das erwartete lineare Verhalten. Die Schätzungen für Anwendungsfälle des Gesamtprojekts weisen einige Ausreißer auf und werden sich insbesondere zu frü-

Details des Essenzschritt-Verfahrens

Unsere Erfahrungen mit der Essenzschritt-Methode sind noch nicht umfangreich genug für ein statistisch abgesichertes Verfahren. Aber einiges lässt sich bereits jetzt feststellen.

Das Verfahren funktioniert gut, wenn die Homogenität in der Tiefe der Essenzschritte gewährleistet ist. Die Regressionskurven sind dann Geraden bzw. sie sind Geraden sehr ähnlich. Variationen bzgl. der Komplexität von Projekten ergeben sich dann in der Verschiebung der Geraden entlang der Y-Achse bzw. in Variationen der Steigung. Betrachten wir folgende Geraden-Gleichung

$$Aufwand = a \cdot n_{Essenzschritte} + b$$

genauer, so liefert b uns den Parameter für die Komplexität und a den projektspezifischen Verlauf der Geraden.

Zu einem frühen Zeitpunkt im Projektverlauf wird die Granularität der Essenzschritte in den Anwendungsfällen nicht

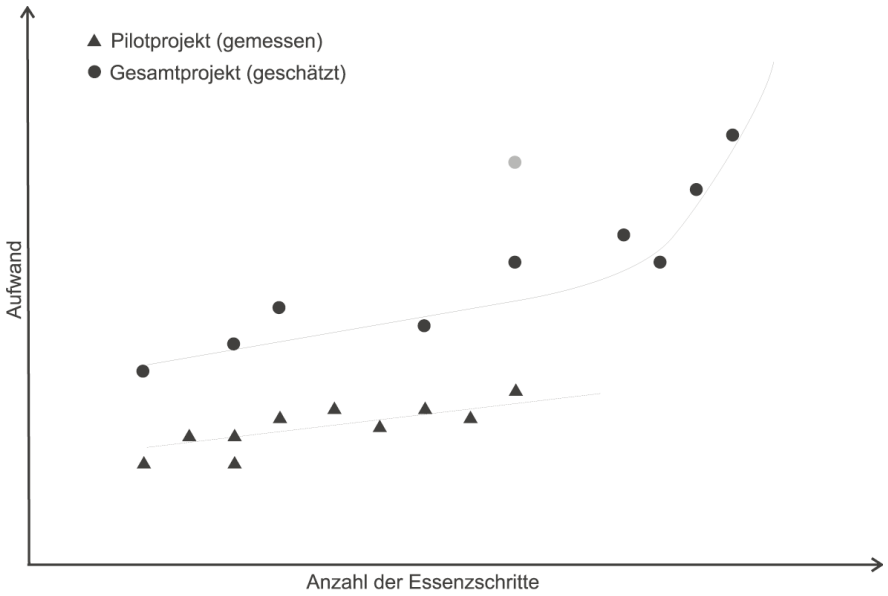

Abb. 4: Durch die Gegenüberstellung gemessener zu geschätzten Aufwänden in einer Grafik werden das Prinzip wie auch typische Probleme in der Praxis deutlich. Bei einem einfacheren Pilotprojekt kommt der lineare Bezug klar heraus. Die geschätzten Aufwände für das Gesamtprojekt weichen nicht nur von dem linearen Verhalten ab, sondern liegen auch in einer anderen Ebene. Dazu kommen Ausreißer wie der grau gefärbte Kreis.

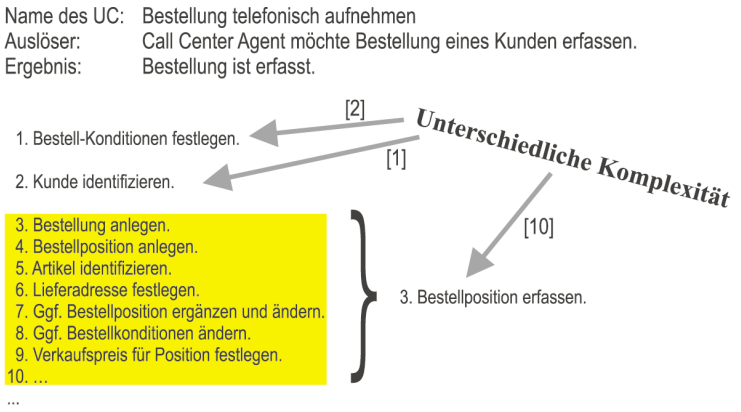

Abb. 5: In der praktischen Arbeit mit Anwendungsfällen taucht schnell das Problem auf, dass die einzelnen Essenzschritte eine unterschiedliche Granularität aufweisen. Die gelb unterlegten Schritte sind viel zu fein-granular und können zu einem Schritt auf Elementarniveau zusammengefasst werden. Trotzdem sind die verbleibenden drei Schritte unterschiedlich komplex, was durch die Werte in eckigen Klammern angedeutet ist.

hen Zeitpunkten nicht so schön linear verhalten. Der genaue Bezug kann durch verschiedene Ausgleichskurven bestimmt werden. Je komplexer ein Projekt ist, desto höher wird die Kurve im Diagramm liegen.

