Компетентності рівня junior для QA	Загальна оцінка компетентності (max. 36)	Компанії											
		ArtJoker	G5 Entertainment	SoftServe	Viseven QA (Content Studio)	Viseven QA (Product/Enterprise)	Knubisoft	CHI Software	Avenga	Akvelon	Intellias	Telesens	SevenPro
1. Фундаментальні знання та навички з тестування програмного забезпечення													
1.1 Загальні питання тестування програмного забезпечення Розуміє процеси циклу тестування (планування тестування, тест-дизайн, виконання тестів та аналіз результатів) Ідентифікує типи тестування (функціональне, інтеграційне, модульне, регресійне, навантажувальне, розподілене тестування та ін.) Розрізняє методи тестування (чорний ящик, білий ящик, сірий ящик) Знає функціональні та нефункціональні вимоги до програмного забезпечення. Розуміє та застосовує в роботі принципи тестування (Seven Testing Principles) Класифікує види тестування (по об`єкту тестування, по знанню внутрішньої будови системи, по ступеню автоматизації, по ступеню ізольованості, по часу проведення тестування, по признаку позитивності сценаріїв, по ступені підготовленості до тестування) Розуміє важливість роботи з вимогами, принципи валідації та верифікації	**33**	3	3	3	3	2	3	3	2	3	3	2	3
1.2 Основи тест-дизайну, техніки тестування Вміє створювати ефективні тест-кейси та тест-сценарії, що відповідають вимогам і мають високе покриття функціональності застосунку. Розуміє різні техніки тест-дизайну (такі як еквівалентність, граничні значення, умовне покриття, парне тестування, структуроване тестування), вміє обирати та застосовувати найефективніші методи для створення тестів. Володіє навичками створення тест-кейсів, які відповідають специфікаціям та покривають важливі сценарії. Здатний виявляти різні сценарії тестування, які включають позитивні, негативні та виняткові умови. Виявляє значущі параметри та варіанти тестування. Використовує методи варіантного тестування, такі як таблиці перетинів, позиційний аналіз або категорійне тестування, для зменшення кількості тест-кейсів та забезпечення високого покриття. Практикує Pairwise testing (комбінаційне тестування або метод двійкових комбінацій). Створює розумні тестові набори даних. Вміє генерувати тестові дані, які відповідають різним умовам тестування та допомагають покрити різні шляхи виконання програми. Застосовує різні інструменти, які допомагають у створенні тест-кейсів та тест-сценаріїв, такі як таблиці рішень (Decision table), генератори даних або автоматичні генератори тестів. Здатний проводити ранжування тестів, може встановлювати пріоритети для тест-кейсів та тест-сценаріїв на основі важливості функцій, ризиків та реальних умов використання.	**32**	3	2	3	2	1	3	3	3	3	3	3	3
1.3 Дефекти та їх життєвий цикл. Bug Report Ідентифікує поняття дефекту в програмному забезпеченні, розуміє різні типів дефектів (функціональні, інтерфейсні, продуктивності, безпеки тощо) Має навички описування дефектів у структурованій та зрозумілій формі, включаючи відтворюваність, опис кроків для відновлення та прикріплення необхідних деталей. Здатний аналізувати дефекти з точки зору їх причин, впливу та потенційних наслідків. Вміє класифікувати дефекти за різними критеріями, такими як серйозність, пріоритет, причина, модуль або функціональна область. Вміє використовувати інструменти управління дефектами для реєстрації, відстеження та контролю над дефектами. Здатний оновлювати статус дефектів, визначати пріоритети, спілкуватися з командою розробки та здійснювати моніторинг виправлення дефектів. Практикує тестування виправлених дефектів для переконання в їхньому виправленні та відсутності побічних ефектів. Використовує ефективні методи та підходи до перевірки дефектів та їх регресії. Знає та використовує правильні термінології тестування та програмування для опису дефектів. Знає метрики дефектів (кількість дефектів, тривалість виправлення, частота повторного виявлення та інші показники)	**36**	3	3	3	3	3	3	3	3	3	3	3	3

