[Process Activities](#3857-1572571056441)

[software specification](#4219-1571298031830)

[software design and implementation](#9578-1572569914966)

[the software design process](#2540-1571298731174)

[structured methods](#8412-1572570956646)

[the debugging process](#3414-1571298979848)

[software validation](#1711-1571298742949)

[software evolution](#6064-1571298054090)

[Rational Unified Process model](#6796-1571297987261)

[通常从三种视角来描述过程](#9937-1571299072102)

[ Inception](#7012-1571298009535)

[• Establish the business case for the system.](#4124-1571300608536)

[ Elaboration](#7967-1571300611727)

[• Develop an understanding of the problem domain and the system architecture.](#2240-1571300616803)

[ Construction](#4724-1571300622794)

[• System design, programming and testing.](#7665-1571300627570)

[ Transition](#5075-1571300631345)

[• Deploy the system in its operating environment](#0043-1571300635242)

[CASE](#7024-1571300593849)

[Activity automation](#1860-1571300835845)

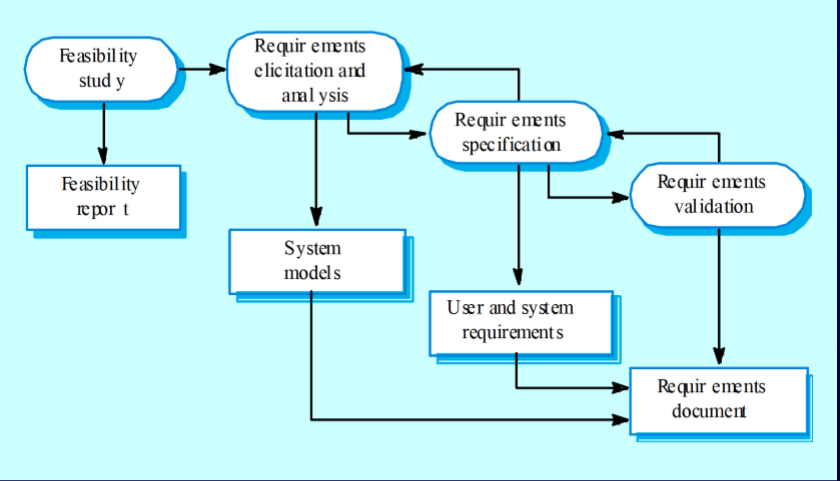
[Case technology](#8910-1571300886932)

[CASE classification](#8766-1571301029044)

[CASE integration](#3653-1571301112563)

**Process Activities**

**software specification**



可行性分析-需求启发和研究

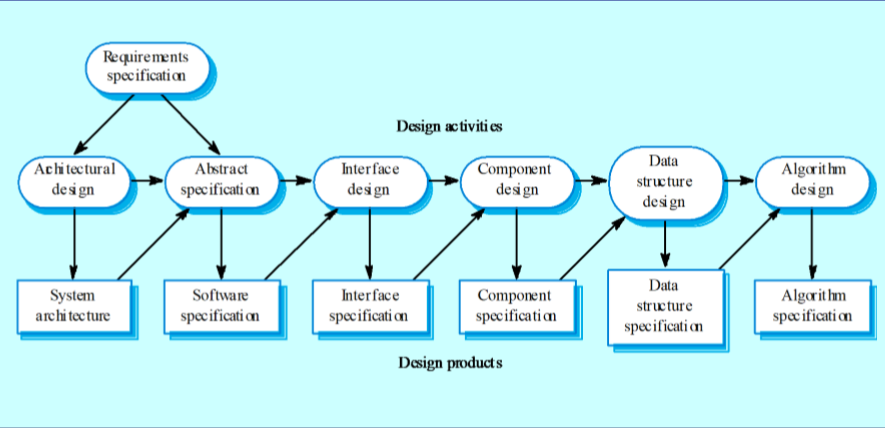
可行性报告 系统模型 需求说明 需求生效

用户和系统需求

需求文件

**software design and implementation**

**the software design process**



需求说明

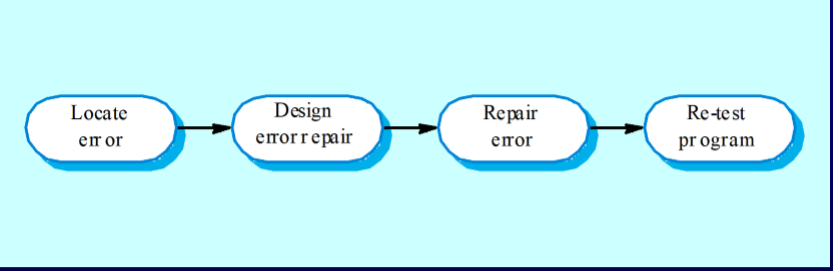
结构设计 抽象说明 用户界面设计 组成部分设计 数据结构设计 算法设计

系统结构 软件说明 用户界面说明 组成部分说明 数据结构说明 算法说明

**structured methods**

Possible models • Object model; • Sequence model; • State transition model; • Structural model; • Data-flow model.

