

# 软件工程课程情况调查

清华大学计算机系软件工程课程教学团队

2020 年 12 月

## 1 简介

本文试图探究学生在课程中的学习情况。我们通过调查问卷搜集数据，对开放性问题的回答文本进行主题分析。余下部分按如下方式组织。第2章介绍使用的问卷调查方法；第3、4章给出调查的直接结果；第6章讨论一些发现，以及本文可能存在的问题；最后，第7章总结本文的调查结果。

## 2 研究方法

本文采用匿名问卷调查的方式收集数据。问卷通过问卷星<sup>1</sup>设计和收集，问卷限制每个微信用户只能填写一次问卷，由此获得的微信名没有用于本文的分析。问卷在微信群中发放，该微信群由助教创建，大部分成员为 2020 年选修课程的学生，群中有助教和老师的参与；在课程进行过程中，相继有学生通过同学或助教加入微信群，截至停止回收问卷时，群中共有 286 名成员。在发放问卷的同时，发放了总金额 128 元的微信随机红包共 100 个，其中 2 个被助教抢到了。问卷题目主要包括如下方面，全部内容见附录 A。

- 背景信息。如院系、年级、项目、使用的语言；
- 学习情况。相关题目尽量采用李克特选项，避免设计的选项对答题者产生影响；
- 主观评估。如每周花费时间、贡献代码行数、大作业难度；
- 开放性问题。包括大作业选题时考虑的因素、以及对小作业的评价，分析开放性问题回答的方法与分析其他问题回答的方法完全不同。

## 3 定量调查结果

### 3.1 背景信息

课程早期分组名单中共有 266 人。发放问卷 24 小时内，共回收 91 份答卷，略低于发放的红包数量 98。通过邮件展开的对开发人员的调查回复率在 6% 到 36% [5]。除去项目选择“其他”的 1 份问卷，我们的答卷大概覆盖 34% 的学生，与 [5] 中报告的回复率相当。填写问卷用时如图 1，用时最短为 64 秒，没有回答选填的开放性问题；最长超过 1 个小

时。回收的 90 份答卷中，年级分布如图 2，包括 87 名计算机系学生。90 人中，32 人在上大学之前学习过编程，占比 35.6%。

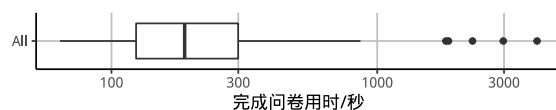


图 1: 完成问卷的时间分布

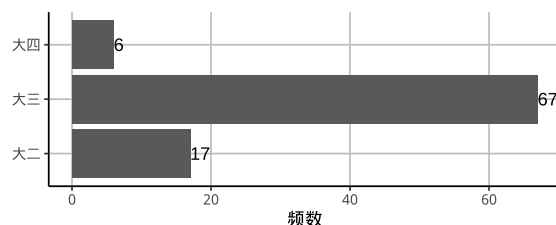


图 2: 年级分布

图 3 呈现了同学们使用的编程语言/框架情况。“其他”选项中的内容被展开，扩充语言/框架列表。由于没有出现在正式选项中，新加入的语言/框架可能与其他选项存在重复，或者没有被其他答卷纳入考虑。新加入的语言/框架包括：AngularJS、jQuery、Lucene、Mongo、NestJS、Node.js、Python、React Native、Scrapy、Solr、SQL、TypeScript。答卷的项目分布如图 4。

### 3.2 学习情况

近八成学生参与编写单元测试，60% 的学生参与配置 CI/CD<sup>2</sup>，如图 5 所示。图 6 将“不同意”与“同意”在基准线两侧分别绘制。五点李克特选项存在趋中倾向，“不确定”被划入“不同意”的范畴，对于大部分问题，这意味者结论倾向于认为教学环节存在问题。大部分学生希望获得更多来自课程的帮助；89% 的学生清楚每周的任务，43% 的学生倾向于觉得自己的大作业不集中在例会前完成。表 1 将本次调查与 2019 年调查中相同的问题进行对比。相比于 2019 年的受访群体，本次填写问卷的学生在遵循课程要求方面更积极，表现为本地执行单元测试和代码风格检查、根据 SonarQube 报告重构代码、做大作业时清楚任务的比例更高；同时，在

