

软件工程se2019

Intro

- 课程主页: <https://github.com/mengning/ase/blob/master/se2019.md>
- 班级博客: http://edu.cnblogs.com/campus/ustc/se2019/join?id=CfDJ8DeHXSeUWr9KtnvAGu7_dX9IMCvo8lX0yoh--X5fjQf2xGCsdVpr97WvYH6ce8OW7E-H0P0DWe7xz_4sU-leaxMINPjaMbdZUz2evpHl9hk-Z345IERSzhBLbRgjYcNQXoI3U2_LSI0lbnYYNpbvokoA
- 主讲教师: 孟宁 mengning@ustc.edu.cn
 - homepage: <http://staff.ustc.edu.cn/~mengning/>
- 助教老师: 杨旭炜 范兴鹏

What you will learn

- Program = data structure + algorithm
- Software =
 Program +
 Software Engineering
- Software Company =
 Software + Business Model

What you will learn

程序员的基本功

- 键盘练习 —— 输入速度、快捷键等
- 代码规范和风格 —— 缩进、命名、注释等
- 测试驱动开发TDD —— 单元测试
- 分解任务Work Breakdown
- 能够熟练操作的工具集 —— VSCode/Vim、正则表达式等

需求分析

- 业务理解与分析
- 原型方法
- 案例The Agile Unified Process: From Analysis to Design

代码的结构、特性和描述方法

- 代码的封装接口
- 设计模式
- 架构风格
- 高内聚低耦合
- 代码结构视图和UML

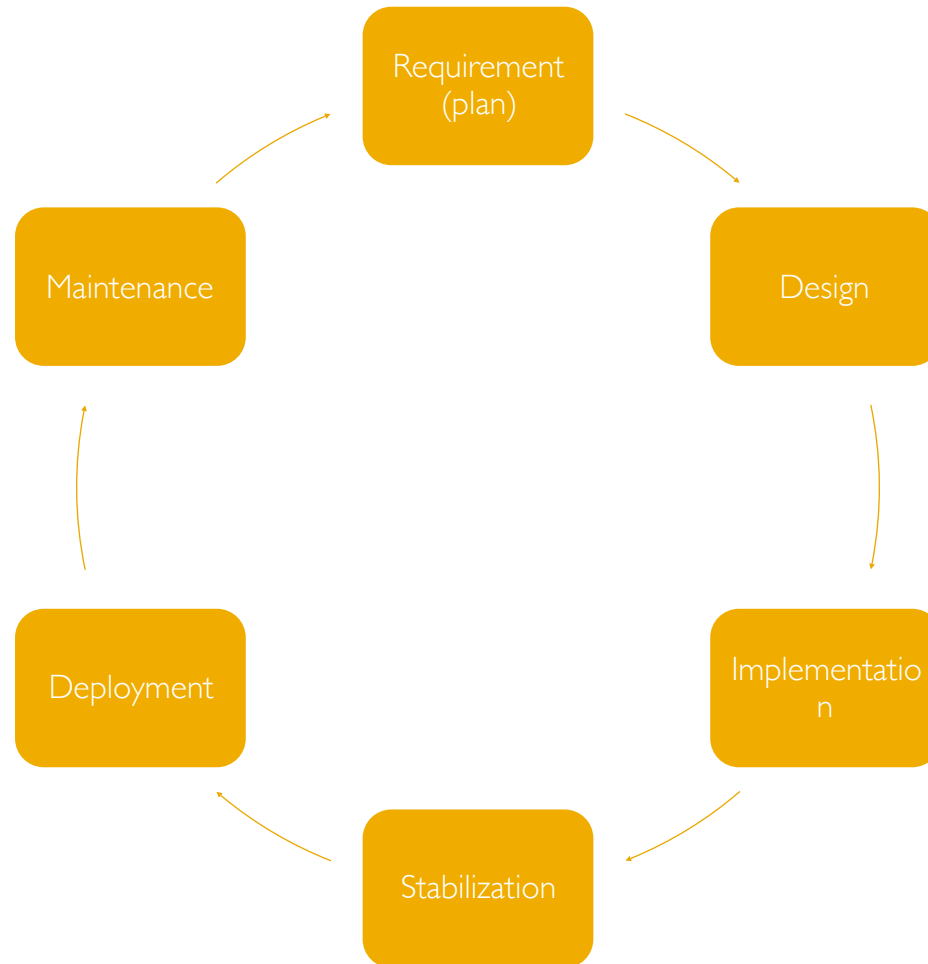
工程过程与项目管理

- 软件的生命周期
- 软件开发过程 —— 敏捷方法、DevOps、CMMI等
- 软件工程质量
- 软件团队
- 软件的交付与维护

这门课怎么学

- 由近及远
- 由浅入深
- 阅读讨论

How SWE is usually taught

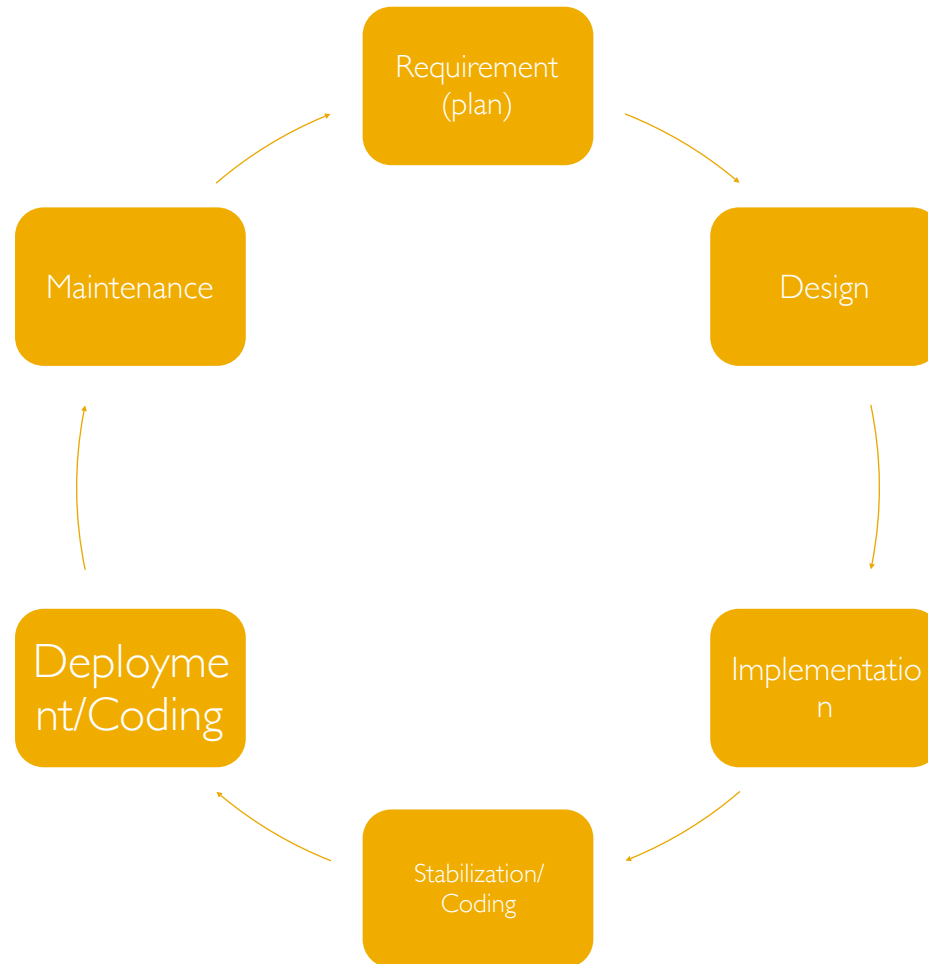


How SE is usually taught

- 需求分析：学生们都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，用设计工具画了各种形状的图形，仅此而已。
- 实现阶段：学生们开始热烈讨论，UML 图早已经扔到一边。
- 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，所有设计过的种种黑箱和白箱测试都无从开始。
- 发布阶段：这个阶段只有一天时间，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！

最后大部分同学们都说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...

