

Kurzbeschreibung iteratec Vorgehensmodell

Stand 08.03.2003



Rahmen und Hilfestellungen

Kurzbeschreibung des iteratec Vorgehensmodells

iteratec hat ein eigenes Vorgehensmodell. Das aus iteratec Projekterfahrungen extrahierte und konsolidierte Erfahrungswissen wird in einer schlanken, auf das Wesentliche konzentrierten Form für die Adaption in Projekten bereitgestellt. Durch das schlanke Gerüst wird für die Durchführung der iteratec Projekte ein Rahmen vorgegeben und konkret anwendbare Hilfestellungen für das Gesamtprojekt als auch für alle Kernaktivitäten gegeben.

Schwerpunkte

Das iteratec Vorgehensmodell zeichnet sich vor allen Dingen durch folgende Schwerpunkte aus:

Prozessunterstützung

- Prozessunterstützung

Selbstverständlich ist entscheidend, welche Ergebnisse in welcher Qualität im Verlauf des Projektes erarbeitet werden. Dabei reicht es jedoch nicht, im Vorgehensmodell die Dokumenttypen im Detail festzulegen, ohne Aussagen zu treffen, wie und wann die Inhalte zu erarbeiten sind. Durch das iteratec Vorgehensmodell erhält das Team einen „roten Faden“, an dem entlang es sich durch das Projekt hangeln kann. Eine Vielzahl von Unterstützungsmaßnahmen, wie z. B. die Bereitstellung von Checklisten, Vorlagen, Anleitungen und Beispieldokumente tragen zur Sicherheit und zu einer beschleunigten Ergebniserstellung bei. Hilfe statt Vorschrift ist hier das Motto.

Risikofokussierendes Vorgehen

- Risikofokussierendes Vorgehen

Oft gibt es in den Projekten noch an einem relativ späten Zeitpunkt ein böses Erwachen. Plötzlich tauchen unerwartete Schwierigkeiten bei der Integration von Teilsystemen auf, technische Komponenten sind inkompatibel zueinander, die Performance stimmt nicht oder aber technische Entwurfsentscheidungen sind nicht umsetzbar. All diese Dinge sind vorhersehbare Projektrisiken, die, wenn man sie rechtzeitig benennt und zielgerichtet bearbeitet, bereits in frühen Projektphasen beseitigt, vermindert oder kontrolliert werden können.

Partizipation des Kunden

- Partizipation des Kunden

Bei der Softwareentwicklung ist es wie beim Hausbau: Ohne intensive Beteiligung und Mitarbeit des Bauherrn kann kein befriedigendes Ergebnis erzielt werden. Das iteratec Vorgehensmodell zielt deshalb darauf ab, den Kunden eng und frühzeitig in den Anforderungsermittlungs- und Lösungskonzeptionsprozess mit einzubeziehen. Für iteratec ist es sehr wichtig, dass frühzeitig ein gemeinsames Verständnis über das zukünftige System bei allen Projektbeteiligten erreicht wird. Mittel dazu sind u.a. Mockup-

Software Architektur Entwurf

Workshops, Klick-Prototypen oder Oberflächenprototypen oder aber der Einbezug in den Lösungskonzeptions- und Validierungsprozess insbesondere über Szenarien. Durch die frühzeitige durch das Vorgehen erzwungene Beschäftigung mit der Lösung wird sichergestellt, dass Missverständnisse eliminiert werden und „zwangsläufige Änderungen“ frühzeitig aufgedeckt und kontrolliert eingesteuert werden können. Dies wird durch ein explizites Änderungsmanagement und Risikomanagement im Rahmen der Projektmanagement-Aktivitäten gewährleistet.

- Software Architektur Entwurf

Besonderer Bedeutung kommt heute dem Software Architektur Entwurfsprozess zu. Aufgrund der sich permanent verändernden technischen Randbedingungen und der erhöhten technischen Komplexität in den Projekten ist ein erhöhtes Risiko im Bereich der Software Architektur gegeben. Leider können wir dabei nur begrenzt vom Erfahrungswissen der älteren Kollegen profitieren. Vieles müssen wir uns in jedem Projekt neu erarbeiten und auch dafür benötigen wir eine systematische Vorgehensweise, die in das Vorgehensmodell integriert sein muss. Wir fassen das unter der Überschrift „architekturzentrierter Entwicklungsprozess“ zusammen.

