



Bernd Oestereich

Management und iteratives Vorgehen in größeren Softwareprojekten

Geschäftsführer
oose.de Dienstleistungen für innovative Informatik GmbH
Oberstraße 14b, 20144 Hamburg
E-Mail: boe@oose.de

Vorweg: Klärung einiger verwendeter Begriffe

Eine **Phase** ist ein zeitlicher bzw. sachlogischer Gliederungsabschnitt eines Projektes. Eine Phase faßt eine Menge von Aktivitäten und Ergebnissen zu einer Planungs- und Kontrolleinheit zusammen. Am Ende jeder Phase steht gewöhnlich ein Meilenstein, der die in der Phase zu erzielenden Inhalte definiert.

Ein **Meilenstein** definiert einen Termin, zu dem eine Menge von Ergebnissen in vorgegebener Detaillierung und Vollständigkeit nachprüfbar und formal dokumentiert vorliegen muss. Liegen die Ergebnisse zum geplanten Termin nicht vor, wird der Meilenstein verschoben. Ein Meilenstein ist ein Hilfsmittel zur Planung und Überwachung eines Entwicklungsprozesses.

Eine **Timebox** definiert einen unverrückbaren Zeitrahmen, in dem eine Menge von Ergebnissen in vorgegebener Detaillierung und Vollständigkeit vorliegen soll. Liegen die Ergebnisse nicht wie geplant vor, werden die offenen Teile in eine nachfolgende Timebox verschoben. Hierzu werden zum geplanten Endtermin die tatsächlich erreichten Ergebnisse bestimmt. Eine Timebox ist ein Hilfsmittel zur Planung und Überwachung eines Entwicklungsprozesses.

Vgl. Scrum: "Sprint"

Vorweg: Klärung einiger verwendeter Begriffe

Eine **Iteration** ist ein in ähnlicher Weise mehrfachvorkommender Zeitabschnitt in einem Prozess. Iterationen können Timeboxen sein.

Ein **Inkrement** ist ein Teilergebnis, das innerhalb einer Iteration entstanden ist.

Ein **Build** ist eine gewöhnlich unvollständige und vorübergehende aber ausführbare Version eines in Entwicklung befindlichen Systems.

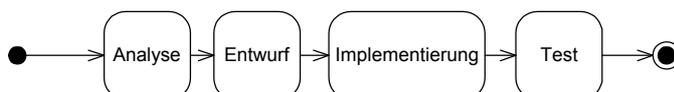
Ein **Release** ist ein Build, das an den Auftraggeber (bzw. Produktempfänger) ausgeliefert werden kann.

Wasserfallmodell – bestechend einfach

- Zitat Sohn von W. Royce:

„Mein Vater war immer ein Vertreter von iterativen, inkrementellen und evolutionären Entwicklungsmodellen. Sein Arbeitspapier beschrieb das Wasserfallmodell als die einfachste Möglichkeit, welches aber nur für die unkompliziertesten Projekte funktioniert.“

- Für jedes Problem gibt es eine Lösung, die einfach, schön und falsch ist:
Wasserfallmodell lässt sich einfach beschreiben (und anwenden?).



Wasserfallmodelladaption

- Adaption als Department of Defence Standard DoD-STD-2167
Autor des Standards: „*He was not familiar with the practice of timeboxed iterative development and evolutionary requirements at the time, and in hindsight, said he would have made a strong IID recommendation, rather than what was said in 2167.*“
[Craig Larman, The historical accident of waterfall validity, in Agile & Iterative Development, 2004]
- Weitere Standardorganisationen ziehen nach: V-Modell, CMM/SEI
- Kritische Masse an Anwendern und Experten wird erreicht

