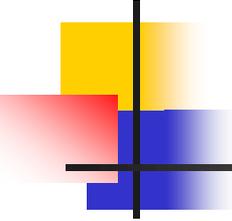


Управление IT проектами

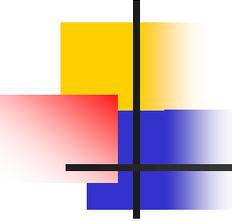
Часть 1

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

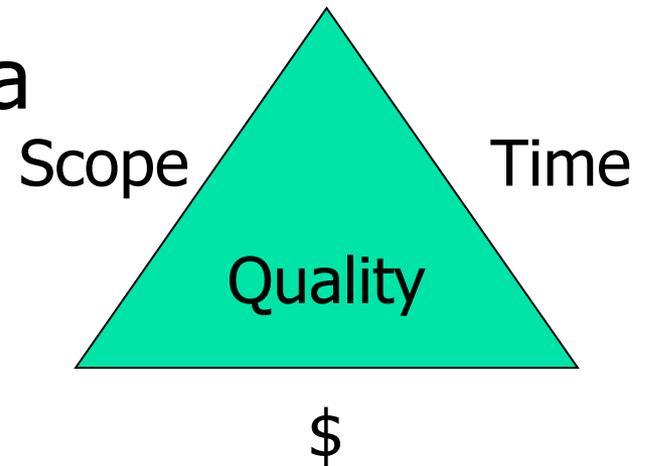


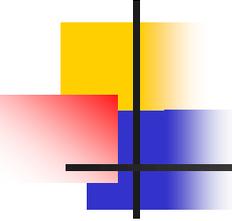
Базис

- 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)

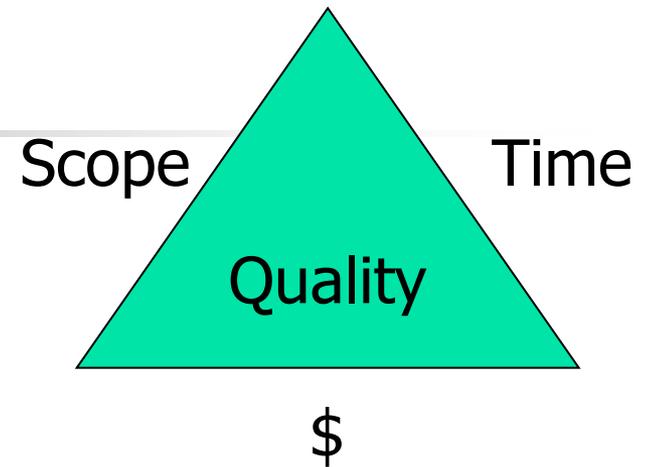




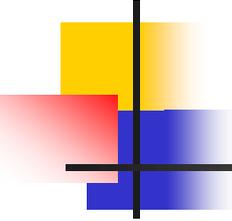
Что является проектом?

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

Наша цель

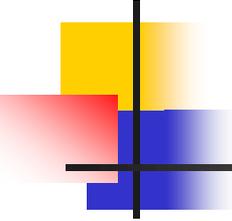


- Успешный проект
 - Отвечает требованиям заинтересованных лиц (stakeholders)
 - Укладывается в сроки и бюджет
 - Выполняется с надлежащим качеством
- Наша цель:
 - Понять, как сделать проект успешным



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)



