



KMU-innovativ

Softwaresysteme und Wissenstechnologien

Methodenhandbuch

Abakus – Eine softwaregestützte Methode zur Kalkulation von Softwareprojekten mittels vergleichender Schätzver- fahren

Version: 1.1
Datum: 2. Januar 18
Autoren: Hartmut Schmitt

Zielsetzung des Abakus-Vorhabens ist die Entwicklung einer softwaregestützten Schätzmethode, die KMU erstmals die Möglichkeit eröffnet, schneller und in einer besseren Qualität als bisher IT-Vorhaben zu planen, zu kalkulieren und rentabel umzusetzen. Der Schätzmethode liegt ein neuartiges Experience Management System (EMS) zugrunde, welches strukturierte Informationen zu abgeschlossenen Projekten enthält. Die Methode verfügt weiterhin über eine intelligente Assistenz. Diese ist in der Lage, relevante aufwands- sowie komplexitätstreibende Faktoren abzufragen und auszuwerten. Das Ergebnis der Methode ist eine strukturierte, möglichst genaue und nachvollziehbare Kalkulation der Projektaufwände.

Das diesem Bericht zugrundeliegende Vorhaben wird mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS15050 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

Inhalt

1	Einleitung	3
1.1	Motivation	3
1.2	Stand der Wissenschaft und Technik	5
1.2.1	Verfahren für die Schätzung von Software-Entwicklungskosten.....	5
1.2.2	Methoden zur Repräsentation und Ähnlichkeitsbestimmung von Anforderungen	6
1.3	Entwicklung der Abakus-Methode	8
1.3.1	Requirements-Engineering-Praxis der an Abakus beteiligten KMU.....	8
1.3.2	Schätzpraxis der an Abakus beteiligten KMU	10
2	Die Abakus-Methode	12
2.1	Phasen/Prozesse	12
2.1.1	Anforderungsanalyse.....	12
2.1.2	Aufwandsschätzung	13
2.1.3	Ergebnisaufbereitung	13
2.1.4	Kalibrierung.....	13
2.2	Rollen.....	14
2.3	Aktivitäten	17
3	Praxisbeispiel.....	30
4	Prinzipien guter Schätzung.....	33
5	Quellen	34

Abbildungen

Abbildung 1: Lösungsskizze der Abakus-Methode	12
Abbildung 2: Kalkulation eines Features im Abakus-Werkzeug.....	30
Abbildung 3: Schätzung von Funktionen im Abakus-Werkzeug	32

1 Einleitung

In kleinen und mittleren Unternehmen (KMU) existiert zurzeit kein geeignetes Standardverfahren, mit dem heterogene Kundenanforderungen präzise erfasst und sicher kalkuliert werden können. In der Regel werden für jedes Projekt entsprechend der individuellen Aufgabenstellung verwandte Dokumente aufwendig recherchiert und daraus relevante Kalkulationselemente übernommen. Auf dieser Basis erfolgt die Kalkulation für jedes Projekt neu, weitgehend unstrukturiert und die Elemente stehen in keiner Verbindung zueinander. Die für eine valide Kalkulation grundlegend wichtige Rückführung der Erfahrungswerte aus abgeschlossenen Projekten erfolgt nicht oder aufgrund fehlender Werkzeugunterstützung nur unzureichend. Auch ist es für KMU häufig schwierig, Kostentreiber gegenüber dem potenziellen Kunden transparent zu machen.

Ziel des Abakus-Vorhabens¹ war daher die Entwicklung einer werkzeuggestützten Schätzmethodik, die eine möglichst präzise Ermittlung des Projektumfangs bereits in der Akquisephase ermöglicht. Anders als bei den bekannten algorithmischen oder hybriden Verfahren sollte bei dieser Methode keine präzise Quantifizierung des Projektumfangs auf Basis abstrakter Metriken erforderlich sein, sondern die Schätzer sollten bei der Festlegung des Mengengerüsts durch eine semi-automatisierte Analyse relevanter Kostenfaktoren unterstützt werden. Dadurch sollte diese neuartige, werkzeuggestützte Methode zuverlässiger sein und in der Anwendung weniger aufwändig als klassische Kalkulationsansätze.

1.1 Motivation

Zuverlässige Aufwands- und Kostenschätzungen seitens der Auftragnehmer sind von größter Bedeutung für die erfolgreiche Umsetzung von Softwareentwicklungsprojekten. Unzureichende Aufwandsschätzungen sind häufig die Ursache einer fehlerhaften Planung des Projektumfangs und führen zu einer Fehlkalkulation der Projektkosten und -dauer. Insbesondere bei KMU können Fehlschätzungen leicht zu einem Liquiditätsengpass, im schlechtesten Fall sogar zur Insolvenz führen. Zu hoch angesetzte Kosten sind hingegen in der Regel nicht konkurrenzfähig und führen dazu, dass Angebote bei der Vergabe von Aufträgen bzw. in EU-Ausschreibungen nicht berücksichtigt werden. Neben den finanziellen Risiken für KMU führen Fehlschätzungen auch dazu, dass Leistungen nur in reduziertem Umfang erbracht oder verspätet ausgeliefert werden, was sowohl innerhalb des Unternehmens als auch in der Kommunikation mit Kunden zu Spannungen und Frustration führen kann. Dies erzeugt einerseits Vertrauens- und Akzeptanzprobleme, insbesondere wenn Kostendifferenzen nicht plausibel begründet werden können, andererseits besteht die Gefahr von Ressourcenengpässen bei der Bearbeitung von Projekten. Ein wesentlicher Grund für solche problematischen Fehlschätzungen ist das Fehlen verlässlicher Schätzungen zu Projektbeginn bzw. in der

¹ Das Bundesministerium für Bildung und Forschung fördert das zweijährige Projekt Abakus im Rahmen der Maßnahme „KMU-innovativ: IKT“ (Förderkennzeichen: 01IS15050). Das Konsortium besteht aus den Unternehmen ICT Solutions, [site]VERTREIBER, HK Business Solutions, Insiders Technologies und OSSENO Software sowie der Universität Trier und dem Fraunhofer-Institut für Experimentelles Software Engineering IESE. Projektträger ist das Deutsche Zentrum für Luft- und Raumfahrt.

vorgelagerten Akquisephase. Zu diesem Zeitpunkt liegen zumeist nur grobe Anforderungen vor. Zudem sind die technischen und organisatorischen Rahmenbedingungen nicht in voller Detailtiefe geklärt. Dies führt bei den Projektverantwortlichen häufig zu einer Unterschätzung der Projekt- und/oder Systemkomplexität und damit zu Fehlkalkulationen.

Es hat sich bislang **kein generalisierbarer, für KMU geeigneter Ansatz** zur Aufwandsschätzung durchgesetzt: Rein algorithmische bzw. datenbasierte Schätzmethoden¹, die auf der Messung und expliziten Berücksichtigung von Aufwandstreibern basieren, sind komplex und abstrakt, wodurch sie für KMU eher ungeeignet sind. Vor allem die Hauptaufwandstreiber Projektumfang (Größe) und Komplexität können bei typischen Projekten im KMU-Umfeld in einer frühen Projektphase meist nicht geeignet erfasst werden, weil die zugrundeliegenden Softwareartefakte nicht oder nur unvollständig existieren und dementsprechend eine Abschätzung der Kostentreiber nur schwer möglich ist. Diesen Umstand adressieren agile Schätzmethoden zwar durch abstrakte subjektive Metriken wie z. B. *Story Points*². Diese sind für viele Verantwortliche allerdings nur schwer verständlich und werden unterschiedlich interpretiert, so dass es zu inkonsistenten Berechnungen der erwarteten Kosten und der Projektdauer kommt. Auch die Vermessung rein funktionaler Eigenschaften stellt eine Herausforderung dar, weil benötigte Informationen in frühen Phasen oft noch gar nicht vollständig spezifiziert sind oder weil diese Vermessung einen hohen zeitlichen wie personellen Aufwand verursacht.

Bei den am Abakus-Vorhaben beteiligten KMU kamen meist expertenbasierte Schätzverfahren zum Einsatz, deren Grundlage die Erfahrungen aus Vorgängerprojekten waren. Die Fehler bei der Zeit- und Kostenschätzung bewegten sich bei diesen KMU in einer Größenordnung von -20 % (weniger benötigt als geplant) bis +70 % (mehr benötigt als geplant), zumeist im Bereich +30 % bis +50 %, wobei insbesondere bei Klein- und Kurzprojekten durchaus auch Abweichungen bis +400 % vorkamen. Dabei zeigte die Erfahrung, dass selbst finanziell gut geschätzte Projekte hinsichtlich der Projektlaufzeit häufig um 10–30 % überschritten werden, da die Kunden ihre eigenen Leistungs- und Mitwirkungsfähigkeiten überschätzten.

1.2 Stand der Wissenschaft und Technik

1.2.1 Verfahren für die Schätzung von Software-Entwicklungskosten

Aufwands- und Kostenschätzungen gehören zu den zentralen Aspekten des Software-Projektmanagements¹. Für ein effektives Projektmanagement sind zuverlässige Aufwands- und Kostenschätzungen unabdingbar, da mangelhafte Aufwandsschätzungen häufig Ursache für große Kostensteigerungen und letztendlich für das Scheitern von Entwicklungsprojekten sind. Ein effektives Projektmanagement bedingt im Besonderen die Möglichkeit, schnell und flexibel auf sich ändernde Anforderungen an die Software reagieren zu können. Deshalb setzen insbesondere KMU zunehmend auf den Einsatz agiler Entwicklungsmethoden. Hierbei wird ein iterativer, inkrementeller Entwicklungsprozess verwendet, der sich schnell an die aktuelle Projektsituation anpassen lässt und mit dem flexibel auf Änderungen von Anforderungen (Change Requests) reagiert werden kann. Doch auch mit agilen Entwicklungsmethoden muss der Aufwand für notwendige Softwareänderungen möglichst zuverlässig abgeschätzt werden. Hierzu wird üblicherweise auf expertenbasierte, algorithmische, hybride oder analogiebasierte Schätzmethoden zurückgegriffen.

