

高级软件工程

第三周 (Sep. 18、Sep. 22)

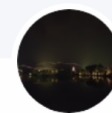
主讲：罗铁坚

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

1. 课堂练习和课外作业点评

9月10日、13日

2. 如何学习和评价成果

3. 软件工程文档的案例分析

IEEE 文档标准和大学生创新软件设计文档等

4. 需求、设计、代码的关系

提纲

1. 课堂练习和课外作业点评

2. 如何学习和评价成果

3. 软件工程文档的案例分析

IEEE 文档标准和大学生创新软件设计文档

4. 需求、设计、代码的关系

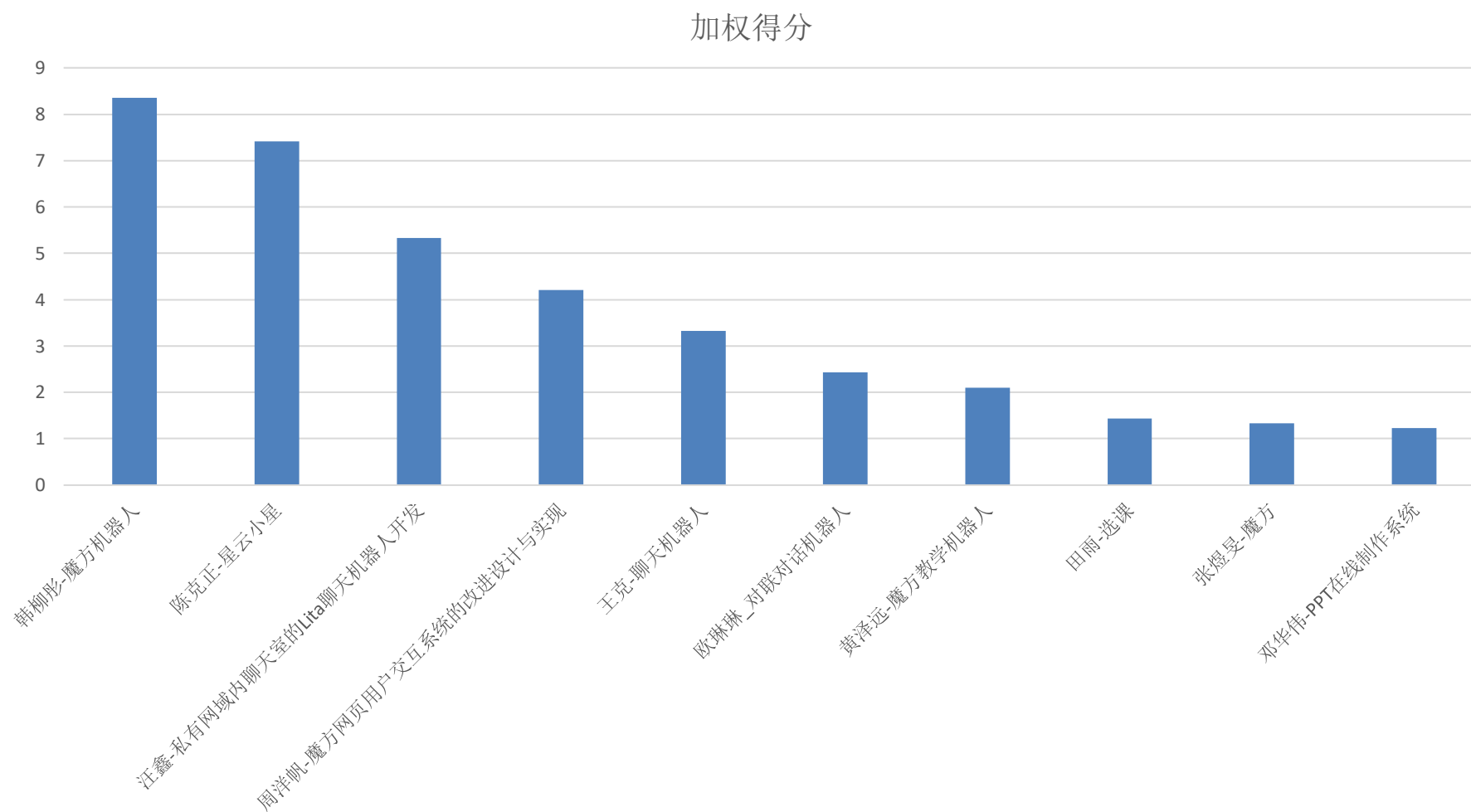
2021年9月10日课外作业完成情况分析

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

自选案特点分析

优点	缺点
应用场景明确	时间计划粗糙
部署平台易于传播和使用	前端实现朴素
详细竞品分析	未实现热更新
软硬件结合	功能不够丰富
用户交互方便	用户交互复杂
模块化	README不清晰全面
架构清晰	
自动化测试	

