

## Рекомендації до виконання дипломного проекту

Суть пояснювальної записки до ДП – це оформлений згідно діючих стандартів та вимог опис розробленого програмного забезпечення (ПЗ). Завдання на розробку ДП видається студенту перед початком проектування.

Викладаючи суть проекту, особливу увагу приділяють практичній цінності роботи, а також питанням сумісності, взаємозамінності, надійності, безпеки, екології, ресурсоощадності.

Якщо в роботі необхідно навести додаткову інформацію, велику за обсягом, або подробиці розроблення, їх вміщують у додатках.

Загальний обсяг основного тексту пояснювальної записки до ДП (останньої сторінки загальних висновків) повинен становити 60 – 100 сторінок.

Пояснювальна записка до ДП повинна складатись з таких розділів:

Анотація українською мовою – до 1 стор.

Анотація англійською мовою – до 1 стор.

Зміст.

Вступ – 1-2 стор.

Аналітичний розділ – 10 – 18 стор.

Постановка задачі та аналіз вимог – 10 – 15 стор.

Розділ проектування та програмної реалізації – 15-20 стор.

Розділ тестування та супроводу – 10 – 20 стор.

Розділ з економіки (згідно вимог)

Загальні висновки до роботи – 1-2 стор.

Перелік використаної літератури

Додатки.

У **вступі** відображається загальна актуальність ДП та описується предметна область і коротко обґрунтовується потреба даної розробки.

У вступі коротко викладають: готові та відомі на даний час програмні рішення, невирішені аспекти, що вирішення таких чи подібних; актуальність даної роботи, мету та галузь застосування; взаємозв'язок з іншими роботами.

В **аналітичному розділі** здійснюється аналіз предметної області та існуючі засоби вирішення проблеми. Аналітичний розділ повинен логічно завершуватись обґрунтуванням актуальності даної роботи та описом сфери застосування програмного рішення. Крім того, тут здійснюється теоретична частина ДП, аналізуються математичні моделі, алгоритми, методи та підходи, використані в роботі.

У цьому розділі слід висвітлити такі питання:

- Огляд предметної області, її специфіки та проблеми, яка буде предметом розгляду;
- Огляд та аналіз існуючих підходів до вирішення проблеми та визначення їх недоліків;

- Обґрунтування та порівняльний аналіз моделі, методу, алгоритму обраного для вирішення задачі;

В цьому розділі необхідними є посилання на актуальну літературу.

**Постановка задачі та аналіз вимог.** У цьому розділі здійснюється чітка постановка задачі, яка повинна містити призначення об'єкта проектування та вимоги до нього, вхідні та вихідні дані проекту. Виходячи з постановки задачі здійснюється аналіз вимог до розроблюваного ПЗ, результатом якого має бути представлення UML-діаграм. Аналіз вимог до розроблюваного ПЗ, проведений в цьому розділі повинен логічно продовжуватись специфікацією розроблюваного програмного засобу.

Розділ повинен містити:

- постановку задачі;
- засноване на аналізі вимог до ПЗ обґрунтування технології його програмної реалізації (з порівняльним аналізом альтернативних технологій) – за потреби;
- специфікацію програмного засобу, який буде розроблено для реалізації поставленої мети (дод. А);

**Розділ проектування та програмної реалізації** повинен містити обґрунтування технології реалізації, аналіз шаблонів проектування та обґрунтування архітектурних рішень, достатньою мірою деталізовану документацію на розроблений програмний засіб, який реалізує поставлені в другому розділі задачі згідно описаних там само вимог. При цьому слід керуватися такими стандартами на документування ПЗ як IEEE Std 830-1998 "IEEE Recommended Practice for Software Requirements Specifications", IEEE Std 1016-1998 "IEEE Recommended Practice for Software Design Descriptions", IEEE Std 1063-2001 "IEEE Standard for Software User Documentation", ДСТУ 4302:2004 Інформаційні технології. Настанови щодо документування комп'ютерних програм (ISO/IEC 6592:2000 МОД), ДСТУ ISO/IEC 15910:2007 Інформаційні технології. Процес розробки документації користувачів програмного забезпечення (ISO/IEC 15910:1999, IDT). При документуванні програмного продукту обов'язковим є використання мови UML.

