

Глава 17

Документ-концепция

Основные положения

- Документ-концепция (Vision document) описывает приложение в целом, включая описания целевых рынков, пользователей системы и функций приложения.
- Документ-концепция определяет на наивысшем уровне абстракции как проблему, так и решение.
- Практически все программные проекты выигрывают от наличия документа-концепции.
- Документ Delta Vision акцентирует внимание на том, что изменилось.

Данная глава посвящена рассмотрению документа-концепции. Наш коллега Филипп Крачтен (Philippe Kruchten) как-то сказал: “Если бы мне разрешили разработать только один документ, модель или другой артефакт для поддержки программного проекта, я бы выбрал краткий, хорошо сформулированный документ-концепцию

Документ-концепция сочетает в себе некоторые основные элементы документа маркетинговых требований и документа требований к продукту. Нам необходимо разработать этот конкретный документ по двум причинам.

1. Каждому проекту нужен документ-концепция.
2. Это поможет нам в демонстрации процесса работы с требованиями, поскольку некоторые ключевые элементы данного процесса будут записаны в этом документе.



Документ-концепция моего проекта

Документ-концепция описывает приложение в общих чертах, а также содержит описания целевых рынков, пользователей системы и функций приложения. Мы неоднократно имели возможность убедиться в его полезности, и у нас стало уже хорошей традицией разрабатывать его при определении любого программного приложения.

Компоненты документа-концепции

Документ-концепция — это важнейший документ программного проекта, который фиксирует потребности пользователя, функции системы и другие общие требования к проекту. Сфера действия документа-концепции распространяется на два верхних уровня пирамиды требований. Таким образом, он описывает на высоком уровне абстракции как проблему, так и решение.



Сфера действия документа-концепции

Документ-концепция программного продукта также служит основой для достижения согласия между тремя основными внутренними сообществами заинтересованных лиц проекта.

1. Отделом маркетинга, который выступает в качестве доверенного лица заказчика и пользователя и отвечает за успех продукта после реализации.
2. Командой проекта, разрабатывающей приложение.
3. Руководством, которое несет ответственность за бизнес-результат **потенциально**.

Документ-концепция является мощным средством, так как представляет все существенные аспекты продукта с различных точек зрения в краткой, абстрактной, доступной и управляемой форме. Документ-концепция крайне важен на ранних фазах проекта, и все затраты, вложенные в процесс получения информации, принесут щедрые плоды на более поздних стадиях.

Поскольку все программные проекты выиграют от наличия документа-концепции, мы собираемся описать его более подробно. Хотя наш пример ориентирован на программный продукт, его достаточно легко модифицировать, чтобы он отражал содержание любого конкретного продукта.

На рис. 17.1 представлена краткая схема документа-концепции, которая (с некоторыми настройками) использовалась для сотен программных продуктов и приложений. Полная версия данного документа приводится в **приложении Б**.

1. Введение

В данном разделе предлагается общий обзор документа-концепции.

1.1. Назначение документа-концепции

В данном документе фиксируются, анализируются и задаются высокоуровневые потребности пользователей и функции продукта.

1.2. Краткое описание продукта

Формулируется цель приложения, версии и новые предоставляемые функции.

1.3. Ссылки

Приводится полный список всех документов, упоминаемых в документе-концепции.

2. Описание пользователя

Кратко представлены общие сведения о пользователях системы.

2.1. Данные о пользователе/рынке

Кратко представлены основные данные о рынке, которые мотивируют решения относительно продукта.

2.2. Типы пользователей

Кратко описываются будущие пользователи системы.

2.3. Среда пользователя

2.4. Основные потребности пользователей

Перечисляются основные проблемы или потребности пользователей.

2.5. Альтернативы и конкуренты

Выявляются все приемлемые (с точки зрения пользователя) альтернативы.

3. Краткое описание продукта

3.1. Общий вид продукта

Предлагается блок-схема продукта или системы и ее интерфейсов с внешней средой.

3.2. Определение позиции продукта на рынке

Предлагается обобщенная краткая характеристика (на самом высоком уровне абстракции) уникальной позиции, которую должен занять продукт на рынке. Мур (Moore) (1991) рекомендует использовать следующую форму.

Для	[целевой клиент]
Который	[формулировка потребности или возможности]
[Название продукта]	является [категория продукта]
Который	[формулировка основных преимуществ, т.е. указание причин, по которым продукт будет покупаться]
В отличие от	[основные конкурирующие альтернативы]
Наш продукт	[формулировка основных отличий]

3.3. Характеристика возможностей

Перечисляются основные возможности и функции, которые будут предоставлены продуктом.

