Erfolgs- und Misserfolgsfaktoren der Anwendungsentwicklung – eine aktuelle empirische Überprüfung und Einordnung

Dipl.-Inform. Heiner Merz, Dipl.-Kfm. Michael Stewen

Lehrstuhl Wirtschaftsinformatik II, Abteilung VIII des
Betriebswirtschaftlichen Instituts der Universität Stuttgart
Keplerstr. 17
70174 Stuttgart
heiner.merz@bwi.uni-stuttgart.de

Abstract: Dieser Konferenzbeitrag stellt Erfolgsfaktoren und Misserfolgsfaktoren der Anwendungsentwicklung dar. Grundlage hierfür bilden empirische Untersuchungen des Jahres 2009. Diese Erfolgs- und Misserfolgsfaktoren werden anhand der Klassifikation der ISO/IEC 12207 und ISO/IEC 15288 gruppiert und diskutiert.

1 Hintergrund und Basis des vorliegenden Konferenzbeitrags

Dieses Paper basiert auf empirischen Untersuchungen, welche im Sommer 2009 vom Lehrstuhl Wirtschaftsinformatik II der Universität Stuttgart durchgeführt wurden. Vorbereitend der Primärdatenerhebung in Form von Experteninterviews erfolgten Sekundärdatenanalysen, die zur Aufstellung von Erfolgs- und Misserfolgsfaktoren[1] in Softwareentwicklungsprojekten führten. Deren aktuelle empirische Prüfung und Bestätigung erfolgte sowohl in den Experteninterviews [St09], als auch als Zusatzergebnis einer am Lehrstuhl derzeit laufenden Dissertation.

Ziel der empirischen Bestätigung ist die Prüfung der Aktualität und der praxeologischen Relevanz der aus den Sekundärdatenanalysen erstellten Erfolgs- und Misserfolgsfaktoren.
Zur Forschungsmethodik, Auflistung der Interviewpartner, Dokumentation und Auswertung der Interviews sei auf [St09] verwiesen.

2 Terminologie

Manche Begriffe finden in verschiedenen Literaturquellen mehrdeutig Verwendung. Nachstehend erfolgt daher eine kurze Begriffsfestlegung für den vorliegenden Beitrag.

[1] bzgl. Terminologie vgl. 2.2.2 Erfolgsgefährdende Faktoren: ‘Misserfolgsfaktoren‘

2.1 Externe Vergabe

‘Externe Vergabe‘ steht im Folgenden für jegliche Art der Vergabe von Auftragsentwicklung an wirtschaftlich oder rechtlich unabhängige Partner oder an Partner in verschiedenen Rechtsgebieten/Hoheitsbereichen, z.B. auch Outsourcing oder Offshoring.

2.2 Erfolgsfaktoren / Erfolgsgefährdende Faktoren: ‘Misserfolgsfaktoren‘

Ein Faktor[2] kann fördernd, hemmend oder gefährdend in Erscheinung treten. Ein Faktor kann ein Ereignis oder auch ein materielles oder immaterielles Mittel oder eine Leistung sein, auch die Eigenschaften eines Objektes können im Softwareentwicklungsprozess einen Faktor darstellen; selbst die Prozessgestaltung kann als Faktor betrachtet werden (vgl. [St09], S. 30).

Hierbei sind erfolgsgefährdende Faktoren, d.h. ‘Misserfolgsfaktoren‘, keine Umkehrung bzw. Negierung von Erfolgsfaktoren (bzw. umgekehrt). Beide Arten von Faktoren sind in einer Anwendungsentwicklung gesondert zu beachten. Die ideale Anwendungsentwicklung ist bestrebt, alle für sie zutreffenden Erfolgsfaktoren herbeizuführen und / oder diese zu etablieren und sie ist bestrebt permanent zu beachten, dass kein Misserfolgsfaktor auftreten kann. Sie ist weiterhin bestrebt, bei Auftreten eines Misserfolgsfaktors unverzüglich Gegenmassnahmen einzuleiten.

2.2.1 Erfolgsfaktoren

Erfolgsfaktoren können das Erreichen der Ziele einer Anwendungsentwicklung fördern, bewirken können sie einen Erfolg jedoch nicht. Es ist eher die umgekehrte Betrachtung zutreffend: Wenn ein Projekt bzw. eine Anwendungsentwicklung auf die in Kap. 4 gelisteten Erfolgsfaktoren hin geprüft wird und diese fehlen, fehlerhaft sind, nicht zutreffen, nicht gültig oder nicht möglich sind, dann wird ein Erfolg unwahrscheinlich.

Kritische Erfolgsfaktoren stellen hingegen notwendige (jedoch auch nicht hinreichende) Voraussetzungen für das Erreichen der Ziele einer Anwendungsentwicklung dar. Anders ausgedrückt: Ein nicht erfüllter, kritischer Erfolgsfaktor kann zum Scheitern eines Softwareentwicklungsprojektes führen (vgl. [CS10]).

