

# **Ch.8 Understanding Requirements**













#### **Requirements Engineering**

- Inception—ask a set of questions that establish ...
  - basic understanding of the problem
  - the people who want a solution
  - the nature of the solution that is desired, and
  - the effectiveness of preliminary communication and collaboration between the customer and the developer
- · Elicitation—elicit requirements from all stakeholders
- Elaboration—create an analysis model that identifies data, function and behavioral requirements
- Negotiation—agree on a deliverable system that is realistic for developers and customers





## Requirements Engineering (cont.)

- Specification—can be any one (or more) of the following
  - A written document
  - A set of models
  - A formal mathematical
  - A collection of user scenarios (use-cases)
  - A prototype
- Validation—a review mechanism that looks for
  - errors in content or interpretation
  - areas where clarification may be required
  - missing information
  - inconsistencies (a major problem when large products or systems are engineered)
  - conflicting or unrealistic (unachievable) requirements.





#### Inception

- Identify stakeholders
  - "who else do you think I should talk to?"
- · Recognize multiple points of view
- Work toward collaboration
- The first set of context-free questions
  - Who is behind the request for this work?
  - Who will use the solution?
  - What will be the economic benefit of a successful solution
  - Is there another source for the solution that you need?





#### 需求工程

1、开始阶段:建立基本概念 万事开头难 搞清楚软件要干嘛 对解决方案相关的人要了解 软件特征(eg.多少人用) 建立沟通机制

2、需求获取阶段:和用户进行充分沟通

3、细化阶段:将用户需求没有异议地描述出来,即模型

4、协商阶段:综合所有人的意见,对模型查漏补缺,得到一个大家都认可的模型

5、要把需求规格说明书建立起来(可以有多种, 把问题表达清楚即可)

doc文档

各种模型 (UML模型) 做需求分析、设计形式化方法,用严格的数学语言 做原型系统

6、确认:查漏补缺,取得一致的认识

7、需求管理

首先识别所有stakeholder 还要注意到不同的观点,认识到谁说的是符合逻 辑的

建立协同机制

有哪些痛点, 获益点 需不需要其它资源等等







## **Eliciting Requirements**

- meetings are conducted and attended by both software engineers and customers
- rules for preparation and participation are established

· an agenda is suggested

 a "facilitator" (can be a customer, a developer, or an outsider) controls the meeting



需求获取阶段 要开会,深入了解用户 有时候要面基,也有在线的 开会首先要定义参与的规则,事先调研,告诉别 人,列出大致问题 要有会议日程,人不要太多,要控制会议规模 需要有个主持人



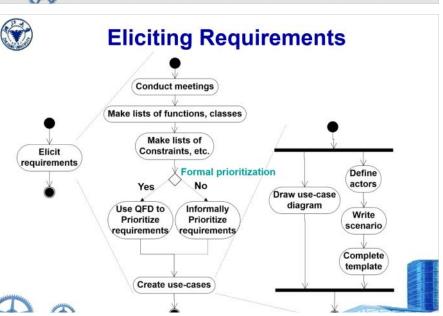


#### **Eliciting Requirements (cont.)**

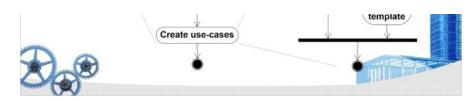
- a "definition mechanism" (can be work sheets, flip charts, or wall stickers or an electronic bulletin board, chat room or virtual forum) is used
- The hardest single part of building a software system is deciding what to build
  - Problem of scope
  - Problem of understanding
  - Problem of volatility

specify a preliminary set of solution requirements





要有问题定义的机制,如录音、录像、各种资料,不能开完会就忘记了,找到不同可能的解决方案





## **Quality Function Deployment**

- Function deployment determines the "value" (as perceived by the customer) of each function required of the system
- Information deployment identifies data objects and events
- Task deployment examines the behavior of the system
- Value analysis determines the relative priority of requirements
- · It identifies three types of requirements
  - Normal requirements
  - Expected requirements
  - Exciting requirements





#### **Non-Functional Requirements**

- Non-Functional Requirment (NFR) quality attribute, performance attribute, security attribute, or general system constraint. A two phase process is used to determine which NFR's are compatible:
  - The first phase is to create a matrix using each NFR as a column heading and the system SE guidelines a row labels
  - The second phase is for the team to prioritize each NFR using a set of decision rules to decide which to implement by classifying each NFR and guideline pair as complementary, overlapping, conflicting, or independent





