Software Process Models

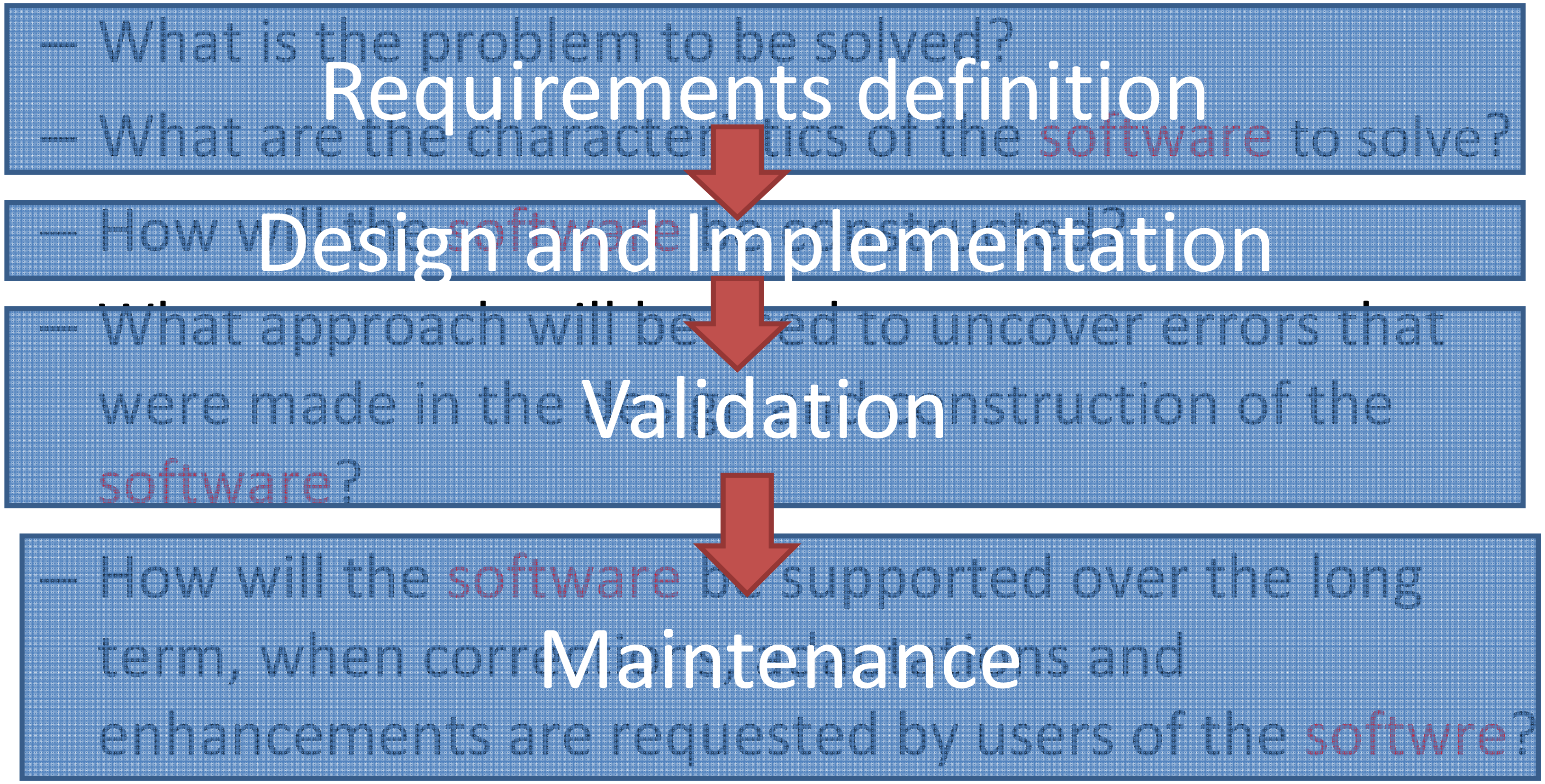
Xin Feng

Outline

- Software development lifecycle
- Process models

Questions to Answer in Software Engineering?

- Questions to answer in software engineering
  - What is the problem to be solved?
  - What are the characteristics of the software to solve?
  - How will the software be constructed?
  - What approach will be used to uncover errors that were made in the design and construction of the software?
  - How will the software be supported over the long term, when corrections, adaptations and enhancements are requested by users of the softwre?



