

# 物品检测与抓取机器人

## 项目开发总结报告(PDSR)

### PDSR102

### V1.0

项目组成员信息		
小组名称	rushbot	
学号	姓名	本文档中主要承担的工作内容
17373060	杨开元	软件系统总结
17373517	王正达	文档总结
15241035	彭文鼎	开发过程与技术总结
17373288	尹俊成	风险与经验总结

2020-06

#### 版本变更历史

版本	提交日期	主要编制人	审核人	版本说明
V1.0.0	2020.06.15	彭文鼎	尹俊成	本小组项目开发总结报告初稿

## 目录

1.1 标识.....	3
1.2 系统概述.....	3
1.3 文档概述.....	3
1.4 术语和缩略词.....	4
2 引用文件.....	4
3 实际开发结果.....	5
3.1 产品.....	5
3.1.1 项目版本说明.....	5
3.1.2 软件单元说明.....	5
3.1.3 数据说明.....	7
3.2 主要功能和性能.....	8
3.2.1 启动和关闭机器人.....	8
3.2.2 初始化场景建模.....	8
3.2.3 货架位置标记.....	8
3.2.3 指定商品获取.....	8
3.2.4 导航避障.....	8
3.2.5 语音控制.....	8
3.3 基本流程.....	9
3.3.1 设置计划.....	9
3.3.2 分析需求.....	9
3.3.3 详细设计.....	9
3.3.4 实际开发.....	10
3.3.5 软件测试.....	10
3.4 进度.....	11
3.5 人员工程量.....	12
4 开发工作评价.....	13
4.1 对开发效率的评价.....	13
4.2 对产品质量的评价.....	13
4.3 对技术方法的评价.....	14
4.4 问题原因分析.....	14
4.5 风险管理.....	15
5 缺陷与处理.....	15
6 经验与教训.....	16

# 1 引言

## 1.1 标识

本文档适用的范围如下

- 系统名称：rushbot 物品检测与抓取机器人
- 编号：102
- 版本号：V3.0

## 1.2 系统概述

本项目是实现一个可以自主移动、主动避障以及对目标物进行检测和抓取的简易机器人，可以应用到超市等场景。它拥有自主分析地图信息以及在此基础上预先进行路径规划的功能。为改善人机交互体验，加入语音识别与控制系统。

系统开发采用迭代-增量模型，先完成了机器人基础功能（移动、建图）的开发，进而完成了进阶功能（导航、避障、检测、抓取、语音）的开发。此外值得说明的是，测试是与开发同步进行的。

由于远程开发的条件限制，系统当前的运行现场是 gazebo 模拟器，计划运行现场是实体机器人的工作环境。

## 1.3 文档概述

本节概述本文档的用途与内容。

本文档是 102 小组 ROS 机器人项目最终的总结报告，主要用于对一个学期以来的软件开发学习过程做一个总结与回顾。

本文档主要包括以下六个部分：

### 1. 引言

概括性描述本项目的内容以及本文档的用途与内容，解释相关术语。

### 2. 引用文件

列出本文档引用的和产生的所有文档及代码的编号、标题、修订版本、日期。

### 3. 实际开发结果

介绍开发的最终产品功能，总结开发的主要流程、进度情况以及总体分工。

### 4. 开发工作评价

从开发效率、系统质量、技术方法、风险管理等角度综合评价开发工作。

## 5. 缺陷与处理

列出了在需求评审阶段、设计评审阶段、代码测试阶段、系统测试阶段和验收测试阶段发生的缺陷及处理情况。

## 6. 经验与教训

列出了从本项目开发工作中得到的最主要的经验与教训及对今后的项目开发工作的建议。

# 1.4 术语和缩略词

表 1 术语和缩略词列表

术语和缩略词	简要解释
ROS	Robot Operating System 编写机器人软件的高度灵活性框架
SLAM	simultaneous localization and mapping 即时定位与地图构建

