

Государственное образовательное учреждение
«Приднестровский государственный университет им. Т.Г. Шевченко»

Инженерно-технический институт

Кафедра «Программное обеспечение вычислительной техники
и автоматизированных систем»

УТВЕРЖДАЮ
Директор института, доцент

 Ф.Ю. Бурменко
«14» 09 2018 г.



РАБОЧАЯ ПРОГРАММА

на 2018/2019 учебный год

учебной дисциплины

**Б1.В.ОД.7 «КОНСТРУИРОВАНИЕ ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ»**

Направление подготовки:
09.03.04 Программная инженерия

Профиль подготовки
Разработка программно-информационных систем

Для набора
2016 года

Квалификация (степень) выпускника
бакалавр

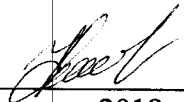
Форма обучения:
очная

Тирасполь, 2018

Рабочая программа дисциплины «Конструирование программного обеспечения» /сост. О.С. Белоконь – Тирасполь: ГОУ ПГУ, 2018. – 17 с.

Рабочая программа предназначена для преподавания обязательной дисциплины вариативной части дисциплин (модулей) студентам очной формы обучения по направлению подготовки 09.03.04 «ПРОГРАММНАЯ ИНЖЕНЕРИЯ»

Рабочая программа составлена с учетом Федерального Государственного образовательного стандарта высшего образования по направлению подготовки 09.03.04 «ПРОГРАММНАЯ ИНЖЕНЕРИЯ», утвержденного приказом Министерства образования и науки Российской Федерации от 12 марта 2015 г. № 229.

Составитель  /О.С. Белоконь, ст. преподаватель кафедры ПОВТиАС
«27» августа 2018 г.

1. Цели и задачи освоения дисциплины

Цель дисциплины: формирование у студентов знаний по методам, инструментам и процессам конструирования надежного, устойчивого и эффективного ПО для средств вычислительной техники автоматизированных и автоматических систем в рамках современных технологий разработки.

Задачи дисциплины:

- раскрыть основные вопросы теории конструирования программного обеспечения;
- сформировать компетентность в области конструирования программного обеспечения;
- обучить студентов применять методы, модели и алгоритмы решения типовых задач, связанных с конструированием программного обеспечения;
- ознакомить с современными тенденциями в области конструирования программного обеспечения.

2. Место дисциплины в структуре ООП ВО

Б1.В.ОД.7 Дисциплины (модули). Вариативная часть. Обязательные дисциплины.

Общая трудоемкость дисциплины составляет 4 зачетные единицы, 144 часа.

Изучение дисциплины требует базовой подготовки в объеме рабочих программ дисциплин «Информатика», «Прикладное программирование», «Объектно-ориентированное программирование», «Введение в программную инженерию», «Разработка и анализ требований».

Знания, умения и навыки, приобретенные при изучении данной дисциплины, используются студентами в дисциплине «Тестирование программного обеспечения», «Проектирование и архитектура программных систем», «Управление программными проектами».

Для успешного освоения дисциплины необходимо:

- умение применять множество различных шаблонов конструирования, сред разработки и конструирования разнообразного программного обеспечения;
- умение конструировать и реализовывать программное обеспечение,
- использовать адекватные метрики качества как средство оценки конструирования ПО;
- оценивать соответствие результатов конструирования ПО поставленным целям.

3. Требования к результатам освоения дисциплины

Изучение дисциплины направлено на формирование следующих компетенций: ОПК-3, ПК-4, ПК-5, ПК-7, ПК-9, ПК-16. Расшифровка компетенций дана в следующей таблице.

Таблица 1 – Формулировка компетенции
для направления 09.03.04 «ПРОГРАММНАЯ ИНЖЕНЕРИЯ»

Код компетенции	Формулировка компетенций
ОПК-3	готовностью применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов;
ПК-4	владением концепциями и атрибутами качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роли людей, процессов, методов, инструментов и технологий обеспечения качества;

ПК-5	владением стандартами и моделями жизненного цикла;
ПК-7	владением методами управления процессами разработки требований, оценки рисков, приобретения, проектирования, конструирования, тестирования, эволюции и сопровождения;
ПК-9	владением методами контроля проекта и готовностью осуществлять контроль версий;
ПК-16	способностью формализовать предметную область программного проекта и разработать спецификации для компонентов программного продукта;

В результате освоения дисциплины студент должен:

3.1 Знать:

- формальные методы, технологии и инструменты разработки программного продукта;
- жизненный цикл программного обеспечения и технологических процессов разработки ПО;
- организация жизненного цикла ПО;
- каскадные и итеративные модели жизненного цикла;
- набор стандартов, регулирующих процессы разработки ПО в целом;
- методология быстрой разработки приложений.

3.2 Уметь:

- конструировать программное обеспечение;
- разрабатывать основные программные документы;
- формулировать и решать задачи проектирования профессионально-ориентированных программных систем с использованием различных методов и решений;
- проводить выбор интерфейсных средств при построении сложных профессионально-ориентированных информационных систем.

