

Lecture1 Introduction

1. 课程介绍

课程组织

本课程着重于提供**设计和开发大型软件系统的实践经验**，并强调**自动化工具和技术**的使用

- 推荐先修课程：CS309 面向对象设计与分析

教材和参考书

- Ian Sommerville, Software Engineering
- Freeman et al., Head First Design Patterns
- Block, Effective Java
- Zeller and Krinke, Essential Open Source Toolset: Programming with Eclipse, JUnit, CVS, Bugzilla, Ant, Tcl/TX and More
- McConnell, Code Complete: A Practical Handbook of Software Construction
- Barrett, Linux Pocket Guide
- Pilone, UML 2.0 Pocket Reference

课程考核与评价

- **Lab - 20% ($\geq 10\%$)**
- **Project - 35%**
- **Exam - 35%**
 - 期末考试（课程的任何方面，包括 lab 的部分）
- **In-Class Exercise/Attendance - 10%**

Project 说明

- 开发一些真正的软件
 - 修复GitHub 上的 Java 开源项目的 issue
 - 4-6 人/组
 - 2 份书写报告
 - 3 次 presentation （前期，中期，后期）
 - 同伴评价
- 交付成果
 - Proposal - 第四周
 - Progress Report - 大概是学期中
 - Final Presentation and Report - 学期末
- 注意
 - 必须记录下使用的过程
 - 一定要说服我们你遵循了你记录下的过程

其它说明

GitHub Classroom

- 分享你发现的bug（注意说明你是在什么时候通过 GitHub 讨论中发现一个 bug 的）
- 互相帮助安装工具

Instructor: Shin Hwei Tan

- Email: tansh3@sustech.edu.cn
 - 如果要发邮件，在邮件名前面包含 **se-sustech**
 - 用英语书写邮件
- Office: 工学院南楼612
- Office hours: 上课结束后或预约

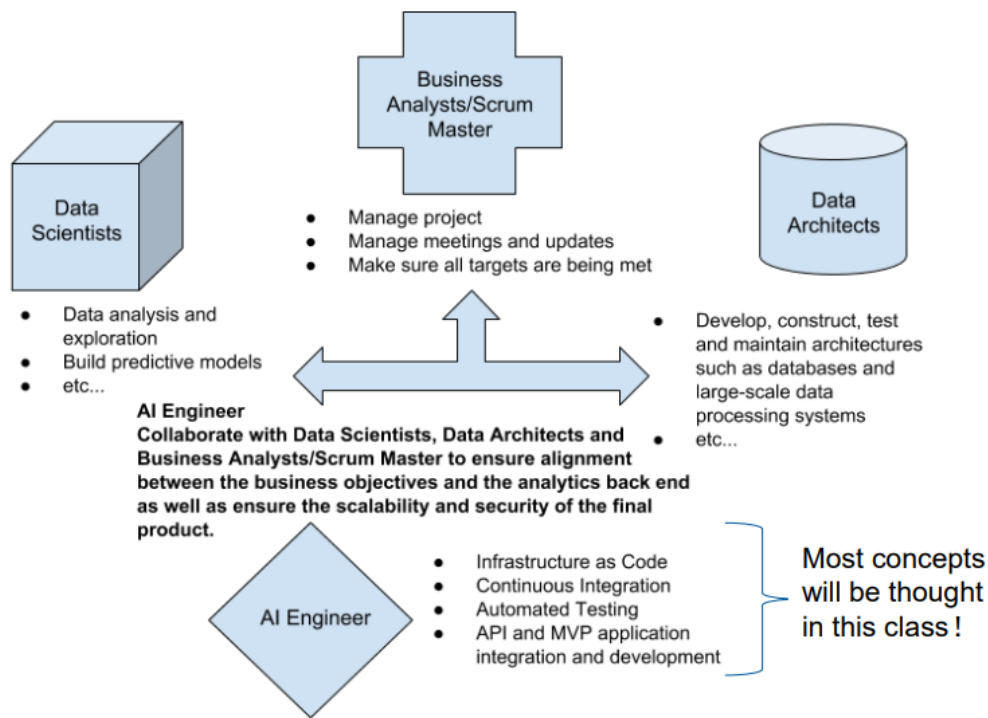
2. 软件工程介绍

为什么要学习软件工程

可以学习下面的知识

- 与其他人在团队中协作
- 保持最新的技术
- 合理的判断与决策
- 定位，组织和评价信息
- 口头交流
- 统计数据
- 写作交流
- 批判性思考
- 有创造力
- 分析复杂的问题
- 应用知识/技能到真正的世界中去
- 知道不同的文化区别
- 与不同背景、知识的人共同工作
- 对全球发展保持最新的了解

