



□ Dr. Dirk Muthig

(dirk.muthig@lhsystems.com)

leitet das Production and System Design-Team im Profitcenter Airline Solutions der Lufthansa Systems AG. Seine Arbeit dreht sich rund um die Definition und Realisierung von geeigneten Systemarchitekturen, Standards und Richtlinien für Softwareprodukte, die von vielen Airlines weltweit genutzt werden.

Solide Produktweiterentwicklung im Windschatten agiler Vorausentwicklung

Die Entwicklung von Prototypen ist ein bekanntes Mittel, um Anforderungen besser zu verstehen oder mögliche technische Lösungen auf ihre Tauglichkeit zu erproben. Während ein virtueller Rundgang durch ein Gebäude relativ eindeutig vom fertigen Gebäude unterschieden werden kann, ist es bei IT-Produkten häufig schwieriger, diesen Unterschied zu erkennen. Dies liegt unter anderem daran, dass der Unterschied in vielen Details steckt, deren Komplexität das fertige IT-Produkt ja vor dem Anwender verbergen soll. Dieser Artikel beschreibt, wie man Prototyping nutzen kann, um agil und schnell zusammen mit den Nutzern die Lösungsfindung voranzutreiben, und dennoch sicherstellt, dass allen Beteiligten die verbliebene Komplexität einer tatsächlich einsetzbaren Lösung bewusst bleibt.

Wenn über neue Funktionen oder Verbesserung aus Nutzersicht nachgedacht wird, ist es heute üblich, die Anwender selbst direkt miteinzubeziehen. Typischerweise geht man hier nach Benutzergruppen vor und adressiert systematisch die wichtigsten Anwendungsfälle. Trotz einer solchen benutzerorientierten Herangehensweise bleibt es dennoch risikobehaftet, ohne Schulung operativ eingesetzte Produkte direkt zu verändern. Dies wird auch dadurch erschwert, dass die Organisation, die für alle relevanten Systemteile zuständig ist, insgesamt nicht die notwendige Agilität mitbringt (vgl. [Mut12]).

Um neue Lösungsansätze zu evaluieren, bieten sich demzufolge Prototypen an, die dem operativen System nahekommen, jedoch auf die zur Evaluierung relevanten Punkte fokussieren. Nach einer erfolgreichen Bewertung werden die neuen Features bzw. das neue Produkt in eine tatsächlich operativ einsetzbare Form überführt und durch den regulären Release-Management-Prozess ausgerollt.

Idealerweise bietet man den Nutzern einen Prototypen an, mit dem sie schon signifikante Teile der tatsächlichen Arbeit erledigen können. Nur dann ist eine praktische Bewertung vollwertig realisierbar, die sich deutlich von einem Test unter Laborbedingungen unterscheidet.

Prototypen lassen (bewusst) einige Punkte offen, die für ein Produkt unerlässlich sind.

Gelingt die Prototypentwicklung jedoch „zu gut“, wollen die an der Evaluierung beteiligten Nutzer und all ihre Kollegen den Prototyp vollwertig produktiv einsetzen. Dass Teile im Hintergrund tatsächlich nur provisorisch „zusammengeschustert“ wurden und in keiner Weise auf eine umfassende und dauerhafte Nutzung in der Breite ausgelegt sind, ist den Anwendern meistens nicht mehr plausibel zu erklären. Denn die neuralgischen Stellen sind häufig interne Komponenten oder Integrationselemente mit der umgebenden Systemlandschaft. Diese führen

in der Folge dann auch oft zu dauerhaften Problemen, da eine echte Korrektur im laufenden Betrieb nicht mehr realisierbar ist.

Die Komplexität eines Produkts hängt jedoch nicht nur mit der Systemarchitektur zusammen. Häufig sind es auch organisatorische Elemente oder das Produkt umgebende Prozesse, die rund um einen Prototyp noch nicht auf den Regelbetrieb vorbereitet sind.

Dabei kann es sich um fehlende Zustimmung von Gremien, wie Betriebsrat, Datenschutzbeauftragte oder Ergonomie-Ausschüsse, handeln. Diese sollen zwar schon früh in die geplanten Entwicklungen miteinbezogen werden, letztendlich aber auch über das finale Produkt befinden. Der ganze Prozess kann typischerweise erst so richtig Fahrt aufnehmen, wenn die grundsätzliche Lösung ermittelt wurde.