# 2 引用文件

表 2 引用文档列表

编号	标题	版本	发行日期
1	启智 ROS 机器人开发手册	V1.1.0	2018.11.09
2	SDP102	V3.0.0	2020.06.07
3	SRS102	V3.0.0	2020.06.07
4	SDD102	V4.0.0	2020.06.07
5	STD102	V5.0.0	2020.06.08

## 3 实际开发结果

### 3.1 产品

#### 3.1.1 项目版本说明

表 3 项目版本

Version 0.1	完成 Gazebo 仿真环境的探索 安装合适的依赖包完成例程的运行 初步划定代码组织结构
Version 1.0	调整代码组织结构 加入 Gmapping 建图测试和键盘控制单元
Version 1.1	加入导航单元 加入使用手动添加航点测试导航单元的方法
Version 1.2	加入导航单元测试方法：使用坐标指定目标航点、导入航点文件并实现巡航
Version 1.3	加入测试控制台 整理基础功能测试代码
Version 1.4	完成每个模块的测试，记录测试结果 完成可靠性、及时响应等测试
Version 2.0	修改 Gazebo 物理环境，在指定位置加入桌面和物品 模拟机器人实际需求
Version 2.1	加入物品检测模块及其测试单元 完成物品检测模块测试
Version 2.2	加入总控模块建立状态机使得机器人可以在没有语音指令的情况下，周期性往返于目标点之间并实现物品的识别和模拟抓取
Version 2.3	加入语音控制模块，使得机器人在语音指令控制下，往返于顾客和货架之间，实现指定物品的识别和模拟抓取。
Version 3.0	完成系统整体测试，验证系统的可靠性和有效性 完成所用文档撰写，明确机器人使用方法

如表 1 所示，为本系统所有的版本更迭及版本差异说明。在总体上，Version 0 及其子版本主要完成环境的探索，依赖项的安装。Version 1 及其子版本主要完成基本功能模块及其测试。Version 2 及其子版本主要完成功能模块的串联和实际功能的实现。Version 3 为实际交付版本。

#### 3.1.2 软件单元说明

Package: wpr\_simulation

存储位置: [github:sebuaa2020/Team102.git/dependency/demo2/wpr\\_simulation](https://github.com/sebuaa2020/Team102.git/dependency/demo2/wpr_simulation)

该包主要为 Gazebo 提供机器人模型、世界模型和相关配置文件。每次运行

机器人任何之前，都应启动该包下的 `wpb_simple.launch` 加载仿真环境。  
该包为外部依赖包，其中约有 40 个配置文件和 1 个关键 `launch` 文件。在地图配置文件中，我们做了相应修改以添加桌面和物品，约添加配置 100 行，其余均为共享代码。

**Package:** `waterplus_map_tools`

存储位置: `github:sebuaa2020/Team102.git/dependency/demo2/`

`waterplus_map_tools`

该包主要和航点功能有关，通过该工具包，可以方便地添加、修改和调用航点。在测试航点添加功能、保存航点时，需要用到该包中的相关 `launch` 文件。

该包为外部依赖包，包含 11 个 `cpp` 文件、12 个 `launch` 文件和若干配置文件，均为共享代码。

**Package:** `wpbh_local_planner`

存储位置: `github:sebuaa2020/Team102.git/dependency/demo2/`

`wpbh_local_planner`

该包主要和导航功能有关，通过该包中的代码，机器人可以在导航时启用最短路算法规划路径并完成避障。在测试导航功能、机器人运行往返于目标点之间时，需要该包中相关节点启动。

该包为外部依赖包，包含 2 个 `cpp` 文件、3 个库文件和若干配置文件，均为共享代码。

**Package:** `xfyun_waterplus`

存储位置: `github:sebuaa2020/Team102.git/dependency/demo2/xfyun_waterplus`

该包主要和语音功能有关，通过该包中的代码，机器人可以在识别麦克风的语音输入，并将信息转换为字符串。在机器人实现需求功能，等待语音指令时，该包中语音识别模块需要启动。

