

# Тестування програмного забезпечення

Викладач: к.т.н. Шитікова Олена Вікторівна

# Лекція 3

## ВЕБТЕСТУВАННЯ ТА ЧЕК-ЛІСТИ

# Поняття вебтестування та основні підходи

Компанії та окремі користувачі все більше залежать в своїй роботі від вебдодатків (web-applications), які дуже динамічні, а їх функціональні можливості постійно зростають.

Безперервно зростає і потоковий трафік засобів інформації і запитів, які формуються переносними та вбудованими пристроями. Вебдодатки стають все більш поширеними і більш складними, граючи, таким чином, основну роль в більшості онлайн-проектів.

Як і у всіх системах, що ґрунтуються на взаємодії між клієнтом і сервером, вразливості вебдодатків зазвичай виникають із-за некоректної обробки запитів клієнта чи недостатньої перевірки вхідної інформації зі сторони розробника.

# Поняття вебтестування та основні підходи

Важливою особливістю вебтестування є часові рамки. Якщо при тестуванні програмного забезпечення в тестувальника є місяці (інколи роки), то у вебтестувальника є лише дні і тижні (в кращому випадку) на тестування запропонованого сайту.

Отже, якщо при тестуванні програм дозволяється і потрібно формувати детальні плани тестування, описувати тестові випадки на основі отриманої від розробника документації, то при тестуванні вебсторінок це може значно продовжити строки публікації матеріалів в мережі Інтернет. Таким чином, на вебтестування може виділятися до 50% загального бюджету і часових ресурсів.



# Поняття вебтестування та ОСНОВНІ ПІДХОДИ

**Вебтестування (Web-testing)** – тестування програмного забезпечення, сфокусоване на вебдодатках (web-applications), яке здійснюється з застосуванням систем, що використовують вебтехнології (web-basedsystem), та вирішують такі питання як: безпека вебдодатків, базова функціональність сайту, доступність користувачам тощо.

# Поняття вебтестування та ОСНОВНІ ПІДХОДИ

**Вебдодатки (Web-applications)** – клієнт серверний додаток, в якому клієнтом виступає браузер, а сервером – вебсервер.

**Вебсайт (Website)** – система з декількох сторінок, що мають одну адресу в мережі Інтернет.

**Верстка (Layout)** – етап дизайну сторінки сайту, що представляє собою просторове розміщення на ній текстових елементів і графічних зображень відповідно до концепції оформлення ресурсу.

**Вебдизайн (Web-design)** – галузь веброзробки, а також різновид дизайну, до завдань якої входить проєктування вебінтерфейсів (Web UI) для сайтів або вебдодатків.



# Етапи тестування (Stages of testing)

**1. Вивчення документації** (Documentation) – вивчається отримана документація, аналізується функціонал за технічним завданням, кінцеві макети сайту і складається план тесту для подальшого тестування. Необхідно визначити цілі сайту і споживачів сайту.

**2. Тестування верстки** (Layout testing) – перевіряється розташування елементів, відповідність їх позицій наданим макетам, а також перевіряється оптимізація зображень і графіки. Завершивши перевірку на валідність, фахівець приступає до перевірки на кроссбраузерність, тобто перевіряє працездатність сайту в різних браузерах та при різних параметрах налаштувань екрану.



# Етапи тестування (Stages of testing)

**3. Функціональне тестування** (Functional testing) – процес верифікації відповідності функціонування продукту його початковим специфікаціям і представляє найбільш тривалий етап перевірки ресурсу.

Перевіряється весь описаний функціонал:

- посилання;
- бази даних;
- кешування;
- фрейми;
- аудіо і відео;
- включення або відключення куків (cookie).
- форми;
- секретність;
- перевірка роботи з браузером;
- анімація;
- друк;



# Етапи тестування (Stages of testing)

**4. Тестування практичності / простоти використання** (Usability testing) – проводиться для оцінки зручності продукту у використанні:

– тестування навігації (Test for navigation) - вебсайт повинен бути простим у використанні, інструкції повинні бути зрозумілими і головне меню повинно бути передбаченим на кожній сторінці і бути послідовним;

– перевірка вмісту (Content checking) – зміст має бути логічним і простим для розуміння без орфографічних помилок, без використання кольорової гами, яка дратує користувачів.