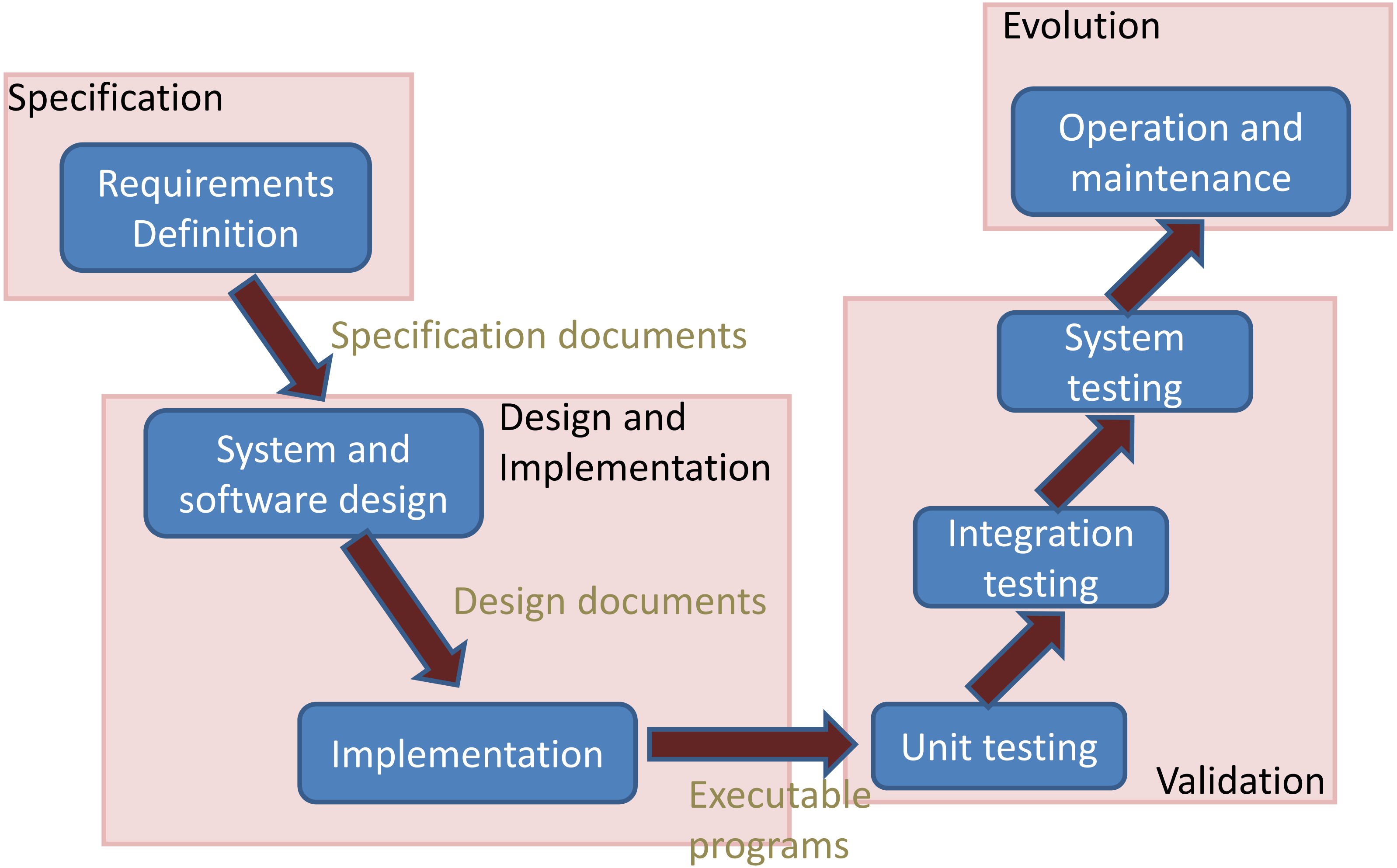
Requirements definition

Design and Implementation

Validation

Maintenance

Software Development Life Cycle



Fundamental Activities

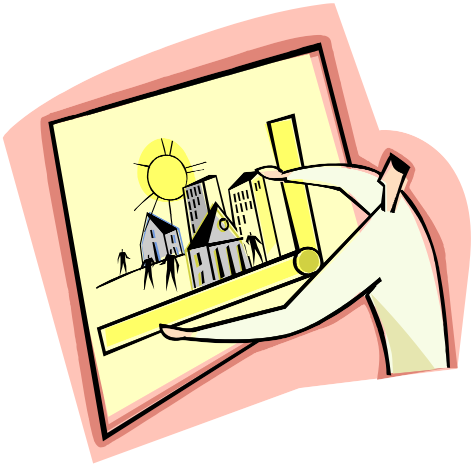
- Software requirements definition (specification 规格说明)
  - understand and define the services
  - Identify the constraints （约束 ）
- Software design and implementation
  - Convert specifications to executable systems
- Software validation
  - Check if a system conforms to its specification (verification （验证 ）) and meets the customer’ s expectation (validation （确认 ）)
- Software evolution
  - Evolve to meet the required changes from customers and bug corrections.

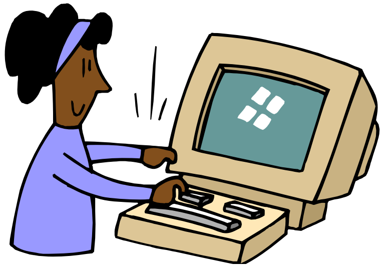
Software Requirements Specification

Customers
/Clients （客户）
/Users

developer
/engineer
/analyst

Design and Implementation



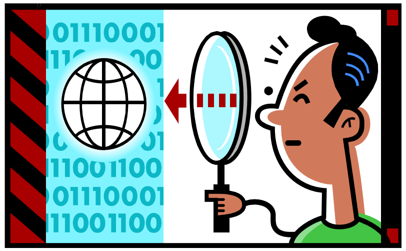


Design the architecture (结构)

&

write program

Software Validation (确认)