该包为外部依赖包，包含十数个 `cpp` 文件、3 个库文件和若干配置文件，均为共享代码。

**Gmapping 建图及其测试单元**

存储位置:

`github:sebuaa2020/Team102.git/rushbot/launch/test/gmapping_test.launch`

`github:sebuaa2020/Team102.git/rushbot/src/rushbot_keyboard_ctrl.cpp`

该单元主要功能为建图及其测试，通过 `rushbot_keyboard_ctrl.cpp` 可以实现机器人的键盘控制移动，`gmapping_test.launch` 主要为建图功能测试启动脚本。

该单元中键盘控制模块参考了 `wpr_simulation` 包中的平行功能，添加了键盘相应帧数和输出信息，约 50 行。`Gmapping` 测试启动脚本为独立开发，约 25 行。

**物品检测及其测试单元**

存储位置: `github:sebuaa2020/Team102.git/rushbot/launch/test/obj_detect.launch`

`github:sebuaa2020/Team102.git/rushbot/src/rushbot_grab.cpp`

该单元主要功能为物品检测及其测试，通过 `rushbot_grab.cpp` 可以实现机器人的物品检测，`obj_detect.launch` 主要为物品检测功能测试启动脚本。

该单元中物品检测模块参考了官方包中的平行代码，修改了状态机及相关代码约 100 行。测试启动脚本为独立开发，约 25 行。

#### 导航测试单元

存储位置：github:sebuaa2020/Team102.git/rushbot/launch/test/nav\_test.launch  
github:sebuaa2020/Team102.git/rushbot/src/test/rushbot\_nav\_goal.cpp

该单元主要功能为根据输入目标点导航及其测试，通过 rushbot\_nav\_goal.cpp 可以实现目标点的输入和信息传递，launch 文件为测试启动相关节点环境。

该单元中代码为独立开发，测试 cpp 文件 37 行，启动脚本 43 行。

#### 测试总控及其他测试单元

存储位置：github:sebuaa2020/Team102.git/test/code/test\_modules.py  
github:sebuaa2020/Team102.git/rushbot/launch/test/\*

测试总控单元主要用于和开发者交互的测试，通过相关指令提示，完成各个模块测试。其余 launch 文件主要用于：语音模块测试、航点测试、巡航测试。

该单元中代码为独立开发，测试总控 py 文件 82 行，其余测试 launch 文件 3 个，约 100 行。

#### 服务单元：

存储位置：github:sebuaa2020/Team102.git/rushbot/src/rushbot\_main.cpp  
github:sebuaa2020/Team102.git/rushbot/launch/rushbot\_main.launch

该服务单元用于串联所有的功能模块，监听客户语音指令，指挥机器人前往目标点，调整与目标平面距离，检测商品，虚拟抓取和返回顾客位置的功能。实现状态机以完成机器人的功能需求。Launch 文件完成所有节点的加载。

该单元为独立开发，cpp 文件 261 行，launch 文件 53 行。

### 3.1.3 数据说明

#### 地图数据：

存储位置：github:sebuaa2020/Team102.git/rushbot/data/map.pgm  
github:sebuaa2020/Team102.git/rushbot/data/map.yaml

