**软 件 描 述 文 档**

**产品名称**

**公司名称**

**软件基本信息**

**产品名称**

**公司名称**

**1、产品标识：**

×××软件标识：

软件名称： ×××

软件型号及版本号：×××

制造商： ×××公司

生产地址： ×××

**2、安全性级别**

××是一种××软件，所以随之而来的软件安全性问题也极为重要。

(a)××软件是一种抽象的逻辑产品，其存在形式是虚拟和动态的……..

(b)软件质量的测度十分困难，其质量的控制重点在软件的需求分析和设计阶段，开发过程中产生错误的难以追踪；……；

(c)硬件有老化现象，失效曲线似浴盆，硬件的维护可通过纠错、修复或更换失效的系统重新恢复功能。而软件的维护复杂，只有通过修改代码来排错。同时软件可能在使用中随着缺陷的发现和消除，而使性能提高。软件的修改看似比硬件容易，却比硬件更难于控制。看上去无关紧要的软件代码修改会在软件的其他地方引起无法预测的、十分关键的问题；

(d)软件的失效防护困难。对硬件可采用预防性维护技术预防故障，采用断开失效部件的办法诊断故障，而软件则不能采用这些技术；但软件的失效会毫无征兆的出现，会因执行一条未经验证的路径而出现故障；而同一软件的冗余不能提高可靠性。

(e)软件的失效是系统性失效，其失效的条件有时比较复杂。因此，可能会无法清晰地洞察其原因，而误归结其为系统中硬件的随机失效。导致无法及时排除软件中的故障，造成隐患的长期存在。

以上论述了××软件的复杂性，以及出现问题无法预测性和软件的实效防护困难。××软件一旦出现问题则很可能导致患者×××或者对患者造成严重的伤害，例如，×××软件一旦在运行过程中失效，机器停止工作则很可能导致患者由于××而变为×××，所以××软件安全性级别为××级。

**3、功能结构**

上位机程序功能描述：

下位机：

a）功能模块网络：

c）下位机程序框图

1. 上位机

a）功能模块网络：

b）上位机程序框图：

**4、硬件关系**

**5、运行环境**

**5.1硬件配置：**

**5.2软件运行环境**

**6、适应范围**

**6.1软件组件整体的功能用途**

**6.2医疗器械的适用范围**

**7、禁忌症**

**8、上市历史**

**软件实现过程**

**产品名称**

**公司名称**

1. **开发综述**

**1.1 嵌入式开发平台**

**1.1.1 单片机的开发平台**

●

**1.2分析测控系统**

在进行单片机应用系统开发时，首先要对该测控系统进行可行性分析以及系统总统方案设计。

1.2.1．可行性分析

可行性分析主要是分析整个设计任务的可能性。

1.2.2．系统总体方案设计

当完成可行性分析后，便进入系统整体方案设计阶段。这里，主要结合国内外相关产品的技术参数和功能特性、本系统的应用要求以及现有条件，来决定本设计所要实现的功能和技术指标。接着，制定合理的计划，编写设计任务书，从而完成该单片机应用系统的总体方案设计。

**1.3器件选型**

**1.4硬件资源分配**

**1.5程序设计**

**1.6仿真测试**

**1.7实际硬件测试**

**2需求规格**

**2.1编写目的**

1.

2.

**2.2背景**

1.

2. 用户：医务专业人员

3. 实现：×××公司

4. 构建平台：

**2.3定义**

1.

2.

3.

4.

5.

6.

**2.4对功能的规定**

**1 功能划分**

1.

2.

3.

4.

5.

6.

7.

8.

**2 功能描述**

1.：

2.

短；

3.

4.

5.

6.

7.

8.

2.5对性能的规定

**2.5.1精度**

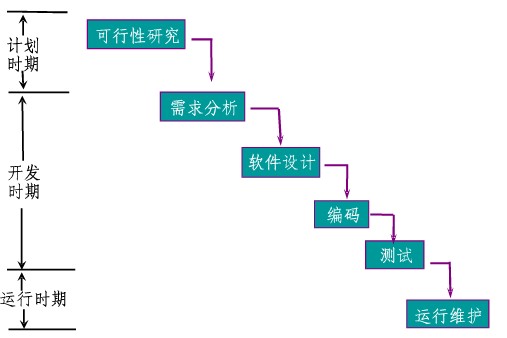
**2.5..2时间特性要求**

新一次。

**2.5..3输入输出要求**

**2.5..4警示信息**

**3.软件的生存周期**



1.确定软件开发的可行性与计划

此阶段是软件开发方与需求方共同讨论，主要确定软件的开发目标及其可行性。在软件开发的可行性研究与计划阶段内，要确定该软件的开发目标和总的要求，要进行可行性分析、投资一收益分析、制订开发计划。

    这个阶段我们需要编制项目开发计划。到了编制项目开发计划阶段，我们要明白编制项目开发计划的目的是用文件的形式，把对于在开发过程中各项工作的负责人员、开发进度、所需经费预算、所需软、硬件条件等问题作出的安排记载下来，以便根据本计划开展和检查本项目的开发工作。  
2.对所开发的软件需求进行分析

    在确定软件开发可行的情况下，对软件需要实现的各个功能进行详细分析。需求分析阶段是一个很重要的阶段，这一阶段做得好，将为整个软件开发项目的成功打下良好的基础。”唯一不变的是变化本身。”，同样需求也是在整个软件开发过程中不断变化和深人的，因此我们必须制定需求变更计划来应付这种变化，

以保护整个项目的顺利进行。

    这个阶段我们需要编写软件需求说明书和数据要求说明书。编制软件需求说明书是为了使用户和软件开发者双方对该软件的初始规定有一个共同的理解，使之成为整个开发工作的基础。内容包括对功能的规定对性能的规定等。