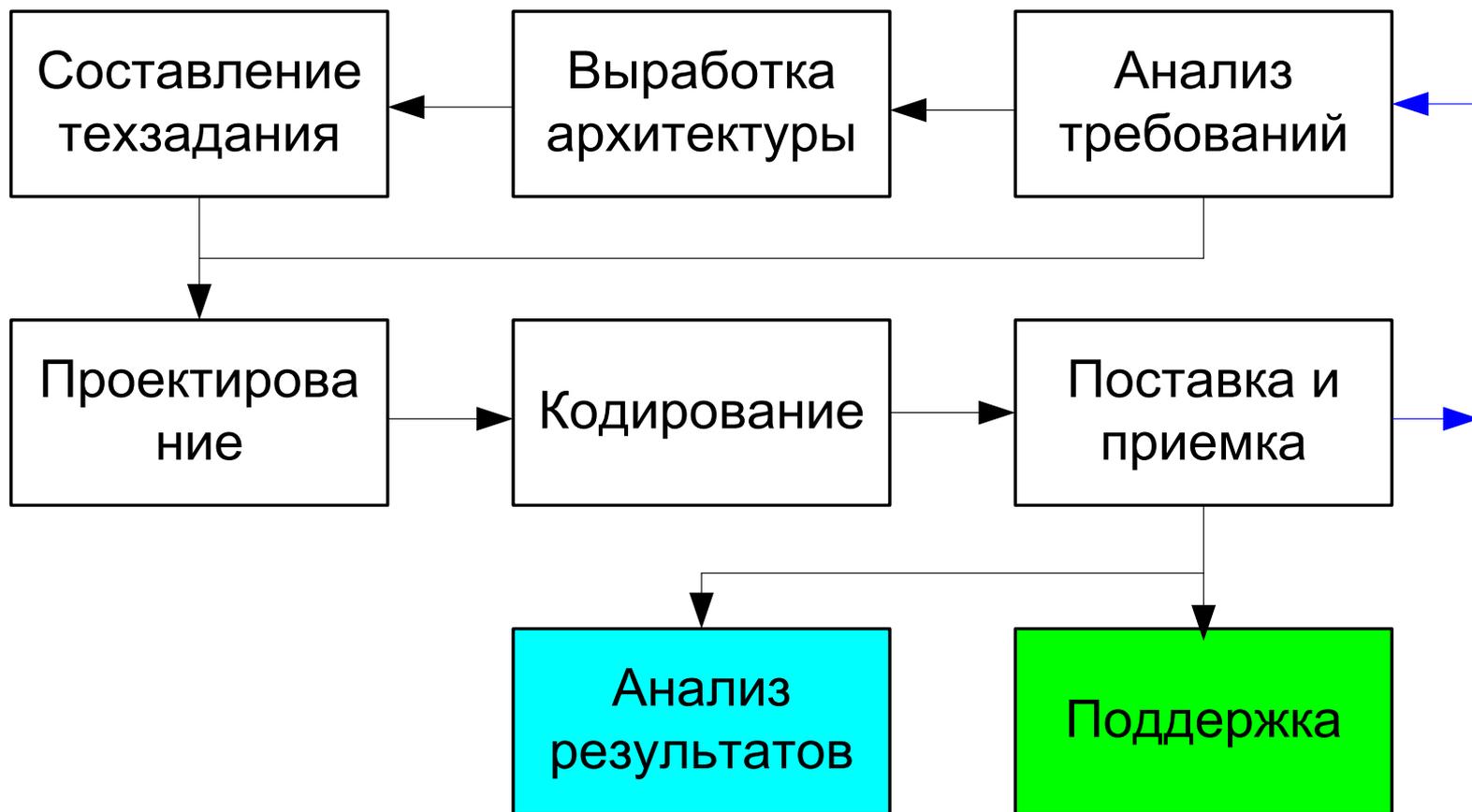
Project Management Lifecycle

- Дикий энтузиазм
 - Разочарование
 - Полная путаница
 - Поиск виноватых
 - Наказание невиновных
 - Награждение непричастных
- (шутка с долей шутки)

