КОНСПЕКТ ЛЕКЦИЙ

|  |  |
| --- | --- |
| *по дисциплине* | Проектирование человеко-машинного интерфейса |
|  | (наименование дисциплины) |

Тема лекционных занятий: 2-3.

Бизнес-анализ и обследование объекта автоматизации. Требования к системе

***Литература:***

1. IBM(R) Rational Unified Process(R) [Электронный ресурс]: Rational Method Composer - URL: <http://dit.isuct.ru/Publish_RUP/>
2. Артефакт: Видение [Электронный ресурс]: Rational Method Composer - URL:<http://dit.isuct.ru/Publish_RUP/#core.base_rup/workproducts/rup_vision_2D6D6F1.html>
3. Вигерс, Карл И. Разработка требований к программному обеспечению. Практические приемы сбора требований и управления ими при разработке программного продукта / Карл И. Вигерс - М.: ИТД “Русская Редакция”, 2004

***Краткое содержание лекций по теме:***

1. *Методологии разработки ПО*
2. *Методологии проектирования*
3. *RUP*
4. *Разработка документа «Видение»*
5. *Требования к системе*

### Методологии разработки ПО

* Каскадная (водопадная) модель (W.W. Royce, [1970 г](http://ru.wikipedia.org/wiki/1970_%D0%B3%D0%BE%D0%B4).)
* Инкрементальная модель
* Спиральная модель (Б. Боэм, [1988 г.](http://ru.wikipedia.org/wiki/1988_%D0%B3%D0%BE%D0%B4))

### Методологии проектирования

Методология проектирования ИТ-систем – набор стандартизованных и апробированных действий, которые позволяют достичь запланированных функциональностей ИТ-систем средствами имеющихся технологий с учетом заданных ограничений. Применение методологии гарантирует упорядоченный подход к промышленной разработке, использованию и сопровождению ИТ-систем, т.е. вносят в процесс создания ПО инженерный подход.

Набор методологий, используемых при проектировании ИТ-систем, достаточно широк. Наиболее распространенные методологии представлены в нижеследующем списке:

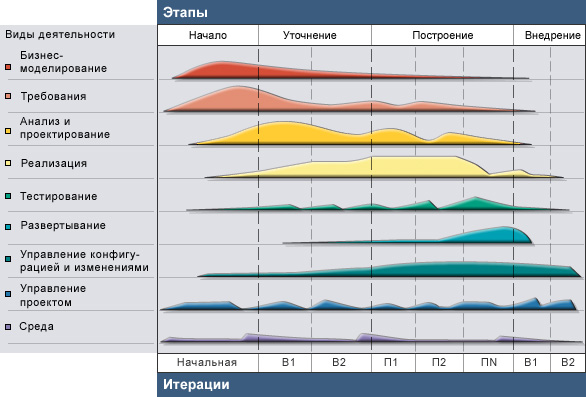
* Как получится (code&fix)
* Cleanroom Software Engineering
* [Итеративная](http://ru.wikipedia.org/wiki/%D0%98%D1%82%D0%B5%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%B0%D1%8F_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0)
* [RUP](http://ru.wikipedia.org/wiki/Rational_Unified_Process)
* OpenUP
* [MSF](http://ru.wikipedia.org/wiki/Microsoft_Solutions_Framework)
* [RAD](http://ru.wikipedia.org/wiki/RAD_%28%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%29)
* [Agile](http://ru.wikipedia.org/wiki/%D0%93%D0%B8%D0%B1%D0%BA%D0%B0%D1%8F_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%8F_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8)
* [Agile Modeling](http://en.wikipedia.org/wiki/Agile_Modeling)
* [Agile Unified Process](http://en.wikipedia.org/wiki/Agile_Unified_Process) (AUP)
* [Agile Data Method](http://en.wikipedia.org/wiki/Agile_Data_Method)
* [DSDM](http://en.wikipedia.org/wiki/DSDM)
* [Essential Unified Process](http://en.wikipedia.org/wiki/Essential_Unified_Process) (EssUP)
* Extreme programming, XP
* [Feature Driven Development](http://en.wikipedia.org/wiki/Feature_Driven_Development) (FDD)
* [Getting Real](http://ru.wikipedia.org/wiki/37signals#.22Getting_Real.22_philosophy)
* [Open Unified Process](http://ru.wikipedia.org/wiki/OpenUP) (OpenUP)
* [Scrum](http://ru.wikipedia.org/wiki/Scrum)
* [Бережливая разработка программного обеспечения](http://ru.wikipedia.org/wiki/%D0%91%D0%B5%D1%80%D0%B5%D0%B6%D0%BB%D0%B8%D0%B2%D0%B0%D1%8F_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F) (Lean Software Development)
* КанБан
* FDD

### RUP

В качестве примеров реализации итеративного подхода ниже рассматривается [методология разработки программного обеспечения, созданные компанией Rational Software](http://ru.wikipedia.org/wiki/RUP) – Rational Unified Process (RUP).