How SWE usually happens



How SWE should be taught

- 需求分析：分析现有软件，用你们写的软件的用户量来证明你了解用户的需求
- 设计阶段：用快速发布来证明设计是有效的，能适应变化的。
- 实现阶段：用各种软件工程的衡量手段来证明大家实现的能力。
- 稳定阶段：证明测试能否覆盖代码的大部分。
- 发布阶段：如期发布，用户量，用户评价。
- 维护阶段：网上的观众或下一个年级的同学能很愿意接手你们的软件。

最后大部分同学们能说：自己做了一个有人用，有生命的软件。
然后下个学期，新的一批学生进来提高这一过程...

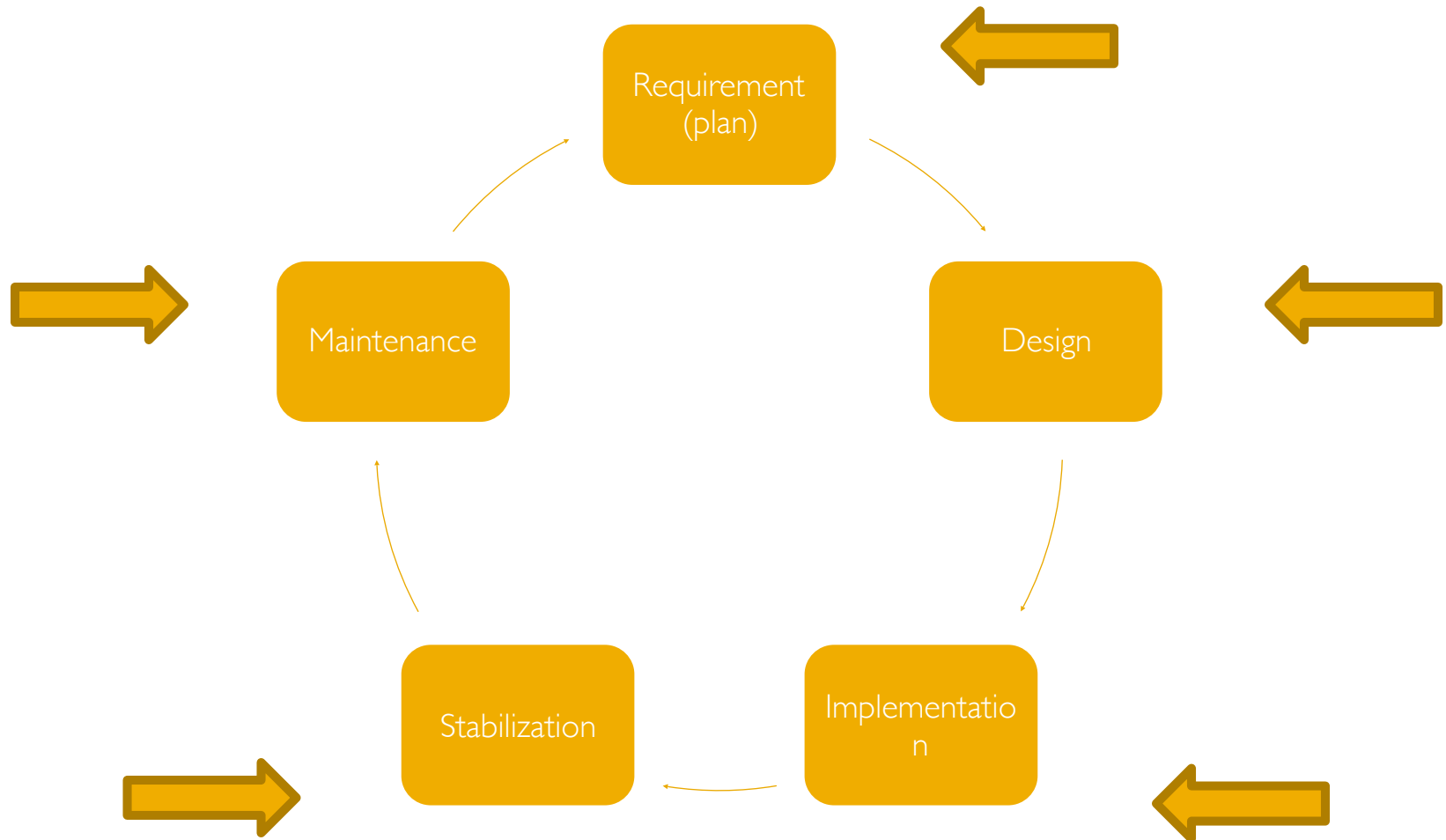
现实 vs. 理想

- 需求分析: 学生们都不懂需求是什么, 上课睡觉。
- 设计阶段: 学生们画了许多 UML 图, 用设计工具画了各种形状的图形, 仅此而已。
- 实现阶段: 学生们开始热烈讨论, UML 图早已经扔到一边。
- 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。
- 发布阶段: 这个阶段只有一天时间, 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!
- 需求分析: 分析现有软件, 归纳初步需求; 基础差的同学学习技术
- 设计阶段: 用快速发布来证明设计是有效的, 能适应变化的。
- 实现阶段: 用各种软件工程的衡量手段来证明大家实现的能力。
- 稳定阶段: 证明测试能否覆盖代码的大部分。
- 发布阶段: 如期发布, 用户量, 用户评价。
- 维护阶段: 网上的观众或下一个年级的同学能很愿意接手你们的软件。

I want to teach it this way

1. 根据工程实践选题找一套代码为基础开始维护程序，理解程序。
2. 找bug，改bug，重构小部分代码，以满足用户的需求。
3. 一部分同学可以开发测试用例
4. 在现有版本的基础上做增量开发
 - a) 理解需求（这个时候理解了客户需求是什么）
 - b) 设计
 - c) 开发
 - d) 回归测试（用到上面开发的测试用例来保证质量）

Which means:



Emphasis

- Learning by doing
 - 在工作中学习 – 做中学，在项目中有针对性的训练程序员的基本功
 - Focus on real user benefit
- “教育是令人羡慕的东西，但是要记住：凡是值得知道的，没有一个是能够教会的。
——奥斯卡·王尔德 (Oscar Wilder)
- Learn from mistakes
 - Just do it!
- Collaboration with peer
