Системная инженерия

Моделеориентированная системная инженерия

Основные вопросы

- Моделеориентированная СИ
- Методика СИ, базирующаяся на функциях (Functions-Based Systems Engineering Method)
- Объектно-ориентированная методика СИ (Object-Oriented Systems Engineering Method)
- Управление конфигурацией и изменениями

МОДЕЛЕОРИЕНТИРОВАННАЯ (MODEL-BASED) СИ

Введение

- MBSE (Model-Based Systems Engineering) – методология СИ, включающая преимущества моделеориентированного подхода «поверх» более традиционного подхода, основанного на документах
  - Ряд моделей и методов моделирования были формализованы и включены в процессы СИ → Эти процессы являются основой MBSE
- MBSE – формализованное применение моделирования для поддержки процессов определения системных требований, проектирования, анализа, верификации и валидации, начиная с этапа концептуального проектирования и продолжая в течение всего периода разработки и последующих этапов ЖЦ [INCOSE Systems Engineering Vision 2020]

Преимущества – 1/2

- MBSE расширяет возможности сбора, анализа, обмена и управления информацией, связанной со спецификацией продукта, что дает следующие преимущества:
  - Общие представления → Упрощение взаимодействия между СХ
  - Управление сложностью системы, так как есть возможность рассматривать модель системы с разных точек зрения (и уровней представления) и анализировать влияние изменений

Преимущества – 2/2

- Однозначная и точная модель системы → Можно оценить согласованность, правильность и полноту → Улучшение качества результата (продукции)
- Модельно-ориентированный подход → Стандартизация процедур и использование встроенных механизмов абстракции → Упрощает сбор и структуризацию знаний, а также их дальнейшее использование → Сокращение цикла и снижение затрат на техническое обслуживание для изменения проектных решений
- Формализация концепций, структуризация → Упрощение преподавания и изучения основ SE

Обзор – 1/2

- MBSE часто противопоставляется традиционному подходу к SE, основанному на документах. В «документальном» подходе SE информация о системе содержится в документах и других артефактах:
  - Спецификациях
  - ICD (документах управления интерфейсом)
  - Документах описания системы
  - Отчеты о выборе альтернатив
  - Аналитических отчетах
  - Планах тестирования (верификации)
  - Процедурах
  - Отчетах

  Информацию в этих документах часто трудно поддерживать в согласованном состоянии и синхронизировать. Также сложно оценить ее правильность, полноту и согласованность

Обзор – 2/2

- MBSE формализует применение СИ посредством использования моделей. В подходе MBSE большая часть информации о системе фиксируется в системной модели или наборе моделей
- Модель системы является основным артефактом процесса СИ
- Степень, в которой информация о системе собирается в моделях и поддерживается в течение всего ЖЦ, зависит от объема работ MBSE

Обзор методологий MBSE – 1/2

- Обобщенно, методология может быть определена как совокупность связанных процессов, методов и инструментов, используемых для поддержки конкретной дисциплины
- Общее понятие методологии специализируется для методологии MBSE: совокупность связанных процессов, методов и инструментов, используемых для поддержки СИ в контексте «на основе модели» (model-based) или «управляемой моделями» (model-driven)
- В 2008 году под было проведено исследование методологий-кандидатов :
  - INCOSE - метод проектирования объектно-ориентированных систем (OOSEM)
  - IBM Rational Telelogic Harmony-SE
  - IBM Rational Unified Process для системного проектирования (RUP - SE)
  - Методология Vitech MBSE
  - Анализ состояния JPL (SA)
  - Дори объектно-процессная методология (OPM)

  Подробнее эти методологии описаны на вики-инициативе INCOSE MBSE http://www.omgwiki.org/MBSE/doku.php

Обзор методологий MBSE – 2/2

- SE handbook описывает два примера MBSE методологий:
  - Метод СИ на основе функций (FBSE)
    - Формально, этот метод не позиционируется как моделеориентированный, но существуют другие основанные на функциях методы (например, методология Vitech MBSE), которые явно образом основаны на модельном подходе
  - ОО-метод СИ (OOSEM)
    - OOSEM определяется как сквозной (end-to-end) метод MBSE, где артефактами метода являются моделирующие артефакты, которые управляемые и отслеживаемые в течение всего процесса СИ

FUNCTIONS-BASED SYSTEMS ENGINEERING METHOD

Введение – 1/2

- FBSE – это подход СИ, отталкивающийся от функциональной архитектуры системы
- Функция – это характерная задача, действие или операция, которые необходимо выполнить для достижения желаемого результата. Функция может выполняться одним или несколькими элементами системы, включающими:
  - Оборудование (аппаратное обеспечение)
  - ПО
  - Встроенное ПО (firmware)
  - Помещения и объекты инфраструктуры (facilities)
  - Персонал
  - Описания выполняющихся процедур (procedural data)

