基于Scrapy的WebUI开发

——小组总结

SY1906514 汪凌风 SY1906508 明　昊

SY1906513 汪丽萍 SY1906431 郑泽西

SY1806220 闫奕涛 SY1906509 邵志钧

2020/06/06

# 版本记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 日期 | 修改说明 | 参与者 | 审核人 |
| 1.0.0 | 2020.06.06 | 完成第一版 | 全体成员 | 汪凌风 |

注：全体成员即汪凌风，邵志钧，明昊，汪丽萍，闫奕涛，郑泽西

目　录

[一、组员心得与建议 4](#_Toc42348649)

[1.1 邵志钧 4](#_Toc42348650)

[1.2 汪凌风 4](#_Toc42348651)

[1.3 明昊 4](#_Toc42348652)

[1.4 汪丽萍 4](#_Toc42348653)

[1.5 郑泽西 4](#_Toc42348654)

[1.6 闫奕涛 4](#_Toc42348655)

[二、有效方法 5](#_Toc42348656)

# 一、组员心得与建议

## 1.1 邵志钧

本学期的软件工程综合实验总的来说有很大的收获。因为上个学期选择了高等软件工程课程，通过高软的课程项目训练了软件过程中在领域、需求和设计阶段的建模和分析能力。软件工程综合实验课程则延续了高软的内容，并升级了课程项目，将课程项目扩展到整个软件过程，包括软件实现阶段、测试阶段以及软件过程中的文档评审。课程项目相比起曾经接触过的课程项目来说，更加贴近于生产环境，由于整个项目的工作量比较大，同时因为疫情的缘故对团队远程协作效率的要求又很高，因此整个项目下来，对软件过程中各项工作具体如何开展，相比之前只在软件工程课程中接触到的软件过程，本次课程中有了更加深刻的理解。同时课程中还引入了Github、MS Project等内容，学习到了很多新的技能

在小组中除了参与到了实验1-5中，还主要负责了项目的配置管理工作，更深刻的对于一个项目来说软件版本迭代和各版本的追根溯源至关重要，配置管理工作要求非常精细，而且从版本变更控制到配置库协同都有着严格的规范，才能保证在多人协作的项目中每一个文档都能被追根溯源，从而最大化小组成员的协作效率。

对于本次课程，个人来说非常满意且有着很大的收获。但是唯一的建议就是，因为软件过程中各个阶段的工期长短决定了工作的安排，应该是一个计划性非常强的过程，但是本学期可能因为疫情只能远程完成课程的缘故，软件测试、软件实现等阶段的时间安排非常的不固定，从而导致整体的时间安排被打乱，这样安排某种程度上容易影响到软件制品的质量。希望在之后的软件工程综合实验课程中可以在学期初提前对课程的时间安排、每一节课的汇报内容、每个阶段的产出物有较为明确的计划，从而各小组也可以更加明确各自的工作计划，在各个阶段也会有更加精彩的表现。

## 1.2 汪凌风

　　上学期上的高等软件工程并没有代码实现环节，更像是一种模拟开发。这也是我第一次参与这种群体开发项目，并走完完整流程，可以说学到了很多东西，至少“开发”这个词不再是只凭想象的东西，而是落到实处去了，以后进行类似的工作都能够有章可循。既然是第一次，总会有一些问题，我认为最首要的问题就是需求分析的方法，在需求分析时，很多时候我们考虑的其实是自己要实现什么功能，而不是用户或者市场需要什么功能。当然这也可以归结于人员、时间和能力所限，后期设计实现时，我也发现一些需求其实无关紧要或者与整体产品不一致。而且设计时间时间较短，开发有些粗糙，导致软件制品和测试容易出现问题，所以具体代码开发也可以有一些提前量。

## 1.3 明昊

1. 组队的时候需要认真了解队友擅长的编程语言和框架，在开发的时候选择适合大部分人的项目，以免开发对队员造成过大的压力

2. 在开发阶段如果条件允许可以组织小组对项目技术进行一次集体学习，这样能节省很多时间

3. 在项目需求阶段，要充分调研功能的实现难度，如果对功能的实现难度不清楚，原则上尽量少加这种需求。

4. 软件开发阶段尽量选择文档齐全的轮子，碰到实现的不是很好的轮子及时断腕。

5. 在进行项目开发进度追踪的时候，最好面对面展示代码开发进度，而非口头说明

6. 对于项目开发时间段要明确。

7. 自由化意味着大量的额外处理逻辑与bug，原则上建议不要写自由度很高的功能，越定制化开发越容易，bug越少。

## 1.4 汪丽萍

软件工程综合课程实验旨在通过完整的参与项目，体会软件开发的完整流程，提升小组成员的协调沟通能力、软件开发管理能力、问题分析能力并能够明确自己的特长和定位。下面是我在基于Scrapy的WebUI开发的各个实验阶段的心得和建议。