- проектну документацію до програмного продукту;
- найбільш важливі деталі та технологічні аспекти програмної реалізації (за потреби);
- інструкцію з використання розробленого продукту.
- Опис отриманих результатів (великі обсяги даних, наприклад у вигляді таблиць, слід виносити в додатки);

У **розділі тестування та супроводу** наводяться результати тестування розробленого програмного продукту та рекомендації щодо його впровадження у виробництво. За необхідності проводиться оцінка ефективності практичного впровадження розроблених алгоритмів, технологій, рекомендацій і т.д. У цьому розділі повинні бути висвітлені такі питання:

- Опис та документацію проведеного тестування ПЗ (див. Дод. Б);

- Статистичну обробку та оцінку достовірності/точності отриманих результатів;
- Оцінку якості розробленого ПЗ (див Додаток В);
- Опис дій на етапі супроводу ПЗ (див Додаток Г)

### **Розділ з економіки**

Див. п.6.16.«Методичні вказівки до дипломного проектування» для студентів спеціальності 7.080403 «Програмне забезпечення автоматизованих систем» / Укл. Камінський Р.М., Павич Н.Я. – Львів: Видавництво НУ «Львівська політехніка», 2004. – 28с.

### **Загальні висновки до роботи.**

Висновки вміщують безпосередньо після викладення суті звіту, починаючи з нової сторінки.

У висновках наводять оцінку одержаних результатів роботи або її окремого етапу (негативних також) з урахуванням світових тенденцій вирішення поставленої задачі; можливі галузі використання результатів роботи; народногосподарську, соціальну значущість роботи.

Текст висновків може поділятися на пункти.

**Перелік використаної літератури** оформляється згідно ДСТУ ГОСТ 7.1:2006 "Система стандартів з інформації, бібліотечної та видавничої справи. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання" та повинен містити не менше ніж 15 джерел (приклад бібліографічного опису згідно ДСТУ ГОСТ 7.1:2006 розміщено на сторінці науково-технічної бібліотеки Львівської політехніки в Інтернеті за адресою <http://library.lp.edu.ua/node/202>).

Перелік джерел, на які є посилання в основній частині звіту, наводять у кінці тексту звіту, починаючи з нової сторінки. У відповідних місцях тексту мають бути посилання.

Бібліографічні описи в переліку посилань подають у порядку, за яким вони вперше згадуються в тексті. Порядкові номери описів у переліку є посиланнями в тексті (номерні посилання).

За необхідності джерела, на які є посилання тільки в додатку, наводять у окремому переліку посилань, який розташовують у кінці цього додатка.

**Оформлення пояснювальної записки** здійснюється згідно ДСТУ 3008-95 "Документація. Звіти у сфері науки і техніки. Структура і правила оформлення".

Пояснювальну записку оформлюють на аркушах формату А4 (210 x 297 мм). За машинного способу виконання записку друкують з розрахунку не більше 40 рядків на сторінці за умови рівномірного її заповнення та висотою літер і цифр не менш ніж 1,8 мм (шрифт Times New Roman, розмір шрифту – 14, міжрядковий інтервал – 1.5).

Текст пояснювальної записки слід друкувати, додержуючись таких розмірів берегів: верхній, лівий і нижній – не менше 20 мм, правий – не менше 10 мм.

Прізвища, назви установ, організацій, фірм та інші власні назви у звіті наводять мовою оригіналу. Допускається транслітерувати власні назви і наводити назви організацій у перекладі на мову звіту, додаючи (при першій згадці) оригінальну назву.

Структурні елементи "ЗМІСТ", "ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ", "ВСТУП", "ВИСНОВКИ", "ПЕРЕЛІК ПОСИЛАНЬ" не нумерують, а їх назви правлять за заголовки структурних елементів.

Розділи і підрозділи повинні мати заголовки. Пункти і підпункти можуть мати заголовки. Заголовки структурних елементів пояснювальної записки і заголовки розділів слід розташовувати посередині рядка і друкувати великими літерами без крапки в кінці, не підкреслюючи. Заголовки підрозділів, пунктів і підпунктів пояснювальної записки слід починати з абзацного відступу і друкувати маленькими літерами, крім першої великої, не підкреслюючи, без крапки в кінці. Якщо заголовок складається з двох і більше речень, їх розділяють крапкою. Перенесення слів у заголовку розділів не допускається. Відстань між заголовком і подальшим чи попереднім текстом має бути не менше ніж два рядки. Відстань між основами рядків заголовку, а також між двома заголовками приймають такою, як у тексті.

Сторінки пояснювальної записки, слід нумерувати арабськими цифрами, додержуючись наскрізної нумерації впродовж усього тексту пояснювальної записки. Номер сторінки проставляють у правому верхньому куті сторінки без крапки в кінці. Титульний аркуш включають до загальної нумерації сторінок записки. Номер сторінки на титульному аркуші не проставляють.

Посилання в тексті пояснювальної записки на джерела слід зазначати порядковим номером за переліком посилань, виділеним двома квадратними дужками, наприклад, "у роботах [1 – 7] ...". Допускається наводити посилання на джерела у виносках, при цьому оформлення посилання має відповідати його бібліографічному опису за переліком посилань із зазначенням номера.

