



New Media Data Analytics and Application

Lecture 4: Software Engineering

Ting Wang

- The Process of Software Development
- System Structure Design
- Testing
- Team Management





上海外国语大学
SHANGHAI INTERNATIONAL STUDIES UNIVERSITY

a management approach to software engineering

The Process of Software Development

The Process of Software Development

Software Crisis

- The First NATO Software Engineering Conference in 1968, Germany.
- How to cope with the difficulty of writing useful and efficient computer programs in the required time.



The Process of Software Development

Difficulties in Software Development

1. Projects running over-budget
2. Projects running over-time
3. Software was very inefficient
4. Software was of low quality
5. Software often did not meet requirements
6. Projects were unmanageable and code difficult to maintain
7. Software was never delivered



The Process of Software Development

What is Software Engineering

- Software engineering is the application of engineering to the **design, development, implementation, testing and maintenance** of software in a systematic method.

From Wikipedia



The Process of Software Development

Basic Elements in Software Engineering

- Development Stages
- Management Pipeline
- Demands Changing
- Cooperative Team Work
- Professional Expert Participation



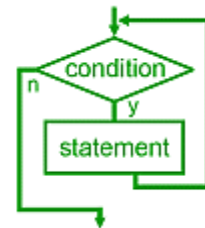
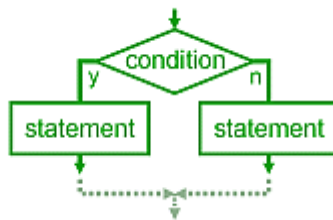
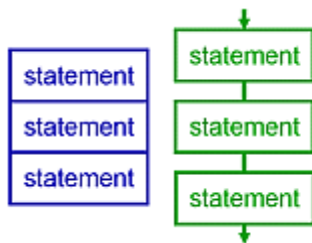
The Process of Software Development

Programming Paradigm 编程范式

- Structured Programming (1) 结构化编程

- Control Structure

1. Sequence
2. Selection: *if..then..else..endif*, *switch*
3. Iteration: *while*, *repeat*, *for*, *do...until*



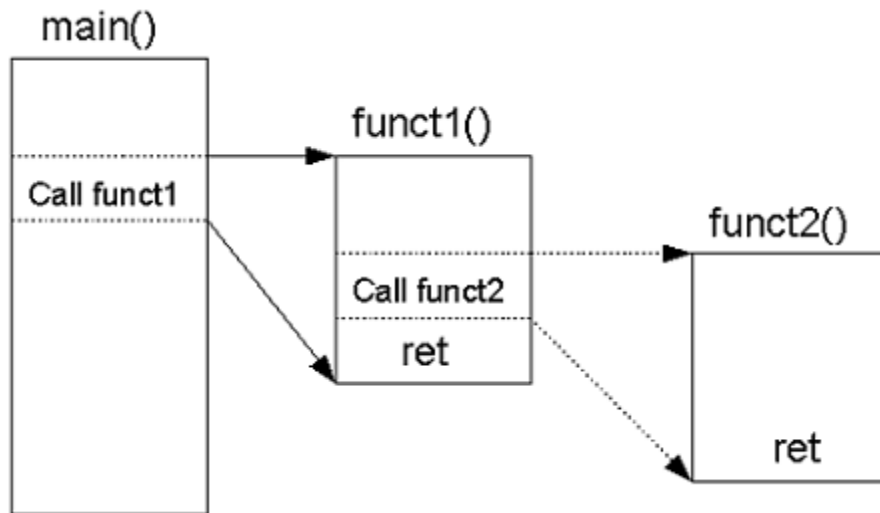
The Process of Software Development

Programming Paradigm 编程范式

- Structured Programming (2) 结构化编程

- Subroutines

子程序



The Process of Software Development

Programming Paradigm 编程范式

- Object Oriented Programming(1)

面向对象编程 针对物件的编程

- Object 对象
- Class 类
- Attribute 属性
- Method 方法



The Process of Software Development

Programming Paradigm 编程范式

- Object Oriented Programming(2)

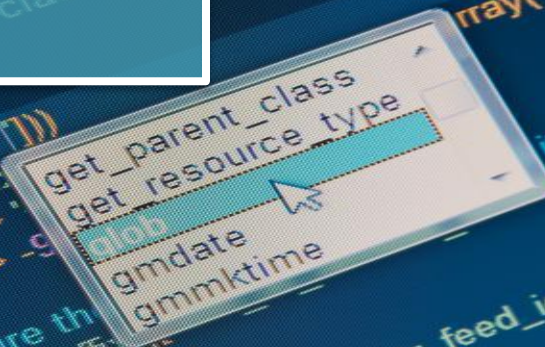
面向对象编程

- Encapsulation 封装
- Inheritance 继承
- Polymorphism 多态



The Process of Software Development

EXAMPLE 1: Class



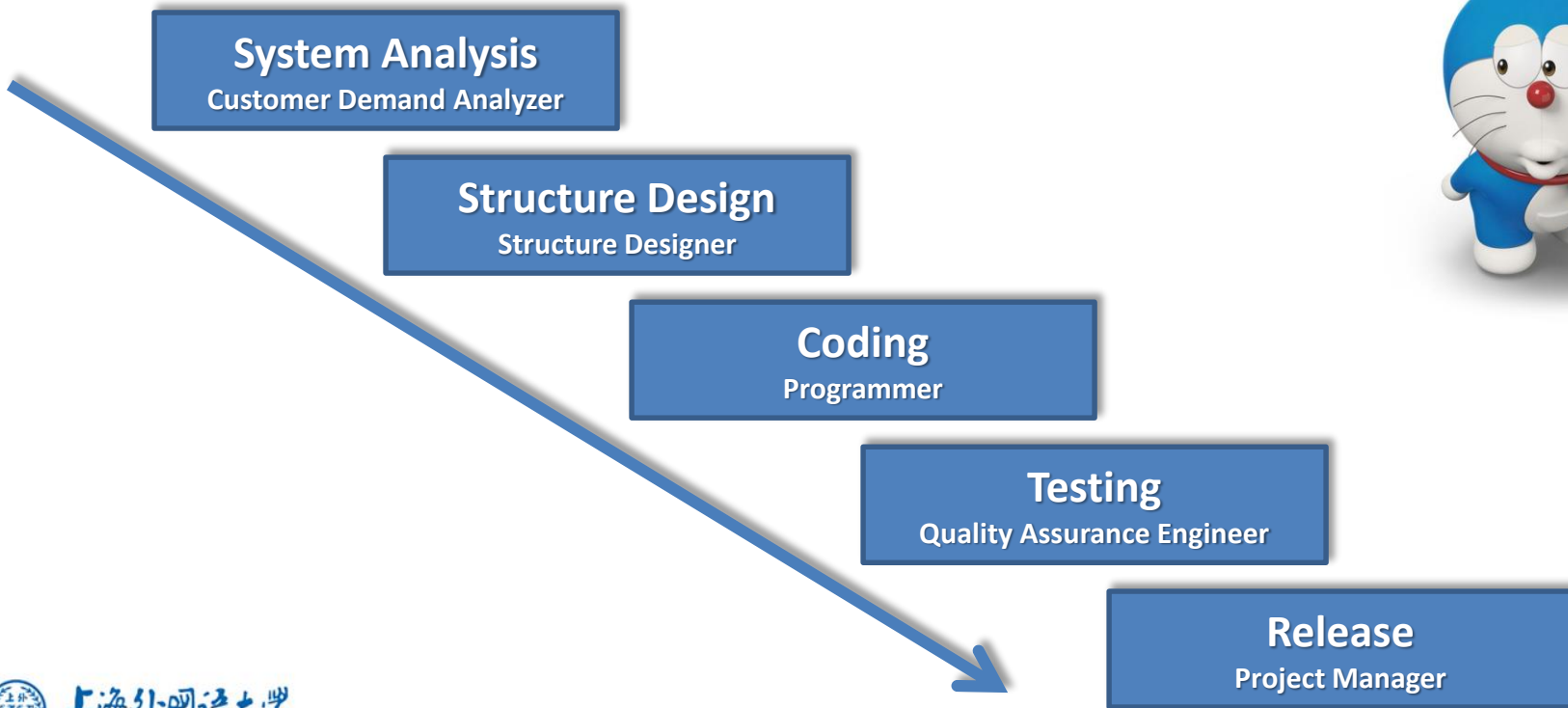
The Process of Software Development

```
1 class Customer(object):
2
3     name = ''
4     password = ''
5
6     def __init__(self, name, password):
7         self.__name = name
8         self.__password = password
9         print('Name: %s' %self.__name)
10
11     def get_validation(self, password):
12         if password=='sisu':
13             return 1
14         else:
15             return 0
16
17 class Student(Customer):
18
19     name = ''
20     password = ''
21     studentID = ''
22
23     def __init__(self, name, password, studentID):
24         Customer.__init__(self, name, password)
25         self.studentID = studentID
26
27     def print_studentID(self):
28         return self.studentID
29
30     def get_validation(self, password):
31         if password=='shisu':
32             return 'Passed'
33         else:
34             return 'Failed'
35
36 Thomas = Customer('Thomas Edison', 'sisu')
37 print('Thomas.get_validation() =', Thomas.get_validation('sisu'))
38 Albert = Student('Albert Einstein', 'sisu', '20160001')
39 print('Albert.print_studentID() =', Albert.print_studentID())
40 print('Albert.get_validation() =', Albert.get_validation('sisu'))
41
```



