

# Управление рисками в программном проекте

С. Архипенков

# Об авторе

- Сергей Архипенков, РМР РМІ.
- Стаж в разработке ПО более 30 лет.
- Автор книг, статей, учебных курсов.
- [www.arkhipenkov.ru](http://www.arkhipenkov.ru)
- [sergey@arkhipenkov.ru](mailto:sergey@arkhipenkov.ru)

## Коротко о главном

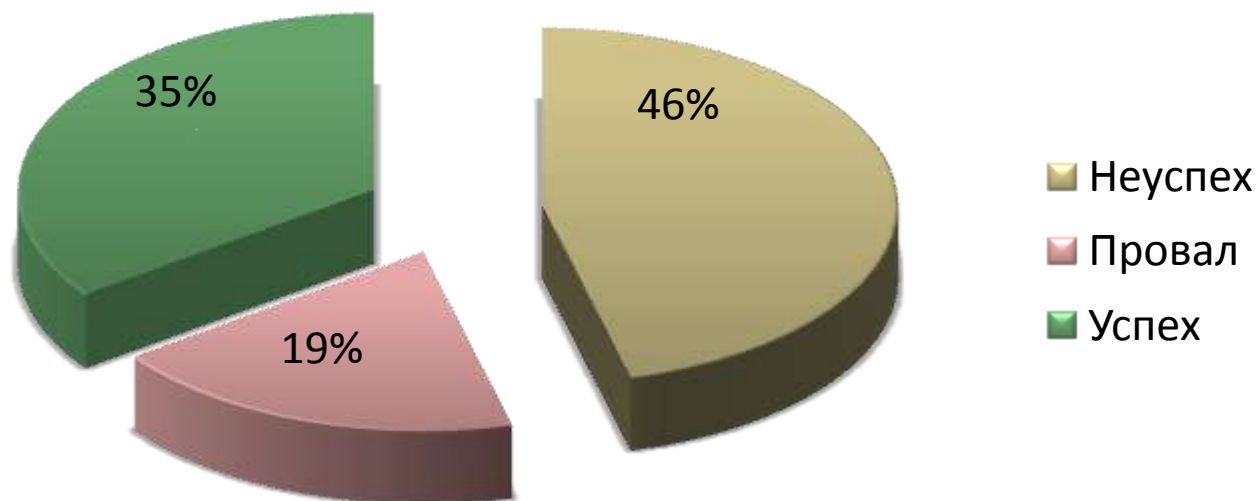
Риски и проекты

Планирование управления рисками

Главные риски программных проектов

Мониторинг и управление рисками

# Статистика производства ПО



*Источник: "The Chaos Report" The Standish Group, 2008*

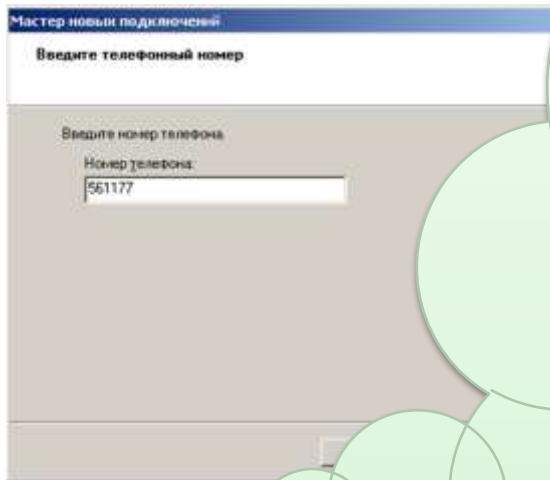
# Проект без риска – удел неудачников

- Риски и выгода всегда ходят рука об руку.
- *Риск* - неопределенное событие или условие, наступление которого отрицательно или положительно сказывается на целях проекта.
- Риск это проблема, которая еще не возникла, а проблема – это риск, который материализовался.
- Риск характеризуется вероятностью наступления и последствиями воздействия на проект. Пример Майка Ньюэлла.



# Откуда берется неопределенность?

Пример. Форма ввода телефонного номера



Мастер новых подключений

Введите телефонный номер

Введите номер телефона

Номер телефона:

Может ли вводиться несколько номеров?  
Должна ли быть проверка номеров на действительность?  
Простая или сложная проверка?  
Если реализуем простую проверку, то не захочет ли клиент заменить ее на более сложную?  
Должна ли проверка работать для иностранных номеров?  
Можно ли воспользоваться готовым решением?  
Каково должно быть качество реализации?  
Вероятность ошибки после поставки?  
Сколько времени потребуется на реализацию и отладку? (зависит от конкретного исполнителя).



# Управлять или не управлять?

## Против

## За

Если сказать правду заказчик испугается и откажется от проекта..

Заказчик не глуп, он поймет, что ему лгут.

Уровень неопределенности слишком велик для того чтобы его показывать целиком.

Высокая точность оценок в IT-проектах может вызвать только сомнения в профессионализме.

Заданный диапазон неопределенности оправдывает плохую работу.

Сотрудников надо мотивировать на достижение лучших результатов. Но надо быть готовым и к худшему.

Управление должно быть нацелено только на успех.

Это значит поставить проект в зависимости только от удачи.

Проект новый и у нас нет данных о рисках.

Большинство значимых рисков IT-проектов известны.

Трудно быть реалистом, когда вокруг одни оптимисты.

Нет контраргументов ☹️

# Категории рисков

- Известные неизвестные
- Неизвестные неизвестные



# Характеристики риска

- Причина или источник.
- Симптомы риска.
- Последствия риска.
- Влияние риска.



# Пример характеристик риска



# Планирование управления рисками

Планировать надо для того, чтобы:

- выделить достаточное количество времени и ресурсов для выполнения операций по управлению рисками,
- определить общее основание для оценки рисков,
- повышает вероятность успешного достижения результатов проекта.



