哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

2017年秋季学期软件工程
实践项目手册

王忠杰
rainy@hit.edu.cn

2017年9月3日

如何做软件实践项目？

- **不希望看到这样的情形－－**
  - 需求分析：学生们不懂企业的需求是什么，上课睡觉；
  - 设计阶段：学生们画了许多 UML 图，用设计工具画了不少矩形框、菱形框，如此而已；
  - 实现阶段：学生们开始讨论非常细节的问题，UML早已经扔到一边；
  - 稳定阶段：学生们中十分之一的人开始写代码，其他人不知道在干什么．代码大部分情况下都不能工作，所有设计好的种种黑箱和白箱测试都无从开始；
  - 发布阶段：这个只有一天时间，就是最后检查的那一天，同时还有人在调试程序；
  - 维护阶段：课程结束了，同学们对自己的产品没有任何维护，放假了！
- **大部分同学们都说这门课特别没用，自己根本没学到什么本事，然后下个学期，新的一批学生进来重复这一过程...**

如何做软件实践项目？

- 面向选定的题目，根据自己的直觉和当前能掌握的技术，马上就进入开发(Code-and-Fix)，形成一个版本；
- 在写程序的过程中，不断理解澄清需求；
- 在现有版本的基础上，利用软件工程的方法进行需求分析和设计，进入迭代，不断完善前版本；
- 继续深入理解需求，循环进行迭代设计和开发；
- 设计测试用例，测试。

- 在一学期内，遵循敏捷开发过程，完成两次迭代。

要开发“真实的、可用的软件”

- **要做“真实的项目”，有真正用户的软件。项目要有活的用户,只有活的用户才有活的需求，才有活的场景，活的测试用例。 只有活的用户才决定大家所写的软件是否值得使用。**
- **只有真实的用户才会迫使项目团队反思在需求分析和设计上的问题。**
- **开发过程中也要遵循真实的软件项目管理流程。**
  - 如何在有限的时间内交付有价值的软件给特定的用户。 “真实”这一条件也促使大家做 “现实”的项目和项目管理。
  - 不要仅仅当成一个作业去应付，也不要做不切实际的空想。
- **三真：真实的用户、真实的需求、真实的使用。**

要开发“真实的、可用的软件”

- 写出一个可用的、有实际用户的软件。 这对大多数人来说是第一次。
- 完整体验软件生命周期，对于生命周期的各个阶段有实际的了解。对于软件设计有实际的掌握， 对软件开发的具体技术有实践能力。
- 了解软件团队的各个角色、各个角色之间的互动，对于其中一个角色有实际的深入体验。
- 学习如何与不同的角色打交道，培养团队精神，学会解决冲突的几种方法。

分组，实践团队合作

- 每组3人，小班内组合，每组有1名组长 (“项目经理”/“产品经理”)
- 当分组有冲突时，班长作为“CEO”负责协调分组 (“软件公司”);
- 为小组起一个有意义的名字 (“产品的名字”);
- 同一班级内各组选题不能重复。
- 建议：班长召集集中的选题会；有意向做组长的学生发言，阐述自己的选题、对组员的技术和能力要求；其他学生表达加入小组的意向；组长根据自己的条件选择组员；如果没有被其他小组选中，由班长指派分组。

班级	总人数	总组数	3人组数目	4人组数目
5班				
6班				
英才				

编程语言与运行环境要求

- **编程语言：JAVA**
- **软件架构：B/S + MVC**
- **运行环境：部署在云上，客户端为PC**

- **需要提交的源代码：**
  - 第一轮迭代系统、最终验收系统 (对应于GitHub项目中的两个Release)；
  - 代码中应包含测试用例，并使用工具评估测试用例的覆盖度；
  - 包含完整的运行环境、配置文件、各种外部库。
- **不符合上述要求的项目不允许参加答辩。**

建议使用的软件工程工具

- 需求管理：VersionOne (用户故事，product backlog)
- 原型设计：MockupBuilder
- 开发工具：Java + Eclipse + Maven
- 需求分析与建模：UML Designer, StarUML、GenMyModel
- 测试工具：jUnit、HTTPUnit、DBUnit、Struts TestCase、EclEmma、jMeter、LoadRunner
- 源代码管理、版本控制、团队协作：Git / GitHub
- 项目管理工具：VersionOne (sprint backlog, burndown chart)

