

Information Technology Project Management

by Kathy Schwalbe · Language: DE

Summary by booksummaryai.com —

<https://booksummaryai.com/country/de/information-technology-project-management-zusammenfassung>

Summary

In 60 Sekunden

Kathy Schwalbe führt dich durch das Projektmanagement für IT-Arbeit, Kapitel für Kapitel, von der Definition, was ein Projekt ist, bis zum Abschluss. Sie verankert alles in den Prozessgruppen und Wissensgebieten des PMBOK Guide und zeigt dann, wie sich jedes davon auf Software- und Systemarbeit anwenden lässt. Daten der Standish Group zeigen, warum das wichtig ist: Die meisten Softwareprojekte überschreiten weiterhin Budgets, Termine oder beides. Das Buch gibt dir konkrete Werkzeuge an die Hand – Schätzungen, Earned Value, Risikomatrizen, RACI-Charts –, damit du Projekte führst, statt von ihnen geführt zu werden.

Warum es sich lohnt

Wenn du mit IT-Projekten zu tun hast – als Manager, Entwickler, Analyst oder Student –, dann ist das aus gutem Grund das Standardlehrbuch. Es übersetzt abstrakte PMBOK-Rahmenwerke in softwarespezifische Praxis, mit echten Zahlen zu Fehlerraten und Schritt-für-Schritt-Werkzeugen, die du sofort anwenden kannst. Lies es, um eine Zertifizierung zu bestehen, dein erstes Projekt zu leiten oder aufzuhören dabei zuzusehen, wie Budgets und Termine davonlaufen.

Die Kernideen

Ein Projekt ist vorübergehend, einzigartig und wird an drei Zielen gleichzeitig gemessen

Schwalbe beginnt damit, festzulegen, worüber sie überhaupt lehrt. Ein Projekt, sagt sie, ist 'ein zeitlich befristetes Vorhaben, das unternommen wird, um ein einzigartiges Produkt, eine Dienstleistung oder ein Ergebnis zu schaffen', und Projektmanagement ist 'die Anwendung von Wissen, Fähigkeiten, Werkzeugen und Techniken auf Projektaktivitäten, um die Projektanforderungen zu erfüllen' – beides direkt aus dem PMBOK Guide. Von dort aus führt sie die magisches Dreieck ein: Jedes Projekt balanciert Umfang, Zeit und Kosten, üblicherweise als Dreieck dargestellt mit der Qualität im Zentrum. Drückt man eine Seite zusammen, verschieben sich die anderen. Diese Geometrie rahmt jedes der folgenden Kapitel.

Das magische Dreieck ist täuschend einfach. Menschen behandeln Umfang, Zeitplan und Budget als getrennte Regler und wundern sich dann, wenn das Anziehen des einen den anderen lockert. Die Qualität sitzt im Zentrum, weil sie den Druck von allen drei Seiten absorbiert. Sobald man

Projekte so betrachtet, wird jede Statusbesprechung zu einer Frage: Welches Eck des Dreiecks gibt nach – und was kostet das an Qualität?

Probier es aus: Zeichne das Dreieck für dein aktuelles Projekt und beschrifte, welche Nebenbedingung derzeit am stärksten unter Druck steht.

Zehn Wissensgebiete, fünf Prozessgruppen, ein chaotisches Ineinander

Stell dir einen neuen Projektmanager mit dem PMBOK Guide in der Hand vor, der auf zehn Wissensgebiete starrt – Umfang, Zeitplan, Kosten, Ressourcen, Kommunikation, Risiko, Beschaffung, Stakeholder, Qualität und Integrationsmanagement – und sich fragt, wie sie zusammenpassen. Dann kommt die zweite Liste: fünf Prozessgruppen – Initiierung, Planung, Ausführung, Überwachung und Steuerung, Abschluss. Der Neuling nimmt eine ordentliche Fließbandproduktion an: erst die Planung abschließen, dann ausführen. Schwalbe stoppt ihn sofort. Die Prozessgruppen überlappen sich in der Praxis. Du führst aus, während du noch planst; du überwachst, während du abschließt; du gehst zurück, wenn sich die Bedingungen ändern. Echte Projekte sind keine Pipeline; sie sind Jonglierakt, und der Rahmen ist ehrlich genug, das zuzugeben.