3.3 Владеть:

- методами конструирования программного обеспечения;
- современными методами, средствами и технологиями разработки программного обеспечения.

Рабочая программа учебной дисциплины рассчитана на 62 часа аудиторных занятий, в том числе 26 часов отводится на лекционные занятия, 18 часов – на лабораторные занятия, 18 часов – на практические занятия.

С целью систематизации и закрепления полученных теоретических знаний и практических умений в рабочей программе учебной дисциплины предусмотрена самостоятельная работа студентов. Внеаудиторная самостоятельная работа студентов – 46 часов. На экзамен – 36 часов.

Для проверки знаний студентов в рабочей программе указаны по окончании изучения каких разделов следует проводить рубежный контроль. Учебная дисциплина изучается 1 семестр и заканчивается промежуточной аттестацией студентов в форме экзамена.

4. Структура и содержание дисциплины

4.1. Распределение трудоемкости в з.е./часах по видам аудиторной и самостоятельной работы студента по семестрам

Семестр	Трудоемкость з.е./часы	Количество часов						Форма итог. контроля
		В том числе				Сам. работа	Экзамен	
		Аудиторных						
		Всего	Лекций	Практических занятия	Лабораторных работ			
6	4/144	62	26	18	18	46	36	Экзамен,
Итого	4/144	62	26	18	18	46	36	Экзамен,

4.2. Распределение видов учебной работы и их трудоемкости по разделам дисциплины

№ раз- дела	Наименование раздела	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа (СР)
			Л	ПЗ	ЛР	
1	Введение. Интеллектуализация информационных систем	6	2	-	-	4
2	Жизненный цикл программного обеспечения и технологических процессов. Стратегии конструирования ПО	18	6	4	-	8
3	Проектирование программных средств на основе концепции и стандартов открытых систем	16	4	4	-	8
4	Технологический цикл конструирования программной системы	32	4	2	18	8
5	Качество ПО. Основные понятия и показатели надежности программных средств	18	6	2	-	10
6	Сертификация программного обеспечения.	18	4	6	-	8
	Итого	108	26	18	18	46
	Подготовка к экзамену	36				
	Всего	4/144				

4.3. Тематический план по видам учебной деятельности

Лекции

№ п/п	Номер раздела дисциплины	Объем часов	Тема лекции	Учебно- наглядные пособия
1	1	2	Введение. Интеллектуализация информационных систем. Сложность программного обеспечения (ПО). Пути ограничения сложности ПО	Слайды презентации
2	2	2	Жизненный цикл ПО и технологических процессов. Классический жизненный цикл ПО	Слайды презентации

3	2	2	Стратегии конструирования ПО. Модели инкрементной и эволюционной стратегии жизненного цикла ПО	Слайды презентации
4	2	2	Процесс производства ПО. Методология разработки сложных программных систем. Унифицированный процесс разработки. Экстремальное программирование	Слайды презентации
5	3	2	Проектирование программных средств на основе концепции и стандартов открытых систем	Слайды презентации
6	3	2	Основные критерии проектирования программных систем Технологический цикл конструирования программной системы	Слайды презентации
7	4	2	Нисходящие процессы разработки ПО	Слайды презентации
8	4	2	Восходящие процессы разработки ПО	Слайды презентации
9	5	2	Качество ПО. Основные аспекты качества ПО по ISO 9126	Слайды презентации
10	5	2	Характеристики и атрибуты качества ПО по ISO 9126.	Слайды презентации
11	5	2	Основные понятия и показатели надежности программных средств	Слайды презентации
12	6	2	Методы контроля качества Сертификация ПО.	Слайды презентации
13	6	2	Стандарты производства. Основные понятия и показатели надежности программных средств	Слайды презентации
	Итого:	26		

Практические (семинарные) занятия

№ п/п	Номер раздела дисциплины	Объем часов	Тема практического занятия	Учебно-наглядные пособия
1	2	2	История конструирования операционных систем.	Слайды презентации
2	2	2	Распределенная система управления образовательной деятельностью университета	Слайды презентации
3	3	2	Конструирование систем управления базами данных, распределенные системы. ERP-системы.	Слайды презентации
4	3	2	Информационные технологии управления ресурсами предприятия	Слайды презентации
5	4	2	Примеры технологий создания программного обеспечения различных компаний-поставщиков. IBM (RUP). Oracle (OracleMethod). Borland (ALM, CaliberRM), Microsoft (MSF)	Слайды презентации
6	5	2	Метрики программного обеспечения для оценки качества. Метрики объектно-ориентированных программных средств	Слайды презентации
7	6	2	Внешние стандарты. Стандарты консорциума OMG – CORBA, UML, MDA. Стандарты международных организаций по стандартизации ISO/IEC, IEEE, TMF.	Слайды презентации