Введение – 2/2

- Целями FBSE являются:
  - Создание функциональной архитектуры, для которой могут быть разработаны системные продукты и процессы
  - Обеспечение основы для определения архитектуры системы

  посредством распределения функций и подфункций по аппаратному / программному обеспечению, БД, помещениям, объектам инфраструктуры и персоналу
- FBSE описывает, что будет делать система, а не как она это сделает
- В идеале процесс описания начинается только после того, как все системные требования были полностью определены. Зачастую это невозможно → эти задачи выполняют итеративно, функциональная архитектура дополнительно определяется по мере развития системных требований

Введение – 2/2

- Целями FBSE являются:
  - Создание функциональной архитектуры, для которой могут быть разработаны системные продукты и процессы
  - Обеспечение основы для определения архитектуры системы

  посредством распределения функций и подфункций по аппаратному / программному обеспечению, БД, помещениям, объектам инфраструктуры и персоналу
- FBSE описывает, что будет делать система, а не как она это сделает
- В идеале процесс описания начинается только после того, как все системные требования были полностью определены. Зачастую это невозможно → эти задачи выполняют итеративно, функциональная архитектура дополнительно определяется по мере развития системных требований

Общее описание метода – 1/6

- FBSE является итеративным процессом, даже в пределах одной стадии жизненного цикла системы
- На верхнем уровне функциональная архитектура определяется как набор функций (в вырожденном случае, единственной функцией системы, а все остальные являются подчиненными), которые задаются документом / спецификацией требований. Каждая функция задается требованиями:
  - Функциональными
  - По производительности
  - Эксплуатационными
  - Описывающими ограничения (способа реализации или взаимодействия с внешним миром и другими системами)

Общее описание метода – 2/6

Общее описание метода – 3/6

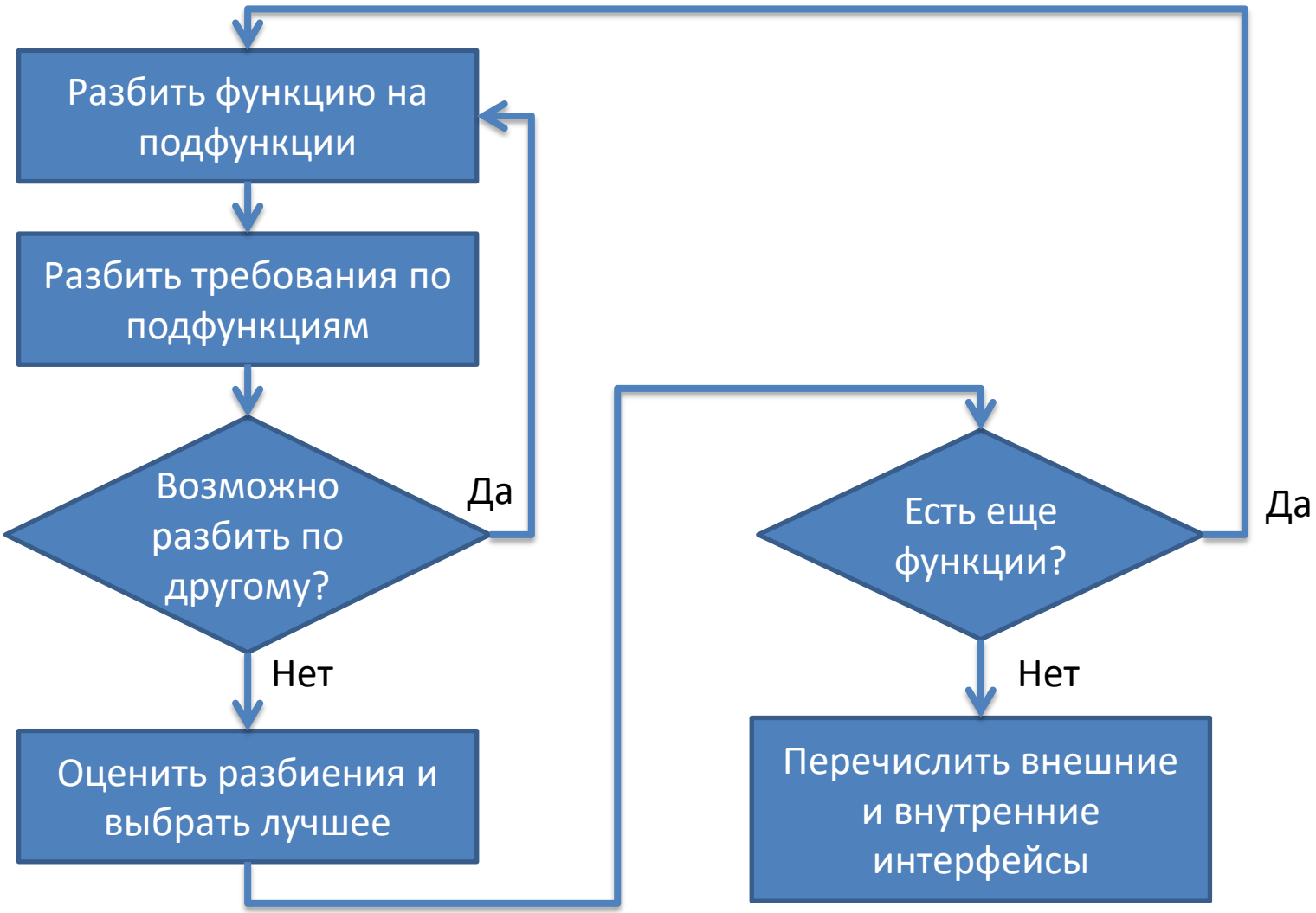
- Каждый более низкий уровень функциональной архитектуры детализируется и оценивается для определения, требуется ли дальнейшее разбиение. Если нужно, то процесс повторяется [для каждой функции текущего уровня], пока дальнейшее разбиение не окажется ненужным

