

Министерство образования и науки, молодежи и
спорта Украины

Харьковский национальный университет

Ревенчук Илона Анатольевна

Программной инженерии

Раздел 4 - Управление процессами предметной области

6.050103 - Программная инженерия, 8.05010301 -
Программное обеспечения систем, 8.05010302 -
Инженерия программного обеспечения

Харьков

2005

Содержание

Введение.....	3
Теория.....	4
Краткий обзор процессов.....	4
Введение в управление процессами предметной области.....	6
Связь с 34 компетенциями	7
Определение процессов предметной области	7
Четыре класса систем разработки программ	9
Модели выбора проектов.....	14
Менеджмент проектов.....	16
Финансовые аспекты	17
Практика.....	23
Вопросы для самопроверки.....	23
Выводы.....	24
Перечень ссылок.....	25

Раздел 4 - Управление процессами предметной области

В разделе рассматриваются: вопросы управления процессами предметной области, классы предметных областей, модели выбора проектов, а также финансовые аспекты.

Теория

Краткий обзор процессов

Процесс - это ограниченный ряд взаимосвязанных действий, в ходе осуществления которых используется один или более исходных продуктов, а затем с помощью одного или нескольких преобразований создается выходной продукт, который представляет ценность для заказчика. Процесс представляет собой действия, инструменты, методы и технологии, в ходе которых преобразуются входные объекты (сырье) в выходные объекты (готовые продукты).

Менеджмент процессов - это один из видов деятельности управления проектами, который необходимо учитывать перед тем, как начинать любую работу, и который позволяет установить структуру методов, используемых при оценке реализации проекта. Менеджмент процессов гарантирует корректное выполнение процедур, стратегий и модели жизненного цикла организации. В ходе осуществления менеджмента процессов выполняется управление действиями по разработке ПО.

В своей книге «Менеджмент процесса разработки ПО» Уоттс Хэмфри пишет:

«Разработка ПО может быть чрезвычайно сложной, и зачастую существует множество альтернативных способов, предназначенных для выполнения различных задач. Определенный процесс может помочь профессионалам в области программного инжиниринга решить проблему выбора с помощью какого-то определенного способа. Благодаря наличию фиксированного определения процесса они могут лучше понимать, что им следует делать, что следует ожидать от своих сотрудников, и что они, как ожидается, обеспечат в свою очередь. Это позволяет сосредотачиваться на выполнении своей работы... Однако разработка ПО не похожа на обычный вид деятельности, который может быть структурирован и упорядочен как повторяющееся производство или конторская процедура. В данном случае рассматривается интеллектуальный процесс, который должен динамически адаптироваться к творческим потребностям профессионалов и их задачам. При этом требуется достижение компромисса между индивидуальной потребностью в гибкости и организационной потребностью в стандартах и последовательностью».

При этом необходимо учитывать следующие факторы:

1. Поскольку программные проекты носят самый различный характер, процессы, применяемые в ходе осуществления программного инжиниринга, также должны различаться.
2. В условиях отсутствия универсального процесса разработки ПО организации и проекты должны определить процессы, которые удовлетворяют их собственным уникальным потребностям.
3. В ходе осуществления процесса, используемого для реализации данного проекта, должен учитываться уровень опыта членов команды, текущий статус продукта, а также доступные

инструменты и возможности.

В области разработки и поддержки программных продуктов необходимо стремиться получить наивысший показатель коэффициента возврата инвестиций (КОИ). Для этого в настоящее время разработан ряд методов:

1. Формальный менеджмент рисков. Эмпирическая стоимость и оценка графика. управление проектами, основанное на использовании метрических показателей. отслеживание прибавочной стоимости. отслеживание дефектов на фоне целей достижения качества. менеджмент программ, ориентированных на персонал.

2. Менеджмент действий по выполнению конфигурации.

3. Трассировка промежуточных требований

4. Проектирование ПО, основанного на системной архитектуре.

5. Способность к взаимодействию данных с базой данных.

6. Формальное определение и контроль интерфейсов.

7. Видимая и контролируемая разработка.

8. Повторное использование, оправданное соображениями цены и качества.

9. Формальные проверки.

Процессы могут разрабатываться и поддерживаться посредством методов, которые аналогичны методам разработки и поддержки продуктов. При этом должно формулироваться следующее:

- требования, определяющие описываемый процесс,
- архитектура и план, поддерживающие информацию о методах определения процесса,
- реализация схемы процесса в проектной или организационной ситуации,
- утверждение описания процесса путем измерений,
- развертывание процесса с созданием широко распространенного действия в пределах организации или проекта, для которого предназначен процесс.

Активы процесса разработки ПО обычно включают:

- стандартный процесс разработки ПО в масштабах организации,
- описания жизненных циклов разработки ПО,
- руководящие принципы и критерии подгонки стандартного процесса создания ПО на уровне организации,
- база данных процесса разработки ПО организации,
- библиотека документации, связанная с процессом разработки ПО.

Процесс, по сути, является парадигмой менеджмента, позволяющая улучшить качество путем:

1. формального определения процесса,
2. измерения процесса,
3. обратной связи и контроля,
4. усовершенствования,
5. оптимизации.

Введение в управление процессами предметной области

Процессы, определенные в предметной области, представляют собой взаимосвязанные действия, специфичные для организации-разработчика программного проекта. Оценка качества программного проекта основывается на том, насколько качественно ПО может разрешать конкретные проблемы, имеющие отношение к предметной области. Для пользователя ПО все определяется его деловой предметной областью, отличной от предметной области специалиста в области информатики либо инженера-программиста. От менеджера проекта, руководящего разработкой ПО, требуется понимание предметной области, в которой программа призвана выполнять специфические задачи.

Существует перечень основных вопросов, от ответа на которые зависит понимание предметной области, в которой работает менеджер проекта. Эти вопросы образуют матричный набор, в котором устанавливается соответствие между предметными областями класса продукта, типами продукта и разрабатываемыми компонентами ПО.

