Программная инженерия, лекция 3: Варианты использования

Пожидаев М. С.

26 сентября 2022 г.

Назначение

Зачем нужны варианты использования?

Вариант использования определяет последовательность действий, которые может выполнить система, приносящих ощутимый и измеримый результат, и значимый для некоторого актанта.

1. Варианты использования дают систематический и интуитивно понятный способ определения функциональных требований, ориентированный на получение необходимых пользователю результатов.
2. Варианты использования управляют всем процессом разработки. Все основные виды деятельности (анализ, проектирование и тестирование) выполняются на основе определенных вариантов использования.

Варианты использования, приносящие результаты с отрицательной ценностью, не считаются вариантами использования. Мы будем называть их «вариантами запрещенного использования».

Оценка критичности

Когда применять варианты использования?

1. В системе преобладают функциональные требования.
2. В системе много типов пользователей, которым она предоставляет разные функциональные возможности (много актеров).
3. В системе много интерфейсов.

Фокусируйтесь на том «что», а не «как».

Определение use cases

Кто разрабатывает варианты использования?

Определение вариантов использования проводится при участии:

- пользователей (экспертиза требований);
- заказчиков;
- разработчиков (экспертиза требований, облегчение обсуждения и помощь пользователям и заказчикам в изложении своих пожеланий).

Модель вариантов использования применяется для достижения соглашения с пользователями и заказчиками о функциях будущей системы.

Мир вокруг

Актанты и роли

Актанты — это среда, в которой существует система.

1. Актанты — это не обязательно люди. Актантами могут быть другие системы или внешнее оборудование компьютера, взаимодействующее с системой.
2. Каждый актант, взаимодействуя с системой, исполняет согласованный набор ролей.
3. Физический пользователь может играть роль одного или нескольких актантов, выполняя их функции по взаимодействию с системой.

Модель использования

Актанты $\leftrightarrow$ варианты использования

Определение модели вариантов использования

Модель вариантов использования — это полный набор актантов и вариантов использования.

Каждая категория пользователей представлена отдельным актантом. Актанты используют систему, взаимодействуя с вариантами использования. Диаграмма использования описывает часть модели вариантов использования, показывая набор вариантов использования с актантами и с ассоциациями между каждой взаимодействующей парой актант — вариант использования.

Три модели

Варианты использования в трёх моделях

1. Модель вариантов использования.
2. Модель анализа.
3. Модель проектирования.

В ходе анализа и проектирования модель вариантов использования преобразуется через модель анализа в модель проектирования, то есть в структуру, состоящую из классификаторов и реализаций вариантов использования.

Классы

Стереотипы анализа

В модели анализа используется три стереотипа классов :

- граничный класс;
- управляющий класс;
- класс сущности.

Пример классов

Вариант для банка

1. Устройство выдачи и интерфейс кассира — это граничные классы. Они в основном используются для моделирования взаимодействия между системой и ее актантами.
2. Управление выдачей наличных — управляющий класс. Классы такого типа в основном используются для представления координации, последовательности, взаимодействия и управления другими объектами.
3. Банковский счет — класс сущности;. Такие классы в основном используются для моделирования инкапсуляции.

Назначение класса

Обязанности и роли

Что такое обязанность класса?

Обязанности класса — это просто сумма всех ролей, которые он исполняет во всех реализациях вариантов использования.

Сложив все роли и удалив перекрывающиеся части ролей, мы получим список всех обязанностей и атрибутов класса.

Кто отвечает?

За определение ролей класса отвечают разработчики, выполняющие анализ и реализацию.

Подсистемы

Группировка классов

Классы группируются в подсистемы!

Для большой системы, состоящей из сотен или тысяч классов, невозможно реализовать варианты использования, пользуясь только классами. Такая система будет слишком велика для того, чтобы в ней можно было обойтись без группировки высшего порядка.

Подсистема — это осмысленная группировка классов или других подсистем. Подсистема имеет набор интерфейсов, которые обеспечивают ее существование и использование.

Спасибо за внимание!

Всё о курсе: https://marigostra.ru/materials/engineering.html
E-mail: msp@luwrain.org
Канал в Телеграм: https://t.me/MarigostraRu