

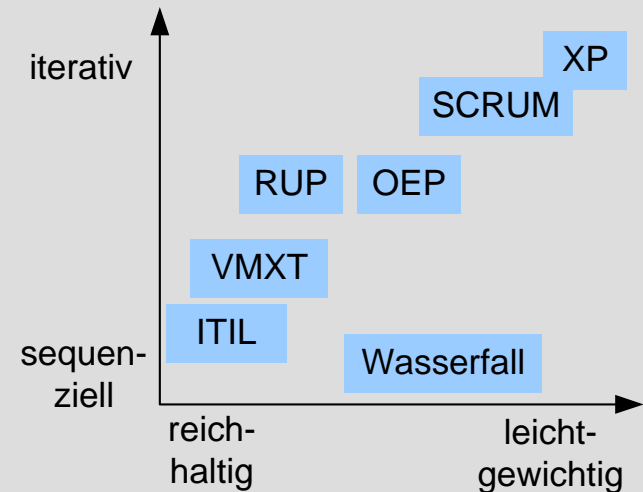


Definitionen und Einordnung

Eine **Methodik** bezeichnet die Gesamtheit einzelner Methoden z.B. zur Aufwandsschätzung oder Modellierung von Teilaspekten. UML ist bspw. keine Methodik, sondern die Anleitung, wie sie sinnvoll anzuwenden ist.

Ein **Vorgehensmodell** beschreibt, welche typischen Aktivitäten und Ergebnisse im Rahmen eines Projektes entstehen, wie diese im Detail aussehen, was dabei zu beachten ist, und wer sie wann und wie durchführt. Es beschreibt neben einer → *Methodik* vor allem auch unterstützende Prozesse (PM, QS etc.).


Ein **Reifegradmodell** stellt Forderungen an ein → *Vorgehensmodell* um Qualität des Softwareprozesses zu bewerten. Es dient vorrangig der Bestimmung von Verbesserungsmaßnahmen.




Charakteristik

Klassisches Vorgehensmodell mit den Phasen *Analyse*, *Design*, *Realisierung* und *Test*, die einzeln hintereinander durchgeführt werden.



- 
- einfach zu planen
 - intuitiv zu befolgen

- 
- Anforderungen müssen anfangs detailliert bekannt sein und stabil bleiben
 - Neue Erkenntnisse erfordern Rücksprung
 - Projektfortschritt schwer abschätzbar
 - Mitarbeiter unterschiedlich belastet

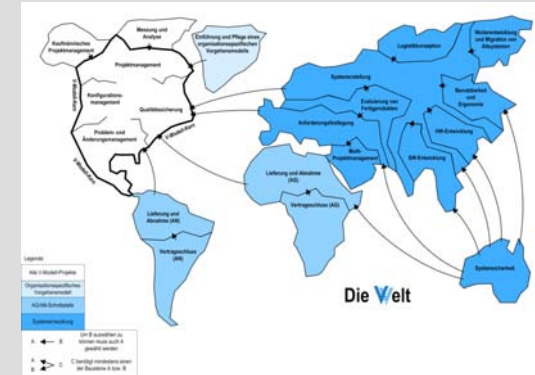
Eignung

- geringvolumige Projekte
- weitgehend klare Anforderungen
- kurze Laufzeit
- einfache, bekannte Architekturen
- risikoarme Projekte

Charakteristik

Offizielles Vorgehensmodell der öffentlichen Hand, das bei der Auswahl von Vorgehensbausteinen unterstützt und auch den Auftraggeber explizit mit einbezieht.

