

A stylized illustration of a workspace on a dark blue background. It includes a laptop with a teal screen and orange keyboard, a stack of books, a pen holder with three pens, a potted plant, and a framed map or diagram with orange lines and a teal grid pattern.

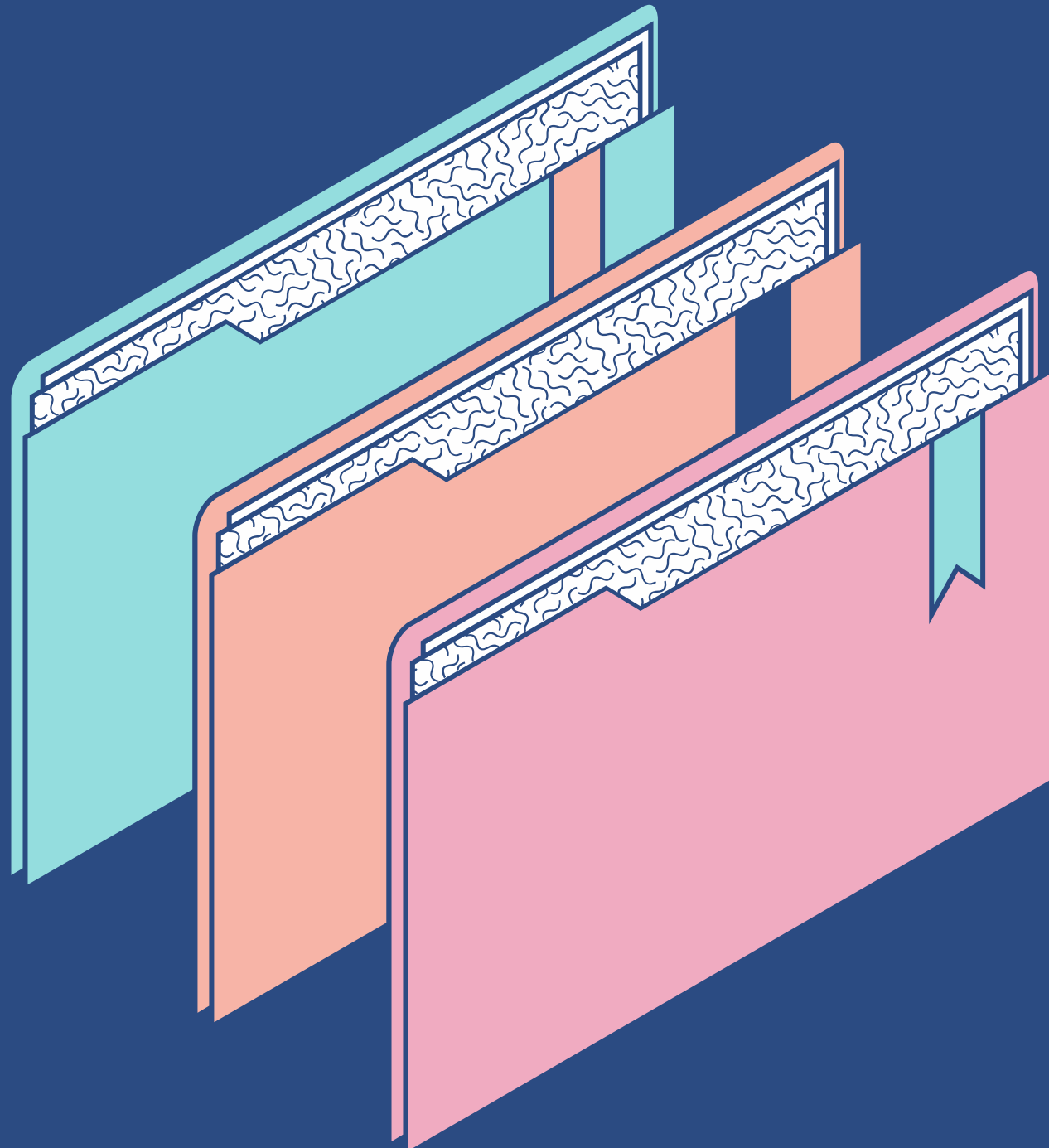
УПРАВЛЕНИЕ IT-ПРОЕКТАМИ  
ЛЕКЦИЯ 3

# Руководство программным проектом

Изотова Анастасия  
Андреевна

# Руководство программным проектом

Руководство программным проектом является защитной деятельностью программной инженерии, захватывающей все виды основной деятельности - подготовку, планирование, моделирование, конструирование и развертывание.



Руководство применяется ко всем четырем "П" разработки: персоналу (тем, кто это делает), процессу (порядку, в котором это делается), проекту (среде, в которой это делается), продукту (итогу всех дел)

Цель любого программного проекта состоит в производстве определенного программного продукта.

# Характерные точки управления проектом.

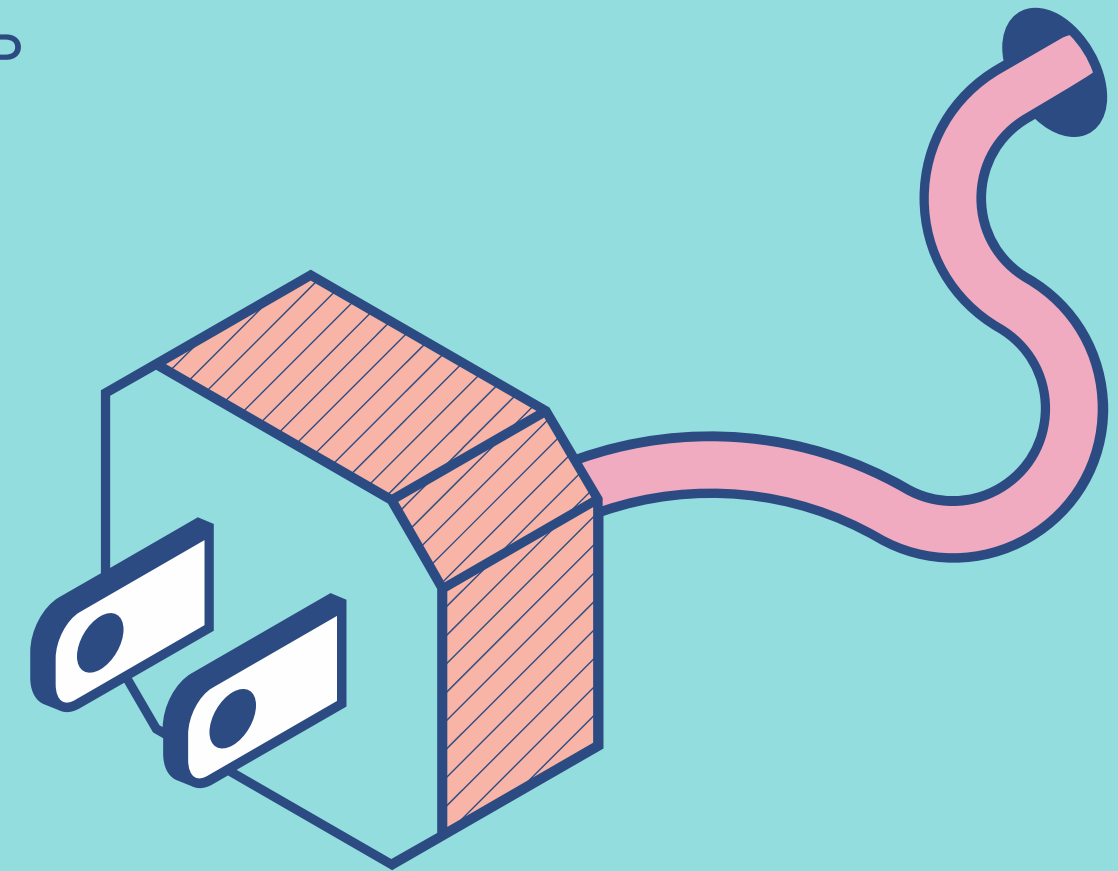
## 1. Начало проекта.

