

Die Mittel und Wege, IT Systeme und IT Infrastrukturen zu realisieren, sind immer zugleich kritisch für den Erfolg und vielfältig in ihren Ausprägungen gewesen.

its.d&m hat es im Laufe von Jahren geschafft, die Bandbreite von kaufmännischen Anforderungen über Projektmanagement bis zu architekturzentrierten Methoden für drei Projekttypen in einem Vorgehensmodell zu integrieren. Damit werden alle Aktivitäten und Ergebnisse eines Projektes untereinander und mit der technischen Basis eng gekoppelt.

Das war nur mit einem Verlust an Allgemeingültigkeit zu erreichen. So gilt das Vorgehensmodell für die Entwicklung von Software nur für Projekte, die das Design der Anwendungen auf Basis der UML durchführen eine konkrete technische Architektur zugrunde legen.

Es wurden Vorgehensmodelle für

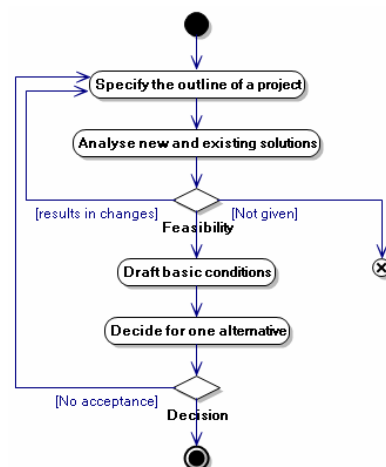
- die Untersuchung und Vorbereitung von Projektvorhaben,
- die Entwicklung von Anwendungen oder Integrationskomponenten und
- die Implementierung oder Migration von IT Infrastrukturen

entworfen. Sie sind als Web Site und ausführlicher in .pdf Dateien dokumentiert. Zu allen Ergebnistypen liegen Templates vor.

Vorgehensmodell zur Analyse von Projektvorhaben

Besonders komplexe oder umfangreiche IT Systeme werden in aller Regel nicht ohne Bewertung von Risiken und Betrachtung der Kosten und Nutzen gestartet.

Das Vorgehensmodell definiert die Prozesse für das Projektmanagement, das Configuration Management, die Sicherung der Qualität und die Vorbereitung des Vorhabens. Die Teilprozesse dieser Vorbereitung sind in der nebenstehenden Graphik zusammengefasst.



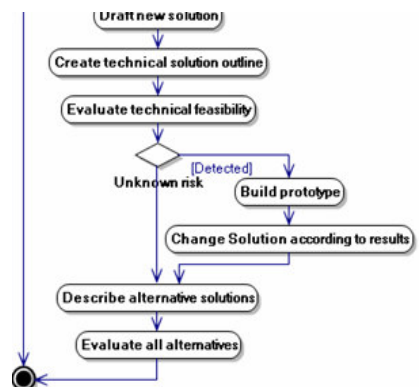
Aus den Aktivitäten resultieren Ergebnisdokumente oder Prototypen, die in eine Entscheidungsvorlage münden.

Es gibt etwa 15 Ergebnisse, zu denen z.B.

- Requirement Specification,
- Technical Keydata Matrix,
- Analysis Report,
- Solution Outline und
- Total Costs of Ownership

gehören. Alle Ergebnisse wurden in praxiserprobten Templates implementiert.

Besonderheiten ergeben sich in derartigen Analysen, wenn Probleme insbesondere in der technischen Machbarkeit vermutet werden. Dann müssen Analysen und Prototypen eingeplant werden.



Es ist häufig schwer, an dieser Stelle Risiken und Kosten im zukünftigen Projekt – nicht in der Voruntersuchung – sachlich gegeneinander abzuwägen.

Wird das Vorhaben positiv entschieden, dann wurde in dieser Phase das Realisierungsprojekt mit wichtigen Ergebnissen vorbereitet. Die wichtigsten Dokumente sind die Business Architecture, die Solution Outline und technische Daten wie Mengengerüste.

Vorgehensmodell zur Softwareentwicklung

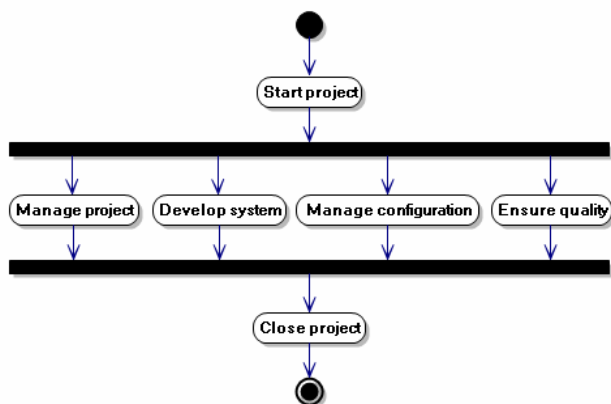
Das Vorgehensmodell der its.d&m zur Entwicklung von Software setzt voraus, dass der Leistungsumfang spezifiziert und das Vorhaben grob skizziert ist. Im besten Fall liegen aus einer Voruntersuchung die Business Architecture mit Anforderungsspezifikation, eine Skizze der technischen Lösung (Solution Outline), Mengengerüste, ein Projektplan und die Kostenrechnungen vor.

