

高级软件工程

第二周 (Sep. 13)

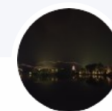
主讲：罗铁坚

助教：周文璋、俞永生、姚敏

上课：周一、三 上午 10:30—12:10

答疑：周一、三 下午 2:30—3:30 (学园2-485)

联系：tjluo@ucas.ac.cn 69671829



2021高级软件工程

群号：544158863



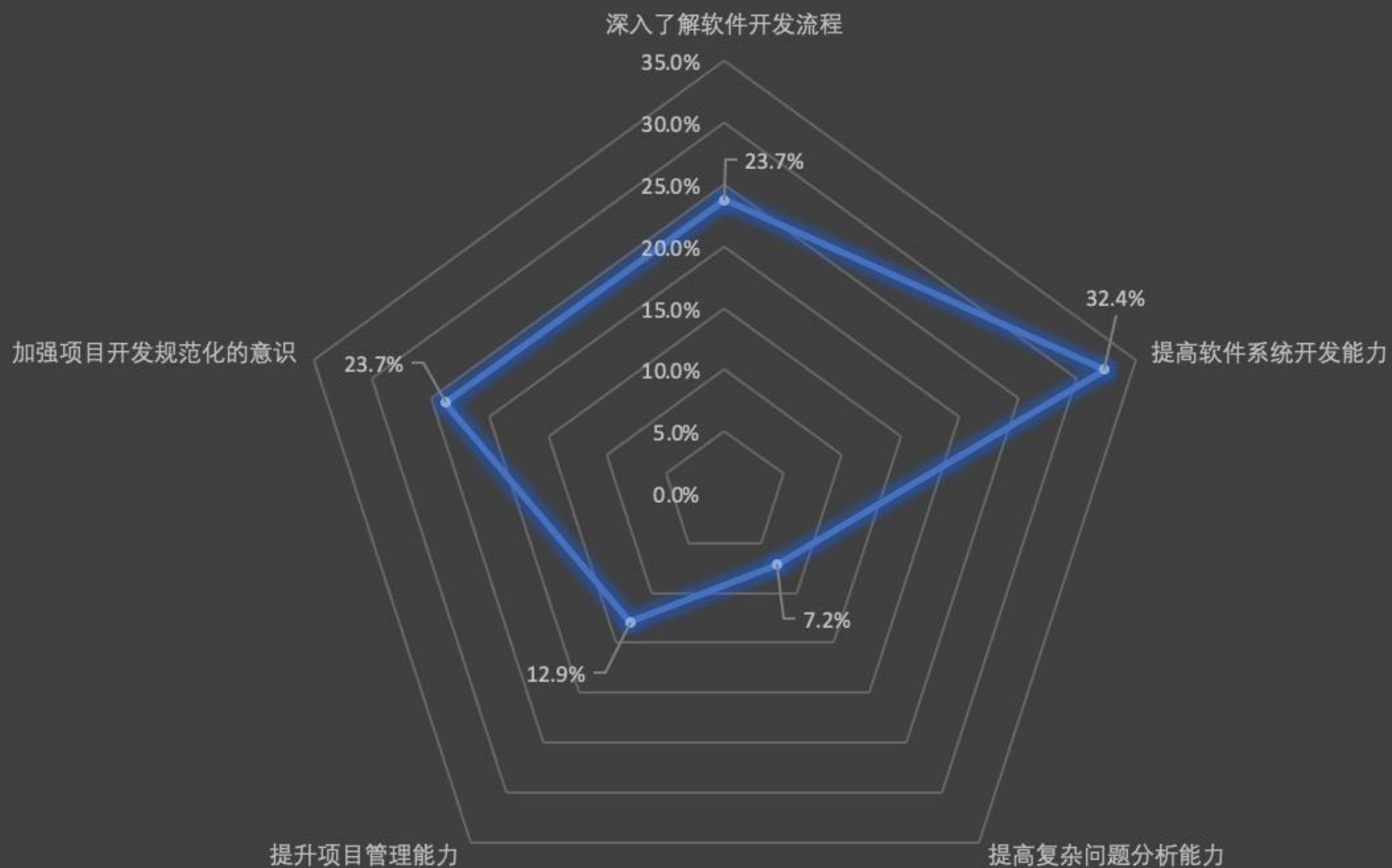
扫一扫二维码，加入群聊。



提纲

- 上周随堂练习点评
- 教学计划简介
- 2020年课程项目总结
- 魔方项目的演示与讨论
- 软件架构及开发模式

课程预期效果和目的



你认为设计和实现一个有价值的软件 难吗



有没有用过git/github?



如果有难度主要在什么方面



提纲

- 上周随堂练习点评
- 教学计划简介
- 2020年课程项目总结
- 魔方项目的演示与讨论
- 软件架构及开发模式

教学计划

大纲章次	章名称	章学时	授课教师	大纲小节次	小节名称	节学时
第1章	引言(第1周)	4		1.1	学科发展动机	1.0
				1.2	预期学习效果	1.0
				1.3	知识能力及学习策略	1.0
				1.4	课程目的和考核要求	1.0
第2章	研究主题（第2周）	4		2.1	理论奠基者	0.8
				2.2	三类问题	0.8
				2.3	二种途径	0.8
				2.4	九项任务	0.8
				2.5	练习与问题	0.8
第3章	领域问题和应用场景（第3、4周）	8		3.1	讨论重点	1.1
				3.2	案例1 操作系统和应用框架（Linux and Ruby on Rail）	1.1
				3.3	案例2 交互程序设计语言环境（Jupyter notebook）	1.1
				3.4	案例3 玩魔方软件（Web 版与实体版）	1.1
				3.5	案例4 池塘大战软件（智能体博弈）	1.1
				3.6	案例5 软件版本管理与持续集成平台（GitHub等）	1.1
				3.7	练习与问题	1.1
第4章	业务建模和数据模型（第5、6周）	8		4.1	讨论重点	1.6
				4.2	领域知识与建模语言	1.6
				4.3	业务数据的外在形式与计算机的内部表示	1.6
				4.4	数据表示与应用功能	1.6
				4.5	练习与问题	1.6
第5章	软件架构和应用框架（第7、8周）	8		5.1	讨论重点	1.6
				5.2	架构与功能	1.6
				5.3	架构与数据	1.6
				5.4	抽象共性与知识重用	1.6
				5.5	练习与问题	1.6