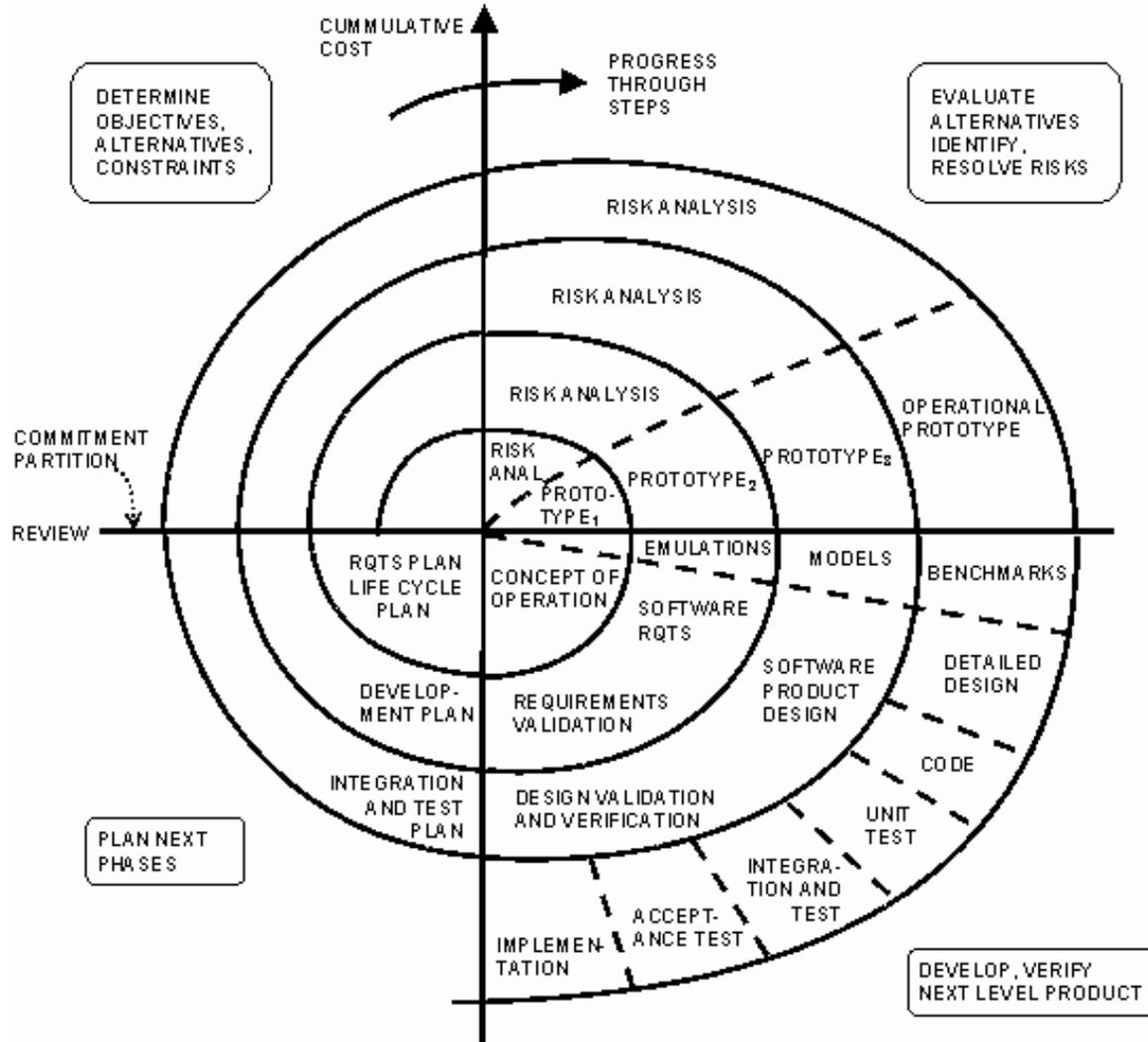
Жизненный цикл IT проекта

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

Жизненный цикл IT проекта



Spiral development

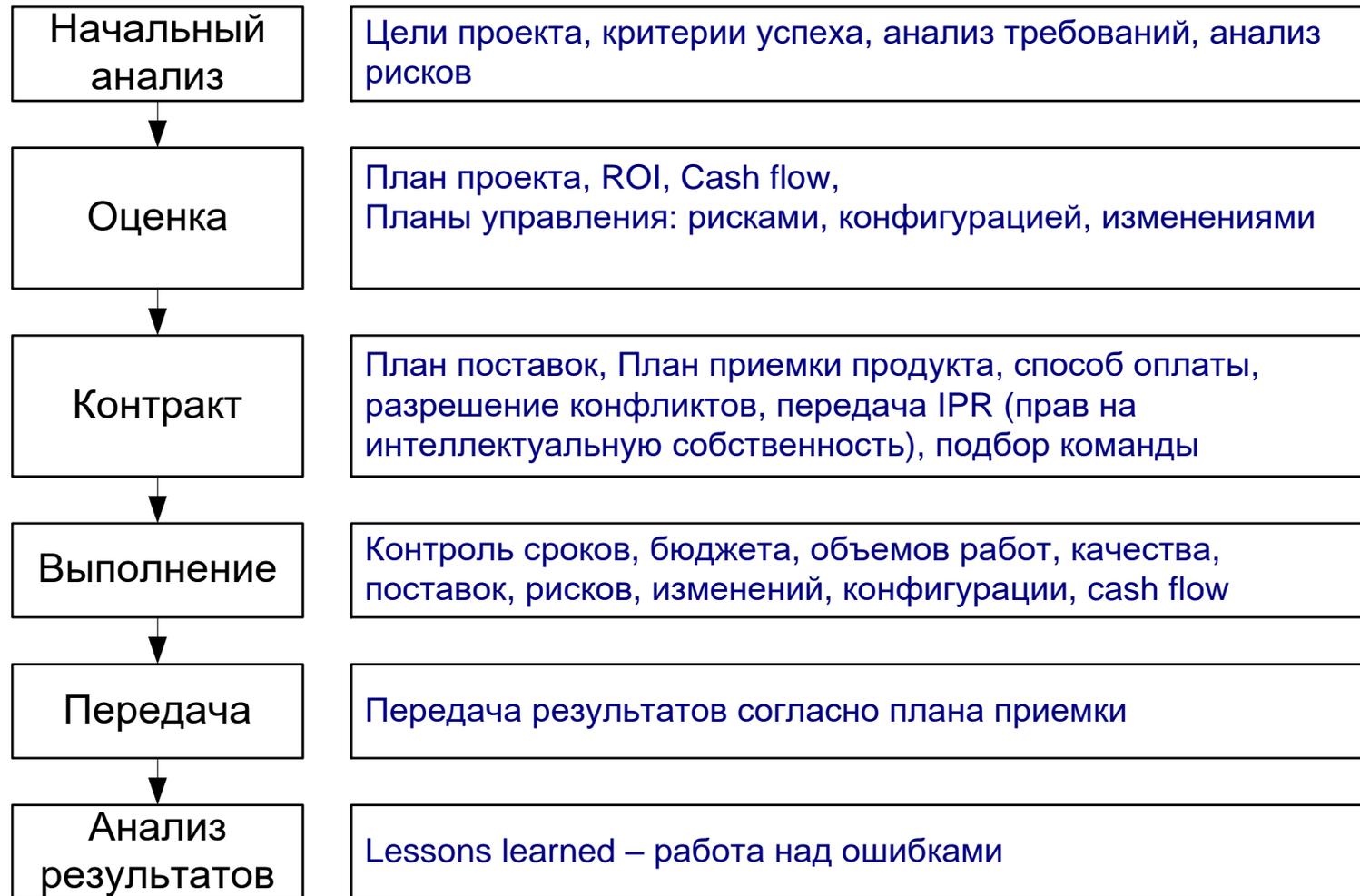


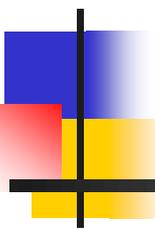
CRISP-DM

процесс для проектов Data mining



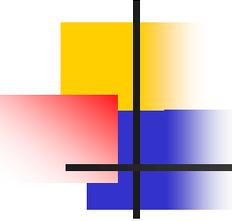
Жизненный цикл управления проектом





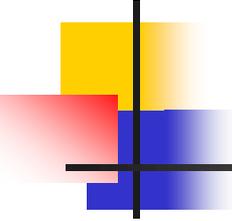
Управление IT проектами

Начальный анализ



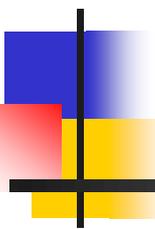
Начальный анализ

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



Артефакты анализа

- Документ Vision
 - в других областях управляемой деятельности называется Project Charter – Хартия проекта
 - Задает краткое описание предметной области, целей проекта и критериев успеха с точки зрения stakeholders (заинтересованных лиц)
