

PLM-Prototyping als Akzeptanzmotor

Mustafa Görmezer
Bremen

Die Einführung einer Lösung für das Product Lifecycle Management (PLM) ist in der Regel mit großen Aufwänden bei Planung und Umsetzung verbunden. Interne Anforderungsermittlung, Teambildung, Lastenhefte, Pflichtenhefte, Feinspezifikationen und nicht zuletzt teilweise immense Koordinationsaufgaben sind neben dem Tagesgeschäft von allen Beteiligten zu erfüllen, und dies lange bevor der Anwender auch nur den Hauch einer Bedienoberfläche zu Gesicht bekommt.

Kann dieser Ablauf den Vorstellungen der Anwender – die ja in diesem Prozess auch als Kunden betrachtet werden – genügen? Oder lassen sich zur Beschleunigung und Akzeptanzbildung im laufenden PLM-Projekt einige Ansätze aus dem »Rapid Prototyping« oder »Agile Prinzipien« verwenden?

Werden heutige Einführungsprojekte genauer betrachtet, lässt sich oft feststellen, dass gerade bei mittleren bis größeren PLM-Vorhaben ein enormer Zeit- und Kostenaufwand

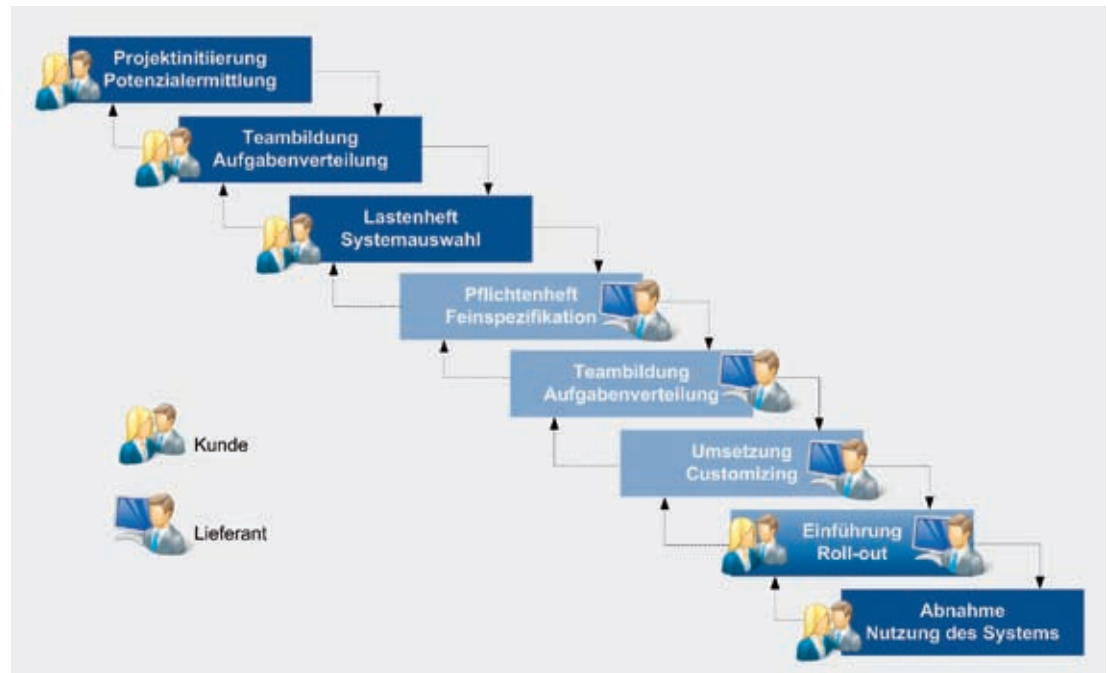
betrieben wird, um die gesteckten Ziele zu erreichen. Nicht selten machen große Projekte sogar in den Medien von sich reden, weil plötzlich unerwartet Schwierigkeiten – welcher Natur auch immer – auftreten. Mal ehrlich, wussten Sie im Jahr 2008, dass wir ein Jahr später eine der schwersten Wirtschaftskrisen der Nachkriegszeit erleben werden und dass Sie Ihr Projekt in 2009 nicht starten können? Selbst ein vorbildlicher Projektplan kann nicht immer verhindern, dass das Gesamtprojekt aus