阶段划分与成绩考核

- **第1-4周：组队、选题、需求列举与优先级、迭代计划；**
- **第5-11周：第一次迭代；**
- **第11周：演示/答辩，教师评价、同行评价、外部评价；**
- **第12-15周：第二次迭代；**
- **第15周：演示/答辩，教师评价、同行评价、外部评价。**

- **成绩：**
  - 第一次迭代：40%
  - 第二次迭代：40%
  - 项目管理：10%
  - 真实用户评价：10%

Part 1: 开题(第4周)

- **包含的内容：**
  - 对题目的理解：所选系统的基本背景、现实意义、用户及需求来源等。
  - 功能需求：简要列出本系统所包含的功能清单(未来可扩展或改变)
  - 性能需求：简要阐述本系统拟达到的各方面NFR
  - 架构：拟开发的系统的基本架构
  - 技术：编程语言、开发环境、运行环境
  - 分工：小组内各成员计划如何分工完成整个项目
  - 开发进度计划：以周为单位，阐述每周拟达到的目标
  - 真实用户来源
  - 从技术、时间、人员的能力、真实用户等角度分析完成项目的可行性
- **无需提交，没有分数，各组自愿与教师做各种形式的沟通，确认对题目的理解是否正确、无偏差（教师/TA扮演项目的用户角色）。**

Part 2: 第一轮迭代检查 (第11周)

- **如何提交：**
  - 将本组目前完成的project第一轮迭代所形成的源代码，打包成rar或zip文件，上传到CMS实践项目区“第一轮迭代（40%）”内；
  - 每组只需由组长提交一份程序即可；
  - 命名规则：班号-小组名称-1st-Code.zip；
  - 同时，在GitHub上应有完整的开发历史 (杜绝deadline之前一次提交，要求在整个迭代期间通过git管理各成员本地机器的代码，有节奏的推送至GitHub)；
- **如何接受检查：**
  - 第11周实验课上，每组自行携带计算机至实验室，现场演示第一轮迭代的开发结果，不需要完整的功能，但已开发的功能必须可运行、可演示。
  - 每组选派1名代表，演示并口头讲解，时间为8分钟；
  - TA提出问题，各组回答并记录，作为第二轮迭代中进行改进的依据，Q&A的时间4分钟。

Part 2: 第一轮迭代检查(第11周)

- **检查的评分标准：**
  - 是否已在CMS上提交源代码（未提交代码的不能进行现场检查）
  - 对需求理解的正确性
  - 已完成的功能规模（针对于原始需求的百分比）
  - 原型的非功能表现（稳定性、健壮性、界面友好性等）
  - Github项目管理情况（是否在Github上提交和管理本组的代码、 Github提交的频度与代码量、3人是否均参与开发并向Github提交）
- **分数：**
  - 三人获得同样的分数；

Part 3: 最终答辩(第15周)

