

## **1. Основные понятия и особенности промышленного рынка программных продуктов.**

В настоящее время можно констатировать, что рынок программных продуктов достаточно сформировался, так как существует реальный спрос на ПО у конкретных потребителей (и этот спрос постоянно растет) и есть производители, которые имеют конечные продукты, ориентированные на удовлетворение этого спроса. Программный продукт вступает в хозяйственный оборот как товар только в случае фиксации его на материальном носителе, однако обладание материальным носителем информации не делает его приобретателя уникальным собственником информации.

Проекты по созданию ПО промышленными методами должны начинаться с прогнозирования, анализа и технико-экономического обоснования (ТЭО). Экономическое прогнозирование проектов в программной инженерии должно опираться на использование достаточно точных методов для оценивания экономических характеристик производства сложных программных продуктов.

**Рынок** – это система экономических отношений, складывающихся в процессе производства, обращения и распределения товаров. Это определенный способ согласования деятельности участников общественного производства; механизм, соединяющий производителя и потребителя на основе спроса и предложения; это саморегулирующаяся и самонастраивающаяся на спрос система

**Субъектами рынка** являются продавцы и покупатели. Субъекты взаимодействуют на рынке, образуя взаимосвязанный «поток» купли-продажи

**Объектами рынка** являются товары (услуги) и деньги

**Товар** – любой продукт производственно-экономической деятельности в материально-вещественной форме

**Услуги** – итог непосредственного взаимодействия поставщика и потребителя и внутренней деятельности поставщика по удовлетворению потребности потребителя

**Программный продукт** - это самостоятельное отчуждаемое произведение, представляющее собой публикацию текста программы на языке программирования или в виде исполняемого кода (является предметом интеллектуального труда, охраняется авторским правом, вовлечение ПП в хозяйственный оборот происходит в процессе коммерциализации (купли-продажи, переуступки прав собственности) и капитализации (постановки на баланс, инвестирования в основной капитал)).

**Программный продукт** – совокупность записанных на носителях данных программных компонентов, являющихся результатом промышленного производства, предназначенных для поставки, передачи или продажи пользователю, снабженных технической документацией, инструкциями по обучению пользователей, а также гарантийными обязательствами по сопровождению и обслуживанию

**Программный модуль** – отдельно компилируемая часть программного кода

**Программный компонент** – это автономный элемент программного обеспечения, предназначенный для многократного использования, который может распространяться для использования в других программах в виде скомпилированного кода

**Программный комплекс** (программная система) – набор взаимодействующих программ, согласованных по функциям и форматам, имеющих единообразные, точно определенные интерфейсы и составляющих полное средство для решения больших задач

**Коробочный программный продукт** – программное обеспечение, предназначенное для неопределенного круга покупателей и поставляемое на условиях «как есть» со стандартными для всех покупателей функциями

**Особенности программного продукта как товара**

- 1) Нематериальная природа существования (ПП нельзя увидеть в процессе конструирования и, следовательно, оперативно повлиять на его реализацию)
- 2) Возможность неоднократной продажи и участия одновременно в нескольких сделках
- 3) Сохранение свойств продукта в процессе использования (не исчезает и не изнашивается)
- 4) Создание продукта в условиях повышенного риска, невозможность точного оценивания временных и финансовых параметров разработки, обусловленная творческим характером труда в процессе интеллектуальной деятельности
- 5) Относительно низкие затраты на тиражирование по сравнению с высокими затратами на разработку, обусловленные ничтожно малой стоимостью производственных операций на создание копий ПП

**Участники рынка программных продуктов**

**Разработчики** (правообладатели)

Факторы, определяющие конкурентоспособность разработчиков и их положение на рынке: (цена на программную продукцию или услугу, качество продукции с точки зрения удовлетворения существующих потребностей пользователей, отличительные особенности продукции, побуждающие пользователя приобретать именно данный программный продукт, гибкость производителя, связанная со способностью реагировать на просьбы покупателя по адаптации либо доработке программного продукта, время (сроки) реагирования производителя на потребности покупателя (время адаптации и внедрения ПП, продолжительность обучения пользователей, период гарантийного сопровождения, временные условия по модернизации и поставке новых версий и пр.)

Причины, препятствующие развитию рынка со стороны разработчиков

- 1) Отсутствие начального капитала на развитие фирмы, наработку требуемых заделов, приобретение лицензионного ПО
- 2) Ориентация производителей на мелкосерийное производство программных продуктов, разрабатываемых обычно под конкретный заказ

- 3) Слабое представление о существующем рынке конкурирующих программных продуктов
- 4) Высокая доля фиксированных затрат в структуре издержек и, как следствие, высокие цены на создаваемые программные продукты
- 5) Использование при разработке пиратских инструментальных программных средств, не позволяющее производителю открыто рекламировать свои продукты, участвовать в выставках и пр.
- 6) Отсутствие эффективных программных средств защиты от копирования, а также экономических и юридических механизмов, препятствующих этим процессам
- 7) Отсутствие опыта по представлению ПП в виде законченного продукта и организации маркетинга по его распространению
- 8) Отсутствие профессиональных менеджеров по продвижению ПП на рынок
- 9) Незнание или несоблюдение отечественных и международных стандартов по управлению жизненным циклом, качеством и документированием П

### **Потребители**

Экономические интересы потребителей, в качестве которых могут выступать государственные (муниципальные) структуры, а также юридические и физические лица, заключаются в приобретении рыночных преимуществ и доходов от использования программных продуктов, либо в удовлетворении в той или иной мере личных потребностей

Проблемы, с которыми сталкиваются потребители при приобретении ПП:

- 1) отсутствие сформированного спроса на программное обеспечение и четкого представления о технологии использования программных продуктов в практической деятельности
- 2) низкая информированность потребителей о рынке предлагаемого ПО
- 3) неспособность четко сформулировать требования к приобретаемым программным продуктам, превышение значимости ценового критерия по сравнению с критерием качества при выборе приобретаемых ПП
- 4) несоответствие между высокими ценами на программное обеспечение и сиюминутными «выгодами» от его использования
- 5) незнание, а чаще игнорирование экономических и нормативно-правовых механизмов цивилизованной работы на рынке (ментальность отечественного потребителя не расценивает факт использования нелегальных копий как хищение собственности производителя)

### **Государство**

▶ Осуществляет регулирование отношений, возникающих по поводу использования программного обеспечения, посредством создания экономических, организационных и нормативно-правовых механизмов, обеспечивающих цивилизованное взаимодействие участников рынка ПП

▶ Интересы государства заключаются, в первую очередь, в получении выгод от надлежащей охраны прав интеллектуальной собственности и повышении эффективности ее использования в интересах развития отраслей экономики

Недостатки государственной политики в сфере регулирования рынка ПП:

- 1) имеющиеся законы об охране авторских прав, защите интеллектуальной собственности, информации, информатизации и защите информации практически не работают, так как нет эффективных механизмов их конкретного применения
- 2) существующая система нормативных документов (ГОСТов), регламентирующих жизненный цикл проектирования и документирования программных средств морально устарела и носит рекомендательный характер
- 3) сертификация как институт, обязывающий создавать программные продукты с определенными параметрами качества, существует преимущественно в добровольной форме и не носит масштабного характера

### **Посредники**

В качестве посредников выступают фирмы, берущие на себя функции маркетинга и распространения программного продукта

Их роль заключается в принятии продукта от разработчика, оценке его готовности к выводу на рынок, осуществлении мероприятий по продвижению ПП и доведению его до конечного пользователя

В рамках этой деятельности разделение функций между разработчиками и посредниками является наиболее эффективным механизмом оптимизации усилий по доставке ПП пользователю, так как в этом случае с разработчика снимается нагрузка по исследованию рынка, рекламе, доставке и пр.

### **Партнеры**

В качестве партнеров могут выступать фирмы, производящие аналогичную продукцию и ориентированные на один и тот же сегмент рынка

Направления сотрудничества:

▶ интеграция в сфере приобретения и совместного использования средств производства программного продукта

▶ освоение каналов распространения и активизация маркетинговой деятельности

### **Рынки ПП**

Промышленный (Горизонтальный рынок ,Вертикальный рынок)

Потребительский

**Промышленный** рынок (рынок корпоративных продаж)

Характеризуется наличием множества товаров промышленного назначения, которые могут использоваться как самостоятельно, так и в составе других товаров или услуг, продаваемых, сдаваемых в аренду или поставляемых другим потребителям

Количество участников рынка невелико. Производители и покупатели обладают высокими профессиональными навыками в области ПП, представленных на рынке, а также навыками коммерческой работы

Объектами переговоров и подписания контрактов с каждым конкретным заказчиком являются такие условия сделки как цена, качество, условия поставки и оплаты

**Горизонтальный** рынок представляет собой совокупность различных изделий и/или услуг общего назначения и состоит из широкого спектра отраслей.

Субъекты (пользователи) горизонтального рынка имеют потребности в решении проблем общего характера. ПП для горизонтального рынка создаются с высокой степенью универсальности для охвата самого широкого спектра потребителей.