- Спецификация требований
 - Еще не финальная версия
 - Описание требований
 - Описание ограничений
 - Начальный анализ проектных рисков

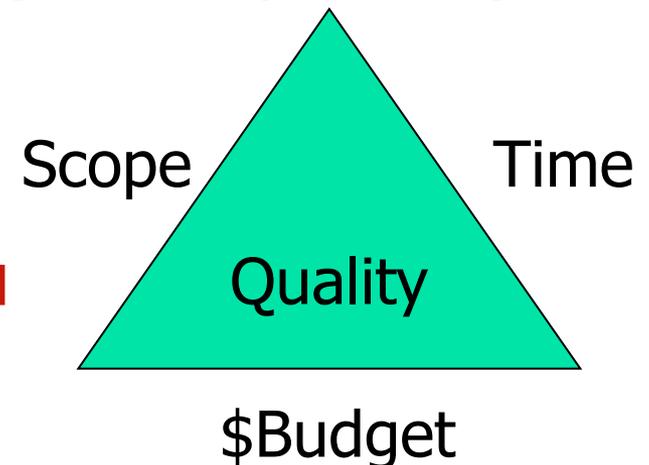


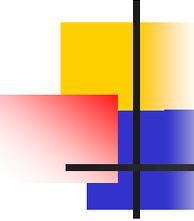
Управление IT проектами

Планирование

Планирование

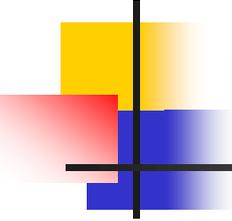
- Что такое ПЛАН?
- последовательность **действий команды** проекта по достижению:
 - Заявленных **результатов (цели проекта)**
 - В указанные **сроки**
 - В рамках **бюджета**
 - С надлежащим **качеством**





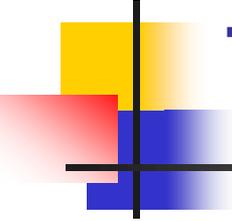
Результаты планирования: Score (Цели)

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



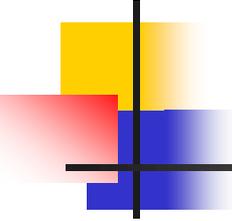
Результаты планирования: Cost (Стоимость)

