

Министерство образования и науки, молодежи и
спорта Украины

Харьковский национальный университет

Ревенчук Илона Анатольевна

Программной инженерии

Раздел 9 - Метод критической цепи: эффективное управление проектами с использованием буферов времени и ресурсов

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

Харьков

2005

Содержание

Введение.....	3
Теория.....	4
Введение.....	4
Традиционный подход к управлению неопределенностью и рисками.....	4
Ключевые определения и концепции методов планирования, организации и контроля проектов.....	12
Метод критического пути в многопроектных разработках с учетом ресурсов.....	16
Пооперационный перечень работ (WBS - Work breakdown structure).....	17
Практика.....	33
Пример 1.....	33
Пример 2.....	36
Вопросы для самопроверки.....	40
Выводы.....	43
Перечень ссылок.....	44

Раздел 9 - Метод критической цепи: эффективное управление проектами с использованием буферов времени и ресурсов

В разделе рассматривается метод критической цепи для управления временем и оптимизации ресурсов.

Теория

Введение

Одной из причин выделения управления проектами в отдельную область знаний является неопределенность. То, как мы управляем неопределенностью в проекте (в том числе и рисками), напрямую влияет на длительность проекта, на его успех.

По данным многочисленных исследований Standish Group для традиционных методов управления проектами, только 44% проектов обычно завершаются вовремя. В среднем проекты занимают 222% процента от изначально запланированной длительности, 189% от начального бюджета. 70% проектов сокращают исходный объем работ проекта, 30% проектов закрываются досрочно. И хотя в последнее время, с развитием инструментов и техник управления проектами, эти цифры стали уменьшаться, общая картина говорит о том, что менеджеры проектов плохо делают свою работу.

Традиционный подход к управлению неопределенностью и рисками

При традиционном подходе к управлению проектами, проблемы, связанные с неопределенностью (т.е. с законом Мерфи: Если какая-нибудь неприятность может произойти, - она случается, законом Паркинсона: Работа стремится занять все время, отпущенное на нее) и одновременной работой сотрудников над несколькими задачами (например, в нескольких проектах), решают следующими способами.

Включение в оценку задачи рисков и неопределенностей

Часто и сотрудник, и его руководитель стремятся заложить в оценку задачи неопределенности и риски, которые они предвидят. Неопределенность, к примеру, может быть связана с такими факторами, как новая технология, неопытность исполнителя в области выполнения задачи, недостаток информации о задаче на момент оценки.

Минимизировать риски пытаются тем, что добавляют резервное время для каждой задачи. Поскольку время окончания задачи определяется не одной цифрой, а распределением вероятности, то графически оценку задачи в традиционном управлении проектами можно изобразить так, как показано на рис. 9.1.

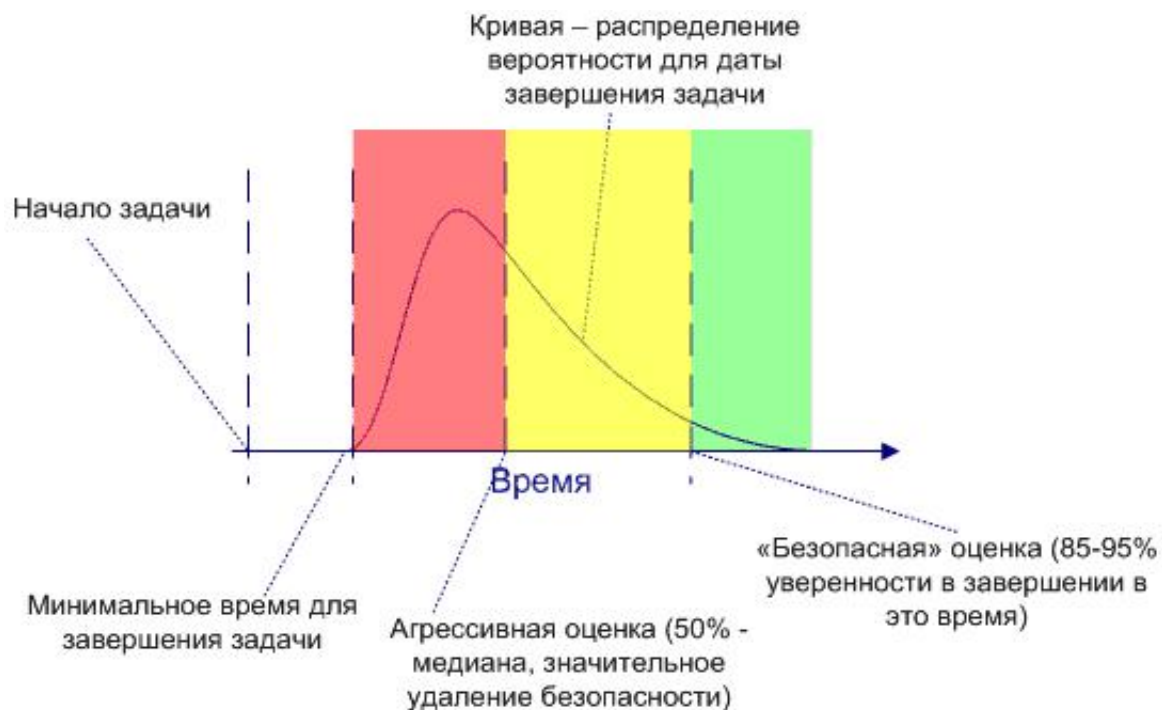


Рисунок 9.1 - Время окончания задачи как распределение вероятности

В оценку задачи включаются риски с целью минимизировать влияние закона Мерфи. И хотя такая оценка делается экспертами или исполнителями оцениваемой задачи (что хорошо и правильно), время на неопределенность часто добавляется «на глаз».

Таким образом, почти каждая задача содержит дополнительный запас прочности, превышающий действительно ожидаемое время завершения данной работы. Часто оценки рисков задачи больше самого времени выполнения работы. Со стороны работника такой подход приводит к следующим негативным тенденциям. Проявляется «синдром студента»: когда работник видит, что у него больше чем достаточно времени для выполнения задачи, он начинает работу позже.

Таким образом, ресурсы выполняют более срочные задачи или тратят время резерва на работу над самой задачей, считая, что все время выделено на работу над ней. И если риски, заложенные в резервном времени, срабатывают, задача запаздывает.

Фокус на запланированных датах начала и окончания задачи

При традиционном подходе к управлению проектами люди стремятся начать и завершить задачу точно в запланированное время, которое содержится в плане и накладывает определенное обязательство на исполнителей.

Этот подход кажется правильным с первого взгляда, но несет в себе значительные недостатки и

не использует возможные положительные события в проекте.

Во-первых, досрочное завершение работ по задаче не приводит к приближению даты окончания проекта. Исполнители, которые должны начать выполнение зависимых задач, не имеют возможности этого сделать, т.к. они заняты выполнением другой работы и не рассчитывали начинать свои задачи ранее запланированного.

Таким образом, раннее завершение задачи не может ускорить запаздывание зависимых от нее задач и положительно повлиять на успех всего проекта. Во-вторых, запаздывание задачи почти всегда приводит к задержке зависимых задач, т.к. на этапе планирования все риски были заложены в самих задачах (см. рис.9.2).

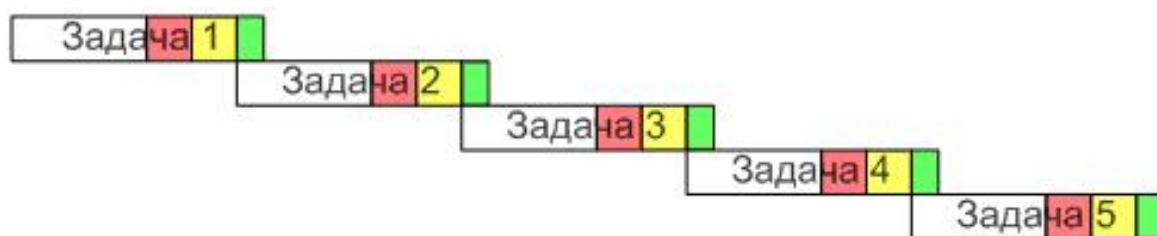


Рисунок 9.2 - Задачи, содержащие в себе значительный резерв по времени, планируются одна за другой

В случае сдвига по времени типичным решением в традиционном управлении проектами является применение корректирующих действий при срабатывании рисков путем урезания объема работ проекта или выделения дополнительных ресурсов. Это не делает счастливым ни заказчика, ни руководителей верхнего уровня.

Управление неопределенностью и рисками с использованием метода критической цепи

Метод критической цепи (МКЦ) был предложен Элияу Голдраттом (Eliyahu Goldratt) в 1997 году.

МКЦ - это метод планирования и управления проектами, который обращает большее внимание на ограничения, связанные с ресурсами проекта.

Он основан на методах и алгоритмах теории ограничений. Этот метод противоположен методам критического пути или PERT в том смысле, что он не предполагает жесткой последовательности задач и жесткого планирования. Напротив, календарный план, составленный с использованием МКЦ, содержит выровненную нагрузку ресурсов по времени, но требует от исполнителей задач быть гибкими по ношению ко времени начала выполнения задач и быстро переключаться между задачами и цепочками задач (но не работать над ними одновременно), с целью удержать весь проект в рамках запланированного времени.

То есть МКЦ предлагает сконцентрировать внимание не на достижении оценок задач и промежуточных вех, а на достижении единственно важной даты - обещанной даты завершения проекта.

МКЦ вводит такое понятие, как критическая цепь задач, или просто критическая цепь. **Критическая цепь** - это последовательность задач, от длительности которых зависит общая длительность всего проекта. МКЦ устраняет перечисленные выше недостатки планирования, выполнения и контроля классического управления проектами с помощью следующих подходов.

Устранение влияния закона Паркинсона

Напомним, закон Паркинсона гласит, что работа займет все время, отведенное на нее, сколько резервного времени мы бы в нее не закладывали.

Работа ресурсов над задачей в традиционном менеджменте проектов занимает все отведенное время ввиду комбинации следующих причин: наличие жестких дат окончания задачи и «безопасных» оценок задачи, включающих резервы времени⁴. Для устранения этих проблем МКЦ предлагает следующие действия:

- Создавать календарный план, используя достаточно плотные оценки длительности задач. Чаще всего в МКЦ в качестве длительности задачи принимается оценка с 50% обеспечением риска, так называемая агрессивная оценка (см. рис.9.1).
- Избавиться от жестких дат окончания задачи (но не проекта). Безусловно, задачи по-прежнему оцениваются и имеют дату окончания в календарном плане. Но эта дата не рассматривается как обязательство исполнителей закончить работу над задачей именно в указанный срок.
- В матричной организационной структуре имеет смысл наделить менеджеров проектов достаточной полнотой власти, чтобы они могли защитить ресурсы проекта от «более срочных» задач других проектов или подразделений. Рассмотрим, что нам дают эти правила.

Поскольку задачи оцениваются с 50% обеспечением риска по времени, то требование МКЦ о построении календарного плана с использованием только времени, необходимого для выполнения задач, выполняется. И т.к. мы убрали обеспечение безопасности из оценок задач, то рассматривать конечные даты завершения каждой задачи как нечто непременно выполнимое, смысла больше не имеет. Таким образом, описанные техники помогают нам избавиться от закона Паркинсона.