8	6	2	Стандарты производителей платформ, операционных сред (<i>Microsoft, Sun Microsystems, CISCO, NOKIA</i>) <i>Win32 API, .NET Framework SDK, Software Development Kit.</i>	Слайды презентации
9	6	2	Стандарты производителей инструментов, систем управления базами данных (<i>Borland, IBM, Microsoft, Sun, Oracle</i>) – <i>UML</i> . Внутренние стандарты.	Слайды презентации
Итого:		18		

Лабораторные работы

№ п/п	Номер раздела дисциплины	Объем часов	Тема лабораторного занятия	Учебно-наглядные пособия
1	4	2	Теоретическое введение в предметную область	электр. вариант ЛБ
2	4	2	Методология функционального моделирования <i>IDEF0</i>	электр. вариант ЛБ
3	4	2	Методология функционального моделирования <i>IDEF0</i>	электр. вариант ЛБ
4	4	2	Методология функционального моделирования <i>IDEF0</i>	электр. вариант ЛБ
5	4	2	Дополнение моделей процессов диаграммами DFD и <i>WORKFLOW (IDEF3)</i>	электр. вариант ЛБ
6	4	2	Дополнение моделей процессов диаграммами DFD и <i>WORKFLOW (IDEF3)</i>	электр. вариант ЛБ
7	4	2	Дополнение моделей процессов диаграммами DFD и <i>WORKFLOW (IDEF3)</i>	электр. вариант ЛБ
8	4	2	Отчеты в <i>BPwin</i>	электр. вариант ЛБ
9	4	2	Отчеты в <i>BPwin</i>	электр. вариант ЛБ
Итого:		18		

Самостоятельная работа студента

Раздел дисциплины	№ п/п	Тема и вид СРС	Трудоёмкость (в часах)
Раздел 1	1	Тема: Сложность программного обеспечения. Пути ограничения сложности программного обеспечения СРС №1: изучение материалов дополнительной литературы, поиск и анализ материала в электронных источниках информации (краткий конспект).	4
Раздел 2	2	Тема: Жизненный цикл программного обеспечения и технологических процессов. СРС №2: изучение материалов дополнительной литературы, поиск и анализ материала в электронных источниках информации (разработка схемы).	4
	3	Тема: Стратегии конструирования программного обеспечения СРС №3: изучение материалов дополнительной литературы, поиск и анализ материала в электронных источниках информации (сравнительная характеристика).	4

Раздел 3	4	Тема: Проектирование программных средств на основе концепции и стандартов открытых систем СРС №4: изучение материалов дополнительной литературы, поиск и анализ материала в электронных источниках информации (подготовка реферата).	4
	5	Тема: Основные критерии проектирования программных систем СРС №5: изучение материалов дополнительной литературы, поиск и анализ материала в электронных источниках информации (формирование таблицы критериев).	4
Раздел 4	6	Тема: Технологический цикл конструирования программной системы СРС №6: изучение материалов дополнительной литературы, поиск и анализ материала в электронных источниках информации (разработка диаграммы).	4
	7	Тема: Нисходящие и восходящие процессы разработки программного обеспечения СРС №7: изучение материалов дополнительной литературы, поиск и анализ материала в электронных источниках информации (подготовка доклада).	4
Раздел 5	8	Тема: Качество программного обеспечения. Основные аспекты качества программного обеспечения по ISO 9126. Показатели надежности программных средств СРС №8: изучение материалов дополнительной литературы, поиск и анализ материала в электронных источниках информации (создание презентации).	10
Раздел 6	9	Тема: Сертификация, валидация, верификация программного обеспечения СРС №9: изучение материалов дополнительной литературы, поиск и анализ материала в электронных источниках информации (подготовка реферата).	8
Итого:			46

5. Примерная тематика курсовых проектов (работ) – учебным планом не предусмотрено

6. Образовательные технологии

Семестр	Вид занятия (Л, ПР, ЛР)	Используемые интерактивные образовательные технологии	Количество часов
6	Л	Компьютерные технологии обучения, проблемная лекция, лекция-диспут, информационно-развивающие технологии, <i>case-study</i> (ситуационный анализ)	26
	ЛР	Компьютерные технологии обучения, Технологии учебного проектирования и моделирования; Ситуационный анализ	18
	ПЗ	компьютерные технологии обучения, информационно-развивающие технологии, <i>case-study</i> (ситуационный анализ)	18
Итого:			62

7. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов

Итоговой формой контроля знаний, умений и навыков по дисциплине является экзамен.

Оценка знаний студентов производится по следующим критериям:

- оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал курса, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами и вопросами, причем не затрудняется с ответами при видоизменении заданий, правильно обосновывает принятые решения, владеет разносторонними навыками и приемами выполнения практических задач;

- оценка «хорошо» выставляется студенту, если он твердо знает материал курса, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения;

- оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических задач;

- оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями решает практические задачи или не справляется с ними самостоятельно.

