

关于食堂 app 项目的工作总结报告

报告名称: 食堂点评 app 失败报告

专 业: 软件工程

组 别: 第一组 Q

小组成员: U201217427 胡家煊

U201217375 杜定益

U201217448 祝远

U201217517 邱鸿宇

目录

引言:	3
一、 食堂 app 项目的现状.....	3
二、 失败原因的分析.....	3
1、 程序的接口未能在前期经过严密的理论推导及严格定义	3
2、 前期对框架的性能瓶颈没有足够的重视，在设计框架时，未能提前将系统性能及其可能导致的结构调整进行充分论证.....	4
3、 一厢情愿，盲目乐观，违背程序编制的客观规律，强调程序员的个人能力及士气.....	5
4、 对各功能点的实施控制过粗，未能使用有效办法控制总体进度	5
5、 前期完全以框架推动进度，后期完全以需求推动进度，分别走向两个极端	6

引言：

食堂 app 是我们小组做的一个有关学生食堂吃饭对比的项目。由于时间比较紧张，所以到目前为止项目并未开发完全。针对一个尚未全部完成的项目，发现问题、找出不足，改正错误、改进方案、调整决策等才是最重要的目的，因此，本报告中只将食堂 app 项目中存在的问题列出，重点是为下一步项目重启和其它项目决策等提供意见。

一、 食堂 app 项目的现状

食堂 app 项目在 2015 年 4 月 10 日之后，已经暂时停止下来。不论在感情上愿不愿意接受，该项目是一个失败的项目：

- 1、 项目的实施时间和难度大大超出先期计划；
- 2、 经过实际工程检验，证明系统框架在稳定性、性能、速度、内存消耗等方面不能达到实用化要求；
- 3、 预期的部分功能，在现有的框架结构上不能实现；
- 4、 开发团队的建设未能完全达到预期的目标；

二、 失败原因的分析

1、 程序的接口未能在前期经过严密的理论推导及严格定义

为了达到团队并行开发的目标，本项目在方案设计时决定采用选菜、评论和数据分析三个同时推进的子系统。这三个子系统之间的接口在最初未能严格定义，主要存在如下的问题：

- A、 接口实现的冗余度及可选度太大；
- B、 接口定义未能完全屏蔽内部实现细节，将太多的内部实现细节及其依赖关系暴露给调用者，未能完全实现接口调用的内部透明化；
- C、 接口的约束关系不是足够明确，接口不够健壮，不正确的格式、参数等处理也能处理，但会导出一个不正确的结果；
- D、 接口的功能集不健全，在后期由于功能的增加，临时增加了相当多的新接口，而新增的接口只以功能上满足特定需求为目标，未能统一推导及定义。

产生该问题的原因有：

- A、食堂 app 项目是一个新的项目，在项目初期许多需求及实现的细节未能明了；
- B、食堂 app 项目是一个非常大的项目，许多接口在实现过程中才能发现其遗漏；或为了实现某些优化处理必须改变接口；
- C、在项目实施初期，功能需求未能真正明了，实现细节未能有实际的效果；

从项目实施的过程来看，接口未能严格定义导致了如下问题：

- A、三个程序组在具体实现某具体功能时才决定增加接口，导致系统的并行开发未能完全高效地开展；
- B、出现多次比较剧烈的程序收敛振荡，许多本已解决的问题或不存在的问题在新版本中重复出现。导致开发进度的控制无法有序进行；三个程序组可能会由于同一个错误同时耽搁时间；
- C、对程序的测试工作加长，测试进度及测试重点无法确定；

2、前期对框架的性能瓶颈没有足够的重视，在设计框架时，未能提前将系统性能及其可能导致的结构调整进行充分论证

系统的开发过程中，针对原有系统架构设计中存在的一些问题，希望提供一个能解决现有系统的新框架，主要有如下问题：

- A、原有系统的框架不适合并行开发；
- B、原有系统扩充不方便；

在设计系统时，提出了元对象的概念，进而提出了服务器元对象树的内部数据定义，各部分接口针对元对象树进行遍历和操作，在理论上提供一种模仿真实世界的一种模型。但是在该模型的实施过程中，对性能、速度等问题考虑得不多，主要有如下方面的问题：

