Aufwandsschätzung in Softwareprojekten

Werden wir es schaffen? Und was wird es kosten?

Diese Fragen stellt sich jeder Projektleiter und Product Owner regelmäßig. Um sie zu beantworten, wird dann mühsam der Aufwand für die Entwicklung dessen geschätzt, was noch vor uns liegt. Wobei wir oft auch spät im Projekt nur ein sehr vages Bild davon haben, was das eigentlich ist. Entsprechend ungenau sind die Schätzungen. Am Ende verstreicht Termin für Termin und wir stecken viel zu lange ganz kurz vor dem Ziel fest.

Warum aber investieren wir immer noch viel Zeit in Aufwandsschätzungen, wenn sie am Ende selten stimmen? Warum nicht einfach einen Budgetrahmen setzen und – beginnend beim Wichtigsten – Funktion für Funktion umsetzen bis das Budget aufgebraucht ist? Genau diese Idee verfolgt der #NoEstimate Ansatz, der gerade in der agilen Community diskutiert wird.

Abb. 1 - Schätzen ist nicht raten und nicht wahrsagen. Auch wenn es manchmal so gemacht wird.

Auf den ersten Blick klingt das planlos und unberechenbar. Aber ist es das wirklich? Gerade im Vergleich zu Projekten, die zwar zu Beginn genau geschätzt und geplant werden, deren Pläne jedoch am Ende nicht halten?

Indem mit den wichtigsten Funktionen begonnen und die Anforderungen nach Wichtigkeit geordnet abgearbeitet werden, ist beim #NoEstimate Ansatz sichergestellt, dass rasch ein Ergebnis vorliegt, welches einen ersten Wert für den Kunden liefert. Am Ende sind zumindest die wichtigen Dinge umgesetzt. Wenn sich etwas nicht mehr ausgeht, dann sind es weniger wichtige Funktionen. Ich finde das ist zumindest ein überlegenswerter Gedanke.

Auch wenn der #NoEstimate Ansatz verlockend klingt – man spart sich den Aufwand fürs Schätzen – gibt es dennoch gute Gründe, um in Projekten Aufwandsschätzungen zu machen: Es zwingt zum Durchdenken des Projektes und Aufdecken von Risiken („Der Plan ist nichts, Planen ist alles."), es schafft eine Basis für die grobe Meilensteinplanung und die detailliertere Aufgabenplanung und in vielen Projekten fordert es schlichtweg der Kunde.

Egal aus welchen Gründen wir Aufwände schätzen, wir sollten dies methodisch und strukturiert tun. Im folgenden Artikel finden Sie einen kurzen Einstieg ins Thema. Es folgt ein Überblick über Schätzverfahren, die sich bei unseren Kunden in agilen und plangetriebenen Projekten bewährt haben und 10 Goldene Regeln, die Ihnen helfen, die Effizienz beim Schätzen sukzessive zu erhöhen und Aufwandsschätzungen erfolgreich einzusetzen.

Wahrsagerei mit System

Egal wie wir es drehen und wenden, eine Schätzung ist immer ein Blick in die ungewisse Zukunft.

Erst wenn ein Projekt wirklich fertig abgeschlossen ist, wissen wir, was es gekostet hat (und nicht einmal dann genau, denn die Gewährleistungs- und Fehlerkosten sind auch hier noch nicht bekannt).

Was also tun, um die Unschärfe etwas zu verringern?

Es gibt dafür einige Möglichkeiten, die auch in diesem Knowledge Letter beschrieben werden wie z.B. mehrere „Wahrsager" schätzen lassen und hoffen, dass dabei etwas Vernünftig(er)es herauskommt. Oder das Schätzobjekt genauer kennenlernen bzw. definieren, damit die Unsicherheitsfaktoren geringer werden, etc.

Man kann also bei der Aufwands-Wahrsagerei mit etwas System und ein paar Methoden durchaus weg vom reinen „Glaskugel lesen" zu einer gewissen Seriosität und Aussagekraft kommen.

Man muss es nur tun und sich dann aber bewusst sein, dass das Ergebnis trotzdem nicht „in Stein gemeißelt", sondern eine unscharfe Schätzung ist. Und dass, Schätzungen von etwas „Neuem" immer mit einer größeren Unschärfe verbunden sind, wie Schätzungen von Themen, die man schon oft in ähnlicher Form umgesetzt hat.

