

Управление программными проектами

7 семестр

Лекций – 20

Лабораторных работ - 32

Практических занятий – 6

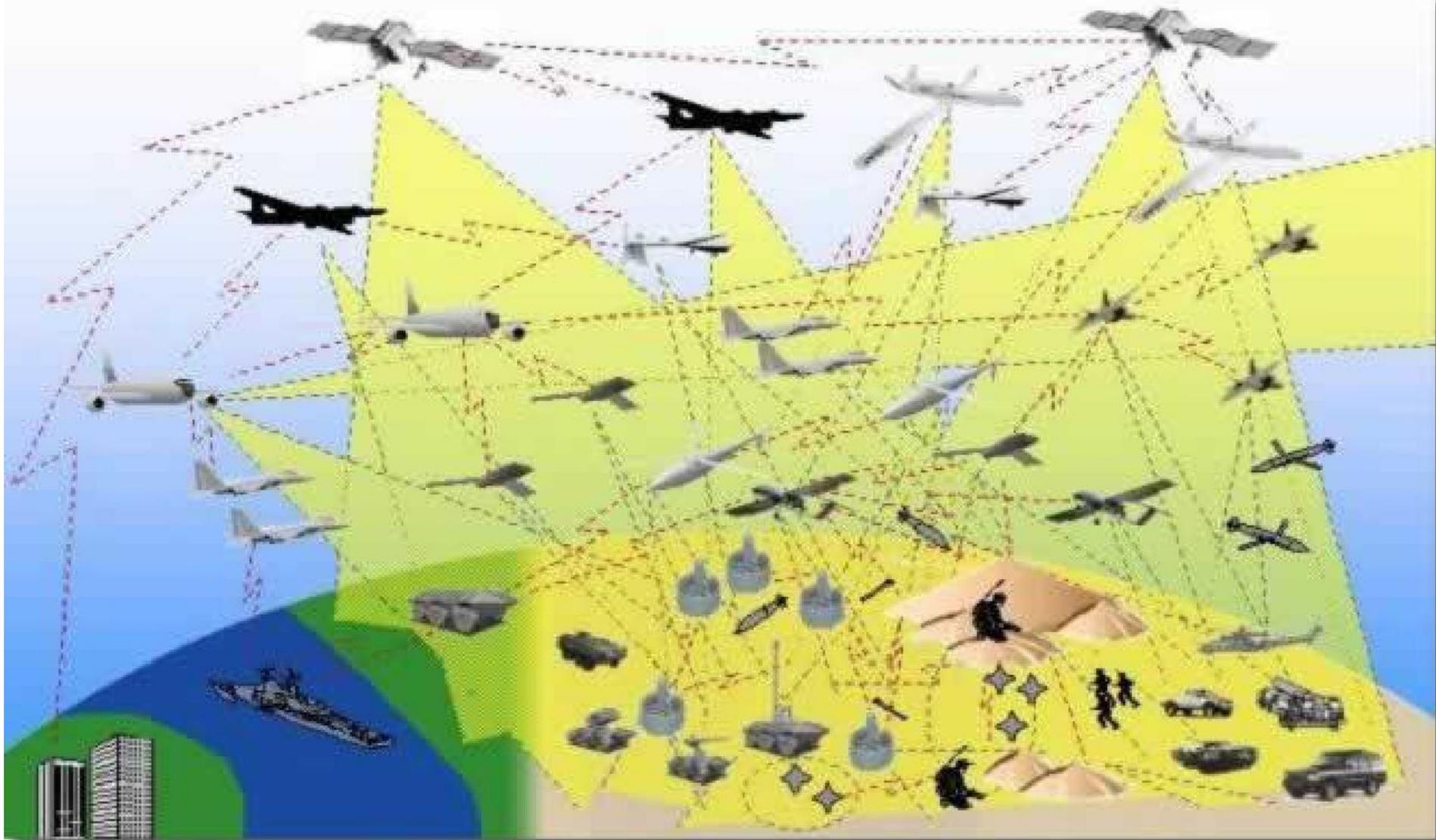
Отчетность: -зачет с оценкой

Гвоздев Владимир Ефимович

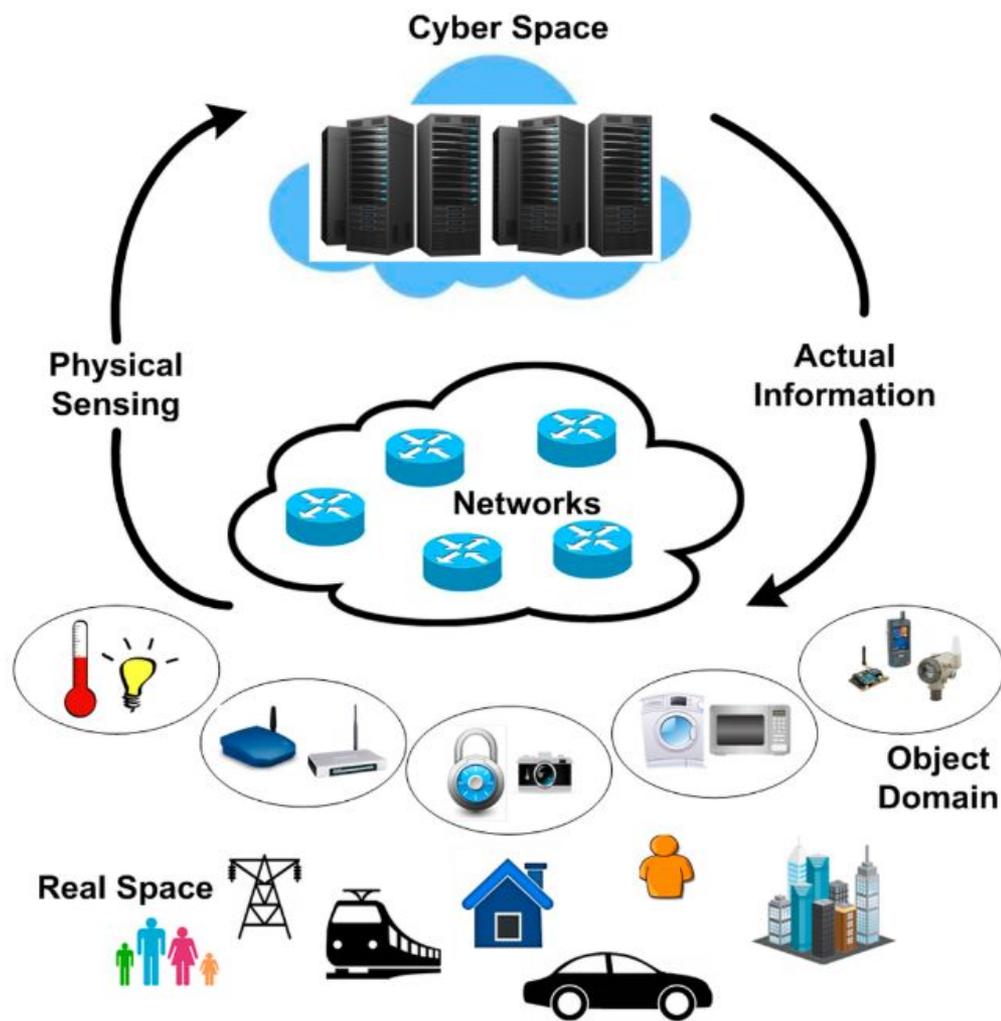
д.т.н., профессор

6-212

Сетецентрическое управление системами систем мобильных и стационарных объектов



“Будущее – это цифровая вещь...”



«The Future
Is a Digital Thing»

[Top Strategic Predictions for 2016 and Beyond]

Gartner

Industry 4.0 | Что это?

From Industry 1.0 to Industry 4.0

First Industrial Revolution

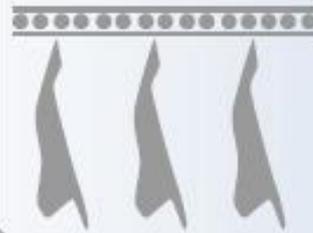
based on the introduction of mechanical production equipment driven by water and steam power



First mechanical loom, 1784

Second Industrial Revolution

based on mass production achieved by division of labor concept and the use of electrical energy



First conveyor belt, Cincinnati slaughterhouse, 1870

Third Industrial Revolution

based on the use of electronics and IT to further automate production



First programmable logic controller (PLC) Modicon 084, 1969

Fourth Industrial Revolution

based on the use of cyber-physical systems



Degree of complexity



1800

1900

2000

Today

Time

Концепция четвертой промышленной революции («Индустрии 4.0») была сформулирована в 2011 году во время Ганноверской ярмарки группой представителей немецкой промышленности и бизнес-сообщества в рамках инициативы по повышению конкурентоспособности Германии

Industry 4.0 | Что это?

Industry 4.0

(Kagermann H., Lukas W., Wahlster W., 2011, Germany)

Integrated Industry

(Bürger & Tragl, 2014, Germany)

Smart Industry or Smart Manufacturing

(Davis, Edgar, Porter, Bernaden, & Sarli, 2012, Germany)

Industrial Internet

(Industrial Internet Consortium, 2012, USA)

- Cyber-Physical Systems (CPS)
 - Internet of Things (IoT)
 - Internet of Services
 - Smart Factory

*[Hermann M., Pentek T., Otto B. Design Principles for Industrie 4.0 Scenarios: A Literature Review. Working Paper No. 01. 2015.

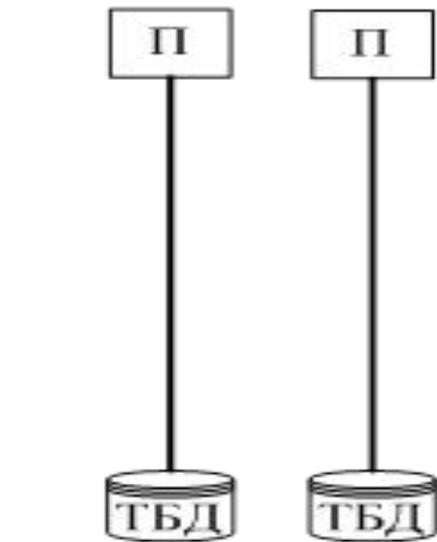
http://www.snom.mb.tu-dortmund.de/cms/de/forschung/Arbeitsberichte/Design-Principles-for-Industrie-4_0-Scenarios.pdf

- Industrial Internet of Things (IIoT)
- Additive Production (3D - the printing)
 - BigData
 - Artificial Intelligence (AI)
- Collaborative Robots (CoBot)
 - Virtual Reality (VR)

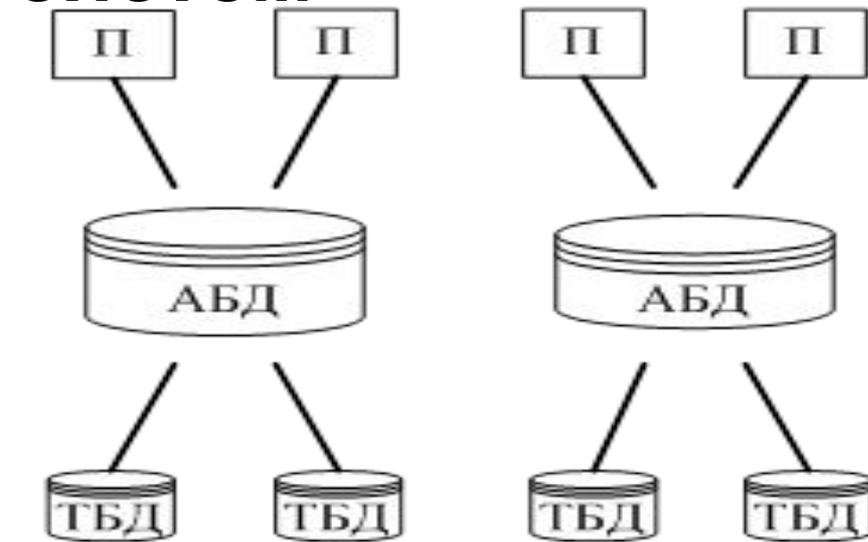
*[Hermann M., Pentek T., Otto B. Design Principles for Industrie 4.0 Scenarios: A Literature Review. Working Paper No. 01. 2015.

http://www.snom.mb.tu-dortmund.de/cms/de/forschung/Arbeitsberichte/Design-Principles-for-Industrie-4_0-Scenarios.pdf

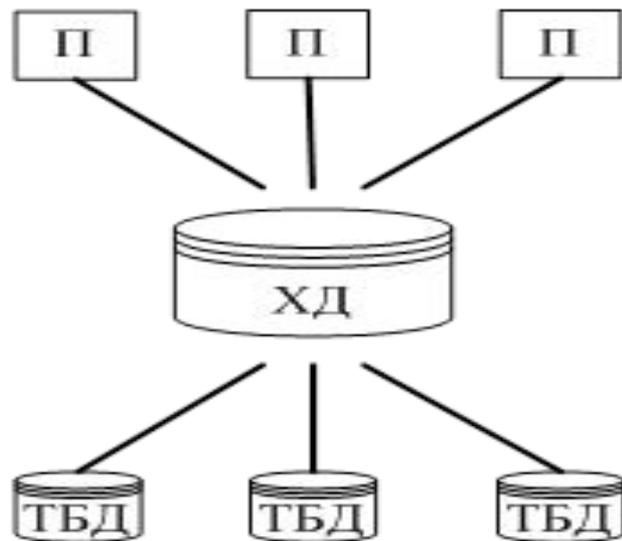
Эволюция архитектур программных систем



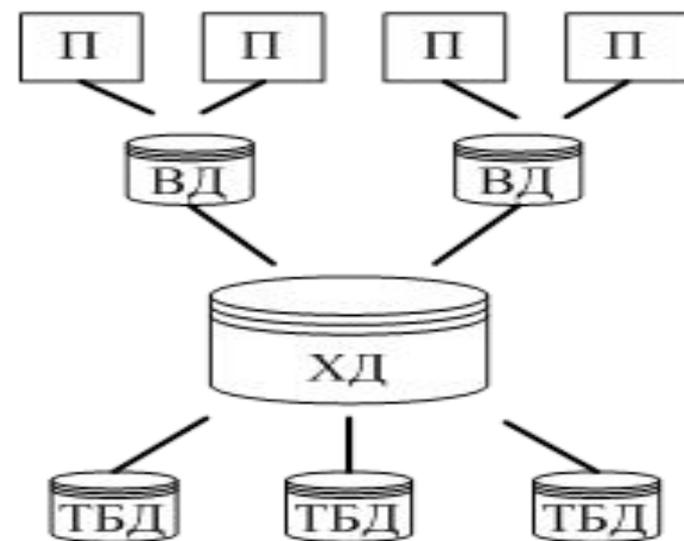
a



б



в



г

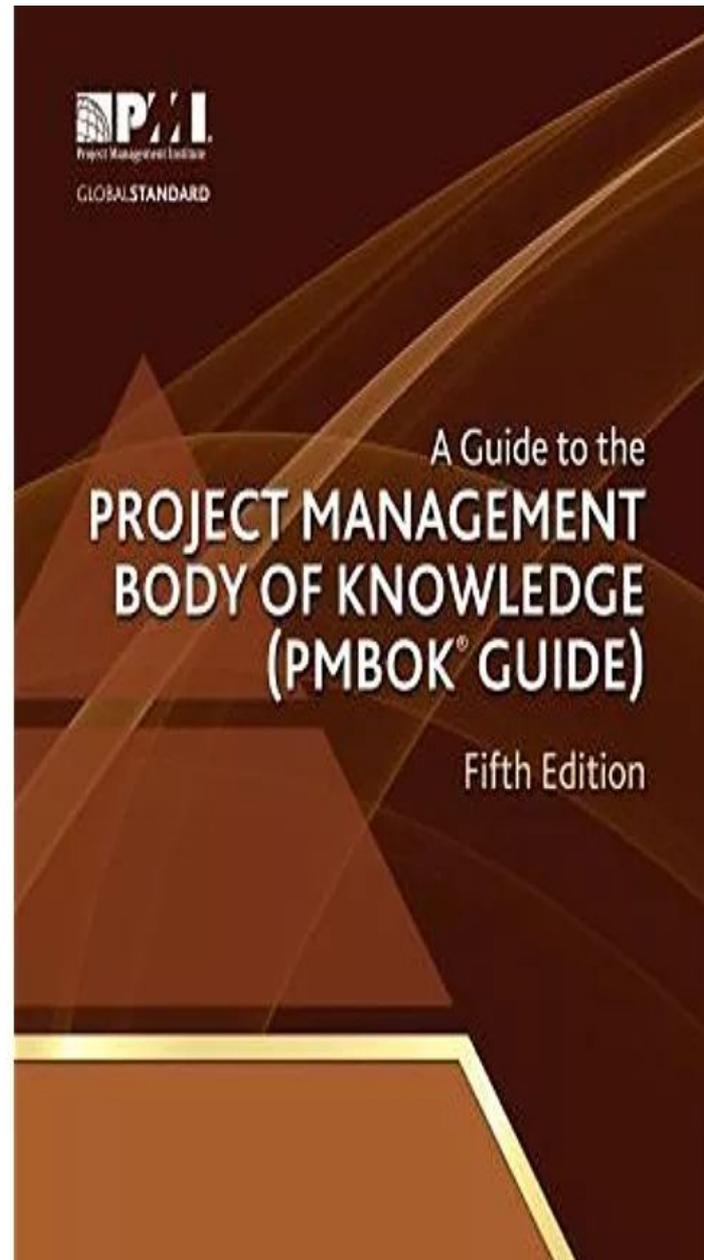
Роли проекта

1. Проект как система предписаний:
создает предписание для изготовления изделия
2. Проект как модель создаваемого объекта : описывает строение, функционирование, внешний/внутренний вид объекта, добиваясь, чтобы его структура удовлетворяла требования заказчика и принципы проектирования

Проект как система предписаний



Проект как система предписаний



Назначение проекта как модели создаваемого объекта

1. Коммуникативная: связывает заказчика, проектировщика и потребителя
2. Объектно-онтологическую: обеспечивает внутри процесса проектирования разработку и создание проектируемого объекта

Отличительные признаки проекта

Пять основных характеристик проекта, отличающих их от других классов сложных систем:

- проекты имеют разовый характер;
- каждый проект по-своему неповторим;
- проекты ограничены четкими временными рамками;
- все проекты сопряжены с изменениями;
- проекты дают определенные результаты.

Принципы проектирования

Принцип реализуемости: по проекту в существующем производстве можно изготовить соответствующий проекту объект.

Принцип соответствия: в проектируемом объекте можно выделить, описать, разработать процессы функционирования и морфологические единицы (единицы строения) и поставить их в соответствие друг другу. То же справедливо в отношении функций и конструкций (сопоставимость устройства и внешнего поведения)

Принципы проектирования (продолжение)

Принцип завершенности: хотя почти любой проект может быть улучшен во многих отношениях, т.е. оптимизирован, тем не менее он удовлетворяет основным требованиям, предъявляемым к нему заказчиком.

Принцип конструктивной целостности: проектируемый объект состоит из элементов, единиц и отношений, которые могут быть изготовлены в соответствующем производстве

Принцип оптимальности: проектировщик стремится к оптимальным решениям

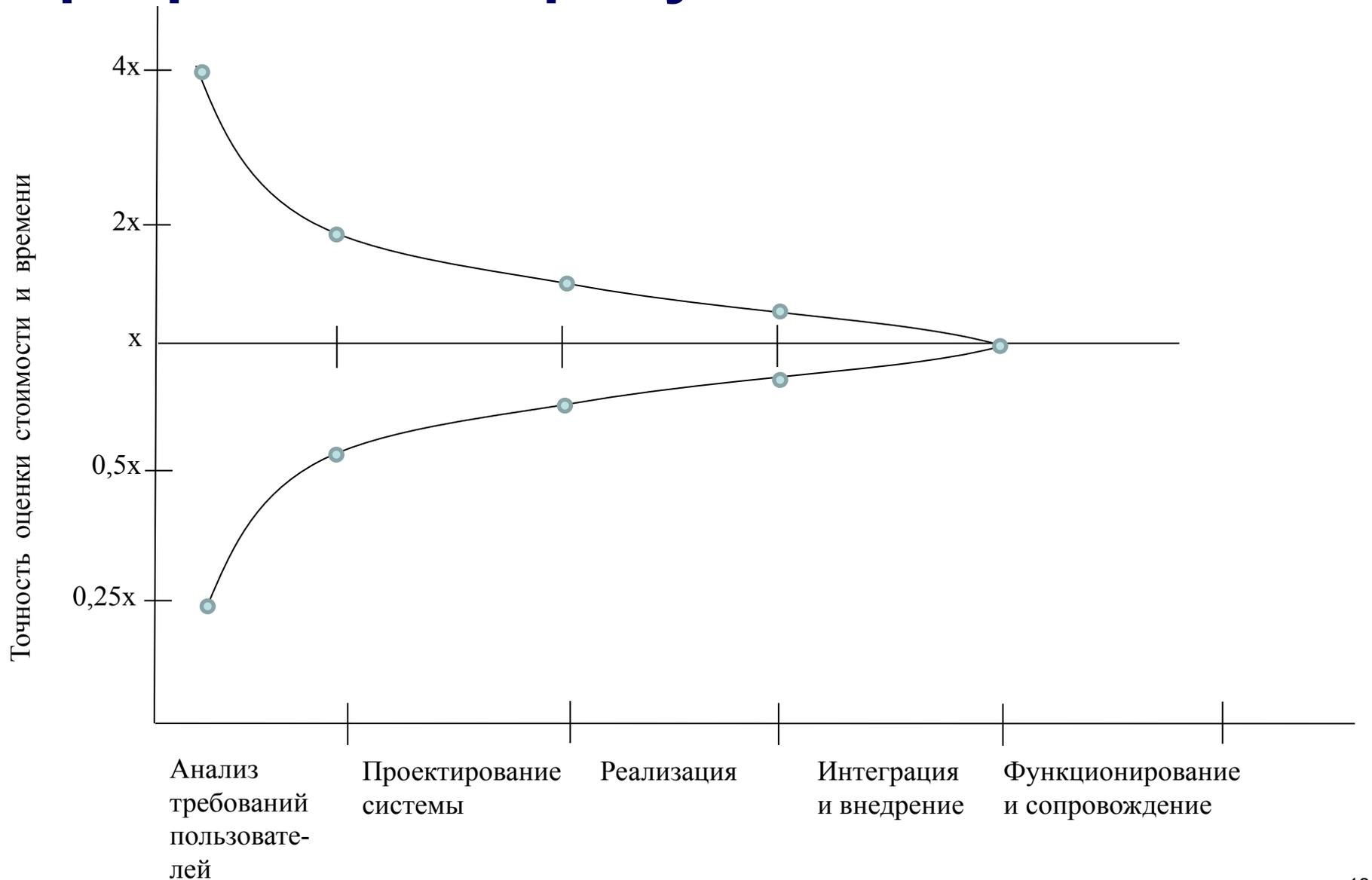
Принципы проектирования (продолжение)

Временной принцип оптимизации проектирования: условная идеализация оптимизационных моделей, в которых учтены лишь исследуемые факторы. Параметры, характеризующие свойства объекта и не являющиеся проектными переменными не могут быть определены с высокой степенью достоверности. Неопределенность проектной информации вынуждает искать решения, устойчивые к погрешностям прогнозных оценок.

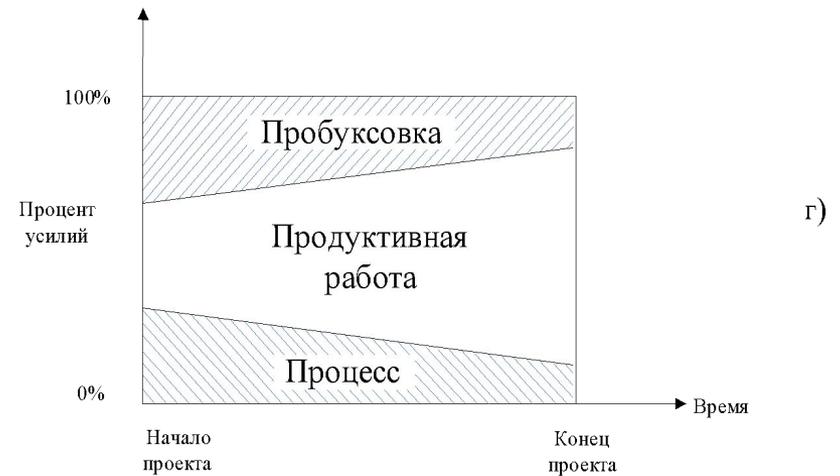
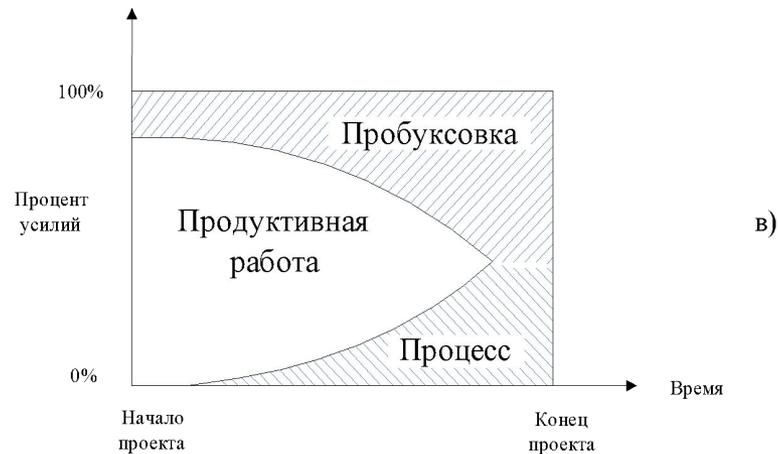
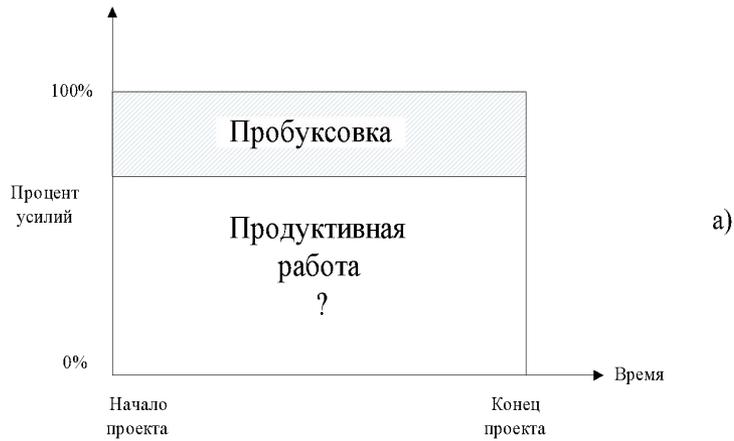
Принципы проектирования (продолжение)

Принцип экологичности: гармонизация создаваемого объекта с окружающей средой на всех этапах его жизненного цикла как по потребляемым ресурсам, так и по воздействию на среду, учет необходимости дальнейшей утилизации.

