Программная инженерия

Создание человеко-программно-аппаратных систем

Лекция 9

Родионов Николай Евдокимович
nerodionov@gmail.com
моб.тел. 224219

Importance of modeling in software engineering (4)

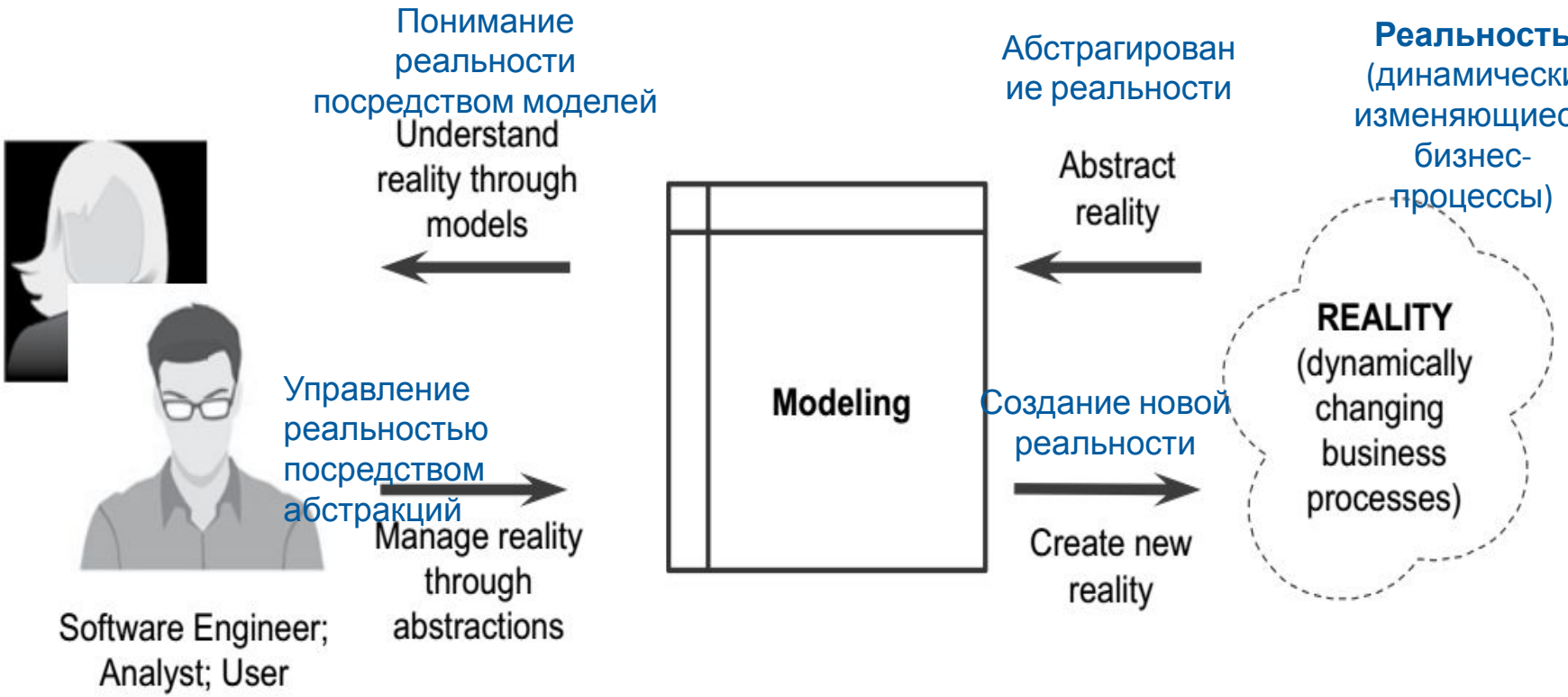


Figure 1.2 Importance of modeling in software engineering: Understand and create reality.

Назначение моделирования в SE: **понимание существующей реальности и создание новой**

Bhuvan Unhelkar Software Engineering with UML CRC Press 2018

The Three Modeling Spaces in Software Engineering

Создание модели для понимания проблемы (анализ)

Разработка модели решения (проект и код)

