

Лекция 3

**Характеристика областей
знаний по инженерии
программного обеспечения —
SWEВOK**

Список литературы

1. *Бабенко Л.П., Лаврищева К.М.* Основы програмної інженерії.– Навч. посібник.–К.: Знання, 2001. –269 с.
2. *Лаврищева Е.М., Грищенко В.Н.* Области знаний программной инженерии – SWEBOOK и подход к обучению этой дисциплине// Управляющие системы и машины.–2005. – №1.– С.38–54.
3. *Иоан Коммервил.* Инженерия программного обеспечения. 6-е издание. – М.; Спб. – Киев, 2002. – 623 с.
4. *Лаврищева К.М.* Основні напрямки досліджень в програмній інженерії і шляхи їхнього розвитку // Проблеми програмування. – 2003. – № 3–4. – С. 44–58.
5. *Лаврищева Е.М.* Методы программирования. Теория, инженерия, практика. – К.: Наук. думка, 2006.–450с.
6. *Основы инженерии качества программных систем / Ф.И.Андон, Г.И. Коваль, Т.М. Коротун, Е.М.Лаврищева, В.Ю. Суслов* – К.: Академперіодика.– 2007. – 678 с.
7. *Лаврищева Е.М., Коваль Г.И., Коротун Т.М.* Подход к управлению качеством программных систем обработки данных // Кибернетика и системный анализ.– 2006.–№ 5.–С.174–185.
8. *Кендалл С.* Унифицированный процесс. Основные концепции.–М.;–СПб.– Киев.–2002.– 157с.

Характеристика областей знаний

- рассматривается теоретический и интеллектуальный базис проектирования - методы, принципы, средства и методологии, представленные областями в ядре знаний программной инженерии SWEBOOK.

Характеристика областей знаний

- Ядро знаний SWEBOOK - основной научно-технический документ, отражающий знания и опыт многих иностранных и отечественных специалистов по программной инженерии [1-10] и согласуется с регламентированными процессами ЖЦ стандарта ISO / IEC 12207.

Документ включает в себя описание 10 областей, каждая из которых представлена в соответствии с принятой всеми участниками формирования ядра SWEBOOK общей схемы описания, которая включает в себя определение понятийного аппарата, методов и средств, а также инструментов поддержки инженерной деятельности. Относительно каждой области определен круг знаний, которые должны практически использоваться при выполнении процессов жизненного цикла.

Характеристика областей знаний

- Для представления понятийного аппарата областей знаний SWEBOOK проведем условное разделение областей на главные (пять областей для разработки ПС, рис. 1.7) и вспомогательные организационные области (пять областей, обеспечивающих инженериию управления разработкой ПС, рис.1.8).
В каждой области приведены ключевые понятия, подходы и методы проектирования различных типов ВС. Распределение областей на основные и вспомогательные соответствует структуре распределения процессов стандарту ISO / IEC 12207 (см. раздел 2), выполнение которых определяется методами и средствами, предложенными в ядре знаний SWEBOOK.
Далее дается обзор каждой области этого ядра, определяется ее роль в проектировании и реализации программных продуктов. В некоторых подразделах показана связь с положениями соответствующих стандартов, регламентирующих и регулирующих выполнение процессов проектирования программных систем.

Основные области знаний SWEBOOK

Организационные области знаний SWEBOOK

Инженерия требований

- Требования к ПО - совокупность свойств, которые должно иметь ПО. Предназначены для адекватного определения функций, условий и ограничений выполнения ПО, а также объемов данных, технического обеспечения и среды его выполнения.

Требования отражают потребности людей (заказчиков, пользователей, разработчиков), заинтересованных в создании ПО. Заказчик и разработчик совместно выявляют требования, анализируют, смотрят, определяют необходимые ограничения и условия, а также описывают их. Различают требования к продукту и к процессу, а также функциональные, не функциональные и системные требования.