# Исходные данные для планирования



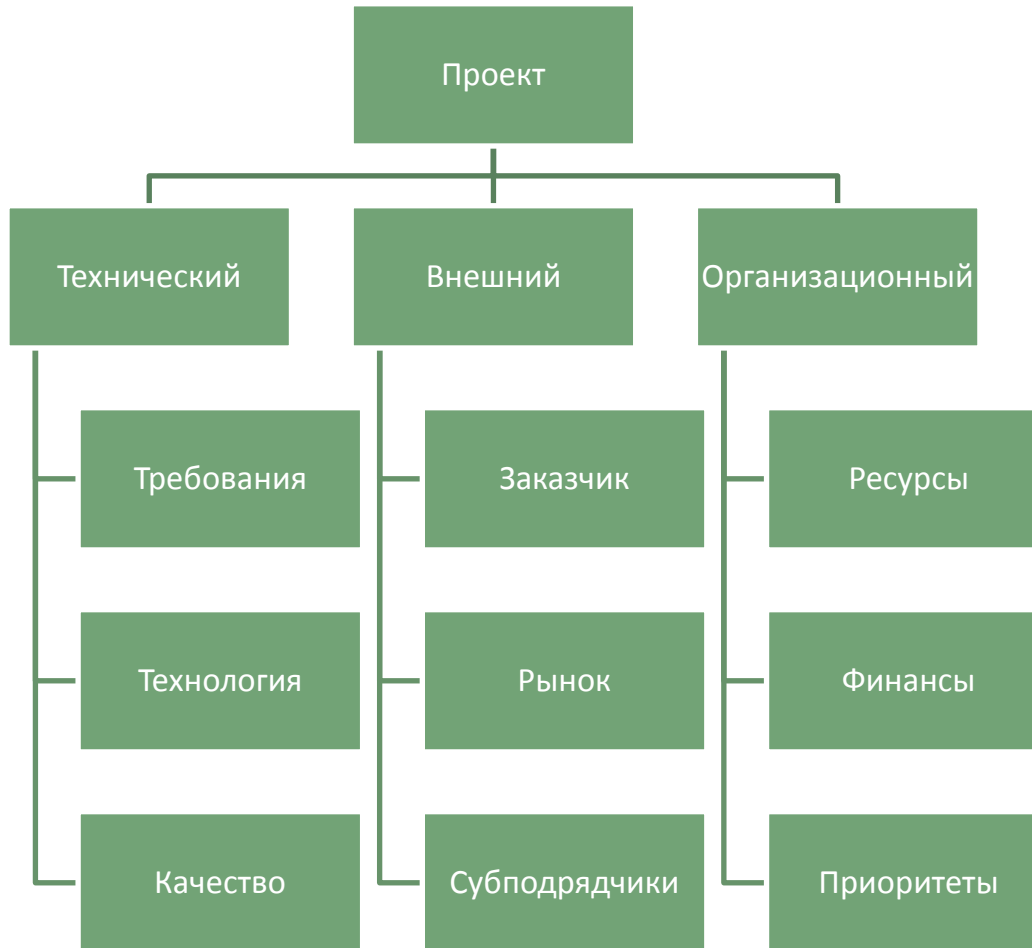
# План управления рисками

План управления рисками включает в себя следующие элементы:

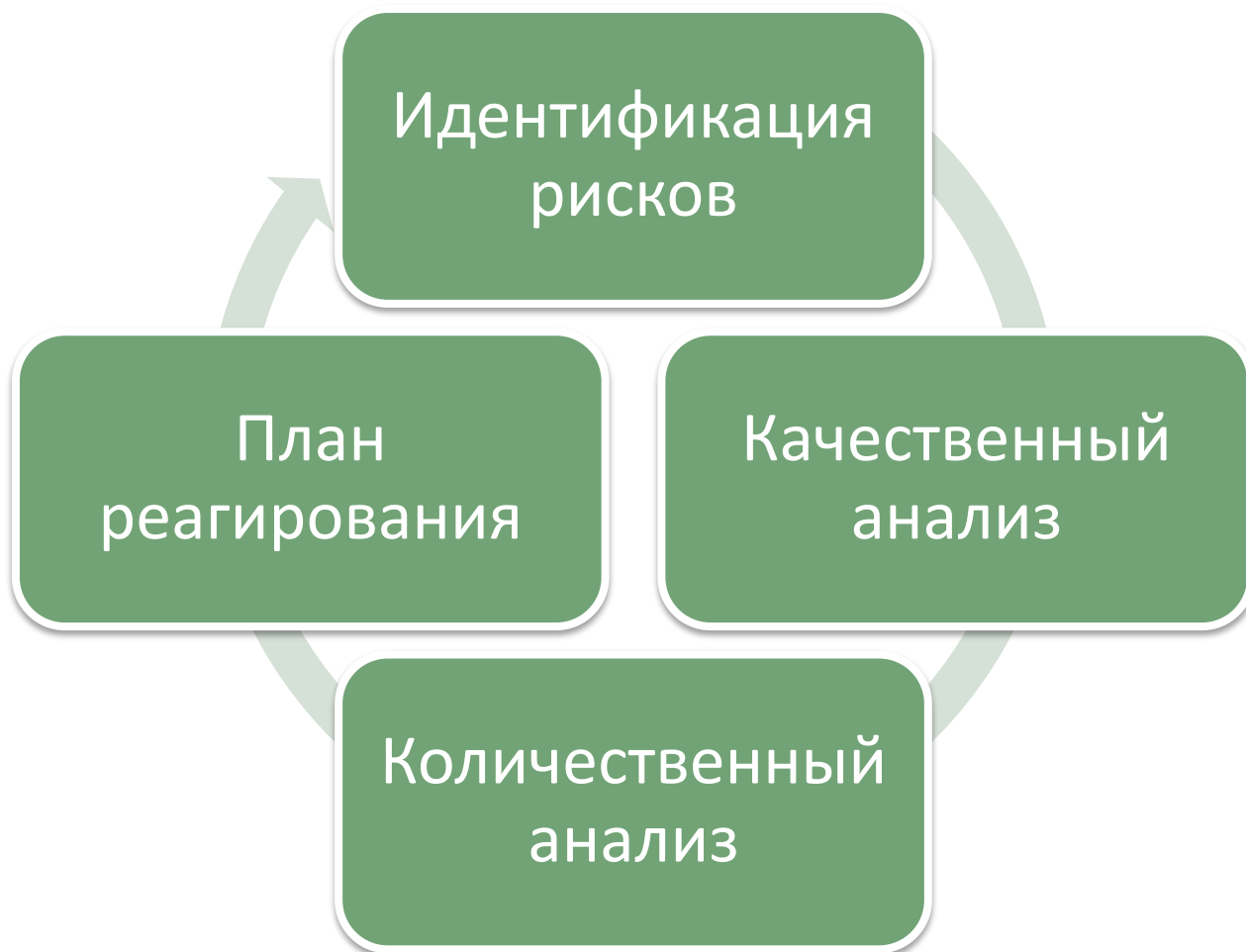
- Категории рисков.
- Определение подходов, инструментов и источников данных.
- Распределение ролей и ответственности.
- Выделение ресурсов и оценка стоимости мероприятий.
- Определение сроков и частоты выполнения процесса управления рисками.
- Общие подходы для определения уровней вероятности, шкалы воздействия и близости рисков на проект.