#### **Elicitation Work Products**

- a statement of need and feasibility.
- a bounded statement of scope for the system or product
- a list of customers, users, and other stakeholders who participated in requirements elicitation
- · a description of the system's technical environment.
- a list of requirements (preferably organized by function) and the domain constraints that apply to each
- a set of usage scenarios that provide insight into the use of the system or product under different operating conditions

标准需求

期望需求:例如期望系统比较稳定

惊喜需求: 意料之外

非功能性需求

各种质量要求、性能要求、安全性要求

用类似矩阵分析的方法来做,需求-功能,形成分析矩阵再分析每个task

表中的每个cell做决策分析,相关性分析

产品可行性

范围

有哪些用户

运行环境

各种约束

场景

- a set of usage scenarios that provide insight into the use of the system or product under different operating conditions.
- any prototypes developed to better define requirements





#### **Use-Cases**

- A collection of user scenarios that describe the thread of usage of a system
- Each scenario is described from the point-of-view of an "actor"—a person or device that interacts with the software in some way
- Each scenario answers the following questions:
  - Who is the primary actor, the secondary actor (s)?
  - What are the actor's goals?
  - What preconditions should exist before the story begins?
  - What main tasks or functions are performed by the actor?





#### **Use-Cases**

- Each scenario answers the following questions (cont.)
  - What extensions might be considered as the story is described?
  - What variations in the actor's interaction are possible?
  - What system information will the actor acquire, produce, or change?
  - Will the actor have to inform the system about changes in the external environment?
  - What information does the actor desire from the system?
  - Does the actor wish to be informed about unexpected changes?



#### A Example -- SafeHome

Our research indicates that the market for home security systems is growing at a rate of 40% per year. We would like to enter this market by building a microprocessor-based home security system that would protect against and/or recognize a variety of undesirable situations such as illegal entry, fire, flooding, and others. The product will use

场景

原型系统定义需求

用例图来定义需求用户的一系列场景

每个用例图代表某几个角色,希望和系统有哪些 交互

每个场景对应一个角色(可能是一个人,可能是一个设备)

一个人或设备可以做多个角色

谁是主要发起者?

主要目标是什么?

先决条件?

主要任务和功能?

外延的功能?

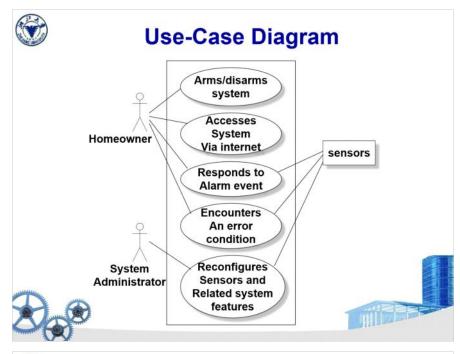
哪些变化性(eg. 上传课件在外部能不能上传[手机上]?) 系统信息?

外界信息需不需要告诉系统?

安防系统

recognize a variety of undesirable situations such as illegal entry, fire, flooding, and others. The product will use appropriate sensors to detect each situation, can be programmed by the homeowner, and will automatically telephone a monitoring agency when a situation is detected.





#### 小人就是角色

软件就是框里的

- 1、报警
- 2、通过互联网访问系统
- 3、检测问题

•••

硬件就是传感器



## **Building the Analysis Model**

- · Elements of the analysis model
  - Scenario-based elements
    - Use-case and user-case diagram
    - · Sequence of activities within certain context
  - Class-based elements
    - Class diagram
  - Behavioral elements
    - · State diagram
  - Flow-oriented elements
    - · Data flow diagram





#### 3、状态图 形容系统的状态及状态迁移

2、基于类的

建立需求模型

1、基于场景的元素

从角色的角度看,

++---

User-case、序列图、泳道图

表示系统的静态数据结构

4、基于流

结构化分析方法,例如数据流、控制流

对得到的原始材料进行消化吸收,变成原始模型



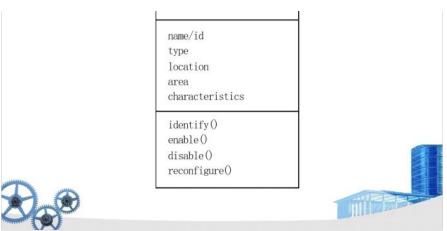
### **Class Diagram**

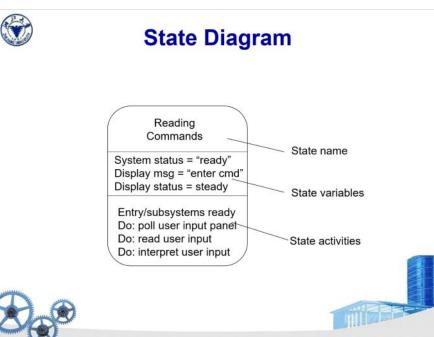
From the SafeHome system ...