Инженерия требований

- **Требования к продукту** и к процессу определяют условия выполнения и режимы работы ПО в операционной среде, ограничение на структуру и память компьютеров и принципы взаимодействия программ.
Функциональные требования определяют назначение и функции системы, а не функциональные - условия относительно выполнения ПО, его переносимости и доступа к данным. **Системные требования** описывают требования к программной системе, которая состоит из взаимосвязанных программных и аппаратных подсистем и различных приложений.

Инженерия требований

- Требования могут быть количественные (например, количество обработанных запросов в секунду, средний показатель ошибок и т.п.). Значительная часть требований касается атрибутов качества: безотказность, надежность и др., А также защиты и безопасности как ПО, так и данных.

Область знаний «Требования к ПО (Software Requirements)»

- Область знаний «Требования к ПО (Software Requirements)» состоит из следующих разделов:
 - Инженерия требований (Requirement Engineering),
 - Выявление требований (Requirement Elicitation),
 - Анализ требований (Requirement Analysis),
 - Спецификация требований (Requirement Specification).
 - Валидация требований (Requirement validation),
 - Управление требованиями (Requirement Management).

Основные понятия

- **Инженерия требований к ПО** - это дисциплина анализа и документирования требований к ПО, заключается в преобразовании предложенных заказчиком требований к системе в описание требований к ПО и их валидации.
- Инженерия базируется на модели процесса определения требований и деятельности лиц, обеспечивающих управление и формирование требований, а также на методах достижения показателей качества. **Модель процесса определения требований** - это схема процессов ЖЦ, которые выполняются от начала проекта и до тех пор, пока не будут определены и согласованы требования. Таким процессом может быть маркетинг и проверка выполнения требований в данном проекте. **Управление требованиями к ПО** заключается в контроле за выполнением требований и планировании использования ресурсов (человеческих, программных, технических, временных, стоимостных) в процессе разработки промежуточных рабочих продуктов на процессах ЖЦ и продукта в целом.

Основные понятия

- **Качество и процесс улучшения требований** - это процесс формулирования характеристик и атрибутов качества (надежность, реактивность и др.), которые должно иметь ПО, методы их достижения на процессах ЖЦ и оценки полученных результатов. **Выявление требований** - это процесс извлечения информации из разных источников (договоров, материалов аналитиков по декомпозиции задач и функций системы и др.), проведение технических мероприятий (собеседований, сбор предложений и др.) для формирования отдельных требований к продукту и к процессу разработки. Требования согласовываются с заказчиком.
- **Анализ требований** - процесс изучения потребностей и целей пользователей, классификация и превращение их в требования к системе, аппаратуры и ПО, установки и разрешения конфликтов между требованиями, определение приоритетов, границ системы и принципов взаимодействия со средой функционирования.

Основные понятия

- **Спецификация требований к ПО** - процесс формализованного описания функциональных и нефункциональных требований, требований к характеристикам качества согласно стандарту качества ISO / IEC 9126, которые будут отрабатываться на процессах ЖЦ ПО.
- В спецификации требований отражается структура ПО, требования к функциям, качеству и документации, а также задается архитектура системы и ПО, алгоритмы, логика управления и структура данных. Специфицируются также системные требования, нефункциональные требования и требования к взаимодействию с другими компонентами и платформами (БД, СУБД, и др.).

Основные понятия

- **Валидация требований** - это проверка изложенных в спецификации требований, выполняется для того, чтобы путем отслеживания источников требований убедиться, что они определяют именно данную систему. Заказчик и разработчик ПО проводят экспертизу сформированного варианта требований для того, чтобы разработчик мог дальше продолжать проектирование ПО. Один из методов валидации - **прототипирование**, то есть быстрая отработка отдельных требований на конкретном инструменте и исследование масштабов изменения требований, измерение объема функциональности и стоимости, а также создание моделей оценки зрелости требований.
- **Верификация требований** - это процесс проверки правильности спецификаций требований относительно их соответствия потребностям, непротиворечивости, полноты и возможности реализации, а также согласованности со стандартами. Как результат проверки требований составляется согласованный выходной документ, устанавливающий полноту и корректность требований к ПО, а также возможность продления его проектирования.

