Begründeter Lösungsansatz

Autoren :Denis Bittante

Version : 0.1

# Versionen

|  |  |  |  |
| --- | --- | --- | --- |
| Autor | Datum | Version | Zusammenfassend |
| Denis Bittante | 27.10.15 | 0.1 | BuildTool  Datenbankzugriff  Datenbank |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Inhaltsverzeichnis

[Java Entwicklung: Build-Tool 2](#_Toc433739329)

[Situation analysieren 2](#_Toc433739330)

[Ziele formulieren 2](#_Toc433739331)

[Lösungsalternativen suchen 2](#_Toc433739332)

[Lösung auswählen 2](#_Toc433739333)

[Vorteile Maven 2](#_Toc433739334)

[Nachteile Maven 2](#_Toc433739335)

[Vorteile Ant 2](#_Toc433739336)

[Nachteile Ant 2](#_Toc433739337)

[Vorteile Eclipse 2](#_Toc433739338)

[Nachteile Eclipse 2](#_Toc433739339)

[Umsetzung der Auswahl planen 3](#_Toc433739340)

[Kontrolle 3](#_Toc433739341)

[Datenbankzugriff 3](#_Toc433739342)

[Situation analysieren 3](#_Toc433739343)

[Ziele formulieren 3](#_Toc433739344)

[Lösungsalternativen suchen 3](#_Toc433739345)

[Lösung auswählen 3](#_Toc433739346)

[Vorteile Einsatz JPA 3](#_Toc433739347)

[Nachteile Einsatz JPA 3](#_Toc433739348)

[Umsetzung der Auswahl planen 4](#_Toc433739349)

[Kontrolle 4](#_Toc433739350)

[DBMS 4](#_Toc433739351)

[Situation analysieren 4](#_Toc433739352)

[Ziele formulieren 4](#_Toc433739353)

[Lösungsalternativen suchen 4](#_Toc433739354)

[Lösung auswählen 4](#_Toc433739355)

[Vorteile MySQL 4](#_Toc433739356)

[Nachteile MySQL 4](#_Toc433739357)

[Vorteile MongoDB 4](#_Toc433739358)

[Nachteile MongoDB 4](#_Toc433739359)

[Umsetzung der Auswahl planen 4](#_Toc433739360)

[Kontrolle 4](#_Toc433739361)

# Java Entwicklung: Build-Tool

## Situation analysieren

Der Knowhow bezüglich Java-Entwicklung ist bei den Teammitgliedern verschieden Tief vorhanden. Der Einsatz eines Build-Tools wird hier festgelegt. Da die Technologie festgesetzt ist werden alle Teammitglieder Eclipse im Einsatz haben und das ist die Basis für diese Evaluation.

## Ziele formulieren

Die Entwicklung soll möglichst effizient und standardisiert gestaltet werden. Der Code soll bei jedem Mitglied gebildet und möglichst einfach erweitert werden können. Wichtig ist das Projekt modular und gekapselt aufbauen zu können.

## Lösungsalternativen suchen

* Maven
* Ant
* Eclipse – Manuelle Installation der Libraries

## Lösung auswählen

### Vorteile Maven

* Eclipse Integration
* Standard bezüglich Software Gliederung und Paketierung
* Dependencies sind definiert und werden von Maven gehandhabt
* Grosse Community
* Jenkins Integration, einfache Installation von CI
* Maven besitzt einen LifeCycle

### Nachteile Maven

* Einarbeitungszeit

### Vorteile Ant

* Ist eine Skriptsprache, könnte alles erstellen
* Sehr ausgereiftes Produkt

### Nachteile Ant

* Keine Vorgabe bezüglich Standard Projektstruktur
* Projektstruktur müssen im Projekt selber definiert und im Build-Prozess propagiert werden
* Ant ist Prozedural aufgebaut darum muss jeder Buildschritt selber definiert werden.
* Ant hat keinen Lifecycle man muss Goal und Goal-Abhöngkeiten selber definieren.

### Vorteile Eclipse

* Wird mitgeliefert

### Nachteile Eclipse

* Veränderungen am Build-Prozess können nicht leicht festgestellt werden damit ist nicht gewährleistet dass jeder Mitarbeitet das Softwarepaket gleich bildet.

## Umsetzung der Auswahl planen

Die Umsetzung beginnt beim Erstellen des ersten Projektes. Die Struktur soll wie Maven sie Vorgibt eingehalten werden und liegt in der Verantwortung jeden Entwicklers diese weiter zu pflegen, der Nachteil der Einarbeitungszeit die bei Ant ebenfalls eintreten würde wir zu gunsten der Modularisierung und entsprechenden Erweiterbarkeit in Kauf genommen.

## Kontrolle

Während der Umsetzung wird de Build-Prozess immer wieder kontrolliert. Das Erstellen der Software soll mittels Maven vereinfacht werden und der richtige Umgang damit, soll in Peer-Programming weiter vermittelt und angelernt werden.