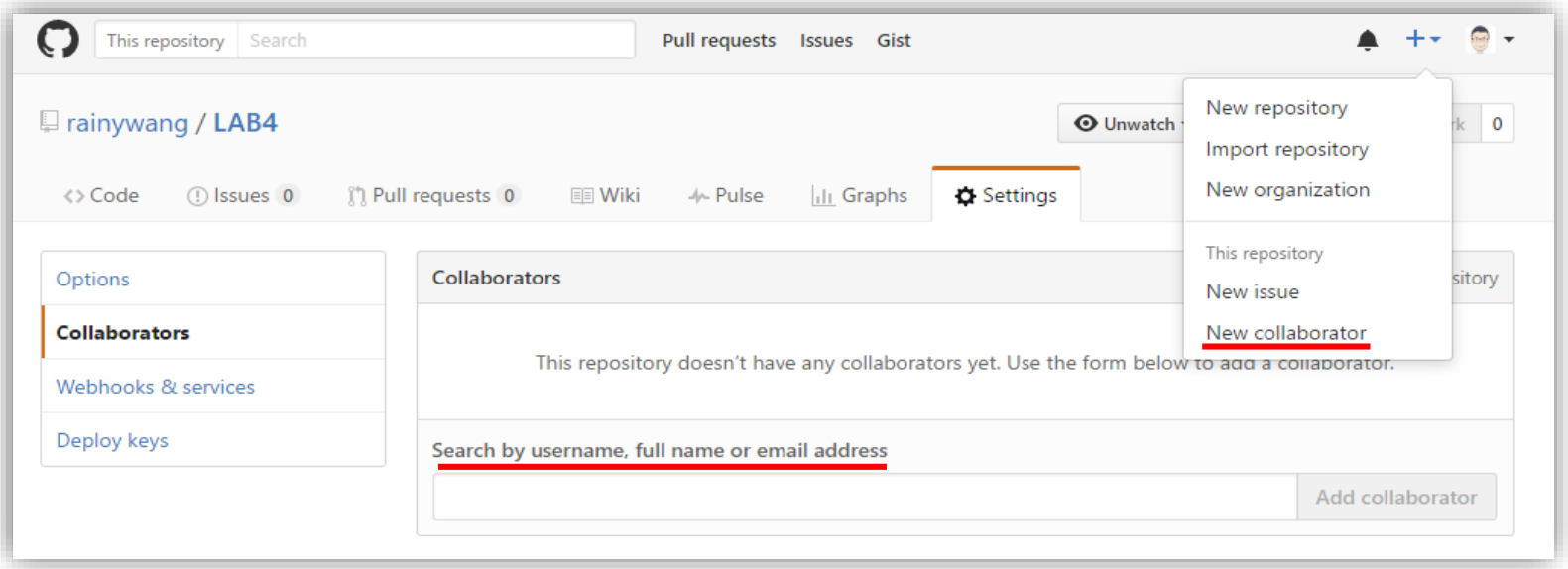
- **最终答辩之前，在CMS上提交电子版poster和demo，以及结题报告**
  - 答辩现场张贴实物poster；
  - 答辩时，现场演示软件系统，教师随机提出测试请求，共12分钟；
  - 需邀请至少1位真实用户到场并进行评价；
  - 其他要求：同第一轮迭代的检查要求
- **打分依据：**
  - 随机测试的结果质量
  - 真实客户的评价
  - demo/poster的质量

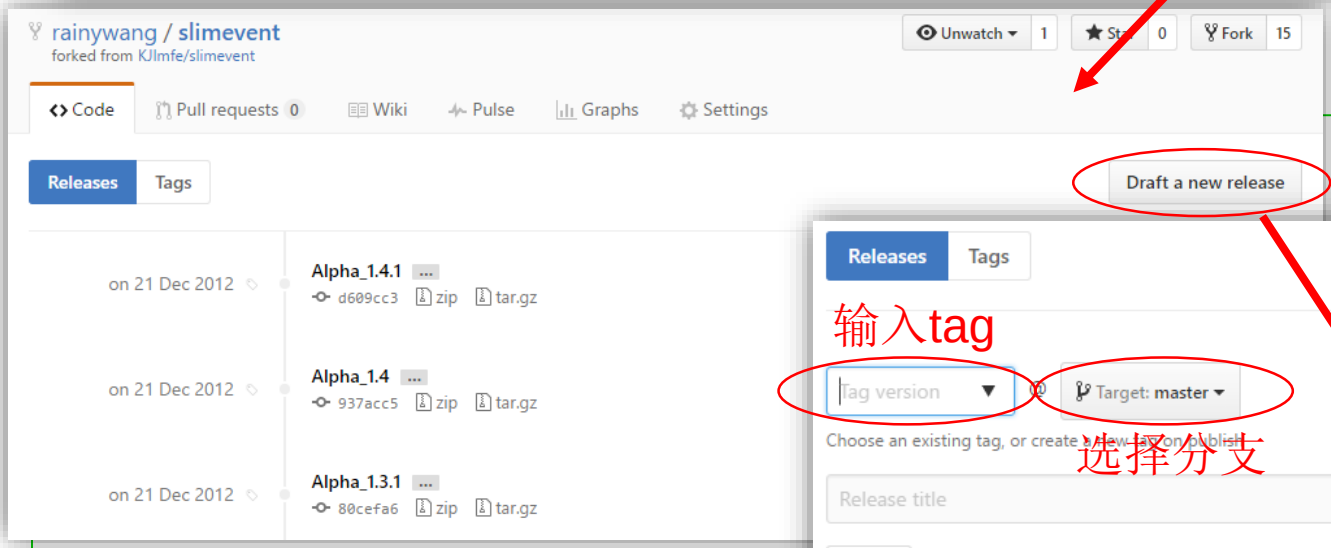
Part 3: 最终答辩(第15周)