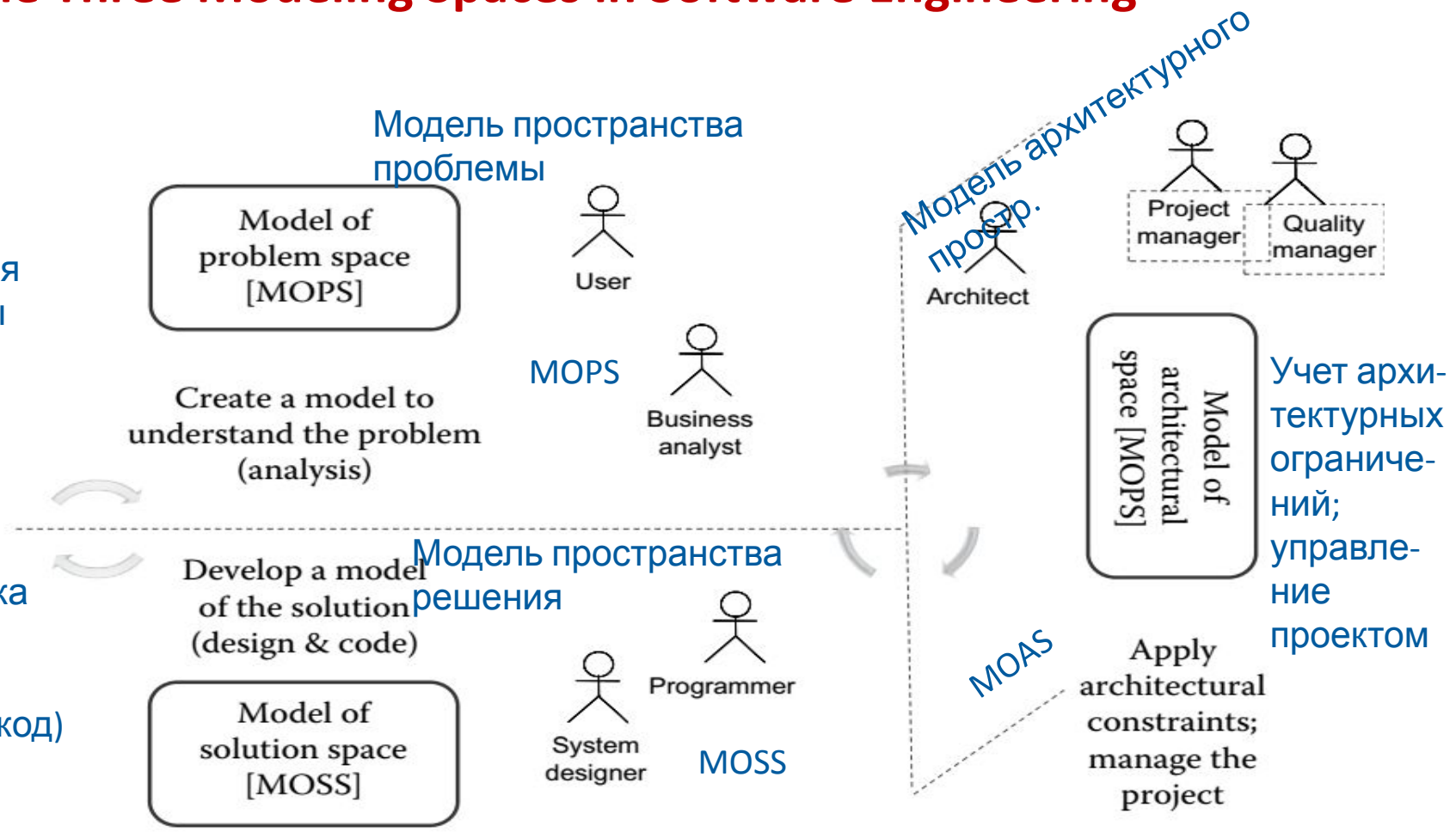
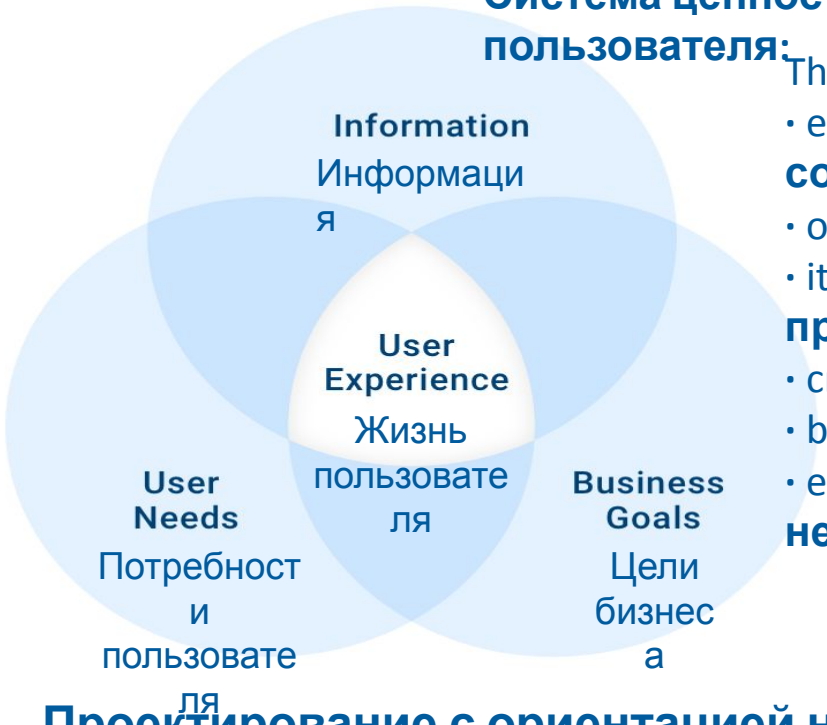


Figure 3.3 Software engineering uses three main modeling spaces and corresponding roles: model of problem space (in identifying the requirements and analyzing them), model of solution space (in creating the solution design), and model of architectural space (in applying constraints).

Bhuvan Unhelkar Software Engineering with UML CRC Press 2018

User-Centered Design: Process and Benefits



Система ценностей проектирования с ориентацией на пользователя:

The value system of user-centered design contains:

- empathy, **эмпатия, умение сопереживать**
- optimism, **оптимизм**
- iteration, **итерационное проектирование**
- creative confidence, **созидательная уверенность**
- belief in making, **вера в успех**
- embracing ambiguity, **допущение неопределенности**

Проектирование с ориентацией на пользователя дает уникальный шанс **разработки вместе с сообществами**. Здесь проектировщики достигают глубокого понимания людей, которым они пытаются служить. Они открывают множество идей и создают новые инновационные продукты, отвечающие на актуальные потребности людей.

User-centered design creates a unique chance to design together with communities. User-centered designers deeply understand the folks they're trying to serve. They create lots of ideas and make innovative new products rooted in people's actual needs.

https://uxplanet.org/user-centered-design-process-and-benefits-fd9e431eb5a9

Using stereotypes for classes in solution space. (2)

Замечание 1:
Атрибуты и операции могут иметь стереотипы
Замечание 2:
Абстрактный класс не может иметь экземпляров. Такой класс обозначается наклонным шрифтом, например: *“Person.”*
Абстракция не является стереотипом.

