



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Магнитогорский государственный технический университет им. Г.И.  
Носова»



УТВЕРЖДАЮ  
Директор ИЭиАС  
В.Р. Храмшин

03.02.2026 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

***ПРОЕКТИРОВАНИЕ ПРОГРАММНЫХ СРЕДСТВ***

Направление подготовки (специальность)  
09.03.01 Информатика и вычислительная техника

Направленность (профиль/специализация) программы  
Программное обеспечение средств вычислительной техники и автоматизированных систем

Уровень высшего образования - бакалавриат

Форма обучения  
очная

Институт/ факультет	Институт энергетики и автоматизированных систем
Кафедра	Вычислительной техники и программирования
Курс	4
Семестр	7


Магнитогорск  
2026 год


Рабочая программа составлена на основе ФГОС ВО - бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника (приказ Минобрнауки России от 19.09.2017 г. № 929)

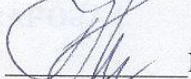
Рабочая программа рассмотрена и одобрена на заседании кафедры  
Вычислительной техники и программирования  
29.01.2026, протокол № 7

Зав. кафедрой  О.С. Логунова

Рабочая программа одобрена методической комиссией ИЭиАС  
03.02.2026 г. протокол № 5

Председатель  В.Р. Храмшин

Рабочая программа составлена:  
доцент кафедры ВТиП, канд. техн. наук  Ю.В. Кочержинская

Рецензент:  
директор НИИ "Промбезопасность", д-р техн. наук  М.Ю. Наркевич

## Лист актуализации рабочей программы

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2027 - 2028 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от \_\_\_\_\_ 20\_\_ г. № \_\_\_\_  
Зав. кафедрой \_\_\_\_\_ О.С. Логунова

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2028 - 2029 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от \_\_\_\_\_ 20\_\_ г. № \_\_\_\_  
Зав. кафедрой \_\_\_\_\_ О.С. Логунова

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2029 - 2030 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от \_\_\_\_\_ 20\_\_ г. № \_\_\_\_  
Зав. кафедрой \_\_\_\_\_ О.С. Логунова

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2030 - 2031 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от \_\_\_\_\_ 20\_\_ г. № \_\_\_\_  
Зав. кафедрой \_\_\_\_\_ О.С. Логунова

### **1 Цели освоения дисциплины (модуля)**

Целями освоения дисциплины «Проектирование программных средств» является ознакомление студентов с основами программной инженерии, формирование навыков разработки архитектуры программного обеспечения и создания технической и проектной документации к нему.

Для достижения поставленной цели в курсе «Проектирование программных средств» решаются задачи:

- формирование системного подхода к разработке программных продуктов;
- изучение видов, способов формирования и формализации требований к программным средствам;
- изучению форм проектирования программных систем;
- формирование навыков создания пре-кодовой документации к разрабатываемым программам.

### **2 Место дисциплины (модуля) в структуре образовательной программы**

Дисциплина Проектирование программных средств входит в часть учебного плана формируемую участниками образовательных отношений образовательной программы.

Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения дисциплин/ практик:

- Метрология и стандартизация программного обеспечения
- Системный анализ
- Программирование

Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы для изучения дисциплин/практик:

- Выполнение и защита выпускной квалификационной работы
- Подготовка к сдаче и сдача государственного экзамена
- Человеко-машинное взаимодействие

### **3 Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля) и планируемые результаты обучения**

В результате освоения дисциплины (модуля) «Проектирование программных средств» обучающийся должен обладать следующими компетенциями:

Код индикатора	Индикатор достижения компетенции
ПК-2	Способность к анализу проблемной ситуации, разработке требований к системе, постановке целей создания системы, разработке концепции системы и технического задания на создание системы, представления концепции, технического задания на систему и изменений в них заинтересованным лицам
ПК-2.1	Оценивает выбор средств и методов для проведения системного анализа при проектировании программного обеспечения для автоматизированных систем
ПК-9	Владеет навыками ввода в эксплуатацию аппаратных, программно-аппаратных и программных средств инфокоммуникационной инфраструктуры совместно с представителями поставщиков оборудования, готов к обслуживанию периферийного оборудования и организации инвентаризации технических средств
ПК-9.1	Оценивает качество ввода в эксплуатацию аппаратных, программно-аппаратных и программных средств инфокоммуникационной инфраструктуры
ПК-9.2	Оценивает качество обслуживания периферийного оборудования и организацию инвентаризации технических средств

#### 4. Структура, объём и содержание дисциплины (модуля)

Общая трудоемкость дисциплины составляет 3 зачетных единиц 108 академических часов, в том числе:

- контактная работа – 55 академических часов;
- аудиторная – 54 академических часов;
- внеаудиторная – 1 академический час;
- самостоятельная работа – 53 академических часов;
- в форме практической подготовки – 0 академических часов;

Форма аттестации - зачет

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в академических часах)			Самостоятельная работа студента	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код компетенции
		Лек.	лаб. зан.	практ. зан.				
1. Основы программной инженерии								
1.1 Введение в область разработки программных средств: определение, виды, общий жизненный цикл программного продукта	7	2			10	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронным учебником	1. Беседа - обсуждение 2. Устный опрос	ПК-2.1, ПК-9.1, ПК-9.2
1.2 Инженерия требований: типовой процесс, формализация, моделирование, инженерия в области проблем и решений, прослеживаемость и управленческие аспекты.		4	6		10	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронным учебником, выполнение лабораторных работ	1. Беседа - обсуждение 2. Устный опрос	ПК-2.1, ПК-9.1, ПК-9.2
Итого по разделу		6	6		20			
2. Пре-кодовая документация программных средств								
2.1 Виды и способы документирования на всех этапах жизненного цикла программных средств, исходные, сопровождающие и результирующие документы	7	4	2		10	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронным учебником, выполнение	1. Беседа - обсуждение 2. Устный опрос	ПК-2.1, ПК-9.1, ПК-9.2

						лабораторных работ		
2.2 Проектная документация: техническое задание на разработку программного изделия, отраслевые стандарты, способы написания, содержание	7	4	18		10	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронным учебником, выполнение лабораторных работ	1. Беседа - обсуждение 2. Устный опрос. 3. Тестовый опрос	ПК-2.1, ПК-9.1, ПК-9.2
Итого по разделу		8	20		20			
3. Архитектура программных средств								
3.1 Трансформация требований в архитектуру, классификация архитектур программных средств, стандарты на архитектуру программных средств	7	4	6			1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронным учебником, выполнение лабораторных работ	1. Беседа – обсуждение. 2. Устный опрос	ПК-2.1, ПК-9.1, ПК-9.2
3.2 Техническая документация: способы описания архитектуры программных средств (структурные, объектно-ориентированные, формальные)			4			13	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронным учебником, обсуждение лабораторных работ	1. Беседа – обсуждение. 2. Устный опрос. 3. Тестовый опрос
Итого по разделу		4	10		13			
Итого за семестр		18	36		53		зачёт	
Итого по дисциплине		18	36		53		зачет	

## **5 Образовательные технологии**

1. Традиционные образовательные технологии, ориентированные на организацию образовательного процесса и предполагающую прямую трансляцию знаний от преподавателя к студенту.

Формы учебных занятий с использованием традиционных технологий:

Информационная лекция – последовательное изложение материала в дисциплинарной логике, осуществляемое преимущественно вербальными средствами (монолог преподавателя).

Лабораторная работа – организация учебной работы с реальными материальными и информационными объектами, экспериментальная работа с аналоговыми моделями реальных объектов.

2. Технологии проблемного обучения – организация образовательного процесса, которая предполагает постановку проблемных вопросов, создание учебных проблемных ситуаций для стимулирования активной познавательной деятельности студентов.

3. Интерактивные технологии – организация образовательного процесса, которая предполагает активное и нелинейное взаимодействие всех участников, достижение на этой основе лично значимого для них образовательного результата.

