

С. Архипенков

# Марш победителей

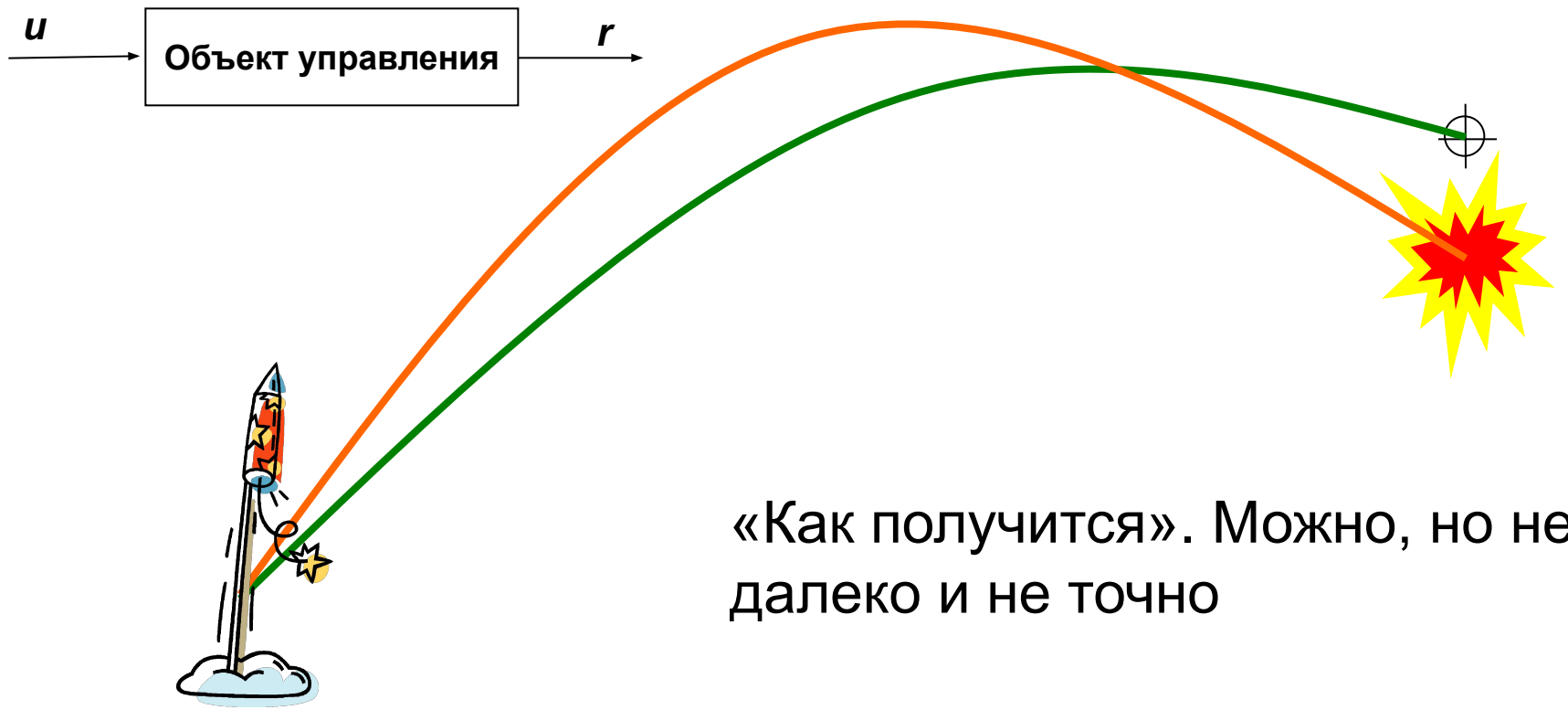
Адаптивное управление  
проектом: принципы и примеры

# Цель доклада

- Хорошо управляемый проект может быть успешно выполнен обычной командой разработчиков.
- Цель – рассказать, как можно и нужно это делать.

