**现代软件工程-案例作业**

**(第4组 课程报告)**

**指导教师 ： 姚郑**

2021年1月

小组成员

|  |  |  |
| --- | --- | --- |
| **小组名称** | **第4组** | |
| **小组成员（按学号排序）** | | |
| **姓名** | **专业** | **学号** |
| 赵旌 | 电子信息 | 2020Z8020682046 |
| 魏庆功 | 电子信息 | 2020Z8020682051 |
| 郎桾侠 | 电子信息 | 2020Z8020682062 |
| 王伟杰 | 电子信息 | 2020Z8020682064 |
| 武帅 | 电子信息 | 2020Z8020682073 |

|  |  |  |
| --- | --- | --- |
| **修订记录** | | |
| 状态 | 修订人 | 内容 |
| 创建 | 全体 | 构思讨论案例主题及分工 |
| 增加 | 武帅 | 创建1,2和4.2 |
| 增加 | 郎桾侠 | 创建3.1和3.2，4.4 |
| 增加 | 王伟杰 | 创建3.3和3.4，4.1 |
| 增加 | 魏庆功 | 创建3.5和3.6，4.3 |
| 修改 | 赵旌 | 创建3.7和4.5，5 |
| 整合 | 王伟杰 | 整合各章内容 |
| 审查完成 | 全体 | 全文 |

**目录**

[1 公司介绍 5](#_Toc62027938)

[2 项目介绍 5](#_Toc62027939)

[2.1 项目背景 5](#_Toc62027940)

[2.2 人员介绍 6](#_Toc62027941)

[3 案例概述 7](#_Toc62027942)

[3.1 项目启动阶段 7](#_Toc62027943)

[3.2 项目需求调研阶段 7](#_Toc62027944)

[3.3 方案设计 7](#_Toc62027945)

[3.4 方案测试 8](#_Toc62027946)

[3.5 上线调试 8](#_Toc62027947)

[3.6 试运行 9](#_Toc62027948)

[3.7 回溯与debug 10](#_Toc62027949)

[4 问题改进 12](#_Toc62027950)

[4.1需求变更 12](#_Toc62027951)

[4.2 工程思维改进 13](#_Toc62027952)

[4.3 版本控制 16](#_Toc62027953)

[4.4供应商选择 17](#_Toc62027954)

[4.5问题改进-交付影响 19](#_Toc62027955)

[5 项目后续 21](#_Toc62027956)

# 1 公司介绍

北京市中科创新科技有限公司是一家移动机器人控制系统及无人巡检解决方案的研发和生产商，由王伟杰等人在2020年6月6日创立，总部位于北京市海淀区。该公司致力于为移动机器人工业、行业用户以及具体定制化巡检应用提供性能最强、体验最佳的革命性智能机器人产品和解决方案。 我们的愿景是，拒绝低效重复工作、解放巡检生产力！公司现有两个部门，研发部和客户部，研发部负责产品的研发制造，客户部负责产品推广 ，销售，客户维护工作。

# 2 项目介绍

## 2.1 项目背景

机器人行业当前正发展得如火如荼，无人驾驶、物流机器人、无人机等行业得机器人市场正在被多方巨头一步步侵吞着，王伟杰、武帅等人本是多个机器人公司的核心高管和技术骨干，但随着越来越多的机器人公司学习公司的运作模式，复制而来，机器人市场同质化严重，该赛道机器人利润不断被压低，内卷严重，庞大的公司一时间难以转型。与此同时，巡检机器人行业的商机渐渐显山漏水，模块化开发，快速完成定制化场景交付，解决巡检行业的低效重复性劳动。进一步解决巡检行业的人力生产力。同时，国家大力支持，多方政策鼓励机器人行业解决该行业的技术难题。

王伟杰等一行人，由于研究生阶段是同学，心怀抱负的男生们不甘再受行动迟缓的机器人公司拖累，多人共享意见，一拍即合，决定各取所长，辞职下海，用这个巡检项目作为公司开门的第一炮。巡检机器人项目就此拉开帷幕。

