**Лабораторна робота №8-2. Реалізація плану тестування вимог до ПЗ**

**Мета роботи**. Навчитися застосовувати на практиці знання щодо організації проведення тестування.

**Завдання на лабораторну роботу**

1. Ознайомитись з теоретичними відомостями, необхідними для виконання роботи.

2. На етапі створення специфікації вимог до програмного забезпечення системи потрібно проводити тестування вимог. Для організації цієї роботи для вашого проекту потрібно розробити специфікацію вимог і план реалізації тестування вимог на основі цієї специфікації.

3. Оформити звіт.

4. Відповісти письмово на контрольні запитання.

4. Висновки щодо цілей, які досягаються через проведення тестування .

**Контрольні запитання**

1. Через сукупність яких UML-діаграм Ви можете визначити вимоги (уточніть які саме)?
2. Для яких супроводжуючих документів до ПЗ важлива специфікація вимог?
3. Чому важливе тестування вимог?

Результати надсилати на електронну адресу викладача [**t.i.lumpova@gmail.com**](mailto:t.i.lumpova@gmail.com)у вигляді текстових файлів з іменем у форматі **QA<Номер групи><Номер лекції / лабораторної>[-<Номер завдання>][літера позначення типу роботи L – лекція, R - лабораторна]<Прізвище англійською>**. Наприклад, **QA4112R**buts.doc. Відповіді повинні бути не довгими і змістовними. Не копіюйте фрагментів з різних інформаційних джерел, подумайте і викладіть свою точку зору. При наявності відповідей-"близнюків" відповідь буде зараховуватися першому за часом надсилання.

**Строк відсилки ЛР ІПЗ-41 17.03.2024**

**ТЕОРЕТИЧНІ ВІДОМОСТІ**

**Специфікація вимог** – це документ, в якому міститься набір вимог до програмного продукту. Вимоги структуруються та описують логіку роботи продукту (функціональні вимоги), його зовнішній вигляд (користувацький інтерфейс), обмеження в розробці, а також нефункціональні вимоги. Для опису функціональних вимог часто використовуються користувацькі сценарії (use cases). У користувацьких сценаріях представлені варіанти того, як користувач може взаємодіяти з ПЗ. Нефункціональні вимоги описують обмеження пов'язані з дизайном продукту та його реалізацією (продуктивність продукту, безпека, надійність, сумісність, проєктні обмеження, стандарти якості).

З яких основних розділів складається специфікація вимог? Документ має стандартну структуру, яка може відрізнятися від проєкту до проєкту, але в цілому виглядає наступним чином:

1. Introduction
2. Overall Description
3. Functional requirements
4. System Features
5. Interface Requirements
6. Other Nonfunctional Requirements
7. Other Requirements

Незважаючи на те, що наявність документально зафіксованих вимог вже є критерієм якості розробленого продукту, самі вимоги можуть бути описані недостатньо добре. Погано описані вимоги згубно позначаються на розумінні всіх членів команди розробки того, який продукт повинен вийти в результаті. Нечіткі вимоги можуть бути інтерпретовані членами команди по-різному, що спричинить за собою ситуацію, коли одна задача реалізується декількома фахівцями одночасно, абсолютно різними способами та в іншому вигляді. Це, звичайно, призводить до виникнення помилок в продукті, чого б не сталося, якби і самі вимоги були уточнені, протестовані та виправлені.

Хороші вимоги визначаються згідно наступних критеріїв якості:

**Атомарність**

Відповідно до цього критерію, вимоги повинні бути описані так, щоб їх не можна було уточнити ще детальніше або розбити одну вимогу на декілька. Приклад поганої вимоги – користувач може зареєструватися та додати у профіль особисту інформацію. Реєстрація та додавання інформації це дві різні функції, які повинні бути описані і уточнені в різних пунктах специфікації. Приклад хорошої вимоги – користувач може підписатися на оновлення інших користувачів.

**Завершеність**

Вимоги до одного функціоналу повинні бути описані в одному пункті специфікації. Не можна допускати ситуацію, коли один і той самий функціонал описаний в різних частинах документа.

**Послідовність**

Вимога не повинна суперечити іншим вимогам та обмеженням системи.