# Datenbankzugriff

## Situation analysieren

Die Applikation soll Daten in einer Datenbank persistieren und von dieser wieder lesen und verändern können. Dazu soll ein Backend gebaut werden welches sich ausschliesslich mit der Businesslogik und Persistenz auseinandersetzt. Welche Art von Zugriff ist noch nicht definiert und soll im Folgenden ausgearbeitet werden

## Ziele formulieren

Der Zugriff auf die Datenbank soll modular und mit möglichst kleiner Schreibarbeit durgeführt werden können. Die Daten sollen Datenbankunabhängig persistiert werden. Datenbankmanagementsystem soll wenn nötig ausgetauscht werden können.

## Lösungsalternativen suchen

* JPA (Java Persistence API) – ORM (object-relational mapping)
  + Implementation : EclipseLink
* JDBC (java.sql) direkter Zugriff auf die Datenbank

## Lösung auswählen

### Vorteile Einsatz JPA

* POJOs werden als Transportobjekte einmalig definiert
* CRUD Operation müssen nicht geschrieben werden
* Zugriff auf die Datenbank findet über EclipseLink statt dass bedeutet die Datenbank ist austauschbar
* Aufwand ist einmalig für den Zugriff auf die Datenbank aufzuwenden und weitere Ressourcen kann auf die Businesslogik verwendet werden.
* Transaktionshandling wird vom Framework organisiert
* SQL-Injection ist bei generierte Abfragen etwas schwieriger und bringt darum etwas mehr Sicherheit mit sich

### Nachteile Einsatz JPA

* Einsatz von OneToMany oder ManyToOne birgt die Gefahr vom Laden gesamter Datenbestände
* Einarbeitung der Mitarbeiter

Die Vorteile die JPA mit sich bringt überwiegen dessen Nachteile es wird die Entwicklung mit JPA angestrebt. Die Vorteile von direkten Zugriffen über JDBC ist hier nicht mehr aufzuführen, denn die Entwicklungszeit soll nicht darin bestehen SQL-Statements zu schreiben. Die Entscheidung fällt zugunsten ORM auch durch dessen Kompatibilität zu den verschiedenen Datenbankmanagement Systemen.

## Umsetzung der Auswahl planen

Die Entwicklung wird mit JPA gestartet und bei der Anbindung der Datenbank wird diese Technologie federführend sein**.** Wie die Transaktionen gehandelt werden ob über Spring Application Context Configuration (application.xml) wird noch nicht definiert je nach Einarbeitungsstadium kann eine solche Alternative in Betracht genommen werden. Die Umsetzung soll möglichst einfach gehandhabt werden darum werden weitere Frameworks wie Hibernate hier nicht auch als Alternativen aufgenommen.

## Kontrolle

Mittels JUnit-Tests sollen die CRUD-Operationen ausgetestet werden.Die korrekte Nutzung wird mittels Reviews und Peer-Programming permanent überprüft.

# DBMS

## Situation analysieren

Es werden primär drei DBMS zur Diskussion gestellt:

MySQL, Oracle und MongoDB eine NoSQL die dokumentorientiert arbeitet.

## Ziele formulieren

Das Ziel ist ein einfaches Datenbankmanagement System zu wählen, dass die Entwicklung vorantreibt und wenige Eigenheiten verbirgt. Das System soll bekannt sein und eine grosse Community für Fragen bieten. Der Aufwand für Installation und Konfiguration soll möglichst gering sein. Grosse Flexibilität und Skalierbarkeit steht in dieser Phase des Projekts nicht im Vordergrund.

## Lösungsalternativen suchen

* MySQL Relationale Datenbank
* Oracle Datenbank
* NoSQL NoSQL Datenbank

## Lösung auswählen

### Vorteile MySQL

* Bekanntheitsgrad bei den Mitarbeiter
* Einfache Installation
* Populärste OpenSource Datenbank der Welt
* Schlanke Datenbank
* Guter Support
* Sehr stabil
* JPA kompatibel

### Nachteile MySQL

* Nicht so viele Features wie Oracle

### Vorteile Oracle

* viele Feature darum Mächtig

### Nachteile Oracle

* Lizenz nötig für kommerziellen Betrieb (Developer free)
* Sehr großer RAM-Verbrauch
* Einarbeitungszeit für Anfänger

### Vorteile MongoDB

* Hohe Leistung
* Hohe Flexibilität
* Geringe Komplexität

### Nachteile MongoDB

* Geringe Verbreitung
* Schwieriges joining der Tabellen
* Kein Support für JPA
* Einarbeitungszeit

## Umsetzung der Auswahl planen

MongoDB scheidet leider aus das sie nicht ORM kompatibel ist.

Die Umsetzung wird mit JPA angetrieben darum sind bei den Alternativen MySQL und Oracle nach den Kriterien: Lizenz, Installation und Einarbeitungszeit gewertet.

MySQl schneidet dabei etwas besser ab, da deren Lizenzkosten auch für Profitorientierte Unternehmen immer frei sind. Die Installation ist ebenfalls einfach und die Einarbeitungszeit wird sich etwa im gleichen Rahmen halten wie die von Oracle.

MySQL wird von Anfang an als DBMS eingebunden.

## Kontrolle

Es wird von Beginn an kontrolliert dass die DBMS stabil läuft. Die Entwicklung wird mit MySQL durchgeführt und in Tests integriert.