Формы учебных занятий с использованием специализированных интерактивных технологий:

Лекция «обратной связи» – лекция–провокация (изложение материала с заранее запланированными ошибками), лекция-беседа, лекция-дискуссия, лекция-пресс-конференция.

4. Информационно-коммуникационные образовательные технологии – организация образовательного процесса, основанная на применении программных сред и технических средств работы с знаниями в различных предметных областях.

## **6 Учебно-методическое обеспечение самостоятельной работы обучающихся**

Представлено в приложении 1.

## **7 Оценочные средства для проведения промежуточной аттестации**

Представлены в приложении 2.

## **8 Учебно-методическое и информационное обеспечение дисциплины**

### **а) Основная литература:**

1. Черткова, Е. А. Программная инженерия. Визуальное моделирование программных систем : учебник для вузов / Е. А. Черткова. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 146 с. — (Высшее образование). — ISBN 978-5-534-18197-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/562413> (дата обращения: 10.03.2026).

2. Чернышев, С. А. Принципы, паттерны и методологии разработки программного обеспечения : учебник для вузов / С. А. Чернышев. — Москва : Издательство Юрайт, 2025. — 176 с. — (Высшее образование). — ISBN 978-5-534-14383-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/567946> (дата обращения: 10.03.2026).

### **б) Дополнительная литература:**

1. Золотухина, Е. Б. Управление жизненным циклом информационных систем (продвинутый курс): Конспект лекций / Золотухина Е.Б., Красникова С.А., Вишня А.С. - Москва :КУРС, НИЦ ИНФРА-М, 2017. - 119 с.: ISBN 978-5-906818-36-2. -

Текст : электронный. - URL: <https://znanium.com/catalog/product/767219> (дата обращения: 21.04.2024). – Режим доступа: по подписке.

2. Гагарина, Л. Г. Введение в архитектуру программного обеспечения : учебное пособие / Л. Г. Гагарина, А. Р. Федоров, П. А. Федоров. — Москва : ФОРУМ : ИНФРА -М, 2020. — 320 с. — (Высшее образование). - ISBN 978-5-8199-0649-1. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1046281> (дата обращения: 21.04.2024). – Режим доступа: по подписке.

**в) Методические указания:**

Приведены в Приложении 1.

**г) Программное обеспечение и Интернет-ресурсы:**

**Программное обеспечение**

Наименование ПО	№ договора	Срок действия лицензии
MS Office 2007 Professional	№ 135 от 17.09.2007	бессрочно
7Zip	свободно распространяемое ПО	бессрочно
Business Studio	Д №18У от 23.10.2007	бессрочно
Eclipse	свободно распространяемое ПО	бессрочно
ArgoUML	свободно распространяемое ПО	бессрочно
ARIS	свободно распространяемое ПО	бессрочно
Ramus 2.0.	свободно распространяемое ПО	бессрочно

**Профессиональные базы данных и информационные справочные системы**

Название курса	Ссылка
Национальная информационно-аналитическая система – Российский индекс научного цитирования (РИНЦ)	URL: <a href="https://elibrary.ru/project_risc.asp">https://elibrary.ru/project_risc.asp</a>

**9 Материально-техническое обеспечение дисциплины (модуля)**

Материально-техническое обеспечение дисциплины включает:

1. Лекционная аудитория ауд. 282. Мультимедийные средства хранения, передачи и представления информации.

2. Компьютерные классы Центра информационных технологий ФГБОУ ВО «МГТУ». Персональные компьютеры, объединенные в локальные сети с выходом в Internet, оснащенные современными программно-методическими комплексами для решения задач в области информатики и вычислительной техники.

3. Аудитории для самостоятельной работы: компьютерные классы; читальные залы библиотеки. Все классы УИТ и АСУ с персональными компьютерами, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета.

4. Аудиторий для групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Ауд. 282 и классы УИТ и АСУ.

5. Помещения для самостоятельной работы обучающихся, оснащенных компьютерной техникой с возможностью подключения к сети «Интернет» и наличием доступа в электронную информационно-образовательную среду организации. Классы УИТ и АСУ.

6. Помещения для хранения и профилактического обслуживания учебного оборудования. Центр информационных технологий – ауд. 372

## Приложение 1. Учебно-методическое обеспечение самостоятельной работы обучающихся

В течение семестра каждый студент выполняет лабораторные работы.

### Лабораторная работа № 1

Определение стейкхолдеров программного продукта

*Цель лабораторной работы* – выявить, упорядочить и описать стейкхолдеров для разрабатываемого программного продукта.

#### *Информация*

Когда мы говорим о некоторой системе, мы рассматриваем её с точки зрения некоторого стейкхолдера – лица (роли), которое заинтересовано в существовании этой системы. Стандарт ISO/IEC/IEEE 42010:2022 «Программное обеспечение, системы и предприятие» (введен в действие с 07.11.2022) называет стейкхолдерами системы следующие группы лиц:

- пользователи системы («users of the system»);
- операторы системы («operators of the system»);
- приобретатели системы («acquirers of the system»);
- владельцы системы («owners of the system»);
- поставщики системы («suppliers of the system»);
- разработчики системы («developers of the system»);
- производители системы («builders of the system»).

Нужно помнить, что каждая конкретная программная система может напрямую не взаимодействовать с какими-то группами интересантов лиц. Более того, при проектировании систем встречается ситуация, когда одни стейкхолдеры озвучивают свои интересы от имени других стейкхолдеров. Например, когда требования к функциональности и интерфейсу продукта формулирует, например, рабочая группа по обеспечению usability из подразделения разработки системы. То есть, за стейкхолдеров-пользователей говорят стейкхолдеры-разработчики. Или ситуация, когда приобретатели системы озвучивают «пожелания» её владельцев. А ведь ещё и разные стейкхолдерские роли в жизни может исполнять один человек, которому сложно переключаться в процессе определения требований. Тем не менее, идентификация стейкхолдеров программного продукта – важный этап на пути к его разработке.

Стейкхолдеры программного продукта имеют определенные цели, движущие мотивы и ожидают определенных результатов от взаимодействия с системой, интересантами которой они являются.

В качестве основы для достижения своих целей, у них есть определенные ресурсы (человеческие и иные), у исполняемых ими ролей есть определенные бизнес-возможности, которые и определяют вектор действий, расходование ресурсов и прочее. Представить эту информацию в графическом виде помогает методология моделирования ArchiMate, графическая нотация которой позволяет работать с различными слоями (уровнями) корпоративной архитектуры внутри определенных бизнес-процессов. Ниже (рисунок) приведен пример диаграммы целей для нескольких стейкхолдеров на достаточно высоком уровне организационной архитектуры. Диаграмма подготовлена средствами онлайн-редактора VirtualParadigm (<https://online.visual-paradigm.com/ru/>), имеющего встроенные инструменты для создания ArchiMate-диаграмм.