<sup>1</sup><https://www.wjx.cn>

<sup>2</sup>参与配置 .gitlab-ci.yml 或 Dockerfile

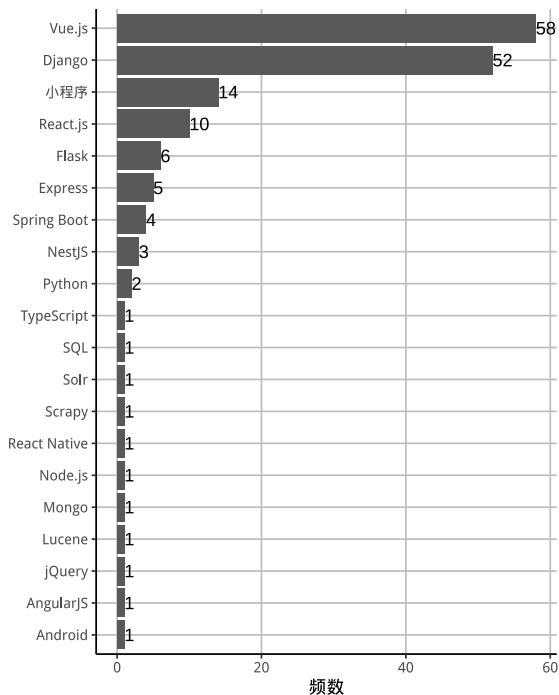


图 3: 编程语言/框架流行度

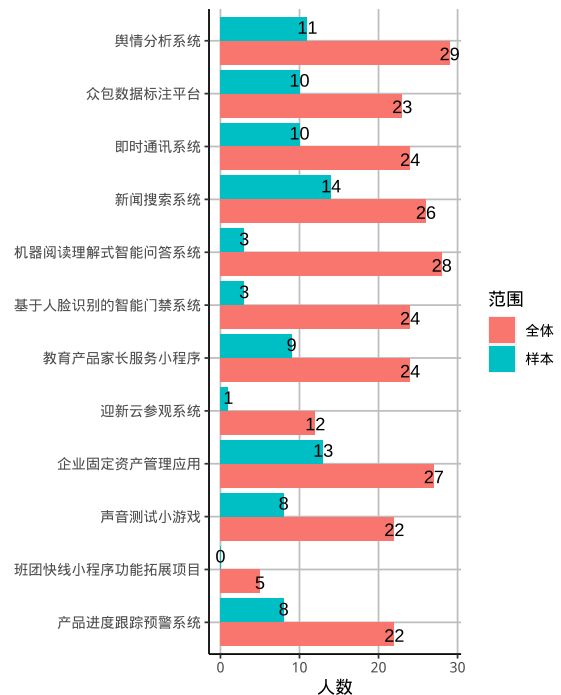


图 4: 项目分布

例会前 24 小时内完成当周任务的比例也明显高于 2019 年的受访群体。

问题“在完成大作业的过程中，你会向谁寻求帮助？”以排序题的形式给出，访问问卷时，选项顺序随机。该题共 6 个选项，其中，“其他”选项可进一步填写内容。对于一份答卷，排在第  $i$  位的选项计  $7-i$  分，没有出现的选项计 0 分，每个选项的平均得分如图 7。主要的解决问题途径是“自行解决”和“同学”，排在第三位的是“老师/助教”，与 2019 年的调查结果相近。

### 3.3 主观评估

受访者估计的每周用于完成大作业时间分布如图 8，贡献的代码行数如图 9。图 10 呈现了学生对大作业难度的主观评价，四成学生觉得大作业有难度，低于 2019 年的调查结果（66%）。

## 4 开放性问题

并非所有答卷都回答了开放性问题；在一份答卷中没有出现的内容，也并不代表答题者对相关内容没有看法。因此，本文不再统计某一类回答出现的数量，而只关心有哪些回答。共有 45 份答卷给出了大作业选题时的考虑，40 份答卷分享了对小作业的看法，21 份答卷在问卷最后留言。我们对上述答卷文本采用主题分析方法 [1] 进行标注、归类。

### 4.1 大作业选题时考虑的因素

在大作业选题时，学生会考虑的因素如图 11 所示。

#### 4.1.1 可行性

项目不应太难；有的同学倾向“看起来比较简单”的项目。对于“易上手”和“工作量”两个方面，摘录答卷文本如下：

（希望）“有比较趁手且成熟的技术栈可用（感觉微信小程序就不具备这一点）”

“涉及对方提供硬件/后端 API 的，倾向于不选，因为 debug 上有一些不方便……”

“内卷空间巨大（多端，网页/APP/小程序）甚至接近无限（按算法效率排序得分）的，不选”

#### 4.1.2 用户代表背景

摘录“用户代表背景”相关答卷文本如下：

“用户代表的技术理解帮助程度”

“用户代表是不是企业的员工”

“项目的背景，涉及平台的社区等”