Предполагается выполнение с последующей защитой лабораторных работ, модульный контроль для оценки уровня усвоения материала студентами

Примерный вариант модульного контроля № 1:

Задание: Отметьте верные утверждения.

1. Жизненный цикл ПО – это

- a) ☐ период времени от установления требований к ней до прекращения ее использования.
- b) ☐ период времени, включающий кодирование, тестирование и эксплуатацию программ

2. Модель, в которой впервые были выделены шаги разработки, четко определенные и следующие в точном порядке

- a) ☐ спиральная
- b) ☐ водопадная (каскадная)
- c) ☐ инкрементная

3. Для разработки небольших проектов или при разработке типовых систем обычно используется следующая модель

- a) ☐ спиральная
- b) ☐ водопадная (каскадная)
- c) ☐ инкрементная

4. Внутренняя структура продукта (компоненты и их связи), основы пользовательского интерфейса продукта, представление извне

- a) ☐ структура ПО
- b) ☐ архитектура ПО

5. Начальный этап жизненного цикла программы - это:

- a) ☐ определение стоимости
- b) ☐ планирование
- c) ☐ оценка осуществимости

6. Спиральная модель ЖЦПО

- a) ☐ серия последовательных итераций при разработке программы
- b) ☐ основывается на разработке работающего прототипа системы
- c) ☐ допускает корректировки по ходу разработки

7. Что включает в себя архитектура ПО

- a) ☐ внутреннюю структуру продукта
- b) ☐ инструменты разработки и управления проектом
- c) ☐ основы пользовательского интерфейса продукт

8. Какие утверждения верны для водопадной модели

- a) ☐ в рамках водопадной модели было введено прототипирование
- b) ☐ не ограничена возможность возвратов на произвольный шаг назад
- c) ☐ допускается возврат только на предыдущий шаг

9. Выберите верные утверждения:

- a) ☐ постановка задачи - один из этапов ЖЦПО
- b) ☐ необходимо документировать ход выполнения работы
- c) ☐ спиральная модель ЖЦПО допускает корректировки во время разработки

10. Какие возвраты невозможны при разработке по водопадной модели

- a) ☐ возврат от кодирования к разработке системных требований
- b) ☐ возврат от тестирования к кодированию
- c) ☐ возврат от тестирования к анализу

11. Стратегия, в которой впервые были выделены шаги разработки, четко определенные и следующие в точном порядке

- a) ☐ спиральная
- b) ☒ водопадная (каскадная)
- c) ☐ инкрементная

12. Каковы достоинства водопадной модели

- a) ☐ интеграция всех результатов в конце разработки
- b) ☐ ограничение возможности возвратов на произвольный шаг назад
- c) ☐ требование полного окончания фазы-деятельности
- d) ☐ неустойчивость к сбоям в финансировании проекта

13. В какой модели конструирования ПО каждый этап представляет собой фазу разработки

- a) ☐ в любой модели
- b) ☐ в водопадной модели
- c) ☐ ни в одной из указанных моделей
- d) ☒ в спиральной модели

14. Выберите сходство между большими и малыми программами:

- a) ☐ гарантия качества
- b) ☐ создаются для получения конкретных результатов
- c) ☐ наличие квалифицированных менеджеров проекта
- d) ☐ применение регламентированных стандартами процессы, этапы и документы

15. Какова цель создания прототипа при использовании водопадной модели

- a) ☐ определение основных рисков
- b) ☐ уменьшение рисков разработки
- c) ☐ обоснованное принятие главных архитектурных решений

16. Модель создания прототипов

- a) ☐ допускает корректировки по ходу разработки
- b) ☐ основывается на разработке работающего прототипа системы
- c) ☐ серия последовательных итераций при разработке программы

17. Какой этап жизненного цикла программы выполняется вначале

- a) ☐ определение стоимости
- b) ☒ оценка осуществимости проекта
- c) ☐ техническое задание

18. Техническое задание:

- a) ☐ разрабатывается заказчиком
- b) ☐ разрабатывается производителем
- c) ☐ разрабатывается заказчиком при согласовании с производителем

19. Оформление постановки задачи по стандартным правилам:

- a) ☐ является одним из этапов жизненного цикла программы
- b) ☐ является ресурсоемкой операцией
- c) ☐ уменьшает вероятность разночтения пунктов задания разработчиками и заказчиком

20. При использовании подхода быстрой разработки приложений (RAD)

- a) ☐ пользователи не должны вовлекаться в процесс разработки.
- b) ☐ желательно, чтобы пользователи вовлекались в процесс разработки.
- c) ☐ пользователи обязательно должны вовлекаться в процесс разработки.

21. Когда осуществляется переход на следующую стадию работ при использовании каскадной модели ЖЦ

- a) ☐ после заключения соответствующего договора.
- b) ☐ после того, как работа на текущей стадии полностью завершена.
- c) ☐ после того, как работа на текущей стадии выполнена не менее чем на 90%.

22. Что не является преимуществом каскадного подхода к проектированию информационной системы?

- a) ☐ на каждой стадии формируется законченный набор проектной документации.
- b) ☐ возможность планировать сроки завершения всех работ.
- c) ☐ возможность планировать затраты на проектирование.
- d) ☐ нет необходимости полной и точной формулировки требований к проектируемой информационной системе на начальной стадии.

23. Спиральная модель ЖЦ

- a) ☐ исключает использование каскадного подхода.
- b) ☐ не исключает использование каскадного подхода на начальных стадиях проекта.
- c) ☐ не исключает использование каскадного подхода на завершающих стадиях проекта, когда требования к системе полностью определены.

24. Основной проблемой спирального ЖЦ является
- a) ☐ необходимость точной формулировки требований к системе.
 - b) ☐ определение момента перехода на следующую стадию разработки.
 - c) ☐ запаздывание с получением результата.
 - d) ☐ высокий риск создания системы не удовлетворяющей потребностям пользователя.
25. При разработке каких приложений следует использовать подход RAD?
- a) ☐ приложений с ярко выраженной интерфейсной частью, разрабатываемых для конкретного заказчика.
 - b) ☐ приложений, в которых отсутствует ярко выраженная интерфейсная часть.
 - c) ☐ приложений, в которых выполняются сложные расчеты.
26. Обычно в качестве нотаций, используемых при построении моделей, применяются
- a) ☐ специализированные формальные языки.
 - b) ☐ сетевые графики.
 - c) ☐ графические диаграммы.
 - d) ☐ универсальные алгоритмические языки.
27. Современная технология проектирования ПО должна обеспечивать поддержку
- a) ☐ всех процессов ЖЦ ПО.
 - b) ☐ основных процессов ЖЦ ПО.
 - c) ☐ вспомогательных процессов ЖЦ ПО.
 - d) ☐ организационных процессов ЖЦ ПО.
 - e) ☐ основных и вспомогательных процессов ЖЦ ПО.
28. Укажите, какое из двух требований является требованием к правильной декомпозиции сложной системы
- a) ☐ связность отдельных частей внутри каждой подсистемы должна быть минимальной.
 - b) ☐ связность отдельных частей внутри каждой подсистемы должна быть максимальной.
29. Что такое жизненный цикл?
- a) ☐ этапы развития от момента зарождения до прекращения функционирования
 - b) ☐ процесс построения ПО
 - c) ☐ реализация и тестирование системы
 - d) ☐ совокупность процессов и этапов развития
30. Выберите из нижеперечисленных пунктов основные этапы модели ЖЦ:
- a) ☐ определение потребностей
 - b) ☐ создание и производство
 - c) ☐ создание всей документации
 - d) ☐ испытание системы
 - e) ☐ распространение и продажа
 - f) ☐ сопровождение и мониторинг
 - g) ☐ снятие с эксплуатации

Примерный вариант модульного контроля № 2:

Задание: Отметьте верные утверждения.

1. Последовательность действий в структурных компонентах описывает
- a) ☐ разработке данных
 - b) ☐ разработке архитектуры
 - c) ☐ процедурной разработке
2. Результат преобразования информационной модели анализа в структуры данных происходит при
- a) ☐ разработке данных
 - b) ☐ разработке архитектуры
 - c) ☐ процедурной разработке
3. Выберите верные утверждения:
- a) ☐ оценка осуществимости позволяет определить стоимость работы
 - b) ☐ для оценки осуществимости необходима декомпозиция системы
 - c) ☐ планирование одна из фаз оценки осуществимости
4. Технология программирования зависит:
- a) ☐ от целевой ЭВМ
 - b) ☐ от коллектива разработчиков
 - c) ☐ от инструментальных средств разработки
5. Какая из указанных практик проводится в начале каждого Sprint?
- a) ☐ Daily Scrum Meeting
 - b) ☐ Sprint Review Meeting
 - c) ☐ Sprint Planning Meeting
6. Какие из перечисленных принципов справедливы для организации процесса по XP (Extreme Programming)?
- a) ☐ тестирование полностью отсутствует
 - b) ☐ заказчик участвует в разработке
 - c) ☐ непрерывная интеграция
7. Для чего используется рабочий продукт?

- a) ☐ для обмена результатами b) ☐ для контроля разработки c) ☐ для устранения накладных расходов

8. Какие специалисты вовлечены в разработку и/или использование ПО?

- a) ☐ менеджеры b) ☐ продавцы-маркетологи
c) ☐ технические писатели d) ☐ программисты

9. Каково назначение пятнадцатиминутных ежедневных совещаний (Daily Scrum Meeting) в Scrum?

- a) ☐ корректировка рабочего плана согласно текущей ситуации
b) ☐ выработка новых требований
c) ☐ обозначение путей решения текущих проблем
d) ☐ понимание того, что произошло со времени предыдущего совещания

10. В чем заключается "плоскость" модели MSF?