После завершения разбиения на категории предметной области необходимо получить представление относительно разницы между проектами и жизненными циклами разработки программного продукта. Эти различия являются очевидными для коммерческих и общедоступных проектов. Понимание этих различий осуществляется в рамках первого этапа

жизненного цикла разработки программного продукта и играет важную роль при взаимодействии между практикой менеджмента качества программного продукта и заказчиками. Процессы предметной области, которые являются общепринятыми для всех организаций, представляют собой определения критериев моделей выбора проекта, анализ портфеля заказов проекта, а также взаимодействие между основными финансовыми соотношениями, проявляющимися при анализе проекта. Каждое из этих обобщенных действий, реализующих менеджмент процесса предметной области, будет рассмотрено.

Связь с 34 компетенциями

При управлении процессами предметной области необходимы следующие навыки менеджмента проектов.

Методики разработки продукта.

1. Оценка альтернативных процессов - оценка различных подходов.
2. Понимание действий по разработке продукта - изучение цикла разработки ПО.

Навыки менеджмента проектов.

3. Отслеживание процессов - мониторинг совместимости проектной команды.

Навыки менеджмента персонала.

4. Проведение эффективных встреч - планирование и проведение эффективных встреч.

Определение процессов предметной области

Процессы предметной области представляют собой взаимосвязанные действия, которые являются специфичными для организации, в которой был разработан рассматриваемый программный проект. 1-ым шагом на этом пути является определение предметной области, для которой в конечном итоге, будет разрабатываться программный проект, т.е. на этапе произвольного анализа предметной области критически важной является точка зрения конечного пользователя. И она должна быть обязательно учтена при оценке различных возможностей программы. Если отсутствует "реальный" заказчик, нам следует ориентироваться на "голос заказчика". Качество программного проекта определяется тем, насколько хорошо ПО решает специфические проблемы, связанные с предметной областью. Исходя из этого при поставке качественного ПО менеджер проекта должен уметь определять предметную область, при использовании в которой ПО будет разрешать специфические потребности.

При разработке программного продукта выделяются шесть классов предметных областей:

1. **Потребительский.** Отдельные пользователи приобретают программные продукты

потребительского класса для личного применения. Подобного рода продукты могут применяться дома, по время путешествий либо па работе. Ключевым моментом в данном случае является то, что потребительский рынок является массовым и обычно ориентируется на покупателя, чувствительного к ценам. Примерами таких потребительских продуктов могут служить мобильные телефоны, автомобили, телевизионное оборудование, персональные компьютеры, а также персональные цифровые помощники.

2. Деловой. Большинство же программных продуктов нацелены на предметную область, связанную с деловым классом. Здесь ключевой задачей является организация поставки деловым заказчикам экономического программного продукта, который способен увеличить норму прибыли в бизнесе. Программные продукты этого класса обычно дороже по сравнению с программами потребительского класса, а также обязательно включают сопровождение, обслуживание и услуги по выполнению установки. Примерами программных продуктов подобного типа являются инструментальные средства баз данных, такие как Oracle, программы планирования корпоративных ресурсов (PeopleSoft), наборы инструментальных средств по разработке (WebSpere и Visual- Age), а также операционные системы (Solaris).

3. Индустриальный. Индустриальные продукты образуют специфичный подкласс деловых программных продуктов. Подобные программные инструменты приобретаются для удовлетворения специфических потребностей, например, автоматизации машин, автоматизации и интеграции деятельности фабрик, а также в качестве управляющего встроенного ПО. Эти программы имеют специальное назначения и обычно используются в отдельной индустриальной отрасли, например, в автомобилестроении, пищевой промышленности либо производстве полупроводников. Примерами подобных программных продуктов могут служить ПО автоматизации деятельности фабрик от компании Factory- Works, встроенные системы разработки от компании Wind River, а также инструментальные средства моделирования процессов (программный продукт разработки фирмы Hewlett Packard).

4. Режим реального времени. Программы из класса режима реального времени используются с целью управления процессами, которые имеют определенный и ограниченный во времени бюджет. Системы реального времени применяются при сборе данных, имеющих отношение к событиям, длящимся менее одной микросекунды. Программы реального времени могут управлять медицинскими встроенными устройствами, которые применяются, например, миротворцами в "горячих точках" планеты. Информация, которая поступает от датчиков в этом случае, может быть обработана за время, соответствующее промежутку между "двумя ударами сердца". Эти продукты могут также использоваться в качестве интерфейса, связывающего наборы аналоговых данных (например, голос либо музыка) и результаты их преобразования в цифровые данные, которые могут быть сохранены на винчестере либо записываемом диске CD-ROM. Все ПО реального времени создается специально для целевого аппаратного обеспечения, совместно с которым оно будет применяться в дальнейшем.

5. Продукт с минимальной задержкой. ПО из класса продуктов с минимальной задержкой, должно разрабатываться с учетом ограниченного запаса времени, выделяемого на его выполнение. Примером подобного ПО является программа для АТМ-компьютера, а также выполняющая верификацию кредитных карточек при выполнении заказов через Internet. Большая часть программных продуктов из этого класса являются частью деловых либо

индустриальных программных продуктов.

6. Научный. Научные программные продукты имитируют действия, происходящие в реальном мире, с помощью математических формул. При этом создаются математические модели, имитирующие объекты реального мира. Например, на компьютере могут моделироваться некоторые из летных характеристик самолета. Могут имитироваться моря, реки и горы. Вообще говоря, моделированию и имитации может подвергаться практически любой объект с известными характеристиками. Результатом имитации является огромное количество вычислений, для выполнения которых может потребоваться вычислительная мощность суперкомпьютеров. По мере совершенствования персональных компьютеров все большее количество лабораторных экспериментов может преобразовываться в компьютерные модели, которые в интерактивном режиме просматриваются студентами. При этом исключаются затраты и риск, с которыми связано проведение реальных экспериментов. К этому классу можно отнести программный продукт Matlab, используемый при разработке больших математических формул, Analytica (разработка крупномасштабных деловых моделей), а также Expert Choice (разработка широкомасштабных систем принятия решений). Научные программы обычно представляют собой наборы инструментальных средств специального назначения, применяемые для решения различных задач.

