

# Системная инженерия



**PraxOS Версия 1.0**

# СИСТЕМНАЯ ИНЖЕНЕРИЯ: ЭТО ВЫГОДНО

- Системная инженерия -- это способ уменьшить затраты за счет исключения переделок (исправлений разных типов ошибок).
- Уменьшает коэффициент **экспоненты убытков на масштабе**, поэтому зависит от масштаба проекта.

Уменьшение стоимости для

- Мелких проектов на 18% (при оптимальной доле работ системной инженерии 5%)
- Средних проектов на 38% (20%)
- Крупных проектов на 63% (33%)
- Очень крупных проектов на 92% (37%)

# Главные идеи системной инженерии

- Общий *междисциплинарный* язык, позволяющий договориться участникам проекта
- Покупка информации, уменьшающей проектные риски.
- Исправление ошибок на как можно более ранней стадии, когда это относительно дешево – идея жизненного цикла.

# Стандарты системной инженерии

- Международные стандарты (базовый стандарт -- ISO 15288 «Системная инженерия - процессы жизненного цикла систем»)
- Стандарты закупок Минобороны США, НАТО, европейских военных
- Стандарты профессиональных организаций (IEEE).
- Используются в крупных проектах: военных, строительных (туннели, мосты), атомной энергетике (ITER)

# Подход системной инженерии к управлению жизненным циклом

Системная инженерия – это гармонизация *подходов*:

- системного (назначение, границы и элементы системы)
  - процессного (деятельность и акторы)
  - архитектурного (методы описания и их группировка)
  - жизненного цикла (4D-эволюция системы)
  - оценки зрелости процессов (стадии ЖЦ процесса)
  - оценки специальных свойств системы (процессные выписки)
- *Подход (approach) - способ сущностного описания. Практики зависят от сущностного описания.*

# Системный подход

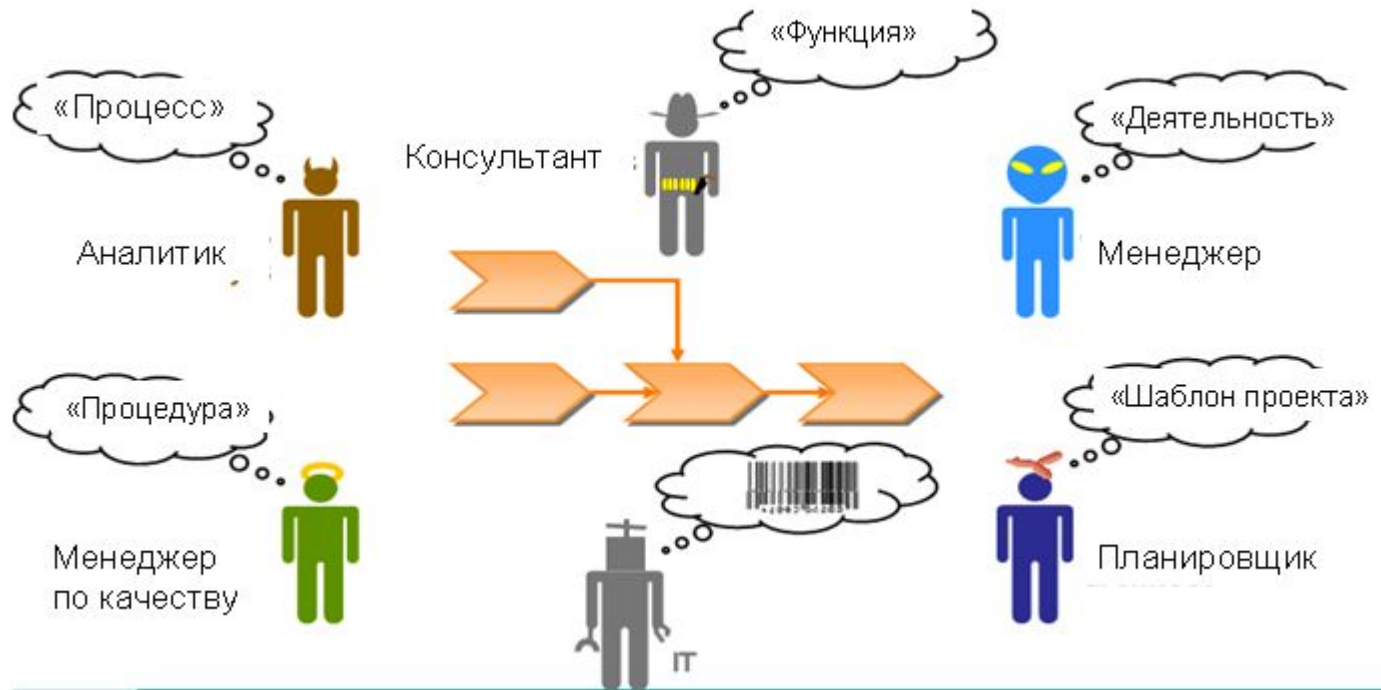
- Система имеет:
  - назначение, элементы, границу системы с окружением, связи элементов (в том числе с окружением)
  - **Описания: полное, включающее архитектурное**
  - **Стейкхолдеров (имеющих к ней интересы)**
  - **процессы, которые с ней выполняются в ходе ее жизненного цикла**
- Система никогда не бывает «вообще», система всегда конкретна (поэтому слово «система» пишется только в общетеоретических текстах, употребление слова «система» вдобавок к названию конкретной системы излишне).
- Примеры систем: АЭС, ГЭС, самолёт, процесс, информационная модель, подход. Система может включать людей и организации.

# Процессный подход

- Процесс: деятельность, разделенная на практики (элементы деятельности, activities), которые выполняют (ролевые) акторы (actors, agents, «деятели») над/с системой.
- Процессы выполняются над/с системой (система эволюционирует под действием различных процессов – это и есть ЖЦ)
- Взаимодействуют **не** процессы, а их Акторы (Акторы *организованы*).  
Описать «связи процессов» = нужно назвать акторов и транзакции между ними (подход DEMO).
- Роли описаний процессов:
  - а) as is – для анализа
  - б) to be – нормы
- процессы состоят из (по ISO TR 24774):
  - а) (под)процессов (до 3х уровней вложенности)
  - б) практик, которые сами состоят из работ



# Договориться о терминологии



По материалам компании FutureModels

Из чего состоит организация?  
Что существенно в организации?

