



# Управление IT проектами

---

## Часть 1

### Введение

Жизненный цикл IT проекта  
Планирование, Gantt-диаграммы  
Оценка проектов



# Программа

---

- Введение
- Планирование
- Управление Рисками
- Контроль хода выполнения
- Финансовое обоснование проекта
- Управление контрактами
- Управление конфигурацией
- Модели проектных организаций



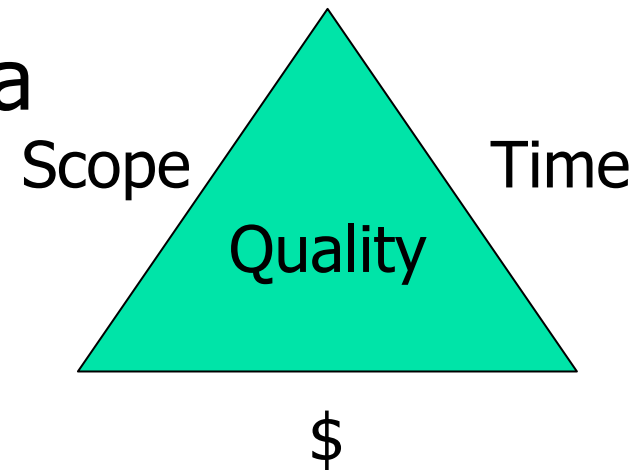
# Базис

---

- PMBOK стандарта PMI (Project Management Institute)
- Управление проектами (программа MBA)
- CMMI SEI (Capability Maturity Model® Integration от Software Engineering Institute)

# Определение Проекта

- Проект – временное предприятие, предназначенное для создания уникального продукта или услуги (РМВОК)
- Характеристики проекта
  - Цель (Scope)
  - Время (Timeline)
  - Цена (Cost)

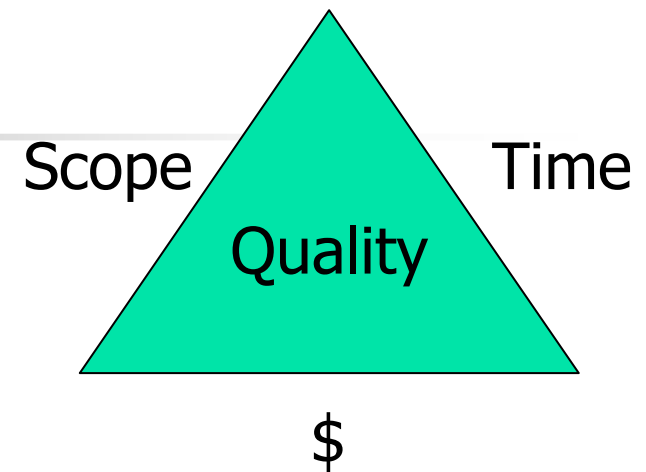




# Что является проектом?

Разработка системы	Да
Поддержка системы	Условно
Внедрение системы поддержки бизнеса	Да
Подготовка плана разработки нового программного продукта	Да

# Наша цель



- **Успешный проект**
  - Отвечает требованиям заинтересованных лиц (stakeholders)
  - Укладывается в сроки и бюджет
  - Выполняется с надлежащим качеством
- **Наша цель:**
  - Понять, как сделать проект успешным



# Project Management

---

- Project Management – использование знаний, навыков, методов, средств и технологий для выполнения проекта с целью удовлетворения потребностей и ожиданий участников (как заказчика, так и исполнителя)



