

SUPER-PUPER ШПАРГАЛКА ДЛЯ QA НАЙПОШИРЕНІШИХ ЗАПИТАНЬ НА

HOME ➔ ТЕСТУВАННЯ ПЗ ➔ SUPER-PUPER ШПАРГАЛКА ДЛЯ QA З ТЕСТУВАННЯ — 1С



Ольга



11 коментарів

Ми постаралися зібрати для Вас тут найпоширеніші запитання з теорії тестування, які зазвичай зустрічаються при працевлаштуванні — як початківцям, так і вже доволі досвідченим тест-інженерам. Одразу ж

За допомогою цієї шпаргалки Вам буде добре => шляхом повторення закріпити знання та швидко освіжити в пам'яті свої знання із теорії тестування, для того щоби успішно підготуватися до

Розпочнемо ми разом із простіших питань теорії тестування і поступово перейдемо до запитань

Для зручності велику кількість питань розділяємо на 8 частин (планується). І Ви теж можете в

Питання #1 Що таке тестування програмного забезпечення?

Тестування програмного забезпечення — це процес оцінки та перевірки певної системи ПЗ станом системи, бізнес вимогами та очікуваною поведінкою. Тестування програмного забезпечення з точки зору: правильності, повноти, зручності використання, продуктивності тощо для заданих

Питання #2 Чому потрібне тестування програмного забезпечення?

Подумати логічно... тестувати програмне забезпечення варто з наступних причин:

1. Тестування дозволяє перевірити, чи правильно реалізовані усі вимоги до ПЗ, що розроблялося.
2. Тестування також демонструє, що створене ПЗ працює відповідно до специфікацій і відповідає вимогам.
3. Тестування гарантує зацікавленим сторонам, що продукт працюватиме так як передбачено і яким очікується.
4. Тестування допомагає у виявленні дефектів, помилок. Та забезпечує розпізнавання критичних дефектів програмного забезпечення.
5. Тестування допомагає перевірити належну інтеграцію та взаємодію кожного компонента в системі.
6. Фактично, тестування економить час розробки ПЗ, виявляючи проблеми та вузькі місця ще на ранніх етапах.
7. Як правило: дефекти, виявлені на початкових фазах SDLC, призводять до меншої вартості та швидкості випуску.
8. Тестування запобігає потраплянню неякісного програмного продукту до кінцевого користувача.
9. Команда із тестування, на відміну від команди розробників дивиться під іншим кутом зору на продукт.

Питання #3 Роль тестування в розробці програмного забезпечення

Отже, **Чому ж тестування настільки важливе?**

Код пишуть люди, а людям властиво помилятися. По статистиці: на кожну тисячу рядків коду багато часу розробки та налагодження ПЗ. Окрім того наслідки «дефектного» програмного забезпечення можуть бути катастрофічними, і навіть смертельними. Тому тестування є одним із найважливіших та найвідповідальніших етапів розробки, а й запобігти їх виникненню у майбутньому!

Питання #4 Отже, які головні цілі тестування?

- Запобігання дефектів;
- Виявлення дефектів;
- Підвищення впевненості у рівні якості програми;
- Надання інформації зацікавленим сторонам для прийняття рішень.

Питання #5 Коли слід припинити тестування?

ПАМ'ЯТАЙТЕ: Тестування програмного забезпечення, як ремонт у будинку — процес нескінченний. Тестування повинно бути зупинено, коли виконується одна або декілька з наступних умов:

1. **Після виконання тестового сценарію** — коли повний цикл тест-кейсів пройдено і останні трекінгові системи не критичні і узгоджені зі значенням допустимого відсотка.
2. **Після завершення кінцевого терміну перевірки (дедлайну)** — за умови, що у системі не знайдено критичних помилок.
3. Програма настільки «забагована» серйозними недоліками, що **подальше тестування** просто неможливе.
4. **Ми зупиняємося, коли нам щось нестерпно заважає!** Немає потрібної інформації, в системі хочемо протестувати. Ми не маємо потрібного устаткування або необхідних інструментів спеціалізовані тести.
5. **Основні баги знайдені, шукати далі економічно не вигідно.** Проте іноді трапляється ситуація, коли виявлення нових проблем не скасує релізу, тому що б ми не знайшли, це не має жодного значення.
6. **На основі рішення зацікавлених сторін.**
7. **Виходячи з коефіцієнта охоплення коду** — тестування може бути зупинено, коли охоплення коду достатніми прохідними показниками, без критичних помилок.

Питання #6 Що таке Забезпечення Якості (Quality Assurance)?

Quality Assurance (Забезпечення Якості) — це технологічний підхід, який перевіряє, чи є процес розробки програмного забезпечення відповідним до вимог. Згідно цього підходу **Quality Assurance** — це система запобіжних заходів, визначити слабкі місця процесу розробки програмного забезпечення. Quality Assurance включає в себе такі заходи, як огляд документації, побудова тест дизайну: планів та розробка тестів, додатковий аналіз тощо.

