

Vorgehensmodell in Softwareprojekten





Inhaltsverzeichnis

WARUM EIN NEARSHORE VORGEHENSMODELL?	3
KOMPONENTEN DES EBS VORGEHENSMODELLS	4
1. DAS GENERISCHE PHASENMODELL	5
2. DER ITERATIVE UND INKREMENTELLE CHARAKTER	5
3. ANALYSE UND ERSTELLUNG DER SPEZIFIKATION	6
4. PROJEKTMANAGEMENT	7
5. RISIKO MANAGEMENT	7
6. ENTWICKLUNGSAKTIVITÄT	8
7. QUALITÄTSSICHERUNG	8
8. ROLLENMODELL	9
9. KONFIGURATIONS MANAGEMENT	9
10. VERSIONSMANAGEMENT	9



Warum ein nearshore Vorgehensmodell?

Das vorliegende Dokument stellt eine Kurzfassung in deutscher Sprache des EBS-Vorgehensmodells für Software-Entwicklungsprojekte dar. Das Originaldokument in rumänischer Sprache ist wesentlich umfangreicher. Es dient einer genauen Beschreibung des gesamten Entwicklungsprozesses.

Zweck dieses Dokumentes ist es, dem Leser einen Überblick über die wichtigsten Aspekte und Prinzipien der Vorgehensweise von EBS zu liefern.

Den Ausgangspunkt für die Entwicklung des Modells bildeten diverse Softwareprojekte von EBS einerseits für den rumänischen Markt und andererseits für Kunden im Nearshore/Offshore-Bereich. Es waren sowohl Standard- als auch kundenspezifische Projekte. Weitere Grundlage waren theoretische Ansätze wie sie in der Literatur zu finden sind (Microsoft Solutions Framework, IBM Rational Unified Process, Agile Vorgehensweisen wie z.B. Scrum). Das EBS Modell entstand über einer Zeitspanne von mehr als 8 Jahren. Entsprechend konkreter Projekterfahrungen wurde es immer weiter entwickelt. Heute werden sowohl kundenspezifische Projekte als auch Standardprojekte von ihm abgedeckt; beide können mit seiner Hilfe organisiert und geleitet werden.

Der Trend der letzten Jahre ging in der Softwareentwicklung in Richtung agiler Methoden. Das veraltete Wasserfallmodell wurde damit verabschiedet. Das Bestreben ist heute Projekte iterativ und inkrementell zu entwickeln und den Kunden stärker in die Projekte ein zu binden. Auch will man im Verlaufe des Projektes Gelegenheit haben, Verbesserungen einzuführen und neue Iterationen bis zu dem Moment zu planen, in dem das Produkt alle Vorgaben erfüllt. Aufwände können nämlich nicht immer genau geschätzt, Fertigstellungstermine langfristig nicht genau bestimmt werden. Die oft später auftauchenden Change Requests können zwar durch neue Iterationen (und Verschiebung des Endtermins) abgedeckt werden. Das steht allerdings im Gegensatz zu den Vorstellungen der meisten Kunden. Sie erwarten eine genaue Planung des Projektes schon im Vorfeld, wollen Risiken minimieren und wünschen Überschaubarkeit und Kontrolle der Kosten.

So kommt in der Praxis öfter der Fall vor, bei dem Änderungen während der Projektlaufzeit gemacht, neue Anforderungen hinzugefügt oder existierende Anforderungen geändert werden müssen. Das EBS Vorgehensmodell versucht solche Aspekte zu berücksichtigen; so bestimmt es eine Reihe von Grundregeln und Prinzipien für den Entwicklungsprozess. Sie bieten sowohl Flexibilität in der Projektgestaltung als auch Sicherheit und Überschaubarkeit auf Kundenseite.