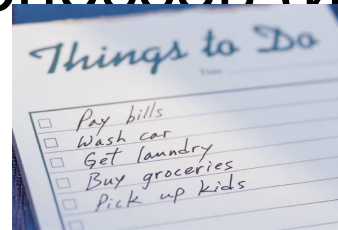


# Описание процесса (из практик)

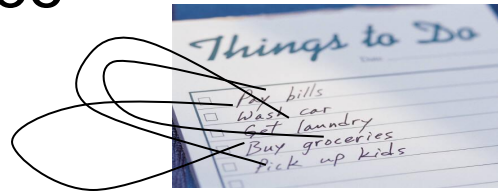
## Описание группы процессов (из процессов)

Архитектурно  
е

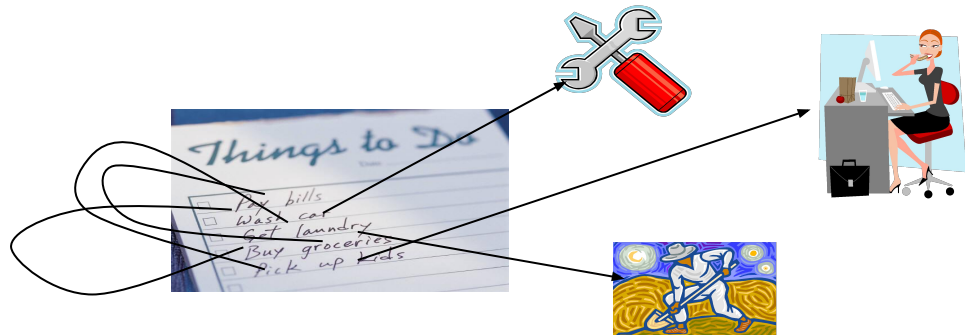
- Опорное  
(функция: что и зачем)



- Принципиальное  
(конструкция: как)



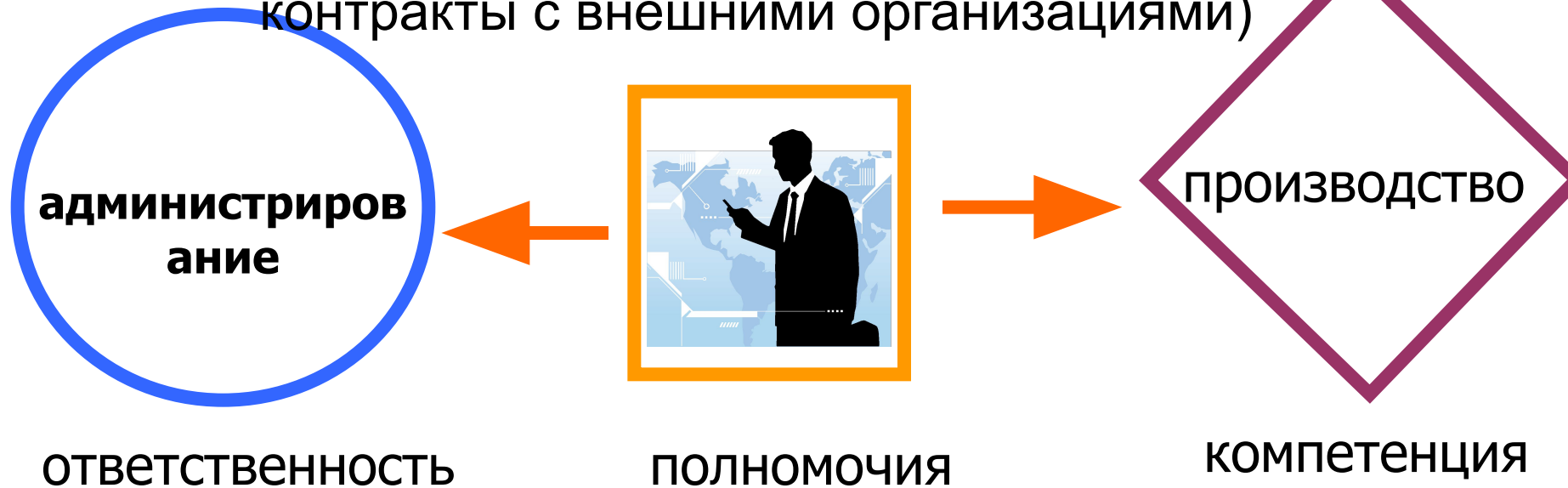
- Выполняемое  
(инструкция: норма)



- Историческое  
(измерения, отчеты, задания, прогнозы)



Внедрить процессы в жизнь  
(регламенты на уровне линейных менеджеров,  
контракты с внешними организациями)

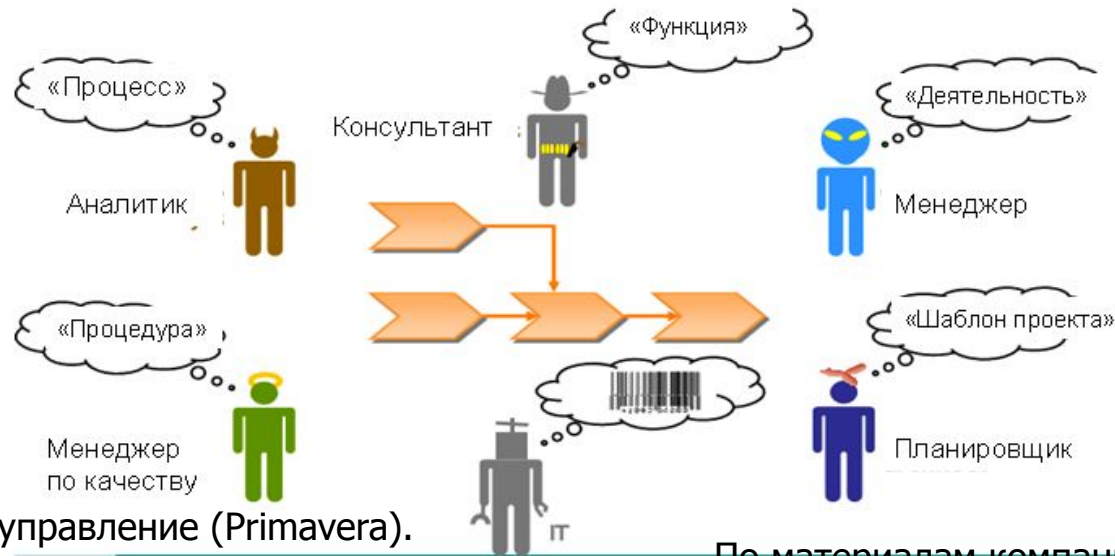


- **Администрирование – договориться, кто что кому когда делает.**
- **Производство – делать то, что надо. Не больше, и не меньше.**
- **Убедиться, что все производство администрируется (то, что надо делать совпадает с тем, что договорились делать).**
- Айтишники не могут прописать правила администрирования (не могут устанавливать полномочия).
- Айтишники не могут прописать правила производства (не могут задавать технологии проектирования и строительства).