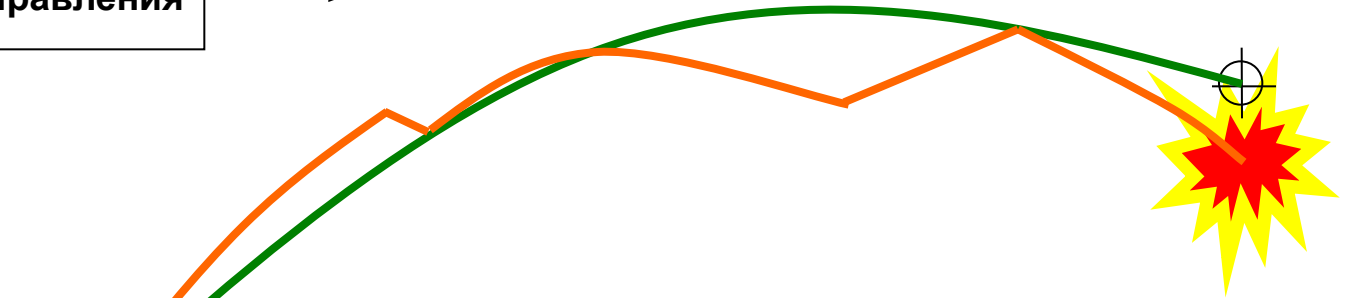
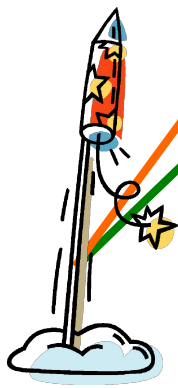
# Классические методы управления не работают

# Баллистический полет



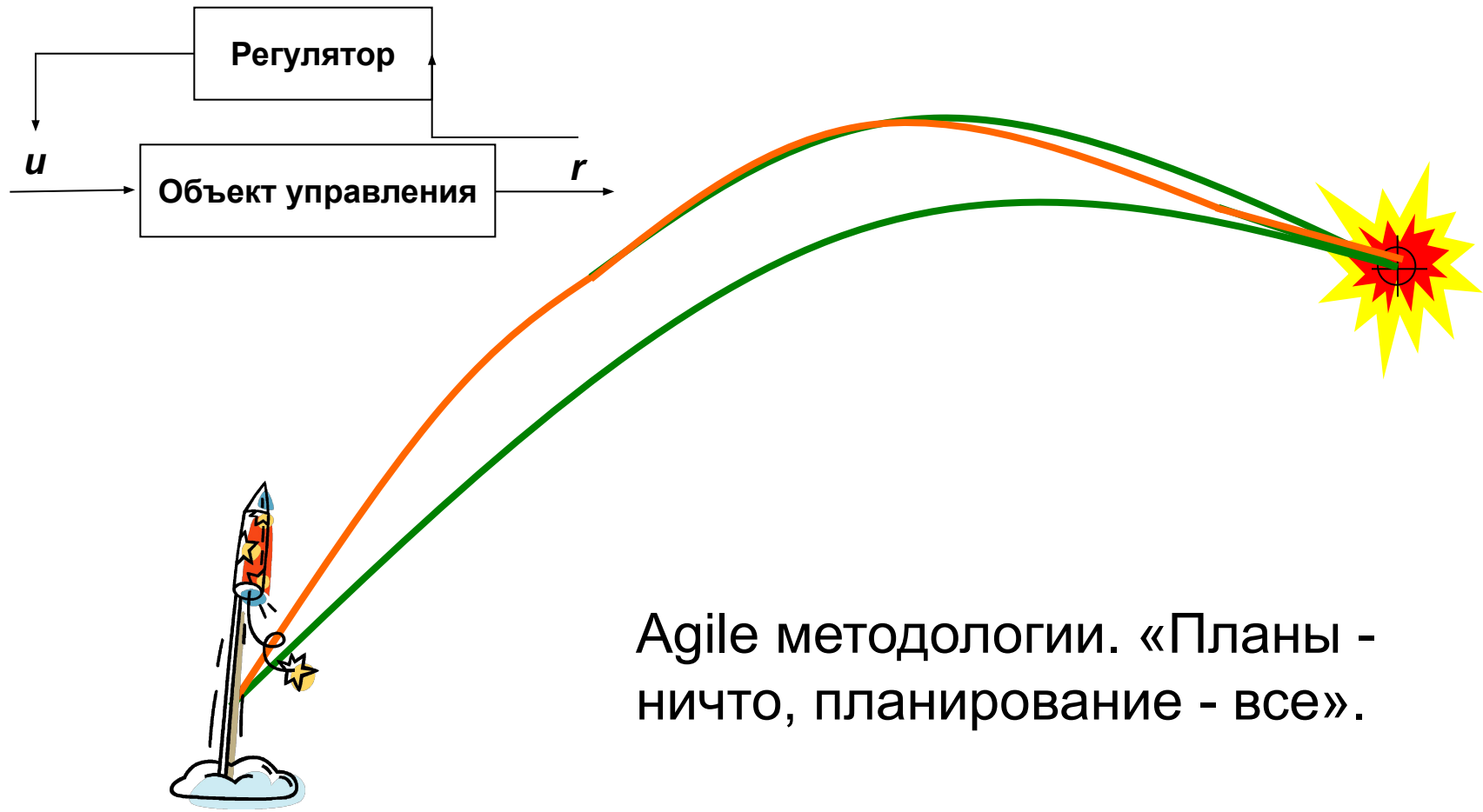
«Как получится». Можно, но не далеко и не точно