Использование положительных моментов досрочного завершения задачи для достижения успеха проекта

МКЦ условно делит ресурсы на две категории:

- ресурсы, выполняющие критические задачи,

- ресурсы некритических задач.

В этом контексте, те ресурсы, о которых мы действительно должны заботиться, - это ресурсы критических задач, т.к. их использование напрямую влияет на длительность проекта. И мы хотим быть уверены, что когда задача в критической цепи завершается, ресурсы для выполнения следующей задачи критического пути будут готовы и доступны.

Два простых шага, выполняемых с известной долей профессионализма, позволят использовать преимущество раннего завершения для задач в критической цепи. Во-первых, необходимо собрать информацию у ресурсов: за сколько нужно их предупредить, что они должны прервать свою текущую работу и переключиться на более важные задачи критической цепи. Во-вторых, требовать, чтобы ресурсы периодически предоставляли оценки о времени, требуемом для завершения их текущих задач («буфер предупреждения»).

Имея эту информацию, мы можем отслеживать когда оценка оставшегося времени текущей задачи критической цепи становится меньше буфера предупреждения исполнителя зависимой задачи, и уведомлять последнего о том, чтобы он был готов вскоре начинать свою задачу. По сравнению с традиционным управлением проектами, это шаг от мониторинга и отчетов вида «что было сделано», используя процент завершения работ (который, кстати сказать, довольно субъективен), к тому, что следует считать имеющим ценность с точки зрения статуса проекта - сколько осталось времени для выполнения незавершенных задач.

Использование буферов времени и ресурсов для предотвращения действия закона Мерфи

Отлично, мы сократили оценки до минимально допустимых, тем самым сконцентрировав все усилия на выполнении задачи. Мы используем преимущества раннего завершения задач. Но если у нас теперь нет «железной» даты, до которой задача должна быть завершена, как мы узнаем, когда ресурс у нас будет доступен для других задач? И, кроме того, как теперь быть с рисками, которые теперь не минимизируются дополнительным временем задач?

Для того чтобы защитить дату окончания всего проекта от вариаций задач, МКЦ использует буферы ресурсов и времени. В этом разделе описывается использование буферов времени. В МКЦ особое внимание уделяется задачам критических цепей. Все, что мы делаем, это аккумулируем резервное время всех задач цепи, которое составляло от 50% до 90% покрытия неопределенности, оставляя для самих задач только 50% покрытия. Эти размазанные по всем задачам резервы суммируются в единый буфер времени, который помещается в конце цепи (см. рис. 9.3).

Таким образом, вариации в критической цепи не имеют прямого влияния на обещанную дату окончания проекта, т.к. они гасятся буфером времени.

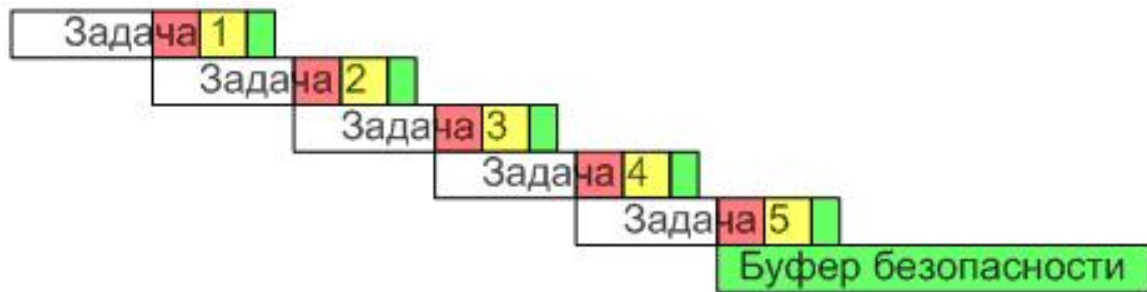


Рисунок 9.3 - Буфер времени, помещенный в конец цепи, предохраняет ее от задержек по времени

Поскольку мы оставляем только 50% покрытия рисков в оценке задачи, мы можем ожидать, что в половине случаев задачи завершатся раньше запланированного, в половине - позже. В МКЦ мы активно используем преимущество раннего завершения задач. Что до запаздывающих задач, то их будет компенсировать буфер цепи. Не вдаваясь в статистику, констатируем, что суммарный буфер должен быть значительно меньше составляющих его отрезков времени от резервов индивидуальных задач (на 30-50%).

Что касается задач некритической цепи, то здесь мы не хотим заниматься микро-менеджментом для каждой задачи и исполнителя, как в случае с критическими цепями, с использованием уведомлений о завершающихся задачах. Тем не менее, мы также хотим, чтобы задачи некритического пути также не повлияли на успех проекта.

Что предполагает традиционный подход к управлению задачами некритического пути? Начинать задачи как можно раньше и надеяться, что запаса времени некритических задач хватит для покрытия рисков.

В отличие от традиционного управления проектами, в МКЦ мы используем не только свободное время (float) для некритических задач, но и тот же подход с буфером в конце цепи (теперь уже некритической), который был описан для критических задач. Этот буфер, назовем его «питающим», предохраняет зависимые критические цепи от вариаций по времени в некритических цепях.

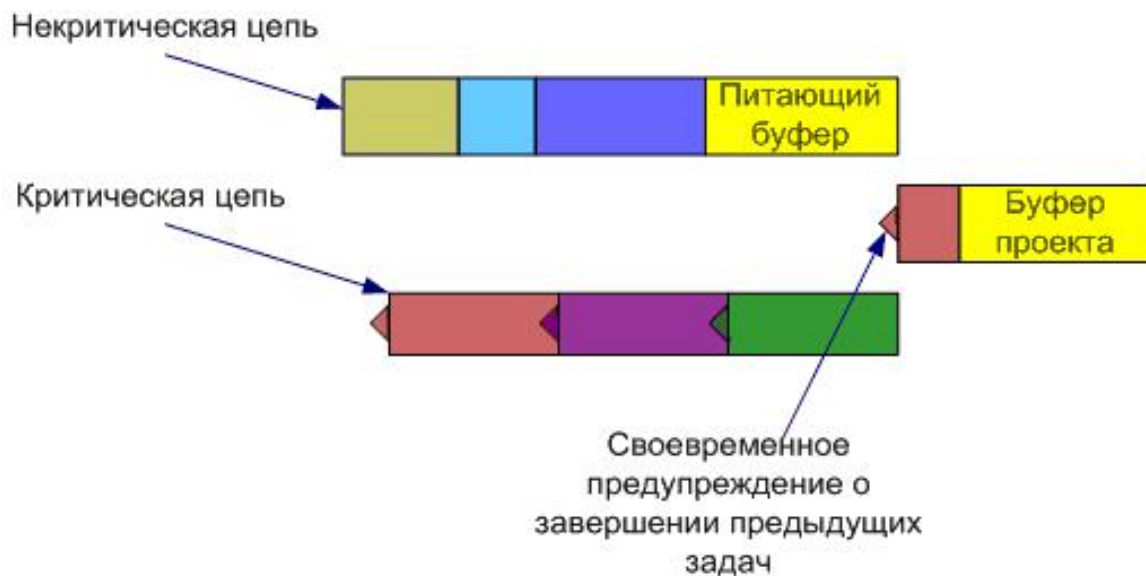


Рисунок 9.4 - Составление расписания по МКЦ

Для некритической цепи мы не используем каких-либо дополнительных инструментов для того, чтобы избежать последствия задержки выполнения задач (например, предупреждение о приближающейся работе). Для таких задач у нас и без того имеется двойной буфер: свободное время некритической цепи (float) и питающий буфер.

Таким образом, составление расписания по МКЦ использует «питающие» и проектные буферы времени, а также буферы ресурсов (описывается в следующем разделе).

Использование буфера ресурсов

Особо интересным инструментом в МКЦ является использование буферов ресурсов. Можно условно выделить два вида таких буферов.

Во-первых, то время, за которое мы предупреждаем исполнителя о том, что скоро у него возникнет задача из критической цепи, и является ресурсным буфером.

Второй тип буфера - это выделение альтернативных (дополнительных) ресурсов для задач критической цепи. Этот буфер имеет смысл, когда задачи могут подвергаться частым изменениям. В этом случае, добавление ресурсов означает защиту от рисков финальной даты окончания проектов.

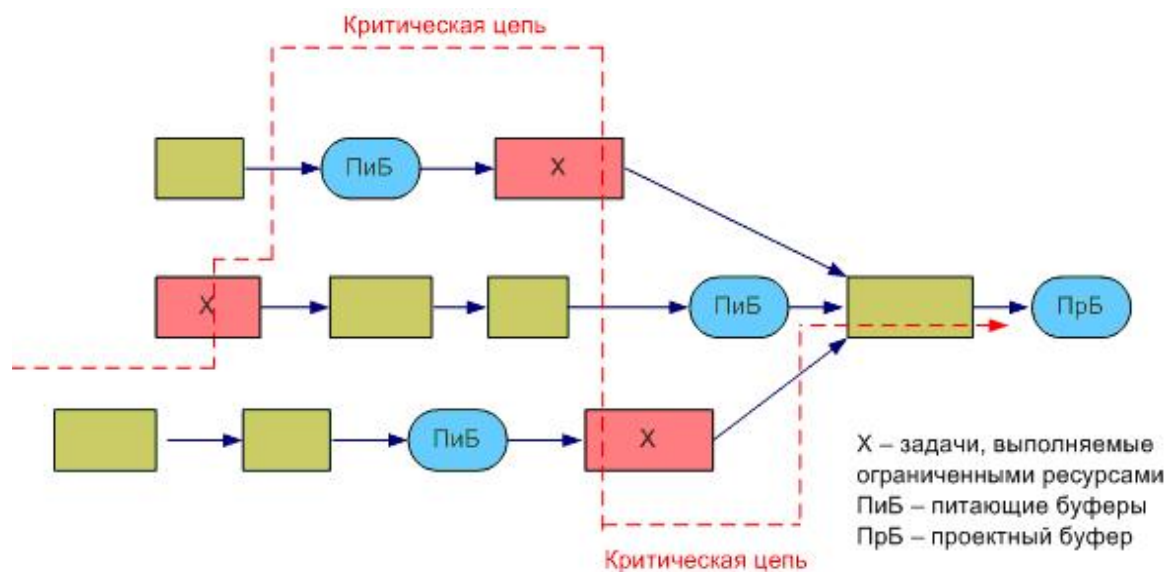


Рисунок 9.5 - Критическая цепь в данном случае состоит из задач, выполняемых ограниченными ресурсами

На рис. 9.5 показана критическая цепь, которая образована ограниченными ресурсами. Соответственно, буферы, используемые в этой ситуации, используются скорее не для того, чтобы предотвратить недостаток времени (неверные оценки). Буферы ресурсов времени в этом случае используются для управления рисками, связанными с ограниченными ресурсами. Питающие буферы стоят перед задачами критических ресурсов, предохраняя таким образом задачи критической цепи от сдвига по времени в случае задержки задач не критических цепей.