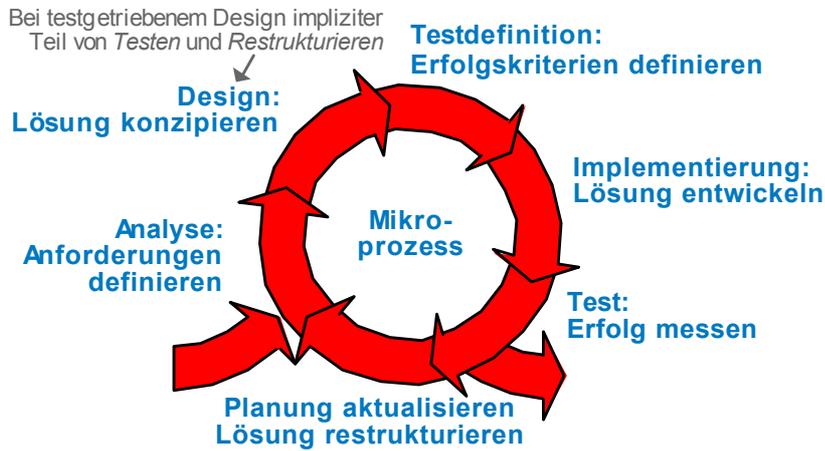
Iteratives Vorgehen

- Sind wir mit iterativem Vorgehen besser?
- Immer?
- Unter welchen Bedingungen?
- Für welche Situationen?
- Welche Grenzen gibt es?
- Sind große Projekte iterativ abzuwickeln?



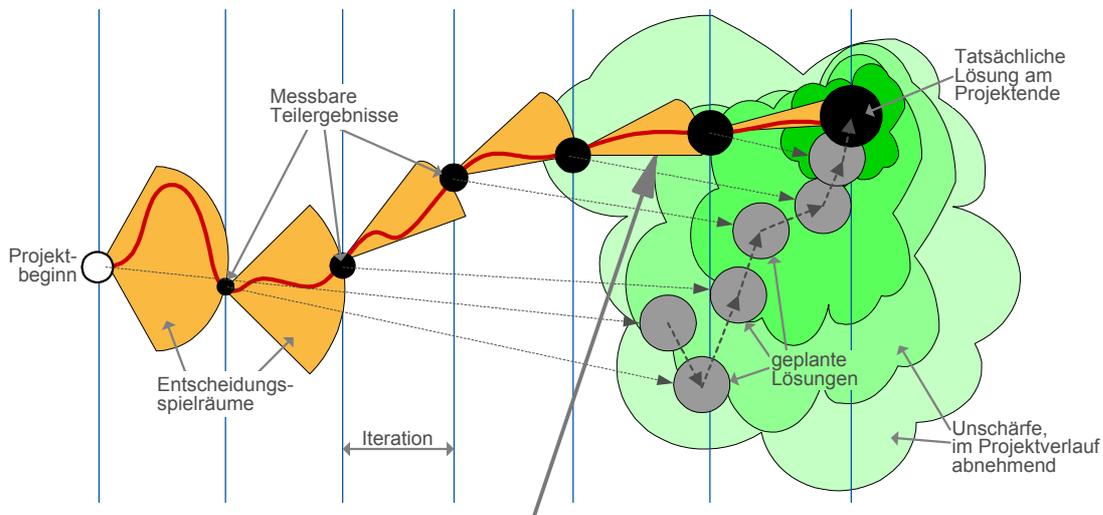
Mikroprozess: Vorgehen innerhalb einer Iteration

In jeder Iteration finden prinzipiell, jedoch mit verschiedenen Ausprägungen die elementaren Entwicklungsaktivitäten Analyse, Design, Implementierung und Test statt.



