



# Управление IT проектами

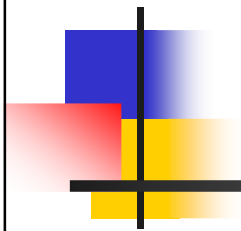
---

## Часть 3

Финансовое обоснование проекта.  
Контракты.  
Управление конфигурацией.  
Модели организаций.

<http://www.inteks.ru/PM/>

# Обоснование проекта



Окупаемость,  
Приведенная стоимость / Present value,  
Поток средств / Cash flow,  
Возврат инвестиций / ROI, Discounted ROI

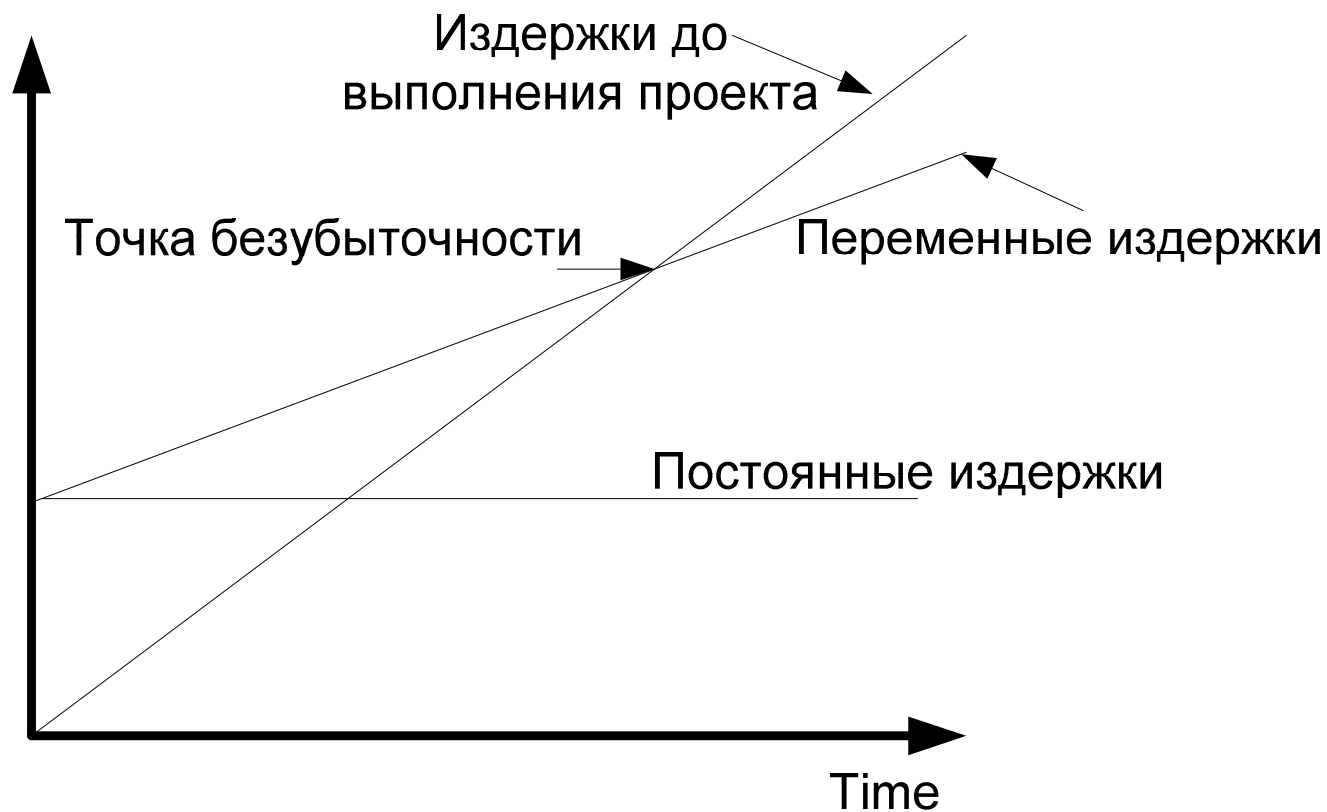


# Обоснование проекта

---

- Всякий ли проект действительно нужен? Выгоден?
- Как посчитать, выгодно ли проведение проекта?
- Как сравнить 2 альтернативных проекта?

