

SOFT620020.01  
Advanced Software  
Engineering

Bihuan Chen, Pre-Tenure Assoc. Prof.

[bhchen@fudan.edu.cn](mailto:bhchen@fudan.edu.cn)

<https://chenbihuan.github.io>

# Course Structure

- In-Class Teaching and **Discussion** (Every Monday 11-13)
- No After-Class Assignment
- Discussion (25%): a range of topics
- Three Presentations (75%): 18 + 2 minutes
  - Paper presentation
  - **Tool demonstration**
  - **Any new ideas**
  - ...

# Course Communication

- All materials are available at the course website  
<https://chenbihuan.github.io/course/ase/>
- Email: [bhchen@fudan.edu.cn](mailto:bhchen@fudan.edu.cn)
- Office: Room 403, Software Building, Zhangjiang Campus  
(make an appointment via email first)
- WeChat group for course announcement



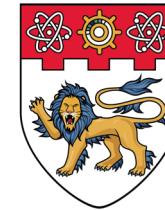
# Course Outline

Date	Topic	Date	Topic
Sep. 09	Introduction	Nov. 04	Mobile Testing
Sep. 16	Testing Overview	Nov. 11	Delta Debugging
Sep. 23	Guided Random Testing	Nov. 18	Presentation 2
Sep. 30	Search-Based Testing	Nov. 25	Bug Localization
Oct. 12	Performance Analysis	Dec. 02	Automatic Repair
Oct. 14	Presentation 1	Dec. 09	Symbolic Execution
Oct. 21	Security Testing	Dec. 16	Big Code Analysis
Oct. 28	Compiler Testing	Dec. 23	Presentation 3

**For each topic, I will introduce two or three approaches proposed in the literature, and organize multiple discussions.**

# About Me

- BSc and PhD from Fudan
- Postdoc from NTU
- Pre-Tenure Assoc. Prof. at Fudan



NANYANG  
TECHNOLOGICAL  
UNIVERSITY  
SINGAPORE

- 1-year teaching experience 🤓

- Software Engineering
- Program Analysis
- Big Code Analysis
- Software Security



# About You

## Let Me Introduce Myself

- \* Hi, my name is ...
- \* I come from ... (country)
- \* I live in ... (city)
- \* I'm ... (age)
- \* There are ... people in my family. They are ...
- \* I'm a student at ...
- \* My major is ... (majors)
- \* My favorite subject is ... (subjects)
- \* My hobbies are ...
- \* In my free time, I also enjoy ... (sports)
- \* I (don't) like/ dislike/ hate ...
- \* My favorite food/ drink is ...
- \* I like ... (movies)
- \* My favorite singer/ band is ...
- \* I sometimes go to ... (places), I like it because ...
- \* I study English because ...
- \* I've been learning English for/ since ...
- \* I would like to be a/ an ... (jobs) because ...



### Subjects

Physics                      Biology  
Math                          English  
Music                        Geography  
Science                     Chemistry  
Literature                   ...

### Movies

Action                       Drama  
Comedy                     Thriller  
Romance                    Cartoons  
Horror                      ...

### Because

- I have had a beautiful memory here.
- This is one of the most beautiful places I have visited.
- It's very important and necessary.
- I love to improve my English skills.
- I want to learn more about this language.
- I love this job.
- ...



### Majors

Accounting                Marketing  
Arts                        Journalism  
Economics                 Sociology  
History                    Philosophy  
Humanities                ...

### Sports

Volleyball                Cycling  
Badminton                Running  
Tennis                     Fishing  
Yoga                        ...

### Hobbies

- Reading books/ newspapers
- Playing computer games/ badminton/ tennis
- Surfing the Internet
- Collecting stamps/ coins
- Listening to music
- Chatting with best friends
- Go shopping/ camping
- ...

# A Quick Journey through Software Engineering

# A Warm-Up Questionnaire



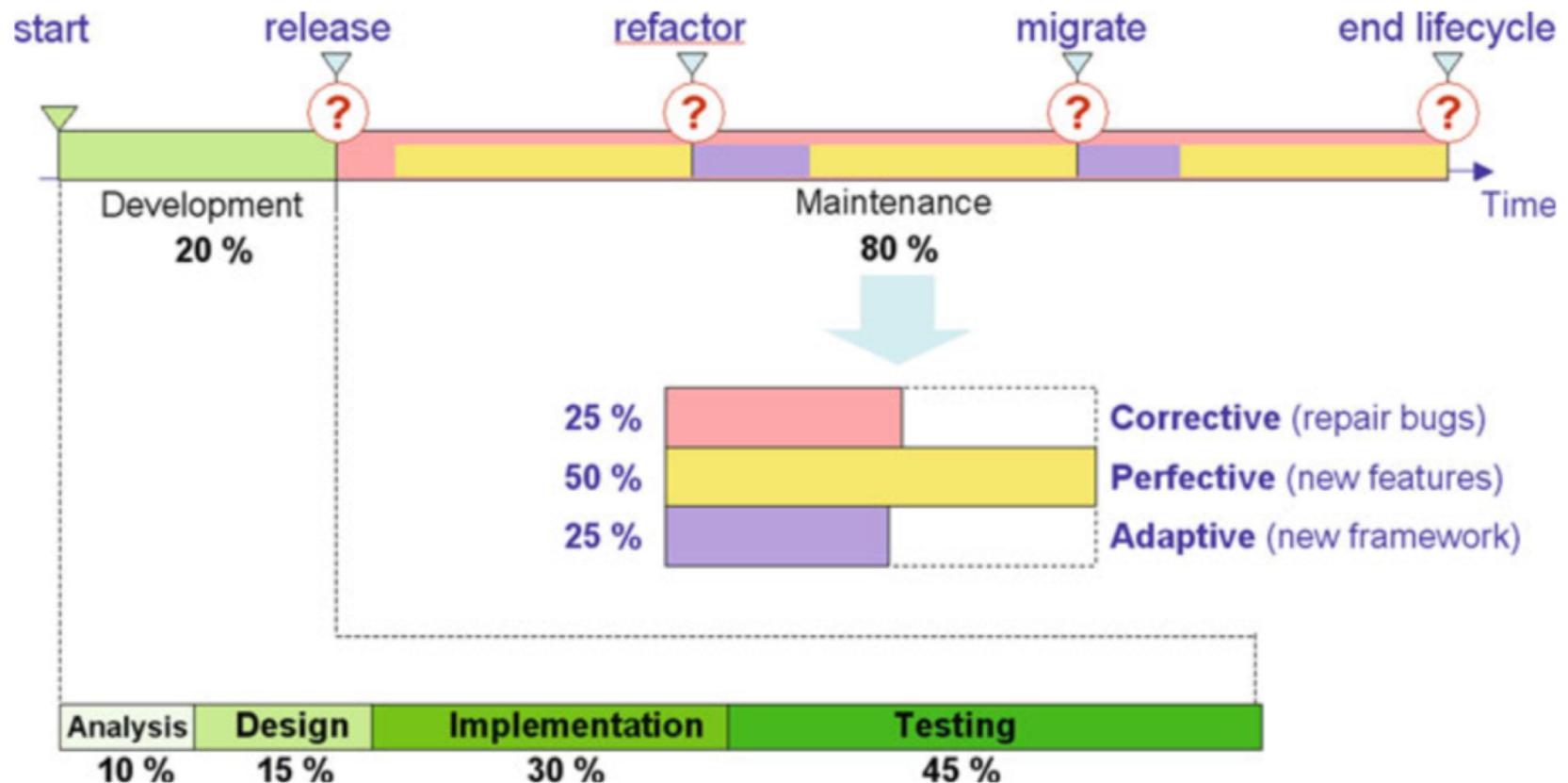
- Did you take the course about software engineering?
- Did you have programming experience of  $\geq$  two years?
- Did you participate in the development of a project?
- Did you write and manage tests for your projects?
- Did you have a debugging nightmare for your projects?
- Did you use any automatic tools to help you test or debug?
- Did you submit bug reports for open-source projects?
- Can you prove that your program is correct?