Перед планирование проекта следует:

- установить цели и проблемную (предметную) область проекта);
  - обсудить альтернативные решения;
- выявить технические и управленческие ограничения.

Без этой информации нельзя обоснованно оценить стоимость, реально распределить задачи проекта, составить такой план руководства, который обеспечивает всестороннее отображение состояния дел.

Цели и область проекта определяются разработчиком и заказчиком. Цели идентифицируют задачи проекта (без обсуждения способов их решения). Проблемная область задает основные функции ПО, их количественные границы, а также иные характеристики.





## Характерные точки управления проектом.

### 2. Измерения, меры и метрики.

- Измерения помогают понять как процесс разработки продукта, так и сам продукт. Измерения процесса производятся в целях его улучшения, измерения продукта - для повышения его качества.
- В результате измерения определяется мера - количественная характеристика какого-либо свойства объекта. Путем непосредственных измерений могут определяться только опорные свойства объекта. Все остальные свойства оцениваются в результате вычисления тех или иных функций от значений опорных характеристик. Вычисления этих функций проводятся по формулам, которые дают числовые значения и называются метриками. В программной инженерии понятия мера и метрика очень часто рассматривают как синонимы.

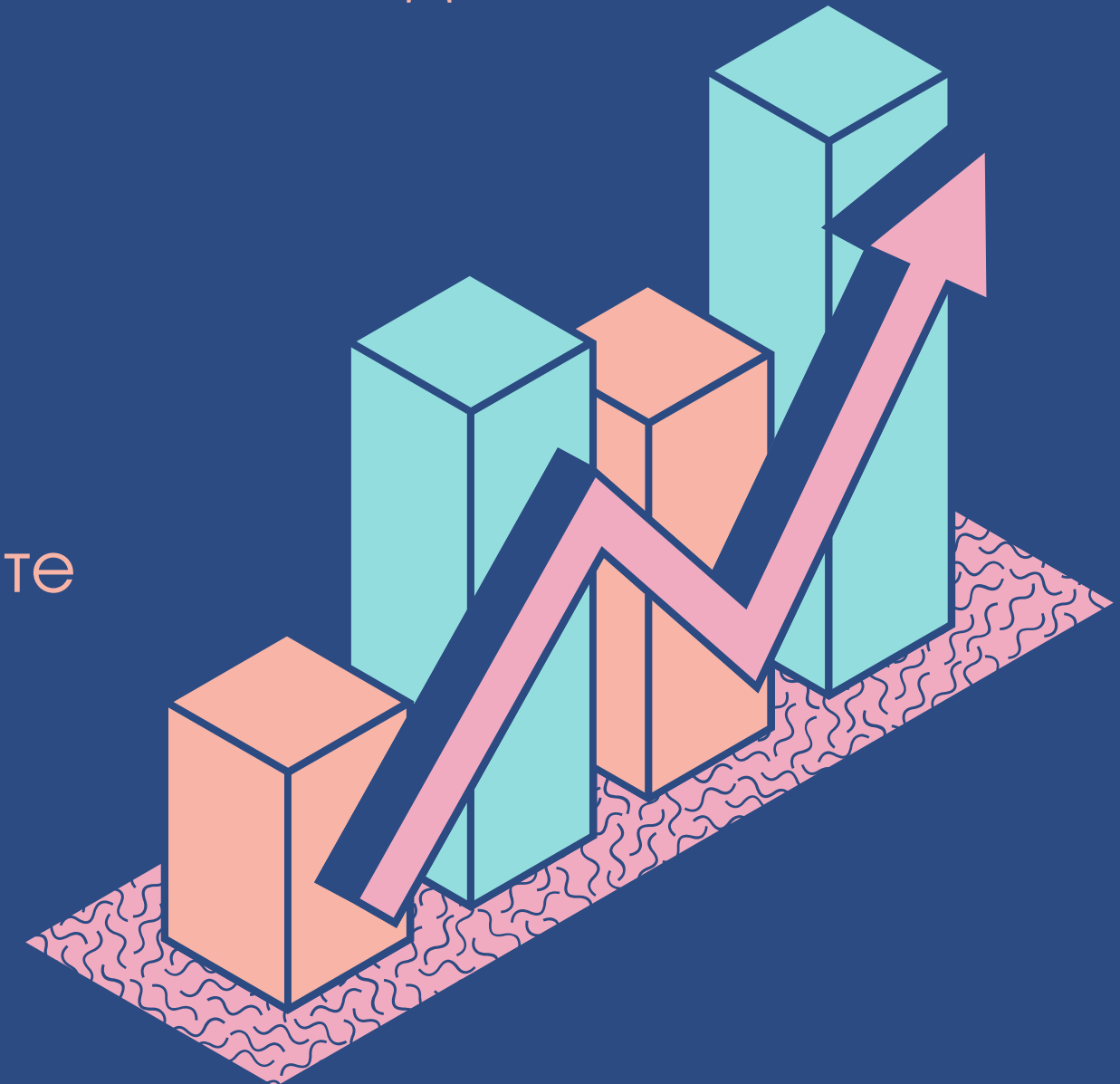
## Характерные точки управления проектом.

### 3. Процесс оценки.

При планировании программного проекта надо оценить людские ресурсы (в человеко-месяцах), продолжительность (в календарных днях), стоимость (в долларах). Обычно исходят из прошлого опыта. Если новый проект по размеру и функциям похож на предыдущий, вполне вероятно, что потребуются такие же ресурсы, время и деньги. Если же аналога нет или прошлого опыта недостаточно, нужно изучать способы оценки.