# Жесткое управление



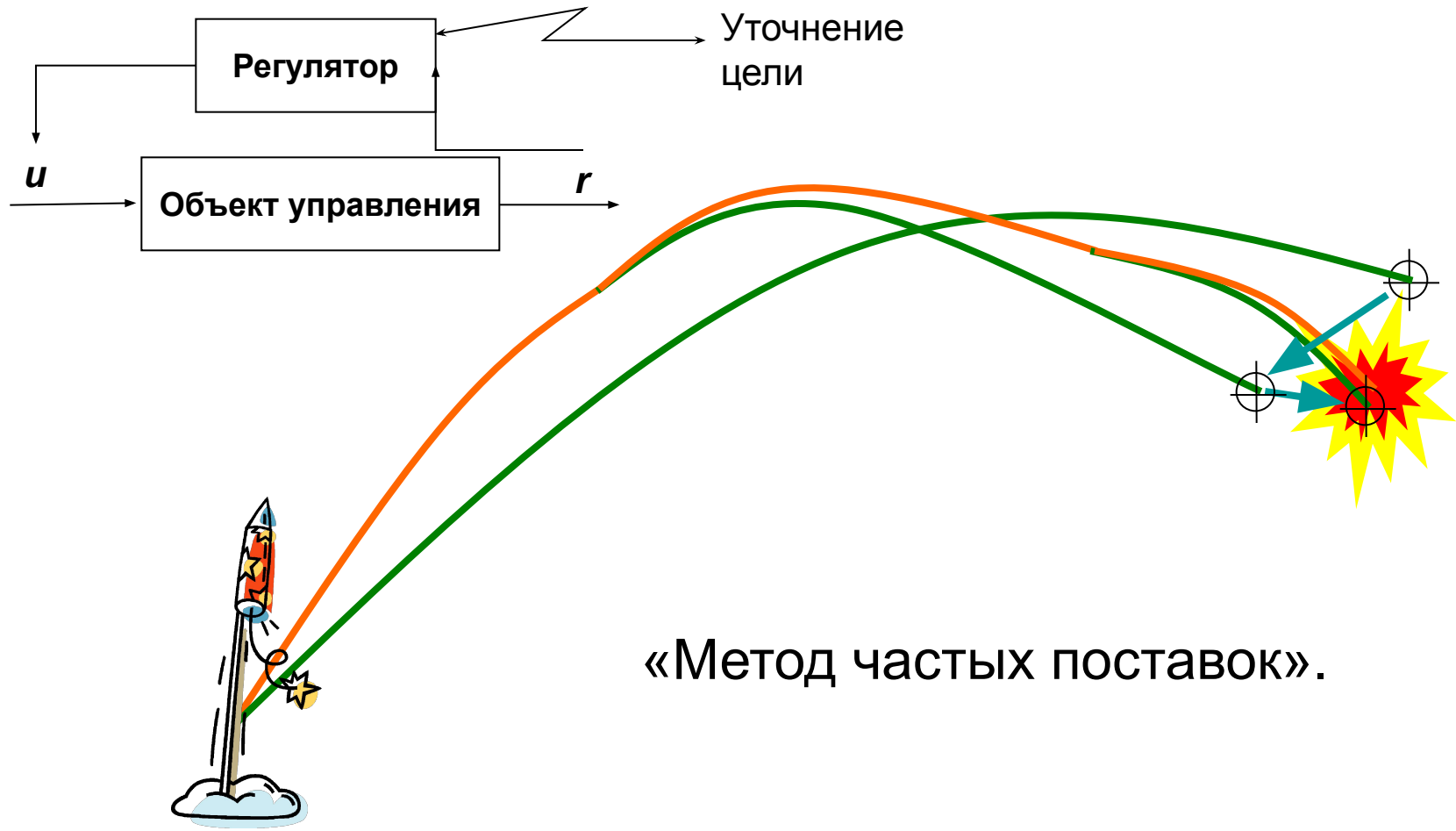
«Водопад». Лучше, но не эффективно