## 2.2 人员介绍

|  |  |  |
| --- | --- | --- |
| 主要人物 | 角色 | 部门 |
| 武帅 | 客户经理/下位机嵌入式工程师 | 客户部门 |
| 王伟杰 | 项目经理/硬件工程师 | 研发/客户 |
| 赵旌 | CTO | 研发 |
| 魏庆功 | 上位机软件工程师 | 研发 |
| 郎桾侠 | 软件算法工程师 | 研发 |
| xxx | 机械外包设计制作 |  |

# 3 案例概述

## 3.1 项目启动阶段

机器人研发部门是公司核心业务的重要组成部分，公司领导层对此高度重视，将骨干人员安排到研发项目组中，并且为了尽早开发出成熟的产品，提高竞争力，还另外从外包公司GKDev招了10位研发人员到项目组中，经过统一培训后，和项目组原有成员一致调遣。另外经过商定，选择了RBT公司作为原材料供应商，采购研发所需的电子器件。

公司召开了誓师大会，王总亲自上台发言，使团队成员士气高涨。他告诉大家，我们团队虽然规模小，人数少，但是都是对行业真正热爱的精英，我相信，大家能够高效率地完成我们的项目，这个项目作为公司的核心，做好了，大家以后都是元老级别的人物，未来可期！

之后，项目组的核心技术人员开会讨论，决定项目管理上借鉴敏捷开发模型。

## 3.2 项目需求调研阶段

甲方非常看好项目的前景，对项目高度重视，从集团内部抽调人员组成专家考察组进行需求调研。在了解了业务，搜集了相关的业务流程和文档资料以后，汇总整理，给出了相关意见。对此，王伟杰将项目根据业务职能部门分为硬件开发组和软件开发组，将技术骨干和外包人员根据自己的技术栈安排到两个开发组中。

需求分析阶段，采用快速迭代方式，使用html展现项目架构，结合ROS构建的模拟场景，将原型系统每周迭代交给专家组进行功能评审。每个分组平均经过五六次迭代后，功能、界面基本确定。在三个月后，经过两轮需求确认工作，各个业务职能部门的负责人在需求确认书上签了字。

## 3.3 方案设计

根据前期的调研，结合了客户的多次修改意见以及其应用场景的描述，最终决定机器人基本的功能包括：识别，拦截，验证和记录报警。定位为巡检机器人。

鉴于客户纷繁复杂的需求，王总带领公司骨干召开了一次座谈会，经过一上午的激烈讨论和对客户要求的认真的分析，最后确定了一套相对满意的方案，客户经理武帅无话可说。

方案中针对巡检机器人最重要的识别问题，算法工程师郎桾侠建议使用基于深度学习的人脸识别技术，软件工程师魏庆功就此技术的应用环境进行了分析，表示使用此技术可以满足客户所要求的环境中的身份识别问题。解决了主要的需求之后，其他辅助的功能的确定也就很快的完成了。其中拦截功能的实现通过机器人识别跟踪功能，不断的在前面跟随陌生人移动，挡住陌生人的去路，同时爆出响亮的报警声以及耀眼的闪光灯来进行报警，同时将违进人员的信息通过无线网上传到服务器，以便记录跟踪。

## 3.4 方案测试

我司采用如火如荼的机器人操作系统ROS的总体架构，来对我公司的巡检机器人进行仿真测试。首先根据确定的机器人尺寸来对机器人的仿真模型进行设计，然后使用gazebo来对机器人功能进行实现和测试，查看在仿真环境下是否具有不可实现的功能和功能之间不可调节的矛盾。

仿真通过后，就要紧锣密鼓的进行实验室环境的机器人实现，将各个部门设计和定制的模块系统交由测试部门测试，确保单个模块没有问题后，然后再让有关联或者需要通信的模块部门手牵手一起联调，联调成功后交由我司总研发部门对产品进行综合测试。