Основные понятия

- **Управление требованиями** - это управление процессами формирования требований на всех процессах ЖЦ, а также изменениями и атрибутами требований, проведение мониторинга - восстановления источника требований. Управление изменениями возникает после того, как ПО начинает работать в заданной среде и выявляет ошибки относительно трактовки требований, невыполнения некоторого отдельного требования и т.п.
- Неотъемлемой составляющей процесса управления является **трассирование требований** для отслеживания правильности установления и реализации требований к системе и ПО на процессах ЖЦ, а также обратный процесс **отслеживания** в полученном продукте реализованных требований. Для уточнения некоторых требований или добавления нового требования составляется план изменения требований, согласуется с заказчиком. Внесенные изменения влекут и изменения в созданном продукте или в отдельных его компонентах.

Проектирование ПО

- **Проектирование ПО** - это процесс определения архитектуры, набора компонентов, их интерфейсов, других характеристик системы и конечного состава программного продукта. Область знаний «Проектирование ПО (Software Design)» состоит из следующих разделов:
 - базовые концепции проектирования ПО (Software Design Basic Concepts),
 - ключевые вопросы проектирования ПО (Key Issue in Software Design),
 - структура и архитектура ПО (Software Structure and Architecture),
 - анализ и оценка качества проектирования ПО (Software Design Quality Analysis and Evaluation),
 - нотации проектирования ПО (Software Design Notations),
 - стратегия и методы проектирования ПО (Software Design Strategies and Methods)

Конструирование программного обеспечения

- Конструирование ПО - создание ПО из конструкций (блоков, операторов, функций) и его проверка методами верификации и тестирования. К инструментам конструирования ПО отнесены языки конструирования, программные методы и инструментальные системы (компиляторы, СУБД, генераторы отчетов, системы управления версиями, конфигурацией, тестированием и др.)

Конструирование ПО

- Область знаний «Конструирование ПО (Software Construction)» включает в себя следующие разделы:
 - снижение сложности (Reduction in Complexity),
 - предупреждение отклонений от стиля (Anticipation of Diversity),
 - структуризация проверок (Structuring for Validation),
 - использование стандартов (Use of External Standards).

Снижение сложности - это минимизация, уменьшение и локализация сложности конструирования.

Конструирование ПО

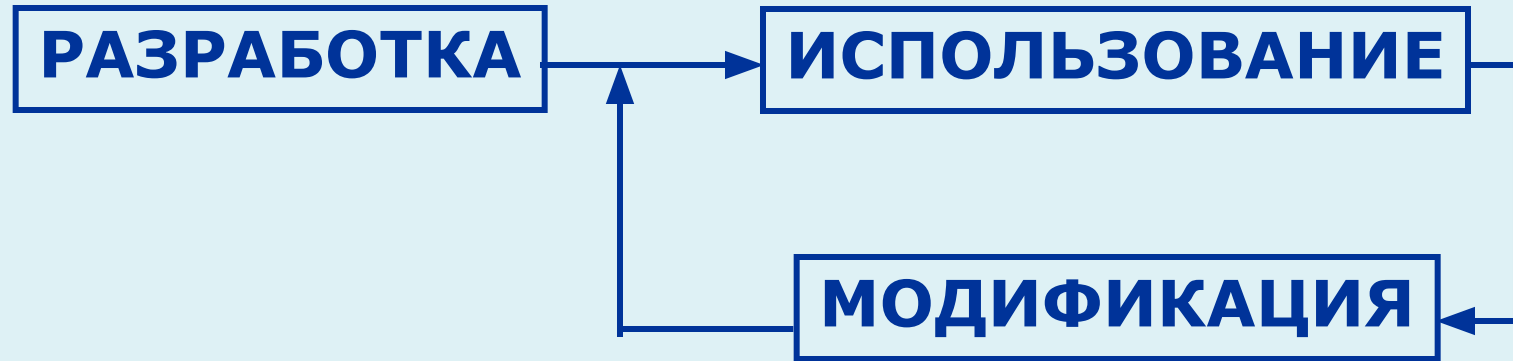
- **Предупреждение отклонений от стиля.** Для решения различных задач конструирования применяются различные стили конструирования (лингвистический, формальный, визуальный).