1.4 Робота з тестовою документацією. Test Plan Знає основні види тестової документації (план тестування, специфікації тестування, тест-кейси, звіти про дефекти та ін.) Вміє розробляти та оформляти тест-план як вихідний документ для управління процесом тестування (включає опис стратегії тестування, цілі, завдання, обсяг, ресурси, графік, критерії завершення тестування та іншу важливу інформацію) Здатний аналізувати специфікації тестування, які містять вимоги до функціональності, нефункціональні вимоги, очікувані результати та іншу інформацію, необхідну для створення тест-кейсів. Має навички розробки ефективних тест-кейсів, які включають опис кроків тестування, очікувані результати, початкові умови, вхідні дані та іншу релевантну інформацію. Вміє організовувати тест-кейси, структурувати їх у набори, надавати пріоритети та позначати стан виконання для забезпечення повноти покриття тестів. Вміє генерувати звіти про дефекти, звіти про прогрес тестування та інші звіти, які надають інформацію про стан тестування, результати тестів, кількість виявлених дефектів та їх стан. Дотримується правил документування, таких як стандарти оформлення, нотації, керування версіями тестової документації та збереження відповідної документації для майбутнього використання.	**28**	3	1	3	2	1	3	3	1 / 2 / 3	1	3	3	3
1.5 Метрики якості програмного забезпечення Розуміє концепції метрик якості програмного забезпечення (важливість вимірювання та оцінки якості ПЗ та кількісного вимірювання аспектів якості) Знає основні метрики якості програмного забезпечення (функціональна придатність, надійність, ефективність, зручність використання, підтримка) Володіє інструментами для збору та аналізу даних для вимірювання метрик якості ПЗ Здатний виявляти ключові аспекти програмного забезпечення, які потрібно вимірювати та оцінювати. Розуміє контекст проєкту та бізнес-вимог для визначення відповідних метрик якості. Здатний аналізувати результати метрик якості та ідентифікувати проблеми або незадовільні аспекти програмного забезпечення, робити висновки та рекомендації на основі аналізу метрик.	**24**	2	1	2	1	2	3	2	2	2	2	2	3
1.6 Test Management, Risk Management Здатний планувати та керувати тестовими проєктами, включаючи визначення обсягу робіт, ресурсів, графіка та розподіл завдань між членами команди. Вміє оцінювати трудомісткість тестування та здійснювати керування ризиками. Має навички управління ризиками в контексті тестування. Вміє виявляти потенційні ризики, оцінювати їх вплив та ймовірність, розробляти план дій для зниження ризиків та моніторити їх стан. Здатний розробляти детальний графік тестових активностей, включаючи створення тест-кейсів, виконання тестів, реєстрацію дефектів, регресійне тестування та інші завдання. Управляє пріоритетами та ресурсами для досягнення максимальної ефективності. Розуміє процес виявлення, реєстрації, відстеження та управління дефектами. Знає про різні системи управління дефектами та вміє аналізувати статистику дефектів для прийняття рішень.	**18**	1	0	3	2	2	1	3	2	1	0	0	3
2. Розуміння Web-архітектури, принципів тестування мобільних додатків													
2.1 Клієнт-серверна архітектура Розуміє ролі та принципи взаємодії клієнтської сторони (фронтенду) та серверної сторони (бекенду) вебдодатка Знає протоколи передачі даних, таких як HTTP і HTTPS та їх статусні коди, заголовки та їх роль у передачі даних. Розрізняє методи HTTP, такі як GET, POST, PUT та DELETE, і їх призначення Розуміє основні принципи роботи IP-протоколу, способи адресації та маршрутизації даних в Інтернеті, версії IP-протоколу (IPv4 і IPv6), та їх особливості. Усвідомлює роль DNS у перетворенні доменних імен на IP-адреси та зворотний процес. Знає структуру URL і його складових частин (протокол, домен, шлях, параметри тощо). Ідентифікує кодування URL та знаки, що використовуються для перекодування спеціальних символів. Розуміє концепції API та їх роль у взаємодії програмного забезпечення. Здатний використовувати різні типи API, такі як веб-API (наприклад, REST або SOAP) і бібліотечні API. Ідентифікує формати даних, використовуваних для передачі даних через API, таких як JSON або XML.	**29**	3	0	3	1	3	3	2 / 3	2	3	3	3	3
2.2 Основні технології веброзробки, такі як HTML, CSS та AJAX Знає синтаксис HTML та його основні елементи для створення структури вебсторінок. Може використовувати тегів та атрибутів HTML для відображення тексту, зображень, посилань, таблиць, форм тощо. Вміє перевіряти правильність розмітки та коректність вкладеності тегів. Знає основи селекторів CSS для вибору та стилізації елементів вебсторінок. Розуміє основні властивості CSS для зміни кольорів, шрифтів, розмірів, відступів, позиціювання, анімації тощо. Вміє застосовувати каскадні правила та управляти пріоритетом стилів. Розуміє концепції AJAX та його використання для асинхронної взаємодії з сервером без перезавантаження сторінки. Здатний використовувати XMLHttpRequest або Fetch API для відправки та отримання даних у форматі JSON або XML. Вміє обробляти результати AJAX-запитів та відображати їх на сторінці без перезавантаження. Вміє використовувати HTML5-атрибути та API, такі як required, pattern, minlength, maxlength для валідації форм.	**22**	3	0	3	1	3	3	2	2	2	0	1	2
2.3 Основні принципи Web-stack Ідентифікує набір технологій та компонентів, що використовуються для розробки вебдодатків, включаючи клієнтську сторону (HTMSS, CSS та JavaScript), сервісну сторону (сервери додатків та вебфреймворки для них з використанням відповідної мови програмування), базу даних (SQL або NoSQL)	**18**	2	0	2	0	2	3	2	2	2	1	1	1

