

Ausgabe 2004 / 1

Erscheinungsart: ca. 4 x jährlich in elektronischer Form

IT-Projektmanagement

weitere in dieser Ausgabe ...

- ⇒ IT-Projektmanagement - Sind Software-Projekte anders?
- ⇒ Die Quality Factory - Ein Modell zur Reduzierung von Projektrisiken in großen IT-Projekten (Gastbeitrag des TÜV-Informationstechnik)
- ⇒ 10 Goldene Regeln (Erfolgsfaktoren) des SW-Projektmanagements
- ⇒ Seminare / Workshops, Zitate

Kurzdefinition / Glossar ...

- ⇒ Ein **Projekt** ist ein Vorhaben, das im Wesentlichen durch die Einmaligkeit der Bedingungen in Ihrer Gesamtheit gekennzeichnet ist, z.B. Zielvorgabe, zeitliche, finanzielle, personelle und andere Begrenzungen, Abgrenzungen gegenüber anderen Vorhaben und projektspezifische Organisation. [DIN 69901]

Zum Verständnis und Dynamik des Begriffs 'IT-Projektmanagement' ...

Was verstehen wir eigentlich unter einem (IT-)Projekt?

Beginnend bei der allgemeinen DIN/ISO-Normdefinition gibt es unzählige weitere Definitionen und Erklärungen dieses Begriffs. Eine interessante Gliederung geben Hölzle und Grüning (siehe Grafik). Diese Gliederung ist in der Praxis gut nachvollziehbar.

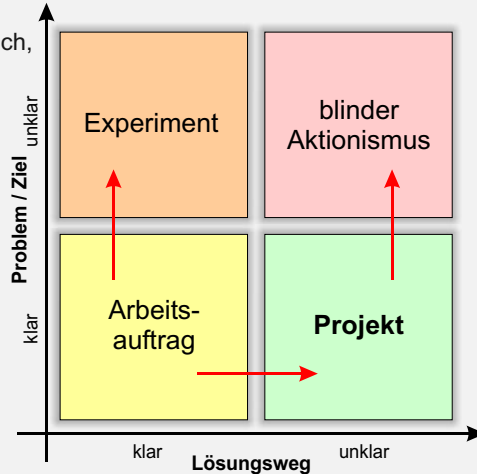
Interessante Aspekte ergeben sich, wenn man diese Betrachtung noch um die zeitliche Dynamik im Bereich der IT-Aufgaben, Projekte, ... erweitert.

In der Grafik sind die in der Praxis häufig vorkommenden Übergänge skizziert.

So wird z.B. aus einem kleinen *Arbeitsauftrag*, für den Ziel und Weg am Anfang klar sind, des Öfteren ein *Projekt* (wenn sich plötzlich zeigt, dass der Lösungsweg doch nicht in der geplanten Form umsetzbar ist).

Mitunter wird daraus jedoch auch ein *Experiment*, wenn im Laufe der Arbeit möglicherweise das Ziel ändert (z.B. durch andere Zielvorgaben vom Kunden, Management, ...). Eine vielfach bekannte und zumeist kritische Zustandsänderung ist, wenn sich in einem *Projekt* die klaren Problemdefinitionen und Ziele gravierend verändern und unklar werden oder ev. ganz aus den Augen verloren werden. In Kombination mit Management-Fehlern des Projektleiters (z.B. wenn die falschen Maßnahmen aufgrund fehlender Erfahrung oder aufgrund von schlechten Vorgehensmodellen ergriffen werden) besteht hier die große Gefahr, dass daraus *'blinder Aktionismus'* wird.

In dieser Situation wird beim Versuch, wieder zu einem ordnungsgemäßen Projekttablauf zurückzukehren, sehr oft eine kontraproduktive Strategie angewandt, die mit dem *Gesetz der abnehmenden Reaktion* zitiert werden kann: *'Je mehr Druck sie ausüben, desto weniger bekommen Sie dafür zurück!* Empfehlenswert wäre es jedoch, statt dessen positiv orientierte Maßnahmen zu ergreifen, um das Projekt wieder zu stabilisieren.



Quelle: Hölzle/Grüning: Projektmanagement: professionell führen - Erfolge präsentieren



**IT-Projekte
Quo vadis?**

Wir wissen, dass viele IT-Projekte nicht erfolgreich abgewickelt werden.

Andererseits kennen wir doch meist auch die Fallstricke und Probleme bei IT-Projekten, die dazu führen, dass diese Situation nun schon seit Jahrzehnten immer weiter besteht.

Theoretisch wüssten wir also, was zu tun ist, um die Situation zu bessern!

Trotzdem kommen IT-Projektleiter immer wieder in derartige Situationen und sehen sich dann außerstande, IT-Projekte erfolgreich abzuschließen.

Warum ist dies so? Sind wir IT-Leute aus unserem vergangenen (IT-)Leidensweg heraus schon so geprägt, dass wir diese Situation als normal ansehen oder ergeben wir uns einfach mangels Aussichtslosigkeit in diesen Situationen?

Es gibt viele Gründe, von denen einige auch in diesem Newsletter angesprochen werden. Ich denke jedoch, dass die Ursachen oft 'hausgemacht' sind und außerdem viele 'weiche' Faktoren wie Kommunikationsprobleme, Angst, Misstrauen, ... als Verstärker wirken.

Dieser Newsletter soll ein weiterer kleiner Versuch sein, die schwierige Situation in vielen IT-Projekten aufzuzeigen und Verbesserungsimpulse zu initiieren.

Es wäre schön, wenn es dadurch gelingt, den einen oder anderen Anstoß zum Nachdenken und zu Verbesserungen zu geben.