- mehrere Projektdurchführungsstrategien
- Tailoring: Abhängigkeiten berücksichtigt, Entscheidungsgründe dokumentiert
- Standardisierung durch Zertifizierung
- viel Konfektionierungsaufwand
- inhaltsgetrieben, d.h. kein Timeboxing
- dokumentenlastig
- keine modernen Architekturen, Prinzipien

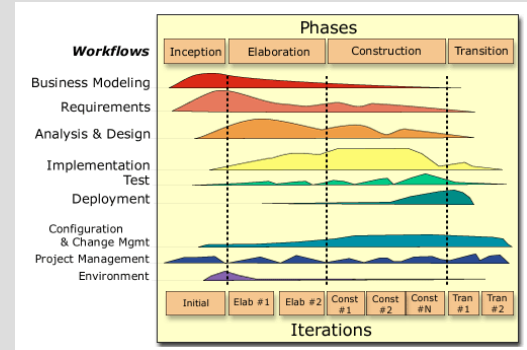


Eignung

- Öffentliche Ausschreibungen
- großvolumige Projekte
- lange Laufzeit
- hoher Formalisierungsgrad
- Sicherheitskritische oder HW-Systeme
- geringe Realisierungsrisiken

Charakteristik

Verbreitetes iteratives Vorgehen der Firma Rational (jetzt IBM), das Phasen und Workflows separat betrachtet und einen starken UML-Bezug aufweist.



- Tailoring durch Method Composer
- umfangreiche Detaillierung für UML-Projekte
- weite Verbreitung, viel Literatur, Foren, Know How
- viel Konfektionierungsaufwand
- UML-lastig
- Herstellerabhängigkeit

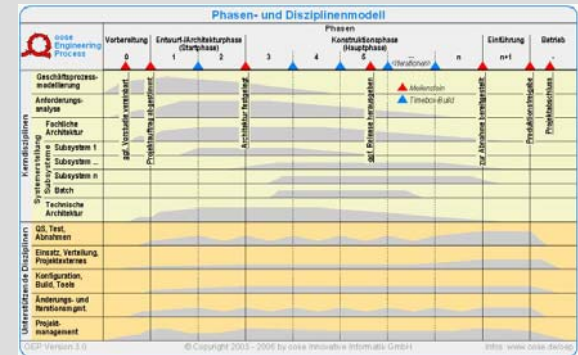
Eignung

- mittelgroße Projekte
- OO-Projekte mit hoher Benutzerinteraktion
- modellbasiertes Entwickeln mit UML
- auch für risikoträchtige neue Technologien

Charakteristik

Iteratives timeboxbasiertes Vorgehen für objektorientierte Softwareentwicklung, das agile Techniken für mittelgroße Projekte adaptiert. Grundstruktur wie Unified Process (UP).

- speziell für Inhouse-Entwicklung mit Mehrschicht-Architekturen und Legacy-System-Anbindung
- frei verfügbar
- sehr konkret und einfach
- Tailoring nur durch Wegstreichen
- UML-lastig
- Konfektionierungsaufwand bei anderen Projekttypen bzw. Architekturen



Eignung

- mittelgroße, risikoträchtige Inhouse-Projekte
- OO-Projekte mit hoher Benutzerinteraktion
- modellbasiertes Entwickeln mit UML
- Mehrschicht-Architekturen

Charakteristik

Sammlung von Arbeitstechniken, Strukturen und Rollen im Kontext eines pragmatischen, leichtgewichtigen Vorgehens, das weitgehend auf Festlegungen verzichtet.



- sehr einfache Prinzipien, leicht zu erlernen
- schnelle Reaktion auf Anforderungsänderungen
- häufige Releases, kaum Verwaltung

- Skalierung auf große Projekte fragwürdig
- wenig langfristige Erfahrungen verfügbar
- eigenverantwortliche Kultur erforderlich
- größere Last beim Fachbereich

Eignung

- kleine Projekte
- vage, kurzlebige Anforderungen
- prototypische Evaluierungen
- risikoträchtige Projekte

Charakteristik

Sehr verbreitetes, agiles Vorgehen für kleine Teams bestehend aus 12 verflochtenen Praktiken, das Realisierung in den Vordergrund stellt und Werte explizit berücksichtigt.



