**XXX软件(项目)测试报告**

[1. 简介 2](#_Toc59352743)

[1.1 编写目的 2](#_Toc59352744)

[1.2 项目背景 2](#_Toc59352745)

[1.3 定义、首字母缩写词和缩略语 2](#_Toc59352746)

[1.4 参考资料 2](#_Toc59352747)

[2. 测试概要 2](#_Toc59352748)

[2.1 测试时间 2](#_Toc59352749)

[2.2 测试范围 3](#_Toc59352750)

[2.3 测试版本 3](#_Toc59352751)

[2.4 测试用例 3](#_Toc59352752)

[2.5 测试策略 3](#_Toc59352753)

[3. 结果分析 4](#_Toc59352754)

[3.1 缺陷趋势 4](#_Toc59352755)

[3.2 缺陷优先级分布 4](#_Toc59352756)

[3.3 缺陷按模块分布 5](#_Toc59352757)

[3.4 重开缺陷情况 5](#_Toc59352758)

[3.5 遗留缺陷情况 5](#_Toc59352759)

[3.6 上线跟踪测试结果 5](#_Toc59352760)

[4. 结论&问题&建议 5](#_Toc59352761)

[4.1 测试结论 5](#_Toc59352762)

[4.2 呈现的问题 5](#_Toc59352763)

[4.3 测试建议 6](#_Toc59352764)

# 1. 简介

## 1.1 编写目的

本文档用于记录测试过程，总结各轮次的测试情况，分析测试数据，归纳测试工作进行过程中暴露的问题与遗留的风险，给出相应的测试建议以供后续项目参考。

## 1.2 项目背景

xx需要一个拥有真实用户的社区化产品，通过真实高信任度用户关系的建立，提高用户粘性，提升活跃会员数，带来长效的增长。在此背景下，以真实用户为基础的社区应运而生。主要具有以下5点意义：

1. 提高社区活跃会员数

2. 提高用户粘度

3. 建立真实（和用户的社区身份相一致）的多维用户信息

4. 建立高信任度的用户关系

5. 达到真实可信用户关系中的用户之间的传播效应

## 1.3 定义、首字母缩写词和缩略语

无

## 1.4 参考资料

各轮系统测试阶段总结

# 2. 测试概要

整个xx项目的测试经历了xx-1.0与xx-1.1两个阶段，共经历了1轮集成测试、6轮冒烟测试和7轮系统测试和1轮上线跟踪测试。整个测试过程中累计执行用例8100条，发现缺陷1026个。截至xx-1.1第四系统测试结束，所发现的高权重问题已得到修复和验证。

## 2.1 测试时间

整个xx项目的测试时间从xx年2月18日开始，到xx年3月27日上线止，期间各阶段工作情况如下：

## 2.2 测试范围

本次测试覆盖的范围包括：功能测试、兼容性测试、接口测试、数据迁移测试、性能测试、安全性测试和品质监控。以下分别对功能测试、兼容性测试、接口测试、数据迁移测试、性能测试和安全性测试进行说明。

功能测试

xx-1.1在xx-1.0基础上更新的主要功能如下：

数据迁移测试

接口测试

兼容性测试

性能测试

参见SVN中的性能测试报告。

安全性测试

整个xx测试过程中先后进行了三轮安全性测试，发现了2个影响较严重的安全性问题，且都已得到修复和验证。

## 2.3 测试版本

下表显示了各轮次测试版本和对应测试范围的分配情况：

## 2.4 测试用例

根据需求文档，测试人员编写和内审了测试用例，为xx项目共计编写用例3558条，累计执行用例8100条。

## 2.5 测试策略

xx-1.0测试策略

1. 测试类型：按阶段划分定义为集成测试和系统测试。

2. 集成测试阶段进行了一轮集成测试，主要以需求挖掘、分析、确认和寻找实现与需求不一致为主要目标

3. 系统测试阶段分三轮进行，基本策略如下：

第一轮为覆盖性测试，测试范围覆盖以上描述的所有范围，关注所有级别的bug；

第二轮对权重为A、B的模块进行功能测试、兼容性测试，权重为C的模块进行冒烟测试，回归测试所有已修复的bug；

第三轮对权重为A的模块进行功能测试、兼容性测试，对权重为B、C的模块进行冒烟测试，回归测试所有待解决的bug，及已关闭的高优先级bug。

每轮测试开始前都进行快速的冒烟测试，通过冒烟确信系统可测时进入下一轮系统测试。

数据迁移测试、接口测试只在第一轮进行。

4. 缺陷评估：每轮测试结束后都组织开发工程师、测试工程师、产品工程师等共同评估产品缺陷，评估内容包括缺陷解决方案、是否涉及需求变更、下一轮开始时间及是否可以结束测试等。

