

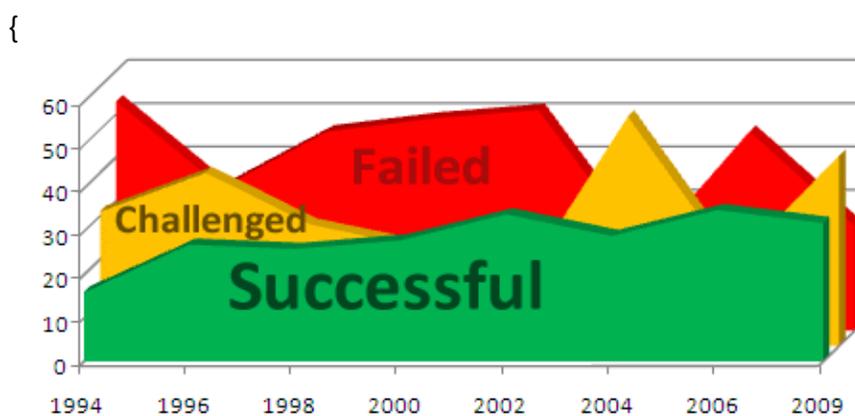
SPM(*guild*);

ТЕОРИЯ И ПРАКТИКА АДАПТИВНОГО УПРАВЛЕНИЯ ПРОЕКТОМ

С. Архипенков. Москва, 16 июля 2013



Еще раз о «Хаосе» ()



(C) Standish Group }

SPM(*guild*);

Миф: разрабатывать ПО просто ()

{ Факты:

- Много лет только 1/3 проектов укладывается в треугольник: качество-стоимость-срок;
- Отраслевой прирост производительности – 3-5% в год;
- 80% – интеллектуальная работа;
- Пространство выбора решения – не ограничено;



SPM(*guild*);

}
3

Миф: разрабатывать ПО сложно ()

{ Факты:

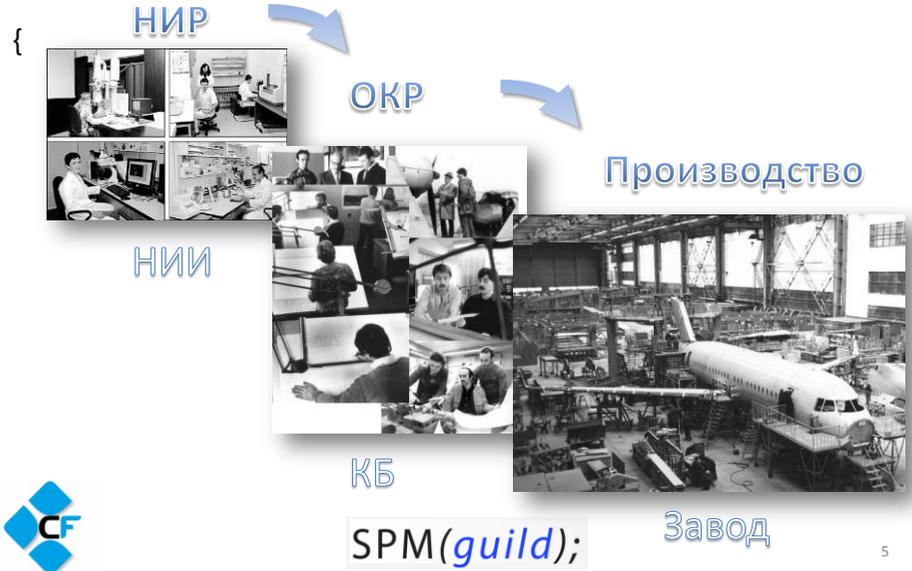
- Время вхождения в профессию меньше, чем в других отраслях;
- Нет физических ограничений;
- Ограниченные усилия могут принести колоссальный эффект;
// Большинство выдающихся программных продуктов создано студентами в гараже;



SPM(*guild*);

