

компьютерного
Центр[®]
(ОБУЧЕНИЯ)
«СПЕЦИАЛИСТ»
при МГТУ им. Н.Э.Баумана

**Тестирование программного
обеспечения**

Управление командой тестировщиков



Ушакова Елена Сергеевна –

e-mail: eushakova@specialist.ru

Рекомендуемая литература



- **Мифический человеко-месяц**
- Фредерик Брукс
- Издательство: Символ-Плюс, 2006 г.
- Эта книга – своего рода библия для разработчиков программного обеспечения во всем мире, написанная Бруксом еще в 1975 году. Полагают, что без прочтения книги Брукса не может состояться ни один крупный руководитель программного проекта

Рекомендуемая литература



- **Ключевые процессы тестирования. Планирование, подготовка, проведение, совершенствование**
- Рекс Блэк
- В этой книге Рекс Блэк выделяет двенадцать процессов тестирования, являющихся ключевыми для достижения успеха. За описанием каждого из этих процессов следует пример использования процесса. Вместо громоздких правил представлены списки контрольных вопросов - легкие, гибкие инструменты для внедрения тестирования, ориентированного на процесс, для сбора измерений и внесения последовательных изменений.

Модуль 1. Тест-менеджмент

- Место тестирования в процессе разработки ПО
- Тестирование и качество. Оценка качества продукта. Метрики
- Базовые принципы тестирования
- Планирование работ по тестированию.
- Виды деятельности, осуществляемые при составлении плана тестирования
- Артефакты тестирования
- Модели зрелости тестирования ПО (ТММi)
- Инструментарий тест менеджера

Что такое тест менеджмент?

Что делает тест менеджер?

- Формирует команду
- Планирует тестирование
- Выбирает инструменты
- Внедряет новое
- Собирает метрики

- Построение, развитие и управление командой
- Управление внутри организации
- Миссия, политики, стратегии и цели
- Доменные и проектные факторы
- Управление внешними связями
- Оценка эффективности и действенности
- Основы проектного менеджмента
- Оценка и отчетность тестирования в проекте

Люди

Стратегия

Оперативная
деятельность

Базовый уровень



- Процесс тестирования
- Риски и тестирование
- Управление дефектами

Продвинутый уровень ТМ



- Тест менеджмент
- Навыки управления командой
- Улучшение процесса тестирования

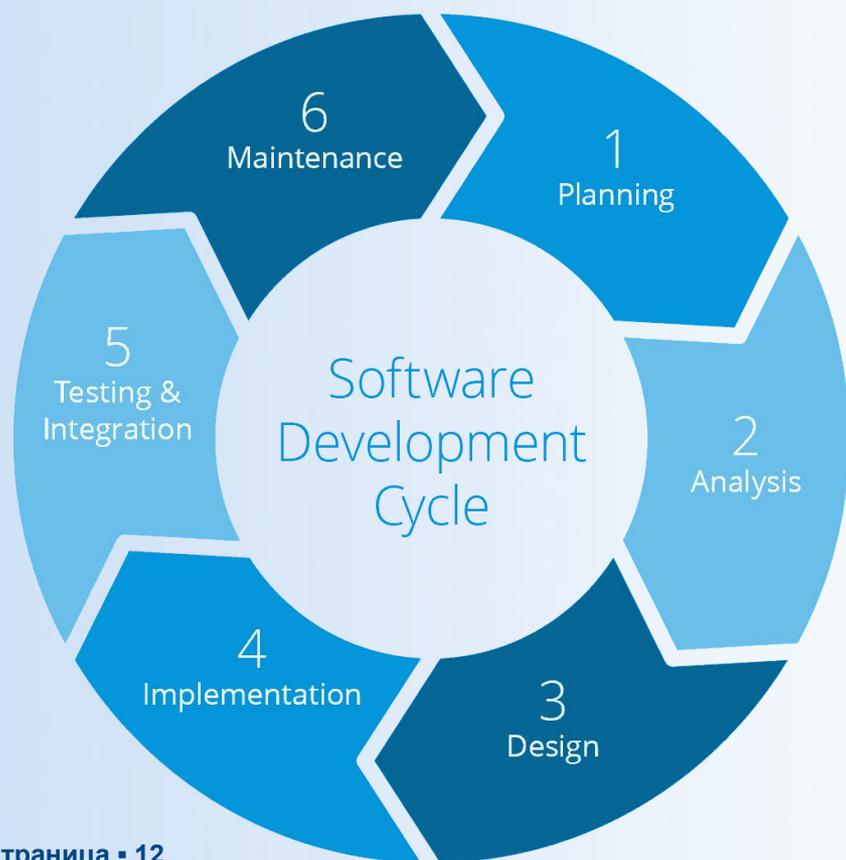
Уровень эксперта ТМ

- Стратегическое управление
- Оперативное руководство
- Управление командой



Место тестирования в процессе разработки ПО

SDLC (Software development lifecycle) - это процесс, используемый индустрией программного обеспечения для проектирования, разработки и тестирования высококачественного программного обеспечения.



1. Планирование и анализ потребностей
2. Определение требований
3. Проектирование и дизайн
4. Разработка ПО
5. Тестирование
6. Развертывание

STLC - жизненный цикл тестирования ПО



1. Анализ требований
2. Планирование
3. Проектирование тестов
4. Настройка тестовой среды
5. Выполнение тестов
6. Закрытие тестового цикла

S
D
L
C

Requirements Gathering

Design

Coding or development

Testing

Deployment

Maintenance

Requirements Reviewed & Understanding

Test Analysis, Test Planning & Test design

Test case writing and Auto-test Development if
Automation is involved

Test Execution (Manual and/or Automation) &
Defect Reporting & Retesting

Final Testing and Preparing Final Test Executive
Summary Report Preparation

Maintenance Testing

S
T
L
C

Оценка качества продукта.

1. Пригодность для использования. Наличие свойств, атрибутов и функций, удовлетворяющих ожиданиям заказчиков и пользователей, и отсутствие свойств, атрибутов и функций, неудовлетворяющих ожиданиям пользователей.
2. Соответствие требованиям. Наличие свойств, атрибутов и функций, удовлетворяющих всем установленным требованиям, и отсутствие свойств, атрибутов и функций, отклоняющихся от требований.

Тестирование — это наблюдение за функционированием ПО в специфических условиях с целью определения степени соответствия ПО требованиям к нему.

Тестирование само по себе не изменяет ПО, а значит, не способно влиять на те метрики качества, которые зависят только от самого ПО.

Тестирование может служить методом контроля качества ПО, а именно тех его характеристик, которые проявляются при функционировании ПО.



ISO/IEC 25010



**ГОСТ Р ИСО/МЭК 25010-2015 Информационные технологии (ИТ).
Системная и программная инженерия. Требования и оценка качества
систем и программного обеспечения (SQuaRE). Модели качества
систем и программных продуктов**

- Модель качества при использовании**
- Модель качества продукта**

Модель качества продукта

Защищенность

Конфиденциальность
Целостность
Неподдельность
Отслеживаемость
Подлинность

Функциональная пригодность

Полнота
Корректность
Целесообразность

Уровень производительности

Временные характеристики
Использование ресурсов
Потенциальные возможности

Переносимость

Адаптируемость
Устанавливаемость
Взаимозаменяемость

Качество системы/ программного продукта

Совместимость

Сосуществование
Интероперабельность

Сопровождаемость

Модульность
Возможность многократного использования
Анализируемость
Модифицируемость
Тестируемость

Удобство использования

Определимость пригодности
Изучаемость
Управляемость
Защищенность от ошибки пользователя
Эстетика пользовательского интерфейса
Доступность

Надежность

Завершенность
Готовность
Отказоустойчивость
Восстанавливаемость

Рабочее определение метрики

Количественный показатель, «снимаемый» с объектов рабочего процесса для удовлетворения определенной информационной потребности

- «снимаемый» = способ измерения, правило расчета
- «информационная потребность»: напр., принятие менеджерского решения путем интерпретации полученного значения

Пример метрики: «Число незакрытых дефектов»

- способ измерения, правило расчета: «всего дефектов» минус «число закрытых» (данные из системы управления дефектами)
- менеджерское решение: «выпускать продукт?»

Оценка качества продукта. Метрики

- Полнота реализации функций. Используется для измерения пригодности.
- Корректность реализации функций. Используется для измерения пригодности.
- Отношение числа обнаруженных дефектов к прогнозируемому. Используется для определения зрелости.
- Отношение числа проведенных тестов к общему их числу. Используется для определения зрелости.
- Отношение числа доступных проектных документов к указанному в их списке. Используется для измерения анализируемости.

