

# Системная инженерия

Понятие жизненного цикла

# Основные вопросы

- Понятие жизненного цикла
- Типовость (уровни воплощения) и разнообразие жизненных циклов, связь жизненных циклов разных уровней структуры вещества в составе системы
- Основные формализмы представления жизненного цикла
- Виды жизненных циклов: последовательный, инкрементальный, итерационный.
- Пошаговое выделение ресурсов (ICM)

Определения

Примеры из разных областей

NASA

Kosiakoff

**ЖЦ СИСТЕМЫ**

# Жизненный цикл системы – 1/4

- Жизненный цикл – это морская свинка (не свинья и не морская):
  - Не «жизненный»: не относится к биологии
  - Не «цикл»: не замкнут (хотя иногда и кажется, что замкнут – sic!)
- Жизненный цикл системы (system life cycle) – деятельность всех обеспечивающих систем, ведущих целевую систему от её замысла («рождения» определения системы) до вывода из эксплуатации («смерти» воплощения системы)
  - Обычно эту деятельность разбивают на этапы (стадии – stages, у военных США: фазы – phases)
  - Этапы могут быть не только последовательными, но и перекрываться во времени друг с другом

## Жизненный цикл системы – 2/4

- Под термином «жизненный цикл системы» обычно понимают эволюцию новой системы в виде нескольких ступеней, включающих такие важные стадии, как: проработка концепции, разработка, производство, эксплуатация и окончательное выведение из нее

*Kossiakoff, Sweet, Seymour, Biemer 2011*

## Жизненный цикл системы – 3/4

- Этапы – отрезки времени. Ими нельзя «управлять» → Управление жизненным циклом – управление деятельностью (управление обеспечивающей системой), обеспечивающей переход от одной стадии жизненного цикла к другой

# Жизненный цикл системы – 4/4

- Жизненный цикл проекта (project life cycle) — часть жизненного цикла системы, укладываемая в рамки проекта. ЖЦ проекта может:
  - совпадать во времени со стадией ЖЦ системы
  - не совпадать
  - NB! В рамки ЖЦ проекта (деятельности проекта) может не попадать вся деятельность стадии ЖЦ системы
  - Проект разбивается на этапы (чтобы этап проекта ≠ стадии ЖЦ системы)

# ЖЦ разных элементов

ПО	Концепция	Разработка	Поддержка	Списание		
Оборудование	Идея	Проектирование	Изготовление	Эксплуатация и поддержка	Списание	
Персонал	Определение требуемых компетенций	Приобретение	Обучение	Использование и рост	Отставка	
Здание	Визуализация	Проектирование сооружения и площадки	Согласование	Строительство	Эксплуатация и поддержка	Разборка
Природный ресурс	Приобретение	Разработка	Эксплуатация	Рекультивация		
Процесс	Определение выхода	Графическое представление	Описание	Пилотное внедрение	Использование и совершенствование	Ликвидация
Система	Идея	Разработка	Изготовление	Использование	Поддержка	Списание



# Типовые ЖЦ проектов

По стандарту ISO/IEC 15288:2002:

- 1) стадия замысла.
- 2) стадия разработки.
- 3) стадия производства.
- 4) стадия применения.
- 5) стадия поддержки применения.
- 6) стадия прекращения применения и списания.

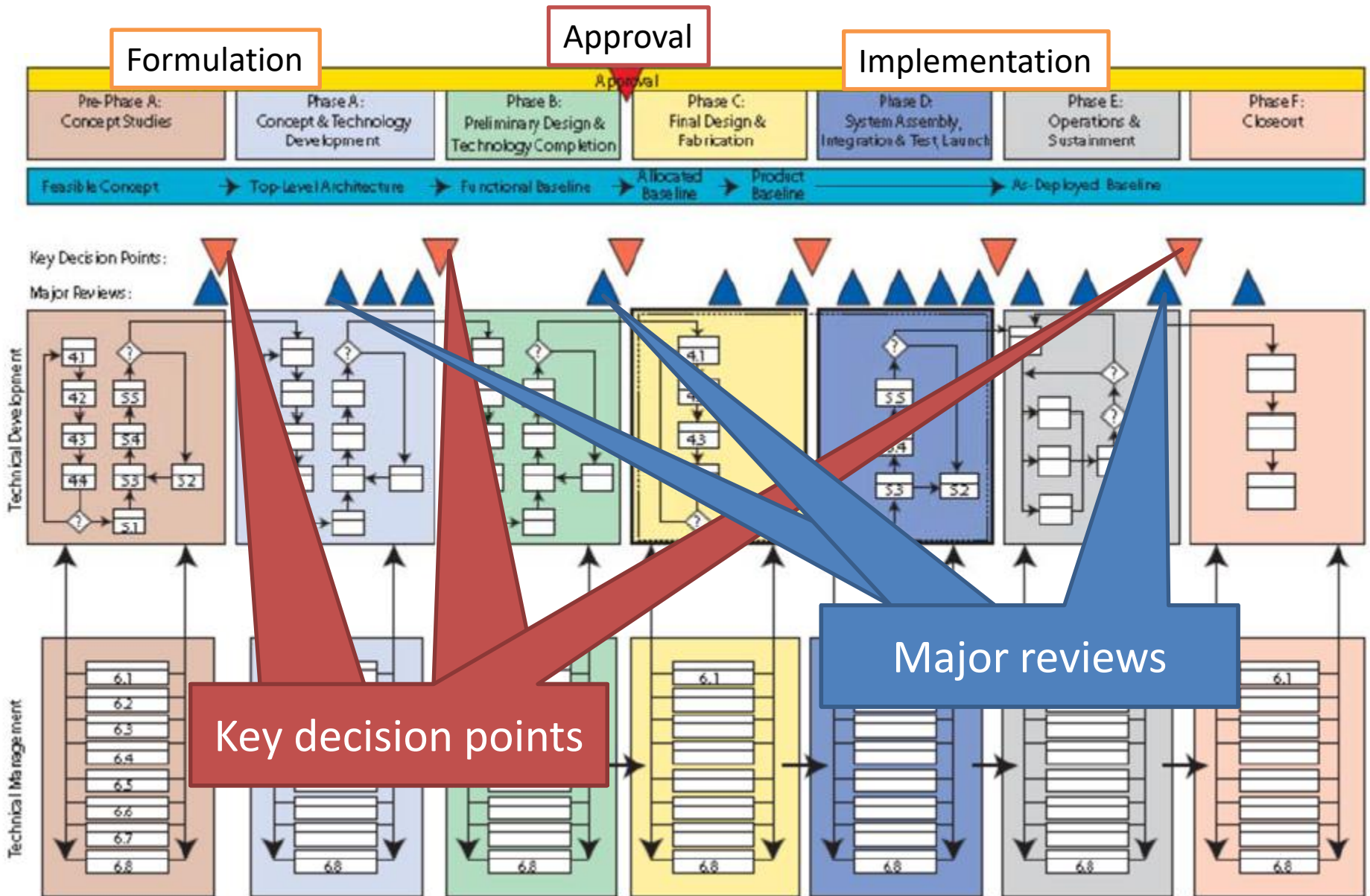
По версии Министерства обороны США:

- 1) анализ;
- 2) разработка технологии;
- 3) инженерная и производственная разработка;
- 4) производство и развёртывание;
- 5) функционирование и поддержка.

По версии Национального общества профессиональных инженеров (NSPE):

- 1) концепция;
- 2) техническая реализация;
- 3) разработка;
- 4) коммерческая валидация и подготовка производства;
- 5) полномасштабное производство;
- 6) поддержка конечного продукта.

