

■ Aufwand für Softwareprojekte realistisch schätzen

Herausforderung für Dienstleister und Auftraggeber





Erfolgsvoraussetzung: realistische Anforderungen

“Eine Sache ist genau das wert, was der Käufer dafür zu zahlen bereit ist (Publius Syrus, Römischer Schriftsteller, 1. Jh. v. Chr.)” Diese betriebswirtschaftliche Definition ist bei Individualentwicklung leider nicht sehr hilfreich, stehen die Kosten doch bei Entwicklungsaufgaben erst nach Abschluss des Projektes definitiv fest.

Die Kunden müssen, oftmals im Rahmen zu genehmigender Budgets, für Kosten-Nutzen-Analysen oder zur Reduzierung des Projektrisikos vorab den Preis festlegen. Daher ist man auf Schätzungen des Aufwandes angewiesen. Und diese Schätzungen sind, was schon in der Bedeutung des Wortes steckt, mit Unsicherheiten verbunden.

Traum Kalkulationsmaschine

Diese Unsicherheiten bieten ein breites Feld für Verhandlungen und Diskussionen, denn der Auftraggeber ist zumeist an einem niedrigen Preis interessiert. Der Auftragnehmer benötigt angesichts von Projektrisiken und zu erwartenden Aufwänden jedoch ein ausreichendes Projektbudget. Gäbe es die exakte, schnelle und kostengünstige Kalkulationsmaschine (lassen Sie mich einfach mal träumen), dann könnte man den Preis der Individualsoftware auf den Stundenpreis reduzieren, der mit betriebswirtschaftlichen Methoden berechnet werden kann.

Man könnte nun meinen, dass Forschung und Industrie angesichts der wirtschaftlichen Bedeutung von Aufwandsschätzungen hierfür ein allgemein akzeptiertes und verlässliches Verfahren entwickelt haben. Die Realität sieht jedoch anders aus.

Abhängig von der Antwort auf die Frage: Ist Softwareentwicklung künstlerisches Wirken, ein (Kunst-) Handwerk oder aber Ingenieursarbeit? wird in der Regel die nächste Frage beantwortet, wie realistisch nämlich Aufwandsschätzungen für Softwareprojekte grundsätzlich sein können.

Die bisher zur Verfügung stehenden Verfahren sind zu aufwändig und fehlerbehaftet. Daher konnten sie sich in der Praxis bisher nur bei sehr großen Projekten durchsetzen, die nach dem Wasserfallmodell abgewickelt werden.

Fehlende Inputdaten

Bei inkrementellem, iterativem Vorgehen im Rahmen leichtgewichtiger Prozesse sind diese Verfahren nicht einsetzbar, da die Inputdaten zu Beginn des Projekts nicht verfügbar sind. Umso wichtiger ist es, dass sich alle Beteiligten der Unsicherheit bewusst sind, die mit einer Schätzung untrennbar einhergeht. Letztlich führt auch jedes Risiko eines Projekts zu einer Unsicherheit in der Schätzung. Und die

Schätzung bildet oftmals auch den Ausgangspunkt zur Bewertung des Projekterfolgs, da nur dann von einem erfolgreichen Projekt gesprochen werden kann, wenn es

- nicht nur pünktlich zum geplanten Zeitpunkt fertig gestellt und ausgeliefert ist,
- alle in Auftrag gegebenen Anforderungen bzw. Eigenschaften enthält,
- die Software bei Auslieferung fehlerfrei und performant ist, sondern wenn zudem
- das Projekt im Rahmen der geplanten Kosten realisiert wird.

Grundvoraussetzung dafür ist im Übrigen: Die gestellten Anforderungen müssen im gegebenen Rahmen auch realisierbar sein. Dieser Rahmen wird aber immer enger. Das führt zur Verschlechterung der Softwarequalität, denn die Entwickler müssen immer komplexere und sich schnell ändernde Anforderungen unter steigendem Zeitdruck umsetzen.