教学计划

第6章	设计模式和代码重构（第9周）	4
第7章	用户体验和接口设计（第10周）	4
第8章	软件验证与自动测试（第11周）	4
第9章	系统安全和防御设计（第12周）	4
第10章	规模应用与性能评价（第13周）	4
第11章	集成部署和运营服务（第14周）	4

罗铁坚	6.1	讨论重点	0.8
	6.2	共性与特殊功能的分离	0.8
	6.3	应对功能变化的策略	0.8
	6.4	案例分析	0.8
	6.5	练习与问题	0.8
	7.1	讨论重点	0.8
	7.2	用户体验设计原则	0.8
	7.3	信息表示的目的与方法	0.8
	7.4	选择合适的可视化形式	0.8
	7.5	练习与问题	0.8
	8.1	讨论重点	0.8
	8.2	程序正确性证明	0.8
	8.3	软件验证方法	0.8
	8.4	构造测试用例	0.8
	8.5	练习与问题	0.8
	9.1	讨论重点	1.0
	9.2	软件安全的原因	1.0
	9.3	防御措施的建构	1.0
	9.4	练习与问题	1.0
	10.1	讨论重点	1.0
	10.2	观察性能指标	1.0
	10.3	提升应用性能	1.0
	10.4	练习与问题	1.0
	11.1	讨论重点	1.0
	11.2	持续集成与部署	1.0
	11.3	软件上线与服务模式	1.0
	11.4	练习与问题	1.0

提纲

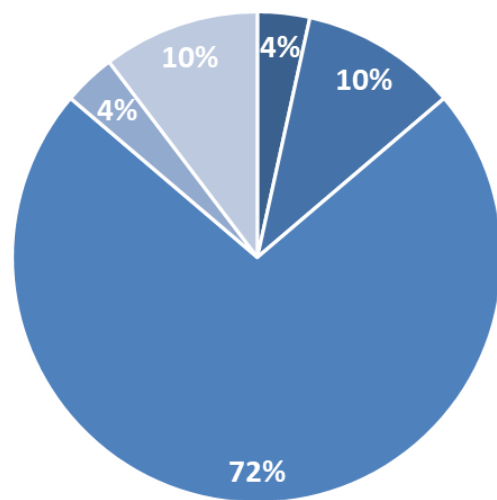
- 上周随堂练习点评
- 教学计划简介
- 2020年课程项目总结
- 魔方项目的演示与讨论
- 软件架构及开发模式

组队选题情况

选课人数共89人，共组成29组

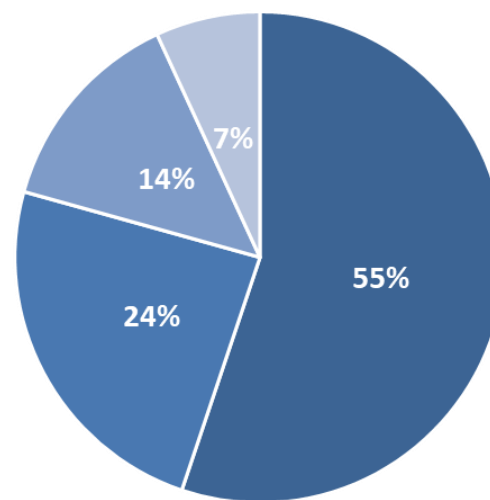
- 3个5人组；1个4人组；21个3人组；3个2人组；1个1人组。
- 16组选题为聊天机器人（共49人）；7组选题为魔方（共20人）；4组选题为选课系统（共12人）；2组选题为自选题目（共8人）。

分组情况



■ 1人组 ■ 2人组 ■ 3人组 ■ 4人组 ■ 5人组

选题情况



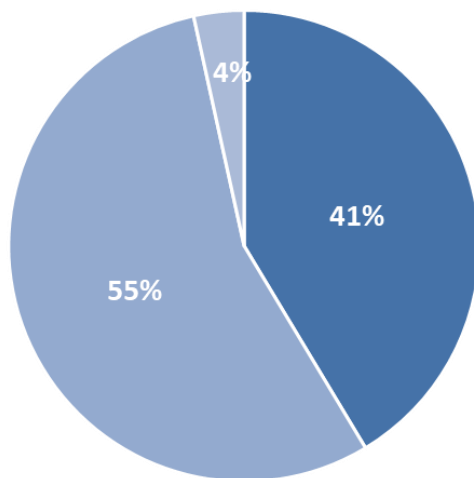
■ 聊天机器人 ■ 魔方 ■ 选课系统 ■ 自选题目

项目完成情况

选课人数共89人，共组成29组

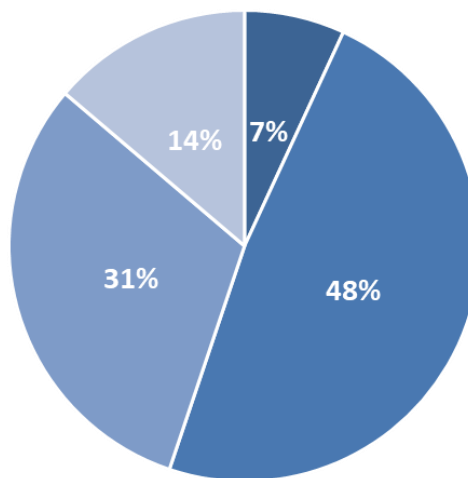
- 16组使用Python；12组使用Ruby；1组使用Java。
- 结题文档10页以下 2组；10-20页14组；20-30页9组；30页以上4组。
- 答辩完成后共24组根据老师意见进行了修改和反馈。