# ЖЦ НАСА – 1/9



# ЖЦ НАСА – 2/9

- Pre-Formulation
  - Pre-Phase A – Concept Studies (Feasible concept – реализуемая концепция, эскизный проект(?))
- Formulation
  - Phase A – Concept and Technology Development (High-Level Architecture – высокоуровневая архитектура, эскизный проект(?))
  - Phase B – Preliminary Design and Technology Completion (Functional baseline – функциональная архитектура, технический проект (?))

# ЖЦ НАСА – 3/9

- Implementation
  - Phase C – Final Design and Fabrication (Allocated baseline → Product baseline)
  - Phase D – System Assembly, Integration and Test, Launch
  - Phase E – Operations and Sustainment (As deployed baseline)
  - Phase F – Closeout

# ЖЦ НАСА – 4/9

Фаза	Назначение	Результат
Pre-Phase A Concept Studies	Создание широкого спектра идей и альтернатив по выполнению задачи для выбора программы / проекта. Определение осуществимости желаемого (желаемой системы), разработка концепции [космической] миссии, черновая версия требований системного уровня, оценка эффективности, стоимости и «вкладывания в срок»; определение потенциальных технологических потребностей и область применения (scope)	Описание концепции реализуемой системы в виде имитационной модели (simulation), анализа, результатов исследований, моделей (models – картинок ^_^), макетов (mockups)

# ЖЦ НАСА – 5/9

Фаза	Назначение	Результат
Phase A Concept and Technology Development	<p>Определение осуществимости и желательности предлагаемой новой системы. Определение первоначальной базовой совместимости со стратегическими планами НАСА.</p> <p>Разработка концепции конечной миссии, требований системного уровня, определение необходимости разработки системных технологий, создание планов технического управления программами / проектами</p>	<p>Описание концепции системы в виде имитационных моделей (simulations), анализа, инженерных моделей и макетов, предоставление выбора вариантов [технологических решений]</p>

# ЖЦ НАСА – 6/9

Фаза	Назначение	Результат
Phase B Preliminary Design and Technology Completion	Создание проекта достаточно подробного для создания baseline достаточного для удовлетворения потребностей [космической] миссии. Разработка требований и предварительного архитектуры конечной и обеспечивающей системы (system structure end product and enabling product)	Описания в виде макетов, результатов выбора альтернатив, спецификаций и интерфейсных документов, прототипов

# ЖЦ НАСА – 7/9

Фаза	Назначение	Результат
Phase C Final Design and Fabrication	Завершение подробной архитектуры системы (а также систем более высокого уровня (associated subsystems), включая использующие системы), изготовление hardware и разработка СПО. Создание конечных версий архитектур для всех конечных продуктов	Конечные версии архитектур для всех конечных продуктов, изготовление компонентов конечных продуктов, разработка СПО



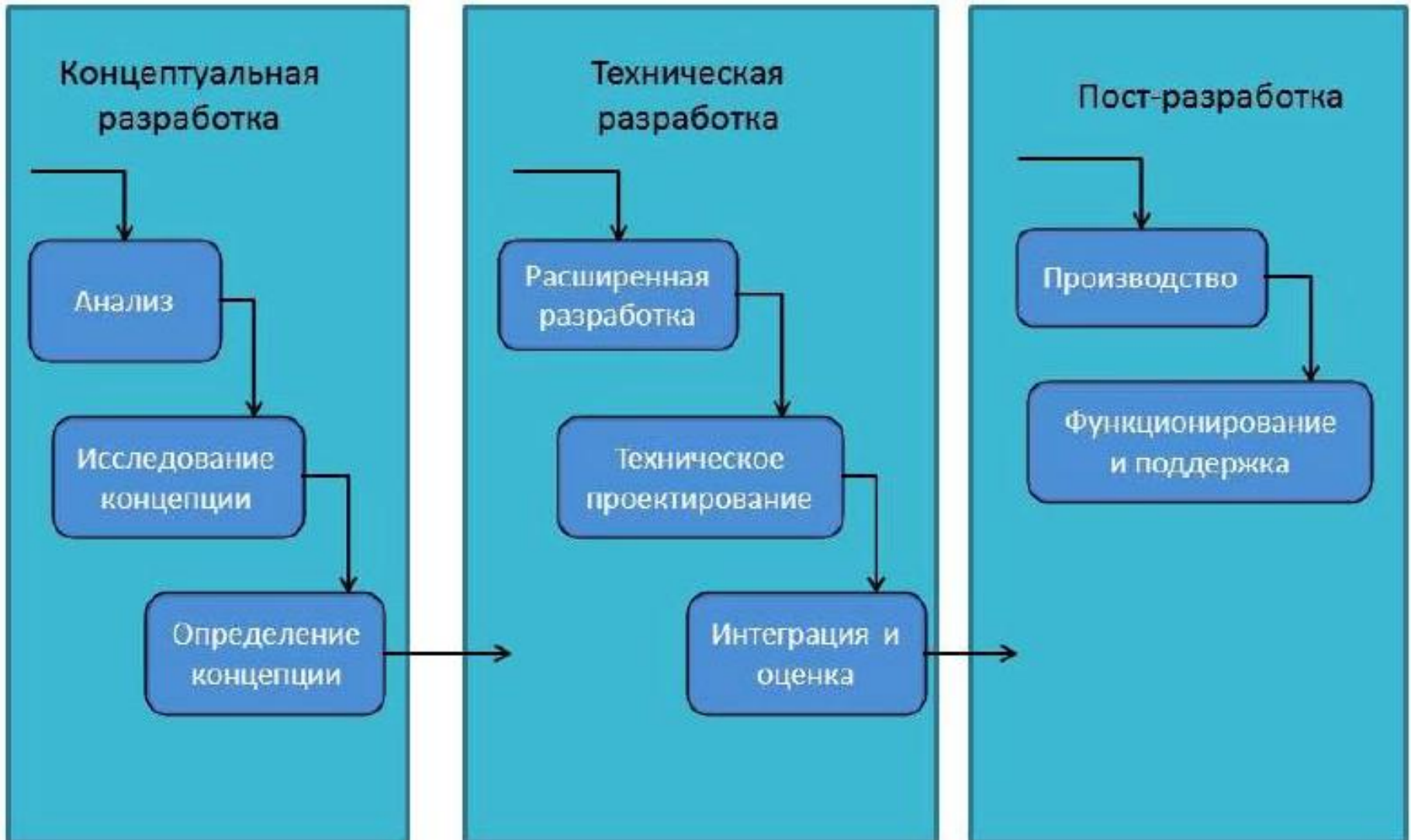
# ЖЦ НАСА – 8/9

Фаза	Назначение	Результат
Phase D System Assembly, Integration and Test, Launch	Сборка и интеграция системы (hardware, software, and humans) с параллельной проверкой удовлетворения требованиям. Запуск и подготовка к использованию	Готовая к использованию система со всеми обеспечивающими продуктами