Dipl.-Ing. Johannes Bergmann

allgemein gerichtlich beideter und
zertifizierter Sachverständiger für Informatik

Der Quality-Newsletter ist ein periodisches Informationsmedium von Software Quality Lab und dessen Partnern mit Schwerpunkt Software-Qualitätsmanagement.

Inhalt: fachliche Beiträge und Schwerpunktthemen, Vorstellung neuer Produkte und Leistungen, neue wissenschaftliche Erkenntnisse, Branchen-News, ...

Aktuelle Fach- und Forschungsbeiträge sind willkommen. Einsendungen an info@software-quality-lab.at.

Weitere Infos zu diesem und anderen Themen finden Sie auf <http://www.software-quality-lab.at>.

IT-Projektmanagement - Sind Software-Projekte anders?

von Dipl.-Ing. Johannes Bergsmann

Immer noch sind erschreckende Statistiken im Bereich Software-Projektentwicklung ein Faktum. Verschiedene Statistiken gehen davon aus, dass ca. 25% aller Softwareprojekte komplett scheitern und weitere ca. 50% nicht termin- oder kostengerecht fertiggestellt werden. Könnten andere Engineering-Branchen wie die Fahrzeugindustrie, der Anlagenbau, die Elektronikindustrie, ... mit diesen oder ähnlichen Tatsachen überleben? Warum ist dies in der Software-Branche anders? Muss dies so sein?

Software-Projekte ...

Ein Software-Projekt ist im Grunde nichts außergewöhnliches im Vergleich zu anderen Projekten. Es ist ebenfalls gekennzeichnet durch die klassischen Projekteigenschaften:

- ➔ inhaltliche Zielsetzung und Neuartigkeit
("Ich möchte ein neues Warenwirtschaftssystem")
- ➔ zeitliche Zielsetzung
("Echtbetrieb des neuen Systems ab 1.9.2004")
- ➔ beschränkte Ressourcen
("Kosten darf es max. €500.000,-")
- ➔ Komplexität
("Es gibt verschiedenste Schnittstellen zu anderen Systemen, die Altdaten sind zu übernehmen, ...")

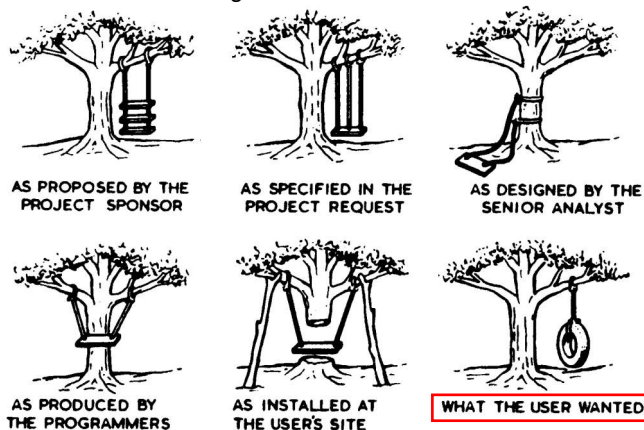
Daraus resultiert natürlich ein entsprechend erhöhtes Risiko (technisch, zeitlich, organisatorisch, personell, finanziell, ...).

Um diese Risiken einigermaßen beherrschbar zu machen, gibt es eine Vielzahl an Methoden und Vorgehensweisen, die in vielen Engineering-Bereichen auch meist mit Erfolg angewandt werden:

- ✓ Detaillierte und klare Spezifikationen
- ✓ Ausführliche Projektplanung
- ✓ Risikoanalyse des Projekts (vor Projektstart)
- ✓ begleitendes Qualitätsmanagement
- ✓ begleitendes Projekt-Controlling (laufend)
- ✓ Ausführliche Tests während der Entwicklung und vor der Auslieferung
- ✓ ...

Für IT-Projekte gibt es diese Methoden und Vorgehensweisen natürlich ebenfalls. Die Wissenschaft hat seit Beginn der ersten Software-Projekte entsprechende Methoden bereitgestellt und weiterentwickelt.

Doch die Praxis zeigt oft ein anderes Bild:



... sind anders?

Warum werden entsprechende Methoden und Techniken in den IT-Projekten oft nicht verwendet oder angenommen (Ausnahmen bilden hier meist die sicherheitskritischen Bereiche)?

Die Gründe dafür sind sehr vielschichtig. Aus Sicht der IT stellen sich die Kernpunkte in den vorher genannten Methoden und Vorgehensweisen wie folgt dar:

➔ Detaillierte und klare Spezifikation

Ein gutes Projekt beginnt normalerweise mit einer Analyse und Spezifikation und wird maßgeblich durch die Qualität der Spezifikation beeinflusst. Es gilt zumeist das Motto „Sage mir wie dein Projekt beginnt und ich sage Dir, wie es endet.“

Da eine gute Spezifikation auch aufwändig zu erstellen ist, ist einer der Hauptfehler, dass hier aus Zeit- und Ressourcen-Gründen versucht wird, zu sparen.

In vielen Fällen wird auch die Verantwortung auf die Lieferanten abgeschoben. Aus Präsentationen und Vorschlägen der Anbieter wird versucht, Ideen zu sammeln und schnell zu einer groben Leistungsbeschreibung zu kommen, die dann oft als Pflichtenheft bezeichnet wird. Manche Auftraggeber geben sich auch noch der Illusion hin, sich mit einer Präsentation und eventuell kurzen Testinstallation auf ein System festlegen zu können. Wenn dies im Zusammenhang mit einer ‚Schmalspur‘-Spezifikation / Ausschreibung praktiziert wird, handelt der Auftraggeber nach dem beim Autokauf bekannten Prinzip „Wie besichtigt und probefahren“. Der Lieferant kann dann relativ leicht Zusatzkosten geltend machen, wenn im Nachhinein vom Auftraggeber Leistungen und Funktionen erwartet werden, die in der präsentierten Software nicht enthalten waren.

