**Z公司Android软件项目风险因素识别及应对策略研究**

**摘要**

**第一章绪论**

**1.1选题背景与问题**

**1.1.1选题背景**

随着科技的不断发展和社会的不断进步，信息行业与传统工业得到了前所未有的发展并逐步两化融合，整个社会的信息化进程离不开软件行业的产生与发展。到目前为止，软件技术被应用于每行每业，不同业务的软件产品如雨后春竹出现在人的平常生活中。软件技术的产生和发展，正明显地改变着人类的生活和生产方式。

然而，我国由于历史的缘故，工业基础领域相比于国外薄弱，这影响了我国工业软件的发展。其次，国内工业软件产品成熟度不足，稳定性与兼容性更与国外的同类产品全系统、全流程、全生命周期的服务存在一定的差距，目前，国内大多数工业软件系统仅仅可以满足简易的业务场景，并不能形成有竞争力的工业软件产品和服务能力，简而言之，国内工业软件和实际的工业应用需求存在许多差异，这不仅是因为各企业有不同的实际业务与特定的场景需求，而且与工业软件开发过程软件管理的需求分析阶段有密不可分的联系，这给软件开发带来极大的风险，极有可能因为软件达不到验收标准，最终导致整个项目失败。

Gathner Group公司的相关调查结果显示，在北美有40%的软件项目是失败的（Group S，1995）。有调查显示，超过一半的项目运行失控归咎于失败的风险管理，并且大部分软件项目未进行风险监控；超出预定开发周期五分之一以上的软件项目占总项目数量七成左右。在化工软件项目开发中，能如期、不超预算实现项目目标的软件项目仅占15%-25%，不能完成项目目标或仅完成部分目标的占总项目数的25%-35%，有二分之一的项目以失败告终。通过这些资料可以得知，化工业信息系统开发是经典的高风险项目。项目的成本巨大，一旦项目开发失败，不仅对公司在经济上造成很大损失，而且有可能对化工企业的整体运营产生不可扭转的严重后果。

Z公司是国内一家知名的化工产品加工生产企业，为了实现两化融合，增强公司信息化管理的整体水平、增加国际竞争力、打入国际市场等综合原因，公司高层决定自主研发W工业软件项目。

化工企业属于我国传统制造业，企业信息化起步晚，覆盖面窄，化工企业缺少软件开发管理经验在业界内众所周知的，相较于专业的软件公司，化工业软件开发技术更加薄弱，不仅缺少经验丰富的软件开发人员，而且公司管理层缺乏软件系统项目管理经验。所以，化工公司在软件开发与构建过程中比软件开发公司将会承担更大的风险。本研究将系统的分析化工企业软件风险因素、风险优先级、风险应对策略，以提高软件项目的成功率。

结合我过化工软件的实际情况，国内化工业自主研发软件还存在很多需要完善的部分。

主要体现在如下：

(1)风险意识不强。化工行业属于传统制造行业，自研团队缺乏丰富的软件项目风险管理经验，大多数化工企业忽视软件风险或者对风险管控的意识弱，并且团队缺少可以借鉴的关于风险管理的组织过程资产和事业环境因素，极少在软件开发全过程执行风险管理。

(2)研发人员对化工业务生疏。通常情况下，研发人员只处理自己负责的功能模块，对化工行业这种特殊制造业的实际业务理解存在一定的误解，做出来的产品很可能与实际需求存在较大的分歧，容易造成验收不通过、项目返工甚至项目延期等各类事故。

(3)PM专业性弱。通常情况下，化工企业中很多公司的软件研发部项目经理是从其他部门抽调过来的，对软件知识了解是片面的，缺乏关于软件项目管理系统的培训和专业的学习，对项目风险的管理形不成体系。

**1.1.2研究问题的提出**

围绕软件风险管理的主要存在三个问题：第一，软件项目风险因素的识别应该采取何种方式？第二，软件项目风险管理中风险优先级应该如何确定？第三，排序后的风险因素应该采取哪些应对措施？鉴于以上三个问题，本文将以软件项目风险管理为研究对象，探讨软件风险管理中的识别风险因素、风险评估并提出相对应的风险应对策略。以期在拓展软件风险管理研究的同时，为后续研究软件风险管理的学者提供参考。

**1.2研究目的与研究意义**

**1.2.1研究目的**

理论上，在软件项目风险管理研究的基础上，细化到化工行业软件风险管理研究，系统识别和分析化工业全过程软件开发中风险因素，深化化工行业软件风险管理研究，为进一步展开化工行业软件风险管理研究提供理论支持。

实践上，就化工行业如何在软件全生命周期开发过程中识别风险因素，风险评估并制定相应的风险应对策略，更好的发挥风险管理在软件开发过程中的实际作用，以期在为软件风险提出相对应的应对策略的同时，为促进我国化工业软件风险管理的发展提供帮助。

**1.2.2研究意义**

软件项目风险管理是对软件项目需求、立项、设计、开发、验收、收尾、维护各阶段的潜在风险和已知风险进行综合的分析、监控与应对的过程。软件行业是高新科技行业，在理论上将软件项目风险分成多种类型，提出了项目软件风险管理的总体思路，在实际项目开发中，也通过定性与定量相结合的管理风险的方法，实现了各类信息系统。然而，随着时代的发展，技术过于新颖以致研发不能应用、现有的软件不满足现有实际场景需求、核心技术人员离职等各种问题，都会给软件开发过程中带来各种不可预期的风险，项目的开发成果不能满足项目干系人的基本业务需求，导致项目CPI>1,SPI<1的情况屡屡发生，对于这些严重后果的发生，最后在验收项目结束后，项目经理整理经验教训入库公司组织过程资产，事后研究分析后发现，假如公司在项目前期制定风险应对计划、对风险因素识别、风险分析，并采取相应的风险应对措施，部分风险因素是可以避免或者减轻对软件项目的整体影响。综上，成功的软件风险管理可以为软件项目分析出风险因素，并制定相应的措施以应对这些因素发生。