Четыре класса систем разработки программ

Четыре класса систем разработки программ представляют методы, с помощью которых может быть создан и поставлен программный продукт (с точки зрения разработчика). Этим четырем классам соответствуют различные планы и жизненные циклы по разработке ПО. Несмотря на то, что любой процесс разработки программного продукта является итеративным, в реальном деловом мире в качестве основы используется существующий портфель программных продуктов. Во время концептуальной стадии менеджер проекта работает над концепцией программного продукта, а также выбирает предварительный жизненный цикл разработки. Результаты более ранней работы оказывают влияние на один или более указанных ниже системных классов программного продукта:

1. Новый программный продукт. Разработка нового программного продукта начинается с определения требований. Затем производится продвижение согласно циклу разработки ПО вплоть до момента его поставки заказчику. При этом могут использоваться некоторые инструментальные средства разработки, а также возможные объектные библиотеки (при необходимости). В этом случае создавался поистине новый программный продукт, в котором использовались преимущества новой технологии (Internet), а также нового набора инструментальных средств программирования (Java). Также открываются новые ниши рынка, поскольку принимаются новые постановления государства, например, в области телекоммуникаций либо касающиеся отмены жесткого регулирования банковской деятельности.

2. Реинжиниринг существующего программного продукта. Реинжиниринг существующего программного продукта проще, чем разработка новых программ. Для подобного рода продуктов изначально используется либо устаревшая технология программирования, либо устаревшее аппаратное обеспечение. Примером программы подобного рода может служить

система сбора данных (реализованная на базе DOS), которая была переработана для использования в среде Linux.

3. Интеграция компонентов. Процесс использования доступных программных продуктов (commercial-off-the-shell, COTS) с последующей их интеграцией в программный продукт именуется интеграцией компонентов. В качестве примера программы этого типа может служить сконструированный на базе доступного встроенного инструментального средства разработки баз данных, инструмента генерирования сценариев, а также графического интерфейса пользователя (GUI) генератор, позволяющий создавать новый программный продукт, используемый для интеграции промышленного оборудования в глобальную производственную систему.

4. "Героическое" сопровождение. Этап "героического" сопровождения начинается в том случае, если компания пытается "втиснуть" последний "выигрышный" бит в существующий программный продукт, который бы подвергнут переработке. Компании по производству ПО проявляют большое внимание вопросам менеджмента и своевременного выпуска новых версий программ, реализующих на практике дополнительные возможности внутри существующего портфеля программных продуктов. Если разрабатывается совершенно новый программный продукт либо подвергается коренной переделке старый продукт, всегда существует риск уменьшения существующего уровня продаж, вместо привлечения новых покупателей. Своевременные решения могут реализовываться по причине задержки выпуска новых программных продуктов. Они могут представлять собой версии старых программных продуктов, на которые в результате осуществления "героических усилий" была "надета новая одежда". Например, обратимся к "старой доброй" DOS: вместо переделки всей системы интерфейс командной строки был просто заменен псевдографическим интерфейсом. Подобная методика в программной индустрии известна под названием "та же собака, но новый ошейник".

Первая матрица, разрабатываемая менеджером проекта, включает идентификацию типа предметной области программного продукта (см. табл. 4.1). Этот тип предметной области находится на пересечении шести классов предметной области программного продукта и классов типа программ. При этом ПО может быть определено таким образом, что будет указываться в нескольких ячейках рассматриваемой матрицы. Например, предположим, что разрабатывается новый программный продукт, использующий возможности Web, который будет применяться для регистрации личных DVD-дисков в торговом клубе, где зарабатываются очки для займа (либо выплачивается небольшой гонорар в случае, если количество очков будет недостаточным). Этот продукт будет "жить" стараниями заказчика, а классы предметных областей своевременного программного продукта в данном случае представляют собой как новый программный продукт, так и результат интеграции компонентов. И хотя концепция программ относится к числу новых, а также может быть разработано новое ПО, доступны многие разработанные ранее библиотеки компонентов. В матрице, представленной на рисунке, соответствующая ячейка помечена "X".

В качестве другого примера может служить расширение возможностей существующего программного продукта, встроенного в производственный процесс, для включения информации прошедших этапов процесса, а также определения оптимального процесса, используемого при производстве программного продукта. При этом используются данные относительно

производственной прибыли в прошлом, а также заказы клиентов. В этом случае можно сразу же сказать, что речь идет о переработке программного продукта, но при этом будет также разрабатываться новое ПО. Разрабатываемый продукт может затрагивать четыре класса предметных областей: деловой, индустриальный, режим реального времени, а также научный. Причина использования делового класса заключается в том, что в данном случае выполняется оценка хронологической информации, касающейся оценки производственной прибыли. Причина использования индустриального класса и класса режима реального времени состоит в том, что речь идет об автоматизированном оборудовании, установленном на фабрике. Наличие научного класса объясняется тем, что используются алгоритмы оптимизации, необходимые для определения лучшего индивидуального потока производства продукта на рассматриваемой фабрике. Сказанное представлено на примере матрицы в виде буквы «О» в соответствующих ячейках матрицы.

Таблица 4.1 - Идентификация типа предметной области программного продукта

1. ШАГ. Идентификация типа предметной области программного продукта	Потребитель	Физический	Индустриальный	Режим реального времени	Режим минимальной задержки	Научный
Разработка нового аппаратного/программного продукта	х	о	о	о	х	о
Реинжиниринг существующего продукта		о	о	о		о
Интеграция компонентов	х				х	
Героическое сопровождение						

Третий этап определения предметной области процесса заключается в назначении классов компонентов продукта. Обзор этого набора классов также производится с точки зрения конечного пользователя. Класс насчитывает всего шесть составляющих элементов, а ключевой вопрос в данном случае звучит следующим образом:

"Поставку каких компонентов должен ожидать заказчик?" В обязанности менеджера по разработке ПО входит получение ответа на вопрос относительно ожиданий конечного пользователя в области разработки продукта. Всего же существует шесть классов компонентов продукта:

1. ПО;
2. аппаратное обеспечение;
3. человеческий фактор;
4. база данных;
5. документация;
6. процедуры.