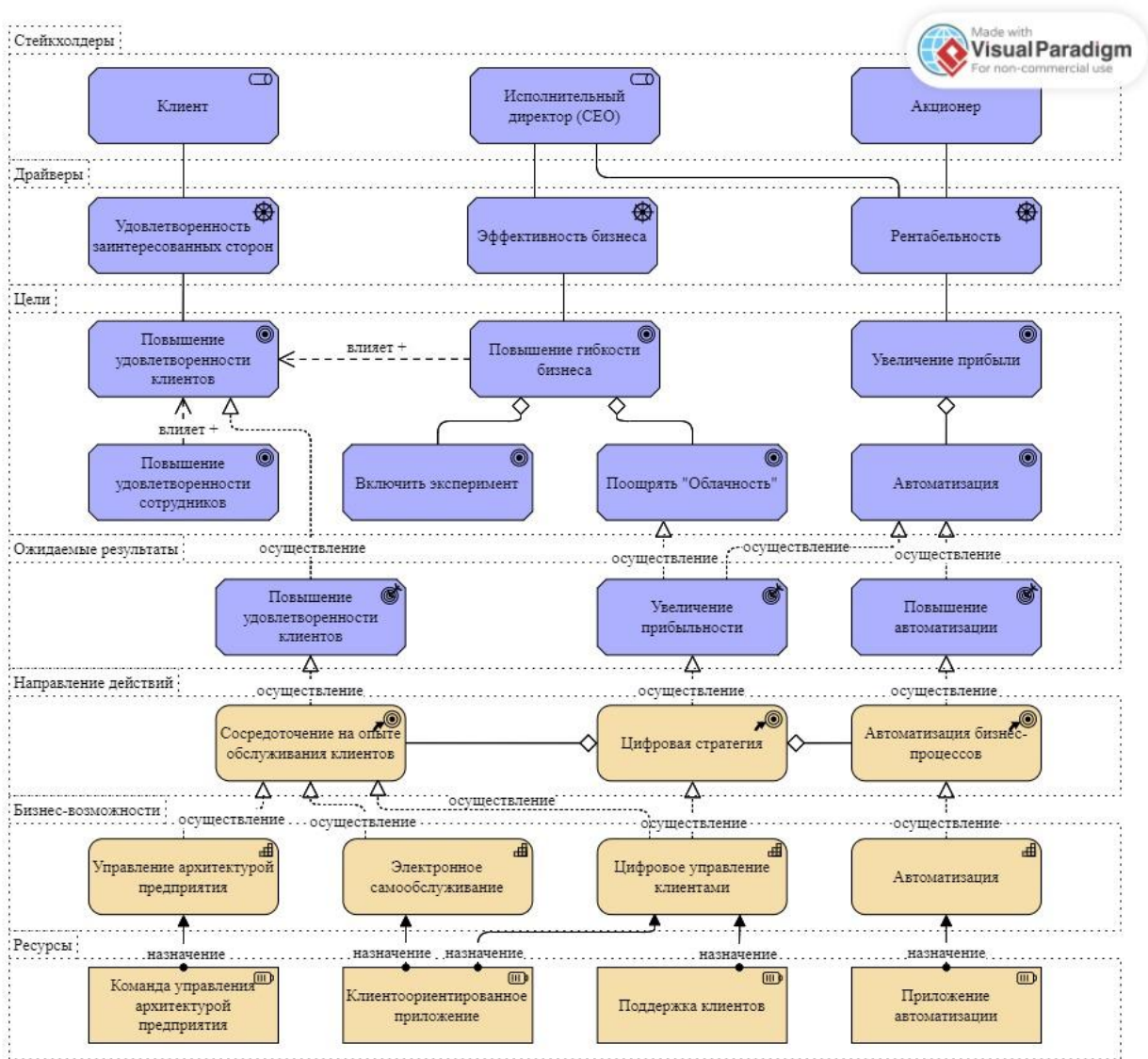


Рисунок. Модель мотивационных слоев разработки цифровой системы

Аналогичные диаграммы можно сделать для более низких уровней вплоть до уровня архитектуры конкретного приложения с визуализацией взаимодействия его компонентов.

### Задание

Используя нотацию ArchiMate создать развернутую диаграмму определения целей разработки программного продукта/системы, предварительно идентифицировав все группы стейкхолдеров вашего программного продукта, их движущие мотив, цели, ресурсы и т.д. Обосновать свои решения.

## Лабораторная работа №2

### Сбор и анализ требований к программному обеспечению

**Цель:** научиться собирать требования к программному продукту, выполнять их предварительный анализ и формировать спецификацию требований.

### Информация

Сбор требований Заказчика подразумевает обучение общению с профессионалом в иной области, нежели разработчик и способность самостоятельного уточнения поставленной задачи. Сбор требований Разработчика учит грамотно оценивать необходимые и достаточные интеллектуальные и технические ресурсы, которые потребуются для разработки ПО.

Среди обязательных для идентификации D-требований (требований разработчика) можно выделить:

- ✓ производительность;
- ✓ надежность и доступность;
- ✓ безопасность;
- ✓ удобство и простота обслуживания.

Для разработчиков и специалистов по обслуживанию от системы требуются и другие характеристики. Часто они называются атрибутами качества разрабатываемой системы. Приведем некоторые из таких атрибутов качества:

- ✓ легкость сопровождения и эксплуатации;
- ✓ мобильность;
- ✓ повторное использование;
- ✓  тестируемость.

### **Производительность**

Требования к производительности определяют насколько быстро и качественно система должна выполнять определенные функции. Они определяют время отклика, пропускную способность и т.д. Жесткие требования к производительности существенно влияют на выбор аппаратных средств, технологию разработки и принимаемые инженерные решения при реализации.

### **Надежность и доступность**

*Надежность* – это вероятность работы системы без сбоев в течение определенного времени. Для измерения надежности может быть использовано среднее время работы системы до сбоя.

Под доступностью понимается *время доступности*, т.е. время, в течение которого система доступна для использования и полностью работоспособна. Это время определяется средним временем до сбоя и зависит от времени планового технического обслуживания.

### **Безопасность**

Эти требования связаны с блокировкой неавторизованного доступа к данным и функциям системы, предотвращением потерь информации и т.п.

### **Удобство и простота обслуживания**

Этот атрибут связан с большим числом факторов, определяющих, по словам пользователей, *дружелюбие* системы к пользователю. Другими словами, система должна использоваться эффективно и необременительно.

### **Легкость сопровождения и эксплуатации**

Этот атрибут определяет насколько просто и удобно модифицировать продукт и исправлять найденные в нем ошибки. Он важен для продуктов, которые подвергаются частым изменениям.

### **Мобильность**

Этот атрибут определяет усилия, необходимые для перенесения продукта из одной операционной среды в другую. Важно на этапе разработке требований точно определить те среды и, возможно, части системы, которые должны быть перемещаемыми.

### **Повторное использование**

Затраты на разработку повторно используемых компонент сравнительно велики, но эффект их использования в дальнейшем может компенсировать эти затраты. Для минимизации затрат в требованиях необходимо перечислить элементы проекта, которые должны быть спроектированы так, чтобы упростить их повторное использование.

### **Тестируемость**

Этот атрибут показывает легкость, с которой компоненты проекта и комплексный

продукт могут быть проверены на наличие ошибок.

В зависимости от вида и статуса разрабатываемой системы, могут собираться и другие требования разработчика.

Все собранные С- и D-требования заносятся в специальный документ, который называется «Спецификация требований».

Спецификация требований представляет собой итоговый набор расположенных по приоритетам требований, который является формальным соглашением заказчика с разработчиком системы.

Требования к разрабатываемой системе должны обладать следующими характеристиками:

1. Недвусмысленность;
2. Проверяемость;
3. Четкость (краткость);
4. Точность;
5. Понятность;
6. Осуществимость;
7. Независимость;
8. Атомарность;
9. Необходимость;
10. Абстрактность. Рассмотрим подробнее, что из себя представляют эти требования.

1. Недвусмысленность – должна существовать только одна трактовка требования. Так, например, следует избегать в тексте требования сокращений.

*Пример плохого требования: R1 Система не должна принимать пароль длиннее 15 символов.*

Данное требование не проясняет, что же должна делать система:

- ✓ Система не должна позволять пользователю вводить пароль длиннее 15 символов.
- ✓ Система должна обрезать введенный пароль до 15 символов.
- ✓ Система должна выводить сообщение об ошибке в случае если пользователь введет больше 15 символов.

*Пример хорошего требования: R1 Система не должна принимать пароль длиннее 15 символов. Если пользователь введет больше 15 символов, система должна отобразить сообщение об ошибке с просьбой исправить пароль.*

2. Проверяемость – тестировщики должны иметь возможность проверить правильно ли требование реализовано в системе. Для этого требование должно быть четким, точным, недвусмысленным.