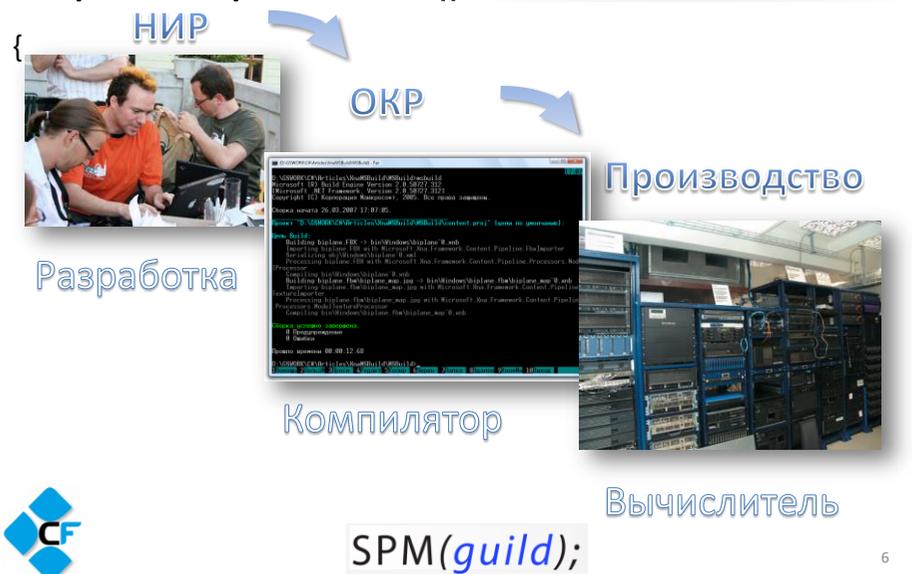
}
4

Миф: разработка ПО похожа на другие производства ()



5

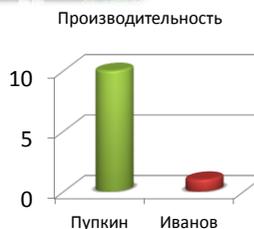
Факт: разработка ПО – это только проектирование ()



6

Миф: разработка ПО – это инженерная дисциплина ()

- { Факт:
- Разработка ПО это гуманитарная дисциплина
 - Гласс: Отсутствует теория;
 - Брукс:
 - Отсутствуют средства визуализации;
 - Неопределенность и изменчивость;
 - Отличие производительности более чем в 10 раз. (Гласс: в 28 раз!);
 - Коуберн: Успешность программного проекта на 100% определяется людьми;

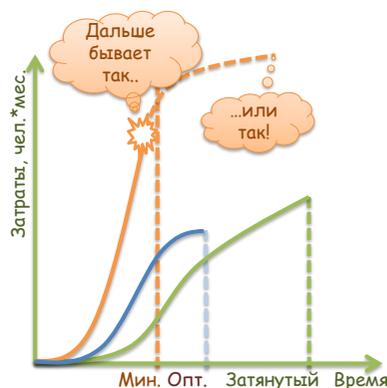


SPM(*guild*);

7

Миф: разработку ПО можно ускорить ()

- { Факты:
- Девять беременных женщин не родят ребенка за месяц;
 - Ни один проект невозможно завершить быстрее, чем за 3/4 расчетного оптимального графика;
 - Демарко: «наиболее эффективно программисты работают без жестких сроков»;



SPM(*guild*);

8

Миф: проблемы можно «закидать деньгами» ()

{ Факты:

- Добавление людей в отстающий проект только увеличит отставание;
- Сверхурочные и авралы снижают производительность;
- Применение новых инструментов снижают производительность, но если повезет могут ее повысить на 5-10%;



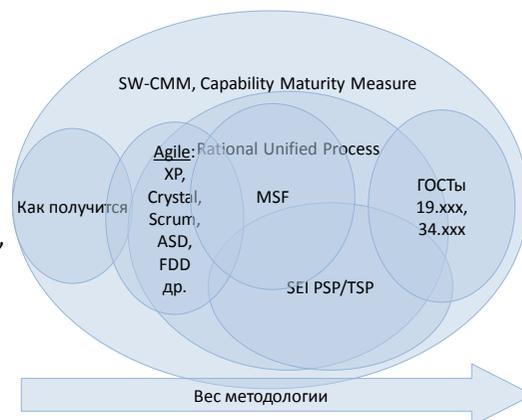
SPM(*guild*);