Разработка ПО под заказ не предусматривается, а производится выбор оптимального варианта среди конкурирующих между собой продуктов

**Вертикальный/отраслевой** рынок представлен продукцией конкретного сегмента рынка, охватывающего организации и предприятия определенного профиля деятельности. Рынок структурируется, как правило, на основе принятых в статистической отчетности групп отраслей экономики согласно Общероссийского классификатора видов экономической деятельности (ОКВЭД). ПП для вертикальных рынков являются достаточно специализированными, чтобы максимально соответствовать требованиям компаний выбранной отрасли или подотрасли, и разрабатываются, как правило, под заказ

### **Потребительский рынок**

Характеризуется наличием товаров и услуг для личного потребления, покупаемых или приобретаемых другим способом отдельными физическими лицами

Количество покупателей в данном сегменте рынка достаточно велико и все они рассматриваются как единая генеральная совокупность

Чаще всего покупатель не осведомлен об истинных характеристиках товара, а больше доверяет рекламе или продавцу- консультанту

Анализ поведения пользователей проводится на представительной выборке. По результатам анализа выявляется среднестатистический потребитель

## **2. Программный проект как основа разработки программного обеспечения.**

**Инициирование программного проекта: концепция проекта. Структурная декомпозиция работ (WBS) как основа планирования проекта.**

**Планирование программного проекта: формирование рабочего графика, ресурсное планирование.**

**ПРОЕКТ** - это ограниченное по времени, целенаправленное изменение отдельной системы с установленными требованиями к качеству результатов, возможными рамками расхода средств и ресурсов и специфической организацией

**ПРОГРАММНЫЙ ПРОЕКТ** — это комплекс взаимосвязанных работ, выполняемых командой проекта с целью получения уникального программного продукта или услуги в течение заданного периода при установленном бюджете и потребляемых в ходе реализации проекта ресурсах в условиях повышенного риска, требующих специфического управления

### **Признаки проекта**

- 1) Наличие цели и направленность на достижение конкретного конечного результата
- 2) Неповторимость / уникальность
- 3) Ограниченность во времени
- 4) Реализация в условиях ограничения требуемых ресурсов
- 5) Последовательная разработка
- 6) Структурная сложность и высокая степень неопределенности
- 7) Специфическая организация проекта

Цели программного проекта следует определять в виде желаемого результата, достигаемого командой проекта, при его успешной реализации

При формулировании **целей** проекта необходимо обеспечить:

- 1) *конкретность* — четкость и однозначность понимания результатов;
- 2) *измеримость* — возможность оценивания степени достижения результата, выраженного как в качественной, так и в количественной форме;
- 3) *реальность* — соизмерение возможностей команды проекта с факторами влияния внешней среды;
- 4) *согласованность* — единство мнений всех участников проекта в понимании конечного результата и координация действий по его достижению;
- 5) *ограниченность по срокам* — установление интервалов времени, по истечении которых необходимо оценивать промежуточные результаты проекта и определять степень достижения конечного результата

При описании **желаемого результата** программного проекта необходимо определять:

- 1) конкретные бизнес-выгоды, которые получит заказчик по завершении проекта;
- 2) функционал продукта (услуги), получаемый по окончании проекта;
- 3) краткое описание и при необходимости ключевые свойства и/или характеристики функционала

### **Жизненный цикл проекта**

**Фаза концепции** - формулирование результатов проекта, сбор исходных данных и предварительное обследование, разработка альтернативных вариантов реализации и сравнительная оценка альтернатив, выработка концепции реализации проекта, ее апробация и экспертиза

**Фаза планирования** - разработка плана проекта, назначение руководителя, формирование команды, заключение договоров с поставщиками и подрядчиками

**Фаза реализации** - выполнение работ проекта, их координация и материально-техническое обеспечение, оперативный контроль и регулирование основных показателей проекта, мотивация и стимулирование команды проекта

**Фаза завершения** - эксплуатационные испытания конечного продукта, подготовка кадров для его эксплуатации, подведение итогов, оформление документации и закрытие проекта

**Участники проекта** (Инициатор, Владелец (иногда это клиент или заказчик), Инвесторы, Исполнитель (команда проекта), Поставщики, подрядчики, потребители, Менеджер, Другие участники (спонсоры, консультанты, конкуренты, лицензиары, общественные группы и население))

## **Фаза инициации**

**Описание ценности.** Опишите проблему и возможности, достигаемые бизнесом цели и другие предположения о том, что будет происходить. Бывает полезно показать на контрасте варианты того, что может быть достигнуто в результате реализации проекта (возможность), и что будет если его вообще не начинать (риски, проблемы)

**Демонстрация альтернатив.** Любую проблему можно решить разными способами. Покажите, почему ваше решение является оптимальным, почему оно принесет больше ценности и/или несет в себе меньше рисков

**Требования к проекту.** Опишите, как вы видите реализацию проекта. Что в нем должно быть, и чего быть не должно

**План реализации.** Ответьте на вопрос, как проект должен реализоваться, укажите основные этапы реализации, определите период реализации и выполните укрупненный анализ финансовых потребностей

**Концепция** (от лат. conceptio — понимание, система), определенный способ понимания, трактовки какого-либо предмета, явления, процесса, основная точка зрения на предмет и др., руководящая идея для их систематического освещения  
Концепция проекта разрабатывается на основе анализа потребностей бизнеса  
Она необходима для согласования единого видения целей, задач и результатов всеми участниками

проекта

Концепция определяет *что и зачем* делается в проекте

Концепция не затрагивает развитие проекта в глубину, а дает максимально широкий охват проблем и определяет долгосрочные перспективы продукта

Концепция проекта - это ключевой документ, который используется для принятия решений в ходе его реализации, а также обеспечивает возможность подтверждения результата на фазе приемки

## **Структура концепции**

- 1) Название проекта
- 2) Цели проекта
- 3) Результаты проекта
- 4) Допущения и ограничения
- 5) Ключевые участники и заинтересованные стороны

- 6) Ресурсы проекта
- 7) Сроки
- 8) Риски
- 9) Критерии приемки
- 10) Обоснование полезности проекта

**Цели проекта** должны отвечать на вопрос, зачем данный проект нужен. Цели проекта должны описывать бизнес-потребности и задачи, которые решаются в результате реализации проекта

Требования к целям:

- 1) Значимость (направленность на достижение стратегических целей компании)
- 2) Конкретность (учет специфических особенностей конкретного проекта)
- 3) Измеримость (наличие проверяемых количественных оценок)
- 4) Реалистичность (достижимость)

**Результаты проекта** отвечают на вопрос, **что** должно быть получено после его завершения. Результаты проекта должны определять: (Какие именно бизнес-выгоды получит заказчик в результате проекта? Что конкретно (какой продукт или услуга) станет итогом его реализации?)

По отношению к результатам должны быть приведены высокоуровневые требования, либо краткое описание и при необходимости ключевые свойства и/или характеристики продукта/услуги

**Допущения**, как правило, тесно связаны с управлением рисками: в разработке ПО часто приходится формулировать риски в виде допущений. Например, оценивая проект разработки и внедрения по схеме с фиксированной ценой, мы должны записать в допущения предположение о том, что стоимость лицензий на стороннее ПО не изменится до завершения проекта

**Ограничения** – это те ситуации, которые сокращают возможности проектной команды в выборе решений. В частности к ним относятся специфические нормативные требования (например, обязательная сертификация продукта на соответствие определенным стандартам), специфические технические требования (например, разработка под заданную программно-аппаратную платформу), специфические требования к защите информации и др.

К ключевым участникам программного проекта, как правило, относятся:

- 1) *Инвестор или спонсор проекта* - лицо или группа лиц, предоставляющая финансовые ресурсы на различных условиях
- 2) *Заказчик проекта* - лицо или организация, которые будут использовать продукт, услугу или результат проекта. Следует учитывать, что заказчик и инвестор проекта не всегда совпадают
- 3) *Пользователи* результатов проекта
- 4) *Куратор проекта* - представитель исполнителя, уполномоченный принимать решение о выделении ресурсов и изменениях в проекте
- 5) *Руководитель проекта* - представитель исполнителя, ответственный за реализацию проекта в срок, в пределах бюджета и с заданным качеством



6) *Соисполнители проекта. Субподрядчики и поставщики*

**Ресурсы** необходимые для реализации программного проекта: (людские ресурсы и требования к квалификации персонала; оборудование, услуги, расходные материалы, лицензии на ПО ; бюджет проекта, план расходов и, при необходимости, предполагаемых доходов проекта с разбивкой по статьям и фазам/этапам проекта )

**Риск** - неопределенное событие или условие, наступление которого отрицательно или положительно сказывается на целях проекта

На этапе инициации, когда нет необходимых данных для проведения детального анализа, часто приходится ограничиваться качественной оценкой общего уровня рисков: низкий, средний, высокий

**Критерии приемки** должны определять числовые значения характеристик системы, которые должны быть продемонстрированы по результатам приемосдаточных испытаний или опытной эксплуатации и однозначно свидетельствовать о достижении целей проекта

**Сущность планирования:** уточнение целей проекта и способов их достижения на основе формирования комплекса работ, которые должны быть выполнены, определения способов выполнения этих работ, увязки необходимых для этого ресурсов и согласования действий участников проекта