Komplexität: ein komplexes Problem

Um das Essenzschritt-Verfahren einsetzen zu können, brauchen wir ein einheitliches Niveau unserer Essenzschritte. Das klingt einfacher als es ist. In der Praxis taucht leider sehr häufig das Problem auf, dass die Essenzschritte alles andere als auf einheitlichem Niveau sind. Ein Beispiel dazu ist **Abbildung 5** zu entnehmen.

Dabei gibt es ein einfaches Verfahren, um ein einheitliches Niveau zu erreichen. Dazu überführen wir die essenzielle Beschreibung in ein einfaches Aktivitätsdiagramm und betrachten alle fachlichen Ausgänge der gesamten Einheit. Jede dieser Ausgänge definiert ein Schritt-Ende. In **Abbildung 6** finden wir ein Beispiel dafür, welches das Vorgehen illustriert. Die so gefundenen Schritte liegen alle auf einem einheitlichen Niveau.

Um die Komplexität von Essenzschritten auf der obersten Ebene oder sogar auf Ebene des Anwendungsfalls insgesamt zu bestimmen, wird die Anzahl der elementaren Aktivitäten innerhalb eines Essenzschritts ermittelt. Eine elementare Aktivität lässt sich nicht mehr weiter unterteilen, ohne gegen die Regeln für essenzielle Use-Cases zu verstoßen. Die einzelnen Aktivitäten in den verschiedenen Abstraktionsebenen spannen also für jeden Anwendungsfall einen Baum auf, dessen Blätter gezählt und an den Knoten summiert werden. Die Summen an den Knoten sind unser Maß für die Komplexität eines Schritts bzw. des ganzen Anwendungsfalls. In **Abbildung 7** ist dies schematisch dargestellt.

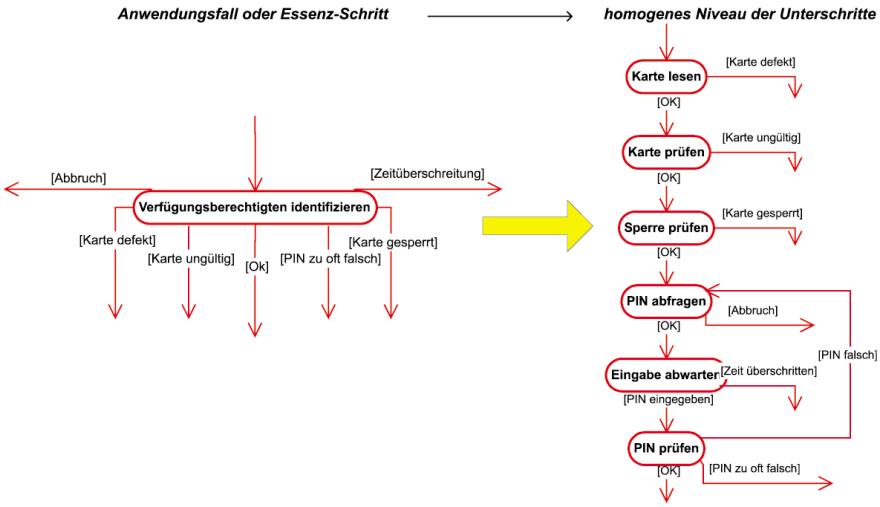

Abb. 6: Wie kommen wir auf ein einheitliches Niveau unserer Schritte? Die fachlich motivierten Ausgänge eines abstrakteren Schritts, wie im obigen Beispiel, oder eines ganzen Anwendungsfalls helfen uns dabei (links). Jeder dieser Ausgänge bestimmt dabei das Ende eines Schritts. Alle so gefundenen Schritte liegen auf einem gemeinsamen Niveau (rechts).