# Категории рисков



# Планирование управления рисками



# Исходные данные для идентификации рисков

- Информация из открытых источников
- Информация о выполнении прежних проектов.
- Допущения проекта, сделанные в описании содержания проекта.



# Инструменты и методы идентификации

Анализ документации.

Методы сбора информации.

Анализ контрольных списков.

Анализ допущений.

Анализ с использованием  
диаграмм.



# Методы сбора информации

Мозговой штурм

Метод Дельфи

Опрос экспертов

Карточки Кроуфорда

SWOT-анализ



# Пример списка выявленных рисков

Первопричина	Симптомы	Последствие	Приносимый ущерб
Нехватка кадров	Могут быть объединены роли разработчиков и тестировщиков	В программном продукте будет содержаться больше ошибок	Заказчик будет менее доволен результатом
Изменения в технологии	Разработчикам придется использовать новый язык программирования	Увеличится время, затрачиваемое на разработку	Наш продукт будет представлен на рынке в более поздние сроки, что приведет к захвату части рынка конкурентами
Организация работы	Часть группы разработчиков находится в Лондоне, а часть – в Лос-Анджелесе	Обмен информацией внутри группы затрудняется	Задержки в сроках сдачи готового продукта и дополнительные трудозатраты

# Шкала оценки вероятности рисков

Вес	Значение	Критерий
3	Очень вероятно	Шансы наступления весьма велики
2	Возможно	Шансы равны
1	Мало вероятно	Наступление события весьма сомнительно

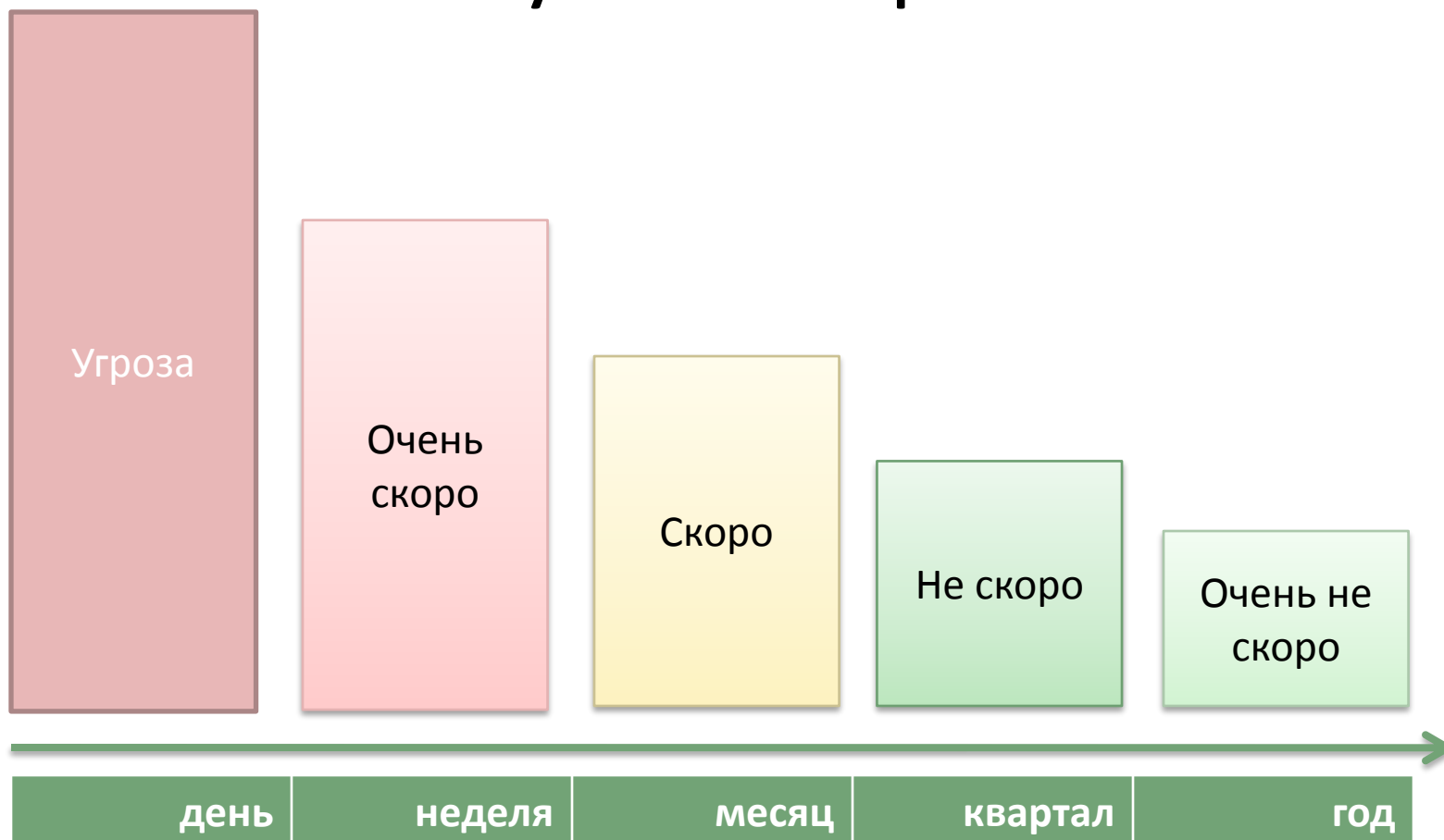


# Шкала оценки воздействия рисков

Вес	Значение	Критерий
3	Катастрофические	Потери более \$50К
2	Критические	Потери от \$10К до \$50К
1	Умеренные	Потери до \$10К



