# 目标

实现按文件增量更新系统在各个环境的版本，更加清晰查看文件变更，减少变更风险，减少文件传输时间，提高版本发布效率。

# 核心概念

1. **持续部署**

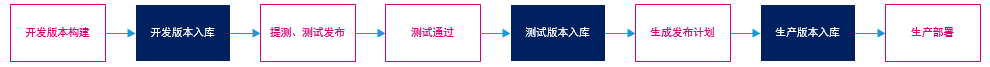
研发持续交付管理平台，以项目为单位，贯穿部署单元开发、测试、UAT、生产上线流程

1. **构建平台**

源码编译、打包，产出可运行模块。负责构建版本打tag，开发版本入库

1. **版本仓库**

用于存放应用运行所需要的所有文件，各个应用之间相同的文件仅保存一份，对文件内容进行MD5哈希得出文件key，持续部署系统通过key来访问所有文件。根据当前研发体系，分为3个版本仓库：开发版本库、测试版本库、生产版本库，对于3个仓库的文件提交需要关联到持续部署流程里，简单更新流程如下



1. **版本文件MD5清单**

持续部署管理的部署单位以版本文件MD5清单为主，所有交付文件流转都使用此清单。一个部署单元的不同版本对应着多个版本文件MD5清单，此清单用于版本更新对比，由版本管理模块统一存储。

1. **版本管理平台**

与持续部署紧密结合，记录部署单元历史版本，仅保存版本MD5清单。具体部署单元文件需按清单从版本仓库提取

# 核心实现逻辑

1. **开发构建**

由jenkins拉取svn/git源码，编译成class或其他文件，执行maven jar及package之前，对所有文件计算MD5编码，把构建结果增量提交到开发版本库，输出版本文件MD5清单提交到版本管理平台，并打上版本tag。

1. **版本入库**

版本库提供API上传文件，先上传MD5列表，版本库根据MD5索引计算此次新增文件返回MD5清单给客户端，客户端再调用文件上传功能进行入库，入库前再进行一次MD5查询防止覆盖。

1. **更新发布（不分测试部署还是生产部署）**

持续部署检查目标服务器模块的版本从版本管理平台拿到版本文件MD5清单，与待发布的清单进行对比，得出此次更新的文件列表，例子如下：

|  |
| --- |
| +685E711ACF2CBFF070E6B921A390191D \com\lz\lsf\util\XmlUtil.class  -F74DF591F4045A794A60E399DE29D8D3 \com\lz\lsf\exception\SystemException.class  +B2C8E3FCD959665998B31961595127FA \com\lz\lsf\exception\SystemErrorCodes.class  +C4345E7EEC7EE6841510D392998B926F \com\lz\lsf\orm\StringValuedEnumReflect.class  +0F501C81999B093772855F3427C43704 \com\lz\lsf\orm\StringValuedEnum.class  FC104230D7243A2D60F0D0BBCB3F480E \com\lz\lsf\util\StringUtil.class  43BA25C445CF0F55D816CFE9C0C013BB \com\lz\lsf\support\SpringTestCase.class  5CB612F57B73A754B2A515CEC89A08B8 \com\lz\lsf\exception\MessageResource.class  98EE94E4496D6934137FE1068EF73372 \com\lz\lsf\config\SpringContextHolder.class  0E1F7B337F3583EFF1C4BCF82043B6A9 \com\lz\lsf\service\query\SortField.class  30F28456CFE3559720217811B49D0092 \com\lz\lsf\exception\ServiceException.class  \*ABCC1E4450DAED1E461B81C80C9C73CB \conf\dubbo.properties |

