

IT-Konzept

Projektname: Beispiel für ein IT-Konzept

Version: 1.00.002

Autor: Gerrit Radziwill, Final Matrix GmbH

Lizenztyp: GNU LGPL v3 oder höher

Syntax: <> = optionale Inhalte

Dokumenthistorie

Aufgabe	Termin	Autor	Status	Version
Menüstruktur erstellen	01.01.2011	Max Mustermann	erledigt	0.01.005
Inhalt anlegen	01.02.2011	Max Mustermann	erledigt	0.02.002
Überprüfung im Team	01.03.2011	Max Mustermann	erledigt	0.03.001
Überarbeitung Kapitel 'Daten'	01.04.2011	Max Mustermann	erledigt	0.04.006
Freigabe zur Abnahme	01.05.2011	Max Mustermann	erledigt	1.01.001
Übersetzung englisch	15.05.2011	Melanie Musterfrau	erledigt	1.02.004
Abnahme für Projektstart	01.06.2011	Markus Mustermanager	erledigt	1.02.005
Übersetzung in weitere Sprachen	offen	offen	offen	offen

Inhaltsverzeichnis

1. Gesamtüberblick	2
2. Basis	2
2.1. Überblick	2
2.2. Architektur	2
2.2.1. Anwendungsdiagramm	2
2.3. Voraussetzungen	3
2.3.1. Software	3
2.3.2. <Hardware>	3
2.3.3. <Daten>	3
2.4. Zielsetzungen	3
2.4.1. <Software>	3
2.4.2. <Performanz>	4
2.4.3. <Verfügbarkeit>	4
2.5. Ansprechpartner	4
2.5.1. <fachlich>	4
2.5.2. <technisch>	4

3.	Daten	4
3.1.	Überblick	4
3.2.	Architektur	4
3.2.1.	Datenflussdiagramm	5
3.3.	Datenentitäten	5
3.4.	<Datenintegrität>	5
3.5.	<Datenmodelle>	5
4.	Steuerung	6
5.	Verarbeitung	6
5.1.	Überblick	6
5.2.	Architektur	6
5.2.1.	< Zugriffsdiagramm >	6
5.2.2.	Klassenmodell	6
5.3.	Verarbeitungsentitäten	7
5.4.	<Datenplausibilität>	7
6.	Protokollierung	7
7.	Qualitätssicherung	7
7.1.	<Programmierrichtlinien>	7
7.2.	Test	7

1. Gesamtüberblick

Ziel des Projektes ist es, IT-Fachkräften mit wenig Erfahrung auf dem Gebiet der Konzepterstellung ein brauchbares Konzeptbeispiel für die ersten eigenen Konzeptstellungsaufgaben in die Hand zu geben.

2. Basis

Die Basis beschreibt die Bedingungen bzw. Voraussetzungen unter denen das Projekt durchgeführt werden soll. Sie ist so etwas wie eine grobe abstrakte Beschreibung der technischen Aufgaben und Ziele eines Projektes.

2.1. Überblick

Allgemeine Informationen oder Anforderungen.

Hier könnte man besondere allgemeine Bedingungen anführen.

Z.B. das alle verwendeten Programmiersprachen plattformunabhängig sein müssen, oder dass eine wöchentliche Iteration im Scrum Modus vorgesehen ist.

2.2. Architektur

Besondere Bemerkungen zur Architektur. Warum z.B. sollen alle Schnittstellen als Webservice realisiert werden.

2.2.1. Anwendungsdiagramm

Grafische Gesamtdarstellung aller beteiligten Systeme aus der hervorgehen sollte, welche Schnittstellen und Daten von und zu welchen Anwendungen / Servern / Instanzen / Abteilungen führen.

Mit Hilfe solcher Diagramme können z.B. noch unberücksichtigte Voraussetzungen oder Risiken von Seiteneffekten aufgedeckt werden.
Dabei richtet sich der Blick nicht nur auf das eigene Projekt sondern auch auf alle anderen beteiligten, existierenden oder möglicherweise noch in Entwicklungsphasen befindlichen Systeme.