**Цель планирования:** выбор и определение способов создания и совершенствования программного обеспечения, которые способны удовлетворить требованиям технического задания, спецификаций и контракта, а также обеспечить уровень качества, соответствующий заданным требованиям

**Планирование** - это определение ясных и точных задач, служащих для достижения поставленной цели

**План** – это модель действий и прогноз состояния проекта и его окружения, это определение того, каким образом требования к проекту будут трансформированы в продукт с учетом временных и финансовых ограничений

**Параметры** проекта, определяемые на этапе планирования

- 1) продолжительность по каждому из контролируемых элементов проекта
- 2) потребность в трудовых, материально-технических и финансовых ресурсах
- 3) сроки поставки сырья, материалов, комплектующих и технологического оборудования
- 4) сроки и объемы привлечения проектных, консалтинговых и других организаций

**Принципы** планирования:

**Целенаправленность.** Планирование рассматривается как процесс развертывания главной цели проекта в иерархическую последовательность задач, действий и работ с определением порядка их выполнения

**Комплексность** планирования означает полный охват научных, проектных, организационных, производственных и других мероприятий и работ, направленных на достижение результатов проекта

**Сбалансированность по ресурсам** означает, что планы не содержат работ и задач, не обеспеченных необходимыми ресурсами

**Системность** планирования предполагает учет влияния на проект факторов его окружения, т.е. рассмотрение проекта как целостной системы с учетом взаимосвязей как внутри, так и вне его

**Гибкость** планирования предполагает способность системы прогнозировать и адекватно реагировать на возможные изменения внешних факторов и их последствия

**Многофункциональность** планирования означает обязательное планирование по всем функциям управления проектом

**Оптимальность** планирования предполагает способность системы формировать не просто приемлемые (т.е. допустимые с точки зрения принятых ограничений) планы, а лучшие планы по выбранным критериям

**Непротиворечивость** планирования обеспечивается преемственностью и взаимоувязанностью всех плановых решений

**Непрерывность** планирования заключается в отслеживании, контроле а при необходимости актуализации плановых решений

**Стабильность** планирования обеспечивается неизменностью основных целей проекта, его жизнеспособностью

Основные **процессы** планирования

- 1) планирование содержания проекта
- 2) определение основных этапов реализации проекта, декомпозиция их на более мелкие и управляемые элементы
- 3) формирование списка конкретных работ, которые обеспечивают достижение целей проекта
- 4) установление последовательности работ, определение и документирование технологических зависимостей и ограничений на работы
- 5) оценка продолжительности работ, трудозатрат и ресурсов, необходимых для выполнения отдельных работ
- 6) планирование ресурсов, определение того, какие ресурсы (люди, оборудование, материалы) и в каких количествах потребуются для выполнения работ проекта, уточнение сроков выполнения работ с учетом ограниченности ресурсов
- 7) составление сметы и оценка стоимости ресурсов, необходимых для выполнения работ проекта
- 8) составление бюджета, привязка сметных затрат к конкретным видам деятельности
- 9) разработка плана проекта, сбор результатов остальных процессов планирования и их объединение в общий документ

### **Структурная декомпозиция работ (WBS, work breakdown structure )**

Структурная декомпозиция работ представляет собой инструмент, применяемый для документирования всех рабочих операций, которые должны быть выполнены при разработке и поставке ПО

Структура WBS консолидирует информацию из различных источников, организуя ее с использованием единого формата, удобного при планировании, оценивании и отслеживании хода реализации проекта

На основе структуры WBS разрабатывается график выполнения проекта

Схема WBS обеспечивает членам команды понимание общих целей и задач по проекту

Схема WBS определяет работы, необходимые для завершения всего проекта.

Она гарантирует, что в состав проекта войдут все работы, выполнение которых позволит успешно его завершить

Схема WBS определяет сроки получения рабочих продуктов. Так как она разделена на низкоуровневые задачи, каждая из которых имеет дату начала и окончания, все участники проекта будут точно знать сроки завершения определенных работ

Если возникнет потребность добавить в существующий проект дополнительные функциональные возможности, это может быть отражено через изменение схемы WBS

Она дает возможность определить на соответствующем уровне детализации плана вехи (ключевые результаты), которые будут играть роль контрольных точек по проекту

На основе схемы WBS распределяется ответственность за достижение целей проекта между его исполнителями

Схема WBS позволяет создать удобную, соответствующую целям проекта структуру отчетности

### **Основания декомпозиции** при построении схемы WBS

- 1) компоненты продукта (объекта, услуги, направления деятельности), получаемого в результате реализации проекта
- 2) процессные или функциональные элементы деятельности организации, реализующей проект
- 3) этапы жизненного цикла проекта, основные фазы
- 4) подразделения организационной структуры
- 5) географическое размещение для пространственно распределенных проектов

### **Основные этапы** построения WBS

- 1) на основе информации, заложенной в концепции проекта, проводится последовательная декомпозиция по заданным основаниям (критериям) работ проекта. Этот процесс продолжается до тех пор, пока все значимые работы, пакеты работ и отдельные задания не будут выделены и идентифицированы в такой степени и таким образом, чтобы они могли планироваться, для них можно было определять бюджет и составлять расписание, выполнять функции управления и контроля
- 2) каждому элементу декомпозиции присваивается уникальный идентификатор, при этом названия элементов на каждом уровне отражают критерии разбиения работ
- 3) для каждой работы, пакета работ, части проекта, выделенных таким образом, определяются имеющие к ним отношение данные (продолжительность и трудозатраты, ответственные исполнители, бюджет и затраты, оборудование и материалы и т.д.)

- 4) по каждой выделенной работе, пакету работ, части проекта проводится критический анализ с их исполнителями (членами команды проекта, менеджерами и другими участниками) для подтверждения правильности выполненной структуризации работ и их параметров

---

#### Возможные **ошибки планирования** проектов

- 1) пропуск стадии структуризации проекта и переход непосредственно к поиску и решению текущих, оперативных проблем проекта
- 2) непонимание того, что схема WBS, которая положена в основу разработки плана, должна охватывать весь проект (обычно недостаточное внимание уделяется начальной и конечной фазам проекта, работе функциональных, обеспечивающих подразделений)
- 3) излишняя или недостаточная детализация
- 4) повторение элементов структуры

#### **Традиционный подход к построению схемы WBS**

- 1) Выделение основных поставляемых частей проекта
- 2) Декомпозиция выделенных частей до уровня действий, дающих представление о технических требованиях каждой части
- 3) Последующая декомпозиция каждого действия до уровня задач (Элементарным уровнем деления действий в схеме WBS рекомендуется считать уровень задач, выполняемых одним человеком за время от 1 до 10 дней)
- 4) Определение длительности каждой задачи либо на основе накопленных данных по реализованным проектам, либо с привлечением экспертного мнения членов команды разработчиков, ответственных за выполнение задач
- 5) Назначение ответственных за выполнение каждой задачи

#### **Проблемы традиционной схемы WBS при разработке ПО**

- 1) традиционные декомпозиции работ создаются преждевременно на основе проектных решений по разработке продукта
- 2) традиционные декомпозиции детализируются, планируются и финансируются либо слишком подробно, либо недостаточно подробно
- 3) традиционные декомпозиции работ специфичны, для каждого проекта, поэтому сравнение разных проектов обычно оказывается затруднительным или невозможным

#### **Построение эволюционных WBS**

Элементами WBS первого уровня являются рабочие процессы (управление проектом, создание рабочей среды, управление требованиями, проектирование, реализация, оценка и внедрение)

Элементы второго уровня определяются для каждой стадии жизненного цикла

Элементы третьего уровня определяются для выделения видов деятельности, в результате которых производятся рабочие продукты каждой стадии. Эти элементы могут либо образовывать самый нижний уровень в иерархии, который позволяет вычислить стоимость отдельного вида рабочих продуктов для данной стадии, либо разбиваться дальше на несколько задач более низких уровней,

которые, взятые вместе, обеспечивают получение одного вида рабочих продуктов

Факторы, учитываемые при построении эволюционирующих WBS Масштаб.

Более масштабные проекты будут иметь больше уровней и подструктур

Организационная структура. Проекты, где задействованы субподрядчики или участвует множество различных организаций, могут иметь ограничения, которые приведут к необходимости иного распределения работ

Объем разработок на заказ. В зависимости от характера проекта в рабочих процессах управления требованиями, проектирования и реализации внимание может уделяться разным аспектам

Бизнес-контекст. Проекты, выполняемые на контрактной основе, требуют более совершенного управления и оценки. Проекты, в которых разрабатываются коммерческие продукты для продажи широкому кругу потребителей, могут потребовать более совершенных структур для внедрения

Предшествующий опыт. Очень немногие проекты начинаются с чистого листа.

Большинство из них разрабатывается либо как новые поколения существующих систем (с устоявшейся WBS), либо в контексте существующих организационных стандартов (с предопределенным построением WBS). Важно подстроиться под эти ограничения для гарантии, что новый проект сумеет воспользоваться имеющимся опытом и достигнутым уровнем

производительности

### **Разработка схемы WBS методом сверху вниз**

- 1) Разработка схемы WBS начинается с самого верхнего элемента (поставляемого программного продукта), после чего идентифицируются большие рабочие продукты
- 2) Руководитель программного проекта вырабатывает общую оценку размера проекта, процесса, среды, персонала и требуемого качества
- 3) Производится приблизительная оценка общих трудозатрат и сроков с использованием модели оценки стоимости
- 4) Менеджер проекта детализирует эту приблизительную оценку трудозатрат на верхнем уровне WBS, используя рекомендации, аналогичные приведенным в таблице 1. На этом этапе детализируются также сроки путем установления основных контрольных точек, и распределяются необходимые трудозатраты в соответствии с квалификацией персонала
- 5) В этот момент на руководителей отдельных направлений проекта возлагается ответственность за разбиение каждого из элементов WBS на элементы более низких уровней, учитывающее их расположение на верхнем уровне, штатное расписание и даты основных контрольных точек в качестве ограничений

### **Планирование проекта методом снизу вверх**

Элементы WBS самого нижнего уровня прорабатываются в виде отдельных заданий, сроки и бюджеты для которых приблизительно оцениваются членами команды, ответственными за данный элемент WBS

Приблизительные оценки суммируются и объединяются в бюджеты и

контрольные точки более высоких уровней

Производится сравнение с бюджетами и контрольными сроками, разработанными сверху вниз. Определяются самые значительные расхождения и делаются уточнения для того, чтобы достигнуть общего согласования между оценками, выполненными сверху вниз и снизу вверх

Для создания **структуры WBS при разработке ПО** необходимо:

- 1) Идентифицировать работы, связанные с созданием программного продукта, отделяя их от работ, связанных с аппаратным обеспечением и от рабочих процессов
- 2) Найти структуру WBS для произвольной системы высшего уровня, отделяя ПО от других систем и компонентов
- 3) Определить программную архитектуру WBS, идентифицируя все ее части и действия, требуемые при ее проектировании
- 4) Наполнить содержимым программную архитектуру WBS, идентифицируя все ее части и действия, требуемые при ее разработке
- 5) Определить категории затрат, связанных с ПО

### **Построение рабочего графика проекта**

- 1) Определение этапов выполнения работ
- 2) Задание связей между этапами
- 3) Оценка ресурсов для каждого этапа
- 4) Распределение персонала по этапам
- 5) Построение графика

### **Этапы ресурсного планирования**

- 1) определение типов ресурсов, необходимых для реализации требуется проекта
- 2) определение количества каждого типа ресурсов
- 3) определение источника поступления ресурсов каждого типа
- 4) назначение ресурсов задачам
- 5) анализ расписания и разрешение возникших противоречий между требуемым количеством ресурсов и количеством, имеющимся в наличии

### **Алгоритмы устранения перегрузки ресурсов**

- 1) изменить календарь работы ресурса;
- 2) назначить ресурс на неполный рабочий день;
- 3) изменить профиль назначения ресурса;
- 4) добавить ресурсу время задержки;
- 5) разбить задачу на этапы и перекрыть по времени их выполнение

## **3. Экономическая модель разработки ПО. Оценка технико-экономических показателей проекта.**

**Технико-экономическое обоснование программного проекта** представляет собой процедуру оценивания трудовых, временных и финансовых ресурсов по созданию программного продукта, соответствующего требованиям заказчика. Объем требуемых ресурсов зависит от: совокупности бизнес-процессов, описывающих предметную область, и их приоритетов для заказчиков; требований к функциональной полноте и качеству реализации каждого бизнес-процесса.

**Исходные данные, используемые для прогнозирования и планирования**

- 1) функции и номенклатура характеристик самого прогнозируемого объекта или процесса
- 2) характеристики прототипов и пилотных проектов, в некоторой степени подобных планируемому объекту, которые уже завершились и в отношении которых известны необходимые экономические характеристики и можно оценить качество процессов планирования и прогнозирования

**Распределение стоимости**

25% - проектирование – разработка и верификация проекта

15% - спецификации – формулировка требований и условий разработки

20% - разработка – кодирование и тестирование компонент

40% - интеграция и тестирование

**Причины для сравнения реальных данных с прогнозируемыми оценками характеристик проекта**

- 1) несовершенство исходных данных при оценивании технико-экономических показателей программных проектов вызывает необходимость периодического пересмотра прогнозных оценок с учетом новой информации, чтобы обеспечить более реальную основу для дальнейшего управления проектом
- 2) вследствие несовершенства методов оценивания технико-экономических показателей программных проектов следует сравнивать прогнозные оценки этих показателей с действительными значениями, формирующимися в ходе реализации проекта, и использовать эти результаты для улучшения самих методов оценивания
- 3) программные проекты имеют тенденцию к изменению характеристик и экономических факторов, поэтому для их успеха необходимо идентифицировать эти изменения и выполнять реалистичное обновление оценок затрат

**Факторы, повышающие точность оценок**

- 1) цели оценивания технико-экономических показателей должны быть согласованы с потребностями в информации, способствующей принятию решений на соответствующем этапе программного проекта
- 2) достоверность оценок должна быть сбалансирована для различных компонентов системы и величина уровня неопределенности для каждого

компонента должна быть примерно одинаковой, если в процессе принятия решения все компоненты имеют одинаковый вес

3) следует возвращаться к предшествующим целям оценивания технико-экономических показателей и изменять их, когда это необходимо для ответственных бюджетных решений, принимаемых на ранних этапах и влияющих на следующие этапы

**Основные параметры оценки при создании ПП:**

- 1) сложность (размеры)
- 2) трудозатраты на разработку
- 3) длительность разработки в целом и ее отдельных этапов
- 4) численность и квалификация специалистов, привлекаемых к созданию ПП

**Экономическая модель разработки ПО**

**Трудозатраты** = (Персонал)(Среда)(Качество)(Размер<sup>Процесс</sup>)

**Размер** – размер конечного продукта (для компонентов, написанных вручную), который обычно измеряется числом строк исходного кода или количеством функциональных точек, необходимых для реализации данной функциональности. В это понятие также должны входить и другие создаваемые материалы, такие как документация, совокупность тестовых данных и обучающие материалы

**Персонал** – возможности персонала, участвующего в разработке ПО, в особенности его профессиональный опыт и знание предметной области проекта. Источниками сложностей могут быть требуемая надежность программного обеспечения, ограничения на производительность и хранение, требуемое повторное использование программных компонентов, а так же опыт работы программистов с данной средой программирования

**Среда** – состоит из инструментов и методов, используемых для эффективной разработки ПО и автоматизации процесса. Т.е. фактически, это приобретенная или потерянная эффективность вследствие уровня автоматизации процесса (большой уровень автоматизации приводит к уменьшению усилий и повышению эффективности)

**Качество** – требуемое качество продукта, что включает в себя его функциональные возможности, производительность, надежность и адаптируемость

**Процесс** – особенности процесса, используемого для получения конечного продукта, в частности, его способность избегать непроизводительных видов деятельности: переделок, бюрократических проволочек, затрат на взаимодействие

**Алгоритм оценки затрат на разработку ПО**

- 1) оценка размера разрабатываемого продукта
- 2) оценка трудозатрат (трудоемкости) в человеко-месяцах или человеко-часах
- 3) оценка продолжительности проекта в календарных месяцах
- 4) оценка стоимости проекта



4. Оценка размера программного продукта в строках программного кода.  
Модели COSOMO и COSOMO II.

### **Последовательное уточнение технико- экономических параметров программного проекта**

1. Экспертные оценки экономических характеристик по прототипам – достоверность оценивания размера продукта 40 – 50%
2. Этап предварительного проектирования
  - 2.1 оценивание экономических характеристик в базовой модели COSOMO – достоверность оценивания размера продукта 20 – 30%
  - 2.2 оценивание экономических характеристик в детализированной модели COSOMO – достоверность оценивания размера продукта 10 – 20%
3. Этап детального проектирования
  - 3.1 оценивание экономических характеристик в упрощенной предварительной модели COSOMO достоверность оценивания размера продукта 5 – 10% (учитывается 7 параметров)
  - 3.2 оценивание экономических характеристик в детальной модели COSOMO II – достоверность оценивания размера продукта 3 – 5% (учитывается 17 параметров)

### **Проблемы использования LOC в качестве единицы измерения размера программного продукта**

- 1) число строк исходного кода зависит от уровня мастерства программиста. Фактически, чем выше мастерство программиста, тем меньшим количеством строк кода ему удастся обойтись для реализации определенной функциональной возможности (или функциональности) ПС
- 2) высокоуровневые языки или языки визуального программирования требуют гораздо меньшего числа строк кода для отражения одной и той же функциональности, чем, например, язык Ассемблера или С.
- 3) фактическое число строк программного кода остается неизвестным до тех пор, пока проект не будет почти завершен. Поэтому LOC сложно использовать для предварительной оценки трудозатрат на разработку и построения плана-графика проекта
- 4) Применение методов оценки с помощью количества строк не регламентируется промышленными стандартами, например ISO. Языковые конструкции, используемые, например, в Visual C++, Ассемблере, Коболе или SQL существенно различаются. Метод же остается общим для любых приложений, в том числе использующих комбинацию различных языков
- 5) заказчику сложно понять, каково соотношение указанных им функциональных и нефункциональных (технических) требований к ПС и объемов программистской работы
- 6) Разработка ПО может быть связана с большими затратами, которые напрямую не зависят от размеров программного кода (это затраты, связанные с разработкой спецификации требований, подготовкой пользовательской документации и пр., которые не включены в прямые затраты на кодирование)

- 7) При подсчете LOC – единиц следует различать автоматически сгенерированный код и код, написанный вручную, что сильно затрудняет применение автоматических методов подсчета
- 8) Генераторы кода зачастую провоцируют его избыточный объем, что может привести к значительным погрешностям в оценке размера ПП
- 9) Единственным способом получения LOC – оценки является сравнение с аналогичными разработками или экспертные мнения, а эти методы изначально не относятся к числу точных

#### **Рекомендации по повышению достоверности LOC - оценок**

- 1) убедитесь, что каждая учитываемая строка исходного кода содержит лишь один оператор. Если в одной строке содержатся два выполняемых оператора, разделенных точкой с запятой, то они должны учитываться как две строки. Если же один оператор разбит на несколько «физических» строк, он будет учитываться как одна строка. В языках программирования допускаются различные правила кодирования, но обычно проще определять в строке один оператор, обрабатываемый компилятором или интерпретатором
- 2) учитывайте все выполняемые операторы. Конечный пользователь может не иметь возможности практически использовать каждый оператор, но все операторы должны поддерживаться данным продуктом, в том числе и утилитами
- 3) определения данных учитывайте лишь один раз
- 4) не учитывайте строки, содержащие комментарии
- 5) не учитывайте отладочный код либо другой временный код (пробное ПО, средства тестирования и пр.)
- 6) учитывайте каждую инициализацию, вызов или включение макроса (директивы компилятора) в качестве части исходного кода, в которой осуществляется какое-либо действие. Не учитывайте повторно используемые операторы

#### **Преимущества использования LOC в качестве единицы измерения**

- 1) Эти единицы широко распространены и могут адаптироваться
- 2) Они позволяют проводить сопоставление методов измерения размера и производительности в различных группах разработчиков
- 3) Они непосредственно связаны с конечным продуктом
- 4) Единицы LOC могут быть оценены еще до завершения проекта
- 5) Оценка размеров ПО производится с учетом точки зрения разработчиков
- 6) Действия по непрерывному улучшению базируются на количественных оценках. При этом спрогнозированный размер может быть легко сопоставлен с реальным размером на этапе постпроектного анализа. Это позволяет экспертам накапливать опыт и улучшать сами методы оценки
- 7) Знание размера программного продукта в LOC – единицах позволяет применять большинство существующих методов оценки технико-экономических показателей проекта (таких как трудозатраты, длительность проекта, его стоимость и др.)

Программисты могут быть незаслуженно премированы за достижение высоких показателей LOC, если служба менеджмента посчитает это высоким признаком продуктивности. Софтверные организации склонны вознаграждать программистов, которые: а) пишут много кода; б) исправляют много ошибок. Соответственно, наилучший способ отличиться в таких условиях – это создать большое количество некачественного кода, а потом героически устранять в нем собственные же промахи

### **Модель оценки стоимости COCOMO**

(COConstructive COst MOdel — конструктивная модель стоимости)

$$\text{Трудозатраты} = C1 * EAF * (\text{Размер})^{P1} \quad \text{Время} = C2 * (\text{Трудозатраты})^{P2}$$

Трудозатраты (работа) — количество человеко-месяцев; C1 — масштабирующий коэффициент

EAF — уточняющий фактор, характеризующий предметную область, персонал, среду и инструментарий, используемый для создания рабочих продуктов процесса

Размер — размер конечного продукта (кода, созданного человеком), измеряемый в исходных инструкциях (DSI, delivered source instructions), которые необходимы для реализации требуемой функциональной возможности

P1 — показатель степени, характеризующий экономию при больших масштабах, присущую тому процессу, который используется для создания конечного продукта; в частности, способность процесса избегать непроизводительных видов деятельности (доработок, бюрократических проволочек, накладных расходов на взаимодействие)

Время — общее количество месяцев

C2 — масштабирующий коэффициент для сроков исполнения

P2 — показатель степени, который характеризует инерцию и распараллеливание, присущие управлению разработкой ПО

### **Допущения модели COCOMO**

- 1) Исходные инструкции конечного продукта включают в себя все (кроме комментариев) строки кода, обрабатываемого компьютером
- 2) Начало жизненного цикла проекта совпадает с началом разработки продукта, окончание — совпадает с окончанием приемочного тестирования, завершающего стадию интеграции и тестирования
- 3) Работа и время, затрачиваемые на анализ требований, оцениваются отдельно, как дополнительный процент от разработки в целом
- 4) Виды деятельности включают в себя только работы, направленные непосредственно на выполнение проекта
- 5) Человеко-месяц состоит из 152 часов

6) Проект управляется надлежащим образом, в нем используются стабильные требования

Название режима	Размер проекта	Описание	Среда разработки	Трудозатраты	Время
Обычный	До 50 KLOC	Некрупный проект разрабатывается небольшой командой, для которой нехарактерны нововведения, разработчики знакомы с инструментами и языком программирования	Стабильная	$3,2 * EAF * (Размер)^{1,05}$	$2,5 * (Tr)^{0,38}$
Промежуточный	50 – 500 KLOC	Относительно небольшая команда занимается проектом среднего размера, в процессе разработки необходимы определенные инновации	Среда характеризуется незначительной нестабильностью	$3 * EAF * (Размер)^{1,12}$	$2,5 * (Tr)^{0,35}$
Встроенный	Более 500 KLOC	Большая команда разработчиков трудится над крупным проектом, необходим значительный объем инноваций	Среда состоит из множества нестабильных элементов	$2,8 * EAF * (Размер)^{1,2}$	$2,5 * (Tr)^{0,32}$

Идентификатор	Уточняющий фактор работ	Очень низкий	Низкий	Номинальный	Высокий	Очень высокий
Атрибуты программного продукта						
RELY	Требуемая надежность	0,75	0,86	1,0	1,15	1,4
DATA	Размер базы данных		0,94	1,0	1,08	1,16
CPLX	Сложность продукта	0,7	0,85	1,0	1,15	1,3
Атрибуты компьютера						
TIME	Ограничение времени выполнения			1,0	1,11	1,50

STOR	Ограничение объема основной памяти			1,0	1,06	1,21
VIRT	Изменчивость виртуальной машины		0,87	1,0	1,15	1,30
TURN	Время реакции компьютера		0,87	1,0	1,07	1,15
Атрибуты персонала						
ACAP	Способности аналитика	1,46	1,19	1,0	0,86	0,71
AEXP	Знание приложений	1,29	1,15	1,0	0,91	0,82
PCAP	Способности программиста	1,42	1,17	1,00	0,86	0,7
VEXP	Знание виртуальной машины	1,21	1,1	1,0	0,9	
LEXP	Знание языка программирования	1,14	1,07	1,0	0,95	
Атрибуты проекта						
MODP	Использование современных методов	1,24	1,1	1,0	0,91	0,82
TOOL	Использование программных инструментов	1,24	1,1	1,0	0,91	0,82
SCED	Требуемые сроки разработки	1,23	1,08	1,0	1,04	1,1

Вид деятельности	Труд (%)	Время (%)
Планирование и определение требований	(+8)	(+36)
Проектирование продукта	18	36
Детальное проектирование	25	18
Кодирование и тестирование отдельных модулей	26	18
Интеграция и тестирование	31	28

Вид деятельности	Бюджет (%)
Анализ требований	4
Проектирование продукта	12
Программирование	44
Тестирование	6
Верификация и аттестация	14
Канцелярия проекта	7
Управление конфигурацией и обеспечение качества	7
Создание руководств	6

### **Достоинства модели COSOMO**

- 1) Метод является достаточно универсальным и может поддерживать различные режимы и уровни программных разработок
- 2) При расчетах используются множители и показатели степени, полученные на основе анализа данных большого количества практически реализованных проектов
- 3) Предложенные драйверы затрат хорошо подгоняются под специфику конкретной организации
- 4) Точность оценок повышается по мере накопления в организации опыта применения модели
- 5) Метод снабжен обширной документацией и прост в применении

### **Недостатки модели COSOMO**

- 1) Все уровни зависят от оценки размера – точность оценки размера оказывает влияние на точность оценки трудозатрат, времени разработки, подбор персонала и оценку производительности
- 2) Метод основан на каскадной модели жизненного цикла и прежде всего не учитывает изменяемость требований
- 3) Слишком поверхностное внимание уделено вопросам обеспечения безопасности и надежности
- 4) Модель не учитывает возможности повторного использования кода, итерационные возвраты по этапам жизненного цикла, объектно-ориентированные технологии разработки ПО

### **Концепция повторного использования**

- 1) Многие программы происходят от предыдущих версий этих же программ
- 2) В результате может быть достигнута экономия средств и/или времени на разработку, а также повышен уровень качества ПО
- 3) Повторному использованию подлежат: программный код, тестовый код, тестовые процедуры, документация, дизайн, спецификации требований
- 4) Код, повторно используемый в полном объеме, имеет идентичную документацию, идентичные тестовые процедуры и сам тестовый код и только одну копию, поддерживаемую системой управления конфигурацией