# Пример шкалы близости наступления риска



# Качественный анализ рисков

Качественный анализ рисков включает:

- Определение вероятности реализации рисков.
- Определение тяжести последствий реализации рисков.
- Определения ранга риска по матрице «вероятность - последствия».
- Определение близости наступления риска.
- Оценка качества данных.

Матрица вероятности и последствий

		Вероятность		
		1	2	3
Воздействие	3	3	6	9
	2	2	4	6
	1	1	2	3

Ранг риска

- высокий
- средний
- низкий

# Оценка качества данных

- Для оценки рисков необходима точная и адекватная информация.
- Использование неточной информации ведет к неверной оценке рисков.
- Критерии оценки точности данных:
  - Степень понимания риска.
  - Доступность и полнота информации о риске.
  - Надежность, целостность и достоверность данных.
- Неверная оценка риска также является риском.



# Пример описания риска

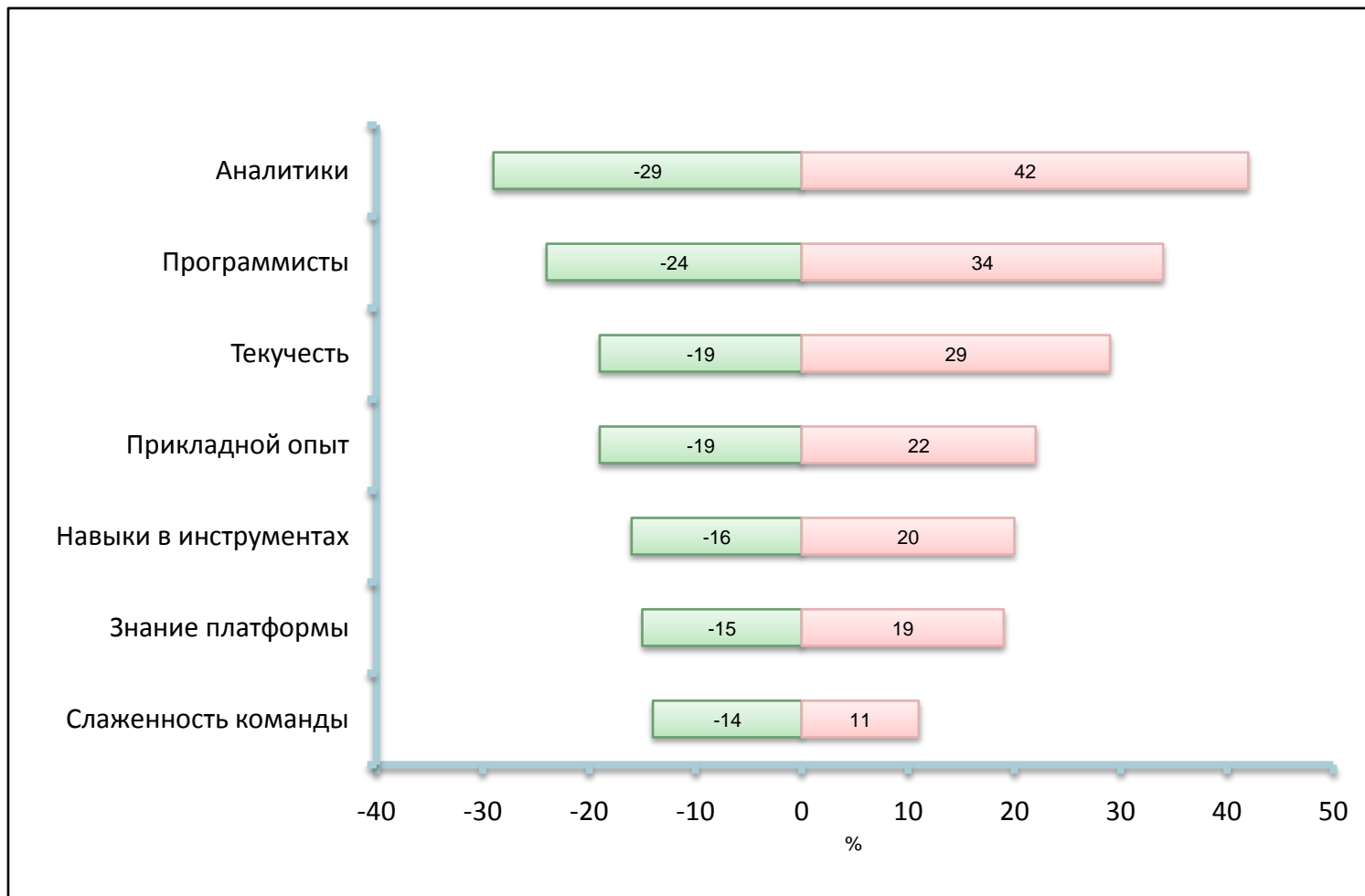
Номер: R-101	Категория: Технологический.
Причина: Быстродействие не может быть обеспечено на опробованных технологиях.	Симптомы: Разработчики будут использовать новый язык Erlang.
Последствия: Низкая производительность разработки	Воздействие: Увеличение сроков и трудоемкости разработки.
Вероятность: Очень вероятно.	Степень воздействия: Умеренная.
Близость: Очень скоро.	Ранг: 3.
Исходные данные: «Содержание проекта», «План обеспечения ресурсами», Протоколы совещаний №21 от 01.03.2009, №27 от 25.03.2009.	