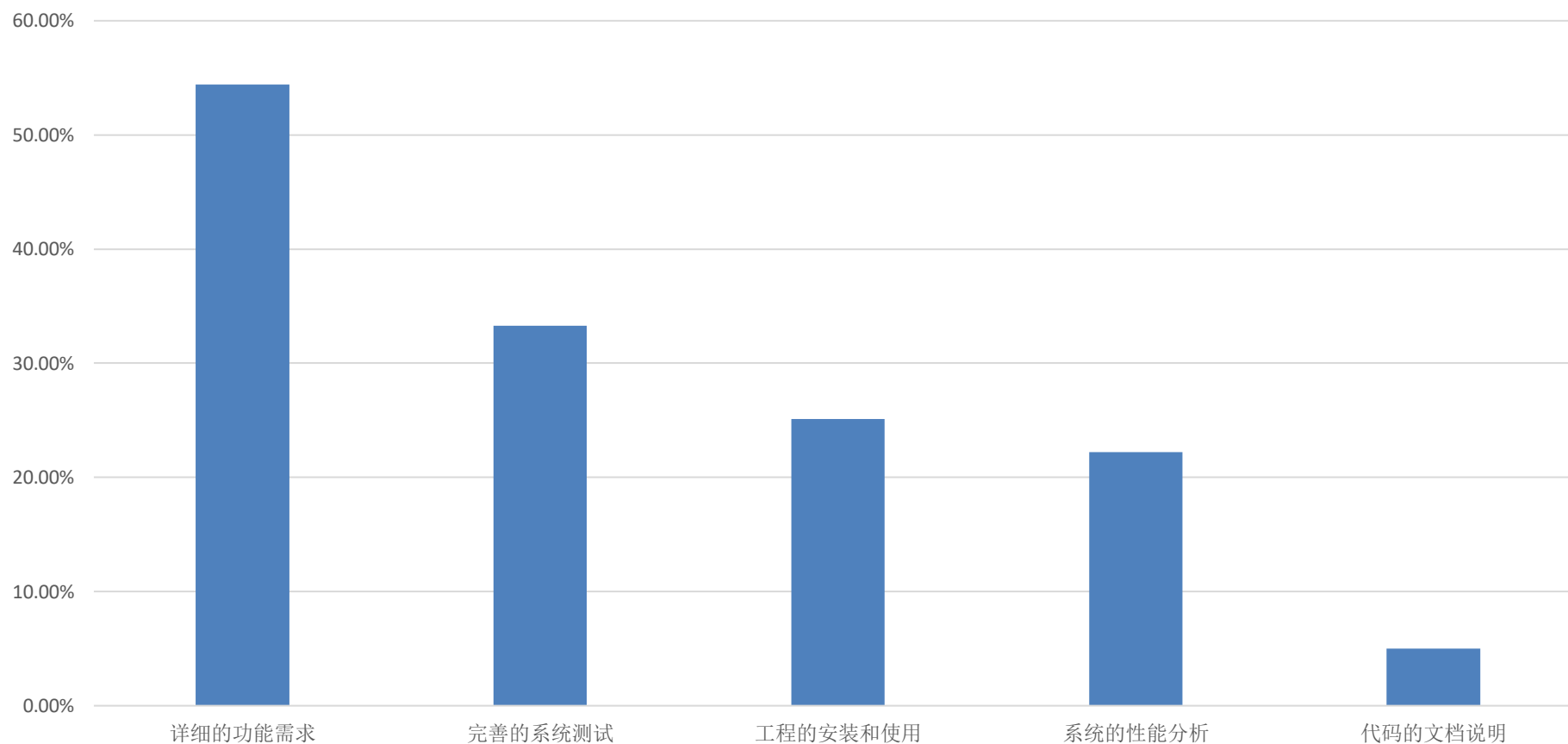
2021年9月10日课外作业一学生打分评价情况



2021年9月10日课外作业一学生打分评价情况

二、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。

关于提交材料中补充内容的统计



2021年9月13日课外作业—学生评价课程项目完成情况

韩柳彤组“魔方机器人项目”特点评价

←

总结：←

优点：←

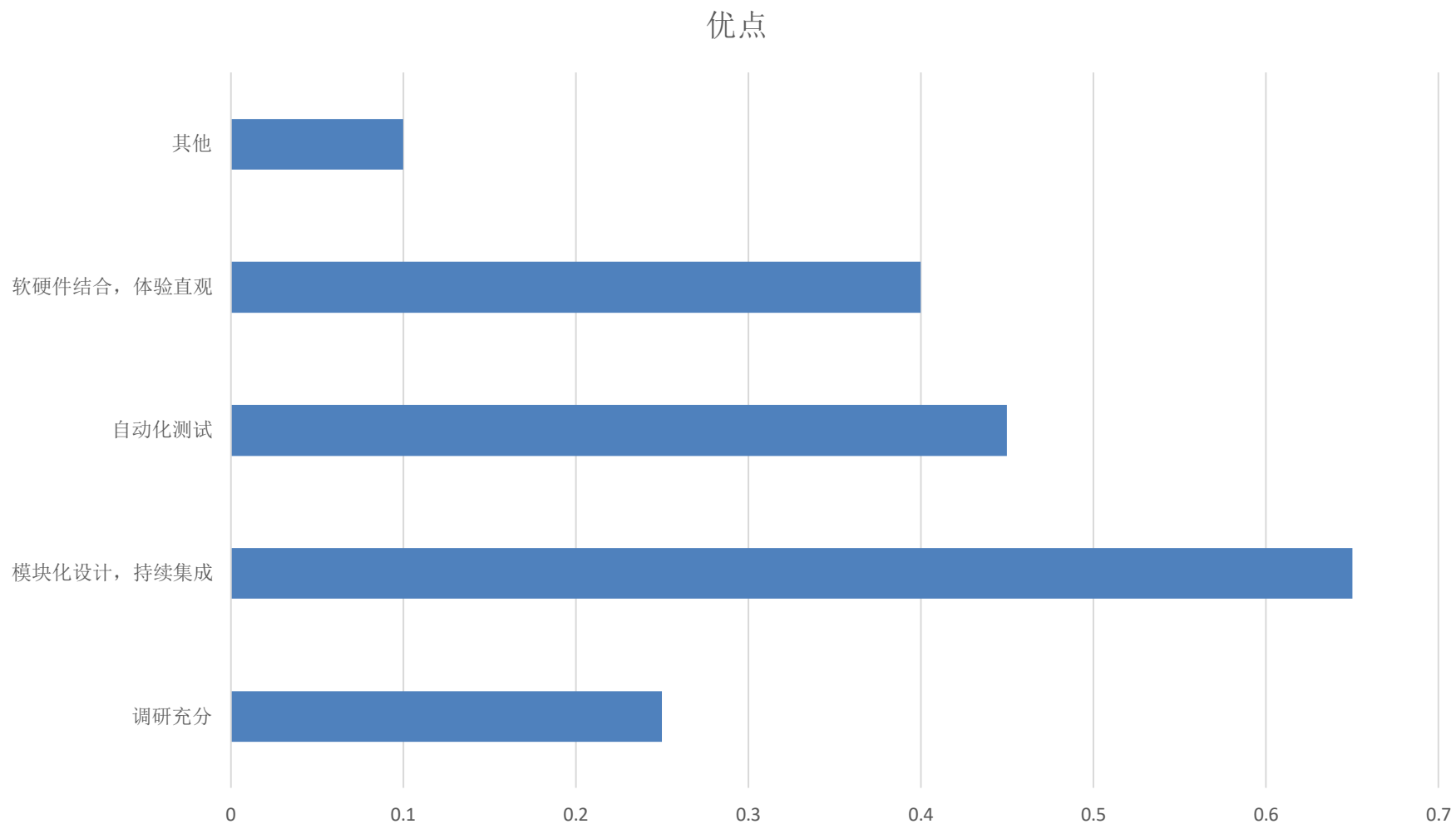
- (1) 调研充分，与现有项目做了对比，明确可改进部分；25%←
- (2) 模块化设计，持续集成；65%←
- (3) 自动化测试；45%←
- (4) 软硬件结合，便于直观体验、量化评估；40%←
- (5) 其他。10%←

缺点：←

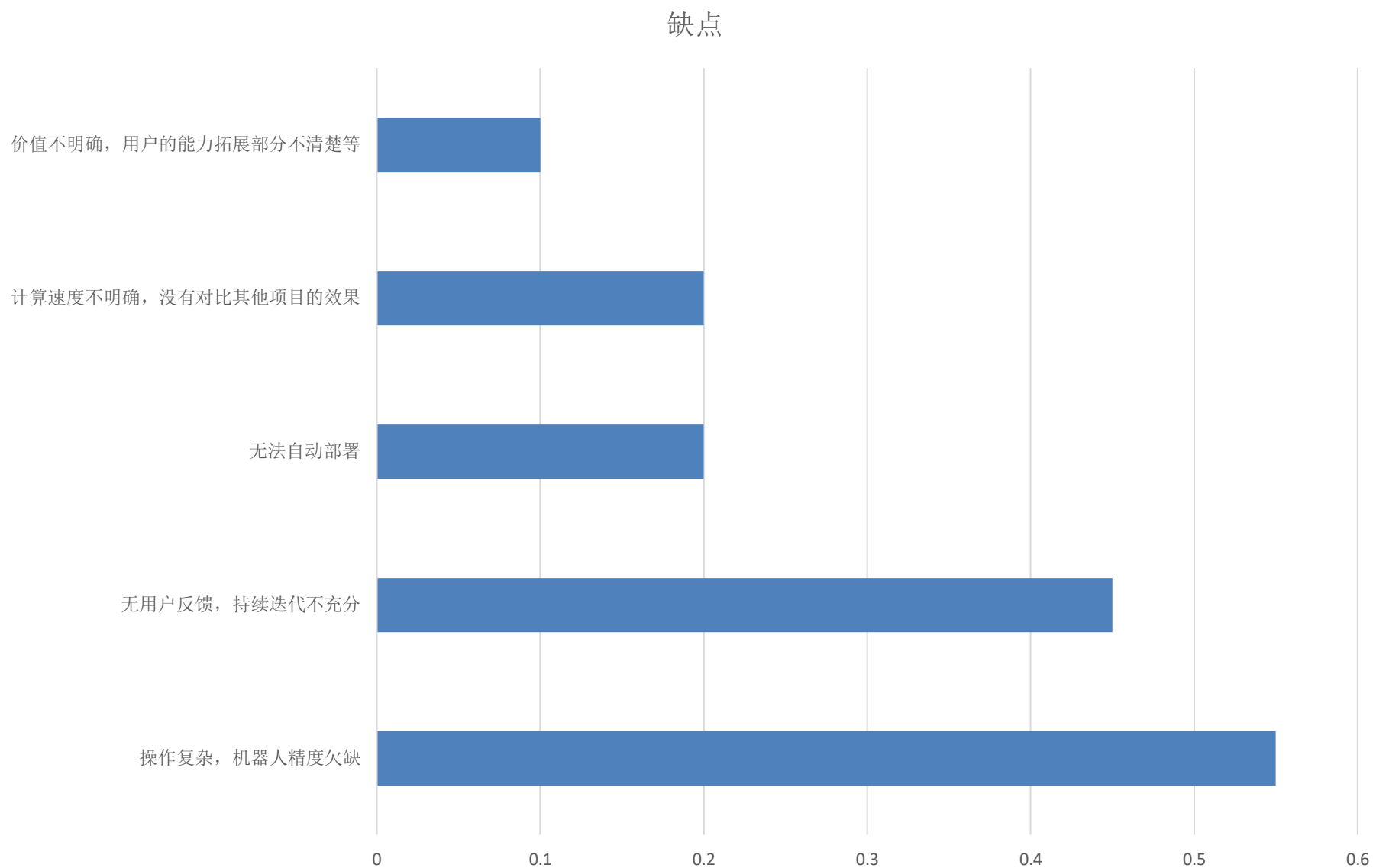
- (1) 照片拍摄上传消耗大，操作复杂，机器人有时旋转精度不够；55%←
- (2) 持续迭代未做充分说明，没有用户反馈，没有做进一步推进，比如更高阶的魔方；45%←
- (3) 无法自动部署；20%←
- (4) 计算速度不明确，也没有对比自己项目的优势；20%←
- (5) 价值不明确，用户的能力拓展不清楚等。10%←