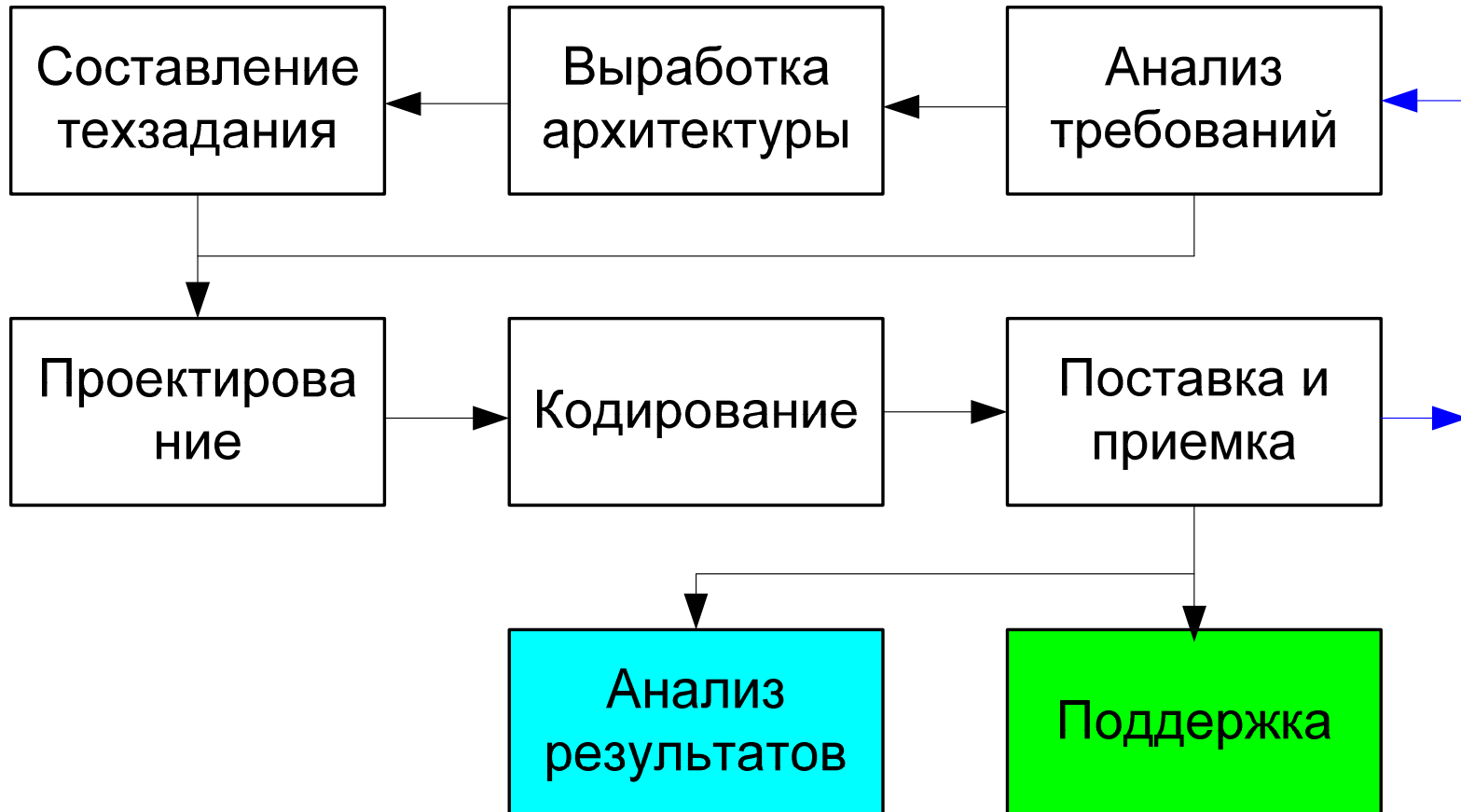
# Project Management Lifecycle

- Дикий энтузиазм
  - Разочарование
  - Полная путаница
  - Поиск виноватых
  - Наказание невиновных
  - Награждение непричастных
- (из лекций Michael Newell, PMP)