2.4 Тестування мобільних додатків. Mobile testing Знає основні мобільні платформи, такі як Android і iOS, їх особливості, архітектуру та обмеження. Розуміє версії операційних систем для мобільних пристроїв та їх відмінності. Ідентифікує функції, що є унікальними для мобільних додатків, таких як GPS, акселерометр, камера, сповіщення тощо. Здатний здійснювати платформозалежне тестування, тестування на різних пристроях та роздільних здатностях екранів, переконання у сумісності з різними версіями операційних систем (такі характеристики як розміри екрана, орієнтація, розміщення кнопок, живлення, мережеве підключення тощо) Здатний здійснювати тестування інтерфейсу користувача (зручність та ергономіку інтерфейсу, якість графічного виконання, адаптивність до різних режимів екрана, коректне розміщення елементів інтерфейсу на різних пристроях та роздільних здатностях екрана) Розуміє особливості тестування стабільності та продуктивності. Знайомий з виконанням тестів навантаження та стресу для перевірки стабільності додатка при великому навантаженні та поганих мережевих умовах Володіє навичками тестування сумісності та інтеграції, тестування безпеки мобільного додатку.	**26**	2	2	2	2	2	3	3	3	1	2	1	3
3. Основи розробки програмного забезпечення														
3.1 Знання життєвого циклу розробки ПЗ та місця тестування в ньому Ідентифікує різні методології розробки, такі як водоспадна модель, ітераційна модель, Agile, Scrum тощо Знає етапи життєвого циклу розробки ПЗ, такі як аналіз вимог, проєктування, реалізація, тестування та випуск, розуміє роль QA в кожному етапі та особливості співпраці з членами команди розробки	**33**	3	2	3	2	3	3	3	3	2	3	3	3
3.2 Знання актуальної мови програмування Має базове розуміння різних мов програмування, таких як Java, C#, Python, JavaScript тощо. Знає основні концепції програмування, такі як змінні, умовні оператори, цикли, функції, об'єкти тощо. Вміє аналізувати та розуміти код розробленого програмного забезпечення	**16**	0	0	3	1	2	0	1	1	3	3	1	1
3.3 Дизайн та архітектура програмного забезпечення Розуміє основні принципи дизайну та архітектури програмного забезпечення (SOLID, DRY, KISS, YAGNI, Separation of Concerns тощо). Знає основні патерни проєктування (фабрика, одиночка, спостерігач тощо)	**13**	0	0	2	0	1	1	1	2	2	2	1	1
3.4 Тестування коду та автоматизоване тестування Володіє методами та підходами до тестування коду (модульне тестування, інтеграційне тестування, тестування одиниць тощо) Вміє створювати автоматизовані тести та використовувати інструменти автоматизації тестування, такі як JUnit, NUnit, Selenium, Appium тощо	**18**	0	0	3	1	2	1	1	1 / 3	3	3	1	1
3.5 Навички роботи SQL Має базові знання SQL syntax, вміє будувати запити (queries) для отримання інформації з бази даних, модифікації даних та виконання інших операцій. Вміє працювати з таблицями та схемами, розуміє зв'язки між таблицями. Здатний працювати з ключами та індексами для забезпечення ефективного доступу до даних. Розуміє типи з'єднань (INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN) та може використовувати їх для об'єднання даних з різних таблиць та отримання з'єднаних даних Може будувати та виконувати SELECT-запити для отримання певних даних з таблиць та фільтрацію даних на основі певних умов, здійснювати їх агрегацію та групування Вміє працювати зі схемами баз даних, встановлювати права доступу користувачів тощо	**20**	1	0	2	1	1	2	2	2	2	2	3	2
3.6 Навички роботи NoSQL Знає різні інструменти та бібліотеки для роботи з NoSQL базами даних, такі як MongoDB, Cassandra, Redis, Couchbase Розуміє різницю між ACID (Atomicity, Consistency, Isolation, Durability) транзакціями, що характерні для реляційних баз даних, та BASE (Basically Available, Soft state, Eventually consistent) підходом, що використовується в NoSQL базах Вміє працювати зі специфічними структурами даних, які забезпечуються NoSQL базами, такими як дерева, графи, документи тощо Має навички тестування взаємодії NoSQL баз даних з іншими компонентами, такими як бекенд, фронтенд, API	**14**	0	0	2	1	2	0	2	2	1	2	1	1
4. Платформи та інструментарій команди з розробки програмного забезпечення														
4.1 Системи контролю версій - GIT Розуміє git flow - методологію роботи з гілками для ефективного управління процесом розробки. Здатний створювати репозиторії та ініціалізувати проєкти з використанням Git. Розуміє та використовує команди Git (commit, push, pull, fetch, checkout, add, status, revert та інші). Вміє працювати з гілками (branches) для розробки функціональності відокремлено від основної гілки (наприклад, розробка нових функцій у власних гілках та їх об'єднання в основну гілку). Знайомий з процедурою створення пулл-реквестів для обговорення та злиття змін з різних гілок.	**20**	0	0	2	1	3	3	2	1	3	2	1	2
4.2 CI/CD (Continuous Integration/Continuous Deployment або Continuous Delivery) Розуміє цілі та переваги CI/CD, включаючи зменшення часу циклу розробки, покращення якості програмного забезпечення та автоматизацію процесу розгортання. Знає про різні етапи CI/CD процесу, включаючи інтеграцію коду, автоматичну збірку (build), автоматизоване тестування, розгортання та неперервну доставку (deployment) програмного забезпечення. Вміє автоматизовувати тестування програмного забезпечення з використанням відповідних інструментів, таких як фреймворки для автоматизованого тестування (наприклад, Selenium, Appium), інструменти для автоматизованого тестування API (наприклад, Postman), інструменти для тестування продуктивності (наприклад, JMeter).	**18**	0	0	2	1	2	3	1	2	3	1	1	2