Stichpunkte für den Diagramminhalt:

Daten
Anwendungen
<Schnittstellen>
<Fremdsysteme>

2.3. Voraussetzungen

Beschreibung allgemeiner Annahmen wie z.B. dass für alle notwendigen administrativen Aufgaben der Dienstleister XYZ werktätlich zwischen 8 und 18 Uhr mit Servicelevel 2 zur Verfügung steht.

2.3.1. Software

Welche Software muss auf welchen Systemen vorhanden sein.

Anwendungen
<Versionen>

2.3.2. <Hardware>

Beispiele:

Es werden 2 Server mit je 16 CPUs benötigt.

Es werden 5 Entwicklerstationen benötigt.

Die Netzwerkanbindung zum Standort XY muss auf n GB erhöht werden.

2.3.3. <Daten>

Beispiele:

Es werden Produktivdaten mit den Datenständen von der Jahresendverarbeitung und den letzten drei Monatsultimos benötigt.

Es soll eine Datenqualitätsanhebung in der Erfassung durchgeführt werden, welche zukünftig doppelte Lieferanten ausschließt.

2.4. Zielsetzungen

Beschreibung allgemeiner Ziele wie z.B. dass der Schwerpunkt auf Wartbarkeit oder auf Sicherheit zu setzen ist.

2.4.1. <Software>

Beispiele:

Das Webinterface soll auf FireFox V6 und IE V8 lauffähig sein.

Die Datenschnittstellen sollen wo immer möglich XML Strukturen verwenden.

Die Anwendungen müssen auf Windows XP als auch Windows 7 laufen.

2.4.2. <Performanz>

Beispiele:

Die Webanwendung darf eine maximale Zugriffszeit von 5 Sekunden bei 100 parallelen Nutzern nicht überschreiten.

Die Anwendung muss alle Szenarien innerhalb von 2 Stunden vollständig durchrechnen können.

2.4.3. <Verfügbarkeit>

Beispiele:

Die Anwendung wird einmal monatlich benötigt und muss am jeweiligen Monatsultimo innerhalb von zwei Tagen mit allen Daten verfügbar sein.

Die Anwendung muss parallel mehrfach redundant lauffähig sein und ihre Instanzen müssen die Daten eigenständig synchronisieren können.

2.5. Ansprechpartner

2.5.1. <fachlich>

<Konzeption>

<Organisation>

2.5.2. <technisch>

<Administration>

<Entwicklung>

3. Daten

Eine allgemeine Beschreibung wie:

Mit Hilfe der Anwendung werden täglich Kundendaten aufgenommen, in Berichtsform zur Verfügung gestellt sowie per Nachtlauft strukturiert an andere Zielsysteme weitergegeben.

3.1. Überblick

Allgemeine Informationen oder Anforderungen.

Hier können Informationen z.B. zur Berücksichtigung der schlechten Datenqualität oder zu besonderen Datenquellen beschrieben werden.

Weitere Beispiele:

Alle Zeichenketten sollen stets im Unicode verwendet werden.

Die Genauigkeit der Nachkommastellen muss mindestens 8 betragen.

Länderspezifische Datums- oder Zeitwerte sollen vor jeglicher Verarbeitung jeweils in ein internes Datumszeitformat gewandelt werden.

3.2. Architektur

Hier könnte eine zusätzliche Beschreibung der gewünschten Architekturansätze stehen.

3.2.1. Datenflussdiagramm

In einem solchen Diagramm sollten die folgenden Themen Berücksichtigung finden:

- <Konfiguration>
- <Datenbanken>
- <Textdateien>
- <Datenströme>
- <Anwenderinteraktion>
- <Anwendungen>

3.3. Datenentitäten

Hier können die Anforderungen an die einzelnen Datenstrukturen und -pakete, welche im Diagramm unter 3.2.1 bereits aufgezeigt wurden, detailliert beschrieben werden.

