## “开发软件系统”课程实践的收获和感悟

首先，通过这次项目实践让我完整的经历了一次软件项目的开发，从一开始的需求分析，到后来的概要设计到详细设计，然后再到代码实现和测试，在这个过程中建了很多的模型，画了很多的UML图，写了多份文档，编写了许多的程序代码，开展了多次的讨论和修改，从一开始的毫无头绪到现在对整个流程比较清晰，每次都会想如果做下一次在开发软件的话，肯定不会做得这么手足无措。如果真的有下一次的话，我觉得会比第一次好很多倍。

1. 心得体会

需求分析真的是软件开发过程中最重要的一个环节，也是最难的一个环节，确定一个很好的创新点是一个很难的过程，在这个过程中我们用了各种方法，包括调查问卷，反复研讨，看了许多的软件项目，都没有一个很好的适合我们的创新点，这也是后面我们项目开展阻碍之一。现在想想创新真的来源于生活，只能平时多注意，才能更好地抓住灵感。

在确定好方向之后，就是进行需求分析，这个过程回想起来非常地失败，由于这个过程一开始没搞好导致了后面返工反复修改，总结一下不足之处如下。

第一，调查不足，在确定了要做的项目之后，我们对软件应用场景的认识大多局限于自己的理解，没有开展真正意义上的调查和资料分析，导致软件需求不够明确和全面。

第二，对需求分析后的功能点分析得不够清晰，虽然后面逐渐完善，但是还是吃了这个亏，以后做其他需求分析的时候一定得把功能点分类清楚，如果功能点相互依赖的话也得把依赖关系搞清楚。

只有每一步做好了才能稳扎稳打，否则后面就得花多倍的时间来弥补。下面介绍一下对概要设计和详细设计的一些认识。

在设计阶段，对部署图、顺序图、活动图等等不知改了多少次，主要是因为对UML语言一知半解。后来随着项目的开展才发现设计的重要性，没有设计编码就没有依据，就会很随意，效果可想而知。不仅要认真进行设计，而且设计的过程必须精益求精，设计完成之后不仅要检查设计做的对不对，好不好，而且团队成员必须就设计达成一致，否则不同的设计模型难以统一，这也导致了我们进行多次的设计返工。当然，这里能看出团队内部的交流真的很重要，我们团队遇到的诸多困难和问题，回想起来主要原因还是相互之间的交流和讨论不够，组织协调不足。

软件设计是用于指导软件实现的，不能只顾着设计而其他不管，经历了几次设计失败（一开始自己通过自己的经验感觉可以实现，后来根据开源社区上提供的各种资料发现，几乎不能实现），我认为在整个软件设计过程中，一定要充分地掌握相关的遗留系统、开源软件等软件资源，如果不事先搞清楚这些东西，就会发现所设计的东西要么环境不支持，要么根本凭团队的力量难以实现。

事实上，做完整个软件开发项目之后，我充分认识到开源社区的重要性，开源社区资源的重要性。通过开源社区学习软件开发技能很重要，利用开源软件资源很重要。通过CSDN学会了android开发技能，学会了ros系统的一些基本知识，还有一些系统运行的知识等等；通过GitHub查询开源软件，通过Stack Overflow解决了众多的android studio的配置问题，还有各种环境问题。总而言之，互联网上的开源软件资源对软件开发而言真的极其重要。

在编程阶段的体会其实是最多的，通过开源的阅读，不仅仅了解了开源怎么用，还学会了怎么更高效率、高质量的编写代码，以及通过对别人代码的学习，对自己编程行为有很大帮助。在实际编程时，基于对代码质量的要求，对程序代码进行不断的修改和调整，纠正了我原有的许多编程陋习。虽然Debug的过程真的很煎熬但是收获颇丰，遇到了许多以前不曾遇到的各种错误（如环境、代码逻辑、兼容性等等），让我学到很多。