Ein Beispiel für einen Erfolgsfaktor in der Anwendungsentwicklung ist, dass eine Anwendungsentwicklung auf einem Vorgehensmodell basiert, welches eine jeweilige Reaktion auf ein Auftreten eines Misserfolgsfaktors klar definiert – z.B. unmittelbare Eskalation an einen Lenkungsausschuss/Steuerkreis o.ä.

[2] „massgebende Wirkungskraft“, vgl. [BI09]

2.2.2 Erfolgsgefährdende Faktoren: ‘Misserfolgsfaktoren‘

Erfolgsgefährdende Faktoren, d.h. Misserfolgsfaktoren können das Erreichen der Ziele einer Anwendungsentwicklung gefährden, schwieriger oder unmöglich machen. Das Auftreten eines oder mehrerer Misserfolgsfaktoren ist somit kritisch für den unmittelbaren Projektzustand (und somit für einen zukünftigen Projekterfolg) zu sehen.

3 Kategorisierung entsprechend ISO/IEC 12207 und ISO/IEC 15288

Angesichts des Sachverhalts, dass Unternehmen in der Praxis zur Strukturierung, Bewertung und Optimierung ihrer Anwendungsentwicklung oder zur Beurteilung der entsprechenden Qualität ihrer externen Dienstleister auf bewährte Reifegradmodelle, wie z.B. CMMI (vgl. [CM10]) oder SPICE (vgl. [15504]) zurückgreifen, orientiert sich auch die Faktorenanalyse strukturell an einem etablierten Prozessrahmenwerk (vgl. [Hi06], S. 212f. und 218).

Die Kategorisierung der im nachfolgenden Kapitel gelisteten Erfolgs- und Misserfolgsfaktoren in der Anwendungsentwicklung wird daher in Anlehnung an die in nachfolgend abgebildeten Schema der Struktur des Prozessrahmenwerks ISO/IEC 12207 gegliedert.

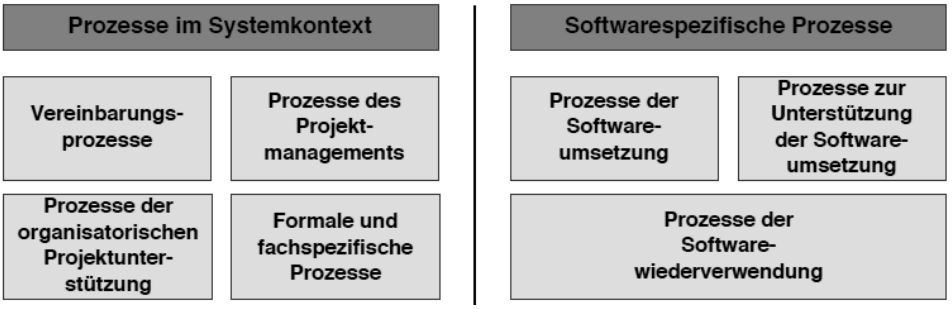

Abbildung 1: Gliederungsschema des Prozessrahmenwerks der ISO / IEC 12207

3.1 ISO/IEC 12207 – Entwicklung, Betrieb und Wartung von Softwaresystemen

Die Norm ISO / IEC 12207 gliedert sich in obig dargestellte sieben Prozessgruppen. Wichtig ist hierbei, dass es sich nicht um Pflichtprozesse handelt, die zu implementieren sind. Vielmehr gilt es bei der Nutzung des Referenzmodells die Prozesse entsprechend den Bedürfnissen der Organisation und in Vereinbarung mit dem Anwendungsbereich auszuwählen und anzupassen.
In ISO / IEC 12207 werden die Bereiche Prozesse im Systemkontext und softwarespezifische Prozesse unterschieden. Die Prozesse im Systemkontext beschäftigen sich mit eigenständigen Softwareprodukten oder Softewaredienstleistungen oder ganzen Softwaresystemen. Die softwarespezifischen Prozesse erweitern den Anwendungsbereich um die Implementierung von Softwareprodukten oder Softwaredienstleistungen in ein grösseres Gesamtsystem (vgl. [12207], S. 14).

3.2 ISO/IEC 15288 – Systementwicklung / Systemlebenszyklus und seine Prozesse

In enger Verbindung zur ISO/IEC 12207 steht die Norm ISO/IEC 15288 mit Fokus auf Prozesse des Systemlebenszyklusses. Beide Normen können parallel eingesetzt werden. Grundsätzlicher Schwerpunkt von ISO/IEC 15288 ist die Bereitstellung eines Rahmens für die Beschreibung des Lebenszyklusses eines künstlich geschaffenen Systems.
Im Verständnis der Norm kann ein System als zentrales Bezugsobjekt sowohl aus Hardware- als auch aus Softwarekomponenten sowie Daten, Menschen, Prozessen, Prozeduren, Anlagen, Materialien und weiteren natürlichen Objekten bestehen. Die ISO/IEC 15288 zielt auf die Bereitstellung eines definierten Satzes an Prozessen für die Kommunikation zwischen Einkäufern, Lieferanten und weiteren Interessengruppen ab (vgl. [15288], S. 1).