Beispiel:

Die CSV-Tabelle XY muss jeden Tag um 18 Uhr in Ordner XYZ abgelegt werden.

Was

- Datentypen
- Datenformate
- <Datenbereiche>

Woher

- <Siehe 3.2.1.>

Wohin

- <Siehe 3.2.1.>

Wann

- <Aktualisierungen>
- <Intervalle>

3.4. <Datenintegrität>

Beispiel:

Es dürfen nur Marktdaten verarbeitet werden, zu denen auch eine Währung hinterlegt ist.

- <Annahmen>

- <Prüfungen>

3.5. <Datenmodelle>

Hier sollten z.B. die Datenmodellierungen zu verwendeten Datenbanken per Skript, Diagramm und oder Referenz auf externe Dateien hinterlegt werden.

Aber auch ein paar Exceltabellen oder z.B. XML-Dateien sind es gelegentlich wert hier ordentlich modelliert zu werden.

4. Steuerung

Hier soll beschrieben werden, wie, von wem und wann genau eine Anwendung aufgerufen werden soll. Eine einfache Rechteverwaltung kann hier mit beschrieben werden. Für eine komplexere Rechteverwaltung ist eher ein eigenes Kapitel empfehlenswert.

Was

- <Anwender>
 - <Berechtigungen>
- <Anwendungen>
 - <Berechtigungen>
- <Services>
 - <Berechtigungen>

Wann

- <Termine>
- <Intervalle>

Wie

- <Oberfläche>
- <Parameter>

5. Verarbeitung

Beschreibung allgemeiner Anforderungen an die Verarbeitung wie z.B. dass der Schwerpunkt auf Wartbarkeit oder auf Sicherheit zu setzen ist.

5.1. Überblick

Allgemeine Informationen oder Anforderungen.

Beispiele:

Alle Anwendungsschnittstellen dürfen ausschließlich Datentypen verwenden, welche auch direkt mit J2EE ansprechbar bzw. verwendbar sind.

5.2. Architektur

Beschreibung von Besonderheiten der Anwendungsarchitektur.

5.2.1. < Zugriffsdiagramm >

<Anwendungsschichten>

Hier ein Beispiel unter http://de.wikipedia.org/wiki/Model_View_Controller

5.2.2. Klassenmodell

Ähnlich eines Datenmodells werden hier die Klassen bzw. Module einer Anwendung beschrieben mit Hilfe von

<Klassen- bzw. Moduldiagramm>

oder auch noch etwas detaillierter in einem

<Funktions- bzw. Prozedurdiagramm>

5.3. Verarbeitungsentitäten

Hier können einzelne Strukturen aus den Diagrammen unter 5.2.2 gesondert detaillierter beschrieben werden.

Was

- <Module>
- <Klassen>
- <Funktionen>
- <Prozeduren>

Wo

- <Dateien>
- <Netzwerk>

Wann

- Startverhalten

5.4. <Datenplausibilität>

Beispiel:

Es dürfen nur Marktdaten verarbeitet werden, deren Abweichung zum Vortag nicht mehr als 10% beträgt.

<Annahmen>

<Prüfungen>

6. Protokollierung

Hier sollte möglichst präzise die benötigte Protokollierung beschrieben werden. Dieses Kapitel wird häufig außer Acht gelassen, ist aber für die zukünftige Wartung ein absolutes Muss!

technische Protokolle

fachliche Protokolle

7. Qualitätssicherung

Allgemeine Informationen oder Anforderungen.

7.1. <Programmierrichtlinien>

7.2. Test

Hier sollten idealer Weise alle technischen Testfälle beschrieben werden. Auch wenn dies vornehmlich technische Szenarien darstellt sollte dieser Teil auch vom Fachbereich einmal kurz überflogen werden.

Diese ausführlichen Beschreibungen führen übrigens oft zu deutlich mehr Klarheit in der Testphase und dienen auch zukünftigen Anwendungserweiterungen als gute Basis.