Festpreiswünsche

Zurück zur Frage der Aufwandsschätzung. Selbstverständlich gibt es bei den Fachkräften eines jeden IT-Dienstleisters, auch bei DMC, die gesamte Bandbreite von Einstellungen zur Schätzbarkeit von Projekten. Fakt aber ist, dass immer häufiger Festpreise vom Kunden gefordert werden, oftmals schon auf der Basis von Fachgrobkonzepten. Daher sollten beide Seiten, Dienstleister und Auftraggeber, eine realistische Einstellung zu Schätzungen und ihrem Zustandekommen haben.

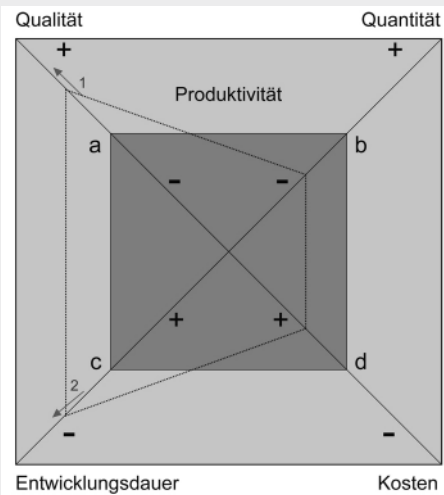
Das hilft auch, die eine oder andere Fehlerquelle auszuschalten. Beispielsweise lässt sich empirisch nachweisen, dass Softwaremanager einmal abgegebene Schätzungen selten zu revidieren bereit sind, selbst wenn die für die erste Schätzung zur Verfügung stehende Zeit unzweifelhaft viel zu kurz war (Tom De-Marco 1). Wer sich dieser menschlichen Neigung bewusst ist, kann an ihrer Überwindung arbeiten.

Teufelsquadrat

Bedenkenswert ist auch das Teufelsquadrat nach Sneed. Es besagt, dass die Produktivität eines Projektteams invariant ist, so dass die Faktoren Quantität und Qualität der Anforderungen sowie Entwicklungsdauer und Kosten voneinander abhängig sind. Das heißt die Qualität kann nur erhöht bzw. die

Dauer verkürzt werden, wenn der Umfang der Anforderungen gesenkt oder aber das Budget erhöht wird.

Steigerungen der Produktivität sind nur durch langfristige Maßnahmen möglich, die für ein einzelnes Projekt keine Steigerung, oftmals sogar eine vorübergehende Verringerung der Produktivität mit sich bringen.



Teufelsquadrat nach Sneed.

Bewusst machen sollte sich jeder Projektverantwortliche auch das Schätzdilemma Garbage in Gabage out (GIGO): Die Schätzung von Softwarekosten kann stets nur so gut sein wie die zugrundegelegten Eingangsgrößen.

Analogiemethode und Bauchwert

In der Praxis sind weiterhin Pseudoschätzungen weit verbreitet. Beispielsweise vergleicht der Experte ein neues Projekt mit ähnlichen Projekten (Analogiemethode), was sich meist schwer nachvollziehen lässt. Oder die Projektlaufzeit wird anhand bereits beendeter Phasen berechnet (Prozentsatzmethode), was bestenfalls nach Abschluss der Analysephase funktioniert – falls die nicht aus Kostengründen abgebrochen wurde!

Praxisnah und realistisch, allerdings wenig hilfreich ist das Parkinson Gesetz: Die Arbeit ist beendet, wenn alle Vorräte aufgebraucht sind. Und die Vorgehensweise Price to Win - die Software-Kosten werden auf das Budget des Kunden geschätzt - führt mittelfristig in den Ruin.