Общее описание метода – 4/6

- FBSE итеративно повторяется для требований, находящихся на текущем уровне:
  - Функциональных
  - Связанных с основным назначением системы
  - По производительности
  - Задающих конструктивные ограничения системы
  - Задающих архитектуру и дизайн

  в результате:
  - Определяется соответствие требований следующего уровня требованиям текущего уровня
  - Формируются альтернативные наборы требований следующего уровня
  - Уточняется определение продукта и выбор технологических решений

Общее описание метода – 5/6



Общее описание метода – 6/6

- FBSE исследует определенную функцию, чтобы определить все подфункции, необходимые для выполнения этой функции. При анализе следует:
  - Учитывать все режимы использования
  - Каждая функция, «появляющаяся» для удовлетворения эксплуатационных требований, рассматривается с точки зрения распределенных функциональных, эксплуатационных и других ограничивающих требований
- При разложении функции на подфункции, требования, связанные с этой функцией, разлагаются вместе с ней
- На каждой итерации процесса могут быть рассмотрены и оценены альтернативные декомпозиции и распределения для каждой функции и одной выбранной версии
- В ходе декомпозиции полностью определяются интерфейсы между каждой из функций и подфункций, а также интерфейсы с внешним миром

Функциональная архитектура – 1/2

- Для выявленных подфункций указывают:
  - Иерархию (наследование от верхнего уровня)
  - Интерфейсы (внутренние и внешние): входные, выходные и функциональные
- Требования по производительности:
  - Устанавливаются последовательно иерархически для каждого функционального требования и интерфейса. При «спуске» по иерархии требования по производительности распределяются по подфункциям
  - Временные требования, которые являются предварительным условием для функции или набора функций, должны быть определены и распределены
  - Результирующий набор требований должен быть определен:
    - В терминах измеримых параметров/характеристик
    - Достаточно подробен для использования в качестве критериев проектирования

Функциональная архитектура – 2/2

- Требования по производительности (продолжение):
  - Для этих требований, хотя они проявляются на всех уровнях иерархии, иногда необходимо оценить/проанализировать несколько уровней декомпозиции, прежде чем разбивать требования к производительности по подфункциям
  - Также, иногда необходимо рассмотреть альтернативные функциональные архитектуры и провести их сравнительный анализ, чтобы определить предпочтительное разбиение. На каждой итерации FBSE оцениваются альтернативные декомпозиции и определяются все интерфейсы

Представление ФА – 1/5

- Функциональная архитектура может быть представлена различными способами, например:

1. IPO (Input–process–output) диаграммы - вид диаграмм потока данных, относящегося к определенному уровню декомпозиции системы. Эта диаграмма изображает все входы и выходы системы, но не показывает декомпозиции
2. Диаграммы поведения – описывают поведение, как ответы на воздействия на уровне системы. Описание [алгоритмов] реакции на воздействия может включать:
   - Временные последовательности операций
   - Параллельно выполняющиеся последовательности
   - Выбор [ответных] действий в зависимости от условий
   - Точки синхронизации (параллельно выполняющихся последовательностей)
   - Информацию о состоянии системы
   - Производительность

