

Министерство науки и высшего образования Российской Федерации
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ**
“Национальный исследовательский университет ИТМО”

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

**ФОРМАЛИЗАЦИЯ КИБЕРФИЗИЧЕСКОГО ПОДХОДА К
ПРОЕКТИРОВАНИЮ В РАМКАХ СТАНДАРТА ESSENCE**

Автор Лалетина Екатерина Александровна
(Фамилия, Имя, Отчество)

(Подпись)

Направление подготовки (специальность) 09.03.01,
(код, наименование)
Информатика и вычислительная техника

Квалификация бакалавр
(бакалавр, магистр, инженер)

Руководитель ВКР Пенской А.В., к.т.н.
(Фамилия, И., О., ученое звание, степень)

(Подпись)

Очень не
могу выйтися от чувства
что получилось
много (до содержания)
но скажи все
не критично

Санкт-Петербург, 2020 г

Обучающийся Лалетина Екатерина Александровна

(ФИО полностью)

Группа P3402 Факультет/институт/клuster Факультет программной инженерии и компьютерной техники

Направленность (профиль), специализация Вычислительные машины, комплексы, системы и сети

Консультант (ы):

a) Болдырева Елена Александровна, ассистент
(Фамилия, И., О., ученое звание, степень) _____

(Подпись) _____
б) _____
(Фамилия, И., О., ученое звание, степень) _____
(Подпись) _____

ВКР принята “ ____ ” 20 ____ г.

Оригинальность ВКР _____ %

ВКР выполнена с оценкой _____

Дата защиты “ ____ ” 20 ____ г.

Секретарь ГЭК _____
(ФИО) _____

(подпись) _____

Листов хранения _____

Демонстрационных материалов/Чертежей хранения _____

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
"НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО"

УТВЕРЖДАЮ

Руководитель ОП

Алиев Т.И.
(Фамилия, И.О.)

_____ (подпись)

«18» «ноября» 2019 г.

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**

Обучающийся Лалетина Екатерина Александровна
(ФИО полностью)
Группа P3402 **Факультет/институт/кластер** Факультет программной инженерии и компьютерной техники

Квалификация бакалавр
(магистр, инженер, бакалавр)

Направление подготовки 09.03.01 – Информатика и вычислительная техника
(код, название направления подготовки)

Направленность (профиль) образовательной программы

Вычислительные машины, комплексы, системы и сети

Специализация –

Тема ВКР Формализация киберфизического подхода к проектированию в рамках стандарта Essence

Руководитель Пенской Александр Владимирович, Университет ИТМО, доцент, к.т.н.
(ФИО полностью, место работы, должность, ученая степень, ученое звание)

2 Срок сдачи студентом законченной работы до «25» «мая» 2020 г.

3 Техническое задание и исходные данные к работе

1. Исследовать предметную область киберфизических систем.
2. Изучить методы и средства для работы с методологиями, предоставляемые стандартом OMG Essence.
3. Формализовать предметную область киберфизических систем в рамках уровня «Kernel» стандарта OMG Essence, определив набор «Альф» и стадии их развития.
4. Апробация предложенного способа формализации на примерах существующих методологий разработки киберфизических систем.
5. Выделить инструмент оценки согласованности аспектов киберфизических систем.

4 Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов)

1. Введение.
2. Постановка цели и задач.
3. Определение структуры «Kernel» киберфизических систем с использованием стандарта OMG Essence.
4. Применение разработанного инструментария к реальным киберфизическим моделям.
5. Заключение.

5 Перечень графического материала (с указанием обязательного материала)

Презентация в формате pptx, включающая слайды:

1. Постановка цели и задач.
2. Актуальность темы.
3. Структура «Kernel».
4. Применение выделенного инструментария на практике.
5. Выводы по результатам работы.

6 Исходные материалы и пособия

1. Lee, Edward Ashford, and Sanjit A. Seshia. Introduction to Embedded Systems: a Cyber-Physical Systems Approach. MIT Press, 2017.
2. Partridge, Chris. Business Objects: Re-Engineering for Re-Use. Butterworth-Heinemann, 2005.
3. About the Essence Specification Version 1.2: Essence – Kernel and Language for Software Engineering Methods [Электронный ресурс]. – 2018. – URL: <https://www.omg.org/spec/Essence/About-Essence/>

7 Дата выдачи задания «18» «ноября» 2019 г.

Руководитель ВКР _____
(подпись)

Задание принял к исполнению _____ «18» «ноября» 2019 г.
(подпись)

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
"НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО"

АННОТАЦИЯ

ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Обучающийся Лалетина Екатерина Александровна
(ФИО)

Наименование темы ВКР: Формализация киберфизического подхода к проектированию в рамках стандарта Essence

Наименование организации, где выполнена ВКР Университет ИТМО

ХАРАКТЕРИСТИКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

1 Цель исследования Формализовать проектные процедуры для предоставления инструментов оценки состояния проекта

2 Задачи, решаемые в ВКР Исследование предметной области киберфизических систем; изучение методов и средств для работы с методологиями, предоставляемые стандартом Essence; формализация предметной области КФС в рамках уровня «Kernel» стандарта OMG Essence; апробация предложенного способа формализации на примерах существующих методологий разработки КФС; создание инструмента оценки согласованности аспектов КФС.

3 Число источников, использованных при составлении обзора 16

4 Полное число источников, использованных в работе 22

5 В том числе источников по годам

Отечественных			Иностранных		
Последние 5 лет	От 5 до 10 лет	Более 10 лет	Последние 5 лет	От 5 до 10 лет	Более 10 лет
2	3	-	3	4	10

6 Использование информационных ресурсов Internet да, 3

(Да, нет, число ссылок в списке литературы)

7 Использование современных пакетов компьютерных программ и технологий (Указать, какие именно, и в каком разделе работы)

Пакеты компьютерных программ и технологий	Раздел работы

Я что бы не читать не учил а изображал

8 Краткая характеристика полученных результатов Создание колоды карт Essence на русском языке, задача которой упростить процесс отслеживания статуса проекта и планирования дальнейших действий командой проекта.

9 Полученные гранты, при выполнении работы _____ нет
(Название гранта)

10 Наличие публикаций и выступлений на конференциях по теме выпускной работы нет
(Да,нет)

a) 1 _____
(Библиографическое описание публикаций)

2 _____

3 _____

б) 1 _____
(Библиографическое описание выступлений на конференциях)

2 _____

3 _____

Обучающийся Лалетина Екатерина Александровна _____
(ФИО) _____
(подпись)

Руководитель ВКР Пенской Александр Владимирович _____
(ФИО) _____
(подпись)

“ _____ ” 20 ____ г.

СОДЕРЖАНИЕ

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ.....	5
ВВЕДЕНИЕ	6
1 ПОДХОДЫ К ПРОЕКТИРОВАНИЮ КИБЕРФИЗИЧЕСКИХ СИСТЕМ.....	9
1.1 Киберфизический системный подход	9
1.2 Проектировани ^и на основе платформы.....	11
1.3 Выводы	13
2 ИНСТРУМЕНТЫ ФОРМАЛИЗАЦИИ МЕТОДОВ	14
2.1 OPEN Process Framework	15
2.2 OMG Essence	16
2.3 Системо-мыследеятельностная методология	17
2.4 Стандарты ЕСКД и ЕСПД	19
2.4 Результаты сравнения <i>В51 Воджин</i>	20
3 СТРУКТУРА ЯДРА В СТАНДАРТЕ ESSENCE	22
3.1 Создание Альф, отображающих особенности предметной области киберфизических систем.....	25
3.2 Детализация Альф и саб-Альф	26
4 АПРОБАЦИЯ ПРЕДЛОЖЕННОГО СПОСОБА ФОРМАЛИЗАЦИИ	30
5 КАРТЫ ESSENCE	33
ЗАКЛЮЧЕНИЕ.....	37
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	40
ПРИЛОЖЕНИЕ 1 (колода карт Essence на русском языке)	43

Можб? *запомнишь*
запомнишь *стар* *есенс*
к *каф*?

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

ВС – вычислительная система.

ВсС – встроенная ВС, embedded system.

ЕСКД – единая система конструкторской документации.

ЕСПД – единая система программной документации.

КФС – киберфизическая система.

ММК – Московский методологический кружок.

ПО – программное обеспечение.

СМД-методология – системо-мыследеятельностная методология.

SME – situational method engineering, ситуационной инженерии методов.

OPF – OPEN Process Framework.

OMG – object management group.

PBD – Platform Based Design.

ВВЕДЕНИЕ

Понятие киберфизических систем появилось относительно недавно – около десяти лет назад. КФС можно отнести к информационным-управляющим системам. Они представляют собой интеграцию физических, вычислительных и сетевых процессов. В основе КФС используется множество современных технологических тенденций: облачные вычисления, Интернет вещей, автономные роботы, большие данные и их аналитика.

На рисунке 1 можно увидеть многообразие элементов и связей между ними, входящих в сферу киберфизических систем.

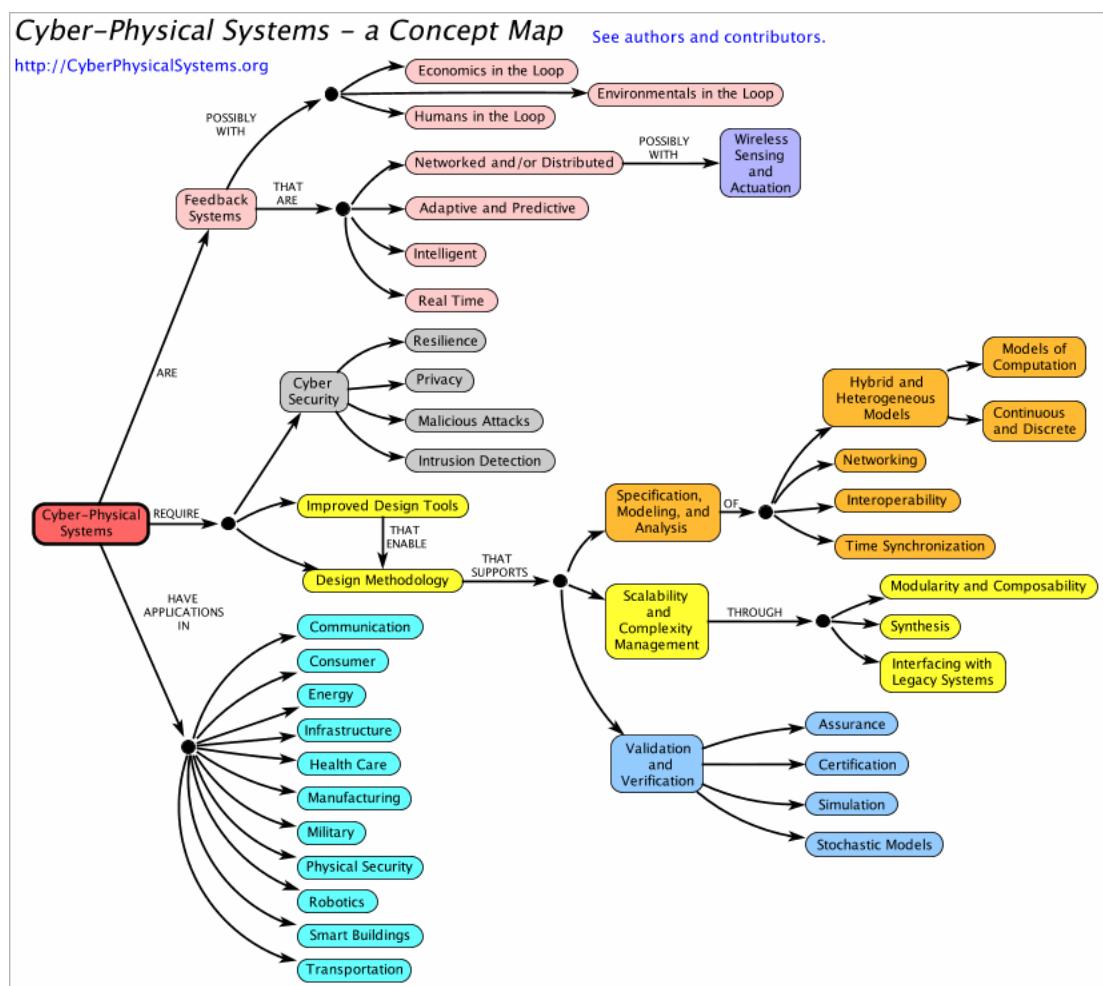


Рисунок 1 – Диаграмма связей КФС [20]

КФС могут применяться в совершенно разных областях, начиная автомобилестроением и заканчивая медициной. Их широкое внедрение в сферу производства, например, сможет решить проблему человеческого

фактора, поспособствует повышению качества товаров и снижению цен на них, так как затраты на их производство снижаются [8, 9].

