

Projektsteuerung einer inkrementellen Systementwicklung bei gleichzeitiger globaler Produktnutzung

Jaroslav Blaha

BlaHa Executive Consulting GmbH
Am Klopferspitz 12
D-82152 Planegg
jbl@blexc.com

Abstract: Die Entwicklung eines zentralen, global genutzten IT-Systems (Transport Management System, TMS) für einen weltweit tätigen Logistik-Konzern erforderte die Segmentierung in fünf Inkremente. Um möglichst frühzeitig produktiven Nutzen zu bewirken wird jedes Inkrement nach Fertigstellung in Produktion gesetzt und Altsysteme entsprechend abgebaut. Die Parallelitäten zwischen der Spezifikation des übernächsten Inkrements, der Entwicklung des aktuellen Inkrements, sowie der Unterstützung und Adaption des jeweils produktiven Inkrements, erfordern ein angepasstes Projektvorgehen. Primäres Ziel und organisatorisches Muss ist die Beherrschung mehrerer paralleler Produktlebenszyklen durch ein geographisch verteiltes Projektteam.

1 Überblick

Die Entwicklung und Einführung eines globalen IT-Systems für die vollständige Abwicklung des Luft- und Seefracht Transport Managements eines der weltweit größten Logistikdienstleister kann aufgrund ihrer Größe, Komplexität und Gesamtdauer von ca. sieben Jahren nur in Inkrementen erfolgen.

Um schnellstmöglich Nutzen zu erreichen wird jedes Inkrement nach einem erfolgreichen Abnahmetest in Produktion gestellt. Während der Projektphase werden aus dem laufenden Betrieb Änderungsanforderungen gestellt. Diese kommen sowohl von den Nutzern, als auch durch legale Notwendigkeiten (z.B. Steueränderungen oder Zollvorschriften); vor Allem letztere müssen zwangsläufig und kurzfristig integriert werden. Jede dieser Änderungen muss allerdings auch parallel in der Entwicklung der nächsten Inkremente berücksichtigt werden, da es Abhängigkeiten zu neuer Funktionalität geben kann.

Insbesondere die Notwendigkeit in eine produktive, global einheitliche Software während der Einführung nationale Besonderheiten einer Vielzahl von Ländern zu integrieren, erfordert ein Team, das diesen multi-nationalen Ansatz widerspiegelt. Dadurch eröffnet sich neben dem temporalen Problem der Abstimmung von paralleler Inkrement-Entwicklung und Integration von Änderungen aus der produktiven Nutzung, auch noch eine geographische Dimension.

Ein wesentliches Merkmal der Projektphilosophie ist die Beherrschung von Anforderungsänderungen in drei parallel aktiven Projekt-/Produktbaselines: Der Produktionsversion (im Rechenzentrum in Deutschland), der aktuell in der Entwicklung oder Test befindlichen Version (beim Softwareentwickler in Deutschland, Indien und Polen), sowie des gerade spezifizierten Inkrements (in Singapur). Oberste Prämisse ist die Vermeidung von „Requirements Creep“. Mit einer Änderungsrate von ca. 2,6% über bis dato vier Jahre scheint dies gelungen.

Das System ist seit drei Jahren - mit zurzeit mehr als 4.200 Nutzern in 92 Ländern - in der produktiven Nutzung. Dabei sind drei von fünf geplanten Inkrementen fertiggestellt. Ermöglicht wurde dies durch ein Projektteam mit ca. 180 Mitgliedern mit 25 Nationalitäten in 12 global verteilten Standorten, sowie einer entsprechend angepassten Projektphilosophie, die den massiv parallelen, technischen und geographischen Randbedingungen genüge trägt.

