

第一版

伽然公司项目发布流程

|  |  |  |
| --- | --- | --- |
| 文件状态：  [√] 草案  [ ] 正式发布  [ ]正在修改 | 文档类别： | 软件开发编码规范 |
| 文档编号： |  |
| 当前版本： | 1.0.0 |
| 作 者： |  |
| 完成日期： |  |

**版 本 历 史**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本/状态** | **作者** | **参与者** | **起止日期** | **备注** |
| 1.0.0 | 杨常贺 |  |  | 创建文档 |
|  |  |  |  |  |

# 目 录

[1. 编写目的 4](#_Toc445460462)

[2. 适用范围 4](#_Toc445460463)

[3. 环境说明 4](#_Toc445460464)

[4. 发布配置说明 5](#_Toc445460465)

[5. 测试环境发布流程 6](#_Toc445460466)

[5. 用户测试环境/预发布环境发布流程 6](#_Toc445460467)

[6. 生产环境发布流程 7](#_Toc445460468)

# 1. 编写目的

明确项目发布过程中的各个事项，建立统一的发布流程规范，为项目顺利发布提供前期保证，避免项目发布过程中产生的不可控因素。

# 2. 适用范围

适用于伽然公司当前所有的Java项目

# 3. 环境说明

伽然公司所有的Java 项目发布均需要经过但不限于以下环境：

**开发环境** --> **测试环境** --> **用户测试环境** --> **生产环境**

**开发环境**: 开发人员宿主机

**测试环境**：集成测试环境

该环境的发布时频繁的，随着开发功能会及时更新

用于团队成员/测试人员/负责人测试，验收使用

**当前统一使用 gitool.glanway.com(121.40.102.225) 作为测试环境**

**用户测试环境**：预发布环境

该环境发布版本应该基本稳定的，阶段性发布

用于客户验收项目情况及预上线测试使用

对于用户测试环境的确定，存在以下两种情况

1. **客户提供用户测试环境**

对于客户提供测试环境的， 直接使用用户测试环境进行发布， 但是需要保证定期同步正式环境与用户测试环境的数据，以便保证该环境与正式环境没有区别

1. **客户不提供用户测试环境**

对于客户不提供测试环境的， 可以采用以下两种方案作为用户测试服务器

1. 发布测试项目到生产环境同一个Tomcat(或其他服务器)，项目使用不同名称，使用该项目进行测试
2. 在生产环境另外搭建一个 Tomcat(或其他服务器)，项目发布到这个Tomcat 进行测试

对于以上两种方式，不论采用哪一种方式，注意数据不能使用生产环境数据

（数据库重新创建，图片等资源另外存储）

**生产环境：**

该环境为最终提供给用户使用的环境

该环境发布版本应该是稳定的，一般不允许随意发布

# 4. 发布配置说明

1. 项目配置说明

所有项目均使用 Maven/Gradle 进行构建, 对于每个项目至少应该对上述每个环境提供一个配置(profile), 及至少应该包含以下几个配置

**dev**, **sit**, **uat**, **production** 分别对应 **开发**，**测试**，**用户测试/预发布**，**生产环境**

**对于以上每个配置，应该是各自独立的，避免互相影响**

1. 项目构建说明

项目统一进行使用持续集成工具Jenkins 进行构建，**每个项目**，对**每个环境**应该提供一个构建任务， **对每个构建任务后续应该提供不同的权限进行控制**

**对于 用户测试/预发布 环境及生产环境 只允许发布 版本控制中的 特定分支**

**Jenkins 统一使用 http://gitool.glanway.com/jenkins/**

1. 版本控制说明

项目统一使用 GIT 进行版本控制

对于一个项目在发布过程中至少应该存在两个分支： 开发分支和发布分支

例如: master/dev

鉴于当前习惯所致， 一般将 master 作为 开发分支，release 最为发布分支

开发时只允许在 master 分支进行开发

发布前将 master 中功能合并到 release 后进行发布，对于每一次发布后，均需要创建相关的 TAG, TAG 命名规则为 RELEASE-yyyyMMdd， 如果需要更加精确的可以使用 RELEASE-yyyyMMddHHmm

