

МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА  
ВЕЛИКОГО»

Институт компьютерных наук и кибербезопасности

Высшая школа технологий искусственного интеллекта

Реферат по дисциплине  
«Управление проектами»  
«Стадии проектирования  
автоматизированных информационных  
систем»

Студент: \_\_\_\_\_

Салимли Айзек Мухтар Оглы

Преподаватель: \_\_\_\_\_

Большаков Александр Афанасьевич

«\_\_\_\_»\_\_\_\_\_ 20\_\_ г.

Санкт-Петербург, 2025

# Содержание

<b>Введение</b>	<b>3</b>
<b>1 Принципы организации проектирования АИС</b>	<b>4</b>
1.1 Системное мышление и поэтапность работ . . . . .	4
1.2 Разделение на слои: данные, логика, интерфейс и их согласованность . . . . .	4
1.3 Фокус на изменяемость предметной области и интеграцию . . . . .	4
<b>2 Этапы проектирования АИС</b>	<b>6</b>
2.1 Содержательное описание объекта и процессов . . . . .	6
2.2 Формализованная схема / математическая модель . . . . .	6
2.3 ТЗ на систему и подсистемы: пилот-проект и внедрение . . . . .	6
<b>3 Классический подход / метамодель</b>	<b>8</b>
3.1 Классика: локальные решения, дорогие изменения схем данных . . . . .	8
3.2 Метамодель: гибкость и переиспользование . . . . .	8
<b>4 Роли и артефакты</b>	<b>9</b>
4.1 Единая терминология и модели: Заказчик → проектировщик → разработчик . . .	9
4.2 Артефакты: Модели требований, Архитектуры, Сценарии поведения . . . . .	9
<b>5 Стандарты жизненного цикла АИС: ISO/IEC 12207 и ISO/IEC 15288</b>	<b>10</b>
5.1 ISO/IEC 12207 (ПО) . . . . .	10
5.2 ISO/IEC 15288 (Системы) . . . . .	11
<b>6 Примеры из реальной жизни</b>	<b>12</b>
6.1 ERP-внедрение   ОАО «Тагмет» (SAP R/3) . . . . .	12
6.2 Авиационная отрасль . . . . .	12
<b>7 ПО для проектирования АИС</b>	<b>13</b>
7.1 Визуальное моделирование . . . . .	13
7.2 Верификация . . . . .	13
7.3 Отечественное ПО . . . . .	14
<b>Заключение</b>	<b>15</b>
<b>Список литературы</b>	<b>16</b>

## **Введение**

Автоматизированная информационная система (АИС) — это комплекс, включающий программное обеспечение, аппаратные средства, и данные, объединенные для сбора, обработки, хранения и выдачи информации в рамках конкретной предметной области. Основная цель АИС — повышение эффективности управления и поддержки бизнес-процессов.

Современные АИС создаются в условиях быстро меняющихся требований и интеграции с чужими системами.

Классические подходы (сверху-вниз/снизу-вверх, процессный/структурный) эффективны для локальных задач, но хуже адаптируются к изменениям.

Подход на основе метамоделей повышает гибкость за счёт описания на более высоком уровне абстракции и согласованного изменения слоёв данных/логики/интерфейса.

# 1 Принципы организации проектирования АИС

## 1.1 Системное мышление и поэтапность работ

Системное мышление в контексте проектирования информационных систем (АИС) предполагает интеграцию различных аспектов работы: от анализа потребностей до внедрения системы в эксплуатацию. Каждое решение в процессе проектирования должно быть принято с учётом всех взаимосвязанных факторов. Это позволяет создавать систему, которая будет работать как единое целое, а не как совокупность отдельных решений.

Процесс проектирования всегда разбивается на несколько этапов, что даёт возможность детально проработать каждый элемент системы и снизить риски. Подход поэтапного проектирования включает в себя анализ, проектирование, верификацию, тестирование и внедрение. Такой подход позволяет гибко адаптироваться к изменениям на каждом из этапов и учитывать реальные потребности бизнеса и заказчика.

