

**Министерство образования и науки Российской Федерации  
ФГАОУ ВО «Севастопольский государственный университет»**

**Институт информационных технологий  
и управления в технических системах  
Кафедра «Информационные системы»**

# **СИСТЕМНЫЙ АНАЛИЗ И ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ**

**Учебно-методическое пособие**  
для студентов всех форм обучения направления подготовки  
09.04.02 «Информационные системы и технологии» (магистратура)

Севастополь  
**2020**

**УДК 004.07**

Учебно-методическое пособие по дисциплине «Системный анализ и проектирование информационных систем» для студентов всех форм обучения направления подготовки 09.04.02 «Информационные системы и технологии» (магистратура) / Разраб. Ю.В. Доронина – Севастополь: Изд-во СевГУ, 2020. – 70 с.

Рассмотрено и утверждено на заседании кафедры «Информационные системы», протокол № \_\_\_\_ от «30» августа 2019 г.

Допущено учебно-методическим центром СГУ в качестве учебно-методического пособия.

## Оглавление

Введение.....	4
1.СИСТЕМНЫЙ АНАЛИЗ КАК МЕТОД СОЗДАНИЯ И ИССЛЕДОВАНИЯ СЛОЖНЫХ СИСТЕМ.....	5
2.ОБЩИЕ СВЕДЕНИЯ О ПРОЕКТИРОВАНИИ ИНФОРМАЦИОННЫХ СИСТЕМ .....	8
2.1 Принципы создания информационных систем .....	<b>Ошибка! Закладка не определена.</b>
2.2 Подходы к созданию информационных систем .....	9
2.3 Стандартизация, комплексы стандартов на информационные системы и технологии .....	10
2.4 Жизненный цикл информационной системы .....	<b>Ошибка! Закладка не определена.</b>
2.5 Основа технологий проектирования ИС.....	11
2.6 Технологии проектирования информационных систем .....	12
2.7 Технологии, основанные на структурном подходе (анализе) .....	13
2.8 Технологии, связанные с модельно-ориентированным подходом .....	14
2.8.1 Технология RAD (Rapid Application Development) .....	15
2.8.2 Технология описания процессов на основе методологии рационального унифицированного процесса RUP (Rational Unified Process) .....	15
2.8.3 Технология проектирования интегрированных информационных систем ARIS (ARchitecture of Integrated Information Systems).....	20
3. ЛАБОРАТОРНЫЙ ПРАКТИКУМ.....	22
3.1 ЛАБОРАТОРНАЯ РАБОТА № 1 .....	22
ТЕХНОЛОГИЯ ОПИСАНИЯ ПРОЦЕССОВ на основе методологии рационального унифицированного процесса RUP (Rational Unified Process) .....	22
3.2 ЛАБОРАТОРНАЯ РАБОТА № 2 .....	23
ИССЛЕДОВАНИЕ АСПЕКТОВ НАДЁЖНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В СИСТЕМНОЙ ИНЖЕНЕРИИ .....	23
3.3 ЛАБОРАТОРНАЯ РАБОТА № 3 .....	32
ИССЛЕДОВАНИЕ ТЕХНИЧЕСКИХ АСПЕКТОВ ПРОЕКТИРОВАНИЯ В СИСТЕМНОЙ ИНЖЕНЕРИИ .....	32
3.4 ЛАБОРАТОРНАЯ РАБОТА № 4 .....	36
ИССЛЕДОВАНИЕ АСПЕКТОВ КОЛИЧЕСТВЕННОГО АНАЛИЗА КОМПЛЕКСА ТЕХНИЧЕСКИХ СРЕДСТВ В СИСТЕМНОЙ ИНЖЕНЕРИИ.....	36
3.5 ЛАБОРАТОРНАЯ РАБОТА № 5 .....	39
ИССЛЕДОВАНИЕ ЭКОНОМИЧЕСКИХ АСПЕКТОВ АНАЛИЗА КОМПЛЕКСА ТЕХНИЧЕСКИХ СРЕДСТВ В СИСТЕМНОЙ ИНЖЕНЕРИИ.....	39
3.6 ЛАБОРАТОРНАЯ РАБОТА №6 ФОРМУЛИРОВКА И АНАЛИЗ ТРЕБОВАНИЙ К ИНФОРМАЦИОННОЙ СИСТЕМЕ.....	42
4.ИНСТРУКТИВНО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО КУРСОВОМУ ПРОЕКТИРОВАНИЮ.....	46
Список литературы и информационных ресурсов.....	57
Приложение 1. Варианты заданий к лабораторным работам .....	59
Приложение 2 .....	59
Приложение 3 .....	61
Приложение 4 .....	62

## **Введение**

Целью учебного пособия является помощь магистрантам направления подготовки 09.04.02 «Информационные системы и технологии» в организации эффективного изучения дисциплины «Системный анализ и проектирование информационных систем». Задача дисциплины состоит в формировании системного представления о современных тенденциях в области проектирования информационных систем.

Методические указания составлены в соответствии с требованиями, предъявляемыми к уровню знаний и формируемым компетенциям Федеральным государственным образовательным стандартом высшего образования по направлению подготовки 09.04.02 «Информационные системы и технологии» (уровень – магистратура) и профессиональным стандартом 5.141118 «Специалист по информационным системам».

Раздел по средствам и технологиям проектирования информационных систем разработан в сотрудничестве с И.В. Дымченко, О.А. Сырых.

Раздел по системному анализу информационных процессов и технологий разработан при участии доц. В.Ю. Калусова.

## 1.СИСТЕМНЫЙ АНАЛИЗ КАК МЕТОД СОЗДАНИЯ И ИССЛЕДОВАНИЯ СЛОЖНЫХ СИСТЕМ

Методологической основой системного анализа является сочетание анализа и синтеза, т.е. любая проблема, которую необходимо решить, вначале разделяется на более мелкие составляющие, они изучаются, вносятся какие-то изменения в состав, структуру (а это и есть анализ), а затем всё снова соединяют уже в другую систему (синтез). Первая система (до анализа), в которой данная ситуация воспринималась как проблема, называется проблемосодержащей системой, вторая (после синтеза) – в которой проблема исчезла, называется проблеморазрешающей. Так вот, анализ невозможен без синтеза, поскольку значение аналитического метода состоит не только и не столько в том, что сложное целое может быть разделено на все менее сложные части, а в том, что эти части, если соединены должным образом, снова образуют единое целое. Вместе с тем, и синтез невозможен без анализа, т.к. для объяснения функций частей целого необходимо разделение этого целого на эти части.

Отсюда следует два вывода:

- анализ и синтез в системных исследованиях неразделимы;
- как анализ, так и синтез не могут выполняться произвольно.

Система - это средство достижения цели или все то, что необходимо для достижения цели (элементы, отношения, структура, работа, ресурсы) в некотором заданном множестве объектов (операционной среде).

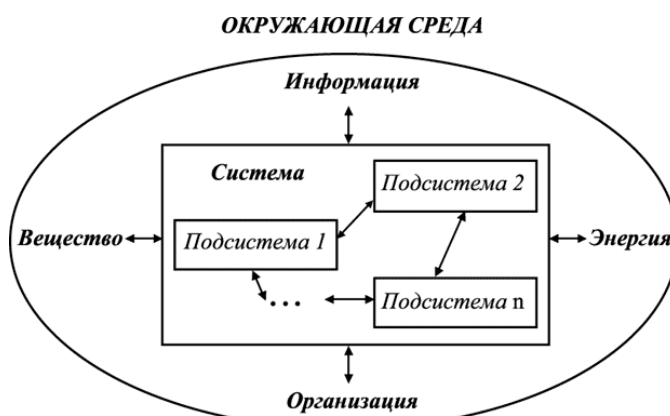


Рисунок 1- Обобщенная структура системы

Для описания системы важно знать, какие она имеет **структуру (строение), функции (работу) и связи (ресурсы) с окружением.**

Любая система имеет внутренние состояния, внутренний механизм преобразования входных данных в выходные (внутреннее описание), а также имеет внешние проявления (внешнее описание).

Внутреннее описание дает информацию о поведении системы, о соответствии (несоответствии) внутренней структуры системы целям, подсистемам (элементам) и ресурсам в системе, внешнее описание - о взаимоотношениях с другими системами, с целями и ресурсами других систем.

**Морфологическое (структурное или топологическое)** описание системы - это описание строения или структуры системы или описание совокупности  $A$  элементов этой системы и необходимого для достижения цели набора отношений  $R$  между этими элементами системы.

Функциональное описание системы - это описание законов функционирования, эволюции системы, алгоритмов ее поведения, "работы".

Информационное (информационно-логическое или инфологическое) описание системы - это описание информационных связей как системы с окружающей средой, так и подсистем системы.

Раньше информационное описание системы называли кибернетическим.

**Пример 1.** Рассмотрим систему "Информационный центр". Входная, выходная и внутрисистемная информация представляется документами, графическими, аудио- и видеофайлами, программами и т.д. Системные функции: предоставление машинного времени, обработка данных, поиск информации, создание и обработка архивов и баз данных. Системные цели: внедрение новых информационных технологий, внедрение новых методов обучения персонала и пользователей, повышение эффективности поиска, получения, обработки и хранения информации. Описание системы:  $x(t+1)=x(t)-a(t)x(t)+b(t)x(t)$ , где  $x(t)$  - эффективность методов работы с информацией в момент времени  $t$ ;  $a(t)$  - коэффициент компьютерной неграмотности пользователей;  $b(t)$  - коэффициент, показывающий степень внедрения новых аппаратно-программных средств.

**Пример 2.** Система "Корпоративная сеть",  $S = \langle A, B, R, V, Q \rangle$ ,  $A = \{\text{Терминал, Файловый Сервер, Почтовый Сервер, Концентратор, Маршрутизатор, Сетевой Принтер}\}$ ,  $B = \{\text{Рабочая станция, Серверная станция, Устройства передачи пакетов из одной подсети в другую}\}$ ,  $R = \{\text{Клиент, Сервер}\}$ .

С точки зрения морфологического описания, система может быть:

- гетерогенной системой - содержащей элементы разного типа, происхождения (подсистемы, не детализируемые на элементы с точки зрения выбранного подхода морфологического описания);
- гомогенной системой - т.е. содержать элементы только одного типа, происхождения;
- смешанной системой - с гетерогенными и гомогенными подсистемами.

Морфологическое описание системы зависит от учитываемых связей, их глубины (связи между главными подсистемами, между второстепенными подсистемами, между элементами), структуры (линейная, иерархическая, сетевая, матричная, смешанная), типа (прямая связь, обратная связь), характера (позитивная, негативная).

Основные признаки системы:

- целостность, связность или относительная независимость от среды и систем (наиболее существенная количественная характеристика системы). С исчезновением связности исчезает и система, хотя элементы системы и даже некоторые отношения между ними могут быть сохранены;

– наличие подсистем и связей между ними или наличие структуры системы (наиболее существенная качественная характеристика системы). С исчезновением подсистем или связей между ними может исчезнуть и сама система;

– возможность обособления или абстрагирования от окружающей среды, т.е. относительная обособленность от тех факторов среды, которые в достаточной мере не влияют на достижение цели;

– связи с окружающей средой по обмену ресурсами;

– подчиненность всей организации системы некоторой цели (как это, впрочем, следует из определения системы);

– эмерджентность или несводимость свойств системы к свойствам элементов.

Целое всегда есть система, а целостность всегда присуща системе, проявляясь в системе в виде симметрии, повторяемости (цикличности), адаптируемости и саморегуляции, наличии и сохранении инвариантов.

При системном анализе объектов, процессов, явлений необходимо пройти (в указанном порядке) следующие этапы системного анализа:

1. Обнаружение проблемы (задачи).
2. Оценка актуальности проблемы.
3. Формулировка целей, их приоритетов и проблем исследования.
4. Определение и уточнение ресурсов исследования.
5. Выделение системы (из окружающей среды) с помощью ресурсов.
6. Описание подсистем (вскрытие их структуры), их целостности (связей), элементов (вскрытие структуры системы), анализ взаимосвязей подсистем.
7. Построение (описание, формализация) структуры системы.
8. Установление (описание, формализация) функций системы и ее подсистем.
9. Согласование целей системы с целями подсистем.
10. Анализ (испытание) целостности системы.
11. Анализ и оценка эмерджентности системы.
12. Испытание, верификация системы (системной модели), ее функционирования.
13. Анализ обратных связей в результате испытаний системы.
14. Уточнение, корректировка результатов предыдущих пунктов.

## 2. ОБЩИЕ СВЕДЕНИЯ О ПРОЕКТИРОВАНИИ ИНФОРМАЦИОННЫХ СИСТЕМ

**Информационная технология**, согласно ГОСТ 34.003-90 [4] – это приемы, способы и методы применения средств вычислительной техники при выполнении функций сбора, хранения, обработки, передачи и использования данных.

Согласно ФЗ № 149-ФЗ [1] **информационная технология** – процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов.

А в соответствии ISO/IEC 38500:2015 [5], **информационная технология** – это ресурсы, необходимые для сбора, обработки, хранения и распространения информации.

**Проектирование информационной системы** – представляет сложный многоступенчатый вид деятельности – процесс преобразования входной информации об объекте проектирования, о методах проектирования и об опыте проектирования аналогичных объектов в проект ИС.

**Целью проектирования ИС** является подбор технического и формирование информационного, математического, программного и организационно-правового обеспечения. Документ, полученный в результате проектирования, носит название **проект** и представляет собой совокупность проектной документации, в которой представлено **описание проектных решений** по созданию и эксплуатации ИС. В реальных условиях **проектирование** – это поиск способа (выбор технологии) создания ИС, который удовлетворяет требованиям функциональности системы средствами имеющихся технологий с учетом заданных ограничений [7].

**Жизненный цикл (ЖЦ) информационной системы** – непрерывный процесс, начинающийся с момента принятия решения о создании информационной системы и заканчивающийся в момент полного изъятия ее из эксплуатации [8].

**Технология проектирования информационных систем** – совокупность технологических операций в их последовательности и взаимосвязи, приводящая к разработке проекта системы, при этом определяется совокупность методов (методология) и средств, направленных не только непосредственно на проектирование ИС, но и на организацию, управление, внедрение и модернизацию проекта ИС, то есть обеспечение всего жизненного цикла системы.

**Методология создания информационных систем** – описание процесса создания и сопровождения информационной системы в виде жизненного цикла, представление его как некоторой последовательности стадий и выполняемых на них процессов, а так же обеспечение управления этим процессом для того, чтобы гарантировать выполнение требований, как к самой системе, так и к характеристикам процесса разработки [6].



**Средства проектирования информационных систем** – комплекс инструментальных средств, обеспечивающих в рамках выбранной методологии проектирования поддержку полного жизненного цикла ИС.

## 2.1 Подходы к созданию информационных систем

Модели, применяемые на стадии конструирования, образуют метафору проектирования или подход к проектированию.

При проектировании информационных систем используют локальный или системный подходы.

Сущность **локального подхода** к проектированию состоит в последовательном наращивании задач, решаемых в системе. В этом случае проектирование информационной системы состоит из решения задач, ориентированных на удовлетворение потребностей конкретных подразделений или требований, связанных с реализацией конкретных условий. При этом данные организуют в отдельные логически структурированные файлы. Этот подход имеет серьезные недостатки:

- избыточность информации;
- противоречивость;
- недостаточная скорость обработки;
- отсутствие гибкости;
- низкая стандартизация программного обеспечения.

**Системный подход** [7], будучи общей методологической базой проектирования информационных систем, основан на концепции интеграции данных, которые описывают все сферы деятельности объекта информатизации. Этот подход предусматривает рассмотрение все элементов и составляющих процесса проектирования в их взаимосвязи, взаимозависимости и взаимном влиянии в интересах оптимального достижения как отдельных, так и общих целей создания информационной системы.

На современном этапе можно выделить четыре основных подхода к проектированию:

– **структурный подход** (**функционально-ориентированное проектирование**), который использует структурные методы для построения функциональной, информационной и других моделей информационной системы;

– **блочно-иерархический подход** к проектированию использует идеи декомпозиции сложных описаний объектов и соответственно средств их создания на иерархические уровни и аспекты, вводит понятие стиля проектирования (восходящее, нисходящее, смешанное), устанавливает связь между параметрами соседних иерархических уровней;

– **объектный подход** (**объектно-ориентированное проектирование**) предлагает набор объектных моделей для описания предметной области;

– *модельный подход (модельно-ориентированное проектирование)* основан на настройке и доработке типовой конфигурации информационной системы в среде специализированных инструментальных систем.

## **2.2 Стандартизация, комплексы стандартов на информационные системы и технологии**

Реальное применение любой технологии проектирования, разработки и сопровождения ИС в конкретной организации и конкретном проекте происходит при соблюдении всеми участниками проекта правил и соглашений, касающихся, рис.1.1:

- проектирования;
- оформления проектной документации;
- пользовательского интерфейса.

**Стандарт проектирования** должен устанавливать:

- набор необходимых моделей (диаграмм) на каждой стадии проектирования и степень их детализации;
- правила фиксации проектных решений на диаграммах, в том числе: правила именования объектов (включая соглашения по терминологии), набор атрибутов для всех объектов и правила их заполнения на каждой стадии, правила оформления диаграмм, включая требования к форме и размерам объектов, и т.д.;
- требования к конфигурации рабочих мест разработчиков, включая настройки операционной системы, настройки CASE-средств, общие настройки проекта и т.д.;
- механизм обеспечения совместной работы над проектом, в том числе: правила интеграции подсистем проекта, правила поддержания проекта в одинаковом для всех разработчиков состоянии (регламент обмена проектной информацией, механизм фиксации общих объектов и т.д.), правила проверки проектных решений на непротиворечивость и т.д.

