

Тема № 2 " Сущность программных продуктов и информационных услуг"

1. Сущность программных продуктов и информационных услуг
2. Основные виды программных продуктов
3. Основные виды информационных услуг
4. Жизненный цикл программного продукта
5. Процесс разработки программного продукта
6. Процесс оказания информационных услуг

1. Сущность программных продуктов и информационных услуг

1.1. Понятие программного продукта и информационной услуги

1.1.1. Сущность программного продукта

Прежде чем погрузиться в изучение маркетинговых вопросов, рассмотрим несколько наиболее важных терминов и определений, относящихся к миру информационных технологий (ИТ).

Компьютерная программа – это последовательность команд компьютера, приводящая к решению поставленной задачи.

Программное обеспечение – это компьютерная программа или совокупность программ, дополненные необходимой для их установки, настройки и эксплуатации технической документацией.

Программный продукт можно охарактеризовать как программное обеспечение, позволяющее решать определенные задачи пользователя и предназначенное для платного или бесплатного распространения на рынке или продажи конкретному клиенту, а не для собственного потребления.

Обобщая данные определения, отметим следующие, важные для маркетингологов аспекты. Компьютерная программа сама по себе – это всего лишь технический феномен, инженерная разработка. С добавлением документации для пользователя и выводом компьютерной программы на рынок последняя трансформируется в продукт, обладающий полезностью для целевых потребителей. А иногда, чтобы стать продуктом компьютерная программа должна быть дополнена еще и непрограммным компонентом, например, мобильное приложение для бега, которое позволяет определить скорость, дистанцию и маршрут очередной пробежки, а также частоту пульса и прочие физиологические параметры, потребует для полноценного функционирования наличия определенного устройства (фитнес-браслета или чего-либо подобного).

С точки зрения маркетинга также целесообразно выделить два основных вида программных продуктов: коробочный и заказной. Коробочный программный продукт предназначается для рынка в целом и может быть продан любому покупателю, заинтересованному в нем. Такой продукт поставляется на условиях «продукт, как он есть» со стандартными для всех покупателей функциями. Заказной программный продукт подразумевает наличие конкретного заказчика и разрабатывается в соответствии с требованиями, выдвинутыми им. По завершении разработки продукт, как

правило, передается заказчику в совокупности со всеми правами собственности на него.

Подобное разделение достаточно важно для маркетолога, т.к. маркетинг коробочного программного продукта востребован в гораздо большей степени и значительно отличается от маркетинга заказного программного продукта.

1.1.2. Структура программного продукта

Большинство существующих сегодня программных продуктов представляют собой сложные комплексы, состоящие из отдельных программных модулей.

Программный модуль – это самостоятельная часть программного продукта, имеющая определенное назначение и обеспечивающая заданные функции обработки автономно от других программных модулей.

Совокупность и взаимосвязь программных модулей между собой принято называть архитектурой программного продукта.

Использование модульного подхода при разработке программного обеспечения не случайно. Такой подход обеспечивает ряд преимуществ, как для разработчика, так и для пользователя программного продукта.

Например, с точки зрения разработчика он позволяет:

- распределять работы по исполнителям, обеспечивая их загрузку и требуемые сроки разработки;
- строить календарные графики проектных работ и осуществлять их координацию в процессе создания программного продукта;
- контролировать трудозатраты и стоимость проектных работ.

Кроме того, разбиение программных продуктов на отдельные составляющие служит основой и для выбора средств их создания. При этом выделяются многократно используемые модули, проводится их типизация и унификация, за счет чего сокращаются сроки и трудозатраты на разработку программного продукта в целом.

Для пользователя (заказчика) деление программного продукта на отдельные модули удобно тем, что позволяет подобрать именно ту конфигурацию программного продукта, которая необходима ему на текущий момент, а кроме того предоставляет возможность изменения конфигурации в процессе использования программного продукта путем приобретения дополнительных или отказа от ранее приобретенных модулей.

Интересен такой подход и с точки зрения маркетинга, т.к. создает предпосылки для осуществления гибкой товарной (за счет создания различных версий продукта) и ценовой (путем использования разных цен для продуктов с различным составом модулей) политик, что позволяет компании следовать стратегии дифференциации, адаптируя продукт под запросы целевых сегментов рынка.

1.1.3. Сущность информационной услуги

Одно из самых простых определений информационной услуги предлагает понимать ее как решение любых задач клиента, основанное на

использовании современных информационных технологий. При этом единого и четкого определения, что именно следует понимать под информационными услугами, не выработано до сих пор.

Так, например, Валерий Гущин, генеральный директор ITG Group определяет информационную услугу как оформленный в виде конечного продукта набор ИТ-процессов, активов и ресурсов, имеющий явно выраженную ценность для бизнеса и предоставляемый пользователю на заранее определенных условиях в целях удовлетворения потребности в ИТ-обеспечении деятельности организации.

Библиотека ITIL, разработанная во второй половине 80-х годов по указанию Британского правительства и фактически ставшая стандартом в сфере управления ИТ, дает два определения информационной услуги.

В более ранней версии ITIL информационная услуга определялась как одна или несколько ИТ-систем, делающих доступными бизнес-процессы клиента. Подобное определение отражает продуктовый подход к понятию ИТ, в котором основную ценность представляет сама ИТ-система, при помощи которой эта услуга предоставляется.

Более поздняя версия ITIL определяет информационную услугу иначе: «Способ предоставления ценности заказчикам через содействие им в получении выходов (результатов на выходе), которых заказчики хотят достичь без владения специфическими затратами и рисками».