Unser architekturzentrierter Entwicklungsprozess stellt in geplanten Iterationen sicher, dass die Software Architektur des Systems frühzeitig tragfähig festgelegt wird. Unter Software Architektur verstehen wir dabei die Struktur und das Verhalten des zukünftigen Systems sowie die Regeln, die bei der Umsetzung einzuhalten sind.

Durch den architekturzentrierten Entwicklungsprozess wird vor der eigentlichen Entwicklung des Systems sichergestellt, dass

- alle Randbedingungen und architekturrelevanten Anforderungen (funktional und nichtfunktional) ermittelt und hinterfragt werden,
- die Kernproblemstellungen herausgearbeitet werden,
- nachvollziehbare Entwurfsentscheidungen getroffen werden,
- nicht-funktionale Anforderungen explizit berücksichtigt werden,
- eine verständliche knappe Software Architektur-Beschreibung, Leitlinien und Richtlinien für die Umsetzung erstellt werden und
- die Tragfähigkeit der gewählten Architektur vor der Übergabe an die Entwicklung validiert wird.

- Iteratives und inkrementelles Vorgehen

Die meisten Softwareprojekte bergen große Risiken in sich, da sie

Iteratives und inkrementelles Vorgehen

Moving Target

- zunächst mit unsicheren und unvollständigen Anforderungen starten,
- sehr groß und komplex sind und
- das Projektteam mit neuen Techniken oder dem spezifischen Domänenwissen noch nicht vertraut ist.

Erschwerend kommt hinzu, dass sich während des Projektes häufig Anforderungen, Projektziele, Randbedingungen (z. B. veränderte Geschäftziele des Kunden, „Politische“ Entscheidungen) und „nicht-funktionale“ Anforderungen erheblich verändern. Der iterative Softwareentwicklungsprozess zielt darauf ab, diese Änderungen („**Moving Target**“, siehe Abbildung 1) und deren Auswirkungen kontrolliert und nicht chaotisch in das Projekt einzubringen, und damit dem Management eine möglichst präzise Steuerung des Projektablaufs zu ermöglichen. Ziel ist es dabei, den kürzesten und preislich günstigsten Weg vom Ausgangspunkt bis zur tatsächlichen Lösung zu wählen. Dabei macht es auch keinen Sinn, die Projektziele anfangs festzulegen und im Projektverlauf keine Veränderungen zuzulassen. In dem Fall müsste mit Erreichen der zu Projektbeginn geplanten Lösung ein Folgeprojekt mit der Umsetzung der nun bekannten Projektziele starten, um die tatsächliche Lösung zu realisieren. Das ist aber in jedem Fall teurer als das Nachhalten der Ziele im Projektverlauf unter ständiger Reduktion des Entscheidungsspielraums.

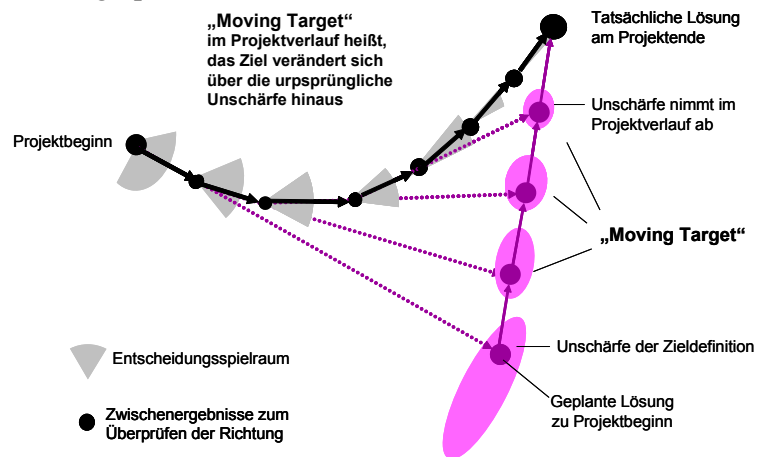


Abbildung 1: Verfolgung des "Moving Target" im iterativen Entwicklungsprozess

Die inkrementell iterative Entwicklung ist ein probates Mittel, um die Komplexität eines Entwicklungsvorhabens zu beherrschen, und bedeutet eine bewusste, mehrfache Überarbeitung von Entwick-