+本次版本新增文件

-本次版本删除文件

\*本次更新的文件，可查看修改比较（文件路径相同而MD5不同）

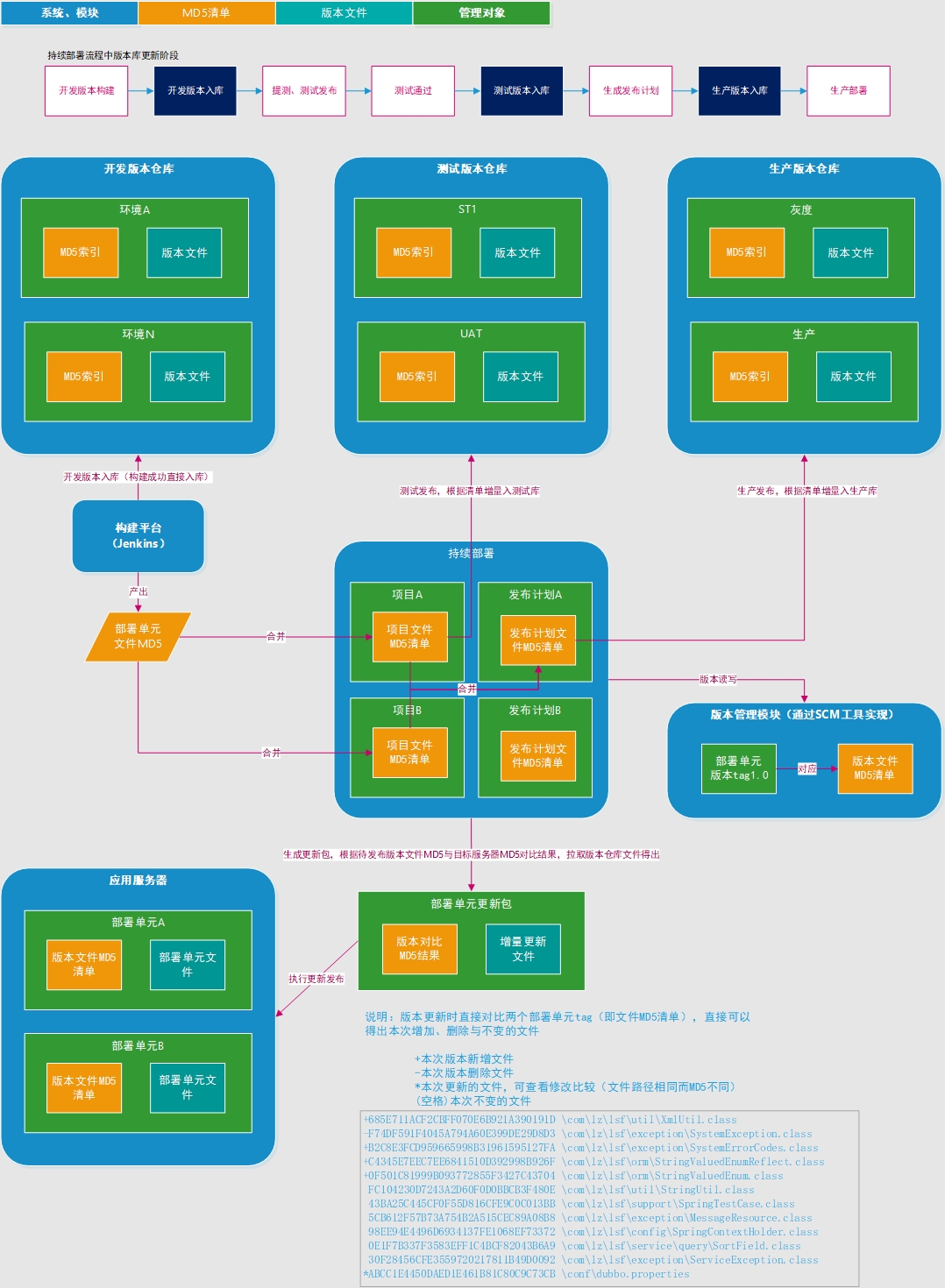
(空格)本次不变的文件

持续部署系统根据上面对比结果去版本仓库提取文件作合成此次增量更新包进行传输，在目标服务器进行增量更新，重启部署单元

# 评审建议

1. 强化生产版本管理概念
2. 生产一致性检查，防止文件被篡改
3. 注意文件所在目录变更的处理
4. 对组装的大包进行MD5校验，保证组装正确，也可以考虑大包内网传输

# 增量发布管理方案系统关系图



# 技术问题

1. 源码多次编译后保持MD5一致方案

pom.xml增加下面构建插件，去掉构建生成的pom.properties里的时间戳，统一加到模块父工程即可。编译环境JDK版本也需统一

|  |
| --- |
| <build>  <plugins>  <plugin>  <groupId>org.apache.maven.plugins</groupId>  <artifactId>maven-jar-plugin</artifactId>  <version>3.1.0</version>  <configuration>  <archive>  <manifest>  <addDefaultImplementationEntries>true  </addDefaultImplementationEntries>  </manifest>  </archive>  </configuration>  </plugin>  <plugin>  <artifactId>maven-jar-plugin</artifactId>  <dependencies>  <dependency>  <!-- MSHARED-494: avoid timestamps in pom.properties -->  <groupId>org.apache.maven</groupId>  <artifactId>maven-archiver</artifactId>  <version>3.1.1</version>  </dependency>  </dependencies>  </plugin>  </plugins>  </build> |