

# Управление информационными системами

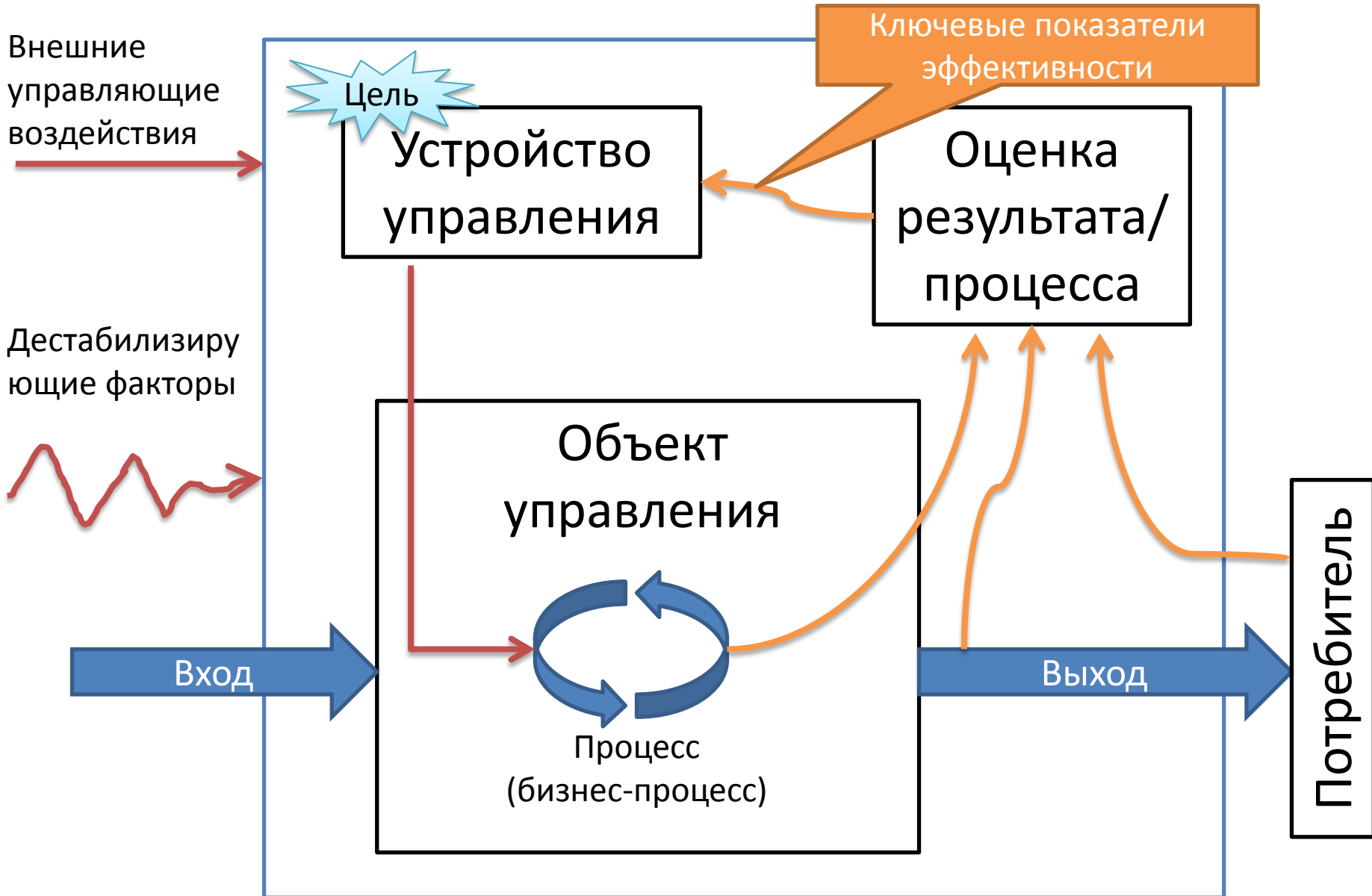
Управление ИТ инфраструктурой

# Основные вопросы

- Повторение: объект управления
- Чем управлять?
  - Состав ИТ инфраструктуры
- Как понять, что что-то не так?
  - Мониторинг
  - Системы отслеживания ошибок
- Как реагировать? Как управлять?

**ПОВТОРЕНИЕ ПРОЙДЕННОГО:  
ИДЕЯ! ИДЕ Я НАХОЖУСЯ?**

# Система управления предприятием



# Объекты управления – 1/2

- Инфраструктура ИТ
  - Техническое обеспечение
    - Сервера, персональные компьютер, системы хранения данных, сети и коммуникационное оборудование
  - Системное программное обеспечение
    - Операционные системы, инструментальные среды разработки, программы поддержки ИТ-менеджмента и средства обеспечения информационной безопасности

# Объекты управления – 2/2

- Приложения – прикладное ПО
  - Обеспечивает поддержку бизнес-процессов предприятия и функциональность отдельных автоматизированных рабочих мест
- Организационная структура службы ИТ
  - Состав подразделений, распределение между ними функций и задач
- ИТ-проекты
  - Проекты внедрения новых ИС, а также модернизацию существующих

# **ТЕХНОЛОГИЧЕСКАЯ АРХИТЕКТУРА**

# Введение

- Область архитектуры предприятия, рассматривающая «традиционные» аспекты построения ИС, которые необходимы для поддержки прикладных систем и информационных ресурсов организации
- Синонимы: платформа, инфраструктура, системная архитектура, или просто ИТ-архитектура