## 1.2 Разделение на слои: данные, логика, интерфейс и их согласованность

Проектирование АИС часто включает разделение на несколько слоёв — данные, логика, и интерфейс. Каждый слой представляет собой отдельную подсистему, решающую свою задачу, но при этом все они должны быть согласованы между собой. Это ключевой момент, поскольку согласованность между слоями обеспечивает стабильность работы системы в целом.

- **Слой данных** включает в себя модели данных, базы данных и связанные с ними процессы обработки данных.
- **Слой логики** включает в себя бизнес-логику и алгоритмы, которые должны обеспечивать выполнение всех операций системы.
- **Слой интерфейса** представляет собой взаимодействие пользователя с системой, будь то графический интерфейс или интерфейсы взаимодействия с другими системами.

Обеспечение согласованности между этими слоями позволяет системе быть гибкой и легко адаптируемой к изменениям, поскольку изменение одного слоя должно минимально затрагивать другие.

## 1.3 Фокус на изменяемость предметной области и интеграцию

Очень важным аспектом проектирования АИС является способность системы адаптироваться к изменяющимся требованиям предметной области и возможности интеграции с внешними системами. В условиях быстро меняющегося рынка, где появляются новые требования и технологии, система должна быть гибкой и адаптируемой.

Процесс проектирования должен учитывать возможности для добавления новых функциональностей или изменения существующих без значительных затрат на переработку всей системы. Также важно предусмотреть интеграцию с уже существующими системами и внешними сервисами, чтобы избежать избыточных затрат и повысить эффективность.

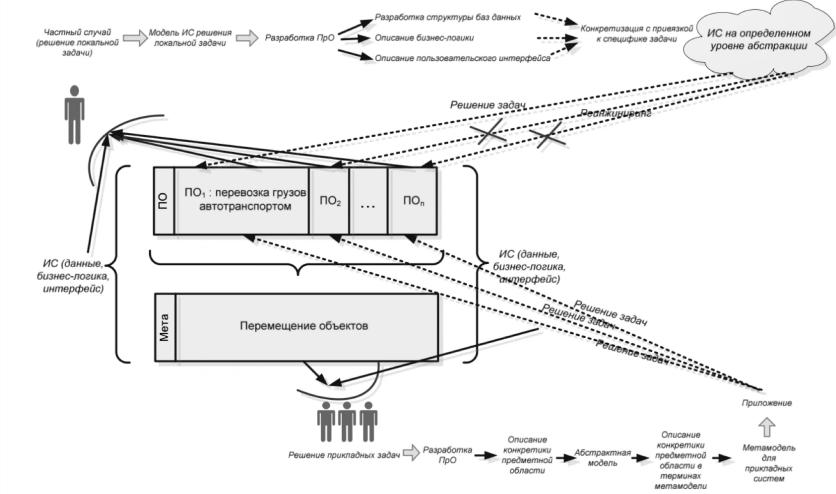


Рис. 1: Схема принципов и планирования

Рис. 1: Схема принципов и планирования.

## **2 Этапы проектирования АИС**

### **2.1 Содержательное описание объекта и процессов**

На начальном этапе проектирования важно правильно описать объект автоматизации и процессы, которые будут в ней отражены. Это описание является основой для дальнейшей работы и включает в себя подробное представление бизнес-процессов, требований заказчика и особенностей работы самой системы.

Данный этап предполагает не только изучение предметной области, но и анализ существующих систем, изучение их сильных и слабых сторон. Важно собрать всю необходимую информацию, которая поможет в дальнейшем процессе разработки.

### **2.2 Формализованная схема / математическая модель**

После того как были собраны все необходимые данные и требования, создаётся формализованная схема или математическая модель системы. Эта модель позволяет чётко определить все связи между компонентами системы и процессы, которые в ней будут реализованы.

Математические модели играют важную роль, так как позволяют протестировать систему на этапе проектирования. С помощью таких моделей можно заранее определить возможные проблемы, оценить эффективность предложенных решений и провести оптимизацию.

### **2.3 ТЗ на систему и подсистемы: пилот-проект и внедрение**

Следующим этапом является создание технического задания (ТЗ) на систему и её подсистемы. Это документ, который чётко определяет, что должно быть реализовано в системе, какие функции она должна выполнять и какие требования предъявляются к её производительности, безопасности и устойчивости.