Экономический и социальный потенциал киберфизических систем огромен, однако, на данный момент не может быть реализован в полной мере. Существенной проблемой в эффективном создании КФС является отсутствие единого подхода к их проектированию.

Подобная проблема также характерна и для встроенных систем (embedded system), которые являются более узкой и имеющей свои особенности в проектировании областью в сфере КФС. А.Е. Платуновым в его работе [3, с. 32] были емко перечислены основные причины:

1. сложность функциональной и пространственной декомпозиции задачи;
2. специфический «портрет специалиста», необходимого для работы в области программирования ВсС;
3. отставание темпов роста эффективности проектирования от потребностей отрасли;
4. низкий коэффициент повторного использования результатов проектирования.

Для решения проблемы с точки зрения вышеуказанных причин было предложено и рассмотрено множество различных концепций [4, 5, 6, 7].

В данной работе рассматривается еще одна немаловажная причина существующей проблемы – в разработке КФС в силу их специфики принимают участие специалисты совершенно разных профилей, поэтому отсутствие эффективной коммуникации между ними в процессе создания и проектирования КФС и инструмента для оценки результатов их работы будет являться центральным вопросом моего исследования.

Цель данной работы – формализовать проектные процедуры для предоставления инструментов оценки состояния проекта. Результатом должна стать наглядная и удобная в использовании колода карт Essence. Карты отражают ключевые показатели эффективности проекта, они так же

полностью независимы от метода, жизненного цикла процесса и могут применяться к любому проекту ~~по~~ разработке КФС.

В первой части работы мной была исследована предметная область киберфизических систем. После изучения и сравнения нескольких подходов к проектированию, а также ознакомления с несколькими инструментами формализации методов, я поставила следующие задачи:

- 1) формализовать предметную область киберфизических систем в рамках уровня «Kernel» стандарта Essence;
- 2) выделить набор Альф и стадии их развития для модифицированного Ядра;
- 3) выделить инструмент оценки согласованности аспектов киберфизических систем;
- 4) апробировать предложенный способ формализации на примерах рассмотренных методологий разработки киберфизических систем.

1 ПОДХОДЫ К ПРОЕКТИРОВАНИЮ КИБЕРФИЗИЧЕСКИХ СИСТЕМ

Несмотря на активное развитие сферы киберфизических систем в последние годы новых исследований или работ в данной области публиковалось относительно немного, а те, что есть – в основном относятся к встроенным системам.

Кроме этого, большинство работ по встроенным системам фокусируется на адаптации уже созданных технологий – языков программирования, операционных систем, сетей и так далее [10, 11]. Эти технологии необходимы (на сегодняшний день) для создания и обеспечения работы встроенных систем, но они не формируют интеллектуальное ядро этих дисциплин, которое должно состоять из моделей и абстракций. Во многом подход к проектированию сейчас определяется конкретными целями, которые должны быть достигнуты в результате работы и спецификой создаваемой системы.

Моя задача заключается в изучении тех методологий и концепций, которые основываются на моделях и их связи с реализацией системы, при этом модель является абстракцией, то есть в ней отсутствуют только несущественные детали. В таком случае мы можем быть уверены в физической реализации систем при оперировании такими моделями.

1.1 Киберфизический системный подход

Одним из известных авторов трудов по встроенным и киберфизическим системам является Эдвард Ли. В своей работе он предлагает сформировать базис для проектирования встроенных систем [2].

Главная идея заключается в разделении процесса разработки ВсС на три части: моделирование, проектирование и анализ (на рисунке 2):.

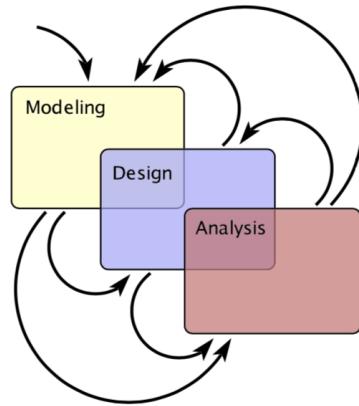


Рисунок 2 – Создание встроенных систем требует итеративного процесса моделирования, дизайна и анализа [2, с.8]

Моделирование интерпретируется как процесс более глубокого понимания системы посредством имитации. Модель, в понимании авторов, это описание определенных аспектов системы, которые призваны дать представление о свойствах системы. Модели КФС, как правило, должны представлять как динамические, так и статические свойства (те, которые не изменяются во время работы системы).

Авторы анализируют и объединяют в киберфизическое целое именно те методы моделирования, которые широко используются инженерами и хорошо поняты.

Дизайн (или проектирование) – это поэтапное создание различных частей системы. Он определяет, как система выполняет свои функции. Авторы акцентируют свое внимание на разработке встроенных систем с учетом той роли, которую они играют в сфере киберфизических систем. В работе рассматривается проектирование некоторых компонентов, которые могут стать частью создаваемой системы (датчики, встроенные процессоры, архитектура памяти).

Анализ – это процесс определения реальных характеристик системы и их взаимодействия. Он показывает, почему система делает то, что она делает (или не в состоянии делать то, что, по словам модели, должна делать).

Каждая система должна быть разработана с учетом определенных требований. Для встроенных систем, которые часто используются в сферах повседневной жизни, требующей особого уровня безопасности (медицина, приборостроение), особенно важно быть уверенным в том, что система соответствует всем необходимым требованиям (спецификациям). Кроме этого необходимо точно формулировать требования. Для этого авторы рассматривают несколько методов формальной верификации, в том числе и проверку моделей.

В целом данный подход направлен на создание абстракций, которые можно было доработать и использовать при разработке разных киберфизических систем. Авторы сосредоточили свое внимание на взаимодействии программного и аппаратного обеспечения с физической средой, в которой выполняет свои функции целевая система.

1.2 Проектирование на основе платформы

Методология проектирования на основе платформы (Platform Based Design) появилась несколько лет назад и в работах Альберто Санджiovанинни-Винчентелли была адаптирована и направлена именно на проектирование встроенных систем [7, 12].

Данный подход, как и предыдущий, был призван решить проблему повторного использования аппаратных и программных решений. Кроме этого, авторы полагают, что с его помощью можно уменьшить затраты на производство и повысить производительность.

В основе концепции лежит понятие платформы. Платформа может быть аппаратной и программной, их объединение является системной платформой.

Аппаратная платформа – это набор архитектур встроенных систем, решающих определенные задачи. Как правило, аппаратная платформа обладает большими свойствами и функциональностью, чем требуется от создаваемой системы.

Программная платформа – это то, что отвечает за взаимодействие аппаратуры и программы, но при этом эта платформа должна быть независима и как можно более унифицирована для работы с большим количеством аппаратных решений.

Системная платформа состоит из аппаратной и программной платформы, которые представляют собой абстракции, и чем выше уровень абстракции, тем больше выбора в конкретной реализации будет у разработчика. В процесс же разработки к системы предъявляются определенные требования и в итоге остается единственное верное решение – определенная аппаратуры и программные средства.

На рисунке 3 показано как системная платформа разделяет процесс создания приложения (верхний треугольник) от процесса создания архитектуры (нижний треугольник). То есть архитектура должна описывать функциональность приложения, но не может зависеть от реализации.

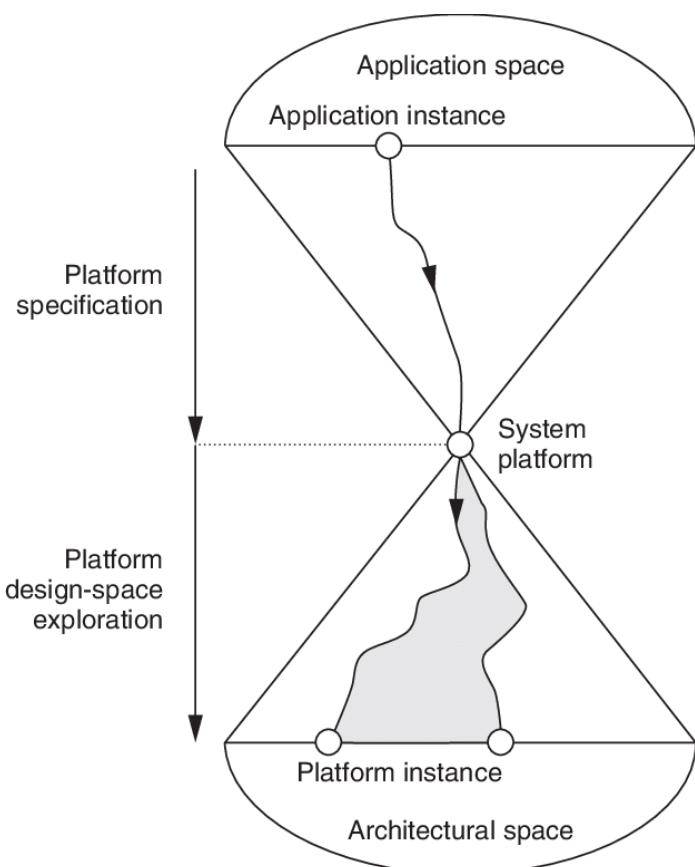


Рисунок 3 – Процесс проектирования [13, с.29]

Фундаментальные
и технические
аспекты
первой
членской

В целом у концепции PBD несколько достоинств – приоритет архитектурного этапа проектирования, ориентация на математическую формализацию процесса разработки и анализ моделей системного уровня – «моделей вычисления» [3, с. 52].

1.3 Выводы

Альберто Санджиованни-Винченцелли предлагает интересные и верные тезисы, которые в прочем в некотором виде использовались разработчиками и до публикаций первых его работ. Однако PBD представляет из себя скорее очень общий подход к проектированию, в то время как концепция, которую рассматривает Эдвард Ли, все же более ориентирована на киберфизические системы.

Главная проблема проектирования на основе платформы в том, что оно показывает как проектировать систему, но не отвечают на вопрос, что именно для этого нужно сделать. Именно поэтому в работе при формализации проектных процедур будет использоваться итерационная модель Эдварда Ли (моделирование-дизайн-проектирование).

2 ИНСТРУМЕНТЫ ФОРМАЛИЗАЦИИ МЕТОДОВ

К проектированию любой системы (software-intensive system), в которой разработка и/или интеграция программного обеспечения являются главной проблемой, в том числе встроенной системы или киберфизической системы, можно подойти с нескольких сторон, используя разные методологии.

Дисциплина, получившая название инженерия методов, начала формироваться в 80 года прошлого века. Она должны была предоставить эффективные средства для создания и поддержки методов управления жизненным циклом разработки программных средств [21]. В результате развития инженерии методов появились различные наборы методов для выполнения определенного вида работ.

Недостатки традиционной инженерии методов привели к появлению ситуационной инженерии методов.

Ситуационная инженерия методов – альтернативный подход к разработке программного обеспечения, который включает в себя все аспекты создания, использования и адаптации методов разработки программного обеспечения для конкретных целей в отличие от приобретения «готовых методов» (off-the-shelf) и использования их без всяких изменений [1].