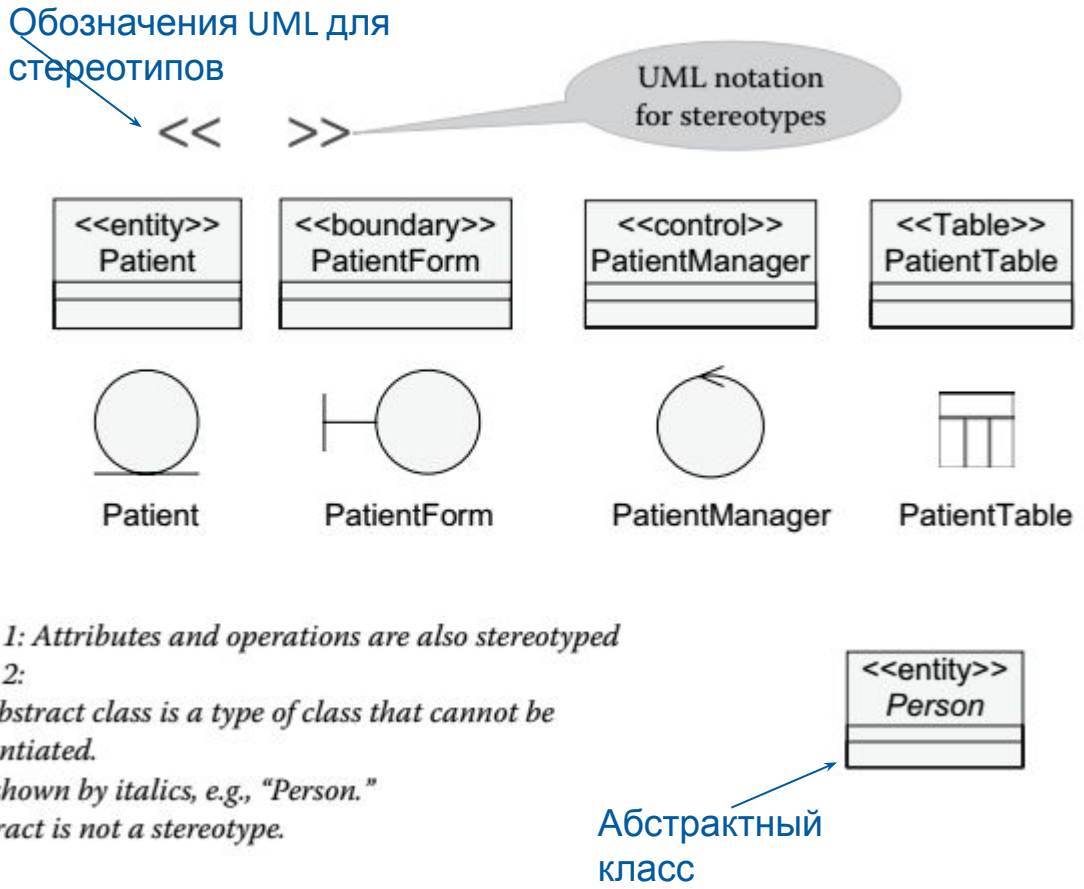


Figure 10.3 Using stereotypes for classes in solution space.

Рисунок 10.3 Использование стереотипов классов в пространстве решений

Feature Driven Development (FDD) in Four Sentences

Разработка на основе функциональности (Feature Driven Development (FDD)) начинается с создания совместно с экспертами предметной области **объектной модели предметной области.**

Используя **информацию** полученную в этом **моделировании**, а также результаты иной **деятельности по исследованию требований**, разработчики создают **список функций (features list).**

Затем создается **ориентировочный план и распределяются обязанности**.

Теперь мы готовы к **реализации групп функций (groups of features)** во время **итераций**, длящихся для всех групп **не более чем две недели (а часто и значительно короче).**

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing
Prentice Hall PTR 2002

FDD in a Nutshell (1)

FDD состоит **из пяти процессов** (рис. 4-2).

В **первом** процессе **эксперты** предметной области и **группы разработки** работают вместе под руководством **опытного разработчика моделей объектов (главного архитектора).**

Эксперты предметной области представляют **начальный общий обзор** области действия системы и ее **контекста**.

Затем **участники** (пользователи – Н.Р.) **предметной области** представляют более **подробные описания каждой ее части.**

После этого **эксперты** и **разработчики** работают в **небольших группах** для создания объектных моделей для **каждой части предметной области.**

Каждая **небольшая группа** представляет свои результаты **команде**, и команда **сравнивает и обсуждает различные модели**, наконец, выбирая наиболее **подходящую модель для каждой части.**

При необходимости создается и корректируется общая модель предметной области.

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing
Prentice Hall PTR 2002

FDD in a Nutshell (2)

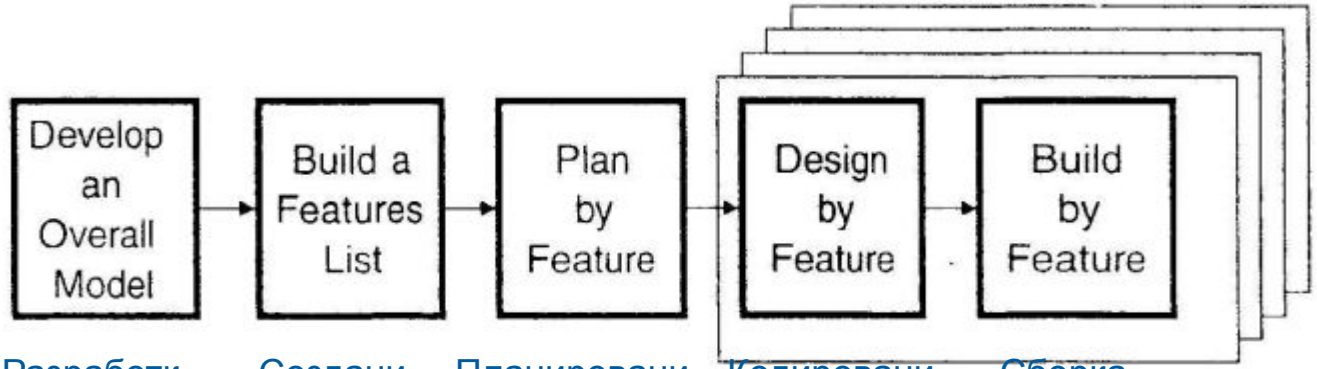


Figure 4-2 The five processes of FDD
Рисунок 4-2 Пять процессов FDD

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing
Prentice Hall PTR 2002

FDD in a Nutshell (3)

Основываясь на **знаниях,** собранных во время **первоначального моделирования**, команда составляет **максимально исчерпывающий список функций** (рис. 4-3).

Функция (feature) определяется как **небольшая, ценимая клиентом (полезная клиенту) (client-valued function) функция**, выраженная в форме: **<действие> <результат> <объект>** (например, «подсчитать общую сумму продажи») ("calculate the total of a sale“).

Существующие **документы требований**, такие как **варианты использования** (прецеденты) или **функциональные** спецификации, также **используются** в качестве **входных данных.**

Предметная область (domain) просматривается с использованием **разбивки**, аналогичной той, которая **используется экспертами во время моделирования.**

Внутри этих *частей предметной области*, также называемых **основными наборами функций (major feature sets)**, функции сгруппированы в **наборы функций**.

Набор функций обычно отражает **конкретную бизнес-деятельность**.

При первоначальной **сборке наборов функций** пользователи и спонсоры системы проверяют **список функций на предмет достоверности и полноты**.

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing
Prentice Hall PTR 2002

FDD in a Nutshell (4)

Третий процесс – упорядочение **наборов функций** или **основных наборов функций** (в зависимости от размера системы) **в общем плане разработки** и **назначение старших программистов** каждому набору.

Наборы функций должны быть упорядочены в соответствии с приоритетом и взаимозависимостями.

Кроме того, **классы**, определенные в **процессе моделирования**, назначаются **отдельным разработчикам**; владелец класса **несет ответственность за его развитие.**

В крупных проектах можно установить **временные рамки основных этапов**, которые ограничивают **время завершения реализации ряда наборов функций.**

Например, **временной интервал в три месяца** может быть установлен для **завершения нескольких первых наборов функций** и формальной **демонстрации результатов высшему руководству.**

Группа также может **использовать окончание временного интервала** для формального **пересмотра плана и получения одобрения руководства на его изменение**.