# Гибкое управление



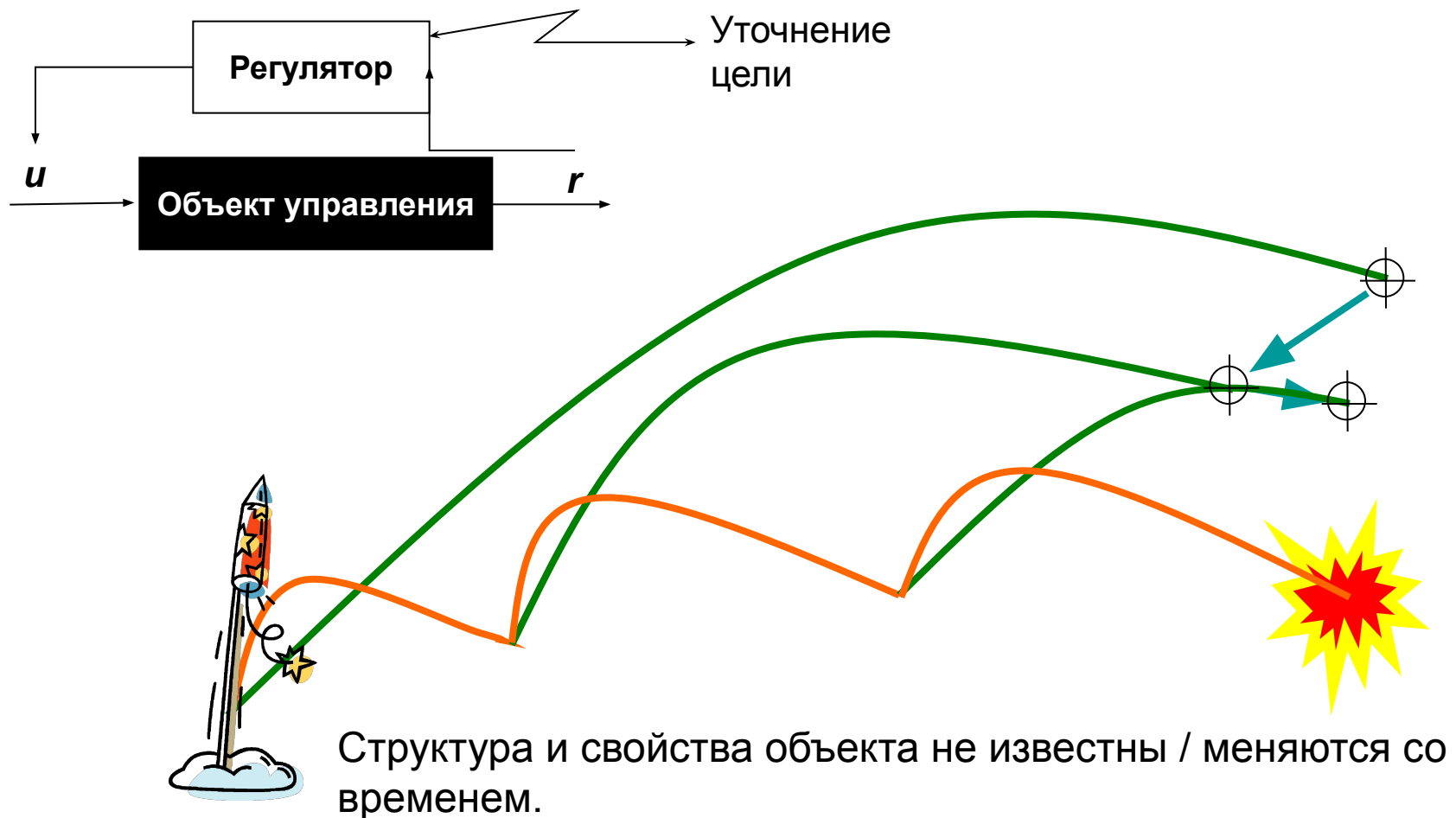
Agile методологии. «Планы -  
ничто, планирование - все».