Конус неопределенности программного продукта

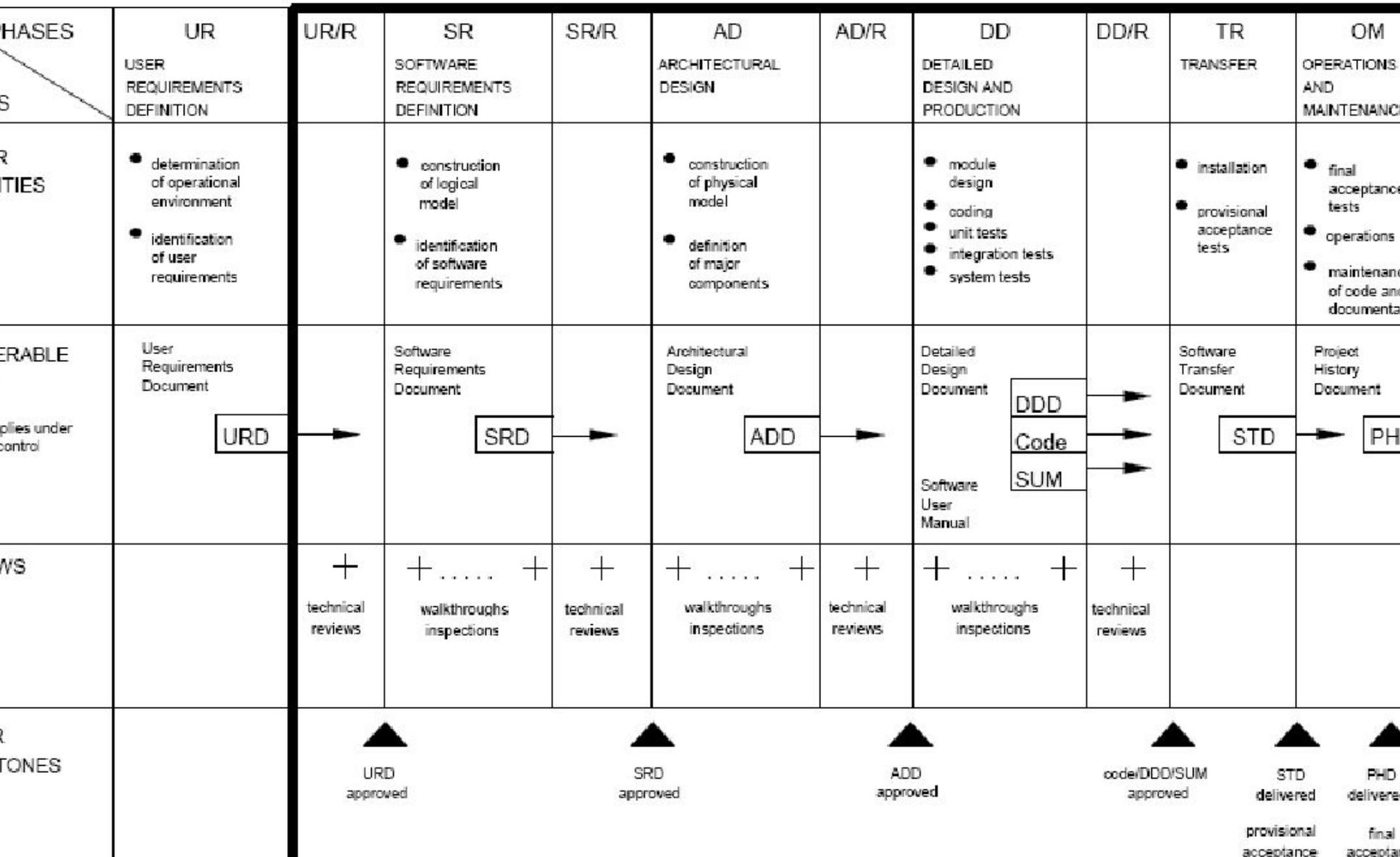


Роль дисциплины при проектировании сложных программных систем



Некоторые модели жизненного цикла

Место спецификации требований в жизненном цикле программной системы



Возможности модели жизненного цикла программной системы (потенциальность модели) должна соответствовать сложности реализации программного продукта.

Сложность реализации программного продукта определяется уровнем неопределенности требований к потребительским свойствам конечного продукта

Code-and-fix model

Реализация программного продукта сводится к непосредственному кодированию задачи в том виде, как она понимается.

Особенностями этой модели являются:

- Трудность модификации и развития ПП из-за недостаточно проработанной проектной стадии.
- Вследствие того, что задача кодировалась как понималась, т.е. стадия изучения и согласования пользовательских требований реализовывалась посредством экспериментирования с уже готовой программой, функциональные возможности программного продукта редко полностью согласуются с потребностями пользователей
- Сложность тестирования программного продукта.

Stagewise model

Разработка программного продукта сводится к следующей последовательности действий:

- Планирование разработки.
- Разработка операционной спецификации.
- Кодирование.
- Параметрическое тестирование модулей.
- Тестирование сборки.
- Опытную эксплуатацию.
- Оценку системы пользователем.

The waterfall approach

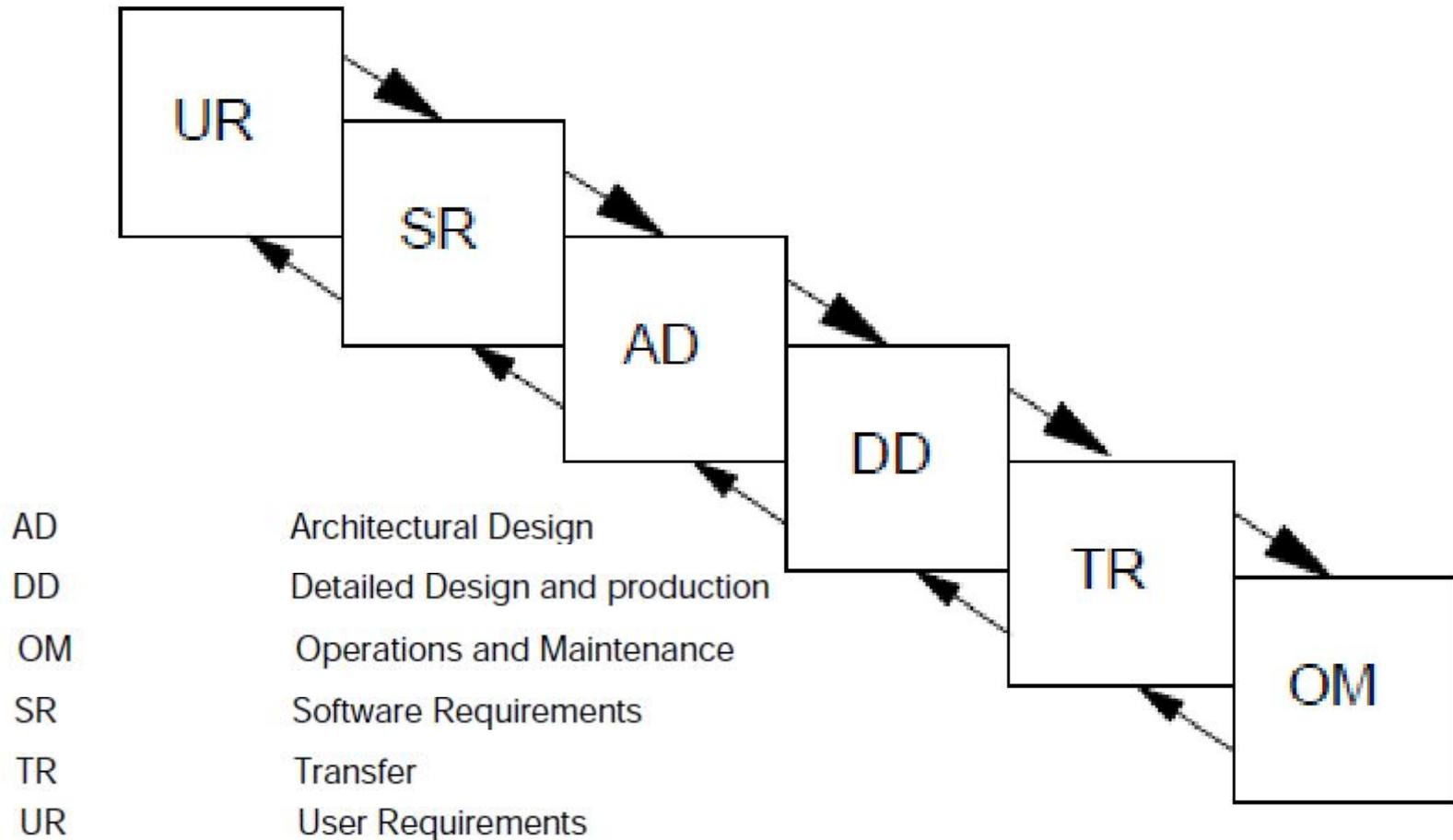


Figure 1.3.1 *The waterfall approach*

The incremental delivery approach

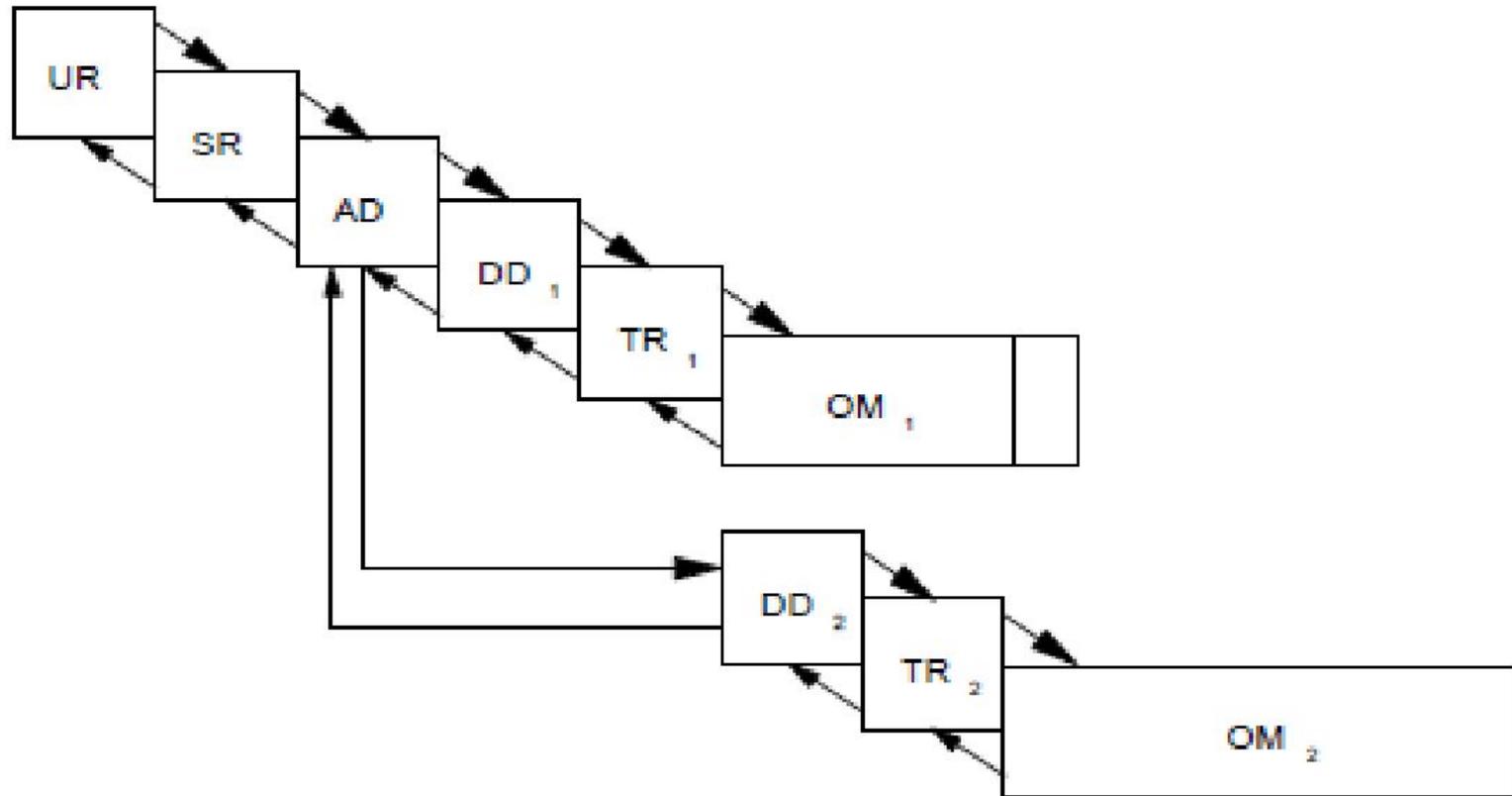


Figure 1.3.2 *The incremental delivery approach*

The evolutionary development approach

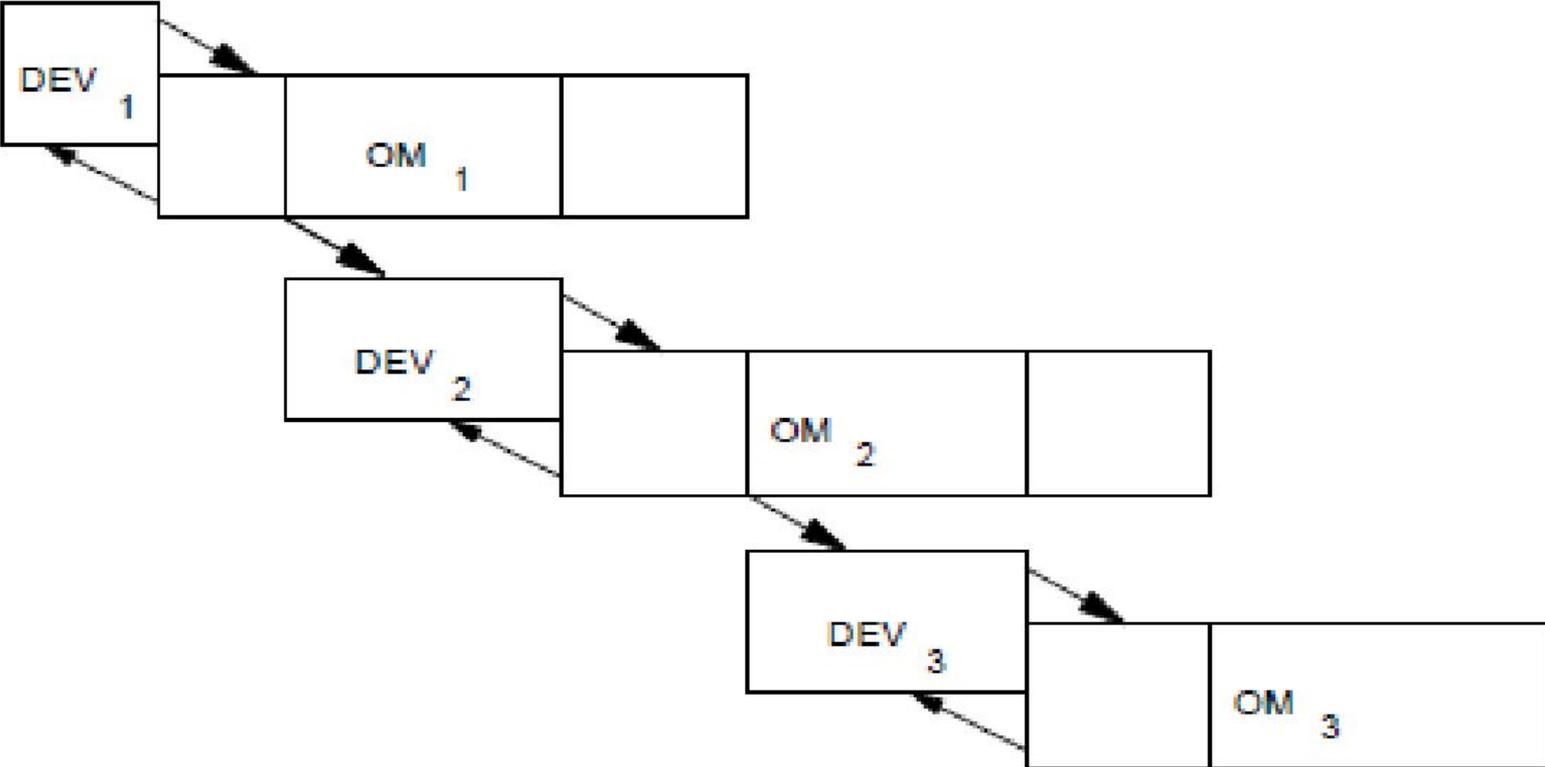
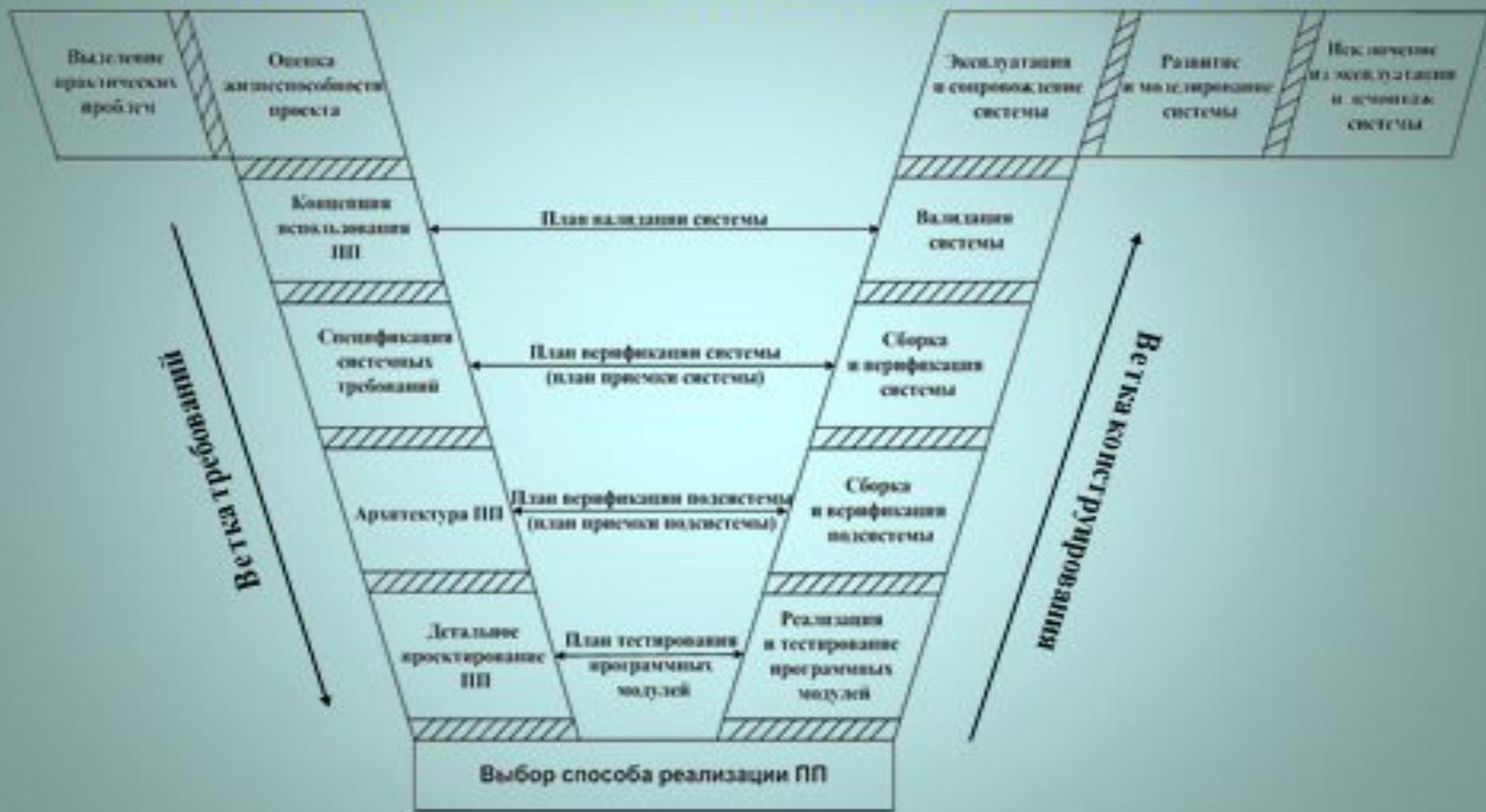
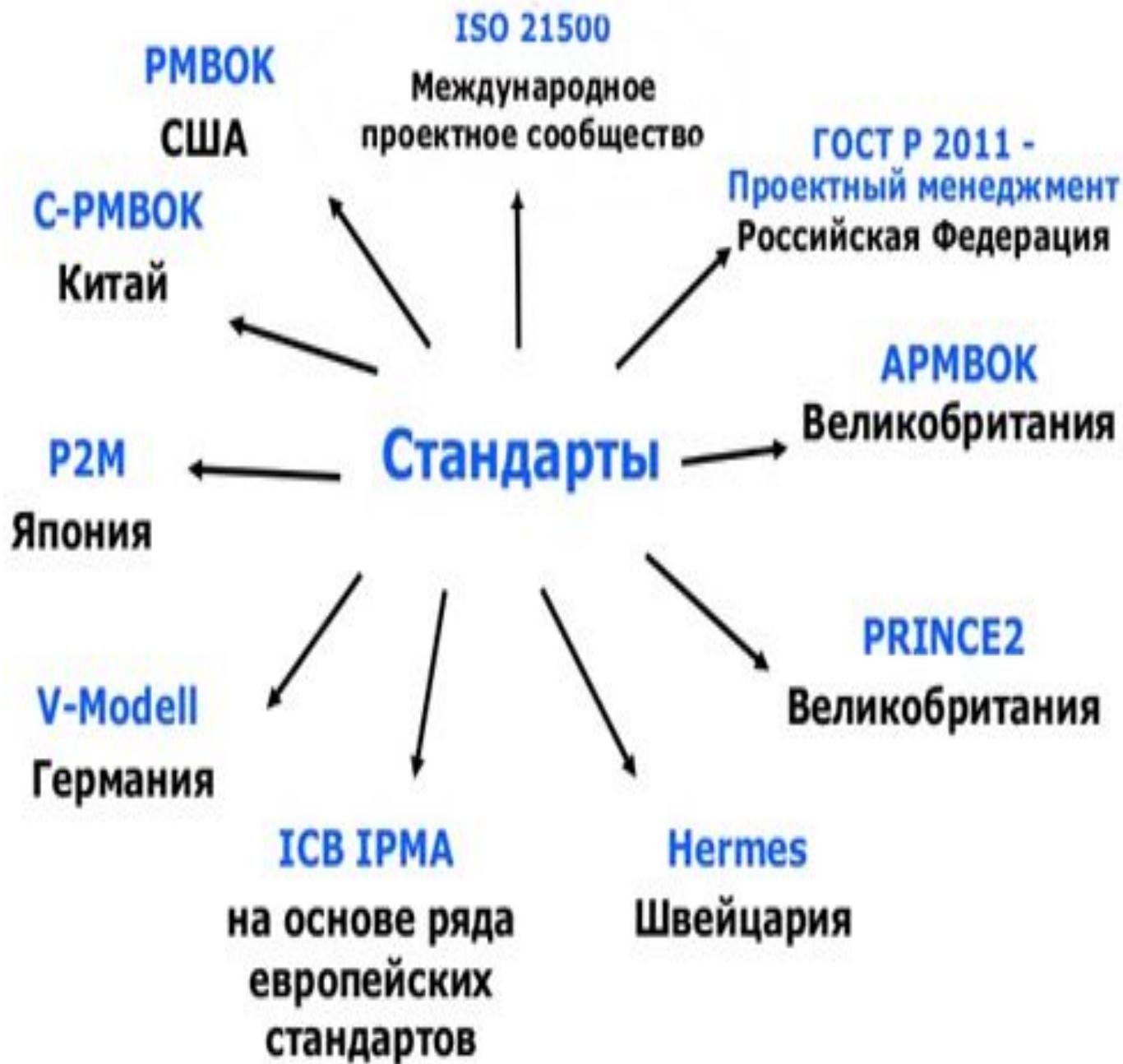


Figure 1.3.3 *The evolutionary development approach*

Общий вид V-модели жизненного цикла





ФЕДЕРАЛЬНОЕ АГЕНТСТВО
ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ



НАЦИОНАЛЬНЫЙ
СТАНДАРТ
РОССИЙСКОЙ
ФЕДЕРАЦИИ

ГОСТ Р
54870—
2011

Проектный менеджмент
ТРЕБОВАНИЯ К УПРАВЛЕНИЮ ПОРТФЕЛЕМ
ПРОЕКТОВ

Издательство «Стандарты-Информ»

Москва
Стандарты-Информ
2011

ФЕДЕРАЛЬНОЕ АГЕНТСТВО
ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ



НАЦИОНАЛЬНЫЙ
СТАНДАРТ
РОССИЙСКОЙ
ФЕДЕРАЦИИ

ГОСТ Р
54871—
2011

**Проектный менеджмент
ТРЕБОВАНИЯ К УПРАВЛЕНИЮ ПРОГРАММОЙ**

Издание официальное

Москва
Стандартинформ
2011

ФЕДЕРАЛЬНОЕ АГЕНТСТВО
ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ



НАЦИОНАЛЬНЫЙ
СТАНДАРТ
РОССИЙСКОЙ
ФЕДЕРАЦИИ

ГОСТ Р
54869 — 2011

Проектный менеджмент
**ТРЕБОВАНИЯ К УПРАВЛЕНИЮ
ПРОЕКТОМ**

Москва
2012

Структура стандартов ESA PSS-05-XX

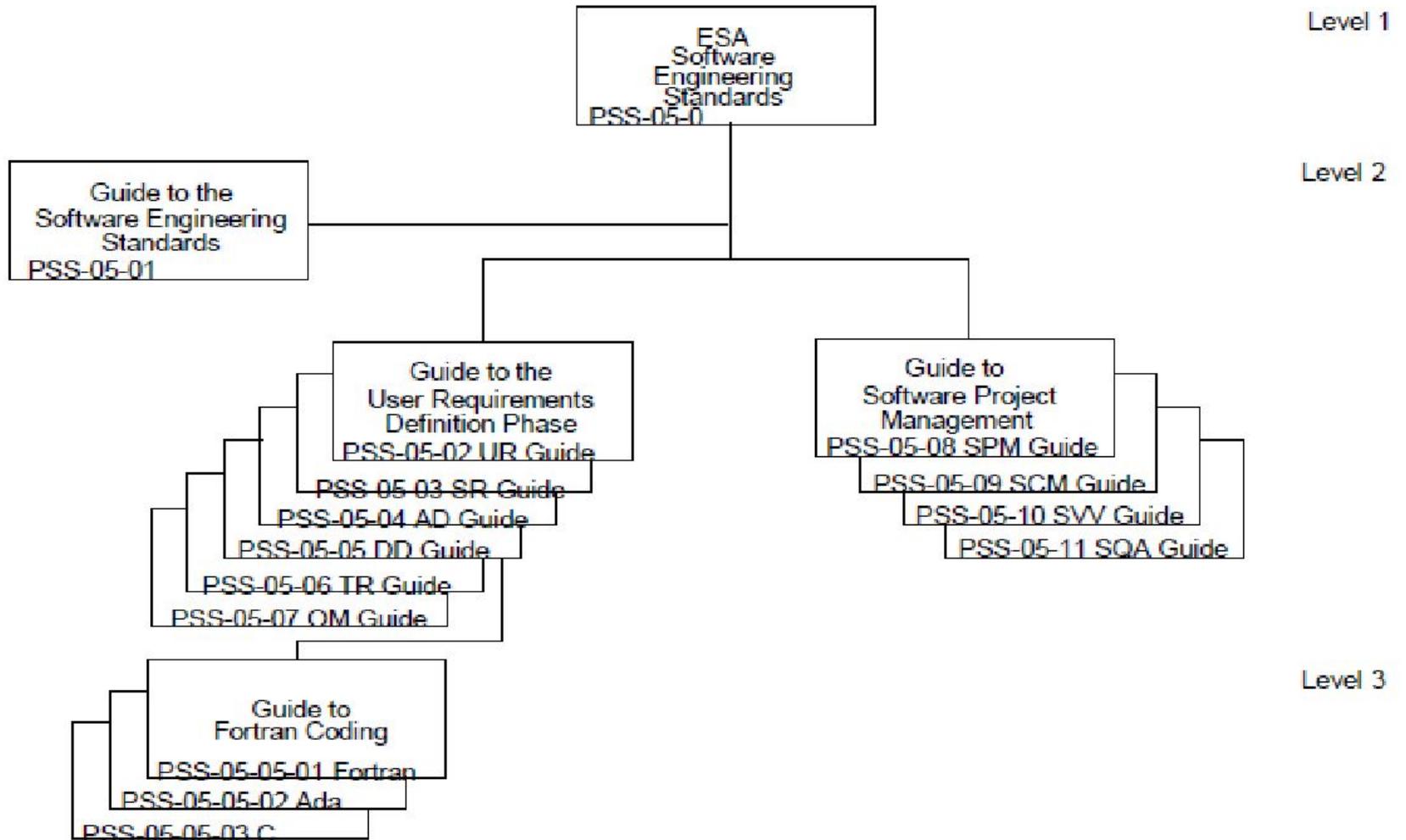


Figure 1: The ESA PSS-05-0 Document Tree

Содержание стандартов

Identifier	Title
ESA PSS-05-02	Guide to the User Requirements Definition Phase
ESA PSS-05-03	Guide to the Software Requirements Definition Phase
ESA PSS-05-04	Guide to the Architectural Design Phase
ESA PSS-05-05	Guide to the Detailed Design and Production Phase
ESA PSS-05-06	Guide to the Transfer Phase
ESA PSS-05-07	Guide to the Operations and Maintenance Phase