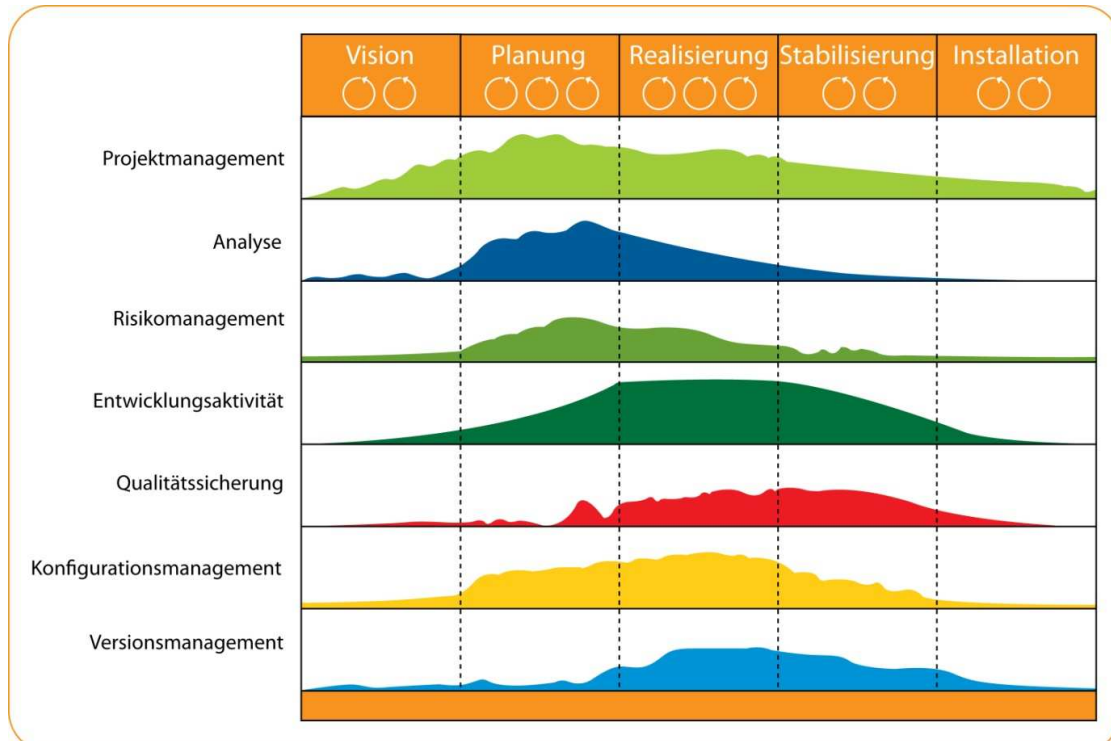
Das Vorgehensmodell in seiner kompletten Form unterstützt besonders gut Lifecycle-Projekte. Also Projekte, welche als Ziel die Realisierung von Softwareprodukten dergestalt haben, dass ein Produkt komplett und von Anfang an nach Kundenwunsch entwickelt und im Rahmen des Projektes auch produktiv



einsatzfähig werden soll. Es sind die Projekte mit der größten Komplexität in ihrem Ablauf. Jede andere Art von Projekt kann als Sonderfall des kompletten Modells betrachtet und behandelt werden.

Komponenten des EBS Vorgehensmodells

Eines der Hauptziele bei der Entwicklung des EBS-Vorgehensmodells war die Gestaltung des Inhaltes derart, dass während der Durchführung eines Projektes eine gute Übersicht sichergestellt ist sowie die einzelnen Hauptelemente und funktionalen Zusammenhänge möglichst einfach erkannt und verstanden werden können. Das gesamte Vorgehensmodell sollte dabei keinen monolytischen Charakter aufweisen. Sondern es sollte aus logischen Bausteinen bestehen, die sich an die spezifischen Aspekte und Einschränkungen eines Projektes anpassen lassen. Das Vorgehensmodell basiert daher insbesondere auf einer Reihe von Prinzipien, welche sowohl aus agilen als auch aus formalisierten Vorgehensmodellen entnommen wurden. Sie bilden zusammen ein flexibles Framework, das sowohl für kleine als auch für große Projekte angepasst werden kann. Die einzelnen Bausteine haben vergleichbare Funktion in jedem Projekt, jedoch je nach Art und Größe des Projektes können diese in unterschiedlichem Verhältnis zueinander stehen.