#### 4.1.3 其它

对于“简单标准的项目”，有如下答卷文本进行说明：

“总的来说需要一个简单标准的项目才能更好地完成对软件工程的实践”

“是不是纯 web 应用”

“希望项目的方向更偏向于软件工程这门课程本身，而不是去学习某一方向的奇技淫巧。”

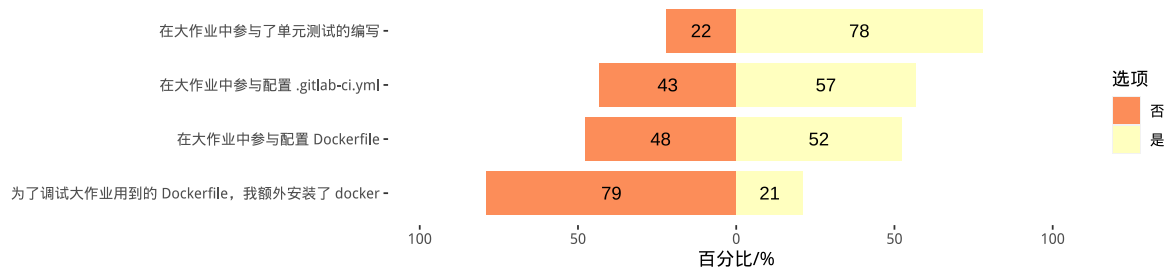


图 5: 是否参与单元测试与配置 CI/CD

表 1: 2019 年与 2020 年调查结果对比。在 2020 年的调查结果中, “参与了 CI/CD 的配置” 指参与配置 .gitlab-ci.yml 或 Dockerfile。

情况/倾向	2019 年 (%)	2020 年 (%)
我在上大学之前学习过编程	39	36
我在大作业中参与了单元测试的编写	70	78
我在大作业中参与了 CI/CD 的配置	66	60
为了调试大作业用到的 Dockerfile, 我额外安装了 docker	31	21
我可以将课堂讲授的内容应用到作业中	35	46
我希望获得更多来自老师、助教的帮助	72	69
我在完成作业时不应该借助老师、助教的帮助	9	7
课程提供的样例仓库帮助我完成了 CI/CD 的配置	59	67
每次提交代码前, 我会在本地执行单元测试	49	63
每次提交代码前, 我会在本地执行代码风格检查	52	63
我会根据 SonarQube 报告重构代码	35	71
单元测试帮助我发现代码实现中的问题	57	53
我的工作每次在例会前 24 小时内完成	24	38
做大作业的每一周, 我清楚自己的任务	73	89

## 4.2 对小作业的评价

### 4.2.1 对大作业的帮助

除了认可小作业的帮助, 另有三种观点。

引导学习 “对于没有相关知识基础的同学是一个很好的起步, 作业也是一个很好的可以强迫同学完成学习的办法”

没有帮助 “前端完成后端没什么帮助”

平时成绩 “没什么帮助, 不过至少提供了能稳定拿分的渠道”

### 4.2.2 建议

细化文档 小作业文档需要更加详细。有同学提到, “文档有很多需要完善的部分, 尤其是 *vue* 的作业”。

增加知识深度 单项小作业可以训练更多的内容。有同学表示, “*git* 训练不够”。

增加覆盖范围 大作业中会用到的语言、框架与现有小作业包含的内容不完全匹配: 大作业中可能会用到小作业中没讲的语言或框架, 小作业中包含的内容不一定会在大作业中用到。此外, 有同学 “希

望未来能以类似小作业的形式介绍单元测试框架、*docker* 等必要知识。解决技术问题是真实软件工程的一部分, 但应该不是大作业要重点锻炼和制造困难的地方。”

定位 小作业的定位是帮助同学掌握部分语言、技术, 降低大作业的学习成本, 但这个目的还没有很好地实现。“前端小作业还是很不会做, 做完还是不知道怎么写前端, 也许给些更多的教程会好些”。一种看法如下:

“但作业也会有一定的功利性, 比如此次前端作业中的框架性很明显, 如果只是为了完成的话, 只要完形填空就可以了, 这样其实并不会真正入门。一种能够引导同学自行探索大胆尝试的形式可能是更好的, 比如可以考虑弱化对于分数和实现方式的要求。”