- A、元对象是一个弱类型结构，其转换及处理性能不佳；
- B、以字符串形式表示各种数据，在处理时再转化为具体类型数据进行运算，性能不佳；
- C、某些特殊处理仍使用元对象，内存消耗大，速度慢；

对于系统性能的优化有许多种解决方案，在决策时应该考虑以下几个方面的综合加权因素：

- A、框架的清晰
- B、稳定
- C、性能
- D、系统的复杂性

在系统实施的前期，强调了框架及稳定，但在系统性能方面则考虑不够，希望首先推出一个能够运行的稳定的框架，再不断地增加功能，最后才考虑性能的瓶颈及性能优化。

食堂 app 项目是一个新开发且比较的系统，因此，在初次设计时不能充分论证系统的可行性，或者论证不彻底等，都是可以接受的，正因为其是一个复杂的系统，必须采用螺旋式开发模式。但本次开发过程中，对螺旋式开发的阶段性划分不够明确，其具体表现在如下几个

方面：

- A、 在最初计划时，对系统开发周期及阶段划分有一定的计划，但计划未能反映各阶段的里程碑及新周期的起始条件；
- B、 对每阶段的起始和结束时间及每阶段需要完成的任务目标未能明确定义；
- C、 对阶段性目标的完成及具体实施控制未能有意识地进行；
- D、 以过程监督的检查作为阶段工作的目标，且每次检查的目标都不够明确，经常为了迎接检查而重新调整计划，导致阶段目标模糊；

3、 一厢情愿，盲目乐观，违背程序编制的客观规律，强调程序员的个人能力及士气

我们第一次采用这样的团队方式进行项目开发。对于该项目具体需要多少人员、时间；到底需要什么层次技能的程序组组长和具体开发人员；以及其它需要考虑的问题等，因为没有经验，只能在实施过程中逐步修正，但在实施过程中，却由于功能不明确，或者任务不明确等，许多程序员在相当一部分时间内具有盲目乐观的表现，其具体表现在以下几个方面：

- A、 过多地强调某人某时编制了多少行的代码量，对其编码质量、编程的客观规律等有一种轻视的态度倾向；
- B、 忽视模块测试、接口测试、白盒测试等,将所有的测试以系统整体形式由测试员统一测试；
- C、 在系统未统一联调之前，我们对工作已完成的阶段缺少认识，普遍存在一种盲目乐观的心情。

4、 对各功能点的实施控制过粗，未能使用有效办法控制总体进度

具体表现在如下几个方面：

- A、 在详细设计时接口变动之后，我们小组未能形成一个有效的方法来记录接口变动说明，每个接口变动都是在我们商量后进行定义，而定义后的接口未能以清晰、明确的方式在各成员之间加以确定；
- B、 系统设计的初期采用了一些规范的方法，拟对系统的开发过程进行控制，包括开发流程、系统分析、概要设计、详细设计、测试过程等，但在实际实施的过程中，将前期的许多工作束之高阁，在开发过程中重新回到一种自发的下意识的控制过程，未能真正地有效地控制住开发过程；
- C、 各成员每周都有一些具体的开发目标，对于目标是否真正达到，或由于某些原因导致的任务延迟，没有具体的方法来判断其情况真实、准确性。进度的实施依赖成员的责任心和自觉性。

5、 前期完全以框架推动进度， 后期完全以需求推动进度， 分别走向两个极端

在系统设计的前期，主推框架，功能需求的工作只作为一种从属的作用。许多工作都是在功能不太明确或是未能详细推导的情况下开始进行，对系统的复杂性、实现时的性能因素等只在问题暴露时才想办法解决。导致了系统框架的多次重构。

开发时，则完全以功能来推动整个系统的进度，以完全、完美地实现项目的所有功能作为主要目标，而实际上，以 2 个月的时间，根本完成不了所有的功能。每个人都明白这个道理，但都不愿意承认这个事实，或一厢情愿地相信我们的能力和责任心，只能以一种完成多少算多少的心态来工作。

实际上，该项目的核心问题是框架的稳定性和可扩充性，后来才转向以重点需求推动框架的形式。