**Стандарти розробки**

Проект UKRENERGO\_I

Система Microsoft Dynamics AX 2012

**ЗАТВЕРДЖЕНО:**

|  |  |
| --- | --- |
| **Від ДП «НЕК «УКРЕНЕРГО»**  Керівник проектів та програм у сфері нематеріального виробництва  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Кравченко Ю.В.** | **Від ТОВ «СМАРТ БІЗНЕС»**  Керівник проекту  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Гривняк А.А.** |

Зміст

[1. Розуміння вимог 3](#_Toc521494245)

[1.1. Функціональний дизайн 3](#_Toc521494246)

[1.2. Опис ролей. 4](#_Toc521494247)

[1.3. Підготовка функціонального дизайну: 4](#_Toc521494248)

[2. Вимоги щодо розробки 6](#_Toc521494249)

[2.1. Створення об'єктів 6](#_Toc521494250)

[2.2. Таблиці 6](#_Toc521494251)

[2.3. Класи 6](#_Toc521494252)

[2.4. Форми 6](#_Toc521494253)

[2.5. Використання міток 7](#_Toc521494254)

[2.6. Коментарі 7](#_Toc521494255)

[2.7. Кращі практики та рекомендації 8](#_Toc521494256)

[2.8. Базові перерахування 8](#_Toc521494257)

[2.9. Контроль версій 9](#_Toc521494258)

[2.10. Префікс об'єктів 9](#_Toc521494259)

[2.11. Модель та прошарок розробки 9](#_Toc521494260)

[2.13. Кінець робочого дня 10](#_Toc521494261)

[3. Вимоги щодо тестування 10](#_Toc521494262)

[Додадок 1 12](#_Toc521494263)

[Функціональний дизайн 12](#_Toc521494264)

[Додадок 2 15](#_Toc521494265)

[Тестовий звіт на розробку або модифікацію системи AX 15](#_Toc521494266)

[3.2. Тестування змінених об'єктів 16](#_Toc521494267)

[3.3. Тестування зміненої функціональності, проходження тест-кейсів 16](#_Toc521494268)

[3.3.1. Тест-кейс 1 16](#_Toc521494269)

[3.3.2. Тест-кейс 2 16](#_Toc521494270)

[3.3.3. Тест-кейс Х 17](#_Toc521494271)

[Використана література: 18](#_Toc521494272)

Стандарти розробки - вимоги, якими керується компанія СМАРТ Бізнес (надалі – Виконавець) і компанія ДП «НЕК «УКРЕНЕРГО» (надалі – Замовник) під час робіт по розробці (створенні нових сутностей) та модифікації (внесення змін в існуючі сутності) системи Microsoft Dynamics AX 2012 (далі – Система) в рамках проекту UKRENERGO\_I. Даний документ може бути доповнений стандартами розробки сторонніх розробників у разі появи таких, проте не може бути змінений в тих місцях, які безпосередньо стосуються Виконавця.

Документ створено відповідно до вимог:

* Договору № 06-6/3755-17 від «26» грудня 2017 р.;
* Статут проекту з впровадження системи – ERP;
* Календарного план-графіку проекту.

1. **Розуміння вимог**

## **Функціональний дизайн**

Функціональний дизайн – документ, відповідно до якого виконуються розробки та модифікації в системі (надалі – ФД) і який складається з таких пунктів:

1. Введення – Обґрунтування ціль і змін
2. Список використовуваних термінів
3. Опис поточного функціоналу системи
4. Опис вимог зазначених та затверджених в документах: Functional Requirement Document (Документ функціональних вимог) та Fit Gap Analysis document (Документ Аналіз покриття).
5. Прийняті припущення
6. Створення нових об'єктів в рамках розробки
7. Опис змін
8. Тестовий сценарій

ФД пишеться консультантом на основі затверджених документів Functional Requirement Document (Документ функціональних вимог) та Fit Gap Analysis document (Документ Аналіз покриття) та відповідає їм і являє собою документ, який описує модифікацію існуючої функціональності або створення нової.