开发语言



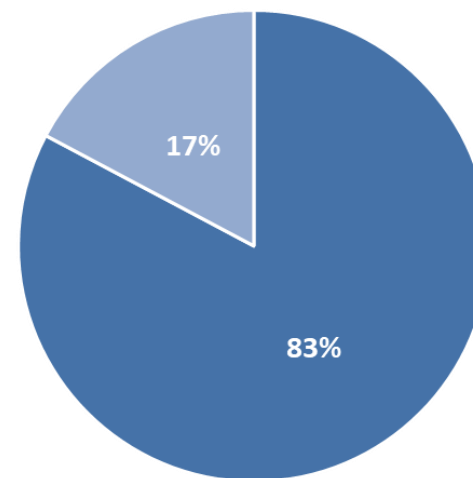
■ Ruby ■ Python ■ Java

文档页数



■ 10页以下 ■ 10-20页 ■ 20-30页 ■ 30页以上

答辩后是否修改

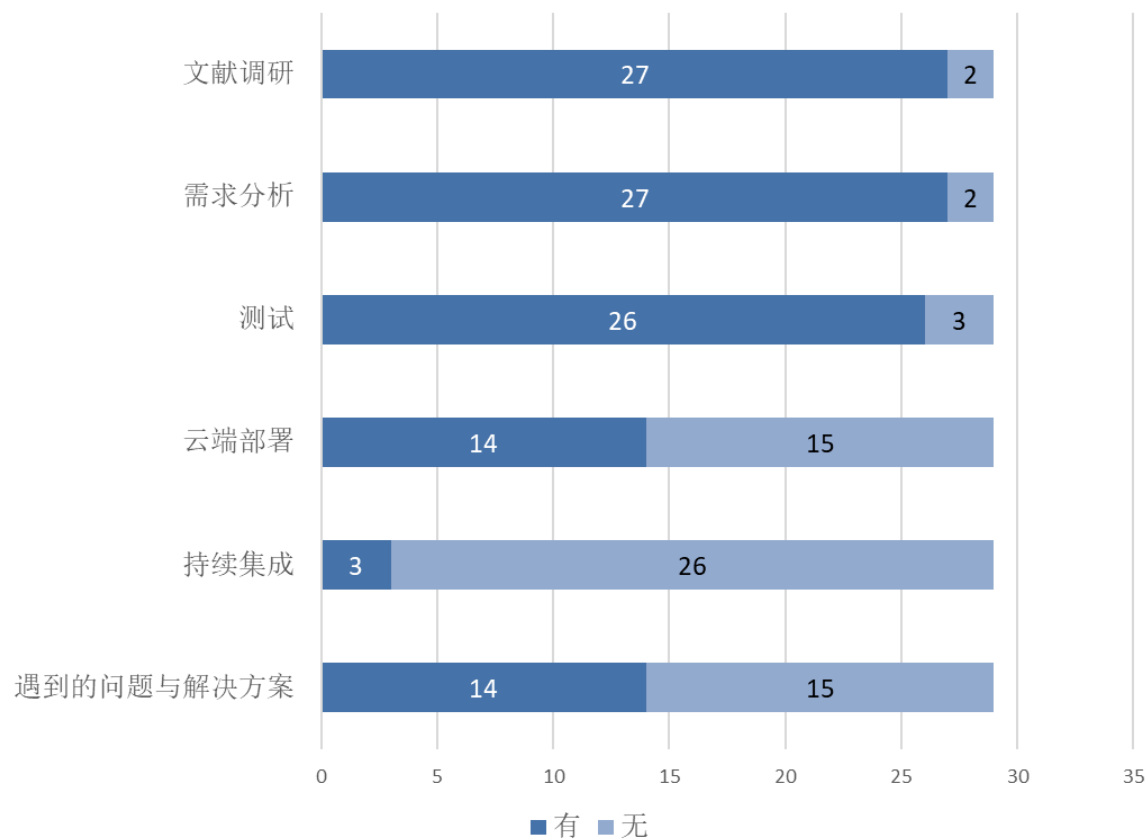


■ 是 ■ 否



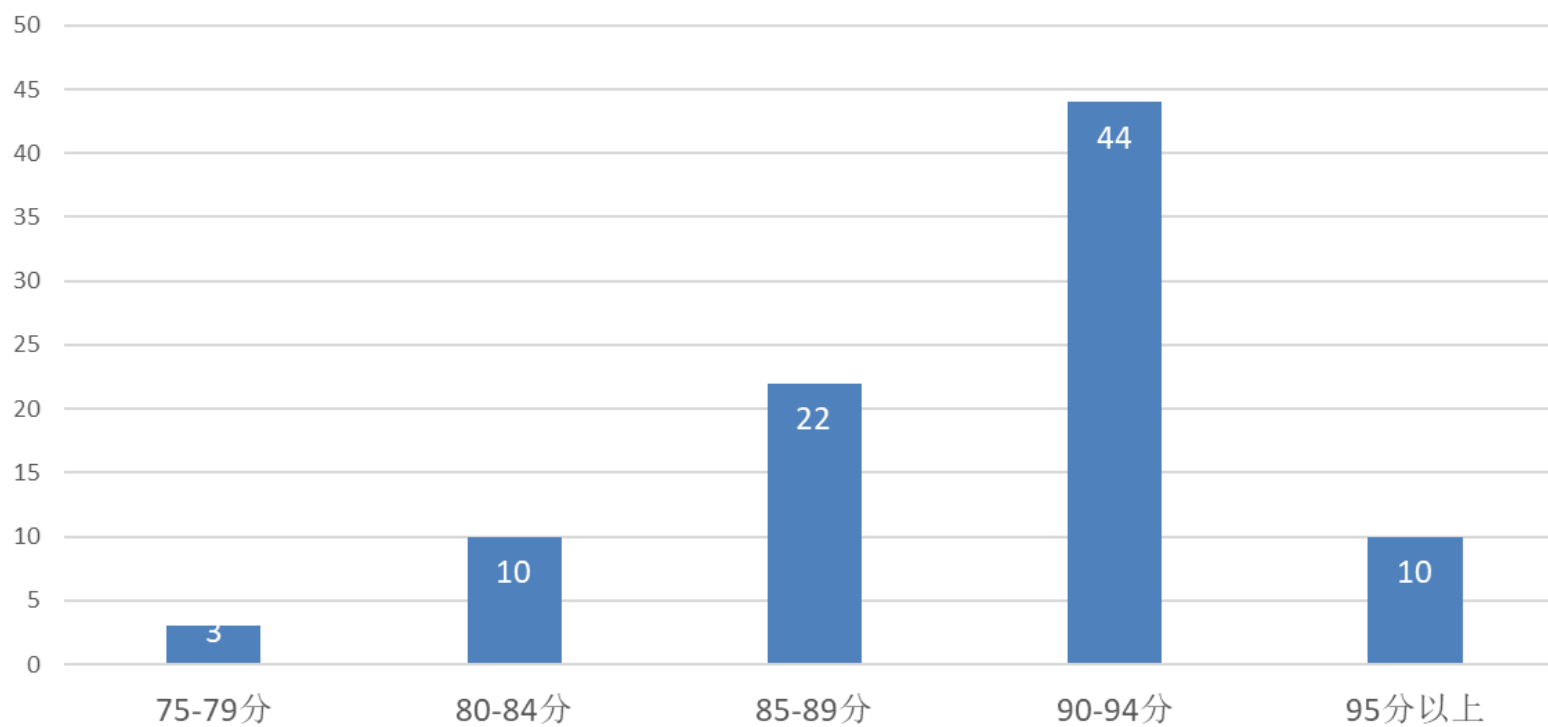
开发各阶段实施情况

29组中，文献调研、需求分析、自动测试、云端部署、持续集成、遇到的问题与解决方案等完成情况



学生成绩分布情况

分数分布



主要问题

1. 文献调研和体验软件的充分性
2. 需求分析的结果物的不完整性
3. 软件架构选择和设计的关注点
4. 没有指出关键代码实现部分
5. 测试范围和测试用例
6. 部署环境的有效性验证



改进建议

1、文献调研的充分性

同类产品或服务、技术选型和实现路径

2、需求分析的结果物

测试数据的检查和准备、验收标准的设立、主要测试用例的描述

3、软件设计的关键点

应用软件运行环境的要求（硬件和软件）、主要的软件组件（版本和来源）、采用什么样的应用开发框架（框架的优点和局限性）、给出可能要修改的代码（文件）位置、格式等要求。



改进建议

4、代码实现的具体要求

有多少种类型的代码？它们的编程要求和例子是什么？如何进行修改后的提交？

5、测试范围和测试用例

测试数据的构建和来源？测试的种类和方法（例子）？如何保障软件质量（如测试覆盖率等指标？）

6、部署环境的有效性验证

你是如何部署上线？是否在修改代码后，部署的时间仍很长或没有自动或半自动化？



提纲

- 上周随堂练习点评
- 教学计划简介
- 2020年课程项目总结
- 魔方项目的演示与讨论
- 软件架构及开发模式

课程项目：魔方机器人

Rubik's Cube



From Wikipedia, the free encyclopedia

Rubik's Cube is a [3-D combination puzzle](#) invented in 1974^{[1][2]} by Hungarian sculptor and professor of architecture [Ernő Rubik](#). Originally called the **Magic Cube**,^[3] the puzzle was licensed by Rubik to be sold by [Ideal Toy Corp.](#) in 1980^[4] via businessman Tibor Laczi and Seven Towns founder [Tom Kremer](#)^[5] and won the [German Game of the Year](#) special award for Best Puzzle that year. As of January 2009, 350 million cubes had been sold worldwide,^{[6][7]} making it the world's top-selling puzzle game.^{[8][9]} It is widely considered to be the world's best-selling toy.^[10]