Identifier	Title
ESA PSS-05-08	Guide to Software Project Management
ESA PSS-05-09	Guide to Software Configuration Management
ESA PSS-05-10	Guide to Software Verification and Validation
ESA PSS-05-11	Guide to Software Quality Assurance

Identifier	Title
ESA PSS-05-05-01	Guide to Fortran coding
ESA PSS-05-05-02	Guide to Ada coding
ESA PSS-05-05-03	Guide to C coding

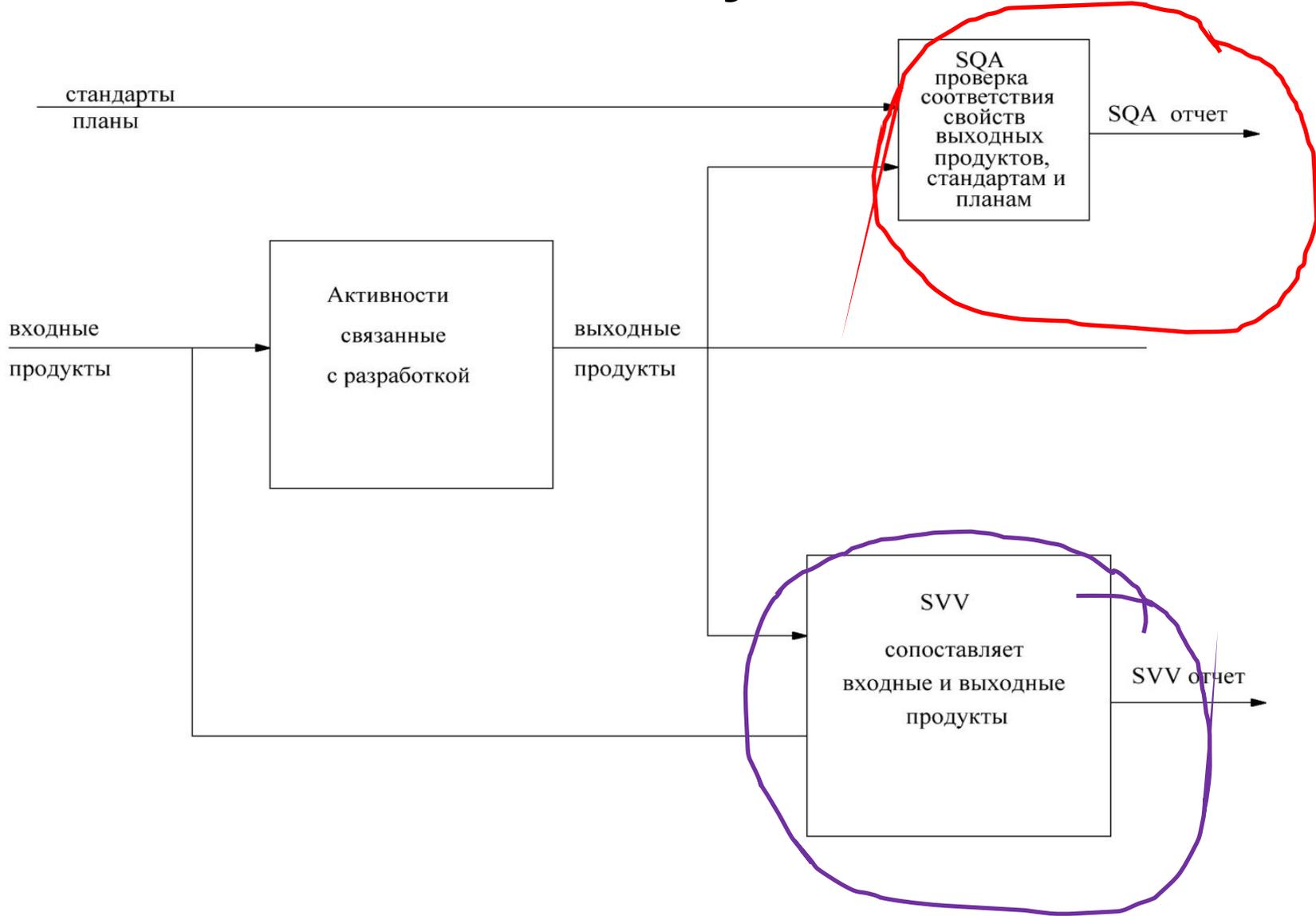
Методическая основа проектирования -
Document-driven approach

Следствие: ориентация на
разработку на основе планов

Концептуальная основа гарантированного управления качеством

- Основа управления качеством – планирование разноаспектной деятельности, связанной с реализацией проекта, стандартизация всех фаз жизненного цикла
- Планы разрабатываются в соответствие с требованиями стандартов
- Процедуры выполняются в соответствие с планами
- Продукты используются в соответствие со стандартами

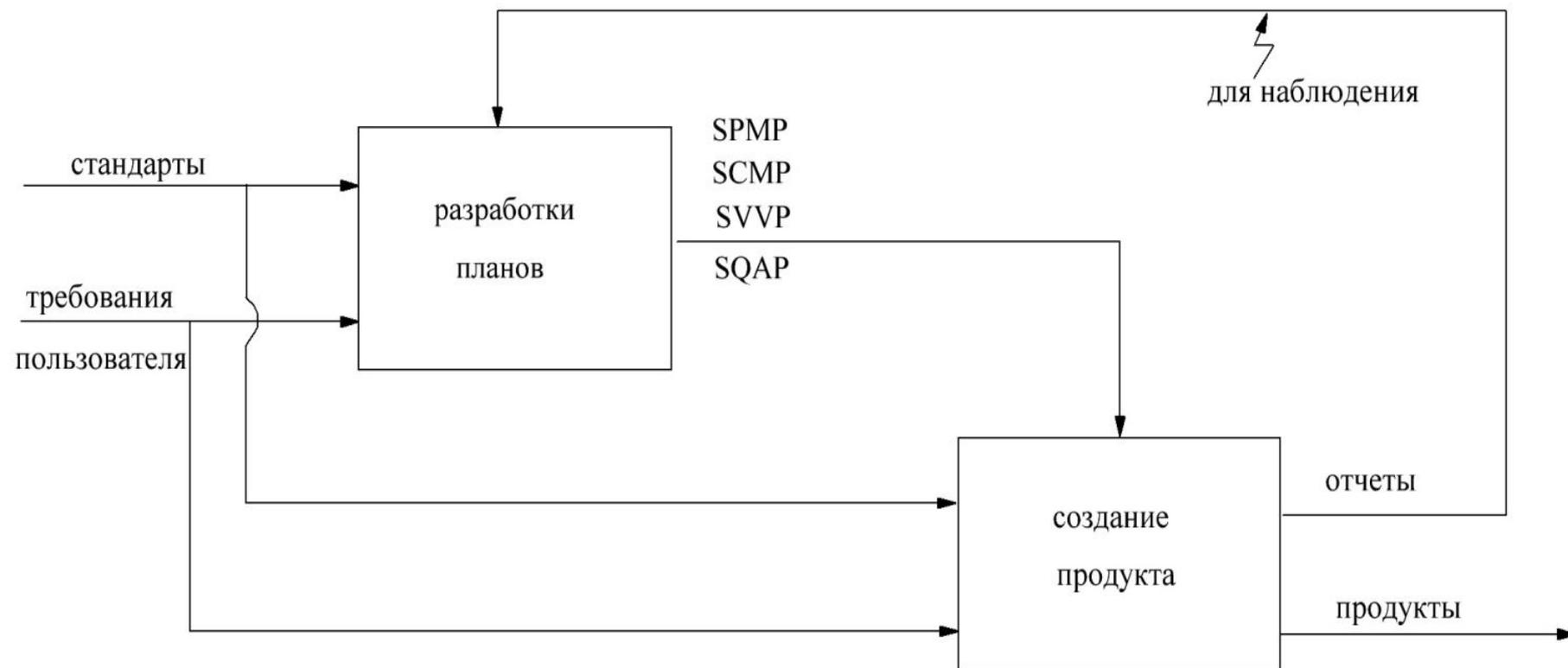
Различие между SQA и SVV



Про

Про

Петля обратной связи как инструмент контроля реализации проекта



SPMP- Software Project Management Plan

SCMP- Software Configuration Management Plan

SCMP- Software Verification and Validation Plan

SQAP- Software Quality Assurance Plan

Планирование проекта

Спецификация требований пользователей является основой разработки системы планов :

- План управления проектом
- План управления конфигурацией
- План управления верификацией

Компоненты плана проекта

- Приоретизированные цели проекта (основа разработки плана).
- Описание всех действий (WBS), а также оценок их предполагаемой стоимости
- Обоснование выбора модели жизненного цикла продукта
- Описание процессной модели (содержащей описание основных активностей), соответствующей фазам модели жизненного цикла
- Описание активностей:
 1. Входы
 2. Реализуемые задачи
 3. Выходы
 4. Критерии завершения

Критическая значимость функциональной безопасности субъектоцентрических систем

1 Ошибка в космическом агентстве

В июне 1996 года специалисты Европейского космического агентства осуществляли запуск ракеты Ariane 5.

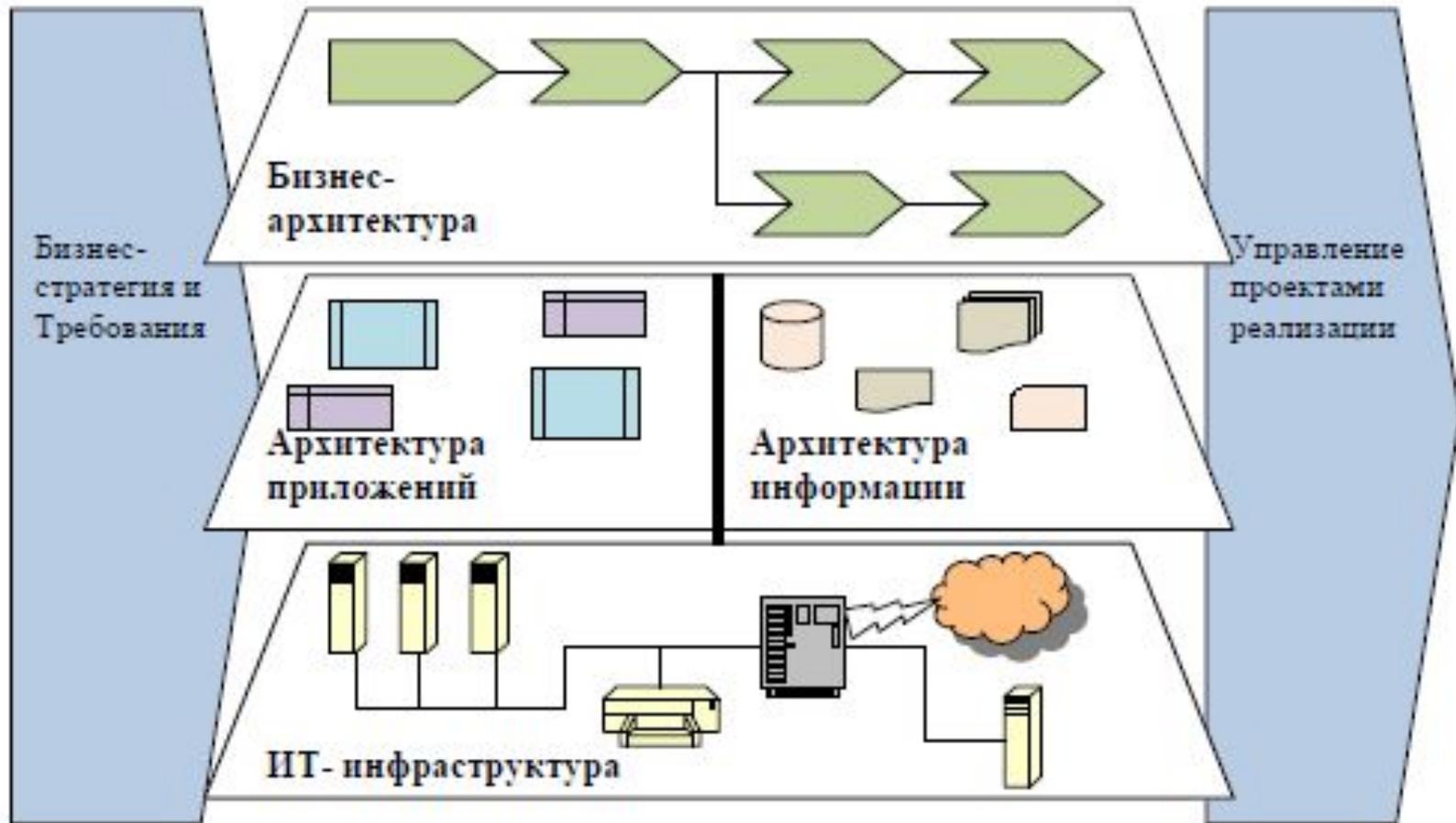


Ошибки программного обеспечения в космическом агентстве

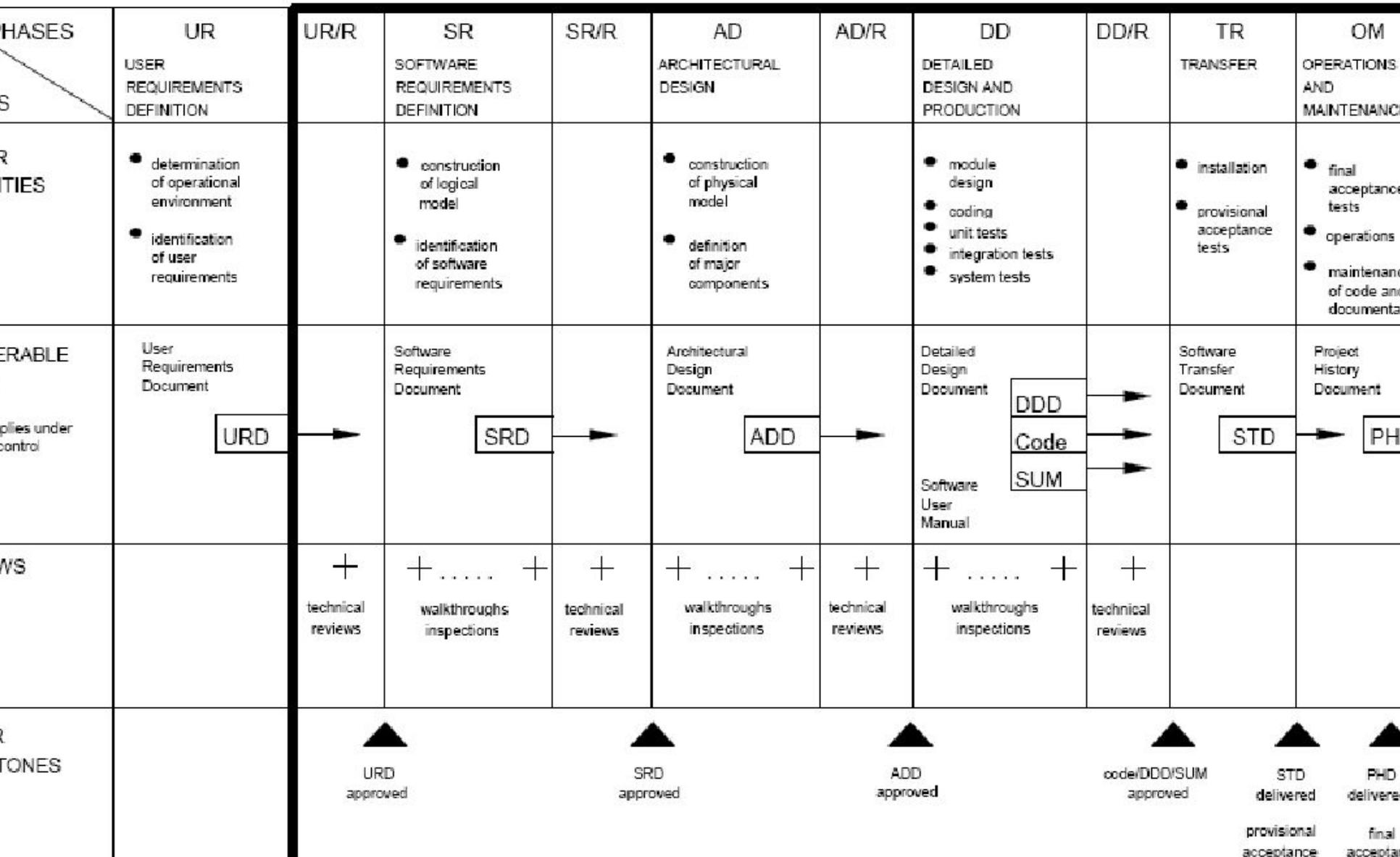


Ошибка в контролирующем программном обеспечении, написанном на языке программирования Ada, вызвало самоликвидацию ракеты через 37 секунд после взлета

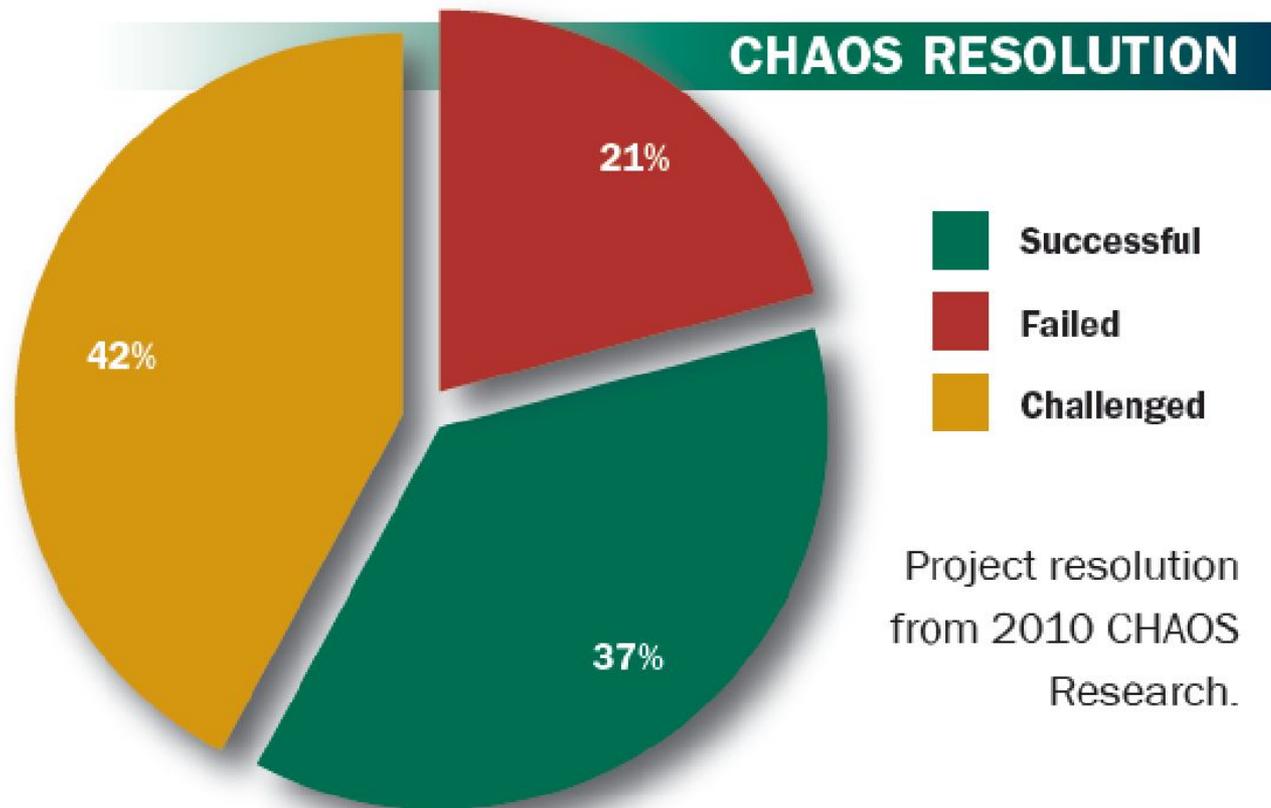
Содержание MDA



Место спецификации требований в жизненном цикле программной системы



Эффективность реализации программных проектов по данным 2010 г.



Динамика эффективности реализации программных проектов

RESOLUTION

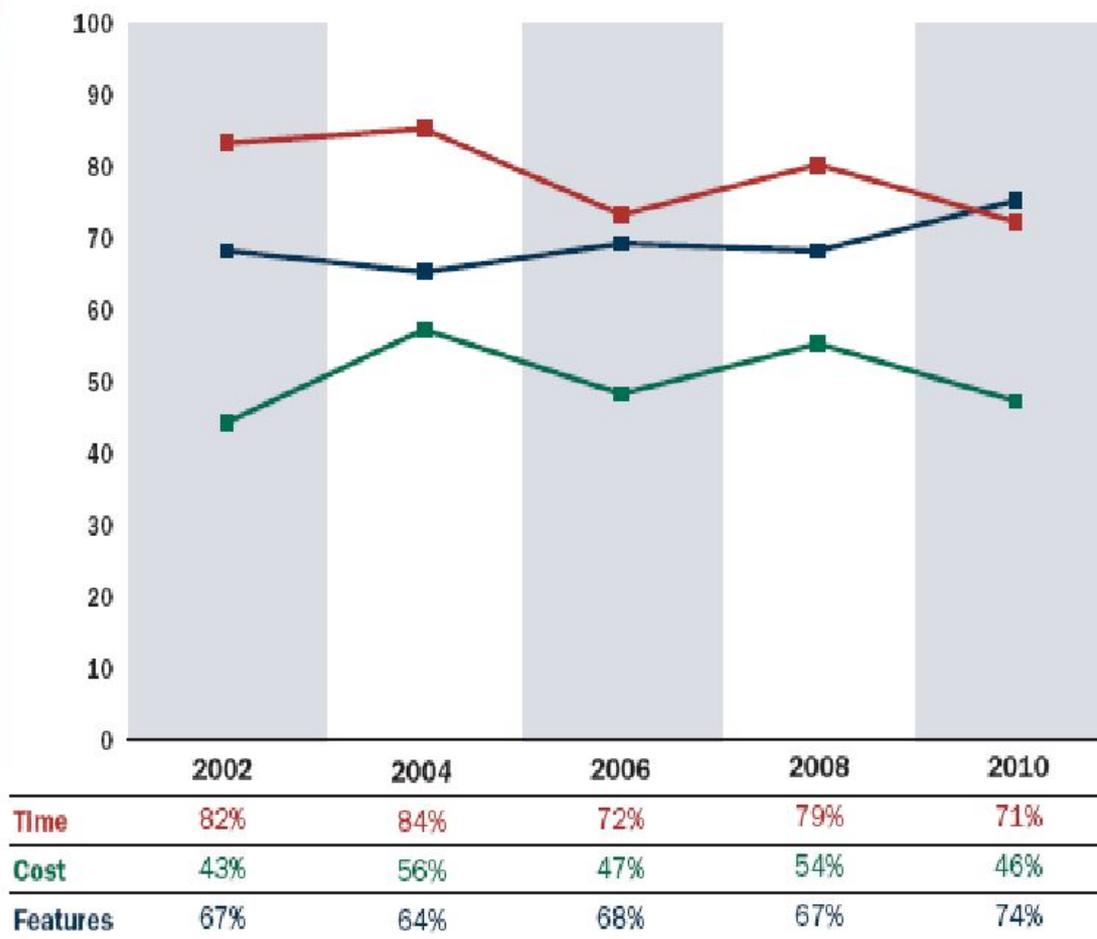
	2002	2004	2006	2008	2010
Successful	34%	29%	35%	32%	37%
Challenged	51%	53%	46%	44%	42%
Failed	15%	18%	19%	24%	21%

Project resolution results from CHAOS Research for years 2002 to 2010.

Последствия недостаточного качества реализации программных проектов

OVERRUNS AND FEATURES

Time and cost overruns, plus percentage of features delivered from CHAOS Research for the years 2002 to 2010.



**Статистические данные о эффективности
реализации программных проектов
(по данным, относящимся к США, продолжение)**

4. Среднее превышение плановой стоимости
-189%
5. Среднее превышение плановых сроков
реализации – 222%
6. Реализуется лишь 61% функциональных и
нефункциональных требований, заявленных в
техническом задании
7. Общие потери, учитывающие упущенные
возможности – триллион долларов

Основной вывод отчета «The Standish Group Report CHAOS. Project Smart», 2014

В настоящее время недочетов в программных проектах больше, чем было пять-десять лет назад, несмотря на радикальное повышение зрелости инструментальных средств и технологий реализации программных продуктов

Управление

Классический подход

Для принятия решения в условиях централизованной постановки задач, когда определены не только задачи, но и способы (варианты) их решения, время выполнения

Объект - люди, ресурсы, технологии (войска, ВВТ)

Субъект- ЛПР (ОВУ)

Законы, принципы, требования, методы

Сетецентрический подход

Интеграция коллективных возможностей, осведомленности, знаний, опыта и процедур принятия решений в условиях обстановки, характеризующейся рассредоточенностью войск, децентрализованным управлением, динамичностью и неопределенностью

Объект - сенсоры, акторы, информационно-управляющие элементы

Субъект- распределённая система ЭУ, привязанная к единому координатно-временному полю

Законы, принципы, требования, методы

По субъектам власти и управления (виды управления)

Государственное управление

Местное самоуправление

Общественное управление

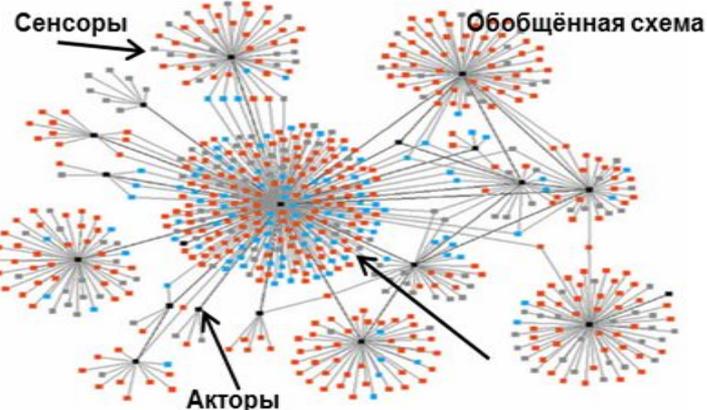
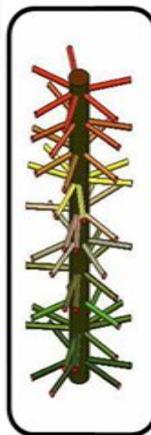
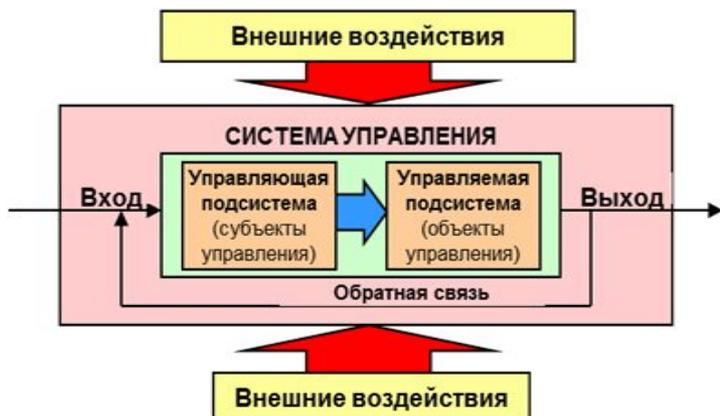
Социальная самоорганизация

Менеджмент

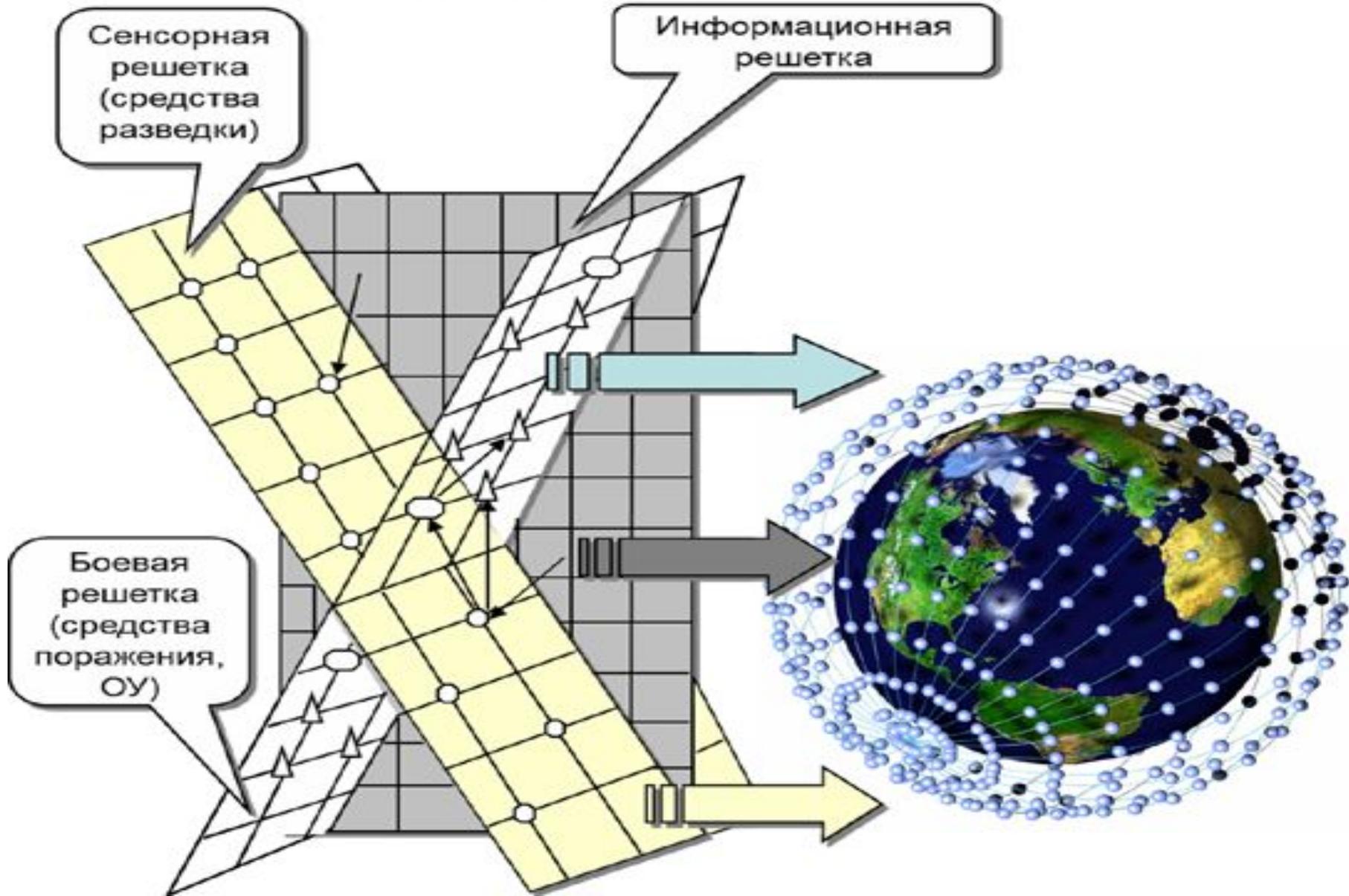
Индивидуальное самоуправление

Достоинства: простота построения, относительная дешевизна, жесткая иерархия в принятии решений в условиях определенности и т.п.
Недостатки: субъективность, низкая адаптивность к изменяющимся условиям, низкая живучесть при выводе из строя звена управления, низкая оперативность реагирования в условиях неопределенности и т.п.