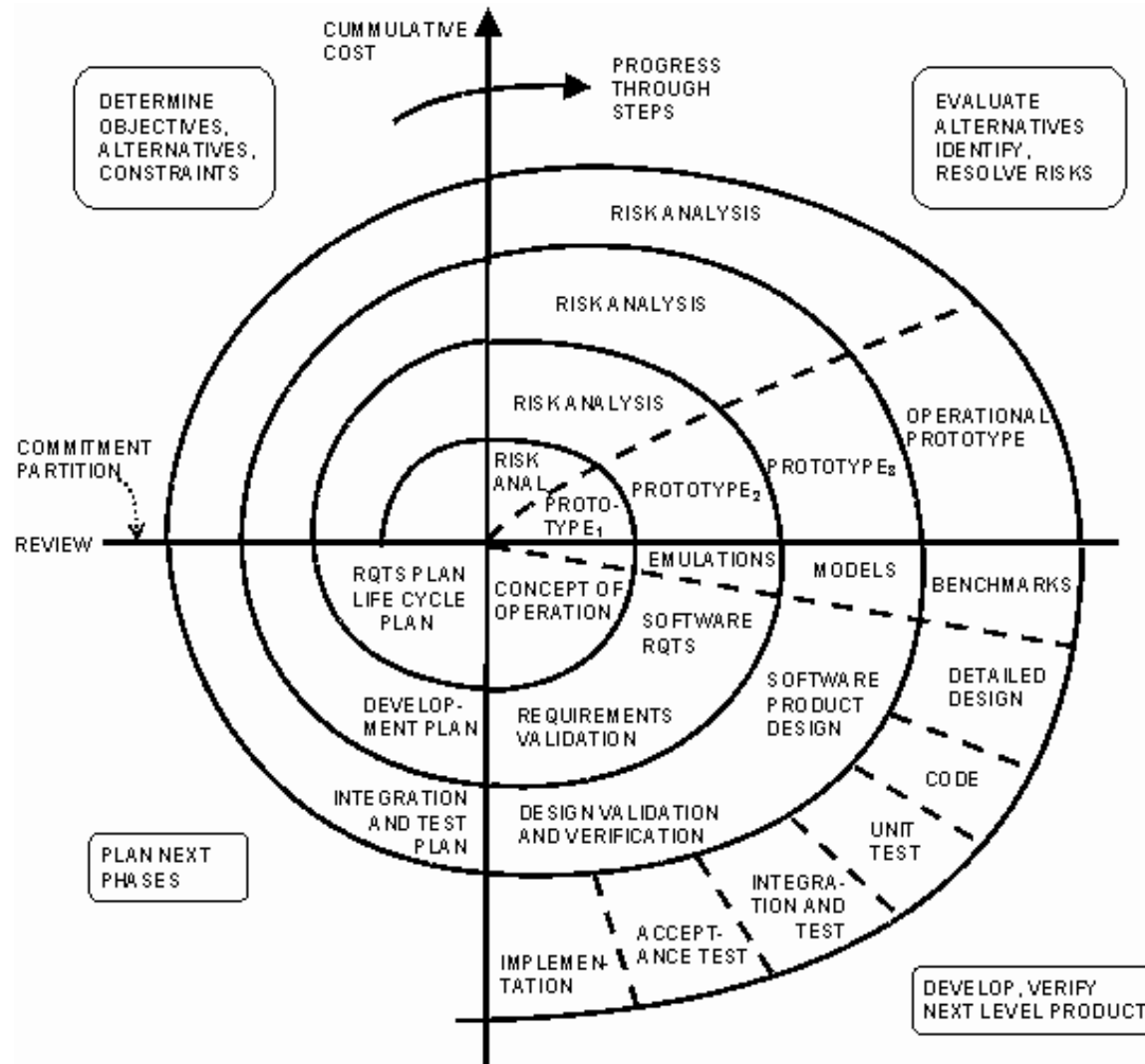
# Жизненный цикл IT проекта

Inception	Начало
Elaboration	Уточнение
Construction	Разработка
Transition	Внедрение
Support	Поддержка
Post-mortem procedures 	Ликвидация

