

---

# Einleitung

## Motivation

Es ist schon frustrierend: 1972 schrieb Edsger W. Dijkstra zum ersten Mal über die »Softwarekrise«. 1996 erschien der erste Chaos Report der Standish Group und zeigte uns, wie wenige Softwareprojekte als erfolgreich gelten können. Heute, 30 Jahre später, gibt es unzählige Bücher über Softwareentwicklung, und dennoch laufen Softwareprojekte immer wieder aus dem Ruder. Die Software wird entweder deutlich teurer als geplant, nicht rechtzeitig fertig oder enthält bei Auslieferung noch inakzeptable Fehler (sprich: Bugs).

Warum ist das so? Haben wir denn seit 1972 nichts gelernt? Doch, haben wir. Inzwischen hat sich in der Industrie auch für die Softwareentwicklung der Prozessgedanke durchgesetzt. Dieser Gedanke stammt ursprünglich aus der Fertigung. Die Grundidee besteht darin, alle Arbeitsschritte klar zu definieren, sodass sie kontrolliert und reproduzierbar ablaufen. Auf diese Weise kann nachweislich eine einheitliche, idealerweise hohe Qualität produziert werden.

Anders als bei Hardware gibt es jedoch für Software keine Trennung zwischen Entwicklung und Produktion. Daher sind Prozesse in der Entwicklung gefragt. Software Engineering ist inzwischen ein Studienfach an Universitäten und Fachhochschulen.

Trotzdem kommt es immer wieder zu spektakulären Fehlschlägen bei Softwareprojekten. Ende März 2016 verlor die japanische Raumfahrtagentur JAXA<sup>1</sup> durch eine eindrucksvolle Verkettung von Hardware- und Softwarefehlern das 286 Millionen US-Dollar teure Röntgenteleskop Hitomi (japanisch: Auge, Pupille). Bei einer Neuausrichtung des Weltraumteleskops kam es zu einer falschen Bestimmung der Lagedaten und damit zur irrtümlichen Meldung, der Satellit rotiere. Um die nicht vorhandene Rotation zu stoppen, wurde das Reaktionsrad aktiviert. Das Teleskop geriet nun tatsächlich ins Trudeln, was

---

1. Japan Aerospace Exploration Agency.

wiederum das Kontrollsystem des Teleskops aktivierte. Unglücklicherweise war die betreffende Funktion vor dem Hochladen nicht ordentlich getestet worden. Statt Hitomi zu stabilisieren, wurden die Schubdüsen in die falsche Richtung aktiviert und somit die Rotation verstärkt, bis die Solarpaneele abbrachen. Damit fand die auf 10 Jahre ausgelegte Mission nach 3 Tagen ein abruptes Ende.

Dieses Beispiel ist natürlich besonders spektakulär, zeigt jedoch eine der typischen Schwierigkeiten, mit der eigentlich alle Softwareprojekte konfrontiert sind: Software ist in der Regel extrem komplex und das Zusammenspiel verschiedener Funktionen ist schwer zu überblicken. Dabei ist es an und für sich leicht, oft sogar viel zu leicht, Funktionen zu programmieren. Das verleitet dazu, »mal eben was zu ändern«. In der Regel sind es jedoch gerade diese Änderungen, die zu unerwarteten Wechselwirkungen mit bereits programmierten Funktionen führen.

Die Erkenntnis, dass Änderungen in unserer schnelllebigen Zeit eher die Regel als die Ausnahme sind, hat in der Softwareentwicklung zu einem Umdenken geführt. Seit 2000 haben sich zunehmend die sogenannten »agilen« Vorgehensmodelle durchgesetzt, bei denen die Software iterativ in möglichst kurzen Zyklen entwickelt wird. Die Kenntnis dieser Vorgehensmodelle und deren Einfluss auf das Projektmanagement gehört heute zum Werkzeugkasten eines modernen Softwareprojektmanagers.