## 4.3 留言

### 4.3.1 团队

团队关系是促使大作业顺利完成的重要因素。一方面, “感觉组里氛围好的时候大家都在一起努力, 就很和谐快乐”; 另一方面, “组到不靠谱的队

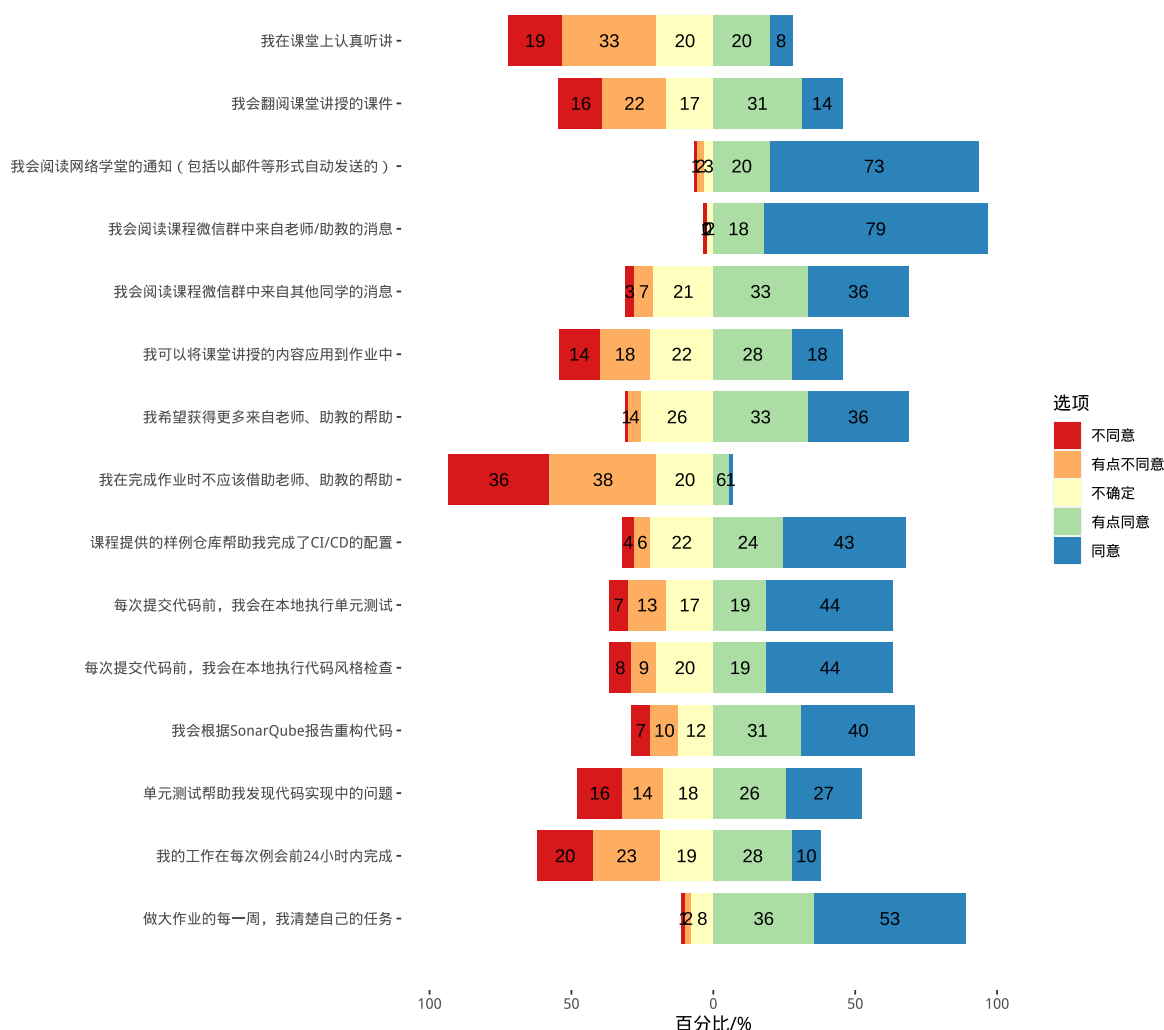


图 6: 学习情况

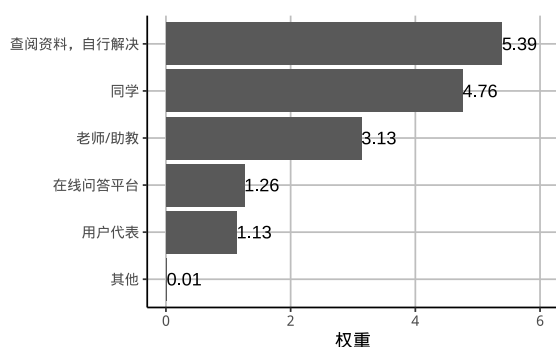


图 7: 在完成大作业的过程中，你会向谁寻求帮助？

友太痛苦了 2333”。一位同学写道，“大作业很大程度上取决于队友配合，而不是个人能力”。

#### 4.3.2 吐槽

学生的吐槽罗列如下：

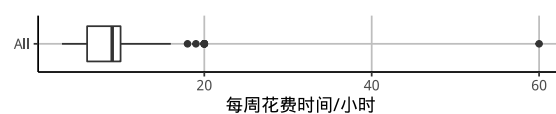


图 8: 每周用于完成大作业的时间估计/小时

“大作业很难，很累，又有点无趣”