### 4. Анализ риска.

На этой стадии исследуется область неопределенности, имеющаяся в наличии перед созданием программного продукта. Анализируется ее влияние на проект. В результате принимается решение: выполнять проект или нет. Специальный аппарат анализа позволяет атаковать риск прежде, чем он атакует программный продукт.





## Характерные точки управления проектом.

### 5. Планирование.

В каждом программном проекте существует планирование, но не все планы одинаковы.

Суть планирования заключается в:

- определении набора задач проекта;
- установлении связи между задачами, оценивании сложности каждой задачи;
- определении людских и других ресурсов;
- создании сетевого графика задач, проведении его временной разметки.

### 6. Трассировка и контроль.

Каждая задача, помеченная в плане, отслеживается руководителем проекта. При отставании в решении задачи применяются утилиты повторного планирования. С помощью утилит определяется влияние этого отставания на промежуточную веху и общее время разработки. Под вехой понимается временная метка, к которой привязано подведение промежуточных итогов.

В результате повторного планирования:

- могут быть перераспределены ресурсы;
- могут быть реорганизованы задачи;
- могут быть пересмотрены выходные обязательства.

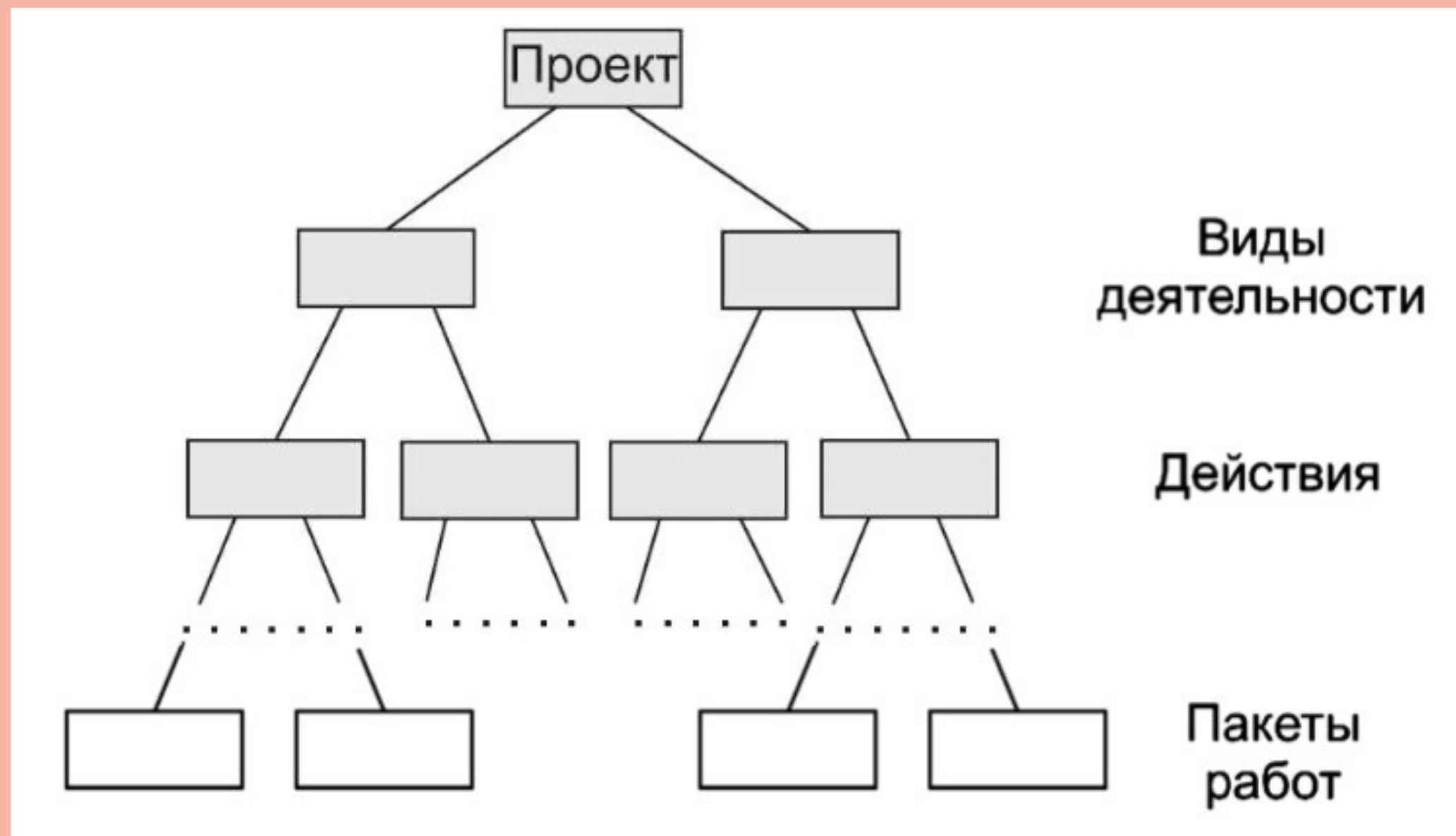


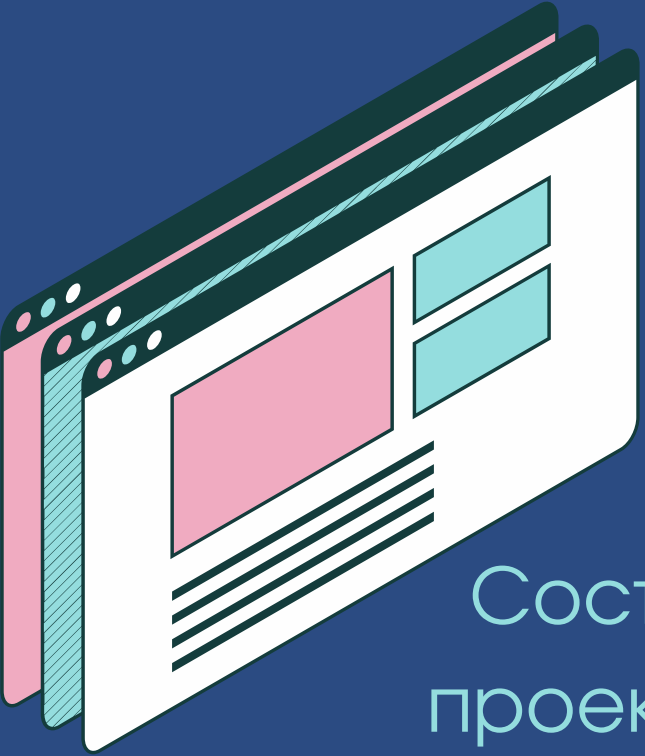
**Последовательность действий  
при планировании**

# Иерархическая структура работ

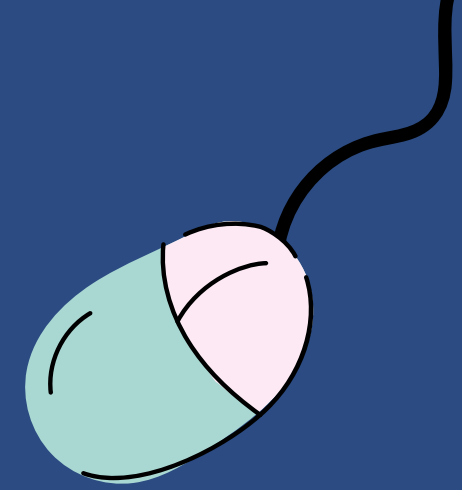
- это ориентированная на результаты иерархическая декомпозиция работ, которые должна выполнять команда проекта для достижения цели, то есть создания требуемых результатов.