**Стандарт оформления** проектной документации должен устанавливать:

- комплектность, состав и структуру документации на каждой стадии проектирования;
- требования к ее оформлению (включая требования к содержанию разделов, подразделов, пунктов, таблиц и т.д.);
- правила подготовки, рассмотрения, согласования и утверждения документации с указанием предельных сроков для каждой стадии;
- требования к настройке издательской системы, используемой в качестве встроенного средства подготовки документации;
- требования к настройке CASE-средств для обеспечения подготовки документации в соответствии с установленными требованиями.

Стандарты в области информационных технологий можно **классифицировать**:

- в зависимости от организации, утверждающей стандарты;

- по уровню утверждающей организации;
- по предмету стандартизации;
- по используемым методическим источникам.

В зависимости от *организации, утверждающей стандарты* [10]:

- официальные международные, официальные национальные или национальные ведомственные стандарты (например, стандарты ISO1, IEC2, ГОСТ);
- стандарты международных консорциумов и комитетов по стандартизации (например, стандарты OMG – Object Management Group (Группа по управлению объектами));
- стандарты «де-факто» – официально никем не утвержденные, но фактически действующие (например, стандартом «де-факто» долгое время были язык взаимодействия с реляционными базами данных SQL4);
- фирменные стандарты (например, Microsoft ODBC5, методика Oracle CDM).

**Объектами стандартизации** при проектировании информационных систем являются:

- процесс (что делать?): ISO12207 [8], ISO15288 [11], IEEE1220 [Ошибка! Источник ссылки не найден.];
- практика (как следует делать?): ISO15504 [12];
- источник (какие данные использовать): ISO10303 [Ошибка! Источник ссылки не найден.];
- описание: IDEF [Ошибка! Источник ссылки не найден.], UML [Ошибка! Источник ссылки не найден.], IEEE 1471 [Ошибка! Источник ссылки не найден.].

Стандартизация информационных технологий и систем повышает их прибыльность за счет снижения затрат на создание и особенно модификацию.

## 2.3 Основа технологий проектирования ИС

Основой технологий проектирования ИС кроме средства, является метод проектирования, который в общем случае включает совокупность трёх составляющих:

- 1) пошаговая процедура, определяющая последовательность технологических операций проектирования;
- 2) критерии и правила, используемые для оценки результатов выполнения технологических операций;
- 3) нотации (графические и текстовые средства), используемые для описания проектируемой системы.

Методы проектирования ИС можно классифицировать по:

**По подходу к автоматизации объекта** методы проектирования различают:

– **метод «снизу - вверх»** – разработка подсистем (процедур, функций), в то время когда проработка общей схемы не закончилась, то есть разработка ведется от отдельных задач ко всей системе;

– **метод «сверху - вниз»** – разработка начинается с определения целей решения проблемы, после чего идет последовательная детализация. При нисходящем проектировании задача анализируется с целью определения возможности разбиения ее на ряд подзадач. Затем каждая из полученных подзадач также анализируется для возможного разбиения на подзадачи. Процесс заканчивается, когда подзадачу невозможно или нецелесообразно далее разбивать на подзадачи;

– **принципы «дуализма» и многокомпонентности** – подход к проектированию ИС заключается в сбалансированном сочетании двух предыдущих.

**По степени автоматизации** методы проектирования разделяются на:

– **методы ручного проектирования**, при которых проектирование компонентов ИС осуществляется без использования специальных инструментальных программных средств, а программирование – на алгоритмических языках;

– **методы компьютерного проектирования**, при которых производится генерация или конфигурирование (настройка) проектных решений на основе использования специальных инструментальных программных средств.

**По степени использования типовых проектных решений** различают следующие методы проектирования:

– **типовое**, предполагающее конфигурирование ИС из готовых типовых проектных решений (программных модулей). Выполняется на основе опыта, полученного при разработке индивидуальных проектов.

– **оригинальное (индивидуальное)**, когда проектные решения разрабатываются «с нуля» в соответствии с требованиями к ИС.

**По степени адаптивности проектных решений** выделяют методы:

– **реконструкции**, когда адаптация проектных решений выполняется путем переработки соответствующих компонентов (перепрограммирования программных модулей);

– **параметризации**, когда проектные решения настраиваются (генерируются) в соответствии с изменяемыми параметрами;

– **реструктуризации модели**, когда изменяется модель проблемной области, на основе которой автоматически заново генерируются проектные решения.

## 2.4 Технологии проектирования информационных систем

Среди технологий проектирования ИС выделяют два основных класса:

- 1) каноническая технология;
- 2) индустриальная технология.

**Технология канонического проектирования** предполагает использование инструментальных средств универсальной компьютерной поддержки и предназначена для создания индивидуальных (оригинальных) проектов локальных ИС. При этом адаптация проектных решений возможна лишь путем перепрограммирования соответствующих программных модулей.

Организация канонического проектирования ИС ориентирована на использование главным образом каскадной модели жизненного цикла ИС. Стадии и этапы работы описаны в ГОСТ 34.601-90 [13].

**Технологии индустриального проектирования** использует инструментальные средства специальной компьютерной поддержки для разработки проектов сложных интегрированных (корпоративных) ИС.

Индустриальная технология проектирования, в свою очередь, разбивается на два подкласса:

- *типовое* (параметрически-ориентированное или модельно-ориентированное) проектирование.
- *автоматизированное проектирование* (использование CASE-технологий).

## 2.5 Технологии, основанные на структурном подходе (анализе)

Все наиболее распространенные технологии, основанные на методологии структурного подхода, базируются на ряде **общих принципов**. В качестве двух базовых принципов используются следующие:

- принцип «разделяй и властвуй» – принцип решения сложных проблем путем их разбиения на множество меньших независимых задач, легких для понимания и решения;
- принцип иерархического упорядочивания – принцип организации составных частей проблемы в иерархические древовидные структуры с добавлением новых деталей на каждом уровне.

Каждой группе средств соответствуют определенные виды моделей (диаграмм), наиболее распространенными среди которых, являются следующие:

- SADT (*Structured Analysis and Design Technique*) модели и соответствующие функциональные диаграммы;
  - DFD (*Data Flow Diagrams*) диаграммы потоков данных;
  - ERD (*Entity-Relationship Diagrams*) диаграммы "сущность-связь".
- Семейство IDEF включает следующие стандарты, рис.1.12 [18]:

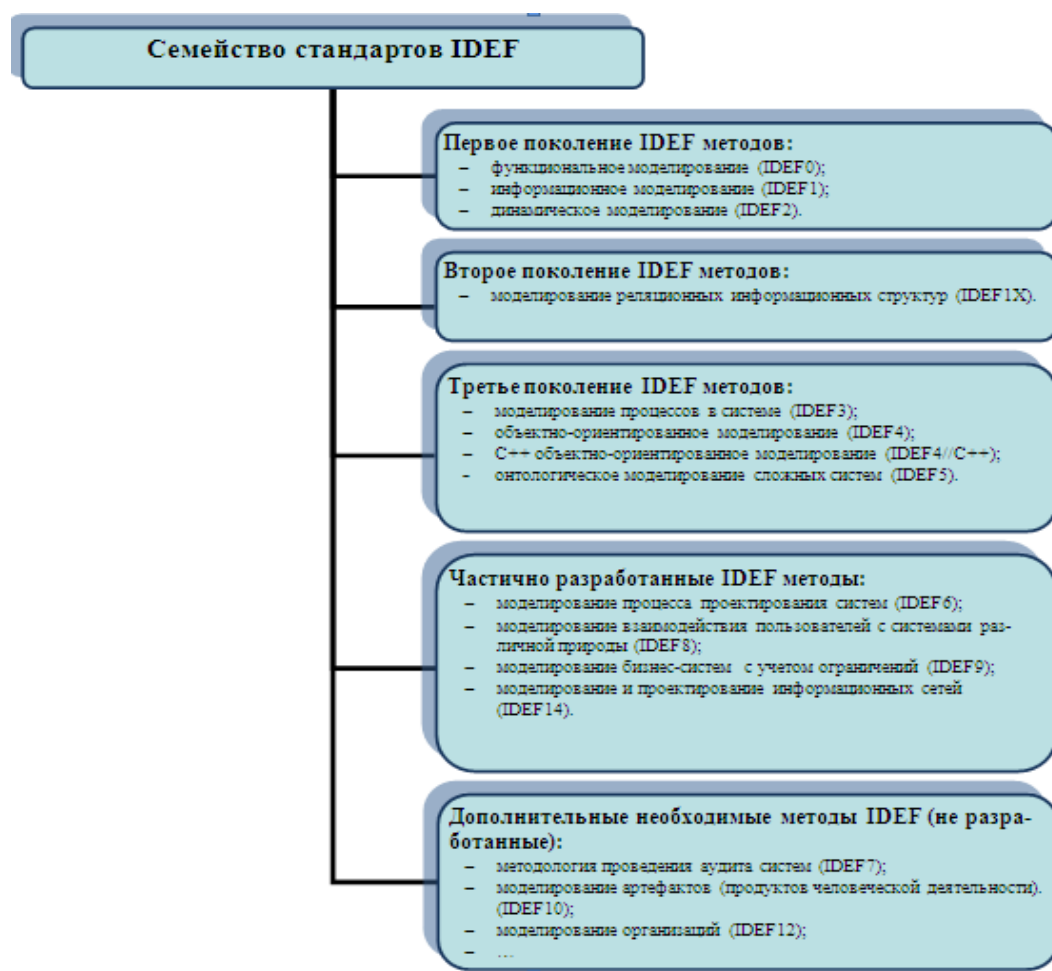


Рис.1.12. Семейство стандартов IDEF [18]

## 2.6 Технологии, связанные с модельно-ориентированным подходом

Модельно-ориентированное проектирование (МОП) – эффективный и экономически выгодный способ разработки систем управления и создания встраиваемых систем.

Типовая ИС в специальной базе метаинформации – *репозитории* – содержит модель объекта автоматизации, на основе которой осуществляется конфигурирование программного обеспечения. МОП ИС предполагает построение модели объекта автоматизации с использованием специального программного инструментария (например, SAP Business Engineering Workbench (BEW), BAAN Enterprise Modeler).

Преимущества МОП перед традиционным подходом проектирования:

- МОП предоставляет общую среду разработки, что способствует взаимодействию группы разработчиков в процессе анализа данных и проверки системы;

- инженеры могут найти и исправить ошибки на ранних стадиях проектирования системы, когда затраты времени и финансовые последствия изменения системы сводятся к минимуму;

– МОП способствует повторному использованию моделей для улучшения системы и создания производных систем с расширенными возможностями.

### **2.6.1 Технология RAD (Rapid Application Development)**

Технология проектирования ИС на основе методологии использования средств быстрой разработки приложений, получила широкое распространение и приобрела название методологии быстрой разработки приложений – RAD (Rapid Application Development) [10].

RAD – это комплекс специальных инструментальных средств быстрой разработки прикладных информационных систем, позволяющих оперировать с определенным набором графических объектов, функционально отображающих отдельные информационные компоненты приложений.

Основные принципы методологии RAD можно свести к следующему:

- используется итерационная (спиральная) модель разработки;
- полное завершение работ на каждом из этапов жизненного цикла не обязательно;
- в процессе разработки информационной системы необходимо тесное взаимодействие с заказчиком и будущими пользователями;
- необходимо применение CASE-средств и средств быстрой разработки приложений;
- необходимо применение средств управления конфигурацией, облегчающих внесение изменений в проект и сопровождение готовой системы;
- необходимо использование прототипов, позволяющее полнее выяснить и реализовать потребности конечного пользователя;
- тестирование и развитие проекта осуществляются одновременно с разработкой;
- разработка ведется немногочисленной и хорошо управляемой командой профессионалов;
- необходимы грамотное руководство разработкой системы, четкое планирование и контроль выполнения работ.

### **2.6.2 Технология описания процессов на основе методологии рационального унифицированного процесса RUP (Rational Unified Process)**

Еще одной ведущей методологией, в которой инструментально *поддерживаются все этапы жизненного цикла* разработки ПО, является методология Rational Unified Process (RUP).

В основе методологии RUP, как и многих других программных методологий, объединяющих инженерные методы создания ПО, лежит «пошаговый подход». Он определяет этапы жизненного цикла, контрольные

точки, правила работ для каждого этапа и, тем самым, упорядочивает проектирование и разработку ПО.

Для каждого этапа жизненного цикла методология задает:

- состав и последовательность работ, а также правила их выполнения;
- распределение полномочий среди участников проекта (роли);
- состав и шаблоны формируемых промежуточных и итоговых документов;
- порядок контроля и проверки качества.

Модели позволяют рассмотреть будущую систему, ее объекты и их взаимодействие еще до вкладывания значительных средств в разработку, позволяют увидеть ее глазами будущих пользователей снаружи и разработчиков изнутри еще до создания первой строки исходного кода. Большинство моделей представляются UML диаграммами [Ошибка! Источник ссылки не найден.].

**Определение требований.** Унифицированный процесс – это процесс, управляемый прецедентами, которые отражают сценарии взаимодействия пользователей. Фактически, это взгляд пользователей на программную систему снаружи. Таким образом, одним из важнейших этапов разработки, согласно RUP, будет этап определения требований, который заключается в сборе всех возможных пожеланий к работе системы, которые только могут прийти в голову пользователям и аналитикам.

Для облегчения этого процесса аналитики используют Диаграммы Прецедентов (Вариантов использования) (Use Case Diagram), рис.1.29. На диаграмме отображаются варианты использования (1) и действующие лица (2), между которыми устанавливаются следующие основные типы отношений: ассоциация между действующим лицом и вариантом использования (3); обобщение между действующими лицами (4); обобщение между вариантами использования (5); зависимости (различных типов) между вариантами использования (6).

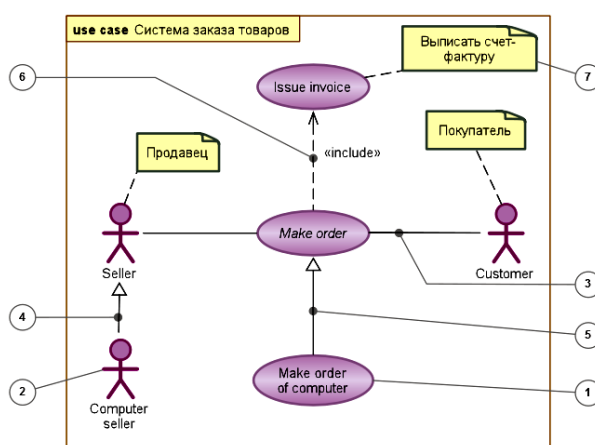


Рис.1.29. Диаграммы Прецедентов (Вариантов использования) (Use Case Diagram)



Для детализации конкретного прецедента используется Диаграмма Активности (Activity Diagram), пример которой дан на рис.1.30. На диаграмме применяют один основной тип сущностей – действие (1), и один тип отношений – переходы (2) (передачи управления и данных). Также используются такие конструкции как развилки, слияния, соединения, ветвления (3), которые похожи на сущности, но таковыми на самом деле не являются, а представляют собой графический способ изображения некоторых частных случаев многоместных отношений.

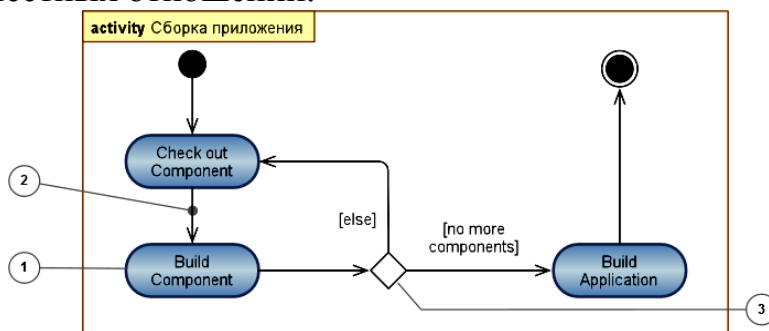


Рис.1.30. Диаграмма Активности (Activity Diagram)

**Анализ.** После определения требований и контекста, в котором будет работать система, наступает черед анализа полученных данных. Аналитическая модель – это взгляд на систему изнутри, в отличие от модели прецедентов, которая показывает, как система будет выглядеть снаружи.

Для отображения модели анализа при помощи UML используется Диаграмма Классов со стереотипами (образцами поведения) «граничный класс», «сущность», «управление», а для детализации используются Диаграммы Сотрудничества (Collaboration), рис.1.31.

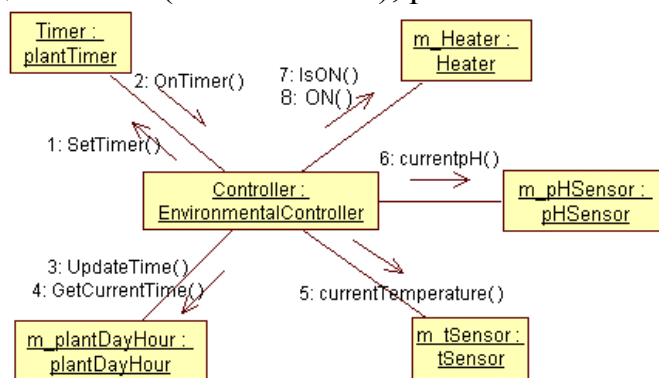


Рис.1.31. Диаграммы Сотрудничества (Collaboration)

Если акцентировать внимание на порядке взаимодействия, то другим его представлением будет Диаграмма Последовательности (Sequence Diagram), рис.1.32.