# Discussion 1 – Software and SE



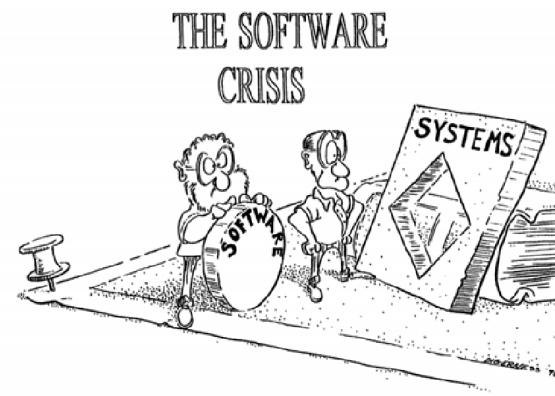
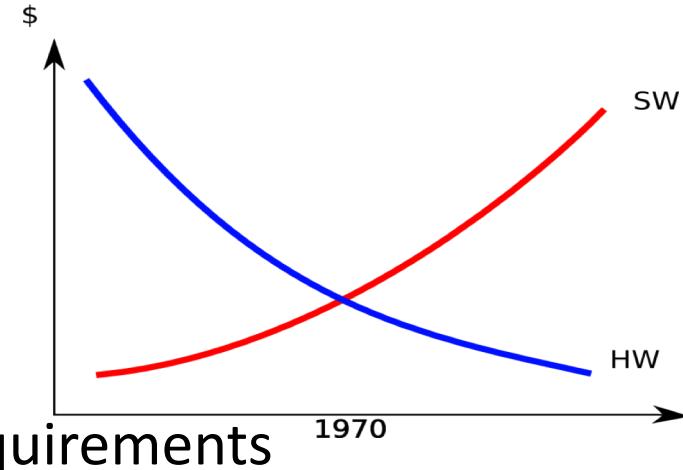
- What is software?
- What are the attributes of good software?
- What is software engineering?
- Software: computer **programs** and associated **documents**
- Good software: deliver the required **functionalities** and **performance** to users, be **maintainable**, **dependable** and **usable**
- Software engineering: an engineering **discipline** that is concerned with **all** aspects of software production

# Discussion 2 – Challenges and Costs



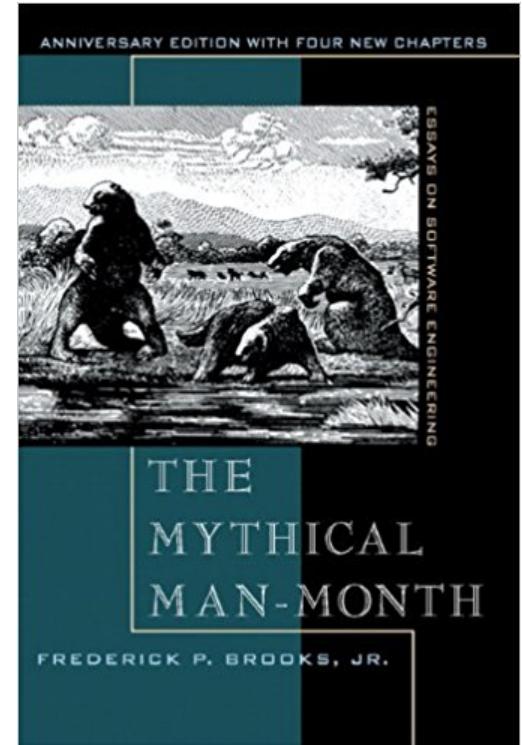
# Software Crisis

- Projects ran over-budget
- Projects ran over-time
- Software was very inefficient
- Software was of low quality
- Software often did not meet requirements
- Projects were unmanageable
- Code was difficult to maintain
- Software was never delivered
- ...



# IBM's System/360 Operating System

- Development period: 1963 – 1966
- Manpower: 5,000 man-year
- Code: one million lines of code
- Costs: hundreds of millions of dollars
- Delivery was delayed
- Cost was several orders of magnitude higher
- More memory was needed as planned
- The first release did not work well
- The system worked well after several releases
- Each release fixed thousands of bugs



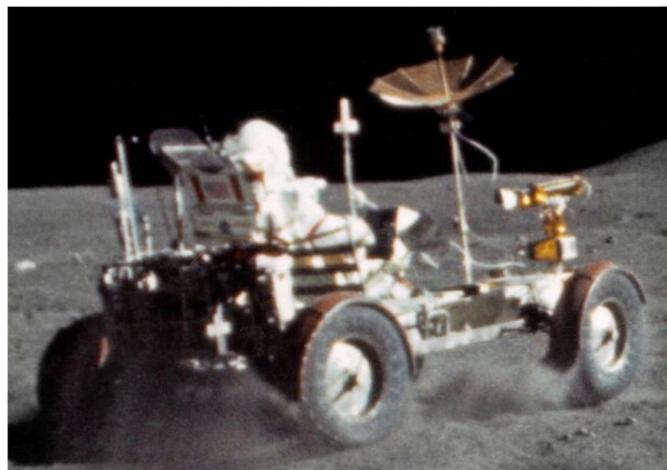
**Frederick Brooks**  
**"Adding manpower to a late software project makes it later"**

