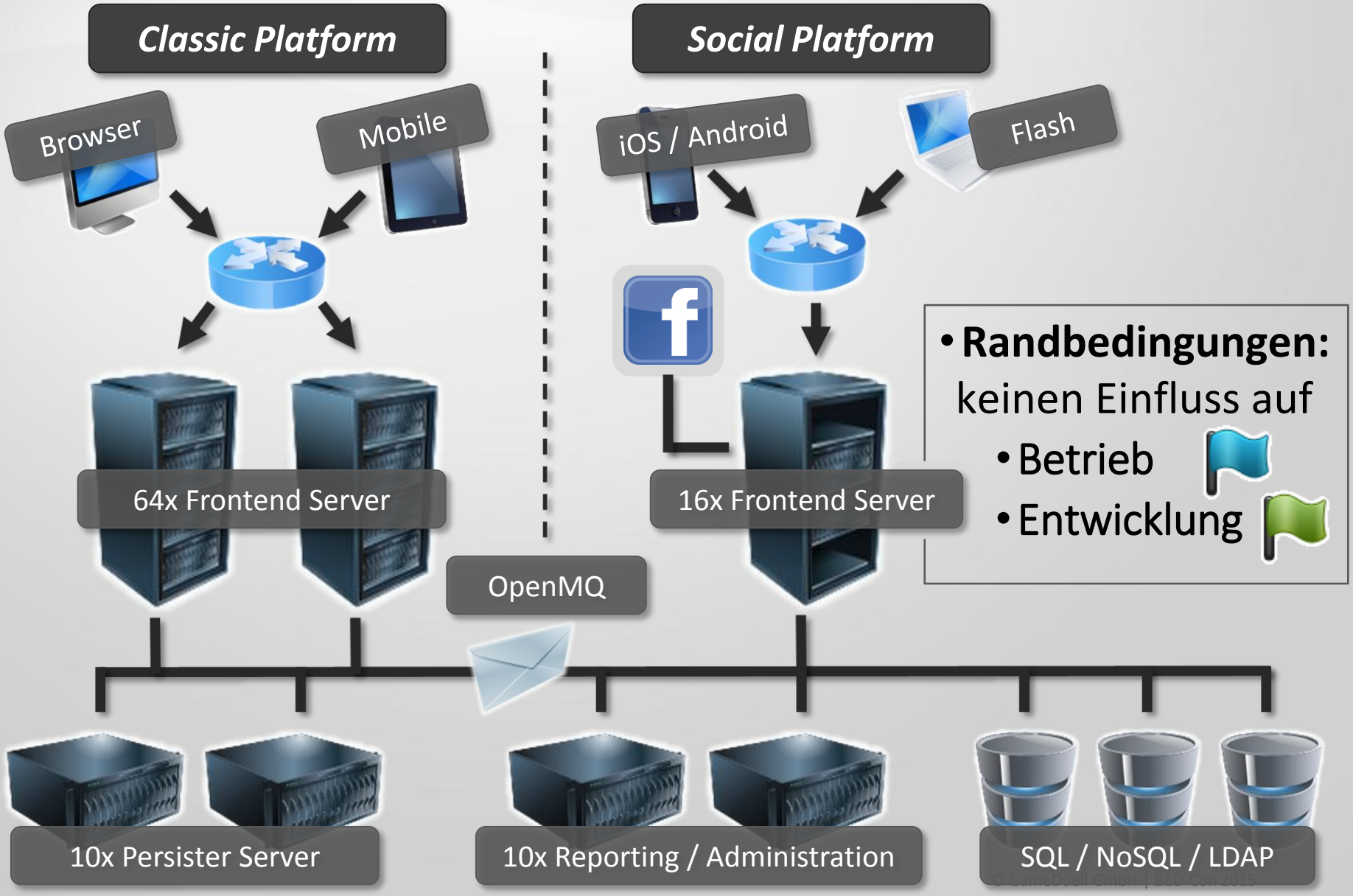
GameDuell

# *Operation am offenen Herzen*

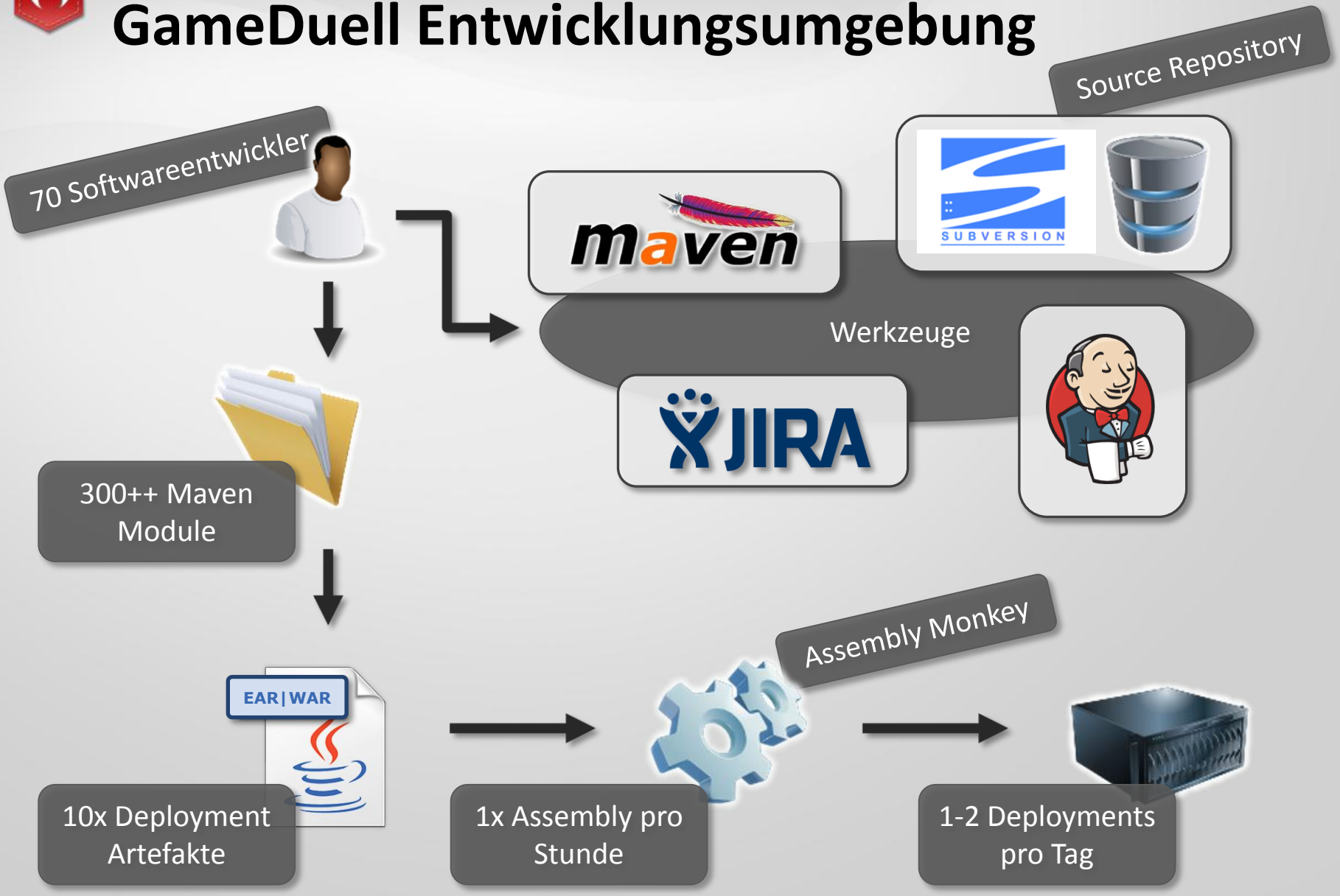
Case Study zur erfolgreichen JEE-7 Migration

