

**ПРАКТИЧЕСКОЕ
ЗАНЯТИЕ №1**

Тема:

**«Структурное
планирование»**

Основные определения

Сетевое планирование и управление (СПУ) – это графическое представление комплекса работ, ориентированное на достижение конечной цели.

Сетевые графики позволяют *наглядно представить весь комплекс работ и их характеристики* во время планирования и исполнения какого-либо проекта, и поэтому их иногда называют сетевыми моделями.

Организация работ на основе метода СПУ включают основные стадии:

- ✓ **формирование полного перечня работ, отражающего их взаимосвязи и технологическую последовательность, которые предстоит выполнить для достижения поставленной конечной цели;**
- ✓ **построение структурного плана – графика и нумерация событий.**
- ✓ **определение ожидаемого времени выполнения работ и расчет параметров сетевого графика;**
- ✓ **оптимизация сетевого графика, т.е. выбор наилучшего варианта организации работ (в настоящее время наиболее полно разработаны и чаще всего применяются способы оптимизации сетевых графиков по времени, т.е. улучшение графика осуществляется с целью, сокращения сроков выполнения намеченного комплекса работ);**

- ✓ **разработка сетевого графика**, отражающего весь комплекс работ в их взаимосвязи и определенной технологической последовательности, которые предшествуют завершающему событию (т.е. окончанию всего объема работ);
- ✓ **календаризация графика;**
- ✓ **оперативное управление и контроль за ходом выполнения работ**, предусмотренных сетевым графиком, — это стадия непосредственного руководства работами на основе информации о состоянии дел на каждом отдельном участке в тот или иной период времени.

Основные понятия и элементы сетевого графика

Сетевые графики основаны на использовании *двух элементов* графического изображения:

– *безразмерных стрелок*, обозначающих **работы** (процессы)

—————→ **действительные работы**

-----→ **фиктивные работы;**

– *кружков* (или других геометрических фигур), обозначающих результаты выполнения одной или нескольких работ. Эти результаты называются **событиями** ○.

Под событием подразумевают свершение какого-либо факта, который соответствует окончанию и одновременно возможности начала одной или нескольких работ.

Каждое событие сети получает свой номер. Нумерация событий начинается с определения *исходного события* сети, ему обычно присваивается номер **1** или **0**.

Затем определяется *завершающее событие* сети, которому присваивается номер, равный количеству событий в сети.

Начальным и конечным событием каждой отдельной работы называются соответственно такие события, которые данная работа связывает.

Работу в сетевом графике принято обозначать двумя цифрами, первая из которых показывает номер начального события, а вторая — номер конечного события. **Например**, 4-е и 6-е события связывает **работа (4, 6)**.

Путем в сетевом графике, называется какая-либо **непрерывная последовательность работ (стрелок)**, от исходного до завершающего события графика.

Длина (продолжительность) пути равна сумме продолжительностей составляющих его работ. Продолжительность пути выражается в единицах времени.

Путь, имеющий наибольшую продолжительность, — **критический путь**.

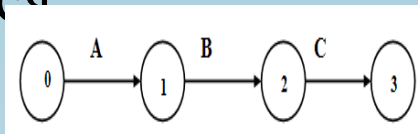
Работы и события, лежащие на критическом пути, называются соответственно **критическими работами и критическими событиями**.

Критический путь сети **на графике изображается жирной линией**.

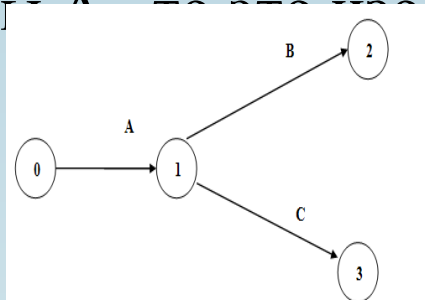
Критический путь служит основой для оптимизации программы работ, т.к. для минимизации срока выполнения всего комплекса работ достаточно сократить только сроки выполнения критических работ.

При построении структурного плана сетевого графика необходимо соблюдать следующие **правила**:

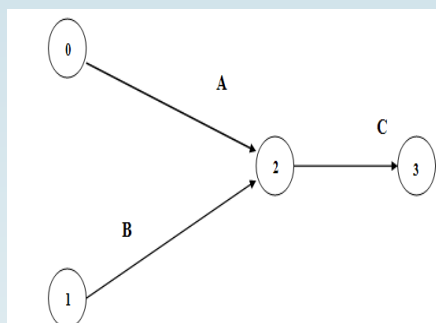
Правило 1. Если работы А, В и С выполняются последовательно, то на графике они изображаются



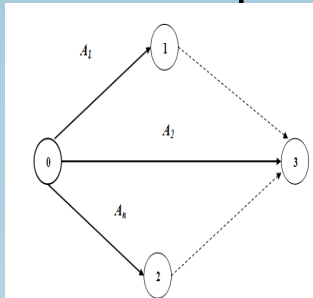
Правило 2. Если для работ В и С, выполняемых параллельно, необходим результат работы А, то это изображается



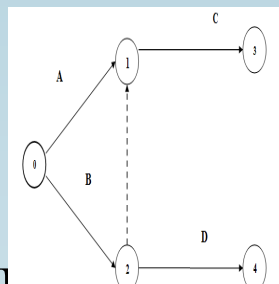
Правило 3. Если для выполнения работы С необходим результат работ А и В, то это изображается