Суть SME заключается в том, что не существует эффективного способа работы, кроме ситуационного, то есть метод должен быть разработан для определенной ситуации и не может быть использован для другой, как это предполагается в традиционной инженерии методов. Однако, определенный базис, который можно было бы применять к разным ситуациям в SME есть. Для этого все методы должны быть разбиты на условные компоненты, а сами компоненты – хранятся в специальном каталоге методов. Метод, подходящий для конкретной ситуации собирается уже из этих самых компонентов.

Таким образом, утверждается, что нельзя предоставить универсальный метод, но можно дать конструктор методов с достаточным количеством

компонентов, который позволит собрать необходимый метод под определенные потребности.

Далее в работе рассматриваются и сравниваются инструменты формализации методов, которые предлагает SME.

2.1 OPEN Process Framework

OPF наряду с ISO 24744 и OMG SPEM 2.0 является стандартом первого поколения SME. Он больше ориентирован на методологов, которым необходимо описывать методы работы и систематизировать их.

OPEN Process Framework является одним из самых первых стандартов и представляет из себя свободный набор компонентов для создания процессов жизненного цикла на уровне проекта.

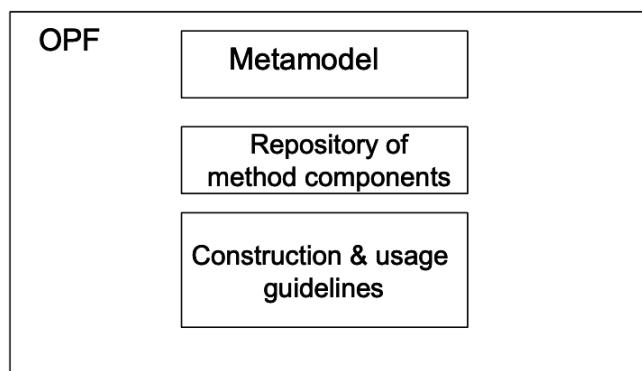


Рисунок 4 – Главные элементы OPF [16, с.2]

На рисунка 4 можно увидеть, что OPF состоит из трех частей: метамодели, репозитория компонентов методов и рекомендаций по использованию, которые описывают ограничения и способы повторного использования компонентов [16, с.3].

Метамодель определяет основные виды компонентов методов и их взаимосвязь. Она больше фокусируется на уровне метода, чем на уровне конкретного проекта.

Репозиторий компонентов метода представляет из себя описаний этих самых компонентов. Организован он в виде иерархии веб-страниц, которые также включают руководства по созданию методов. Стандартный подход при использовании репозитория OPF включает в себя:

1. проверка метода, документирование и утверждение;
2. обучение команд, в соответствии с выбранными инструментами;
3. сам метод применяется столько раз, сколько необходимо для достижения результата.

Руководств, необходимых для инженера методов, существует несколько. Они описывают как конструировать и адаптировать метод и расширять метамодель.

Кроме этого, в руководствах также указаны правила связывания, которые обеспечивают корректную взаимосвязь между компонентами методов.

Руководство рассказывает как создавать компоненты или выбирать подходящие из репозитория, чтобы получить новый метод. Руководство помогает выбрать необходимые продукты, исполнителей и акты деятельности, а также способствует распределению актов деятельности в процесс работы.

2.2 OMG Essence

Стандарт OMG Essence можно назвать логичным продолжением OPF, так как он входит во второе поколение SME. В отличие от OPF этот стандарт был больше ориентирован на инженеров (то есть пользователей тех описаний, которые создают методологии).

В основе OMG Essence лежит понятие Языка (the Language), который представляет из себя минимальный набор понятий с помощью которого описываются методы программной инженерии.

Кроме этого в стандарте есть Ядро (the Kernel), которое определяет базовые концепции и предметную область. Оно включает в себя основные элементы (Альфы), которые отражают все этапы разработки [10].

Ядро помогает различным специалистам (архитекторам, разработчикам, например) сравнивать методы и принимать решение об их дальнейших действиях, отслеживать прогресс и состояние разработки. Ядро описывается с использованием Языка, который в свою очередь

поддерживает объединение практик для формирования новой практики и объединение этих практик в методы.

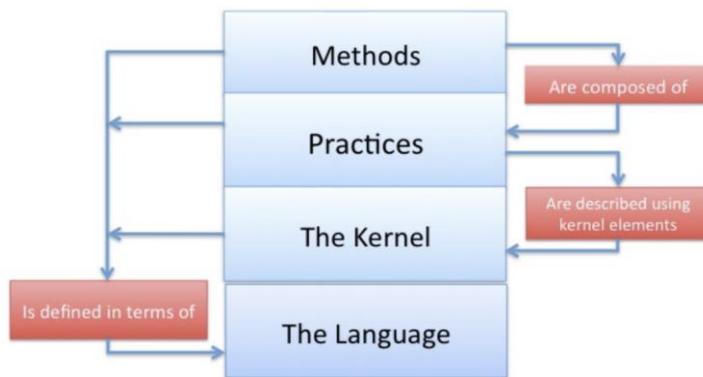


Рисунок 4 –Архитектура метода [17, с.10]

То есть методы – это совокупность практик. Они не просто описывают инструкции для разработчиков, они отражают их деятельность. Таким образом, здесь меняется традиционное понимание метода. Метод – это не просто описание того, что нужно сделать, это описание того, что на самом деле уже сделано.

Практика – это повторяющийся подход к тому, чтобы достичь какой-либо определенной цели. Практика обеспечивает систематический и проверяемый способ решения определенного аспекта выполняемой работы. Практика может быть частью многих методов.

Отсюда можно определить основные идеи OMG Essence:

1. практики можно использовать множество раз в разных областях и с разными технологиями;
 2. метод остается максимально приближенным к практике, так как он развивается и адаптируется к их конкретному контексту и задачам;
 3. нужно отделять результат от документации и сущность от деталей
- [17, с 11].

2.3 Системо-мыследеятельностная методология

Возвращаясь к предпосылкам цели исследования, можно заметить, что на вопрос об эффективной коммуникации специалистов при разработке и проектированию КФС в разной степени отвечают и OPF, и Essence, так как

оба этих стандарта показывают определенные пути организации процессов работы.

Однако, кроме них мне хотелось бы в рамках данной работы упомянуть также своеобразное философское-методологическое направление, сформированное на основе идей, разрабатываемых с 1952 в рамках Московского методологического кружка [14].

Идейным вдохновителем и организатором ММК на протяжении многих лет был Г.П. Щедровицкий. Именно им была выработана особая концепция управленческой деятельности в жизни общества. Для того, чтобы отделить этот подход от концепций других школ и движений, был даже введен термин – ОРУ, который является аббревиатурой от трех основных составляющих, определяющих специфические черты этой деятельности – организация, руководство, управление.

С одной стороны концепция ОРУ показывает оригинальный подход к представлению и трактовке человеческой деятельности вообще, деятельности профессиональной в частности и оргуправленческой в особенности – системо-мыследеятельностный подход.

В своих лекциях Г.П. Щедровицкий высказывает одну из основных мыслей своей концепции: «Подобно тому, как мы представляем мир в виде построек из атомов, молекул, точно так же мы считаем, что мир деятельности состоит из элементарных актов, которые организуются в сложные цепи, или «молекулы», деятельности, за счет связей кооперации, коммуникации, за счет введения определенных технологий и т. д.» [15, с. 52].

На рисунке 5 отражена схема акта деятельности, который, как считает Г.П. Щедровицкий и является элементарной единицей деятельности.

Человек обладает определенными способностями и интериоризованными средствами (например, языком, на котором мы говорим или языком алгебры). Кроме этого у человека также есть табло сознания, которое всегда активно, а не пассивно.

Человек с помощью действий (или операций) и орудий (ЭВМ) преобразует или превращает исходный материал в продукт.

Есть еще цели и знания, которые приходят извне. В целом это и есть структура акта деятельности, деятельности преобразования, которую мы совершаем каждый раз, когда разрабатываем или проектируем систему.

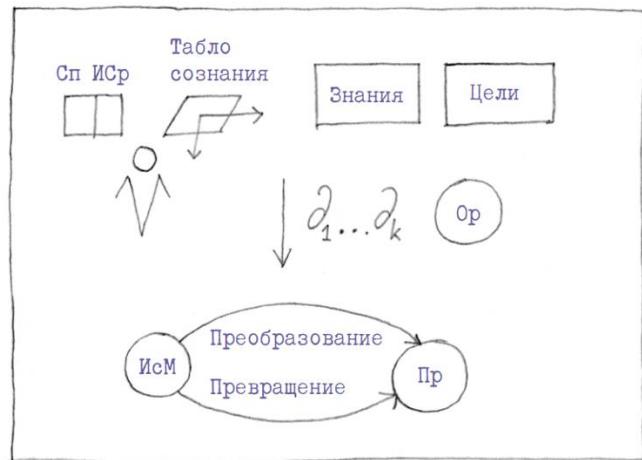


Рисунок 5 – Состав и структура акта деятельности [15, с. 52]

По большей части в лекциях Г.П. Щедровицкий рассуждает именно о том, на что и каким образом может повлиять в этой схеме воздействие организации, руководства и управления, и что в итоге приведет к эффективному и успешному результату.

2.4 Стандарты ЕСКД и ЕСПД

Противопоставить абстрактной и философской СМД-методологии можно региональные стандарты (ГОСТ), которые в России применяются в рамках стандартизации.

Несмотря на то, что ГОСТы с 2003 года не являются обязательными, они все же нередко используются на территории страны, например, в учебных организациях или на государственных предприятиях.

В контексте данной работы мне хотелось бы рассмотреть стандарты единой системы конструкторской документации и единой системы программной документации.

ЕСКД в отличие от ЕСПД гораздо новее, в 2013 был создан новый стандарт взамен тому, что появился в 1993 году, соответственно, в нем учитываются изменения, которые претерпела сфера машиностроения и приборостроения за последние десятилетие.

В ЕСКД устанавливаются единые правила и требования к оформлению конструкторской документации, которые обеспечивают:

1. взаимообмен конструкторской документацией без ее переоформления;
2. автоматизацию обработки КД и содержащейся в них информации;
3. расширение унификации и стандартизации при проектировании изделий и разработке конструкторской документации;
4. сокращение сроков и снижение трудоемкости подготовки производства;
5. оперативную подготовку документации для быстрой переналадки действующего производства;
6. создание и ведение единой информационной базы;
7. гармонизацию стандартов ЕСКД с международными стандартами (ИСО, МЭК) в области конструкторской документации [18].

ЕСПД содержит набор стандартов, которые применяются при разработке программного обеспечения. Он описывает требование к схемам и алгоритмам программ, стадиям разработки и к программным документам [19].

В отличие от Essence, OPF и СМД-методологии, которые отражают и человеческую сторону процесса разработки, так как человек является неотъемлемой его частью, ЕСКД и ЕСПД больше сконцентрированы на описание общих требований к технической части.

2.4 Результаты сравнения

Выводы

В предыдущих разделах были рассмотрены спецификации OPEN Process Framework, OMG Essence, системо-мыследеяательностная методология и стандарты ЕСКД и ЕСПД.

Возвращаясь к реализациям SME, как более структурированным и подходящим для их дальнейшей модернизации под проектирование КФС, мне остается выбрать подходящий стандарт для дальнейшей работы.

В целом главная идея этих двух стандартов схожа – унифицировать общие для разработки элементы и создать механизм, который будет поддерживать разработку различных программных средств, используя общую для этого основу. Однако, так как Essence более ориентирован на инженеров и включает в себя хорошо проработанный набор базовых типизированных понятий (Ядро), отталкиваясь от которого будет легче разработать специализированный набор для КФС, я остановила свой выбор на нем.

3 СТРУКТУРА ЯДРА В СТАНДАРТЕ ESSENCE

Ядро – это упрощенный набор определений, которые отражают суть эффективной, масштабируемой разработки системы. Его основная задача – предоставить общую основу для определения методов и практик для дальнейшего их использования под конкретные потребности команды проекта.