Im Kontext der agilen Entwicklung werden Aufwandsschätzungen primär auf der Basis von Expertenmeinungen abgeleitet². Solche **expertenbasierte Schätzverfahren**^{2,3} basieren ausschließlich auf Meinungen eines oder mehrerer Experten. Basierend auf der Produktspezifikation und eigenen Erfahrungswerten aus bereits abgeschlossenen Projekten schätzen die Experten die Kosten, die für die Entwicklung benötigt werden. Da expertenbasierte Schätzverfahren keine quantitativen Messdaten erfordern, werden sie häufig eingesetzt. Für KMU sind solche rein expertenbasierten Methoden im agilen Kontext jedoch nur bedingt geeignet, da

- jedes Mal, wenn neue Schätzungen erforderlich sind, erfahrene Mitarbeiter verfügbar sein müssen,
- jede Schätzung mit hohem personellen Aufwand verbunden ist, und
- jeder Schätzung eine subjektive Beurteilung zugrunde liegt, was zu ungenauen Prognosen führen kann.

Das Ergebnis solcher Schätzungen kann von Experte zu Experte jedoch stark variieren. Die Ursachen hierfür liegen darin, dass Experten oftmals zu optimistisch sind, ihre eigenen Fähigkeiten überschätzen oder wesentliche Risikofaktoren im Projektkontext falsch beurteilen⁴.

Um diese Probleme zu vermeiden, werden in der Forschung **algorithmische Schätzverfahren**⁶ vorgeschlagen. Diese sollen beeinflussende Effekte des Urteilsvermögens von Experten reduzieren oder kompensieren. Die Methoden helfen dabei, Risikofaktoren (z. B. Neuartigkeit des Softwareprodukts oder Erfahrungen der Entwickler) und deren Einfluss auf den Projektaufwand explizit und systematisch zu berücksichtigen. Bei den algorithmischen Schätzmethoden muss jedoch explizit ein umfangreiches Mengengerüst bezüglich des geplanten Softwareprodukts ermittelt werden. Hierfür werden z. B. *Function Points*⁵ beziehungsweise im agilen Umfeld *Story Points* vorgeschlagen. Die Ermittlung dieses Mengengerüsts ist aber in frühen Phasen eines Projektes, in denen kaum strukturierte Artefakte vorliegen, schwierig bzw. unmöglich. Daher werden diese Methoden in der Praxis selten eingesetzt. Darüber hinaus erfordern algorithmische Schätzungen größere Mengen quantitativer Messdaten und sie setzen Kenntnisse von teilweise sehr komplexen Analyseverfahren voraus¹. Dieses Know-how ist jedoch nur selten in einem KMU vorzufinden.

Für die Durchführung dieser Verfahren stehen proprietäre Schätzmodelle wie z. B. COCOMO⁶ mit speziell darauf ausgerichteten Schätzwerkzeugen zur Verfügung. Auch diese Modelle benötigen jedoch eine ausreichend große Menge quantitativer Messdaten und liefern lediglich grobe Schätzungen.

Das Ziel **hybrider Schätzverfahren**^{7,8,9} ist es, die Vorteile expertenbasierter Ansätze mit den Vorteilen quantitativ-struktureller Ansätze in einer hybriden Methodik zu vereinen. Hierzu werden quantitative Messdaten (z. B. über strukturelle Änderungen oder Änderungen der inneren Qualität) mit expertenbasierten Einschätzungen (z. B. Welche Bausteine sind zu ändern? Wie groß ist der zu erwartende Änderungsaufwand für einen Baustein?) kombiniert. Dies erlaubt es dem Projektverantwortlichen, quantitative Informationen mit den Schätzungen von Experten abzugleichen und anschließend den Aufwand für die notwendigen Änderungen der Software zu prognostizieren. Diese benötigen jedoch ebenfalls Angaben zur Größe und Komplexität, z. B. in Form von *Function Points*.

Um die oben genannten Probleme zu vermeiden werden **analogiebasierte Ansätze**¹⁰ vorgeschlagen, mit denen der Aufwand eines geplanten Projektes anhand ähnlicher Projekte aus der Vergangenheit abgeleitet werden kann. Der Abgleich erfolgt dabei jedoch in der Regel auf Basis einer Analyse von gesamten Projekten, nicht von deren Artefakten, z. B. in Form einzelner Anforderungen oder kontextrelevanter Informationen. Um mit Analogiemethoden annähernd präzise schätzen zu können, wird daher eine möglichst große Sammlung von in der Vergangenheit abgeschlossenen Projekten vorausgesetzt, die dann z. B. mittels Methoden des *Case-based Reasoning*¹¹ analysiert werden können.

Eine KMU-taugliche Aufwandsschätzung erfordert daher einen neuen, maßgeschneiderten Ansatz, der die Vorteile der verschiedenen Schätzmethoden kombiniert und deren Nachteile minimiert. Um in der Praxis die Akzeptanz für einen solchen Ansatz zu erhöhen, braucht man eine zusammenhängende Kette automatisierter Werkzeuge, die das Projektteam beim Zusammenführen relevanter Informationen, bei der Analyse von Ähnlichkeiten zu früheren Softwareprojekten sowie bei der Aufwandsvorhersage und der Abschätzung ihrer Genauigkeit unterstützt. Eine solche softwaregestützte Projektkalkulation bietet KMU erstmals die Möglichkeit, schneller und in einer besseren Qualität als bisher, IT-Vorhaben zu planen, zu kalkulieren und rentabel umzusetzen.

1.2.2 Methoden zur Repräsentation und Ähnlichkeitsbestimmung von Anforderungen

Sowohl in der Praxis als auch in der Forschung und Wissenschaft werden Anforderungen zumeist in textueller (d. h. in natürlicher Sprache) oder in modellbasierter (d. h. in semi-formaler Sprache) hinterlegt. Die Anforderungen lassen sich hierbei in zwei Gruppen unterteilen: Funktionale Anforderungen legen fest, was die Software leisten soll; nichtfunktionale Anforderungen spezifizieren weitere (Qualitäts-)Eigenschaften der Software¹².

In der Praxis am häufigsten vorzufinden sind Anforderungen in **textueller Form**, da zur Nutzung der natürlichen Sprache keine besonderen Kenntnisse bezüglich Notationen oder Werkzeuge notwendig sind. Eine mögliche Repräsentationsform sind Use-Case-Beschreibungen¹³, die detailliertere Informationen bezüglich der Interaktionen innerhalb eines *Use Cases* enthalten. Die Spezifikation erfolgt zumeist gemäß einer ta-

bellarischen Vorlage. Textuelle Anforderungen können auch gemäß einer Satzschablone formuliert werden, die die Satzstruktur und Inhalte einer Anforderung vorgeben um gewisse Qualitätseigenschaften der Anforderungen sicherzustellen oder deren Sicherstellung zumindest zu unterstützen¹⁴. Insbesondere in der agilen Softwareentwicklung sind Anforderungen häufig in Form von *User Stories*¹⁵ vorzufinden, welche ebenfalls Vorgaben bezüglich der Satzstruktur folgen. Das Problem einer Repräsentation in textueller Form ist jedoch die geringe Lesbarkeit für Systeme aufgrund einer fehlenden formalisierten Syntax und Semantik. Zuverlässige Ähnlichkeitsanalysen auf Basis natürlicher Sprache existieren derzeit nur für Englisch¹⁶ oder Koreanisch¹⁷, jedoch nicht für Deutsch. Abgesehen von umfangreichen Text-Mining-Systemen, die im Umfeld der Anforderungsanalyse bisher jedoch kaum Anwendung finden, gibt es somit derzeit keine Möglichkeit, Inhalte von Anforderungen automatisiert zu verstehen und entsprechende Ähnlichkeitsbestimmungen darauf auszuführen. Insbesondere sind dem Konsortium keine Forschungsvorhaben bekannt, die sich mit der Anforderungsähnlichkeit im Hinblick auf die im agilen Umfeld häufig vorzufindende Beschreibungsformen wie *User Stories* beschäftigen. Ferner ist in bestehenden Verfahren häufig unklar, inwiefern sich diese Ansätze auf die Fragestellungen dieses Projekts hinsichtlich Aufwandsschätzung anpassen lassen.

Modellbasierten Repräsentationen liegt eine semi-formale Syntax zugrunde, um bestimmte Eigenschaften einer Software darzustellen zu können. Grundsätzlich bieten diese Repräsentationen ganz andere Möglichkeiten, um erfolgreich Ähnlichkeitsanalysen durchführen zu können. So geben beispielsweise Use-Case-Diagramme einen syntaktisch eindeutigen Überblick, welche Funktionen dem Nutzer des Systems zur Verfügung stehen und mit welchen externen Partnern Interaktionen möglich sind¹⁸. Für die Software relevante Daten können unter anderem mittels Entity-Relationship-Modellen¹⁹ oder UML-Klassendiagrammen charakterisiert werden. Zur Repräsentation von Verhalten in geschäftsprozessgetriebenen Systemen eignen sich UML-Aktivitätsdiagramme oder andere Prozessnotationen wie BPMN²⁰.

Obwohl in der Praxis oftmals **Mischformen** aus modellbasierten und textuellen Repräsentationen von Anforderungen eingesetzt werden, dominieren nach wie vor textuelle Repräsentationsformen. Ein für KMU tauglicher Ansatz zur Ähnlichkeitsbestimmung muss diesem Sachverhalt Rechnung tragen und erfordert daher eine leichtgewichtige syntaktische Erweiterung von textuellen Anforderungen, um die Struktur und Inhalte solcher Anforderungen besser analysieren und vergleichen zu können. Dies kann beispielsweise durch Attributierung und Tagging von Anforderungen mittels semantischer Zusatzinformationen erfolgen.

1.3 Entwicklung der Abakus-Methode

In einer frühen Phase des Abakus-Projekts wurden bei den beteiligten KMU Analysen der Methoden zur Anforderungsermittlung und -dokumentation sowie der verwendeten Schätzmethode durchgeführt, um daraus Anforderungen an die zu entwickelnde Methode bzw. die angestrebte Werkzeugunterstützung abzuleiten. Dazu wurden existierende Abläufe realer Prozesse bei den Anwendungspartnern aufgenommen und analysiert, um ein detailliertes Verständnis der Vorgänge, Rollen und Werkzeuge gewinnen zu können. Die Analysen wurden von den beiden Forschungspartnern des Abakus-Projekts unter Hauptverantwortung des Fraunhofer IESE durchgeführt. Die Anwendungspartner (KMU) berücksichtigten und beschrieben jeweils ihre relevanten Prozesse, Werkzeuge und Datenquellen.