# Етапи тестування (Stages of testing)

Тестування практичності одне з найдорожчих, тому що найбільш цінну інформацію можна отримати тільки від реальних користувачів, спостерігаючи за їх роботою з сайтом. Такі дослідження вимагають дорогої інфраструктури і тимчасових витрат, їх складно автоматизувати. Тестування практичності / простоти використання (usability testing) частково перетинається з тестуванням якості виконання сайту і зручності інтерфейсу.



# Етапи тестування (Stages of testing)

**5. Тестування безпеки (Security testing)** – перевіряється доступ користувача до службових/закритих сторінок, а також проводиться перевірка захисту всіх критично важливих сторінок (наприклад, розділу адміністрування сайту) від зовнішнього впливу і включає наступні дії:

- виявлення помилок в кодї сайту та програмному забезпеченні сервера;
- виявлення наявності таких вразливостей, як SQL Injection або ін'єкція XSS (Cross SiteScripting), PHP includes, биті посилання (broken links);
- тестування на стійкість до комбінованих методів атак і нестандартних технік зламування.

# Етапи тестування (Stages of testing)

На відміну від інших тестів, тестування безпеки слід проводити регулярно. Крім того, тестуванню піддається не тільки сам сайт або вебдодаток, а весь сервер повністю – і вебсервер, і операційна система, і всі мережеві сервіси. Як і у випадку інших тестів, програма «прикидається» реальним користувачем-хакером і намагається застосувати до сервера всі відомі методи атаки і перевіряє всі вразливості. Результатом роботи буде звіт про знайдені вразливості та рекомендації щодо їх усунення. Зазвичай такі сканери безпеки продаються як самостійний продукт.



# Етапи тестування (Stages of testing)

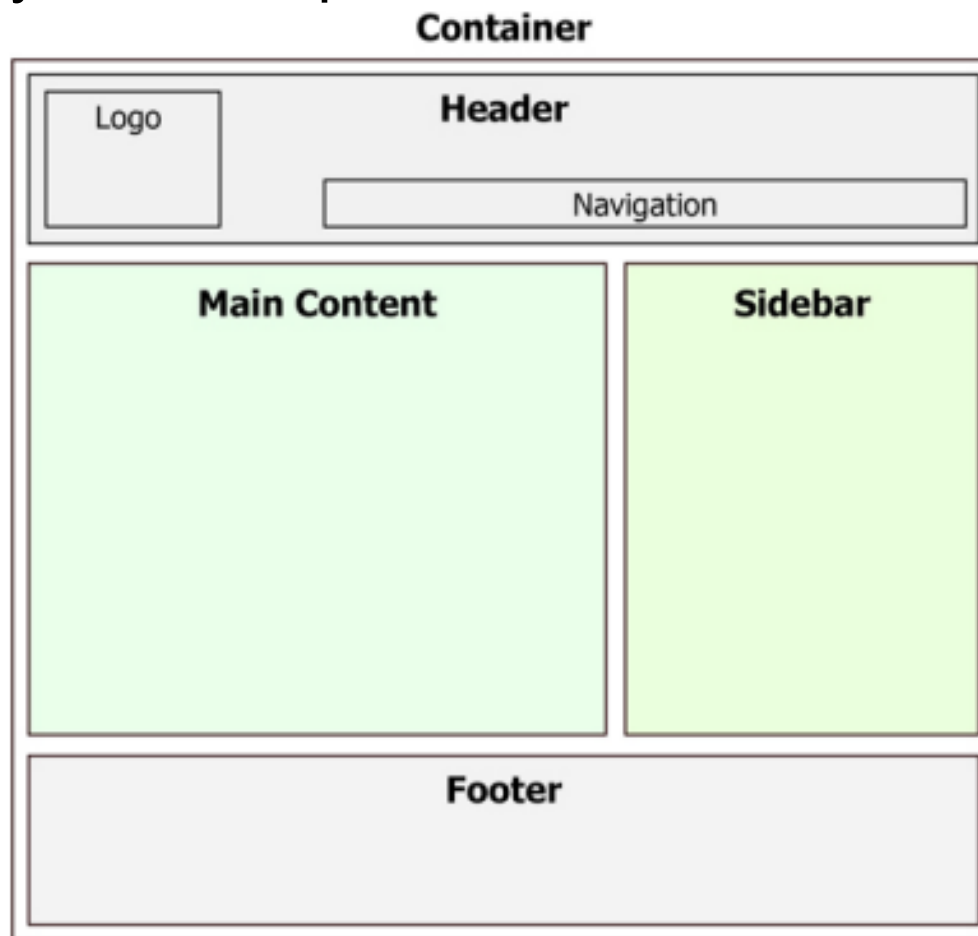
**6. Тестування продуктивності сайту** (Performance testing) Проводиться з метою визначення швидкодії сайту або частини під певним навантаженням.

