**Технико-экономическое обоснование разработки**

В данной главе дипломного проекта выполняется технико-экономическое обоснование разработки репозитория инсталляционных пакетов с целью определения этапов работ, связанных с процессом разработки программной системы. Основными задачами, которые решаются на данном этапе, являются:

* Определение этапов работ по создания программного обеспечения;
* Расчет трудоемкости проекта;
* Определение численности исполнителей;
* Построение сетевого графика выполнения проекта;
* Построение календарного графика выполнения работ.

**Расчет трудоемкости проекта**

Расчет трудоемкости проекта необходим для определения затрат на реализацию репозитория инсталляционных пакетов. Через трудоемкость в конченом итоге оценивается один из основных затратных показателей – совокупные затраты на оплату труда исполнителей проекта.

Методы и модели оценки стоимости ПО можно разделить на две группы: неалгоритмические методы и алгоритмические модели. К неалгоритмическим методам относятся Price-to-win, оценка по Паркинсону, экспертная оценка, оценка по аналогии. К алгоритмическим моделям относятся SLIM и COCOMO.

Сущность неалгоритмических методов состоит в том, что при оценке стоимости ПО используются определенные схемы и принципы, а не математические формулы.

Price-to-win. Метод основывается на принципе «клиент всегда прав». Суть метода состоит в том, что независимо от предполагаемых реальных затрат на разработку проекта, оценка стоимости ПО корректируется в соответствии с пожеланиями заказчика. Price-to-win фактически является политикой проведения переговоров с клиентом, поэтому часто применяется компаниями, не имеющими средств для качественной оценки проектов.

Оценка по Паркинсону. Метод основывается на принципе: «Объем работы возрастает в той мере, в какой это необходимо, чтобы занять время, выделенное на ее выполнение». В применении к разработке программных проектов, закон Паркинсона используется в виде следующей схемы: чтобы повысить производительность труда разработчика, необходимо уменьшить время, отведённое на разработку.

Экспертная оценка. Метод основывается на принципе экспертной оценки и применяется в проектах, использующих новые технологии, новые процессы или решающих инновационные задачи. К процессу оценки привлекаются инженеры-разработчики, которые сами оценивают курируемую ими часть проекта. После этого созывается собрание, на котором результаты отдельных оценок интегрируются в единую, целостную систему. В результате достигается баланс оценки при интеграции отдельных компонентов в общую систему.

Оценка по аналогии. Являясь разновидностью экспертной оценки, часто выделяется в отдельный метод. Оценка по аналогии, как и алгоритмические модели, использует эмпирические данные о характеристиках завершённых проектов. Ключевое различие состоит в том, что алгоритмические модели используют эти данные косвенным образом, например, для калибровки параметров моделей, а метод оценки по аналогии с помощью эмпирических данных позволяет отобрать схожие проекты.

Модель Путнэма (SLIM). Наиболее распространенная модель аналитической группы. Основывается на утверждении, что затраты на разработку ПО распределяются согласно кривым Нордена-Рэйли, которые являются графиками функции, представляющей распределение рабочей силы по времени.

Модель COCOMO (COnstructive COst MOdel) – это алгоритмическая модель оценки стоимости разработки программного обеспечения, разработанная Барри Боэмом (Barry Boehm). Модель использует простую формулу регрессии с параметрами, определенными из данных, собранных по ряду проектов. Для оценки трудозатрат на базовом уровне модели COCOMO применялась следующая формула:

, (51)

где, a и b – константы, которые зависят от режима использования модели. В соответствии с этой формулой трудозатраты вообще нелинейно зависят от размера проекта и скачкообразно изменяются при смене режима.

В данной работы был выбран метод экспертной оценки, как наиболее подходящий виду поставленной задачи и ее масштабу. В данном методе необходимо самостоятельно рассчитать ожидаемую продолжительность работ по формуле:

, (51)

где Tmin и Tmax - минимальная и максимальная продолжительность работы. Они назначаются в соответствии с экспертными оценками, а ожидаемая продолжительность работы рассчитывается как математическое ожидание для β - распределения.

Полный перечень этапов работ выполнения проекта, а также трудоемкость каждого этапа представлен в таблице 51.