Ein weiteres Faktum im Bereich der Spezifikation ist, dass sehr oft nur die funktionalen Bereiche der Software („Das System soll die Artikel wie folgt erfassen ...“, „Wir benötigen einen Report X“, ...) im Detail spezifiziert werden. Die nicht funktionalen Bereiche (Qualität, Ergonomie, Sicherheit, ...) werden zumeist - wenn überhaupt - nur sehr ungenau spezifiziert. Die Gründe sind vor allem, dass diese Bereiche einerseits sehr subjektive Eindrücke des Auftraggebers widerspiegeln und andererseits sehr schwer ‚greifbar‘ sind und so einen großen Aufwand für die Spezifikation bedeuten.

Leider führen in Software-Projekten jedoch genau diese nicht funktionalen Bereiche oft zu Konflikten und nachträglich höheren Aufwänden.

Ein exzellenter Projektleiter und ein ebensolches Projektteam können natürlich Fehler in der Spezifikation zudecken und dem Projekt trotzdem noch zu einem Erfolg verhelfen. In den meisten Fällen jedoch werden Mängel und Fehler in der Spezifikation den Projektverlauf massiv negativ beeinflussen.

Fortsetzung auf nächster Seite >>>

➤ Ausführliche Projektplanung und Risikoanalyse des Projekts

Dies ist nur dann sinnvoll möglich, wenn vorher auch die Anforderungen klar spezifiziert wurden. Da dies - wie vorher dargestellt - oft nicht oder nur unzureichend passiert, wird die Projektplanung und Risikoanalyse in diesem Fall nur auf der Basis von Vermutungen, Annahmen und vagen Schätzungen durchgeführt, sodass das Projekt dann ab einem gewissen Komplexitätsgrad zwangsläufig zum Scheitern verurteilt ist.

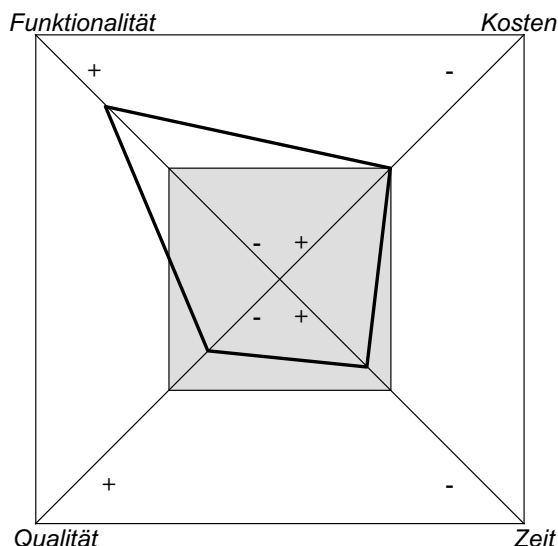
➤ begleitendes Qualitätsmanagement und ausführliche Tests

Qualität kostet (kurzfristig betrachtet) Geld! Und die meisten Auftraggeber beauftragen natürlich entsprechend sparsam. Jedoch wird hier leider nur dem Anschein nach und sehr kurzfristig gespart.

Schlechte Qualität wirkt sich meist langfristig aus (z.B. in steigenden Wartungs- und Betreuungskosten). Oft können durch schlechte Qualität verursachte Kosten auch nicht leicht erfasst werden und verschwinden dann meist in den Gemeinkosten (z.B. wenn aufgrund eines Designfehlers die Anwender für einen Vorgang 5 Minuten benötigen, der bei optimalem Design in der Hälfte der Zeit abzuwickeln wäre).

➤ begleitendes Controlling (laufend)

Ein Software-Projekt befindet sich immer in einem Spannungsfeld aus den Parametern Funktionalität, Zeit, Kosten und Qualität.



Diese Parameter müssen permanent kontrolliert werden und bei Abweichungen oder Anomalien müssen entsprechende Maßnahmen zur Gegensteuerung durchgeführt werden.

Durch die Projektdynamik verändern sich diese Parameter im Projektverlauf fast unvermeidbar von selbst und führen bei mangelndem Projektcontrolling am Projektende oft zu unliebsamen Überraschungen.

Dabei ist es meist so, dass sich die Funktionalität/Leistung mit der Zeit immer weiter erhöht. Da die Kosten oft pauschaliert sind und Terminüberschreitungen auch nur in geringem Maß toleriert werden, ist der einzige verbleibende variable Parameter die Qualität, die oft zu wenig spezifiziert wurde und damit schlecht kontrollierbar ist. Dadurch sinkt in solchen Fällen die Qualität der Ergebnisse entsprechend.

Weitere Knackpunkte für den Erfolg von Softwareprojekten liegen nicht in den funktionalen oder technischen Bereichen sondern sind oft nicht technische wie z.B. die Person des Projektleiters, die Kommunikation zwischen den Projektpartnern, ...

Durch den permanenten Druck der IT-Industrie mit ihren extrem kurzen Innovationszyklen und der dadurch millionenfach verkauften fehlerbehafteten Software haben wir in vielen Bereichen gelernt, mit Software-Fehlern zu leben.

Da leider in der Vergangenheit viele Personen, Unternehmen und Organisationen schlechte Erfahrungen mit Software-Projekten und -Produkten gemacht haben, leiden diese von vornherein unter einem schlechten Image:

- Software-Projekte kosten immer mehr, als man vereinbart.
- Software ist fehlerbehaftet, man muss mit den Fehlern leben.