- schnelle Reaktion auf Anforderungsänderungen
- häufige Releases, kaum Verwaltung
- weite Verbreitung, viel Literatur, Foren, Know How

- eigenverantwortliche Kultur erforderlich
- sehr hohe Last beim Fachbereich

Eignung

- kleine Projekte
- vage, kurzlebige Anforderungen
- prototypische Evaluierungen
- risikoträchtige Projekte
- Firefighting-Projekte

Charakteristik

De-facto-Standard für eine IT-Infrastruktur zur Umsetzung eines IT-Service-Managements, vor allem für Betriebsübergang, Pflege und Verbesserung von Services.



- Standardisierung von SLA-Definitionen
- Beurteilung von Service-Qualität
- weltweite Verbreitung und Akzeptanz
- nicht auf Entwicklungsprojekte anwendbar
- recht komplex und dokumentenlastig

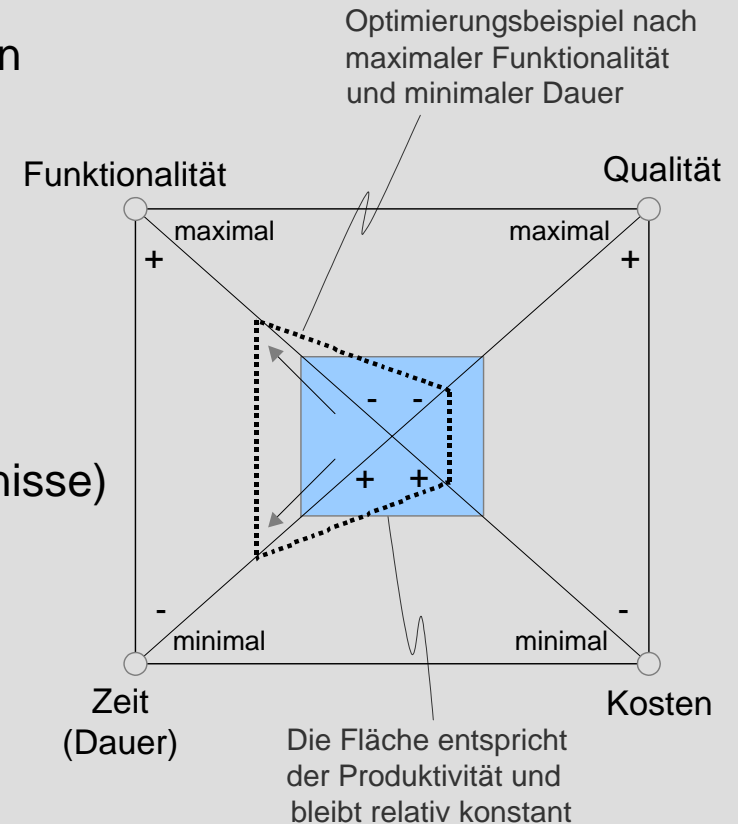
Eignung

- IT-Betriebsprojekte
- Lebenszyklus-Planung, Applikations-Management
- größere Projekte mit langer Laufzeit

Was ist zu tun?

- das richtige Vorgehensmodell strategisch festlegen
 - Typische Projektmerkmale identifizieren
 - Marktsituation betrachten
 - Optimierungsstrategie festlegen
- das Tyloring definieren
- die Aktualität gewährleisten
- die Artefakte benennen (Rollen, Aufgaben, Ergebnisse)
- die Beschreibungsform bestimmen (z. B. mit EPF)
- das Wissen multiplizieren
- für Akzeptanz sorgen (!)

*Das Beschreiben ist wichtiger
als die Beschreibung ...*



typische Anpassungen

- Äußere Form
Farben, Corporate Identity, Logos, anderer Name
- Integration spezieller Konzepte
z.B. weitere Meilensteine, „Quality Gates“, Rahmenrichtlinien
- Anpassung der Terminologie
Umbenennung von Phasen, Meilensteinen, Aktivitäten, Ergebnissen, Akteuren etc.
- Anpassung der Softwareentwicklungsmethodik
bspw. wegen spezieller Umgebungen, Werkzeuge, Architekturen etc.
- Anpassung der Projektmanagement-Methodik
bspw. wegen standortverteilter Projekte, spezielle Zulieferbeziehungen etc.
- Branchen- und Domänenspezifika
Spezielle Qualitäts- und Sicherheitsanforderungen, Zertifizierungen, Dokumentationspflichten
- Organisationsspezifika
Berichts-, Kontroll- und Entscheidungs-Standards, Verantwortlichkeiten etc.