Таблица 51. Перечень этапов выполнения проекта.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Этап | № работы | Содержание работы | Трудоемкость t | |
| чел. час | чел. дни |
| 1. Подготовительный этап | 1 | Анализ требований ТЗ | 4,2 | 0,525 |
| 2. Проектирование программного обеспечения | 1 | Уточнение требований | 3,8 | 0,475 |
| 2 | Выбор архитектуры | 12 | 1,5 |
| 3 | Выбор программной платформы | 9,6 | 1,2 |
| 3. Реализация компонентов системы | 1 | Разработка базы данных | 3,8 | 0,475 |
| 2 | Разработка классов | 4,8 | 0,6 |
| 3 | Написание кодов программы на выбранном языке программирования | 170 | 21,25 |
| 4. Тестирование программного обеспечения, завершение работы | 1 | Тестирование. | 7 | 0,875 |
| 2 | Внесение поправок | 44 | 5,5 |
| 5. Подготовка документации | 1 | Подготовка сопроводительной документации. | 19 | 2,375 |

В таблице 52 приведена экспертная оценка продолжительностей работ разработки репозитория инсталляционных пакетов.

Таблица 52. Экспертная оценка продолжительностей работ.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № этапа | № работы | (чел. час) | (чел. час) | (чел. час) |
| 1 | 1 | 3 | 6 | 4,2 |
| 2 | 1 | 3 | 5 | 3,8 |
| 2 | 10 | 15 | 12 |
| 3 | 8 | 12 | 9,6 |
| 3 | 1 | 3 | 5 | 3,8 |
| 2 | 4 | 6 | 4,8 |
| 3 | 150 | 200 | 170 |
| 4 | 1 | 5 | 10 | 7 |
| 2 | 40 | 50 | 44 |
| 5 | 1 | 15 | 25 | 19 |

Общая трудоемкость разработки проекта  определяется как совокупные затраты труда на выполнение каждого из его этапов:

, (52)

где ***ti*** - затраты труда на выполнение ***i*** *-го* этапа проекта. После подстановки в данную формулу продолжительность работ каждого этапа, была получена общая трудоемкость чел. час. = 34.775 чел. дней.

**Определение численности исполнителей**

Для оценки возможности выполнения проекта, в соответствии с имеющимся в распоряжении разработчика штатным составом исполнителей, рассчитывается их средняя численность. Она определяется соотношением:

, (53)

где Q – затраты труда на выполнение проекта, чел/мес.;

F – фонд рабочего времени, мес.

Величина фонда рабочего времени определяется соотношением:

, (54)

где Т - время выполнения проекта, мес. (как правило, устанавливается заказчиком проекта);

FM - фонд времени в текущем месяце, ч. (мес.)., который рассчитывается из учета общего числа дней в году, числа выходных и праздничных дней:

, (55)

где tp - продолжительность рабочего дня, ч.;

DK - общее число дней в году;

DB - число выходных дней в году;

DП - число праздничных дней в году.

Тогда, фонд времени в текущем месяце , фонд рабочего времени часов. Средняя численность исполнителей . Следовательно, для выполнения данной работы достаточно 1 человека.

Студенту необходимо указать перечень таких требований, установленных предприятием-производителем.

**Построение сетевого графика выполнения проекта**

Сетевой график – это графическое представление логической последовательности работ в целях выполнения проекта. Сетевой график устанавливает взаимосвязь между всеми работами проекта и позволяет определить продолжительность как отдельных этапов, так и всего проекта в целом.

Построение сетевого графика предполагает использование метода сетевого планирования, на базе которого разрабатывается информационно-динамическая модель процесса выполнения проекта. Построение сетевой модели включает оценку степени детализации комплекса работ, определения логической связи между отдельными работами и временные характеристики выполнения этапов проекта В сетевой модели выделяют события и работы. В качестве событий, например, принимают факты начала проекта, окончания разработки отдельных модулей, интерфейсов, выполнения отладки и т.п. Все события нумеруются по порядку от исходного к завершающему.

В процессе достижения каждого из событий реализуется определенная последовательность работ. Конечным событием является выполнение всего проекта по разработке программного обеспечения. Каждой работе присваивается «код работы», состоящий из номера наступившего события и номера того события, которое достигается в результате выполнения данной работы, например, если 0 - начало проекта, а 1 - событие "разработка структуры данных завершена", то 0-1 - определяет работу по разработке структуры данных. В качестве работы может выступать и "фиктивная работа", которая определяет ожидание окончания связанных работ и длительность которой равна 0 чел.-дней. Кодовые номера работ каждого этапа указываются в соответствующем блоке строк, относящегося к этому этапу.

Основные события и работы по разработке репозитория инсталляционных пакетов представлены в таблице 54.

