

Manual QA

Види тестування

Manual QA

План уроку

- Види тестування
 - за запуском коду на виконання;
 - за доступом до коду та архітектури застосунку;
 - за рівнем деталізації застосунку;
 - за рівнем автоматизації;
 - за очікуваним результатом;
 - за ступенем важливості функцій, що тестуються;
 - за цілями та завданнями;
- Рівні тестування
- Практика

Що було минулого разу

Manual QA

План уроку

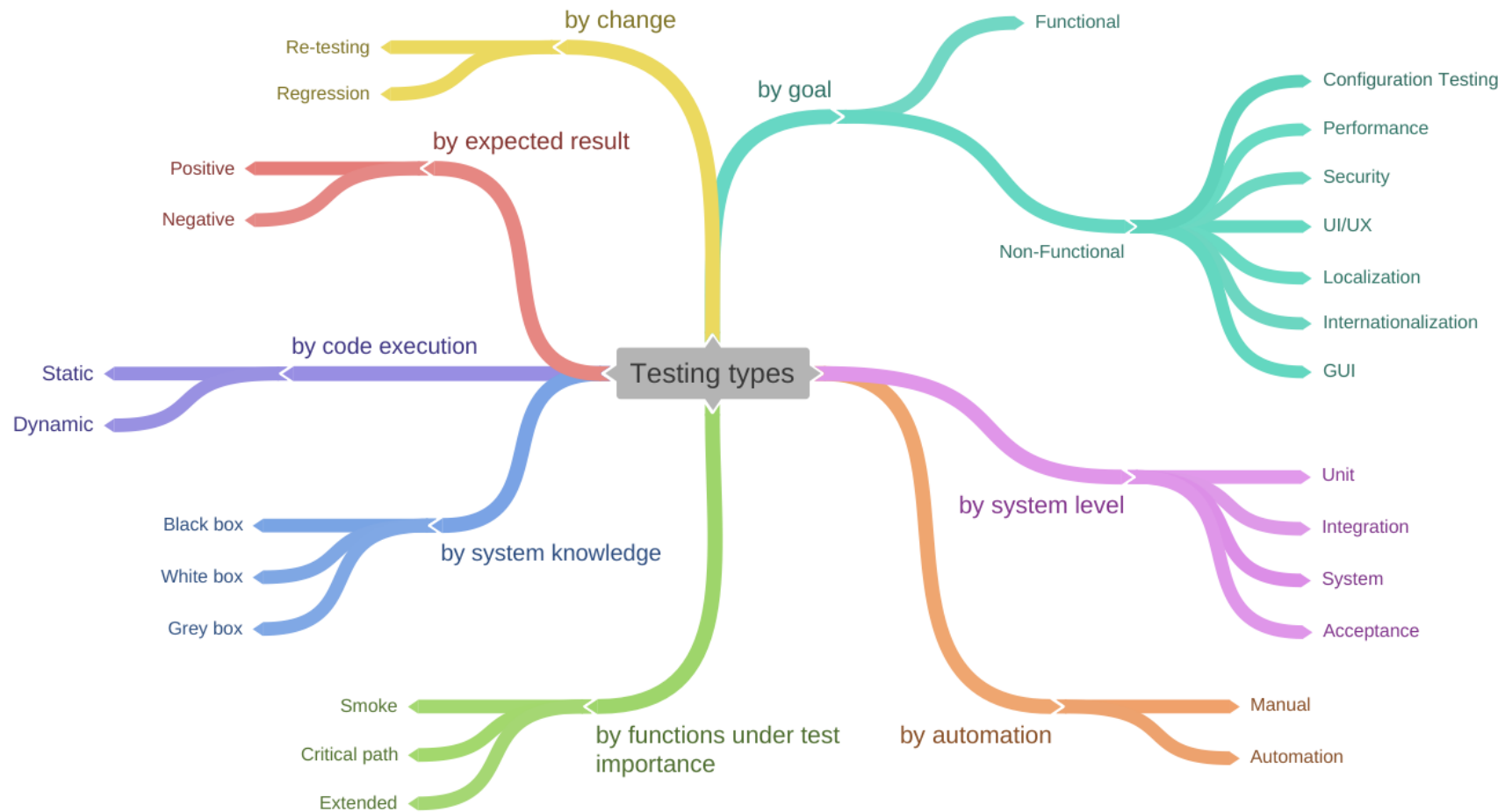
- Позитивні та Негативні тести
- Black box
 - Попарне тестування (Pairwise testing)
 - Тестування переходів станів (State transition testing)
 - Користувацькі сценарії (Use case testing)
- Experience based
 - Передбачення помилки (Error guessing)
 - Дослідницьке тестування (Exploratory testing)
 - Інтуїтивне тестування (Ad-hoc testing)
- Як писати гарні тест-кейси

