



中山大學  
SUN YAT-SEN UNIVERSITY

# 软件过程改进 课程报告

—— 晚上班第四组

组员： 吴嘉华、魏婉婷、戴旋、黄雄镖、郑越升、周腾腾、关宏朗  
2015 年 6 月 20 日

## 一、项目详情

### 1. 项目名称

中大 Team

### 2. 项目仓库

<https://github.com/jayjiahua/sysuteam>

### 3. 项目完成情况

完成基本功能，进阶功能有待开发。

- 1) 完成用户登录、用户注册功能
- 2) 完成个人队伍创建、加入、删除功能
- 3) 完成大型活动创建，删除功能
- 4) 完成大型活动的创建队伍，加入队伍功能。

### 4. 项目选用框架

后端：Mysql, Expressjs (使用 livescript 编写)

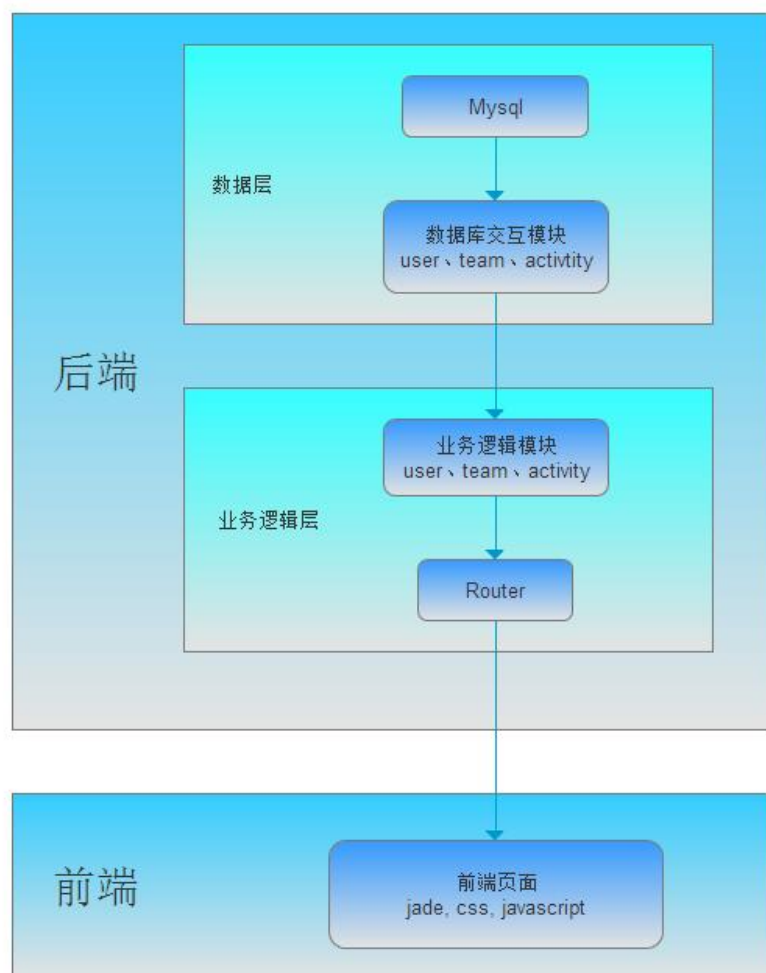
前端：Flat-ui, bootstrap

Jade

Css

Livescript

### 5. 项目整体结构图



## 二、小组成员贡献

### 吴嘉华:

1. 框架的选取
2. 对组员进行分工
3. 协调前后端工作
4. 数据库交互模块的编写
5. 后台业务逻辑部分代码的编写
6. 部分前后端对接

### 黄雄镖:

1. 前后端开发（主要为用户功能部分）
2. 网站原型的设计
3. 数据库基本交互方法编写
4. 分离业务逻辑与数据交互
5. node-mysql 文档的翻译
6. 编写了前后端交互 api 文档等。