最后是对代码质量的体会，代码质量不好真的是硬伤，读开源软件代码的时候，有的开源真的是太难以理解了，而且兼容性、可移植性太差，这学期的软件工程课程学完了之后，再结合课程实践，突然发现自己的代码风格有所改进，但是代码的可维护性等质量问题还是存在，而且代码质量跟踪做得不够，所以后面修改的次数也很多。总之，从分析和设计就应该注意质量问题，整个工程开发过程中还得跟踪质量问题。

1. 困难困惑
* 软件开发经验不足

这是一个硬伤，一开始的时候，从软件的整体框架结构，到编程语言，甚至建模语言都很模糊，这导致了概要设计和详细设计的时候出现了许多问题，如建模符号使用不当、体系结构设计的不清楚。在建模过程中，将自己的设计结果用UML表达出来很费劲，出现了许多问题，包括设计的内容在模型中没有得到体现，UML模型图画的不正确等等。这些都是由于自己第一次使用UML建模、缺乏相关的开发经验造成的。除了设计方面的经验不足，在建模和程序设计语言的使用，开发工具和环境的使用等方面也存在经验不足的问题。第一次用android studio写程序写界面、第一次用面向对象程序设计语言Java来编写代码，都出现了许多现在想来不该出现的问题。

* 资料收集不够

这一点真的是深有体会，因为实践所涉及的Android开发平台，ROS操作系统，Turtlebot机器人等自己之前都没有接触过，也不了解，但是一开始只在大致解一下的基础上就开始进行软件设计和实现，导致后来发现整个设计思路存在诸多的问题，甚至不可能完成。在编程和实现阶段也是如此，资料收集出了问题导致反复更换实现方法。

* 分工配合不够

该方面主要体现在一致性和积极性方面，一致性方面最主要是沟通不够及时，团队成员之间没有交流到位，会出现两个人同时干一件重复的工作，或者两个人实现的模块的接口完全不搭配。这一点其实完全可以内部多协调就可以解决好。其次是积极性方面，有的时候时间不够，或者有的人打酱油就会把其他人的积极性给降低，导致整体工作效率不高。

* 目的不够明确

在开发项目的过程中有时候会偏离自己实现的功能，或者就没有去考虑软件需求，有时候某些功能实现不了只能换其他方法来实现，导致这些状况的主要原因是没有围绕软件需求来开展设计和实现。

概括起来，实践中遇到的最大的困难来自于缺乏工程开发的经验，对开发和运行环境不熟导致如Android，ROS等环境配置了好久，对开发的流程不熟悉导致所有一些看起来很简单的问题却一犯再犯。

1. 收获和认识
* 熟悉了各种开源社区（如GitHub、CSDN、Stack Overflow等），同时也掌握了如何借助开源社区来解决实践过程中遇到的困难和问题。
* 大致掌握了软件开发的整体过程，感觉有了一个总体的思路和映像，知道如何展开，知道应该避免哪些问题，也对下一次开发更有信心。
* 学会了如何在实践中进行分工合作，虽然整个合作过程失败了很多次，但是失败的同时也给了我们很多的经验和教训，开发是团队的事情，每个人都需要参与进来才能完成好软件系统的开发。
* 提高了软件开发和运行环境配置的能力，对Android开发环境、ROS环境有了更深入的认识，知道如何根据问题找解决方案，哪些问题可能是由哪些原因造成的，其实通过对环境配置的了解，对系统的运行也有了更深的掌握。
* 编程能力确实提升了，更多的是工程开发的能力提升了，会思考如何让一个代码简单易懂可维护，同时对整个项目的结构组织运行有了更深的了解。
* 开源社区很重要，它不仅仅是一个获取群智资源的平台，更是一个开展交流以分享经验、解决问题、开展合作的平台，通过各种开源社区，能够获取软件开发经验，帮助解决开发问题。
* 对程序代码进行管理其实很重要，虽然还不够熟悉，但是基本掌握了对项目进行管理和控制的基本技能。
* 通过实践，真正明白了文档、程序、数据三者之间的关系，同时也才明白了质量、分析、设计、测试等的作用和重要性。