1. 项目准备阶段

项目准备阶段涉及到项目的选择，自由程度很高，选择范围很广，这既是一个机会也是挑战，机会在于可以选择大多数组员都比较熟悉的项目，为接下来的项目推进奠定一个比较好的基础，挑战在于小组内每个成员的背景不同，很难保证所有同学擅长的领域都是相同的，以及在大家都不熟悉的阶段，项目的确定就比较耗费时间了。

1. 软件需求阶段

在这个阶段我们认真考虑了系统的各个使用场景，从相关的用户群体出发设计需求，但是对于需求的可行性分析不够直接导致后续设计阶段有些功能在较短时间内不能按照需求的要求实现，只能简化，且导致需求有多次迭代。因此需求阶段的可行性分析是十分重要的。

1. 评审阶段

在需求评审阶段阶段，通过对别组的评审，看到了别的小组在需求分析的优点，也通过发现别的小组需求中的问题让自己对如何进行软件需求分析有了更深的了解。

在测试评审阶段通过对别组软件制品和测试用例的评审，对如何通过测试手段保障软件质量有更深入的了解。

1. 软件设计与实现阶段

在软件设计与实现阶段，对由于项目设计到的框架和技术都是以前不曾接触过的，通过此次的软件实现学习了很多新的知识。这个阶段对平时开发经验不多的同学是一个很好的锻炼机会，但是这期间绕的弯路也更加证明了软件需求分析中对需求可行性分析的重要性。

1. 测试阶段

在测试阶段测试需求分析、需求分析文档和设计文档要有一一对应的关系，且测试用例的设计要尽可能的全面充分，这样才能使测试成为保证软件的质量行之有效的手段。

1. 项目的计划

在每周的结束都应该对下周的工作进行细化的安排。在使用MS Project进行项目计划管理时，一开始由于不熟悉该工具，当计划调整重新设置基线时总是将已经完成的任务的基线也更改掉，耗费了很多时间在纠正错误上。

由于五一假期的原因，软件设计与实现的工作以及软件测试和软件测试评审的工作没有在周次结束后自动结束，而是交织在一起，在实现阶段也进行着测试用例设计的工作，在测试评审的前半周进行的是软件测试阶段的工作。建议以后课程实验的周次安排能够在课程的初始就考虑到放假的因素。

## 1.5 郑泽西

本次在实验过程中，我和项目组的成员们一起完成了软件工程开发所要求的开发过程，在目前位置完整的进行了软件项目计划阶段、软件需求分析阶段、软件需求评审阶段、软件测试阶段、软件测试评审阶段。除此之外，又在开发过程中和项目组的成员一起通过软件进度计划与控制、工作量估计与统计分析、配置管理这些工作对整个项目开发进行管理和控制，在保证项目按照计划执行的同时，还保留了项目执行和开发过程中的各项证据，进一步形成了证据链，使得项目的开发与执行更具有说服力。虽然对于我承担的有能力实现的任务，能尽力保质保量完成，但是因为之前对软件工程这门课程没有太多了解，同时因为个人的能力问题，所以很多阶段的工作没有完成得比较好，比如一开始对RUCM图，类图等图例不够了解所以绘制得不够精确，比如开发阶段因为没有对开发周期有一个清楚的规划，致使后期出现比较严重的问题，不得已削减了功能，一些阶段没有主动承担一些任务，所以对于我对于我个人在本次实验中的工作是不太满意的。对于这次的问题，其实是有办法解决的，比如提前了解实验流程，预习相关知识，合理规划开发计划，针对软件开发中出现的问题及时求教，都有可能避免这些问题，这些教训值得吸收。