Представление ФА – 2/5

3. Диаграммы потоков управления – описывают набор всех возможных последовательностей, в которых операции могут выполняться системой или программным обеспечением. Существует несколько видов представления потоков управления:
   - Структурные схемы (sic! Может быть структура функций и связей между ними)
   - Блок-схемы
   - Диаграммы перехода состояний
4. Диаграммы потоков данных (Data flow diagrams – DFD) – описывает взаимосвязь операций, выполняемых системой. В качестве связей выступают входные и выходные данные. Также на схеме показываются промежуточные хранилища данных. Каждая DFD должна быть проверена на соответствие IPO или DFD более высокого уровня
5. Диаграммы «сущность-связь» (ER) – описывают набор объектов (в т.ч. функций) и логические отношения между ними

Представление ФА – 3/5

6. Функциональные [блок-]схемы (Functional flow block diagrams – FFBD) – связывают входы и выходы функций системы в многоуровневую пошаговую блок-схему последовательности операций https://en.wikipedia.org/wiki/Functional_flow_block_diagram
7. IDEF диаграммы (Integrated definition for functional modeling – IDEF) – описывают связи между функциями по последовательным входным и выходным потокам. Также показываются элементы управления процессом и ресурсы, необходимые для выполнения функции
8. Словари данных – документы, описывающие потоки данных, элементы данных, файлы и т.д. Используются для согласования понимания между организациями, занимающимися разработкой системы

Представление ФА – 4/5

9. Модели – абстрактное представление нужных характеристик системы. Используются для облегчения понимания, общения, проектирования и оценки системы. Они используются до того, как система будет построена и пока она проверяется или находится в эксплуатации
10. Результаты имитационного моделирования – данные, порожденные имитационными моделями поведения / работы системы при использовании наборов входных значений

Представление ФА – 5/5

- Целью функциональной декомпозиции является разработка иерархии FFBD, отвечающей всем функциональным требованиям системы
  - NB! Эта иерархия лишь необходимая, но не достаточная часть ФА
  - ФА не завершена, пока не будут учтены все требования по производительности, ограничения на реализацию
- Должно быть разработано описание каждой функции в иерархии, включающее:
  - Его место в структуре, показывающее ее взаимосвязь с другими функциями уровня (например, в виде диаграмм FFBD или IDEF0 / IDEF1)
  - Набор функциональных требований, которые были ему назначены, и определение того, что он делает
  - Описание входов и выходов (внутренних и внешних)
- Нет ни одного предпочтительного способа представления ФА. Использование более чем одного формата позволяет «проверить и сбалансировать» процесс анализа и поможет в общении с командой разработчиков системы

Tools

- Инструменты FBSE включают:
  - Средства для проведения анализа
  - Средства для построения моделей (схем) и создания имитационных моделей
  - Средства прототипирования
  - Инструменты поддерживающие трассировку требований

FBSE Measures

- В FBSE для измерения/оценки процессов/продуктов могут быть использованы, например, следующие меры:
  - Количество завершенных обоснованных выборов среди альтернатив относительного общего числа рассматриваемых
  - Процент завершения действий по анализу
  - Количество функций без соотнесенных им требований
  - Количество недекомпозированных функций
  - Количество вариантов декомпозиции
  - Количество не [полностью] описанных внутренних и внешних интерфейсов
  - Глубина [текущей] иерархии функциональной декомпозиции относительно требуемой (target depth)
  - Процент требований по производительности, относящихся с нижнему уровню функциональной иерархии