1. Das generische Phasenmodell

Grundlage für den gesamten Aufbau des Vorgehensmodells ist ein generisches Phasenmodell, das die wichtigsten Etappen in der Entstehung eines Softwareproduktes definiert und diese mit entsprechenden Beschreibungen und Empfehlungen versieht, durch die sie bei der Erstellung des Projektplans verwendbar werden. Die meisten Elemente, die im Rahmen des generischen Phasenmodells definiert werden, können ohne weiteres als Meilensteine in den Projektplan übernommen werden. Das generische Phasenmodell bildet die Grundlage für den Projektplan. Das Modell stellt für den Projektleiter auch Bedingungen für den Abschluss bzw. Beginn jeder Phase zur Verfügung, sowie dafür wie weit das Produkt nach jeder abgeschlossenen Phase sein muss. Somit werden quantitative und qualitative Metriken zur Verfügung gestellt, die in den einzelnen Projekten als Indikatoren für den Fortschritt, Planeinhaltung und korrekten Ablauf des Projektes verwendet werden können.

Das generische Phasenmodell definiert die 5 Phasen Vision, Planung, Entwicklung, Stabilisierung und Installation, die den gesamten Lebenszyklus eines Softwareproduktes von der Bedarfsfeststellung bis zum Produktivbetrieb beschreiben.

Im konkreten Einzelfall wird in Abstimmung mit dem Kunden festgelegt, welche der Indikatoren und Metriken in dem Projekt zu verwenden sind. Anhand dieser im Voraus bestimmten Indikatoren erlaubt das Vorgehensmodell eine bessere Transparenz und eine leichtere Überwachung des Projektverlaufes.

2. Der iterative und inkrementelle Charakter

In der aktuellen Literatur, die Projektmanagement zum Gegenstand hat, wird das veraltete Wasserfallmodell nicht mehr empfohlen, ja sogar als Gegenbeispiel zu den modernen Methoden präsentiert. Eigene konkrete Projekterfahrungen bestätigen das. Die heutigen Bedürfnisse der Projekte machen es notwendig, sehr viel Flexibilität in die Projektgestaltung einzubauen um sehr leichte Möglichkeiten zu haben, auf Änderungen zu reagieren und diese mit minimalem Zusatzaufwand in ein Projekt zu übernehmen. Es ist in der Praxis oft der Fall, dass die konkreten Projektbedingungen es nicht erlauben eine detaillierte Planung zu erstellen und diesen auch genau einzuhalten, da während des Projektes immer wieder die Notwendigkeit entsteht unterschiedliche Einschränkungen oder Anforderungen zu berücksichtigen.

Diese Notwendigkeit führte im EBS Vorgehensmodell zur Einführung des iterativen und inkrementellen Elements, das seine Ursprünge in den Agilen Methoden hat. Das wird bereits im Projektplan und in den Aufwandschätzungen deutlich. Die Realisierung einzelner Komponente wird nicht in einem einzigen Anlauf geplant, sondern abhängig vom erwarteten Arbeitsvolumen in mindestens 2 Etappen. Das gesamte Projekt wird in solchen Etappen (Sprints) organisiert und innerhalb jedes Sprints wird die gleiche (oder eine ähnliche) Reihe von Aktivitäten durchgeführt. Ziele der Sprints werden zusammen mit dem Kunden abgestimmt. Innerhalb eines Sprints werden keine Änderungen der Anforderungen zugelassen. Die