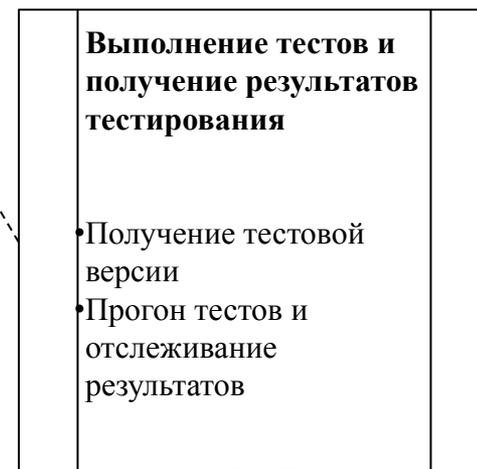
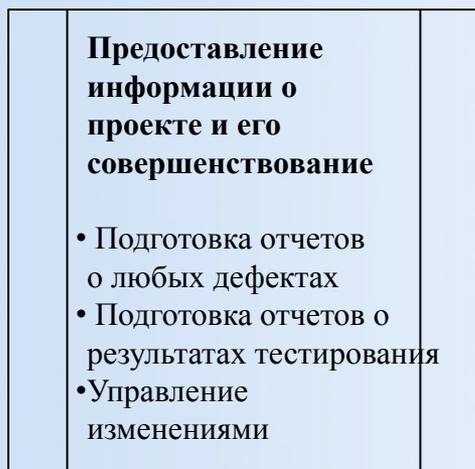
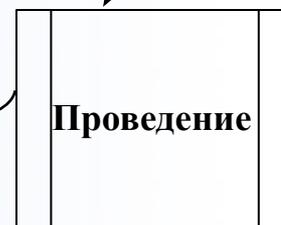
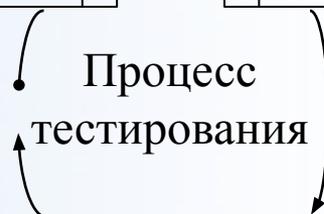
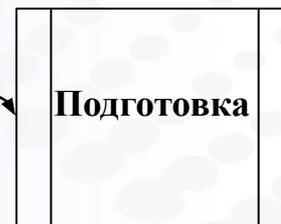
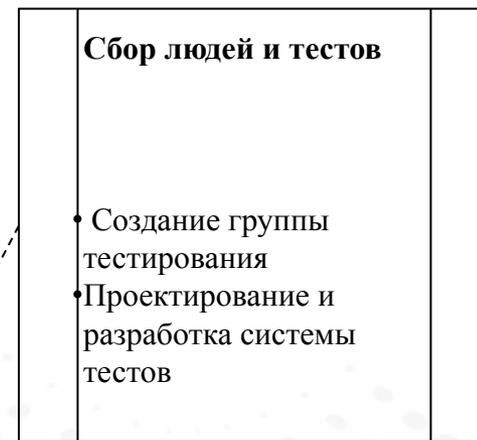
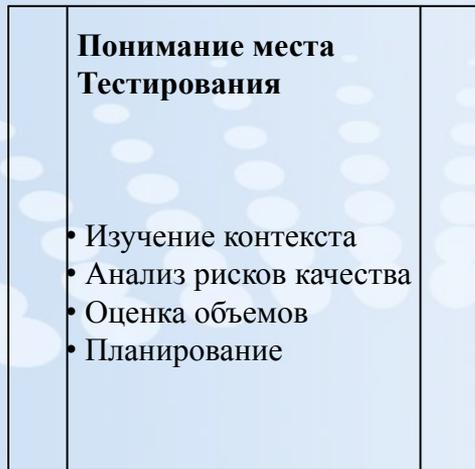
Базовые принципы тестирования

1. Тестирование демонстрирует наличие дефектов.
2. Исчерпывающее тестирование невозможно.
3. Раннее тестирование.
4. Скопление дефектов.
5. Парадокс пестицида.
6. Тестирование зависит от контекста.
7. Заблуждение об отсутствии ошибок.

Процессы тестирования

Процессы тестирования - это последовательность действий, работ и наблюдений, предпринимаемых для достижения цели.

1. *Планирование* – понимание места тестирования
2. *Подготовка* – сбор людей и тестов
3. *Проведение* – выполнение тестов и сбор результатов
4. *Совершенствование* – предоставление информации о проекте и его улучшение.



Процесс тестирования

Этап	Описание
1.	Планирование понимания места тестирования.
1.А	Определить функциональный (система, проект и процесс) и организационных контексты, в рамках которых выполняется тестирование.
1.В	Определить и расставить приоритеты рисков качества системы, достичь консенсуса заинтересованных сторон проекта относительно возможностей тестирования по смягчению этих рисков.
1.С	Оценить временные, ресурсные и финансовые затраты, необходимые для выполнения тестирования и согласованные с пунктом 1.В, получить поддержку оценки у руководства
1.Д	Запланировать задачи, зависимости и состав участников, которые необходимы для смягчения рисков качества системы, и получить поддержку плана у заинтересованных сторон проекта.

2.	Подготовка: сбор людей и подготовка тестов.
2.А	Создать команду специалистов - тестировщиков, обладающих соответствующими знаниями, навыками, отношением к делу и мотивацией, посредством поиска персонала и его обучением.
2.В	Спроектировать, разработать, получить и верифицировать систему тестов, которую группа тестирования будет использовать для оценки качества системы, предназначенной для тестирования.

3.	Проведение: выполнение тестов и сбор результатов
3.А	Получить и установить тестовую версию, состоящую из всех или нескольких компонентов системы, предназначенной для тестирования.
3.В	Определить, отслеживать и управлять комплектом тестов, с помощью которого планируется проверять каждую из тестовых версий.
4.	Совершенствование: предоставление информации о проекте и его улучшение.
4.А	Документировать ошибки, обнаруженные в ходе выполнения тестов.
4.В	Обсудить результаты тестирования с ключевыми заинтересованными лицами.
4.С	Управлять изменениями и уточнить процесс тестирования.

Уровни зрелости TMMi. Цели тестирования

Процессные
области



Модели зрелости тестирования ПО (ТММi)

Процессные
области

2. Управляемый процесс

- 2.1 Политика и стратегия тестирования
- 2.2 Планирование тестирования
- 2.3 Отслеживание и контроль тестирования
- 2.4 Проектирование и выполнение тестов
- 2.5 Тестовое окружение

3. Определенный процесс

- 3.1 Подразделение по тестированию
- 3.2 Программа подготовки тестировщиков
- 3.3. Жизненный цикл тестирования интегрирован в жизненный цикл разработки продукта
- 3.4 Нефункциональное тестирование
- 3.5 Экспертная оценка

4. Измеримый процесс

- 4.1 Измерения тестирования
- 4.2 Оценка качества продукта
- 4.3 Расширенная экспертная оценка

5. Оптимизация

- 5.1 Предотвращение дефектов
- 5.2 Оптимизация процесса тестирования
- 5.3 Контроль качества

1. Тестирование есть

Конкретные цели

Пример. 2 уровень. Конкретные цели

SG 1 Внедрение политики тестирования

SP 1.1 Определение целей тестирования

SP 1.2 Определение политики тестирования

SP 1.3 Распространение политики тестирования среди заинтересованных сторон

SG 2 Внедрение тестовой стратегии

SP 2.1 Определение общих рисков продукта

SP 2.2 Определение стратегии тестирования

SP 2.3 Распространение стратегии тестирования среди заинтересованных сторон

SG 3 Внедрение показателей эффективности тестирования

SP 3.1 Определение показателей эффективности

SP 3.2 Внедрение показателей эффективности

SP 1.1 Определение целей тестирования

1. Изучите бизнес потребности и цели
 - a. *Формулировка миссии*
 - b. *Потребности бизнеса и пользователей в отношении продукта*
 - c. *Движущие силы бизнеса*
 - d. *Основные цели качества продукта*
 - e. *Тип бизнеса (на пример, уровень риска разрабатываемого продукта)*
2. Обеспечьте обратную связь для уточнения бизнес потребностей и целей по мере необходимости.
3. Определите цели тестирования, связанные с бизнес потребностями и задачами.
 - f. *Подтвердить «пригодность к использованию» продукта*
 - g. *Предотвратить появление дефектов в работе продукта*
 - h. *Проверка на соответствие внешним стандартам*
 - i. *Обеспечение прозрачности в отношении качества продукта*
 - j. *Сокращение времени выполнения тестов*

4. Согласование целей тестирования с заинтересованными сторонами
5. Пересмотр целей тестирования по мере необходимости (например, ежегодно)

SG 3. Внедрение показателей эффективности тестирования

SP 3.1 Определение показателей эффективности тестирования.

Показатели эффективности тестирования определяются на основе политики и целей тестирования, включая порядок сбора, хранения и анализа данных.

a. Показатели

b. Процедура сбора, хранения, анализа

2. Изучите политику и цели тестирования, например, цели улучшения процесса тестирования.
3. При необходимости предоставьте обратную связь для уточнения политики и целей тестирования.
4. Определите индикаторы производительности теста, связанные с политикой и целями тестирования.
 - a. Усилия и стоимость тестирования
 - b. Время выполнения теста
 - c. Количество обнаруженных дефектов
 - d. Процент обнаружения дефектов
 - e. Тестовое покрытие
 - f. Уровень зрелости теста