#### 航点文件：

存储位置：github:sebuaa2020/Team102.git/rushbot/data/waypoints.xml

#### Rviz 配置文件：

存储位置：github:sebuaa2020/Team102.git/rushbot/rviz

#### 仿真环境配置文件：

存储位置：

github:dependency/demo2/wpr\_simulation/worlds/ROS-Academy.world

#### 测试数据日志：

存储位置：github:sebuaa2020/Team102.git/test

## 3.2 主要功能和性能

### 3.2.1 启动和关闭机器人

本产品由于使用仿真环境开发，启动和关闭机器人主要使用启动脚本完成，在一定程度上完成了该功能，和项目开发计划中稍有不同。

### 3.2.2 初始化场景建模

本产品达到了开发计划中初始化场景建模这一功能需求，能够在指定指令下，根据键盘控制，完成场地穿梭和建图，并保存地图。

经过测试，本产品在键盘控制中延迟几乎不可见，较为流畅；在绕场一周的情况下，能够清晰完成地图的绘制；绘制完毕后，本产品能够顺利保存地图至指定位置。

### 3.2.3 货架位置标记

本产品达到开发计划中对货架位置标记这一功能的要求。在地图建立并保存的情况下，由工作人员完成货架位置的标定，之后保存配置。

经测试，本产品在标定货架位置时无明显延迟，保存和导入配置文件均能正确执行。

### 3.2.3 指定商品获取

本产品基本达到开发计划中对指定商品获取这一功能的要求。该功能指本产品接受指令后，导航至货架位置，完成物品的检测和抓取，之后导航返回顾客位置。这一功能中，物品抓取由于硬件问题，没能实现。

经测试，本产品导航路径规划正确、响应迅速，能咋指定时间内到达目标地点。物品检测过程中，能精准调整与平面距离，正确识别平面和物体并进行标定，识别迅速。

### 3.2.4 导航避障

本产品完成了导航避障功能的开发，超过了需求说明书中的要求。该功能使得机器人在行进过程中，能够绕行突然出现的障碍物。

经测试，本产品导航避障鲁棒性高，在避障后能正确前往指定位置。

### 3.2.5 语音控制

本产品完成了语音控制功能的开发，超过了需求说明书中的要求。该功能使得机器人在等待服务状态时，能够监听用户的语音指令以开始服务。



经测试，本产品语音识别控制相应迅速，识别准确，能够正确对机器人状态进行改变。

### 3.3 基本流程

#### 3.3.1 设置计划

这一部分主要介绍用于在开发前明确项目背景、功能，项目使用的相关开发模型，明确时间节点和分工，列出项目开始之前我们想到的技术难点，对整个项目起指导作用。

具体到实施，我们首先基于 ROS 系统和机器人的大体发展情况以及我们小组自己的设想提出了我们机器人项目的目标及潜在应用领域。

紧接着我们对所设想的每个功能要依赖的硬件软件，需要的开发环境，技术选型及其实现的大体思路做了一定的计划与安排。

然后，我们对项目开发、测试和部署过程中可能遇到的风险，做出了预估，并给出了一些应对风险的策略。

最后，我们基于上述分析，初步确定了项目开发的计划进度。

#### 3.3.2 分析需求

这一部分我们主要基于小组在设置计划时提出的项目目标与潜在应用领域对项目的需求与业务逻辑进行了更加细致的分析。依次分析了项目的业务需求、数据需求、功能需求、非功能性需求以及项目的基本开发环境。

具体到实施，我们首先采用从逻辑层面将需求划分为几个不同的用例，对每个用例我们采用文字+流程图的方式进行两方面的详细描述。

#### 3.3.3 详细设计

这一部分主要针对我们小组提出的需求进行详细的设计，主要采用的是面向对象的体系结构，为了满足我们之前需求分析中提出的各个用例，我们设计了不同的模块与各个模块间的接口。