OBJECT-ORIENTED SYSTEMS ENGINEERING METHOD

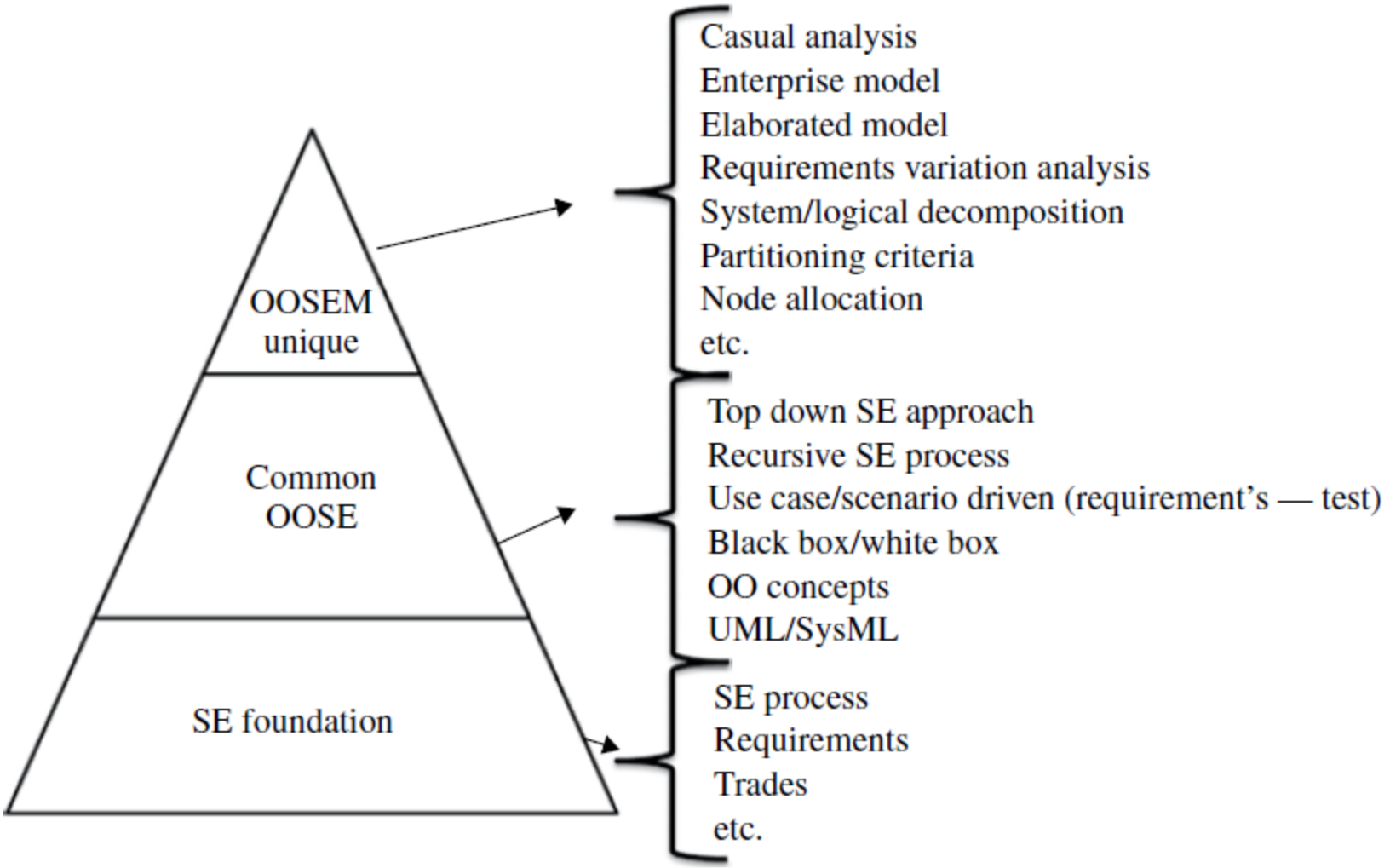
Введение

- OOSEM (Estefan, 2008) объединяет концепция объектно-ориентированного подхода с традиционными методами SE, основанными на моделях, чтобы помочь в разработке гибких и расширяемых систем, которые соответствуют развивающимся технологиям и меняющимся требованиям
- OOSEM применим в процессах спецификации, анализа, проектирования и тестирования систем
- Результаты, полученные в ходе OOSEM используются также:
  - [Легко!] при разработке ПО если она ведется на базе ООП
  - При разработке HW
  - Во время верификации и валидации

Цели

- Цели OOSEM следующие:
  - Сбор в течение всего ЖЦ информации, достаточной для определения, анализа, проектирования, проверки и проверки систем
  - Интеграция методов MBSE с объектно-ориентированным программным, аппаратным и другими инженерными методами
  - Поддержка процессов повторного использования и развития (design evolution) системы