9

Миф: проблемы можно решить при помощи процесса ()

{ Факты:

- Существуют десятки методологий, но ни одна не гарантирует результат;
- Коуберн не обнаружил корреляции между успехом/провалом проектов и методологиями, которые применялись в проектах;



SPM(*guild*);

10

Миф: работу программиста можно измерить ()

{ Факты:

- Не работает;
- НЕ РАБОТАЕТ!
- Работу программиста можно оценить;



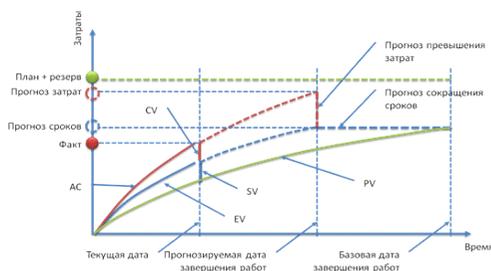
SPM(guild);

11

Миф: работу программистов нельзя измерить ()

{ Факт – обязательно надо измерять:

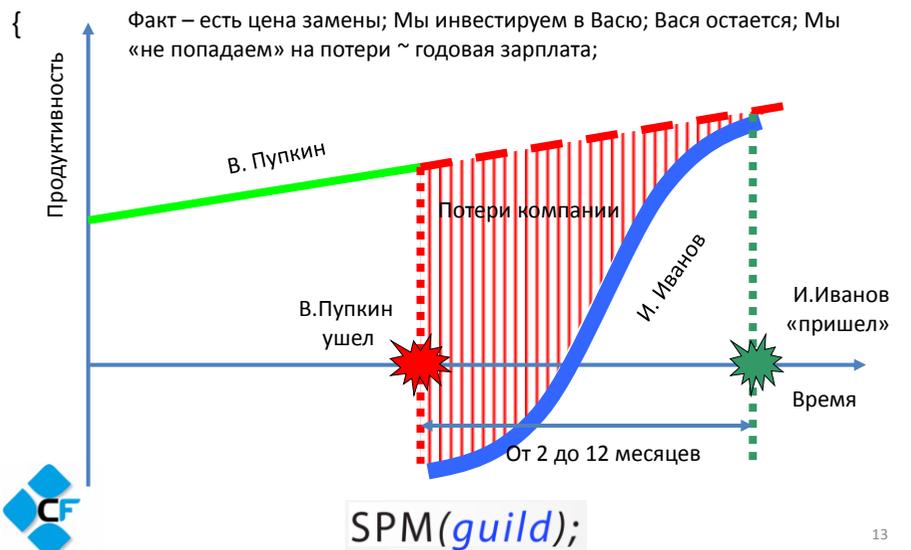
- Количество протестированных функций;
- Количество изменений;
- Объем кода;
- Средняя производительность;
- Дефектность продукта;
- Качество архитектуры;



SPM(guild);

12

Миф: незаменимых людей нет ()



Миф: программисты анти-бюрократичны ()

{ Факты:

- Программисты – анти-идеотичны;
- Программисты сторонятся тех, кто усложняет их работу;
- Программисты самоорганизуются вокруг тех, кто облегчает их работу;



SPM(guild);

14

Миф: о железном треугольнике ()

{ Факты:

- Разработка продукта длилась 5 лет вместо одного года;
- Бюджет проекта был превышен более, чем в 5 раз;
- Это - провал?

// Нет - это был MS Word!