什么是 AI 软件工程师



与数据科学家、数据架构师和业务分析师合作，确保业务目标和分析后端之间的一致性，并确保最终产品的可扩展性和安全性

- 基础代码架构
- 持续集成
- 自动化测试
- API 和 MVP 应用程序集成和开发

软件工程涉及的主题

- 过程 Process
- 工具 Tool
- 技术 Techniques
- 模型 Models (软件的开发)
- 建模 Modeling (开发的系统)

什么不是软件过程

- 不只是开发软件
 - 个人 vs. 团队配合
- 不只是一个流程
 - 软件工程是研究几种不同过程的领域

什么是软件工程

The application of a **systematic, disciplined, quantifiable** approach to the **development, operation, and maintenance** of software. (IEEE 610)

3. 软件过程 Software Process

A set of activities performed towards a specific purpose (IEEE 1074)

The steps a particular group follows to develop software (Johnson)

所有的团队都遵循相同的过程：**XP（极限编程 Extreme Programming）**

有很多的软件过程

- 敏捷的 Agile
 - XP, Scrum
- 理论性的 Theoretical
 - Waterfall
- 正式的 Formal
 - Rational Unified Process (RUP) 统一软件开发过程
 - Cleanroom
- 分布式的，开源的 Distributed, open-source
 - Bazaar

Chicken & the Pig 寓言

A Pig and a Chicken are walking down the road.

The Chicken says: "Hey Pig, I was thinking we should open a restaurant!"

Pig replies: "Hm, maybe, what would we call it?"

The Chicken responds: "How about 'ham-n-eggs'?"

The Pig thinks for a moment and says: "No thanks. I'd be committed, but you'd only be involved."

The Chicken is involved, but the Pig is committed

该寓言被 Scrum 框架引用来说明两种类型的项目成员

- 猪，他们完全致力于项目并对其结果负责
- 鸡，他们咨询项目并被告知其进度

这个类比是基于猪提供培根的能力（一种需要猪死亡的祭品）与鸡提供鸡蛋的能力（非牺牲）

对于 Scrum 项目

- 开发团队，产品负责人和 Scrum Master 被认为是致力于项目的人
- 而利益相关者，客户和执行管理层被视为参与但不致力于项目的人

XP

- XP 包含的人员：客户、开发者、教练
- Scrum
 - 猪（开发团队，产品负责人和 Scrum Master）
 - 鸡（客户和执行管理层）

- Activities
 - 写策划 Write Stories
 - 规划策略 The Planning Game
 - 测试优先 Test-First
 - 结对编程 Pair Programming
 - 持续集成 Continuous Integration
 - 重构 Refactoring

- 规划策略(The Planning Game);
- 结对编程(Pair programming)
- 测试(Testing)
- 重构(Refactoring)
- 简单设计(Simple Design)
- 代码集体所有权(Collective Code Ownership)
- 持续集成(Continuous Integration)
- 现场客户(On-site Customer)
- 小型发布 (Small Release)
- 每周40小时工作制 (40-hour Week)
- 编码规范 (Code Standards)
- 系统隐喻 (System Metaphor)

Activities 在 IEEE 中的意思

- **项目管理 Project Management**
 - 项目启动 Project initiation
 - 项目监控 Project monitoring and control
 - 软件质量管理 Software quality management
- **开发 Development**
 - 需求 Requirements
 - 设计 Design
 - 实现 Implementation
- **后开发 Post-Development**
 - 安装 Installation
 - 操作和支持 Operation and support
 - 维护 Maintenance
 - 停止 Retirement
- **完整过程 Integral processes**
 - 检验和确认 Verification and validation
 - 软件配置管理 Software configuration management
 - 文档开发 Documentation development

4. 如何选择一个开源项目 / issue

[Search · hacktoberfest \(github.com\)](#)

一个好的项目的特点

Hacktoberfest is a month-long celebration of open source software.
[See topic](#)

5,561 repository results Sort: Most stars ▼

iluwatar/java-design-patterns Sponsor
Design patterns implemented in Java
[java](#) [design-patterns](#) [snippets](#) [principles](#) [awesome-list](#) [hacktoberfest](#) [snippets-library](#)
[snippets-collection](#)
☆ 73k ● Java Updated 2 days ago