# Приоритеты перехода к системной инженерии

- Договориться о *теориях и практиках*, которые лягут в основу процессов жизненных циклов для разных систем.
- Договориться, кто в *организации* какие *практики* будет обеспечивать.
- Сделать так, чтобы в назначенных *организационных местах* появились люди, владеющие нужными *теориями и практиками*, и эти люди эффективно *обменивались знаниями* для склеивания *расширенной организации* в одно целое.
- Дать *ресурсы* этим *знающим людям*.

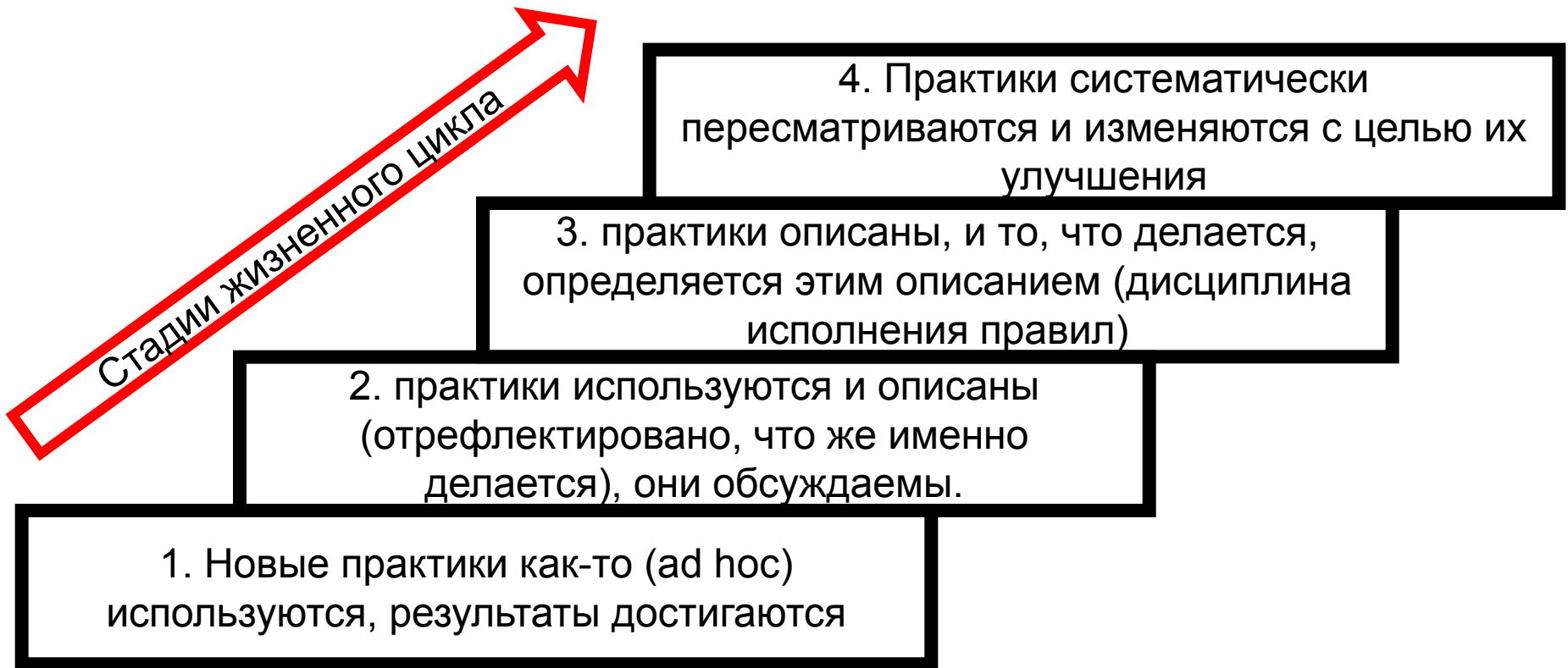
## Кто хозяин процессов на предприятии? По чьим регламентам жить?



1. Проектное управление (Primavera).
  2. Процессы/качество – стандарты серии ISO 9000.
  3. Безопасность/качество – ПОКАС, МАГАТЭ.
  4. Заложенные в купленный софт (они там невидимы, но есть)
  5. Особенности работы, определяемые специалистами (технологами).
  6. Расказанные консультантами по реинжинирингу бизнес-процессов.
  7. Стандарты системной инженерии (кто ответственный за их соблюдение?)
  8. Оставшиеся обычаи, происхождение и обоснование которых мало кто помнит.
- По материалам компании FutureModels

Айтишники правила работы не устанавливают и не могут настоять на их выполнении, но почему-то при слове «процесс» всегда зовут их.