2021年9月13日课外作业一学生评价课程项目完成情况



2021年9月13日课外作业一学生评价课程项目完成情况



补充内容↵	例子↵
详细的功能需求↵	<ol style="list-style-type: none"> 1. 需求分析需要更加完备，要考虑到项目具体使用场景，以及↵对于不同使用人群而言，不同的需求。↵ 2. 需求分析说明书：需求分析是一个项目开展前的必备工作，充分了解需求才能开展项目。↵
完善的系统测试↵	<ol style="list-style-type: none"> 1. 测试阶段，要进行单元测试和集成测试，包括系统的负载均衡等都要进行考虑，测试系统最大并发量。↵ 2. 应该需要补充对于该项目如果已经有了现成的案例应该将其其他的项目与自己的项目进行测试分析比较来评估自己项目的特点与好坏。我认为应该需要强调用户体验感如何以及继承测试方面的具体介绍。↵
工程的安装和使用↵	<ol style="list-style-type: none"> 1. 软件部署手册↵ 2. 本项目的安装教程及使用说明↵
系统的性能分析↵	<ol style="list-style-type: none"> 1. 项目部署和发布后，要考虑到可能会出现的问题，例如服务器崩溃，Web 端安全问题，服务器安全问题等。↵ 2. 使用软件工程规范的开发方法和过程进行系统开发，在系统实现和设计时应充分考虑系统的可维护性和可扩充性。↵
代码的文档说明↵	<p>添加对代码的说明文档，对系统部署方式的说明文档，对运行平台要求的说明文档，对乐高机↵</p> <p>器人搭建、编程的说明文档。这些内容需独立提供，不能放在结题报告里↵</p>
其他↵	<ol style="list-style-type: none"> 1. 在系统开发过程中用到的软件工程领域的知识和方法，开发一个完善的系统需要工程性的方法指导，所以在待提交的课程项目中应该记录使用到的软件工程领域的方法或思想。↵ 2. 项目遇到的困难与未来的展望。↵ 3. 我觉得还要有一定的观赏效果，有的组没有较好的观赏效果，可能会让人忽略一些其做的工作。↵



提纲

1. 课堂练习和课外作业点评

9月10日、13日

2. 如何学习和评价成果

3. 软件工程文档的案例分析

IEEE 文档标准和大学生创新软件设计文档等

4. 需求、设计、代码的关系

人类学习理论创立者之一

Table D.4. Revised Bloom's Cognitive Skill list [13]

	B-I Remembering	B-II Understanding	B-III Applying	B-IV. Analyzing	B-V Evaluating	B-VI. Creating
Definitions	Exhibit memory of previously learned materials by recalling facts, terms, basic concepts, and answers.	Demonstrate understanding of facts and ideas by organizing, comparing, translating, interpreting, giving descriptions,	Solve problems to new situations by applying acquired knowledge, facts, techniques and rules in a different way.	Examine and break information into parts by identifying motives or causes. Make inferences and find evidence to support	Present and defend opinions by making judgments about information, validity of ideas, or quality of	Compile information together in a different way by combining elements in a new pattern or proposing alternative
Verbs	Choose, Define, Find, How, Label, List, Match, Name, Omit, Recall, Relate, Select, Show, Spell, Tell, What, When, Where, Which, Who, Why	Classify, Compare, Contrast, Demonstrate, Explain, Extend, Illustrate, Infer, Interpret, Outline, Relate, Rephrase, Show, Summarize, Translate	Apply, Build, Choose, Construct, Develop, Experiment, Interview, Make, use, of, Model, Organize, Plan, Select, Solve, Utilize	Analyze, Assume, Categorize, Classify, Compare, Conclusion, Contrast, Discover, Dissect, Distinguish, Divide, Examine, Function, Inference, Inspect, List, Motive, Relationships, Simplify, Survey, Take part in, Test for, Theme	Agree, Appraise, Assess, Award, Choose, Compare, Conclude, Criteria, Criticize, Decide, Deduct, Defend, Determine, Disprove, Estimate, Evaluate, Explain, Importance, Influence, Interpret, Judge, Justify, Mark, Measure, Opinion, Perceive, Prioritize, Prove, Rate, Recommend, Rule on, Select, Support, Value	Adapt, Build, Change, Choose, Combine, Compile, Compose, Construct, Create, Delete, Design, Develop, Discuss, Elaborate, Estimate, Formulate, Happen, Imagine, Improve, Invent, Make up, Maximize, Minimize, Modify, Original, Originate, Plan, Predict, Propose, Solution, Solve, Suppose, Test, Theory



(布隆姆)

Benjamin Bloom

Born	Benjamin Samuel Bloom February 21, 1913 Lansford, Pennsylvania
Died	September 13, 1999 (aged 86) Chicago
Nationality	United States citizen
Education	Ph.D. in Education
Alma mater	Pennsylvania State University, University of Chicago
Occupation	Educational psychologist
Employer	American Educational Research Association

- 1、[Bloom, B. S.](#); Engelhart, M. D.; Furst, E. J.; Hill, W. H.; [Krathwohl, D. R.](#) (1956). Taxonomy of educational objectives: The classification of educational goals. Handbook I: **Cognitive domain**. New York: David McKay Company
- 2、[Krathwohl, D. R.](#); [Bloom, B. S.](#); Masia, B. B. (1964). Taxonomy of educational objectives: The classification of educational goals. Handbook II: **the affective domain**. New York: David McKay Company.
- 3、Bloom, B. S. (1994). "Reflections on the development and use of the taxonomy". In Rehage, Kenneth J.; Anderson, Lorin W.; Sosniak, Lauren A. Bloom's taxonomy: A forty-year retrospective. Yearbook of the National Society for the Study of Education. **93**. Chicago: National Society for the Study of Education. [ISSN 1744-7984](#).



人类学习理论模型

(布隆姆)