Innerhalb eines Projekts bzw. mit einem festen Team auch über mehrere Projekte hinweg ist es so möglich, eine einheitliche Granularität der Essenzschritte zu erreichen. Wollen wir nachträglich verschiedene Projekte vergleichen, wird eine einheitliche Granularität kaum gegeben sein. Um solche Projekte unserer Datenbasis hinzuzufügen, benötigen wir eine projektübergreifende Messgröße, um unser Verfahren quasi zu kalibrieren. *Widget-Point*-Verfahren (**siehe Kasten 1**) bieten diesen Vorteil (vgl. [Oes01-b]). Für zukünftige Projekte kann über das oben beschriebene Verfahren eine gewisse Einheitlichkeit der Granularität der Essenzschritte erreicht werden.

Das Problemkind: der Bezug zwischen Use-Case und Aufwand

Es hört sich so einfach an: den Aufwand für einen Use-Case ermitteln. Aber der Teufel steckt wie immer im Detail. Eine inkrementell-iterative Planung erfolgt auf der Basis von Arbeitspaketen, die auf Iterationen verteilt werden. Ein Arbeitspaket ist eine deutlich kleinere Größe als ein Use-Case. Auch lassen sich nicht alle Arbeitspakete einem Anwendungsfall zuordnen (vgl. [Oes01-b]). Ein Arbeitspaket zum Aufbau einer Testumgebung ist ein solches Beispiel.

Wie kann der geforderte Bezug zwischen Anwendungsfall und Aufwand hergestellt werden? Wir sehen da zwei Möglichkeiten:

1. Möglichkeit

Wenn sich die meisten Arbeitspakete eindeutig einem Anwendungsfall zuordnen lassen, findet diese Zuordnung statt. Nicht zuordenbare Arbeitspakete werden in zwei Bereiche aufgeteilt:

- Pauschalaufwände, die keine Abhängigkeit zur Anzahl der Use-Cases haben, wie das Beispiel mit dem Aufbau einer Testumgebung, und

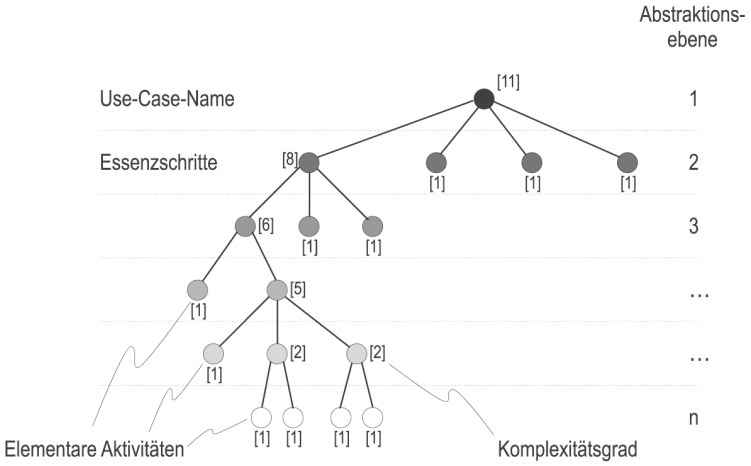

Abb. 7: Auf Basis der einzelnen Schritte auf den unterschiedlichen Abstraktionsebenen einer essenziellen Beschreibung kann ein Maß für die Komplexität eines einzelnen Schritts bzw. des ganzen Anwendungsfalls definiert werden: Die Anzahl der unter einem Schritt hängenden, elementaren Aktivitäten bildet die Messgröße für die Komplexität eines Schritts. Eine elementare Aktivität ist nicht mehr weiter in essenzielle Schritte einer tieferen Abstraktionsebene zu unterteilen.

- Arbeitspakete, die eine Abhängigkeit von der Anzahl und Komplexität der Use-Cases haben. Beispiele dafür sind das Erstellen eines Anwendungsfall-Diagramms, das Aufstellen von Feature-Listen oder andere Analysetätigkeiten.

Die erste Gruppe wird dem Gesamtaufwand pauschal geschätzt mit hinzugeschlagen. Für die zweite Gruppe kann auf Basis statistischer Werte eine Aufwandsformel gefunden werden, welche die Proportionalität ausdrückt. Mathematisch ließe sich das folgendermaßen ausdrücken:

$$Aufw_{gesamt} = Aufw_{UC1} + Aufw_{UC2} + \ldots + Aufw_{UCn} + Auf_{pauschal} + Aufw(n)$$

2. Möglichkeit

Der Gesamtaufwand des Referenzprojekts, also z. B. unseres Pilotprojekts, wird ermittelt und gewichtet auf die Anwendungsfälle verteilt. Die Gewichtung wird dabei geschätzt und unter Umständen durch einige Werte des Projekt-Controllings gestützt. Für jeden Anwendungsfall wird also sein prozentualer Anteil am Gesamtaufwand geschätzt. Der Zusammenhang der Aufwandsanteile ist also:

$$Aufw_{gesamt} = Anteil_{UC1} + Anteil_{UC2} + \ldots + Anteil_{UCn}$$

Ergo

Beide Verfahren haben ihre Schwächen. Wir möchten aber unsere Intention des Essenzschritt-Verfahrens betonen: Es geht uns darum, frühe Schätzungen zu einem Zeitpunkt, wo andere Verfahren noch nicht greifen, auf eine bessere Grundlage zu stellen.