## Додаток А. Шаблон оформлення специфікації вимог

### Специфікація вимог до програмного продукту для <Назва проекту>

#### 1. Вступ

##### 1.1 Призначення, мета

<Визначити продукт, вимоги до якого описані в цьому документі. Описати межі продукту, зокрема, якщо цей документ описує лише частину системи чи окрему підсистему.>

##### 1.2 Продукти-аналоги (при наявності таких)

<Навести у вигляді таблиці порівняння результати аналізу основних функціональних і нефункціональних характеристик продуктів-аналогів. Можна включати зразки користувацьких інтерфейсів та посилання на Web адреси.>

#### 2. Загальний опис

##### 2.1 Характеристики продукту

<Резюмувати основні характеристики продукту або істотні функції, які він здійснює чи дозволяє здійснювати користувачу. Деталі представляються в Розділі 3, тому тут потрібне узагальнення вищого рівня. Ефективним є представлення основних груп пов'язаних вимог і їхніх зв'язків діаграмами варіантів використання.>

##### 2.2 Класи користувачів та їх характеристики

<Визначити різні класи користувачів, які будуть використовувати продукт. Класи користувачів можуть бути диференційовані базуючись на частоті використання, підмножині функцій продукту, яка використовується, технічній експертизі, рівнях безпеки чи привілеїв, рівню освіти чи досвіду. Описати доцільні характеристики кожного класу користувачів. Відділити пріоритетні класи користувачів від тих, що є менш важливими.>

##### 2.3 Середовище функціонування

<Описати середовище в якому буде функціонувати продукт, включаючи апаратну платформу, операційну систему, її версії, і будь-які інші програмні компоненти чи аплікації, з якими воно має коректно співіснувати.>

#### 3. Характеристики системи

<Ця частина ілюструє організацію функціональних вимог до продукту через характеристики системи, основні сервіси, які надає продукт. Цей розділ можна представити варіантами використання, режимами операцій, користувацькими класами, , чи їх комбінації, залежно від того, що найбільш логічно для продукту.>

##### 3.1 Характеристика системи 1

<Не записуйте "Характеристика системи 1". Задавайте **конкретну**, змістовну назву характеристики кількома словами.>

###### 3.1.1 Опис і пріоритет

<Надайте короткий опис характеристики і відзначте, якого вона пріоритету Високого, Середнього, чи Низького.>

###### 3.1.2 Послідовності дія/відгук

<Список послідовностей дій користувача і відгуків системи, що спричиняють режим визначений для цієї характеристики. Це відповідає елементам діалогу асоційованим з варіантами використання.>

###### 3.1.3 Функціональні вимоги

<Перелічити детальні функціональні вимоги асоційовані із цією характеристикою. Це можливості продукту, які мають бути реалізовані, щоб користувач скористався сервісами чи виконав варіант використання. Включаючи, як продукт повинен реагувати на помилкові умови чи неправильні введення. Вимоги мають бути короткими, повними, недвозначними, верифіковуваними, і необхідними.>

*<Кожна вимога має бути унікально ідентифікована номером чи значущою міткою певного виду, наприклад*

*REQ-1.1:*

*REQ-1.2: .>*

### **3.2 Характеристика системи 2 (і т.д.)**

## **4. Вимоги зовнішніх інтерфейсів**

### **4.1 Користувацькі інтерфейси (при потребі)**

*<Описати логічні характеристики кожного інтерфейсу між ПЗ та користувачами. Може включати зразки зображень екрану.>*

### **4.2 Апаратні інтерфейси (при потребі)**

*<Описати логічні та фізичні характеристики кожного інтерфейсу між ПЗ та апаратними компонентами системи. Може включати типи підтримуваних пристроїв, природу даних і керуючих взаємодій між ПЗ та апаратними засобами і комунікаційні протоколи, які будуть використані.>*

### **4.3 Програмні інтерфейси (при потребі)**

*<Описати зв'язок між продуктом і іншими специфічними програмними компонентами (назва і версія), включаючи бази даних, операційні системи, інструменти, бібліотеки і інтегровані комерційні компоненти. Визначити дані і повідомлення, які поступають в систему і виходять з неї і описати мету кожної.>*

### **4.4 Комунікаційні інтерфейси (при потребі)**

*<Описати вимоги, що пов'язані з комунікаційними функціями, необхідними цьому продукту, зокрема, електронна пошта, веб браузер, мережеві протоколи, електронні форми і т.п. Визначити прийнятні формати повідомлень. Визначити комунікаційні протоколи, які будуть використовуватись, такі як FTP чи HTTP. Визначити безпеку комунікацій чи питання шифрування, швидкість передачі даних, і механізми синхронізації.>*