Ядро состоит из трех частей:

1. *Пространства деятельности* – представление основных задач.

Здесь также описываются основные проблемы, с которыми сталкивается команда при разработке и поддержке программного обеспечения и то, что команда может сделать для их решения.

2. *Альфы* – представление основных вещей, с которыми нужно работать. Альфы описывают то, чем команда будет управлять, что она будет производить и использовать в процессе разработки и поддержки ПО.

3. *Компетенции* – представление ключевых возможностей, необходимых для выполнения работы по разработке ПО.

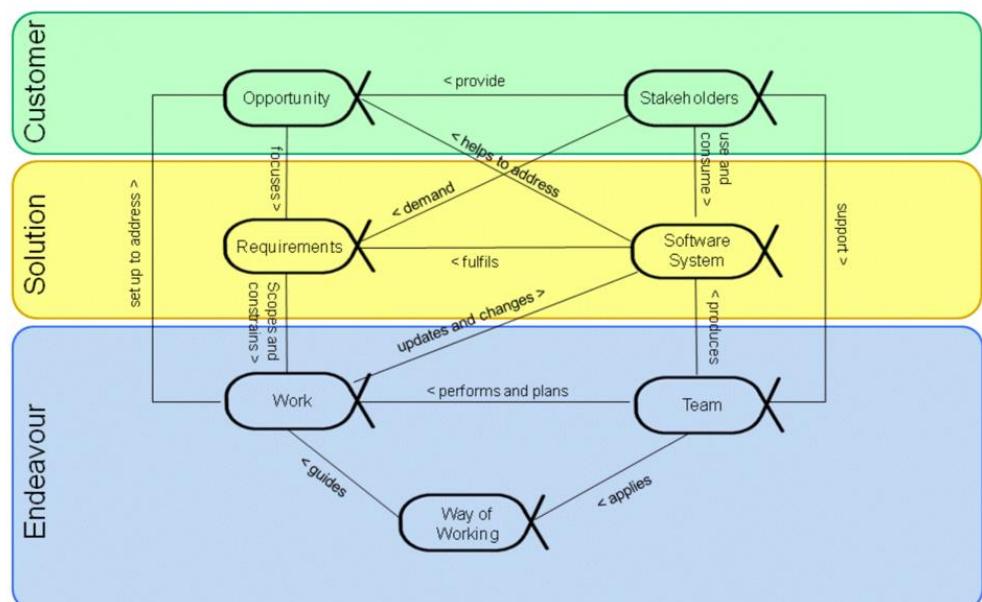


Рисунок 6 – Альфы Ядра [17, с.17]

*Ключевые
свойства*

Далее в работе подробно рассматривается только одна часть Ядра – Альфы.

На рисунке 6 видно, что Ядро разделено на три проблемные области:

1. Заказчика (Customer). В этой области находится все, что связано с использованием ПО, когда оно будет разработано.
2. Решения (Solution). В этой области содержится то, что имеет отношение к спецификации и разработке программного обеспечения.
3. Стремления (Endeavor). Эта область содержит все, что относится к команде и к ее работе.

Каждая область содержит связанные с ней Альфы. Альфы в Ядре выполняются несколько функций. Они отражают ключевые концепции, связанные с разработкой ПО, обеспечивают возможность отслеживания и оценки прогресса разработки ПО, обеспечивают общую основу для определения методов и практик разработки ПО.

У каждой Альфы есть набор состояний, с помощью которых можно провести оценку прогресса разработки проекта. У каждого состояния есть контрольный список (checklist). Через состояния можно итерироваться необходимое количество раз, то есть можно возвращаться к предыдущему состоянию на разных этапах, если, например, не все пункты из контрольного списка были выполнены.

В области Заказчика находятся:

1. Заинтересованные стороны (Stakeholders) – группа людей или организация, которые влияют или на которых влияет разрабатываемое ПО. Заинтересованные стороны предоставляют возможность и являются источником требований к разрабатываемому продукту. Они участвуют в разработке программного обеспечения – поддерживают команду, контролируют, что разрабатываемая система будет сделана надлежащим образом.

2. Возможность (Opportunity) – совокупность обстоятельств, которые делают целесообразным разработку или изменение системы программного обеспечения. Возможность ясно формулирует причину создания нового ПО. Она представляет понимание потребностей заинтересованных сторон командой, помогает формировать требования к новой системе путем обоснования ее развития.

В области Решения команда должна установить общее понимание требований, а также внедрить, протестировать и поддержать ПО, которое она создает:

1. Требования (Requirements) – это то, что должно делать ПО, чтобы удовлетворять интересы заинтересованных сторон. Важно определить, что требуется от системы и обозначить это, как для заинтересованных сторон, так и для команды разработки, чтобы использовать их в дальнейшем для разработки и тестирования ПО.
2. Система программного обеспечения (Software System) – система, состоящая из программного и аппаратного обеспечения.

В области Стремления должна быть сформирована команда и способ ее работы.

1. Команда (Team) – группа людей, которая активно вовлечена в разработку, управление и поддержку системы программного обеспечения.
2. Способы работы (Way-of-Working) – специализированный набор методов и инструментов, используемых командой для выполнения своей работы.
3. Работа (Work) – деятельность, включающая умственные и физические усилия, предпринимаемые для достижения результата.

3.1 Создание Альф, отображающих особенности предметной области киберфизических систем

Рассматривая вместо системы программного обеспечения киберфизические системы, можно модернизировать ядро таким образом, чтобы оно более точно отражало специфику разработки КФС.

Вместо Альфы Системы программного обеспечения будет использоваться Киберфизическая система.

Киберфизическая система – система, взаимодействующих вычислительным компонентов, которые находятся в интенсивной связи с окружающим физическим миром и процессами, протекающими в нем.

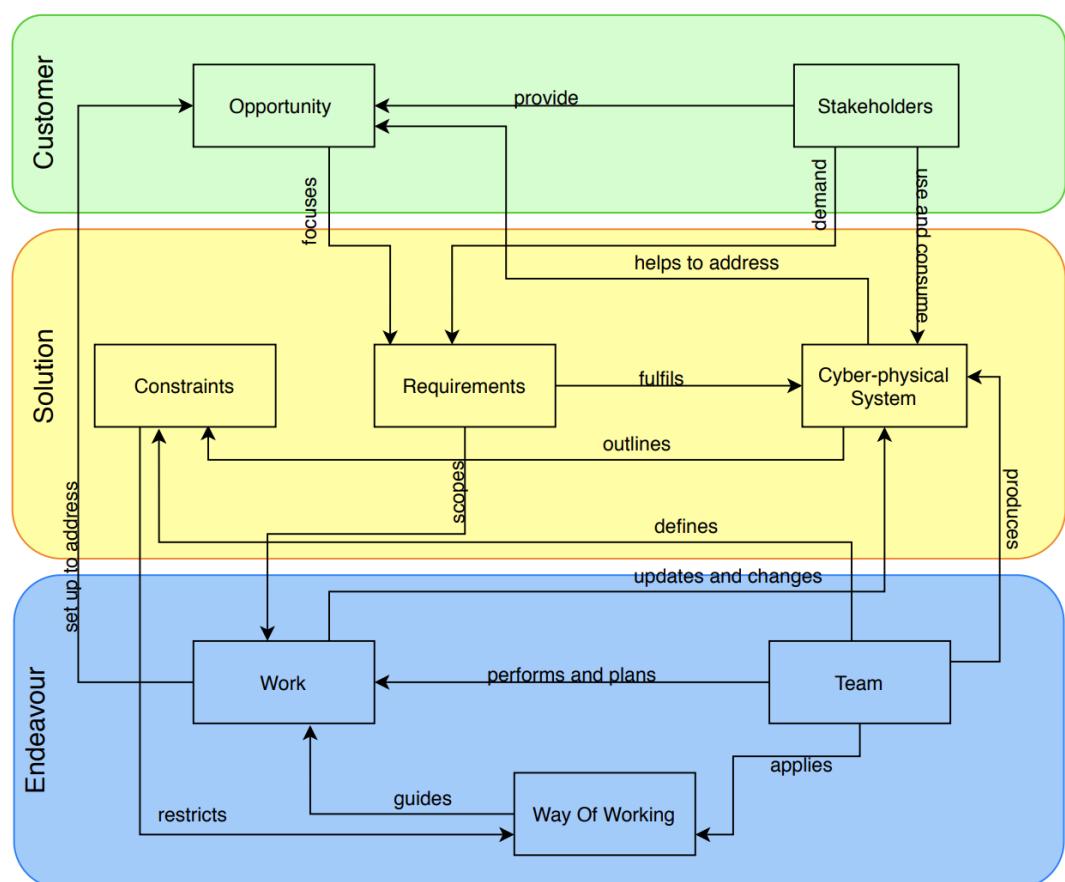


Рисунок 7 –Альфы Ядра для киберфизических систем

На рисунке 7 видно, что область Заказчика осталась точно такой же, изменения затронули только две другие области.

В области Решения помимо Требований (Requirements) появились Ограничения (Constraints). В данном контексте под Требованиями

понимаются функции системы и ее компонентов. Требования исходят от Заинтересованных сторон, и именно Требования определяют функциональность Киберфизической системы. В то время как Ограничения касаются технологий, методов, ресурсов, необходимых для создания Киберфизической системы, поэтому Киберфизическая система, к которой предъявляются определенные Требования, обрисовывает некоторые Ограничения, но последнее слово остается за Командой. Именно она формирует окончательный вариант Ограничений.

Ограничения так же влияют и корректируют Способ Работы. Применяя определенный Способ Работы, Команда делает Работу.

3.2 Детализация Альф и саб-Альф

Понятие «Альфа» на самом деле является акронимом и расшифровывается как (Alpha – Abstract Level Progress Health Attribute) атрибут состояния прогресса на абстрактном уровне.

Альфы – это те объекты, развитие которых мы хотим направлять и контролировать. Как уже говорилось выше, основные этапы разработки системы можно выразить в терминах набора состояний Альф. Таким образом переход от одного состояния в другое означает прогресс в достижении целей разработки.

На рисунках 14 и 15 можно увидеть описание новых Альф – Киберфизической системы и Ограничений – и соответствующий им набор состояний.



Киберфизическая система

Архитектура выбрана

Система, взаимодействующих вычислительных компонентов, которые находятся в интенсивной связи с физическим миром и процессами, протекающими в нем

Наглядна

Используема

Готова

В эксплуатации

Не поддерживается

- Киберфизическяя система должна соответствовать требованиям
- Киберфизическяя система определяет ограничения, характерные для ее области
- Киберфизическяя система является расширяемой, тестируемой и обслуживаемой

Рисунок 14 – Характеристики Киберфизической системы



Ограничения

Представлены

Определяют нефункциональные аспекты киберфизической системы, такие как технологии, ресурсы и методы, которые будут использоваться при разработке

Определены

Используются

- Ограничения должны отображать специфику киберфизической системы
- Ограничения помогают скорректировать способ работы
- Ограничения должны определяться командой
- Ограничения влияют на выбор архитектуры системы

Рисунок 15 – Характеристики Ограничений

Альфа также может содержать в себе коллекцию других Альф. Вместе эти саб-Альфы влияют на состояние супер-Альфы, однако явной связи между состояниями саб-Альф и супер-Альфы не существует.

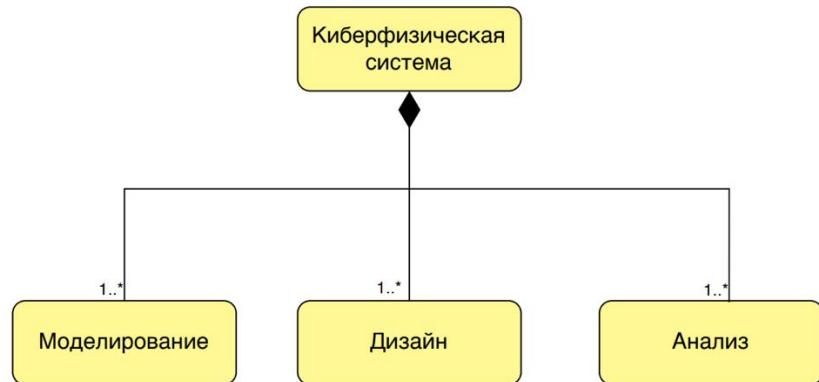


Рисунок 16 – Саб-Альфы Киберфизической системы

На рисунке 16 можно увидеть, что у Альфы Киберфизическая Система есть три саб-Альфы: Моделирование, Дизайн и Анализ.