Если в ходе осуществления проекта разрабатывается "чистый" программный продукт, конечный пользователь вправе ожидать получить набор установочных носителей ПО либо ключ доступа к удаленному узлу, позволяющий загрузить программный продукт. Именно этим способом большинство потребителей приобретают ПО - полученные элементы представляют собой носитель данных либо цифровой файл.

Многие программы готовы к сдаче "под ключ", причем разработанное ПО интегрировано в состав аппаратного обеспечения. Так, при покупке мобильного телефона непроизвольно приобретается сопровождающее ПО. Хотя и программы являются критическим системным компонентом, потребитель приобретает аппаратное обеспечение.

"Человеческий фактор" является критической частью многих программных продуктов. Корпоративные программные системы, используемые при осуществлении финансового менеджмента, контроля над работой предприятия, а также при разработке продукта, предполагают оказание консультационных услуг, чтобы быть "приобретенными" наряду с ПО, предназначенными для облегчения установки программ, проведения интеграции, а также адаптации к условиям специфической среды.

Несмотря на то, что продукты баз данных самым очевидным образом принадлежат к ИО, они выделены в отдельный класс в соответствии с ожиданиями, сопровождающими приобретение этого класса сложных программ. Программный продукт базы данных обычно приобретается в виде отдельного набора инструментов общего назначения, используемого в качестве дополнения к другим информационным системам. Многие "программные" продукты поставляются в виде внедренного пакета базы данных, входящего в состав продукта. Для заказчика в этой ситуации важно то, что он может не только приобрести "новое" ПО, но также получить доступ к продукту базы данных.

Практически всегда неотъемлемой частью поставляемого продукта является документация. В некоторых случаях она может прилагаться в виде книг и руководств, которые приобретаются в комплекте с готовой программой. При загрузке из Internet цифровые файлы могут включать файл "readme", а также, по возможности, полный комплект документации для данной копии программного обеспечения. При получении ПО из некоторых источников, таких как SourceForge (www.SourceForge.com), может отсутствовать какая-либо другая документация, отличная от исходного кода программ.

Процедуры либо деловые правила представляют собой финальный класс компонентов. В случае, когда заказчик приобретает системы и программное обеспечение, используемое для поддержки принятия решений, контроля оборудования, а также интеграции компонентов. При этом заказчику важно представлять готовые модули какой-либо процедуры. Обычно при заказной разработке привлекаются собственные силы либо консультанты, приглашенные из компании по разработке программного обеспечения. Эта область является "весьма серой", поэтому важно, чтобы менеджер проекта понимал назначение всех сборочных узлов проекта, причем еще на ранней стадии жизненного цикла разработки ПО. Особенно важно обращать внимание на те компоненты, которые могут вызвать недовольство пользователя.

Теперь, когда был определен третий набор классов предметной области, менеджер проекта может заносить сведения, как минимум, в две матрицы. После заполнения следующей матрицы

возможна идентификация критических компонентов продукта, как показано (см. табл.2). Данная матрица представляет собой таблицу, в которой устанавливается соответствие между классами компонента продукта с классами систем продукта. В этой матрице поддерживается информация о промежуточных компонентах определенного продукта, которая видоизменяется в зависимости от того, разрабатывается ли новое ПО, выполняется ли переработка существующих программ, реализуется ли интеграция компонентов продукта, либо выполняется "героическое" сопровождение наследственного продукта в составе портфеля компании. Помните о том, что Web-пример обозначается литерой «Х», а интеграция в составе фабрики - литерой «О».

Итак, данный этап - назначение классов компонентов продукта. Обзор этого набора классов также производится с точки зрения конечного пользователя. Класс насчитывает шесть составляющих: ПО (носитель данных либо цифровой файл), аппаратное обеспечение (встроенное ПО, например, в мобильных телефонах), человеческий фактор (оказание консультационных услуг), база данных (внедренный пакет БД), документация, процедуры (деловые правила для компонентов систем ППР, контроля оборудования и интеграции компонентов).

Матрица идентификации критических компонентов продукта для предыдущего примера имеет вид таблицы 4.2.

Таблица 4.2 - Идентификация критических компонентов продукта

Шаг 2. Идентификация критических компонентов продукта	ПО обеспечен	Аппаратн люди	База данны	Докум мент	Процедуры	
Разработка нового аппаратного/программного продукта	X o		0	0	X 0	0
Реинжиниринг существующего продукта	O		0	0	0	0
Интеграция компонентов	x				x	
Героическое сопровождение						

Например, вернемся к нашему программному продукту, основанному на использовании Web, который предназначен для регистрации вашей личной фильмотеки, размещенной на DVD-ROM. Этот продукт определяется в качестве нового продукта и интеграции компонента. В роли критических компонентов данного продукта выступают ПО и документация. Этот продукт основан на использовании Web и выполняется в окне броузера, запущенного на выполнение на оборудовании заказчика. При этом заказчик не доступна база данных, персонал разработчиков либо код процедур. Для него доступна лишь документация в виде инструкций на Web-страницах.

Другим примером продукта является расширение возможностей существующего программного продукта, причем в этом случае происходит переработка существующего продукта и разработка некоторого нового ПО. Основываясь на прогнозе относительно объема продаж разработанного продукта, заказчик может видеть все классы компонентов, за исключением аппаратного обеспечения. Он может ожидать поставку ПО, а также то, что инженер по эксплуатации выполнит установку и тестирование пригодности на предприятии заказчика.