Ergebnisse jedes Sprints werden mit dem gesamten Team und unter Beteiligung des Kunden analysiert und bewertet. Bei dieser Gelegenheit können auch Änderungen am Projekt(plan) beschlossen werden. Risiken werden zusammen mit dem Kunden neu analysiert, der Verlauf und Inhalt der folgenden Etappen (Sprints) kann bestimmt werden usw.. Die Sprints können je nach Projekt unterschiedliche Dauer haben, jedoch haben sich 4-Wochen in der Praxis bewährt. Sprint Review Meetings sind dafür geeignet den Projektstatus zu bewerten, den aktuellen Projektstand dem Kunden zu präsentieren, den bisherigen Verlauf im Rahmen des Sprints unter der Lupe zu nehmen und mögliche Verbesserungsmaßnahmen zu ergreifen. Der Kunde hat somit die Möglichkeit in das Projekt einzugreifen und Änderungen vorzuschlagen bzw. vorzunehmen, das führt über die gesamte Laufzeit des Projektes zu einem kontinuierlichen Verbesserungsprozess, der auch von Kundenseite mitgestaltet werden kann.

3. Analyse und Erstellung der Spezifikation

Innerhalb des EBS Vorgehensmodells spielt die Analyseaktivität eine bedeutende Rolle, insbesondere in den ersten beiden Phasen, in denen die Erwartungen und Notwendigkeiten des Kunden erfasst und in entsprechenden Dokumenten schriftlich festgelegt werden müssen.

Das EBS Vorgehensmodell legt fest, dass Anforderungen des Systems immer schriftlich zu erfassen sind. Während der gesamten Projektlaufzeit werden diese Dokumente permanent ergänzt, so dass sie den jeweils aktuellen Stand der Kundenanforderungen reflektieren. Für diese Art von Tätigkeit sind spezielle Rollen innerhalb des Rollenmodells vorgesehen, die für Analyse zuständig sind.

Die Dokumente, die als Ergebnis der Analyse entstehen, werden vom Kunden abgenommen. Damit bestätigt er, dass diese die Anforderungen für das System entsprechend wiedergeben. Somit sind die Analysedokumente die ersten Ergebnisse, die in den Entwicklungsprozess eingehen. Sie bilden den eigentlichen Input für diesen Prozess und werden als Ausgangspunkt für die Schätzungen, Testpläne und den Projektplan verwendet.

Innerhalb des EBS Vorgehensmodells werden 2 Typen von Dokumente definiert, die für die komplette Dokumentation eines Softwareprodukts hinreichend sind. Zunächst wird ein Fachkonzept erstellt, das die wesentlichen fachlichen und funktionalen Merkmale des Systems festlegt. Die Ergebnisse der ersten Gespräche mit dem Kunden sind schon Bestandteil des Fachkonzepts, noch in einer groben Form. Der Inhalt der Dokumente wird dann in der nächsten Phase (Planung) verfeinert. Das geschieht wieder in Zusammenarbeit mit dem Kunden; die Fortführung der Dokumentation wird jetzt von Fachleuten übernommen. Die Dokumente beinhalten weiterhin hauptsächlich funktionale und fachliche Details, es werden keine oder nur wenige technische Aspekte dokumentiert. Ziel ist es die Notwendigkeiten und Anforderungen des Kunden zu dokumentieren und nicht die Lösung, die dafür entwickelt wird.

Die zweite Dokumentation ist die technische Spezifikation. Im Unterschied zum Fachkonzept wird in diesen Dokumenten die Lösung (das System) beschrieben die während des Projektes erstellt werden soll. Die



Dokumente beinhalten alle technischen Details, die notwendig sind um das System zu entwickeln. Es sind ausgeprägt technische Papiere. Modellierung mithilfe von UML Diagramme ist keine Voraussetzung, ist aber in den meisten Fällen Bestandteil der Dokumentation. Die Spezifikation beinhaltet (je nach Notwendigkeit) sowohl eine Architekturbeschreibung als auch Definition von Performance Parametern, Beschreibung der Anwenderoberflächen, Schnittstellen mit Fremdsystemen, auf Portabilität bezogene Aspekte, Datenbankmodelle usw.. Diese Dokumentation wird von Fachkräften erstellt und benötigt nur sehr wenig Unterstützung von Seite des Kunden, da es im Wesentlichen aufgrund des Fachkonzeptes erstellt wird.