Лингвистический стиль основан на использовании словесных инструкций и выражений для представления отдельных элементов (конструкций) программ. **Формальный стиль** используется для точного и однозначного определения компонентов системы, минимального количества ошибок, которые могут возникнуть в связи с неоднозначностью определений или неудачных обобщений объектов конструирования ПО.

Визуальный стиль - наиболее универсальный для конструирования прикладного ПО. Он позволяет представлять элемент конструирования в наглядном виде. Визуальный язык проектирования UML предоставляет разработчику набор диаграмм для представления статической и динамической структур ПО. При его применении создается текстовое и диаграммное описание конструктивных элементов ПО, которое выводится на экран дисплея для просмотра и корректировки.

Конструирование ПО

- **Структуризация проверок** предполагает, что построение ПС структурировано таким образом, что упрощается поиск ошибок, дефектов и различных сбоев в процессе проверок как на стадии независимого тестирования, так и в процессе эксплуатации. Структуризации проверок способствуют характеристики, инспектирование, передача, модульное тестирование с применением автоматизированных средств тестирования и др..
- **Использование внешних стандартов.** Конструирование ПО зависит от применимых внешних стандартов, связанных с языками программирования, инструментальными средствами и интерфейсами. При конструировании должен быть определен достаточный набор стандартов для управления и обеспечения координации между определенными видами деятельности и группами операций, минимизации сложности, внесения изменений, анализа рисков и т.п.. К таким стандартам относятся: языки программирования (Java, C++ и др.)



Подходы к проектированию ПО –

- 1) Строго последовательное выполнение всех этапов жизненного цикла ПО (модель водопада);**
- 2) Поэтапное создание ПО (пошаговая модель);**
- 3) Метод создания прототипа разрабатываемого ПО;**
- 4) CASE-технологии (CASE – Computer-Aided Software Engineering)**

Примеры крупномасштабных систем ПО:

- Распределенная банковская система;
- Операционная система;
- Компьютерная игра;
- Система контроля и безопасности полетов...

Особенности разработки – усилия многих людей на протяжении длительного времени.

Технология разработки ПО включает основные принципы (жизненный цикл ПО, модульность, шаблоны проектирования), а также средства и методы разработки ПО.

Нисходящая методология

Сложная задача сводится к набору более простых задач, решение которых приведет к выполнению исходной задачи. При этом образуется иерархическая система последовательных уточнений, которая отображается в модульную структуру, совместимую с императивной парадигмой программирования

Восходящая методология

Определяются отдельные задачи внутри системы, затем изучается как решение этих задач. Может использоваться в качестве абстрактных инструментов для решения более общих задач. При этом иерархии нет, а модули могут быть равноправными. При этом сложная система ПО строится из стандартных, свободно продаваемых компонентов.

Тестирование ПО - это процесс проверки готовой программы в статике (просмотры, инспекции, налаживание исходного кода) и в динамике (прогон на наборе тестовых данных) с целью проверки различных путей выполнения программы и сравнение полученных результатов с заранее заданными.

Тестирование по принципу «прозрачного ящика»

На основе принципа

Парето
Тестирование основных путей

Тестирование по принципу «черного ящика»

На основе анализа граничных условий

На основе применения избыточности

Метод упрощенных версий