Возможности клиентов	Поддерживающие функции
Возможность 1	Функция
Возможность 2	Функция
Возможность 3	Функция

3.4. Предположения и зависимости

3.5. Затраты и цены

4. Атрибуты функций

Описываются атрибуты функций, которые будут использоваться для оценки, отслеживания, задания приоритетов и управления функциями. Некоторые из них перечислены ниже.

Статус	Предлагаемый, принятый, включенный
Приоритет	Число голосов по результатам накопительного

	голосования или критический, важный, полезный
Трудоемкость	Низкая, средняя, высокая;командо-недели; человекo-месяцы
Риск	Низкий, средний, высокий
Стабильность	Низкая, средняя, высокая
Целевая версия	Номер версии
Предназначен для	Фамилия
Причина	Текстовое поле

5. Функции продукта
В данном разделе перечисляются функции продукта.

5.1. Функция №1

5.2. Функция №2

6. Ключевые прецеденты
Описываются основные прецеденты, которые важны с точки зрения архитектуры или наиболее полезны для того, чтобы помочь читателю понять, как будет использоваться система.

7. Другие требования к продукту

7.1. Применяемые стандарты
Перечисляются все стандарты, которым должен соответствовать продукт.

7.2. Системные требования
Задаются все системные требования, которым должно соответствовать приложение.

7.3. Лицензирование и установка
Описываются все инсталляционные требования, которые оказывают влияние на создание программного кода или вызывают потребность в создании отдельного инсталляционного программного обеспечения.

7.4. Требования к производительности
Этот раздел используется для подробного описания требований к производительности.

8. Требования к документации
Описывается, какую документацию необходимо разработать для успешного развертывания приложения.

8.1. Руководство пользователя
Описывается цель и содержание руководства пользователя.

8.2. Интерактивные подсказки
Требования к интерактивным подсказкам, средствам предупреждения и т.п.

8.3. Руководства по инсталляции, конфигурация и файлы “Read Me”

8.4. Маркировка и упаковка

9. Глоссарий

Рис. 17.1. Схема документа-концепции некоего программного продукта

Итак, документ-концепция кратко описывает все, что вы считаете наиболее важным для продукта или приложения. Он представляет собой достаточно подробное описание на естественном языке, поэтому основным участникам проекта легко с ним работать.

Документ Delta Vision

Разработка документа-концепции и работа с ним, являясь центром приложения действий многих участников (заказчиков, пользователей, представителей руководства проекта и маркетинга), могут играть заметную роль в успехе (или неудаче) программного проекта. Зачастую к разработке и пересмотру этого документа привлекается даже дирекция компании. Ведение документа-концепции является важным профессиональным приемом, который в состоянии значительно повысить общую производительность работы над проектом.

Чтобы это было легче осуществить, нужно сделать документ-концепцию как можно более кратким, сжатым и “по существу”. При создании первой версии документа это не так уж сложно, так как практически все пункты в перечне будут новыми для данного проекта или, по крайней мере, должны быть переформулированы с учетом содержания данного приложения.

Однако неэффективно в последующие версии повторно записывать вошедшие в предыдущие версии функции, а также прочую информацию, которая не претерпела изменений (описание пользователей и обслуживаемых рынков). Для решения данной проблемы мы предлагаем использовать *документ изменений концепции (Delta Vision document)*. Однако перед тем, как заняться его разработкой, рассмотрим развитие документа-концепции на протяжении жизненного цикла нового проекта.

Документ-концепция версии 1.0

Для нового продукта или приложения необходимо разработать и исследовать практически все элементы документа-концепции. Если некий элемент не рассматривается, вы просто удаляете его из оглавления документа и ничего не пишете о нем. Обязательными элементами документа-концепции являются следующие (рис. 17.2).

- Общая информация и введение
- Сведения о пользователях системы и описание обслуживаемых рынков, функций, которые предполагается реализовать в версии 1.0
- Прочие требования (регуляторные и требования среды)
- Будущие функции, которые были выявлены, но не вошли в версию 1.0

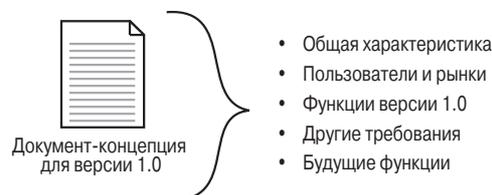


Рис. 17.2. Документ-концепция версии 1.0

Данный документ служит основой версии 1.0 и разработки более подробных программных требований и прецедентов, которые будут более полно описывать **симу**.