Как видим, акцент в определении сместился с технологии на результат. И в этом есть определенный смысл. Заказчикам услуг не нужны ни ИТ-системы, ни сами информационные услуги. Единственное, чего хотят заказчики – это удовлетворить свои потребности и решить вытекающие из них задачи, а как именно со стороны ИТ будет выглядеть предоставляемая им услуга, заказчику в определенной степени безразлично. Такой подход к определению понятия информационной услуги получил название сервисного.

1.1.4. Проблема различения продукта и услуги

То, что потребитель (пользователь, заказчик) не настроен разбираться в тонкостях определений, находит свое отражение и в том, что порой бывает достаточно сложно провести четкую грань между программным продуктом и информационной услугой.

Дополнительные трудности вызывает и то обстоятельство, что программный продукт является товаром нематериальным. Наличие материального носителя, например компакт-диска, на котором он может быть записан, ситуацию не меняет, так как потребительская ценность диска несоизмерима с той потребительской ценностью, которую предполагает программный продукт сам по себе. Как следствие, компании и их клиенты, могут иметь различные точки зрения на вопрос: «продукт или услуга?».

Потребитель, прежде всего, понимает «продукт» как нечто, передаваемое в его распоряжение навсегда или на длительный срок, и рассчитывает на то, что сможет использовать полученный продукт любым способом по своему усмотрению. В то же время «услуги» воспринимаются им

как единоразовые или повторяющиеся действия ИТ-поставщика, решающие некоторую проблему потребителя и приносящие ему те или иные выгоды.

Например, всемирно известная компания Google предлагает широкий спектр сервисов – электронную почту, систему поиска, хранение и обработку фотографий, видеофайлов и документов, онлайн-переводчик и т.п. Подобные сервисы воспринимаются потребителями преимущественно как услуги, хотя для самой компании они выступают как программные продукты.

Стоит ли удивляться, что и в научной литературе можно найти различные и не всегда однозначные трактовки рассматриваемой области. (Впрочем, это в принципе характерно для литературы по проблемам маркетинга услуг). Авторское понимание данного вопроса представлено на рисунке 1.

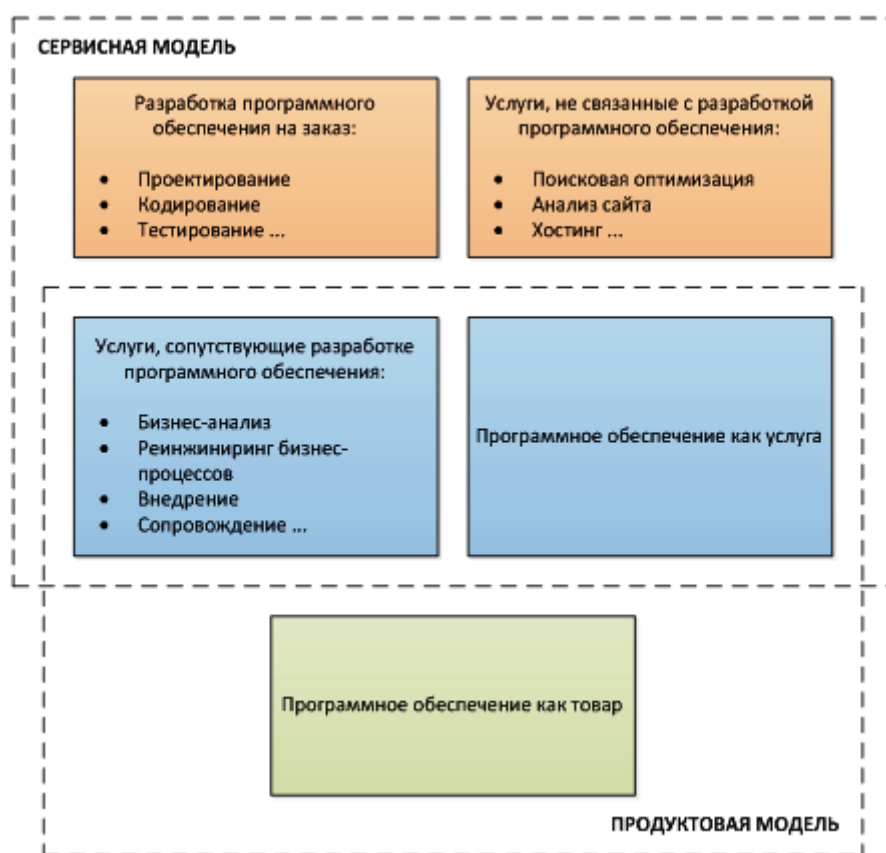


Рисунок 1 – Классификация программных продуктов и информационных услуг

1.2. Основные виды программных продуктов

В общем виде программные продукты принято делить на три основные группы: системное программное обеспечение, средства разработки приложений, пользовательские приложения.

Системное программное обеспечение – программы, которые обеспечивают работу компьютерной системы и управление отдельными ее компонентами, такими как процессор, оперативная память, устройства ввода-вывода и

другими, выступая в качестве посредника между аппаратной частью устройства и приложениями пользователя.

Системное программное обеспечение представлено различными программными продуктами, среди которых можно выделить:

- операционные системы, обеспечивающие распределение вычислительных ресурсов компьютера между отдельными приложениями, а также предоставляющие им доступ к компонентам и периферийным устройствам;
- операционные или системные оболочки, облегчающие пользователю работу с файловой системой;
- драйверы, обеспечивающие работоспособность внутренних компонентов компьютера и его периферийных устройств;
- справочно-информационные и тестирующие программы, анализирующие состав компонентов и периферийных устройств, а также проверяющие их работоспособность;
- антивирусные пакеты, обеспечивающие безопасную работу компьютера и его защиту от различных вредоносных программ;
- архиваторы, «сжимающие» информацию в целях уменьшения занимаемого ею объема памяти;
- различные библиотеки, системные утилиты и другие программные продукты.