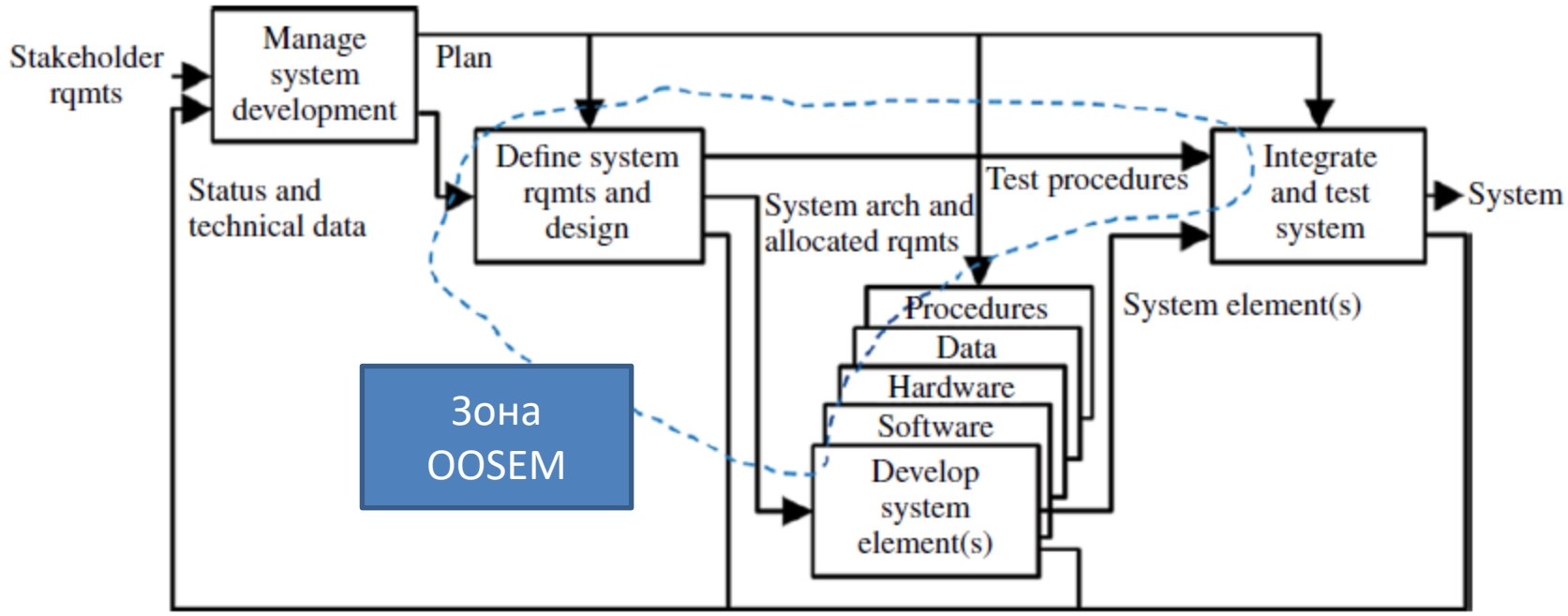
Состав OOSEM – 1/2



Состав OOSEM – 2/2

- OOSEM включает в себя основополагающие практики СИ, концепции объектно-ориентированного подхода и другие уникальные методы, позволяющие справиться со сложностью системы
- Технологически, в OOSEM используется SysML, поддерживающий:
  - Использование классов и объектов
  - Концепции инкапсуляции и наследования
- Уникальные для OOSEM методы:
  - Параметрическое сокращение потока
  - Системное / логическое разложение
  - Анализ изменений требований

Место OOSEM – 1/2



Место OOSEM – 2/2

- OOSEM включаются СИ в виде подпроцессов в показанные на схеме процессы:
  - Управления разработкой системы
    - Планирования и контроля технических усилий
    - Управление рисками
    - Управление конфигурацией
    - Измерения
  - Определения системных требований и дизайна
    - Определение системных требований
    - Разработку архитектуры системы
    - Распределение системных требований по системным элементам
  - Разработки элементов системы
    - Разработка, реализация и тестирование элементов, удовлетворяющих применимым к нему требованиям
  - Интеграции и тестирования системы
    - Интеграция элементов системы и проверка на соответствие системным требованиям, индивидуально и совместно

Место OOSEM – 3/3

- Процессы OOSEM вписываются в V-процессы ЖЦ. Они может применяться рекурсивно и итеративно на каждом уровне иерархии системы
- Подробной OOSEM описывается в документах:
  - Object-Oriented Systems Engineering Method (OOSEM) Tutorial, Version 3.11. Bethesda
  - Tutorial Material—Model-Based Systems Engineering Using the Object-Oriented Systems Engineering Method (OOSEM)

Термины и определения

Инструменты

УПРАВЛЕНИЕ КОНФИГУРАЦИЕЙ И ИЗМЕНЕНИЯМИ

Определения

- Управление конфигурацией (configuration management) – это про связь наших ожиданий (проектов) с реальностью