Буфер как средство мониторинга и контроля

Прогресс проекта и точность планирования с использованием МКЦ часто отслеживается не по классической технике анализа освоенного объема (earned value analysis), а по проценту использованных буферов. Т.е. чем больше времени, запланированного в качестве буфера, использовано, тем большее влияние неопределенности на проект в виде реализовавшихся рисов. Отслеживание оставшегося буферов времени для задачи используется в МКЦ для мониторинга статуса задачи: при достижении минимального порогового значения буфера необходимо предпринять корректирующие действия. Аналогично, процент использования проектного буфера служит как триггером для определения выполнимости обещанной даты завершения, так и показателем успешности проекта (например, если использовали не более 50% проектного буфера времени, проект следует считать очень успешным).

Практические шаги для использования МКЦ

Суммируя все описанное, можно порекомендовать следующий комплекс практических шагов для использования МКЦ:

- Объяснить исполнителям, что они должны защищать свои оценки от давления со стороны начальства и других заинтересованных сторон проекта.
- В качестве оценки длительности задач брать оценки с 50% покрытием неопределенности.
- Исключить конкуренцию за ресурсы путем выравнивания нагрузки. Это также уберет необходимость переключения ресурсов между задачами. Критическая цепь теперь может быть определена как наиболее длинная цепь пути и зависимости ресурсов.
- Вставить проектный буфер в конце проекта для аккумуляции резервного времени (изначально - примерно 50% от длины пути критической цепи).
- Защитить критические цепи от недоступности ресурсов с помощью буферов ресурсов.
- Рассчитать и расставить питающие буферы для всех путей, от которых зависят критические цепи.
- Планировать задачи, которые не зависят ни от каких других задач, от конечной даты проекта к его началу. Это будет дополнительным обеспечением отсутствия многозадачности ресурсов.
- Отслеживать и добиваться запланированной производительности ресурсов. Исполнители должны работать над задачами так быстро, как это возможно, и отдавать результат своей работы, как только она завершена.
- Предоставлять ресурсам информацию о длительности и об оценочном времени начала задачи, но не о вехах проекта. Это должно заставлять ресурсы отдавать результаты своей работы как только она закончена.
- На этапе выполнения проекта происходит активное управление выделенными буферами времени и ресурсов, чтобы минимизировать задержки выполнения задач, но сохранить реалистичным одно из главных обещаний проекта - дату его завершения.

Таким образом, МКЦ помогает избежать действия закона Паркинсона и, в то же время, защититься от закона Мерфи. Рассмотренный метод концентрирует внимание менеджера проекта на производительности команды, предлагает новый метод отслеживания прогресса проекта на основе использования буферов времени и ресурсов. МКЦ может применяться практически в любых областях, где применяется методология управления проектами, не требуя значительного перестроения процессов в проекте.

Ключевые определения и концепции методов планирования, организации и контроля проектов

Работа в плане проекта представляет некоторую деятельность, необходимую для

достижения конкретных результатов (конечных продуктов нижнего уровня). Таким образом, работа является основным элементом (дискретной, компонентой) деятельности на самом нижнем уровне детализации, на выполнение которого требуется время, и который может задержать начало выполнения других работ. Момент окончания работы означает факт получения конечного продукта (результата работы). Работа является базовым понятием и предоставляет основу для организации данных в системах управления проектами. На практике для ссылки на детальный уровень работ часто используется термин задача. В общем смысле эти два термина являются синонимами. Термин задача, однако, принимает и другие формальные значения в специфических контекстах планирования. Например, в аэрокосмической и оборонной областях задача часто относится к верхнему суммарному уровню работ, который может содержать множественные группы пакетов работ. Далее термин задача используется только в своем общем смысле, как синоним работы.

Веха - событие или дата в ходе осуществления проекта. Веха используется для отображения состояния завершенности тех или иных работ. В контексте проекта менеджеры используют вехи для того, чтобы обозначить важные промежуточные результаты, которые должны быть достигнуты в процессе реализации проекта. Последовательность вех, определенных менеджером, часто называется **план по вехам**. Даты достижения соответствующих вех образуют **календарный план по вехам**. Важным отличием вех от работ является то, что они не имеют длительности. Из-за этого свойства их часто называют **событиями**.

Связи предшествования (логические зависимости) -- отображают природу зависимостей между работами. Большинство связей в проектах относятся к типу "конец-начало", когда последующая работа может начаться только по завершении предшествующей работы. Связи предшествования образуют структуру сети. Комплекс взаимосвязей между работами часто также называют **логической структурой проекта**, поскольку он определяет последовательность выполнения работ.

Сетевая диаграмма (сеть, граф сети, PERT диаграмма) -- графическое отображение работ проекта и их взаимосвязей. В планировании и управлении проектами под термином сеть понимается полный комплекс работ и вех проекта с установленными между ними зависимостями.

Сетевые диаграммы отображают сетевую модель в графическом виде как множество вершин, соответствующих работам, связанных линиями, представляющими взаимосвязи между работами. Этот граф, называемый сетью типа вершина-работа или диаграммой предшествования, является наиболее распространенным представлением сети на сегодняшний день.

Существует другой тип сетевой диаграммы, называемый сетью типа вершина-событие, который на практике используется реже. При данном подходе работа представляется в виде линии между двумя событиями (узлами графа), которые в свою очередь отображают начало и конец данной работы. PERT-диаграммы являются примерами этого типа диаграмм. Хотя в целом различия между этими двумя подходами представления сети незначительны, представление более сложных связей между работами сетью типа вершина-событие может быть достаточно затруднительно, что и является причиной более редкого использования данного типа.

Сетевая диаграмма не является блок-схемой в том смысле, в котором это средство используется для моделирования деловых процессов. Принципиальным отличием от блок-схемы является то, что сетевая диаграмма моделирует только логические зависимости между элементарными работами. Она не отображает входы, процессы и выходы, и не допускает повторяющихся циклов или петель.

Методы сетевого планирования - методы, основная цель которых заключается в том, чтобы сократить до минимума продолжительность проекта. Основываются на разработанных практически одновременно и независимо методе критического пути МКП и методе оценки и пересмотра планов PERT (Program Evaluation and Review Technique). Первый метод разработан в 1956 году для составления планов-графиков крупных комплексов работ по модернизации заводов фирмы "Дюпон". Второй метод разработан корпорацией "Локхид" и консалтинговой фирмой "Буз, Аллен энд Гамильтон" для реализации крупного проекта разработки ракетной системы "Поларис"

Критический путь - максимальный по продолжительности полный путь в сети называется критическим; работы, лежащие на этом пути, также называются критическими. Именно длительность критического пути определяет наименьшую общую продолжительность работ по проекту в целом. Длительность выполнения всего проекта в целом может быть сокращена за счет сокращения длительности задач, лежащих на критическом пути. Соответственно, любая задержка выполнения задач критического пути повлечет увеличение длительности проекта. Концепция критического пути обеспечивает концентрацию внимания менеджера на критических работах. Однако, основным достоинством метода критического пути является возможность манипулирования сроками выполнения задач, не лежащих на критическом пути.

Метод критического пути позволяет рассчитать возможные календарные графики выполнения комплекса работ на основе описанной логической структуры сети и оценок продолжительности выполнения каждой работы, определить критический путь проекта.

Временной резерв или запас времени - это разность между самым ранним возможным сроком завершения работы и самым поздним допустимым временем ее выполнения. Управленческий смысл временного резерва заключается в том, что при необходимости урегулировать технологические, ресурсные или финансовые ограничения проекта он позволяет менеджеру задержать работу на это время без влияния на общую продолжительность проекта и продолжительность непосредственно связанных с ней задач. Работы, лежащие на критическом пути, имеют временной резерв, равный нулю.

Диаграмма Ганта - горизонтальная линейная диаграмма, на которой задачи проекта представляются протяженными во времени отрезками, характеризующимися датами начала и окончания, задержками и возможно другими временными параметрами.

Структура Разбиения Работ -- иерархическая структура последовательной декомпозиции задач проекта на подзадачи. Структура разбиения работ (СРР) является изначальным инструментом для организации работ, обеспечивающим разделение общего объема работ по проекту в соответствии со структурой их выполнения в организации. На нижнем уровне детализации выделяются работы, соответствующие детализированным элементам деятельности, отображаемым в сетевой модели. СРР предоставляет иерархический формат,

который помогает разработчику в:

- структуризации работ на основные компоненты и подкомпоненты,
- обеспечении направленности деятельности на достижение всего комплекса целей,
- разработке системы ответственности за выполнение работ проекта разработке системы,
- отчетности и обобщения информации по проекту.

Структурная Схема Организации (ССО) имеет формат подобный формату СРР. Каждому элементу нижнего уровня в СРР должны соответствовать один или несколько элементов из ССО. Таким образом, ССО является средством определения ответственных за выполнение работ в сложных организациях и обеспечивает основу для разработки структуры системы отчетности.

Ресурсы - обеспечивающие компоненты деятельности, включающие исполнителей, энергию, материалы, оборудование и т.д. Соответственно, с каждой работой можно связать функцию потребности в ресурсах.

Назначение и выравнивание ресурсов. Методики назначения и выравнивания ресурсов позволяют менеджеру проанализировать сетевой план, построенный с помощью метода критического пути с тем, чтобы обеспечить доступность и использование определенных ресурсов на протяжении всего времени выполнения проекта. Назначение ресурсов состоит в определении потребности каждой работы в различных типах ресурсов.

Методики выравнивания ресурсов представляют собой как правило, программно-реализованные эвристические алгоритмы планирования при ограниченных ресурсах. Эти средства помогают менеджеру создать реальное расписание проекта, с учетом потребности проекта в ресурсах и фактически доступных в данный момент времени ресурсов

Ресурсная гистограмма - гистограмма, отображающая потребности проекта в том или ином виде ресурсов в каждый момент времени

Ресурсное календарное планирование - планирование сроков начала работ при ограниченных наличных ресурсах. Проверка ресурсной реализуемости календарного плана требует сопоставления функций наличия и потребности в ресурсах проекта в целом. Сдвигая не критические работы вплоть до их поздних сроков начала (окончания), можно видоизменить ресурсный профиль, обеспечивая оптимальное использование ресурсов.

Информация, полученная в результате ресурсного анализа проекта, помогает заострить внимание менеджера и членов команды на тех моментах работ, где эффективное управление ресурсами будет являться ключевым фактором успеха.

Анализ реализуемости проекта - понятие реализуемости имеет ряд своих разновидностей: логическая реализуемость (учет логических ограничений на возможный порядок выполнения работ во времени); временной анализ (расчет и анализ временных характеристик работ: ранняя/поздняя дата начала/окончания работы, полный, свободный временной резерв и другие); физическая (ресурсная) реализуемость (учет ограниченности наличных или доступных

ресурсов в каждый момент времени выполнения проекта); финансовая реализуемость (обеспечение положительного баланса денежных средств как особого вида ресурса).

Исходный план - план выполнения работ проекта, содержащий исходные сведения об основных временных и стоимостных параметрах работ, который принят к исполнению. В исходном плане обычно фиксируются объемы работ, плановые даты начала и окончания задач проекта, длительности задач, расчетные стоимости задач.

