



Jörg Friedrich

Agile Teams

Effektives Aufgaben- und Projektmanagement



Der Autor hat an der Ruhr-Universität Bochum Elektrotechnik studiert und von dort einen Abschluss als Diplom-Ingenieur. Er studierte an der University of Florida, Gainesville, FL, U.S.A., und promovierte an der Purdue University, West-Lafayette, IN, U.S.A. über ein Thema der Halbleiterphysik.

Seine industrielle Laufbahn begann er bei der ANT in Backnang, wo er verantwortlich für die Entwicklung digitaler Filter und Kodierverfahren für Richtfunkgeräte war. Später führte er das Team für digitale Signalverarbeitung bei der Bosch Telecom in Backnang. Er gründete dort die Gruppe für eingebettete Software und leitete später die Abteilung für Softwareentwicklung im Bereich Richtfunkgeräte.

Im Jahr 2003 wurde Jörg Friedrich als Professor an die Hochschule Esslingen berufen. Als Mitglied der Fakultät Informationstechnik ist er verantwortlich für die Lehre und Forschung im Bereich eingebetteter Systeme und Echtzeitsysteme. Als Koautor hat er ein Buch über die Programmierung von Echtzeitsystemen verfasst.

Dr. Friedrich hat viele Firmen, die softwareintensive Systeme herstellen, bei Prozessverbesserungsaktivitäten beraten. Aufgrund seiner industriellen Praxis und der Beratungstätigkeit besitzt er umfangreiche Erfahrungen in der Prozessanalyse, im Anforderungsmanagement und im Design softwareintensiver Systeme.

Dr. Friedrich gründete 2005 das Steinbeis-Transferzentrum Task Management Systems und leitet dieses bis heute. Er ist der führende Kopf hinter der Entwicklung von Trackplus, einem Werkzeug für Projekt- und Aufgabenmanagement.

Jörg Friedrich

Agile Teams

Effektives Aufgaben- und Projektmanagement



Steinbeis-Transferzentrum
Task Management Systems

Impressum

©2012 Steinbeis-Edition

Alle Rechte der Verbreitung, auch durch Film, Funk und Fernsehen, fotomechanische Wiedergabe, Tonträger jeder Art, auszugsweisen Nachdruck oder Einspeicherung und Rückgewinnung in Datenverarbeitungsanlagen aller Art, sind vorbehalten.

Jörg Friedrich
Agile Teams. Effektives Aufgaben- und Projektmanagement

1. Auflage 2012 | Steinbeis-Edition, Stuttgart
ISBN 978-3-943356-39-7

Satz: Jörg Friedrich
Titelbild: Getty Images 90620129, Hugh Sitton
Druck: Übelmesser Druck, Stuttgart

Steinbeis ist weltweit im Wissens- und Technologietransfer aktiv. Zum Steinbeis-Verbund gehören derzeit rund 850 Steinbeis-Unternehmen sowie Kooperations- und Projektpartner in 50 Ländern. Das Dienstleistungsportfolio der fachlich spezialisierten Steinbeis-Unternehmen im Verbund umfasst Beratung, Forschung & Entwicklung, Aus- und Weiterbildung sowie Analysen & Expertisen für alle Management- und Technologiefelder. Ihren Sitz haben Sie überwiegend an Forschungseinrichtungen, Universitäten und Hochschulen.

Dach des Steinbeis-Verbundes ist die 1971 ins Leben gerufene Steinbeis-Stiftung, die ihren Sitz in Stuttgart hat. Die Steinbeis-Edition verlegt ausgewählte Themen aus dem Steinbeis-Verbund.

157416-2012-11 | www.steinbeis-edition.de | www.trackplus.de

Vorwort

Was ist zu tun? Wer kümmert sich darum? Aufgaben- und Vorgangsmanagement ist das kleine Einmaleins des Projektmanagements. Wenn Sie schon Erfahrungen im klassischen Projektmanagement gesammelt haben und sich fragen, warum Planungen so schwierig sind und es in der Durchführungsphase immer wieder zu Reibungsverlusten, Intransparenz und Kommunikationsstörungen kommt, dann gibt Ihnen dieses Buch klare Hinweise, wie Sie derartige Probleme in Zukunft vermeiden oder zumindest deutlich reduzieren können.

Sie werden undogmatisch an bewährte Prinzipien des Aufgaben- und Projektmanagements herangeführt. Welche Möglichkeiten gibt es, zuverlässig zu planen, und wo sind die Grenzen? Wie detailliert soll geplant werden? Wie beherrschen Sie komplexe Projekte mit vielen tausend Aufgaben? Wie stellen Sie einen angemessenen Informationsfluss sicher, ohne Informationsflut und ohne Informationsdefizite? Welche guten Prinzipien stecken hinter agilen Methoden wie Scrum und Kanban?

Aufgaben- und Vorgangsmanagement gibt es auch außerhalb von Projekten, z. B. als Teil eines Produkt-Supports oder im Rahmen von sich häufig wiederholenden Arbeitsabläufen. Beispielszenarien zeigen Ihnen, wie Produktentwicklung und Produkt-Support über das Aufgaben- und Vorgangsmanagement verknüpft werden können.

Alle vorgestellten Methoden sind praxiserprobt. Die Beispielszenarien basieren auf tatsächlichen Anwendungsfällen aus den Bereichen Hardware- und Softwareentwicklung, Maschinenbau, Automatisierungstechnik, Automobilbau und der Luftfahrtindustrie.

Jörg Friedrich

Danksagung

Ich danke meinen Mitarbeitern, vor allem Tamas Ruff und Adrian Bojani, für viele hilfreiche Diskussionen. Dankbar bin ich auch den Menschen, die ich in Firmen, auf Konferenzen und in meiner Beratungstätigkeit kennen gelernt habe und die mir geholfen haben, die Herausforderungen des Aufgaben- und Projektmanagements in den unterschiedlichsten Anwendungsbereichen zu verstehen.

Nicht zuletzt möchte ich meiner Frau Bärbel sowie meinen Kindern Kathrin, Babette und Christoph für ihre liebevolle Unterstützung und Geduld danken. Sie bringen viel Verständnis auf, wenn ich Zeit für das Schreiben von Artikeln, Büchern oder Software verwende, die eigentlich ihnen gehören würde.

Jörg Friedrich

Inhaltsverzeichnis

Teil I – Projekt- und Aufgabenmanagement	13
1 Herrschaft und Überblick	15
1.1 Den Überblick behalten	15
1.2 Aufgaben, Vorgänge und Projekte.	16
1.3 Beobachten und Steuern	17
1.4 Und was habe ich davon?.	19
1.5 Woran scheitert es?.	20
1.6 Der Star unter den PM-Werkzeugen	23
1.7 Müssen Sie gläubig sein?	24
2 Planung	27
2.1 Das Planungsproblem	27
2.2 Divide et impera	29
2.3 Aufwandsschätzung.	30
2.4 Frohe Botschaften	31
2.5 Planen oder nicht planen	32
2.6 Wie detailliert darf es denn sein?	33
2.7 Planen mit Zwischenzielen	34
2.8 Bottom-up und Top-down	37
2.9 Sortierung von neuen Leistungsmerkmalen	38
2.10 Abschätzungen mit Planungspoker	39
3 Organisation und Ausführung	41
3.1 Aufgaben, Vorgänge und Projekte.	41
3.2 Vorgangsmarkierungen und Attribute	45
3.3 Eingangskorb	47
3.4 Nächste Aktionen	48
3.5 Geplante Vorgänge.	48
3.6 Kalender	49
3.7 Delegiert/wartend	50
3.8 Abfall.	50

3.9	Brutkasten	50
3.10	Referenz	52
3.11	Vorgangsverknüpfungen	52
3.12	Flusskontrolle	53
4	Zugangskontrolle	55
4.1	Räume	55
4.2	Phasen	57
4.3	Räume und Ablagen	58
4.4	RACI-Rollen und die RACI-Matrix	59
5	Kommunikation	61
5.1	Babel und die Folgen	61
5.2	Informationsaustausch	62
5.3	Rollenbasierter Informationsfluss	63
6	Beobachten und steuern	65
6.1	Berichte	65
6.2	Die Leistungswertmethode (Earned Value)	66
6.3	Aktueller Projektstand	68
6.4	Projektfortschritt und Trend	68
6.5	Ressourcenauslastung	69
6.6	Hotspots im Projekt	69
6.7	Aufwände und Abrechnungen	72
6.8	Prozessperformanz	73
Teil II – Anwendungsszenarien		75
7	Szenario-Grundlagen	77
7.1	Übersicht	77
7.2	Allgemein gültige Prinzipien	77
7.3	Gemeinsame Vorgangsattribute	78
8	Issue tracking	81
8.1	Prinzip	81
8.2	Vorgangstypen	82
8.3	Prozess	83

8.4	Vorgangsattribute	84
8.5	Releases	85
8.6	Verknüpfungen	85
8.7	Berichte und Auswertungen	86
9	Klassisches Projektmanagement	89
9.1	Übersicht.	89
9.2	Vorgangstypen	90
9.3	Prozess	91
9.4	Vorgangsattribute	93
9.5	Releases	93
9.6	Verknüpfungen	94
9.7	Berichte und Auswertungen	95
10	Scrum	99
10.1	Übersicht.	99
10.2	Prozess-Übersicht	100
10.3	Sprint.	102
10.4	Rollen	103
10.5	Backlogs verwalten	104
10.6	Vorgangstypen	105
10.7	Vorgangsattribute	105
10.8	Verknüpfungen	107
10.9	Berichte und Auswertungen	107
10.10	Die Essenz	109
11	Kanban	111
11.1	Übersicht.	111
11.2	Vorgangsattribute	112
11.3	Berichte und Auswertungen	113
12	Technischer Support	115
12.1	Übersicht.	115
12.2	Vorgangstypen	115
12.3	Prozess	116
12.4	Vorgangsattribute	118
12.5	Releases	119
12.6	Verknüpfungen	120

12.7	Berichte und Auswertungen	120
13	Mini-Projekte	123
13.1	Übersicht.	123
	Literaturverzeichnis	126
	Abkürzungen	130
	Index	132

Abbildungsverzeichnis

1.1	Elemente des Projektmanagements	16
1.2	Aufgaben- und Projektmanagement	17
1.3	Herausforderungen an das Management	18
1.4	Eine Methode erfolgreich machen	21
2.1	Release und Phase	35
2.2	Berechnung der zugewiesenen Kapazität je Teammitglied	36
3.1	Beispiele für Vorgangstypen	43
3.2	Das Workflow-Modell in der Übersicht	43
3.3	Notwendige und optionale Vorgangsmarkierungen und Attribute	46
3.4	Die Zeitmanagement-Matrix	51
4.1	Vorgänge in Räumen	56
4.2	Teilprojekte für Komponenten	57
4.3	Beziehung zwischen Eingangskorb und Räumen	58
4.4	Das RACI-Rollenschema	59
4.5	Ein Beispiel für eine RACI-Matrix	60
5.1	Informationsklassifikation	63
6.1	Analyse basierend auf Leistungswertmethode	67
6.2	Meilenstein-Trendanalyse	69
6.3	Burndown-Chart	70
6.4	Anzahl Vorgänge je Zustand über der Zeit	70
6.5	Geplante Auslastung von Ressourcen über der Zeit, graphisch	71
6.6	Geplante Auslastung von Ressourcen über der Zeit, tabellarisch	71
6.7	Mehrfach wiedereröffnete Vorgänge	72

6.8	Durchschnittliche Zeit bis zur Schließung über der Zeit . .	73
6.9	Durchschnittliche Antwortzeit über der Zeit	74
8.1	Geöffnete vs. geschlossene Vorgänge	86
8.2	Durchschnittliche Zeit bis zur Erledigung	86
9.1	Wasserfall-Modell	92
9.2	Earned-Value-Diagramm zur Fortschrittsbeobachtung. . .	95
9.3	Meilenstein-Trendanalyse	96
9.4	Budget-Übersicht	96
10.1	Scrum-Backlogs	100
10.2	Scrum Sprint	103
10.3	Scrum-Burndown-Diagramm	108
10.4	Scrum-Tafel	108
11.1	Kanban-Tafel	112

Teil I

Projekt- und Aufgabenmanagement

1

Herrschaft und Überblick

Denn von anderer Geburt bin ich als sie und aus anderem Holz und habe einen allgemeinen Kopf, darum bin ich Vorsteher geworden. Die Werkenden in den Betrieben fragen dich nicht, was du kannst, sondern wer du bist, denn ein ander Können ist es, das damit verbunden, und zur Aufsicht ist es geschaffen. Wer da reden kann vor dem Herrn wie du, und wem sich derart die feinen Gedanken fügen nebst ihren Worten, der soll nicht gebückt sitzen übers einzelne, sondern soll zwischendurch wandeln an meiner Seite. Denn im Worte und nicht in der Hand ist Herrschaft und Überblick.

aus Joseph und seine Brüder, Thomas Mann

1.1 Den Überblick behalten

In „Joseph und seine Brüder“ [Ma36] beschreibt Thomas Mann, wie der Meier des Potiphar den Joseph in die Geschäfte eines Verwalters einführt und „den Überblick lehrt“. Die Ägypter führen über alles akribisch Listen, und Joseph macht sich diese Methode schnell zunutze. Hat er am Anfang noch Bedenken, weil er nicht töpfern, schustern, kleben und Stühle herstellen kann, so lernt er, dass dies zum Verwalten auch gar nicht notwendig ist. Sorgfältig lässt er alles eintragen, was geerntet, produziert, verkauft und verbraucht worden ist. Über regelmäßig aktualisierte Inventarverzeichnisse kontrolliert er die Bediensteten Potiphars und wenn jemand auffällt, wird ihm angedroht, dass er „in die Asche kommt“.

Sie sollten heute niemandem mehr ankündigen, ihn „in die Asche zu bringen“. Sie sollten heute aber immer noch den Überblick behalten und das ist durch die immaterielle Natur vieler Arbeitsergebnisse schwieriger als im alten Ägypten. Joseph wusste, was ein Schuh ist, aber nicht jeder Projektmanager kann sich unter den Arbeitsergebnissen seiner Mitarbeiter immer etwas vorstellen.

Als Projektmanager spüren Sie vielleicht, dass Ihre Planungen schnell veralten. Trotz vieler Besprechungen fällt es Ihnen schwer, den Projektfortschritt objektiv festzustellen. Manchmal haben Sie das Gefühl, Sie steuern Ihr Team mit verbundenen Augen. Ihr Team arbeitet fleißig, aber Sie haben keine Vorstellung vom tatsächlichen Stand.

Wenn Ihnen diese Situationen bekannt vorkommen, lohnt es sich für Sie, weiterzulesen. Die beschriebenen Methoden helfen Ihnen, Ihre Projekte klar zu planen, zu steuern und mit hoher Aktualität zu wissen, wo Sie stehen.

1.2 Aufgaben, Vorgänge und Projekte

Die wichtigsten Aktivitätsfelder im Projektmanagement (PM) sind in Abb. 1.1 dargestellt. In dieser Publikation konzentrieren wir uns auf die

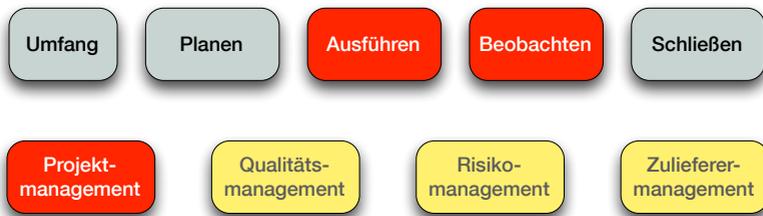


Abbildung 1.1: Elemente des Projektmanagements.

Bereiche, die in Projekten den größten Aufwand erfordern und in jedem Fall notwendig sind. Das sind die Felder

- Planung mit Aufwandsschätzung
- Beobachtung und Steuerung des Projekts

Diese Aktivitäten sind entscheidend für den Projekterfolg. Die anderen Bereiche haben ebenfalls ihre Bedeutung, sind jedoch mehr unterstützender Natur.

In diesem Buch beschränken wir uns nicht auf Projektmanagement. Auch außerhalb von Projekten fallen Aufgaben und Vorgänge an, die gesteuert werden müssen. Dazu gehören z. B. Aufgaben im Rahmen des Produkt-Supports oder einfache Handlungen und Prozesse, deren struktureller Zusammenhang zu flexibel ist, um sie mit vertretbarem Aufwand in eine standardisierte Unternehmenssoftware zu integrieren. Abb. 1.2 illustriert den Zusammenhang zwischen Aufgabenmanagement, Projektmanagement, Produkt-Support und Workflows. Aufgabenmanagement ist

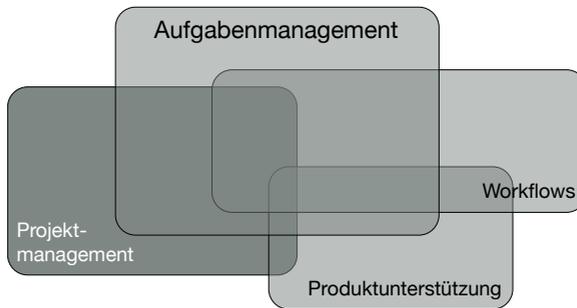


Abbildung 1.2: Aufgaben- und Projektmanagement.

ein wichtiger Teil des Projektmanagements, erstreckt sich jedoch darüber hinaus. Bei den in Abb. 1.2 gezeigten Workflows handelt es sich um Teilprozesse, die nicht statisch strukturiert werden können, sondern ein hohes Maß an Disposition erfordern. Es geht also nicht um die Bearbeitung von Urlaubsanträgen oder Erstellung von Standardangeboten, sondern z. B. um die Fehlersuche aufgrund einer Kunden-Problemmeldung.

1.3 Beobachten und Steuern

Die Beobachtung und Steuerung eines Projekts stellt Sie als Projektleiter vor große Herausforderungen. Die größte Anzahl von Aktivitäten liegt in der Ausführungsphase eines Projekts. Hier stellt sich heraus, wie gut die dem Plan zugrunde liegenden Annahmen waren. Nur durch ein kon-

sequentes Aufgabenmanagement können Sie die Planung umsetzen. Bei der Ausführung stellen Sie fest, dass Ihr Plan nicht statisch sein kann und Sie jetzt mit vielen Änderungen zurecht kommen müssen.

Was macht die Beobachtung und das Steuern eines Projektes so schwierig? Betrachten wir einmal die Anforderungen:

- Sie müssen in der Lage sein, aktivitätsorientiert und agil auf sich ändernde Anforderungen zu reagieren.
- Sie müssen u. U. geographisch verteilte Teams koordinieren.
- Sie müssen u. U. Teams von mehreren Organisationen oder Firmen koordinieren.
- Sie müssen hohe Komplexität aufgrund stark vernetzter Teams und Projekte beherrschen können.
- Sie müssen in der Lage sein, auf Termindruck angemessen zu reagieren.



Abbildung 1.3: Herausforderungen an das Management.

Was brauchen Sie unter solchen Randbedingungen, damit Sie Ihre Projekte gut steuern können?

- Sie müssen jederzeit bestens informiert sein, insbesondere bei Änderungen.
- Sie müssen klar kommunizieren, was Sie von jedem Teammitglied erwarten.

- Sie müssen Engpässe und kritische Pfade schnell erkennen können.
- Sie müssen den Projektfortschritt in Echtzeit beobachten können.

Es genügt nicht, wöchentliche oder monatliche Besprechungen abzuhalten, wo Ihnen jeder mal erzählt, wie es bei ihm so aussieht. Oft ist bis kurz vor dem Liefertermin alles Bestens. Die Profis warten mit den schlechten Nachrichten solange, bis jemand anderes seinen Termin nicht einhalten kann. Dann können sie sich dahinter verstecken.

Es genügt nicht, dass Sie ganz persönlich Ihren Kopf für einen revidierten Projektplan hinhalten, der auf mehr oder weniger verbindlichen Aussagen des Teams in solchen Besprechungen fußt.

Es genügt nicht, eine Excel-Liste zu erstellen mit Aufgaben, die jeder mitnimmt und bis zur nächsten Besprechung zu erledigen hat.

Das bringt uns zu einem Grundgesetz des Projektmanagements:



Das Fundament für ein gutes Projektmanagement ist ein funktionierendes Aufgabenmanagement. Ohne systematisches Aufgabenmanagement bleibt Projekterfolg zufällig.

Wir werden uns später damit beschäftigen, wie wir das gesamte Team in das Aufgaben- und Projektmanagement mit einbeziehen, so dass der Fortschritt jederzeit objektiv erkennbar ist, dass jeder weiß, was er bis wann zu erledigen hat, und alle relevanten Informationen umgehend an genau den Personenkreis verteilt werden, der sie braucht.

1.4 Und was habe ich davon?

Als Projektleiter sind Sie abhängig von Ihrem Projektteam. Sie können sich die schönsten Methoden ausdenken und einführen, doch um Erfolg zu haben, müssen Sie Ihren Teammitgliedern eigene Vorteile in Aussicht stellen. Das wusste schon der Prophet Mohammed:

Die Gottesfürchtigen (dagegen) haben (großes) Glück (mafaaz) zu erwarten, Gärten und Weinstöcke, gleichaltrige (Huris) mit schwelenden Brüsten und einen Becher (mit Wein, bis an den Rand) gefüllt (dihaaq).

Koran Sure 78,31-33

Was wünscht sich Ihr Team? Aus Sicht der Teammitglieder gibt es hier und da unsinnige Vorgaben, Entscheidungsstaus und unzureichende Kommunikation. Unklare und häufig wechselnde Aufgabenstellungen machen dem Team das Leben schwer. Die Vorgesetzten verdrängen, dass die Mitarbeiter schon voll ausgelastet sind und nehmen deren Abschätzungen nicht wirklich ernst.

Wir schmunzeln über die Dilbert-Kartoons, weil in ihnen so viel Wahrheit steckt. Was wünschen sich denn Ihre Teammitglieder?

- Sie möchten gut informiert sein, insbesondere wenn sich etwas ändert.
- Sie möchten wissen, was Ihre Vorgesetzten von Ihnen erwarten.
- Sie möchten objektiv darlegen können, wenn sie überlastet sind.
- Sie möchten darstellen können, was sie schon erreicht haben.

Wenn Sie Methoden anbieten, die diese Punkte ansprechen, haben Sie Ihr Team mit ins Boot genommen. Wenn nur Sie selbst gut informiert sind, oder nur Sie selbst darlegen können, wie effektiv Sie gearbeitet haben, wird Ihr Team nicht mitmachen.

1.5 Woran scheitert es?

Es gibt eine große Menge an Informationen zum Thema Projektmanagement. Viele Werkzeuge sind entwickelt worden, die unterschiedliche Aspekte dieses Themenbereiches ansprechen. Wenn Sie Teil einer großen Organisation sind, haben Sie vielleicht schon Versuche miterlebt, große, mächtige und teure Systeme einzuführen. In der Praxis stellt sich jedoch heraus, dass eine moderne Variante der „ägyptischen Methode“ alle anderen dominiert: Listen mit offenen Punkten werden in einer Tabelle mit Hilfe einer Tabellenkalkulationssoftware wie Excel verwaltet.

Warum werden die anderen Methoden und leistungsfähigeren Werkzeuge von den Mitarbeitern so selten angenommen?

Betrachten wir einmal ein bekanntes Werk über Projektmanagement [Pmb]: Es hat einen Umfang von 459 Seiten. Ein anderes umfassendes Buch zum gleichen Thema hat 748 Seiten [Hir11]. Ein typisches Buch zu einer Standard-Projektmanagement-Software hat ca. 400 Seiten.

Viele Mitarbeiter haben nicht die Zeit, neben Ihrer „eigentlichen“ Arbeit dieses Wissen aufzunehmen und zu beherrschen. Die Methoden und Werkzeuge werden nicht oft genug angewendet, um sicher damit umgehen zu können. Für eine gelegentliche Nutzung sind sie zu kompliziert.

Teure, komplexe Werkzeuge haben sich auch deshalb nicht durchgesetzt, weil häufig aus Kostengründen nicht alle Mitarbeiter Zugriff darauf erhalten. In einem Team darf Vorgangsmanagement jedoch nicht allein Aufgabe des Projektleiters oder eines Projektmanagement-Büros sein.

Ein dritter Grund für die mangelnde Akzeptanz liegt darin, dass Mitarbeiter für sich keinen Vorteil in der Benutzung einer Methode oder eines Werkzeugs sehen. Es genügt bei der Einführung eines Systems nicht, wenn allein das Management davon profitiert. Alle Mitarbeiter müssen persönlich gewinnen, wenn sie mit einer neuen Methode und einem neuen Werkzeug arbeiten.



Abbildung 1.4: Eine Methode erfolgreich machen.