Питання #7 Що таке контроль якості?

Quality Control (Контроль якості) — це продуктивний підхід. Порівнюючи з попереднім підходом, Quality Control — це перевірка продукту на відповідність до зазначеним вимогам — тобто банальна перевірка на баги, дефекти. Quality Control включає в себе такі заходи, як тестування, дослідницьке тестування, тестування продуктивності, usability тестування та ін.

Питання #8 Опишіть характерні відмінності між Quality Assurance

Quality Assurance включає в себе Quality Control поряд з іншими процесами щодо поліпшення якості. **Quality Assurance (забезпечення якості)** — це превентивний процес, завданням якого є забезпечення якості продукту. Quality Assurance більш орієнтований на процес.

Quality Control (контроль якості) — це процес знаходження помилок в продукті, з метою їх подальшого усунення. Quality Control орієнтований на якість продукту, що розробляється. Говорячи іншими словами, Quality Assurance гарантує, що процес поставлений правильно і дає продукт задовольняє вказаному набору вимог.

ЧИТАЙТЕ роз'яснення з життєвими прикладами у дописі: [Quality Assurance у кожен бізнес](#)

Також **ЗВЕРНІТЬ УВАГУ!** на поняття:

(Quality Management) або Менеджмент Якості — скоординована діяльність з керівництва та контролю, яка включає в себе розробку політики у сфері якості та цілей у сфері якості, планування яким чином досягти поліпшення якості чи забезпечення стабільної якості. [ISO 9000-2011; Глосарій стандарту терміни]

Total Company Quality

QA

QC

Питання #9 Яка різниця між Verification (Верифікація) and Validation

#	Verification	Validation
1.	Верифікація — процес оцінки системи і її компонентів з метою співставлення результатів поточного етапу розробки, початково сформованим умовам. Тобто, виконання заданих завдань, цілей, строків по розробці продукту та відповідність стандартам.	Валідація — це процес перевірки того, що розроблене програмне забезпечення відповідає визначеним бізнес-вимогам та потребам користувачів.
2.	Це статичний процес аналізу, в першу чергу документації, а не фактичного кінцевого продукту.	Передбачає динамічне тестування програмного продукту шляхом його запуску і різноманітних перевірок.
3.	Верифікація — це процесно орієнтований підхід.	Перевірка — продуктово орієнтований підхід.
4.	Відповідає на запитання: “Чи правильно ми будуємо продукт?”	Відповідає на запитання: “Ми на шляху побудови правильного продукту?”
5.	Помилки виявлені під час фази Верифікації вимагають на правки меншої вартості коштів/витрачених ресурсів на фіксацію порівняючи з вартістю помилок знайдених під час фази Валідації .	Помилки виявлені під час фази Валідації вимагають більших витрат вартості / ресурсів / часу. А ще пізніша помилка ця ж сама помилка може коштувати ще дорожче.