RUP – это методология создания программного обеспечения, оформленная в виде размещаемой на Web базы знаний, которая снабжена поисковой системой [1]. RUP использует итеративную модель разработки. Полный жизненный цикл разработки продукта состоит из четырех фаз, каждая из которых включает в себя одну или несколько итераций:

* Начало (inception)
* Уточнение, проектирование (elaboration)
* Построение, конструирование (construction)
* Внедрение (transition)



В матрице RUP итерации (время) расположены по вертикали, а по горизонтали – рабочие процессы (workflows), которые разделены на рабочие процессы процесса (core process workflows)

* бизнес-моделирование
* требования
* анализ и проектирование
* реализация (implementation – выполнение)
* тестирование
* развертывание (deployment) и рабочие процессы поддержки (core supporting workflows)
* управление конфигурациями
* управление проектом
* взаимодействие с окружением

**Разработка документа «Видение»**

Документ «Видение» является артефактом бизнес-моделирования и создается на этапе Начала. Начало (inception) подразумевает предварительное взаимодействие с заинтересованными лицами (заказчик, пользователи, инвесторы и др. stakeholders). На этом этапе формируются видение и границы проекта, создается экономическое обоснование, определяются основные требования, ограничения и ключевая функциональность продукта, создается базовая версия [модели прецедентов](http://ru.wikipedia.org/w/index.php?title=%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%BF%D1%80%D0%B5%D1%86%D0%B5%D0%B4%D0%B5%D0%BD%D1%82%D0%BE%D0%B2&action=edit&redlink=1), оцениваются риски [2].

Какие навыки прокачиваются-используются на этом этапе:

* Использование подходов и методологий:
  + мозговой штурм
  + 5WHY
  + SWOT
  + черный ящик
  + функциональное моделирование (IDEF0)
  + контекстная диаграмма
  + майндкарта
* Работа с документами:
  + Изучение
  + Деловая переписка
  + Ведение документов
* Коммуникативные навыки [3, см. главы 6-7]:
  + Фасилитатор - направляет разговор
  + Медиатор - уравновешивает противоположные мнения до решения, которое будет выгодно всем
  + Интервьюирование – точные вопросы (да\нет, выбор одного из вариантов ответа, приоритизация)
* Работа с требованиями [2], [3, см. глава 5, приложение Г Документ об образе и границах проекта]:
  + Целеполагание (позиционирование) – необходима правильная постановка проблемы (задачи). Заказчик (ребенок) хочет Луну. Вы строите космический корабль и отправляете человека на Луну. Однако Заказчик плачет - ребенок хотел игрушку. В том числе необходимо определить, в чем заключается конкурентоспособность разрабатываемого продукта
  + Выявление заинтересованных лиц – знакомство и сбор контактных данных. Определение степени влияния на продукт каждого заинтересованного лица, их потребности и степень заинтересованности.
  + Определение границ системы – определение зон ответственности (и где эта зона заканчивается), выявление интерфейсов аппаратного, программного, пользовательского взаимодействий, фильтрация требований в связи с установленными границами системы. Границы системы могут быть представлены контекстной диаграммой.   
    
  + Выявление требований

### Требования к системе

Требование – это условие или возможность, которой должна соответствовать система [1].

Требование – это: 1. условия или возможности, необходимые пользователю для решения проблем или достижения целей; 2. условия или возможности, которыми должна обладать система или системные компоненты, чтобы выполнить контракт или удовлетворять стандартам, спецификациям или другим формальным документам; 3. документированное представление условий или возможностей для пунктов 1 и 2. (IEEE Standard Glossary of Software Engineering Terminology).

Типы требований [3, см. глава 1]:

1. Функциональные
   1. Бизнес-требования
   2. Пользовательские требования
   3. Функциональные требования
   4. Системные требования
2. Нефункциональные
   1. Бизнес-правила
   2. Атрибуты качества
   3. Внешний интерфейс
   4. Ограничения

*Нормативные документы по работе с требованиями:*

Разработки IEEE:

* IEEE 1362 “Concept of Operations Document”.
* IEEE 1233 «Guide for Developing System Requirements Specifications».
* IEEE Standard 830-1998, «IEEE Recommended Practice for Software Requirements Specifications»
* IEEE Standard Glossary of Software Engineering Terminology/IEEE Std 610.12-1990
* IEEE Guide to the Software Engineering Body of Knowledge (1) – SWEBOK®.

Отечественные ГОСТ:

* ГОСТ 34.601-90. Информационная технология. Автоматизированные системы. Стадии создания.
* ГОСТ 34.602-89. Информационная технология. Техническое задание на создание автоматизированной системы
* ГОСТ 19.201-78. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению.

Книги

* Вигерс К. Разработка требований к программному обеспечению/Пер, с англ. — М.:Издательско-торговый дом «Русская Редакция», 2004. —576с.: ил.
* Леффингуелл Д., Уидриг Д. Принципы работы с требованиями к программному обеспечению. М.: ИД “Вильямс”, 2002.
* Коберн А. Современные методы описания функциональных требований к системам. М.: издательство «Лори», 2002. – 263 с.
* Мацяшек Л. Анализ требований и проектирование систем. Разработка информационных :с Диалектика-Вильямс
* Орлик С., Булуй Ю. Введение в программную инженерию и управление жизненным циклом ПО Программная инженерия. Программные требования

Разработал:

Яцык А.А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.