# The Idea of Software Engineering

<https://www.youtube.com/watch?v=ZbVOFOUk5IU>

- Margaret Hamilton, the lead Apollo flight software designer

*"When I first came up with the term, no one had heard of it before, at least in our world. It was an ongoing joke for a long time. They liked to kid me about my radical ideas."*



The Apollo 11 on the moon

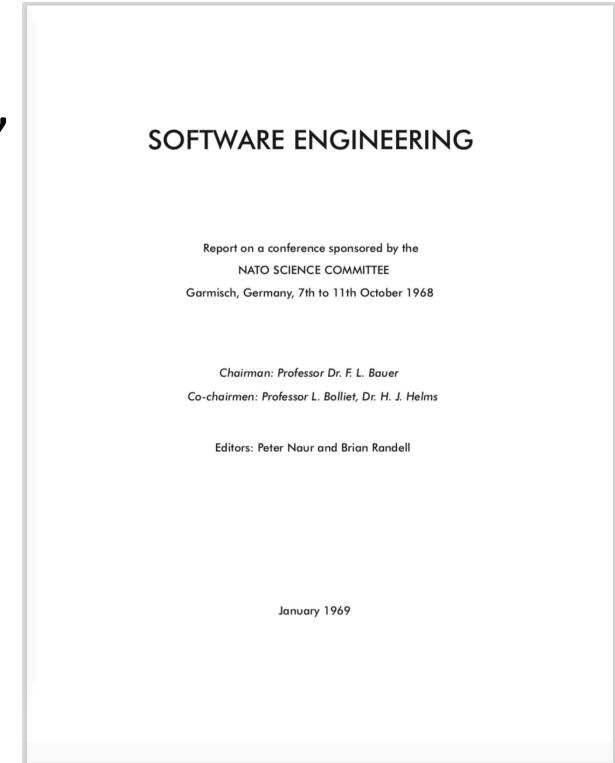


Standing next to listings of the software

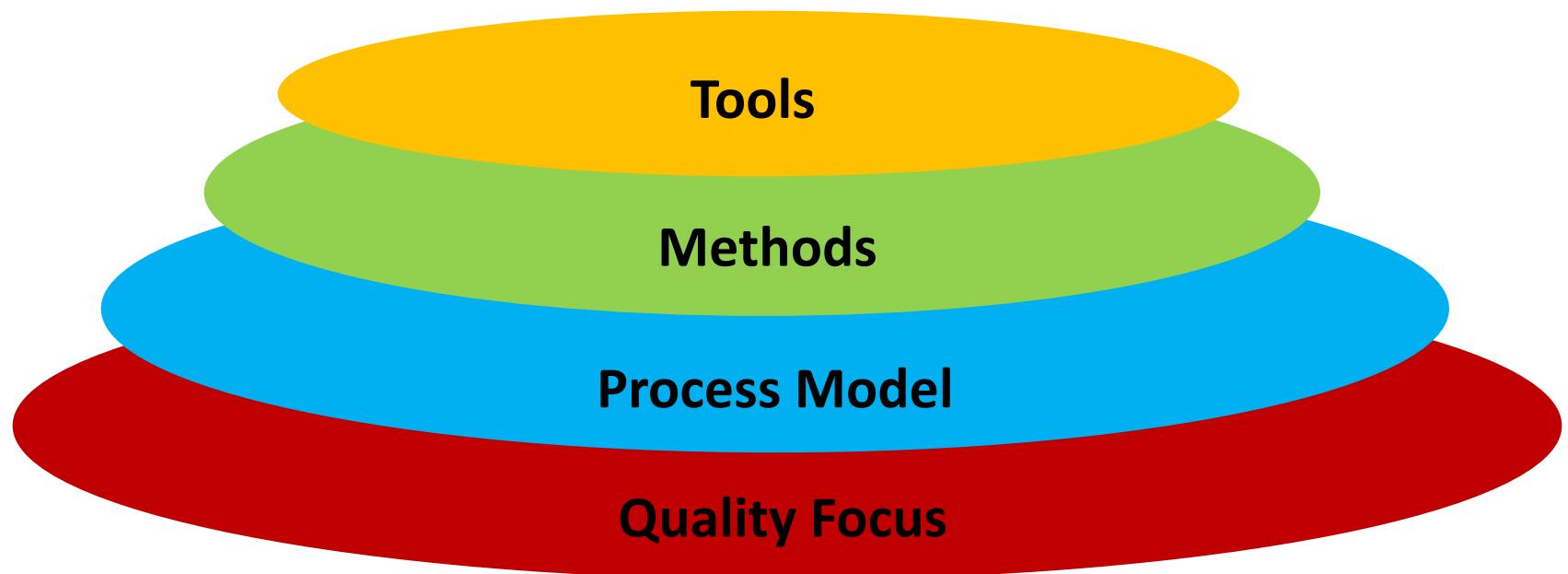
Her concepts (e.g., asynchronous software, priority scheduling and software reuse) became the foundation for ultra-reliable software design

# The Birth of Software Engineering

- **Fritz Bauer**, at NATO in 1968
- Suggest the term “Software Engineering” as a way to conceive both the problem and the solution of software crisis
- *“The establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines”*