Manual QA

Види тестування

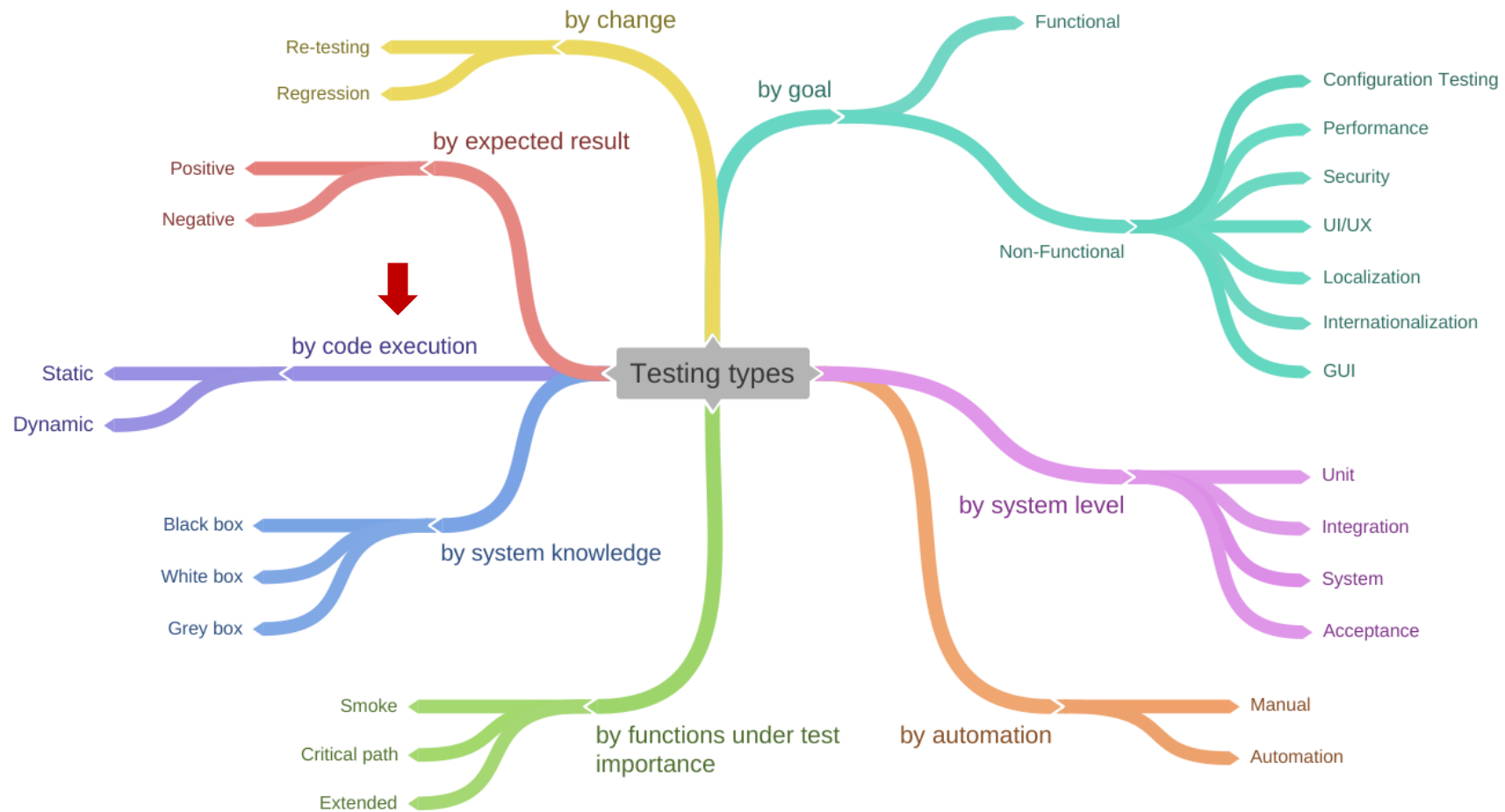
Manual QA

Види тестування



Manual QA

Види тестування

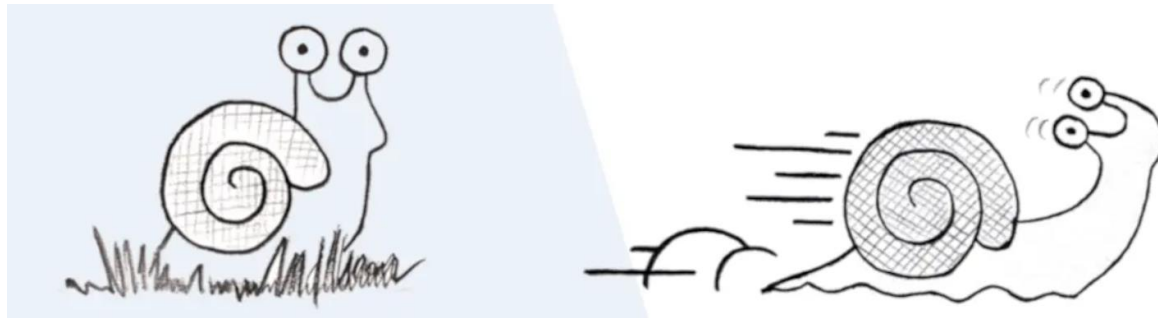


Manual QA

За запуском коду на виконання

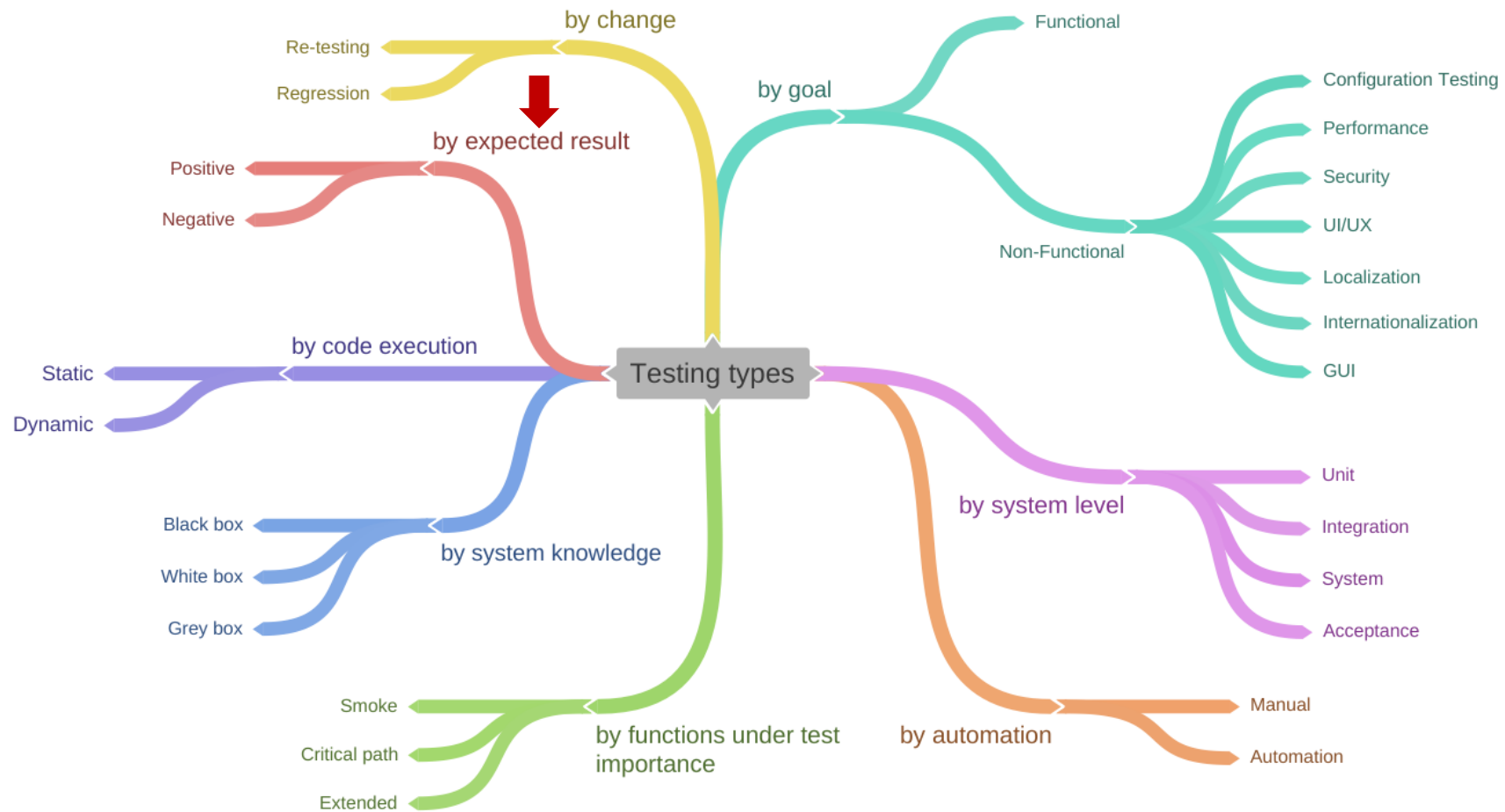
Статичне тестування — це процес, який зазвичай асоціюють із аналізом документації чи програмного коду, тобто **не** на запусченому ПЗ.