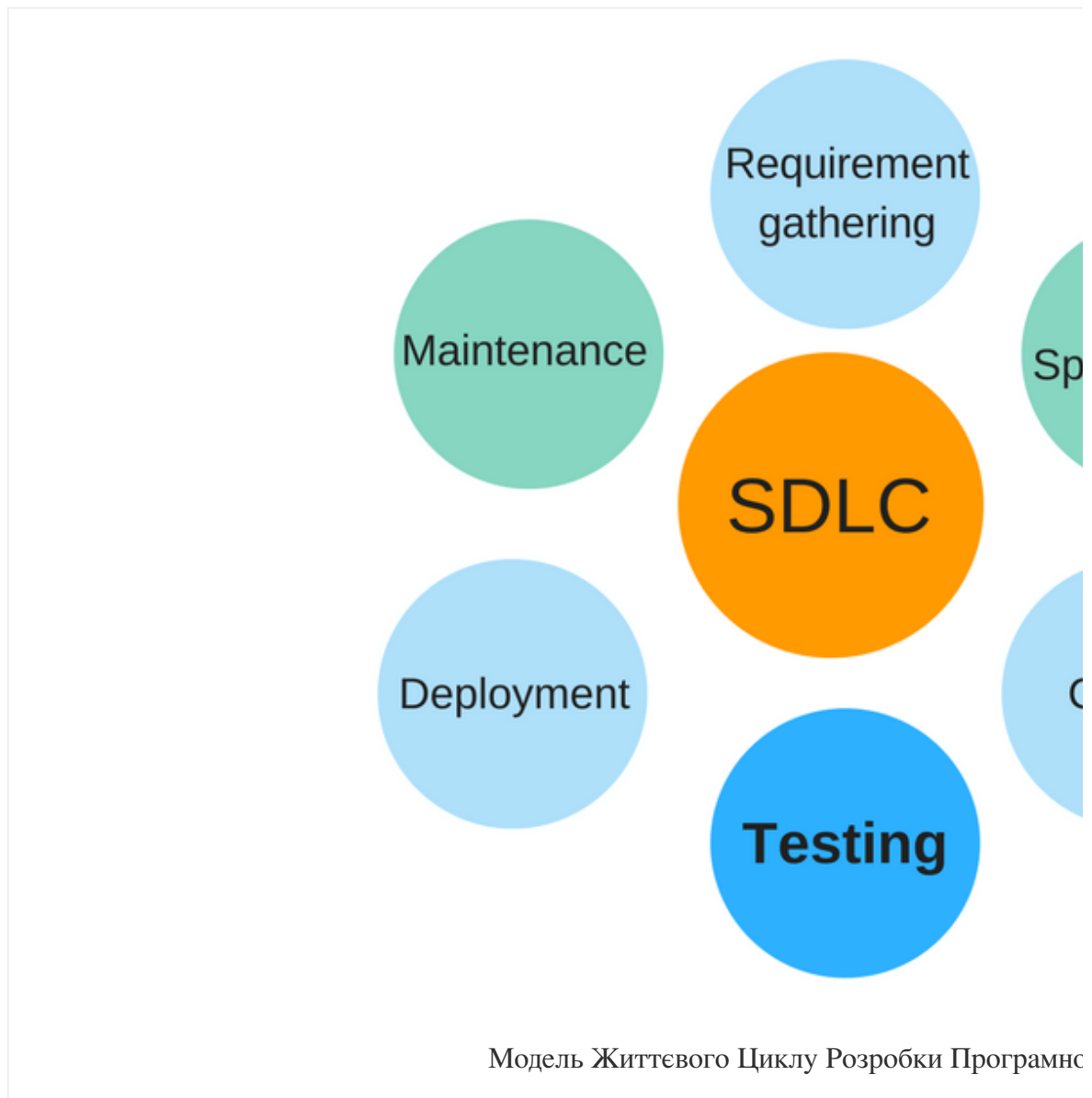
Питання #10 What is SDLC?

SDLC (Software Development Life Cycle) — Життєвий Цикл Розробки Програмного Забезпечення

Здебільшого у кожній моделі Життєвого циклу Розробки Програмного Забезпечення виділяють ці

1. Розробка та аналіз вимог;
2. Проектування;
3. Впровадження (реалізація);
4. Тестування;
5. Розгортання;

6. Тримання обслуговування



Як схематично зображено на ілюстрації, розробка будь-якого програмного забезпечення неможл

Перший етап є основним напрямком діяльності керівників проектів та замовників. У ході розрс використовуватиме систему, з якими даними вона працюватиме, які функції повинна викон формулюванні вимог до проекту. Після збору вимог здійснюється аналіз і на його основі формує

На другому етапі визначається загальна архітектура системи, формуються її технічні характери об'єкти тестування.

Далі робота умовно розподіляється ще на модулі. Починається найдовша фаза розробки, а саме

Отриманий код перевіряється на відповідність вимогам, щоб переконатися, що продукт дійсно розробки вимог. На цьому етапі проводяться всі типи нефункціональних та функціональних випр

Після того, як окремі програмні модулі готові, вони об'єднуються в єдине ціле. Це ще не і виконувати, хоча б частково, свої основні завдання. Такий варіант програми називають р

Альфа-тестування — це етап налагодження і перевірки альфа-версії програми. Це можуть

...тестування — це етап напередодні і перевірки аналізу версії програми, що жоду і договором. У будь-якому разі це кваліфіковані фахівці, які вміють працювати із спеціалізованими методиками.

Після успішного тестування продукт розгортається і передається клієнту. За його бажанням починається **відомого нам альфа-тестування?**

Головне завдання бета-тестів — оцінити можливості і стабільність роботи програми з то запрошуються звичайні люди, які мають досвід роботи з програмами такого типу або, або ще і професійними тестувальниками. Також у ролі бета-тестерів може виступати Support команда за коли на роль бета-тестерів запрошують всіх бажаючих. Тестувальникам початківцям ми реком Linux, Майкрософт, Apple, Selenium та ін. відомих програмних продуктів. Це гарний шанс здобути Якщо на етапі бета-тестування виникає потреба у внесенні змін або були виявлені грубі помилки виправлення помилок відбувається остаточне розгортання.

Коли клієнти починають використовувати розроблену систему, виникають поточні проблеми і їх і до сучасніших версій. Цей процес, в якому увага приділяється підтримці тону розробленого пр

Питання #11 Поясніть STLC – Software Testing life cycle.

