

## **Use Case диаграми и изисквания в софт. инженерство - резюме**

**Моделиране на изисквания:** Създават се два типа модели: **на изисквания и дизайн** модели

- Модели на изисквания - изискванията на клиента в три домейна:

1/ Информационен; 2/ Функционален; 3/ Поведенчески;

- Модели на дизайна - характеристиките на софтуера

които помагат на практикуващите да го конструират ефективно:

1/ Софт.арх.; 2/ потребит. интерфейс; 3/ component-level details;

**Видове изисквания: потребителски, системни, функ., нефунк.**

**Сценариите** са примери от живота как една система

може да се използва. Включват описание на: 1/ началната ситуация; 2/ поток от събития; 3/ това, което може да се обърка; 4/ др. съпътств. дейности; 5/ състоянието, когато сценарият приключи

**Use Case диаграми** – UML графично представяне на взаимод. в с-мата със сценарии. Състои се от:

**-Актьори** - външни потребители/ с-ми, които взаимод. със с-мата

**-Use cases** - случаи на употреба, описващи конкретни действия

**- Associations (relationships)** – взаимодействие между актьор и Use Case- solid line; Стрелката често се използва за указване на посоката

на първоначалното възникване на връзката (но не и на

посока на обмен на информация); Multiplicities

**- Other relations:** 1/Include (включване на друг use case) 2/ Extend (разширяване на друг use case) 3/Generalize 4/Depend

**- System boundary boxes** - (по избор) - правоъгълници около

случайте на употреба за посочва обхвата на вашата система

**-Packages** - (по избор) - UML конструкции, които ви позволяват организирате елементи на модела в групи

**-Collaboration**

**- Realization**

Flow of Events Of Use Case: 1/ Basic (the straight arrow) and alternative flows of events (the curves) 2/ A pre-condition 3/ A post-condition

**Use Cases да бъдат: 1/Concrete 2/Abstract (italics)**

Могат да имат: **ID, Rank, Leaf and Root of use cases**

Документиране на изисквания - включва описание на:

Preface; Въведение; Речник на термините Glossary; Потребителски изисквания; Архитектура на системата; Системни изисквания; Системни модели; Еволюция на системата; Appendices; Index

**Ключови принципи:** 1/Пълнота и съгласуваност на изискванията 2/ Яснота и недвусмисленост 3/Проследимост 4/Възможност за проверка и валидация

Методологията на **Use Case** диаграмите помага на разработчиците да разберат: 1/**Кои са потребителите на системата** 2/ **Какви функции трябва да изпълнява системата** 3/ **Как различните компоненти си взаимодействат** 4/**Какви са очакванията на потребителите**

**Use Case** диаграмите са ключов инструмент в обектно-ориентириания анализ и дизайн, който позволява ясно и структурирано описание на изискванията към дадена софт. система.

 **Цел на Use Case диаграмата** - Use Case диаграмите обикновено се разработват в ранните етапи на разработката на софтуер. Те се използват за следните цели:

- **Определяне на контекста на системата**
- **Улавяне на изискванията** към системата
- **Валидиране на архитектурата** на системата
- **Насочване на имплементацията** и създаване на тестови случаи
- **Разработват се от анализатори заедно с домейн експерти**

 **Общ поглед върху Use Case диаграмите-основни елементи:**

## 1Актьор (Actor)

- Представлява **някой**, който взаимодейства със системата.
- Именува се с **съществително**.
- Актьорът играе роля в бизнеса (например: *Потребител*, *Администратор*).
- Един и същ потребител може да има различни роли (*Преподавател* и *Изследовател*).
- **Задейства Use Case** и има очаквания към системата (вход/изход).

## 2Use Case

- Описва **функционалност** на системата (процес, който може да е автоматизиран или ръчен).
- Именува се с **глагол + съществително** ("Добавяне на продукт", "Изпращане на съобщение").

- Всеки актьор трябва да бъде свързан с Use Case, но не всеки Use Case изисква актьор.

### 3 Връзка за комуникация (Communication Link)

- Показва участието на актьора в Use Case чрез **солидна линия**.
- Показва как актьорът и Use Case комуникират чрез съобщения.

### 4 Граница на системата (System Boundary)

- Определя се от документа с изискванията.
- За големи системи всяка подсистема може да има своя граница (например в ERP система: *Персонал*, *Заплати*, *Счетоводство* и т.н.).

#### 🔗 Структуриране на Use Case диаграмата с връзки

##### 👉 Видове връзки между Use Cases:

**Extends (<<extends>>)** Показва, че един Use Case **добавя допълнително поведение** към основен Use Case. Например: "*Невалидна парола*" разширява "*Вход в акаунт*". Изобразява се с **пунктирана стрелка**, сочеща към основния Use Case.

**Include (<<include>>)** Показва, че един Use Case **включва функционалност** от друг Use Case. Например: "*Потвърждение на поръчка*" включва "*Изчисляване на обща сума*". Използва се за **повторно използване** на общо поведение.

**Generalization (Обобщение)** Показва **родител-дете връзка** между Use Cases. Дъщерният Use Case **наследява поведението** на родителския и може да го разширява или променя. Изобразява се с **стрелка с триъгълна глава**, сочеща към родителя.