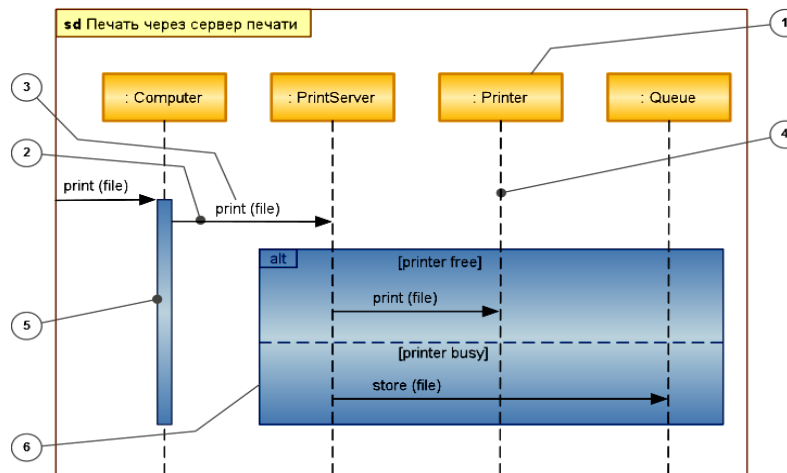


Рис.1.32. Диаграмма Последовательности (Sequence Diagram)

Эта диаграмма позволяет взглянуть на обмен сообщениями во времени, наглядно отобразить последовательность процесса. На диаграмме применяют один основной тип сущностей – экземпляры взаимодействующих классификаторов (1) (в основном классов, компонентов и действующих лиц), и один тип отношений – связи (2), по которым происходит обмен сообщениями (3).

**Проектирование** в ходе которого на основании моделей, созданных ранее, создается модель проектирования. Для создания модели проектирования используются целый набор UML диаграмм: Диаграммы Активности, Диаграммы Классов, Диаграммы Коммуникации, Диаграммы Взаимодействия.

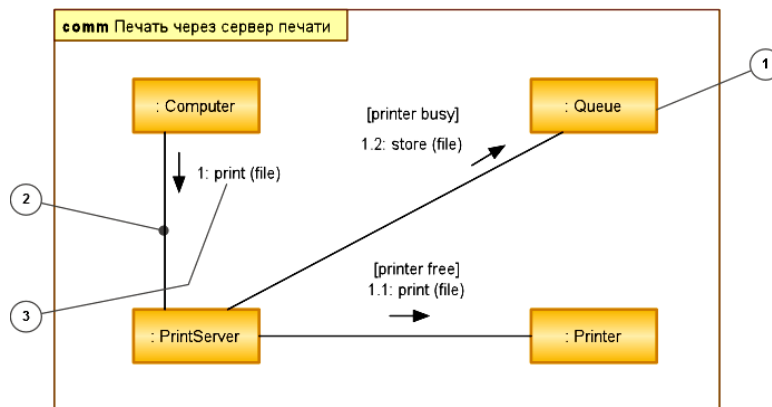


Рис.1.33. Диаграммы Коммуникации (Communication Diagram).

Дополнительно в этом рабочем процессе может создаваться модель развертывания, которая реализуется на основе Диаграммы Развертывания (Deployment Diagram), рис.1.35. Это самый простой тип диаграмм, предназначенный для моделирования распределения устройств в сети. На диаграмме присутствует два типа сущностей: артефакт (1), который является реализацией компонента (2) и узел (3) (может быть как классификатор, описывающий тип узла, так и конкретный экземпляр), а также отношение ассоциации между узлами (4), показывающее, что узлы физически связаны во время выполнения. Если одна сущность является частью другой, применяется

либо отношение зависимости «deploy» (5), либо фигура одной сущности помещается внутрь фигуры другой сущности (6).

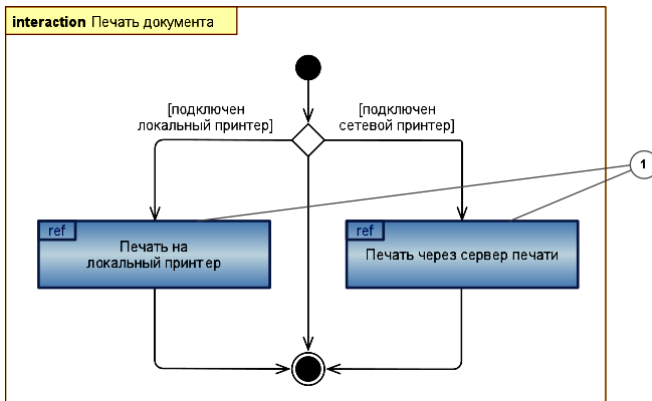


Рис.1.34. Диаграммы Взаимодействия.

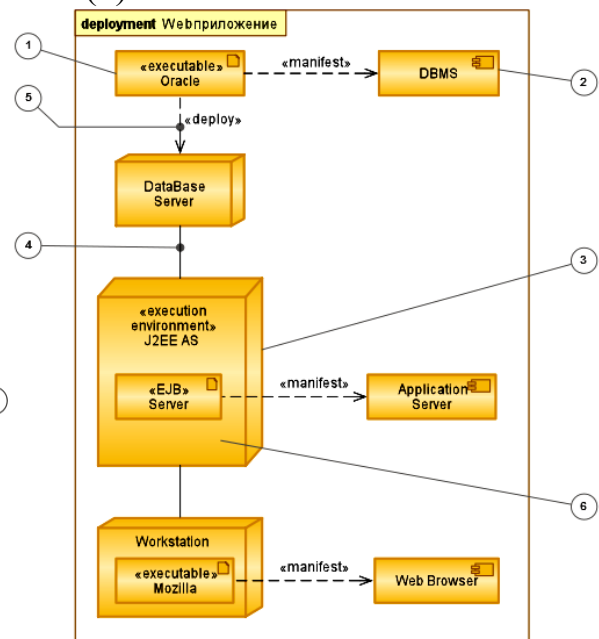


Рис.1.35. Диаграммы Развертывания

**Реализация.** Основная задача процесса реализации – создание системы в виде компонентов – исходных текстов программ, сценариев, двоичных файлов, исполняемых модулей и т.д. Данная модель описывает способ организации этих компонентов в соответствии с механизмами структурирования и разбиения на модули, принятыми в выбранной среде программирования и представляется Диаграммой Компонентов (Component Diagram), рис.1.36.

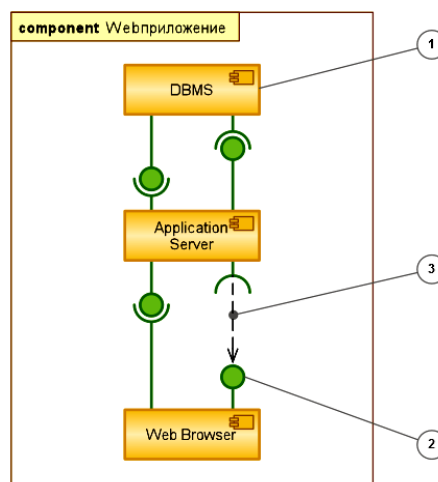


Рис.1.36. Диаграммы Компонентов (Component Diagram).

Основной тип сущностей на диаграмме компонентов – это сами компоненты (1), а также интерфейсы (2), посредством которых указывается взаимосвязь между компонентами. На диаграмме компонентов применяются следующие отношения:

- реализации между компонентами и интерфейсами (компонент реализует интерфейс);
- зависимости между компонентами и интерфейсами (компонент использует интерфейс) (3).

RUP довольно обширен, и содержит рекомендации по ведению различных программных проектов, от создания программ группой разработчиков в несколько человек, до распределенных программных проектов, объединяющих тысячи человек на разных континентах.

### **2.6.3 Технология проектирования интегрированных информационных систем ARIS (ARchitecture of Integrated Information Systems)**

Концепция Архитектуры Интегрированных Информационных Систем - ARIS (ARchitecture of Integrated Information Systems) разработана профессором А.В. Шеером (Scheer).

Эта концепция имеет два основных преимущества:

- позволяет выбрать методы и интегрировать их, опираясь на основные особенности моделируемого объекта;
- служит базой для управления сложными проектами, поскольку благодаря структурным элементам содержит встроенные модели процедур для разработки интегрированных информационных систем.

Архитектура ARIS явилась основой ARIS Toolset – инструментальной среды, разработанной компанией IDS Scheer AG.

В методологии ARIS [22] для описания различных подсистем организации используется более ста типов моделей, отражающих различные аспекты деятельности и реализующих различные методы моделирования (в том числе событийная цепочка процесса EPC (Event driven Process Chain):

- модель «сущность-связь» ERM (Entity Relationship Model);
- модели методики объектно-ориентированного моделирования OMT (Object Modeling Technique);
- модели BSC (Balanced Scorecard) – система сбалансированных показателей;
- модели UML и многие другие.

Все многообразие типов моделей ARIS подразделяется на пять видов описания в соответствии с основными подсистемами предприятия:

- организационной;
- функциональной,
- подсистемой данных;
- подсистемой процессов;
- подсистемой продуктов/услуг,

Остальные подсистемы могут моделироваться с использованием типов объектов, входящих в перечисленные виды описания.

На рисунке 1.37 показана системная организация и взаимодействие принципов, стандартов и подходов к проектированию ИС



Рисунок 1.37. Взаимодействие принципов, стандартов и подходов к проектированию ИС

### 3. ЛАБОРАТОРНЫЙ ПРАКТИКУМ

**Цель проведения лабораторных занятий:** закрепление теоретических знаний, полученных на лекционных занятиях и привитие навыков самостоятельной работы с различными аспектами системной инженерии.

#### 3.1 ЛАБОРАТОРНАЯ РАБОТА № 1

#### **ТЕХНОЛОГИЯ ОПИСАНИЯ ПРОЦЕССОВ на основе методологии рационального унифицированного процесса RUP (Rational Unified Process)**

**Цель:** изучить автоматизированные средства моделирования рационального унифицированного процесса RUP .

**Время:** 4 часа

#### **Порядок выполнения лабораторной работы**

1. Исследовать доступный функционал для методологии описания процесса создания ИС на основе методологии рационального унифицированного процесса RUP (Rational Unified Process).
2. Выбрать вариант задания по порядковому номеру в списке группы.
3. Осуществить построение возможных для заданной предметной области диаграмм (**обязательны к построению диаграммы: Прецедентов, Активности, Развертывания**).
4. Сформировать отчет.

#### **Содержание отчета №1 по лабораторной работе**

1. Комплекс диаграмм, созданных с помощью методологии рационального унифицированного процесса RUP.
2. Все возможные доступные отчеты по RUP-диаграммам.
3. Выводы о применимости технологии для реализации НИР.

#### **Контрольные вопросы:**

1. К какому направлению создания ИС относится методология RUP?
2. Перечислить основные типы диаграмм, относящихся к RUP.
3. Пояснить содержание Диаграммы Коммуникации.
4. Может ли методология RUP применяться к корпоративным и интегрированным ИС?

### 3.2 ЛАБОРАТОРНАЯ РАБОТА № 2

## ИССЛЕДОВАНИЕ АСПЕКТОВ НАДЁЖНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В СИСТЕМНОЙ ИНЖЕНЕРИИ

#### 1. Цель работы:

- Получить практические навыки в построении моделей отказов технических и программных средств в среде MatLab Simulink.
- Оценить надёжность программного обеспечения (ПО) с использованием моделей Нельсона и Коркорена.

#### Теоретические сведения

Базовой составной частью информационных систем (ИС) является ПО, в связи с чем изучение подходов для создания качественного ПО представляется важным для системно инженерного взгляда на ИС.

Успешное решение задачи создания надёжного ПО позволяет избежать негативных последствий и затратных модификаций на этапе эксплуатации. По определению [1], программа обладает свойством надёжности в той мере, в которой она способна удовлетворять выполнению функции, для которых она предназначена. Под “удовлетворительно” понимается отсутствие изъянов, для устранения которых необходимо внести изменения в тексты программных модулей.

Для точного определения надёжности в математическом смысле, используют термины статистики. В частности, Нельсоном предложено следующее [2].

1. Программа представляется как некоторая вычисляемая функция  $F$  на множестве  $E$  наборов входных данных. Элемент этого множества  $E_i$  представляет собой совокупность значений данных, необходимых для **однократного** выполнения программы.

$$E = \{E_i : i = 1, 2, \dots, N\}.$$

2. Совокупность действий, включающая ввод комбинации данных  $E_i$ , выполнение программы, завершаемое получением результата  $F(E_i)$ , называется **прогоном программы** или просто **прогоном**.

3. В ходе выполнения прогона происходит определение значения функции  $F(E_i)$  для каждого набора  $E_i$ . Наличие отклонений от спецификации, вольных или невольных, приводит к отклонению  $F(E_i)$  от требуемого. Иногда это отклонение находится в допустимых пределах

$$|F^V(E_i) - F(E_i)| \leq \delta_i$$

и остаётся незамеченным. В противном случае результат вычисления признаётся неприемлемым, и фиксируется так называемый “**рабочий отказ**”.

В результате последовательности прогонов может быть образовано (определено) множество наборов  $E_{Er}$  данных, приводящих к рабочим отказам.

Таким образом, вероятность безотказного функционирования (надёжности) ПО составит

$$R = 1 - P_{отк} = 1 - \frac{N_{Er}}{N}, \quad (1)$$

где  $N$  и  $N_{Er}$  – мощности множеств  $E$  и  $E_{Er}$  соответственно. Формула (1) верна в том случае, когда выбор набора входных данных осуществляется равновероятно. Так же использованию (1) препятствует невозможность исчерпывающего тестирования ПО.

На практике, выбор определённого теста определяется режимами работы ПО, которые необходимо проверить и связан с **распределением вероятностей**  $P_i = P(E_i)$ , множество которых называется **функциональным разрезом**. В этом случае, вероятность того, что прогон программы на множестве входных данных  $E$  закончится отказом, составит

$$P_{отк} = \sum_{i=1}^N q_i \cdot p_i,$$

где  $q_i = \{0, 1\}$  – “динамическая” переменная, единичное значение которой соответствует рабочему отказу. Отсюда, вероятность безотказной работы

$$R = 1 - P_{отк} = \sum_{i=1}^N q_i \cdot (1 - p_i)$$

Поскольку в каждом прогоне используется свой функциональный разрез, а число прогонов ограничено, надёжность ПО будет определяться вероятностью того, что ни один из последовательности из  $n$  прогонов на различных функциональных разрезах не закончится отказом

$$R_n = \prod_{j=1}^n (1 - P_{отк(j)}) = e^{\sum_{j=1}^n \ln(1 - P_{отк(j)})} = e^{-\sum_{j=1}^n P_{отк(j)}}, \quad (2)$$

где  $j$  – номер прогона, а  $P_{отк(j)}$  – вероятность отказа на этом прогоне.

Модель Нельсона использует гипотезу о том, что ошибки фиксируются в ходе прогонов программы, но не исправляются.

Для оценки надёжности функционирования ПО также применяется модель Коркорэна [3], которая учитывает динамику его отладки:

$$R_n = \frac{1}{n} \left( n_0 + \sum_{j=1}^T k_j (n_j - 1) \right), \quad (3)$$

где  $n_0$  – число успешных испытаний в серии из  $n$  прогонов программы;  $T$  – число типов ошибок, по которым ведётся учёт;  $n_j$  – число ошибок  $j$ -го типа, выявленных за  $n$  прогонов;  $k_j$  – масштабирующий коэффициент, определяемый ступенчатой функцией

$$k_j = \begin{cases} q_j, & n_j > 0, \\ 0, & n_j = 0 \end{cases} \quad (4)$$

где  $q_j$  – вероятность устранения ошибки  $j$ -го типа.

## Принципы моделирования процесса отладки ПО

С точки зрения теории вероятностей, обнаружение ошибки в процессе прогона является составным событием, комбинирующим два элементарных



события:

- выбор из множества  $E$  набора  $E_i$  входных данных в соответствии с функциональным разрезом исследуемой программы;
- проявление рабочего отказа на выбранном наборе.

Поэтому для имитационного моделирования понадобятся два независимых источника (генератора) псевдослучайных чисел с различными законами распределения: распределение 1, имитирующее функциональный разрез, и распределение 2, моделирующее появление отказа. Псевдослучайные числа сравниваются между собой, и, в зависимости от результатов сравнения, генерируется сигнал рабочего отказа.

Схема моделирования представлена на рисунке 1 ниже. Остановимся на описании блоков MatLab Simulink, входящих в её состав [5].

Генерацию псевдослучайных чисел будем осуществлять с помощью блоков MatLab Simulink разделов конструктора Library: Sources/Constant и Library: NonLinear/MatLab function. Блоки в рабочее поле модели помещаются “перетаскиванием” из соответствующих разделов библиотеки конструкций.

Двойной щелчок на элементе открывает вкладку его свойств. На вкладке свойств константы есть поле для приёма её значения (которое после ввода отобразится на схеме). На вкладке MatLab function – поле для приёма арифметического выражения, которое может иметь довольно сложную структуру. Для блока Raspred1 использована запись вида `exprnd(u,1,1)`, в которой обозначено: `exprnd` – функция генерирующая ряд чисел, отвечающих экспоненциальному распределению, `u` – принятое в стандарте MatLab Simulink обозначения входного сигнала блока, а две единицы – размер генерируемой матрицы (поскольку нас интересует одно число за цикл моделирования).