Dipl.-Ing. Johannes Bergsmann
Staatl. befugter und beeideter Ingenieurkonsulent für Informatik

Der Quality Knowledge Letter ist ein periodisches Informationsmedium von Software Quality Lab und dessen Partnern mit dem Schwerpunkt SW-Qualität.

Inhalt: fachliche Beiträge und Schwerpunktthemen, Vorstellung neuer Produkte und Leistungen, neue wissenschaftliche Erkenntnisse und andere Fachbeiträge aus unseren Themenbereichen.

Aktuelle Fach- und Forschungsbeiträge sind willkommen. Einsendungen an info@software-quality-lab.com

Weitere Infos zu diesem und anderen Themen finden Sie auf **www.software-quality-lab.com**

Aufwandsschätzung in Software Projekten

Aufwandsschätzungen brauchen wir in beinahe jedem Software Projekt, egal ob agil oder plangetrieben. Sie ermöglichen die Zeit- und Budgetplanung und sind die Basis vieler Entscheidungen und die Wurzel von Schweiß und Tränen. Viele Projekte geraten gegen Ende unter immensen Druck, oft auch aufgrund falscher Schätzungen. Dabei gibt es eine Reihe guter Schätzmethoden, von Delphi und 3-Punkt-Schätzungen auf der klassischen Seite, bis zu T-Shirt-Sizing und Planning Poker in agilen Methoden. Darüber hinaus helfen einfache Grundregeln dabei, die Schätzgenauigkeit zu erhöhen und den Umgang mit Schätzwerten erfolgreicher zu gestalten.

Schätzen als Grundlage für Entscheidungen

Nach wie vor kommt man in den wenigsten Software Projekten ohne Schätzung und zumindest ein Minimum an Planung aus. Der Projektsponsor will wissen, was es kosten wird. Der Auftraggeber möchte sichergehen, dass die Projektziele erreicht werden und maximaler Nutzen generiert wird. Die Anwender wollen wissen, wann das neue System verfügbar sein wird. Für die Beantwortung all dieser Fragen braucht man frühzeitig belastbare Schätzungen von Aufwand und Zeit. Diese Schätzungen sind dann die Basis für kaufmännische Entscheidungen (Go/No-go, Make-or-Buy) und organisatorische Rahmenbedingungen des Projektes (Teamgröße, Zeitplan etc.).

Aufwandsschätzungen sind auch eine wesentliche Komponente zur Reduzierung des Projektrisikos. Je besser wir den Aufwand abschätzen können, desto geringer ist das Risiko von Zeit- oder Kostenüberschreitungen (Abb. 2).

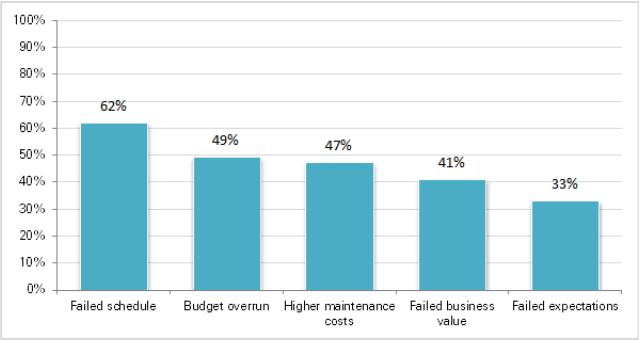

Abb. 2- Viele Softwareprojekte erreichen Termin- oder Budgetziele nicht. [1]

Auch technische Entscheidungen werden basierend auf Aufwandsschätzungen getroffen. Beispielsweise beeinflusst der Aufwand für die Programmierung die Auswahl eines technischen Frameworks ebenso, wie der Aufwand für Wartung und laufende Weiterentwicklung.

Schätzung ≠ Wissen

Egal wofür Schätzungen verwendet werden, man muss sich bewusst sein, dass es eine Schätzung ist und kein gesichertes Wissen. Schätzungen bergen immer das Risiko, falsch oder zumindest ungenau zu sein.

In unserer Projekterfahrung hat sich herauskristallisiert, dass gute Teams eine Schätzgenauigkeit von +/-15% erreichen. Je nach Projektgegenstand und Zeitpunkt der Schätzung kann dies aber deutlich variieren, gerade in frühen Projektphasen ist die Schwankungsbreite wesentlich höher (Abb. 3).