## 3.5 上线调试

测试完成后，产品将进行上线部署和试运行。上线部署是把我们开发的整套系统安装到使用现场，并针对现场的场景进行一些特别的调试。  
 1、我们的机器人是一个独立的系统，由于机器人的工控机是X86平台，安装有Ubuntu操作系统，并在上面安装了ROS环境，机器人的全部功能均基于ROS实现，所以调试只需要使用电脑用SSH连接Ubuntu即可进行。如果场地环境中有大面积覆盖的WIFI网络，理论上调试也可以通过远程进行，不过要是涉及到硬件的调试或者为了保证现场的安全，原则上调试过程必须有专业工程师在现场。  
 2、我们的机器人系统一般不需要外部环境中安装其他的辅助设施，除了在附近场地提供一个充电桩之外，最好能提供WIFI覆盖的环境。  
 3、我们考虑到调试现场遇到的问题五花八门，现场的情况可能只有现场的工程师最清楚，所以我们把操作流程的决定权下放到了调试工程师的手里，并未制定一套具体的调试流程规定。调试工程师在遇到问题时具体分析，并电话与相关功能的开发工程师保持实时沟通来确定原因，并制定解决方案，然后执行。  
 4、调试工程师会维护一个《xxx项目调试记录》表格文档，上面详细记录了发生的时间、出现的问题、软件系统的版本号、是否完美解决、采用的解决方案。该文档作为调试工程师的重要工作产出，会在定期的会议中进行总结汇报，并作为绩效考核的主要依据。  
基于以上的情况，我们的上线调试主要流程为：  
 将机器人运到现场——————连接WIFI网络——————启动机器人，测试每项功能是否正常——————若出现问题，由现场调试工程师制定方案并解决，开发工程师实时远程支持，并记录文档

## 3.6 试运行

机器人系统部署完成后，将进行为期168个小时（一周）的试运行过程，来测试机器人产品的稳定性、可靠性、与环境的适配性，并由甲方评估产品的功能性是否满足需求，是否有需求上的改动等。  
 试运行过程由我方派驻的调试工程师进行现场维护，他们将解决机器人在运行中出现的小bug（如某硬件故障更换、螺丝松动、接触不良、操作系统死机重启等），记录并上报重大bug（如功能性bug，在现场环境下光线问题影响图像识别功能）和重复出现的bug（如系统时间每次开机都不同步，需手动同步问题），并根据运行情况与产品经理、甲方人员沟通确定是否有可优化的地方。  
 如果试运行中产品没有出现致命错误（如核心功能无法使用），并通过了甲方的确定，则产品可以初步上线运行。后续bug修复、功能增强、系统优化则在后续的迭代升级中逐步进行。  
 如果出现了致命错误导致试运行失败，则由负责人开会总结问题并制定补救方案。

## 3.7 回溯与debug

历经千辛万苦终于将巡检机器人试运行了，但是整个前期开发过程中却出现了很多的问题，原本计划的是整个过程采用敏捷方式进行管理，将整个机器人拆分成不同的需求部分来进行不断的迭代开发，然后根据实际运行结果来进行随机应变增改需求，可是实际面临的变数远超最开始的想象。

* 对软件结构设计得不够灵活：由于开发人员并不多，最开始大家都是集中在一个大工程进行开发，每个人负责一部分功能，开发后进行整合，但是最后下来却发现由于为了赶时间，许多代码仅仅是为了实现相关功能，基本没有复用性，整个工程的耦合性较强，一旦发生了需求的变更与修改，将是一个巨大的工程，一个的变更就会牵扯到一大堆模块的适应性修改，浪费了许多时间。
* 对整个工程的把握不到位：每个人都有一个单独的代码风格，虽然每个人独立的完成了自己的模块，但是开发人员之间的工作却没有可替代性，没有对整个软件的开发规范制定一个统一的标准，工作之间相互很难看懂对方的代码，由于没有及时留存项目文档，工作的交接基本靠口传心授，因此对中间涉及到人员变动时工作交接造成了极大的不便，甚至还会有已经离开项目组的工程师被叫回来重新指导新的成员进行工作。
* 没有及时进行版本管控：虽然前期开发阶段并没有显现出太多问题，但是一旦到了后期的测试以及需求变更时麻烦就接连而至，版本管控的不及时导致新的需求到来时经常使得机器处于停工状态，旧的功能实现都遭受了影响，并且很难还原到之前的工作状态，还要对先前已经测试过的问题重新进行再一次的测试，现场实施人员叫苦不堪。
* 供应商选择的不可靠：由于整个机器人各个零部件是由不同的供应商进行提供，就算同一个零部件也出现了不同的供应商都只能满足一部分需求的情况，完美的零部件要么不存在要么超出预算，然而这是先前进行调研过程中所疏忽的。
* 现场实施过程中的各种麻烦：机器人在现场实际应用时，对当地原有的生产生活环境造成了影响，甚至有些测试导致的事故造成了一定的损失，使得测试不得已而暂停，对受损设备和零部件进行更换维修。