*Пример плохого требования: R1 Поиск в системе должен осуществляться по имени, фамилии пользователя и т.д.*

В данном примере критерии поиска должны быть раскрыты полностью. Иначе нельзя проверить выполняется ли требование в системе.

3. Четкость (краткость) – требование не должно содержать лишней информации. Оно должно быть изложено четко и просто.

*Пример плохого требования: R1 Пользователь иногда может вводить код аэропорта, который должен определяться системой. Но также иногда код аэропорта должен заменяться названием ближайшего города, таким образом пользователю не требуется знать код аэропорта, но система по-прежнему должна знать код аэропорта.*

Это требование может быть изменено следующим образом: *R1 Система должна определять аэропорт, как по коду аэропорта, так и по названию города.*

4. Точность – требование должно содержать в себе истинные факты.
5. Понятность – требование не должно содержать грамматических ошибок, должно быть изложено последовательно.
6. Осуществимость – требование должно быть выполнимо в рамках существующих ограничений (время, деньги, существующие ресурсы)
7. Независимость – для понимания требования не нужно знать других требований.

*Пример плохого требования: R1 Список доступных рейсов должен содержать информация о номере полета, времени посадки и приземления. R2 Они должны быть отсортированы по цене.*

8. Атомарность – требование должно содержать одну связанную сущность.

*Пример плохого требования: R1 В системе должна быть предусмотрена возможность забронировать полет, купить билет, зарезервировать номер в отеле, арендовать машину.*

9. Необходимость – заинтересованные лица должны нуждаться в данном требовании. 10. Абстрактность – в требовании не должно содержаться информации о том, как оно будет реализовано.

Также существуют определенные характеристики, которыми должен обладать набор требований:

- ✓ постоянство – не должно быть конфликтов между требованиями.
- ✓ неизбыточность – каждое требование должно быть описано один раз и не должно содержать в себе других требований.
- ✓ полнота – требование должно быть определено для всех возможных ситуаций.

Существуют дополнительные критерии, но они содержат в себе описанные выше, например:

- ✓ модифицируемость – если требование атомарно и полное, оно модифицируемо.
- ✓ трассировка (прослеживаемость) – если требование атомарно и имеет уникальный идентификатор, оно прослеживаемо.

*Состав спецификации требований*

#### 1. Идентификация сторон разработки.

Здесь приводится информация о том, кто является заказчиком системы и на основании чего будет вестись разработка, список уполномоченных лиц и их функциональных обязанностей; а также информация о разработчике системы, какие люди и их функциональные обязанности работают над требованиями, границы полномочий разработчика.

НЕ приводятся реквизиты, адреса юрлиц и т.д. Может приводиться контактная информация перечисленных людей с обеих сторон.

Например:

Иванов Андрей Петрович, ведущий дизайнер группы автоматизации производственного процесса Заказчика, ответственность: дизайн экранного интерфейса программы, конт. тел. 89164726593.

Понятно, что информация может быть расположена в табличном виде. Часто упорядочивается по важности и влиянию на проект того или иного персонажа.

#### 2. Требования заказчика

- 1 подраздел
  - 2 подраздел
  - 3 подраздел...
- и т.д.

#### 3. Требования разработчика

- 1 подраздел
  - 2 подраздел
  - 3 подраздел...
- и т.д.

Требования упорядочиваются по важности, среди них нужно выделить ключевые требования (в каждом подразделе).

Требования можно условно разделить на функциональные, структурные и требования к дизайну.

По важности требования разделяют на Обязательные (безусловные), желательные (обсуждаемые) и условные. 3. Приложения

Объемная иллюстрирующая информация к требованиям выносится в приложения

к списку требований. Приложения формируются с учётом общей логики изложения требований, т.е. информация, относящаяся к одному источнику, логически привязанная к одному объекту или однородная по своей сути объединяется в одном приложении.

Приложения именуется заглавными буквами русского алфавита А, Б, В, ... и т.д. Информация внутри них нумеруется арабскими цифрами «1», «2», «3», ... и т.д.

Обычно если есть проекты интерфейса, цветовые схемы и т.п., то они заносятся в Приложение А, кроме номера рисунка, схемы и т.д., возле каждого из них делается краткая поясняющая надпись, например: «Схема расположения управляющих элементов в основном рабочем окне программы», дополнительно может из списка требований дублироваться его источник/функциональная роль, например: «ведущий дизайнер рабочей группы Заказчика»).

Если в тексте требований есть ссылки на любые стандарты, инструкции и т.д., то в тексте возле каждого из них дается ссылка вида [В1], [Б8], [Г13], а подробная информация о источнике приводится в приложении (одном или нескольких) с указанием всех выходных данных этого источника:

Авторы, название (полностью, если есть и уточняющее название, тоже приводится), категория документа, кем и когда издано, сроки и область действия и т.д. Похоже на оформление списка литературы, но в отличие от него никаких сокращений и т.д. здесь не делается, указывается дополнительная информация об актуальности.

Обычно, если такого рода информации много, она объединяется по нескольким приложениям. Например, стандарты в «Приложение Б», инструкции в «Приложение В» и т.д.

### **Задание**

Для заданной преподавателем темы произвести:

- 1) сбор (*пороговый уровень*)
- 2) систематизацию (*средний уровень*)
- 3) анализ и обобщение (*высокий уровень*)  
требований заказчика и разработчика.

### **Контрольные вопросы**

1. Что такое требования?
2. Для чего нужно собирать требования?
3. На каких этапах жизненного цикла ПО целесообразно производить сбор требований?
4. В чем заключаются и что могут содержать требования заказчика?
5. Есть ли различия между заказчиком системы и её пользователем?

## **ЛАБОРАТОРНАЯ РАБОТА №3**

Разработка технического задания на программный продукт, с использованием  
ГОСТ 34.602-2020

*Цель работы:* научиться разрабатывать техническое задание для программных продуктов в соответствии с действующими стандартами.

### *Информация*

Основным документом, регламентирующим состав и содержание технического задания на автоматизированные системы, является ГОСТ 34.602-2020.

ГОСТ 34.602-2020 введен в действие 1 января 2022 года взамен ГОСТ 34.602-89 и распространяется на автоматизированные системы (АС), предназначенные для автоматизации различных видов деятельности (управление, проектирование, исследования и т. п.), включая их сочетания, устанавливает требования к составу, содержанию, правилам оформления документа «Техническое задание на создание (развитие или модернизацию) автоматизированной системы» (далее – ТЗ на АС).

ТЗ на АС разрабатывают на систему в целом, предназначенную для работы самостоятельно или в составе другой системы.

Логически состав и содержание ТЗ на АС можно разбить на несколько частей: общую, специальную и организационную.

К общей части относятся следующие разделы:

- общие сведения;
- цели и назначение создания автоматизированной системы;
- характеристика объектов автоматизации;
- источники разработки.

К специальной части относятся разделы:

- требования к автоматизированной системе;
- требования к документированию.

К организационной части относятся разделы:

- состав и содержание работ по созданию автоматизированной системы;
- порядок разработки автоматизированной системы;
- порядок контроля и приемки автоматизированной системы;
- требования к составу и содержанию работ по подготовке объекта автоматизации к вводу автоматизированной системы в действие.

**В ТЗ на АС все разделы включаются в том порядке, в котором они приведены в ГОСТ 34.602-2020:**

- общие сведения;
- цели и назначение создания автоматизированной системы;
- характеристика объектов автоматизации;
- требования к автоматизированной системе;
- состав и содержание работ по созданию автоматизированной системы;
- порядок разработки автоматизированной системы;
- порядок контроля и приемки автоматизированной системы;
- требования к составу и содержанию работ по подготовке объекта автоматизации к вводу автоматизированной системы в действие;
- требования к документированию;
- источники разработки.

В ТЗ на АС могут быть включены приложения.