Die Wissensgebiete sind eine Checkliste alles dessen, was ein Projekt zum Scheitern bringen kann; die Prozessgruppen sagen, wann man sich um jedes davon kümmert. Anfänger erwarten eine lineare Pipeline – planen, dann tun, dann prüfen. Schwalbes Punkt, dass sich die Gruppen überlappen, ist wichtig, weil er Iteration erlaubt: Du planst weiter, während du ausführst, und überwachst weiter, während du abschließt. Das Integrationsmanagement verbindet die anderen neun miteinander – deshalb existiert es überhaupt.

Probier es aus: Ordne die Aktivitäten deines aktuellen Projekts den fünf Prozessgruppen zu und markiere, wo zwei gleichzeitig laufen.

Die CHAOS-Zahlen: Warum Softwareprojekte immer wieder scheitern

Ein 'Media Snapshot' im einleitenden Kapitel liefert den Weckruf. Die CHAOS-Studien der Standish Group verfolgen die Ergebnisse von Softwareprojekten, und der Bericht von 2015 fand, dass nur 29 % der Softwareprojekte erfolgreich waren, 19 % vollständig scheiterten und der Rest angefochten war – verspätet, über Budget oder zusammengestrichen. Schwalbe greift auch weiter zurück: Die ursprüngliche CHAOS-Studie von 1994 fand, dass nur 16 % der Softwareprojekte pünktlich und im Budget abgeschlossen wurden. Sie weist darauf hin, dass diese düsteren Statistiken von 1994 das Interesse an Projektmanagement-Zertifizierungen befeuerten.

Zwei Jahrzehnte Messung, und die Erfolgsquote hat sich ungefähr verdoppelt – von 16 % auf 29 % – was Fortschritt und Anklage zugleich ist. Fast jedes fünfte Projekt scheitert weiterhin vollständig. Die Zahlen rechtfertigen die gesamte Disziplin: Wenn Softwareprojekte von Natur aus erfolgreich wären, bräuhete niemand Zertifizierungen, Prozessgruppen oder dieses Buch. Scheitern in diesem Ausmaß ist ein Managementproblem, nicht nur ein Programmierproblem.

Probier es aus: Schlage die letzten fünf Projekte deiner Organisation nach und stufe jedes als erfolgreich, angefochten oder gescheitert ein – dann zähle.

Dein Organigramm entscheidet, wie viel Autorität du tatsächlich hast

Bevor es um irgendeine Technik geht, lässt Schwalbe dich darauf schauen, wo du sitzt. Sie beschreibt vier Organisationsstrukturen – funktional, Matrix (in schwacher, ausgewogener und starker Ausprägung) und projektorientiert – und erklärt, wie sich die Autorität des Projektmanagers in jeder unterscheidet. In einer funktionalen Struktur haben Projektmanager wenig Gewicht; in einer projektorientierten halten sie echte Macht über Menschen und Budgets. Die Matrix-Varianten liegen dazwischen, wobei die Autorität davon abhängt, ob die Matrix schwach, ausgewogen oder stark ist.

Gleiches Projekt, gleicher Plan, völlig unterschiedliche Erfolgschancen – je nach Struktur. Ein Projektmanager in einer funktionalen Organisation verhandelt um Ressourcen; in einer projektorientierten verfügt er über sie. Deine Struktur zu kennen, sagt dir, welche Kämpfe du auswählst und woher dein Einfluss tatsächlich kommt – aus formaler Autorität, Überzeugungskraft oder dem Wohlwollen der Funktionsmanager, die dein Team kontrollieren.

Probier es aus: Ermittle, welche Struktur deine Organisation verwendet, und notiere deine drei größten Einflussquellen innerhalb davon.

Der SDLC: vier Phasen von der Idee bis zum laufenden System

Schwalbe verankert IT-Projekte im Systementwicklungslebenszyklus (SDLC) und beschreibt dessen vier Phasen: Systemplanung, Systemanalyse, Systementwurf und Systemimplementierung. Die Planung fragt, ob sich der Bau des Systems lohnt; die Analyse erfasst, was es leisten muss; der Entwurf legt fest, wie; die Implementierung baut es und liefert es aus. Dieses Framework gibt IT-Projekten eine gemeinsame Sprache, die generischen Projektmanagement-Texten fehlt.