# Количественный анализ рисков

- Методы количественного анализа рисков и моделирования:
  - Анализ чувствительности.
  - Анализ дерева решений.
  - Моделирование и имитация.



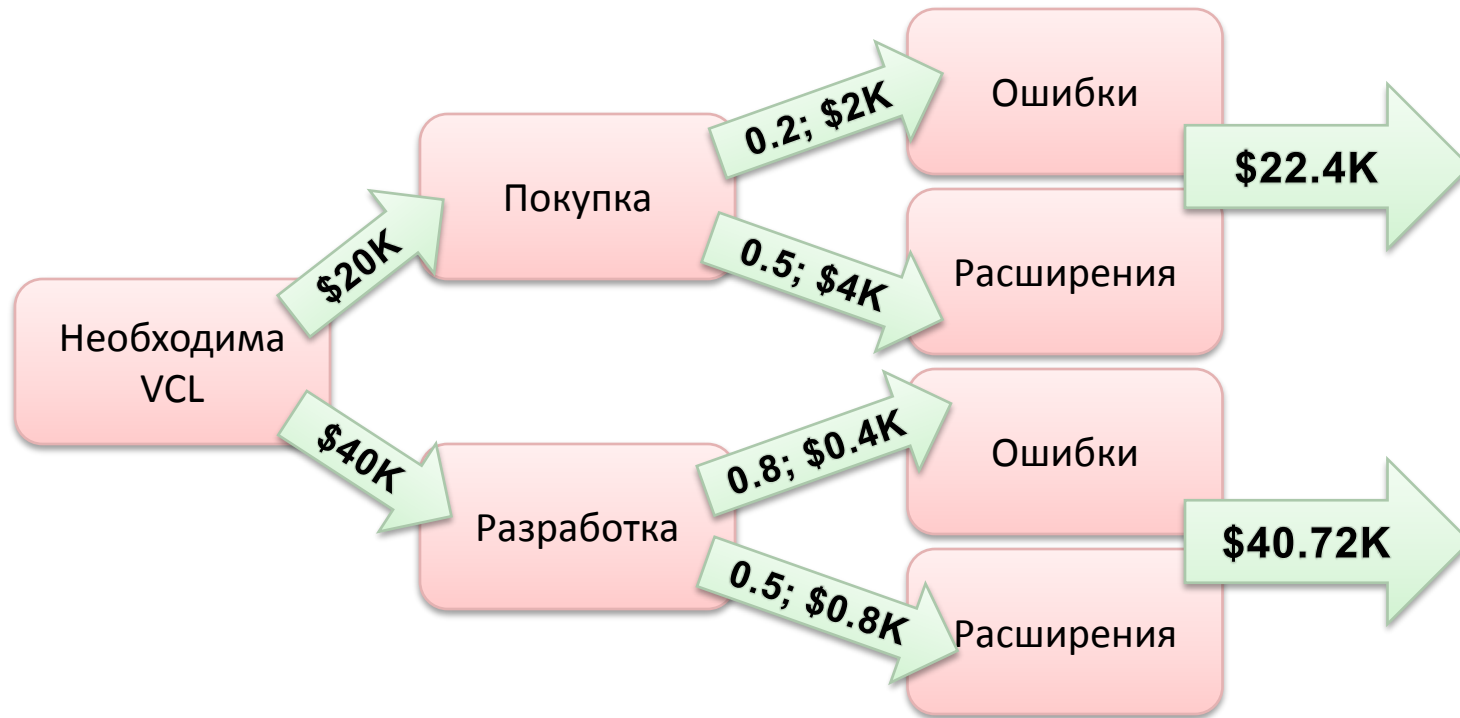
# Анализ чувствительности (диаграмма торнадо)



*(с) СОСОМО II. Суммарный эффект способен изменить оценку в **22 раза***

# Анализ дерева решений

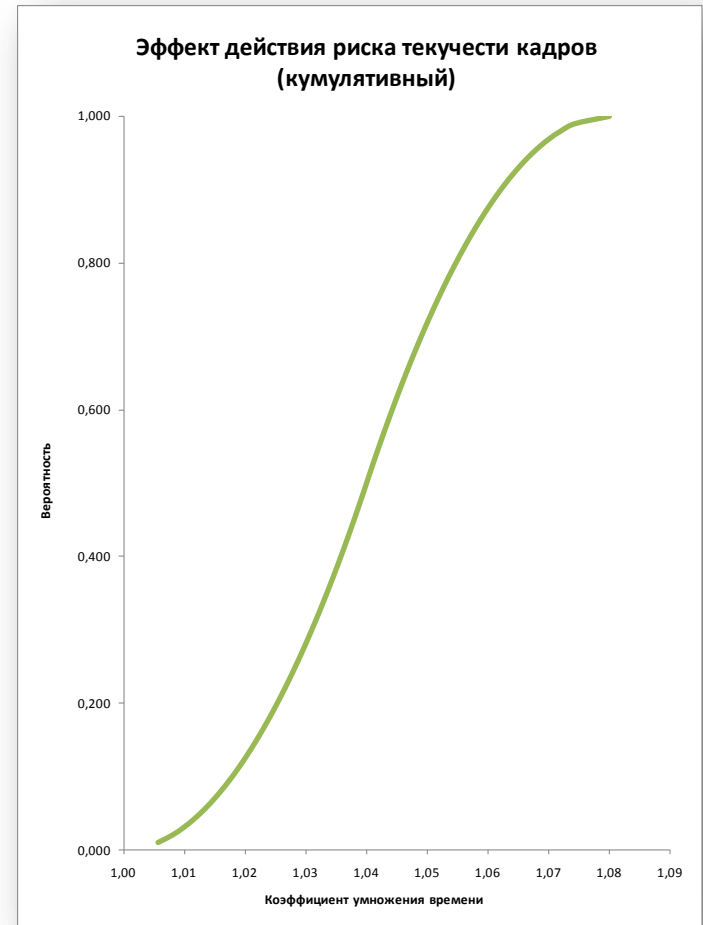
Проводится на основе анализа диаграммы дерева решений, которая описывает рассматриваемую ситуацию с учетом каждой из имеющихся возможностей выбора и возможного сценария.