 - I:I
 - I:many



- About ancient Roman soldier, which of the following is correct?
- a) No training, they were thrown into real battles right away
 - b) They only learn theory, don't touch weapon
 - c) They use fake weapon, half the weight of real weapon
 - d) weapon has **same** weight
 - e) weapon has **twice** the weight

为啥那么多动手的练习？

- 因为以前动手太少
- 没有量的积累，不会有进一步讨论和探索的基础，也没有掌握理论的条件。

陶艺课的故事，把学陶艺课的学生分成两组，一组以量评分（只看数量），一组以质评分（只看质量）。一个学期后，真正好的陶艺作品出现在哪里？

陶艺课

一个学期后，真正好的陶艺作品反而是出现在那以量评分的组里。

- 要敢于开始
- 所以，请停止抱怨自己实力的不济和各种客观条件的不好，停止对开始的恐惧。只要迈出了第一步，埋着头认真做，不停的尝试，重复，失败在尝试，只要不放弃，就一定能成功的。
- 阅读：

<http://www.cnblogs.com/codingcrazy/archive/2011/02/28/1967503.html>



为啥要做难的项目？

- 请先看视频

- [http://weibo.com/tv/v/F6Udt9Say?
fid=1034:8003d777aa3f955d0a2d22dfe63bf02e](http://weibo.com/tv/v/F6Udt9Say?fid=1034:8003d777aa3f955d0a2d22dfe63bf02e)

- “老师， 为什么这些软件工程技术在我的项目中没有什么用？”

- 因为你的项目水太浅！

这个课要花多少时间？我基础差...

- 请看这个博客，和下面的回答
 - <http://www.cnblogs.com/huangxman/p/5868204.html>
- 学校的软件工程课有标准，老师不能降低标准，否则就是教学事故。
- 编码训练、新技术学习、上课、阅读、讨论、写作、参与工程实践、用软件产品改变世界！

Text Books

■ 课本

- 构建之法 – 现代软件工程
- 《软件工程——理论与实践》

■ 必读书

- (必读) 《梦断代码》 <http://bestcbooks.com/dreamingincode/>
- (必读) 《人月神话》 Frederick Brooks