Заказчик также может заказать разработку базы данных с целью поддержки состояния продукта, относящегося к классу реального времени, а также информацию о затратах и процедуры, предназначенные для реализации алгоритмов оптимизации. Документация будет иметь значение для инженеров компании, выполняющих установку ПО, а также заказчиков после того, как продукт будет принят.

Третья матрица, создаваемая менеджером проекта, предназначена для определения предметной области, которая связана с предметными областями продукта для производимых компонентов. Эта матрица представлена в таблице 4.3, в которой классы компонентов продукта сопоставляются с классами предметной области продукта. Эта матрица поддерживает сведения относительно промежуточных компонентов определенного продукта, основываясь на которых можно установить продукты, относящиеся к следующим предметным областям: деловой, индустриальной, режима реального времени, продукта с минимальной задержкой либо научной.

А теперь снова обратимся к нашим примерам: программный продукт, основанный на Web, предназначенный для регистрации в торговом клубе персональных фильмов на DVD-ROM, что позволит "оживить" их для заказчиков. В данном случае мы имеем дело с классом предметной области продукта с минимальной задержкой. При этом в качестве поставляемых продуктов выступают ПО и документация. Второй пример представляет собой расширение возможностей существующего продукта, т.е. интеграцию в производство. При этом затрагиваются четыре класса предметной области продукта: деловой, индустриальный, режима реального времени и научный. В качестве поставляемых компонентов выступают ПО, персонал, база данных, документация и процедуры.

Таблица 4.3 - Связь предметных областей продукта с компонентами

Шаг 3. Связь предметных областей продукта с компонентами	ПО	Аппаратные средства	Людские ресурсы	База данных	Документация	Процедуры
Потребитель	x				x	
бизнес	0		0	0	0	0
индустрия	0		0	0	0	0
Режим реального времени	0		0	0	0	0
Режим минимальной задержки	x				x	
Наука	0		0	0	0	0

Модели выбора проектов

Кроме определения предметной области необходимо также уметь выбирать проекты.

Процесс выбора проектов является ключевым деловым процессом. В этом процессе участвуют менеджеры проектов, менеджеры линий продуктов, руководители подразделений, а также все участники совместного бизнеса. Причем определение нужного количества последних,

вовлеченных в определение и выбор проекта, является критичным для успешного осуществления разработки проекта (тут возможны две схемы - частный бизнес и правительственные заказы) и является первым процессом при определении проектов. Следующий процесс- определение критериев выбора проекта. Здесь должна производиться оценка отдельных проектов либо групп проектов, а затем выбираться те из них, реализация которых обеспечивает достижение целей организации.

Процесс выбора проекта должен основываться на четко сформированной стратегии. На него большое влияние оказывает возможный риск (на этапе разработке и осуществлении продажи). А следовательно необходимо ответить на вопросы: «может ли быть разработан продукт?» и «может ли быть продан разработанный продукт?».

А для принятия решения по поводу выбора проектов потребуется ответить на целый ряд вопросов:

1. Какие организаторы совместного дела вовлечены в процесс принятия решения по поводу выбора проекта?

2. Какова степень ознакомления членов команды с технологией, применяемой при изготовлении продукта?

3. Насколько хорошо члены команды осведомлены относительно рынка данного продукта?

4. Какова степень сложности технологии, применяемой при создании продукта?

5. Насколько сложен продукт для объяснений конечному пользователю?

6. Каков опыт организации в разработке подобных типов продуктов?

7. Насколько велики предполагаемые трудозатраты, необходимые для разработки продукта?

8. Каким образом определяется жизненный цикл проекта?

9. Каким образом определяется жизненный цикл продукта?

10. Какова величина риска на этапе разработки продукта?

11. Возможный риск при разработке данного продукта?

12. Насколько высок приоритет данного проекта?

13. Является ли данный проект своего рода оперативной необходимостью?

14. Служит ли данный проект средством поддержки необходимого уровня конкуренции

15. Как соотносится данный проект с текущими проектами?

Ответы на данные вопросы образуют основу для выбора проекта. Как только он сделан, выполнение проекта переходит на следующий этап, когда назначается менеджер проекта и определяется собственник организации. Также должен быть разработан предварительный план менеджмента программного проекта и установлена среда разработки проекта.

Менеджмент проектов

При осуществлении менеджмента проектов интересным понятием является так называемый портфель проектов.

Портфель проектов - это группа проектов, относительно которых реализуется спонсорство и/или менеджмент со стороны организации. Группировка проектов производится в силу следующего ряда причин:

- принадлежность к одному и тому же семейству продуктов;
- устранение дефицита ресурсов;
- взаимная независимость между сборочными узлами проекта и промежуточными компонентами, проявляющимися на этапе разработки;
- наличие нескольких противоречащих целей организации;
- критерии выбора являются количественными или качественными, причем при их определении учитываются специфические цели организации;
- влияние политических интересов.

При выборе проектов могут применяться различные модели. Портфель проектов определяет стратегический процесс предметной области внутри организации. В моделях портфелей проектов учитываются относительные значения проектов, а также взаимодействие между ресурсами.

При выборе проектов могут применяться различные модели. Эти модели должны быть связаны со стратегией корпорации, в отличие от простой максимизации дохода либо продвижения единственного продукта. Благодаря группировке проектов в виде портфелей процесс разработки ориентируется на преобразование из хаотического в управляемое состояние с помощью стратегической концентрации на продукте.

Портфель проекта определяет стратегический процесс предметной области внутри организации. Этот процесс весьма специфичен для различных организаций.

В этом случае абсолютно необходим исполнительный менеджмент, определение направления, фокуса, а также распределение бюджета. Необходима регулярная корректировка модели портфеля проектов для гарантирования того, что адаптированная стратегия соответствует текущей деловой и экономической среде.