Метод критического пути в многопроектных разработках с учетом ресурсов

Одним из важнейших вопросов, возникающих при создании автоматизированных систем управления сложными системами, является выбор приемлемых моделей для описания управляемых процессов, наиболее приемлемой моделью является сетевая модель комплекса работ.

Постановки оптимизационных задач можно разбить на два основных класса:

- при заданных ограничениях на ресурсы минимизация суммарного отклонения расчетных сроков выполнения целевых событий от директивных;
- оптимального в определенном смысле использования ресурсов при заданной продолжительности комплекса работ (равномерная загрузка ресурсов).

При этом нет принципиальной разницы в подходах к решению указанных классов задач. В обоих случаях прежде чем приступить к распределению ресурсов производят расчет сетевых графиков, каждый из которых построен на основании технологии и принятой организации операций по каждому проекту. Затем полученные таким образом показатели (чаще всего полные резервы времени операций, либо их начальные сроки) используются для определения коэффициентов приоритетности с целью обеспечения операций ресурсами и получения расписания их выполнения в планируемом периоде.

Метод критического пути представляет собой метод анализа сети расписания, проводимого при помощи модели расписания.

При методе критического пути рассчитываются теоретические даты раннего старта и раннего финиша и позднего старта и позднего финиша для всех плановых операций без учета ограничений по ресурсам. Этот расчет производится путем проведения анализа прямого и обратного прохода по путям сети расписания проекта. Полученные даты раннего и позднего старта и финиша не обязательно представляют собой расписание проекта; они скорее показывают периоды времени, в пределах которых следует планировать данную операцию, исходя из длительности операций, логических взаимосвязей, опережений, задержек и прочих известных ограничений. Рассчитанные ранний старт и ранний финиш и поздний старт и поздний финиш могут быть, а могут и не быть одинаковыми на любом пути в сети, поскольку общий временной резерв, обеспечивающий гибкость расписания, может быть положительным,

отрицательным и равным нулю.

На любом пути в сети гибкость расписания измеряется по положительной разности между ранними и поздними датами и называется "общим временным резервом".

У критических путей общий временной резерв может быть отрицательным или равным нулю, а плановые операции на критическом пути называются **"критическими операциями" или "работами"**.

Для получения сетевых путей с положительным или нулевым общим временным резервом могут потребоваться корректировки длительности операций, логических взаимосвязей, опережений и задержки прочих ограничений. Как только общий временной резерв на пути в сети оказывается нулевым или положительным, можно также определить т.н. **свободный временной резерв** - количество времени, на которое плановая операция может быть отложена, не вызывая задержки раннего старта непосредственно примыкающей последующей операции на данном сетевом пути.

Таким образом, **критический путь** - это задача (или последовательность задач), определяющая дату окончания проекта (максимальная сумма длительностей работ).

Длительность проекта будет увеличиваться или уменьшаться вместе с длительностью задачи, лежащей на критическом пути.

Согласно теореме о критических путях сетевой модели, отображающей многопроектную разработку с учетом ресурсов следует, что в данном случае проблема состоит не в получении календарного плана работ на основании известных сетевых графиков, каждый из которых определяется технологией и принятой организации работ в проекте, а в определении сетевой модели, отображающей многопроектную разработку с учетом ресурсов (ресурсного графа) при соблюдении технологии проектирования каждого проекта в качестве ограничения.

Пооперационный перечень работ (WBS - Work breakdown structure)

Введение

Структура пооперационного перечня работ (Work breakdown structure, WBS) является основой плана проекта, на его основе проектируются другие компоненты проекта.

Структура WBS представляет собой инструмент, применяемый для документирования всех рабочих операций, которые должны быть выполнены при разработке и поставке ПО надлежащим образом. Хотя иногда может показаться, что информация, поддерживаемая этой структурой, является избыточной по причине наличия других документов, создаваемых в ходе осуществления программного проекта (область применения работ, спецификация требований ПО, документы по разработке проекта и т.д.).

Однако это не так, поскольку структура WBS консолидирует информацию из различных источников, организуя ее с применением единого формата, удобного при планировании, оценивании и отслеживании.

Структура WBS образует своего рода каркас, на основе которого разрабатывается график выполнения проекта. Благодаря этому можно переходить от действий верхнего уровня в рамках проекта (действия типа сделать) через ряд упрощенных и сравнительно незначительных действий, предназначенных для создания поставляемых продуктов проекта, до тех пор, пока не будут выбраны легко управляемые действия. В результате выбора подобной модели действий каждый пользователь сможет разобраться во взаимосвязях между задачами и действиями, воспользовавшись простыми и "прозрачными" методами.

Структура WBS представляет собой иерархический перечень рабочих действий, необходимых для завершения проекта. В этот перечень включаются управленческие, административные, интегральные и программистские действия, с помощью которых:

- выполняется разработка ПО;
- происходит управление проектом;
- обеспечивается поддержка для всех действий, выполняемых в ходе осуществления проекта;
- выполняются любые другие действия, требуемые для достижения целей проекта и удовлетворения требований пользователей, например, создание документов, разработка учебных программ, инструментальных средств разработки, закупки, командировки и т.д.

Структура WBS представляет собой описание выполняемой работы, разбитой на отдельные ключевые компоненты вплоть до самого нижнего уровня.

Таким образом, при разбиении проекта на отдельные управляемые части размер каждого компонента может быть изменен, а также возможна оценка трудозатрат, понесенных на этапе разработки этого компонента. При разработке программных проектов подавляющая часть трудозатрат по разработке напрямую связана с размером и сложностью программного продукта.

На рис. 9.6 показано, каким образом реализуются этапы что и как при выполнении программного проекта. На этой схеме структура WBS, ориентированная на разрабатываемые продукты, идентифицирует действия на уровне применимости персоналом, обладающим необходимыми в данном случае навыками. Как только будет определено количество участников команды, а также составлен перечень требуемых навыков, оценки трудозатрат могут применяться наравне с составлением графика с целью определения длительности выполнения проекта, а также момента достижения определенных этапов проекта.

Результатирующий график выполнения проекта обычно отображается в виде диаграмм Ганта. В этот момент времени завершается этап планирования как, после чего могут начинаться работы по выполнению проекта. В процессе выполнения каждое действие из структуры WBS может отслеживаться в масштабе всего проекта.



Рисунок 9.6 - Действия по определению размеров и оценке ПО попадают в центральную область последовательности задач по планированию программных проектов.

Структура WBS, ориентированная на конечные продукты, отображает этапы процесса разработки продукта, детализованные на уровне его компонентов. Эта структура реализует управление планированием на этапах что и как, а также закладывает основы отслеживания рабочих действий, величины затрат и хода выполнения графика на этапе выполнения. При этом руководитель работ получает возможность взглянуть на происходящее с "глобальной точки зрения". Подобная структура отображает "содержание" работ, выполняемых в ходе осуществления проекта. Учитывая эти обстоятельства, можно заключить, что данный инструмент является весьма ценным в работе менеджеров проектов.

Ранее уже упоминалось о том, что тройные ограничения для проектов включают область действия, график и затраты. Эти ограничения напрямую связаны с размером ПО, сроком завершения работ и трудозатратами, связанными с распределением ресурсов. Изначально

выполняется идентификация цели и области действия. Затем рассматриваются трудозатраты и рабочий график. Как упоминалось ранее, для выполнения бюджетных и детализированных оценок работы, а также для разработки реального графика требуется структура WBS, ориентированной на продукты. Потребность в ней особенно ощущается при выполнении следующих трех действий в рамках проекта.

Оценка затрат:

- обеспечение оценки выполняемых действий;
- обеспечение соответствия каждого элемента оценки рассматриваемому действию;
- "пересчет" затрат на производство отдельных элементов в итоговые суммы затрат на производство подэлементов, а также системы в целом.

Счета затрат:

- обеспечивают распределение работ и "зарядку" соответствующих центров затрат на основе специфических элементов структуры WBS;
- позволяют детализировать фактические затраты.

Выполнение графика:

- обеспечивается отслеживание завершенных действий;
- возможна оценка хода выполнения проекта.

Структура WBS может быть связана со счетами затрат проекта, позволяющих выполнять контроль, а также со сметой затрат организации, созданной с целью управления выполняемыми работами. Благодаря структуре WBS эти объекты могут отображаться наравне со специфическими элементами, включенными в рабочий график на завершающих этапах.

Структура пооперационного перечня работ может быть описана различными способами: в виде дерева (рис.9.7) и в виде списка (рис.9.8).

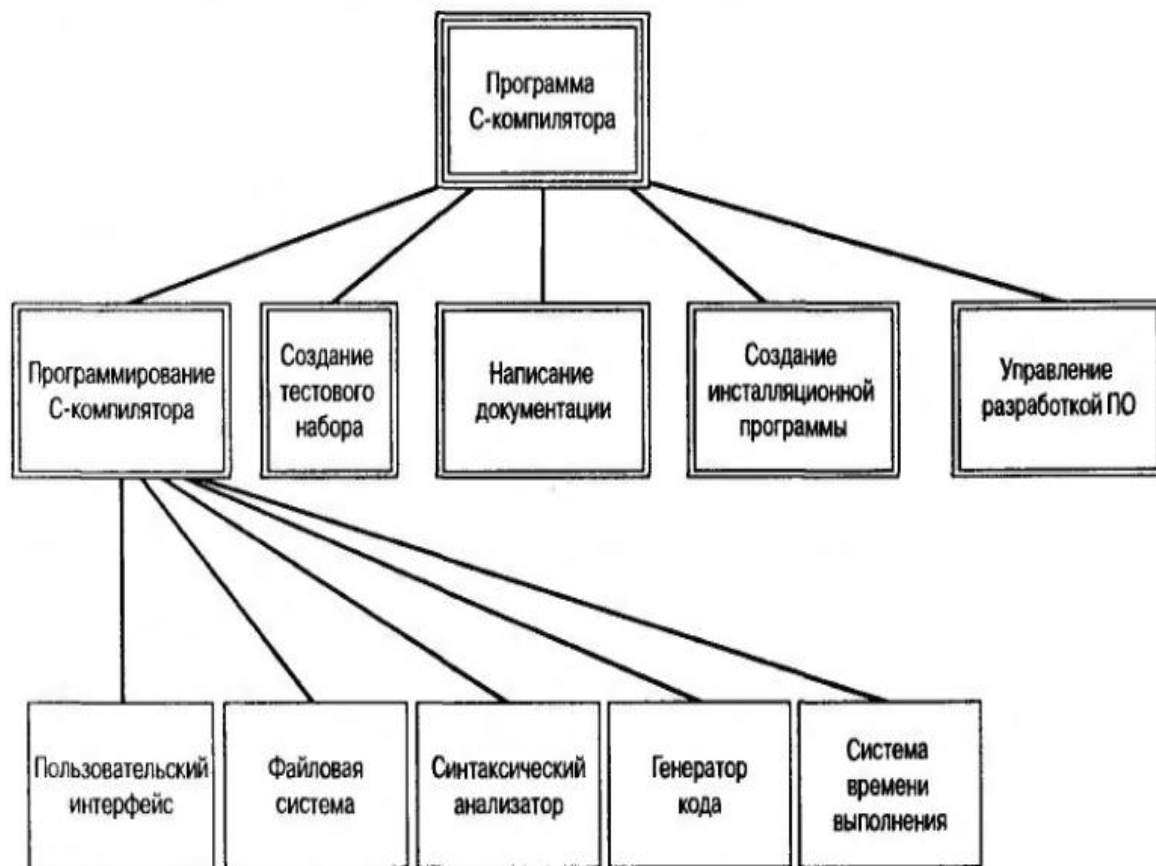


Рисунок 9.7 - Пример структуры пооперационного перечня работ, отображенной в виде дерева

