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

13.02.2024 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
2024 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
25.01.2024, протокол № 5

Зав. кафедрой _____ О.С. Логунова

Рабочая программа одобрена методической комиссией ИЭиАС
13.02.2024 г. протокол № 4

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 2025 - 2026 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 2026 - 2027 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 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

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 обслуживание 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 акад. часов, в том числе:

– контактная работа – 55 акад. часов:

– аудиторная – 54 акад. часов;

– внеаудиторная – 1 акад. часов;

– самостоятельная работа – 53 акад. часов;

– в форме практической подготовки – 0 акад. час;

Форма аттестации - зачет

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. 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 лабораторных работ	1. Беседа - обсуждение 2. Устный опрос	ПК-2.1, ПК-9.1, ПК-9.2

2.2 Проектная документация: техническое задание на разработку программного изделия, отраслевые стандарты, способы написания, содержание		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	ПК-2.1, ПК-9.1, ПК-9.2
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 личностно 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 (модуля)

а) Основная литература:

1. Назаров, С. В. Архитектура и проектирование программных систем : монография / С.В. Назаров. — 2-е изд., перераб. и доп. — Москва : ИНФРА-М, 2023. — 374 с. — (Научная мысль). — DOI 10.12737/18292. - ISBN 978-5-16-011753-9. - Текст : электронный. - URL: https://znanium.com/catalog/product/1895672 (дата обращения: 21.04.2024). – Режим доступа: по подписке.

2. Введение в программную инженерию : учебник / В. А. Антипов, А. А. Бубнов, А. Н. Пылькин, В. К. Столчнев. — Москва : КУРС : ИНФРА-М, 2019. — 336 с. - ISBN 978-5-906923-22-6. - Текст : электронный. - URL: https://znanium.com/catalog/product/1035160 (дата обращения: 21.04.2024)

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: https://elibrary.ru/project_risc.asp

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

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 C- 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 отобрать 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 требование 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 обоих 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 одном объекту или однородная по своей сути объединяется в одном приложении.

Приложения именуются 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 различи между заказчиком системы и её пользователем?

Лабораторная работа №2

Разработка технического задания на автоматизированную систему
согласно ГОСТ 34.602-2020. Часть 1

Цель: научиться разрабатывать техническое задание для программных продуктов согласно действующим стандартам.

Информация

ГОСТ 34.602-2020 является основным рабочим стандартом для создания технического задания (ТЗ) на разработку автоматизированных систем (АС) на сегодняшний день. ТЗ на АС является основным документом, определяющим требования и порядок создания (развития или модернизации - далее создания) автоматизированной системы, в соответствии с которым проводится разработка АС и ее приемка при вводе в действие.

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

Логически ТЗ можно разбить на несколько частей: общую, специальную и организационную. В настоящей лабораторной работе мы будем учиться писать общую часть ТЗ.

К общей части относятся следующие разделы:

- «Общие сведения»
- «Назначение и цели создания системы»
- «Характеристики объекта автоматизации»

Рассмотрим их содержание подробнее. В разделе «Общие сведения» указывают:

1) полное наименование системы и ее условное обозначение;

2) шифр темы или шифр (номер) договора;

3) наименование предприятий (объединений) разработчика и заказчика (пользователя) системы и их реквизиты;

4) перечень документов, на основании которых создается система, кем и когда утверждены эти документы;

5) плановые сроки начала и окончания работы по созданию системы;

6) сведения об источниках и порядке финансирования работ;

7) порядок оформления и предъявления заказчику результатов работ по созданию системы (ее частей), по изготовлению и наладке отдельных средств (технических, программных, информационных) и программно-технических (программно- методических) комплексов системы.

Раздел «Назначение и цели создания (развития) системы» состоит из подразделов:

1) назначение системы;

2) цели создания системы.

В подразделе «Назначение системы» указывают вид автоматизируемой деятельности (управление, проектирование и т. п.) и перечень объектов автоматизации (объектов), на которых предполагается ее использовать.