- a) ☐ ответственность за проект несет официальный лидер команды
b) ☐ коммуникации внутри группы четкие, ясные, дружественные
c) ☐ уровень заинтересованности каждого в результате очень высок
d) ☐ все отвечают за проект в равной степени

11. Какой этап следует за анализом результатов и пересмотром требований при использовании метода Scrum?

- a) ☐ планирование итерации b) ☐ выполнение итерации c) ☐ создание требований к продукту

12. В чем заключается принцип гибкости в методологии MSF?

- a) ☐ в том, что условия проекта изменяются вместе с эффективностью управленческой деятельности
b) ☐ в том, что условия проекта непрерывно изменяются при неизменной эффективности управленческой деятельности
c) ☐ в том, что требования формулируются в начале проекта и после этого существенно не изменяются

13. Главными областями программной инженерии не являются:

- a) ☐ конструирование ПО b) ☐ инженерия требований
c) ☐ процесс инженерии ПО d) ☐ управление конфигурацией

14. Архитектурная схема может быть:

- a) ☐ клиент-серверная b) ☐ сосредоточенная
c) ☐ компонент-серверная d) ☐ распределенная

15. Способы ограничения сложности ПО

- a) ☐ декомпозиция b) ☐ иерархический подход
c) ☐ отказ от больших по объему программ d) ☐ абстракции

16. Технологический цикл конструирования программной системы включает следующие процессы:

- a) ☐ анализ b) ☐ синтез c) ☐ оптимизация
d) ☐ сопровождение e) ☐ соединение

17. К какому типу проектов относятся проекты по разработке ПО

- a) ☐ к промышленным проектам b) ☐ и к творческим, и к промышленным проектам
c) ☐ к творческим проектам

18. Какое свойство зависит от размера программных объектов

- a) ☐ сложность b) ☐ нематериальность
c) ☐ согласованность d) ☐ изменяемость

19. Надежность ПО определяется следующими подхарактеристиками (выберите нужные):

- a) ☐ анализируемость b) ☐ устойчивость к отказам c) ☐ понятность
d) ☐ удобство установки e) ☐ способность к восстановлению работоспособности при отказах
f) ☐ зрелость, завершенность g) ☐ пригодность к определенной работе
h) ☐ точность, правильность

20. Выберите из списка 6 характеристик качества ПО, определенных стандартом ISO 9126

- a) ☐ изменяемость, удобство внесения изменений b) ☐ сопровождаемость
c) ☐ функциональность d) ☐ устойчивость к отказам e) ☐ стоимость
f) ☐ эффективность g) ☐ переносимость h) ☐ понятность
i) ☐ практичность j) ☐ надежность k) ☐ качественность

21. Методология IDEF0 это

- a) ☐ методология моделирования данных
b) ☐ методология объектно-ориентированного моделирования.
c) ☒ методология функционального моделирования.

22. Каждая отдельно взятая модель IDEF0 должна разрабатываться исходя

- ☐ из единственной, заранее определенной точки зрения (например, только исходя из точки зрения руководителя организации, или только из точки зрения руководителя некоторого отдела).
☐ из различных, заранее определенных точек зрения (например, из точки зрения руководителя организации и из точки зрения руководителя некоторого отдела).
☐ из различных точек зрения, меняющихся во время хода работы над проектом (например, вначале модель разрабатывалась исходя из точки зрения руководителя предприятия, а потом исходя из точки зрения руководителя некоторого отдела).
☐ из точки зрения руководителя организации, для которой разрабатывается модель.

23. В модели IDEF3 определены

- ☐ два вида связей (временное предшествование и объектный поток).
☐ три вида связей (временное предшествование, объектный поток и нечеткое отношение).
☐ четыре вида связей (временное предшествование, объектный поток, нечеткое отношение и функциональная связь).

24. Сколько типов асинхронных соединений определено в модели IDEF3?

- ☐ Два. ☐ Три. ☐ Четыре. ☐ Пять.

25. В диаграммах DFD стрелки могут начинаться

- ☐ на правой стороне блока. ☐ на нижней стороне блока. ☐ на левой стороне блока.
☐ на верхней стороне блока. ☒ на любой стороне блока.

26. Внешние сущности на диаграммах DFD в нотации, используемой BPwin, обозначаются в виде

- ☐ овалов. ☐ прямоугольников с закругленными углами.
☒ прямоугольников. ☐ стрелок.

27. В нотации IDEF прямоугольником обозначается

- ☒ независимая сущность. ☐ зависимая сущность.
☐ идентифицирующая связь. ☐ не идентифицирующая связь.

29. В нотации IDEF идентифицирующая связь обозначается

- ☐ прямоугольником.
☒ сплошной линией между сущностью-родителем и сущностью-потомком с точкой на конце линии у сущности-потомка.
☐ пунктирной линией между сущностью-родителем и сущностью-потомком с точкой на конце линии у сущности-потомка.
☐ сплошной линией между сущностью-родителем и сущностью-потомком с точкой на конце линии у сущности-родителя.
☐ пунктирной линией между сущностью-родителем и сущностью-потомком с точкой на конце линии у сущности-родителя.