### **Повторное использование кода**

- 1) Новый код – это код, разработанный для нового приложения, который не включает большие порции ранее написанного кода
- 2) Модифицируемый код – это код, разработанный для предыдущих приложений, который будет пригоден для использования в новом приложении после внесения умеренного объема изменений
- 3) Повторно используемый код – это код, разработанный для предыдущих приложений, который будет пригодным для новых приложений без внесения каких-либо изменений

## **COSOMO II**

- 1) Модель композиции приложения – это модель, которая подходит для проектов, созданных с помощью современных инструментальных средств. Единицей измерения служит объектная точка
- 2) Модель ранней разработки архитектуры. Эта модель применяется для получения приблизительных оценок проектных затрат периода выполнения проекта перед тем как будет определена архитектура в целом. В этом случае используется небольшой набор новых драйверов затрат и новых уравнений оценки. В качестве единиц измерения используются функциональные точки либо KSLOC
- 3) Постархитектурная модель – наиболее детализированная модель COCOMO II, которая используется после разработки архитектуры проекта. В состав этой модели включены новые драйверы затрат, новые правила подсчета строк кода, а также новые уравнения

### Модель композиции приложения

Данная модель используется на ранней стадии конструирования ПО, когда:

- 1) рассматривается макетирование пользовательских интерфейсов
- 2) оценивается производительность
- 3) определяется степень зрелости технологии

### Правила подсчета объектных точек

Вид	Простые	Умеренные	Сложные
Экранные формы	1	2	3
Отчеты	2	5	8
Модули(яп 3 поколения)	10	10	10

$NOP = (\text{Объектные точки}) \times [(100 - \%RUSE) / 100]$  – новые объектные точки

$ТРУДОЗАТРАТЫ = NOP / PROD$  [чел.- мес.]

Опытность/ возможности разработчика Зрелость/ возможности среды разработки	PROD
Очень низкая	4
Низкая	7
Номинальная	13
Высокая	25
Очень высокая	50

### Модель ранней разработки архитектуры

$Трудозатраты = 2,45 * EArch * (Размер)^P$ ,

$EArch = PERS * RCPX * RUSE * PDIF * PREX * FCIL * SCED$

Размер — KSLOC (предпочтительно для подсчета KSLOC предварительно подсчитать количество функциональных точек)

$Время = 3,0 * (Трудозатраты)^{(0.33 + 0.2 * (p-1.01))}$

		Оценка уровня множителя трудоемкости					
		Очень низкий	Низкий	Норм	Высокий	Очень высокий	Сверх высокий
PERS	возможности персонала (ACAP-PCAP-PCON)	1.62	1.26	1.00	0.83	0.63	0.5
RCPX	надежность и уровень сложности разрабатываемой системы (RELY-DATA-CPLX-DOCU)	0.60	0.83	1.00	1.33	1.91	2.72
RUSE	повторное использование компонентов	n/a	0.95	1.00	1.07	1.15	1.24
PDIF	сложность платформы разработки (TIME-STOR-PVOL)	n/a	0.87	1.00	1.29	1.81	2.61
PREX	опыт персонала (AEXP-PEXP-LTEX)	1.33	1.22	1.00	0.87	0.74	0.62
FCIL	средства поддержки (TOOL-SITE)	1.30	1.10	1.00	0.87	0.73	0.62
SCED	график работ , сроки	1.43	1.14	1.00	1.00	1.00	n/a

### Модель этапа постархитектуры

Трудозатраты =  $2,45 * E_{App} * (Размер)^P$ ,

$E_{App}$  — результат применения семнадцати уточняющих факторов постархитектурных этапов разработки

Время =  $3,0 * (Трудозатраты)^{(0.33 + 0.2 * (p-1.01))}$

Идентификатор	Уточняющий фактор работ	Изменение в COSOMO II
RELY	Требуемая надежность	Без изменений относительно COSOMO
DATA	Размер базы данных	Без изменений относительно COSOMO
CPLX	Сложность продукта	Без изменений относительно COSOMO
RUSE	Требуемый уровень повторного использования	
DOCU	Документация	Добавлен. Определяет насколько документация соответствует требованиям жизненного цикла





PREC	Отражает предыдущий опыт организации в реализации проектов данного типа. Очень низкий уровень этого показателя означает отсутствие опыта, наивысший уровень указывает на компетентность организации-разработчика в данной области ПО	6,2	4,96	3,72	2,48	1,24	0
FLEX	Отображает возможность изменения процесса разработки ПО. Очень низкий уровень этого показателя означает, что процесс определен заказчиком заранее, наивысший — заказчик определил лишь общие задачи без указания конкретной технологии процесса разработки ПО	5,07	4,05	3,04	2,07	1,01	0
RESL	Отображает степень детализации анализа рисков, основанного на анализе архитектуры системы. Очень низкий уровень данного показателя соответствует поверхностному анализу рисков, наивысший уровень означает, что был проведен тщательный и полный анализ всевозможных рисков	7	5,65	4,24	2,83	1,41	0

TEA М	Отображает степень сплоченности команды и их способность работать совместно. Очень низкий уровень этого показателя означает, что взаимоотношения в команде сложные, а наивысший — что команда сплоченная и эффективная в работе, не имеет проблем во взаимоотношениях	5,48	4,38	3,29	2,19	1,1	0
РМАТ	Отображает уровень развития процесса создания ПО в организации-разработчике	7	6,24	4,68	3,12	1,56	0

### Достоинства модели СОСОМО II

- 1) возможен учет достаточно полной номенклатуры факторов, влияющих на экономические характеристики производства сложных программных продуктов
  - 2) метод является достаточно универсальным и может поддерживать различные размеры, режимы и уровни качества продуктов
  - 3) фактические данные подбираются в соответствии с реальными проектами и факторами корректировки, которые могут соответствовать конкретному проекту и организации
  - 4) прогнозы производственных процессов являются варьируемыми и повторяемыми
  - 5) метод позволяет добавлять уникальные факторы для корректировки экономических характеристик, связанные со специфическим проектом и организацией
  - 6) возможна высокая степень достоверности калибровки с опорой на предыдущий опыт коллектива специалистов
  - 7) результаты прогнозирования сопровождаются обязательной документацией
- модель относительно проста в освоении и применении

### Недостатки модели СОСОМО II

- 1) все результаты зависят от размера программного продукта: точность оценки размера, оказывает определяющее влияние на точность прогноза трудозатрат, длительности разработки и численности специалистов
- 2) игнорируются требования к характеристикам качества программного продукта
- 3) не учитывается зависимость между интегральными затратами и количеством времени, затрачиваемым на каждом этапе проекта

- 4) игнорируется изменяемость требований к программному продукту в процессе производства
- 5) не достаточно учитывается внешняя среда производства и применения программного продукта
- 6) игнорируются многие особенности, связанные с аппаратным обеспечением проекта

## **5. Функционально-ориентированные метрики измерения программного продукта. Оценка технико- экономических параметров программного проекта на основе метода функциональных точек.**

### **Функционально-ориентированные метрики измерения программного продукта**

Функционально-ориентированные метрики косвенно измеряют программный продукт и процесс его разработки. При этом рассматривается не размер, а функциональность или полезность продукта. В качестве количественной характеристики применяется понятие количества функциональных точек FP (function points)

Метод функциональных точек позволяет:

- 1) оценивать категории пользовательских бизнес-функций
- 2) разрешить проблему, связанную с трудностью получения LOC – оценок на ранних стадиях жизненного цикла
- 3) определять количество и сложность входных и выходных данных, их структуру, а также внешние интерфейсы, связанные с программной системой

**Трудоемкость** вычисляется на основе функциональности разрабатываемой системы, которая, в свою очередь, определяется путем выявления функциональных типов — логических групп взаимосвязанных данных, используемых и поддерживаемых приложением, а также элементарных процессов, связанных с вводом и выводом информации

**Внешний ввод** (EI, транзакция, получающая данные от пользователя) — элементарный процесс, перемещающий данные из внешней среды в приложение. Данные могут поступать с экрана ввода или из другого приложения. Данные могут содержать как управляющую, так и деловую информацию. Обработываемые данные могут соответствовать одному или нескольким внутренним логическим файлам

**Внешний вывод** (EO, транзакция передающая данные пользователю) — элементарный процесс, перемещающий данные, вычисленные в приложении, во внешнюю среду и связанный с созданием и/или обработкой выходной информации приложения — выходного отчета, документа, экранной формы

**Внешний запрос** (EQ, интерактивный диалог с пользователем, требующий от него каких-либо действий) — элементарный процесс, состоящий из комбинации «запрос/ответ», не связанный с вычислением производных данных или обновлением внутренних логических файлов (базы данных)

**Внутренний логический файл (ILF**, информация, которая используется во внутренних взаимодействиях системы) — выделяемые пользователем логически связанные группы данных или блоки управляющей информации, которые поддерживаются внутри продукта и обслуживаются через внешние вводы