# SE: A Layered Technology



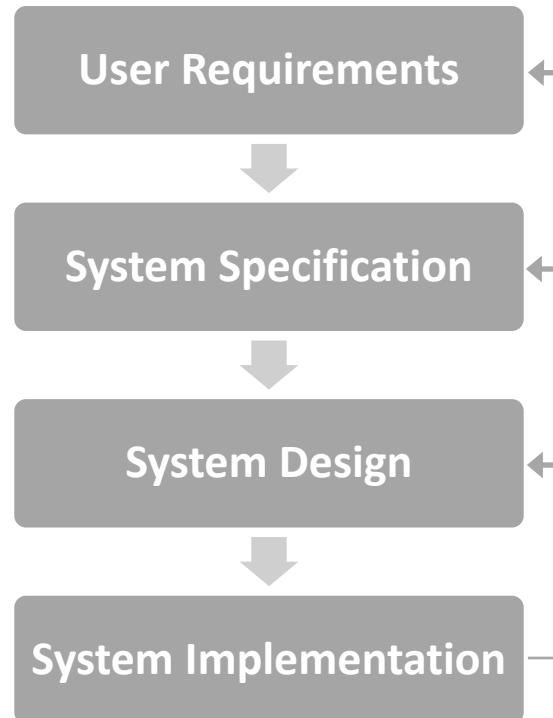
# Activities in Software Engineering

**Do we get the right requirements?**

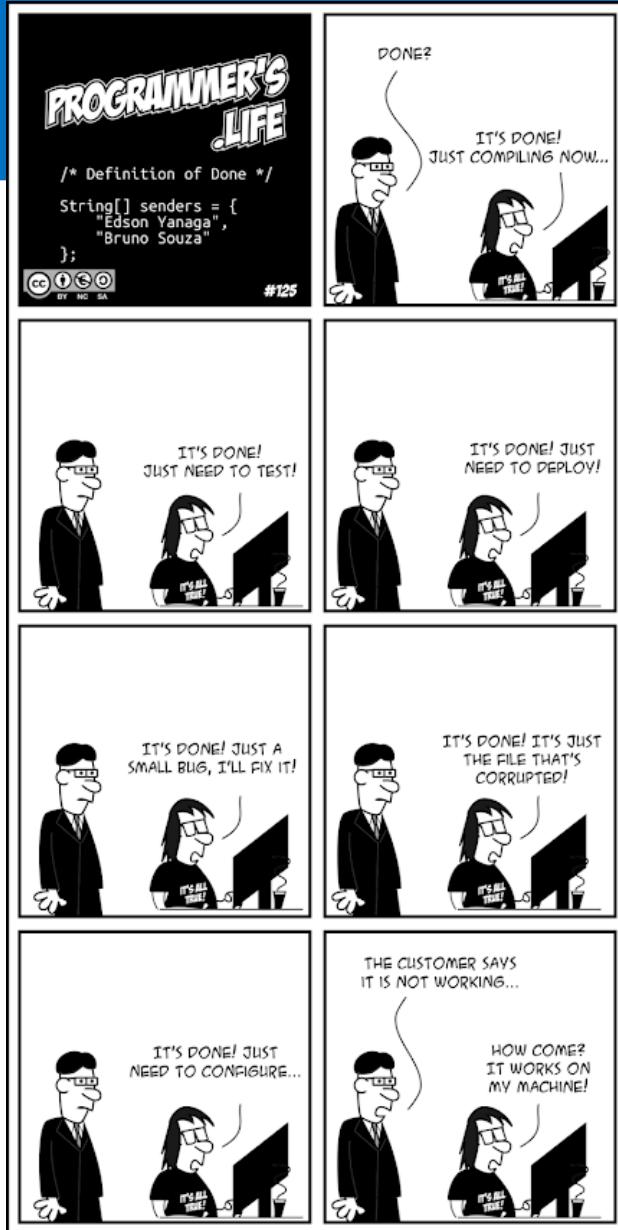
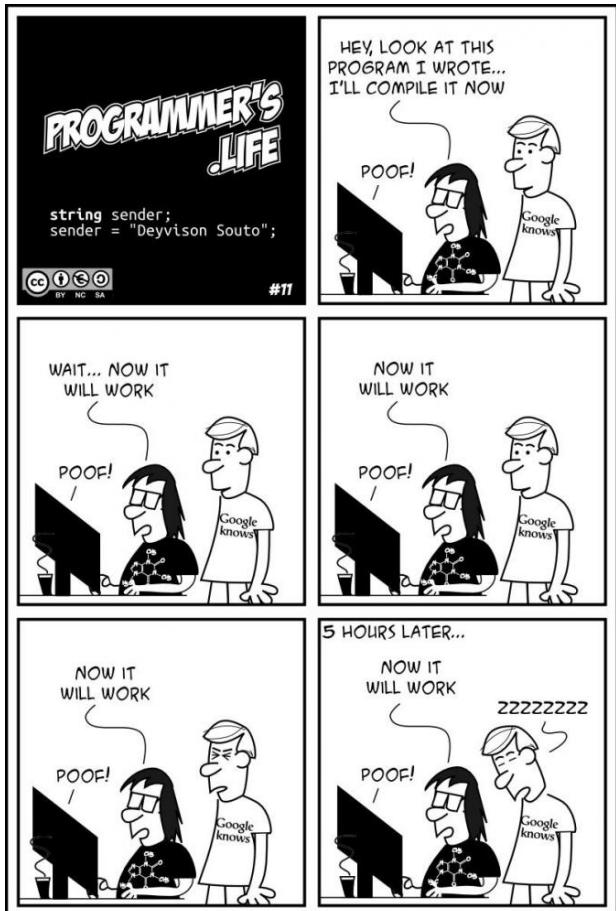
**Is the specification equivalent to requirements?**

**Does the design satisfy the specification?**

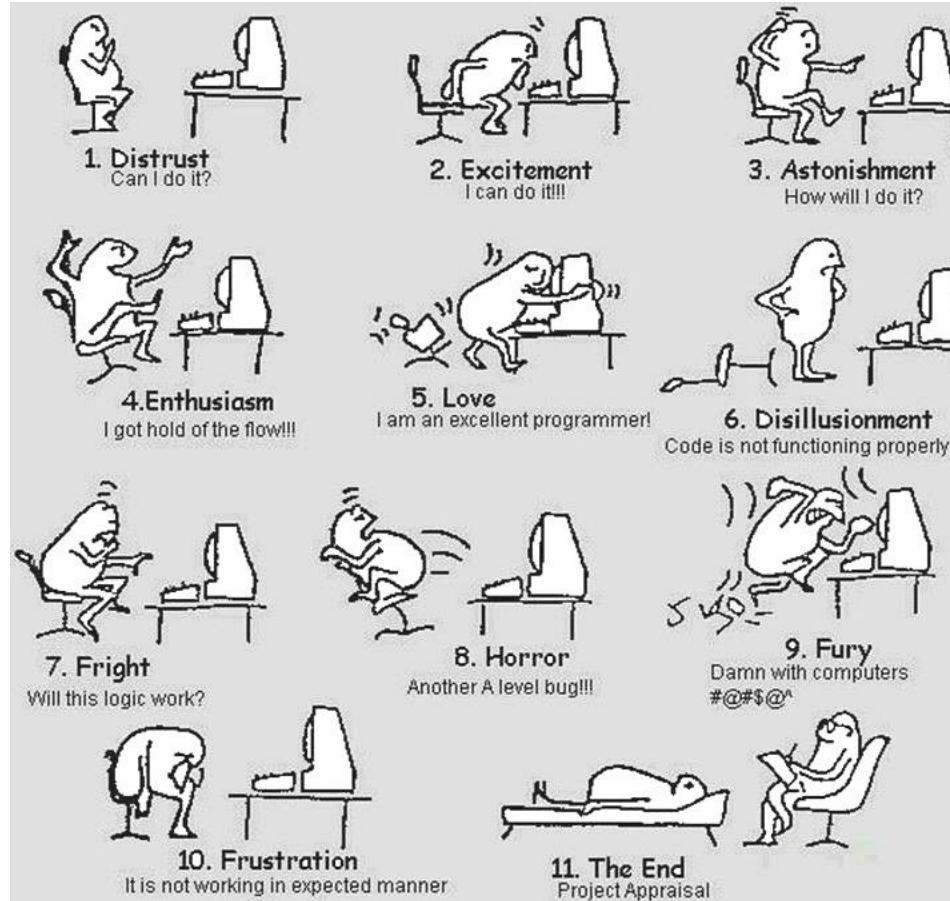
**Is the design correctly implemented?**