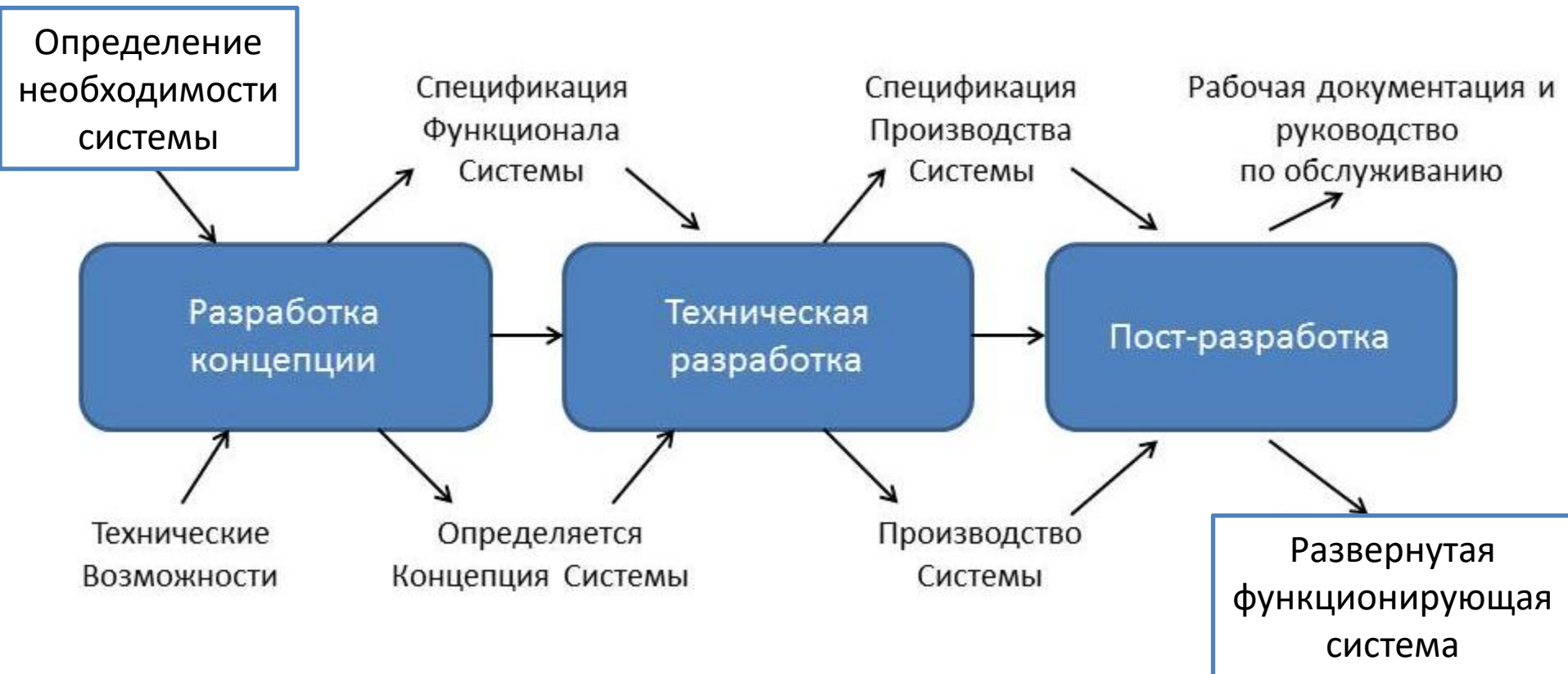
# ЖЦ НАСА – 9/9

Фаза	Назначение	Результат
Phase E Operations and Sustainment	Управлением выполнением миссии и достижение поставленных задач. Выполнение требующихся операций поддержки. Выполнение плана миссии	Требовавшаяся система
Phase F Closeout	Выполнение плана вывода из эксплуатации разработанного на фазе E. Анализ полученных результатов (данных и/или образцов)	Закрытие проекта

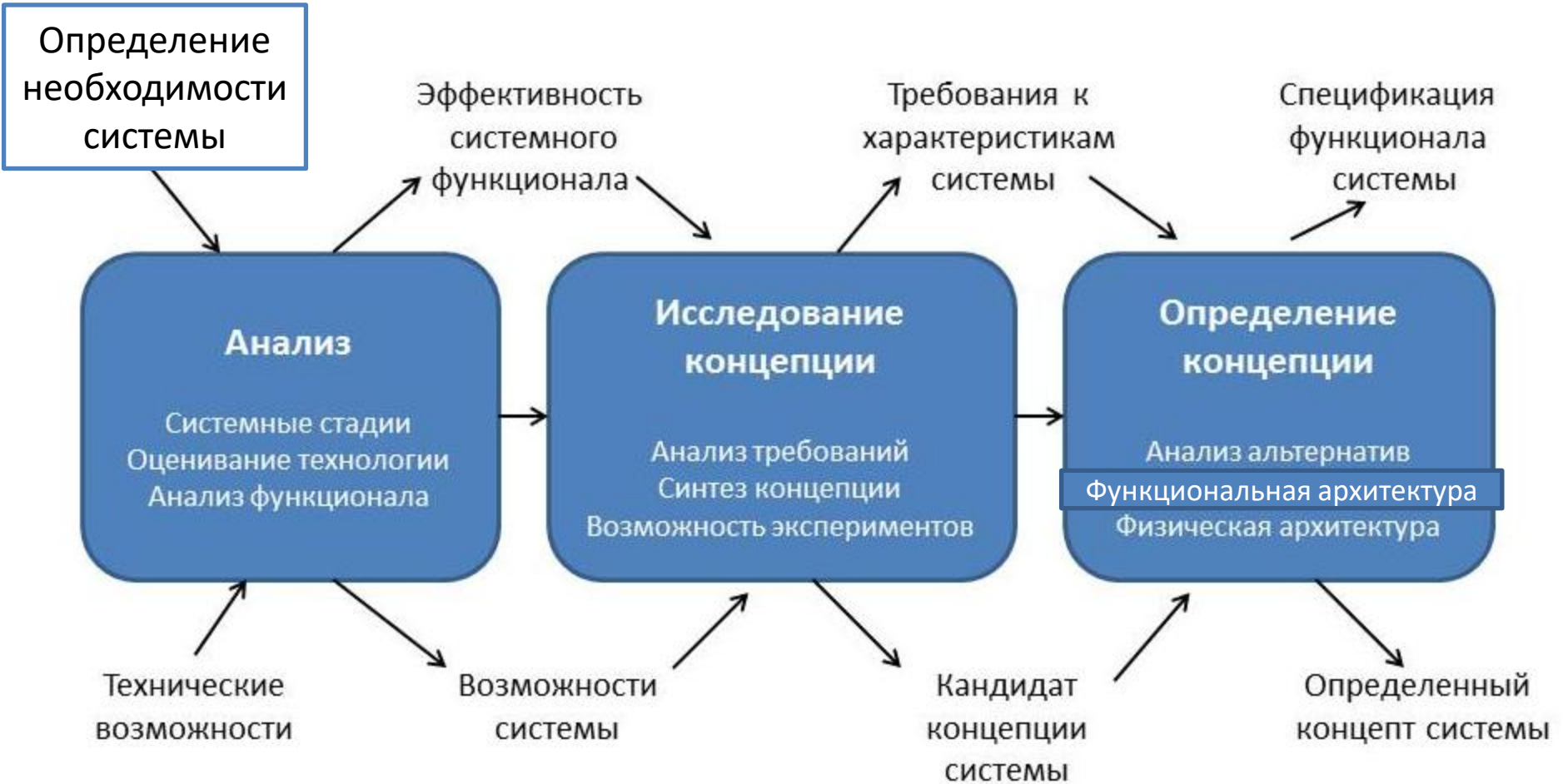
# ЖЦ по Kossiakoff – 1/2



# ЖЦ по Kossiakoff – 2/2



# Kossiakoff: Концептуальная разработка



# Kossiakoff: Техническая разработка



Определения

Гейты, вехи

ICM

**УПРАВЛЕНИЕ ЖЦ СИСТЕМЫ**

# Управление ЖЦ – 1/4

- Управление жизненным циклом (life cycle management)
  - инженерная дисциплина:
    - Охватывает полный ЖЦ системы (множество проектов)
    - Сосредотачивается не на «сдаче вовремя», а на:
      - Содержательном объединении работ разных стадий жизненного цикла
      - Применении необходимых инженерных практик
- Не путать с Управлением проектами:
  - Календарное планирование
  - Контроль выполнения плана
  - Обеспечение плана ресурсами