# Жизненный цикл процесса: вверх по ступенькам зрелости



# ISO 15288: «Что делать»

## 25 обязательных процессов системной инженерии

### Обеспечения проектов

- управление описанием жизненного цикла
- управление инфраструктурой
- управление портфелем проектов (программой)
- управление персоналом
- управление качеством

### Проектные

- управление проектами
  - планирование проекта
  - управление выполнением и контроль проекта
- поддержка проектов
  - управление решениями
  - управление рисками
  - управление конфигурацией
  - управление информацией
  - измерения

### Контрактацции

- Закупка
- Поставка

### Технические

- сбор требований
- анализ требований
- архитектурный дизайн
- изготовление
- интеграция
- проверка (Verification)
- переход к эксплуатации
- приёмка (Validation)
- эксплуатация
- обслуживание
- вывод из эксплуатации

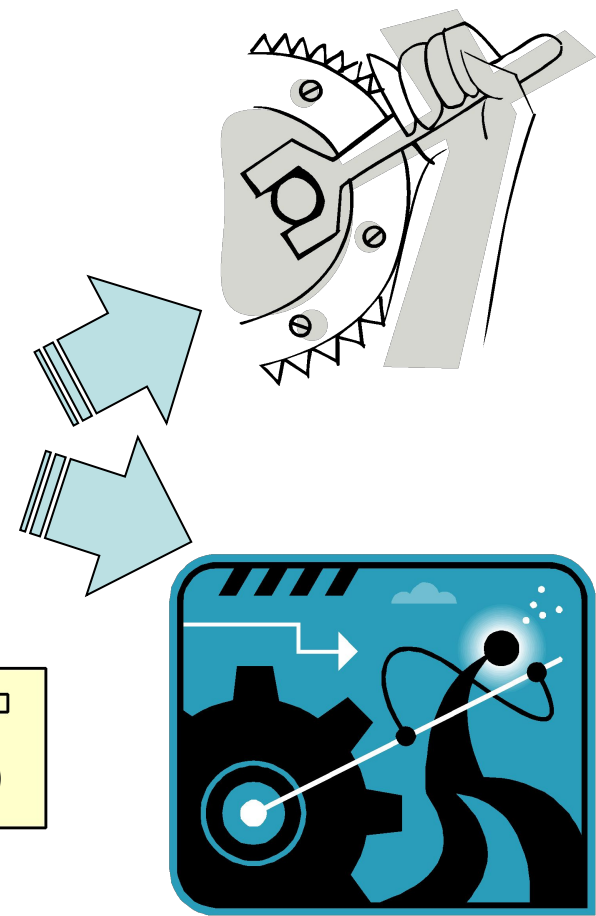
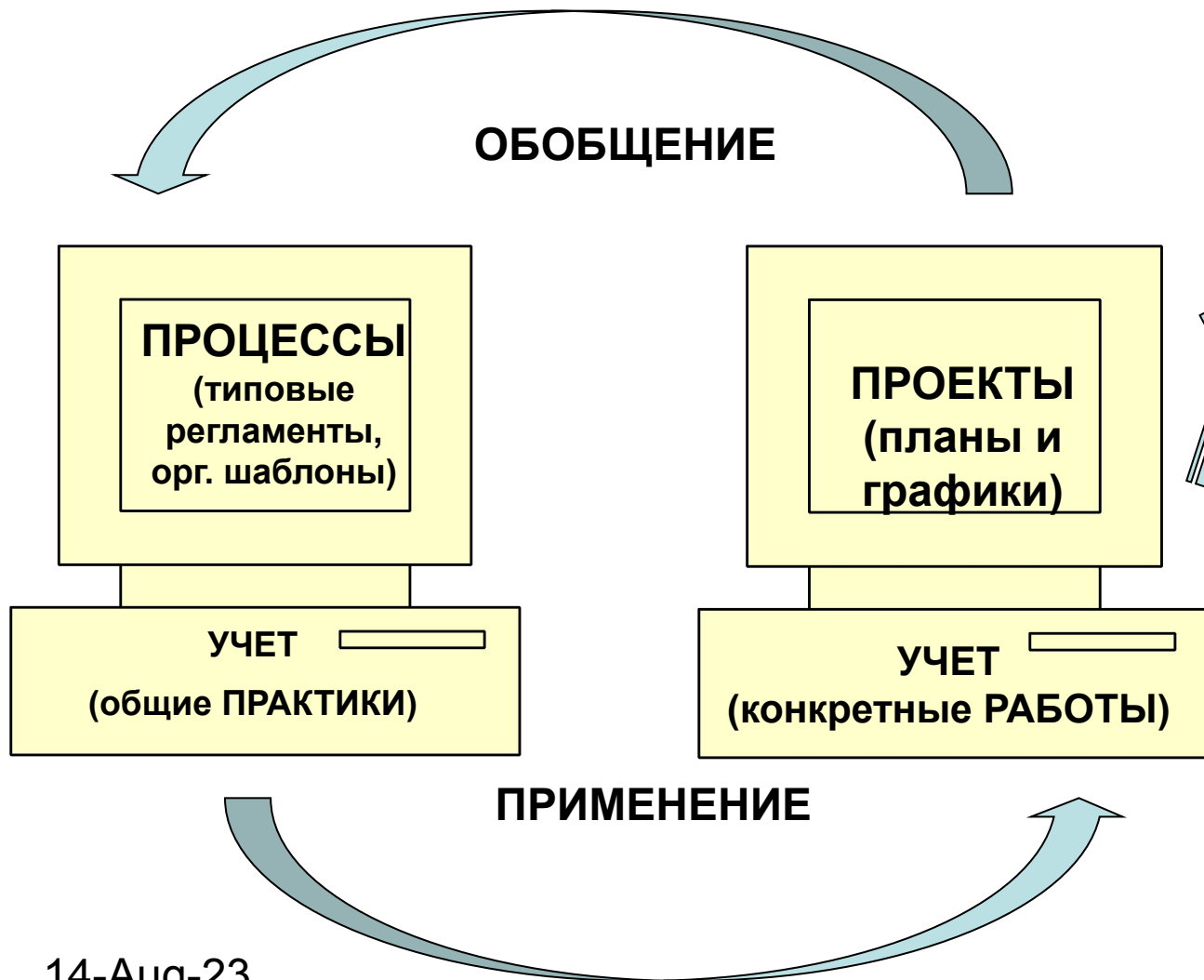


# Технологическая нейтральность практик международных стандартов

## Стандарт ISO «Жизнь»

- Каждый день ты обязан выполнять следующие процессы:
  - Есть
    - цель: чтобы была энергия;
    - результат: нет чувства голода
    - действия: 1. найти пищу. 2. положить пищу в рот
  - Работать:
    - Цель: иметь средства на «есть» и «отдыхать».
    - Результат: продукт труда покупается за деньги
    - Действия: 1. найти работу. 2. научиться. 3. выполнять выученные действия для найденной до удовлетворения клиента. 4. убедиться, что деньги заплачены.
  - Отдыхать
    - цель: чтобы не подорвать силы
    - результат: нет чувства усталости
    - Действия: выполнять отдых
- **Здравый смысл, ничего больше («лучшие практики», «хорошие практики»).**
- **Ни малейшего намека, как жить! Нужно выбрать способы (методы, технологии, инструменты) выполнения каждой практики. Нужно выбрать организацию работ.**
- **Тем не менее, очень легко проверить выполнение.**