“建议下次硬件水平提高不要动不动全局 GC”

“希望可以多提供一些框架的 ci/cd 配置（比如 vue 的，虽然我自己没写 vue，但听一个同学说他在配 vue 的 ci/cd 时遇到了一些困难）”

“希望关于 ci/cd，持久化配置以及 soner-cube 的知识再多讲一点”

“感觉课堂小测没什么意义，大多数时间都是速览 PPT 后作答... 不过这课想要在课堂上让同学们有很大收获确实挺难的，

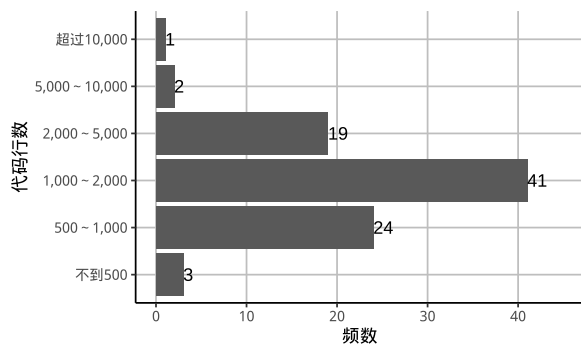


图 9: 最后一次例会的大作业版本中, 贡献代码行数估计

期待教学团队的课改”

“感觉到有的地方用户需求文档说的还是比较含糊, 使得最后提交的时候不清楚什么是必选什么是可选。另外单从描述上来看觉得各个项目难度还是相差较大, 不太均衡。”

“在写的代码中, 用于单元测试的代码比实际逻辑代码还要多。”

#### 4.3.3 其他

“版本控制学得好软工会轻松很多”

“对于每周花的时间不太确定, 但基本零碎时间都用去了”

“现有课程形式挺好”

## 5 相关性分析

本章对于问卷选项之间的关联进行探讨。

从问卷中, 共抽取 30 个变量进行分析。对于年级, 将“大二”、“大三”、“大四”分别编码为 2、3、4。对于“我在上大学之前学习过编程”与图 5 中的 4 个问题, 将“否”、“是”分别编码为 1、2。对于图 6 中的 15 个问题, 将“不同意”、“有点不同意”、“不确定”、“有点同意”、“同意”分别编码为 1、2、3、4、5。类似地, 对于代码行数贡献、大作业难度两个问题, 将选项按顺序编码。对于完成大作业中寻求帮助对象的排序问题, 对 6 个选项分别计算  $7-i$  以表示该选项的重要程度, 其中  $i$  为位次; 选项在答卷中未出现的, 其重要程度为 0。在上述 29 个变量以外, 对每周花费时间的估算也被纳入考虑。

对于上述变量应用 NOTEARS 线性模型<sup>3</sup>进行因果推断, 结果如图 12。NOTEARS 使用结构方程刻画变量之间的因果关系, 并基于损失函数在变量之间寻找构成有向无环图的因果关系 [6]。NOTEARS 没有找到 7 个变量<sup>4</sup>与其它变量之间的关联, 因此, 图中没有呈现这 7 个变量。分析表明,

<sup>3</sup><https://github.com/xunzheng/notears>

<sup>4</sup>本地安装 docker、不应借助老师/助教帮助、例会前 24 小时完成工作、优先向同学寻求帮助、优先自行解决、优先从其他渠道获取帮助、大作业难度

向老师/助教/用户代表寻求帮助、阅读微信群中其他同学的消息、借助样例仓库会降低每周在大作业上花费的时间。

由于问卷内容的局限性, 上述变量可能存在未被纳入考量的共因, 变量间的关系也很可能远比线性关系复杂。因此, 图 12 中呈现的每一条边还需要进一步验证与解释。

## 6 讨论

### 6.1 发现与认识

基于调查结果, 我们对学生作业完成情况主要有如下发现与认识。

发现 1 在大作业与课程内容之间, 学生希望更多练习。

学生独立完成任务的能力与完成作业需要的能力之间可能存在差距, 此时, 适当的帮助可以促进学生完成任务并提升能力上限 [4]。小作业中, 学生可以练习课堂讲授的内容。为了应对不同基础的学生, 小作业需要提供细致的说明。