1.3.1 Requirements-Engineering-Praxis der an Abakus beteiligten KMU

Im Fokus der ersten Untersuchung standen die eingesetzten Methoden zur Anforderungsermittlung und -dokumentation. Konkret wurde im Rahmen der Analyse des Ist-Zustands der Requirements-Engineering(RE)-Praxis untersucht,

- welche Vorgehensweisen die KMU im RE anwenden,
- welche Rollen hierbei involviert sind,
- welche Artefakte die KMU im RE erstellen,
- welche Werkzeuge hierbei verwendet werden,
- wie die RE-Verantwortlichen die Verhältnismäßigkeit des RE einschätzen (z. B. im Hinblick auf eingesetzten Aufwand) und
- welche Herausforderungen sie in Bezug auf das RE wahrnehmen.

An der Umfrage zur Analyse beteiligten sich insgesamt 14 Teilnehmer aus den fünf in Abakus beteiligten KMU. Die meisten dieser Teilnehmer haben Positionen in der Geschäftsführung bzw. im Projektmanagement inne, mit einer Erfahrungsspanne im Bereich RE von meist 6 bis 15 Jahren.

Grundlagen

86 % der Teilnehmer gaben an, dass in ihrem Unternehmen RE-Aktivitäten stattfinden, primär zur Entwicklung spezifischer Lösungen im Rahmen individueller Kundenaufträge. Dabei entsteht in den meisten Fällen eine Grobspezifikation für das zu entwickelnde Gesamtsystem, die bei Bedarf detailliert wird. Häufig beginnt die Entwicklung/Beschaffung jedoch bereits vor Abschluss der Spezifikation, so dass Ergebnisse und Erkenntnisse aus der Entwicklung in die Spezifikation einfließen können. 85 % der Teilnehmer gaben an, dass in ihren Unternehmen kein dediziertes Budget bzw. kein Team für die oben genannten RE-Aktivitäten zur Verfügung steht.

Vorgehen

62 % der Teilnehmer gaben an, dass in Softwareentwicklungsprojekten immer die gleichen RE-Prozesse bzw. RE-Aktivitäten durchgeführt werden. Fast alle Teilnehmer (92 %) gaben an, dass die Anforderungserhebung in ihrem Unternehmen als paralleler/durchlaufender Prozess stattfindet. Dabei werden überwiegend Techniken eingesetzt, welche eine aktive Einbindung der Stakeholder ermöglichen, wie persönliche

Befragungen der Stakeholder, Selbstaufschreibung durch die Stakeholder, On-the-fly-Ermittlung sowie aktive Einbindung der Stakeholder in Entwurfsentscheidungen. Der Auftraggeber gilt als wichtigste Quelle für die Anforderungserhebung. Darüber hinaus werden häufig Entwickler/Architekten, Fachexperten sowie Endbenutzer aktiv in die Anforderungserhebung involviert. Bei den RE-Praktiken, die in den Unternehmen besonders gut angewendet werden können, wurden die Erhebung funktionaler Anforderungen sowie die Überprüfung der Machbarkeit am positivsten eingeschätzt. Verbesserungsbedarf besteht etwa bei der Erhebung nichtfunktionaler Anforderungen, bei der systematischen Verwaltung, dem Sicherstellen der Erfolgbarkeit sowie der Priorisierung von Anforderungen.

Personen

Die Anzahl der Personen, die typischerweise in Projekte involviert sind, liegt bei den befragten Unternehmen bei maximal 10 Personen. In den meisten Fällen besteht das RE-Team aus 2–3 Personen, aber auch Teams aus 4–6 Personen sind nicht selten. Als häufigste Quelle für den Erwerb von RE-Wissen gelten autodidaktische Aneignung (beispielsweise mit Büchern oder Zeitschriften). Coachingmaßnahmen, Schulungen, Vorlesungen oder Konferenzbesuche werden gar nicht bzw. nur im Falle weniger Personen als Wissensquelle genutzt. Während eines Projektes arbeiten die Personen, die RE in den befragten Unternehmen durchführen, vorwiegend an sonstigen Aufgaben und arbeiten maximal 40 % ihrer Arbeitszeit im RE. In den meisten Fällen fallen auf RE-Aufgaben lediglich 20 % oder weniger der Arbeitszeit.

Artefakte

Als häufigste Dokumentationstechniken werden in den befragten Unternehmen Prototypen bzw. GUI-Mock-Ups eingesetzt. Darüber hinaus finden natürliche Sprache (mit und ohne Vorlagen), Use Cases und User Stories zur Spezifikation von Anforderungen häufigen Einsatz. Systemmodelle, Prozessmodelle oder auch formale Modelle werden dagegen nie oder nur selten als Dokumentationstechnik verwendet. Etwas weniger als die Hälfte der TN (45 %) gab an, dass für die erfassten Anforderungen Akzeptanzkriterien definiert werden, die etwa in manuellen wie auch (teil-)automatisierten Tests oder auch in Akzeptanztests überprüft werden.

Werkzeuge

Als Werkzeug für die Unterstützung von RE-Aktivitäten wird in den befragten Unternehmen vorwiegend Office-Software eingesetzt. Darüber hinaus finden Werkzeuge zum Dokumentenmanagement, Issue-Tracker, Notizprogramme sowie Prototyping-/Mockup-Tools häufigen Einsatz. Eher mächtige Werkzeuge wie Requirements-Management(RM)-Tools, Modellierungswerkzeuge oder Application Lifecycle Management Tools werden gar nicht oder nur sehr selten verwendet. Zu teure Lizenzen, inadäquate Unterstützung der RE-Aktivitäten sowie schlechte Integration mit anderen Werkzeugen gelten als die größten Herausforderungen im Hinblick auf RE-unterstützende Werkzeuge.

Verhältnismäßigkeit

Typischerweise haben Projekte einen Gesamtumfang von 1–3 Personenmonaten. Dabei wird in den meisten Fällen weniger als 5 % des Gesamtaufwands für RE-Aufgaben verwendet. Für etwas mehr als die Hälfte der TN ist der prozentuale Aufwand für RE-Aufgaben genau richtig bzw. angemessen, für die anderen TN wird der prozentuale Aufwand als eher zu gering eingeschätzt. Der Umfang eines Anforderungsdokumentes liegt typischerweise bei ca. 11–20 Seiten, manchmal auch etwas weniger (5–10 Seiten). Dieser

Umfang wird von den meisten Teilnehmern als genau richtig bzw. angemessen eingeschätzt. Die übrigen Teilnehmer schätzen den Umfang dagegen als zu gering ein, was als Folge in ungenügender Qualität (z. B. unpräzise/mehrdeutige Anforderungen) resultiere. Auch führe der geringe Umfang dazu, dass häufig Hintergrundinformationen bzw. fachliche Zusatzinformationen fehlen, um das Verständnis zu erhöhen. Zudem halte sich (fast) niemand an das Dokument.

Herausforderungen

Zu den größten Herausforderungen, die in den befragten Unternehmen generell vorherrschen, gehören die Handhabung von Anforderungsänderungen, die Nachverfolgbarkeit von Anforderungen, die Wiederverwendung/Standardisierung von Anforderungen, die Ermittlung/Kommunikation von Anforderungen sowie die Motivation für die Einführung und Einhaltung von RE. Während der Anforderungsermittlung und -dokumentation liegen die größten Herausforderungen darin, zu ermitteln, welcher Detailgrad der Dokumentation angebracht ist bzw. wie man Wünsche von relevanten Anforderungen abgrenzen kann. Des Weiteren herrscht bei einigen Teilnehmern Unklarheit darüber, wie man Anforderungen nicht verfügbarer Stakeholder ermitteln kann bzw. was man erheben und beschreiben muss. In Bezug auf Anforderungsdokumente gelten Mehrdeutigkeit der Anforderungen, fehlende Gewichtung sowie schlechte Verfolgbarkeit und Unvollständigkeit als wesentliche Herausforderungen.

RE-Prozess

Aus der Analyse der RE-Aktivitäten konnte folgender grober Prozess abgeleitet werden: Anforderungen werden zunächst erhoben, typischerweise zusammen mit dem Kunden mittels verschiedener Techniken (z. B. persönliche Befragungen der Stakeholder). Nachdem die Anforderungen gesammelt wurden, werden diese dokumentiert und anschließend mit dem Kunden abgestimmt und bei Bedarf aktualisiert. Bei der Dokumentation entstehen verschiedene Artefakte, z. B. Spezifikationen mit Use Cases oder User Stories, Prototypen u. ä. Neben der Dokumentation der Anforderungen erfolgt in einigen Unternehmen eine Prüfung der Anforderungen etwa im Hinblick auf Wiederverwendung oder Aufwandsschätzung. Basierend auf diesen Prüfungen werden detaillierte Angebote abgeleitet, die auch als Abarbeitungsleitfaden dienen. Schließlich werden die Anforderungen umgesetzt.

1.3.2 Schätzpraxis der an Abakus beteiligten KMU

Im Fokus der zweiten Untersuchung stand die bisherige Schätzpraxis bei den am Projekt beteiligten KMU. Es wurde eine Studie bestehend aus einem Fragebogen und Interviews entworfen, mit dem Ziel weiterführende Erkenntnisse über die aktuell verwendeten Schätzprozesse zu gewinnen. Die bereitgestellten Informationen umfassten neben demographischen Daten zu den Unternehmen Informationen zum jeweiligen Anwendungskontext, den derzeit im Einsatz befindlichen Aufwandsschätzmethoden, den zur Schätzung genutzten Daten und Werkzeugen, den involvierten Rollen sowie den Erfahrungen mit der derzeitigen Schätzpraxis. Zusätzlich wurden auch Erwartungen an die Abakus-Methodik und die entsprechende Werkzeugumgebung dokumentiert.

Ziel der Studie war es, die existierenden Schätzprozesse der vier KMU-Partner im Kontext von Softwareentwicklungsprojekten zu untersuchen und zu beurteilen in Bezug auf dessen

- Operationalisierung (Wie sieht der Schätzprozess genau aus? Schritte, Inputs, Outputs, Eingaben, Akteure/Beteiligte),
- Effektivität (Wie gut sind die Schätzungen?) und
- Effizienz (Wieviel kostet die Schätzung?).