Auch systematische Integrations- und Systemtests sind abschließend im Gesamtverbund mit allen Teilen durchzuführen.

Dies passiert typischerweise im Rahmen des Release-Management-Prozesses. Dort werden die Versionen des Produkts freigegeben, die dann in einen Regelbetrieb übernommen werden können.

Für einen regulären Einsatz sind weitere Prozesse betroffen, zum Beispiel Bestell- und Support-Prozesse. Dort müssen die Produkteigenschaften und -teile bekannt sein, um erwartungsgemäß zu funktionieren. Auch hier sind Vorbereitungen zu treffen, die zwischen einer Pilotierung und dem Regelbetrieb notwendig sind.

Prototypen müssen von fertigen Produkten unterscheidbar sein

Ein Prototyp soll sich eben nur auf die zu untersuchenden Aspekte konzentrieren. Um dies effizient zu ermöglichen, ist die Wiederverwendung von vorhanden Komponenten ein probates Mittel (vgl. [KuM04]). Dadurch erreicht man schnell ein nahe an die Realität heranreichendes Produkt, behält aber eben jene Gefahr, dass der Prototyp kurzerhand zum operativen System erklärt wird. Um dies zu verhindern, muss man den Prototyp (ggf. künstlich) so gestalten, dass die Distanz zum richtigen Produkt jederzeit deutlich wird.

Eine Strategie besteht darin, den Prototypen zunächst als *zweite Lösung neben der eigentlichen Lösung* anzubieten. Auf die letztere Lösung muss zurückgegriffen werden, wenn bestimmte Anwendungsfälle nicht mit der neuen Lösung zu bearbeiten sind. Hier kann man bewusst auf die Unterstützung einiger Spezialfälle verzichten oder diese sogar ausblenden, um einen imperfekten Zustand explizit zu erreichen.

Häufig erschwert die *Beachtung aller Sicherheitsanforderungen und die Genehmigung geplanter Lösungen* die Realisierung einer schnellen, einfachen Lösung. Diese im Rahmen von Prototypen *zunächst beiseitezulassen*, kann daher vieles erleichtern. Da ein Prototyp jedoch praxisnah erprobt werden soll, ist hier häufig eine Sondergenehmigung zur expliziten Aufhebung einiger Sicherheitsaspekt für diesen Zweck nötig. In diesem Zuge kann man *die Rolle von Sicherheit in Bezug auf Aufwand* beim Kunden *explizit sichtbar platzieren*. Ein Hinweis für jeden Teilnehmer am Prototyping ist ebenfalls zu empfehlen, um diesen Aspekt bewusst zu machen.

Darüber hinaus besteht dann ja sowie so immer noch das Risiko, dass eine so ge-

fundene Lösung sich später aufgrund von Sicherheitsauflagen gar nicht wie geplant in der Praxis realisieren lässt. Dies wäre jedoch ein weiterer Baustein, um dem Kunden die Einflüsse auf mögliche Lösungen bewusst zu machen. Vorseilender „Gehorsam“ führt zwar oft zu Lösungen, die sicherheitstechnisch abgenommen werden, aber zu Unzufriedenheit bei den Nutzern bezüglich ungenügender Bedienbarkeit führen.

Ein weiterer Punkt ist das *Herabsetzen der Service Levels*. Dazu zählen üblicherweise Einschränkungen der Verfügbarkeit, längere Entstörungszeiten u. ä. Gründe hierfür sind die Kostenobergrenzen für Prototypen und die Vermeidung von Vorlaufzeiten für den Aufbau weiterer Betriebsinfrastruktur.

Ebenso sollen sich nicht alle Support-Mitarbeiter mit den Besonderheiten einer potenziell nur temporär eingesetzten Lösung auseinandersetzen und darauf geschult werden. Darüber hinaus kann eine kontrollierte Einschränkung dazu dienen, den Nutzern bewusst werden zu lassen, dass die Lösung noch nicht vergleichbar mit einem produktiven System ist. Dies muss natürlich durch entsprechende Kommunikation sichergestellt werden bzw. durch entsprechende Reaktionen und Hinweise des Prototypen, dass eine Funktionalität aus bestimmten Gründen oder Kontextfaktoren aktuell nicht verfügbar ist.

Dies kann durch eine *Bedienoberfläche ergänzt werden, die den unfertigen Zustand sichtbar macht*. Hier muss man natürlich darauf achten, dass man die Evaluierung nicht beeinträchtigt oder verfälscht. Da für die meisten Nutzer jedoch die Benutzerschnittstelle in ihrer Wahrnehmung das vollständige Produkt ist, ist es eben auch der Teil, der den prototypischen Charakter veranschaulichen muss.