**1.3研究内容与研究方法**

**1.3.1研究内容**

本文基于国内外学者对软件项目风险管理研究，探索性地将化工企业自主研发的软件系统作为研究对象，本文采用问卷调查法与专家判断法相结合的方式对此软件系统进行风险因素识别和风险评估，并针对性地对各个风险提出相对应的风险应对策略，论文的主要研究内容如下：

第一章：绪论。首先阐述了本选题的研究背景，提出了当前化工企业自主研发软件系统项目存在的问题，并在此研究背景下，引出了本文的研究问题。并且同时分析了本文的研究目的和研究意义，然后阐述了本文的研究内容与研究方法，在本章的最后，阐述了本文可能存在的创新点。

第二章：基本概念与文献综述。第一步对研究中涉及到的项目风险管理基本理论、项目管理计划、项目风险识别、项目风险分析、项目风险应对、项目风险控制进行概念阐述，然后对国内外软件风险管理研究进展进行梳理，指出当前研究中存在的不足，进而提出研究内容：Z集团Android项目风险因素识别及应对策略研究。

第三章：Z集团Android软件开发项目介绍及风险因素分析。首先介绍了Z集团的基础资料和W项目的研发背景，其次识别W软件项目所存在的所有风险点，最后对识别出来的风险进行优先级排序和概率影响分析。

第四章：Z集团Android软件开发项目风险因素已被识别，本章通过定性风险分析与定量风险分析相结合的方式，分析出Android的软件项目主要风险。

第五章：Android项目关键风险规避与对策。根据Android软件项目所有风险点对整个软件项目的影响程度，将其划分为关键风险、重要风险、一般风险和可接受风险，并提出对应的风险应对策略与风险规避措施。

第六章：总结与展望。结合Android项目风险管理进行本研究总结。根据本研究得到的结论，对化工行业软件风险识别、风险评估、风险应对策略提供一些参考建议。最后，指出本次研究中的不足，以及后期的研究展望。

**1.3.2研究方法**

（1）文献研究法：广泛地收集与查阅中外文献材料，对化工业软件风险管理之识别软件全过程开发过程的风险因素，对软件风险因素进行优先级排序，并提出相应的软件风险应对策略。

（2）问卷调查法：在明确风险管理内涵和内容维度的基础上，编写调查问卷，通过线上和线下相结合的方式在全国范围内完成500份化工行业的风险因素的问卷调查。

（3）德尔菲法：在确定化工行业风险因素的基础上，组成专家讨论小组，向专家提出所要预测的风险因素发生的概率与影响，对专家的意见进行统计，最后经过计算对化工行业风险因素进行排序整理成表。

（4）案例研究法：通过分析Android项目存在的问题，总结归纳，提出相应的风险应对策略。

**3研究创新点**

本文的研究思路是综合考虑Z公司Android项目与传统IT软件项目有共通性，但与传统IT软件项目之间仍存在独特性，遂以传统IT软件项目风险管理模型作为理论基础， 结合理论研究和实践，采用定性研究为辅，定量研究为主的方式。

本文的主要创新点有：

1.分析软件项目风险管理过程，依次从项目风险规划、项目风险识别、项目风险分析、项目风险应对、项目风险跟踪和项目风险监控出发，对每个过程详细分析。

2.使用层次分析法与STAR模型相结合的方法，进行软件项目风险定量分析。第一步通过层次分析法构造判断矩阵，计算特征值和权重，根据权重对风险进行排序；第二部对项目目前可以实现风险应对的风险点采取行动与措施，重新构造判断矩阵，计算特征值与权重，以证明所采取行动和措施有效。

3.针对Z公司软件实施过程中，存在很多风险，本文罗列出其中的一些关键风险并对风险产生进行分析，最后，提出各风险点的应对办法与规避措施。

4.Z公司属于传统化工企业，Android项目是Z公司首个自主研发的信息化项目，很少有实证研究将明确的风险管理转移到工业，本项目的风险管理研究将为化工企业提供“两化融合”的现实理论与实践经验。

**第二章 基本概念与文献综述**

**2.1 Android系统的基本概念**

**2.1.1Android系统的起源**

Android是由Google推出的一款针对于手机的操作系统。它的Logo是一个绿色的小机器人，它的名称来源于科幻小说《未来夏娃》里面一个名字叫做android的小机器人。最初，一个名为android的公司创作了android系统，这家公司的CEO与研发团队主要将android系统应用在移动智能手表，后来android公司被Google收购，android系统也逐渐从智能手表移植到手机。2006年，Google发布了android 1.0系统并将其开源，随着android系统健康快速地发展和不断的升级，越来越多的手机厂商选择android系统作为手机操作系统。

**2.1.2Android的四大组件**

Android的四大组件分别是Activity、Service、Contentprovider和BroadcastReceiver。

Activity的主要作用是与用户交互，Activity类似于一个窗口，用于显示Android定制化界面。开发者在Activity的OnCreate方法中绑定xml，并Activity里面实现逻辑代码处理点击、触摸、滑动等一系列事件。