Zusätzlich kommt erschwerend hinzu, dass Software allgegenwärtig ist, jeder damit in Berührung kommt und suggeriert wird, dass der Computer und die Software kinderleicht zu bedienen sind.

Durch neue Technologien wie Web-Services oder Rapid-Application-Development werden derartige Fehleinschätzungen noch weiter verstärkt, da die Hersteller dieser Tools und Technologien glaubhaft machen, dass auch komplexe Software-Projekte nur noch sehr wenig Aufwand bedeuten (vgl. Werbe-Aussagen wie „Web-Shop-Erstellung in wenigen Minuten“).

Da das alles so einfach und unkompliziert präsentiert wird, müssen dann auch die Software-Projekte, die man im eigenen Unternehmen startet oder beauftragt, entsprechend einfach abzuwickeln sein.

Natürlich spielt auch die Technologie eine wichtige Rolle. Jedoch ist diese Rolle einerseits bei weitem nicht so gewichtig, wie es manchmal dargestellt wird, und andererseits auch nicht an erster Stelle ("Bevor man eigentlich noch genau weiß, was man eigentlich benötigt, weiß man jedenfalls schon, dass die neue Lösung web-service-fähig sein soll!").

SW-Projekte sind klassische Engineering-Projekte!

Software-Projekte sind objektiv betrachtet eigentlich nicht anders als klassische Engineering-Projekte!

Um die oft vorherrschende problematische Situation bei vielen Software-Projekten zukünftig zu verbessern, ist auch eine Bewusstseinsbildung bei Entscheidungsträgern und Projektverantwortlichen notwendig.

Es gibt einige einfache Grundsätze, die jeder am Projekt Beteiligte wissen und anwenden sollte, um grobe Fehleinschätzungen und Fehlschläge zu vermeiden:

- Software-Projekte sind (fast immer) sehr komplex.
- Software-Projekte sind wie ein klassisches Engineering-Projekt zu sehen und daher auch so abzuwickeln.
- Software, die an die eigenen Bedürfnisse angepasst wird (auch wenn es sich um günstige Standard-Software handelt), kostet entsprechend viel (wenn man ein günstiges Auto kauft und es gerne etwas tiefer gelegt, um 30 cm verlängert und mit gestreifter Lackierung haben will, kostet dies ja auch unverhältnismäßig mehr als die Standard-Ausführung).
- Qualitativ hochwertige Ingenieur-Leistungen benötigen entsprechende Zeit für Planung und Abwicklung (ein Haus wird ja meistens auch nicht nach der ersten schnellen Handskizze gebaut, sondern bedarf detaillierter Entwürfe und Planungen, um die Bedürfnisse optimal abzudecken).

Die Quality Factory - Ein Modell zur Reduzierung von Projektrisiken

von DKfm. Werner Achtert

Prokurist und Geschäftsstellenleiter, TÜV Informationstechnik

Viele IT-Projekte bergen erhebliche Risiken, man muss davon ausgehen, dass nur ein Drittel aller größeren Entwicklungsprojekte innerhalb des geplanten Budget- und Zeitrahmens abgeschlossen werden. In jüngster Vergangenheit haben einige spektakuläre Projekte wie z.B. das deutsche Mautsystem für Aufsehen gesorgt. Auf Basis langjähriger Erfahrung stellt dieser Beitrag Konzepte zur Realisierung der Qualitätssicherung in großen IT-Projekten vor.

Betrachtet man Problem-Projekte, so fallen einige typische Gemeinsamkeiten auf:

- Es handelt sich durchwegs um umfangreiche Projekte mit anspruchsvollen fachlichen und technischen Anforderungen.
- Es kommen neue Technologien zum Einsatz, mit denen die Beteiligten nur wenig Erfahrungen haben.
- Die Projekte stehen unter hohem Termindruck.
- Die Auftragnehmer sind große, renommierte Systemhäuser mit langjähriger Projekterfahrung.

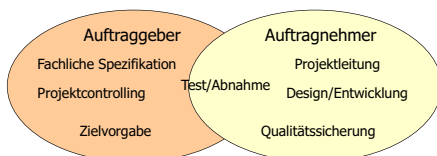
Da trotz der Erfahrung der Systemhäuser, trotz oftmals aufwändigen Vorgehensmodellen und Qualitätsmanagementsystemen solche Projekte immer wieder scheitern, muss die kritische Frage nach den Ursachen gestellt werden.

Ausgangspunkt der Überlegungen ist die Tatsache, dass in allen Projekten Fehler gemacht werden. Manche Projekte kommen jedoch trotz Fehlern zum Ziel, andere scheitern vollkommen.

Offensichtlich sind also oft nicht die Fehler an sich die Ursache für das Scheitern von Projekten, sondern auch die Art und Weise, wie mit den Fehlern umgegangen wird.

Die Erfahrung in vielen Projekten zeigt, dass Fehler oftmals nicht rechtzeitig gefunden werden und auf gefundene Fehler nicht schnell und konsequent genug reagiert wird.

Es fällt auf, dass in vielen Projekten die Qualitätssicherung zwar durchaus vorhanden ist, aber aus verschiedenen Gründen durch den Auftragnehmer realisiert wird. Zum Einen will der Auftraggeber die Verantwortung für die Abwicklung für das gesamte Vorhaben in einer Hand haben. Zum Anderen hat der Auftraggeber oftmals nicht die Kapazität bzw. Erfahrung zur Qualitätssicherung in komplexen IT-Projekten.