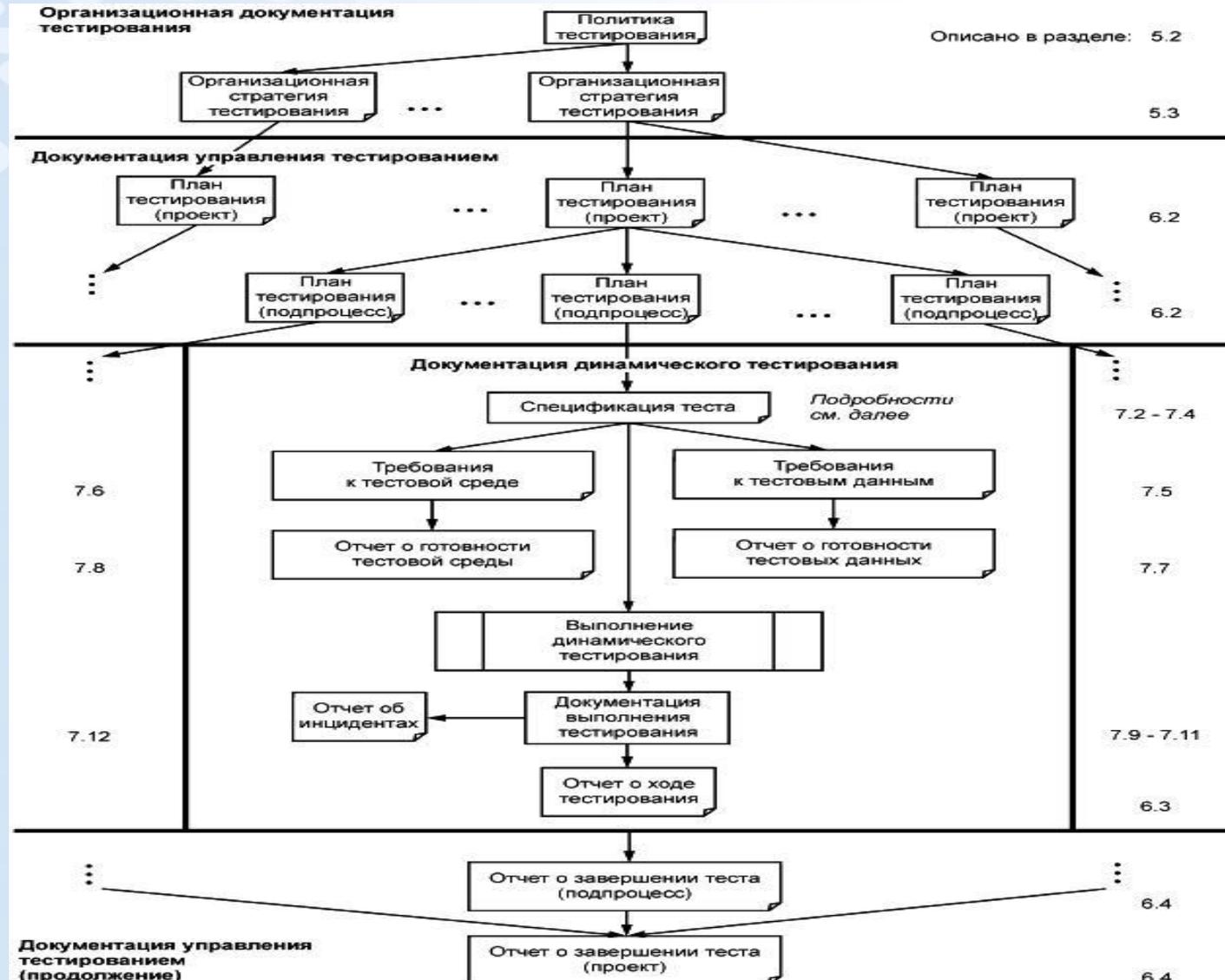
TPI Next

	Initial	Controlled				Efficient				Optimizing		
1 Stakeholder commitment	1	2	3	4	1	2	3		1	2	3	
2 Degree of involvement	1	2	3	4	1	2	3		1		2	
3 Test strategy	1	2	3	4	1	2	3		1		2	
4 Test organization	1	2	3	4	I	I	J	J	1	2	3	
5 Communication	1	2	3	4	1	2	3		1		2	
6 Reporting	1	2	3	4	1	2	3		1		2	
7 Test process management	1	2	3	4	1	2	3		1		2	
8 Estimating and planning	1	2	3	4	1	2	3	4		2	3	
9 Metrics	1	2	3	4	1	2	3	4	1		2	
10 Defect management	1	2	3	4	1	2	3	4	1	2	3	
11 Testware management	1	2	3	4	1	2	3		1	2	3	
12 Methodology practice	1	2	3	4	1	2	3	4	1		2	
13 Tester professionalism	1	2	3	4	1	2	3	4	1	2	3	
14 Test case design	1	2	3	4	1	2	3	4	1	2	3	
15 Test tools	1	2	3	4	1	2	3	4	1	2	3	
16 Test environment	1	2	3	4	1	2	3	4	1	2	3	

Метрики в тестировании

- Плотность дефектов ($SDD = \text{Число дефектов} / \text{Размер кода}$)
- Плотность дефектов после поставки ($PDDD = \text{Число дефектов после поставки} / \text{Размер кода}$)
- Доля отклоненных дефектов ($DDR = \text{Число отклоненных дефектов} / \text{Число дефектов}$)
- «Убойность» тестов ($DP = \text{Число дефектов} / \text{Число тестов}$)
- Эффективность тестирования ($TE = \text{Число дефектов} / \text{Трудозатраты тестирования}$)
- Доля покрытия требований ($RCR = \text{Число требований, покрытых тестами} / \text{Число требований}$)
- Плотность покрытия требований ($RCD = \text{Число тестов} / \text{Число требований}$)
- Доля повторно открытых дефектов ($RDR = \text{Число повторно открытых дефектов} / \text{Число дефектов}$)

Артефакты тестирования



Планирование

- План тестирования

проектирование

- Спецификация тестирования

Выполнение

- Документация выполнения теста

Анализ
результатов

- Отчет о завершении тестирования

ГОСТ Р 56922-2016/ISO/IEC/IEEE 29119-3:2013 Системная и программная инженерия. Тестирование программного обеспечения. Часть 3.
Документация тестирования

Политика тестирования

- **Цели и определение тестирования**

Определяет назначение, цели и полную область применения тестирования в организации. Устанавливает позицию организации, почему выполняется тестирование и какую цель преследует.

- **Процесс тестирования**

Идентифицирует процесс тестирования, которому будет следовать организация. Это может быть ссылкой на конкретный документ, предоставляющий подробную информацию о процессе тестирования.

- **Организационная структура тестирования**

- **Подготовка тестеров**

- **Этика тестера**

- **Стандарты**

Стратегии тестирования

Стратегия тестирования содержит следующую информацию:

- Как использовать тестирование для управления рисками проекта и продукта?
- Как разделить процесс тестирования на разные уровни тестирования?
- Что такое тестирование высокого уровня?
- Какую стратегию тестирования следует использовать в какой ситуации? Стратегии могут различаться в зависимости от требований проекта, таких как нормативные требования, уровни риска и различные методологии разработки программного обеспечения.
- Общие условия входа в тест и выхода из него
- Действия и процессы, упомянутые в Стратегии тестирования, должны соответствовать Политике тестирования организации.

Стратегия Тестирования

Общий неспецифичный для конкретного проекта документ организационного уровня, который обеспечивает руководящие принципы для проектов в областях их применения

- Подпроцессы тестирования
- Практические результаты
- Методы проектирования тестов
- Критерии начала и окончания
- Требуемые метрики
- Требования к данным и средам
- Регрессионное тестирование
- Критерий приостановки и возобновления

Примеры стратегий

- Аналитическая стратегия
 - На основе анализа рисков
 - На основе требований
- Стратегия на основе модели
 - Модель существующей или предполагаемой ситуации
 - Модель данных (входных, выходных)
- Методическая стратегия
 - Соответствие стандарту качества
 - Стандартные контрольные списки проверок (на пример, безопасность)
- Стратегия соответствия стандартам или процессов
- Реактивная стратегия
 - На основе поставленного кода
 - На основе дефектов, обнаруженных в реальной системе
- Консультативная стратегия
- Стратегия предотвращения регрессии

Виды деятельности, осуществляемые при составлении плана тестирования



Инструменты

Принципы выбора

- Соотношение затрат к выгоде
- Оценка необходимости обучения
- Определение внутренних требований (передача знаний, опыта..)
- Бесплатный пробный период
- Интеграция с другими инструментами
- Возможности улучшения процесса тестирования

Инструменты. Системы управления тестированием

- ALM Octane
- Test IT
- TestRail
- Zephyr
- Allure EE
- TM4J
- Qase
- PractiTest
- Testuff
- Azure
- MTM TFS
- Kualitee
- Trello
- Ситечко
- Jira
- Trac
- Bugzilla
- Asana
- ...

<https://software-testing.ru/library/around-testing/management/3457-top-12-best-test-management-systems-2020>

Модуль 2. Риски тестирования. Команда тестирования.

- Работа с рисками. Риски проекта и продукта. Риски в тестировании
- Типичные риски. Технические, проектные, организационные.
- Участники команды. Роли и сферы ответственности.
- Сравнительная характеристика централизованной группы и локальной.
- Задачи руководителя: делегирование, контроль, мотивация.
- Формирование новой командой. Управление «старой» командой
- Найм и интеграция сотрудников в команду.

Риски. Определение

- **Риск** — действующий или развивающийся фактор процесса, обладающий потенциалом негативного влияния на ход процесса
- **Риск** — лишь потенциальный, а не обязательный результат

**Опасен не сам риск, а его
последствия**

Работа с рисками. Риски проекта и продукта

Риски в тестировании



Основные составляющие управления рисками

- **Выявление риска:** первоначальный мозговой штурм по выявлению риска, последующая сортировка и определение какого-либо механизма для обеспечения постоянного действия данного процесса.
- **Анализ воздействия риска:** количественная оценка каждого риска в терминах вероятности его наступления и потенциального ущерба.
- **Планирование реагирования на риски:** что вы собираетесь делать, если и когда данный риск наступит.

- **Ослабление риска:** меры, которые должны быть приняты предварительно, чтобы обеспечить возможность и эффективность проведения запланированных действий, если они потребуются.
- **Мониторинг и управление рисками:** отслеживание рисков, выделенных в качестве объектов управления, выявление материализации рисков.

Извлечение уроков

- Извлечение уроков формализует усвоение накопленного опыта в форме, доступной для использования как внутри группы, так и на уровне всей компании
- Даже если Вы уверены, что все помнят усвоенные уроки, занесите информацию в базу знаний

Где есть неопределенность, там появляется риск.

- Требования к системе: Что именно должна делать система?
- Обеспечение стандартов взаимодействия: Как будет система взаимодействовать с людьми-операторами и другими системами того же уровня?
- Влияние изменяющейся среды: Как во время разработки будут изменяться потребности и цели?
- Ресурсы: Какие ключевые навыки и знания исполнителей возможно будет (при необходимости) привлечь по мере продвижения работы над проектом?
- Управление: Хватит ли у руководства таланта, чтобы создать эффективные команды, поддерживать боевой дух, обеспечивать низкую текучесть кадров и координировать сложные комплексы взаимосвязанных задач?

- **Сеть поставок:** Будут ли другие участники проекта действовать так, как ожидалось?
- **Политика:** Каков может быть результат использования политической силы для навязывания ограничений, несовместимых с успехом проекта?
- **Конфликты:** Как различные участники проекта найдут компромисс между своими, зачастую несовместимыми, целями?
- **Инновации:** Как уникальные для данного проекта технологии и методы влияют на возможный результат?
- **Масштаб:** Как повлияет на осуществление проекта увеличение масштаба работ, если раньше у разработчика не было соответствующего опыта?