Service是Android系统的服务，没有任何界面，即使App不在前台运行，Service依然活跃在后台。Service一般可以通过在Activity中编写代码调用启动，主要有两种启动方式，分别是StartService和BindService。两种方式虽然都可以启动Service，但由于其生命周期的不同存在些许差异：使用StartService方式启动服务，即使在与之绑定的Activity被销毁时，Service仍然作为一个独立的进程在后台运行，而使用BindService方法Service会随着绑定的Activity的被销毁而被终止。Service是Android系统的核心，Android系统之所以能够稳定的运行，与Framework层的PakageManagerService、ActivtyManagerService、WindowMangerService等Service进程有非常重要的关系，如果其中任意一个Service运行出错，Android系统就可能死机、重启或无法启动的情况。

ContentProvider的字面意思是内容提供者，当一个App应用A想要访问另一个App应用B的数据库时，这涉及到数据安全性问题，每一个App是一个单独的进程，进程间直接获取数据代码中会编译不过或获取到Null数据或者抛异常后APP闪退。正确做法是应用B通过实现ContentProvider必须实现的六个抽象方法并对外暴露接口，应用A带权限访问应用B的数据库。

BroadcastReceiver是Android系统的广播，广播不仅可以在进程内通信与传送信息，而且可以实现进程间通信。以优先级划分，广播分为有序广播和普通广播，以注册方式区分，广播分为静态注册方式与动态注册方式。广播事件发生在用户使用手机时的时时刻刻，例如开机广播、息屏广播、收到短信时声音提醒等等。

**2.1.3Android的开发语言**

2017年之前，Android的首选开发语言是Java，Java是印度尼西亚爪哇岛的英文名称，该岛屿盛产咖啡，Java之父James Gosling[]{[1]Nicolle, L. James Gosling[J]. Computer Bulletin, 1998, 40(2):16-17.}起初用于于1995年发布了Java语言，与C语言面向过程编程不同，Java是一门面向对象的语言，这种编程模式带给Java语言多个优点：

a.方便代码复用复写以及重构

b.方便代码的编写与维护

c.JavaBean对象易于解析，提升运行效率

2017年，Google开完I/O大会，官方宣布Kotlin讲作为Android开发的第一语言，相较于Java语言，Kotlin更适合用于Android开发。Kotlin与JDK完美兼容，并且安装插件后，Java代码可以转换为Kotlin代码，除此之外，Kotlin是一门轻量级的语言，其代码较Java代码更为简洁，减轻代码的冗余。

**2.1.4Android的开发工具**

Android系统发展初期，adt(android development tool)是使用Oracle公司的Eclipse作为软件开发工具，开发者在使用eclipse作为Android开发工具时需要自己下载JDK(Java Development Kit)和Android SDK(Software Development Kit)，由此可见，在进行Android开发前期的配置工作过于复杂。

2013年，Google首次提出自研开发工具Android Studio这一概念并于第二年12月8日上线Android studio 1.0稳定版本，开发者只需在官方网站下载开发工具一键安装，Android studio会自动下载Android开发所需要的Java JDK与Android SDK。

**2.1.5Android设计架构**

最初的Android设计架构是MVC模式，即Model-View-Controller(模型-视图-控制器)，结构模型如图所示。开发工程师在模型层编写代码处理网络数据或本地数据库数据的增删改查，Android数据库一般是轻量级的，例如Sqlite数据库和JSON数据库。在Activity中进行View层的绑定，View层是一个xml格式的文件，该文件中最常用的布局方式是RelativeLayout、LinearLayout、FrameLayout和TableLayout。Controller层代码编写在Activity或Fragment中，开发者会在Activity和Fragment中进行数据操作、事件监听、界面设置。这样导致Activity和Fragment类中代码量非常大，耦合性相对比较高，而且不利于其他开发工程师的阅读与理解。

为降低耦合性，MVP设计架构被引入到Android开发。MVP是Model-View-Presenter（模型-视图-）的缩写，具体结构模型下图所示。MVP模式是基于MVC模式的优化与改善版本，由于View层与Model层耦合性高，在MVP模式中，将View层所需要实现的业务逻辑方法抽象到Presenter层，View层实现Presenter层所有方法，在View层调用Presenter层方法，通过这样的方式解耦Model层与View层。MVP设计架构主要应用于大型项目，当业务逻辑较少时仍可采用MVC模式。

最新的Android设计架构是MVVM模式，即Model-View-ViewModel，结构模型如图所示。MVVM模式与MVP模式结构上很相似，区别在与MVVM使用databinding将Model层与View层进行绑定，其中一层数据的变化会影响的另一侧数据相同的变化。ViewModel层是View层与Model层沟通的桥梁，不参与UI界面的刷新，只关注业务逻辑的处理。因此，相比于MVP模式，MVVM模式解耦的效果更彻底。

**2.2项目风险管理的基本概念**

**2.2.1风险的定义**

PMBOK[AXELOS, Managing Successful Projects With PRINCE2®, sixth ed., AXELOS, 2017]和PRINE2[PMI, A guide to the project management body of knowledge (PMBOK® guide),sixth ed., Project Management Institute, Inc., Newtown Square, PA USA, 2017.]中的风险被定义为一个不确定时间或一组不确定时间，如果他们发生，将对一个或者多个项目目标产生负面或积极的影响。不确定性带来风险，也就是潜在的损失，风险通常被定量地定义为概率乘以潜在损害成本，Boehm（1989）将这种成本称为“风险暴露”。然而，Charette（1989）提出每个潜在的损失都必须用一个情景来确定。因此风险分析定义了可能出错的情况（S）、出错的可能性（L）和出错的值，得到以下公式：



这个定义还有一个缺点，它将低概率高成本的情况等同于高概率低成本的情况。为了克服这一局限性，另一学者添加了效用或结果（O），这是一种度量重要性或严重性的方法，得到了以下新公式：