# 4 问题改进

## 4.1需求变更

项目初步的模型已经定义完全，在前期也进行了紧张有序的测试和调试。然而一个产品是由一个或多个项目组成，所以需要考虑的情况有很多。所以不管是互联网的产品，还是传统IT的甲方项目，需求不定是常事，变是唯一的不变。需求的变更发生在任意的一个项目过程，它针对的是一个过程的影响，进而给整个产品带来重大甚至致命性的影响，轻则延期发布，重则分崩离析。任何项目都有一个特色，那就是项目之前群情激昂，至于过程和结果，有的怨声载道，有的劫后余生，万象丛生都很正常。

就在和甲方客户第一次聚头洽谈的时候，甲方客户针对我们的初次方案进行了仔细的研究和谈论，提出了下面许多需要改动变动的地方。

识别率问题：现在的人脸识别领域已经可以做到百分之九十多的人脸识别技术，然而由于客户不太了解技术方面的细节和实现，竟然提出需要百分之百的识别。甲方人员表示人能够识别的场景机器人也同样要能识别出来，这样才能显示我们这项技术的应用价值，而且要求在各个场景中（光线强弱，人脸有部分遮挡等情况）的识别率均达到百分之百，这种变态的要求需要算法软件工程师尽力去提高，实在不行就只能靠客户经理来进行调解。

提拉需求：在本次项目中，项目的识别与拦截是该项目的核心和重点。这也是我们需要关注的点，但是甲方人员表示，由于最近他们公司发生了一起十分重大的盗窃案件，警方已经进行了深入的调查。在和公安局配合的过程中，公安局的警务人员对于公司不能提供一个月前的公司人员的活动记录和外界人员的活动报告大感恼火，因为这是破解案件的关键，所以到现在这起案件还是不温不火，未能侦破。鉴于以上的情况，甲方负责人也是甲方公司的大领导特别强调，这次的巡检机器人一定要弥补这方面的短板，所以要求我司必须能够提供至少90天的数据可追溯和保存服务。这相当于强行要求我们将记录的需求放在首位。

增加查询功能：甲方人员绞尽脑汁又想到了一个新的需求，那就是能够提供大概的业务查询和地图导航功能。甲方表示，由于公司实在是太大，许多第一次来办业务的人员都会迷路。或者不懂得业务的办理流程，常常需要回一楼大厅向前台进行询问，这实在是不太智能，于是甲方领导要求本次的巡检机器人能够提供类似大型商场的导航机器人的功能。要求在机器人的某个部分上能够加上一个触摸屏，在屏幕上能够把本机器人所在楼层或者区域描述出来，当有人在查询的时候，可以直接用手指来进行愉快的交互。其中的功能包括：地图查询，业务查询，紧急报警。

针对这次的洽谈，王总颇为头痛，没想到甲方这么会出幺蛾子。但是提出的问题和需求还是要解决的，王总根据Kano model对需求解决的优先级进行了分类，看看那些需求是立马就需要解决的，哪些需求是到最后才能决定是否能够更改或者添加。

王总经过彻夜的研究，决定：首先本次项目的Must-be Quality是识别，所有的功能都建立在这个基础之上，所不论是在技术上进行尽可能的提高，还是让产品经理进行沟通说服，一定要把识别率这个问题解决掉；然后研发人员需要针对甲方公司数据可回溯的这样一个紧急需求开出一个方案；至于查询功能，这是一个Attractive Quality，若果有余力就尽力去解决，并提高估价。