具体到实施，我们首先对需求分析的成果进行了一定的回顾，基于我们需求分析的结果，我们抽取出了项目中存在的不同实体，并据此完成了数据库的设计。

之后，我们将为了完成需求系统所要实现的功能划分到不同的功能模块，划分指导思想是高内聚、低耦合，之后完成模块间接口设计。

对于不同的模块，我们首先画出模块功能流程图，基于流程图以及 ROS 系统提供的支持，我们将不同模块抽象成不同的类，完成类的设计并以类图的形式进行呈现。

在这一步，我们同样进行了需求可追踪性的说明，对不同功能型需求是如何在我们的详细设计中实现的以及不同的非功能性需求是如何被我们的详细设计所支持的进行了详细的说明。

### 3.3.4 实际开发

我们主要依据详细设计说明书进行实际开发。

具体到实施，由于我们采用迭代-增量模型，因而我们首先完成了机器人基础功能（移动、建图）的开发，进而完成了进阶功能（导航、避障、检测、抓取、语音）的开发。

需要说明的是，我们小组的实际开发过程是与软件测试过程同步进行的。

### 3.3.5 软件测试

表 4 测试样例总体说明

测试编号	用例标识	对应需求	测试种类	设计依据	覆盖情况
1	TC0	0.1	黑盒测试	移动程序所有要完成的功能	100%
2	TC0	0.1	白盒测试	移动程序所有分支情况	100%
3	TC1	1.1	黑盒测试	语音程序所要完成的功能	100%
4	TC2	2.1	黑盒测试	建图程序所要完成的功能	/
5	TC3	2.2	黑盒测试	设置航点程序所要完成的功能	100%
6	TC4	2.3/2.4	黑盒测试	导航程序所要完成的功能	100%
7	TC5	2.3/2.4	黑盒测试	多航点巡航功能	/
8	TC6	2.3/2.5	黑盒测试	物品识别程序所要完成的功能	/
9	TC7	5	功能测试	通过总控文件完整运行程序	/
10	TC8	7	功能测试	考察有价值的时间	/
11	TC9	3.2	功能测试	由于机械臂原因，简单输出	/
12	TC10	2.3/2.4/3.1/3.2	集成测试	机器人系统提供的所有服务	100%

测试过程数据看[这里](#)。

a.列出本系统的开发流程；

b.列出需求分析的功能模型和设计说明的体系结构，说明需求可追踪性；

- c.列出单元测试用例设计依据，分别说明基于白盒/黑盒测试的覆盖情况；  
d.针对测试所发现错误及修改过程，列出测试过程数据的存储地址和访问链接。

### 3.4 进度

表 5 进度说明

序号	应完成的工作内容	原计划时间节点	实际完成时间节点	状态	原因
1	完成并提交项目开发计划	2020.03.09	2020.03.09	准时	/
2	配置环境	2020.03.15	2020.03.15	准时	/
3	调研并学习 ROS 有关操作	2020.03.30	2020.03.30	准时	/
4	完成需求分析文档	2020.03.30	2020.03.30	准时	/
5	需求分析评审	2020.04.01	2020.04.01	准时	/
6	实现机器人基本移动	2020.04.07	2020.04.07	准时	/
7	实现机器人 SLAM 实时建图	2020.04.14	2020.04.09	提前	ROS 系统自带建图功能
8	实现机器人路径规划	2020.04.21	2020.04.13	提前	ROS 系统自带路径规划功能
9	实现机器人动态避障与物品抓取	2020.04.28	2020.6.4	推后	1.仿真环境不支持机械臂 2.后期其他课程压力比较大
10	基础功能全部开发完成	2020.05.01	/	改变	第一部分基础功能较早就完成了并完成了代码对接，第二部分（如 9）比较晚才完成
11	代码对接	2020.05.05	/	改变	同上
12	开发增量功能	2020.05.12	/	改变	同上
13	系统调试	2020.05.19	2020.05.19	准时	/

14	准备评审内容	2020.05.20	2020.05.20	准时	/
15	软件测试，编写测试文档	2020.05.26	2020.05.26	准时	/
16	测试文档一次评审	2020.05.27	2020.05.27	准时	/
17	继续进行测试，改进文档不足	2020.06.02	2020.06.02	准时	/
18	测试文档一次评审	2020.06.03	2020.06.03	准时	/
19	项目总结文档提交	原本此任务未知	2020.06.15	改变	/