2.2.2风险管理的定义

风险管理被定义为一组协调的活动，允许项目干系人对风险进行指导和控制。这就需要系统地应用原则、方法和过程来、制定风险管理计划、风险识别、风险分析、风险评估和实施风险应对的任务，以及关于与每个利益相关者进行的活动的沟通；风险管理的目标是在风险因素发生并成为潜在威胁之前，识别、指导并消除它们。

（1）项目风险管理计划

项目风险管理计划一般制定于项目启动前或者项目刚刚启动的阶段，制定项目风险管理计划的目标是判断组织内部对风险项目风险的发生概率和影响的预期与项目实际风险的概率与影响是否一致。项目风险管理计划为项目风险管理提供适当的工具与技巧，项目风险管理计划还定义了如何对应风险的负责人和相应的风险应对方法、风险上报的流程。

（2）项目风险识别

风险过程可以发现、识别和描述不同的风险，并识别每个原因和潜在后果。Sipayung认为[56]将风险识别定义为风险评估的早期阶段，通常允许确定应用开发项目所涉及的风险。这个阶段应该能够识别不同的风险元素，然后保存在风险元素的列表，因为他们将建立标准和测量，使我们能够确定风险的概率，风险的影响因素和风险进化的演变。这一阶段还提供了一个程序，允许计算风险暴露值。同样，Jaiswal等人[]（cost and risk of software project using function points, in: N. Meghanathan,D. Nagamalai, N. Chaki (Eds.), Advances in Computing and Information Technology. Advances in Intelligent Systems and Computing, Springer Verlag,2013, pp. 77–86, , h）认为，正是在这个阶段，应该编制一份清单，其中包含项目中存在的可能危及其成功的具体风险因素。Elzamly[]（[1] Elzamly A , Hussin B . Modelling and Evaluating Software Project Risks with Quantitative Analysis Techniques in Planning Software Development[J]. Journal of Computing & Information Technology, 2015, 23(2):123-139.）提出了风险识别阶段的思想，包括三个阶段，(a)制定风险计划：它的目的是建立所有参与软件项目的人的责任(例如，项目领导，软件开发团队成员，项目干系人等等)。它还产生了组织方案，该方案应该遵循使管理风险成为可能的迭代方法；(b)风险识别阶段，该阶段能够获得一般的风险期望，同时也识别软件风险因素和管理策略，此外，它阐明了风险对度量的影响——例如对数量和质量技术的影响(c)风险优先级，这是一个允许将风险程度分类的阶段，由特定风险的范围、概率和影响决定。此外，Keshlafy Hashim[]（A.A. Keshlaf, K. Hashim, A model and prototype tool to manage software risks, in:C. T.Y., T. T.H. (Eds.), 1st Asia-Pacific Conference on Quality Software, APAQS2000, P.O.Box 3633, Tripoli, Libyan Arab Jamahiriya, Institute of Electrical and Electronics Engineers Inc., Software Engineering Research Laboratory,Information Department, Industrial Research Center, 2000, pp. 297–305.）认为，风险识别是一个阶段，应该能够识别项目中的存在的所有潜在风险；它可以从两个不同的角度出发:

1. 使用可以在任何类型的软件开发项目中使用的风险数据；
2. 在可能影响计划项目的特定项目中使用的风险数据。

风险识别发展至今，实践中已经有许多风险识别的方法，项目中较为常见的风险识别方法主要有以下几种：

(a)故障树分析法

(b)德尔菲法

(c)头脑风暴法

（3）项目风险分析

这一过程使我们能够了解风险的性质，同时也确定了风险的水平（以所涉风险的后果和原因来表示）。此外，对于风险评估也是如此，它还承担着为风险评估提供基础的任务，作为解决、处理每种风险最适合的策略和方法等各类决策的输入。林德霍尔姆·[57]，在她关于风险分析的讨论中，认为这是与确定的每个危险情况相关的风险；它通过风险严重程度和风险概率的尺度进行估计，其中风险值是通过将风险的严重程度乘以概率值来定义的。Elzamly等人在他们的贡献中指出，风险分析有助于分析涉及那些已确定的风险的风险概率及其后果。他们还认为，在风险分析中，有可能对影响、暴露和风险之间的关系进行估算，并分析缓解风险的备选办法和实际使用的缓解战略。Haisjackl[]([1] Haisjackl C , Felderer M , Breu R . RisCal -- A Risk Estimation Tool for Software Engineering Purposes[C]// Software Engineering & Advanced Applications. IEEE, 2013:292-299.)证实，在风险分析中，风险暴露可以通过应用某些指标来计算。Jaiswal等人[]（[1] Jaiswal A , Sharma M . Expert Webest Tool: A Web Based Application, Estimate the Cost and Risk of Software Project Using Function Points[J]. 2013.）指出，在风险分析中，可以使用性能模型、网络分析和成本模型等来评估所确定的每个风险要素的损失幅度和概率。

（4）项目风险评估

风险评估是将风险分析结果与每个标准或职权范围进行比较的过程，风险评估允许对风险进行优先排序，并就风险处理的行为做出决定。Lindholm[]（C. Lindholm, Involving user perspective in a software risk management process, J.Softw.: Evol. Process 27 (2015) 953–975）认为风险评估是对每一个风险与降低风险的关系的决策，这些决策是通过基于控制措施的标准做出的，这些措施可以进行讨论后部署，以达到降低风险的效果。如果执行风险应对措施过程中发现新的风险，应该将新的风险记录并进行分析，以便确定风险的整体规模。Elzamly（2015）认为，风险评估的目的是确定识别出的所有风险的优先级别，以便项目管理者可以实施不同的风险应对策略，同时也防止风险成为项目延期完成的原因。