Работа в рисками

- Наименование риска
 - Источник риска 1
 - Источник риска 2
 - ...
- *Последствие 1*
- *Последствие 2*
- ...
- Ослабление 1
- Ослабление 2
- ...

Подготовка проекта

- **Наименование риска:** Неполная оценка трудозатрат
- **Источник риска**
 - Производится только оценка трудозатрат всего проекта менеджером проекта
 - Специалисты по тестированию не привлекаются ни к проведению оценки, ни к ревью получившейся оценки
- **Последствия риска**
 - Недостаток ресурсов тестирования
 - Недостаток времени для активностей тестирования
- **Ослабление**
 - Привлекать тестировщиков для ревью трудозатрат
 - Проводить независимую оценку трудозатрат тестировщиками (методики, литература)

Типичные риски

Риски продукта:

- Нахождение критичных ошибок после ввода системы в эксплуатацию
- Нахождение «дыр» в системе безопасности
- Возникновение проблем при поставке и развертывании системы
- Использование новых технологий
- Использование продуктов третьих производителей

Типичные риски

- Планирование
 - Неправильное определение границ работ
 - Неправильный выбор архитектуры
 - Неправильная оценка ресурсов
- Организационные
 - Частое и противоречивое изменение требований заказчиком
 - Текучесть кадров

Типичные риски в тестировании (пример)

- Большие и НЕУЧТЁННЫЕ затраты на перетестирование
- Использование группы тестирования как отладочного механизма
- Не определен порядок работы с требованиями
- Не определены критерии начала и завершения тестирования
- Нет единого решения по пользовательским интерфейсам
- Раннее проведение нагрузочного тестирования
- Тестирование выполняется в среде разработки
- Невозможно идентифицировать версию объекта тестирования
- Коммуникация и исправление дефектов «на лету»
- Дефекты возникают из-за неверной конфигурации системы / среды тестирования

Топ -10 рисков

1. Дефицит специалистов.
2. Нереалистичные сроки и бюджет.
3. Реализация несоответствующей функциональности.
4. Разработка неправильного пользовательского интерфейса.
5. “Золотая сервировка”, перфекционизм, ненужная оптимизация и оттачивание деталей.
6. Непрерывающийся поток изменений.
7. Нехватка информации о внешних компонентах, определяющих окружение системы или вовлеченных в интеграцию.
8. Недостатки в работах, выполняемых внешними (по отношению к проекту) ресурсами.
9. Недостаточная производительность получаемой системы.
10. “Разрыв” в квалификации специалистов разных областей знаний.

Барри Боэмом [Boehm, 1988]

Предположим, что вы работаете тест-менеджером над проектом разработки нового продукта. Никаких письменных требований или спецификаций на дизайн не будет представлено, но у вас есть доступ к основным заинтересованным сторонам продукта и проекта, включая некоторых первых пользователей, которые взяли на себя обязательство приобрести продукт, когда он будет завершен. Предположим, что функциональность, надежность, производительность и безопасность являются ключевыми качественными характеристиками этого продукта. Проект придерживается гибкой модели жизненного цикла, которую он адаптировал для своих собственных целей.

Какие стратегии тестирования можно использовать для этого проекта?

- a. Тестирование на основе рисков
- b. Тестирование на основе требований
- c. Реактивное тестирование (*Реактивная стратегия -тесты разрабатываются и реализуются только после поставки реального программного обеспечения. Таким образом, тестирование основано на дефектах, обнаруженных в реальной системе.*)
- d. Тестирование на основе модели
- e. Тестирование на основе процессов
- f. Автоматизация функционального регрессионного тестирования

Оценка и отчетность по тестированию проекта

Предположим, что вас только что повысили до старшего менеджера тестирования.

Для тестирования различных частей проекта назначены шесть групп тестирования. Четыре группы предназначены для тестирования отдельных компонентов и выполнения интеграционного тестирования между компонентами. Руководители этих групп отчитываются перед вами. Из двух других групп одна - отвечает за системное тестирование, а другая - за приемочное тестирование (UAT). Менеджеры этих двух команд подчиняются вам напрямую.

Выберите три из следующих параметров, которые будут наиболее эффективными для менеджеров тестирования компонентов для управления своей частью тестового проекта.

- a) Динамика процента покрытия требований тестами.
- b) Число не устранённых дефектов в сценариях пользователей
- c) График количества дефектов, обнаруженных каждым членом группы тестирования компонентов
- d) Тенденция соотношения между открытыми и закрытыми дефектами в расчете на компонент
- e) Соотношение количества дефектов, обнаруженных при тестировании компонентов, и дефектов, не обнаруженных при тестировании компонент, и обнаруженных при системном тестировании
- f) Динамика процента выполненных интеграционных тестов со статусом Pass
- g) Динамика процента выполненных тестов end-to-end со статусом Pass

Участники команды. Роли и сферы ответственности.



Обзор ролей в тестировании

ОСНОВНЫЕ РОЛИ

- Тест-менеджер, менеджер проекта по тестированию
- Тест-аналитик
- Тест-дизайнер
- Тестировщик, Инженер по тестированию

ВСПОМОГАТЕЛЬНЫЕ РОЛИ

- Администратор тестовой системы, приложений поддерживающих жизненный цикл тестирования
- Администратор баз данных, менеджер баз данных
- Разработчик тестов

Зачем нужно понимать какие роли куда относятся?

- Менеджеру нельзя заниматься вспомогательной деятельностью
- Специализация даёт прирост в эффективности: выделяйте роли в команде
- Специализаций и роли – основа перехода к процессному управлению
- Роли – первые «горизонтальные» ступеньки для роста ваших людей

Роль	Описание
Тест-менеджер, менеджер проекта по тестированию (Test Manager, Test Project Manager)	Производит управленческий контроль (management oversight) Ответственность: <ul style="list-style-type: none"> • Обеспечивает техническое направление • Получает необходимые ресурсы • Обеспечивает управленческую отчётность
Тест дизайнер (Test Designer)	Определяет, приоритизирует и обеспечивает разработку тестовых случаев Ответственность: <ul style="list-style-type: none"> • Разрабатывает план тестирования • Разрабатывает модель тестирования • Оценивает эффективность тестирования
Тестировщик, Инженер по тестированию (Tester)	Выполняет тесты Ответственность: <ul style="list-style-type: none"> • Выполняет тесты • Фиксирует результаты • Восстанавливает тесты и систему после сбоев • Документирует запросы на изменение
Администратор тестовой системы, приложений поддерживающих жизненный цикл тестирования (Test System Administrator)	Обеспечивает управление и поддержку тестовых окружений и данных Ответственность: <ul style="list-style-type: none"> • Администрирует систему управления тестированием • Инсталлирует и управляет доступом к тестовым системам
Администратор баз данных, менеджер баз данных (Database Administrator, Database Manager)	Обеспечивает управление и поддержку тестовых данных (баз данных) Ответственность: <ul style="list-style-type: none"> • Администрирует тестовые данные (базы данных)
Тест-дизайнер (Designer)	Устанавливает и определяет операции, атрибуты и связи тестовых классов Ответственность: <ul style="list-style-type: none"> • Устанавливает и определяет тестовые классы • Устанавливает и определяет тестовые наборы (пакеты)
Разработчик тестов (Implementer)	Разрабатывает юнит тесты (unit tests), тестовые классы и тестовые наборы (пакеты) Ответственность: <ul style="list-style-type: none"> • Создаёт тестовые классы, собирает тестовые пакеты и интегрирует их с тестовой моделью

Senior

- Проверка требований (обсуждение с FA)
- Стратегия тестирования (Тест-план)
- Обсуждение с девами (просмотр кода?)
- Изучение (или знание) стандартов (ISO 9126)
- Оптимизация (методологии тестирования)
- Использование прошлых наработок (Q-pattern)
- Написание тест-кейсов

Middle

- Подготовка тестового окружения
- Настройка серверов
- Написание AUTs
- Автоматизация (создание и поддержка)
- Поиск вспомогательных программ (BOM, AutoIT)

Junor

- Выполнение тест-кейсов
- Оформление багов
- Создание баг-репорта
- Баг-верификация
- Постоянное отслеживание новых дефектов

"Дорогой" сотрудник должен делать "дорогую" работу.

Как только это правило нарушается – фирма теряет деньги. Если "дешёвый" сотрудник делает "дорогую" работу – фирма теряет в качестве.

Разделение труда является причиной повышения общей производительности труда организованной группы специалистов за счет:

- Выработки навыков и автоматизма совершения простых повторяющихся операций
- Сокращения времени, затрачиваемого на переход между различными операциями

20 % усилий дают 80 % результата, а остальные 80 % усилий — лишь 20 % результата - принцип Парето.

T-shaped специалист

T-shaped специалист – это человек, который является экспертом как минимум в одной области, но при этом разбирается во многих других и может свободно поддерживать общение с другими специалистами на базовом уровне.



I – shaped
Эксперт только в
одной области



Специалист широкого
профиля. Разбирается во
многих областях, но не
является экспертом ни в
одной



T-shaped

T-shaped специалист

Analyst	Programmer	Test Engineer	Web Designer	System Engineer
Write Executable Documents	Write Unit Test Code (xUnit)	Write Automated Tests	UX Design	DevOps
Requirements Engineering	Write Production Code	Functional Testing	Java Script, HTML, CSS, LeSS	Python, Perl, Go, shell
Write User Manuals	Design System Architecture, DB	Write Test Plan	Image, Icon, Logo Design	System and OS
...

T-shaped специалист

Широкий

Analyst	Programmer	Test Engineer	Web Designer	System Engineer
Write Executable Documents	Write Unit Test Code (xUnit)	Write Automated Tests	UX Design	DevOps
Requirements Engineering	Write Production Code	Functional Testing	Java Script, HTML, CSS, LeSS	Python, Perl, Go, shell
Write User Manuals	Design System Architecture, DB	Write Test Plan	Image, Icon, Logo Design	System and OS
...