STLC (Software Testing life cycle Життєвий цикл тестування програмного забезпечення) —

Фази включають:

Requirement Analysis & Clarification questions — на цьому етапі аналізуються вимоги, зад визначаються обсяги тестування.

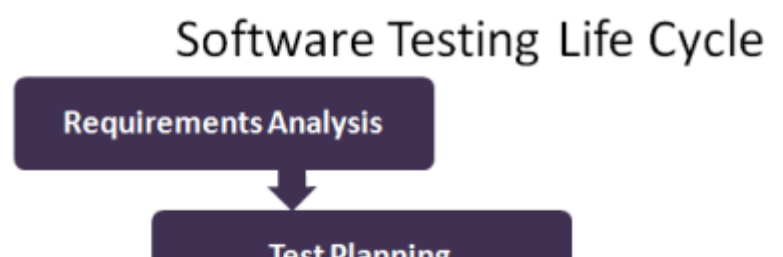
Test planning — цьому етапі визначається стратегія тест-плану, робиться попередня оцін обговорюються можливості застосування автоматизованого тестування.

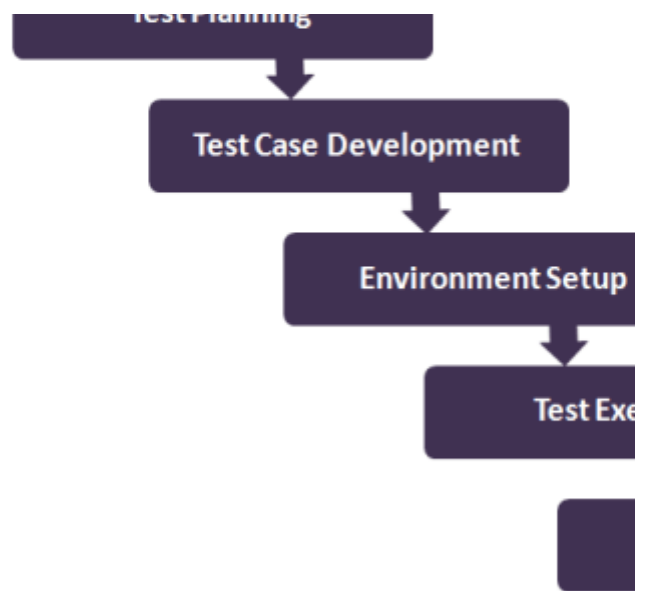
Test Design — тестовий дизайн та аналіз. На цьому етапі розробляються тестова архітектура скрипти імплементуються.

Налаштування тестового середовища — тестове середовище готується заздалегідь, воно модел

Test execution — на цьому етапі виконуються тест-кейси, рапортується про баги та про поми тестування після їх фіксації.

Test closure and reporting — відбувається підбиття підсумків, складається тест-репорт з остат





<http://istqbExamCertification>

Питання #12 Що таке тестове середовище?

Тестове середовище — це тестовий майданчик, котрий використовується для тестування про обладнання та програмного забезпечення випробуваної програми, включаючи операційну сист забезпечення, фреймворки, база даних тощо.

Питання #13 Артефакти тестування:

Артефакт — це деякий штучно створений людиною продукт. Банально... список первісної люд продукти праці тестувальника, які створюються і породжуються у процесі тестування програм і в

Найбільш поширеними тестовими артефактами є:

План тестування (Test Plan) — це документ який описує весь обсяг робіт з тестування, п закінчення тестування, необхідного в процесі роботи обладнання, спеціальних знань, а також оц

Набір тест-кейсів і наборів (Test Case & Test suite) — це послідовність дій, за якою можна пер

Чек Ліст (Check List) — це документ, який описує те що повинно бути протестовано. Потрібен поділу обов'язків.

Чіт-ліст (Cheat Sheet) — це список перевірок, який повторюється.

Баг-репорт (Bug Reports / Defects) — це заключний документ, що описує ситуацію або посліди із зазначенням статистичних даних, обґрунтування причин і очікуваного результату.

У наступних питаннях більш детально ➡

Питання #14 Що таке план тестування?

План тестування (Test Plan) — це офіційний документ, що описує весь обсяг робіт з те

план тестування (test plan) — це офіційний документ, що описує увесь обсяг робіт з тестування, починаючи з опису об'єкта, стратегії, розкладу, критеріїв початку і закінчення тестування, а також оцінки ризиків з варіантами їх вирішення. Тест план розробляється згідно вимог програмного забезпечення), **Software Requirement Specifications** (специфікації програми).

Тест-план у пунктах повинен відповідати на питання:

- Що треба тестувати?
- Що будемо тестувати?
- Як будемо тестувати?
- Коли будемо тестувати?
- Критерії початку тестування.
- Критерії закінчення тестування.