# Роль и место – 1/3

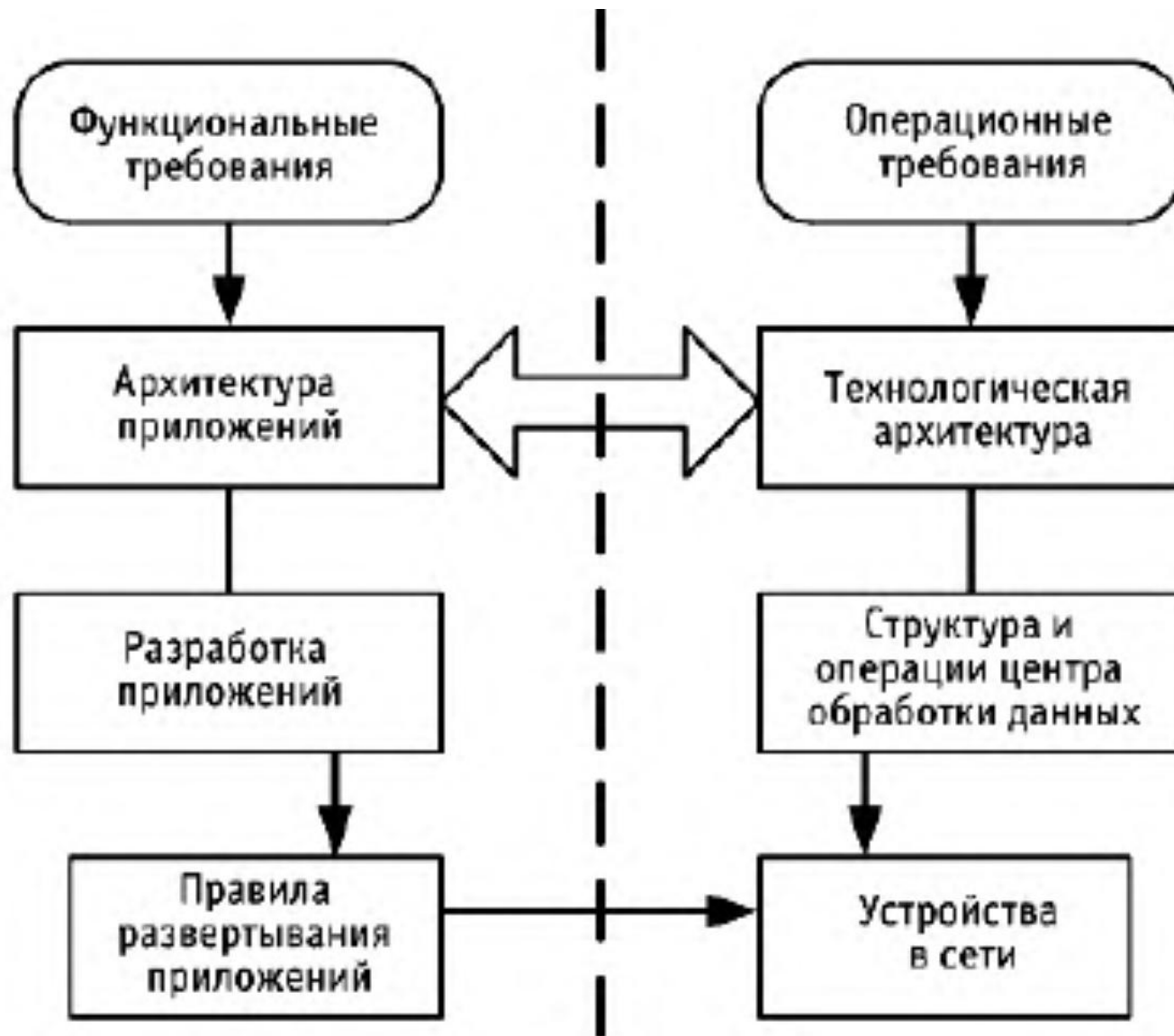
Программная архитектура

Учитывает  
поддержку  
программных  
систем

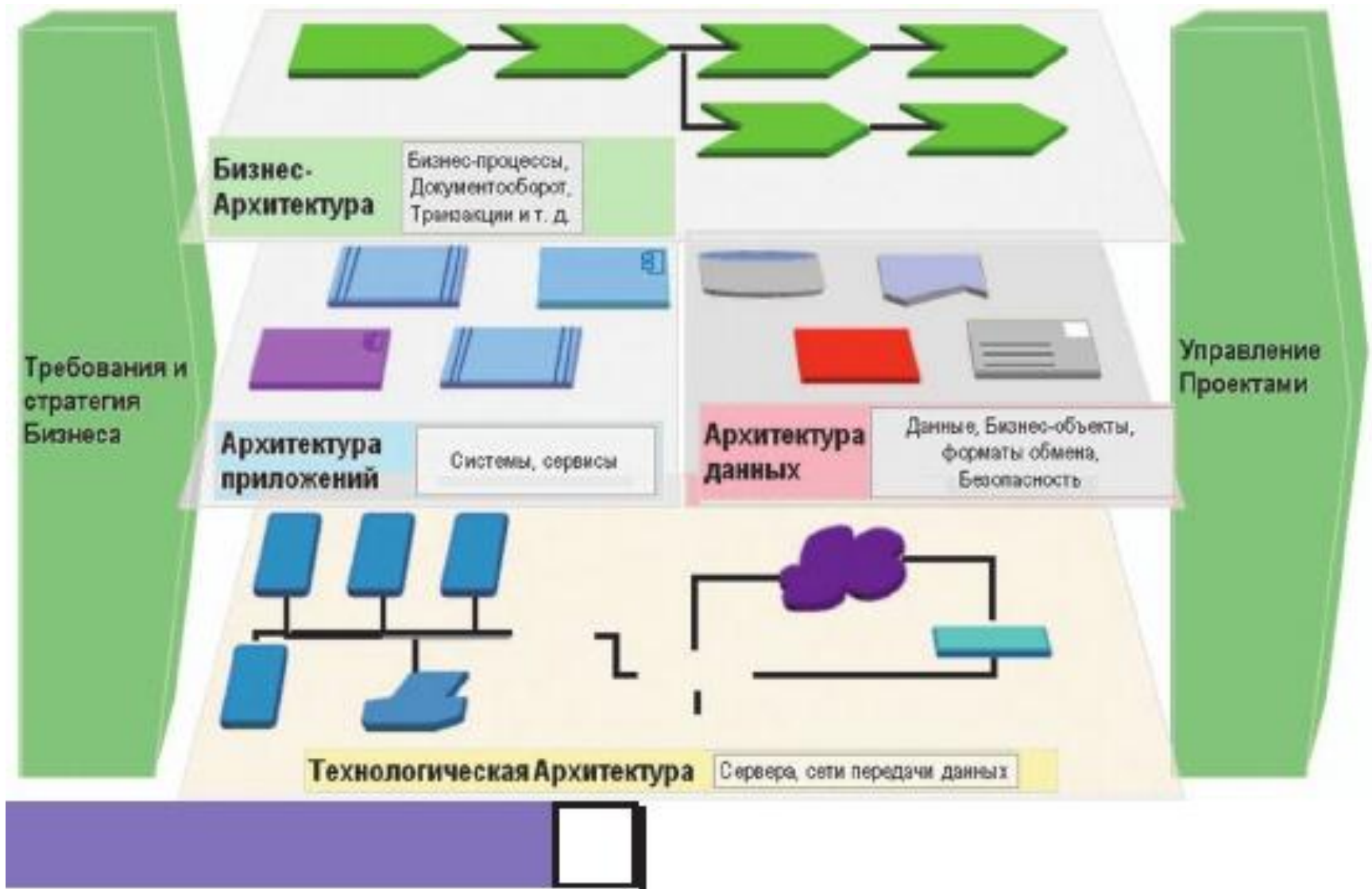
Эффективно  
использует  
технологическую  
архитектуру