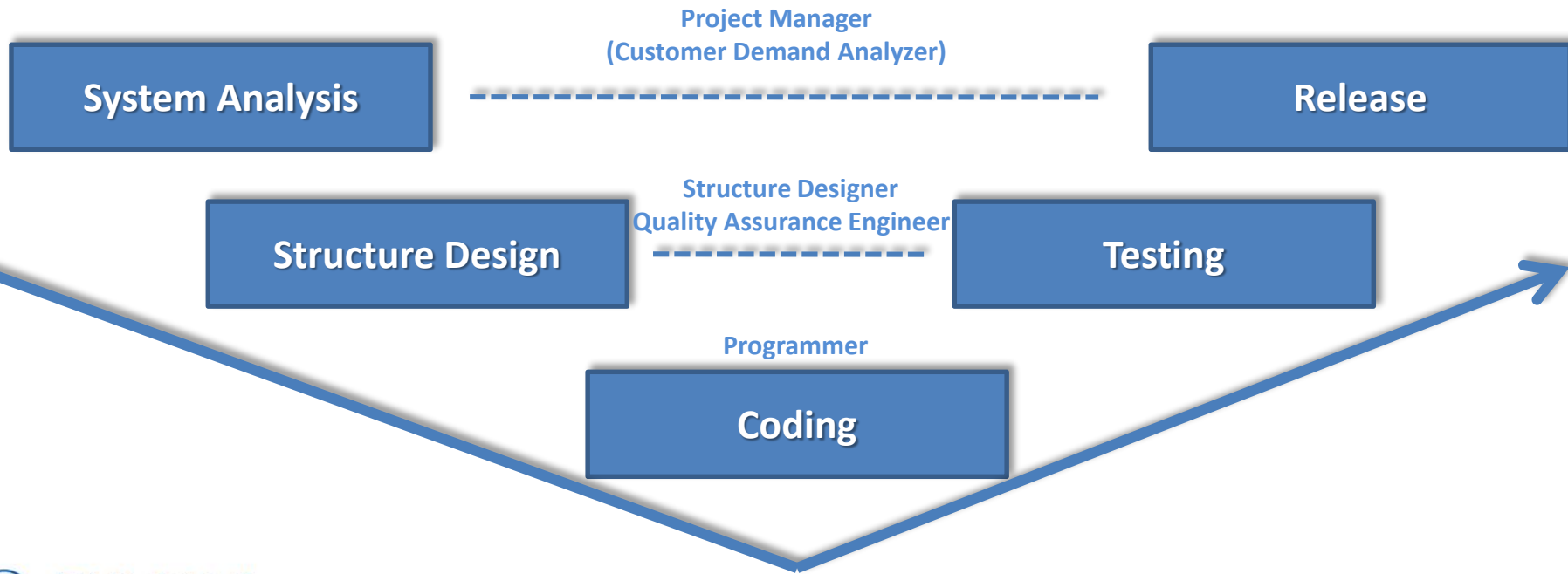
The Process of Software Development

- Water Fall Model



The Process of Software Development

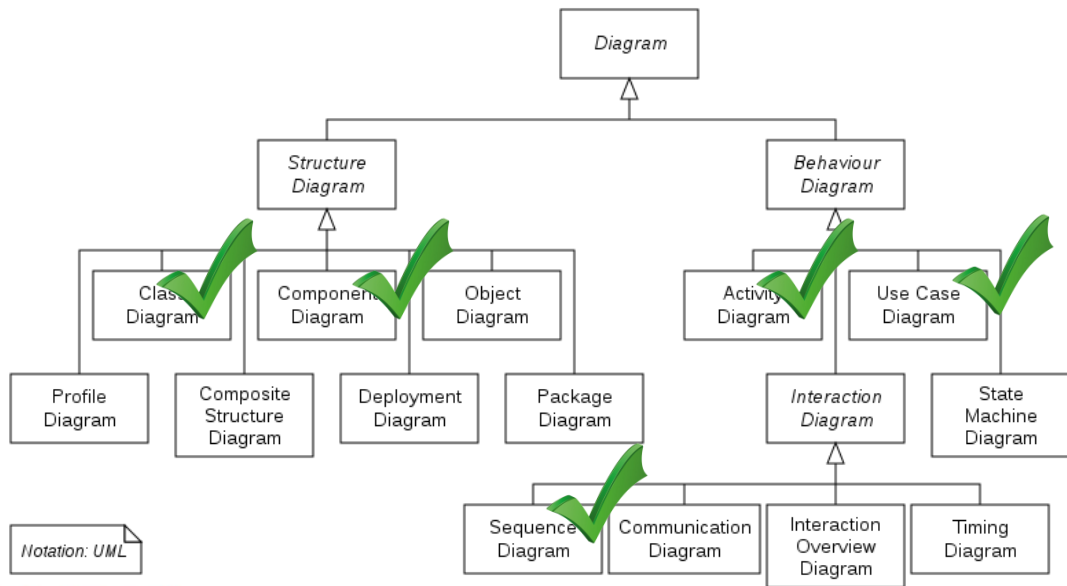
- V-Model



The Process of Software Development

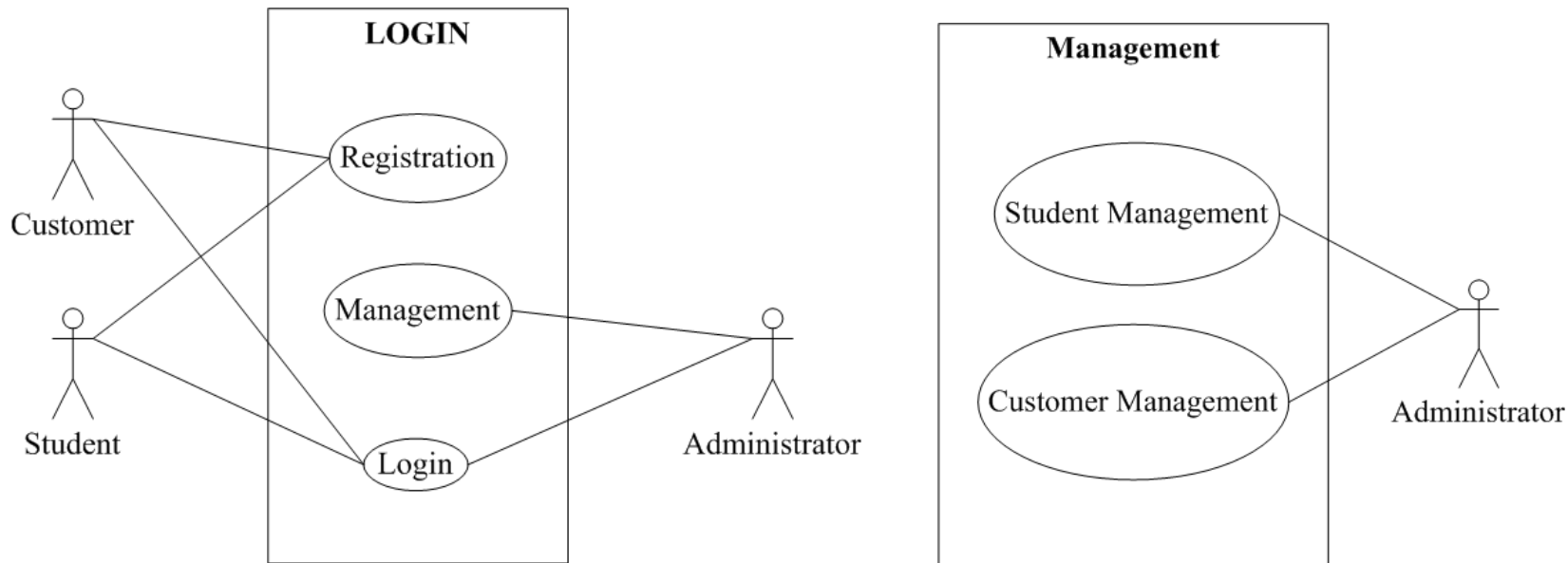
Unified Modeling Language (UML)

A general-purpose, developmental, modeling language in the field of software engineering, that is intended to provide a standard way to visualize the design of a system