在已有的小作业之外, 学生还希望小作业覆盖更多语言和框架。然而, 助教团队掌握的技能很可能无法包括大作业中用到的全部语言和框架; 基于部分了解而设计的作业则很难起到降低学生学习成本的目的。

发现 2 相比于授课内容, 学生更关心和作业相关的内容。

课程传递的信息主要包括课堂讲授, 以及通过网络学堂和课程微信群发布的通知。“阅读课程微信群中其他同学的消息”则不在课程要求之内。然而, 如图 6 所示, 相比于课堂上认真听讲或翻阅课件, 更大比例的学生会阅读课程微信群中其他同学的消息。与之对应的, 超过六成的学生遵循课程要求的最佳实践, 但近四成学生的工作在例会前 24 小时内完成。这表明, 授课内容被放在相对成绩更次要的地位。另一方面, 微信群中其他同学遇到的问题可能也是自己遇到的问题, 对微信群中消息的留意表明授课内容不能很好地解决大作业中实际遇到的问题。

发现 3 学生在大作业选题时有诸多考量。

学生在大作业选题时会考虑可行性、用户代表的背景、助教水平、个人喜好、文档详细程度、项目的实际意义等诸多因素。由于问卷设计的缺陷, 我们没能在大部分答卷文本中分辨出受访者的倾向。例如, 对于“难度”这一因素, 存在“选择更有挑战性的项目”和“选择更简单的项目”两种倾向。

问题 1 队友情况影响大作业体验。

小组作业中可能存在学生划水, 但学生之间也确实存在差异。好的团队关系使人愉悦; 而局限于分工的合作关系 [2] 则会出现相互甩锅、关系紧绷的现象。团队内部如何合作是软件工程的重要议题 [3], 但在当前的软件工程课程中还未涉及。



图 10: 大作业难度评价

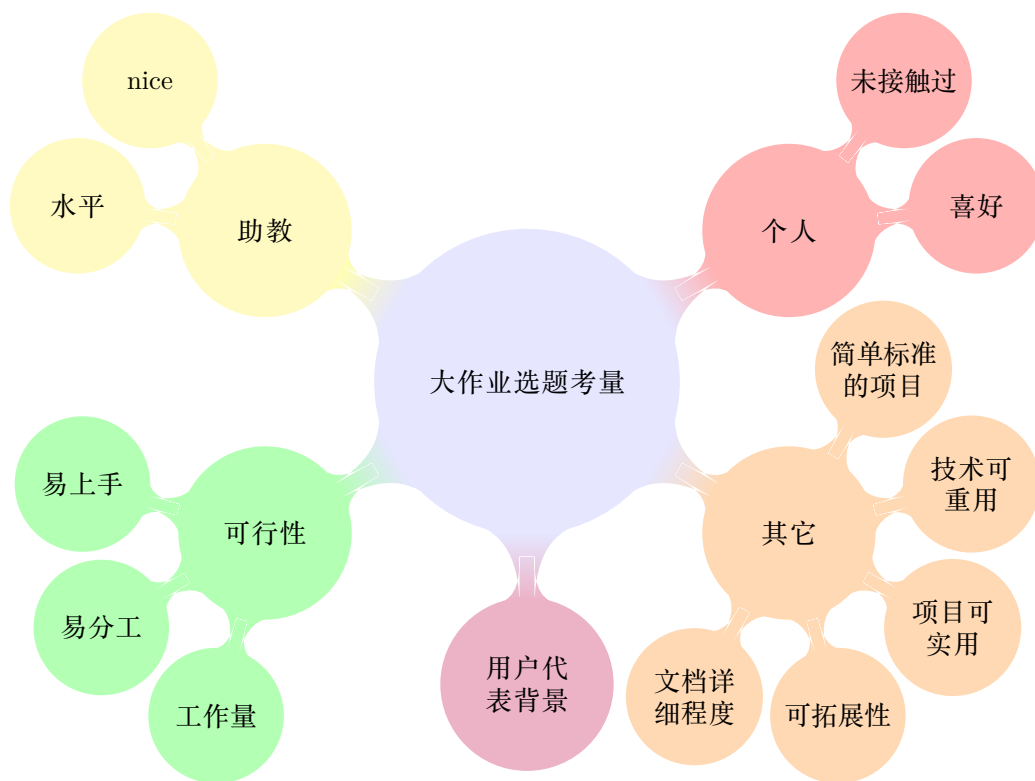


图 11: 在大作业选题时，学生会考虑的因素

## 6.2 有效性检验

### 6.2.1 构造有效性

构造有效性表征研究方法本身的有效性。微信群没有包含全部选课学生，不是所有选课学生都填写了问卷，调查结果与总体情况之间的差异不明确。例如，图 6 显示 97% 的受访者会关注微信群中来自老师/助教的消息，则不经常关注微信群的学生可能没有留意到调查问卷。我们发现一份答卷选择的项目与微信号对应的学生不一致。我们没有纠正该“错误”，也无力检验各份问卷的回答是否属实。