# Анализ безубыточности (Break Even chart)



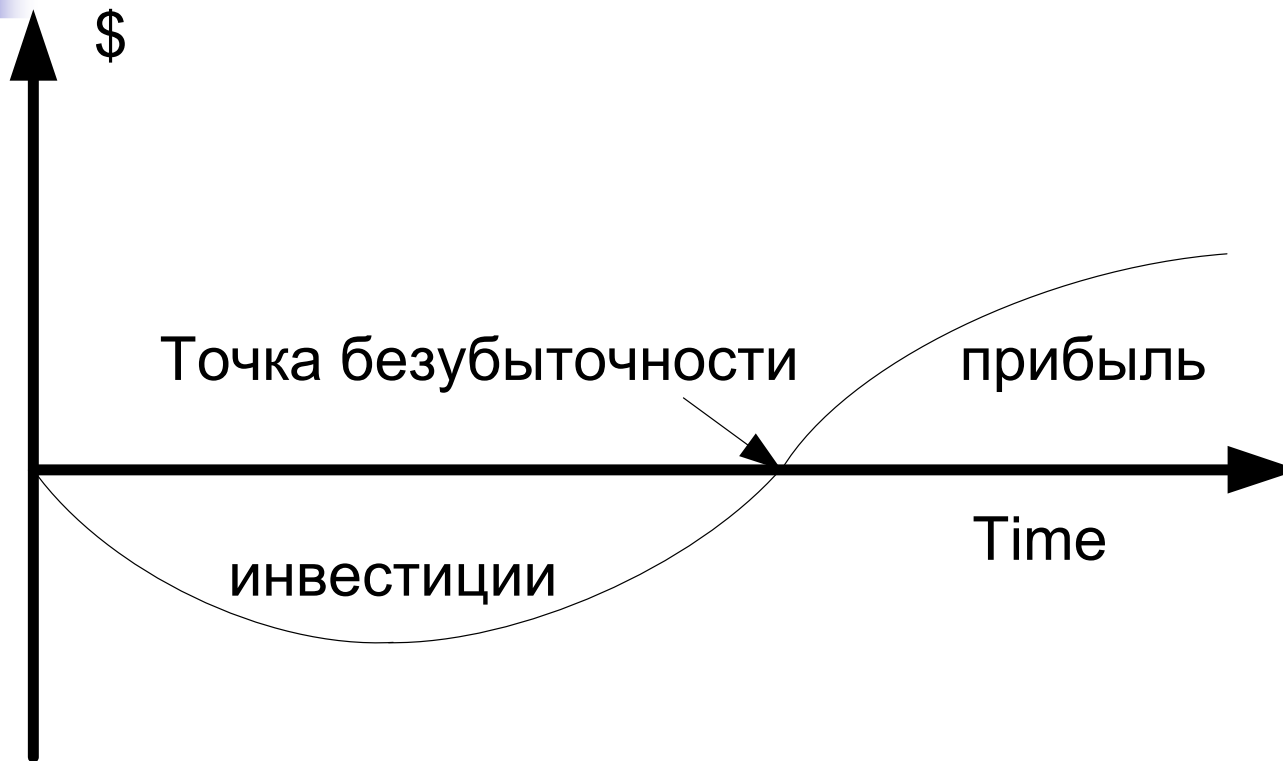


# Период окупаемости

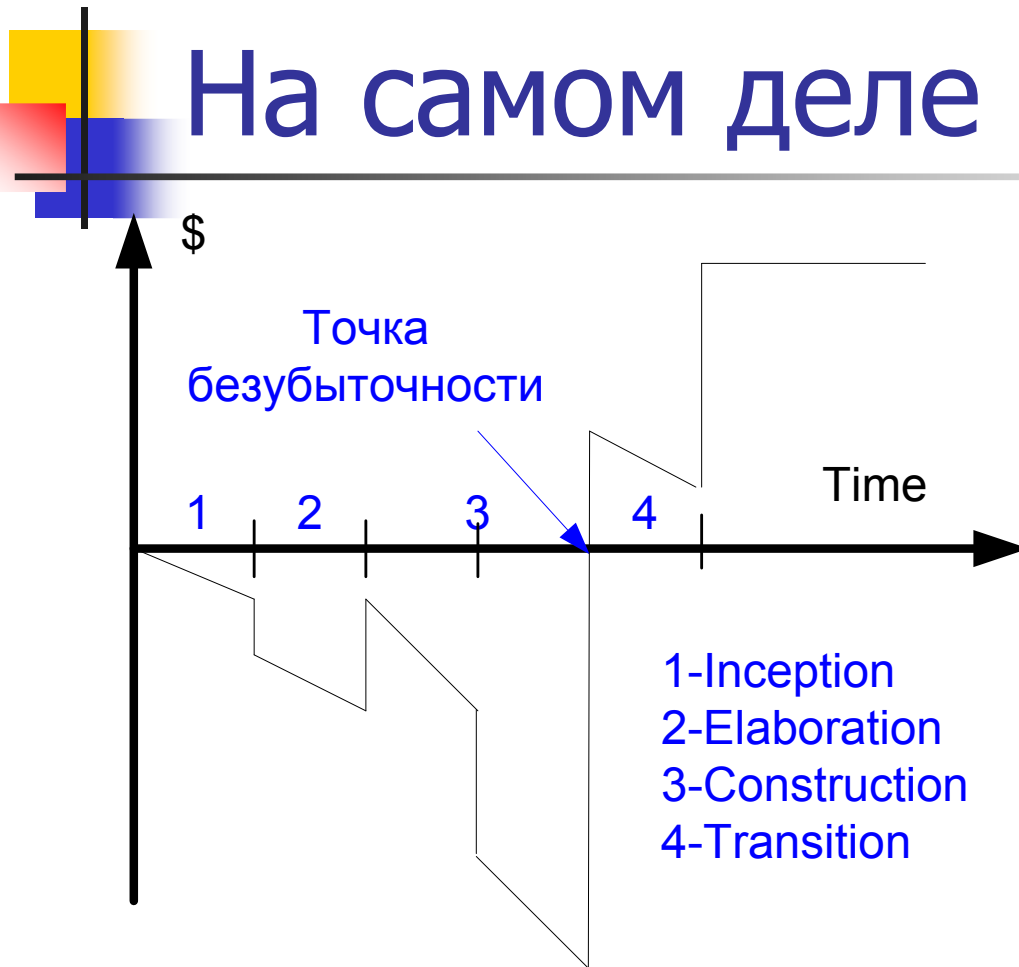
---

- Payback period – период времени до момента, когда чистый поток денежных средств станет положительным
- Обычно используется как грубый способ обоснования и ранжирования проектов
- Обычно учитывает только основные затраты

# Поток денежных средств



# На самом деле ...



**1 деление = 1 месяц**

**1** – постоянные расходы, разовые выплаты

**2** – постоянные расходы, разовые выплаты, получение оплаты

**3** – постоянные расходы, разовые выплаты, получение оплаты

**4** - постоянные расходы, разовые выплаты, получение оплаты

Инвестиции могут быть весьма значительными



# Методы исправления ситуации

---

- Кредиты
- Оптимизация расходов
- Предоплата
- Дробление проекта на фазы с проведением приемки-передачи (Acceptance) и полной оплатой фаз



# Пример

---

- Вкладываем в проект по \$1000 в течение 3 лет
- Получаем оплату по \$700 в год, плюс
  - За второй год \$100
  - За третий \$900
- Определите прибыль и точку безубыточности
- Прибыль =  $700 \cdot 3 + 100 + 900 - 3000 = 100$   