- **结题报告 (30页ppt)：**
  - 最终系统的下载、安装和使用方式说明
  - 题目与分组信息
  - 项目进展过程中的团队分工：成员1、2、3
  - 项目计划与实际进展：使用敏捷开发中的燃尽图(Burndown Chart)描述项目的最初计划和项目的实际进展情况
  - Git/GitHub版本历史与协作；
  - 项目的业务特色、技术特色、协作特色、项目管理特色；
  - 当前版本的主要功能；
  - 当前版本所能达到的NFR指标；
  - 当前的真实用户使用情况，以及真实用户评价1、2、3；
  - 项目宣传海报
  - 本次项目收获的经验和教训
  - 对软件工程课实践项目环节的建议

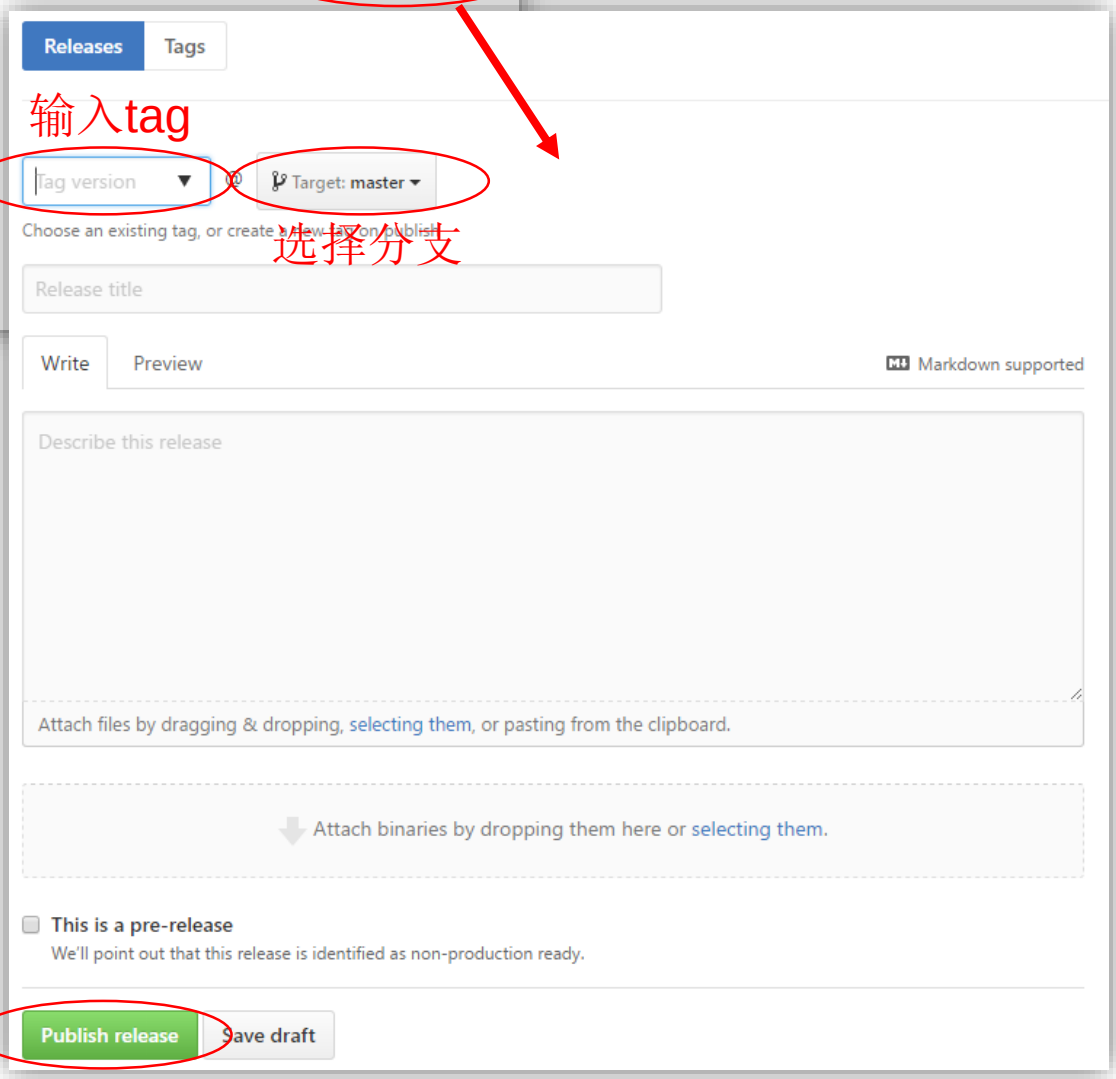
Part 4: 项目管理

- **使用Git/Github管理你的项目**
  - 组长在Github上建立项目repository，各成员以collaborator加入该项目，开题时提交项目URL；
  - 使用项目管理软件编制计划、创建burndown chart，每周将实际进展的burndown chart发布在GitHub项目中；
  - 随项目进展随时提交代码，交叉评审撰写评注。
  - 最终会有两个release：第一轮迭代系统、最终验收系统。





创建Release



发布Release

Part 4: 项目管理

- 通过小组blog整理并公开发布本组的开发经验和教训；
- 只有不断的总结，才有可能的提升；
- 任意blog服务，任意时间、任意主题；
- 言之有物，不要为了获得成绩而写。

Part 4: 项目管理

- **评分标准：**
  - 小组成员提交代码和push到GitHub上的及时性、频繁性、均衡性；
  - 各小组成员所做贡献的均衡性；
  - 小组成员的合作程度；