Die gewonnenen Erkenntnisse sollten die Grundlage für die weiteren Arbeiten im Projekt bilden und den Rahmen für die Entwicklung der Abakus-Methodik und der geplanten Werkzeugunterstützung vorgeben. Dazu gehörten:

- Charakterisierung des Stands der Praxis von angewendeten Methoden zur Aufwandabschätzung,
- Identifikation der Vor- und Nachteile angewendeter Methoden zur Aufwandabschätzung,
- Erhebung von Anforderungen an Methoden zur Aufwandschätzung und
- Erhebung der aktuellen Datenlage für die Entwicklung der Abakus-Schätzmethodik.

Die Abakus-Schätzmethode und deren Werkzeugkomponenten sollten in verschiedenen Projektsettings einsetzbar sein und zu einer Schätzung bereits in einer frühen Phase des Projekts beitragen – in der jeweiligen benötigten Abstraktionsebene. Zusätzlich sollte eine effiziente Möglichkeit bereitgestellt werden, die Schätzergebnisse in Kundenangeboten zu verwenden.

Die Ergebnisse der Studie geben einen Überblick über den Stand der Praxis bezüglich der Aufwandsschätzung in KMU. Aufgrund der geringen Fallzahl lassen sich zwar keine allgemeingültigen Aussagen für sämtliche deutschen KMU der IKT-Branche ableiten, jedoch lassen sich relevante Rahmenbedingungen und Problemstellungen in diesem Umfeld erschließen. Hierzu wurden anwendungspartnerübergreifende Gemeinsamkeiten, aber auch Unterschiede im Spannungsfeld der Aufwandsschätzung identifiziert. Die Ergebnisse zeigen auf, dass sehr heterogene Projektumgebungen zu berücksichtigen sind, von agiler Entwicklung bis zum klassischen Wasserfall-Ansatz. Aufwandsabschätzungen werden bei allen Partnern im frühen Projektverlauf getroffen. Dennoch variieren der Zweck, der Schätzprozess und die Granularität der Schätzungen zwischen den einzelnen KMU, was entweder zu Schätzungen des Aufwands (in Personentagen) oder der Komplexität (in Story Points) führt. Die wichtigste Gemeinsamkeit aller KMU ist der Einsatz von rein auf Experteneinschätzungen beruhenden Aufwandsschätzverfahren, die jeweils bottom-up durchgeführt werden. Die Verwendung der Schätzergebnisse hingegen ist stark vom Entwicklungskontext, insbesondere dem Geschäftstyp (Produktentwicklung gegenüber Projektgeschäft), aber auch von den konkreten Entwicklungsprozessen sowie der Art und Granularität der Artefakte abhängig.

Zielstellung bei der Entwicklung der Abakus-Methodik und der Werkzeugunterstützung war es daher, einerseits ausreichend flexibel sein, um in den unterschiedlichen Anwendungsumgebungen einsetzbar zu sein, andererseits aber hinreichend konkret und verständlich sein, um ohne große Lernhürde auch in einem von hohem Projektdruck getriebenen Umfeld angewendet werden zu können.

2 Die Abakus-Methode

2.1 Phasen/Prozesse

Die Abakus-Methode basiert auf Artefakten und kontextrelevanten Informationen aus abgeschlossenen Projekten, die in einer zentralen Wissensbasis (s. Abbildung 1: Experience Management System, kurz: EMS)² abgelegt werden und als Grundlage für die Kalkulation zukünftiger Projekte dienen. Die Anforderungen neuer Projekte liegen – z. B. in der Praxis der beteiligten KMU – oft als Lastenheft oder auch als vages Briefing-Dokument vor. Diese oft nur rudimentären Anforderungen werden im Abakus-Werkzeug zunächst aufbereitet und dann zu einer vergleichenden Aufwandsschätzung genutzt. Das Ergebnis stellt eine erste strukturierte, präzise Schätzung dar, die dann zu einem Angebot oder einer internen Projektkalkulation aufbereitet und erweitert werden kann.



Abbildung 1: Lösungsskizze der Abakus-Methode

2.1.1 Anforderungsanalyse

Die Anforderungsanalyse erfolgt auf Basis der Anforderungen, die die Kunden zu einem Projekt liefern. Um die darin enthaltenen Aufwands- oder Komplexitätstreiber analysieren zu können, nutzt das Abakus-Werkzeug einen Adapter, der Kundenanforderungen aus Drittsystemen automatisch in eine standardisierte Form transformiert, die einen Vergleich der aktuell erfassten Anforderung mit den Inhalten der Wissensbasis ermöglicht. Im Gegensatz zu klassischen algorithmischen Verfahren erfolgen der Abgleich und die Aufwandsabschätzung hier nicht ausschließlich auf der Ebene des Gesamtprojekts, sondern im Bezug zu den

² Eine entsprechende Werkzeugumgebung wurde im Abakus-Projekt implementiert. Die Darstellung der (softwaregestützten) Abakus-Methode in den folgenden Abschnitten setzt an vielen Stellen ein entsprechendes Tool voraus, beispielsweise müssen Anforderungen in einer maschinenlesbaren Form in einem System hinterlegt sein, um eine automatisierte Ähnlichkeitsbestimmung von Anforderungen zur Nutzung im Rahmen von Schätzverfahren durchführen zu können. Die Darstellung erfolgt aber abstrahiert von dieser exemplarischen Implementierung. In Kapitel 4 werden diverse Schätzpraktiken empfohlen, die auch ohne Werkzeugunterstützung angewendet werden können.

einzelnen Anforderungen und dem Aufwand zur Umsetzung vergleichbarer Anforderungen in der Wissensbasis.

Um vergleichbare Anforderungen zu identifizieren enthält die Wissensbasis strukturierte Daten über bereits abgeschlossene und kalkulierte Projekte sowie Kontextinformationen und unterstützt den Nutzer beim Abgleich der Anforderungen. Die gespeicherten Daten beinhalten insbesondere funktionale sowie nichtfunktionale Anforderungen, deren Umsetzungskosten sowie weitere Artefakte aus abgewickelten Projekten und deren Zusammenhänge. Über eine Rückverfolgbarkeit kann abgefragt werden, woher die Daten kommen und wie sie entstanden sind. Des Weiteren werden Informationen über Größe, Komplexität sowie Umweltinformationen und Lösungskonzepte in den zugrundeliegenden Altprojekten hinterlegt.

2.1.2 Aufwandsschätzung

Die eigentliche Aufwandsschätzung erfolgt in der Abakus-Methode in den zwei Schritten: **Retrieve** und **Reuse**. Im Retrieve-Schritt werden die aktuellen Projektanforderungen mit den Inhalten der Wissensbasis (also Anforderungen und Kontextinformationen aus abgeschlossenen Projekten) abgeglichen. Dabei werden ähnliche und identische Anforderungen bzw. Kontextfaktoren aus der Wissensbasis ermittelt und für das zu kalkulierende Projekt aufbereitet. Ist diese Ähnlichkeitsanalyse erfolgreich, werden die ermittelten kostentreibenden Parameter in den Reuse-Schritt übernommen. Ziel ist es, mit Hilfe der in der Wissensbasis gespeicherten Daten aus abgeschlossenen Projekten und projektspezifischen Charakteristika eine Schätzung durchführen zu können. Die Präzision der Schätzung wird durch die Bestimmung der Ähnlichkeitsgrade zwischen neuen und historischen Projektdaten verbessert. Diese Ähnlichkeitsabschätzungen können zusätzlich als Grundlage für die Kalkulation des Projektrisikos verwendet werden. Beispiele für projektspezifische Charakteristika können technologische Faktoren, z. B. Neuheitsgrad von Anwendungen oder das angestrebte Sicherheitsniveau der Software, oder auch organisatorische Faktoren, z. B. die Größe bzw. die Internationalität des Projektteams sein. Durch diese Art der Aufwandsschätzung ermöglicht die Abakus-Methode eine präzise Schätzung ohne den Systemumfang und die Komplexität der zu entwickelnden Software vollständig durchdringen zu müssen. Weiterhin werden für die Kalkulation des Projektaufwandes möglichst keine abstrakten Größen wie Function Points oder Lines of Code gewählt, sondern Personentage, so wie sie auch im späteren Angebot üblicherweise aufgeführt werden.

2.1.3 Ergebnisaufbereitung

In der Ergebnisaufbereitung wird eine ansprechende, überzeugende und nachvollziehbare Repräsentation der Kalkulation mit Drill-Down-Funktionalität und Begründung der Preise auf Ebene einzelner funktionaler und nichtfunktionaler Anforderungen erzeugt. Der Nutzer der Methode hat an dieser Stelle die Möglichkeit, die Kalkulation zu korrigieren. Ergebnis der Aufbereitung ist eine präzise und nachvollziehbare Kalkulation als Basis für ein Angebot und für die Projektplanung.

2.1.4 Kalibrierung

Nach Abschluss des Projektes wird der kalkulierte Aufwand für die Umsetzung einzelner Anforderungen dem tatsächlich benötigten Aufwand gegenübergestellt und auf Differenzen analysiert. Die Abweichungen werden in der Wissensbasis nachkalibriert, um die Kalkulation sukzessive zu verbessern.

2.2 Rollen

Im Folgenden sind alle Rollen beschrieben, die bei der Abakus-Methode relevant sind und in diesem Dokument genannt werden. Dies sind die Anwender der Methode, insbesondere die Nutzer der Wissensbasis, aber auch weitere Stakeholder, die ein Interesse an den in Kapitel 2.1 beschriebenen Prozessen haben.