### 关宏朗:

1. 负责“用户”、“队伍”模块的业务逻辑
2. 各种路由的布置
3. 原型图的设计

### 戴旋:

1. 负责前端部分
2. 参与了数据库设计，交互设计等部分
3. 重构了前端的部分代码
4. 完成和后台的交互

### 魏婉婷:

1. 前期贡献了方向灵感和撰写了主要的设计文档，在第一次宣讲过后重新调整了思路，从用户角度出发进行了大幅度的修改和目标明确
2. 后期参加了前后端 API 交互接口的构思撰写
3. 用户体验报告的撰写

### 郑越升:

1. 前端大部分代码的编写
2. 对前端后台上数据的交流上提出建议

### 周腾腾:

1. 负责数据库的设计与修改
2. 定义了一些开发细则

（注：排名不分先后，以上只是给出了文字叙述，有关代码贡献的量化分析还请查看 [github](#)）

### 三、心得体会

#### 黄雄镖:

通过这门课程学到了很多东西,无论是前端还是后台,接触到很多之前没用过的新东西如 express, meteor, coffescript, livescript, jade, sass 等等。对 JS 有了更深入的认识,掌握了其中更为深入的东西。通过完成一个完整的项目,对软件过程的改进有了基本的认识。大致了解到了软件生产流程和其中可能出现的问题以及应对方法等。这次的课程,让我对协同开发有了更深的体会,因为之前没有过这样几个人一起做个东西出来。其实我觉得对于一个团队来说,每个人都是很重要的,每个人都有自己擅长的方面,让每个人发挥自己所长是很好的,也能够互相学习。但是如果分配的任务有人没做好的话就会拖了整个团队的进度,最后没办法也只能有人来多做一些。我觉得在团队开发中,有个比较重要的问题就是整体的代码风格等,因为这直接决定了维护这个项目的代价,如果代码风格参差不齐的话维护起来就会很麻烦。我们通过先约定好了前后端接口规范和代码编写规范来。总的来说,通过这门课程学到了很多东西,前端和后台方面的技能的都收获不少,也能够熟悉地使用 github 进行团队开发工作了。

#### 关宏朗:

##### 团队项目总结:

自五月份起,课程任务从个人项目变成了团队项目,虽然时间没之前的紧迫,但是团队项目更需要一段长时间去明确需求,定位开发方向,编辑页面逻辑,分析用户事件等开发前需要详细思考好的问题。一次又一次的阶段总结让我们的前进方向逐渐清晰,老师及其他同学的意见和建议让我们少走了不少弯路。

项目预备阶段,大家一起详细讨论了用户角色、用户关系、用户需求的定位,各自发表了相关的意见,以及 APP 名称的确立,大家都相当踊跃。在分析页面逻辑及布局时,我和雄镖利用 Axure 软件做出了所有前端页面的交互模拟,具体化了界面布局,让队友们直观的了解到前端的大致逻辑,方便了后续开发。经过周腾腾及组长等队友的数据库布置后 (SQL),我们终于进入了代码开发阶段。在开发初期,我被安排到了后台组,与组长负责'用户'、'队伍'模块的业务逻辑,以及各种路由的布置。前期阶段由于我的电脑系统与 SQL 出现了不兼容的问题,因此组长完成了绝大部分的工作,分配了一些优化思路及交互注意事项让我思考。

在开发过程中我整理了小组们在 github 项目上的每一次改进的记录,提出的问题,作出的改动等,方便后续开发中的借鉴。在开发后期,我被安排到了前端组开发,协助队友们进行前端页面的增添及改进。到 16 周中期,我们的 APP 终于初步开发完成。

最后感激我的团队,每一位都认真负责,能力突出,给予我百般帮助与支持。

##### 软件过程改进课程总结:

久仰王青老师大名,以及传闻中的作业难度,一开始给了我极其震撼的课程压力。与之前的 WEB 2.0 不是同一维度的存在。环形菜单唤醒了我的 CSS 记忆,表单排序巩固了我的 JS 能力,异步调用开阔了我的编程视野, TOP 思想改变了我的编程习惯, express JS, meteor JS 建立了我的框架思想,整个学习过程下来,受益匪浅。