(но этот ответ не совсем верен)



## Почему ответ неверен?

---

- Во что мне обойдутся сегодня \$100, которые я отдам через 2 года?
- Или... Сколько надо иметь сегодня, чтобы через 2 года иметь \$100 ?
- Или... Что я мог бы получить за те \$1000, которые вложил в первый год проекта из нашего примера? Только ли \$100 через 2 года?



# Приведенная стоимость / Present Value

---

- Время = тоже деньги
- Present Value – стоимость *сегодня* денег, получаемых или выплачиваемых *завтра*
- Допустим, ставка рефинансирования ЦБ РФ по \$ = 10% годовых
- $\$100/1.1/1.1 = \$82.64$
- $\$82.64 = PV \$100$  через 2 года



# Поток денежных средств

---

- Cash Flow - показывает приток и отток наличности во времени
- Представлен в виде *приведенной стоимости* денег (дисконтирование)
- Приведенная стоимость (t) = стоимость \*  
учетная ставка (t)
- Идея: Сравнивать сравнимые величины, то есть стоимость денег, приведенную к одному моменту времени



# ROI – возврат инвестиций

---

- ROI – Return On Investment
- $ROI = (\text{приток-отток}) / \text{отток денежных средств}$
- Discounted ROI – ROI с учетом дисконтирования