Тестування продуктивності включає в себе наступні види тестування продуктивності:

- Навантажувальне тестування;
- Тестування швидкодії

# Анатомія вебсторінки

Елементи вебсторінки забезпечують життєво важливу і естетичну функцію сторінки.



# Тестування верстки

**1. Візуальна частина.** Тестування проводиться в одному з найчастіше використовуваному браузері в наступній послідовності:

- перевіряються помітні оку поламані блоки, невідповідність кольору, некоректне відображення тексту навколо зображень;
- точність відповідності макету;
- перевіряється сітка (вертикальні/горизонтальні вирівнювання).
- сайт перевіряється в різних розширеннях.

# Тестування верстки

- перевіряється зменшення розмірів вікна менше мінімального по технічному завданню – не повинно нічого «ламатися», фони не повинні «розпливатись»;
- перевіряється масштабування сторінки. У розумних масштабах (діапазон 75-150%) сторінка повинна виглядати без візуальних неточностей;
- перевіряється чи нормально підсвічуються поля у фокусі;
- перевіряється чи нормально підсвічуються поля з помилками.



# Тестування верстки

**2. Доступність** включає наступні етапи:

- чи виділяється текст в текстових блоках;
- чи натискаються «клікабельні» елементи;
- чи встановлюється фокус в поля форм;
- в ідеальному випадку всі активні елементи повинні реагувати на наведення, не доступні/неактивні – не повинні;
- клікабельні елементи, призначення яких не очевидно, повинні бути забезпечені підказками (title);

# Тестування верстки

- лого на головній сторінці посиланням бути не повинно, на всіх внутрішніх сторінках – повинно;
- перевірка друку сторінки (якщо було в ТЗ);
- написи (особливо логотип і головне меню сайту) повинні залишатися читабельними, для всіх інформаційних картинок повинні бути присутні підписи;
- працездатність при вимкненому JavaScript. Весь критично важливий функціонал сайту повинен бути доступний без JS.

# Тестування верстки

***3. Коректна робота при введенні реального тексту, надійність верстки.***

Верстка повинна «тягнутися», не розвалюватися та не втрачати дизайнерський вигляд при зміні контенту на сторінці. Контенту може бути більше або менше, ніж на макеті:

- Перевірка введення та видалення даних.
- Перевірка коректності роботи стилів.

# Тестування верстки

**4. Перевірка 404-их запитів.** Помилка 404 зобов'язана вписуватися в загальну концепцію сайту, мати ідентичний стиль, надавати відвідувачам можливість повідомляти адміністратора щодо непрацюючих URL.

**404 сторінка** зобов'язана вирішувати такі **завдання**:

- привертати увагу,
- пояснювати користувачам ситуацію,
- підказувати можливі варіанти подальших дій.

# Тестування верстки

Щоб сторінка 404 приносила ресурсу користь, вона повинна включати наступні елементи:

- перехід на головну сторінку;
- кнопка повідомлення про бите посилання;
- список популярних матеріалів;
- форму пошуку;
- посилання на карту сайту.

# Чек-лісти

**Чек-ліст** – це документ, який описує що має бути протестовано. При цьому чек-ліст може бути абсолютно різного рівня деталізації. На скільки детальним буде чек-ліст залежить від вимог до звітності, рівня знання продукту співробітниками й складності продукту.

Чек-ліст дозволяє:

- не забувати про важливі тести;
- фіксувати результати роботи;
- відстежувати статистику про статус ПЗ.

# Чек-лісти

***В чек-лісті має бути:***

- ✓ список перевірок (з необхідним ступенем деталізації);
- ✓ оточення перевірки:
  - збірка, на якій проводилося тестування;
  - тестове оточення (якщо є);
  - інформація про тестувальників;
- ✓ результат перевірки.