После разработки ТЗ приступают к созданию пилотного проекта — прототипа системы. Пилот позволяет протестировать систему в реальных условиях и выявить возможные ошибки или недочёты. На основе результатов тестирования пилота вносятся корректизы, после чего система внедряется в эксплуатацию.



Рис. 2: Диаграмма этапов

### 3 Классический подход / метамодель

#### 3.1 Классика: локальные решения, дорогие изменения схем данных

Классический подход к проектированию АИС основывается на создании локальных решений, каждый из которых решает свою задачу. Это решение может быть простым и эффективным для определённого набора задач, однако оно часто ограничивает гибкость системы. Одним из недостатков классического подхода является высокая стоимость изменений в структуре данных, так как любые изменения на одном уровне системы могут потребовать переработки и других её частей.

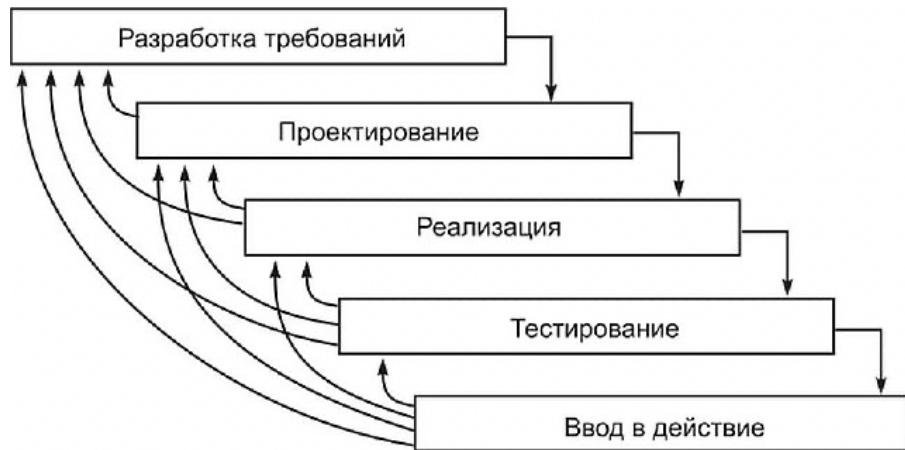


Рис. 3: Классический подход

#### 3.2 Метамодель: гибкость и переиспользование

Метамодельный подход позволяет описывать систему на более высоком уровне абстракции. Вместо того чтобы разрабатывать каждое решение отдельно, создаются метамодели, которые описывают не только саму систему, но и её основные компоненты, а также их взаимосвязи. Это даёт возможность:

- Гибко реагировать на изменения в требованиях.
- Переиспользовать компоненты для различных проектов.
- Легче интегрировать новые технологии и решения.

Метамодели включают в себя:

- **Метаданные** — данные о данных, которые описывают их структуру и логику.
- **Металогика** — описание логики работы системы на более высоком уровне абстракции.
- **Мета-интерфейсы** — интерфейсы, которые обеспечивают взаимодействие между различными компонентами системы, при этом эти интерфейсы могут быть адаптированы или изменены с минимальными затратами.

## 4 Роли и артефакты

### 4.1 Единая терминология и модели: Заказчик → проектировщик → разработчик

Процесс проектирования требует четкой согласованности терминологии и моделей между всеми участниками проекта. Заказчик, проектировщик и разработчик должны использовать единую терминологию, чтобы избежать недопонимания и ошибок в процессе разработки.

- **Заказчик** формулирует требования к системе.
- **Проектировщик** разрабатывает архитектуру системы и её функциональные характеристики.
- **Разработчик** реализует систему в соответствии с архитектурой и требованиями.

### 4.2 Артефакты: Модели требований, Архитектуры, Сценарии поведения

Во время проектирования создаются различные артефакты, которые являются результатами работы на каждом этапе:

- **Модели требований** описывают, какие функции и задачи должна выполнять система.
- **Архитектуры** системы — это её структура, включающая все компоненты и их взаимодействия.
- **Сценарии поведения** — это детализированные описания того, как система должна реагировать на различные входные данные и взаимодействие с пользователями.