  - 能够有效利用git commit/branch/release/tag等对项目进展做出准确标识；
  - 使用VersionOne进行项目计划(sprint)的合理性、完整性、准确性；
  - 项目进展(story board、burndown chart)是否正常；
  - 项目小组发布博客的频度与质量。

Part 5: 真实用户

- 开发出的系统要来自真实的需求、被真实用户所使用；
- 各个候选题目中均给出了对真实用户数量的具体要求；
- 从真实用户处获得“真实的需求”，杜绝空想的需求；
- 每次迭代之后，通过真实用户的试用，获得反馈，据此修正系统；

- 通过微博、微信、QQ等SNS渠道宣传、发布你的软件，吸引用户；
- 在校园网、学院内部宣传你的软件，让你的老师/同学成为用户；
- 任课教师/TA会是你的真实用户之一。

- 评分标准：真实用户的数量、来源的多样性、“真实性”、这些用户对项目的评价和建议

候选题目

1 跟着电影去旅行★★★★

- **电影中提及的很多场景(某个城市、某个景点、某个建筑等)，在观众心里留下深刻印象；**
- **“跟着电影去旅行”，亲自到那些场景所在的地点，体验电影情节：**
  - 用户选择特定的电影/电视剧，查询或增加其中包含的地点信息，记录该地点发生的相应“情节”，在Google Maps（或其他地图服务）上进行地点的标注；
  - 利用Google Maps展示某电影中被自己和其他用户标注出的全部地点，或者根据“情节”的关键词来查询地点；
  - 当在地图上点击某个地点时，展示该地点相关的所有电影信息；
  - 用户可以根据兴趣选择某些地点，连接起来形成旅行路线，并可分享到SNS去（例如微信朋友圈、微博等）；
  - 用户可查看其他人分享的路线，并可follow兴趣相投的用户。
- **该系统以Web网站或手机App形式运行。**