Для более коротких проектов обычно нет необходимости включать этот этап в план.

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing
Prentice Hall PTR 2002

FDD in a Nutshell (5)

Процессы 4 и 5 - это **сердце (engine room)** разработки**.**

Главный программист выбирает **небольшую группу функций** для **разработки** в течение следующих **нескольких дней (не дольше двух недель)**.

Он **определяет классы, которые могут быть задействованы**, и соответствующие **владельцы классов** образуют собой **группу исполнителей** для этой итерации.

Эта команда разработчиков создает **подробные диаграммы последовательностей** для функций и пишет **вводные части (prologues)** классов и методов.

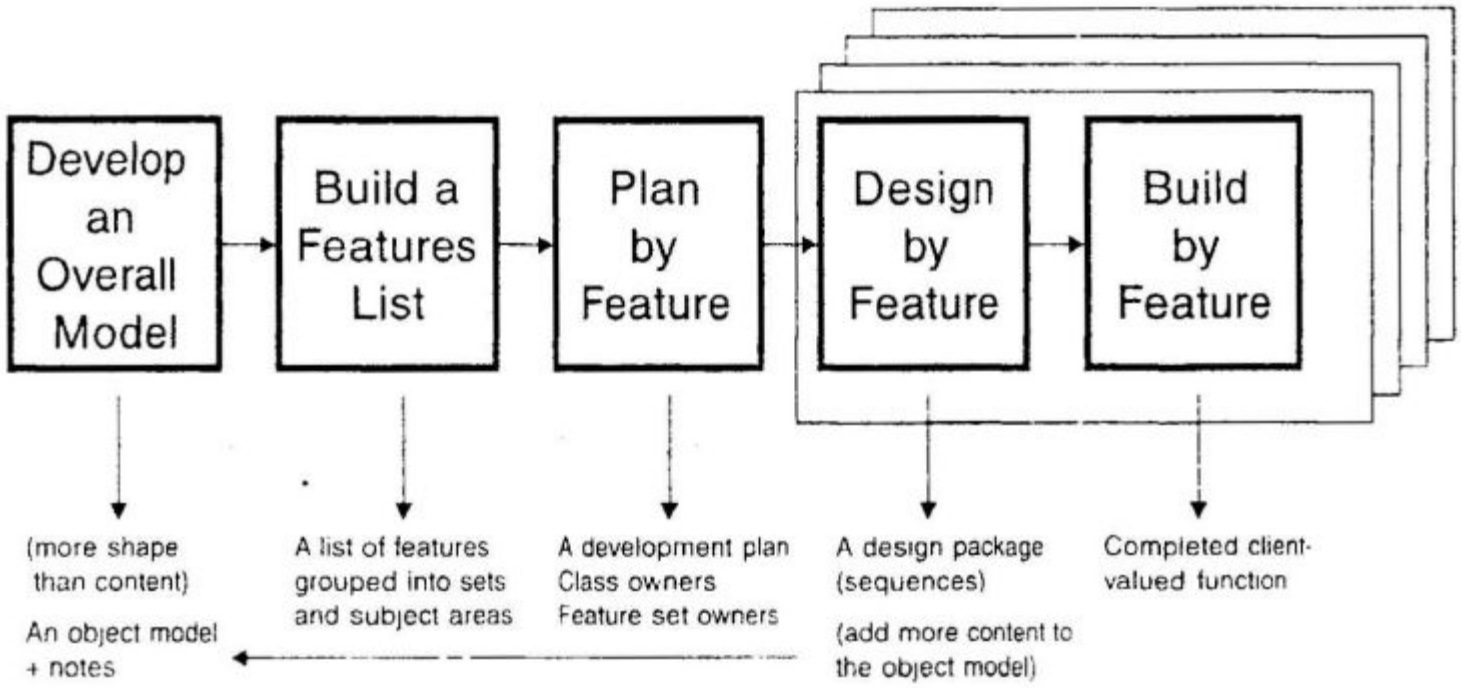
Затем группа проводит проверку текущего состояния разработки.

После успешной проверки **владельцы классов** добавляют **фактический код** для своих **классов**, проводят **модульное тестирование**, объединяют **код** и проводят тестирование **объединенного кода.**

Если **главный программист** удовлетворен **результатами тестирования**, завершенные **функции** перемещаются в **основную сборку**, а **процессы** 4 и 5 повторяются со **следующей группой функций.**

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing
Prentice Hall PTR 2002

FDD in a Nutshell (6)



(упорядочение информации) Объектная модель (ОМ) и заметки

Список функций, группированных по наборам и предметным областям

План разработки
Владельцы классов
Владельцы наборов функций

Проектная документация (последователь-сти)
добавление информации в ОМ

Созданный нужный клиенту функционал

Объектная модель - Object model - A hierarchical set of objects and its members - methods, properties, and events - that a particular component provides - Microsoft

Figure 4-3 The five processes of FDD with their outputs

Рисунок 4-3 Пять процессов FDD и их выходы

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing
Prentice Hall PTR 2002

FDD in Detail (1)

Вы когда-нибудь пытались заставить своих разработчиков следовать руководству по процессу на сто страниц?

Питер Коуд рассказывает историю о том, как он написал **многостраничное руководства по процессам** , и при этом обнаружил, что разработчики читали **только четыре страницы резюме.**

Каждый из пяти процессов в FDD описан на **одном или двух листах формата A4**

Это **все, что нужно разработчикам**, и, скорее всего, все, что **вы сможете заставить их прочитать**, в любом случае.

На самом деле **требуются дополнительные усилия**, чтобы писать **с простотой, ясностью и краткостью** (если эта книга показалась вам слишком длинной, - это потому, что **у нас не хватило времени, чтобы сделать ее короче**).

Наличие в команде **кого-то**, кто читал это **руководство** раньше и **может отвечать на вопросы о процессах по мере их возникновения,** - это подход, который с гораздо **большей вероятностью приведет к успеху**, чем указание разработчикам найти ответы в **стостраничном документе.**

После того, как **менеджер разработки** и **главные программисты** «поймут это», они смогут предоставить всю **повседневную помощь**, необходимую для **использования процессов** тем, кто с ними работает.

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing
Prentice Hall PTR 2002

FDD in Detail (2) The FDD Process Template (1)

A. Rajashima представил **шаблон Entry-Tasks-Verification-Exit (ETVX)**.

Используя этот шаблон, **каждый процесс FDD** описывается в **четырех разделах** (Таблица 4-1).