## 4.2 工程思维改进

整套机器人系统成功落地，在众多的投资人和媒体的长枪短炮关注下，顺利的完成自动唤醒、自动巡检、触发报警、低电量自动充电等多项智能任务，巡检人员只需要坐在温暖的巡检室等待报警触发，或者随意观察下机器人传回大屏幕的实时图像就好。一时间，受到了极大的关注，这意味着，兄弟们的梦想已经成功的迈出了第一步，接下来只要拿着投资人的投资，成功量产、售出、资金回笼、售后支持，成功树立和巩固住当前的成果，那么 之后如果顺利上市，兄弟们实现财务自由便指日可待。

大家沉浸在这种快乐的幻想中，量产的计划也在推进着，这个时候，有趣的事情发生了，第一批发到客户手中的机器人投入使用的一周后，出现了一些故障，无法正常完成巡检任务了。根据客户的描述如下

* 正常运行几天后，再次启动无法执行巡检任务。
* 机械结构没有损坏、指示灯全部可以正常亮起
* 上位机远程控制无法唤醒传感器

于是根据初步判断，可能是软件部分出了问题，公司决定让软件工程师出到现场为客户解决问题。为了不耽误客户的工作周期，软件工程师在周五晚上背上电脑，趁着夜色踏上了征程。第二天顶着黑眼圈去到现场为客户进行错误排查。软件工程师打开机器人，连上电脑，熟悉的运行着测试指令，只见机器人哔哔~~两声，咻的一下抬起了头，底盘电机嗡嗡作响，进入工作状态，但是视频镜头迟迟打不开。不好。。软件工程师心头暗道一声。是摄像头模块坏了。再一想还好摄像头是定制的模块化产品，更换一下即可。转眼，软件工程师拿着各个型号的螺丝刀捣鼓了起来，由于机器人整体没有采用良好的模块化设计，机械模块间耦合性极高，牵一发而动全身，里面的布线弯弯绕绕，就为了拆个摄像头出来，差点把软件工程师折腾死，不过幸好，这次的更换只涉及了上半身的执行结构，下半身的底盘没有太多被波及到。

直到黄昏，软件工程师终于把摄像头换了上去，勉勉强强的把机器人安装起来后，上电、启动、指令测试，机器人运作了起来。正当软件工程师准备松一口气的时候，只听得机器人内部的一声“叮”响，短路指示灯顷刻亮起，机器人呜的一声停在了那里，和工程师大眼瞪小眼。工程师哭爹骂娘的边骂边检查，发现是一颗螺丝没安装好，机器人走动的时候一晃荡，小螺丝直接掉落，该巧不巧的落在了下位机嵌入式硬件板上，大面积的电路顷刻间短路，过电流的强热瞬间烫起了覆铜板上松香的清香。得，软件工程师彻底蔫了，电路板的设计采用的集中式设计，一损全损，软件工程师按下摄像头这边的瓢头，那边扬起了硬件板的瓢尾。

无奈，这不是他自己能搞定的事情了。只好让客户帮忙，一起把机器人发回公司进行整体维修。一路上，软件工程师都在反思这次的问题，复盘着，如何避免问题的出现？如果什么样的情况下，自己一个人就可以搞定这个问题？

回到公司后，大家没有相互埋怨，这次发生的比较早的维修，让大家深刻的认识到了机器人工程的重要性。大学课上学的东西此时貌似开始出现在他们的脑海里。

* 注意机器人系统之中的机械、硬件、软件相互配合，
* 各模块之间要实现高内聚低耦合，方便维修和更换
* 各个单元要有能够简易的单独测试的能力，方便Debug
* ....等等。

这个时候王老板把大家召集到一起，开始回顾大家的设计历程，当时整体分为了底盘、上身执行机构

底盘由机械和硬件工程师完成，只负责运动，克服地形困难，留出通讯接口等待上身结构发出指令。大家有过多次的机器人比赛经历，所以这一块做的还是比较好。但是，硬件工程师为了图方便，把底盘的整体板拼凑到了一张板子上，在板子上方还没有做很好的隔离保护，就导致了后面的惨案发生。机械工程师在整体结构设计上一心求快速落地，忽略了结构的耦合性，没有和硬件工程师很好的沟通线路走线情况，导致安装时线路混乱，生产和调试效率都很低下。