При этом:

- Отслеживание графика центрально
- Работает именно с проектами → проекты это малая часть ЖЦ



# Управление ЖЦ – 2/4

- Управление ЖЦ:

= Управление конфигурацией

- Включает управление инженерной документацией, управление ЖЦ продукта (product life cycle management) + управление информацией
- Основная задача — предотвращение конфигурационных коллизий (ошибок, возникающих от несоответствия и противоречивости различных документов и моделей друг другу, а также их несоответствие воплощённой системе)

# Управление ЖЦ – 3/4

- Управление ЖЦ (продолжение) :
  - Это распределение инженерных практик по стадиям ЖЦ
    - Практики относятся к инженерии требований, инженерии системной архитектуры, etc.
    - Распределение:
      - Последовательное выполнение практик
      - Параллельное выполнение практик
      - Регулярность проведения проверок
      - Ритмичность проведения совещаний
      - Ритмичность перепланирования
      - etc.

# Управление ЖЦ – 4/4

- Управление ЖЦ (продолжение):
  - = Ситуационная инженерия методов
    - Стандарты ситуационной инженерии методов:
      - OMG Essence
      - ISO 24744 – Software Engineering — Metamodel for Development Methodologies

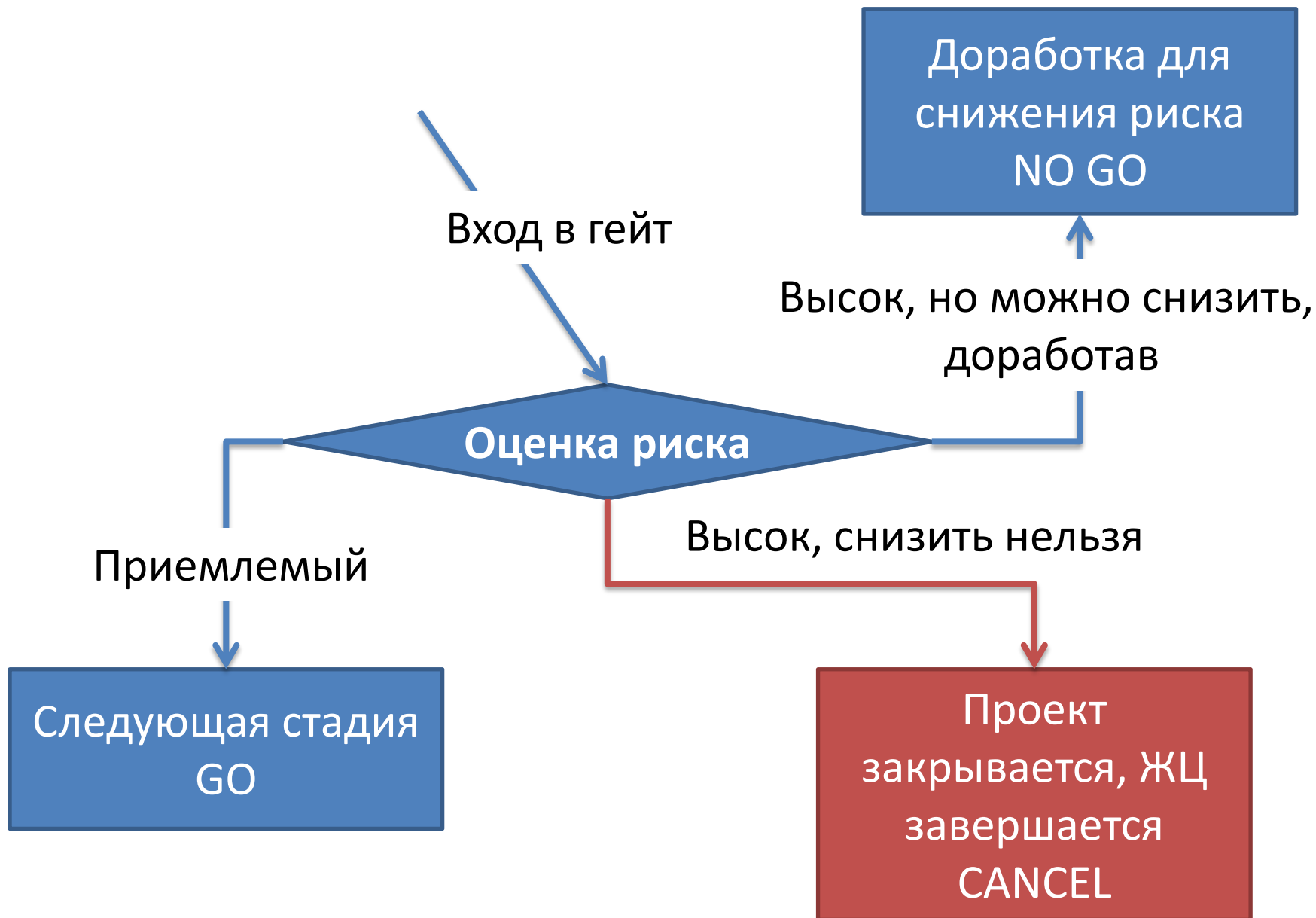
# Гейты – 1/5

- Гейт (decision gate, точка принятия решения) – переход со стадии на стадию ЖЦ
  - Гейт обычно находится «между проектами». Разные проекты выполняются разными командами, так как разные стадии ЖЦ требуют разной специализации
  - Решение, принимаемое в гейте – пересмотр выделения ресурсов на проект, выполняющаяся для синхронизации:
    - Параллельно ведущихся инженерных разработок
    - Менеджерских (логистических и инвестиционных) операций
  - Наиболее вероятный момент выявления проблем – стыковка/интеграция, поэтому гейты предусматривают именно там

# Гейты – 2/5

- Операции при прохождении гейтов:
  - Усиленный контроль конфигурации:
    - Сверки информационных систем
    - Создание проектных базисов — baselines («утверждённых версий» проектных моделей и документации, изменения в которых затем могут проводиться только по специальной процедуре со множеством дополнительных проверок)
  - Проведение испытаний, проверка инженерных обоснований (в том числе с привлечением независимых экспертов)
- Гейт завершается принятием осознанного решения «Go — No Go — Cancel»

# Гейты: Go — No Go — Cancel – 3/5



## Гейты – 4/5

- Решение «Go» означает принятие решения по дальнейшему финансированию проекта
  - Даже если деньги гарантированно были выделены на прохождение нескольких стадий ЖЦ, в гейтах это выделение ресурсов пересматривается — проходит commitment review
  - Одна из методик управления жизненным циклом называется — ICM, incremental commitment model (пошаговое выделение ресурсов)

