智能演讲起草系统风险评估报告

目录

[智能演讲起草系统风险评估报告 1](#_Toc6577)

[一、风险综述 1](#_Toc25157)

[1、软件管理对到软件的影响 1](#_Toc14880)

[二、项目管理的风险 2](#_Toc2105)

[1、软件项目管理的风险 2](#_Toc7589)

[三、软件技术风险 3](#_Toc7211)

[四、软件过程风险 3](#_Toc27239)

[1、软件需求阶段的风险 3](#_Toc20412)

本文主要针对智能演讲起草系统开发过程中涉及到的风险，包括在软件开发周期过程中可能会出现的风险以及软件实施过程中外部环境的变化可能引起的风险等进行评估。在文中对所提到的风险都一一做了详细的分析，并提出了相应的风险回避措施。由于风险是在项目开始之后才开始对项目的开发起负面的影响，所以风险分析的不足，或是风险回避措施不得当，都很有可能造成本系统开发的失败。风险分析是在事前的一种估计，凭借一定的技术手段和丰富的经验，基本能够对项目的风险做出比较准确的估计，经过慎重的考虑提出可行的风险回避措施，是避免损失的重要环节。

# 一、风险综述

任何软件的开发，其主要风险均来自两个方面：一是软件管理，二是软件体系结构。软件产品的开发是工程技术与个人创作的有机结合。软件开发是人的集体智慧按照工程化的思想进行发挥的过程。软件管理是保证软件开发工程化的手段。软件体系结构的合理程度是取决于集体智慧发挥的程度和经验的运用。

1、软件管理对到软件的影响   
　　软件是否能够按工期的要求完成：软件的工期常常是制约软件质量的主要因素。很多情况下， 软件开发商在工期的压力下，放弃文档的书写，组织，结果在工程的晚期，大量需要文档进行协调的工作时，致使软件进度越来越慢。软件的开发不同于其他的工程，在不同的工程阶段，需要的人员不同，需要配合的方面也不同，所有这些都需要行之有效的软件管理的保证。

软件需求的分析是否深入透彻：软件的需求是确保软件正确反映用户的对软件使用的重要的文档，探讨软件需求是软件开发的起始点，但软件的需求却会贯穿整个软件的开发过程，软件管理需要对软件需求的变化进行控制和管理，一方面保证软件需求的变化不至于造成软件工程的一改再改而无法按期完成；同时又要保证开发的软件能够为用户所接受。软件管理需要控制软件的每个阶段进行的完成度，不能过细造成时间的浪费，也不能过粗，造成软件缺陷。   
 软件实现的技术手段是否满足性能要求：软件的构造需要对软件构造过程中的使用的 各种技术进行评估。软件构造技术通常是这样：最成熟的技术，往往不能体现最好的软件性能；先进的技术，往往人员对其熟悉程度不够，对其中隐含的缺陷不够了解。软件管理在制定软件开发计划和定义里程碑时必须考虑这些因素，并做出合理的决策。   
　　软件质量体系是否能够被有效地保证：任何软件管理忽略软件质量监督环节都将对软件的开发造成巨大的风险。而制定卓有成效的软件质量监督体系，是任何软件开发组织必不可少的。软件质量保证体系是软件开发成为可控制过程的基础，也是开发商和用户进行交流的基础和依据。  
2、软件体系结构对软件质量的影响  
　　软件的可伸缩性：是指软件在不进行修改的情况下适应不同的工作环境的能力。由于硬件的飞速发展和软件开发周期较长的矛盾，软件升级的需要显得非常迫切。如果软件的升级和移植非常困难，软件的生命期必定很短，使得化费巨大人力物力开发出的软件系统只能在低性能的硬件或网络上运行，甚至被废弃不用，造成巨大的浪费。   
　　 软件的可维护性：软件的维护也是必然的事情，为了保证软件的较长使用寿命，软件就必须适应不断的业务需求变化，根据业务需求的变化对软件进行修改。修改的成本和周期都直接和软件的体系结构相关。一个好的软件体系结构可以尽可能地将系统的变化放在系统的配置上，即软件代码无需修改，仅仅是在系统提供的配置文件中进行适当的修改，然后软件重新加载进入运行状态，就完成了系统部分功能和性能要求的变化。对于重大改动，需要打开源代码进行修改的，也仅仅是先继承原先的代码，然后用新的功能接替原先的调用接口，这样将把软件改动量减小到最低。  
　　软件易用性：软件的易用性是影响软件是否被用户接受的关键之关键因素。在软件产品中，设计复杂，功能强大而完备，但因为操作繁复而被搁置者屡见不鲜。造成的主要原因在于缺乏软件开发中软件体系结构的宏观把握能力。另一方面，缺乏有效的手段进行软件需求的确定和对潜在需求的挖掘。

# 二、项目管理的风险

1、软件项目管理的风险

软件项目管理的风险来自软件项目自身的特点：  
　　软件产品不可见：开发的进展以及软件的质量是否符合要求难于度量，从而使软件的管理难于把握。   
 软件的生产过程不存在绝对正确的过程形式：可以肯定的是不同的软件开发项目应当采用不同的或者说是有针对性的软件开发过程，而真正合适的软件开发过程是在软件项目的开发完成才能明了的。因此项目开发初只能根据项目的特点和开发经验进行选择，并在开发过程中不断的调整。   
 大型软件项目往往是"一次性"的。以往的经验可以被借鉴的地方不多。回避和控制软件管理风险的唯一办法就是设立监督制度，项目开发中任何较大的决定都必须有主要技术环节甚至是由用户参与进行的。在该项目中项目监督由项目开发中的质量监督组来实施。  
2、软件开发的人员和其责任分析

项目组长1人  
 主要职责：进行全局把握，对组员的任务进行分工，确保工作能在限定期限内完成。协同项目负责人进行软件系统的分析和设计工作，书写软件需求分析和系统设计相关文档。在软件实现阶段进行测试策略的编制和对性能测试的指导。

项目负责人1人  
 主要职责：制定项目开发计划和开发策略，参与项目核心系统的分析设计，同时努力保证开发计划的按时完成和开发策略的真正贯彻落实。  
 校对测试组2人  
 主要职责：编制软件质量控制计划，并负责落实；控制必要文档的生产，通过文档，监督项目实施过程中软件的质量，并产生软件质量报告，提请项目经理和项目负责人审阅；对于项目中出现的质量问题，主持召开质量复审会议。已经实现的软件组件、构件或系统进行正确性验证测试，整合后的系统的性能测试等。编写测试报告。

程序员3人  
 主要职责：协助分析人员进行详细设计，和软件系统的代码实现，并进行适当的白盒测试。  
 文档组2人  
 主要职责：对各部门产生的文档进行格式规范、版本编号和控制、存档文件的检索；进行软件质量监督。通过适当的人员配备和职责划分，能有效的降低软件开发在后期的失控的可能性，和软件对关键人员的依赖性。

# 三、软件技术风险

　　回避和控制这部分风险的办法是在项目进行的过程不断的对该阶段进行风险估计和指定有效的里程碑。同时采用"范例"方式提高开发人员的构件组件的分析识别能力，适时调整构件组件的数量和粒度。

# 四、软件过程风险

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《需求分析》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、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。  
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。  
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支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。