Da die beiden Normen ISO/IEC 12207 und ISO/IEC 15288 weitestgehend harmonisiert sind, gleicht die strukturelle Gliederung der Norm ISO / IEC 15288 grossteils den Prozessgruppen der Systemkontextprozesse von ISO/IEC 12207.
Die Kategorisierung nachfolgend gelisteter Erfolgs- und Misserfolgsfaktoren nach ISO/IEC 12207 ist somit übertragbar auf ISO / IEC 15288 und beinhaltet die Anwendungsentwicklung, sowohl aus Softwaresicht als auch aus Systemsicht.
Eine Produktentwicklung kann entsprechend erfolgen.

4 Erfolgs- und Misserfolgsfaktoren der Anwendungsentwicklung

Die prozessorientierte Klassifizierung der Erfolgs- und Misserfolgsfaktoren entspricht i.d.R. den realen Rahmenbedingungen eines Softwareentwicklungsprojektes bzw. einer Anwendungsentwicklung.
So ist z.B. das V-Modell XT "… kompatibel mit aktuellen (Quasi-)Standards … wie zum Beispiel ISO/IEC 15288." ([VM10], S. 1-10)
Eine Übertragung und Anwendung nachfolgend gelisteter Erfolgs- und Misserfolgsfaktoren auf reale Tätigkeiten der Anwendungsentwicklung ist somit durch deren Kategorisierung entsprechend ISO/IEC 12207 und ISO/IEC 15288 unterstützt.

4.1 Erfolgs- und Misserfolgsfaktoren der Prozesse im Systemkontext

4.1.1 Vereinbarungsprozesse

In den Vereinbarungsprozessen erfolgen die Abstimmungen und Vereinbarungen zwischen Anforderer und Leistungserbringer. Bei externer Vergabe sind Ergebnisse der Vereinbarungsprozesse das Pflichtenheft sowie sämtliche abzuschliessende Verträge.[3]

[3] i.d.R. bestehen diese aus dem konkreten Werk-, Dienst- oder Kaufvertrag plus einen Rahmenvertrag, welcher generelle Fragen der Zusammenarbeit (z.B. Haftung, Vertraulichkeitsvereinbarungen, Gerichtsstand usw.) regelt. Bei ersten Aufträgen, unregelmäßiger Zusammenarbeit usw. wird auf den Rahmenvertrag jedoch oft verzichtet so dass dessen Inhalt in den Einzelverträgen berücksichtigt werden muss.

Generelle Erfolgsfaktoren in Vereinbarungsprozessen[4]

- Vereinbarung realistischer Ziele / Definition klarer Projektziele

Übereinstimmend erachten, sowohl die untersuchten Literaturquellen, als auch die Interviewpartner der Experteninterviews, die Vereinbarung realistischer Ziele und Definition klarer Projektziele als den für alle Arten der Anwendungsentwicklung allgemein gültigen Erfolgsfaktor. Dieses eigentlich selbsterklärende Ergebnis bedingt jedoch, dass ein fundiertes Requirements-Engineering der Spezifikation vorausgeht sowie dass ein etabliertes, ebenfalls fundiertes Change-Management von während der Projektlaufzeit / Anwendungsentwicklung oft kommenden Änderungs- oder Anpassungsanforderungen stattfindet.

Vereinbarungsprozesse bei externer Vergabe haben zusätzlich folgende Erfolgsfaktoren

- Umfassende Branchenkenntnisse des Auftragnehmers
- Finanzielle Stabilität des Anbieters
- Angemessene politische und rechtliche Stabilität im Land des Auftragnehmers
- Partnerschaftliche Beziehung zwischen Auftraggeber und Auftragnehmer
- Passende Unternehmensgrösse des Auftragnehmers zum Auftraggeber

Während die ersten vier Erfolgsfaktoren selbsterklärend sind, erklärt sich der Faktor "Passende Unternehmensgrösse" durch geforderte / vorhandene Zertifizierungen, angewendete Vorgehensmodelle und die oft ähnlichen, d.h. dann oft "automatisch passenden" organisatorischen Abläufe und Hierarchien in Organisationen vergleichbarer bzw. 'passender' Grösse.

Misserfolgsfaktoren in Vereinbarungsprozessen[5]

Generelle (d.h. für sowohl intern durchgeführt, als auch extern vergeben, als auch Mischformen) Misserfolgsfaktoren konnten nicht festgestellt werden.

Misserfolgsfaktoren in Vereinbarungsprozessen bei externer Vergabe sind jedoch

- Ungeklärte Eigentumsrechte
- Unklare Verträge
- Festpreisprojekte ohne diesbezügliche Erfahrung
- Vertrauensmangel

[4] Quellen: vgl. [AW05], [Be08], [Jo96], [Ko02], [SG01]
[5] Quellen: vgl. [BJS00], [Jo96], [Ko02]

Der letzte Punkt, "Vertrauensmangel", kann als Umkehrung o.g. Erfolgsfaktors "partnerschaftliche Beziehung" gesehen werden; die ersten zwei Punkte betonen, dass aus nicht besprochenem / vereinbartem (und somit nicht geklärtem) Unklarheiten entstehen, die zu Missverständnissen führen was in Vertragsverhältnissen zur jeweils für sich selbst vorteilhafteren Sicht jedes Partners führt. Streitigkeiten oder gerichtliche Auseinandersetzungen sind hier somit vorprogrammiert. Auftragnehmer, welche Festpreisprojekte akzeptieren ohne diesbezügliche Erfahrung zu haben, tragen evtl. entstehende Mehrkosten (womit o.g. Erfolgsfaktor "finanzielle Stabilität des Anbieters" u.U. gefährdet ist).