В ідеалі тест-план повинен описувати:

- об'єкт тестування (продукт який тестуємо);
- цілі тестування та скоупи робіт;
- стратегія тестування (як буде проводитися тестування, які інструменти будуть застосовуватися);
- пріоритети проекту і роль тестування для нього;
- процес тестування (коли можна почати тестування, як слід інтерпретувати результати і т.д.);
- опис тестового середовища;
- власне тестові сценарії (Test Scenario), тестові випадки (Test Cases).

Тест-план — це головний документ, яким оперує QA інженер.

Дуже часто в реальності слідувати усім пунктам тест-плану не вдається. Причиною відмови подібних випадках звичайно тест-план обмежується мінімальним описом тестового середовища

Питання #15 Що таке Check List?

Як ми вже згадували вище **Чек Ліст (Check List)** — це документ, який описує що має бути протестовано. На скільки детальним буде **Check List** залежить від вимог до звітності, рівня знання продукту споживачем.

Що має бути зазначено в Check List:

- Список перевірок (з необхідної ступенем деталізації);
- Статус перевірок: (збірка, на якій проводилося тестування, тестове оточення (якщо є, то де), версія продукту, дата тестування).

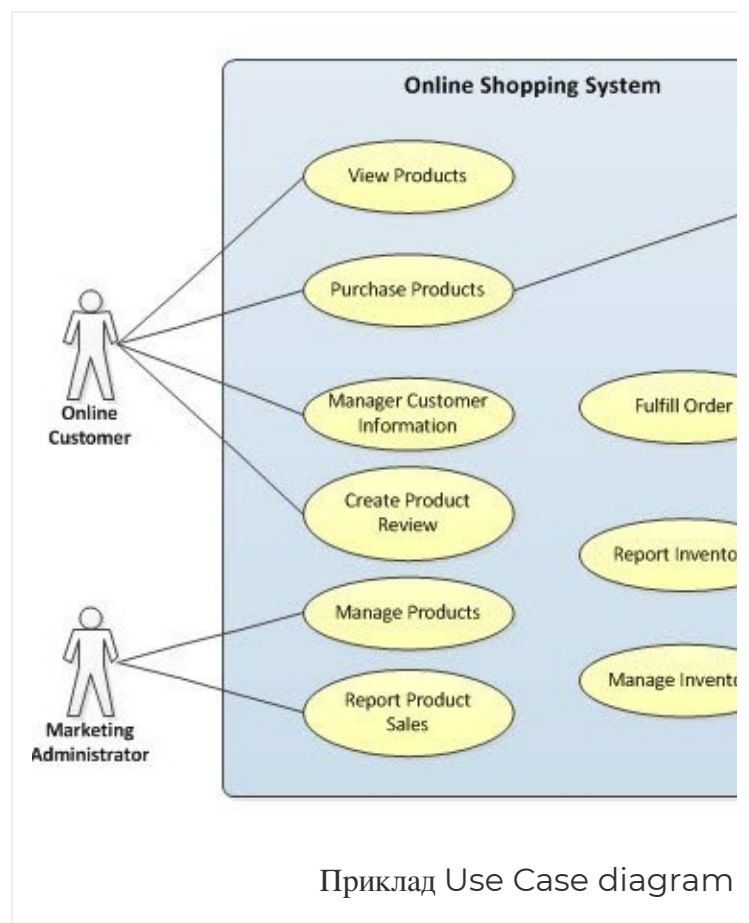
- Результати перевірки.

В цілому, грамотно складені **Check List** -и в значній мірі можуть ефективно справлятися з функціональними випадками раціонально буде витратити час на написання тест-кейсів. Рішення про їх необхідність має бути універсальним для усіх проектів.

Питання #16 Що таке Use Case?

Юзкейс (Use Case) — це перелік дій, сценарій по якому **CAME** користувач взаємодіє з системою для досягнення конкретної мети. Тестування по юзкейсам проводиться для того, щоб виявити додаткові дефекти окремих індивідуальних модулів, частин програми окремо один від одного. Юзкейс тестування може проєктувати

Для кращого сприйняття Use Case часто малюють у вигляді діаграм з переходами:



Питання #17 Що таке тестовий сценарій?

Test Scenario — тестовий сценарій походить від **Use Case**. Суть документу з назвою Тестовий сценарій особливо корисне, коли під час тестування існують часові обмеження.

Питання #18 Що таке Test Case?

Test Case — це сукупність умов з передумовами, вхідними значеннями та очікуваними

За видами тестові випадки бувають:

- 1. **Позитивний тест-кейс** — використовує тільки коректні дані і перевіряє правильність виконани
- 2. **Негативний тест-кейс** — використовує у якості вхідних даних, як коректні, так і некоректні , виняткових ситуацій (спрацьовування валідаторів), а також перевіряє, чи викликається додатком

Питання #19 Які деякі атрибути, структура Test Case?