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



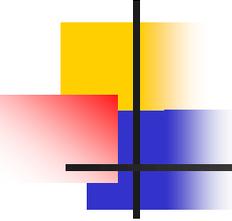
Timeline (График и сроки)

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



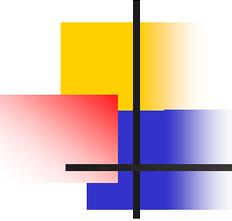
Quality (Качество)

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



План составления плана

- 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

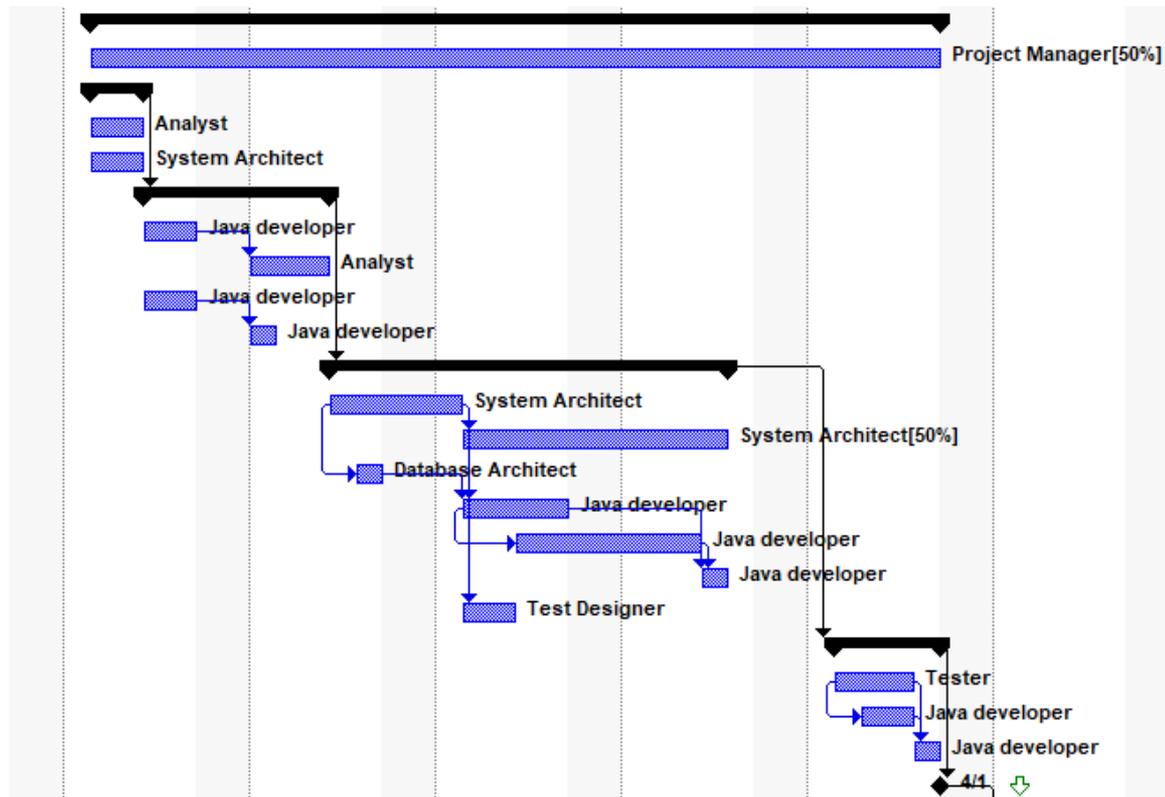


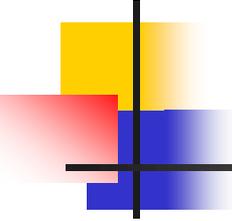
Work Breakdown Structure

- 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)

▲ Название проекта	24 days	392 hrs
Управление проектом	24 days	96 hrs
▲ Начало	2 days	32 hrs
Спецификация требований	2 days	16 hrs
Оценка проекта	2 days	16 hrs
▲ Уточнение	5 days	64 hrs
Функциональный прототип	2 days	16 hrs
Обсуждение прототипа	3 days	24 hrs
Архитектурный прототип	2 days	16 hrs
Стресс-тестирование	1 day	8 hrs
▲ Разработка	11 days	160 hrs
Разработка архитектуры	3 days	24 hrs
Контроль кода	8 days	32 hrs
Схема БД	1 day	8 hrs
Бизнес-логика	4 days	32 hrs
Пользовательский интерфейс	5 days	40 hrs
Инсталляционный пакет	1 day	8 hrs
План тестирования	2 days	16 hrs
▲ Приемка/передача	4 days	40 hrs
Тестирование	3 days	16 hrs
Исправление ошибок	2 days	16 hrs
Установка на полигон Заказчика	1 day	8 hrs
Проект завершен	0 days	0 hrs