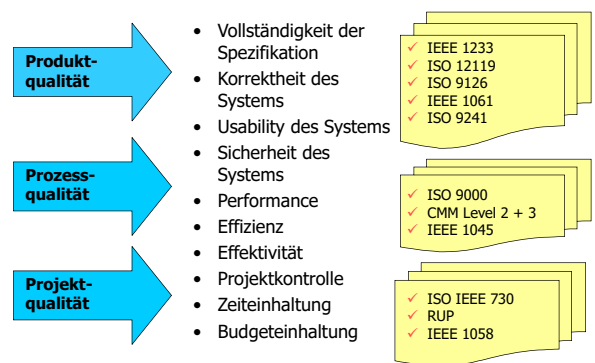
Qualitätssicherung in der Verantwortung des Auftragnehmers

Wenn die Verantwortung für die Qualitätssicherung in der Hand des Auftragnehmers liegt, besteht jedoch die Gefahr, dass Fehler zwar erkannt, aber nicht konsequent eskaliert werden. Dies ergibt sich natürlicherweise aus dem Spannungsfeld zwischen der Projektleitung einerseits (z.B. hoher Termin- und Kostendruck, Zwang zur wirtschaftlichen Projektabwicklung) und der Qualitätssicherung andererseits (z.B. Einhaltung von Qualitätsstandards, langfristige Wartbarkeit).

Damit besteht die Gefahr, dass die Qualitätsziele des Auftraggebers nicht umgesetzt werden, weil keine unabhängige

Qualitätssicherung und Risikobetrachtung stattfindet.

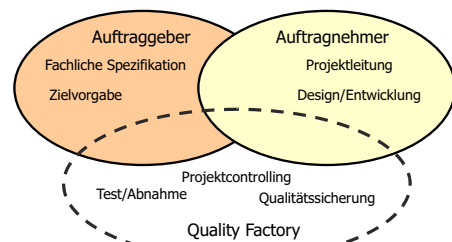
Unabhängigkeit der Qualitätssicherung ist somit ein wichtiger Erfolgsfaktor für Projekte, um Abweichungen im Projektverlauf und der Qualität der Ergebnisse zu erkennen und darauf zu reagieren.



Qualitätsaspekte und -Standards

Wenn im Verlauf eines Projektes unvorhergesehene Änderungen der Rahmenbedingungen auftreten, so kann es durchaus nötig und gerechtfertigt sein, die Qualitätsziele den Erfordernissen anzupassen. Dies muss jedoch immer in einem kontrollierten Prozess geschehen, in dem Auftraggeber und Auftragnehmer solche Veränderungen mit einer ausreichenden Begründung dokumentieren. Finden solche Absprachen informell statt, wie es leider immer wieder in Projekten zu beobachten ist, so ist gerade bei lang laufenden Projekten später kaum mehr nachvollziehbar, wer eine Abweichung von den Qualitätsvorgaben zu verantworten hat.

Um eine echte Unabhängigkeit der Qualitätssicherung in IT-Projekten sicherzustellen, bietet sich die Verlagerung dieser Aufgaben an einen unabhängigen Dritten an, wie dies auch in anderen Wirtschaftszweigen längst üblich ist. So müssen z.B. Hersteller und Betreiber von Aufzügen zur Prüfung der Sicherheit immer einen unabhängigen Dritten einschalten.



Unabhängige Qualitätssicherung

Fortsetzung auf nächster Seite >>>

Fortsetzung - Die Quality Factory >>>

Das hier vorgeschlagene Konzept einer unabhängigen Qualitätssicherung basiert auf der Zusammenfassung aller Funktionen zum Qualitäts- und Risikomanagement in einer Quality Factory, die durch einen unabhängigen Dritten betrieben wird.

Eine Quality Factory umfasst folgende Bereiche:

➤ Projektunterstützung

Im Rahmen der Projektunterstützung werden Projektmetriken eingeführt, um den Projektverlauf zu bewerten. Damit kann im Rahmen eines Coachings die Projektleitung beraten und unterstützt werden. Besondere Bedeutung hat hierbei das Lieferantenmanagement, da in vielen großen Projekten mittlerweile Zulieferungen von mehreren Lieferanten für einzelne Spezialthemen koordiniert werden müssen.

➤ Prozessunterstützung

Im Rahmen der Prozessunterstützung werden Prozessbeschreibungen als Grundlage für die wichtigsten Entwicklungsprozesse vorgegeben. Eine Qualitätsorganisation legt die Aufbau- und Ablauforganisation der qualitätsbezogenen Prozesse fest. Über die Verwendung von Prozessmetriken werden die Prozesse kontrolliert und gesteuert.

➤ Produktunterstützung

Im Rahmen der Produktunterstützung wird schließlich die Qualität der entwickelten Produkte überwacht. Dazu werden wiederum Metriken z.B. zur Bewertung von Sourcecode definiert, mit denen im Rahmen von Reviews die Einhaltung der Qualitätsziele kontrolliert wird.