# IRR - Internal Rate of Return

---

- Внутренний уровень возврата инвестиций
- Отвечает на вопрос: какой уровень дисконтирования даст нулевой чистый поток денежных средств через период (напр. 10 лет)
- Используется для сравнения инвестиционной привлекательности проектов



# Пример

---

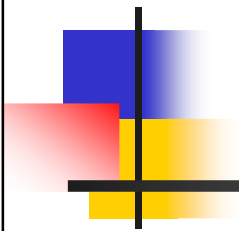
- Вкладываем в проект по \$1000 в течение 3 лет
- Получаем оплату по \$700 в год, плюс
  - За второй год \$100
  - За третий \$900
- Определите прибыль, точку безубыточности, ROI, IRR при ставке 13%
- Какую надо получить предоплату, чтобы получить нулевую прибыль?



# Пример

Flow	Discount rate	Present Value
-\$ 300	1.00	-\$ 300
-\$ 200	0.85	-\$ 169.49
\$ 600	0.72	\$ 430.91
\$ 100		-\$ 38.58

# Контракты и Поставки



Контракт, специфика контракта в ИТ, жизненный цикл контракта, авторские права  
Типы контрактов; риски покупателя и продавца  
Поставки; общие, форвардные и разобщенные  
Делать или купить – критерии выбора



# Контракт (договор) - это

---

- Соглашение между компетентными сторонами
- Для достижения законных целей
- С четкими условиями
- С употреблением действительных обменных единиц

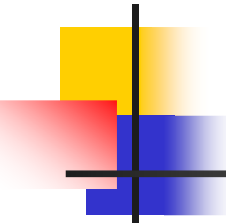
Источник: РМВОК



# Примеры контрактов в ИТ

---

- Контракт на разработку ПО
- Передача части проекта с субподряд
- Аренда помещений
- Аренда техники
- Подключение к Интернет
- Внешний хостинг
- .....



# Специфика контрактов на разработку ПО

---

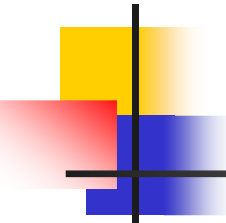
- Трудность определения цен
  - Уникальность услуги
  - Отсутствие «рыночной цены товара»
- Трудность фиксации соответствия объекта поставки спецификациям
  - Частично разрешается SRS, Test Plan, Acceptance Plan
- Лицензии на компоненты
- Авторское право
  - Требуется согласования процедуры передачи прав



# Авторские права

---

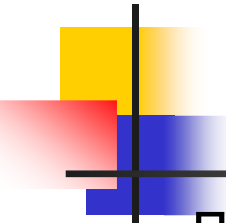
- Неимущественные права
  - Право на имя (неотторжимо)
- Имущественные права (права на воспроизведение, использование, передачу, модификацию)
  - Исключительные
  - Неисключительные



# Законодательство РФ об авторском праве

---

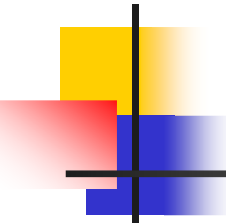
- Гражданский кодекс РФ, часть 4я
- Охрана прав разработчика и заказчика ПО
- ПО охраняется как литературное произведение



# Типичное содержание контракта на разработку ПО

---

- Предмет договора
  - Обычно ссылается на техническое предложение
- Цена и порядок расчетов
  - Фиксирует цену и моменты оплаты с привязкой к поставкам
- Права и обязанности сторон
  - Отчетность, привлечение 3-х лиц, ...
- Ответственность сторон
  - Штрафы, пени ...
- Авторские правоотношения
  - Порядок передачи имущественных прав
- Приемка-передача ПО
  - Порядок и условия подписания акта приемки-передачи (Acceptance procedure)
- Действие договора
  - Порядок вступления в силу и порядок расторжения



# Жизненный цикл контракта на разработку ПО

---

- Техническое предложение
- Согласование и подписание контракта
- Выполнение задач проектного плана
- Передача объектов поставки
- Приемочное тестирование
- Подписание акта приемки передачи
- Передача исходных текстов
- Выставление счета (invoice)
- Получение оплаты



