

Министерство науки и высшего образования Российской Федерации

Томский государственный университет
систем управления и радиоэлектроники

М. Е. Антипин

Программная инженерия

Методические указания по выполнению самостоятельной работы студентов

Томск
2022

УДК 004.02
ББК 3стд2-02
А 72

Рецензент:

Лобода Ю.О., доцент каф. управления инновациями ТУСУР,
канд. пед. наук

Антипин, Михаил Евгеньевич

А 72 Программная инженерия: Методические указания по выполнению самостоятельной работы студентов/ М.Е. Антипин. – Томск: Томск. гос. ун-т систем упр. и радиоэлектронники, 2022. – 13 с.

Методические указания содержат рекомендации и материалы, необходимые для самостоятельной работы студентов по дисциплине «Программная инженерия».
Для студентов высших учебных заведений.

Одобрено на заседании кафедры УИ, протокол № 1 от 31.01.2022.

УДК 004.02
ББК 3стд2-02

© Антипин М.Е., 2022
© Томск. гос. ун-т систем упр. и
радиоэлектронники, 2022

Оглавление

1. Общие положения	4
2 Разделы и содержание дисциплины	5
3 Организация самостоятельной работы студентов.....	6
4 Терминология дисциплины.....	7
5 Вопросы для самоконтроля	8
6 Тестовые вопросы по дисциплине	9
7 Контрольные вопросы.....	12
Список рекомендуемой литературы.....	13

1. Общие положения

Данные методические указания разработаны для студентов, обучающихся в Томском государственном университете систем управления и радиоэлектроники (далее - Университет).

Структура дисциплины «Программная инженерия» предполагает выполнение студентами самостоятельной работы как по освоению теоретического материала, так и в рамках выполнения практических заданий. Рекомендации по выполнению практических заданий и лабораторных работ приведены в соответствующих методических указаниях.

В ходе выполнения самостоятельной работы студентам прививаются навыки работы с учебно-методической документацией, умения увязывать теоретические знания с практикой, четко излагать свои мысли, отвечать на вопросы, оформлять и представлять результаты работы.

Рекомендации подготовлены с целью помочь студентам в успешном освоении дисциплины и прохождении аттестации, давая информацию об ее структуре и оценочных средствах.

2 Разделы и содержание дисциплины

Дисциплина «Программная инженерия» содержит следующие разделы:

1. Введение в предметную область:
Основные понятия и определения программной инженерии и программных продуктов. Сущность программной инженерии. Зарождение и становление программной инженерии. Становление отечественной технологии программирования. Содержание базовых стандартов. Модель зрелости процесса конструирования ПО.
2. Жизненный цикл программных продуктов. Основные модели жизненного цикла:
Основные модели жизненного цикла программных продуктов и этапы жизненного цикла. Методологии разработки программ и жизненные циклы для разных методологий. Этапы: Подготовка, разработка требований, анализ, проектирование, разработка, тестирование, оптимизация, внедрение, эксплуатация, сопровождение, модернизация.
3. Формальные и полужформальные методы спецификации, верификации и доказательства правильности:
Методы формальной спецификации требований. Модели. Дедуктивный анализ и проверка моделей. Методы доказательства корректности программ. Абстрактная интерпретация. Формальные методы в тестировании. Полужформальные методы разработки и тестирование. Полужформальные методы верификации разработки и тестирование. Зависимость методов.
4. Теория и методы проектирования. Новые подходы к разработке:
Теория проектирования программных продуктов (ПП). Инженерия изготовления ПП. Процессы разработки ПП и инженерия доменов. Инженерия повторного использования. Основы метода генерации ПП. Инструментальные средства. Фабрики программ и интеграции компонент.
5. Управление программными проектами:
Управление содержанием и сроками реализации проекта. Управление качеством программного продукта. Управление рисками. Риски и рискообразующие факторы в проекте. Качественный и количественный анализ рискообразующих факторов. Стратегии управления рисками. Способы выявления, оценки, мониторинга, минимизации и устранения влияния рисков.
6. Технологии разработки программного обеспечения:
Технологи разработки программных проектов. Использование CASE-технологии в проектировании ПП. Автоматический контроль и генерация кода на основе моделей. Повторное использование. Применение методологии прототипирования. Автоматизированное документирование. Коллективная разработка. Модели ПП и инструментальные средства для их создания.
7. Теория языков программирования. Классификация языков программирования
Языки программирования. Классификация. Синтаксис языковых конструкций, описания переменных, записей, структур и функций. Работа с регулярными выражениями. Конечных автоматы. Структурные, процедурные, объектно-ориентированные. функциональные, мультипарадигмальные, графические, баз данных, аппаратные, параллельные