Dirk Ehms, GameDuell GmbH

# GameDuell Plattform Topologie



# GameDuell Entwicklungsumgebung



# Schritte zum Erfolg

1. **Maven Dependency Management, Maven Profile**
2. VCS Branches für nicht-kompatiblen Source Code
3. Continuous Integration
4. Continuous Delivery / Release Pipeline
5. Rolling Deployment / Inkrementeller Rollout
6. Laufzeitverhalten überwachen
7. Aufräumarbeiten

# Dependency Management: Module POM

```
<project>
    <parent>
        <groupId>de.gameduell</groupId>
        <artifactId>jee-parent</artifactId>
        <version>2.8.0</version>
    </parent>
    ...
    <dependencies>
        ...
        <dependency>
            <groupId>javax</groupId>
            <artifactId>javaee-api</artifactId>
            <version>6.0</version>
            <scope>provided</scope>
        </dependency>
    </dependencies>
</project>
```

```
$ mvn clean install
```

```
$ mvn clean install -P jee7
```

# Dependency Management: Parent POM

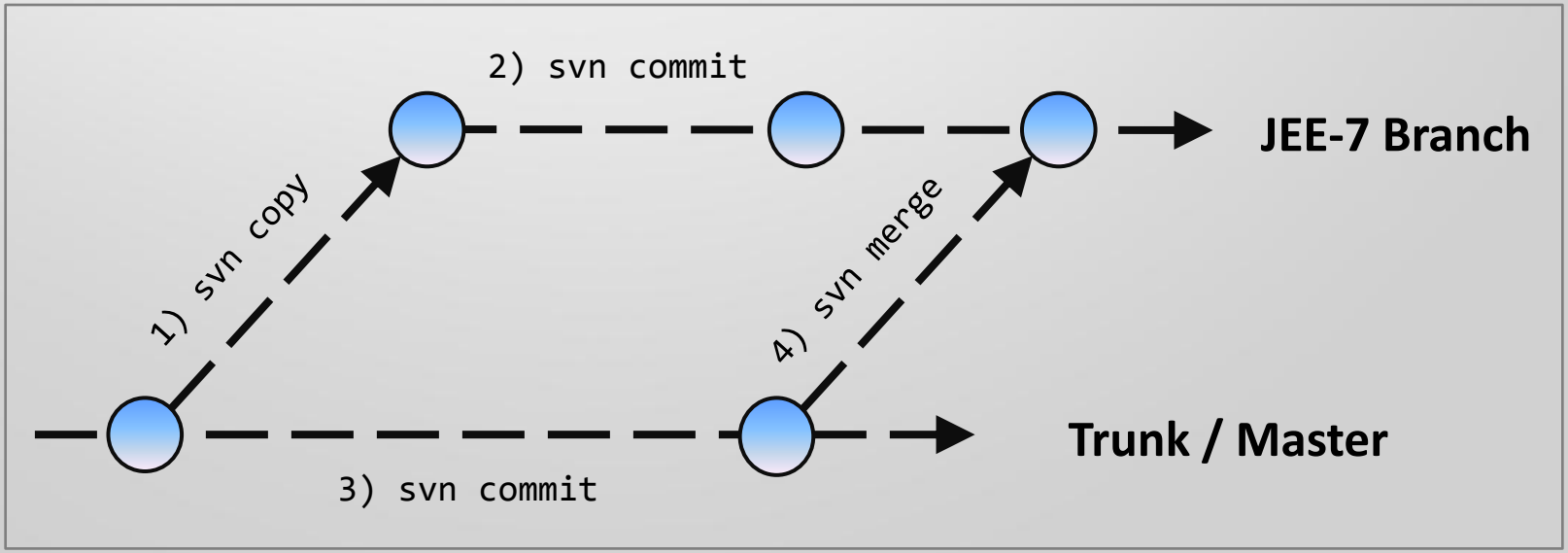
```
<project>
    ...
    <profiles>
        <profile>
            <id>jee6</id>
            <activation>
                <activeByDefault>true</activeByDefault>
            </activation>
            <dependencyManagement>
                <dependencies>
                    <dependency>
                        <groupId>javax</groupId>
                        <artifactId>javaee-api</artifactId>
                        <version>6.0</version>
                        <scope>provided</scope>
                    </dependency>
                </dependencies>
            </dependencyManagement>
        </profile>
        <profile>
            <id>jee7</id>
            <dependencyManagement>
                <dependencies>
                    <dependency>
                        <groupId>javax</groupId>
                        <artifactId>javaee-api</artifactId>
                        <version>7.0</version>
                        <scope>provided</scope>
                    </dependency>
                </dependencies>
            </dependencyManagement>
        </profile>
       ...
```