# Гейты и вехи – 5/5

- Вехи — дополнительные точки контроля, на которых проверяется выполнение графика работ, но не ожидается принятие решений «Go-NoGo-Cancel»
- Гейты и вехи — это особенные стадии жизненного цикла
  - ISO 24744 – гейты и вехи не имеют продолжительности
  - В жизни гейты (и вехи тоже) могут занимать ощутимое время (чаще всего это верно для гейта «приёмка в эксплуатацию»
    - Пример: Для строек приемка может занимать до полугода. Если скрыть длительность, то откуда возьмутся ресурсы на проведение всех необходимых работ в эти полгода?
- Гейты и вехи позволяют договариваться менеджерам и системным инженерам:
  - Для менеджеров это даты принятия и исполнения бюджетных, ресурсных и других логистических решений
  - Для инженеров это означает сроки проведения инженерных мероприятий, выполнения всех необходимых для создания целевой системы работ



**ПОШАГОВОЕ ВЫДЕЛЕНИЕ РЕСУРСОВ –  
ICM (INCREMENTAL COMMITMENT  
MODEL)**

# Определение, источники

- Метод поэтапного выделения ресурсов (ICM, Incremental Commitment Model) – метод управления ЖЦ («software and systems process»)
  - <http://csse.usc.edu/csse/TECHRPTS/2009/usc-csse-2009-500/usc-csse-2009-500.pdf>

# Принципы – 1/4

## 1. Достаточность и подотчетность

- Выделяются достаточные человеческие ресурсы: системных инженеров, разработчиков и менеджеров
- Подотчетность обеспечивается короткими этапами разработки (development increment)
- Обоснование: слишком легко «переобещать и смыться» (overpromise and depart)

## 2. Согласование результата

- На начальном этапе: Переговоры по установлению удовлетворяющему всех СХ набору системных требований, решений и планов
- В дальнейшем: управление [предлагаемыми] изменениями, при условии сохранения взаимно удовлетворяющего результата

# Принципы – 2/4

## 3. Эволюционность

- Описание системы (system definition) и выделение ресурсов СХ (stakeholder commitment) поэтапно и эволюционно наращиваются (growth)
- Обоснование:
  - Требования и ресурсы для сложной системы не могут быть монолитными и/или полностью предварительно определены (специфицированы) → появляются постепенно по мере проведения экспериментов, прототипирования, использования ранних образцов
  - Доверие СХ, описание системы и выделение ресурсов развиваются эволюционно

# Принципы – 3/4

## 4. Итеративность

- Разработка и описание системы повторяются
- Обоснование: повторяющиеся этапы (итерации) приводят к постепенному улучшению требований, решений и планов разработки

## 5. Параллельность

- Одновременно происходят описание системы и ее разработка:
  - На начальном этапе:
    - Одновременное формулирование требований и решений
    - Интегрированное описание продукта и процесса
  - В дальнейшем: сочетание стабилизированной разработки текущего этапа с одновременной связанной с изменениями переработкой (rebaselining) требований, решений и планов базиса следующего этапа
  - Позволяет не ждать каждый раз, когда будут окончательно сформулированы будущие требования

# Принципы – 4/4

## 6. Обоснованное выделение ресурсов

- Определяются контрольные точки (milestones) для принятия решений о выделении ресурсов
- Для принятия решений требуется доказательство [возможности дальнейшего продвижения] с учетом управления рисками (evidence-based, risk-driven)
- В контрольных точках происходит оценка доказательства достижимости требований независимыми экспертами, после чего принимаются решения об изменении требований, выделении ресурсов, доработках или наоборот, пропусках стадий и т.д.

# Особенности – 1/2

- Поэтапное выделение (commit) ресурсов на разработку
  - Уменьшение риска СХ

# Предотвращение несостыковок – 1/2

- Проблема: Большие проекты → Множество параллельных работ → Рассинхронизация по времени и результатам → Требуются [многочисленные] переделки в момент обнаружения нестыковок
- Для предотвращения нестыковок по времени:
  - Создаются контрольные точки специального типа: точки привязки (anchor points)
    - Отдельные работы ведутся по принципу «по времени» (timeboxing – столько времени, сколько отведено)
    - Иногда говорят про «график как независимая переменная». Управление происходит как изменение baseline (rebaselining) – в меньшую или большую сторону в зависимости от запаса времени и бюджета



## Предотвращение несостыковок – 2/2

- Для предотвращения нестыковок по результатам:
  - В обязательные результаты стадии добавляется документ Обоснование реализуемости (Feasibility Evidence Description):
    - Готовится не к заданному сроку (независимо от состояния работ), а по итогам готовности результатов стадии
    - Целостно оценивает результаты стадии на момент ее завершения

# Feasibility Evidence Description – 1/5

- Документ Обоснования реализуемости, утверждает, что если система будет построена по предлагаемым описаниям (архитектуре, чертежам и т.д.), то она:
  - Будет удовлетворять требованиям:
    - возможностей (capability),
    - интерфейсов,
    - уровня обслуживания,
    - развития (evolution)
  - Будет поддерживать Концепцию эксплуатации (operational concept, набор сценариев использования)
  - Уложится в бюджет и в сроки, обозначенные в планах ее создания

# Feasibility Evidence Description – 2/5

- Документ Обоснования реализуемости (продолжение). Система:
  - Породит ожидаемый возврат на инвестиции
  - Породит удовлетворительные результаты для всех критических для успеха системы СХ
  - Избежит всех больших рисков. При этом:
    - Недостатки в доказательствах описываются как [новые] риски и
    - Учитываются в [обновленных] планах управления рисками
  - Послужит основанием выделения заинтересованными сторонами ресурсов для продолжения работ

# Feasibility Evidence Description – 3/5

- Обоснование реализуемости не может содержать только трассировку требований к решениям
- Документ является одним из основных результатов стадии:
  - На его подготовку и оценку независимыми экспертами выделяется достаточно времени и финансирования
  - Он должен включать в себя доказательство совместимости и целостности параллельно разработанных элементов (NB! нельзя обойтись доказательством реализуемости отдельных элементов)

# Feasibility Evidence Description – 4/5

- Документ может включать, например:
  - Результаты прототипирования: сетей, роботов, пользовательских интерфейсов, взаимодействия с покупными товарами
  - Замеры: производительности, масштабируемости, точности
  - Эксперименты: производительности, взаимодействия, безопасности
  - Модели (models): стоимости, графика работ, производительности, надежности, «развилки» (tradeoffs – вариантов)
  - Имитационные модели (simulations): масштабируемости, производительности, надежности
  - Ранние рабочие версии: инфраструктуры, интеграции данных с уменьшением их объема (data fusion), совместимости с предыдущими версиями (legacy compatibility)
  - Ссылки на прошлый опыт
  - Комбинации всего перечисленного

# Feasibility Evidence Description – 5/5

- Примеры плохих обоснований:
  - Наши инженеры чудовищно креативны. Они найдут для этого решение
  - Мы имеем три алгоритма, которые удовлетворяют техническим условиям на типовых маломасштабных примерах. Как минимум один из них можно масштабировать и обработать нетиповые ситуации
  - Мы все построим и затем наладим, чтобы удовлетворить техническим условиям
  - Поставщик готового оборудования заверил нас, что обеспечит сертифицированную по условиям безопасности версию к тому моменту, когда нам нужно будет сдавать работу
  - Мы уже демонстрировали решения для каждой подсистемы разным заказчикам. Нам потребуется просто интегрировать все вместе