这个阶段，我们可以通过编制数据要求说明书，来向整个开发时期提供关于被处理数据的描述和数据采集要求的技术信息。

3.软件开发的设计阶段

    此阶段主要根据需求分析的结果，对整个软件系统进行设计，如系统框架设计，数据库设计等等。软件设计一般分为总体设计和详细设计。好的软件设计将为软件程序编写打下良好的基础。这个结算我们需要编写概要设计说明书、详细设计说明书、数据库设计说明书和测试计划初稿。

概要设计说明书:概要设计说明书又可称系统设计说明书，这里所说的系统是指程序系统。编制的目的是说明对程序系统的设计考虑，包括程序系统的基本处理流程、程序系统的组织结构、模块划分、功能分配、接口设计。

运行设计、[数据结构](http://www.csai.cn/incsearch/search.asp?key=%CA%FD%BE%DD%BD%E1%B9%B9" \t "_blank)设计和出错处理设计等，为程序的详细设计提供基础。

详细设计说明书:详细设计说明书又可称程序设计说明书。编制目的是说明一个软件系统各个层次中的每一个程序(每个模块或子程序)的设计考虑，如果一个软件系统比较简单，层次很少，本文件可以不单独编写，有关内容合并人概要设计说明书。

数据库设计说明书:数据库设计说明书的编制目的是对于设计中的数据库的所有标识、逻辑结构和物理结构作出具体的设计规定。测试计划初稿:这里所说的测试，主要是指整个程序系统的组装测试和确认测试。

本文件的编制是为了提供一个对该软件的测试计划，包括对每项测试活动的内容、进度安排、设计考虑、测试数据的整理方法及评价准则。

4.实现阶段

    此阶段是将软件设计的结果转换成计算机可运行的程序代码。在程序编码中必须要制定统一，符合[标准](http://www.csai.cn/incsearch/search.asp?key=%B1%EA%D7%BC" \t "_blank)的编写规范。以保证程序的可读性，易维护性，提高程序的运行效率。这个阶段我们需要编写模块开发卷宗和用户手册完工操作手册。

模块开发卷宗(开始编写):模块开发卷宗是在模块开发过程中逐步编写出来的，每完成一个模块或一组密切相关的模块的复审时编写一份，应该把所有的模块开发卷宗汇集在一起。编写的目的是记录和汇总低层次开发的进度和结果，

以便于对整个模块开发工作的管理和复审，并为将来的维护提供非常有用的技术信息。用户手册完工操作手册:操作手册的编制是为了向操作人员提供该软件每一个运行的具体过程和有关知识，包括操作方法的细节。

5.测试阶段

    在软件设计完成后要经过严密的测试，以发现软件在整个设计过程中存在的问题并加以纠正。整个测试过程分单元测试、组装测试以及系统测试三个阶段进行。测试的方法主要有白盒测试和黑盒测试两种。

在测试过程中需要建立详细的测试计划并严格按照测试计划进行测试，以减少测试的随意性。

这个阶段我们需要编写模块开发卷宗和项目开发总结报告。 模块开发卷宗(此阶段内必须完成)测试分析报告:测试分析报告的编写是为了把组装测试和确认测试的结果、发现及分析写成文件加以记载。

项目开发总结报告:项目开发总结报告的编制是为了总结本项目开发工作的经验，说明实际取得的开发结果以及对整个开发工作的各个方面的评价。

6.运行与维护阶段

    软件维护是软件生命周期中持续时间最长的阶段。在软件开发完成并投人使用后，由于多方面的原因，软件不能继续适应用户的要求。要延续软件的使用寿命，就必须对软件进行维护。软件的维护包括纠错性维护和改进性维护两个方面。

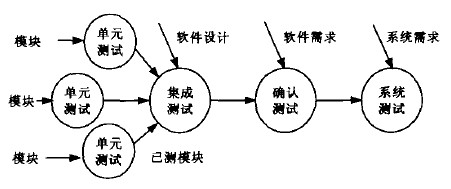
总之，失败的软件项目各有其失败，而成功的软件项目都一样:离不开规范的软件开发过程。想要设计出优秀的，适合实际需要的软件作品，科学规范的软件开发流程是必须遵守的。当然，软件开发工作也需要与时俱进，这一套软件开发六部曲也并不是永远适用的。

我们需要在平时的工作中多多总结，才能做到与时俱进，才能一直保持，实现科学的软件开发。

**4.软件的验证和确认**

软件测试工作量往往占软件开发总工作量的40%以上。软件测试之所以在软件生命周期占有如此重要地位，是因为它贯穿了软件定义与开发的整个生命周期，是软件质量保证的重要手段。需求分析、概要设计、详细设计以及源程序，都应成为软件测试的对象。与开发过程类似，测试过程也需要分步骤进行，后一个步骤在逻辑上是前一个步骤的继续。在众多测试中，确认测试检查软件能否按合同要求进行工作，即是否满足软件需求说明书中的确认标准，是软件发布之前不可或缺的重要测试之一。

软件测试阶段的划分模型



单元测试，是在程序编码阶段对各单元模块进行单独的测试，旨在及时发现并纠正程序单元中暗藏的缺陷。集成测试主要是考察模块的集成，包括这些模块组成的 (子)系 统的功能、性能及其外部接口等质量特性。确认测试，是根据软件需求规格说明中对功能性需求和非功能性需求的描述,考察软件各项特征是否符合要求。系统测试，则是测试由软件和硬件集成的完整系统，以检查其 是否符合需求 。