# Schritte zum Erfolg

1. Maven Dependency Management, Maven Profile
2. **VCS Branches für nicht-kompatiblen Source Code**
3. Continuous Integration
4. Continuous Delivery / Release Pipeline
5. Rolling Deployment / Inkrementeller Rollout
6. Laufzeitverhalten überwachen
7. Aufräumarbeiten

# Source Code Branches

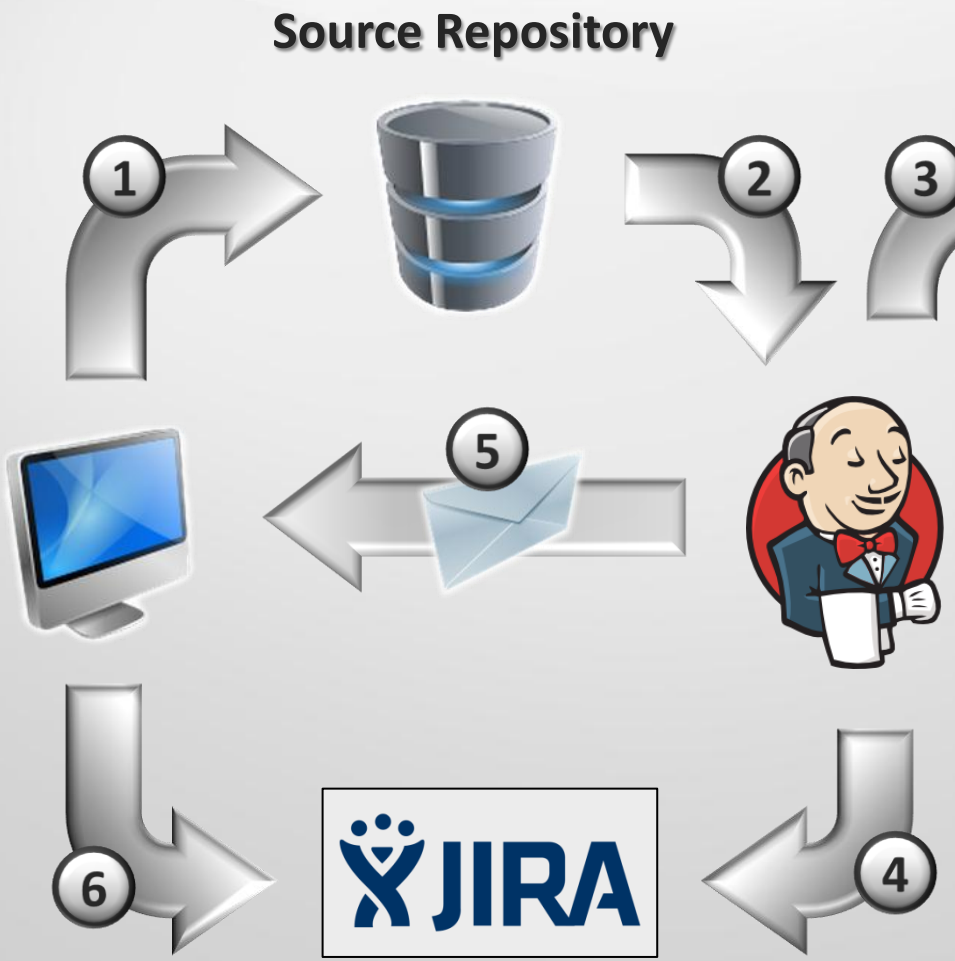
1. Anlegen einer Kopie vom Trunk / Master
2. Notwendige JEE-7 Anpassungen durchführen
3. Weiterentwicklung erfolgt wie bisher auf dem Trunk
4. Änderungen automatisch zusammenführen



# Schritte zum Erfolg

1. Maven Dependency Management, Maven Profile
2. VCS Branches für nicht-kompatiblen Source Code
3. **Continuous Integration**
4. Continuous Delivery / Release Pipeline
5. Rolling Deployment / Inkrementeller Rollout
6. Laufzeitverhalten überwachen
7. Aufräumarbeiten

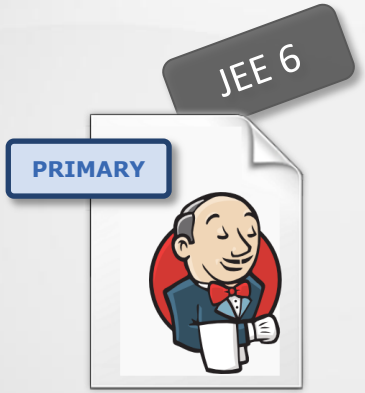
# GameDuell Build Pipeline




1. Check-in
2. Trigger Jenkins
3. Artifact Deploy
4. Update JIRA Ticket
5. Success eMail
6. Status: Integrated

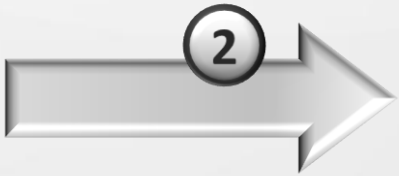
```
$ svn ci –m “XYZ-4711, … //minor”
```

