

Компоненти и диаграми на внедряване - Резюме

Имплементация на софтуер: Етап в процеса на софтуерно инженерство, при който се разработва изпълнима софтуерна система.

Проектиране и имплементация: взаимосвързани дейности, вкл. идентифициране на софт. компоненти и техн. връзки, базирани на изискванията на клиента, и реализиране на дизайна като програма.

Процес на обектно-ориентирано проектиране. Етапи на процеса:

- Определяне на контекста и начина на използване на системата;
- Проектиране на архитектурата на системата;
- Идентифициране на основните обекти в системата;
- Разработване на дизайнерски модели;
- Определяне на интерфейсите на обектите.

Модел на контекста на системата: Структурен модел, който показва другите системи в околната среда на разработваната система.

Модел на взаимодействията: Динамичен модел, който показва как с-мата взаимодейства с околната среда по време на използването си.

Архитектурен дизайн: Включва идентифициране на основните компоненти на системата и техните взаимодействия, и организиране на компонентите с помощта на архитектурен модел като слоест или клиент-сървър модел.

Идентификация на обектни класове. Подходи за идентификация:

- 1/Граматичен- базиран на естествено езиково описание на системата
- 2/Идентификация, основана на осезаеми неща в приложната област
- 3/Поведенчески- базиран на участието на обектите в опр. поведение

4/Анализ, базиран на сценарии.

Дизайнерски модели:

1/**Статични** - описват статичната структура на системата чрез класове на обекти и техните връзки

2/**Динамични** - описват как обектите взаимодействат помежду си във времето

Модел на имплементация: Колекция от компоненти и имплементационни подсистеми, които ги съдържат

Имплементационната подсистема е колекция от компоненти и други имплементационни подсистеми, които се използват за структуриране на модела на имплементация чрез разделяне на по-малки части.

Пакети от компоненти представляват групи от логически свързани компоненти или основни части на системата.

Компоненти: Представляват части от софтуерен код или файл, съдържащ информация. Могат да имат стереотипи като <<component>>, <<subsystem>> и др.

Диаграми на компоненти: Показват колекция от декларативни (статични) моделни елементи като компоненти и имплементационни подсистеми и техните връзки.

Кръг с интерфейс, прикрепен към иконата на компонента, означава, че компонентът поддържа конкретния **интерфейс**. Няма изрична връзка със стрелка между компонента и неговите интерфейси.

Спецификация на компонента **съдържа:**

-Общи– стереотипи(Main Program, Package Body, Package Specification, Subprogram Body, Subprogram Specification, Task Body, Task Specification) и език.

-Детайли – декларации (като #Include).

-Realizes – класове, изграждащи компонента.

-Прикрепени файлове или URL адреси.

Зависимости: Зависимост от компонент А към компонент В показва, че А има компилационна или изпълнителна зависимост към В.

Зависимост за компилация съществува от един компонент към компонентите, които са нужни за компилиране на компонента (например import в Java).

Съединителни връзки (Assembly connectors) в UML 2.*

-Съединителната връзка свързва необходимия интерфейс на един компонент с предоставения интерфейс на друг компонент

-Съединителната връзка позволява на един компонент да предоставя услугите (boll), които друг компонент изисква (контакта).

Портове: Моделират свързани интерфейси и показват предоставяне услуги или поведение към околната среда, както и услуги или поведение, които компонентът изисква. Входи и изходи

Съединители за сглобяване: Свързват изисквания интерфейс на един компонент с предоставения интерфейс на друг компонент.

Диаграми на внедряване/Deployment/: Показват конфигурацията на елементите за обработка в реално време и софт. процеси, които се изпълняват в тях. Представят топологията на системата и разпределението на изпълнимите подсистеми към процесорите.

Диаграмата за внедряване показва процесори, устройства и връзки. Всеки модел съдържа единствена диаграма за внедряване, която показва връзките между процесорите и устройствата и разпределението на процесите върху процесорите.

Възли/Nodes/: Хардуерен или софтуерен ресурс, който може да хоства софтуер или свързани файлове. **Възлите** могат да бъдат хардуерни (сървъри, раб. станции) или софт. (операционни системи).

Artifact в възел- показва, че артефактът е внедрен в този възел

Процесор: Идентифицира процесите и специфицира типа на планиране на процесите (прекъсваемо, непрекъсваемо и т.н.).

Устройство: Хардуерен компонент с ограничена изчислителна мощност(модем, терминал)

Връзка: хардуерно свързване между две единици (процесор или устройство).