Sensor

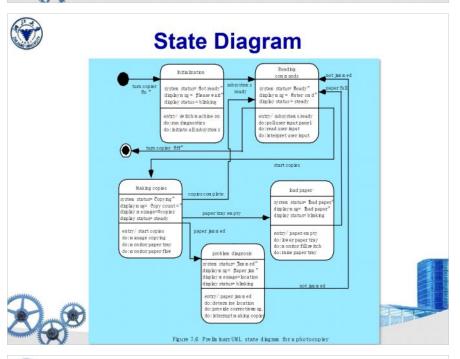
name/id
type

每个类包含类名、属性、成员函数





状态图分为状态和状态变化 每个状态由状态名字、状态变量(这个状态下系统的参数怎么样)、状态活动(提供各种动作, eg. 显示用户输入)



用箭头将状态连接,表示状态迁移,箭头上面有事件,事件来了之后会触发状态迁移 Eg. reading状态已经输入密码了,就会迁移状态

#### **Analysis Patterns**

Pattern name: A descriptor that captures the essence of the pattern.

Intent: Describes what the pattern accomplishes or represents

分析模式,模式是经验的总结

- 1、名字
- 2、意图
- 3、内容
- 4 结果

ше рацеті.

Intent: Describes what the pattern accomplishes or represents

**Motivation:** A scenario that illustrates how the pattern can be used to address the problem.

Forces and context: A description of external issues (forces) that can affect how the pattern is used and also the external issues that will be resolved when the pattern is applied.

Solution: A description of how the pattern is applied to solve the problem with an emphasis on structural and behavioral issues.

- 2、意图
- 3、内容
- 4、结果



#### **Analysis Patterns (cont.)**

Consequences: Addresses what happens when the pattern is applied and what trade-offs exist during its application.

**Design:** Discusses how the analysis pattern can be achieved through the use of known design patterns.

Known uses: Examples of uses within actual systems.

Related patterns: One or more analysis patterns that are related to the named pattern because (1) it is commonly used with the named pattern; (2) it is structurally similar to the named pattern; (3) it is a variation of the named pattern.





## **Negotiating Requirements**

- · Identify the key stakeholders
  - These are the peone negotiation

If different customers/users cannot agree on requirements, the risk of failure is very

- Determine each of
  - Win condition are not usual
- Negotiate
  - Work toward a set of requirements that lead to "win-win"





### Validating Requirements

. Is each requirement consistent with the overall

6、设计

5、结果

需求协商

目的是将不同人的观点协商一致 客户希望功能越多越好、越方便越好,但可能对 开发者来说很累

知道用户的底线,希望达到双赢



#### Validating Requirements

- Is each requirement consistent with the overall objective for the system/product?
- Have all requirements been specified at the proper level of abstraction? That is, do some requirements provide a level of technical detail that is inappropriate at this stage?
- Is the requirement really necessary or does it represent an add-on feature that may not be essential to the objective of the system?

做确认需求 和整个定义是否一致。

需求是否在一个恰当的抽象级

看一下需求的必要性





#### Validating Requirements (cont.)

- · Is each requirement bounded and unambiguous?
- Does each requirement have attribution? That is, is a source (generally, a specific individual) noted for each requirement?
- Do any requirements conflict with other requirements?
- Is each requirement achievable in the technical environment that will house the system or product?
- Is each requirement testable, once implemented?

是否没有歧义 每个需求都要有归属

不能相互矛盾

每个需求是否可以实现

是否可测试?





#### Validating Requirements (cont.)

- Does the requirements model properly reflect the information, function and behavior of the system to be built.
- Has the requirements model been "partitioned" in a way that exposes progressively more detailed information about the system.
- Have requirements patterns been used to simplify the requirements model. Have all patterns been properly validated? Are all patterns consistent with customer requirements?





恰当地反馈系统的功能





## **Requirements Monitoring**

Especially needes in incremental development

- •Distributed debugging uncovers errors and determines their cause
- •Run-time verification determines whether software matches its specification
- •Run-time validation assesses whether evolving software meets user goals
- •Business activity monitoring evaluates whether a system satisfies business goals
- •Evolution and codesign provides information to stakeholders as the system evolves

对需求进行监控 对需求进行编号

调试环节

运行环境确认

运行环境验证