# Jenkins konfigurieren (Trunk)

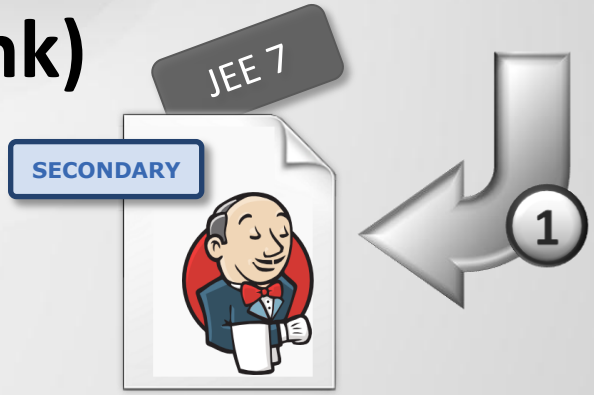



Jenkins Job


social-services-2.9.1.jar




```
$ mvn clean verify -P jee7
```

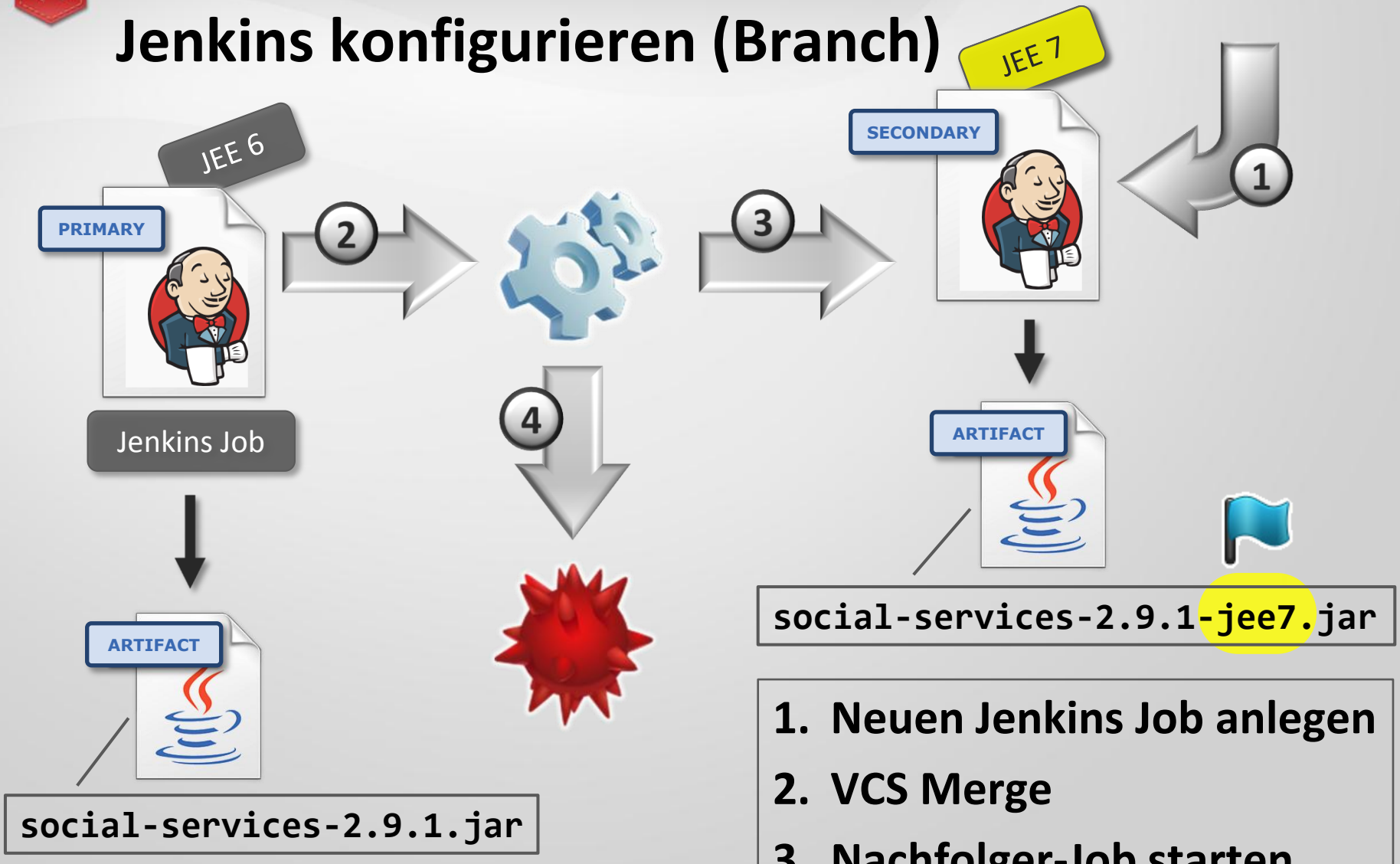




social-services-2.9.1.jar

1. Neuen Jenkins Job anlegen
2. Nachfolger-Job starten

# Jenkins konfigurieren (Branch)

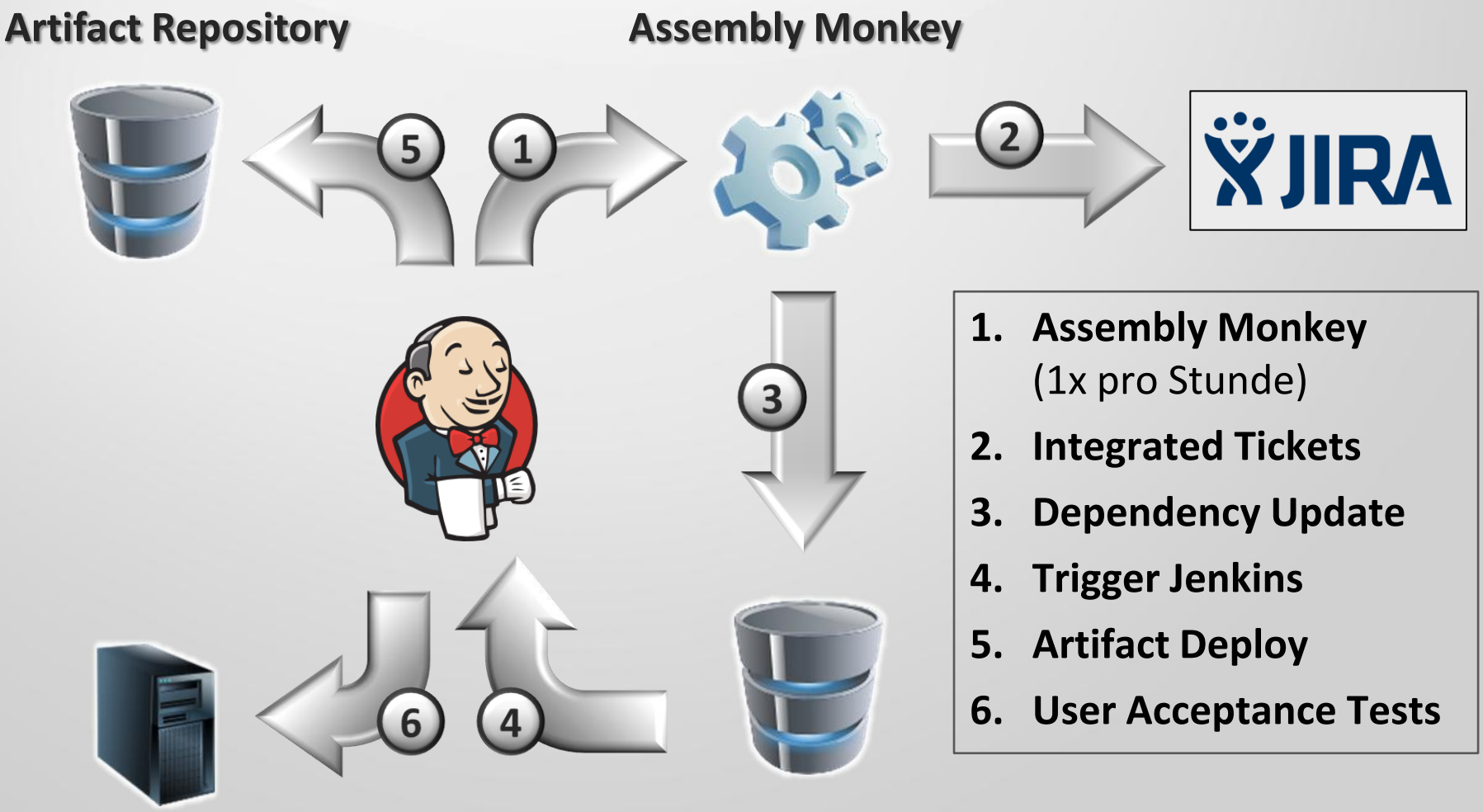



1. **Neuen Jenkins Job anlegen**
2. **VCS Merge**
3. **Nachfolger-Job starten**
4. **Merge Konflikt** (optional)

# Schritte zum Erfolg

1. Maven Dependency Management, Maven Profile
2. VCS Branches für nicht-kompatiblen Source Code
3. Continuous Integration
4. **Continuous Delivery / Release Pipeline**
