

基础设施建设路线图

网络建设

使用IP与端口提供服务是十分原始的

- ip不表意，不便于记忆，也容易弄错
- ip与端口会暴露主机信息
- 服务位置变更时，ip往往也会变更，需要修改

域名解析

通过域名解析服务使用域名访问服务

- 可以使用语义化域名，便于记忆与传播，也不容易弄错
- 域名不需要变更，如有需要只要修改ip于域名的对应关系即可

端口映射

所有可直接访问的web服务都应该映射到80端口

用户目录

应使用专门的用户目录服务管理组织结构

- 便于其他服务接入组织结构及权限配置
- 便于维护
- OA, CRM, 邮箱, SVN等均应使用一套用户服务

资源仓库

组件仓库

- 基于用户目录的权限提供nexus仓库的访问权限
- 组件资源提供唯一接口，不需要权限即可访问组件
- 组件上传权限应基于用户个人权限，而不是以往的仓库级别权限。这样每个用户只需要配置自己的认证即可根据该用户的组织权限进行上传。
- 仓库入口唯一并分配域名，可以考虑分布式

代码仓库

- 代码管理必须给所有开发开放分支权限
- 文档如果一定要和代码存放在一个仓库下，应该放置在不同分支中

文档仓库

建议使用wiki式在线文档而不是无法跟踪版本，无法及时更新的离线文档

共享仓库

提供开发资源仓库，共享文件，ftp，或在线式均可，最好可以标签化管理

即时通信工具

废弃基于局域网的飞秋，使用基于互联网的C/S结构的即时通讯工具。最好能提供接口，可以通过二次开发或者插件方式与其他工具集成，可以极大提高工作效率(腾讯的RTX也是支持二次开发的)。这方面可以参考一下现在流行的同类工具。

质量控制

测试驱动开发流程

1. 先有需求
2. 根据需求拆分功能点
3. 根据功能点编写自动化单元测试用例
4. 在单元测试用例指导下进行功能实施，并保证能通过单元测试(可以通过持续集成的方式验证)
5. 提交功能分支，并申请合并到开发主分支；功能分支进行集成测试，通过后合并入开发主分支
6. 开发主分支进行自动化测试，并发布开发快照

评审单元测试用例即可，不需要单独做代码评审，因为代码始终能通过单元测试。而且单元测试用例可以直接面向功能点编写，易于评审和验证。

自动化

- 尽量通过自动化的方式完成重复任务
- 通过持续集成的方式自动化构建项目，发布文档，部署项目至测试环境