**the debugging process**

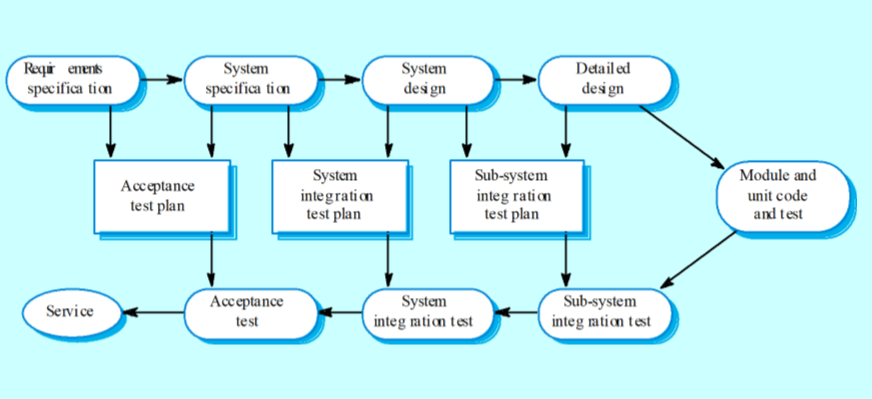


**software validation**

Component or unit testing

System testing

Acceptance testing

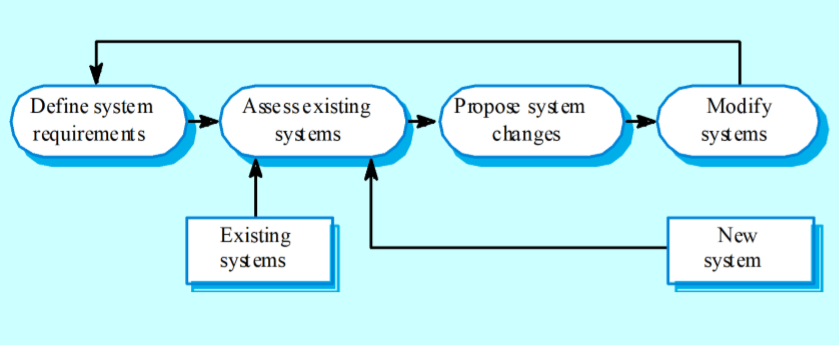


需求说明 系统说明 系统设计 细节化设计

验收测试计划 系统集成测试计划 子系统集成测试计划 单元代码和测试

服务 验收测试 系统集成测试 子系统集成测试

**software evolution**



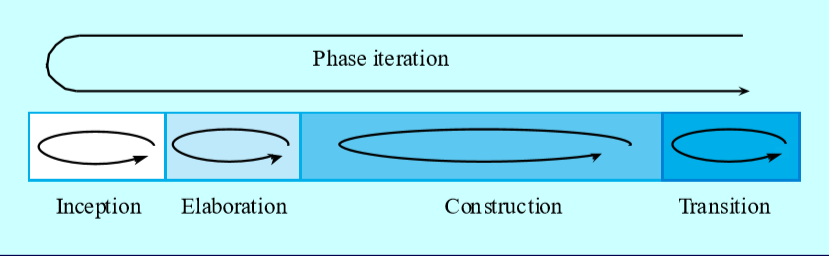
**Rational Unified Process model（RUP）**

• A dynamic perspective that shows phases over time;

• A static perspective that shows process activities;

• A practive perspective that suggests good practice.

阶段反复



RUP分离了软件过程中的四个独立阶段

开端 Inception

• Establish the business case for the system.

建立系统的业务案例。识别所有与系统交互的外部实体（人和系统）并定义这些交互。然后使用这些信息评估系统对业务的贡献。如果这个贡献是微笑的，那么项目就要在此阶段结束时被取消了。

细化 Elaboration

• Develop an understanding of the problem domain and the system architecture.

得到系统的需求模型，可能是一组用UML描述的用例，体系结构描述和开发计划

构造 Construction

• System design, programming and testing.

转换 Transition

• Deploy the system in its operating environment

**CASE**

**Activity automation**

• Graphical editors for system model development;

• Data dictionary to manage design entities;

• Graphical UI builder for user interface construction;

• Debuggers to support program fault finding;

• Automated translators to generate new versions of a program.

**Case technology**

Case technology has led to significant improvements in the software process. However, these are not the order of magnitude improvements that were once predicted

• Software engineering requires creative thought this is not readily automated;

• Software engineering is a team activity and, for large projects, much time is spent in team interactions. CASE technology does not really support these.

**CASE classification**

Classification helps us understand the different types of CASE tools and their support for process activities.

 Functional perspective

• Tools are classified according to their specific function.

 Process perspective

• Tools are classified according to process activities that are supported.

 Integration perspective

• Tools are classified according to their organisation into integrated units.

**CASE integration**

 Tools

• Support individual process tasks such as design consistency checking, text editing, etc.

 Workbenches

• Support a process phase such as specification or design, Normally include a number of integrated tools.

 Environments

• Support all or a substantial part of an entire software process. Normally include several integrated workbenches.