教学理论创立者之一

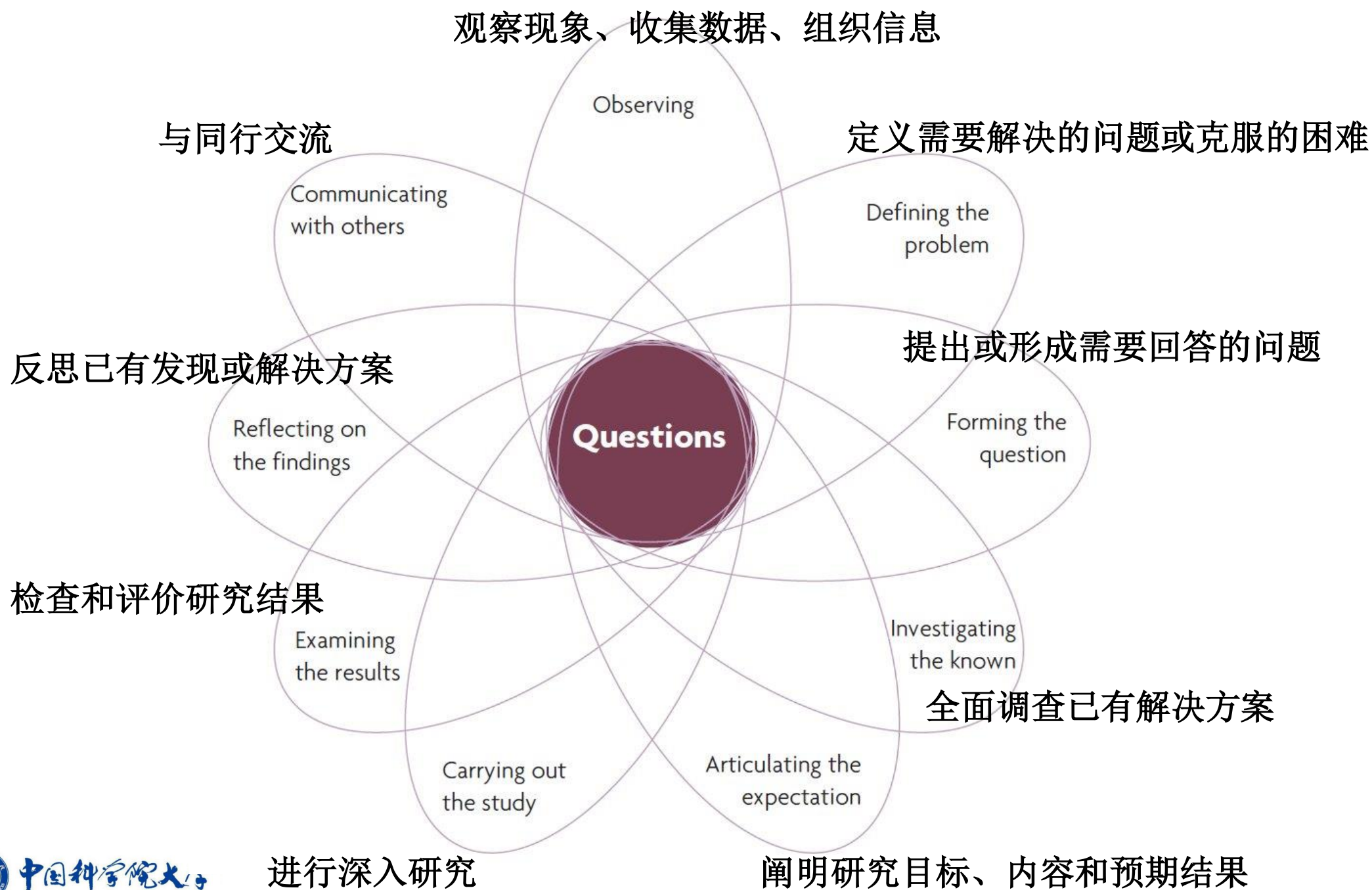
Ralph W. Tyler (1902-1994) 泰勒

教学设计三阶段

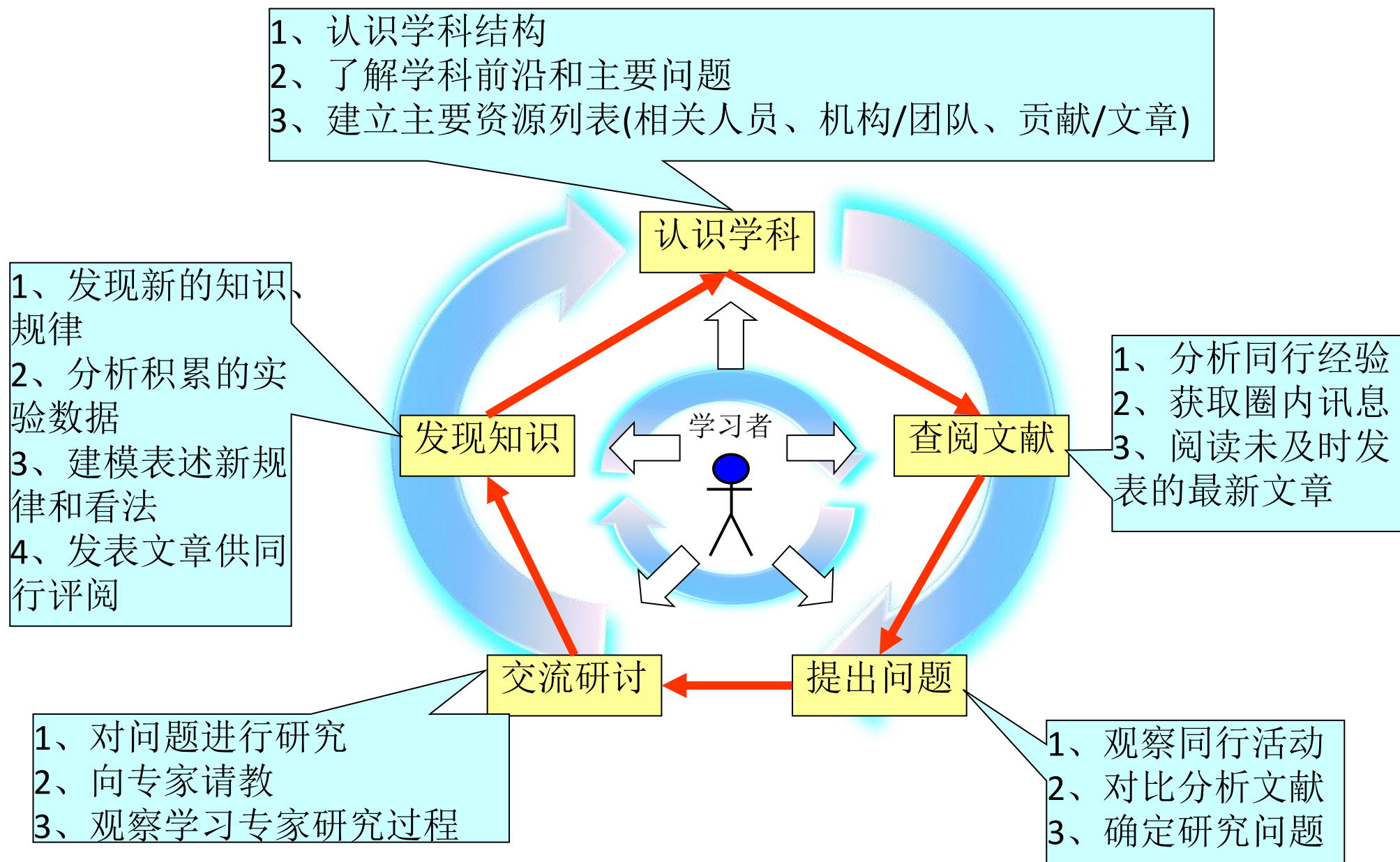
- 1、规划教学目标和内容
- 2、设计体验以便获取学习效果
- 3、评估学生理解程度以便改进



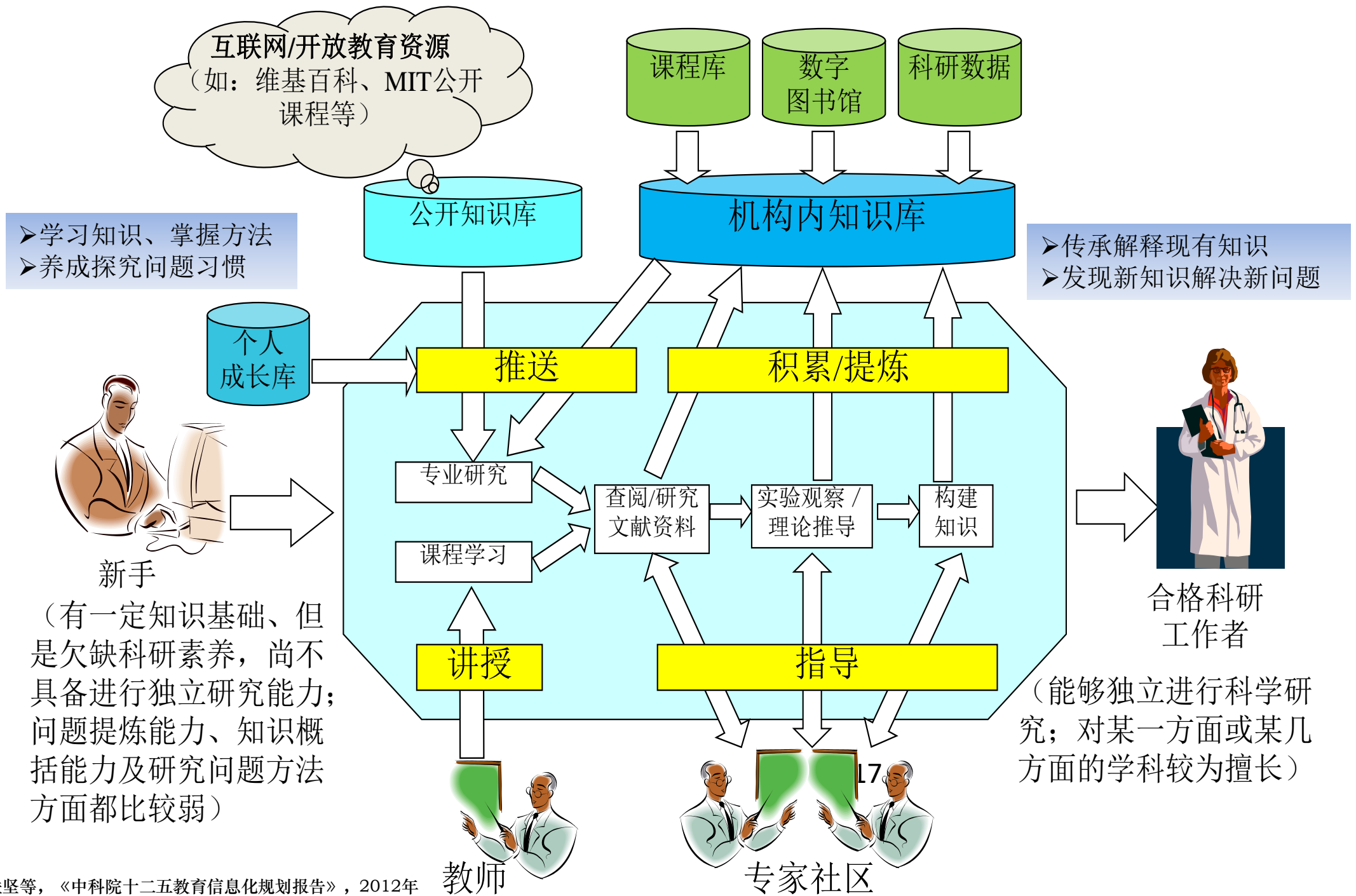
培养学生的根本目标：提出问题和探寻求解



我们提出培养创造力模型



我们提出的促进学习成长框架



提纲

1. 课堂练习和课外作业点评

2. 如何学习和评价成果

3. 软件工程文档的案例分析

IEEE 文档标准和大学生创新软件设计文档

4. 需求、设计、代码的关系

软件工程文档的案例分析

IEEE 文档标准

大学生创新软件设计文档

1. 是否帮助理解软件价值、系统构成及和代码组织等关键思想。
2. 理解软件价值、重现的方法、设计理由、测试方法及测试用例。

回顾软件工程学科三类根本问题

1. Ensure Quality: How to ensure the software system with high quality? Optimization: low Cost (Technology + People), limit Time => high Quality.
2. Good Architecture : Code organization principles, Define and develop API, Design protocol to connect different components.
Code Refactor: System evolution need to refactor code.
3. Value for User: cost-effective, minimum invest and maximum return, sustainable evolution.