Rolle	Kunde
typische Aufgaben	<ul style="list-style-type: none"> eigene Wünsche und Anforderungen äußern bzw. dokumentieren bei der Anforderungsermittlung mitwirken erstellte Angebote prüfen und gegebenenfalls beauftragen
Kenntnisse/Voraussetzungen	<ul style="list-style-type: none"> Kenntnis der eigenen Fachdomäne Kenntnis von Einflussfaktoren in der eigenen Branche, z. B. Standards, gesetzliche Anforderungen u. ä.
Zusammenarbeit mit ...	<ul style="list-style-type: none"> Projekt-/Prozessverantwortlicher Anforderungsermittler Vertriebsmitarbeiter Entwickler

Rolle	Projekt-/Prozessverantwortlicher
typische Aufgaben	<ul style="list-style-type: none"> Gesamtprozess der Aufwands- und Kostenschätzung koordinieren neue Kunden und Projekte anlegen neue Kontext-Tags anlegen neue Stundensätze anlegen
Kenntnisse/Voraussetzungen	<ul style="list-style-type: none"> Prozesswissen (Aufwands- und Kostenschätzung) Kommunikationsfähigkeit Koordinationsfähigkeit
Zusammenarbeit mit ...	<ul style="list-style-type: none"> Kunde Anforderungsermittler Schätzer Vertriebsmitarbeiter Entwickler

Rolle	Anforderungsermittler
typische Aufgaben	<ul style="list-style-type: none"> • Kundenwünsche³ und Anforderungen erheben und mit den Kunden abstimmen • Kundenwünsche und Anforderungen als Input für die Aufwands- und Kostenschätzung bereitstellen
Kenntnisse/Voraussetzungen	<ul style="list-style-type: none"> • Techniken der Anforderungsermittlung und -dokumentation
Zusammenarbeit mit ...	<ul style="list-style-type: none"> • Kunde • Projekt-/Prozessverantwortlicher • Schätzer • Entwickler

Rolle	Schätzer
typische Aufgaben	<ul style="list-style-type: none"> • passende Features, Lösungsvarianten und Funktionen ermitteln • historische Daten mit aktuellen Kundenwünschen und Anforderungen vergleichen • Kontext-Tags anlegen • Aufwände und Kosten schätzen
Kenntnisse/Voraussetzungen	<ul style="list-style-type: none"> • Kenntnis der verwendeten Technologien
Zusammenarbeit mit ...	<ul style="list-style-type: none"> • Projekt-/Prozessverantwortlicher • Anforderungsermittler

³ Nach dem Verständnis des Abakus-Projekts ist ein Kundenwunsch eine einfache, aber möglichst konkrete Aussage darüber, was der Kunde von der Software verlangt, beispielsweise „Die Unternehmenswebsite soll es ermöglichen, mit dem Unternehmen in Kontakt zu treten.“

Rolle	Vertriebsmitarbeiter
typische Aufgaben	<ul style="list-style-type: none"> • Angebote erstellen • Angebote mit den Kunden verhandeln
Kenntnisse/Voraussetzungen	<ul style="list-style-type: none"> • Kenntnis der ermittelten Kundenwünsche und Anforderungen • Verhandlungsgeschick
Zusammenarbeit mit ...	<ul style="list-style-type: none"> • Projekt-/Prozessverantwortlicher • Kunde

Rolle	Entwickler
typische Aufgaben	<ul style="list-style-type: none"> • beauftragte Features, Lösungsvarianten und Funktionen umsetzen
Kenntnisse/Voraussetzungen	<ul style="list-style-type: none"> • Kenntnis der verwendeten Technologien • Kenntnis der ermittelten Kundenwünsche und Anforderungen
Zusammenarbeit mit ...	<ul style="list-style-type: none"> • Projekt-/Prozessverantwortlicher • Anforderungsermittler • Kunde

2.3 Aktivitäten

In den folgenden Übersichten werden die einzelnen Aktivitäten der Abakus-Methode im Detail erläutert. In der Regel handelt es sich um Aktivitäten, die systemgestützt stattfinden, z. B. mit einer entsprechenden Erfahrungsdatenbank. Im Rahmen des Abakus wurde aber auch eine Reihe von Prinzipien bzw. Praxistipps formuliert (siehe Kapitel 4), die tool- und technologieunabhängig befolgt werden können.

Pro Aktivität wird kurz das Ziel beschrieben, die ausführende(n) Rolle(n) genannt und aufgeführt, welcher Input benötigt wird bzw. welche Voraussetzungen für die jeweilige Aktivität erfüllt sein müssen. Neben dem Standardablauf werden bei Bedarf Varianten oder Optionen beschrieben, die beim Ablauf in Frage kommen, es wird das Ergebnis bzw. der Output der Aktivität aufgeführt und bei Bedarf werden erklärungsbedürftige Punkte zu dieser Aktivität erläutert. Bei der Beschreibung wird davon ausgegangen, dass diese Aktivitäten toolgestützt stattfinden. Die Beschreibung erfolgt allerdings abstrahiert von bestehenden Technologien oder einer konkreten Toolumsetzung.

Aktivität	Kunde anlegen
Ziel	Im System soll ein neuer Kunde angelegt werden, damit ein Projekt für diesen Kunden geschätzt werden kann.
Rolle(n)	Projekt-/Prozessverantwortlicher
Voraussetzungen/Input	Der Anwender ist am System angemeldet und hat die Kundenliste geöffnet.
Ablauf	Der Anwender legt einen neuen Kunden im System an (mit Name und Kurzbeschreibung des Kunden).
Varianten/Optionen	–
Ergebnis/Output	Der Kunde ist angelegt und erscheint in der Kundenliste.
Erläuterungen	Neben der Funktion zum Anlegen von Kunden sollte es stets auch eine Möglichkeit geben, bereits angelegte Kunden einzusehen, zu bearbeiten und zu löschen.

Aktivität	Kostensatz anlegen
Ziel	Im System soll ein neuer Kostensatz hinterlegt werden, der für künftige Schätzungen benötigt wird.
Rolle(n)	Projekt-/Prozessverantwortlicher

Voraussetzungen/Input	Der Anwender ist am System angemeldet und hat die Liste der verfügbaren Kostensätze geöffnet.
Ablauf	Der Anwender legt einen neuen Kostensatz im System an (mit Bezeichnung, Einheit, Dimension [z. B. Stunde, Stück] und Betrag).
Varianten/Optionen	–
Ergebnis/Output	Der Kostensatz ist angelegt und erscheint in der Liste der verfügbaren Kostensätze.
Erläuterungen	Neben der Funktion zum Anlegen von Kostensätzen sollte es stets auch eine Möglichkeit geben, bereits angelegte Kostensätze einzusehen, zu bearbeiten und zu löschen.

Aktivität	Projekt anlegen
Ziel	Im System soll ein neues Projekt angelegt werden, damit dieses geschätzt werden kann.
Rolle(n)	Projekt-/Prozessverantwortlicher
Voraussetzungen/Input	Der Anwender ist am System angemeldet. Der Kunde, dessen Projekt angelegt werden soll, ist im System hinterlegt.
Ablauf	Der Anwender legt ein neues Projekt an (mit Projektname, Projektleiter, Projektteam, Projekttyp und Kunde).
Varianten/Optionen	Optional verknüpft der Anwender Kontext-Tags mit dem Projekt (vgl. Aktivität „Projektbezogenes Kontext-Tag anlegen“).
Ergebnis/Output	Das Projekt ist angelegt und erscheint in der Projektliste.
Erläuterungen	Neben der Funktion zum Anlegen von Projekten sollte es stets auch eine Möglichkeit geben, bereits angelegte Projekte einzusehen, zu bearbeiten und zu löschen.

Aktivität	Projektbezogenes Kontext-Tag anlegen
Ziel	Ein konkretes Projekt soll mit einem bestimmten Kontext-Tag versehen werden.
Rolle(n)	Projekt-/Prozessverantwortlicher
Voraussetzungen/Input	Der Anwender ist am System angemeldet. Das betreffende Projekt ist im System angelegt.
Ablauf	Der Anwender legt ein neues Kontext-Tag an (mit Name).
Varianten/Optionen	Optional gibt der Anwender einen Wert für das Kontext-Tag ein (z. B. einen Rabatt oder einen Risikozuschlag).
Ergebnis/Output	Das Kontext-Tag ist angelegt und mit dem Projekt verknüpft. Das Kontext-Tag kann auch bei anderen Projekten und Varianten verwendet werden (steht in der Auswahlliste zur Verfügung).
Erläuterungen	Neben der Funktion zum Anlegen von projektbezogenen Kontext-Tag sollte es stets auch eine Möglichkeit geben, bereits angelegte Kontext-Tags einzusehen, zu bearbeiten und zu löschen. Je mehr Kontextfaktoren (Kontext-Tags) zwischen dem aktuellen Projekt und historischen Projekten übereinstimmen, desto höher ist die Ähnlichkeit der Projekte.

Aktivität	Kundenwünsche anlegen
Ziel	Die Kundenwünsche für ein konkretes Projekt sollen manuell angelegt werden.
Rolle(n)	Anforderungsermittler, Schätzer
Voraussetzungen/Input	Der Anwender ist am System angemeldet. Das betreffende Projekt ist im System angelegt.
Ablauf	Der Anwender wählt das betreffende Projekt aus und legt einen neuen Kundenwunsch manuell an.

	Er wiederholt diesen Vorgang so lange, bis alle Kundenwünsche im System erfasst sind.
Varianten/Optionen	–
Ergebnis/Output	Der Kundenwunsch ist angelegt und erscheint in der Liste der Kundenwünsche für dieses Projekt.
Erläuterungen	Neben der Funktion zum Anlegen von Kundenwünschen sollte es stets auch eine Möglichkeit geben, bereits angelegte Kundenwünsche einzusehen, zu bearbeiten und zu löschen.

Aktivität	Kundenwünsche einlesen
Ziel	Die Kundenwünsche für ein konkretes Projekt sollen semiautomatisch eingelesen werden.
Rolle(n)	Anforderungsermittler, Schätzer
Voraussetzungen/Input	Der Anwender ist am System angemeldet. Das betreffende Projekt ist im System angelegt.
Ablauf	Der Anwender wählt das betreffende Projekt aus und startet das Tool zum semiautomatischen Einlesen von Kundenwünschen. Er wählt eine Datei (z. B. MS Word oder Excel) von einer lokalen Festplatte aus und öffnet diese. Das Tool zeigt ihm die eingelesenen Inhalte an. Der Anwender wählt die gewünschten Inhalte/Kundenwünsche aus und übernimmt diese in das System.
Varianten/Optionen	Bei Bedarf kann der Anwender die eingelesenen Inhalte vor dem Übernehmen bearbeiten.
Ergebnis/Output	Die ausgewählten/bearbeiteten Kundenwünsche sind im System gespeichert und erscheinen in der Liste der Kundenwünsche für dieses Projekt.
Erläuterungen	Neben der Funktion zum Einlesen von Kundenwünschen sollte es stets auch eine Möglichkeit geben, bereits eingelesene Kundenwünsche einzusehen, zu bearbeiten und zu löschen.