TheAlgorithms/Java Sponsor
All Algorithms implemented in Java
[algorithms](#) [data-structures](#) [java](#) [hacktoberfest](#) [algorithm](#) [search](#) [sort](#) [sorting-algorithms](#)
[algorithm-challenges](#) [algorithms-datastructures](#)
☆ 44.3k ● Java MIT license Updated yesterday 2 issues need help

halo-dev/halo
一款现代化的开源博客/CMS系统。
[blog](#) [cms](#) [blog-engine](#) [halo](#) [hacktoberfest](#) [halocms](#)
☆ 20.9k ● Java GPL-3.0 license Updated 21 hours ago 4 issues need help

alibaba/Sentinel
A powerful flow control component enabling reliability, resilience and monitoring for microservices. (面向云原生微服务的高可用流控防...)
[java](#) [microservice](#) [reliability](#) [rate-limiting](#) [circuit-breaker](#) [alibaba](#) [hacktoberfest](#)
☆ 18.5k ● Java Apache-2.0 license Updated yesterday **14 issues need help**

jenkinsci/jenkins Sponsor
Jenkins automation server
[jenkins](#) [continuous-delivery](#) [cicd](#) [devops](#) [java](#) [groovy](#) [pipelines-as-code](#) [continuous-integration](#)
[continuous-deployment](#) [hacktoberfest](#)
☆ 18.4k ● Java MIT license Updated 4 hours ago

- 在 hacktoberfest 的名单里面
- 有很多的 stars
- 最近有更新
- 很多需要修复的 issue

一个好的 issue 的特点

Pinned issues

Wanted: Who is using Sentinel
#18 opened on 1 Aug 2018 by sczyh30
Open 93

How to contribute | 如何参与开源贡献
#391 opened on 7 Jan 2019 by sczyh30
Open

[IMPORTANT] Log4j CVE-2021-44228 漏洞影响说明
#2490 opened on 14 Dec 2021 by sczyh30
Open

Filters Labels 59 Milestones 2 New issue

364 Open ✓ 1,355 Closed

Author Label Projects Milestones Assignee Sort

记一个 MetricFetcher: 在多用户服务8719 端口连接不通导致实时监控没有数据问题 [area/dashboard](#)
#2566 opened 23 hours ago by su787910081

基于ConcurrentLinkedHashMap实现的CacheMap会导致在高CPU核数下, 单个接口产生的ParameterMetric对象很大的问题 [area/performance](#) [good first issue](#)
#2555 opened yesterday by LeoMe1

Spring Cloud Gateway 链路链路路径错乱, 没有展示为URL [area/gateway-flow-control](#) [kind/question](#)
#2561 opened 4 days ago by feng1632009

Global circuit breaker for all Dubbo RPC services | 如何默认将所有的 Dubbo RPC接口加上全局默认的熔断策略, 并且忽略指定的业务异常? [area/circuit-breaking](#) [kind/question](#)
#2557 opened 16 days ago by Hachiko-JQ

Circuit breaking by the exception ratio not support threshold equals 1 [area/circuit-breaking](#)
#2554 opened 18 days ago by JasonKai9527

Adapters for MySQL client | 有考虑 Sentinel 提供与 Mysql Java 整合吗? [area/integrations](#)
1 1

- 有 `Good First Issue` 的标签 (适合刚开始的人)
- 不是问题/文档
- 没有标识 `Pinned` (通常意味着开发人员已经计划解决这个问题, 所以当你为你的项目工作时, 这个问题已经解决了)

如何开始对一个开源项目进行贡献

<https://github.com/alibaba/Sentinel/wiki/%E5%BC%80%E6%BA%90%E8%B4%A1%E7%8C%AE%E6%8C%87%E5%8D%97%E5%88%9B%E5%BB%BA-issue-pr>

在你开始解决一个 issue 之前

- 构建整个项目
- 运行测试样例
- 说明贡献指引
- 签署 (贡献者许可协议 Contributor License Agreement) CLA

在着手开始解决 issue 时

- 从“good first issue”或开发者说“很简单”的问题开始
- 阅读问题, 问问题, 告诉开发人员“你好, 我有兴趣贡献。我能解决这个问题吗?”
- 添加测试 (如果不可用)
- 如果开发人员回复了, 请与开发人员讨论您计划的解决方案

一个示例

<https://github.com/JabRef/jabref/issues/6198>

GitHub workflow



leitianjian commented on 16 Apr 2020 • edited

Contributor



Hi, we want to work on this bug, but it seems difficult to fix it.