Das Vorgehensmodell setzt technische und methodische Rahmenbedingungen voraus. Dazu gehören:

- Das Design der Anwendungen erfolgt objektorientiert und auf Basis der UML Notation.

- Eine technische Architektur (Blueprint) definiert den Aufbau der Komponenten.
- Der technische Blueprint legt die Verwendungsweise der Entwicklungsumgebung fest.
- Die Skills der Personen beruhen in allen Rollen auf Informationstechnologie.

Viele Aspekte des Vorgehensmodells können besonders effizient verwendet werden, wenn Architektur und Entwicklungsumgebung auf .NET aufgebaut sind.



Die Prozesse sind in eine – oft unterbetonte – Startphase, den Entwicklungsprozess und die begleitenden Prozesse aufgeteilt (siehe folgende Abbildung).

Die Ergebnisse aus diesen Prozessen sind umfangreich und vor allem im Bereich Design und Build nach den Spezifika der eingesetzten Tools dokumentiert. Eine einfache Dokumentenstruktur ist nur in Teilen möglich.

Ein Überblick über die Ergebnistypen und deren Aufbau wird in weiterführenden Dokumenten (*Short Story Software Development Process.pdf* und *White Paper Software Development Process.pdf*) gegeben.

Die Charakteristik aktueller Entwicklungsprojekte, die moderne Entwicklungsumgebungen einsetzen, erzwingt weit reichende technische Qualifikationen aller Beteiligten. Das hat unter anderem zur Personalpolitik von its.d&m geführt.

Vorgehensmodell zur IT Infrastrukturentwicklung

Das Vorgehensmodell zur Entwicklung von IT Infrastrukturen wurde von einem der letzten Releases des Vorgehensmodells zur Softwareentwicklung abgeleitet. Der Anlass war ein sehr umfangreiches MS Windows Migrationsprojekt, während dessen Planung viele Analogien im Vorgehen beobachtet wurden.

Auch der Einsatz dieses Vorgehensmodells erfordert die Einhaltung spezifischer Randbedingungen. So wird ein technisches Architekturtemplate (Blueprint) vorausgesetzt, das Topologie und Konfiguration von IT Services auf eine für Unix und Windows typische Weise skizziert. Außerdem richten sich Design und Implementierung der Betriebsführung nach ITIL (*IT Infrastructure Library*).

Unterschiede ergeben sich natürlich im Entwicklungsprozess.

Ein Überblick über die Prozesse und Ergebnisse wird in weiterführenden Dokumenten (*Short Story IT Infrastructure Development Process.pdf* und *White Paper IT Infrastructure Development Process.pdf*) gegeben.

Architektur und Technologie

Der Blueprint der technischen Architektur von Anwendungen besteht aus einer Dokumentation zu

- Richtlinien und Konventionen,
- Nutzungskonzepten von speziellen Technologien wie z.B. ADO.NET oder SSPI,
- wichtigen Designkriterien,
- dem Modell der Layer
 - Design des Data Access Layer mit
 - Struktur der Komponenten und
 - Prinzipien in grundlegenden Abläufen
 - Design der weiteren Layer (Business Layer, Design des Presentation Layer, Design des Integration Layer, Design des System Management Layer und Design des Common Layer) nach ähnlichen Schemata.
- Architektur der IT Infrastruktur
 - Topologie und Konfiguration des Netzwerkes
 - Topologie und Konfiguration der Systemkomponenten (Server, Storage, Client, IT Services)
 - Aufbau, Integration und Konfiguration des System Management
- Test
 - Methoden und Tools
 - Grundlegende Testszenarien

- Spezifikation der Entwicklungsumgebung
 - Tools
 - Projektstruktur
 - Modellstruktur
 - Configuration Management
 - Change Management
 - Build Management

Ähnliche Ansätze ergaben sich im Bereich Security oder für das Testen der Anwendungen.

Eine stufenweise detaillierte Einführung in den Blueprint für die Entwicklung von Anwendungen wird in den Dokumenten *Short Story Software Architecture Blueprint.pdf* und *White Paper Software Architecture Blueprint.pdf* gegeben.

Das letzte Release des Blueprint wurde in Projekten eingesetzt und validiert, die Anwendungen auf Basis von .NET realisierten. Einige Patterns und vor allem Nutzungskonzepte lassen sich nur effizient verwenden, wenn der .NET Framework eingesetzt wird.

Einige Strukturen und Abläufe verwenden bestimmte Patterns und wurden in den letzten Projekten so stereotyp eingesetzt, dass Generatoren zu ihrer Implementierung erstellt wurden. Sie sind in das Visual Studio von Microsoft integriert. Ein Beispiel dafür ist der Data Access Layer, dessen Basisklassen hohe Anteile anwendungsneutraler Funktionen haben:

