

# Мастер-класс «Управление разработкой ПО»

---



## Сергей Архипенков

Эксперт в управлении программными проектами, PMP PMI. В разработке ПО более 30 лет. Создавал имитационные модели сложных космических систем в Центре управления полетами. Руководил коммерческой разработкой ПО и проектами организационного развития в компаниях PriceWaterhouseCoopers, Luxoft, CBOSS. Выполнял проекты по заказу Европейского космического агентства, «Даймлер-Бенц Аэроспейс», корпорации «Боинг», ЦБ РФ, ОАО «Газпром». Автор 5 книг, около 100 статей, докладов и учебных курсов по информационным технологиям и

управлению программными проектами. [Резюме >](#)

## От автора

За более чем тридцать лет разработки программного обеспечения и руководства программистами в ряде очень разных по своей культуре компаний у меня накопился определенный практический опыт, который позволил мне сформировать ясное видение того, как надо правильно управлять разработкой ПО, а как - не надо.

Затем потребовалась долгая работа, направленная на то, чтобы сформулировать свое видение на понятном для других людей языке и сделать его доступным для коллег. Для этого необходимо было найти проекцию своей картины мира на современные знания в областях управления проектами, психологии, социологии, социологии, сопоставить свои представления с точкой зрения других профессионалов в отрасли программной инженерии. Что из этого получилось, можно прочитать на моем сайте в разделе [«Публикации»](#).

Однако эксперты по эффективности коммуникаций утверждают, что слова могут передать только 10% информации, а остальные 90% передаются лишь при личном общении. Это действительно так. Сомнения, живая дискуссия, обсуждение с коллегами выполненных проектов позволяют намного быстрее изменить свое видение мира и адаптировать новые знания. В ходе тренингов методические рекомендации иллюстрируются примерами из жизни и закрепляются на практических занятиях, основанных на реальных проектах разработки ПО. Накопленный опыт управления разработкой ПО и руководства разработчиками сгруппирован в две учебные программы. Каждая из программ включает пять относительно независимых тренингов. Каждый тренинг подробно рассматривает вполне определенную область знаний в управлении разработкой ПО и не требует предварительного прохождения других тренингов.

## Программа «Практическое управление программным проектом»

Главной причиной большинства неудач в разработке ПО является не столько сложность и масштабность разрабатываемых продуктов, сколько применение неадекватных методов

управления программными проектами. Разработка ПО отличается от материального производства, поэтому опыт управления проектами, накопленный в этих отраслях, не всегда эффективен и применим в программных проектах. Цель программы передать слушателям профессиональные знания и практические навыки, специфичные в управлении проектами разработки ПО.

### ***Целевая аудитория***

Руководители проектов и групп разработчиков ПО, руководители функциональных подразделений.

### ***Предварительная подготовка***

Опыт участия в проектах разработки ПО.

### ***Продолжительность программы***

40 ак.час., из них теория – 25 ак. час. практические занятия – 15 ак. час.

### ***Продолжительность каждого тренинга***

8 ак.час., из них теория – 5 ак. час. практические занятия – 3 ак. час.

## **Успешный проект: от концепции до внедрения**

То, что производят программисты, нематериально – это коллективные мысли и идеи, выраженные на языке программирования. Я не утверждаю, что производство ПО суперсложная интеллектуальная деятельность. Отрасль еще только зарождается. Время вхождения в профессию намного меньше, чем в других инженерных дисциплинах. Разрабатывать ПО точно не сложнее, чем делать ракеты. Просто в силу уникальности отрасли управлять разработкой ПО надо иначе. Цель са - дать слушателям обзор методических и практических знаний, необходимых для эффективного управления программным проектом.

