

Feature-driven development

Возможные вопросы к презентации

Слайд «введение»

Вопрос: вы говорите, будто feature driven development объединяет **лучшие практики**, но в своей презентации вы не говорите ни о каких таких лучших практиках; лишь в общем описываете feature driven development, как он работает, почему?

Что это за «лучшие практики»?

Ответ: да, вы абсолютно правы; мы не покрыли данный аспект в своей презентации; но лишь потому, что мы считаем, что «лучшие практики» — вещи, не так сильно зависящие от методологии, не уникальные для отдельной методологии. И именно поэтому мы не стали много говорить о них.

Что это за «лучшие практики»? В их список входят, например, код-ревью, регулярные сборки, и постоянные отчеты, позволяющие в любой момент времени четко отслеживать прогресс и результаты.

Вопрос: что это за исследования, на которые вы ссылаетесь?

Ответ: в нашей презентации есть ссылка на него; вы можете скачать нашу презентацию по представленному QR-коду

Вопрос: что это еще за *project management institute*, никогда о них не слышал. Какие-то шарлатаны, продвигающие *feature driven development*?

Ответ: project management institute — международное профессиональное объединение, существующее с 1969 года, в котором состоит более 680 тысяч членов из более чем двухсот стран.

Оно занимается разработкой стандартов в своей сфере, много лет предлагает услуги профессиональной сертификации

Вы можете быть уверены, что их данные насчет использования feature driven development в различных организациях абсолютно честны

Слайд «история»

Вопрос: вы говорите про одного Джеффа Де Луку, но Википедия говорит о еще нескольких людях, приложивших руку к созданию feature driven development! Например, про Питера Кода (Peter Coad)

Ответ: вы правы; feature driven development, как и любая другая методология, возникла не из пустоты; на нее большое влияние оказал подход к объектному моделированию, который Код описал в своей книге. В действительности, *feature driven development* во многом построен именно на объектном моделировании, это очень легко проследить и достаточно очевидно.

Тем не менее, во всей своей целостности feature driven development сформулировал именно Джефф Де Лука

Вопрос: что это за «крупный сингапурский банк»?

Ответ: эти сведения, по всей видимости, конфиденциальны; по крайней мере, они нигде никогда не разглашались. К сожалению, мы не можем дать вам ответ на данный вопрос

Слайд «разработка общей модели»

Вопрос: я не понимаю, все же, что такое этот «высокоуровневый анализ решаемого круга задач и контекста системы»?

Ответ: под высокоуровневым анализом понимается анализ общей картины: у нас есть цель, как ее достичь, какие компоненты должны быть у нашей программы? В самых общих чертах, не углубляясь в подробности

Допустим, мы разрабатываем интернет магазин. На этапе разработки общей модели мы определяем его базовые компоненты, что должно быть у него?

Ну, это определенно какой-то каталог товаров, отдельные странички товаров с более подробной информацией о них, личный кабинет с возможностью регистрации и авторизации, совершения платежей, некоторые другие общие моменты

На этом, собственно, общий анализ и заканчивается. Пока что мы не углубляемся в детали отдельных компонентов: а, например, как мы будем обеспечивать безопасность данных пользователей? А какими платежными шлюзами мы будем пользоваться?

Это то, что мы оставляем на дальнейшие этапы, когда мы собственно будем проводить **детальный анализ каждой отдельной моделируемой области**

Вопрос: вы говорите, что описания моделей составляются в небольших группах и выносятся на дальнейшее обсуждение и экспертную оценку. Кто осуществляет эту экспертную оценку?

Ответ: как правило это ведущие программисты; люди, имеющие большой опыт в моделировании систем.

Вопрос: вы говорите, что модели каждой области задач объединяются в общую итоговую модель, которая изменяется в процессе работы; как так, она итоговая, но изменяется?