© 2003 by oose.de GmbH

Die Annäherung von Ziel und Ergebnis im Projektverlauf



Die Beweglichkeit (Agilität) nimmt bei iterativem Vorgehen ab

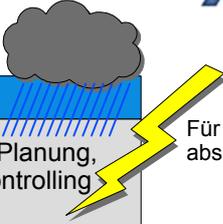
© 2003 by oose.de GmbH

Möglichkeiten und Grenzen des iterativen Vorgehens

- Iteratives Vorgehen ist viel agiler als Wasserfall
- Restrukturierungen auf Mikroebene ist unproblematisch
- Keine Einschränkungen für kleine/kurze Projekte
- Aber: Möglichkeiten zur Beweglichkeit nehmen im Projektverlauf ab bzw. Restrukturierungsaufwände auf Makroebene sind kaum durchsetzbar - dies trifft Großprojekte

© 2003 by oose.de GmbH

Iterationsdauer

	 Vorteile	 Nachteile
Kurze Iterationen	mehr Feedback mehr Handlungsoptionen	mehr Aufwand für Planung, Steuerung und Controlling
Lange Iterationen	weniger Aufwand für Planung, Steuerung und Controlling	weniger Feedback abstraktere Ziele weniger Ergebnisorientierung

Für Großprojekte absolut kritisch

- Iterationsdauer ist abhängig von Teamgröße

Mitarbeiterzahl	Iterationsdauer
4	2 - 4 Wochen
10	3 - 5 Wochen
40	6 - 8 Wochen
>40	6 - 10 Wochen

Vgl. SCRUM:
7 MA -> 6 Wochen (30 Arb.tg.)

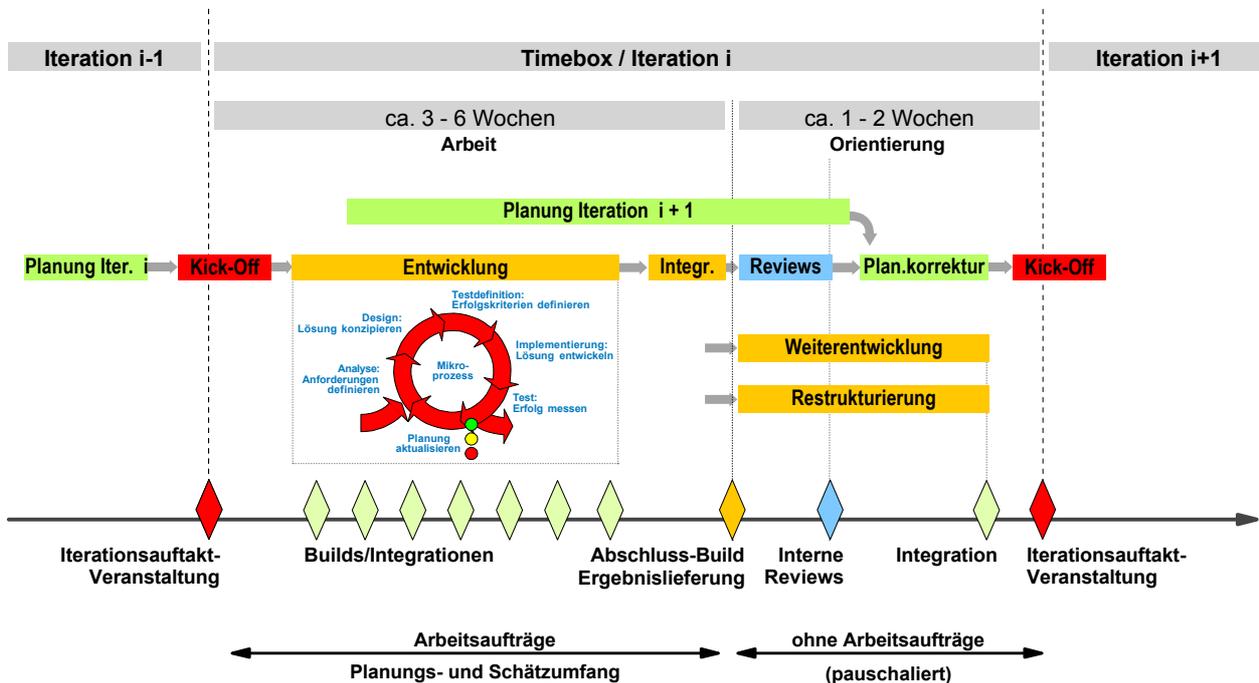
Vgl. Kruchten/RUP:
40 MA -> 12 Wochen
150 MA -> 32 Wochen

Vgl. Jutta Eckstein: 1 Woche
"Je größer das Projekt,
desto kürzer die Iterationen."