## **5. Інші нефункційні вимоги**

### **5.1 Вимоги продуктивності**

*<Якщо є вимоги продуктивності до продукту в різних середовищах, описати їх та пояснити. Визначити часові залежності для систем реального часу. Описати такі вимоги настільки точно, як це можливо.>*

### **5.2 Вимоги безпеки**

*<Визначити вимоги, що стосуються безпеки чи питань секретності, щодо використання продукту чи захисту даних, які використовуються чи створюються. Визначити вимоги аутентифікації користувачів.>*

### **5.3 Атрибути якості програмного продукту**

*<Визначити додаткові якісні характеристики до продукту, які будуть важливими для замовників чи розробників. Зокрема, це може бути: адаптовуваність, придатність, коректність, гнучкість, функціональна сумісність, супроводжуваність, портативність, надійність, стійкість, тестопридатність та зручність використання.>*

## **6. Інші вимоги**

*<Визначити інші вимоги, що не розкриті в SRS. Це може включати вимоги бази даних, вимоги інтернаціоналізації, юридичні вимоги і т.п.>*

## Додаток Б. Рекомендації до створення звіту про тестування

1. Вступ.
2. Розробка тестів (які види тестування використовувались, які були розроблені тестові випадки).
  - 2.1. Функціональні тести.
  - 2.2. GUI тести.
  - 2.3. Тести на безпеку.
  - ...
  - 2.N. Тестування продуктивності.
3. Функціональне тестування.
  - 3.1. Пройдені тести, їх результат.
  - 3.2. Не пройдені тести.
4. GUI тестування.
- ...
- N-2. Тестування продуктивності.
- N-1. Метрики (покриття програмних вимог тестами).
- N. Критерій прийняття/відхилення релізу.

2. Розробка тестів. Під час фази розробки тестових випадків було спроектовано тести для функціонального тестування, тестування безпеки чи GUI тестування та сценарії для тестування продуктивності (або вказати інші види тестів, які використовувалися). Кожен тестовий випадок містить детальні кроки, тестові дані і очікувані результати.

2.1. Функціональні тестові випадки. Під час фази розробки тестів було спроектовано 34 функціональні тестові випадки. Таблиця показує розподіл функціональних тестових випадків і наборів тестових даних для цих випадків за варіантами використання.

| Варіанти використання | Тестові випадки | Тестові дані |
|-----------------------|-----------------|--------------|
| Setup Account         | 2               | 7            |
| Trade Securities      | 9               | 79           |
| View Performance      | 4               | 13           |
| Edit Security         | 9               | 27           |
| Edit Trader Account   | 7               | 11           |
| View System Reports   | 3               | 168          |
| <b>Загалом</b>        | <b>34</b>       | <b>168</b>   |

Додаток 2 «Специфікація тестів для функціонального тестування» містить деталі функціональних тестових випадків.

| #  | Тестові випадки                       | Тестові дані |
|----|---------------------------------------|--------------|
| 1. | Verify Image And Text                 | 18           |
| 2. | Verify Sort Order Fields              | 67           |
| 3. | Verify Scrolling                      | 8            |
| 4. | Verify Required Fields                | 38           |
| 5. | Verify Max Length In Edit Fields      | 31           |
| 6. | Verify Links                          | 25           |
| 7. | Verify Default Values In Fields       | 17           |
| 8. | Verify Data Validation In Edit Fields | 41           |
| 9. | Verify Buttons                        | 39           |

| #  | Тестові випадки             | Тестові дані |
|----|-----------------------------|--------------|
| 10 | Verify Confirmation Message | 25           |
| 11 | Verify UI Skin-ability      | 18           |
|    | <b>Загалом</b>              | <b>327</b>   |

2.2. *GUI тестові випадки.* Було розроблено 11 GUI тестових випадків. Нижче наведена таблиця показує список GUI тестових випадків з кількістю наборів тестових даних для кожного тестового випадку.

Додаток 3 «Специфікація тестів для GUI тестування» містить детальні GUI тестові випадки.

...

2.п. Аналогічний опис тестових випадків для інших видів тестування

### 3. Функціональне тестування

#### *Результати тестування*

Результати функціонального тестування наведено в додатку *Результати виконання функціонального тестування*.

#### *Підсумок тестування*

33 тести з 34 функціональних тестів наведених в Додатку2 «Специфікація тестів для функціонального тестування» пройшли успішно (169 із 170 множин тестових даних пройшли), отже функціональне тестування розглядається як частково успішне – 97% тестових випадків пройшли, 99.94% наборів тестових даних пройшли.