Динамічне тестування - це процес тестування, що відбувається над працюючим ПЗ. Цей вид тестування не може бути здійснений без запуску застосунку.



Manual QA

Види тестування



Manual QA

Позитивні та Негативні тести



Позитивні тести – дозволяють перевірити , чи працює система в нормальних умовах так, як задумувалося.

Іншими словами

Позитивні тести – перевіряють, що те, що має працювати – працює



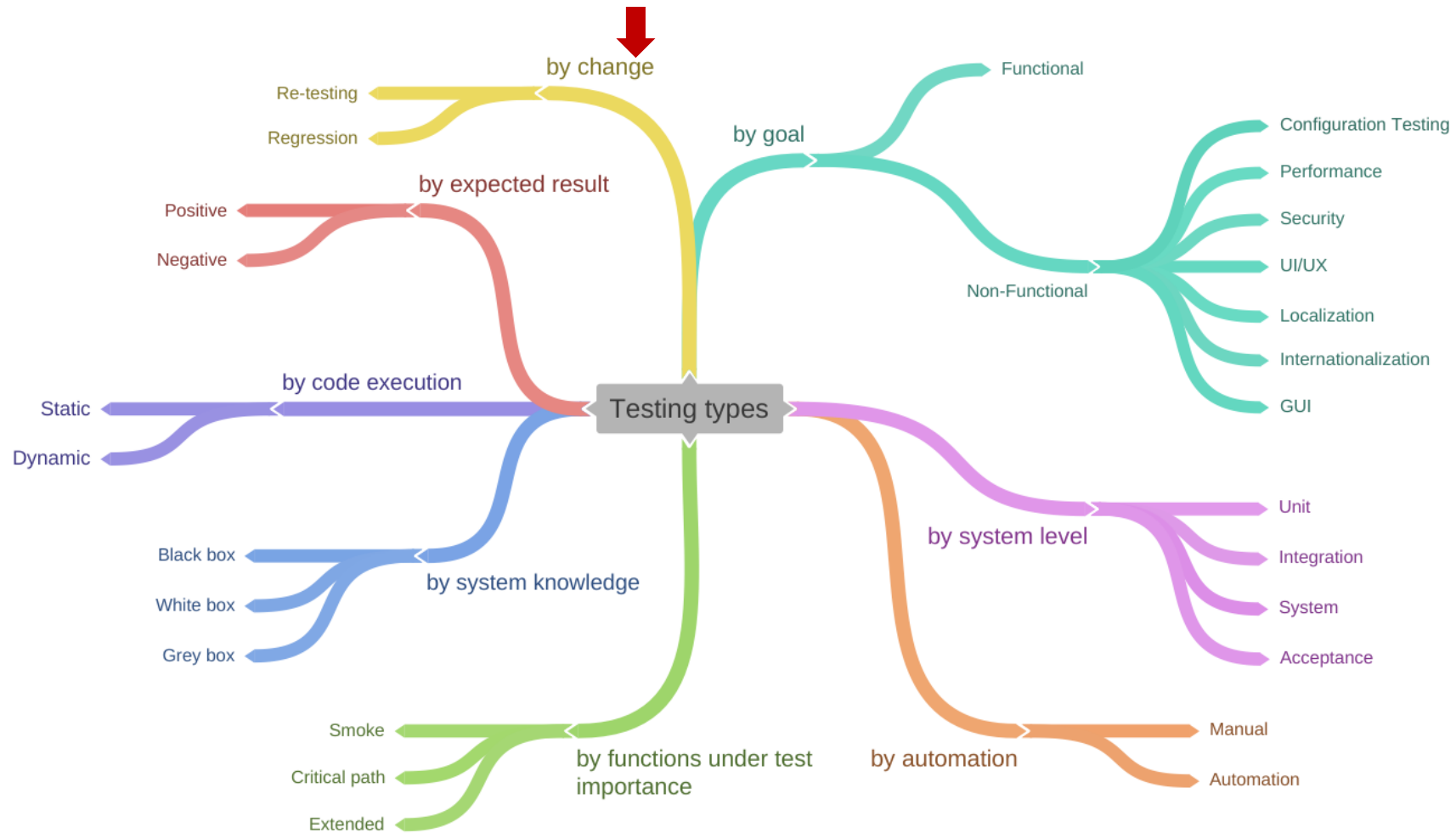
Негативне тестування – гарантує, що застосунок продовжить роботу у разі помилки або непередбаченої поведінки