5. Rolling Deployment / Inkrementeller Rollout
6. Laufzeitverhalten überwachen
7. Aufräumarbeiten

# GameDuell Release Pipeline

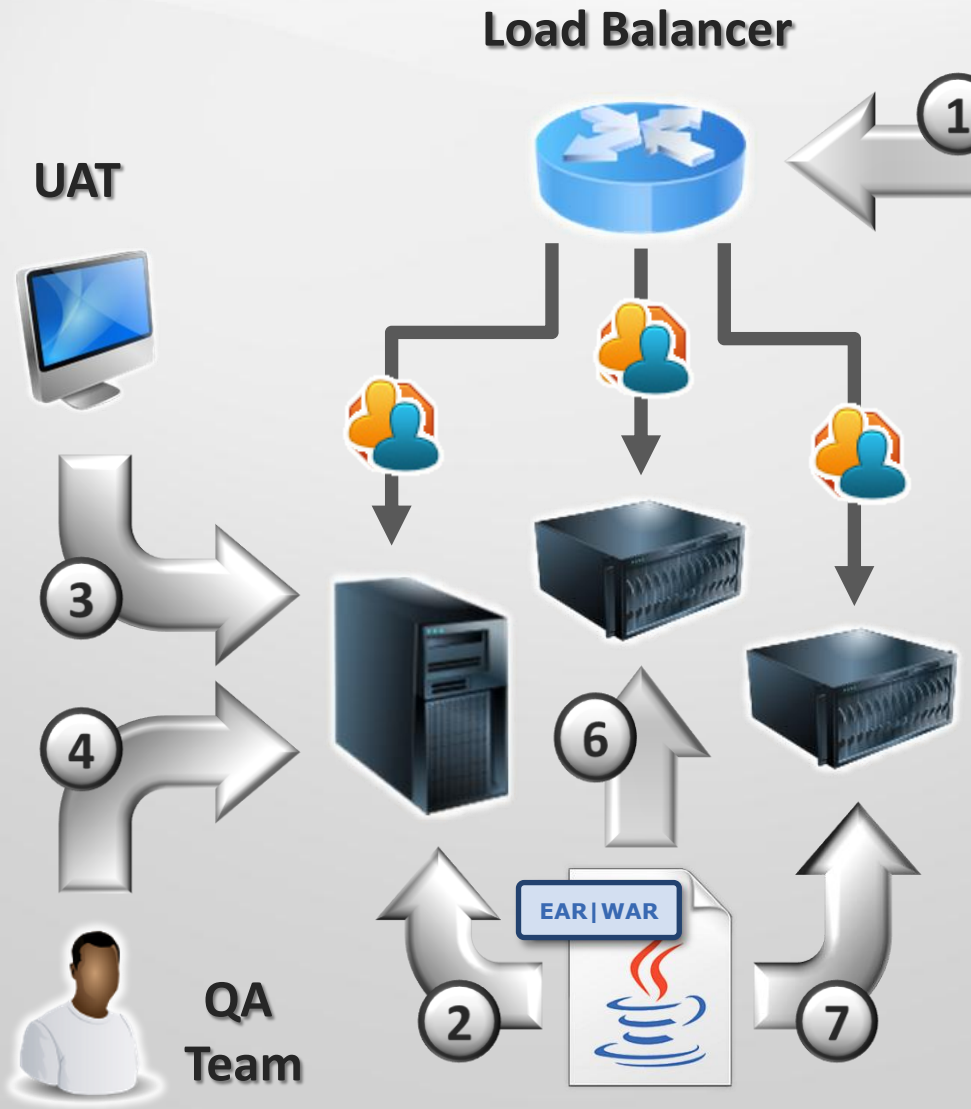



1. **Assembly Monkey** (1x pro Stunde)
2. **Integrated Tickets**
3. **Dependency Update**
4. **Trigger Jenkins**
5. **Artifact Deploy**
6. **User Acceptance Tests**

# Schritte zum Erfolg

1. Maven Dependency Management, Maven Profile
2. VCS Branches für nicht-kompatiblen Source Code
3. Continuous Integration
4. Continuous Delivery / Release Pipeline
5. **Rolling Deployment / Inkrementeller Rollout**
6. Laufzeitverhalten überwachen
7. Aufräumarbeiten