# Самонаведение



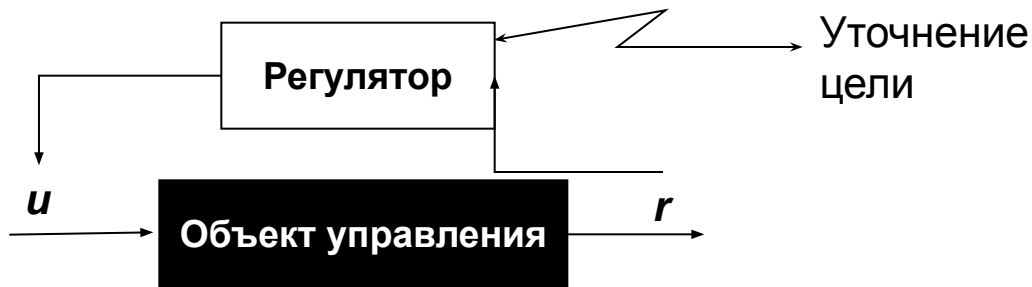
«Метод частых поставок».

# Классические методы не работают



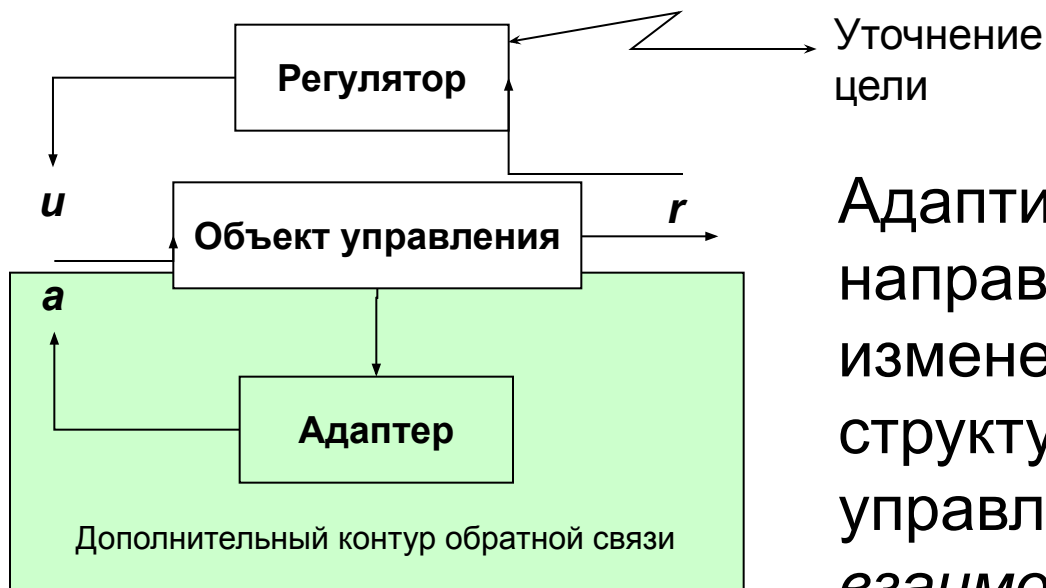


# Проект будет безнадежным



- Множество управляющих воздействий ( $u$ ) ограничено.
- Если у команды проекта низкая производительность, то постоянное давление, авралы, сверхурочные и субботники.

# Адаптивное управление



Адаптивное управление, направленно на изучение и изменение свойств и структуры объекта управления: *людей и их взаимодействия.*

Задачи руководителя:

1. Обеспечить эффективность каждого участника рабочей группы.
2. Обеспечить эффективные процессы взаимодействия.

# 1-я задача



Обеспечить эффективность каждого участника рабочей группы

# Принцип 1. Принцип достаточного разнообразия

Для «хорошего» управления количество возможных состояний управляющего устройства (разнообразие) должно быть не меньше, чем количество состояний объекта управления.

*Источник: У.Р.Эшби “Введение в кибернетику” М., ИЛ, 1959*



# Принцип 2. 4 условия эффективной работы

Для того чтобы ваш сотрудник мог эффективно решить поставленную вами задачу, необходимо и достаточно выполнение четырех условий:

1. Понимание целей работы.
2. Умение ее делать.
3. Возможность ее сделать.
4. Желание ее сделать.