# Типы контрактов

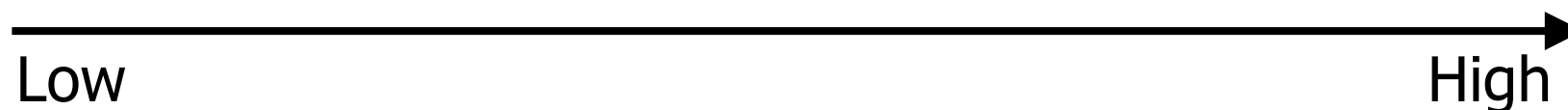
---

- Fixed Price (FP)
  - Firm Fixed Price
  - FP + Economic Adjustment
  - FP + Incentive Fee
- Time & Material (T&M)
  - Cost + Fixed Fee
  - Cost + Incentive Fee
- Cost reimbursable (CPFF, CPIF)



# Риски заказчика и исполнителя

Заказчик



FFP	FP+I; FP+A	TM+A	TM+I	TM+F
-----	------------	------	------	------



Исполнитель



# Сделать или купить?

---

- Наличие свободных ресурсов
- Наличие/отсутствие навыков
- Необходимость в спец. оборудовании
- Желание контролировать процесс
- Секретность проекта



## 3 стратегии закупок

---

- Форвардные (Forward Buying)
- Общие (Blanket Orders)
- Разобщенные (Splitting Orders)



# Forward Buying

---

- Стратегия «закупать в избытке по отношению к требуемому минимуму»
- Предохраняет от нехватки
- Скидки на количество
- Защита от роста цен
- «-» замораживание средств в пассивах
- Пример в IT: лицензии



# Blanket Orders

---

- Долгосрочный заказ вперед
- «+» Скидка на количество
- «-» Заказанный ресурс может не потребоваться в полном объеме
- Пример в IT: «аренда» команды разработчиков (outsourced IT department)



# Splitting Orders

---

- Проекту требуются некие критические компоненты
- Известно, что поставщик имеет 90% вероятность поставить компонент вовремя
- Что даст нам выбор еще одного (второго) поставщика, который также имеет 90% вероятность своевременной поставки?
- «+» 99%
- «-» более высокая цена

# Управление проектной конфигурацией



---

СММІ об управленіи конфигурацией  
Управление требованиями: Источники  
неприятностей (обзоры Standish group), US AirForce



# Управление конфигурацией

---

- Что такое конфигурация? Почти все результаты работ (work products)
- Требования
- Архитектурные решения
- Исходные тексты, используемые библиотеки, их версии
- Средства разработки и их настройка
- Объекты поставок, документация



Ниже приведены основные постулаты конфигурационного управления по СММ (дословный перевод требований):

любые действия по направлению конфигурационного управления заранее запланированы;

любые программные работы идентифицированы, управляются и являются общедоступными;

любые изменения в продукте являются управляемыми;  
заинтересованные группы и индивидуумы постоянно информируются о состоянии развития проекта.



# 1994 - Кризис Software Development

---

- 31% проектов остановлен до планового завершения
- 53% проектов стоили 189% своих оценок
- 9% завершены вовремя и в рамках бюджета (крупные компании)
- 16% завершены вовремя и в рамках бюджета (мелкие компании)

- Standish Group survey, 1994, (8000 projects)



# Источники проблем

---

- Недостаточное взаимодействие с пользователями - 12,8%
- Неполнота требований и спецификаций - 12,3%
- Изменение требований - 11.8 %
- Новизна технологии - 3.7%  
(для сравнения)

- Standish Group survey, 1994, (8000 projects)



# US Air Force project

---

## Источники проблем

- **Требования** - **41%**
- **Архитектура** - **28%**
- **Ошибки в данных** - **6%**
- **UI** - **6%**
- **Системное окружение** - **5%**
- **Ошибки людей** - **5%**
- **Документация** - **2%**
- **Другие** - **7%**

- Sheldon F., "Reliability Measurement from Theory to Practice", 1992

# Стоимость исправления ошибок в требованиях по фазам

• Inception phase	- 0,1 – 0,2
• Design	- 0,5
• Coding	- 1
• Unit Testing	- 2
• Acceptance Test	- 5
• Maintenance	- 20



- Davis A., “Software Requirements – Objects, Functions and States”, 1993



# Где зарыта собака?

---



- Требования обсуждаются вербально
- Документируются на естественном языке
- Сложно разделить «изменение» и «уточнение» требований
- Неоднозначности трактуются в пользу клиента



# Управление требованиями

---

- Систематический подход к выявлению, структуризации и документированию требований
- Четкий процесс изменения требований к программному решению



# Управление требованиями

---

- Документирование требований
  - SRS – главный документ; описание и приоритеты требований
  - функциональные, нефункциональные требования и требования к процедуре приемки
  - прототипы, их утверждение заказчиком
- Контроль требований и их изменений
  - статус по требованиям ( (НЕ)реализовано /изменено)
  - база данных изменений
  - стоимость каждого изменения определяется и утверждается ДО того, как изменение принято к исполнению
- Метризация процесса управления требованиями
  - количество изменений
  - количество изменений, (НЕ)включенных в SRS



# Управление изменениями

---

Хороший Менеджер может выиграть тендер за

**\$ 0,00**

и выполнить проект в точном соответствии с контрактом.

