**永银ERP系统升级改造建设方案2**

**V0.2**

**2016/2/25**

**一、**项目背景

目前ERP系统使用中存在一些问题，归纳总结如下：

1. 目前的系统开发基于OSGi框架，技术上不成熟，存在一些技术问题：

1) 系统稳定性差，经常出现莫名其妙的问题，重启系统或者重新打包部署能够解决；

2) 系统发生错误难以追踪，技术人员调试困难，维护开发成本非常高；

3) OSGi框架因其复杂性和各种技术问题已被基于浏览器的WEB应用彻底抛弃，无法从社区得到技术支持。

2. 缺少统一的运营日志功能，出现问题难以追查，运维难度大，建议建立一套独立的日志系统，所有用户操作均可追溯。

3. 后台JOB处于黑箱状态，无法监控运行状态，也无法通过界面配置运行时间等参数。

4. 现系统基于一套第三方的OA软件作为底层框架，此基础框架并没有源代码，系统使用中出现问题，如果与基础框架有关，根本无从调试。随着代码量越来越大，继续基于这个第三方类库进行二次开发的风险也越来越大。

5. 架构不合理，没有分表和分库机制，性能非常慢，而且随着时间推移会越来越慢，尤其是销售单、出库单和财务凭证这几部分核心功能。

6. 系统设计难以扩展，哪怕新增一个简单的类似功能，都要改动较多代码，不能满足快速部署上线的需要。

7. 很难在现有框架基础上开发APP应用。

8. 系统使用的WEB开源框架struts1已停止更新，很多JS的类库版本太低，主流浏览器支持较差，随着各种浏览器版本越来越高，使用中遇到的兼容性问题也会越来越多。

**二、**建设方案

基于以上存在的问题，我们综合考虑了以下两种解决方案：

**2.1 完全重新实现整个ERP系统**

优点：能够从技术层面解决现有系统的所有问题

缺点：1、风险大，因完全重新开发，存在非常大的不确定性，在上线过程中有可能会对单位业务造成严重影响。

2、周期长，从需求分析调研，到系统开发、测试，最后上线实施，需要花费2~3年时间。

3、工作量大，在新系统开发实施过程中，需要有公司内部负责实施人员专职全程参与。并且需要培训单位所有业务办理人员熟悉新系统，整个过程会非常耗时。

4、在开发过程中，现有ERP系统发现的问题或新的需求仍然需要在新系统中重新实现，属于重复劳动。

**2.2 基于现有ERP系统逐步完善**

优点：1、风险小，业务代码基本不用改动，用户操作完全按照现有模式，对单位的业务开展影响非常小。

2、周期短，基于现有系统，业务和代码都非常熟悉，从开发到测试、实施，最快可以在半年之内达成。

3、单位投入人力小，在现有系统上升级，单位实施人员对整个系统已经非常熟悉，实施过程可控，不用投入太多精力。

缺点：现有系统代码相对陈旧、封闭，开发维护成本较高，但现有开发人员已非常熟悉系统代码，也在可控范围。

**经过慎重考虑和综合比较，我们推荐第二套实施方案，考虑对现有系统进行分步升级，分阶段根据优先级顺序逐步完善，争取花三年左右的时间完成以下目标：**

**第一期(2016年)：在现有系统架构上做最小升级变更，解决最主要的技术问题，确保既有业务基本不受影响。**

1. 解决现有系统第一痛点：去掉OSGi框架，将所有WEB模块(各种oa\_xxx\_portal包）整合到同一WEB模块中，便于后续开发维护；重写系统中所有与OSGi相关的代码。通过去除OSGi，我们可根据业务需要，灵活引入第三方业务插件，而这个需求在现有系统中非常难以实现，因为引入的第三方插件也必须支持OSGi技术，才能够被现有系统使用。

2. 设计灵活、独立的日志系统，所有系统操作均可追溯，便于系统运维；

3. 系统后台JOB状态可监控，一些关键的JOB参数可配置；

4. 清理系统中大量无用的功能模块，减少维护工作量；

5. Java版本从6.0升级到最新版8.0，为后续升级打好基础；

**第二期(2017年上半年)：重点完善系统性能、多浏览器支持等，相应升级系统中用到的第三方框架。**

1. 研究数据库集群、分表、分库，以及缓存机制，使得应用性能可水平扩展，不会因为数据量越来越大而减低系统响应时间。

2. 系统中用到的JS框架jQuery存在多个版本，并且版本号已非常低，不能满足现代主流浏览器的兼容性要求，需升级到较高版本，并使系统兼容主流浏览器，如IE10+，360，Chrome，Firefox等；

3. 系统后台使用的Strust1框架已停止维护，使用主流的Spring MVC框架取代Struts1，并重写相应后台程序，并将Spring框架升级到最新版本4.2.4；