xx-1.1测试策略

1. 测试类型：按阶段划分定义为系统测试。

2. 系统测试分四个轮次进行，基本策略如下：

第一轮为覆盖性测试，测试范围覆盖以上描述的所有范围，关注所有级别的bug；

第二轮对权重为A、B的模块进行功能测试、兼容性测试，权重为C的模块进行冒烟测试，回归测试所有已修复的bug；对系统进行性能测试。

第三、四轮对权重为A的模块进行功能测试、兼容性测试，对权重为B、C的模块进行冒烟测试，回归测试所有待解决的bug，及已关闭的高优先级bug；

每轮测试开始前都进行快速的冒烟测试，通过冒烟确信系统可测时进入下一轮系统测试。

数据迁移测试、接口测试只在第一轮进行。

3. 缺陷评估：每轮测试结束后都组织开发工程师、测试工程师、产品工程师、QA等共同评估产品缺陷，评估内容包括缺陷解决方案、是否涉及需求变更、下一轮开始时间及是否可以结束测试等。

# 3. 结果分析

整个xx测试过程中累计发现有效缺陷1026个，其中A级缺陷3个，B级21个，C级800个，D级169个，E级33个。经项目组成员评估，到xx-1.1发布止遗留缺陷51个，其余975个缺陷均已修复且全部验证通过。下面分别从xx-1.0和xx-1.1两个阶段、从不同角度对缺陷进行分析。

## 3.1 缺陷趋势

xx-1.0

整个测试过程中累计发现缺陷734个，各轮次缺陷分布情况如下表。

下图显示了xx-1.0测试过程中缺陷的发展趋势：

xx-1.1

整个测试过程中累计发现缺陷292个，各轮次缺陷分布情况如下表。

下图显示了xx-1.1测试过程中缺陷的发展趋势：

从缺陷趋势图中可以看出，xx-1.0和xx-1.1两个测试阶段，缺陷均随着测试过程的推进呈现收敛趋势，这符合测试缺陷的发展规律，证明测试计划和策略是可靠有效的。

## 3.2 缺陷优先级分布

xx-1.0

每轮次各级别缺陷分布情况如下：

xx-1.1

每轮次各级别缺陷分布情况如下：

整个xx项目测试过程种中发现的C级以上（包括C级）缺陷824个，占总缺陷数的80.31%，这说明系统在测试过程中处于不稳定状态，存在大量较为严重的问题，但随着测试过程的推进，高优先级问题又逐渐减少，整个系统趋于稳定。

## 3.3 缺陷按模块分布

下表显示了整个xx测试过程中发现缺陷在各模块中的分布情况

从下图中可以看出各模块缺陷的分布趋势：

从以上缺陷分布情况看，所有缺陷中有近30%是和产品需求相关的，诸如需求定义欠明确、需求描述有歧义、需求没有定义、实现和需求不一致等。

## 3.4 重开缺陷情况

从上表可以看出整个Space测试过程中，各轮验证缺陷的重开比率都偏高，这是我们后续项目中需要关注和提高的地方。

## 3.5 遗留缺陷情况

到xx-1.1发布止，整个Space项目遗留缺陷51个，且这些缺陷均通过PDT相关成员评估后确信可以遗留，待后续版本规划处理。具体缺陷信息此处略去。

## 3.6 上线跟踪测试结果

xx-1.1于3月27日7:00上线后，我们在当日的7:00-12:00进行了集中跟踪测试，且在此之后安排有2名测试工程师，每天用一些时间跟踪上线情况、客服反馈问题的最新动态。截止4月2日上午11:00上线跟踪测试结果是：累计缺陷40个，且都是C级以下。其中属需求相关问题5个，因上线后环境差异导致问题35个，开放中缺陷4个，已关闭缺陷16个，已解决待验证缺陷20个。