Технологическая архитектура

# Роль и место – 2/3



# Роль и место – 3/3



# Назначение и состав – 1/3

- Основное назначение технологической архитектуры
  - Обеспечение надежных ИТ-сервисов, предоставляемых в рамках всего предприятия в целом
- Управление сервисами обычно централизовано
- В зависимости от особенностей (и размера!) предприятия отдельные аспекты (безопасность, интеграция, иногда разработка) могут быть выделены в отдельные области (домены) архитектуры предприятия

# Назначение и состав – 2/3

- Технологическая архитектура определяет набор принципов и стандартов
  - Индустриальных стандартов
  - Стандартов, связанных с продуктами
  - Конфигураций
- Набор стандартов позволяет обоснованно выбрать и использовать технологические решения:
  - Платформы (аппаратные и ОС), СУБД
  - Средства разработки, языки программирования
  - ПО промежуточного слоя
  - Сервисы электронной почты, каталоги, системы безопасности, сетевая инфраструктура и т.д.

# Назначение и состав – 3/3

- Одна из частных задач, решаемых в рамках технологической архитектуры, – формирование «списка закупаемых технологий»
  - Обычно инфраструктурные сервисы стандартизированы и используются сразу несколькими прикладными системами, расположенными над уровнем инфраструктурных сервисов и непосредственно обеспечивающих выполнение бизнес-процессов
  - При таком построении новые прикладные системы (для выполнения новых бизнес-процессов или реализации новых стратегий) могут быть построены достаточно быстро и эффективно
- Предпосылка динамичности и гибкости ИТ-архитектуры и предприятия в целом

# Основные компоненты – 1/2

- Сервисы данных
  - СУБД, хранилища данных
  - СППР, средства анализа, средства подготовки отчетов
- Прикладные сервисы
  - Языки программирования, средства разработки
  - Средства коллективной работы (почта, групповая работа, управление документами)

## Основные компоненты – 2/2

- ПО промежуточного слоя
- Вычислительная инфраструктура
- Сетевые сервисы
- Сервисы безопасности



# Принципы построения



# Как управлять?

- Основные этапы / процессы управления технологической инфраструктурой:
  - Оценка состояния / анализ успешности внедрения → формулировка требований к модификации
  - Формирование решений (новых архитектурных проектов)
  - Внедрение
  - Поддержание текущего состояния

# Оценка состояния

- Оценивается соответствие требованиям бизнес-стратегии
- Критерии Питера Кина:
  - Функциональные возможности
    - Возможности по выполнению бизнес-активностей
      - Простых. Например, пересылка информации (сообщений)
      - Сложных. Например, транзакций, производимых совместно сотрудниками, а также поставщиками и клиентами
  - Охват
    - Объединение физических мест расположения и/или групп людей

# **АДАПТИВНАЯ ТЕХНОЛОГИЧЕСКАЯ АРХИТЕКТУРА**

# Что делается само? – 1/2

- Самоконфигурирование
  - Организация системы в соответствии с [текущими] требованиями
- Самозащита
  - Предотвращение сбоев в системе в результате нарушения работы компонент системы и потери целостности данных

# Что делается само? – 2/2

- Самовосстановление
  - Диагностика неисправностей, локализация ошибок и устранение их последствий
- Самооптимизация
  - Наиболее рациональное использование имеющихся ресурсов без вмешательства оператора

# Принципы

- Все ИТ-ресурсы являются общими и разделяемыми
- Выделение ресурсов конкретным приложениям производится автоматически в соответствии с требованиями бизнеса
- Качество обслуживания является предсказуемым и стабильным, несмотря на непредсказуемый спрос на ресурсы