До початку дій із розробки необхідно ясно розуміти вимоги, які повинні міститись в ФД для модифікації. Додаткова інформація може бути отримана з інших документів, написаних під час етапів аналізу та побудови прототипу системи.

Шаблон документу «Функціональний дизайн» додається та передбачає собою набір блоків, які використовуються для створення документа, проте потреба у використанні кожного з них визначається консультантом під час створення документа та залежить від конкретної задачі.

## **Опис ролей.**

Консультант – представник Замовника або Виконавця, який відповідає за функціональний опис змін в Системі.

Архітектор Замовника – представник Замовника, який відповідає за комплексу та цілісну роботу Системи відповідно до бізнес-вимог.

Архітектор функціонального рішення – представник Виконавця, який визначає методи та шляхи реалізації бізнес-вимог замовника в системі, забезпечує цілісність роботи системи в цілому.

Провідний розробнику проекту – представник Виконавця, який визначає технічну реалізацію бізнес-вимог в Системі.

Розробник – представник Замовника або Виконавця, які виконують зміни в роботі Системи.

Тестувальник – представник Замовника або Виконавця, який виконує тестування коректності виконаних змін в Системі.

## **Підготовка функціонального дизайну:**

* Консультант на підставі документу «Аналіз покриття» (Fit&Gap Analysis) створює ФД.
* Архітектор функціонального рішення зі сторони Виконавця оцінює правильність ФД з точки зору функціональності. Якщо він вважає, що ФД некоректно описує потрібну функціональність або цю функцію можна реалізувати по-іншому, стандартними засобами системи, то ФД відправляється на доопрацювання консультанту – автору ФД.
* На розгляд архітектору замовника ФД надається в форматі word;
* За результатом розгляду пропозиції, зауваження та доповнення відображаються в документі у форматі правок або коментарів;
* Архітектор Замовника розглядає та погоджує ФД в частині вимог, які зазначені та затверджені в документах: Functional Requirement Document (Документ функціональних вимог) та Fit Gap Analysis document (Документ Аналіз покриття).
* Підготовка, розгляд та погодження ФД, зі сторони Замовника та Виконавця мають відповідати термінам, які визначені в календарному план-графіку проекту. Сторони застосовують правила зміни термінів у відповідності до Статуту проекту
* Після погодження архітекторами, ФД відправляється на розгляд провідному розробнику проекту.
* Провідний розробник оцінює правильність оформлення ФД, можливість технічної реалізації модифікацій, складність і орієнтовний час. Якщо у провідного розробника є зауваження до ФД, то він відправляє ФД на доопрацювання консультанту – автору ФД. Провідний розробник має право змінити інтерфейс форм і звітів, алгоритми заповнення полів, а також може вимагати перегляду і зміни ФД, якщо він вважає, що:
  + - описана модифікація не є коректною з технічної точки зору (порушує цілісність системи та\або порушує практики розробки, описані в пункті [2.7](#_Кращі_практики_та) цього документу);
    - модифікація може зайняти неприпустимо довгий час через складності реалізації;
    - запропоновані алгоритми є некоректними або неефективними, уточнюючи причину свого рішення та пропонуючи альтернативні варіанти вирішення задачі.
* ФД, погоджений архітектором функціонального рішення, архітектором замовника та провідним розробником, передається розробнику.
* Якщо у розробника у процесі роботи над ФД виникають питання функціонального характеру (тобто такі, які описують поведінку системи з точки зору бізнес процесів), то він має право задавати їх консультанту – автору ФД, або архітектору функціонального рішення. Питання технічного характеру (тобто такі, які передбачають збереження цілісності системи та дотримання практик розробки), повинні направлятися провідному розробнику.
* Якщо в процесі реалізації ФД консультант або розробник з'ясували, що необхідно змінити ФД, то консультант повинен змінити ФД. При цьому номер версії ФД збільшується на одиницю. Нові версії ФД обов'язково повинні зберігатись. Якщо у ФД були внесені незначні зміни, які не впливають на загальний процес розробки (складність реалізації, час розробки, реалізація основних алгоритмів), то дозволяється не змінювати номер версії ФД. Нова версія ФД або внесенні зміни в поточну версію ФД обов’язково мають бути погоджені архітектором замовника, архітектором функціонального рішення та провідним розробником.
* Розробка/модифікація не може бути виконана без погодженого ФД архітектором замовника, архітектором функціонального рішення.