По сути, существует три класса моделей портфеля проектов:

- *модель экономической прибыли* (финансовые измерения применяются для определения нормы прибыли, чистой приведенной стоимости, маргинальной стоимости капитала, прибыли на инвестированный капитал, величины активов и инвестированного капитала, а также средневзвешенной стоимости капитала (Weighted average cost of capital, WACC),

- *модели «затраты-прибыль»* (применяется при сравнении альтернативных проектов, когда некоторые преимущества практически неощутимы (как для внутренних, так и для общедоступных проектов). Для каждого проекта определяются относительные преимущества, которые затем умножаются на величину затрат/относительных затрат. Затем происходит распределение проектов в соответствии с рейтингом "прибыль-затраты", когда лучшими считаются большие показатели рейтинга. Определения рейтингов («затраты-прибыль»),

- *модели исследования рынка* (применяются исключительно при разработке новых продуктов.

Неадекватный анализ рынка является основной причиной неудач при разработке продуктов. Используются некоторые методики, применяемые при исследовании рынка: выделенные группы, обзор рынка, панели потребителя и тестовый маркетинг.

Финансовые аспекты

И наконец, последний процесс предметной области, который следует осознать менеджеру проекта, - это учет финансовых аспектов проекта. Многие из методик, включенные в экономические модели выбора.

ROI (Return on Investment) - коэффициент окупаемости инвестиций. Данный показатель является одним из способов измерения эффективности ваших вложений.

Каждый владелец веб-сайта является инвестором, который инвестирует своё время, силы и деньги в развитие интернет-ресурса. И как любого инвестора, его прежде всего интересует, как быстро окупятся его инвестиции и сколько прибыли они ему принесут. Интересует отдача от ресурса. И количественно эту отдачу можно измерить при помощи этого коэффициента. Интернет-маркетинг тем и хорош, что здесь намного проще и точнее осуществляются измерения каких либо действий, в том числе и окупаемость инвестиций можно измерить значительно проще, чем в обычном маркетинге.

Например, при использовании рекламы, владелец веб-сайта наверняка захочет узнать, насколько она эффективна и стоит ли дальше проводить эту рекламную кампанию. Если выручка от продаж, полученная в результате этой рекламной кампании, больше, чем затраты на неё, то ROI в данном случае является положительным.

Расчет данного коэффициента осуществляется по следующей формуле:

$$ROI = (\text{Доход} - \text{Себестоимость}) / \text{Затраты}$$

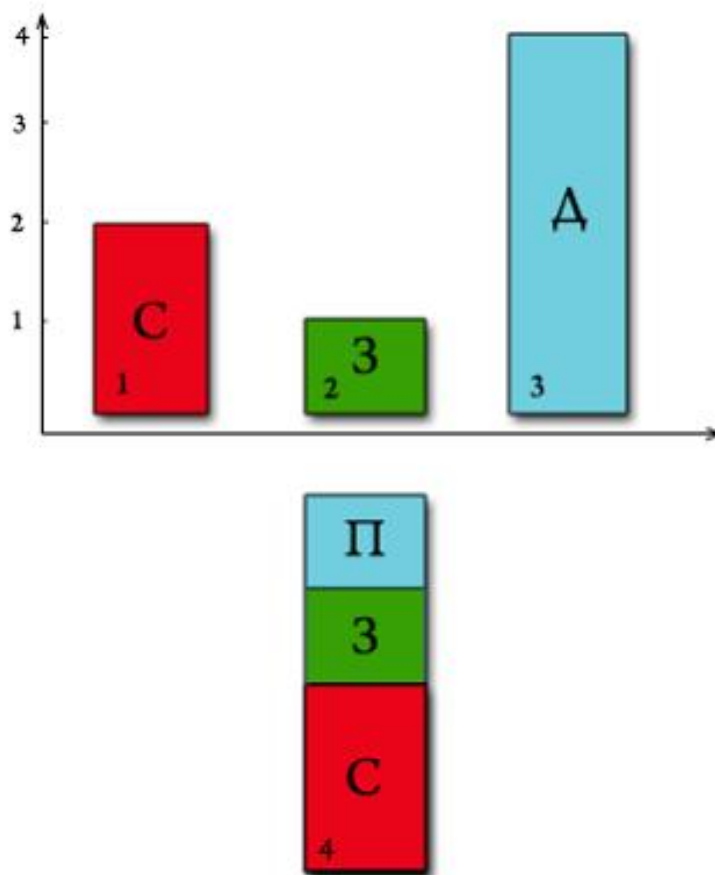
Если желаете видеть результат в процентах, то ещё надо умножить на 100.

Значение больше единицы ($ROI > 1$), говорит о финансовой выгоде от совершенных инвестиций.

Значение меньше единицы ($ROI < 1$) о том что совершённые вложения не являются выгодными для вас, они убыточны.

Если же $ROI = 1$, то это говорит о безубыточности инвестиций. Ваши затраты полностью окупаются.

Графически это выглядит так:



На данном примере можно увидеть, что если себестоимость нашего товара/услуги равна 2-ум ед. (столбец №1) и у нас вышло, что мы потратили на его рекламу, к примеру, в Яндекс.Директе, 1 ед. (столбец №2), а доход от этой рекламы получили равным 4-ём ед., то как можно заметить (столбец №4), мы оказались в плюсе, то есть ROI оказался больше 1 ($ROI > 1$). Можно произвести расчет, чтобы в этом убедиться.

$$ROI = (4 - 2) / 1 = 2 ,$$

$$ROI = 2$$

В данном случае реклама оказалась эффективной, мы заработали дополнительную денежную единицу. Прибыль равна 1-ой единице. Если бы ROI был меньше 1-го ($ROI < 1$), то это бы свидетельствовало о неэффективности данной рекламы.

DuPont Model (Модель Du Pont) - метод, который можно использовать для анализа доходности компании, используя традиционные инструменты для определения эффективности менеджмента. Для этого модель DuPont интегрирует компоненты Отчета о прибылях и убытках и Балансового отчета.

История возникновения модели финансового анализа Du Pont.