А. И. Левенчук

- TBD

Общие понятия – 1/3

- Конфигурация системы — это то, из чего на какой-то момент времени состоит система. Различают:
  - Конфигурацию системы (воплощенной или воплощаемой)
  - Конфигурацию определения системы
- А. И. Левенчук: Конфигурация – это знание о том, какова работающая (актуальная) система, выделенная из множества ее возможных вариантов
  - Например:
    - Какие насосы лежат запчастями на складе, а какие таки установлены и работают
    - Какой из пяти рассматривавшихся вариантов реализации предохранительного клапана пошёл в проект, а какие только остались в виде эскизов
    - Какие из требований были отклонены, а какие утверждены и поэтому требуют внимания

Общие понятия – 2/3

- Конфигурация находится под контролем, если конфигурация определения системы соответствует конфигурации воплощения системы
- NB! В больших системах непрерывно меняется как определение, так и воплощение системы

Общие понятия – 3/3

- Конфигурационная коллизия – состояние, когда части этих конфигураций не соответствуют друг другу. Примеры (по частоте возникновения):
  - Части описания не согласованы
  - Реально существующая система не соответствует описанию
  - Представление системы о системе в разных компонентах отличается
- Контроль конфигурации обеспечивается тем, что конфигурация может быть собрана в любой момент. Это требует:
  - Знаний о том, где лежат отдельные описания
  - Контроля разных версий отдельных частей описания

Управление конфигурацией – 1/2

- Практика **управления конфигурацией** (configuration management) – практика системноинженерного менеджмента – обеспечивает контроль за конфигурацией (configuration control), то есть поддерживает целостность системы на протяжении всего жизненного цикла
  - Практика управления конфигурацией включает получение различных описаний. Например, выпускаются различные виды спецификаций закупаемого/изготавливаемого оборудования/изделий — BOM (bill of materials, список комплектующих)

Управление конфигурацией – 2/2

- Управление конфигурацией + системная инженерия → конфигурация обязательно относится ко всей системе → Управляющий конфигурацией – системный инженер, также, как инженер по требованиям, системный архитектор, интегратор

ИС управления конфигурацией – 1/2

- ИС управления версиями – программная инженерия
- PDM (product data management) – системы хранения информации проекта (design)
- PLM (product life cycle management) – PDM + система управления изменениями + поддержка интерфейсов в ИС других стадий ЖЦ (например, системы закупок)

ИС управления конфигурацией – 2/2

- EAM (enterprise asset management) – система управления активами – используется для учёта установленного оборудования стадии эксплуатации

Термины

- Базис (baseline) – утвержденная конфигурация
- Версия/ревизия (version/revision) – конкретная конфигурация (на конкретный момент времени, в конкретном месте)
- «Управленческиконфигурационный шовинизм» добавляет общую теорию учёта/регистрации – каким образом обеспечивать взаимное соответствие реальности и записей о реальности
  - Примечание: Общее знание для бухгалтерского, депозитарного, складского, регистрационного, конфигурационного учёта

Практики – 1/2

- Идентификация – поддержка инженерных разбиений (классификаций, кодировок) и именования/кодировки отдельных конфигурационных единиц (configuration items)
  - Именно тут обсуждаются PBS, GBS, WBS и разные системы кодировок типа RDS-PP, KKS, RTM, S1000D и т.д.
- Контроль версий (version/revision control) – обеспечение того, что базис (утвержденная для каких-то целей конфигурация) собирается из взаимно соответствующих версий частей системы
  - Версии могут относиться к проектной или исполнительной документации, или к самой системе

Практики – 2/2

- Конфигурационный учет/регистрация – административное обеспечение взаимного соответствия:
  1. Проекта (включая требования)
  2. Исполнительной документации (as built, «что мы думаем о реальной системе»)
  3. Самой системы «в железе и бетоне»

  Обычно обеспечивается:
  - Наличием конфигурационной базы данных (CMDB – configuration management database)
  - Административными процедурами по её ведению

  Учёт включает административные процедуры по:
  - Назначению ведущего учёт (регистратора)
  - Передаче ведения учёта от регистратора регистратору
  - Делегированию полномочий по учёту в порядке распределенной учётной деятельности etc.

Централизация vs. Nihil – 1/2

- Централизация: один административный центр, который вводит
  - Обязательную идентификацию
  - Обязательный регламент учёта
  - Централизованное версионирование

  Для этого этот центр должен обладать:
  - Правом это все ввести
  - Возможностью проверить выполнение этих правил
  - ~~Силой всех расстрелять~~

Централизация vs. Nihil – 2/2

- Распределенная разработка (collaborative engineering, concurrent engineering) – каждая из участвующих в проекте организаций имеет собственные предпочтения по управлению конфигурацией (кодировки, учётные регламенты, версионирование). Проблемы:
  - Любая PLM-система поддерживает управление конфигурацией → Если в расширенной организации (extended enterprise) используется несколько разных PLM-систем → проблемы
  - Нет системы управления ЖЦ (полноценной: организация + софт); есть только PLM-система без [исполняемых] организационных решений → ПРОБЛЕМЫ