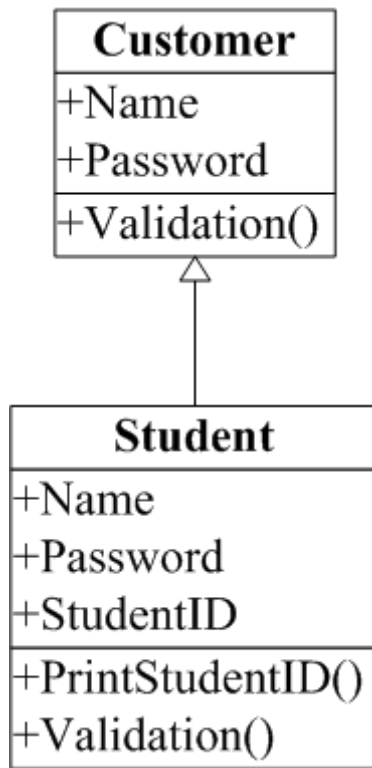
The Process of Software Development

- Use Case Diagram



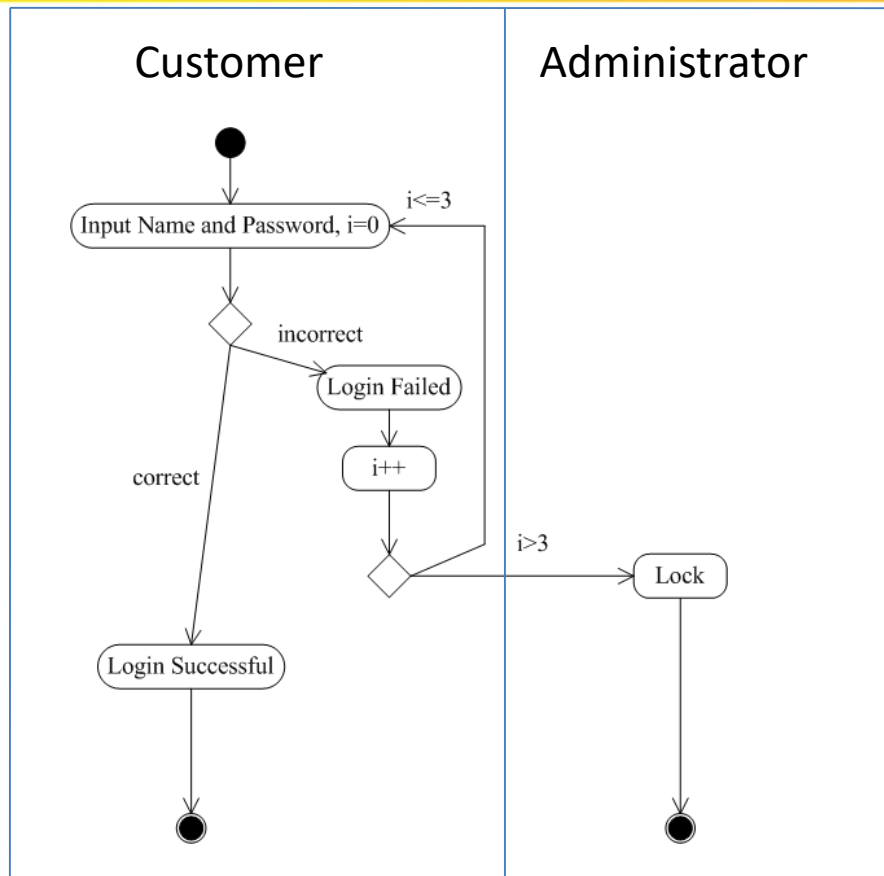
The Process of Software Development

- Class Diagram



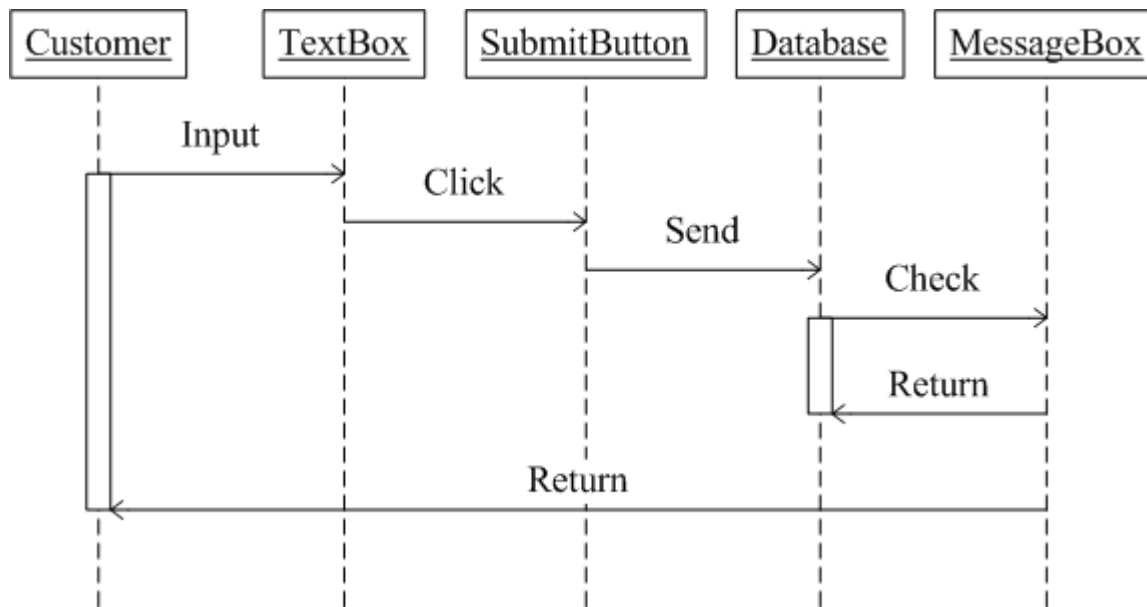
The Process of Software Development

- Activity Diagram



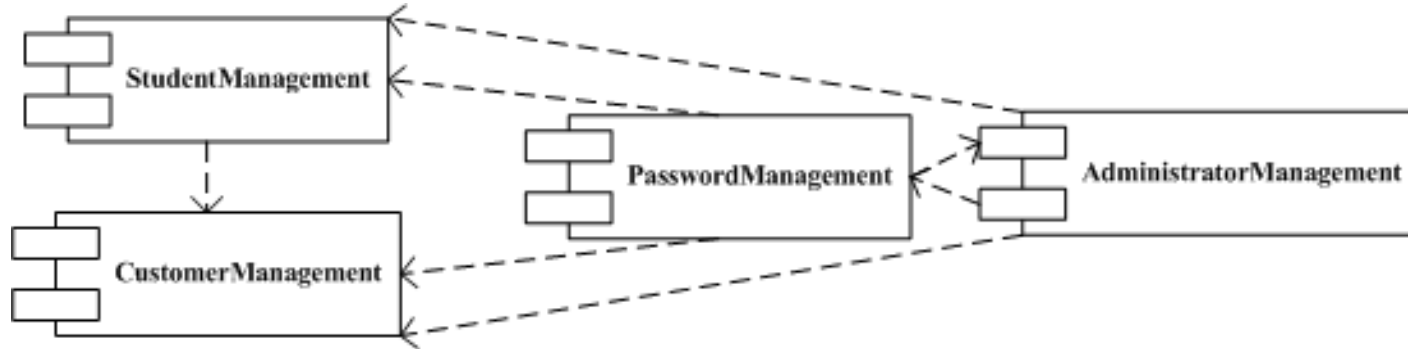
The Process of Software Development

- Sequence Diagram



The Process of Software Development

- Component Diagram



The Process of Software Development

- Function Structure Diagram



The Process of Software Development

Time Estimation for Software Projects

- Man-Month
- Man-Day
- Basic Function: Insert, Delete, Update, Select
 - Slow: *1 Basic Function per day*
 - Common: *2 Basic Functions per day*
 - Fast: *4 Basic Functions per day*





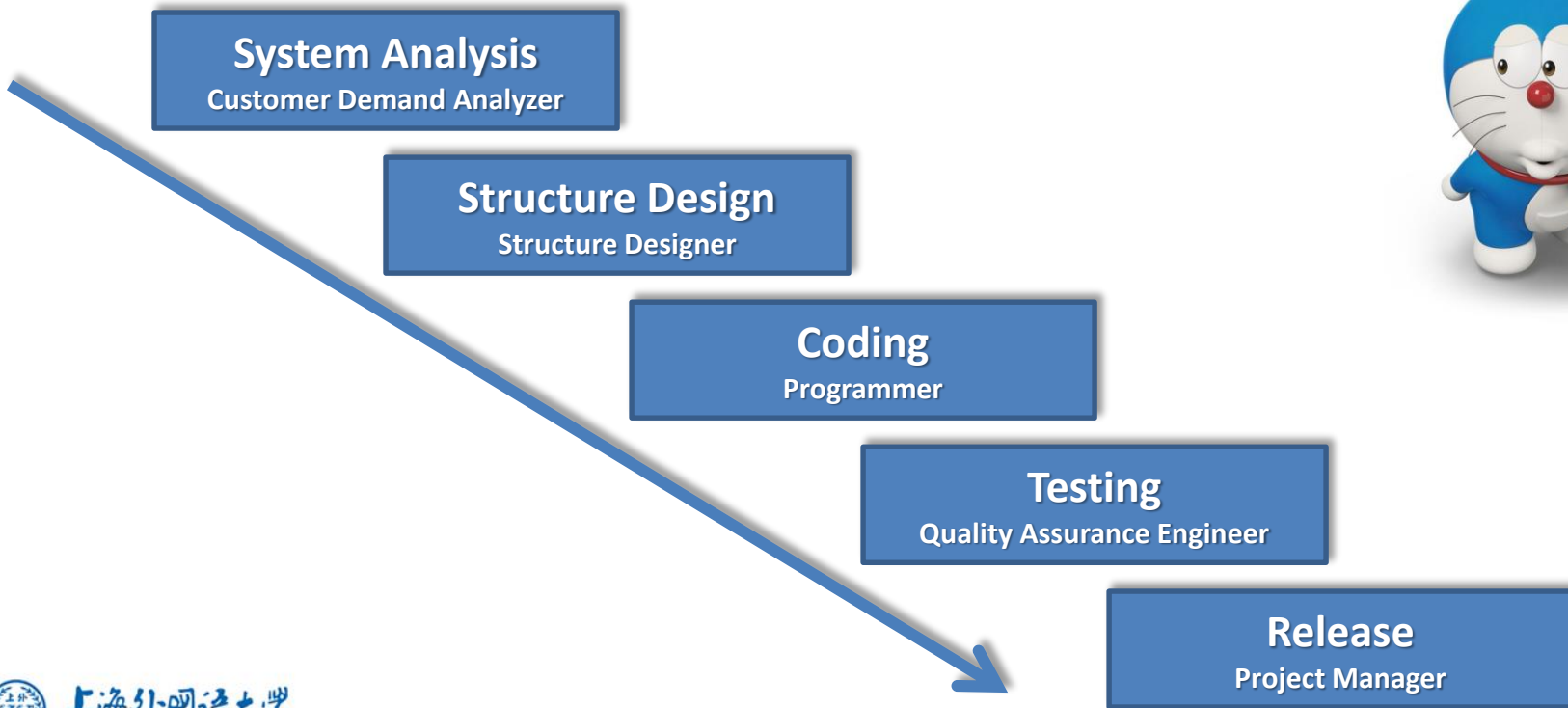
上海外国语大学
SHANGHAI INTERNATIONAL STUDIES UNIVERSITY

links from the world to the systems

System Structure Design

System Structure Design

A Review: Water Fall Model



System Structure Design

Two Sub-stages:

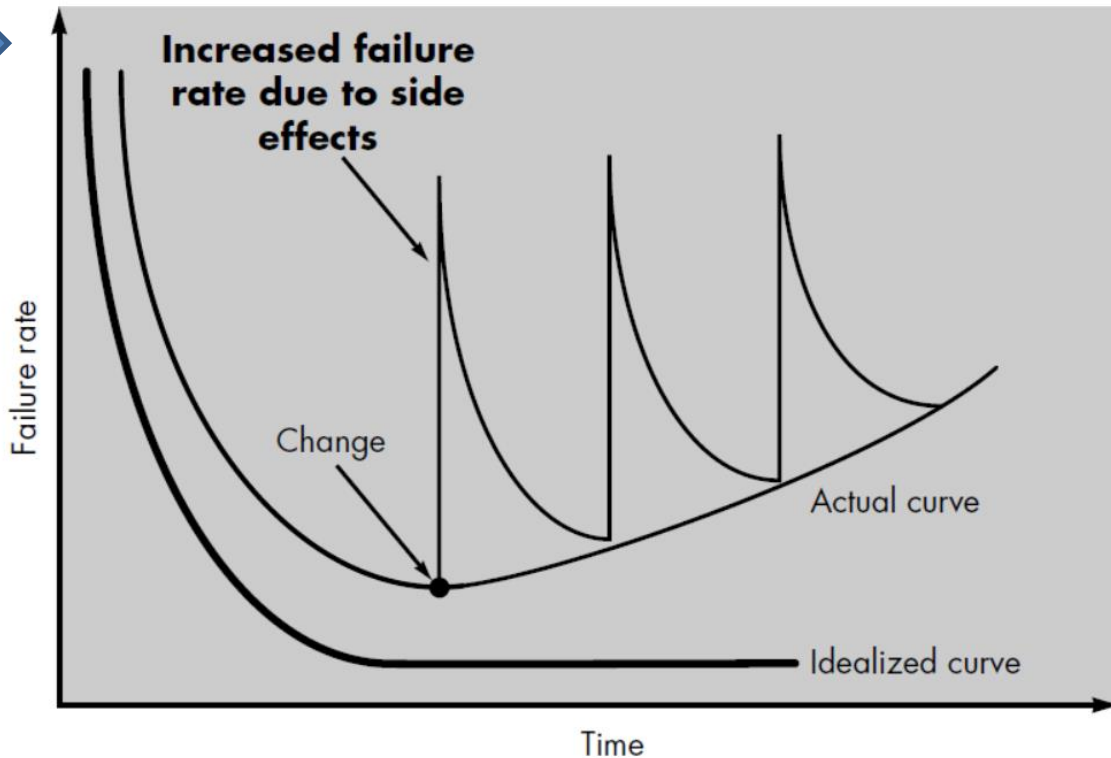
1. Overall Design **总体设计**
General Design 概要设计
2. Detailed Design **详细设计**



System Structure Design

Why Overall Design?

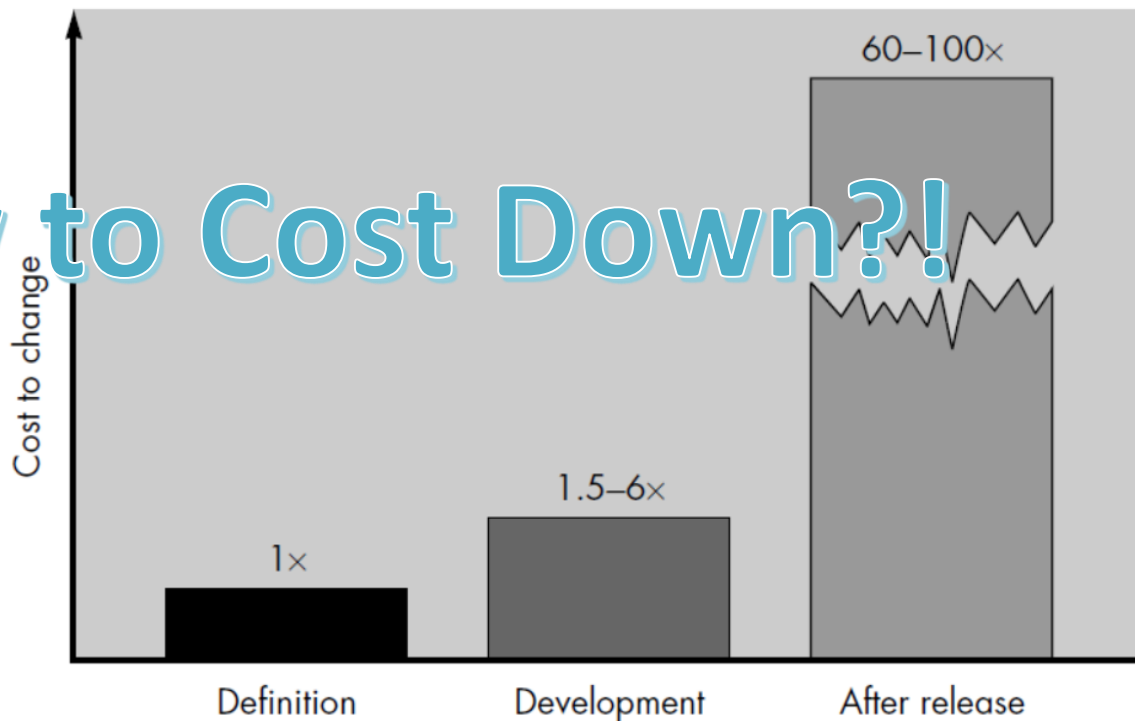
Idealized and actual failure curves for software



System Structure Design

Cost Change

How to Cost Down?!



System Structure Design

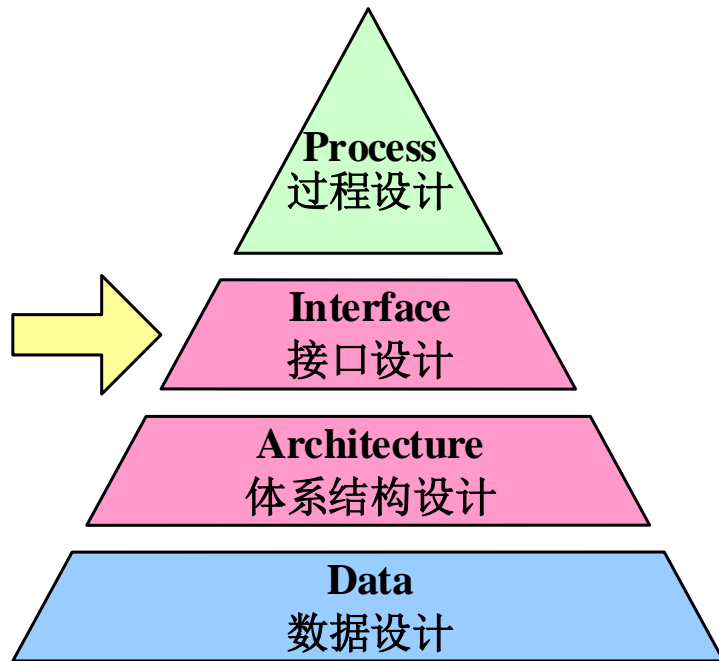
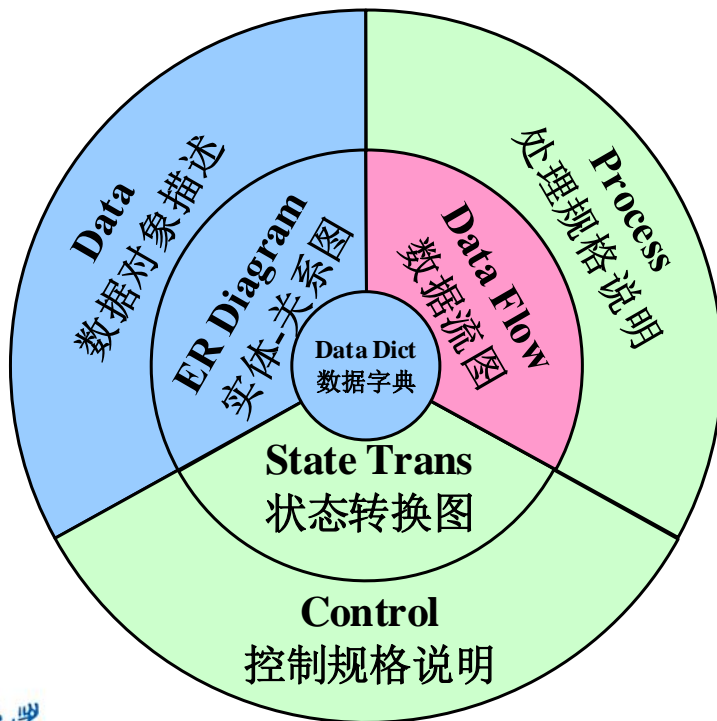
Overall Design

- Overall Design aims to propose an optimal project plan for software products, which can reduce the cost and enhance the quality.



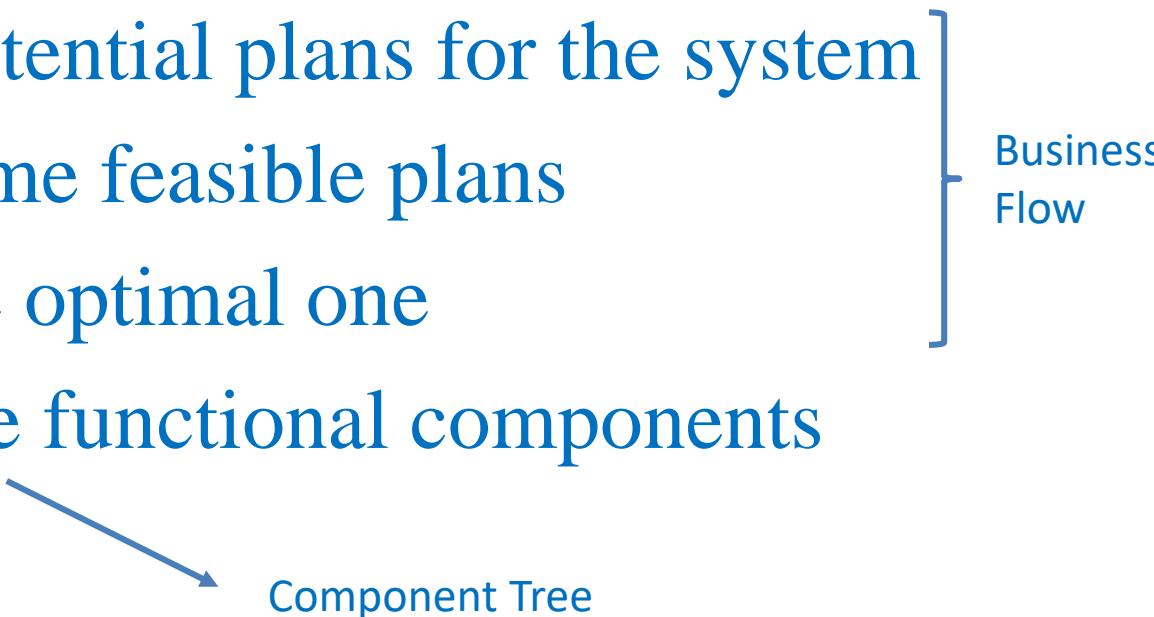
System Structure Design

Overall Design



System Structure Design

Steps to Overall Design

1. To list all potential plans for the system
 2. To select some feasible plans
 3. To select the optimal one
 4. To define the functional components
- 
- The first three steps (1, 2, and 3) are grouped by a large right-facing curly bracket. To the right of this bracket is the text "Business Flow". An arrow points from the fourth step, "To define the functional components", down and to the right towards the text "Component Tree".

