

# AC 组期末项目报告

组长：叶祥俊  
组员：林依挺  
奉卓瑶  
张秋怡  
唐文强  
卢建业  
张创涛

## 一、改进记录

### 1.重新探讨原型

经过小组的重新探讨，发现项目的立案无法走捷径。于是从原型开始重新着手项目。小组成员通过 Usability session 的方式（即俗称“贴纸条”）把项目的原型重新设计。并一一记录。对于期间遇到的问题，如“学校信息的录入”以及“课程名称的选择”进行了取舍和探讨。基本上对于整个项目的框架有了新的认识，也了解到了跳过上述步骤的话会产生怎样的后果。虽然这一步上花费了一些时间，但对后续的项目编写有十分大的帮助。

### 2.对数据模型的重新编写

在之前的课上曾经对数据模型有着疑问，如用户和课程相匹配。看原来框架的数据模型的编写，它以一个新的表来一一建立用户和帖子的关系。经小组探讨以及参考老师的意见。我们选择了抛弃这样的模型结构，而是在用户模型中建立一个以课程 ID 为元素的数组，在课程模型中建立一个以用户 ID 为元素的数组。如此，二者的联系就建立了起来，关键是要理清关系型数据库和文档型数据库的区别。不能以关系型数据库的思想硬套在文档型数据库之上。

### 3.修改页面层级

确立了原型和数据模型之后，开始对页面的层级进行修改，从原有的基础上加封一层新的层级。更改项目的主页，并将话题归类到课程板块之下，并相应添加课程管理页面。

#### 4.关联课程和帖子

修改帖子的数据模型，添加其分属的课程并关联，使得在对应课程板块下发出的作业贴仅仅能在这个版块下被看见，并添加订阅课程的功能。使得订阅的课程能够置顶。

#### 5.修改样式

根据需要修改样式

## 二、个人贡献

叶祥俊：收藏课程、关联课程和用户，建立和修改数据模型

涉及模块： control, proxy, model, view

张秋怡：修改页面层级

涉及模块： control, proxy, view

林依挺：用户信息录入补全，关联话题和课程

涉及模块： control, proxy, view

卢建业：增加课程添加管理页面

涉及模块： control, proxy, view

奉卓瑶： logo 设计，配色设计，网站样式修改

涉及模块： public, view

张创涛：添加新的 UI，网站样式修改

涉及模块： public, view

唐文强：网站样式修改

涉及模块： public, view

### 三、心得感想

13331239 唐文强

想想自己已经上了一个学期的改进，对 web 的制作有了新的认识，这个学期，学到了很多语言，也见识到了一些语言的方便性，更重要的是老师所说的那种思想，虽然没有完全吃透，但至少在自己的脑海中留下了印象，这也是基础的巩固吧，自己缺少的就是各种项目的锻炼，优秀的程序员总是在无数个项目中成长起来的。自己也要向着这个方向前进。

这门课前几周，确实感觉很难，因为之前自己也没用学过 web 的知识，所以起点有点低，而且作业不是那么容易，不过，自己还是坚持了下来，也同时学到了很多。也认识了许多大神，从他们的身上自己也学到了很多。

后面几周，老师让我们做自己的项目，在这个小团队里面，还是很开心的，上课大家能够在一起讨论项目的问题。然后模拟自己项目展示的过程，学到了许多。

感觉自己还欠缺许多知识，以后自己会慢慢补上的。

13354069 奉卓瑶

回首一个学期的课程，最记忆犹新的是开学第一次课老师说“这门课只有勇者才能坚持到最后”。舍友们纷纷退课，确实也给我带来了不小的压力... 再加上自己开学初态度懒散，第一次作业浑浑噩噩，这让我一下子质疑了自己。于是从第二次作业开始，软改成了我学习的重心。