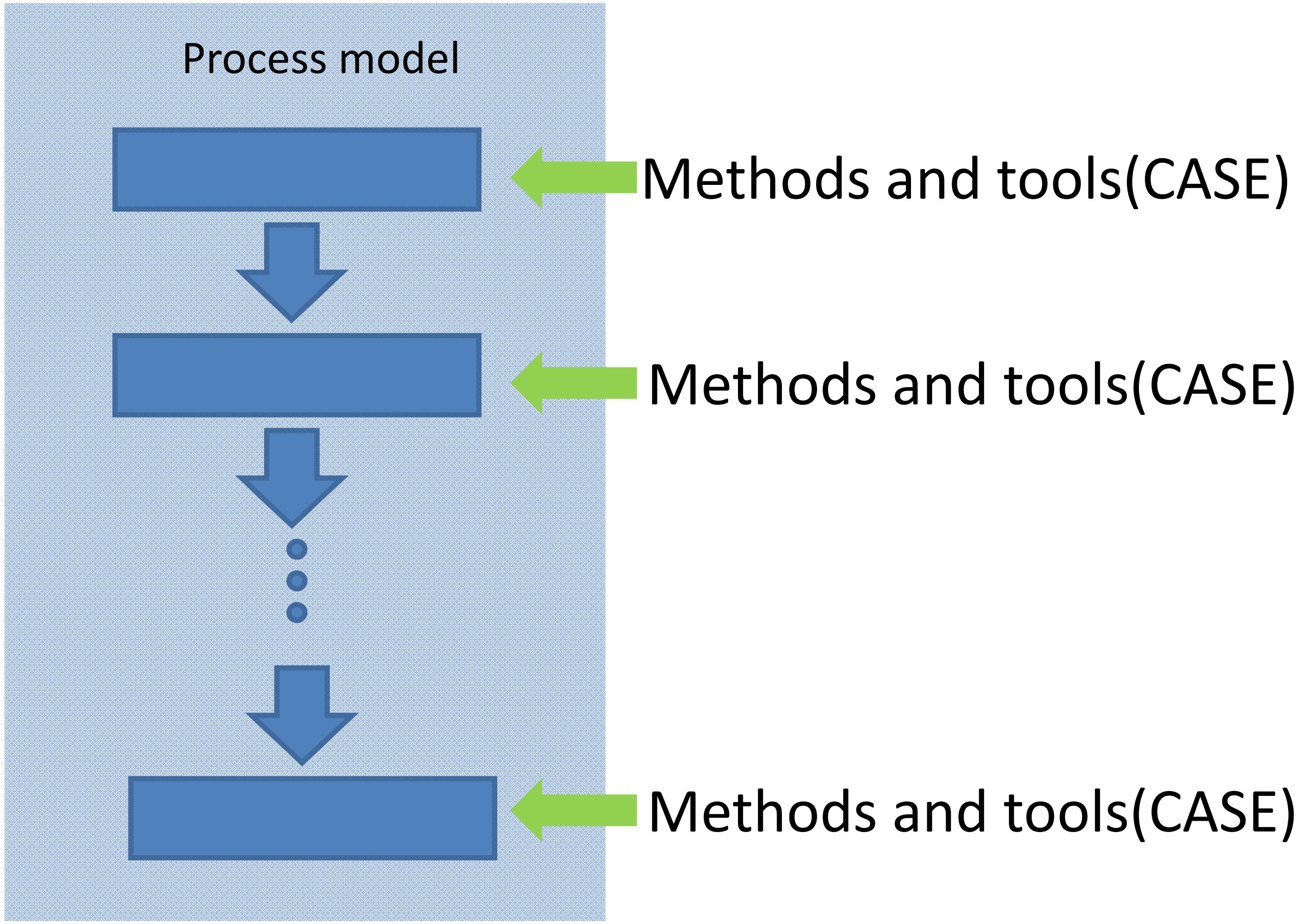


Is the system right???

Maintenance

1. More functions added
2. Bugs corrected
3. New versions delivered

Software Engineering



Software Process and Process Model

- A software process is
  - a set of activities whose goal is the development or evolution (演变) of software
- Software process model
  - a simplified representation of a software process from a specific perspective.

