摘要：

本人于2018年1月参与了中石化XX油田XX采油厂“用电管理系统”的项目建设，该系统建设目标是实现分单位、分线路、分系统评价、优化、考核，全面提升采油厂用电管理水平。在该项目组中我担任系统架构师一职，主要负责系统整体架构设计。本文以该项目为例，先简单介绍几种目前比较主流的软件可靠性设计技术，然后讨论可靠性设计技术在项目中的具体应用。由于此系统的核心功能是对客户端请求消息的分阶段解析与处理，并要求在HTTP 报头分离、SOAP报文解析等过程中具有高可靠性。项目组结合软件可靠性设计与应用的原则，结合此前类似系统的设计实现经验，决定应用管道－过滤器架构风格和责任链设计模式，并使用RocketMQ作为消息中间件对高并发消息进行处理，对交互的数据进行强数据验证机制，软件系统中采用软件容错的N版本程序设计技术，对消息通讯、封层架构模块进行负载均衡设计。最终系统上线后，获得了用户的一致好评。

正文：

“用电管理系统”项目是采油厂能源管控中心系统的一个子系统。能源管控中心是中石化集团公司十三五规划中的“能效倍增”计划在胜利油田分公司的示范应用项目，该示范项目能够在实现企业节能目标管理、能源计量统计、节能潜力识别、能效分析优化的同时，有效支撑企业实施节能技术改造、促进企业用能水平不断提升。“用电管理系统”的建设目标是建立覆盖厂、区两级用电管理一体化体系，实现分单位、分线路、分系统评价、优化、考核，达到电网运行质量实时监控、异常情况精准管控、能耗总量全面受控，按照运行产量的方式运行电量，全面提升采油厂用电管理水平。该项目功能设计参考PDCA闭环管理的理念，共设计包括用电计划、用电分析、用电优化、用电考核、设备管理等五大功能模块。

我作为单位技术骨干之一，主持并参与了项目计划制定、需求分析、整体架构设计与技术选型、底层设计、部分编码等多项工作。下面，我将首先介绍几种目前比较主流的软件可靠性设计技术，然后详细介绍“用电管理系统”的分析和设计过程中所采用的可靠性设计技术及其原因。

一般来说，被认可的且具有应用前景的软件可靠性设计技术主要有容错设计、检错设计和降低复杂度设计等技术。其中常用的软件容错技术主要有恢复块设计、N版本程序设计和冗余设计三种方法，主要适用于软件失效后果特别严重的场合，恢复块设计就是选择一组操作作为容错设计单元，从而把普通的程序块变为恢复块。一个恢复块中包含有若干功能相同、设计差异的程序块，每一时刻有一个程序块处于运行状态，一旦某程序块出现故障，则用备份程序块予以替换。N版本程序设计的核心是通过设计出多个模块或不同版本，对于相同初始条件和相同输入的操作结果进行多数表决（防止因其中

某一软件模块／版本的故障而提供了错误的服务，以实现软件容错）。冗余设计的思路来源于硬件系统，但有所不同。软件冗余设计技术是采用多种不同路径、不同算法或不同实现方法的模块或系统作为备份，在出现故障时进行替换，维持系统的正常运行。；检错设计主要应用于无需在线容错的地方或不能采用冗余设计技术的部分；降低复杂度设计的思想就是在保证实现软件功能的苺础上，简化软件结构，缩短程序代码长度，优化软件数据流向，降低软件复杂度，从而提高软件可靠性。除此之外，还有故障树分析（FTA)、失效模式与效应分析（FMEA)等硬件可靠性技术也应用到了软件可靠性设计领域之中。

项目启动后，在架构设计工作的开始阶段，我们便意识到软件的可靠性设计对项目有着重要的影响。由于油田系统的复杂性与特殊性，对软件可靠性有着较高的要求，要提高软件产品的可靠性指标，首先要分析影响软件产品可靠性的原因。一般来说影响产品可靠性的原因有如下一些：

1、运行环境，软件可靠性定义是相对于运行环境而言的，一样的软件在不同的运行环境下其可靠性是不一样的。不同的用户操作习惯不同，会影响软件的可靠性。软件的可靠性是软件缺陷和用户的可预测性的一个复杂函数。