К средствам разработки приложений относятся специальные программы, которые используются для создания прикладных программ (приложений). В их числе такие виды программного обеспечения, как:

- а) средства проектирования приложений, в том числе приложения для:
 - графической нотации моделирования бизнес-процессов;
 - построения блок-схем;
 - построения UML-диаграмм;
- б) средства реализации программного кода, включая:
 - языки программирования;
 - средства создания пользовательского интерфейса;
 - средства управления версиями программного кода;
 - средства получения исполняемого кода;
 - средства управления базами данных;
 - отладчики;
- в) средства тестирования программ:
 - средства анализа кода;
 - средства для тестирования функциональности;
 - средства для тестирования производительности.

Пользовательские приложения – программы, которые удовлетворяют конкретные потребности конечных пользователей. Это самая многочисленная группа программных продуктов. Среди них:

- текстовые редакторы;
- электронные таблицы;
- графические редакторы;

- музыкальные редакторы;
- приложения для обработки видео;
- браузеры;
- видеоигры
- и целый ряд других приложений самого разного назначения.

1.3. Основные виды информационных услуг

В научной и нормативной литературе выделяют различные подходы к классификации информационных услуг. Остановимся на 2 принципиально отличающихся подходах: классификации, используемой государственными статистическими органами, и классификации, используемой независимыми аналитическими агентствами.

Национальный статистический комитет Республики Беларусь в своих статистических публикациях в категории «Компьютерные и информационные услуги» выделяет следующие виды информационных услуг:

- консультационные услуги по компьютерным техническим средствам, связанные с их приобретением, установкой и эксплуатацией;
- сопровождение программного обеспечения, консультационные услуги по программному обеспечению;
- консультационные услуги по информационному обеспечению и обработке данных;
- услуги по подготовке, обработке и вводу данных;
- услуги по разработке автоматизированных информационных систем, систем для научных исследований, систем проектирования и управления на основе баз данных, системных и прикладных программных средств;
- услуги по техническому обслуживанию и ремонту оргтехники для офисов, электронных вычислительных машин и используемого совместно с ними периферийного и сетевого оборудования;
- электронные информационно-справочные услуги;
- услуги телематических служб и передачи данных;
- услуги по обучению в области информационных и коммуникационных технологий;
- прочие услуги, связанные с информационными и коммуникационными технологиями.

Схожую классификацию предлагает и Росстат для субъектов российского рынка. Независимые аналитические агентства подразделяют информационные услуги иначе. Так, например, компания IDC, один из ведущих мировых аналитиков рынка информационных технологий, предлагает следующую классификацию (рисунок 2).



Рисунок 2 - Классификация информационных услуг в терминологии IDC

Как видно из рисунка, IDC выделяет три основные категории информационных услуг: проектно-ориентированные услуги, аутсорсинг-ориентированные, и услуги, ориентированные на поддержку и тренинг (регулярные услуги). Каждая категория в свою очередь содержит ряд подкатегорий.

1.4. Жизненный цикл программного продукта

1.4.1. Этапы жизненного цикла программного продукта

Как известно, любой продукт существует на рынке ограниченный отрезок времени, который принято называть жизненным циклом продукта. Этот отрезок состоит из отдельных частей, каждая из которых характеризует особое состояние программного продукта. Подобные состояния называются этапами или стадиями жизненного цикла. В их числе:

1. Анализ потребностей и формирование требований к программному продукту подразумевает анализ потребностей конкретного заказчика (если речь идет о заказной разработке) или потенциальных покупателей продукта, обоснование необходимости разработки, формирование требований к программному продукту.

2. Разработка концепции и технического задания на разработку включает разработку и утверждение концепции программного продукта, удовлетворяющей сформулированным на предыдущем шаге требованиям, а также составление технического задания на разработку продукта.

3. Проектирование программного продукта предполагает разработку архитектуры создаваемого программного продукта, физических и логических моделей базы данных, написание спецификаций с функциональными и нефункциональными требованиями к программному продукту.

4. Непосредственная реализация связана с написанием программного кода, разработкой дизайна пользовательского интерфейса, интеграцией приложения с базой данных и другими сервисами.

5. Тестирование и отладка обеспечивают проверку соответствия функциональности программного продукта потребностям пользователей, сформулированным на этапе разработки технического задания, а также поиск и исправление выявленных дефектов.

6. На стадии внедрения осуществляется инсталляция и настройка программного продукта на рабочих местах пользователей, а также может осуществляться их обучение.

7. Эксплуатация и сопровождение. Последняя стадия включает в себя техническую поддержку работоспособности программного продукта, а также обновление системы по мере развития программного продукта.

В рамках задач маркетинга значимыми и критически важными являются:

- стадия анализа (чем тщательнее и точнее проведен анализ запросов целевых пользователей, тем более полным будет удовлетворение их потребностей);

- стадия разработки концепции и технического задания (вытекающие из технического задания требования должны быть в максимальной степени направлены на удовлетворение нужд целевых пользователей программного продукта);

- стадии внедрения и эксплуатации (качественное обслуживание и оперативное реагирование на сигналы клиента значат не меньше, чем программный продукт сам по себе).

1.4.2. Модели жизненного цикла программного продукта

Термин «жизненный цикл программного продукта» хотя и отражает наличие различных последовательных стадий, через которые проходит любой программный продукт, еще не объясняет, как именно эти стадии могут быть осуществлены на практике. Ответ на вопрос о конкретных вариантах реализации того или иного этапа, а также о связи отдельных этапов друг с другом, как правило, дают соответствующие модели жизненного цикла.