SPM(*guild*);

15

Миф: программистами невозможно управлять ()

{ Факты:

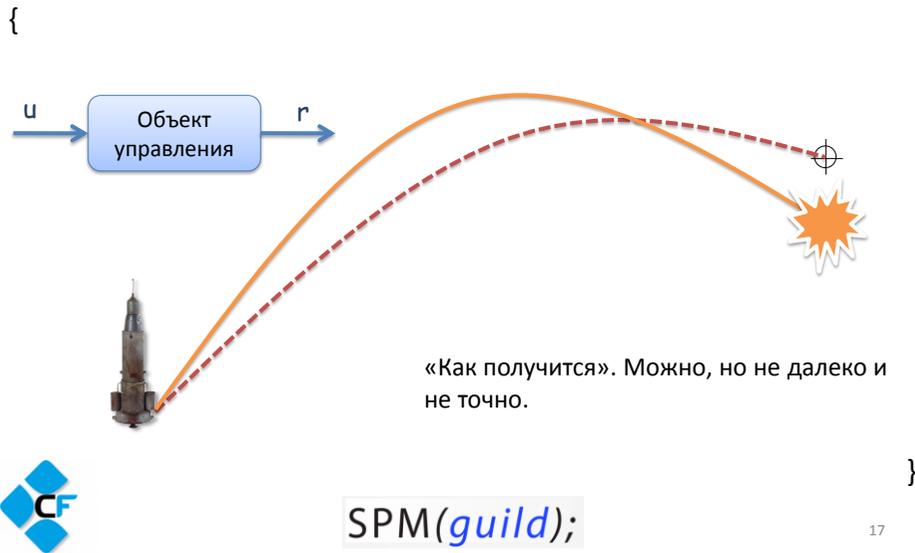
- Методы Макдональдса не работают;
- Управлять разработкой ПО надо иначе;



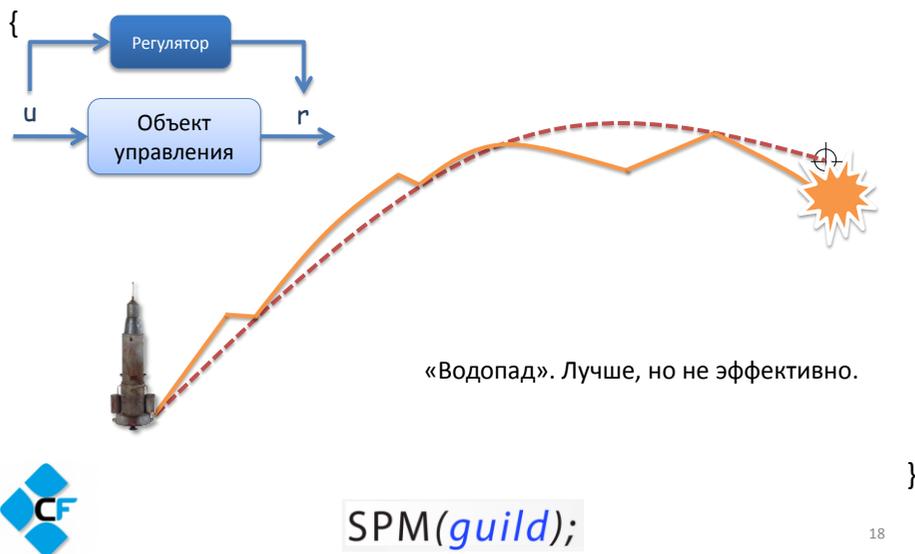
SPM(*guild*);

16

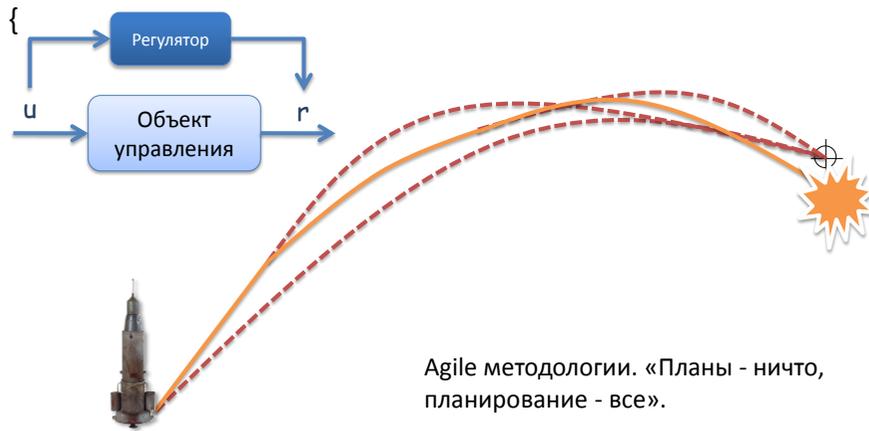
Баллистический полет ()