Section	Description
Entry	A brief overview of the process and a set of criteria that must be met before we canstart the process
Tasks	A list of tasks to perform as part of that process
Verification	The means by which it is verified that the process has been completed satisfactorily. Is the result good enough?
Exit	A list of the outputs from the process

Table 4-1 The ETVX Template

Таблица 4-1 Шаблон ETVX

Вход критериев, выполняться	**Краткий обзор процессов** и **множество** которые должны **присутствовать и перед стартом** процесса FDD
Задачи	**Список задач**, выполнение которых является частью процесса FDD
Верификация	**Перечень средств**, **удостоверяющих** факт удовлетворительного завершения процесса FDD
Выход	**Список результатов** процесса FDD

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing Prentice Hall PTR 2002

FDD in Detail (3) The FDD Process Template (2)

Задачи внутри каждого процесса разделены на **заголовок** и краткое **текстовое описание** (рис. 4-4).

Строка заголовка содержит **название задачи**, **роль, отвечающая за эту задачу**, и **индикатор**, указывающий, является ли задача **обязательной или необязательной.**

…**Общий результат** такого подхода - **простая и понятная структура**, которая кратко сообщает, что **разные люди должны делать в каждом виде деятельности.**

(Иногда организации в таких отраслях, как **фармацевтика, здравоохранение или государственные подрядчики,** не могут согласиться с процессом, описанным на пяти листах бумаги, - **требуется больше формальностей по юридическим или политическим причинам**).

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing
Prentice Hall PTR 2002

FDD in Detail (4) The FDD Process Template (3)

Задачи

Создание группы моделирования/ Менеджер проекта/Обязательная

Группа моделирования состоит из постоянных членов, представляющих предметную область и ее части, в частности, экспертов в предметной области и главных программистов. Реализуется ротация других сотрудников проекта через сеансы моделирования, чтобы у всех была возможность принять участие и увидеть процесс моделирования в действии.

Обзор части предметной области/ Группа моделирования/Обязательная

Эксперт части предметной области дает обзор этой части, которая будет моделироваться. Это должно включать информацию, относящуюся к этой части предметной области, но не обязательно используемую в ее реализации.

Figure 4-4 Task descriptions in FDD
Рисунок 4-4 Описание задач в FDD

FDD in Detail (5) ETVX, Use Cases, and Operations (1)

Шаблон ETVX покажется знакомым тем людям, которые используют **формальный шаблон для написания содержимого прецедентов.**

Оба эти метода **описывают то, что что-то делает (алгоритм),** учитывая **набор предположений об условиях в начале (предварительные условия)** (preconditions).

Оба описывают результаты в конце этого действия (post conditions). …

Конечно, многие **шаблоны прецедентов** предназначены для **более сложных ситуаций**, но **общий шаблон для описания «поведения» существует.**

Здесь **поведение** определяется без указания того, как оно **реализовано**; для описания реализации необходимы **описание алгоритма** или фактический **исходный код**.

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing
Prentice Hall PTR 2002

FDD in Detail (6) ETVX, Use Cases, and Operations (2)

Идея предусловий и постусловий кажется широко распространенной при **описании поведения всех видов систем.**

В разделе верификации шаблона ETVX есть кое-что еще.

В нем **описывается, как проверить качество результата**, чтобы убедиться, что он «достаточно хороший».

Прецеденты описывают действия детерминированной компьютерной системы.

ETVX описывает действия проекта разработки программного обеспечения - человеческой системы.

Человеческая система - люди , которые выносят **оценочные суждения** о **качестве разработки, фрагменте кода, списке требований и плане проекта.**

Компьютер, безусловно, можно использовать **для принятия этих оценочных суждений**.

Например, **набор автоматизированных тестовых проверок и сценарии метрик** можно запускать в отношении **фрагмента кода**, чтобы **выявить потенциальные проблемы**.

Результаты проверок могут использоваться для **оценки качества кода.**

Однако компьютеру обычно не доверяют выносить оценочные суждения.

FDD in Detail (7) Process 1: Develop an Overall Model (1)

Process 1 - **первоначальная деятельность** в рамках всего **проекта**, при которой **специалисты предметной области и разработчики** работают вместе **под руководством опытного разработчика моделей в роли главного архитектора.**

Специалисты предметной области выполняют **анализ всей системы и ее контекста**.

Затем они выполняют подробный **анализ каждой части предметной области**, которая должна быть смоделирована.

После этого **формируются небольшие группы, состоящие из специалистов и разработчиков**.

Каждая **малая группа** составляет свою **собственную модель части предметной области** и представляет свои **результаты для коллегиального обзора и обсуждения**.

Одна из **предложенных моделей или их обобщение** выбирается **консенсусом** и становится **моделью для этой части предметной области.**

Модель части области объединяется с **объектной моделью (object model)** с необходимой корректировкой общей модели.

Затем **объектная модель** итеративно **обновляется** содержимым в **процессе 4 «Кодирование функций».**

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing
Prentice Hall PTR 2002

FDD in Detail (8) Process 1: Develop an Overall Model (2)

Вход Критерии

Для проекта выбраны эксперты предметной области, главные программисты и главный архитектор

Задачи

Создание группы моделирования/ Менеджер проекта/Обязательная

Группа моделирования состоит из постоянных членов, представляющих предметную область и частей области разработки, в частности, экспертов в предметной области и главных программистов.

Реализуется ротация других сотрудников проекта через сеансы моделирования, чтобы у всех была возможность принять участие и увидеть процесс в действии.

Обзор части предметной области/ Группа моделирования/Обязательная

Эксперт части предметной области дает обзор этой части, которая будет моделироваться.

Это должно включать информацию, относящуюся к этой части предметной области, но не обязательно используемую в ее реализации.

FDD in Detail (9) Process 1: Develop an Overall Model (3)

Изучение документов/Группа моделирования/Необязательно

Группа изучает доступные справочные документы или документы требований, такие как объектные модели, функциональные требования (в традиционном формате или в формате прецедентов) и руководства пользователя

Разработка моделей малыми группами./Малая группа моделирования/Обязательно

Формируются группы не более трех человек, каждая группа составляет модель части предметной области.

Главный архитектор может предложить модель «соломенного человечка», чтобы облегчить продвижение команд.

Иногда группы могут также набросать одну или несколько неформальных диаграмм последовательности для проверки формы модели.