# Концепции – 1/2

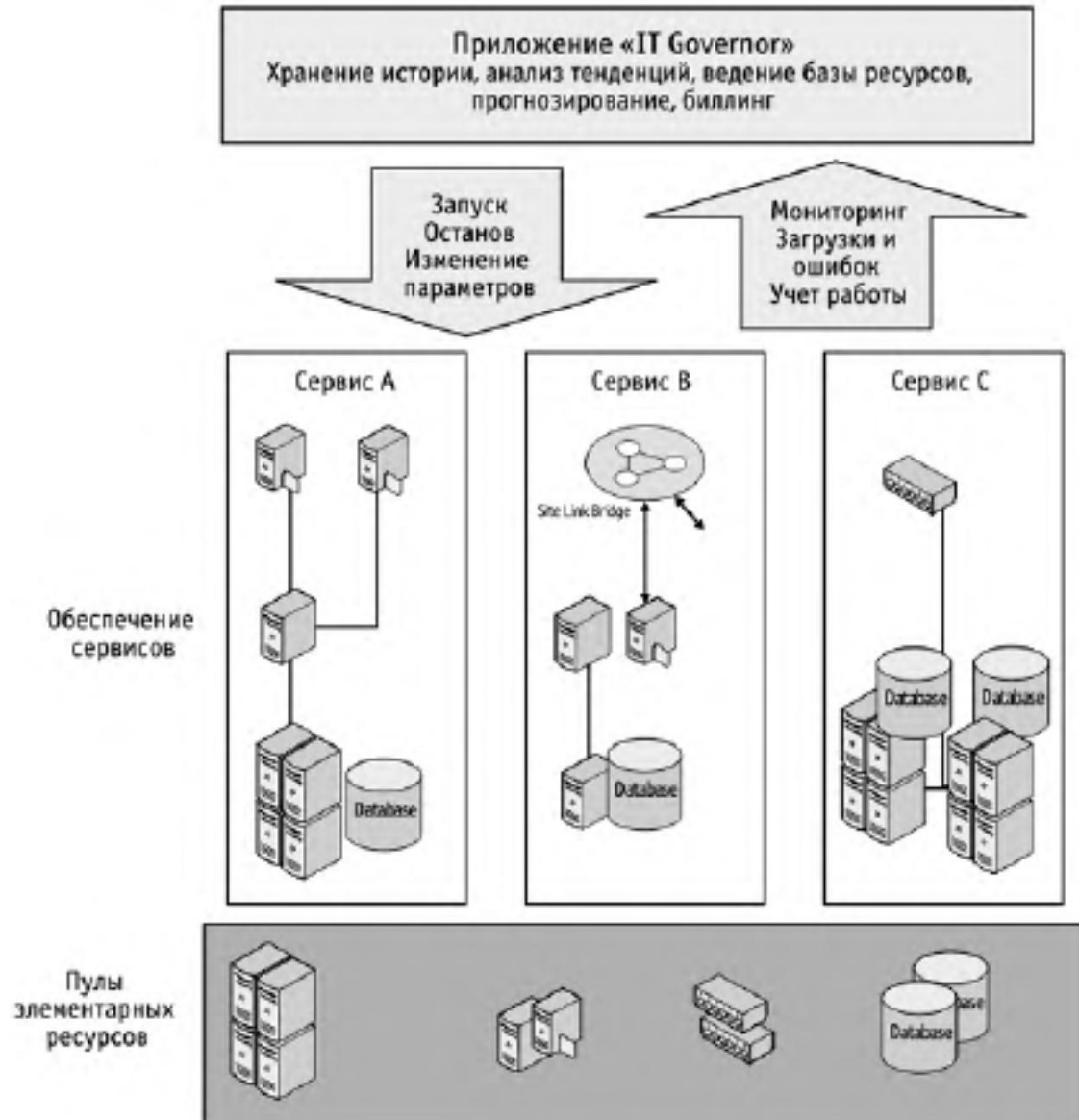
- Organic IT от компании Forrester
  - «...это инфраструктура, построенная из относительно **дешевых компонентов** с такой **избыточностью**, которая обеспечивает автоматическое разделение между всеми приложениями и администрирование корпоративных информационных ресурсов, в том числе программного обеспечения, процессоров, систем хранения данных и сетей»
  - Позволяет повысить «коэффициент полезного использования» ресурсов
  - Упрощает интеграцию различных технологических компонент между собой
  - Облегчает управляемость системы в целом



## Концепции – 2/2

- Real-Time Infrastructure – от Gartner
  - «Разделяемая между участниками, бизнес-подразделениями или приложениями инфраструктуру информационных систем, в которой бизнес-политики и соглашения об уровнях услуг (SLA) определяют ее динамическую и адаптивную оптимизацию для сокращения затрат при увеличении гибкости и качества сервиса»

# Пример – 1/4



## Пример – 2/4

- Элементарными «строительными блоками» являются отдельные ресурсы
  - Серверы и отдельные процессоры
  - Виртуальные устройства хранения данных
  - Сетевое оборудование
  - Программные серверы приложений
  - СУБД
  - etc.

## Пример – 3/4

- Обеспечение требований бизнеса производится в рамках модели предоставления сервисов
  - Каждый сервис реализуется на основе нужной комбинации программных и аппаратных ресурсов, преимущественно в рамках многозвенной архитектуры:
    - уровень СУБД – уровень приложений – презентационный уровень

# Пример – 4/4

- **Управляющий модуль (IT governor)**
  - Мониторит
    - Времена реакции сервисов на запросы, прогнозируемые и исторические значения потребностей приложений, наличие ошибок, выход из строя элементов системы и т.п.
  - Управляет
    - Переконфигурирует виртуальные сервера, запускает сервера приложений, управляет пулами однородных устройств, изменяет приоритеты выполняемых приложений
  - Обеспечивает биллинг (учет реально использованных приложениями ресурсов)