1. **Вимоги щодо розробки**

Розробка – процес виконання модифікації та розробки в Системі.

### **2.1. Створення об'єктів**

Об’єкт – це ключове поняття об'єктно-орієнтованих технологій проектування та програмування, тобто є абстрактним описом сутностей в системі.

Всю інформацію стосовно нижченаведених термінів та об’єктів знаходиться за посиланнями:

<https://msdn.microsoft.com/en-us/library/aa843974.aspx>  
<https://msdn.microsoft.com/en-us/library/hh881815.aspx>

### **2.2. Таблиці**

Таблиці – це сутності, які містять в собі групу пов'язаних полів. Поля містять в собі індивідуальні частини даних2.

Всі поля, в яких виводяться коди об'єктів Системи, повинні допускати операцію "Перехід до основної таблиці".

Назви тимчасових таблиць (таблиці, в яких дані не зберігаються, а існують лише в рамках буферу окремої користувальницької сесії1) повинні містити префікс «Tmp»

Назва індексу (фізичний показник в системі, який прискорює пошук серед рядків на таблицях в базі даних2) повинна містити суфікс Idx

Всі додаткові поля, групи полів, методи та інші сутності, створені в стандартних таблицях (тут і надалі під стандартними маються на увазі такі об’єкти, які були створені в системі поза рамками цього проекту на прошарку розробки, **isv** або нижче, для забезпечення працездатності системи), мають містити префікс проекту **UKN.**

### **2.3. Класи**

Всі додаткові поля та методи, створені в стандартних класах (сутності, які містять код, що використовується в системі2) а також нові класи мають містити префікс проекту **UKN.**

Інтерфейси (спеціальний тип абстрактних класів, який визначає загальну структуру для поведінки класів, які реалізують методи цього інтерфейсу1) повинні закінчуватися приставкою «able». Наприклад : BatchRunable, Batchable.

### **2.4. Форми**

Всі додаткові поля, методи, групи полів та інші сутності, створені в стандартних формах (сутності, які визначають влаштування елементів на користувальницькому інтерфейсі та використовуються для перегляду даних в системі2), а також нові форми мають містити префікс проекту **UKN.**

### **2.5. Використання міток**

Рекомендації по використанню міток (сутності, які використовуються для забезпечення текстовими представленнями елементів, які кінцеві користувачі бачать в системі2):

* Весь текст, який буде відображатися в інтерфейсі, необхідно представляти мітками
* Мітка в коді має бути обгорнута подвійними лапками (")
* При імпорті методів необхідно перевіряти мітки, які були використані в цьому методі.
* Повідомлення і написи Системи не повинні містити граматичних помилок (на будь-якій мові).
* Назва файлу міток повинна бути описана у документації до проекту. (Визначається до початку розробки)

Для даного проекту використовується файл міток «**UKN**» з двома мовами: **ru** та **en-us**.