Модель «соломенного человечка The Straw Man Proposal is … method for problem solving… allows for a brainstorm-like approach to discussing any deficiencies that may lead to problems https://www.toolshero.com/problem-solving/straw-man-proposal/

Модель «соломенного человечка The Straw Man Proposal is … method for problem solving… allows for a brainstorm-like approach to discussing any deficiencies that may lead to problems https://www.toolshero.com/problem-solving/straw-man-proposal/

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing Prentice Hall PTR 2002

FDD in Detail (10) Process 1: Develop an Overall Model (4)

Разработка групповой модели/ Группа моделирования/Обязательно

Член каждой малой группы представляет модель, предложенную этой группой для части предметной области.

Главный архитектор может также предложить дополнительные варианты общей объектной модели.

Группа моделирования выбирает одну из предложенных моделей или составляет модель, объединяя идеи из предложенных моделей.

Уточнение общей объектной модели/ Главный архитектор, Группа моделирования/Обязательно Время от времени группа обновляет общую объектную модель с помощью новых вариантов модели, созданных итерациями предыдущих двух задач.

Написание заметок о модели/ Главный архитектор, главные программисты/Обязательно

Пишутся примечания к подробным или сложным вариантам моделей и значительным альтернативным моделям для дальнейшего использования в проекте.

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing
Prentice Hall PTR 2002

FDD in Detail (11) Process 1: Develop an Overall Model (5)

Верификац
ия
Внутренняя и внешняя оценка/Группа моделирования, Бизнес/Обязательно

Специалисты предметной области, активно участвующие в процессе, проводят внутреннюю оценку (самооценку).

Внешняя оценка выполняется по мере необходимости путем обращения к бизнесу (пользователям) для утверждения или разъяснения вопросов, влияющих на модель.

Выход Критерии

Для выхода из процесса группа моделирования должна создать объектную модель, удовлетворяющую главного архитектора.

Объектная модель состоит из

- Диаграмм классов, ориентированных на экземпляр модели, классы в объектной модели, как они связаны друг с другом и с какими ограничениями, плюс любые идентифицированные операции и атрибуты
- Диаграмм последовательности, если таковые имеются.
- Примечания, фиксирующие, почему была выбран конкретный экземпляр модели и/или какие альтернативы были рассмотрены

FDD in Detail (12) Process 2: Build a Features List (1)

Первоначальная деятельность в рамках проекта для **определения всех функций (features)** , необходимых **для поддержки требований**.

Формируется **Команда** (обычно состоящая **только из главных программистов** из процесса 1), для **функциональной декомпозиции** предметной области**.**

На основе **разделения предметной области экспертами в процессе 1**, команда разбивает область на **несколько областей (основных наборов функций)**, которые далее разбиваются на **ряд действий (наборов функций).**

Каждый шаг в рамках действия идентифицируется как функция.

Результатом является список функций с иерархической организацией.

Вход Критерии

Группа моделирования успешно завершила процесс 1 FDD «Разработка общей модели».

Задачи

Создание группы, формирующей список функций/ Менеджер проекта, менеджер по разработке/Обязательно

Группа включает в себя главных программистов из команды моделирования процесса 1.

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing
Prentice Hall PTR 2002

FDD in Detail (13) Process 2: Build a Features List (2)

Создание списка функций/ Группа формирования списка/Обязательно

Используя **знания**, полученные в **процессе I**, **группа формирования списка** определяет **функции (features)** .

Группа также может использовать любые существующие **справочные документы** или **документы требований**, такие как **объектные модели**, **функциональные требования** (традиционный формат или формат прецедента), **руководства пользователя** с указанием исходного документа.

Эта задача представляет собой **простую функциональную декомпозицию**, начиная с **рассмотрения** разделения **предметной области на части**, произведенного в процессе 1.

Каждая **часть предметной области** представляет собой **основной набор функций (major feature sets)**, которые, в свою очередь включают в себя действия (наборы функций) (activities (feature sets)) , при этом **каждая из этих функций – это шаг в подобном действии**.

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing
Prentice Hall PTR 2002

FDD in Detail (14) Process 2: Build a Features List (3)

Features - это **детализированные функции**, выраженные в **терминах клиентских ценностей**, с использованием **шаблона : <действие> <результат> <объект>**

Например, «подсчитать общую сумму продажи» и «рассчитать общее количество, данного товара, проданного розничной точкой».

Реализация такой **детализированной функции** должна занимать **не более двух недель**, но функция **не должна быть настолько детализированной**, чтобы быть на **уровне простых операций get и set.**

Две недели - это верхний предел; большинство функций занимают меньше времени.

Если **шаг представляется большим, чем две недели**, команда разбивает его на **более мелкие шаги**, которые и будут являться **детализованными функциями (features).**

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing
Prentice Hall PTR 2002

FDD in Detail (15) Process 2: Build a Features List (4)

Верификация

Внешнее и внутреннее оценивание/ Группа формирования списка, Бизнес/Обязательно

Проводится **совместно** членами **группы бизнес - моделирования** и представителями **бизнеса**.

Выход Критерии

Для выхода из процесса группа по составлению списка функций должна составить **список функций**, удовлетворяющий **менеджера проекта и менеджера разработки.**

Список функций состоит из

- **Списка основных наборов функций** для каждой части предметной области (областей)
- **Набора функций (действий)** внутри каждого основного набора функций
- **Списка функций (шагов)** внутри каждого действия.

FDD in Detail (16) Process 3: Plan by Feature (1)

Первоначальная деятельность в рамках проекта по составлению плана развития.

Менеджер проекта, менеджер разработки и главные программисты планируют **порядок реализации функций** на основе **зависимостей функций**, **нагрузки на команду** разработчиков и **сложности функций**, подлежащих реализации.

Основные **задачи в этом процессе не следуют в строгой последовательности**, как и многие действия по планированию, они **рассматриваются** вместе **с уточнениями**, внесенными в одну или несколько задач, **с последующим повторным обсуждением**.

Типичный сценарий состоит в определении **последовательности разработки**, затем происходит **назначение наборов функций главным программистам** и обсуждается **назначение ключевых классов программистам** (помня, что главный программист, также является разработчиком).

После этого происходит распределение оставшихся классов программистам.

FDD in Detail (17) Process 3: Plan by Feature (2)

Вход Критерии

Команда по составлению **списка функций** успешно **завершила процесс 2** FDD , **построен список функций**

Задачи

Формирование группы планирования /Менеджер проекта/Обязательно

Группа планирования состоит из менеджера проекта, менеджера по развитию и главных программистов.

Определение последовательности разработки/ Группа планирования/Обязательно

Группа планирования назначает дату (только месяц и год) для завершения каждого набора функций.