提纲

1. 课堂练习和课外作业点评

2. 如何学习和评价成果

3. 软件工程文档的案例分析

IEEE 文档标准和大学生创新软件设计文档

4. 需求、设计、代码的关系

讨论重点

挑战问题：每个软件项目的目标是构建和交付“正确”的产品或服务。但是没有人（设计者或用户）提前知道他们要什么。

1. 一个具体领域或某项业务的问题是什么？ 有没有已经解决这个业务问题的软件系统？
2. 开发解决业务问题的软件的必要性和可行性？ 回答软件交付和运营后是否产生价值？
3. 寻找业务问题与软件需求之间的关系。潜在的用户在什么应用场景中使用，并获得什么好处，效率或效果体现在哪？，有没有具体的量化指标评价这些效果或效率？
4. 如何用领域专家和软件设计人员都能理解的“语言”表达需求。

讨论软件需求、设计、代码的目的及关系

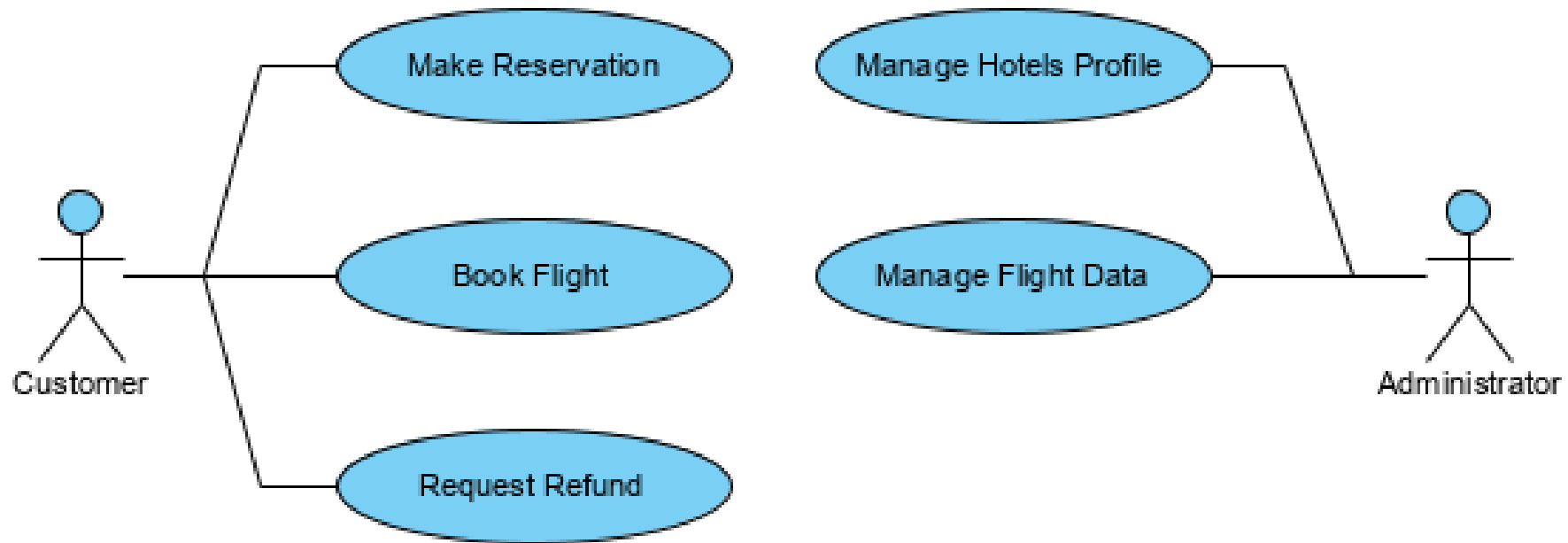
挑战问题：每个软件项目的目标是构建和交付“正确”的产品或服务。但是没有人（设计者或用户）提前知道他们要什么。

1. 发现解决领域问题的核心价值
2. 创作出用计算机解决问题的用例
3. 规划出实现软件功能的具有场景
4. 为设计软件架构、代码和测试提供依据
5. 基本思想：逻辑分层、归纳共性、演绎实例。
6. **软件第一定律：任何用户价值实现一定通过与系统交互完成，而每个具体交互有一个明确的目的和操作。**

评价案例：<http://math-cs.gordon.edu/courses/cs211/ATMExample/>

如何通过用例等建模过程发现价值、规划功能和设计代码？

每个软件项目的目标是构建和交付“正确”的产品或服务。但是没有人（设计者或用户）提前知道他们要什么。



Ivar Jacobson invented use cases in the late 1960s while working on telephony systems at Ericsson. Two decades later, he introduced them to the object-oriented programming community, where they were recognized as filling a significant gap in the way people were working.

[J92] Jacobson, I. et al. Object-Oriented Software Engineering: A Use-Case Driven Approach, Addison-Wesley, Reading, MA, 1992.

Alistair Cockburn, Structuring Use Cases with Goals, 1997

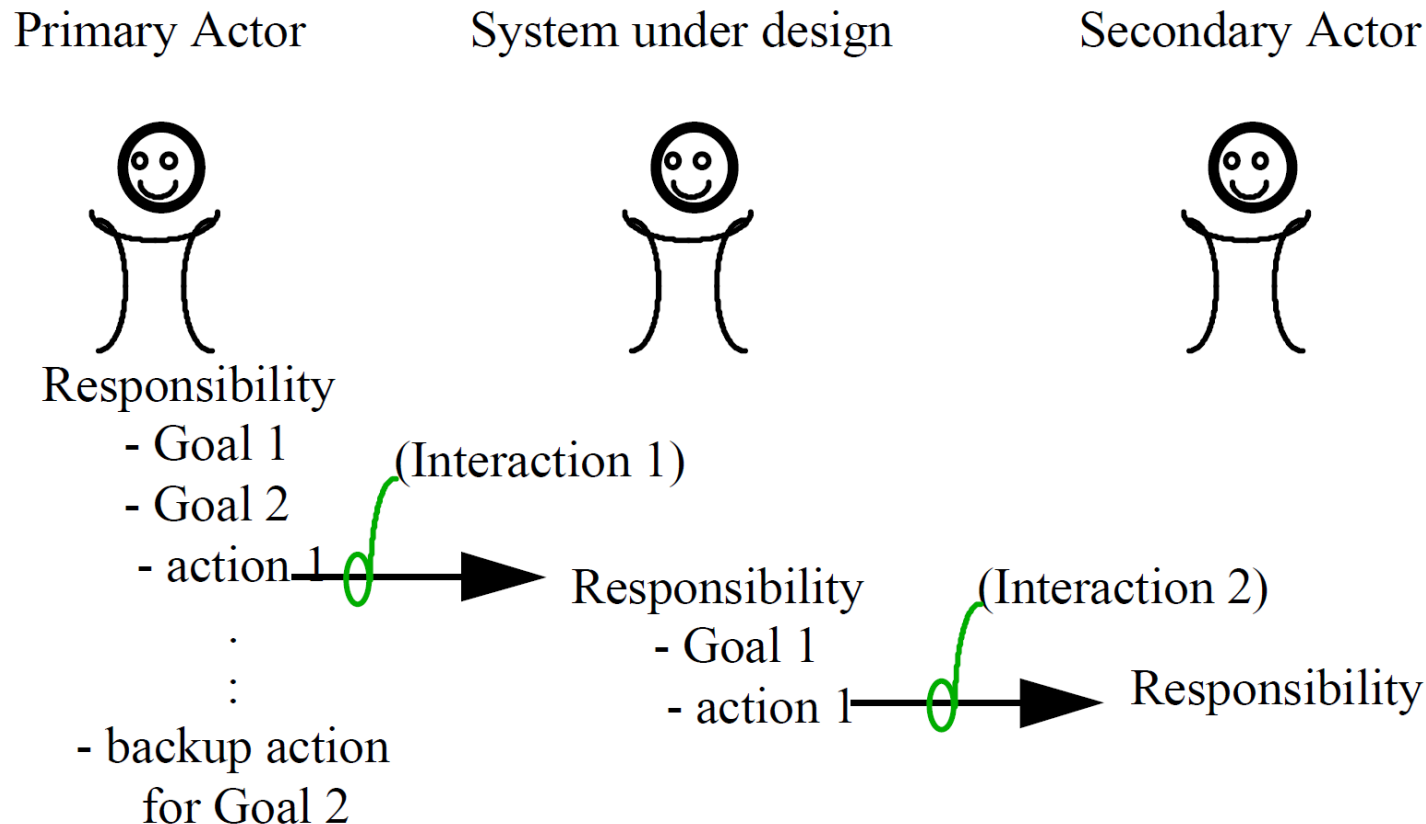
Ivar Jacobson, etc., USE-CASE 2.0 the Hub of Software Development, ACMQUEUE Jan.-Feb. 2016

Jacobson, I. 1987. Object-oriented software development in an industrial environment. In Conference Proceedings of Object-oriented Programming Systems, Languages, and Applications [OOPSLA 87].