### **Рассматриваемые темы**

- Обзор современных методологий разработки ПО. Классификация методологий. Выбор и адаптация методологии под проект.
- Основные понятия и концепции управления проектами. Роль и место управления проектами. Типовые роли участников проекта разработки ПО. Жизненный цикл проекта. Фазы и продукты.
- Тест на выживание проекта. Как сделать проект успешным.
- Инициация проекта. Определение концепции проекта. Ресурсы. Сроки. Риски. Обоснование полезности проекта.
- Планирование проекта. Уточнение содержания. Правила эффективного планирования. Управление рисками. Подходы к оценке трудоемкости работ. Базовое расписание проекта.
- Реализация проекта. Мониторинг и управление работами. Принципы количественного управления. Руководство командой.
- Завершение проекта. Обратная связь. Рекомендации по итоговому отчету. Уроки проекта.

## Планирование – все..., планы – ничто

Многие программные проекты погубило не отсутствие планов, а их неполнота и нереальность. Как правильно планировать программный проект? Что забывают при планировании? Как организовать проектную команду? Главные причины провалов проектов. “Агрессивное” расписание и его последствия. Как надо расставлять приоритеты работ, чтобы увеличить вероятность успеха? Диаграмма Ганта и критический путь. Эти и другие вопросы будут рассмотрены на тренинге.

### Рассматриваемые темы

- Цель и задачи планирования. Уточнение содержания. Иерархическая структура работ. Правила эффективного планирования. Планирование управления содержанием проекта. Анализ и оценка изменений.
- Организационная структура проекта. Управление конфигурациями. Управление качеством.
- Выявление рисков. Анализ рисков. Планирование и осуществление мер по снижению рисков. Пример списка выявленных рисков. Оценка и ранжирование. Методы реагирования на риски. Главные причины неудачных проектов.
- Подходы к оценке трудоемкости работ. Влияние масштаба проекта. Влияние тип проекта. Другие факторы, влияющие на оценки. Основные причины недооценки трудоемкости и сроков. Негативные последствия недооценки.
- Планирование, направленное на раннее разрешение рисков.
- Определение зависимостей работ. Типы зависимостей. Базовое расписание проекта. Пример базового расписания. Критический путь в проекте. Пример «суперпроекта».

## Проект без рисков – удел неудачников!

Риски и выгода всегда ходят рука об руку. С рисками не надо бороться. Рисками надо управлять. Категории и характеристики рисков. Планирование управления рисками. Идентификация рисков. Качественный анализ рисков. Количественный анализ рисков. Планирование реагирования на риски. Главные риски программных проектов и способы реагирования. Управление проектом, направленное на снижение рисков. Мониторинг и контроль рисков.

### Рассматриваемые темы

- Статистика производства ПО. Общий уровень рисков проекта. Определение риска. Категории рисков. Характеристики риска. Пример характеристик риска. Управление рисками.
- Исходные данные для планирования. План управления рисками. Пример иерархической структуры рисков. Оценки воздействия рисков, вероятности рисков, близости наступления риска.
- Исходные данные для идентификации рисков. Инструменты и методы идентификации. Анализ документации. Методы сбора информации. Анализ контрольных списков. Анализ допущений. Анализ с использованием диаграмм. Методы сбора информации. Мозговой штурм. Опрос экспертов. Метод Дельфи. Карточки Кроуфорда. SWOT-анализ. Анализ допущений. Диаграмма зависимостей процесса. Пример диаграммы зависимостей. Результаты идентификации рисков. Пример списка выявленных рисков.
- Качественный анализ и ранжирование рисков. Оценка качества данных. Пример описания риска. Количественный анализ рисков. Методы сбора и представления данных. Опросы. Экспертные оценки. Методы количественного анализа рисков и моделирования. Анализ

чувствительности. Анализ дерева решений. Моделирование и имитация. Пример применения метода Монте-Карло. Планирование реагирования на риски. Методы реагирования на риски. Уклонение от риска. Передача риска. Снижение рисков. Принятие риска.

- Главные риски программных проектов и как им противодействовать. Требования заказчика отсутствуют / не ясны / подвержены частым изменениям. Отсутствие необходимых ресурсов и опыта. Отсутствие рабочего взаимодействия с заказчиком. Неполнота планирования. «Забытые работы».
- Управление, направленное на снижение рисков. Этапы в проектах с высоким уровнем рисков. Мониторинг и управление рисками. Пересмотр рисков. Аудит рисков. Анализ отклонений и трендов. Техническое измерение исполнения. Оценка и прогноз отклонений. Анализ резервов.