Zu den konkreten Aufgaben der Quality Factory gehören somit

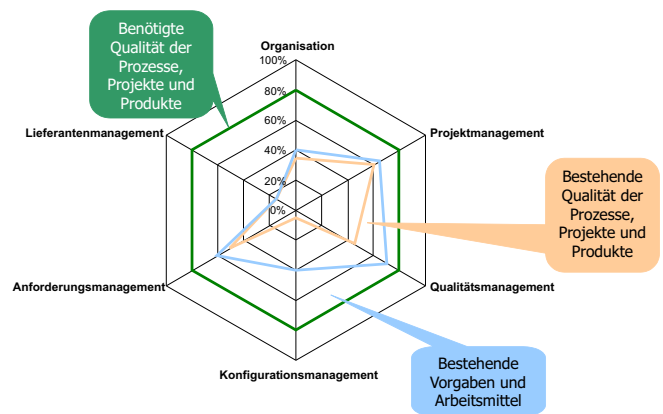
- ✓ Abstimmung des QMS zwischen Auftraggeber und Auftragnehmer
- ✓ Aufstellen projektbezogenen Qualitätspläne
- ✓ Organisation und Durchführung der Qualitätssicherungs-Maßnahmen
- ✓ Review von Entwicklungsergebnisse
- ✓ Audit des Projektverlaufs
- ✓ Betrieb eines Testcenters für Test und Abnahme von IT-Systemen
- ✓ Moderation zwischen Auftraggeber und Auftragnehmer bei Problemen im Projektverlauf
- ✓ Bestätigung und Zertifizierung der Einhaltung von Normen

Eine solche Quality Factory kann im Fall von Großprojekten für ein einzelnes Projekt installiert werden. In größeren Organisationen mit regelmäßigen IT-Projekten ist es wirtschaftlicher, die Quality Factory projektübergreifend zu organisieren.

Der Aufbau einer Quality Factory ist ein Prozess mit mehreren Stufen von der Bestandsaufnahme der aktuellen Q-Prozesse über die Einführung von Regelwerken und Werkzeugen bis zur Etablierung von Prozessen zur kontinuierlichen Verbesserung der Entwicklungsprozesse.

Die folgende Übersicht zeigt ein Stufenmodell zur Einführung einer Quality Factory:

- Stufe 0: Assessment und Analyse
- Stufe 1: Vorgabenerstellung
- Stufe 2: Vorgabenanwendung
- Stufe 3: Metrikkontrollierte Produkte und Prozesse
- Stufe 4: Unabhängiges Test-Center
- Stufe 5: Einheitliche und steuerbare Prozesse



Beispiel für ein Assessmentergebnis zur Einführung einer Quality-Factory

Praktische Erfahrungen zeigen, dass in durchschnittlichen Unternehmen Aufbau und Einführung einer solchen Quality Factory ein bis zwei Jahre in Anspruch nimmt. Diesem relativ hohen Aufwand steht allerdings mittelfristig eine Verringerung der Entwicklungskosten um 10-20% und eine Verringerung des Aufwand für Pflege der Systeme um bis zu 50% als Nutzen gegenüber.

Die Erfahrungen dabei sind durchweg positiv. Zum Einen müssen die Auftraggeber durch diese Art des Outsourcing keine eigenen Kapazitäten für die Qualitätssicherung aufbauen und können trotzdem auf das jeweils nötige Spezialwissen projektbezogen zurückgreifen. Zum Anderen sehen auch die Auftragnehmer externe Qualitätssicherung mittlerweile durchaus positiv.

Auch wenn die Qualitätssicherung durch einen unabhängigen Dritten zunächst als zusätzliche Belastung empfunden wird, so hilft eine unabhängige Sichtweise auch dem Entwickler, Probleme schon frühzeitig zu erkennen und gemeinsam mit dem Auftraggeber nach Lösungen zu suchen.

Viele Systemhäuser sehen mittlerweile, dass dies zu einer Verbesserung des Vertrauensverhältnis zwischen Auftraggeber und Auftragnehmer beiträgt und mittelfristig die Kundenzufriedenheit verbessert.

Zuletzt darf nicht übersehen werden, dass auch die Auftraggeber in großen Projekten durchaus gravierende Fehler machen, die von einer unabhängigen Qualitätssicherung ebenfalls gefunden werden können.

Somit fungiert die Quality Factory in vielen Fällen auch als Schlichter und Moderator zwischen Auftraggeber und Auftragnehmer in schwierigen Projektsituationen.

Damit ist die externe Qualitätssicherung durch eine Quality Factory für alle Projektbeteiligten ein wirksames Mittel zur Reduzierung von Projektrisiken.

10 Goldene Regeln (Erfolgsfaktoren) des SW-Projektmanagements

primär aus der Sichtweise eines Auftraggebers

1 Exakt und entsprechend detailliert spezifizieren!

Viele Probleme resultieren aus einer unklaren oder unvollständigen Spezifikation. Wer hier spart, spart am falschen Platz. Die Spezifikation soll inhaltlich vollständig sein und so detailliert, um das Projektrisiko in einem wirtschaftlich vernünftigen Rahmen zu halten.

Ab einem gewissen Detaillierungsgrad übersteigt jedoch der Aufwand für die Erstellung der Spezifikation den Nutzen durch die Risikominimierung. Außerdem erhöht sich ab einem gewissen Detaillierungsgrad der Änderungsaufwand für die Spezifikation entsprechend.

Es ist daher wichtig, auf dieser Gratwanderung zwischen Risikominimierung und Erstellungsaufwand die richtige Balance zu finden.

Wenn intern keine Zeit, zu wenig Know-How, Unsicherheiten bezüglich des richtigen Detaillierungsgrades oder ev. auch nur Schwierigkeiten bei der richtigen Formulierung vorhanden sind, sollte hier auf jeden Fall Hilfe von Außen hinzugezogen werden.

Im einfachsten Fall ist z.B. ein Spezifikationsreview sehr hilfreich. Dies ist eine der kostengünstigsten und effizientesten Arten, die Qualität einer Spezifikation zu verbessern.

Jedoch sollte man nicht den Fehler machen, einen potentiellen Software-Anbieter mit der Spezifikationserstellung oder dem Review zu betrauen. Dann erhält man sehr oft ein System, das der Anbieter liefern kann (möchte) und nicht das System, das man eigentlich benötigen würde.

