

Projektplanung in komplexen Entwicklungsprojekten

Jörg Friedrich, INDAL, Münster

Zwei Ziele hat ein Entwicklungsprojekt: Ein Ergebnis, welches den Auftraggeber zufrieden stellt und die Einhaltung eines vorgegebenen Kosten- und Zeitplanes. Aber die Erfahrung fast jedes Projektmanagers zeigt, dass beides zusammen selten zu haben ist. Und das gilt nicht nur für Softwareprojekte, die gleichen schmerzhaften Erfahrungen machen. Werbefachleute genauso wie Produktentwickler in der Automobilindustrie.

Wenn Sie diese Seiten gelesen haben, werden Sie nicht nur wissen, warum das so sein muss. Sie werden nicht nur Verständnis für gestresste Projektverantwortliche in ehrgeizigen Entwicklungsprojekten bekommen, die sich immer wieder für Verzögerungen rechtfertigen müssen, die immer wieder nach neuen Mitteln rufen oder die am Schluss ein Produkt präsentieren, das keiner so recht haben will.

Sie werden beim Lesen eine Vorstellung davon bekommen, wie die althergebrachten Methoden der Projektplanung, die vor einem Jahrhundert entwickelt wurden und die heute noch immer den Kern fast aller Projektmanagementsoftware bestimmen, modifiziert werden müssen, damit Entwicklungsteams erfolgreich Produkte hervorbringen und sich dabei an einmal abgegebene Kosten- und Terminzusagen halten können.

Man muss sich zunächst vergegenwärtigen, auf welcher Grundlage Zeit- und Kostenberechnungen für Projekte entstehen. Die Basis dafür bilden heute wie vor 100 Jahren Netzpläne. Netzpläne beruhen auf dem Ansatz, dass sich ein Projekt in einzelne Vorgänge zerlegen lässt. Jedem dieser Vorgänge kann ein gewisser Ressourcenverbrauch (insbesondere benötigte Arbeitszeit) zugeordnet werden. Außerdem können für jeden Vorgang diejenigen anderen Vorgänge bestimmt werden, welche seine Vorgänger sein müssen, das heißt, welche stattgefunden haben müssen, damit dieser Vorgang beginnen kann.

Die grafische Darstellung der Vorgänge und ihrer (gerichteten) Beziehungen untereinander ist ein Netz, deshalb spricht man vom Netzplan. Das alles ist nicht neu, es wird in fast jedem Projektmanagementseminar unterrichtet. Was oft nicht gesagt, sondern stillschweigend vorausgesetzt wird: In diesem Netzplan dürfen keine Schleifen auftreten. Das heißt: unter keinen Umständen darf eine Kette von Vorgängen dazu führen, dass ein Vorgang, der in dieser Kette weit vorne steht, noch einmal zum Nachfolger wird.

An dieser Stelle der Theorie wirft der Leiter eines Entwicklungsprojektes jedes Lehrbuch in die Ecke und schaut es nie mehr an, ja, er bereut jede Minute, die er

mit dieser Theorie verbracht hat. Denn er weiß: Entweder er strukturiert und plant sein Projekt strikt nach einem Vorgehensmodell ohne Schleifen oder er bekommt ein vernünftiges Ergebnis. Beides zusammen scheint unmöglich zu sein.

Ein Beispiel: Blättert man in alten Lehrbüchern zu Softwareprojekten, findet man als Idealmodell des Softwareentwicklungsprozesses das Wasserfallmodell: Analyse, Design, Implementierung, Test, Installation sollen in wohlgeordneter Abfolge aufeinander folgen. Lange Zeit wurde jedes Abweichen von diesem Modell in realen Projekten als Fehler, als schlechte Planung angesehen: Musste man aus der Implementierung noch einmal in's Design wechseln, lag das am „schlechten“ Design, musste aufgrund von Tests noch Funktionalität implementiert werden, waren dies „Programmierfehler“.

Dabei weiß jeder projekterfahrene Entwickler, dass sich solche Rückschritte nicht vermeiden lassen: häufig können Anwender ihre Anforderungen erst richtig definieren, wenn sie die Software das erste Mal sehen. Re-Design und Neuimplementierungen nach ersten Tests ergeben sich zwangsläufig.

In den letzten Jahren ist man deshalb dazu übergegangen, den Anwender schon sehr früh in das Entwicklungsprojekt einzubeziehen. Prototyping und iterative Verfahren führen dazu, dass der künftige Anwender sich schon in sehr frühen Entwicklungsstadien ein Bild von der zukünftigen Software machen kann. Er kann seine Anforderungen präzisieren, Missverständnisse können beseitigt werden.

Auf diese Weise entstehen Ergebnisse, die sich bei den Anwendern hoher Akzeptanz erfreuen, allerdings sind solche Projekte kaum planbar hinsichtlich Kosten und Terminen. Denn wer weiß schon, wie viele Iterationen gebraucht werden, bis der Anwender zufrieden ist?