(шутка с долей шутки)



# Управление IT проектами

---

## Управление ресурсами. Модели организаций

Проектная, Функциональная, Матричная  
модели.

Преимущества и недостатки



# Модели организаций

---

- Проектная
- Функциональная
- Матричная

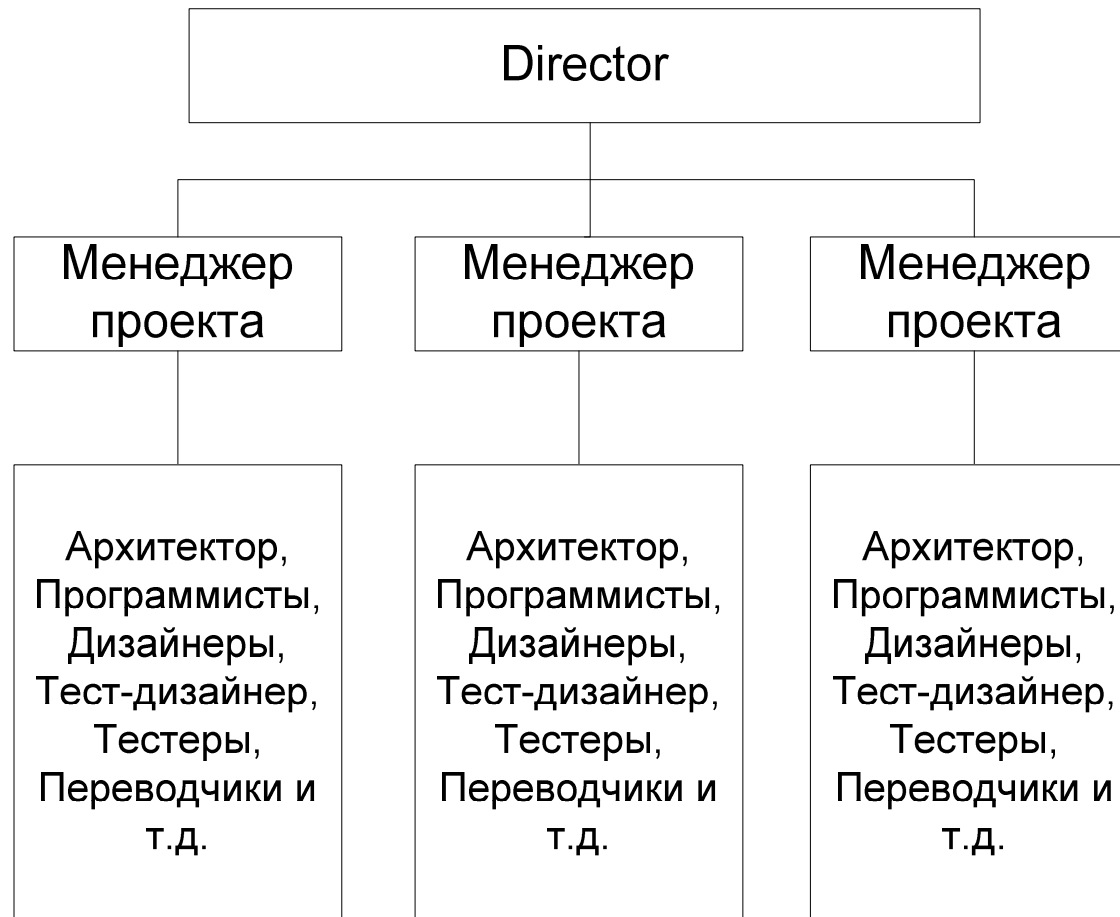


# Проектная организация

---

- Полная власть проектного менеджера
- Ресурсы распределены по проектам
- На период проекта – высокая мотивация
- Что будет с людьми по окончании проекта?
- Обычно невелика по размеру
- Сложно обеспечивать общие процессы, такие как контроль качества, аттестацию и развитие персонала