**NB! При таких обоснованиях необходимо прототипировать!**

## Особенности – 2/2

- FED проверяется (валидируется) независимыми экспертами. Валидация, так как надо не только проверить (верифицировать) соответствие требованиям, но и репрезентативность выбранных для обоснования сценариев, полноты тестирования и т.д.
  - NB! Правило 20-80 (20% «неучтенных» сценариев обычно дают 80% трудоемкости переделок)

# Точки привязки

- Операции в точках привязки (anchor points):
  - Рассматриваются Обоснования реализуемости
  - Происходит пересмотр выделения ресурсов (commitment review) заинтересованными сторонами
  - Поэтому точки привязки часто так и называют: «Пересмотр выделения ресурсов»
  - Каждый пересмотр выделения ресурсов сопровождается принятием следующих решений:
    - GO: Переход к новой стадии (с утверждением новых требований и нового финансирования)
    - NO-GO: Доработки в рамках предыдущей стадии
    - CANCEL: Прекращение всего проекта
    - Пропуск следующей стадии ввиду незначительных рисков



# ЖЦ в ИСМ

- Два периода:
  - Период I: поэтапное (incremental) описание системы с тремя пересмотрами выделениями ресурсов:
    - исследования / нужды и возможности, готовность заинтересованных лиц
    - оценивание (valuation) / объем работы, бизнес-кейсы, высокоуровневая архитектура, форма ЖЦ
    - основание / детальные показатели, требования, архитектура, планы, выбранные партнеры-аутсорсеры
  - Период II: поэтапные (incremental) разработка и эксплуатация системы – повторяющиеся (iteration) стадии с регулярно повторяющимися пересмотрами:
    - стадия-(increment)-1 (параллельно): разработка-1 + основания-2 / в разработке-1 использование результатов основания-1 вплоть до валидации и верификации, в основах-2 пересмотр базиса для разработки-2 (в конце стадии две процедуры пересмотра выделения ресурсов: для разработки-1 и основание-1)
    - стадия-2 (параллельно): эксплуатация-1 + разработка-2 + основание-3
    - и так далее
- Для каждой стадии ЖЦ описаны рекомендации по составу практик и рабочих продуктов [для их воплощения]

Линейный

Инкрементный

Итерационный

**ВИДЫ ЖЦ СИСТЕМЫ**

# ЛИНЕЙНАЯ МОДЕЛЬ

# Waterfall – 1/2

Определение требований

Проектирование системы

Реализация и тестирование

Интеграция и тестирование

Функционирование и  
поддержка



# Waterfall – 2/2

- Достоинства:
  - Формализованная, результатом каждой фазы является документ (утвержденный и подписанный)
  - Заказчик не вмешивается в большую часть работы
- Недостатки:
  - Линейность
  - Требования в дальнейшем не уточняются
  - Все архитектурные решения принимаются на ранней стадии

# **ИТЕРАТИВНАЯ МОДЕЛЬ**

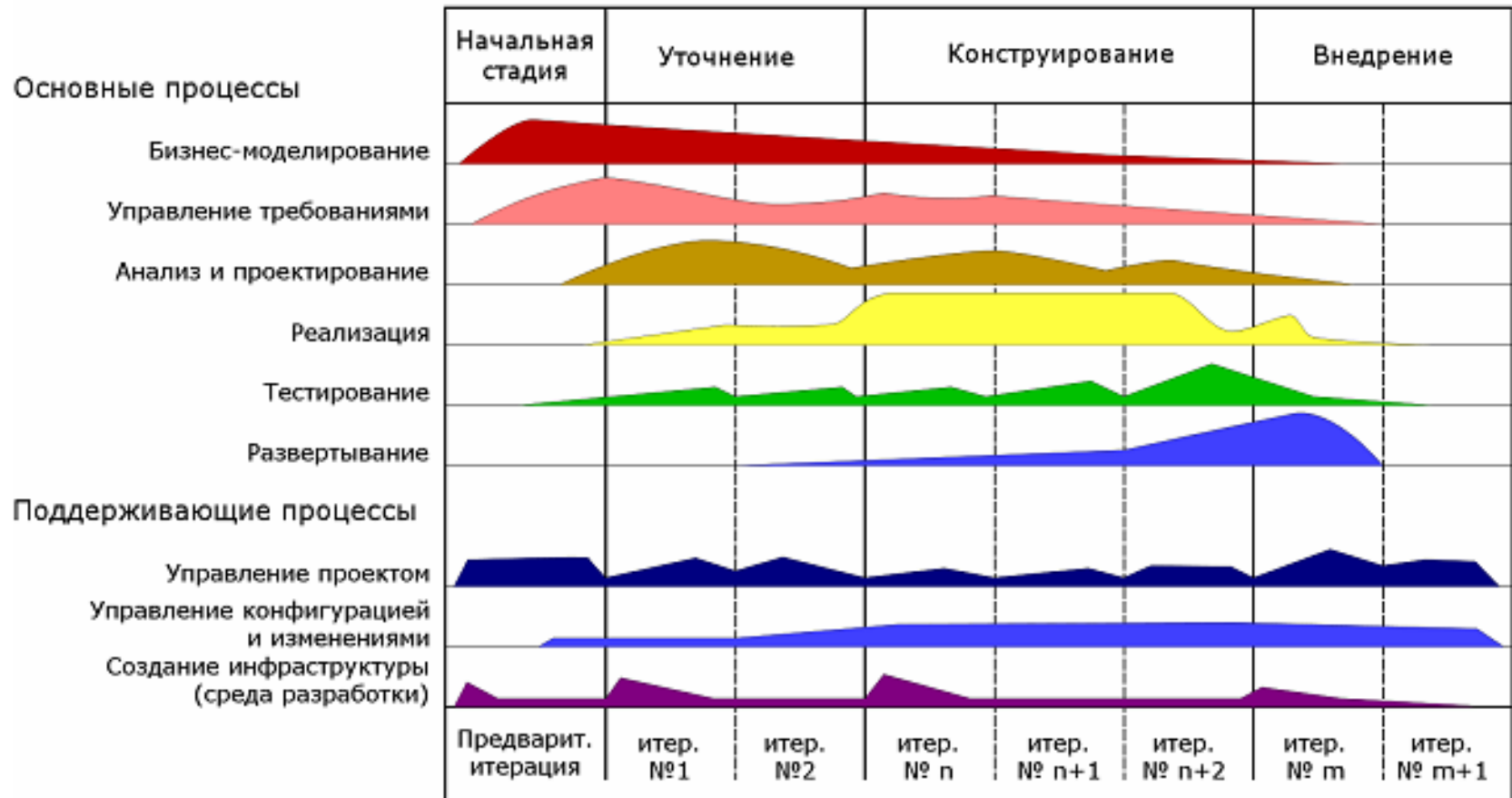
# Итеративная модель – 1/5



# Итеративная модель – 2/5

Рабочие процессы

Стадии



Итерации

Rational Unified Process (RUP)



# Итеративная модель – 3/5

- Достоинства:
  - Снижение проектных рисков и минимизация затрат на их устранение
  - Эффективная обратная связь с потребителем
    - Концентрация усилий на наиболее важных и критичных направлениях
    - Раннее обнаружение конфликтов между требованиями, моделями и реализацией проекта
  - Непрерывное итеративное тестирование, позволяющее оценить успешность всего проекта в целом