Aktivität	Feature anlegen
Ziel	Im System soll ein neues Feature ⁴ angelegt werden, das in den bereits erfassten Projekten noch nicht existiert.
Rolle(n)	Schätzer
Voraussetzungen/Input	Der Anwender ist am System angemeldet und hat die Liste der verfügbaren Features geöffnet.
Ablauf	Der Anwender legt ein neues Feature im System an (mit Name und Kurzbeschreibung des Features).
Varianten/Optionen	–
Ergebnis/Output	Das Feature ist angelegt und erscheint in der Liste der verfügbaren Features.
Erläuterungen	Neben der Funktion zum Anlegen von Features sollte es stets auch eine Möglichkeit geben, bereits angelegte Features einzusehen, zu bearbeiten und zu löschen.

Aktivität	Variante auswählen
Ziel	Für ein konkretes Projekt soll eine existierende Variante eines bestimmten Features ausgewählt werden.
Rolle(n)	Schätzer
Voraussetzungen/Input	Der Anwender ist am System angemeldet. Das zu schätzende Projekt mitsamt Kundenwünschen ist im System angelegt. Der Anwender hat das passende Feature aus der Liste der verfügbaren Features ausgewählt.

⁴ Nach dem Verständnis des Abakus-Projekts ist ein Feature eine Sammlung zusammenhängender Funktionalitäten in einer potentiell wiederverwendbaren Softwarekomponente. Eine vom Kunden geforderte Möglichkeit, mit dem Unternehmen in Kontakt zu treten, kann beispielsweise über Features wie „Kontaktformular“, „Chat“, „Forum“ oder „Mailto-Link“ realisiert werden.

Ablauf	Der Anwender wählt aus der Liste der verfügbaren Varianten die passende Variante für das aktuelle Projekt aus.
Varianten/Optionen	–
Ergebnis/Output	Die Variante ist ausgewählt und kann bei Bedarf weiterbearbeitet werden.
Erläuterungen	Neben der Funktion zum Auswählen von Varianten sollte es stets auch eine Möglichkeit geben, eine bereits getroffene Auswahl zu ändern bzw. rückgängig zu machen.

Aktivität	Variante anlegen
Ziel	Für ein bestimmtes Feature soll eine neue Variante angelegt werden, die in den bereits erfassten Projekten noch nicht existiert.
Rolle(n)	Schätzer
Voraussetzungen/Input	Der Anwender ist am System angemeldet. Das zu schätzende Projekt mitsamt Kundenwünschen ist im System angelegt. Der Anwender hat das passende Feature aus der Liste der verfügbaren Features ausgewählt.
Ablauf	Der Anwender legt eine neue Variante für das Feature im System an (mit Name und Kurzbeschreibung der Variante).
Varianten/Optionen	Optional verknüpft der Anwender Kontext-Tags mit der Variante (vgl. Aktivität „Variantenbezogenes Kontext-Tag anlegen“).
Ergebnis/Output	Die Variante ist angelegt und kann bei Bedarf weiterbearbeitet werden. Die Variante wird künftig in der Liste der verfügbaren Varianten dieses Features angezeigt.
Erläuterungen	Neben der Funktion zum Anlegen von Varianten sollte es stets auch eine Möglichkeit geben, bereits angelegte Varianten einzusehen, zu bearbeiten und zu löschen.

Aktivität	Funktion auswählen
Ziel	Für eine konkrete Variante eines existierenden Features soll eine bestimmte Funktion ausgewählt werden.
Rolle(n)	Schätzer
Voraussetzungen/Input	Der Anwender ist am System angemeldet. Der Anwender hat das passende Feature geöffnet und aus der Liste der verfügbaren Varianten die passende Variante ausgewählt.
Ablauf	Der Anwender wählt aus der Liste der verfügbaren Funktionen die passende Funktion für die aktuelle Variante aus. Er verknüpft diese Funktion mit der Variante und gibt bei Bedarf eine Kurzbeschreibung für die Funktion ein. Er wiederholt diesen Vorgang so lange, bis alle benötigten Funktionen ausgewählt sind.
Varianten/Optionen	Optional kann der Anwender zu jeder ausgewählten Funktion eine oder mehrere Dateien hochladen.
Ergebnis/Output	Die ausgewählte Funktion ist mit der aktuellen Variante des Features verknüpft.
Erläuterungen	Neben der Funktion zum Auswählen von Funktionen sollte es stets auch eine Möglichkeit geben, eine bereits getroffene Auswahl zu ändern bzw. rückgängig zu machen.

Aktivität	Funktion anlegen
Ziel	Für ein existierendes Feature soll eine neue Funktion angelegt werden, die in den bereits erfassten Varianten noch nicht existiert.
Rolle(n)	Schätzer
Voraussetzungen/Input	Der Anwender ist am System angemeldet und hat das passende Feature geöffnet.

Ablauf	Der Anwender legt eine neue Funktion für das Feature im System an (mit Name und bei Bedarf mit Kurzbeschreibung der Funktion).
Varianten/Optionen	Optional kann der Anwender zu der Funktion eine oder mehrere Dateien hochladen.
Ergebnis/Output	Die Funktion ist angelegt und erscheint bei dem betreffenden Feature in der Liste der verfügbaren Funktionen. Die Funktion kann nun mit einer Variante verknüpft werden.
Erläuterungen	–

Aktivität	Tätigkeit anlegen
Ziel	Im System soll eine neue Tätigkeit angelegt werden, die für die Realisierung einer bestimmten Funktion benötigt wird.
Rolle(n)	Schätzer
Voraussetzungen/Input	Der Anwender ist am System angemeldet und hat die passende Variante eines bestimmten Features geöffnet.
Ablauf	Der Anwender wählt aus der Liste der verfügbaren Funktionen eine bestimmte Funktion aus. Er gibt einen Namen für die Tätigkeit ein, definiert die Anzahl (z. B. der benötigten Stunden oder Stückzahl) und wählt einen Kostensatz aus. Er wiederholt diesen Vorgang so lange, bis alle Tätigkeiten, die für die Realisierung der Funktion benötigt werden, angelegt sind.
Varianten/Optionen	–
Ergebnis/Output	Die Tätigkeit ist angelegt und erscheint bei der betreffenden Funktion in der Liste der durchzuführenden Tätigkeiten.
Erläuterungen	–

Aktivität	Variantenbezogenes Kontext-Tag anlegen
Ziel	Eine konkrete Variante soll mit einem bestimmten Kontext-Tag versehen werden.
Rolle(n)	Schätzer
Voraussetzungen/Input	Der Anwender ist am System angemeldet und hat die passende Variante eines bestimmten Features geöffnet.
Ablauf	Der Anwender legt ein neues Kontext-Tag an (mit Name).
Varianten/Optionen	Optional gibt der Anwender einen Wert für das Kontext-Tag ein (z. B. einen Rabatt oder einen Risikozuschlag).
Ergebnis/Output	Das Kontext-Tag ist angelegt und mit dieser Variante verknüpft. Das Kontext-Tag kann auch bei anderen Projekten und Varianten verwendet werden (steht in der Auswahlliste zur Verfügung).
Erläuterungen	Neben der Funktion zum Anlegen von variantenbezogenen Kontext-Tags sollte es stets auch eine Möglichkeit geben, bereits angelegte Kontext-Tags einzusehen, zu bearbeiten und zu löschen.

Aktivität	Variante mit Funktionen verknüpfen
Ziel	Eine konkrete Variante soll mit bestimmten Funktionen verknüpft werden.
Rolle(n)	Schätzer
Voraussetzungen/Input	Der Anwender ist am System angemeldet und hat die passende Variante eines bestimmten Features geöffnet.
Ablauf	Der Anwender wählt aus der Liste der verfügbaren Funktionen alle Funktion aus, die für diese Variante benötigt werden.
Varianten/Optionen	–

Ergebnis/Output	Die Funktionen sind mit der betreffenden Variante verknüpft. Neben den Funktionen werden auch alle Tätigkeiten angezeigt, die für die Realisierung der Funktionen benötigt werden.
Erläuterungen	Neben der Funktion zum Verknüpfen von Funktionen mit Varianten sollte es stets auch eine Möglichkeit geben, bereits angelegte Verknüpfungen zu löschen.

Aktivität	Variante mit Kundenwunsch verknüpfen
Ziel	Eine konkrete Variante soll mit einem bestimmten Kundenwunsch verknüpft werden.
Rolle(n)	Schätzer
Voraussetzungen/Input	Der Anwender ist am System angemeldet. Mit einem Kundenwunsch als Ausgangspunkt hat er die passende Variante eines bestimmten Features geöffnet.
Ablauf	Der Anwender fügt die Variante den Leistungen hinzu (z. B. über eine Warenkorb-Funktion), die die Grundlage für sein Angebot bilden sollen.
Varianten/Optionen	–
Ergebnis/Output	Die ausgewählte Variante ist mit dem Kundenwunsch verknüpft, den der Anwender als Ausgangspunkt gewählt hatte. Die ausgewählte Variante erscheint im Leistungskatalog (z. B. in der Warenkorbansicht).
Erläuterungen	Neben der Funktion zum Verknüpfen von Varianten mit Kundenwünschen sollte es stets auch eine Möglichkeit geben, bereits angelegte Verknüpfungen zu löschen.

Aktivität	Schätzung auf Feature-Ebene durchführen
Ziel	Der Aufwand für ein bestimmtes Feature soll geschätzt werden.

Rolle(n)	Schätzer
Voraussetzungen/Input	Der Anwender ist am System angemeldet und hat das Feature geöffnet, dessen Aufwand geschätzt werden soll.
Ablauf	Der Anwender startet das Tool zum Vornehmen einer Schätzung auf Feature-Ebene und bestimmt in einer Auswahlbox die relative Aufwändigkeit des Features (gering, mittel, hoch bzw. unbekannt). Basierend auf den historischen Projektdaten bestimmt das System einen geschätzten Aufwand für das Feature.
Varianten/Optionen	–
Ergebnis/Output	Dem Nutzer werden ein erwarteter Wert, eine untere Grenze und eine obere Grenze angezeigt. Der erwartete Wert wird vom System als Wert für die Kalkulation vorgeschlagen; dieser Wert kann vom Nutzer überschrieben werden.
Erläuterungen	Je mehr historische Daten zu abgeschlossenen Projekten bzw. entwickelten Features im System hinterlegt sind, desto zuverlässiger sind die für den Aufwand ermittelten Werte. Die untere und obere Grenze definieren einen Bereich, in dem mit einer Wahrscheinlichkeit von über 66 % der tatsächliche Aufwand liegen wird (bei Annahme einer Normalverteilung).