4.1.2 Prozesse der organisatorischen Projektunterstützung

Prozesse der organisatorischen Projektunterstützung dienen der Schaffung und Erhaltung der Leistungsfähigkeit einer Organisation. Hierzu gehören z.B. die Bereitstellung notwendiger Infrastruktur und Ressourcen (vgl. [12207], S. 15), das Prozess- und Qualitätsmanagement sowie ein Projektportfoliomanagement (vgl. [12207, S. 25ff.).

Generelle Erfolgsfaktoren in Prozessen der organisat. Projektunterstützung[6]

- Definition von Projektstandards, Einrichtung einer effizienten IT-Infrastruktur
- Aufgabengrösse beherrschbar halten, evtl. Aufteilung in Teilziele
- Frühzeitiges Einleiten eines notwendigen Change-Management Prozesses
- Anhaltende Managementunterstützung
- Geeignetes Projektteam zusammenstellen, einschlägige Erfahrungen der Projektleiter

Während die ersten drei Erfolgsfaktoren konstruktiver Natur sind, d.h. diese Prozesse bzw. Standards werden in der Projekt- und Produktentwicklung entsprechend etabliert, ist der vierte Faktor, "anhaltende Managementunterstützung", eine Führungsfrage psychologischer Art – doch ist er im Falle auftretender Probleme entscheidend wichtig. Der letzte Punkt ist ein Faktor der Kategorie 'Soft Skills'[7], d.h. er betrifft die Eigenschaften, Qualifikationen und das Zusammenarbeiten der Mitarbeiter.

[6] Quellen: vgl. [AW05], [Ke98], [Ko02], [SG01]

[7] „Soft Skills stehen für einen ganzen Katalog von überfachlichen Kompetenzen, die dem persönlichen und zwischenmenschlichen Bereich zuzuordnen sind. Das Spektrum der Soft Skills reicht beispielsweise von Menschenkenntnis und Einfühlungsvermögen sowie Kommunikations- und Kritikfähigkeit bis hin zum Durchsetzungsvermögen und der Fähigkeit, andere Menschen einzubinden und für ein gemeinsames Ziel zu begeistern." [BG10]

Bei externen Vergaben der Anwendungsentwicklung kommen hier noch folgende Erfolgsfaktoren hinzu:

- Angemessenes technisches Verständnis auf Auftraggeberseite
- Hohe Qualifikation der Mitarbeiter auf Auftragnehmerseite
- Sprachkenntnisse, kulturelle Sensibilität und internationale Unternehmenskultur

Während der erste Faktor technisches Know-How auch auf Auftraggeberseite fordert, was bedingt, dass der Auftraggeber fundierter über Probleme oder Fortschritte im Projekt befinden kann und nicht dem Auftragnehmer und 'nur' dessen Darstellungen Glauben schenken muss, beinhaltet der zweite Faktor eine Know-How Forderung auf Auftragnehmerseite – nämlich Fachwissen über die zu entwickelnde Anwendung. Ein gegenseitiges Verstehen des jeweilig anderen Fachwissens ist somit erleichtert. Der letzte Punkt ist wieder eindeutig in der Kategorie 'Soft Skills' (s.o.)

Misserfolgsfaktoren in Prozessen der organisatorischen Projektunterstützung[8]

- Kein handlungsfähiger Lenkungsausschuss
- Keine Kontrolle des Projektes / keine Benutzung automatischer Planungswerkzeuge
- Nicht genügend Ressourcen bzw. unrealistisches Budget
- Projektleitung mit zu wenig operativem Entwicklungs-Know-How
- Personelle Defizite, ungenügendes Wissen oder Erfahrungen bei Mitarbeitern
- Konflikte zwischen Abteilungen

Dass aufgrund des ersten Punktes ein Projekt scheitern kann, erstaunt evtl. etwas – doch führt ein handlungsunfähiger Lenkungsausschuss zu Akkumulation von Problemen. Probleme, welche evtl. einzeln zwar durch Workarounds[9] lösbar sind, in der Summe jedoch ein Projekt zu Fall bringen können. Die restlichen Faktoren sind selbsterklärend, wobei der Teilaspekt "Personelle Defizite" wieder in die Kategorie 'Soft Skills' einzuordnen ist.