verschiedenen Gründen in schwere See gerät – zumal nicht alles bis ins Detail vor auszuplanen ist. Doch was macht PLM-Projekte so aufwändig, dass so viele Risiken entstehen können? Da genügt ein Blick auf den typischen, linearen Projekttablauf, der auch als »Wasserfallmodell« bezeichnet wird (Bild 1). Offensichtlich kostet die Gesamteinführung eines PLM-Systems nach diesem Modell Zeit – zu viel, wie wir meinen. Speziell die Entwicklung des Lasten- und Pflichtenhefts verschlingt einen großen Teil des Budgets und birgt zudem Risiken, die sich oftmals erst im Projektverlauf zeigen. Diese mit dem klassischen Vorgehen verbundenen Nachteile gelten im Übrigen generell für alle IT-Einführungsprojekte, bei denen kundenindividuelle Anpassungen eine substantielle Rolle spielen. Wenn also die klassische Vorgehensweise bei vielen PLM-Projekten nicht den gewünschten Erfolg zeigt, warum nicht neue Wege gehen? Auch wenn sie noch selten zu finden sind: Es gibt bereits einige Beispiele, wo die aus der Software-Entwicklung bekannten Methoden des »Rapid Prototyping« oder »Agile Prinzipien« zu guten Ergebnissen bei der PLM-Einführung geführt haben.

Rapid Prototyping ist zum einen das in der Fertigungsindustrie etablierte Verfahren zur direkten Erzeugung von physischen Prototypen aus 3D-CAD-Daten. Zum anderen wird dieser Begriff verwendet, wenn der Prototyp eines Anwendungsprogramms schnell, aber nicht mit dem Anspruch auf funktionale Vollständigkeit erstellt wird. Dieser wird dann den zukünftigen Anwendern vorgestellt, um ihr Feedback frühzeitig in die Software-Entwicklung einfließen zu lassen. Im Kern haben beide Anwendungsfälle in ihrem jeweiligen Kontext dasselbe Ziel: treffsicher ein bedarfsgerechtes Produkt zu liefern. Die »Agile Software-Entwicklung« wird seit einiger Zeit als Alternative zu den herkömmlichen Verfahrensmodellen diskutiert, um schneller als bisher zu Lösungen zu kommen, und zudem die Anforderungen der Anwender besser erfüllen zu können.

Wir fassen hier unsere Praxiserfahrungen aus PLM-Einführungsprojekten mit Vorge-



hensweisen des »Rapid Prototyping« und der »Agilen Software-Entwicklung« unter dem Begriff »PLM-Prototyping« zusammen. Gegenstand des PLM-Prototyping ist nach unserem Verständnis die Adaption einer funktional hoch entwickelten, aus vielen Kundenwünschen entstandenen PLM-Standardanwendung an kundenspezifische Anforderungen. Hier ist also nicht die Entwicklung einer komplett neuen PLM-Applikation gemeint. Der Ansatz für das PLM-Prototyping leitet sich aus den Werten und Handlungsgrundsätzen ab, die im Jahr 2001 von den Mitgliedern einer amerikanischen Gruppe von Praktikern mit dem Namen »Agile Alliance« im »Agile Manifesto« [1] niedergeschrieben wurden. Im Kern wird dort gefordert, Anwender über eine persönliche Kommunikation dauerhaft in einen iterativen Entwicklungsprozess einzubinden (Bild 2). Das Endprodukt eines solchen Prozesses soll funktionierende Software sein, die genau die Erwartungen der Anwender trifft und schnell verfügbar ist. Damit einher gehen hohe Ansprüche an die fachlichen und kommunikativen Fähigkeiten der Teammitglieder im Entwicklungsprozess. Die permanente (Rück-) Koppelung zwischen entsprechend qualifizierten Entwicklern und Anwendern wird als effizienter für den Entwicklungsprozess angesehen als die einer vorab erstellten umfangreichen Spezifikation. Aus diesem Leitbild heraus entstanden praktisch anwendbare Methoden wie »Scrum« [2] oder »Extreme Programming«. Die klassischen Erfolgsfaktoren der Software-