PROGRAM MISTAKE MODEL

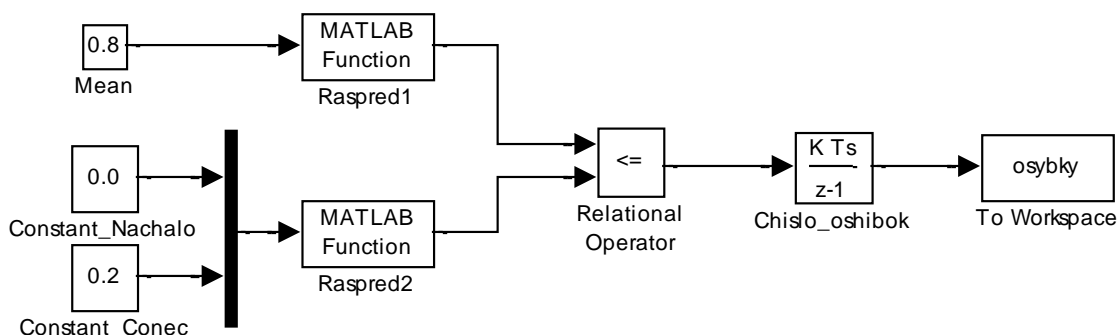


Рисунок 1 – Имитационная модель процесса тестирования

Для блока Raspred2 – `unifrnd(u(1),u(2),1,1)`, где – функция генерирующая равномерно распределенные значения в интервале от  $u(1)$  до  $u(2)$ . Чтобы сгруппировать входные значения блока в вектор используется элемент Library: Connection/Mux, имеющий единственный параметр –

число входов. Порядковый номер элемента в векторе определяется номером входа  $u_{ix}$ . Поскольку сам по себе элемент  $u_{ix}$  мало информативен, его графическое изображение “ужимают” манипуляторами до размера жирной линии. При этом название блока пропадает.

Для решения часто возникающей проблемы с генераторами случайных чисел, целесообразно использовать следующее описание:

```
%Raspred1
function y = fcn(u)
coder.extrinsic('exprnd','rng');
y = 0;
persistent atTime0
if isempty(atTime0)
rng('shuffle');
atTime0 = false;
end
y = exprnd(u,1,1);
```

```
%Raspred2
function y = fcn(u)
coder.extrinsic('unifrnd','rng');
y = 0;
persistent atTime0
if isempty(atTime0)
rng('shuffle');
atTime0 = false;
end
y = unifrnd(u(1),u(2),1,1);
```

Фрагменты кода перед генерацией случайных чисел функциями `unifrnd` и `exprnd`, позволяют производить инициализацию генераторов случайных чисел случайным образом, что позволяет произвести опыт и получить результаты. Данная проблема достаточно часто встречается согласно иностранным форумам и варьируется от версии к версии Matlab и Simulink.

Блок раздела конструктора моделей Library: NonLinear / Relational Operator. Проверяет отношение, указанное на блоке, и, в случае его выполнения, генерирует сигнал “1”, в противном случае “0”. Отношение может быть изменено на любое из известных в математике, на вкладке свойств блока в выпадающем списке. Изменение отразится на изображении блока.

Блок Library: Discrete / Discrete-Time Integrator используется для подсчёта единичных импульсов, поступающих от блока сравнения. В блоке выполняется интегрирование по методу прямоугольников, и, тем самым определяется число ошибок.

Блок Library: Sinks / To Workspace используется для связи с программой на внутреннем языке MatLab, с целью последующей обработки. Связь осуществляется по идентификатору, задаваемом на вкладке свойств и отображаемом на изображении блока, в нашем случае употреблено имя `osybky`.

Имитационная модель может запускаться как в “ручном”, так и в “автоматическом” режимах. Перед началом запуска, в любом случае, на вкладке меню Simulations / Parameters задайте значения полей Stop Time, в поле Type установите значение Fixed-step, а в поле рядом discrete (no continuous states).

В поле Stop Time можно установить значение 100, которое достаточно для набора статистики по математическому ожиданию и удобно для расчётов.

Для однократного “автоматического” запуска модели необходимо использовать конструкцию MatLab `sim('PO_imit')`, и чтобы файл с моделью находился в одной директории с программой, осуществляющей запуск и обработку.

Доступ к выходным данным модели производится по ссылке, в нашем случае это `osybyk.signals.values`.

Каждый из запусков модели интерпретируется как испытание по одной из функций разреза Stop Time наборами данных.

В частности, для экспоненциального распределения Raspred1 со значением среднего 0.8 и равномерного распределения Raspred2 в интервале (0.0, 0.2) получена следующая диаграмма.

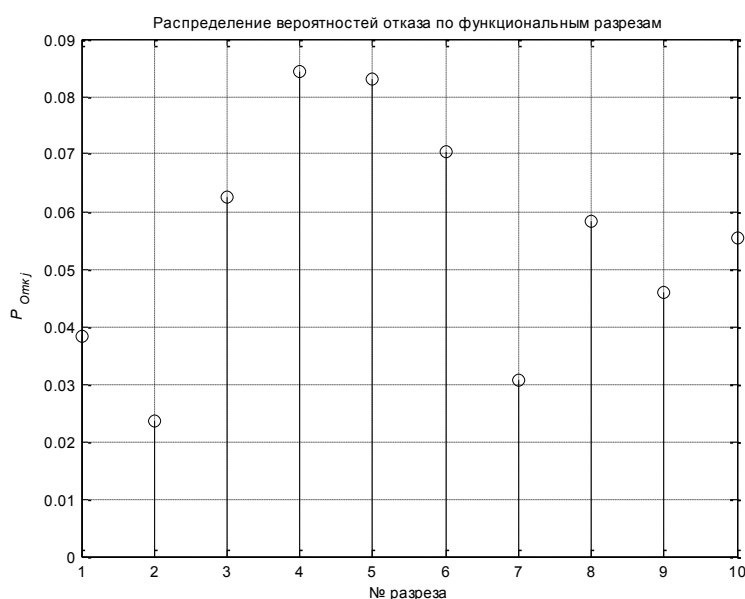


Рисунок 2 – Распределение оценок вероятностей отказов

Из зависимостей следует, что все функциональные модули содержат ошибки и нуждаются в доработке, наихудшим образом дело обстоит для модулей №№ 4 – 6. Если изменения в ПО не вносить, то вероятность безотказной работы, рассчитанная, согласно (2) будет убывать, как это видно на рисунке, приведённом ниже.

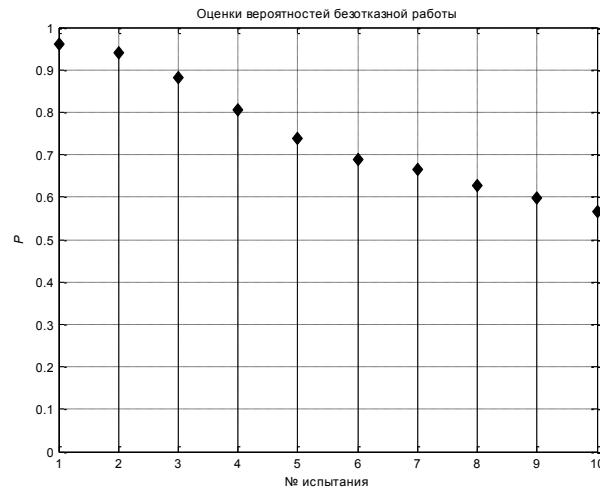


Рисунок 3 – Оценка надёжности в функции от номера прогона

Зависимость демонстрирует вполне очевидные вещи: если ошибки не устранять, то надёжность ПО будет низкой. Сводка по статистике моделирования и оценка Коркорэна при внесении исправлений по ходу тестирования ПО, рассчитанная по формуле (3), в предположении об устранении всех обнаруживаемых ошибок при модификации программного кода и отсутствии привносимых ошибок представлена на рисунке 3.

## РЕЗУЛЬТАТЫ

Число функциональных разрезов 10  
Число испытаний 1000  
Оценка Коркорэна 0.99

Оценка надёжности Нельсона 0.594851

Рисунок 4 – Сводный отчёт по работе программы

## ХОД РАБОТЫ

1. Получить вариант задания в виде типов распределений и их параметров. С учётом параметров, необходимых для генерации распределения соответствующего типа, построить схему моделирования, аналогичную представленной на рисунке 1. Некоторые процедуры генерации, используемые в MatLab для решения задач в лабораторной работе, приведены в таблице 1.

Таблица 1 – Встроенные процедуры генерации псевдослучайных чисел

Тип распределения	Параметры вызова
Нормальное	<code>normrnd(&lt;среднее&gt;, &lt;СКО&gt;, 1, 1)</code>
Пуассоновское	<code>poissrnd(&lt;среднее&gt;, 1, 1)</code>
Равномерное	<code>unifrnd(&lt;нижн_гр&gt;, &lt;верх_гр.&gt;, 1, 1)</code>
Хи-квадрат	<code>chi2rnd(&lt;среднее&gt;, 1, 1)</code>
Экспоненциальное	<code>exprnd(&lt;среднее&gt;, 1, 1)</code>

2. На вкладках меню Simulations / Parameters задать значения полей: Stop Time установить равным 100.0, в поле Type установить значение Fixed-step, а в поле рядом discrete (no continuous states).

3. Написать программу для управления имитационной моделью.

Модель необходимо запускать в цикле, число повторений которого npusk равно количеству функциональных разрезов.

Для решения этой задачи понадобятся следующие языковые средства и конструкции среды MatLab:

- for k=1 : npusk; ... end; – конструкция цикла с заданным числом повторений;
- sim('PO\_imit'); – запуск модели с именем, заключённым в апострофы;
- osybky.signals.values) – обеспечивает доступ к результатам работы имитационной модели, полученных посредством блока To Workspace.

4. Дополнить программу управления инструкциями для расчётов показателей надёжности (2) и (3)

Можно использовать:

- mean(x) – возвращает среднее значение массива чисел. Эту процедуру необходимо выполнять в цикле моделирования с каждым текущим вектором osybky.signals.values.
- cumprod(x) – вычисление накопленного произведения элементов вектора x, позволит изящно рассчитать последовательность  $R_n$  согласно (2);
- sum(x) – вычисление суммы вектора x.

При расчёте по формуле (3) необходимо учесть, что  $n$  определяется произведением Stop Time на число пусков имитационной модели npusk. Параметр  $n_0$  – есть разность между  $n$  и суммой элементов вектора усреднённых ошибок, полученных от имитационной модели. Также принимается гипотеза, что найденные ошибки гарантированно устраняются.

5. Дополнить программу управления операторами для построения графиков [6];

- stem(x,y,<параметры линии>) – позволит вывести график дискретных значений в виде набора вертикальных линий, например, операторы stem(1:npusk,p\_otk,'k'); – использован для вывода рисунка 1,  
stem(1:npusk,r\_n,'MarkerFaceColor','black','Marker','diamond') – для рисунка 2;
- title('Текст'); – используется для вывода названия графика;
- xlabel('Текст'); – для подписывания оси абсцисс;
- ylabel('Текст'); – для подписывания оси ординат;

- `grid`; – формирует сетку на экране;
- `hold` – фиксирует текущее изображение и позволяет пополнять его новыми элементами.

Управление шрифтами, цветом et c., смотри справочную и иную литературу среде MatLab.

6. Построить окно итогового отчёта.

Указанная процедура описана в [6] и включает ряд шагов.

- `axis('off');` – уборка из графического окна подокна с осями координат;
- `h1=text(<гор>,<вер>,'Текст','FontSize',20);` – вывод текста в позицию (<гор>,<вер>) окна кеглем № 20;
- `h1=text(<гор>,<вер>,sprintf('R=%g',r_c),'FontSize',12);` – вывод текста и цифровых значений, формат вывода аналогичен форматам языка Си.

7. Оформить отчёт.

## Контрольные вопросы

1. Что такое “прогон”?
2. Как вы понимаете термин “рабочий отказ”?
3. Как вы объясните, что означает термин “функциональный разрез”?
4. Почему для моделирования отказа ПО нам нужны два генератора псевдослучайных чисел?
5. Поясните назначение отдельных блоков схемы моделирования
6. Каким образом осуществляется связь имитационной модели и обрабатывающей программы в MatLab?
7. В чём сходство и различие в оценке надёжности ПО для моделей Нельсона и Коркорэна?
8. Какие еще модели надёжности ПО Вам известны [4]?
9. Какой вид, по Вашему мнению, может иметь распределение вероятностей ошибок по функциональному разрезу?

## Варианты заданий

Вариант задания выбирается как остаток от деления двух последних цифр зачётной книжки по модулю 25. Параметры распределения согласовываются с преподавателем.

№	Тип распределения 1	Тип распределения 2
0	Нормальное	Нормальное
1	Нормальное	Пуассоновское
2	Нормальное	Равномерное
3	Нормальное	Хи-квадрат

№	Тип распределения 1	Тип распределения 2
4	Нормальное	Экспоненциальное
5	Пуассоновское	Нормальное
6	Пуассоновское	Пуассоновское
7	Пуассоновское	Равномерное
8	Пуассоновское	Хи-квадрат
9	Пуассоновское	Экспоненциальное
10	Равномерное	Нормальное
11	Равномерное	Пуассоновское
12	Равномерное	Равномерное
13	Равномерное	Хи-квадрат
14	Равномерное	Экспоненциальное
15	Хи-квадрат	Нормальное
16	Хи-квадрат	Пуассоновское
17	Хи-квадрат	Равномерное
18	Хи-квадрат	Хи-квадрат
19	Хи-квадрат	Экспоненциальное
20	Экспоненциальное	Нормальное
21	Экспоненциальное	Пуассоновское
22	Экспоненциальное	Равномерное
23	Экспоненциальное	Хи-квадрат
24	Экспоненциальное	Экспоненциальное

### Библиографический список

1. Бозм Б.В. Характеристики качества программного обеспечения [Текст]: / Б.В. Бозм. – М.: Мир, 1981.
2. Nelson E.C. Statistical Basic for Software Reliability Assessment, TRW-SS-73-03, March 1973.
3. Corcoran W.J. Estimating Reliability after Corective Action / W.J. Corcoran, H. Weingarten, P.W. Zehna. Management Science, 10, No 4, p.787 – 795 (July 1954).
4. Тейер Т. Надёжность программного обеспечения. / Т. Тейер, М. Липов, Э. Нельсон. – М.: Мир, 1981.
5. Гульяев А.К. : MatLab 5.2: имитационное моделирование в среде Windows [Текст]: / А.К. Гульяев СПб.: КОРОНА принт, 1999. – 288 с.
6. Лазарев Ю.Ф. MatLab 5.x. [Текст]: / Ю.Ф. Лазарев. – К.: Издательская группа BHV, 2000. – 384 с.

### 3.3 ЛАБОРАТОРНАЯ РАБОТА № 3

## ИССЛЕДОВАНИЕ ТЕХНИЧЕСКИХ АСПЕКТОВ ПРОЕКТИРОВАНИЯ В СИСТЕМНОЙ ИНЖЕНЕРИИ

#### Цель работы

- Получить практические навыки формирования комплекса технических средств.
- Получить практические навыки в проведении количественного анализа вычислительных систем.

#### Теоретические сведения

При решении задачи структурного синтеза в системной инженерии целесообразно уметь выбирать и оценивать средства ввода и регистрации, обработки, передачи данных, т.е. решать проблему обеспечения процессов сбора, обработки, передачи данных в ИС.

#### Расчётные выражения для выбора средств ввода и регистрации

При выборе средств ввода и регистрации данных используются следующие оценки.

Общее количество средств регистрации:

$$N_{\text{Общ}} = \sum_{i=1}^{n_p} N_i \cdot K_H, \quad (1)$$

где  $n_p$  – количество пунктов регистрации данных;  $N_i$  – количество средств, установленных в  $i$ -м пункте;  $K_H$  – коэффициент, который учитывает неравномерность поступления данных. Этот коэффициент определяется так.

$$K_H = \frac{t_{CM}}{t_{CM} - t_O}, \quad (2)$$

где  $t_{CM}$  – длительность смены (нахождения терминала во включённом состоянии в течении какого-либо фиксированного кванта времени),  $t_O$  – среднее время отсутствия данных в течение времени  $t_{CM}$ .

Для определения количества средств  $N_i$ , используется выражение

$$N_i = \sum_{j=1}^e \frac{M_{pj}(1 + K_{KH})}{B_{i,j} \cdot t_{\text{Доп}j}}, \quad (3)$$

где  $e$  – число видов регистрируемых данных;  $M_{pj}$  – объём массива регистрируемой информации;  $K_{KH} - B_{i,j}$  – нормативная характеристика производительности  $i$ -м устройством  $j$ -ых типов данных;  $t_{\text{Доп}j}$  – допустимое время приёма  $j$ -ого типа данных.

Выбранный состав средств должен удовлетворять условию технологического цикла регистрации данных вида

$$\Pi_P \leq \Pi_{\text{ФСВ}} - \Pi_{\text{ОБ}} - T_{\text{ПИ}}, \quad (4)$$



где  $\Pi_{\Phi CY}$  – цикл функционирования системы управления;  $\Pi_{OB}$  – цикл обработки данных;  $T_{\Pi}$  – время передачи информации.