## **Modernes Softwareprojektmanagement**

Womit wir beim Thema wären. Dieses Buch richtet sich an alle Mitarbeiter in Softwareprojekten, die direkt oder indirekt mit Projektmanagementaufgaben betraut sind. Dies können »hauptamtliche« Projektmanager mit Weisungsbefugnis sein, ebenso wie Softwareentwickler, die für die Zeit des Projektes ihren Kollegen als Primus inter Pares vorstehen. In agilen Vorgehensmodellen verschwimmen die Grenzen zum klassischen Projektmanager noch weiter, da das Projektteam teilweise Projektmanagementaufgaben übernimmt. Wenn in diesem Buch die Rede vom »Projektmanager« ist, meinen wir daher immer die Rolle und nicht zwangsläufig eine bestimmte Person.

Dieses Buch vermittelt Grundlagen, die jeder Projektmanager kennen und verstehen sollte, um erfolgreich zu sein. Der Inhalt des Buches entspricht dem Lehrplan des »ASQF<sup>®</sup> Certified Professional for Project Management« (kurz: CPPM). Dieser Lehrplan wurde 2016 vollständig überarbeitet, wobei zwei Aspekte im Vordergrund standen:

1. Modernisierung der Lehrinhalte und Angleichung an aktuelle Normen sowie
2. verstärkte Betonung der Soft Skills.

Da dieses Buch als Lehrbuch zum Kurs gedacht ist, beschreiben die zwei Punkte der Liste auch die Unterschiede zum Vorgängerbuch »Basiswissen Softwareprojektmanagement. Es werden systematisch sowohl die »klassisch« sequenzielle als auch die agile Vorgehensweise dargelegt. Wir gehen in jedem Kapitel darauf ein, wie die jeweils beschriebenen Tätigkeiten im einen wie im anderen Umfeld umgesetzt werden. Darüber hinaus richtet sich die Terminologie des Lehrplans (und damit auch dieses Buches) nach dem erstmalig 2012 erschienenen »Leitfaden zum Projektmanagement«, dem internationalen Standard ISO 21500.

Der verstärkte Fokus auf das Thema »Soft Skills« hängt eng mit der Zielgruppe des Kurses zusammen. Softwareprojektmanager sind oft eher technisch vorgebildet. Die fachliche Kenntnis ist jedoch nur die eine Seite der Medaille. Viel wichtiger und im Zweifelsfall auch entscheidender für den Projekterfolg ist jedoch die menschliche Komponente. Gerade Softwareentwickler mit lateraler Führungsverantwortung müssen sich mit dem Gedanken vertraut machen, dass ihre profunden Kenntnisse (z.B. der Programmiersprache) weniger ausschlaggebend für den Erfolg sind als ihre Soft Skills. Sie, die dafür ausgebildet wurden, Softwarearchitekturen und -algorithmen zu entwerfen, sollen plötzlich als Moderator, Verhandlungsführer, Schlichter und Motivator auftreten. Daher gehen wir in jedem Kapitel auch auf die jeweils erforderlichen »weichen Kompetenzen« ein, die für die jeweilige Rolle erforderlich sind.

## Der ASQF® CPPM

Die erste Version des Kurses »Certified Professional for Project Management« (CPPM) wird seit 2004 vom International Software Quality Institute (kurz: iSQI) angeboten. Zuvor hatte sich eine Arbeitsgruppe des Arbeitskreises Software-Qualität und Fortbildung e.V. (ASQF®) gebildet, um einen schlanken, speziell auf Softwareprojektmanagement ausgerichteten Lehrplan zu verfassen. Auch die aktuelle Überarbeitung ist auf eine ASQF-Arbeitsgruppe zurückzuführen, an der die Autoren dieses Buches beteiligt waren. Der aktuelle Lehrplan ist hier zu finden: <https://www.isqi.org/de/project-management.html>.

Im Gegensatz zu den bekannten Projektmanagementschulungen der International Project Management Association (IPMA)<sup>2</sup> und des amerikanischen Project Management Institute (PMI) fokussiert der ASQF<sup>®</sup> CPPM ausschließlich auf das für Softwareprojekte erforderliche Grundwissen. Weiterführende Themen wie Multiprojektmanagement werden explizit ausgespart, um den Kurs nicht zu überfrachten. Auch dieses Buch folgt der gleichen Logik, liefert jedoch den einen oder anderen Hinweis über den Tellerrand hinaus.