Bild 1: Hier wird die Einführung einer IT-Plattform nach dem Wasserfallmodell dargestellt.

Das PLM-Prototyping hat seine Grundlagen im »Agile Manifesto«

Bild 2: Die Grafik verdeutlicht das iterative Vorgehensmodell beim PLM-Prototyping.



Entwicklung – wie eine gute Planung, eine sorgfältige Vertragsgestaltung und Dokumentation oder der Einsatz effizienter Werkzeuge – gelten ansonsten natürlich auch bei der »Agilen Software-Entwicklung«!

Bei PLM-Einführungen können die Werte und Prinzipien der »Agilen Software-Entwicklung« also durchaus eine Bereicherung sein: Grund genug, sie in die Praxis einfließen zu lassen. Doch welches sind die Voraussetzungen, um ein PLM-Projekt erfolgreich abzuwickeln?

Am Anfang der Überlegung, ob der Einsatz von PLM-Prototyping in einem Einführungsprojekt sinnvoll sein könnte, steht ein Blick auf die Unternehmenskultur der beteiligten Partner. Sowohl das Kundenunternehmen als auch der Software-Lieferant sollten Aufgeschlossenheit und Erfahrung in der Anwendung innovativer Vorgehensmodelle haben. Dazu gehören gelebte Werte von Innovationskulturen wie »Kunden- und Zielorientierung«, »Handlungsfreiheit«, »Eigenverantwortung und Vertrauen«, »Motivation und Dynamik« und »Offene Kommunikation« [3].

Die grundlegenden Voraussetzungen für den zukünftigen Erfolg

Eine weitere Voraussetzung für das PLM-Prototyping ist die Besetzung des Projektes mit geeigneten Mitarbeitern. Die »Key User« der Einführungsphase sollten die bereits angesprochenen fachlichen und kommunikativen Fähigkeiten sowie genügend Zeit mitbringen, da die enge und häufige Interaktion mit ihnen ein wesentliches Element des PLM-Prototyping darstellt. Auch spielt die Expertise der externen Projektmitglieder eine große Rolle: Ihre Aufgabe liegt speziell beim PLM-Prototyping darin, den Software-Adaptionsprozess auf Basis ihrer »best practices«-Erfahrungen aus anderen Projekten beratend und gestaltend zu begleiten. Wichtig ist nicht zuletzt auch eine der Vor-

gehensweise angepasste Vertragsgestaltung; Werkverträge über das Gesamtprojekt sind hier wenig geeignet. Entsprechend der iterativen Vorgehensweise beim PLM-Prototyping ist es vielmehr sinnvoll, die jeweils nächsten Meilensteine mit den zu erbringenden Leistungen in enger Zusammenarbeit zwischen Kunde und Lieferant zu spezifizieren und zu kalkulieren. Diese Art der Vertragsgestaltung wird ebenfalls maßgeblich von der Unternehmenskultur der Vertragspartner und ihrem Umgang miteinander bestimmt: Mut, Offenheit, Motivation und Vertrauen sind auch hier die Erfolgsfaktoren [4].