Способы УК

- Теория идентификации
- Теория учёта
- Теория версионирования

Операции УК

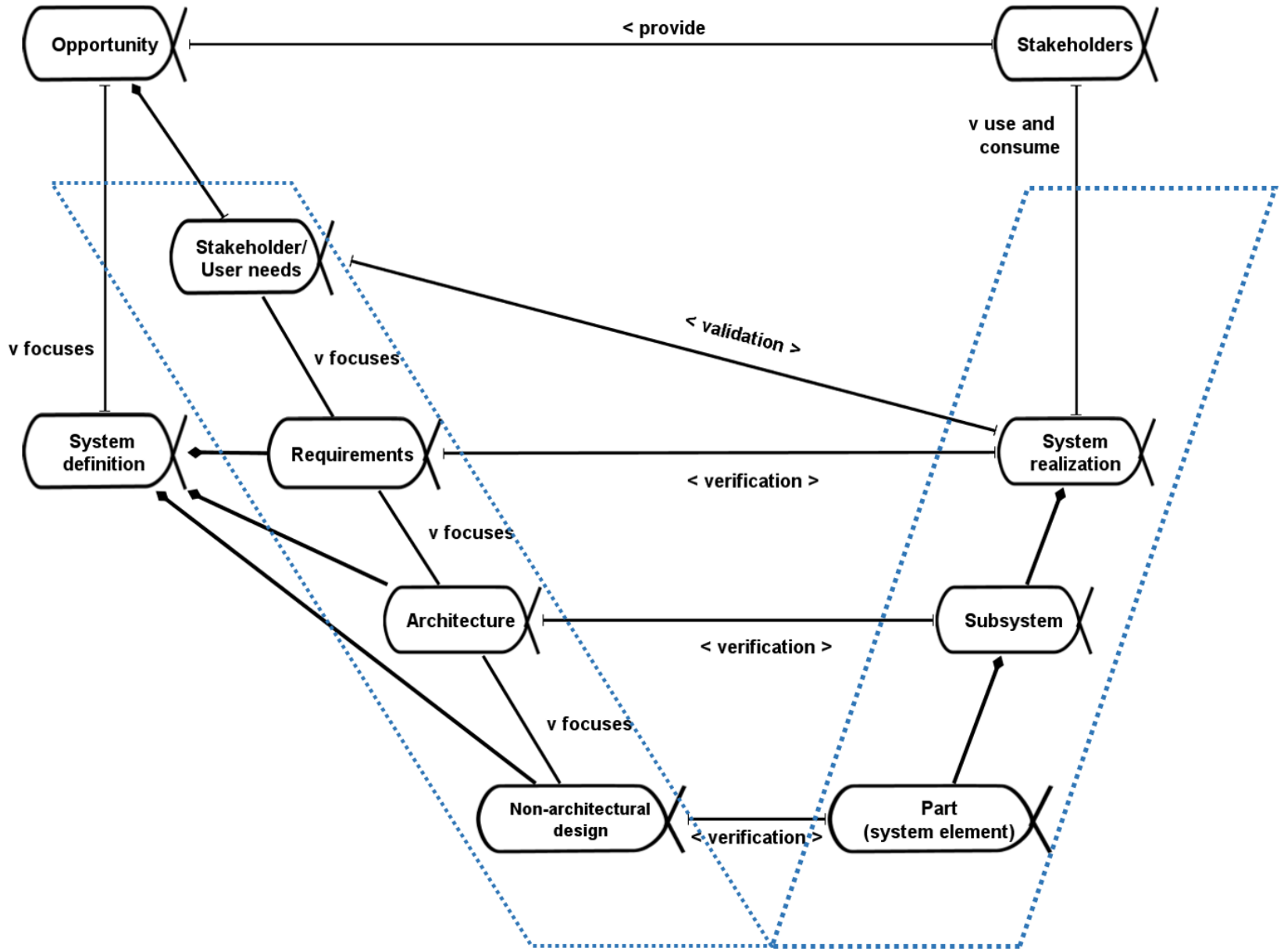
- Обучение людей (и ~~расстрелы~~!)
- Развертывания ПО в организациях:
  - Конфигурационных баз данных
  - Справочников по кодировкам
  - Систем версионирования etc.
- Оперативный контроль исполнения регламентов

Термины и определения

Инструменты

TBD

V-диаграмма определений



V-диаграмма определений