Для АСУ дополнительно указывают перечень автоматизируемых органов (пунктов) управления и управляемых объектов.
В подразделе «Цели создания системы» 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 системы.
В разделе «Характеристики объекта автоматизации» приводят:
1) краткие сведения об объекте автоматизации или ссылки на документы, содержащие такую информацию;
2) сведения об условиях эксплуатации объекта автоматизация и характеристиках окружающей среды.

Примечание: Для САПР в разделе дополнительно приводят основные параметры и характеристики объектов проектирования.

Задание

разработать первую часть технического задания на создание автоматизированной системы. В качестве системы берётся та, для которой были собраны требования в предыдущей лабораторной работе.

Контрольные вопросы

1. На основании каких документов может создаваться автоматизированная система?
2. Что такое объект автоматизации?
3. Как расшифровывается аббревиатура «САПР»?
4. Как расшифровывается аббревиатура «АСУ»?
5. Может ли раздел «Описание объекта автоматизации» включать схемы, рисунки, графики?

Лабораторная работа №3

Разработка технического задания на автоматизированную систему
согласно ГОСТ 34.602-2020. Часть 2

Цель: научиться разрабатывать техническое задание для программных продуктов согласно действующим стандартам.

Информация

ГОСТ 34.602-2020 является основным рабочим стандартом для создания технического задания (ТЗ) на разработку автоматизированных систем (АС) на сегодняшний день. ТЗ на АС является основным документом, определяющим требования и порядок создания (развития или модернизации - далее создания) автоматизированной системы, в соответствии с которым проводится разработка АС и ее приемка при вводе в действие.

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

Логически ТЗ можно разбить на несколько частей: общую, специальную и организационную. В настоящей лабораторной работе мы будем учиться писать специальную часть ТЗ.

К специальной части относится один большой раздел, которому в ГОСТ 34.602-2020 соответствует пп. 2.6 под названием «Требования к системе». В идеале, в содержимое этого раздела должны переходить требования из спецификации требований (Лабораторная работа №1). Он служит для окончательного их упорядочивания и структурирования. Рассмотрим содержание этого раздела подробнее.

Раздел "Требования к системе" состоит из следующих подразделов:

1) требования к системе в целом;
2) требования к функциям (задачам), выполняемым системой;
3) требования к видам обеспечения.

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

[2.6.1.] В подразделе ***"Требования к системе в целом"*** указывают:

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

[2.6.1.1.] В **требованиях к структуре и функционированию** системы приводят:

1) перечень подсистем, их назначение и основные характеристики, требования к числу уровней иерархии и степени централизации системы;
2) требования к способам и средствам связи для информационного обмена между компонентами системы;
3) требования к характеристикам взаимосвязей создаваемой системы со смежными системами, требования к ее совместимости, в том числе указания о способах обмена информацией (автоматически, пересылкой документов, по телефону и т.п.);
4) требования к режимам функционирования системы;
5) требования по диагностированию системы;
6) перспективы развития, модернизации системы.

[2.6.1.2.] В **требованиях к численности и квалификации персонала** АС приводят:

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

[2.6.1.3.] В **требованиях к показателям назначения** АС приводят значения параметров, характеризующие степень соответствия системы ее назначению.

Для АСУ указывают:

- степень приспособляемости системы к изменению процессов и методов управления, к отклонениям параметров объекта управления;
- допустимые пределы модернизации и развития системы;
- вероятностно-временные характеристики, при которых сохраняется целевое назначение