#### *Відомі дефекти*

##### *Дефект #1*

Коли адміністратор редагує профіль користувача система не інформує адміністратора, що користувач в даний момент залогований.

#### *Опис:*

Відповідно до тестового випадку Edit Trader Account (UTASK10298) система повинна інформувати адміністратора, що користувач в даний момент залогований, якщо адміністратор намагається редагувати профіль користувача.

Дефект полягає в тому, що система не інформує адміністратора про те, що користувач залогований. Адміністратор може змінювати профіль користувача у той час коли користувач залогований у систему.

#### *Подолання дефекту:*

Цей дефект неможливо виправити без шкоди для WEB сервера системи. Проблема полягає в тому, що доступ до контейнера сесії є обмеженим і небезпечним.

### *GUI тестування*

#### *Результати тестування*

Результати GUI тестування представлені у додатку *Результати виконання GUI тестування*.

#### *Підсумок тестування*

10 з 11 GUI тестів наведених у Додатку Test Design Specification GUI Testing.doc [3] пройшли успішно (324 із 327 наборів тестових даних пройшли), отже GUI тестування можна розглядати як частково успішне- 90.9% тестових випадків пройшли успішно, 99.1% наборів тестових даних пройшли.

#### *Відомі дефекти*

##### *Дефект #2*

Користувач не може переміщувати вказівник прокрутки на сторінці Портфолію користувача. Цей дефект проявляється лише в оглядачі FireFox.

#### *Опис:*

Користувач не може переміщувати вказівник прокрутки на сторінці Портфолію користувача. Вказівник прокрутки можна перемістити лише з допомогою стрілок прокрутки. Цей дефект проявляється лише в оглядачі FireFox.



*Подолання дефекту:*

Це дефект FireFox v x.x: вказівник прокрутки фрейму об'єкта втрачає фокус.

*Критерій успіх/провал проекту*

Умови тестування, які визнавалися успішними були наступні:

Розробка тестів:

- Всі заплановані тестові випадки розроблено;
- Покриття тестами програмних вимог досягає 100%;
- Покриття тестами варіантів використання досягає 100%;

Тестування:

- Всі розроблені тестові випадки виконано;
- Виконано тестування продуктивності, вимоги продуктивності задоволено;
- Всі внутрішні дефекти виправлені і виправлення підтверджено.

Всі наведені умови задоволено, проект вважається успішним.

## **Додаток В. План забезпечення якості програмного продукту**

SQAP включає розділи, відповідно до стандарту *IEEE 730 Standard for Software Quality Assurance Plans* (Стандарт планування забезпечення якості ПЗ - IEEE 730). Розділи повинні бути впорядковані в описаній послідовності. Якщо немає інформації, що відноситься до розділу, то нижче заголовку розділу потрібно вказати: "Цей розділ не відповідає даному плану", разом з відповідними причинами виключення.

SQAP складається з наступних пунктів:

1. Мета.
2. Управління.
3. Документація.
4. Стандарти, практики, узгодження і метрики.
5. Тестування.
6. Звіти про помилки та коригувальні дії.
7. Засоби, методи та методології.
8. Медіа-контроль.
9. Контроль постачання.
10. Збір, підтримка і зберігання обліку.
11. Навчання.
12. Управління ризиками.

Деякі матеріали можуть бути наведені в інших документах. В такому випадку посилання на ці документи повинні бути присутні в SQAP.

### **4.1. (Нумерацію ставити згідно диплому) Мета (розділ 1 SQAP)**

Цей розділ повинен визначати конкретні цілі та границі окремого SQAP. Тут наводиться перелік програмних засобів, які охоплюються SQAP, і плановане використання програмного забезпечення. Для кожної одиниці програмного забезпечення повинен бути наведений опис життєвого циклу, який охоплюється SQAP.

### **4.2. Управління (розділ 2 SQAP)**

Цей розділ описує структуру організації проекту, її завдання, ролі та обов'язки (див. IEEE Std 1058™ -1998 [B13]).

#### **4.2.1. Організація**

Цей розділ описує організаційну структуру, яка контролює і забезпечує якість програмного забезпечення. Повинен бути наведений опис кожного з основних елементів організації разом з ролями і делегованими повноваженнями. Рівень організаційної свободи і об'єктивність оцінки та контролю якості програмного забезпечення, повинні бути чітко описані і задокументовані. Крім того, повинні бути ідентифіковані організації, відповідальні за підготовку і підтримку SQAP.