**В случае отсутствия требований по разделу, соответствующий раздел сохраняется, и в нем приводится запись об отсутствии требований.**

В зависимости от вида, назначения, специфических особенностей объекта автоматизации и

условий функционирования АС допускается **оформлять разделы ТЗ в виде приложений, вводить дополнительные разделы ТЗ.**

Разделы ТЗ могут быть разделены на подразделы. Допускается вводить дополнительные, **исключать или объединять подразделы ТЗ.**

Рассмотрим содержание разделов подробнее (можно просто внимательно прочитать ГОСТ 34.602-2020).

1. В разделе «Общие сведения» указывают следующее:

- полное наименование АС и ее условное обозначение;
- шифр темы (при наличии);
- наименование организации-заказчика АС, наименование организации-разработчика (при наличии сведений о ней);
- перечень документов, на основании которых создается АС, кем и когда утверждены эти документы;
- плановые сроки начала и окончания работ по созданию АС;
- общие сведения об источниках и порядке финансирования работ.

*Примечание.* К документам, на основании которых или в соответствии с которыми создается АС, могут относиться, например, следующие:

- договорные документы на создание АС;

- нормативно-правовые и нормативно-технические документы, регламентирующие создание АС;
- техническое задание на создание ранее разрабатывавшейся АС.

2. **Раздел** «Цели и назначение создания автоматизированной системы» состоит из следующих подразделов:

- цели создания АС;
- назначение АС.

2.1 В подразделе «Цели создания АС» приводят наименования и требуемые значения технических, технологических, производственно-экономических или других показателей объекта автоматизации, которые должны быть достигнуты в результате создания АС, и указывают критерии оценки достижения целей создания АС.

2.2 В подразделе «Назначение АС» указывают вид автоматизируемой деятельности (управление, проектирование и т. п.) применительно к объекту автоматизации в целом. Для сложного объекта автоматизации приводится общий перечень объектов, на которых планируется использовать АС.

3. В **разделе** «Характеристика объекта автоматизации» приводят следующую информацию:

- основные сведения об объекте автоматизации или ссылки на документы, содержащие такие сведения;
- сведения об условиях эксплуатации объекта автоматизации и характеристиках окружающей среды.

Примечание. В разделе приводят основные сведения об объекте автоматизации, позволяющие однозначно его идентифицировать и сформировать правильное представление о масштабах разработки.

4. **Раздел** «Требования к автоматизированной системе» состоит из следующих подразделов:

- требования к структуре АС в целом;
- требования к функциям (задачам), выполняемым АС;
- требования к видам обеспечения АС;
- общие технические требования к АС.

Состав требований к АС, включаемых в данный раздел ТЗ на АС, устанавливают в зависимости от вида, назначения, специфических особенностей и условий функционирования конкретной автоматизированной системы. В каждом подразделе приводят ссылки на действующие нормативно-технические документы, определяющие требования к автоматизированным системам соответствующего вида.

4.1 В **подразделе** «Требования к структуре АС в целом» указывают следующее:

- перечень подсистем (при их наличии), их назначение и основные характеристики. Дополнительно могут быть приведены **требования к числу уровней иерархии и степени централизации АС**;
- требования к способам и средствам обеспечения информационного взаимодействия компонентов АС;
- требования к характеристикам взаимосвязей создаваемой АС со смежными АС, требования к интероперабельности, требования к ее совместимости, в том числе указания о способах обмена информацией;
- требования к режимам функционирования АС;
- требования по диагностированию АС;
- перспективы развития, модернизации АС.

4.2 В **подразделе** «Требования к функциям (задачам), выполняемым АС», приводят перечень функций (задач), подлежащих автоматизации для АС в целом или для каждой подсистемы (при их наличии). В перечень включаются в том числе функции (задачи), обеспечивающие взаимодействие частей АС.

Для каждой функции (задачи) должен быть указан результат ее выполнения и, при необходимости, приведены основные характеристики результата.

При необходимости дополнительно могут быть указаны следующие данные:

- временной регламент реализации каждой функции (задачи);
- требования к реализации каждой функции (задачи), к форме представления

выходной информации, характеристики необходимой точности и времени выполнения, требования одновременности выполнения группы функций, достоверности выдачи результатов;

- перечень и критерии отказов для каждой функции, по которой задаются требования по надежности.

4.3 В подразделе «Требования к видам обеспечения АС» приводят требования к математическому, информационному, лингвистическому, программному, техническому, метрологическому, организационному, методическому и другим видам обеспечения АС.

4.3.1 Для *математического* обеспечения АС приводят требования к составу, области применения (ограничениям) и способам использования в АС математических методов и моделей, типовых алгоритмов и алгоритмов, подлежащих разработке.

4.3.2 Для *информационного* обеспечения АС приводят следующие требования:

- к составу, структуре и способам организации данных в АС;
- к информационному обмену между компонентами АС и со смежными АС;
- к информационной совместимости со смежными АС;
- по использованию действующих и по разработке новых классификаторов, справочников, форм документов;
- по применению систем управления базами данных;
- к представлению данных в АС;
- к контролю, хранению, обновлению и восстановлению данных.

4.3.3 Для *лингвистического* обеспечения АС приводят следующие требования:

- к языкам, используемым в АС, и возможности расширения набора языков (при необходимости);
- к способам организации диалога;
- к разработке и использованию словарей, тезаурусов;
- к описанию синтаксиса формализованного языка.

4.3.4 Для *программного* обеспечения АС приводят следующую информацию:

- требования к составу и видам программного обеспечения;
- требования к выбору используемого программного обеспечения;
- требования к разрабатываемому программному обеспечению;
- перечень допустимых покупных программных средств (при наличии).

4.3.5 Для *технического* обеспечения АС приводят следующие требования:

- к видам технических средств, в том числе к видам комплексов технических средств, программно-технических комплексов и других комплектующих изделий, допустимых к использованию в АС;
- к функциональным, конструктивным и эксплуатационным характеристикам средств технического обеспечения АС.

4.3.6 В требованиях к *метрологическому* обеспечению АС приводят следующую информацию:

- количественные значения показателей метрологического обеспечения;
- требования к методам (методикам) измерений и измерительного контроля параметров и их характеристик;
- требования к средствам измерений и измерительного контроля;
- требования к метрологическому обеспечению испытаний АС;
- требования к программе метрологического обеспечения АС;
- требования к метрологической совместимости технических средств АС;
- требования проведения метрологической экспертизы технической документации (при необходимости).

4.3.7 Для *организационного* обеспечения АС приводят следующие требования:

- к структуре и функциям подразделений, участвующих в функционировании АС или обеспечивающих эксплуатацию;
- к организации функционирования АС и порядку взаимодействия персонала и пользователей АС;
- к организации функционирования АС при сбоях, отказах и авариях;
- к порядку обеспечения нормативными документами, необходимыми для

разработки АС.

4.3.8 Для **методического** обеспечения АС приводят следующую информацию:

- перечень применяемых при разработке и функционировании АС нормативно-технических документов (стандартов, нормативов, методик, профилей и т. п.);
- порядок и правила обеспечения разработчиков АС нормативно-технической документацией.

4.4 В **подразделе** «Общие технические требования к АС» указывают следующее:

- требования к численности и квалификации персонала и пользователей АС;
- требования к показателям назначения;
- требования к надежности;
- требования по безопасности;
- требования к эргономике и технической эстетике;
- требования к транспортабельности для подвижных АС;
- требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов АС;
- требования к защите информации от несанкционированного доступа;
- требования по сохранности информации при авариях;
- требования к защите от влияния внешних воздействий;
- требования к патентной чистоте и патентоспособности;
- требования по стандартизации и унификации;
- дополнительные требования.

4.4.1 В требованиях к **численности и квалификации персонала** и пользователей АС приводят следующее:

- требования к численности персонала и пользователей АС;
- требования к квалификации персонала и пользователей АС, порядку их подготовки и контроля знаний и навыков;
- требуемый режим работы персонала и пользователей АС.

4.4.2 В требованиях к **показателям назначения АС** приводят значения параметров, характеризующих степень соответствия АС ее назначению (при их наличии).