Декомпозиция заключается в разделении работ на более мелкие элементы. Она выполняется до тех пор, пока работы не будут соответствовать уровню пакетов работ.





# Структура графика работ программного продукта



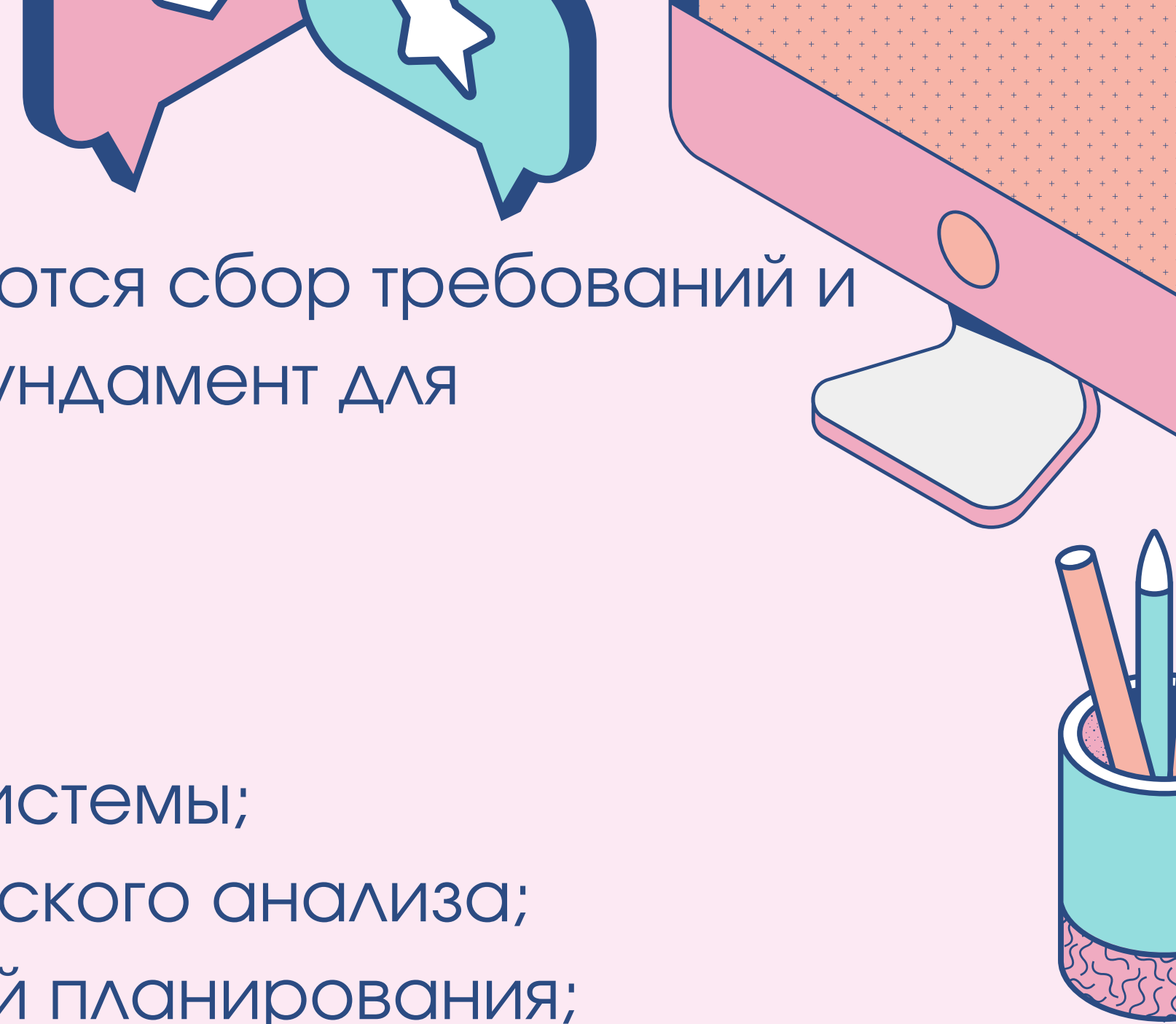
Составление графика - одна из самых ответственных работ менеджера проекта. Менеджер оценивает длительность проекта, определяет ресурсы, необходимые для реализации рабочих задач и разворачивает последовательность задач по времени.

Планирующие утилиты позволяют:

- определить критический путь (цепочку задач, задающих длительность всего проекта);
- определить длительность критического пути;
- установить для каждой задачи наиболее вероятную временную оценку;
- вычислить границы, определяющие временное окно для отдельной задачи.



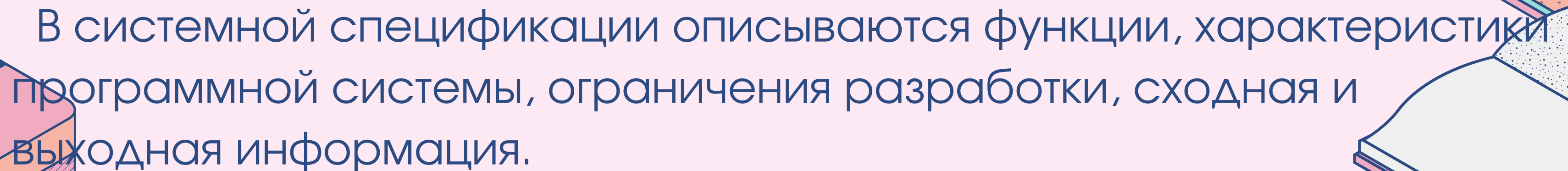




Первыми выполняемыми задачами являются сбор требований и анализ требований. Они закладывают фундамент для последующих параллельных задач.