# Процессы/организация работ и проекты



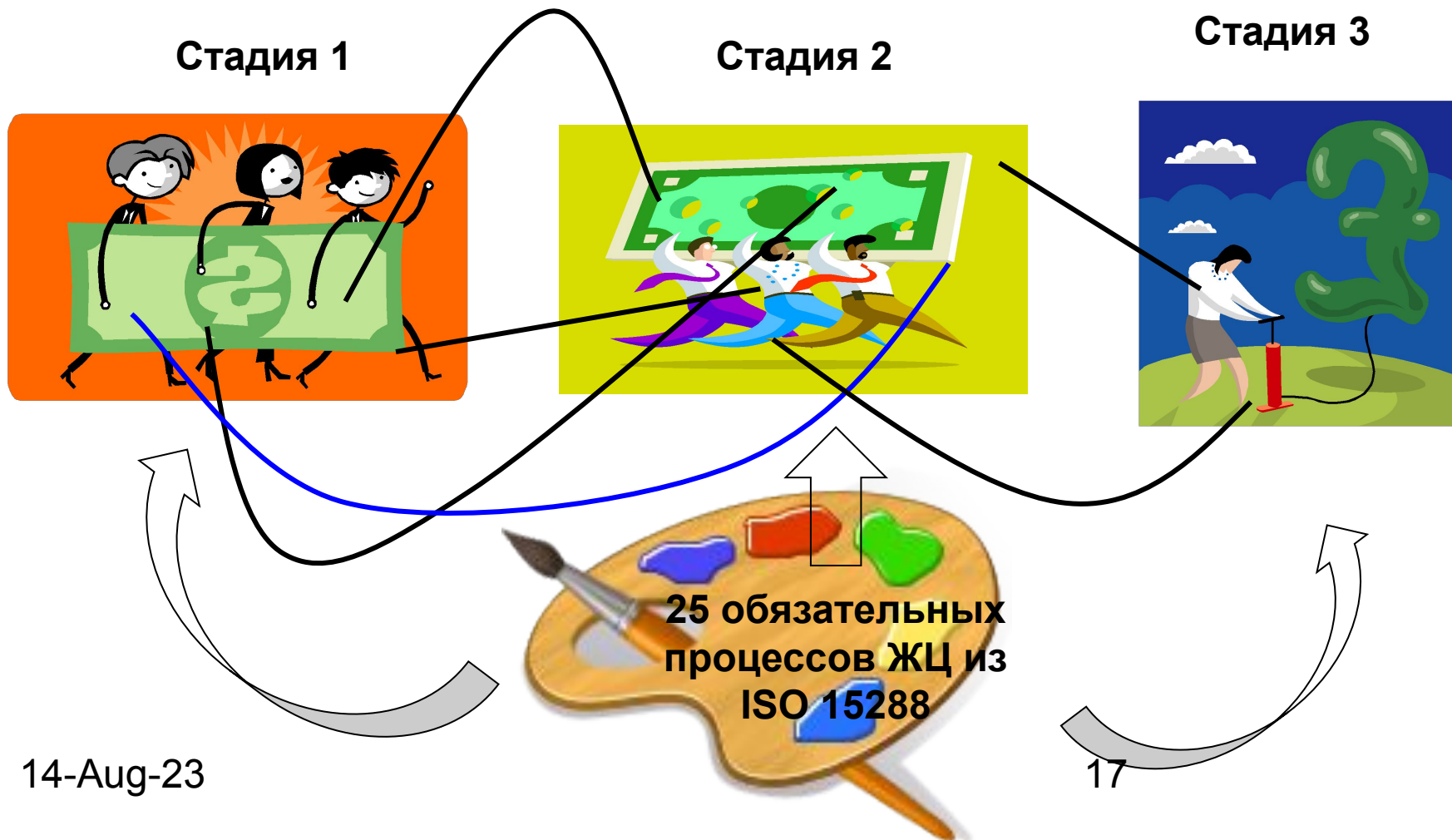


Вложенность процессов и связи между их акторами

Процесс «Управление ЖЦ X» =  $\sum$  процессов «Управление Стадией N ЖЦ X»

Процесс «Управление Стадией N ЖЦ X» =  $\sum$  «процессов ЖЦ» стадии N ЖЦ X

## Управление ЖЦ X



# Разнообразие жизненных циклов



# Вариант жизненного цикла непрерывного производства по версии ISO 15926

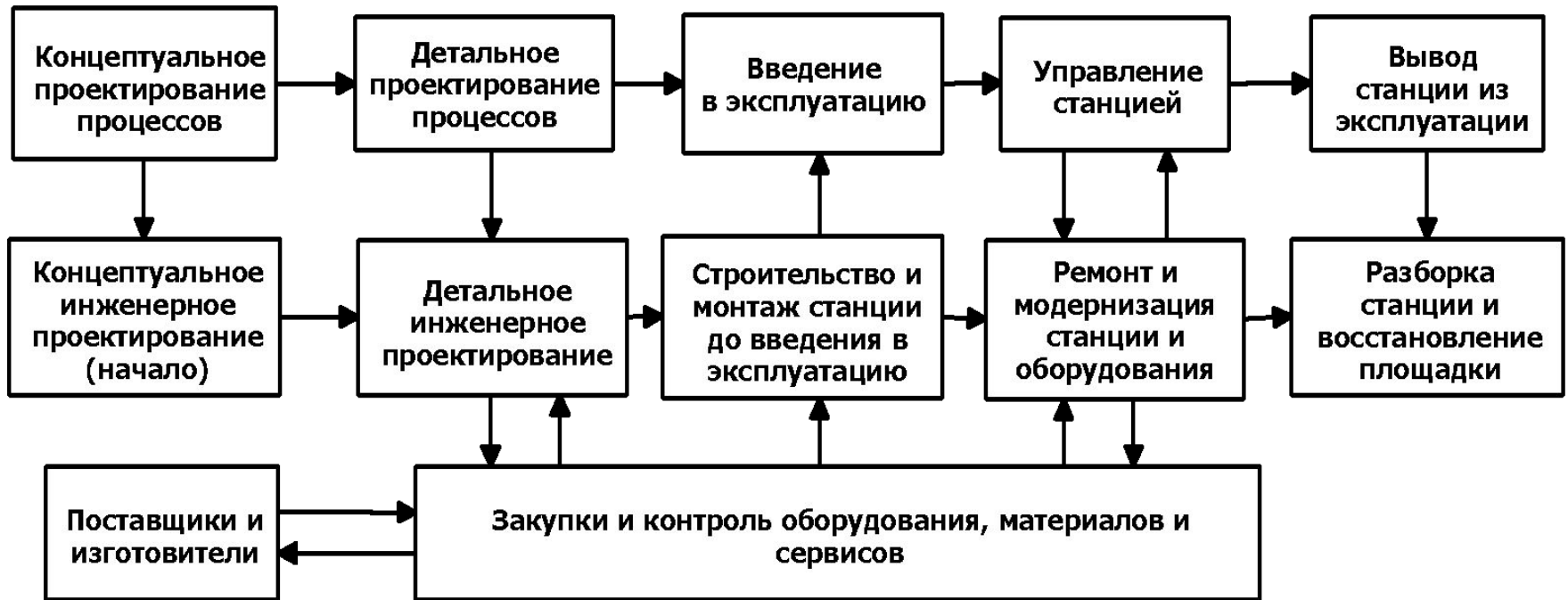


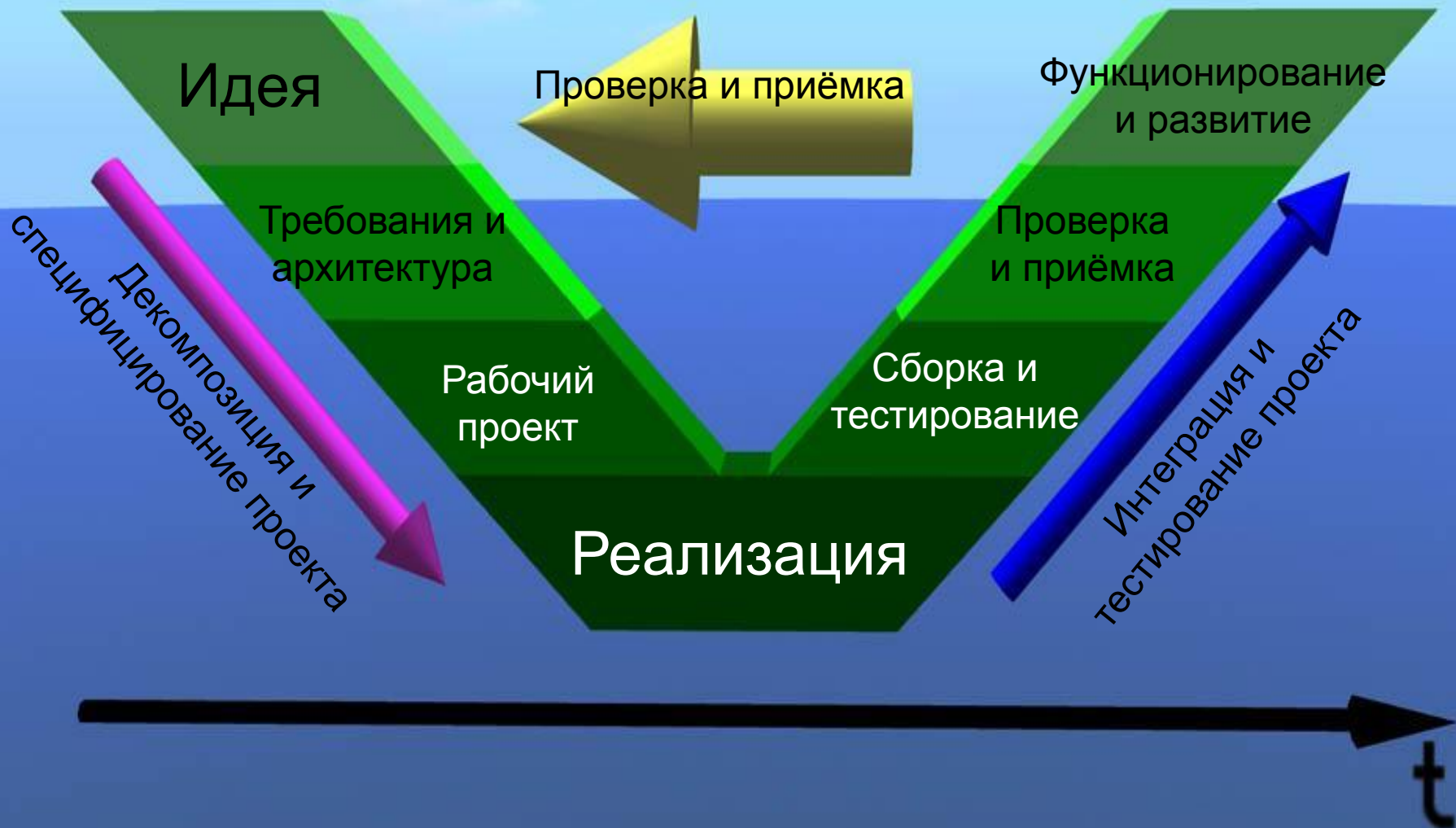
Схема из стандарта ISO 15926-1:2003г. (по схеме Process Industries STEP Consortium 1994г.)