4.1测试方法

我们采用了以下两种方法来测试软件：

(1)等价类划分：把程序的输入域划分成数据类，据此可以导出测试用例，一个理想的测试用例能独自发现一类错误。

(2)边值分析：确定程序处理的边界情况，设计使程序运行在边界情况附近的测试方案。诸如：下标、纯量、数据结构和循环等边界附近 。

4.2系统测试计划见附件

4.3用户测试计划见附件

**5.核心算法**

**1）上位机程序核心算法：**

▼数字滤波无需其他的硬件成本，只用一个计算过程，可靠性高，不存在阻抗匹配问题。尤其是数字滤波可以对频率很低的信号进行滤波，这是模拟滤波器做不到的。

▼数字滤波使用软件算法实现，多输入通道可共用一个滤波程序，降低系统开支。

▼只要适当改变滤波器的滤波程序或运算，就能方便地改变其滤波特性，这对于滤除低频干扰和随机信号会有较大的效果。

本程序是采用的是数字滤波算法中的算术平均滤波算法，该算法的基本原理很简单，就是连续取N次采样值后进行算术平均。

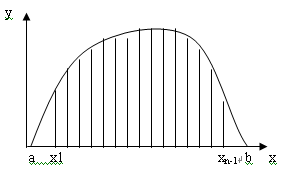
说明：算术平均滤波算法适用于对具有随机干扰的信号进行滤波。这种信号的特点是有一个平均值，信号在某一数值附近上下波动。信号的平均平滑程度完全到决于N值。当N较大时，平滑度高，灵敏度低；当N较小时，平滑度低，但灵敏度高。为了方便求平均值，N一般取4、8、16、32之类的2的整数幂，以便在程序中用移位操作来代替除法。

波形从左走到右，需要18s；纵坐标一共140点，20个点代表1kPa。选择20个点作为－1kPa，每次刷新波形时候，先把压力轴从上到下清除（即用屏幕底色填充），在把当前压力值换算到纵坐标上，从当前点画蓝色线一直到时间轴上。随着时间的推移，便产生了蓝色的填充的压力波形。当时间轴到了最大值时，回到最小值。压力轴范围为-1到6kPa，压力数值超出了上下限范围后，就强制把波形置为-1或6kPa。

流速波形也是采用的数字滤波中的算术平均算法，波形纵坐标代表流速值，横坐标代表时间。时间轴每50ms更新一次，一共360点，波形从左走到右，需要18s；纵坐标一共140点，分为上下半轴，各占70个点，上半轴为吸气波形，下半轴为呼气波形。每次刷新波形时候，先把流速轴从上到下清除（即用屏幕底色填充），每隔50ms采集一次流速值，即50ms之内累加的脉冲数换算为流速值，在根据当前是吸气还是呼气状态，把流速值换算到纵坐标上，从当前点画浅蓝色线一直到时间轴上。随着时间的推移，便产生了浅蓝色的填充的流速波形。当时间轴到了最大值时，回到最小值。流速轴范围为0-70，流速数值超出了上下限范围后，就强制把波形置为70。

**2）下位机程序核心算法**

下位机潮气量的算法：由于打气的气流不是一个恒定的流速，所以采用了积分的算法，即气流流速的积分等于潮气量。



上图曲线为潮气量流速波形，x代表时间，y代表流速。潮气量则为阴影部分的面积，采用积分算法在【a，b】内插入若干个分点

a=x0<x1<x2<x3<…<xn-1<xn =b

1.把【a，b】分成n个小区间【xi-1 ,xi】长度为△xi =xi -xi-1；

2.近似替代：在每个【xi-1 ,xi】上任取一点ξi，以【xi-1 ,xi】为底，f（ξi）为速度的小矩形面积为A= f（ξi）△xi

3.求和：面积的近似值为A≈∑ni=1 f（ξi）△xi

4.取极限：当分割无线加细，即小区间的最大长度=max{△x1，△x2，△x3，…△xn} 0时，有小矩形面积和∑ni=1 f（ξi）△xi A，即可达到积分计算公式也就是潮气量计算公式

A=lim∑ni=1 f（ξi）△xi

λ→0

**软 件**

**风 险 管 理 报 告**

**产品名称**

**×××公司**

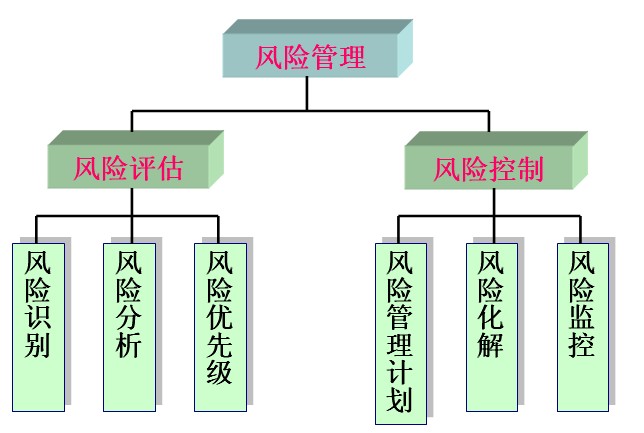
**1.概要：**

**软件风险概述：**

* + 使软件项目的实施受到影响和损失、甚至导致失败的、可能会发生的事件
  + 例如，人员的临时流失，计划过于乐观，设计的低劣

**软件风险的特点**

* + 事先难以确定
  + 带来损失，影响项目实施，甚至会导致项目失败



**风险管理的组成**

**风险评估**

* + 风险识别：识别风险，形成风险列表
  + 风险分析：判定每一个风险出现的概率、产生的影响及其重要性
  + 风险优先级：按照每个风险的重要性排出一个风险优先级

**风险评估是风险控制的基础**

**风险控制**

* + 风险管理计划：针对各个重要风险制定风险管理计划，确保各个单独的风险管理计划之间以及它们与相互计划之间的一致性
  + 风险化解：执行风险管理计划，以缓解或消除风险
  + 风险监控：监控风险化解的过程，可能会识别出新的风险