30. Пусть между двумя сущностями установлена идентифицирующая связь. Какое из правил ссылочной целостности не может быть применено при удалении экземпляра родительской сущности?

- ☐ RESTRICT.
☐ CASCADE.
☒ SET NULL.
☐ Любое из перечисленных правил.

31. Что такое проектирование?

- a. процесс определения архитектуры, компонентов, интерфейсов, других характеристик системы
b. организация процесса разработки
c. предварительный макет, эскиз, план системы на бумаге
d. разработка требований

Вопросы для подготовки к экзамену

1. Жизненный цикл ПО. Классификация стратегий конструирования ПО.
2. Стратегии конструирования ПО. Классическая модель проектирования ПО
3. Стратегии конструирования ПО. Прототипирование
4. Стратегии конструирования ПО. Инкрементная модель
5. Стратегии конструирования ПО. Спиральная модель
6. Стратегии конструирования ПО. Быстрая разработка приложений (RAD)
7. Стратегии конструирования ПО. Rational Unified Process
8. Стратегии конструирования ПО. Экстремальное программирование
9. Стратегии конструирования ПО. Методология SCRUM
10. Управление требованиями. Требования в программных проектах. Проблемы определения требований.
11. Разработка требований. Выявление и анализ требований.
12. Документирование и организация требований.
13. Документирование требований. Виды документов.
14. Изменения требований.
15. Планирование и управление требованиями. Прослеживание требований.
16. Программные средства управления требованиями
17. Риски в программных проектах.
18. Управление риском в программных проектах: идентификация, анализ, ранжирование
19. Управление риском в программных проектах: планирование, разрешение, наблюдение.
20. Ресурсы в программных проектах. Управление ресурсами
21. Роли участников в программных проектах
22. Проектные активности программных проектов
23. Временные сущности программных проектов
24. Визуализация плана. Наблюдение за программным проектом
25. Понятие дефекта программного обеспечения. Характеристики дефектов
26. Жизненный цикл дефекта.
27. Системы управления дефектами.
28. Предпосылки для версионирования ПО. Ветвление.
29. Системы контроля версий. Типы СКВ. Общие принципы организации.
30. Системы контроля версий. Типовые операции
31. Сборка программных проектов. Проблемы при сборке программных проектов
32. Сборка программных проектов. Окружение для сборки
33. Сборка программных проектов. Общие требования к системе сборки
34. Сборка программных проектов. Версии в программных проектах
35. Качество программного обеспечения. Характеристики качества.
36. Оценка качества ПО. Программометрика.
37. Метрики программного обеспечения. Метрики Чидамбера и Кемерера.
38. Метрики программного обеспечения. Метрики Лоренца и Кидда
39. Метрики программного обеспечения. Метрики MOOD
40. Метрики сложности ПО.
41. Применение метрик ПО. Аудит программного кода.
42. Методы обеспечения качества ПО. Верификация, метод проверки моделей
43. Методы обеспечения качества ПО. Статический анализ ПО
44. Методы обеспечения качества ПО. Рефакторинг
45. Тестирование ПО. Функциональное и структурное тестирование
46. Организация тестирования ПО. Тестирование элементов
47. Организация тестирования ПО. Тестирование интеграции. Тестирование правильности
48. Организация тестирования ПО. Системное тестирование.
49. Организация тестирования ПО. Особенности тестирования приложений с ГПИ
50. Документирование ПО. Виды программной документации.

51. Документирование ПО. Использование UML для документирования
52. Документирование ПО. Генерация документации
53. Документирование ПО. Промышленные технологии документирования.
54. Качество процесса проектирования ПО. Модель зрелости процесса проектирования CMM

8. Учебно-методическое и информационное обеспечение дисциплины (модуля):

8.1. Основная литература

1. Гагарина Л.Г. и др. Технология разработки программного обеспечения: учеб. пособие для вузов. – М.: ИД ФОРУМ:ИНФРА-М, 2009. – 400 с.
2. Конструирование программного обеспечения: учебное пособие / составитель А. А. Романов. – Ульяновск: УлГТУ, 2016. – 126 с.
3. Орлов С.А., Цилькер Б.Я., Технологии разработки программного обеспечения: Учебник для вузов. 4-е изд. – СПб.: Питер, 2012. – 608 с.
4. Рудинский И.Д. Технология проектирования автоматизированных систем обработки информации и управления. Учебное пособие для вузов. – М.: Горячая линия - Телеком, 2011 – 304 с.