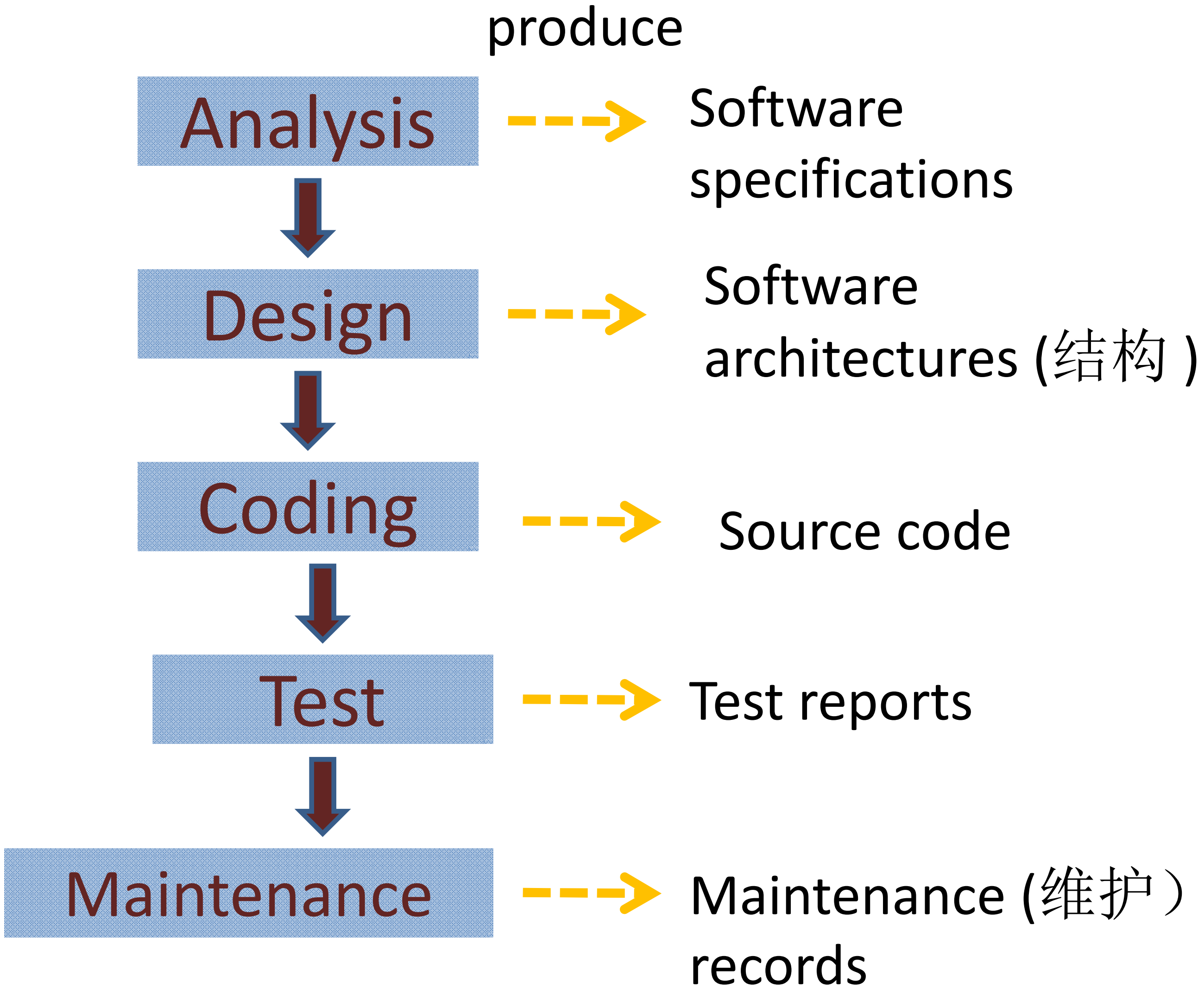
Process Models

- The waterfall (瀑布) model
- The incremental （增量）process model
- Prototype
- Spiral （蜘蛛） model

Process Models

- Formal method
  - VDM (Vienna Development Method)
  - The B method

The Waterfall Model



produce

Analysis → Software specifications

Design → Software architectures (结构)

Coding → Source code

Test → Test reports

Maintenance → Maintenance (维护） records

An Example

Requirements initial description:

Develop a software program that can calculate arithmetic expressions and display the result in a micro second.

The Waterfall Model

- Analysis
  - Functions
  - Behaviors
  - Interfaces
  - Performance （性能）
  - Specification forms
    - Tables
    - Diagrams （图表 ）
    - Natural languages

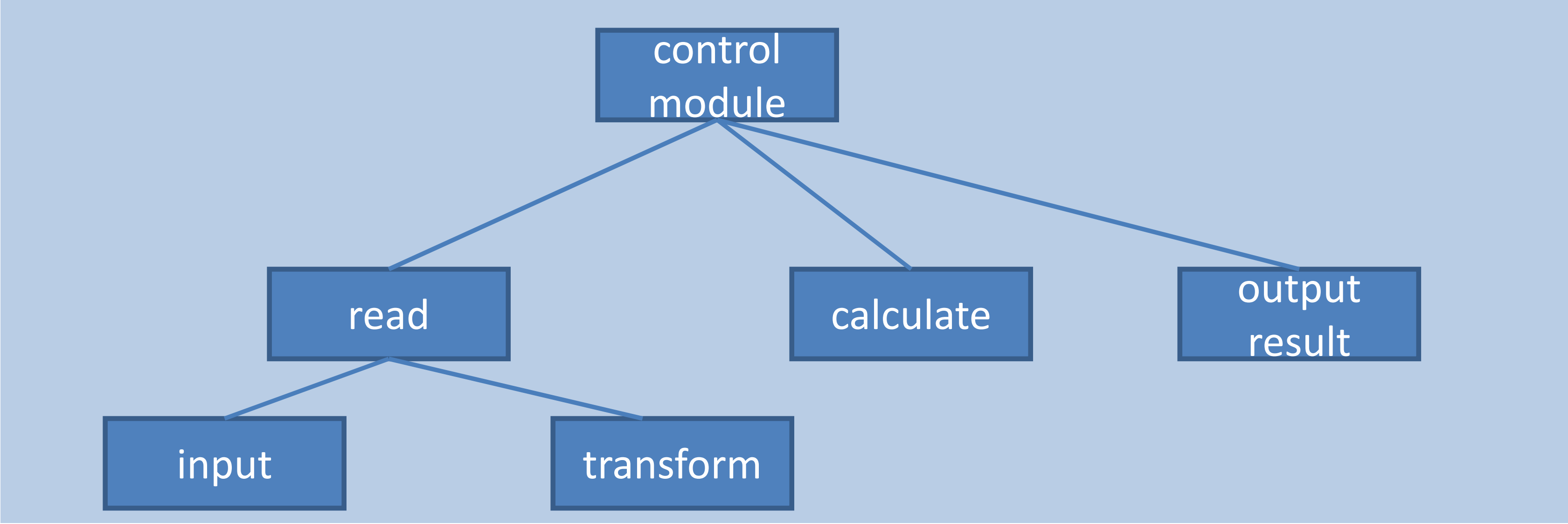
The Waterfall Model - Specification

- Functions
  - Calculation
- Behaviors
  - Input
  - Output
  - Display
- Interfaces
  - A window for displaying input and output
  - Buttons
- Performance （性能）
  - Response time: a micro-second