2 软件需求的众包市场★★★

- **潜在用户：有开发需求的公司或个人(C)，有软件开发技能的学生(S)；**
- **C方要开发某款软件，但自身缺乏足够的开发团队，需要招聘临时开发人员；**
- **你要开发一个web网站：**
  - C方发布自己的软件需求、对开发者的人数和能力要求、工期要求、报酬预算等；S方查询网站上已发布的各需求，对要参与的需求进行投标；C方获得所有有意向的S方，进行遴选，做出选择决策；系统为双方生成合同；
  - S方发布自己的编程能力及证明材料、可用时间、有意向参与的开发任务类型等，网站据此对其进行推荐(C方已发布的需求)，S方对感兴趣的进行投标；
  - C方发布需求之后，网站也会对其进行S方的推荐；
  - 如果C方发布的需求需要N>1人的开发团队(C需明确对每个角色的人数和能力要求)，则需要等待N个S方投标之后形成团队，方可进行决策。网站可根据当前团队的构成情况对C方进行人员推荐，由C方主动向被推荐的S方发出邀请。

3 网页更新订阅★★★★★

- **潜在用户：对某些网站感兴趣的个人用户；**
- **背景(示例)：**
  - 我对知乎的某个问题及回答感兴趣，但又不想频繁登录知乎网页或App查看该问题下面的回答更新。
  - 我对工大计算机学院的网页发布的通知/新闻感兴趣，但也不想频繁通过历览器查看。
- **开发一个网站，用户输入自己感兴趣的网页URL，系统定时自动追踪这些网页的更新 (识别网页内容的变化)，并把更新内容加以提取，通过email或手机短消息推送至用户。**
- **支持用户对所关注网页URL进行增删改管理，也可暂停/重启已追踪的网页URL；**
- **针对用户关注的全部网页URL，每天特定时间(如20:00)将它们在本日内的全部更新汇聚起来推送给订阅用户。**

4 社交好友分析★★★★★

- **潜在用户：对自己的朋友感兴趣的个人用户；**
- **背景：我对我在某些社交网络上的好友的动态很感兴趣，每天关注他/她的微博/微信朋友圈/知乎/GitHub等，但是我难以充分获得他/她的活动规律；**
- **开发一个Web网站：**
  - 用户选择N个社交网络服务，指定自己关心的社交好友(及其在N个社交网络中的账号/主页等)；
  - 网站自动抽取好友在这N个服务中的更新记录，汇总在一起展示给当前用户；
  - 网站可分析好友在N个服务中的活跃度随时间/日期变化的规律；
  - 网站可分析好友的“兴趣”随日期变化的规律；
  - 网站可感知与该好友具有密切社交互动的其他人，并分析他/她们之间互动的规律(活跃度、主题)等。
  - 当好友发布某些特定主题的内容时，网站可主动推送给当前用户。

5 学术日历与学术地图★★★★

- **计算机领域的新思想层出不穷，技术日新月异；**
  - 计算机领域的研究成果更多在学术会议上发表，而不是传统的学术期刊——学术会议周期短、受众多，影响力扩散快。
- **学术会议多，每个会议的主题侧重不同、重要性不同、时间地点都不同，作为工科IT男(女)，你未来的学术生涯就要在一个个会议之间奔波。何不为自己编制一个“学术日历和学术地图”？**
  - 自动汇集：利用http://www.guide2research.com/topconf、http://idc.hust.edu.cn/~rxli/csrank.htm等会议列表，利用搜索引擎查询各会议当年举行的时间、地点、主题、URL、deadline等；
  - 手工汇集：用户手工录入会议的信息；
  - 用户提供某个学术领域的关键词，系统检索相应的学术会议列表，将日程更新至用户的Google Calendar（或其他web日历、手机日历），通过Google地图（或其他地图）展示选定的会议地点；
  - 用户可关注若干会议，系统自动为其做出各个deadline的提醒。
- **开发一个Web网站或手机App，实现上述需求。**

6 学术师承树★★★★

- **学术也是江湖，师生关系、师兄弟关系构成了学术江湖上的SNS；**
- **开发一个B/S网站，支持学术师承树的构建、查询、维护**
  - 用户可以在网站上建立个人的师承关系树：向上是导师、导师的导师、导师的师兄弟、…；向下是学生、学生的学生、…；
  - 用户也可以在其他人的师承关系树上进行补充（增加某些人）或修正（对错误的关系进行修改删除）；
  - 根据师生关系，可将多棵树连接在一起；
  - 树中唯一的关系类型是“师生关系”，该关系有时间属性（意即在哪个时间段内两人产生师生关系）。
  - 一个人可以有多个导师（多重继承）。
  - 用户输入某个人，查询他的师承关系、学生、师兄弟等等；
  - 当用户点击某个节点时，可连接到LinkedIn或Google Scholar查看该节点人员的职业和publication等信息。