Bei allen zwangsläufigen Unzulänglichkeiten kanalisiert das Essenzschritt-Verfahren unsere *Bauch-Schätzungen* und gibt uns ein erstes Maß, den Komplexitätsbegriff besser fassen zu können. Wie jedes statistische Verfahren wird es mit breiterer Datenbasis immer besser.

Produktgröße vs. Aufwand

Genau betrachtet liefert die Analyse der Essenzschritte von Anwendungsfällen eine Schätzung der Produktgröße (**in den Abb. 3 und 4** sollten dann die Ordinate die Produktgröße beschreiben und damit nur indirekt den Aufwand).

Betrachten wir nur ein Projekt, ist dieser Unterschied marginal, da das Verhältnis von Aufwand zu Produktgröße eine Konstante liefert. Wollen wir hingegen zukünftig mehrere, unterschiedliche Projekte vergleichen, so muss der Bezug zwischen Aufwand und Produktgröße explizit hergestellt werden. Dies kann über einen Analogieschluss erfolgen, wie er im vorherigen Abschnitt erläutert wurde. Es gibt aber noch andere Möglichkeiten.

In der Literatur sind diverse Gleichungen und Faustformeln für diesen Bezug aufgestellt worden. Am bekanntesten ist sicher die Softwaregleichung von Putnam aus den 70er Jahren, die er auf der Grundlage statistischer Analyse ermittelt hat. Dabei werden Produktgröße, Projektdauer und Projektkosten in Relation gesetzt. Besonders interessant erscheinen die von Hartmut Krasemann modifizierten Faustformeln von Capars Jones. Sie sind einfach in der Anwendung und mit ausreichender Datenbasis aktueller Projekte in Java, C++ und Smalltalk unterlegt (vgl. [Oes01-b]).

Wie kann es weiter gehen?

Um die Essenzschritt-Methode zu einer *richtigen* Schätzmethode auszubauen, bedarf es weiterer Praxiserfahrung. Insbesondere muss die Datenbasis abgeschlossener Projekte deutlich verbreitert werden. Das Verfahren scheint aber prinzipiell dazu in der Lage zu sein, bereits aus frühen Analyseergebnissen Schlüsse auf den resultierenden Aufwand zu ziehen. Es ist einfach in der Umsetzung und könnte zuerst ohne großen Aufwand parallel zu anderen Verfahren eingeführt werden. So kann die notwendige Sicherheit erreicht werden, um die aus der Essenzschritt-Methode gewonnenen Ergebnisse als Entscheidungsgrundlage zu nutzen. In Kürze wollen wir einen ersten Erfahrungsbericht mit dem Essenzschritt-Verfahren veröffentlichen.

Die Voraussetzungen, die für die Essenzschritt-Methode erfüllt sein müssen, sind kein Selbstzweck, sondern sinnvolle Schritte im Rahmen der Analyse eines Softwareprojekts. Der Mehrwert an Information ist beträchtlich. ■

Literatur

[Boe81] B.W. Boehm, Software Economics, Prentice Hall, 1981

[Boe00] B.W. Boehm et al, Software Cost Estimation with COCOMO II, Prentice Hall, 2000

[Bun00] M. Bundschuh, A. Fabry, Aufwandsschätzungen von IT-Projekten, MITP, 2000

[Con99] L.L. Constantine, L.A.D. Lockwood, Software for Use – A Practical Guide to the Models and Methods of Usage-Centered Design, Addison-Wesley, 1999

[Oes01-a] B. Oestereich, Objektorientierte Softwareentwicklung – Analyse und Design mit der UML, 5. völlig überarbeitete Auflage, Oldenbourg, 2001

[Oes01-b] B. Oestereich et al, Erfolgreich mit Objektorientierung – Vorgehensmodelle und Managementpraktiken, 2. Auflage, Oldenbourg, 2001

[OMG01] OMG UML Specification 1.4, 2001 (siehe http://www.omg.org/uml)