On the original classic Rubik's Cube, each of the six faces was covered by nine stickers, each of one of six solid colours: white, red, blue, orange, green, and yellow. The current version of the cube has been updated to coloured plastic panels instead, which prevents peeling and fading.^[11] In currently sold models, white is opposite yellow, blue is opposite green, and orange is opposite red, and the red, white and blue are arranged in that order in a clockwise arrangement.^[12] On early cubes, the position of the colours varied from cube to cube.^[13] An internal pivot mechanism enables each face to turn independently, thus mixing up the colours. For the puzzle to be solved, each face must be returned to have only one colour. Similar puzzles have now been produced with various numbers of sides, dimensions, and stickers, not all of them by Rubik.

Although the Rubik's Cube reached its height of mainstream popularity in the 1980s, it is still widely known and used. Many [speedcubers](#) continue to practice it and similar puzzles; they also compete for the fastest times in various categories. Since 2003, the [World Cube Association](#), the Rubik's Cube's international governing body, has organised competitions worldwide and recognises world records.

Rubik's Cube



Other names	Magic Cube, Speed Cube, Puzzle Cube
Type	Combination puzzle
Inventor(s)	Ernő Rubik
Company	Rubik's Brand Ltd
Country	Hungary
Availability	1977: as Hungarian Magic Cube, first test batches released in Budapest 1980: as Rubik's Cube, worldwide–present Official website

Contents [\[hide\]](#)

- 1 [Conception and development](#)
 - 1.1 [Similar precursors](#)