An dieser Stelle soll betont werden, dass unsere Vorstellung des PLM-Prototyping nicht von jeder Software, geschweige denn von der Unternehmensphilosophie aller PLM-Anbieter unterstützt wird. Denn das PLM-Prototyping setzt schon zu Projektbeginn auf eine funktionierende Gesamtanwendung auf. Völlig unabhängig davon, was im Lasten- oder Pflichtenheft beschrieben steht, wird die Software beispielsweise mit einem bestimmten Versionsmechanismus ausgeliefert, hat vordefinierte Rollen und Rechte für branchenübliche Benutzergruppen und kann Dokumente unterschiedlicher Herkunft mit unterschiedlichen Workflows verwalten. Anwendungen zur Verwaltung von Artikeln, Projekten oder Änderungen müssen nicht erst konfiguriert oder entwickelt werden, sondern stehen als Standardkomponenten zur Verfügung. Neben der Basisfunktionalität der verwendeten PLM-Technologie stellt deren Anpassbarkeit an die kundenspezifischen Anforderungen den entscheidenden Mehrwert dar. Somit steht die Adaptierung der ausgelieferten Software im Mittelpunkt der PLM-Einführung, wobei der Kunde am Beispiel der vorkonfigurierten Standards einen ersten Eindruck davon bekommt, was mit dem neuen System machbar ist. Auf dieser Grundlage kann er in enger Abstimmung mit seinem Lieferanten Schritt für Schritt entscheiden, wie seine künftige Lösung aussehen soll – und durch die nachfolgend am Scrum-Vorgehensmodell orientierten Schritte das PLM-Prototyping zugleich als internen Akzeptanzmotor nutzen.

Verzicht auf ein detailliertes Lasten-/Pflichtenheft: Die Features der zu entwickelnden Lösung werden beim PLM-Prototyping nicht vorweg in einem alle Realisierungsphasen umfassenden Lasten- beziehungsweise Pflichtenheft detailliert formuliert. Scrum liefert mit dem so genannten Product Backlog eine Vorlage für die zu entwickelnden Lösungen, wobei dieses Dokument kontinuierlich und

projektbegleitend erweitert, an aktuell geänderte Anforderungen angepasst und fortgeschrieben wird. Vom Kunden hochpriorisierte Features werden detaillierter beschrieben, also in wesentliche Funktionen mehr Aufwand investiert. Dies stellt sicher, dass Kernbestandteile oder Hauptfunktionalitäten vorrangig und innerhalb kurzer Zeit realisiert werden.

»Sprints« erleichtern die Priorisierung und Terminplanung: Die Entscheidung, welche Anforderungen aus dem Product Backlog umgesetzt werden, trifft das Kernteam des Kunden. »Sprints« im Sinne des PLM-Prototyping stellen Anpassungs- beziehungsweise Iterationsschleifen mit dem Ziel einer kontinuierlichen Verbesserung dar, wobei die für das Customizing Verantwortlichen die Anforderungen selbstorganisiert umsetzen. Ein

- hochgradig bedarfsgerechte Lösungen für die verschiedenen Zielgruppen im Unternehmen sowie
- schneller ROI (Return-On-Investment) durch die iterative Annäherung an eine PLM-Gesamtlösung.

Nicht zu unterschätzen ist auch, dass neue Vorgehensmodelle frischen Wind in die eigene Organisation bringen, die Arbeitsatmosphäre auflockern und zu hoher Motivation bei den Projektbeteiligten beitragen können. Besonders in wirtschaftlich unsicheren Zeiten wie heute stehen strategische IT-Projekte unter enormen Druck, mit noch weniger Zeit und Geld realisiert zu werden. Hier können innovative Methoden des Projektmanagements – wie das PLM-Prototyping – Abhilfe schaffen und die nötigen Argumente für die finanzielle