# Моделирование

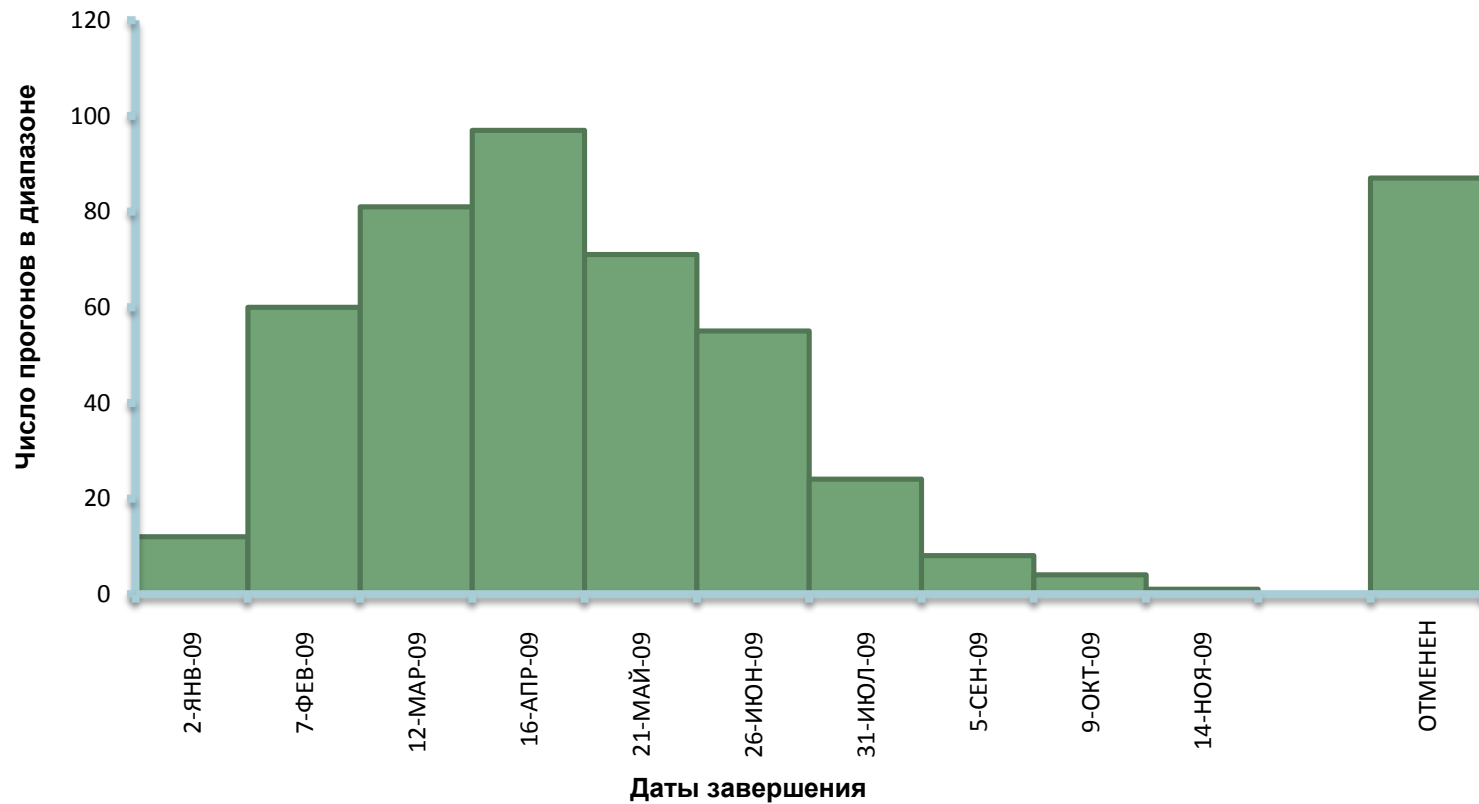
Моделирование обычно проводится с помощью метода Монте-Карло.

Пример применения метода Монте-Карло:  
[www.pmo.ru/riskology](http://www.pmo.ru/riskology). ©  
2005 Том ДеМарко, Тимоти Листер.



# Результаты моделирования

"Суперпроект": Моделирование проекта (500 прогонов)



# Планирование реагирования на риски

## Методы

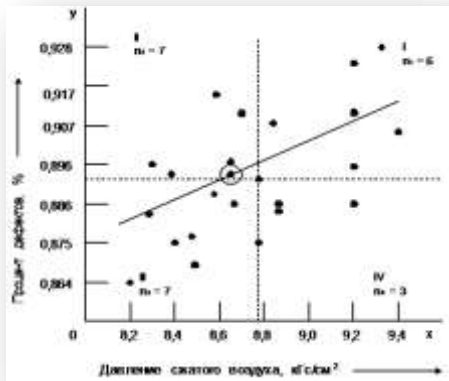
- Уклонение от риска (Risk Avoidance).
- Передача риска (Risk Transference).
- Снижение рисков (Risk Mitigation).
- Принятие риска (Risk Acceptance).

*Важно помнить о **вторичных рисках** (Secondary Risks), которые возникают в результате применения реагирования на риски.*



# Главные причины провалов проектов

- Требования заказчика отсутствуют / не полны / подвержены частым изменениям.
- Отсутствие рабочего взаимодействия с заказчиком.
- Отсутствие необходимых ресурсов и опыта.
- Неполнота планирования. «Забутые работы».
- Ошибки в оценках трудоемкостей и сроков работ.