[8] Quellen: vgl. [AW05], [Bo91], [BJS00], [Jo96], [Ke98], [Ko02]
[9] hier i.S.v. Umgehung, Ersatzlösung, anderweitige Kompensation

4.1.3 Prozesse des Projektmanagements

Die Prozessgruppe des Projektmanagements umfasst in ISO / IEC 12207 sieben grundlegende Prozesse. So werden beispielsweise in einem Projektplanungsprozess die allgemeinen Projektanforderungen und die Realisierbarkeit eines Projekts festgelegt. Weiterhin werden Vorgehensweisen zur Projektplanung definiert, es wird ein Projektbewertungs- und Projektsteuerungsprozess abgegrenzt, ein Prozess zur Entscheidungsunterstützung, der Risikomanagementprozess, das Konfigurationsmanagement, das Informationsmanagements und ein Messungs- und Bewertungsprozess sind hier ebenfalls mit beinhaltet (vgl. [12207], S. 32ff.).

Generelle Erfolgsfaktoren in Prozessen des Projektmanagements[10]

- Effiziente Organisationsstruktur, schnelle Entscheidungsfindung
- Früher Einsatz von Werkzeugen für Abschätzungen / frühzeitige und regelmässige Kontrolle der Projekt(teil)ergebnisse / verlässliche Schätzungen
- Formales Berichtswesen / standardisierte, dokumentierte Prozesse / stringente Projektmanagement-Methoden
- Permanentes Risikomanagement während der gesamten Projektlaufzeit

Während die ersten beiden Punkte offensichtlich sind, etabliert der dritte ein Mindestmass an Bürokratie. Ein formales Berichtswesen sorgt jedoch dafür, dass Information regelmässig und an die richtigen Stellen fliesst. Und so kann z.B. ‘versehentlich‘ nichts verschwiegen werden, das Management muss permanent involviert bleiben. Der Punkt “permanentes Risikomanagement“ begründet sich vor allem durch die, während der Projektlaufzeit i.d.R. unweigerlich kommenden Änderungen. Das Risikomanagement sollte Teil des Change-Managements sein oder zumindest von diesem bei jedem Change automatisch angestossen werden.

Hinzukommende Erfolgsfaktoren bei externen Vergaben

- Geografische Nähe des Anbieters
- Umfassende Erfahrung mit IT-Outsourcingprojekten

Während der zweite Punkt offensichtlich und einleuchtend ist, ist der erste Erfolgsfaktor, die geografische Nähe des Anbieters, zu Zeiten des Internets, Telekonferenzen usw. zunächst verblüffend. Doch kulturelle und mentale Unterschiede wachsen oft mit der geografischen Entfernung und die Auswirkung von Arbeitszeiten in unterschiedlichen Zeitzonen (mit der für Echtzeitkommunikation dann oft nur kleinen möglichen Kommunikationszeit-Schnittmenge) wird nach wie vor stark unterschätzt.

[10] Quellen: vgl. [AW05], [Be08], [Bo91], [BJS00], [Fe04], [Jo96], [Ko02]

Misserfolgsfaktoren in Prozessen des Projektmanagements[11]

- Andauernde Änderungen von Anforderungen, ineffektives Change-Management
- Fehlendes Risikomanagement / keine Kontrolle der Kosten, Kostenüberschreitungen
- Unrealistische Aufwandsabschätzungen / Planungen (Zeit, Kosten, Ressourcen usw.)
- Kein angemessenes Konfigurationsmanagement
- Zu grosser Projektumfang

Ein fundiertes und etabliertes Change-Management wurde in allen, die Sekundärdatenanalyse bestätigenden Interviews (vgl. [St09], S. 63ff.) als einer der wichtigsten Erfolgsfaktoren, sowohl in der Entwicklung, als auch im Betrieb von Anwendungen genannt. Andauernde Änderungen von Anforderungen zeugen entweder von einem unzureichenden Requirements-Engineering oder stellen zumindest den derzeitigen Zweck und das Ziel der Anwendung überhaupt in Frage. Zudem führt eine permanente ad-hoc Berücksichtigung von Änderungen, neben dem Aufwand (Chaos?) auch zur Frustration der Entwickler. Kostenüberschreitungen, welche i. d. R. aus fehlendem oder mangelhaftem Risikomanagement, aus zu vielen Änderungen oder aus unrealistischen Aufwandsabschätzungen entstehen, führen zwar nur selten zum gänzlichen Misserfolg, sie bewirken jedoch oft das Fertigstellen einer Version ohne 'Features' / Begeisterungsmerkmale.

Misserfolgsfaktoren speziell bei externer Vergabe von Anwendungsentwicklungen

- Verspätung extern entwickelter Komponenten bzw. extern durchgeführter Aufgaben
- Konflikte zwischen internen und externen Mitarbeitern

Externe Vergabe erzeugt Abhängigkeiten – und was nützt es z.B. später vor Gericht Recht zu bekommen, wenn ein komplettes Grossprojekt aufgrund eines kleinen, extern vergebenen Teilprojektes scheiterte? Der zweite Punkt benennt die bereits erwähnten 'Soft Skills'.

4.1.4 Formale (technische) und fachspezifische Prozesse

Diese Prozessgruppe umfasst den Prozess zur Definition der Anforderungen aller Anspruchsgruppen an das Softwaresystem, die Ableitung einer Systemspezifikation aus den Anforderungen sowie die Überprüfung derselben auf ihre Realisierbarkeit.