У спрощеному вигляді у Test Case можуть бути розписані наступні атрибути:

#	Параметри:	Опис:
1.	Test Case Id	унікальний ідентифікатор Test Case
2.	Test Summary	ім'я, короткий зміст чи назва. Назва повинна бут короткою та інформативною, щоб розкривати суть тес кейсу без його повного прочитання.
3.	Description	докладний опис тесту.
4.	Prerequisite condition or pre-	необхідна умова попередня умова або набір передумов, яких слід дотримуватися пере виконанням тестових кроків.
5.	Test Steps	тестові кроки для виконання тесту.
6.	Expected result	очікуваний результат від того, щоб пройти тест.
7.	Actual result	фактичний результат після виконання тестових кроків.
8.	Test Result	статус виконання тесту, мінімально Pass / Fail Blocked / Not Applicable.
9	Priorities	пріоритетність (high, medium, low, normal) тощо
10.	Automation Status	статус автоматизації вказує нам чи програм автоматизована чи ні.
11.	Attachments	додатки картинок екрану, відео, файлів, логів і т.д., щ може бути використане, як наявний доказ поведінк системи.
12.	Comments	важливі коментарі
13.	Reference	посилання на джерела тест кейсу (виз дизайн, інше) а також на інші пов'язані тест і

14.	Date	дата виконання випробувань.
15.	Executed by	Ім'я особи, яка виконує Test Case.
16.	Postponed	відкладений, в коментарі необхідно написати причину

Дізнатися дещо більше про Test Case рекомендуємо тут на сайті, у

Питання #20 Що таке тестовий скрипт?

Тестовий скрипт — це автоматизований Test Case, запущений будь-якою спеціалізованою прог.

Це, в основному, сукупність інструкцій для оцінки роботи програми.

Питання #21 Що таке помилка BUG?

Помилка або помилки в програмному продукті, які викликають його непередбачене функціонування.

Питання #22 Що таке дефект?

Дефект — це невідповідність вимогам продукту (requirements). Вада в компоненті або системі.

Дефект може призводити до відмови (**Failure**) роботи, експлуатації програми її компонента чи системи.

Існує ще така «заковирка» **Помилка з назвою (Error)** — це дія людини, яка призводить до нег.

Гадаю різницю коли Вас запитують Ви зможете пояснити?

Питання #23 Які деякі атрибути звітності про дефекти?

#	Параметри:	Опис:
1.	Defect Id	унікальний ідентифікатор дефекту.
2.	Defect Summary	Короткі відомості про дефекти — короткий опис дефекту в одному рядку, більше схожий на назву дефекту.
3.	Defect Description	докладний опис дефекту.
4.	Steps to reproduce	Шляхи відтворення — кроки до відтворення дефекту.
6.	Expected result	Очікуваний результат — очікувана поведінка, з якої і відхиляється через дефект.

7.	Defect Severity	Виходячи з критичності дефекту, це поле можна встановити на незначну, середню, основну або пробку.
8.	Priority	Виходячи з терміновості дефекту, це поле можна встановити шкалою від P0 до P3.

Питання #24 Які інструменти управління помилками або дефектами?

Найпоширеніші **Баг-трекінгові системи** — Jira, Bugzilla, Redmine, Mantis, Quality Center тощо.

Питання #25 Що таке щільність дефектів?

Щільність дефектів — це показник щільності дефектів у системі. Його можна розрахувати шляхом поділу кількості дефектів на кількість кодів (методів або класів) у програмі або аплікації.

Питання #26 Що таке пріоритет дефектів?

Пріоритетом дефекту є оцінка важливості фіксації дефекту. Зазвичай пріоритет дефекту встановлюється на основі найбільшої терміновості виправлення.

Питання #27 Що таке Priority severity?

Priority severity (Серйозність дефектів) — це ступінь тяжкості дефекту, що впливає на формулювання пріоритету. Він використовується для відрізняти різні ступені дефектів, починаючи від незначних (**minor**) до критичних (**critical**) або повільних.

Питання #28 Наведіть приклади Low priority, High priority-Low, High priority-High severity

- **Low priority-Low severity** (Низький пріоритет / Низька ступінь тяжкості) — орфографічна помилка в тексті.
- **Low priority-High severity** (Низький пріоритет / Висока ступінь тяжкості) — аплікація не може завантажитися.
- **High priority-Low severity** (Високий пріоритет / Низька ступінь тяжкості) — незначні зміни кольору.
- **High priority-High severity** (Високий пріоритет / Висока ступінь тяжкості) — проблема з функціями входу.

Питання #29 Що таке критична помилка BUG?

Критичний Баг — це помилка, яка впливає на основні функціональні можливості програми, що призводить до краху програми. Відрізняється від блокуючого багу (**blocker bug**), оскільки це не впливає або не блокує роботу програми.

Питання #30 Що таке блокатор Blocker bug?