На рисунках 17, 18 и 19 можно увидеть описание и список всех возможных состояний для трех саб-Альф.



Рисунок 17 – Характеристика Моделирования



Дизайн

2020.05

Упорядочен

Поэтапное создание частей кибер-физической системы. Он определяет как система выполняет свои функции

Уточнен

- Дизайн должен помогать системе достичь ключевых функциональных возможностей
- Дизайн должен делать систему пригодной для использования и наделять ее желаемыми характеристиками
- Дизайн должен предлагать хотя бы одно архитектурное решения для системы

Имплементирован

Рисунок 18 – Характеристика Дизайна



Анализ

2020.05

Специфицирует

Процесс определения реальных характеристик системы и их взаимодействия.

Описывает

- Анализ должен помогать понять Ограничения кибер-физической системы
- Анализ должен создавать спецификации для системы на основе Требований
- Анализ должен показывать почему система делает то, что делает (или не в состоянии делать то, что по словам модели, должна)

Исследует

Проверяет

Рисунок 19 – Характеристика Анализа

4 АПРОБАЦИЯ ПРЕДЛОЖЕННОГО СПОСОБА ФОРМАЛИЗАЦИИ

Ядро Essence позволяет с помощью практик определять жизненные циклы, упорядочивая ряд шаблонов для их определенной фазы или этапа. В стандарте приведены иллюстрации нескольких типичных жизненных циклов разработки программного обеспечения. Например, жизненный цикл каскадной модели (The Waterfall Lifecycle).

Создание жизненного цикла включает в себя не только упорядочивание состояний альф, но и добавление элементов в их контрольные списки и формирование различных рекомендаций по планированию и анализу.

Все жизненные циклы в стандарте проиллюстрированы с помощью данного шаблона (рисунок 23).

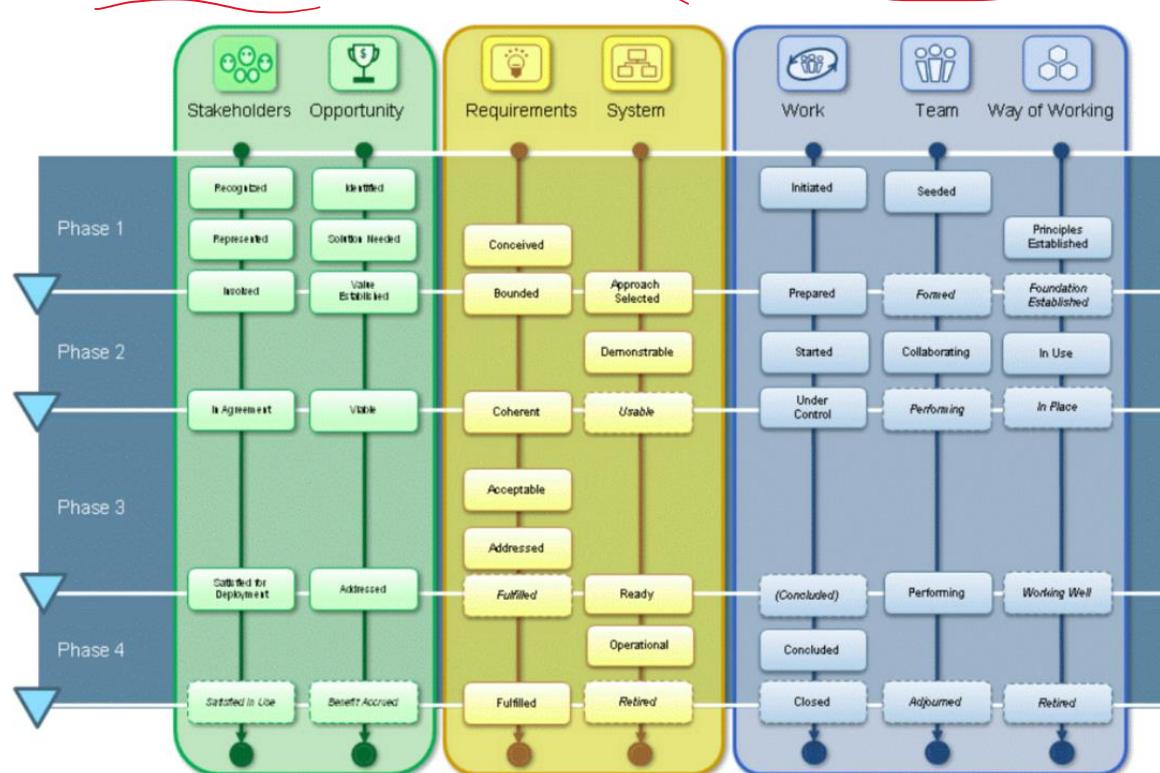


Рисунок 23 – Общий шаблон жизненного цикла [23]

Каждая Альфа и ее состояния отображены в вертикальных столбцах, соответствующих определенной области (Заказчика, Решения, Стремления). Состояния указываются сверху вниз, начиная созданием Альфы и заканчивая ее разрушением. Фазы цикла представляют из себя прямоугольные

пересекающие перпендикулярно области прямоугольники. Переход от одной фазы к другой отображен перевернутыми треугольниками. Фазы содержат в себе состояния Альф, которые должны быть достигнуты для успешного прохождения фазы. Там, где достижение состояния для перехода к следующей фазе носит рекомендательный характер, само состояние отображено пунктирной линией и курсивом.

Данный шаблон также можно использовать для описания применения уже существующей методологии проектирования.

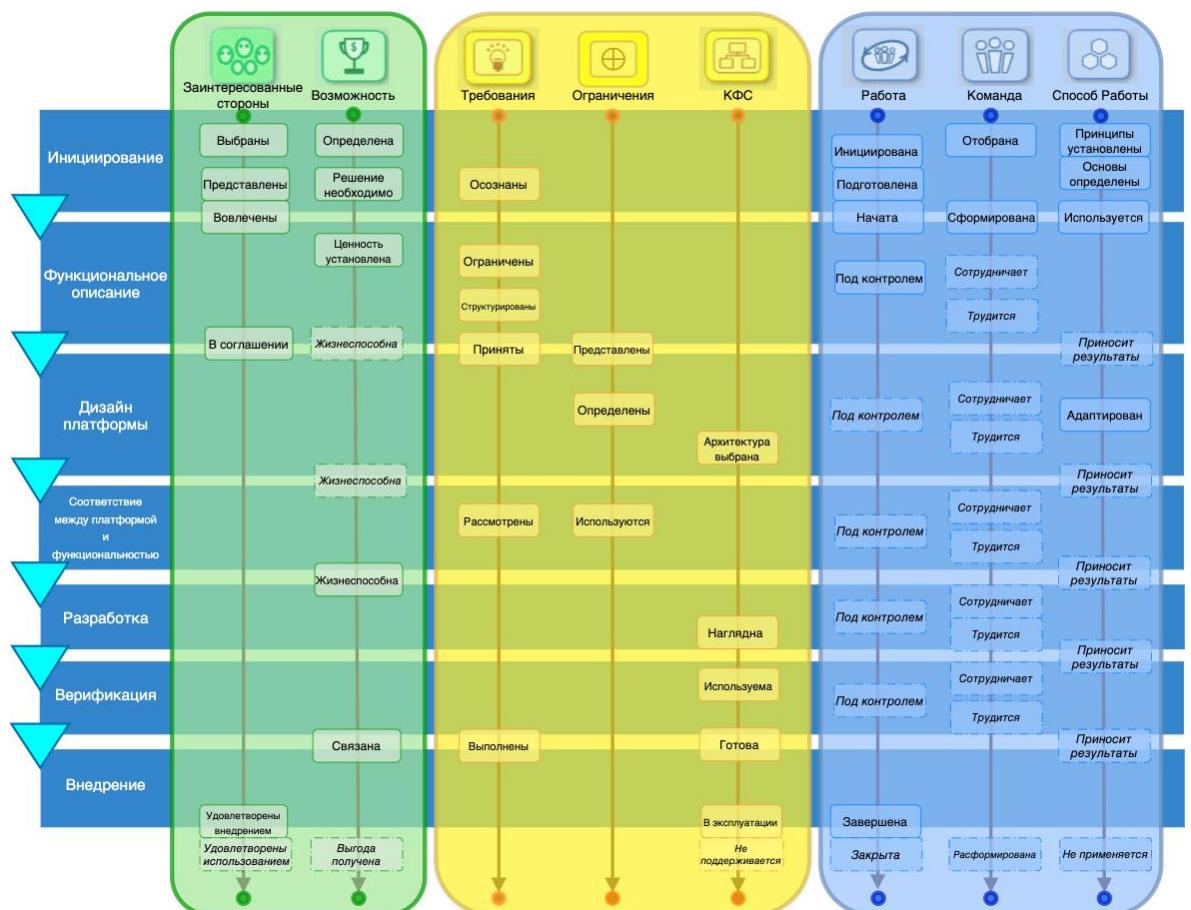


Рисунок 24 – Жизненный цикл проекта, разработка которого основана на концепции PBD

На рисунке 24 видно, что жизненный цикл состоит из семи фаз, они отражают идею проектирования на основе платформы – разделение создания функциональной части системы, то есть программной платформы («функциональное описание») от выбора и проектирования архитектуры, аппаратной платформы («дизайн платформы»). Фаза «соответствие между

платформой и функциональностью» означает создание системной платформы.

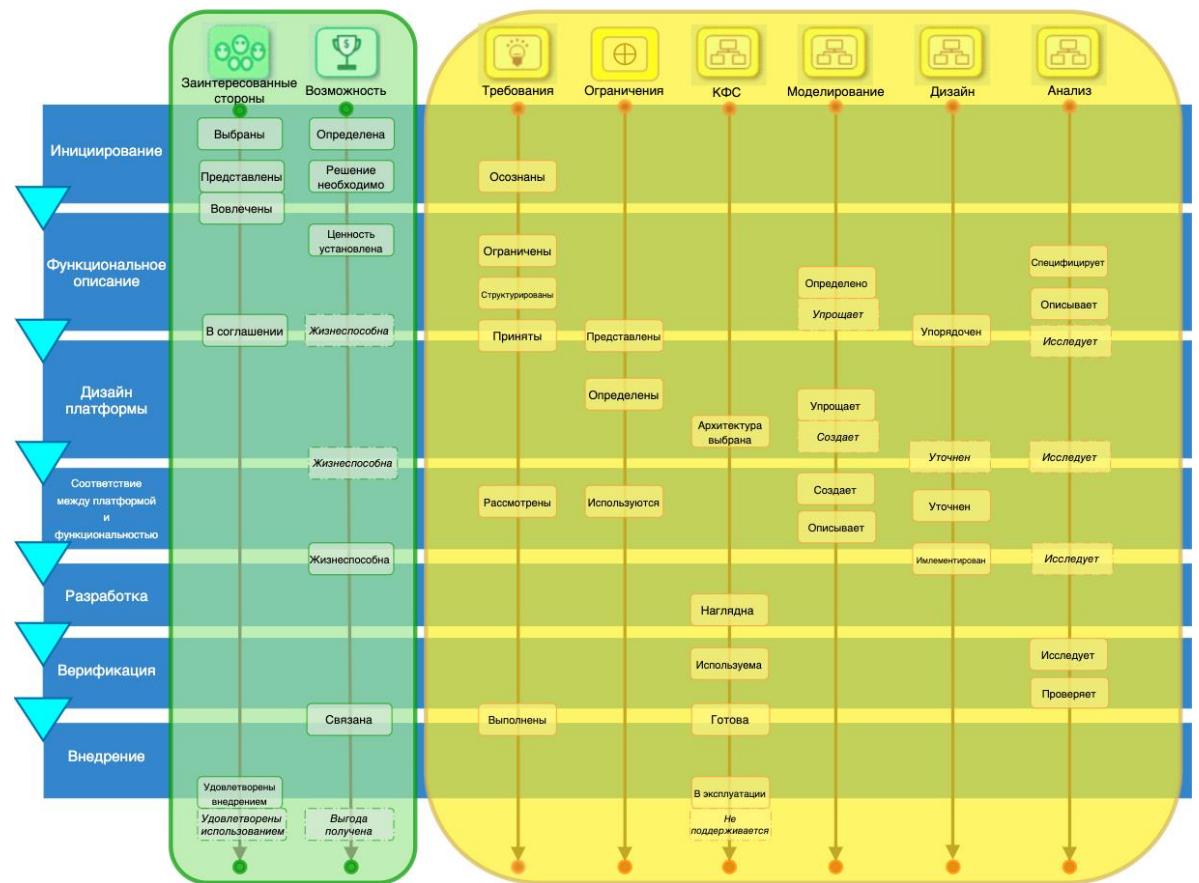


Рисунок 25 – Расширенный жизненный цикл проекта

На рисунке 25 в области Решения также отражены три саб-Альфы Киберфизической системы.