häufige Ziele der Anwender

- Einführung moderner Softwareentwicklungsmethodik
Verwechslung von Methodik und Vorgehensmodell
z.B. Methodik wird benötigt, Vorgehensmodell wird als Lösung gesehen
- Vorgehensmodell als Ersatz für Know-How-Aufbau im Unternehmen
Vorgehensmodell statt Mitarbeiterausbildung
- Gezielte Verwendung einzelner Aspekte
z.B. nur iteratives Vorgehen, nur Adaption agiler Prinzipien für große Projekte
- Einführung von Standards
z.B. UML
- Unternehmensweite Standardisierung
von methodischen Erweiterungen oder Eigenerfindungen
- Behebung typischer Defizite
z.B. beim Umgang mit UML, Werkzeugen etc.

typische Reaktionen auf Einführungen

- Krampfhaftes Festklammern an „Standards“
- kritiklose Anwendung des Vorgehensmodells (Unsicherheit)
- Instrumentalisierung für Machtinteressen
- Offene Ablehnung wegen Einschränkungen/Gängelung der Betroffenen
- Ablehnung wegen Unverständnis, Unkenntnis, Missverständnisse
- Extrem selektive Anwendung
- Nichtbeachtung, Desinteresse
- Wohlwollende Annäherung, Neugierde, Interesse
stirbt schnell, wenn kein unmittelbarer Nutzen ersichtlich ist

technische und formale Maßnahmen zur Erhöhung der Akzeptanz

- Muster- und Beispieldokumente anbieten
(meistens nur für Projektleiter und unterstützende Disziplinen interessant)
- Wiki-Integration für Anwender-Feedback und gemeinsame Wissensbasis
(funktioniert weniger gut, wenn keine Wiki-Kultur vorhanden ist)
- Corporate Identity, spezifische Terminologie etc. berücksichtigen
- Schulung und Motivation des Vorgehensmodells
z. B. zielgruppenspezifische Schulungen
- Zielgruppenspezifische Sichten auf das Vorgehensmodell
- Vorgehensmodellverwendung durch/beim Coaching von Projekten
- Explizites Marketing
(z. B. einprägsame Grafiken)
- Explizite Unterstützung durch das Top-Management

inhaltliche / fachliche Maßnahmen zur Erhöhung der Akzeptanz

- Verwendung möglichst einfacher Werkzeuge (SVN, Excel, Whiteboard, Digitalkamera, Wiki etc.)
- Im Zweifel: Verständlichkeit hat Vorrang vor formaler Korrektheit oder Vollständigkeit
- Anregungen + Leitlinien zur Förderung der Kommunikation und Ergebnisorientierung (bspw. durch direkte Vorschläge von Workshops mit Musteragenda etc.)
- Möglichst einfache Rahmenbedingungen, Vereinfachung durch Weglassen z. B. Konzentration auf elementare UML-Elemente und Diagramme
- Durchgängige praxisnahe Methodik
Checklisten, Leitfragen, objektive Entscheidungskriterien zur unmittelbaren Unterstützung der Mitarbeiter (bspw. „wo fängt ein Use Case an?“, „wie schätze ich Aufwände mit Use Cases?“)
- Explizite Motivation von kritischem Umgang mit dem Vorgehensmodell um blindgläubiges (blindwütiges) Anwenden der Vorgehensmodells zu vermeiden.