“用例”建模理论基础

软件第一定律：任何用户价值实现一定通过与系统交互完成，而每个具体交互有一个明确的目的和操作。

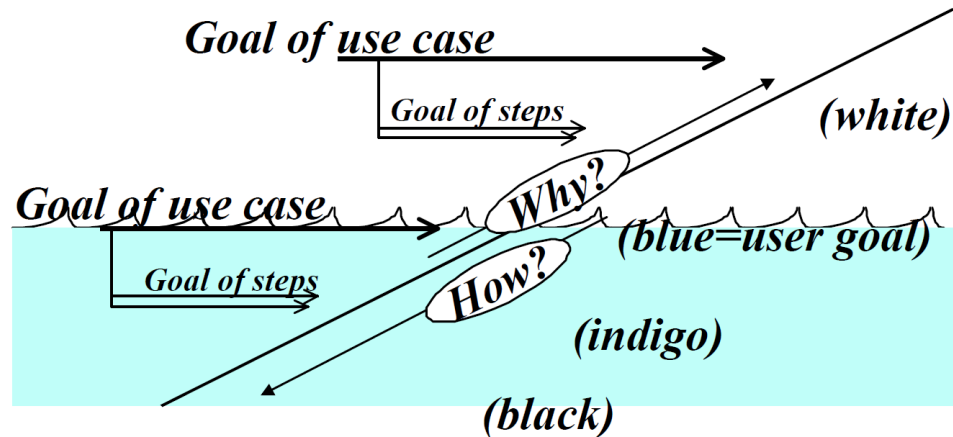
An action connects one actor's goal with another's responsibility.



Use Case. A collection of possible scenarios between the system under discussion and external actors, characterized by the goal the primary actor has toward the system is declared responsibilities, showing how the primary actor's goal might be delivered or might fail.

通过“用例”建模，发现和理解领域知识，优化业务流程实现用户价值

基本方法：分层级和找共性，用业务术语描述战略目标和用户目标、用具体操作定义子功能。



通过回答为什么来找价值；
通过回答如何实现来定义待实现的功能；
要保持分层、价值、功能的一致性。
每个功能都有一个具体例子来验证这个功能的实例：Z、Y、X、W、V五要素齐全。

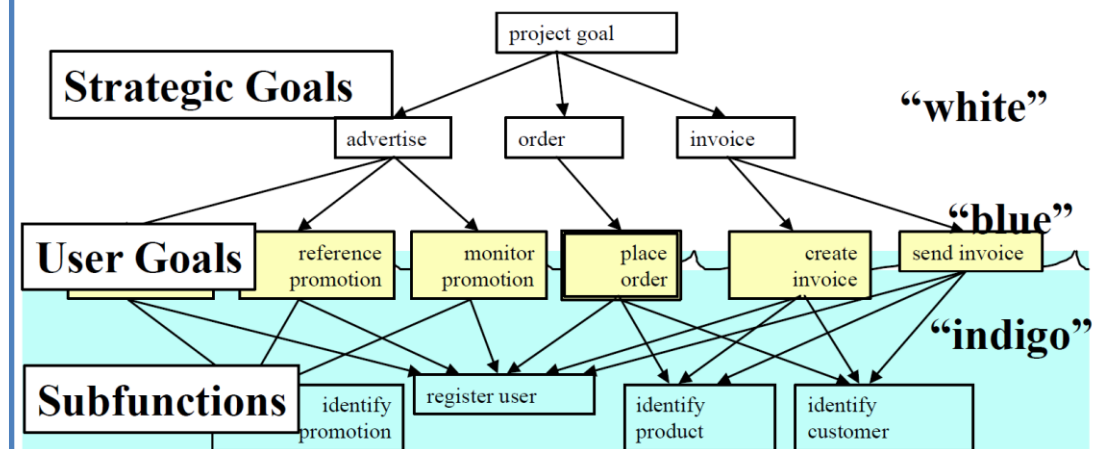
Core form

At the core of the writing is the form:

"At time z, actor y does x to actor w with data v"

Strategic use cases, tasks, and subfunctions link together as a graph.

Sailboat image: User tasks are at sea level.



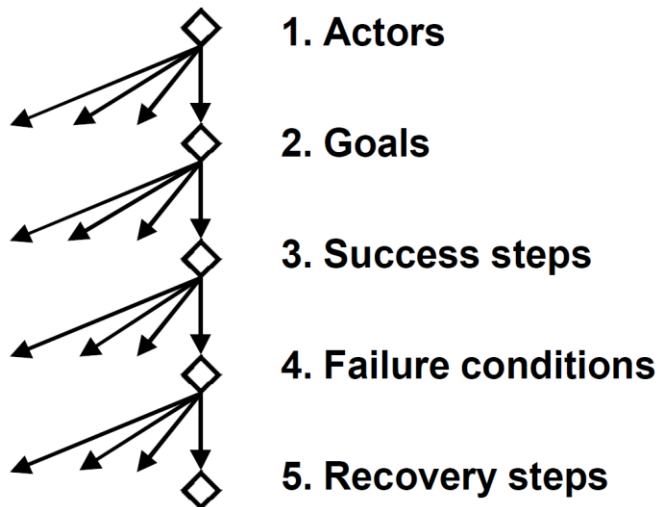
创作“用例”的基本方法

宽度优先：列举全和可区分；纵深要素齐全，每个要素需要回答的问题和明确的信息是不同的。

纵深探索5要素的领域知识

cash-withdrawal use case of a cash machine.

**Build use cases breadth first:
Actors, then goals, then success, then failures.**



basic flow

1. insert card
2. validate card
3. select cash withdrawal
4. select account
5. confirm availability of funds
6. return card
7. dispense cash

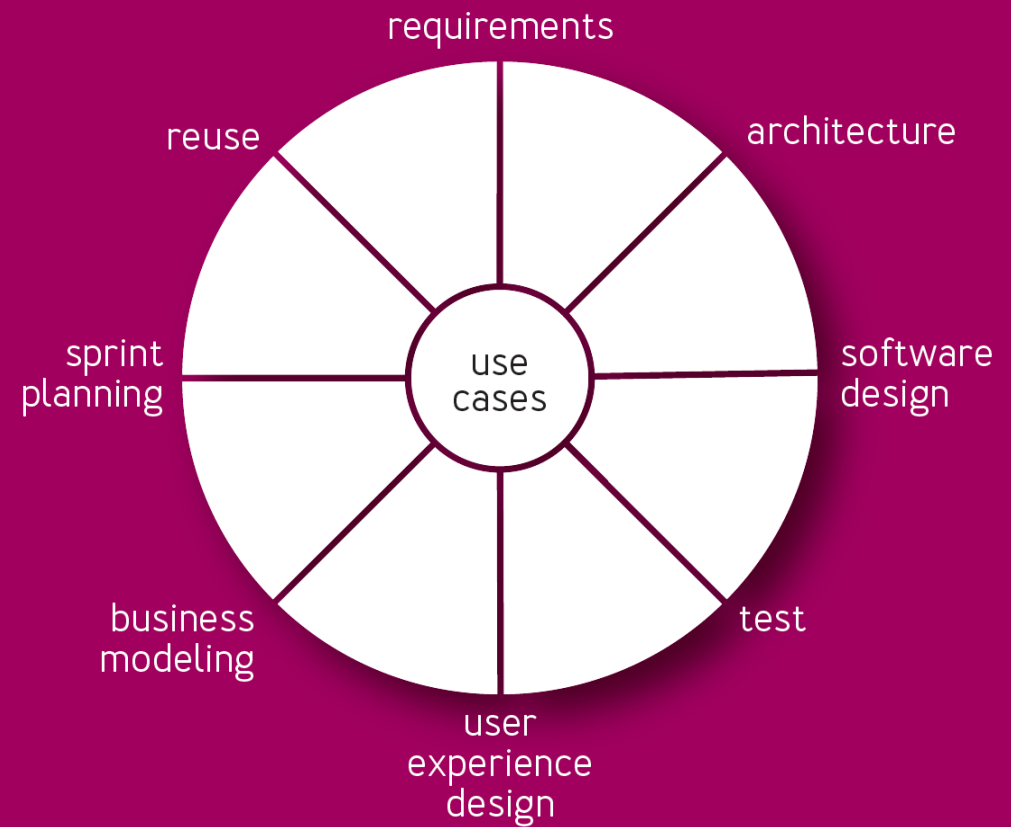
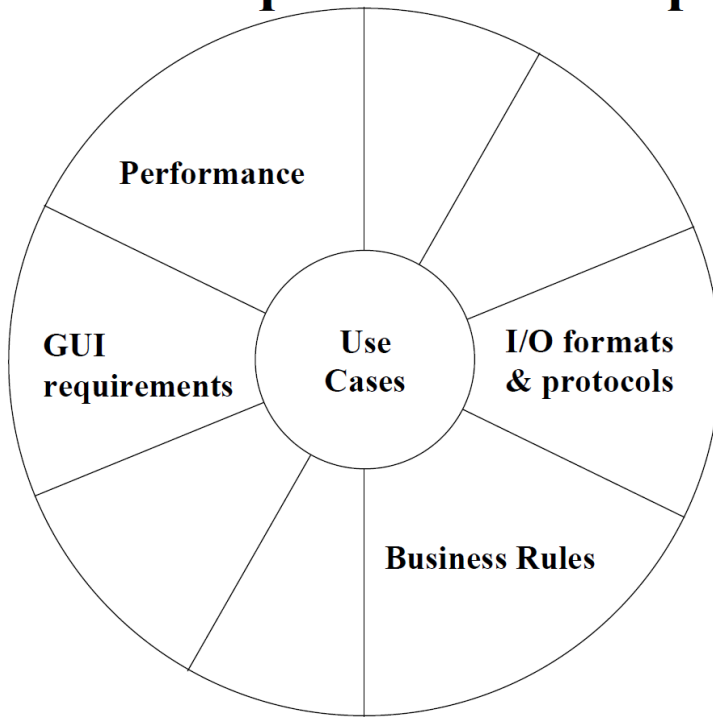
alternative flows

- A1 invalid card
- A2 non-standard amount
- A3 receipt required
- A4 insufficient funds in ATM
- A5 insufficient funds in account
- A6 would cause overdraft
- A7 card stuck
- A8 cash left behind
- etc.