4.3 Знання Linux-середовища, Docker Розуміє основні концепції та принципи роботи з операційною системою Linux. Вміє працювати з командним рядком Linux, виконувати команди, керувати файловою системою, управляти користувачами та правами доступу. Працює зі скриптовими мовами, такими як Bash, для автоматизації рутинних завдань та конфігурації середовища. Вміє створювати, налаштовувати та управляти Docker контейнерами для окремих мікросервісів. Знає Docker-команди для роботи з образами, контейнерами, мережами, томами та іншими компонентами Docker. Здатний налаштовувати середовище для тестування у контейнерах, зокрема використовуючи Docker Compose для створення багатоконтейнерних середовищ. Ідентифікує інструменти для автоматизованого тестування у контейнерах, такі як Selenium Grid, для запуску тестів у розподіленому середовищі. Вміє працювати з інструментами моніторингу, такими як Prometheus, Grafana, для відстеження стану контейнерів та ресурсів серверів. Виконує тести масштабованості та навантаження, використовуючи Docker Swarm або Kubernetes для розгортання та керування кластером контейнерів.	**9**	0	0	2	0	2	0	0	1	2	0	1	1
4.4 Робота з Confluence Має навички створення структури сторінок та підрозділів для організації документації. Застосовує сторінки-заготовки, шаблони для стандартизації документів та використання повторюваних елементів. Використовує функціонал сповіщень та підписки для сповіщення про зміни у документації, здатний коментувати та обговорювати сторінки в Confluence Вміє налаштовувати права доступу для різних користувачів та груп. Ефективно використовує пошукові можливості Confluence для знаходження необхідної інформації	**18**	1	1	2	1	3	0	1	2	3	1	1	2
5. Платформи та інструментарій для тестування													
5.1 Робота з браузерами (Chrome, IE, FF, Opera тощо) Ідентифікує популярні веббраузери, такі як Google Chrome, Mozilla Firefox, Microsoft Internet Explorer (IE), Microsoft Edge, Opera та Safari. Знає особливості кожного браузера, його підтримку стандартів, можливості та обмеження. Вміння встановлювати та налаштовувати різні версії браузерів для тестування на різних платформах, працювати з параметрами конфігурації браузера, такими як налаштування кешу, кукі, безпеки та інші налаштування. Здатний запускати та тестувати вебдодатки на різних браузерах. Може проводити автоматизоване тестування у різних браузерах, таких як Selenium WebDriver. Вміє виконувати тестування на мобільних браузерах, включаючи емуляцію та перевірку респонсивного дизайну.	**28**	3	0	2	3	3	3	3	3	3	1	2	2
5.2 Навички роботи системами керування проєктами і дефектами (Jira, Bugzilla, Redmine) Вміння створювати нові задачі, дефекти та підзадачі в Jira з необхідною інформацією. Здатний налаштовувати поля дефекту, такі як пріоритет, тип, власник, дата тощо, при створенні задач; присвоювати та змінювати відповідальних осіб для завдань. Має навички змінювання статусів завдань, включаючи перехід між статусами та управління життєвим циклом дефектів(create/reopen/close тощо). Розуміє процес планування завдань та встановлення пріоритетів. Здатний оцінювати часові рамки та складність завдань. Вміє застосовувати функцію пошуку та фільтрації в Jira для знаходження необхідних завдань. Використовує JQL (jira query language) Вміє використовувати різні типи звітів для відстеження прогресу тестування та інших метрик.	**26**	3	2	2	2	3	0	3	2	3	1	2	3