4.4.3 В требованиях к **надежности** включают:

- состав и количественные значения показателей надежности для АС в целом или ее подсистем (составных частей);
- перечень аварийных ситуаций, по которым должны быть регламентированы требования к надежности, и значения соответствующих показателей;
- требования к надежности технических средств и программного обеспечения;
- требования к методам оценки и контроля показателей надежности на разных стадиях создания АС в соответствии с действующими нормативно-техническими документами.

4.4.4 В требованиях по **безопасности** включают требования по обеспечению безопасности при монтаже, наладке, эксплуатации, обслуживании и ремонте технических средств АС (защита от воздействий электрического тока, электромагнитных полей и т. п.), по допустимым уровням вибрационных и шумовых нагрузок, а также по обеспечению экологической безопасности.

4.4.5 В требованиях к **эргономике и технической эстетике** включают следующие требования:

- эргономические требования к организации и средствам деятельности персонала и пользователей АС, в том числе к средствам отображения информации и организации рабочего места;
- требования к технической эстетике, определяющие композиционную целостность, информационную выразительность, рациональность формы и культуру производственного исполнения создаваемого изделия, в том числе реализации человеко-машинного интерфейса.

4.4.6 В требованиях к **транспортабельности для подвижных АС** включают конструктивные требования, обеспечивающие транспортабельность технических средств АС, а также требования к транспортным средствам, включая условия

транспортирования, возможность перевозки в готовом к функционированию состоянии, необходимость защиты элементов АС от внешних воздействующих факторов при транспортировании, а также требования безопасности перевозки.

4.4.7 В требования к **эксплуатации, техническому обслуживанию, ремонту и хранению компонентов АС** включают:

- условия и регламент (режим) эксплуатации, которые должны обеспечивать использование технических средств (ТС) и программно-технических средств (ПТС) АС с заданными показателями;
- требования к видам, периодичности и объему технического обслуживания, контролю технического состояния и ремонта или допустимость работы без обслуживания;
- предварительные требования к допустимым площадям для размещения персонала и технических средств АС, к параметрам сетей энергоснабжения, вентиляции, охлаждения и т. п.;
- требования к составу, размещению и условиям хранения комплекта запасных частей, инструментов и принадлежностей, а также к нормам расхода запасных частей;
- требования к регламенту обслуживания.

4.4.8 В требования к **защите информации от несанкционированного доступа** включают требования, установленные в НТД, действующей в отрасли (ведомстве) заказчика.

4.4.9 В требованиях по **сохранности информации** приводят перечень событий: аварий, отказов технических средств (в том числе – потеря питания) и т. п., при которых должна быть обеспечена сохранность информации в АС.

4.4.10 В требованиях к **защите от внешних воздействий** приводят:

- требования к радиоэлектронной защите средств АС;
- требования по стойкости, устойчивости и прочности к внешним воздействиям (среде применения).

4.4.11 В требованиях к **патентной чистоте и патентоспособности** указывают требования по патентной чистоте и патентоспособности АС и ее частей, включая требования по проведению патентных исследований.

4.4.12 В требованиях к **стандартизации и унификации** включают показатели, устанавливающие следующее:

- требуемую степень использования стандартных, унифицированных методов реализации функций (задач) АС, поставляемых программных средств, типовых математических методов и моделей, типовых проектных решений, унифицированных форм документов, общероссийских классификаторов и классификаторов других категорий в соответствии с областью их применения;
- требования к использованию типовых автоматизированных рабочих мест, компонентов и комплексов.

4.4.13 В **дополнительные** требования включают:

- требования к оснащению АС учебно-тренировочными средствами и документацией на них;
- требования к сервисной аппаратуре, стендам для проверки элементов АС;
- требования к АС, связанные с особыми условиями эксплуатации;
- специальные требования по усмотрению разработчика или заказчика АС.

5. Раздел «Состав и содержание работ по созданию автоматизированной системы» должен содержать перечень этапов работ по созданию АС и сроки их выполнения.

6. В разделе «Порядок разработки автоматизированной системы» приводят следующее:

- порядок организации разработки АС;
- перечень документов и исходных данных для разработки АС;
- перечень документов, предъявляемых по окончании соответствующих этапов работ;
- порядок проведения экспертизы технической документации;
- перечень макетов (при необходимости), порядок их разработки, изготовления, испытаний, необходимость разработки на них документации, программы и

- методик испытаний;
- порядок разработки, согласования и утверждения плана совместных работ по разработке АС;
- порядок разработки, согласования и утверждения программы работ по стандартизации;
- требования к гарантийным обязательствам разработчика;
- порядок проведения технико-экономической оценки разработки АС;
- порядок разработки, согласования и утверждения программы метрологического обеспечения, программы обеспечения надежности, программы эргономического обеспечения.

7. В разделе «Порядок контроля и приемки автоматизированной системы» указывают следующую информацию:

- виды, состав и методы испытаний АС и ее составных частей;
- общие требования к приемке работ, порядок согласования и утверждения приемочной документации;
- статус приемочной комиссии (государственная, межведомственная, ведомственная и др.).

*Примечание.* Порядок согласования и утверждения приемочной документации, а также статус приемочной комиссии указываются при необходимости.

8. В разделе «Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу автоматизированной системы в действие» приводят перечень мероприятий, которые необходимо осуществить при подготовке объекта автоматизации к вводу АС в действие.

В перечень мероприятий включают следующее:

- создание условий функционирования объекта автоматизации, при которых гарантируется соответствие создаваемой АС требованиям, содержащимся в ТЗ на АС;
- проведение необходимых организационно-штатных мероприятий;
- порядок обучения персонала и пользователей АС.

9. В разделе «Требования к документированию» приводят следующую информацию:

- перечень подлежащих разработке документов;
- вид представления и количество документов;
- требования по использованию ЕСКД и ЕСПД при разработке документов.

При отсутствии государственных стандартов, определяющих требования к документированию элементов АС, дополнительно включают требования к составу и содержанию таких документов.

10. В разделе «Источники разработки» должны быть перечислены документы и информационные материалы (технико-экономическое обоснование, отчеты о законченных научно-исследовательских работах, информационные материалы на отечественные, зарубежные системы-аналоги и др.), на основании которых разрабатывалось ТЗ и которые должны быть использованы при создании АС.

### **Замечания по оформлению ТЗ на АС:**

Разделы и подразделы ТЗ на АС должны быть размещены в порядке, установленном в разделах 1-10 лабораторной работы (им соответствует раздел 4 ГОСТ 34.602-2020).

ТЗ на АС оформляют в виде текстового документа.

При необходимости в ТЗ на АС могут включаться схемы, рисунки, таблицы и др. иллюстративный материал.

Значения показателей, норм и требований указывают, как правило, с предельными отклонениями или максимальным и минимальным значениями. Если эти показатели, нормы, требования однозначно регламентированы НТД, в ТЗ на АС следует приводить ссылку на эти документы или их разделы, а также дополнительные требования, учитывающие особенности создаваемой АС. Если конкретные значения показателей, норм и требований не могут быть установлены в процессе разработки ТЗ на АС, в нем следует сделать запись о порядке установления и согласования этих показателей, норм и

требований: «Окончательное требование (значение) уточняется... и согласовывается ...». При этом в текст ТЗ на АС изменений не вносят.

### *Задание*

Используя приведенный ГОСТ, разработать техническое задание на свой программный продукт. Используйте, в том числе, материалы ранее сделанных лабораторных работ по курсам «Проектирование программных средств», «Человеко-машинное взаимодействие», «Базы данных», «Управление сложными системами» и др.

## **Лабораторная работа №4**

Оформление алгоритмов в соответствии с ГОСТ 19.701-90

*Цель работы:* научиться правильно использовать нотации стандарта.

### *Информация*

ГОСТ 19.701-90 охватывает построение следующих графических схем: схема данных, схема программы, схема работы системы, схема взаимодействия программ, схема ресурсов системы.