**2.风险评估**

2.1风险识别

* 风险的类别
  + 计划编制
  + 组织和管理
  + 开发环境
  + 最终用户
  + 客户
  + 需求
  + 产品外部环境
  + 人员
  + 设计和实现
  + 过程

2.1.1计划编制风险

* 计划、资源和产品的定义完全由客户或上层领导决定，忽略了项目组的意见，并且这些决定不完全一致
* 计划忽略了必要的任务和活动
* 计划不切实际
* 计划基于特定小组成员，而这样的小组成员根本得不到
* 产品规模估算过于乐观
* 工作量估算过于乐观
* 进度的压力造成生产率的下降
* 目标日期提前，但没有相应地调整产品范围和可用资源
* 一个关键任务的延迟导致其他相关任务的连锁反应

2.1.2组织和管理风险

* 缺乏强有力、有凝聚力的领导(项目组、企业)
* 解雇员工导致项目小组能力下降
* 削减预算打乱项目计划
* 仅由管理层和市场人员进行技术决策，导致进度延长
* 低效的项目组组织结构降低生产率
* 管理层审查/决策的周期比预期时间长
* 管理层作出了打击项目组积极性的决定
* 非技术的第三方的工作比预期要长(如, 采购硬件设备)
* 计划性太差，无法适应期望的开发速度
* 项目计划由于压力而放弃，导致开发混乱
* 管理方面的英雄主义，忽视客观确切的状态报告，降低发现和改正问题的能力

2.1.3开发环境风险

* 设施不能及时到位
* 设施到位，但不配套
* 开发工具未能及时到位
* 开发工具不如期望的那样有效，开发人员需要更多的时间，或者更换工具
* 开发工具的学习期比预期的要长
* 开发工具的选择不是基于技术需求，不能提供计划要求的功能

2.1.4最终用户风险

* 最终用户坚持新的需求
* 最终用户对最后交付的产品不满意，要求重新设计和重做
* 最终用户不买进项目产品，无法提供后续支持
* 最终用户的意见未被采纳，造成产品最终无法满足用户要求

2.1.5客户风险

* 客户坚持新的需求
* 客户对规划、原型和规格的审核/决策超出预期
* 客户没有参与规划、原型和规格的审核，导致需求不稳定，以及长时间的变更
* 客户答复的时间比预期的要长
* 客户坚持技术决策而导致计划延长
* 客户对开发进度管理过细，导致实际进度变慢
* 客户提供的组件无法与开发的产品匹配，导致需要额外的设计和集成工作
* 客户提供的组件质量欠佳，导致额外的测试、设计或者功能不完善
* 客户要求的支持工具与环境不兼容，性能差或者不完善，导致生产率降低
* 客户不接受交付的软件，尽管它满足了所有的规格
* 客户期望的开发速度是开发人员所无法达到的

2.1.6需求风险

* 需求已经成为项目基准，但仍在变化
* 需求定义欠佳：不清晰、不准确、不一致
* 增加额外的需求

2.1.7产品风险

* 错误发生率高的模块，需要更多的时间对它进行测试、设计和实现
* 矫正质量低下的不可接受的产品需要更多的时间对它进行测试、设计和实现
* 由于功能错误，导致需要重新进行设计和实现
* 开发额外不需要的功能延长了进度
* 要满足产品规模和速度要求，需要更多的时间
* 严格要求与现有系统兼容，需要更多的时间
* 要求软件重用，需要更多的时间

2.1.8外边环境风险

* 产品依赖政府规章，而规章的改变不可预期
* 产品依赖草拟中的技术标准，而最后的标准不可预期

2.1.9人员风险

* 招聘人员所需的时间比预期要长
* 作为人员参与工作的先决条件(如培训、其他项目的完成等)不能按时完成
* 开发人员与管理层关系不佳导致决策迟缓、影响全局
* 项目组成员没有全身心地投入到项目中，因而无法达到所需的产品功能和性能需求
* 缺乏激励措施、士气低下，降低生产能力
* 缺乏必要的规范，增加工作失误，重复工作，降低工作质量
* 缺乏工作基础(语言、经验、工具等)
* 项目结束前，项目组成员离开项目组
* 项目后期，加入新的开发人员，额外的培训和沟通降低了项目组成员的开发效率
* 项目组成员不能有效的在一起工作
* 由于项目组成员之间的冲突，导致沟通不畅，设计欠佳，接口错误和额外重复的工作
* 有问题的项目组成员没有调离项目组，影响其他成员的积极性
* 项目组的最佳人选没有加入项目组，或者加入项目组但没有合理使用
* 关键任务只能兼职参与
* 项目人员不足
* 任务的分配和人员的技能不匹配
* 人员工作的进展比预期的要慢
* 项目管理人员怠工导致计划和进度失效
* 技术人员怠工导致工作遗漏、质量低下，工作需要重做

2.1.10设计和实现风险

* 设计过于简单，考虑不仔细、不全面，导致重新设计和实现
* 设计过于复杂，导致一些不必要的工作，影响效率
* 设计质量低下，导致重新设计和实现
* 使用不熟悉的方法，导致需要额外的培训时间
* 产品使用低级语言编写，导致效率较低
* 分别开发的模块无法有效集成，需要重新设计和实现

2.1.11过程风险

* 跟踪不准确，导致无法预知项目进展是否落后于计划
* 前期的质量保证行为不真实，导致后期的重复工作
* 质量跟踪不准确，导致无法得知影响进度的质量问题
* 不能有效遵循标准，导致沟通不足，质量问题和重复工作
* 风险管理粗心，导致没有发现重大的项目风险

**3.风险列表**

|  |  |
| --- | --- |
| 编号 | 风险名称 |
| 1 | 计划过于乐观 |
| 2 | 由于要完全支持自动从主机更新数据而照成额外的需求 |
| 3 | 由于市场变化而需额外的需求 |
| 4 | 图形格式子系统接口不稳定 |
| 5 | 设计欠佳，需要重新设计 |