Цикл работ по регистрации данных  $\Pi_p$ , в свою очередь, есть сумма времён, определяемых функцией от продолжительности работы алгоритмов, составляющих технологический цикл.

$$\Pi_p = \sum f(F_j, R_j, A_j, D_j), \quad (5)$$

где  $n$  – число выполняемых алгоритмов;  $F_j$  – время работы алгоритма формирования (фиксации) данных;  $R_j$  – время работы алгоритма регистрации (внесения) данных;  $A_j$  – время работы алгоритма накопления (аккумулирования) данных;  $D_j$  – время работы алгоритма задержки времени перед передачей на сервер. Указанные алгоритмы могут выполняться как программно, так и вручную.

При ручном способе формирования и внесения первичных данных

$$\Pi_p = \sum_{j=1}^e \left( \frac{F_j}{N_F} + \frac{R_j}{N_R} + \frac{A_j}{N_A} + D \right), \quad (6)$$

где  $N_F$  – количество источников данных;  $N_R$  – количество пунктов регистрации (удалённых терминалов);  $N_A$  – количество пунктов накопления информации (локальных станций). В выражении (6) первые два слагаемых определяют производительность труда персонала или устройств, а вторые два – цикл функционирования системы в целом.

При комбинированном способе формирования данных получаем следующее

$$\Pi_p = \sum_{j=1}^e \left( \frac{F_j}{N_F} + \frac{R_j + A_j}{N_{R,A}} + D \right). \quad (7)$$

Если данные формируются в полностью автоматическом режиме, то имеем

$$\Pi_p = \sum_{j=1}^e \left( \frac{F_j + R_j}{N_{F,A}} + \frac{A_j}{N_A} + D \right). \quad (8)$$

### Расчётные выражения для выбора средств передачи

Основное условие, которое можно соблюдать при выборе, чтобы фактическое время, требуемое для передачи данных  $t_{\Pi}$  было меньше некоторого заданного допустимого времени  $t_{доп}$ .

Необходимое количество каналов связи определяется по формуле

$$N_{KAN} = \sum_{i=1}^d \frac{n_i}{r_i} + N_{PE3}, \quad (9)$$

где  $d$  – количество датчиков;  $n_i$  – количество датчиков  $i$ -го вида;  $r_i$  – количество датчиков  $i$ -го вида присоединяемых к одному каналу;  $N_{PE3}$  – количество резервных каналов, на перспективу развития системы.

Суммарная разрядность определяется выражением

$$W_{\Pi} = \sum_{i=1}^{N_{КАН}} r_i \cdot W_i + W_{PEЗ}, \quad (10)$$

где  $r_i$  – количество датчиков, присоединённых к  $i$ -му каналу;  $W_i$  – разрядность датчиков, присоединённых к  $i$ -му каналу;  $W_{PEЗ}$  – число резервных разрядов.

Объём передаваемой информации  $M_{\Pi}$  есть произведение суммарной разрядности на частоту передачи

$$M_{\Pi} = W_{\Pi} \times f_{\Pi}. \quad (11)$$

Время передачи информации для различных типов аппаратуры зависит от объёма передаваемых данных и составляет

$$t_{\Pi} = \frac{M_{\Pi}}{V_{\Pi i} \cdot R_v} + t_c, \quad (12)$$

где  $V_{\Pi i}$  – скорость передачи по  $i$ -му типу аппаратуры;  $R_v$  – коэффициент снижения скорости передачи для повышения точности передаваемых данных ( $0,5 < R_v < 0,7$ );  $t_c$  – время, затрачиваемое на установление соединения.

Количество линий передачи может быть оценено как

$$N_{\Pi} \geq \frac{t_{\Pi}}{t_{доп}}. \quad (13)$$

### Расчётные выражения для выбора средств обработки

Необходимо сравнить степень соответствия параметров  $P_C$  – сравниваемой вычислительной системы и  $P_T$  – требуемых, полученных в результате расчётов. Соответствия могут быть заданы совокупностью баллов  $\{b_i\}$ , полученных в результате экспертного оценивания и относительного соизмерения сравниваемых величин  $i$ -х показателей

$$b_i = \frac{P_{C,i}}{P_{T,i}} \cdot z_i + \left(1 - \frac{P_{C,i}}{P_{T,i}}\right) \cdot z_{д,i}, \quad (14)$$

где  $z_i$  – коэффициент значимости  $i$ -го показателя;  $z_{д,i}$  – коэффициент допустимости несоответствия  $i$ -го параметра по отношению к требуемому значению.

Оценка удовлетворения всей системы средств обработки производится по одной из формул

$$\eta_{уд} = \begin{cases} \sum_{i=1}^n b_i, & a) \\ \frac{1}{n} \sum_{i=1}^n b_i, & б) \\ \prod_{i=1}^n b_i, & в). \end{cases} \quad (15)$$

Выражения 15, а и 15, б имеют тенденцию к сглаживанию резких отклонений расчётных параметров от желаемых. При использовании выражения 15, в, эти отклонения оказывают существенное воздействие. Окончательный выбор производится по максимальному значению  $\eta_{уд}$ . Затем по

отобранному варианту оценивается максимальная загрузка обрабатывающего комплекса

$$V = \frac{M_o}{T_p \cdot K_{доп}}, \quad (16)$$

где  $M_o$  – объём массива обрабатываемой информации;  $T_p$  – время решения задач;  $K_{доп}$  – коэффициент дополнительного времени.

### **ХОД РАБОТЫ**

1. Получить вариант задания. Провести расчеты по приведенным формулам.
2. Написать программу для расчёта.
4. Дополнить программу конструкциями для вывода графиков.
5. Оформить отчёт, защитить результаты выполнения лабораторной работы.

### **КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Что такое цикл работ по регистрации данных?
2. Какое ограничение необходимо удовлетворить при выборе средств передачи?
3. Как оценить количество каналов связи?
4. Как рассчитать количество линий передачи?
5. От чего зависит, насколько система удовлетворяет требуемым параметрам?

### 3.4 ЛАБОРАТОРНАЯ РАБОТА № 4

## ИССЛЕДОВАНИЕ АСПЕКТОВ КОЛИЧЕСТВЕННОГО АНАЛИЗА КОМПЛЕКСА ТЕХНИЧЕСКИХ СРЕДСТВ В СИСТЕМНОЙ ИНЖЕНЕРИИ

### 1. Цель работы

- Получить практические навыки формирования количественных оценок комплекса технических средств.
- Получить практические навыки в проведении количественного анализа вычислительных систем.

### 2. Теоретические сведения

В основе анализа комплекса технических средств лежит определение соответствия объёмов работ и временных ограничений по количеству и производительности технических средств.

Основным ограничивающим временным фактором является цикл работ (Ц) обработки информации. Под этим термином понимается промежуток времени от начала поступления данных на обработку  $T_H$  до срока готовности задачи  $T_G$ :

$$Ц = T_G - T_H. \quad (1)$$

В свою очередь, цикл работ будет складываться из допустимых продолжительностей выполнения подготовительных операций  $T_{II}$  и времени решения задач на ПЭВМ (вычислений)  $T_B$ . Продолжительность решения задач на ПЭВМ зависит от реализации основных этапов и режимов обработки информации.

Композиционно время решения задачи решения разбивается на этапы

$$t = t_I + t_P + t_E + t_O, \quad (2)$$

где  $t_I$  – ввод (загрузка, input) программы и данных;  $t_P$  – процессорное время (processing), затрачиваемое на счёт непосредственно;  $t_E$  – время, затрачиваемое на обмен в процессе счёта с внешними (external) устройствами;  $t_O$  – формирование и вывод (output) результатов работы.

Время ввода определяется по элементарной формуле

$$t_I = \frac{M_I}{v_I}, \quad (3)$$

где  $M_I$  – объём массива вводимых данных;  $v_I$  – быстродействие устройства.

Процессорное время может рассчитываться двояко, в зависимости от степени известности информации об организации вычислительного процесса. В первом случае

$$t_P = \bar{t}_{op} \cdot k_{CT} \cdot M_I, \quad (4)$$

где  $\bar{t}_{op} = \sum_j t_j \cdot q_j$  – среднее время выполнения операции,  $k_{CT}$  – коэффициент соотношения операций, показывающий, какое количество операций приходится на байт введённой информации;  $t_j$  – время выполнения  $j$ -й операции;  $q_j$  –

удельный вес  $j$ -й операции.

Для определения коэффициентов соотношения операций и удельных весов операций задачи разбиваются на типовые группы.

I. Задачи перегруппировки и упорядочивания информации, нахождения сумм, выборок, сечений по относительно простым критериям. Удельный вес логических операций достигает 70%, арифметических – до 30%. Коэффициент соотношения находится в пределах 50 – 100 операций на байт введенной информации.

II. Манипуляция несколькими массивами, часть которых находится на внешних запоминающих устройствах. Объем арифметических операций незначителен: нахождение сумм, парных произведений, операции с матрицами. Удельный вес логических операций достигает 90 – 95%, арифметических – до 5 – 10%,  $k_{CT} \in (1000 – 2000)$  операций на байт.

III. Задачи экстремального характера: численная реализация сложных математических методов, изобилует циклами, итерациями и т. п. Удельный вес арифметических операций достигает 90%, а удельный вес логических – всего около 10%,  $k_{CT} \in (1000 – 3000)$  операций на байт.

Во втором случае, если априори известен алгоритм решения задачи, то расчёты целесообразно вести по формуле

$$t_p = \sum_j t_j \cdot h_j, \quad (5)$$

где  $h_j$  – количество  $j$ -х операций применённых в реализации алгоритма.

Затраты времени на обмен с внешними устройствами рекомендуется определять по формуле

$$t_E = \frac{k_{rw}}{v_{rw}} \sum_j f_j \cdot V_j, \quad (7)$$

где  $k_{rw}$  – коэффициент, учитывающий контрольные операции, при проведении операций считывания или записи (например, для контрольного суммирования  $k_{rw} = 2$ );  $v_{rw}$  – скорость чтения или записи;  $f_j$  – количество обращений к внешним устройствам;  $V_j$  – объём считываемой или записываемой порции данных. Коэффициенты, находящиеся в (7) перед знаком суммы, могут быть, в случае разнообразия устройств ввода и вывода, использованы в качестве множителей под знаком.

Время вывода результата определяться быстродействием конечного устройства как отношение

$$t_o = \frac{M_o}{v_o} = \frac{M_I \cdot k_{Io}}{v_o}, \quad (8)$$

где  $M_o$  – объём массива выводимых данных;  $v_o$  – быстродействие устройств;  $M_I$  – объём массива вводимых данных;  $k_{Io}$  – коэффициент соотношения выводимой информации по отношению к объёму входной.

Общее время решения последовательности задач составит

$$t_\Sigma = \sum_I t_i, \quad (9)$$

где  $t_i$  определяется (2) для каждой задачи.

Отсюда потребность в вычислительных устройствах составляет

$$N \geq \frac{Ц}{t_{\Sigma}}. \quad (10)$$

### 3. ХОД РАБОТЫ

1. Получить вариант задания. Провести расчеты по приведенным формулам.
2. Написать программу для расчёта.
4. Дополнить программу конструкциями для вывода графиков.
5. Оформить отчёт, защитить результаты выполнения лабораторной работы

### 4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое цикл работ обработки информации?
2. Для чего задачи, решаемые на ЭВМ, разбиваются на типовые группы
3. Какова роль соотношений между видами операций при расчёте процессорного времени?
4. Что такое “время решения последовательности задач”?

### 3.5 ЛАБОРАТОРНАЯ РАБОТА № 5

## ИССЛЕДОВАНИЕ ЭКОНОМИЧЕСКИХ АСПЕКТОВ АНАЛИЗА КОМПЛЕКСА ТЕХНИЧЕСКИХ СРЕДСТВ В СИСТЕМНОЙ ИНЖЕНЕРИИ

### 1. Цель работы

- Получить практические навыки определения экономических показателей комплекса технических средств.
- Получить практические навыки в проведении стоимостного анализа вычислительных систем.

### 2. Теоретические сведения

Цель стоимостного анализа заключается в обеспечении наибольшей эффективности на единицу затрат машинной обработки информации при соблюдении заданных условий.

Одним из методов проведения экономического анализа комплекса технических средств является оценка себестоимости информации  $C$ . Практически, себестоимость складывается из ряда статей, из которых основными являются: прямые расходы по заработной плате; административно-управленческие и общехозяйственные расходы; общепроизводственные расходы.

Расходы по заработной плате есть вся заработная плата производственного с отчислениями в различные фонды, например, на социальное страхование.

Административно-управленческие расходы включают зарплату административно-управленческого персонала и обслуживающего персонала, расходы на командировки, канцелярские и прочие управленческие расходы.

Общепроизводственные расходы условно делятся на три группы: расходы на эксплуатацию оборудования (электроэнергия, текущий ремонт оборудования и т. п.); прочие расходы (стоимость расходных материалов, бумаги, тонера, расходы на подготовку и повышения квалификации кадров); распределяемые расходы (охрана труда, содержание зданий, износ малоценного инструментария и инвентаря, транспортные расходы).

Суммарный характер расходов не позволяет определить расходы по отдельной решаемой задаче. Поэтому объём расходов на практике распределяют пропорционально использованию машинного времени или объёма перерабатываемой информации.

В первом случае, это себестоимость машинного времени полезной работы вычислительной системы, то есть

$$S_{MT} = \frac{C}{T_p}, \quad (1)$$

где  $C$  – себестоимость, полученная суммированием ряда статей, упомянутых выше,  $T_p$  – полезное время работы.

Второй подход в качестве калькуляционной единицы использует единицу вводимой информации. Себестоимость обработки единицы вводимой информации будет равна

$$S_m = \frac{C}{M}, \quad (2)$$

где  $M$  – объём перерабатываемой информации.

По предельным объёмом перерабатываемой информации  $M_{\Pi}$  будем понимать такой объём, который может быть обработан за некоторый отрезок (период) времени  $\tau_{BM}$  функционирования системы. Указанный объём может быть рассчитан по формуле

$$M_{\Pi} = \frac{\tau_{BM} \cdot F_{ДН} \cdot K_{И}}{\bar{t} \cdot K_{ДОП}}, \quad (3)$$

где  $F_{ДН}$  – дневной (суточный) фонд работы системы, обычно задаётся в часах;  $K_{И}$  – коэффициент использования вычислительной системы;  $\bar{t}$  – среднее время обработки единицы информации;  $K_{ДОП}$  – коэффициент дополнительного времени.

Среднее время обработки единицы информации рассчитывается как сумма

$$\bar{t} = \sum_{N_I} t_I \cdot q_I + \sum_{N_P} t_P \cdot K_P \cdot q_P + \sum_{N_O} t_O \cdot K_O \cdot q_O. \quad (4)$$

где  $N_I$  – количество разновидностей способов ввода информации;  $t_I$  – время ввода единицы информации;  $q_I$  – удельный вес разновидности ввода;  $N_P$  – количество групп сложности задач;  $t_P$  – среднее время выполнения операции;  $q_P$  – удельный вес задач соответствующей сложности;  $K_P$  – коэффициент соотношения операций;  $N_O$  – количество разновидностей способов вывода информации;  $t_O$  – время вывода единицы информации;  $q_O$  – удельный вес вида выводимой информации;  $K_O$  – коэффициент соотношения вывода.

На рисунке 1 показан принцип анализа систем для комплексов 1 и 2. Комплекс 1 имеет меньшие эксплуатационные расходы, чем комплекс 2, но он, по сравнению с ним, является маломощным, поскольку имеет меньший объём перерабатываемой информации. Желание увеличить этот объём приведёт к увеличению мощностей вычислительного оборудования, что автоматически приведёт к росту эксплуатационных расходов, и к увеличению стоимости обработки.



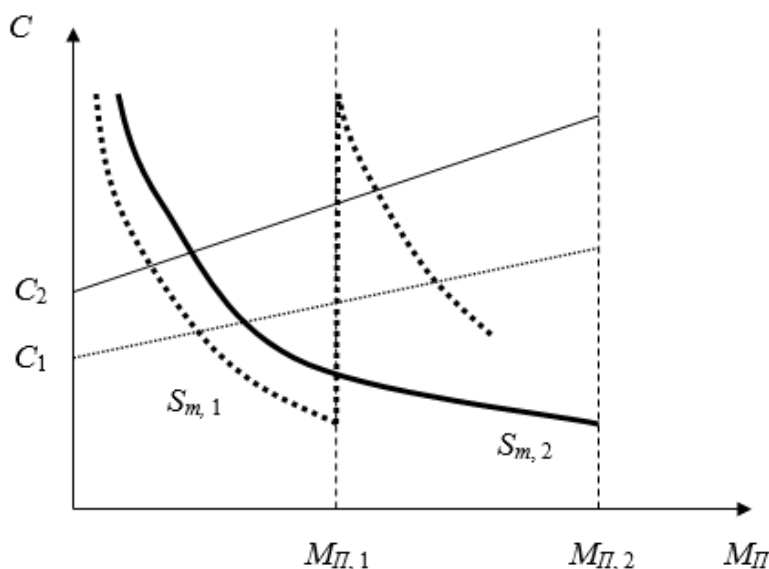


Рисунок 1 – Качественные графики анализа комплексов

Скачок на графике для линии  $S_{m,1}$  объясняется как раз этой причиной. Таким образом, слева от  $M_{П,1}$  будет эффективен 1-й комплекс, а от  $M_{П,1}$  до  $M_{П,2}$  второй. После границы  $M_{П,1}$  аналогичный скачок по тем же причинам будет претерпевать  $S_{m,2}$ .

