

Компоненти и диаграми на внедряване - Резюме

Проектиране и имплементация:

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

-Проектиране и имплементация: Тези дейности са взаимосвързани и включват идентифициране на софтуерни компоненти и техните връзки, базирани на изискванията на клиента, и реализиране на дизайна като програма.

Процес на обектно-ориентирано проектиране

Етапи на процеса:

1/Определяне на контекста и режимите на използване на системата.

2/Проектиране на системната архитектура.

3/Идентифициране на основните системни обекти.

4/Разработване на дизайнерски модели.

5/Специфициране на интерфейсите на обектите.

Модели на контекста и взаимодействията:

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-та 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 подход, базиран на естествено езиково описание на системата.

2/Идентификация, базирана на осезаеми неща в приложната област.

3/Поведенчески подход, базиран на участието на обектите в определено поведение.

4/Анализ, базиран на сценарии.

Дизайнерски модели:

1/**Статични** модели: Описват статичната структура на системата в термини на обектни класове и връзки.

2/**Динамични** модели: Описват динамичните взаимодействия между обектите.

Спецификация на интерфейси: Интерфейсите на обектите трябва да бъдат специфицирани, за да могат обектите и другите компоненти да бъдат проектирани паралелно.

Компоненти и диаграми на внедряване

Модел на имплементация в изгледа на компонентите

Модел на имплементация: Колекция от компоненти и имплементационни подсистеми, които ги съдържат. Компонентната диаграма има по-високо ниво на абстракция от диаграмата на класовете.

Компоненти: Представяват части от софтуерен код (изходен, бинарен или изпълним), или файл, съдържащ информация. Компонентите 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.

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

Портове: Моделират свързани интерфейси и позволяват специфициране на услуги или поведение към околната среда, както и услуги или поведение, които компонентът изисква.

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

Диаграми на внедряване: Показват конфигурацията на елементите за обработка в реално време и софтуерните процеси, които живеят в тях. Представят топологията на системата и разпределението на изпълнимите подсистеми към процесорите.

Възли: Хардуерен или софтуерен ресурс, който може да хоства софтуер или свързани файлове. Възлите могат да бъдат хардуерни (сървъри, работни станции) или софтуерни (операционни системи, контейнери).

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

Устройство: Хардуерен компонент с ограничена изчислителна мощност.

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