1. 项目风险应对

风险应对被视为一个可以通过选择应对计划的实施来改变风险的过程，这个过程旨在消除风险或改变风险发生的概率或风险发生造成的后果或三者兼有。Islam[]([1] Islam S , Mouratidis H , Weippl E R . An empirical study on the implementation and evaluation of a goal-driven software development risk management model[J]. Information & Software Technology, 2014, 56(2):117-133.)认为，风险应对的重点是及时采取应对风险的行动从而实现风险应对目标，风险应对的主要目标也是尽快获得风险控制，最好是从需求分析阶段开始。风险应对使得为减轻风险和满足项目目标而采取的控制行动进行建模、分析、推理和跟踪。Elzamly(2014)关注风险应对的四种策略可以应对风险，这些分别是：

A.风险回避

当风险发生的概率非常大，对项目影响巨大的时候，以下风险应对措施不足以解决当下风险，此时，需要采取措施规避风险。在风险尚未发生的情况下，项目负责人可以采取改变项目目标或者直接放弃项目，这样的好处是可以使风险的概率降至几乎为零，且对项目的整体运行无实质影响。风险回避是一把双刃剑，它可以降低公司的项目成本，但是也可能由于消极地放弃项目而使得公司错过一个优秀的项目。总之，在执行风险回避时，应该更加谨慎。

B.风险转移

当公司内部不愿意承担某类或者某些风险时，公司会通过其他方式把项目存在的风险转移到其他团队或者其他机构，这旨在与其他团队或者机构分享风险。一般地，比较常见的方式是通过外包或者购买保险实现风险转移，公司自身不承担对应风险。然而，这种方式有可能会给公司带来新的风险，比如外包公司延期交付项目风险、与保险公司的合同内容签订不明确导致保险公司拒绝赔付风险。

C.风险减轻

项目经理或者项目干系人采取一系列措施，旨在减少风险的概率或影响或两者兼有。这种风险应对措施的目的在项目发生之前或者风险刚发生时及时付诸行动，此刻风险发生的成本更容易被降低，其可以有效杜绝风险发生后造成的损失过于庞大。

D.风险接受

风险接受是指在风险规划初期或者风险发生时，选择接受风险发生时所造成的的损失和影响。选择接受风险的原因是能够分析风险发生的后果和影响，风险发生的概率较小并且可能发生的风险处于可控的范围内。虽然风险接受任由风险发生，但这不代表风险发生时，项目负责人不需要对该风险采取任何行动，项目负责人需要准备项目储备资金，当风险发生时需要采取积极的措施。如果在风险发生时，采取消极的风险接受方式，可能会使风险转换为项目的关键风险，对更替项目造成极大的损失甚至项目失败。

Keshlaf和Hashim(2001)在一个被称为控制的模型阶段提到了这种风险治疗的活动；这是基于风险的严重程度，以及适合降低风险的技术的执行。在这个阶段，可以考虑一个缓解、应急或危机的计划。他们还建议，在应用减少技术后，应对风险采取的行动进行重新评估、重新评估和记录。

在许多研究中，风险管理被视为影响项目成功的主要因素之一[12-15] 。这种成功将在很大程度上取决于风险经理和项目经理[13]的经验和能力，以及与不同相关各方的感知和期望相关的因素，以及旨在解决风险[16]的行动的同步性。同时，风险管理不善或完全缺乏风险管理也是[15,17,18]项目失败的主要原因之一。PM对参与项目管理的专业人员进行的一项研究表明，对风险和机会的不良管理仍然是项目失败的主要原因之一；除了这些原因之外，项目经理和组织还缺乏应对所出现的挑战的能力。在本报告中，PMI还强调了标准化项目管理实践的使用的要点 能够降低风险，并获得更好的结果。这是 只要这些做法在整个组织中推广；这就产生了降低风险和控制成本的能力，也使其能够适应不断变化的市场状况，并提高每个项目的价值交付 。

**2.2.3 软件项目风险管理经典模型**

**2.3国内外研究现状及文献综述**

**2.3.1国外研究现状**

20世纪80年代开始，国际上软件风险管理的研究逐渐兴起，软件项目管理第一次引入软件项目风险管理这一概念。Tom DeMarco[]{[1]・, 迪, 马,等. 与熊共舞:软件项目风险管理[M]. 清华大学出版社, 2004.}在《软件项目风险管理》中阐述了风险管理的九大应对措施，他认为应该在软件项目执行之前调查出软件项目将要面临的风险，并确认软件项目有哪些核心的风险。风险管理之父Barry Boehm认为“成功的项目经理都是优秀的风险经理”，Boehm 的软件风险管理理论是一门融入所有开发人员实践研究的概念，1989年，Boehm 提出了用于软件开发的由风险驱动的螺旋式模型[]([1] Boehm B W . A Spiral Model of Software Development[J]. Computer, 1988, 21(5):61-72.)，。Robert Charette结合了日本质量概念的 Kaizen（持续改进）、Kansei（扩大利益环境）和 Keiretsu（控制供应商），从不同的角度构建了动态的螺旋式风险管理。Robert Charette 认为由于风险伴随软件项目的立项到收尾，所以风险管理是一个循环往复的过程，每个步骤之间应当是相互重叠并反复交错的[]([1] Charette R N . Software Engineering Risk Analysis and Management[M]. McGraw-Hill Book Co, 1989.)，也就是说，软件风险管理应该贯穿软件项目的全生命周期并循环往复地进行风险识别、风险评估、风险应对以及风险控制。随着科技的进步和时代的发展，项目管理人员和企业更加重视风险管理，微软公司将项目风险管理分为风险辨识、风险分析、风险管理计划、风险追踪、风险监控五个阶段。PMI对最近对参与项目管理的专业人员进行的一项研究表明，对风险和机会的不良管理仍然是项目失败的主要原因之一，除此之外，项目经理和组织还缺乏应对所出现的挑战的能力。PMI还强调了标准化项目管理的使用要点是能够降低风险并获得更好的结果，只要标准化项目管理的做法能够在整个组织中得到推广，就能够产生降低风险和控制成本的效果，也能够适应不断变化的市场需求并提高每个项目的价值交付。