Der Prozess

lungsergebnissen, verbunden mit einer stufenweisen Realisierung des Gesamtsystems. Alle Aktivitäten im Projektverlauf von der Planung über die Anforderungsanalyse bis hin zur Umsetzung und der Einführung werden in einem geplant iterativen und risikofokussierenden Prozess durchgeführt. D. h. gesteuert durch das Risikomanagement werden Ergebnisse erstellt, überprüft und – ggf. mehrfach – überarbeitet.

Wie bei der klassischen Vorgehensweise durchläuft der **Prozess** verschiedene Phasen, in denen zunächst das Problem zu verstehen ist, später die Lösung definiert und erstellt wird bevor das System letztendlich eingeführt werden kann. Der wesentliche Unterschied zum „Wasserfallmodell“ liegt darin, dass bereits von Anfang an die gewonnenen Erkenntnisse in lauffähige Software umgesetzt werden (von Risikoprototypen über technische und fachliche Durchstichfunktionen bis zu lauffähigen Komponenten), um an Hand der eigenen Erfahrungen und des Feedbacks der Anwender zu lernen.

Bei der Iterationsplanung wird die Funktionalität des Systems in Liefer-Pakete, den Inkrementen, untergliedert und eventuell den verschiedenen Teilprojekten bzw. Teams zugeordnet oder bei kleineren Projekten sequentiell abgearbeitet. Jedes Inkrement wird selbst wieder iterativ entwickelt, wobei die Anzahl der Iterationen (min. 2, max. 4) von der Komplexität und dem fachlichen Reifegrad der zu erstellenden Funktionalität abhängt. Die Gesamtfunktionalität des zu entwickelnden Systems wächst mit jedem erstellten Inkrement. Zusätzlich wird mit jeder Iteration ein Projektfortschritt durch ein höheres Maß an Präzision und Vollständigkeit erreicht.

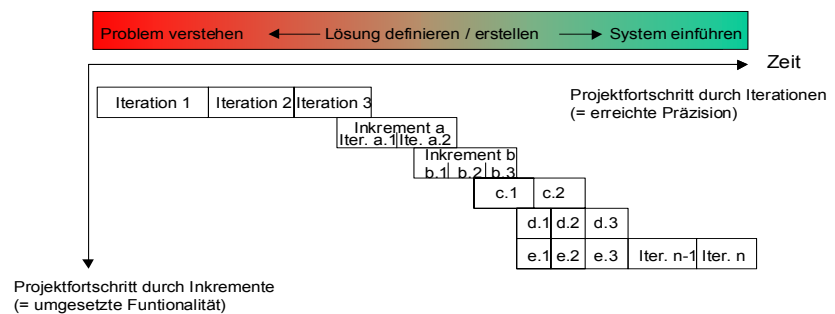


Abbildung 2 : Iterationen und Inkremente

Innerhalb jeder Iteration werden alle wesentlichen Aktivitäten wie Projektmanagement, Analyse, Design, Implementierung und Test durchgeführt, jedoch in Abhängigkeit vom Projektfortschritt mit unterschiedlichen Schwerpunkten und Intensitäten. Jede Iteration hat

klar definierte Ziele und Termine. Der Charakter der Iterationen verändert sich mit der Zeit. Liegt anfangs das Gewicht noch auf Analyse und Design, überwiegt bei späteren Iterationen Implementierung und Test. Iteration n-1 und n aus Abbildung 2 liefern keine nennenswerte zusätzliche Funktionalität mehr sondern dienen der Überarbeitung der Gesamtfunktionalität und der Fehlerbehebung.

Inkremete liefern lauffähige Softwarestände (Liefer-Pakete). Diese können projektintern für die Integration zu größeren Auslieferungseinheiten verwendet werden oder auch an den Anwender zum Test oder als Produktiv-Version ausgeliefert werden.