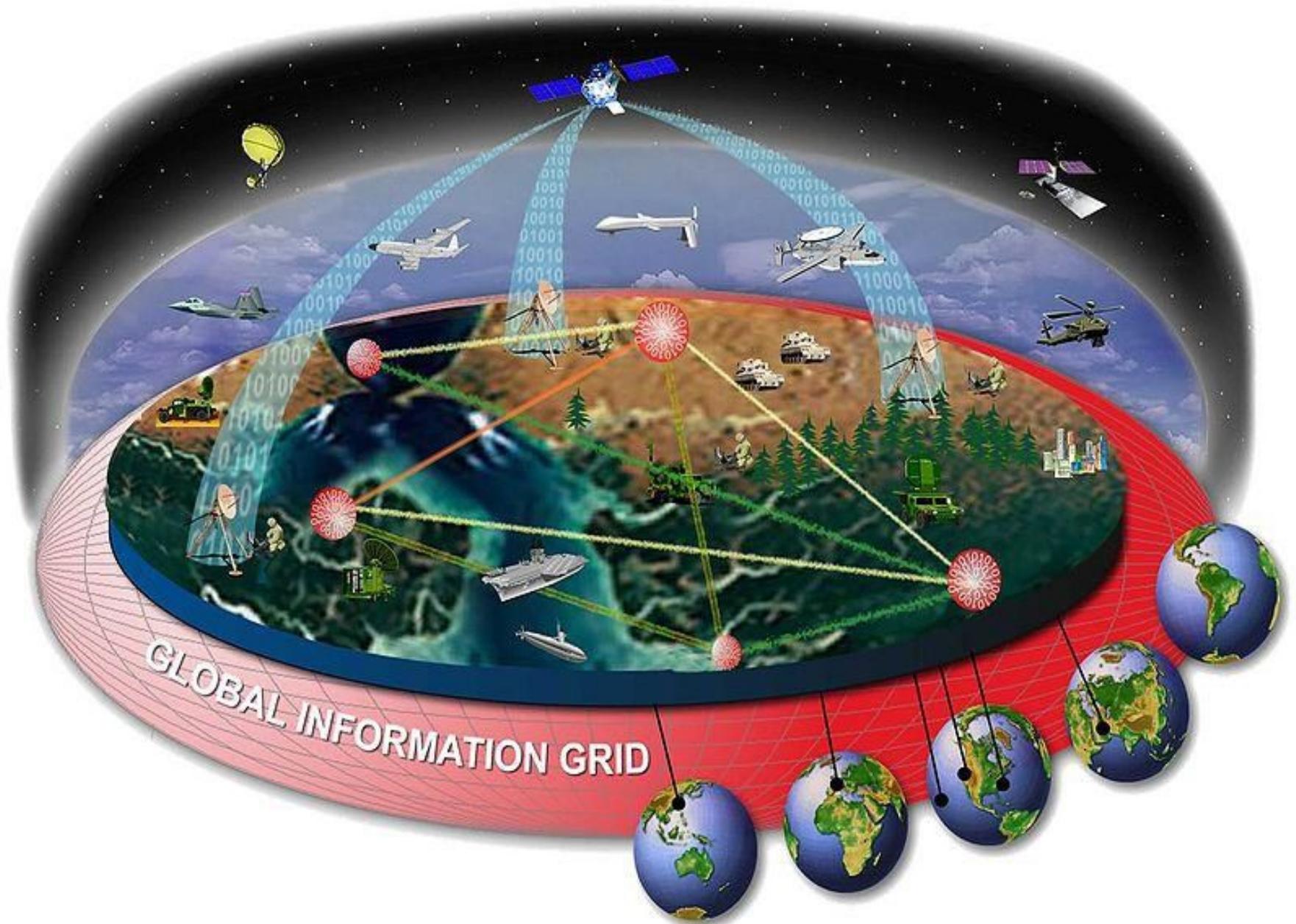
Достоинства: открытость, самоорганизация, слабая иерархия в контуре принятия решений, способность порождать цели внутри себя и т.п.
Недостатки: чрезмерная опора на высокие технологии, трудности обеспечения безопасности информации в сети, низкая теоретическая проработка, чрезмерная зависимость от информации, увеличивающаяся сложность военных систем, высокая стоимость и т.п.



Обобщённая модель сетецентрического подхода в военном деле



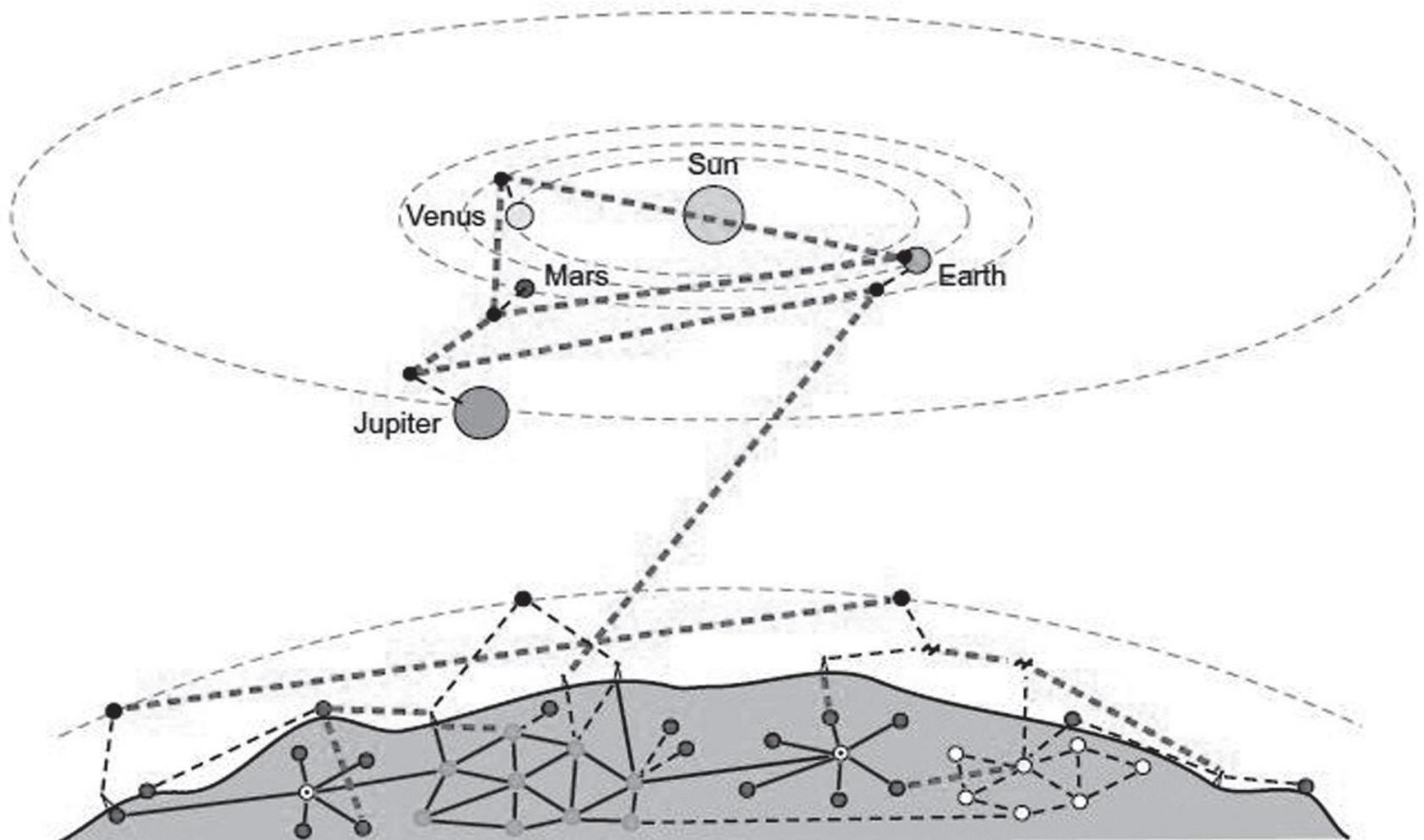
Идеальная GIG



(Delay) Tolerant Networking)

Основные проблемы, с которыми приходится сталкиваться в таких системах, состоят в следующем: задержки, искажения или разрывы связи.

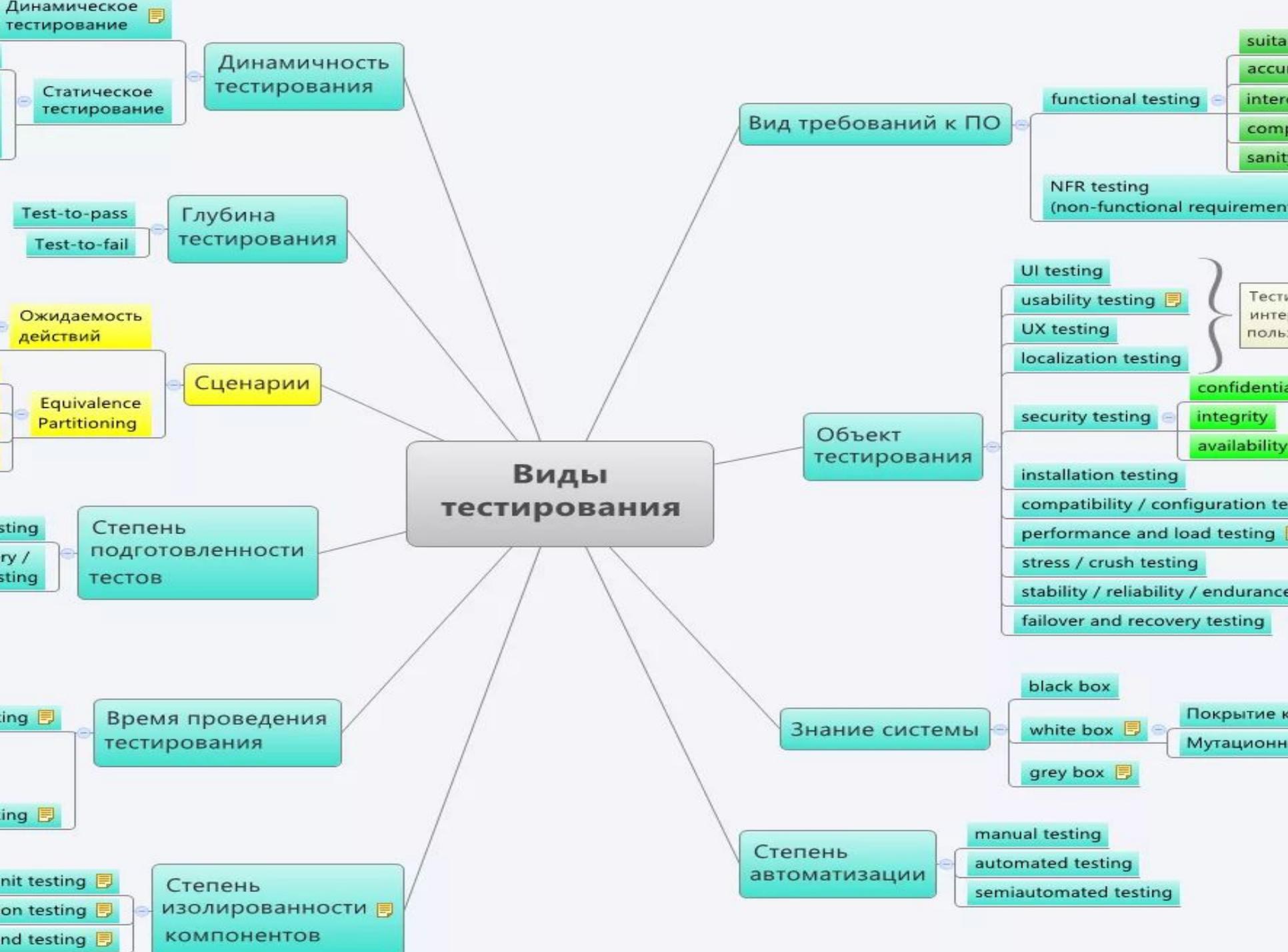
Схема информационных взаимодействий в сети DTN (Disruption



Реактивные подходы к управлению функциональной безопасностью

**обоснование множественности
концепций испытаний:**

**программная система есть
разновидность сложных систем**



Изменение парадигмы разработки
программных систем
(От классического до ad hoc)

Виды тестирования

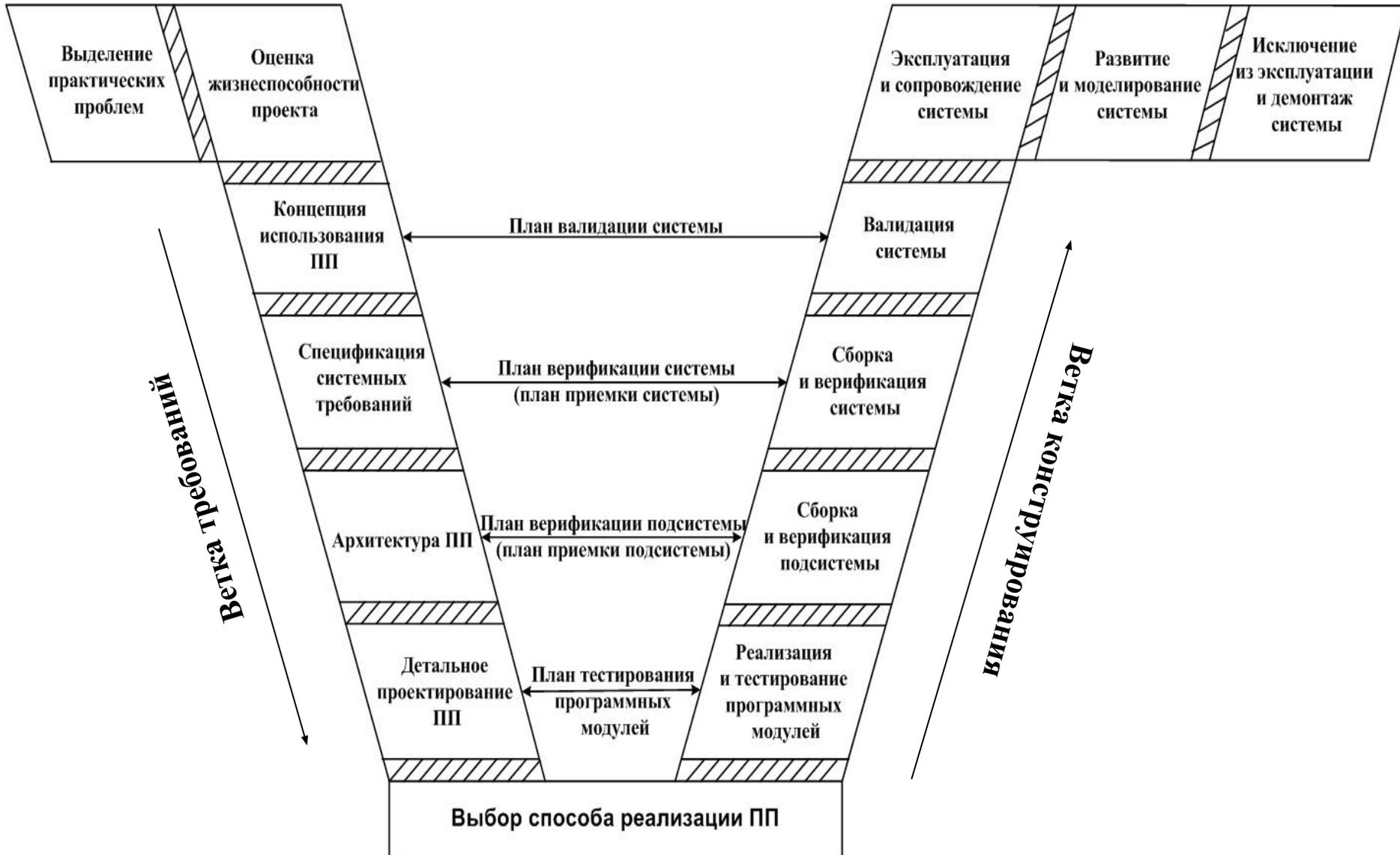




Способы тестирования

- Тестирование проводится не только на той стадии разработки программ, которая специально для этого предназначена, но и на предшествующих стадиях – при автономной отладке программ, еще до объединения их в единый программный комплекс. Такое тестирование называется **автономным**. Его обычно проводят сами разработчики, которые проверяют точное соответствие программы выданной им спецификации.
- **Комплексное тестирование** призвано проверить все аспекты работы программы от правильности взаимодействия внутренних программных компонентов до правильности взаимодействия программного комплекса с его пользователями.
- Во время **пользовательского тестирования** результаты работы программы проверяются с прикладной точки зрения.
- **Техническое тестирование** дает возможность проверить безопасную и эффективную работу созданной программы в нормальном и пиковом режимах ее использования. Функциональность на этом этапе проверяется только в смысле ее влияния на важнейшие технические параметры программы, например, на время реакции системы на запрос пользователя.
- Важной в тестировании является возможность проведения **регрессивного тестирования**. Регрессивные тесты, повторяемые после каждого исправления программы, позволяют убедиться, что функциональность программы, не связанная с внесенным исправлением, не затронута этим исправлением и не утрачена из-за него.

Общий вид V-модели жизненного цикла

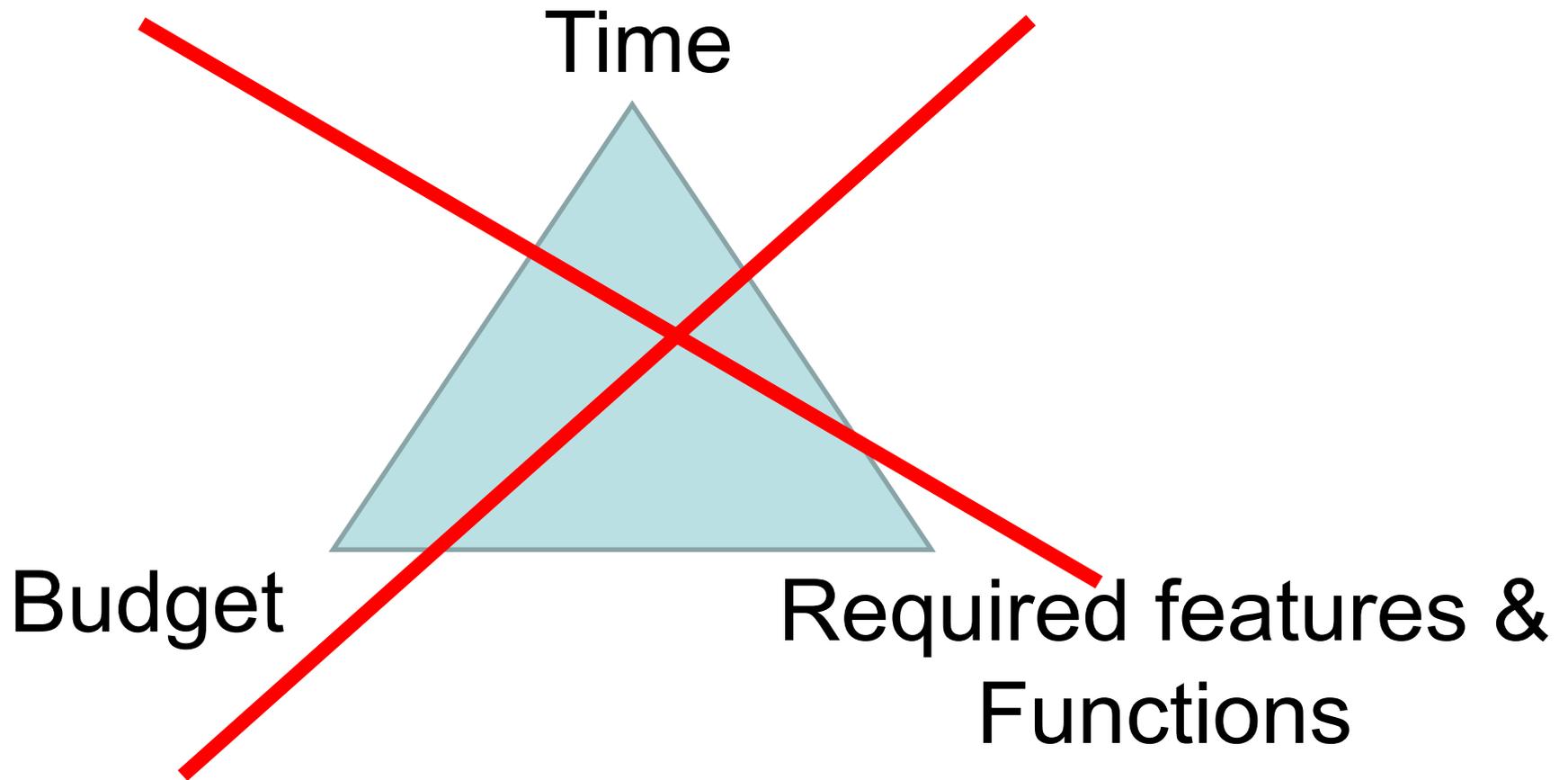


Сценарное тестирование

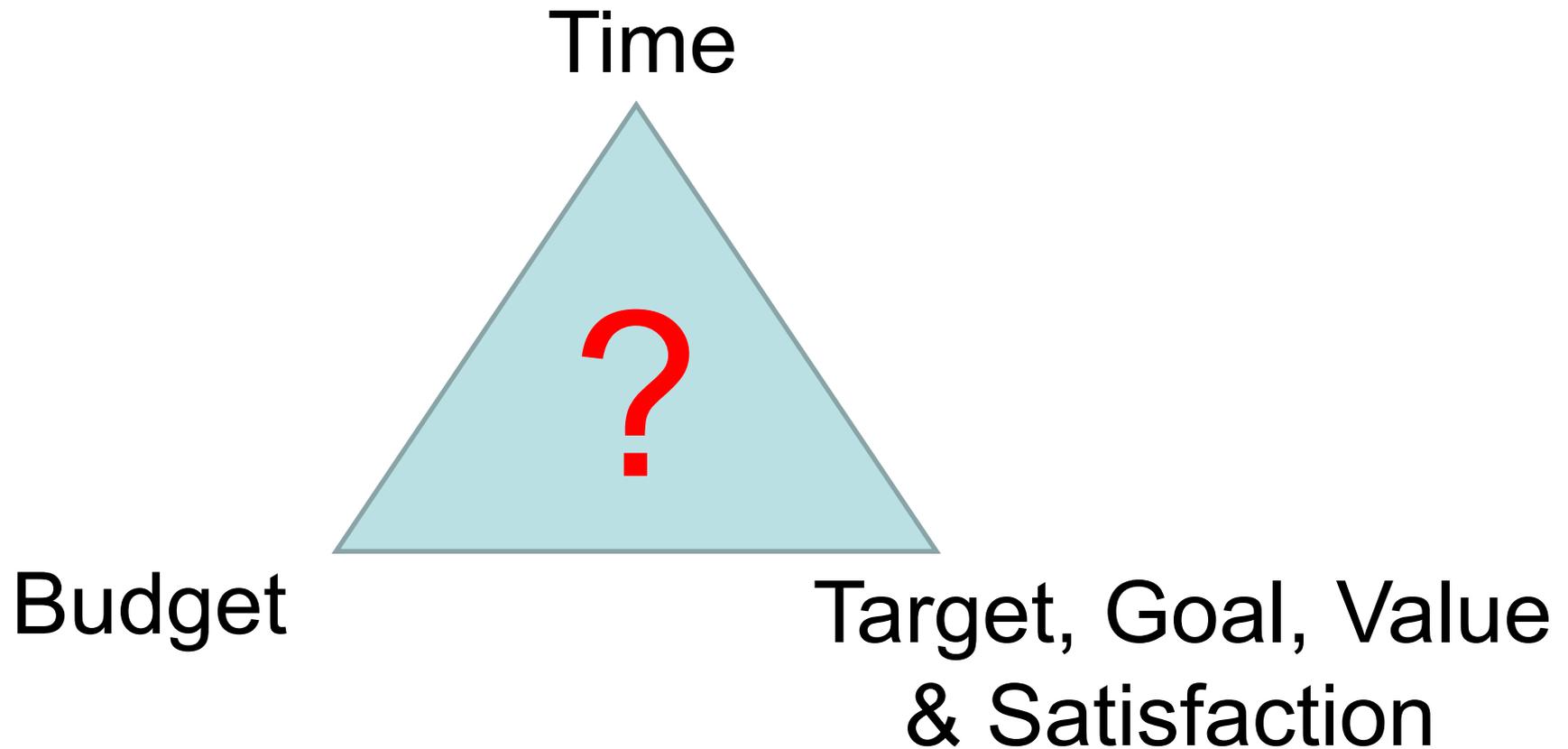
Сценарное тестирование-
классическое тестирование
по предварительно написанным
и задокументированным сценариям.

В пользу сценарного тестирования:
сравнительная легкость планирования:
тест-кейсы можно легко поделить между
различными тестировщиками или
командами.