上身执行结构为机器人核心功能的实现，需要机械、硬件、软件工程师的通力配合。机械+硬件实现了摆头云台，软件和算法对摄像头的采集数据进行处理，识别环境和巡检现场情况，检测后并向执行机构和底盘发送指令，

首先机械设计时没有把上身的整体结构设计成快拆式的，导致软件工程师拆个摄像头用了整整一天。

其次软件工程师的远程调试功能设计的不够合理，不具备单独调试功能，使得其只由到达现场才能进行调试，效率低下，不够赛博朋克。应该调整这部分的软件架构，并且设计一套简单的用户可自行做排查的软件测试流程 ，能不出差尽量不出差。节省时间和人力资源。

一通反思后，大家决定推迟2个月的上线时间，对市场上的测试机执行若故障必召回的策略，公司上下，同心协力，开始对结构和产线进行优化调整。从机器人工程的角度重新审视这个项目。

## 4.3 版本控制

我们的产品开发过程按功能模块和软硬件部分进行了任务分割，每个模块由一个小组或者一个人来完成，每个小组里可能又有再细的分割，每个人又维护多个版本的代码，所以代码的版本控制是一件值得下功夫的事情。 在项目初期，版本控制并没有受到团队的重视，采用的策略是瀑布分割任务，任务到每个人再又个人进行敏捷开发，给了个人较大的自由度，每个人使用的版本控制软件都各不相同，有人用了git作为工具，使用了github作为托管平台，因为github在国内访问速度很慢，而且容易超时，他就长期使用科学上网工具翻墙来加速。另一个人同样使用git，但是托管平台使用了位于国内的gitee，但是部分功能需要付费充值才能使用。另一个开发者干脆直接用自建的文件夹来组织不同时间的修改，虽然也能用，但是比较麻烦容易出错。 因为缺乏统一的版本控制，系统的各功能模块联调的时候大家就各自拿出自己较为满意的一版代码，东拼西凑的将各模块拼在一起。虽然每次联调都能成功，但是费时费力，效率很低，且联调后确定的一版整体的可运行的代码命名也很随意，导致经过多次版本迭代后，大家各自也都忘记了自己负责的那个模块用的是哪个版本，然后就出现了一些可笑的情景：在上一个版本中已经修复好的bug，迭代了两次竟然又出现了,因为另一个模块的开发者用错了这个模块版本，等等。 我们其实一早也意识到了这个问题，但是大家谁都没有提出来，直到出现了很多次因版本不可控导致的严重问题，负责人王伟杰终于坐不住了，为此专门开了一个研发部门的全体会议，讨论建立一套体系完善、执行严格的版本控制规则。经会上友好、和谐的交流，我们最终制定出了一套方案： 1、为了统一版本控制工具，规定都使用git，考虑到商业代码的安全性，托管平台不使用第三方，公司内部将专门部署一台服务器，其上部署开源的托管工具GitLab，并且仅公司内网用户可以访问。设立一个服务器维护职位，由IT部门的老哥兼职担任，平台管理员又负责人王伟杰亲自担任，拥有最高权限。 2、对gitLab服务器进行多机冗余备份、定期备份，确保数据安全。 3、每个项目建立一个仓库，由管理员创建对应模块的目录，并邀请开发人员账号加入项目，开通指定的访问权限。 4、每个开发者维护自己的模块目录，commit自己的代码部分，并且由系统进行版本控制。 经过这次会议后，大家统一了版本控制的工作。虽然规定发布的初期有部分人不熟悉、不习惯这套工具和流程，出了一些忘记提交、提交被误删、无法同步commit等小问题，但是长期使用下来，版本控制显然奏效了，因版本控制出的问题大幅减少。

## 4.4供应商选择

项目采购是指在整个项目过程中从外部寻求和采购各种项目所需资源的过程。机器人研发过程中需要采购的零件由很多部分组成，必须提前设想好合适的元器件组合，在保证质量符合要求的情同时也要维持在一定预算内，如果没有事先考虑周全，可能会在开项目研发后期新增许多开支。

