

7 Documentation requirements

1. Що таке технічна документація?

1. Документація на програмне забезпечення (software documentation) – це комплекс документів, в яких докладно описуються технічні характеристики та споживчі якості ПЗ, а також відомості про процес його розробки, застосування та супроводу. Software documentation поділена на 2 основні категорії: Product documentation — описує продукт, що розробляється та Process documentation — описує процес на проекті.

До Product documentation відносяться:

- Вимоги (Requirements) – заяви, які ідентифікують атрибути, можливості, характеристики або якості системи. Це основа того, що буде або вже було реалізовано.
- Архітектура/дизайн (Architecture/Design) – огляд програмного забезпечення. Включає зв'язки з середовищем і принципи побудови, які будуть використовуватися при проектуванні програмних компонентів.
- Технічна документація (Technical documentation) – іноді так називають абсолютно всю документацію для програмного забезпечення, хоча це не зовсім правильно. Безпосередньо техдокументація включає опис програмного коду та виконуваних ним функцій, структур даних, алгоритмів, API та інтерфейсів.
- Документація із забезпечення якості (Quality Assurance documentation) – включає у себе:
 - Test strategy — описує підхід який ми будемо використовувати у тестування, щоб досягнути тестових цілей;
 - Test plan — розкриває інформацію, що має бути протестовано в даний момент;
 - Test case specifications — це набір детальних дій для перевірки кожної функції продукту;
 - Test Checklists — набір тестів що мають бути проведені.
- Посібник з обслуговування та допомоги (Maintenance and help guide) — описує відомі проблеми з системою та їх вирішення. Також представляє залежності між різними частинами системи.
- Документація кінцевого користувача (End-User Documentation) – посібники для кінцевого користувача, системних адміністраторів і персоналу служби підтримки.

До Process Documentation відносяться:

- Plans, estimates, and schedules (Плани, кошториси та графіки) - ці документи зазвичай створюються до початку проекту та можуть бути змінені в міру розвитку продукту.
- Reports and metrics (Звіти та показники) — відображають, як час і людські ресурси були використані під час розробки.

2. Яка мета документа вимог до програмного забезпечення (software requirements document)?

2. Документ вимог до програмного забезпечення (software requirement document) – це структурований набір вимог/запитів (функціональність, продуктивність, конструктивні обмеження та атрибути) до програмного забезпечення та його зовнішніх інтерфейсів. Призначений для встановлення бази для угоди між замовником і розробником про те, як повинен функціонувати програмний продукт.

Наявність чіткого набору вимог (software requirements specification) є гарантом того, що команда розробників створить програмне забезпечення, що відповідає потребам клієнта. SRS (software requirements specification) допоможе оцінити вартість робіт та охопити обсяг проекту. Він також дає програмістам уявлення про технологічний стек (набір інструментів, що застосовується під час роботи у проектах і включає мови програмування, системи управління базами даних, компілятори тощо), який їм знадобиться, та допомагає планувати роботу. Але це ще не все: дизайнери одержують уявлення про проект через документи SRS, щоб вони могли адаптувати дизайн до варіанта використання. Тестувальники отримують рекомендації щодо створення тестових випадків, щоб мати змогу якісно протестувати продукт. Ще SRS важливий, тому що це єдине джерело інформації, яке запобігає непорозумінню як між менеджером проекту та командою, так і між замовником та командою розробників.

3. Назвіть два інструменти, які використовуються для відстеження вимог до програмного забезпечення?

3. До найпростіших інструментів, які використовуються для відстеження вимог, можна віднести такі програми, як Microsoft Word та Microsoft Excel. Також відстежувати вимоги можна за допомогою додатків Google, таких як Google Docs та Google Sheets. Ще приклади інструментів управління вимогами (requirements management tools) - ReQtest, Jama, Jira.

4. Чому ми повинні використовувати user stories замість функціональних вимог (functional requirements)?