Всё в порядке!

5 КАРТЫ ESSENCE

Карты состояний Альф – это удобный способ отслеживания статуса проекта и планирования дальнейших действий.

Они могут быть использованы для:

- 1) понимания текущего состояния проекта;
- 2) устранения проблемных областей в разработке;
- 3) постановки целей для будущих итераций;
- 4) определения того, что можно улучшить;
- 5) проведения ретроспективы.

Карты обычно располагаются там, где все члены команды могут их видеть, чтобы в любой момент можно было бы к ним обратиться, посмотреть, где сейчас находится команда и определить, на чем она планирует сосредоточиться в ближайшее время.

и слоняются

Использование карт Essence может быть имплементировано в методологию Agile [22] и применяться, например, в спринтах Scrum или итерациях Kanban.

Agile предоставляет набор ценностей, которые влияют и формируют то, как разработчики выполняют свою повседневную работу и взаимодействуют друг с другом, со своими клиентами и заинтересованными сторонами.

Он также дает много методов, которые разделяют общие принципы, но отличаются на практике. Такие методы дают команде отличную отправную точку в их agile путешествии, но методы должны развиваться, чтобы соответствовать изменяющимся потребностям команды и отражать те уроки, которые команда выучила за время своего путешествия. Применение Essence в agile практике поможет команде обнаружить проблему на ранних этапах, оценить сильные и слабые стороны своего способа работы, постоянно следить за тем, как работает команда и своевременно делиться этой информацией с другими командами, установить основные правила

разработки в рамках организации, определить процедуры управления и оценки качества [22, с.55].



Рисунок 20 – Англоязычный и русский вариант карты состояний Альфы

На рисунке 20 можно увидеть то, как выглядят карты. Для каждого состояния делается отдельная карта с контрольным списком, отражающим особенности этого состояния.

На рисунке 21 отражена структура карты. В заголовке указывается символ и название элемента. В теле карты отражены конкретное состояние и контрольный список. Внизу карты для наглядности указывается порядковый номер состояния и количество всех состояний для данной Альфы.

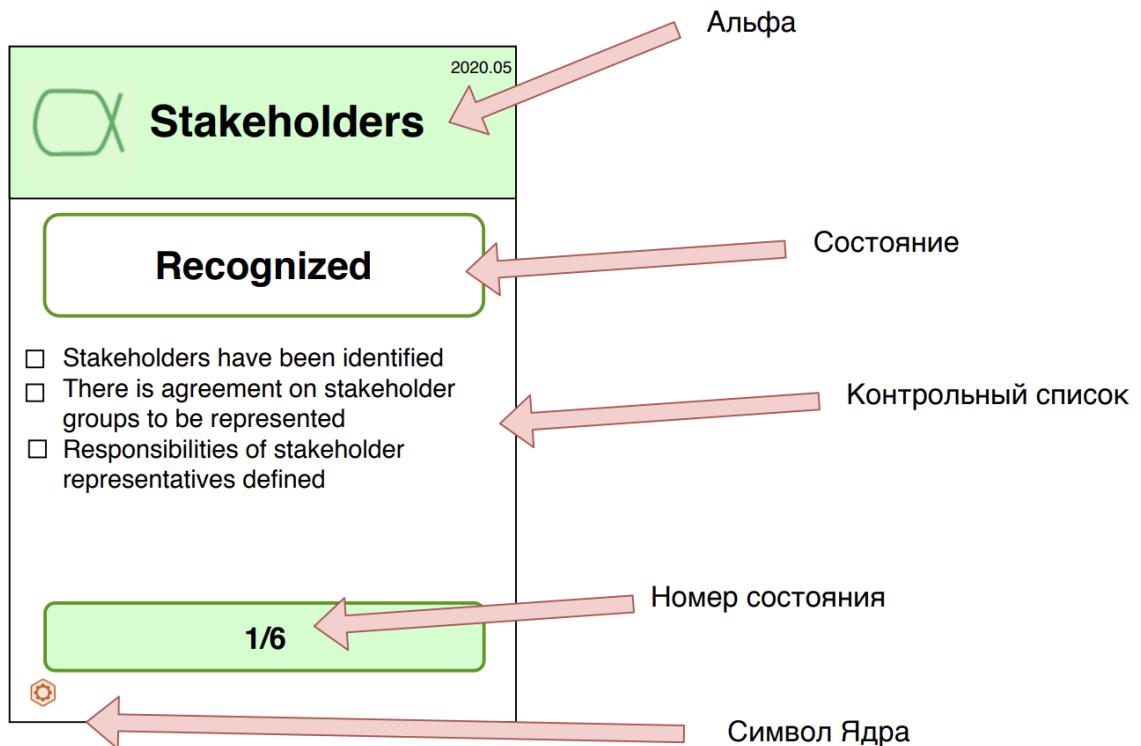


Рисунок 21 – Структура карты

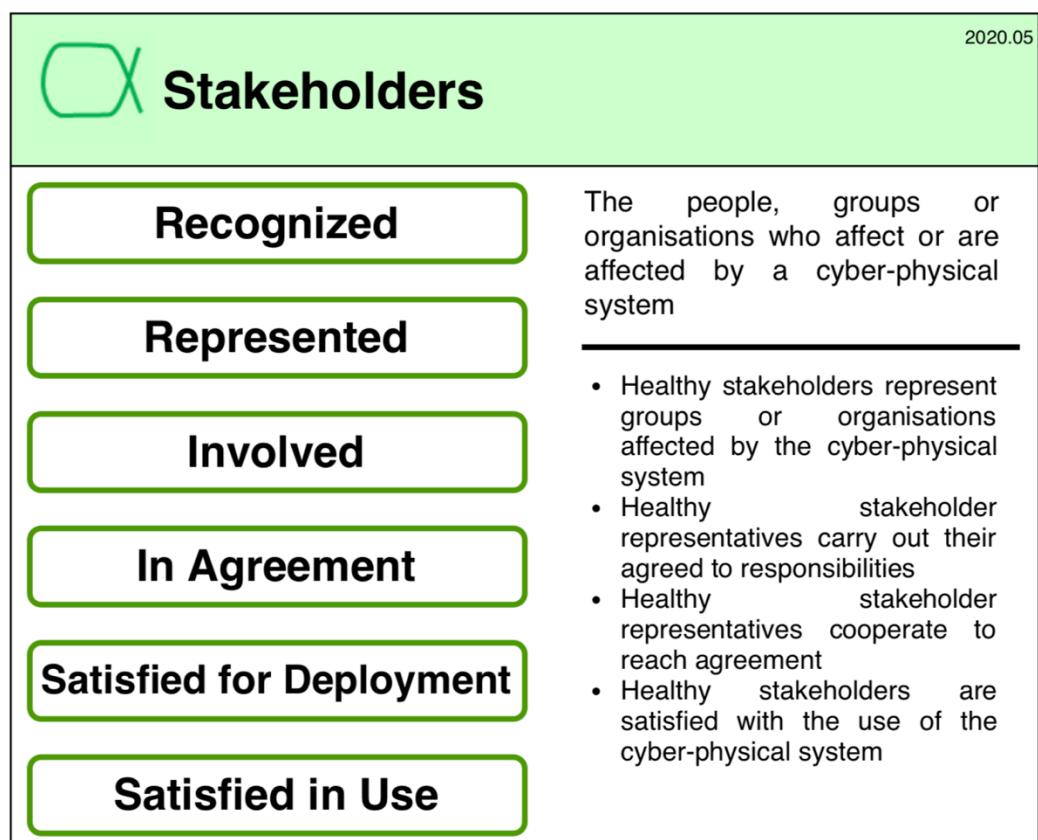


Рисунок 22 – Карта определения Альфы

Кроме карт состояний Альф есть также карты определения (рисунок 22). В них с левой стороны отражены все возможные состояния Альфы, а справа находятся краткое описание и характеристики элемента. Карты определения можно использовать для того, чтобы проследить взаимосвязь между состояниями. Для перехода из одного состояния в другое необходимо выполнить все пункты контрольного списка.

Полный комплект карт на русском языке можно найти в Приложении

1.
отработка
избег.

избег.

Тыс 2 раза. ~~сокращать разделять~~
~~и обзор и общество под ноги.~~

ЗАКЛЮЧЕНИЕ

Человек взаимодействует в своей повседневной жизни со многими сложными объектами и системами. Практически все они управляются компьютерами, некоторые из которых непосредственно влияют на физические процессы в окружающем мире. Такие системы, состоящие из взаимодействующих вычислительных элементов, находящихся в интенсивной связи с внешним миром, называются киберфизическими системами.

Киберфизические системы можно встретить в различных элементах нашей повседневной жизни, таких как энергетические сети, различные автоматизированные склады и фабрики, а также в самолетах, поездах, автомобилях. Подобные системы сложны в создании и управлении, их разработка требует слаженных четких процессов и не допускает ошибок, так как эксплуатация таких систем, как правило, связана напрямую с безопасностью людей.

Внедрение КФС в различные сферы поднимет качество жизни на новый уровень, однако на данный момент разработка и проектирование КФС связаны с несколькими проблемами, которые препятствуют их широкому распространению.

Одной из таких проблем, акцент на которую был сделан в данной работе, является неэффективная междисциплинарная коммуникации при разработке и проектировании киберфизических систем. Для возможного решения обозначенной проблемы был разработан инструмент оценки состояния проектов КФС.

В начале исследования были рассмотрены несколько методологий и концепций проектирования. Был проведен сравнительный анализ итерационной модели моделирование-дизайн-анализ, предложенной Эдвардом Ли и концепции проектирование на основе платформы, которая рассматривалась в том числе и в работах Альберто Санджиновани-Винчентелли.

Эдвард Ли предлагает рассматривать проектирование системы как непрерывный процесс моделирования, дизайна и анализа, который может переходить из одного состояния в другое неопределенное количество раз.

Проектирование на основе платформы – предлагает разделить процесс создания системы от процесса выбора архитектуры, с помощью создания аппаратной, программной и системной платформ.

Данный подход был призван решить проблему повторного использования аппаратных и программных решений. Однако выбор в результате сравнение был сделан в пользу итерационной модели Эдварда Ли, так как PBD все-таки рассказывает как проектировать систему, но не отвечает на вопрос, что именно для этого нужно сделать.

Следующим этапом теоретической части работы были изучение и сравнение инструментов формализации методов. Мной были рассмотрены средства, предложенные ситуационной инженерии методов, которая решает недостатки традиционной инженерии методов. В рамках SME был рассмотрен стандарт OPF, который является стандартом первого поколения SME и стандарт второго поколения OMG Essence. Также на данном этапе были рассмотрены и проанализированы системо-мыследеятельностная методология и региональные стандарты (ГОСТ).