Deshalb verlassen sich viele lieber auf einfache Verfahren, die sie schon verstehen und beherrschen. Machen Sie deshalb vor der Einführung einer

neuen Methode oder eines neuen Werkzeugs diesen Lackmus-Test:



- *Sind die Methoden und Werkzeuge schnell und einfach zu verstehen?*
- *Können Sie die Methoden und Werkzeuge einfach einführen?*
- *Sehen die Teammitglieder für sich einen Vorteil darin, mit den Methoden und Werkzeugen zu arbeiten?*

Schnell und einfach zu verstehen bedeutet, dass Sie sich die ganze Idee in etwa zwei Stunden zu Eigen machen können. Sie sollten nicht umfangreiche Bücher lesen oder auf Seminare gehen müssen, um den Sinn der Methode und ihre Anwendung begreifen zu können.

Einfach einzuführen bedeutet, dass Sie nicht hunderttausende Euro in Beratung und Werkzeuge investieren müssen, damit die Methode in Ihrer Organisation etabliert werden kann. Es muss möglich sein, die Vorgehensweise von unten herauf zu verbreiten, beginnend mit einem einzelnen Team oder Projekt. Wenn Sie Erfahrung in der Anwendung gewonnen haben, können Sie die Methode in größerem Umfang einsetzen.

Vorteil in der Anwendung bedeutet, dass Sie Ihre Mitarbeiter nicht jahrelang missionieren müssen, oder Sie sogar dazu zwingen, etwas zu verwenden, das sie nutzlos finden. Sie würden nur die Kreativität Ihrer Leute herausfordern, um die Vorschriften herum zu arbeiten.

Deshalb gilt:



Wenn ein Werkzeug oder eine Methode Erfolg haben soll, muss sie allen zur Verfügung stehen, einfach zu verwenden sein und dem gesamten Team nützlich erscheinen.

1.6 Der Star unter den PM-Werkzeugen

Die Tabellenkalkulations-Anwendung Excel ist das am weitesten verbreitete Werkzeug für das Aufgaben- und Projektmanagement. Dafür gibt es eine Reihe von Gründen:

- Schon die alten Ägypter verwalteten mit Erfolg ihre Vorgänge in Listen.
- Viele Mitarbeiter kennen diesen Ansatz aus dem Bereich des persönlichen Zeit- und Produktivitätsmanagements (Filofax, Time/-System etc.).
- Das Prinzip ist intuitiv zu erfassen und flexibel.
- Man kann einfach filtern.
- Man kann schnell einfache Berichte erstellen.
- Das Werkzeug steht praktisch jedem zur Verfügung.
- Viele Leute kennen das Werkzeug aus anderen Anwendungsbereichen.

Wenn Excel so gut funktioniert, warum sollten wir noch etwas anderes einzusetzen? Die meisten Abläufe und Projekte erfordern tatsächlich keineswegs mächtige Projektmanagement-Methoden und Werkzeuge. Allerdings sind viele Projekte zu komplex, um sie angemessen mit einer Tabellenkalkulation zu verwalten.

Die Excel-Methode versagt, sobald mehr als drei bis fünf Leute an Ihrem Projekt beteiligt oder Ihre Teammitglieder geographisch verteilt sind. Wie kommt das?

- Es gibt keine Versionskontrolle. Oft werden die Tabellen per E-Mail verschickt, und nach kurzer Zeit gibt es verschiedene, inkonsistente Versionen. Bei zentraler Speicherung auf einem Dateiserver bleibt unklar, wer die Dateien geändert hat und was geändert wurde.
- Sie können nicht steuern, wer was sieht oder bearbeiten kann. Alle sehen alles oder nichts.
- In Projekten mit mehreren Firmen ist es schwierig, eine zentrale Kopie auf einem für alle zugänglichen Dateiserver zu halten.

- Es gibt keine Unterstützung für Arbeitspaketstrukturen (*engl. work breakdown structures, WBS*), nur flache Listen. Das macht die Verwaltung großer Projekte schwierig.
- Es gibt keine automatischen Änderungsbenachrichtigungen. Wenn etwas geändert wird, ist es dem jeweiligen Bearbeiter überlassen, davon Betroffene zu informieren.
- Es gibt keine Unterstützung für vorgegebene Arbeitsabläufe. Aktivitäten, wie eine Aufgabe einem Bearbeiter zuweisen und ihn darüber zu informieren, müssen manuell erledigt werden.

Die Methode, alle Vorgänge in Listen zu verwalten, ist einfach und mächtig. Dieses Prinzip in einer Tabellenkalkulation abzubilden ist jedoch mit ernsthaften Einschränkungen verbunden. Sobald das Projektteam aus mehr als drei bis fünf Leuten besteht, und sobald im Projekt mehrere Firmen beteiligt sind oder die Teammitglieder örtlich verteilt sind, bemerken Sie die Defizite dieser Vorgehensweise. Deshalb gilt:



Projekt- und Aufgabenmanagement mit Excel funktioniert mit Einschränkungen nur in kleinen, lokalen Teams.

1.7 Müssen Sie gläubig sein?

Im Bereich der Management-Methoden und Prozesse gibt es immer wieder Moden, die viel Heil versprechen und ihre Gläubigen um sich scharen, bis sich herausstellt, was davon wirklich hilft und was nur ein besonders geschicktes Marketing war.

Eine Geschichte berichtet von einem indischen Guru, der jeden Abend seine Jünger um sich versammelte und sie unterwies zur Erleuchtung. Der Guru hatte eine Katze, die immer wieder die Meditation störte. Daraufhin lies der Guru die Katze jedes mal zu Beginn der Versammlung an ein Stuhlbein binden.

Nach einiger Zeit starb der Guru, und seine Jünger führten die Meditationen ohne ihn fort. Eines Tages starb auch die Katze und die Jünger be-

sorgten sich eine neue. Sie hatten gelernt, dass zu Beginn der Meditation eine Katze an ein Stuhlbein gebunden werden musste.

Hoffentlich ergeht es Ihnen nicht wie diesen Jüngern. Sie sollten verstehen, was hinter den einzelnen Elementen der hier vorgestellten Methodik steckt und für sich selbst bewerten, was Ihnen hilft und was nicht. Die besprochenen Verfahren sind keine magischen Heilsformeln, sondern basieren auf bekannten Methoden und langjährigen Erfahrungen vieler unterschiedlicher Menschen.

Die guten Ideen von David Allen [GTD] und Stephen R. Covey [Cov04] bilden die Grundlage für den Teil, der aus dem persönlichen Zeit- und Produktivitätsmanagement stammt.

Erfahrungen aus dem Projektmanagement mehrerer Jahrzehnte finden sich in verschiedenen Standards wieder. Wir haben insbesondere Ideen aus dem PMBOK [Pmb], dem V-Modell [IAB09], dem CMMI [CMD] [CMS] sowie einigen neueren Trends wie Scrum [Pri10] [Sch11] aus dem Bereich der Softwaretechnik und Kanban [And10] zu Rate gezogen. Die dort beschriebenen Modelle sind nützlich in der Entwicklung effizienter Prozesse und zur Prozessverbesserung.

Die hier vorgestellte Methodik ist einfach zu verstehen, einfach einzuführen, und es macht Spaß, damit zu arbeiten. Probieren Sie es aus!

2

Planung

Denn wer ist unter euch, der einen Turm bauen will und setzt sich nicht zuvor hin und überschlägt die Kosten, ob er genug habe, um es auszuführen.

Lk 14,28

2.1 Das Planungsproblem

Planen bedeutet, durchzuspielen was man tun könnte (siehe [Doe92], Seite 234ff). Man ordnet eine Aktion nach der anderen in einer Kette an, prüft, was passieren könnte und untersucht, ob die zusammengestellte Aktionskette einen dem gewünschten Projektziel näher bringt.

Das Planen selbst ist relativ einfach. So schrieb Kant 1783:

Plane machen ist mehrmalen eine üppige, prahlerische Geistesbeschäftigung, dadurch man sich ein Ansehen von schöpferischem Genie gibt, indem man fodert, was man selbst nicht leisten, tadelt, was man doch nicht besser machen kann, und vorschlägt, wovon man selbst nicht weiß, wo es zu finden ist.

Schwierig wird es erst, wenn der Plan umgesetzt werden soll. Die Vorentwurfsplanung der Hamburger Elbphilharmonie schätzte die Baukosten im Juli 2005 auf 186 Mio. Euro. Nach der Ausschreibung und dem Bieterwettbewerb lag das günstigste Angebot bei 241 Mio. Euro. Im April

2007 wurde mit dem Bau begonnen. Etwas über ein Jahr später stiegen die veranschlagten Baukosten auf ca. 500 Mio. Euro.

Dörner erwähnt in [Doe92] Berthold Brechts Dreigroschenoper. Im gleichnamigen Roman [Bre98] heißt es im „Lied von der Unzulänglichkeit menschlichen Strebens“:

*Ja mach nur einen Plan
Sei nur ein großes Licht!
Und mach dann noch 'nen zweiten Plan,
Gehn tun sie beide nicht.*

Der Bettlerkönig Jonathan Jeremiah Peachum ist geschäftlich in eine Zwickmühle geraten, und verfolgt in genialer Weise seinen Plan, sich aus der prekären Situation wieder zu befreien. Wenn eine Aktion nicht den gewünschten Erfolg bringt, weiß er schon eine Abhilfe. Er hat nicht nur zwei Pläne, er passt seinen Plan laufend der Situation an.

Weniger erfolgreich agierte eine Gestalt aus der Märchenwelt. Dörner zitiert Rumpelstilzchen, von dem wir wissen, dass es nicht immer so läuft, wie geplant:

*Heute back ich,
morgen brau ich,
übermorgen hole ich der Frau Königin ihr Kind.*

Was macht es so schwierig, einen Plan zu finden, der auch realisierbar ist? Die Schwierigkeiten entspringen vornehmlich drei Quellen:

- Der Suchraum für eine Lösung ist unüberschaubar.
- Jede Aktion in einem Plan ist an Annahmen und Randbedingungen geknüpft. Diese ändern sich im Laufe der Zeit.
- Die im Laufe eines Projektes immer auftretenden „Friktionen“ werden bei der Planung nicht berücksichtigt.

Schon bei einem einfachen, mit klaren und invarianten Randbedingungen versehenen Projekt wie einem Schachspiel kommen Menschen und Computer schnell an ihre Grenzen. Beim ersten Zug kann Weiß aus 20

Möglichkeiten auswählen, Schwarz ebenfalls. Damit ergeben sich nach jeweils einem Zug schon 400 mögliche Stellungen. Unter der Annahme, dass jeder Spieler bis zum fünften Zug 20 Möglichkeiten hat und danach noch zehn, kommt man nach nur zehn Spielzügen auf eine Anzahl von

$$20^{10} \cdot 10^{10} \approx 10^{23}$$

möglicher Züge und Gegenzüge, einer Zahl mit 23 Nullen, in Worten hundert Billionen mal eine Trillion. Obwohl es theoretisch möglich wäre, alle Zugvarianten bis zum Ende eines Spiels durchzurechnen, ist es schon nach den ersten zehn Zügen nicht mehr praktikabel.

Um in überschaubarer Zeit zu einer Lösung zu kommen, muss der Suchraum eingeschränkt werden. Dazu gibt es eine Reihe Verfahren, wie z. B. Vorwärts- und Rückwärtsplanung und Planung mit Zwischenzielen, die z. T. weiter unten besprochen werden.

2.2 *Divide et impera*

„Teile und herrsche“ ist ein seit frühester Zivilisation verwendetes Herrschaftsprinzip. Große, umfangreiche Strukturen werden aufgeteilt, um sie zu beherrschen. Dazu werden hierarchische Strukturen verwendet.

Ganz oben in der Planung steht das Gesamtprojekt oder Gesamtprodukt. Es ist möglich, die Unterteilung nach Aktivitäten oder nach den Ergebnissen von Aktivitäten vorzunehmen. So kann z. B. ein Vorgang „Anforderungen für Subsystem XY erstellen“ heißen oder der Vorgang kann durch das Ergebnis definiert sein, also „Anforderungen für Subsystem XY“.

Die erste Variante nennt sich **Arbeitspaketstruktur**, *engl. work breakdown structure (WBS)*, die zweite Variante heißt **Projektstrukturplan (PSP)** oder **Produktstrukturplan**, *engl. deliverables breakdown structure (DBS)*. Es ist in der Regel besser, einen Produktstrukturplan zu erstellen als eine work breakdown structure, da ein Produkt, z. B. ein Dokument, leichter geprüft und nachverfolgt werden kann als eine Aktivität. Wenn es ein fertiges Dokument gibt, muss es jemand erstellt haben, d. h. es gab auch eine Aktivität dazu. Wenn aber jemand vier Wochen an der Erstellung eines Dokuments gearbeitet hat, bedeutet das nicht notwendigerweise, dass es dieses Dokument in einer brauchbaren Form gibt.



Es ist in der Regel besser, die Planung entlang von nachprüfbaren Ergebnissen anstatt anhand von Aktivitäten zu strukturieren.

Das Gesamtprodukt wird eine Hierarchieebene tiefer in Teilprodukte aufgeteilt. Diese können selbst wieder weiter heruntergebrochen werden. Auf jeder Hierarchieebene sollte der PSP oder die WBS zwischen drei und zehn Elemente aufweisen. Bei fünf Hierarchieebenen lassen sich damit schon zwischen 250 und 100.000 Vorgänge organisieren.

Große Projekte oder Projekte mit von der Art her unterschiedlichen Teilkomponenten können auf Projektebene in Teilprojekte unterteilt werden. So können Sie z. B. ein Hardwareteilprojekt, ein Softwareteilprojekt und ein Mechanikteilprojekt definieren, um ein komplexes mechatronisches System zu entwickeln. Teilprojekte haben den Vorteil, dass sie in voneinander unabhängige Phasen unterteilt werden können, und dass der Zugriff auf sie einzeln gesteuert werden kann (siehe Abschnitt 4.1 auf Seite 57).

2.3 Aufwandsschätzung

Ein wichtiger Maßstab für den Projekterfolg ist die pünktliche Lieferung eines Produktes oder einer Dienstleistung zu einem zuvor abgesprochenen Preis. Termine und Kosten lassen sich nur einhalten, wenn die Planung und Aufwandsschätzung zuverlässig ist.

Es gibt viele Methoden und Ansätze für Planungen und Abschätzungen. Ohne Erfahrungen mit einem ähnlichen Projekt bleibt Planen und Abschätzen jedoch ein besseres Raten.

Das ist unabhängig davon, welche Methoden und Kostenschätzungsmodelle Sie einsetzen. Viele dieser Verfahren haben Parameter, die erst kalibriert werden müssen. Woher soll die Kalibrierung bei einem völlig neuen Projekt kommen? Auch bei neueren Methoden wie dem Planungspoker [Coh05] (siehe Abschnitt 2.10 auf Seite 39) werden Sie wenig Freu-

de an den Ergebnissen haben, wenn Sie nicht schon ähnliche Aufgaben bewältigt haben. Die Bibel weiß dazu:

Er sagte ihnen aber auch ein Gleichnis: Kann auch ein Blinder einem Blinden den Weg weisen? Werden sie nicht alle beide in die Grube fallen?

Lk 6,39

Es ist deshalb leider so:



Bei der Planung und Abschätzung ist Erfahrung durch nichts zu ersetzen.

2.4 Frohe Botschaften

Die zuverlässigsten Planungen und Abschätzungen stützen sich auf ein ähnliches Vorgängerprojekt und nehmen nicht an, dass der allgemeine technische Fortschritt den Aufwand halbiert hat. Im Entwicklungsbereich gibt es nur drei Verfahren, den Aufwand signifikant zu senken:

- Wiederverwendung schon existierender Ergebnisse und Komponenten
- Nutzung einer höheren Abstraktion bei der Implementierung der Produkte
- verbesserte Organisation

Bleiben Sie skeptisch, wenn Ihnen jemand eine Methode oder ein Werkzeug anpreist mit dem Versprechen, deutliche Produktivitätsgewinne zu erzielen. Prüfen Sie, ob das Verfahren eines der o. g. Kriterien deutlich erfüllt. Besteht tatsächlich eine höhere Abstraktion, oder bricht diese schon bei der kleinsten Anpassung wieder zusammen? Werden wirklich Teile unmodifiziert wiederverwendet oder wird kopiert und dann verändert? Wird wirklich das Zusammenwirken der Menschen und Sachmittel in ihrem strukturellen Zusammenhang so andersartig gestaltet,

dass man eine völlig neue Art der Organisation hat, oder handelt es sich um belanglose Variationen?

In den meisten Fällen gibt es keine Nachweise für die behaupteten Verbesserungen. Es ist eben praktisch unmöglich, ein Multimillionen-Euro-Projekt dreimal nach verschiedenen Methoden zu entwickeln, nur um zu sehen, welche die am besten geeignete war.

Einige Firmen beschäftigen Prediger, die die frohe Methodenbotschaft verkündigen und die den Mangel an Fakten durch Charisma und Redekunst angenehm verschleiern. Werden Ihnen Produktivitätssteigerungen versprochen, bedenken Sie:



Wenn nicht eine spürbar verbesserte Organisation, Wiederverwendung oder eine höhere Abstraktion hinter einer versprochenen Produktivitätssteigerung steckt, bleiben Sie skeptisch.

2.5 Planen oder nicht planen

Jedes Projekt und jedes Teammitglied hat es mit zwei Arten von Vorgängen zu tun:

- vorhersehbare Vorgänge
- nicht vorhersehbare Vorgänge

Es gibt Personen, die nur mit nicht vorhersehbaren Vorgängen zu tun haben. Niemand kann jedoch ausschließlich nach Plan arbeiten, ohne jemals durch ein unvorhersehbares Ereignis gestört zu werden. Vorhersehbare Vorgänge finden sich vor allem in Projekten und sich zyklisch wiederholenden Arbeitsabläufen.

Nicht vorhersehbare Vorgänge können sowohl im Rahmen von Projekten als auch als Teil eines ereignisgesteuerten Workflows auftreten. Beispiele dafür sind Support Center, die auf Kundenanfragen reagieren, eine Testgruppe, die Probleme mit getesteten Produkten aufzeichnet, oder ein Ver-

triebsingenieur, der sich um den technischen Teil von Kundenanfragen kümmert.

Wenn Sie hauptsächlich nacheinander voneinander unabhängige Vorgänge so gut und schnell wie möglich abarbeiten müssen, brauchen Sie keine Pläne. In einer solchen Umgebung kommt es vor allem darauf an, einen optimalen Durchfluss durch Ihre Organisation zu erreichen. Es sollte nirgendwo in der Prozesskette zu einer Anhäufung an unerledigten Vorgängen kommen und es sollten sich in keinem Eingangskorb zu viele Vorgänge sammeln.

Um einen solchen Prozess zu steuern, genügt es, die Anzahl der jedem Teammitglied zugewiesenen Vorgänge zu beobachten und zu kontrollieren. Hilfreich ist dabei, wenn die Vorgänge vom Arbeitsaufwand her ähnlich sind.

Bevorzugen Sie diese Vorgehensweise, denn sie erfordert den geringsten Planungs- und Verwaltungsaufwand.

Wenn Sie dagegen etwas zu einem bestimmten Termin liefern müssen, und wenn Sie Vorgänge haben, die später weiter detailliert werden und die unterschiedliche Arbeitsaufwände erfordern, dann ist es notwendig, dass Sie Abschätzungen für den anfallenden Arbeitsaufwand erstellen und die Arbeiten planen.

2.6 Wie detailliert darf es denn sein?

Eine große Herausforderung bei der Erstellung eines Plans ist es, den richtigen Detaillierungsgrad zu finden. Es wäre unsinnig, Dinge zu planen, die von vielen Randbedingungen abhängen und bei denen Sie in keiner Weise vorhersehen können, wie diese Bedingungen zur Zeit der Realisierung des Plans aussehen werden.

Nicht jeder planbare Vorgang sollte auch geplant werden. In komplexen, schnell veränderlichen Situationen hat es sich bewährt, nur grob zu planen und die Detailplanung an die für die Ausführung Verantwortlichen zu delegieren.

Je unsicherer sich der Planer fühlt, desto detaillierter werden seine Pläne. Das wiederum führt zu noch mehr Unsicherheit. Eine zu detaillierte Pla-

nung wird unübersichtlich und erfordert einen hohen Aufwand, um sie aktuell zu halten.

Wie detailliert ein Plan sein muss, hängt davon ab, wie schnell Sie merken müssen, dass er nicht eingehalten wird. Als Faustregel gilt:



Ein Vorgang ist detailliert genug, wenn seine Dauer die Hälfte der Zeit zwischen Ihren Projektfortschrittsbesprechungen beträgt.

Wenn Sie also jede Woche eine Projektfortschrittsbesprechung angesetzt haben, sollte ein Vorgang in Ihrem Plan eine Dauer von zwei bis drei Tagen haben; das genügt. Detaillierter zu planen ist nur sinnvoll, wenn verschiedene Bearbeiter involviert sind und diese entsprechend informiert werden müssen.

2.7 Planen mit Zwischenzielen

Für die meisten Organisationen ist es nicht praktikabel, weit im Voraus für jeden einzelnen Vorgang den Aufwand und die Termine zu planen. Das liegt daran, dass die Ziele oft nicht klar definiert sind, und die Randbedingungen der Planung wie Verfügbarkeit von Ressourcen sich häufig ändern. Wenn das Ziel nicht klar ist, können Sie auch nicht rückwärts planen.

In solchen Fällen hat sich das Planen mit Zwischenzielen bewährt. Die Zwischenziele werden so gewählt, dass sie auf dem Weg zum Gesamtprojektziel liegen. Es kann auch hilfreich sein, solche Situationen anzusteuern, die eine gute Ausgangsposition mit möglichst vielen Handlungsoptionen für das weitere Vorgehen bieten.

Wir nennen die Zeit zwischen zwei Zielen eine „**Phase**“. Wir fassen die geplanten Vorgänge in Phasen zusammen und entscheiden lediglich, welcher Phase wir einen Vorgang zuordnen wollen. Abhängigkeiten müssen dabei berücksichtigt werden.

Anstatt also wie im traditionellen Projektmanagement 200 Vorgänge mit 200 Anfangs- und Endterminen zu planen, legen wir zunächst lediglich ein paar wenige Phasen fest, mit jeweils einem Anfangs- und Endtermin. Dann weisen wir die Vorgänge den Phasen zu, abhängig von der in jeder Phase verfügbaren Kapazität und dem geschätzten Arbeitsaufwand.

Phasen sollen sich immer nur über ein paar Wochen erstrecken, damit sie gut verwaltet und geplant werden können.

Damit wir den Arbeitsfortschritt innerhalb einer Phase verfolgen können, benutzen wir die **Leistungswertmethode**, engl. *earned value method (EVM)*, wie weiter unten beschrieben. Auf diese Art besitzen wir ein fundiertes und unaufwändiges Planungs- und Beobachtungsverfahren, das robust gegenüber Änderungen und unvorhersehbaren Ereignissen im Projektablauf ist.

Das entspricht dem Ansatz von Scrum [Sch11] und Kanban [And10]:



Planen Sie nur die nächsten Schritte so weit, wie Sie das Vorgehen einigermaßen sicher vorhersagen können, und achten Sie auf einen guten Fluss der Aufgaben durch ihr Team.

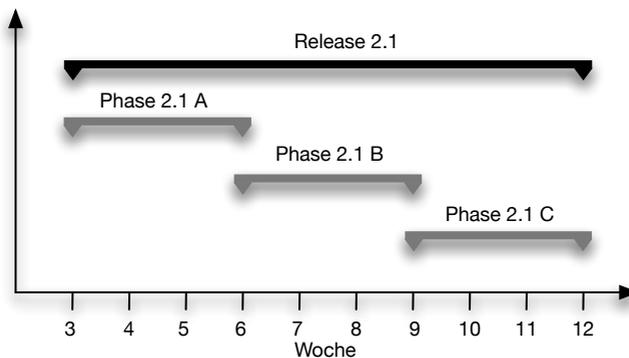


Abbildung 2.1: Release und Phase.

Zu diesem Zweck weisen wir jedem geplanten Vorgang einer Phase zu,

die einen definierten Start- und Endzeitpunkt hat. Jeder Vorgang, egal ob er geplant ist, im Korb für die nächste Aktion liegt, oder im Kalender, besitzt einen **Planwert** (siehe Abschnitt 6.2). Jedes Teammitglied pflegt einen persönlichen Kalender, in dem es die Tage markiert, wo seine zur Verfügung stehende Kapazität vom Standardwert (z. B. acht Stunden) abweicht. Das kann z. B. eine Folge von Ferien, Reisen oder Wochenendarbeit sein.