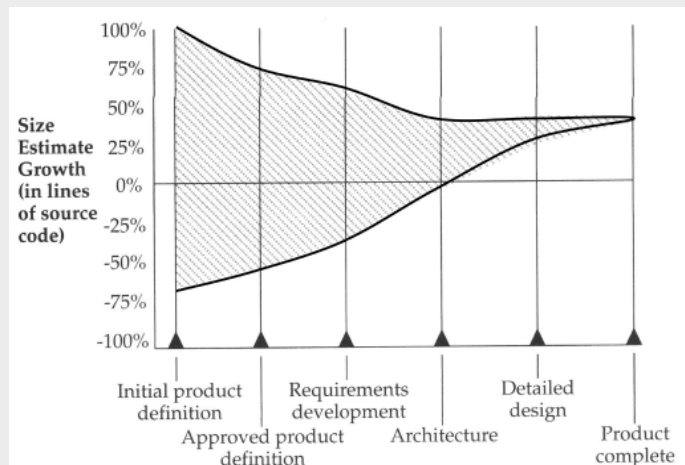
Wenig zielführend, aber weit verbreitet ist die Neigung, eine Schätzung entsprechend der angenommenen Erwartung, eventuell zuzüglich einer Sicherheitsmarge, abzugeben, und auch der Einsatz von Bauchwerten ist zwar sehr beliebt und üblich, aber nicht unbedingt Grundlage zutreffender Aufwandsschätzungen.

Wissenschaftlich fundiert

Da diese nun aber, wie gesagt, für Auftraggeber und Dienstleister immer wichtiger werden, werden wissenschaftlich fundierte Vorgehensweisen ins Kalkül gezogen, beispielsweise das LoC-basierte Verfahren (COConstructive COSt MOdel; COCOMO II) oder die Function-Point-Methode in verschiedenen Varianten: Der Aufwand ist aber stets gewaltig, und die Function Points beispielsweise sind im Vorfeld selten bekannt.

Es gibt weitere ganz praktische Probleme für realistische Schätzungen:

- Beispielsweise kann niemand sich selbst und die eigene Arbeitsleistung realistisch einschätzen.
- Als Schätzung für Kosten und Termin wird gerne die optimistischste Vorhersage genommen, deren Eintrittswahrscheinlichkeit nicht gleich null ist, bzw. der erstmögliche Termin, bei dem man nicht sicher beweisen kann, dass man nicht fertig wird (Kidder).



Schätzbreite, nach Steve McConnell.



- Schließlich ist die Ansicht weit verbreitet, aber nicht unbedingt zielführend, dass Schätzungen als Motivationsanreiz (Zielvorgabe) eingesetzt werden können bzw. sollen.

Vom Wie zum Wie lange

Tatsache ist: Aufwandsschätzungen für Softwareprojekte fallen notgedrungen praxisfern aus, solange nicht bekannt ist, wie die gestellten Anforderungen umgesetzt werden. Steve McConnell formuliert im "Software Project Survival Guide": "The cone of uncertainty implies that it is not only difficult to estimate a project accurately in the early stages, it is theoretically impossible." Und: "Early in the project you can have firm cost and schedule targets or a firm feature set, but not both."

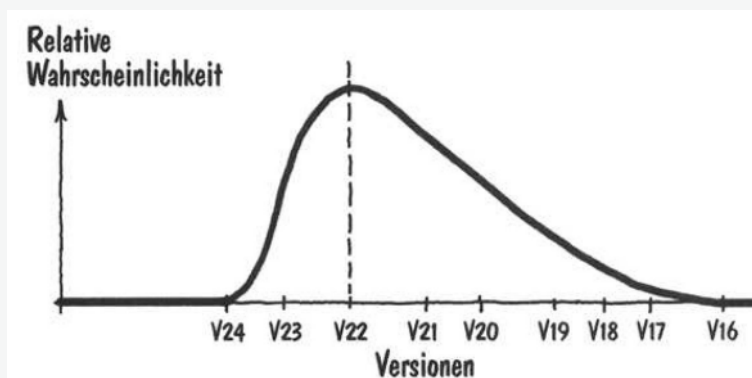
Besser sieht die Situation aus, wenn ein IT-Dienstleister seinen Kunden und seine fachlichen Bedarfe durch langjährige Zusammenarbeit kennt. Dann kann er über die Projekte hinweg Erfahrungswerte für die Realisierung einzelner Features sammeln und damit eine realistische Basis für Auftragsschätzungen aufbauen.

Zudem verfügen manche Großunternehmen über solche Erfahrungswerte aus den eigenen Projekten.