Глубокий

Сравнительная характеристика централизованной группы и локальной

Преимущества централизованной группы:

- Гибкость такой команды, возможность быстрее переключить тестировщика с одного проекта на другой и выработка общего подхода к тестированию.
- Тестировщики могут рецензировать тестовые сценарии, результаты тестирования, подготовленные коллегами.
- Укрепляется боевой дух.
- Дает возможность для специализации, совершенствования навыков и карьерного роста.
- Наличие отдельной относительно независимой группы тестирования дисциплинирует. Требования будут написаны, дизайн будет задокументирован, никто не станет без необходимости менять интерфейсы не согласовав с другими заинтересованными лицами.

Недостатки:

- Проекты становятся зависимыми по ресурсам, т.е. задержки в одном начнут сказываться на другом;
- Система управления усложняется, больше накладных расходов на управление;
- Может ухудшиться взаимодействие разработчиков и тестировщиков;
- Потребуются более квалифицированные сотрудники (чтобы их можно было перераспределять между проектами) - нужно налаживать процесс обучения;

Локальные команды

Преимущества:

- Лучшее знание предметной области.
- Нет зависимости между проектами по ресурсам.
- Нет размытой ответственности за сроки и качество релизов.
- Проще система управления

Недостатки:

- Дополнительные сложности при переброске тестировщиков с проекта на проект
- Меньше возможности для развития и карьерного роста

Задачи руководителя

- Сопровождение с руководством и подчиненными
- Общение с заказчиком
- Проверка работы подчиненных
- Обсуждение спорных вопросов с руководством и подчиненными
- Постановка задач подчиненным
- Планирование задач подчиненных
- Ведение проектной документации
- Сбор отчетности и предоставление оценок
- Подготовка отчетов

Круг задач

- Контроль объема проекта
- Планирование и контроль выполнения задач по тестированию
- Управление персоналом
- Контроль рисков, связанных с тестированием
- Мотивация персонала

АКТИВНОСТИ

- Согласование миссии тестирования
- Получение обязательств по тестируемости продукта
- Оценка и отстаивание уровня качества
 - План развития качества продукта
- Оценка и улучшение производительности тестирования

Обязанности

- Выработка стратегии тестирования;
- Разработка планов тестирования;
- Написание и согласование документов, регламентирующих организацию работ по тестированию;
- Постановка задач, связанных с тестированием;
- Осуществление контроля за процессом тестирования;
- Координация деятельности группы тестирования с другими участниками проекта;
- Критический просмотр тестовых процедур;
- Анализ и обсуждение способов тестирования;
- Решение спорных вопросов с разработчиками;
- Сбор данных и составление отчётов о статусе тестирования и имеющихся проблемах;
- Поиск и подбор тестировщиков;
- Обучение новых сотрудников;
- Постоянное совершенствование процесса тестирования.

Формирование новой команды



Найм и интеграция сотрудников в команду

- Именно руководитель составляет план и определяет количество нужных ресурсов
- Не соглашайтесь с урезанием ресурсов «безвозмездно», потребуйте либо увеличить бюджет, либо отпущенное время, либо усилить команду
- Согласие с «безвозмездным» урезанием ресурсов подрывает ваш авторитет

Подбор исполнителей

- Персональный состав команды согласовывается с руководством
- Руководитель может обоснованно отказаться работать с кем-либо из подчиненных
- Но если руководитель работает с подчиненными и не сообщает о проблемах, то это автоматически означает, что:
 - Это не отразится на качестве работы данного работника
 - Это не отразится на сроках выполнения работ
 - Это не отразится на качестве работы самого руководителя

Составление вакансии и «портрета» кандидата

- Компания
- Проект
- Заказчик
- Технологии
- Процесс разработки
- Достижения команды
- Требования к позиции
- Перспективы

Собеседование

Обратная реакция

- Общение с кандидатом:
 - Полная честность
 - Абсолютная нейтральность
 - Подсказки
 - Прокачка на уверенность

- Что ловим:
 - Речь
 - Взгляд
 - Жестикуляции
 - Положение тела

Полезные подсказки

- Не смотреть на часы
- Комфортность общения
- Трудная ситуация
- Подсказки/логика
- Давление
- Открытость новой информации
- Изучайте психологическую реакцию

Ограничения

Взаимодействие:

- Лично
- Телефон

Длительность:

- Минимум: 15 минут
- Оптимально: 45 минут
- Максимально: 1.5 часа

Ступени интервью:

- От 1 до 5
- В среднем 3

Участники:

- От 1 до 5
- Оптимально: 1-2

Модуль 3. Выстраивание отношений в команде. Основы тайм-менеджмента.

- Создание условий работы в команде. Стили управления.
- Выстраивание отношений. Коммуникации.
- Постановка задач. SMART.
- Хронофаги –поглотители времени
- Оперативное планирование.
- Приоритет задач. Матрица Эйзенхауэра.
- Работа с почтой.

- Формирование успешной команды Что значит хорошая команда?
 - Создание условий для работы в команде
 - Методы работы с командой
 - Командный дух и как его воспитывать
 - Информационные каналы и передача информации (эффективное проведение совещаний, электронная переписка)
 - Выстраивание отношений
 - Качества хорошего лидера
-
- Тайм-менеджмент для руководителя
 - Эффективные коммуникации
 - Авторитет тестирования в компании

Формирование успешной команды. Что значит хорошая команда?

Лидер команды говорит о целях команды и организации, команда предлагает способы их достижения, решения проблем, и лидер активно призывает ее к этому, хвалит любые идеи, таланты участников, успехи команды усердно отмечаются;

участники расспрашивают друг друга о личной жизни, шутят, оценивают работу друг друга, принимают критику, говорят о том, что им нужно и нравится;

лидер команды учит их активному слушанию, обратной связи и поиску компромисса, сообщает организации о нуждах команды.

Когда работник будет представлять себе, что делает его компания, что делает в ней он, возможно, сможет предложить рациональные решения теперь уже ясных задач, болеть за свое дело

КАЧЕСТВА ХОРОШЕГО ЛИДЕРА

- На лидере ответственность за людей, которые ему подчиняются
- Ответственность за то, что делает его команда
- Не боится говорить правду
- Умеет аргументировано отстаивать свое мнение
- Всегда контролирует ситуацию и имеет представление о ходе проекта в целом
- Немного психолог, немного воспитатель, немного учитель и всегда надежная опора, как для РМ так и для подчиненных и коллег
- Умеет организовывать свой день и работу своих подчиненных, так, чтобы успевать делать все запланированное
- Всегда развивается, анализируя свои ошибки и успехи

ОШИБОЧНОЕ ПОВЕДЕНИЕ НОВИЧКОВ

- Пытается все сделать сам. («Самоделкин»).
- Рисование диаграмм Ганта (MS Project, OpenProj) – люди не линейны.
- Не четкие задачи.
- Все начинается с отбора людей.
- Конструктивная обратная связь.
- «Подражание подчиненных». Следите как Вы говорите о заказчиках, разработчиков.
- «Невидимки». Как Ваша команда выглядит в глазах других людей (желательно в глазах вышестоящих менеджеров).
- Жадность. Думают только о своей команде.
- Не учатся.
- Обсуждать проблемы нужно с теми кто их может решить.

Делегирование задач. Наставничество, развитие сотрудников их мотивация

Делегирование задач

- Почему зачастую руководитель намного более занят, чем подчиненный?
- Если вы помогаете решать проблему вашего подчиненного, она не становится вашей
- Избегайте «размывания» ответственности
- Избегайте коллективной ответственности
- Учитывайте индивидуальность

Авторитет тестирования в компании

Коммуникация «внутри»

Общение со смежными отделами



Чего от нас ждут, как от менеджеров?

Исходя из того, что мы люди здоровые ☺

- Что нужно руководству компании?
- Что нужно вашему заказчику?
- Что нужно вашему ПМ-у?
- Что нужно вашему ведущему разработчику?

- Оказывайте проекту сервис, а не будьте помехой. Вы поставляете информацию, а не насаждаете процессы ради процессов.
- Предоставьте менеджерам информацию, которая им требуется для принятия решений, а затем позвольте им принять эти решения.
- Помните, что продукт не обязательно должен соответствовать вашему стандарту качества.
- Проблемы, которые замедляют тестирование, крайне важны, поскольку из-за них долго не обнаруживаются ошибки, и найти их становится все труднее. Поэтому сообщайте не только об ошибках в продукте, но и о проблемах, которые замедляют тестирование.
- Покажите, что ваша работа – это исследование продукта, требующее знаний, и разъясните менеджерам, как мы на самом деле обнаруживаем проблемы.
- Обратите внимание (поскольку это обычное дело), почему тестирование занимает столько времени – как мало времени вы тратите на собственно тестирование продукта, и как много времени вы тратите на исследование ошибок и отчетность, работы по настройке, совещания, решение организационных вопросов и другие работы, чтобы обеспечить тестовое покрытие.

Создание условий для работы в команде

- Действия должны обладать тремя характерными признаками WWW = Who, What, When.

Люди делятся на визуалов, аудиалов, кинестетиков и дискретов. Причем большинство людей – визуалы, то есть лучше всего воспринимают информацию глазами.

В ситуациях с противоречиями, конфликтами между Вашими подчиненными нужно помнить одно правило: “ПО ВОЗМОЖНОСТИ, никогда не выступать третейским судьей для Ваших подчиненных. Либо они договорятся САМИ между собой, либо обоим будет плохо.”