В стандарте ISO 12207 «System and software engineering – Software life cycle processes» модель жизненного цикла определяется как структура, состоящая из процессов, работ и задач, включающих в себя разработку, эксплуатацию и сопровождение программного продукта, и охватывающая «жизнь» программного продукта от установления требований к нему до прекращения его использования.

Конкретные модели определяются особенностью решаемых при разработке задач, установленными ограничениями на ресурсы, имеющимся опытом разработчиков и т.д. Тем не менее, можно выделить некоторые типовые модели жизненного цикла программного продукта, главными из которых являются каскадная и спиральная модели.

Каскадная модель предполагает следование следующим основным принципам:

- строго последовательное выполнение фаз;
- наличие определенных критериев входа и выхода для каждой из фаз;

- полное документирование каждой фазы;
- переход от одной фазы к другой только после формального обзора с участием заказчика;
- неизменность сформулированных требований (технического задания);
- соответствие продукта установленным требованиям как основной критерий качества достигнутого результата.

Каскадная модель имеет ряд преимуществ. Во-первых, она проста и понятна, как заказчикам, так и разработчикам. Во-вторых, она позволяет привлекать к работе над отдельными стадиями независимые команды исполнителей (так как все требования полностью документированы и не меняются в ходе реализации проекта). И наконец, модель позволяет достаточно точно планировать сроки и затраты на разработку программного продукта.

Основные недостатки модели сводятся к тому, что попытка вернуться на одну или две фазы назад, чтобы исправить какую-либо проблему, приводит к значительному увеличению затрат и сбою в графике. К тому же интеграция компонент, на которой обычно выявляется большая часть ошибок, выполняется в конце разработки, что сильно увеличивает стоимость устранения ошибок.

Несмотря на распространенное в последние годы мнение, что модель безнадежно устарела, она не утратила своей актуальности при решении задач, использующих неизменное определение продукта (например, при разработке операционных систем или компиляторов), при повторной разработке типовых программных продуктов, при выпуске новой версии уже существующего продукта, если вносимые изменения вполне определены и управляемы.

Спиральная модель была предложена как альтернатива каскадной модели и базируется на следующих принципах:

- разработка нескольких вариантов продукта, соответствующих различным вариантам требований с возможностью вернуться к их пересмотру;
- создание прототипов программного продукта как средства общения с заказчиком для уточнения требований;
- планирование следующих вариантов с оценкой альтернатив и анализом рисков, связанных с переходом к следующему варианту;
- переход к разработке следующего варианта до завершения предыдущего в случае, если риск завершения очередного варианта (прототипа) становится неоправданно высок;
- использование каскадной модели как схемы разработки очередного варианта;
- активное привлечение заказчика к работе над проектом.

Преимущества спиральной модели вполне очевидны. Во-первых, поэтапное уточнение требований в процессе выполнения итераций позволяет более точно удовлетворить требования заказчика. Во-вторых, использование промежуточных результатов (прототипов) помогает заказчику точнее оценивать реальные сроки выполнения и объемы финансирования. Наконец,

модель предоставляет возможность разработки сложного проекта «по частям», выделяя на первых этапах наиболее значимые требования.

Основные недостатки спиральной модели связаны с ее сложностью. Это и сложность анализа и оценки рисков при выборе вариантов, и сложность поддерживания версий продукта (хранение версий, возврат к ранним версиям, комбинация версий), и сложность оценки точки перехода на следующий цикл. Кроме того, недостатком выступает и «бесконечность» модели – на каждом витке заказчик может выдвигать новые требования, которые приводят к необходимости следующего цикла разработки.

Спиральную модель целесообразно применять при наличии следующих условий:

- когда пользователи не уверены в своих потребностях или когда требования слишком сложны и могут меняться в процессе выполнения проекта;
- когда проект является сложным, дорогостоящим и обоснование его финансирования возможно только в процессе его выполнения;
- когда речь идет о применении новых технологий, что связано с риском их освоения и достижения ожидаемого результата;
- при выполнении очень больших проектов, которые в силу ограниченности ресурсов можно делать только по частям.

Каскадная и спиральная модели являются основными моделями жизненного цикла программного продукта, однако существуют и некоторые другие типы, которые можно рассматривать как «промежуточные». В их числе: итерационная модель, инкрементная модель, V-model, Cleanroom, Dual Vee Model, модель быстрого прототипирования.

Выбор конкретной модели жизненного цикла зависит и от особенностей разрабатываемого продукта, и от того, для кого именно (конкретного заказчика или рынка в целом) он разрабатывается, а также от той контрактной модели (время и материалы, фиксированная цена), согласно которой работают заказчик и разработчик в случае заказной разработки.

1.4.3. Промышленные технологии создания программных продуктов

Многие модели или их комбинации представлены в виде так называемых промышленных технологий создания программных продуктов. Такие технологии, как правило, поддерживаются набором CASE-средств, охватывают все этапы жизненного цикла продукта и успешно применяются для решения практических задач.

Среди наиболее известных примеров промышленных технологий: Microsoft Solution Framework и Rational Unified Process.

Технология Microsoft Solution Framework (MSF) построена на использовании как каскадной, так и спиральной моделей. Она ориентирована на так называемые вехи или ключевые точки проекта, характеризующие достижение какого-либо значимого результата. В рамках MSF выделяют следующие фазы:

- Создание общей картины приложения. Первая фаза предполагает анализ существующей ситуации, определение целей и структуры проекта, подбор участников проектной команды, разработку концепции решения и оценку рисков. Достигаются первые вехи: "Организован костяк команды" и "Создана общая картина решения".