Für Projektmanager ist der SDLC wichtig, weil jede Phase andere Risiken und andere Ergebnisse (Deliverables) mit sich bringt. Probleme, die in der Planung entdeckt werden, kosten weit weniger als Probleme, die erst in der Implementierung auftauchen. Wer die Phasen kennt, kann die Intensität seines Projektmanagements an die tatsächliche Position des Projekts anpassen – früh stark auf Machbarkeit fokussiert, spät stark auf Kontrolle.

Probier es aus: Ordne dein aktuelles Projekt einer SDLC-Phase zu und liste die Ergebnisse auf, die fällig sind, bevor es in die nächste Phase übergehen kann.

Drei Schätzungen für drei Momente im Leben eines Projekts

Schwalbe beschreibt drei Arten von Kostenschätzungen und wann jede im Projektlebenszyklus am Platz ist. Die Grobschätzung (Rough Order of Magnitude, ROM) kommt früh, wenn wenig bekannt ist und weite Spannen akzeptabel sind. Die Budgetschätzung folgt, wenn sich die Planung verfestigt. Die definitive Schätzung kommt spät und ist präzise genug, um echtes Geld

zu binden. Jede Art passt zur Informationsmenge, die in ihrer Phase verfügbar ist.

Der Fehler besteht darin, zu ROM-Zeitpunkten definitive Zahlen zu verlangen oder zu definitiven Zeitpunkten ROM-Schätzwerte zu akzeptieren. Die Genauigkeit der Schätzung sollte mit dem wachsenden Wissen des Projekts steigen. Führungskräfte, die Budgets auf Basis früher Grobzahlen einfrieren, zementieren den Fehler; Projektmanager, die endlos neu schätzen, legen sich nie fest. Die Disziplin besteht darin, den Schätzungstyp an die Lebenszyklusphase anzupassen.

Probier es aus: Prüfe, auf welcher Art von Schätzung dein aktuelles Budget beruht, und ob das zum Fortschrittsstand des Projekts passt.

Earned Value: die Mathematik, die dir sagt, ob du Geld verbrennst

Schwalbe erklärt das Earned-Value-Management, den technischen Kern des Kostenkapitels. Du erfasst geplanten Wert (PV), Earned Value (EV) und Ist-Kosten (AC), berechnest dann die Kostenabweichung als $CV = EV - AC$, die Terminabweichung als $SV = EV - PV$ und den Kostenleistungsindex als $CPI = EV/AC$. Diese Formeln vergleichen, was du ausgeben wolltest, was du erreicht hast und was du tatsächlich ausgegeben hast – zum selben Zeitpunkt.

Klassisches Tracking vergleicht Ausgaben mit dem Budget und sagt dir nichts über den Fortschritt. Ein Projekt kann unter dem Budget liegen und massiv im Rückstand sein – oder im Zeitplan liegen und Geld verlieren. Earned Value zwingt die drei Zahlen in ein einziges Bild: Ein CPI unter 1 bedeutet, dass jeder Dollar weniger als einen Dollar Arbeit kauft. Es ist das Nächste, was das Projektmanagement zu einer Warnleuchte im Cockpit hat.

Probier es aus: Berechne diese Woche den CPI deines Projekts: Earned Value geteilt durch Ist-Kosten. Unter 1 bedeutet Ärger.

PERT: drei Schätzungen schlagen eine

Im Termin-Kapitel beschreibt Schwalbe die Program Evaluation and Review Technique (PERT). PERT nimmt drei Dauerschätzungen – optimistisch, am wahrscheinlichsten und pessimistisch – und verschmilzt sie zu einem gewichteten Durchschnitt: $(\text{optimistisch} + 4 \times \text{am wahrscheinlichsten} + \text{pessimistisch}) / 6$. Die Formel gewichtet die wahrscheinlichste Schätzung stark, während Best- und Worst-Case das Ergebnis in Richtung Realität ziehen.