# 4. 结论&问题&建议

## 4.1 测试结论

1. 经过前后两个阶段的多轮测试，虽遗留了一些缺陷没有解决，但系统功能已趋于稳定，且项目确定的范围、策略和计划均已实现，项目测试可以结束、xx-1.1可以上线。

2. 通过测试觉得产品在用户体验方面有待后续版本进一步改进，不排除用户在使用该产品时有“晕”的感觉。

## 4.2 呈现的问题

1. 需求问题。特别是xx-1.0项目需求，虽然陆续看到了好多需求文档，但这些文档给人的感觉是：需求分析不完整、需求描述不清晰，需求文档的逻辑性、可读性、可实现性、可测试性比较差，需求的歧义性较大。从而感觉在整个xx-1.0测试过程中不断地在挖掘需求、确认需求、变更需求和评审需求。xx-1.1的项目需求有了很大改观，xx本身需求经过和收集、分析、确认和评审的过程，但对各接口产品的需求仍然没有进行统一的分析、确认和评审，这部分需求的歧义性较大且变更较多，整个需求文档的可读性、可测试性、完整性和清晰性仍然较差。

2. 变更控制问题。这在xx-1.0的测试过程中体现的比较明显，如项目需求的变更、项目责任人的变更、项目计划的变更等。xx-1.0整个测试过程中一直在确认和变更需求，且需求变更的机制没有规约，一个会议、一封mail或是一个口头传达就可能变更需求。xx-1.1测试过程中这一问题得到了较好的改进，但变更控制规约的实施欠佳。所有的需求变更均没有及时很好的更新至需求文档，xx-1.0体现的更为明显。

3. 版本控制问题。xx-1.0测试过程中没能进行版本管理；xx-1.1测试过程中对xx本身的代码进行了版本管理，各接口产品的代码均由各技术负责人进行管理，在这期间出现过代码覆盖的情况、代码忘记上传或遗漏部署的情况。难以保证每轮测试版本的清晰、和发布版本与测试版本的一致性。

4. 测试环境问题。xx-1.1测试期间测试环境和开发环境没能很好的分离，导致测试和开发修复缺陷不能并行；测试期间有开发工程师直接在测试环境上修复缺陷和修改测试环境的情况；测试环境不稳定，如hosts设置不正确等。

5. 项目计划欠明确、人员职责欠清晰。

## 4.3 测试建议

1. 遗留缺陷。建议在xx-1.1上线后以patch方式，或在后续版本中解决遗留缺陷，以提升产品的稳定性和用户体验。

2. 需求建议。不论是xx本身还是各接口产品，建议进一步加强需求收集、分析、确认和评审过程，进一步提升需求文档的质量：减少需求的歧义性，提升需求的完整性、描述的清晰性、一致性、可读性、可实现性和可测试性。同时建议在后续项目中能对设计文档（如UI/UE等）进行评审，以增强产品的使用性、提升用户体验。

3. 变更控制。建议在后续项目中进一步加强变更控制策略和规约制定，并强化变更控制规约的执行。不怕变更，关键要控制好变更的时机和策略。

4. 版本控制。加强xx本身，特别是各接口产品的版本控制策略，以保证测试版本的清晰性、发布/上线版本和最终测试版本的一致性。

5. 测试环境。期望在后续项目中xx及各接口产品的测试环境和开发环境完全分开，或阶段性完全独立，且各部分环境有专门的接口人负责，在测试期间严格禁止在测试环境上修复缺陷或更改环境配置（如确实需要更改配置，请提前通知测试及其它相关负责人）。以减少因此带来的沟通、反复侦测的成本。

6. 项目管理。主要是建议加强项目的计划性，诸如：进度计划、人力资源计划、风险预防机制等，这也将更利于项目成员间高效的配合：大家能更适时的、更合理的制定各自工作计划，也更清楚到什么时候我会输出什么、我将配合他人做些什么。减少项目进行过程中的紧张和慌乱、项目也变得更加易控和可控。