# Жизненный цикл IT проекта

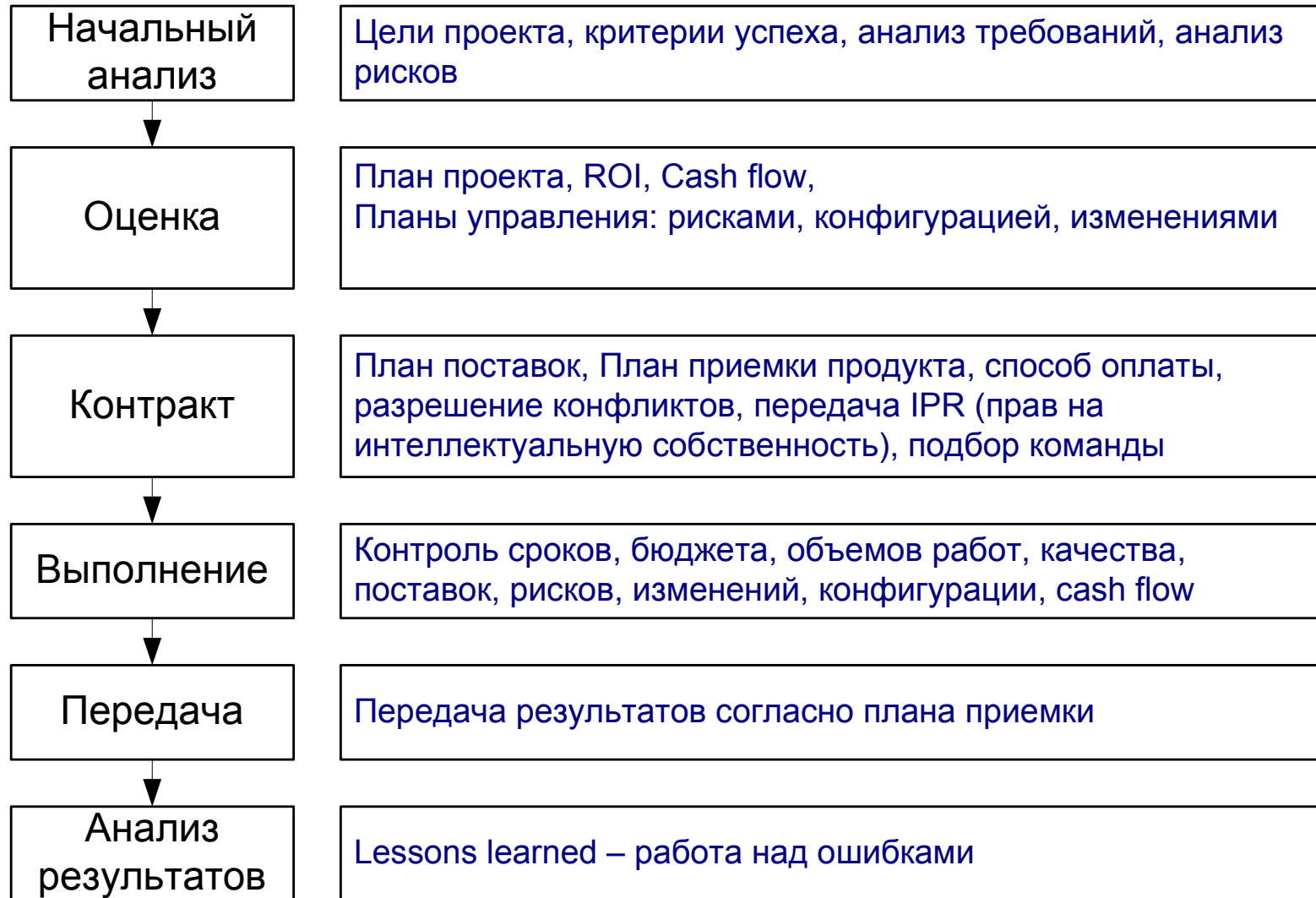


# Spiral development





# Жизненный цикл управления проектом





# Управление IT проектами

---

## Начальный анализ



# Начальный анализ

---

- Цели проекта
- Критерии успеха
- Требования
- Ограничения
- Анализ рисков

Артефакты: Vision, SRS (частично)



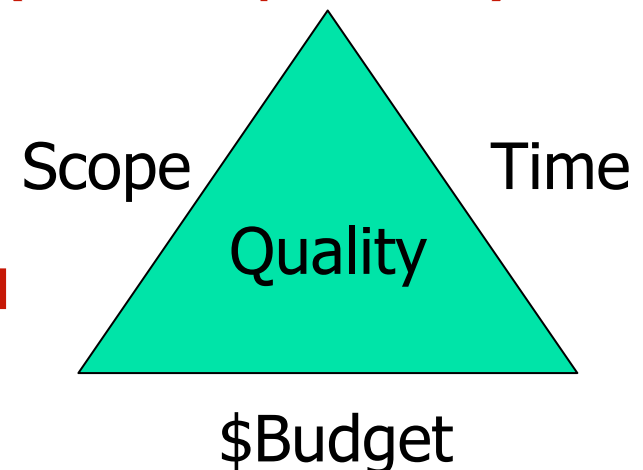
# Управление IT проектами

---

## Планирование

# Планирование

- Что такое ПЛАН?
- последовательность **действий команды** проекта по достижению:
  - Заявленных **результатов (цели проекта)**
  - В указанные **сроки**
  - В рамках **бюджета**
  - С надлежащим **качеством**





## Результаты планирования: Score (Цели)

---

- Результаты проекта (реализация требований, функциональных и не функциональных)
- Объекты поставок (финальной и промежуточных)
- WBS / декомпозиция проекта на набор четко формулируемых задач



## Результаты планирования: Cost (Стоимость)

---

- Трудозатраты на каждую задачу
- Ресурсы каждой задачи
- Стоимость рисков
- Бюджет проекта
  - Operating Budget / Оперативный
  - Contingency Budget / Покрытия рисков
  - Management Reserve / Резерв