Іншими словами

Негативні тести – перевіряють реакцію застосунку на невалідні дані/поведінку

Manual QA

Види тестування



Manual QA

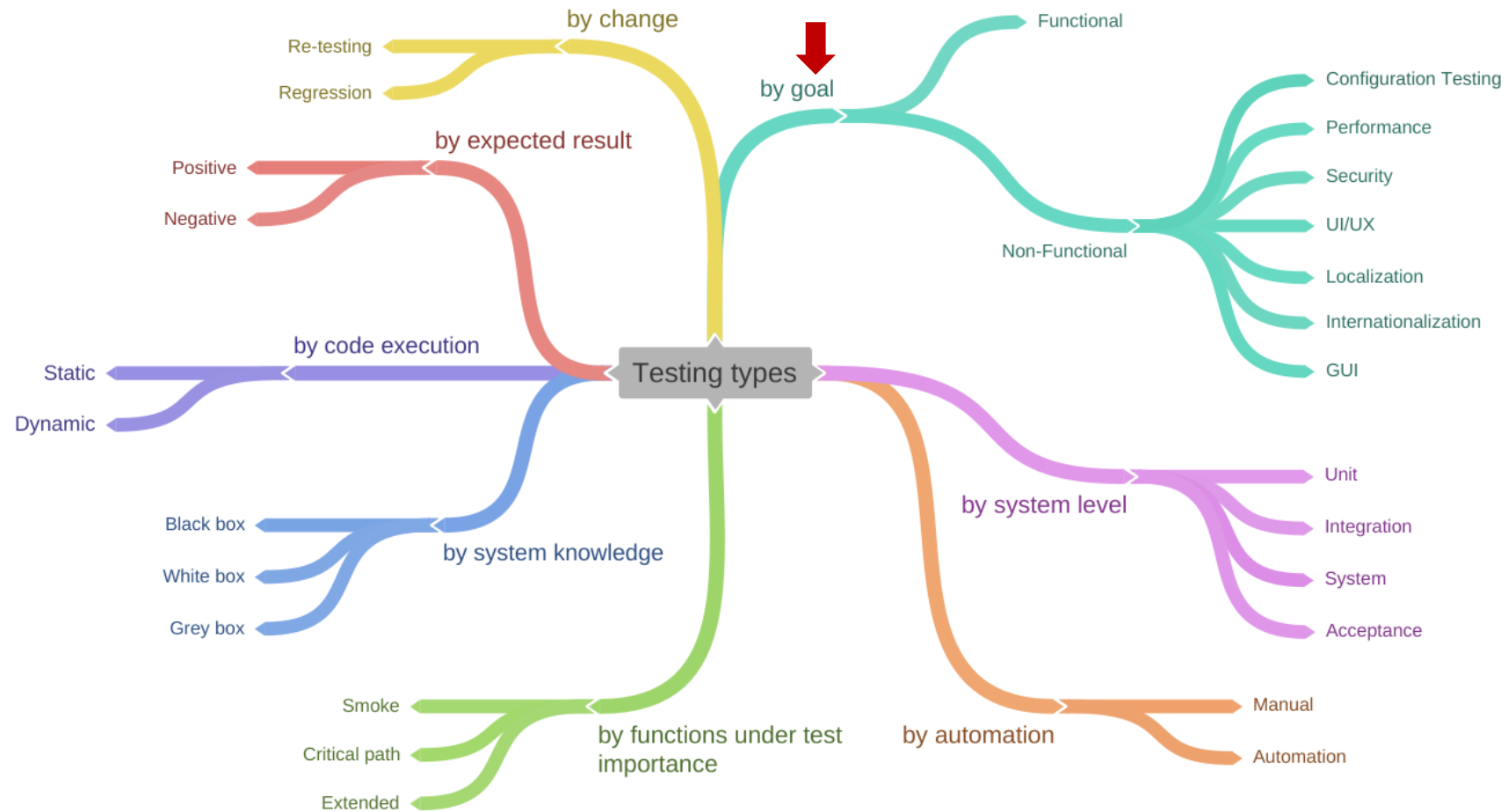
За змінами

Регресійне тестування (regression testing) – це тестування відсутності деградації системи. Простіше кажучи – тести, які проходили раніше, повинні пройти і зараз.

Повторне тестування (re-testing) – виконання тест-кейсів, які раніше виявили дефекти, з метою підтвердження усунення дефектів.

Manual QA

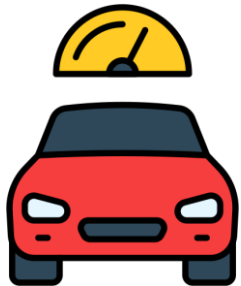
Види тестування



Manual QA

Функціональне

Функціональне тестування визначає, чи виконує програмне забезпечення свої прямі функції (що система робить)



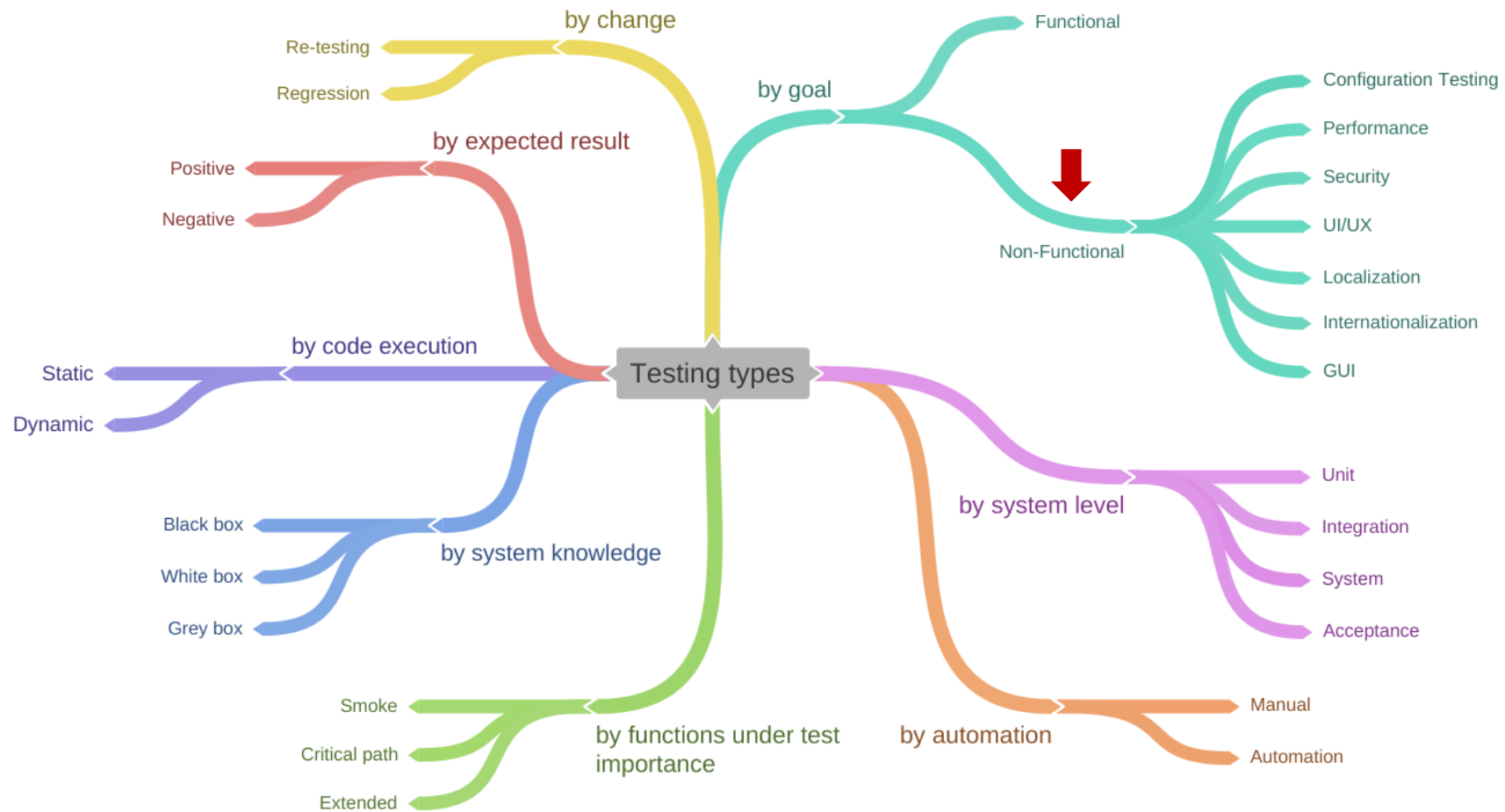
При натисканні педалі газу
автомобіль прискорюється

При натисканні на кнопку
запалюється ліхтарик



Manual QA

Види тестування



Нефункціональне тестування. Безпека

- **Аутифікація (authentication)** — процедура перевірки справжності, наприклад перевірка справжності користувача шляхом порівняння введеного ним пароля з паролем, збереженим у базі даних.
- **Авторизація (authorization)** — надання певній особі або групі осіб прав на виконання певних дій.