# Итеративная модель – 4/5

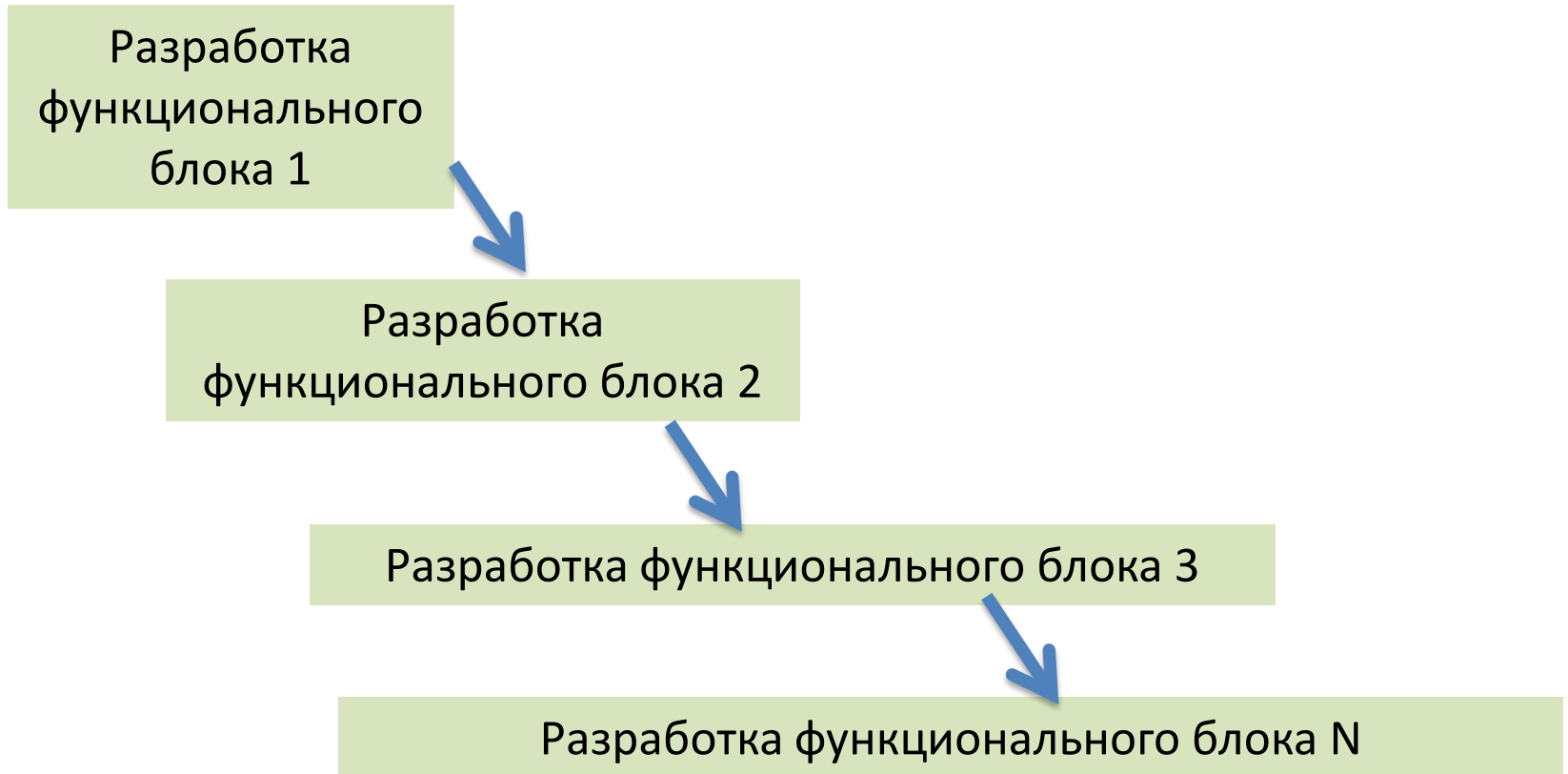
- Достоинства (продолжение):
  - Более равномерная загрузка участников проекта
  - Эффективное использование накопленного опыта
  - Реальная оценка текущего состояния проекта и, как следствие, большая уверенность заказчиков и непосредственных участников в его успешном завершении

# Итеративная модель – 5/5

- Недостатки:
  - Неопределенность сроков завершения проекта при изменяющихся внешних условиях
  - Постоянная модификация архитектурной части проекта
  - Неэффективность и/или большой объем тестирования

# **ИНКРЕМЕНТНАЯ МОДЕЛЬ**

# Инкрементная модель – 1/2



# Инкрементная модель – 2/2

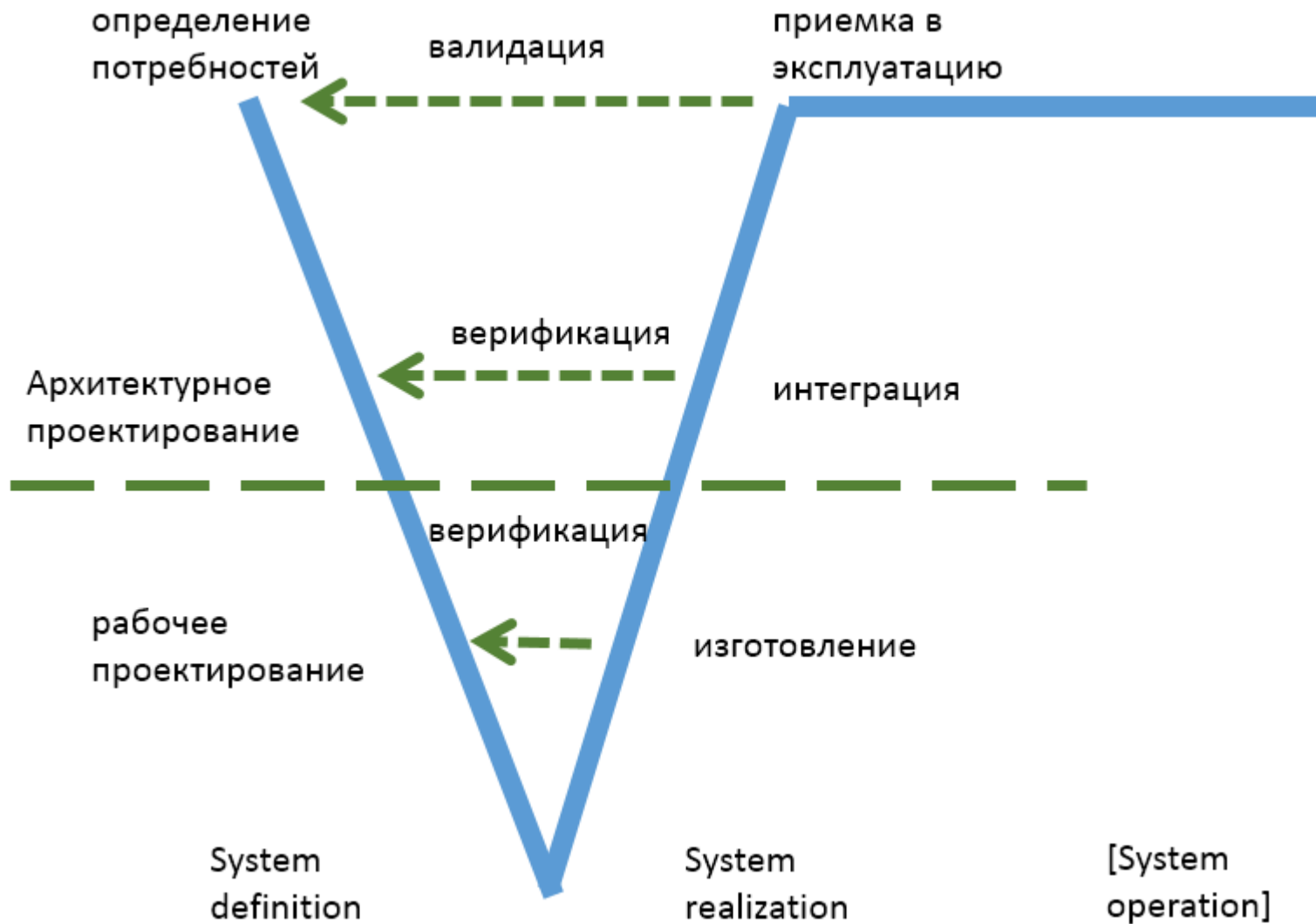
- Достоинства:
  - Быстрое получение частичной функциональности
  - Возможность корректировки последовательности разработки блоков
- Недостатки:
  - Разработка каждого последующего блока может влиять на функционирование предыдущих
  - Сложность обеспечения взаимодействия между блоками