Stilmittel können hierfür zum Beispiel, das Anzeigen von Systemwerten am Rande, die für den Benutzer nicht relevant sind (vergleichbar mit Messinstrumenten im Auto bei der Erprobung neuer Steuergeräte), kleinere, manuelle Schritte statt Automatisierung (manueller Konsistenzcheck mit OK-Button), usw. sein.

Darüber hinaus sind die Benutzer des Prototyps auch diejenigen, die die Bewertung durchführen; demzufolge ist auch *regelmäßig Feedback bezüglich Eindruck und Verbesserungsvorschlägen* durchzuführen, das man – wenn möglich – sogar direkt über den Prototyp realisieren kann.

Prototyping umfasst mehr als nur den Prototyp

Um zielgerichtetes Feedback zu bekommen, ist die Zielstellung der Erprobung vor der Erstellung eines Prototypen explizit festzuhalten und mit den Beteiligten abzustimmen. Die Festlegung von Erfolgskriterien und die Ermittlung der damit verbundenen Erwartungshaltungen ist essenziell, um sinnvolle Ergebnisse erzielen zu können, die Entscheidungsfindungen über Folgeaktivitäten direkt und klar unterstützen.

Neben der initialen Ermittlung und Abstimmung der Zielsetzung ist es notwendig, diese allen Beteiligten kontinuierlich in Erinnerung zu rufen, um zu den angestrebten Ergebnissen zu kommen. Dazu gehört insbesondere die Dokumentation der Zielsetzung, des Vorgehens und der getroffenen Entscheidungen bzw. Aspekte, die grundsätzlich noch offen sind, obwohl sie provisorisch zunächst auf eine bestimmte Art und Weise geschlossen wurden.

Letzteres ist von besonderer Bedeutung, wenn man nicht mit Provisorien dauerhaft leben will, die nur in Zwischenschritten entstehen, weil die richtige Lösung nicht im Rahmen von Prototyping-Aktivitäten realisiert werden soll. Im Gegenteil, denn der ermittelte Umfang und die Komplexität aller notwendigen Änderungen ist letztendlich ein wichtiges Ergebnis.

Änderungen bewegen sich irgendwo im Spektrum von „leicht und wenig Aufwand“ bis zu „komplex und viel Aufwand“. Die Abwägung zwischen notwendigen Änderungen und erreichbaren Verbesserungen für die Nutzer ist genau die Entscheidung, die nach dem Prototyping getroffen werden muss, um entsprechende Folgeaktivitäten zu beginnen.

Zu diesen gehört dann auch die Abarbeitung der Aktivitäten, die man im Rahmen des Prototyping nicht oder nur provisorisch erledigt hat. Diese müssen daher kontinuierlich in einem Backlog gesammelt werden, inklusive des assoziierten Aufwands (vgl. [Ste12]).

Prototyping kann agil im Voraus entwickeln, was danach wasserfallartig ins Produkt einfließt

Prototyping zielt darauf ab, neue Features oder Veränderungen zu evaluieren. Dazu ist eine enge und direkte Einbindung von Kunde und Nutzern notwendig, weshalb ein agiles Vorgehen der Natur der Sache am besten entspricht.

Wie weit man beim Prototyping gehen muss und wann eine Entscheidung über Folgeaktivitäten gut genug vorbereitet ist, kann nur der Kunde sagen, denn er muss diese Entscheidung treffen. Die Priorisierung der möglichen Schritte gehört direkt zu dieser Fragestellung dazu.

Der Kunde kommt dadurch früh mit in die Verantwortung und erlebt die Komplexität im Hintergrund. Er kann demzufolge besser überblicken, was die Realisierung einer doch so einfach erscheinenden Lösung erschwert und mithelfen an den richtigen Stellschrauben zu drehen. ■

Referenzen

[Mut12] Dirk Muthig: Agilität – eine Frage der Organisation, ObjektSpektrum, Online Themenspecial Agilität 2012

[KuM04] Marco Klemm, Dirk Muthig: Kostenoptimierung durch agile Methoden und Produktlinien, OBJEKTspektrum, Nr.5, 2004.

[Ste12] Chris Sterling: Managing Software Debt: Building for Inevitable Change, Addison-Wesley Professional, 2012