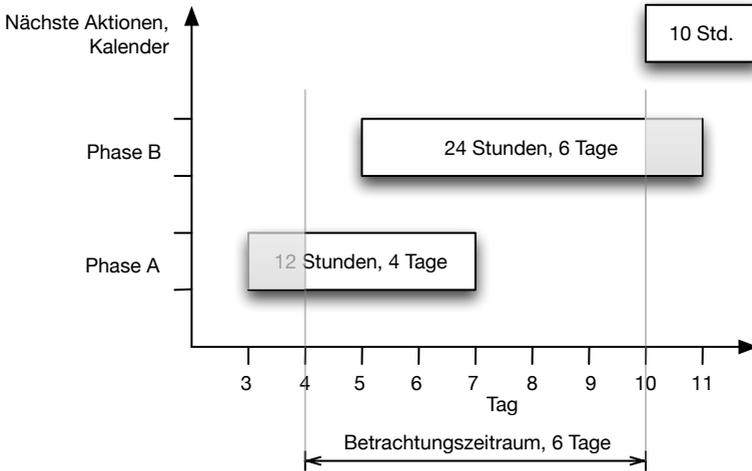


Abbildung 2.2: Berechnung der zugewiesenen Kapazität je Teammitglied.

Für jedes Zeitintervall können wir so für jedes Teammitglied die zugewiesene Arbeitsbelastung berechnen. Dazu addieren wir die Anteile der verschiedenen Restaufwände (geschätzter Restaufwand, *engl. estimated remaining work* (ERW), siehe Abschnitt 6.2) für jeden geplanten Vorgang, der zu der entsprechenden Phase gehört und der ganz oder teilweise innerhalb des betrachteten Zeitintervalls liegt. Den so errechneten Wert können wir mit der verfügbaren Kapazität vergleichen und feststellen, ob das Teammitglied überlastet ist oder ob noch Kapazität verfügbar ist.

Abb. 2.2 illustriert dieses Vorgehen. Das Teammitglied ist in unserem Beispiel in zwei Phasen beteiligt. Im Zeitintervall von Tag 4 bis Tag 10 berechnen wir die zugewiesene Arbeit als

$$\text{zugewiesene Arbeit} = \frac{12 \text{ Std.}}{4 \text{ Tage}} \cdot (7-4) \text{ Tage} + \frac{24 \text{ Std.}}{6 \text{ Tage}} \cdot (10-5) \text{ Tage} = 29 \text{ Std.}$$

Die Länge des Zeitintervalls ist 6 Tage. Wenn wir eine 40-Stundenwoche zugrunde legen, hätten wir 11 Stunden in diesem Intervall übrig, z. B. um uns um nächste Aktionen zu kümmern. Würden wir das Zeitintervall erweitern, z. B. von Tag 3 bis Tag 11, benötigten wir 36 Stunden für geplante Vorgänge und wir könnten nicht mehr alle nächsten Aktionen und Kalender-Einträge unterbringen.

Zusammenfassend können wir festhalten, dass diese niederschwellige Art der Planung folgende Schritte erfordert:

- Sie müssen für jeden Vorgang die Werte für Planwert, *engl. planned value* (PV), aktuelle Kosten, *engl. actual cost* (AC) und ERW eingeben (für die Definitionen siehe Abschnitt 6.2).
- Sie müssen Ihre geplanten Vorgänge zeitlich in Phasen organisieren.
- Sie müssen Ihren Planer oder Projektmanager über Zeiten der Nichtverfügbarkeit informieren.

2.8 Bottom-up und Top-down

In vielen Organisationen ist die Planung ein rekursiver Prozess, bei dem gewünschte oder erforderliche Termine und Budgets zunächst von oben vorgegeben werden. Diejenigen, die für die Ausführung zuständig sind, stellen ihre eigenen Zeit- und Aufwandsabschätzungen dagegen.

In solchen Fällen ist es gut, zwischen den geforderten bzw. Wunschterminen und zur Verfügung stehenden Budgets und den Terminen und Aufwänden, zu denen sich ein Bearbeiter verpflichtet, zu unterscheiden. Dann werden für jeden Vorgang die folgenden Attribute benötigt:

- **Vorgabe Starttermin**, Top-down-Vorgabe für den Beginn der Arbeit an diesem Vorgang
- **Vorgabe Endtermin**, Top-down-Vorgabe für den Endtermin bzw. Fertigstellungstermin für diesen Vorgang

- **zugesagter Starttermin**, Bottom-up zugesagter Starttermin für die Arbeit an diesem Vorgang
- **zugesagter Endtermin**, Bottom-up zugesagter Fertigstellungstermin für diesen Vorgang
- **Budget**, Top-down-Vorgabe für die maximale Menge an Arbeit, die für diesen Vorgang aufgewendet werden darf
- **Planwert**, Bottom-up Schätzung für den Aufwand zur Fertigstellung dieses Vorgangs

2.9 Sortierung von neuen Leistungsmerkmalen

Wenn Sie in der Produktentwicklung tätig sind, haben Sie eine Liste mit gewünschten Leistungsmerkmalen. Sie müssen für jedes neue Release entscheiden, welche neuen Merkmale Sie darin aufnehmen [Sch11] wollen und können. Um hier nach objektiven Kriterien sortieren zu können, bieten sich die folgenden Attribute an:

- **Nutzen**, ein relativer Zusatznutzen, den dieses Merkmal zum Endprodukt beitragen würde
- **Risiko**, eine relative Zahl für die Wahrscheinlichkeit, dass dieses Merkmal nicht in geeigneter Weise (z. B. in Bezug auf Termin, Aufwand oder Technologie) implementiert werden könnte
- **Priorität**, eine relative Priorität für dieses Leistungsmerkmal. Je höher die Priorität, desto früher sollte dieses Merkmal implementiert werden.
- **Notwendigkeit**, ein wahr-/falsch-Ausdruck, der bestimmt, ob ein Merkmal unbedingt notwendig ist, unabhängig von seinem Wert, seinem Risiko oder seiner Priorität

Mit diesen Attributen lassen sich Leistungsmerkmale in eine Reihenfolge bringen. Je höher der Zusatznutzen, je niedriger das Risiko und je höher die Priorität, umso früher sollte das Merkmal implementiert werden.

2.10 Abschätzungen mit Planungspoker

Eine der schwierigsten und folgenreichsten Schritte in einem Projekt ist die Aufwandsplanung. Dazu wollen wir Ihnen hier ein paar bewährte Regeln an die Hand geben:

- Damit Sie eine verlässliche Aufwandsschätzung erstellen können, sollten Sie relativ genau wissen, was Sie liefern müssen; sonst können Sie nur raten. Nicht genau zu wissen, was zu liefern ist, kann daran liegen, dass Sie keine Erfahrung haben, worauf Sie sich einlassen. Es kann auch bedeuten, dass Ihr Kunde nicht wirklich weiß, was er will.
- Wenn Sie noch nie etwas Ähnliches erstellt oder geliefert haben, riskieren Sie, dass Sie mit Ihrer Aufwandsschätzung weit daneben liegen. Erfahrung ist durch nichts zu ersetzen, auch nicht durch die im Folgenden beschriebene Methode.
- Wenn Sie denken, dass es diesmal viel schneller geht als beim letzten Mal, sollten Sie eine nüchterne Begründung haben. Neue Tools und Methoden gehören nicht dazu.
- In manchen Technologiebereichen spielt die Performanz des einzelnen Teammitglieds eine wesentliche Rolle. Insbesondere im Bereich der Wissensarbeit sind Produktivitätsunterschiede von 500% nicht ungewöhnlich. Außer bei sehr großen Teams, wo sich diese Unterschiede ausmitteln, sollten Sie deshalb Ihre Teammitglieder kennen und bei der Aufwandsschätzung mit einbeziehen.
- Handeln Sie den von Ihren Teammitgliedern genannten Aufwand nicht herunter. In der Regel unterschätzen Ihre Mitarbeiter die auftretenden Friktionen und überschätzen ihre Produktivität.

Der Planungspoker ist eine interessante Methode, um zu belastbaren Aufwandsabschätzungen zu kommen, nachdem Sie die Punkte oben alle beachtet haben. Voraussetzung für einen Planungspoker ist eine Liste mit den gewünschten Projektergebnissen. Diese müssen so genau beschrieben sein, dass die Schätzenden verstehen können, worum es geht.

Für den Planungspoker treffen sich die Schätzenden unter Leitung eines Moderators. Jeder Schätzende erhält einen identischen Stapel Karten. Sol-

che Karten können Sie kaufen; ein Anbieter verwendet darauf z. B. die Zahlen 0, $\frac{1}{2}$, 1, 2, 3, 5, 8, 13, 20, 40, 100 sowie ein ? für die Aussage, dass ein Schätzer sich nicht in der Lage fühlt, eine Schätzung abzugeben. Die Zahlen können den Aufwand in Tagen bedeuten, oder in einem anderen, vorher festgelegten konstanten Verhältnis zum realen Aufwand stehen.

Zu Beginn stellt ein Produktmanager die zu liefernden Projektergebnisse vor. Das Team kann nachfragen, wenn ihm etwas unklar erscheint.

Danach legt jedes Teammitglied die Karte mit seiner Schätzung verdeckt vor sich auf den Tisch. Wenn jeder seine Karte abgelegt hat, werden die Karten gleichzeitig herum gedreht.

Den Schätzern mit dem höchsten und dem niedrigsten Schätzwert wird Gelegenheit gegeben, ihre Schätzung zu erläutern und zu begründen. Die Gruppe kann über das Vorgebrachte diskutieren. Die Diskussionszeit kann durch den Moderator oder Projektmanager begrenzt werden.

Der Schätzprozess mit anschließender Diskussion wird solange wiederholt, bis die Gruppe einen Konsens erzielt hat. Der für die Lieferung eines Teilergebnisses zuständige Schätzer hat in diesem Prozess immer ein größeres Gewicht als die anderen Teilnehmer.

Die Nummerierung der Karten berücksichtigt, dass die Schätzungsgenauigkeit umso größer ist, je größer die Schätzung ausfällt. Der Schätzer kann so bei größeren Werten keine vermeintliche Genauigkeit vorspiegeln, sondern muss sich für einen pessimistischen oder optimistischen Wert entscheiden.

3

Organisation und Ausführung

Das erstaunlichste an der modernen Industrie ist, dass sie soviel verlangt und so wenig leistet. Die moderne Industrie scheint in einem Ausmaß leistungsunfähig zu sein, das die gewöhnliche Vorstellungskraft übersteigt. Daher bleibt diese Unfähigkeit unbemerkt.

Small is Beautiful, E.F. Schumacher

3.1 Aufgaben, Vorgänge und Projekte

Im Rahmen der Ablauforganisation fallen vorhersehbare und nicht vorhersehbare Vorgänge an. Ein „**Vorgang**“ ist für uns alles, worum wir uns kümmern müssen. So kann ein Vorgang Teil eines Projekts sein oder im Rahmen einer Kundenunterstützung oder eines regelmäßig auftretenden Arbeitsablaufs auftreten.

Der Begriff des Vorgangs ist sehr allgemein. Wenn wir einen Vorgang näher spezifizieren wollen, sprechen wir z. B. von „Task“, „Aufgabe“, „Ticket“, „Offener Punkt“, „Anforderung“, „Verbesserungsvorschlag“, „Testfall“ oder „Issue“. Bei einigen dieser „**Vorgangstypen**“ handelt es sich nicht direkt um Vorgänge, sondern um Ergebnisse eines Vorgangs. Wir unterscheiden deshalb zwei Arten von Vorgängen:

- ergebnisorientierte Vorgänge
- prozessorientierte Vorgänge

Ergebnisorientierte Vorgänge benennen wir typischerweise nach dem Ergebnis, also z. B. einem Anforderungsdokument, einer Konstruktionszeichnung oder einem Quellcode. Prozessorientierte Vorgänge lösen eine Aktion aus. Beispiele dafür sind Support-Anfragen, Problemmeldungen, Aufgaben oder Aktionen.

Aus pragmatischen Gründen bleiben wir bei dem Überbegriff „Vorgang“, auch wenn wir über Ergebnisse von Vorgängen sprechen. Im Englischen verwenden wir den Begriff „item“, also in etwa „Ding“. Unser Arbeitsalltag besteht darin, dass wir Vorgänge bearbeiten, erstellen oder erledigen.

Abb. 3.2 zeigt unser Workflow-Modell. Dieses Modell ist hier aus der Perspektive eines Teammitglieds dargestellt; jedes Teammitglied besitzt diese Perspektive für sich. Die Struktur basiert im Wesentlichen auf dem Getting-Things-Done (GTD) Prinzip von David Allen [GTD] und ist erweitert worden, um Teamarbeit zu unterstützen.

Alles beginnt mit „Vorgängen“, die David Allen „Zeug“ (engl. „things“) nennt, und die in [GTD] so definiert sind:

„Alles, dem Sie Zutritt zu Ihrer psychischen oder physischen Welt gewährt haben und das sich nicht dort befindet, wo es hingehört, und für das Sie noch nicht bestimmt haben, was daraus werden soll oder was als nächstes damit zu tun ist.“

Beispiele für solche „Vorgänge“ sind Ideen, offene Punkte, Notizen, Fehlermeldungen, Kundenanfragen, Reisepläne usw. Sie können den Zustrom von „Vorgängen“ nicht steuern, sie kommen in jedem Fall auf Sie zu. Alle Vorgänge werden in einem persönlichen Eingangskorb gesammelt. Um Teamarbeit zu unterstützen, sollte der Eingangskorb wie jede andere Ablage ein elektronisches Verzeichnis sein. Sie müssen regelmäßig durch die Vorgänge in Ihrem Eingangskorb gehen. Die Analyse kann zu drei verschiedenen Ergebnissen führen:

- Ein Vorgang ist „**zu bearbeiten**“, d. h. um das durch diesen Vorgang angeregte gewünschte Ergebnis zu erzielen, müssen Sie etwas tun.

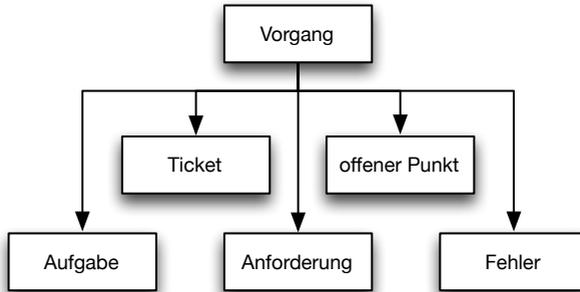


Abbildung 3.1: Beispiele für Vorgangstypen.

- Ein Vorgang ist „**ablegbar**“; Sie können ihn z.B. für eine spätere Nutzung oder als Referenzmaterial ablegen.
- Ein Vorgang erfordert eine ganze Folge von Aktionen, wobei viele andere Personen involviert sein können und wobei sich die Aktionen über einen längeren Zeitraum verteilen. Als Ergebnis der Ana-

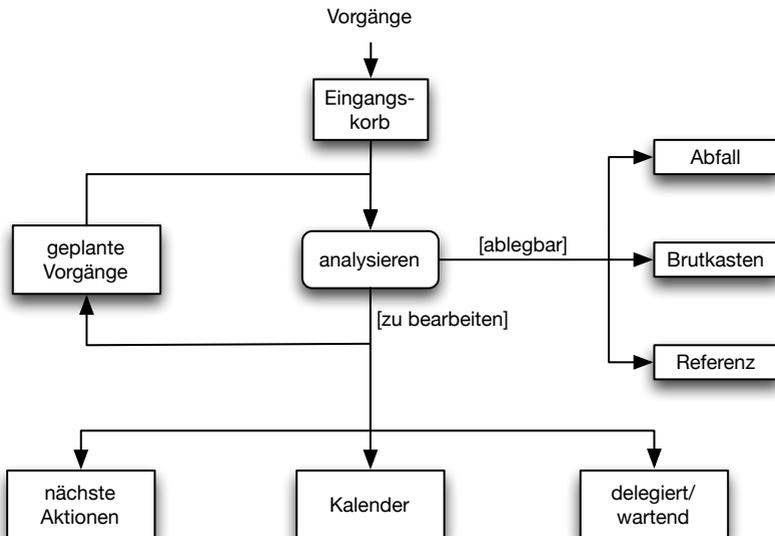


Abbildung 3.2: Das Workflow-Modell (basierend auf [GTD]).

lyse eines solchen Vorgangs erstellen oder ändern Sie „**geplante Vorgänge**“. „Geplante Vorgänge“ können z. B. Teil eines PSP oder einer WBS sein, oder zu einem Scrum-Produkt-Auftragsbestand gehören.

Für ablegbare Vorgänge stehen Ihnen drei Ablagen zur Verfügung:

- **Abfall:** Sie denken über diesen Vorgang nie wieder nach.
- **Brutkasten:** Hier bringen Sie Vorgänge unter, die weder sofort noch in der näheren Zukunft Handlung erfordern, aber die Sie eventuell später wieder berücksichtigen wollen. Diese Vorgänge dienen als Ausgangspunkte für zukünftige Arbeiten. Beispiele für Vorgänge in dieser Ablage sind Ideen, Anforderungen, Verbesserungsvorschläge usw.
- **Referenz:** Diese Ablage enthält Material, das Sie irgendwann wieder gebrauchen könnten, wie z. B. häufig gestellte Fragen, *engl. frequently asked questions* (FAQ), Anforderungen, Normen oder Designs.

Für zu bearbeitende Vorgänge stehen Ihnen ebenfalls drei Ablagen zur Verfügung:

- **Nächste Aktionen:** Hier platzieren Sie die Vorgänge, an denen Sie in naher Zukunft arbeiten wollen. Es gibt keine direkte Planung mit Terminen für diese Vorgänge. Wenn Termine einzuhalten wären, würden Sie die entsprechenden Vorgänge in der Kalender-Ablage unterbringen. Nächste Aktionen können z. B. Anrufe sein, Reaktionen auf Unterstützungsanfragen oder Leistungsmerkmale, an denen Sie arbeiten wollen.
- **Kalender:** Hier bringen Sie Vorgänge unter, die einen direkten Bezug zum entsprechenden Zeitpunkt haben, z. B. einen Termin, oder einen Zeitpunkt für eine Besprechung. Genauer betrachtet enthält die Kalender-Ablage drei Arten von Einträgen:
 - Verabredungen
 - datumsspezifische Aktionen
 - datumsspezifische Informationen, also etwas, das Sie an oder bis zu einem bestimmten Zeitpunkt wissen müssen

- **Delegiert/wartend:** Diese Ablage enthält Vorgänge, die Sie delegiert haben. Wenn Sie einen Vorgang in diese Ablage legen, erscheint er automatisch im Eingangskorb der Person, an die Sie diesen Vorgang delegiert haben. In Ihrer eigenen Ablage halten Sie eine Erinnerungsnotiz.

Die Ablage „geplante Vorgänge“ schauen Sie regelmäßig durch, z. B. zweimal pro Woche. Ein Teil der darin enthaltenen Vorgänge verschieben Sie in eine der zu bearbeitenden Ablagen, oder Sie ändern Vorgänge in dieser Ablage nur, belassen sie aber dort.

3.2 Vorgangsmarkierungen und Attribute

Um Teamarbeit effektiv zu unterstützen, besitzen Vorgänge Markierungen und Attribute. Markierungen helfen Ihnen, die Vorgänge zu kategorisieren und zu strukturieren. Nicht alle Markierungen, die hier erwähnt werden, müssen für alle Vorgänge benutzt werden. Es kann auch sein, dass Sie eigene Markierungen und Attribute erstellen wollen, die Ihre persönlichen Anforderungen abdecken.

Abb. 3.3 zeigt die Markierungen und Attribute, die wir für unser Workflow-Modell benötigen.

- Der **Raum** dient hauptsächlich dazu, den Zugang zu Vorgängen zu steuern, d. h. ob jemand den Vorgang sehen oder ihn ändern kann. Die **Phase** strukturiert den Raum über der Zeit, z. B. für Versionen, Iterationen, Releases oder Sprints. Mehr Informationen dazu erhalten Sie in Abschnitt 4.1 auf Seite 55. Phasen sind optional. In manchen Szenarien entspricht der Raum einem Projekt, in anderen einem Produkt.
- **RACI** bezieht sich auf ein Schema, das mit jedem Vorgang vier Rollen assoziiert, die bestimmen, wer was mit dem Vorgang machen darf und wer wann informiert werden soll. Sie erhalten mehr Informationen zu diesem Thema in Abschnitt 4.4 auf Seite 59.
- Der **Vorgangstyp** beschreibt einen Vorgang genauer, z. B. als ein Leistungsmerkmal, eine Fehlermeldung, eine Kundenanfrage, oder eine einfache Aktion. Vorgangstypen können spezielle Workflows

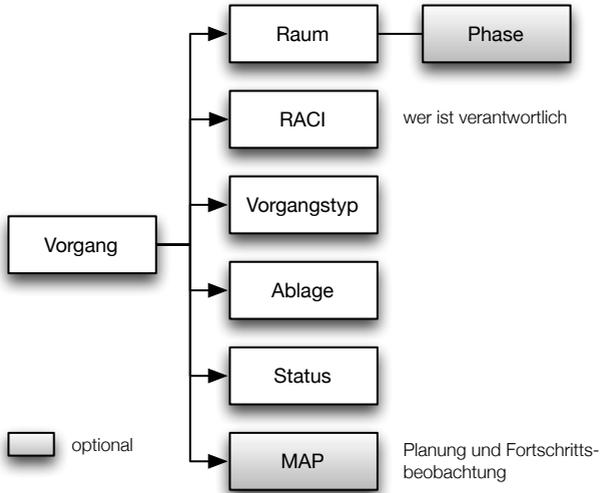


Abbildung 3.3: Notwendige und optionale Vorgangsmarkierungen und Attribute.

und spezifische Attribute zugeordnet sein. Es kann hilfreich sein, einen Vorgangstyp „unbekannt“ zu haben für Dinge, die Sie noch nicht näher einordnen konnten.

- Die **Ablage** bestimmt die Position eines Vorgangs in Ihrem **persönlichen Workflow-Modell**, wie in Abb. 3.2 auf Seite 43 illustriert.
- Der **Zustand** bestimmt die Position des Vorgangs im **Gesamt-Workflow-Modell**. Typische Werte sind „offen“, „geschlossen“, „analysiert“ oder „zurückgestellt“.
- Die unter **beobachten und planen, engl. monitoring and planning (MAP)**, aufgeführten Attribute erlauben es, den Arbeitsfortschritt in einer objektiven und strukturierten Weise mit Hilfe der **EVM** zu beobachten. Weiterhin gibt es Attribute zur Sortierung von Vorgängen nach Wert, Priorität und Risiko. Dazu erhalten Sie weitere Informationen in Abschnitt 2.7 auf Seite 34 und Abschnitt 6.2 auf Seite 66. MAP-Attribute sind optional.

Generell sollten Sie die Anzahl der Attribute minimal halten, denn Sie müssen gepflegt werden. Sie können beliebig weitere Markierungen

und Attribute definieren, wenn es Ihnen hilft. Hier sind noch ein paar Vorschläge:

- **Kontext:** eine hierarchische Struktur, um einen Vorgang in einen oder mehrere Arbeitskontexte einzuordnen. Ein Kontext könnte z. B. ein Ort sein („Büro“, „Reise“, „Zuhause“, „Bücherei“, „Zentrale“), oder ein bestimmter Arbeitsmodus („verwalten“, „entwickeln“, „planen“).
- **Priorität oder Dringlichkeit:** Es kann sinnvoll sein, Vorgänge nach einem Prioritätsschema zu sortieren und abzuarbeiten, z. B. „sofort“, „bald“, „irgendwann“. Wenn Ihre Vorgänge mit Terminen versehen sind, kann die Priorität durch den noch vorhandenen Spielraum ersetzt werden. Priorität wird häufig als eine Mischung von Dringlichkeit und Wichtigkeit verstanden, was die Sache etwas vereinfacht.
- **Wichtigkeit:** beschreibt, wie gravierend es für Sie persönlich oder Ihre Organisation wäre, wenn Sie sich nicht um den Vorgang kümmern würden. Typische Werte sind z. B. „hoch“, „mittel“, „niedrig“ oder „sehr wichtig“, „wichtig“, „egal“ .

3.3 Eingangskorb

Der Eingangskorb ist der Sammelplatz für alle Vorgänge, die Sie betreffen können. In einem elektronischen System sind das die Vorgänge, die andere Ihnen oder Sie selbst sich in den Korb gelegt haben.

Es ist wichtig, dass Sie Ihren Eingangskorb regelmäßig durchgehen. Solange Sie einen Vorgang nicht aus dem Eingangskorb in eine der anderen Ablagen für zu bearbeitende Vorgänge, ablegbare Vorgänge oder geplante Vorgänge verschoben haben, bleibt er dort. Sie sollten Ihren Eingangskorb regelmäßig leeren.

Vorgänge erscheinen in Ihrem Eingangskorb, wenn Sie nach dem RACI-Schema neu dafür verantwortlich gemacht worden sind (siehe Abschnitt 4.4 auf Seite 59 für mehr Information). Das ist der Fall, wenn Sie als der neue Bearbeiter bei einem neuen oder schon existierenden Vorgang eingetragen worden sind.