Blocker bug — це помилка високого пріоритету та високого ступеня важкості, яка буквально блокує основної частини програми.

Питання #31 Що таке заглушка у програмі?

Заглушка — це макет ще не реалізованого модуля програми, необхідний при низхідній реакції на виведення вхідних даних, або програма автоматично повертає до вищестоящих модулів. Прояв наявності заглушок — уточнюйте.

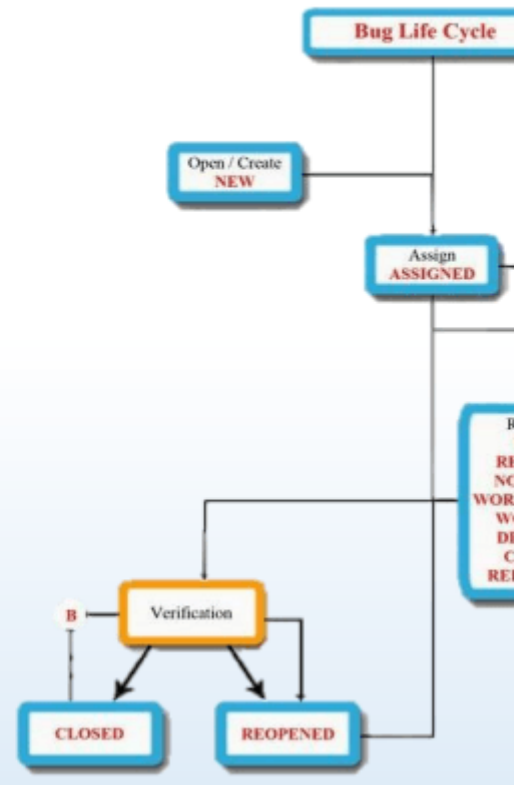
Питання #32 Поясніть життєвий цикл Багу або опишіть різні статуси

Bug при розробці програмного забезпечення проходить наступні етапи:

Опис:	Статус:
New	коли Баг або дефект щойно виявлені, то вони знаходяться в New state у багтрекінговій системі.
Assigned	нещодавно виявлений Баг повертається відповідному розробнику у якого він, наприклад, виник або тому хто повинен його пофіксити.
Open	коли розробник працює над помилкою, Баг перебуває в Open state також позначка означає, що Баг потрібно виправити.
Rejected/Not a bug	Баг відхилено, якщо розробник вважає, що помилки не має або вона не підтверджена, оскільки Баг не змогли відтворити — невалідний Баг.
Deferred	означає, що виправлення Багу відкладається на деякий час (до наступних релізів), виходячи з терміновості та критичності помилки.
Fixed	помилка спричинена розробником, позначається ним як виправлена.
Test	виправлений баг висить за тестером, доки він його не протестить після виправлення. Якщо проблема все ще відтворюється, то відкривається наступний статус, “Наново відкритий”.
Reopened	якщо тестера не задовольняє виправлення, помилка переміщується до статусу повторного відкриття.
Verified	після завершення етапу тестування, якщо тестер відчуває що помилку, вирішено, він буде Баг позначати як підтверджений.
Closed	після того, як помилка перевірена, вона переміщується в статус “Закритий”

Те саме ілюструємо, блок-схема показує основні статуси і можливі переходи від одного до наступного

ЖИТТЕВИЙ ЦИКЛ БАГУ

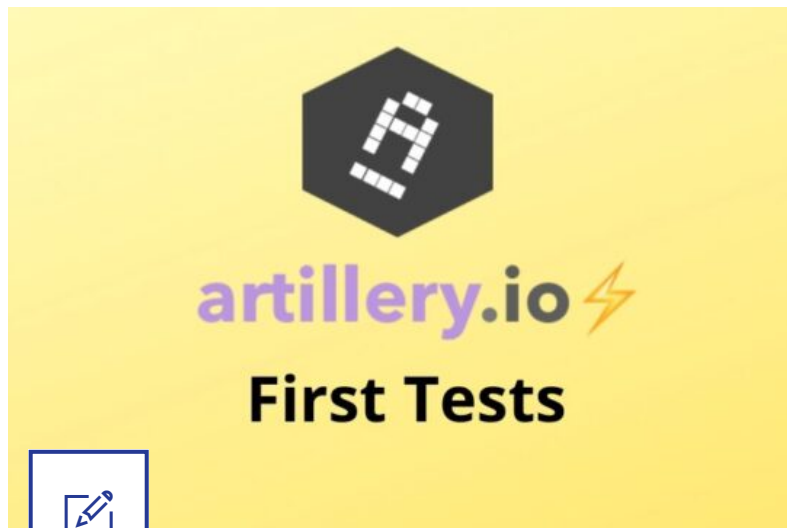


Ілюстрація з презентації Bugs & Bug I

ПОРАДА: намагайтеся, коли працюєте уникати складних схем перех

Share: [f](#) [t](#) [G+](#) [p](#)

Related posts



/ 0/ 0 Олька

/ 0/ 0 Олька

Тестування Навантаження З Artillery — Перші Тести

Поняття Estimate Та Technical I

Існує багато tool -зів для Тестування Навантаження (Performance Testing). Серед них, як...