[11] Quellen: vgl. [Bo91], [BJS00], [FLS02], [Jo96], [Ke98], [Ko02]

Weiterhin umfasst sie den Prozess des Entwurfs und der Überprüfung einer Systemarchitektur auf Basis der Spezifikationen. Der Realisierungsprozess ist hier ebenso beschrieben wie der Systemintegrationsprozess, welcher einzelne Systemelemente zu einem Gesamtsystem zusammenführt. Ergänzt werden diese Prozesse durch Systemeignungstests und die Installation des geprüften Systems (vgl. [12207], S. 42ff.).

Generelle Erfolgsfaktoren in formalen (technischen) und fachspez. Prozessen[12]

- Detaillierte Spezifikation, Erstellen eines umfassenden Business-Case
- Einbindung von Benutzern / Akzeptanz bei Benutzern
- Stabile Basisanforderungen, die den 'minimalen' Funktionsumfang ergeben
- Abstimmung der Implementierungsmethode auf die Organisation
- Verwendung von Standardelementen

Ebenso wichtig wie eine umfassende, detaillierte Spezifikation ist die fachliche Beschreibung der Funktionalität der späteren Anwendung. Hierdurch wird vermieden, dass Entwickler den Kontext der Funktionalität missverstehen und dass die Entwickler somit, trotz formal korrekter einzelner Umsetzungen, das ganze Produkt am Bedarf vorbei entwickeln. Die Einbindung von Benutzern in einen Anwendungsentwicklungsprozess bewirkt i.d.R. dasselbe. Zum dritten Punkt: Bei der Erhebung der Anforderungen an die Software sollte eine Konzentration auf stabile Basisanforderungen erfolgen, d.h. zunächst nur wenige, jedoch grundlegenden Anforderungen, die schnell umsetzbar sind. Dies präsentiert Nutzern oder Kunden schon früh ein verifizierbares Ergebnis. Der vierte Punkt ist offensichtlich – eine Verwendung von Standardelementen erspart Entwicklungsaufwand sowie fast sämtliche sonstige Tests (ausser Integrationstests) und weitere analytische Qualitätssicherungsmassnahmen einer Neuentwicklung.

Misserfolgsfaktoren in formalen (technischen) und fachspezifischen Prozessen[13]

- Andauernde Änderung der Anforderungen / keine Kontrolle der Anforderungen
- Anforderungen fehlen oder wurden nicht bzw. falsch verstanden
- Erwartungen der Anwender nicht beachtet / erfüllt, fehlende Benutzerakzeptanz
- Zu hohe Komplexität der Software
- Inkompatibel zu bestehenden Anwendungen / Nichtbeachtung von Industriestandards
- Unklare oder unvollständige Spezifikationen, Fehlen eines Pflichtenhefts
- Fehler in der Implementierung

[12] Quellen: vgl. [AW05], [Fe04], [Jo96], [Ke98], [Ko02], [SG01]
[13] Quellen: vgl. [Bo91], [BJS00], [FLS02], [Jo96], [Ke98], [Ko02]

Während der erste Punkt wieder die Bedeutung eines ‘gelebten‘ Change-Managements unterstreicht, belegen die weiteren fünf Punkte die fundamentale Bedeutung eines fundierten und etablierten Requirements-Engineerings. Lediglich der letzte Punkt dieser Liste, welche formale (technische) Prozesse betrifft (sowie teilweise der Punkt “Zu hohe Komplexität der Software“) beinhaltet tatsächlich Probleme der technischen Umsetzung.

4.2 Softwarespezifische Prozesse

Die nun nachfolgenden Erfolgs- und Misserfolgsfaktoren sind in der Anwendungsentwicklung aus der Entwicklung von Software hergeleitet. Ob eine Übertragbarkeit auf weitere Bereiche, wie z.B. eine Produktentwicklung stattfinden kann, muss durch jeweilige Betrachtung der durchzuführenden Entwicklung festgestellt werden. Eine solche Übertragbarkeit kann in vielen Fällen jedoch durchaus gegeben sein, zudem stellt die Softwareentwicklung einen entscheidenden Anteil aller Entwicklungstätigkeiten dar.

4.2.1 Prozesse der Softwareumsetzung

Die Prozessgruppe zur Softwareumsetzung besteht aus sieben Prozessdefinitionen, die die Herstellung spezifischer Systemelemente im Sinne von Softwarekomponenten und deren Implementierung in das gesamte Softwaresystem beschreiben (vgl. [12207], S. 57). Die Prozessempfehlungen dieses Bereichs weisen starke Analogien zu Teilprozessen aus der Gruppe der formalen (technischen) und fachspezifischen Prozesse auf und deckt sich in Bezug auf relevante Erfolgs- und Misserfolgsfaktoren mit denen jener Prozessgruppe (vgl. [St09], S. 52).

4.2.2 Prozesse zur Unterstützung der Softwareumsetzung

Generelle Erfolgsfaktoren der Prozesse zur Unterstützung der Softwareumsetzung[14]

Die betrachteten Literaturquellen benennen als Erfolgsfaktoren zur Unterstützung der Softwareumsetzung analytische Massnahmen (vgl. [SBM83], S. 38ff.). Insbesondere werden hier genannt

- Code-Inspektionen, Design-Reviews und formale Testmethoden

Analytische (Qualitätssicherungs-)Massnahmen sind diagnostischen Massnahmen und messen das Qualitätsniveau. Diese statischen (“prüfen“) und dynamischen (“testen“) Massnahmen sammeln gezielt und mit analytischen Mitteln Informationen. Sie geben somit den Status der Anwendungsentwicklung zum jeweiligen Prüf-/Testzeitpunkt wieder.