#### **1. Схема данных**

1.1. Схемы данных отображают путь данных при решении задач и определяют этапы обработки, а также различные применяемые носители данных.

1.2. Схема данных состоит из:

- 1) символов данных (символы данных могут также указывать вид носителя данных);
- 2) символов процесса, который следует выполнить над данными (символы процесса могут также указывать функции, выполняемые вычислительной машиной);
- 3) символов линий, указывающих потоки данных между процессами и (или) носителями данных;
- 4) специальных символов, используемых для облегчения написания и чтения схемы.

1.3. Символы данных предшествуют и следуют за символами процесса. Схема данных начинается и заканчивается символами данных (за исключением специальных символов, указанных в п. 3.4 ГОСТа, выложен на Образовательном портале в разделе Справочные материалы).

#### **2. Схема программы**

2.1. Схемы программ отображают последовательность операций в программе.

2.2. Схема программы состоит из:

- 1) символов процесса, указывающих фактические операции обработки данных (включая символы, определяющие путь, которого следует придерживаться с учетом логических условий);
- 2) линейных символов, указывающих поток управления;
- 3) специальных символов, используемых для облегчения написания и чтения схемы.

#### **3. Схема работы системы**

3.1. Схемы работы системы отображают управление операциями и поток данных в системе.

3.2. Схема работы системы состоит из:

- 1) символов данных, указывающих на наличие данных (символы данных могут также указывать вид носителя данных);
- 2) символов процесса, указывающих операции, которые следует выполнить над данными, а также определяющих логический путь, которого следует придерживаться;
- 3) линейных символов, указывающих потоки данных между процессами и (или) носителями данных, а также поток управления между процессами;

- 4) специальных символов, используемых для облегчения написания и чтения блок-схемы.

#### **4. Схема взаимодействия программ**

4.1. Схемы взаимодействия программ отображают путь активаций программ и взаимодействий с соответствующими данными. Каждая программа в схеме взаимодействия программ покатывается только один раз (в схеме работы системы программа может изображаться более чем в одном потоке управления).

4.2. Схема взаимодействия программ состоит из:

- 1) символов данных, указывающих на наличие данных;
- 2) символов процесса, указывающих на операции, которые следует выполнить над данными;
- 3) линейных символов, отображающих поток между процессами и данными, а также инициации процессов;
- 4) специальных символов, используемых для облегчения написания и чтения схемы.

#### **5. Схема ресурсов системы**

5.1. Схемы ресурсов системы отображают конфигурацию блоков данных и обрабатывающих блоков, которая требуется для решения задачи или набора задач.

5.2. Схема ресурсов системы состоит из:

- 1) символов данных, отображающих входные, выходные и запоминающие устройства вычислительной машины;
- 2) символов процесса, отображающих процессоры (центральные процессоры, каналы и т. д.);
- 3) линейных символов, отображающих передачу данных между устройствами ввода-вывода и процессорами, а также передачу управления между процессорами;
- 4) специальных символов, используемых для облегчения написания и чтения схемы.

#### *Задание*

Используя приведенный ГОСТ 19.701-90, для своей выпускной квалификационной работы (или темы, разрабатываемой в течение семестра) разработать две любых схемы в соответствии с графической нотацией ГОСТа. Обосновать свой выбор, схемы пояснить, сделав для графических блоков на диаграммах легенду.

### **Лабораторная работа № 5**

#### **Выбор прототипа и разработка архитектуры ПО**

#### *Цель лабораторной работы*

Произвести прототипирование, определить каркас программного продукта и осуществить разработку базовой архитектуры для него.

#### *Информация*

Выбор прототипа и базовой архитектуры программного продукта производится на основе собранных требований и во многом определяется дальнейшие качественные характеристики разрабатываемого продукта, такие как: качество, масштабируемость, расширяемость, лимит пользователей и др.

#### **Выбор архитектуры ПО**

Выбор архитектуры определяет способ реализации требований на высоком уровне абстракции. Именно архитектура почти полностью определяет такие характеристики ПО как надежность, переносимость и удобство сопровождения. Архитектура значительно влияет и на удобство использования и эффективность ПО, которые определяются также и реализацией отдельных компонентов. Значительно меньше влияние архитектуры на

функциональность – обычно для реализации заданной функциональности можно использовать различные архитектуры.

Поэтому выбор между той или иной архитектурой определяется, прежде всего, именно нефункциональными требованиями и необходимыми свойствами ПО в аспектах удобства сопровождения и переносимости. При этом для построения хорошей архитектуры надо учитывать возможные противоречия между требованиями к различным характеристикам и уметь выбирать компромиссные решения, дающие приемлемые значения по всем показателям.

Так, для повышения эффективности в общем случае выгоднее использовать монолитные архитектуры, в которых выделено небольшое число компонентов (в пределе – единственный компонент) – этим обеспечивается экономия как памяти, поскольку каждый компонент обычно имеет свои данные, а здесь число компонентов минимально, так и времени работы, поскольку возможность оптимизировать работу алгоритмов обработки данных имеется также, обычно, только в рамках одного компонента.

С другой стороны, для повышения удобства сопровождения, наоборот, лучше разбивать систему на большое число отдельных компонентов, с тем, чтобы каждый из них решал свою небольшую, но четко определенную часть общей задачи. При этом, если возникают изменения в требованиях или проекте, их обычно можно свести к изменению одной-нескольких таких подзадач, и, соответственно, изменять только отвечающие за решение этих подзадач компоненты.

С третьей стороны, для повышения надежности лучше использовать дублирование функций, т.е. сделать несколько компонентов ответственными за решение одной подзадачи. Причем, поскольку ошибки в ПО чаще всего носят неслучайный характер (т.е. они повторяемы, в отличие от аппаратного обеспечения, где ошибки связаны прежде всего со случайными изменениями характеристик среды и могут быть преодолены простым дублированием компонентов, без изменения их внутренней реализации), лучше использовать достаточно сильно различающиеся способы решения одной и той же задачи в разных компонентах.

### **Разработка и оценка архитектуры**

При проектировании архитектуры системы на основе требований, зафиксированных в виде вариантов использования, возможные шаги состоят в следующем:

1. Выбирается набор «основных» сценариев использования – наиболее существенных и часто используемых
2. Определяются, исходя из опыта проектировщиков, выбранного архитектурного стиля и требований к переносимости и гибкости, компоненты отвечающие за определенные действия – решение определенных подзадач – в рамках этих сценариев
3. Сценарии разбиваются на последовательности обмена сообщениями между полученными компонентами
4. При возникновении дополнительных хорошо выделенных подзадач, добавляются новые компоненты, и сценарии уточняются
5. Для каждого компонента в результате выделяется его интерфейс – набор сообщений, которые он принимает от других компонентов и посылает им
6. Рассматриваются «неосновные» сценарии, которые так же разбиваются на последовательности обмена сообщениями с использованием, по возможности, уже определенных интерфейсов
7. Если интерфейсы недостаточны – они расширяются
8. Если интерфейс компонента слишком велик или компонент отвечает за слишком многое – он разбивается на более мелкие
9. Там, где это необходимо в силу требований эффективности или удобства сопровождения, несколько компонентов могут быть объединены в один
10. Все это делается до тех пор, пока не выполняются следующие условия:

- Все сценарии использования реализуются в виде последовательностей обмена сообщениями между компонентами в рамках их интерфейсов
- Набор компонентов достаточен для обеспечения всей нужной функциональности, достаточно удобен для сопровождения и точки зрения переносимости и не вызывает заметных проблем с эффективностью
- Каждый компонент имеет небольшой, четко определённый круг решаемых задач и четко определенный, сбалансированный по размеру интерфейс.