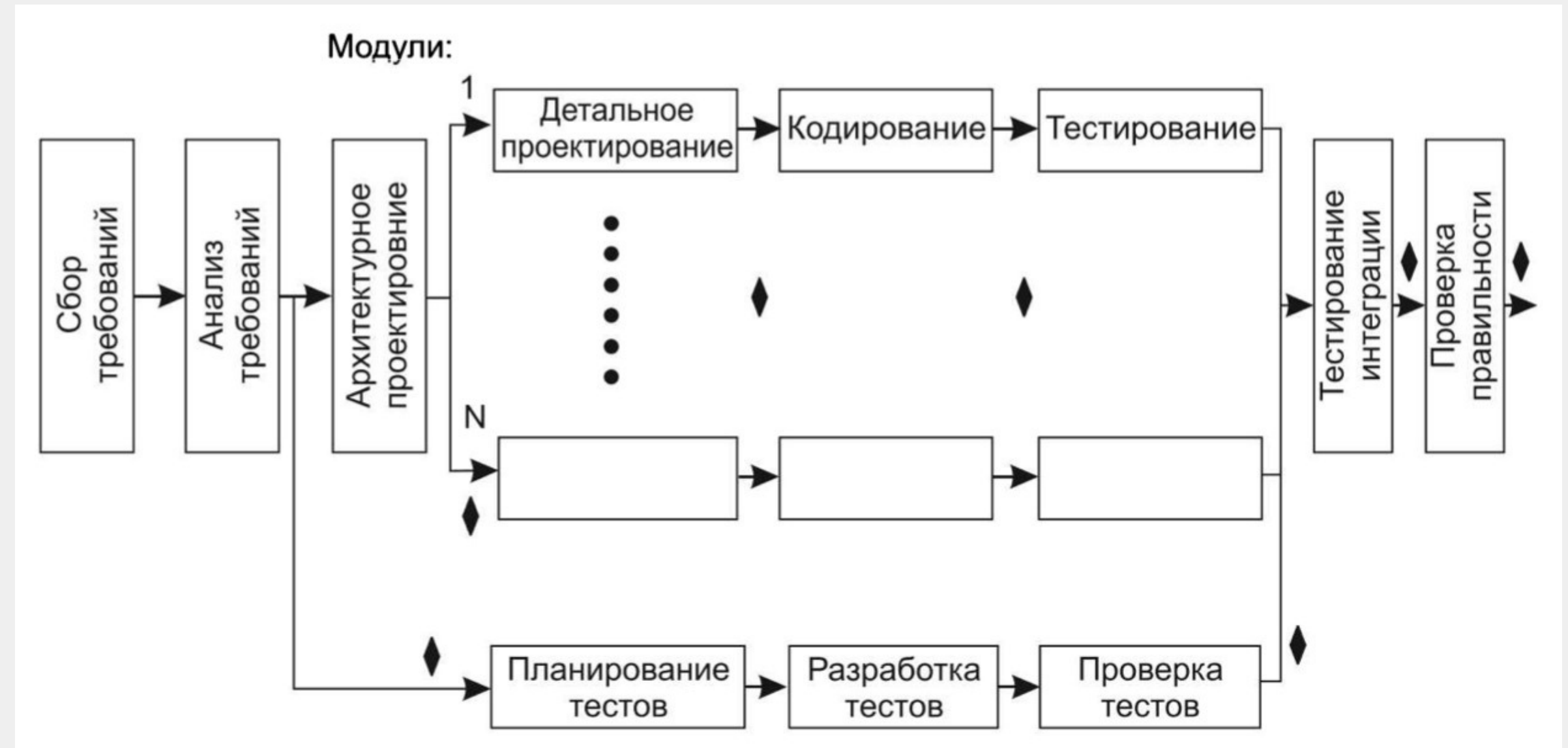
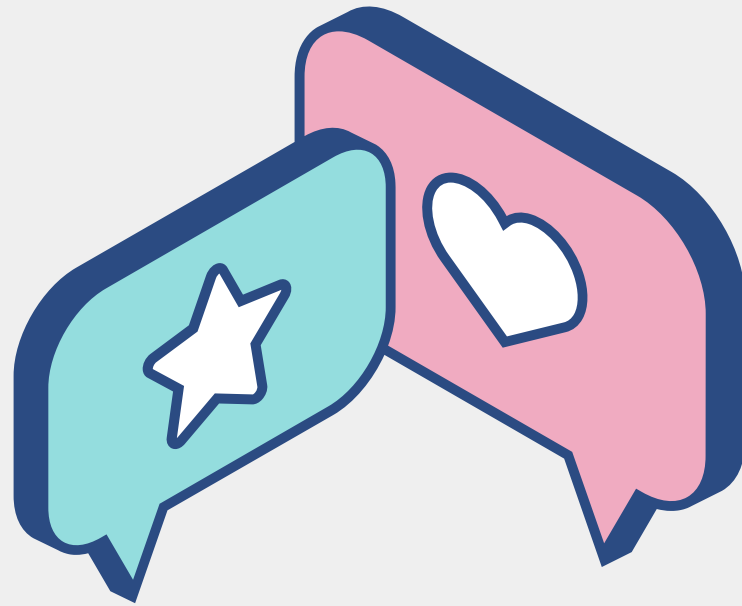
Сбор требований проводится с целью:

- 1) выяснения потребностей заказчика;
- 2) оценки выполнимости программной системы;
- 3) выполнения экономического и технического анализа;
- 4) определения стоимости и ограничений планирования;
- 5) создания системной спецификации.



В системной спецификации описываются функции, характеристики программной системы, ограничения разработки, сходная и выходная информация.

# Типовая сетевая диаграмма работ проекта



## Анализ требований дает возможность:

- уточнить функции и характеристики программного продукта;
- обозначить интерфейс продукта с другими системными элементами;
- определить проектные ограничения программного продукта;
- построить модели: процесса, данных, режимов функционирования продуктов;
- создать такие формы представления информации и функций системы, которые можно использовать в ходе проектирования.

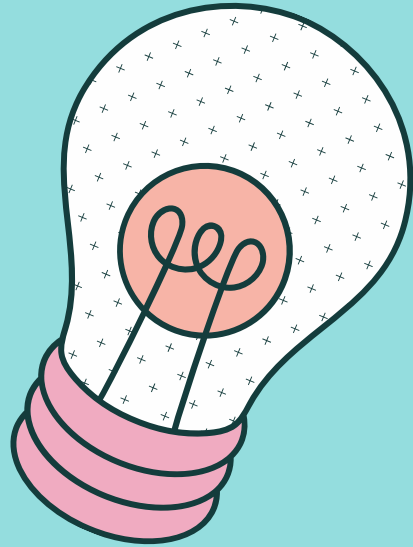
Результаты анализа сводятся в спецификацию анализа, содержащую конкретизированные требования к программному продукту.