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing
Prentice Hall PTR 2002

FDD in Detail (18) Process 3: Plan by Feature (3)

Идентификация **набора функций** и **даты завершения** (и, таким образом, последовательность разработки) основывается на

- **Зависимости между функциями** с точки зрения задействованных классов
- **Балансировке нагрузки** между владельцами классов
- **Сложности** реализуемых функций
- **Присвоения высокого приоритета** наборам функций с высоким риском или сложным
- **Учет любых внешних** (видимых) **вех**, такие как **бета-версии**, **предварительные просмотры, контрольные точки обратной связи и "целые продукты",** которые соответствуют таким этапам

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing
Prentice Hall PTR 2002

FDD in Detail (19) Process 3: Plan by Feature (4)

Назначение наборов функций старшим программистам/Группа планирования/Обязательно

Группа планирования назначает **главных программистов** владельцами **наборов функций**. Назначение основано на:

- **Последовательности** разработки
- **Зависимостях** между функциями с точки зрения участвующих классов
- **Распределении нагрузки** между владельцами классов (помня, что главные программисты также являются владельцами классов).
- **Сложности** реализуемых функций

Назначение классов разработчикам/Группа планирования/Обязательно

Группа планирования назначает разработчиков владельцами классов.
Разработчики владеют несколькими классами.
Присвоение классов разработчикам основано на:

- **Балансировке** нагрузки между разработчиками;
- **Сложности** классов;
- **Ожидаемой интенсивности** использования классов
- **Последовательности** разработки

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing
Prentice Hall PTR 2002

FDD in Detail (20) Process 3: Plan by Feature (5)

Верификация

Самооценка/Группа планирования/Обязательно

Планирование - это групповая деятельность, поэтому **самооценка** достигается за счет активного **участия менеджера проекта и менеджера разработки с главными программистами,** которые используют **знания, полученные в процессе I**, для принятия **более обоснованных решений.**

Выход Критерии

Чтобы выйти из процесса 3, группа планирования должна составить **план развития**, удовлетворяющий менеджера проекта и менеджера разработки.

План разработки состоит из:

- **Наборов функций** с датами завершения (месяц и год)
- **Основных наборов функций** с датами завершения (месяц и год), полученных от последней даты завершения соответствующих наборов функций
- **Списка главных программистов**, назначенных для наборов функций
- **Списка классов и разработчиков**, которым они принадлежат (список владельцев классов)

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing
Prentice Hall PTR 2002

FDD in Detail (21) Process 4: Design by Feature (1)

Действие для каждой функции (per-feature activity) по созданию пакета проектирования функций.

Главный программист выбирает **функции для разработки** из своего «почтового ящика» назначенных функций (на практике часто бывает так, что **главный программист одновременно планирует разработку нескольких небольших групп функций**).

Он может **выбрать несколько функций**, которые **будут использоваться одни и те же классы** (то есть **разработчики**).

Такая **группа функций** образует **рабочий пакет главного программиста**.

Затем **главный программист** формирует **группу реализации функций**, определяя **владельцев классов (разработчиков),** которые могут быть вовлечены в разработку выбранной функции.

Эта **группа** создает **подробную (-ые) схему (-ы) последовательности** для выбранных функций.

Затем **главный программист** уточняет **объектную модель** (на основе содержания **диаграмм последовательности**).

Разработчики пишут прологи классов и методов.

Проводится проверка проекта.

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing
Prentice Hall PTR 2002

FDD in Detail (22) Process 4: Design by Feature (2)

Вход Критерии
Группа планирования успешно завершила процесс 3 FDD.

Задачи

Создание функциональной группы/ Главный программист/Обязательно

Главный программист определяет **классы**, которые, вероятно, будут задействованы в разработке **группы функций.**

Из **списка владельцев классов** главный программист определяет **разработчиков, необходимых для формирования функциональной группы.**

В рамках этого шага **главный программист** запускает **новый пакет разработки** для функции (ей) как **части рабочего пакета**.

Консультации по поводу предметной области/Эксперт предметной области/Необязательно

По запросу главного программиста **эксперт предметной области** знакомит **функциональную группу** с **деталями** алгоритмов, правил, формул и элементов данных, **необходимыми для разработки функции.**

Это необязательная задача, зависящая от сложность функций и/или их взаимодействия.

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing
Prentice Hall PTR 2002

FDD in Detail (23) Process 4: Design by Feature (3)

Изучение документации/Функциональная группа/Необязательно

Функциональная группа изучает **документы, указанные в списке функций** для разрабатываемых функций, а также любые **другие относящиеся к делу документы**, включая любые **подтверждающие записки, дизайн экранов и спецификации интерфейса внешней системы**.

Это **необязательная задача**, зависящая от **сложности функций и/или их взаимодействия и наличия таких документов.**

Разработка диаграмм (-ы) последовательности/Функциональная группа/Обязательно

Функциональная группа разрабатывает **подробные диаграммы последовательности**, необходимые для каждой разрабатываемой функции.

Группа подробно описывает и фиксирует любые **альтернативные разработки**, **проектные решения, предположения, пояснения требований** в **примечаниях** раздела альтернатив дизайна или в **примечаниях** к проектной документации.

FDD in Detail (24) Process 4: Design by Feature (4)

Уточнение объектной модели/ Главный программист/Обязательно

Главный программист создает **область** функциональной группы (**feature team area**) для функции (функций)

Эта **область** является либо **каталогом на файловом сервере**, либо **каталогом на его персональном компьютере** (при необходимости поддерживается главным программистом) или **использует поддержку рабочей области в системе управления версиями проекта**. **Функциональная группа** использует **область** функциональной группы, чтобы делиться **незавершенной работой** и **сделать ее видимой для функциональной группы**, но **не для остальной части проекта.**

Главный программист уточняет модель, чтобы добавить **дополнительные классы**, **операции и атрибуты** и/или вносить **изменения в существующие классы, операции или атрибуты** на основе **диаграмм последовательности**, определенных для функций.

Связанные **исходные файлы на языке реализации** обновляются (вручную или автоматически с помощью какого-либо инструмента) в **области** функциональной группы.

Главный программист создает **диаграммы моделей в формате, доступном для публикации**.

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing
Prentice Hall PTR 2002

FDD in Detail (25) Process 4: Design by Feature (5)

Написание прологов классов и методов/Функциональная группа/Обязательно

Используя обновленные **исходные файлы языка реализации** из задачи «Уточнение объектной модели» в **области** функциональной группы, каждый **владелец класса** записывает **прологи класса и метода для каждого элемента, определенного функцией и диаграммой последовательности**

Сюда входят **типы параметров, возвращаемые типы, исключения и сообщения**.