# Rolling Deployment / Zero Downtime




1. 50% der Frontends auf Standby
2. Deployment auf einem Frontend
3. User Acceptance Tests
4. Exploratives Testen
5. “Echte” User auf das Frontend
6. Rollout fortsetzen
7. Vollständiger Rollout

# Inkrementeller Rollout

GD Release Watch

| HOST | Java | GLASSFISH | APP-EAR | SOCIAL-WEBAPP | GAMESERVER-EAR | STDBY | SOCIAL | MPGC | USER | GC CONN |
|---|---|---|---|---|---|---|---|---|---|---|
| socialfe-1 | 1.7.0_71 | 4.1_b13 | 6.25.1 | 3.31.1 | 4.3.6 | - | active | 1,2,3,4 | 865 | 185 |
| socialfe-2 | 1.7.0_71 | 4.1_b13 | 6.25.1 | 3.31.1 | 4.3.6 | - | off | 1,2,3,4 | 0 | 0 |
| socialfe-3 | 1.7.0_71 | 4.1_b13 | 6.25.1 | 3.31.1 | 4.3.6 | - | off | 1,2,3,4 | 0 | 0 |
| socialfe-4 | 1.7.0_71 | 4.1_b13 | 6.25.1 | 3.31.1 | 4.3.6 | - | off | 1,2,3,4 | 0 | 0 |
| socialfe-5 | 1.7.0_71 | 4.1_b13 | 6.25.1 | 3.31.1 | 4.3.6 | - | off | 1,2,3,4 | 0 | 0 |
| socialfe-6 | 1.7.0_71 | 4.1_b13 | 6.25.1 | 3.31.1 | 4.3.6 | - | off | 1,2,3,4 | 0 | 0 |
| socialfe-7 | 1.7.0_71 | 4.1_b13 | 6.25.1 | 3.31.1 | 4.3.6 | - | off | 1,2,3,4 | 0 | 0 |
| socialfe-8 | 1.7.0_71 | 4.1_b13 | 6.25.1 | 3.31.1 | 4.3.6 | - | off | 1,2,3,4 | 0 | 0 |
| socialfe-9 | 1.7.0_71 | 4.1_b13 | 6.24.0 | 3.30.1 | 4.3.4 | - | active | 1,2,3,4 | 1098 | 346 |
| socialfe-10 | 1.7.0_71 | 4.1_b13 | 6.24.0 | 3.30.1 | 4.3.4 | - | active | 1,2,3,4 | 1082 | 329 |
| socialfe-11 | 1.7.0_71 | 4.1_b13 | 6.24.0 | 3.30.1 | 4.3.4 | enabled | standby | 1,2,3,4 | 4 | 0 |
| socialfe-12 | 1.7.0_71 | 4.1_b13 | 6.24.0 | 3.30.1 | 4.3.4 | - | active | 1,2,3,4 | 1083 | 326 |
| socialfe-13 | 1.7.0_71 | 4.1_b13 | 6.24.0 | 3.30.1 | 4.3.4 | - | active | 1,2,3,4 | 1042 | 329 |
| socialfe-14 | 1.7.0_71 | 4.1_b13 | 6.24.0 | 3.30.1 | 4.3.4 | - | active | 1,2,3,4 | 1078 | 330 |
| socialfe-15 | 1.7.0_71 | 4.1_b13 | 6.24.0 | 3.30.1 | 4.3.4 | - | active | 1,2,3,4 | 1074 | 350 |
| socialfe-16 | 1.7.0_71 | 4.1_b13 | 6.24.0 | 3.30.1 | 4.3.4 | - | active | 1,2,3,4 | 1111 | 366 |