3.4 Nächste Aktionen

Die Ablage für „Nächste Aktionen“ ist der Mittelpunkt Ihrer täglichen Management-Aktivitäten. Sie beinhaltet alle Vorgänge, die Sie selbst als nächstes abarbeiten wollen.

Alle Vorgänge, die weniger als zwei Minuten in Anspruch nehmen, erledigen Sie sofort, ohne sie erst in diese Ablage zu übernehmen. Nur Sie selbst können Vorgänge in diese Ablage verschieben und niemand sonst.

Vorgänge, die zu einem bestimmten Termin erledigt sein müssen, verschieben Sie in die Kalender-Ablage. Vorgänge, die Sie delegiert haben, verschieben Sie in die delegiert-/wartend-Ablage.

Sobald ein Vorgang entweder an einen neuen Bearbeiter übertragen wurde oder erledigt worden ist, wird er aus der Ablage für nächste Aktionen entfernt.

3.5 Geplante Vorgänge

Es kann sein, dass ein Vorgang im Eingangskorb erfordert, ihn mit einer ganzen Reihe von Arbeitsschritten abzuarbeiten, und es kann auch sein, dass dazu eine ganze Reihe Leute erforderlich sind. Solche Vorgänge gehören in die Ablage „geplante Vorgänge“ und werden dort weiter detailliert.

Vorgänge erscheinen in Ihrer Ablage für geplante Vorgänge, wenn Sie nach dem RACI-Schema als Bearbeiter oder Manager eingetragen sind (siehe Abschnitt 4.4 auf Seite 59 für mehr Information). Sie sehen hier auch alle Vorgänge in den Ablagen anderer Leute für alle Räume oder Projekte, für die Sie als Verwalter zuständig sind.

Geplante Vorgänge werden typischerweise in Projekten organisiert. Dort sind sie Teil einer WBS oder eines PSP, eines Produkt-Auftragsbestands, eines Release-Auftragsbestands oder eines Iterations-Auftragsbestands. Es würde über die Zielsetzung dieses Buches hinausgehen zu beschreiben, wie solche Strukturen zu entwickeln sind. Wie schon in Abschnitt 2.2 erläutert, kann man sagen, dass es hilfreich ist, die geplanten Vorgänge an zu liefernden Produkten wie Zeichnungsunterlagen, Dokumenten, Quell-

code oder Tests festzumachen. Das erleichtert später die Beobachtung des Projektfortschritts. Es ist viel einfacher zu prüfen, ob ein Dokument wirklich fertiggestellt ist, als zu prüfen, ob bei einem Vorgang „Anforderungen schreiben“ wirklich schon 67 % abgearbeitet sind. Sie sollten deshalb einen Produktstrukturplan immer einer reinen WBS vorziehen.

Die Ablage für geplante Vorgänge kann in Räume und Unterräume strukturiert werden. Sie können solche Räume und Unterräume nutzen, um damit

- Projekte
- Subprojekte und Komponenten
- Produkte und Produktkomponenten
- Produkt-Leistungsmerkmale

zu modellieren.

Entlang der Zeitachse können Sie geplante Vorgänge Phasen zuordnen. Phasen können z. B. Releases, Iterationen oder Sprints sein. Sie erhalten dazu mehr Informationen im Abschnitt 4.1 auf Seite 55.

3.6 Kalender

Nach [GTD] enthält die Kalender-Ablage drei Arten von Einträgen:

- Verabredungen
- datumsspezifische Aktionen
- datumsspezifische Informationen

Verabredungen sind z. B. solche zu einer Projektbesprechung oder für einen Kundenbesuch. Das bedeutet, Sie müssen zu einer bestimmten Zeit an einem bestimmten Ort jemanden treffen.

Datumsspezifische Aktionen sind solche Dinge, die Sie irgendwann an einem bestimmten Tag erledigen sollten, z. B. jemanden anrufen oder ein Angebot erstellen. Der genaue Zeitpunkt ist dabei nicht wichtig.

Datumsspezifische Informationen sind Dinge, die Sie bis zu diesem Termin wissen sollten. Sie benötigen z. B. Informationen für Ihre Projektplanung, oder Sie versuchen eine Besprechung zu organisieren und Sie brauchen das Einverständnis der Beteiligten. Sie können in diese Ablage auch Erinnerungen aufnehmen, z. B. wenn Sie jemanden anrufen sollen, nachdem er aus dem Urlaub oder von einer Dienstreise zurückgekehrt ist.

3.7 Delegiert/wartend

Wenn Sie für einen Vorgang nicht der richtige Bearbeiter sind, können Sie ihn delegieren. Sie müssen selbst eine Notiz behalten, die Sie daran erinnert, dass Sie den Vorgang weitergegeben haben.

Die delegiert-/wartend-Ablage enthält alle Vorgänge, die Sie delegiert haben und die noch nicht erledigt sind. In Ihrer Ablage behalten Sie entweder eine Kopie des Vorgangs zurück, und das Original geht an den neuen Bearbeiter. Oder Sie behalten nur einen Verweis auf den Originalvorgang. An die Kopie oder den Verweis können Sie ein Wiedervorlagdatum heften, so dass Sie an diesen Vorgang zu einem bestimmten Termin wieder erinnert werden.

3.8 Abfall

Es kann sein, dass in Ihrem Eingangskorb Vorgänge erscheinen, die von Ihrer Seite aus keinerlei Aktion erfordern und die keine Information enthalten, die es wert wäre, aufbewahrt zu werden. Solche Vorgänge können Sie direkt in den Abfall legen. Sie sollten Ihren Abfallkorb regelmäßig leeren. Vorgänge, die in den Abfall gelegt wurden, werden im Gesamt-Workflow-Modell automatisch in den Zustand „geschlossen“ oder „erledigt“ versetzt.

3.9 Brutkasten

Die Brutkasten-Ablage erlaubt es Ihnen, Dinge zu sammeln, die nicht dringend sind, die Sie aber irgendwann noch einmal aufgreifen wollen.

Vorgänge im Brutkasten dienen als Anregung für zukünftige Arbeiten und Projekte. Sie können sich den Brutkasten als „Parkplatz für zukünftige Projekte“ [GTD] vorstellen.

Es kann z. B. sein, dass Ihre Kunden Ihnen wertvolle Rückmeldungen geben, wie Sie Ihre Produkte verbessern können. Sie haben aber momentan nicht die Ressourcen, diese Vorschläge umzusetzen. In diesem Fall legen Sie die Vorschläge im Brutkasten ab. Sie schauen regelmäßig in Ihren Brutkasten, und irgendwann haben Sie die Zeit und Ressourcen, die Vorschläge in Ihrer Produktentwicklung zu berücksichtigen.

	dringend	nicht dringend
wichtig	Krisen drückende Probleme terminkritische Projekte	Prävention Planung
nicht wichtig	Unterbrechungen einige Anrufe einige Besprechungen einige E-Mails	Triviales Zeitverschwender einige E-Mails einige Anrufe

Abbildung 3.4: Die Zeitmanagement-Matrix nach [Cov04].

Einen anderen Blickwinkel auf diese Ablage können Sie erhalten, wenn Sie die Vorgänge nach „**Dringlichkeit**“ und „**Wichtigkeit**“ einordnen. Stephen R. Covey [Cov04] beschreibt in seinem Buch, wie wir unsere Zeit in einer von vier Kategorien verbringen, wie in Abb. 3.4 dargestellt. Nach diesem Schema werden alle Aktivitäten nach Dringlichkeit und Wichtigkeit eingeordnet. Dringlichkeit bedeutet, wie bald Sie einen Vorgang anpacken müssen, und Wichtigkeit bezeichnet den Wert des Ergebnisses.

Der Brutkasten sollte Dinge aufnehmen, die wichtig sind, aber nicht dringend.

3.10 Referenz

In der Referenz-Ablage sammeln Sie Informationen, die Ihnen selbst oder anderen helfen könnten. Es ist wichtig, dass Sie die Vorgänge hier in geeigneter Weise ordnen und strukturieren, damit sie leicht wieder auffindbar sind.

Beispiele für Vorgänge in der Referenz-Ablage sind FAQs, Normen, Präsentationen, Verträge, wichtige Problemlösungen und projektspezifische Dokumentation.

3.11 Vorgangsverknüpfungen

Vorgänge können in unterschiedlicher Weise miteinander in Beziehung stehen. Eine offensichtliche Beziehung ist die in einer Hierarchie wie der WBS: Ein Vorgang ist das Kind eines übergeordneten Vorgangs. Das könnte z. B. bedeuten, dass der Elternvorgang nicht erledigt sein kann, solange untergeordnete Kindvorgänge noch offen sind.

Weitere Beispiele für Vorgangsverknüpfungs-Typen sind:

- Ein Vorgang darf nicht geschlossen werden, solange der verknüpfte Vorgang noch offen ist. Das ist ähnlich wie bei einer Eltern-Kind-Beziehung, aber die Beziehung darf sich auch außerhalb der normalen Vorgangshierarchie bewegen. Das ist z. B. nützlich, wenn es gemeinsame Komponenten gibt, die von verschiedenen Projekten verwendet werden (Libraries, Frameworks, Baukästen).
- Ein Vorgang kann nicht beginnen, solange ein verknüpfter Vorgang noch offen ist. Damit können Sie zeitliche Abhängigkeiten modellieren, wie es in traditionellen Projektplänen der Fall ist. Es können auch andere Zeitbeziehungen angegeben werden, z. B. wenn ein Vorgang zehn Tage nach Start eines anderen Vorgangs beginnen muss, oder ein Vorgang zusammen mit einem anderen beendet werden soll.
- Ein Vorgang ist ein Duplikat eines anderen Vorgangs.
- Ein Vorgang ist ein Testfall für einen anderen Vorgang, der z. B. eine Anforderung sein könnte.

3.12 Flusskontrolle

Jedes Datenverarbeitungssystem hat eine begrenzte Verarbeitungskapazität. Das trifft auch auf Menschen zu. Wenn sich zu schnell zu viele Vorgänge in Ihrem Eingangskorb häufen, könnte es passieren, dass Ihre Kapazität nicht ausreicht, alle zu verarbeiten. Darüber hinaus kann es psychologisch problematisch sein, wenn Sie sich von den vielen zu erledigenden Vorgängen überwältigt fühlen.

Um solche Situationen zu vermeiden gibt es die Möglichkeit, Gegen-
druck auszuüben, indem die Hereinnahme weiterer Vorgänge über ein
bestimmtes Limit hinaus verweigert wird. Wenn jemand einen Vorgang
in Ihren Eingangskorb legen will, und dieser ist schon am Limit, würde
die andere Person über die Situation in Kenntnis gesetzt und der Vorgang
würde zurückgewiesen.

4

Zugangskontrolle

Das sollt ihr aber wissen: Wenn ein Hausherr wüsste, zu welcher Stunde der Dieb kommt, so ließe er nicht in sein Haus einbrechen.

Lk 12,39

4.1 Räume

Bis jetzt haben wir das Workflow-Modell aus Ihrer persönlichen Perspektive betrachtet. Um Teamarbeit zu unterstützen, müssen wir den Zugang zu Vorgängen steuern können. Nicht alle Vorgänge sind rein privater Natur, und nicht alle Teammitglieder sollen alle nicht-privaten Vorgänge sehen können. Zugangskontrolle hat zwei Vorteile:

- Sie können nur auf Vorgänge zugreifen, für die Sie eine Berechtigung haben.
- Sie werden nicht mit Informationen belästigt, die Sie gar nicht betreffen.

Wir nutzen „**Räume**“, um den Zugang zu allen Vorgängen zu steuern. Räume können selbst wieder Räume beinhalten, also hierarchisch angeordnet werden.

Ein Vorgang gehört immer zu genau einem Raum.

Was Sie in einem Raum und seinen Unterräumen tun können, wird durch die Rollen bestimmt, die Ihnen für diesen Raum zugeteilt wurden. Wenn Sie nur für einen Unterraum eine Rolle haben, bedeutet das keine Rollenzuordnung für einen übergeordneten Raum.

Sie können mehrere Rollen im gleichen Raum einnehmen, und Sie können unterschiedliche Rollen in verschiedenen Räumen besitzen.

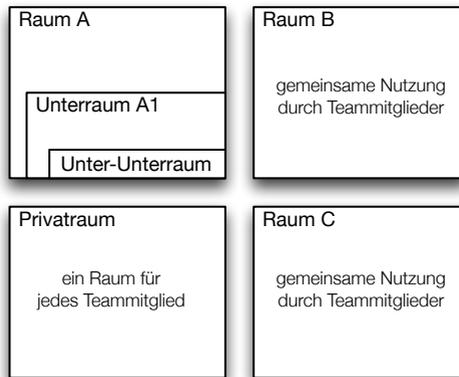


Abbildung 4.1: Wir organisieren alle Vorgänge in Räumen.

Jeder hat seinen eigenen privaten Raum, der nicht mit anderen geteilt werden kann (siehe Abb. 4.1). Alle Vorgänge im privaten Raum können von niemandem sonst eingesehen werden, egal welche Rolle er besitzt. Private Räume besitzen keine Unterräume.

Wenn Sie keine Zugangsberechtigung für einen Raum besitzen, ist dieser Raum für Sie unsichtbar. Typische Zugangsberechtigungen definieren z. B., dass Sie in diesem Raum einen Vorgang anlegen können, dass Ihnen ein Vorgang in diesem Raum als Bearbeiter zugewiesen werden kann oder dass Sie alle Vorgänge in diesem Raum sehen können.

Wenn Sie Projekte managen, können Sie für jedes Projekt einen Raum reservieren. Wenn Sie mehr Produkte pflegen und verwalten, können Sie einem Produkt einen Raum zuordnen und den Produktkomponenten Unterräume. Unterräume zu verwenden ist nur sinnvoll, wenn Sie große Teams haben und nicht alle Teammitglieder Zugang zu allen Vorgängen haben sollen.

Wenn Sie z. B. technische Projekte mit Hardware-Komponenten, Software und Mechanik verwalten, können Sie ein Gesamtprojekt erstellen und darunter jeweils Teilprojekträume für die Hardware, Software und Mechanik einrichten. In großen Projekten könnten Sie darunter weitere Teilprojekträume eröffnen, z. B. für einzelne Komponenten. Sie könnten auch auf der ersten Unterebene Teilprojekträume für Komponenten erstellen und erst darunter in Hardware, Software und Mechanik verzweigen.

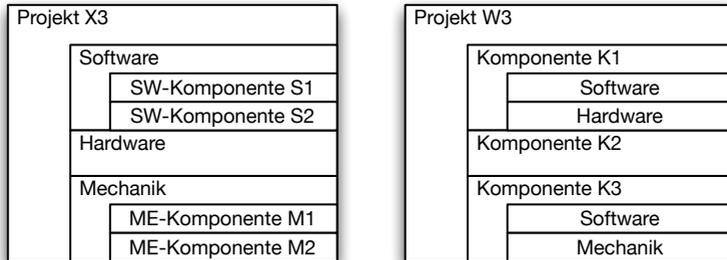


Abbildung 4.2: Teilprojekte für Komponenten.

Eine spezielle Situation entsteht dann, wenn Berechtigungen durch die Zugehörigkeit zu einer Organisationseinheit oder Gruppe und nicht über eine Rolle vergeben werden sollen. Wenn Sie z. B. ein Projekt mit mehreren Firmen durchführen, sollen vielleicht alle Mitarbeiter der gleichen Firma die Vorgänge aller anderen Mitarbeiter dieser Firma sehen können, aber nicht die Vorgänge von Mitarbeitern einer anderen Firma. In diesem Fall müssen sich die Berechtigungen aus der Zugehörigkeit zu einer Gruppe ergeben und nicht allein aus der Rolle.

4.2 Phasen

Sie können Räume nicht nur hierarchisch strukturieren, sondern auch über der Zeit. Dazu weisen Sie die Vorgänge „Phasen“ zu. Eine Phase ist ein Zeitintervall, an dessen Ende Sie eine vorher definierte Menge an geplanten Vorgängen erledigt haben.

Phasen können ebenfalls hierarchisch organisiert werden. Wenn Sie z. B. nach dem Scrum-Modell [Sch11] arbeiten, teilen Sie die Entwicklung Ih-

res Produktes in sogenannte „Sprints“ auf. Sie fassen eine Anzahl solcher Sprints zusammen, um ein Release zu bilden. In diesem Fall wäre das Release eine Phase, und das Release würde aus einer Anzahl Sprints bestehen, die selbst wieder Phasen wären.

Phasen können Attribute haben. Sie können einen Zustand haben wie „geplant“, „freigegeben“ oder „archiviert“, sie können ein Start- und Enddatum haben und sie können eine Zeitbudget sowie ein Geldbudget besitzen.

4.3 Räume und Ablagen

Abb. 4.3 hilft Ihnen zu verstehen, wie Räume und Ablagen miteinander in Beziehung stehen. Ablagen können sich über alle Räume erstrecken, auf die Sie Zugriff haben.

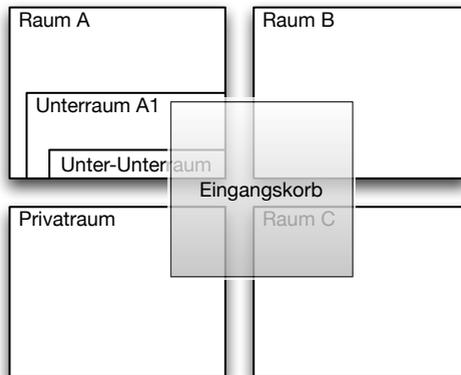


Abbildung 4.3: Beziehung zwischen Eingangskorb und Räumen.

Der Eingangskorb z. B. enthält Vorgänge aus Ihrem privaten Raum sowie allen anderen Räumen, in denen Sie eine Rolle haben. Das gleiche gilt für alle Ablagen, wie die Referenz-Ablage oder die Nächste-Aktionen-Ablage.

Dieses Prinzip ermöglicht es, auch in anderer Leute Ablage hineinzuschauen. Der Zugriff auf einen Vorgang ist nicht durch die Ablage be-

stimmt, in der er sich befindet, sondern durch den Raum, zu dem er gehört.

4.4 RACI-Rollen und die RACI-Matrix

Wir nutzen „Räume“ um den generellen Zugang zu Vorgängen zu steuern. Die Rollen mit ihren zugeordneten Berechtigungen bestimmen, was Sie in einem Raum tun können.

Neben diesen globalen Rollen können Sie zu einem einzelnen Vorgang in besonderer Beziehung stehen. So sind Sie z. B. der Ersteller eines Vorgangs, oder Sie sind gerade für ihn verantwortlich. Diese **vorgangsspezifischen Rollen** strukturieren wir in unserem Modell nach dem RACI-Schema [JaK09].

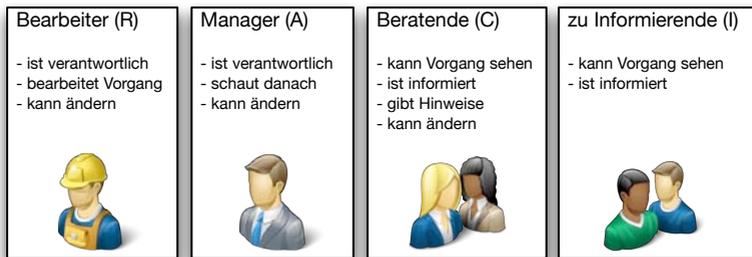


Abbildung 4.4: Das RACI-Rollenschema.

Als **Bearbeiter** (*engl. Responsible*) für einen Vorgang sind Sie dafür verantwortlich, dass der Vorgang zum gewünschten Ende gebracht wird. Kein anderer wird sich darum kümmern. Wenn Sie einen Vorgang delegieren, ändern Sie effektiv den Bearbeiter. Es ist in der Regel am besten, wenn eine einzelne Person einem Vorgang als Bearbeiter zugewiesen ist. Nur in besonderen Fällen früh im Workflow kann es hilfreich sein, eine Gruppe einzutragen. Der erste verfügbare Mitarbeiter aus der Gruppe kann sich dann selbst den Vorgang zuordnen.

Als **Manager** (*engl. Accountable*) für einen Vorgang schauen Sie danach, dass dieser Vorgang nicht unbearbeitet bleibt. Es könnte z. B. sein, dass ein Mitarbeiter im Urlaub ist, oder sogar das Projekt oder die Firma verlassen hat und noch einige Vorgänge unerledigt sind. Als Manager sor-

	Eileen	Carl	Babette	Thomas
Projektplanung	A	R	C	
technische Dok. vorbereiten	A		R	I
Besprechungsprotokolle	AR			I
Verhandlung mit Kunden	A	C		R
Kunden unterstützen	A	R	I	C
Verträge vorbereiten	A		R	
Vertrag unterzeichnen	AR	I	I	I

Abbildung 4.5: Ein Beispiel für eine RACI-Matrix.

gen Sie dafür, dass diese Vorgänge einem anderen Bearbeiter zugewiesen werden. Als Manager sind Sie dafür verantwortlich, wie viel Zeit und Geld für die Erledigung eines Vorgangs zur Verfügung steht.

Als **Berater** (*engl. Consulted*) für einen Vorgang sind ggfs. Beiträge von Ihnen z. B. in Form einer Beratung erforderlich, damit der Vorgang erfolgreich erledigt werden kann. Es ist durchaus üblich, dass mehrere Personen auf der Beraterliste für einen Vorgang stehen.

Wenn Sie für einen Vorgang **zu Informierender** (*engl. Informed*) sind, dann benötigen Sie Informationen von diesem Vorgang, um Ihre eigene Arbeit erfolgreich erledigen zu können. Es ist durchaus üblich, dass mehrere Personen auf der Liste der zu Informierenden für einen Vorgang stehen.

Die RACI-Matrix (siehe Abb. 4.5) definiert für jeden Vorgang, wer welche Rolle einnimmt. Nach diesem Schema sollten Sie keine Reihen ohne „R“ und „A“ haben; idealerweise hat jede Reihe genau ein „R“ und ein „A“. Wenn es zu viele „R“ in einer Spalte gibt, haben Sie eventuell einen Flaschenhals in Ihrem Prozess. Wenn Sie eine Menge „I“ und „C“ in Ihrer Matrix haben, verteilen Sie vielleicht zu viel Informationen an Leute, die sie gar nicht brauchen.

5

Kommunikation

Man soll erst denken und dann sprechen, denn es ist eine Verschwendung der Zeit, sich nach Dingen zu erkundigen, die man sich so leicht denken kann.

Winnetou in Old Surehand III, Karl May

5.1 Babel und die Folgen

Auf den ersten Seiten der Bibel sind neben dem ganz großen Projekt, der Erschaffung der Welt und aller Kreaturen, noch zwei weitere erwähnt: der Bau der Arche durch Noah und der Bau eines Turms zu Babel. Wie Sie wahrscheinlich wissen, waren die ersten beiden Projekte erfolgreich, das dritte aber nicht. Man hat sich nicht mehr verstanden.

Auch heute gibt es Verständigungsprobleme in Projekten. Das liegt nicht an den Worten und Sätzen selbst, sondern an deren unklarer Bedeutung. Viele unterschätzen die Bedeutung einer sauberen, für alle klar verständlichen Terminologie mit präzisen Begriffen. Was meint man mit „Gerät“, was ist eine „Komponente“, was eine „Spezifikation“? Wird hier eine Anforderung spezifiziert, ein Design oder ein Test?

Wenn möglich greifen Sie auf Normen oder andere öffentliche Standards zurück. Firmeninterne Begriffe entwickeln sich meistens unbewusst und nach Bedarf. Es lohnt sich, hin und wieder den Begriffsdschungel zu durchleuchten und Begriffe zu klären. Diese kann man in einem Firmen-

Glossar oder einer Firmen-Norm festschreiben. Auch die Werkzeuge, mit denen gearbeitet wird, sollten in der Lage sein, sich der spezifischen Terminologie anzupassen.

5.2 Informationsaustausch

In der Projektarbeit ist es wichtig, alle Teammitglieder in geeigneter Weise zeitnah mit den für sie erforderlichen Informationen zu versorgen. Eine der großen Herausforderung im modernen Projektmanagement ist die Bewältigung der Informationsflut. Wichtige Information muss sofort verfügbar sein, und unwichtige sollte erst gar nicht beim Empfänger ankommen und ihn stören.

Wir können Informationsaustausch nach folgendem Schema klassifizieren:

- **Echtzeit:** Alle Parteien müssen gleichzeitig anwesend sein. Dies ist die effektivste Art, sich miteinander zu synchronisieren, aber diese Art des Austausches ist auch am schwierigsten zu organisieren. Beispiele dafür sind Besprechungen, Telefonkonferenzen, Webkonferenzen.
- **Quasi-Echtzeit:** Die Parteien müssen nicht direkt antworten, eine gewisse Verzögerung wird akzeptiert. In diesem Fall kann eine Synchronisierung etwas Zeit in Anspruch nehmen, aber man muss keinen gemeinsamen Termin ausmachen. Beispiele sind Chats und Pinboards.
- **Offline:** Die Parteien erwarten keine Antwort innerhalb weniger Stunden. Diese Art des Austausches hat einen recht langsamen Rückkanal und den geringsten Anteil an interaktiver Kommunikation von allen drei Austauscharten. Beispiele sind E-Mail, Foren und Wikis.