3 Организация самостоятельной работы студентов

Самостоятельная проработка лекционного материала направлена на получение навыков работы с конспектом, структурирования материала, а также умения выделить основные пункты и положения, изложенные на лекции. Целесообразно ознакомиться с информацией, представленной в файлах, содержащих презентации лекций, предоставляемых преподавателем. Кроме того, проработка лекционного материала способствует более глубокому пониманию и прочному запоминанию теоретической части дисциплины. Проработка лекционного материала включает деятельность, связанную с изучением рекомендуемых преподавателем источников, в которых отражены основные моменты, затрагиваемые в ходе лекций.

Важное место отведено работе с собственноручно составленным конспектом лекций. При конспектировании во время лекции помните, что не следует записывать все, что говорит и/или демонстрирует лектор: старайтесь выявить главное и записать только это. Цель конспекта – формирование целостного логически выстроенного взгляда на круг вопросов, затрагиваемых в ходе изучения соответствующей темы.

При проработке лекционного материала необходимо: - отработать прослушанную лекцию (прочитать конспект, прочитать дополнительную литературу по аналогичной теме и сопоставить записи с конспектом) и восполнить пробелы в знаниях, если таковые обнаружались; - перед каждой последующей лекцией прочитать предыдущую, чтобы обновить знания для восприятия последующей новой информации.

В ходе изучения дисциплины некоторые из тем курса выносятся исключительно на самостоятельное изучение. Следует обратить внимание на то, что работа по этим темам включает как подбор источников, так и изучение их содержания. В зависимости от особенностей усвоения учебного материала студентами и объема аудиторной работы некоторые из вопросов, рассматриваемые в ходе проведения лекций и лабораторных работ, могут быть также вынесены в формат самостоятельного изучения.

4 Терминология дисциплины

Чтобы свободно ориентироваться в материалах дисциплины студенту следует ознакомиться с применяемой терминологией:

- Программа (program) – это набор операторов, который может быть представлен как единое целое в некоторой вычислительной системе и который используется для управления поведением этой системы.
- Программирование (в узком смысле) – процесс кодирования и отладки программы в рамках реального проекта.
- Программирование (programming) (в широком смысле) – все технические операции, необходимые для создания программы, включая анализ требований и все стадии разработки и реализации.
- Программная инженерия - системный подход к анализу, проектированию, оценке, реализации, тестированию, обслуживанию и модернизации программного обеспечения. Программная инженерия занимается разработкой систематических моделей и надежных методов производства высококачественного программного обеспечения, и данный подход распространяется на все уровни — от теории и принципов до реальной практики создания программного обеспечения. ПИ является отраслью информатики (computer science), это инженерная дисциплина, которая изучает вопросы построения компьютерных программ, отражает закономерности развития программирования, обобщает опыт программирования в виде комплекса знаний и правил регламентации инженерной деятельности разработчиков ПО.
- Технология программирования – процессы программной инженерии, методы программирования, инструментальные средства.
- Программное обеспечение – совокупность программ для обработки информации и комплект документации, необходимой для ее эксплуатации.
- Жизненный цикл – развитие системы, продукта, услуги, проекта начиная со стадии идеи и заканчивая завершением применения.
- Конфигурационное управление (англ. Software configuration management, SCM) в программной инженерии — комплекс методов, направленных на систематический учёт изменений, вносимых разработчиками в программный продукт в процессе его разработки и сопровождения, сохранение целостности системы после изменений, предотвращение нежелательных и непредсказуемых эффектов, формализацию процесса внесения изменений.