# Timeline (График и сроки)

---

- Tasks Relationships / Связи задач
- Tasks Estimations / Оценки задач
- Resource availability / Доступность ресурсов в нужное время
- Schedule baseline / График
- Milestones / Контрольные точки



# Quality (Качество)

---

- Метрики качества
- План тестирования
- План приемки
- Внутренний аудит



# План составления плана

---

Как составить план проекта:

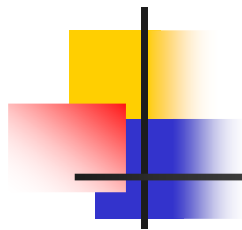
- Провести декомпозицию работ (WBS)
- Установить зависимости задач
- Оценить стоимость ресурсов и продолжительность задач / (PERT analysis)
- Установить требования к квалификации персонала
- Назначить ресурсы
- => Определить время исполнения и цену проекта



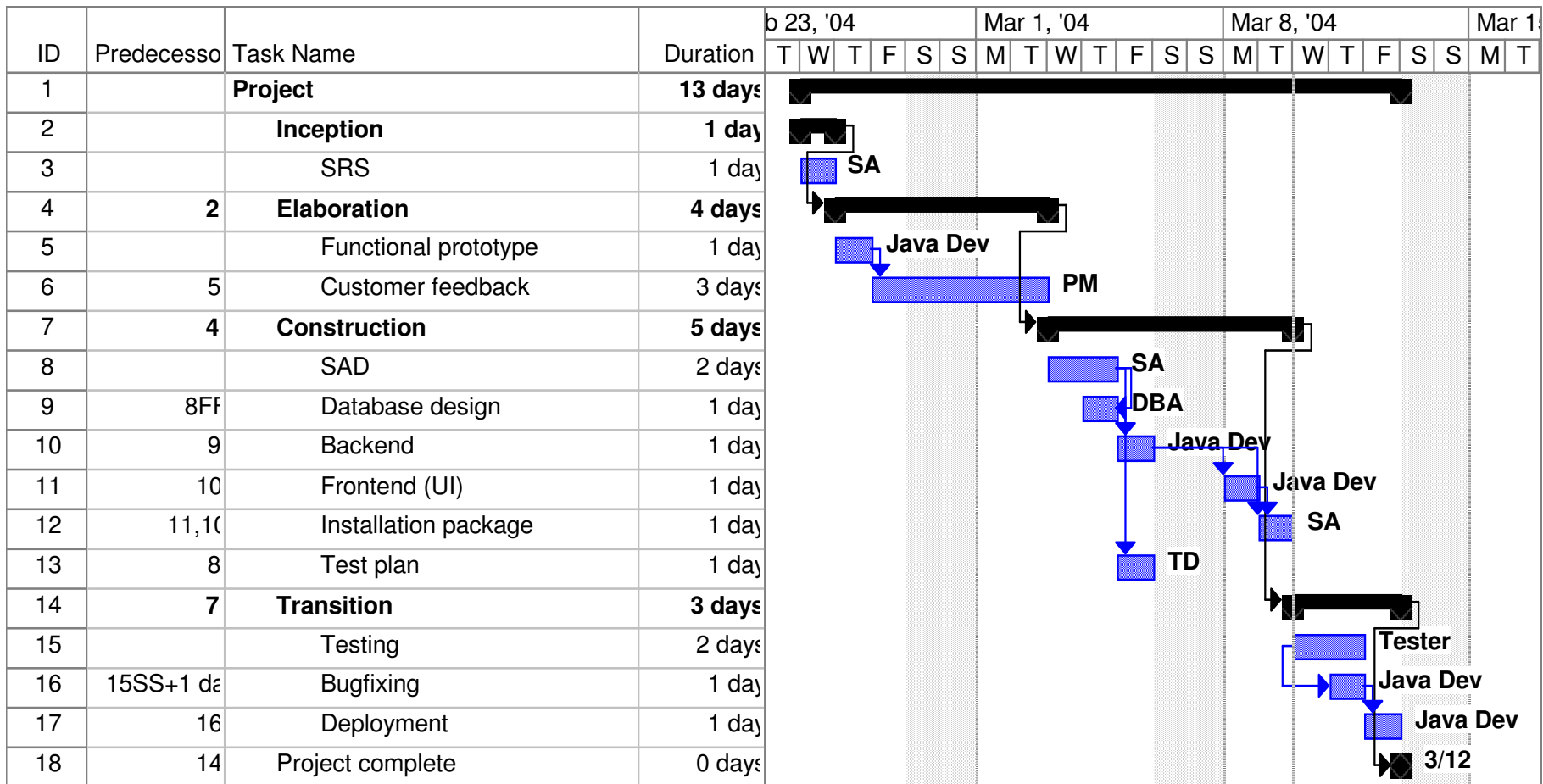
# Work Breakdown Structure

---

- Что такое WBS:
  - Последовательность ВСЕХ задач, которые нужно выполнить для завершения проекта
- Что нужно для построения WBS?
  - Project objectives / цели проекта
  - Project deliverables and requirements / объекты поставок и требования
  - Problem domain expertise / экспертиза



# Пример WBS (Gantt chart)





# Зависимости задач

---

- Типы зависимостей
  - FS, Finish-to-Start
  - FF, Finish-to-Finish
  - SS, Start-to-Start
  - SF, Start-to-Finish (?!)
- Input-based (FS)
- Parallel tasks (FF, SS)

*Зависимости должны быть логическими!  
Избегайте зависимости по ресурсам !  
Используйте leveling delay*



# WBS tips

---

- Каждый уровень WBS должен содержать ВСЕ элементы, необходимые для выполнения проекта
- Метод входов/выходов
  - Каждая задача имеет вход (другую задачу или исходные материалы)
  - Каждая задача имеет выход (другую задачу или объект поставки)
  - **Вопрос:** Возможны ли исключения?



# Meta Plan

---

Как составить план проекта:

- Провести декомпозицию работ (WBS)
- Установить зависимости задач
- **Оценить стоимость ресурсов и продолжительность задач / (PERT analysis)**
- Установить требования к квалификации персонала
- Назначить ресурсы
- Определить время исполнения и цену проекта



# Оценка проекта

---

- Точность оценок
  - Order of Magnitude      -25% to +75%
  - Budget                      -10% to 25%
  - Definitive                  -5% to 10%
  
- Методы оценок
  - Expert
  - Analog
  - Parametric



# Аналоговая оценка

---

- Использование опыта компании
- Algorithm:
  - Найти похожий проект(задачу)
  - Понять различия