4. 升级API框架，以便于开发手机APP客户端；

**第三期(2017年下半年~2018年)：此阶段相对来说最复杂，实施风险较高，对业务影响也较大，属于可选升级。建议单位做好长期规划，花1～2年时间完成此项目标。**

1. 去掉底层的第三方OA框架，减少对第三方无源代码类库的依赖，底层代码全部自己实现或者基于可控制、有源代码的开源库，降低系统维护成本及技术风险；

2. 在去掉OA框架过程中，数据库结构尽量保持不变，现有数据做完整迁移；

3. 设计自动生成代码功能，使得一些基本的增删改查功能可以自动生成相关业务代码，满足ERP业务快速迭代上线的需要；

三、工作量评估

一期：275个人天

二期：220个人天

三期：属于可选项升级，工作量较大，暂不评估。

|  |  |  |  |
| --- | --- | --- | --- |
| 阶段 | 分类 | 内容 | 工作量  (人天) |
| 一期  (总计275) | 系统架构  (总计85) | 去掉OSGi框架 | 15 |
| ~~Java6升级到Java8~~ | ~~15~~ |
| 基于ELK设计独立的日志系统架构 | 15 |
| ~~后台JOB状态可监控，参数可配置~~ | ~~30~~ |
| ~~升级JOB引擎Quartz到版本2.2.2~~ | ~~10~~ |
| 系统开发  (总计130) | 将现系统中的15个基于OSGi的oa\_xxx\_portal模块整合为同一个包 | 30 |
| 去掉现有系统中31个oa\_xxx\_manager/oa\_xxx\_inner/oa\_xxx\_glue包中所有依赖于OSGi的代码 | 60 |
| 基于ELK实现各业务模块的日志功能 | 30 |
| ~~清理系统中大量无用的功能模块~~ | ~~10~~ |
| 系统测试  (总计60) | Beta1 | 20 |
| Beta2 | 15 |
| Beta3 | 10 |
| RC(Release Candidate) | 15 |
| 实施上线  (总计30) | 系统实施上线并根据用户实际需求调整系统 | 30 |
| ~~二期~~  ~~(总计220)~~ | 系统开发  (总计135) | 数据库集群及分库、分表机制解决性能问题 | 30 |
| 多浏览器支持，并升级jQuery等基础JS框架 | 30 |
| 升级Spring到版本4.2.4，并替代Struts1搭建后台框架 | 15 |
| 重写15个oa\_xxx\_portal包中基于Struts1的后台Action | 60 |
| 系统测试  (总计55) | Beta1 | 20 |
| Beta2 | 15 |
| Beta3 | 5 |
| RC(Release Candidate) | 15 |
| 实施上线  (总计30) | 系统实施上线并根据用户实际需求调整系统 | 30 |
|  |  |  |  |

另外有两点建议：

1、采购一台新的数据库服务器，与应用分开部署以降低系统负载。费用在10万左右。

2、现有测试服务器很慢，建议采购一台测试服务器，费用在2万左右的高配置PC即可。

3、建议单位员工的PC至少应该升级到Win7，Windows XP已得不到微软官方支持，安全性存在很大隐患。

**三、实施计划**

计划投入2~3名资深开发人员(8年+)在1年半时间内完成1、2期建设目标:

**备注：三期属于远期可选升级，暂不做规划。**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **分期** | **阶段** | **时间** | **任务** | **备注** |
| 一期 | 启动 | 2016年  3~4月 | 1. 需求分析，同时做好系统框架设计 |  |
|  | 系统开发 | 2016年  5~7月 | 1. 完成所有系统功能开发 |  |
|  | B1 | 2016.8月 | 1. 发布测试版B1，开发人员内部测试 |  |
|  | B2 | 2016.9月上旬 | 1. 发布测试版B2，开发人员内部测试 |  |
|  | B3 | 2016.9月下旬 | 1. 发布测试版B3，开发人员内部测试 |  |
|  | RC | 2016.10 | 1. 发布Release Candidate版本，部署到客户服务器，由用户试运行测试 |  |
|  | 系统实施 | 2016年  10~11月 | 1. 正式上线实施 |  |
| 二期 | 系统开发 | 2017年  1~3月 | 1. 完成所有系统功能开发 |  |
| B1 | 2017.4月 | 1. 发布测试版B1，开发人员内部测试 |  |
| B2 | 2017.5月上旬 | 1. 发布测试版B2，开发人员内部测试 |  |
| B3 | 2017.5月下旬 | 1. 发布测试版B3，开发人员内部测试 |  |
| RC | 2017.6月 | 1. 发布Release Candidate版本，部署到客户服务器，由用户试运行测试 |  |
| 系统实施 | 2017.7月 | 1. 正式上线实施 |  |
|  |  |  |  |  |