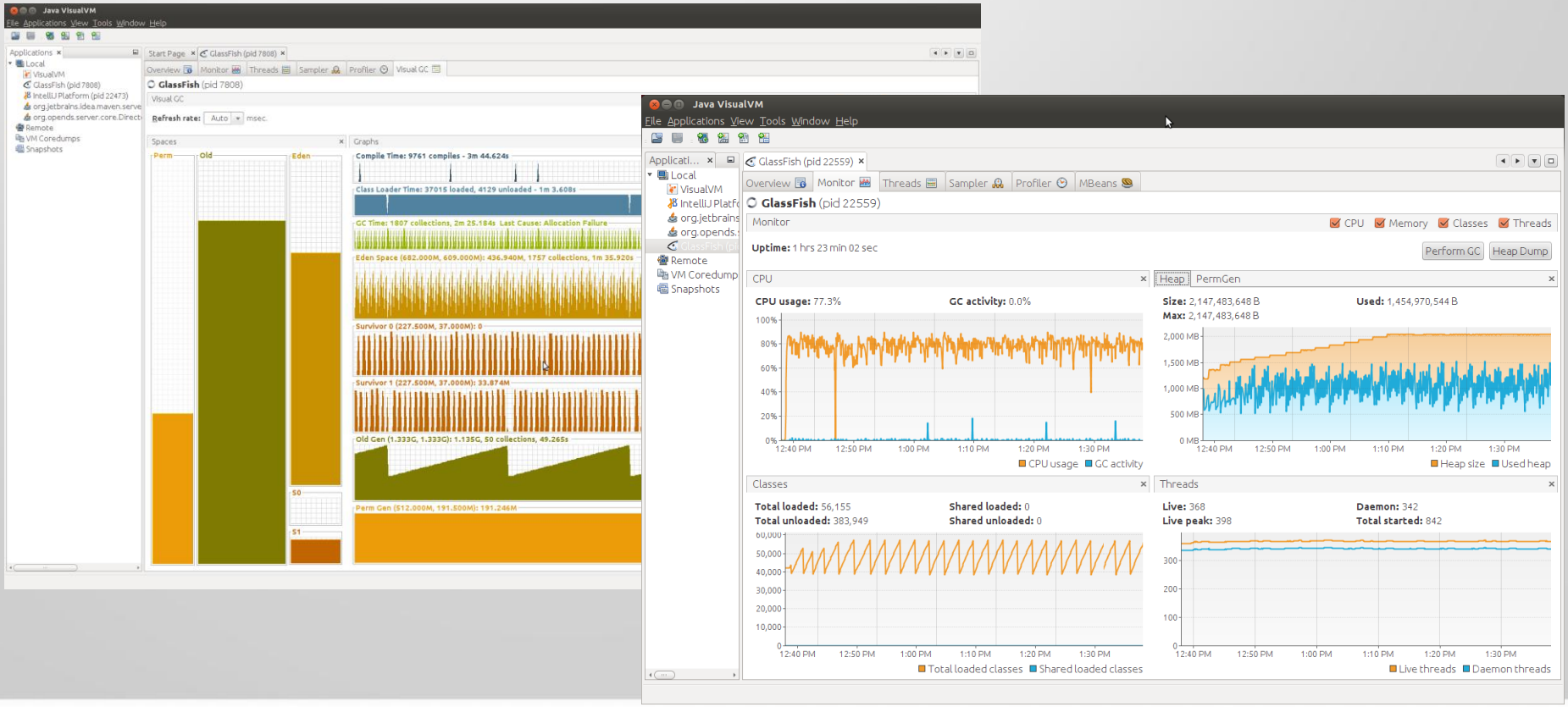
4.1_b13

# Schritte zum Erfolg

1. Maven Dependency Management, Maven Profile
2. VCS Branches für nicht-kompatiblen Source Code
3. Continuous Integration
4. Continuous Delivery / Release Pipeline
5. Rolling Deployment / Inkrementeller Rollout
6. **Laufzeitverhalten überwachen**
7. Aufräumarbeiten

# Laufzeitverhalten überwachen

- **Vergleich zwischen GF3 und GF4** (Speicherverbrauch, CPU, Threads, GC)
- **Tools: JConsole, Java VisualVM, Java Mission Control, Java Flight Recoder**

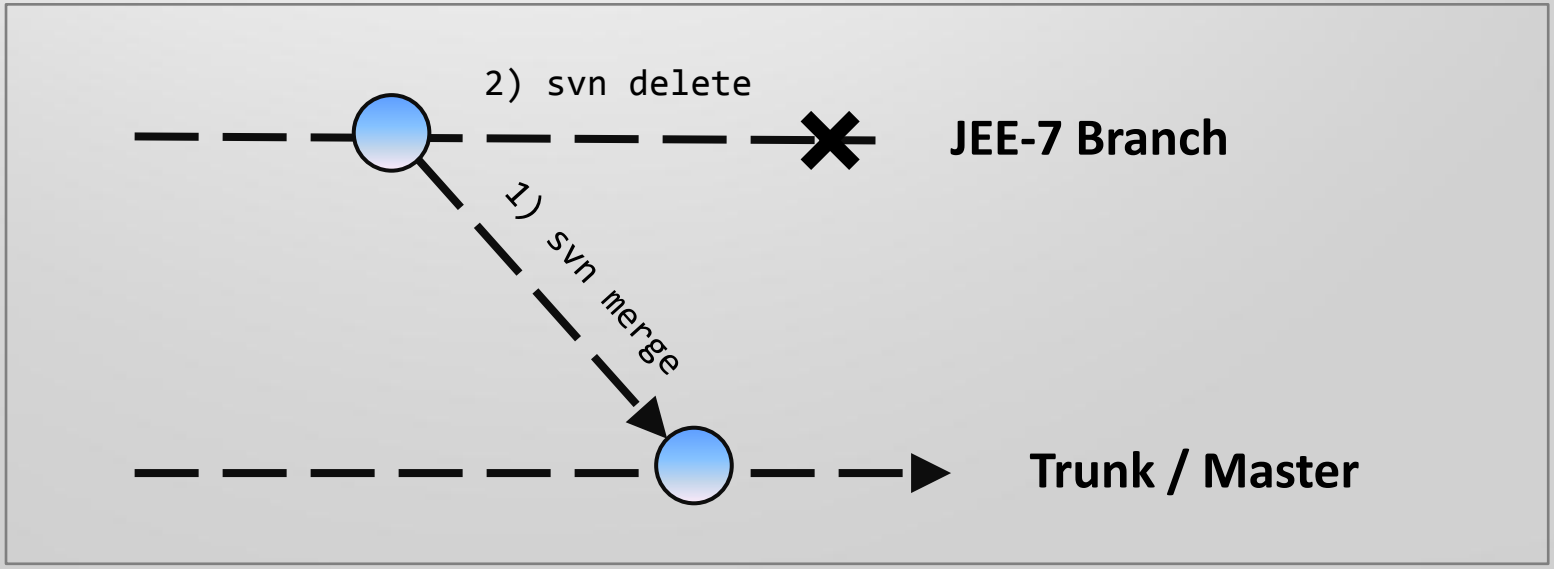


# Schritte zum Erfolg

1. Maven Dependency Management, Maven Profile
2. VCS Branches für nicht-kompatiblen Source Code
3. Continuous Integration
4. Continuous Delivery / Release Pipeline
5. Rolling Deployment / Inkrementeller Rollout
6. Laufzeitverhalten überwachen
7. **Aufräumarbeiten**

# Aufräumarbeiten

1. Integration aller Banches in die zugehörigen Trunks
2. Umstellen der Entwicklungsumgebung auf JEE-7
3. Primäre Jenkins Jobs anpassen (Maven Profil)
4. Alle zusätzlichen Jenkins Jobs löschen
5. Optional: Branches löschen



# Ausgewählte Glassfish Migrationsbeispiele

# GF3: JAX-RS (web.xml)

```
<web-app>
    ...
    <servlet>
        <servlet-name>Jersey Web Application</servlet-name>
        <servlet-class>com.sun.jersey.spi.container.servlet.ServletContainer</servlet-class>
        <init-param>
            <param-name>com.sun.jersey.config.property.packages</param-name>
            <param-value>de.gameduell.rest.accounting</param-value>
        </init-param>
        <init-param>
            <param-name>com.sun.jersey.api.json.POJOMappingFeature</param-name>
            <param-value>true</param-value>
        </init-param>
        <load-on-startup>1</load-on-startup>
    </servlet>
    <servlet-mapping>
        <servlet-name>Jersey Web Application</servlet-name>
        <url-pattern>/*</url-pattern>
    </servlet-mapping>
    ...
</web-app>
```