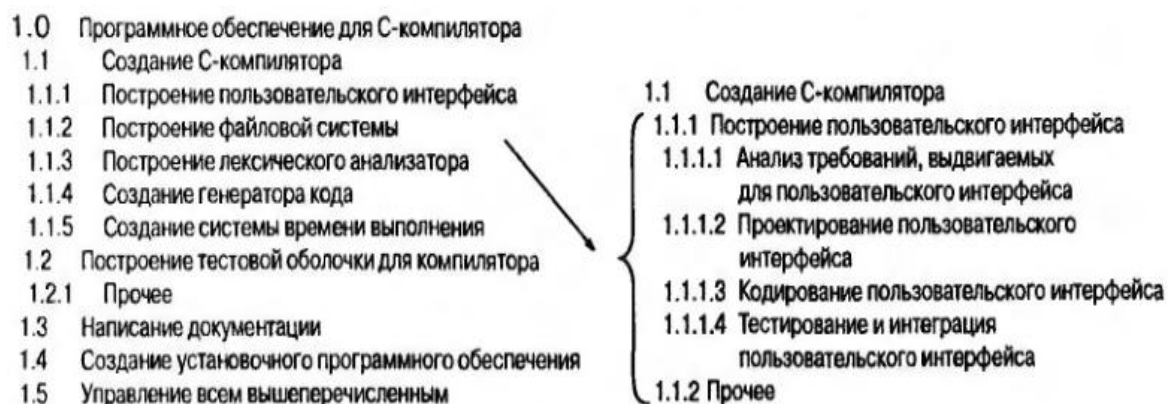


Рисунок 9.8 - Пример структуры пооперационного перечня работ, представленной в виде списка с отступами

Обратите внимание на схему нумерации из рис. 9.8, применяемую для пометки элементов структуры WBS. Эта схема является весьма полезной и позволяет отличать такие обобщенные

действия как разработку проекта, кодирование и тестирование от многих других программных компонентов. В рассматриваемом примере действие 1.1.1.3 применяется для разработки пользовательского интерфейса, а действие 1.4 применяется для создания сценария программы установки. Благодаря наличию подобных особенностей обеспечивается требуемая доля специфичности при пометке действий проекта.

Используемая в данном случае схема нумерации устраняет двусмысленность при сопоставлении меток действиям проекта. Большая часть программ, применяемых для составления графиков проектов, могут поддерживать автоматическую расстановку отступов и структуру WBS либо структурную нумерацию. Также обратите внимание на то, что подобная работа должна быть выполнена на различных уровнях структуры WBS.

Структура WBS может быть создана для двух наиболее общих представлений продукта.

- Представление продукта указывает иерархические взаимосвязи среди элементов продукта (подпрограммы, модули, подсистемы и т.д.). Однако не путайте эту структуру с материальной ведомостью.
- Представление проекта указывает на иерархические взаимосвязи среди рабочих действий (элементы процесса). Часто распределяется среди организационных линий.

Оба набора элементов и действий следует учитывать при вычислении размера и оценке создаваемого ПО.

Как правило, для каждого различного жизненного цикла существует уникальный пооперационный перечень работ, который может использоваться в самой организации. Даже если в большинстве жизненных циклов разработки ПО применяются многие общие действия, такие как декомпозиция системных требований, выполнение разработки архитектуры, создание тестовых данных и управление проектом, порядок выполнения этих действий для каждого жизненного цикла может быть совершенно различным. Некоторые действия могут пропускаться в определенных ситуациях, например, при разработке прототипов, либо в случае миниатюрных проектов. Во многих организациях создаются структуры WBS с целью сравнения предварительно определенных жизненных циклов для процессов и действий (на базе шаблона, определяемого с помощью стандарта IEEE 1074), которые являются типовыми для определенных видов проектов. К подобным проектам относится разработка новых системных Web-приложений, расширения возможностей существующих систем либо специализированные базы данных. Во многих случаях применяется классификация в соответствии с размером проекта: большой, средний либо малый.

Методы создания структуры WBS

Структура WBS может создаваться с применением самых различных методов, но, как правило, лучше всего ранжировать действия в соответствии с основными рабочими и поставляемыми продуктами, разработанными в соответствии с требованиями заказчиков.

Здесь необходимо различать рабочие продукты (что-либо материальное, производимое в ходе выполнения проекта, например, техническое задание, документы:

- SOW (Statement of work) - рабочий план,
- SPMP (Software project management plan) - план управления программными проектами,
- SRS (Software requirements specification) - спецификация требований по разработке ПО,
- SDD (software design description) - описание разработки программного проекта, код, наборы тестов и т.д.), и поставляемые продукты (рабочие продукты, модули исполняемого кода, документация и т.д.). Обратите внимание на рис. 9.7, где иерархическое информационное "дерево" включает:
- компилятор;
- основные части компилятора;
- этапы проектирования основных составляющих частей компилятора.

Первый из перечисленных объектов представляет собой поставляемый продукт, второй - основные составляющие части рабочего продукта, применяемые поставляемыми продуктами, третий - действия, которые могут применяться при создании основных компонентов. Данная прогрессия демонстрирует, каким образом действия в разветвлениях структуры WBS соотносятся с поставляемыми продуктами. Организация структуры WBS вокруг поставляемых продуктов и выполняемых при их производстве действий способствует тому, что необязательная работа, не будет выполняться в ходе выполнения проекта.

Хотя организация структуры WBS вокруг рабочих и поставляемых продуктов является неплохим способом, позволяющим избежать планирования дополнительной работы в рамках проекта, возможен, также другой вариант. Иногда более удобным является сегментирование работы путем ее организации на высшем уровне, а не на уровне рабочих и поставляемых продуктов. При этом также оценивается степень контроля проекта при условии наличия организаций, которые ответственны за разработку компонентов заключительного рабочего продукта.

В результате расширяются возможности команды, и улучшается контроль над выполняемым программным проектом. Дополнительные варианты приводятся на рис. 9.9. Они могут применяться в качестве базисных элементов проекта. Могут также использоваться другие элементы, например, поставляемые продукты, навыки, виды ответственности либо хронология. В иллюстрируемых на рис. 9.9 методах предполагается, что имеет место декомпозиция на системном уровне, а рассмотрению подвергаются элементы ПО. В этом случае пооперационный перечень работ может включать аппаратное обеспечение, ПО, а также деловые процессы.

Подход А	Подход Б
Система ПО Продукты Компоненты Этапы процесса	Система ПО Организации Продукты Этапы процесса
Подход В	Подход Г
Система ПО Этапы процесса Продукты Компоненты	Система ПО Продукты Организации Этапы процесса

Рисунок 9.9 - Некоторые возможные варианты структуры пооперационного перечня работ

Структура WBS может создаваться в направлении "сверху-вниз" либо в направлении "снизу-вверх".

Подход "сверху-вниз" влечет за собой последовательную декомпозицию.

Подход "снизу-вверх" применяется метод "мозгового штурма" и диаграммы связности.

Обычно в начале осуществления проекта на этапах что и как применяется подход "сверху-вниз". Зачастую при определении структуры WBS используется представление в форме дерева, поскольку в этом случае количество уровней будет ограниченным. Это представление является весьма удобным при отображении документов, связанных с техническим заданием проекта. В этом случае менеджеру проекта значительно проще идентифицировать большие рабочие компоненты, которые будут выполняться в дальнейшем, а также определить приближенный **порядок разброса оценок выполняемой работы (Order of magnitude, ROM)**. Подобные оценки обычно основываются на широко распространенном "правиле большого пальца", заключающемся, например, в экстраполяции, использующей сведения о предыдущих программных проектах либо о размерах (большой, средний, малый) и типе прогнозируемого жизненного цикла разработки ПО.

Подход "сверху-вниз" часто применяется в том случае, когда работа была выполнена преждевременно, а команда разработчиков проекта хорошо осознала этапы выполняемого проекта. При использовании этого подхода разработка начинается с самого верхнего элемента (обычно поставляемый программный продукт, например, "завершенное ПО С-компилятора"). Затем производится дальнейшая разбивка, причем на каждом последующем уровне возрастает достигаемый уровень детализации. Разбиение продолжается до тех пор, пока фрагменты

работы не смогут быть выполнены "одной единицей ресурса" за "относительно короткий период времени".

В качестве "одной единицы ресурса" может выступать один человек, одна секция, один отдел либо другой элемент организационной структуры, который может выполнять работу. "Относительно короткий период времени" может означать один день, неделю, две недели, месяц либо другую единицу времени, обладающую приемлемым уровнем детализации в масштабах области действия проекта, позволяющим производить измерения во время выполнения проекта. Как правило, в программных проектах происходит разбиение выполняемой работы до тех пор, пока она не будет выполняться одним человеком либо группой за период времени, равный одной либо двум неделям. При этом производится одна единица рабочего продукта.

В результате поддерживается равновесие, предотвращающее эффект "большого взрыва" при управлении поставляемыми продуктами.

Подход "снизу-вверх" идеально подходит при разработке проектов новых типов, когда команда разработчиков плохо ознакомлена с этапами, которые им предстоит реализовать на практике. Применение этого подхода на практике обычно означает использование методики "мозгового штурма" по отношению ко всему, что должно выполняться в рамках проекта. Затем происходит группировка действий с одинаковыми уровнями детализации (при этом выполняется приблизительная оценка объема выполняемой работы). Также выполняется группирование всех действий до тех пор, пока не будет достигнут элемент высшего уровня. Этот процесс именуется созданием диаграмм связности.

При использовании каждого из этих подходов следует избегать чрезмерной детализации, которая может привести к созданию чрезвычайно громоздких планов. Не планируйте слишком много деталей. Например, если на разработку определенного модуля кода отводится две недели, дальнейшее разбиение плана на детализированные этапы, включающие идентификацию элементов данных, создание диаграмм потока данных, а также обзор заказчиков являются излишними. В то же самое время менеджер проекта полагается на инженерное образование разработчиков проекта.

Распространенной ошибкой в этой ситуации является продуцирование чрезмерного количества деталей на начальных этапах планирования. Как только будет выполнено описание, достаточное для четкого инструктирования участников разработки, а также реализована приемлемая оценка общего объема трудозатрат, необходимо остановиться. Сначала нужно распределить (делегируют) действие, позднее потребуется завершить планирование.

Воспользуйтесь услугами команд и подгрупп для выполнения работы по частям.