Методы работы с командой

Командный дух и как его воспитывать

- Вы не можете наблюдать за всем. То, за чем вы должны наблюдать обязательно – это персонал. Люди должны знать, что вы не потерпите плохой работы.
- Всегда пытайтесь обсудить внутреннюю поддержку на самом нижнем уровне. Вам нужна поддержка людей, выполняющих непосредственную работу и лучший путь её получить непосредственно в обсуждениях.
- Если кто-то не смотрит, не спрашивает, не анализирует, то попросите его уйти.
- Рабочее время персонала очень важно. Вы должны быть внимательны как менеджер, понимающий значение других людей и ценящий их время (то есть поручаемая работа и организуемые совещания должны быть действительно необходимы). Там, где это возможно, вы должны оградить персонал от ненужной работы (например, можно игнорировать некоторые запросы или их инициатору можно направлять отказ).

Все люди – разные

Существует множество способов классификации психологических типов. Классификация (с точки зрения отношения к работе):

- «Пунктуальный». Не пренебрегает деталями, но всегда занят и не способен расставить приоритеты
- «Хочу понравиться всем». Работает старательно и добросовестно, но не умеет говорить «нет» и часто перегружен чужой работой
- «Всегда опаздываю». Способен справиться с большим объемом работ в краткие сроки, но всегда откладывает ее до последнего – иначе ему неинтересно
- «Самодостаточный». Ответственен, но избегает обращаться за помощью и советами, из-за чего часто затягивает даже простые задачи
- «Творец». Способен найти нетривиальный подход, эффективно решать сложные задачи, но часто игнорирует планы и бизнес-потребности, избегает рутинной работы

Не пытайтесь научить «всему и сразу»

Чрезмерное внимание множеству нюансов приводит к сильной информационной перегрузке и усложняет в разы усвоение основ, на которых в будущем будет строиться понимание специфики отдельных областей специальности.

Твёрдое усвоение необходимого минимума является более приоритетным, чем увлекательный рассказ о «вершках» множества аспектов профессии

Три фактора, необходимых для мотивации сотрудника:

Ответственность за задачу

SMART

Инструменты и знания

Для того, чтобы выполнить этот пункт, убедитесь что исполнитель:

обладает достаточными знаниями и умениями для выполнения задачи.

обладает необходимым software, hardware и прочими, нужными для выполнения задачи, инструментами;

Регулярная отчетность

Буква	Значение	Пояснение
S	Specific (Конкретность)	Объясняется, что именно необходимо достигнуть.
M	Measurable (Измеримость)	Объясняется в чем будет измеряться результат. «Сколько вешать в граммах?»(С).
A	Attainable (Достижимость)	Объясняется за счет чего планируется достигнуть цели. И возможно ли ее достигнуть вообще?
R	Relevant (Актуальность)	Определение истинности цели. Действительно ли выполнение данной задачи позволит достичь желаемой цели.
T	Time-bound (Ограниченность во времени)	Определение временного промежутка по наступлению/окончанию которого должна быть достигнута цель (выполнена задача).

Выстраивание отношений

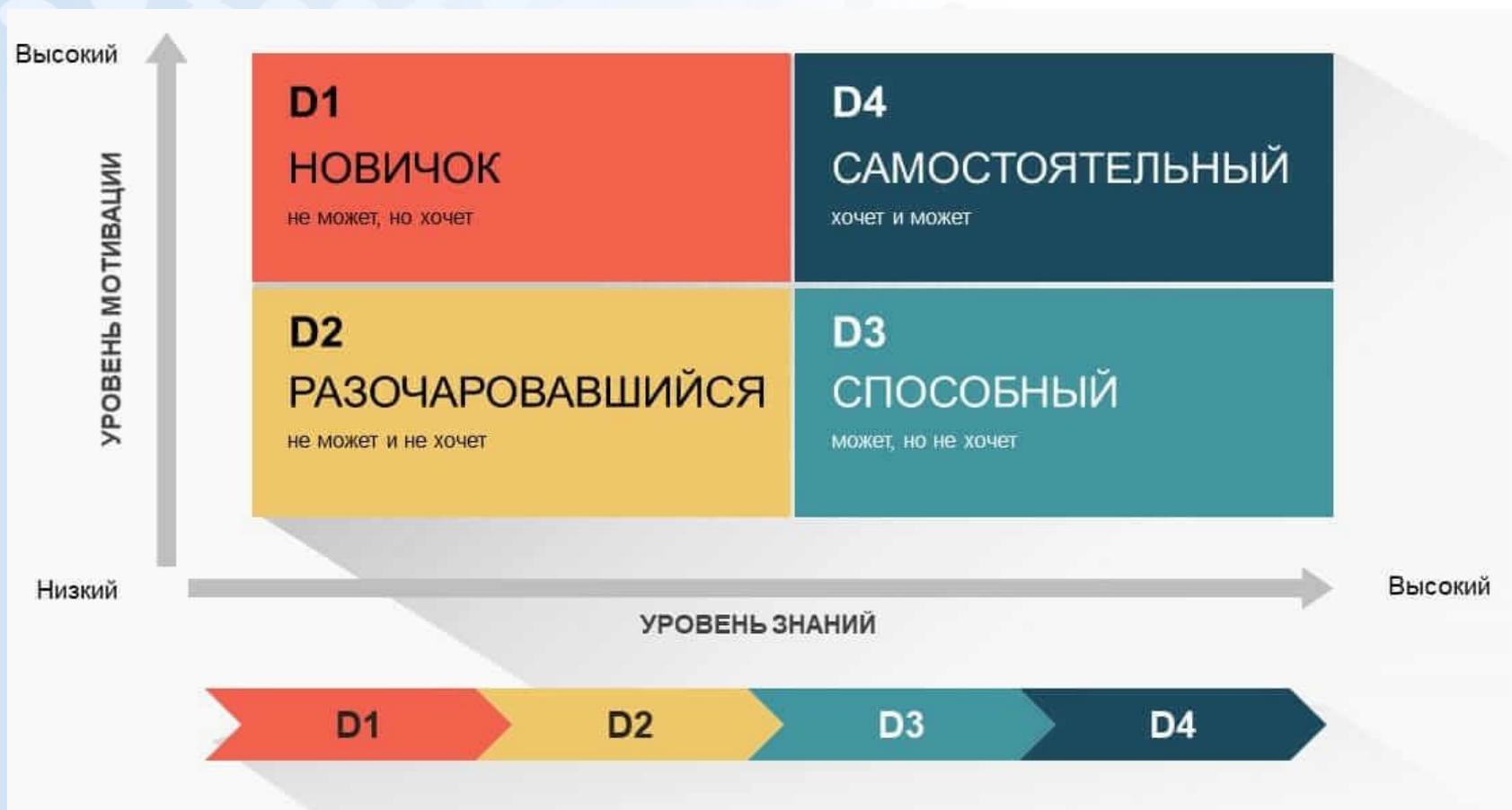
Люди не делают чего-то по 4 причинам:

- Не четкая цель (если человек новый удостоверитесь, что Вас поняли)
- Не умеют (не всегда люди готовы признать, что они что-то не умеют)
- Не могут (не успел – человек занят в нескольких проектах)
- Не хотят

4 стиля управления

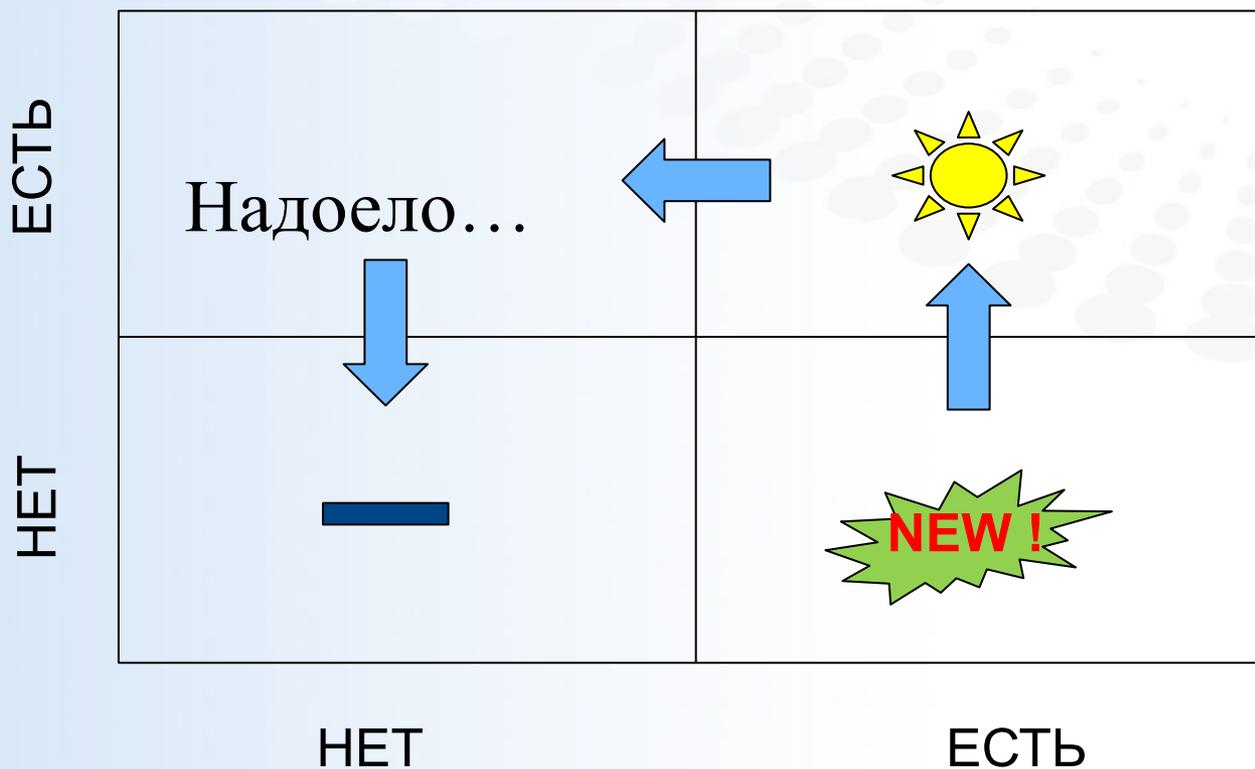


Уровни развития сотрудников



Матрица интереса

КОМПЕТЕНТНОСТЬ



ИНТЕРЕС

Матрица осознанности и компетентности

ОСОЗНАННОСТЬ



НЕТ

ЕСТЬ

КОМПЕТЕНТНОСТЬ

ОСОЗНАННОСТЬ

ЕСТЬ

Да тут все не так просто ...