# Принцип 3. 4 функции руководителя

Руководителю необходимо использовать четыре типа адаптационных воздействий. Это:

- 1) «направлять»,
- 2) «обучать»,
- 3) «помогать»,
- 4) «вдохновлять»



# Пример. «Хочет, может, но не делает»

- *Ситуация.* Программист стремится найти наиболее общее решение задачи, учесть все возможные последующие изменения и расширения. Старается разработать самый быстрый алгоритм, требующий минимальных ресурсов. Использует в решении все лучшие практики, паттерны проектирования, самые новые инструменты.
- *Классификация.* Неоправданное усложнение задачи. Программист неадекватно понимает цели проекта и приоритеты.
- *Решение.* «Направлять».

# Пример. «Хочет, но не может»

- *Ситуация.* Программист своевременно приходит на работу. Не отвлекается. Настойчиво работает над решением поставленных задач. Часто задерживается, чтобы уложиться в срок. Сроки, которые он сам оценивает, постоянно срываются.
- *Классификация.* Нет опыта.
- *Решение.* «Обучать».



# Пример. «Может, но не хочет»

- *Ситуация.* Программист имеет глубокие знания, нацелен на решение трудных задач. Пользуется авторитетом среди коллег. В начале активно выдвигал новые идеи и добивался их внедрения. В середине проекта потерял интерес. Постоянно заваливает сроки, делает глупые ошибки. Расхолаживающе влияет на команду.
- *Классификация.* Программист - «генератора идей», не мотивирован на методичную реализацию.
- *Решение.* «Вдохновлять».

# Пример. «Может, но не хочет 2»

- *Ситуация.* Программист активен, самостоятелен, напорист. По любому вопросу имеет свое мнение. Всегда стремится быть победителем в конфликтах. Часто оценивает других и указывает им на недостатки. Использует любой повод, чтобы продемонстрировать свое превосходство.
- *Классификация.* Эгоист.
- *Решение.* Избавиться.

# 2-я задача



Обеспечить эффективные процессы  
взаимодействия

# Принцип 4. Принцип лидерства

Руководитель программного проекта должен стать лидером, вокруг которого сплотится эффективная команда.



# Принцип 5. 4 стратегии лидера

Не существует одной лучшей стратегии руководства. В зависимости от готовности участников рабочей группы выполнять задания руководителя, он должен использовать одну из 4-х стратегий:

S1. «Директивное управление».

S2. «Объяснения».

S3. «Участие».

S4. «Делегирование».



*Источник: Hersey P., Blanchard K.H. "Management of Organizational Behavior", 6th ed., Englewood Cliffs: Prentice-Hall, 1993.*

# Пример. «Ситуационное лидерство»



# Командообразование

4 стадии развития:

4. Performing

3. Norming

2. Storming

1. Forming

Эффективность



# Пример. «Шумиха»

- *Ситуация.* Частые смены приоритетов задач. Споры о том, что надо делать, а что не надо. Сомнения в реальности сроков. Недовольство отсутствием прогресса. Много вопросов по каждой задаче.
- *Классификация.* Команда находится на первом этапе образования – «Forming».
- *Решение.* Стратегия S1. «Директивное управление». Функции: «направлять» и «помогать».



# Пример. «Неразбериха»

- *Ситуация.* На совещаниях бесконечные неконструктивные споры и дискуссии. Постоянно доминируют одни и те же лица. Мнения высказываются как объективные факты. Постоянно даются оценки. Присутствует агрессия.
- *Классификация.* Команда находится на стадии «Storming».
- *Решение.* Стратегия S2. «Объяснения». Функции: «обучать» и «помогать».