5.3 Робота з API (Postman, Swagger) Вміє створювати запити до API за допомогою Postman, встановлювати заголовки, параметри та тіло запиту. Здійснює автоматизацію запитів, створення колекцій запитів, використання змінних та середовищ для управління запитами Знає структуру Swagger-специфікації, її основні компоненти та особливості використання для опису API. Здатний працювати з Swagger UI для перегляду та взаємодії з API, використовуючи документацію Swagger. Здатний проводити валідацію відповідей API за допомогою різних методів, таких як перевірка статус-коду, перевірка вмісту JSON/XML, перевірка значень полів тощо. Вміє використовувати вбудовані асерти та власні скрипти для валідації даних. Розуміє різні методи авторизації та аутентифікації, таких як базова авторизація, токени, OAuth тощо. Вміє виконувати авторизовані запити до API за допомогою відповідних заголовків або параметрів. Володіє інформацією про інші інструменти тестування API, такі як REST Assured, Insomnia, SoapUI тощо. Вміє використовувати додаткові можливості цих інструментів для тестування та автоматизації API.	**24**	2	0	3	1	3	3	2 / 3	2	2	2	2	2
5.4 Робота з Test Management Tools (TestRail, Zephyr, TestLink) Вміє створювати нові тест-кейси в TestRail з детальним описом кроків, очікуваних результатів та прикріплених файлів. Здатний категоризувати та організувати тест-кейси в логічні секції та суїти. Володіє інформацією про різні типи полів та налаштування в TestRail, наприклад, пріоритет, відповідальна особа, статус тощо. Може створювати тест-плани в TestRail, включаючи вибір відповідних тест-кейсів для включення у план. Володіє навичками планування тест-кейсів відповідно до пріоритетів та встановлення термінів виконання. Здатний генерувати звіти про результати тестування, здійснювати процес перевірки та валідації результатів тестування в TestRail. Вміє налаштовувати взаємодію TestRail з системами керування дефектами, такими як Jira, Bugzilla тощо.	**23**	3	1	3	2	2	0	3	2	3	1	1	2
5.5 Тестування продуктивності (Apache JMeter, Gatling, LoadRunner) Здатний встановлювати та конфігурувати Apache JMeter, Gatling і LoadRunner для проведення тестів продуктивності. Вміє створювати тестові сценарії, налаштовувати параметри навантаження, визначати поведінку користувачів, робити заміри та встановлювати метрики продуктивності. Знає розподілену архітектуру тестів продуктивності, включаючи моніторинг серверів, розподіл навантаження та аналіз результатів. Може проводити аналіз та інтерпретацію результатів тестів продуктивності, включаючи графіки, діаграми та інші метрики продуктивності. Здатний знаходити проблеми та бутлеки, що впливають на продуктивність, та рекомендувати заходи для поліпшення продуктивності системи.	**11**	2	0	2	1	2	0	1	1 / 2	2	0	0	1