Eine zweite Dimension der Klassifikation von Informationsaustausch-Aktivitäten ist

- **gezielt:** Eine Partei sendet aktiv Information an bestimmte andere Parteien oder versucht gezielt, einen Kommunikationskanal zu etablieren. Beispiele sind Telefonanrufe, E-Mails und Chats.

- **abonniert:** Eine Partei veröffentlicht Information auf einem Informationskanal. Die Empfänger abonnieren den Kanal und bekommen so alles mit, was auf diesem Kanal an Informationen verbreitet wird. Beispiele sind News Feeds und Pinboards.
- **gesucht:** Die Empfänger suchen aktiv nach Informationen. Beispiele sind Dokumentenserver, Websites und Wikis.

	Echtzeit	Quasi-Echtzeit	offline	gezielt	abonniert	gesucht
Besprechung	■			■		
Anrufe	■					
E-Mail			■			
Chat		■				
Pinwand		■	■		■	
Forum		■	■		■	■
Wiki			■			■
Dokumentenserver			■			■
News feed		■			■	
Webpräsenz			■		■	■

Abbildung 5.1: Informationsklassifikation.

Für Wissensarbeiter hat sich der Informationsaustausch über quasi-echtzeitfähige und abonnierte Informationskanäle als die in vielen Fällen effizienteste Art der Kommunikation herausgestellt, d. h. Chats und Pinboards. Chats erlauben Kommunikation mit wenig Verzögerung ohne eine gleichzeitige Präsenz zu erfordern, und Pinboards erlauben es dem Empfänger, das zu abonnieren, was für ihn interessant ist.

5.3 Rollenbasierter Informationsfluss

Beim Abonnieren muss es möglich sein, Informationen gezielt zu selektieren. Abhängig von der Rolle, die der Empfänger in Bezug auf eine Information oder einen Vorgang einnimmt, und abhängig vom Informationsinhalt kann er festlegen, ob er die Information erhält oder nicht.

Zu diesem Zweck bietet sich das RACI-Rollenschema an. So möchte z. B. der Manager eines Vorgangs wissen, wenn dieser überfällig ist. Oder ein zu Informierender will wissen, wann ein Vorgang erledigt ist, aber nicht, wenn irgend etwas davor mit diesem Vorgang passiert.

Ein rollenbasiertes Informationssystem erlaubt die Festlegung von Informationsregeln basierend auf der Rolle, die der mögliche Empfänger in Beziehung zu einem Vorgang einnimmt. Ist der mögliche Empfänger z. B. der Bearbeiter eines Vorgangs, und wird irgendetwas an diesem Vorgang geändert, soll der Bearbeiter darüber informiert werden. Ist der mögliche Empfänger dagegen in der Manager-Rolle, möchte er z. B. nur informiert werden, wenn der Vorgang als erledigt markiert ist. Alle anderen Änderungen sind für ihn nicht wichtig.

6

Beobachten und steuern

*Wenn Sie nicht wissen, wo Sie sind, hilft Ihnen auch keine Karte.
unbekannt*

6.1 Berichte

Der vom Aufwand her größte Teil eines Projektes liegt in der Durchführungsphase. Der vorher erstellte Plan wird darin ausgeführt. Es ergeben sich Störungen, auf die Sie reagieren müssen, z. B. indem der Plan angepasst wird. Dazu müssen Sie wissen, wo das Projekt steht.

Berichte informieren Sie regelmäßig über Ihr Projekt. Wir teilen die Berichte in folgende Kategorien ein:

- **Aktueller Projektstand:** gibt Ihnen Informationen zum aktuellen Stand des Projekts, z. B. über die Anzahl an offenen und erledigten Vorgängen oder die Anzahl Vorgänge je Bearbeiter.
- **Projektfortschritt und Trend:** zeigt den Verlauf von Projektparametern über der Zeit, z. B. die Entwicklung von neuen Vorgängen gegenüber erledigten Vorgängen oder den Verlauf der Anzahl erledigter Vorgänge.
- **Auslastung der Ressourcen:** zeigt auf, wie Ressourcen über der Zeit mit Vorgängen belegt sind.
- **Projekt-Hotspots:** weist auf Projekt-Problemzonen hin, z. B. überfällige Vorgänge oder Vorgänge, die mehrmals erledigt und wieder geöffnet worden sind.

- **Aufwände und Abrechnungen:** fasst die verbuchten Aufwände eines Projekts zusammen.
- **Prozessperformanz:** gibt Informationen über die Güte von Prozessen, z. B. die Reaktionszeit bei Störungen, die Termin- und Budgettreue oder die Anzahl von Problemmeldungen für ein bestimmtes Produkt.

Als Grundlage für die Verfolgung des Projektfortschritts verwenden wir die Leistungswertmethode (Earned Value Method). Sowohl die Projektplanung als auch eine Reihe von Berichten stützen sich auf die Daten dieser Methode.

6.2 Die Leistungswertmethode (Earned Value)

Der „Fast fertig“-Effekt führt dazu, dass Sie aufgrund der Berichte Ihrer Mitarbeiter in der Mitte des Projektes denken, Sie hätten 90% geschafft. Diese 90% bleiben dann für ein paar Monate konstant, obwohl heftig gearbeitet wird und Sie wundern sich, dass am Ende des Budgets noch so viel Projekt übrig ist.

Damit das nicht passiert, verwenden wir einige grundlegende Konzepte aus der Leistungswertmethode (EVM). Für diese Methode benötigen wir drei neue Attribute pro Vorgang:

- **Planwert (PV):** *engl. planned value*, auch bekannt als budgeted cost of work scheduled (BCWS), der geplante Arbeitsaufwand, um diesen Vorgang zum gewünschten Schluss zu bringen.
- **aktuelle Aufwand (AC):** *engl. actual cost*, die bisher für diesen Vorgang tatsächlich aufgewendete Arbeit.
- **geschätzter Restaufwand (ERW):** *engl. estimated remaining work*, zu jedem Zeitpunkt der geschätzte Restaufwand, um diesen Vorgang zum gewünschten Schluss zu bringen. Am Anfang ist dieser Wert identisch mit dem Planwert.

Es gibt bei der Leistungswertmethode zwei abgeleitete Parameter:

- **Leistungswert (EV):**, *engl. earned value*, oder auch budgeted cost of

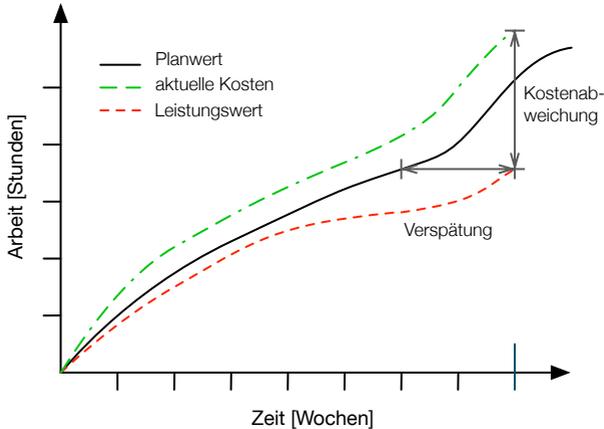


Abbildung 6.1: Analyse basierend auf Leistungswertmethode.

work performed (BCWP), ist ein Prozentsatz des Planwerts, je nach Zustand eines Vorgangs. Im einfachsten Modell beträgt der Leistungswert null, wenn der Vorgang nicht komplett abgeschlossen ist, und 100 %, wenn der Vorgang vollständig abgearbeitet ist. Alternativ können Sie für jeden Zustand einen zugeordneten Fertigstellungsgrad definieren, z. B. „implementiert“ entspricht 75 %.

- **aktueller Planwert (APV)**:, engl. *actual planned value*, die aktuellen bisher aufgelaufenen Aufwände (AC) zuzüglich dem geschätzten Restaufwand (ERW)

$$\text{actual planned value} = \text{actual cost} + \text{estimated remaining work}$$

Wenn Sie die drei Werte PV, AC und EV über der Zeit beobachten, können Sie frühzeitig Hinweise auf Aufwandsabweichungen und Verzögerungen erhalten (siehe Abb. 6.1).

Diese Methode funktioniert sehr gut, wenn der Leistungswert nur mit der nachprüfaren Erledigung des zugehörigen Vorgangs verbucht wird. Während an dem Vorgang gearbeitet wird, soll der Leistungswert trotz Fortschritts bei null verbleiben.

Diese Vorgehensweise ist konservativ und gibt Ihnen ein ziemlich realistisches Projektbild, wenn Sie sich bezüglich des Detaillierungsgrads von Vorgängen an die Empfehlung auf Seite 34 gehalten haben. Aus der Vielzahl an Vorgängen resultiert eine relativ glatte Kurve.

6.3 Aktueller Projektstand

Die folgenden Berichte helfen, den aktuellen Projektstand darzustellen:

- Ampeldarstellung des Projekts, basierend z. B. auf der prognostizierten Termin- und Budgettreue sowie der Ressourcenauslastung
- Anzahl unerledigter Vorgänge, evtl. gruppiert nach Zustand, Bearbeiter und Komponente
- Anzahl neuer Vorgänge im Berichtszeitraum, evtl. gruppiert nach Zustand, Bearbeiter und Komponente
- Anzahl erledigter Vorgänge im Berichtszeitraum, evtl. gruppiert nach Zustand, Bearbeiter und Komponente
- Budgetübersicht mit einer Darstellung, wie weit der Planwert schon verbraucht worden ist und wie viel Restarbeit noch zu erledigen ist

6.4 Projektfortschritt und Trend

Zur Darstellung des Projektfortschritts werden Planwert und Leistungswert herangezogen. In einfacheren Fällen genügt es auch, nur die Anzahl erledigter Vorgänge über der Zeit zu betrachten. Die folgenden Berichte helfen, den Projektfortschritt und Trend darzustellen:

- Verlauf von Planwert, Leistungswert und aktuellem Aufwand über der Zeit, siehe Abb. 6.1
- Meilenstein-Trendanalyse, siehe Abb. 6.2
- Burndown-Chart, siehe Abb. 6.3
- Verlauf der akkumulierten Anzahl Vorgänge je Zustand über der Zeit, siehe Abb. 6.4

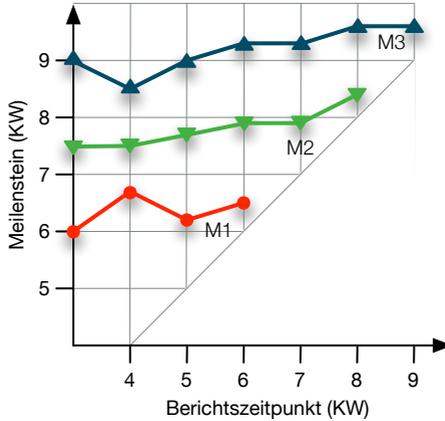


Abbildung 6.2: Meilenstein-Trendanalyse.

6.5 Ressourcenauslastung

Die folgenden Berichte helfen, die Ressourcenauslastung darzustellen:

- graphische Darstellung der Ressourcenauslastung über der Zeit, siehe Abb. 6.5
- textuelle Darstellung der Ressourcenauslastung über der Zeit, siehe Abb. 6.6

6.6 Hotspots im Projekt

Als Projektmanager ist es für Sie wichtig, problematische Bereiche identifizieren zu können. Die folgenden Berichte helfen dabei:

- Liste mit mehrfach erledigten und wiedereröffneten Vorgängen, siehe Abb. 6.7. Solche Vorgänge deuten auf nicht beherrschte Probleme hin und erfordern u. U. den Einsatz zusätzlicher Maßnahmen oder anders qualifizierten Personals.
- Liste von Vorgängen mit überfälligem Termin, überzogenem Budget oder solchen, bei denen der Termin aufgrund des noch zu leistenden Aufwands absehbar nicht gehalten werden kann. Stehen sol-

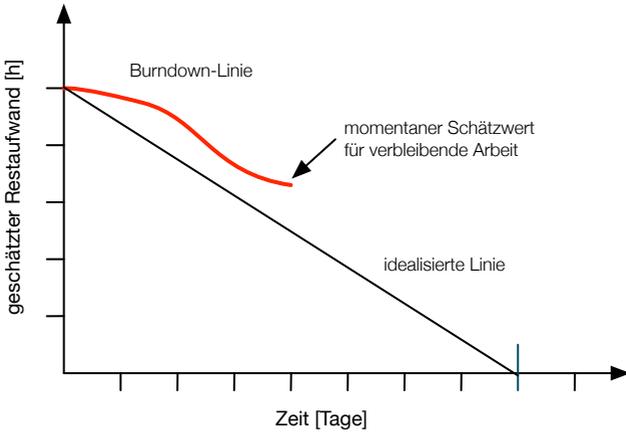


Abbildung 6.3: Burndown-Chart.

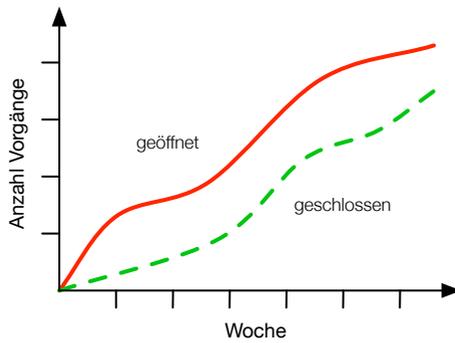


Abbildung 6.4: Anzahl Vorgänge je Zustand über der Zeit.

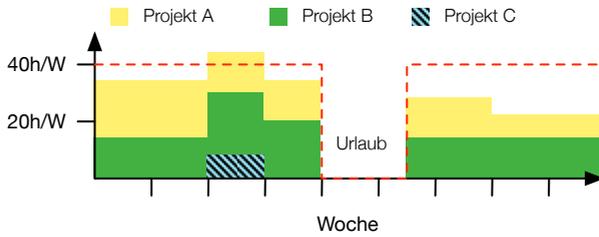


Abbildung 6.5: Geplante Auslastung von Ressourcen über der Zeit, graphisch

Person	KW	Projekt	geplante Std.	akt. Schätzwert
Mustermann, Max		Summe Person:	126.0	119.0
	ungepl.	Summe ungepl.:	30.0	30.0
		technischer Support	30.0	30.0
	39/2012	Summe KW:	13.0	13.0
		Beispielprojekt	13.0	13.0
	40/2012	Summe KW:	36.0	36.0
		Beispielprojekt	1.0	1.0
		Software-Projekt	35.0	35.0
	41/2012	Summe KW:	46.0	39.0
		Beispielprojekt	11.0	4.0
		Software-Projekt	35.0	35.0
	42/2012	Summe KW:	1.0	1.0
		Beispielprojekt	1.0	1.0
		Software-Projekt	0.0	0.0
Musterfrau, Steffi		Summe Person:	112.0	123.0

Abbildung 6.6: Geplante Auslastung von Ressourcen über der Zeit, tabellarisch.

che Vorgänge im kritischen Pfad des Projektes, ist der Fertigstellungstermin gefährdet.

- Liste von Anforderungen ohne Test, ein spezieller Bericht für Entwicklungsprojekte mit besonderen Anforderungen an die Sicherheit, z. B. SIL-Anforderungen.
- Liste von schon lange unerledigten Vorgängen ohne Termin. Solche Vorgänge werden irgendwann vergessen und hinterlassen, wenn es sich um Problemmeldungen oder Anforderungen von Kunden handelt, bei diesen einen schlechten Eindruck.

Vorgang	Titel	# Redos
PR1-1217	Brenner fällt sporadisch aus	6
TB2-441	Einlassventil schaltet gelegentlich ab	4
SRT-8744	Drehteil #345.3665.78 Bestellung 78.65.88 außerhalb Toleranz	4
SRT-8765	Lager wird heiß	4
PR1-1406	Abnahme für Fertigungsfreigabe	3
RPP-43	Zu viele Kugeln im Kugellager	3
RPP-46	Systemabsturz nach ca. 48 h Betrieb	2
TCD-7843	Unit-Test Modul ORM WorkItemPeer fails	2
SRT-7843	Systemabnahme für Temperatursteuerung	2
...

Abbildung 6.7: Mehrfach wiedereröffnete Vorgänge.

6.7 Aufwände und Abrechnungen

Zu Erstellung der Projekthistorie, zur Verbesserung des Prozesses und zur Abrechnung intern oder mit Kunden ist es hilfreich, die tatsächlichen Aufwände über Zeit, z. B. täglich, wöchentlich oder monatlich darzustellen. Wenn Sie Konten einrichten und diese mit der Art der Aufwände verknüpfen, können Sie Informationen zur Prozessoptimierung erhalten. Die folgenden Berichte sind in diesem Zusammenhang hilfreich:

- Detaillierte Aufwände, gruppiert nach Person, Projekt und Konto. Damit lässt sich genau darstellen, woran eine Person mit welchem Aufwand gearbeitet hat. Dieser Bericht hilft vor allem als Dokumentation der geleisteten Arbeiten des Auftragnehmers gegenüber dem Auftraggeber.
- Akkumulierte Aufwände, gruppiert nach Person, Projekt und Konto. Dieser Bericht eignet sich für eine wöchentliche oder monatliche Übersicht bzw. Abrechnung. Wenn Sie die Konten entsprechend einrichten, lässt sich auch entnehmen, für welche Art von Arbeit die Aufwände angefallen sind, z. B. Fehlerbehebung, Besprechungen oder Tests.
- Akkumulierte Aufwände, gruppiert nach Projekt, Kostenstelle, Konto und Person. Dieser Bericht hilft, eine Übersicht zu erhalten, welche Kostenstellen mit welchen Personen für welche Art

von Tätigkeiten Aufwände verbucht haben. Damit erhalten Sie eine Kontrolle und Hinweise für Prozessoptimierungen.

6.8 Prozessperformanz

Wenn mit Kunden Service Level Agreements (SLA) bestehen, müssen Sie nachweisen, dass Ihre Antwortzeiten den Vereinbarungen entsprechen. Auch ohne solche Vereinbarungen möchten Sie evtl. Ihre Durchlaufzeiten optimieren. Die folgenden Berichte liefern die dazu notwendigen Informationen:

- graphische Darstellung der durchschnittlichen Zeit bis zur Erledigung eines Vorgangs, siehe Abb. 6.8
- graphische Darstellung der maximalen Antwortzeit über der Zeit, siehe Abb. 6.9
- Anzahl der Eskalationen über der Zeit

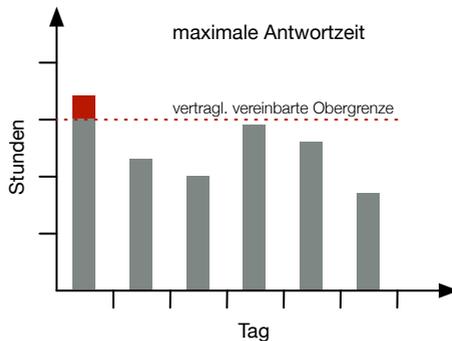


Abbildung 6.8: Durchschnittliche Zeit bis zur Schließung über der Zeit.

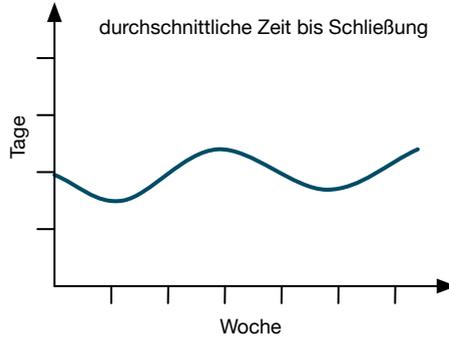


Abbildung 6.9: Durchschnittliche Antwortzeit über der Zeit.

Teil II

Anwendungsszenarien

7

Szenario-Grundlagen

Ich habe eiserne Prinzipien. Wenn sie Ihnen nicht gefallen, habe ich auch noch andere.

Groucho Marx

7.1 Übersicht

In den vorangegangenen Kapiteln haben Sie ein leistungsfähiges Projekt- und Aufgabenmanagement-Planungsmodell kennengelernt. Im zweiten Teil dieses Buchs wird Ihnen anhand von Beispielszenarien die Anwendung dieses Modells zur Lösung konkreter Probleme erläutert. In diesem Kapitel lernen Sie die für alle Anwendungsfälle gleichermaßen gültigen Regeln und Prinzipien kennen. Die darauf folgenden Kapitel beschreiben dann einige spezifische Anwendungsfälle.

7.2 Allgemein gültige Prinzipien

Es gibt einige allgemein gültige Prinzipien, die Sie beim Design Ihres Aufgabenmanagement-Prozesses beachten sollten.

- **Halten Sie es einfach.** Erstellen Sie keine komplizierten Abläufe. Schreiben Sie nicht vor, dass Dokumente erstellt werden, die niemandem außer einem imaginären Qualitätssicherungsbeauftragten dienen. Erstellen Sie nur Dokumente, die für andere oder Sie selbst

nützlich sind. Sie sollten immer in der Lage sein, eine konkrete Person zu benennen, für die ein Dokument erstellt wird. Wenn Ihnen niemand einfällt, lassen Sie das Dokument weg, es erzeugt nur Nebel im Informationsdschungel.

- **Halten Sie es offen.** Schränken Sie nicht ohne Not den Zugriff von Teammitgliedern zu Informationen ein. Es wäre erstaunlich, wenn Sie alle möglichen Situationen vorhersehen könnten. Ermöglichen Sie es Ihrem Team, flexibel und selbstorganisiert zu reagieren. Bevormunden Sie Ihre Mitarbeiter nicht.
- **Fluten Sie Ihr Team nicht mit Information.** Ihre Teammitglieder sollen entscheiden können, welche Informationsströme Sie abonnieren. Geben Sie dem Einholen von Informationen Vorrang vor dem aktiven Verschicken von Informationen.

7.3 Gemeinsame Vorgangsattribute

Tabelle 7.1 stellt die für alle Anwendungsszenarien gültigen Vorgangsattribute dar.

Attribut	Beschreibung	Beispiele
Raum	Raum, zu dem ein Vorgang gehört	Produkt B
Vorgangstyp	Art des Vorgangs	Aufgabe, Aktion
Autor	Person, die den Vorgang angelegt hat	Susie, Micha
Bearbeiter	Person, die den Vorgang bearbeiten soll	Tamas
Manager	Person, die den Vorgang überwacht	Adrian
Titel	Kurzbeschreibung des Vorgangs	
Beschreibung	Beschreibung des Vorgangs	
Zustand	Zustand und Position im Workflow	geöffnet, erledigt
Ablage	Ablage, in der sich der Vorgang befindet	Eingangskorb

Tabelle 7.1: Gemeinsame Vorgangsattribute.

Tabelle 7.2 listet optionale Attribute auf, die in verschiedenen Szenarien verwendet werden können. Es handelt sich um die Phase zur zeitlichen

Strukturierung von Aufgaben-Clustern sowie um zwei der vier RACI-Rollen.

Attribut	Beschreibung	Beispiele
geplante Phase	die Phase, in der der Vorgang erledigt wird	Release A2, Sprint A2.1
Beratende	Personen, die Hinweise geben können	Matthias, Jane
zu Informierende	Personen, die informiert werden müssen	Maria, Kathrin

Tabelle 7.2: Optionale gemeinsame Vorgangsattribute.

Im Allgemeinen ist es vorteilhaft, so wenig Attribute wie möglich zu verwenden. Die dazu gehörende Eingabe und Pflege kostet Zeit und Geld und reduziert die Bereitschaft Ihrer Mitarbeiter, den Prozess einzuhalten.

Es kann helfen, sich die Frage zu stellen, wer von dem zusätzlichen Attribut profitiert und wen es Zusatzarbeit kostet. Der Nutzen eines Attributes sollte die Kosten deutlich überwiegen.

Es sollte nur wenige Attribute geben, bei denen die Kosten ausschließlich bei einer Person und der Nutzen ausschließlich bei einer anderen Person liegen. Wenn Sie als Manager viel Nutzen für sich sehen, aber Ihr Team keinen davon hat, wird die Bereitschaft, die Attribute zu pflegen, irgendwann erschöpft sein.

8

Issue tracking

8.1 Prinzip

In der Softwareentwicklung ist es schon seit vielen Jahren notwendig, Aufgaben und Rückmeldungen in einem Projekt systematisch zu verfolgen. Software-Produkte sind oft komplex. Die Anzahl von Arbeitsschritten und zu erstellenden Zwischenprodukten während der Entwicklungsphase sowie die Problemmeldungen in der Integrationsphase und nach der Auslieferung erreicht schnell ein Ausmaß, das zwingend ein systematisches Aufgabenmanagement erfordert.