системы.
[2.6.1.4.] В **требования к надежности** включают:
1) состав и количественные значения показателей надежности для системы в целом или ее подсистем;
2) перечень аварийных ситуаций, по которым должны быть регламентированы требования к надежности, и значения соответствующих показателей;
3) требования к надежности технических средств и программного обеспечения;
4) требования к методам оценки и контроля показателей надежности на разных стадиях создания системы в соответствии с действующими нормативно-техническими документами.
[2.6.1.5.] В **требования по безопасности** включают требования по обеспечению безопасности при монтаже, наладке, эксплуатации, обслуживании и ремонте технических средств системы (защита от воздействий электрического тока, электромагнитных полей, акустических шумов и т.п.), по допустимым уровням освещенности, вибрационных и шумовых нагрузок.
[2.6.1.6.] В **требования по эргономике и технической эстетике** включают показатели АС, задающие необходимое качество взаимодействия человека с машиной и комфортность условий работы персонала.
[2.6.1.7.] Для подвижных АС (например, АС для передвижных лабораторий, узлов связи и т.д.) в **требования к транспортабельности** включают конструктивные требования, обеспечивающие транспортабельность технических средств системы, а также требования к транспортным средствам.
[2.6.1.8.] В **требования к эксплуатации, техническому обслуживанию, ремонту и хранению** включают:
1) условия и регламент (режим) эксплуатации, которые должны обеспечивать использование технических средств (ТС) системы с заданными техническими показателями, в том числе виды и периодичность обслуживания ТС системы или допустимость работы без обслуживания;
2) предварительные требования к допустимым площадям для размещения персонала и ТС системы, к параметрам сетей энергоснабжения и т.п.;
3) требования по количеству, квалификации обслуживающего персонала и режимам его работы;
4) требования к составу, размещению и условиям хранения комплекта запасных изделий и приборов;
5) требования к регламенту обслуживания.
[2.6.1.9.] В **требования к защите информации от несанкционированного доступа** включают требования, установленные в НТД, действующей в отрасли (ведомстве) заказчика.
[2.6.1.10.] 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 системе.
[2.6.1.11.] В **требованиях к средствам защиты от внешних воздействий** приводят:
1) требования к радиоэлектронной защите средств АС;
2) требования по стойкости, устойчивости и прочности к внешним воздействиям (среде применения).
[2.6.1.12.] В **требованиях по патентной чистоте** указывают перечень стран, в отношении

которых должна быть обеспечена патентная чистота системы и ее частей.
[2.6.1.13.] В **требования к стандартизации и унификации** включают:
показатели, устанавливающие требуемую степень использования стандартных, унифицированных методов реализации функций (задач) системы, поставляемых программных средств, типовых математических методов и моделей, типовых проектных решений, унифицированных форм управленческих документов, установленных ПР 50.1.019-2000, общесоюзных классификаторов технико-экономической информации и классификаторов других категорий в соответствии с областью их применения, требования к использованию типовых автоматизированных рабочих мест, компонентов и комплексов.
[2.6.1.14.] В **дополнительные требования** включают:
1) требования к оснащению системы устройствами для обучения персонала (тренажерами, другими устройствами аналогичного назначения) и документацией на них;
2) требования к сервисной аппаратуре, стендам для проверки элементов системы;
3) требования к системе, связанные с особыми условиями эксплуатации;
4) специальные требования по усмотрению разработчика или заказчика системы.
[2.6.2.] В подразделе ***"Требования к функциям (задачам)"***, выполняемым системой, приводят:
1) по каждой подсистеме перечень функций, задач или их комплексов (в том числе обеспечивающих взаимодействие частей системы), подлежащих автоматизации; при создании системы в две или более очереди - перечень функциональных подсистем, отдельных функций или задач, вводимых в действие в 1-й и последующих очередях;
2) временной регламент реализации каждой функции, задачи (или комплекса задач);
3) требования к качеству реализации каждой функции (задачи или комплекса задач), к форме представления выходной информации, характеристики необходимой точности и времени выполнения, требования одновременности выполнения группы функций, достоверности выдачи результатов;
4) перечень и критерии отказов для каждой функции, по которой задаются требования по надежности.
[2.6.3.] В подразделе ***"Требования к видам обеспечения"*** в зависимости от вида системы приводят требования к математическому, информационному, лингвистическому, программному, техническому, метрологическому, организационному, методическому и другим видам обеспечения системы.
[2.6.3.1.] Для **математического обеспечения** системы приводят требования к составу, области применения (ограничения) и способам использования в системе математических методов и моделей, типовых алгоритмов и алгоритмов, подлежащих разработке.
[2.6.3.2.] Для **информационного обеспечения** системы приводят требования:
1) к составу, структуре и способам организации данных в системе;
2) к информационному обмену между компонентами системы;
3) к информационной совместимости со смежными системами;
4) по использованию общесоюзных и зарегистрированных республиканских, отраслевых классификаторов, унифицированных документов и классификаторов, действующих на данном предприятии;
5) по применению систем управления базами данных;
6) к структуре процесса сбора, обработки, передачи данных в системе и представлению данных;