Рис. 4: Артефакты процесса управления

## 5 Стандарты жизненного цикла АИС: ISO/IEC 12207 и ISO/IEC 15288

### 5.1 ISO/IEC 12207 (ПО)

Этот стандарт описывает процессы, которые должны быть выполнены при разработке программного обеспечения. Он делится на несколько категорий процессов:

- 5 основных процессов:

- **Заказ:** определение требований и планирование.
  - **Поставка:** внедрение и настройка ПО.
  - **Разработка:** создание и тестирование программного обеспечения.
  - **Эксплуатация:** использование ПО в реальных условиях.
  - **Сопровождение:** поддержка и развитие системы после её внедрения.

- 4 организационных процесса:

- Управление проектом, инфраструктура, совершенствование процессов и обучение.

- 8 вспомогательных процессов:

- Включает в себя документацию, управление конфигурацией, контроль качества, верификацию и аттестацию, аудит и решение проблем.

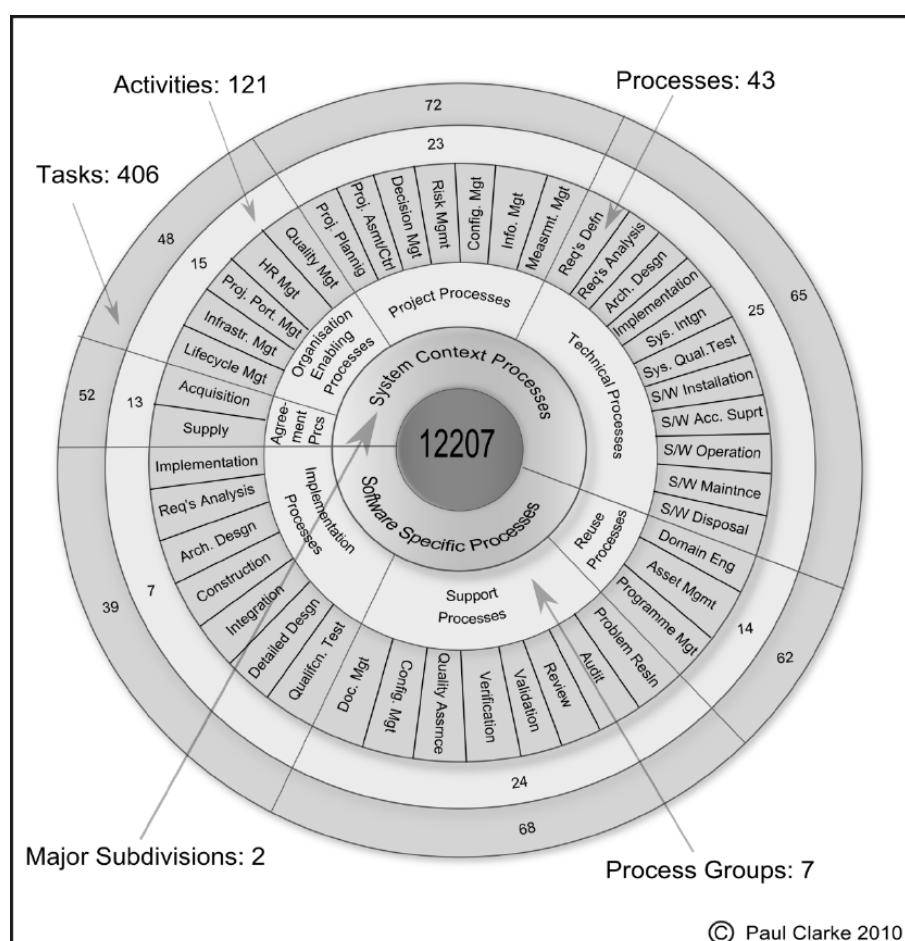


Рис. 5: Диаграмма стандарта ISO-12207

## 5.2 ISO/IEC 15288 (Системы)

Этот стандарт охватывает более широкий спектр, чем 12207, так как включает в себя процессы, связанные с проектированием и жизненным циклом сложных систем (не только ПО). Он разделяется на несколько групп процессов, включая технические, проектные и обеспечивающие процессы.