# A Programmer's Life



# A Programmer's Life (cont.)



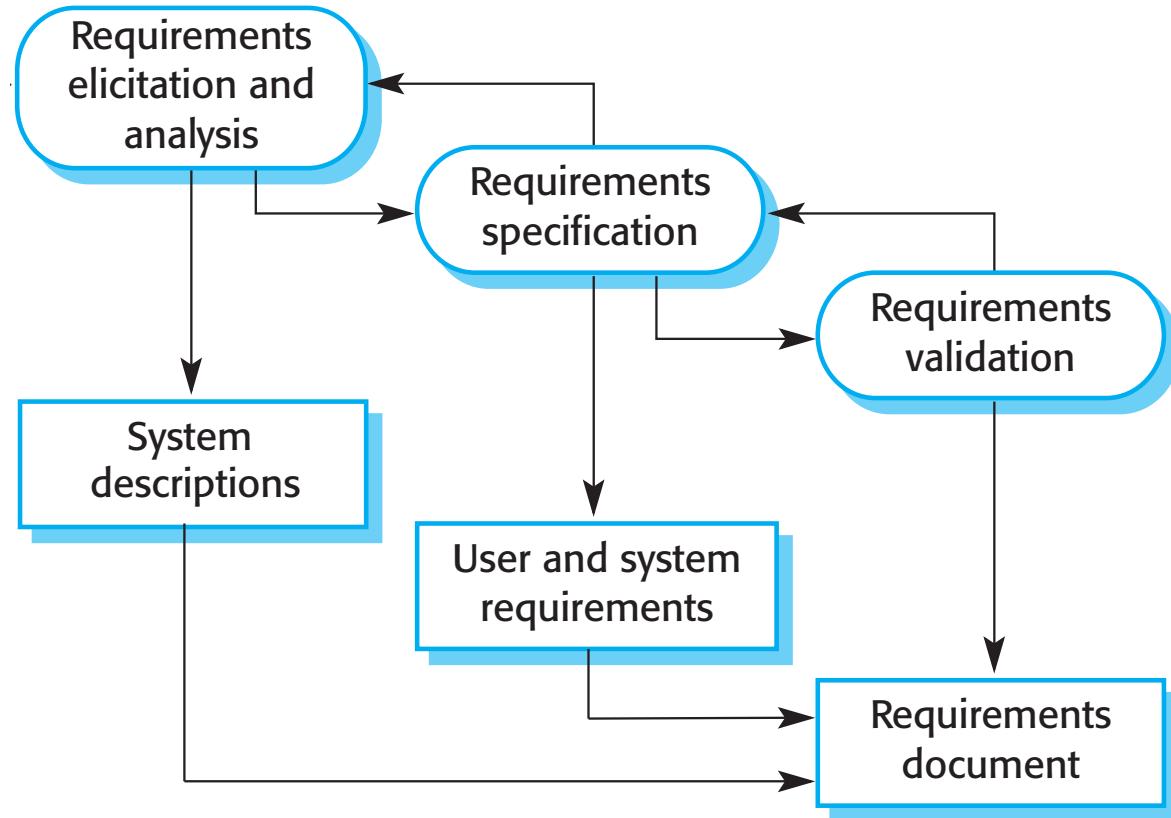
# Discussion 3 – Your Experience?



- What bad programming experiences did you encounter? Why?

# Requirements and Specification

- Establish the required services and the constraints on the system's operation and development