**Внешний интерфейсный файл (EIF**, файлы, участвующие во внешних взаимодействиях с другими системами) — выделяемые пользователем логически связанные группы данных или блоки управляющей информации, на которые ссылается продукт, но которые поддерживаются вне продукта

Функциональность логических файлов (ILF и EIF) оценивается путем подсчета количества типов элементов записей (**RET**) и количества типов элементов данных (**DET**), входящих в соответствующие логические группы данных. При этом под количеством RET обычно понимается количество различных логических подгрупп данных, выделяемых в файле с точки зрения пользователя, или количество различных используемых форматов записей, а под количеством DET - количество различных элементарных полей в этих записях

Типы элементов - записей (RET)	Элементы данных (DET)					
	1-19 ILF	EIF	20-50 ILF	EIF	>50 ILF	EIF
1	Низ (7)	Низ (5)	Низ (7)	Низ (5)	Сред (10)	Сред (7)
2-5	Низ (7)	Низ (5)	Сред (10)	Сред (7)	Выс (15)	Выс (10)
>5	Сред (10)	Сред (7)	Выс (15)	Выс (10)	Выс(15)	Выс(10)

В качестве DET для внешних вводов (EI) учитываются:

- 1) каждое нерекурсивное поле, принадлежащее внутреннему логическому файлу (ILF) (или поддерживаемое им) и обрабатываемое во вводе
- 2) каждое поле, которое пользователь хотя и не вызывает, но оно через процесс ввода поддерживается во внутреннем логическом файле (ILF)
- 3) логическое поле, которое физически представляет собой множество полей, но воспринимается пользователем как единый блок информации
- 4) группа полей, которые появляются во внутреннем логическом файле (ILF) более одного раза, но в связи с особенностями алгоритма их использования воспринимаются как один DET
- 5) группа полей, которые фиксируют ошибки в процессе обработки или подтверждают, что обработка закончилась успешно
- 6) действие, которое может быть выполнено во вводе

Ссылки на файлы (FTR)	Элементы данных (DET)		
	1-4	5-15	>15

0-1	Низкий (3)	Низкий (3)	Средний (4)
2	Низкий (3)	Средний (4)	Высокий (6)
>2	Средний (4)	Высокий (6)	Высокий (6)

В качестве DET для внешних выводов (EO) учитываются:

- 1) каждое распознаваемое пользователем нерекурсивное поле, участвующее в процессе вывода
- 2) поле, которое физически отображается в виде нескольких полей его составляющих, но используется как единый информационный элемент
- 3) каждый тип метки и каждое значение числового эквивалента при графическом выводе
- 4) текстовая информация, которая может содержать одно слово, предложение или фразу

Ссылки на файлы (FTR)	Элементы данных (DET)		
	1-4	5-19	>19
0-1	Низкий (4)	Низкий (4)	Средний (5)
2-3	Низкий (4)	Средний (5)	Высокий (7)
>3	Средний (5)	Высокий (7)	Высокий (7)

В качестве DET для внешнего запроса (EQ) по входу учитываются:

- 1) каждое распознаваемое пользователем нерекурсивное поле, появляющееся во вводной части запроса
- 2) каждое поле, которое определяет критерий выбора данных
- 3) группа полей, в которых выдаются сообщения о возникающих ошибках в процессе ввода информации или подтверждающих успешное завершение процесса ввода
- 4) группа полей, которые позволяют выполнять запросы

В качестве DET для внешнего запроса (EQ) по выходу учитываются:

- 5) каждое распознаваемое пользователем нерекурсивное поле, которое появляется в выводной части запроса
- 6) логическое поле, которое физически отображается как группа полей, однако воспринимается пользователем как единое поле
- 7) группа полей, которые в соответствии с методикой обработки могут повторяться во внутреннем логическом файле (ILF)

Ссылки на файлы (FTR)	Элементы данных		
	1-4	5-19	>19
0-1	Низкий (3)	Низкий (3)	Средний (4)
2-3	Низкий (3)	Средний (4)	Высокий (6)
>3	Средний (4)	Высокий (6)	Высокий (6)

### Алгоритм применения метода функциональных точек

- 1) Подсчитываются функции для каждой категории (вводы, выводы, запросы, структуры данных, интерфейсы)

- 2) Устанавливаются требования для каждой категории
- 3) Определяется сложность каждой функции (высокая, средняя, низкая)
- 4) Каждая функция умножается на соответствующий ей параметр, а затем суммируется с целью получения общего количества функциональных точек

$FP = \text{Общее количество} * (0,65 + 0,01 * \sum Fi)$ ,

где  $Fi$  — 14 коэффициентов регулировки сложности

Каждый коэффициент может принимать следующие значения:

0 - нет влияния, 1 - случайное, 2 - небольшое, 3 - среднее, 4 - важное, 5 - основное

№	Системный параметр	Описание
1	Передача данных	Сколько средств связи требуется для передачи или обмена информацией с приложением или системой?
2	Распределенная обработка данных	Как обрабатываются распределенные данные и функции обработки?
3	Производительность	Нуждается ли пользователь в фиксации времени ответа или производительности?
4	Эксплуатационные ограничения	Насколько сильны эксплуатационные ограничения и каков объем специальных усилий на их преодоление?
5	Частота транзакций	Как часто выполняются транзакции (каждый день, каждую неделю, каждый месяц) ?
6	Оперативный ввод данных	Какой процент информации надо вводить в режиме онлайн?
7	Эффективность работы конечных пользователей	Приложение проектировалось для обеспечения эффективной работы конечного пользователя?
8	Оперативное обновление	Как много внутренних файлов обновляется в онлайн-транзакции?
9	Сложность обработки	Выполняет ли приложение интенсивную логическую или математическую обработку?
10	Повторная используемость	Приложение разрабатывалось для удовлетворения требований одного или многих пользователей?
11	Легкость инсталляции	Насколько трудны преобразование и инсталляция приложения?
12	Легкость эксплуатации	Насколько эффективны и/или автоматизированы процедуры запуска, резервирования и восстановления?

13	Количество возможных установок на различных платформах	Была ли спроектирована, разработана и поддержана возможность инсталляции приложения в разных местах для различных организаций?
14	Простота изменений (гибкость)	Была ли спроектирована, разработана и поддержана в приложении простота изменений?

## **6. Основы ценообразования на программные продукты. Бюджет как основа планирования деятельности. Определение фонда оплаты труда на разработку программного продукта.**

Много продаж по малой цене или мало продаж по большой цене ?

**Рынок чистой конкуренции:** состоит из множества производителей и потребителей какого-либо программного продукта, реализующего выполнение конкретной функции. Ни один отдельный производитель или потребитель не оказывает большого влияния на уровень текущих рыночных цен. На рынке чистой конкуренции роль маркетинговых исследований, политики цен, рекламы, политики сбыта и прочих мероприятий минимальна

**Рынок монополистической конкуренции:** дифференцированный подход к ценообразованию программного продукта предполагает, что цены на него могут устанавливаться в широком диапазоне. Наличие диапазона цен объясняется способностью производителей предложить потребителям разные варианты услуг по продаже и сопровождению ПП, включая сопутствующие программные продукты и дополнительные сервисы по поддержке пользователя. Чтобы выделиться чем-то помимо цены на ПП и перечня услуг разработчики ориентируются на разные потребительские сегменты, широко используют рекламу и другие каналы продвижения продукции

**Олигополистический рынок** состоит из небольшого числа производителей, чувствительных к политике ценообразования и маркетинговой стратегии друг друга. Новым претендентам тяжело проникнуть на этот рынок: если какая-то компания снизит цены на свои продукты, то другим производителям придется либо тоже снижать цены, либо предлагать дополнительные сервисы

**Рынок чистой монополии:** на таком рынке присутствует всего один продавец. В случае регулируемой монополии государство может устанавливать для такого продукта цену, обеспечивающую получение «справедливой нормы прибыли», позволяющую организации не только поддерживать производство, но и при необходимости расширять его. При нерегулируемой монополии фирма сама вправе устанавливать любую цену, которую выдержит рынок

### **Альтернативные стратегии ценообразования**

- 1) Обеспечение выживаемости ПП на рынках. Проблемы могут быть вызваны активизацией конкурентов или изменившимися запросами потребителей. Понимая, что в какой-то момент времени выживание для фирмы важнее прибыли, она может установить низкие цены на свои продукты в надежде на благоприятную реакцию потребителей. До тех пор, пока снижение цены



на программный продукт покрывает издержки по его разработке продукт следует продвигать

- 2) Максимизация прибыли. При реализации данной стратегии фирмы ориентируются на уровень цен, обеспечивающих в краткосрочном периоде получение максимальной прибыли, и не рассматривают долгосрочные перспективы
- 3) Максимальное увеличение объема продаж. Фирма снижает цены на свою продукцию до минимально допустимого уровня, повышая долю своего рынка, добивается снижения издержек на единицу продукции и на этой основе может и дальше снижать цены. Эта политика хорошо срабатывает, если чувствительность рынка к ценам велика
- 4) «Снятие сливок» за счет установления высоких цен. Фирма устанавливает на новый продукт, который выходит на рынок максимально возможную цену благодаря сравнительным преимуществам новой разработки. Как только продажи продукта по данной цене сокращаются, фирма снижает цену, привлекая к себе следующий слой потребителей, достигая в каждом сегменте целевого рынка максимально высокого оборота
- 5) Достижение лидерства по качеству. Фирма, которая способна закрепить за собой репутацию лидера по качеству, устанавливает высокую цену, чтобы покрыть значительные издержки, связанные с обеспечением высокого качества продукта

#### **Методы ценообразования, ориентированные на затраты**

Основаны на учете расходов производителя по созданию и реализации продукта. При определении цены необходимо выявить диапазон цен, обеспечивающих покрытие переменных и постоянных затрат и получение желаемой прибыли

#### **Методы ценообразования, ориентированные на спрос**

Предусматривают готовность потребителей оплачивать программные продукты по верхней границе цены

При использовании данных методов не прослеживается связь между затратами и ценой, кроме того, что цена не может быть ниже уровня нижней границы

Оценка спроса имеет прогнозный характер и его труднее выразить в количественных величинах, чем издержки

Задача по оценке спроса усложняется, если на рынок выводится новый продукт в силу отсутствия статистики за прошлые периоды

#### **Методы ценообразования, ориентированные на конкурентов**

Основываются на сравнении потребительских свойств ПП разработчика и конкурентов

Потребительская ценность программного продукта зависит от ряда факторов, таких как качество

продукта, возможность подключения других приложений, платформонезависимость и др.