- Планирование. Охватывает планирование и проектирование продукта. На основе анализа требований разрабатываются проект и основные архитектурные решения, подготавливаются спецификации требований на разработку, составляются планы и календарные графики, выбирается среда для разработки, тестирования и пилотной эксплуатации.

- Разработка. Создается определенный вариант решения проблемы, представляющий собой прототип с высокой степенью проработки и детализации, а также подготавливается необходимая документация к стереотипу, в том числе спецификации и сценарии тестирования. Веха этого этапа - "Окончательное утверждение области действия проекта".

- Стабилизация. Осуществляется подготовка к выпуску окончательно утвержденной версии продукта, которая доводится до заданного уровня качества. На этом же этапе выполняется комплекс работ по тестированию, проверяется сценарий развертывания продукта.

- Развертывание. Подготовленное решение в совокупности со всеми необходимыми аппаратными и программными компонентами устанавливается на соответствующих серверах, осуществляется стабилизация работы программного обеспечения в промышленных условиях и его передача команде сопровождения.

Технология Rational Unified Process (RUP) ориентирована на поэтапное моделирование и разработку программного продукта с помощью нотаций языка UML, опираясь при этом на принципы итеративно-инкрементной и отчасти каскадной моделей. В общей сложности технология предполагает выделение 4 основных этапов и 9 видов деятельности. Основными этапами RUP являются:

- Inception (начало проекта). Первый этап ориентирован на постановку основных целей и определение бюджета проекта; выбор основных средств его осуществления: технологий, инструментов, проектной команды; а также на разработку предварительных планов реализации проекта.

- Elaboration (проработка). Второй этап подразумевает анализ ключевых требований к программному продукту и разработку базовой архитектуры решения на его основе.

- Construction (построение). Осуществляется детальная проработка всех требований к программному продукту, подготовка необходимой технической документации и непосредственная разработка программного продукта.

- Transition (передача). На последнем этапе программный продукт развертывается в предполагаемой рабочей среде и становится доступным пользователям.

При этом каждый из этапов может предполагать проведение нескольких итераций, количество которых определяется сложностью выполняемого проекта.

В 2001 году были сформулированы принципы новой технологии разработки программного обеспечения, получившей название agile software development и подразумевавшей максимальную гибкость в процессе создания программного продукта. Примером технологий такого рода является технология экстремального программирования Extreme Programming (XP), основными этапами которой можно считать:

- «Вброс» архитектуры. Проект начинается с обдумывания концепции и установления границ программного продукта, выбора архитектуры и технологии разработки. Результатом выполнения данного этапа является метафора программного продукта, просто и понятно описывающая проектной команде основные принципы работы и назначение программного продукта.

- Истории использования. Требования к программному продукту фиксируются в виде различных сценариев выполнения отдельных функций, которые записываются на специальных карточках. Истории использования формируют требования для планирования очередного релиза программного продукта и в то же время служат основой при разработке приемочных тестов.

- Планирование версий (релизов). Список функциональных возможностей, которые должны быть реализованы в очередной версии программного продукта, утверждается совместно с заказчиком путем выбора соответствующих историй использования. Цель этапа – выровнять ожидания команды и заказчика относительно того, что и в каком виде может быть сделано в ближайшие 2-3 недели, т.е. к дате выхода следующей версии продукта.

- Собственно разработка. Осуществляется в соответствии с планом и включает только те функции, которые были отобраны на этапе планирования.

- Приемочное тестирование. Проводится с участием заказчика, который участвует в составлении тестов для него.

- Выпуск релиза. Осуществляется передача заказчику подготовленной версии программного продукта для ее ввода в коммерческую эксплуатацию или дополнительного бета-тестирования.

По завершении данного цикла происходит переход на следующую итерацию разработки.

1.5. Процесс разработки программного продукта

Для простоты восприятия рассмотрим основные этапы разработки программного продукта на примере каскадной модели жизненного цикла.

1.5.1. Анализ потребностей пользователя

Анализ потребностей пользователя – это процесс сбора требований к программному продукту, включающий в себя их систематизацию, документирование и анализ.

В зависимости от типа разрабатываемого программного продукта и взаимоотношений с пользователем / заказчиком могут использоваться различные подходы к анализу потребностей. Для небольших проектов вполне подойдут гибкие технологии разработки и составление историй использования. Крупные и сложные проекты, как правило, предполагают более формализованные подходы.

Процесс анализа в общем виде можно разделить на 3 взаимосвязанных этапа: сбор требований, их анализ и документирование.

Сбор требований предполагает идентификацию и опрос всех заинтересованных лиц. В число таких лиц в рамках терминологии бизнес-анализа могут входить:

- владелец проблемы – основной потребитель будущего решения (например, владелец компании, бизнес-процессы которой автоматизируются);
- финансовый спонсор – лицо, ответственное за финансирование разработки и которое может принимать решение о выборе предпочтительного варианта с точки зрения затрат на его разработку (например, руководитель финансового отдела);
- эксперты предметной области – специалисты, которые могут дать экспертные оценки и провести консультации (например, по спортивной диете в случае фитнес-приложения);
- пользователи – конечные пользователи, обладающие той или иной заинтересованностью, навыками и предпочтениями по использованию программного продукта;
- технические специалисты – специалисты, которые будут обслуживать программный продукт в будущем;
- и другие.