V-диаграмма

«Горбатая» диаграмма

**ФОРМАЛИЗМЫ ЖЦ СИСТЕМЫ**

# V-диаграмма – 1/3





# V-диаграмма – 2/3

- V-диаграмма проявляет следующие черты системноинженерного процесса:
  - Разница между:
    - Практиками определения системы → работа с информацией
    - Реализации системы → работы с веществами и полями
    - Использованием системы
  - Видна идея максимизации затрат [ресурсов] на более ранних стадиях → для экономии ресурсов на более поздних стадиях
  - Соответствие определений и воплощений системы, поддерживаемое через проверки (верификация) и приёмки (валидация)
  - Ведущие практики жизненного цикла: дисциплины-рабочие продукты-инструменты

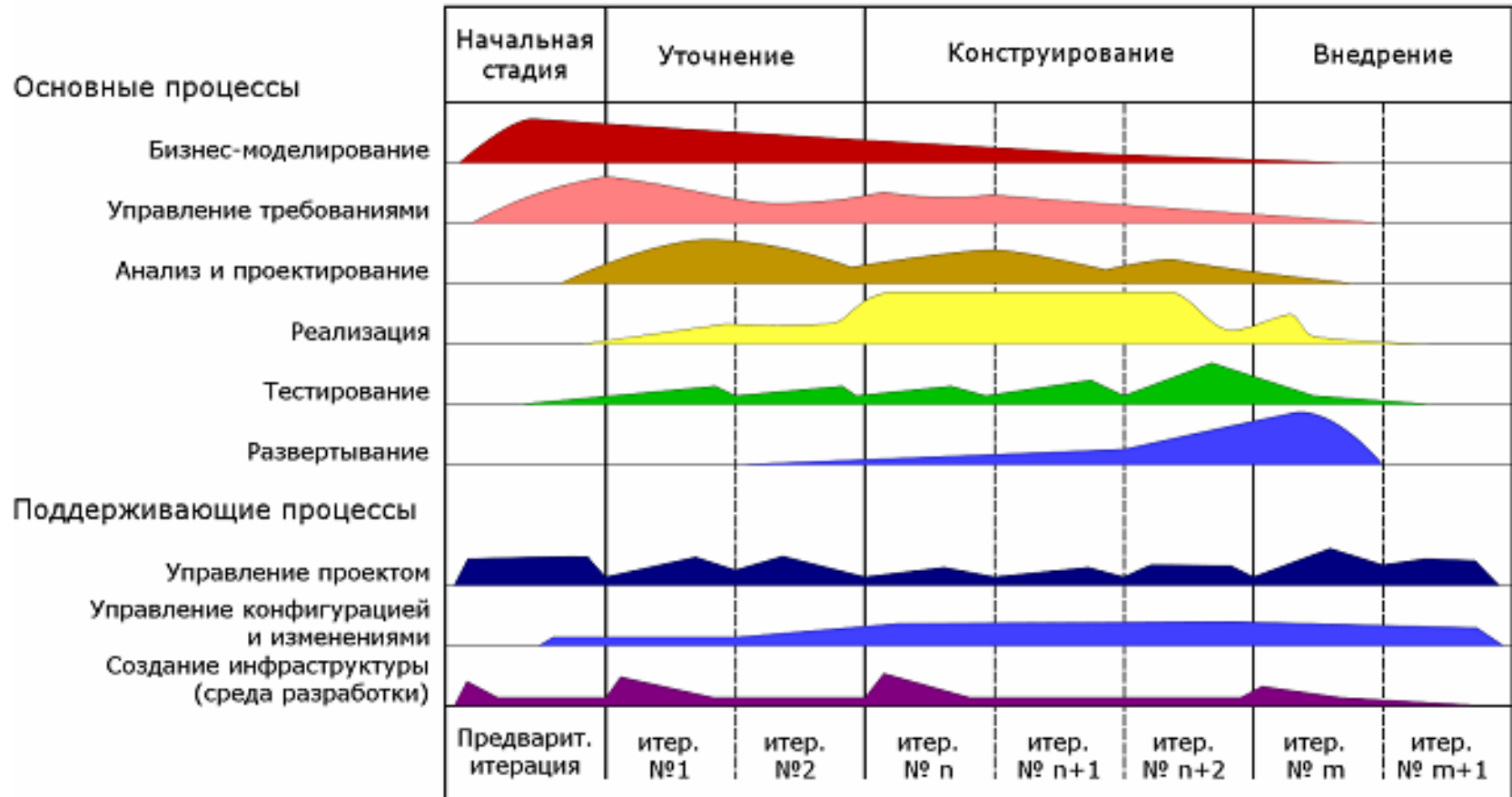
# V-диаграмма – 3/3

- Черты системноинженерного процесса:
  - Разницу между
    - Системноинженерными практиками (выше пунктирной линии) → имеют дело с системой в целом
    - «Обычными» инженерными практиками → имеют дело с частями системы
  - Взаимодействие между практиками
    - Работа идёт отнюдь не только в той практике-стадии, которой соответствует точка времени на диаграмме
    - Одновременно задействована вся «вертикаль» практик:
      - Архитектор общается и с инженерами по требованиям, и с занимающимися рабочим проектированием
      - Инженер-интегратор общается и с эксплуатационщиками, и с производителями оборудования

# «Горбатая» диаграмма

Рабочие процессы

Стадии



Итерации

Rational Unified Process (RUP)

# **ВОДОПАД VS. AGILE**

# Типы моделей ЖЦ – 1/2


- Разные модели ЖЦ занимают место между предсказывающими (predictive) и адаптивными (adaptive)
- **Предсказывающие** модели предполагают разработку детальных требований, подробное планирование и минимальное планирование итераций между стадиями разработки
  - Исходно, предсказывающие модели предполагали последовательное выполнение стадий разработки ПО с пересмотром результатов предыдущих стадий только в случае острой необходимости

# Типы моделей ЖЦ – 2/2

- **Адаптивные** модели разработаны для приспособления к внезапным изменениям требований и итеративной подгонке планов
  - Адаптивные модели характеризуются циклическим повторением циклов разработки
- **Промежуточные** модели предполагают наращивание [готовой] функциональности:
  - либо на основе заранее определенного графика (ближе к «предсказывающей» части шкалы классификации)
  - либо с очередным обновлением продукта в коротких циклах разработки (ближе к «адаптивной» части шкалы классификации)

# SDLC

## Предсказывающие

- 
- Линейная [«водопадная» (waterfall), каскадная]
  - Итеративная
  - Инкрементная
  - Спиральная
  - Гибкие модели (agile)

## Адаптивные