■ 其它参考书

- <http://book.douban.com/doulist/1204928/>

怎么才能证明学会了？

- 研发出符合用户需求的软件
 - 必须公开发布，有实际的用户，一定的用户量和持续使用量（3天后能保持10 - 100个用户）
 - 可以是PC/Web/手机应用/源代码模块等
 - **而不是**: 做没有用户使用的软件
- 通过一系列工具，流程，团队合作，能够在预计的时间内发布“足够好”的软件
 - 有项目规划/需求/设计/实现/发布/维护
 - 有定时的进度发布
 - **而不是**: 通过临时熬夜，胡乱拼凑，大牛一人代劳，延迟交付等方式糊弄
- 并且通过数据展现软件是可以维护和继续发展的。
 - **而不是** 找不到源代码，代码无文档，代码不能编译，没有task/bug等项目发展的资料
- 能做到这三点，就是学会了软件工程。

这个课相当于技术培训么？

- 不是！

大学教育 vs. IT行业培训

■ 大学：

- Preparing students for jobs that do not exist, using technologies that have not been invented, in order to solve problems we do not even know are problems. (Karl Fish)
- **课堂练习：请翻译上面这句话**
- 目的：打下基础，培养自我学习、探索的技能，为将来的未知挑战做准备。

■ IT 行业培训

- 用已有的技术解决现有的问题，获得实用技能
- 目的：人才马上能用

■ 大学当然可以讲目前流行的技术，解决目前的问题

- 目的是要知其然，并知其所以然，在其中锻炼自我学习， 探索的能力。
- 要不断提问， 不断深入探索

■ 讨论：很多学校的计算机课程有很多“打代码”的练习

- 学生上机就是把代码输入，通过测试，下机
- 这是“大学教育” 还是 “短期IT 行业培训”？
- 这种练习的效果如何？

Student Score 分数组成

- 考试成绩50%
- 随堂测试40%
- 平时成绩（阅读讨论写作开发等）10%，默认全部0分，只留给某方面作出杰出表现的同学，鼓励自主自愿投入，由内部动机带来的思考和成果。

对学生的期望

■ 主动

- 主动问问题，发博客描述自己不懂的问题
- 主动动手做事，记录犯错误的过程，从错误中学习
 - 在理论上，理论和实践是一样的
 - 在实践中，它们却不一样
- 给老师，助教直接的反馈

■ 坚持

- 很多人上这门课开始很有兴趣，后来觉得太累，最后放弃
- 坚持到最后，就能获得成就
 - 很多世界著名的MOOC 公开课，坚持到最后的学生只占 ~5%

■ 用结果说话

- 写了多少行代码，多少博客
- 做的项目有多少用户，多少好评
- 上课之初的疑问自己能回答多少？

“loser” 矛盾的行为和期望

- 父母，既看不起自己的孩子，又希望孩子出人头地；
- 孩子，既厌恶父母设计的人生，又怕走错路辜负了父母的期望；
- 老师，既要求学生乖乖听话，又希望学生有创新精神；
- 学生，既不认同老师的观点，又怕得不到那鸡肋的分数。

如何看待作业中的各种失败？

- 很多同学害怕失败，纠结于某个作业的分
- 请把这门课看作是去游乐场玩耍
→
 - 不要纠结于一时的失误
 - 玩尽可能多的游戏
 - 希望最后还能学到点东西 😊

就像迪斯尼乐园，所有的小孩子即使摔倒了也不会哭。为什么呢，因为玩都来不及，要赶快爬起来去玩。如果你把自己的人生当作游乐场，你跌倒了就不会哭。

Logistics

- Create accounts in source code control system
 - It could be GitHub...
- Decide where to write blogs
 - 默认选择 (www.cnblogs.com)
 - 加入班级博客: http://edu.cnblogs.com/campus/ustc/se2019/join?id=CfDj8DeHXSeUWr9KtnvAGu7_dX9IMCvo8IX0yoh--X5fjQf2xGCsdVpr97WyH6ce8OW7E-H0P0DWe7xz_4sU-leaxMINPjaMbdZUz2evpHI9hk-Z345IERSzhBLbRgjYcNQXoI3U2_LSIOlbnYYNpbvokoA
 - 【所有作业都通过该博客系统提交, 请务必严格遵守作业截止时间】

可以选择的作业/练习

- <http://www.cnblogs.com/xinz/p/3803035.html>
- 《C编码实践篇》 <http://mooc.study.163.com/course/USTC-1000002006>
- Vue.js及其应用开发
- 除了班级博客上发布的作业必做外，不必做完所有的作业，您可以自由选择