Project Triangle (PMI-1994)



Project Triangle (Standish Group-2015)

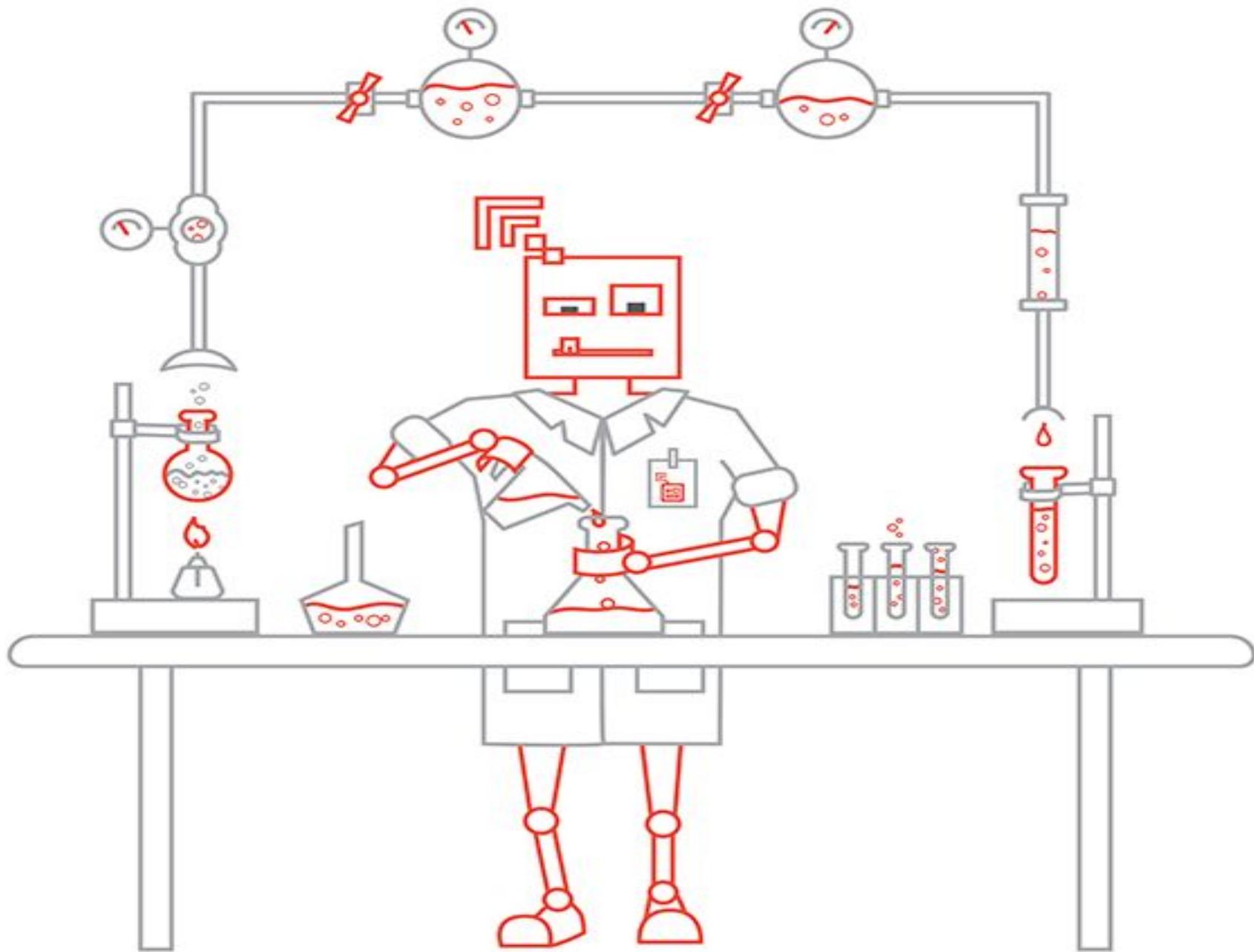


Ad hoc тестирование



Содержание Ad hoc тестирования

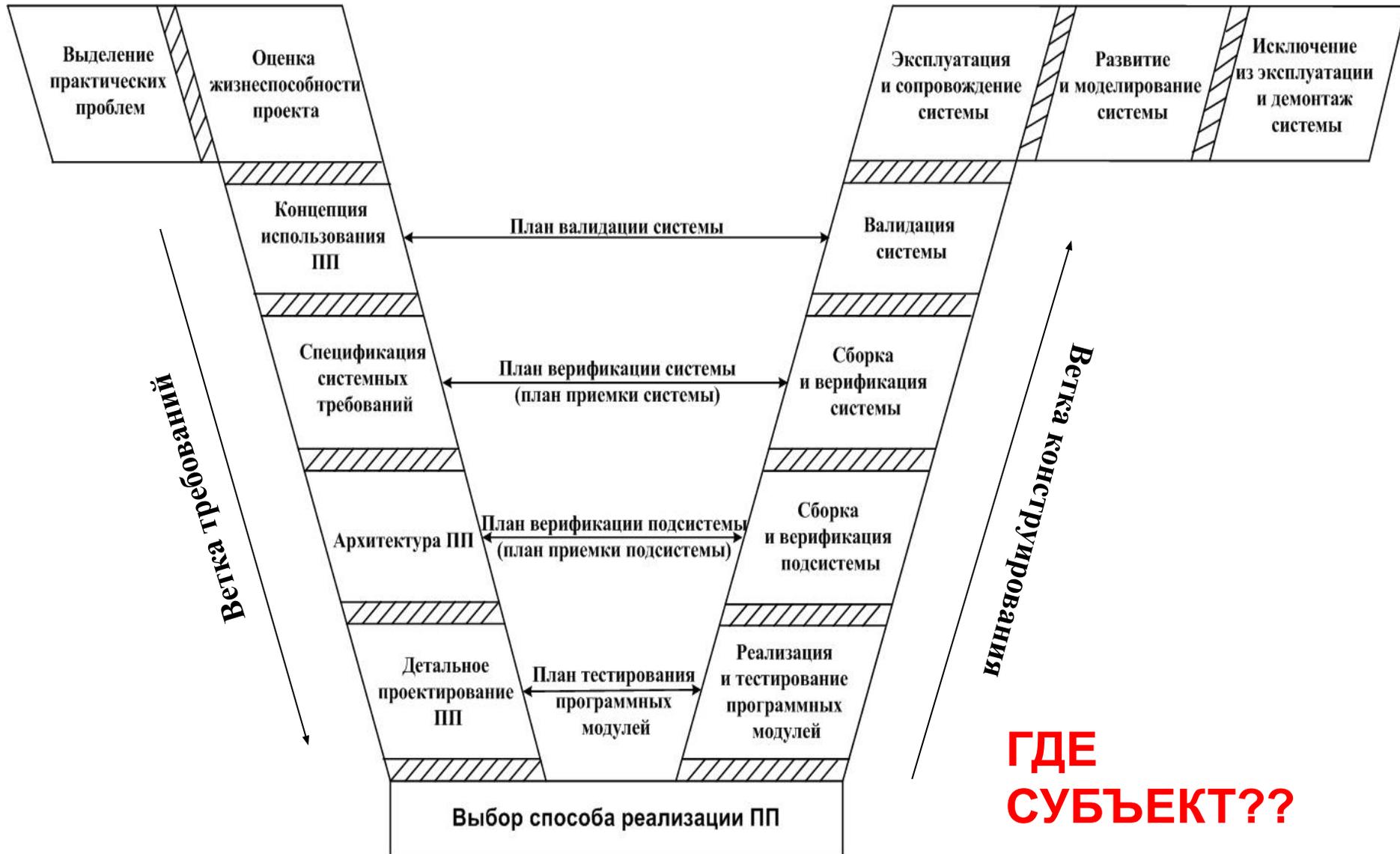
Свободное тестирование (ad-hoc testing) – это вид тестирования, который выполняется **без подготовки к тестированию продукта, без определения ожидаемых результатов, проектирования тестовых сценариев**. Это неформальное, импровизационное тестирование. Такой способ тестирования в большинстве случаев дает большее количество заведенных отчётов об ошибке. Это обусловлено тем, что тестировщик на первых шагах приступает к тестированию основной функциональной части продукта и выполняет как позитивные, так и негативные варианты возможных сценариев.



Понятие исследовательского тестирования

Исследовательское тестирование (exploratory testing) — это одновременное изучение программного продукта, проектирование тестов и их выполнение. Это неформальный метод проектирования тестов, при котором тестировщик активно контролирует проектирование тестов и то, как эти тесты выполняются, и использует полученную во время тестирования информацию для проектирования новых тестов. Если каждый следующий тест, который выполняет тестировщик, выбирается по результатам предыдущего теста, это означает, что мы используем исследовательское тестирование.

Общий вид V-модели жизненного цикла



ВИДЫ тестирования





Тимофеев Алексей Николаевич
Генеральный директор ООО .САТУРС., к.т.н..

ика

тирования систем

научно-образовательный журнал, 2017

Почему падают ИТ-проекты?

Эвергетика

Виттих Владимир Андреевич

9.07.1940-18.08.2017



Эвергетика – субъективно-ценностно-ориентированная наука о процессах интерсубъективного управления в сложных системах

Центральным понятием эвергетики является «**неоднородный актер**» – субъект, вовлеченный в урегулирование проблемной ситуации

Вовлеченность означает заинтересованность субъекта в изменении ситуации и обладание полезными, с точки зрения урегулирования ситуации, ресурсами

Термином «неоднородный актер» подчеркивается то обстоятельство, что одна и та же ситуация **по-разному** воспринимается разными актерами

Жизненный мир

На каких уровнях приходится искать решения при урегулировании проблемных ситуаций?

Подходы к управлению сложными системами
(урегулированию проблемных ситуаций)

Структура системы
(плодотворное объяснение)

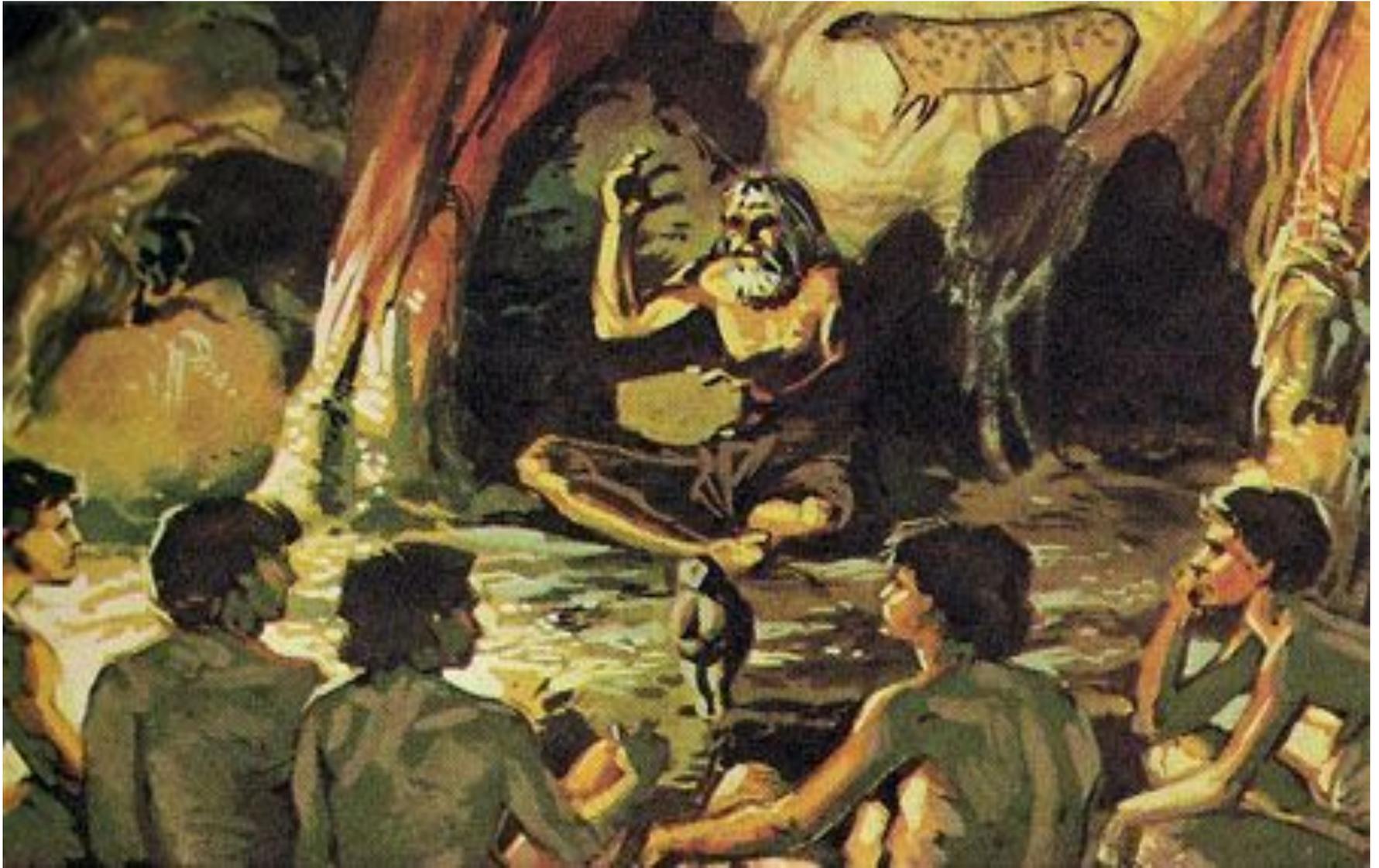


Закономерности поведения
(гибкое объяснение)



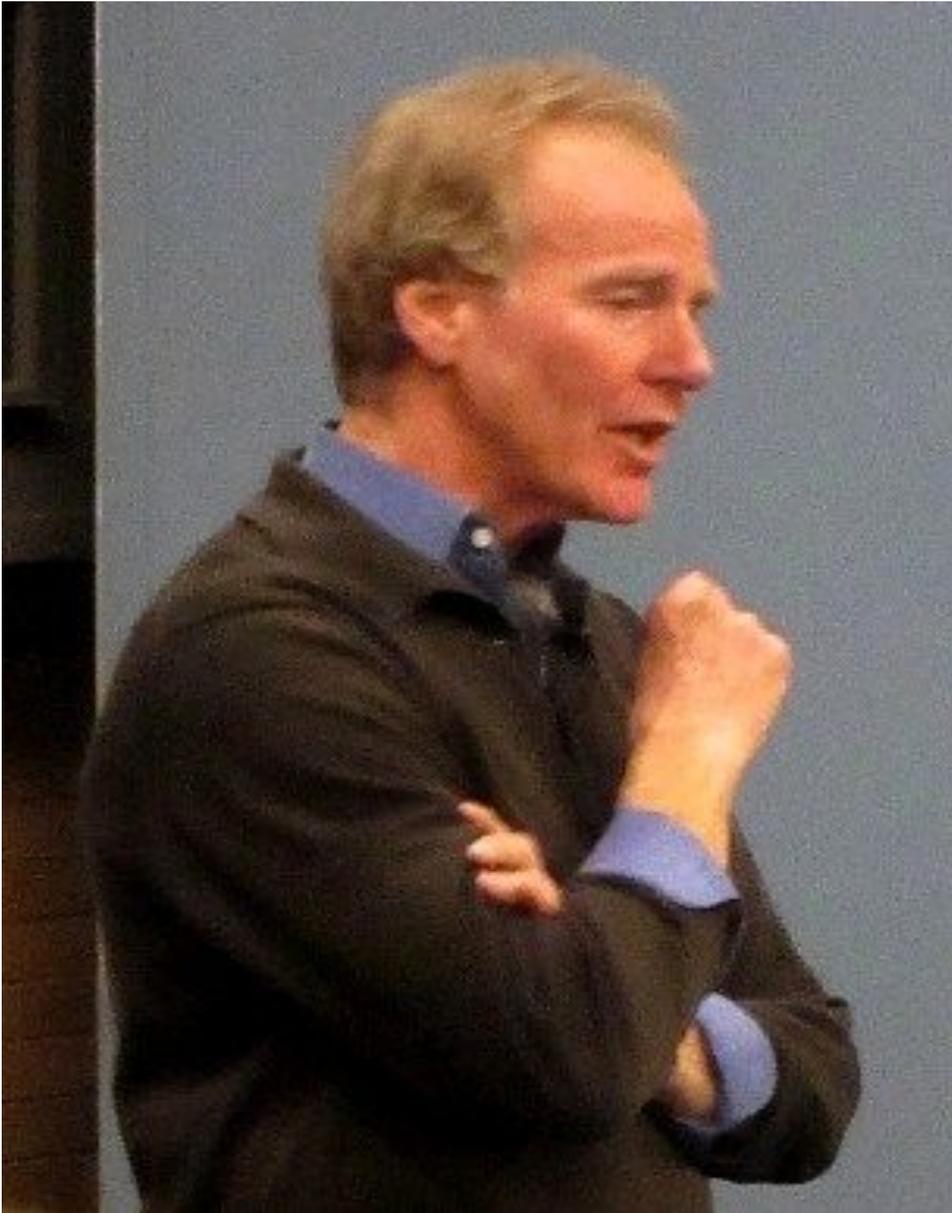
Ссылки на давление событий
(механическая реакция)

Обыденный подход (Commonplace approach)



Пещера Платона





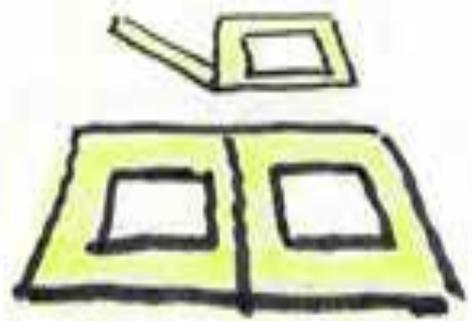
Born	1947 Stanford, California
Alma mater	MIT Ph.D, 1978; M.S., 1972 Stanford University B.S.
Known for	The Fifth Discipline , Learning organization
Scientific career	
Fields	Systems science

Подходы отдельных акторов к урегулированию проблемной ситуации, обусловлены различием персональных онтологических моделей (персональных интеллектуальных моделей)

...Двое с разными *интеллектуальными* моделями смотрят на одно и то же, но описывают его по-разному, потому что подмечают разные детали...

...*Интеллектуальные модели* – это то, что формирует наши действия, что формирует наши представления об окружающей действительности...

P. Senge



Искусство системного мышления

Необходимые знания
о системах и творческом
подходе к решению проблем

**Джозеф О'Коннор
и Иан Макдермотт**

Перевод с английского

7-е издание



Москва
2013

Особенности системного мышления

1. Решение проблемы в первую очередь требует *преодоления мышления, порождающего проблемы.*

Системное мышление — не просто комплексное и всестороннее, оно и вертикальное, и горизонтальное, и глубокое, и цикличное

2. Проблемы — порождение событий и того, что мы о них думаем. Мы сами представляем собой неременный элемент всех своих проблем и, как сказал Эйнштейн , не в состоянии решить проблему, оставаясь на том же уровне мышления, который ее породил.

**В чем ограниченность узких
специалистов???**

«...Новое платье короля» – это классический рассказ не о людской глупости, а об интеллектуальных моделях, застилающих глаза людей. Только представление о королевском достоинстве мешают видеть, что король голый...»

P. Senge

Ментальные модели

Любую нашу деятельность направляют глубоко укоренившиеся идеи, стратегии, способы понимания и руководящие идеи. В литературе по системному мышлению они известны как *ментальные модели* .

Они были полезны в прошлом и, как мы надеемся, пригодятся в будущем.

Глубоко укоренившиеся в нас ментальные модели определенным образом организуют наше восприятие мира.





- А здорово ты придумал - пиво с раками!

Известный английский исследователь технического творчества, автор фундаментального метода проектирования Э. Мэтчетт , в основу своего курса обучения в так называемой «Школе Мэтчетта» в Бристоле положил такое определение: «Хороший проект – это оптимальное решение, удовлетворяющее *сумме истинных потребностей (т.е. ассоциируемых с разными правообладателями)* в конкретном комплексе обстоятельств».

Проблема согласования интересов неоднородных акторов в результате компромисса

(Питер Пауль Рубенс. Суд Соломона)

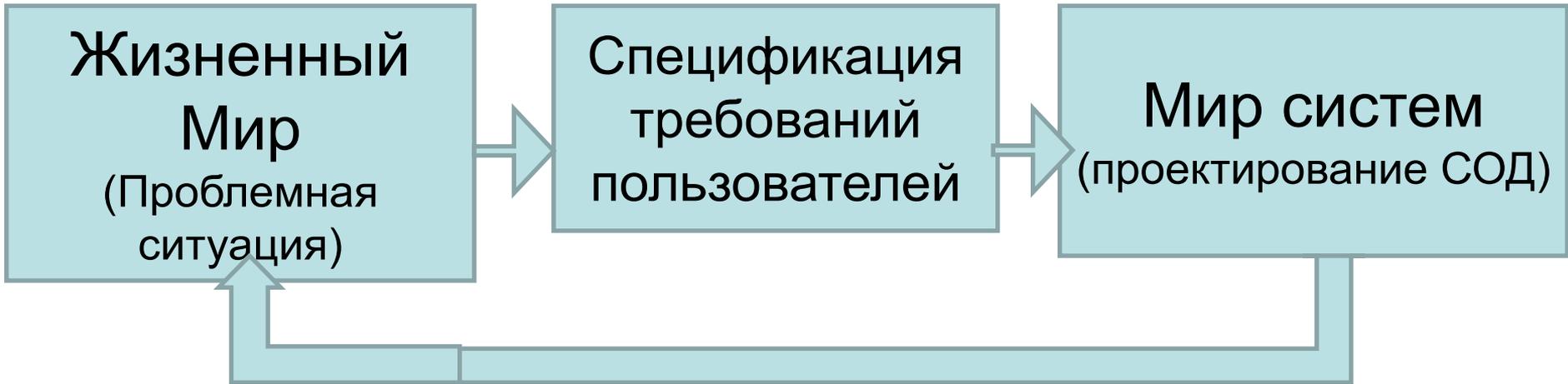


Условия нахождения консенсуса

1. Более системно видеть и понимать мир
2. Размышлять о неявных предпосылках (неявных целях системы)
3. Говорить о собственных целях и мечтах
4. Слушать других, когда они говорят о своем
5. Быть внимательным к тому, как другие воспринимают мир

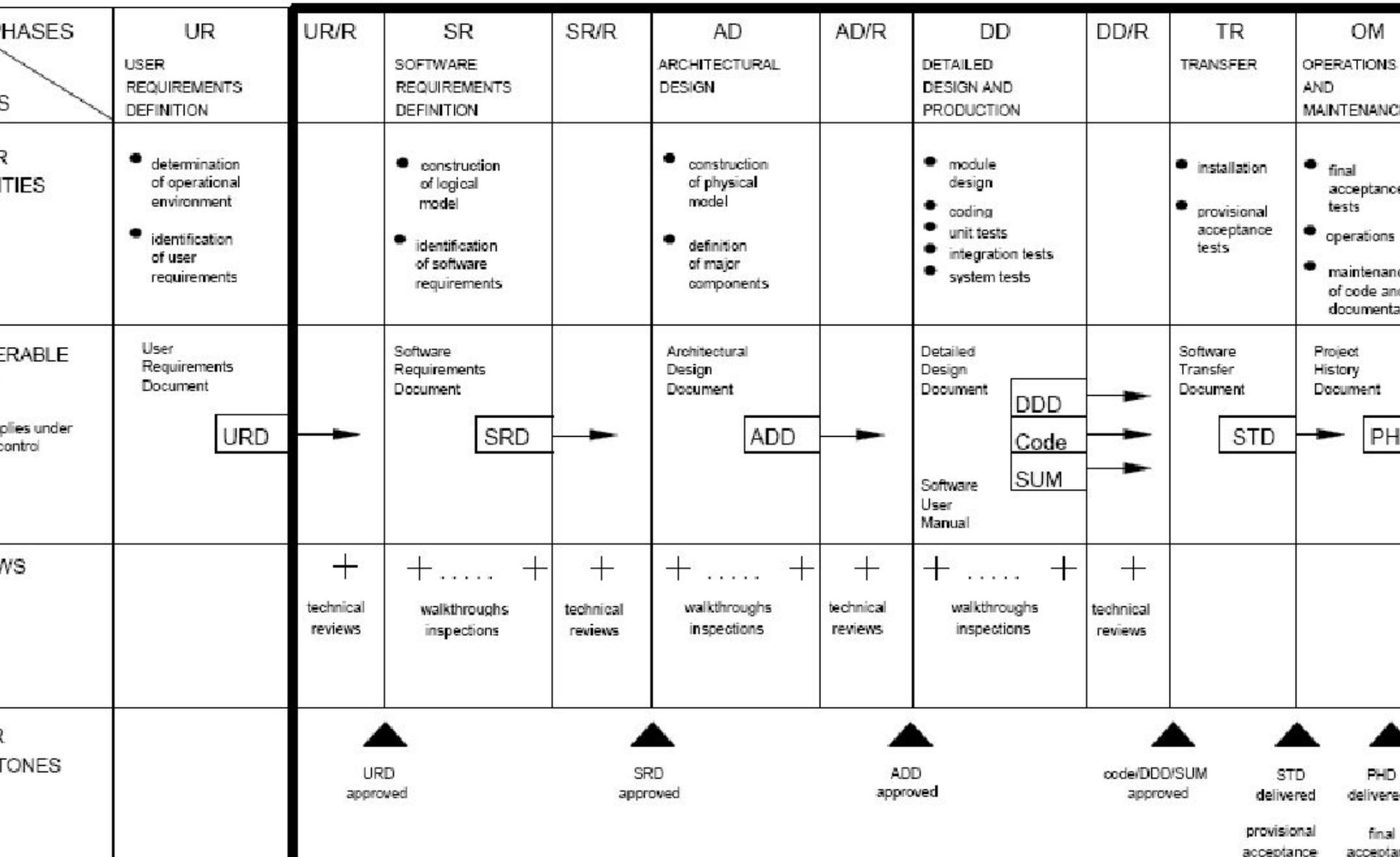
**Основное условие формирования
эффективной стратегии урегулирования
проблемной ситуации – целостное
восприятие проблемной ситуации**

Контур урегулирования проблемной ситуации



Спецификация внешнего
облика системы
обработки данных
(результатов предпроектного
обследования) - «мост»
между жизненным миром
(слабо формализуемым) и
миром систем (более сильно
формализуемым)

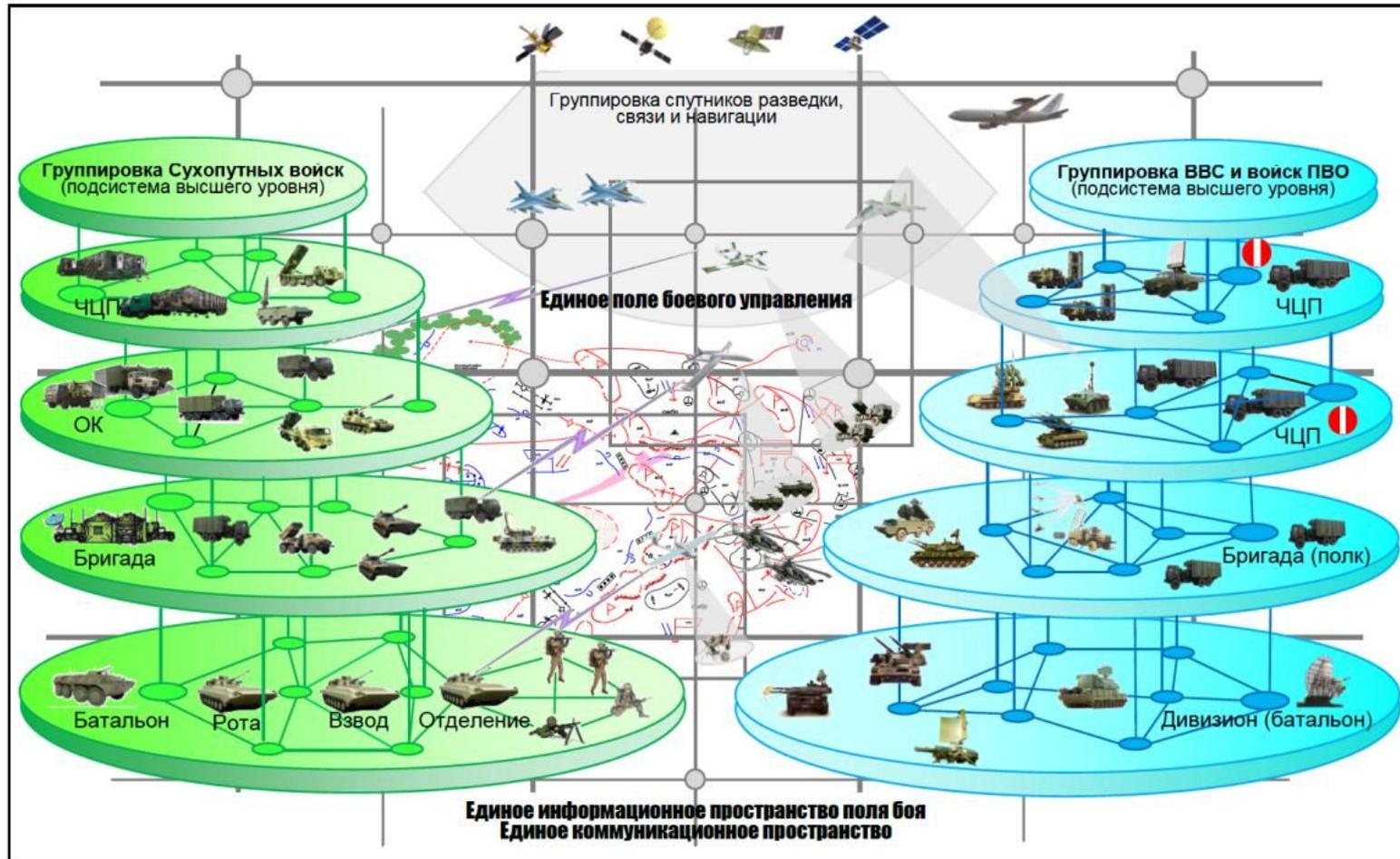
Место спецификации требований в жизненном цикле программной системы