Жесткое управление ()



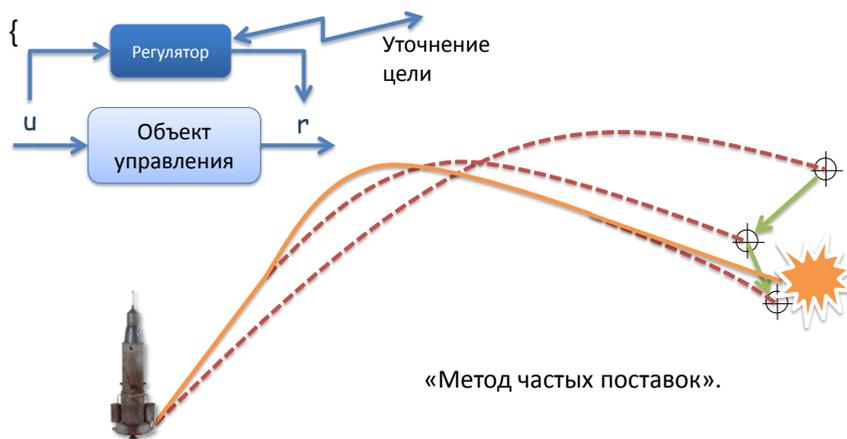
Гибкое управление ()



SPM(*guild*);

19

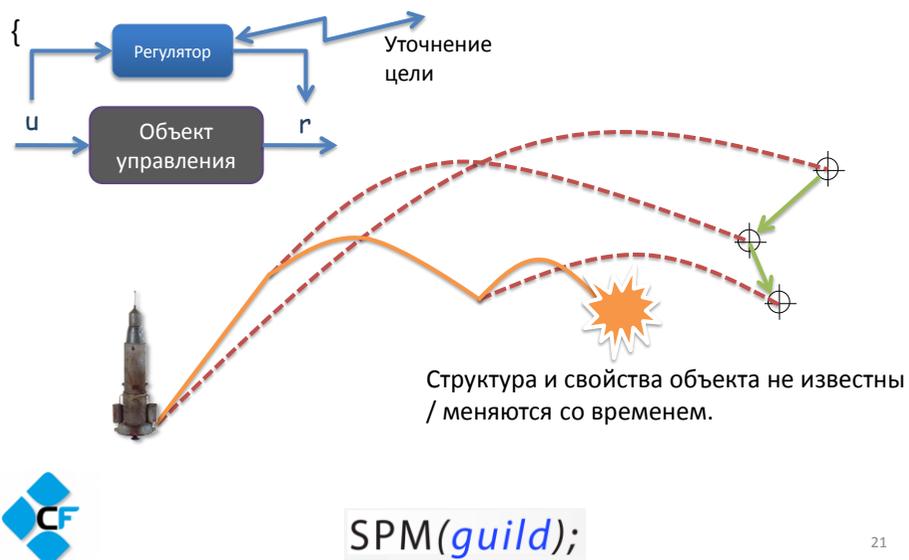
Самонаведение ()



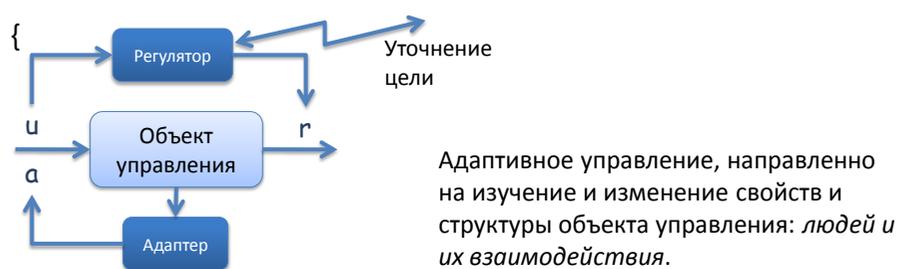
SPM(*guild*);