Нефункціональне тестування. Продуктивність

Performance testing – тестування, яке проводиться з метою визначення, як швидко працює програмне забезпечення або його частина під певним навантаженням.

Для підтвердження результатів цього виду тестування використовуються метрики.

ТЕСТУВАННЯ
НАВАНТАЖЕННЯ



автоматизоване тестування, що імітує роботу певної кількості бізнес-користувачів на якомусь загальному (розділюваному ними) ресурсі

СТРЕС-ТЕСТУВАННЯ



дозволяє перевірити, наскільки застосунок та система загалом працездатні в умовах підвищення інтенсивності виконання операцій до дуже високих значень. Зазвичай перевіряє ступінь стійкості ПЗ

ТЕСТУВАННЯ
СТАБІЛЬНОСТІ



перевірка працездатності застосунку при тривалому (багатодинному) тестуванні із середнім рівнем навантаження.

Нефункціональне тестування. Користувацький інтерфейс

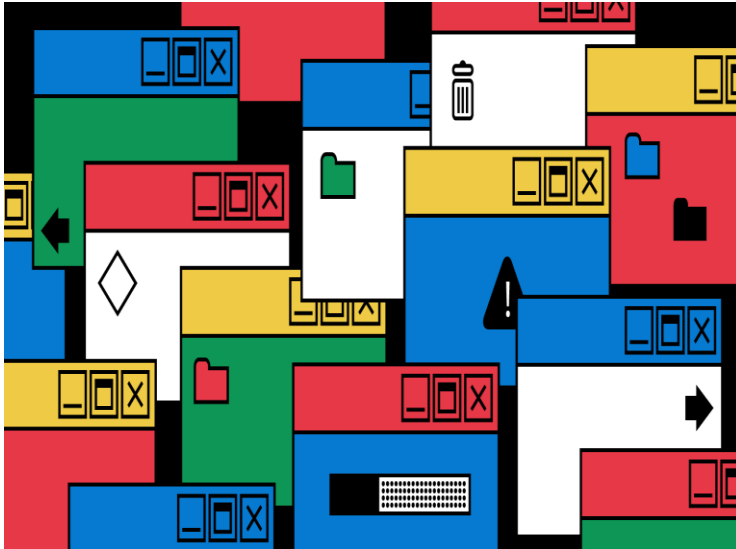


User Interface – відноситься до екранів, кнопок, перемикачів, значків та інших візуальних елементів, з якими ви взаємодієте при використанні вебсайту, застосунку або іншого електронного пристрою.



User Experience – це те, який досвід/враження отримує користувач від роботи з вашим інтерфейсом. Чи вдається йому досягти мети і наскільки просто чи складно це зробити.

Нефункціональне тестування. GUI



Тестування графічного інтерфейсу користувача (GUI testing) – це процес тестування графічного користувацького інтерфейсу продукту, щоб переконатися, що він відповідає специфікаціям.

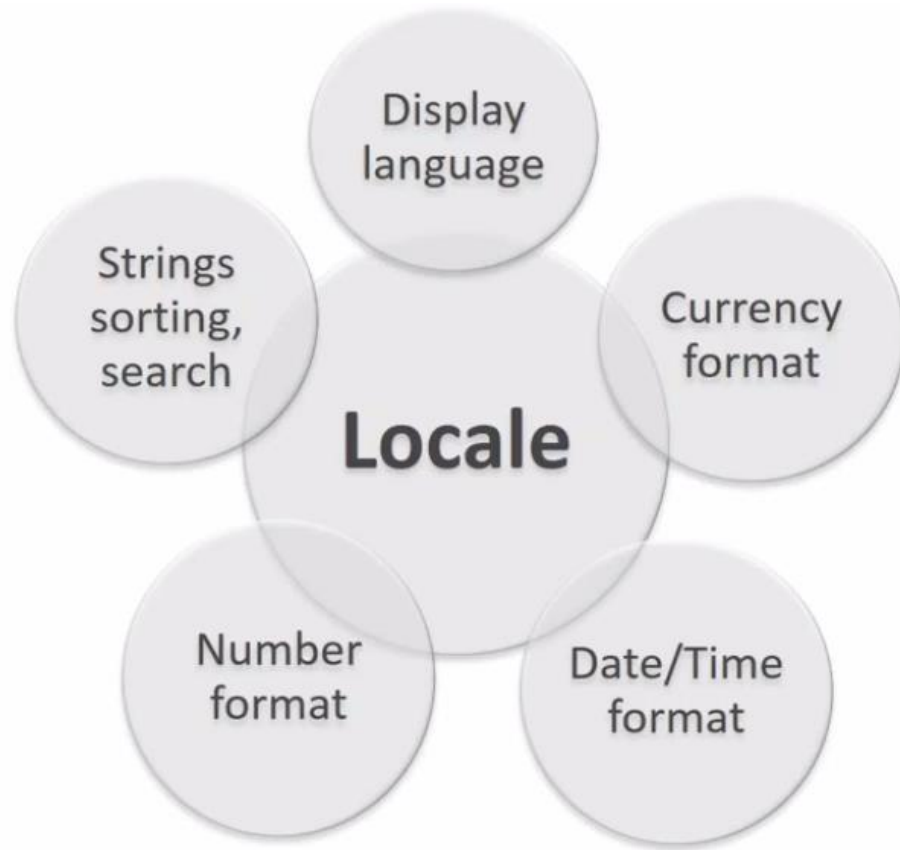
- Навігація
- Форматування
- Колір та шрифти
- Смуги прокручування
- Елементи управління та вирівнювання
- Орфографія та граматики
- Клавіші за замовчуванням та поєднання клавіш
- Поведінка елементів керування
- Контраст
- Зображення
- Повідомлення про помилки

Нефункціональне тестування. Інсталюваність

Конфігураційне тестування (Configuration testing) — спеціальний вид тестування, спрямований на перевірку роботи програмного забезпечення при різних конфігураціях системи (заявлених платформах, драйверах, що підтримуються, при різних конфігураціях комп'ютерів тощо).