The Waterfall Model

- Design
  - Software architecture
  - Data structure
  - Design forms
    - Diagrams
    - Tables
    - Pseudo-code （伪代码）

The Waterfall Model - Architecture



The Waterfall Model - Coding

- Code
  - Code generation
    - automatically
    - manually （手工）

```
struct Exp transform(char *expression)
{
    …
}
```

The Waterfall Model
– Test Reports

- Test
  - – Detect defects in the programs
  - – Automatically or manually （手工的）

Defect （错误）report:

Module （模块） name: transform

Test data 1: 10 + 30

Output: 4

Test data 2: 20 * 30

Output 50

The Waterfall Model – Maintenance Records

- Maintenance （维护）
  - Correct bugs
  - Implement the changes

Maintenance report:

Time: 1 January 2000

Module: transform

Reason: Fix bugs

Bug description: …

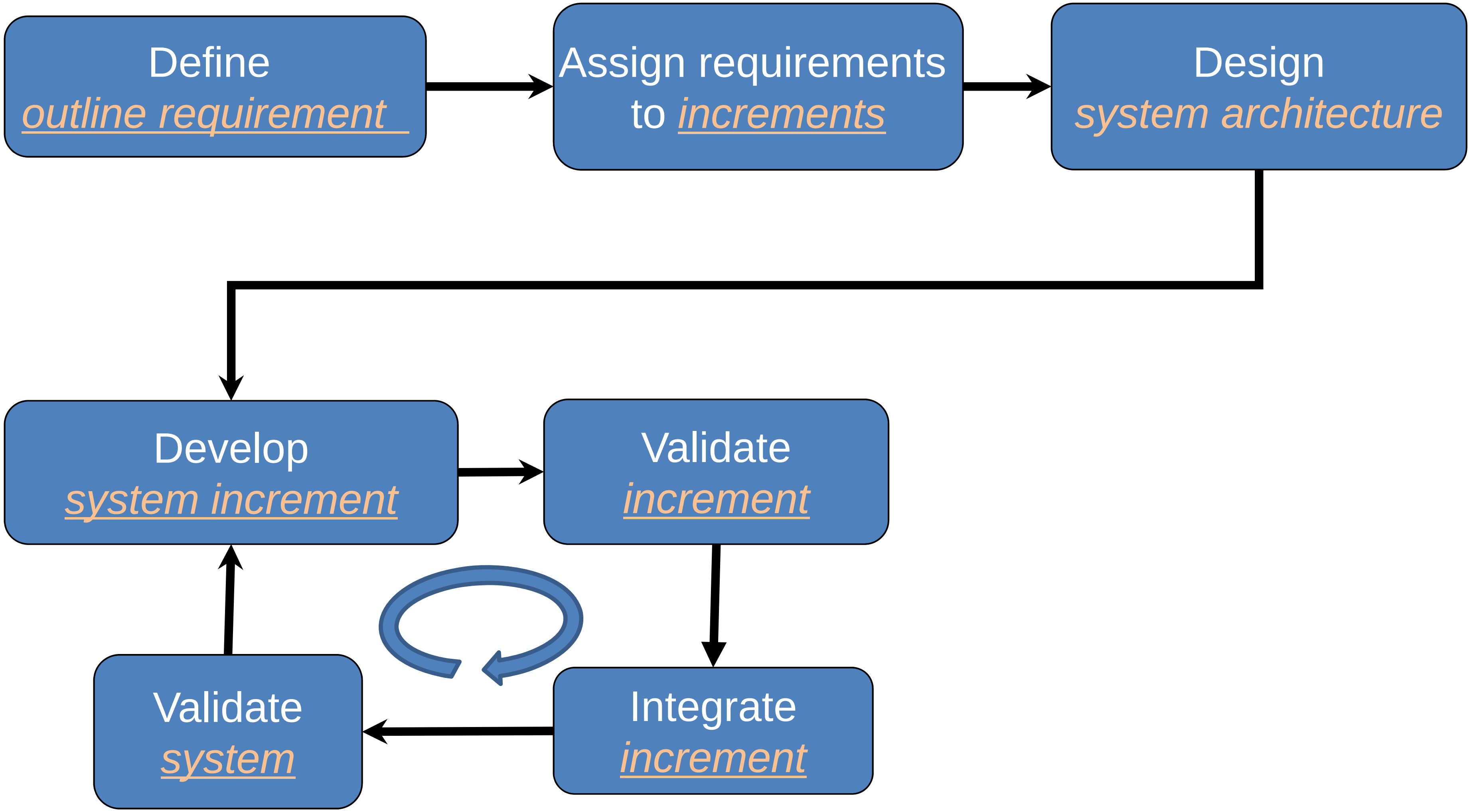
The Waterfall Model

- Advantages
  - Straightforward （率直的）
  - Disciplined （严格）
- Arguments （争拗）
  - Users cannot describe their requirements very clearly at the beginning
  - It is hard to follow
  - The projects developed with this process are often delayed （延迟）

Incremental Development

- Clients do NOT need to specify the requirements for the whole system at the beginning
- Clients can specify the most important parts to finish first
- A number of increments （增加）are defined

