

ВООРУЖЕННЫЕ СИЛЫ РЕСПУБЛИКИ БЕЛАРУСЬ

Военная академия Республики Беларусь

А. В. Жвакина

КОНСТРУИРОВАНИЕ ПРОГРАММ

Курс лекций

*Рекомендовано учебно-методическим объединением
по военному образованию*

В трех частях

Часть 3

**ПРОЦЕСС РАЗРАБОТКИ ПРОЕКТА
И ОСНОВЫ ЯЗЫКА UML**

**Минск
ВА РБ
2017**

УДК 004
ББК 32.97
Ж41

Жвакина, А. В.

Ж41 Конструирование программ: курс лекций: в 3 ч. Ч. 3.
Процесс разработки проекта и основы языка UML /
А. В. Жвакина. – Минск: ВА РБ, 2017. – 94, [1] с.
ISBN 978-985-544-587-7

Издание предназначено для курсантов второго курса специальности 1 – 95 02 05 «Эксплуатация автоматизированных систем обработки информации» учреждения образования «Военная академия Республики Беларусь», изучающих дисциплину «Конструирование программ».

**УДК 004
ББК 32.97**

© УО «Военная академия
Республики Беларусь», 2017

**ISBN 978-985-544-587-7 (ч.3)
ISBN 978-985-544-328-6**

ЛЕКЦИЯ 1. ТРЕБОВАНИЯ К ПРОГРАММНОМУ ПРОДУКТУ

Введение

Жизненный цикл программных средств (ПС) – это совокупность процессов, работ и задач, включающая в себя разработку, эксплуатацию и сопровождение ПС, начиная с анализа его концепции или потребности в заказе до прекращения его использования.

Основные стороны, участвующие в жизненном цикле (ЖЦ) ПС: заказчик, поставщик, разработчик, оператор и персонал сопровождения.

Согласно международному стандарту **ISO/IEC 12207:2008 – Системная и программная инженерия – Процессы жизненного цикла программных средств** и стандарту Республики Беларусь **СТБ ИСО/МЭК 12207-2003 – Информационная технология – Процессы жизненного цикла программных средств**, которые являются базовыми в области жизненного цикла программных средств и систем, выделяют следующие **основные процессы** [3]:

- *заказ* (определение потребностей заказчика в программном продукте (ПП), подготовка и выпуск заявки на подряд, выбор поставщика и управление процессом заказа до завершения приемки);
- *поставка* (подписание договора с заказчиком, определение процедур и ресурсов, необходимых для управления и обеспечения проекта);
- *разработка* (анализ требований, проектирование, программирование, сборка, тестирование, ввод в действие и приемка ПП);
- *эксплуатация* (эксплуатация ПП и поддержка пользователей в процессе эксплуатации);
- *сопровождение* (изменение существующего ПП при сохранении его целостности).

Важнейшая задача при создании программного продукта – это выработка требований или анализ требований к продукту. Заказчик чаще всего представляет весьма размытую идею о том, каким должен быть конечный результат, и не имеет представления о том, как должна работать программа. Незаконченные, нелепые, а иногда противоречащие друг другу требования распознаются хорошими инженерами на этой стадии.

Ошибки, допущенные на стадии сбора требований, составляют от 40 до 60 % всех дефектов проекта. Две наиболее распространенные проблемы, о которых сообщается в большом европейском обзоре индустрии ПО, – определение и управление требованиями заказчика [3].

В классическом техническом подходе совокупность требований используется на стадии проектирования ПО. Требования также используются в процессе проверки ПО, так как тесты основываются на определенных требованиях.

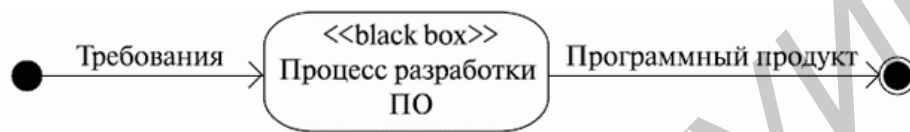


Рис. 1 прекрасно иллюстрирует типичный процесс создания продукта или «решения» (поскольку продукт решает проблему заказчика), как любят говорить в Microsoft.