Kunde A

- Vor 5 Jahren eingeführt.
- Managementtechniken wie iteratives Vorgehen, Timeboxing wurden etabliert, dies wäre wohl auch ohne VM gegangen.
- Muster, Beispiele (z.B. Projektauftrag, Berichte) fanden bei Projektleitern und dem übrigen Führungspersonal Akzeptanz und führten zu einer Vereinheitlichung der Projekte bzgl. des Berichtswesen und äußerer Projektschnittstellen.
- Methodik-Standards (z.B. use-case-getriebenes RE, UML-Erweiterungen) sind etabliert, wobei das VM nur ein unterstützendes Element war; maßgeblich waren Schulungen und Coaching der Projektmitarbeiter. Diese Standards wurden vorwiegend nicht durch das VM verordnet, sondern haben sich in der Praxis entwickelt und flossen in das VM ein.
- Die Rahmenrichtlinien (wie entstehen Projekte, Organisation von Projekten, z.B. welche Abteilung stellt fachlichen, welche technischen Projektleiter etc.), die mit dem VM festgeschrieben wurden/werden sollten, wurden über mehrere Jahre zwischen verschiedenen Parteien ausgefochten/verhandelt. Das VM war hier Anlass bzw. Reibungsfläche für die Entzündung vorhandener, schwelender Interessenkonflikte.

Fazit:

VM ist etabliert, war und ist wichtig, die Bedeutung für die Projektpraxis ist mäßig.

Kunde B

- Vordergründige Motivation: Einführung durch zentrale Methodenabteilung mit dem Ziel die Entwicklung an den verschiedenen (bisher eigenständigen, nun untergeordneten) Standorten zu vereinheitlichen.
- Methodenabteilung ist wohlwollend gegenüber Projekten, aber geistig weit weg von den Herausforderungen des Projektgeschäftes.
- Innere Motivation: VM wird verwendet, um Bedeutung der Methodenabteilung im Unternehmen zu stärken.
- Anpassung der Terminologie
- Einführung spezieller Meilensteine und benannter Iterationen
- Das VM hilft den Prozess der angestrebten Vereinheitlichung zu unterstützen. Im VM manifestieren sich Interessen, die zwischen Beteiligten auszubalancieren sind.

Fazit:

Um nach dieser Abstimmung auch in die Projekte hinein zu wirken, ist nicht die Festschreibung eines angepassten VM relevant, sondern das Etablieren durch Coaching, Schulung, Vorbild-Projekte.

Kunde C

- Ziel: Entwicklung eines eigenen Vorgehensmodells für die Abwicklung von Softwareprojekten als Dienstleister zusammen mit dessen Auftraggebern/Kunden.
- Der OEP wurde mit vollen Rechten erworben und „ausgeschlachtet“. Der Kauf des OEP und die Übernahme zahlreicher brauchbarer Elemente war deutlich billiger als eine vollständige Eigenentwicklung.
- Wesentlich war der Erwerb des im OEP steckenden methodischen Know-hows.
- Über die Akzeptanz ist nichts bekannt.

Fazit:

Verfügbare, vorgefertigte Vorgehensmodelle auszuschlachten ist oft preiswerter als die Neuentwicklung.

Akzeptanz hat viele Facetten

- Es gibt viele verschiedene Gründe, warum Vorgehensmodelle eingeführt werden.
- Verschiedene Parteien haben verschiedene Interessen.
- „Akzeptanz von Vorgehensmodellen“ ist also relativ zu sehen:
 - Was heißt „Akzeptanz“ im Kontext der jeweiligen Einführungsziele und Interessen?
 - Welche Akteure müssen welche Teile des Vorgehensmodells akzeptieren?
 - Welcher (individuelle) Grad von Akzeptanz ist anzustreben?
- Die Anwendung des Vorgehensmodells durch die Entwickler oder durch alle Akteure ist nicht immer gefordert – zumindest selten zu realisieren. Trotzdem kann das Vorgehensmodell eine wichtige Rolle spielen oder sinnvoll sein.
- Viele sehr gut arbeitende Projekte haben gar kein explizites und definiertes Vorgehensmodell.
- Viele Projekte arbeiten trotz Vorgehensmodell ineffektiv und weniger erfolgreich.
- Das meiste Wissen über Verbesserungspotenziale steckt in den Köpfen der eigenen Mitarbeiter. Es aufzugreifen und sichtbar zu machen ist die eigentliche Aufgabe.