20

Классические методы не работают ()



Адаптивное управление ()



Задачи руководителя:

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

История. Винни-Пух против Пятачка ()

{

// Коммуникация

- Пятачок: «Я переживаю за качество нашего кода! Пользователи нашей системы люди, а не машины!»;
- Винни-Пух: «Заказчик нам платит за функциональность, а не за качество кода или «бантики» на интерфейсе!»;

// Результат

- Пятачок: «Это Винни думает только о сиюминутной прибыли! Доброе имя компании для него пустой звук!»;
- Винни: «Этот Пятачек заботится только о том, чтобы все всегда были довольны! Прибыль и интересы компании для него ничего не значат!»;



}



SPM(guild);

23

Принцип 1. Принцип достаточного разнообразия ()

{

«Только разнообразие может уничтожить разнообразие»;

// Источник: У.Р.Эшби
"Введение в кибернетику"
М., ИЛ, 1959



}



SPM(guild);

История. Программист Ашманова ()

{ // Программист:

- Ну, не знаю, у меня на машине всё работает;
- Я уже неделю ночами работаю, а вы меня укоряете за срыв срока;
- К пятнице готово не будет, но в понедельник - точно. Или во вторник.
- Чего там планировать, я быстрее сделаю и всё уже будет работать;
- Планировать разработку бессмысленно, жизнь всё равно богаче;
- Программные проекты всегда срывают сроки потому, что это сложное и творческое дело, вроде научных исследований;



SPM(*guild*);

25

Принцип 2. Четыре условия эффективной работы ()

{

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

- Понимание целей работы;
- Умение ее делать;
- Возможность ее сделать;
- Желание ее сделать;



SPM(*guild*);

26

История. Делаем все по правилам ()

{

// Программист

- Стремиться сделать наиболее общее решение задачи, учесть все возможные последующие изменения и расширения;
- Стараются разработать самый быстрый алгоритм, требующий минимальных ресурсов;
- Использует в решении все лучшие практики, паттерны проектирования, самые новые инструменты;



SPM(*guild*);

27

}

История. Все достало ()

{

// Старший программист

- Имеет глубокие знания и развитый интеллект, быстро осваивает все новое, нацелен на решение трудных задач. Пользуется заслуженным авторитетом среди коллег;
- В начале проекта активно выдвигал новые идеи, убедительно их обосновывал, добивался их признания всеми. Находил неизвестные возможности, существенно сократившие трудоемкость работ по проекту;
- В середине проекта потерял интерес. Стал «витать в облаках» и отвлекаться на изучение каких-то новых технологий. Постоянно заваливает сроки, делает глупые ошибки, непростительные для его опыта. Расхолаживающе воздействует на команду;



SPM(*guild*);

28

}

Принцип 3. Четыре функции руководителя ()

{



Штурман-
направляет



Наставник-
обучает



Помощник-
обеспечивает



Вдохновитель-
мотивирует

}



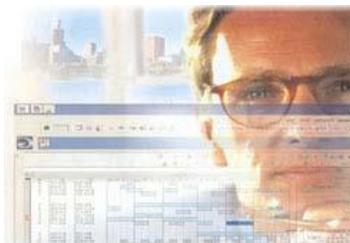
SPM(*guild*);

29

Программист ()

{

- Проводит декомпозицию задачи и проектирует ее решение;
- Адекватно оценивает затраты на выполнение;
- Планирует свою работу и составляет график;
- Соблюдает принятые стандарты;
- Обеспечивает требуемое качество, минимизируя затраты и риски;
- Выполняет тестирование и отладку кода;
- Анализирует найденные дефекты и отклонения от графика;
- Корректирует свой рабочий процесс для их предотвращения в будущем;
- Старший программист, дополнительно - наставничество, передача знаний и опыта.