#### **4.2.2. Завдання**

У цьому розділі наводиться опис:

1. Етапів життєвого циклу програмного забезпечення, охоплених SQAP.
2. Завдань, що мають бути виконані.
3. Вхідних і вихідних критеріїв для кожного завдання.
4. Взаємозв'язків між цими завданнями і планованих основних контрольних точок. Також повинні бути вказані послідовність завдань та їх зв'язок з графіком проекту.

#### **4.2.3. Ролі та обов'язки**

Цей розділ має визначити конкретні організаційні елементи, які є відповідальними за виконання кожного завдання.

#### **4.2.4. Оцінка ресурсів, необхідних для забезпечення якості.**

Цей розділ передбачає оцінку ресурсів і витрат, які будуть витрачені на забезпечення якості і завдання контролю якості.

### **4.3. Документація (розділ 3 SQAP)**

#### **4.3.1. Мета**

Цей розділ повинен виконувати наступні функції:

- а) Визначити документацію для регулювання розробки, контролю та перевірки, використання та технічного обслуговування програмного забезпечення.
- б) Описати документи, які повинні бути розглянуті і перевірені на відповідність. Для кожного документа в списку, визначити перегляди і аудити, які будуть проводитися, і критерії, за якими відповідність повинна бути підтверджена.

#### **4.3.2. Мінімальні вимоги до документації**

Опис мінімальних вимог до написання документів, метою яких є перевірка того, що реалізація програмного забезпечення задовольняє технічні вимоги.

##### **4.3.2.1. Опис вимог до програмного забезпечення (SRD)**

SRD повинен визначати вимоги для конкретного програмного продукту, програми або набору програм, які виконують певні функції в конкретному середовищі. SRD може бути записаний постачальником (внутрішнім або зовнішнім), замовником, або обома. У SRD слід розглянути основні питання функціональності, зовнішні інтерфейси, продуктивність, атрибутів та обмежень на реалізацію. Кожна вимога повинна бути однозначно ідентифікована та визначена так, що її задоволення здатне бути об'єктивно перевіреним і підтвердженим (див. IEEE Std 830™ -1998 [B5]).

#### **4.3.2.2. Опис дизайну програмного забезпечення (SDD)**

SDD повинен описувати, як програмне забезпечення буде будуватися відповідно до вимог SRD. SDD повинен описувати компоненти й підкомпоненти розробки програмного забезпечення, включаючи бази даних і внутрішні інтерфейси. SDD може бути підготовлений як архітектура (верхній рівень SDD) і повинен бути в подальшому розширений для отримання докладних SDD (див. IEEE Std 1016™ -1998 [B11]).

#### **4.3.2.3. Плани верифікації та валідації**

Верифікація використовується для визначення того, чи розроблені програмні продукти відповідають вимогам, і чи є програмне забезпечення продукція відповідною призначенню та очікуванням користувачів. це включає в себе: аналіз, контроль, оцінку і тестування програмних продуктів і процесів, які стосуються випуску продукту. Крім того, тестування програмного забезпечення, атестація і перевірка процесів застосовуються при інтеграції зовнішніх частин розроблюваного продукту.

План визначає завдання верифікації та валідації і входів і виходів, необхідних для підтримки потрібного рівня цілісності програмного забезпечення. Він також надає засоби тестування та розробки вимог SRD, щодо розробки, як описано в SDD, і щодо тестування, як описано в проектній документації.

#### **4.3.2.4. Звіт з верифікації та валідації**

У звіті про результати верифікації та валідації повинні бути описані результати верифікації та валідації програмного забезпечення та заходи, що проводяться відповідно до плану перевірки.

#### **4.3.2.5. Користувацька документація**

Користувацька документація містить опис встановлення, експлуатації, управління та обслуговування (не застосовується при зміні вихідного коду програмного забезпечення) програмних продуктів.

Документація повинна описувати контроль вихідних даних, вхідних послідовностей, параметри програми, обмеження, і всі інші необхідні дані для програмного продукту. Всі повідомлення про помилки повинні бути визначені і описані. Повинні бути описані всі коригувальні дії, направлені на виправлення помилок, описаних в повідомленнях про помилку. Документація поширюється на будь-яку частину вбудованого програмного забезпечення, з якими користувач взаємодіє безпосередньо.

#### **4.3.2.6. План конфігураційного управління (SCMP)**

SCMP повинен документувати дії з управління конфігурацією програмного забезпечення (SCM), що повинні бути виконані, а також те, як вони повинні бути виконані, хто відповідає за конкретні завдання, графік заходів, і які ресурси будуть використані. SCMP повинен також визначати методи і засоби, що використовуються для підтримки, збереження, захисту і документування контрольованих версій та пов'язані з ним артефакти, виявлені на всіх етапах життєвого циклу програмного забезпечення.