2、软件规模，也就是软件的大小。一个只有几百行代码的软件和一个几千万行代码的软件是不能相提并论的。

3、软件内部结构，结构对软件可靠性的影响主要是软件的复杂程度，一般来说，结构越复杂的软件，所包含的软件缺陷数就可能越多。在进行软件设计时就要有意识地采用各种降低复杂度的架构策略，如模块化设计，分层设计等等。分而治之的方法是最好的降低复杂度的方法。

4、软件的开发环境和开发方法，软件工程表明，软件的开发方法对软件的可靠性有显著地影响。例如，与非结构化开发方法相对，结构化方法可以明显减少软件的缺陷数。

5、软件的可靠性投入，软件在生命周期中的可靠性投入包括可靠性设计、可靠性测试、可靠性管理和可靠性评价等方面投入的人力、资源、资金和时间等。

我根据系统本身的特点，结合以上五个影响软件可靠性的因素，除了加强可靠性管理外，我制定了提高系统可靠性的三点措施。

一、应用架构设计风格和设计模式，降低软件设计的复杂度。好的设计是成功的一半。在项目开始我就牢牢把握设计关。在进行系统架构设计时,项目组一致认为采用分层设计比较符合实际情况。按层次由上到下分为应用层、中间层和数据层。中间层和数据层都采用工厂方法设计模式,来获得具体的所需对象。分层设计的优点是层与层之间

通过接口通信,下层为上层提供“虚拟机”,这种设计方法为数据采集子系统支持各种类型的表计和丰富的现场总线提供了方便。在通讯子系统中,我们采用管道-过滤器风格的设计,并辅以设计模式的命令模式。命令帧的格式可以明显分成三部分:帧框架处理、应用数据处理和具体功能处理。帧框架处理器对帧长度和帧校验进行处理,成功后将命令帧的帧头帧尾、帧长和校验码去除,提取出应用数据后交给应用数据处理器,应用数据处理器主要进行帧序号处理、帧时限处理和用户密码验证,成功后提取出具体的功能码传递给功能处理器。具体功能实现采用命令模式,这样可以将功能执行部分和命令分析部分解耦。

二、采用补采及数据校验机制,保证数据的完整性和正确性。用电信息采集系统对采集数据的完整性和正确性要求非常高,完整性要达到98%,正确性要达到100%。我们对采集子系统进行分析,发现影响采集成功率的主要原因是采集信道的不稳定,现场总线目前主要有RS485和电力线载波。而电力线载波的抗衰减和抗干扰的能力都比较差,导致采集成功率降低。为了达到要求,数据采集子系统增加补采功能,对于未采集成功的数据进行多次重试。数据在存储和传输时都进行数据校验,最大限度防止出错。在采集终端中,数据文件是主要的存储方式,我们采用“校验和”的方式对数据文件进行正确性校验。采集数据在从表计到采集终端这一部分主要采用电力线载波进行传输。由于电力线载波的不稳定性,极易导致数据出错。我中加入CRC校验的方式来保证数据的正确性。另外由于表计行度等用电信息都是采用BCD码来传输和保存,在数据处理之前对数据进行BCD码验证,发现非BCD码则说明数据错误。通过这些手段,有效地保证了数据的完整性和正确性。

三、在采集终端中采用看门狗和进程心跳检测机制。采集终端安装在现场,由于维护较麻烦,需要提高采集终端的可靠性。采集终端采用LINUX系统,多进程设计。守护进程负责喂看门狗和对各子功能进程进行监测,发现子功能进程不正常则进行子进程重启。

最终项目成功上线，凭借软件设计过程中出色的可靠性设计，系统正常运行了近一年,收到各方好评。通过本次开发实践我明白了软件的可靠性在实际应用中的重要地位。要提高软件的可靠性就要在先期开发时就重视软件的可靠性设计，实施可靠性管理，并将软件的可靠性原则和思想贯穿于在项目的各阶段和各生命周期。另外软件可靠性设计技术还需要多种技术的有机组合，所以要求设计者和编程者要熟悉掌握各种技术，灵活运用，为提高软件可靠性而努力。这些都是我在今后的系统架构设计工作中需要注意与改进的地方，也是日后我应该努力的方向。