Component Tree



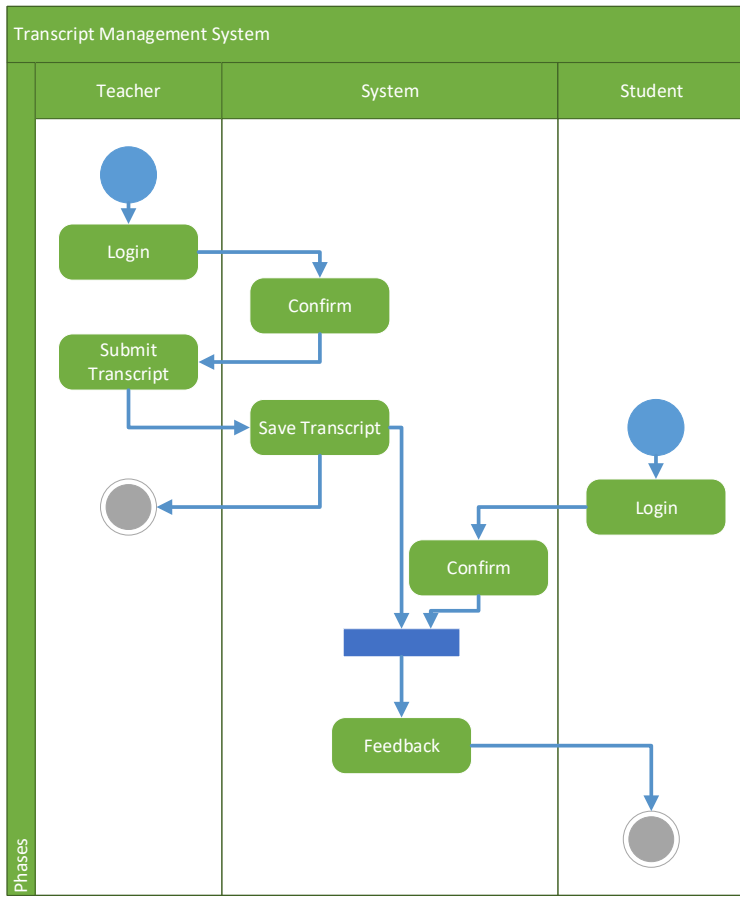
System Structure Design

Business Flow

Review:

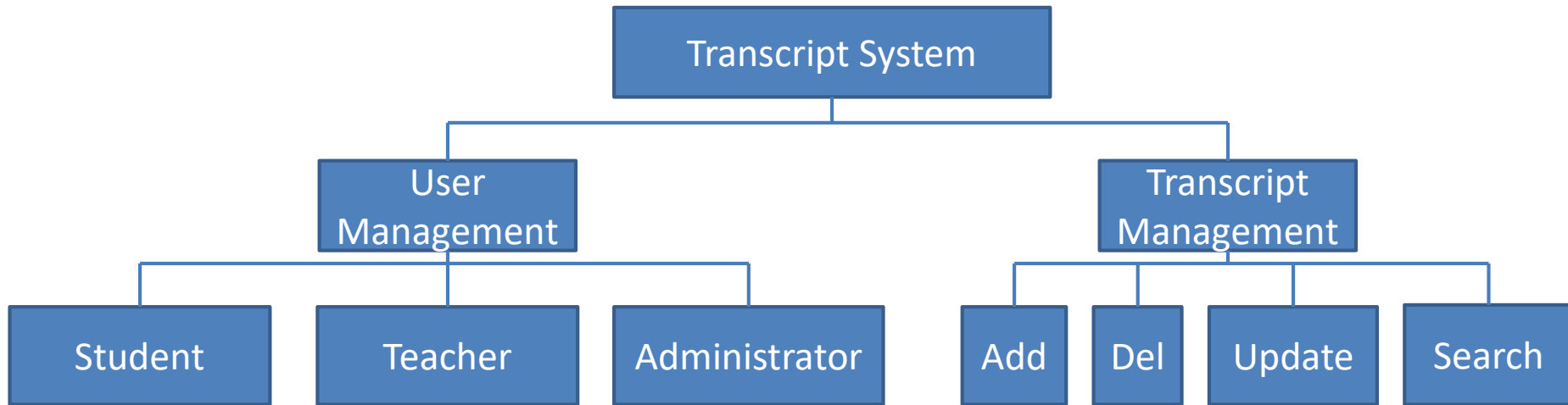
UML: Activity Diagram

Other Corresponding Diagrams in Visio:



System Structure Design

Component Tree



System Structure Design

Detailed Design

Detailed Design aims to divide each function to different subsystems, and decide the corresponding algorithms for each function.



System Structure Design

Steps to Detailed Design

1. To design the system —→ Interface, Use Case, Data Flow, Sequence
2. To design the data bases —→ ER Diagram, Database Doc
3. To make the testing plan —→ Test Plan Doc
4. To write the progress reports
5. Review



System Structure Design

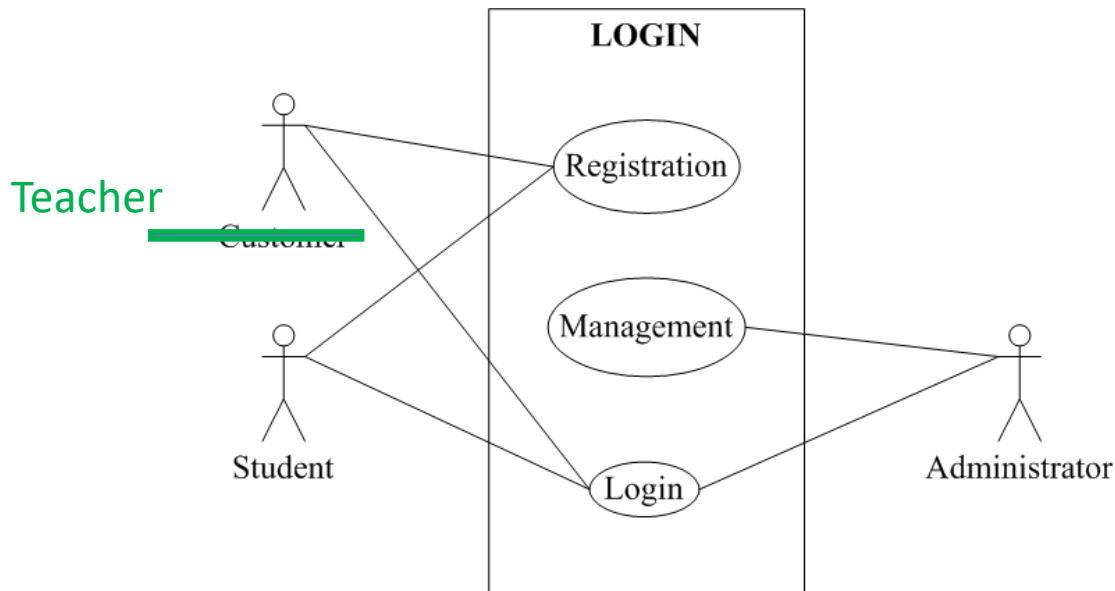
Interface and Prototype Design

- **Axure RP Pro** is a wireframing, rapid prototyping, documentation and specification software tool aimed at web and desktop applications.
- **References**
<https://www.axure.com/>
<https://www.axure.com.cn/>
- **Download**
<https://www.axure.com/download>