Модель DuPont финансового анализа была сформулирована F. Donaldson Brown, электротехником, который присоединился к финансовому отделу крупной химической компании в 1914. Немного позже, DuPont купила 23 процента акций General Motors Corp. и поставила перед Brown задачу привести в порядок запутанные финансы автомобильной компании. Это был, возможно, первый широкомасштабный проект реинжиниринга в США. Согласно Alfred Sloan, бывшему руководителю GM, во многом последующий успех GM принадлежал системе планирования и управления, разработанной Brown. Последующий успех приобрел известность модели DuPont в всех крупных корпорациях США. Она была доминирующей формой финансового анализа до 1970-х гг.

Рентабельность активов (Return on Assets) = Рентабельность чистой прибыли (Net Profit Margin) x Общая оборачиваемость активов (Total Assets Turnover) = Чистая операционная прибыль после налогов (Net Operating Profit After Taxes) / Продажи (Sales) x Средние нетто-продажи (Average Net Assets) / Продажи (Sales).

Наиболее широко применяется Финансовая модель DuPont, которая показана на рис.4.1 - 4.2

The DuPont Model

Net Profit Margin x Total Assets Turnover = Return on Assets

$$\frac{\text{Net Operating Profit After Taxes}}{\text{Sales}} \times \frac{\text{Sales}}{\text{Average Net Assets}} = \text{Return on Assets}$$

Measures of the effectiveness with which assets are used to produce revenues.

Measures of investments in working capital assets needed for sustaining ongoing operations.

Measures of investments in long-term, revenue-producing assets.