#### **4.3.3. Інші документи**

Визначаються інші документи, що застосовуються в проекті розробки програмного забезпечення і програмних продуктів. Інші документи можуть включати наступне:

1. План процесу розробки.
2. Опис стандартів розробки.
3. Опис інженерних методів / процедур / засобів.
4. План управління проектом (див. IEEE Std 1058 <sup>™</sup> -1998)
5. План обслуговування (див. IEEE Std 1219 <sup>™</sup> -1998)
6. Плани безпеки програмного забезпечення (див. IEEE Std 1228)
7. План інтеграції програмного забезпечення

#### **4.4 Стандарти, практики, конвенцій, і метрики (див. розділ 4 SQAP)**

##### **4.4.1 Мета**

Цей розділ повинен:

- а) визначити стандарти, практики, конвенції, статистичні методи, що будуть використовуватися, вимоги до якості і показники, які повинні застосовуватися. Показники характеристик продуктів і процесів повинні бути включені в метрики, що використовуються, і можуть бути визначені в окремому плані.
- б) Положення про те, як повині контролюватися та гарантуватися відповідності між цими елементами.

##### **4.4.2 Зміст**

Питання, що розкриваються у змісті повинні включати основні технічні, дизайнерські і інші аспекти діяльності зокрема пов'язані із програмуванням, наприклад, документація, змінна і іменування модулів, програмування, контроль якості і тестування.

- а) Стандарти документації;
- б) Стандарти дизайну;
- в) Стандарти кодування;
- г) Стандарти коментування;
- е) Визначення стандартів і практик;
- е) Обраний продукт забезпечення високої якості програмного забезпечення, гарантії та метрики даного процесу;

#### **4.5 Тестування (розділ 5 SQAP)**

Цей розділ визначає всі тести, не включені до етапу верифікації та валідації програмного забезпечення, охоплених SQAP і повинні вказувати на методи, які будуть використовуватися. Якщо окремий план тестування існує, то він повинна бути вказаний.

#### **4.6 Звітність проблем і коригувальних дій (розділ 6 SQAP)**

Цей розділ повинен:

- а) Описати методи і процедури, які використовуються для звітності, відстеження і вирішення проблем або питань, визначених в обох процесах - розробки та супроводу.
- б) Вказати конкретні організаційні обов'язки, пов'язані із їх вирішенням.

#### **4.7 Засоби, методи і методики (розділ 7 SQAP)**

Цей розділ визначає програмні засоби, методи, що використовуються для підтримки SQA процесів. Для кожного із них, даний розділ визначає призначення, застосування, або обставини, при яких він буде, чи навпаки, не буде використовуватися, а також обмеження.

#### **4.8 Контроль середовища (розділ 8 SQAP)**

У цьому розділі зазначаються методи і засоби, які будуть використовуватися для:

- а) Визначення середовища для кожного проміжного результату та документації, у тому числі процесу копіювання і відновлення.
- б) Захист комп'ютерних програм на фізичних носіях від несанкціонованого доступу або випадкового пошкодження чи деградації на всіх етапах життєвого циклу програмного забезпечення. Також диний етап може бути забезпечений як частина SCMP.

#### **4.9 Контроль розробника (розділ 9 SQAP)**

У цьому розділі повинні бути вказані положення, які гарантують, що програмне забезпечення, що надається розробниками відповідає встановленим вимогам. Крім того, в цьому розділі зазначаються методи, які будуть використовуватися для забезпечення того, що отримувач отримує програмне забезпечення адекватне і відповідне вимогам. Для раніше розроблених програм, цей розділ повинен вказати методи, які будуть використовуватися для забезпечення придатності продукту для використання засобів програмного забезпечення, описаних у SQAP.

У цьому розділі зазначаються також методи, які будуть використовуватися для забезпечення відповідності вимогам згідно стандарту. Якщо програма має бути розроблена відповідно контракту, то процедури перегляду договору і оновлення повинні бути описані.

#### **4.10 Збір коментарів, їх підтримка і збереження (розділ 10 SQAP)**

Цей розділ визначає SQA документацію, що буде збережена, визначає методи та засоби, щоб будуть використовуватися для складання, захисту і збереження даної документації, і визначає термін її зберігання.

#### **4.11 Навчання (розділ 11 SQAP)**

Цей розділ визначає заходи з підготовки кадрів, необхідних для задоволення потреб SQAP.

#### **4.12 Управління ризиками (розділ 12 SQAP)**

У цьому розділі зазначаються методи та процедури, які використовуються для ідентифікації, оцінки, моніторингу і контролю зони ризику, що виникають в ході життєвого циклу програмного забезпечення, які охоплюються SQAP.