“用例”建模的附加价值

为软件其他需求提高核心指引（架构设计、验收测试、性能、UI、业务模型、重用或重构软件等）

**Use cases form the hub of requirements.
The other requirements are spokes.**



“用例”建模的分层和细化

General Activity	Make Reservation		Book Flight		Request Refund	Manage Hotels Profile	Manage Flight Data
General Epic	Search hotel	Book a room	Search flight	Book a flight	Request refund	Add/ update/ remove hotel	Add/ update/ remove flight data

Release 1



Perform a hotel search	Book a room	Search for a one-way flight	Book a flight		Add a new hotel	Add a new flight
Keep the search settings within the	View and print E-receipt for booking details	Search for a round-trip flight				

Release 2



Use promo code in room booking	Perform a 'multi-city' flight search	Use promo code in flight booking	Request a refund for room booking	Edit the details of a hotel	Edit the details of a flight
--------------------------------	--------------------------------------	----------------------------------	-----------------------------------	-----------------------------	------------------------------

Unscheduled

Enter remarks in booking		Request a refund for flight booking	Remove a hotel	Remove a flight
--------------------------	--	-------------------------------------	----------------	-----------------

Scenario（与系统交互过程）

1. Enter city, arrival, departure, room type and click Search. |
2. **SYSTEM** Display a list of hotels
3. Click on a hotel's logo to read its detail.
4. **SYSTEM** Display hotel details.

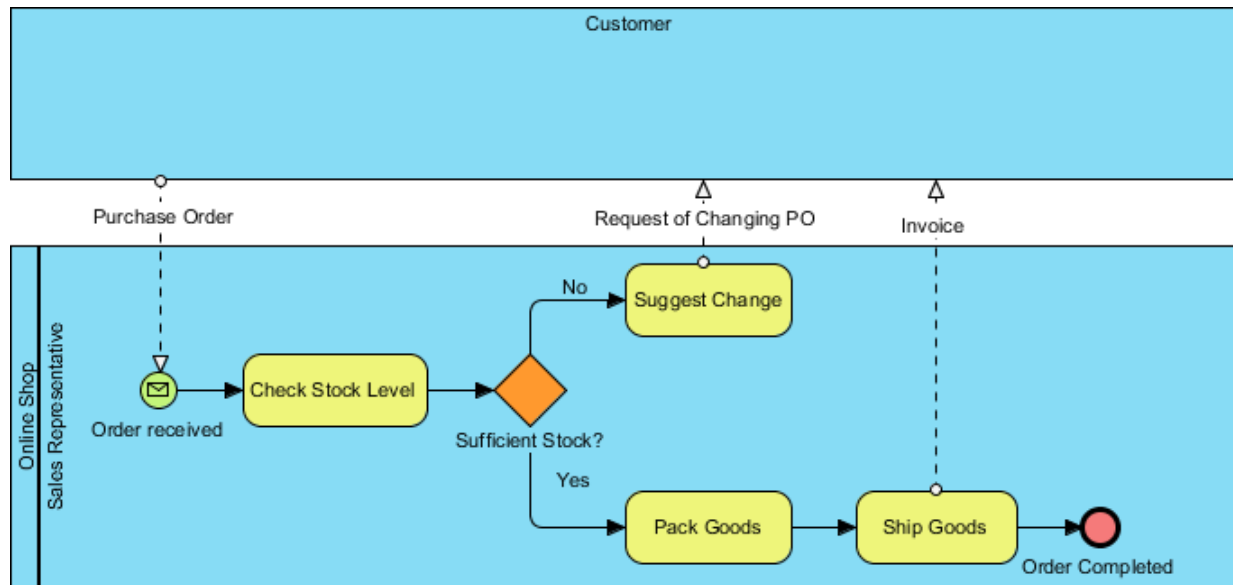
Website Wireframe（用户界面）

A wireframe of a web browser window titled "Browser". The browser has a navigation bar with back, forward, and refresh buttons, and a search bar. The main content area contains a search form with the following fields:

- City: A dropdown menu.
- Arrival: A text input field.
- Departure: A text input field.
- Room type: A dropdown menu.

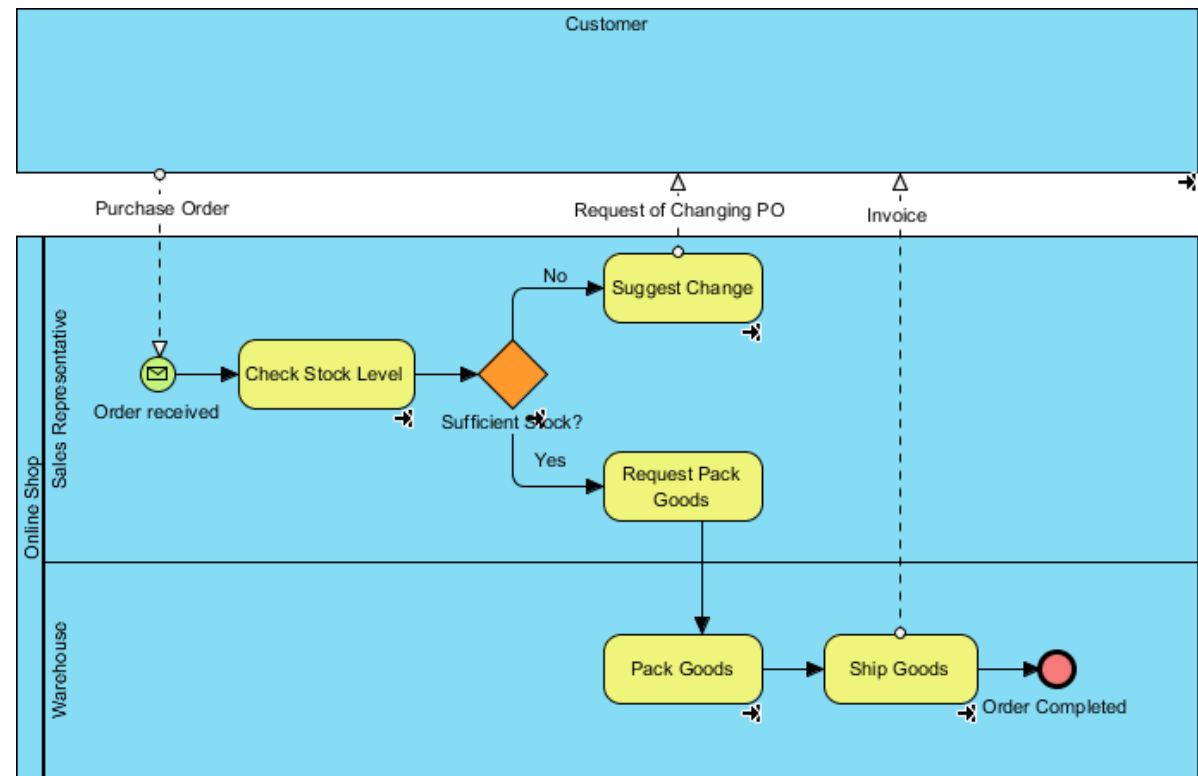
A "Search" button is located at the bottom right of the form.

How to Develop As-Is and To-Be Business Process?