Aber: Je mehr ein Projekt in der Krise,
desto kürzer die Iterationen.

© 2003 by oose.de GmbH

Das Iterationsmodell des OEP: Meilensteine, Mikroprozess, Planung



© 2003 by oose.de GmbH

Dauer und Größe

- Iterationsdauer 4 - 8 Wochen je nach Projekteigenschaften
 - Während eines Projektes immer gleichlange Iterationen
 - Abweichung von +/- 3 Tage unkritisch
- Iteration besteht aus Arbeits- und Orientierungsabschnitt
 - Langer Arbeitsabschnitt wird gesteuert durch explizite Arbeitsaufträge, konzentriertes, ungestörtes, eigenverantwortliches Arbeiten
 - Kurzer Orientierungsabschnitt zum Rückblick, Review, Planungsanpassung und Neuausrichtung des Projektes

Großprojekte in Teams von 4 - 8 Mitarbeitern aufteilen
- Gesetzmäßigkeiten kleiner und agiler Projekte auf Großprojekte übertragen
 - kleine unabhängige Teams von 4 - 8 Personen bilden
 - Jedes Team bekommt einen Sprecher (kann iterationsweise wechseln) als Schnittstelle nach Außen
- Ggf. regelmäßige *Stabilisierungs-Iterationen* einplanen:
 - Dauer 1 - 2 Wochen
 - Inhalt: keine neue Funktionalität, sondern nur Fehlerkorrektur, Robustheit erhöhen, Restrukturieren

© 2003 by oose.de GmbH

Zeitliche Gliederung in Phasen

- Unterteilung in Architektur- und Entwicklungsphase

Meilenstein der Architekturphase:

- Oberflächliche, aber in der Breite vollständige Festlegung aller Anforderungen
- Beherrschung aller elementaren Projektrisiken
- Langer Arbeitsabschnitt wird gesteuert durch explizite Arbeitsaufträge, konzentriertes, ungestörtes, eigenverantwortliches Arbeiten
- Kurzer Orientierungsabschnitt zum Rückblick, Review, Planungsanpassung und Neuausrichtung des Projektes

Ziel: der natürlichen
Beweglichkeitsabnahme
entgegen wirken

- Zweistufige Festpreise bzw. zweistufige Beauftragung möglich:
 - Initiale Festlegung von Terminen, Budget, Qualität, Funktionalität etc. auf Basis angenommener sehr grober Anforderungen (geringe Schätzgenauigkeit), z.B. Anzahl der Anwendungsfälle, Features, Dialoge etc.
 - Zweite fundierte Schätzung auf Basis in der Breite vollständiger Anforderungen und nachvollziehbare Darstellung des Zusammenhanges zu initialer Schätzung, z.B. Änderung in der Anzahl der Anwendungsfälle, Features, Dialoge sowie erkannter Risiken

(Normale, einstufige) Festpreisprojekte: geht auch, aber weniger gut -> Vortragszeit ist zu kurz für dieses Thema...

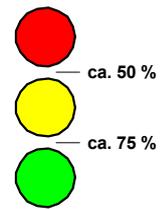
Anforderungen

- Beschreibung der Anforderungen aus Anwender- bzw. Auftraggebersicht d.h. geschäftlicher Wert im Vordergrund
- Form der Anforderungen für gesamte Projektdauer einheitlich und gemeinsam mit Auftraggeber festlegen
 - z.B. Anwendungsfälle, Features etc.
 - z.B. XP-User Stories
- Entwickler und Anwender priorisieren gemeinsam die Anforderungen und berücksichtigen somit:
 - geschäftlichen Nutzen durch Anwender
 - technische Implikationen durch Entwickler
- Oberflächliche aber in der Breite vollständige Festlegung der Anforderungen
 - Abgrenzung des Projektauftrages (hilfreiches Konzept: essenzielle Use Cases)
 - Initiale Festlegung von Terminen, Budget, Qualität, Funktionalität etc. Nachvollziehbare Schätzung von Terminen, Kosten etc.
 - Regelmäßige Überprüfung und ggf. Anpassung des Projektauftrages auf dieser Basis

