**Beet Seed — відпрацюй навички на базовому рівні.**

В баг-трекінговій системі (Jira) опиши 3 баги, які знайдеш на сайті <https://www.headhunterhairstyling.com/>  
  
**Додала баги в Hanna Alymova Sprint.**

**Beet Sprout— детальніше заглибся в практику.**

**Severity - Critical/Priority - Low**

**1. У формі замовлення email необов’язковий, але створення доставки неможливе без нього.**

На моїй попередній роботі інтернет магазин був підключений до CRM системи Bitrix. Для користувачів сайту було не обов'язковим заповнювати поле електронної пошти, проте менеджери (внутрішні користувачі) стикались з тим, що створити доставку для замовлень без email було неможливо.

**Severity: Critical**, оскільки відбувалось часткове блокування бізнес-процесу: без email доставка не створюється автоматично.

Співробітники змушені вигадувати фейкові пошти, витрачати час, ризикувати помилками.

**Priority: Low,** оскільки користувач з фронту не бачить проблеми: форма приймає замовлення. Виправлення не є терміновим з точки зору клієнта, хоча й вкрай потрібне для бек-офісу.

**2. Документ про переміщення товару в 1С не відображається через автоматичне зняття галочки при подвійному кліку на назву складу. При одинарному кліку, який передбачає система 1С все працює вірно.**

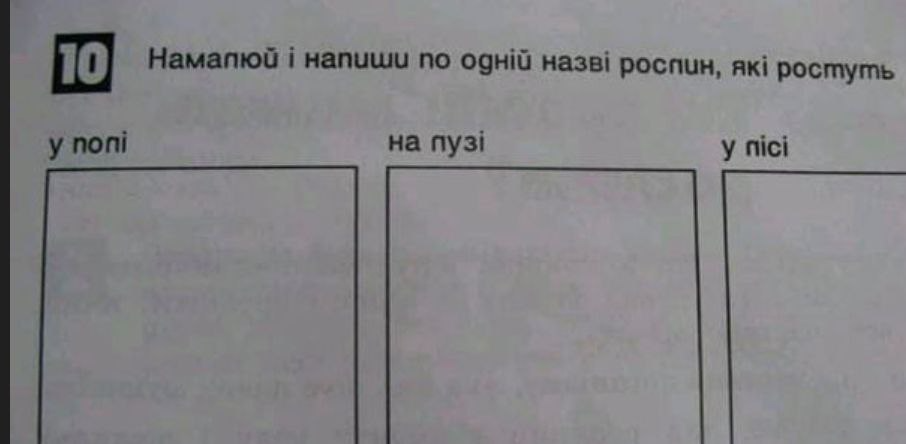
**Severity: Critical,** адже документ переміщення не фіксується у загальному списку і це впливає на точність обліку, звітність, відстежуваність документів.

**Priority: Low,** адже відбувається виключно у випадку, коли користувач двічі клікає саме по назві складу, і не помічає, що знялася галочка (бо вона маленька).

Така поведінка не є типовою — більшість користувачів не натискає двічі (механіка 1С зазвичай розрахована на 1 клік), або одразу бачить галочку.

Тому це не масовий баг, і виправлення не є терміновим.

**Severity - Minor / Priority - Highest**

**1. Відомий мем про завдання з дитячого підручника, де обрали дуже невдалий шрифт тексту. Це приклад несерйозної на вигляд баги, яка має високий пріоритет через контекст і потенційні наслідки.**

**Severity: Minor**, адже проблема не порушує функціональність підручника як навчального матеріалу — текст читається, задача зрозуміла. Це візуальна помилка — шрифт спричиняє двозначність.

**Priority: Highest,** оскільки проблема викликає неочікувану інтерпретацію, яка непристойна. У дитячому контексті — це репутаційний ризик для видавництва/школи. Батьки можуть обуритися, діти — сміятися, вчителі — зніяковіти. Тому потрібно виправити якнайшвидше, хоча технічно це не критичний баг.

**2. Два подібних кейси, стосовно дизайну, який може сприйматись неоднозначно кінцевим користувачем.**

2.1 Спіннер VW (Volkwagen) — «випадкова свастика».  
„Volkswagen go brrrrr“ - “When you actually spin it then no, it doesn’t show a swastika. However the company was founded in Nazi Germany.”

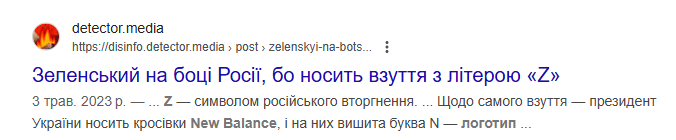
Тобто користувачі підтверджують: при певному ракурсі та швидкості логотип може випадково приймати форму свастики — це оптичний ефект, а не задумка дизайнера. Це лого використовувалось в рекламній кампанії Volkwagen.

**Severity: Minor,** адже проблема не стосується бізнес-процесів і дотримується графічних стандартів компанії. **Priority: Highest,** оскільки проблема може викликати гостру негативну реакцію або репутаційний скандал.



2.2 Логотип кросівок New Balance має простий і симетричний логотип “N”, що має схильність до оптичної метаморфози та при певному ракурсі може виглядати, як “Z”.

**Severity: Minor,** адже це не впливає на функціональність продукту та дотримується затвердженого дизайну логотипу компанії.

**Priority: Highest,** оскільки літера “Z” зараз часто асоціюється з російською агресією, і в Україні це дуже чутлива тема. Тому навіть натяки можуть викликати негативну реакцію та вплинути на репутацію компанії. Нижче реальний приклад тратування цього лого:

**Mighty Beet — різнобічно опануй тематику уроку.**

Ти як керівник/ця маєш створити життєвий цикл багу.

Які статуси туди будуть входити?

В якій послідовності?

Чому саме такі?

**Для проєкту я би обрала такі статуси життєвого циклу багу:**

**New** — баг зафіксований, але ще не проаналізований і не переданий у роботу.

**In Work** — баг у роботі, розробник вже почав над ним працювати.

**QA Re-test** — зміни внесено, і тестувальник перевіряє, чи проблему дійсно усунули.

**Reopened by QA** — тестувальник виявив, що баг не виправлено повністю або виправлення викликало нові проблеми.

**Verified** — тестувальник підтвердив, що баг виправлений, і проблема більше не відтворюється.

**Refused** — баг визнано недійсним, дубльованим або таким, що не буде виправлятись.

**Послідовність руху багу:**

**Базовий шлях (успішне виправлення)**:

New → In Work → QA Re-test → Verified

**Повторне відкриття після невдалого фіксу:**

New → In Work → QA Re-test → Reopened by QA → In Work → QA Re-test → Verified

**Відхилення багу:**

New → In Work → Refused

**Чому саме такі статуси:**

Ці статуси дозволяють прозоро відстежувати життєвий цикл бага, розуміти, на якому етапі зараз проблема, чітко розподіляти відповідальність між розробниками й тестувальниками, підвищувати якість продукту через контроль кожного етапу виправлення дефекту.