# **СТАНДАРТЫ ТЕХНОЛОГИЧЕСКОЙ АРХИТЕКТУРЫ**

# Стандарты ТИ – 1/2

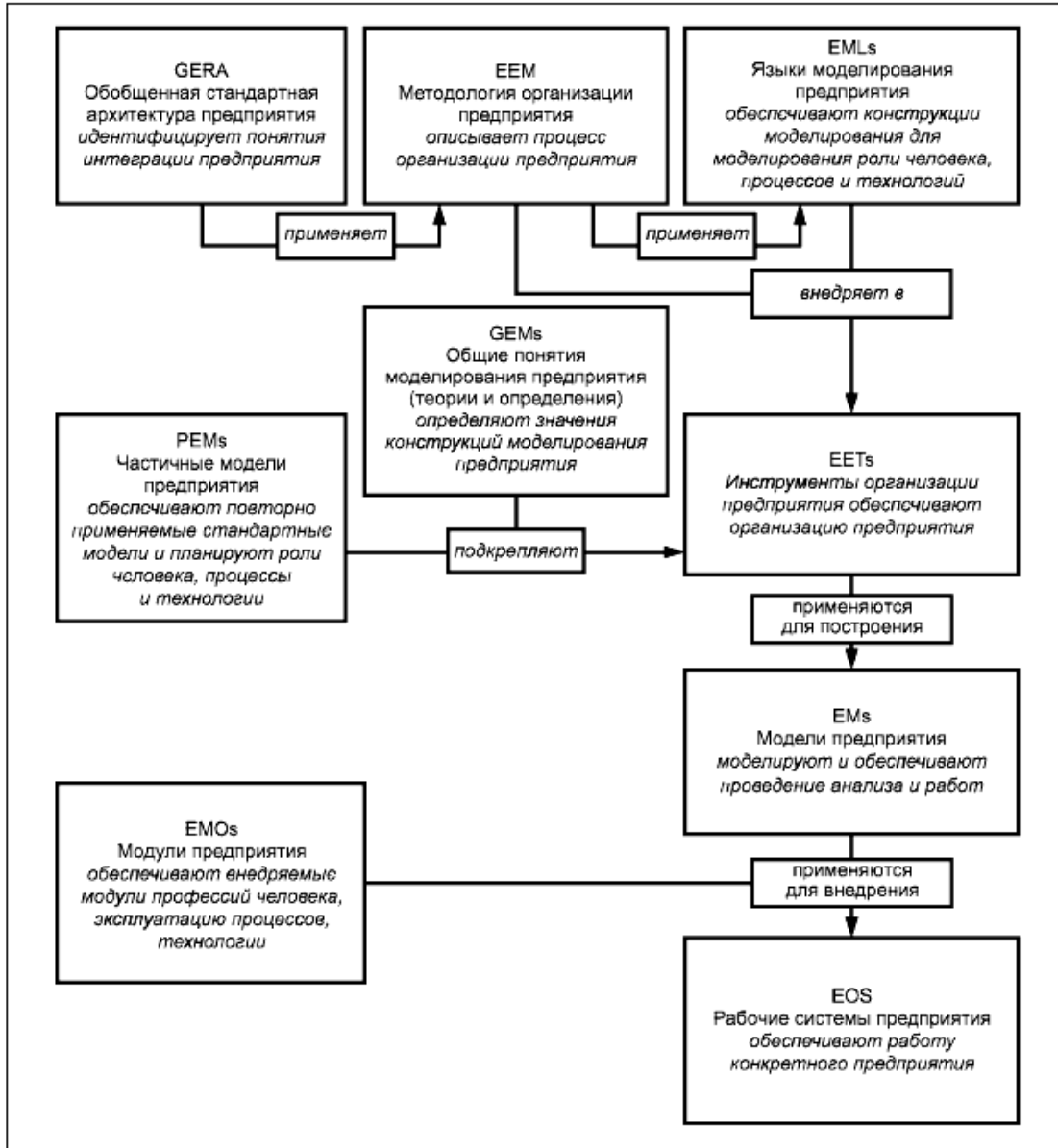
- Технологические
  - Определяют особенности реализации протоколов, интерфейсов, языков программирования, etc.
  - Например: спецификации WWW консорциума W3C

## Стандарты ТИ – 2/2

- Жизненного цикла (Framework – «Рамочные»)
  - ISO 15704 = ГОСТ Р ИСО 15704-2008 –  
Промышленные автоматизированные системы.  
Требования к стандартным архитектурам и  
методологиям предприятия
  - ISO 15288 – Systems Engineering Process Rollout
  - ISO/IEC 12207 – Systems and software  
engineering — Software life cycle processes