Документ-концепция версии 2.0

По мере развития проекта более четко определяются функции; это означает, что они будут более полно описаны в документе-концепции. Кроме того, выявляются новые функции и добавляются в него. Таким образом, документ разрастается, и одновременно возрастает его значение для команды. Приступая к разработке версии 2.0, мы, безусловно, хотели бы сохранить документ, который так хорошо нам служил. В данной ситуации представляется логичным “откопать” будущие функции, которые включены в документ для версии 1.0, но не реализованы, и запланировать их для реализации в версии 2.0. Другими словами, мы хотим найти и “раскрутить” некоторые будущие функции, представляющие интерес для версии 2.0. Можно также запланировать проведение дополнительного совещания, посвященного требованиям, или осуществление любого другого процесса выявления требований для обнаружения новых функций, запланированных для реализации в версии 2.0, а также тех, которые нужно будет внести в документ в качестве новых будущих функций. Некоторые из этих функций, основанные на обратной реакции заказчика, уже будут очевидны, другие возникнут как следствие полученного командой опыта. В любом случае эти вновь обнаруженные функции следует записать в документ-концепцию версии 2.0 как запланированные для реализации в версии 2.0 или будущие функции.



Документ-
концепция v2.0

Может оказаться, что некоторые реализованные в версии 1.0 функции не достигают поставленной цели (возможно, из-за того, что внешняя среда изменилась за время разработки и функция больше не нужна или должна быть заменена новой, либо из-за того, что она просто не нужна клиентам, хотя они предполагали обратное). В любом случае скорее всего обнаружится, что в следующей версии некоторые функции необходимо удалить. Как отразить эти “антитребования”? В данной ситуации нужно просто использовать документ-концепцию для указания того, что определенная функция должна быть удалена из следующей версии.

В процессе работы документ постоянно растет. Это естественно, так как он определяет растущую систему. К сожалению, может случиться, что со временем документ будет все труднее читать и понимать. Почему? Потому, что он теперь гораздо длиннее и содержит много информации, которая не претерпела изменений со времени предыдущей реализации. Например, определение позиции продукта и целевые пользователи, скорее всего, остались неизменными, как и 25–50 функций, реализованных в версии 1.0, которые сохранились в документе-концепции версии 2.0.



Документ
Delta Vision v2.0

Поэтому мы предлагаем вести *документ изменений концепции (Delta Vision document)*. В нем отражается только то, что изменилось, а также любая другая информация, которую следует включить для ясности. Эта информация включается для того, чтобы напомнить команде концепцию проекта или помочь войти в курс дела новым членам команды.

В результате получается документ изменений, в котором *основное внимание уделяется тому, что нового включено в данную версию и что отличает ее от предыдущих версий*. При работе со сложными системами информации полезно применять данный метод, который позволяет *сконцентрировать внимание на том, что изменилось*. Воспользовавшись им, мы получаем модель, представленную на рис. 17.3.

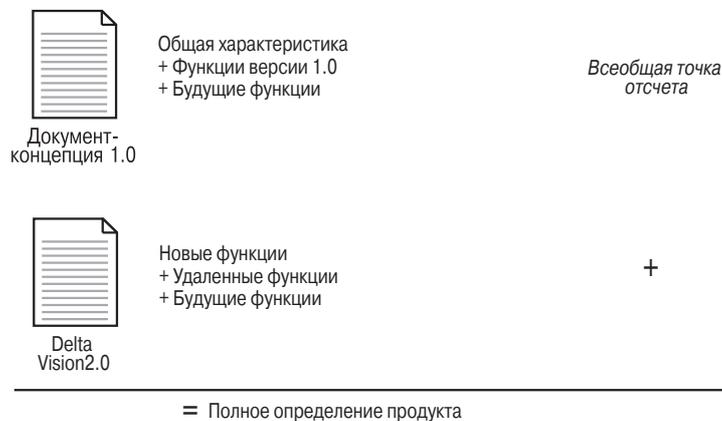


Рис. 17.3. Документ Delta Vision

- Документ-концепция 1.0 является *всеобъемлющей точкой отсчета*; здесь представлено все, что необходимо знать о нашем проекте.
- Delta Vision 2.0 определяет то, что отличает данную версию.
- Объединение этих двух документов задает *полное определение продукта*.