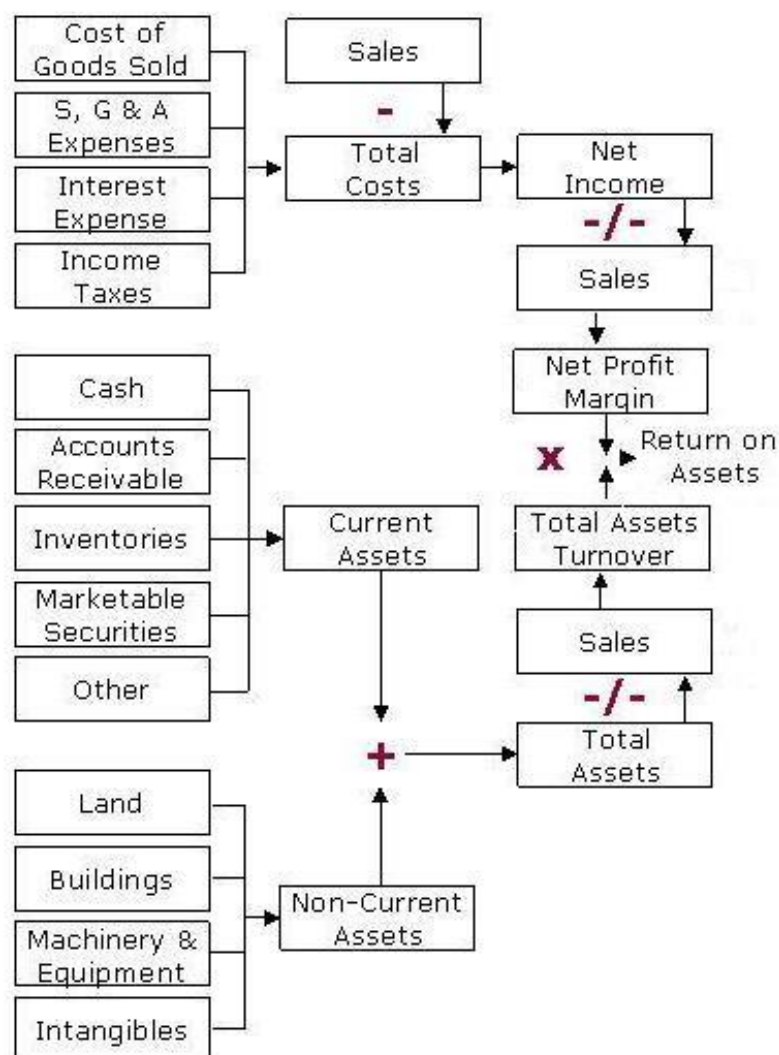


Рисунок 4.1 - Финансовая модель DuPont

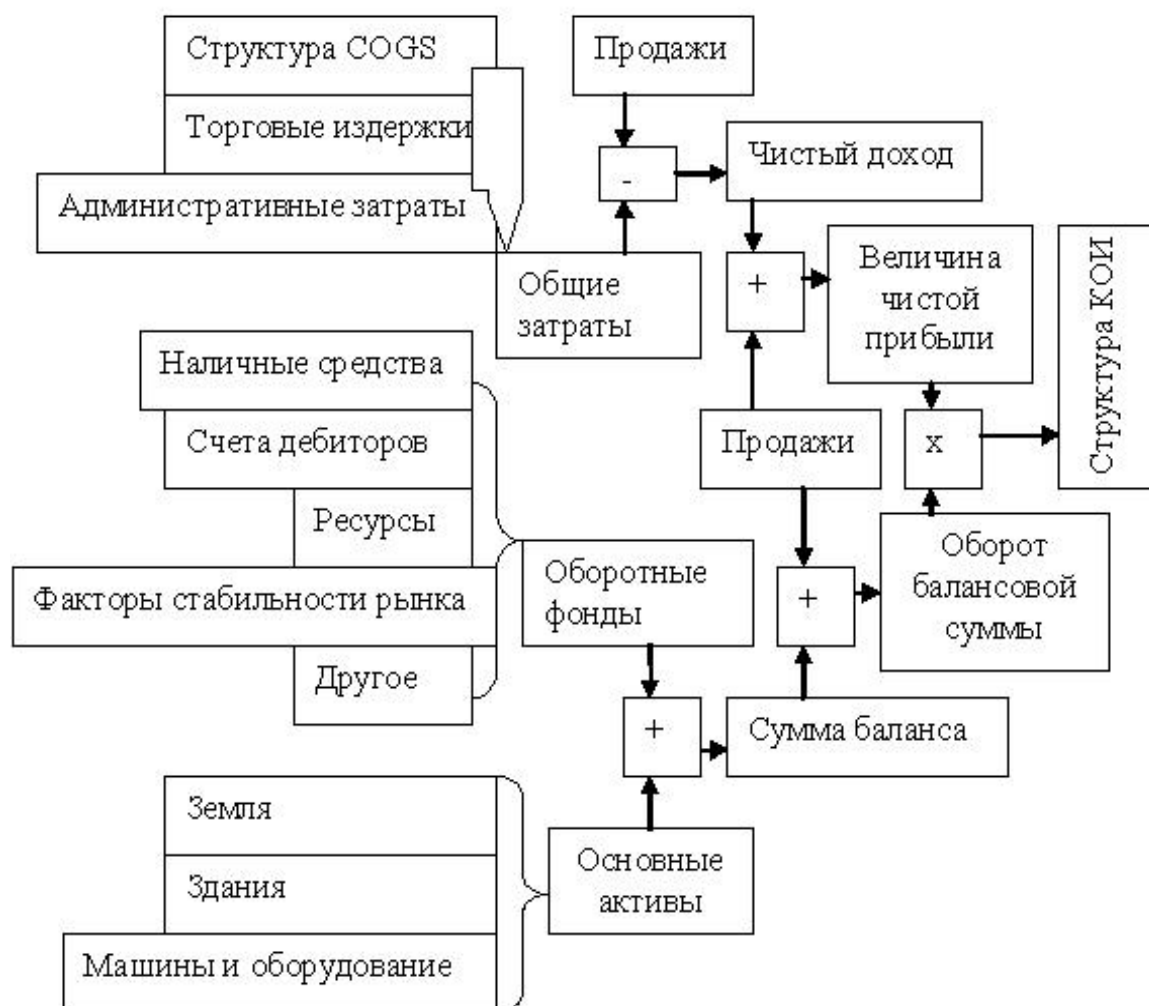


Рисунок 4.2 - Финансовая модель DuPont (*КОИ - коэффициент возврата инвестиций (Return of investment, ROI)).

Преимущества Модели Du Pont.

- Простота. Очень хороший инструмент для того, чтобы показать сотрудникам как они могут повлиять на результаты компании.
- Можно легко связать с планами компенсаций. Может быть использована для того, чтобы убедить менеджмент в необходимости определенных действий для повышения профессионального уровня функций закупок или сбыта.

Ограничения анализа DuPont. Недостатки.

- Основан на показателях бухгалтерского учета, которые не всегда надежны.
- Не включает стоимость капитала.

В общем случае каждый процесс выбора проекта либо решение относительно критерия анализа портфеля проектов, оказывающие влияние на цены продукта, стоимость единицы продукта, объем либо эффективность, будет определять величину прибыли или коэффициент оборота. И любое решение, оказывающее влияние на величину и вид долга, а также на величину собственных средств, будет оказывать влияние на финансовую структуру в той же степени, как это происходит с величиной стоимости.

Финансовые аналитики предметной области и фининспекторы во многих организациях применяют подобную методику анализа для принятия решений относительно финансирования проектов.

Несомненно, что для менеджеров проектов и специалистов по маркетингу продукта весьма важным является правильное понимание основных финансовых концепций, без чего была бы невозможной квалифицированная помощь заказчикам, которых тоже можно отнести к организаторам совместного дела, и успешная реализация выполняемых проектов и разрабатываемых продуктов. Использование финансового анализа в процессе выбора проектов предоставляет универсальное средство оценки - деньги.

Практика

Вопросы для самопроверки

1. Классы предметных областей при разработке программного продукта.
2. Назовите класс, в который входят программные продукты для личного применения.
3. Назовите класс, в который входят программные продукты определенной предметной области.
4. Назовите класс, в который входят программные продукты, нацеленные на определенную предметную область, для удовлетворения специфических нужд производства.
5. Назовите класс, в который входят программные продукты, используемые с целью управления процессами, которые имеют определенный и ограниченный во времени бюджет.
6. Назовите класс, в который входят программные продукты, разрабатываемые с учетом ограниченного времени.
7. Назовите класс, в который входят программные продукты, основанные на использовании математических моделей, имитирующие объекты реального мира.
8. Что такое портфель проектов?
9. Три класса моделей портфеля проектов.
10. Характеристики модели экономической прибыли.
11. Характеристики модели «затраты - прибыль».
12. Характеристики модели «исследование рынка».
13. Финансова модель DuPoint.
14. Коэффициент возврата инвестиций. Характеристика и порядок нахождения.

Вывод к разделу 4 - Управление процессами предметной области

В разделе рассматривались вопросы управления процессами предметной области, классы предметных областей, модели выбора проектов, а также финансовые аспекты.

Перечень ссылок

Источники, использованные в материалах

Американский национальный стандарт ANSI/PMI 99-001-2004. Руководство к Своду знаний по управлению проектами. Введ. 2004.- Третье издание. (Руководство PMBOOK®). 401с.

Мари Кантор. Управление программными проектами. Практическое руководство по разработке успешного программного обеспечения . СПб. Вильямс. 2002. -642с.

Управление программными проектами. Достижение оптимального качества при минимуме затрат. Роберт, Т. Фатрелл, Дональд Ф. Шафер, Линда И. Шафер / М-СПб-К. Вильямс. 2003. -1118с.

Уокер Ройс. Управление проектами по созданию программного обеспечения . М. Лори. 2002. -450с.

Элейн Маркел. Microsoft Project 2002. Библия пользователя. М. Диалектика. 2003. -880с.

Microsoft Project 2003 course certification materials. Trainer kit. [Электронный ресурс] Режим доступа: <http://www.cheltenhamcourseware.com/>