

# 食堂点评风险分析报告

报告名称: 食堂点评 app 风险分析

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

- 一、 概述.....3
  - 1.1 目的和适用范围： .....3
  - 1.2 产品描述.....3
    - 1.2.1 概况.....3
    - 1.2.2 预期用途.....3
  - 1.3 风险管理计划简述.....3
- 二、 风险管理人员.....4
  - 2.1 风险管理小组.....4
  - 2.2 软件过程风险.....4
    - 2.2.1 软件需求阶段的风险.....4
    - 2.2.2 设计阶段的风险.....4
    - 2.2.3 实现阶段引入的风险.....4
    - 2.2.4 维护阶段的风险.....5
- 三、 项目风险表.....5

# 一、 概述

## 1.1 目的和适用范围：

本文是对食堂APP进行风险管理的报告，报告中对食堂APP的开发进行总体评价，所有的可能风险以及每一个风险产生的原因进行了判定,对于每种风险可能产生损害的严重度和风险的发生概率进行了估计,在某一风险水平不可接受时，采取了降低风险的控制措施，同时，对采取风险措施后的剩余风险进行了可接受性评价，证实对产品的风险已进行了管理，并且控制在可接受范围内。

## 1.2 产品描述

### 1.2.1 概况

对食堂点评、今日菜单、菜品大全、数据分析、精品推送，并及时得更新食堂的数据。学生能够通过自己的偏好选择特定的食堂；食堂也能够根据就餐人数的数据分析对自身进行改善。

### 1.2.2 预期用途

提高学生在食堂就餐的选择性。

## 1.3 风险管理计划简述

食堂APP项目开始策划立项时，我们就针对该产品进行了风险管理活动的策划，指定了风险管理计划，该风险管理计划确定了风险管理参加人员及职责的分配、基于制造商决定可接受风险方针的风险可接受性准则，包括在损害发生概率不能估计时的可接受风险的准则等内容。

## 二、风险管理人员

### 2.1 风险管理小组

根据本小组人员情况，所有成员均参与对项目风险的管理。

### 2.2 软件过程风险

#### 2.2.1 软件需求阶段的风险

软件的开发是以用户的需求开始，在大多数情况下，用户需求要靠软件开发方诱导才能保证需求的完整，再以书面的形式形成《用户需求》这一重要的文档。需求分析更多的是开发方确认需求的可行性和一致性的过程，在此阶段需要和用户进行广泛的交流和确认。需求和需求分析的任何疏漏造成的损失会在软件系统的后续阶段被一级一级地放大，因此本阶段的风险最大。

#### 2.2.2 设计阶段的风险

设计的主要目的在于软件的功能正确的反映了需求。可见需求的不完整和对需求分析的不完整和错误，在设计阶段被成倍地放大。设计阶段的主要任务是完成系统体系结构的定义，使之能够完成需求阶段的即定目标；另一方面也是检验需求的一致性和需求分析的完整性和正确性。

设计本身的风险主要来自于系统分析人员。分析人员在设计系统结构时过于定制，系统的可扩展性较弱，会给后期维护带来巨大的负担，和维护成本的激增。对用户来说系统的使用比例会有明显的折扣，甚至造成软件寿命过短。反之，软件结构的过于灵活和通用，必然引起软件实现的难度增加，系统的复杂度会上升，这又会在实现和测试阶段带来风险，系统的稳定性也会受到影响。从另一个角度上看，业务规则的变化，或说用户需求和将来软件运行环境的变化都是必然的情况，目前软件设计的所谓"通用性"是否就能很好的适应将来需求和运行环境的的变化，是需要认真折衷的。这种折中也蕴涵着很大的风险。

设计阶段蕴涵的另一种风险来自于设计文档。文档的不健全不仅会造成实现阶段的困难，更会在后期的测试和维护造成灾难性的后果，例如根本无法对软件系统进行版本升级，甚至是发现的简单错误都无从更正。

#### 2.2.3 实现阶段引入的风险

软件的实现从某种意义上讲是软件代码的生产。原代码本身也是文档的一部分，同时它又是将来运行于计算机系统之上的实体。源代码书写的规范性，可读性是该

阶段的主要风险来源。规范的代码生产会把属于程序员自身个性风格的成分引入代码的比例降到最低限度，从而减小了系统整合的风险。

### 2.2.4 维护阶段的风险

软件维护包含两个主要的维护阶段，一个是软件生产完毕到软件试运行阶段的维护，这个阶段是一种实环境的测试性维护，其主要目的是发现在测试环境中不能或未发现的问题；另一个阶段是当软件的运行不再能适应用户业务需求或是用户的运行环境（包括硬件平台，软件环境等）时进行的软件维护，具体可能是软件的版本升级或软件移植等。

从软件工程的角度看，软件维护费用约占总费用的 55%~70%，系统越大，该费用越高。对系统可维护性的轻视是大型软件系统的最大风险。在软件漫长的运营期内，业务规则肯定会不断发展，科学的解决此问题的做法是不断对软件系统进行版本升级，在确保可维护性的前提下逐步扩展系统。

在软件系统运营期间，主要的风险源自于技术支持体系的无效运转。科学的方法是有一支客户持队伍不断收集运行中发现的问题，并将解决问题的方法传授给软件系统的所有使用者。

## 三、项目风险表

综上所述得出以下项目分析表：

风险	概率	影响
规模估计过低	60%	严重的
交付期限太紧张	50%	严重的
需求变化频繁	75%	严重的
技术达不到预期效果	30%	轻微的
质量保证体系的措施实施不利	60%	严重的
软件体系结构设计不合理	40%	灾难性的
人员流动	10%	严重的