### **2.6. Коментарі**

Рекомендації по використанню коментарів:

* Коментар використовується для додаткових пояснень створеного фрагменту коду
* Коментар повинен доповнювати код, а не перефразувати його.
* Не потрібно коментувати очевидні речі
* Коментар до фрагмента коду, потрібно писати з тим самим відступом, що і у самого фрагмента
* Коментар повинен бути коротким і точним

Первісний код повинен зберігатися у вигляді закоментованого коду. Модифікуючи існуючий об'єкти, необхідно модифіковані рядки обгортати блоком коментарів:

*// \_compName PROJ \_projName\_ BEGIN \_taskId \_Description: by \_userId on \_dateTime*

TreeNode projectTreeNode;

TreeNodeIterator projectIterator;

*// \_compName PROJ \_projName\_ END \_taskId \_Description: by \_userId on \_dateTime*

Якщо змінено один рядок, то після нього потрібно вставляти однорядковий коментар:

ProjectListNode projectListNode *// \_compName PROJ \_projName\_ \_taskId \_Description: by \_userId on \_dateTime*

Якщо в існуючий об'єкт додається метод, тоді необхідно заповнити xml документацію, а також додати однорядковий коментар:

*/// <summary>*

*/// Indicates, if name is general*

*/// </summary>*

*/// <param name="name">*

*/// Name for check*

*/// </param>*

*/// <returns>*

*/// True, if name is general, otherwise – false*

*/// </returns>*

*/// <remarks>*

*/// No additional information required*

*/// </remarks>*

*// \_compName PROJ \_projName\_ \_taskId \_Description: by \_userId on \_dateTime*

**boolean** isGeneral(**str** \_name)

{

**return** **false**;

}

При створенні нового об'єкта, до методу classDeclaration також необхідно додавати однорядковий коментар та xml документацію. При створення нових методів у новому класі в рамках однієї модифікації однорядковий коментар додавати не потрібно.

Змінні, які використовуються в коментарях:

* **\_compName** – назва (ідентифікатор) компанії-виконавця модифікації. Для Виконавця, \_compName = **SB**
* **\_projName** – назва (ідентифікатор) проекту, в рамках якого відбувається модифікація. Для цього проекту, \_projName = **UKRENERGO\_I**
* **\_taskId** – номер задачі системи стеження. Для Виконавця, системою стеження є SMART Tracker.
* **\_Description** – опис задачі системи стеження (мова може відрізнятися від опису задачі в системі стеження, але має бути затвердженою проектною документацією)
* **\_userId** – ідентифікатор користувача, який проводив модифікацію (з системи АХ).
* **\_dateTime** – дата та час проведення модифікації.

### **2.7. Кращі практики та рекомендації**

Кращі практики для розробників додатків задокументовані Microsoft за посиланнями:

<https://technet.microsoft.com/en-us/library/jj687763.aspx>  
<https://msdn.microsoft.com/en-us/library/aa658028.aspx>

та у Microsoft Dynamics® AX 2012 Best practices for developing customizations3.

Необхідно, щоб всі нові об'єкти по закінченню модифікації не містили порушень Вest Рractices (повідомлення про порушення рекомендацій). Виключення ВР необхідно документувати.

### **2.8. Базові перерахування**

Додаючи новий елемент в стандартні перерахування (списки літералів, які використовуютьсяв середовищі розробки2) необхідно вказати числове значення елемента, яке значно перевищує найбільше числове значення стандартних елементів.

Наприклад для перерахування AccountRole, елемент «Opening» має найбільшу цифрове значення, яке дорівнює 4. Додаючи новий елемент для нього, необхідно вказати значення 104.



## 

## **2.9. Контроль версій**

Використання контролю версій дозволяє швидко і зручно переглядати історію об'єктів та в разі необхідності завантажувати минулі версії об'єктів. Система контроля версій, яка застосовується для даного проекту – MorphX SVC, з параметрами відхилення помилок компілятора та рекомендацій, з виключеннями на такі елементи як проекти та роботи. Виконавець працює в проекті **SB\_UKN\_SVC**.

Всі створені/змінені об'єкти повинні бути збережені в контролі версій відразу після створення/внесення перших змін. Після завершення розробки, об'єкт обов'язково повинен бути доданий в контроль версій. При цьому, коментар повинен мати номер модифікації та/або задачі (визначається системою стеження), для якої даний об'єкт було створено/змінено.

