

|  |
| --- |
| Development process  BookIt |
|  |

| Related Artifacts | |
| --- | --- |
| Ref. | Name |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Abbreviations and Acronyms | |
|  |  |
|  |  |

Contents

[1 Section 1 **Error! Bookmark not defined.**](#_Toc419294329)

[1.1 Section 2 5](#_Toc419294330)

[1.1.1 Section 3 5](#_Toc419294331)

[1.1.1.1 Section 4 5](#_Toc419294332)

[Appendix A. Example of 1st lvl appendix **Error! Bookmark not defined.**](#_Toc419294333)

[Appendix A.1. example of 2nd lvl appendix 7](#_Toc419294334)

# методология процесса разработки

Для управления процессами разработки используется методология Feature Driven Development (FDD).

FDD – это итеративная методология разработки программного обеспечения, которая принимает за основу важную для заказчика функциональность разрабатываемого программного обеспечения. Основной целью методологии является разработка реального, работающего программного обеспечения в поставленные сроки.

FDD построена на основе лучших практик в разработке программного обеспечения. В FDD используется:

* Объектное моделирование предметной области.
* Разработка по функциям – любая функция, которая оценивается больше 2х недель, разбивается на подфункции.
* Индивидуальное владение классом(кодом) – над каждым участком кода одновременно работает только один разработчик, который отвечает за его согласованность с остальной системой и производительность.
* Команда по проектированию интерфейсов и функций – небольшая, динамически формируемая команда разработчиков, занимающаяся небольшой подзадачей. Позволяет нескольким разработчикам участвовать в разработке контрактов функций и повышает общий уровень знаний разработчиков о работе всей системы.
* Проверка кода – обеспечивают хорошее качество кода.
* Конфигурационное управление – помогает определять ответственность классов за функционал и поддерживать историю изменения кода при добавлении и изменении функционала.
* Регулярные сборки (билды) – гарантирует наличие актуальной версии системы для показа заказчику.
* Прозрачность процессов и планов разработки для заказчика.

Основные этапы разработки программного обеспечения по FDD:

1. Разработка архитектуры приложения.
2. Составление списка функций/задач.
3. Составление плана разработки функций – устанавливается приоритет, зависимости между задачами.
4. Проектирование функций/задач.
   1. Анализ.
   2. Дизайн.
   3. Проверка дизайна.
5. Реализация функций/задач.
   1. Написание кода.
   2. Проверка кода.

5.3. Включение в сборку.



Рисунок 1 - Основные этапы разработки Программного обеспечения по методологии FDD

## Разработка архитектуры системы

Разработка архитектуры системы проводится под руководством координатора и ведущего разработчика проекта.

Разработка архитектуры системы состоит из следующих этапов.

1. Формирование команды разработчиков, отвечающих за архитектуру системы. В команду должны входить постоянные участники данного проекта по разработке системы, координатор и ведущий разработчик проекта.
2. Обзор разрабатываемой системы и ее предметной области. Проводится координатором и ведущим разработчиком проекта.
3. Изучение документации. Команда изучает документы с требованиями.
4. Команда разбивается на небольшие группы не больше 3х человек.
5. Каждая группа предоставляет свой вариант архитектуры системы или подсистемы. Координатор и ведущий разработчик также предоставляют свои варианты. Выбирается наиболее подходящий вариант архитектуры системы.
6. Улучшение архитектуры. Повторяются шаги 5 и 6.
7. Документирование выбранной архитектуры, возможного развития архитектуры, альтернативных решений.
8. Проверка. Внутренняя оценка архитектуры проводится координатором проекта и ведущим разработчиком. Архитектура предоставляется заказчику на проверку соответствия будущей системы его ожиданиям. При необходимости пункты с 5 по 8 повторяются.

Архитектура считается удовлетворительной, если ее одобрили координатор, ведущий разработчик и заказчик.



Рисунок 2 – Разработка архитектуры системы

Результатом разработки архитектуры является:

1. Структура системы – основные модули и связи между ними. Контракты модулей.
2. Диаграмма основных классов бизнес-логики.
3. Диаграммы последовательностей.
4. Документация по выбранной архитектуре и альтернативным вариантам с объяснением выбора.

This is the body copy style. Always use Trebuchet MS, size 10. Style is Body (not Normal). Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Table 1 Table name

| Table heading | |
| --- | --- |
| Column name1 | Column name2 |
| Table text  Example of table number list:   1. This is the first item in a list. 2. This is the second item in a list. | Table text  Example of table bullet list:   * This is the first item in a list; * This is the second item in a list. |

## Section 2

Body text



Figure 1 Figure name

### Section 3

Body text

#### Section 4

Body text

1. This is the first item in a 1st lvl list.
2. This is the second item in a 1st lvl list.
   1. This is the first item in a 2nd lvl list.
   2. This is the second item in a 2nd lvl list.
      1. This is the first item in a 3rd lvl list.
      2. This is the second item in a 3rd lvl list.
3. This is the third item in a 1st lvl list.
4. This is the fourth item in a 1st lvl list.

[Hyperlink](http://elements.epam.com/)

* This is the bullet style;
* Secondary bullet;
* Tertiary bullet.

1. This is the style of Note – a modified numbering style linked to NoteStyle paragraph, with Hanging but no Indent.
2. This is the style of Warning that will look exactly like Note Style except for the caption

Code example. Courier new.

1. Example of 1st lvl appendix

This is Appendix A.

* 1. example of 2nd lvl appendix

This is Appendix A.1.

| REVISION HISTORY | | | | | |
| --- | --- | --- | --- | --- | --- |
| Ver. | Description of Change | Author | Date | Approved | |
| Name | Effective Date |
| n.n |  |  | dd-mmm-yyyy |  | dd-mmm-yyyy |
|  |  |  |  |  |  |
|  |  |  |  |  |  |