**4.风险分析**

4.1评估风险发生的概率

* 主观性较强，采用方法
  + 熟悉系统、有经验的人参与评估
  + 多人独立评估，综合折中
  + 采用分类：非常可能(0.8-1.0), 很可能(0.6-0.8)，或许(0.4－0.6)，不太可能(0.2-0.4)，不可能(0-0.2)

|  |  |  |
| --- | --- | --- |
| 编号 | 风险名称 | 概率 |
| 1 | 计划过于乐观 | 50% |
| 2 | 由于要完全支持自动从主机更新数据而照成额外的需求 | 5% |
| 3 | 由于市场变化而需额外的需求 | 35% |
| 4 | 图形格式子系统接口不稳定 | 25% |
| 5 | 设计欠佳，需要重新设计 | 15% |

4.2评估风险发生造成的损失

* 可以基于“进度”，“成本”或者“工作量”来进行估算

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 风险名称 | 概率 | 损失（人周） |
| 1 | 计划过于乐观 | 50% | 5 |
| 2 | 由于要完全支持自动从主机更新数据而照成额外的需求 | 5% | 20 |
| 3 | 由于市场变化而需额外的需求 | 35% | 8 |
| 4 | 图形格式子系统接口不稳定 | 25% | 4 |
| 5 | 设计欠佳，需要重新设计 | 15% | 15 |

4.3计算风险危险度

风险危险度 = 风险概率 × 风险损失

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 风险名称 | 概率 | 损失（人周） | 危险度 |
| 1 | 计划过于乐观 | 50% | 5 | 2.5 |
| 2 | 由于要完全支持自动从主机更新数据而照成额外的需求 | 5% | 20 | 1.0 |
| 3 | 由于市场变化而需额外的需求 | 35% | 8 | 2.8 |
| 4 | 图形格式子系统接口不稳定 | 25% | 4 | 1.0 |
| 5 | 设计欠佳，需要重新设计 | 15% | 15 | 2.25 |

4.4风险优先级

* 统计表明，项目80%成本用于解决20%的问题
* 风险管理重点关注20％重要的部分
* 根据风险的危险度确定风险的重要性，忽略其他的部分

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 风险名称 | 概率 | 损失（人周） | 危险度 |
| 3 | 由于市场变化而需额外的需求 | 35% | 8 | 2.8 |
| 1 | 计划过于乐观 | 50% | 5 | 2.5 |
| 5 | 设计欠佳，需要重新设计 | 15% | 15 | 2.25 |
| 2 | 由于要完全支持自动从主机更新数据而照成额外的需求 | 5% | 20 | 1.0 |
| 4 | 图形格式子系统接口不稳定 | 25% | 4 | 1.0 |

**5.风险监控**

* 检查风险的化解程度及其变化(概率、损失)
* 风险监控的方式
  + 监控和跟踪重要的(前10个)风险，记录风险危险度的变化以及风险化解的进展
  + 中间审查，在每个里程碑后进行小规模的走查
  + 任命风险官员(适合于大项目)，警告项目风险，防止项目经理和开发人员忽略计划中的风险管理

**软 件 集 成 测 试**

**计 划**

**产品名称**

**××××公司**

**1． 目的**

《软件集成测试计划》用于明确软件产品确认测试过程中测试设计、测试执行及测试总结工作的具体任务分解、人员安排、进度及输出结果。以使整个测试工作有计划地顺利进行。本文规定了测试的编写格式及要求。

**2适用范围**

本规范适用于软件项目与软件产品的集成测试活动。

**3职责**

* 1. 项目负责人：负责集成测试计划的审批。
  2. 开发人员： 负责编制《集成测试计划》。

**4．集成测试计划规范**

4.1集成测试一般集中于各个模块接口间的测试。

集成测试界于单元测试和系统测试之间。集成测试内容必须源于《系统设计报告》。集成测试着重于集成版本的外部接口的行为。因此，测试需求须具有可观测、可测评性。

4.2 集成测试计划制定的时间

在《系统设计报告》完成之后，依据《系统设计报告》制定集成测试计划。

|  |  |  |  |
| --- | --- | --- | --- |
| 类别 | 递交时间 | 制定者 | 审批者 |
| 《集成测试计划》 | 在系统设计阶段就可以起草，最迟可在实现阶段之初递交。 | 开发人员 | 项目负责人 |

4.3 集成测试计划内容：

1）明确测试所需要的资源（人员、人员的任务分派、各小组的联系人、时间进度、team里面的各种server、Lab环境等等），各种支持资源由谁负责维护等。

2）明确测试策略。就是什么测，什么不测。测的东西，测到什么程度，是不是用自动化，等等。若有本次测试不涉及的地方，需在测试计划里注明：哪些内容不测、不测的原因是什么。

3）必须列明产品通过的准则。

4）指明测试中可能遇到的问题与对策

4.4 集成测试计划制定后，需经过项目相关人员的评审。

**系 统 测 试**

**计 划**

**产品名称**

**××××公司**

目 录

**[1 目的 1](#_Toc162946843)**

**[2 适用范围 1](#_Toc162946844)**

**[3 图例说明 1](#_Toc162946845)**

**[4 规程文档结构 1](#_Toc162946846)**

**[5 规程 2](#_Toc162946847)**

[5.1 进入准则 2](#_Toc162946848)

[5.2 测试策划 2](#_Toc162946849)

*[5.2.1](#_Toc162946850)**[流程图 2](#_Toc162946850)*

*[5.2.2](#_Toc162946851)**[活动描述 3](#_Toc162946851)*

[5.3 测试设计 3](#_Toc162946852)