Кожен день сповіщення на пошту буде повідомляти про модифіковані об'єкти які не були додані в систему контроля версій.

## **2.10. Префікс** **об'єктів**

Усі нові об'єкти, створені в АОТ (Application Objects Tree – Дерево об’єктів системи, що містить всі визначення елементів, які використовуються для побудови системи та забезпечує візуальне представлення цих елементів2), мають містити префікс. Префікси використовуються для уникнення ситуації дублювання імен об'єктів з різних проектів.

Нові об'єкти повинні мати префікс з трьох букв, які були вказані у файлі міток.

## **2.11. Модель та прошарок розробки**

Модель та прошарок розробки – це сутності, що вміщають в собі елементи, які представлені в АОТ.

Всі модифікації в рамках даного проекту виконуються на прошарку **CUS** (Customer layer – включає в себе конкретну для клієнта функціональність2). Виконавець працює в моделі **[SB\_DEV\_UKN].**

**2.12. Проекти в АОТ**

Кожний об’єкт, створений або модифікований в рамках модифікації додається до нового проекту в АОТ, прив’язаний до цієї модифікації. Маска коду проекту має відповідати наступним вимогам:

UKN\_taskId\_Description, де:

taskId – описаний в пункті 2.6 цього документу.

Description має наступний вигляд:

XX\_YY\_Text, де XX – номер функціональної області бізнес-процесу із документу Аналіз покриття (Fit&Gap Analysis);

YY – номер бізнес-вимоги в межах функціональної області бізнес-процесу із документу Аналіз покриття (Fit&Gap Analysis). Один проект не може містити реалізацію декількох бізнес-вимог одночасно.

Text – додатковий текст, який коротко описує суть бізнес-вимоги

## **2.13. Кінець робочого дня**

По закінченню робочого дня, всі об'єкти повинні бути справні і не містити помилок компіляції (система не має видавати жодної помилки при компіляції проведених модифікацій та при повній компіляції). Також рекомендується зберігати всі зміни в системі контроля версій.

1. **Вимоги щодо тестування**

Тестування проводиться окремо виділеним тестувальником або консультантом. Основою для тестування є ФД. Тестування проводиться за заздалегідь складеним сценарієм тестування. Сценарій тестування являє собою певну послідовність дій в системі, яка виконується з використанням набору тестових даних. Ця послідовність дій повинна найбільш повно перевіряти виконані модифікації з урахуванням можливих помилкових ситуацій. Також, слід звертати увагу на роботу стандартної функціональності системи, щоб виявити ситуації, в яких ця функціональність була ненавмисно модифікована. Тестування вважається успішно пройденим, якщо отримані наступні результати:

1. В результаті введення тестових даних були отримані вірні вихідні результати (запити, звіти).
2. Система реагує на дії тестувальника так, як це було описано в ФД.
3. Не виникає системних помилок (трасування стека, порушення захисту пам'яті).
4. Стандартна функціональність не піддалася випадковим змінам.
5. Можливі помилкові ситуації обробляються коректно і не призводять до збою в роботі системи (втрата даних, припинення роботи системи і т.д.). При цьому слід враховувати можливі обмеження закладені спочатку при написанні ФД, якщо ці обмеження були чітко визначені.

Якщо тестувальник виявив помилки, він повинен оформити їх у вигляді звіту, який направляється розробникам для виправлення. Рекомендована форма документа «Тестовий звіт на розробку або модифікацію системи AX» додається та передбачає собою набір блоків, які використовуються для створення документа, проте потреба у використанні кожного з них визначається людиною, яка проводить тестування, під час створення документа та залежить від конкретної задачі. Секція «Детальний опис аналізу», в цьому додатку, описує дії, які проводив тестувальник, і в разі виявлення помилки, її можна відтворити, виконуючи вказані дії. Провідний розробник проекту, спільно з архітектор замовника та архітектором функціонального рішення може приймати рішення про доцільність виправлення тих чи інших помилок, визначати пріоритети їх виправлення, а також скасовувати виправлення помилок.