Inzwischen werden auch zuvor einfache Produkte komplexer, zumeist, weil der Umfang an darin enthaltener Software stetig wächst. Werkzeugmaschinen, Messgeräte, eingebettete Systeme, mechatronische Systeme und selbst Haushaltsgeräte besitzen einen immer höheren Softwareanteil. Das erfordert bei der Entwicklung und im Support eine entsprechende Vorgehensweise.

Issue tracking konzentriert sich auf nicht planbare Vorgänge, die in der Regel nicht hierarchisch angeordnet werden müssen. Wenn ein Problem auftritt, wird es eingetragen und systematisch bearbeitet, bis der entsprechende Vorgang erledigt werden kann. Ein Vorgang wird üblicherweise einem Projekt oder Produkt zugeordnet.

8.2 Vorgangstypen

Für die Anwendung im issue tracking sind die folgenden Arten von Vorgängen typisch:

- Problembereich (oder Issue)
- Defekt (bzw. Bug oder Fehler)
- Anforderung (oder Verbesserungsvorschlag)
- Aktion

Die Unterscheidung zwischen einem Problembereich und einem Defekt erlaubt eine bessere Identifizierung von Prozessbereichen mit Verbesserungspotential.

Ein **Problembereich** signalisiert eine Abweichung vom erwarteten Verhalten. Das kann z. B. daran liegen, dass das Produkt schwierig zu bedienen ist. Die Ursache könnte z. B. ein suboptimaler Anforderungsentwicklungsprozess sein. Ein Problembereich könnte aber auch auf einen tatsächlichen Fehler hinweisen.

Ein **Defekt** oder **Bug** stellt ein Abweichen vom spezifizierten Verhalten dar. In diesem Fall ist das Produkt tatsächlich fehlerhaft. Das könnte auf einen ungenügenden Implementierungs- und Testprozess deuten.

Eine **Anforderung** beschreibt eine gewünschte Änderung zum bestehenden Produkt oder Produktentwurf. Sie kommt oft von Kunden und schlägt sich im Projekt als neues oder geändertes Leistungsmerkmal nieder. Es kann nützlich sein, noch einmal zwischen einer neuen Anforderung, einer **Anforderungsänderung** und einem **Verbesserungsvorschlag** zu einer bestehenden Funktion zu unterscheiden. Das würde die Analyse des Anforderungsprozesses erleichtern, z. B. hinsichtlich der Stabilität von Anforderungen.

Der Vorgangstyp **Aktion** ist für alles nützlich, was nicht in eine der anderen Kategorien passt. Dazu gehören z. B. Punkte, die aus einer Besprechung resultieren, oder auch Unterpunkte, die von Änderungsanforderungen oder Problembereichen herrühren.

8.3 Prozess

Ein typischer Prozess für issue tracking lässt sich wie folgt beschreiben:

1. Ein Kunde oder ein Testingenieur meldet ein Problem. Das Problem wird einem Support-Ingenieur oder einer Gruppe von Support-Ingenieuren zugewiesen. Im zweiten Fall nimmt sich einer aus der Gruppe des Vorgangs an, sobald er Zeit hat. Der Zustand des Vorgangs ist „geöffnet“.
2. Der Support-Ingenieur prüft, ob das Problem aus einem Missverständnis, einem Problem mit der Dokumentation oder aus einem echten Fehler resultiert. Er prüft auch, ob das gleiche Problem schon einmal gemeldet wurde. Falls nicht, versucht er das Problem zu reproduzieren. Der Vorgangszustand ist „analysiert“.
3. Falls das Problem tatsächlich reproduzierbar ist, wird entschieden, ob es behoben oder ignoriert wird.
4. Wenn das Problem behoben werden soll, weist der Support-Ingenieur den Vorgang einem Entwickler zu. Nun ist der Entwickler der Bearbeiter und für diesen Vorgang verantwortlich. Der Vorgangszustand ist nun „implementieren“. Alternativ kann der Support-Ingenieur auch einen untergeordneten Vorgang erstellen und diesen dem Bearbeiter zuweisen. Das ist z. B. dann nützlich, wenn der Kunde zwar den von ihm gemeldeten Originalvorgang einsehen darf, aber keinen Einblick in die Interna der Bearbeitung erhalten soll.
5. Sobald der Entwickler das Problem behoben hat, weist er den Vorgang einem Test-Ingenieur zu. Der Vorgangszustand steht nun auf „prüfen“.
6. Wenn die Tests erfolgreich waren, setzt der Test-Ingenieur den Vorgangszustand auf „geprüft“. Nach einiger Zeit wird ein neues Release erstellt, das die Fehlerbehebung enthält. Nun wird der Kunde informiert. Der Support-Ingenieur oder der Kunde selbst setzt den Vorgangszustand auf „erledigt“.

Dieser Ablauf ist nur ein Beispiel und es existieren dazu viele Variationsmöglichkeiten. Das generelle Prinzip ist aber immer gleich.

Issue tracking erfordert in der Regel keine hierarchische Organisation von Vorgängen in einer WBS. Issue tracking kann einfache Workflows benötigen, z. B. wenn Sie aus Qualitätssicherungsgründen verhindern möchten, dass jedes Teammitglied einen Vorgang schließen kann, oder wenn Sie Vorgänge automatisch an den jeweils nächsten Bearbeiter weiterleiten möchten.

8.4 Vorgangsattribute

Für das issue tracking sind zusätzlich zu den Standardattributen aus Tabelle 7.1 die in Tabelle 8.1 aufgeführten Attribute hilfreich. Kursiv gesetzte Attribute sind optional.

Vorgangsattribut	Beschreibung	Beispiel
Release festgestellt	die Phase oder das Release, in dem der Fehler festgestellt wurde	Release A2
Release geplant	die Phase oder das Release, mit dem der Fehler behoben sein sollte	Release A3
Priorität	Dringlichkeit, mit der das Problem behoben werden sollte	sofort, bald
<i>Auswirkung</i>	Kosten, wenn das Problem nicht behoben wird	hoch, niedrig
<i>Fälligkeitsdatum</i>	bis wann das Problem behoben sein soll	21.9.2012
<i>Komponente</i>	betroffene Komponenten	CAN driver
<i>Planwert</i>	der geschätzte Aufwand, um das Problem zu beheben	12 hrs
<i>gemeldet von</i>	falls der Autor nicht der Melder ist	J. Guttenberg

Tabelle 8.1: Spezifische Vorgangsattribute für issue tracking.

In vielen Organisationen ist issue tracking ein „best-effort“-Prozess. In einem solchen Fall ist ein Fälligkeitsdatum nicht erforderlich. Das gleiche gilt für den Planwert und die tatsächlichen Aufwände. Wenn der Kunde nicht direkt für den Service bezahlt, lohnt sich diese Buchhaltung nur, um die internen Prozesse zu verbessern.

8.5 Releases

Komplexe Produkte werden in mehreren Schritten entwickelt und verkauft, die wir hier „Releases“ nennen. Aus diesem Grund muss für das issue tracking jeder Vorgang die folgenden zwei Attribute besitzen:

- das Produkt-Release, bei dem ein Problem festgestellt wurde
- das Produkt-Release, für das die Behebung eines Fehlers oder die Implementierung einer Anforderung geplant ist

Während des Entwicklungsprozesses können Releases wieder in mehrere Phasen aufgeteilt werden, z. B. eine Reihe von „Sprints“. So ergibt sich im Laufe der Zeit eine große Anzahl von Phasen, von denen die meisten nach der Freigabe eines Releases nicht mehr interessant sind. Es ist deshalb hilfreich, wenn man solche untergeordneten Phasen in das übergeordnete Release zusammenfassen kann.

8.6 Verknüpfungen

Es gibt Vorgänge, die miteinander in Beziehung stehen, und zwar anders als in einer Eltern-Kind-Beziehung. So kann es z. B. vorkommen, dass das gleiche Problem von mehreren Kunden berichtet worden ist. Die Problemmeldungen deuten auf denselben Fehler und sollten deshalb als Duplikate erkennbar sein. Das kann durch eine entsprechende Verknüpfung der zugehörigen Vorgänge erreicht werden.

Manchmal darf ein Vorgang nur geschlossen werden, wenn ein anderer Vorgang erledigt worden ist. Das ist z. B. nützlich, wenn Sie mit Bibliotheken oder Frameworks arbeiten und ein Fehler in diesen gemeinsam genutzten Teilen bekannt geworden ist. Durch eine entsprechende Verknüpfung können Sie sicherstellen, dass die zugehörigen Problemmeldungen in den einzelnen Produkten erst geschlossen werden können, wenn der Fehler im Framework behoben worden ist.

8.7 Berichte und Auswertungen

Issue tracking ist zumeist ereignisgesteuert: Wenn ein Problem gemeldet wird, muss der Support und eventuell auch die Entwicklung reagieren. In der Regel findet keine Projektplanung und kein Projektmonitoring statt.

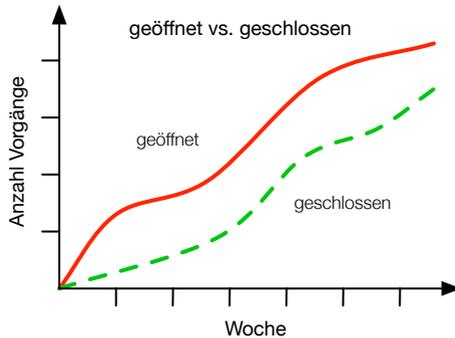


Abbildung 8.1: Geöffnete vs. geschlossene Vorgänge.

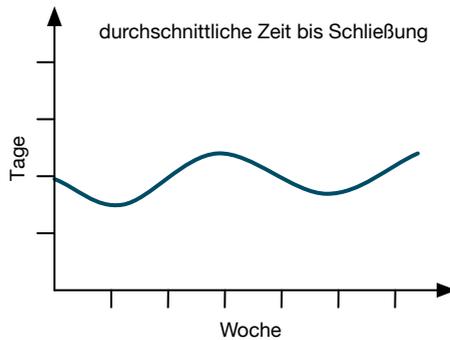


Abbildung 8.2: Durchschnittliche Zeit bis zur Erledigung.

Die Effektivität der zum issue tracking gehörenden Prozesse lässt sich mit einer Reihe von Berichten charakterisieren und optimieren. Die folgenden Berichte können für diesen Anwendungsfall hilfreich sein:

- offene Probleme pro Release
- Support-Aufwand pro Produkt und/oder Kunde

- neue Vorgänge vs. geschlossene Vorgänge über der Zeit
- durchschnittliche Zeit vom Eintreffen eines Vorgangs bis zum Schließen des Vorgangs
- Anzahl der offenen Vorgänge pro Ingenieur
- wie oft wurde ein Vorgang wiedereröffnet
- welche Vorgänge wurden mit welchem Release als erledigt markiert (Release Notes)
- Vorgänge, die schon sehr lange offen sind
- Vorgänge, die in letzter Zeit neu hereingekommen sind

9

Klassisches Projektmanagement

9.1 Übersicht

Unter „klassischem“ Projektmanagement verstehen wir hier vor allem das Management solcher Projekte, die zu einem bestimmten Termin innerhalb eines gegebenen Kostenrahmens ein definiertes Produkt liefern müssen. Sie können also nicht während des Projekts verhandeln, was geliefert werden kann. Häufig ist es auch so, dass die Lieferung von Teilmengen nicht möglich ist. Wenn z. B. die Produktion eines neuen Fahrzeugmodells für den nächsten August geplant ist, können Sie nicht darüber verhandeln, ob das Motorsteuergerät dann schon serienreif sein muss oder nicht, oder ob Sie bis dahin überhaupt eines benötigen.

Klassisches Projektmanagement erfordert eine sorgfältige Planung von Aufwänden für die einzelnen Arbeitsschritte bzw. Teilprodukte. Dadurch ergibt sich schon am Anfang eine große Anzahl an Vorgängen, die verwaltet werden müssen.

Eine der wichtigsten Methoden, große, komplexe Strukturen handhaben zu können, ist die hierarchische Gliederung. Die Bibel berichtet davon, dass Mose von seinem Schwiegervater Jitro Besuch bekam. Jitro fällt auf, dass Mose ganz allein Recht spricht und damit sich und das Volk ermüdet. So gibt er ihm den Rat:

Sieh dich aber unter dem ganzen Volk um nach redlichen Leuten, die Gott fürchten, wahrhaftig sind und dem ungerechten Gewinn Feind. Die setze über sie als Oberste über tausend, über hundert, über fünfzig und über zehn.

2. Mose 18,21

Für die Gliederung wird oft die 7er-Regel angewendet: Auf jeder Hierarchieebene sollten sich in etwa sieben Elemente befinden; es dürfen auch mal drei sein, oder auch zehn. Diese Regel findet man in vielen Organisationen und in der Gliederung der Zeit wieder:

Sechs Tage sollst du arbeiten; am siebenten Tage sollst du ruhen, auch in der Zeit des Pflügens und des Erntens.

2. Mose 34,21

Im Projektmanagement bezeichnen wir die hierarchische Gliederung der einzelnen Arbeitspakete als Arbeitspaketstruktur (WBS) oder auch Produktstruktur (PSP). Eine gute WBS oder ein guter PSP orientieren sich an der Struktur der zu erstellenden Produkte.

9.2 Vorgangstypen

Für das klassische Projektmanagement sind die folgenden Vorgangstypen nützlich:

- **Aufgaben:** alle planbaren Aktivitäten
- **Aktionen:** alle nicht planbaren Aktivitäten, die Aufgaben weiter detaillieren
- **Issues:** Vorgänge, die spontan entstehen und erledigt werden müssen
- **Anforderungen:** wenn Sie sicherheitskritische Produkte entwickeln, benötigen Sie für Zertifizierungen den Nachweis, dass Ihre Anforderungen in das Produkt eingeflossen sind.
- **Meilensteine**

- **Testprotokolle:** wenn Sie sicherheitskritische Produkte entwickeln, benötigen Sie für Zertifizierungen den Nachweis, dass alle Anforderungen durch Tests geprüft worden sind.
- **Risiken:** eine Liste mit den zehn oder zwanzig wichtigsten Risiken

Sie können natürlich weiter differenzieren, wenn Sie tiefere Einblicke in Ihre Prozesse nehmen möchten. Sie könnten z. B. Anforderungsänderungen beobachten, um Ihren Anforderungsentwicklungsprozess zu optimieren. Was Sie noch hinzufügen, hängt davon ab, was Sie damit erreichen wollen und welche Art von Projekten Sie verwalten müssen. Die hier vorgestellte Liste sollte ein guter Startpunkt sein.

9.3 Prozess

Aus der Übersichtsperspektive lassen sich die wesentlichen Aktivitäten des Projektmanagements wie in Abb. 9.1 darstellen. Die Managementaktivitäten gehören zu einer dieser vier Kategorien (siehe auch Abb. 1.1):

- Projektmanagement
- Qualitätsmanagement
- Risikomanagement
- Zulieferer-Management

Es gibt viele unterschiedliche Arten von Projekten wie Entwicklungsprojekte, Beratungsprojekte, Bauprojekte usw. Selbst für eine bestimmte Art von Projekt hängen die Details der Prozesse von der Domäne (Maschinenbau, IT, chemische Industrie), kulturellen und organisatorischen Faktoren ab. Es ginge über die Absicht dieser Publikation hinaus, die allen gemeinsamen Elemente detailliert zu beschreiben. Das meiste, was in diesem Zusammenhang wichtig ist, wurde sehr gut im „Project Management Body of Knowledge“ [Pmb] zusammengefasst.

Eine der größten Herausforderungen für Sie als Projektmanager eines klassischen Projektes ist es, die richtige Detailebene für die zu planenden Aktivitäten zu finden (siehe auch Abschnitt 2.6 auf Seite 33). Ist der Plan zu detailliert, veraltet er schnell und es ist ein hoher Aufwand nötig, ihn aktuell zu halten. Häufig vernachlässigen Sie dann die Aktualisierung

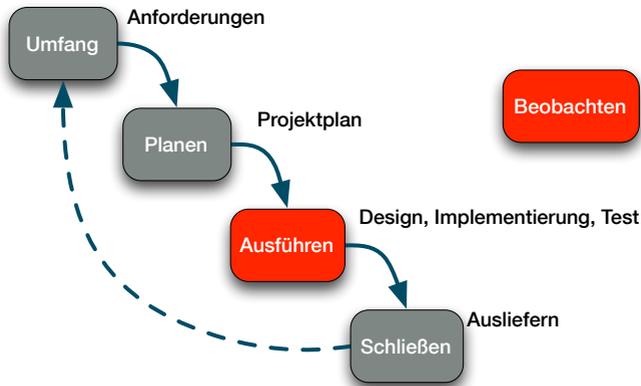


Abbildung 9.1: Die Elemente des PM im traditionellen Wasserfall-Modell.

nach einiger Zeit, weil Sie keine Zeit mehr dazu haben, und der Plan wird ungültig.

Wenn Sie auf der anderen Seite einen Plan erstellen, der zu grob ist, werden Sie große Schwierigkeiten haben, den Projektfortschritt zu beobachten. Wenn es z. B. ein paar Wochen lang keine Meilensteine oder zu liefernde Teilprodukte gibt, fahren Sie ihr Projekt in dieser Zeit blind, egal wie viele Projektbesprechungen Sie ansetzen.

Ab einem bestimmten Detailgrad wird es für Sie schwierig zu verstehen, was zu tun ist. Sie müssen Ihren Experten vertrauen, und diese sollten den Plan für die von ihnen verantworteten Bereiche selbst bestimmen und aktualisieren.

Eine bewährte Methode in dem Konflikt zwischen zu hohem und zu niedrigem Detaillierungsgrad ist die Unterscheidung zwischen geplanten Aktivitäten, die wir hier „**Aufgaben**“ nennen, und ungeplanten Aktivitäten, die wir hier „**Aktionen**“ nennen. Die Aufgaben umfassen einen Zeitraum von ein oder zwei Wochen, während die Aktionen die Aufgaben in Arbeitspakete von ein oder zwei Tagen Dauer gliedern.

Sie können die Detaillierung der Aufgaben vornehmen oder vornehmen lassen, wenn Sie sich dem Zeitpunkt nähern, zu dem eine Aufgabe begonnen werden sollte. Zu diesem Zeitpunkt sind die Randbedingungen

schon besser bekannt als zu Beginn des Projektes und die Planungsgenauigkeit ist höher.

9.4 Vorgangsattribute

Über die in den Tabellen 7.1 und 7.2 aufgeführten Standardattribute hinaus können eine Reihe zusätzlicher Attribute für das klassische Projektmanagement nützlich sein. Diese sind in Tabelle 9.1 dargestellt.

Vorgangsattribut	Beschreibung	Beispiel
PSP	zeigt die Position im PSP	2.3.5
Release	Phase, zu der der Vorgang gehört	Release A2
<i>Komponente</i>	Komponente, zu der der Vorgang gehört	CAN driver
Anfang	Beginn der Arbeit an diesem Vorgang	21.9.2012
Ende	Fertigstellungstermin für diesen Vorgang	21.9.2012
<i>Wunschbeginn</i>	Top-down Beginn der Arbeit an diesem Vorgang	21.9.2012
<i>Wunschende</i>	Top-down Fertigstellungstermin für diesen Vorgang	21.9.2012
Planwert	der geschätzte Aufwand	12 Stunden
aktuelle Kosten	der schon aufgelaufene Aufwand	17 Stunden
geschätzter Restaufwand	der noch zu leistende Aufwand	5 Stunden

Tabelle 9.1: Projektmanagement-spezifische Attribute.

Wichtig sind hier die Attribute, die die EVM unterstützen. Mit ihrer Hilfe lässt sich der Projektfortschritt zuverlässig beobachten und es lassen sich objektive Vorhersagen erstellen. Ebenfalls wichtig ist die Unterstützung durch das PSP-Attribut, das eine hierarchische Gliederung aller Vorgänge erlaubt.

9.5 Releases

Komplexe Produkte werden häufig in verschiedenen Versionen entwickelt und ausgeliefert. Diese Versionen nennen wir „Releases“. Sie

müssen in der Lage sein zu unterscheiden, welcher Vorgang zu welchem Release gehört.

Während eines Entwicklungsprozesses kann es hilfreich sein, Releases selbst wieder zu unterteilen, z. B. in einzelne Muster-Phasen, Sprints oder Builds. So ergibt sich im Laufe der Zeit eine große Anzahl von Phasen, von denen die meisten nach der Freigabe eines Releases nicht mehr interessant sind. Es ist erforderlich, dass man solche untergeordneten Phasen in das übergeordnete Release zusammenfassen kann.

Es kann auch nützlich sein, ein spezielles Release zu definieren mit dem Namen „noch nicht geplant“. In dieses Release stellt man alle Vorgänge ein, für die noch nicht entschieden ist, mit welchem Release sie abgearbeitet werden sollen.

Releases können ein Anfangs- und Endedatum haben, genauso wie Projekte. Sie können auch ein Gesamtbudget besitzen, mit denen die geplanten und aktuell aufgelaufenen Aufwände verglichen werden können.

9.6 Verknüpfungen

Im Projektmanagement können Vorgänge von einander abhängig sein. Hier ist eine Liste möglicher Verknüpfungen (E steht für „Ende“, A steht für „Anfang“):

- **EA-Verknüpfung:** Vorgang A muss X Tage erledigt sein, bevor Vorgang B beginnt
- **AA-Verknüpfung:** Vorgang A muss X Tage nach Beginn von Vorgang B beginnen
- **EE-Verknüpfung:** Vorgang A muss X Tage nach Ende von Vorgang B erledigt sein
- **AE-Verknüpfung:** Vorgang A muss X Tage nach Ende von Vorgang B starten
- Vorgang A ist ein Testfall für Vorgang B, z. B. für die Anforderungsverfolgung.
- Vorgang A ist eine Anforderung für Vorgang B, z. B. für eine Anforderungs-Testmatrix.

- Vorgang A darf nicht geschlossen werden, solange Vorgang B noch offen ist.

Diese Liste ist nicht vollständig, insbesondere wenn Sie die Anwendung im Projektmanagement mit Anwendungen als Helpdesk oder für das issue tracking kombinieren.

9.7 Berichte und Auswertungen

Ein großer Teil aller Aktivitäten im Projektmanagement befasst sich mit dem Beobachten des Projektfortschritts, der Identifikation von Problemen, und der geeigneten Reaktion auf Änderungen und Störungen. Das Berichtswesen im Projekt fokussiert sich deshalb auf diese Bereiche.

Abb. 9.2 zeigt zwei ausdrucksstarke Grafiken, die den Projektfortschritt kompakt zusammenfassen. Das Earned-Value-Diagramm wurde schon in Abschnitt 6.2 auf Seite 66 besprochen.

Die Meilenstein-Trendanalyse in Abb. 9.3 illustriert, wohin sich die Meilensteine des Projekts auf der Zeitachse bewegen. Im besten Fall wären alle Linien horizontal oder hätten eine negative Steigung.

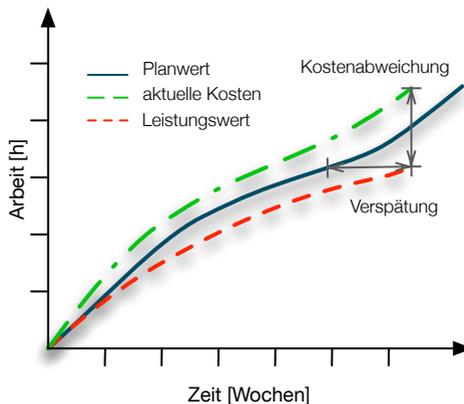


Abbildung 9.2: Earned-Value-Diagramm zur Fortschrittsbeobachtung.

Die Tabelle in Abb. 9.4 ist gut für wöchentliche Projektbesprechungen geeignet. Diese Tabelle visualisiert fällige und überfällige Vorgänge für die

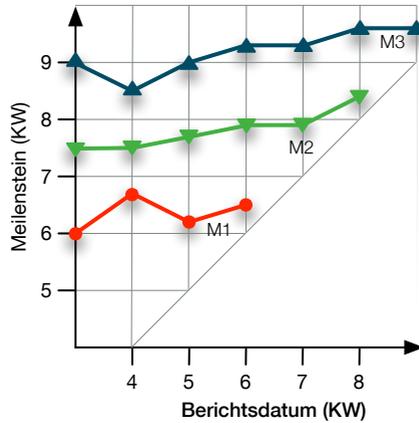


Abbildung 9.3: Meilenstein-Trendanalyse.

nächste Zeit und markiert kritische Vorgänge, d. h. solche, bei denen das Budget voraussichtlich überschritten wird. Darüber hinaus sind folgende Berichte hilfreich für wöchentliche oder monatliche Projektstatustreffen:

		Anz. Vorg.	PW [h]	AK [h]	GRA [h]	Erlad. [%]	Verb. [%]
überfällig	Vorgänge mit mir als Bearbeiter	2	7	4	3	57	100%
	Vorgänge mit mir als Manager	4	14	12	4	67	114%
	alle überfälligen Vorgänge	7	26	19	9	68	108%
diese Woche	Vorgänge mit mir als Bearbeiter	3	12	3	8	27	92%
	Vorgänge mit mir als Manager	8	82	34	45	67	96%
	alle diese Woche fälligen Vorg.	16	135	23	128	15	119%
nächste Woche	Vorgänge mit mir als Bearbeiter	2	7	4	3	57	100%
	Vorgänge mit mir als Manager	4	14	12	4	67	114%
	alle nächste Woche fälligen Vorg.	7	26	19	9	68	108%
andere	Vorgänge mit mir als Bearbeiter	2	7	4	3	57	100%
	Vorgänge mit mir als Manager	4	14	12	4	67	114%
	alle anderen Vorgänge	7	26	19	9	68	108%
	Summe überfällig:		9 h				
	Summe fällig diese Woche:		128 h				

Abbildung 9.4: Budget-Übersicht.