Таблица 54. Основные события и работы проекта.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Событие** | **Код работы** | **Содержание работ** | **Трудоемкость,**  **чел/часы.** | **Трудоемкость,**  **чел/дни.** |
| 0 | Начало работ | 0-1 | Анализ требований ТЗ | 4,2 | 0,525 |
| 1 | Проанализированы требования ТЗ | 1-2 | Уточнение требований | 3,8 | 0,475 |
|  | 1-3 | Выбор архитектуры | 12 | 1,5 |
| 2 | Уточнены требования | 2-3 | Фиктивная работа. Ожидание завершения работы 1-3 | 0 | 0 |
| 3 | Анализ завершен | 3-4 | Выбор программной платформы | 9,6 | 1,2 |
| 4 | Выбрана программная платформа | 4-5 | Разработка базы данных | 3,8 | 0,475 |
|  | 4-6 | Разработка классов | 4,8 | 0,6 |
| 5 | Завершена разработка базы данных | 5-6 | Фиктивная работа. Ожидание завершения работы 4-6 | 0 | 0 |
| 6 | Завершена разработка БД и классов | 6-7 | Написание кодов программы на выбранном языке программирования | 170 | 21,25 |
| 7 | Завершено написание кодов программы | 7-8 | Тестирование | 7 | 0,875 |
| 8 | Завершено тестирование | 8-9 | Внесение поправок | 44 | 5,5 |
| 9 | Внесены поправки | 9-10 | Подготовка сопроводительной документации | 19 | 2,375 |

Графическое отображение сетевой модели (сетевой график) содержит окружности, которые отображают основные события проекта, и векторы, соединяющие эти окружности, которые определяют необходимость выполнения соответствующих работ. Реальные работы изображаются сплошной линией, фиктивные – штриховой, а работы, лежащие на критическом пути – линией двойной толщины.

Окружности разделены на четыре сектора, в каждом из которых показаны номер данного события (в нижнем секторе), значение раннего срока наступления текущего события (в левом секторе), значение резерва времени текущего события (в верхнем секторе) и значение позднего срока наступления события (в правом секторе), как показано на рисунке 5.1.

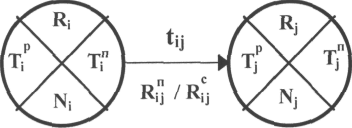


Рисунок 51 – Обозначение основных элементов сетевого графика

Обозначение основных элементов сетевого графика:

* Ni, Nj – номер события;
* – ранний срок наступления события i;
* – поздний срок наступления события i;
* Ri – резерв времени события i;
* tij - продолжительность работы i-j.

В соответствии с содержанием таблицы основных событий и работ проекта строится графическая модель сетевого графика.

Ранний срок совершения события определяет минимальное время, необходимое для выполнения всех работ, предшествующих данному событию и равен продолжительности наибольшего из путей, ведущих от исходного события к рассматриваемому, и рассчитывается при движении по графу слева направо (от начала проекта к его завершению) по формуле:

 (53)

Таким образом, двигаясь по графу слева направо, необходимо для каждого из предшествующих событий выбрать то, для которого сумма раннего срока и длительности работы, соединяющей его с данным событием, максимальна. Результаты расчета приведены ниже на сетевом графе.

Поздний срок совершения события – это максимально допустимое время наступления данного события, при котором сохраняется возможность соблюдения ранних сроков наступления последующих событий. Поздние сроки вычисляются, начиная с последнего события – завершения проекта, по критическому пути (т.е. справа налево по графику). Они равны разности между поздним сроком совершения j-го события и продолжительностью ij работы. Поздний срок совершения события определяется по формуле:

. (5.4)

Резерв времени события определяется по формуле:

. (5.5)

Полный резерв времени работы – это максимальное время, на которое можно отложить начало работы или увеличить ее длительность, чтобы, тем не менее, уложиться в сроки проекта. Полный резерв времени работы определяется по формуле:

 (5.6)

Из формулы (5.6) видно, что превышение опоздания в процессе выполнения работы i-j (или до ее начала) значения (5.6) приведет к «сползанию» дальнейших работ на разность между этим опозданием и полным резервом (5.6), что в конечном итоге означает запаздывание и последнего события, обозначающего полное завершение проекта.

Свободный резерв времени определяется согласно формуле (5.7) и обозначает время, на которое можно отложить начало работы или увеличить ее продолжительность, чтобы, тем не менее, успеть завершить проект в срок, при условии, что все работы будут выполняться по наиболее оптимистичному графику (в свои ранние сроки):

 (5.7)

Формула (5.7) показывает, что даже в случае запаздывания, превышающего значение полного резерва (5.6), возможно уложиться в срок разработки проекта, «наверстав» упущенное время за счет оптимального выполнения последующих работ.

Таким образом:

* если отставание от графика не превышает полного резерва времени, у руководителя проекта нет причин для беспокойства;
* если отставание превышает полный резерв времени, то существует возможность догнать график за счет оптимального выполнения всех оставшихся работ;
* если отставание превышает свободный резерв времени, то догнать график в рамках текущего плана невозможно. Требуются дополнительные ресурсы.