7) к защите данных от разрушений при авариях и сбоях в электропитании системы;
8) к контролю, хранению, обновлению и восстановлению данных;
9) к процедуре придания юридической силы документам, продуцируемым техническими средствами АС (в соответствии с ГОСТ 6.10.4).
[2.6.3.3.] Для **лингвистического обеспечения** системы приводят требования к применению в системе языков программирования высокого уровня, языков взаимодействия пользователей и технических средств системы, а также требования к кодированию и декодированию данных, к языкам ввода-вывода данных, языкам манипулирования данными, средствам описания предметной области (объекта автоматизации), к способам организации диалога.
[2.6.3.4.] Для **программного обеспечения** системы приводят перечень покупных программных средств, а также требования:
1) к независимости программных средств от используемых СВТ и операционной среды;
2) к качеству программных средств, а также к способам его обеспечения и контроля;
3) по необходимости согласования вновь разрабатываемых программных средств с фондом алгоритмов и программ.
[2.6.3.5.] Для **технического обеспечения** системы приводят требования:
1) к видам технических средств, в том числе к видам комплексов технических средств, программно-технических комплексов и других комплектующих изделий, допустимых к использованию в системе;
2) к функциональным, конструктивным и эксплуатационным характеристикам средств технического обеспечения системы.
[2.6.3.6.] В требованиях к **метрологическому обеспечению** приводят:
1) предварительный перечень измерительных каналов;
2) требования к точности измерений параметров и (или) к метрологическим характеристикам измерительных каналов;
3) требования к метрологической совместимости технических средств системы;
4) перечень управляющих и вычислительных каналов системы, для которых необходимо оценивать точностные характеристики;
5) требования к метрологическому обеспечению технических и программных средств, входящих в состав измерительных каналов системы, средств встроенного контроля, метрологической пригодности измерительных каналов и средств измерений, используемых при наладке и испытаниях системы;
6) вид метрологической аттестации (государственная или ведомственная) с указанием порядка ее выполнения и организаций, проводящих аттестацию.
[2.6.3.7.] Для **организационного обеспечения** приводят требования:
1) к структуре и функциям подразделений, участвующих в функционировании системы или обеспечивающих эксплуатацию;
2) к организации функционирования системы и порядку взаимодействия персонала АС и персонала объекта автоматизации;
3) к защите от ошибочных действий персонала системы.
[2.6.3.8.] Для **методического обеспечения САПР** приводят требования к составу нормативно-технической документации системы (перечень применяемых при ее функционировании стандартов, нормативов, методик и т.п.).

Задание

Разработать вторую часть технического задания на создание автоматизированной системы, используя требования ГОСТ 34.602-2020. В качестве системы берётся та, для которой были собраны требования и написана первая часть ТЗ в предыдущих двух лабораторных работах.

Контрольные вопросы

1. Что такое лингвистическое обеспечение системы?
2. Что такое методическое обеспечение системы?
3. Что такое техническое обеспечение системы?
4. Что такое организационное обеспечение системы?
5. Что такое информационное обеспечение системы?