Распределите работу таким образом, чтобы организационные единицы в целом несли ответственность за выполнение действий на нижнем уровне структуры WBS. Также во избежание затруднений в организации и управлении, попытайтесь ограничить глубину дерева либо списка с отступами таким образом, чтобы эти объекты были хоть в какой-то степени управляемыми. Слишком большое количество подуровней структуры приводит к путанице. Старайтесь придерживаться принципов максимальной простоты, оставляя при этом возможность выражения структуры и иерархии работы в рамках проекта.

Определение стадий проекта

Стадия - значимое событие проекта, обычно связанное с основным рабочим продуктом либо поставляемым продуктом.

Эти события служат своего рода "путевыми знаками" на пути к завершению проекта. Причем в состав каждого проекта должно входить достаточное количество стадий, равномерно распределенных в графике, благодаря чему может легко измеряться прогресс в направлении достижения конечной цели. Если количество стадий невелико, мы будем иметь дело с так называемым проектом "большого взрыва", когда никто толком не будет знать срок завершения проекта, для которого составлен график.

Однако слишком большое количество стадий может привести к чрезмерному замедлению темпов выполнения проекта. Стадии имеют нулевую длительность, поскольку просто отмечают момент времени, в который завершается некий важный этап проекта. Стадии могут определять завершение одного либо большего количества действий. Также они могут определяться для рабочего или поставляемого продукта или для определенной группы из указанных выше объектов. С целью индикации времени завершения лучше всего использовать определенные возможности языка, например, "готово" либо "завершено".

Для примера с С-компилятором на рис. 9.7 стадия для элемента 1.0 может иметь вид "разработка ПО С-компилятора завершена".

Фазы не эквиваленты стадиям, а представляют собой наборы взаимосвязанных действий по разработке продукта. Стадии могут применяться для отметки времени завершения фазы

Проектирование рабочих пакетов

В основном вся работа выполняется на нижнем уровне структуры WBS. Здесь находятся "листья", формирующие дерево, или действия, которым соответствуют наибольшие по величине отступы (при использовании списков с отступами). Этот уровень именуется рабочим пакетом, который, как правило, является результатом производства рабочего продукта. Все рабочие пакеты проекта отличаются друг от друга.

В пакете обычно описывается вся необходимая информация, необходимая для выполнения специалистом данного вида работы. На уровне программных проектов рабочие пакеты обычно соответствуют наименьшим идентифицируемым объектам либо модулям, которые создаются в рамках поставляемой системы.

Рабочий пакет может содержать следующую информацию:

- описание ожидаемого рабочего продукта-производимые элементы ПО;
- требования к персоналу - какого рода специалисты и в каком количестве требуются для выполнения данного действия;

- имена ответственных лиц- персональная ответственность за ход выполняемых работ;
- указанная в графике дата начала и завершения выполнения какого-либо действия;
- выделенный бюджет (деньги, часы либо другие единицы измерения) - оценка трудозатрат при выполнении действия;
- критерии приемки работы -уровень дефектов либо другой вид оценки достигаемого уровня качества.

Если персонал разработчиков проекта уже знаком с упомянутыми выше моментами, что является общепринятой практикой в организациях, где группа инженеров - программистов работают вместе изо дня в день, создание формального рабочего пакета вовсе не будет обязательным, поскольку каждый уже ознакомлен со всеми задачами, выполнение которых крайне необходимо для завершения действия.

Однако, если организовывается новая группа сотрудников, которые никогда не работали вместе ранее, потребуется больший уровень контроля. Усиленный контроль также требуется в случае отсутствия общепринятой практики создания предварительных схем программных проектов.

В этом случае регистрация информации о рабочих пакетах является весьма полезной, поскольку в этом случае назначение этих пакетов становится очевидным.

Регистрация подобной информации может происходить также в том случае, если рабочий пакет проекта определяет область действия работы в рамках целого подпроекта, который подвергается дальнейшему разбиению на большее количество действий, управляемых подобно проекту, но с приложением больших усилий. Подпроект от самостоятельного проекта отличается тем, что не существовать самостоятельно, вне контекста большего проекта. Поставляемый рабочий продукт подпроекта может представлять собой автономную часть ПО, используемую в контексте, лежащем вне области действия текущего проекта. Хорошим примером программного подпроекта является проект по созданию пользовательских программных утилит. В большинстве инструментов календарного планирования, обеспечивающих управление проектами, таких как Microsoft Project, помечать сведения о каждом выполняемом действии.

Создание структуры WBS при разработке ПО

Структура WBS является ключевым рабочим продуктом, необходимым при выполнении оценок в рамках программного проекта. Во многих проектах наибольший вред наносит не плохая оценка чего-либо, а то, что оценка вообще не выполняется. При подготовке к процессу оценки размеров выполняемой работы следуйте указаниям, изложенным в данной главе. Ранее уже упоминалось о том, что при построении структуры WBS применяются два подхода, "сверху-вниз" и "снизу-вверх".

При планировании любого проекта следуйте простому правилу: если элемент является слишком сложным для управления, он представляет собой перечень более простых элементов.

Ниже приводятся пять этапов, применяемых при конструировании структуры WBS для программного проекта.

1. Идентифицируйте работу, связанную с разработкой программного продукта (отделите ее от работы, связанной с аппаратным обеспечением, и от рабочих процессов).
2. Найдите структуру WBS для произвольной системы высшего уровня (отделите ПО от других систем и компонентов).
3. Определите программную архитектуру WBS (идентифицируйте все части и действия, требуемые при ее разработке).
4. Наполните содержимым программную архитектуру WBS (идентифицируйте все части и действия, требуемые при ее проектировании).
5. Определите категории затрат, связанных с ПО (подготовка к действиям по оцениванию).

Создание структуры пооперационного перечня работ

Действия, связанные с разработкой ПО. Просмотрите всю доступную документацию (по крайней мере, ту, которая доступна на момент выполнения данного проекта) и составьте полный перечень всех элементов, влияющих на понесенные затраты.

На этом этапе доступными должны быть следующие исходные документы.

- документ SOW (обычно наилучший документ, требуемые для начала работы);
- спецификации, концепция деятельности;
- любые документы, определяющие требования;
- документы по разработке проекта;
- стандарты (внутренние и внешние);
- результаты переговоров с заказчиками;
- проверочные критерии либо ожидания.

Итоговый перечень должен включать позиции что и где для каждого элемента, как показано на рис. 9.10. Благодаря этому ничего не будет упущено.

Документ	Параграф	Описание
Процесс SOW	1.3.4	Разработка ПО компилятора
Процесс SOW	2.3.3	Командировки с целью обзора разработки проекта
Контракт	7.13.2.a	Следование стандарту ISO 5432f
Документы по специф.требований	3.4	Использование сжатия данных
Заказчик	6/5/99 Mtg.	Кодирование всего ПО на C++

Рисунок 9.10 - Перечень элементов, оказывающих влияние на разработку ПО

Поиск структуры WBS для произвольной системы высшего уровня

Определите, существует ли структура WBS для произвольной системы высшего уровня (проект либо программа высшего уровня), а также установите, приемлемо ли это ПО в данном случае. Для этого во многих организациях применяется структура WBS, имеющая стандартную архитектуру. Многие менеджеры системных проектов устанавливают структуру WBS, применимую к разбитому на более мелкие части ПО.

На рис. 9.11. показаны два возможных разбиения. При этом важно определение момента, при наступлении которого ПО будет сопоставимо с действиями высшего уровня, поскольку это может оказать влияние на то, каким образом структурируются и выполняются части программного проекта. Например, менеджер системного проекта может иметь специфический подход к порядку расположения элементов структуры WBS, к количеству уровней просмотра, к тому, каким образом следует соотносить определенные виды затрат и т.д.

Подход, проиллюстрированный в левой части рис. 9.11, представляет ПО, используемое в качестве внедренного элемента на базе аппаратного обеспечения в рамках системного проекта. В правой части рисунка демонстрируется ПО, выступающее в качестве отдельного, но эквивалентного элемента по отношению к аппаратному обеспечению. В этом случае наблюдается тенденция к изоляции планирования ПО от остальной части системы, в результате чего появляются несовместимые интерпретации требований.

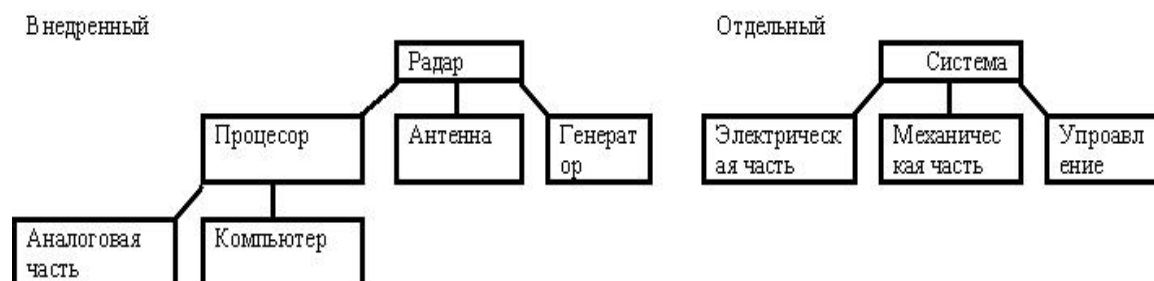


Рисунок 9.11 - Пример структуры WBS высшего уровня, демонстрирующего место размещения ПО

Определение программной архитектуры WBS

На этом этапе требуется определить логическую структуру (архитектуру) для программных частей структуры WBS. Как описывалось ранее и показывалось на рис. 9.9, существует множество различных программных архитектур WBS. В распоряжении многих организаций оказываются стандартные программные архитектуры WBS, благодаря чему можно убедиться в совместимости и лучше отслеживать затраты. Для различных программных продуктов и разных жизненных циклов разработки ПО требуются разные структуры WBS. На рис. 9.12 представлены возможные программные архитектуры, применяющие подходы А и В (рис. 9.9).