- Группы процессов:

- Соглашение (приобретение, постановка)
- Обеспечивающие на уровне предприятия
- Проектные (планирование, риски, конфигурация)
- Технические (требования, архитектура, реализация, интеграция, вериф./валидация, эксплуатация, сопровождение, списание)

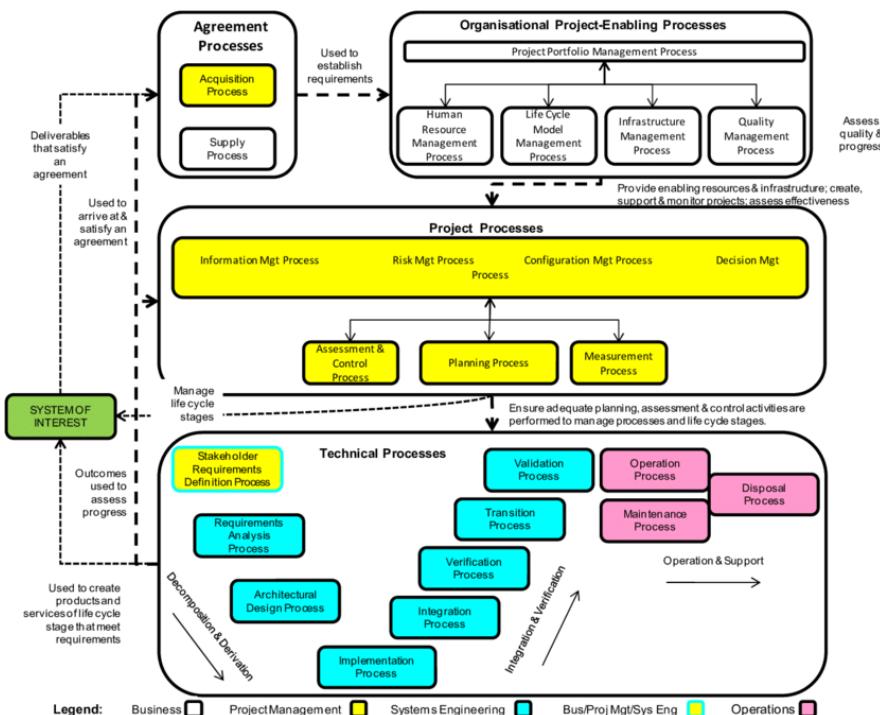


Рис. 6: Диаграмма стандарта ISO-15288

## **6 Примеры из реальной жизни**

### **6.1 ERP-внедрение | ОАО «Тагмет» (SAP R/3)**

Внедрение ERP-системы на базе SAP R/3 на ОАО «Тагмет» показало высокие затраты и длительный процесс адаптации системы под специфику конкретного предприятия. Адаптация ERP-системы требует серьёзных усилий по настройке и изменению её функционала, что может занять месяцы. Это доказывает, что для специфических предприятий целесообразно разрабатывать собственные АСУ, которые будут более гибкими в адаптации и соответствовать потребностям бизнеса.