感觉前 8 次 lab 就像打怪升级一样，代码上由简到繁，而思想上却又由繁到简。过程中，技能树的点亮如下：

- 使用 CSS3 控制网页上元素的动画及效果
- JS 的进阶使用，包括异步及面向对象
- Coffeescript, less, jade 的使用，其中 Coffeescript 成为了我现在最经常使用的
- GruntJS 和 MeteorJS 的学习，终于让我可以承认程序员不是体力活而是脑力活
- 单页面应用与多页面应用的灵活应用
- Git 的高效使用，代码协作

遗憾的是最后没能实施“作业改进”，我前几次 lab 曾尝试在老师讲解之后去改进代码，发现确实有很多地方可以提高，我想这样的“改进”是需要的也是一个程序员面对自己的代码所要不停去做的事情。

可以说整个过程是辛苦的，但是更是欣慰的。这一学期下来所新养成的技能比整个大一——年都要更丰富和实用。如果说上学期的 Web2.0 给了我学习 web 的信心和决心，那么这学期的软件过程改进就是打破所有原来的幻想让我端正态度。我还有太多的技术还没有学会，要走的路还很长！

13331333 张创涛

眨眼间，一个学期很快就要过去了，在这个学期里，很高兴能选到软件过程改进这一门课，让我体会到了辛酸与快乐。还记得上个学期刚开始选课的时候并不是很了解这一门课，当时一听到名字的时候我还以为是做电脑软件应用的开发，后来选上课了才发现这一门课主讲的是 web 应用的开发。一开始的时候，我其实并不了解 web 应用的开发是什么意思，只是很单纯地认为 web 就是一些网页堆砌而成，并没有一个应用的概念，但是后来上课才逐渐意识到 web 也是一种应用，并不仅仅只是网页的堆砌，同时需要学习的东西也是很多，过程也是十分辛苦。当时，每个星期主要花的时间就是在完成软改作业上面，从星期三一讲完课布置了作业开始，除了上课，基本就是在思考怎么完成软改作业和去实现软改作业，每次都是战战兢兢地在周六晚上提交了作业，而且由于每个星期学习的都是新东西，所以学起来就更加辛苦，也少不了抱怨，也不是很理解为什么要学习那么多的框架和语言。后来我们学习了 MVC、METEOR 之类的框架，在这其中我印象很深刻的就是写一个作业提交的网页应用的那个作业，老师为了让我们更好地理解不同框架间的差异及优劣性，让我们用不同框架地去实现同样的功能，这一点能让我们很清楚地明白学习到的这些知识点，我也对之前为什么要学习那么多框架的疑问有了新的看法。除了学习到的大量语言，老师教给我们的最重要的就是 **thinking oriented programming**，以前在写网页的时候，基本都是比较随意写的，想到了什么就写什么，所以代码写起来就很难看，但是后来从老师那里学习到了面向对象这一思想，对这个问题有了很大的帮助，现在我在写代码的时候都会仔细认真地思考整个脉络架构，而不是想到什么就写什么，当然这个过程也是十分艰辛的，因为你要预先想好可能会出现的所有情况，不然就一旦中间过程中出现了问题，整个代码改起来就会十分困难。在学习的整个过程中，我觉得团队合作也是比较重要的一环，团队之间需要沟通和交流，需要把一些细枝末节的东西考虑清楚，这样才有可能完成好整个工作，同时每个人的部分也不要拖拉，之前就是因为我的拖拉，导致了整个项目被拖了一下进度，这也让我意识到在团队合作中每个人的工作都是十分重要的。以上就是我对软件课程改进这一门课程的总结和一些想法。

13331183 卢建业

本课程的前半部分从基础的 js 开始讲起，让我们对 js 闭包，prototype 等概念有了更深入的理解，对后来学习一些高级框架，如 express JS，Angular JS 以及 meteor 等有很大的帮助，前半部分主要是知识的积累和工具的学习，无疑在前半段时间我们每周收获都是很明显的，同时为了完成当周对应的 project 也是蛮拼的。