Incremental Development Process



Define *outline requirement*

Assign requirements to *increments*

Design *system architecture*

Develop *system increment*

Validate *increment*

Integrate *increment*

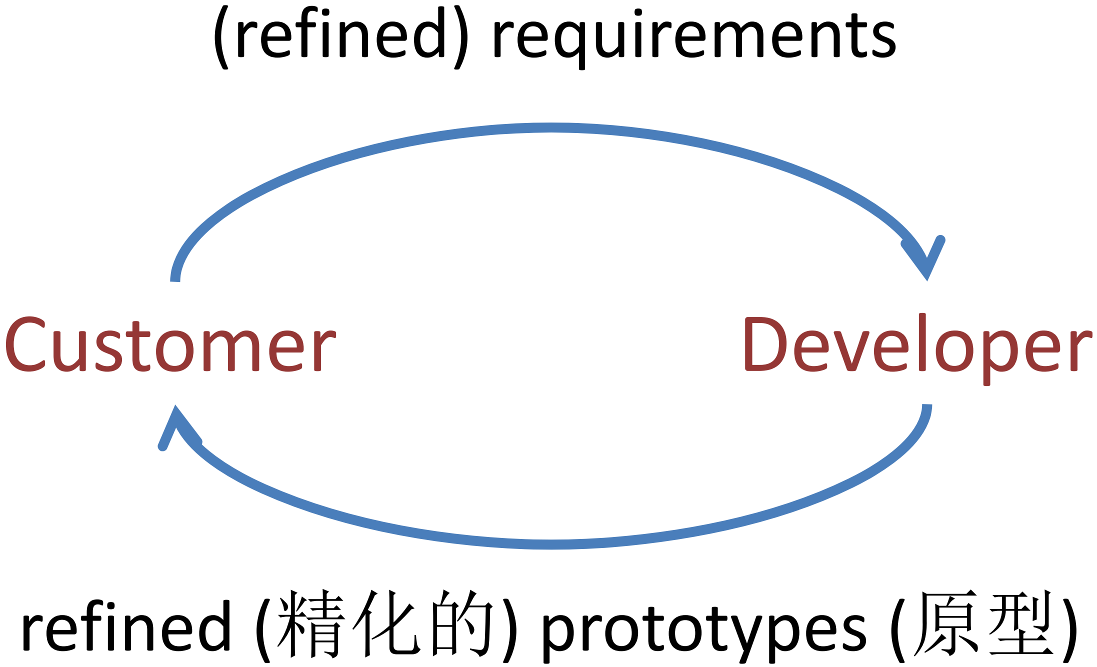
Validate *system*

The Incremental Model

- Advantages
  - The risk that the whole system fails is lower (compared to the waterfall model)
  - Shorten the delivery （交付）time
- Disadvantages
  - The system structure is loose
  - Hard to define the increments

The Prototyping Model

- A quick design -> the construction of a prototype
- Identify software requirements



(refined) requirements

Customer

Developer

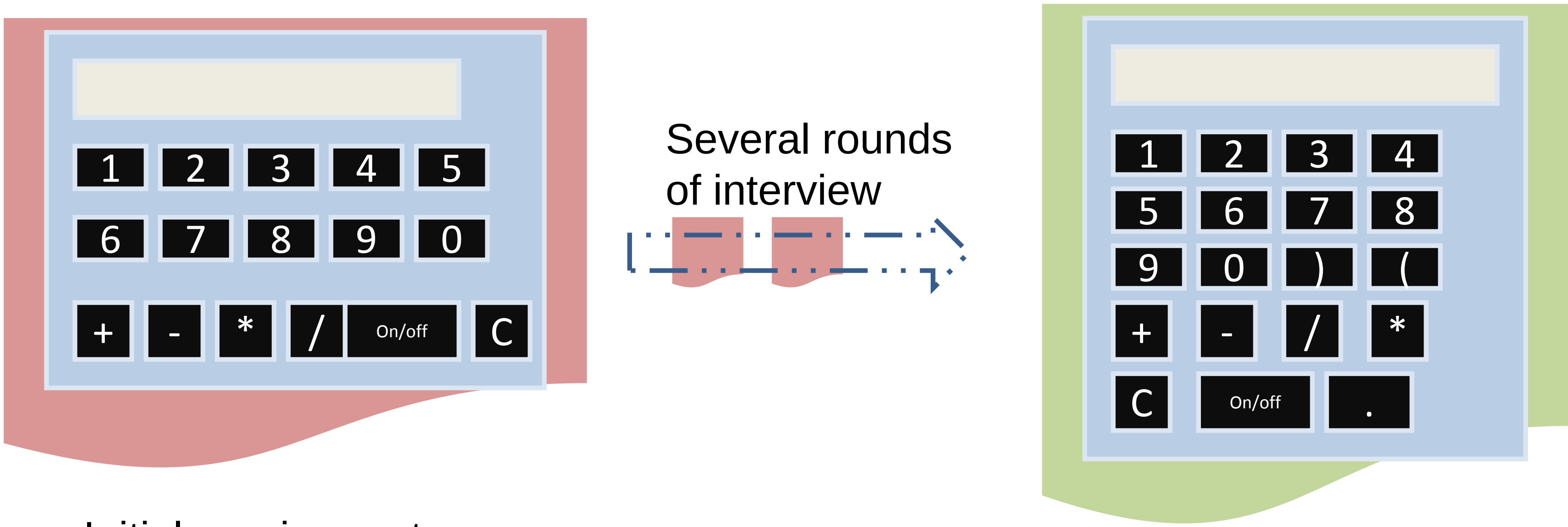
refined (精化的) prototypes (原型)

An Example

Requirements statement:

Develop a software system that can calculate arithmetic expressions and display the result in a micro second.