Ответ: это отличный вопрос, спасибо; впрочем, тут, по-моему, довольно все просто. Мы все понимаем, что даже если мы все детально обсудили и согласовали всей командой, и эксперты признали, что все хорошо — это не гарантирует отсутствие каких-либо просчетов

Может обнаружиться, что мы забыли что-то учесть в нашей модели. Или попросту наш заказчик ввел какие-либо новые требования

Но не стоит думать, будто этот анализ лишь формальность: мы толком не продумаем ничего, а потом у нас в ходе работы будет тысяча правок.

Нет, все-таки feature driven development подход предполагает детальный анализ. Это обязательный этап, к которому можно подходить со всей ответственностью

Слайд «создание списка функций»

Вопрос: поясните, пожалуйста, что такое предметная область, функции, и насколько маленькими должны быть эти функции?

Ответ: как было сказано, каждая отдельная область соответствует какому-либо бизнес-процессу. Однако такое пояснение на самом деле не вносит никакой ясности. Что такое бизнес-процесс?

Согласно одному из определений, **бизнес-процесс** — это многократно повторяющаяся, логически связанная последовательность действий, направленная на создание ценности и формирование результата.

Возможно, я сейчас буду говорить не совсем жестко формально правильные вещи, но в общем я точно не солгу. Просто для формирования общего представления о природе вещей.

Итак, предметные области можно понимать как элементы, дочерние каждой смоделированной нами ранее области.

Вернемся к нашему примеру с интернет-магазином. Мы сказали, что у него должен быть личный кабинет. На какие предметные области, или же бизнес-процессы, можно разбить нашу модель личного кабинета?

Ну, конечно, у него по крайней мере должна быть возможность регистрации и авторизации, вероятно, просмотра своего профиля, а может и чужих. Это и есть бизнес-процессы данной моделируемой области.

Что такое функции? Это элементы, дочерние уже не областям, которые мы моделируем, а бизнес-процессам

Например, регистрация может включать в себя проверку почты пользователя, проверку пароля пользователя, отправку письма для подтверждения регистрации, собственно создание самого личного кабинета. Это и есть функции

Вопрос: почему разработка каждой функции должна занимать не более двух недель?

Ответ: Если разработка функции занимает больше двух недель — это просто служит показателем, что разбиение на функции было проведено не идеально; так как функции — это по своей идее маленькие части функциональности; и если для разработки «маленькой части функциональности» требуется очень большое время — это просто знак, что, вероятно, в качестве «функции» вы обозначили слишком большую задачу, которую на самом деле можно и нужно разбить на несколько подзадач, которые и будут функциями.

Слайд «планирование»

Вопрос: кто составляет план разработки и назначает программистам ответственность за реализацию функций?

Ответ: это решается совместно, с учетом компетенций разработчиков

Вопрос: стоп, реализация в виде классов? То есть, эта методология предполагает использование объектно-ориентированного подхода?

Ответ: абсолютно; данная методология построена на объектном моделировании. В ней предполагается объектно-ориентированный подход

Слайд «проектирование функций»

Вопрос: что такое диаграммы последовательности?

Ответ: это очень хороший вопрос; диаграммы последовательности — вещь, не уникальная для нашей методологии, а повсеместно встречающаяся, когда речь идет об объектном моделировании

Согласно определению, это диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл объекта и взаимодействие актеров (то есть, действующих лиц) информационной системы в рамках определенного сценария использования (называемого прецедентом)

Например, сценарий схемы обслуживания в ресторане:

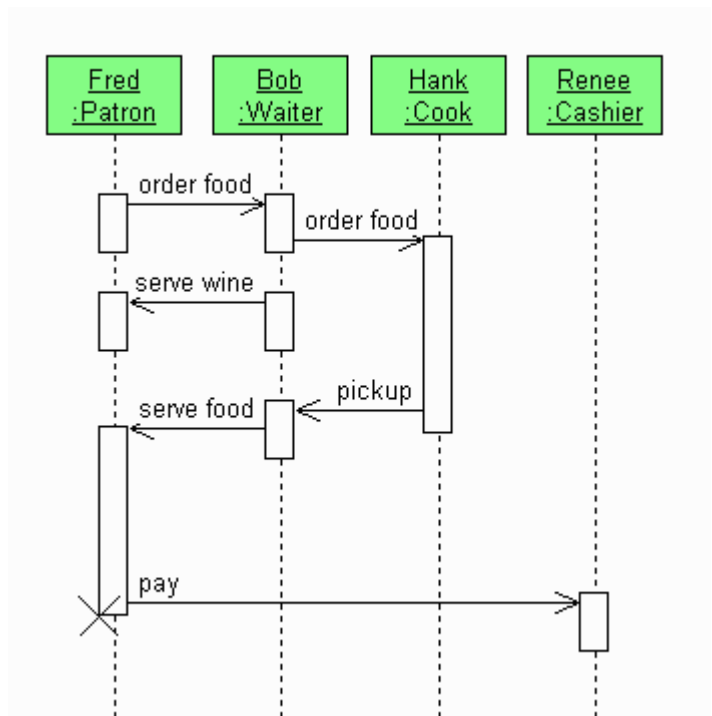
В нем в качестве актеров (действующих лиц) выступает собственно клиент, официант, повар и кассир

И мы видим, как на нем изображено взаимодействие действующих лиц: клиент не заказывает еду непосредственно у повара, но делает это через официанта

Пока еда готовится, официант уже может подавать вино

Когда еда же готова, он забирает ее у повара и подает клиенту

В конце клиент производит оплату кассиру. И это является концом моделируемого сценария.



Вопрос: зачем ведущий программист вмешивается в этот этап? Почему не сами разработчики соответствующего класса одни разрабатывают диаграммы?

Ответ: такое вмешательство, на самом деле, служит лишь одной цели — минимизации ошибок в конечном продукте.

Ведущий программист на этом этапе еще раз проверяет: полна ли модель? Не забыли ли мы что в ней?

Чтобы минимизировать шанс столкнуться с ошибкой на выходе

Вопрос: что такое «болванки» классов и методов?

Ответ: это буквально классы и методы, определенные в коде; но без реализации

Слайд «реализация»

Вопрос: что такое юнит-тестирование?

Ответ: юнит-тестирование, также называемое модульным или блочным тестирование — процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы

То есть, на этапе юнит-тестирования мы проверяем каждые компоненты отдельно: корректно ли работает авторизация, корректно ли работает регистрация; различные другие компоненты

Вопрос: что такое код-ревью?

Ответ:

Рецензирование кода, обзор кода, ревизия кода ([англ. code review](#)) или **инспекция кода** ([англ. code inspection](#)) — систематическая проверка [исходного кода программы](#) с целью обнаружения и исправления [ошибок](#), которые остались незамеченными в начальной фазе разработки. Целью обзора является улучшение качества программного продукта и совершенствование навыков [разработчика](#).

Слайд «достоинства и недостатки»

Вопрос: почему, все-таки, feature-driven development плохо подходит для небольших проектов?

Ответ: хоть это и легковесная методология, но все-таки в ней отдельно и довольно скрупулезно рассматривается предварительное проектирование; проектирование играет в данной методологии большую роль: проводится анализ, разбивается на предметные области.. для всего-всего составляются диаграммы

Это все вместе составляет большой объем работы, который несомненно полезен, если мы разрабатываем большой проект, ошибки на выходе в котором могут обернуться очень большими убытками

Но для небольших проектов скрупулезное проектирование, скорее, будет избыточно

К тому же, она предполагает составление отчетов о проделанной работе на каждом этапе; что может быть очень полезно, когда над нами стоит дядька-заказчик, и всегда хочет знать, за что он платит нам деньги, но не очень актуально, когда это, например, какой-либо персональный или учебный проект