2 Lösungsstrategien

2.1 Funktionale Separation

Bei der Identifikation des Funktionsumfanges bzw. -zuschnittes der Inkremente zu Projektbeginn wurde eine weitestgehend disjunkte Funktionsmenge für jedes Inkrement festgelegt. Jedes der fünf Inkremente des zukünftigen Systems befriedigt eine klar umgrenzte Menge an zusammengehöriger fachlicher Funktionalität (z.B. Luftfracht Import), die jeweils einen kommerziellen Nutzen für das Unternehmen generieren wird. Die logistischen Funktionen jeden Inkrements bilden zugleich, soweit möglich, die Organisationsstruktur des Unternehmens ab. Z.B. sind Luftfracht Import und Seefracht Import zumeist verschiedene Abteilungen in einer Niederlassung; dadurch können diese Abteilungen unabhängig voneinander in verschiedenen Projektphasen auf das neue System umgestellt werden. Dieses Vorgehen erfüllt zwei Ziele:

- Eine Organisationseinheit des Unternehmens muss nur einmal für die Einführung des Systems umgestellt werden. Neue Funktionen eines nachfolgenden Inkrements betreffen diese Organisationseinheit nur noch in geringem Maße. Dadurch reduzieren sich Aufwände für Schulung, Einführungsunterstützung, Datenmigration, Prozessanpassung etc. bei Folgeinkrementen auf Modifikationen und Weiterentwicklungen des „interessanten“ Funktionsumfangs für eine Organisationseinheit.

- Die Spezifikation und Entwicklung eines nachfolgenden Inkrements kann größtenteils unabhängig von der realen Nutzung des Systems erfolgen. Fundamentale Konzepte (z.B. Struktur und Terminologie einer Frachtsendung oder eines Transportweges) werden am Projektanfang definiert und müssen durchgängig genutzt werden; die Besonderheiten von z.B. Import vs. Export können jedoch unabhängig voneinander sequentiell entwickelt werden. Dies ist eine Vorbedingung für die Möglichkeit von bis zu drei parallelen Inkrementen in Spezifikation, Entwicklung bzw. Nutzung.

Funktionale Separation ist bei querschnittlichen Funktionen, wie z.B. den Buchhaltungs-/Abrechnungsmodulen, nur begrenzt möglich. Zusatzaufwände für Weiterentwicklungen (z.B. Datenmodellerweiterung und entsprechende Datenmigration) über alle Inkremente hinweg sind unvermeidbar.

Die Brauchbarkeit dieses Konzeptes wurde vor Allem dadurch nachgewiesen, dass nach Projektbeginn aus unternehmerischen Gründen die Reihenfolge der Inkremente umgestellt werden musste. Diese Änderung war für die Projektdauer und -kosten nahezu neutral.

Anmerkung: Zu Projektbeginn wurde diskutiert, ob anstatt eines inkrementellen auch ein agiler Ansatz sinnvoll wäre. Aufgrund der Größe und Komplexität der abzubildenden logistischen Prozesse wurde dies verneint. Der aktuelle Umfang jeden Inkrements wurde als die Mindestfunktionsmenge betrachtet, die jeweils für eine produktive Einführung (mit allen dazugehörigen organisatorischen Anpassungen in den weltweiten Geschäftsstellen) sinnvoll ist. Fachliche Änderungen zu einem produktiven Inkrement werden jedoch nach Bedarf in sehr kurzen Zyklen zwischen 72 Stunden (von Initiierung bis Produktivsetzung!) und wenigen Monaten agil eingeflochten.

2.2 Projektorganisation

Die Projektorganisation ist aufgrund der Parallelität zwischen Entwicklung und Produktion als „Zwitter“ ausgeprägt:

- Die Projektleitung in Deutschland stellt zugleich in Doppelrolle das Service-management inkl. der Verwaltung von getrennten Projekt- und Servicebudgets, sowie der Administration von Entwicklungs-, Wartungs- und Betriebsverträgen, sicher.