Лабораторная работа №4

Разработка технического задания на автоматизированную систему
согласно ГОСТ 34.602-2020. Часть 3

Цель: научиться разрабатывать техническое задание для программных продуктов согласно действующим стандартам.

Информация

ГОСТ 34.602-2020 является основным рабочим стандартом для создания технического задания (ТЗ) на разработку автоматизированных систем (АС) на сегодняшний день. ТЗ на АС является основным документом, определяющим требования и порядок создания (развития или модернизации - далее создания) автоматизированной системы, в соответствии с которым проводится разработка АС и ее приемка при вводе в действие.

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

Логически ТЗ можно разбить на несколько частей: общую, специальную и организационную. В настоящей лабораторной работе мы будем учиться писать последнюю, организационную часть ТЗ.

К организационной части относится информация, представленная в разделах:

- Состав и содержание работ по созданию системы;
- Порядок контроля и приёмки системы;
- Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие;
- Требования к документированию;
- Источники разработки.

Рассмотрим содержание этих разделов подробнее.

Раздел “Состав и содержание работ по созданию (развитию) системы” должен содержать перечень стадий и этапов работ по созданию системы в соответствии с ГОСТ 24.601, сроки их выполнения, перечень организаций — исполнителей работ, ссылки на документы, подтверждающие согласие этих организаций на участие в создании системы, или запись, определяющую ответственного (заказчик или разработчик) за проведение этих работ.

В данном разделе также приводят:

1) перечень документов, по ГОСТ 34.201, предъявляемых по окончании соответствующих стадий и этапов работ;

2) вид и порядок проведения экспертиза технической документации (стадия, этап, объем проверяемой 'документации, организация-эксперт);

3) программу работ, направленных на обеспечение требуемого уровня надежности

разрабатываемой системы (при необходимости);

4) перечень работ по метрологическому обеспечению на всех стадиях создания системы с указанием их сроков выполнения и организаций-исполнителей (при необходимости).

В разделе “Порядок контроля и приемки системы” указывают:

1) виды, состав, объем и методы испытаний системы и ее составных частей (виды испытаний в соответствии с действующими нормами, распространяющимися на разрабатываемую систему);

2) общие требования к приемке работ по стадиям (перечень участвующих предприятий и организаций, место и сроки проведения), порядок согласования и утверждения приемочной документации;

ЗУ статус приемочной комиссии (государственная, [межведомственная, ведомственная).

В разделе “Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие” необходимо привести перечень основных мероприятий и их исполнителей, которые следует выполнить при подготовке объекта автоматизации к вводу АС в действие.

В перечень основных мероприятий включают:

1) приведение поступающей в систему информации (в соответствии с требованиями к информационному и лингвистическому обеспечению) к виду, пригодному для обработки с помощью ЭВМ;

2) изменения, которые необходимо осуществить в объекте автоматизации;

3) создание условий функционирования объекта автоматизации, при которых гарантируется соответствие создаваемой системы требованиям, содержащимся в ТЗ;

4) создание необходимых для функционирования системы подразделений и служб;

5) сроки и порядок комплектования штатов и обучения персонала.

В разделе “Требования к документированию” приводят:

1) согласованный разработчиком и Заказчиком системы перечень подлежащих разработке комплектов и видов документов, соответствующих требованиям ГОСТ 34.201 и НТД отрасли заказчика; перечень документов, выпускаемых на машинных носителях; требования к микрофильмированию документации;

2) требования по документированию комплектующих элементов межотраслевого применения в соответствии с требованиями ЕСКД и ЕСПД;

3) при отсутствии государственных стандартов, определяющих требования к документированию элементов системы, дополнительно включают требования к составу и содержанию таких документов.

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 н др.), на основании которых разрабатывалось ТЗ и которые должны быть использованы при создании системы.