- in der letzten Woche erledigte Vorgänge, gruppiert nach Bearbeiter
- diese Woche fällige Vorgänge, gruppiert nach Bearbeiter
- überfällige Vorgänge, gruppiert nach Bearbeiter
- geleistete Arbeit je Bearbeiter während der letzten Woche oder dem letzten Monat
- geplante Arbeit je Bearbeiter für die nächste Woche oder den nächsten Monat
- Vorgänge, die von anderen Vorgängen abhängen und deren Zeitpuffer negativ geworden ist (z. B. EA-Verknüpfung, kritische Pfadanalyse)
- Anforderungen ohne zugehörige Testfälle
- Testfälle ohne zugehörige Anforderungen
- Release Notes, also z. B. mit einem Release behobene Fehler und implementierte Anforderungen

10

Scrum

10.1 Übersicht

Scrum ist ein Prozessrahmen, um Projekte „agil“ zu managen. Scrum ist am besten für die Entwicklung von Produkten und Anwendungen geeignet. Scrum hält sich nicht an das klassische Wasserfallmodell, sondern erfordert ein inkrementelles und iteratives Vorgehen.

Auf der Zeitachse strukturiert Scrum alles, was zu tun ist, in kurze Iterationen. In Scrum wird eine solche Iteration „**Sprint**“ genannt. Nach [Sch11] ist ein Sprint so definiert:

„Ein Sprint ist ein Zeitfenster von einem Monat oder kürzer, in dem ein nutzbares und potentiell auslieferbares Produktinkrement erzeugt wird. Sprints haben während eines Entwicklungsvorhabens eine gleichmäßig feste Dauer.“

Ein neuer Sprint beginnt direkt, nachdem der vorherige abgeschlossen ist.

Nach Scrum teilen Sie ihre Projektmannschaft in kleine, selbst-organisierende und funktionsübergreifende Teams auf. Ihr Produkt bauen Sie aus vielen kleinen Einzelteilen zusammen, wobei jedes Einzelteil einen eigenen, nachprüfbaren und mit einem Wert versehenen Beitrag zum Gesamtprojekt leistet und evtl. separat geliefert werden kann. Die Teams nehmen

sich der Reihe nach die Entwicklung der Einzelteile vor und arbeiten die Liste ab, bis ein nützliches Gesamtprodukt entstanden ist.

10.2 Prozess-Übersicht

Das zentrale Steuerungselement in Scrum ist eine priorisierte Liste von Anforderungen, genannt das „**Produkt-Backlog**“. Das Produkt-Backlog basiert auf Eingaben von Interessenvertretern (*Stakeholder*) und wird vom „**Produkteigentümer**“ (*Product Owner*) verwaltet.

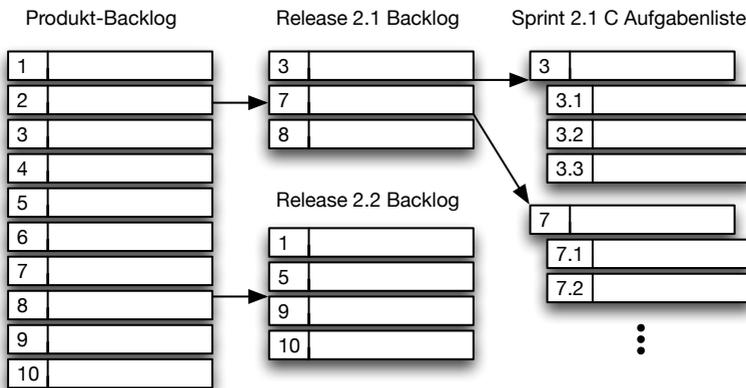


Abbildung 10.1: Scrum-Backlogs und Sprint-Aufgabenliste.

Zu Beginn eines jeden Sprints selektiert das funktionsübergreifende Team unter Berücksichtigung der Priorisierung Einträge aus dem Produkt-Backlog. Das Team verpflichtet sich, die mit den Einträgen verbundenen Ergebnisse am Ende des Sprints zu liefern. Während ein Sprint läuft, darf die Liste der ausgewählten Einträge von niemandem verändert werden.

Nachdem die Teams die mit einem Sprint zu liefernden Ergebnisse ausgewählt haben, detaillieren sie diese weiter in einzelne Arbeitspakete oder Aufgaben. Diese detaillierte Liste wird „**Sprint-Aufgabenliste**“ genannt.

Die Liste an Aufgaben in einem Sprint, die noch nicht abgearbeitet ist, heißt „**Sprint-Backlog**“. Bei größeren Projektvorhaben ist es möglich, das

Leistungsmerkmal	Priorität	Wert	Initial gesch. Aufwand	gesch. Restaufwand		
				Sprint 1	Sprint 2	...
Workflow-Editor	1	70	80 h	80 h	0 h	
Excel-Export	2	30	10 h	10 h	7 h	
Diagramme	3	50	12 h	14 h	18 h	

Tabelle 10.1: Produkt-Backlog Beispiel.

Projektergebnis in mehreren Ausbaustufen oder „**Releases**“ zu liefern. In einem solchen Fall gibt es noch ein „**Release-Backlog**“, wie in Abb. 10.1 dargestellt.

Der Projektfortschritt wird täglich in kurzen Teambesprechungen („**Daily Scrum**“) festgehalten. Wenn erforderlich werden Anpassungen vorgenommen, um die verbliebene Arbeit termingerecht erledigen zu können.

Produkt- Backlog- Eintrag	Sprint Aufgabe	Initial gesch. Aufwand	gesch. Restaufwand			
			Tag 1	Tag 2	...	Tag n
Workflow- Editor	WED Design	10 h	10 h	2 h	...	
	WED Impl.	60 h	60 h	60 h	...	
	WED Test	10 h	10 h	10 h	...	
Excel-Export	Export Design	5 h	2 h	0 h	...	
	Export Impl.	20 h	18 h	2 h	...	
	Export Test	5 h	5 h	5 h	...	
Diagramme	Grafik	4 h	0 h	0 h	...	
	Layout	8 h	6 h	0 h	...	
	Datenformate	2 h	2 h	0 h	...	
Summe		124 h	113 h	79 h	...	

Tabelle 10.2: Sprint-Backlog.

10.3 Sprint

Ein Sprint (siehe Abb. 10.2) besteht aus einer Reihe von Ereignissen und Aktivitäten, wie im Folgenden beschrieben.

- **Sprint-Planungstreffen:** Bevor ein Sprint gestartet wird, selektiert das Scrum-Team, bestehend aus dem Produkteigentümer, dem Entwicklungsteam und dem Scrum-Master aus dem Produkt-Backlog oder dem Release-Backlog eine Anzahl von Leistungsmerkmalen. Das Team verpflichtet sich dazu, die mit diesen Merkmalen zusammen hängenden Ergebnisse am Ende des Sprints zu liefern.

Dabei berücksichtigt jedes Teammitglied seine eigene zeitliche Verfügbarkeit während des Sprints. Das Team erstellt einen Plan, genannt Sprint-Backlog. Darin wird detailliert beschrieben, wie jedes Leistungsmerkmal in diesem Sprint in ein funktionales Inkrement für das Endprodukt überführt wird.

Das Sprint-Planungstreffen dauert nicht länger als acht Stunden für einen Sprint von einem Monat Dauer. Die Hälfte der Zeit wird auf die Frage, was getan werden soll, verwendet. Die andere Hälfte der Zeit dient der Beantwortung der Frage, wie vorgegangen werden soll.

- **Daily Scrums:** Dies sind kurze Treffen des Entwicklungsteams, die nicht länger als 15 Minuten dauern dürfen. Während des Daily Scrum werden die Aktivitäten der letzten 24 Stunden betrachtet, mögliche Hindernisse diskutiert, und die Aktivitäten der nächsten 24 Stunden geplant.
- **Entwicklungsarbeit:** Das ist die eigentliche Arbeit, um die gewünschten Ergebnisse zu erzielen.
- **Sprint-Review:** Am Ende eines Sprint kommen das Scrum Team sowie die Interessenvertreter zusammen und prüfen, was in dem Sprint erreicht wurde. Das Entwicklungsteam präsentiert, welche Backlog-Einträge abgearbeitet werden konnten und berichtet über die dabei aufgetretenen Schwierigkeiten. Basierend auf diesem Review wird das Produkt-Backlog überarbeitet und für das nächste Sprint-Planungstreffen aktualisiert. Das Sprint-Review ist ein informelles Treffen von höchstens vier Stunden Dauer für einen Sprint

von einem Monat.

- **Sprint-Rückschau:** Am Ende eines Sprints kommt das Scrum-Team zusammen, um den eigenen Prozess zu inspizieren, beim Sprint Dazugelerntes zusammenzufassen und Verbesserungspotential zu identifizieren. Die Sprint-Rückschau ist ein maximal dreistündiges Treffen für einen Sprint von einem Monat.

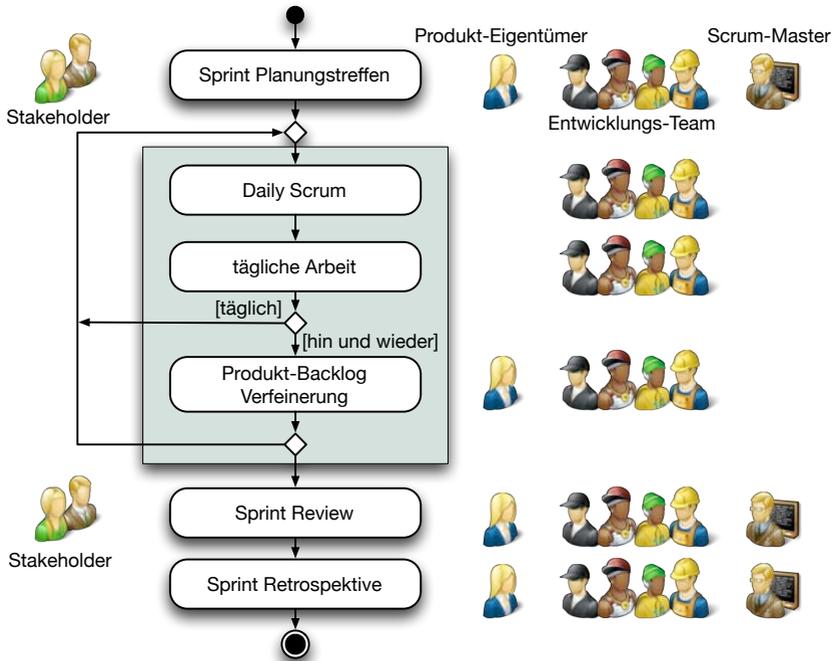


Abbildung 10.2: Scrum Sprint.

10.4 Rollen

Scrum definiert eine Reihe von Rollen, die im Folgenden beschrieben werden sollen.

- **Interessenvertreter:** Interessenvertreter können z. B. Kunden, Manager, Nutzer, das Scrum-Team selbst oder Produktmanager sein.

Sprint-Länge:	2 Wochen		
Arbeitstage im Sprint:	8 Tage		
Teammitglied	verfügbare Tage im Sprint	verfügbare Std. pro Tag	Summe Stunden
Daniel	8	6	48
Babette	4	5	20
Kathrin	3	7	21
Maria	6	3	18

Tabelle 10.3: Sprint-Planung.

- Produkteigentümer:** Der Produkteigentümer ist eine Person, die für den Wert des Produktes verantwortlich ist. Der Produkteigentümer verhandelt mit den Interessenvertretern die Anforderungsliste. Er oder sie ist die einzig verantwortliche Person für den Inhalt und die Reihenfolge der Einträge im Produkt-Backlog und Release-Backlog. Der Produkteigentümer ist verantwortlich dafür, dass aus der vom Entwicklungsteam geleisteten Arbeit maximaler Nutzen entsteht.
- Entwicklungsteam:** Das Entwicklungsteam besteht aus drei bis neun Personen und sollte alle Fähigkeiten umfassen, die zur Erstellung des Produktinkrements erforderlich sind. Das Team ist selbst-organisierend und bestimmt völlig autark, welche Leistungsmerkmale aus dem Produkt-Backlog in einem Inkrement von ihm geliefert werden.
- Scrum-Master:** Der Scrum-Master ist eine [Sch11] „Diener-Führer“-Rolle und hilft dem Entwicklungsteam und der Organisation, den Scrum-Prozess zu verstehen und richtig umzusetzen.

10.5 Backlogs verwalten

Backlogs helfen in Scrum, die zu liefernden Ergebnisse entlang der Zeitachse zu ordnen. Leistungsmerkmale im Produkt-Backlog kann man verstehen als „noch nicht eingepflanzt“. Einträge im Release-Backlog bedeu-

ten „eingepflichtet für Release x.y“. Merkmale im Sprint-Backlog bedeuten „wird mit Sprint x.y.z geliefert“.

So gesehen kann jedes Backlog als eine „Phase“ gemäß unserer früheren Definition (siehe Abschnitt 4.2 auf Seite 57 und Abschnitt 2.7 auf Seite 34) verstanden werden. Releases sind Unterphasen von Projekten und Sprints sind Unterphasen von Releases.

10.6 Vorgangstypen

Für Scrum benötigen wir drei Arten von Vorgängen:

- Backlog-Einträge
- Aufgaben
- Issues

Für die Backlog-Einträge ist vom Kontext her klar, dass es sich um Einträge im Produkt-Backlog, Release-Backlog oder Sprint-Backlog handelt. Aufgaben sind Teil der Sprint-Aufgabenliste und Issues sind alle solche Dinge, die auftreten und nicht geplant werden konnten.

Es empfiehlt sich, Aufgaben als untergeordnete Elemente von Produkt- oder Release-Backlog-Einträgen zu modellieren.

10.7 Vorgangsattribute

Über die in Tabelle 7.1 aufgeführten Standardattribute sind für Scrum eine Reihe zusätzlicher Attribute hilfreich. Diese sind in Tabelle 10.4 zusammengefasst. Kursiv gesetzte Attribute sind optional. Der **Wert** kann eine beliebige Zahl auf einer benutzerdefinierten Skala sein, z. B. von 0 bis 100. Zusammen mit anderen Attributen kann er benutzt werden, um die Einträge im Produkt-Backlog oder Release-Backlog zu ordnen. Eine mögliche Formel für diese Ordnung ist

$$\text{Position} = \frac{\text{Wert} \cdot \text{Priorität}}{\text{Planwert} \cdot \text{Risiko}}$$

Vorgangsattribut	Beschreibung	Beispiel
Wert	der relative Wert dieses Merkmals für die Interessenvertreter	1, 80
Phase	das Backlog oder der Sprint, für den dieses Merkmal geplant ist	Sprint A3.1, Release B4
Priorität	eine Zahl für die Dringlichkeit eines Merkmals	10, 5, 1
Risiko	eine Zahl für die Wahrscheinlichkeit eines Scheiterns der Implementierung	5%, 15%
<i>Notwendigkeit</i>	1 wenn das Merkmal unverzichtbar ist, sonst 0	0, 1
Planwert	der anfängliche Schätzwert für den Umsetzungsaufwand	12 h
Geschätzte Restarbeit	der momentane Schätzwert für den verbleibenden Umsetzungsaufwand	6 h

Tabelle 10.4: Scrum-spezifische Vorgangsattribute.

Die **Priorität** ist eine positive Zahl auf einer benutzerdefinierten Skala, typisch zwischen 0 und 10. Sie drückt die Dringlichkeit eines Leistungsmerkmals aus. Es kann hilfreich sein, einen Eintrag für absolut notwendige Leistungsmerkmale zu reservieren, z. B. mit einem Wert von 1000. Das bedeutet, dieses Merkmal muss unter allen Umständen so schnell wie möglich implementiert werden.

Es kann hilfreich sein, den Wert und die Priorität auf einer nichtlinearen Skala auszudrücken, um die Unterschiede klarer zu machen. Eine gute Sequenz ist z. B. 0, 1, 2, 3, 5, 8, 13, 20, 30, 50, 100.

Die **Notwendigkeit** ist optional. Es ist besser, notwendige Merkmale über eine spezielle Priorität zu markieren als über ein zusätzliches Attribut.

Für Erklärungen zu dem Planwert und den geschätzten Restaufwand schauen Sie bitte in Abschnitt 6.2 auf Seite 66 nach. Scrum verwendet nicht die tatsächliche Arbeit, nur die geschätzte Restarbeit wird berücksichtigt.

10.8 Verknüpfungen

Auch in Scrum-Projekten können Vorgänge voneinander abhängig sein. Hier ist eine Liste möglicher Verknüpfungen, entsprechend der für ein klassisches Projekt (E steht für „Ende“, A steht für „Anfang“):

- **EA-Verknüpfung:** Vorgang A muss X Tage erledigt sein, bevor Vorgang B beginnt
- **AA-Verknüpfung:** Vorgang A muss X Tage nach Beginn von Vorgang B beginnen
- **EE-Verknüpfung:** Vorgang A muss X Tage nach Ende von Vorgang B erledigt sein
- **AE-Verknüpfung:** Vorgang A muss X Tage nach Ende von Vorgang B starten
- Vorgang A ist ein Testfall für Vorgang B, z. B. für die Anforderungsverfolgung
- Vorgang A ist eine Anforderung für Vorgang B, z.B. für eine Anforderungs-Testmatrix
- Vorgang A darf nicht geschlossen werden, solange Vorgang B noch offen ist

Diese Liste ist nicht vollständig, insbesondere wenn Sie die Anwendung im Scrum-Prozess mit Anwendungen als Helpdesk oder für das issue tracking kombinieren.

10.9 Berichte und Auswertungen

Den Projektfortschritt beobachten Sie in Scrum anhand der geschätzten Restarbeit. Dieser Wert muss in jedem Sprint, jedem Release und für das Gesamtprodukt in Echtzeit sichtbar sein. Eine bewährte Methode zur Darstellung des Fortschritts ist in Scrum das Burndown-Diagramm, wie in Abb. 10.3 illustriert.

Für die Daily Scrums hat sich die Scrum-Tafel wie in Abb. 10.4 gezeigt bewährt. Die zu liefernden Leistungsmerkmale werden für jeden Sprint

auf der Tafel angebracht. So wie sie durch den Prozess weiterbewegt werden, werden sie auch auf der Tafel verschoben. Verschiedene Farben können z. B. für verschiedene Teammitglieder stehen oder die Farben können auch mit einer anderen Kategorisierung wie einer Komponente oder einem Kontext verknüpft werden.

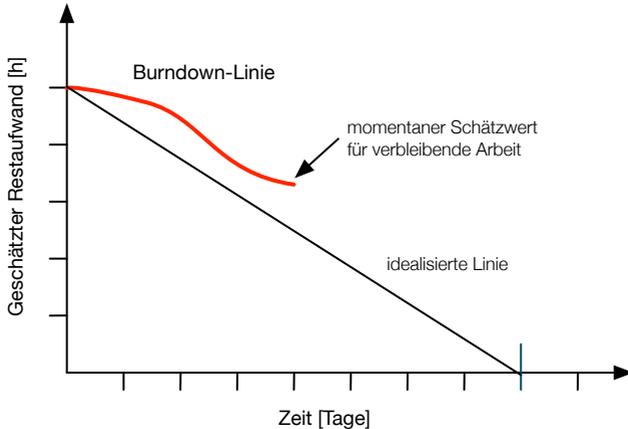


Abbildung 10.3: Scrum-Burndown-Diagramm.

offen	in Arbeit	erledigt
Vorgang 3.5	Vorgang 7.3	Vorgang 3.1
Vorgang 3.6	Vorgang 3.3	Vorgang 3.2
Vorgang 3.7	Vorgang 3.4	Vorgang 7.1
Vorgang 7.4		Vorgang 7.2
Vorgang 7.5		

Abbildung 10.4: Scrum-Tafel.

Darüber hinaus können die folgenden Berichte nützlich sein:

- Für Planungsprozesse ist es hilfreich, einen Bericht zur verfügbaren Kapazität eines jeden Teammitglieds über den für einen Sprint oder

ein Release geplanten Zeitraum zu haben. Es genügt u. U. schon, auch nur die schon verplante Kapazität für diesen Zeitraum zu betrachten, und dann manuell mit der verfügbaren Kapazität abzugleichen.

- Für die Daily Scrums ist eine Liste mit den während der letzten 24 Stunden erledigten Leistungsmerkmalen und Aufgaben nützlich.
- Ein Bericht mit der Liste der noch unerledigten Leistungsmerkmale und Aufgaben für ein spezifisches Produkt, für ein Release oder einen Sprint dient als Übersicht über die verschiedenen Backlogs.

10.10 Die Essenz

Scrum wird vor allem in der Softwareentwicklung eingesetzt, und dort vorwiegend in der Anwendungsentwicklung. Einige Proponenten der Methode treten recht missionarisch auf und beschwören die buchstabengetreue Befolgung der Methode als Voraussetzung für den Erfolg. Welche Prozesselemente sind wirklich gut und wo sind die Grenzen von Scrum?

- Scrum eignet sich gut für einen „best effort“-Entwicklungsprozess, bei dem der Liefertermin eines Produktes oder der Leistungsumfang verhandelbar ist. Wenn Sie z. B. ein komplettes Motorsteuergerät mit Hardware und Software nächsten März abliefern müssen, werden sie Scrum mit klassischen Planungsmethoden ergänzen müssen.
- Scrum biete Vorteile, wenn das Gesamtprodukt sich aus Leistungsmerkmalen so zusammensetzt, dass auch schon eine Teilmenge der Merkmale marktfähig ist. Das ist bei vielen softwareintensiven Produkten leider nicht der Fall (z. B. Motorsteuergeräte, Fahrradtacho).
- Die Rhythmisierung des Prozesses in überschaubare Teilschritte erhöht die Planungsgenauigkeit und lässt früh erkennen, ob der Plan auch eingehalten werden kann.
- Die Strukturierung und Priorisierung der Leistungsmerkmale in der Art, dass jedes Merkmal mit einem Wert versehen wird und für sich nachprüfbar sein muss, hilft bei der Fokussierung auf das Wesentliche.

- Die täglichen kurzen Besprechungen fördern den Teamgeist und Informationsaustausch. Eventuell auftretende Probleme werden frühzeitig erkannt.
- Die abschließende Reflexion der Teams zur Erkennung möglicher Verbesserungen im Prozess und zur Konservierung der gewonnenen Erfahrungen unterstützt einen kontinuierlichen Verbesserungsprozess.

11

Kanban

11.1 Übersicht

Der Kanban-Kerngedanke ist recht einfach: Begrenzen Sie die Anzahl unerledigter Aufgabenpunkte und beginnen Sie erst wieder mit etwas Neuem, wenn die Menge offener Punkte unter eine vorher definierte Grenze gefallen ist.

Mit Kanban organisieren Sie Ihre Arbeit entlang eines Arbeitsablaufs (*engl. workflow*) mit einer Anzahl von „Stationen“. Beispiele für solche Stationen sind „zu erledigen“, „in Arbeit“, „erledigt“, „zugewiesen“, „im Test“, „Konstruktion“, „Entwicklung“, „Versand“, oder auch „Peter“, „Paul“ und „Marie“.

Sie messen die Zeit, die Sie benötigen, um einen Vorgang von seinem Anfangszustand in seinen Endzustand bzw. von seiner ersten Workflow-Station bis zu seiner letzten Workflow-Station zu bringen. Diese Zeit wird „**Zykluszeit**“ genannt, und Sie optimieren Ihren Prozess so, dass diese Zeit minimiert wird.

Ein wesentliches Kanban-Prinzip lautet [And10]:

Wenn sich unerledigte Vorgänge an einer Workflow-Station stauen, müssen alle Teammitglieder helfen, diesen Engpass zu beseitigen.

Kanban benutzt Signalkarten (*japanisch kanban*), um die jeweils aktuelle Situation zu visualisieren (siehe Abb. 11.1).