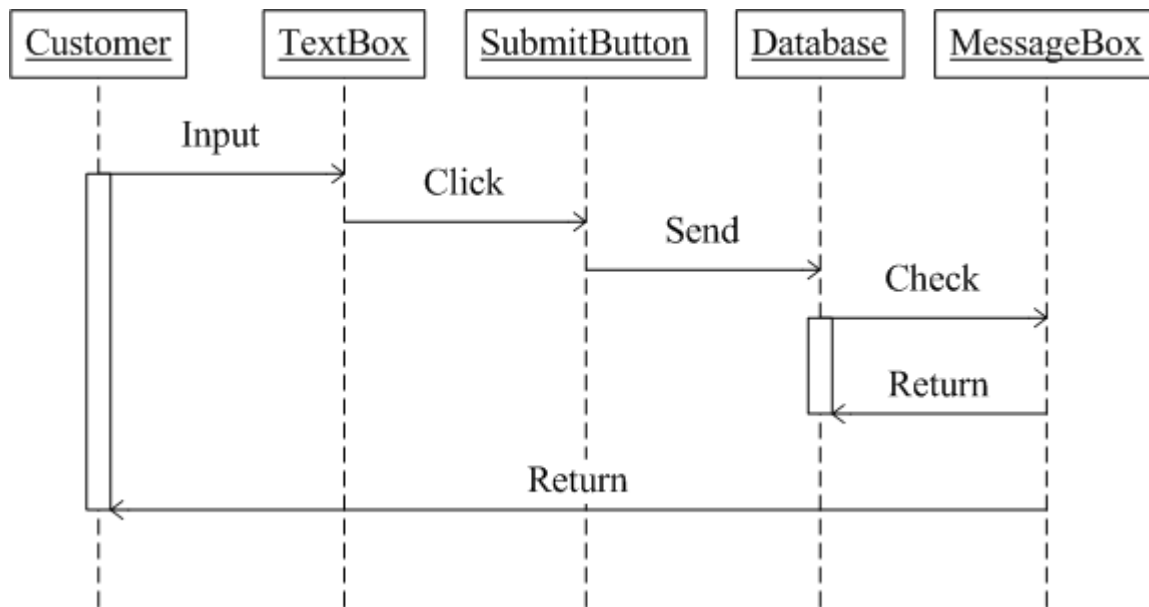
System Structure Design

Review: Use Case Diagram



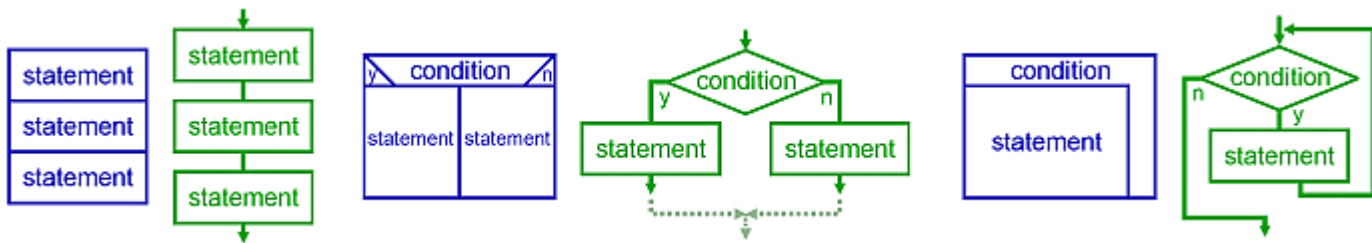
System Structure Design

Review: Sequence Diagram

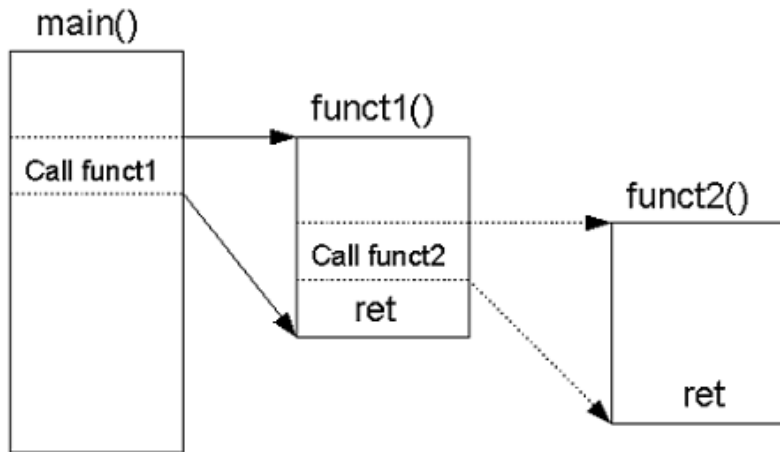


System Structure Design

Data Flow for Algorithms

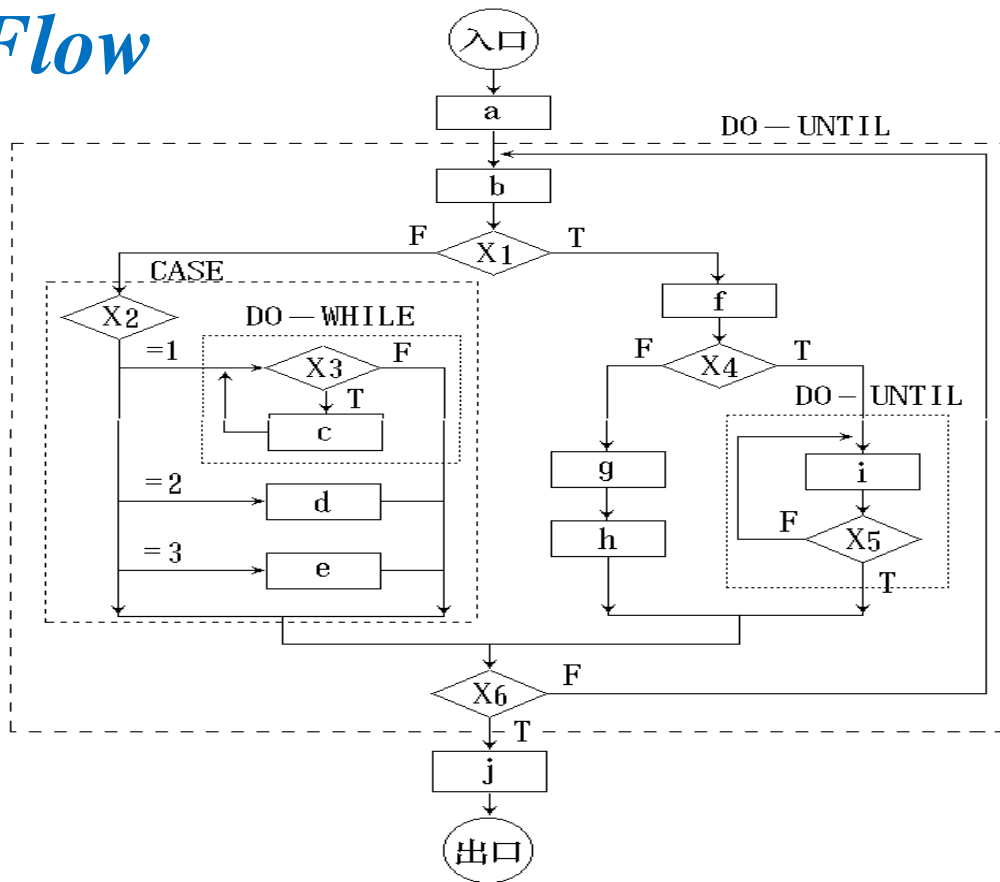


Reuse



System Structure Design

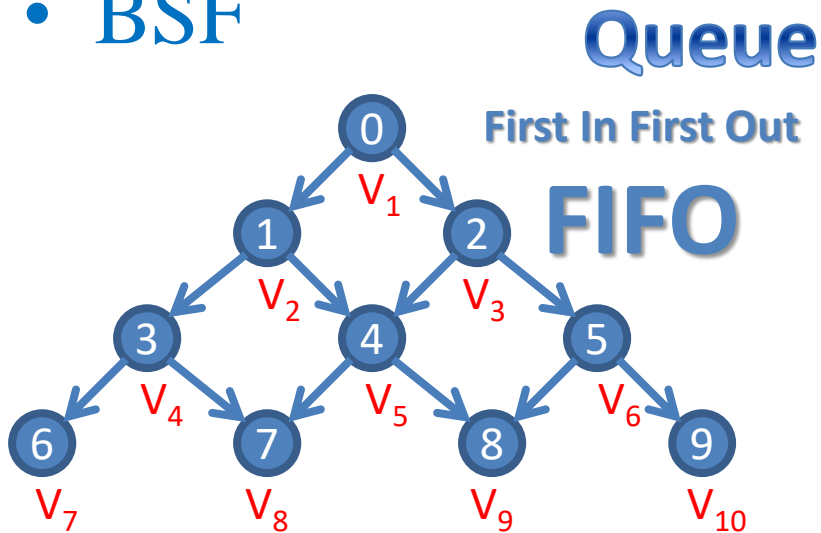
Example of Data Flow



System Structure Design

Pseudo-code for Algorithm Description

- BSF



Algorithm Breadth-First Search (BFS)

Require: Initial node v , graph/tree $G(V; E)$, queue Q

1: return An ordering on how nodes are visited

2: Enqueue v into queue Q ;

3: $\text{visitOrder} = 0$;

4: while Q not empty do

5: $\text{node} = \text{dequeue from } Q$;

6: if node not visited then

7: $\text{visitOrder} = \text{visitOrder} + 1$;

8: Mark node as visited with order visitOrder ;
 //or print node

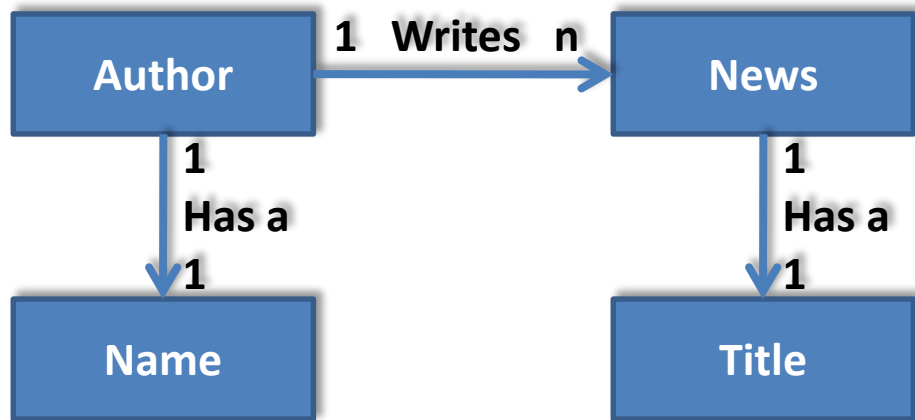
9: Enqueue all neighbors/children of node into Q ;

10: end if

11: end while

System Structure Design

Review: ER Diagram



System Structure Design

Database Document



档案名称。	SYS_ADMIN_MESSAGE。					
档案用途。	管理留言资料档。					
主键(PK)。	SYS_ADMIN_MESSAGE_PK: MESSAGE_ID(Cluster Index)。					
附键(AK)。						
INDEX NAME。	栏位。	用途。				
SYS_ADMIN_MESSAGE_FK1。	MESSAGE_FROM。	FK: ADMIN_INFO(ADMIN_ID)。				
SYS_ADMIN_MESSAGE_FK2。	MESSAGE_TO。	FK: ADMIN_INFO(ADMIN_ID)。				
序号	栏位名称。	栏位说明。	资料形态。	长度。	Null。	Default。
01	MESSAGE_ID。	留言编号。	Number。		X。	
02	MESSAGE_NAME。	留言标题。	Char。	200。	X。	
03	MESSAGE_INFO。	留言内容。	Text。			
04	MESSAGE_TO。	收言人员编号。	Number。		X。	
05	READ_FLAG。	已读标识。	Number。		X。	0。
06	STATE。	状态。	Number。		X。	0。
06	CREATE_USER_ID。	创建人编号。	Number。		X。	1。
07	CREATE_DATE。	创建日期。	Date。		X。	
08	UPDATE_DATE。	更新日期。	Date。		X。	