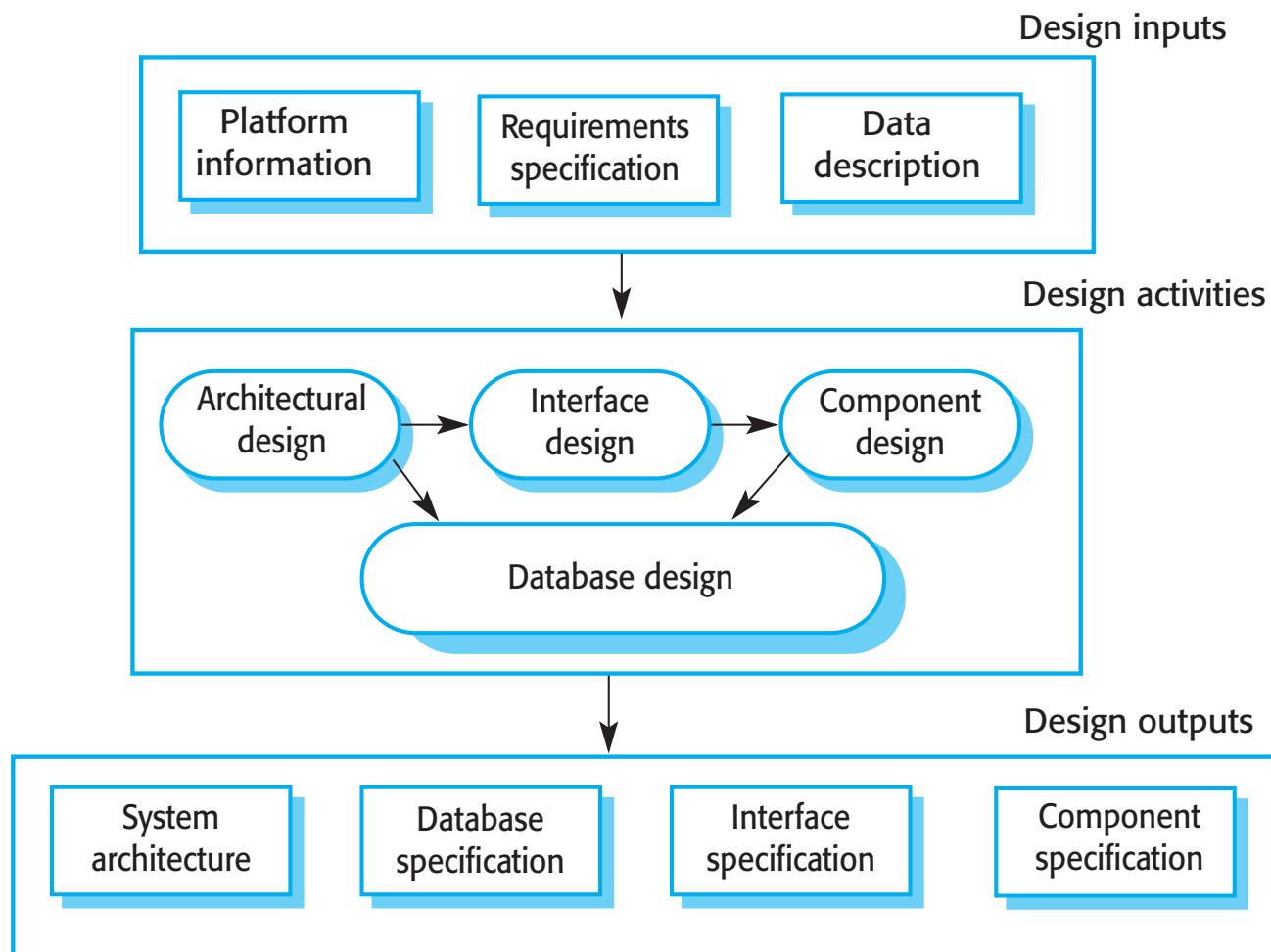


# Requirements and Specification (cont.)

- Example: sort an array of integers within certain time
  - Red: functional requirement
  - Green: quality requirement
- Formal specification languages
  - The Z language, VDM, the B language, etc.
  - CSP, CCS, etc.