- Die Anforderungsdefinition beschäftigt sich schwerpunktmäßig mit der kompletten Spezifikation eines Inkrements en bloc. U.a. um die Mitarbeiter dieses Teams von den täglichen Einflüssen aus dem Betrieb (z.B. Incidentanalyse, Klärung von Anforderungen vs. Defekte) zu schützen wurde es in Singapur angesiedelt. Ein kleiner Teil des Teams mit Sitz in Deutschland wird vorwiegend mit der Spezifikation von Änderungen (Changes) ggü. der aktuellen Anforderungsbaseline eingesetzt. Dieses Teilteam fängt die zwangsläufig entstehende Volatilität und Dynamik aus dem Produktivbetrieb auf, isoliert das Singapur-Team von den dadurch entstehenden „Rüstkosten“ und ermöglicht folglich konzentriertes Arbeiten abseits vom Produktionsalltag. Allerdings ermöglichen das Team in Singapur (Deutschland +6h) und ein kleineres Team in New York (Deutschland -6h) in Notfallsituationen aus der Produktion rund um die Uhr Expertise zu fachlichen Anforderungen einzuholen.
- Die Entwicklung ist vollständig an externe Lieferanten ausgelagert. Auch hier gibt es die Trennung zwischen einem Kernteam für die Entwicklung des nächsten Inkrements und einem Wartungsteam, das Fehlerbehebung und Weiterentwicklung der produktiven Software durchführt. Die wesentliche Schwierigkeit besteht darin mehrere parallele Software-/Releasebaselines zu verwalten und unter Konfigurationskontrolle (inkl. „Branching“ und „Merging“) zu halten. Die dazugehörigen Methoden und Werkzeuge sind im Software Engineering etabliert und weitestgehend problemlos.
- Ein festes Testteam führt sowohl die Abnahmetests für das nächste Inkrement, als auch Regressionstests für Wartungsreleases der produktiven Software durch. In diesem Team wird eine Trennung in zwei Gruppen vermieden. Stattdessen wird dynamisch und mit Hilfe eines Test Management Tools und einer vorhandenen Menge an Testfällen und –skripten der jeweilige Fokus des Gesamtteams abhängig von der Projektphase zwischen Abnahme- und Regressionstests verschoben. Neben dem zentralen Team existieren nationale Testteams, die aus Sicht der fachlichen Anforderung ihres Landes ebenfalls Regressions- und Abnahmetests („User Acceptance Test“) durchführen. Alle Testergebnisse werden im gleichen Test Management Tool zusammengeführt, um einen ganzheitlichen Wissenspool zu erhalten.
- Das Deploymentteam ist für die Einführung der Software (z.B. durch Prozessanpassung und –beratung, initiale und fortlaufende Schulung) in den Zielländern zuständig. Um den lokalen Gegebenheiten Rechnung zu tragen ist das Team auf die drei Regionen Europa, Asien, Nordamerika verteilt, um dort mit lokaler Fachexpertise und Sprachkenntnis zu unterstützen. Dieses Team trägt die Hauptlast des Zielkonflikts zwischen Produktivnutzung und Weiterentwicklung. Da das Team sowohl die fachliche Nutzung der produktiven Software betreuen muss, als auch zeitgleich die Einführung eines neuen Inkrements unter Berücksichtigung der angesammelten Erfahrungen und organisatorischen Anpassungen in den Zielländern vorbereitet und begleitet, ist eine hohe Belastung der Mitglieder unvermeidbar.
- Release Management: Siehe Kapitel 2.4

Eine der wesentlichen Herausforderungen ist es allen Mitarbeitern in allen Teams in allen Projektstandorten das Bewusstsein für die jeweilige Projektphase zu vermitteln. Diese erfolgen in Wellen: Ein Inkrementzyklus dauert ca. 1,5 Jahre (ca. sechs Monate Spezifikation und ca. ein Jahr Entwicklung, Test, Schulung, Einführung). Dadurch ist jedes Team (außer der Anforderungsdefinition) ca. sechs Monate pro Jahr schwerpunktmäßig mit der Einführung eines neuen Inkrementes beschäftigt, während in der Restzeit die Betreuung des Produktivsystems intensiviert wird.

2.3 System- und Softwarearchitektur

Da jedes Inkrement ca. 1,5 Jahre von Spezifikation (ca. 6 Monate) bis Produktionseinführung benötigt (und die Software damit ca. 2 Jahre Lebensdauer hat bis sie vom nächsten Inkrement abgelöst wird), kann es aus bis zu ca. 30 Softwarereleases bestehen:

- Mehrere Releases, die Abnahmetests bis zur Produktionsreife durchlaufen.
- Reguläre, periodische Releases für die Produktion, die fachliche Änderungen (Changes) implementieren.
- Hot fixes, die kritische Defekte oder Changes in der Produktion beheben. Z.B. kann die Zollverwaltung Vorschriften ändern, die innerhalb von 30 Tagen in einem produktiven TMS umgesetzt werden müssen.