Die beiden Dokumenttypen werden nach einer vorgegebenen Struktur erstellt, kontinuierlich über die gesamte Projektdauer gepflegt und decken die Notwendigkeiten eines Software Projektes vollständig ab.

4. Projektmanagement

Dem Projektleiter kommt eine zentrale Rolle im EBS Vorgehensmodell zu. Er organisiert als (interne) Hauptaufgabe den gesamten Entwicklungsprozess. Jede Aktivität in jedem Bestandteil des Vorgehensmodells wird durch den Projektleiter geplant und genehmigt. Er behält die Übersicht über das Projektgeschehen und steuert den Verlauf. In den meisten Fällen ist er auch erster Ansprechpartner für den Kunden.

Für größere Projekte sieht das Modell vor, dass der Projektleiter sich ausschließlich auf organisatorische Aufgaben konzentriert. Je nach Komplexität und Größe eines Projekts kann der Projektleiter aber auch Aufgaben aus anderen Bereichen übernehmen. So kommt es relativ oft vor, dass er in kleineren Projekten zusätzlich technische Aufgaben übernimmt oder das Projekt aus technischer Sicht leitet.

Der Projektleiter ist dafür zuständig die einzelnen Iterationen zu definieren und zu überwachen sowie Change Requests einzuleiten. Er ist Moderator für die Sprint Reviews. Das Vorgehensmodell definiert Templates für Dokumente (Protokolle, Statusreports usw.), die regelmäßig an den Kunden gesendet werden.

5. Risiko Management

Insbesondere bei Projekten, die Offshore durchgeführt werden, sind zusätzliche Risiken vorhanden. Das Vorgehensmodell definiert eine effiziente Methode um diese Risiken entsprechend zu handhaben. Risikomanagement ist ein kontinuierlicher Prozess, der in Zusammenarbeit mit dem Kunden über die gesamte Dauer des Projektes erfolgt. Die Praxis hat gezeigt, dass die beiden ersten Phasen eines Projektes ausschlaggebend sind für ein erfolgreiches Risikomanagement. Die wichtigsten Risiken sind dort schon zu erkennen und sollten auch bereits dort behandelt werden.

Einer der wichtigen Meilensteine des generischen Phasenmodells liegt am Ende der zweiten Phase (Planung). Ein Hauptziel ist die Identifizierung, Kontrolle und Reduzierung der wichtigsten bekannten



Risiken, obwohl in dieser Phase bereits sehr viele unterschiedliche Aktivitäten laufen und vergleichsweise viele wichtige Ergebnisse erzeugt werden (Fachkonzept, Spezifikation, Projektplan). Dementsprechend verläuft das Risikomanagement in dieser Phase mit maximaler Intensität.

Das Risikomanagement setzt folgende 3 Schritte voraus:

Identifizierung der Risiken

Analyse und Bewertung der Risiken

Maßnahmen zur Risikobeseitigung

Das Vorgehensmodell definiert ein Template, mit dessen Hilfe solche Informationen festgehalten werden können. Risiken werden erfasst, zusammen mit dem Kunden analysiert und bewertet. Entsprechende Maßnahmen werden zusammen getroffen. Der iterative Charakter des Projektverlaufes bietet eine gute Möglichkeit dieses Verfahren periodisch über die gesamte Projektdauer zu wiederholen, so wie es das Risikomanagement vorschreibt.

6. Entwicklungsaktivität

Unabhängig der Notwendigkeit zur Formalisierung bleibt der Aufwand für die Entwicklungsaktivitäten der Bedeutendste in den allermeisten Entwicklungsprojekten. Diverse Erfahrungswerte konnten hier im Laufe der Zeit in der Praxis ermittelt werden, wurden zusammengefasst und als Richtlinie für die Programmierung in das EBS Vorgehensmodell eingefügt. Code Richtlinien, Best Practices, Architekturmodell, wiederverwendbare Komponenten und Frameworks stehen zur Verfügung und können sofort in neuen Projekten verwendet werden.