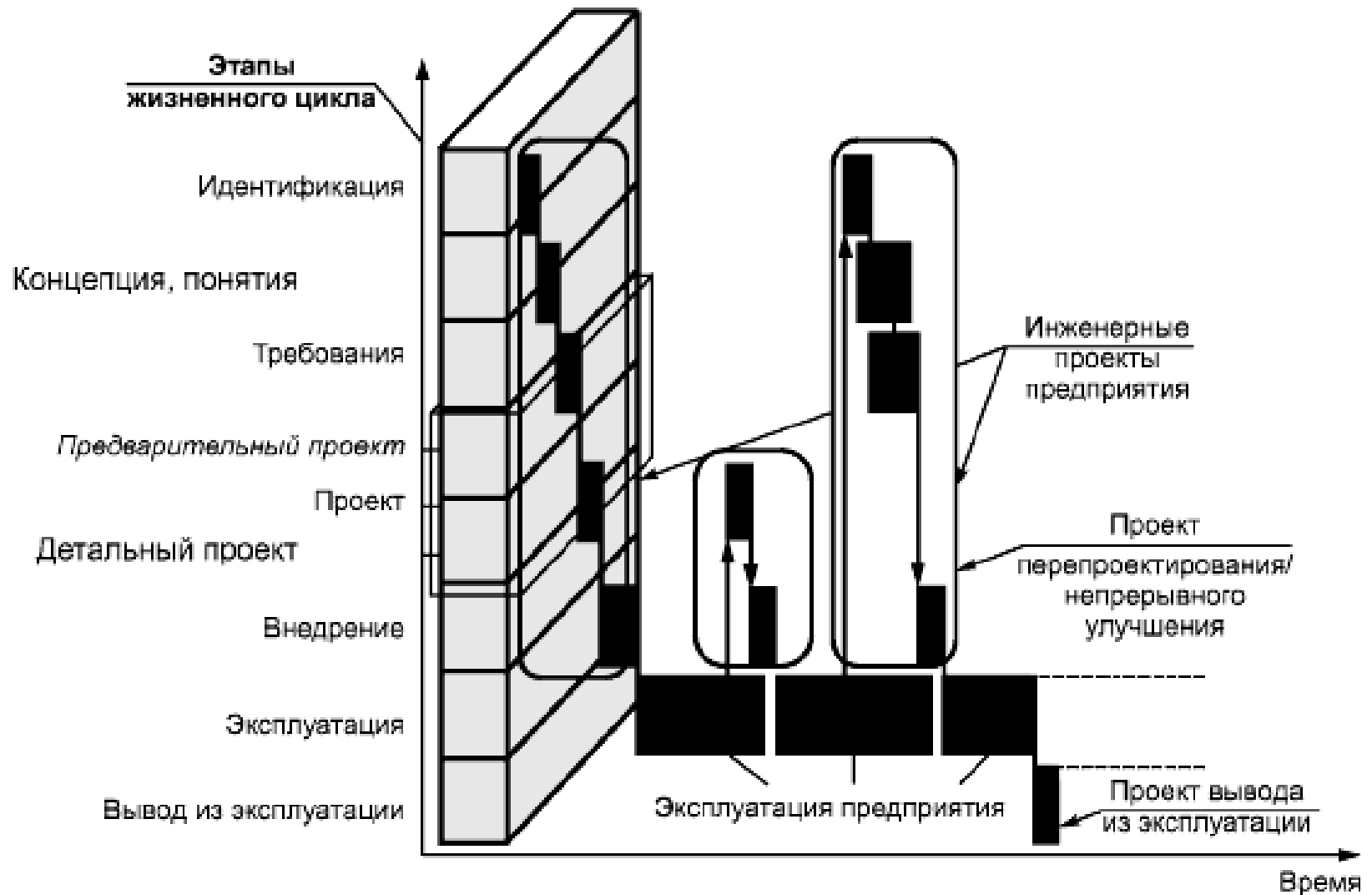


# ISO 15704 – 1/4

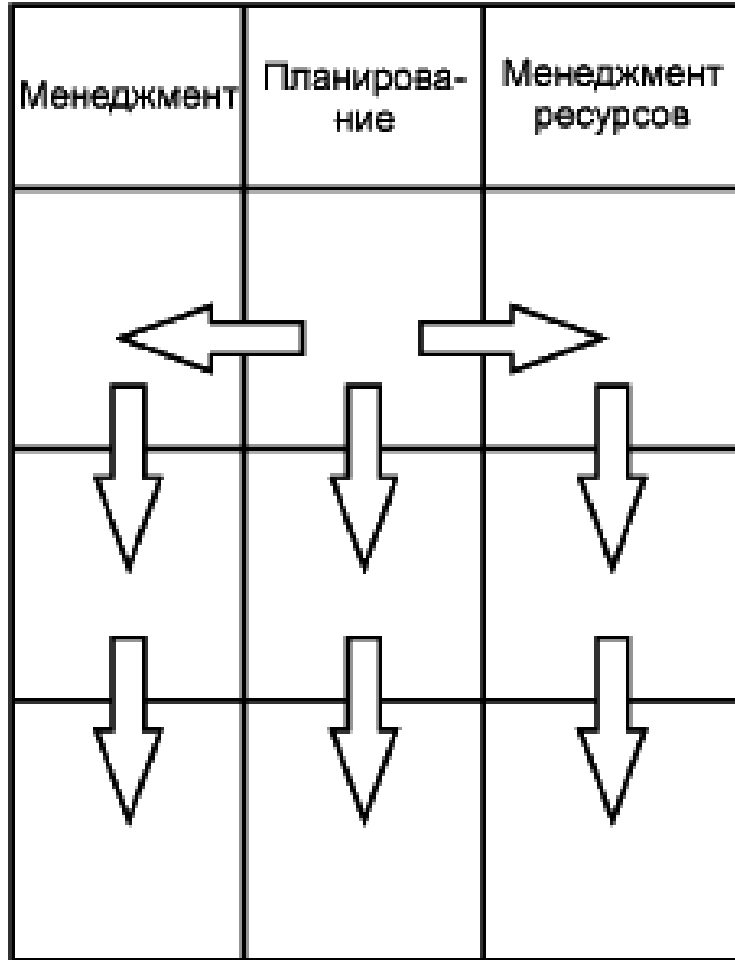


Обобщенная  
стандартная  
архитектура  
предприятия и  
методология

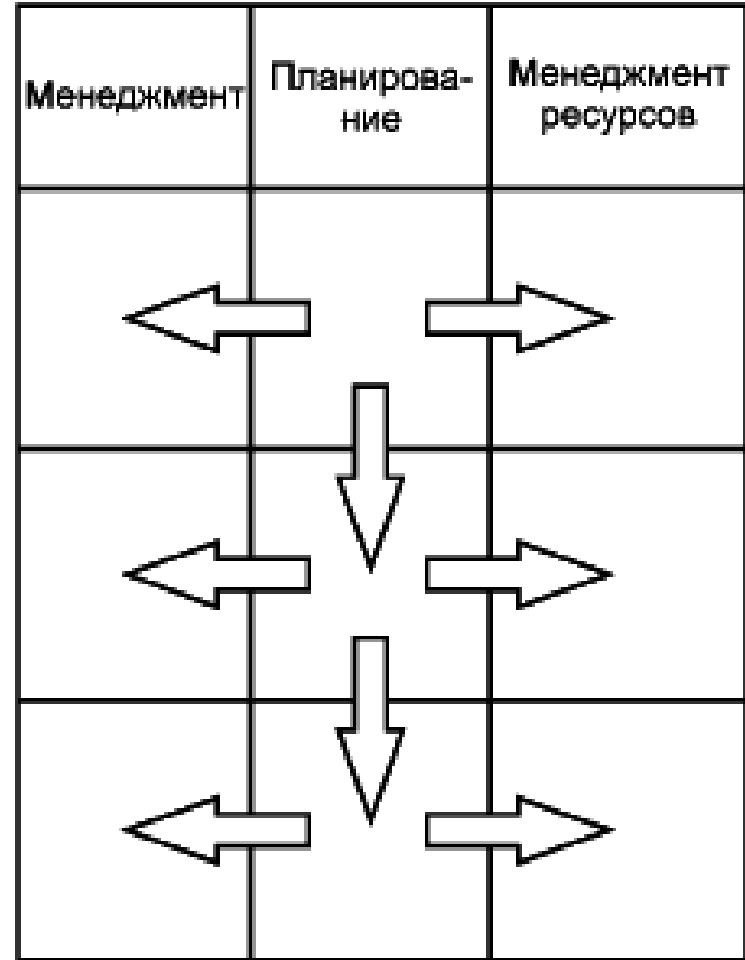
# ISO 15704 – 2/4



# ISO 15704 – 3/4

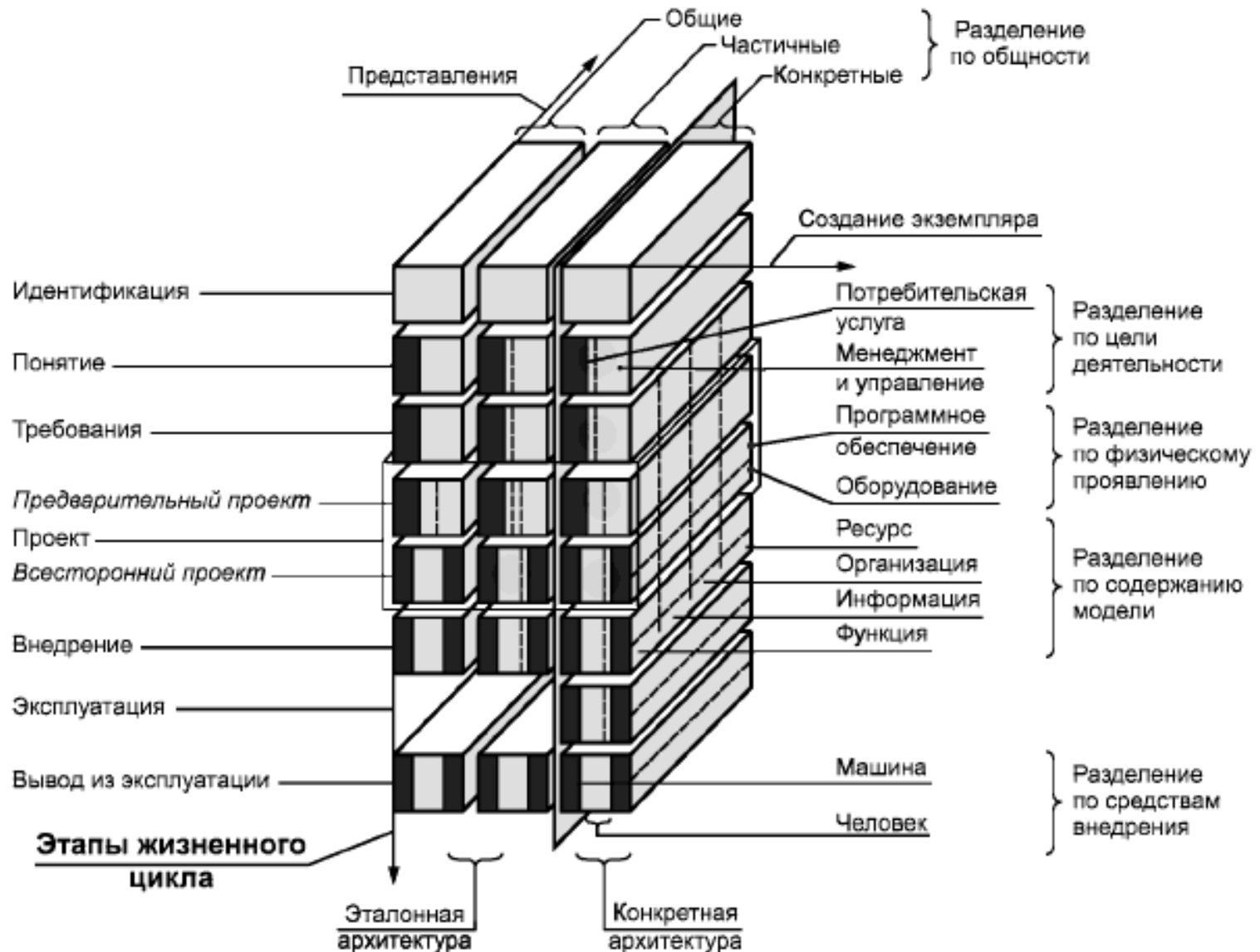


Координационная структура



Синхронизирующая структура

# ISO 15704 – 4/4



**МОНТОРИНГ**

# Определения

- Мониторинг — непрерывный процесс
  - Наблюдения параметров объекта
  - Регистрации значений
  - [Простого] анализа значений (например, сравнение с заданными значениями)
  - Обычно: количество характеристик небольшое
  - Обычно: непрерывный = периодический.  
Период такой, чтобы наблюдаемые значения не могли существенно измениться
- Предназначен для вынесения суждения о поведении/состоянии наблюдаемого объекта в целом

# Виды

- Параметров
  - Наблюдаются значения параметров
- Состояния
  - Наблюдаемые значения параметров интерпретируются
- Обнаружение ошибок, критических ситуаций, опасных ситуаций

# Способы построения

- **Агенты**
  - На наблюдаемых хостах устанавливаются специализированные приложения
- **Службы ОС**
  - Определение характеристик хостов получается через службы ОС
- **Специализированные протоколы**
  - Для получения информации используются стандартные протоколы



# Типичные функции – 1/3

- Интерфейсы:
  - Визуализация состояния
  - Наличие веб-интерфейса
  - Наличие протокола управления / выдачи
  - Информирование: Почта, SMS, ...
  - Количество подключаемых «наблюдателей»
  - Управление доступом
- Отчетность:
  - Хранение журналов
  - Формирование отчетов