### Оценка проектов: шарлатанство или шаманство?

Неадекватные оценки трудоемкости и срока разработки ПО послужили причиной провала многих программных проектов. Профессор А.Н. Терехов в своем отзыве на мою книгу [«Лекции по управлению программными проектами»](#) назвал метод определения объема будущего ПО на основе функциональных точек «несколько шарлатанской идеей». Я докажу, что это не так. Это не шарлатанство, это - шаманство! :) Об этом и других подходах к оценке программных проектов, о влиянии на оценки масштаба проекта и сложности продукта пойдет речь на тренинге.

#### Рассматриваемые темы

- Подходы к оценке трудоемкости работ. Какая оценка лучше? Определение хорошей оценки. Субъективные факторы в оценке. Негативные последствия недооценки. Оценка – всегда вероятностная величина.
- Оценка PERT. Пример оценки. Распределение трудозатрат по процессам проекта. Влияние масштаба проекта. Влияние сложности продукта.
- Метод функциональных точек. История. Определение. Назначение. Достоинства. Процедура анализа. Определение типа оценки. Определение области оценки и границ продукта. Логические данные. Мера сложности данных. Пример: оценка объекта данных «клиент». Оценка транзакций. Мера сложности транзакций. Пример: оценка управляющей транзакции. Определение значения фактора выравнивания. Расчет количества выровненных функциональных точек.
- Метод оценки COSOMO II. История. Формула оценки трудоемкости проекта. Оценки размера SLOC / UFP. Факторы масштаба. Множители трудоемкости. Оценка многокомпонентного продукта. Оценка длительности проекта. Зависимость длительности от трудоемкости. Номинальная длительность проекта.
- Уточнение оценок на протяжении жизненного цикла проекта.

### Можно ли управлять тем, что нельзя измерить?

Основная тема - мониторинг и управление работами проекта. Управление - это расчленение, анализ, определение последовательности действий, конкретная реализация. Управление фокусируется на нижнем уровне: как сделать это наилучшим образом? Эта компетенция руководителя определяет эффективность движения по выбранному пути. Как правило, менеджеры, вышедшие из программистов, без особого труда овладевают необходимыми управленческими навыками. Но технические люди с подозрением относятся к тому, что нельзя

измерить. А все ли можно измерить в программном проекте? А как управлять тем, что измерению не подлежит?

### **Рассматриваемые темы**

- «Нельзя управлять тем, что нельзя измерить» - так ли это?
- Правда ли, что, чем больше метрик, тем лучше?
- Соотношение интеллектуальной и рутинной деятельности в разработке ПО.
- Рабочее планирование.
- Принципы количественного управления.
- Работает ли метод освоенного объема в программной инженерии?
- Семь основных метрик программного проекта.
- Показатели прогресса проекта.
- Показатели качества продукта.
- Практическое использование метрик.
- Статистические ловушки.
- Отраслевая статистика.
- Что вредно измерять в программном проекте.

## **Программа «Психология управления программным проектом»**

Руководителями программных проектов, как правило, становятся технические люди. Первая проблема, с которой они сталкиваются, заключается в том, что разработчики ПО не программы или технические устройства, которыми можно управлять. Поведение людей не вычислимо и плохо предсказуемо. Порой, требуются годы на то, чтобы научиться понимать людей и эффективно взаимодействовать с ним. У меня было время на то, чтобы найти ответы на ключевые вопросы. Почему только треть проектов разработки ПО завершаются в срок и укладываются в бюджет? Почему сверхвысокого IQ недостаточно для того, чтобы эффективно руководить программистами? Почему одни программисты могут быть на порядок, эффективнее других? Почему сотрудники не выполняют мои поручения? Почему правильно подобранные «лебедь, рак и щука» оказываются гораздо эффективнее «родственных душ»? И много других. Оказалось, что использование лучших языков и технологий программирования, самых совершенных инструментов разработки и систем качества не гарантируют успешность программного проекта. Люди – это главное, что определяет успех. В данной программе представлены результаты моего осмысления темы Peopleware.