# Додадок 1

# Функціональний дизайн

Проект Код\_Процес

Бізнес процес «Назва/Опис»

Номер специфікації

|  |  |
| --- | --- |
| Консультант |  |
| Розробник |  |
| Бізнес архітектор |  |
| Системний архітектор |  |

Маркування розробки в розрізі задачі

|  |  |
| --- | --- |
| Код проекту |  |
| Номер задачі |  |
| Результат |  |
| Опис |  |

Необхідність змін/написання тестового скрипта і його сценаріїв

|  |  |
| --- | --- |
| Номер і назва скрипта |  |
| Примітки |  |

Підключення і взаємодія з базами

|  |  |
| --- | --- |
| Код проекту |  |
| Номер задачі |  |
| Результат |  |
| Опис |  |

Підключення і взаємодія з базами

|  |  |
| --- | --- |
| Версія клієнта |  |
| Детальна інформація про базу |  |
| Використовувані інтерфейси |  |
| Використовувані мови інтерфейсу |  |

Введення

Обґрунтування і ціль змін

Текст

Джерело - підстава

|  |  |
| --- | --- |
| Джерело | Примітки |
|  |  |

Список використовуваних термінів

Текст

Опис поточного функціоналу системи

Текст

Опис вимог

Текст

Створення нових об'єктів в рамках розробки

|  |  |  |  |
| --- | --- | --- | --- |
| Object type | ID | ENU | RUS |
|  |  |  |  |

Опис змін

Текст

Тестовий сценарій

Тестовий план 1 (позитивний кейс)

Текст

Тестовий план 2 (позитивний кейс)

Текст

...

Тестовий план X (позитивний кейс)

Текст

# Додадок 2

# Тестовий звіт на розробку або модифікацію системи AX

Проект «Код проекту»

Бізнес процес «Назва / опис»

Номер специфiкації

|  |  |
| --- | --- |
| Тестування виконав |  |
| Дата тестування |  |
| Консультант |  |
| Розробник |  |

Підключення i взаємодія з базами

|  |  |
| --- | --- |
| Версія клієнта |  |
| Детальна інформація про базу |  |
| Використовувані інтерфейси |  |
| Використовувані мови інтерфейсу |  |

Введення

Текст

* 1. Джерело - пiдстава

|  |  |
| --- | --- |
| Джерело | Примiтки |
|  |  |

Аналіз проектної документації та змінених об’єктів

Текст

|  |  |  |
| --- | --- | --- |
| Джерело | Детальний опис аналiзу | Результат |
|  |  |  |

Тестування функціональності

## Тестування змінених об'єктів

|  |  |  |
| --- | --- | --- |
| № п/п | Детальний опис аналізу | Результат |
| 1 |  | **Пройдено**  **Не пройдено** |

## Тестування зміненої функціональності, проходження тест-кейсів

### Тест-кейс 1

|  |  |  |
| --- | --- | --- |
| № п/п | Детальний опис аналізу | Результат |
| 1 |  | **Пройдено**  **Не пройдено** |

### Тест-кейс 2

|  |  |  |
| --- | --- | --- |
| № п/п | Детальний опис аналізу | Результат |
| 1 |  | **Пройдено**  **Не пройдено** |

...

### Тест-кейс Х

|  |  |  |
| --- | --- | --- |
| № п/п | Детальний опис аналізу | Результат |
| 1 |  | **Пройдено**  **Не пройдено** |

# Використана література:

1. Microsoft Corporation. “Development III in Microsoft Dynamics® AX 2012” / Microsoft Official Training Materials for Microsoft Dynamics® (2011); p.1-09 – 3-25
2. Microsoft Corporation. “Development I in Microsoft Dynamics® AX 2012” / Microsoft Official Training Materials for Microsoft Dynamics® (2011); p.1-8 – 3-2
3. Microsoft Corporation. Microsoft Dynamics® AX 2012 Best practices for developing customizations