На основе возможных сценариев использования или *модификации* системы возможен также анализ характеристик архитектуры и оценка ее пригодности или сравнительный анализ нескольких архитектур. Это так называемый *метод анализа архитектуры ПО* (Software Architecture Analysis Method, SAAM). Основные его шаги следующие:

1. Определить набор сценариев действий пользователей или внешних систем, или сценариев использования некоторых возможностей, которые могут уже иметься в системе или быть новыми. Сценарии должны быть значимы для конкретных заинтересованных лиц, будь то пользователь, разработчик, ответственный за сопровождение, представитель контролирующей организации и пр. Чем полнее набор сценариев, тем выше будет качество анализа. Можно оценить частоту появления, важность сценариев.
2. Определить архитектуру (или несколько сравниваемых архитектур). Это должно быть сделано в понятной всем участникам оценки форме.
3. Классифицировать сценарии. Для каждого сценария из набора должно быть определено, поддерживается ли он данной архитектурой или нужно вносить в нее изменения, чтобы этот сценарий стал выполним. Сценарий может поддерживаться, т.е. его выполнение не потребует внесения изменений ни в один из компонентов, или же не поддерживаться, если его выполнение требует изменений в описании поведения одного или нескольких компонентов или изменений в их интерфейсах. Поддержка сценария означает, что лицо, заинтересованное в его выполнении оценивает степень поддержки как достаточную, а необходимые при этом действия как достаточно удобные.
4. Оценить сценарии. Для каждого неподдерживаемого сценария надо определить необходимые изменения в архитектуре – внесение новых компонентов, изменения в существующих, изменения связей и способов взаимодействия. Если есть возможность, стоит оценить трудоемкость внесения таких изменений.
5. Выявить взаимодействие сценариев. Определить какие компоненты требуется изменять для неподдерживаемых сценариев, компоненты, которые требуется изменять для поддержки нескольких сценариев (такие сценарии называют взаимодействующими), оценить степень смысловой связанности взаимодействующих сценариев. Малая связанность по смыслу между взаимодействующими сценариями означает, что компоненты, в которых они взаимодействуют, выполняют слабо связанные между собой задачи и их стоит декомпозировать. Компоненты, в которых взаимодействует много сценариев, также являются возможными проблемными местами.
6. Оценить архитектуру в целом (или сравнить несколько заданных архитектур). Для этого надо использовать оценки важности сценариев и степень их поддержки архитектурой.

#### *Задание 1*

Выполнить прототип для будущего программного продукта и обосновать свой выбор. Определить границы прототипа.

#### *Задание 2*

Исходя из собранных в работе №2 требований, выбрать из базовых архитектур наиболее подходящую, адаптировать её и визуализировать результат для своего проектируемого программного продукта. Результат представить в виде набора UML-диаграмм.

*Контрольные вопросы*

1. Что такое прототип?
2. В чем цель прототипирования?
3. Что такое связность?
4. Что такое сцепление?
5. Как соотношение связности и сцепления влияет на качество программного продукта?
6. Что такое архитектура программного обеспечения?
7. Какие базовые виды архитектур Вы знаете?
8. На что влияет в дальнейшем выбранная архитектура?

## Приложение 2. Оценочные средства для проведения промежуточной аттестации

### а) Планируемые результаты обучения и оценочные средства для проведения промежуточной аттестации:

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
<b>ПК-2. Способность к анализу проблемной ситуации, разработке требований к системе, постановке целей создания системы, разработке концепции системы и технического задания на создание системы, представления концепции, технического задания на систему и изменений в них заинтересованным лицам</b>		
ПК 2.1	Оценивает выбор средств и методов для проведения системного анализа при проектировании программного обеспечения для автоматизированных систем	<p>Жизненным циклом программного обеспечения называется</p> <ol style="list-style-type: none"> <li>1) период времени, за начало которого берется момент принятия решения о необходимости создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации;</li> <li>2) период времени, за начало которого берется момент начала разработки программного продукта и заканчивается в момент закрытия технической поддержки на продукт;</li> <li>3) период времени, за начало которого берется момент принятия решения о необходимости создания программного продукта и заканчивается в момент передачи заказчику в опытно-промышленную эксплуатацию</li> </ol> <p>Спиралевидная модель разработки ПО основана на:</p> <ol style="list-style-type: none"> <li>1) цикле Деминга;</li> <li>2) диаграмме Исикавы;</li> <li>3) модели Хэллмана</li> </ol> <p>Анализ осуществимости требований</p> <ol style="list-style-type: none"> <li>1) позволяет заинтересованным лицам понять риски, связанные с реализацией каждого требования;</li> <li>2) позволяет определить приоритеты требований, чтобы быть уверенным, что команда реализует самую важную и своевременную функциональность;</li> <li>3) содержит определения всех элементов и структур данных, связанных с системой</li> </ol>
<b>ПК-9: Владеет навыками ввода в эксплуатацию аппаратных, программно-аппаратных и программных средств инфокоммуникационной инфраструктуры совместно с представителями поставщиков оборудования, готов к обслуживанию периферийного оборудования и организации инвентаризации технических средств</b>		
ПК 9.1	Оценивает качество ввода в эксплуатацию аппаратных, программно-аппаратных и программных средств инфокоммуникационной инфраструктуры	<p>Взаимоисключающие требования – это</p> <ol style="list-style-type: none"> <li>1) частое явление для крупного продукта, поскольку разные пользователи могут иметь разные мнения по одному и том же вопросу;</li> <li>2) описание поведения системы, когда она взаимодействует с кем-то (или чем-то) из внешней среды;</li> <li>3) варианты, которые используются для обеспечения максимальной гибкости в процессе проектирования функциональности системы</li> </ol>

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		<p>Раздел «Состав и содержание работ по созданию системы» должен содержать перечень стадий и этапов работ по созданию системы в соответствии с ГОСТ...</p> <ol style="list-style-type: none"> <li>1) 39.602-01</li> <li>2) 34.601-90</li> <li>3) 34.601-09</li> </ol> <p>Для лингвистического обеспечения системы приводят требования:</p> <ol style="list-style-type: none"> <li>1) к применению в системе языков программирования высокого уровня;</li> <li>2) к языкам ввода-вывода данных;</li> <li>3) к средствам ведения фактической документации</li> </ol>
ПК 9.2	Оценивает качество обслуживания периферийного оборудования и организацию инвентаризации технических средств	<p>Что определяет выбор архитектуры создаваемого программного продукта?</p> <ol style="list-style-type: none"> <li>1) способ реализации требований на высоком уровне абстракции;</li> <li>2) способ формирования требований на высоком уровне абстракции;</li> <li>3) вариативность функциональных критериев программной системы</li> </ol> <p>Какие из перечисленных характеристик ПО частично определяет выбранное архитектурное решение?</p> <ol style="list-style-type: none"> <li>1) надежность;</li> <li>2) переносимость;</li> <li>3) удобство сопровождения;</li> <li>4) качество UI</li> </ol> <p>Техническое задание - это ...</p> <ol style="list-style-type: none"> <li>1) документ, используемый заказчиком в качестве средства для описания и определения задач, выполняемых при реализации договора;</li> <li>2) документ, содержащий правила, указания или руководства, устанавливающие порядок и способ выполнения или осуществления чего-либо;</li> <li>3) это документ или любой другой источник, информация в котором упорядочена с помощью разбивки на небольшие статьи, отсортированные по названию или тематике.</li> </ol>

**б) Порядок проведения промежуточной аттестации, показатели и критерии оценивания:**

Промежуточная аттестация по дисциплине «Проектирование программных средств» включает теоретические вопросы, позволяющие оценить уровень усвоения обучающимися знаний, и практические задания, выявляющие степень сформированности умений и владений, проводится в форме зачета.

Зачет по дисциплине проводится по результатам отчетности на лабораторных занятиях с опросом в устной форме по этапам выполнения и активного выступления в беседе-обсуждении на лекционных занятиях, выполнения тестовых заданий. Пороговый уровень освоения для каждого элемента – 65% от максимального количества баллов, которые можно получить за освоение этого элемента.