4. User story (історія користувача) - це опис функціональної можливості ПЗ простими, загальними словами, складений з точки зору кінцевого користувача. Вона пишеться для роз'яснення, як саме функціональна можливість принесе користь клієнту. В основу розробки ПЗ ставиться людина, і користувальницькі історії якраз служать для того, щоб у центрі обговорення

завжди були фактичні користувачі. Історії пишуться простою мовою, без технічної специфіки, і є контекстом для команди розробників та їх діяльності. За допомогою user stories можна простими словами донести до усіх учасників опис функціональності продукту без технічних деталей. Прочитавши історію користувача, команда знає, чому вона створює те, що створює, і яку цінність це формує. User story пишеться з метою роз'яснення того, як саме виконання робочого завдання призведе до створення конкретної цінності клієнта. Історії користувача складаються з декількох пропозицій, що описують необхідний результат простою мовою і в загальних рисах. Вони не містять дрібниць. На основі написання Acceptance Criteria до user stories, формуються функціональні вимоги до продукту.

5. Яка різниця між feature та user stories?

5. Feature (функція) — це окремий елемент функціональності програмного забезпечення. User story (історія користувача) — це невеликий аспект функції, за допомогою якого можна отримати відгук від зацікавлених сторін і дізнатися, чи робить команда щось не так.

User story (історія користувача) — це неформальна заява мовою клієнта, яка відображає намір чогось, чого клієнт хоче досягти. Можна розглядати історію користувача як неформальну заяву про вимоги, як опис функції. Feature (функція) програмного забезпечення — це окрема характеристика програмного забезпечення, яка сприяє загальному дизайну та функціональності програмного забезпечення.

6. Що таке Acceptance Criteria (AC) для QA?

6. Для забезпечення якості Acceptance Criteria (критерії прийомки) визначають обсяг того, що необхідно перевірити. Вони повідомляють QA, що має бути перевірено, а що не повинно бути перевірено. Вони описують, яку функціональність реалізує програмне забезпечення. Також Acceptance Criteria описують будь-які відхилення в поточному проекті.

Acceptance Criteria інформують тестувальників, коли починати тестування і коли завдання можна вважати виконаним. Крім того, це допомагає їм зрозуміти, що саме потрібно тестувати. Критерії прийомки є важливою передумовою для проведення приймальних випробувань. Acceptance Criteria допомагають визначити, чи відповідає продукт встановленим вимогам і чи всі вимоги є протестованими. Критерії прийомки надають тестувальникам достатню інформацію для документування детальних тестових сценаріїв і випадків. Вони також допомагають встановити зв'язок між розробниками та тестувальниками для координації між собою та обговорення будь-яких відхилень від встановлених вимог.

7. Напишіть Acceptance Criteria для такої User Story: *As a user, i can login the website to perform restricted operations (Як користувач, я можу увійти на веб-сайт для виконання обмежених операцій).*

7.

Scenario: Виконання обмежених операцій на веб-сайті

Given: Користувач заходить на головну сторінку веб-сайта

When: Користувач проходить реєстрацію на веб-сайті

And: Користувач, використовуючи реєстраційні дані, входить на веб-сайт як авторизований користувач

Then: Користувач має можливість виконувати обмежені операції.

8. Напишіть Acceptance Criteria для такої User Story:
As a customer of "Rozetka" i want to see oreders in my account, so that i can track order and know when it will arrive (Як клієнт «Розетки», я хочу бачити замовлення у своєму обліковому записі, щоб я міг відстежувати замовлення та знати, коли вони прибудуть).

8.

1. Користувач повинен бути зареєстрований на сайті «Rozetka»
2. Користувач заходить у свій акаунт та переходить у розділ «Мої замовлення»
3. Користувач знаходить свої замовлення у розділі «Мої замовлення»
4. Користувач натискає кнопку «Відстежити замовлення»
5. Відбувається перехід на сторінку відстеження замовлень.

9. Напишіть Acceptance Criteria для такої User Story:

As a member of conference "Save Planet", i want to be able to registe online, so i do not need to attend "Save Planet office (Як учасник конференції «Save Planet», я хочу мати можливість зареєструватися онлайн, тому мені не потрібно відвідувати офіс «Save Planet».).