### 3. ХОД РАБОТЫ

1. Получить вариант задания. Провести расчеты по приведенным формулам.
2. Написать программу для расчёта.
4. Дополнить программу конструкциями для вывода графиков.
5. Оформить отчёт, защитить результаты выполнения лабораторной работы

### 4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Каковы цели и задачи стоимостного анализа?
2. Что такое административно-управленческие расходы?
3. Что такое общепроизводственные расходы?
4. Почему для изначально маломощного комплекса при возрастании объёма обрабатываемой информации себестоимость изменяется скачкообразно?

### 3.6 ЛАБОРАТОРНАЯ РАБОТА №6\*

## ФОРМУЛИРОВКА И АНАЛИЗ ТРЕБОВАНИЙ К ИНФОРМАЦИОННОЙ СИСТЕМЕ

(выполняется по согласованию с преподавателем)

### 1. Цель работы

- ознакомиться с процессом формирования и анализа требований;
- осуществить сбор, формулировку и документирование требований к рассматриваемой системе.

**Время:** 4 часа

### 2. Краткие теоретические сведения

Системная инженерия – это практическое применение научных, инженерных и управленческих навыков, необходимых для преобразования операционных требований в описание конфигурации системы, которая наилучшим образом удовлетворяет этим требованиям. Это общий процесс решения проблем, который применяется ко всему техническому управлению в проекте, посвященном разработке системы, предоставляя механизм формулирования и совершенствования определений изделий и процессов системы.

Системная инженерия включает в себя пять функций.

- определение проблемы – указание потребностей и ограничений путем анализа требований и взаимодействия с заказчиком;
- анализ решений – выделение набора возможных способов удовлетворения потребностей и ограничений, их анализ и выбор оптимального;
- планирование процессов – определение задач, которые должны быть выполнены, объема ресурсов и затрат, необходимых для создания изделия, очередности задач и потенциальных рисков;
- контроль процессов – определение методов мониторинга проекта и процессов, измерение прогресса, оценка промежуточных изделий и принятие по мере необходимости корректирующих действий;
- оценка изделий – определение качества и количества создаваемых изделий путем оценочного планирования, тестирования, демонстрации, анализа, верификации и контроля. Системная инженерия определяет систему как совокупность компонентов (физические системы, аппаратное обеспечение, программное обеспечение и персонал), которые совместно реализуют предполагаемые функции конечного продукта. Разработка продуктов – это логическая последовательность оценок затрат на циклы жизни продукта, управление исследованием технических компромиссных решений, анализ эффективности затрат, моделирование эффективности, проектирование архитектуры и процесса изготовления. Все вместе эти шаги преобразуют цели в параметры эффективности и предпочтительную конфигурацию системы.

На рис.5.1 показан типичный итеративный процесс системной инженерии. В его рамках выполняются попытки оценить варианты системной архитектуры и выбрать из них наилучшие.

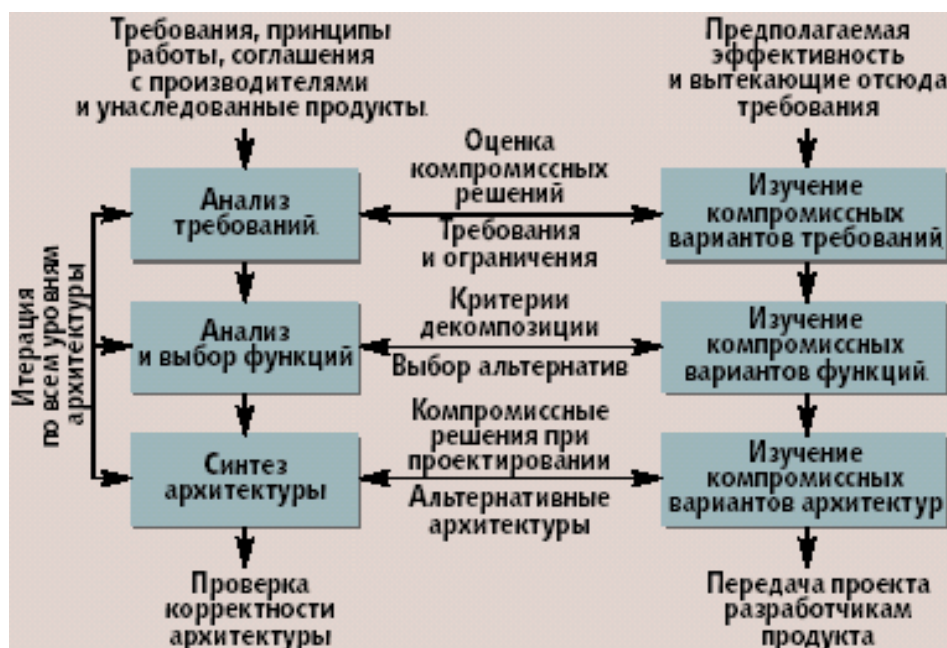


Рисунок 5.1 Типичный процесс системной инженерии при определении архитектуры

Первый шаг в любой активности – определение требований, их систематизация, документирование в виде спецификации системных требований. (перечисленная совокупность действий, дополненная анализом, выявлением противоречий, неполноты, разрешением конфликтов в процессе разработки – в англоязычной литературе называют инженерией требований).

Процесс формирования и анализа требований проходит через ряд этапов:

- разработка требований:
  - выявление требований;
  - анализ требований;
  - спецификация требований;
  - проверка требований;
- управление требованиями.

Выявление и анализ требований заключается в выявлении заинтересованных лиц (в англоязычной литературе – стейкхолдеров – физические лица или организации, имеющие права, долю, требования или интересы относительно системы или её свойств, удовлетворяющих их потребностям и ожиданиям (ISO/IEC 15288:2008, ISO/IEC 29148:2011)). Опрос стейкхолдеров является широко используемой техникой при сборе требований.

Собранные требования могут быть отнесены к следующим типам:

– **требования клиентов**, то есть тех, кто выполняет основные функции системного проектирования, со специальным акцентом на пользователе системы как ключевом клиенте. Пользовательские требования определяют

главную цель системы и, как минимум, ответят на следующие вопросы: где система будет использоваться? (требования эксплуатации или развёртывания); как система достигнет целей миссии? (профиль миссии или сценарий); какие параметры системы являются критическими для достижения миссии? (требования производительности); как различные компоненты системы должны использоваться? (сценарии использования); насколько эффективной должна быть система для выполнения миссии? (требования эффективности); как долго система будет использоваться? (эксплуатационный жизненный цикл); каким окружением система должна будет эффективно управлять? (окружающая среда);

- **функциональные требования** объясняют, что должно быть сделано. Они идентифицируют задачи или действия, которые должны быть выполнены, определяют связь входа/выхода в поведении системы;

- **нефункциональные требования** определяют критерии работы системы в целом, а не отдельные сценарии поведения (системные свойства такие как производительность, удобство сопровождения, расширяемость, надёжность, средовые факторы эксплуатации);

- **производные требования** – подразумеваются или преобразованы из высокоуровневого требования.

*Методы документирования требований* (составления спецификация требований):

- **списки требований** – традиционный способ документировать требования (используются и по сей день, хотя крайне неэффективны в современном анализе, в сложной системе такие списки требований могут занимать сотни страниц);

- **пользовательские истории** – определяют требования обычным языком;

- **измеримые цели** – рассматривается составленный список требований как подсказки и постоянно задается вопрос: «почему?», до тех пор, пока не будут выявлены истинные цели; разрабатываются тесты, измеряющие, какой уровень каждой цели был достигнут. Как только набор критических, измеримых целей установлен, быстрое прототипирование и короткие этапы разработки могут дать реальную ценность ещё до окончания проекта (являются практикой целеориентированной системной инженерии);

- **прототипирование** – построение макета (модели) системы. Макеты дают возможность представить систему, которая ещё не построена;

- **сценарии использования** – варианты использования (англ. Use Case) – техника для документации потенциальных требований для создания новой системы или изменения существующей. Каждый вариант описывает один или несколько способов взаимодействия системы с конечным пользователем или другой системой, для достижения определённой цели. Варианты использования обычно избегают технического жаргона, предпочитая вместо этого язык конечного пользователя или эксперта в данной области (являются практикой объектно-ориентированной системной инженерии и моделиориентированной системной инженерии);

Чтобы определить и выразить различные аспекты процесса разработки, в том числе и этапа формирования и анализа требований, применяются такие методики, как:

- сети Петри (помогают определить параллельность и синхронизацию);
- машины с конечным числом состояний (состояния и режимы);
- структурный анализ (потoki данных);
- PSL/PSA (Problem Statement Language/Problem Statement Analysis);
- язык Unified Modeling Language (OMG Unified Modeling Language Specification, Object Management Group, Sept. 2001).

### **Задание на лабораторную работу**

1. Определить требования пользователя к рассматриваемой системе.
2. Определить требования к рассматриваемой системе в рамках целеориентированной системной инженерии.
3. Определить обязанности рассматриваемой системы, то есть указать, что должна делать система (функциональные требования) (модель «черный ящик» – зафиксировать и перечислить входные и выходные параметры рассматриваемой системы, связи со средой функционирования), определить связь входа/выхода в поведении системы.
4. Определить нефункциональные требования (критерии работы системы).
5. Построить сценарии использования рассматриваемой системы.

### **Содержание отчета**

1. Требования пользователя к рассматриваемой системе (список требований или пользовательские истории).
2. Список измеримых целей системы.
3. Функциональные требования к рассматриваемой системе (входные/выходные параметры, их связь, внешние воздействия).
4. Нефункциональные требования (критерии, которым должна соответствовать функционирующая система).
5. Сценарии поведения системы.

### **Контрольные вопросы**

1. Какова основная задача системной инженерии?
2. Перечислить этапы (фазы) процесс формирования и анализа требований?
3. Какова цель процесса определения требований в соответствии с ГОСТ Р ИСО/МЭК 15288-2005?
4. Перечислить и раскрыть суть методов документирования требований.

## **4.ИНСТРУКТИВНО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ РГР (КУРСОВОМУ ПРОЕКТИРОВАНИЮ)**

### **4.1 Цели и задачи РГР (курсового проектирования)**

**Целью РГР (курсового проекта)**, выполняемого на первом курсе магистратуры (II семестр), является активизация исследовательской деятельности магистрантов в рамках подготовки выпускной квалификационной работы (ВКР) с использованием требуемого набора средств проектирования и моделирования сложных ИС, а также систем различной физической природы.

Курсовой проект по дисциплине **«Системный анализ и проектирование информационных систем»** представляет собой первый этап унифицированного процесса оформления научно-исследовательской работы, которую студенты выполняли в течение I семестра 1 курса магистратуры по форме ВКР.

### **4.2. Содержание РГР (курсового проекта) и пояснительной записки**

Общее содержание РГР (курсового проекта):

- 1) Выявление множества (комплекса) технологий проектирования, эффективно реализующих задачу по каждому этапу ЖЦ.
- 2) Формальное описание комплекса технологий проектирования, эффективно реализующих задачу по каждому этапу ЖЦ.
- 3) Оценка эффективности ИС и(или) ИТ в задаче.
- 4) Оценка возможных эксплуатационных характеристик ИС и(или) ИТ в задаче.

**Содержание пояснительной записки (ПЗ) к РГР (курсовому проекту)** следующее:

#### **1.Постановка задачи**

В постановку задачи включаются:

- 1.1 Описание предметной области.
- 1.2 Описание входных и выходных данных ИС или ИТ.
- 1.3 Описание требований к ИС или ИТ.
- 1.4 Описание ограничений при создании ИС или ИТ.

#### **2.Описание комплекса технологий для проектирования ИС и (или ИТ)**

- 2.1 Обоснование выбора модели жизненного цикла при проектировании ИС или ИТ.
- 2.2 Описание вложенных уровней проектирования ИС или ИТ на основе диаграммы Венна<sup>1</sup>.
- 2.3 Обоснование комплекса технологий при проектировании ИС или ИТ.
- 2.4. Формализованное описание комплекса технологий при проектировании ИС или ИТ.

---

<sup>1</sup> Игнатъев А.В., «Методы и средства проектирования ИС и ИТ», эл. Учебник. 2014г.

### **3. Оценка эффективности выбранного комплекса технологий при проектировании ИС или ИТ.**

3.1. Выбор критериев оценки комплекса технологий при проектировании ИС или ИТ.

3.1. Расчет показателя эффективности оценки комплекса технологий при проектировании ИС или ИТ.

### **4. Оценка возможных эксплуатационных характеристик ИС и(или) ИТ в задаче НИР.**

4.1. Расчет информационных характеристик БД (требуемый объем жесткого диска при учете интенсивности приращения информационного фонда).

**Заключение.**

**Библиографический список.**

**Приложения.**

#### **4.3 Порядок защиты РГР (курсового проекта)**

Согласно «Положения о планировании и организации учебного процесса», раздел 7.4.8: задание выдается в письменном виде не позднее, чем через 14 дней после начала семестра, в котором выполняется проект.

Прием защиты РГР (курсового проекта, работы) производится комиссией и (или) руководителем проекта (работы) и оценивается отметками «отлично», «хорошо», «удовлетворительно» и «неудовлетворительно».

Защита РГР (курсовой работы, проекта) состоит в кратком докладе студента по выполненной теме (с использованием слайдов, графиков и других наглядных пособий) и его ответах на вопросы, задаваемые присутствующими на защите.

Защита курсовой работы оценивается по следующим критериям:

- степень усвоения студентом понятий и категорий по теме курсового исследования;
- умение работать с документальными и литературными источниками;
- умение формулировать основные выводы по результатам анализа конкретного материала;
- грамотность и стиль изложения материала;
- самостоятельность работы, оригинальность мышления в осмыслении материала;
- наличие презентации;
- умение доложить полученные результаты.

Выполненные проекты (работы) после их защиты хранятся на кафедре один год, а затем списываются по акту, утвержденному заведующим кафедрой, и уничтожаются. Часть проектов, представляющих научно-практический интерес, может быть оставлена на кафедре по решению заведующего кафедрой.

Итоги выполнения и защиты курсовых проектов (работ) обсуждаются на заседаниях кафедр. Студент, не защитивший до начала экзаменационной сессии курсовой проект (работу), к экзаменам не допускается.

#### 4.4 Выбор приемлемой модели жизненного цикла по ИС

Выбор приемлемой модели жизненного цикла разработки ПО для проекта может осуществляться в ходе использования следующего процесса [65].

1. Проанализируйте следующие отличительные категории проекта, помещенные в таблицах 1-4:
  - Требования: таблица 1.
  - Команда разработчиков: таблица 2.
  - Коллектив пользователей: таблица 3.
  - Тип проекта и риски: таблица 4.
2. Ответьте на вопросы, приведенные для каждой категории, обведя кружочком слова "да" или "нет".
3. Расположите по степени важности категории или вопросы, относящиеся к каждой категории, относительно проекта, для которого выбирается приемлемая модель.
4. Воспользуйтесь упорядоченными категориями для разрешения противоречий, возникающих при сравнении моделей, если общие полученные показатели сходны или одинаковы.

В табл. 1-2 приведен набор матриц-тестов, предназначенных для выбора модели жизненного цикла.

**Требования.** Категория требований (таблица 1) состоит из вопросов относительно требований, которые предъявляет пользователь к проекту. В терминологии их иногда называют свойствами системы, которая будет поддерживаться данным проектом.

**Вывод:** если требования не могут быть заранее определены, а в ходе работ будут часто изменяться, то наиболее подходящими являются модель прототипирования и спиральная модель.

**Команда разработчиков.** По возможности, в состав команды разработчиков лучше всего отобрать персонал еще до того, как будет выбрана модель жизненного цикла. Характеристики такой команды (таблица 2) играют важную роль в процессе выбора, поскольку она несет ответственность за удачное выполнение цикла и может оказать помощь в процессе выбора.

**Таблица 1. Выбор модели жизненного цикла на основе характеристик требований**

Требования	Каскадная	V-образная	Прототипирование	Спиральная	RAD	Инкрементная
Являются ли требования легко определяемыми и/или хорошо известными?	Да	Да	Нет	Нет	Да	Нет
Могут ли требования заранее определяться в цикле?	Да	Да	Нет	Нет	Да	Да



Требования	Каскадная	V-образная	Прототипирование	Спиральная	RAD	Инкрементная
Часто ли будут изменяться требования в цикле?	Нет	Нет	Да	Да	Нет	Нет
Нужно ли демонстрировать требования с целью определения?	Нет	Нет	Да	Да	Да	Нет
Требуются ли для демонстрации возможностей проверка концепции?	Нет	Нет	Да	Да	Да	Нет
Будут ли требования отражать сложность системы?	Нет	Нет	Да	Да	Нет	Да
Обладает ли требование функциональными свойствами на раннем этапе?	Нет	Нет	Да	Да	Да	Да

**Таблица 2. Выбор модели жизненного цикла на основе характеристик участников команды разработчиков**

Команда разработчиков проекта	Каскадная	V-образная	Прототипирование	Спиральная	RAD	Инкрементная
Являются ли проблемы предметной области проекта новыми для большинства разработчиков?	Нет	Нет	Да	Да	Нет	Нет
Является ли технология предметной области проекта новой для большинства разработчиков?	Да	Да	Нет	Да	Нет	Да
Являются ли инструменты, используемые проектом, новыми для большинства разработчиков?	Да	Да	Нет	Да	Нет	Нет
Изменяются ли роли участников проекта во время ЖЦ?	Нет	Нет	Да	Да	Нет	Да
Могут ли разработчики проекта пройти обучение?	Нет	Да	Нет	Нет	Да	Да
Является ли структура более значимой для разработчиков, чем гибкость?	Да	Да	Нет	Нет	Нет	Да

Таким образом, существует множество различных моделей или представлений жизненного цикла разработки ПО. Все они представляют собой логически построенную последовательность действий, начиная с определения потребности и заканчивая производством ПО. Каждая модель представляет собой процесс, который структурно состоит из этапов, направленных на

обеспечение целостности соответствующих субкомпонентных действий. Каждая фаза снижает степень риска при выполнении проекта, что достигается благодаря применению критериев входа и выхода для определения дальнейшего хода действий. По завершении каждой фазы получают внутренние или результативные внешние действия.