Wenn von Projektmanagementkursen die Rede ist, sollte auch ein Hinweis auf PRINCE2 nicht fehlen. PRINCE steht für »Projects in Controlled Environments«. Dahinter verbirgt sich eine Projektmanagementmethode, die sich bis heute großer Beliebtheit erfreut. Inzwischen gibt es auch »PRINCE2 Agile«, ein Ergänzungsmodul, das das eher auf sequenziellen Vorgehensmodellen aufsetzende PRINCE2-Handbuch um die wichtigsten agilen Prinzipien erweitert.

Der ASQF<sup>®</sup> CPPM steht nicht im Widerspruch zu den hier genannten Kursen, hat allerdings auch nicht den gleichen Anspruch auf Vollständigkeit. Noch einmal: Es geht hier ausschließlich um Softwareprojekte unter meist lateraler Leitung.

## Ein paar Worte zum Buch

Dieses Buch gliedert sich in 12 Kapitel und folgt damit nahezu 1 zu 1 der Gliederung des ASQF<sup>®</sup>-CPPM-Lehrplans.

Kapitel 1 bietet einen Überblick über das Thema Softwareprojektmanagement und enthält Definitionen der wichtigen Begriffe. Das Kapitel orientiert sich eng am Standard ISO 21500 mit dem Ziel, eine einheitliche Terminologie zu schaffen, die auch im Buch durchgängig verwendet wird.

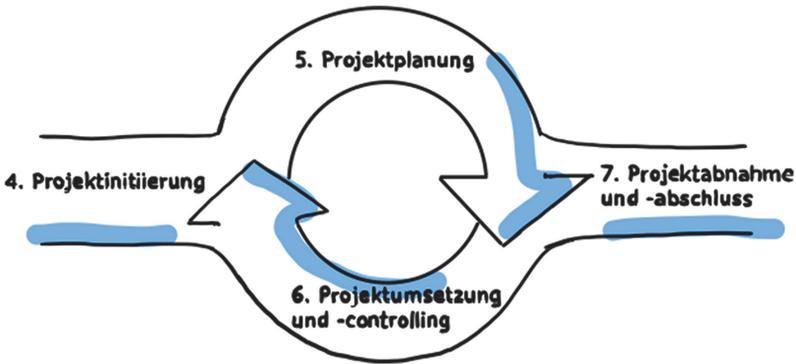
Kapitel 2 behandelt mögliche Projektorganisationsformen, wobei zwischen Aufbau- und Ablauforganisation unterschieden wird.

In Kapitel 3 werden verschiedene Prozess- und Vorgehensmodelle beschrieben. Wir unterscheiden prinzipiell zwischen sequenziellen und agilen Vorgehensmodellen mit ihren exemplarischen Vertretern »V-Modell« und »Scrum«.

Kapitel 4 bis 7 beschreiben den Ablauf eines Projektes, beginnend mit der Projektinitiierung (Kap. 4) über die Projektplanung (Kap. 5), Projektumsetzung und -controlling (Kap. 6) bis hin zu Projektannahme und -abschluss (Kap. 7).

---

2. Das IPMA wird in Deutschland durch die Deutsche Gesellschaft für Projektmanagement (GPM) vertreten.



**Abb. 0-1**  
Zusammenspiel der  
Kapitel 4 bis 7

Abbildung 0-1 zeigt das Zusammenspiel dieser vier Kapitel. Planung bzw. Umsetzung und Controlling werden während eines Projektes meist mehrfach durchlaufen, während die Initiierung naturgemäß nur einmal zu Beginn und Abnahme und Abschluss einmal am Ende des Projektes stattfinden.