Ага, вот оно как!

НЕТ

А что тут делать?



Я не понимаю почему,
но правильно вот так



НЕТ

ЕСТЬ

КОМПЕТЕНТНОСТЬ

Кто мотивирует?

Как правило, в тестировании задачи по мотивации исполняет тест-менеджер

В этом контексте его обязательные признаки:

- Право подбора исполнителей – я тебя выбрал, а не тебя назначили в мой проект
- Право распределения премий – я определяю размер твоей премии
- Право запрета выпуска версии – я не просто смотрю, я управляю проектом и качеством продукта

Как мотивировать?

Типичная схема мотивации

- **Материальная**
 - Ставка
 - Премия по результатам личной деятельности
 - Премия по результатам деятельности отдела
 - ...
- **Нематериальная**
 - Личное общение
 - Внимание и защита
 - Сопричастность
 - Влияние

Иерархия потребностей (по Маслоу)



Что это означает на практике?

Мотивирующие факторы для различных уровней (по Маслоу)

5. Клубы (кружки) качества, группа по организации процессов, подарки при значимых событиях...
4. Название должности, представительские обязанности, имидж компании, аксессуары...
3. Участие в управлении, обратная связь с руководством, обучение, информация о долгосрочных перспективах...
2. Конкурентная з/п, комфортное рабочее место и т.п.
1. Выполнение 80% требуемого объема работы оплачивается на уровне средней з/п на рынке

Гордость

Польза

Новые знания

Сложные задачи

Право голоса

Награды

Исключительность

Работать весело

Почему лидер важен для команды.

Выстраивание отношений

Одной из проблем нового руководителя является то, что все ждут от него решения своих проблем.

Текущая деятельность обычно не оставляет времени для того, чтобы вы могли думать. Вы должны выкроить время для того, чтобы «понюхать розы». При вашей работе вы должны иметь время для того, чтобы понять последствия ваших действий.

Руководитель может не знать, как должна выполняться работа, но он знает, чего он хочет.

Модуль 4. Оценка трудозатрат на тестирование.

- Определение задач, которые должны быть выполнены.
- Оценка трудоемкости задач. Эмпирическое правило Брукса. Практические соображения.
- Метод анализа видов ошибок и их влияния (FMEA). Упрощенный вариант.
- Практическая работа: оценка трудозатрат на тестирование учебного проекта с использованием упрощенного варианта FMEA.

Оценка задач

- Задачи должны оцениваться теми, кто их будет выполнять
- Проводите оценку задач совместно с подчиненными. При этом помните, что:
 - Исполнители всегда называют неверные сроки
 - Понять, насколько ошибается конкретный исполнитель можно лишь методом проб и ошибок
 - «Силовые» методы как средство борьбы с неправильными оценками не работают

Оценка трудозатрат на тестирование

1. Определение задач, которые должны быть выполнены. На этом этапе вполне достаточно разбить работу на задачи.
2. Оценка трудозатрат на решение отдельных задач и всего ЖЦ тестирования. Трудозатраты представлены в виде произведения количества исполнителей на затраченное ими время.
3. Определение времени, требуемого для решения каждой задачи и длительности всего ЖЦ тестирования.
4. Построение подробного расписания и поэтапного графика каждой тестовой задачи.
5. Оценка рисков невыполнения графика работ и формулировка планов их снижения.

Планирование работ по тестированию, оценка трудозатрат

1. Изучение тестовой основы (требования, ТЗ, проекты разработчиков и т.д.);
2. Выделение объектов тестирования.
3. Для каждого объекта определяется вид тестирования.
4. По каждому объекту выписывается список функций (уровень детализации не очень высокий)
5. Определение на каких стендах нужно будет тестировать
6. На основе всего этого делаем вывод о трудоемкости тестирования с учетом возможных рисков и собственного опыта.

О чём часто забывают

При оценке не забывайте о «скрытых» задачах и «запасе прочности»

При составлении планов и оценке учитывайте риски

...слабо развиты наши методы оценок. В сущности, они отражают молчаливое и совершенно неверное предположение, что все будет идти хорошо.

Фредерик Брукс

Как правило, план не выполняется из-за неучтенных задач или рисков

Оценка трудоемкости задач

COCOMO II (Constructive Cost Model – конструктивная модель стоимости)

Вот базовые уравнения COCOMO:

Трудоемкость = $a_b(KSLOC)^{b_b}$ [человеко-месяцев]

Срок разработки или длительность = $c_b(\text{Трудоемкость})^{d_b}$ [месяцев]

Число разработчиков = Трудоемкость / Срок разработки [человек]

Тип проекта	a_b	b_b	c_b	d_b
Органический (маленькие команды с хорошим опытом работы и не жесткими требованиями к разработке)	2.4	1.05	2.5	0.38
Полуразделенный (средние по размеру команды со смешанным опытом разработки и со смешанными требованиями (как жесткими, так и нет).	3.0	1.12	2.5	0.35
Встроенный (разрабатываются с учетом множества жестких ограничений (по аппаратному, программному, операционному обеспечению и т.д.)	3.6	1.20	2.5	0.32

Эмпирическое правило Брукса

«В течение ряда лет при планировании разработки программного обеспечения я пользуюсь следующим эмпирическим правилом:

- 1/3 — планирование,
- 1/6 — написание программ,
- 1/4 — тестирование компонентов и предварительное системное тестирование,
- 1/4 — системное тестирование при наличии всех компонентов.»

Практические соображения

- Помните о правиле «треугольника» **Время-Деньги-Ресурсы**. Фиксировать можно не более 2 параметров
- Если какой-либо параметр меняется более, чем на 10%, то воздействие на другие два будет **квадратичным**
- Менеджер вносит на порядок большие ошибки в оценки проекта, чем инженеры при оценке индивидуальных задач

- Подстраивайте график работы группы под нужды проекта
- Учитывайте разницу во времени между офисами и заказчиком
- Учитывайте культурные различия

Контроль и ведение проекта

Контроль

- Будьте в курсе, чем занят каждый подчиненный и когда он окончит свое задание
 - Планируйте загрузку с избытком («горизонт задач»)
- Если вы узнаете о плохой работе подчиненных от руководства – это тревожный симптом!

Информация

- Обеспечьте совместное использование (sharing) имеющегося опыта
- Избегайте возникновения «Тайных Знаний» (Secret Knowledge)
- Поддерживайте создание и развитие Базы Знаний

Если проект не укладывается в сроки, то добавление рабочей силы задержит его еще больше.

Отчеты

- Регулярный доклад о ходе тестирования.
Доклад о ходе работ должен дать ответ на вопрос "насколько далеко мы продвинулись в тестировании?".
- Отчетный доклад об обнаруженных дефектах. Предназначен для выяснения степени успешности тестовых мероприятий в контексте обнаружения дефектов. Одна из его целей состоит в получении текущей оценки качества тестируемого программного продукта.
- Доклад по результатам тестирования.
Предназначен для документирования результатов тестирования с тем, чтобы в будущем можно было получить ответ на вопросы о том, что подвергалось тестированию, какие были получены результаты, и какого мнения придерживалась группа тестирования относительно готовности продукта к выпуску.

Коммуникации и встречи

Проводите ежедневные собрания группы

- Выявляйте текущие проблемы, но не пытайтесь их решать
- Не допускайте длинных неконструктивных дискуссий и обсуждения личностей
- Помните, ежедневные собрания имеют также и мотивационную функцию (о твоих проблемах знают, твои проблемы решаются)

FMEA (Failure Mode and Effects Analysis) метод анализа видов ошибок и их влияния. Сильно упрощенный ☺

- Делим проект на отдельные компоненты .
- Анализируем риски для каждого компонента
Определяем степень вероятности по шкале от 1 до 5 наступления одного из так называемых факторов риска;

Далее, оценки по каждому критерию перемножаются. И компоненты ранжируются так:

- Риск 1 – 125: Приоритет 1 (низший)
- Риск 126 – 250: Приоритет 2
- Риск 251 – 375: Приоритет 3
- Риск 376 – 500: Приоритет 4
- Риск 501 – 625: Приоритет 5 (самый высокий)

Факторы:

1. Impact (влияние): показывает, по шкале от 1 до 5, какое влияние на пользователя окажет то, что этот компонент окажется сломан полностью или частично.
2. Probability (вероятность) : вероятность того, что пользователь будет использовать эту функцию и нарвется на ошибку в ней.
3. QA probability (вероятность, оцененная тестировщиком): вероятность того, что в компоненте будет ошибка с точки зрения опыта и знаний QA .
4. Complexity (сложность): на сколько сложно программистам будет исправить ошибку, найденную в функции и какая вероятность регрессий после это исправления (оценивается ПРОГРАММИСТАМИ)

Четыре главных принципа

1. Концентрация внимания
2. Материализация в систему
3. Снижение переключений
4. Формирование полезных привычек

Хронофаги – поглотители времени

1. Нечеткая постановка цели.
2. Отсутствие приоритетов в делах.
3. Попытка слишком много сделать за один раз.
4. Отсутствие полного представления о предстоящих задачах и путях их решения.
5. Плохое планирование трудового дня.
6. Личная неорганизованность, «заваленный» письменный стол.
7. Чрезмерное чтение.
8. Скверная система досье.
9. Недостаток мотивации (индифферентное отношение к работе).
10. Поиски записей, памятных записок, адресов, телефонных номеров.
11. Недостатки кооперации или разделения труда.
12. Отрывающие от дел телефонные звонки.
13. Незапланированные посетители.
14. Неспособность сказать "нет".
15. Неполная, запоздалая информация.
16. Отсутствие самодисциплины.
17. Неумение довести дело до конца.
18. Отвлечение (шум).
19. Затяжные совещания.
20. Недостаточная подготовка к беседам и обсуждениям.
21. Отсутствие связи (коммуникации) или неточная обратная связь.
22. Болтовня на частные темы.
23. Излишняя коммуникабельность.
24. Чрезмерность деловых записей.
25. Синдром «откладывания».
26. Желание знать все факты.
27. Длительные ожидания (например, условленной встречи).
28. Спешка, нетерпение.
29. Слишком редкое делегирование (перепоручение) дел.
30. Недостаточный контроль за перепорученными делами.