Приоритет был отдан в пользу ситуационной инженерии методов, которая предоставляет конкретные инструменты, в отличие от СМД-методологии и отражает так же человеческую сторону процесса разработки, тогда как ЕСКД и ЕСПД сконцентрированы на описание общих требований к технической части.

Из двух стандартов мной был выбран OMG Essence, так как OPF больше ориентирован на методологов, которым необходимо описывать методы работы и систематизировать их, тогда как Essence создавался специально для инженеров.

В рамках теоретической части был также изучен сам стандарт Essence, структура его Ядра и методы формализации, которые он предлагает.

В практической части исследования было модернизировано Ядро Essence, таким образом, чтобы оно отражало специфику КФС, также были изменены и созданы новые Альфы, в том числе, представляющие из себя концепцию итерационной модели Эдварда Ли, выбранной мной на ранних этапах работы.

В практической части был апробирован предложенный способ формализации – с помощью модернизированного Ядра была описана концепция PBD.

Результатом практической части стало создание колоды карты Essence, с помощью которой можно отслеживать состояние проекта и регулировать взаимодействие между членами команды проектирования и разработки. Карты могут быть интегрированы в различные методологии и практики.

В результате работы были выполнены все поставленные изначально задачи, однако для того, чтобы утверждать о степени достижения цели исследования необходимо проверить эффективность и результативность созданного инструмента в реальных проектах и скорректировать его в соответствии с новыми условиями.

6 Теории должны быть отсортированы в
порядке уменьшения

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Henderson-Sellers, Brian, Ralyté Jolita, Ågerfalk Pär J., and Matti Rossi. Situational Method Engineering. Springer, 2016. – 307 с.
2. Lee, Edward A., and Sanjit A. Seshia. Introduction to Embedded Systems: a Cyber-Physical Systems Approach. Cambridge, Massachusetts: MIT Press, 2017. – P. 543
3. Платунов А.Е. Теоретические и методологические основы высокоуровневого проектирования вычислительных систем: дис. ... доктора. техн. наук: 25.00.15. – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, Санкт-Петербург, 2010 – 468 с.
4. Sangiovanni-Vincentelli A. Quo Vadis SLD: Reasoning about Trends and Challenges of System-Level Design // Proceedings of the IEEE, March → 2007. 95(3). P. 467–506.
5. Lee E.A., Sangiovanni-Vincentelli A. Comparing Models of Computation. Proceedings of ICCAD. San Jose, California, USA. November 10-14, 1996.
6. Jozwiak L., Nejah N., Figueroa M. Modern development methods and tools for embedded reconfigurable systems: A survey. // INTEGRATION, the VLSI journal, 07.2009, P. 1–33.
7. Sangiovanni-Vincentelli A. Defining platform-based design. // EEDesign. February 2002.
8. Khaitan, Siddhartha Kumar, and James D. Mccalley. Cyber Physical System Approach for Design of Power Grids: A Survey. // IEEE Power & Energy Society General Meeting, 2013.
9. Jin, Ruofan, Bing Wang, Peng Zhang, and Peter B. Luh. Decentralised Online Charging Scheduling for Large Populations of Electric Vehicles: a Cyber-Physical System Approach. // International Journal of Parallel, Emergent and Distributed Systems 28, no. 1 (2013): 29–45.
10. Barr, M. and A. Massa. Programming Embedded Systems. // O'Reilly, 2009 – P. 336

- 11.Pottie, G. and W. Kaiser. Principles of Embedded Networked Systems Design // Cambridge University Press, 2005.
12. Keutzer K., Malik S., Newton R., Rabaey J., Sangiovanni-Vincentelli A. System Level Design: Orthogonalization of Concerns and Platform-Based Design. // IEEE Transactions on Computer-Aided Design of Circuits and Systems. Vol. 19, No. 12. December 2000.
13. Sangiovanni-Vincentelli, A., and G. Martin. Platform-Based Design and Software Design Methodology for Embedded Systems. // IEEE Design & Test of Computers 18, no. 6. December 2001.
14. А. А. Пископпель. В. М. Розин. Ф. И. Голдберг. В. С. Бернштейн. А. В. Александров. – Системо-мыследеятельностная методология. / Гуманитарная энциклопедия: Концепты [Электронный ресурс] // Центр гуманитарных технологий, 2002–2020 (последняя редакция: 08.02.2020). – URL: <https://gtmarket.ru/concepts/7268>
15. Щедровицкий Г.П. Оргуправленческое мышление: идеология, методология, технология. // Студия Арт. Лебедева, 2013 – 468 с.
- 16.D. Zowghi , D.G. Firesmith, B. Henderson-Sellers. Using the OPEN Process Framework to Produce a Situation-Specific Requirements Engineering Method. // 1 University of Technology, 2005 – P.18
- 17.About the Essence Specification Version 1.2: Essence – Kernel and Language for Software Engineering Methods [Электронный ресурс]. – 2018. – URL: <https://www.omg.org/spec/Essence/About-Essence/> (дата обращения 13.12.2019).
18. ГОСТ 2.001-2013. Единая система конструкторской документации. Общие положения. – Взамен ГОСТ 2.001-93 01.06.2014. – М.: Стандартинформ, 2018
- 19.ГОСТ 19.001-77. Единая система программной документации. Общие положения. – М.: Стандартинформ, 2010
- 20.Cyber-Physical Systems [Электронный ресурс]. – 2012. – URL: <https://ptolemy.berkeley.edu/projects/cps/>

21. Brinkkemper, S. Method Engineering: Engineering of information systems development methods and tools. *Information and Software Technology*, 38(4), pp. 275-280, 1996
22. Jacobson, Ivar, et al. Agile and SEMAT. *Communications of the ACM*, vol. 56, no. 11, 2013, pp. 53–59.

ПРИЛОЖЕНИЕ 1 (колода карт Essence на русском языке)

<p>2020.05</p> <p> Заинтересованные стороны</p> <p>Выбраны</p> <ul style="list-style-type: none"><input type="checkbox"/> Заинтересованные стороны были определены<input type="checkbox"/> Представленные заинтересованные стороны были одобрены<input type="checkbox"/> Определены обязанности представителей заинтересованных сторон <p>1/6</p> 	<p>2020.05</p> <p> Заинтересованные стороны</p> <p>Представлены</p> <ul style="list-style-type: none"><input type="checkbox"/> Представители заинтересованных сторон были назначены<input type="checkbox"/> Представители заинтересованных стороны согласились взять на себя обязанности и ответственность<input type="checkbox"/> Подход к сотрудничеству был определен<input type="checkbox"/> Представители уважают работу команды <p>2/6</p> 
--	--

<p>2020.05</p> <p> Заинтересованные стороны</p> <p>Вовлечены</p> <ul style="list-style-type: none"><input type="checkbox"/> Представители заинтересованных сторон выполняют свои обязанности<input type="checkbox"/> Представители обеспечивают обратную связь и принимают своевременное участие в принятии решений<input type="checkbox"/> Представители заинтересованных сторон активно коммуницируют с заинтересованными сторонами <p>3/6</p> 	<p>2020.05</p> <p> Заинтересованные стороны</p> <p>В соглашении</p> <ul style="list-style-type: none"><input type="checkbox"/> Представители заинтересованных сторон согласны с тем, что их вклад оценивается командой<input type="checkbox"/> Представители согласны с расставленными приоритетами<input type="checkbox"/> Представители согласовали свои минимальные ожидания насчет внедрения КФС <p>4/6</p> 
---	--

 <p>Заинтересованные стороны</p> <p>Удовлетворены внедрением</p> <ul style="list-style-type: none"> <input type="checkbox"/> Представители заинтересованных сторон предоставляют обратную связь со стороны их заинтересованной группы <input type="checkbox"/> Представители подтверждают готовность системы к внедрению <p>5/6</p> <p></p>	 <p>Заинтересованные стороны</p> <p>Удовлетворены использованием</p> <ul style="list-style-type: none"> <input type="checkbox"/> Система оправдала или превзошла минимальные ожидания заинтересованных сторон <input type="checkbox"/> Потребности заинтересованных сторон удовлетворены <p>6/6</p> <p></p>
--	--

 <p>2020.05</p> <p>Возможность</p> <p>Определена</p> <ul style="list-style-type: none"> <input type="checkbox"/> Выявлена возможность, которая может быть реализована с помощью решения <input type="checkbox"/> Заинтересованная сторона хочет инвестировать, чтобы лучше разобраться в ценности данной возможности <input type="checkbox"/> Определены другие заинтересованные стороны, разделяющие данную возможность <p>1/6</p> <p></p>	 <p>2020.05</p> <p>Возможность</p> <p>Решение необходимо</p> <ul style="list-style-type: none"> <input type="checkbox"/> Необходимость в решении подтверждена <input type="checkbox"/> Определены потребности заинтересованных сторон <input type="checkbox"/> Выявлены основные проблемы и сложности <input type="checkbox"/> Предложено по крайней мере одно решение <p>2/6</p> <p></p>
--	---

 <p>Возможность</p> <p>Ценность установлена</p> <ul style="list-style-type: none"> <input type="checkbox"/> Ценность успешного решения установлена <input type="checkbox"/> Влияние решения на заинтересованные стороны определено <input type="checkbox"/> Ценность и значение киберфизической системы понятны <p>3/6</p> 	 <p>Возможность</p> <p>Жизнесспособна</p> <ul style="list-style-type: none"> <input type="checkbox"/> Решение было обрисовано <input type="checkbox"/> Решение может быть разработано и внедрено в рамках определенных ограничений <input type="checkbox"/> Риски контролируются <p>4/6</p> 
---	--

 <p>Возможность</p> <p>Связана</p> <ul style="list-style-type: none"> <input type="checkbox"/> Было представлено решение, которое напрямую связано с возможностью <input type="checkbox"/> Предложенное решение осуществимо <input type="checkbox"/> Заинтересованные стороны удовлетворены представленным решением <p>5/6</p> 	 <p>Возможность</p> <p>Выгода получена</p> <ul style="list-style-type: none"> <input type="checkbox"/> Использование приносит ощутимые выгоды <input type="checkbox"/> Отдача от инвестиций не ниже ожидаемого <p>6/6</p> 
---	---

<p> Команда</p> <p>2020.05</p> <p>Отобрана</p> <ul style="list-style-type: none"> <input type="checkbox"/> Миссия определена <input type="checkbox"/> Ограничения известные и определены <input type="checkbox"/> Обязанности изложены <input type="checkbox"/> Требуемые компетенции определены <input type="checkbox"/> Количество членов определено <input type="checkbox"/> Модель управления выбрана <p>1/5</p> <p></p>	<p> Команда</p> <p>2020.05</p> <p>Сформирована</p> <ul style="list-style-type: none"> <input type="checkbox"/> Найдено достаточное количество участников <input type="checkbox"/> Роли понятны <input type="checkbox"/> Члены Команды знают как выполнять свою работу <input type="checkbox"/> Организация Команды и индивидуальные обязанности определены <input type="checkbox"/> Механизм обратной связи внутри Команды определен <p>2/5</p> <p></p>
---	---

<p> Команда</p> <p>2020.05</p> <p>Сотрудничает</p> <ul style="list-style-type: none"> <input type="checkbox"/> Общение внутри Команды открытое и честное <input type="checkbox"/> Все работают как единое целое <input type="checkbox"/> Все сосредоточены на миссии команды <p>3/5</p> <p></p>	<p> Команда</p> <p>2020.05</p> <p>Трудится</p> <ul style="list-style-type: none"> <input type="checkbox"/> Адаптируется к новым обстоятельствам <input type="checkbox"/> Последовательно выполняет свои обязанности <input type="checkbox"/> Доработки минимальна <input type="checkbox"/> Дает качественный результат <p>4/5</p> <p></p>
---	--