[注:] 已读标识: 0-未读, 1-已读, 2已删, 3为彻底删除。

状态: 0-正常, 1-已删除, 2为彻底删除。



System Structure Design

Principles:

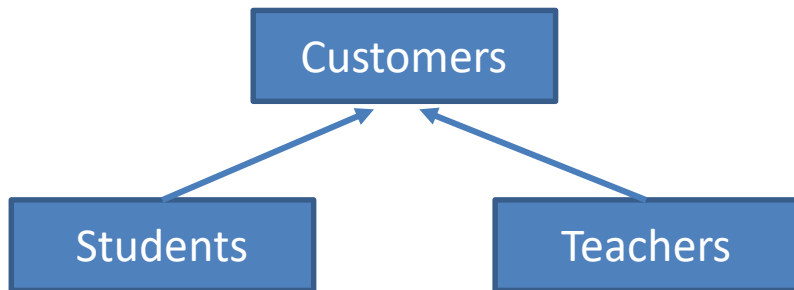
1. Abstraction 抽象
2. Information Hiding and Localization
信息隐藏与局部化
3. Modularity 模块化
4. Refinement 自顶向下，逐步求精



System Structure Design

Abstraction 抽象

1. Extract the same parts from different things



2. Give levels to analyze them



Grady Booch
IBM Fellow

“Abstraction is one of the fundamental ways that we as humans cope with complexity.”

—Grady Booch



System Structure Design

Information Hiding and Localization

信息隐藏与局部化

- Modules should be specified and designed so that information contained within a module is inaccessible to other modules that have no need for such information.



System Structure Design

Modularity 模块化

1. Divide and Conquer
2. Software architecture is divided into components called modules.

- *Low Coupling, High Cohesion*
低耦合，高内聚



System Structure Design

Refinement

自顶向下，逐步求精

- It is the process of elaboration. A hierarchy is developed by decomposing a macroscopic statement of function in a step-wise fashion until **programming language statements are reached**. In each step, one or several instructions of a given program are decomposed into more detailed instructions. Abstraction and Refinement are complementary concepts.



System Structure Design

Complexity of the Algorithms

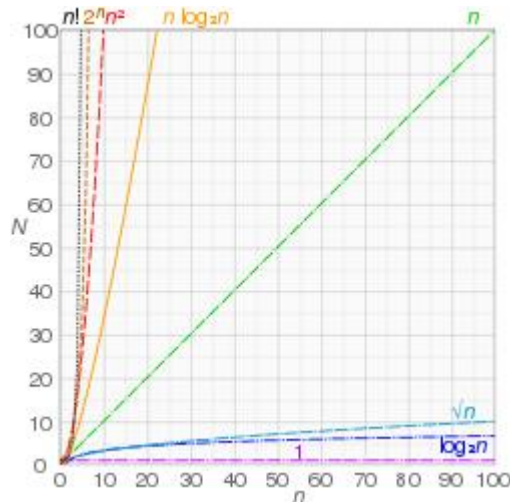
- Time Complexity 时间复杂度

```
1 sum = n*(n+1)/2;           //时间复杂度O(1)
```

```
1 for(int i = 0; i < n; i++){  
2     printf("%d ",i);  
3 }  
4 //时间复杂度O(n)
```

```
1 for(int i = 0; i < n; i++){  
2     for(int j = 0; j < n; j++){  
3         printf("%d ",i);  
4     }  
5 }  
6 //时间复杂度O(n^2)
```

```
1 int i = 1, n = 100;  
2 while(i < n){  
3     i = i * 2;  
4 }  
5 //设执行次数为x.  $2^x = n$  即  $x = \log_2 n$   
6 //时间复杂度O(log2n)
```



System Structure Design

- Space Complexity 空间复杂度

- Relevant to Time Complexity

- Including:

- Initialized data
- Algorithm data
- Some additional data





上海外国语大学
SHANGHAI INTERNATIONAL STUDIES UNIVERSITY

software quality assurance

Testing

A Review: Water Fall Model

System Analysis
Customer Demand Analyzer

Structure Design
Structure Designer

Coding
Programmer

Testing
Quality Assurance Engineer

Release
Project Manager



Testing Preparation Stages



Testing Plan

- **Testing Case** is a specification of the inputs, execution conditions, testing procedure, and expected results that define a single test to be executed to achieve a particular software testing objective, such as to exercise a particular program path or to verify compliance with a specific requirement.
- **Automatic Testing Tools**
 - Web: selenium, QTP
 - Function: loadrunner, jmeter
 - Interface: SoapUI, postman
 - Cellphone: robotium, appium



Testing Types and Stages:

1. White Box: Programmer
 2. Black Box: Programmer and Testing Engineer (same group)
 3. Integration Testing: Programmer and Testing Engineer (different groups)
 4. Regression Testing: Programmer and Testing Engineer
 5. Release Testing: Testing Engineer (all groups)
 6. Disaster Recovery Testing : Testing Engineer
 7. Alpha Testing: Testing Engineer (all groups)
 8. Beta Testing: User
- Developing Coding Stages
- Testing Stages

Testing Report

- Introduction
- Testing Results
- Results Analysis
- Conclusions
- Cost and Consumption





finish your project before the deadline

Project Progress Management

Project Progress Management

*When you want to start a new project,
you should know:*

1. When is the deadline;
2. How many people you have;
3. How many components in this projects;
4. Which components can be done in parallel;
5. Risks.



Project Progress Management

How to do:

- To give out the schedule start from the deadline
- To know the advantages of your team members
- To divide the system into components
- To avoid the risks