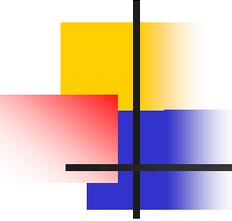




Зависимости задач

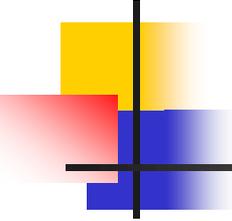
- Типы зависимостей
 - FS, Finish-to-Start
 - FF, Finish-to-Finish
 - SS, Start-to-Start
 - SF, Start-to-Finish (?!)
- Input-based (FS)
- Parallel tasks (FF, SS)

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



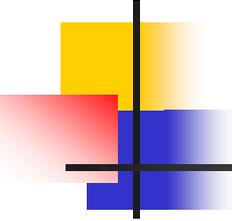
WBS tips

- 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?



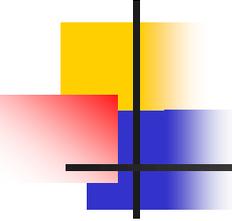
План составления плана

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



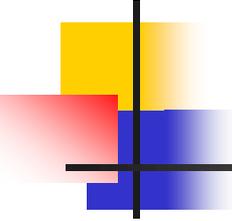
Оценка проекта

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



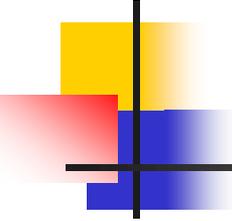
Аналоговая оценка

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



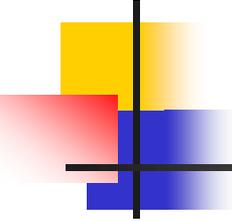
Параметрическая оценка

- Использование моделей
- Каждая модель хороша в определенных условиях
 - COSOMO (базовая величина KLoC, но откуда ее взять?)
 - FPA (Function points analysis)
 - U/C points (G.Kamer, Rational, 1993)
 - Use-Case metrics (из опыта Web-проектов: от 40 h/UC)



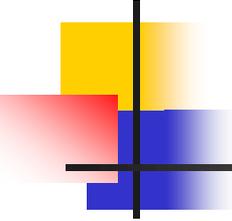
Use-case points

- Тип актера (1-3) : UAW
- Тип use-case (1-3) : UUCW
 - => $UUCP = UAW + 5 * UUCW$
- Техническая сложность
$$TCF = 0.6 + \sum_i (0.01 * TF_i)$$
- Фактор среды $EF = 1.4 + (-0.03 * EF)$
 - => $Adjusted\ UCP = UUCP * TCF * EF$
- Оценка: 15-30 m/h на Adjusted UCP



Тип актера

- 1: Простой - другая система с известным API
- 2: Средний - система, взаимодействующая через протокол)
- 3: Сложный - персона, взаимодействующая через GUI или Web

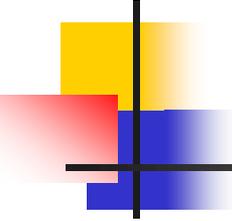


Тип use-case

- 1: Простой (<5 аналитических классов)
- 2: Средний (5-10 аналитических классов)
- 3: Сложный (>10 аналитических классов)

Другой способ:

- 1: Простой (простой UI, 1 DB entity)
- 2: Средний (более сложный UI, 2-3 DB entities)
- 3: Сложный (сложный UI, >3 DB entities)



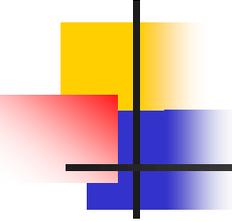
УСР: Техническая сложность

- TF – technical factor
 - Распределенная система : 2
 - Сложная обработка данных : 1
 - Легкость установки : 0.5
 - Переносимость : 2
 - ... всего 13 параметров, с суммой баллов 14
- Вес каждого TF умножаем на значимость для нашей системы по шкале от 0 до 5

Итоговая сложность

$$TCF = 0.6 + \text{SUM}_i(0.01 * TF_i)$$

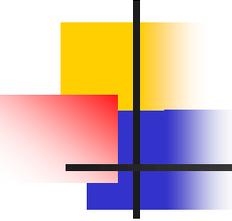
Варьируется от 0.6 до 1.3



UCP: Environment factors

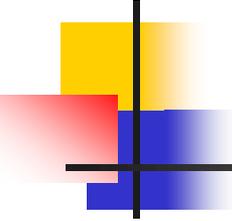
- EF – фактор окружения (команды)
 - Знакомство команды с UML: 1.5
 - Стабильность требований: 2
 - Сотрудники на part-time: -1
 - Опыт в OOP/OOAD: 1
 - Мотивация команды: 1
- Вес каждого EF умножаем на значимость для нашей системы по шкале от 0 до 5