Следует использовать обе версии вместе, если согласно требованиям заказчика или регулирующих инструкций необходимо предоставить полное определение продукта. Их совместное использование, несомненно, полезно для новых членов команды. Однако в этом случае приходится читать о функциях версии 1.0, которых уже нет в версии 2.0, так как они были позднее удалены, и нужно всегда отслеживать эти изменения при воссоздании полного определения.

Если это необходимо, можно достаточно просто соединить содержимое документа-концепции 1.0 и Delta 2.0 в новый документ-концепцию 2.0, который представляет всеобъемлющую и полную картину проекта.

Не существует строгих правил относительно определений этих документов или того, что каждый из них содержит. В других обстоятельствах может оказаться удобным использовать Delta Vision только при относительно небольших модификациях (таких, как версии 1.1 и 1.2) и начинать все сначала и пересматривать определение продукта в целом для каждой крупной реализации (версии 2.0 или 3.0). В любом случае применение документа Delta Vision поможет лучше справиться с процессом управления требованиями, так как позволит команде сконцентрироваться на «том, что действительно важно» на каждом конкретном этапе.

Документ Delta Vision для уже существующей системы

Крайне редко практикуется документирование полных требований крупномасштабной существующей системы.

Одна из сложнейших проблем при управлении требованиями состоит в применении методов управления требованиями к эволюции существующих IS/IT-систем. Крайне редко существуют полные и адекватные спецификации требований для миллионов строк

кода и сотен человеко-лет трудозатрат, отражением которых являются эти системы. Также непрактично останавливаться и повторно документировать прошлое. При этом можно упустить время и не выполнить свою задачу, записывая исторические требования тогда, когда следовало писать код!

Таким образом, если приходится начинать с нуля или с минимальной документации, следует использовать все имеющиеся в вашем распоряжении ресурсы (программный код, спецификации, знания членов команды о предыстории), чтобы прийти к пониманию того, что система делает в настоящий момент. Затем мы рекомендуем применить процесс создания Delta Vision и задать функции и прецеденты, описывающие *изменения*, которые вы собираетесь вносить в существующую систему. Следуя этому процессу, можно сконцентрироваться на том, *что нового в данной реализации и что отличает ее от предыдущих реализаций*; в результате ваши заказчики и команда получат несомненную пользу от хорошо организованного процесса управления требованиями. Кроме того, созданные вами записи требований будут служить документацией для ваших последователей.

Глава 18

Лидер продукта

Основные положения

- Лидер продукта отвечает за концепцию проекта.
- Каждому проекту нужен лидер или небольшая лидирующая группа для защиты интересов продукта.
- При разработке программных продуктов лидером часто является представитель маркетинга.

В главе 1 мы проанализировали проекты, в которых возникли затруднения, и выявили множество разнообразных причин этого, причем управление требованиями оказалось в верхней части списка. В главе 17 мы определили документ-концепцию как ключевой документ сложного жизненного цикла программы. Он непосредственно ориентирован на решение проблемы требований и является единственным документом, к которому можно обратиться в любой момент, чтобы увидеть, что продукт, приложение или система должны делать, а что не должны. В целом документ-концепция представляет суть продукта, и его необходимо защищать так, как если бы весь успех проекта зависел от него (потому что *так оно и есть*).

В некоторый момент возникает закономерный вопрос: “Кто же разрабатывает и поддерживает этот исключительно важный документ? Кто управляет ожиданиями заказчика? Кто ведет переговоры с командой разработчиков, заказчиком, отделом маркетинга, менеджером проекта и руководством компании, которое проявило такой интерес к проекту именно теперь, когда подходит срок сдачи?”.

Практически в каждом успешном проекте (от машинок для аттракционов до систем искусственного дыхания, которые спасли десятки тысяч жизней, не допустив ни единого сбоя программного обеспечения), которым мы занимались, был лидер. Мы можем оглянуться на эти проекты и указать некоего человека или (в случае более крупного проекта) небольшую группу из нескольких человек, которые играли роль, “большую, чем жизнь”. Лидеры ставят концепцию продукта (системы или приложения) во главу угла, как будто это самое важное в их жизни. Они думают о ней постоянно, за едой и во время сна.

Роль лидера продукта

Лидером продукта может быть кто угодно: менеджер продукта, менеджер проекта, менеджер по маркетингу, менеджер проектирования, менеджер информационных технологий, руководитель проекта. Однако название должности в данной ситуации не имеет никакого значения; обязанности одинаковы. И они достаточно велики. Лидер должен заниматься следующим.