5 Вопросы для самоконтроля

При изучении материала дисциплины очень важно самостоятельно контролировать освоение материала. Сделать это удобно, отвечая на вопросы для самоконтроля:

1. Определите понятия «программный продукт» и «программное обеспечение». Чем отличаются «коробочные» программные продукты от «заказных»?
2. Что понимается под «жизненным циклом ПО»?
3. Определите понятия «инженерия» и «программная инженерия». Что общего и в чем различия между программной инженерией и другими инженерными отраслями?
4. Определите основные черты технологий модульного и структурного программирования, технологии структурного проектирования.
5. Определите основные черты технологии объектно-ориентированного программирования.
6. Определите и дайте краткую характеристику основным нефункциональным требованиям к программному продукту в соответствии с ГОСТ Р ИСО/МЭК 25010-2015.
7. Прокомментируйте понятия «жизненный цикл ПО», «стадия разработки» и «модель жизненного цикла ПО».
8. Проанализируйте и прокомментируйте эталонные модели процесса анализа требований программных средств и процесса детального проектирования программных средств, определенных стандартом ISO 12207-2010.
9. Опишите основные черты, преимущества, недостатки и области эффективного применения каскадной модели.
10. Опишите основные черты V-модели. Перечислите функции разработчиков и тестировщиков на фазе проекта «Детальное проектирование компонентов программной системы».
11. Дайте общую характеристику эволюционной модели. В каких условиях и за счет чего применение этой модели может привести к сокращению сроков выполнения программного проекта?
12. В чем отличие метода пробных разработок от метода прототипирования? Назовите условия применения этих методов.
13. Опишите основные черты спиральной модели. Почему спиральную модель относят к категории гибридных итерационных моделей? Чем отличается эта модель от каскадной и эволюционной моделей?
14. Используя соответствующие профессиональные стандарты определите состав обобщенных трудовых функций и требования к квалификационным уровням IT-специалистов, работающих по профессиям «программист», «специалист по тестированию», «архитектор ПО».
15. Дайте краткую характеристику типовой ролевой модели команды программного проекта. Перечислите основные функции разработчика, тестировщика, технолога и инженера по качеству

6 Тестовые вопросы по дисциплине

Тестирование является обязательной частью аттестации по дисциплине, а также важным средством проверки остаточных знаний студентов. Подготовка к тестированию предполагает повторение материала по всем разделам дисциплины. Для тестирования может использоваться следующий перечень вопросов (с вариантами ответов):

1. Какие вопросы решает программная инженерия?
 - а) вопросы жизненного цикла разработки ПО;
 - б) вопросы управления процессом разработки ПО;
 - в) вопросы улучшения процесса разработки ПО;
 - д) вопросы применения инструментальных средств для разработки ПО.
2. К какому типу проектов относятся проекты по разработке программных систем и ПО:
 - а) к творческим проектам;
 - б) к промышленным проектам;
 - в) и к творческим, и к промышленным проектам.
3. В стандарте ISO/IEC 12207 «Процессы поддержки» в жизненном цикле не включают:
 - а) управление конфигурацией ПО;
 - б) установку ПО;
 - в) валидацию ПО.
4. Что не является проблемой анализа требований пользователями:
 - а) пользователи не понимают то, что они хотят;
 - б) у пользователей нет ясного представления об их требованиях;
 - в) пользователи не соглашаются с ранее записанными требованиями;
 - г) пользователи настаивают на новых требованиях после того, как стоимость и график работ установлены;
 - д) пользователи участвуют в обзорах требований;
 - е) пользователи технически не подготовлены.
5. Мобильность программного обеспечения это:
 - а) способность ПО выполнять набор функций, которые удовлетворяют потребности пользователей;
 - б) способность ПО выполнять определенный набор функций в заданных условиях в течение заданного периода времени;
 - в) способность ПО быть перенесенным из одной среды (аппаратного / программного) в другую.
6. Основные причины появления программной инженерии как методологии проектирования:
 - а) Низкая надежность работы вычислительной техники.
 - б) Частое изменение средств программирования.
 - в) Низкая оплата труда программистов.
 - г) Необходимость привлечения различных специалистов для разработки ПП.
 - д) Низкое качество управления процессами разработки ПП.
 - е) Нечеткая формулировка и частое изменение требований со стороны заказчика.
7. Программная инженерия определяется как системный подход к:
 - а) анализу, проектированию, оценке, реализации, тестированию, обслуживанию и модернизации программного обеспечения;
 - б) оценке, реализации, тестированию, обслуживанию программного обеспечения;
 - в) анализу, оценке, проектированию, реализации и тестированию программного обеспечения;
 - г) тестированию, обслуживанию и модернизации программного обеспечения.