何为软件过程改进,以下是我的个人理解:

1. 接受了一个任务,先运用自己所学的知识,慢慢地摸索去做。
2. 在做的过程中,遇到了什么问题,自己怎样去解决的,一一记录下来。
3. 每做完一 part,总结一下优点与不足,自我梳理,然后再做下一 part。

4. 姑且封闭地完成这个任务，最后自我回顾。

以上是软件过程。

5. 重新看这个任务的需求，跟他人交流，查阅相关资料，再找一个更适合的方法，重复以上步骤再次完成这个任务。
6. 在这个过程中不断学习没有接触过的技术，改进需要改进的地方。
7. 最终，找到合适的方法，合适的编程习惯。

以上是改进过程。

我认为，这一个学期的学习，技术是其次，关键是方法，老师的教学目的就是让我们找到一套自己的 软件过程-软件改进 的编程办法，使我们在日后的工作学习当中，出类拔萃。最后，感激老师一个学期以来的辛勤劳动，祝工作顺利。

### 戴旋：

一个学期下来，在这门课程中花的时间很多，几乎每周的作业分量都很足，需要去学习很多东西才能动笔，动笔之后又要经过一番思考才能完成（或者不能完成）这份作业。也正因为花的时间很多，也确实学到了很多。譬如 JavaScript 的原型、闭包、作用域、this 等，也在老师的指导下接触了许多 web 领域先进的技术，如 Jade、Less/Sass、CoffeeScript/LiveScript、Node.js；又比如前端自动化工具，Grunt 和 Gulp；还有很多框架与库，如 Express.js、Meteor 和 Backbone.js、knockout.js，以及 MongoDB 等等。在最终的大作业中也学到了如何利用 git 与 Github 来实现协同工作。

通过这门课程确实学到很多，也动手去做了很多有趣的事，感谢老师和 TA~

### 魏婉婷：

虽然上学期学习 web 的时候也旁听了老师的周六课程，但因为时间关系没有坚持下去。这学期的难度是预料到了，每周都是几乎全新的内容，老师会以实际生产中的标准要求我们，也接触到许多较新的技术。一开始并不清楚软改具体的学习内容，后来觉得是升级版的 web2.0，直到在课程中的相互评价、期末 project 中慢慢体会出来。这门课程中最大的收获也许不是掌握了多少技术性内容，而是真正了解到现实社会中软件的生产和改进过程。

### 郑越升：

我觉得软改和其他课程不太一样，接触的学的都是最前端的可以直接使用的技术。但是这门课同时压力也很大，前半段的课程学习技术跳跃幅度很大，真的一周不花两天根本跟不上节奏，后半段的课程又考验团队配合以及实际项目开发的一些事项。

我觉得虽然通过这门课真的能学到很多东西，但是我觉得课程的设计还是有那么一点紧张，但也是没办法的事，WEB 的技术涉及方面太广，只有全部涉猎一下才知道自己适不适合 web，适合前端还是后台等。

我觉得个人对 web 技术还是挺感兴趣的，以后也会花多点时间在 web 上。

### 周腾腾：

1. 懂得团队的力量挺大的，能从队友这里学到很多的东西，接触一些自己平时不怎么接触的东西，队友的督促，也能让自己更好的做一件事

2. 一开始我挺喜欢 web 的，但是 web 牵涉太多的技术，而且技术涉及面很广，我也只学到了，怎么用，很多原理方面的东西都不懂，我还是想学习底层的东西，后阶段热情不够。转向了数据库和操作系统，偏爱一些更底层的東西。