ЖЦ разработки ПО иногда называют методиками менеджмента жизненных циклов. Эти методики охватывают все стандарты и процедуры, оказывающие влияние на планирование, сбор требований и анализ, разработку проекта, конструирование и внедрение программной системы. С целью обеспечения эффективности произвольного жизненного цикла его потребуется аккуратно выбрать и зачастую настроить (подогнать и разработать) в соответствии с задачами и целями определенного проекта.

Из приведенных рассуждений следует, процедура выбора модели ЖЦ должна осуществляться на основе рассмотрения не отдельных критериев, а их комплекса. При этом существенную роль будут играть особенности реальной ситуации [65].

#### **4.5 Формализованное описание комплекса технологий при проектировании ИС или ИТ**

В общем виде технология проектирования ИС (ТП ИС) это совокупность следующих составляющих:

$$TD: \langle D(M^D, S^D), OD(M^{OD}, S^{OD}), OR(M^{OR}, S^{OR}) \rangle, \quad (1)$$

где  $D(M^D, S^D)$  – концептуальные методы и средства (методологии) проектирования (design) ИС,  $OD(M^{OD}, S^{OD})$  – средства организации проектирования, то есть управления процессом создания,  $OR(M^{OR}, S^{OR})$  – средства организации модернизации проекта информационной системы.

Соответственно,  $M, S$  – обобщенное представление методов и средств ТП ИС.

На рис.4.2 изображен пример описания комплекса технологий для реализации проекта. В строках обозначены процессы, методологии и технологии создания системы организационного управления процессом реализации геоинформационной системы (ГИС).



Рисунок 4.1 – ПРИМЕР схемы комплекса технологий объекта проектирования

Для детализации и представления комплекса технологий сформирована таблица 3, по которой построено формализованное описание комплексной технологии в общем виде.

Таблица 4 – Пример сводной таблицы технологий, методологий и программных средств для проектирования

Обозначение	О	А	Д	Р	Т	М
Процессы ЖЦ	Процессы описания	Анализ требований	Процесс проектирования	Построение	Верификация, тестирование	Моделирование
Методологии	Текстовые	IDEF0	-IDEF0 IDEF3 -RUP	Формализация описания моделей МОП	МОП	
ПО	MSWord  Visio DIA	MSExcel	Visio  Ramus Educational	Maple  Matlab	Any Logic	

В соответствии с таблицей 3 была составлена формализованное описание технологии проектирования в общем виде:

$$TD = \langle O \langle M_1^O, S_1^O; M_1^O, S_2^O \rangle; A \langle M_1^A, S_1^A \rangle; D \langle M_1^D, S_1^D; M_1^D, S_2^D; M_2^D, S_1^D; M_2^D, S_2^D \rangle; R \langle M_1^R, S_1^R; M_1^R, S_2^R \rangle; T \langle M_1^T, S_1^T \rangle; M \langle M_1^M, S_1^M \rangle \rangle,$$

где  $M_1^O$  соответствует текстовым методологиям, обеспечивающим описание требований к технологии проектирования на первом этапе анализа ЖЦ;

$S_1^O$  – средство текстового редактирования MS Word, средство визуального редактирования MS Visio;

$S_2^O$  – средство визуального редактирования DIA;

$M_1^A$  – методология функционального моделирования IDEF0;

$S_1^A$  – средство табличного анализа MS Excel;

$M_1^D$  – методология функционального моделирования IDEF0, методология моделирования и стандарт документирования процессов, происходящих в системе IDEF3; и.т.д.

#### 4.6 Теоретические сведения и примеры выполнения задания на курсовое проектирование. Информационные расчеты ИС

Длина логической записи  $j$ -ого файла определяется как сумма длин полей, её составляющих и указателей, если они организуются разработчиком. Так, для наиболее общего случая длину логической записи можно определить по формуле:

$$L_j = \sum_{i=1}^{M_j} l_{ij} \text{ [байт]}, \quad (4.1)$$

где  $M_j$  – число групп полей в записях,  $l_{ij}$  длина группы [байт].

Если отдельным атрибутам объекта соответствуют переменная длина полей, то определяется максимальная  $\max L_j$  и средняя длины записи  $m\{L_j\}$ .

Для реляционной модели характерно фиксированное число как для длины каждого поля записи, так и для количества полей каждой записи, поэтому (4.1) можно представить в следующем виде:

$$L_j = \sum_{j=1}^M l_j \text{ [байт]}, \quad (4.2)$$

где  $M$  – число полей в записи  $j$ -ого файла,  $l_j$  – длина  $j$ -ого поля.

Объем памяти, необходимый для размещения информационного фонда без учёта системных данных и указателей составит

$$I = \sum_{j=1}^N L_j \cdot K_j \text{ [байт]}, \quad (4.3)$$

где  $N$  – количество отношений РБД,  $K_j$  – количество записей  $j$ -го файла.

Приращение информационного фонда

$$\Delta I = \sum_{j=1}^{N_\lambda} L_j \cdot \lambda_j \text{ [байт-1]}, \quad (4.4)$$

где  $N_\lambda$  – число добавленных типов записей,  $\lambda_j$  – интенсивность добавления записей в файл  $j$ -го типа.

Зная первоначальный объём  $V_{общ}$  памяти, выделенной под развёртывание БД, и объём программного обеспечения  $V_{ПО}$ , представляется возможным оценить время заполнения информационного фонда

$$T_{зап} = \frac{V_{общ} - V_{по} - I}{\Delta I} \text{ [время]}, \quad (4.5)$$

Полученное значение при расчете по выражению (4.5) следует оценить с точки зрения соответствия устареванию системы, т.е. если полученное время

значительно превышает предполагаемое время устаревания, необходимо выбрать меньшее значение  $V_{общ}$ . Практически это реализуется как выделение дисковых квот.

Время резервного копирования определяется несколькими способами: 1) интенсивностью отказов, сопровождающихся потерей данных

$$T_{pk} = \frac{0.5 \div 0.8}{\mu} [\text{время}], \quad (4.6)$$

где  $\mu$  - интенсивность отказов, сопровождающихся потерей данных.

2) Если данные такого рода отсутствуют, то копирование производится через промежутки времени, в которые поступает порция данных порядка 20% первоначального объема БД:  $T_{pk} = \frac{0.2 \cdot I}{\Delta I}$ . (4.7)

Количество обращений к логическим записям определяется как

$$n_j = \sum_{j=1}^N n_{ij}, \quad (4.8)$$

где  $n_{ij}$  – количество обращений к записям  $j$ -го типа в  $i$ -м запросе.

Интенсивность обращений к информационному фонду:

$$\eta = \sum_{i=1}^Z n_{ij} \cdot \varphi_i, \quad (4.9)$$

где  $\varphi_i$  – частота выполнения  $i$ -того запроса,  $Z$  - число запросов, обработка которых предусмотрена СУБД.

Средний объем данных, предоставляемый пользователю во время выполнения  $i$ -ого запроса определяется выражением

$$V_{данных} = \sum_{j=1}^N n_{ij} \cdot L_j. \quad (4.10)$$

На рисунках ниже приведены исходные данные и результаты моделирования эксплуатационных параметров БД, выполненных с помощью средств MS Excel.

редоты МБ Экспон.

время заполнения информационного фонда			T_зап=	-76077548	
Мера количества информации ▾			V_общ=	1048576	байт
МБ ▾ 1 = 1048576			V_ПО=	1073741821	байт
Мера количества информации ▾			T_зап=	-33,971631	
ГБ ▾ 1 = 1073741824			V_общ=	200	
			V_ПО=	500	

T_зап=	24,4681	дн.
V_общ=	1024	Мб
V_ПО=	500	Мб

$$T_{зап} = \frac{(V_{общ} - V_{ПО} - I)}{\Delta I} [\text{время}], \quad (5)$$

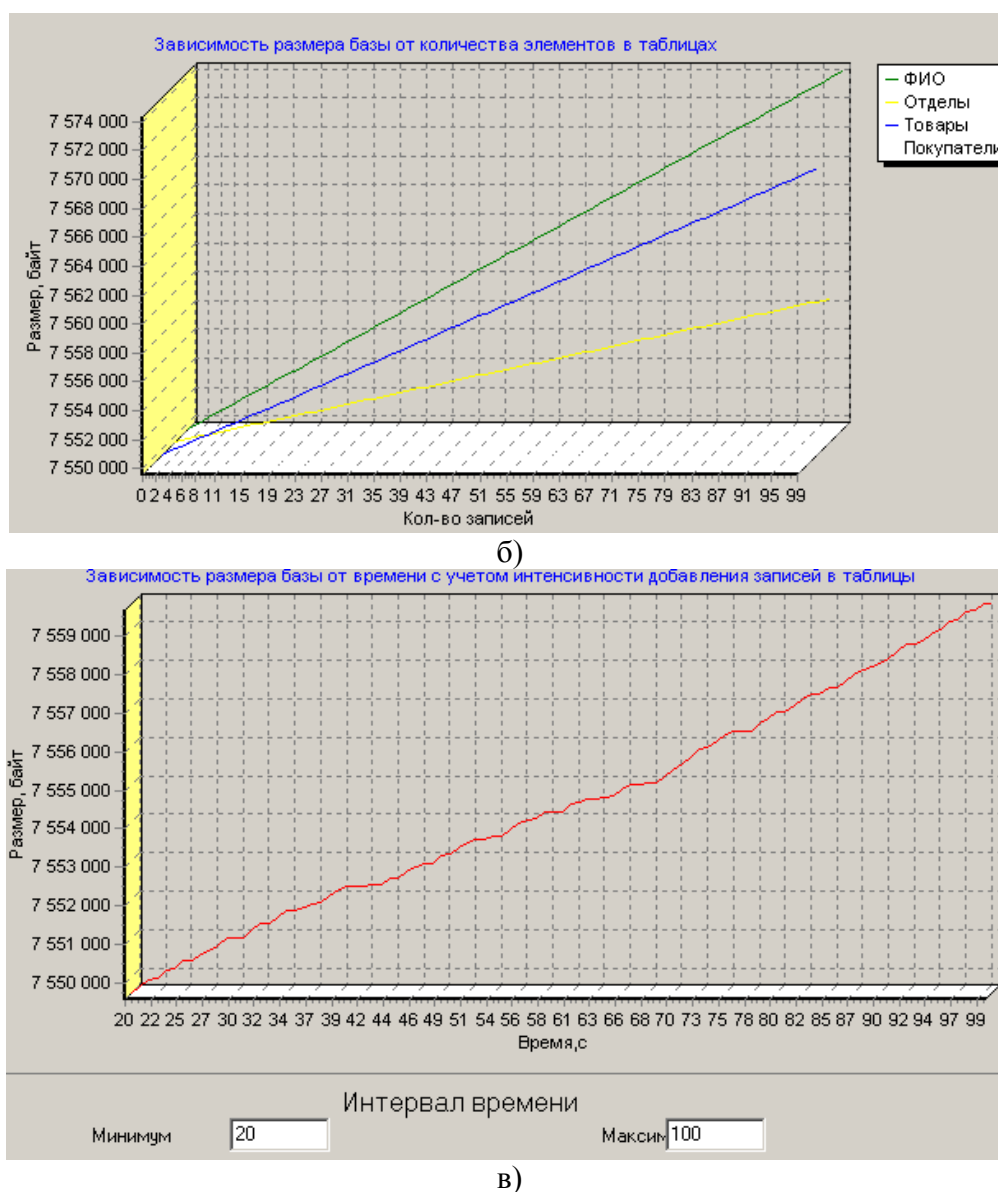
время резервного копирования, 1 способ			T_рк=	0,69444444	мес
			μ_max=	0,9	-
			T_рк=	1,5625	мес
интенсивность отказов			μ=	0,4	-
			T_рк=	6,25	мес
			μ_min=	0,1	-
время резервного копирования, 2 способ			T_рк=	2,53900709	мес

2) Если данные такого рода отсутствуют, то копирование производится через промежутки времени, в которые поступает порция данных порядка 20% первоначального объема БД:

$$T_{pk} = \frac{0.2 \cdot I}{\Delta I} \quad (7)$$

интенсивностью отказов, сопровождающихся потерей данных

$$T_{pk} = \frac{(0.5 \div 0.8)}{\mu} [\text{время}], \quad (6)$$



в)

Рисунок 4.1 – Примеры реализации графиков приращения информационного фонда

Анализ функционирования ИС по выражениям (4.5) – (4.7) оформить в форме графиков изменения объема жесткого диска от времени или объема БД от времени. При построении графиков необходимо учесть, что объем жесткого диска ПК или объем БД может измениться, исходя из частоты и вида резервного копирования.

#### 4.7 Общие требования к оформлению

Текст работы должен быть напечатан на одной стороне стандартного листа формата А4 (270 x 297 мм) через полтора интервала.

Поля должны оставаться по всем четырем сторонам печатного листа: левое –25 (30) мм, правое –10 мм, нижнее и верхнее –20 мм, количество знаков на странице – примерно 2000.

При печати нужно соблюдать следующие условия:

- текстовый редактор (рекомендуемый) – Microsoft Word;
- шрифт: гарнитура «Times New Roman», кегль – 12 пт;

- межстрочный интервал по основному тексту – полуторный;
- отступ абзаца – 1,25 см;
- расстановка переносов – автоматическая;
- выравнивание текста – по ширине страницы.

Допустимо применение в таблицах и рисунках кегля ниже 14-го (10-12 пт) и одинарного межстрочного интервала. Недопустимо применение в основном тексте «курсива» или «полужирного» шрифта, кроме выделения отдельных слов и словосочетаний.

Допускается использовать одинарный межстрочный интервал в «Оглавлении» и «Списке использованных источников».

Рекомендуется использование режима автоматического составления (добавления) «Оглавления» в тексте ПЗ.

Разделы (главы), подразделы (параграфы), пункты и подпункты основной части работы нумеруются арабскими цифрами (например, раздел:2, подраздел: 2.1, пункт: 2.1.1). Так, второй подраздел первого раздела получает номер 1.2. Между номером раздела, подраздела, пункта, подпункта и его названием точка не ставится. Слова «раздел» («глава»), «подраздел» («параграф»), «пункт», «подпункт» в заголовках и содержании ВКР не пишутся. Заголовки глав и параграфов выделяются **полужирным шрифтом**.

Перенос слов и их подчеркивание в заголовках не допускается.

Расстояние между *структурным заголовком* (разделом) и последующим текстом составляет два полуторных интервала (две пустые строки), между заголовком раздела и подраздела – тоже два полуторных интервала (две пустые строки). Каждый раздел (структурный заголовок) работы нужно начинать с новой страницы. Выравнивание заголовка раздела по центру, БЕЗ абзацного отступа.

Заголовки подразделов (1.1...) печатают строчными буквами (кроме первой) с абзацного отступа. Точку в конце заголовка не ставят. Если заголовок состоит из двух или более предложений, их разделяют точкой. Расстояние перед заголовком подраздела составляет два полуторных интервала (две пустые строки), после – один полуторный интервал (одна пустая строка). Выравнивание по ширине.

Заголовки пунктов (1.1.1...) печатают строчными буквами (кроме первой) с абзацного отступа. Расстояние перед заголовком пункта составляет один полуторный интервал (одна пустая строка), текст начинается с новой строки непосредственно после заголовка.

Если при перечислении перед каждым пунктом перечисления ставят среднее тире (дефис); либо строчную букву (за исключением ё, з, о, г, ь, и, ы, ь), после которой ставится скобка; либо цифру, после которой ставится скобка, тогда текст каждого пункта перечисления следует начинать со строчной буквы. После каждого пункта перечисления, кроме последнего, ставится точка с запятой.

Размещение таблицы рекомендуется выполнять по одному из вариантов: непосредственно под текстом, где она упоминается впервые, наследующей

странице (не далее) или в приложении. В приложении выносятся таблицы, которые содержат более 8-10 строк или свыше 7-8 граф. В текст работы включаются таблицы меньшего объема. Допускается размещать таблицу вдоль длинной стороны листа документа (альбомная ориентация страницы), так, чтобы для ее чтения надо было повернуть лист по часовой стрелке.

Номер таблицы записывают через точку: номер раздела, точка, номер таблицы в разделе (без пробелов). Название таблицы размещают над таблицей. Без абзацного отступа записывают: слово «Таблица», ее номер, далее – через тире – наименование таблицы с большой буквы. Точку в конце названия таблицы не ставят. Расстояние от заголовка таблицы до таблицы должно быть такое, чтобы текст заголовка таблицы плотно не прилегал к таблице (например, 3 пт). Расстояние от текста до таблицы и от таблицы до последующего текста равно одной строке.

Таблицы, вынесенные в приложения, имеют самостоятельную, отдельную нумерацию в той последовательности, в какой из них дается ссылка в тексте работы. Перед номером таблицы в этом случае ставится обозначение приложения. Например, «Таблица Б.3.» На все таблицы работы должны быть ссылки в тексте.