## **Додаток Г. Методичні вказівки щодо складання плану впровадження та супроводу ПЗ**

Впровадження ПЗ в експлуатацію та його супровід є останніми ключовими етапами життєвого циклу ПЗ. Етап впровадження, як правило, є найкоротшим за тривалістю серед всіх інших етапів, але потребує надзвичайного уважного планування і врахування багатьох факторів. Етап супроводу часто виявляється найбільш тривалим етапом життєвого циклу ПЗ і забезпечує підтримку функціонування та еволюцію програмного продукту. Зміст цих етапів докладно розкривається в стандартах ISO/IEC 12207 «Software Life Cycle Processes» («Процеси життєвого циклу ПЗ»), ISO/IEC 14764:2006 «Software Engineering – Software Life Cycle Processes – Maintenance» («Інженерія ПЗ – Процеси життєвого циклу ПЗ – Супровід») та IEEE 1219-1993 «IEEE Standard for Software Maintenance» («Стандарти IEEE для супроводу ПЗ»).

Особливість етапів впровадження та супроводу полягає в тому, що в більшості випадків вони (особливо, етап супроводу) виконуються сторонніми організаціями, які не приймали участі в розробці програмного продукту. В зв'язку з цим надзвичайно важливим стає передача всієї необхідної інформації від розробників до безпосередніх виконавців робіт з впровадження та супроводу. Зокрема, для грамотного виконання впровадження та супроводу ПЗ розробляють відповідні плани, що висвітлюють організаційні, технічні та економічні аспекти робіт з впровадження та супроводу.

В залежності від масштабу програмної системи, плани її впровадження та супроводу можуть бути поєднані в одному документі або рознесені по різних документах. В мережі Інтернет існує ціла низка шаблонів для створення планів впровадження та супроводу. Зокрема, такі шаблони можна знайти в <http://www.mysoftwaretemplates.com/software-development-templates/>

Загальна структура плану впровадження програмного продукту:

1. Вступ (загальна інформація про призначення плану впровадження)
2. Ролі і відповідальності (перелік осіб, задіяних у впровадженні системи з їх ролями та обов'язками)
3. Часовий план (розклад впровадження системи із зазначенням всіх задач і відведеного на них часу)
4. Навчання персоналу (порядок і терміни навчання користувачів системи)
5. Робоче середовища (конфігурація обчислювального середовища для встановлення розробленої системи)
6. Процедура інсталяції (покроковий опис процедури розгортання системи в робочому середовищі з базовою перевіркою працездатності системи)
7. Запуск в експлуатацію (міграція актуальних даних в системи і переведення останньої в робочий режим)

В процесі розробки плану впровадження необхідно звернути увагу на наступні діяльності:

- Виведення існуючої системи з експлуатації, якщо мова йде про заміну існуючої системи на нову;
- Підготовка робочого середовища для запуску нової системи, яке може базуватися на виділені серверах, розподілених серверах або на сервісах хмари;
- Розгортання розробленої системи в підготовленому робочому середовищі відповідно до інструкцій, отриманих від розробників;
- Інтеграція з зовнішніми сервісами, такими як сервіси географічних карт, сервіси оплати кредитними картками, сервіси валідації поштових адрес тощо;
- Міграція даних зі старою системи в нову, що може потребувати не лише копіювання, але й трансформації даних;
- Навчання користувачів роботі з новою системою, що включає підготовку і проведення тренінгу, віддалено або на базі замовника чи розробника.

План супроводу в загальному випадку має таку структуру:

1. Вступ (загальна інформація про призначення плану супроводу)
2. Огляд системи (коротка характеристики програмної системи, її основні функції та задачі)
3. Середовище супроводу (опис всіх допоміжних систем та засобів, які забезпечують процес супроводу)
4. Персонал супроводу (особи, задіяні в роботах з супроводу, їх функції та обов'язки)
5. Процедури супроводу (докладний опис всіх процедур супроводу із висвітленням організаційних, технічних та економічних аспектів)

План супроводу повинен докладно висвітлювати всі процедури супроводу, прийняті для даного програмного продукту. Основними процедурами супроводу є:

- Звітування, аналіз, виправлення та відслідковування дефектів в програмній системі;
- Подання, реалізацію та відслідковування запитів на розширення функціональності системи;
- Додаткові процедури підтримки, такі як резервне копіювання бази даних, системні поновлення, моніторинг завантаженості обчислювальних ресурсів, модернізація апаратного забезпечення, тощо.

Для складних систем плани впровадження та супроводу можуть виступати в ролі агрегуючих документів і мати посилання на окремі документи, такі як керівництво з розгортання системи або інструкція користувача.