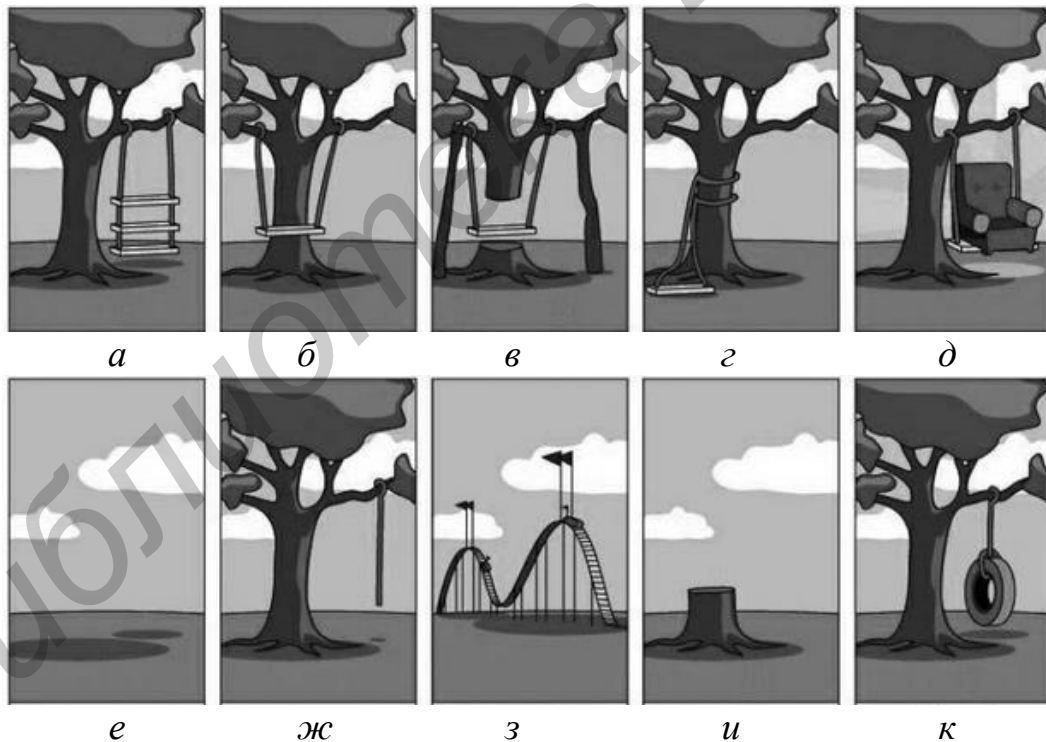


Рис. 1. Результат нечеткой спецификации создаваемого продукта:

а – так объяснил заказчик; *б* – так понял лидер проекта; *в* – так спроектировал аналитик; *г* – так реализовал программист; *д* – так описал бизнес-консультант; *е* – так проект был документирован; *ж* – так продукт был проинсталлирован; *з* – такой счет был предъявлен заказчику; *и* – так осуществлялась техническая поддержка; *к* – а вот чего на самом деле хотелось заказчику

СОДЕРЖАНИЕ

ЛЕКЦИЯ 1. ТРЕБОВАНИЯ К ПРОГРАММНОМУ ПРОДУКТУ ...	3
Введение	3
1. Характеристика требований.....	5
2. Разработка требований	9
3. Спецификация требований.....	11
Заключение	15
ЛЕКЦИЯ 2. ОСНОВНЫЕ ПОНЯТИЯ ЯЗЫКА UML.....	16
Введение	16
1. Особенности графического представления диаграмм языка UML	17
2. Модели и пакеты в языке UML, их графическое изображение.....	20
3. Диаграмма вариантов использования.....	27
4. Диаграмма классов.....	31
Заключение	38
ЛЕКЦИЯ 3. КАНОНИЧЕСКИЕ ДИАГРАММЫ ЯЗЫКА UML.....	39
Введение	39
1. Диаграммы взаимодействия.....	40
2. Диаграмма автомата.....	46
3. Диаграмма деятельности	50
4. Диаграммы реализации	54
Заключение.....	73
ПРИЛОЖЕНИЕ	
Формирование требований к программному продукту	74
Список литературы	93