# GF4: JAX-RS (web.xml)

```
<web-app>
    ...
    <servlet>
        <servlet-name>Jersey Web Application</servlet-name>
        <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
        <init-param>
            <param-name>jersey.config.server.provider.packages</param-name>
            <param-value>de.gameduell.rest.accounting</param-value>
        </init-param>
        <init-param>
            <param-name>jersey.config.server.provider.classnames</param-name>
            <param-value>org.glassfish.jersey.jackson.JacksonFeature</param-value>
        </init-param>
        <load-on-startup>1</load-on-startup>
    </servlet>
    <servlet-mapping>
        <servlet-name>Jersey Web Application</servlet-name>
        <url-pattern>/*</url-pattern>
    </servlet-mapping>
    ...
</web-app>
```

# JAX-RS (@Application)

```
@ApplicationPath("/")
public class RestfulServiceApp extends Application {

  @Override
  public Set<Class<?>> getClasses() {
    Set<Class<?>> appClasses = new HashSet<Class<?>>();
    appClasses.add(ShopResource.class);
    // enable request debugging with LoggingFilter
    appClasses.add(LoggingFilter.class);
    // substitution for class ::
    // com.sun.jersey.api.container.filter.RolesAllowedResourceFilterFactory
    appClasses.add(RolesAllowedDynamicFeature.class);
    // use Jackson Feature instead of POJOMappingFeature
    appClasses.add(JacksonFeature.class);
    return appClasses;
  }

  @Override
  public Set<Object> getSingletons() {
    Set<Object> singletons = new HashSet<>();
    // substitution for class :: com.sun.jersey.api.container.filter.PostReplaceFilter
    singletons.add(new HttpMethodOverrideFilter(new HttpMethodOverrideFilter.Source[0]));
    return singletons;
  }
}
```

# JAX-RS Type Mapping (GF3 -> GF4)

| Jersey 1.x | JAX-RS2 / Jersey 2.x |
|---|---|
| com.sun.jersey.api.client.Client | javax.ws.rs.client.Client |
| com.sun.jersey.api.client.WebResource | javax.ws.rs.client.WebTarget |
| com.sun.jersey.api.client.ClientResponse | javax.ws.rs.core.Response |
| com.sun.jersey.spi.container.ContainerRequestFilter | javax.ws.rs.container.ContainerRequestFilter |
| com.sun.jersey.spi.container.ContainerRequest | org.glassfish.jersey.server.ContainerRequest |
| com.sun.jersey.spi.container.ContainerResponse | org.glassfish.jersey.server.ContainerResponse |
| com.sun.jersey.core.util.ReaderWriter | org.glassfish.jersey.message.internal.ReaderWriter |
| com.sun.jersey.api.json.JSONConfiguration | org.glassfish.jersey.jettison.JettisonConfig |
| com.sun.jersey.api.json.JSONJAXBContext | org.glassfish.jersey.jettison.JettisonJaxbContext |

# JPA / EclipseLink

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.0“ xmlns="http://java.sun.com/xml/ns/persistence"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
    http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd">

  <persistence-unit name="social-bus" transaction-type="JTA">
    <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
    <jta-data-source>jdbc/busDataSource</jta-data-source>
    <exclude-unlisted-classes>false</exclude-unlisted-classes>
    <shared-cache-mode>ENABLE_SELECTIVE</shared-cache-mode>
    <properties>
      <!-- Multiple persistence units clash with EclipseLink 2.5 annotation pre-processor -->
      <property name="eclipselink.canonicalmodel.subpackage" value="bus" />
    </properties>
  </persistence-unit>
  <persistence-unit name="social-mps" transaction-type="JTA">
    <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
    <jta-data-source>jdbc/mpsDataSource</jta-data-source>
    <exclude-unlisted-classes>false</exclude-unlisted-classes>
    <shared-cache-mode>ENABLE_SELECTIVE</shared-cache-mode>
    <properties>
      <property name="eclipselink.canonicalmodel.subpackage" value="mps" />
    </properties>
  </persistence-unit>
</persistence>
```

# Byte Code Incompatibility

```
public class SomeManagedBean {

   public void foo() {
      FaceletContext faceletContext = (FaceletContext) FacesContext
         .getCurrentInstance().getAttributes()
         .get(FaceletContext.FACELET_CONTEXT_KEY);
      String formId = (String) faceletContext.getAttribute("formId");
      ...
   }
   ...
}
```

```
// JEE-6
public abstract class FaceletContext extends ELContext {

   public static final String FACELET_CONTEXT_KEY =
      "com.sun.faces.facelets.FACELET_CONTEXT";
}
```

```
// JEE-7
public abstract class FaceletContext extends ELContext {

   // The key in the FacesContext attribute map for the FaceletContext instance.
   public static final String FACELET_CONTEXT_KEY = "javax.faces.FACELET_CONTEXT";
}
```

# @PostConstruct & Checked Exceptions

```
// Glassfish-3
@Stateless
public class AccountingService {

   @PostConstruct
   public void init() throws IOException {
      ...
   }
   ...
}
```

```
// Glassfish-4
@Stateless
public class AccountingService {

   @PostConstruct
   public void init(){
      try {
         ...
      } catch(IOException e) {
         throw new EJBException(e);
      }
      ...
   }
   ...
}
```

# Resource Injection

```
@Stateless
public class DataRetriever {

    @Resource(name = "jdbc/reporting_3")
    private DataSource dataSource;

    ...
}
```

```
@Stateless
public class DataRetriever {

    @Resource(lookup = "jdbc/reporting_3")
    private DataSource dataSource;

    ...
}
```

# Weitere Migrationsbeispiele

- Jackson-Lib package name (ObjectMapper)
  `org.codehaus.jackson.map → com.fasterxml.jackson.databind`
- Geändertes Verhalten der JAXB Implementation
- Neue Atmosphere-Lib Version (1.0.15 → 2.1.7)
- CDI standardmäßig aktiv
- Fehlende Methoden für handgeschriebene Test-Mocks
- `HttpServletRequest.login(...)` in Verbindung mit Custom Authentication Modules
- JAX-RS: Default Konstuktoren für Result-Objekte bei WADL Generierung
- Dateistruktur `domain.xml` für Embedded Glassfish
- RESTful Ressource innerhalb von EAR-Artefakten können `@Stateless` nicht mehr verwenden

# Vielen Dank

https://jaxenter.de/operation-am-offenen-herzen-jee-7-migration-im-laufenden-betrieb-24684