Задание

Разработать третью часть технического задания на создание автоматизированной системы, используя требования ГОСТ 34.602-2020. В качестве системы берётся та, для которой были собраны требования и написаны первая и вторая части ТЗ в предыдущих трёх лабораторных работах.

Контрольные вопросы

1. Какие разделы можно условно определить к специальной части технического задания?
2. Что регламентирует ГОСТ 34.201?
3. Что регламентирует ГОСТ 24.601?

4. Что включается в перечень стадий и этапов работ по созданию системы?
5. Приведите примеры документов, которые можно отнести к НТД?

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 отличии 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 выполнятся 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 сточки 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 выполнение 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 ***(пороговый уровень)***
Выполнить прототип для будущего программного продукта и обосновать свой выбор. Определить границы прототипа.
Задание 2 ***(средний уровень)***
Исходя из собранных в работе №1 требований, выбрать из базовых архитектур наиболее подходящую, адаптировать её и визуализировать результат для своего проектируемого программного продукта. Результат представить в виде набора UML-диаграмм.
Задание 3 ***(высокий уровень)***
Представить этот проект в виде развернутой ArchiMate схемы.

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
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	Жизненным циклом программного обеспечения называется 1) период времени, за начало которого берется момент принятия решения о необходимости создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации; 2) период времени, за начало которого берется момент начала разработки программного продукта и заканчивается в момент закрытия технической поддержки на продукт; 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
		Спиралевидная модель разработки ПО основана на: 1) цикле Деминга; 2) диаграмме Исикавы; 3) модели Хэллмана
		Анализ осуществимости требований 1) позволяет заинтересованным лицам понять риски, связанные с реализацией каждого требования; 2) позволяет определить приоритеты требований, чтобы быть уверенным, что команда реализует самую важную и своевременную функциональность; 3) содержит определения всех элементов и структур данных, связанных с системой
ПК-9: Владеет навыками ввода в эксплуатацию аппаратных, программно-аппаратных и программных средств инфокоммуникационной инфраструктуры совместно с представителями поставщиков оборудования, готов к обслуживание периферийного оборудования и организации инвентаризации технических средств		
ПК 9.1	Оценивает качество ввода в эксплуатацию аппаратных, программно-аппаратных и программных средств инфокоммуникационной инфраструктуры	Взаимоисключающие требования – это 1) частое явление для крупного продукта, поскольку разные пользователи могут иметь разные мнения по одному и том же вопросу; 2) описание поведения системы, когда она взаимодействует с кем-то (или чем-то) из внешней среды;

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		3) варианты, которые используются для обеспечения максимальной гибкости в процессе проектирования функциональности системы
		Раздел «Состав и содержание работ по созданию системы» должен содержать перечень стадий и этапов работ по созданию системы в соответствии с ГОСТ… 1) 39.602-01 2) 34.601-90 3) 34.601-09
		Для лингвистического обеспечения системы приводят требования: 1) к применению в системе языков программирования высокого уровня; 2) к языкам ввода-вывода данных; 3) к средствам ведения фактической документации
ПК 9.2	Оценивает качество обслуживания периферийного оборудования и организацию инвентаризации технических средств	Что определяет выбор архитектуры создаваемого программного продукта? 1) способ реализации требований на высоком уровне абстракции; 2) способ формирования требований на высоком уровне абстракции; 3) вариативность функциональных критериев программной системы
		Какие из перечисленных характеристик ПО частично определяет выбранное архитектурное решение? 1) надежность; 2) переносимость; 3) удобство сопровождения; 4) качество UI
		Техническое задание - это ... 1) документ, используемый заказчиком в качестве средства для описания и определения задач, выполняемых при реализации договора; 2) документ, содержащий правила, указания или руководства, устанавливающие порядок и способ выполнения или осуществления чего-либо; 3) это документ или любой другой источник, информация в котором упорядочена с помощью разбивки на небольшие статьи, отсортированные по названию или тематике.

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.