Наприклад: із соціальних мереж повинен запитуватися телефонний номер користувача. Очевидно, що дана вимога суперечить обмеженням соціальних мереж, так як номер телефону є прихованою інформацією, яку соціальні мережі не надають. Користувач повинен сам ввести номер телефону, щоб система його отримала. Подібних протиріч не повинно бути і між пунктами специфікації. Наприклад, одна і та ж кнопка повинна називатися однаково у всьому документі.

**Відстеження (Трасування)**

Критерій, який дозволяє зрозуміти, чому було прописано саме таку вимогу. Наприклад, для сайту з продажу алкогольних напоїв вимогою може бути реєстрація тільки повнолітніх користувачів у зв'язку із законодавчими обмеженнями.

**Актуальність**

Критерій, який перевіряє відповідність вимог до сучасних реалій (наприклад, із законодавчого/технічного боку або з боку зручності використання продукту, актуальності його користувацького інтерфейсу). Не варто прописувати у вимогах застарілі браузери або застарілі мобільні пристрої та їх ОС, тому що вони практично не використовуються.

**Здійсненність**

Перевірка на те, що вимогу можливо реалізувати за допомогою актуальних існуючих на даний момент технологій. Наприклад, вимога того, що система повинна давати швидкий відгук (тут потрібна конкретизація – не довше, ніж 3 секунди) при помірному навантаженні (тут також потрібна конкретизація за кількістю користувачів та виконуваних ними дій) може існувати. Але вимога відгуку менше секунди при дуже великому навантаженні – нездійсненна в сучасних реаліях вимога.

**Зрозумілість (доступність)**

Вимога має бути сформульована  достатньо чітко, конкретно та однозначно, для того, щоб вона однаково розумілася всією командою розробки.

**Верифікованість**

Критерій, за яким визначається, чи можна порівняти готовий продукт з вимогою і перевірити, чи виконується вона. Якщо вимога описана дуже розмито, перевірити її не вийде. Наприклад: фото повинно завантажуватися швидко/налаштування профілю повинні бути інтуїтивно-зрозумілими/оформлення замовлення повинно відбуватись легко. Для того, щоб перевірити такі вимоги, явно необхідні уточнення, як це швидко/інтуїтивно-зрозуміло/легко.

**Обов'язковість**

Визначення того, наскільки важливе та пріоритетне виконання даної вимоги. Всім вимогам в специфікації повинні бути призначені важливість, стабільність та пріоритет. За допомогою цього вибираються першочергові значення для виконання командою розробки завдання.

**Ідентифікація**

У кожної вимоги повинен бути унікальний ідентифікатор, за допомогою якого вимога прив'язується до інших артефактів проєкту. Наприклад, в описі тест-кейса вказується ідентифікатор вимоги, яка перевіряється цим тест-кейсом.

**Модифікація**

Вимоги повинні бути легко модифіковані (внесення змін) за необхідності.

**Повнота**

Найскладніший критерій, згідно з яким вимоги повинні вичерпно описувати весь функціонал системи. Все, що система повинна виконувати, повинно бути зафіксовано у вимогах, інакше можна зробити серйозну помилку під час проєктування архітектури. Складність у тому, що на початкових етапах проєктування системи дуже важко точно вказати та описати всі функції. Вимоги спочатку описуються більш загальними твердженнями, а далі уточнюються.

#### Як тестуються вимоги?

Тестування вимог – це необхідний етап, що дозволяє поліпшити їх шляхом уточнення, деталізації, а також забезпечити взаєморозуміння між членами команди та уникнути різного трактування. Окрім цього, тестування вимог допоможе зрозуміти, чи можуть вони бути реалізовані загалом (чи достатньо ресурсів, часу, бюджету або можливо це з точки зору технологій).

**Існують наступні техніки тестування вимог:**

1. ***Взаємний перегляд.*** Підрозділяється в свою чергу на:

*Побіжний перегляд* – автор вимог надає документ на швидкий перегляд колегам, які дають свої зауваження, рекомендації, ставлять питання у формі простого неформального обговорення.

*Технічний перегляд* – вимоги надаються автором на перегляд групі фахівців.

*Формальна інспекція* – вимоги проглядаються великою групою фахівців із документуванням всіх зауважень.

2. ***Ставити запитання.*** Якщо під час вивчення вимог виникають запитання – всі суперечливі моменти уточнюються у досвідчених колег або замовника.