# «Видение» консорциума FIATECH



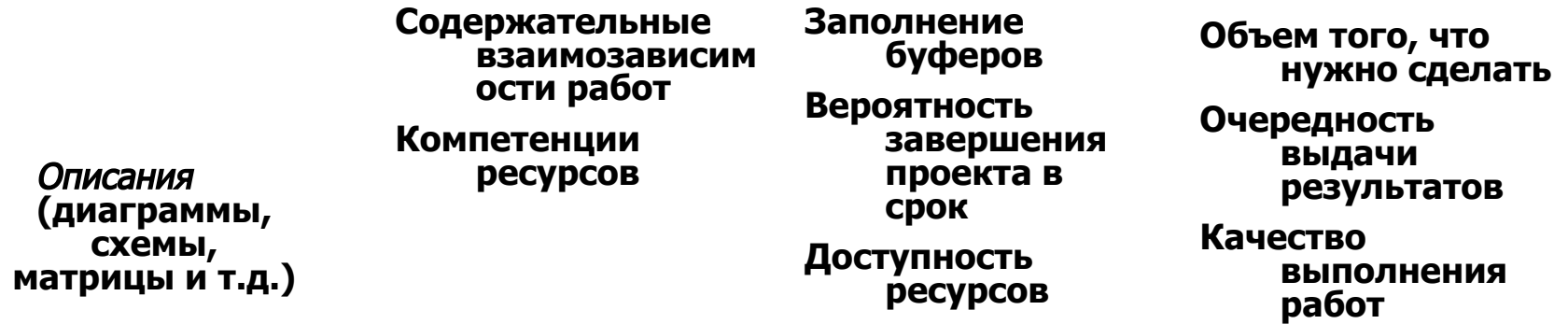
**Управление жизненным циклом и интеграция информации**

# V – модель

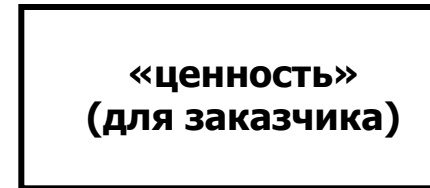


# Три основных «проектных» группы описаний

## жизненного цикла

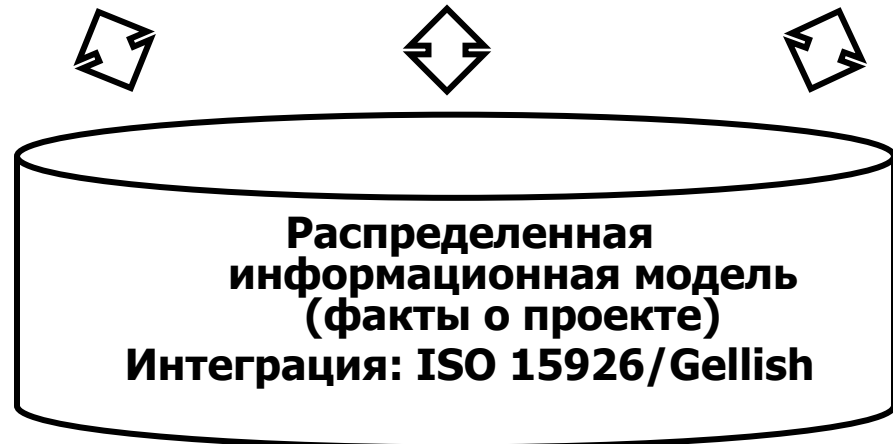


*Методы описания (нотации, софт и т.д.):*



**Организация проекта (кто кому что поручил/пообещал) не видна!**

**Должна быть еще одна точка зрения -- административная!**



# Концепция жизненного цикла

1. Общие замечания (кто и как писал Концепцию, где найти свежую версию)
2. Подход к управлению жизненным циклом (опора на какие стандарты)
3. Нормативные клаузы для каждой из основных систем:
  - Опорное описание системы, ее стейкхолдеров и их интересов
  - опорное описание процесса «управление жизненным циклом»
  - Принципиальное описание процесса «управление жизненным циклом»
  - Специальные группы описаний (безопасность, надежность и т.д.)
4. Пути реализации Концепции (основные идеи постановки процессов, необходимых для реализации):
  - Управление описанием жизненного цикла
  - Управление инфраструктурой
  - Управление персоналом
  - Управление информацией
  - Управление проектами
5. Оценка управления жизненным циклом (выбор стандарта).

# Подгонка жизненного цикла

- Жизненный цикл системы есть, даже если он не описан как «процессы». Чтобы его улучшить, нужно его обсудить. Для обсуждения его нужно описать.
- Для каждого жизненного цикла существует уникальный профиль рисков, зависящих от небольшого числа факторов:
  - Размер и сложность
  - Скорость изменений требований
  - Критичность системы
  - Доступные готовые компоненты
  - Организационные и кадровые возможности
- Одинаковый жизненный цикл для любых систем (одни и те же проверки, одни и те же методы контрактации, одни и те же предоставляемые отчеты и т.д.) не подойдет для всех систем – нужна подгонка.
- Подгонка заключается в адресном принятии практик системной инженерии, максимально отвечающим на имеющиеся риски.



# Покупка информации

- Максимальные риски – в начале проекта.
- Причина ошибок – отсутствие информации.
- Часто информацию можно купить (например, создав прототип, или проведя исследование).
- Покупать информацию нужно как можно ближе к началу работ – там она приносит максимальную пользу.

# Баланс плана и гибкости

- Гибкость: одновременная работа над требованиями, архитектурой (проектом), рабочей документацией, сооружением и т.д. – итерации в рамках каждого этапа.
- План: синхронизация различных жизненных циклов за счет прохождения комплексных проверок (результатов предыдущих этапов и планов на следующий этап), фиксация версий.
- Признать, что в жизни всегда присутствует гибкость. Но эта гибкость не отражена в нормах и правилах, поэтому ее нельзя использовать в полную силу на пользу делу.
- Для подгонки процессов жизненного цикла использовать специальные методы увеличения гибкости при сохранении плановой дисциплины, например ICM (подход приращения обязательств).

# Доказательства

- Вместо «отчета по этапу» происходит **доказательство** приемлемости рисков по соответствию требованиям -- техническим, соблюдения сроков, неперевышения бюджета и т.д..
- Форма – рассмотрение материалов «дела» независимыми экспертами (аналогия с судом).
- Происходит на границе стадий жизненного цикла.
- По итогам принимается решение:
  - Доказательств недостаточно, продолжить работу в рамках предыдущей стадии
  - Доказательств достаточно, проект закрыть ввиду неприемлемости рисков его реализации.
  - Доказательств адресуемости рисков достаточно, перейти к работам следующей стадии жизненного цикла.

# Интеграция оборудования, программного обеспечения и людей

- У оборудования, ПО и людей абсолютно разные жизненные циклы
- ПО и людей «забывают» включать в жизненный цикл системы. Это влечет огромные риски.
- Нормы и правила ориентированы на жизненный цикл оборудования.
- Методы интеграции на сегодняшний день слабо разработаны, нормы и правила их не отражают.