21世纪以来，随着互联网技术的高速发展，从项目可行性分析到项目的正式收尾，项目的开发方对于项目的开发进度、项目成本和项目质量的要求越来高，Kevin[]([1] Vlaanderen K , Jansen S , Brinkkemper S , et al. The agile requirements refinery: Applying SCRUM principles to software product management[J]. Information & Software Technology, 2011, 53(1):58-70.)邓然将敏捷方法SCRUM原理应用于软件产品管理，产品经理能够轻松应付敏捷开发环境下的项目需求，并能够将SCRUM敏捷开发的应用于组织中的实际项目。

**2.3.2 国内研究现状**

由于管理技术、社会环境等原因，我国的风险管理发展晚于国外，在1980年以前，我国风险管理的绝大多数理论是借鉴西方国家的，缺乏学者更深刻的研究。在全球，项目风险管理的实践已普遍存在，然而，在我国仍然有很多人对项目管理知之甚少，甚至一无所知，项目风险管理的推进任重而道远，需要更多的研究人员付出更多的实践工作。这一事实最明显的体现是国内缺少专门介绍软件项目风险管理类的书籍[2]。国内开启风险管理研究[3]的时期是在十九世纪80年代后期，清华大学教授郭中伟发布了著名书籍《风险分析与决策》，标志着中国学术正式将风险纳入研究对象。

经过多年努力，我国投入了大量的人力、物力、财力，在软件项目风险管理方面也取得了许多的成果。1989年，研究人员吴鸣在其研究著作《经济风险论——从风险角度对中国现实经济问题的研究》。1999年，吴明在其著作《投资项目风险分析》中引入了风险管理的概念，详细阐述了投资项目的风险分析的方法与基础理论。2006年，中国电子工业出版发行的图书《中国项目管理知识体系》(C-PMBOK2006)，书中第7.7节详细阐述了项目风险管理的四步骤。

赵有强[](赵有强. 风险管理在管理软件开发项目中的重要作用研究[J]. 科学与财富,2020(7):395.)介绍了软件风险管理的一些小技巧，可以有效地节约软件全生命管理过程中的成本、节省不必要的时间成本并减少项目资源的无效投入。郭晶[](郭晶,万向,张慧,等. M公司软件开发项目风险管理研究[J]. 科技资讯,2021,19(9):141-143,147. DOI:10.16661/j.cnki.1672-3791.2103-5042-6012.)等人提出一种新的风险管理模型应用于互联网公司软件项目，并在他们的研究中，采用多种研究方法对软件项目中潜在的风险进行识别、分析及应对。温林芝[](温林芝. 基于生命周期的软件开发动态风险管理[J]. 电脑知识与技术,2021(23).)在其研究中认为在软件开发过程中的不同的生命周期中，应该动态的分析执行风险管理。

**第三章Z集团Android软件开发项目介绍及风险因素分析**

**3.1 Z公司简介**

Z集团是江苏省一家著名的集农药研发、生产和销售为一体的国家农药生产化工企业，是国家高新技术企业。公司成立于1988年，经过三十年持之以恒的奋斗，公司于2018年在A股成功上市。2021年，Z集团营业收入总计12亿人民币。目前，公司在全国范围内拥有超过3000名员工，在江苏、上海、安徽、湖南等26个省份设立多个子公司。

2020年年初集团在原有规模上，投入大量资金增设软件部门和建设新的化工厂区。公司坚持走“科技型企业”路线，集团进一步扩张信息化业务，期望实现信息化带动工业发展，为工业节约成本，工业化促进信息科技行业进步的目标。

Z公司未来主要关注三个方向：新产品的研发、人才培养、信息化保障安全生产。Z公司是农药生产和销售领域的领先企业，拥有国内农药企业四分之一的市场份额。某证券公司研究公司的业绩表明，2019-2021年全国范围内化工市场的总收入分别是8.2亿、9.9亿、12亿，年度平均业绩增长超20%，Z公司正处于高速发展与扩张阶段。2021年公司决定建设信息化平台和与之交互的Android移动应用APP，以期实现 “两化融合”,提升企业核心竞争力，在国际的舞台上崭露头角。

**3.2 Android软件项目简介**

Z公司是一家传统的化工企业，2017年使用某公司的SAAS一体化软件系统，系统使用过程中，遇到诸多问题。例如：服务器宕机必须联系该公司售后重新启动服务器、公司员工个人信息掌握在该公司、租赁费用高昂且得不到应有的技术支持、软件功能与实际需求存在差别等。

由于上述问题难以从根本上解决，2020年年底，Z集团高层经过会议讨论，W项目正式立项。于2021年初3个月，集团为研发Android项目，招聘软件开发和管理人才超过50人，其中高级产品经理1人，高级项目经理1人，Java Web工程师15人，Android端高级开发工程师5人，Android中级开发工程师10人，Android初级工程师5人，前端Html5工程师6人，测试工程师10人。集团的目标是建设集成重大危险源监管、人员在岗在位管理、文档管理等一体化Android软件。