### ***Целевая аудитория***

Руководители проектов и групп разработчиков ПО, участники проектов, руководители функциональных подразделений.

### ***Предварительная подготовка***

Опыт участия в проектах разработки ПО.

### ***Продолжительность программы***

40 ак.час., из них теория – 25 ак. час. практические занятия – 15 ак. час.

### ***Продолжительность каждого тренинга***

8 ак. час., из них теория – 5 ак. час. практические занятия – 3 ак. час.

## **«Все модели неверны, но некоторые - полезны». Психологические модели в руководстве командой разработчиков ПО**

Существуют сотни поведенческих моделей психики человека. Джордж Бокс сказал: «В сущности, все модели неверны, но некоторые - полезны». О тех психологических моделях, которые автору оказались полезны при систематизации многолетнего опыта руководства разработчиками ПО, пойдет речь на тренинге. Применение психологических моделей иллюстрируется примерами из практического опыта.

### **Рассматриваемые темы**

- Чем определяется поведение человека?
- Эмоциональный интеллект и проактивность.
- Опыт и поведение.
- Применение типологии Майрс-Бриггс в управлении разработчиками ПО.
- Поведение и командные роли по М.Белбину.
- Мотивация.
- Патологии поведения.
- Полезные и вредные конфликты. Конфликтология.
- Как сделать коммуникации эффективными.

## **Адаптивное управление проектом: примеры и принципы**

Что общего между ракетами, футболом и разработкой ПО? Почему классические методы управления не эффективны в разработке ПО? Как управлять, если структура и свойства объекта неизвестны или меняются со временем? Принципы адаптивного управления. Две главные задачи менеджера программного проекта. Необходимые и достаточные условия эффективной работы. Эффективные взаимодействия. Командообразование. Динамика команды.

### **Рассматриваемые темы**

- Аналогия эволюции методов управления программными проектами с историей АСУ. «Как получится»- можно, но не далеко и не точно. «Водопад» - лучше, но не эффективно. Agile методологии - «Планы - ничто, планирование - все». Самонаведение - «метод частых поставок». Адаптивное управление.
- Эффективность каждого участника рабочей группы. Принцип достаточного разнообразия. Все люди разные. Правильные люди. Неправильные люди. Четыре условия эффективной работы. Четыре функции руководителя. Проект – «кооперативная игра». Мотивация. Зависимость мотивов от опыта. Ошибки мотивации.
- Эффективные процессы взаимодействия. Принцип лидерства. Качества лидера. «Ситуационное лидерство». Делегирование. Главная работа менеджера - «точить пилу». Принцип четырех «П». Рекомендации по эффективности коммуникаций. Конфликты. Команды. Доктрина командного менеджмента. Четыре фазы командообразования. Принцип цикличности.

## **Построение эффективной команды с нуля**

Идеи командного менеджмента на Западе зародилась в начале 80-х годов. Эффективность команд в новых экономических условиях одними из первых оценили такие гиганты, как Procter & Gamble

и Boeing. Энтузиазм некоторых коллег, по поводу приходящих к нам с Запада разнообразных гибких методологий в разработке ПО, скорее всего от недостатка информации и опыта. Суть всех этих подходов состоит в том, что они декларируют своей высшей ценностью *командную работу*, ориентированность на людей и их взаимодействие, а не на процессы и средства. На мой взгляд, эти методологии похожи на попытки «переоткрытия» представителями индивидуалистской западной культуры принципов коллективизма, хорошо зарекомендовавших себя в России с давних времен.