Kapitel 8 widmet sich dem Qualitätsmanagement. Dabei konzentrieren wir uns auf die Bedeutung der Qualitätssicherung in Projekten. Schließlich ist dies kein Lehrbuch für Softwaretester. Etwas ausführlicher wird die Frage der Qualitätssicherung für Prozesse behandelt, da diese den Projektmanager zumindest indirekt betreffen.

Kapitel 9 handelt vom Risikomanagement. Auch hier geht es zunächst einmal um den Grundgedanken des Risikomanagementprozesses, bevor die einzelnen Aktivitäten der Risikoermittlung, -bewertung, -beherrschung und des Risikocontrollings beschrieben werden. Ein kleiner Exkurs behandelt das Risikomanagement in sicherheitskritischen Bereichen.

Kapitel 10 ist ein besonders wichtiges Kapitel dieses Buches, denn hier geht es um Personalmanagement. Neben Phasen und Aktivitäten des Personalmanagements werden Themen wie Teamentwicklung, soziale Kompetenz und Mitarbeitermotivation adressiert. Der Leser bekommt einen Einblick in die Bedeutung der menschlichen Aspekte im Projektmanagement, die nur allzu gerne unterschätzt wird.

Kapitel 11 liefert einen Überblick über Reifegradmodelle und wie diese verwendet werden können, um Prozesse einerseits zu bewerten und andererseits zu verbessern.

Kapitel 12 ist das einzige Kapitel, das kein direktes Äquivalent im CPPM-Lehrplan hat. Es enthält eine kurze Zusammenfassung des Buches.

Das Thema »Soft Skills« zieht sich wie ein roter Faden durch das ganze Buch. Wann immer angebracht, gehen wir darauf ein, welche nicht technischen Kompetenzen ein Projektmanager benötigt, um die jeweils beschriebenen Aufgaben erfolgreich zu meistern.

## Das Fallbeispiel

Wie Johann Wolfgang von Goethe so schön sagte: »Grau, teurer Freund, ist alle Theorie (...)« Das soll nicht sein! Deshalb haben wir in diesem Buch ein durchgängiges Fallbeispiel verwendet, das realistischer nicht sein könnte, da es auf tatsächlichen Erfahrungen der Autoren basiert.

Bei unserem Fallbeispiel handelt es sich um ein Inhouse-Projekt zur Entwicklung einer Plattform. Es gab also keinen externen Kunden. Die zu entwickelnde Grafikplattform war dazu bestimmt, von Applikations- und Kundenprojekten verwendet zu werden. Konkret sprechen wir von Grafikfunktionen, die beispielsweise dazu dienen, Instrumententafeln für Maschinen oder Fahrzeuge zu realisieren.

Insgesamt gab es zwei Projektteams. Sieben Mitarbeiter entwickelten die Grafik-Engine, fünf weitere die Grafikbibliothek. Ihnen vorgesetzt war ein Projektmanager, der gleichzeitig auch organisatorisch der Teamleiter war. Dieser Projektmanager bündelte alle Informationen und war als einziger Stakeholder den Teams bekannt.

Wie es in der Realität so ist, sah sich das Projekt mit einer Reihe von Herausforderungen konfrontiert, die durch gezieltes Projektmanagement hätten vermieden werden können. Beispielsweise wurde das Projekt bis zu einem gewissen Punkt *ohne* dokumentierte Anforderungen betrieben. Der Projektmanager hatte alle Informationen im Kopf und steuerte die Mitarbeiter einzeln aus. Generell ließ die Kommunikation im Projekt zu wünschen übrig. Es gab keine erkennbare Abstimmung mit abhängigen Projekten. Die Teammitglieder wussten wenig bis nichts von der Arbeit ihrer Kollegen, wodurch das Projekt für sie extrem unattraktiv wurde. Erschwerend kam hinzu, dass es so gut wie keine Projektmanagementaktivitäten hinsichtlich Planung, Controlling, Qualitätssicherung und Risikomanagement gab.

Wer schon in Softwareprojekten gearbeitet hat, wird sich das Setup vorstellen können. Wir nutzen dieses durchgängige Fallbeispiel im Buch, um zu zeigen, welche Projektmanagementaktivität an welcher Stelle hätte helfen können.