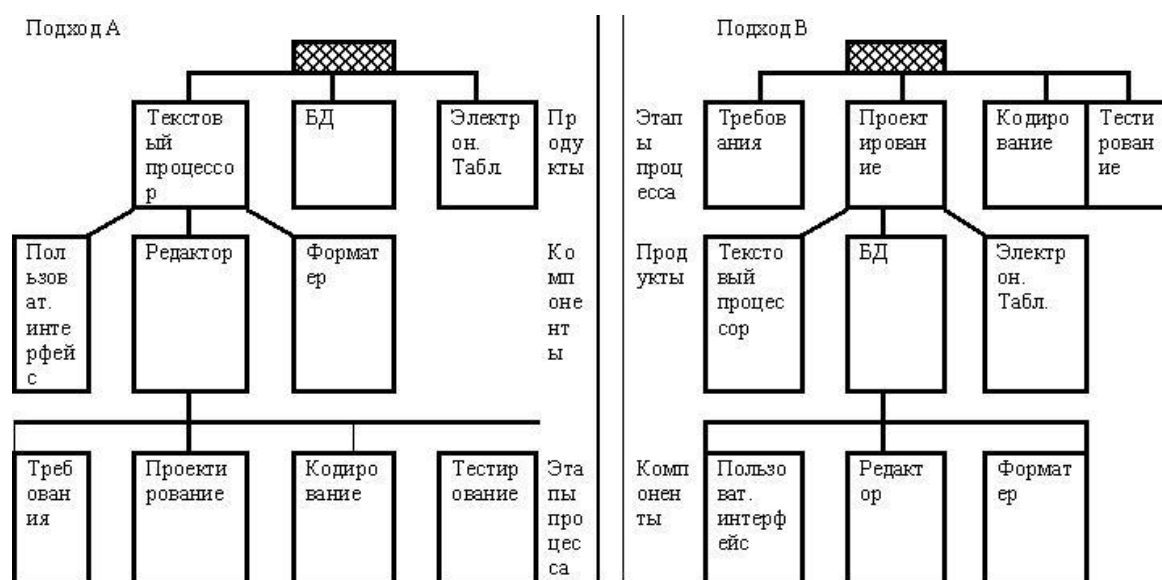


Рисунок 9.12 - Пример архитектур WBS для ПО, проиллюстрированного на рис. 9.9

Указание сведений для программной архитектуры WBS

Заполните выбранную структуру WBS действиями, в ходе внедрения которых выполняется работа, идентифицированная в доступной документации (SOW, или другие документы, указанные на этапе 1). Созданная на основе данных из рис. 9.10 и 9.12, матрица перекрестных ссылок на рис. 9.13 отображает стандартные элементы структуры WBS для ПО (в правой части рисунка). В левой части рисунка находятся соответствующие исходные документы, номера параграфов, а также описания, связанные с созданием элементов структуры WBS в правой части рисунка. При этом снижается вероятность пропуска каких-либо элементов в структуре WBS. Если один параграф SOW отображается в нескольких элементах структуры WBS (либо наоборот), можно создать несколько различных записей в матрице перекрестных ссылок.

В данном случае структура WBS может выступать в роли элемента отслеживания требований либо она может быть дополнена. Иногда стандартная структура WBS является более детализированной, чем исходные документы. В подобных случаях отслеживание всегда следует выполнять на наивысшем уровне во избежание появления чрезмерного количества деталей.

Информация об источнике	Требование	Элемент программной структуры WBS
Процесс SOW 1.1.1	Разработка С-компилятора	1.0 ПО С-компилятора
Процесс SPEC 2.0	Разработка компилятора	1.1 Создание С-компилятора
Процесс SPEC 2.1	Пользовательский интерфейс для ПК	1.1.1 Создание пользовательского интерфейса
PROCSTD3.4	Анализ требований	1.1.1.1 Анализ требований для польз. интерфейса
PROCSTD3.5	Проектирование	1.1.1.2 Проектирование польз. интерфейса
Процесс SPEC 2.2	Файловая система	1.1.2 Создание файловой системы
Процесс SPEC 3.0	Тестирование в среде компании	1.2 Создание тестовой оболочки
Процесс SOW 2.3.4	Поддержка руководства пользователя	1.3 Документация

Рисунок 9.13 - Пример матрицы перекрестных ссылок для ПО из рис. 9.9

Определение категорий затрат для ПО

Определите категорию оценки затрат для каждого элемента в структуре WBS. Этот последний этап не является необходимым для каждого проекта, однако он является важным для тех проектов, где отслеживаются затраты на уровне единиц в структуре WBS. Если этот этап не будет выполнен в данный момент времени, его придется выполнить позднее, во время осуществления процесса оценки затрат. В некоторых проектах с целью упрощения применяется лишь одна категория затрат.

Обычно при оценке учитываются только время, затрачиваемое на разработку. В более сложных проектах требуется большее количество категорий в силу того, что возникает потребность в различных единицах. Категория оценки затрат определяет, каким образом будут оцениваться затраты для каждого элемента (рис. 9.14).

Например, капитальное оборудование, обычно оценивается в деньгах, в то время, как трудозатраты оцениваются в часах, неделях либо месяцах, в течение которых выполняется работа (эти результаты могут быть легко преобразованы таким образом, что будут выражаться в денежном эквиваленте). Элементы накладных расходов, например, при выполнении управления проектами, контроля изменений, а также менеджмента конфигурации могут оцениваться с помощью пропорциональных добавлений в базовые оценки трудозатрат на разработку ПО (процентные соотношения).

Документ	Параграф	Номер пункта WBS	Описание	Категория
Процесс SOW	1.3.4	1.1.2.2	Разработка ПО компилятора	ПО
Процесс SOW	2.3.3	1.7.1	Командировки с целью обзора разработок проектов	Фактические денежные затраты!

Рисунок 9.14 - Пример категорий затрат

Практика

Пример 1

Сетевой график - необходимый элемент сложного производства, состоящего из нескольких связанных и зависящих друг от друга этапов. Выявление критического пути и временных резервов производства - основная задача, решаемая построением сетевого графика. Такие задачи могут быть представлены в виде графа и в виде отображающей его таблицы. Для нахождения критического пути (последовательности этапов работы, определяющих длительность всего проекта и не имеющих резерва по времени) применяются вычислительные методы. Одним из таких методов является табличный метод и применяется для данных, представленных в виде таблицы.

Дано: сетевой граф, рис.9.15.

Вершины графа - этапы работ.

Рёбра графа - выполнение работы. Рёбра имеют длину, обозначающую продолжительность работы и направление, обозначающее последовательность выполнения работы.

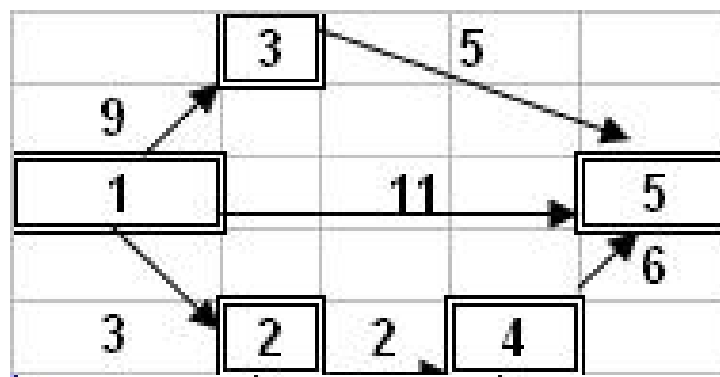


Рисунок 9.15 - Сетевой график А

Требуется найти:

1. Минимальное количество человек для выполнения проекта с работами 1- 5 сетевого графика А.

2. Критический путь и работы критического пути.
3. Сформировать календарный график выполнения.
4. Произвести оптимизацию графика.

На сетевом графике **A** все операции могут производиться с постоянной интенсивностью потребления ресурсов.

Число ресурсов для каждой операции фиксировано.

На основании данных графа и количества человек для каждой операции заполним результирующую таблицу.

В таблице на рис.9.16 представлены исходные данные решения задачи, где

i - номер начального события;

j - номер конечного события;

$t(i, j)$ - продолжительность операции;

$n(i, j)$ - требуемое число людских ресурсов;

$D(i, j)$ - трудоемкость операции в чел/дн., вычисляется как $t(i, j) \times n(i, j)$.

нач. событ.	конечн ое событи е	длительн ость	трудоем кость $t \times n$	количество человек
i	j	$t(i, j)$	$D(i, j)$	$n(i, j)$
1	2	3	3	1
1	3	9	27	3
1	5	11	33	3
2	4	2	2	1
3	5	5	20	4
4	5	6	18	3

Рисунок 9.16 - Исходные данные

1. Работы 1-2, 1-3, 1-5 выполняются параллельно.

Работы 2-4, 3-5 выполняются тоже параллельно, а работа 4-5- является последовательной.

Однако т.к. длительность работы 2-4 составляет 2 дня, то мы формируем 2 блока параллельных работ.

Это 1-й блок работ 1-2, 1-3, 1-5 3-5 , и 2-й блок работ 2-4, 3-5, 4-5.

Для 1-го блока максимальное количество людей =3, а для 2-го - 4 человека.

Т.е. на выполнение операций, представленных в сетевом графике А на рис.9.15 выделено всего 7 человек.

2. Согласно рис.9.15 найдем длительность для каждого из возможных вариантов прохождения графа от первой до последней вершины

Длительность работ 1-3-5 составляет 14 дней,

Длительность работ 1-5 составляет 11 дней,

Длительность работ 1-2-4-5 составляет 11 дней.

Критический путь составляет 14 дней , а работы 1-3-5 являются критическими работами.

3. После заполнения результирующей таблицы и расчета критического пути сформируем календарный график выполнения.

вариант 1 неоптимизированный																		
i	j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	2	1	1	1														
1	3	3	3	3	3	3	3	3	3	3								
1	5	3	3	3	3	3	3	3	3	3	3	3						
2	4				1	1												
3	5										4	4	4	4	4			
4	5												3	3	3	3	3	3
		7	7	7	7	7	6	6	6	6	7	7	7	7	7	3	3	3

Рисунок 9.17 - Распределение ресурсов (вариант 1)

На третьем шаге распределения ресурсов предпочтение отдано операции (3-5), которая в

сетевом графике рис.9.15 находится на критическом пути, а не операции (4-5), имеющей резерв времени.

Рассмотрим другой возможный вариант распределения этих же ресурсов, отдав предпочтение на третьем шаге операции (4-5), имеющей резерв времени, а не операции (3-5), лежащей на критическом пути. Результаты этого случая представлены на рис.9.18

вариант 2 оптимизированный													дни					
i	j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	2	1	1	1														
1	3	3	3	3	3	3	3	3	3	3								
1	5	3	3	3	3	3	3	3	3	3	3	3						
2	4				1	1												
3	5												4	4	4	4	4	
4	5										3	3	3	3	3	3		
		7	7	7	7	7	6	6	6	6	6	6	7	7	7	7	4	0

Рисунок 9.18 - Распределение ресурсов (вариант 2)

Рисунок 9.19 демонстрирует оптимизацию в длительности всего проекта в 15 дней, но при условии увеличения количества людей с 6-го по 11-й день.

вариант 3 оптимизированный при условии увеличения ресурсов													дни									
i	j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17				
1	2	1	1	1															3			
1	3	3	3	3	3	3	3	3	3	3									27			
1	5	3	3	3	3	3	3	3	3	3	3	3							33			
2	4				1	1													2			
3	5										4	4	4	4	4				20			
4	5						3	3	3	3	3	3	3	3	3	3			30			
		7	7	7	7	7	9	9	9	9	10	10	7	7	7	3	0	0				

Рисунок 9.19 - Распределение ресурсов (вариант 3)

Пример 2

Дано: сетевой граф, рис.9.20.

Вершины графа - этапы работ.

Рёбра графа - выполнение работы. Рёбра имеют длину, обозначающую продолжительность работы и направление, обозначающее последовательность выполнения работы.

