

软件工程实验总结

——从选题到结束

其实一开始让我选软件工程实验这门课的时候，我是拒绝的。因为在我的印象中，这是一门枯燥无味，虚度光阴的课，但是不得已，这是专业必修课，必须选，所以也就硬着头皮来上了。

但是随着课程的进行和实验的深入，我渐渐意识到我错了，这门课并不像我所想的那样学不到东西，并不像我所想的那样乏味，反而很有趣很长见识。口说无凭，下面我就从项目的选题到项目的结束，来说说我对软件工程实验的感悟和看法。

选题。由于老师要求是对一个开源软件进行需求分析，而我们组成员没有一个看过开源项目，因此刚听到这个要求的时候着实让人眉头紧皱。但是冷静下来分析以后，我们决定每个人先去调研一个项目，了解项目的背景，大致的功能以及粗略的代码行数。在每个成员对各自调研的项目有了一定的了解以后，我们立刻组织了一次研讨会，分析各个项目的利弊，最终确定了 **redis** 为我们的选题。整个过程干净利落，没有浪费一点时间，讨论的结果也令人愉快（因为 **redis** 是我调研的）。

在选题过程中，我就开始意识到团队的力量是无穷的。如果让我一个人去调研 4 个开源项目，我肯定会坚持不下去，就算坚持下去了，一个人对每一个软件的看法和观点总是单一和存在偏见的，无法客观公正地对比每一个软件利弊。而团队分工合作，各自调研各自的软件，最后开会讨论，各抒己见。这个模式让每个人都可以尽情描述自己调研的软件，让大家都各个软件的了解更深入，从而更好地确定选题。

需求分析。需求分析的过程是最煎熬的。大家对 **redis** 软件都只是有个初步认识，该软件的模块和功能都补肾了解。这时我们又采取了个果断和有效的决策：每个成员分模块学习 **redis**。这个分工合作方法有一次奏效，我们通过查阅资料和书籍，对各自的模块进行深入的了解，并通过开会交流各自模块的功能，接着进行需求建模。然而，四个小组成员中有三个都是第一次接触 **RUCM** 建模和使用 **RUCM** 软件。一切都是新的，又得从头学过。所幸，还有一人对 **RUCM** 比较了解。通过他共享的资料，我们很快入门。

在需求分析阶段，首先学到的就是 **RUCM** 建模，它可以很清楚的展现出需求分析的建模。学习了如何对一个软件进行需求分析，包括知道了怎么画用例图，怎么写测试用例等。在需求分析阶段过后，我对我们的小组越来越自信，每个人都有很强的学习能力和沟通能力。

实验 6-8。实验 6-8 包括软件进度计划与控制，工作量估计以及变更控制和版本管理。其实，变更控制和版本管理是贯穿实验的整个过程的，在一开始，老师就推荐我们用 **github** 作为变更控制工具。由于我之前对 **git** 的操作就比较熟悉，于是我就先负责将我们组的文档上传和同步到 **github** 上。后来发现这并没有达到变更控制的效果，因为如果由我一个人上传，就无法记录其他同学的变更历史。因此就转换为每个小组成员各自上传自己的实验输出到 **github**。工作量估计由组长全权负责，软件进度计划由另一小组成员负责，这里就不细说。

我觉得实验 6-8 是一个贯穿于整个实验过程的实验，作为一个单独的实验 6-8 有点不太合适。因为软件进度的计划，版本管理和变更控制，都是需要在每一个实验中都有记录的，实验 6-8 只是应该将这些结果在一个合适的时间进行检查和汇报分析。

总之，软件工程实验给我的感觉就是，从痛苦到收获。痛苦在于，对于软件项目的不熟悉和各种工具方法的不熟悉，所有东西都得从头学过。收获在于，正是这一切的不熟悉，给自己带来了能力的提升和团队的提升。