在课程的后半部分时间里，我们是以项目为导向进行课程学习。我们小组做的是一个叫 AfterClass 的课程作业交流平台，在合作开发这个平台的过程中还是遇到了挺多的困难和波折的。后半段的学习相当与实战环节，对于我们这些平时接触大型合作项目比较少的学生来说，是一次不错的经历。

总结起来，这门课不但是一门很干货，让我们从中学到很大新技术的课程，并且还能在其中得到许多老师关于软件工程的理解，在项目开发中也学到很多以前没有得到的经验，用一个比较客套的词来讲，但却很能表达，那就是在这门课中受益匪浅。

13331157 林依挺

选修这门课的时候，完全不知道《软件过程改进》这门课是要学习些什么，一开始还以为会是比较枯燥的理论课程，但是上课之后才发现跟自己所想象中的完全不相同。现在回过头来看这门课，我觉得可以用“刺激”这个词来形容。“刺激”的感受不仅来自于每周六 8 点令人绝望的 deadline，更加来自于不断改进方法、不断学习新知识的充实体验。

#### **语言、框架、工具：**

之前在 Web2.0 的课程上学习过 css/JavaScript，当时以为挺简单的，不就是调调样式、增加一下交互嘛。现在看来当初所学习到的不过是皮毛，根本没有掌握到这些语言的精髓部分。比如 JavaScript 最精彩也最折磨人的部分：Object-Oriented JavaScript，通过本门课程，我也才真正算是接触到了这些部分。同时，在这门课上，我们才学习了多个 web 开发框架，比如 ExpressJs，Meteor 等，通过这些框架，我们可以非常快速便捷的创建出一个 web app，并且更加重要的是，在学习这些框架的过程中，我了解了 web app 的 server 端、client 端、cookie 等概念，学习了 MVC、MVVM 框架，为以后的 Web 编程打下坚实

的基础。完车课程学习的过程，我也学习到了一些便捷高效的工具的使用，比如使用 git 来进行代码版本控制。不管是语言、框架的学习，还是工具的使用，可以说都让我从之前比较随意的编程状态，转入一种更加专业化、更加高端的方式，在《软件过程改进》这门课上学习到的这些东西，真的让人受益匪浅。

### **Thinking Oriented Programming :**

TOP 的编程思想是老师介绍的一种编程思想。其实刚开始的时候我还是有疑惑的，但是通过完成实验作业以及更加深入的实践学习，终于理解了这种编程思想。在完成实验作业的过程中，我们使用 grunt 自动化地构建 JavaScript，使用 LiveScript、Sass 等语言来更加直接简单地编写代码，这些语言、工具的使用让我们“focus on coding”，“code what we think”；在编写复杂程序时，依照“自顶向下，逐步求精”的思想，将复杂问题分解为更加容易解决的子问题，使解决问题的思路清晰化。通过这些实践，我逐步接受和理解这种编程思想，相信这种编程思想还将继续指导我们的编程实践。

### **软件过程改进：**

就像老师强调的那样，软件过程改进首先重要的是“过程”。软件的编写是有一定的过程规范需要的遵循的。我们在完成小组项目的过程中，在最开始的时候，由于没能够遵循这个过程，走了不少弯路。从需求分析到界面设计，再到框架选择，再到编写代码，严格按照这一过程走，不仅能够少走弯路，更能够写出质量高的应用。再说一说“改进”的事，老师是通过在课上“小组汇报”、“答疑”的方式来进行这一部分的。通过这种方式，每次课我们都能发现自己项目中的问题。在这个过程中，老师给了我们很多建议，帮助我们不断改进自己的项目。通过后面这一段时间的项目编写，我们渐渐学会以团队合作的方式来共同组织编写，使用 github 来进行版本控制、任务分配和问题反馈，通过有团队交流来帮助我们更好地掌控项目的进度。虽然最后我们没有能够完全实现我们的整个项目，但是我们依然在整个