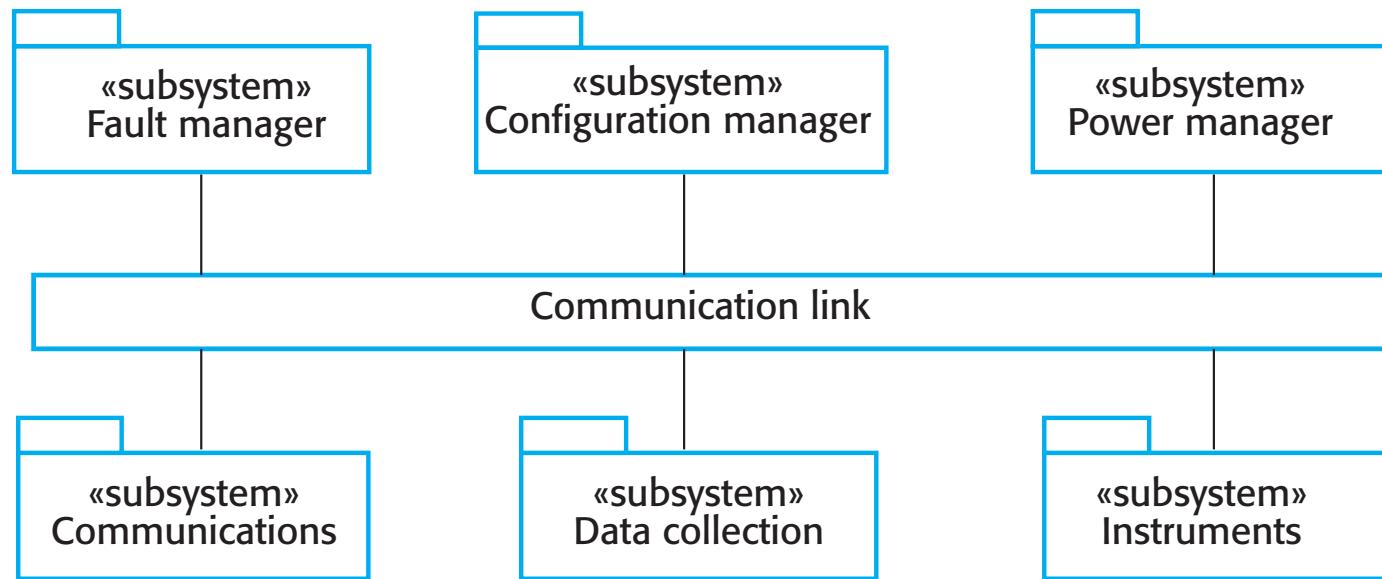
# System Design

- **Design a software structure that realizes the specification**



# System Design (cont.)

- Example: high-level architecture of a weather station system

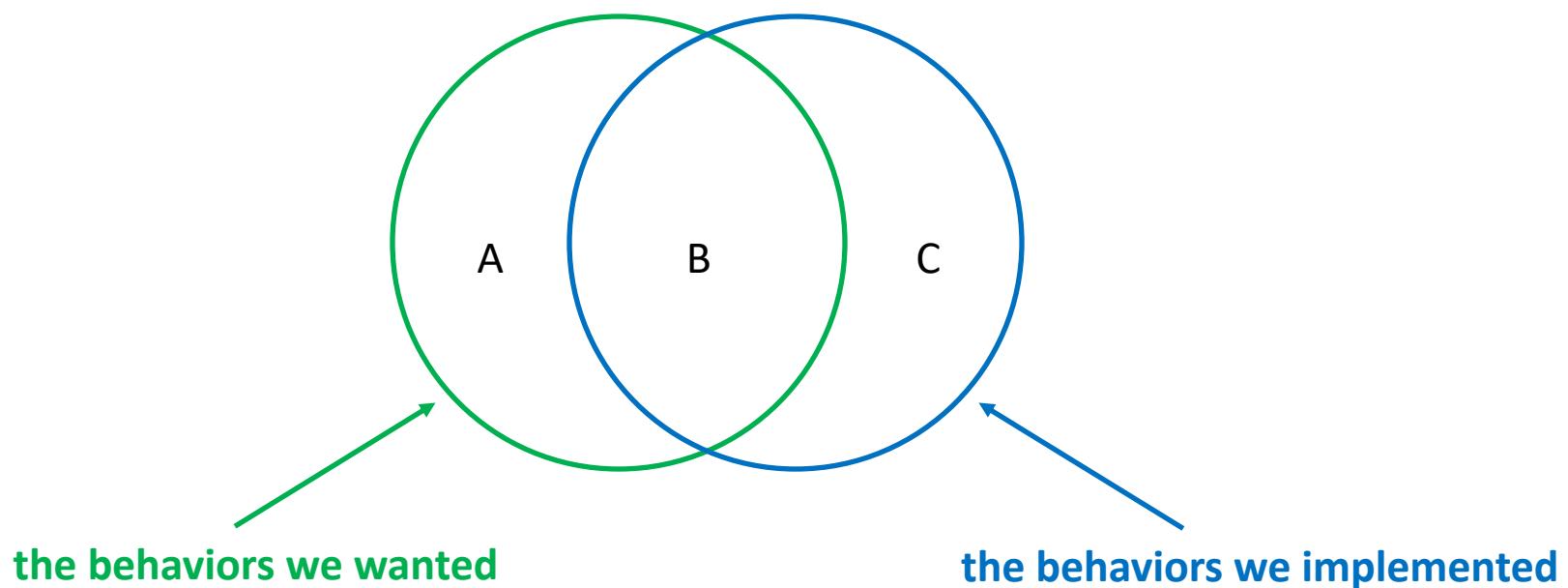


# System Implementation

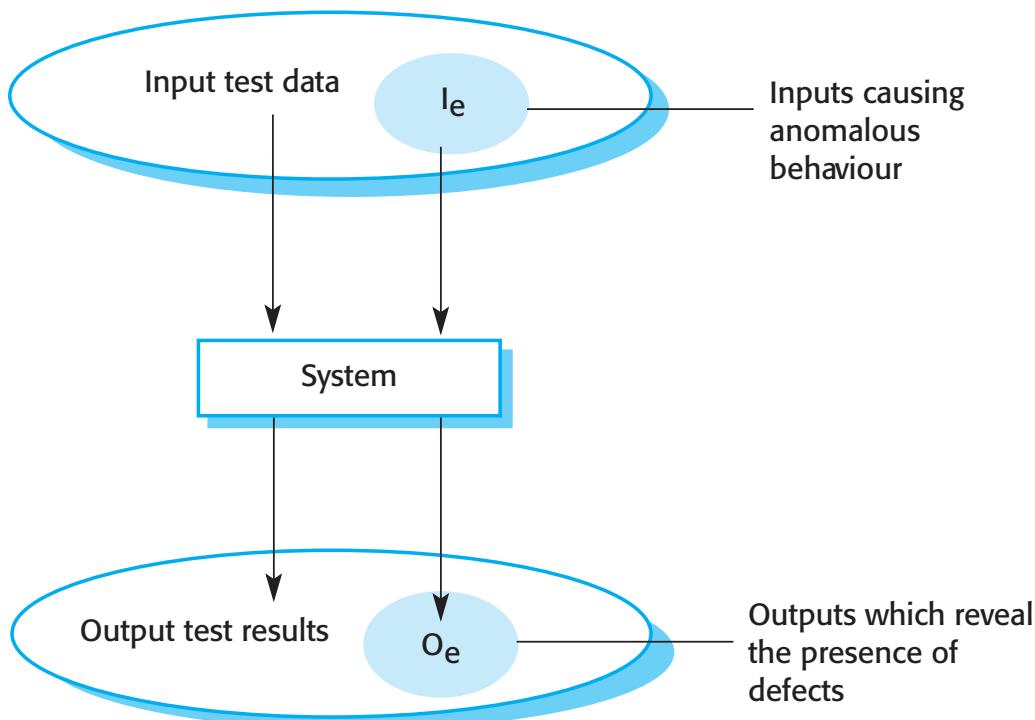
- Translate the software structure into an executable program
- The focus here is not on programming but on other implementation issues
  - Reuse
  - Configuration management
  - Host-target development
  - ...
- The activities of implementation and testing are often closely related and may be interleaved

# Testing

- Testing is the most commonly used verification and validation activity



# Testing (cont.)

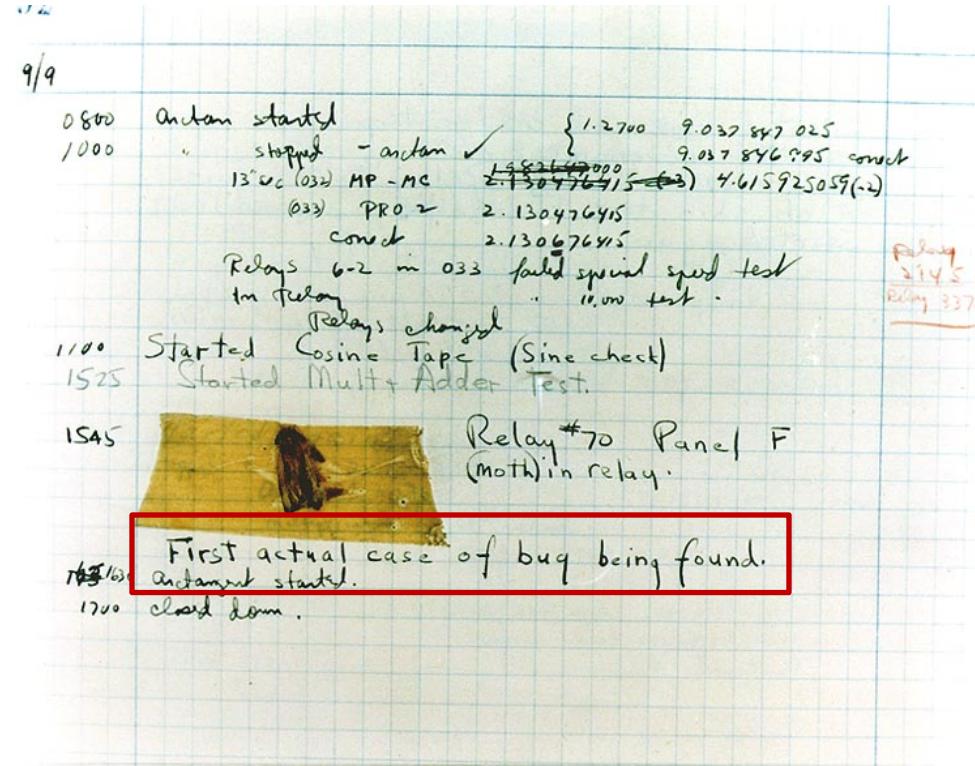


# Discussion 4 – Your Experience?



- Did you follow the previous process model? Why?

# The Origins of Bug



Grace M. Hopper's associates discovered a **moth** that was stuck in a relay; and the moth impeded the operation of the relay

# Discussion 5 – Your Experience?



- unexpected program behaviours manifested as crashes, errors, loss of data, denial of service, poor performance, high energy consumption, etc.

- What kinds of bugs did you encounter?