# Пример структуры проектной организации



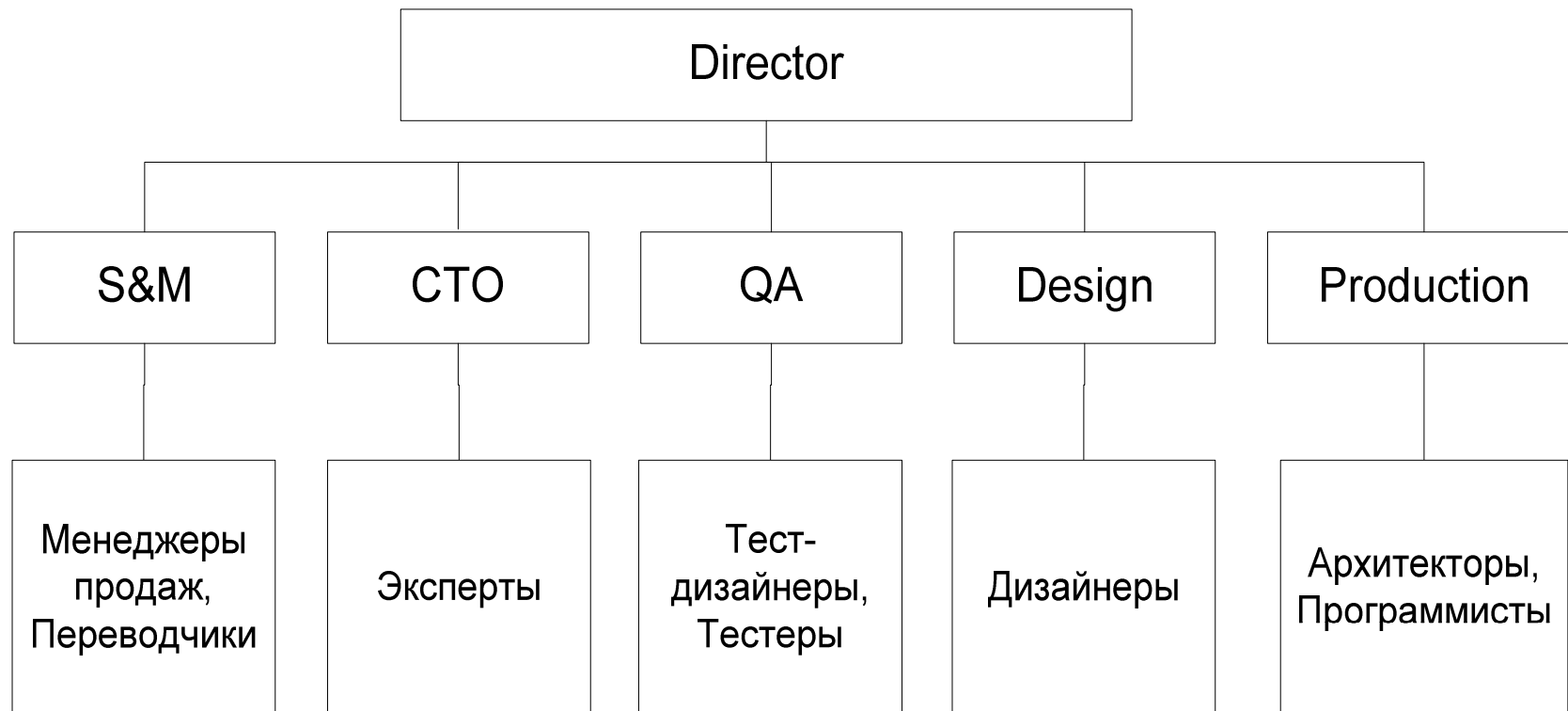


# Функциональная организация

- Фокусировка на навыках (обыкновенно - по отделам)
- Функциональный менеджер хорошо знает людей
- Очень стабильная организация
- Один большой босс, много функциональных менеджеров
- Недостаточно внимания на конкретные проекты
- Затруднен переход из проекта в проект, ресурсы распределены скорее по функциональным отделам чем по проектам
- Свойственна большим организациям