При анализе собранных требований выясняются общие ожидания целевых пользователей / заказчика относительно функциональности разрабатываемого программного продукта. На этом этапе нужно, во-первых, систематизировать и обобщить требования различных заинтересованных лиц, которые могут не только не совпадать, но и противоречить друг другу, и, во-вторых, достаточно глубоко их детализировать, так, чтобы степень проработки собранных требований позволяла сформулировать четкое техническое задание разработчикам программного продукта.

Документирование требований может быть выполнено в различных формах. Остановимся на наиболее часто используемых вариантах:

- Списки требований. Предполагают скрупулезную фиксацию всех имеющихся требований. При всей простоте и внешней надежности подхода, такие списки перечисляют отдельные требования независимо друг от друга и в отрыве от контекста использования, что затрудняет их дальнейший анализ.
- Прототипы. Прототип представляет собой макет будущего программного продукта, соответствующий исходным требованиям заказчика.
- Сценарии использования. Содержат текстовое описание всех способов, которыми пользователи могут работать с программным продуктом

для решения тех или иных задач. Сценарии использования формулируют функциональные требования к программному продукту.

- Спецификация требований к программному продукту. Спецификация требований является полным описанием поведения создаваемого программного продукта и включает ряд сценариев использования, описывающих все виды взаимодействий пользователей с программным продуктом. В отличие от сценариев использования спецификации содержат ряд нефункциональных требований, таких как требования эффективности или надежности работы.

1.5.2. Проектирование и кодирование программного продукта

Проектирование – этап, на котором создается образ программного продукта, который, как правило, включает в себя: общую архитектуру решения, описание отдельных компонентов, а также описание пользовательских интерфейсов, определяющих внешний вид программного продукта.

Процесс проектирования тесным образом связан с построением различного рода моделей и визуализаций, каждая из которых помогает лучше понять тот или иной аспект создаваемого программного продукта.

Состав, назначение и степень детализации моделей, используемых в каждом конкретном случае, зависят от ряда факторов:

- сложности проектируемого программного продукта;
- требуемой полноты его описания;
- требований, предъявляемых к используемой нотации моделирования (UML, BPMN, IDEF и др.)
- времени, отведенного на проектирование.

Соответствующим образом варьируется и список артефактов, получаемых на этом этапе. В их числе могут быть названы:

- структурные модели (диаграммы классов, диаграммы компонентов);
- модели поведения (диаграммы вариантов использования, диаграммы состояния, диаграммы действий);
- физическая и логическая модели баз данных;
- макеты отдельных страниц и вкладок программного продукта;
- и др.

Программирование или кодирование подразумевает написание программ на конкретном языке программирования. Оно может осуществляться для различных технологических платформ (.NET, J2ME, SAP и др.) и операционных систем (MS Windows, Linux, Android и др.), и, как следствие, основываться на использовании различных языков программирования (C++, C#, Java, Perl, HTML, XML, JSP, PHP и др.). Выбор конкретной платформы и языка программирования определяется сущностью и назначением разрабатываемого программного продукта.

1.5.3. Тестирование программного продукта

Тестирование программного продукта подразумевает проверку на наличие соответствия между тем, что было получено в результате разработки, и теми ожиданиями, которые закладывались в программный продукт на этапе проектирования и подготовки технического задания. Тестирование позволяет выявить дефекты реализации и внести соответствующие исправления.

Тестирование – процесс анализа программного средства и сопутствующей документации с целью выявления дефектов и повышения качества продукта.

Все виды тестирования программного обеспечения, в зависимости от преследуемых при этом целей, можно условно разделить на следующие группы.

По объекту тестирования: функциональное тестирование, тестирование производительности, тестирование удобства использования, тестирование интерфейса пользователя, тестирование безопасности.

По степени автоматизации: ручное тестирование, автоматизированное тестирование, полуавтоматизированное тестирование.

По времени проведения тестирования: альфа-тестирование (тестирование специалистами компании), бета-тестирование (тестирование пользователями).

Уровень тестирования определяет объект проверки. Это может быть отдельный компонент, интерфейс, обеспечивающий взаимодействие между различными компонентами, или вся система в целом:

- Компонентное тестирование осуществляет проверку функциональности отдельных частей программного продукта (программных модулей, объектов, классов и т.д.), при условии, что они могут быть протестированы автономно от других компонент.

- Интеграционное тестирование позволяет проверить связи между компонентами, а также их взаимодействие с другими частями программного продукта (операционной системой, оборудованием и т.д.).

- Системное тестирование направлено на проверку полноты и качества реализации функциональных и нефункциональных требований к программному продукту. По его итогам выявляются дефекты, связанные с нерациональным использованием системных ресурсов, уязвимостью с точки зрения информационной безопасности продукта, непредусмотренные сценарии использования и т.д.

- Приемочное тестирование предполагает выполнение ряда заранее составленных тестов, моделирующих типичные сценарии использования программного продукта, а также проверку соответствия продукта заданным заказчиком приемочным критериям. По результатам такого тестирования программный продукт либо принимается как полностью соответствующий заданным требованиям, либо отправляется на доработку.

1.5.4. Внедрение программного продукта

Внедрение программного продукта – это процесс его развертывания в требуемой рабочей среде и последующей настройки под определенные

условия использования. В случаях со сложным программным обеспечением внедрение также предполагает обучение пользователей.

Внедрение программного продукта в большей степени относится к рынку корпоративных программных продуктов, чем к обычным пользовательским приложениям. Внедрение последних, как правило, сводится к простым действиям: принятию условий пользовательского соглашения и установке продукта на устройство пользователя.

Внедрение же больших информационных систем промышленного назначения представляет собой сложный и трудоемкий процесс, с юридической точки зрения нередко оформляемый в виде отдельного проекта. В зависимости от масштаба предполагаемых работ, текущего состояния баз данных и каналов связи, готовности аппаратных средств и персонала заказчика процесс внедрения может занимать от двух недель до одного года.