# Рекомендуемое правило распределения затрат

Рекомендуемое правило распределения затрат - 40-20-40:

- на анализ и проектирование приходится 40% затрат (из них на планирование и сбор требований - 5%);
- на кодирование - 20%;
- на тестирование и отладку - 40%.

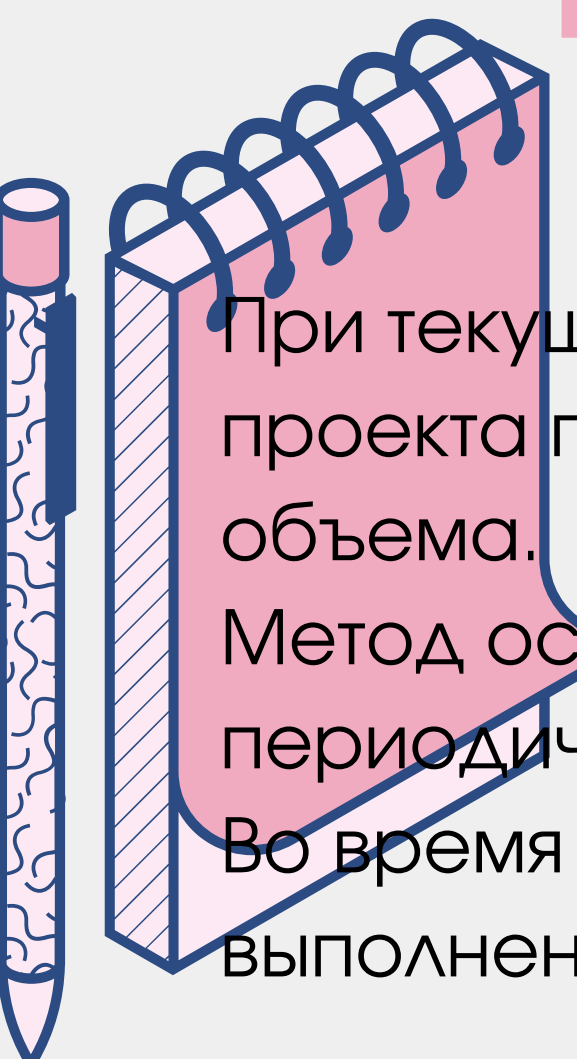


Данное правило отражает накопленный мировой опыт индустрии программной инженерии и базируется на следующих фактах:

- Работы, связанные со сбором и формализацией требований, определением и детализацией архитектуры ПО, наиболее трудоемки. Они требуют привлечения специалистов высочайшей квалификации, работающих в условиях существенной неопределенности.
- Кодирование (программирование) хорошо проработанных проектных решений имеет меньшую сложность. Здесь разработка переходит на тактический уровень. Степень определенности существенно выше, применяются хорошо известные, испытанные приемы и методики.
- Трудоемкость тестирования и отладки очень велика. Это обусловлено доминирующим стилем человеческой деятельности: "методом проб и ошибок". На поиск и устранение ошибок тратится много усилий.



# Контроль хода программного проекта - метод освоенного объема

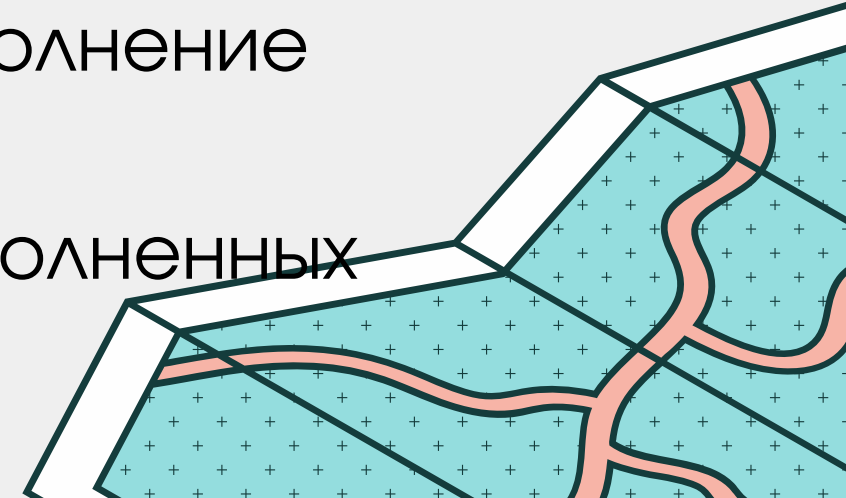


При текущем контроле программного проекта необходимо анализировать отклонения проекта по срокам и затратам. Делается это при помощи метода освоенного объема.

Метод освоенного объема EVM (Earned value management) предусматривает периодическую оценку состояний проекта для прогнозирования будущего.

Во время оценки состояния измеряется ход исполнения расписания и стоимость выполненных работ. Затем показывается окончательная стоимость проекта.

С помощью EVM измеряют и отслеживают три ключевых показателя:

- 1) Освоенный объем EV (Earned Value) - плановая стоимость выполненных работ (в терминах утвержденного бюджета, выделенного на эти работы);
  - 2) Плановый объем PV (Planned Value) - плановая стоимость работ, выполнение которых было запланировано к текущему моменту.
  - 3) Фактическая стоимость AC (Actual Cost) - фактическая стоимость выполненных работ.
- 

Суть оценки состоит в следующем. Сначала определяется отклонение по срокам SV (Schedule Variance) в денежных единицах:  $SV = EV - PV$ .

Допустим, что на текущий момент реализовано 30 требований, на каждое из которых было запланировано затратить по 40 000 руб., поэтому освоенный объем равен:  $EV = 30 * 40\,000 = 1\,200\,000$  руб.

Если же на текущий момент планировалось реализовать 40 требований, то плановый объем равен:  $PV = 40 * 40\,000 = 1\,600\,000$  руб.

Следовательно, проект отстает по срокам (отклонение отрицательное) на величину:  $SV = EV - PV = 1\,200\,000 - 1\,600\,000 = -400\,000$  руб.