[14] Quellen: vgl. [Jo96], [Ko02]

Misserfolgsfaktoren der Prozesse zur Unterstützung der Softwareumsetzung[15]

- Fehlen von Qualitätssicherung (insbesondere fehlende Integrationstests, Audits und Reviews), keine entwicklungsbegleitenden Code-Inspektionen und Design-Reviews
- Nicht vorgesehene Abnahmekriterien
- Hoher Grad an Inkompatibilität

Misserfolgsfaktoren speziell bei externer Vergabe von Anwendungsentwicklungen

- Defizite in extern ausgeführten Aufgaben oder extern hergestellten Komponenten

Der erste Punkt genereller Misserfolgsfaktoren der Prozesse zur Unterstützung der Softwareumsetzung stellt hier die Umkehrung obigen Erfolgsfaktors dar; Methoden der analytische Qualitätssicherung fehlen und fundierte Aussagen über den Zustand oder Fortschritt der Anwendungsentwicklung können somit nicht erfasst und dokumentiert werden. Der zweite Punkt benennt einen Teilbereich dieses Themas: Tests o.ä. können nicht sinnvoll durchgeführt werden, wenn ein Sollzustand undefiniert oder unbekannt ist, gegen diesen nämlich das 'Ist' getestet werden soll. Hieraus, sowie aus fehlerhaften oder undefinierten Anforderungen entstehen Inkompatibilitäten, insbes. bei externer Vergabe.

4.2.3 Prozesse der Softwarewiederverwendung[16]

Die Prozesse zur Softwarewiederverwendung nehmen, im Vergleich zu anderen Prozessen, im Rahmen von Softwareentwicklungsprojekten eine besondere Stellung ein, da sie über die Grenzen eines einzelnen Projektes hinweg Anwendung finden. Deren Intention ist es, Organisationen bei der projektübergreifenden Wiederverwendung (ein Aspekt einer nachhaltigen Anwendungs- und Produktentwicklung) einzelner Softwareelemente zu unterstützen (vgl. [12207], S. 17).
Der Domain Engineering Prozess unterstützt auf Basis aktueller und angenommener Bedürfnisse von Stakeholdergruppen die Softwarewiederverwendung durch die Ausgestaltung und Verwaltung thematisch abgegrenzter Bereiche, auch Domänen genannt. Aufgabe des Prozesses ist, die Festlegung der Struktur und des Umfangs solcher Domänen sowie der entsprechende Aufbau und die Pflege der darin zusammengefassten Objekte, wie zum Beispiel Softwarecode oder Dokumentationsmuster (vgl. [12207], S. 78ff.). In direktem Zusammenhang damit steht der Reuse Asset Management Prozess, welcher die Verwaltung einzelner Wiederverwendungsobjekte übernimmt. Aufgaben dieses Prozesses sind zum Beispiel die Implementierung einer adäquaten Speicherungslösung für die zu verwaltenden Objekte sowie die objektbezogene Überwachung der Wiederverwendung. Sollten Probleme mit einzelnen Wiederverwendungsobjekten auftreten, trägt der Prozess auch eine Informations- beziehungsweise Berichtsfunktion (vgl. [12207], S. 80ff.).

[15] Quellen vgl. [Bo91], [BJS00], [FLS02], [Ko02]
[16] vgl. [St09], S. 55f.

Zudem zählt der Reuse Program Management Prozess zu dieser Gruppe. Seine Aufgabe ist die Planung und Steuerung der Softwarewiederverwendungstätigkeiten des Unternehmens. Auch sind Überwachungsaktivitäten und die gezielte Förderung der Softwarewiederverwendung Gegenstand des Prozesses (vgl. [12207], S. 82).

Der gesamte vorliegende Bereich der Softwarewiederverwendung ist als ein Erfolgsfaktor in Softwareentwicklungsprojekten zu berücksichtigen (vgl. [Va05], S. 29; [Jon96], S. 6).
Werden einzelnen Codebausteine oder ganze Softwarekomponenten projektübergreifend wiederverwendet, kann dies eine verkürzende Wirkung auf die Entwicklungszeit haben. Grundlage hierfür ist die gute Dokumentation der entsprechenden Komponenten, so dass die Entwickler notwendiges Wissen für die Implementierung leicht erlangen können (vgl. [Va05], S. 29).

5 Fazit

Die Erkenntnisse, welche dieser Beitrag der doch bereits reichlich vorhandenen Literatur zu Erfolgs- bzw. Misserfolgsfaktoren hinzufügt, bestehen in der Prüfung dieser Faktoren auf praxeologische Relevanz sowie Kategorisierung / Einordnung der, in den Experteninterviews empirisch als wichtig und besonders praxisrelevant begründeten Faktoren, direkt in ein Schema, welches gängigen Vorgehens- und Reifegradmodellen oft zugrunde liegt.