2 Einen realistischen Kosten- und Zeitrahmen kalkulieren!

Erfahrungsgemäß ändern sich Anforderungen im Laufe des Projekts, was in der Folge zumeist eine Kostenerhöhung bedeutet. Eine einfache Regel, die bei den meisten Projekten zutrifft, hilft hier sehr oft:

Addiere bei ungenauer Spezifikation ca. 50-100% und bei genauer Spezifikation ca. 20% zu den Preisen und Projektzeiten der Anbieter).

Ein weiterer Punkt, der oft unterschätzt bzw. teilweise auch gar nicht berücksichtigt wird, sind die internen Kosten (Personalkosten, Kosten durch Betriebsstörungen und ev. sogar Produktionsstillstand in der Einführungsphase, ...)

Die internen Kosten dürfen auf keinen Fall vernachlässigt werden. Dies würde sonst die Projektkalkulation verfälschen und könnte im schlimmsten Fall sogar das Projekt gefährden!

3 Das System nicht nur aus der Sicht von heute planen, sondern die Anforderungen von morgen spezifizieren!

Es besteht sonst die Gefahr, dass man ‚alten Wein in neuen Schläuchen‘ erhält. Innovative Wege werden dadurch oft nicht erkannt und eventuell sogar auf Jahre blockiert.

Nutzen Sie die Chance und überdenken Sie im Rahmen der *Planung* für ein neues IT-System auch die bisherigen Prozesse und Strukturen.

4 Auch die nicht funktionalen Anforderungen berücksichtigen!

Anforderungen hinsichtlich Qualität, Ergonomie, Sicherheit, Dokumentation, Testdurchführung, ... werden oft gar nicht oder teilweise nur sehr ungenau spezifiziert.

Dies kann dann zum Projektende oder am Beginn des Echtbetriebs zu großen Konflikten zwischen Auftraggeber und Lieferant führen, da sich der Auftraggeber natürlich (implizit) auch eine entsprechende Qualität, ergonomische Handhabung vollständige Dokumentation, ... erwartet, der Lieferant jedoch bei seiner Angebotskalkulation all die Punkte nicht berücksichtigt hat, die nicht spezifiziert wurden und nicht unbedingt für die Erfüllung des Vertrags (=Spezifikation!) notwendig sind.

5 Auf die Projekt-Verantwortung nicht vergessen - auch der Auftraggeber benötigt einen Projektleiter!

Es kommt häufig vor, dass der Lieferant nach dem Projektstart bis zum vereinbarten Realisierungstermin großteils allein gelassen wird (z.B. aus Zeitmangel oder fehlendem Projektleiter-Wissen beim Auftraggeber).

Dadurch verliert der Auftraggeber die Kontrolle über das Projekt. Es ist unbedingt erforderlich, dass es auf Auftraggeberseite auch einen (1) Verantwortlichen für das Projekt gibt, der sich auch über den gesamten Projektverlauf entsprechend darum kümmert (Projektmanagement, Projektcontrolling, Qualitätssicherung, ...).

Wenn der Auftraggeber selbst dazu nicht in der Lage ist, kann diese Aufgabe eventuell auch ein externer Projektmanager übernehmen. Allerdings sollte dieser dann schon am Beginn des Projekts (ab der Planung) einbezogen werden. Denn wenn ein Projekt schon in einer aussichtslosen Situation ist, kann auch externe Hilfe meist nur noch Schadensbegrenzung betreiben.

6 Nicht zu sehr auf die Technologie konzentrieren.

Manche Projekt-Mitarbeiter (und teilweise auch Verantwortliche) sind oft 'technologie-verliebt' und konzentrieren sich daher möglicherweise zu stark auf technologische Feinheiten, die eigentlich für den Projektverlauf irrelevant sind. Dadurch geht viel wertvolle Zeit verloren.

Die meisten Projekte scheitern jedoch nicht an der Technologie, sondern an anderen Problemfeldern wie der Methodik, der Kommunikation, den Personen, ...

7 Darauf achten, dass möglichst alle künftigen ‚Stakeholder‘ berücksichtigt und in das Projekt einbezogen werden!

Wenn die späteren Nutznießer und auch Benutzer einbezogen werden, verläuft das Projekt meist zielgerichteter und die Akzeptanz wird deutlich verbessert.

Fortsetzung auf nächster Seite >>>

Fortsetzung - 10 Goldene Regeln >>>

8 Die Anbieter-Auswahl sorgfältig vornehmen!

Das Studium der Funktionsliste oder eine gute Verkaufs-Präsentation reicht oft nicht aus, um den wahren Gehalt eines Software-Systems beurteilen zu können.

Die Zeit für eine Probeinstallation, in der man sich dann natürlich auch entsprechend intensiv mit der Software auseinandersetzen sollte, ist sicherlich gut investiert.

Um den Blick für das Wesentliche nicht zu verlieren und sich nicht von einigen netten 'Gimmicks' blenden zu lassen, sollte außerdem eine detaillierte Nutzwertanalyse zur Bewertung durchgeführt werden.

Dadurch ist die Wichtigkeit der einzelnen Bereiche für beide Partner transparent und auch die Abnahme scheitert dann nicht an einer Kleinigkeit.

9 Die Balance im Spannungsfeld Funktionalität, Zeit, Kosten und Qualität halten und 'ein faires Spiel spielen'!

Die Komponente 'Funktionalität' wächst im Projektverlauf meist unscheinbar aber stetig aufgrund von Anforderungsänderungen des Auftraggebers.