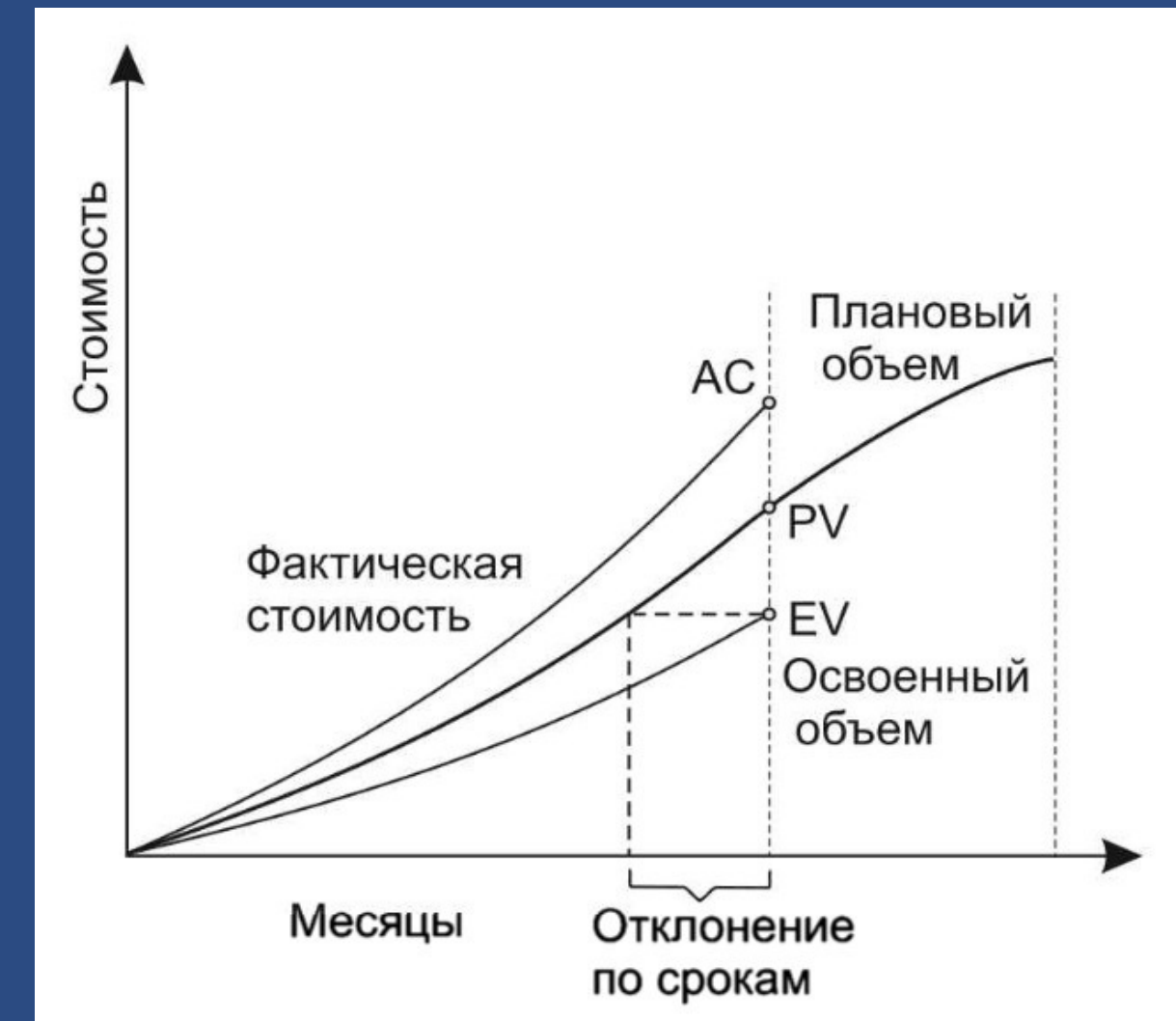
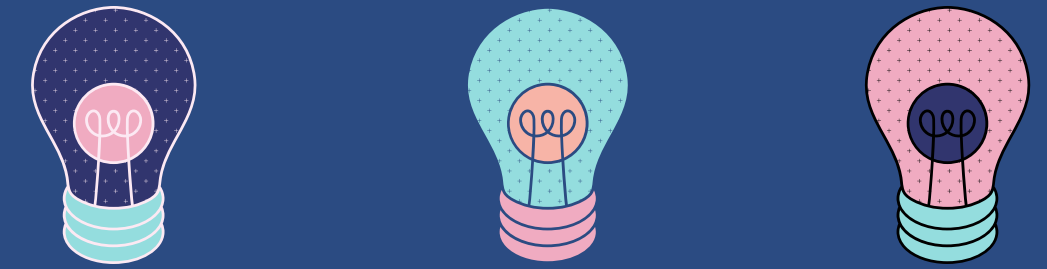
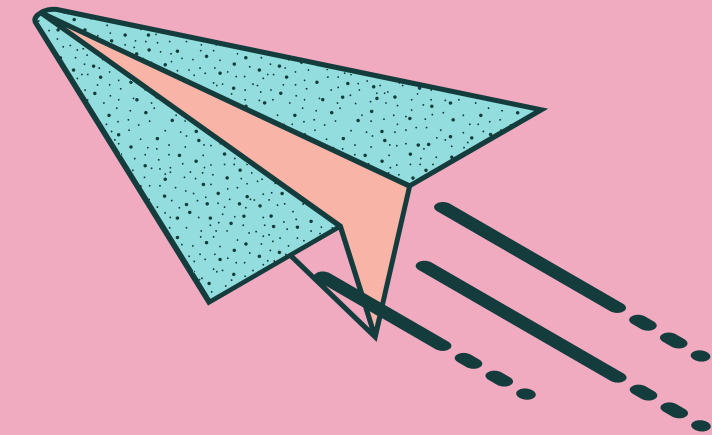


График освоенного объема

Далее вычисляется отклонение по стоимости CV (Cost Variance):  $CV = EV - AC$ .  
Если работу выполняли приглашенные специалисты с коэффициентом оплаты 1,5, то фактически стоимость составила:  $AC = 30 * 40\,000 * 1,5 = 1\,800\,000$  руб.

Поэтому отклонение по затратам в нашем случае равно:  $CV = EV - AC = 1\,200\,000 - 1\,800\,000 = -600\,000$  руб.

Отрицательная величина свидетельствует о перерасходе денежных средств.



При определении отклонений руководствуются следующими принципами:

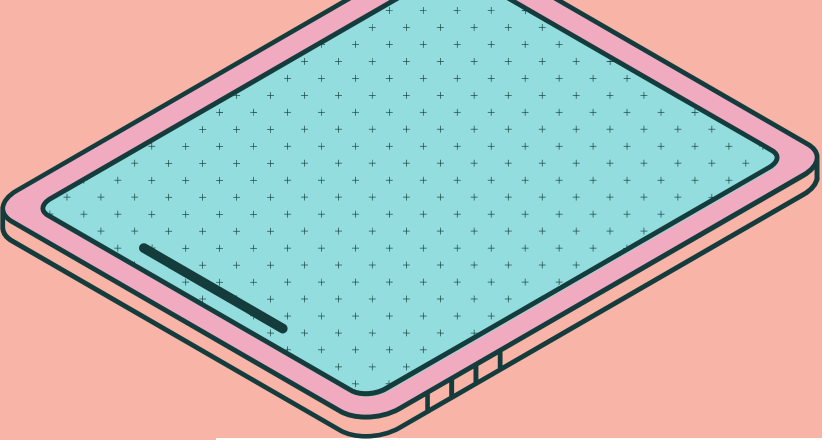
- отрицательное отклонение является "плохим";
- отклонения по стоимости и срокам рассчитываются как разность между освоенным объемом и какой-то другой мерой.