*[5.3.1](#_Toc162946853)**[流程图 3](#_Toc162946853)*

*[5.3.2](#_Toc162946854)**[活动描述 4](#_Toc162946854)*

[5.4 测试执行 4](#_Toc162946855)

*[5.4.1](#_Toc162946856)**[流程图 4](#_Toc162946856)*

*[5.4.2](#_Toc162946857)**[活动描述 5](#_Toc162946857)*

[5.5 缺陷管理 6](#_Toc162946858)

*[5.5.1](#_Toc162946859)**[流程图 6](#_Toc162946859)*

*[5.5.2](#_Toc162946860)**[活动描述 7](#_Toc162946860)*

*[5.5.3](#_Toc162946861)**[缺陷等级 8](#_Toc162946861)*

*[5.5.4](#_Toc162946862)**[缺陷状态 8](#_Toc162946862)*

[5.6 变更管理 9](#_Toc162946863)

[5.7 退出准则 9](#_Toc162946864)

[5.8 规程接口 9](#_Toc162946865)

**[6 文件清单 9](#_Toc162946866)**

**[7 图表索引 10](#_Toc162946867)**

**目的**

使软件系统测试活动是规范和有计划的

及时对软件产品进行系统测试，验证系统是否满足需求规格的定义，找出与需求规格不相符合或与之矛盾的地方

测试中发现的缺陷保证被记录、跟踪直到关闭；对项目的缺陷数据进行量化的分析

**适用范围**

本规程适用于以下人员

项目经理

软件测试组

软件项目组

配置管理工程师

**图例说明**

在本文中用到了以下图例，特此说明：

 过程：用于描述要执行的活动和过程；

 判定点：用于描述流程中是否满足某个条件后的分流点；

 接口：用于描述引用的接口规程；

 文档：用于描述输入、输出的文档；

 开始节点：用于表示流程的入口；

 结束节点：用于表示流程的结束；

**规程文档结构**



图表1：规程文件结构

文档结构图说明：

1．《系统测试计划》是根据项目的规模和重要性来确定，是可裁减的。一般来说，《系统测试进度计划》是必须的。但对于公司认定的小项目，可以没有《系统测试进度计划》，具体要求按照公司的相关规定《小项目软件过程管理指南》执行；

2．《系统测试用例》在测试管理工具TestDirector中进行编写和管理，可以根据需要导出到Word；

3．项目组可根据系统的实际需要，通过提交功能或兼容性或性能测试任务单来启动相应的测试任务，后二者的测试一般都需要在完成功能测试的基础上才可执行；

4．《缺陷记录跟踪表》是指从TestDirector里导出缺陷记录到Excel时的文件格式，测试中发现的缺陷都将使用TestDirector来进行记录、跟踪和管理；

**规程**

**进入准则**

项目启动，并由项目经理与部门（高级）经理、测试组组长商量，确定此项目需要安排独立的测试人员加入。

**测试策划**

**流程图**



图表2：测试策划流程

**活动描述**

《系统测试计划》或《系统测试进度计划》都需要项目组提供相关的项目文档作为主要编写依据；

《系统测试计划》是可裁剪的，《系统测试进度计划》是必须的（小项目除外）；不管有没有编写《系统测试计划》，都需要对《系统测试进度计划》进行评审，以确保时间上能与开发进度相一致，并且保证整个软件测试活动是有序的；

《系统测试计划》的评审需要软件工程组各类成员的代表共同参与（限于篇幅在图中没有表达出来，特在此说明），评审的具体形式可根据项目开发计划中的要求进行；对《系统测试进度计划》的评审形式一般为走查；

**测试设计**

**流程图**



图表3：测试设计流程

**活动描述**

《系统测试用例》的评审需要软件工程组各类成员的代表共同参与，特别是熟悉需求和设计的代表必须参加，评审的具体形式可根据项目开发计划中的要求进行；

编写《系统测试用例》的时候，需要遵循《系统测试用例编写规范》上的具体要求。应以《系统需求规格说明书》作为主要的依据，设计文档、实际可运行的程序雏形都可作为编写的辅助材料。在编写过程中，应多与分析、设计人员沟通，不明确的地方以项目经理的答案作为衡量标准；

根据公司的实际情况，测试用例的编写和评审都将采取迭代的方式进行，逐步完善；在整个设计过程中，可以阶段性的部分评审，但在正式测试前应至少完成一次整体性的评审；

《系统测试用例》评审时，测试人员应讲解测试用例的编写思路，主要的关键点，并提出不能确定的问题，由熟悉需求和设计的人员进行确认和补充，以达到确保《系统测试用例》有效覆盖系统需求的目的；

测试组内部走查的时候，需要使用《系统测试用例检查表》。

**测试执行**

**流程图**



图表4：测试执行流程

**活动描述**

项目组在单元测试完成后，可提交系统测试的任务。项目经理应在演示前提交《系统功能测试任务单》，任务单上应列出该版本修改了哪些特征项，以及指明测试的要点和时间要求；

测试评估是指，由项目经理或指定人员把《系统功能测试任务单》上的本次测试的范围在测试系统中进行功能演示，在演示的同时进行业务重点的讲解。在对系统进行第一次演示时，项目组需要结合需求说明的PPT文档，先对测试人员介绍系统需求以及业务逻辑关系，然后再开始系统的操作演示。双方对系统的可测试性达成一致后，即可开始测试；如果双方对可测试性无法达成一致时，则应及时向高级经理反映，并相互协商解决问题；