课程项目：魔方机器人

nature
machine intelligence

ARTICLES

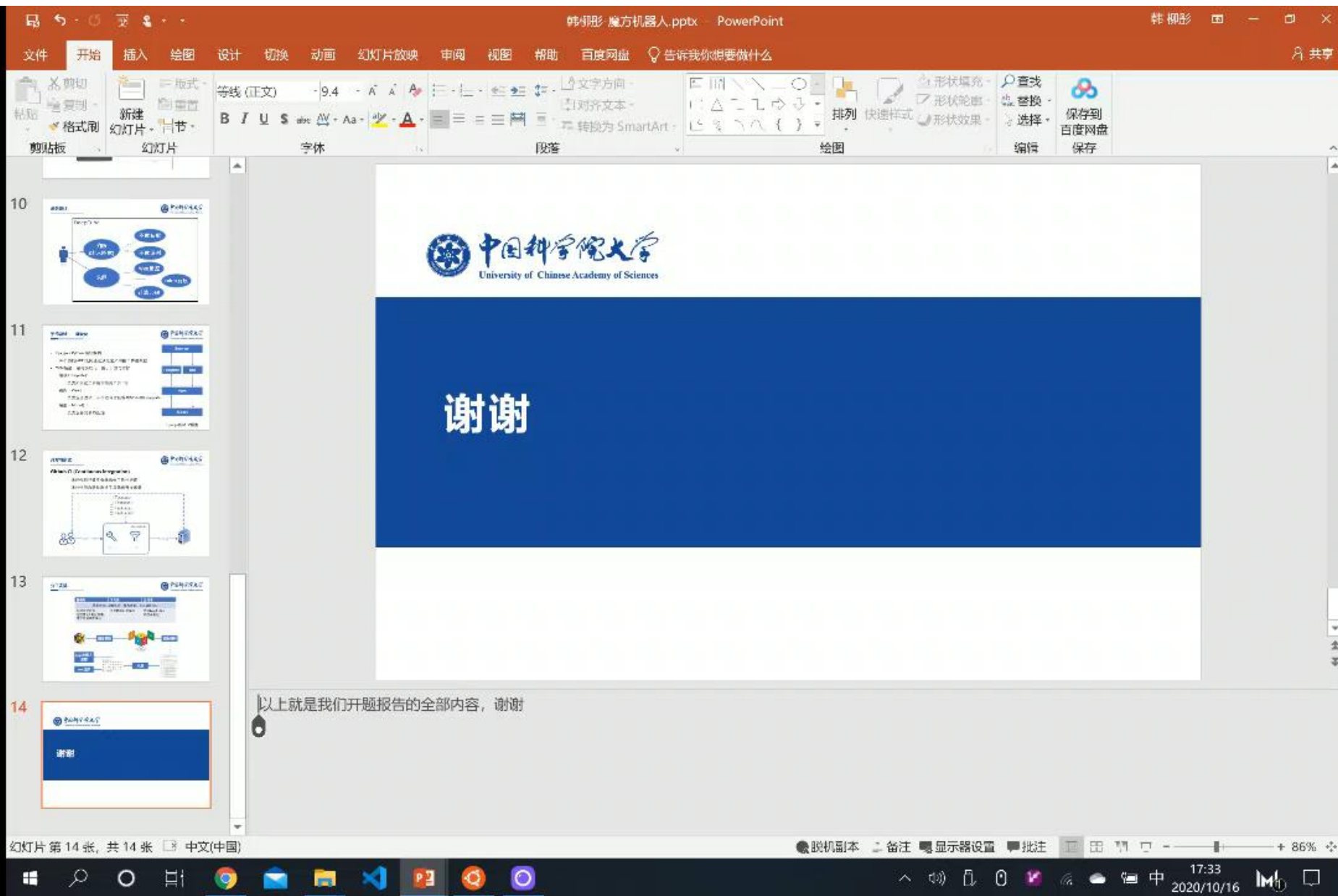
<https://doi.org/10.1038/s42256-019-0070-z>

Solving the Rubik's cube with deep reinforcement learning and search

Forest Agostinelli^{1,3}, Stephen McAleer^{2,3}, Alexander Shmakov^{1,3} and Pierre Baldi^{1,2*} 

The Rubik's cube is a prototypical combinatorial puzzle that has a large state space with a single goal state. The goal state is unlikely to be accessed using sequences of randomly generated moves, posing unique challenges for machine learning. We solve the Rubik's cube with DeepCubeA, a deep reinforcement learning approach that learns how to solve increasingly difficult states in reverse from the goal state without any specific domain knowledge. DeepCubeA solves 100% of all test configurations, finding a shortest path to the goal state 60.3% of the time. DeepCubeA generalizes to other combinatorial puzzles and is able to solve the 15 puzzle, 24 puzzle, 35 puzzle, 48 puzzle, Lights Out and Sokoban, finding a shortest path in the majority of verifiable cases.

课程项目： 魔方机器人

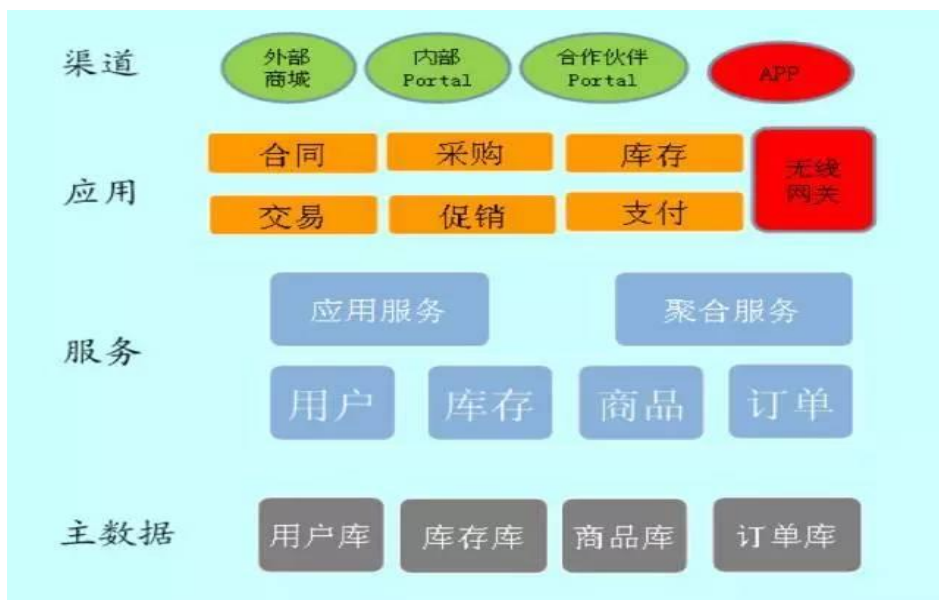




提纲

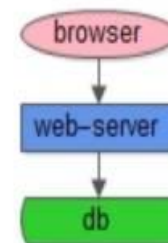
- 上周随堂练习点评
- 教学计划简介
- 2020年课程项目总结
- 魔方项目的演示与讨论
- 软件架构及开发模式