### 3.5 人员工程量

表 6 原计划分工

开发人员	承担工作(侧重)
王正达	项目经理 文档评审
杨开元	程序开发与架构
彭文鼎	程序开发与架构
尹俊成	系统测试 文档整理

实际分工：无变化

团队合作主要依靠微信群进行，于此同时也在 github 上以看板、issue 的方式进行部分管理。

表 7 人员工程量分配

提交物编号版本	名称	王正达	杨开元	尹俊成	彭文鼎
SDP V3.0	项目开发计划	27%	37%	17%	19%
SRS V3.0	软件需求规格说明	25%	30%	28%	17%
SDD V4.0	软件设计说明	35%	30%	13%	22%
STD V5.0	软件测试说明	30%	32%	26%	12%

PDSR V1.0	项目开发总结 报告	33%	31%	11%	25%
总体工程量分 配	——	30%	32%	19%	19%

## 4 开发工作评价

### 4.1 对开发效率的评价

开发周期：3月初-6月中旬 按三个半月算

程序总量：3687行 平均每人每月 263行

**表 8 文档总量及平均效率**

文档标识	字数
SDP	4465
SRS	4490
SDD	10229
STD	6745
PDSR	7878
总字数	33807
平均每人每月（按三个半月计算）	2414.786

### 4.2 对产品质量的评价

#### a.系统功能性评价及依据

产品的目标是实现一个可以自主移动、主动避障以及对目标物进行检测和抓取的简易机器人。它拥有自主分析地图信息以及在此基础上预先进行路径规划的功能。为改善人机交互体验，可以根据需要加入语音识别与控制系统。经过对功能系统功能测试，

根据这些功能需求，我们针对每一条功能需求进行了细粒度的功能性测试，包括机器人移动、语音交互、自主前往目标、目标物体识别与抓取等，以及整体的系统运行测试。我们的机器人成功完成了以上多项、多次测试，测试结果表明系统功能性质量优秀，符合预期。

#### b.系统非功能性评价及依据。

在功能性需求之外，我们还需要保证机器人能够稳定地完成高质量的工作，即系统的可靠性。此外，系统的可扩展性需求也很重要，机器人系统应能够适应一定程度内系统容量的扩大和管理内容的增加。最后，为了改善用户的体验以及

扩大机器人的适用人群，机器人系统使用起来应该足够简单。

根据这些非功能性需求，我们对系统进行了几项非功能性测试，包括可维护性、可扩展性测试、可靠性测试、易用性测试、及时响应测试。由于疫情原因，我们的测试都是在仿真环境模拟，仿真条件下的测试结果表明，机器人系统具有可靠性、易用性，能够及时响应。

### 4.3 对技术方法的评价

a. 所采用的模型、技术、方法、工具、手段等；

综合考虑本项目的具体任务，由于部分任务之间存在依赖性，我们选择迭代-增量式过程模型，可以提高稳定性和效率。软件开发主要技术：

项目开发采用的管理工具有 **github** 看板、**issue**、里程碑等。

小组保持每周平均一到两次的线上会议，来交流进度、讨论问题、商定计划，保持每位成员掌握当前进度和各自任务。

b. 给出所采用技术方法的应用效果评价：

本次系统开发和测试表明，迭代-增量的过程开发模型适合本项目的开发，我们能够及时改进需求，调整任务和增加功能，最终完成预期系统功能的开发和测试。

项目管理工具帮助我们管理和规划任务，如 **issue** 和自动化配置的看板可以记录任务优先级和分配任务，但有些工具实际没有完全发挥其作用如 **milestone**。

### 4.4 问题原因分析

a. 项目前期文档各部分的撰写风格不统一。这是因为我们的每个文档都是由团队的全体人员共同撰写，没有规定统一的风格。之后为了解决这个问题，我们还是分工写文档，但是让主编文档的同学负责整理文档的风格。这样使得文档整个的撰写风格较为统一。

b. 时间分配不是很合理。项目测试环节前期我们投入的时间不够，导致第一次测试课堂展示时我们还没有来得及做白盒测试。原因主要在于我们的执行力还有所欠缺，没有严格按照之前的项目计划实施；其次，也和仿真环境发布得较晚有关。在此之后，我们还是通过迅速调整，完成了白盒测试。

c. 项目需求有所简化。最终的项目我们去掉了用深度学习方法做物品识别的功能。一方面是开发时间不够，另一方面则是在技术上还有所欠缺，此外，仿真环境下的物品模型和现实情况差异较大，即使实现了实际也参考性较低。