Путь сетевого графика – это непрерывная цепочка работ и событий. Полный путь – сумма всех работ от исходного до завершающего события. Критический путь – это самый продолжительный из полных путей. Критический путь определяет продолжительность всего проекта.

Сетевой график работ по разработке репозитория инсталляционных пакетов представлен на рисунке 52. На данном сетевом графике критическим является путь 0-1-3-4-6-7-8-9-10. Его продолжительность равна 33,825 чел. дням.

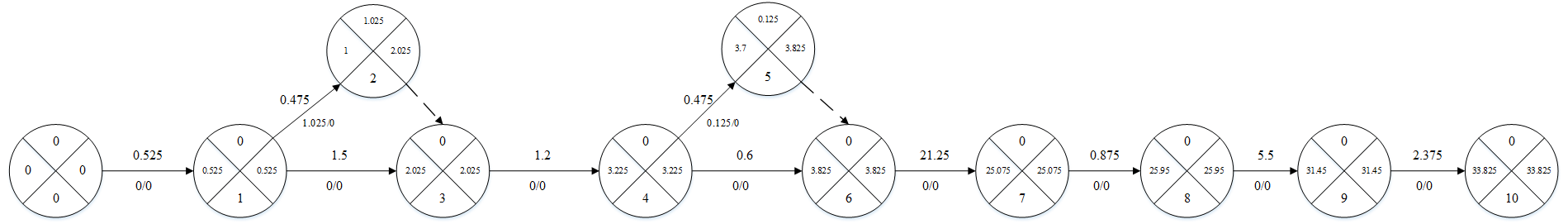


Рисунок 52 – Сетевой график выполнения работ

**Построение календарного графика выполнения проекта**

Для иллюстрации последовательности проводимых работ проекта применяют ленточный график (календарно-сетевой график или диаграмму Гантта). На диаграмме Гантта на оси Х показывают календарные месяцы от начала проекта до его завершения. По оси Y - выполняемые этапы работ.

Если отдельные этапы проекта могут выполняться параллельно различными исполнителями, то они отображается в виде нескольких нумерованных отрезков (или прямоугольников), размещенных на временных интервалах.

Диаграмма Гантта позволяет наглядно представить процесс выполнения работ во времени, показать ответственных за каждую работу. В тоже время, взаимосвязи между отдельными работами в явном виде на диаграмме не отображаются, что создает трудности при необходимости корректировки общего графика работ из-за изменения сроков выполнения какой-либо работы. В таком случае, необходимо вернуться назад и пересмотреть структуру декомпозиции работ и календарные планы для того, чтобы найти место и время вновь появившимся дополнительным видам деятельности

Диаграмма Гантта для работ по разработке репозитория инсталляционных пакетов представлена на рисунке 53.

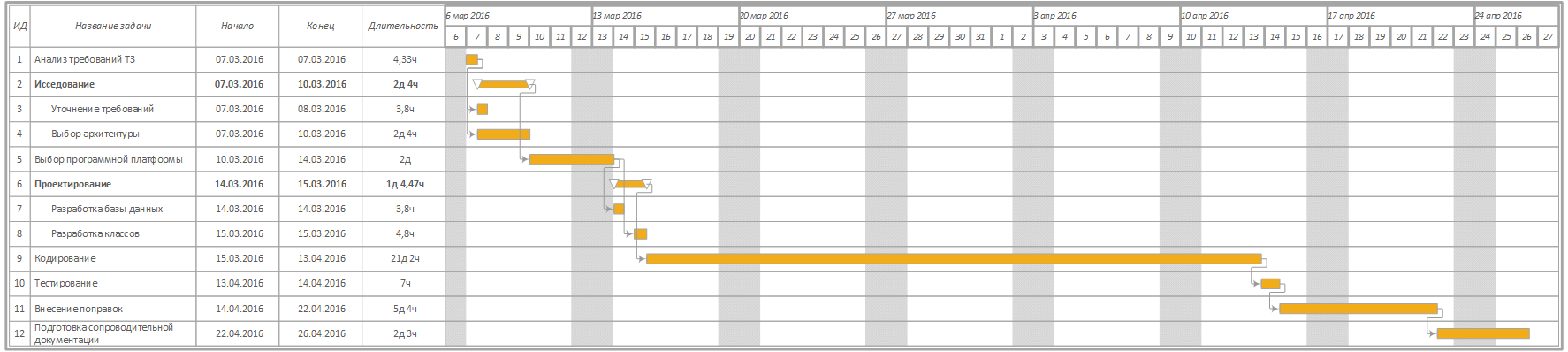


Рисунок 53 – Календарный график выполнения работ (диаграмма Гантта)

**Расчет затрат на разработку проекта**