应用软件技术架构演化



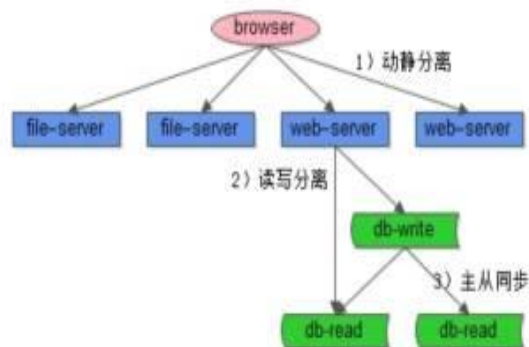
架构-ALL IN ONE

- 架构图
- 架构特点
 - (1) 单机系统 (all in one)
 - (2) 程序耦合 (all in one)
 - (3) 逻辑核心是CURD



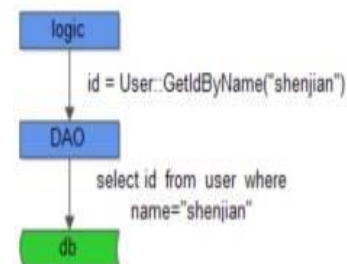
架构-分布式

- 架构图
- 架构特点
 - (1) 分布式系统
 - (2) 动静分离
 - (3) 读写分离 (主从同步)



技术

- DAO
 - (1) Data Access Object
 - (2) 像访问对象一样访问数据
- ORM
 - (1) Object Relation Mapping
 - (2) 简化数据库查询过程