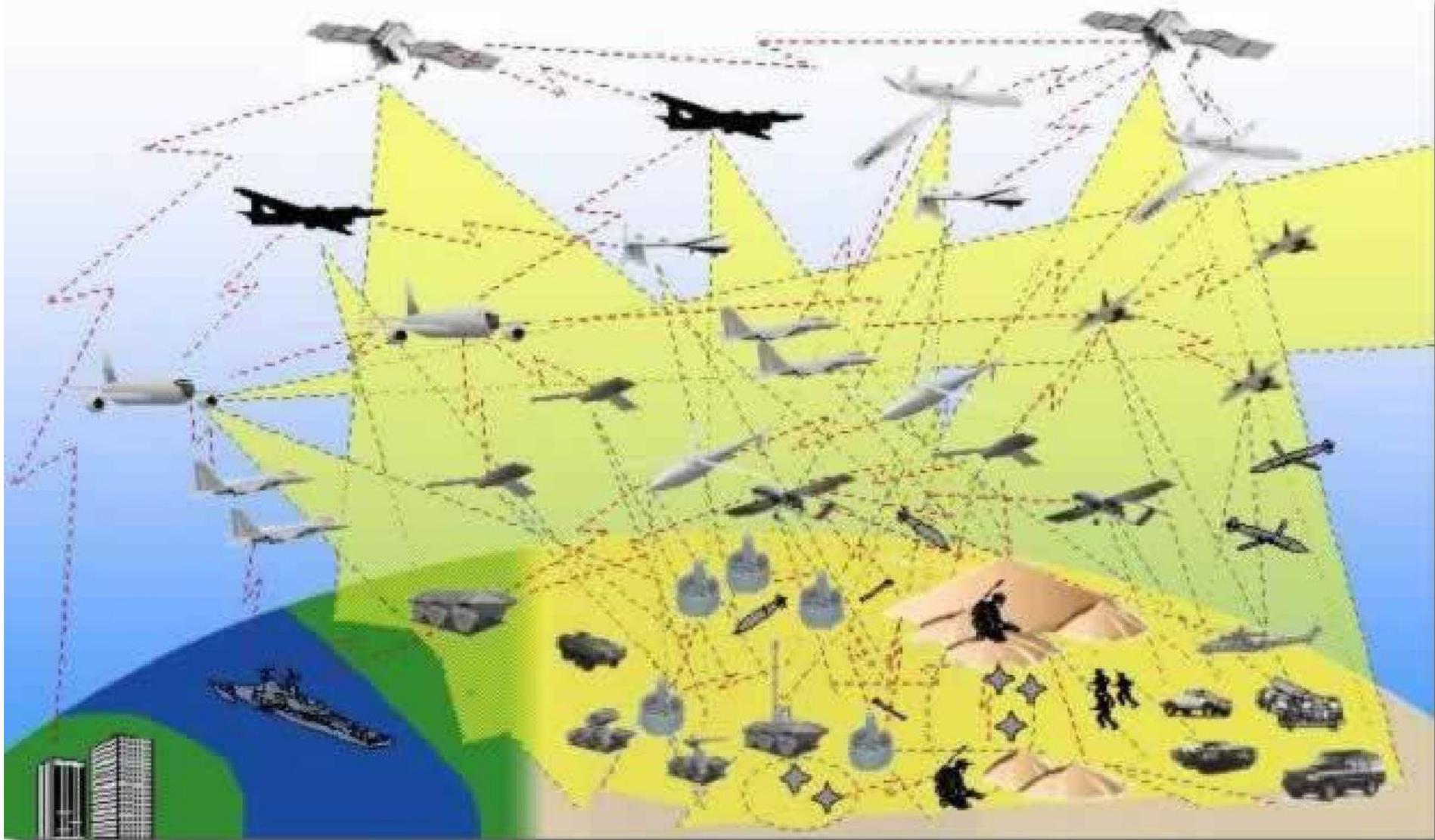
Текущее положение дел

сложность объекта управления
(сетевое управление распределенными
гетерогенными
системами)

Масштаб и сложность системы систем

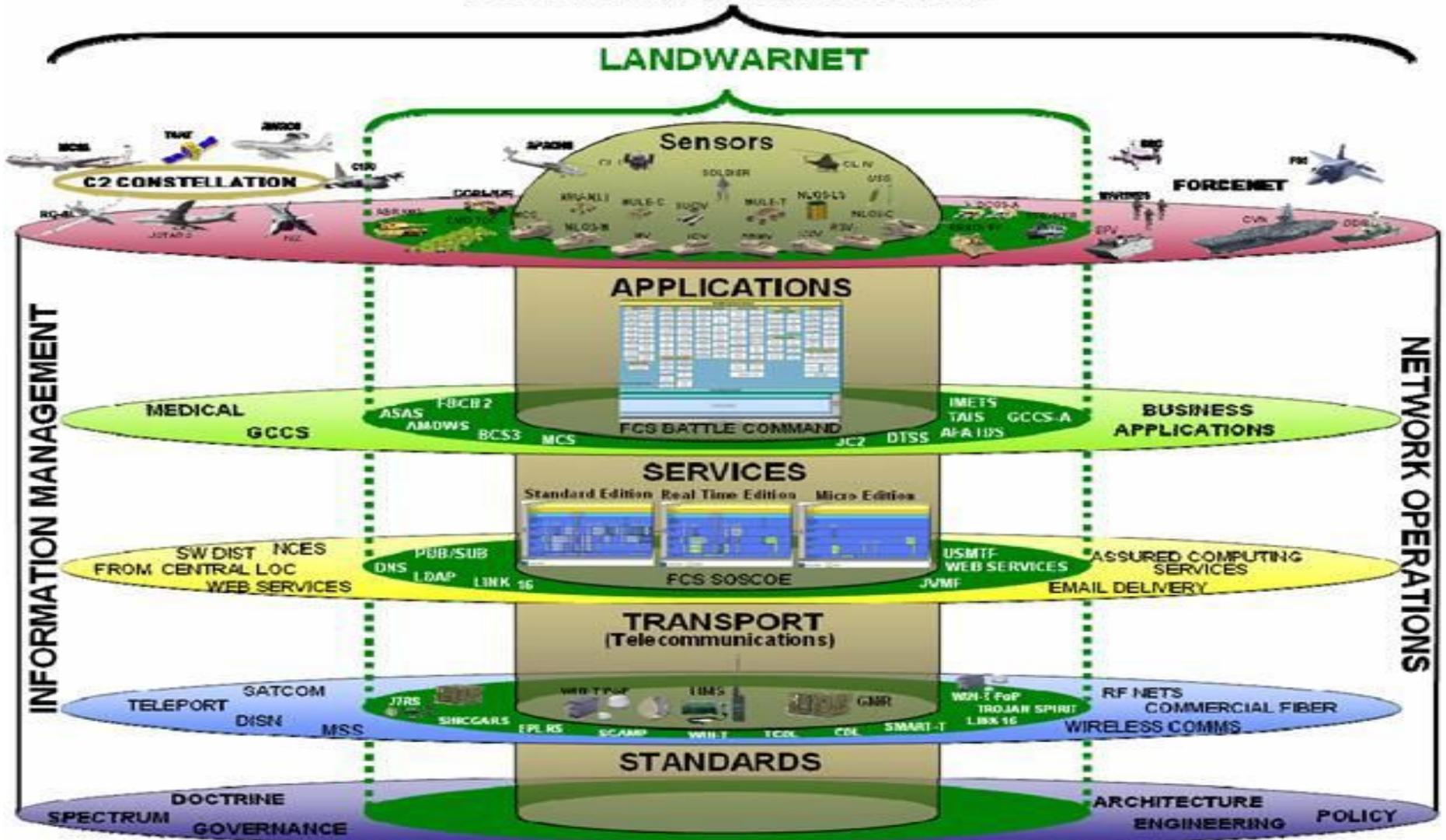


Сетецентрическое управление системами систем мобильных и стационарных объектов



Комплексная программа министерства обороны США *Future Combat Systems*

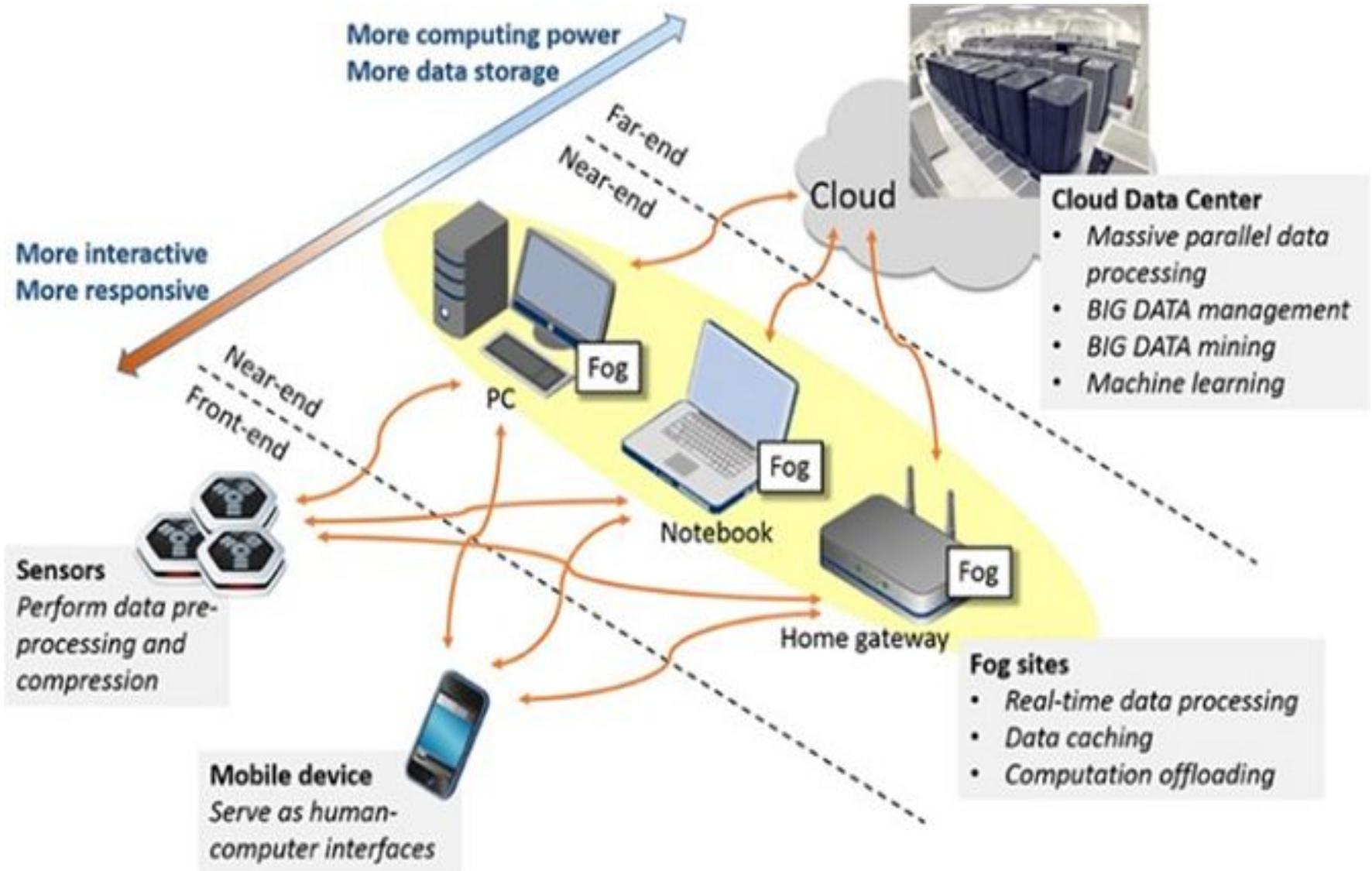
GLOBAL INFORMATION GRID



Полиморфизм вычислительно-коммуникационных систем

Физическая архитектура

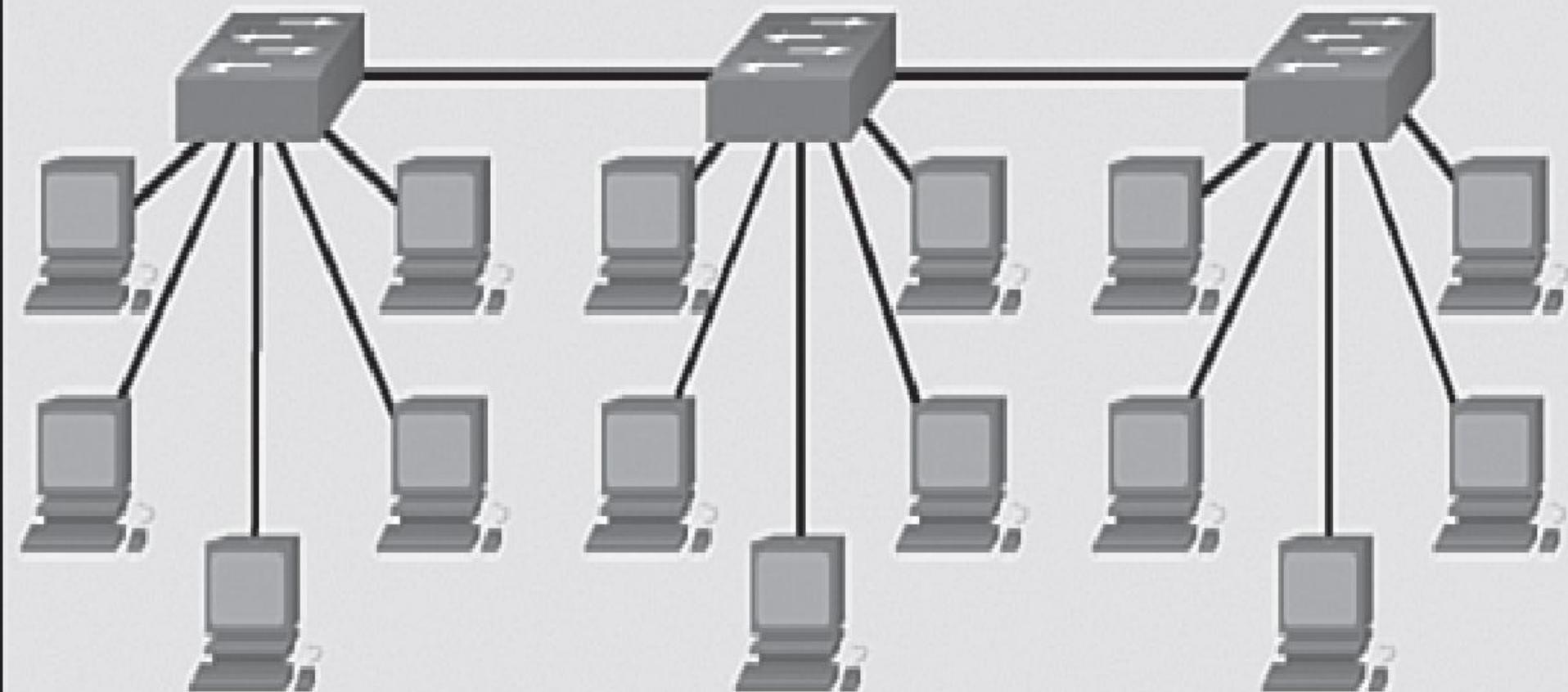
Cloud, Fog, Edge computing



Сложность логической организации информационных систем

Функциональная архитектура

Flat Switched Network

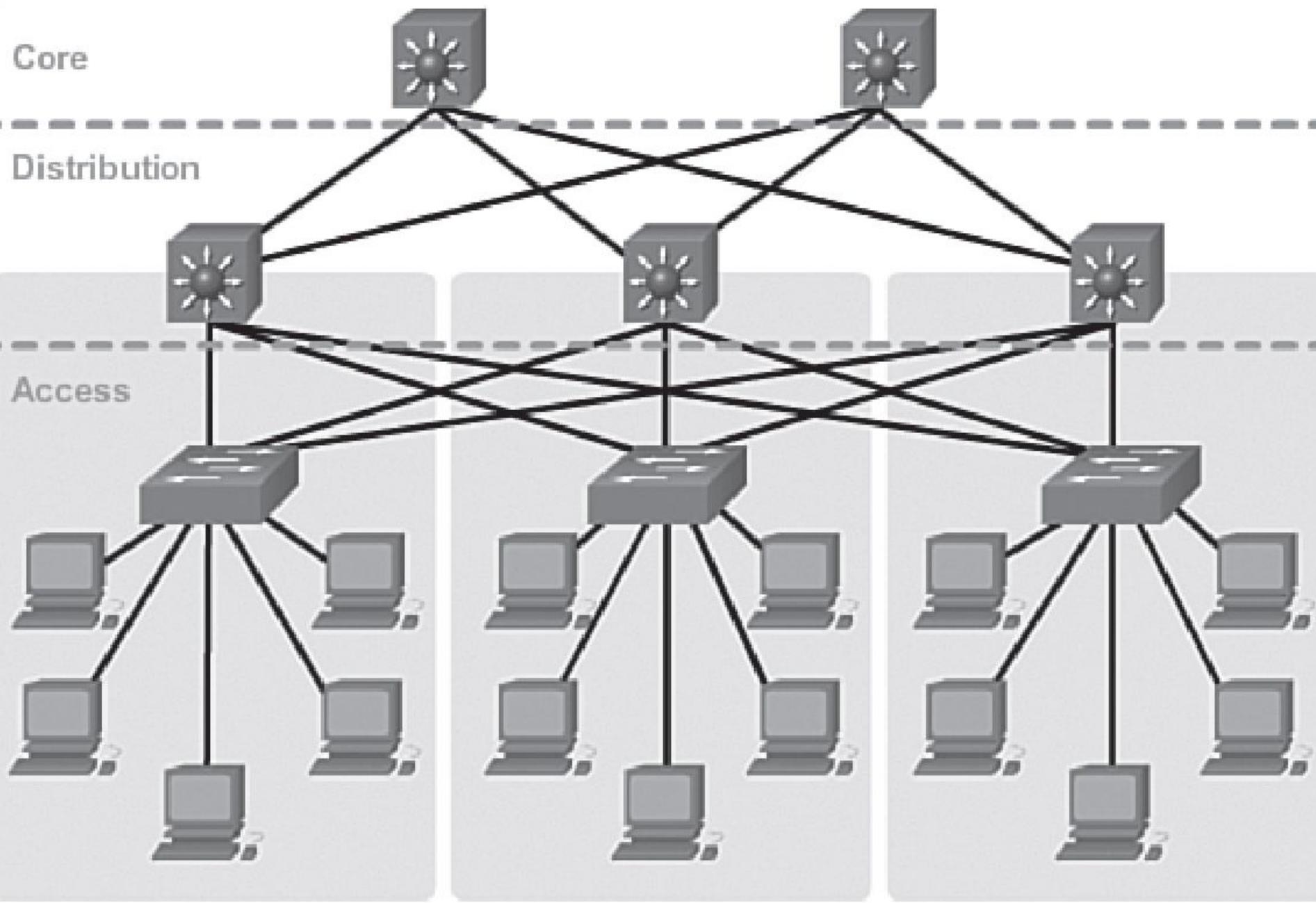


One large broadcast domain

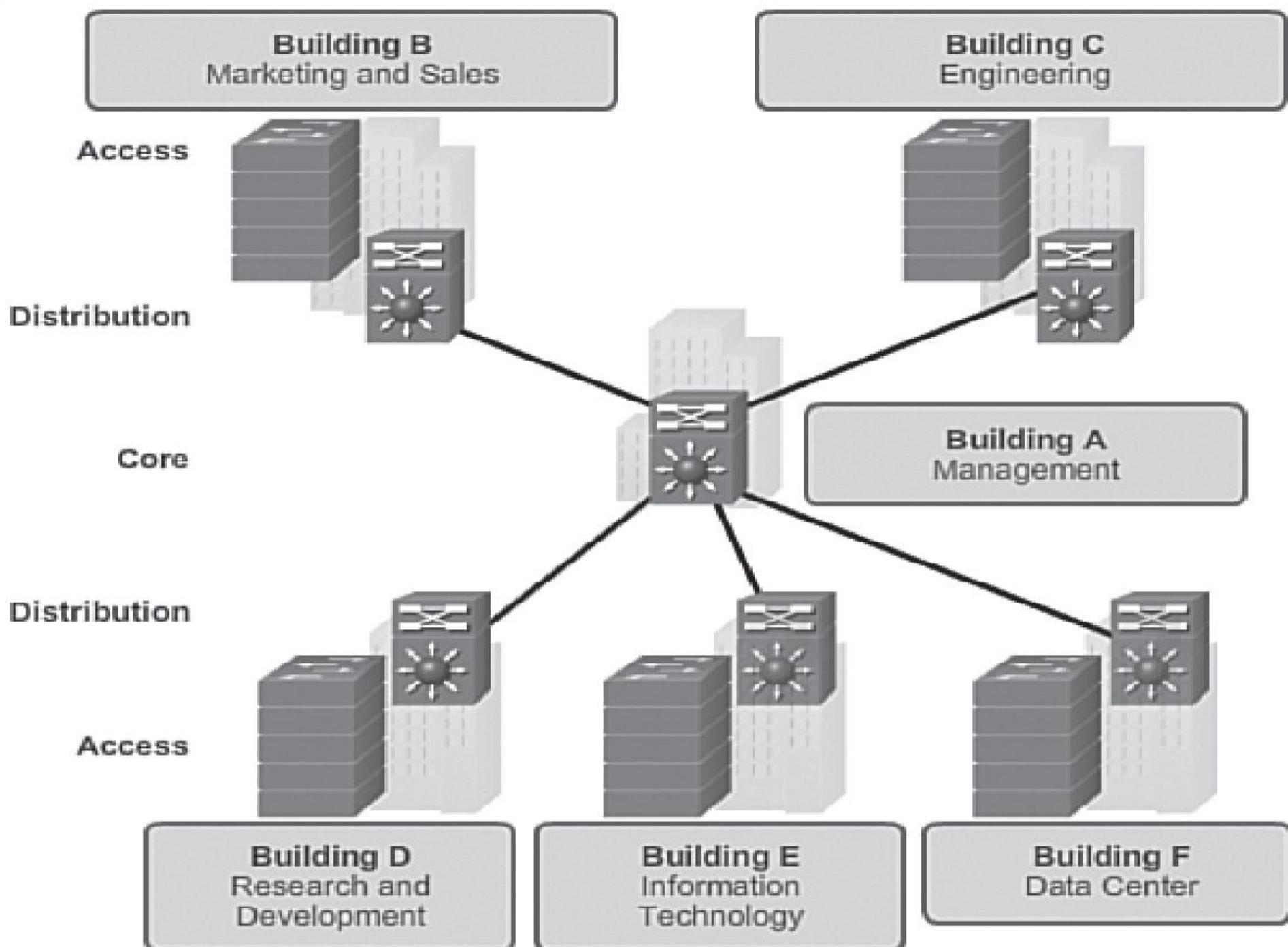
Core

Distribution

Access



Three separate broadcast domains



Логическая архитектура

СИСТЕМ

Источник: Integrated research agenda Cyber-Physical Systems

Internet of Things, Data, Services (Smart Cities)

Cyber-Physical Systems(e.g.intelligent networked road Junction)

Networked embedded Systems(e.g.autonomous aviation)

Embedded systems (e.g. airbag)