Es ist oft so, dass der Auftraggeber dann meint, bei einem Fixtermin und pauschalierten Kosten dem Lieferanten auch noch verschiedenste zusätzliche Funktionalitäten und Anforderungsänderungen im Projektverlauf 'unterjubeln' zu können. Wenn der Druck entsprechend groß ist, wird dies vom Lieferanten auch hingenommen werden.

Der einzige Parameter, der dann noch für den Lieferanten veränderbar ist, ist die Qualität. Er wird daher dann versuchen, in diesem Bereich einen Ausgleich zu schaffen, da dies vom Auftraggeber meist nicht spezifiziert wurde und daher kaum kontrollierbar und einforderbar ist.

Und zumeist schlägt sich schlechte Qualität dann langfristig auch in der Wartung nieder. Der Auftraggeber wird daher oft für kurzfristige, scheinbare Verhandlungserfolge langfristig mit erhöhten Projektkosten zur Kasse gebeten.

10 Genügend Zeit für den Test nehmen und eine umfassende Abnahme der Software durchführen!

Gerade hier wird oft der Fehler gemacht, dass man nicht so genau prüft, ev. auch auf einen Probetrieb und eine detaillierte Abnahme verzichtet und sich auf die Gewährleistung verlässt.

Sofern eine Gewährleistungsfrist nicht ausgeschlossen wurde, muss der Lieferant natürlich eine gewisse Zeit lang nachbessern.

Jedoch ist das System dann eventuell schon im Echtbetrieb und daher jede Störung um ein Vielfaches teurer als vor der Inbetriebnahme.

Außerdem kann es sein, dass in dieser Zeit schon Wartungsarbeiten oder eventuell zusätzlich beauftragte Änderungen am System vorgenommen werden. Eine Trennung zwischen Aufwänden für Fehlerbehebung im Rahmen der Gewährleistung und kostenpflichtigen Wartungsarbeiten wird dadurch für den Auftraggeber immer schwieriger.

Seminare/Workshops ...

Projekt- und Qualitätsmanagement

- ⇒ Software Qualitätsmanagement - Effiziente Techniken und praxisorientierte Methoden im Überblick
- ⇒ Requirements Engineering und Spezifikation - Die Wiege des Projekt-(Miss-)Erfolgs
- ⇒ Spezifikationen ausschreibungstauglich erstellen
- ⇒ IT-Projektmanagement - Hintergründe und Tips für die Realisierung erfolgreicher Software-Projekte

Qualitätssicherung und Metriken

- ⇒ Reviews - Die effektivste Art, Fehler zu vermeiden!
- ⇒ Testen und Testorganisation - Wenn es der Kunde merkt, ist es zu spät!
- ⇒ Produkt- und Prozess-Metriken im Überblick - Nur mit Messen ist eine gezielte Verbesserung möglich!

'Best Practices' und Vorgehensmodelle

- ⇒ CMM(I) und SPICE im Überblick - anerkannte 'best practice'-Standards in der Software-Entwicklung
- ⇒ Vorgehensmodelle - Was ist die richtige Vorgehensweise für meine Software-Entwicklung?
- ⇒ ITIL - Mode-Erscheinung oder der Schlüssel zum erfolgreichen Management von IT-Prozessen?

IT-Controlling und IT-Revision

- ⇒ IT-Controlling (u.a. mit Schwerpunkt Balanced-Score-Card für die IT) - Mehr Effizienz in der IT!
- ⇒ IT-Revision - Die IT im Griff!
- ⇒ Qualitätsaspekte beim Outsourcing der IT sowie Prüfung und Überwachung von Software-Lieferanten

Weitere spezielle Themen

- ⇒ Aufwandsschätzung vor Vertragsabschluss - Wahrsagerei oder methodisch beherrschbar?
- ⇒ Bewertung von Software-Entwicklungen und IT-Systemen bei Unternehmensübernahmen, Konkurs, ...

Die angegebenen Seminare sind kompakte 1-Tages-Seminare / Schulungen in der Form von firmeninternen "Impuls-Workshops". Es sind jedoch prinzipiell auch andere Varianten (z.B. Kombination von Themen) denkbar oder detailliertere mehrtägige Workshops.

Spezielle Ausbildungen

- ⇒ 3-tägiger Ausbildungslehrgang zum ASQF Certified Tester. in Kooperation mit der imbus AG

Zitate ...

- ⇒ Hochproduktive Projekte wenden anteilmäßig sehr viel mehr Zeit für den Entwurf auf. *DeMarco, 'Der Termin'*
- ⇒ Eine Spezifikation, die keine vollständige Aufzählung der Ein- und Ausgaben enthält, ist ein Flop; sie erfüllt die Anforderungen an eine Spezifikation nicht einmal ansatzweise. *DeMarco, 'Der Termin'*
- ⇒ Der Titanic-Effekt: Der Glaube, dass eine Katastrophe ausgeschlossen ist, führt oft zu ungläublichen Katastrophen. *Weinberg, Systemdenken und Softwarequalität*
- ⇒ Warum haben wir eigentlich nie Zeit, die Sache richtig zu machen, aber immer Zeit, sie nochmal zu machen? *Weinberg, Systemdenken und Softwarequalität*

Impressum:

Herausgeber und für den Inhalt verantwortlich:

Die Beiträge wurden sorgfältig ausgewählt und bearbeitet.
Für Druckfehler und Irrtümer wird nicht gehaftet.

Software Quality Lab

Fliederstrasse 8
A-4222 Langenstein

www.software-quality-lab.at

info@software-quality-lab.at

Tel.: +43-(0)664-16 20 220, Fax: +43-(0)7237-4941