7 目标设定与实现进度监控★★★

- **个人成长过程中需要设定各种长中短期目标，并为目标实现而努力。例如：“2017年12月之前攒钱10000元”、“2018年6月通过英语六级考试”、“大学毕业之前找到女朋友”、“获得保研推免资格”。**
- **目标要有“可实施性”，即能够分解为一个或多个子指标，每个子指标均能量化度量并有时间进度，并可监控其实施进度。例如：“2018年6月通过英语六级考试”，可分解为“背单词数量3000”、“阅读英文文章200篇”、“做模拟考试题30套”等一系列子目标。**
- **开发一个“个人目标管理”软件：**
  - 用户在其中建立自己的预期目标并进行分解，设定量化指标和一系列milestone（特定日期拟达到的子目标值）；
  - 设定目标后的任何日期，用户可随时更新每个子指标的进展情况；
  - 系统展示各目标的实现进度，监控各milestone的实现情况，对落后的子目标/milestone提前做出提醒。
  - 丰富的可视化。

8 文献阅读笔记★★★

- **研究生在做研究过程中要阅读大量的论文，了解他人的工作，获取自己的研究思路。但论文一多，单靠文件系统管理起来很麻烦。**
- **有不少文献管理工具，但你要自己开发一个web系统，支持以下需求：**
  - 要区分未阅读的论文、已精读的论文、已粗读的论文，并对其源文件(文件系统中的PDF、网站链接URL等)进行管理；
  - 用户可建立自己的研究分类树(按研究领域从宽到窄，例如：软件工程→配置管理→版本控制→变化分析)，并将每一篇论文链接到该树的特定节点；
  - 用户可针对某篇论文，记录其阅读笔记；
  - 系统维护用户对论文操作的log(加入系统、分类、已粗读、已精读、记录笔记、删除、更改分类、修改笔记、等)；
  - 针对一篇特定论文，用户可将源文件、阅读笔记、操作log汇集到一起，生成标准文档，并可对外分享；
  - 针对分类树上的某个非叶节点，系统可提取该节点及其所有后代节点附着的论文源文件，打包下载并对外分享。
  - 系统可生成用户的“阅读时间线”。

9 会议嘉宾接送计划编排★★★

- **举行一场会议有若干嘉宾从外地参加，需要对其进行从机场到会场的“接”和“送”。**
- **开发一个Web系统，嘉宾在系统中提供自己的行程（航班、火车等），系统维护可用车辆列表，进而根据嘉宾的行程，编制最佳的接送计划，将接送计划发送给各个参与司机，司机按计划进行接送。**
- **要求：**
  - 各嘉宾无法同时提供行程信息，故可能需要做多次计划编制，在信息不完整的情况下编制计划；
  - 嘉宾可能更改行程，故接送计划也需要随之调整，你的计划编制方法应尽可能降低调整所带来的代价；
  - 航班/火车可能延误或提前，你所编制的计划应具有良好的健壮性；
  - 车辆有容量限制，接送需要时间，在编制计划时要考虑这几个方面的限制；
  - 目标：成本最低、嘉宾的等待时间最短、司机不能过于疲劳

10 个人数据采集★★★★

- **个人每天生活当中随时产生各种个人相关的数据，这些数据如果能被随时记录下来，可形成个人的历史生活轨迹；**
  - 例如：所支付的每一笔钱(日期、金额、用途、支付方式等)、所吃的每一顿饭(日期、地点、吃什么、跟谁吃、钱数)、所加的每一次油(日期、加油站地点、单价、总金额等)
- **开发一个web网站(手机App也许更合适)：**
  - 用户可定制自己感兴趣 (拟采集)的个人数据对象(PDO)及其属性；
  - 当用户要记录数据时，选择之前定制的某个PDO，网站自动生成表单或表格，用户录入，网站存储；
  - 网站支持用户导入Excel格式的个人数据到某个特定的PDO，或者根据Excel的表头自动生成新的PDO
  - 用户可按任何维度查询个人数据，至少可以按时间线、地点作为索引展示个人数据；
  - 用户也可建立起两个PDO所属的数据之间的关系(例如某次吃饭的记录关联着某笔支付的钱)