Функциональные компоненты сетевой операционной системы

Сетевая операционная система

Средства управления локальными ресурсами

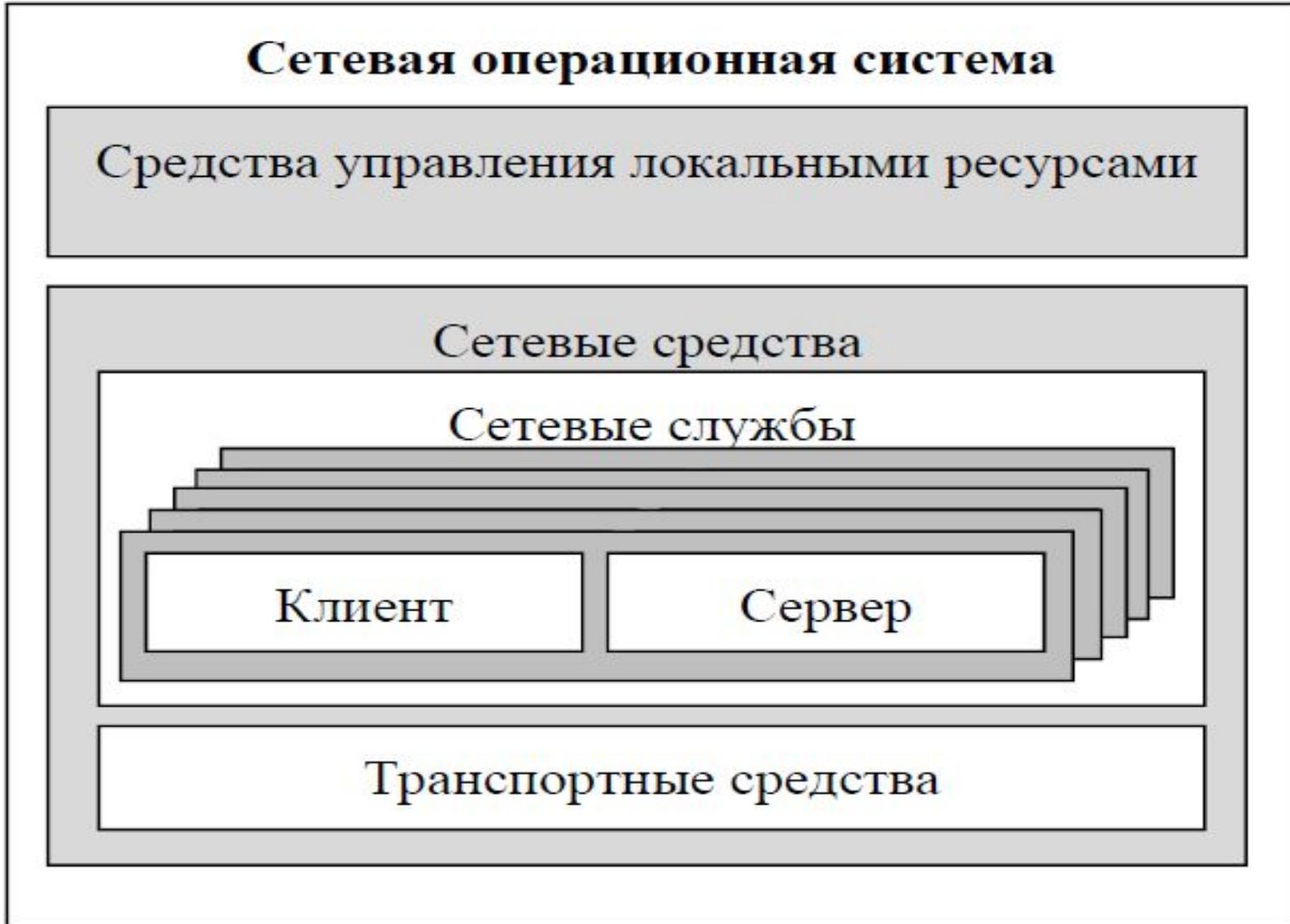
Сетевые средства

Сетевые службы

Клиент

Сервер

Транспортные средства



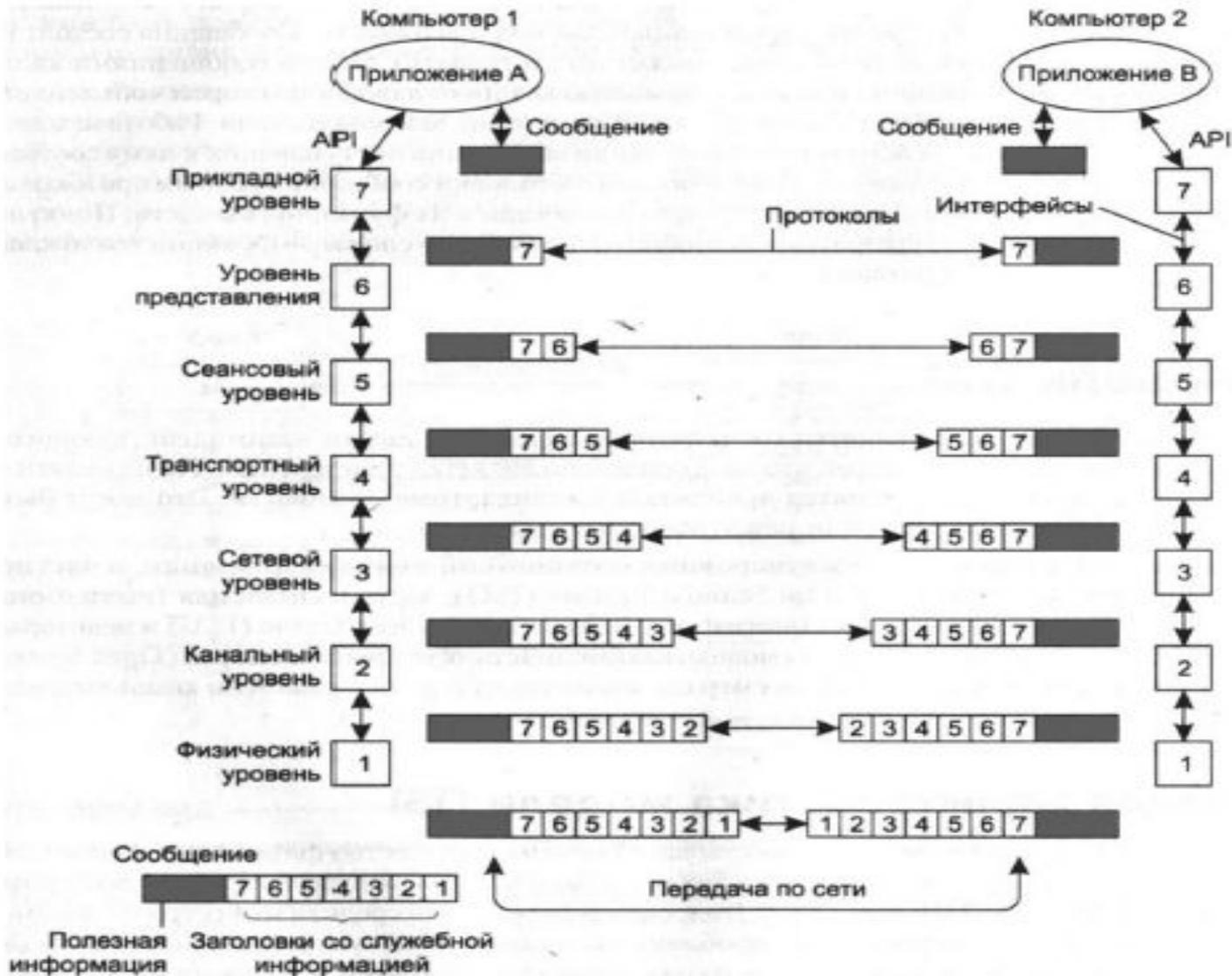
Интерфейсы и протоколы

Протокол — совокупность формализованных правил, определяющих последовательность и формат сообщений, которыми обмениваются сетевые компоненты, лежащие на одном уровне, но в разных узлах сети.

Интерфейс — совокупность формализованных правил и стандартизованных форматов сообщений, определяющих способ взаимодействия модулей соседних уровней в составе одного узла.

Стек коммуникационных протоколов — иерархически организованный набор протоколов, достаточный для организации взаимодействия узлов в сети.

Модель взаимодействия открытых систем ISO/OSI

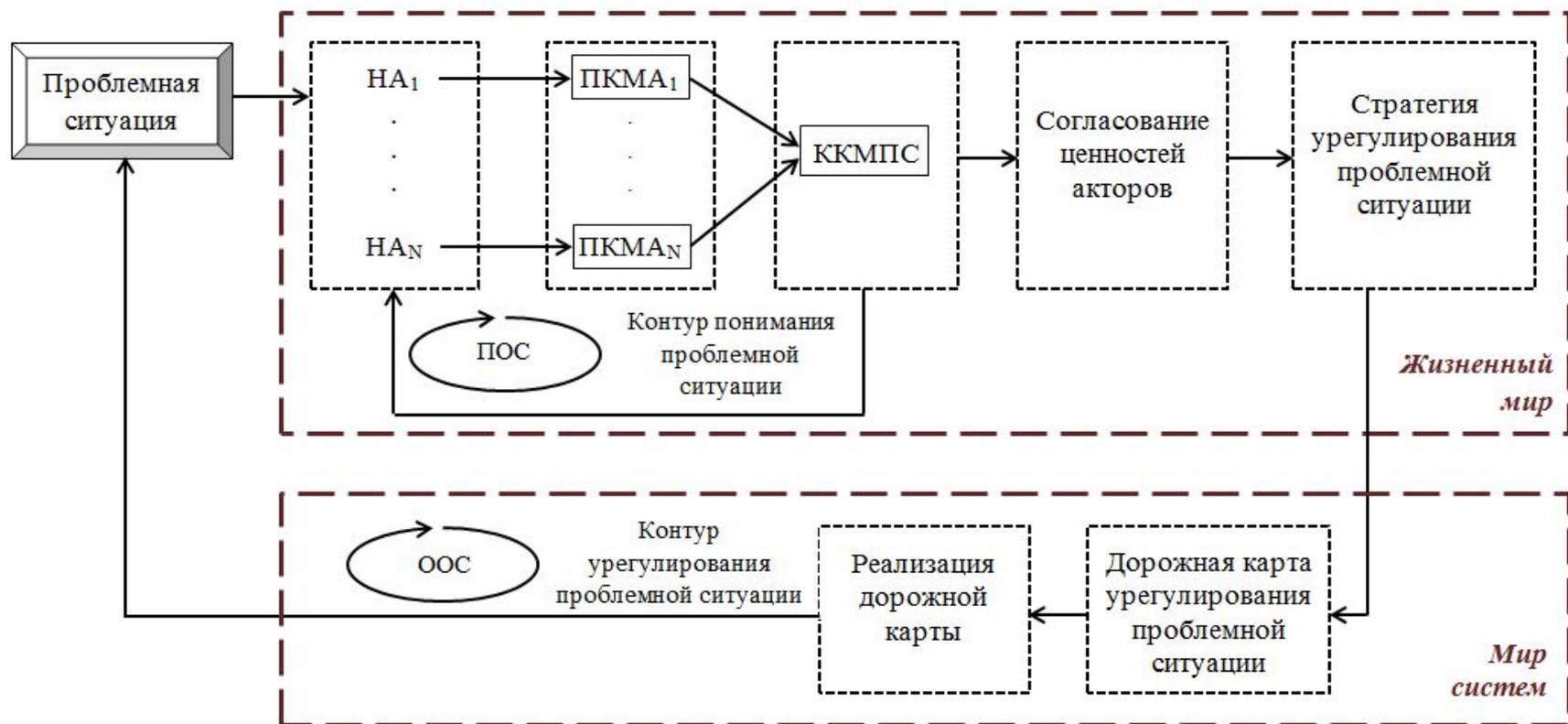


СТРУКТУРА ПРОЦЕССА ПРЕДПРОЕКТНОЙ СТАДИИ

Схема предпроектной стадии создания компонентов цифровой экосреды



Модель управления урегулированием проблемной ситуации



НА_i – неоднородный актер

ПКМА_i – персональная когнитивная модель i-го актора

ККМПС – консолидированная когнитивная модель проблемной ситуации

ПОС – положительная (развивающая) обратная связь

ООС – отрицательная (стабилизирующая) обратная связь

Что означает «Консолидированное
мнение?»»

компромисс

Под **компромиссом** чаще всего понимается ущемление своих интересов в угоду интересам другого...

Дискурс

... дискурс означает некоторый образ мышления, идеологию и то, как она проявляется словесно. Часто люди, которые интересуются дискурсом в этом смысле слова, ставят задачу выявить, что думает говорящий, на основании того, как он говорит...

Консенсус

Консенсус - это общее согласие по спорным вопросам, когда ни у одной из сторон нет существенных возражений.

Консенсус не обязательно предполагает полное единодушие. Обычно консенсус достигается на переговорах путем обсуждения и взаимных уступок.

Консенсус

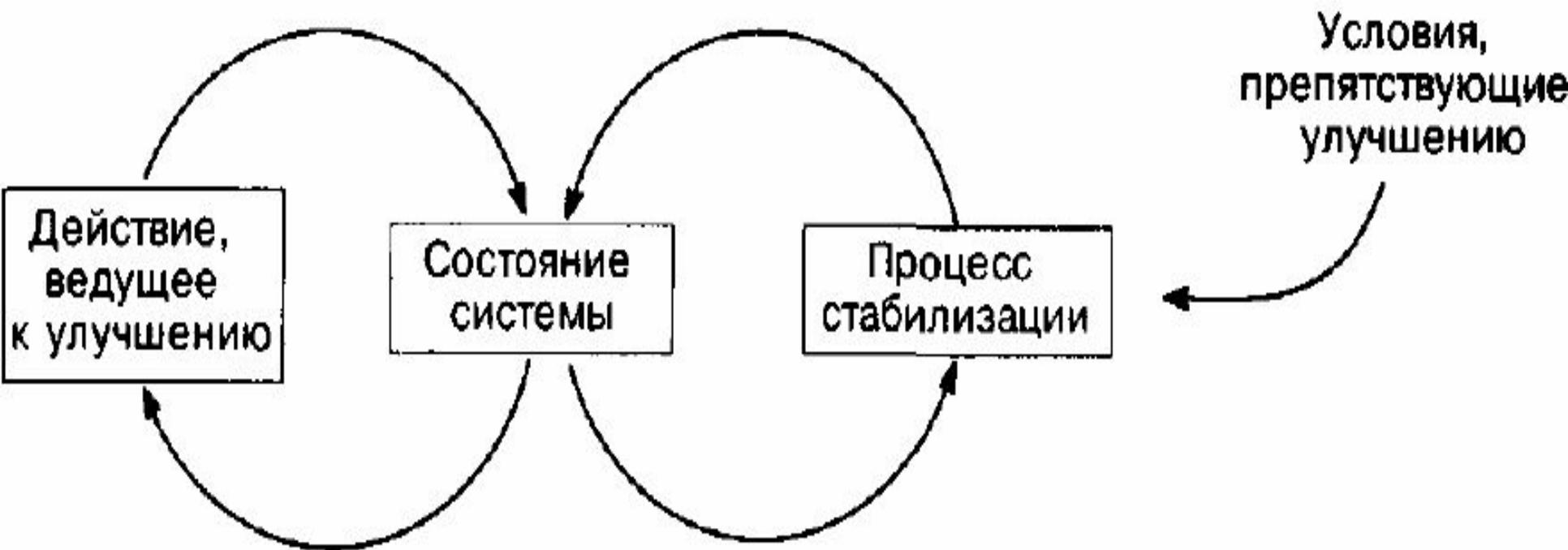


... Мы боимся изменений, но не можем жить без них...

...Люди сопротивляются не переменам, а тому, чтобы оказаться объектами перемен...

P. Senge

Архетип «Сопротивление внешнему влиянию»



Базовым понятием интерсубъективного управления является формирование неоднородными акторами консолидированного мнения относительно стратегии урегулирования проблемной ситуации

Ошибки и дефекты программных продуктов и программных проектов

В ЧЕМ СОДЕРЖАНИЕ ПРОБЛЕМЫ?

Неопределенность среды функционирования



Пространство функциональной безопасности компонент цифровой экосреды



Функциональная безопасность компонент цифровой экосреды как многосвязная система

(организационный аспект)

Инструменты
(ограниченная
функциональность)



**Цифровая
экосреда**

Информация
(ограниченная
рациональность)

Психология
разработчика

Конвенциональная концепция

...невозможна реальность, которая была бы полностью независима от ума, постигающего её...

...Истина – это то, что считается правильным большинством...

А. Пуанкаре

Понятие конвенции

Конвенция — (от лат. conventio —соглашение) — договор, соглашение, условие. Разнообразные К. играют значительную роль в науке и в повседневной жизни. Спор, дискуссия, коллективное обсуждение к.-л. Проблемы всегда опираются на соглашение относительно значений используемых слов, терминов, выражений.

Wikipedia

А. Пуанкаре
28.04.1854 – 17.07.1912



один из создателей топологии
и теории относительности,
автор понятия «конвенциональная рациональность»

Пример консолидированного мнения неоднородных акторов

У древних индийцев



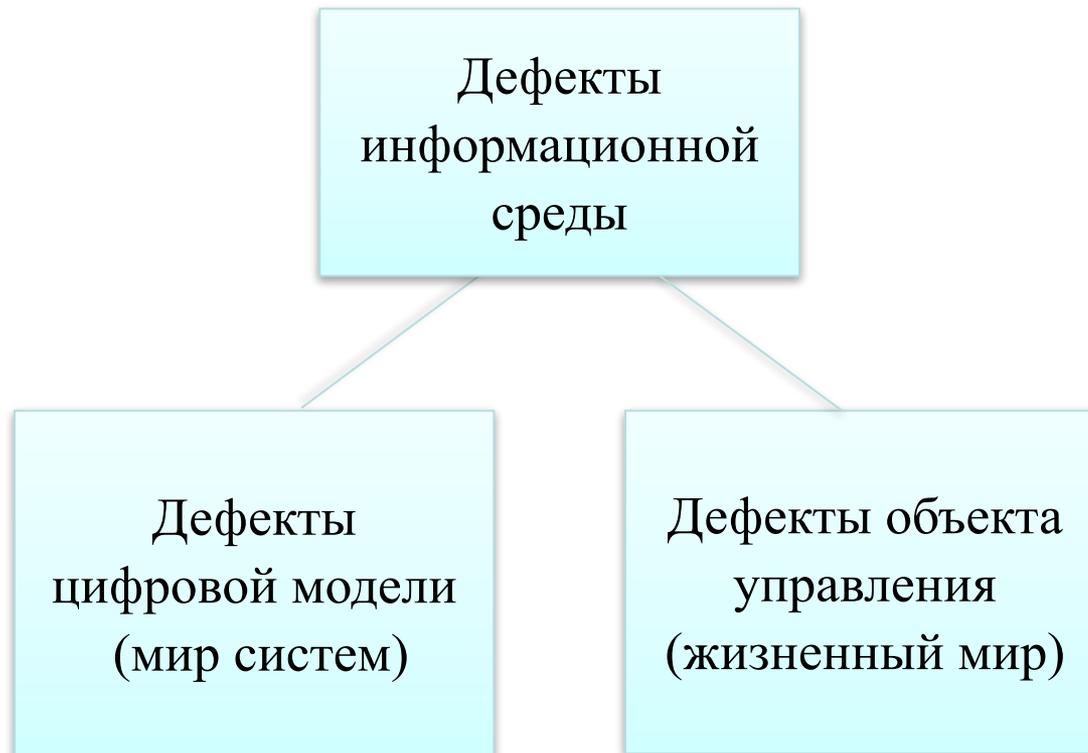
ПОСМОТРИТЕ НА ЭКРАНАХ НОВЫЙ ЗАГРАНИЧНЫЙ ФИЛЬМ



ГИБЕЛЬ ТИТАНИКА

Кинорежиссер Джеймс Кэмерон

Классификация дефектов



***Лимитирующий
фактор***

**Ограниченные области
применимости
формальных
моделей**

***Лимитирующий
фактор***

**Ограниченная
рациональность
субъектов
управления**

Термины и определения

Defect – всё, что может повлиять на неудовлетворенность покупателя свойствами продукта, системы, сервиса в течение их жизненного цикла

Error – дефект, не выявленный на стадии разработки, но установленный до того, как продукт попал к потребителю

Mistake – дефект, выявленный на стадии разработки

Классы ошибок, допускаемых на стадии подготовки требований



Классы ошибок, допускаемых на стадии проектирования программных продуктов



Классификация ошибок создания информационных систем



David Embry, «Understanding Human Behavior and Error»,
Human Reliability, 2005

Модель роста типов дефектов

Типы
дефектов

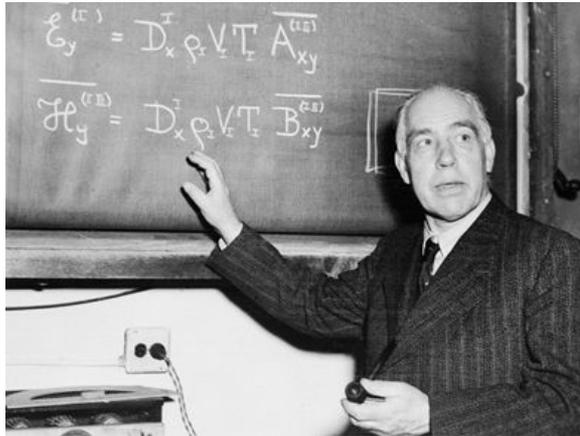
							Дефекты эксплуатации ПП	
						Дефекты развертывания ПП	Дефекты развертывания ПП	
					Дефекты проектирования ПП	Дефекты проектирования ПП	Дефекты проектирования ПП	
				Дефекты спецификации ПП	Дефекты спецификации ПП	Дефекты спецификации ПП	Дефекты спецификации ПП	
			Дефекты внешнего облика	Дефекты внешнего облика	Дефекты внешнего облика	Дефекты внешнего облика	Дефекты внешнего облика	
		Дефекты контракта	Дефекты контракта	Дефекты контракта	Дефекты контракта	Дефекты контракта	Дефекты контракта	
	Дефекты предложения	Дефекты предложения	Дефекты предложения	Дефекты предложения	Дефекты предложения	Дефекты предложения	Дефекты предложения	
Дефекты КМПС	Дефекты КМПС	Дефекты КМПС	Дефекты КМПС	Дефекты КМПС	Дефекты КМПС	Дефекты КМПС	Дефекты КМПС	
Формирование консолидированной когнитивной модели проблемной ситуации (КМПС)	Подготовка предложения на IT-сервисы	Подготовка контракта	Дефекты специфицирования внешнего облика	Дефекты специфицирования программного продукта	Дефекты проектирования программного продукта	Дефекты развертывания ПП	Дефекты эксплуатации ПП	Стадии жизненного цикла IT-сервисов

Типовые классы ошибок подготовки предложений по представлению IT-сервисов

1. Необоснованные предположения о реальных ценностях заказчика
2. Необоснованные сведения о рисках и возможностях поставщика сервисов
3. Необоснованные сведения о угрозах и возможностях внешней среды поставщика сервисов
4. Отсутствие описания классов задач, связанных с реализацией сервисов
5. Отсутствие описания изменений в бизнес-процессах и системе управления бизнес-процессами потребителя сервисов, обусловленных представлением сервисов
6. Отсутствие критериев и сведений о процедурах одобрения представляемых сервисов

Дефекты - ключевой фактор
развития цифровой экосреды

Принцип дополнительности в исследовании дефектов цифровой экосреды



Принцип дополнительности – один из важнейших методологических и эвристических принципов науки, основанный на введении взаимоисключающих классов понятий, каждый из которых применим в особых условиях, а их совокупность позволяет воспроизведение целостности изучаемых объектов

Сформулирован Н.Бором в 1927 г. для описания квантовомеханических явлений. Бор полагал, что принцип дополнительности применим не только в физике, но имеет

широкую методологическую значимость ■

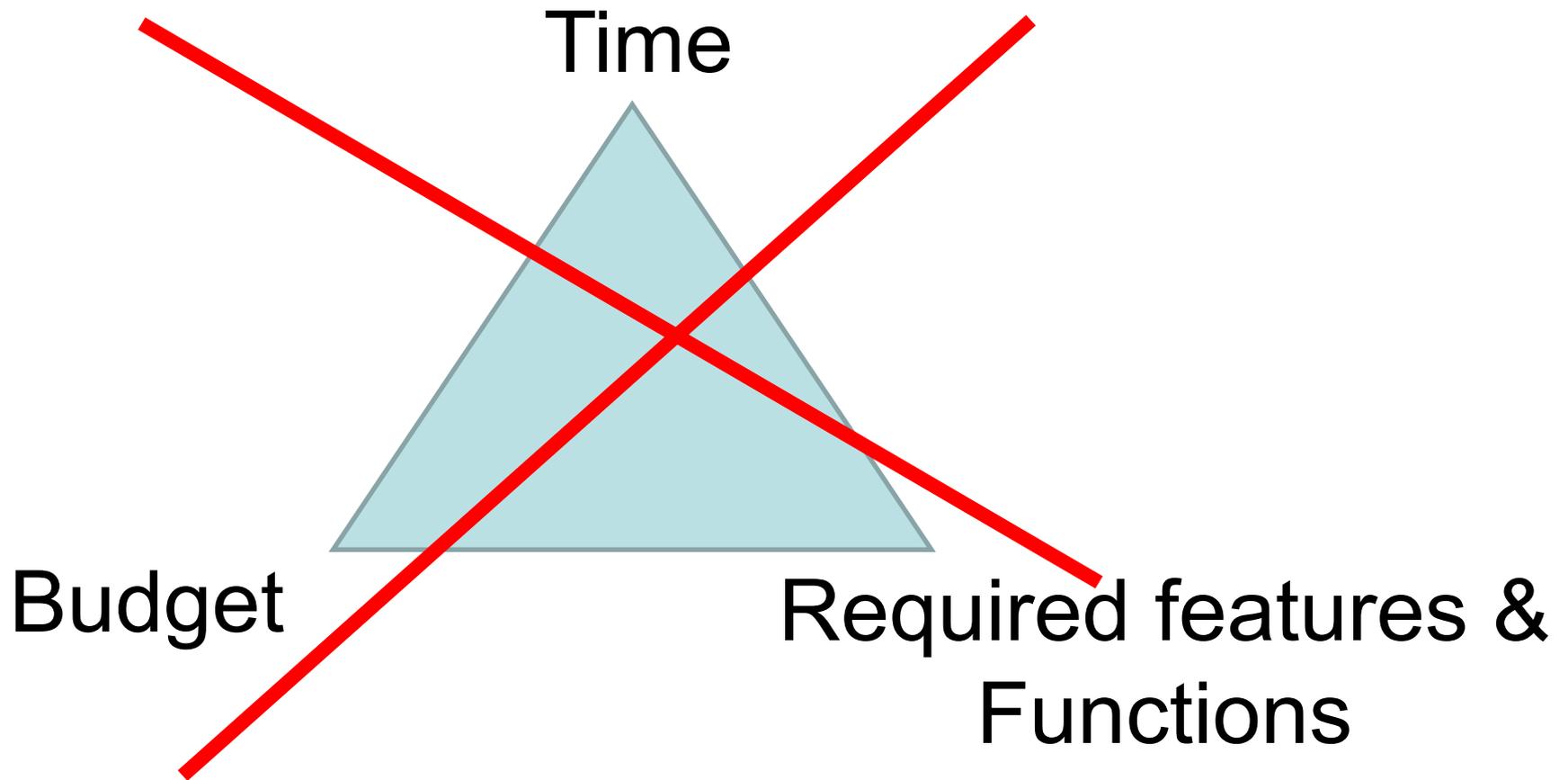
1. Дефекты – фактор, снижающий ценность информационных сервисов для потребителей
2. Дефекты – ключевой фактор развития гибко перестраиваемой обучаемой организации на основе получения новых знаний, поступающих в цифровую экосреду «умного предприятия», а также основа совершенствования системы предоставления сервисов



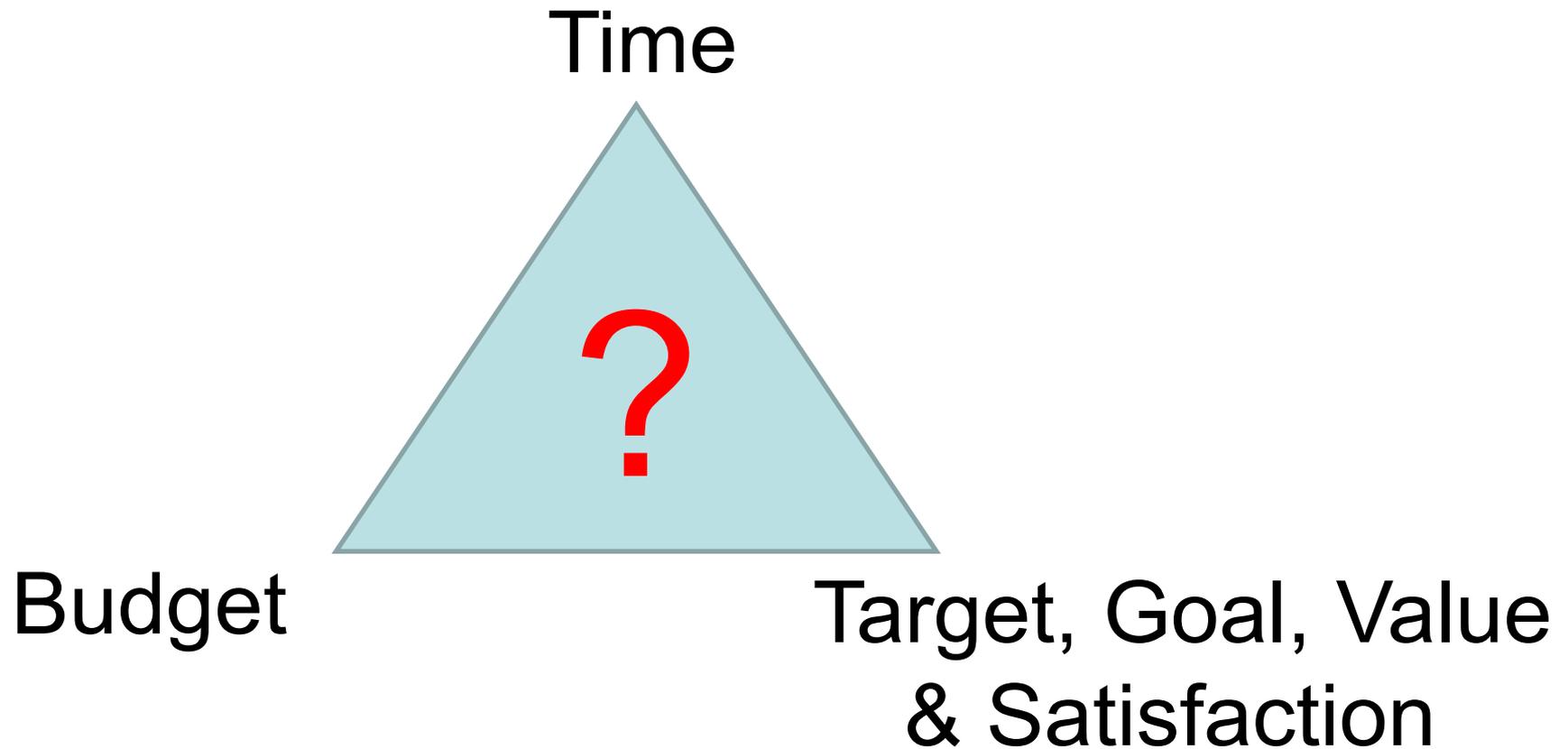
Необходимо смещение акцентов от вопросов эффективного управления дефектами (defect management) к вопросам создания условий, исключающих возможность возникновения дефектов (defect prevention)