软件开发和运维模式的演化

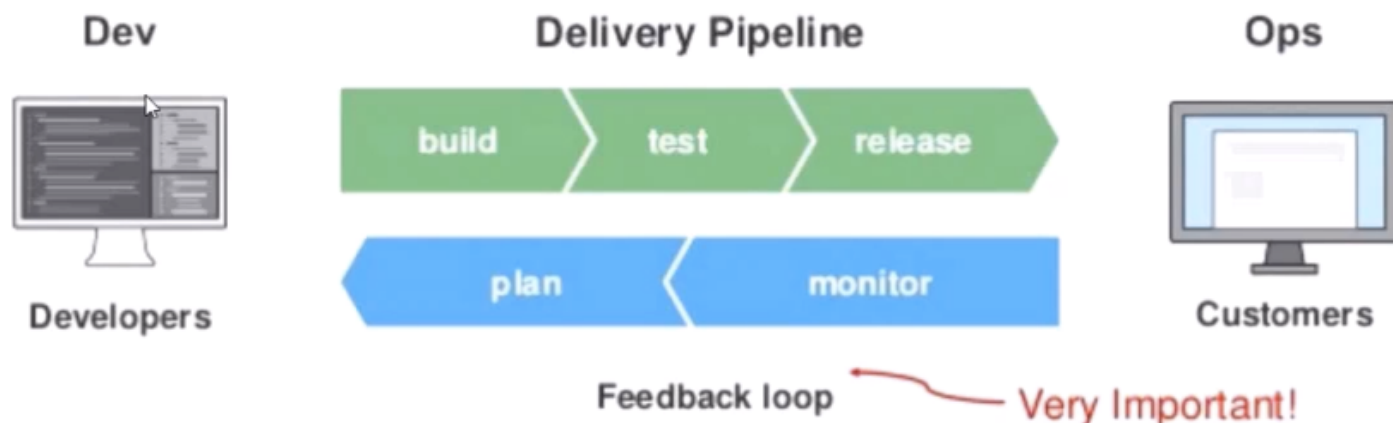


Event Driven

从开发到部署的过程中，每一个环节都可能产生问题，因此必须建立一个feedback loop用于及时有效地传达问题。事件驱动是将“**问题左移**”的有效手段。

问题左移指的是尽可能在代码阶段发现，并解决可能存在的部署问题。这要求建立一个完整的从构建、测试、到部署的链条，而消息系统是这个链条中重要的构件。

DevOps at a Glance

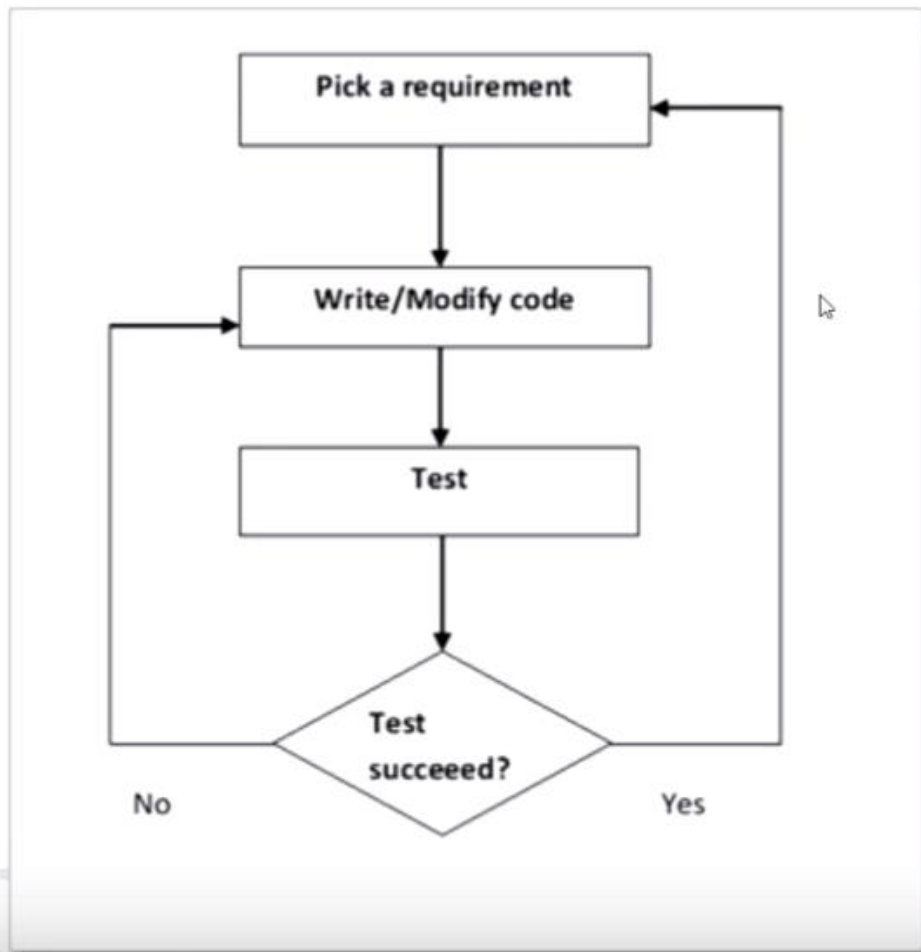


- **DevOps** = efficiencies and techniques that speed up this lifecycle

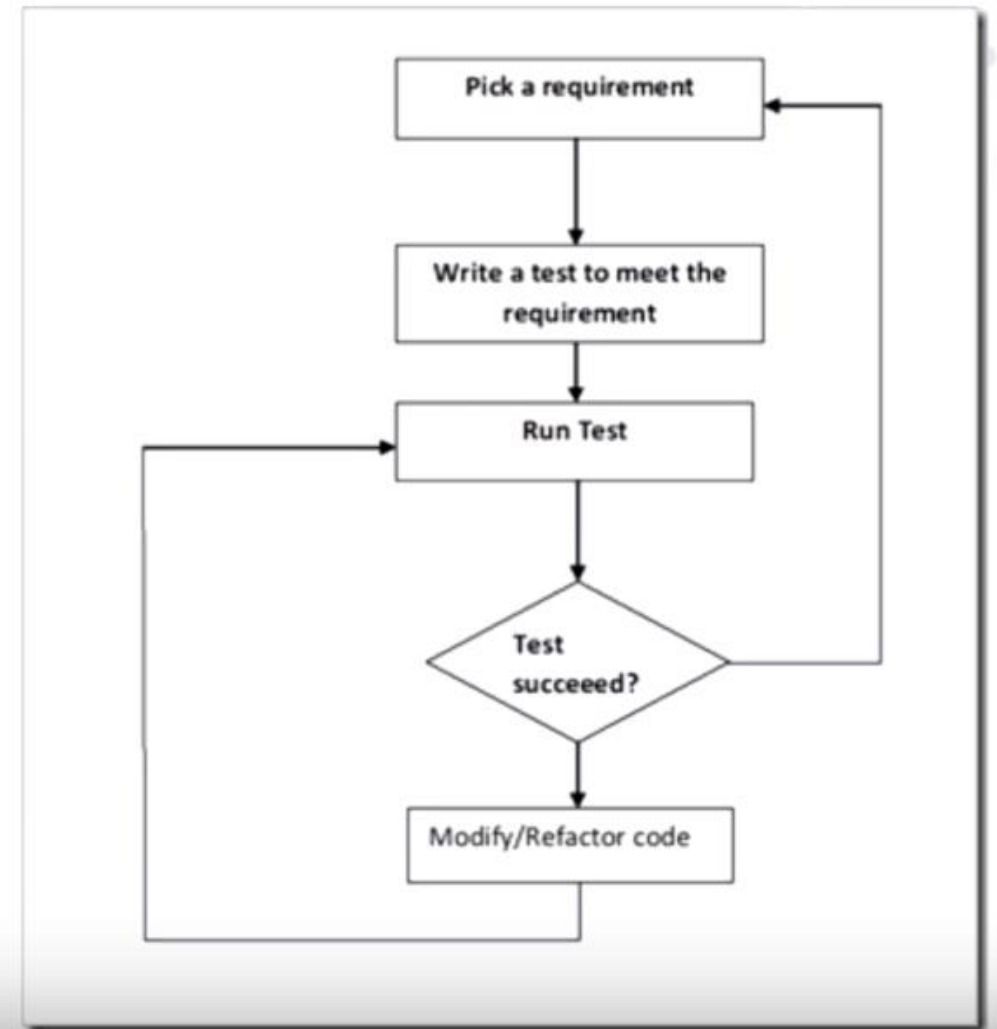
传统与测试驱动开发模式

Test Driven Development (TDD)