系统提交测试时，一般都需要演示，如果项目组认为不需要演示，则必须经过测试组组长的同意

演示任务必须在测试环境中执行，项目组需要将程序发布到测试组的服务器上，在一次测试当中程序不允许更新，具体可参见《系统测试环境发布流程》。

测试组接收测试任务前系统演示要达到的要求：1）演示时没有任何明显的错误；2）上次测试发现的缺陷都已修改。

项目组给测试组系统测试计划预留时间至少为2天。

一轮测试结束后，测试人员应及时将测试结果及主要情况、缺陷统计反馈填写在《系统功能测试任务单》上，并提交给项目经理审核。

在一个项目中进行多次测试时，每次测试的结果都反馈在《系统功能测试任务单》上，在整个项目的测试工作完成后，应编写《系统测试总结报告》；在项目开里程碑会议的时候，需要提交阶段性的《系统测试总结报告》；

测试组组长应对各个项目的《系统测试总结报告》进行审核，项目组也应对各类测试报告进行走查；

测试人员应严格按照测试进度计划的安排，进行测试工作，如果根据项目的实际情况，时间上有差异的，则应尽力配合项目组方面的时间安排，有需要时应主动采取加班等方式来保证按时按质地完成测试任务，必要时可向测试组组长申请加派人手增援；

测试人员应严格按照测试用例执行测试，并在测试中不断对测试用例进行完善和补充；

项目组可根据系统的实际需要，向测试组提出功能测试以外的测试请求，比如：兼容性测试和性能测试，需要提交相应的《兼容性测试任务单》或《性能测试任务单》，这两种类型的测试不需要演示，只要具备测试条件（一般在完成功能测试的前提下）即可进行。

执行性能测试前需要编写《性能测试计划》，跟项目经理或主要开发人员确定性能测试的范围和各项性能预期指标，编写性能测试脚本，组织和执行性能测试，并作好观察和记录。性能测试一般需要2天的准备时间，所以项目组有性能测试需求时，可尽量提前一些通知测试组，以方便做准备工作。

**缺陷管理**

**流程图**



图表5：缺陷管理流程

**活动描述**

在TD的缺陷库中新增一个缺陷之前，应先检查有没有类似的缺陷已经存在，避免重复。缺陷填写的详细要求请参照《缺陷填写规范》；

在任何情况下，“拒绝”、“暂缓”和“冻结”缺陷的时候，都需要在TD的备注中填写“拒绝”、“暂缓”或“冻结”的原因；并与测试人员当面沟通，避免因为理解上的偏差或系统版本的偏差等原因而随意地放过一个缺陷；暂缓缺陷和冻结缺陷，需要经过评审，由产品经理对缺陷定位和解决.在缺陷管理中只有产品经理有权限对缺陷进行“冻结”。

对”拒绝”和”暂缓”的缺陷有争议，测试组可以在系统演示时向项目组提出。对于每条暂缓的缺陷，项目组与测试组协商确定是否需要再测试此类缺陷。

原来暂缓和拒绝的缺陷，如果缺陷不存在了，由测试组关闭。测试组每两个月进行批量关闭。

在项目组中只有项目经理具有“Rejected”缺陷的权限；对每一条缺陷需要填写“原因分析”；

当测试人员对项目经理“拒绝”的缺陷有争议并无法达成一致的时候，可向评审委员会提出申请。评审委员会，是指由高级经理、项目经理、开发、测试以及其他相关的专家组成的评审小组，对有争议的缺陷进行最终的确认。（评审委员会的成员并不是固定的，而是根据实际情况的需要组成）；

从上面的流程图可以看出，对缺陷的管理实际上就是通过TestDirector来对缺陷的状态进行变化；也就是说缺陷从产生到关闭都是在TD里进行的，如果需要将缺陷记录形成文件，可以通过TD来导出到Excel或其他格式的文件；

**缺陷等级**

|  |  |  |  |
| --- | --- | --- | --- |
| 严重程度  名称 | 缺陷等级 | 英文名称 | 描 述 |
| 建议 | 4 | Suggestion | 测试人员对系统的一些建议，不影响系统的使用功能，开发人员可根据实际情况选择是否修改。 |
| 高 | 3 | High | 功能不能使用或在使用中出现的问题影响了系统的稳定性、造成数据存储错误或将错误数据带入下一环节、一些重要特性或性能不能达到指定的要求等。 |
| 中等 | 2 | Medium | 功能可以使用、在出错后做出一定处理，操作能够继续进行或功能实现有误，但问题的出现应不影响本功能或其他功能的实质性使用。 |
| 低 | 1 | Low | 用户界面显示、对齐、文字错误等。 |

具体的缺陷等级示例可以参考“缺陷填写规范”。

图表6：缺陷等级

**缺陷状态**

|  |  |  |
| --- | --- | --- |
| 状态名称 | 英文名称 | 描 述 |
| 新发现 | New | 是指当测试工程师在执行测试时新发现一个问题的时候的状态 |
| 打开 | Open | 是指当项目经理把新发现的问题分配给某位开发工程师以后的状态 |
| 已修改 | Fixed | 是指开发工程师完成被分配问题的修改后的状态 |
| 被拒绝 | Rejected | 是指项目经理在评审新发现的问题时认为该问题与其他问题重复或者不是一个缺陷的时候，才可以标识为该状态，并需要说明理由。只要是缺陷都不应被标识为拒绝。 |
| 重新打开 | Reopen | 是指测试工程师对已修改的问题进行验证时发现该问题仍然存在，则将此问题标识为该状态 |
| 已关闭 | Closed | 是指测试工程是对已修改的问题进行验证以后，认为该问题已经修正。 |
| 暂缓 | Respite | 是指项目经理认为需要修改，但因为特殊原因暂缓修改的。该类缺陷需经过评审,，产品经理同意暂缓的缺陷，需写明原因及修改时间。未修改的暂缓缺陷在项目收尾阶段分析、整理后提交到维护组。 |
| 冻结 | frozen | 是指项目组因技术或其它原因而没法修改并确定不修改的缺陷，该类严缺陷需经过评审，同意冻结的，由产品经理标识为该状态。 |