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



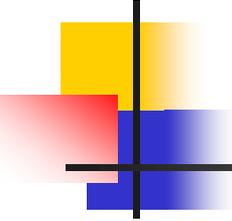
UCP: Итоговая оценка

- Трудозатраты = $20 \text{ m/h} * UUCP * TCF * EF$
- Недостатки очевидны:
 - Дает только оценку трудоемкости, ничего не говоря о сроках
 - Не учитывает риски проекта, особенностей требований по документации, качеству или производительности
- Хорошо подходит для принятия бюджетных решений вида: «имеет ли смысл» делать проект



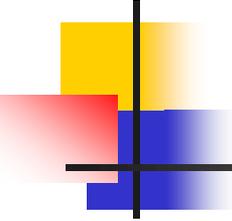
Экспертная оценка

- Каждая задача оценивается экспертом
- Требуется четкого определения ВСЕХ задач
- Требуется наличия экспертов по всем типам задач в самом начале проекта
- Самый трудоемкий и точный вид оценки



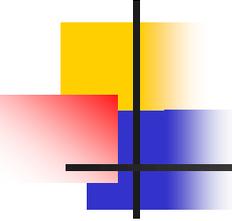
PERT Analysis

- PERT - Program Evaluation and Review Technique
- Используется при экспертной оценке
- Увеличивает точность оценки сроков и трудозатрат проекта
- Использует три (в упрощенном варианте два) уровня оценки на каждую задачу



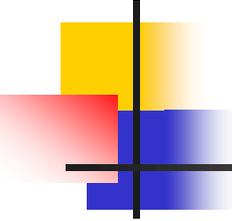
3 уровня оценки

- 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

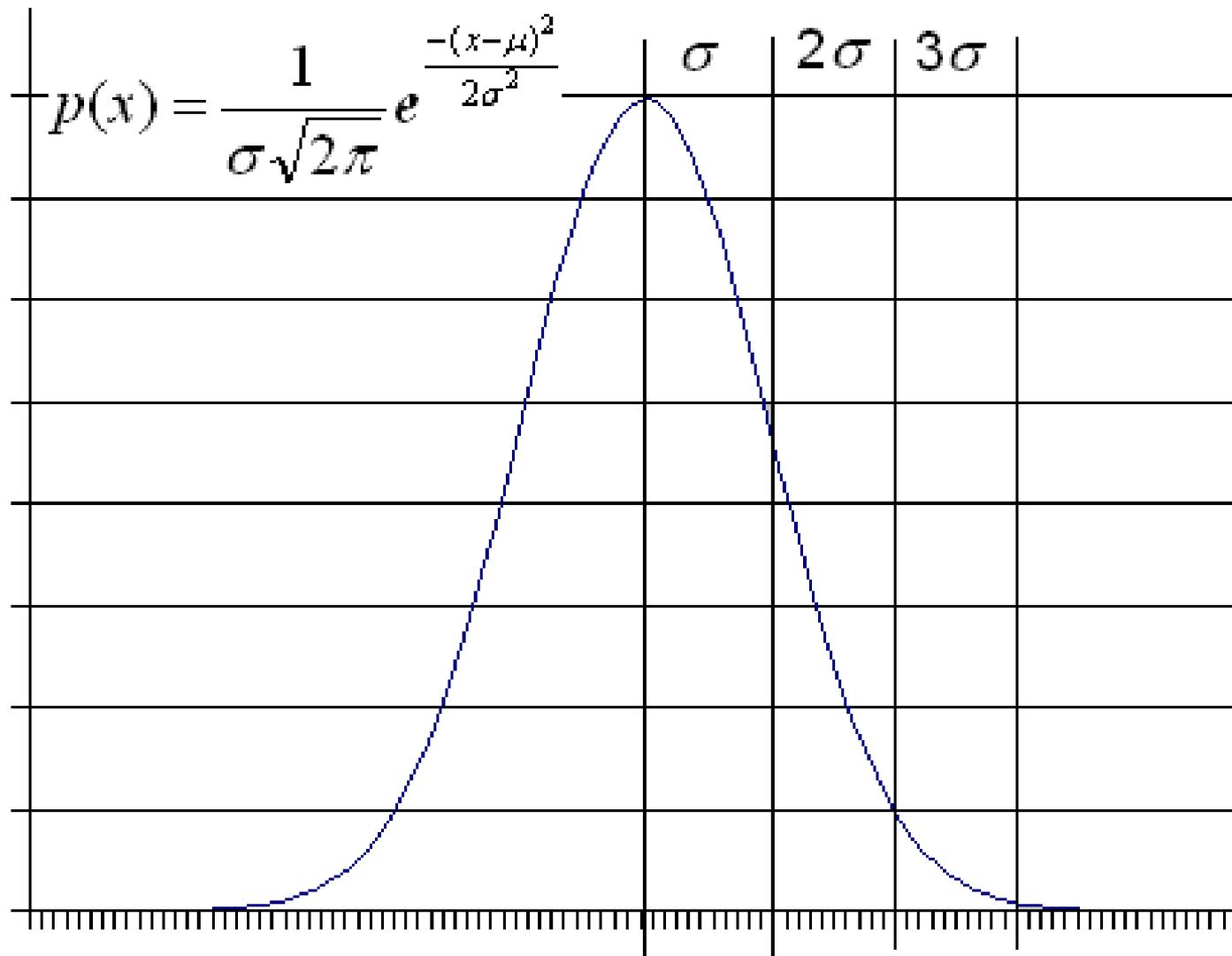
- Имеется набор случайных нормально распределенных величин W_i с дисперсией $D_i = S_i^2$
- Чему равно стандартное отклонение S суммы W_i ?
- Что можно сказать о распределении суммы, если распределение составляющих суммы нельзя считать строго нормальным?