中科创新科技有限公司现有的技术人员大部分是软件开发出身，其余少数硬件开发者只专精于部分领域，对机器人研发的器件采购也是只有一个大概的观念。由于大家对机器人项目落地缺乏经验，在元器件采购的过程中出现了些许问题。

公司研发的巡逻机器人，核心技术之一就是人脸识别，公司的软件开发工程师对于人脸识别的技术都有深入的研究，在最初从RBT公司采购的摄像头上应用了第一版人脸识别代码。从结果来看，人脸识别的效果不错，在大部分情况下能够正确识别人脸信息。大家都满怀喜悦，觉得项目进展得非常顺利。然而在一次会议上，王总在听完了进度报告后，敏锐地发现，研发组并没有对多人同时出现在场景内的情况进行性能测试。为了应对人流量较大的场景，研发组之后进行了多人脸识别的测试，随之出现了识别人脸失败、处理过程缓慢等情况。

这让本来轻松许多的技术人员愁眉苦脸，大家加班加点开始优化算法，发现性能提升幅度不大。正在大家一筹莫展之际，硬件工程师小王提出了一个观点，是不是选择的摄像头有问题？公司之前在RBT采购的摄像头是双目RGB摄像头，在计算三维深度的时候会收到许多外界因素的干扰，如果使用RGB-D深度相机，能够通过物理手段直接测量出图像的深度信息，会不会对结果产生改进？

小王的想法让大家觉得眼前一亮，觉得摄像头应该是问题所在，于是打算重新采购一批RGB-D摄像头。在选择新的摄像头型号时，大家一如既往地在老牌供应商RBT公司的官网浏览产品信息，找到了对应的RGB-D摄像头产品。而作为一个相机爱好者的小王敏锐地发现，RBT公司的这款RGB-D摄像头除了多了深度检测的功能之外，其他的基本的相机参数，比如分辨率还赶不上之前采购的双目RGB摄像头，如果贸然采购，说不定做出来的人脸识别效果还不如之前的版本。于是大家又重新在网上筛选信息，物色合适的产品。最终找到了一家国外的摄像头供应商CMR公司，它提供的RGB-D摄像头和RBT公司的差不多，但是能够根据客户的需求定制生产相机的参数，也因此其价格要远高于RBT公司的两种相机产品，这让大家又不得不犹豫了起来，如果换用新供应商的摄像头元器件，用高额的成本换取更好的人脸识别效果，是否值得。

大家为此商议了一番，在这期间王总对此高度关注，也参与到了商讨会议中。在听完大家的讨论之后，王总展现出了企业家的魄力，拍板决定，如果CMR公司的RGB-D摄像机效果能满足需求，那么就将摄像头元器件的供应商换成CMR公司。

于是公司的采购部门向CMR公司提交了定制的RGB-D相机订单，在经历了一个多月后，一批崭新的RGB-D摄像头远渡重洋运送到了公司里。研发人员兴奋地调试好摄像头，并迅速将代码部署，应用到新的方案中，经过数轮人脸识别测试，终于取得了满意的效果。

在新一周的会议上，王总听取了测试报告，满意地点头说，能够做出出色的产品，就是成功的一步。虽然经历了这次改进，我们产品的制造成本提高了不少，但是还是值得的。只要有利润在，只要我们的产品够出色，就不怕没人买，就不怕赚不到钱。与此同时，王总也指出，一个成熟的采购方案对于公司是至关重要的。在项目前期，要对产品供应链有更全面的调查，既要明确自己的需求，也要了解供应商的信息，同时也要拟定好备用计划，防止意外发生。

这次会议后，大家学习了CMMI的供应商管理的知识，经过科学合理的评价、比较，最终选择了CMR公司作为中科创新科技有限公司巡逻机器人项目的摄像头供应商，提供后续生产过程中的RGB-D摄像头元器件。虽然CMR公司的RGB-D摄像头不是价格最优的，但是能够提供定制化的需求，是最适合公司巡逻机器人的产品。如果之后的项目流程中，能够将CMMI的理论有效地应用到供应商制造、供货、产品接收等多个环节中，那么对于整个项目的顺利实行是非常有益处的。