8.2. Дополнительная литература

1. Ф.Брукс. «Мифический человеко-месяц или как создаются программные системы»
2. Макконнелл С., «Совершенный код. Практическое руководство по разработке программного обеспечения»
3. Фаулер М., Бек К., Брант Д., Робертс Д., Алдайк У. «Рефакторинг: улучшение существующего кода»
4. McConnell S., Rapid Development: Taming Wild Software Schedules.
5. Hans van Vliet, «Software Engineering: Principles and Practice», (2000)

8.3. Программное обеспечение и Интернет-ресурсы

Программное обеспечение: ОС *Windows*, *BPWin 4.0*

Интернет-ресурсы

1. Software Engineering Conference (Russia) 2005, 2006, 2007 <http://www.secr.ru/>
2. Software Engineering - Guide to the Software Engineering Body of Knowledge (SWEBOK) TECHNICAL REPORT ISO/IEC TR 19759 IEEE First edition 2005-09-15. <http://www.secr.ru/>
3. CMMI® for Development, Version 1.2, CMU/SEI-2006-TR-008 ESC-TR-2006-008

8.4. Методические указания и материалы по видам занятий

Методические указания к лабораторным работам по дисциплине «Конструирование программного обеспечения» в электронном варианте

9. Материально-техническое обеспечение дисциплины (модуля):

Учебный кабинет, лаборатория ИТО ИТИ.

10. Методические рекомендации по организации изучения дисциплины:

Студент, изучающий дисциплину, должен, с одной стороны, овладеть общим понятийным аппаратом, а с другой стороны, должен научиться применять теоретические знания на практике.

В результате изучения дисциплины студент должен знать основные определения, понятия, основы методик конструирования программного обеспечения, основные метрики программного обеспечения.

Успешное освоение курса требует напряженной самостоятельной работы студента. В программе курса отведено минимально необходимое время для работы студента над темой. Самостоятельная работа включает в себя:

- чтение и конспектирование рекомендованной литературы;

- проработку учебного материала (по конспектам занятий, учебной и научной литературе), подготовку ответов на вопросы, предназначенные для самостоятельного изучения, доказательство отдельных утверждений, свойств, решение задач;

- подготовка к экзамену.

Руководство и контроль за самостоятельной работой студента осуществляется в форме индивидуальных консультаций.

Важно добиться понимания изучаемого материала, а не механического его запоминания. При затруднении изучения отдельных тем, вопросов следует обращаться за консультациями к лектору.

Рабочая учебная программа по дисциплине «Конструирование программного обеспечения» составлена в соответствии с требованиями Федерального Государственного образовательного стандарта ВО по направлению 09.03.04 «ПРОГРАММНАЯ ИНЖЕНЕРИЯ» и учебного плана по профилю подготовки «Разработка программно-информационных систем».

ТЕХНОЛОГИЧЕСКАЯ КАРТА ДИСЦИПЛИНЫ

Курс 3

Семестр 6

Группа ИТ16ДР62ПИ1

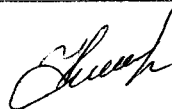
Преподаватель — лектор Белоконов О.С.

Преподаватели, ведущие практические занятия – Белоконов О.С.

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Наименование дисциплины/курса	Уровень образования (бакалавриат, специалитет, магистратура)	Статус дисциплины в учебном плане (А, Б, В)	Количество зачетных единиц	
Конструирование программного обеспечения	бакалавриат	А	4	
СМЕЖНЫЕ ДИСЦИПЛИНЫ ПО УЧЕБНОМУ ПЛАНУ:				
«Объектно-ориентированное программирование», «Введение в программную инженерию», «Разработка и анализ требований», «Тестирование программного обеспечения», «Проектирование и архитектура программных систем», «Управление программными проектами».				
БАЗОВЫЙ МОДУЛЬ (проверка знаний и умений по дисциплине)				
Тема, задание или мероприятие текущего контроля	Виды текущей аттестации	Аудиторная или внеаудиторная	Минимальное количество баллов	Максимальное количество баллов
Тест №1	Т1	аудиторная	10	20
Лабораторная работа №1	ЛР1	аудиторная	5	10
Лабораторная работа №2	ЛР2	аудиторная	5	10
РУБЕЖНЫЙ КОНТРОЛЬ	РК		20	40
Тест №2	Т2	аудиторная	20	40
Лабораторная работа №3	ЛР3	аудиторная	5	10
Лабораторная работа №4	ЛР4	аудиторная	5	10
РУБЕЖНАЯ АТТЕСТАЦИЯ	РА		30	60
Итого			50	100

Составитель, ст. преподаватель



О.С. Белоконов

Рабочая учебная программа рассмотрена методической комиссией инженерно-технического института протокол № 1 от «12» 09 2018 г. и признана соответствующей требованиям Федерального Государственного образовательного стандарта и учебного плана по направлению 09.03.04 «ПРОГРАММНАЯ ИНЖЕНЕРИЯ».

Председатель МК ИТИ



Е.И. Андрианова

Зав. кафедрой ПОВТ и АС, доцент



С.Г. Федорченко