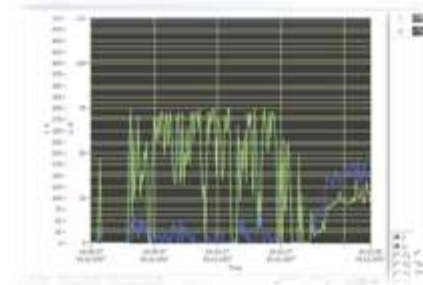
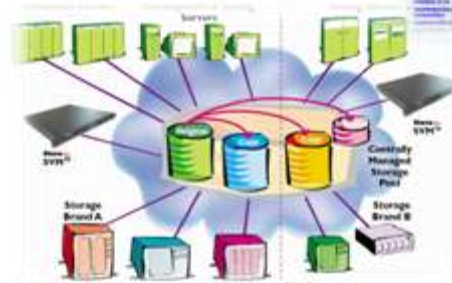
# Неполнота требований

## Функциональные:

- Программы установки, настройки, конфигурации.
- Миграция данных.
- Интерфейсы с внешними системами.
- Справочная система.

## Нефункциональные:

- Производительность.
- Надежность.
- Открытость.
- Масштабируемость.
- Безопасность.
- Портитруемость.
- Эргономичность.



# Требования подвержены частым изменениям

Способы реагирования:

- Переоценка проекта каждый раз, когда требования добавляются / изменяются (уклонение).
- Итерационная разработка. Контракт на основе «Time&Materials» (передача риска Заказчику).
- Учет (резервирование) в оценках трудоемкости и сроков возможности роста требований, например, на 50% (резервирование риска).

*Принцип минимализма. «Рассказ закончен не тогда, когда в него нечего добавить, а тогда, когда из него нечего больше выкинуть». Вольтер.*

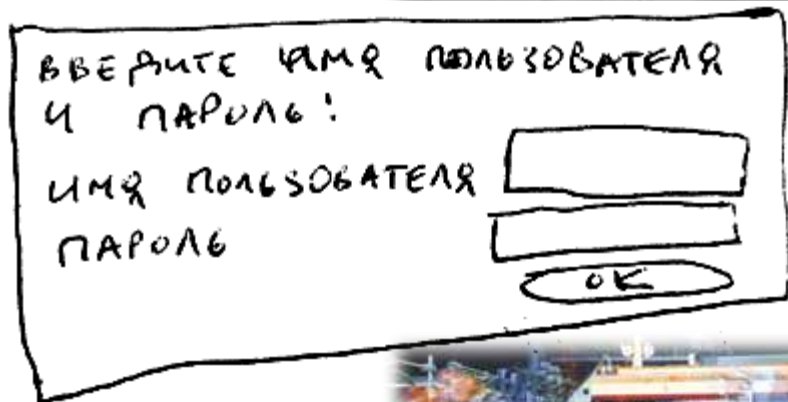




# Отсутствие рабочего взаимодействия с заказчиком

Способы реагирования (снижение):

- Открытость, установление доверительных отношений.
- Непрерывное взаимодействие.
- Прототипирование и согласование пользовательских интерфейсов.
- Периодические поставки тестовых версий конечным пользователям.



# Неполнота планирования

## Проектные:

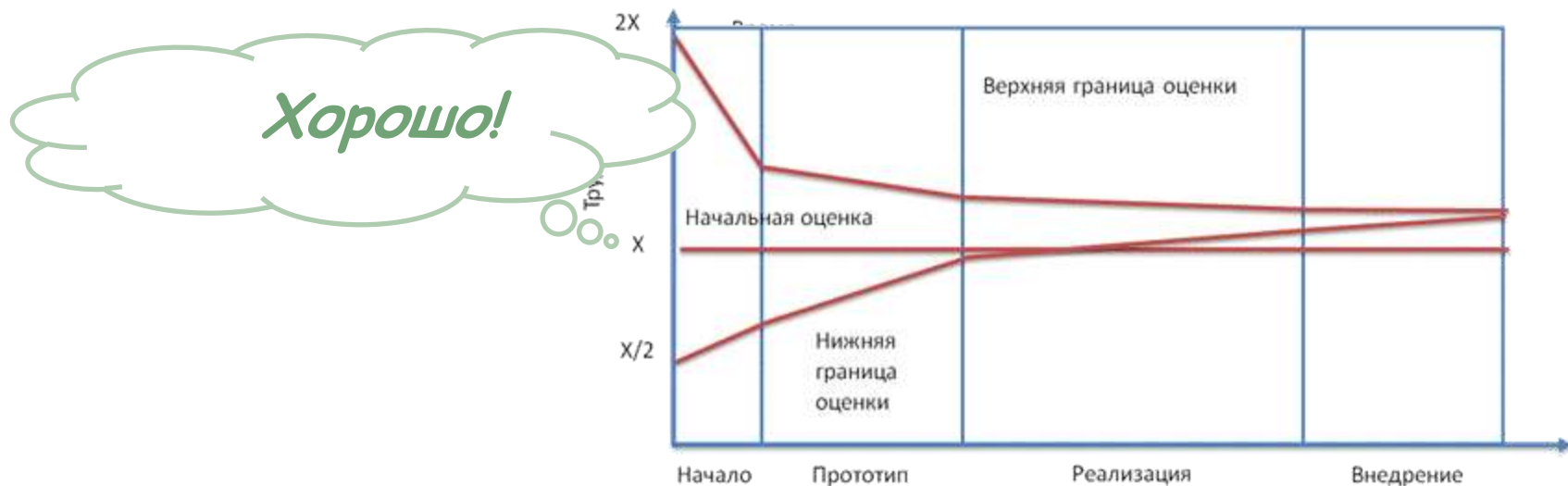
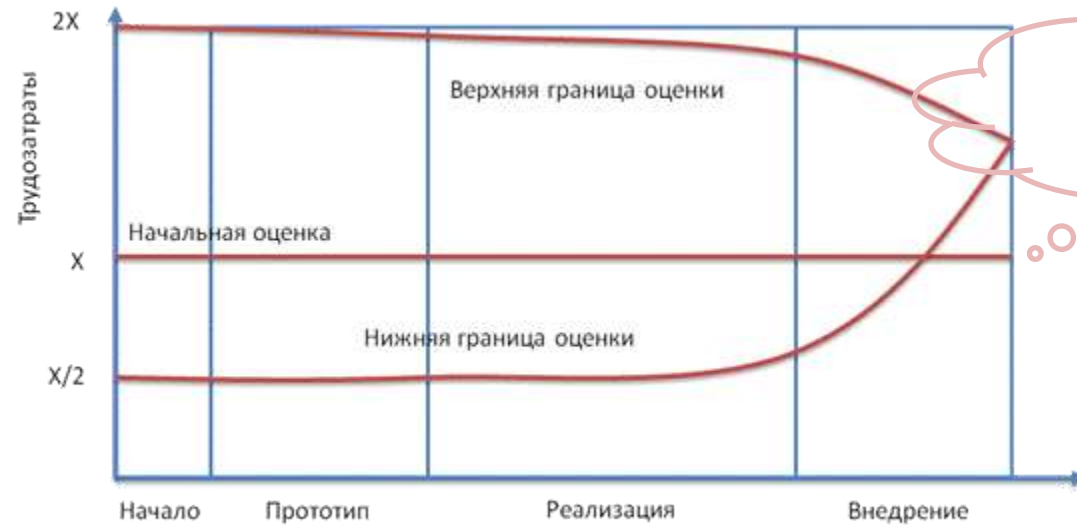
- Обучение.
- Координация.
- Уточнение требований.
- Управление конфигурациями.
- Поддержка автосборки.
- Разработка автотестов.
- Создание тестовых данных.
- Обработка запросов на изменения.