# Пример. «Группомыслие»

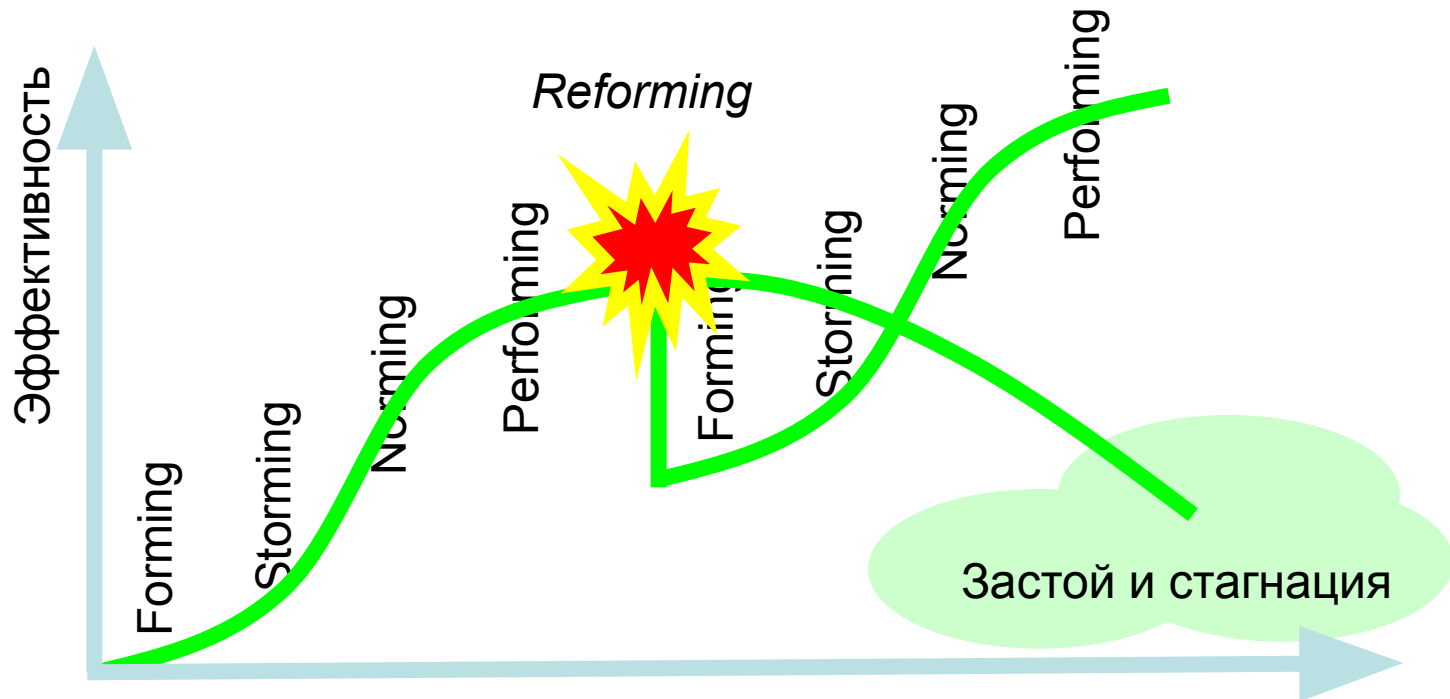
- *Ситуация.* Девиз участников проекта: «Давайте работать, а не конфликтовать!» Все стараются избегать конфликтов и поддерживать согласие. Как правило, никто не спорит с руководителем и следуют его указаниям.
- *Классификация.* «Парадокс Абилина».
- *Решение.* Стратегия S3. «Участие». Функции: «помогать» и «вдохновлять».

# Пример. «Менеджер должен занимать очередь...»

- *Ситуация.* Ни одно предлагаемое участником команды решение не принимается на веру. Активно анализируются возможные негативные последствия или упущенные возможности. Конфликты носят исключительно производственный характер. «Менеджер проекта должен занимать очередь, чтобы покритиковать сотрудника, не выполняющего свои обещания» (с) Том Демарко
- *Классификация.* Команда находится на стадии «Performing».
- *Решение.* Стратегия S4. «Делегирование». Функции: «направлять» и «вдохновлять».

# Принцип 6. Принцип цикличности

4 стадии становления команды должны циклически повторяться, чтобы обеспечить непрерывный рост эффективности.



С. Архипенков, SEC(R)  
2008

28 Время

# Принцип 7. Принцип победителей

Программист состоит из четырех компонентов: тело, сердце, разум и душа.

1. Телу необходимы деньги и безопасность.
2. Сердцу - любовь и признание.
3. Разуму – развитие и самосовершенствование.
4. Душе – самореализация.



# Благодарности

- Спасибо всем моим бывшим и настоящим коллегам, совместно с которыми (на которых?) я приобретал свой опыт.
- Отдельная благодарность Гусу Хидинку и сборной России по футболу.

Спасибо за внимание.  
Вопросы?

Адаптивное управление  
проектом: принципы и примеры

Сергей Архипенков  
[www.arkhipenkov.ru](http://www.arkhipenkov.ru)  
[contact@arkhipenkov.ru](mailto:contact@arkhipenkov.ru)