При использовании данной политики существуют две проблемы:

- у потенциальных пользователей нет навыков объективного анализа потребительских свойств ПП

■ точка зрения пользователя и точка зрения разработчика на потребительские свойства ПП может различаться

Потребители как правило отдают предпочтение продуктам, позволяющим получить максимальную выгоду при их использовании, определяемую как разность между потребительской ценностью и общей стоимостью продукта, включающей единовременные затраты на приобретение и внедрение и текущие расходы на эксплуатацию

При сравнении ПП с продуктами конкурентов не обязательно, чтобы он превосходил их по всем параметрам. Необходимо выбрать главное конкурентное преимущество и на нем делать акцент

**Базовая цена**, установленная по любому из перечисленных методов ценообразования, может быть скорректирована в сторону уменьшения за счет:

- 1) установления скидок при большом заказе количества лицензий
- 2) предложения различных форм оплаты: с предоплатой, оплатой по факту получения продукта, оплаты с рассрочкой и т.д.
- 3) установления скидок определенным группам клиентов
- 4) включением в цену продажи сервисов по техническому сопровождению и поддержке пользователей
- 5) предоставления возможности выбора набора функциональности ПП и способа поставки (стандартная, профессиональная, промышленная версии)

**Бюджет** - это выраженный в деньгах количественный план реализации проекта, подготовленный и принятый на определенный период времени и показывающий планируемую величину дохода, его источники и расходы, предстоящие в течение этого периода, а также в случае необходимости – величину привлекаемого капитала

Назначение бюджета:

- 1) контроль производственной ситуации (в отсутствие бюджета руководитель может только констатировать текущее состояние проекта, вместо того, чтобы реагировать на возникающие проблемы)
- 2) объективная оценка результатов деятельности как в целом, так и по отдельным направлениям реализации, за счет сравнения плановых и фактических показателей
- 3) выявление за счет сравнения плановых показателей бюджета и фактических результатов тех ситуаций, которые требуют корректирующего воздействия

**Основные показатели, используемые для оценки стоимости ПП**

- 1) Сложность (размер)
- 2) Трудозатраты на разработку
- 3) Длительность разработки продукта в целом и ее отдельных этапов
- 4) Численность и квалификация специалистов, привлекаемых к созданию программного продукта
- 5) Размеры фондов оплаты труда специалистов на создание ПП в целом и по каждому этапу жизненного цикла
- 6) Прочие прямые затраты и накладные расходы, связанные с созданием ПП

### Этапы составления бюджета

- 1) Определение основных параметров бюджета
- 2) Планирование доходной части
- 3) Планирование бюджета основных расходов
- 4) Планирование бюджета накладных расходов

### Нормативы, используемые при разработке бюджета:

- 1) Рыночная цена продажи одной лицензии на ПП
- 2) Ожидаемый объем продаж ПП
- 3) Ожидаемые трудозатраты реализации проекта по разработке ПП в целом и его отдельных этапов
- 4) Расчетная ставка стоимости одного часа труда специалиста, занятого в реализации проекта
- 5) Норматив отчисления на накладные расходы
- 6) Нормы расхода материальных ресурсов на единицу стоимости ПП



### В зависимости от объемов производства

Постоянные затраты (не зависят от изменения объемов производства)

Переменные (изменяются вместе с изменением объемов производства)

### В зависимости от роли в реализации основных видов деятельности

Основные затраты (непосредственно связаны с технологическим процессом изготовления продукции)

Накладные расходы (затраты на организацию и обслуживание производства, реализацию продукции, управление организацией)

### В зависимости от времени возникновения и отнесения на себестоимость продукции

Текущие затраты (затраты по производству и реализации продукции текущего отчетного периода)

Расходы будущих периодов (произведенные затраты текущего отчетного периода, которые будут отнесены на себестоимость продукции, которая будет выпущена в последующие периоды)

## **Классификация затрат по статьям калькуляции**

- 1) заработная плата работников
- 2) отчисления на социальные нужды
- 3) увеличение стоимости основных фондов (приобретение компьютерной и офисной техники)
- 4) приобретение расходных материалов и комплектующих
- 5) затраты на подготовку и переподготовку персонала
- 6) налог на прибыль
- 7) оплата коммунальных услуг
- 8) оплата услуг связи
- 9) командировочные расходы

**Себестоимость** – это стоимостная оценка используемых в процессе выполнения работ (оказания услуг) материальных и трудовых ресурсов, основных фондов, энергии, а также других затрат на производство и реализацию. Затраты, образующие себестоимость продукции: (Материальные затраты, Затраты на оплату труда, Отчисления на социальные нужды, Амортизация основных фондов, Прочие затраты)

### **Материальные затраты**

- 1) Стоимость приобретенных на стороне компонентов, которые входят в разрабатываемые программные продукты
- 2) Стоимость покупных материалов, используемых для обеспечения нормального технологического процесса производства (бумага, картриджи, сменные носители и др.) и реализации продукции (например, внешнего носителя, на который записывается дистрибутив, или упаковки), а также технической литературы (при необходимости)
- 3) Стоимость работ (услуг) производственного характера, выполняемых сторонними организациями
- 4) Коммунальные платежи: отопление, электроэнергия...

### **Затраты на оплату труда**

- 1) Выплаты заработной платы за фактически выполненную работу
- 2) Выплаты стимулирующего характера, премии и другие вознаграждения по итогам работы
- 3) Стоимость выдаваемых бесплатно предметов, остающихся в постоянном пользовании
- 4) Оплата очередных и дополнительных отпусков, компенсация за неиспользуемый отпуск
- 5) Выплаты работникам, высвобождаемым с предприятий по сокращению штатов
- 6) Оплата учебных отпусков, предоставляемых работникам, обучающимся в вечерних и заочных учебных заведениях, заочной аспирантуре
- 7) Оплата труда работников, не состоящих в штате организации, за выполнении ими работ по заключенным договорам гражданско-правового характера

**Отчисления на социальные нужды** - обязательные отчисления по установленным законодательством нормам в органы государственного страхования, пенсионный фонд, государственный фонд занятости населения, фонд медицинского страхования.

#### **Амортизация основных фондов**

Отражает сумму амортизационных отчислений на полное восстановления основных средств, исчисленную исходя из их балансовой стоимости

**Прочие затраты** включают расходы на канцелярские товары, командировки, оплату телефонных переговоров, мобильной связи, подписки на СМИ и ПО, представительские расходы и пр.

**Определение фонда оплаты труда на разработку программного продукта** производится на основании показателя производительности труда программиста, который выражается в количестве строк программного кода, написанного за один рабочий день

В качестве основания для определения фонда оплаты труда используются:

- 1) длительность реализации каждого этапа жизненного цикла разработки программного продукта
- 2) количественный и качественный состав специалистов, привлекаемых на каждом этапе разработки
- 3) базовая месячная ставка специалиста-программиста

**Трудозатраты** на разработку ПП:

$T = R / P$ , где  $R$  – размер программного кода (LOC),  $P$  – производительность труда специалистов

**Средняя численность** специалистов:  $Z = T / D$ , где  $D$  – длительность реализации проекта

Этапы жизненного цикла	Типы специалистов, %		
	аналитики	программисты	технические специалисты
1. Анализ предметной области и разработка требований	40	20	40
2. Проектирование	35	35	30
3. Программирование	10	65	25
4. Тестирование и комплексные испытания	15	60	25

Фонд заработной платы для реализации  $i$ -го этапа проекта:

$$Z_i = \sum_{j=1}^3 Z_{ij} * D_i * S_i$$

$D_i$  – длительность  $i$ -го этапа проекта,  $S_i$  – месячный фонд заработной платы специалиста  $j$ -го типа,  $Z_{ij}$  - Численность каждого типа специалистов

Соотношение месячной ставки программиста к месячной ставке системного аналитика составляет 1:1,3, а к месячной ставке технического специалиста 1:0,7