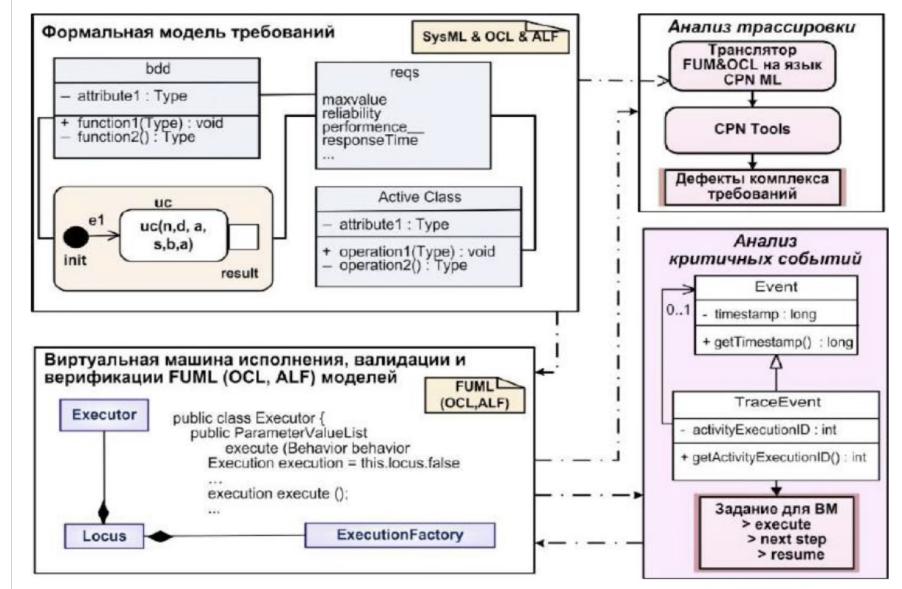
### **6.2 Авиационная отрасль**

В авиационной отрасли на ранних стадиях проектирования часто выявляются дефекты, которые можно было бы избежать, если бы использовались более продвинутые подходы к моделированию. Использование архитектурных моделей, например, на AADL, и проведение анализа безопасности помогает значительно улучшить проект, выявляя возможные уязвимости ещё до того, как система будет построена. Это подтверждает важность верификации на стадии проектирования и моделирования.

## 7 ПО для проектирования АИС

### 7.1 Визуальное моделирование

Для проектирования АИС широко используются инструменты визуального моделирования, такие как **IBM Rhapsody**, **Sparx EA**, **Modelio**, **Eclipse Papyrus**. Эти инструменты позволяют строить модели, использовать стандарты SysML/UML для представления архитектуры системы и её компонентов.



Caption

Рис. 7: Диаграмма требований к функциональным возможностям АИС

### 7.2 Верификация

Для верификации систем используются такие инструменты, как **CPN Tools** и **Rodin**, которые поддерживают моделирование и симуляцию сетей Петри и других формальных моделей.

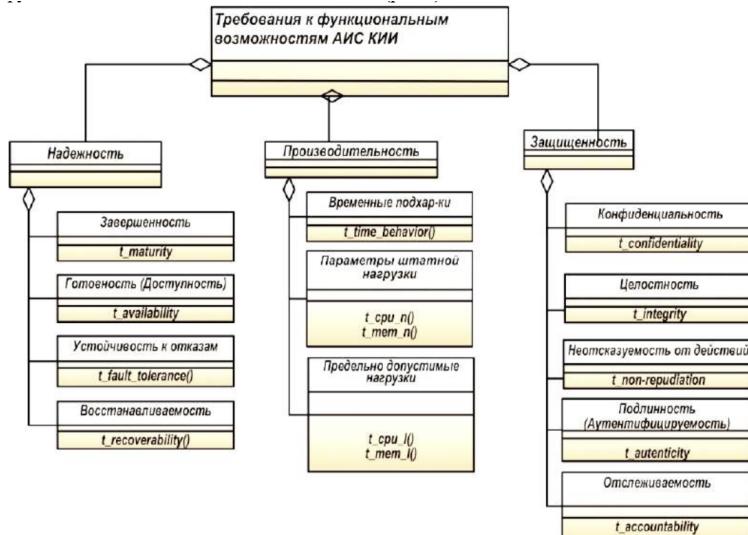
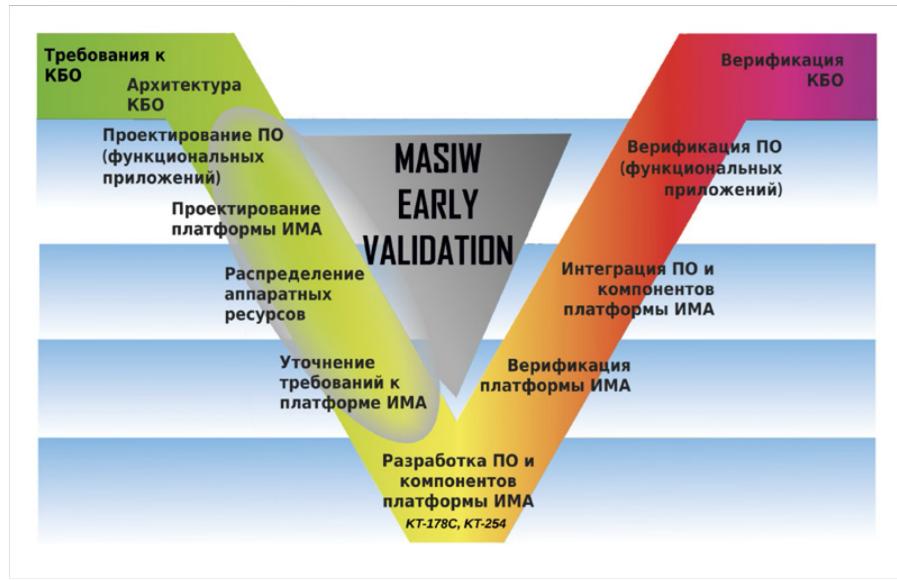