Punktschätzungen verbergen die Unsicherheit; PERT legt sie offen, weil du Best- und Worst-Case benennen musst. Die Gewichtung zugunsten des wahrscheinlichsten Werts hält den Durchschnitt bodenständig, während die Ränder das Risiko einfangen, das optimistische Planer ignorieren. Nach drei Zahlen zu fragen verändert auch das Gespräch – die Menschen denken gründlicher darüber nach, was schiefgehen kann, wenn sie einen pessimistischen Fall benennen müssen.

Probier es aus: Nimm deine nächste Aufgabenschätzung und erstelle alle drei PERT-Zahlen statt nur einer.

RACI: ein Ende für das Problem 'Ich dachte, du hast das'

Schwalbe beschreibt die Verantwortungsmatrix und ihre bekannteste Form, das RACI-Diagramm. Für jede Aufgabe zeigt RACI, wer Verantwortlich (Responsible), Rechenschaftspflichtig (Accountable), Konsultiert (Consulted) und Informiert (Informed) ist. Die Matrix macht explizit, was Teams normalerweise der Annahme überlassen – wer genau die Arbeit macht, wem das Ergebnis gehört, wer gefragt werden muss und wer nur Bescheid wissen muss.

Die meisten Koordinationsfehler sind Eigentumsfehler. Zwei Menschen hielten jeweils den anderen für zuständig; niemand war Accountable. Die Kraft des RACI-Diagramms liegt in seiner Unverblümtheit: eine Accountable-Person pro Aufgabe, schriftlich benannt. Wer eines für ein Problemprojekt ausfüllt, findet die Lücken meist innerhalb einer Stunde – Aufgaben ohne Besitzer oder drei Personen, die alle überzeugt waren, bloß informiert zu sein.

Probier es aus: Erstelle ein RACI-Diagramm für dein nächstes Ergebnis und prüfe, dass jede Aufgabe genau eine Accountable-Person hat.

Teams storm before they perform—and conflict has five resolutions

Schwalbe describes the Tuckman model of team development: forming, storming, norming, performing, and adjourning. Teams do not gel immediately; they clash before they click. She pairs this with five conflict resolution modes—confrontation solving, compromise, smoothing, forcing, and withdrawal—and notes that confrontation is generally the best approach. Not confrontation as shouting, but as directly working the problem together.

Storming is a stage, not a crisis. Managers who panic at early friction and force harmony often freeze teams in forming. Likewise, smoothing and withdrawal feel kind but leave problems intact; confrontation—naming the issue and solving it—actually resolves conflict. Knowing the stages lets you treat friction as progress and choose your response deliberately instead of reflexively.

Probier es aus: Name your default conflict mode from the five, then pick confrontation deliberately for your next disagreement.

What actually motivates people: Maslow, Herzberg, and Theory X vs Y

Schwalbe brings in three classic motivation frameworks. Maslow's hierarchy of needs arranges human drives from basic physiological needs up through safety, belonging, esteem, and self-actualization. Herzberg's motivation-hygiene theory separates factors that prevent dissatisfaction—like working conditions—from factors that genuinely motivate, like achievement and recognition. McGregor's Theory X and Theory Y contrasts managers who assume people avoid work with those who assume people seek responsibility.

The frameworks converge on one point: you cannot motivate people with hygiene factors alone, and your assumptions about people become self-fulfilling. A Theory X manager who watches

every hour gets workers who work only when watched. Herzberg explains why fixing salaries and offices stops complaints without creating engagement—motivation comes from the work itself, recognition, and growth.

Probier es aus: Audit your team's last month: how much effort went to hygiene factors versus genuine motivators like recognition?

Score your risks before they score you

Schwalbe describes using a probability-impact matrix to prioritize risks, calculating a risk factor or risk score for each identified threat. The matrix forces two questions per risk: how likely, and how bad? Scores let you rank risks instead of treating them all as equally terrifying. She then lays out four response strategies—risk avoidance, risk acceptance, risk transfer, and risk mitigation—and explains when each is appropriate.

Unranked risk lists are anxiety documents; scored ones are decision documents. The four responses map to different situations: avoid when the risk is intolerable, mitigate when you can shrink it, transfer when someone else absorbs it better, accept when the cost of responding exceeds the exposure. Choosing deliberately beats defaulting to worry—or to ignoring risks entirely.