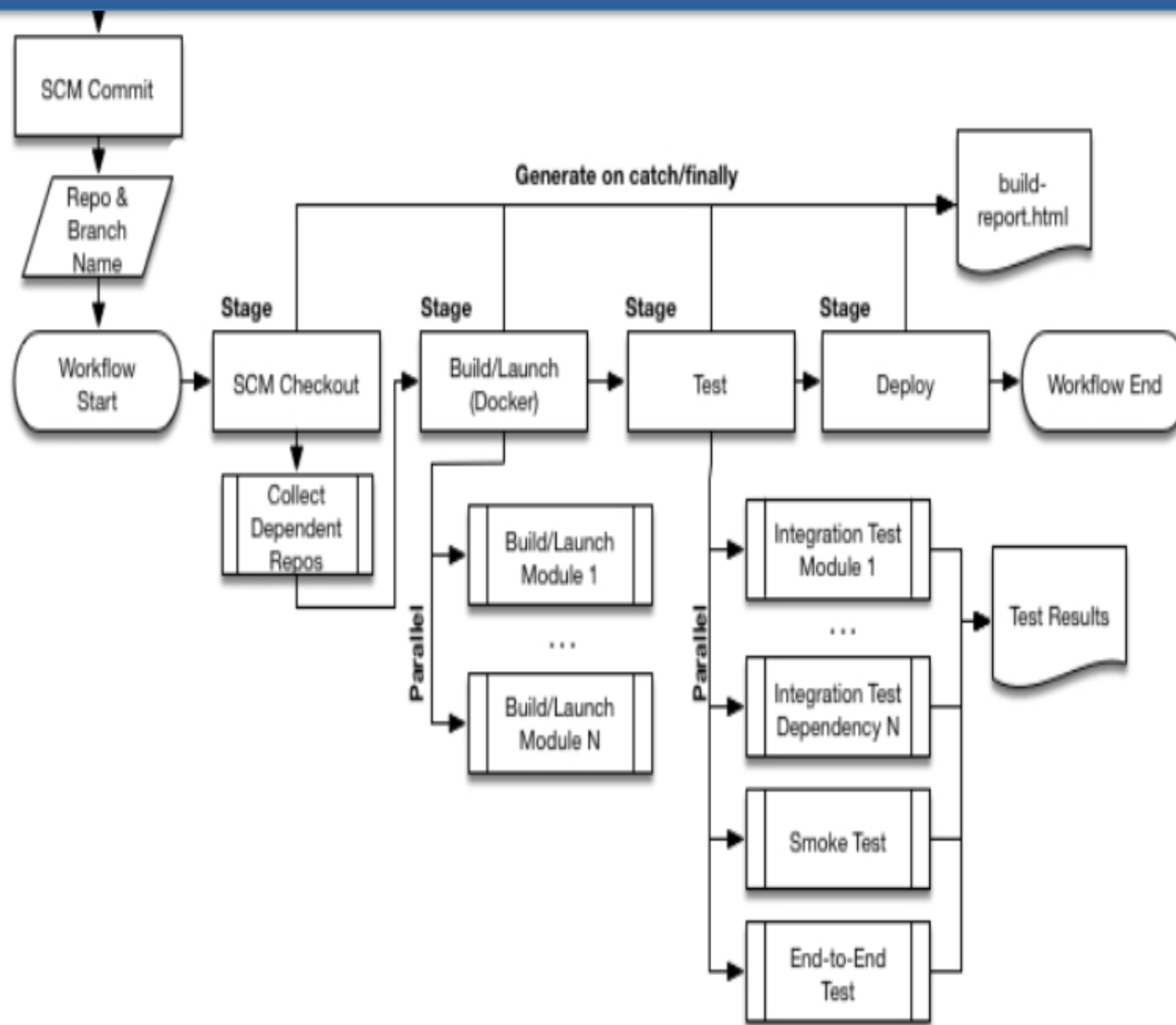
传统开发流程



测试驱动开发流程



软件开发到生产部署的内部流程



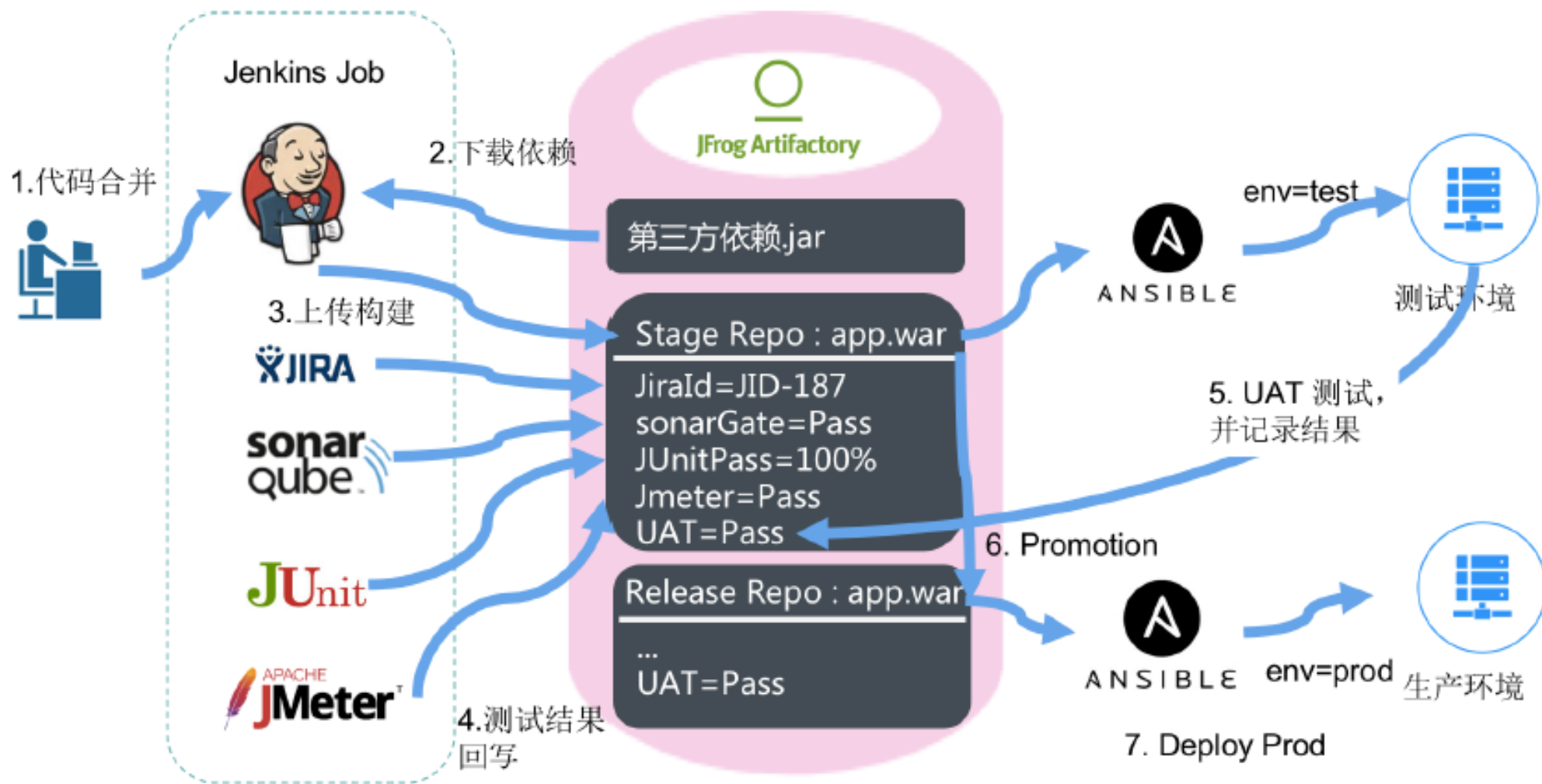
Development



Production



从代码到系统部署自动化过程



本周练习

一、课外作业要求

针对该课程的课程项目，我们提供了2020年秋季高级软件工程的十个课程案例，请浏览十个案例（包括PPT、课程报告、视频等），并对十个案例进行排名；对自己选择的第一名案例进行深度分析。给出该案例的优点和缺点，各2~3条。

二、你如何在这个学期的课程项目提高水平

经过自己的深度分析和点评后，针对该课程期末需要提交的课程项目材料，你认为还应该补充或者强调哪些内容。

三、要求9月17日晚10:00前提交到课程网站