The **red** and **green** library contains a class with same name but different function, and I needed the class in the **red** library



Computer A

Computer B

Root cause: use **File.listFiles** to load the two libraries, which has no guarantee of any specific order.

# Discussion 6 – Debugging

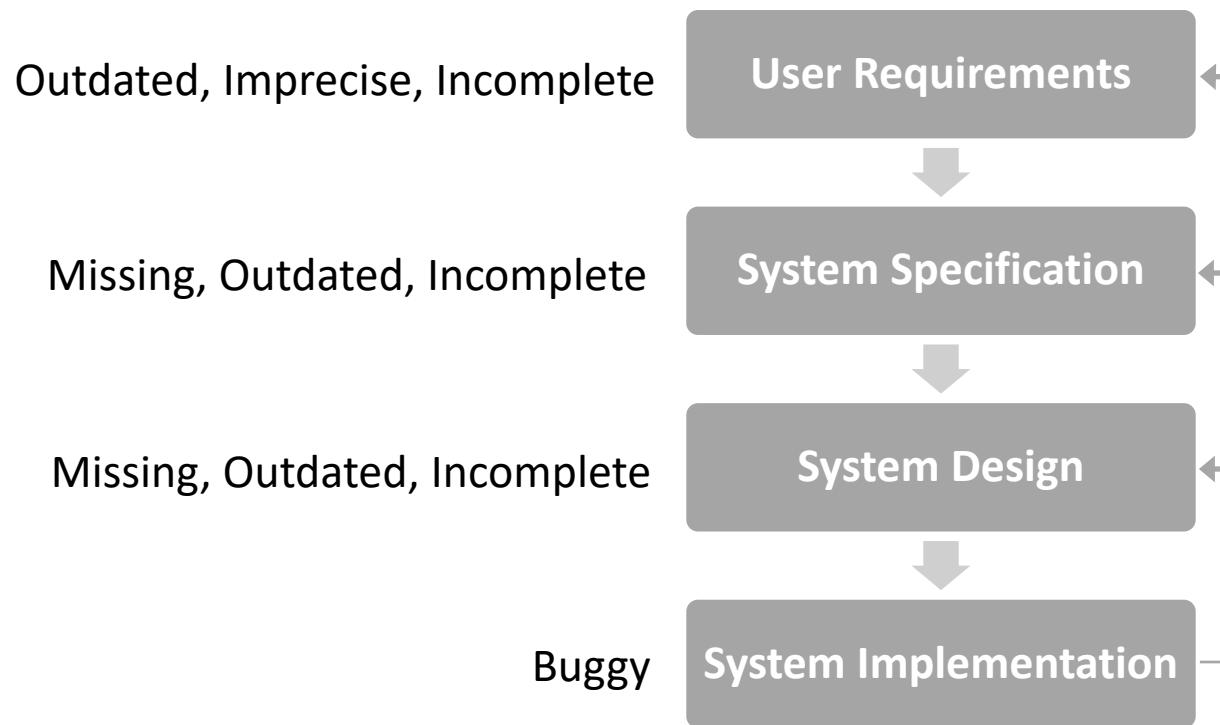


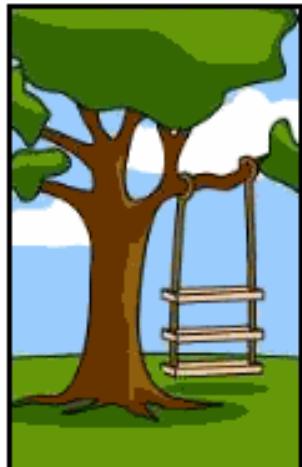
- How do you debug the following buggy sort?

```
class BuggySort {  
    public static void sort (int[] input) {  
        int n = input.length;  
        for (int i = 1; i < n - 1; i++) {  
            for (int j = i; j < n - 1; j++) {  
                if (input[j] > input[j+1]) {  
                    int tmp = input[j];  
                    input[j] = input[j+1];  
                    input[j+1] = tmp;  
                }  
            }  
        }  
    }  
}
```

```
class CorrectSort {  
    public static void sort (int[] input) {  
        int n = input.length;  
        for (int i = 0; i < n - 1; i++) {  
            for (int j = 0; j < n - i - 1; j++) {  
                if (input[j] > input[j+1]) {  
                    int tmp = input[j];  
                    input[j] = input[j+1];  
                    input[j+1] = tmp;  
                }  
            }  
        }  
    }  
}
```

# Software Engineering





How the customer  
explained it



How the Project  
Leader understood it



How the Analyst  
designed it



How the Programmer  
wrote it



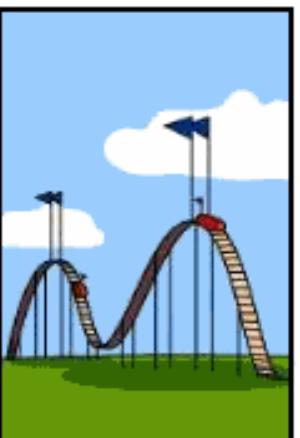
How the Business  
Consultant described it



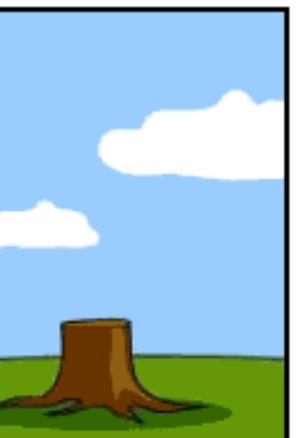
How the project  
was documented



What operations  
installed



How the customer  
was billed



How it was supported



What the customer  
really needed

# Revisit Course Plan

Date	Topic	Date	Topic
Sep. 09	Introduction	Nov. 04	Mobile Testing
Sep. 16	Testing Overview	Nov. 11	Delta Debugging
Sep. 23	Guided Random Testing	Nov. 18	Presentation 2
Sep. 29	Search-Based Testing	Nov. 25	Bug Localization
Oct. 07	Performance Analysis	Dec. 02	Automatic Repair
Oct. 14	Presentation 1	Dec. 09	Symbolic Execution
Oct. 21	Security Testing	Dec. 16	Big Code Analysis
Oct. 28	Compiler Testing	Dec. 23	Presentation 3

**For each topic, I will introduce two or three approaches proposed in the literature, and organize multiple discussions.**

# Q&A?

Bihuan Chen, Pre-Tenure Assoc. Prof.

[bhchen@fudan.edu.cn](mailto:bhchen@fudan.edu.cn)

<https://chenbihuan.github.io>

# Discussion 7 – AI Developers



Do you believe that AI can replace human developers?