Перспективным направлением реализации превентивного подхода является развитие теоретических положений, модельных основ и инструментальных средств психологии программирования в составе системы управления информационными сервисами, базовыми компонентами которой, помимо психологии программирования, являются положения системной инженерии, программной инженерии, надежности ПО, когнитологии

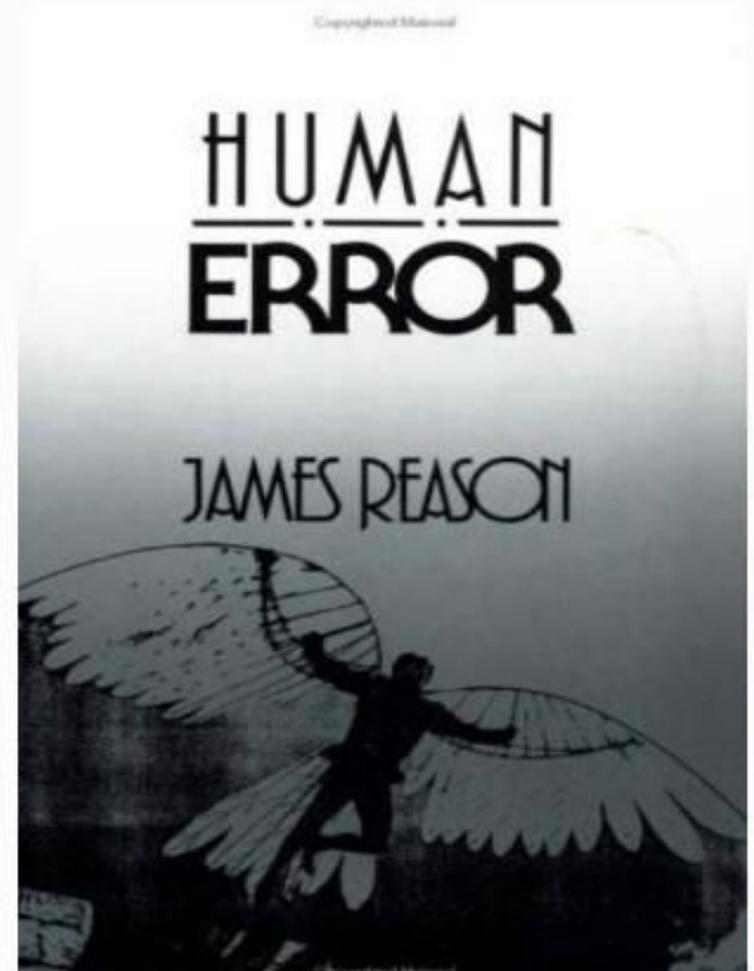
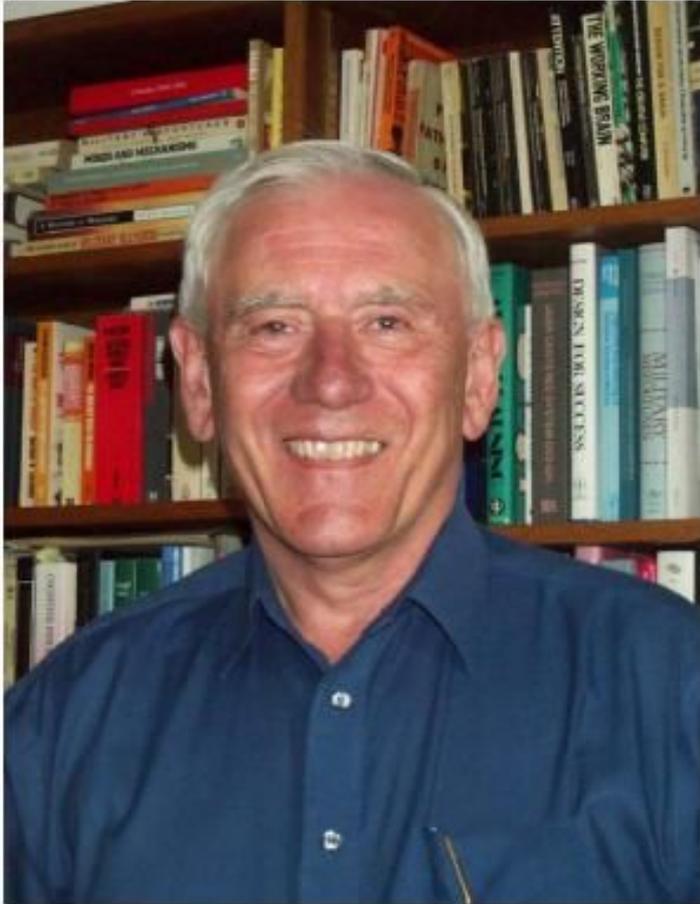
Project Triangle (PMI-1994)



Project Triangle (Standish Group-2015)

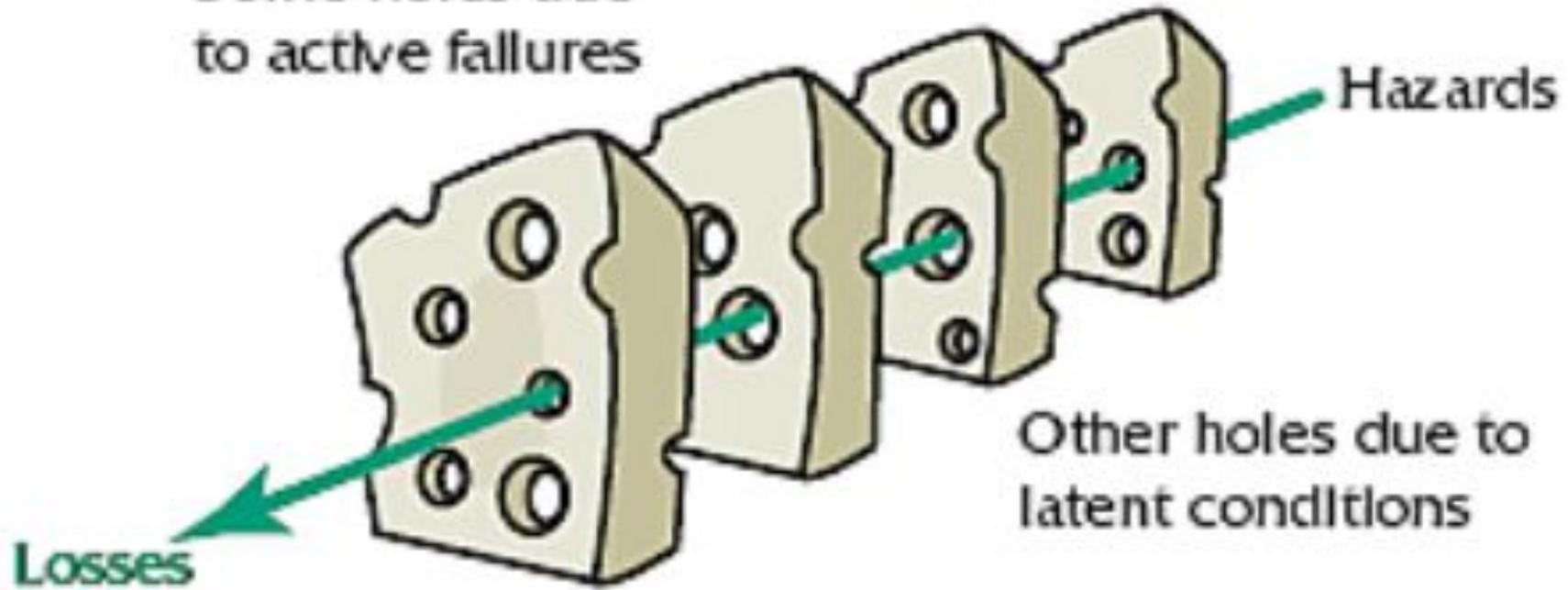


Human Error – James Reason



The Swiss Cheese Model of Accident Causation

Some holes due to active failures



Other holes due to latent conditions

Losses

Successive layers of defenses, barriers, & safeguards

Классификация защитных барьеров



Содержание метафоры «Swiss Cheese Model»

1. В любой системе защиты имеется множество «дыр» меняющих свое положение в пространстве и во времени.
2. Наличие «дыр» само по себе не обязательно приводит к негативным событиям.
3. Негативные события возникают тогда, когда источники опасности и «дыры» оказываются на одной линии.

Уровни защиты	Примеры «дыр»
Извлечение требований	Потребности потребителей отождествляются с потребностями пользователей
	Неверное оценивание состояния окружающей среды
	Плохое понимание ролей в организационной структуре
Специфицирование требований	Несоответствие содержания требований в спецификациях
	Необоснованные допущения
	Неполное использование/потеря требований, полученных от потребителей
Автоматизация разработки и испытаний	Семантические ошибки/ошибки в выборе модели ЖЦ ПП
	Ошибки планирования
	Ошибки исполнения

Роли Swissse Chesse Model

1. SCM как концепция.
2. SCM как коммуникационная основа.
3. SCM как база для анализа.
4. SCM как основа построения прогностических моделей.

SCM как концепция

Фокусом концепции является положение о том, что никакой инцидент не может быть обусловлен единственной причиной. Инцидент является результатом непредсказуемого сочетания нескольких факторов, истоки которых разнесены в пространстве и во времени. Выделяется системная составляющая инцидента, обусловленная нерациональными организационными, проектными и технологическими решениями, и являющаяся причиной возникновения латентных дефектов. Случайные, непредсказуемые внешние воздействия приводят к активизации латентных дефектов («дыр» в слоях системы).

SCM как коммуникационная основа

Фокусом этой роли является то, что SCM позволяет на систематической основе обеспечить коммуникации специалистов в различных предметных областях при расследовании причин инцидентов.

SCM как база для анализа

Инцидент объясняется возникающей во времени каузальной цепочкой различного рода недостатков (англ. - deficiencies). Наличие недостатков не означает, что инцидент обязательно произойдет. При формировании каузальных цепочек исходят из того, что одно негативное событие может породить другое негативное событие, что в итоге может привести к инциденту. Однако в рамках этой роли не представляется возможным объяснить инциденты, имеющие место в случае, когда нет видимых причин инцидентов, т.е. в случае, когда все контролируемые параметры компонентов SCM лежат в допустимых границах.

SCM как основа построения прогностических моделей

SCM ориентирует на выделение ограниченного набора «показателей здоровья» сетецентрических систем, исследование изменения которых во времени создает основу для решения задач краткосрочного прогнозирования. Результатом решения задачи прогнозирования является оценка возможности возникновения разных инцидентов (областью применимости моделей является ограниченное число (10-12) классов отказов). При этом точность оценивания времени и места возникновения инцидентов достаточно низкая. **(СИСТЕМНЫЕ АРХЕТИПЫ)**

Прогностические модели, реализованные в рамках SCM, позволяют получить достаточно грубые (но устойчивые) результаты, обладающие низкой разрешающей способностью. Эти модели ориентированы на выделение в системе функций, повреждение которых может послужить причиной различного рода инцидентов.

Методические ограничения SCM

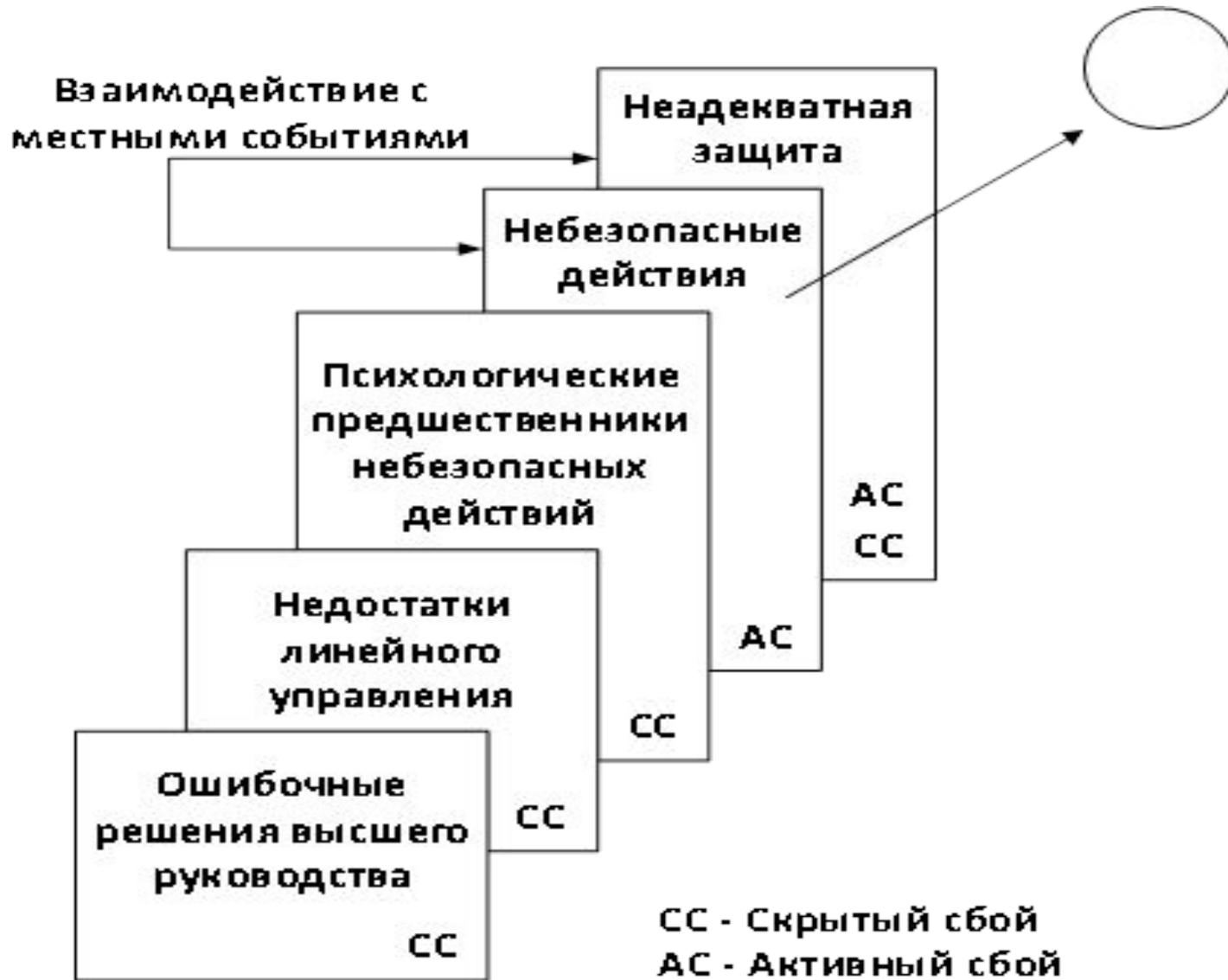
1. Предполагается линейная схема преобразования источника опасности в инцидент. Не учитывается то обстоятельство, что латентные дефекты в вышестоящих слоях иерархической системы могут быть обусловлены ошибочным реагированием на отказы, ранее имевшие место в нижележащих слоях.
2. Предполагается, что казуальные цепочки возникают хаотично. Отсутствуют подходы к ранжированию возможности реализации инцидентов, обусловленных различными казуальными цепочками.
3. Не предусмотрена возможность одновременного возникновения инцидентов разной природы.
4. Постулируется линейная упорядоченность событий во времени. Однако из того, что событие A предшествовало событию B , не следует, что A является причиной B . В рамках SCM не представляется возможным указать события – коренные причины инцидентов, разнесенные в пространстве и во времени.
5. Не определен подход для оценивания вклада субъективной, организационной и технологической составляющих в возникновение латентных дефектов.

Ограничения на практическое использование Swiss cheese model

1. SCM на настоящем уровне развития не позволяет предвидеть (предсказывать) инциденты.
2. Модель не проходила верификацию.
3. Отсутствует стандарт, регламентирующий ее применение.
4. Неопределенность выявления потенциальных казуальных цепочек.
5. Неопределенность содержания «дыр» (holes) в слоях сыра.
6. Не определены причины изменчивости местоположения и размеров «дыр» во времени.

Примеры подходов реализующих Swiss Cheese Model

Модель Mark-I



Концептуальная основа модели Mark-I

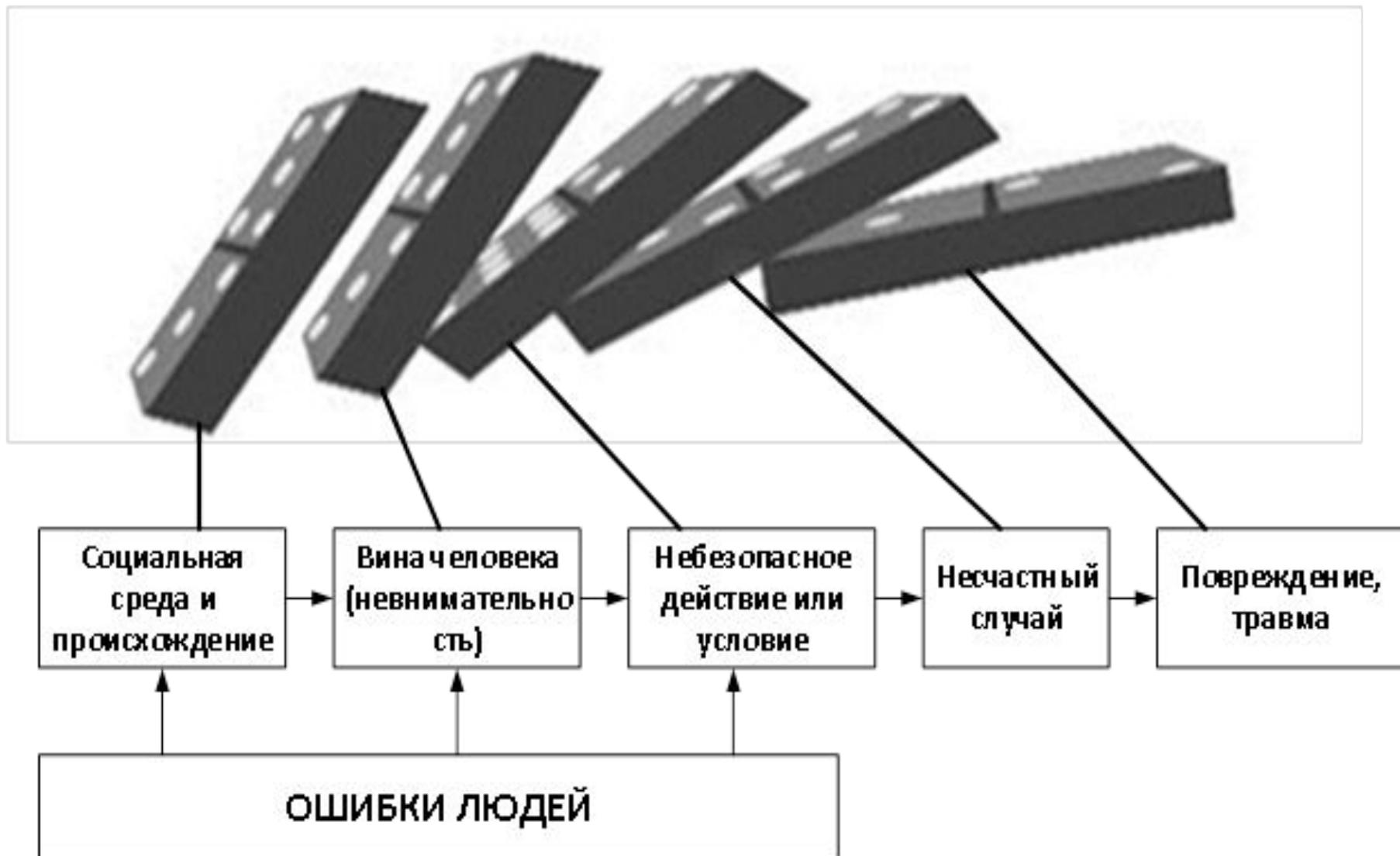
Концептуальная основа Mark-I заключается в том, что **инцидент есть результат реализации последовательности событий** (принцип Heinrich Domino). При исследовании инцидентов на первом месте должен находиться не вопрос «кто виноват?», а вопрос «почему не сработала система защиты?». Выделенные проекции послужили базой формирования многослойной модели, «дыры» (holes) в слоях которой соответствовали событиям/условиям, результатом размещения которых на одной линии являлся инцидент.

Содержание модели

J.Reason и John Wreathall в рамках предложенной модели выделили пять базовых проекций объекта управления, соответствующих жизненно важным органам обеспечения функционирования субъектоцентрических сложных систем:

1. Принятие субъективных решений на высшем уровне управления (top level decision makes).
2. Принятие субъективных решений на тактическом (линейном) уровне управления (line management).
3. Выявление предпосылок к возникновению инцидентов (precondition).
4. Производственная деятельность (productivity activity).
5. Парирование инцидентов (defenses).

Модель «Эффект Домино» является составной частью модели Mark-I



Ограничения модели Mark-I

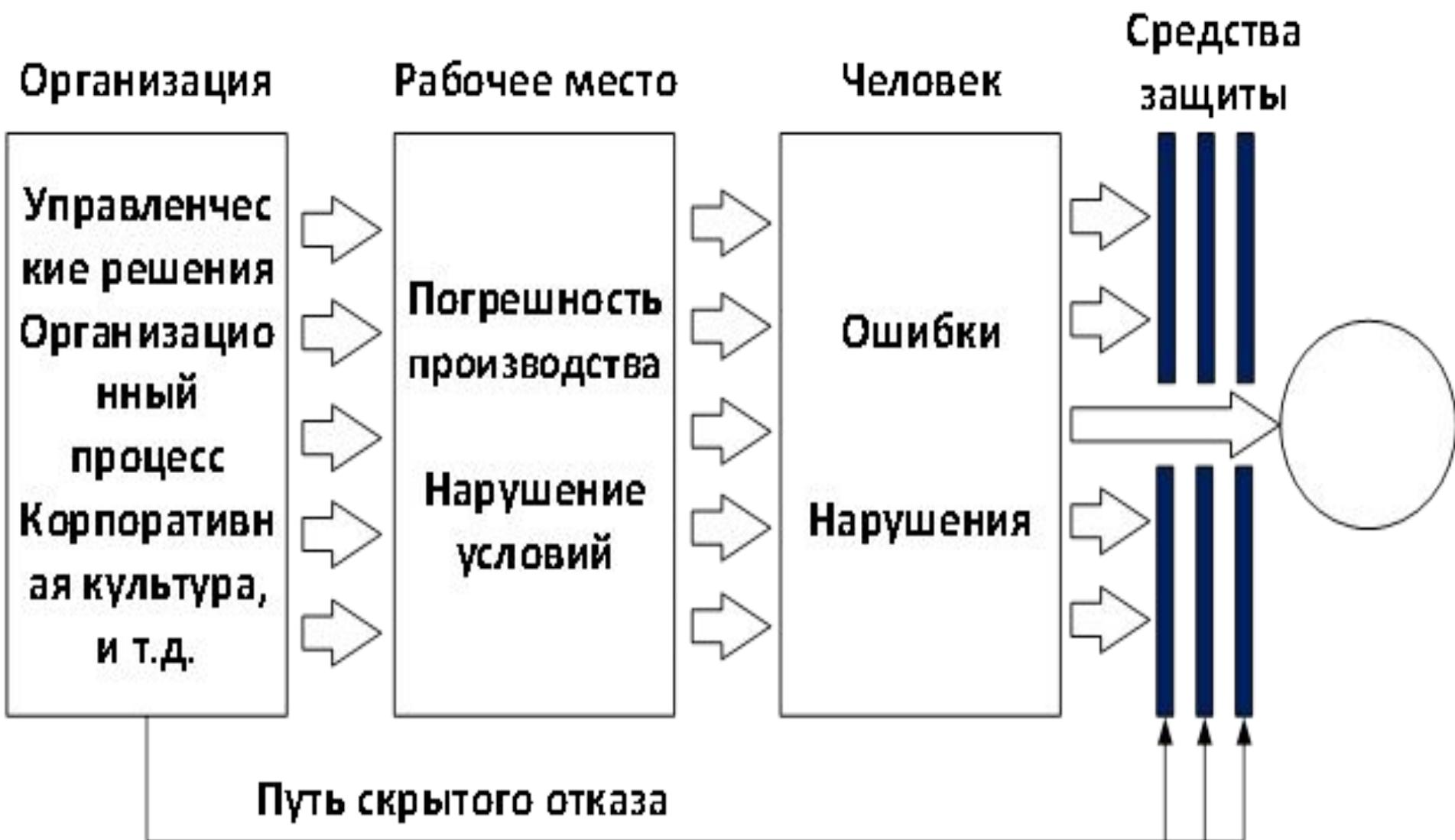
1. Постулируется линейная упорядоченность событий во времени.
2. Не предусмотрена возможность выявления случайных совпадений.
3. Не учитывается возможность совершения ошибок на верхнем уровне управления как реакции на ошибочные действия операторов (не выделены цепи обратной связи).
4. Ориентирована на построение статических моделей (казуальных цепочек). Отсутствует возможность построения динамических моделей.
5. Человеческая составляющая инцидентов не выделяется на фоне латентных факторов иной природы.

Трёхслойная модель обеспечения безопасности

Метафоре Swiss Cheese Model ставится в соответствие трёхслойная модель. Слои модели отражают следующие жизненно важные (с точки зрения возникновения инцидентов) производственные плоскости:

1. Организация;
2. Рабочее место;
3. Человек.

Модель Mark-II



Концептуальная основа модели Mark-II

Инциденты объясняются не только неблагоприятным стечением обстоятельств, но и отказами в защитных барьерах. Главным вопросом исследования инцидентов становится не «кто виноват?», а «почему не сработала система защиты?». Смещение акцентов от источников опасности к защитным барьерам выражает следующую точку зрения: **в силу того, что невозможно предсказать те условия, в которых может оказаться система, более обоснованным подходом является усиление защитных свойств системы нежели выявление внешних источников негативного воздействия.**

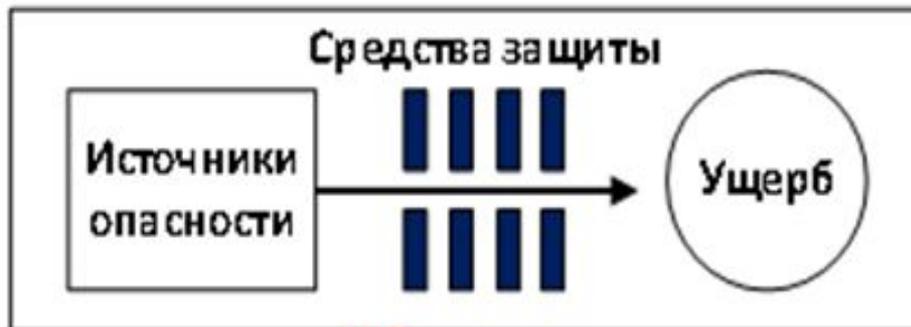
Реализация закона обратных связей

Модель Mark-III

ЧТО?

КАК?

ПОЧЕМУ?



Пути скрытых условий

Небезопасные действия

Местные факторы на рабочем месте

Организационные факторы



Причины

Исследования

Содержание модели

Фокусом исследований становится не инцидент, а связанные с ним негативные последствия (потери ценностей для субъектов). **Инцидент возникает в результате непредсказуемого возникновения каузальной цепочки событий, обусловленных сочетанием внешних событий и латентных дефектов.** Одной из возможных причин возникновения каузальных цепочек даже в «хороших», с точки зрения функциональной безопасности, системах является вариативность (в статистическом смысле) внешних нагрузок и характеристик состояния системы. **Инцидент может явиться результатом сочетания событий, вероятность возникновения каждого из которых крайне мала.**

Методологическая особенность Mark-III

В рамках Mark-III подчеркивается, что исследование инцидентов нельзя сводить лишь к выявлению каузальных цепочек. Выявление «коренных» причин не позволяет сформировать достаточную основу для предупреждения дефектов. В методологическом плане особенностью Mark-III является положение о том, что в состав модели инцидента помимо прочих обязательно должны входить три концепта:

1. источник опасности;
2. система защиты;
3. ущерб.

КОНЕЦ ЛЕКЦИЙ