7. Qualitätssicherung

Qualitätssicherung ist ein Pflichtbaustein des EBS Vorgehensmodells. Entwicklungsprozesse werden grundsätzlich nur mit paralleler, begleitender Qualitätssicherung organisiert. Der QS-Prozess beginnt schon in der ersten Phase des Projektes und endet erst, nachdem das Produkt die Abnahmetests bestanden hat. Die Planung des gesamten Qualitätssicherungsprozesses ist Bestandteil des Projektplanes. Je nach Komplexität des Projektes und Umfang des QS-Teams werden funktionale Tests, Integrationstests, fachliche Tests und Performance Tests mit Hilfe von Testszenarien und Testfällen geplant und beschrieben. Der gesamte Prozess wird dokumentiert und durch Tools (Webbasierte Error Tracking Systeme) unterstützt, wodurch alle Informationen, die aus dem Prozess resultieren, sowohl dem Kunden als auch den Teammitgliedern zur Verfügung stehen. Testprotokolle werden regelmäßig erstellt. Entsprechend dem iterativen und inkrementellen Charakter des Projektes werden die Ergebnisse des QS-Prozesses während der Sprint Review analysiert und bewertet.



8. Rollenmodell

Wie in den meisten Vorgehensmodellen werden auch hier die einzelnen Aktivitäten, die in dem Projekt vorkommen, Rollen zugeordnet. Die Rolleninhaber sind dann für die Durchführung verantwortlich. Das Rollenmodell ist spezifisch durch die Notwendigkeiten jedes Projektes geprägt. Das Vorgehensmodell definiert 4 große Gruppen, in denen die einzelnen Rollen gruppiert werden..

9. Konfigurationsmanagement

Insbesondere bei Projekten, in denen größere Teams über längere Zeit tätig sind, erweist sich die Notwendigkeit einer genauen Verwaltung der Konfigurationen, die ein Softwaresystem zu jedem Zeitpunkt des Entwicklungsprozesses hat. Das Konfigurationsmanagement verläuft im engen Zusammenhang mit der Entwicklungsaktivität und mit der Qualitätssicherung. Das Vorgehensmodell empfiehlt eine Reihe von Tools, mit denen man diesen Prozess effektiv organisieren kann. Im gleichen Zusammenhang sind noch weitere Methoden (z.B. Nightly Build) zur Überprüfung und Konsolidierung der jeweils aktuellen Konfiguration empfohlen.

10. Versionsmanagement

Versionsmanagement ist von Bedeutung insbesondere bei den Projekten, bei denen ein Produkt in Produktion gegeben wird. Das EBS Vorgehensmodell bestimmt 3 unterschiedliche Arten von Versionen, die in einem Projekt verwaltet werden können. Es handelt sich um interne Versionen, die im Projektplan vorgesehen werden und im Verlaufe der Entwicklungsphase erreicht werden, konfigurationsbezogene Versionen und auf den Entwicklungsprozess bezogene Versionen.

Für die Konfiguration werden innerhalb des Modells die Versionen Alpha, Beta, Release Candidate und Release To Manufacturer definiert. Diese werden schon frühzeitig geplant und erscheinen als Meilensteine im Projekt.

Für den Entwicklungsprozess gibt es zusätzlich 3 weitere Meilensteine, die zwar nicht unbedingt geplant werden, aber zu einem bestimmten Zeitpunkt erreicht werden. Es sind Code Complete, Freeze und Master. Für diese Versionen sind mehrere Restriktionen definiert, die den Ablauf der Entwicklungsaktivität regeln und die ab dem Augenblick gültig sind, an dem eine Version erreicht wird.