3. ***Верифікація вимог.*** Для можливості перевірки можна спробувати створити чекліст або тест-кейс для конкретної вимоги. Якщо виходить швидко придумати перевірки для чекліста або тест-кейса – це вже непогано.

4. ***Уявити поведінку реалізованої системи*** – роботу користувача із системою, створеною за тестованими вимогами. Можливо вийде помітити незрозумілі або неоднозначні моменти в роботі з системою.

***Графічна візуалізація та прототипування*** – зображення інформації у вигляді малюнків, схем, створення прототипу системи (користувацького інтерфейсу) допомагає краще аналізувати інформацію у специфікації, знаходити невідповідності та неточності.

Крім того, що необхідно добре протестувати вимоги та знайти неточності, потрібно правильно сформулювати і залишити зауваження до вимог. На які помилки при складанні зауважень слід звернути увагу і чого слід уникати:

* Не змінювати формат файлу з документацією. Не потрібно видаляти або змінювати вихідний текст, потрібно залишати коментарі до тексту або пропонувати правки. Документ повинен залишатися в форматі, придатному для редагування (TXT/Excel/DOC), формат .pdf або картинки не підійдуть.
* Відзначати в коментарях слід тільки проблемні місця. Вимоги, які сформульовані добре, ніяк відзначати не потрібно. Це лише ускладнює роботу із зауваженнями, оскільки серед усіх коментарів доведеться вибирати ті, які потрібно виправляти.
* Не описувати одне і те ж зауваження в декількох місцях. Тут працює той же критерій якості, що і з самими вимогами. Якщо є необхідність писати одне і те ж зауваження до однієї і тієї ж інформації, краще це зауваження винести в кінець документа та перерахувати список пунктів, до яких воно відноситься. А в самих пунктах робити посилання на зауваження в кінці документа.
* Точно вказувати місце в тексті, до якого відноситься зауваження. Не варто виділяти весь абзац, якщо зауваження стосується до одного речення. У тому ж Word можна виділити потрібну частину тексту та написати коментар саме до неї.
* Якщо необхідно додати уточнююче запитання, його слід формулювати дуже точно та продумано. Наприклад, між запитаннями «Що таке налаштування за замовчуванням?» та «Які налаштування за замовчуванням?» велика різниця. Друге уточнює важливу для специфікації інформацію, а перше – абсолютно безглузде та некомпетентне.
* Не варто писати дуже довгі коментарі, короткий чітко сформульований та структурований текст без орфографічних помилок сприймається набагато легше.
* Не писати зауваження у вигляді критики тексту або його автора, тільки конструктивні коментарі. Категоричні зауваження в стилі «це реалізувати неможливо» також потрібно доводити.
* Не редагувати самостійно вимоги без узгодження. Вносити правки у специфікацію можна тільки після узгодження з відповідальними особами. Інакше може виникнути вкрай серйозна ситуація, коли щось в продукті реалізоване не так, як планувалося, через неузгоджені зміни у вимогах.

Дуже доступно про вимоги та їх тестування написано у книзі Святослава Куликова «Тестування програмного забезпечення. Базовий курс».

Специфікація вимог – важливий документ, який в ідеалі повинен бути описаний для кожного продукту. Добре описані вимоги спрощують роботу всієї команди розробки – програмістам легше зрозуміти, як повинен бути реалізований продукт і як правильно спроєктувати архітектуру, а для тестувальників, крім усього іншого, це ще й джерело очікуваного результату при перевірці. Вміння добре писати та тестувати вимоги – показник компетентності та професіоналізму.

**Загальний аналіз і підготовка вимог**

Вимоги до програмним продуктам (ПП), що розробляються, визначаються на цьому етапі. Для більшості проектів цей крок є дуже важливим. На даному етапі створюється спільна група проекту та призначаються Менеджери проекту як з боку розробника, так і з боку клієнта. Причому вже на цій стадії проекту, окрім бізнес-аналітиків, працюють усі категорії персоналу команди, що бере участь у проекті.

На даному етапі укладається договір на передпроектне дослідження, яке завершується створенням технічної документації, що складається з Проектного завдання (Technical Design Document (TDD), Плану проекту (Project Plan) та Тест-плану (Test Plan), а також пропонуються терміни та вартість виконання робіт.