通过和项目组成员一起参加了本次软件工程综合实验过程，我最大的体会是代码开发只是软件工程的一部分,其他部分同样重要。从工作量统计来看,开发计划只占到了一部分,其他的诸如需求分析、软件测试、文档编写、进度控制等,在软件工程过程中的工作不容小觑,同样重要, 每个阶段的制品都会影响下几个阶段制品的质量，甚至影响整个软件开发的流程，那些撰写的文档并不是可有可无的部分，他们对软件的开发与完善有着至关重要的影响，其他方面的不完善会影响实际项目代码质量与进度完成情况。同时我认识到团队的合作关系是软件开发的成败与否的关键性因素，组员间的默契关系和工作的合理分配使得项目组的成员工作方面各尽所能,大大提高了工作效率与项目进度，同时项目组的成员之间能够相互学习，在推动软件开发进度的同时共同推动整个项目组的进步。

同时针对该课程的改进，我也有自己的一些想法和建议，该课程理论和实践相互结合，让学生能够更加地了解软件工程过程中的各个阶段，并且亲身的体验，非常有意义，但是有些同学可能没有这方面的基础，所以建议老师可以在课堂上多讲一些相关的内容，同时更清楚地布置需要课下做的完成的工作，可能能帮助我们更加地了解软件工程过程中的各个阶段。

## 1.6 闫奕涛

* 工作介绍

　　在本课程中，除了有专人负责的实验6与实验7外，我参与了课程的所有内容。实验内的工作任务包括参与了实验1到实验5所有文档的撰写，实验8文档的撰写，与WebUI词云相关的开发与文档工作，参与评审与评审意见的修改；实验外的内容包括所有文档的汇总、更新与格式，参与了每一次会议讨论，负责了两次PPT汇报任务。

* 工作收效

　　实验内，顺利完成了所有与WebUI词云有关的文档撰写，也顺利完成了实验8文档的一半内容；顺利完成了词云的开发工作并部署成功，尽管有些许功能没有按照心意实现；评审过程中，也按时按量地对被评审组进行评审，对被评审组的修改进行审核，根据评审建议进行修改，也通过了评审组的审核。

　　实验外，尽管在汇总文档时出现过一次差错，但也算顺利完成了所有文档的汇总工作；在会议中较为积极地提出自己的见解，在理解问题的同时引导会议的进程；PPT汇报中没有出现任何差错，完整且详尽地汇报了一周的任务完成情况。

* 心得体会

通过本次课程，我体会到了从立项开始到项目结束这一软件生命的全部过程。

从需求到实现环节，我发现了在需求提出的过程中，脑子里大部分天马行空的想法是无法落地的，一方面是技术的困难，另一方面是多人工作时协调的困难；我体会到了需求文档的重要性，在开发时要不断地核实自己是否偏离了需求规格说明书，是否与其他人产生了冲突；我理解到了代码开发虽然重要，但也只是软件生命过程中的一环，甚至不是占比最大的一环。

在测试过程中，我体会到了测试工作的冗杂和无趣，测试工作成为了开发过程中最复杂的部分，但又不能逃避；同时我也明白了多人合作的协调工作也是很麻烦的；其他评审组发现的问题确实很棘手，有时甚至是无法修改的，只能选择妥协。

除此之外，我第一次感受到了实验任务追踪、版本管理、工作量统计等任务的复杂，要提前进行若干规划，要进行每周的汇报，每周都要更新文档内容，如果计划与现实不符合要进行解释，统计工作是非常麻烦的。

* 建议

在每次课程结束后，我们小组讨论下一周的任务时，经常会问这周到底要做什么，做成什么样子，要么是翻出往届文档，要么是询问老师，这样是很不方便的。希望老师可以在每次课程结束的时候，明确说明下周汇报时要哪些文档，要汇报哪些内容，文档中需要哪些具体的内容，每次在具体要完成的任务上纠结真的很烦躁。

另外在项目代码实现阶段，老师公布的时间线与实际的时间线有严重的冲突，虽然表面上还是要努力完成任务，但这种时间线的变动对于代码工作而言影响过大，严重影响了我们项目的进度安排。