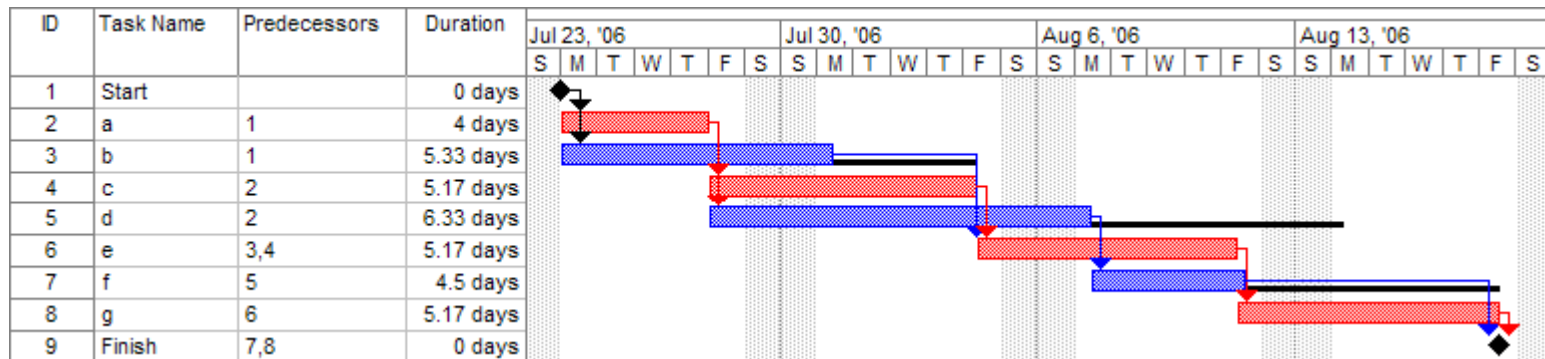


Project Progress Management

Diagrams

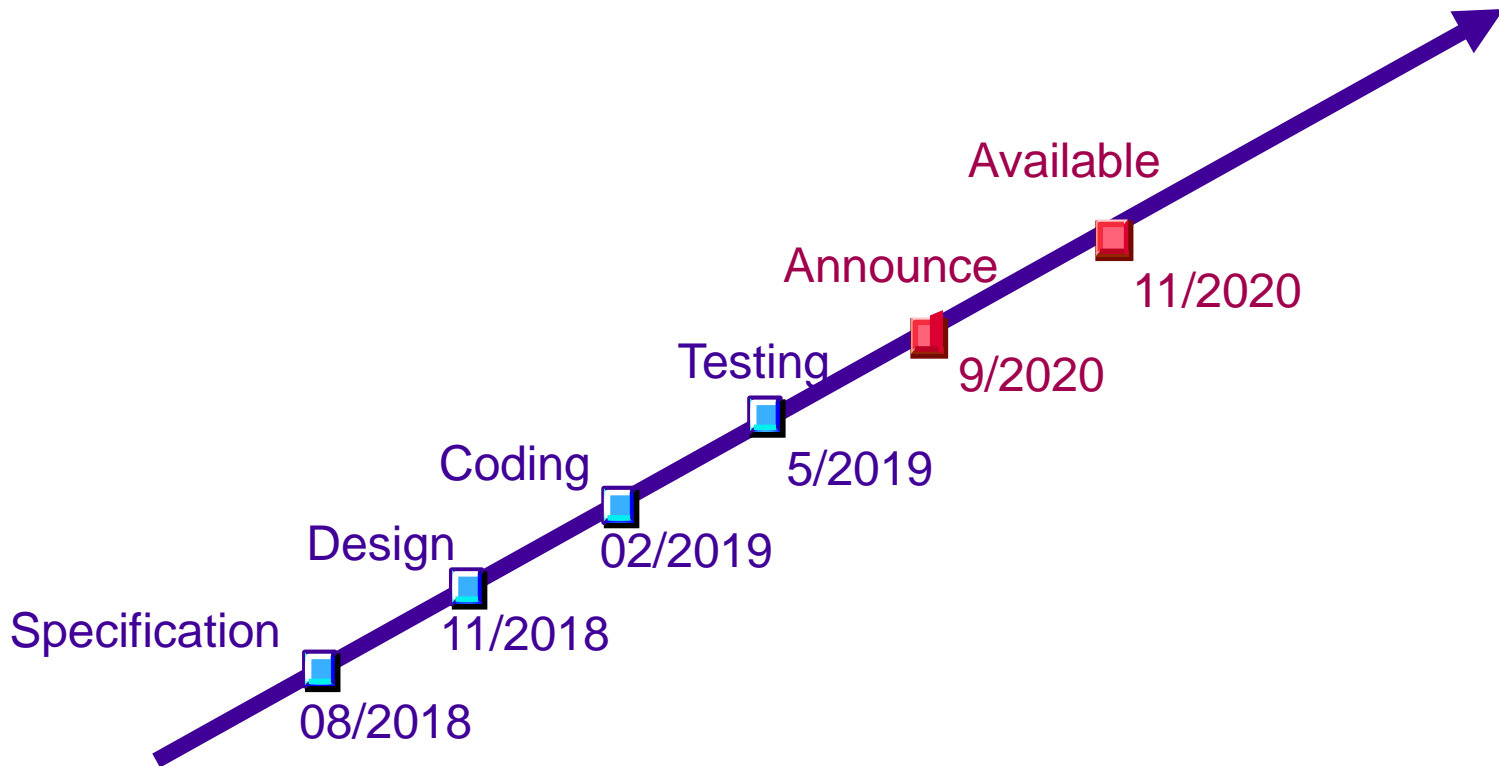
- Gantt chart

Activity	Predecessor	Time estimates			Expected time (T_E)
		Opt. (O)	Normal (M)	Pess. (P)	
<i>a</i>	—	2	4	6	4.00
<i>b</i>	—	3	5	9	5.33
<i>c</i>	<i>a</i>	4	5	7	5.17
<i>d</i>	<i>a</i>	4	6	10	6.33
<i>e</i>	<i>b, c</i>	4	5	7	5.17
<i>f</i>	<i>d</i>	3	4	8	4.50
<i>g</i>	<i>e</i>	3	5	8	5.17



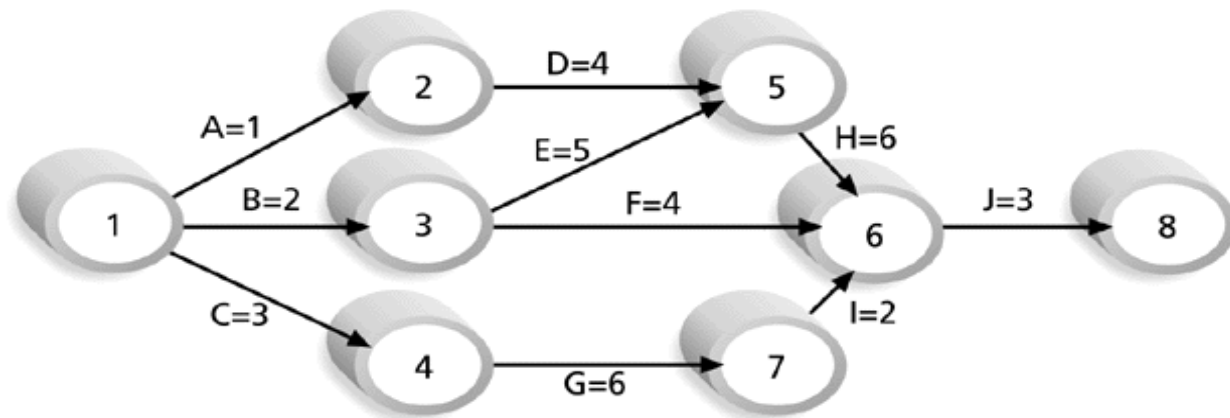
Project Progress Management

Milestone



Project Progress Management

Critical Path



Note: Assume all durations are in days.

Path 1: A-D-H-J Length = $1+4+6+3 = 14$ days

Path 2: B-E-H-J Length = $2+5+6+3 = 16$ days

Path 3: B-F-J Length = $2+4+3 = 9$ days

Path 4: C-G-I-J Length = $3+6+2+3 = 14$ days

Since the critical path is the longest path through the network diagram, Path 2, B-E-H-J, is the critical path for Project X.

Project Progress Management

Resource Consumption

- Human
- Time
- Equipment
- Investment
- ...



Risks



Project Progress Management

One More Important Risks:

- Demand Changing

Demand Confirm is very very very important!

Revision Control

- CVS
- SVN
- Git
- VSS



Project Progress Management

Report:

- 项目开发计划
- 开发进度月报
- 项目开发总结报告





上海外国语大学
SHANGHAI INTERNATIONAL STUDIES UNIVERSITY

a group working method

Team Management

Team Management

Team Member

- Customer
- Your Group
- Vender
- Provider

Project Manager
System Analyzer
System Designer
Database Administrator
Programmer
Testing Engineer
Sales
...



Team Management

How to run a team ?

- Culture
- Rules
- Good administrative director
- Motivation
- Promotion
- Backup important roles
- Good management of documents and codes



案例题

你是一个项目的项目经理，项目已经接近尾声，项目组一些成员已经分配到其他的项目组中，其中的一个设计人员由于还有一些事情，所以还留在项目继续工作，但是，这个设计人员突然提出来希望离开这个项目，因为另外一个项目需要他做项目经理的工作，他不想失去这个机会，这时作为项目经理，你应该如何做：



- A) 找另外一个合适的人完成剩下的工作，同意他到新的项目中，但是要求做好交接工作，同时要求他参加必要的会议
- B) 要求他不要离开这个项目，因为他是最好的人选
- C) 不管怎样，他必须完成项目的收尾工作
- D) 同意他接手新的项目，但是要求他周末或者晚上的时候负责原来项目的收尾工作



Team Management

Ways to Influence that Help and Hurt Projects

- Projects are more likely to succeed when project managers influence with
 - expertise
 - work challenge
- Projects are more likely to fail when project managers rely too heavily on
 - authority
 - money
 - penalty



Suggestions for Improving Project Communications

- Manage conflicts effectively
- Develop better communication skills
- Run effective meetings
- Use templates for project communications

Email is always the best!



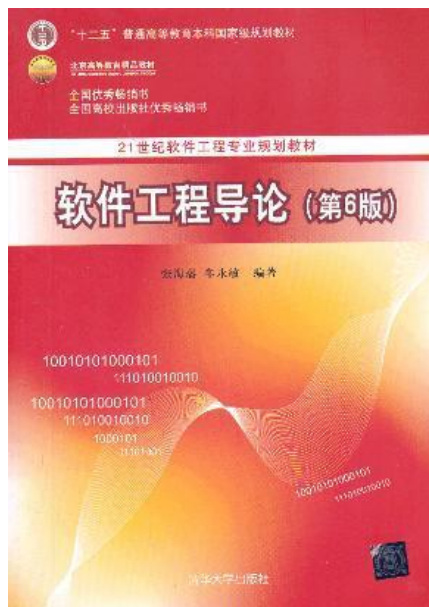


上海外国语大学
SHANGHAI INTERNATIONAL STUDIES UNIVERSITY

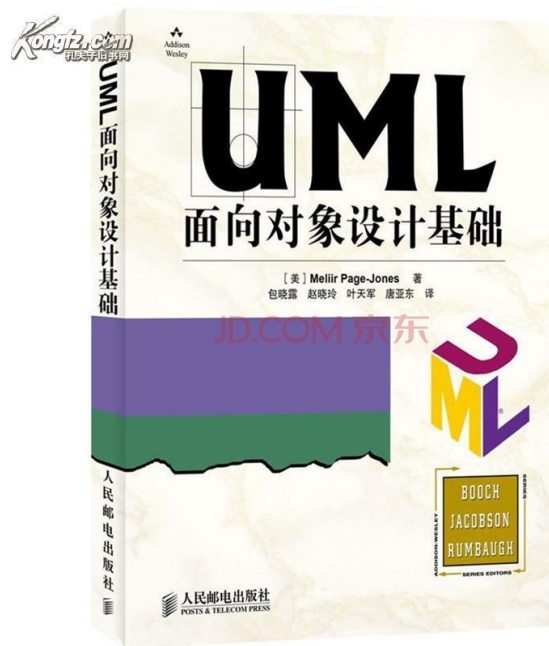
Reference

References

软件工程导论（第6版）



UML面向对象设计基础



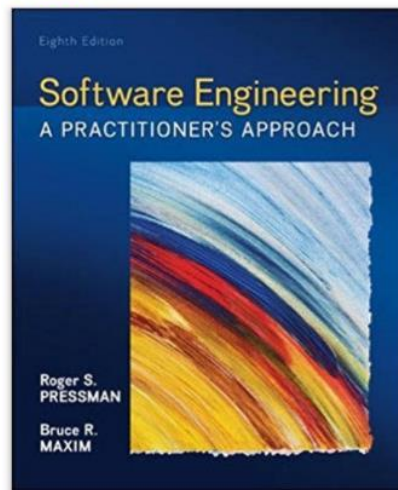
References

Books › Computers & Technology › Programming

Software Engineering: A Practitioner's Approach 8th Edition

by [Roger S. Pressman](#) (Author), [Bruce Maxim](#) (Author)

★★★★☆ 23 customer reviews



Hardcover

\$30.38 - \$100.88

Paperback

\$43.48

Other Sellers

See all 5 versions

☐ Rent

☒ **Buy new**

In Stock.

Sold by [TEXTBookAMAZING](#) and [Fulfilled by Amazon](#).

This item ships to **China**. [Learn more](#)

ISBN-13: 978-0078022128



上海外国语大学
SHANGHAI INTERNATIONAL STUDIES UNIVERSITY



上海外国语大学
SHANGHAI INTERNATIONAL STUDIES UNIVERSITY

Home Work

Please set up a group with a size of 4-6 people, and start a project in the following fields:

1. Film Box Office Prediction 电影票房预测
2. Matching Film Stars and Product Advertisements
影视明星与广告产品的适配度
3. Social Media and Public Opinion Mining
社交媒体与舆情分析
4. Prediction of the Influence about the Articles from WeChat Public Accounts
微信公众号的影响力预测
5. Others you like. 其他





The End of Lecture 4

Thank You

<http://www.wangting.ac.cn>