GIT 统一使用 **http://gitool.glanway.com**

# 5. 测试环境发布流程

对于测试环境的发布，主要发布代码为开发分支，需按照以下流程进行

**发布准备 –> 通知发布 --> 备份 --> 发布 --> 测试 --> 确认/回滚**

**发布准备：**

发布前应该准备好，所需要的资源，包含但不限于：

1. 数据库更新脚本 （此次更新的增量脚本）
2. 经过开发测试并提交的代码
3. 图片资源等

**通知发布：**

准备好以上资源后， 应该和团队内相关成员进行沟通，确认是否可以发布，发布资源是否有缺失，**如果有需要应该和项目经理沟通。**

**备份：**

发布前应该将重要数据， 修改前的内容进行备份，包括但不限于：

1. 数据库结构， 数据库数据
2. 图片等资源

**发布：**

以上工作处理好后， 可以进行发布数据库脚本，代码等资源

**测试：**

发布完成后，应该通知进行测试，主要包含：

1. 是否发布成功
2. 整体性功能是否收到影响（修改是否产生新的问题等）
3. 发布功能是否正常、
4. BUG是否修复

**确认/回滚**：

如果测试通过，则可以通知相关人员发布成功

否则根据情况确认是否需要回滚

# 6. 用户测试环境/预发布环境发布流程

在项目经过测试环境验收后，才可以发布到用户测试环境，用户测试环境发布流程如下

**发布准备 –> 通知发布 --> 备份 --> 合并分支 --> 发布 --> 测试 --> 确认/回滚**

**发布准备**：

参见《测试环境发布流程》

**通知发布：**

对于用户测试环境的发布，必须以邮件形式通知到客户，经客户确认后才可以发布（或和项目负责人沟通确认后才允许发布），确认时必须确认发布的时间点

**备份：**

参见《测试环境发布流程》

**合并分支：**

在发布前，如果需要发布开发的功能，需要将开发分支的代码合并到发布分支(现在为master 合并到 release), 分支合并后，需要添加 TAG， 完成合并后，需要提交合并和TAG，确认提交成功后，才可以执行发布操作

**发布：**

对于以上操作执行完毕后，最终发布前，应该和客户或项目负责人再次确认后，进行发布。

**测试：**

测试参见《测试环境发布流程》， 另外最终需要进行一次整体性测试

**确认/回滚：**

参见《测试环境发布流程》， 如果发布后出现问题，应该立刻回滚版本到备份文件，保证系统的正常运行，待问题解决后再次发布

# 7. 生产环境发布流程

**发布准备 –> 通知发布 --> 备份 --> 发布 --> 测试 --> 确认/回滚**

**发布准备**：

参见《用户测试环境发布流程》

**通知发布：**

对于用户测试环境的发布，必须以邮件形式通知到客户，经客户确认后才可以发布（或和项目负责人沟通确认后才允许发布），确认时必须确认发布的时间点

**注意：**

发布时间一般应该确定在 下班后，晚上等非高峰时间来进行

对于数据库等如果没有权限操作的问题，应该提前邮件发送给客户，请客户协助完成。

**备份：**

参见《用户测试环境发布流程》

对于需要客户协助的应该提前邮件通知客户进行备份。

另外 需要对之前版本进行全量备份包括：

项目 war 文件

图片资源（如果在磁盘存储）

数据库

**合并分支：**

在发布前，如果需要发布开发的功能，需要将开发分支的代码合并到发布分支(现在为master 合并到 release), 分支合并后，需要添加 TAG， 完成合并后，需要提交合并和TAG，确认提交成功后，才可以执行发布操作

**发布：**

1. 发布的分支应该是发布分支，该分支在用户测试环境发布时已经合并，且测试通过
2. 发布前确认需要客户备份的数据是否已经备份，如果备份后才可以往下执行
3. 再次确认当前发布时间是否可行
4. 发布前应该先停止服务，防止用户访问行为 (比如关闭端口)
5. 如果需要请客户协助完成 内容更新,
6. 更新相关资源
7. 发布项目

**测试：**

测试参见《用户测试环境发布流程》

对于可以测试的功能应该全面测试，

对于无法测试的功能，应该通知到客户进行测试

注意： 此时服务并不对外开放， 应该可以通过其他端口访问

**确认/回滚：**

参见《用户测试环境发布流程》

如果发布后出现问题，应该立刻回滚版本到备份文件，保证系统的正常运行，待问题解决后再次发布

发布成功或回滚后，应该开启对外服务（例如端口）保证系统正常运行