#### **Рассматриваемые темы**

- Лидерство и управление. Роли и стратегии лидера. Проблемы неисполнения. Ситуационное лидерство.
- Чем отличаются рабочая группа и самоорганизующаяся команда. Доктрина командного менеджмента. Командные роли. Этапы формирования команды.
- Практики демотивации. Антипаттерны руководства командой разработчиков. Некомпетентный руководитель. Параноидальное управление. Последствия применения антипаттернов.
- Мотивация. Гуманистическая теория мотивации. Мотивация и тип личности. Мотивация и опыт.
- Подбор и развитие команд. Набор сотрудников. Как проводить техническое интервью. Как сохранить и умножить интеллектуальный капитал. Оценка и развитие персонала. Сколько надо платить сотрудникам?

### **Эффективный командный игрок – кто он?**

Известно, что производительность программистов может отличаться в десятки раз. Личная эффективность не определяется уровнем IQ. Для того чтобы стать эффективным, надо понять себя и научиться управлять собой. Но и этого недостаточно. Необходимо научиться эффективно взаимодействовать с другими людьми и стать командным игроком. На тренинге рассказывается о том, как можно этого достичь.

#### **Рассматриваемые темы**

- Почему сверхвысокого IQ недостаточно для эффективности. Ступени роста личной эффективности. Коэффициент эмоционального интеллекта. Проактивность – основа личной эффективности. Эффективное самоуправление. Компетенции профессионального разработчика.
- Почему люди поступают иначе, чем вы? Поведение и тип личности. Типы Майерс-Бриггс. Тип личности программистов. Темперамент программистов. Теория мотивации. Мотивы. Опыт и мотивация. Тип личности и мотивация. Ошибки мотивации.
- Доктрина командного менеджмента. Гибкие методологии. Традиции коллективизма в России. Группа или команда. Командные роли. Сбалансированность команды – залог эффективности. Динамика команды.
- Думать и действовать в духе «Выиграл/Выиграл». Цели коммуникации. Эффективность коммуникаций. Каналы передачи информации при личном общении. Язык жестов – 60% информации. Тип личности и стиль коммуникаций. «Я-сообщения». Открытые вопросы. Отклонение от темы. Доминирование. Проекция. Смещение. Стереотипы. Рекомендации по проведению собраний.

- Антипаттерны командного поведения. Классификация антипаттернов. Недостатки, с которыми приходится мириться. Недостатки, которые следует исправлять. Патологии.
- Заключение. Идеальный командный игрок.

## Как провалить проект. Антипаттерны управления проектом

Некоторые руководители программных проектов, в первую очередь начинающие, в работе постоянно наступают на одни и те же грабли. В компьютерной науке для подобных частонаступаемых граблей придумано название «антипаттерны» – это повторно используемые практики, которые могут давать видимость эффекта и даже временный эффект, однако, их применение наносит несоизмеримый ущерб конечному результату. Цель тренинга – собрать вместе некоторые наиболее тяжелые грабли (антипаттерны), которые встречаются на пути управления проектами разработки ПО, и постараться объяснить, как на них не наступать.

### Рассматриваемые темы

#### *Управление процессом*

- «Мертворожденный проект»
- «Зачем думать о плохом!»
- «В новом проекте все будет по-другому!»
- «Забытые работы»
- «Безграничный проект»
- «Нам нужен быстрый прогресс!»
- «Идеальный фреймворк»
- «Слепой полет»
- «Как написано в ТЗ, так и делали!»

#### *Руководство командой*

- «Я сделал все, что мог!»
- «Yes-man!»
- «Охота на ведьм».
- «Нет времени точить пилу!»

- «Агрессия».
- «Управление грибами».
- «Микроменеджмент».
- «Методологическое безумие».

#### *Командное поведение*

- «О сферических конях».
- «Все достало!»
- «Программист Ашманова».
- «Делаем все по правилам!»
- «Хороший парень».
- «Тихоня».
- «Непорядочность».
- «Прима».
- «Одинокий ковбой».
- «Социальный паразит».
- «Анархист».

## Контакты

[sergey@arkhipenkov.ru](mailto:sergey@arkhipenkov.ru)

[www.arkhipenkov.ru](http://www.arkhipenkov.ru)

Моб. тел.: +7 (916) 162 9454