Проверка разработки/Функциональная группа/Обязательно

Функциональная группа проводит проверку разработки **до или после выполнения задачи модульного тестирования.**

В этой проверке могут участвовать другие участники проекта.

Главный программист принимает решение: проводить проверку **силами функциональной группы** или **совместно с другими членами проектной группы**.

При принятии **решения о завершении проверки**, создается **список задач** для каждого **затронутого класса**, и каждый **член команды** добавляет свои **задачи в свой список дел.**

FDD in Detail (26) Process 4: Design by Feature (6)

Верификация

Проверка разработки/Функциональная группа/Обязательно

Проверка разработки- это проверка результатов процесса 4 FDD. Эта задача описана на **предыдущем слайде**.

Выход Критерии

Для выхода из процесса **функциональная группа** должна создать успешно проверенный **проектный пакет**, который включает:

- Сопроводительную записку или документ для рецензентов, который описывает проектный пакет.
- Функциональные требования (если есть) в форме документов и всех связанных подтверждающих записок и сопроводительной документации
- Диаграммы последовательности
- Альтернативы проектирования (если есть)
- Объектную модель с новыми / обновленными классами, методами и атрибутами
- Сгенерированные вашими инструментами выходные данные для прологов классов и методов, созданных или измененных этим проектом
- Записи в списке задач входных данных для элементов действий затронутых классов (для каждого члена группы)

FDD in Detail (27) Process 5: Build by Feature (1)

Реализация каждой функции (per-feature activity) для создания завершенной функции (feature), полезной клиенту.

Работая на основе **пакета проектирования**, созданного в **процессе 4** FDD, **владельцы классов** реализуют **элементы, необходимые для их классов**.

Затем **разработанный код** подвергается **модульному тести**рованию и **проверке кода**, **порядок** которых **определяется главным программистом**.

После успешной проверки кода он перемещается в сборку.

Вход Критерии

Специализированная группа успешно завершила процесс 4 FDD для выбранных функций. Это означает, что проверка проектного пакета из процесса 4 была успешно завершена.

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing
Prentice Hall PTR 2002

FDD in Detail (28) Process 5: Build by Feature (2)

Задачи

Реализация классов и методов/Функциональная группа/Обязательно

Владельцы классов реализуют элементы, необходимые для **удовлетворения требований своих классов** для функций в рабочем пакете.

Это также **включает в себя разработку** любого необходимого **кода модульного тестирования.**

Проверка кода/Функциональная группа/Обязательно

Функциональная группа проводит проверку кода **до или после** выполнения задачи **модульного тестирования**.

Главный программист решает, проводить ли проверку **внутри функциональной группы или с другими членами проектной группы.**

FDD in Detail (29) Process 5: Build by Feature (3)

Модульное тестирование/Функциональная группа /Обязательно

Каждый **владелец класса тестирует свой код**, чтобы убедиться, что все **требования к их классам для функций** в рабочем пакете **выполняются**.

Главный программист определяет нужно ли **модульное тестирование на уровне функциональной группы.**

Другими словами - **какое необходимо тестирование в рамках классов, разработанных для функции (функций).**

Продвижение к сборке кода/Главный программист, Функциональная группа/Обязательно

Классы, реализованные функциональной группой, могут быть **переведены в сборку** только после **успешной проверки кода и модульного тестирования**.

Главный программист является **субъектом интеграции для всех функций** и отвечает за отслеживание **передачи в сборку классов** (либо **путем их личного продвижения**, либо посредством **обратной связи с разработчиками** в функциональной группе)

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing
Prentice Hall PTR 2002

FDD in Detail (30) Process 5: Build by Feature (4)

Верификация

Проверка кода и модульное тестирование/Главный программист, Функциональная группа/Обязательно

Успешная проверка кода плюс успешное завершение модульного теста - это проверка результатов процесса 5**.**

Эти процедуры описаны на предыдущем слайде.

Выход Критерии

Для выхода из процесса **функциональная группа должна завершить** разработку **одной или нескольких features** (функций, полезных для клиентов (client-valued functions)).

Для этого **разработка** должна **удовлетворять требованиям сборки** новых и усовершенствованных классов, поддерживающих эти функции, а эти **классы** должны успешно пройти **процедуры проверки кода и модульного тестирования**.

Summary (1)

FDD решает **ключевую проблему в разработке программного обеспечения** – обеспечивает **правильную** разработку **правильного** программного обеспечения **в срок**.

FDD выполняет **достаточные объемы предварительных работ** (до кодирования – Н.Р.), чтобы обеспечить **устойчивую концептуальную основу для кодирования** - **объектную модель** (структуру) предметной области и **список функций** (требования).

В большой степени **повторяющийся, самоорганизующийся, контролируемый хаос итераций** процесса 4 и процесса 5 обеспечивает **гибкую операционную структ**уру, которая может быстро **адаптироваться к изменениям.**

Этот **сбалансированный подход**, пропагандируемый FDD, позволяет **избежать аналитического паралича**, часто встречающегося в командах, которые следуют **традиционным процессам с длительными фазами анализа** (которые просто **неработоспособны** для **краткосрочных проектов**, когда **бизнес-требования** меняются **ежемесячно**, если не **еженедельно**).

FDD также позволяет избежать большого количества **ненужной доработки кода**, которая почти **гарантирована**, когда **команда сразу погружается в кодирование без какого-либо разумного общего понимания проблемы, которую необходимо решить**.

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing
Prentice Hall PTR 2002

Summary (2)

Как и **другие гибкие методологии**, FDD позволяет командам **быстрее добиваться реальных результатов без ущерба для качества**

Это **итеративная методология**, ориентированная на **людей и результат**.

FDD ломает **традиционные перегородки**, отделяющие **экспертов / аналитиков предметной области и бизнеса** от **разработчиков и инженеров**.

Аналитиков вытаскивают из их абстракций и они непосредственно **участвуют в построении системы** совместно с **разработчиками и пользователями**.

FDD, по сути, **состоит из пяти коротких процессов**, каждый из которых описывается **одной-двумя страницами текста**, в соответствии с простым **шаблоном ETVX**.

Другие действия по разработке (не вошедшие в эти пять процессов – Н.Р.), такие как **определение / сбор начальных требований, управление изменениями и формальное тестирование**, обеспечивают **входные данные** или используют **выходные данные** основных процессов FDD.

A Practical Guide to Feature-Driven Development by Stephen R. Polmer and John M. Felsing
Prentice Hall PTR 2002