化工厂采用了大量传感器和其他数据采集设备对重大危险源要素以及可燃有毒气体进行温度、压力、电流、电压进行实时监测，便于实时掌握厂区的生产状态，防患于未然。

化工生产具有易燃、易爆、有毒、高温、高压等危险，稍有不慎很容易发生火灾、触电事故，造成很大的损失。人员在岗在位管理首要任务是将集团所有员工的在岗信息和录入到一体化平台中，其次将人员定位模块的虚拟ID与对应员工在信息化平台上进行绑定。负责人可以登录手机App，查询公司所有员工的人员实时位置和行动轨迹，在发生安全事故时有利于对未能够及时撤离的公司员工实施救援工作。

文档管理包括教育培训、制度化管理、应急预案管理，公司员工自行使用手机登录App进行三级安全培训和日常培训，员工完成后提交试卷，App会请求网络数据，根据服务器题库里的正确答案对试卷进行打分，不合格的员工，App在指定时间内推送给不合格员工复试，知道员工达到合格线。使用该App节约了管理人员批阅试卷的人力成本，而且这还有助于公司节约资源，减少了对纸质试卷的使用，公司所有员工可以登录手机App查询公司制度和应急预案，与查阅公司档案室纸质文档相比，节约了大量的时间成本。

**3.2 Android软件项目风险因素识别**

**3.2.1需求风险因素**

1. 需求不明确风险因素

Nguyen[]([1] Seiffert B M . α-Complementation assay for HIV envelope glycoprotein-mediated fusion[J]. Virology, 2004, 319(2):343-352.)认为项目需求在收集时应当被清晰地定义，需要简单化项目干系人之间复杂的利益关系、减少利益冲突，从而在项目初期制度项目计划产生正向推动的效果。由于Z集团属于传统的化工制造行业，软件项目经理和产品经理对Android应用端较为熟悉，而对化工领域的相关知识相对来说较为陌生，在与项目干系人手机需求时可能发生偏差，导致最后实现的功能与项目干系人所描述功能存在很大的差异。

1. 需求蔓延风险因素

Android软件项目在开发过程中，项目干系人对项目目标存在质疑，增加大量项目计划阶段未提及的需求，而且类似的需求可能会被无休止的添加进需求文档并投入大量的人力物力实施这些需求，每个项目的资金是有限的、项目周期是是固定的，需求蔓延的最终结果会导致成本超支、进度延误。

(3)需求持续变更风险因素

项目执行阶段前，项目经理已对项目进行WBS分解，将分解的工作包下发给相应的开发人员。项目进入实施阶段，项目干系人经常对之前已确定需求提出新的意见和建议，并希望修改或者推翻需求文档，会导致项目开发人员循环进入返工修改，影响项目关键里程碑。

**3.2.2技术风险因素**

(1)技术过于新颖风险因素

每年Google公司组织的I/O大会带来Android技术的更新，Android1.0已经升级到Android10，Android SDK和Android API也经过了多次的更新。如今Android开发语言已经逐步从Java转换到Kotlin，Android设计架构由最初的MVC模式调整成MVVM模式。公司新招的Android开发工程师来自不同的公司，选择过于新颖的开发技术，可能会使部分使用传统技术的开发人员难以跟上团队的步伐，造成项目实施前期无法推动、进度缓慢的后果，这可能会改变Android项目的关键路径。

(2)网络攻击风险因素

Android应用许多数据通过向服务器网络请求的方式获取数据。2021年11月，Log4j被发现存在远程代码执行漏洞，国内不少公司因为Log4j漏洞被黑客攻击，导致服务器数据库被加密、数据库被删、公司被勒索钱财，甚至黑客使用公司服务器做为服务器肉鸡，攻击其他公司服务器。除此之外，xss攻击、SQL注入攻击一直是黑客钟爱的入侵方式。服务器的安全性需得到保障，一旦服务器被攻击，Android应用无法登录、界面空白的情况很容易发生。

1. 开发工具的选择

Android开发工具有最初的adt和目前最为流行的Android Studio，还有较为小众的IntelliJ IDEA，每一个开发工具都有优缺点，只有选择合理的开发工具，才能保证项目顺利运行。

1. 开发语言的选择

Java、Kotlin是当前Android开发中占比最高的两种语言，其次是JSP和Html5。选择不合适的开发语言，每位开发者阅读和分析代码时会增加障碍，不利于开发人员快速定位问题，导致时间成本增加，容易造成进度延误。

**3.2.3人力资源风险风险因素**

(1)人员离职风险因素

为留住人才，各地出台一系列政策。这是因为人才是每个城市的第一资源，建设好城市离不开各专业人才坚持不懈的努力，对公司也是一样的，为保障公司项目平稳地得到推进，应尽量减少技术人员的频繁离职，项目开发一半，核心技术人员离职特别容易对项目造成不可挽回的损失，重新招员工需要时间不确定，即使迅速有人员替补，熟悉代码和项目业务也需要必要的时间，并且公司关键开发人员的离开容易造成士气下降、项目进度落后等后果。  
 (2)开发人员的团队协作能力

开发人员的软件开发能力固然重要，开发人员在展现个人能力的同时，需要加强与团队其他人员的合作能力，同事之间互相帮助，取长补短，发挥团队精神以期提升团队的工作效率。在软件项目开发过程中，开发人员的团队协作能力相当重要。  
 (3)项目成员缺少主动性

项目成员如果缺乏责任感，工作效率不高，对负责的功能模块不能够按时完成，项目开发人员的主观能动性是项目能否成功的关键因素。

(4)项目经理的专业能力

项目经理是否有类似的Android软件开发管理经验是这个项目是否能事半功倍的关键性因素，在项目执行过程中，项目经理作为本软件管理者的管理能力是项目能够达到项目目标的核心，而混乱的管理会使项目走向深渊。