  - Зная фактическую стоимость похожего проекта (задачи), оценить с учетом различий



# Параметрическая оценка

- Использование моделей
- Каждая модель хороша в определенных условиях
  - COSOMO (LoC) – напр. Портирование
  - FPA (Function points analysis)
  - U/C points (G.Kamer, Rational, 1993)
  - Use-Case metrics (из опыта Web-проектов: 20 h/UC) – простые клиент-сервер системы



# Use-case points

---

- Actor type (1-3) : UAW
- Use case type (1-3) : UUCW
- $UUCP = UAW + UUCW$
- Technical complexity  $TCF = 0.6 + (0.01 * TF)$
- Environmental factor  $EF = 1.4 + (-0.03 * EF)$
- Adjusted UCP =  $UUCP * TCF * EF$
- 20m/h per Adjusted UCP



# Actor types

---

- 1: Simple - другая система с известным API
- 2: Average - система, взаимодействующая через протокол)
- 3: Complex - персона, взаимодействующая через GUI или Web-page



# Use-case types

---

- 1: Simple (<5 аналитических классов)
- 2: Average (5-10 аналитических классов)
- 3: Complex (>10 аналитических классов)

Другой способ:

- 1: Simple (простой UI, 1 DB entity)
- 2: Average (более сложный UI, >2 DB entities)
- 3: Complex (сложный UI, >3 DB entities)



# UCP: Technical factors

---

- Distributed system: weight = 2
- Complex processing: weight=1
- Easy to install: weight=0.5
- Portable: weight=2

$$TCF=0.6+SUM_i (0.01*TF_i)$$



# UCP: Environment factors

---

- Familiar with RUP: 1.5
- Stable requirements: 2
- Part-time workers: -1
- Object oriented experience: 1

$$EF = 1.4 + \text{SUM}_i(-0.03 * EF_i)$$



# Экспертная оценка

---

- Каждая задача оценивается экспертом
- Требуется четкого определения ВСЕХ задач
- Требуется наличия экспертов по всем типам задач в самом начале проекта



# PERT Analysis

---

- PERT - Program Evaluation and Review Technique
- Используется при экспертной оценке
- Увеличивает вероятность правильной оценки
- Использует 3 уровня оценки на задачу



## 3 уровня оценки

---

- Optimistic – дается в предположении что все пойдет хорошо
- Most Likely – наиболее вероятная оценка, дается с учетом опыта
- Pessimistic – дается в предположении наихудшего развития событий



# Вопросы

---

- Имеется набор случайных нормально распределенных величин  $W_i$  с дисперсией  $D_i = S_i^2$
- Чему равна дисперсия суммы?
- Что можно сказать о распределении суммы, если распределение составляющих суммы нельзя считать строго нормальным?