# Типичные функции – 2/3

- Описание наблюдаемых систем
  - Линейность – Иерархичность – Более сложные
  - Автоматическое обнаружение элементов
- Поддерживаемые способы:
  - Агент
  - SNMP

# Типичные функции – 3/3

- Обнаружение опасных ситуаций:
  - Задание границ значений
  - Сигнализация о потере связи с наблюдаемым объектом
  - Возможность описать пользовательский алгоритм
  - Предсказание событий (trend analysis)
- Объекты мониторинга:
  - Состояние хостов (серверов)
  - Сетевое оборудование
  - Сервисы

# Пример: ZABBIX – 1/5

- Объекты мониторинга:
  - Сервисы компьютерной сети
  - Сервера
  - Сетевое оборудование
- Способы мониторинга:
  - Simple checks
    - Проверяет доступность (наличие отклика) стандартных сервисов
    - Не требуется установка ПО на наблюдаемом хосте
    - Примеры сервисов: SMTP, HTTP

# Пример: ZABBIX – 2/5

- Способы мониторинга (продолжение):
  - ZABBIX agent
    - Устанавливается на UNIX-подобных или Windows хостах
    - Выдает данные о нагрузке процессора, использовании сети, дисковом пространстве
  - External check
    - Выполняются внешние программы
  - ZABBIX поддерживает мониторинг через SNMP

# Пример: ZABBIX – 3/5

- Архитектура:
  - Zabbix сервер
    - Ядро программного обеспечения Zabbix
    - Функциональность:
      - Проверка функционирования сетевых сервисов (simple checks)
      - Хранит все конфигурационные, статистические и оперативные данные
      - Обеспечивает интерфейс с администраторами. В том числе оповещает их в случае возникновения проблем с любым контролируемым оборудованием

# Пример: ZABBIX – 4/5

- Архитектура (продолжение):
  - Zabbix прокси
    - Принадлежит Zabbix серверу
    - Собирает данные о производительности и доступности от Zabbix Agents
    - Собранные буферизуются и передаются Zabbix серверу
    - Используется для:
      - Организации иерархичности (централизованный удаленный мониторинг мест, филиалов, сетей, не имеющих локальных администраторов)
      - Распределения нагрузки одного Zabbix сервера

# Пример: ZABBIX – 5/5

- Архитектура (продолжение):

- Zabbix агент

- Мониторинг локальных ресурсов и приложений (Например: жесткие диски, память, статистика процессора)
    - Системно зависим → эффективен

- Веб-интерфейс

- Часть Zabbix сервера
    - Требует наличия запущенного веб сервера
    - Форматы жестко привязаны к проприетарной структуре БД



# Пример: AdRem NetCrunch – 1/2

- Коммерческая
- Не использует агенты
- Кроссплатформенная
- Объекты мониторинга:
  - Сетевые сервисы – 65 типов
  - Приложения Windows
  - Платформы (ОС): Windows, Linux, NetWare, BSD, Mac OS X, устройства поддерживающие SNMP версий 1-3

# Пример: AdRem NetCrunch – 2/2

- Дополнительная функциональность:
  - Централизованное информирование о сбоях путем сбора и анализа событий из следующих источников:
    - Windows Event Log
    - Syslogs
    - SNMP traps
  - Представляет физическую и логическую топологии сети в автоматически обновляемом графическом виде

# **СИСТЕМЫ ОТСЛЕЖИВАНИЯ ОШИБОК**

# Системы отслеживания ошибок

- Bug tracking system – прикладное ПО позволяющее:
  - Учитывать и контролировать:
    - Ошибки и неполадки, найденные в программах
    - Пожелания пользователей
  - Следить за процессом устранения этих ошибок и выполнения или невыполнения пожеланий
- Обычно позволяют регистрировать проблему большому числу пользователей
- Используются для оценки качества разработки, обслуживания клиентов или сопровождения системы

# Жизненный цикл дефекта – 1/2

- Новый – дефект зарегистрирован тестировщиком
- Назначен — указан ответственный за исправление дефекта
- Разрешён — дефект переходит обратно в сферу ответственности тестировщика
- Далее проводится проверка исправления:
  - Назначен (если он описан как исправленный, но не исправлен)
  - Закрыт
- Открыт повторно — вновь найден в другой версии

# Жизненный цикл дефекта – 2/2

- Разрешён → Обычно указывается результат работы с дефектом:
  - Исправлено (с указанием версии)
  - Дублирование [описания дефекта]
  - Отказано
    - Работает в соответствии со спецификацией
    - Имеет слишком низкий приоритет
    - Исправление отложено до следующей версии etc.
  - Невоспроизводимо
    - Запрашивается дополнительная информация об условиях, в которых дефект проявляется

# Описание дефекта – 1/2

- Минимальное:
  - Номер (идентификатор)
  - Кто сообщил
  - Дата и время обнаружения
  - Короткое описание
  - Ожидаемый vs. фактический результат
  - Текущее состояние (статус)
- Описание окружения:
  - Версия продукта, в которой обнаружен дефект
  - Описание шагов для воспроизведения
  - Конфигурация программного средства

## Описание дефекта – 2/2

- Назначаемые в процессе работы:
  - Серьёзность (критичность) дефекта и приоритет решения
  - Ответственный за устранение
  - Обсуждение возможных решений и их последствий
  - Версия продукта, в которой дефект исправлен.
- Расширенное описание:
  - Текст
  - Прикрепленные файлы, помогающие описать проблему. Например, дампы памяти, логи или скриншоты



# Пример: MantisBT

- Лицензия: GPL
- Веб-приложение → не требует для работы специального ПО кроме веб-браузера
- Хорошо конфигурируется, что позволяет использовать её в качестве системы учёта заявок для Helpdesk
- Возможность интеграция с wiki-движком для создания документации (DokuWiki)
- Название Mantis (богомол) происходит от того, что богомол питается жуками (bug)