8. Ключевые атрибуты в технологическом процессе создания программного продукта:
- а) методы;
 - б) жизненный цикл;
 - в) использование средства автоматизации разработки программ;
 - г) стандарты;
 - д) процедуры.
9. Классические модели жизненного цикла ПО:
- а) каскадная;
 - б) тройная спираль;
 - в) эволюционная;
 - г) объектная;
 - д) спиральной;
 - е) реляционная;
 - ж) V-модель;
 - з) модель водопада;
 - и) сетевая.
10. Выберите элементы IDEF0 модели:
- а) вход;
 - б) механизм;
 - в) функция;
 - г) выход;
 - д) процедура;
 - е) управление.
11. Выделите этапы жизненного цикла ПО:
- а) анализ требований;
 - б) проектирование;
 - в) структурный анализ;
 - г) конструирование;
 - д) абстрагирование;
 - е) тестирование;
 - ж) эксплуатация и сопровождение;
 - з) утилизация.
12. Области знаний по разработке ПО определенные стандартом SWEBOOK:
- а) анализ требований;
 - б) проектирование;
 - в) конструирование;
 - г) структурный анализ кода;
 - д) абстрагирование;
 - е) тестирование;
 - ж) эксплуатация;
 - з) моделирование.
13. Основные характеристики качества ПО определенных в стандарте ISO/МЭК 9126:1991:
- а) эргономика;
 - б) мобильность;
 - в) надежность;
 - г) стоимость;
 - д) производительность;
 - е) функциональные возможности;
 - ж) эффективность;
 - з) сопровождаемость;

- и) практичность;
 - к) применяемость.
14. Процесс разбиения сложной задачи на несколько простых подзадач:
- а) абстракция;
 - б) декомпозиция;
 - в) реинжиниринг;
 - г) верификация.
15. Правильный фрагмент последовательности этапов при каскадной модели жизненного цикла разработки ПО:
- а) Определение требований -> Тестирование -> Реализация;
 - б) Анализ -> Проектирование -> Кодирование -> Тестирование;
 - в) Проектирование -> Определение требований -> Реализация.
16. Легкость применения программного обеспечения это:
- а) характеристики ПО, позволяющие минимизировать усилия пользователя по подготовке исходных данных для применения ПО;
 - б) отношение уровня услуг, предоставляемых пользователю ПО при заданных условиях, к объему используемых ресурсов;
 - в) характеристики ПО, позволяющие минимизировать усилия по внесению изменений для устранения в нем ошибок;
 - г) характеристики ПО, позволяющие минимизировать усилия по модификации ПО;
 - д) способность ПО выполнять набор функций, которые удовлетворяют потребности пользователя.
17. UML:
- а) язык программирования, имеющий синтаксис языка Си;
 - б) унифицированный язык моделирования использующий нотацию диаграмм;
 - в) набор спецификаций для определения качества программного обеспечения.
18. Кто разрабатывает uml модель?
- а) программист;
 - б) архитектор;
 - в) программный менеджер.
19. Устойчивость программного обеспечения — это:
- а) свойство, характеризующее способность ПО автоматически завершать функционирование, несмотря на ошибочные входные данные;
 - б) свойство, характеризующее способность ПО продолжать корректное функционирование, несмотря на ошибочные входные данные;
 - в) свойство, способное противостоять преднамеренным или непреднамеренным деструктивным действиям пользователя;
20. На этапе тестирования пользователь выполняет:
- а) синтаксическую отладку;
 - б) выбор тестов и метода тестирования;
 - в) определение формы выдачи результатов;
 - г) определение архитектуры ПО.
21. Выполнение тестирования ПО без знания как оно спроектировано и реализовано называют тестированием методом:
- а) белого ящика;
 - б) прозрачного ящика;
 - в) черного ящика;
 - г) жёлтого ящика.