Arbeitsaufträge und "Läufer"

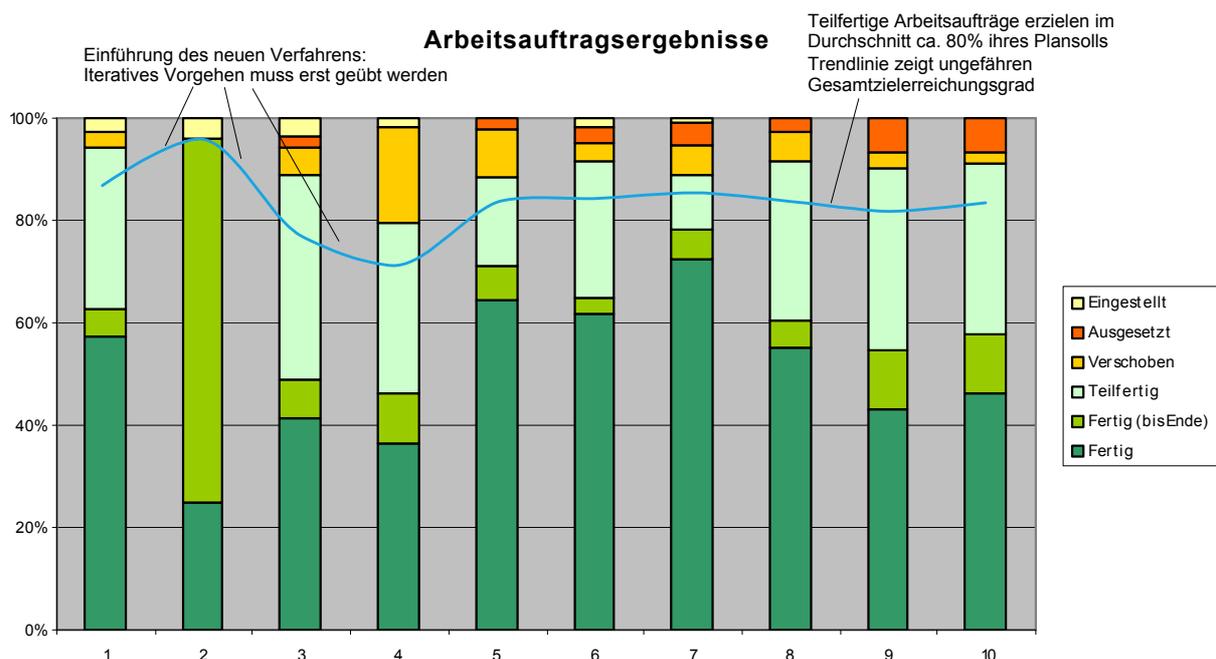
- Entwicklung von Arbeitsaufträgen auf Basis der priorisierten Anforderungen
Für jeden Arbeitsauftrag gilt:
 - 1 Ergebnisverantwortlicher formuliert Arbeitsauftrag und erwartete Ergebnisse und sucht sich weitere Mitarbeiter und andere notwendige Ressourcen
 - 0..n weitere Mitarbeiter
 - 1 Gutachter zur späteren Ergebnisbeurteilung wird von Anfang an festgelegt
 - Formulierung objektiv messbarer Ergebnisse
 - Einmalige Aufwandschätzung durch Ergebnisverantwortlichen, ggf. zusammen mit den übrigen Beteiligten