9.

1. Користувач заходить на сайт «Save Planet»
2. Користувач проходить реєстрацію на сайті «Save Planet»
3. Користувач заповняє форму реєстрації для участі у конференції
4. Користувач отримує підтвердження про успішну реєстрацію на конференцію на E-mail.

10. **Напишіть Acceptance Criteria для такої User Story:**

As a candidate who is looking for a job I want report from Linkedin to be sent to my inbox daily So that i can monitor progress of my CV (Як кандидат, який шукає роботу, я хочу, щоб звіт із Linkedin надсилався на мою поштову скриньку щодня, щоб я міг стежити за розвитком свого резюме.)

10.

1. Користувач зареєстрований на сайті Linkedin
2. Користувач має сформоване резюме на сайті Linkedin
3. Користувач вмикає функцію надсилання на e-mail щоденного звіту зі сайту Linkedin та обирає час, у який буде приходити звіт.
4. У обраний час користувач отримує лист на e-mail зі сформованим за останню добу звітом зі сайту Linkedin.

11. У вас є вимоги до продукту. Розбийте task Аналіз вимог до програмного забезпечення згідно Work Breakdown Structure.

11.

1. Перевірки вимог на послідовність та логічність
2. Перевірки на двозначність
3. Перевірка можливості на неможливості тестування вимог
4. Перевірка технічної можливості реалізації вимог
5. Аналіз сценаріїв
6. Виявлення залежностей у вимогах
7. Перевірка на несуперечність (тестування вимог із питаннями «а що якщо так?» «а що якщо інакше?»)
8. Декомпозиція вимог (Розбиття вимог на більш дрібні та чіткі вимоги до цілісних функцій).

12. У вас є продукт, а також можливі сценарії:

Сценарій 1: Команда потребує 30 год, щоб здійснити тестування. При цьому сценарії не виникає жодних непередбачуваних обставин (*best-case scenario(b)*)

Сценарій 2: Команда потребує 45 год, щоб здійснити тестування. Нормальний сценарій, коли очікується виникнення проблем і більшість завдань буде виконано з меншою кількістю відставання в майбутніх спринтах (*most-likely scenario (m)*)

Сценарій 3: Команда потребує 60 год, щоб здійснити тестування. Сценарій, коли майже вся ситуація йде не так через брак кваліфікованих ресурсів або через те, що багато членів нещодавно введено в команду *worst-case scenario (W)*.

Здійсніть естімацію для продукту базуючих на техніках які вам відомі з лекції (оберіть доречну техніку у даному випадку).

12. Для виконання цього завдання використовуємо такий метод оцінки зусиль для завдань (Effort estimation for tasks), як Three Point Estimation (3-точкова техніка оцінки тестування програмного забезпечення). Метод оцінки зусиль для завдань (Effort estimation for tasks) відноситься до Metric-based technique (техніка на основі метрик).

Найкраща оцінка (b) – 30 годин

Найгірша оцінка (w) – 60 годин

Оптимальна оцінка (m) – 45 годин

Зусилля для даного завдання розраховуються за наступною формулою:

$$E=(b+4m+w)/6$$

$$E=(30+4*45+60)/6=270/6=45$$

Отже, загальна оцінка зусиль для виконання тестування буде становити 45 годин.

13. У вас є аплікація - Онлайн банкінг. У ній є три модулі програми із Weightage=5, п'ять модулів програми із Weightage=3, чотири модулі програми із Weightage=1. Одна Function Points =5 год/одиночку. Підрахуйте загальну оцінку зусиль на тестування.

13. Для виконання цього завдання використовуємо такий метод оцінки зусиль для завдань (Effort estimation for tasks), як Functional Point Method (Метод функціональної точки).

	Weightage	Amount of test points	Total
Simple	1	4	4
Medium	3	5	15
Complex	5	3	15
Total			34

Одна Function Point=5 годин на одиницю

$$34 * 5 = 170$$

Отже, загальна оцінка зусиль для виконання тестування буде становити 170 годин.