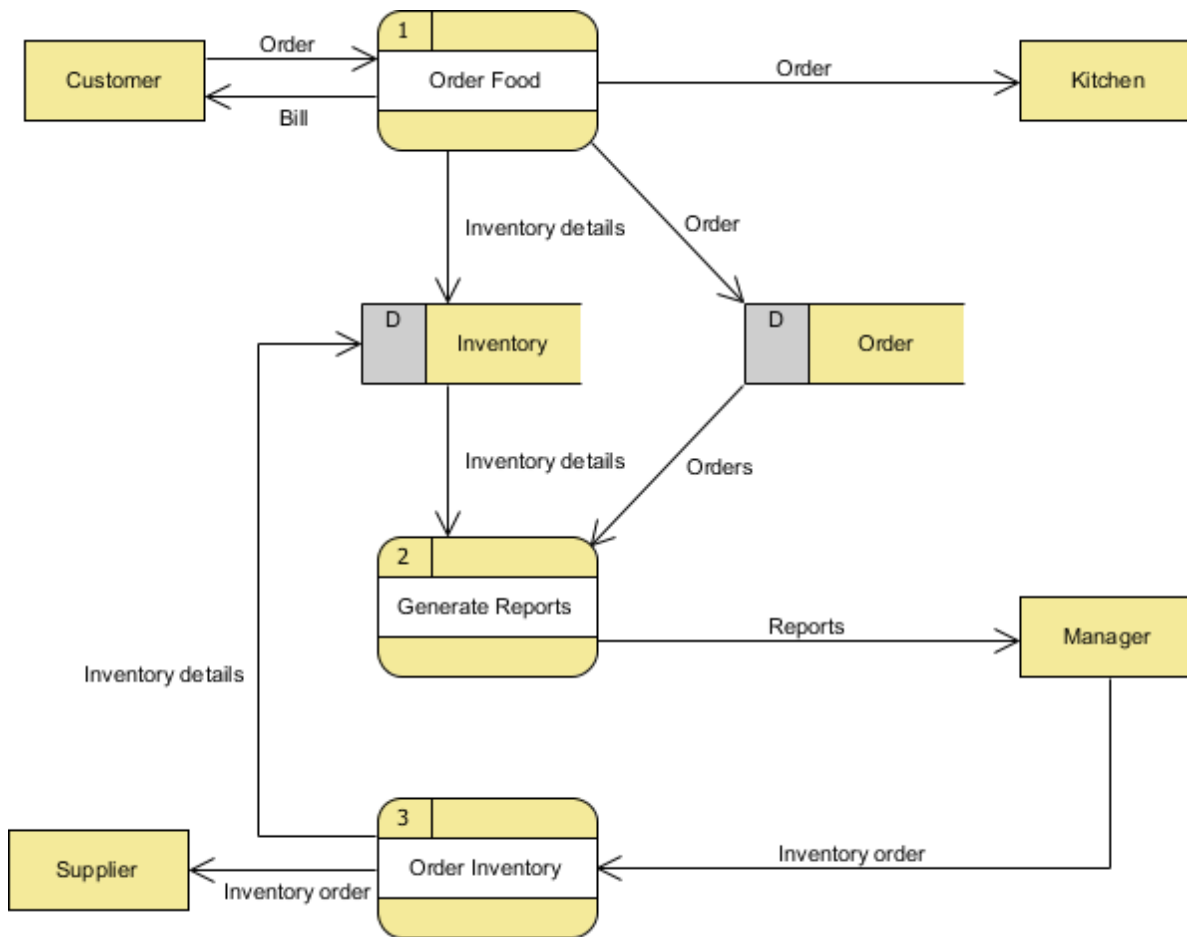


通过创造“用例”这个中间产出物来发现和理解领域知识、设计优化流程。获取利益和实现价值。

One effective technique for transforming vision into results is to develop and populate an As-Is and To-Be BPMN diagrams. The As-Is diagram describes **the present state** of the organization's process, culture, and capabilities. The To-Be diagram describes **the future state**; in other words, how the organization's process, culture, and capabilities will appear in the future. This studying of As-is and To-be process results in **identifying the difference between the current and target business state, known as gap**, which is an important part of any business process reengineering/improvement initiative.



Food Order System Data Flow Diagram



Keep in mind that Data Flow Diagram was designed for representing the exchange of information. Connectors in a Data Flow Diagram are for representing data, not for representing process flow, step or anything else. When we label a data flow that ends at a data store "a request", this means we are passing a request as data into a data store. Although this may be the case in implementation level as some of the DBMS do support the use of functions, which intake some values as parameters and return a result, in Data Flow Diagram, we tend to treat data store as a sole data holder that does not possess any processing capability. If you want to model the system flow or process flow, use **UML Activity Diagram** or **BPMN Business Process Diagram** instead. If you want to model the internal structure of a datastore, use an **Entity Relationship Diagram**.

The Food Order System Data Flow Diagram example contains three processes, four external entities, and two data stores. Based on the diagram, we know that a **Customer** can place an **Order**. The **Order Food** process receives the **Order**, forwards it to the **Kitchen**, store it in the **Order data store**, and store the updated **Inventory details** in the **Inventory** data store. The process also delivers a **Bill** to the **Customer**.

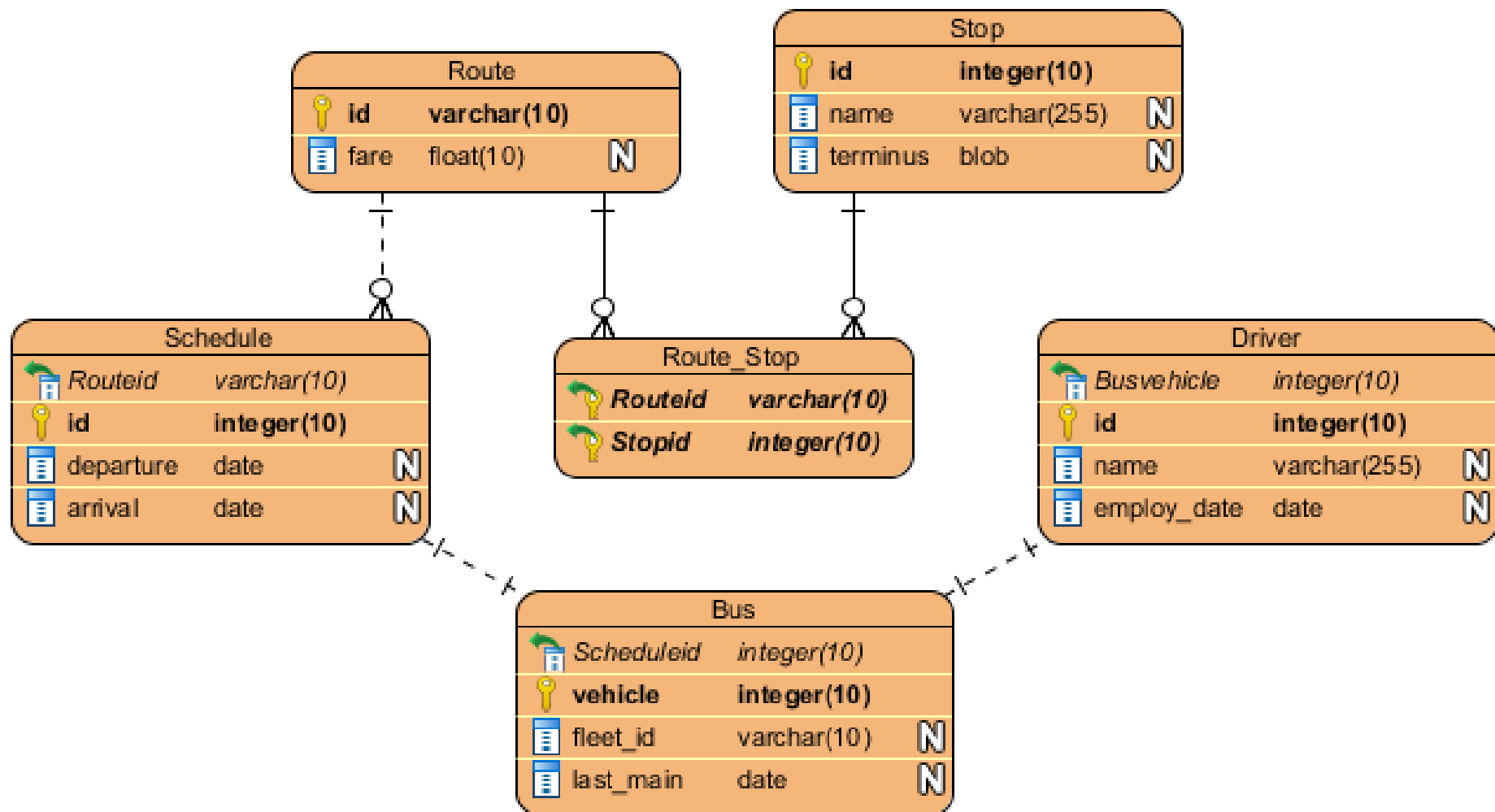
The **Manager** can receive **Reports** through the **Generate Reports** process, which takes **Inventory details** and **Orders** as input from the **Inventory** and **Order** data store respectively.

The **Manager** can also **initiate the Order Inventory process by providing Inventory order**. The process forwards the **Inventory order** to the **Supplier** and stores the updated **Inventory details** in the **Inventory** data store.

Data Modeling

How to Model Relational Database Design with ERD?

The Entity-Relationship (ER) model was originally proposed by Peter Chen in 1976. An entity relationship diagram (ERD) is a graphical representation of entities and their relationships to each other, typically used for modeling the organization of data within databases or information systems.



总结

1. 课堂练习和课外作业点评

创造的前提是学会欣赏和发现特色，围绕3大问题、2种方法和9项任务展开。

2. 如何学习和评价成果

观察、提问、分析和评价、创造软件充分条件。

3. 软件工程文档的案例分析

文档目的作用：理解软件价值、重现的方法、设计理由、测试方法及测试用例。

4. 需求、设计、代码的关系

保持（价值 \longleftrightarrow 用例 \longleftrightarrow 功能 \longleftrightarrow 代码）的一致性；设计是逻辑分层、归纳共性、演绎实例的技艺。

课外练习

1. 写出课程项目中的5个候选题目的应用场景和软件价值（由你自己创作和构思）。
2. 阅读“Build Chatbot Interactions”书，在书里介绍的十个任务中，请选择一个来写软件应用场景和软件需求（找出用例并描述清楚）。
3. 2021年9月29日晚10点前在课程网站上提交。