11 国际合作活动日志★★★

- **HIT建设双一流大学，国际合作日益频繁，大量的校内师生走出去，大量海外学者/留学生来访。学院的国际合作秘书的一项重要工作是记录各种国际合作项目/活动：**
  - 项目：一个人/团队的单次出访/来访(例如CMU AA教授9月1-10日来访)；
  - 活动：一次出访/来访中的一项具体事项(例如AA此次来访由三个活动构成：给学生上课、给研究生做学术报告、与校长会面)
- **开发一个网站，帮助秘书完成记录，面向校内各种国际合作项目/活动：**
  - 国外学者来访、留学生来留学；
  - 校内教师学生出访(短期如参加会议/课程交流、长期如合作研究/联合培养)；
- **在国际合作项目与活动两个层面，完成以下功能：**
  - 事前记录数据(项目信息、日程安排、人员等)，发布通告；
  - 事后记录数据(实际活动信息、实际费用数据等)，撰写总结，发布新闻，材料(照片等)存档。
- **按各种维度查询，做统计分析(时间段、人、项目类别、活动类别等)。**

12 校园预约★★

- **随着MOOC的兴起，传统大学教育的优势一点一点被剥夺，唯一不可被剥夺的优势就剩下了一个：师生之间的面对面互动。**
  - 大学四年，学生与老师有多少在一起的时光？师生间的互动，不应只在课堂上进行。
- **目前的情形是：学生想认识教授们，却不敢，或不知道对方何时可以接见他们。教师们应该主动一点，主动把自己的零散时间开放出来给学生。**
- **开发一个在线预约的服务，以B/S网站形式运行：**
  - 教师针对自己的可用时间，发布若干时间段(比如每半个小时作为一个时间段)，学生可以从中选择适合自己的一个时间段，提交预约请求。
  - 一旦被预约，教师的这个时间段就不能再安排别的事情，但其他未被预约的时间段可以随时调整。
  - 这样的话，教师和学生的时间都可以灵活安排，答疑也就变成了充分利用双方的碎片时间。
  - 再扩展一点，可以让院长/教授们也用它，学生可以预约院长/教授的时间段，与其面对面交流。

13 GitHub小组项目的进展监控★★★★★

- **本需求来自于软件工程课程，各小组的3-4位成员之间进行协作，并在GitHub上维护自己的项目代码/文档；**
- **教师需要实时了解各小组的项目进展情况，希望能够：**
  - 项目的历史提交 (Commits)情况，历次提交的代码量分布、历次提交在时间线上的分布情况；
  - 项目的分支进展情况；
  - 选定特定日期范围，分析各成员的贡献情况，如提交次数、各次提交的代码量、相对贡献率；
  - 各成员之间的协作情况 (例如成员A对成员B的代码做了多少修改)，形成成员协作关系图；
  - 各成员的个性化工作习惯分析 (如提交时间等)；
  - 根据提交的message对每次提交的“目的”进行分析；
  - 将上述各结果以可视化的方式进行展示，并导出为Excel文档。
- **开发一个web网站或一个本地客户端，教师批量录入各小组的GitHub项目URL地址、项目成员的GitHub账号 (以及与学号/姓名的对应关系)，系统通过调用git指令完成上述功能。**

14 自选题目 ★★★★★

- 小组成员自拟题目；
- 但需通过email将自选题目的现实应用背景、实际应用价值、典型需求清单、潜在目标用户、拟开发的软件形态、核心技术等内容写入一个word文档，email发送给任课教师，经批准后方可进入开题准备；
- 自选题目不能与前面的13个候选题目相似；
- 未取得任课教师同意而直接做自选题目，无效/无成绩；
- 每个小班最多2组自选题目。

哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

结束

2017年9月3日