Ein vordefiniertes „Staging Model“ beschreibt, wie neue Releases sequentiell zwischen Entwicklungs-, Test-, Schulungs- und Produktivsystemen verteilt werden.

Rechenzentrumsseitig existiert nur eine (massive redundante) Systeminstanz für die globale, produktive Nutzung. Daneben existieren mehrere (je nach Phase bis zu zwölf) ebenfalls zentrale Systeminstanzen für Entwicklung, Abnahmetests, Lasttests, sowie Schulung. Die Inbetriebnahme eines neuen Softwarereleases erfordert folglich nur die einmalige Installation der Software und ggf. eine dazugehörige Datenmigration pro Instanz. Der fundamentale Nachteil der globalen Nichtverfügbarkeit des Produktivsystems während der Installation wird dadurch aufgewogen, dass alle Nutzer weltweit immer auf einem identischen Softwarestand arbeiten. Unter anderem

- weiß die Supportorganisation immer auf welchen Release sich Nutzerprobleme beziehen und muss jeweils nur Wissen über den aktuellen Release vorhalten;
- da die Nutzerorganisation geographisch verteilt ist, um z.B. regionale Sprachen abzudecken bzw. aus globaler Sicht einen „Follow-the-Sun“ Service anzubieten, kann jede Servicestelle jedem Nutzer weltweit gleichermaßen helfen;
- können langlaufende Transaktionen in Produktion nie mehrere Releases mit potentiell verschiedenen Datenmodellen überspannen;
- müssen nur Testsysteme für zwei Releases (das produktive und das nachfolgende) betreiben werden;
- ist die Behebung von Software- oder Konfigurationsdefekten mit einer zentralen Aktion sofort global wirksam.

Aufgrund des notwendigen Funktionsumfangs ist die Verwendung eines „Thin-Client“ (z.B. browserbasiert) nutzerseitig nicht möglich. Stattdessen wird über den Java WebStart Mechanismus bei jedem neuen Release die jeweils aktuelle Clientsoftware automatisch auf den PC des Nutzers übertragen. Während dies technologisch leicht zu implementieren ist und den Nutzer, sowie den lokalen Support, von der Notwendigkeit jeglicher Installation befreit, gibt es einen Nachteil. In den Stunden nach einem Releasewechsel laden alle Client-PCs bei Starten der Anwendung durch den Nutzer die neue Clientsoftware. Die dadurch entstehende Lastspitze auf dem Kommunikationsnetzwerk muss bei der Systemeinführung in einer Geschäftsstelle antizipiert und ggf. durch Bandbreitenanpassung kompensiert werden.

2.4 Release Management Organisation

Der Aufbau eines separaten, vom Projekt unabhängigen, Produkt- oder Release-managements in der Linienorganisation erschien unmöglich, da die oben beschriebenen Abhängigkeiten und die Komplexität des Staging Modells von zwei parallelen Organisationen nur unvollständig kontrolliert werden könnten. Deshalb umfasst die Projektorganisation auch das vollständige Management aller Software Releases (Produktiv, Test, Entwicklung) und das Lebenszyklusmanagement, inkl. der dazugehörigen ITIL Prozesse für den Betrieb des Rechenzentrums. Parallel zur Verschiebung des Schwerpunkts von verbleibender Entwicklung zu produktiven Anteilen wird die Struktur des Projektes systematisch und schrittweise hin zu einer Linienorganisation umgebaut. Ziel ist es am Ende des Projektes, mit den gleichen Mitarbeitern, ein Produktmanagement innerhalb der existierenden Aufbauorganisation des Konzerns zu etablieren.

Insbesondere kann dadurch den Mitarbeitern bereits während der Projektphase ein Übergang in Linienfunktionen und dadurch eine Karriereplanung über das Projektende hinaus angeboten werden. Dies stellte sich für die Motivation und Teamstabilität als förderlich heraus.