Auf das klassische Projektmanagement ist eine solche Struktur nicht abbildbar, somit kann kein Netzplan erstellt, kann kein kritischer Pfad berechnet werden .

Was ist zu tun?

Zunächst muss der einzelne Prozess untersucht werden. Als Beispiel möge hier der die Entwicklung einer einfachen GUI-Komponente dienen. Dieser Prozess besteht aus Elementarschritten, welche iterativ so lange wiederholt werden, bis der Anwender zufrieden ist. Jeder Elementarschritt besteht aus der Programmierung, dem Test durch den Anwender und der Diskussion der Testergebnisse, die in der nächsten Iteration zu berücksichtigen sind. Bei der Untersuchung solcher Prozesse stößt man auf charakteristische Aufwandsprofile, wie sie in Abbildung 1 a und b dargestellt sind.

Abbildung 1 a: Aufwandsverlauf mit initialem Prototypen



Abbildung 1 b: Aufwandsverlauf ohne Prototypentwicklung



Aus diesen charakteristischen Aufwandsprofilen lassen sich vier Größen ableiten, mit denen sich die meisten elementaren Entwicklungsprozesse beschreiben lassen:

- Aufwand für den Hauptschritt A_H
- Faktor für die Abnahme des Aufwandes von Schritt zu Schritt a_f
- Faktor für das Verhältnis des Aufwandes für den Prototypen A_P zu A_H : a_p
- Zahl der Schritte

Ziel der folgenden Überlegungen ist es, zu zeigen, wie diese Größen geschätzt oder erfasst werden können, wie daraus Kostenberechnungen abgeleitet werden

und wie es schließlich möglich ist, ein Modell des Projektes zu erstellen, mit welchem Terminplanungen sowie eine Planung des Ressourceneinsatzes möglich sind.

Zunächst soll weiter der Einzelprozess, wie z.B. die Entwicklung einer einzelnen GUI-Komponente betrachtet werden. Für diesen Prozess ergibt sich der Aufwand als

$$A_G = A_H \cdot (a_p + \sum_{i=0}^n a_f^i)$$

wobei $i=0$ für den Hauptschritt und n die Zahl der Folgeschritte ist. Der (geschätzte) Aufwand für den Gesamtprozess ist also mit einem Faktor a_g zu multiplizieren, für den gilt:

$$a_g = a_p + \sum_{i=0}^n a_f^i$$

Im Allgemeinen ist es erfahrenen Entwicklern möglich, die Dauer des Hauptschrittes aus den Anforderungen, wie sie aus dem Lastenheft oder einem Pflichtenheft hervorgehen, zu schätzen. Die Frage, die der Entwickler zu beantworten hat, lautet: „Wie groß ist der Aufwand in Personentagen, um eine erste testreife Version der Komponente bereitzustellen?“ zu dieser Schätzung sind die Aufwendungen für die Tests durch die Anwender sowie Kommunikationsaufwendungen hinzuzurechnen.

Schwieriger stellt sich die Herausforderung dar, die übrigen Größen zu bestimmen. Letztlich sollten diese Zahlen aus einer darauf abgestimmten Projektzeiterfassung ermittelt werden. Solange dies nicht möglich ist, muss mit Schätzungen gearbeitet werden.

Empirische Untersuchungen haben gezeigt, dass bei betriebswirtschaftlichen Entwicklungsprojekten etwa folgende Größen angesetzt werden können:

Prozesstyp	a_p	a_f	n	a_g
Datenbankdesign Bereich Bewegungsdaten	0,0	0,3	3	1,4
Datenbankdesign Bereich Stammdaten	0,0	0,2	2	1,2
Datenbankdesign Bereich Basiscode-Tabellen	0,0	0,1	1	1,1
GUI-Komponenten Bereich Bewegungsdaten	0,3	0,7	5	3,2
GUI-Komponenten Bereich Stammdaten	0,1	0,2	2	1,3
GUI-Komponenten Bereich Basiscode-Tabellen	0,0	0,1	1	1,1
Auswertungen/Reports	0,3	0,6	5	2,7
Berechnungsalgorithmen	0,1	0,5	3	2,0

Allerdings sind diese Zahlen stark von der Projektgröße und von der Teamstruktur abhängig. Sie müssen unbedingt durch eine Vor-Ort-Istdatenerfassung verifiziert werden.

Dazu muss das Projekt in Prozesse zerlegt werden, ähnlich wie man es vom klassischen Projektmanagementansatz kennt. Die Projektmitarbeiter erfassen täglich ihre Arbeitszeiten zu den Prozessen. Dazu muss erfasst werden, an welchem Iterationsschritt gerade gearbeitet wird.