5.6 Тестування безпеки (OWASP ZAP, Burp Suite) Знає основні типи загроз безпеки, такі як перехоплення даних, SQL-ін'єкції, XSS (Cross-Site Scripting), CSRF (Cross-Site Request Forgery) тощо. Використовує різні техніки тестування безпеки, такі як фаззинг, перехоплення трафіку, тестування меж. Розуміє стандарти безпеки, такі як OWASP Top 10, і здатний виконувати тестування згідно з цими стандартами. Використовує основні функції OWASP ZAP (Burp Suite), включаючи сканування вразливостей, перехоплення трафіку, перевірку на вразливості XSS, внесення змін в запити та відповіді, SQL-ін'єкції тощо. Вміє налаштовувати відповідний застосунок для проведення тестів безпеки та аналізу результатів. Вміє виявляти та класифікувати різні типи вразливостей безпеки, використовуючи OWASP ZAP, Burp Suite або інші інструменти.	**8**	1	0	2	0	1	0	0	0	2	0	1	1
5.7 Автоматизація тестування (Selenium WebDriver, Appium, Cucumber, JUnit, TestNG, Robot Framework) Володіє основними підходами до автоматизації тестування (розробка тестових скриптів, побудова тестових фреймворків, використання інструментів) Вміє створювати автоматизовані тестові скрипти для вебдодатків, використовуючи локатори елементів, взаємодію з елементами сторінки, перевірку стану сторінки. Застосовує спеціальні застосунки для створення автоматизованих тестових скриптів для мобільних додатків використовуючи роботу з елементами мобільного інтерфейсу, взаємодію з мобільними пристроями, перевірку стану додатків. Вміє створювати тестові сценарії та бібліотеки, виконувати тести, аналізувати результати та генерувати звіти. Ідентифікує різні патерни автоматизації тестування, такі як Page Object, Data-driven, Keyword-driven, інтеграція з CI/CD тощо. Здатний застосовувати патерни автоматизації для покращення якості та стабільності автоматизованих тестів.	**15**	1	0	3	1	2	0	0	1	3	1	0	3
6. Розмовна англійська													
6.1 Правильна побудова фрази з узгодженням часу Вміє використовувати відповідні часові форми дієслів і структури речень для точного вираження часу в комунікації. Здатний правильно використовувати часові форми, такі як Present Simple, Present Continuous, Past Simple, Past Continuous, Future Simple, для передачі правильного значення в повідомленні.	**23**	1	0	3	2	2	3	0	2	3	2	2	3
6.2 Професійна IT термінологія в розмові з замовником Знає і використовує професійні IT-терміни та скорочення, які використовуються в IT-індустрії, для чіткого та зрозумілого спілкування зі замовником. Вміє перекладати складні технічні концепції на доступну мову для замовника, не знайомого з IT-галуззю.	**18**	1	0	2	1	2	2	0	2	3	1	1	3
6.3 Професійна IT термінологія у діловому звіті Знає і використовує спеціалізовану IT-термінологію та фразеологію при написанні ділових звітів. Здатний детально описувати технічні поняття, процеси або результати в діловому звіті. Вміє передати інформацію про технічні питання та проєкти зрозуміло та конкретно, використовуючи адекватні IT-терміни та абревіатури.	**18**	1	0	2	1	2	2	0	2	3	1	1	3
6.4 Професійна термінологія для QA Знає та розуміє специфічну термінологію, пов'язану з різними інструментами та технологіями, які використовуються в QA Розуміє та використовує специфічні терміни, патерни, алгоритми, фреймворки, бібліотеки та інші інструменти, які використовуються в тестуванні додатків	**24**	1	0	3	1	3	3	0	2	3	2	3	3
6.5 Неформальне спілкування (small talks) Вміє вести неформальну розмову, розпочинати діалоги та підтримувати невимушену атмосферу. Знає загальновживану термінологію та фрази, які використовуються у неформальних розмовах, таких як загальні питання про погоду, цікаві події, особисті захоплення, спорт, фільми тощо.	**14**	2	0	1	1	2	1	0	1	2	0	2	2