Отклонения SV и CV, выражающиеся в абсолютных денежных единицах, неудобны для сравнения разномасштабных проектов. Более универсальны относительные показатели:

- индекс выполнения сроков SPI (Schedule Performance Index):  $SPI = EV / PV$ ;
- индекс выполнения стоимости CPI (Cost Performance Index):  $CPI = EV / AC$ .

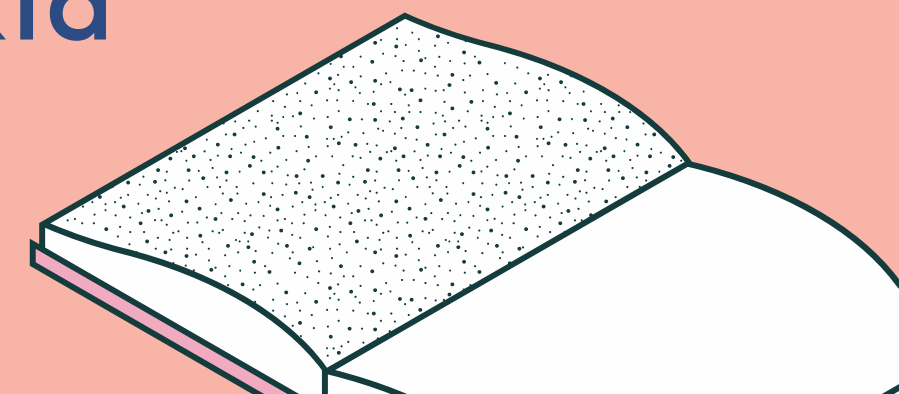
Они характеризуют проект независимо от его размера. Если значения обоих индексов больше единицы, то это свидетельствует о благополучном положении, а если меньше - о проблемах в проекте.





Показатели		Расписание		
		<b>SV &gt; 0 &amp; SPI &gt; 1.0</b>	<b>SV = 0 &amp; SPI = 1.0</b>	<b>SV &lt; 0 &amp; SPI &lt; 1.0</b>
Стоимость	<b>CV &gt; 0 &amp; CPI &gt; 1.0</b>	Опережение расписания Экономия бюджета	Точно по расписанию Экономия бюджета	Отставание от расписания Экономия бюджета
	<b>CV = 0 &amp; CPI = 1.0</b>	Опережение расписания Точно по бюджету	Точно по расписанию Точно по бюджету	Отставание от расписания Точно по бюджету
	<b>CV &lt; 0 &amp; CPI &lt; 1.0</b>	Опережение расписания Превышение бюджета	Точно по расписанию Превышение бюджета	Отставание от расписания Превышение бюджета

Возможные состояния программного проекта

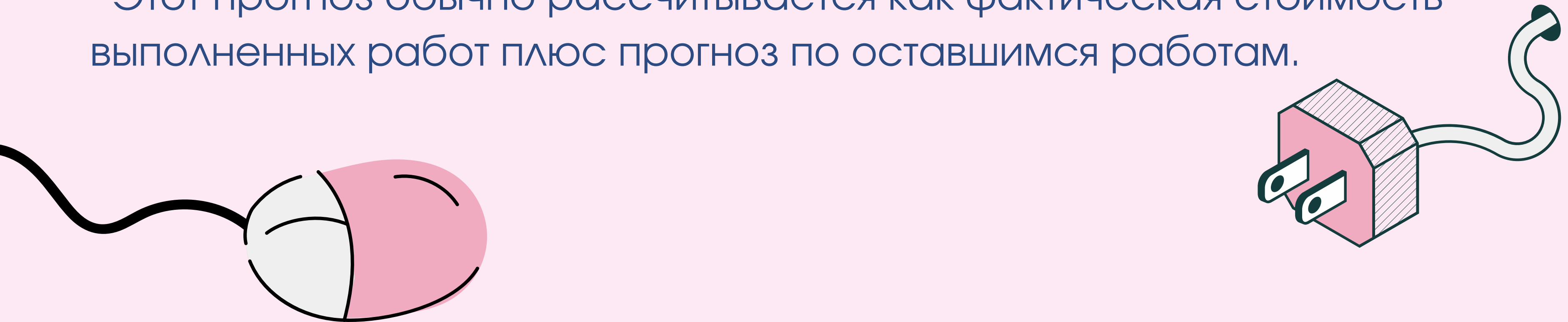




Самая сильная черта EVM - возможность предсказывать окончательные результаты проекта в течение большей части времени его исполнения. Предсказание становится обоснованным и статистически надежным только при условии выполнения более 15% работ.

Прогноз денежных затрат на момент завершения проекта может существенно отличаться от спланированного заранее бюджета по завершению (БПЗ).

Этот прогноз обычно рассчитывается как фактическая стоимость выполненных работ плюс прогноз по оставшимся работам.



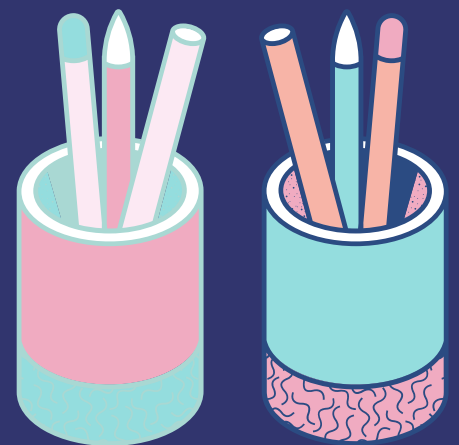
При прогнозе, как правило, используются кумулятивные значения SPI и CPI.

Методы прогноза:

1) Прогноз для варианта, когда оставшаяся работа будет оплачиваться по заложенным в бюджет ставкам:  $\text{Прогноз} = AC + (\text{БПЗ} - EV)$ ;

2) Прогноз для случая, когда для оставшейся работы будет справедлив текущий CPI. Допускается, что в будущем проект продолжится с тем же индексом выполнения стоимости, какой был достигнут к текущему моменту:  $\text{Прогноз} = \text{БПЗ} / \text{CPI}$ ;

3) Прогноз с учетом обоих текущих индексов, как по стоимости, так и по срокам:  $\text{Прогноз} = AC + ((\text{БПЗ} - EV) / (\text{SPI} * \text{CPI}))$ .



СПАСИБО ЗА ВНИМАНИЕ!