图表7：缺陷状态

**变更管理**

在项目的开发过程中，当有影响到测试的变更（如需求变更、开发进度变更）发生的时候，项目经理应及时通知相关的测试人员，测试人员接到变更通知后，应对该次变更情况作出分析，然后根据分析结果，对《测试进度计划》或《系统测试用例》等测试资源进行修改或补充（即将变更部分重新走测试策划、设计、执行的流程），并采取走查方式跟相关的开发人员进行确认；

项目上线后一个月集中以会议形式给测试组演示需求/功能上的变更，在演示后，测试人员根据演示情况更新测试用例。

**退出准则**

系统测试用例通过了评审，测试用例达到了有效覆盖系统所有可测试的需求（功能/特征项）；

按照系统测试进度计划完成了系统测试；

测试结果，系统满足需求规格说明书的要求；

在系统测试中发现的缺陷都已经得到合理地处理（理论上只要不是“Rejected”和“frozen”的缺陷都应该被“Closed”）

**图表索引**

[图表1：规程文件结构 2](#_Toc162958785)

[图表2：测试策划流程 3](#_Toc162958786)

[图表3：测试设计流程 4](#_Toc162958787)

[图表4：测试执行流程 5](#_Toc162958788)

[图表5：缺陷管理流程 7](#_Toc162958789)

[图表6：缺陷等级 8](#_Toc162958790)

[图表7：缺陷状态 9](#_Toc162958791)

**用 户 软 件 测 试**

**计 划**

**产品名称**

**××××公司**

目录

1. 目的
2. 适用范围
3. 职责

4. 工作程序

4.1 用户测试前的准备

4.2 测试实施

4.3 问题处理

**1. 目的**

规范协助客户确认软件或系统集成项目已达到合同规定的功能和质量要求的程序。

1. **适用范围**

适用于向用户交付的软件和/或系统集成项目。

1. **职责**

3.1 管理部门：根据客户要求进行软件和/或系统集成项目测试前的准备，协助客户完成测试。

3.2 开发部门：根据客户要求进行软件产品或软件项目测试前的准备，协助客户完成测试。

**4. 工作程序**

4.1 测试前的准备

在产品测试之前，管理部门开发部门根据合同中的测试准则检查所有软件项及其配置是否完整，做好测试准备。

4.2 测试实施.

4.2.1 根据合同的要求或双方协商的结果，确定测试时间，进度安排、测试准则、软/硬件环境和资源，以及双方的具体职责。测试工作应由客户负责主持，管理部门或开发部门应协助客户以便顺利测试，如需测试，本公司的测试人员还应协助客户制定“测试计划”。

4.2.2 测试测试

根据合同规定或客户要求，测试可以是现场测试或在客户认可的环境下进行。测试测试尽可能由客户实施，开发部门给予协助。

4.2.3 测试过程以测试准则为依据，所有与测试准则不一致的地方，均认为存在问题，并由双方确认问题的类型，记入“测试报告”。

4.3 问题处理

在测试测试过程中发现的问题根据合同规定来处理。如果合同中没有规定，应指明问题类型和责任归属，由开发部门与客户协商解决办法。

4.3.1 测试过程中存在的问题及解决办法写入“测试报告”。

4.3.2 “测试报告”由项目管理部门存档。

**软 件 系 统 测 试 报 告**

机器名称：

任务编号/序号： 工作名称：

程序（ID）： 程序名称：

编程员： 测试完成日期： 年 月 日

测试工程师： 测试完成日期： 年 月 日

1. 安装：

是 否

（1）程序运行环境已经正确设定 □ □

1. 程序代码检查：

（1）程序单位首部有程序说明和修改备注 □ □

（2）变量、过程、函数命令符合规则 □ □

（3）程序中有足够的说明信息 □ □

（4）修改注释符合要求 □ □

（5）类库的使用符合要求 □ □

1. 画面及格式检查：

（1）画面和格式符合规定需求 □ □

（2）程序命名符合格式需求 □ □

（3）画面和字段位置和宽度与设计文档一致 □ □

1. 功能测试：

（1）呼吸参数齐全正确 □ □

（2）功能键、触发键、按钮、菜单、选择项功能正确 □ □

（3）报警参数齐全正确 □ □

（4）呼吸模式符合要求 □ □

（5）设计文档规定的其它功能

测试内容：

□ □

□ □

□ □

□ □

□ □

* □
* □

1. 正确性测试：

（1）读/写/删除操作结果正确

（2）各种组合条件之查询正确

（3）设计文档规定的其它操作

测试内容： □ □

1. 可靠性测试：

（1）非法键容错测试

（2）异常字符容错测试

（3）程序负作用检查

（4）残留文件检查

1. 效率测试：
2. 画面效率测试：

延迟时间： □ □ □ □

1. 显示参数的刷新效率测试：

最小时间：□ □ □ □

最大时间：□ □ □ □

1. 其它测试：

测试内容： □ □

**测试备忘：**

**用 户 软 件 测 试 报 告**

机器名称：

任务编号/序号： 工作名称：

程序（ID）： 程序名称：

编程员： 测试完成日期： 年 月 日

测试用户： 测试完成日期： 年 月 日

1. 功能测试：

（1）呼吸参数齐全正确 □ □

（2）功能键、触发键、按钮、菜单、选择项功能正确 □ □

（3）报警参数齐全正确 □ □

（4）呼吸模式符合要求 □ □

（5）设计文档规定的其它功能

测试内容： □ □

1. 效率测试：
2. 画面效率测试：

延迟时间： □ □ □ □

1. 显示参数的刷新效率测试：

最小时间：□ □ □ □

最大时间：□ □ □ □

1. 其它测试：

测试内容： □ □

**测试备忘：**