3. 总体上，个人还是具备开发小型的应用，比如博客，能写出几千行的原生 javascript，

基本上懂得一些常用的技术和术语，算是入门级别吧。以后有机会再，抽时间来学习 web 了。

### 吴嘉华：

经过本学期对该课程的学习，无论是技术水平，还是团队协作能力都得到了极大的提升。

上学期学习过 web2.0 课程，曾经以为对于前端，只需学会 html, css, javascript 就满足日后前端开发的需求。然而，这实在是十分幼稚的想法。正是因为学习了这门课程，才让我清醒过来。技术这东西，是日新月异的，是永无止境的。

还清楚地记得，第一次课程作业是环形菜单。自认为对 css 比较熟悉的我，还是在写代码的过程中遇到了许多的问题。当晚做到凌晨两点才把效果实现了。由于本人有强迫症，看到一些小圆圈的大小和位置不太对，因此不断进行微调。到了凌晨四点，才完成了一个算是比较完美的版本。从那一刻起，我就觉得这门课程并不是想象中的那么简单，每次作业都必须十分认真地对待，不能蒙混过关。

首先学习的是 github 的使用，利用 github 进行项目管理，我觉得使用 github 提交作业这个主意十分赞，可以让更多的同学们熟悉 github。

每一节课的课程难度都会大大增加，每一节课都会接触一些新的技术、新的方法，因此要学习的东西也越来越多，但这些东西都是非常实用的。比如 grunt, jade 和 livescript，它们让我感受到了前所未有的编程快感。所以我认为，只会 html+css+js 是十分低级的。这门课程就等于为我们打开了一个崭新世界的大门。

后来，初次接触到 TOP 这个概念，面向思维的程序设计。我觉得，TOP 还是要建立在一定的技术基础上面的，要有一种足够简单的语言，使得程序员不纠结于过于繁琐的语法；要有一种比较智能的机制，如 grunt 等自动化工具。把他们的精力集中放在思考如何解决问题上面。还记得以前写 html 的时候，写一个标签，还必须要写一个闭合的标签。而现在使用了 jade，这一切都方便得多了；Jquery 也为程序员简化了 coding 的繁琐性。后来用了 livescript，与 Jquery 的配合也简直惊艳。可见，技术的发展推动着 TOP 的发展，TOP 也将是未来的趋势。

之后也学到了 nodejs，尽管它性能和功能十分强大，但是作为服务器端，异步编程这一点让我又爱又恨（其实更多的是恨，当 js 作为前端语言的时候并没有这种感觉）。多次掉进异步调用坑。用到现在还是没办法适应这种代码结构，多层回调，写的时候思维可能会被打乱。在我的思维中，代码都是顺序执行的，然而我并不能适应这种代码结构，尽管有一些解决方案如 promise。直到学习了 meteor，我对 nodejs 的看法才有了改观。但我并没有认为 nodejs 本身是不好的，只是个人对不同语言的喜好程度不同罢了。对我而言，还是喜欢一些非回调的如 python。但是对于 nodejs 还是需要了解和运用一下的，见识更多的事物总没有不好的（面试的时候才不会被人鄙视嘛...）。

随着课程的继续深入，作业也越来越难，交作业的同学也越来越少了。由于课程作业需要的知识十分紧凑，前一个星期的知识没有掌握，这个星期也无法做作业。技术的积累也决定着后面的课程项目是否能够顺利完成。这也让班里的同学慢慢分化。但我相信，能坚持到最后就一定能够收获到许多东西，事实也的确如此。

然后就到了课程的下半部分：课程项目。从一开始想法的产生，到项目规划书的完成。都是要通过小组成员之间的讨论和完善。用户的需求不是主观地凭借自己的想法来产生的，而是要通过调查来确定的。例如用户为什么组队？他们希望从组队中得到什么？怎样把组队的流程最简化？这过程中也少不了老师对我们项目的建议和指导。