# Пример структуры функциональной организации



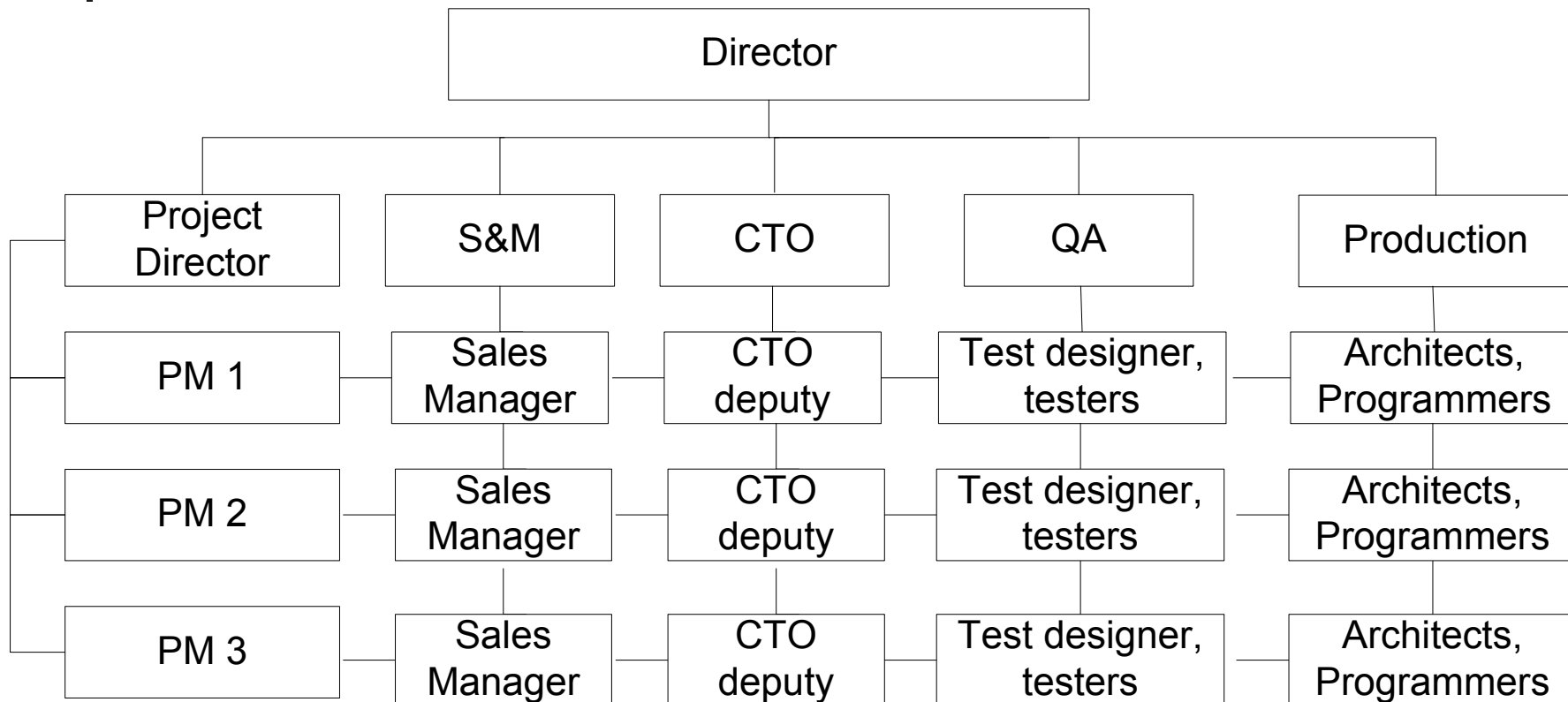


# Матричная организация

---

- Характеристики проектной и функциональной одновременно
- Общие ресурсы, люди входят и выходят из проектов легко и по необходимости
- Много менеджеров проектов
- Сложные коммуникации
- Проектно-гибкая структура, эффективно поддерживает много проектов
- Необходимо иметь больше менеджерских кадров
- Высокая проектная мотивация сотрудников

# Пример матричной структуры





# Матричная структура предполагает

---

- Использование средств многопроектного управления (MS Project Server, OpenProject)
- Наличие системы распределения человеческих ресурсов



# Орг. структура и проект

<b>Тип организации Характеристика</b>	<b>Функциональная</b>	<b>Матричная</b>	<b>Проектная</b>
<b>Ответственность менеджера</b>	Мала	Средняя- Высокая	Полная
<b>Занятость РМ в проекте</b>	Частичная	Полная	Полная
<b>Процент персонала проекта, занятого в нем на 100%</b>	Менее 50%	50-95%	85-100%

✓ Не стоит принимать цифры как абсолютные, они дают лишь качественную картину