## 4.5问题改进-交付影响

历经千辛万苦，多次软件版本的迭代联调，全组人员没日没夜的共同努力，巡检机器人终于将开始实地试运行，也即将进入交付的最终阶段，但是交付的现场却不断地遇到了麻烦。  
 刚刚到达现场开始部署实地测试环境，一切看起来还算顺利，通过几个不同的人来进出公司测试机器人在不同时间、不同环境下的工作状态，第一个半天时间里机器人能够像在实验室环境下一样准确的来对不同人员进行识别和对未记录人员的追问，但是到了下午突然出现了问题，就在一个走廊的拐角处机器人发现了一个新来人员便向其前进来执行判断任务，但是越过拐角后却突然失去控制直接向墙上撞去，最终触发机械式防撞机构而停止。  
 现场的测试人员慌了，刚第一天的试运行就出现了莫名奇妙难以理解的问题，马上电话联系了CTO，CTO立刻叫齐技术人员们开会调查，从现场日志开始调查，在抓取了多个模块的工作日志后，逐步将问题由软件业务层定位到算法层，再由算法层定位到了硬件层，最后确认了问题的两个点，一是业务逻辑方面对于异常处理不够严谨，第二是由于工作地点在对方公司写字楼，电磁环境复杂，机器人为了自身的续航和机动性仅搭载了一台性能并不是非常强的电脑用于导航和采图及一些前期的简单运算等工作，而机器人用于存储面部体态信息的数据库和一部分识别算法的运算过程经常需要通过路由器AP来和一个外置的服务器进行通信，因此在过道处由于AP受到干扰而导致了异常的发生。  
 当即就总结了为什么会发生这两个问题，业务逻辑方面不够严谨是因为对于这类运行环境缺少经验，没有想到这类的异常，因此今后可能要在类似的开发过程中就不断通过头脑风暴和半实物仿真来找出类似可能存在的问题，同时还要提高整个软件框架的健壮性。而第二个问题就比较严重了，这是先前完全没有预料到的，组员们当即想出两种解决方案，一是更换其他频段的AP，但可能需要重写相关驱动和通信协议并重新进行测试；二是将服务器直接部署于机器人本体上，但可是一个巨大的工程，需要给机器人身上加一台足够强大的服务器，同时需要更换更强大的电源来保证能源供应，而这不但需要调整机械结构，更会降低机器人自身的机动性和续航能力，若机器人出现损坏则可能同时损坏机载服务器而导致已有数据丢失，但唯一一点的好处是这样的机器人即将完全独立于服务器的存在，更具自主性，更像一个“人”了，也对以后这样的结构可能造成的问题进行了一个规避。  
 随后开始从风险、成本、周期等角度来对两种解决方案进行评估，最终还是决定选择保守方式，采用更换其他频段AP的方式来解决，同时将这个需求变更排上日程，安排出相关模块的开发及测试工作。  
在之后的实地测试中，又遇到了新的问题，由于巡检机器人经常要穿行于一条车来车往的过道，在经过岔路口时无法看到两边情况而会出现与车辆相撞的风险，且出现过多起机器人横穿路口而逼停车辆的问题，给对方公司造成了一定的麻烦，接到了不少次的投诉，客户经理赶忙去跟对方公司进行交涉，同时承诺会将这个作为最高优先级的问题来处理，在分析了所发生的情况之后，开发团队决定从已有的工作逻辑入手，在通过路口的过程中不断观察两边道路情况来决定是否继续前进或原地避让。  
 不断类似的场景实践丰富了团队在这方面的从业经验，新需求（或bug修复）的插入-实际场景分析-制定解决与开发计划-进行开发-模块用例测试和整体测试-部署，经过不断的迭代，使得机器人系统日渐成熟，鲁棒性和代码的可复用性也提升了。

# 5.项目后续

最终机器人成功交付运行了，而开发团队也有不少的收获，从最开始的不知所措手忙脚乱到后来的井井有条，CTO发现这个项目不仅让开发人员互相对对方的水平认识得更加深刻，更是催生了一套标准化的开发框架和代码规范，可以在后续其他项目中充分借鉴，PM发现无论是需求分析还是供应链管理均已达到CMMI二级标准且接近三级，在公司的发展道路上是一个新的里程碑。