An Example



Several rounds of interview

Initial requirements

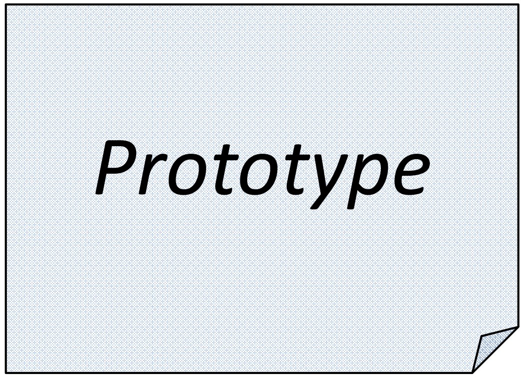
Refined requirements

Updated requirement statement:

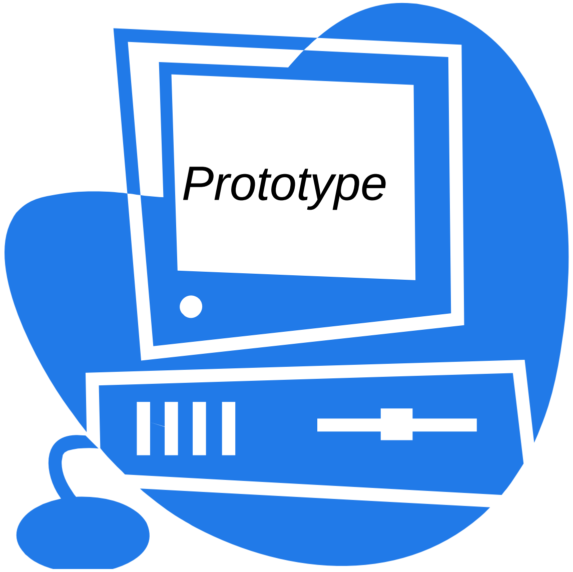
Develop a software system that can calculate arithmetic expressions and display the result in a micro second. The operands can be real and the precedence of the calculation can be designated using ()