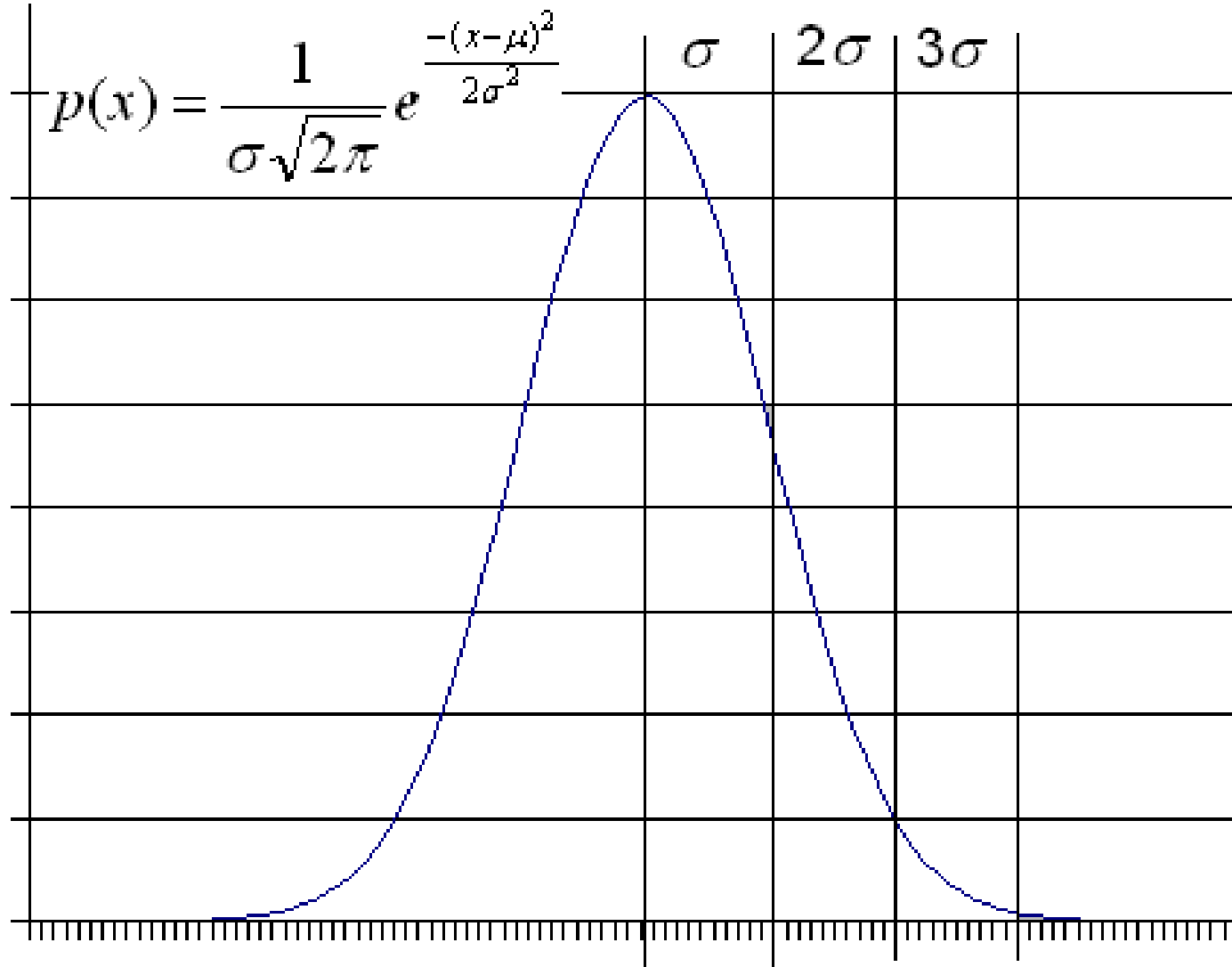


# Ответы

---

- $S = \sqrt{\sum S_i^2}$
- Центральная предельная теорема
- ... И важный вывод:
- Оценка, полученная суммой оценок более мелких задач даст в итоге меньшую ошибку