Рассмотрим основные этапы процесса на примере внедрения сложного программного обеспечения, которое уже разработано, но требует адаптации под нужды конкретного клиента.

1. Предпроектное обследование и разработка технического задания на внедрение. Предпроектное обследование предполагает изучение текущего состояния объекта автоматизации, используемых предприятием технологий для загрузки, хранения, обработки и выгрузки данных, требований к информационной безопасности продукта, а также дает ответ на вопрос о предполагаемых объемах работ по внедрению программного продукта.

Результатом обследования должны стать подготовленные для заказчика рекомендации, которые помогут ему грамотно и точно составить техническое задание на внедрение. В задании прописываются требования заказчика к внедряемому программному продукту, в том числе приводятся алгоритмы расчетов, конфигурация информационной системы, формы выходных документов, порядок и формат обмена данными.

2. Заключение договора и составление графика выполнения работ. По результатам анализа технического задания, выполняемого совместно двумя сторонами, заключается договор на внедрение, а также составляется график выполнения предусмотренных проектом работ с перечнем отдельных этапов и указанием дат их начала и окончания. График работ по внедрению является неотъемлемой частью договора.

3. Предварительная настройка программного обеспечения и подготовка баз данных. Перед непосредственной установкой программного продукта выполняется предварительная настройка и адаптация программ и баз данных, направленная на подготовку программного обеспечения к развертыванию в рабочем окружении заказчика, а также подготавливаются необходимые инсталляционные пакеты и установочные файлы.

4. Установка программного продукта. Набор конкретных действий, выполняемых на данном этапе, зависит как от специфики самого внедряемого продукта, так и от условий, в которых приходится это делать. Тем не менее, в числе наиболее характерных можно отметить следующие действия: развертывание базы данных на серверах заказчика, установку клиентской

части программного обеспечения, настройку подключений между серверной и клиентской частями и другие.

5. Обучение пользователей работе с программным продуктом. В завершении работ по внедрению производится обучение персонала заказчика, как обычных сотрудников, так и системных администраторов, в обязанности которых в дальнейшем будет входить выполнение большинства требуемых настроек и обслуживание программного продукта.

1.5.5. Эксплуатация и сопровождение программного продукта

Эксплуатация – это процесс использования программного продукта потребителем для решения им своих задач.

Сопровождение программного продукта – это процесс улучшения, оптимизации и устранения дефектов программного продукта, осуществляемый разработчиком после передачи продукта в эксплуатацию. Сопровождение необходимо для того, чтобы программный продукт на протяжении всего периода эксплуатации удовлетворял требованиям пользователей.

Деятельность персонала, осуществляющего сопровождение программного продукта, включает четыре ключевых аспекта:

- поддержку управляемости программного продукта в течение всего периода эксплуатации;
- поддержку нормальной работоспособности различных модификаций программного продукта;
- совершенствование существующего функционала;
- контроль производительности программного продукта и предотвращение его падения до неприемлемо низкого уровня.

В общем случае это подразумевает решение следующих задач:

- устранение дефектов и сбоев;
- оптимизацию дизайна;
- расширение функциональных возможностей;
- разработку интерфейсов взаимодействия с другими (внешними) системами;
- адаптацию программного продукта (например, портирование) для возможности работы на другой аппаратной платформе, применения новых системных возможностей, функционирования в среде обновленной телекоммуникационной инфраструктуры и т.п.;
- вывод программного продукта из эксплуатации.

В сопровождение также входят послепродажные коммуникации с пользователями, предоставляющие последним возможность получить консультацию и техническую помощь по тем или иным аспектам функционирования программного продукта. Кроме того, такие коммуникации позволяют накапливать обратную связь от пользователей: отзывы, претензии, пожелания по развитию функционала и т.п. и учитывать полученный потребительский отклик при дальнейшем развитии программного продукта.

1.6. Процесс оказания информационных услуг

1.6.1. Основные этапы взаимодействия с заказчиком при оказании информационных услуг

Несмотря на большое разнообразие предлагаемых рынком информационных услуг, а также существенную разницу в содержании отдельных из них, можно выделить некоторые ключевые этапы, присутствующие при взаимодействии ИТ-компании с заказчиком.

1. Анализ потребностей заказчика. Как и в случае с разработкой программных продуктов в первую очередь необходимо четко определить потребности заказчика, а также возможные пути удовлетворения этой потребности.

Так, например, если клиента интересуют услуги по разработке Интернет-магазина, на этом этапе необходимо определить, устраивают ли его готовые шаблонные решения или требуется более уникальный продукт, необходимы ли ему дизайнерские услуги, услуги по хостингу и администрированию, требуется ли интеграция сайта с системами онлайн-платежей и т.д.

2. Подбор необходимой услуги или перечня услуг. В результате проведения анализа формируется перечень услуг, оказание которых необходимо клиенту, а также разрабатываются спецификации требований (например, требования к серверу, на котором будет осуществлен хостинг) и составляются технические задания, если это необходимо характером услуги.

В зависимости от сущности вошедших в этот перечень услуг, следующий этап во взаимоотношениях ИТ-компании с заказчиком может выглядеть по-разному. Так, например, следует различать единоразовые (разработка дизайна для сайта) и многократно оказываемые (администрирование программного обеспечения) услуги. Не менее важно проводить различие между типовыми услугами и услугами, носящими индивидуальный характер и предполагающими высокую степень кастомизации (адаптации) для каждого конкретного клиента.