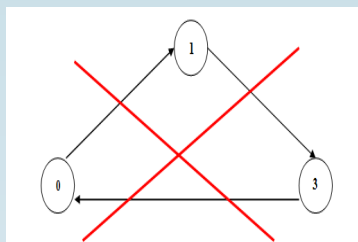
Правило 4. Если n работ A_1, A_2, \dots, A_n начинаются и заканчиваются одним и тем же событием, то для того, чтобы эти работы имели разные коды, вводится $n-1$ фиктивная работа



Правило 5. Если работа С следует за двумя параллельными работами А и В, а работа Д следует только за работой В, то такая ситуация изображается



Правило 6. Не должно быть замкнутых контуров



Правило 7. Каждое событие, за исключением исходного и завершающего, должно иметь не менее одной предшествующей и одной последующей работы.

Показатели для расчета сетевого графика

Ранний срок свершения события – t^p_i – определяет календарную дату наиболее раннего из возможных сроков выполнения предшествующих ему работ и определяется величиной наиболее длительного пути, от исходного до рассматриваемого события.

Расчет ранних сроков осуществления событий выполняют последовательно, начиная от исходного события, для которого ранний срок осуществления принимается равным – нулю.

Ранний срок наступления события, которое имеет *одну предыдущую работу*, равняется сумме раннего срока осуществления начального события этой работы и ее длительности. Для определения раннего срока осуществления события, которое имеет *несколько предыдущих работ*, необходимо к ранним срокам осуществления начальных событий этих работ прибавить их длительность и из полученных результатов выбрать *максимальный*

Поздний срок свершения события – t^n_i – определяет календарную дату наиболее позднего из допустимых сроков выполнения предшествующих ему работ. Если установлен срок свершения завершающего события, являющегося результатом всего комплекса проводимых работ, то каждое промежуточное событие должно наступить не позже этого срока. Он является предельно допустимым сроком свершения события.

Расчет поздних сроков осуществления событий выполняется последовательно, начиная с завершающего события, для которого поздний срок осуществления принимают равным – раннему сроку его осуществления. Поздний срок осуществления события, которое имеет одну последующую работу, равняется разнице между поздним сроком осуществления конечного события этой работы и её длительностью. Если же событие имеет *несколько последующих работ*, то необходимо выполнить отмеченную выше операцию для каждой из этих работ и из полученных результатов выбрать **минимальный**

Ранний срок начала работы (или раннее начало)

– самый ранний из всех возможных сроков начала работы. Ранний срок начала работы, равняется раннему сроку осуществления начального события этой работы.

Ранний срок окончания работы (или раннее

окончание) – самый ранний из возможных сроков окончания работы. Ранний срок окончания работы равняется сумме раннего срока осуществления начального события этой работы и её длительности.

Поздний срок начала работы (или позднее

начало работы) – самый поздний срок начала работы, при котором общий срок окончания всех работ не изменяется. Поздний срок начала работы равняется разнице между поздним сроком осуществления конечного события этой работы и её длительностью.

Поздний срок окончания работы (или позднее

окончание работы) – самый поздний срок окончания работы равняется позднему сроку осуществления конечного события этой работы.

Зная самый поздний и самый ранний сроки свершения событий, определяют *резерв времени наступления событий* – R_i

Полный (общий) резерв времени (R_{i-j}) – максимальное время, на которое можно увеличить длительность данной работы или задержать начало её выполнения, не изменяя при этом срока выполнения всей разработки.

Полный резерв времени работы равняется разнице позднего и раннего сроков начала работы, или разнице позднего и раннего сроков окончания работы.

Критический путь – это наиболее длительный путь, который проходит от начального к завершающему событию. Он определяет длительность выполнения всего комплекса разработки и равняется сумме продолжительностей всех расположенных на нем работ. Резервы времени работ и событий, которые находятся на критическом пути, равняются **нулю**.

Пример 1.

Последовательность и код работ проекта представлены в таблице 1.1. Найти критический путь и построить сетевой график.

Таблица 1.1

Исходные данные

Код работ <i>i-j</i>	Длительность работ <i>t_{i-j}</i>
0-1	2
1-2	1
1-3	1
1-10	2
2-4	2
3-5	1
3-6	1
3-9	1
4-10	1
5-7	3
6-9	2
7-8	3
8-9	1
9-10	1
10-11	1
11-12	3

Пример 2.

Последовательность и код работ проекта представлены в таблице 1.2. Найти **критический путь** и **построить сетевой график**.

Таблица 1.2

Исходные данные

Код работы	Длительность работ
0-1	4
0-2	1
1-2	1
1-3	2
1-4	3
2-5	3
2-6	2
3-7	2
4-6	1
4-7	3
5-7	1
6-7	1

Пример 3.

Последовательность и код работ проекта представлены в таблице 1.3. Найти **критический путь** и **построить сетевой график**.

Таблица 1.3

Исходные данные

Код работы	Длительность работ
0-1	2
0-2	3
0-3	2
2- 3	3
2-4	2
3-4	0
3- 5	3
3- 6	2
4- 5	7
4- 6	5
5- 6	6