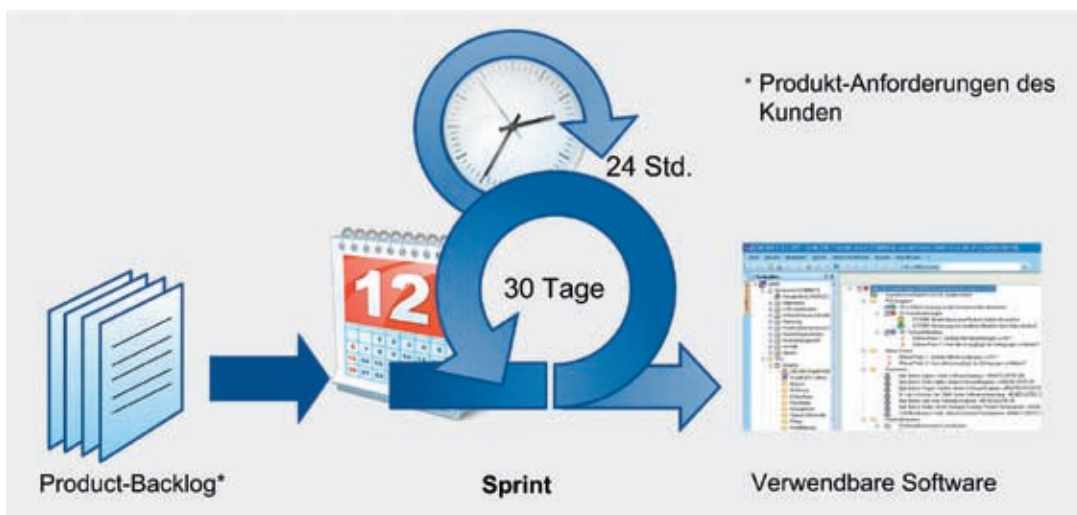


Bild 3: Vor jedem Sprint werden die Elemente im Product Backlog neu bewertet und priorisiert.

Sprint-Zyklus nach dem Scrum-Modell wird mit 30 Tagen veranschlagt und so oft wiederholt, bis die hochpriorisierten Funktionalitäten optimal ausgelegt sind. »Sprints« (Bild 3) zählen zu den zentralen Elementen im Scrum-Entwicklungszyklus und sind bei vielen modernen Software-Herstellern im Einsatz. Die bisherigen Praxiserfahrungen mit diesen in der Software-Branche schon bewährten Methoden im Kontext von PLM-Einführungsprojekten zeigen, dass das PLM-Prototyping – richtig aufgesetzt und gelebt – von Anwendungsunternehmen durchweg als positiv erfahren wird. Gründe dafür sind vor allem:

- großes Einsparpotenzial durch den Wegfall von aufwändigen Lasten-/Pflichtenheften und Feinspezifikationen,
- sehr schnell verfügbarer Prototyp der PLM-Anwendung,
- hohe Anwenderakzeptanz durch die frühzeitige Einbindung künftiger Systemnutzer,

wie auch personelle Ausstattung der Projekte liefern. Selbst in zähen, durch lange Spezifikationsphasen nicht voran kommenden oder bei »vor sich hin dümpelnden« Projekten kann PLM-Prototyping zu deutlich mehr Auftrieb verhelfen. Denn etwas Neues in festgefahrenen Strukturen auszuprobieren, kann oft wahre Wunder bewirken – mit handfesten Vorteilen für die Unternehmen, die bereit sind, sich darauf einzulassen. -fr-

Contact Software GmbH
www.contact-software.de

Quellenverzeichnis:

[1] The Agile Alliance: Manifesto for Agile Software Development. Utah 2/2001 (agilemanifesto.org/).
 [2] Ken Schwaber (Autor), Thomas Irlbeck (Übersetzer): Agiles Projektmanagement mit Scrum. Microsoft Press Deutschland, 2007, ISBN 978-3-86645-631-0.
 [3] Anna Erett / Rolf Stübbe: Der Wandel zur Innovationskultur. Zeitschrift Management & Qualität, 6/2008.
 [4] Peter Stevens: 10 Contracts for your next Agile Software Project. agilesoftwaredevelopment.com, 4/2009.