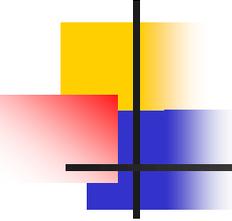


Ответы

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

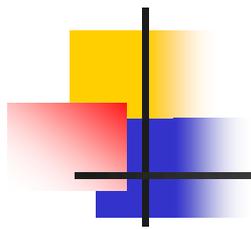
Нормальное распределение





PERT

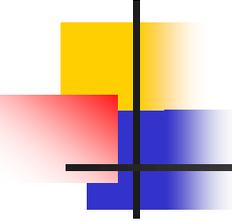
- Исходим из гипотезы о нормальном распределении вероятности оценок задач, тогда верны приближенные формулы:
 - $E_i = (O + 4*ML + P) / 6$
 - $S_i = (P - O) / 6$
 - $E = \text{Sum} (E_i)$
 - $S = \text{Sqrt} (\text{Sum} (S_i^{**2}))$



Пример PERT

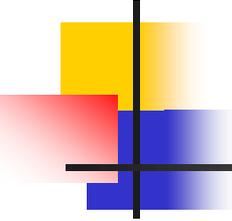
	Optimistic	Most Likely	Pessimistic	Si	E	Si ²
Задача 1	2	3	4	0,3	3,0	0,1
Задача 2	3	4	6	0,5	4,2	0,3
Задача 3	2	3	4	0,3	3,0	0,1
Задача 4	4	5	8	0,7	5,3	0,4
TOTALS	11	15	22		15,5	0,957

FINAL ESTIMATE (+3S)	18
-----------------------------	-----------



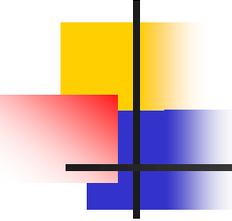
Экспертная оценка

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



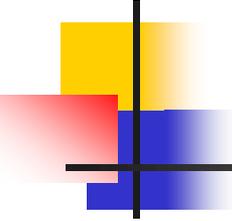
Назначение ресурсов

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



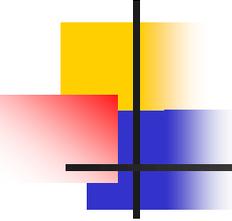
Ограничения задач

- 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



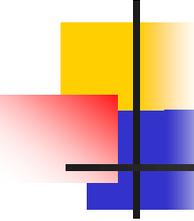
Выравнивание ресурсов

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



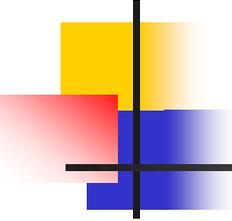
Критический путь

- 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



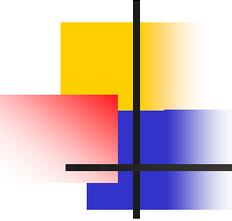
Планирование времени выполнения

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



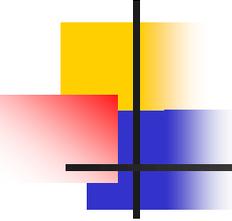
Планирование срывов плана

- “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		Project	10 days														
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