Allerdings ist es eine Illusion zu hoffen, dass ein Entwickler oder ein Designer oder auch ein Tester praktisch in jedem Moment sagen könnte, in welchem Iterationsschritt er sich gerade befindet. Deshalb muss das Zeiterfassungssystem so konfiguriert werden, dass die Zuordnung zu einem Iterationsschritt quasi intern erfolgt. Das ist auf unterschiedliche Weise möglich:

1. Es werden Meilensteine festgelegt, die im System erfasst werden müssen (z.B. Meetings, in denen Testergebnisse kommuniziert werden, Versionsübergabe, Versionisierung in der Quelltextverwaltung). Das Auftreten eines solchen Meilensteins wird als Übergang zu einer neuen Iteration definiert.
2. Es werden Tätigkeitsmuster definiert, die einen Übergang von einer Iteration zur nächsten charakterisieren (z.B. Test-Implementierung)

Wichtig ist, dass die für den jeweiligen Projekttyp charakteristischen Prozesstypen identifiziert werden. Aus dem so gesammelten empirischen Material können dann mit mathematischen Mitteln die obigen Parameter ermittelt werden.

Auf dieser Basis werden nun Vorgangsprofile definiert. Dabei werden die Prozesstypen mit den ermittelten Parametern verknüpft.

Abbildung 2: Erstellen von Vorgangsprofilen

Stammdatenpflege für Tabelle PR_VORGANGSPROFIL

Bezeichnung: GUI Komponente Bewegungsdaten

Dauer Hauptschritt: []

Prozentuale Dauer Prototyp: 0,300

Prozentuale Dauer Folgeschritt: 0,700

Zahl der Schritte: 7

Letzter Schritt bis Projektende:

Neuer Datensatz

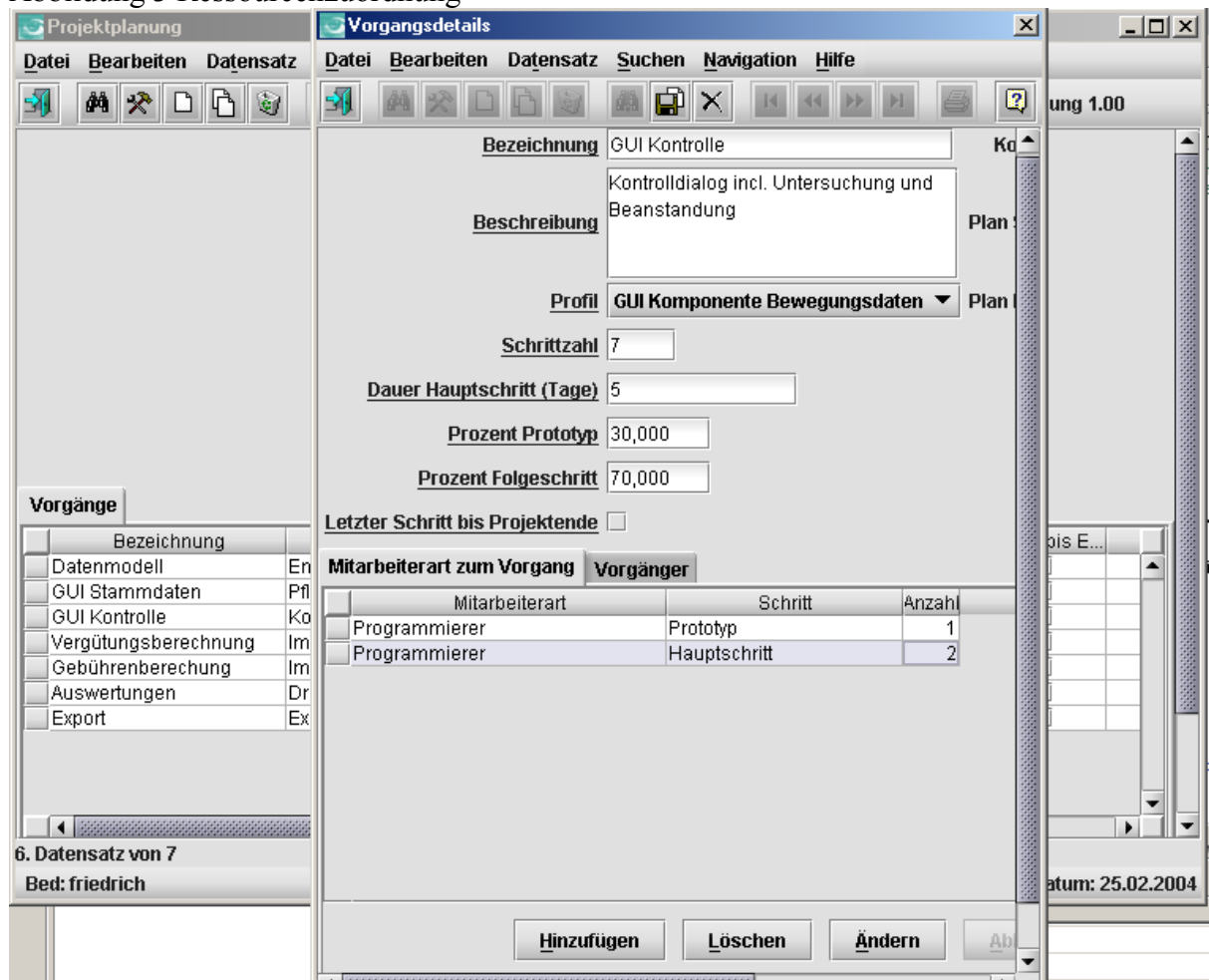
Bed: friedrich

Datum: 25.02.2004

Bei der Planung eines neuen Projektes kann nun jeder identifizierte Prozess mit einem Vorgangsprofil versehen werden. Aus der Schätzung des Aufwandes für den Hauptschritt ergibt sich damit eine plausible Schätzung des Aufwandes für den Gesamtprozess.

Obwohl dies schon einen entscheidenden Fortschritt gegenüber dem klassischen Ansatz darstellt, kann man noch wesentlich weiter gehen. Die Zerlegung in die einzelnen prognostizierten Iterationen erlaubt eine genaue Zuordnung der Ressourcen zu den einzelnen Iterationen.

Abbildung 3 Ressourcenzuordnung



Damit wird die Tür zu einer genauen Ressourcenplanung geöffnet. Darüber hinaus ist eine Budgetplanung möglich, falls Kostensätze für die Ressourcen verfügbar sind. Das ist jedoch ein anderes Thema.

Um eine wirkliche Ressourcenplanung gewährleisten zu können, müssen die Verknüpfungen zwischen den Prozessen betrachtet werden. Auch hier eröffnet der

Ansatz, den einzelnen Prozess als eine Kette von Iterationen zu betrachten, bedeutende Chancen.

Praktisch ist es häufig so, dass ein Folgeprozess bereits beginnen kann, wenn ein Vorgängerprozess ein bestimmtes Stadium erreicht hat. Betrachtet man das hier entwickelte Modell eines Prozesses, ergeben sich ganz natürlich zwei praktisch relevante Fälle:

- Ein Folgeprozess kann beginnen, wenn der Vorgängerprozess seine Prototypphase beendet hat
- Ein Folgeprozess kann beginnen, wenn der Vorgängerprozess seinen Hauptschritt beendet hat

Abbildung 4: Definition von Abhängigkeiten zwischen den Prozessen

Mitarbeiterart zum Vorgang	Vorgänger	Nach Schritt
<input type="checkbox"/>	Datenmodell	Hauptschritt
<input type="checkbox"/>	GUI Stammdaten	1. Folgeschritt
<input type="checkbox"/>	Vergütungsberechnung	Prototyp
<input type="checkbox"/>	Gebührenberechnung	Prototyp

Hinzufügen Löschen Ändern Abbrechen Ok

Die Hilfskonstruktion der „Anfang-Anfang-Beziehung mit zeitlicher Verzögerung“ aus dem klassischen Projektmanagement kann also in eine natürliche Ende-Anfang-Beziehung zwischen Iterationen aufgelöst werden.

Genau genommen gibt es also quasi eine Mikro-Ebene des Projektes: die der einzelnen Iterationen. Auf der Makro-Ebene findet man die Prozesse oder Vorgänge, welche auf einfache Weise gemäß des obigen Modells beschrieben werden können. Die Konsequenz aus dem gesagten lautet:

Auf der Makroebene wird das Projekt geplant, indem Prozesse definiert werden und bestimmten Vorgangsprofilen zugeordnet werden.
Auf der Mikroebene der Iterationen werden Termine, Ressourcenplanungen und Kostenschätzungen berechnet.

Der dargestellte Ansatz gibt dem Projektmanager also die Möglichkeit, sein Entwicklungsprojekt so zu strukturieren, dass ein erfolgreicher Projektabschluss in jedem Sinne möglich wird:

1. Durch iteratives Vorgehen werden alle Beteiligten frühzeitig einbezogen, Fehlentwicklungen und Missverständnisse vermieden
2. Durch das entsprechende Prozessmodell wird eine zuverlässige Kosten- und Ressourcenplanung genauso möglich wie ein stabiler Terminplan.

Kontakt für weitere Informationen:
INDAL OHG, Seminarzentrum und Softwarehaus
Jörg Friedrich, Geschäftsführer
Johann-Krane-Weg 37
48149 Münster

Tel: 0251 / 41 44 6-0
EMail: jf@indal.de
<http://www.indal.de>