Тестування локалізації (Localization testing) — тестування, спрямоване на перевірку коректності та якості адаптації продукту до використання тією чи іншою мовою з урахуванням національних та культурних особливостей.

Нефункціональне тестування. Інтернаціоналізація

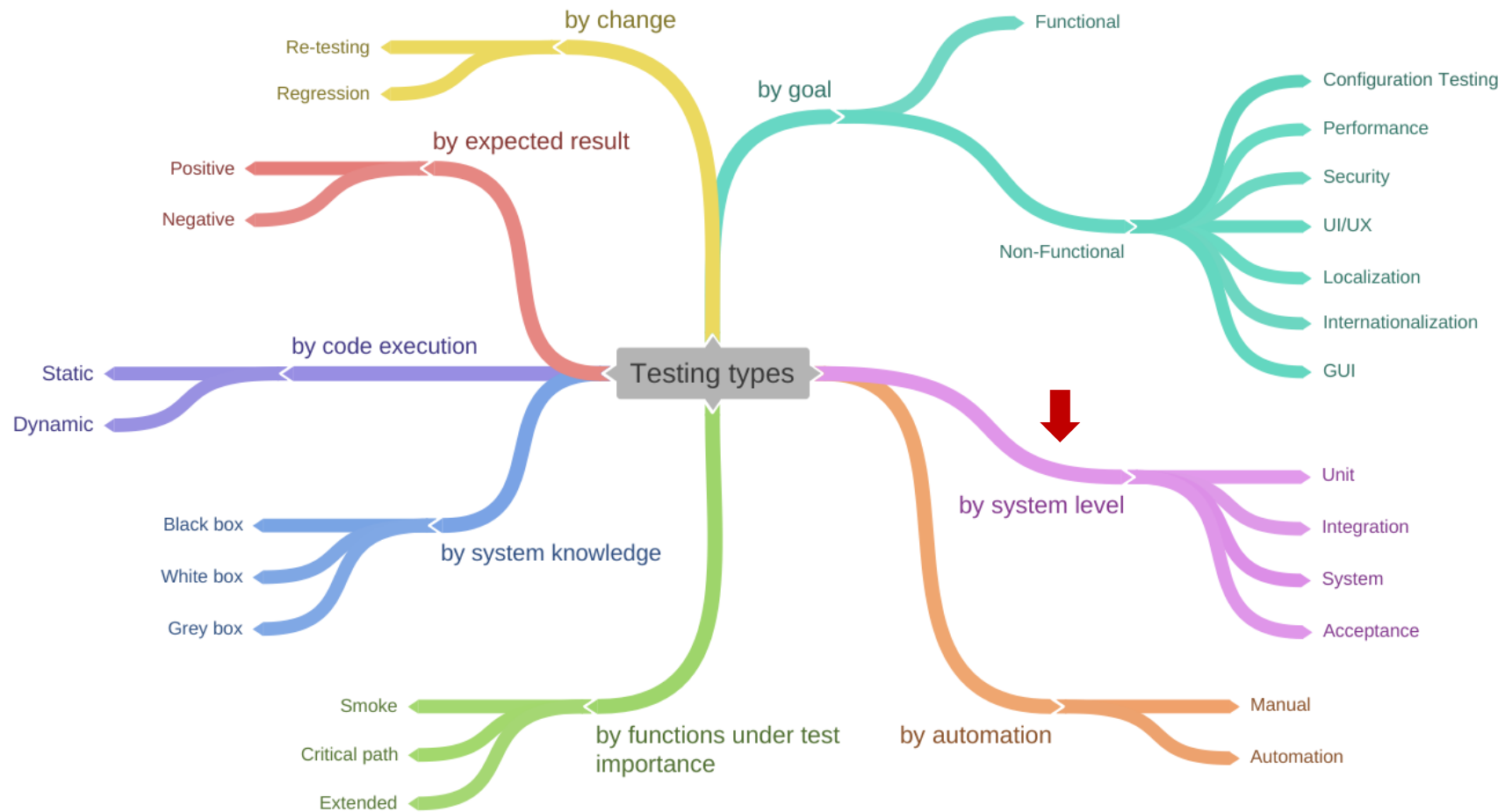


Тестування інтернаціоналізації (internationalization testing, i18n testing) – тестування, спрямоване на перевірку готовності продукту до роботи з використанням різних мов і з урахуванням різних національних та культурних особливостей.

Цей вид тестування не має на увазі перевірки якості відповідної адаптації.

Manual QA

Види тестування



Manual QA

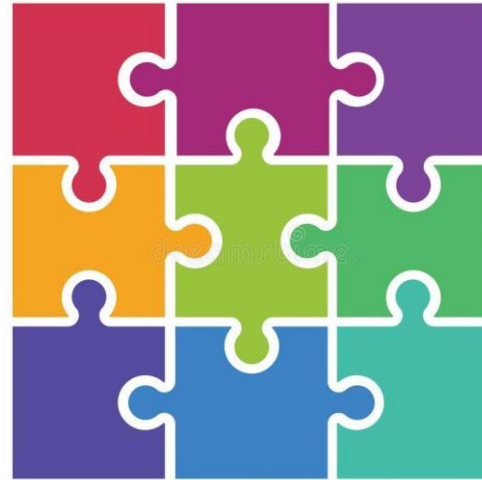
РІВНІ ТЕСТУВАННЯ



Компонентний



Інтеграційний



Системний



Приймальний

За рівнем системи. Рівні тестування

Модульне (компонентне) тестування (Unit testing) — перевіряються окремі невеликі частини застосунку.