Зазвичай програмні продукти будують як цеглина за цеглиною. Систематично їх...

11 КОМЕНТАРІВ

Maryna

24 КВІТНЯ, 2019 О 02:52

Доброго дня! Знайшла дефект на сайті, впливає на юзабіліті/ергономіку. Прошу врахувати це, якщо буде ва українською, дякую Вам за увагу!

Bug Report

Summary:Таблиці у мобільній та планшетній версії перегляду не доступні у повній мірі при зміні масштабу

Description:Дані в таблицях наявні не в повній мірі (обрізані) при перегляді інформації на сторінці у мобіль

Configuration: Android version: 5.1

Browser: Google Chrome version:68.0.3440.91

Steps to reproduce:

1.зайти на сайт:<https://www.quality-assurance-group.com/shpargalka-dlya-qa-z-testuvannya-100>

2. -Перевірити наявність всіх колонок таблиці у повному обсязі:

-зменшивши масштаб таблиці

-збільшивши масштаб таблиці

3. Перевірити результат

Expected result:

Колонки мають бути наявні у повному обсязі при зміні масштабування у мобільній та планшетній версіях пер

Відповісти



Ольга

24 КВІТНЯ, 2019 О 08:13

Дуже дякуємо Маріє за Вашу роботу та ініціативу!

Над сайтом <https://www.quality-assurance-group.com>

На жаль Ваші баги враховані не будуть, оскільки сайт очікує в майбутньому суттєва реновація 😞

Якщо Ви дісно бажаєте допомогти нашому ресурсу. Він функціонує завдяки волонтерству. Ви скористал контентом, тому що у цьому напрямі робота і надалі продовжується!

Нам потрібні переклади статей — джерела ми надаємо.

А також є варіант просто скопіювати і розмістити статті на наш сайт, з прикольного сайту, який зник із

Відповісти

Мах

27 липня, 2019 о 17:35

Доброго дня, якщо актуально міг би допомогти з перекладами статей з англomовних джерел 😊
Напишіть на пошту.

Відповісти



Олька

28 липня, 2019 о 22:29

Вже пишу, дякуємо за починання і гарний приклад. Людоньки приєднуйтеся теж 😊

Відповісти

Tanya

17 січня, 2020 о 02:00

А можна ще приєднатися до перекладу?

Відповісти



Олька

17 січня, 2020 о 10:09

Звісно, ми тільки раді допомозі у поповненні скарбнички нашої україномовної Бази Знань для QA

Відповісти

Дмитрий

7 травня, 2019 о 19:04

В баг репорте обязательно нужно указывать актуальный результат.

Відповісти

Jan Dra

8 БЕРЕЗНЯ, 2020 О 00:28

Порада. Коли пишете про зменшення/збільшення масштабу, необхідно вказувати на скільки саме необхідно

Відповісти



Олька

15 КВІТНЯ, 2020 О 11:39

Дякуємо, врахуємо 😊

Відповісти

Юлія

16 СЕРПНЯ, 2019 О 12:16

це тільки мені видно, що в заголовку написано 100+ питань, а по факту 32? де інші можна побачити?

Відповісти



Олька

22 СЕРПНЯ, 2019 О 08:40

Ось, за цими посиланнями:

<https://www.quality-assurance-group.com/rivni-testuvannya/>

<https://www.quality-assurance-group.com/tehniky-testuvannya-shpargalka-dlya-qa-chastyi>

<https://www.quality-assurance-group.com/vydy-testuvannya-ta-vidminnosti-mizh-nymy-sh>

Відповісти

LEAVE A COMMENT

Коментар

Name

Email

Comment

Цей сайт використовує Akismet для зменшення спаму. Дізнайтеся, як обробляються ваші дані коментарі

Підписуйся на листівки

Раз у місяць ми надсилатимемо на пошту свіжі статті

☒ Введіть email у поле

ABOUT US

— найбільша українськомовна Інформаційна База Знань у галузі Забезпечення Якості в ІТ.

Дізнатися більше →

USEFUL LINKS

- Теорія тестування
- Книги
- Шпаргалки
- Корисні посилання
- Проектний менеджмент
- Ефективна комунікація
- QA Freelance
- HR Management

MONTHLY OFFERS

CONTACT US

Address

м. Львів Соборна пл.14

Phone

☎+38 (098) 903 64 45;

☎+38 (099) 376 65 05;

Email

quality.assurance.group.lviv@gmail.com