7 Контрольные вопросы

Приведенный ниже перечень вопросов рекомендуется использовать студенту для подготовки к аттестации по дисциплине:

1. Дайте понятие программного проекта и перечислите его специфические особенности.
2. Приведите определение программного продукта.
3. Перечислите свойства ПП как объекта интеллектуальной собственности.
4. Дайте понятия цели, результата и ограничений программного проекта.
5. Перечислите и прокомментируйте требования к формулировке целей.
6. Раскройте смысл характеристик «железного треугольника» при управлении программными проектами.
7. Приведите понятие жизненного цикла программного продукта и назовите стандарты, регламентирующие этапы ЖЦ.
8. Перечислите и прокомментируйте содержание девяти областей знаний стандарта РМВОК.
9. Перечислите и прокомментируйте содержание пяти этапов жизненного цикла программного проекта стандарта РМВОК.
10. Перечислите и прокомментируйте содержание процессов управления программным проектом стандарта ГОСТ Р ИСО/МЭК 12207–2010.
11. Дайте определение и приведите примеры понятий «риск» и «рискообразующий фактор».
12. Приведите пример и прокомментируйте по схеме «условие» – «последствие» – «воздействие» описание внутренних факторов риска программного проекта.
13. Приведите пример и прокомментируйте по схеме «условие» – «последствие» – «воздействие» описание внешних факторов риска программного проекта.
14. Раскройте содержание и методы описания показателей вероятности и негативных последствий рискообразующих факторов.
15. Поясните процедуру ранжирования рискообразующих факторов по степени опасности последствий от их наступления.
16. Раскройте содержание модели функциональных зависимостей определения рисков программного проекта.
17. Раскройте содержание стратегий по управлению рисками, приведите примеры конкретных мероприятий по каждой из стратегий.
18. Раскройте содержание этапа мониторинга и управления рисками.
19. Стандарты в области программной инженерии.
20. Модели жизненного цикла.
21. Методы спецификации требований.
22. Методы верификации ПО.
23. Временные диаграммы.
24. Тестирование программных продуктов.
25. Управление проектом.
26. Языки программирования.
27. Ведение репозитория проектов.
28. Лицензирование.
29. Стандарты в области программной инженерии.
30. Спиральная модели жизненного цикла.
31. Методы формальной спецификации требований.
32. Формальные методы верификации ПО.
33. Сети Петри.
34. Методы покрытия тестами программ.
35. Параллельные языки программирования.
36. Методы и технологии программирования.

Список рекомендуемой литературы

1. Программная инженерия. Визуальное моделирование программных систем : учебник для вузов / Е. А. Черткова. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2020. — 147 с.
2. Программная инженерия и технологии программирования сложных систем : учебник для вузов / Е. М. Лаврищева. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2020. — 432 с.
3. Технологии и методы программирования : учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2020. — 235 с.
4. Практическое введение в программную инженерию : учебное пособие для вузов / В. К. Волк. — 2-е изд., стер. — Санкт-Петербург : Лань, 2022. — 100 с.