Phase	Minimum	Maximum	Range
Initial Concept	-75%	300%	16,00x
Approved Product Definition	-50%	100%	4,00x
Requirements Complete	-33%	50%	2,25x
User Interface Design Complete	-20%	25%	1,60x
Detailed Design Complete	-10%	10%	1,20x

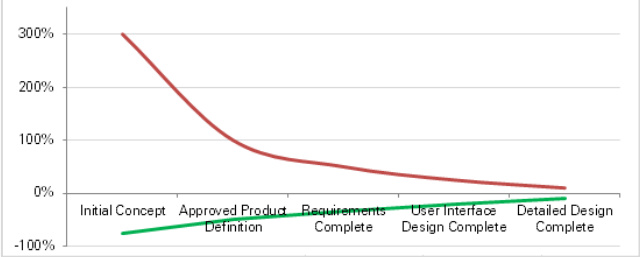

Abb. 3 – Die Schätzungenauigkeit ist zu Beginn sehr hoch. [2]

Fixe Zusagen basierend auf Schätzungen sind riskant. Trotzdem werden in der Praxis immer wieder Schätzungen als verbindliche Zusagen interpretiert und sofort Budget- und Zeitpläne darauf aufgebaut. Diese werden an Kunden kommuniziert und fix versprochen, ohne darauf hinzuweisen, dass es sich um Schätzungen mit zum Teil erheblicher Unschärfe handelt. Die Konsequenz sind oft Projekte, die schon vor der ersten Zeile Code stark unter Druck stehen.

#NoEstimate – Ein Budget reicht?

Vor Beginn eines Projektes möchte der Auftraggeber wissen, was es kosten wird. Zu diesem Zeitpunkt sind die Anforderungen ungenau und oft fehlt Wissen zu Technologien. Vielfach wird nun versucht, mit viel Aufwand die Anforderungen genau zu erheben die Kosten abzuschätzen. Am Ende stellt sich heraus, dass die tatsächlich benötigten Funktionen eigentlich ganz andere waren und die Schätzung der Aufwände nicht stimmte.

Der #NoEstimate Ansatz versucht diese Probleme zu lösen. Statt einer genauen Aufwandsabschätzung basierend auf detaillierten und vollständigen Anforderungen wird lediglich ein grobes Budget aus groben Zielen und Features abgeleitet. Dieses wird als Obergrenze fixiert. Danach wird iterativ das System entwickelt, bis das Budget aufgebraucht ist. Aufgabe des Product Owners ist dabei, dafür zu sorgen, dass jederzeit am Richtigen gearbeitet wird. Eine genaue Schätzung der einzelnen Funktionen oder Aufgabenpakete erfolgt dann nicht mehr [3].

Vollständiger Knowledge Letter Zugang

Wir freuen uns, dass Sie an diesem Thema Interesse haben und den Knowledge Letter von Software Quality Lab bis hierher gelesen haben.

Dieser Knowledge Letter ist eine Vorschau (gekürzte Version des gesamten Artikels).

Wenn Sie den ungekürzten Knowledge Letter lesen möchten, registrieren Sie sich bitte unter
http://www.software-quality-lab.com/download/knowledge-letter/anfrage-knowledge-letter/

Sie erhalten nach der Registrierung vollen Zugang zu allen bisherigen Knowledge Lettern von Software Quality Lab und erhalten automatisch künftige Knowledge Letter per E-Mail.

Software Quality Days — Die größte Konferenz zum Thema „Software Qualität“ in Europa!

Besuchen Sie die Top-Konferenz mit allen Infos rund um Software Qualität.

Beste Qualität der Vorträge und Tutorials sowie eine Mischung aus praktischen und wissenschaftlichen Beiträgen machen die Software Quality Days zum Top-Event.

In den 3 praktischen Tracks werden anwendungsorientierte Vorträge präsentiert. Der wissenschaftliche Track zeigt Beiträge mit hohem Innovationsgrad und praktischer Anwendbarkeit, basierend auf Forschungs-ergebnissen. Im Solution Provider Forum präsentieren Aussteller ihre neuesten Tools mit Praxis-Beispielen.

Nähere Infos unter

www.software-quality-days.com