Коментарі до таблиці

Оцінки компетентностей вказані станом на 2023/2024 рр., по шкалі від 0 до 3.

«3» - компетентність вкрай важлива, якщо кандидат нею не володіє, його подальше просування на вакансію неможливе.

«2» - компетентність, знання якої обов'язково знадобиться кандидату при подальшому професійному зростанні.

«1» - некритична компетентність, якою можна знехтувати.

«0» - неактуальна компетентність.

Зауваження від компаній

ArtJoker

1.4 - тест-план як докумен не використовуємо зараз

G5 Entertainment

Оцінки висталені для джуніор тестувальника мобільних додатків.

CHI Software

1.1 - Варто додати рівні тестування

1.6 - Risk management для джуна може бути на рівні 2 - точно треба буде оволодіти цим, але можна трошки пізніше

3.2-3.4, 4.1-4.3, 5.7 - Якщо мова йде про мануального тестувальника

Avenga

1.4 - There are different types of activities and documentions and is better to devide them.
Test Documentation:
- Test Plan - 2 (creating of Test Plan is not resposbility for Junior QA - only understanding)
- Test Specification -2 (direct responsibility of Junior QA)
- Test Cases - 3 (direct responsibility of Junior QA)
- Defect Reports - 3 (direct responsibility of Junior QA)
- Summary/Status Reports - 1 (creating of Summary/Status Reports is not responsibility of Junior QA - only understanding)

1.5 - Junior QA Engineer should be aware of these knowledge areas but practical usage and deep understanding is relevant more for Middle QA Engineers.

1.6 - Junior QA Engineer should be aware of these knowledge areas but practical usage and deep understanding is relevant more for Middle QA Engineers.

3.4 - For Junior Manual QA:
a) Володіє методами та підходами до тестування коду (модульне тестування, інтеграційне тестування, тестування одиниць тощо) - 1 (Only general Theoretical knowledge).
b) Вміє створювати автоматизовані тести та використовувати інструменти автоматизації тестування, такі як JUnit, NUnit, Selenium, Appium тощо - 1 (Only general Theoretical knowledge).

For Junior Automation QA:
a) Володіє методами та підходами до тестування коду (модульне тестування, інтеграційне тестування, тестування одиниць тощо) - - 1
b) Вміє створювати автоматизовані тести та використовувати інструменти автоматизації тестування, такі як JUnit, NUnit, Selenium, Appium тощо - 3

5.5 - General Understanding is required for Junior QA Engineer.

5.6 -Security Testing is not a responsibility of Junior QA Engineer

5.7 - General Understanding is required for Junior QA Engineer.

Akvelon

Необхідно додати навчики оцінки задачі (estimation). Навіть фахівці рівня Junior повинні мати розуміння як оцінити задачі свого рівня. Звісно, більш мастшабні та складні задачі - out of scope

1.2 - Було б гарно додати навички пов'язані з роботою з чек-лістом. Повинно бути розуміння цього виду тестової документації. Бо іноді чек-лісти використовуються як єдиний інструмент, в інших випадках на їх основі робляться тест-кейси. Взагалі, вони допомагають з тестовим покриттям функціональності.