(5)团队成员沟通不足

项目经理与开发人员、开发人员之间的沟通和交流对项目的成败起着至关重要的作用，一个团队如果缺乏沟通，引起的矛盾、信息阻塞导致项目不能顺利进展的情况时有发生，缺乏沟通的团队往往会造成团队成员没有明确的项目目标，团队成员之间存在隔阂、难以达成合作共赢、团队效率低、项目成本增加、进度延迟、质量下降。

**3.2.4管理风险因素**

(1)项目经理专业性不强

项目经理缺乏经验，对软件项目管理的能力不足，文档整理不规范、对技术人员的具体工作不熟悉、对项目进度控制、质量控制、成本控制的能力不足。  
 (2)缺乏公司高层支持

公司高层管理对项目的支持力度和积极参与直接影响一个项目的成败，如果一个项目缺乏公司高层管理人员的持续支持，软件项目的研发工作将受到非常大的影响。

**3.2.5用户风险因素**

(1)用户预期过高

Z集团Android应用的用户是本集团全体员工，此应用的设计初中是OA办公、简化员工的工作内容、方便员工办公，Android应用1.0版本至少满足合理的界面设计、流畅的界面切换、整体功能无bug等。用户会与各大手机应用市场上商业化APP比较，他们对Android应用要求高于工业化软件的实际需求。Android项目开发过程中，不切实际的期望与需求，会增加项目开展的不确定性。  
 (2)用户与软件开发人员有冲突

集团用户仍处于传统工业的思维，他们担心技术创新带来的不确定因素，有用户认为集团接入信息化软件可能会取代他们从事的工作，所以有员工对信息化平台开发存在抵触情绪。软件开发人员作为技术和项目的推动者，在推动过程中遭到部分用户的抵制，两者容易发生冲突，部分用户对软件开发人员的否定态度和行为将会使得Android软件项目的进展不顺利，进而影响工期。

（3）用户参与度不足

化工业Android软件定制化需求较多，可复用的界面和代码较少，因此用户对该项目的重视程度十分重要，对项目能够顺利开展与推进有极大的影响。Android项目界面是否能符合用户习惯、功能使用效果是否满足用户需求，都需要用户的积极参与和配合测试，如果用户参与度不够，将可能增加集中测试阶段的时间成本。

**3.2.6供应商风险因素**

(1)供应商延迟交付风险  
 由于行业原因，Z集团实时记录员工的行动轨迹和位置，以防止发生重大事故时能定位受困员工的位置，方便应急队伍前往救助，因此，公司将选择一家人员定位卡供应商，Android应用开发工程师将在收到供应商部分样机后先接入人员定位卡SDK，在成功接入后，进行阶段性测试，如果出现供应商在交付合同中规定的时间内无法提供所有产品，阶段性测试将无法如期进行，影响项目进度。

(2)供应商技术支持力度不够

Android开发工程师接入人员定位卡SDK时，根据在供应商官网文档下载的SDK文档接入第三方SDK时，遇到开发人员无法解决的问题需要供应商技术支持，技术支持总是反馈不及时，导致开发人员无法快速将问题顺利解决，导致项目进度落后。

**3.2.7组织环境风险因素**

(1)项目资源向其他项目偏移  
 在项目开发过程中，公司的战略目标发生改变，新实施的其他项目优先级高于目前的Android应用项目，这意味着这个项目在整个集团的地位下降，从而导致集团会优先将这个项目的资源投入到优先级更高的项目中；或者如果集团对于本项目的投资效益明显降低时，集团对于本项目的投资金额也会相应的减少，项目也会因此发生项目执行进度缓慢甚至出现项目进度延迟。

(2)软件经验教训文档少

集团是化工制造业企业，软件开发项目罕见，因此以前的项目文档中，基本不存在软件开发过程中可以借鉴的文档。

**3.2.8政策法律风险因素**

(1)Google公司对中国Android取消授权  
 中美贸易战以来，美国将中国的多家中国上市公司列为制裁对象，国际形势正在悄然发生变化。由于美国杜比公司不提供专利授权，华为公司不得已退出很多芯片项目。Android系统也是美国Google研发的手机开发系统，如果美国对Z集团的取消Android授权，本软件项目将终止研发。

(2)Android应用无法上架应用市场

由于各方面原因，Android Apk经过审核之后难以通过工商管理机构，App需要经过整改才能无法上线各大应用市场，返工导致进度延误、成本上升。

1.技术风险 2.人员离职风险 3.需求蔓延风险 4.合同风险

5.政策法律风险 6.客户期望过高风险 7成本超支风险 8 进度缓慢风险

9.沟通风险 10.供应商风险 11.用户对平台的认同风险 12.数据丢失风险

13.网络攻击风险 14.

4主要风险 ：（1）技术风险 （2）人力资源风险 （3）管理风险

（4）需求风险

（1）技术风险：所用技术过于新颖；服务器被入侵风险；Web漏洞风险

（2）人力资源风险：人员离职风险；缺少关键技术专家风险；

（3）管理风险：项目进展缓慢风险；项目支出超标风险；交流风险；

（4）需求风险 需求蔓延风险 需求变更风险

次要风险：（5）政策风险 政策法律风险

**第四章Z集团Android项目风险分析**

**第五章Z集团Android项目关键风险规避与对策**

**第六章 总结与展望**

**参考文献**

[1] Group S. CHAOS: Charting the Seas of Information Technology. 1995.

[2]薛华成. 管理信息系统(第三版)[M]. 清华大学出版社, 2003.

[1] Boehm B W . A Spiral Model of Software Development[J]. Computer, 1988, 21(5):61-72.

[1] Charette R N . Robert Charette on Risk Management[J]. The software testing & quality engineering magazine, 2003, 5(2):p.10-11.