- Wöchentliche Wahrscheinlichkeitsabfragen zur Zielerreichung der Arbeitsaufträge
"Wie groß ist die Wahrscheinlichkeit, dass dieser Arbeitsauftrag am Ende der Iteration wie geplant fertig ist?"
 - Keine Restaufwandschätzungen, nur Wahrscheinlichkeitswert (0 - 100%)
 - Einsatz eines "Läufers": jede Woche wechselnder Projektmitarbeiter, der Montagsvormittag alle Ergebnisverantwortlichen kurz besucht und Wahrscheinlichkeitswerte abfragt - und dabei auch viele andere wichtige Informationen erfährt.
 - Arbeitsauftragsspezifische Fakturierung der Arbeitszeit der einzelnen Mitarbeiter (für Controlling und Metriken)



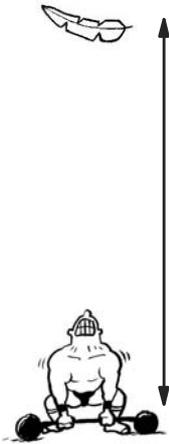
© 2003 by oose.de GmbH

Arbeitsauftragsergebnisse



© 2003 by oose.de GmbH

Arbeitsaufträge: Verschiedene Agilitätsgrade zwischen XP und schwergewichtig



- Task-Cards, Planungs-Pinwand, 1 - 4 Wochen/Iteration, 1 - 10 Mitarbeiter
- Arbeitsaufträge, Arbeitsauftragsliste, Wiki, Excel, 2 - 6 Wochen/Iteration, 6 - 15 Mitarbeiter
- Einfache Arbeitsauftragsdatenbank, 2 - 12 Wochen/Iteration, 6 - 40 Mitarbeiter
- Arbeitsauftragsdatenbank mit Projektmanagementsoftware-Export
Verwendung von Projektmanagementsoftware als Berichtsgenerator
Einhaltung schwergewichtiger Unternehmensstandards
- Arbeitsauftragsdatenbank mit Abhängigkeitsplanung innerhalb einer Iteration
- Abhängigkeitsplanung innerhalb einer Iteration mit Projektmanagementsoftware (Gantt-Diagramme)

Bearbeitung der Arbeitsaufträge

- Möglichst kontinuierliche Integration
- Möglichst tägliches Build
- Hoher Grad von Testautomatisierung

Integriertes Risikomanagement

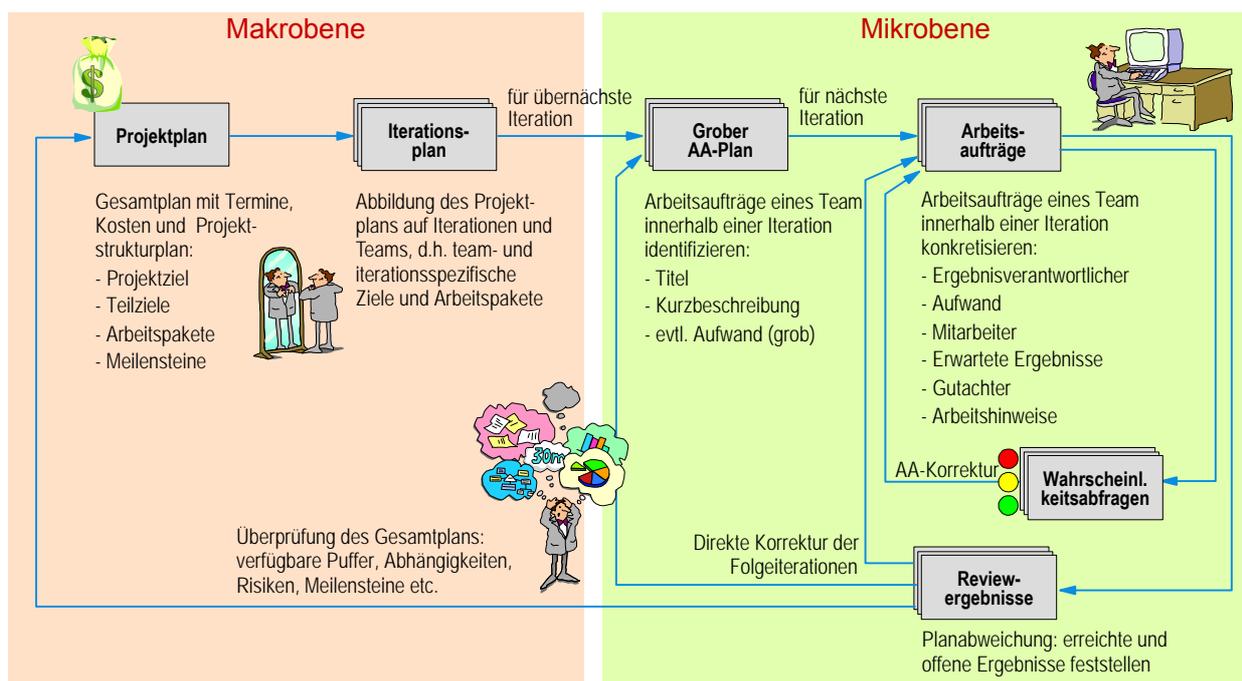
- Risiko-Workshops vor jeder Iteration
 - Identifikation und Dokumentation von Risiken
 - Untersuchung und Bewertung von Risiken
 - Entwicklung von Maßnahmen zur Vermeidung und Schadenminimierung
 - Planung der Risiko-Handhabung
ggf. Entwicklung von eigenen Arbeitsaufträgen hierzu
- Transparenz für Auftraggeber und Anwender herstellen