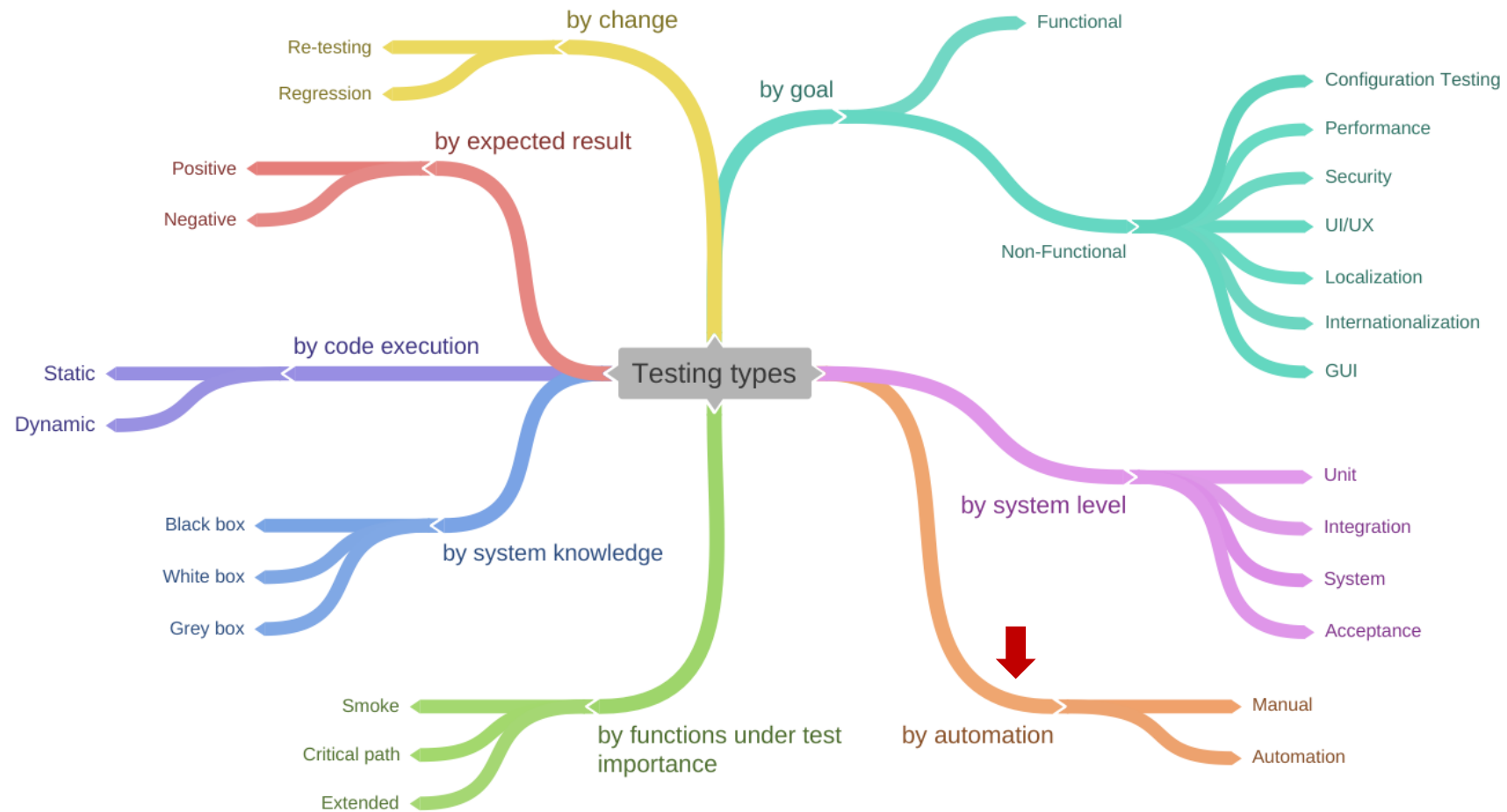
Інтеграційне тестування (Integration testing) — перевіряється взаємодія між декількома частинами застосунку.

Системне тестування (System Testing) — застосунок перевіряється як єдине ціле.

Приймальне тестування (Acceptance testing) — тестування, спрямоване на перевірку застосунку з точки зору кінцевого користувача/замовника та винесення рішення про те, чи приймає замовник роботу у виконавця (проектної команди).

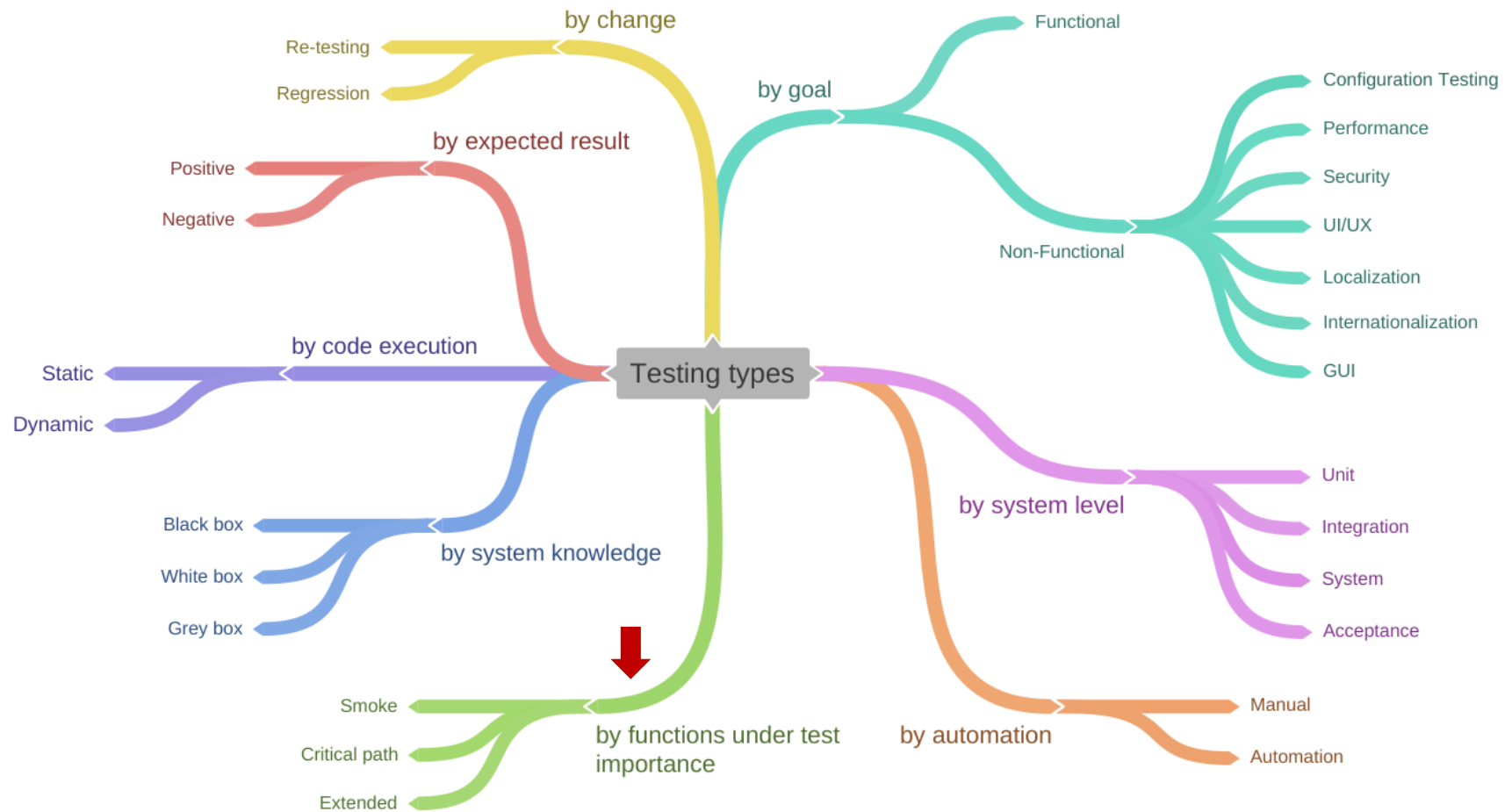
Manual QA

Види тестування



Manual QA

Види тестування



Manual QA

За ступенем важливості функцій, що тестуються

Димове тестування (Smoke testing) — перевірка найважливішої функціональності програмного продукту.