1.4 - Потрібне концептуальне розуміння що це і для чого, вміння користуватися готовими. Але створення тест-планів, зазвичай, роблять спеціалісти вищого рівня.

1.6 - Потрібне загальне розуміння, але більш глибокого розуміння та використання, зазвичай, не вимагають від рівня junior.

2.3 - Цей пункт можна об'єднати з пунктом 2.2, оцінка "2" буде актуальною для об'єднанного варіанту.

3.1 - Додати Kanban, який зустрічається на практиці

3.2-3.4 - Актуально для позиції SDET. Для Manual QA це може бути 0.

3.6 - Рідко зустрічається на продакшен проєктах. Вузько-спрямована компетенція. Це, якщо виникне потреба, можна опановувати окремо. Але як базова компетенція на практиці не так часто зустрічається.

4.1 - Актуально для позиції SDET. Для Manual QA це може бути 1.

4.2 - Актуально для позиції SDET. Для Manual QA це може бути 0

5.4 - Актуально для позиції Manual QA. Для SDET це може бути 2.

5.5-5.7 - Актуально для позиції SDET. Для Manual QA це може бути 0.

Intellias

1.1 - Рекомендуємо додати ще питання на визначення різниці між Quality Assutance та Quality Control

1.2 - для Junior позицій не вимагаємо знання complex test design techniques, як-то pairwise testing, decision tables, state transition diagrams.
Також від Junior не вимагаємо самостійно встановлювати пріорітети тест-кейсів.

1.4 - Від джуніора очікується знання основних секцій тест плану, але розробляє він лише секції про test design та test execution. Секції, які включають стартегію, планування, оцінку заповнює лід або менеджер.

1.5 - Від джуніорів очікується, що вони знають навіщо потрібні метрики та можуть навести приклади простих метрик та їх формули (test coverage, # of open defects, # of passed test cases etc.

1.6 - Зовсім не актуально для джуніорів. Очікується, що вони будуть піднімати ризики, але не керувати ними.

2.2 - не актуально для джуніор тестувальника, більш для розробника.

3. Основи розробки програмного забезпечення - Оцінки проставлені для junior test automation

3.3 - Для тест-автоматизаторів необхідні патерни Page Object та Page Factory. Знає та може пояснити основні принципи OOP (Abstraction, Encapsulation, Inheritance, Polymorphism, Single responsibility principle)

3.6 - Джуніор повинен розуміти різницю між реляційною та нереляційною базою

4.2 - Джуніор має знати лише основні терміни (Continuous Integration/Continuous Deployment або Continuous Delivery) та що вони означають

4.3 - Не потрібно

5.3 - Джуніор повинен розуміти основні принципи тестування АПІ, через Swagger або Postman

6.1 - Очікуваний рівень англійської B1. Добре розуміти та оперувати термінами англійською мовою

Telesens

1.4 - Все окрім тест-плану

4.3 - На співбесіді питаємо лише базові консольні команди, не більше

4.4 - Мінімально лише базовий функціонал (без прав, налаштувань і т.д.)

5.2 - Використовується Jira, тому це основний інструмент по якому і задаються питання

5.5 - Не рівня Junior

5.6 - Достатьо мінімального розуміння на рівні базових OWASP TOP10

SevenPro

Також для Junior QA важливо:

Знати різновиди Environments і чим вони відрізняються: Stage, Pre-prod, Prod	3
Розуміти Deployment process від початку розробки до Production (Створення feature гілки розробником --> Створення Feature build Розуміння що таке Beta версія, як і коли її перевіряти.	3
Знання різних видів релізів: Major, Minor, Patch.	3
Знання iOS та Android guidlines для успішної подачі білда в стор.	3
Вміння працювати зі сніфферами (Web debugging proxy application) такими як Charles/Fiddler	2
Розуміння для чого потрібні логи, де їх знайти та як вони допомогають відтворити баги. Розуміння що таке fatal та non-fatal crashes.	2