У ході робіт, проводиться детальний аналіз функціональних вимог, існуючого ІТ ландшафту, безпеки, питань міграції даних, інтеграції майбутнього рішення із зовнішніми підсистемами, підбір відповідних технологічних інструментів для розробки, проводиться аналіз термінів, вартості, ліцензійної складової проекту, можливості подальшої модернізації, варіанти та вартість супроводу.

Після завершення аналізу вимог та затвердження технічної документації, команда проекту може перейти до наступного етапу розробки – дизайну продукту.

**Що таке перевірка вимог?**

Перевірка вимог — це процес підтвердження того, що системні вимоги містять усі необхідні елементи добре написаних вимог. Перевірка вимог є критично важливим кроком у розробці системи, оскільки вона допомагає переконатися, що продукт відповідає своїм цілям і функціям за призначенням.

Перед проектуванням вимоги мають бути перевірені та затверджені, щоб запобігти переробці. Якщо критерії якості не перевіряються, перевірка вимог неминуче буде виконана під час розробки та створення продукту. Відсутні або неправильні вимоги можуть призвести до того, що продукти не відповідають очікуванням клієнтів. Перевірку вимог важливо проводити на ранній стадії та часто, щоб запобігти цим проблемам.

**Важливість перевірки вимог:**

Основними цілями перевірки вимог є забезпечення повноти, правильності та узгодженості системних вимог.

Цей етап може виявити відсутні вимоги, неоднозначні чи недійсні, зменшуючи переробку та перевитрати коштів. Набагато ефективніше вирішити невелику проблему заздалегідь, ніж у майбутньому, коли потрібно буде відстежити та виправити сотні рядків коду або повністю виготовлений складний продукт.

Перевірка вимог необхідна, оскільки вона допомагає переконатися, що система, яка буде побудована, відповідатиме своїм цілям і функціям, як це передбачено. Неповні, неправильні або суперечливі вимоги можуть призвести до проблем під час розробки, тестування та розгортання системи.

**Різниця між верифікацією та валідацією:**

Люди часто плутають верифікацію та валідацію. Насправді вони не однакові.

Згідно з глосарієм інженерної термінології IREB:

Валідація — це процес підтвердження того, що елемент (система, робочий продукт або його частина) відповідає потребам зацікавлених сторін. У розробці вимог валідація – це процес підтвердження відповідності задокументованих вимог потребам зацікавлених сторін; іншими словами: **чи вказано правильні вимоги**.

Верифікація — це процес підтвердження того, що елемент (система, робочий продукт або його частина) відповідає його специфікації. Перевірка вимог – це процес підтвердження того, що вимоги були належним чином задокументовані та задовольняють критерії якості вимог; іншими словами, **чи правильно визначено вимоги**.

Простіше кажучи, перевірка вимог — це процес підтвердження того, що системні вимоги містять усі необхідні елементи добре написаних вимог. Перевірка вимог — це процес підтвердження відповідності письмових вимог запитам зацікавлених сторін.

Іншими словами, під верифікацією йдеться про перевірку того, чи вимоги є повними, правильними та послідовними. Перевірка полягає в тому, щоб перевірити, чи вимоги описують передбачувані цілі та функції системи.

**Критерії якості під час перевірки вимог:**

Якість вимог має вирішальне значення для успішної розробки програмного забезпечення. Існує кілька наборів критеріїв якості, яким мають відповідати вимоги, наприклад критерії, визначені IREB або правилом SMART. Навчальний план IREB визначає шість критеріїв якості, яким має відповідати кожна окрема вимога, в тому числі *адекватність, необхідність, однозначність, повнота, зрозумілість і перевіреність*. Крім того, набір вимог повинен відповідати іншим критеріям якості, включаючи послідовність, відсутність надлишків, повноту, можливість модифікації, відстежуваність і відповідність.

Важливо вибрати найбільш релевантний набір критеріїв якості для проекту та переконатися, що вимоги їм відповідають. Однак не всі вимоги повинні однаково відповідати всім критеріям якості. Замість цього цінність кожної вимоги слід розглядати в контексті проекту, а відповідні критерії якості мають бути відповідно пріоритетними. Перевіривши відповідність вимог відповідним критеріям якості, можна зменшити ризик невдач у розробці програмного забезпечення.