3. Определение временных и трудовых затрат на оказание услуги. Для ряда типовых услуг (например, уже упоминавшихся выше услуг по хостингу сайтов или разработке сайтов на основе типовых шаблонов) эта стадия является скорее формальной. Затраты на хостинг будут напрямую связаны с объемом выкупаемого на сервере дискового пространства, а также будут зависеть от того, осуществляется ли этим сервером поддержка, например, PHP и MySQL. Все возможные варианты выполнения этих условий не составит труда учесть в стандартном прайс-листе. В то же время для более сложных и уникальных по своему характеру информационных услуг этот этап имеет важное значение. Именно на нем производится оценка сложности предстоящих работ, определяются потребность в специалистах и наиболее вероятные сроки выполнения работ. Подобные оценки служат исходными данными при составлении бюджета на оказание информационных услуг.

4. Определение стоимости или составление бюджета. Как уже отмечалось ранее, для сравнительно простых по своему характеру услуг их

стоимость определяется на основании прайс-листов и зависит от некоторого (как правило, небольшого) числа параметров (стоимость одного часа работы ИТ-консультанта, умноженная на количество часов, и т.п.).

Для более сложных работ, например, работ по системной интеграции программного продукта в бизнес-процессы компании, составляется график выполнения работ и утверждается бюджет проекта, в котором также могут быть заложены дополнительные затраты на различные неучтенные при предварительном планировании работы.

5. Заключение соглашения об уровне сервиса (SLA). Одним из ключевых документов, регулирующих взаимоотношения сторон во время оказания информационной услуги, в большинстве случаев является соглашение об уровне предоставления услуги (Service Level Agreement, SLA). Это документ, который содержит согласованный сторонами уровень качества предоставления услуги. В таком соглашении оговариваются:

- описание доступности сервиса (время простоя в год или месяц, период доступности 24x7 или в рабочие дни 8x5, общее количество простоев, плановые простои);
- количество и размещение пользователей и/или оборудования, использующих данную услугу;
- перечень возможностей и регламенты по добавлению новых приложений, пользователей, услуг;
- частота проведения регламентных работ (например, архивирования, резервирования и восстановления данных);
- максимальный срок устранения инцидентов (устранение сбоев, мешающих работе);
- стоимость услуг, описание платежей, связанных с сервисом;
- ответственности клиентов при использовании услуги (подготовка, поддержка соответствующих конфигураций оборудования, программного обеспечения);
- описание уровней обеспечения конфиденциальности;
- порядок внесения изменений в SLA;
- условия расторжения SLA (передача всех данных потребителю, помощь в миграции, уничтожение резервных копий и архивов, время обеспечения конфиденциальности).

1.6.2. Технология оказания информационных услуг

Технологии оказания информационных услуг во многом зависят от того, какого рода услуги будут оказываться: системная интеграция программного обеспечения, консультационные услуги, хостинг или техническая поддержка. Существенные отличия в содержании каждой из перечисленных выше услуг обуславливают и разницу в применяемых при их оказании технологий. Системная интеграция отталкивается от своих принципов и алгоритмов, техническая поддержка от своих.

В этом контексте, проще всего говорить об аутсорсинго-ориентированных услугах, направленных на разработку программных

продуктов под заказ. Технология оказания такого рода услуг давно отработана и фактически совпадает с рассмотренной выше технологией разработки программного обеспечения. Некоторая разница может заключаться в отсутствии отдельных этапов разработки. Например, этапа анализа потребности и составления технического задания, если заказчик приходит к разработчику с уже готовым техническим заданием или этапа сопровождения и технической поддержки, в том случае, когда разработанный программный продукт со всей сопутствующей документацией передается заказчику и дальнейшее сопровождение продукта в процессе его эксплуатации осуществляется заказчиком.

Однако услуги, связанные с разработкой программного продукта, это только часть всех оказываемых на рынке информационных услуг. Охватить специфику оказания всех возможных информационных услуг не представляется возможным, да и вряд ли целесообразно. Остановимся подробнее на двух наиболее ярких примерах.

Услуги по системной интеграции. Технология оказания услуг по системной интеграции может быть сведена к следующим этапам:

- технический аудит бизнес-процессов заказчика;
- выбор экономически оправданного, интегрированного решения для реализации конкретных задач заказчика;
- работы по созданию и запуску в эксплуатацию необходимых клиенту систем (ПО или программно-аппаратных комплексов), в том числе разработка специализированных решений, включающих поставку аппаратной платформы и разработку ПО; интеграция разработанного решения с другими приложениями заказчика; интеграция различных аппаратных и программных средств в единые подсистемы; разработка, производство, монтаж, поддержка и обслуживание программно-аппаратных комплексов, предназначенных для решения определенных заказчиком задач.

Техническая поддержка. Стандартная техническая поддержка, осуществляемая, например, компанией Oracle, подразумевает предоставление пользователям следующих возможностей:

- доступ к электронной информационной системе технической поддержки;
- прямые телефонные консультации;
- получение технической информации и дополнительных программных компонент (так называемых patch-ей) для разрешения проблем и ошибок, обнаруженных в программном обеспечении Oracle;
- бесплатное получение обновлений поддерживаемого программного обеспечения (updates) для обеспечения эффективной совместимости с новыми версиями операционных систем;
- бесплатное получение обновленных версий поддерживаемых продуктов с новыми функциональными возможностями (upgrades), по мере их поступления;
- возможность миграции поддерживаемого программного обеспечения при переходе из одной операционной среды в другую.

Широта представленных на рынке информационных услуг накладывает свой отпечаток и на содержание маркетинга: каждой конкретной информационной услуге может потребоваться свой специфический подход.