Рис. 8: Диаграмма связанных ПО, для проектирования АИС

### 7.3 Отечественное ПО

Примером отечественного ПО является **MASIW** (ИСП РАН), предназначенное для проектирования и верификации бортовых ПАК на основе AADL. Этот инструмент позволяет анализировать ресурсы, интерфейсы и сети, а также генерировать конфигурации для сложных систем, что очень важно в высокотехнологичных отраслях.



Caption

Рис. 9: Отечественное ПО, MASIW (ИСП РАН)

## Заключение

В работе последовательно рассмотрены принципы организации, этапы и стандарты жизненного цикла автоматизированных информационных систем, а также средства их проектирования и проверки. Показано, что результативная разработка опирается на системное мышление и поэтапное выполнение работ: от содержательного описания предметной области и бизнес-процессов — к формальной модели, далее к корректно оформленному техническому заданию, пилотированию и промышленному внедрению. Разделение на слои (данные, логика, интерфейс) и поддержание согласованности между ними обеспечивает управляемость изменений и упрощает интеграцию с внешними системами.

Сравнение классического и метамодельного подходов показывает, что классическая схема быстро даёт локальный результат, но дорого обходится при изменении схем данных и масштабировании. Метамодельная инженерия, опирающаяся на метаданные, металлогику и мета-интерфейсы, обеспечивает переиспользование решений, трассируемость артефактов и устойчивость к изменениям требований на всём горизонте жизни системы. Единая терминология между заказчиком, проектировщиком и разработчиком, а также связные модели требований, архитектуры и сценариев поведения снижают риск несоответствий на стыках работ.

Стандарты ISO/IEC 12207 (для программного обеспечения) и ISO/IEC 15288 (для систем) задают полный набор процессов: от соглашений и организационного обеспечения до проектных и технических процедур, включая верификацию, валидацию, сопровождение и списание. Их применение позволяет формализовать управление качеством, конфигурацией и рисками, а также обеспечить воспроизводимость и контролируемость результатов.

Анализ практических ситуаций подтверждает выводы: ERP-внедрения уровня SAP R/3 требуют значительных затрат и длительной адаптации под отраслевую специфику; при высокой уникальности процессов целесообразно рассматривать собственные АСУ. В авиационной тематике раннее архитектурное моделирование (AADL), анализ безопасности и генерация целевых конфигураций (например, под VxWorks 653) позволяют выявлять дефекты на стадии проектирования и сокращать стоимость исправлений.

Инструментальная база поддерживает методологию: средства визуального моделирования (IBM Rhapsody, Sparx EA, Modelio, Eclipse Papyrus) обеспечивают описание архитектуры и трасируемость; формальные средства (CPN Tools, Rodin) — проверку корректности моделей; платформы для исполняемых предметно-ориентированных языков (GEMOC Studio) и отечественные решения (MASIW для AADL) — поддержку сложных и критичных доменов. В совокупности это снижает технические и проектные риски, повышает качество, сопровождаемость и интегрируемость АИС.

## Список литературы

- [1] Шевченко, О.В. *Анализ современных подходов проектирования информационных систем.* — Известия ЮФУ. Технические науки, с. 89–93.
- [2] Назарова, О.Б. *Реализация процессов жизненного цикла сложных автоматизированных систем на основе стандартов программной инженерии.* — (научная статья, описание стандартов ISO/IEC 12207 и ISO/IEC 15288).
- [3] Рогозов, Ю.И.; Финаев, В.И. *Этапы проектирования автоматизированных систем управления.* — Известия ТРТУ.
- [4] Самонов, А.В. *Методы и средства разработки автоматизированных информационных систем на основе онтологии «Управление качеством программно-технических комплексов».* — Труды ИСП РАН, т. 31, вып. 5, 2019, с. 165–182. DOI: 10.15514/ISPRAS-2019-31(5)-13.