### 6.2.2 内部解释性

内部解释性指研究对象是否被正确度量。问卷是一种方便的统计方法，问卷议题由研究者设定，可能影响答题者的判断。我们采用五点李克特选项设计单选题，以尽量避免研究人员引入的选项对答题者的额外影响。对工作时间和代码贡献行数的估计可能与事实存在偏差，相关数据没有进一步进行解释。开放性问题允许受访者更完整地表达自己的意见，但对文本进行主题分析时，不免受到研究人

员既定认识的影响；此外，大量答卷文本并不完整，给主题分析带来了困难。本文小心地组织文本信息，尽量避免对文本进行过度解读。

### 6.2.3 外部解释性

外部解释性指研究结果是否适用于其他群体。答案是否定的。如表 1 所示，调查结果在不同届学生之间存在差异。不同课程、不同院系、不同学校亦不可照搬调查结果。

## 7 总结

本文通过调查问卷搜集数据，并对答卷文本进行主题分析。我们发现，

- 在大作业与课程内容之间，学生希望更多练习；
- 相比于授课内容，学生更关心和作业相关的内容；
- 学生在大作业选题时有诸多考量。

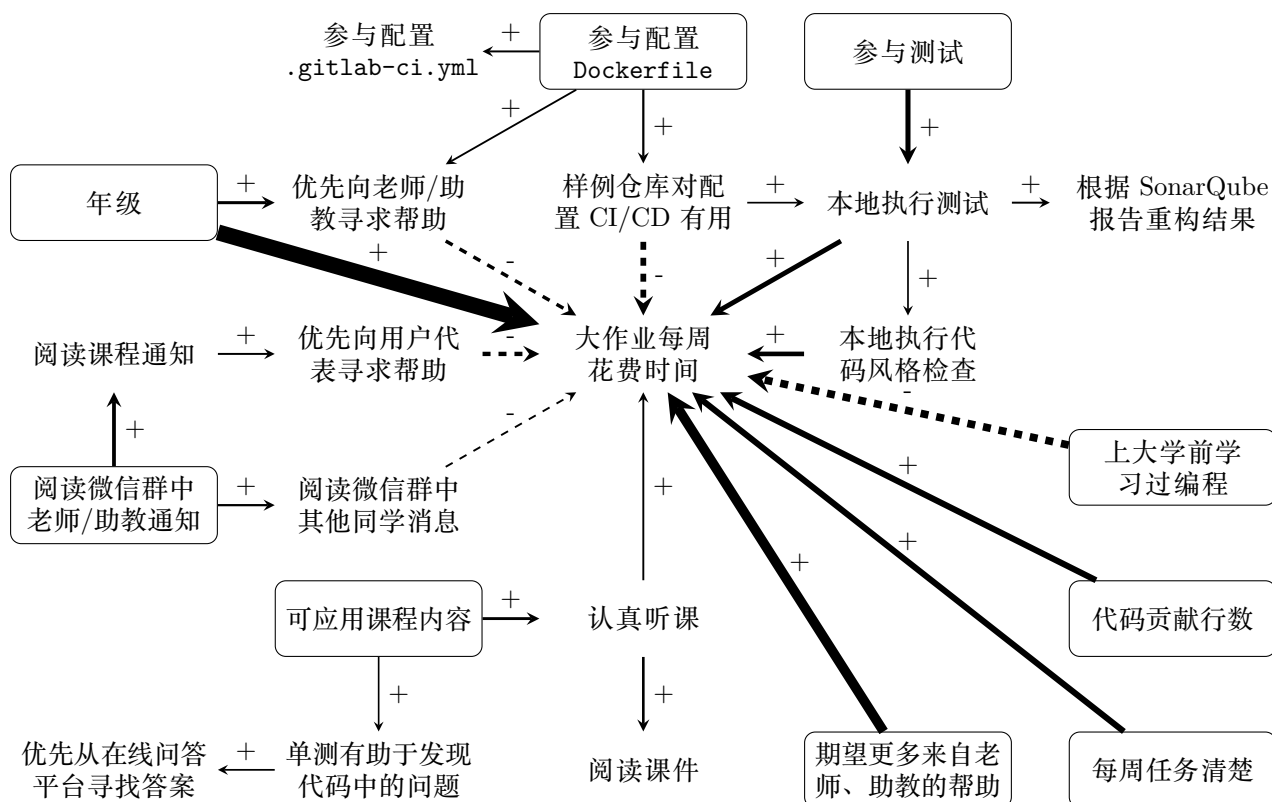


图 12: 相关性分析。实线表示变量之间为正相关，虚线表示变量之间为负相关；连线方向表示推断的因果方向，由因指向果；连线粗细为 NOTEARS 线性模型拟合时的权重。