# Нормальное распределение

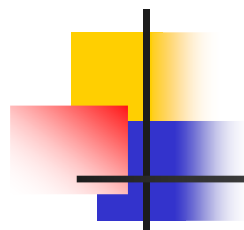




# PERT

---

- Исходим из гипотезы о нормальном распределении вероятности оценок задач
- $E_i = (O + 4*ML + P) / 6$
- $S_i = (P - O) / 6$
- $E = \text{Sum} ( E_i )$
- $S = \text{Sqrt} ( \text{Sum} ( S_i^{**2} ) )$



# Пример расчета PERT

	Optimistic	Most Likely	Pessimistic	Si	E	Si <sup>2</sup>
Задача 1	2	3	4	0,3	3,0	0,1
Задача 2	3	4	6	0,5	4,2	0,3
Задача 3	2	3	4	0,3	3,0	0,1
Задача 4	4	5	8	0,7	5,3	0,4
<b>TOTALS</b>	<b>11</b>	<b>15</b>	<b>22</b>		<b>15,5</b>	<b>0,957</b>

<b>FINAL ESTIMATE (+3S)</b>	<b>18</b>
-----------------------------	-----------



# Экспертная оценка

---

- Трудозатраты оцениваются по PERT
- Продолжительность проекта следует из плана работ
- Итоговая оценка верифицируется при помощи аналоговых или параметрических оценок



# Назначение ресурсов

---

- В плане не должно быть задач с не назначенными ресурсами
- Избегайте назначать несколько ресурсов на одну задачу (локализация ответственности)



# Ограничения задач

---

- Используются для задания условий на начало или окончание задачи
- 8 типов ограничений
  - **ASAP, As Soon As Possible** – default
  - **MSO, Must Start On** – just-in-time tasks
  - ALAP, As Late As Possible
  - FNET, Finish Not Earlier Than (on or after)
  - FNLT, Finish Not Later Than (on or before)
  - SNET, Start Not Earlier Than (on or after)
  - SNLT, Start Not Later Than (on or before)
  - MFO, Must Finish On



# Выравнивание ресурсов

---

- Automatic (поддерживается в MS Project)
- Manual – в ИТ проектах не так уж много одновременных задач, ручное выравнивание дает лучшие результаты



# Критический путь

---

- Последовательность задач, имеющая малый или нулевой запас (slack)
  - Slack – время, на которое можно задержать окончание задачи без угрозы срыва сроков проекта
- => Ошибка в оценке задач на критическом пути влечет ошибку в предсказании сроков проекта в целом



# Планирование времени выполнения

---

- Исходные данные: WBS, связи задач, назначенные ресурсы
- Используйте ограничения
- Выровняйте ресурсы
- Определите критический путь
- Запланируйте запасы на критическом пути (исходя из данных PERT)



# Планирование срывов плана

- “Promises kept are more important than early delivery”
- Не добавляйте месяц к концу плана – не спасет
- Используйте запасы на критическом пути – если проблема существует, она проявится раньше
- Планируйте критические задачи ближе к старту проекта



# Управление критическим путем

---

- На критическом пути найти задачи с малым запасом
- Уточнить их продолжительность (по возможности)
- Добавить запас для тех задач, чью продолжительность тяжело оценить ТОЧНО (использовать результаты PERT-анализа)



# Контрольные вопросы

---

- «Треугольник» проектного управления
- WBS – определение. Как связаны WBS и Gantt chart?
- PERT analysis – что это? Зачем используется?
- Типы оценок
- Критический путь – определение. Что дает нам знание критического пути?

# Контрольная задача

ID	Predecessor	Task Name	Duration	May 24, '04							May 31, '04						
				S	M	T	W	T	F	S	S	M	T	W	T	F	S
1		<b>Project</b>	<b>10 days</b>														
2		Task A	2 days														
3	2	Task B	4 days														
4	3SS	Task C	3 days														
5	3	Task D	4 days														
6	4,5	Project complete	0 days														

Укажите критический путь проекта и рассчитайте запас для критического пути при условии, что оценки задач соответствуют приведенным в таблице:

	Optimistic	Most Likely	Pessimistic
Task A	1	2	3
Task B	3	4	5
Task C	1	3	5
Task D	2	4	6