Fazit Nummer 1: Wer eine realistische Aufwandschätzung möchte, sollte entweder auf Erfahrungswerten aufbauen können oder aber auf ersten Überlegungen zur Realisierung von Anforderungen.

Unsicherheit quantifizieren

Wenn dies nicht gegeben ist, lässt sich auch Unsicherheit quantifizieren und damit Wissen darüber gewinnen, wie ver-



Umgehen mit Unsicherheiten: beispielsweise Termin fest, Umgang variabel (Tom DeMarco 2).

lässlich genau der eine ins Auge gefasste Termin bzw. ein Budget ist. Das heißt es lassen sich Termine schätzen, zu denen mit einer bestimmten Wahrscheinlichkeit ein Projekt keinesfalls, mit fünfzig Prozent Wahrscheinlichkeit bzw. auf jeden Fall fertiggestellt sein wird. Wie gesagt wird nun oft der früheste Termin, also der mit einer Fertigstellungswahrscheinlichkeit von beinahe null, genannt Nano-Prozent-Datum, genommen, statt dass das Unsicherheitsfenster konkretisiert wird: Welcher Zeitraum ist in Wahrheit noch akzeptabel? Welcher Umfang an Anforderungen ist mit Blick auf einen unaufschiebbaren Termin wirklich unverzichtbar?

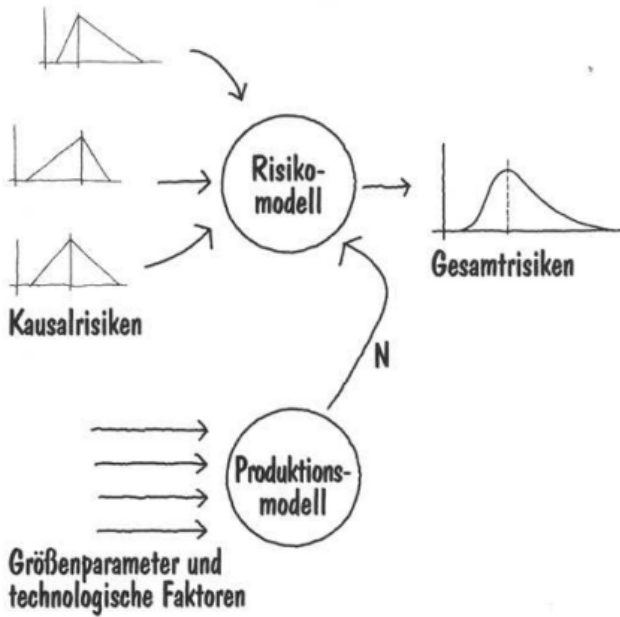
Dabei sollte auch gezielt nach all den Problemen und Schwierigkeiten gefragt werden, die bekanntermaßen auftauchen können, auch bzw. gerade weil die meisten Softwareprojektmanager recht gut darin sind, Aufgaben abzuschätzen, die erledigt werden müssen, und ziemlich schlecht darin, Aufgaben abzuschätzen, die möglicherweise erledigt werden müssen. (Tom DeMarco 2)

Fazit Nummer 2 lautet also: Statt wie bisher Termin und Ziel auf das Nano-Prozent-Datum zu legen, sollte der Termin nach dem Ziel und dieses nach dem Nanoprozentdatum gelegt werden.

Risiken berechnen

Natürlich lassen sich für ein Softwareprojekt Teil- und Gesamtrisiken erfassen. Die Kernrisiken sind bekannt, nämlich ein fehlerhafter Zeitplan, weil er gleichsam als Druckmittel verwendet wird und weil in jedem Projekt zusätzliche notwendige Arbeiten auftauchen, sich aber niemals Aufgaben, die am Anfang eines Projekts als nötig aufgelistet wurden, als unnötig erweisen.

Die Inflation der Anforderungen im Laufe des Projekts ist ein todsicherer Killer für das Projekt bzw. den Projekterfolg. Schwer vermeidbar und damit unbedingt zu berücksichtigen sind die Mitarbeiterfluktuation, ein Spezifikationskollaps und die Varianzbreite der Mitarbeiterproduktivität, wobei bekanntlich die Produktivität immer erst im Nachhinein gemessen bzw. verglichen werden kann. Und: Überzogene Projektzeit ist nicht gleichzusetzen mit geringer Produktivität!