Types of Prototypes

Paper work



Computer programs



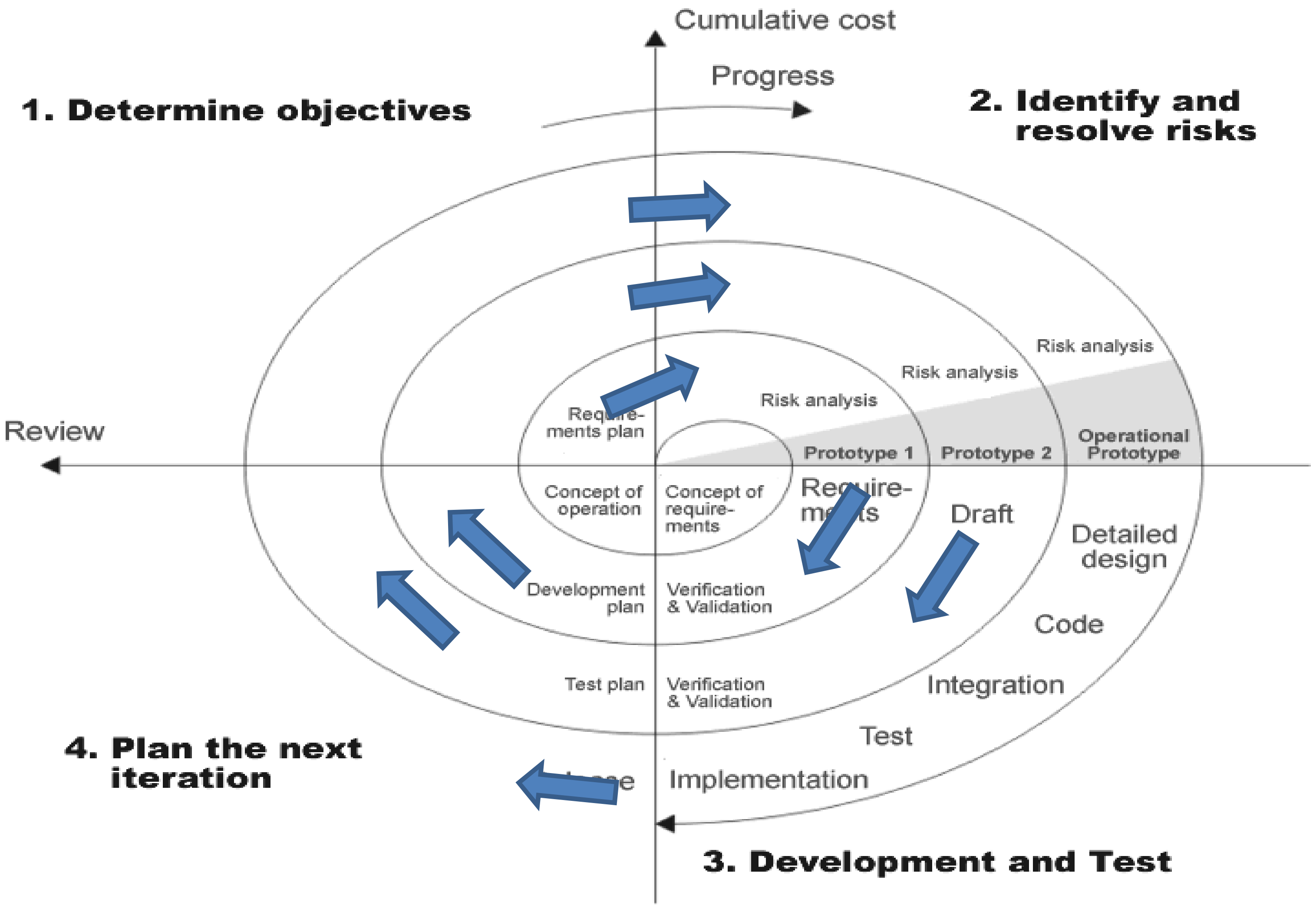
The Prototyping Model

- Advantages
  - Users can have an initial look of the system
  - Developers have a quick design
  - The prototype can be developed to a system
- Arguments
  - Rush to get it work
  - Hard to ensure the quality and to maintain

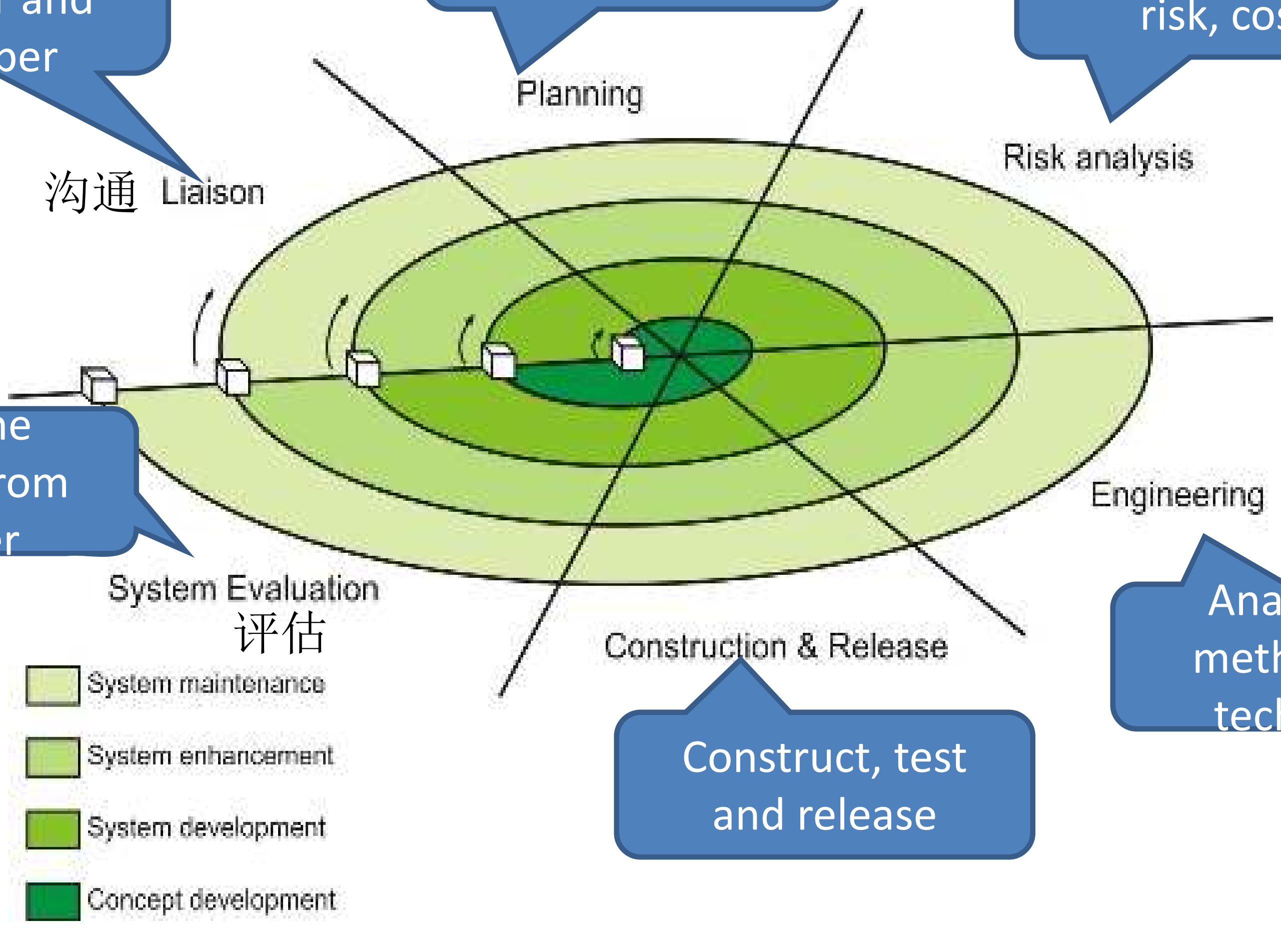
The Spiral Model

- An evolutionary（演变） (iterative) software process model
- Combine both waterfall model and prototyping model

The Spiral Model (Bohem)



A Variant of the Spiral Model



Communication between customer and developer

Define resource, time schedule

Assess technical risk, cost

Obtain the feedback from customer

Analyze the methods and techniques

Construct, test and release

The Spiral Model

- Advantages
  - Developers can using prototyping in each evolutionary level
  - Reduce the risk
- Arguments
  - Continual risk analysis
  - Document maintenance

Summary

- Activities in the phases of software development life cycle
- Process models
  - waterfall,
  - incremental,
  - prototyping,
  - Spiral
- Advantages and disadvantages of each model