Aktivität	Schätzung auf Funktionsebene durchführen
Ziel	Der Aufwand für die Funktionen eines bestimmten Features sollen geschätzt werden.
Rolle(n)	Schätzer
Voraussetzungen/Input	Der Anwender ist am System angemeldet und hat das Feature geöffnet, bei dem der Aufwand der Funktionen geschätzt werden soll.
Ablauf	Der Anwender startet das Tool zum Vornehmen einer Schätzung auf Funktions-Ebene und bestimmt

	in einer Auswahlbox pro Funktion die relative Aufwändigkeit der Funktion (gering, mittel, hoch bzw. unbekannt). Basierend auf den historischen Projektdaten bestimmt das System einen geschätzten Aufwand für die Funktionen.
Varianten/Optionen	–
Ergebnis/Output	Dem Nutzer werden pro Funktion ein erwarteter Wert sowie ein Unsicherheitsintervall angezeigt, außerdem ein geschätzter Gesamtaufwand aller Funktionen mit unterer Grenze und oberer Grenze. Der erwartete Gesamtwert wird vom System als Wert für die Kalkulation vorgeschlagen; dieser Wert kann vom Nutzer überschrieben werden.
Erläuterungen	Je mehr historische Daten zu abgeschlossenen Projekten bzw. entwickelten Funktionen im System hinterlegt sind, desto zuverlässiger sind die für den Aufwand ermittelten Werte. Die untere und obere Grenze definieren einen Bereich, in dem mit einer Wahrscheinlichkeit von über 66 % der tatsächliche Aufwand liegen wird (bei Annahme einer Normalverteilung).

Aktivität	Warenkorb bearbeiten
Ziel	Die einzelnen Positionen des Leistungskatalogs für ein Projekt sollen geprüft und bei Bedarf angepasst werden.
Rolle(n)	Schätzer, Vertriebsmitarbeiter
Voraussetzungen/Input	Der Anwender ist am System angemeldet. Es befinden sich Varianten im Warenkorb.
Ablauf	Der Anwender öffnet das betreffende Projekt in der Warenkorbansicht. Das System zeigt alle ausgewählten Varianten als Positionen im Leistungskatalog an (inkl. Aufwand und Kosten). Der Anwender kann die Kontext-Tags des Projekts bearbeiten und einzelne Positionen als optional markieren, so

	dass diese nicht in der Kalkulation berücksichtigt werden.
Varianten/Optionen	–
Ergebnis/Output	Die Änderungen des Anwenders sind gespeichert und die Warenkorbansicht ist aktualisiert.
Erläuterungen	–

Aktivität	Tatsächliche Aufwände erfassen
Ziel	Die tatsächlich angefallenen Aufwände eines Projekts sollen bei den einzelnen Positionen des Leistungskatalogs dieses Projekts hinterlegt werden.
Rolle(n)	Schätzer, Vertriebsmitarbeiter
Voraussetzungen/Input	Der Anwender ist am System angemeldet.
Ablauf	Der Anwender öffnet das betreffende Projekt in der Warenkorbansicht. Das System zeigt den Leistungskatalog mit allen Positionen an (Varianten inkl. Aufwand und Kosten). Der Anwender kann pro Positionen erfassen, ob der tatsächliche Aufwand vom geschätzten Aufwand abwich.
Varianten/Optionen	Optional kann der Anwender eine Begründung für die Abweichung hinterlegen.
Ergebnis/Output	Die Änderungen des Anwenders sind gespeichert und die Warenkorbansicht ist aktualisiert.
Erläuterungen	Für die Schätzung künftiger Projekte werden stets die tatsächlichen Aufwände herangezogen, die für diese Variante in diesem Projekt benötigt wurden.

3 Praxisbeispiel

Im Folgenden wird die Anwendung der Abakus-Methode kurz anhand eines praktischen Beispiels demonstriert. Hierbei kommt das Werkzeug zur Anwendung, das von der Uni Trier im Rahmen des Abakus-Projekts entwickelt wurde. Dieses Werkzeug wurde als Webanwendung auf Basis des Frameworks CodeIgniter realisiert und mittels Bootstrap responsiv aufgebaut

Nach Möglichkeit sollte ein Werkzeug den gesamten Prozess der Aufwands- und Kostenschätzung unterstützen, der in Kapitel 2.1 beschrieben ist. Dieser beginnt mit dem Anlegen von Kundenwünschen bzw. Anforderungen, die entweder manuell erfasst oder semi-automatisch importiert werden können. Bei der Nutzung eines geeigneten Werkzeugs können Lastenhefte oder Briefingdokumente gescannt und auf Kundenwünsche hin analysiert werden. Identifizierte Kundenwünsche lassen sich auf diese Weise dann leicht über einen einfachen Kontrollmechanismus ins Tool übernehmen. Eine natürlichsprachliche Anforderung wie „als Webseitenbesucher möchte ich mit dem Betreiber Kontakt aufnehmen können“ kann sofort als Kundenwunsch übernommen werden.

Im nächsten Schritt kann der Anwender dann die Erfahrungsdatenbank nach Features durchsuchen, die zu den erfassten Kundenwünschen passen. Abbildung 2 zeigt einen Screenshot des Werkzeugs. Zur Umsetzung des oben genannten Kundenwunschs wurde hier das Feature „Kontaktformular“ ausgewählt.

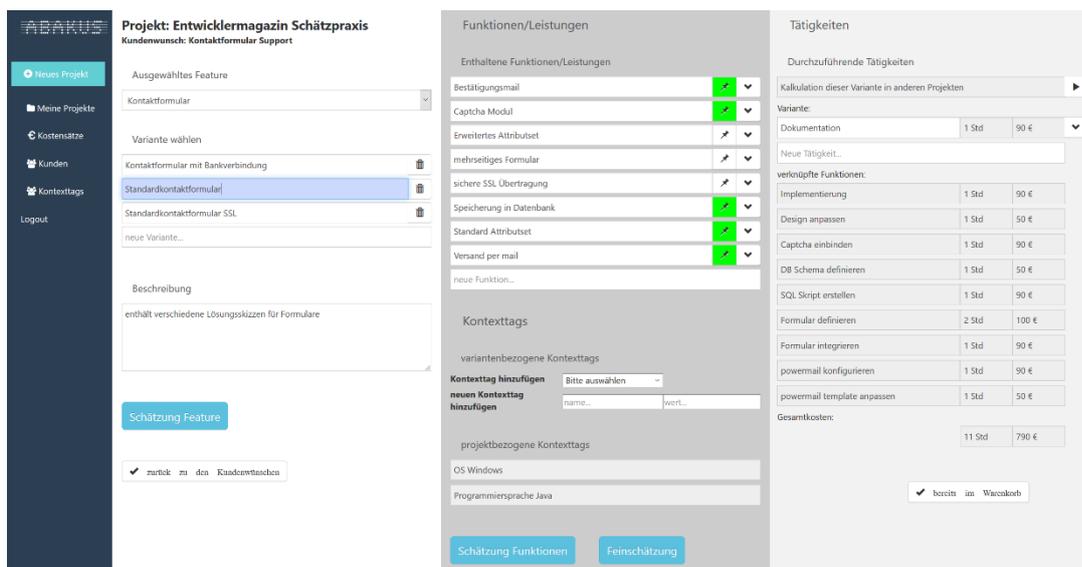


Abbildung 2: Kalkulation eines Features im Abakus-Werkzeug

Oft hat ein Unternehmen in vergangenen Projekten bereits Features entwickelt, die zu einem aktuellen Kundenwunsch passen (so wie dies auch in diesem Beispiel der Fall ist). Existiert für einen Kundenwunsch noch kein passendes Feature, so kann dieses natürlich im Werkzeug angelegt werden.

Auf diese Weise wird ein Gesamtprojekt nach dem Prinzip „Teile und herrsche“ in einzelne Features zerlegt. Meist werden viele grundlegende Features gefunden, die bei der Softwareentwicklung im eigenen Umfeld bereits entwickelt oder verwendet wurden, auch wenn ein Projekt neu ist und die Anforderungen an das Projekt zum Zeitpunkt der Schätzung noch nicht vollständig ausgearbeitet sind. Solche identifizier-

ten Features stellen eine deutlich stabilere Basis für anschließende Schätzungen dar als der direkte Vergleich zwischen Projekten auf Basis ihres Gesamtumfangs. Existierende Features wurden oft auch schon in verschiedenen Varianten umgesetzt. Bei einem Kontaktformular kann es beispielsweise eine Basisvariante geben oder eine aufwändigere Varianten mit Spamschutz, komplexerer Formularlogik oder SSL-Verschlüsselung. Solche Varianten eines Features unterscheiden sich folglich hinsichtlich ihres Funktionsumfangs.

In der exemplarischen Implementierung des Abakus-Werkzeugs werden die Funktionen, die einem Feature zugeordnet sind, in der Mitte des Bildschirms konfiguriert. Die Funktionen können mit der ausgewählten Variante eines Features verknüpft werden (ganz anschaulich mit grünen Pins) und fließen dann automatisch in die Kalkulation dieses Features ein. Dies ist möglich, da alle gespeicherten Features mitsamt ihren Varianten in der Wissensbasis gespeichert sind. Die Varianten, die in früheren Projekten realisiert wurden, enthalten immer auch Informationen zu den Aufwänden, die in den jeweiligen Projekten für die Umsetzung benötigten wurden. Bei Bedarf können den Funktionen auch ergänzende Dokumente zugeordnet werden (z. B. ein Foliensatz oder kommentierter Bildschirmausdruck), die die Funktionen detailliert beschreiben und die vom Anwender eingesehen werden können.

Anschließend findet die Kalkulation der Funktionen und Lösungsvarianten statt. Diese erfolgt über die Zuordnung von Tätigkeiten, die die notwendigen Arbeiten bei der Realisierung beschreiben. Allen Tätigkeiten wird der für die Umsetzung benötigte Personalaufwand sowie ein entsprechender Kostensatz zugeordnet. Die zur Umsetzung der ausgewählten Variante notwendigen Tätigkeiten befinden sich in Abbildung 2 in der Spalte ganz rechts. Die Kosten für die Umsetzung eines Features und seiner Funktionen sind bei der Arbeit mit dem exemplarisch entwickelten Abakus-Werkzeug jederzeit sichtbar und unterstützen den Kalkulator so bei seiner Arbeit.