项目规划书完成了，接着是原型图的绘制，由于项目规划书写得比较详细，于是很快就完成了。然后就是技术的选型。主要考虑是使用本学期所学到的知识，学以致用。没有选用

MVVM 框架（如 meteor），而选用了 MVC 框架（express），因为部分组员还不能熟悉 MVVM 框架。数据库类型的选择也经过慎重的考虑，本来想使用较为熟悉的 mongodb，但是由于数据的结构十分特殊，有一个 user 表，还有一个 team 表，每个 team 里面含有一个 user 的字段，另外还有一个 activity 表，每个 activity 里含有一个 team 字段。这样的多重嵌套关系，使用 mongodb 会在更新和删除的时候十分麻烦，因此最终选用了 mysql。

作为组长，需要协调整个队伍的工作。我把小组分为前端小组和后端小组，分组的依据主要看组员们的意愿。对于打代码能力较弱的组员，就让他们做产品的工作，如撰写体验报告等。不同的组员有不同的特长，让他们分别负责擅长的领域使团队协作的效益达到最大。

接下来就到了开发阶段，由于没用过 mysql，在设计表的时候没有什么想法。还好我们的组员周腾腾有设计数据库的经验，将表设计好了。首先是设计数据库交互模块，在这一个模块中花费的时间是最长的。与此同时，前端也开始开发了。

前端页面做出来了，但是发现跟我理解中的不一样。于是我意识到组员们相互交流是十分重要的。之后我要求他们对项目有任何不清楚的地方要马上提出来。组员们会积极在 github 上提出问题，也会有热烈的讨论，项目仿佛有了一种生命力。

最终，项目还算是基本完成，但还是跟预期还是有些差距，一些界面和功能还有待完善。但是我觉得更重要的是项目开发过程中的团队交流，思想碰撞，不断改进的过程，这也是“软件过程改进”的课程目标。这门课程让我真真正正体验了一个应用从想法的萌生再到开发的整个过程。这也是一次十分难得的经验。也让我觉得，开发项目也是一件很有趣的事情。相信以后还会在软件开发的道路上走很远很远，这门课程就是一个很好的起点。

最后，万分感谢王青老师和 TA 们在课程中给我们的指导！

## 四、课程建议

### 吴嘉华：

课程前半部分：建议恢复 review 环节，宁愿减少作业次数。我觉得课程的重点是怎样去写出更高质量的代码。另外可以布置一些团队作业，如能够将一个团队作业分成几个部分或者模块，让组长根据每个组员的能力去分配任务，然后再合并汇总提交作业。

课程后半部分：希望能够提供更多的项目开发指导。例如项目开发的一般流程，注意事项等等。增加 TA 跟进每个小组（如果 TA 资源足够），对于每个小组遇到的问题能够及时给出解答并能反馈给老师。

### 戴旋：

我前面列了一些技术方面的收获，光从这些，大家可能觉得这是一门关于 web 的技术课。那好，就把这门课当技术课看待吧，内容的含量可谓非常的庞大。这样庞大的知识量可能会导致学生的不求甚解，只会使用吧。而这样我想并不是一件好事。这门课程的名字是软件过程改进。关于软件过程改进这方面，对我来说，学到的真的不够。可能是我自己的原因吧...

### 魏婉婷：

这学期比较不幸地两个重要课程的每周代码作业 deadline 都在周六和周日，而课程都是在周三进行，半个星期里完成两项任务难免有些顾此失彼。而就实际排课来说周一周二是最有空的，所以既然简化了互评-修改环节，希望以后的课程能在了解同学具体情况后适当延长作业完成时间到 4-5 天。

**周腾腾：**

这门课，其实不能掉线的，一旦落下一次作业，后面基本就很难补了，希望老师对下一届，采取更严厉一些的做法，学生有些时候还是需要一些更严厉的惩罚的，大多数的学生时间不够，其实也就是把时间花在了一些其他的无聊的事情上。不过确实，静心学习不是一件容易的事情，记得以前高中分秒必争，大学何曾有学生有这样的想法，都是慵懒了，我自己也是，也只是尽量做的更好一些。