Оперативное планирование

Типы наших ежедневных задач

Жесткие

(привязанные к точному времени)

Гибкие

(не привязанные к точному времени)

Бюджетлируемые

(требующие определенного ресурса времени)

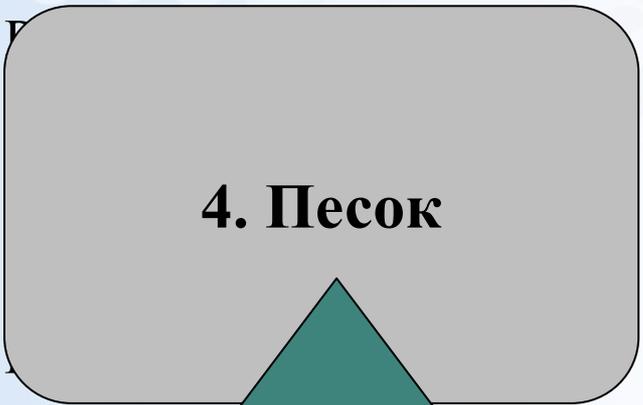
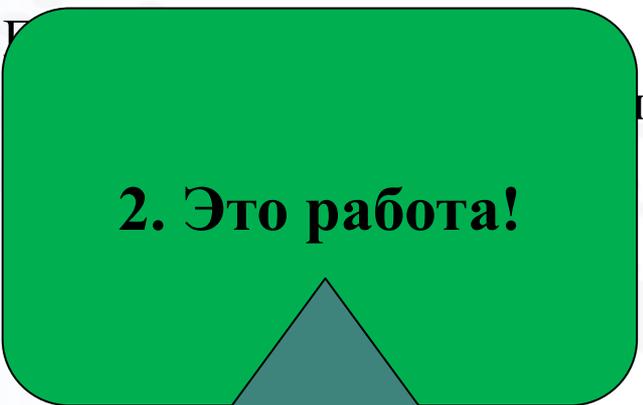
Алгоритм жестко-гибкого планирования дня

Зафиксировать в ежедневнике с привязкой к точному времени

Просто список с приоритетами

Выделить ресурс времени, но без жесткой привязки к началу исполнения

Матрица Эйзенхауэра

	МЕНЕЕ ВАЖНО	ВАЖНО
МЕНЕЕ СРОЧНО	 <p>4. Песок</p>	 <p>2. Это работа!</p>
СРОЧНО	 <p>3. Спешка</p>	 <p>1. Пожары</p>

Распределите все задачи по срочности и важности

	Менее важно	Важно
Менее срочно	D	B
Срочно	C	A

Выполняйте задачи в такой последовательности:

- **A** – срочные и важные задачи
- **B** – несрочные, но важные задачи
- **C** – неважные, но срочные задачи
- **D** – неважные и несрочные задачи

Не допускайте перехода задач из типа B в тип A

Когда её делать... или не делать?

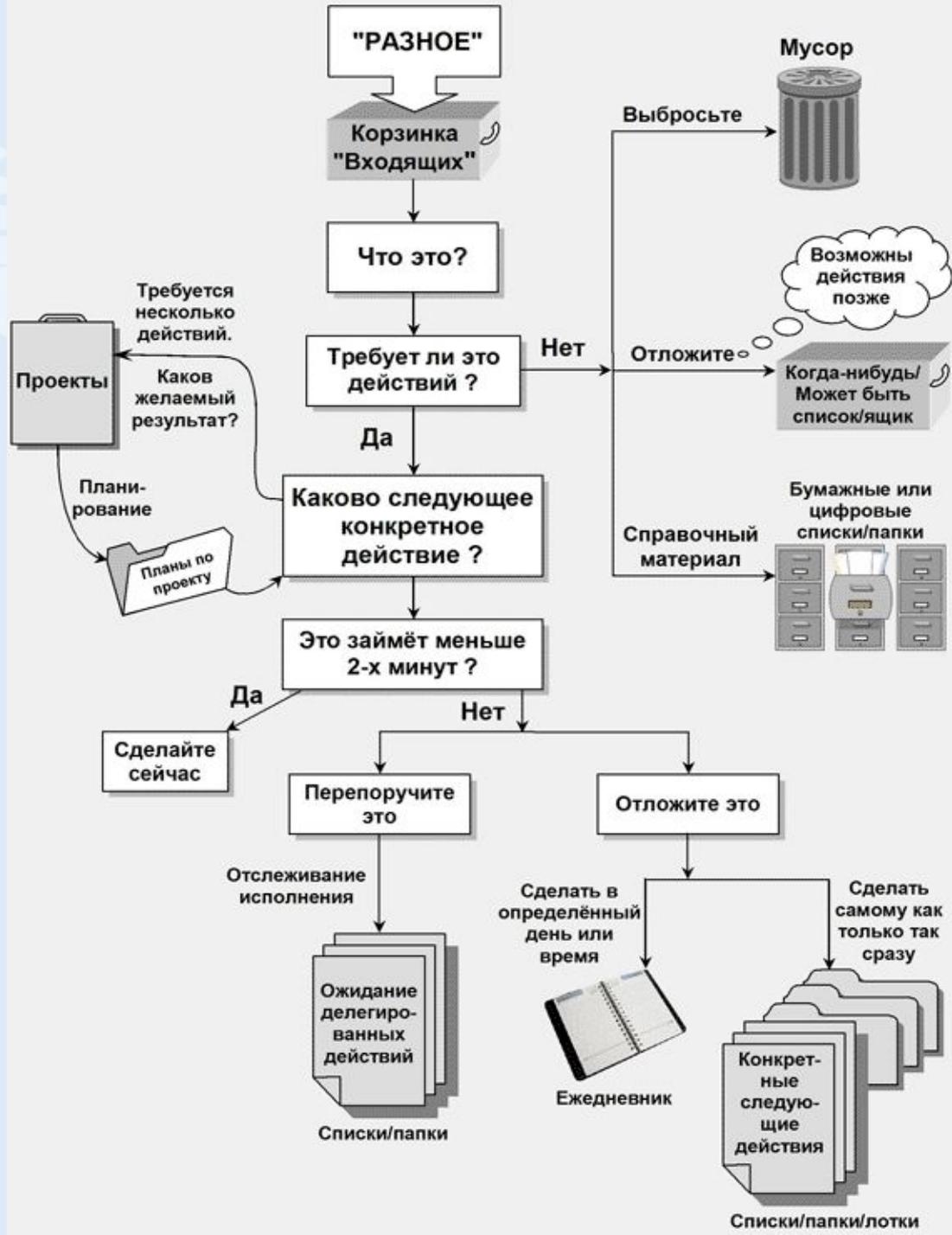
А – не допускать перехода из типа В

В – бюджетировать время в планах

С – избавляться!!!

Д – уже не делать

1. Использовать стратегии отказа от задач
2. Избавляться от поглотителей времени
3. Если есть сомнения, всегда задавать уточняющий вопрос
4. Структурировать работу с почтой
5. Всегда предлагать решения, а не высказывать проблему
6. Завершить разговор договоренностью

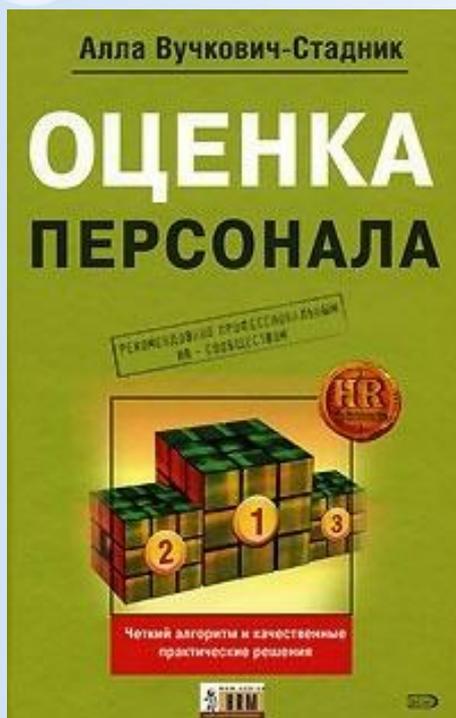


Как заставить себя умыться по утрам

Основные принципы

- Чтобы начать управлять временем требуются первоначальные затраты
- Ведите хронометраж рабочего дня. Попробуйте продержаться хотя бы 1-2 недели ☺
- Фиксируйте запланированные и незапланированные дела, помехи, эмоциональное и физическое состояние
- Не старайтесь угодить всем. Берегитесь «хронофагов»
 - Пожиратели времени, люди которые постоянно спрашивают тебя, не затрудняясь поискать ответ самостоятельно
- Имейте список целей. Без этого невозможно составлять планы, а без плана невозможно управлять временем
- Научитесь говорить «нет» и делегировать задачи

Рекомендуемая литература



- **Оценка персонала. Четкий алгоритм действий и качественные практические решения**
- Алла Вучкович-Стадник
- Издательство: Эксмо, 2008 г.
- Вы сможете грамотно выстроить систему оценки своего персонала, а ценный практический опыт автора позволит избежать многих ошибок в процессе ее внедрения.