## Непроектные:

- Сопровождение действующих систем.
- Повышение квалификации.
- Участие в пресейл.
- Административная работа.
- Отпуска, праздники, больничные.



# Управление на снижение рисков



# Приоритеты требований



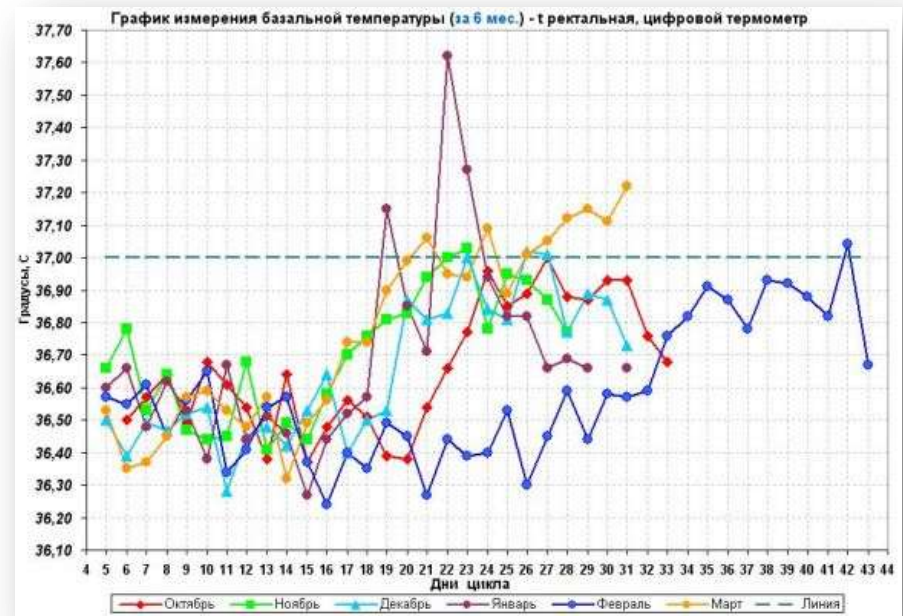
# Принципы количественного управления

- Измерения по каждому проекту выполняются периодически (не реже одного раза в 1-2 недели).
- Для каждого измеримого показателя должны быть определены его плановые значения.
- Для каждого планового значения должны быть определены три области критичности отклонений:
  - Допустимые отклонения.
  - Критичные отклонения.
  - Недопустимые отклонения.
- Все измерения сохраняются в репозитории проекта.



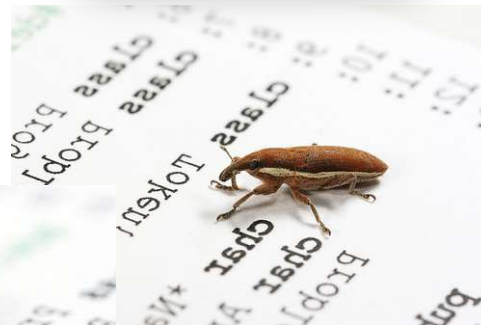
# Показатели управления проектом

- Прогресс проекта.
- Стабильность проекта.
- Текущий размер проекта.
- Производительность.



# Показатели качества программного продукта

- Дефектность продукта.
- Доля не устраненных дефектов.
- Качество проектирования.
- Документированность кода.



# Показатели производительности

		India	Japan	USA	Europe etc.	TOTAL
Projects		24	27	31	22	104
LOC/ Month	median	209	469 cf. 389 in 1990	270 cf. 245 in 1990	436	374
Bugs/ 1000 LOC	median	.263	.020 cf. .20 in 1990	.400 cf. .80 in 1990	.225	.150

13

*(c) Michael A. Cusumano, MIT Sloan School of Management, Software Engineering Conference in Russia, SEC(R) 2005*

# Управлять!

- Том Демарко: «Рассматривать только благоприятные сценарии и встраивать их в план проекта – настоящее ребячество. И все же мы постоянно так поступаем».
- Не управлять рисками в проекте – это все равно, что не иметь в кинотеатре огнетушителей и плана эвакуации.

*Если тех, кто говорит о возможных проблемах до открытия проекта, называют troublemakers, а тех кто сдает проект спустя 2 месяца после обещанного срока, работая при этом по 60-80 часов в неделю, – героями, то у вас плохая команда.*



# Источники и дополнительная литература

- «PMBOK. Руководство к Своду знаний по управлению проектами», 3-е изд., PMI, 2004.
- Том ДеМарко, Тимоти Листер, «Вальсируя с Медведями. Управление рисками в проектах по разработке программного обеспечения», М., Компания p.m.Office, 2005.
- С. Архипенков «Лекции по управлению программными проектами», М., 2009 (<http://www.arkhipenkov.ru>)