Zusammenfassung OEP-Timebox-Verfahren

- Iterationsdauer 4 - 8 Wochen
 - Iteration besteht aus Arbeits- und Orientierungsabschnitt
 - Großprojekte in Teams von 4 - 8 Mitarbeitern aufteilen
- 
- Beschreibung der Anforderungen aus Anwender-/Auftraggebersicht (Anwendungsfälle, Features etc.)
 - Entwickler und Anwender priorisieren gemeinsam die Anforderungen
 - Entwicklung von Arbeitsaufträgen auf Basis der priorisierten Anforderungen (jeweils: 1 Verantwortlicher, 0..n Mitarbeiter, 1 Gutachter), Formulierung objektiv messbarer Ergebnisse, einmalige Aufwandschätzung durch Beteiligte
 - Wöchentliche Wahrscheinlichkeitsabfragen zur Zielerreichung der Arbeitsaufträge (keine Restaufwandschätzungen) mit Hilfe eines "Läufers" Arbeitsauftragspezifische Fakturierung der Arbeitszeit (für Controlling und Metriken)
 - Kontinuierliche Integration, tägliches Build, hoher Grad von Testautomatisierung
 - Risiko-Workshops vor jeder Iteration (Identifikation von Risiken, Planung der Risiko-Handhabung)
 - Für größere Projekte: Unterteilung in Architektur- und Entwicklungsphase.
Meilenstein der Architekturphase:
 - Oberflächliche, aber in der Breite vollständige Festlegung aller Anforderungen
 - Beherrschung aller elementaren Projektrisiken
- Ziel: der natürlichen Beweglichkeitsabnahme entgegen wirken

© 2003 by oose.de GmbH

Planungsdokumente im iterativen Entwicklungsprozess



© 2003 by oose.de GmbH



Vielen Dank.

Literatur / Weitere Infos

- B. Oestereich: *Beweglich bleiben: Möglichkeiten und Grenzen iterativen Vorgehens*; Objekt-Spektrum 1/2005
- B. Oestereich: *Die Macht des Rhythmus, das OEP-Timepacing-Verfahren*; Objekt-Spektrum 2/2000
- www.oose.de/publikationen
- Jutta Eckstein: *Agile Softwareentwicklung im Großen*, dpunkt.verlag, 2004
- Besuchen Sie unsere Schulungen!