# 高等软件工程总结

经过一学期的学习，我在高等软件工程这门课上，收获颇丰，总结起来主要有如下三方面：

第一，是知识上的拓展。在这门课上，我真正了解到了一个软件的完整的生命周期，从需求的提出，领域背景的分析，需求文档的撰写，到设计、实现、测试，最终的维护，由于本科是其他专业，再加之以往写的程序太小，虽然意识到大型软件开发绝非以往写的一些小程序能比，可是，真正学了这一门课后，我才发现，软件开发的方法学，是一门极其重要的学问。而就我目前的理解，大型软件开发难处在于“描述”，保持一致性。从需求阶段来说，来自各行各业人士对需求的“描述”，需要被专业软件开发人员所能完完全全的理解，否则，双方之间的理解偏差越大，越到后期发现，所需要的改动也越大。从软件设计角度来讲，对软件设计的“描述”必须易于理解，能够被开发团队中的每个成员熟知，否则，前期精心构建的软件架构，在后期阅读时，出现理解偏差或是捕获不到之前设计的细节，轻者导致软件质量的下降，重者导致项目的失败。正是基于“描述”的需要，在高等软件工程这一门课上，我们学习了各种阶段的各种各样的建模方法，才能够高效地分析软件需求，进行软件系统的分析设计。我们所学习的主要的建模方法是UML，后面也学习了RUCM、OCL等建模方法，但UML建模贯穿始终，在项目的各个阶段都有应用。UML中常用的模型有用例图、类图、活动图、时序图和状态图，这些模型通过从不同的视角来展现一个软件的结构和功能实现，让我从另一个角度理解了软件开发不仅仅是写代码，写逻辑，更是写设计，写结构。

第二，将理论应用到实践的能力大大提升。老师做点评时，我记得最清晰的一句话是‘Run the example’。光是课堂上，老师经常会给我们找小例子让我们练习，确保我们能够理解之前所讲的内容。不光如此，由于我们整个课程是以项目为中心，每一节课的课堂学习结束后，都必须应用老师课堂上所传授的知识，来完成当前的工作。不过可惜的是，课堂上老师所选的例子大多十分有代表性，所建立的模型十分精妙，尽管每次练习都思考了许久，但是最终建立的模型与老师所建立的模型有很大的差距，而且我们组的项目由于项目本身原因，建模并不是很复杂，所以总感觉自己目前的各种建模方法的掌握比较有限，掌握的规范更多一些，凭借目前的能力尚不足以建立较为复杂的模型，日后，仍应当找机会，好好锻炼自己建模的能力。

第三，本次课程中，收获最大的是项目的经历。总的来说，这次项目的进展，可以说充满了艰难与曲折。首先是对需求的理解。在项目的初期，我们小组花费了大量的时间来讨论这个系统的功能需求，例如对于项目的输入——“给定的自然语言文档”，对这个文档的涉及到的领域、应当包含哪些信息、以什么样的格式进行组织等等问题，我们都需要进行深入分析、自己定义，对于项目的输出——“结构化需求文档”，我们也是搜集了网上的各种需求文档的资料，尝试着自己去进行定义这个文档，知道中期才明确了输入是GWT文档、输出是RUCM。其次，是软件的设计。项目要求是用自然语言处理的方式进行转化，然而我们小组没有一位成员的课题组研究方向涉及到自然语言处理，对于这方面的知识，大家都要现学现卖，以至于最终的实现部分，我们用自然语言处理所作的工作有限。软件设计的最大的问题，在于大家不同想法的碰撞。每个人知识背景、思考角度不同，所提出的设计也大不相同，有的能够相互融汇结合，而有的就是冲突对立。 这种情况在团队协作中是很普遍的现象，从一开始大家不太善于处理这种分歧，经常因为这种分歧耗费大量的时间，到后来，随着团队的进一步磨合，成员之间的沟通效率的提升，对不同的意见，经讨论后或舍弃或保留，都是有理有据令人信服，最终的目标都是更好、更高效地完成这个项目，有的想法尽管很好，但是受制于现实因素，就只能作为会议里的一条记录。最后，很感谢团队中的每一位成员，尽管遇到了诸多困难，但是大家内部十分团结，每一位成员都为这个项目投入尽可能多的经历；每次评审后都需要大量修改时，团队内部从未出现过指责之声；分配任务时也都是根据大家各自的时间富余程度、个人擅长方面进行协商分配，从来没有人因为分配不公而抱怨过；感谢团队中的每一位成员，贡献出自己的努力，没有让这个项目fail掉。

最后的最后，感谢吴际老师，能够以这样的一种方式来开展高等软件工程的教学工作，认真审阅我们的项目，仔细指出项目目前的不足之处，让高等软件工程这门课充实而又意义！