Risikoabschätzung (Tom DeMarco 2).

Bekannt sind auch die branchentypischen Risiken für Entwicklungsprojekte; mit Hilfe von Excel lässt sich schon allein auf dieser Basis eine Schätzung zur Fertigstellungswahrscheinlichkeit erstellen und damit das Unsicherheitsfenster quantifizieren. (<http://www.systemguild.com/riskology/>). Überwunden werden müssen dazu allerdings auch noch Aspekte einer negativen Unternehmenskultur, beispielsweise die Haltung „Das schaffen wir schon (irgendwie)!“ oder „Failure is no option!“, das Abladen der Lösungsfindung auf den, der ein Problem aufzeigt, das Zurückstellen von Risiken angesichts einer spannenden technischen Herausforderung oder aber die Schuldzuweisung einer manipulierten Terminplanung, wenn jemand vorzeitig liefert, genauso wie die künftige Korrektur von Schätzungen entsprechend einer einmalig erzielten kürzeren Realisierungszeit.

Frühzeitig anfangen!

Wenn sich Risiken erfassen lassen, lassen sie sich auch vermeiden. Und hier liegen auch, **Fazit Nummer 3**, unsere Empfehlungen als IT-Dienstleister für Auftraggeber von Entwicklungsprojekten:

- Ein früher Anfang ist bei den meisten Pro-

jekten der einzig wirksame Weg, das Risiko einer verspäteten Fertigstellung zu begrenzen.

- Das heißt die Fachabteilungen sollten das Risiko eines frühzeitigen Projektbeginns eingehen, damit für die Softwareentwicklung angemessene Realisierungszeit bleibt.
- Wenn der Nutzen, den das Projekt bringen wird, groß genug ist, lohnt sich auch das Risiko (einer Terminüberschreitung).

Kritische Kette

Abhängig von der jeweiligen Unternehmenskultur werden für die einzelnen Vorgänge in Softwareentwicklungsprojekten bisweilen allzugroße Sicherheitsreserven einkalkuliert oder aber es wird eben dies unterstellt.

Trotzdem scheitern bekanntlich sehr viele IT-Projekte: Weil Zeitpuffer im Vorfeld ungenutzt verstreichen, weil verfügbare Zeiten eben gefüllt werden, weil IT-Fachleute oft in mehreren Projekten im Einsatz sind, so dass zwar Verzögerungen voll auf die Endtermine durchschlagen, Vorsprünge aber selten genutzt werden können.

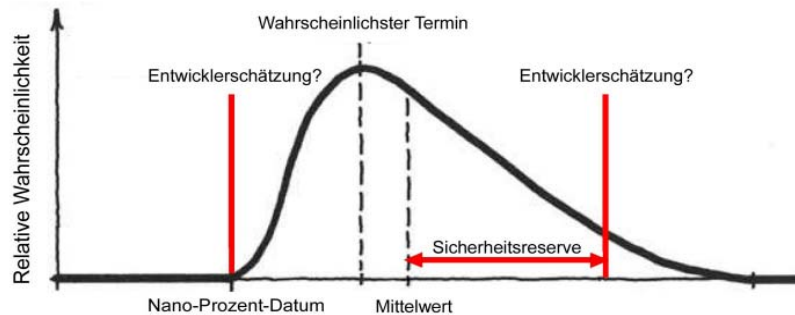
Wer hieraus Konsequenzen ziehen möchte, kann das Vorgehen nach der Critical Chain (Eliyahu Goldratt) in Betracht ziehen: Es werden keine Sicherheitsreserven für einzelne

Vorgänge eingebaut, nach der Staffellauf-Methode werden einzelne Aktivitäten abgearbeitet, Multitasking der Projektmitarbeiter wird unterlassen, dafür wird auf Zwischenpuffer, Projektpuffer und ein globales Puffer-Management gesetzt sowie auf eine Planung mit spätestmöglichem Anfang, so dass keine scheinbaren Zeitreserven verbummelt werden, statt für Verzögerungen zur Verfügung zu stehen.