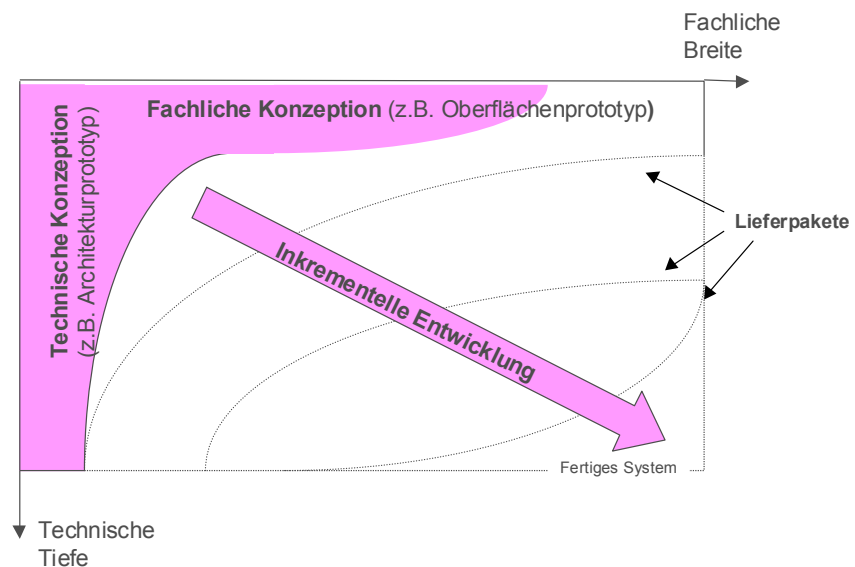


Abbildung 3 : Inkrementelle Entwicklung

Der inkrementell iterative Prozess eignet sich hervorragend, um die eigenen Planungen fortlaufend zu verbessern. Jede Iteration benötigt einen Vorlauf zur Planung (z. B. Festlegung der Ergebnisse des Iterationsschrittes) und einen Nachlauf zum Review der Ergebnisse. Da nach jeder Iteration die Planung mit der Realität abgeglichen werden kann, ergeben sich Messpunkte, um zukünftige Planungen genauer durchführen zu können.

- Phasen und Kernaktivitäten

Das iteratec Vorgehensmodell ähnelt in seinen Grundzügen dem RUP und gibt dem Projekt durch die Aufteilung in Zeitabschnitte (Phasen) und vorgegebene Kernaktivitäten, die in den unterschiedlichen Phasen mit unterschiedlichem Schwerpunkt und Intensität durchgeführt werden, eine Orientierung.