}



SPM(*guild*);

30

История. Делай, как я сказал ()

{

- Менеджер проекта дает задание программисту и объясняет, как его надо делать;
- Программист начинает спорить и доказывать, что решать задачу надо другим путем;
- Менеджер уверен, что этот путь тупиковый и настаивает на своем подходе: «Делай, как я сказал!»



SPM(*guild*);

31

}

Принцип 4. Принцип лидерства ()

{

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



SPM(*guild*);

32

}

Навыки лидера ()

- Видение целей и стратегии их достижения;
- Глубокий анализ проблем и поиск новых возможностей
- Нацеленность на успех, стремление получить наилучшие результаты;
- Способность сочувствия, понимания состояния участников команды;
- Искренность и открытость в общении;
- Навыки в разрешении конфликтов;
- Умение создавать творческую атмосферу и положительный микроклимат;
- Терпимость, умение принимать людей какие они есть, принятие их права на собственное мнение и на ошибку;
- Умение мотивировать правильное профессиональное поведение членов команды;
- Стремление выявлять и реализовывать индивидуальные возможности для профессионального роста каждого;
- Способность активно "обеспечивать", "доставать", "выбивать" и т.д.



SPM(*guild*);

33

Принцип 5. Четыре стратегии лидера ()



SPM(*guild*);

34

История. Звездун ()

{

- Имеет опыт и квалификацию;
- Активен, самостоятелен, напорист;
- По любому вопросу имеет свое собственное мнение;
- Всегда стремится быть победителем в конфликтах;
- Часто оценивает других и указывает им на недостатки;
- Умничает. Использует любой повод, чтобы продемонстрировать свое превосходство;
- Переоценивает свой личный вклад в общее дело, поэтому считает, что он должен работать меньше, чем его «менее способные» коллеги;



}

SPM(*guild*);

35

Не стоит нянчиться ()

{

- Людей воспитывают в детском саду, ну еще немного в начальной школе;
- Каждый взрослый человек имеет то, к чему он осознанно или неосознанно стремится;
- Нянчиться и воспитывать взрослого человека - это значит ограждать его от проблем, закрывать ему путь к переосмыслению своего опыта и развитию, «загонять болезнь внутрь» при помощи «социального аспирина»;



}

SPM(*guild*);

36



КАЧЕСТВА КОМАНДНОГО ИГРОКА



SPM(*guild*);

37

Инициатива ()

- { • Занимает активную позицию, стремится расширить свою ответственность и увеличить личный вклад в общее дело.



SPM(*guild*);

38

}

Саморазвитие ()

- Постоянно приобретает новые профессиональные знания и опыт, выдвигает новые идеи, направленные на повышение эффективности достижения общих целей, добивается распространения своих знаний, опыта и идей среди коллег.



SPM(*guild*);

39

Самореализация ()

- Получает удовольствие от своей работы, гордится ее результатами и стремится, чтобы эти же чувства испытывали все коллеги.



SPM(*guild*);

40

Взаимоуважение ()

- { • Четко осознает свои личные и общие цели, понимает их взаимообусловленность, настойчиво стремится к их достижению.



SPM(*guild*);

41

Открытость ()

- { • Уверен в себе и в своих коллегах, объективно оценивает их достижения и успехи, внимательно относится к их интересам и мнениям, активно ищет взаимовыгодное решение в конфликтах.



SPM(*guild*);

42

Оптимизм ()

- Является оптимистом, при этом твердо знает, что окружающий мир несовершенен. воспринимает каждую новую проблему, как дополнительную возможность подтвердить собственный профессионализм в своих глазах и во мнении коллег.