Firstly, we have reproduced the bug successfully.

Then, thanks for your tips, we successfully understood the source of the bug

Secondly, we tried to solve it by only store the name instead of style in method
JabRefPreferences.storePreviewPreferences(PreviewPreferences previewPreferences)

1. we add a field called previewStyleName in class PreviewPreferences
2. modify the method JabRefPreferences.storePreviewPreferences(PreviewPreferences previewPreferences) to put PREVIEW_STYLE_NAME attribute into preferences
3. try to modify method JabRefPreferences.getPreviewPreferences() to search style by name.
4. we found a method PreviewTabViewModel.findLayoutByName(String name) which can convert the name to style.
5. We try to transfer the field PreviewTabViewModel.availableListProperty, which is used to convert name to style, to the JabRefPreferences.
6. point 5 is a silly idea, because the PreviewTabViewModel is relied on singleton JabRefPreferences. If we pass the PreviewTabViewModel.availableListProperty to the JabRefPreferences can lead to a big problem.
7. The solution we think now is to construct the availableListProperty in the JabRefPreferences, but this solution is not very suitable we thought, because the availableListProperty is not suitable to be included in Preferences, which store setting only.

As a result, we are sticking at this bug without a good solution.

We sincerely waiting for your ideas. It is hard for us to solve this bugs without more info.

It is OK for late reply, we might working on other works recently, please forgive our late reply.

It is OK if you don't think we could working on this bug, we will try something else easier.

In advance, many thanks for your any instructions/tips.

要解决多少个 issue?

每个人一个学期解决两个

- 一个在 4 月 25 号之前完成 (Progress Report)
- 一个在 5 月 16 号之前完成 (Final Presentation)

5. Git 简单入门

Branching & Merging

<https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging>

Branching git 操作

你已经决定要在公司使用的任何问题跟踪系统中处理第53个 issue

要创建一个新的分支并同时切换到它，你可以使用 -b 分支运行 git checkout 命令

```
1 $ 切换并创建一个新的分支，名叫"iss53"
2 git checkout -b iss53;
```

上面的命令是下面两个命令的简写

```
1 git branch iss53
2 git checkout iss53
```

Merging git 操作

假如你已经完成了第53的 issue 的问题并且决定把它 merge 到 `master` 分支中

为了完成这件事，你必须将 `iss53` 分支 merge 到 `master` 中

你需要做的事情是检查你想要 merge 的分支然后运行 git merge 命令

```
1 $ 切换到分支"master"
2 git checkout master
3 $ merge iss53
4 git merge iss53
```

Submodules

<https://gist.github.com/gitaarik/8735255>

在 git 中，你可以添加一个子模块进你的仓库，它就像是一个嵌入到你的主仓库中的子仓库

使用 submodule 的好处

- 可以将代码分离到不同的仓库中
 - 如果你有一个包含大组件的代码库，这很有用，你可以把一个组件变成一个子模块
 - 这样你就有了一个更干净的 Git 日志(提交是特定于某个组件的)
- 你可以将子模块添加到多个仓库去
 - 如果有多个共享相同组件的仓库，通过这种方法，可以轻松地更新所有仓库中作为子模块添加的组件，这比将代码复制粘贴到存储库中要方便得多。
- 适用于如果你们小组选择修复**两个及以上**的开源项目

git 操作

假设你正在做一个叫做 Slingshot 的项目，与此同时，在另一个库中，您有另一个名为 Rock 的项目——它只是一个通用的 Rock 库，但您认为它对于 Slingshot 来说是完美的

你可以添加 Rock 作为 Slingshot 的子模块，在 Slingshot 的仓库中：

```
1 git submodule add https://github.com/<user>/rock rock
```

此时，你将在 Slingshot 中拥有一个 Rock 文件夹，但如果你想查看这个文件夹，根据你的 Git 版本，你可能会看到，什么都没有

更新版本的 Git 会自动完成这个操作，但老版本会要求你明确地告诉 Git 下载 Rock 的内容

```
1 git submodule update --init --recursive
```

如果一切正常，你可以提交这个更改，你将在 Slingshot 仓库中拥有一个 Rock 文件夹，其中共享了 Rock 存储库中的所有内容

<https://github.blog/2016-02-01-working-with-submodules/>

jaw6 Add rock as submodule	
rock @ 371eec0	Add rock as submodule
rubber-band	Begin writing rubber-band
y-shaped-stick	Begin writing y-shaped-stick
.gitmodules	Add rock as submodule