Таблицу с большим числом строк допускается переносить на другой лист (страницу). При переносе части таблицы на другой лист (страницу) слово «Таблица», ее номер и наименование указывают один раз слева над первой частью таблицы, а над другими частями также слева пишут слова «Продолжение таблицы» и указывают номер таблицы. Наименование таблицы повторять не нужно. Если «шапка» таблицы велика, допускается ее не повторять: в этом случае следует пронумеровать колонки и повторить их нумерацию на следующей странице.

На весь приведенный иллюстративный материал должны быть ссылки в тексте работы, например, «... в соответствии с рисунком 2» (при сквозной нумерации) или «... в соответствии с рисунком 1.2» (при нумерации в пределах главы). Иллюстрации каждого приложения обозначают отдельной нумерацией арабскими цифрами с добавлением перед цифрой обозначения приложения, например, рисунок А.3.

Для написания формул использовать только редактор формул Microsoft Equation не ниже версии 3.0. Найти его можно по следующему пути: вкладка Вставка → Объект → Тип объекта: Microsoft Equation 3.0.



## Список литературы и информационных ресурсов

1. Федеральный закон от 27.07.2006 N 149-ФЗ (ред. от 31.12.2014) «Об информации, информационных технологиях и о защите информации». Режим доступа – [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_173622/](http://www.consultant.ru/document/cons_doc_LAW_173622/).
2. ГОСТ 34.321-96 Информационные технологии. Система стандартов по базам данных. Эталонная модель управления данными. Режим доступа – <http://gostexpert.ru/gost/gost-34.321-96>.
3. ISO/IEC 2382-1:1993 Information technology; Vocabulary; Part 1: Fundamental terms (Информационные технологии. Словарь. Часть 1. Основные термины) Режим доступа – <http://rossert.narod.ru/alldoc/info/2z23/g28647.html>.
4. ГОСТ 34.003-90 Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения. Режим доступа – <http://base.garant.ru/187632/>.
5. ISO/IEC 38500:2015 Information technology – Governance of IT for the organization (Информационная технология. Корпоративное управление информационными технологиями) Режим доступа – [http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=62816](http://www.iso.org/iso/catalogue_detail.htm?csnumber=62816).
6. Новиков А.М., Новиков Д.А. Методология. – М.: СИНТЕГ. – 663 с. Режим доступа – [http://www.methodolog.ru/books/methodology\\_full.pdf](http://www.methodolog.ru/books/methodology_full.pdf).
7. Теория систем и системный анализ: Учебник/В.М. Вдовин, Л.Е. Суркова, В.А. Валентинов. – М.: Издательско-торговая корпорация «Дашков и К», 2010. – 640 с. Режим доступа – <http://ir.nmu.org.ua/bitstream/handle/123456789/143657/a261cee4cef2ef83d70df07a16c9ded6.pdf?sequence=1>.
8. ГОСТ Р ИСО/МЭК 12207-99 Информационная технология. Процессы жизненного цикла программных средств. Режим доступа – <http://www.complexdoc.ru/lib/ГОСТ%20Р%20ИСО%7СМЭК%2012207-99>.
9. РД 50-680-88 Методические указания. Автоматизированные системы. Основные положения. Режим доступа – <http://www.gostinfo.ru/catalog/Details/?id=2033907#.VZIGFC-5GDM>.
10. Избачков Ю.С., Петров В.Н., Васильев А.А., Тепина И.С. Информационные системы: Учебник для вузов. 3-е изд. – СПб.: Питер, 2011. – 544 с: ил.
11. ГОСТ Р ИСО/МЭК 15288-2005 Информационная технология. Системная инженерия. Процессы жизненного цикла систем. Режим доступа – <http://vsegost.com/Catalog/20/2011.shtml>.
12. ГОСТ Р ИСО/МЭК 15504-2009 Информационные технологии. Оценка процессов. Режим доступа – <http://protect.gost.ru/document.aspx?control=7&id=175314>.
13. ГОСТ 34.601-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания. Режим доступа – <http://vsegost.com/Catalog/10/10698.shtml>.
14. Проектирование информационных систем: учебное пособие. Авторы: Абрамов Г.В., Медведкова И.Е., Коробова Л.А. Издательство: ВГУИТ, 2012 г.
15. Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. – М.: Финансы и статистика, 1998. – 176 с.
16. Зыков С.В. Основы проектирования корпоративных систем / С.В. Зыков; Нац.исслед.ун-т «Высшая школа экономики»: Изд. дом Высшей школы экономики; Москва; 2012.
17. Дэвид А. Марк, Клемент Мак-Гоуэн Методология структурного анализа и проектирования SADT. М.:Метатехнология, 1993.
18. Цуканова О. А. Методология и инструментарий моделирования бизнес-процессов: учебное пособие – СПб.: Университет ИТМО, 2015. – 100 с. [Электронный ресурс] Режим доступа – <http://books.ifmo.ru/file/pdf/1720.pdf>.
19. Галямина И. Г. Управление процессами: Учебник для вузов. Стандарт третьего поколения. – СПб.: Питер, 2013. – 304 с: ил.
20. Построение диаграмм потоков работ – WFD. [Электронный ресурс] Режим доступа – <http://studopedia.info/2-82371.html>.

21. Дроздов А.Л., Коптелов А.К. Использование средств описания процессов при внедрении корпоративных информационных систем. Проблемы теории и практики управления, 2006, №10. [Электронный ресурс] Режим доступа – <http://businessprocess.narod.ru/index17.htm>.
22. ARIS (ARchitecture of Integrated Information Systems). Учебные материалы. Режим доступа – [http://bainr.ru/study\\_materials\\_methodology\\_ARIS.html](http://bainr.ru/study_materials_methodology_ARIS.html)
23. MathWorks. Центр компетенций. Модельно-ориентированное проектирование. Режим доступа – <http://matlab.ru/solutions/mbd/mbd>.
24. The Scrum Guide. The definitive Guide to Scrum: The Rules of the Game. Ken Schwaber, Jeff Sutherland. Режим доступа – <http://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-us.pdf>.
25. Х. Книберг, М. Скарин. Scrum и Kanban: выжимаем максимум. С4Media, Издательство InfoQ.com. Режим доступа – <http://www.infoq.com/resource/news/2010/01/kanban-scrum-minibook/en/resources/KanbanAndScrum-Russian.pdf>.
26. Тайчи Оно. Производственная система Тойоты. Уходя от массового производства. Библиотека ИКСИ. Издательство: Институт комплексных стратегических исследований. 2008. – 194 с.
27. Руководство к своду знаний по управлению проектами PMI PMBOK® Guide 5th Edition (2012). Режим доступа – [http://startupseminar.ru/ld/0/17\\_301907\\_2D9D3\\_pm.pdf](http://startupseminar.ru/ld/0/17_301907_2D9D3_pm.pdf).
28. Куперштейн В. И. Microsoft® Project 2013 в управлении проектами. – СПб.: БХВ-Петербург, 2014. – 432 с: ил.
29. Dia. Documentation. Manual. [Электронный ресурс]. Режим доступа – <http://dia-installer.de/doc/en/>.
30. Dia Tutorial (Учебник Dia). [Электронный ресурс]. Режим доступа – <http://younglinux.info/book/export/html/169>.
31. Project Planning with OpenProj the open source Project Management software Jürgen Bruns & Associates. [Электронный ресурс]. Режим доступа – <http://faculty.caes.uga.edu/pthomas/hort4091.web/Instructions.pdf>.
32. Маклаков, С.В. Создание информационных систем с AllFusion Modeling Suite / С.В. Маклаков. – М.: ДИАЛОГ-МИФИ, 2005. – 432 с.
33. Официальный русскоязычный сайт проекта Ramus. Режим доступа –
34. Официальный сайт проекта CA ERwin® Data Modeling. Режим доступа – <http://erwin.com/worldwide/russian-russia>.
35. Проектирование информационных систем с CA ERwin Modeling Suite 7.3 :учебное пособие / В.И. Горбаченко, Г.Ф. Убиенных, Г.В. Бобрышева – Пенза: Изд-во ПГУ, 2012. – 154 с.
36. Официальный сайт консалтинговой компании Business Process Solutions. Режим доступа – <http://bps.org.ua/aris.html>.
37. ARIS Community. Table of contents of ARIS Express help. [Электронный ресурс]. Режим доступа – <http://www.ariscommunity.com/help/aris-express>.
38. Сухарев М. С, Монахов Ю. М. Модель оценки функциональной устойчивости бизнес-процессов в условиях развитой телекоммуникационной инфраструктуры//Перспективные технологии в средствах передачи информации: Материалы IV Международной научно-технической конференции (г. Суздаль, 29 июня 1 июля 2011 года)/ Владимир-Суздаль: Изд-во ГОУ ВПО «Владимирский государственный университет им. А. Г. и Н. Г. Столетовых», 2011. 271 с.
39. Серова, Е.Г. Современные методологические и инструментальные подходы моделирования бизнес-задач / Е.Г. Серова // Int. Journal Information Technologies and Knowledge Decision Making and Business Intelligence Strategies and Techniques. 2008. – № 2. [Электронный ресурс]. Режим доступа – <http://simulation.su/uploads/files/default/2008-serova-1.pdf>.
40. Официальный сайт The AnyLogic Company. Учебные материалы. [Электронный ресурс]. Режим доступа – <http://www.anylogic.ru/books>.

## Приложение 1. Варианты заданий к лабораторным работам

Вариант 0. Конструкторское бюро по проектированию сложных технических объектов. Использование унифицированной САПР. Данные сохраняются в БД. Важность формирования целостного изображения.

Вариант 1. Станция экологического контроля параметров на химическом предприятии. ИС должна функционировать в оперативном режиме работы, решения об уровне загрязнения принимает система, подтверждает оператор. Необходимость защиты данных для хранения и при передаче на другие объекты.

Вариант 2. Предприятие по выпуску пищевой продукции обеспечено автоматизированной системой управления. Основные этапы технологического процесса: передача объектов по конвейеру, при котором необходим контроль качества выполненных операций. ИС собирает информацию со специальных датчиков и обрабатывает ее, сравнивая показатели объектов на конвейере с контрольными, при несовпадении отбраковывает, снимая с конвейера. Данные сохраняются в БД.

Вариант 3. Предприятие интернет торговли. Информационная система управляет пользователями, обеспечивает доступ, регистрацию и возможность скидок. Наличие конкурентов должно быть сопоставлено с возможностью защиты данных в режиме реального времени и их сохранности.

Вариант 4. Юридическая контора обслуживается ИС с двумя БД различного назначения (клиенты, их дела и нормативно-правовая информация). При этом БД связаны. БД клиент-дело особо секретна, БД справочная открыта и имеет доступ по всей юридической компании. Юристы, ведущие дела, могут оперативно добавлять информацию по делам по мобильной связи.

Вариант 5. Предприятие лечебно-диагностического характера. ИС клиентов, где рyanятся процедуры, описание диагнозов и процесса лечения. Поступление данных в БД из различных источников: датчики и устройства (напрямую с ультразвукового аппарата, от магнитно - резонансной диагностики, носимых устройств: например, пульс, частота дыхания, аритмия). Необходимость быстрого реагирования на особые ситуации в текущем режиме (информация с носимых устройств) и при диагностике.

Вариант 6. Администрация учебного заведения. ИС «школьный дневник» с возможностью обмена сообщениями с классным руководителем родителями учащегося при нестандартных ситуациях в оперативном режиме. Поступление данных с различных источников: проходная школы, журнал оценок, столовая, медицинский кабинет. Возможность сохранения и анализа данных за неделю, месяц, год.

Вариант 7. Предприятие контроля параметров технических устройств. Датчики и средства контроля передают комплекс информации о контролируемом устройстве в БД, откуда происходит их извлечение и анализ. Далее, на основании анализа формируется отчет о состоянии устройства. Возможность реализации контроля в скоростном режиме.

Вариант 8. Бюро технической инвентаризации (БТИ). Данные о состоянии жилого фонда вносятся как вручную, так и с помощью устройств фотосъемки, аудиозаписи (например, при определении шума при изоляции квартиры). Возможность передачи данных по локальной сети между подразделениями БТИ. Совместимость с другими БД.

Вариант 9. Станция гидрологического контроля параметров в морском порту. ИС должна функционировать в оперативном режиме работы, требуется наличие совместимых форматов с другими БД подобного профиля. Контроль параметров осуществляется как автоматически, так и вручную. Возможность предупреждения об опасных явлениях (резкое повышение уровня моря, увеличение скорости течения).

**Приложение 2**

Образец единого титульного листа к отчетам по лабораторным работам и  
практическим занятиям

Министерство образования и науки Российской Федерации  
ФГАОУ ВО «Севастопольский государственный университет»

Институт информационных технологий и управления в технических  
системах

Кафедра Информационных систем

**Сводный отчет по лабораторному практикуму  
по дисциплине «Системный анализ и проектирование  
информационных систем»**

№ п/п	Оценка выполнения			Дата	Подпись
	Теория	Выполнение	Итог		
1					
2					
3					
4					
5					
6					
7					
зачет					

Выполнил: студент(ка) группы

\_\_\_\_\_  
ФИО

Принял: должность ФИО

г. Севастополь

20\_\_ г.

Министерство образования и науки Российской Федерации  
ФГАОУ ВО «Севастопольский государственный университет»  
Институт информационных технологий и управления в технических системах

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
к РГР или (курсовому проекту)  
по дисциплине «Системный анализ и проектирование  
информационных систем»  
на тему: «                                »

Выполнил: ст. гр. ИС/м-XX-о(3)  
*ФИО студента*  
 Проверил:  
*ФИО преподавателя*

Севастополь  
20\_\_

## Приложение 4

Министерство образования и науки Российской Федерации  
ФГАОУ ВО «Севастопольский государственный университет»

Институт информационных технологий и управления в технических системах  
(название учебного заведения)

Кафедра \_\_\_\_\_ Информационные системы

**ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ**  
**по дисциплине «Системный анализ и проектирование**  
**информационных систем»**

**Ф И О**

(фамилия, имя, отчество)

1. Тема курсовой работы \_\_\_\_\_ АНАЛИЗ КОМПЛЕКСА ТЕХНОЛОГИЙ ДЛЯ  
ПРОЕКТИРОВАНИЯ ИС  
(ИТ): \_\_\_\_\_

**ТЕМА**

2. Срок сдачи студентом \_\_\_\_\_  
работы \_\_\_\_\_

3. Необходимые данные для выполнения работы

3.1. Постановка задачи. Входные данные: параметры исследуемого процесса, выходные данные: параметры ожидаемого результата;

3.2. Результаты выполнения лабораторных работ по дисциплине «Методы исследования информационных процессов и технологий»: схемы и модели локальных процессов;

3.3. \_\_\_\_\_

4. Содержание пояснительной записки (перечень вопросов для разработки)

1.Постановка задачи (Описание предметной области.Описание входных и выходных данных ИС или ИТ. Описание требований к ИС или ИТ. Описание ограничений при создании ИС или ИТ.)

2.Описание комплекса технологий для проектирования ИС и (или ИТ) (Обоснование выбора модели жизненного цикла при проектировании ИС или ИТ.Описание вложенных уровней проектирования ИС или ИТ. Обоснование комплекса технологий при проектировании ИС или ИТ. Формализованное описание комплекса технологий при проектировании ИС или ИТ.)

3. Оценка эффективности выбранного комплекса технологий при проектировании ИС или ИТ. (Выбор критериев оценки комплекса технологий при проектировании ИС или ИТ. Расчет показателя эффективности оценки комплекса технологий.)

4. Оценка возможных эксплуатационных характеристик ИС и(или) ИТ в задаче НИР. (Расчет информационных характеристик БД).

Заключение; Библиографический список; Приложения.

5. Список графического материала (схемы, графики, диаграммы)

5.1. Презентация доклада (от 5-10 слайдов по содержанию ПЗ);

6. Дата выдачи задания \_\_\_\_\_  
Руководитель \_\_\_\_\_/ФИО/

Задание принял для выполнения \_\_\_\_\_/ФИО/

## КАЛЕНДАРНЫЙ ПЛАН

	Название этапов курсовой работы	Сроки выполнения	Примечание
1	<b>1.Постановка задачи</b> 1.1.Описание предметной области 1.2.Описание входных и выходных данных ИС или ИТ 1.3.Описание требований к ИС или ИТ 1.4.Описание ограничений при создании ИС или ИТ		
2	<b>2.Описание комплекса технологий для проектирования ИС и (или) ИТ)</b> 2.1 Обоснование выбора модели жизненного цикла при проектировании ИС или ИТ		
3	2.2 Описание вложенных уровней проектирования ИС или ИТ 2.3 Обоснование комплекса технологий при проектировании ИС или ИТ		
4	2.4 Формализованное описание комплекса технологий при проектировании ИС или ИТ		
5	<b>3.Оценка эффективности выбранного комплекса технологий при проектировании ИС или ИТ</b> 3.1 Выбор критериев оценки комплекса технологий при проектировании ИС или ИТ		
6	3.2 Расчет показателя эффективности оценки комплекса технологий		
7	<b>4. Оценка возможных эксплуатационных характеристик ИС и(или) ИТ в задаче НИР</b> 4.1 Расчет информационных характеристик БД		
8	Оформление ПЗ		
9	Подготовка презентации доклада к защите курсового проекта		
10	Защита курсового проекта		

Студент \_\_\_\_\_/ФИО /

Руководитель \_\_\_\_\_/ФИО/