Literaturverzeichnis

[AW05] Amberg, M. und Wiener, M. (2006). Kritische Erfolgsfaktoren für Offshore-Softwareentwicklungsprojekte, eine explorative Studie des Lehrstuhls Wirtschaftsinformatik III, Friedrich-Alexander-Universität Erlangen-Nürnberg; URL: http://www.competence-site.de/it-outsourcing/Kritische-Erfolgsfaktoren-fuer-Offshore-Softwareentwicklungsprojekte (Zugriff am 30. Sep. 2010)

[Be08] Bennicke, M., Hofmann, A., Lewerentz, C. und Wichert, K. H. (2008), Software Controlling, in: Informatik-Spektrum, 31, 2008, 6, S. 556–565

[Bo91] Boehm, B. (1991), Software risk management: principles and practices, in: IEEE Software, 8, 1991, 1, S. 32–41

[BJS00] Boos, M., Jonas, K. J. und Sassenberg, K. (Hrsg.) (2000), Computervermittelte Kommunikation in Organisationen, Bd. 3, Internet und Psychologie: Neue Medien in der Psychologie, Göttingen, Bern u. a. 2000.

[BG10] Bologna-Glossar (2010). Universität Paderborn, URL: http://www2.uni-paderborn.de/~Studienreform/Einrichtung%20STUGA/Bologna-Glossar.htm (Zugriff am 19. Juni 2010)

[BI09] Bibliographisches Institut & F. A. Brockhaus AG (2009), Brockhaus Lexikon; URL: http://www.brockhaus.de/suche/index.php?begriff=Faktor&bereich=mixed&x=0&y=0 (Zugriff am 24. Jan. 2009)

[CS10] California State University Monterey Bay (2010), Information Technology – Data Warehouse Glossary, URL: www.csumb.edu/site/x7101.xml (Zugriff am 30. Sep. 2010)

[CM10] Carnegie Mellon University (2008). WebPage des Software Engineering Institute der Carnegie Mellon University, Pittsburgh (USA); URL: http://www.sei.cmu.edu/cmmi/ (Zugriff am 29. Sep. 2010)

[Fe04] Feyhl, A. W. (2004), Management und Controlling von Softwareprojekten. Software wirtschaftlich auswählen, entwickeln, einsetzen und nutzen, 2. Aufl., Wiesbaden 2004.

[FLS02] Frühauf, K., Ludewig, J. und Sandmayr, H. (2002), Software-Projektmanagement und -Qualitätssicherung, 4. Auflage, Zürich 2002.

[Hi06] Hindel, B., Hörmann, K., Müller, M. und Schmied, J. (2006). Basiswissen Software-Projektmanagement: Aus- und Weiterbildung zum Certified Professional for Project Management nach iSQl-Standard, 2., überarb. und erw. Aufl., dpunkt.verlag GmbH, Heidelberg, ISBN: 3-89864-390-5

[Jo96] Jones, C. (1996), Patterns of Software System Failure and Success, London und Boston 1996.

[Ke98] Keil, M., Cule, P. E., Lyytinen, K. und Schmidt, R. C. (1998), A framework for identifying software project risks, in: Commun. ACM, 41, 1998, 11, S. 76–83

[Ko02] Kotulla, A. (2002), Management von Softwareprojekten: Erfolgs- und Misserfolgsfaktoren bei international verteilter Entwicklung, Diss.: Universität Hohenheim, Wiesbaden 2002.

[SBM83] Schmitz, P., Bons, H. und van Megen, R. (1983). Software-Qualitätssicherung - Testen im Software-Lebenszyklus, 2., durchgesehene Auflage, Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig und Wiesbaden 1983.

[SG01] Standish Group International (2001), Extreme Chaos, URL: http://www.standishgroup.com/sample_research/showfile.php?File=extreme_chaos.pdf, (Zugriff am 2. März 2009)

[St09] Stewen, Michael (2009), Erfolgsfaktoren in extern vergebenen Softwareentwicklungsprojekten Diplomarbeit am LSt. Wirtschaftsinformatik II, Universität Stuttgart, 2009.

[Va05] Valtinat, T. (2005), Entwurf von Softwareanpassungsprojekten bei gescheiterter Inbetriebnahme, Diss.: RWTH Aachen, Düsseldorf 2005.

[VM10] V-Modell XT, Version 1.3 (2010). Dokumentation des V-Modell XT als .pdf (Hrsg.): Der Beauftragte der Bundesregierung für Informationstechnik; URL: http://ftp.tu-clausthal.de/pub/institute/informatik/V-Modell-xt/Releases/1.3/V-Modell-XT-Gesamt.pdf (Zugriff am 30. Sep. 2010)

[12207] ISO/IEC 12207:2008(E), Systems and Software Engineering – Software Life Cycle Processes, 2008.

[15288] ISO/IEC 15288:2008(E), Systems and Software Engineering – System Life Cycle Processes, 2008.

[15504] ISO/IEC 15504-5:2006 Process Assessment Model