过程中学习到了很多。

12330402 张秋怡

经过了一个学期的学习，终于迎来了结课。个人心得如下：

1. 课程内容安排比较与时俱进，可以对当前较前沿的 Web 开发技术有一定的了解
2. 能在课堂上体会到 git 协作带来的便利
3. 用多页/Angular/Meteor 重复实现同一个应用，加深了对多种开发模式的理解

一些建议：

1. 在 GitHub 提交作业或许用分支来分开小组/个人而不是用文件夹会比较好
2. 前面的几次作业比较简单，其实可以通过自动化测试来评分，也减轻了 TA 的负担
3. 后面的几次作业因为环境问题 TA 没有实际运行打分，或许可以用 Docker 帮忙
4. 可以加入 ECMAScript 6 和 Virtual DOM 相关的知识，ES6 标准已经通过，明年再开课的时候 Node.js 应该已经合并完 io.js 的代码，目前常青型浏览器的 JavaScript 殷勤对 [ES6 的跟进](#) 也很快，并且可以用 Transpiler 转译到 ES5/ES3，异步控制和 CoffeeScript/LiveScript 这一块的内容可以跟着更新。

课程中遇到的一些问题：

1. 很多同学反映 clone 老师的 repo 再安装时会卡在 Sass，主要原因是 C 写的 libsass 在很多人的 windows 环境下需要用 node-gyp 编译，对 Visual Studio 的版本有一定要求，最好给同学一点提示或者换用纯 JavaScript 实现的 Less 避开编译问题。
2. 异步控制相关的几次作业里有一个并发请求的要求，给出的 server.js 用的是 Node.js 自带的 HTTP 模块，只实现了 HTTP/1.x 协议，依照目前浏览器的限制需要给 ajax 请求加时间戳来避开 HTTP Head-of-line blocking，否则看不到并发的效果。可以在服务器端换用 HTTP/2 的实现或者给同学一点提示。
3. 最后面的课程设计的上课方式（Q & A）似乎不太科学。



一个学期下来，收获良多。虽然不能完全掌握课程要领，但是还算是跟上了课程吧。这课程如同老师一开始所说的，很难。每周的任务可以说都让我头疼不已。一开始学习这门课程的时候方法错误了，单单去注意这周的作业要怎么写，怎么去模仿老师或同学的代码。而没有真正的去了解和钻研课程，使得学习起来非常的吃力。并且即使作业做完了也没有掌握要领，使得学习十分的被动。主要是一开始太急功近利了，心里很急躁，想要赶紧完成作业。到后来抛弃了这种模式，把作业当作是一项检验，而更多的功夫花在钻研之上。课程的后半段进入了真正的软件过程改进的学习，一开始觉得很不对口味，在想为什么要花那么多功夫在改进之上，直接一步到位不好吗？后来在项目中才慢慢发现，课上教的东西是对的。开始异想天开的一步到位，然后在途中就发现各种漏洞，各种与想法的出入，使得整个项目停滞不前。后来重头开始，从原型开始，项目才慢慢有了进展。整个学习的过程十分的充实，学到了很多的东西，关键是学习到了很多软件过程的思想。接下来还要继续学习这些思想，不管是去到软件行业的哪个方向，这些思想都将受用终身。

课程开始时的难点在于闭包和回调函数。上学期学的 web2.0 那仅有的知识在这节课上被打得体无完肤，只能自己下来去补。然后就是各种框架的使用，首先是框架的不熟悉运用加上各种安装不成功，各种配置环境不成功可以说对我的打击是挺大的。曾经也有过不写了不做了的想法，现在回头看也十分的可笑。当了小组项目的时候，各种弊端慢慢的浮现了出来，首先是对项目的了解不够，一开始各种天马行空，把项目的各个功能都想一遍，然后就想开始做，一开始做就处处碰壁，停滞不前。小组的分工也没有分好，还出现了整个小组等一个人的工作完成的情况，身为组长我感到十分的惭愧。后来依照老师上课所说的，慢慢理解了软件过程改进的真正含义。慢慢做了点像样的东西。

未来的日子，还是要多多回顾这课的内容，这可是受用终身的知识。