# Техническая платформа, «Проект»

- Что проектируется «с нуля», что перепроектируется «глядя на прототип», а что просто «перештамповывается»?
- Что выносится за скобки жизненного цикла системы? Повторноиспользуемость результатов работы – огромный резерв уменьшения стоимости системы.
- Обычно есть три разных жизненных цикла:
  - Технологической платформы
  - Конкретного «проекта»
  - Конкретной системы

# «Сетецентрические» СИСТЕМЫ ИЗ СИСТЕМ

- Объединение уже готовых систем (часто с разными собственниками и автономными жизненными циклами) в новую систему, свойства которой отличаются от свойств составляющих ее систем.
- Примеры:
  - Объединение разных предприятий для большого проекта (организационная инженерия)
  - Единая энергетическая система
- Мало методических материалов, самая актуальная тема на сегодня
- Основной механизм: стандарты

# Моделеориентированность

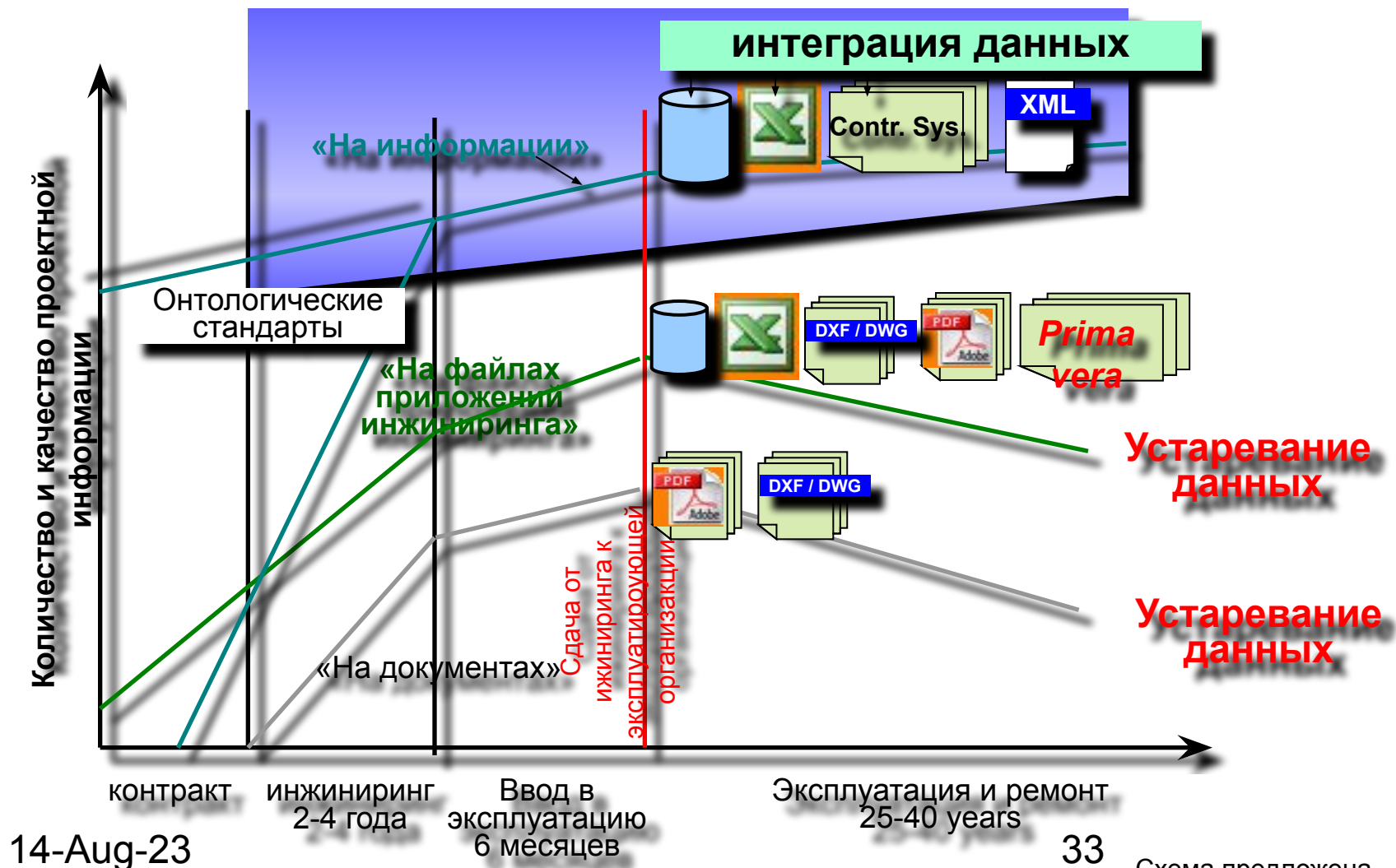
- Зачем нужна моделиориентированность:
  - уточнение представления системы для разных целей
  - связь разных представлений о системе по клику мышки (или составление «сводного отчета» по разным базам данных), за счет чего
- Это сверхвыгодно: по оценке NIST для больших проектов выигрыш до 30% от стоимости системы
- Моделиориентированность противопоставляется документоориентированности.
- Дело не в электронной форме документа, а в другом отношении к их содержанию (так, база данных – не документ, а документы для нее либо «первичка», либо «выписки»).
- Стандарты интеграции данных (прежде всего – ISO 15926).

# Структурированные данные – основа для моделеориентированного подхода





1. Общая информационная платформа ИЛИ
2. Общая интеграционная (софтверная) платформа ИЛИ
3. Общая документационная платформа.



# Моделецентрический подход к управлению информацией строительства электростанций описан в ISO 15926

- Стандартизует интеграцию данных, подготовленных в разных стандартах
- Стандартизует 4D-онтологию
- Стандартизует способ, которым приложения соединяются и обмениваются информацией (архитектуру платформы)
- Использует стандарты W3C (интернета)

# Спасибо за внимание

Анатолий Левенчук

<http://ailev.ru>

[ailev@asmp.msk.su](mailto:ailev@asmp.msk.su)

Виктор Агроскин

[vic5784@gmail.com](mailto:vic5784@gmail.com)

TechInvestLab.ru

+7 (495) 748-5388

Дополнительные материалы:

<http://www.praxos.ru>