**Методи, що використовуються для перевірки вимог:**

Існує кілька методів, які можна використовувати для перевірки вимог, щоб переконатися, що вимоги відповідають необхідним критеріям якості. Деякі з найбільш часто використовуваних методів включають:

1. **огляд:** Ця методика передбачає систематичний перегляд вимог групою експертів для виявлення будь-яких помилок, упущень або невідповідностей. Його можна проводити вручну або за допомогою автоматизованих засобів.
2. **Тестування:** Тестування передбачає розробку та виконання тестів для перевірки того, що вимоги відповідають бажаній функціональності та критеріям якості. Його можна проводити на різних рівнях, таких як модульне тестування, інтеграційне тестування та приймальне тестування.
3. **Покрокове керівництво:** Під час покрокового керівництва вимоги переглядаються групою зацікавлених сторін, які надають відгуки та визначають будь-які проблеми чи проблеми. Зазвичай це менш формально, ніж перевірка.
4. **Прототипування:** Створення прототипу передбачає створення спрощеної версії програмного забезпечення для підтвердження вимог і виявлення будь-яких проблем або обмежень. Це може допомогти зацікавленим сторонам візуалізувати та краще зрозуміти систему.
5. **Моделювання:** Моделювання передбачає створення моделі системи та тестування її поведінки за різними сценаріями. Це може допомогти виявити проблеми з вимогами, які можуть бути неочевидними в статичній документації.
6. **Аналіз простежуваності:** Аналіз відстежуваності передбачає відстеження зв’язків між вимогами та іншими артефактами, такими як проектні документи та тестові випадки, щоб переконатися, що вимоги є повними, послідовними та такими, що можна перевірити.

Ці методи можна використовувати окремо або в комбінації, щоб перевірити відповідність вимог необхідним критеріям якості та зменшити ризик помилок і невідповідностей у кінцевому програмному продукті.

Перевірка вимог — це процес, який використовується для того, щоб переконатися, що вимоги до системи або продукту відповідають набору критеріїв якості, який гарантує, що вони були належним чином задокументовані. Важливість перевірки вимог неможливо переоцінити, оскільки вона може допомогти запобігти дорогим помилкам і затримкам у майбутньому. Існує ряд інструментів і методів, які можна використовувати для перевірки вимог, і Visure Requirements ALM Platform є одним із таких інструментів. Завдяки потужним функціям та інтуїтивно зрозумілому інтерфейсу Visure Requirements ALM Platform може допомогти вам забезпечити виконання вимог вашого проекту вчасно та в межах бюджету.

# Тестування на основі вимог

## Що таке тестування на основі вимог

Тестування на основі вимог — це тип тестування, який оцінює, чи відповідає система функціональним і нефункціональним вимогам, визначеним замовником. Основна мета тестування на основі вимог — визначити, чи відповідає розроблений програмний продукт усім поставленим цілям чи ні. Це гарантує відсутність розбіжностей між тим, що було зазначено, і тим, що було доставлено. Цей підхід до тестування допомагає виявити будь-які невідповідності, пов’язані з продуктивністю, надійністю, масштабованістю, зручністю використання та безпекою на ранній стадії розробки. Тести на основі вимог можна розділити на тести чорного ящика (які перевіряють, наскільки добре програма працює відповідно до очікувань користувача) і тести білого ящика (які перевіряють кожен аспект коду). Тести, проведені з використанням цього підходу, повинні включати як позитивні, так і негативні тести. Тестерам важливо добре розуміти вимоги, щоб розробляти ефективні тести та сценарії. Тестування на основі вимог є важливою частиною процесу тестування програмного забезпечення, яке допомагає розробникам створювати якісні продукти.

Використовуючи тестування на основі вимог, організації можуть гарантувати, що вони постачають програмне забезпечення високої якості відповідно до очікувань клієнтів. Такий підхід підвищує довіру користувачів до продукту, а також зменшує загальні витрати, пов’язані з розробкою та обслуговуванням. Результати цього типу тестування можна використовувати для виявлення будь-яких слабких ланок у системі та вжиття відповідних заходів для виправлення. Крім того, гарантуючи, що розроблений продукт відповідає всім визначеним вимогам, компанії можуть уникнути дорогої переробки або редизайну в майбутньому. Таким чином, організаціям важливо використовувати тестування на основі вимог для створення якісних продуктів.