同时，课程存在如下问题，

- 小作业的说明不够清晰；
- 队友情况影响大作业体验。

## 致谢

感谢同学们对课程的支持。感谢用户代表在课程教学实践中的付出。感谢腾讯云提供云计算资源。

## 参考文献

- [1] Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101, January 2006. doi: 10.1191/1478088706qp0630a.
- [2] Mats Daniels and Åsa Cajander. Constructive controversy as a way to create “true collaboration” in an open ended group project setting. In *Proceedings of the Twelfth Australasian Conference on Computing Education - Volume 103, ACE '10*, page 73–78, AUS, 2010. Australian Computer Society, Inc. ISBN 9781920682842.
- [3] Frederick P. Brooks Jr. The surgical team. In 杨伟国, editor, *The Mythical Man-Month*, chapter 3, pages 27–37. 中国电力出版社, 北京, 2003.
- [4] Saul A. McLeod. What is the zone of proximal development?, 2019. URL <https://www.simplypsychology.org/Zone-of-Proximal-Development.html>.
- [5] E. Smith, R. Loftin, E. Murphy-Hill, C. Bird, and T. Zimmermann. Improving developer participation rates in surveys. In *2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, pages 89–92, May 2013. doi: 10.1109/CHASE.2013.6614738.
- [6] Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. DAGs with NO TEARS: Continuous Optimization for Structure Learning. In *Advances in Neural Information Processing Systems*, 2018.



## A 调查问卷

编号	问题	题型
1	院系 计算机系 其他	单选
2	年级 大一 大二 大三 大四 其他	单选
3	你参与的项目是	单选
4	你在大作业中使用的编程语言/框架有 其他	多选 填空
5	以下描述是否符合你的情况	单选，是/否
5.1	我在上大学之前学习过编程	
5.2	我在大作业中参与了单元测试的编写	
5.3	我在大作业中参与配置.gitlab-ci.yml	
5.4	我在大作业中参与配置 Dockerfile	
5.5	为了调试大作业用到的 Dockerfile，我额外安装了 docker	
6	你在多大程度上认可以下论述	五点李克特选项
6.1	我在课堂上认真听讲	
6.2	我会翻阅课堂讲授的课件	
6.3	我会阅读网络学堂的通知（包括以邮件等形式自动发送的）	
6.4	我会阅读课程微信群中来自老师/助教的消息	
6.5	我会阅读课程微信群中来自其他同学的消息	
6.6	我可以将课堂讲授的内容应用到作业中	
6.7	我希望获得更多来自老师、助教的帮助	
6.8	我在完成作业时不应该借助老师、助教的帮助	
7	在完成大作业的过程中，你在多大程度上认可以下论述	五点李克特选项
7.1	课程提供的样例仓库帮助我完成了 CI/CD 的配置	
7.2	每次提交代码前，我会在本地执行单元测试	
7.3	每次提交代码前，我会在本地执行代码风格检查	
7.4	我会根据 SonarQube 报告重构代码	
7.5	单元测试帮助我发现代码实现中的问题	
7.6	我的工作每次例会前 24 小时内完成	
7.7	做大作业的每一周，我清楚自己的任务	
8	在完成大作业的过程中，你会向谁寻求帮助？ 同学 老师/助教 在线问答平台 查阅资料，自行解决 用户代表 其他	排序，选项顺序随机
9	你大概每周花多少小时用于完成软件工程专业大作业？	填空
A	最后一次例会的大作业版本中，你大概贡献了多少行代码？ 不到 500 500 ~ 1,000 1,000 ~ 2,000 2,000 ~ 5,000 5,000 ~ 10,000 超过 10,000	填空 单选
B	你觉得大作业难吗？	五点李克特选项
C	在大作业选题时，你会考虑哪些因素？	
D	你对小作业有什么评价吗？	



编号	问题	题型
	例如，完成小作业对完成你的大作业有帮助吗?	
	你希望获得调查报告吗?	单选
	不用了	
	邮箱	填空
	有什么想说的吗?	