SPM(*guild*);

43

История. Менеджер должен занимать очередь ()

- Ни одно предлагаемое решение не принимается на веру. Все требуют факты для его обоснования;
- Активно анализируются возможные негативные последствия или упущенные возможности при принятии решения;
- Конфликты носят исключительно производственный характер; При решении конфликтов активно ищутся взаимовыгодные возможности;
- «Менеджер проекта должен занимать очередь, чтобы покритиковать сотрудника, не выполняющего свои обещания»;



SPM(*guild*);

44

История. Сверхлояльность ()

{

//Девиз группы: «Давайте работать, а не конфликтовать!»;

- Все члены команды стараются избегать конфликтов и поддерживать согласие;
- Как правило, никто не спорит, все соглашаются с мнением руководителя и следуют его указаниям;
- При возникновении трудных ситуаций все ждут решения от руководителя;
- Редкие противоречия разрешаются путем взаимных уступок;



}

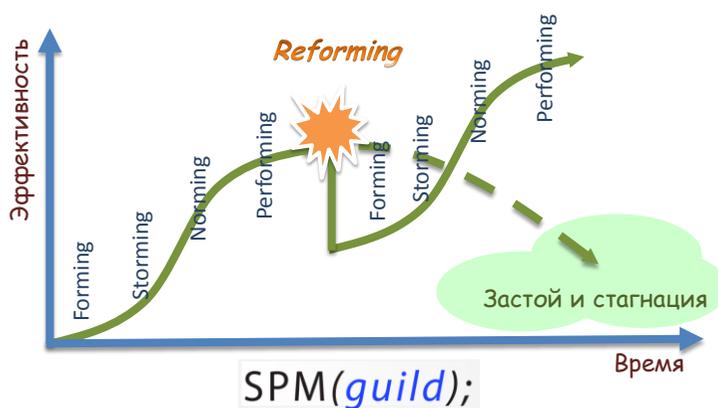
SPM(*guild*);

45

Принцип 6. Принцип цикличности ()

{

Четыре фазы становления команды должны циклически повторяться, чтобы обеспечить непрерывный рост эффективности;



46

Точить пилу ()

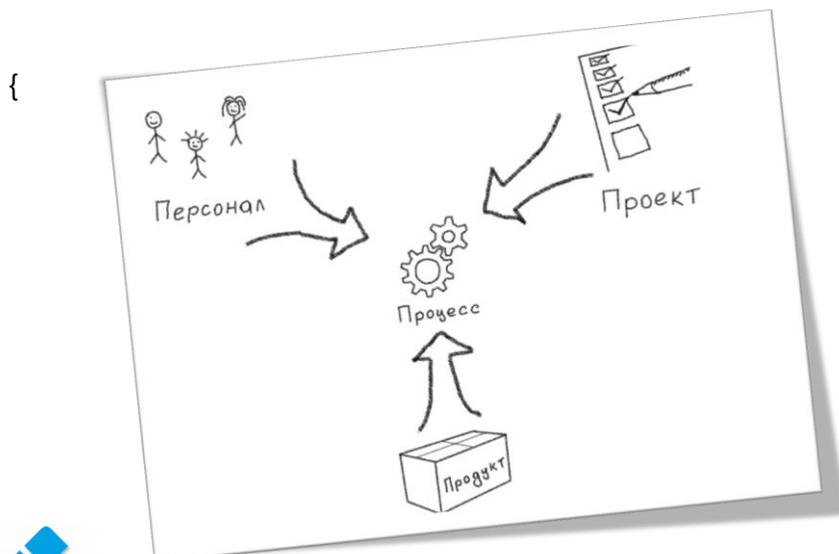
- { // Работать на опережение, «играть от защиты»:
- Постоянный мониторинг и оценка эффективности всех процессов, используемых в проекте. «Что лишнее мы делаем?» «Что можно делать проще?» «Что угрожает проекту?». Сокращение ненужных усилий вместо «стремления к новым победам»;
 - Определение узких мест и применение корректирующих действий там, где процессы начинают буксовать или риски слишком велики;
 - Важно. Не команда должна приспосабливаться к процессам, а процессы должны перестраиваться под команду по мере ее развития и становления;



SPM(*guild*);

47

Принцип 7. Принцип четырех «П» ()



SPM(*guild*);

48

О счастье ()

{

// 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;



}



SPM(*guild*);

49

ВОПРОСЫ



SPM(*guild*);

50