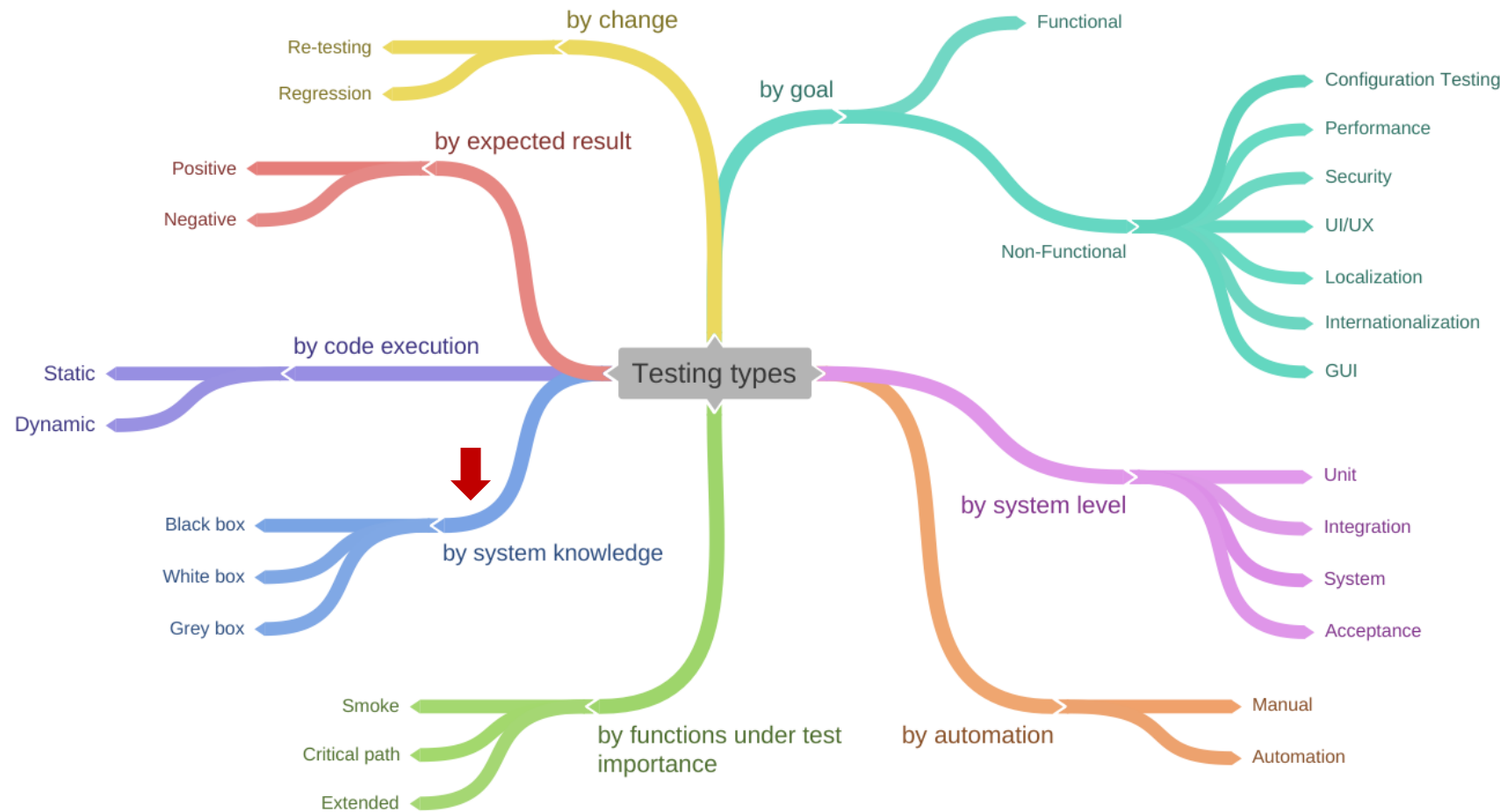
Sanity testing — це базова перевірка працездатності певного функціоналу

Тестування критичного шляху (critical path testing) — перевірка функціональності, що використовується типовими користувачами у повсякденній діяльності.

Розширене тестування (extended testing) — перевірка всієї заявленої функціональності.

Manual QA

Види тестування



Manual QA

За доступом до коду та архітектури застосунку

Тестування чорної скриньки



Тестування не передбачає
знання внутрішнього
влаштування компоненту чи
системи

Тестування білої скриньки



Тестування, засноване на аналізі
внутрішньої структури
компоненту чи системи

Тестування сірої скриньки



Комбінація White Box та Black Box
підходів, доступ до внутрішньої
структури та алгоритмів роботи ПЗ
для написання максимально
ефективних тест-кейсів

Manual QA

Практика

Опис завдання:

Валідатор паролів повинен перевіряти вхідний пароль на відповідність заданим правилам безпеки.

Правила безпеки для паролів:

1. Пароль повинен містити щонайменше 8 символів.
2. Пароль повинен містити щонайменше одну велику літеру.
3. Пароль повинен містити щонайменше одну малу літеру.
4. Пароль повинен містити щонайменше одну цифру.
5. Пароль може містити спеціальні символи (!, @, #, \$, %, ^, &, *).

Завдання:

Визначте класи еквівалентності для валідатора паролів на основі правил безпеки.

Що ми сьогодні вивчали

Manual QA

План уроку

- Види тестування
 - за запуском коду на виконання;
 - за доступом до коду та архітектури застосунку;
 - за рівнем деталізації застосунку;
 - за рівнем автоматизації;
 - за очікуваним результатом;
 - за ступенем важливості функцій, що тестуються;
 - за цілями та завданнями;
- Рівні тестування
- Практика

Manual QA

Підсумки

Що одне, найголовніше, ви дізнались сьогодні?

Manual QA

Дякую за увагу! До нових зустрічей!

Інформаційний відеосервіс для розробників програмного забезпечення