## 4.5 风险管理

- a. 初期预计的风险
  - a) 未如期获得硬件/软件资源
  - b) 需求发生变化
  - c) 低估了软件规模
  - d) 开发过程中技术出现变化
  - e) 管理模式、人员分工发生变化
- b. 实际发生的风险
  - a) 未如期获得硬件/软件资源
  - b) 需求发生变化
  - c) 低估了软件规模
- c. 风险减缓或消除情况
  - a) 未如期获得硬件资源风险，由于课程组后来提供了仿真环境，该风险已经很大程度上得到减缓。但是还是对项目的开发造成了不小的影响。
  - b) 需求发生变化风险，实际的开发和预先制定的计划会有所出入，开发过程中我们尽力地降低需求变化带来的风险。举例来说，由于仿真环境中我们没有实现机械臂，因此最终的抓取部分有所简化。
  - c) 低估了软件规模风险，项目之初我们对 ROS 的了解不够，以为项目代码需要自己从 0 开始搭建。实际已经提供了很多机器人功能的 demo 代码，我们需要的是理解代码的原理，将不同模块组装起来实现我们项目需要的功能。

## 5 缺陷与处理

需求评审阶段：

一开始对于机器人采用哪种运动方式（语音跟随还是键盘操控）犹豫不决，一定程度上也拖累了开发进度，后来根据仿真环境的实际情况确定了键盘操控的方式。

设计评审阶段：

模块类图设计有一些问题，由于当时撰写文档时对 ROS 体系理解有限，对于节点的概念理解不够透彻，一些类图的构建有一些问题。之后在学习了相关知识后，结合节点间通信的知识，重新完善了类图设计。

抓取功能后续发现在仿真环境下无法实现，于是修改了相应设计。

测试阶段：

需要根据系统的需求、功能和规格测试来对系统进行分析和验收，在这个过程中发现系统存在一些之前未检测到的问题，比如在不同环境(虚拟机与双系统)测试可能会出现难以复现的 bug，如更新代码后立即启动 gazebo 会崩溃，而重启后则正常等问题。针对这些问题，我们设计的解决方案是设定开发规范：每次修改代码 push 到 github 代码仓库后都要求写明详细的 commit 信息，在 readme 里写清楚测试的使用说明，包括前置条件和预期结果等，便于快速发现和定位问题；同时在微信群里通知组员，以同步小组进度。这样规范下来，我们的效率有所提高，加快了测试和迭代进度。

## 6 经验与教训

王正达：

良好的架构是成功的一半，一定要在一开始就做好顶层设计，这样才能对之后的工作起到好的指导作用，而不是后面发现了问题再去前面找补。

杨开元：

早点开始，早做准备总是一个好的选择。没有实际动手开发项目就不知道项目开发的难易程度，但是早一点开始可以给自己赢得选择的余地，亦可以给自己留下优化改进的时间。

尹俊成：

团队项目不是个人的单打独斗，大家好才是真的好。整个项目的开发过程中，需要每个人共同参与，讨论研究项目开发的大致方向和实现细节。具体到个人时，需要保证可以高质量的完成自己负责的那部分工作。

彭文鼎：

团队项目开发需要多交流，及时反馈。工欲善其事，必先利其器，尤其是远程工作下，利用好科学方法和工具如看板等可以提高效率。对于长期的项目，需要设立里程碑的基础上快速迭代，敏捷开发就不要畏惧尝试和风险，try fast and fail fast，同时需要持续对需求进行评估和改进。