2020.05

 **Команда**

Расформирована

- Обязательства выполнены
- Миссия завершена
- Члены Команды доступны для других команд

5/5



2020.05

 **Работа**

Инициирована

- Инициатор Работы известен
- Результат ясен
- Ограничения понятны
- Способы финансирования установлены
- Приоритет работы установлен

1/6



2020.05

 **Работа**

Подготовлена

- Проведена оценка затрат
- Есть финансирование и ресурсы для начала работы
- Критерии приема работы ясны
- Принятые рисков
- Задачи определены, для них установлены приоритеты
- Хотя бы один член команды готов начать работу

2/6



 <p>Работа</p> <p>Начата</p> <ul style="list-style-type: none"> <input type="checkbox"/> Разработка началась <input type="checkbox"/> Прогресс контролируется <input type="checkbox"/> Задачи в процессе выполнения <p>3/6</p> 	 <p>Работа</p> <p>Под контролем</p> <ul style="list-style-type: none"> <input type="checkbox"/> Задачи выполняются и закрываются <input type="checkbox"/> Незапланированная работа и доработки находятся под контролем <input type="checkbox"/> Работа выполняется в заданные сроки <input type="checkbox"/> Риски под контролем <p>4/6</p> 
---	---

 <p>Работа</p> <p>Завершена</p> <ul style="list-style-type: none"> <input type="checkbox"/> Работа для получения результатов была завершена <input type="checkbox"/> Результаты достигнуты <input type="checkbox"/> Клиент принял разработанную киберфизическую систему <p>5/6</p> 	 <p>Работа</p> <p>Закрыта</p> <ul style="list-style-type: none"> <input type="checkbox"/> Уроки извлечены, метрики работы команды доступны <input type="checkbox"/> Абсолютно все задачи были выполнены <input type="checkbox"/> Бюджет закрыт <input type="checkbox"/> Команда распущена <p>6/6</p> 
---	--

	Способ Работы	2020.05
Принципы установлены		
<ul style="list-style-type: none"><input type="checkbox"/> Принципы установлены<input type="checkbox"/> Заинтересованные стороны согласны с принципами<input type="checkbox"/> Метод и ограничения понятны<input type="checkbox"/> Контекст для работы ясен		
1/6		
		

	Способ Работы	2020.05
Основы определены		
<ul style="list-style-type: none"><input type="checkbox"/> Ключевые методы и инструменты выбраны<input type="checkbox"/> Расхождения между методами и инструментами проанализированы и понятны<input type="checkbox"/> Метод, необходимый для начала работы, согласован		
2/6		
		

	Способ Работы	2020.05
Используется		
<ul style="list-style-type: none"><input type="checkbox"/> Методы и инструменты используются<input type="checkbox"/> Адаптирован к изменяющемуся контексту и ограничениям<input type="checkbox"/> Регулярно проверяется<input type="checkbox"/> Механизм обратной связи работает		
3/6		
		

	Способ Работы	2020.05
Адаптирован		
<ul style="list-style-type: none"><input type="checkbox"/> Все члены команды применяют Способ Работы<input type="checkbox"/> Вся команда имеет доступ к инструментам и методам для выполнения своей работы		
4/6		
		

 <h2>Способ Работы</h2> <p>2020.05</p> <div style="border: 1px solid #ccc; padding: 5px; border-radius: 10px;"> Приносит результаты </div> <ul style="list-style-type: none"> <input type="checkbox"/> Прогнозируемый прогресс достигнут <input type="checkbox"/> Методы применяются естественным образом <input type="checkbox"/> Инструменты поддерживают способ работы <div style="background-color: #e0f2ff; border-radius: 10px; padding: 5px; text-align: center; margin-top: 20px;"> 5/6 </div> <div style="text-align: center; margin-top: 10px;">  </div>	 <h2>Способ Работы</h2> <p>2020.05</p> <div style="border: 1px solid #ccc; padding: 5px; border-radius: 10px;"> Не применяется </div> <ul style="list-style-type: none"> <input type="checkbox"/> Больше не используется Уроки извлечены, сделаны выводы <div style="background-color: #e0f2ff; border-radius: 10px; padding: 5px; text-align: center; margin-top: 20px;"> 6/6 </div> <div style="text-align: center; margin-top: 10px;">  </div>
---	---

 <h2>Требования</h2> <p>2020.05</p> <div style="border: 1px solid #ccc; padding: 5px; border-radius: 10px;"> Осознаны </div> <ul style="list-style-type: none"> <input type="checkbox"/> Заинтересованные стороны согласны с тем, что система должна быть создана <input type="checkbox"/> Пользователи определены <input type="checkbox"/> Спонсоры определены <div style="background-color: #ffffcc; border-radius: 10px; padding: 5px; text-align: center; margin-top: 20px;"> 1/6 </div> <div style="text-align: center; margin-top: 10px;">  </div>	 <h2>Требования</h2> <p>2020.05</p> <div style="border: 1px solid #ccc; padding: 5px; border-radius: 10px;"> Ограничены </div> <ul style="list-style-type: none"> <input type="checkbox"/> Цель и результаты системы согласованы <input type="checkbox"/> Критерии успеха понятны Возможные допущения обозначены <input type="checkbox"/> Механизм обработки требований понятен и определен <div style="background-color: #ffffcc; border-radius: 10px; padding: 5px; text-align: center; margin-top: 20px;"> 2/6 </div> <div style="text-align: center; margin-top: 10px;">  </div>
---	--

 <p>Требования</p> <p>2020.05</p> <p>Структурированы</p> <ul style="list-style-type: none"> <input type="checkbox"/> Приоритеты ясны <input type="checkbox"/> Конфликты разрешены <input type="checkbox"/> Общая картина ясна и понятна всем <p>3/6</p> 	 <p>Требования</p> <p>2020.05</p> <p>Приняты</p> <ul style="list-style-type: none"> <input type="checkbox"/> Ценность, которая будет получена, ясна <input type="checkbox"/> Поддаются тестированию <input type="checkbox"/> Приемлемое решение описано <p>4/6</p> 
---	--

 <p>Требования</p> <p>2020.05</p> <p>Рассмотрены</p> <ul style="list-style-type: none"> <input type="checkbox"/> Достаточное количество требований реализовано, чтобы система стала приемлемой для заинтересованных сторон <input type="checkbox"/> Заинтересованные стороны согласны с тем, что систему можно ввести в эксплуатацию <p>5/6</p> 	 <p>Требования</p> <p>2020.05</p> <p>Выполнены</p> <ul style="list-style-type: none"> <input type="checkbox"/> Система полностью удовлетворяет требованиям и потребностям <p>6/6</p> 
--	--

 Киберфизическая система Архитектура выбрана <ul style="list-style-type: none"> <input type="checkbox"/> Выбрана архитектура, учитывающая ключевые технические риски и ограничения <input type="checkbox"/> Критерии выбора архитектуры согласованы <input type="checkbox"/> Технологии и методы для разработки выбраны в соответствии с ограничениями <p>1/6</p>	 Киберфизическая система Наглядна <ul style="list-style-type: none"> <input type="checkbox"/> Ключевые архитектурные характеристики показаны <input type="checkbox"/> Измеряется производительность системы <input type="checkbox"/> Показано взаимодействие с окружающей средой <p>2/6</p>
--	--

Ключевые
текст.
термины
измерив
гуманитарных

 Киберфизическая система Используема <ul style="list-style-type: none"> <input type="checkbox"/> Система пригодна для использования и обладает необходимыми характеристиками <input type="checkbox"/> Пользователи могут управлять системой <input type="checkbox"/> Функциональность и производительность были протестированы <p>3/6</p>	 Киберфизическая система Готова <ul style="list-style-type: none"> <input type="checkbox"/> Пользовательская документация доступна <input type="checkbox"/> Представители заинтересованных сторон принимают систему <input type="checkbox"/> Представители заинтересованных сторон хотят запустить систему в эксплуатацию <p>4/6</p>
--	---

 <p>Киберфизическая система</p> <p>В эксплуатации</p> <ul style="list-style-type: none"> <input type="checkbox"/> Система находится в эксплуатации <input type="checkbox"/> Система доступна всем предназначенным пользователям <input type="checkbox"/> Поддерживаются согласованные уровни обслуживания системы <p>5/6</p> <p></p>	 <p>Киберфизическая система</p> <p>Не поддерживается</p> <ul style="list-style-type: none"> <input type="checkbox"/> Больше не поддерживается <input type="checkbox"/> Обновления системы больше не выпускаются <input type="checkbox"/> Работа системы была остановлена или система была заменена другой системой <p>6/6</p> <p></p>
--	---

 <p>Ограничения</p> <p>Представлены</p> <ul style="list-style-type: none"> <input type="checkbox"/> Специфика киберфизической системы ясна <input type="checkbox"/> Формат Ограничений согласован <input type="checkbox"/> Ответственные за работу с Ограничениями определены <p>1/3</p> <p></p>	 <p>Ограничения</p> <p>Определены</p> <ul style="list-style-type: none"> <input type="checkbox"/> Все Ограничения определены <input type="checkbox"/> Понятны всем участникам <input type="checkbox"/> Влияние Ограничений на систему определено <p>2/3</p> <p></p>
--	---

2020.05

Ограничения

Используются

- Способ работы скорректирован в соответствии с Ограничениями
- Ограничения и Требования не противоречат друг другу

3/3



2020.05

Моделирование

Определено

- Создано описание определенных аспектов системы
- Все Требования учтены *ограничения?*
- Определено поведение системы

1/4



Упрощает

- Система разделена на компоненты
- Определено развитие системы со временем *б/з?*

2/4



Правильное
голосование
вроде
же
б/з?

<p>2020.05</p> <p>Моделирование</p> <p>Создает</p> <ul style="list-style-type: none"> <input type="checkbox"/> Создано достаточно моделей для отражения всех аспектов системы <input type="checkbox"/> Модели имеют математические свойства <p><i>Было создано достаточно моделей для отражения всех аспектов системы.</i></p> <p>3/4</p>	<p>2020.05</p> <p>Моделирование</p> <p>Описывает</p> <ul style="list-style-type: none"> <input type="checkbox"/> Модели точно описывают свойства системы <p><i>Модели описывают свойства системы с необходимостью точностью</i></p> <p>4/4</p>
--	---

<p>2020.05</p> <p>Дизайн</p> <p>Упорядочен</p> <ul style="list-style-type: none"> <input type="checkbox"/> Все компоненты, которые должны быть разработаны для системы, выбраны и описаны <input type="checkbox"/> Выбранные компоненты реализуют все необходимые системе функции <p>1/3</p>	<p>2020.05</p> <p>Дизайн</p> <p>Уточнен</p> <ul style="list-style-type: none"> <input type="checkbox"/> Предложено хотя бы одно архитектурное решение <input type="checkbox"/> Все компоненты системы реализованы <p>2/3</p>
---	---

*Разные это не синонимы?
Что?*

2020.05



Дизайн

Имплементирован

- Все разработанные компоненты взаимодействуют друг с другом
- Желаемый функционал системы реализован

3/3



2020.05



Анализ

Специфицирует

- Все Требования преобразованы в спецификации
- Спецификации согласованы

1/4



2020.05



Анализ

Описывает

- Желаемое и нежелательное поведение системы описано
- Условия, которые должны быть проанализированы для моделей и дизайна, созданы

2/4



↗ Методика
проблематика
требование/ограничение
определение

 Анализ	2020.05
<div style="border: 2px solid orange; padding: 10px; text-align: center;">Исследует</div> <p><input type="checkbox"/> Модели и дизайн проверены в соответствии со всеми условиями</p>	<div style="border: 2px solid orange; padding: 10px; text-align: center;">Проверяет</div> <p><input type="checkbox"/> Дизайн корректный образом реализует спецификацию <input type="checkbox"/> Все несоответствия в работе моделей выявлены и исследованы <input type="checkbox"/> Все свойства системы проверны</p>

3/4



4/4