2.5 Prozesse und Werkzeuge

Für die wichtigsten Prozesse (Change Management, Incident Management, Problem/Defect Management) werden sowohl für die Projektphase als auch für die Produktionsbetreuung die gleichen Werkzeuge und Datenbestände verwendet. Changes, Incidents, Problems werden unabhängig von Ihrer Quelle (Projekt vs. Produkt) in den gleichen Datenbankinstanzen der Tools verwaltet. Die Quellangabe erfolgt durch Verweis auf die Releasenummer. Die Zuordnung von Einträgen (z.B. Defekte) zu Produktivnutzung (z.B. für Garantiesprüche) vs. Projektentwicklung (z.B. zur Bewertung der Abnahmewürdigkeit ggü. einem Lieferanten) erfolgt durch einfache Filterung.

Die Komplexität erhöht sich dadurch, dass sich Einträge inhaltlich auf mehrere Releases und Inkremente beziehen können. Wird z.B. ein Defekt in einer produktiven Softwareversion (aus einem Incident heraus) entdeckt, so ist es wahrscheinlich, dass der gleiche Defekt auch im aktuell in Entwicklung befindlichen Softwarerelease auftritt; dort muss er aber separat bewertet, behoben und nachgetestet werden. Die Verwaltung identischer Einträge für verschiedene Releases mit verschiedenen Bearbeitungszyklen wird von den untersuchten und letztlich ausgewählten Tools nicht unterstützt. Als Behelf werden für solche Einträge Kopien pro Release/Inkrement erstellt und unabhängig verwaltet; die logische Zusammengehörigkeit ergibt sich durch Querverweise auf z.B. die Defektnummern in den Kopien.

Generell sehen die Prozesse verschiedene Bearbeitungspfade vor: Defekte, Incidents, Changes

- die aus dem produktiven Release stammen, müssen in den meisten Fällen kopiert und ebenfalls für in Entwicklung befindliche Folge-Releases initiiert werden;
- die aus einem im Abnahmetest befindlichen Release stammen sind zumeist singular. Zumindest bei schwerwiegenden Defekte wird zusätzlich geprüft, ob sie nicht auch (bis dato unentdeckt) in der Produktionssoftware auftreten;
- können zudem Auswirkungen auf das übernächste, in der fachlichen Spezifikation befindliche Inkrement haben.

Die Prozesse und Datenstrukturen in den Tools wurden darauf ausgelegt, dass im Extremfall z.B. ein Incident aus dem Produktivsystem zu einem Defekt in zwei Releases (in Produktion = V.x, in Abnahmetest= V.x+1) führt, einen Change (z.B. zur Verbesserung der Ausnahmebehandlung) gegen das in Spezifikation befindliche Inkrement (V.x+2) motiviert, eine vertragliche Garantieleistung ggü. V.x, und eine vertragliche Abnahmeverhinderung ggü. V.x+1 auslöst.

Durch die Bündelung aller Informationen zu allen Inkrementen/Releases in jeweils einer Toolinstanz ergibt sich ein Wissenspool der insbesondere für den Support extrem hilfreich ist. Aktuell wurden mit diesem Ansatz ca. fachliche 650 Changes, 3.000 Incidents und 11.000 Defekte über ca. 80 Softwarereleases miteinander verwoben.

3 Zusammenfassung

Angepasste Mechanismen, Strukturen, Werkzeuge und die Berücksichtigung kritischer Erfolgsfaktoren ermöglichen es ein komplexes, geographisch verteiltes Projekt und dessen Organisation über mehrere Jahre von der reinen Systementwicklung, über eine hybride Phase aus Projekt- und Produktmanagement, hin zu einer Linienorganisation zu überführen.

Entscheidend ist es diese Parameter bei der initialen Projektkonfiguration technologisch, organisatorisch und auch budgetär vorzusehen.

Literaturverzeichnis

- [Jws10] Java(TM) WebStart Guide: <http://download.oracle.com/javase/6/docs/technotes/guides/javaws/developersguide/contents.html>.
- [Of07] Office of Government Commerce (ed.): ITIL: The Official Introduction to the ITIL Service Lifecycle, London 2007.