Fortlaufende Planung

Phasen und Kernaktivitäten

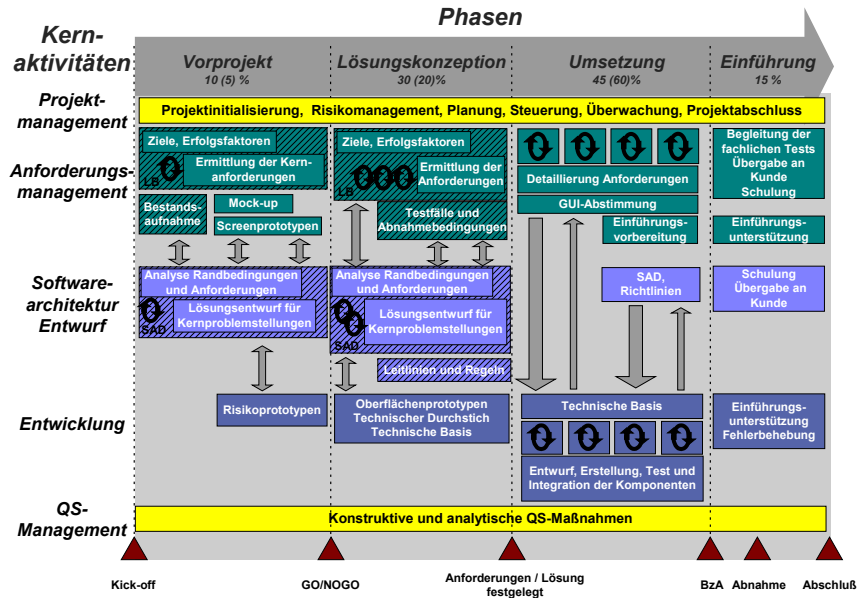


Abbildung 4 : Phasenmodell

Phasen und Kernmeilensteine

Unterschieden werden die Phasen Vorprojekt, Lösungskonzeption, Umsetzung und Einführung. Jede der Phasen schließt mit einem Meilenstein ab, der aufgrund seiner Bedeutung und zur Unterscheidung von weiteren projektspezifischen Meilensteinen als Kernmeilenstein bezeichnet wird.

Phase	Kernmeilenstein als Abschluss	Erläuterung
Vorprojekt	Entscheidung Projektdurchführung / Machbarkeit / Zielrichtungsdefinition (Go or NoGo)	Ziele, Kernanforderungen und Erfolgsfaktoren aufgenommen und abgestimmt. Randbedingungen und Anforderungen mit größtem Einfluss auf Software Architektur ermittelt (Architekturtreiber). Grobes Lösungskonzept erstellt und abgestimmt: - architekturelevante Vorgaben und Anforderungen ermittelt. - ggf. Lösungsalternativen aufgezeigt. - Machbarkeit untersucht. Entscheidungsgrundlage für den Auftraggeber erstellt. Detaillierte Planung für die Lösungskonzeption Grobe Zeit- und Kostenschätzung für das Gesamtprojekt erstellt. Wirtschaftlichkeit analysiert.

Phase	Kernmeilenstein als Abschluss	Erläuterung
Lösungskonzeption	Anforderungen und Lösung festgelegt ¹ / festpreisfähig	Anforderungen vollständig erhoben, Testfälle und Abnahmebedingungen festgelegt, mit allen Projektbeteiligten abgestimmt und vom Auftraggeber abgenommen. (Basis für Change Request Verfahren) Software Architektur (Lösungskonzeption und Vorgaben für die Umsetzung des Systems) für das zu erstellende System festgelegt und deren Tragfähigkeit sichergestellt. Basis für die Entwicklung des Systems gelegt (Technische Basis, Leitlinien und Regeln). Detaillierte Projektplanung und Kostenabschätzung für das restliche Projekt (Umsetzung und Einführung)
Umsetzung	Bereitstellung zur Abnahme	Softwareentwicklung im iterativ inkrementellen Prozess (Basis Komponenten der Software Architektur) abgeschlossen. Entwickler- und Integrationstests erfolgreich durchgeführt.
Einführung	Abnahme System eingeführt	Abnahme des Systems aufgrund der festgelegten Testfälle und der Abnahmebedingungen. Alle Dokumente an Kunden übergeben. Abnahmetests erfolgreich durchgeführt. Anwender geschult. System in Betrieb und an den Auftraggeber übergeben. Anlaufphase abgeschlossen.

¹ Detaillierungsebene hinreichend für Change Request Verfahren, weitere Präzisierung erfolgt in der Entwicklungsphase. Hilfestellung durch Checklisten.

Kernaktivitäten

Zur Strukturierung aller Tätigkeiten, die für die Erstellung der Projektergebnisse, d. h. zur Erreichung der Kernmeilensteine, durchgeführt werden müssen, definiert das iteratec Vorgehensmodell Kernaktivitäten, wie z. B. Projektmanagement, Anforderungsmanagement, Software Architektur Entwurf, Entwicklung oder Qualitätsmanagement. Diese Kernaktivitäten werden geplant iterativ durchgeführt, d. h. es wird jeweils festgelegt, welche und wie viele Iterationen mit welcher Dauer, welchem Vorgehen und welchen Zielen durchlaufen werden – „trial and error“ ist strikt zu vermeiden. Für die Durchführung der Aktivitäten werden den jeweils Verantwortlichen klare inhaltliche und terminliche Orientierungspunkte gegeben. Ziel ist es dabei, einen roten Faden zu legen, anhand dessen das Projektteam durch das Vorgehensmodell navigieren kann und sicher zu seinem Ziel gelangt.