Sowohl ganze Projekte als auch Features und ihre Varianten können zusätzlich mit sogenannten Kontext-Tags versehen werden. Diese Tags enthalten beispielsweise Informationen zur Systemumgebung oder zu bestimmten nichtfunktionalen Anforderungen wie Sicherheit, Wartbarkeit oder Usability. Anhand der Kontext-Tags können generelle Aufwandstreiber greifbar und in Form von Aufschlägen bzw. Rabatten kalkulierbar gemacht werden, etwa wenn in einem Projekt erstmals eine neue Technologie verwendet wird oder wenn die Kundenbeziehung zu dem Auftraggeber in der Vergangenheit nicht ganz einfach war.

Das Werkzeug berücksichtigt diese Kontext-Tags bei der Kalkulation und kann so das aktuelle Projekt besser mit bereits abgeschlossenen Projekten vergleichen. Das Wissen über den Projektkontext sowie die Beziehungen zwischen den Features, ihren Varianten und den darin enthaltenen Funktionen bilden die Basis für die Suche nach historischen Aufwandsdaten. Nachdem eine entsprechende Variante eines Features konfiguriert und kalkuliert wurde, kann diese über den Button „In den Warenkorb“ mit dem Kundenwunsch verknüpft werden.

Neben der gezeigten Möglichkeit zur Sichtung und Übernahme von Daten aus historischen Projekten bietet das Werkzeug auch die Möglichkeit, die vorhandenen Daten direkt in eine durch das Werkzeug bereitgestellte Schätzung einfließen zu lassen. Abhängig vom Kenntnisstand über das zu schätzende Projekt werden hierbei die Möglichkeiten einer Grobschätzung von Features und Funktionen angeboten als auch eine Hilfestellung beim Auffinden einer möglichst ähnlichen Variante des zu schätzenden Features, die dann als

Ausgangsbasis für eine Feinschätzung durch einen Experten dienen kann. Neben den Aufwandsdaten aus der Erfahrungsdatenbank berücksichtigt das Werkzeug also bei einer Schätzung bis zu drei weitere Informationsquellen:

- das implizite Erfahrungswissen des Anwenders/Schätzer, denn dieser wird um eine grobe persönliche Einschätzung (gering, mittel, hoch) der Aufwändigkeit gebeten,
- die Ähnlichkeit des aktuellen Projekts den verfügbaren historischen Projekten, denn diese wird anhand der vergebenen Kontext-Tags mittels eines asymmetrischen Ähnlichkeitsmaßes bestimmt, und
- die Überlappung der Funktionen in der zu schätzenden Featurevariante mit den Funktionen in den Varianten historischer Projekte.

Den Kern eines jeden Schätzergebnisses bildet ein Wert für den erwarteten Aufwand. Dieser ist mit einem Unsicherheitsintervall versehen, das mit Rücksicht auf die verfügbare Datenlage den Schwankungsbereich bezüglich des tatsächlichen Werts zum Ausdruck bringt. Abbildung 3 zeigt das Beispielergebnis einer Schätzung des Tools für den voraussichtlichen Gesamtaufwand bei der Umsetzung einer gewählten Lösungsvariante. Hierbei wird eine Unter- und Obergrenze für die Gesamtschätzung angezeigt, daneben der individuelle Beitrag der einzelnen Funktion zum erwarteten Gesamtaufwand sowie die durch diese verursachte Unsicherheit in der Schätzung. Das letzte Wort bei der Schätzung hat jedoch stets der Anwender. Dieser ist für den Feinschliff der Schätzung verantwortlich und kann die vorgeschlagenen Werte anpassen bzw. überschreiben.

Relative Aufwändigkeit Funktionen:

Geschätzter Aufwand
 - Untere Grenze (66% Intervall) = 10
 - Erwarteter Wert (im Mittel) = 12
 - Obere Grenze (66% Intervall) = 14

Rang	Funktionen	Entw. Aufw.	Unsicherheit
1.	Erweitertes Attribusset	0	± 1
2.	mehrsätige Formular	2	± 1
3.	Versand per mail	2	± 1
4.	Bestätigungsmail	2	± 0
5.	Captcha Modul	1	± 0
6.	sichere SSL Übertragung	0	± 0
7.	Speicherung in Datenbank	2	± 0
8.	Standard Attribusset	3	± 0

Wert für Kalkulation:

Abbildung 3: Schätzung von Funktionen im Abakus-Werkzeug

4 Prinzipien guter Schätzung

Vom Abakus-Projekt wurden folgende Praxistipps für verbesserte Aufwandsschätzungen erarbeitet, die angewendet werden können, auch ohne, dass die Abakus-Methode selbst zum Einsatz kommt. Personen, die häufiger Schätzungen vornehmen, können von diesen Prinzipien profitieren und die Aufwandsschätzungen im eigenen Umfeld optimieren:

- Dokumentiere die Erfahrungen, die bei der Umsetzung von Projekten gemacht wurden, insbesondere hinsichtlich des tatsächlich benötigten Aufwands zur Umsetzung von Features. Nur so lässt sich eine verlässliche Basis für zukünftige Schätzungen zu schaffen.
- Teile und herrsche: Auch wenn die Anforderungen an das Projekt zum Zeitpunkt der Schätzung noch nicht vollständig ausgearbeitet sind – zerlege das Projekt und identifiziere zumindest die Features, die durch die Software bereitgestellt werden sollen. Jedes Projekt ist neu, die grundlegenden Features, die bei der Softwareentwicklung im eigenen Umfeld häufig eine Rolle spielen, sind aber deutlich stabiler. Unsystematische Schätzfehler in den Featureschätzungen mitteln sich in der Gesamtschätzung zudem häufig aus.
- Ziehe für die Schätzung eines Features nach Möglichkeit die tatsächlichen Aufwände heran, die bei der Umsetzung des Features in früheren Projekten angefallen sind. Auch erfahrene Schätzer neigen nachweislich zu systematischen Fehlern in ihren Schätzungen. Diese Fehler lassen sich am besten begrenzen, indem die tatsächlichen Aufwände die Ausgangsbasis künftiger Schätzungen bilden.
- Berücksichtige Gemeinsamkeiten und Unterschiede zwischen dem aktuellen Projekt und früheren Projekten, um Abweichungen nach oben oder unten in der Schätzung zu dokumentieren. Annahmen, die gemacht werden, werden damit explizit und die Schätzung wird nachvollziehbarer.
- Bestimme nicht nur den wahrscheinlichen Schätzwert, sondern auch einen minimalen und einen maximalen Wert. Berücksichtige dabei insbesondere die Spannweite der tatsächlichen Aufwände bei der Umsetzung des Features in früheren Projekten.

5 Quellen

- ¹ A Trendowicz and R. Jeffery, *Software Project Effort Estimation. Foundations and Best Practice Guidelines for Success*. Springer Verlag, 2014.
- ² M. Cohn, *Agile estimating and planning*. Prentice-Hall, Upper Saddle River, NJ, 2005.
- ³ M. Jørgensen and S. Grimstad, "Software Development Effort Estimation: Demystifying and Improving Expert Estimation," in A. Tveito, A.M. Bruaset, and O. Lysne (ed.) *Simula Research Laboratory by Thinking Constantly about it*. Springer Verlag, 2010, pp. 381–403.
- ⁴ D. Kahneman, A. Tversky, "Intuitive Prediction: Biases and Corrective Procedures." In S. Makridakis and S. C. Wheelwright, Eds., *Studies in the Management Sciences: Forecasting*, Amsterdam, Holland, 1979.
- ⁵ ISO/IEC 20926 - IFPUG Functional Size Measurement Method 2009, 2nd Ed. International Standardization Organization, Geneva, 2009.
- ⁶ B. W. Boehm, *Software Engineering Economics*. Prentice Hall, 1981.
- ⁷ B. Boehm, C. Abts, A. Brown, S. Chulani, B. Clark, E. Horowitz, R. Madachy, D. Reifer, B. Steece, *Software cost estimation with COCOMO II*. Prentice Hall, Upper Saddle River, 2000.
- ⁸ E. Mendes, "Predicting Web Development Effort Using a Bayesian Network." In *Proceedings of the 11th International Conference on Evaluation and Assessment in Software Engineering*, Keele University, UK, 2-3 April 2007, pp. 83–93.
- ⁹ A. Trendowicz, *Software Cost Estimation, Benchmarking, and Risk Assessment. The Software Decision Maker's Guide to Predictable Software Development*. Springer Verlag, 2013.
- ¹⁰ M. Shepperd, C. Schofield, "Estimating software project effort using analogies." In *IEEE Transactions on Software Engineering*, no. 23, vol. 11, pp. 736–743, 1997.
- ¹¹ A. Aamodt and E. Plaza, „Case-based reasoning: foundational issues, methodological variations, and system approaches“, *AI Communications*, vol. 7, no. 1, 1994, pp.39–59.
- ¹² S. Robertson and J. Robertson. *Mastering the requirements process: Getting requirements right*. Addison-Wesley, 2012.
- ¹³ A. Cockburn, *Writing effective use cases*. Addison-Wesley, 2001.
- ¹⁴ C. Rupp, *Requirements Engineering: Ein Überblick*. dpunkt.verlag, 2012.
- ¹⁵ Beck, Kent, et al. *Manifesto for Agile Software Development*. 2001.
- ¹⁶ J. Natt och Dag, R. Björn, P. Carlshamre, M. Andersson, and J. Karlsson, "Evaluating automated support for requirements similarity analysis in market-driven development." *Proceedings of the 7th International Work-shop on Requirements Engineering: Foundation for Software Quality*, Interlaken, Switzerland, 2001. Pp. 190–210.
- ¹⁷ S. Park, H. Kim, Y. Ko, and J. Seo, "Implementation of an efficient requirements-analysis supporting system using similarity measure techniques", *Information and Software Technology*, vol. 42, no. 6, 15 April 2000, pp. 429–438.
- ¹⁸ D. Kulak and E. Guiney. *Use cases: requirements in context*. Addison-Wesley, 2012.
- ¹⁹ P. P. Chen, "The entity-relationship model—toward a unified view of data." In *ACM Transactions on Database Systems (TODS) 1.1*, pp. 9-36, 1976.
- ²⁰ S. A. White, *Introduction to BPMN. IBM Cooperation 2.0*, 2004