## Важливість тестування на основі вимог

Тестування на основі вимог відіграє важливу роль у тестуванні програмного забезпечення та допомагає організаціям створювати якісні продукти, які відповідають потребам їхніх клієнтів. Він використовується для перевірки того, що всі вказані вимоги належним чином реалізовані в системі, таким чином забезпечуючи високий рівень гарантії функціональності та зручності використання продукту. Такий підхід до тестування гарантує, що будь-які зміни, внесені в систему, відповідають її оригінальним специфікаціям дизайну, допомагаючи розробникам уникнути дорогих переробок або редизайну в майбутньому. Тестування на основі вимог також допомагає тестувальникам виявляти будь-які розбіжності між очікуваними та фактичними результатами на ранніх стадіях розробки, заощаджуючи таким чином час і гроші. Крім того, цей тип тестування підвищує довіру користувачів до продукту, гарантуючи, що він відповідає їхнім потребам і очікуванням.

Загалом, тестування на основі вимог є важливою частиною процесу тестування програмного забезпечення. Це допомагає організаціям створювати якісні продукти, які відповідають усім вимогам і очікуванням клієнтів. Використовуючи цей підхід, компанії можуть заощадити час і гроші, уникаючи дорогих переробок або редизайну в майбутньому. Крім того, це підвищує довіру користувачів до продукту, гарантуючи, що він відповідає їхнім потребам і очікуванням. Таким чином, тестування на основі вимог є важливим елементом будь-якого успішного проекту розробки програмного забезпечення.

## Кроки для тестування на основі вимог

1. **Визначте системні вимоги:** Першим кроком у цьому типі тестування є розуміння та визначення системних вимог. Важливо чітко визначити всі функціональні та нефункціональні вимоги, які необхідно протестувати перед початком будь-якої діяльності з тестування.
2. **Розробка тестових випадків:** Після визначення системних вимог на їх основі можна створювати тестові приклади. Ці тестові випадки повинні охоплювати як позитивні, так і негативні сценарії; вони також повинні включати граничні значення, граничні випадки тощо для ретельного тестування програми.
3. **Виконати тести:** Створивши відповідні тестові приклади, тестувальники можуть розпочати їх виконання та записувати свої результати, наприклад статус «пройшов/не пройшов» або будь-які проблеми, що виникли під час виконання.
4. **Аналізуйте результати:** Після того, як тести виконані, тестувальники повинні проаналізувати їх результати, щоб виявити будь-які невідповідності в системі. Цей аналіз допомагає з’ясувати, чи відповідає система вимогам замовника чи ні.
5. **Переробити та повторно перевірити:** Якщо під час тестування на основі вимог виникнуть будь-які проблеми, їх слід негайно виправити та перевірити повторно, щоб переконатися, що вони не впливають на загальну продуктивність системи.

Тестування на основі вимог є важливою частиною розробки програмного забезпечення, і його слід регулярно використовувати для виявлення будь-яких проблем із продуктом перед його випуском у виробництво. Це допомагає організаціям гарантувати, що їхні клієнти отримають високоякісні продукти, які відповідають усім встановленим вимогам. Дотримуючись відповідних кроків для тестування на основі вимог, компанії можуть переконатися, що вони надають своїм клієнтам якісний продукт.

## Тенденції тестування на основі вимог

У сучасному світі розробки програмного забезпечення гнучкі процеси, такі як Scrum, спринти та планування, відкинули тестування на основі вимог на користь підходу Waterfall.

У Waterfall статичні терміни та встановлені етапи є вирішальними для успішного виконання; навпаки, Agile наголошує на важливості якнайшвидшого надання цінності клієнту з дуже невеликою кількістю попередніх умов. Замість того, щоб обтяжувати кожен етап надто багатьма вимогами, потрібно лише те, що необхідно для успішного просування вперед.

У наші дні команди з розробки програмного забезпечення обирають тестування на основі вимог набагато рідше, ніж детально-орієнтований метод Waterfall.

Підсумовуючи, тестування на основі вимог є важливою частиною процесу розробки програмного забезпечення, яка допомагає компаніям створювати програми, які відповідають очікуванням і вимогам клієнтів. Цей тип тестування забезпечує високу загальну якість продукції шляхом виявлення будь-яких функціональних або нефункціональних невідповідностей на ранній стадії. Використовуючи цей підхід, організації можуть зменшити витрати, пов’язані з розробкою, обслуговуванням і переробкою в майбутньому. Тому компаніям настійно рекомендується використовувати тестування на основі вимог як частину життєвого циклу розробки програмного забезпечення.