# Чек-лісти

## Приклад чек-ліста

	Тестувальник	Ольга Д.	Анна П.	Анна П.	Ольга Д.	ID баз-репорту
	Перевірки	Google Chrome 68.0.3285.140	Mozilla Firefox 58.0.2	Opera 51.0.28.30.34	Safari 5.1.7	
1	Наявність сторінки логіна	Passed	Passed	Passed	Passed	-
2	Авторизація з валідними даними	Passed	Passed	Passed	Passed	-
3	Авторизація з невалідними даними	Failed	Failed	Failed	Failed	005475
4	Авторизація з порожніми полями	Failed	Failed	Failed	Failed	005477
5	Реєстрація з валідними даними	Passed	Passed	Failed	Passed	005485
6	Реєстрація з невалідними даними	Passed	Failed	Failed	Passed	005493
7	Лист підтвердження реєстрації	Blocked	Blocked	Blocked	Blocked	005495



# Чек-лісти

Пункти можуть містити як лінійну структуру, так і деревоподібну, з розділами/підрозділами або без них. Вони повинні бути максимально короткі та в той же час зрозумілі тестувальнику, який ще не знайомий з додатком. Всі пункти повинні бути оформлені однією мовою.

Як правило, кожен чек-ліст має кілька стовпців. Кожен стовпець призначений для тестування на окремій платформі. Слід завжди вказувати назву пристрою, назви браузерів та їх версії.



# Чек-лісти

При проходженні чек-лістів тестувальник зазначає статус навпроти кожного тестованого пункту. Можливі такі варіанти статусів:

- «Passed» – перевірка пройдена успішно, багів не знайдено;
- «Failed» – знайдений один або більше багів;
- «Blocked» – неможливо перевірити, тому що один з багів блокує поточну перевірку;
- «In Progress» – поточний пункт, над яким працює тестувальник;
- «Not run» – ще не перевірено;
- «Skipped» – не буде перевірятися з будь-якої причини. Наприклад, поточний функціонал ще не реалізований.

# Чек-лісти

Для більшої наочності, як правило, кожен зі статусів має свій колір.

Всі заведені по чек-лісту баг-репорти повинні бути додані в примітки до комірок зі статусом «Failed». Цілком припустимо додавати примітки до комірок з іншими статусами, якщо це необхідно.

Для комірок зі статусом «Blocked» примітки з посиланнями на баг-репорти також обов'язкові. Однак, як правило, примітка в комірці зі статусом «Blocked» посилається на раніше заведений баг-репорт, який в одному з попередніх пунктів вже відзначений як «Failed». Іншими словами: пункт, одного разу зазначений як «Failed», може бути блокером для декількох або всіх наступних пунктів чек-ліста

# Чек-лісти

***Основні моменти, які варто враховувати при роботі з чек-лістами:***

- ✓ статус «Passed» встановлюється тільки для пунктів, які перевірені і не містять помилок;
- ✓ всі комірки зі статусом «Failed» і «Blocked» обов'язково повинні мати примітки з посиланнями на баг-репорти;
- ✓ після завершенні проходження чек-ліста не повинно залишитися комірок зі статусом «Not run».

# Чек-лісти

## Правила складання чек-лістів:

- 1. Один пункт – одна операція.** Пункти чек-ліста – це однозначні атомарні і повні операції.
- 2. Починати пункти з іменника.** Складаючи пункти слід дотримуватися уніфікованої форми.
- 3. Складати чек-ліст за рівнями деталізації.** Для зручності проходження чек-ліста, найкраще складати тести в послідовності використання функціоналу.

# Чек-лісти

## Переваги чек-лістів:

- ✓ Використання чек-лістів сприяє структуруванню інформації у співробітника.
- ✓ Чек-ліст стає пам'яткою, яка допомагає не забути важливі тести.
- ✓ При правильному запису необхідних дій у співробітника з'являється однозначне розуміння завдань. Це сприяє підвищенню швидкості навчання нових співробітників.

# Чек-лісти

## Переваги чек-лістів:

✓ Чек-лісти допомагають уникнути невизначеності та помилок пов'язаних з людським фактором. Збільшується покриття тестами програмного продукту.

✓ Підвищується ступінь взаємозамінності співробітників.

✓ Економія робочого часу. Написавши чек-ліст одного разу його можна використовувати повторно, з огляду на актуальність інформації.



# Чек-лісти

## Переваги чек-лістів:

- ✓ Використання статистики: хто, коли, що проходив (з деталізацією по збірці продукту та оточенню, на якому проводилося тестування).
- ✓ Можливість оцінити стан продукту, готовність до випуску.



# Чек-лісти

## Недоліки чек-лістів:

- ✓ початківці тестувальники не завжди ефективно проводять тести без достатньо докладної документації;
- ✓ чек-лісти неможливо використовувати для навчання початківців співробітників, так як в них недостатньо докладної інформації;
- ✓ замовнику або керівництву може бути недостатньо того рівня деталізації, який пропонують чек-лісти.