Probier es aus: List your top five project risks and assign each one of the four response strategies by name.

Contracts: who carries the cost risk?

Schwalbe describes three types of contracts and when each fits. Fixed price or lump sum contracts put cost risk on the seller, who agrees to deliver for a set amount. Cost reimbursable contracts put risk on the buyer, who pays actual costs plus something extra. Time and material contracts sit between, paying for effort and supplies as consumed. The choice determines who loses money when estimates go wrong.

Contract type is risk allocation in disguise. Buyers love fixed price until scope shifts and sellers pad every quote to cover the uncertainty they now carry. Cost reimbursable fits genuinely uncertain work but removes the seller's incentive to economize. Matching contract type to how well the work can be specified is the entire decision.

Probier es aus: Check your current vendor contracts and identify who absorbs the cost risk if estimates run over.

Das Machtgitter: Stakeholder nach Quadranten managen

Schwalbe beschreibt das Machtgitter, das Stakeholder danach einordnet, wie viel Macht sie über das Projekt haben und wie stark sie sich dafür interessieren. Jeder Quadrant verlangt eine andere Einbindungsstrategie – Stakeholder mit hoher Macht und hohem Interesse werden eng geführt; bei denen mit geringer Macht und geringem Interesse hält sich der Aufwand in

Grenzen. Das Gitter verwandelt eine unüberschaubare Stakeholderliste in eine handhabbare Karte, die zeigt, wo sich Aufmerksamkeit auszahlt.

Stakeholdermanagement scheitert, wenn man den Aufwand gleichmäßig auf alle verteilt. Die Erkenntnis des Gitters lautet: Einbindung muss maßgeschneidert sein – ein mächtiger, desinteressierter Manager braucht eine andere Behandlung als ein interessierter Endanwender ohne Macht. Wenn du deine Stakeholder einträgst, siehst du, wo du in die Unbeteiligten zu viel und in die Leute zu wenig investierst, die das Projekt kippen oder retten können.

Probier es aus: Trage die Stakeholder deines Projekts in ein Machtgitter ein und identifiziere einen Quadranten, den du vernachlässigst.

Die besten Zitate

- "ein zeitlich begrenztes Vorhaben, das durchgeführt wird, um ein einzigartiges Produkt, eine Dienstleistung oder ein Ergebnis zu schaffen" - Kathy Schwalbe (unter Verweis auf den PMBOK Guide)

- "die Anwendung von Wissen, Fähigkeiten, Werkzeugen und Techniken auf Projektaktivitäten, um die Projektanforderungen zu erfüllen" - Kathy Schwalbe (unter Verweis auf den PMBOK Guide)

- "PERT-gewichteter Durchschnitt = (optimistisch + 4 × wahrscheinlichster Wert + pessimistisch) / 6" - Kathy Schwalbe

Kritik

Das Buch stützt sich stark auf die Rahmenwerke des PMBOK Guide, was eher wie Taxonomie als wie Urteilsvermögen wirken kann – lange Listen von Inputs, Outputs und Prozessen, die eher beschreiben, was zu tun ist, als wie man unter Druck entscheidet. Seine Beispiele, aus IT-Kontexten gezogen, können neben dem, was Praktiker mit agilen Teams und wanderndem Scope erleben, generisch wirken. Die Standish-Group-Statistiken, die es zitiert, sind selbst dafür kritisiert worden, wie Erfolg und Misserfolg definiert wurden. Leser, die schlanke oder Startup-artige Ansätze suchen, werden merken, wie die traditionelle, plangetriebene Ausrichtung ihr Alter verrät.

In einem Satz

Projekte scheitern vorhersehbar – also steuere Scope, Zeit, Kosten, Menschen, Risiken und Stakeholder bewusst, mit Werkzeugen, die das Unsichtbare sichtbar machen.

Für wen sich das Buch eignet

IT-Profis, angehende Projektmanager und Studierende, die sich auf eine Zertifizierung vorbereiten und die PMBOK-Rahmenwerke in softwarespezifische Praxis übersetzt brauchen. Ebenso wertvoll für technische Führungskräfte, die immer wieder zusehen, wie Budgets und Termine davonlaufen, und die namentlich genannte Werkzeuge wollen, um das Bluten zu

stoppen.