# 二、有效方法

|  |  |  |  |
| --- | --- | --- | --- |
| 实验 | 遇到的问题 | 解决方法 | 结果与体会 |
| 项目计划书 | 在计划阶段的冲劲儿总是美好的，没有能够深入理解 | 在跌跌撞撞的理解中，开始汇报PPT和完成需求说明 | 计划阶段的一些想法可以在后期改变，不然过于美好 |
| 如何确定实验的项目 | 针对项目组成员的熟悉语言和工具，调研一些流行工具的改进难度，综合分析确定 |  |
| 项目需求分析 | 使用微信群讨论问题的时候，无法及时的做出回应 | 每周预定时间集中开会讨论解决问题 | 非常有效，基本解决问题 |
| 需求分析中RUCM和UML模型经常会遇到需要改动的情况，但按照用例给每个人分工，可能会出现模型不一致或部分遗失的情况 | 专人负责整理和管理项目的RUCM和UML模型，从而保证模型的一致性和模型的版本按规范迭代，实现模型可追溯 | 多人协作项目中，就算有非常完善的规则，也有可能会因为每个人对规则的理解不同，而导致产出物相去甚远，因此需要专人来对容易出现不一致情况的地方来进行把控 |
| 对于需求分析的对象是Scrapy框架还是WebUI没有明晰 | 小组成员开会讨论发现由于该WebUI的开发是基于Scrapy框架的，因此在需求分析中对Scrapy框架和WebUI功能都进行了分析 | 完成了需求分析 |
| 很多时候我们考虑的其实不是用户或者市场的需要，而是自己想（能）实现什么功能，导致后期实现了了一些无关紧要的东西，或者说，与整体产品理念不一致的东西。 | 没有在本实验解决。 | 最终实现了一些与整体不一致的需求，这是以后可以改进的地方。以后可以亲自使用和调研同类软件，从自己的感受出发，探索要实现的软件可以有哪些需求。 |
| 如何确定软件的需求 | 调研相关资料，综合一些比较针对该项目比较常见和流行的功能，并作出适当的增删最终确定。 |  |
| 软件产品改进与展示 | 软件实现初期，对自己能实现的能力过于自信，一直未能彻底实现 | 采取了一些折中的方法实现了基本需求，但不够完善 | 对于任何需求都要第一时间尝试做，一定会遇到很多困难 |
| 软件测试 | 只设计了黑盒测试用例 | 参考系统设计文档，以及老师给的软件测试的资料，发现可以对脚本管理和自动生成脚本设计单元测试用例 | 改进了测试需求规格说明书 |
|  |  |  |
| 软件测试评审 | 找不到别的组现有的测试用例的问题 | 换个角度思考，发现针对整个系统现有测试用例并不充分，有可以增加的测试用例 | 测试用例是为保障软件质量而设计的，因此可以从现有测试用例能否保障软件质量来发现测试问题 |
| 由于对对方使用工具不熟悉导致评审工作不顺利 | 查阅相关资料，并于对方积极交流，在熟悉对方实用工具之后做针对性的评审 |  |
| 软件进度计划与控制 | 由于对所用工具不熟悉导致进度计划与控制不顺利 | 查找资料 | 经过几周的实践后弄懂 |
| 软件实现阶段时间有些仓促，导致很多开发工作有些粗糙。 | 设计和测试都出现了一些困难，走了一些弯路。 | 可以参考敏捷开发的思想，写代码的工作也可以在需求确定后直接进行，有一些提前量。 |
| 配置管理 | Github仓库中包括源码，目录结构复杂且冗杂 | 分离文档仓库和代码仓库，文档仓库用于配置管理，代码仓库用于开发 | 在开发和配置管理过程中，应尽可能将无关的内容分离，从而提高效率 |
| 文档仓库目录结构复杂，多人协作容易导致目录结构混乱 | 通过保护master分支，并建立多个个人分支，通过pull request和review来保证合并到master分支的内容符合要求 | 因为每个人的想法和理解不同，因此会存在偏差，所以需要专门的审核来保证内容的正确性，从而避免更多不必要的返工 |
| Github的更改监测机制无法检测word、excel等格式文档的修改内容，难以通过commit来统计个人工作量 | 将个人工作统计放在文档编写的文字、图表上进行统计，可以更加精确地计算出个人工作量 | Git的Commit更多是作为一个参考，具体的分数计算方案应结合本组实际来设计，从而保证组内的公平性 |
| 工作量估计与统计分析 | 开会讨论的工作量为平均分配，但是得分计算最后高低分差距明显 | 加入实验外工作积分项，并且提高实际用时权重 | 基本解决问题 |