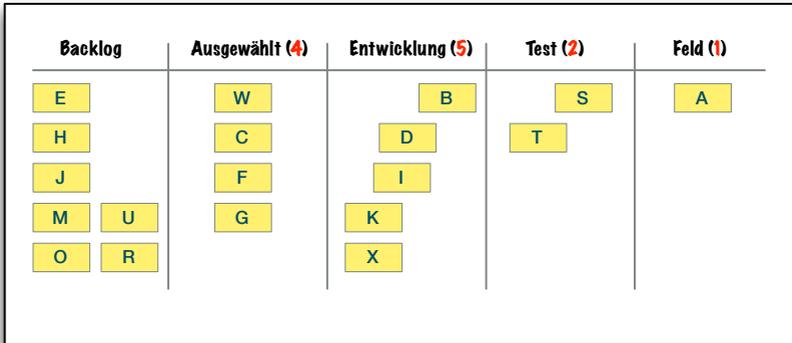


Abbildung 11.1: Kanban-Tafel.

In diesem Beispiel haben wir fünf Workflow-Stationen: „Backlog“, „Ausgewählt“, „Entwicklung“, „Test“ und „Feld“. Die roten Nummern hinter den Stationen bezeichnen die maximale Anzahl von Vorgängen, die sich an dieser Station befinden dürfen.

Nehmen wir einmal an, die Leute in der Entwicklung arbeiten richtig schnell und könnten Vorgang B zur Testabteilung geben. Allerdings haben wir im Test schon zwei Vorgänge, und mehr sind nicht erlaubt.

Die Entwickler können nun keinen weiteren Vorgang mehr von der „Ausgewählt“-Station holen, da sie selbst auch schon die maximale Anzahl an Vorgängen bei sich haben, nämlich fünf. Sie sind damit dazu verurteilt, eine Pause einzulegen. Noch besser wäre es, wenn Sie den Leuten in der Testabteilung helfen würden, ihre Vorgänge zu erledigen.

Kanban ist keine vollständige Projektmanagement-Methode, sondern lediglich ein singuläres Prinzip, das auf eine ganze Reihe von Prozessen angewendet werden kann. Sie können dieses Verfahren gut mit traditionellen Methoden und auch mit Scrum kombinieren.

11.2 Vorgangsattribute

Kanban erfordert keine eigenen Vorgangsattribute.

11.3 Berichte und Auswertungen

Mit Kanban ist es wichtig, jederzeit die Anzahl an offenen Vorgängen je Workflow-Station sichtbar machen zu können. Eine Darstellung ähnlich zu Abb. 11.1 ist dafür hilfreich.

Als ein Ziel in Kanban versuchen Sie, Ihre Durchlaufzeiten bzw. Zykluszeiten zu minimieren. Dazu ist ein Bericht zur durchschnittlichen Durchlaufzeit nützlich.

12

Technischer Support

12.1 Übersicht

Im IT-Bereich ist es schon lange üblich, Rückmeldungen in einem Projekt oder zu Produkten und Dienstleistungen systematisch zu verfolgen. Inzwischen werden auch einst hauptsächlich mechanisch geprägte Produkte zunehmend mit Software versehen. Dadurch erhöht sich sowohl die interne Komplexität als auch die Komplexität in der Anwendung und die Kunden haben einen höheren Unterstützungs- und Beratungsbedarf.

Das hier beschriebene Szenario stellt den Prozess zwischen Produktanwendern und Produktentwicklern dar. Auf der Anwenderseite geht es um Unterstützung bei Problemen mit einem Produkt. Diese Probleme müssen u.U. durch die Produktentwicklung gelöst werden, z. B. wenn es sich um systematische Produktfehler handelt.

Der Prozess ist ähnlich dem des issue tracking.

12.2 Vorgangstypen

Für die Anwendung im technischen Support sind die folgenden Arten von Vorgängen typisch:

- Problembereich (oder Issue)

- Defekt (bzw. Bug oder Fehler)
- Anforderung (oder Verbesserungsvorschlag)
- Aktion

Die Unterscheidung zwischen einem Problembericht und einem Defekt erlaubt es, Prozessbereiche mit Verbesserungspotential genauer zu identifizieren.

Ein **Problembericht** signalisiert eine Abweichung vom erwarteten Verhalten. Das kann z. B. daran liegen, dass das Produkt schwierig zu bedienen ist. Die Ursache könnte ein suboptimaler Anforderungsentwicklungsprozess sein. Ein Problembericht könnte aber auch auf einen tatsächlichen Fehler hinweisen.

Ein **Defekt** oder **Bug** stellt eine Abweichung vom spezifizierten Verhalten dar. In diesem Fall ist das Produkt tatsächlich fehlerhaft. Das könnte daran liegen, dass etwas mit dem Implementierungs- und Testprozess nicht stimmt.

Eine **Anforderung** beschreibt eine gewünschte Änderung zum bestehenden Produkt oder Produktentwurf. Sie kommt oft von Kunden und schlägt sich im Projekt als neues oder geändertes Leistungsmerkmal nieder. Es kann nützlich sein, noch einmal zwischen einer neuen Anforderung, einer **Anforderungsänderung** und einem **Verbesserungsvorschlag** zu einer bestehenden Funktion zu unterscheiden. Das würde die Analyse des Anforderungsprozesses erleichtern, z. B. hinsichtlich der Stabilität von Anforderungen.

Der Vorgangstyp **Aktion** ist für alles nützlich, was nicht in eine der anderen Kategorien passt. Das können z. B. Punkte sein, die aus einer Besprechung resultieren, oder auch Unterpunkte, die von Änderungsanforderungen oder Problemberichten herrühren.

12.3 Prozess

Ein typischer Prozess im technischen Support lässt sich wie folgt beschreiben:

1. Ein Kunde meldet Probleme z. B. bei der Inbetriebnahme eines Produktes. Der Kunde wendet sich an seinen Vertriebsingenieur oder direkt an den 1st-Level-Support.
2. Das Problem wird erfasst, der Zustand steht auf „erfasst“.
3. Wenn das Problem nicht direkt gelöst werden kann, wird es an den 2nd-Level-Support weitergeleitet.
4. Der 2nd-Level-Support nimmt das Problem an und setzt den Zustand auf „in Bearbeitung“. Er analysiert das Problem, z. B. indem er versucht, es zu reproduzieren.
5. Es stellt sich heraus, dass ein Produktfehler vorliegt, und es wird ein entsprechender Untervorgang mit Priorisierung geöffnet. Der Kunde wird direkt oder über den Vertriebsingenieur informiert. Auch das Produktmanagement und die Qualitätssicherung werden unterrichtet.
6. Die Entwicklung kümmert sich um die Behebung des Fehlers. Der Zustand des ursprünglichen Vorgangs wird auf „in Bearbeitung–Bug Fixing“ gesetzt.
7. Die modifizierte Version geht zurück an den 2nd-Level-Support und wird dort verifiziert. Der Zustand wird nach erfolgreichem Test auf „getestet“ gesetzt und dem Produktmanagement und der Qualitätssicherung zur Freigabe vorgelegt.
8. Nach der Freigabe durch das Produktmanagement und die Qualitätssicherung wird das modifizierte Produkt in die Fertigung gegeben. Der Zustand des Vorgangs wird auf „erledigt“ gesetzt.

Ein ähnliches Szenario ergibt sich, wenn kein Produktfehler vorliegt, sondern ein Problem im Zusammenspiel mit nicht standardkonformen Komponenten anderer Hersteller. Hier könnte ein typisches Szenario so aussehen:

1. Ein Kunde meldet Probleme z. B. bei der Inbetriebnahme eines Produktes. Der Kunde wendet sich an seinen Vertriebsingenieur oder direkt an den 1st-Level-Support.
2. Das Problem wird erfasst, der Zustand steht auf „erfasst“.

3. Wenn das Problem nicht direkt gelöst werden kann, wird es an den 2nd-Level-Support weitergeleitet.
4. Der 2nd-Level-Support nimmt das Problem an und setzt den Zustand auf „in Bearbeitung“. Er analysiert das Problem, z. B. indem er versucht, es zu reproduzieren.
5. Es stellt sich heraus, dass das Problem auf ein nicht standardkonformes Verhalten des Produktes eines anderen Herstellers zurückzuführen ist.
6. Der 2nd-Level-Support findet einen Workaround. Er informiert den Kunden, die Vertriebsingenieure und das Produktmanagement entsprechend. Der Vorgang wird auf den Zustand „abgeschlossen“ gesetzt und in einen FAQ-Raum überführt.
7. In Zukunft können die Vertriebsingenieure und der 1st-Level-Support in dem FAQ-Raum nachschauen und so direkt eine Problemlösung anbieten.

Es gibt selbstverständlich viele Variationsmöglichkeiten zu diesem Ablauf. Das generelle Prinzip ist aber immer gleich.

Technischer Support erfordert keine hierarchische Organisation von Vorgängen in einer WBS. Technischer Support kann einfache Workflows benötigen, z. B. wenn Sie aus Qualitätssicherungsgründen verhindern möchten, dass jedes Teammitglied einen Vorgang schließen kann, oder wenn Sie Vorgänge automatisch an den jeweils nächsten Bearbeiter weiterleiten möchten.

12.4 Vorgangsattribute

Für den technischen Support sind zusätzlich zu den Standardattributen aus den Tabellen 7.1 und 7.2 die in Tabelle 12.1 aufgeführten Attribute nützlich. Kursiv dargestellte Attribute sind optional.

Technischer Support ist meistens ein „best-effort“-Prozess. In manchen Fällen gibt es service level agreements (SLAs). Dann muss eine Eskalation des Problems je nach Verursacher der Problemmeldung nach einer gewissen Zeit automatisch sichergestellt werden.

Vorgangsattribut	Beschreibung	Beispiel
Release festgestellt	die Phase oder das Release, in dem der Fehler festgestellt wurde	Release A2
Release geplant	die Phase oder das Release, mit dem der Fehler behoben sein sollte	Release A3
Priorität	Dringlichkeit, mit der das Problem behoben werden sollte	sofort, bald
<i>Auswirkung</i>	Kosten, wenn das Problem nicht behoben wird	hoch, niedrig
<i>Fälligkeitsdatum</i>	bis wann das Problem behoben sein soll	21.9.2012
<i>Komponente</i>	betroffene Komponenten	CAN driver
<i>Planwert</i>	der geschätzte Aufwand, um das Problem zu beheben	12 hrs
<i>Gemeldet von</i>	falls der Autor nicht der Melder ist	S. Musterfrau

Tabelle 12.1: Spezifische Vorgangsattribute für technischen Support.

12.5 Releases

Komplexe Produkte werden in mehreren Schritten entwickelt und verkauft, die wir hier „Releases“ nennen. Aus diesem Grund muss für den technischen Support jeder Vorgang die folgenden zwei Attribute besitzen:

- das Produkt-Release, bei dem ein Problem festgestellt wurde
- das Produkt-Release, für das die Behebung eines Fehlers oder die Implementierung einer Anforderung geplant ist

Während des Entwicklungsprozesses können Releases wieder in mehrere Phasen aufgeteilt werden, z. B. eine Reihe von „Sprints“. So ergibt sich im Laufe der Zeit eine große Anzahl von Phasen, von denen die meisten nach der Freigabe eines Releases nicht mehr interessant sind. Es ist deshalb hilfreich, wenn man solche untergeordneten Phasen im übergeordneten Release zusammenfassen kann.

12.6 Verknüpfungen

Es gibt Vorgänge, die miteinander in Beziehung stehen, und zwar anders als in einer „Eltern-Kind-Beziehung“. So kann es z. B. vorkommen, dass das gleiche Problem von mehreren Kunden berichtet worden ist. Die Problemmeldungen deuten auf denselben Fehler und sollten deshalb als Duplikate erkennbar sein. Das kann durch eine entsprechende Verknüpfung der zugehörigen Vorgänge erreicht werden.

Manchmal darf ein Vorgang nur geschlossen werden, wenn ein anderer Vorgang erledigt worden ist. Das ist z. B. nützlich, wenn Sie mit Bibliotheken oder Frameworks arbeiten und ein Fehler in diesen gemeinsam genutzten Teilen bekannt geworden ist. Durch eine entsprechende Verknüpfung können Sie sicherstellen, dass die zugehörigen Problemmeldungen in den einzelnen Produkten erst geschlossen werden können, wenn der Fehler im Framework behoben worden ist.

12.7 Berichte und Auswertungen

Technischer Support ist zumeist ereignisgesteuert: Wenn ein Problem gemeldet wird, muss der Support und eventuell auch die Entwicklung reagieren. In der Regel findet keine Projektplanung und kein Projektmonitoring statt. Wenn es SLAs gibt, muss ein Nachweis geführt werden, dass die vereinbarten Reaktionszeiten eingehalten werden.

Die Effektivität der zum technischen Support gehörenden Prozesse lässt sich mit einer Reihe von Berichten charakterisieren und auch optimieren. Die folgenden Berichte können für diesen Anwendungsfall hilfreich sein:

- offene Probleme pro Release
- Support-Aufwand pro Produkt und/oder Kunde
- neue Vorgänge vs. geschlossene Vorgänge über der Zeit
- durchschnittliche Zeit von Eintreffen eines Vorgangs bis zum Schließen des Vorgangs
- Anzahl der offenen Vorgänge pro Ingenieur
- wie oft wurde ein Vorgang wieder eröffnet

- welche Vorgänge wurden mit welchem Release erledigt (Release Notes)
- Vorgänge, die schon sehr lange offen sind
- Vorgänge, die in letzter Zeit neu hereingekommen sind

13

Mini-Projekte

13.1 Übersicht

Viele Organisationen arbeiten immer wieder ähnliche Projekte ab, die vom Umfang her überschaubar sind und eigentlich nur aus einer Anzahl von zehn bis zwanzig größtenteils vorgegebener Arbeitsschritte bestehen.

Oft lohnt es sich nicht, dafür ein komplettes Workflow-System aufzusetzen, zumal die Fähigkeit erhalten bleiben muss, im Einzelfall auch außerhalb der Standardabläufe reagieren zu können. Es wäre auch völlig übertrieben, wenn man für solche Mini-Projekte ein komplettes Projektmanagement etablieren würde. Die Verwaltung über Excel-Listen kommt allerdings schnell an Grenzen, wenn mehrere Personen oder sogar Organisationseinheiten involviert sind.

Zur Illustration nehmen wir ein Beispiel aus dem technischen Vertrieb bzw. dem Pre-Sales Support.

Es gibt Produkte, die immer kundenspezifisch angepasst werden müssen. Relativ früh im Verkaufsprozess müssen die spezifischen Kundenanforderungen analysiert werden und zur Angebotserstellung müssen die Ingenieure und Spezialisten beim Lieferanten konsultiert werden. Ein typischer Arbeitsablauf könnte so aussehen:

1. Ein Kunde sendet eine Anfrage. Dadurch entsteht beim Lieferanten ein kleines „Projekt“ mit einem vorgegebenen Arbeitsablauf.

2. Die Anfrage wird an einen Pre-Sales-Support-Ingenieur und einen Verkaufsmanager gegeben.
3. Der Ingenieur prüft mit der Konstruktionsabteilung, ob die Anfrage durch Anpassen eines existierenden Produktes oder eine Neukonstruktion erfüllt werden kann, und erstellt zusammen mit den Beteiligten eine Aufwandsabschätzung.
4. Sobald der Support-Ingenieur alle technischen Fragen geklärt hat, versorgt er den Vertriebsmanager mit allen erforderlichen Informationen zur Erstellung des Angebots.
5. Der Vertriebsmanager erstellt das Angebot und schickt es an den Kunden.
6. Der Kunde bestellt das Produkt.
7. Das Produkt wird konstruiert.
8. Das Produkt wird hergestellt und konfiguriert.
9. Das Produkt wird an den Kunden verschickt.

In diesem Ablauf gibt es eine Anzahl offizieller Meilensteine und Besprechungen mit definierten Entscheidungspunkten und Checklisten.

In einem solchen Szenario ist es nicht sinnvoll, für jede Anfrage ein „Projekt“ bzw. einen Projektraum aufzumachen, da ja bei jeder Anfrage die gleichen Leute involviert sind und der zeitliche Umfang kein iteratives Vorgehen erfordert.

Am besten wird ein solches Mini-Projekt als ein Elternvorgang modelliert, und alle Arbeitsschritte als Kindvorgänge dieses Elternvorgangs. Der Elternvorgang wäre vom Vorgangstyp „Projekt“ und die Arbeitsschritte von einem anderen Vorgangstyp, z.B. „Aufgabe“, „Aktion“ oder „Besprechung“.

Wenn es mehrere Typen von Mini-Projekten gibt, können diese über unterschiedliche Vorgangstypen abgebildet werden, wenn Sie unterschiedliche Vorgangsattribute und Eingabemasken erfordern.

Da jedes Projekt zunächst einmal nach einem Standardablauf bearbeitet wird, ist es hilfreich, sich dafür Vorlagen anzulegen, die schon sämtliche Arbeitsschritte, Meilensteine und Checklisten beinhalten. Die Vorlagen

können in einem eigenen Vorlagen-Projektraum verwaltet werden und bei Initiierung eines Projektes in den Arbeits-Projektraum kopiert werden. Beim Kopieren müssen alle Zuordnungen von Bearbeitern und Managern sowie alle Termine entsprechend umgesetzt werden.

Literaturverzeichnis

- [And10] David J. Anderson, *Kanban: Successful Evolutionary Change for Your Technology Business*, Blue Hole Press, 2010
- [Bib84] Die Bibel nach der Übersetzung Martin Luthers, revidierte Fassung von 1984, Deutsche Bibelgesellschaft, Stuttgart, 1999
- [Bre98] Berthold Brecht, *Dreigroschenroman*, Suhrkamp, Frankfurt am Main, 1998
- [Bro95] Frederick P. Brooks, *The Mythical Man-Month*, Addison-Wesley Professional, 1995
- [CMD] Software Engineering Institute, *CMMI for Development, Version 1.3*, CMU/SEI-2010-TR-033, Pittsburg, PA November, 2010
- [CMS] Software Engineering Institute, *CMMI for Services, Version 1.3*, CMU/SEI-2010-TR-034, Pittsburg, PA November, 2010
- [Coh05] Mike Cohn, *Agile Estimating and Planning*, Prentice Hall, New York, 2005
- [Cov04] Stephen R. Covey, *The 7 Habits of Highly Effective People*, 1. Ausgabe, Free Press, New York, 2004
- [Doe92] Dietrich Dörner, *Die Logik des Misslingens*, Rowohlt, Reinbek bei Hamburg, 1992
- [Dru08] Peter F. Drucker, *The Essential Drucker: The Best of Sixty Years of Peter Drucker's Essential Writings on Management*, HarperBusiness, New York, 2008
- [GTD] David Allen, *Getting Things Done*, 1. Ausgabe, Piatkus, London 2001

- [Har05] Keith Harrison-Broninski, *Human Interactions: The Heart And Soul Of Business Process Management: How People Really Work And How They Can Be Helped To Work Better*, Meghan Kiffer Pr, 2005
- [Hir11] Dieter Hirsch und Norbert Marschall, *Was Sie schon immer über Projektmanagement wissen wollten...*, Norbert Marschall Consulting, Mötzingen, 2011
- [IAB09] IABG, *The V-Model XT*, 2009
- [JaK09] Mike Jacka und Paulette Keller, *Business Process Mapping: Improving Customer Satisfaction*, John Wiley and Sons, 2009
- [Kor12] Der Koran, deutsche Übersetzung,
<http://www.koransuren.de/koran/sure78.html>
- [Kni10] Henrik Kniberg und Mattias Skarin, *Kanban and Scrum - making the most of both*, C4Media, 2010
- [Ma36] Thomas Mann, *Joseph und seine Brüder*, Band 2: *Joseph in Ägypten*, S. Fischer, 1936
- [Pmb] Project Management Institute, PMI Standards Committee, *A Guide to the Project Management Body of Knowledge (PMBOK)*, 4. Ausgabe, Newton Square, PA, USA, 2008
- [Pri10] Kim H. Pries und Jon M. Quigley, *Scrum Project Management*, CRC Press, 2010
- [Sch11] Ken Schwaber und Jeff Sutherland, *The Scrum Guide*,
<http://www.scrum.org/Portals/0/Documents/Scrum Guides/Scrum Guide-DE.pdf>, Oktober 2011
- [SCH73] E. F. Schumacher, *Die Rückkehr zum menschlichen Maß*, Rowohlt, Reinbeck bei Hamburg, 1977

Bildquellennachweis

1. Umschlagbild: Getty Images 90620129, Hugh Sitton.
2. Alle Icons: Icon Experience V-Collection.
3. Alle anderen Abbildungen und Tabellen: Eigene Darstellung.

Abkürzungen

- AC aktuelle Kosten, *engl. actual cost*
- DBS Produktstrukturplan, *engl. deliverables breakdown structure*
- ERW geschätzter Restaufwand, *engl. estimated remaining work*
- EVM Leistungswertmethode, *engl. earned value method*
- FAQ häufig gestellte Fragen, *engl. frequently asked questions*
- GTD Getting-Things-Done
- MAP beobachten und planen, *engl. monitoring and planning*
- PM Projektmanagement
- PSP Projektstrukturplan
- PV Planwert, *engl. planned value*
- SLA service level agreement
- WBS Arbeitspaketstruktur, *engl. work breakdown structure*

Index

A	
Abfall	44, 50
Ablage	
Abfall	50
Brutkasten	51
delegiert	50
Eingangskorb	42, 47
geplante Vorgänge	48
nächste Aktionen	48
Referenz	52
Ablage – ablegbar	
Abfall	44
Brutkasten	44
Referenz	44
Ablage zu bearbeiten	
delegiert/wartend	44
Kalender	44
nächste Aktionen	44
ablegbar	43
Abstraktion	32
AC	66
actual cost	66
actual planned value	67
Aktivität	29
aktuelle Kosten	66
aktueller Aufwand	66
aktueller Planwert	67
Anforderung	82, 116
Anforderungsänderung	82, 116
Arbeitsablauf	111
Kanban	111
Arbeitspaketstruktur	90
Attribute	45
Auftragsbestand	48
Aufwand	31
aktueller	66
Aufwandsabweichung	67
B	
Backlog	48
BCWP	67
BCWS	66
Bearbeiter	59
Berater	60
Berichte	
Kategorien	65
Scrum	107
Brutkasten	44, 51
Budget	37
Budget-Übersicht	95
Budgetvorgabe	37
Bug	82, 116
Bug Tracking	81, 115
Burndown-Diagramm	107
D	
Daily Scrum	101, 102
datumsspezif. Aktionen	49
datumsspezif. Inform.	50
Defect Tracking	81, 115
Defekt	82, 116
delegiert	50
delegiert/wartend	45
Detailierungsgrad	33
Dringlichkeit	47

Durchfluss	33	Framework	
		Verknüpfung	85, 120
E			
earned value	67	G	
Earned-Value-Diagramm	95	Gegendruck	53
Earned-Value-Methode	66	geplante Vorgänge	44, 48
Eingangskorb	42, 47	geschätzter Restaufwand	66
Einschränkungen	94, 107	Gliederung	
Endtermin		hierarchisch	89
Vorgabe	37	Glossar	61
zugesagt	37	Gruppe	57
Entwicklungsteam		H	
Scrum	104	Helpdesk	115
estimated remaining work	66	Hierarchie	29, 30, 89
EV	67	Hotspot	69
EVM	66	I	
AC	66	Informationsaustausch	62
actual cost	66	Interessenvertreter	
actual planned value	67	Scrum	104
aktuelle Kosten	66	Issue	82, 116
aktueller Planwert	67	Issue Tracking	81
Aufwandsabweichung	67	Issue tracking	
BCWP	67	Prozess	83
BCWS	66	Iteration	48
CD	67	iteration	
cost deviation	67	Scrum	99
earned value	67	K	
estimated remaining work	66	Kalender	44
EV	67	datumsspezif. Aktionen	49
geschätzter Restaufwand	66	datumsspezif. Inform.	50
Leistungswert	67	Verabredungen	49
Planned Value	66	Kanban	111
Planwert	66	Arbeitsablauf	111
PV	66	Zykluszeit	111
Excel	23	Kanban-Tafel	111
F			
Flusskontrolle	53		

planbar	32
Sichtbarkeit	83
Verknüpfung	52, 85, 120
zu bearbeiten	42
Vorgangstyp	41
Vorgangszustand	83, 116
analyzed	83
erledigt	83
geöffnet	83
geprüft	83
implementieren	83
prüfen	83

W

Wasserfall-Modell	91
WBS	29, 48
Wichtigkeit	47
Work Breakdown Structure	29
Workflow-Station	111
Kanban	111
Wunschtermin	37

Z

Zeitbeziehung	52
Zeitpuffer	52
zu bearbeiten	42
zu Informierender	60
Zugangskontrolle	55, 59
Gruppe	57
Organisationseinheit	57
zugesagter Endtermin	37
zugesagter Starttermin	37
Zwischenziel	34
Zykluszeit	111

track⁺
Task Management Software

Steinbeis GmbH & Co.KG
STZ Task Management Systems
Eugen-Ruoff-Str. 30
71404 Korb

www.trackplus.de