Einen kleinen Haken hat das wenig aufwändige Vorgehen Kritische Kette nach Goldratt, das eine durchaus erfolgreiche Alternative zu Unsicherheits- und Risikokalkulationen ist: Sie stellt einen Paradigmenwechsel dar und verlangt eine entsprechende Unternehmenskultur. Und die Sicherheitsreserven müssen tatsächlich vorhanden sein, was wohl eher in bürokratisch organisierten Großprojekten als in Zwei-bis-drei-Mann-Teams der Fall ist.

Kommentare zur Aufgabe	ID	Aufgabenname	Dauer	17. Apr 2005							24. Apr 2005							1. Mai 2005							8. Mai 2005							15. Mai 2005						
				18	19	20	21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18				
Unsprüngliche Planung	1	Phase A	10t	[Bar chart showing Phase A from Apr 18 to Apr 28]														[Bar chart showing Phase B from Apr 28 to May 8]																				
	2	Phase B	10t	[Bar chart showing Phase A from Apr 18 to Apr 28]														[Bar chart showing Phase B from Apr 28 to May 8]																				
Verzögerungen	3	Phase A	12t	[Bar chart showing Phase A from Apr 18 to May 1]														[Bar chart showing Phase B from May 1 to May 11]																				
	4	Phase B	10t	[Bar chart showing Phase A from Apr 18 to Apr 28]														[Bar chart showing Phase B from Apr 28 to May 8]																				
Vorsprung Phase A	5	Phase A	8t	[Bar chart showing Phase A from Apr 18 to Apr 26]														[Bar chart showing Phase B from Apr 26 to May 5]																				
	6	Phase B	10t	[Bar chart showing Phase A from Apr 18 to Apr 28]														[Bar chart showing Phase B from Apr 28 to May 8]																				

Zeitliche Verzögerungen in Projekten schlagen sich fast immer im Projekterfolg nieder, während Zeitgewinne bei einzelnen Vorgängen kaum genutzt werden können.



Schätzunsicherheit: Schätzen Entwickler Reserven ein - die ungenutzt bleiben - oder tun sie das nicht? Diese entscheidende Frage wird sehr unterschiedlich beantwortet.

lich, bei komplexen Projekten schon in der Konzeptionsphase alle Aspekte zu berücksichtigen. Sind die Anwendung oder Teile davon verfügbar, dann sind - besonders wenn die Endanwender erst jetzt in das Projekt einbezogen werden - vielfache Änderungswünsche die Regel. Diese nachträglich zu berücksichtigen, erfordert internen oder externen Aufwand, so dass ein Budget und ein geordnetes Verfahren zur Behandlung von „Change Requests“ in jedem Projekt vorzusehen sind.

Schlussfazit

Aufwandsschätzungen für Entwicklungsprojekte sind erforderlich. Beide Seiten sollten aber mit realistischen Erwartungen an sie herangehen und dafür die bestmöglichen Grundlagen legen.

Festpreis und Budget

Nach Vorliegen der Schätzung und Bestimmung der Unsicherheiten kann ein Projektbudget bzw. ein Festpreis festgesetzt werden. Das intern eingeplante Budget des Auftraggebers muss aber höher als dieses Budget sein, da sich während der Laufzeit des Projekts oftmals die Anforderungen aufgrund von Ursachen außerhalb des Projekts, z.B. gesetzliche Änderungen oder Marktsituation, ändern. Zudem ist der „IKIWISI“-Effekt (I know it when I see it) zu berücksichtigen. Es ist nahezu unmög-

Ansprechpartner

Dr. Alexander Rickert, Leiter Technik
Tel. 089 42774-189 • E-Mail alexander.rickert@dmc-group.de

DMC Datenverarbeitungs- und Management-Consulting GmbH

Valentin-Linhof-Straße 8 • 81829 München
Tel. 089 42774-0 • Fax: 089 42774-199
E-Mail: dmc@dmc-group.de • www.dmc-group.de
Geschäftsführer: Angelo W. Zenz • Dr. Matthias Kulesa