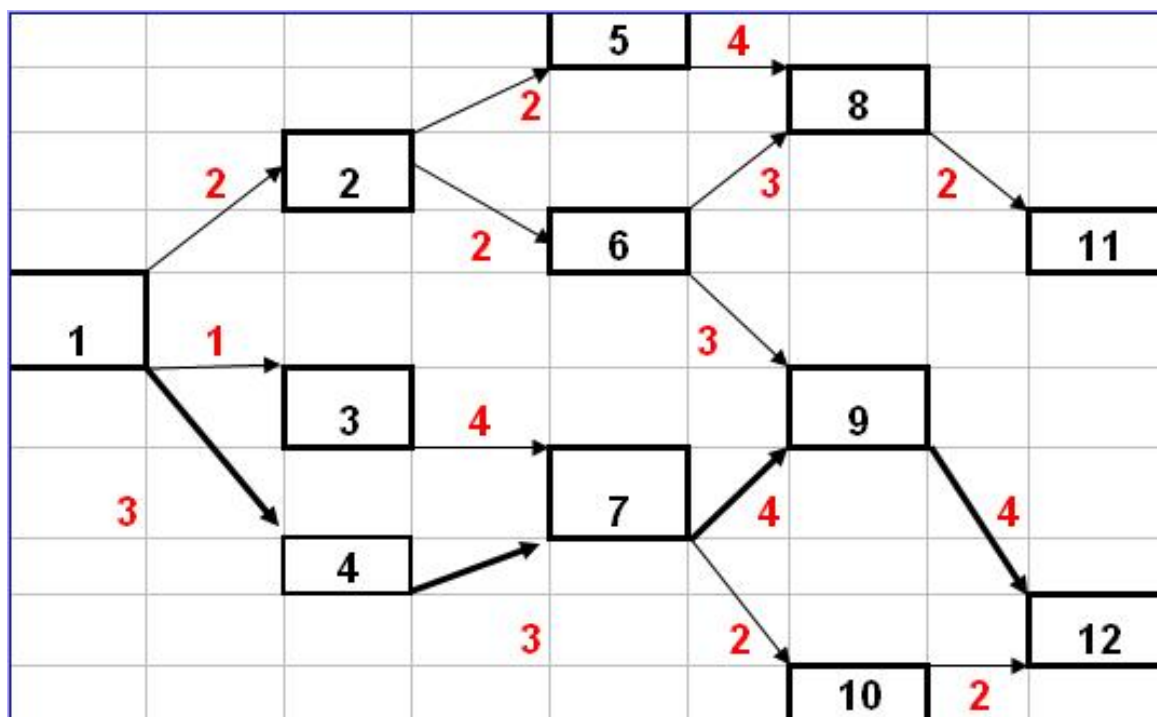


Рисунок 9.20 - Сетевой график В

Требуется найти:

1. Минимальное количество человек для выполнения проекта с работами 1- 5 сетевого графика А.
2. Критический путь и работы критического пути.
3. Сформировать календарный график выполнения.
4. Произвести оптимизацию графика.

На сетевом графике **В** все операции могут производиться с постоянной интенсивностью потребления ресурсов.

Число ресурсов для каждой операции фиксировано.

На основании данных графа и количества человек для каждой операции заполним результирующую таблицу.

В таблице на рис.9.21 представлены исходные данные решения задачи, где

i - номер начального события;

j - номер конечного события;

$t(i, j)$ - продолжительность операции;

$n(i, j)$ - требуемое число людских ресурсов;

$D(i, j)$ - трудоемкость операции в чел/дн., вычисляется как $t(i, j) \times n(i, j)$.

i	j	t(i, j)	n(i, j)	d(i, j)
1	2	2	1	2
1	3	1	1	1
1	4	3	1	3
2	5	2	1	2
2	6	2	2	4
3	7	4	2	8
4	7	3	1	3
5	8	4	1	4
6	8	3	2	6
6	9	3	1	3
7	9	4	2	8
7	10	2	2	4
8	11	2	1	2
9	12	4	2	8
10	12	2	2	4

Рисунок 9.21 - Исходные данные

1. Блоки параллельности выделены жирными линиями и цветом, а количество людей выделено синим цветом на рис.9.21.

На выполнение операций, представленных в сетевом графике В выделено всего 7 человек.

2. Согласно рис.9.20 найдем длительность для каждого из возможных вариантов прохождения графа от первой до последней вершины.

11 (1,2,6,9,12)
9 (1,3,7 10,12)
10 (1,2,5,8,11)
13 (1,3,7,9,12)
10(1,4,7,10,12)
9(1,2,6,8,11)
14 (1,4,7,9,12)

Рисунок 9.22 - Длительность для каждого из возможных вариантов прохождения графа от первой до последней вершины

Критический путь составляет 14 дней , а работы 1-4-7-9-12 являются критическими работами.

3. После заполнения результирующей таблицы и расчета критического пути сформируем календарный график выполнения.

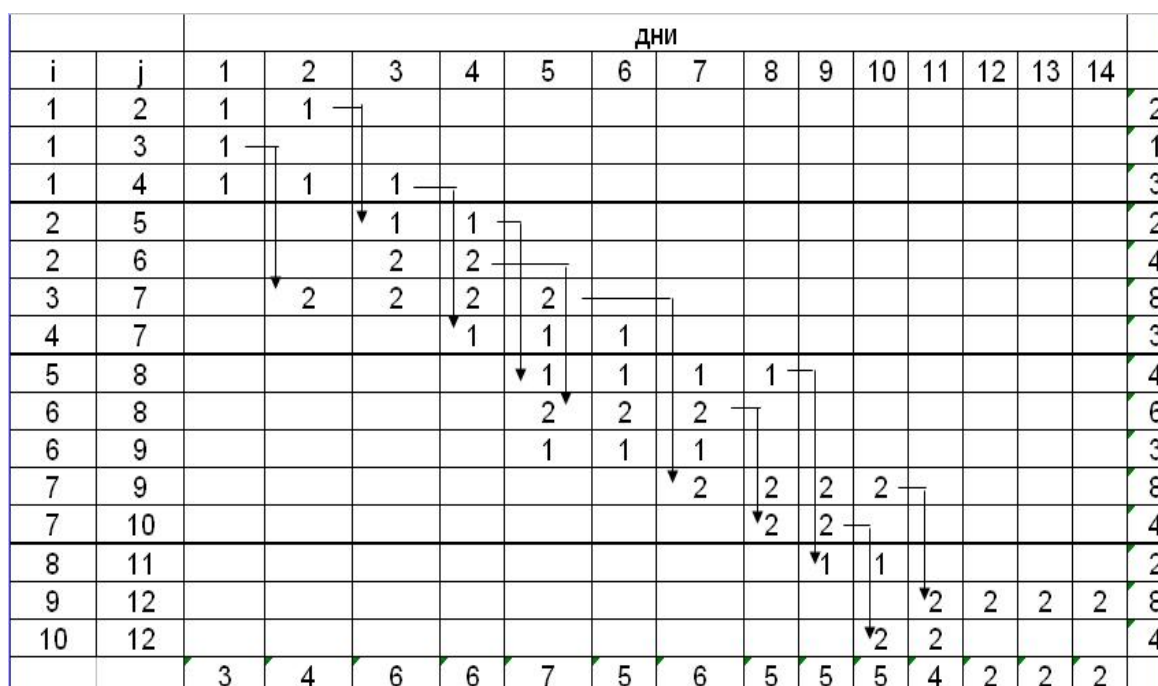


Рисунок 9.23 - Распределение ресурсов

Резерв времени = 0, оптимизация графика без привлечения дополнительного количества людей не имеет смысла.

Вопросы для самопроверки

1. Что такое критический путь?
2. Что такое критические работы?
3. Для чего используется диаграмма Ганта.
4. Что такое буфер или резерв времени?
5. Что такое критический путь?
6. Что такое критические работы?
7. Для чего используется диаграмма Ганта?
8. Что такое буфер или резерв времени?
9. Каково основное назначение WBS.

10. Что такое WBS?
11. Содержание пакета работ.
12. Нижний уровень WBS.
13. Какой процесс не относится к управлению временем?
14. Какая зависимость свидетельствует о параллелизме?
15. Что представляет собой нижний уровень структуры пооперационного перечня работ для сетевой диаграммы календаря?

Решить задачу.

Дано:

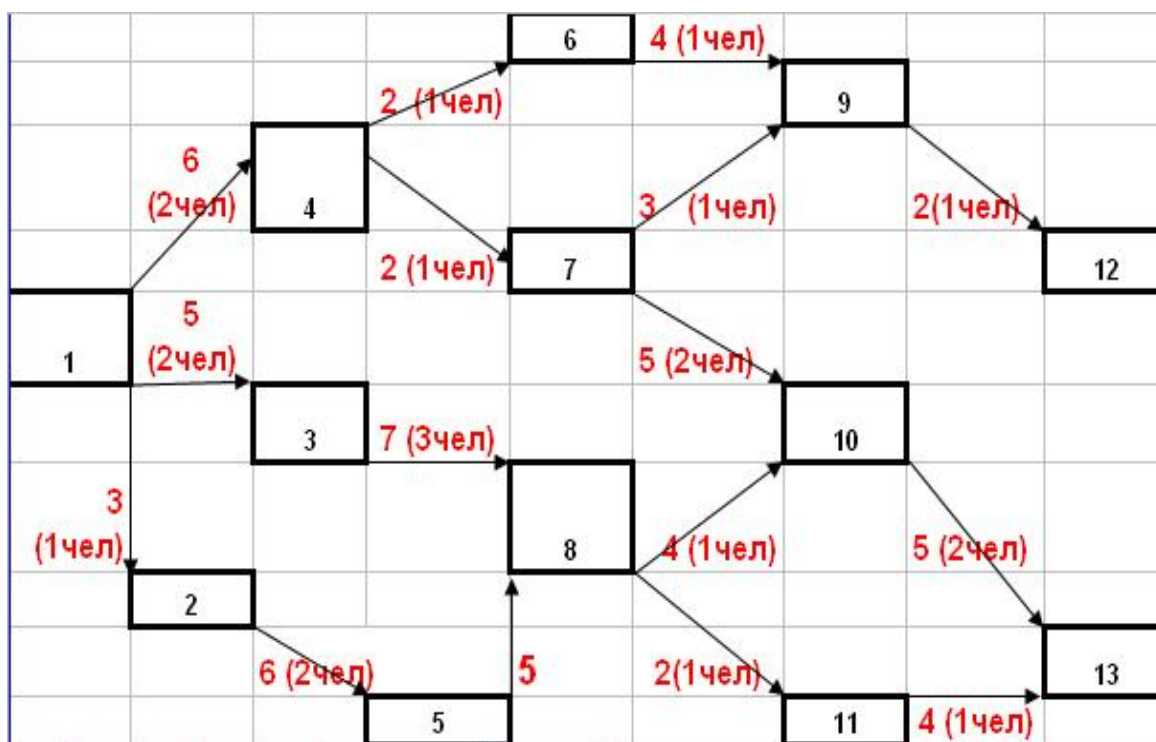


Рисунок 9.24 - Сетевой граф

Определить:

1. Трудоемкость работ.

2. Длительность всех вех.
3. Минимальное количество трудовых ресурсов
4. Длительность и работы критического пути
5. Составить график распределения трудовых ресурсов.
6. Сделать вывод об эффективности.

Вывод к разделу 9 - Метод критической цепи: эффективное управление проектами с использованием буферов времени и ресурсов

В разделе рассматривался метод критической цепи для управления временем и оптимизации ресурсов.

Перечень ссылок

Источники, использованные в материалах

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

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

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

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

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

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