TestHuset

TEST TRENDS

WHAT DOES QUALITY MEAN?

Fit for purpose?

Does it work?

Does it provide value?

faster?

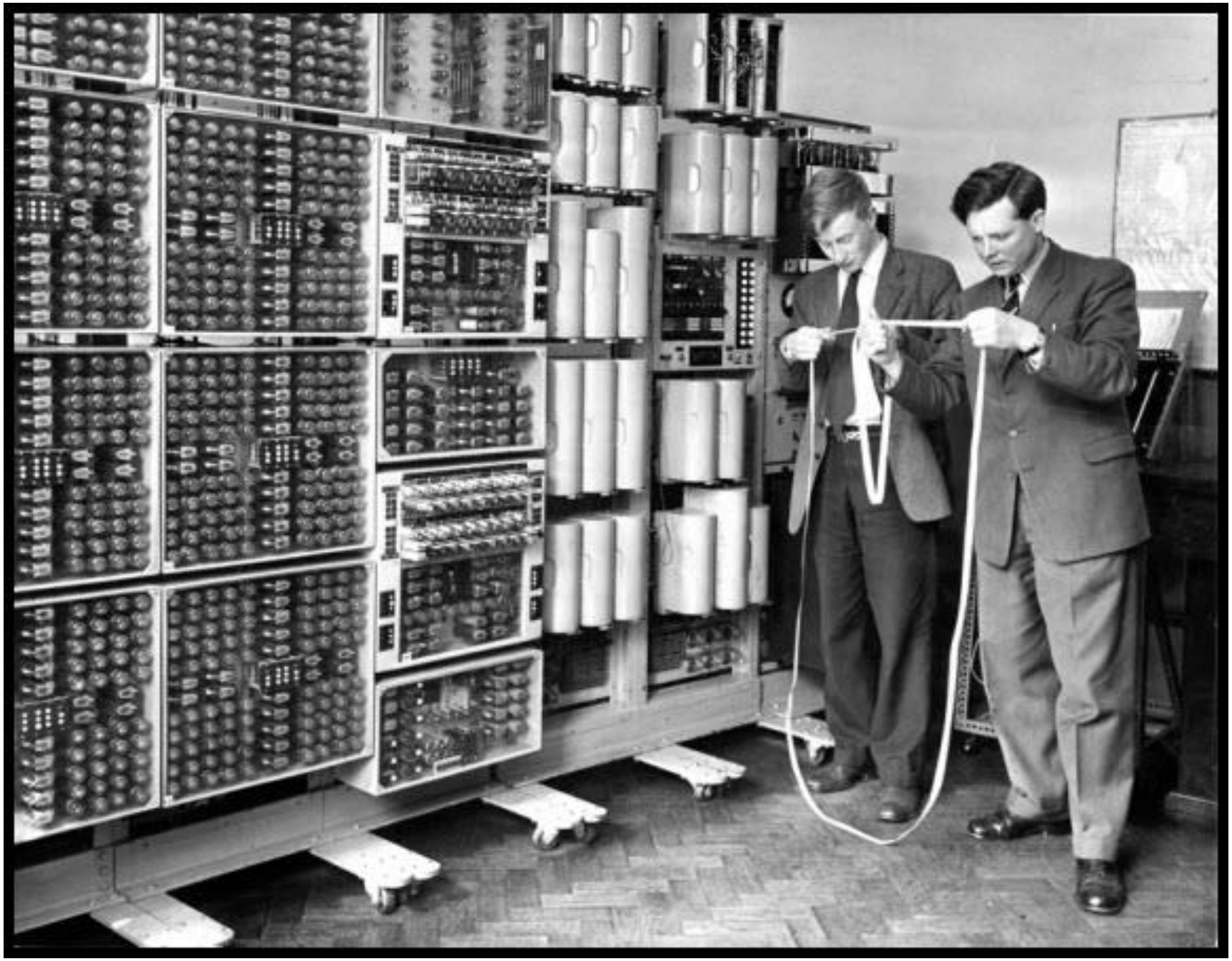
better?

safer?

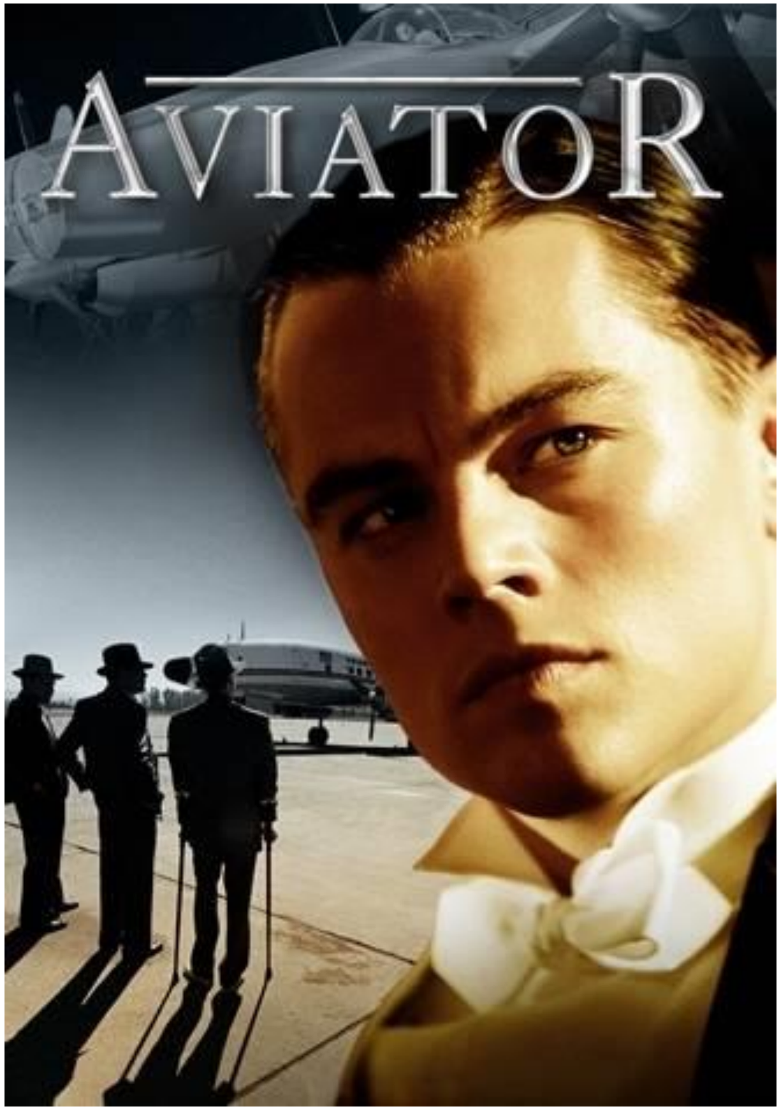
Does it make us better? cooler?

TestHuset

WHAT HAVE WE LEARNED FROM HISTORY?



TESTING…

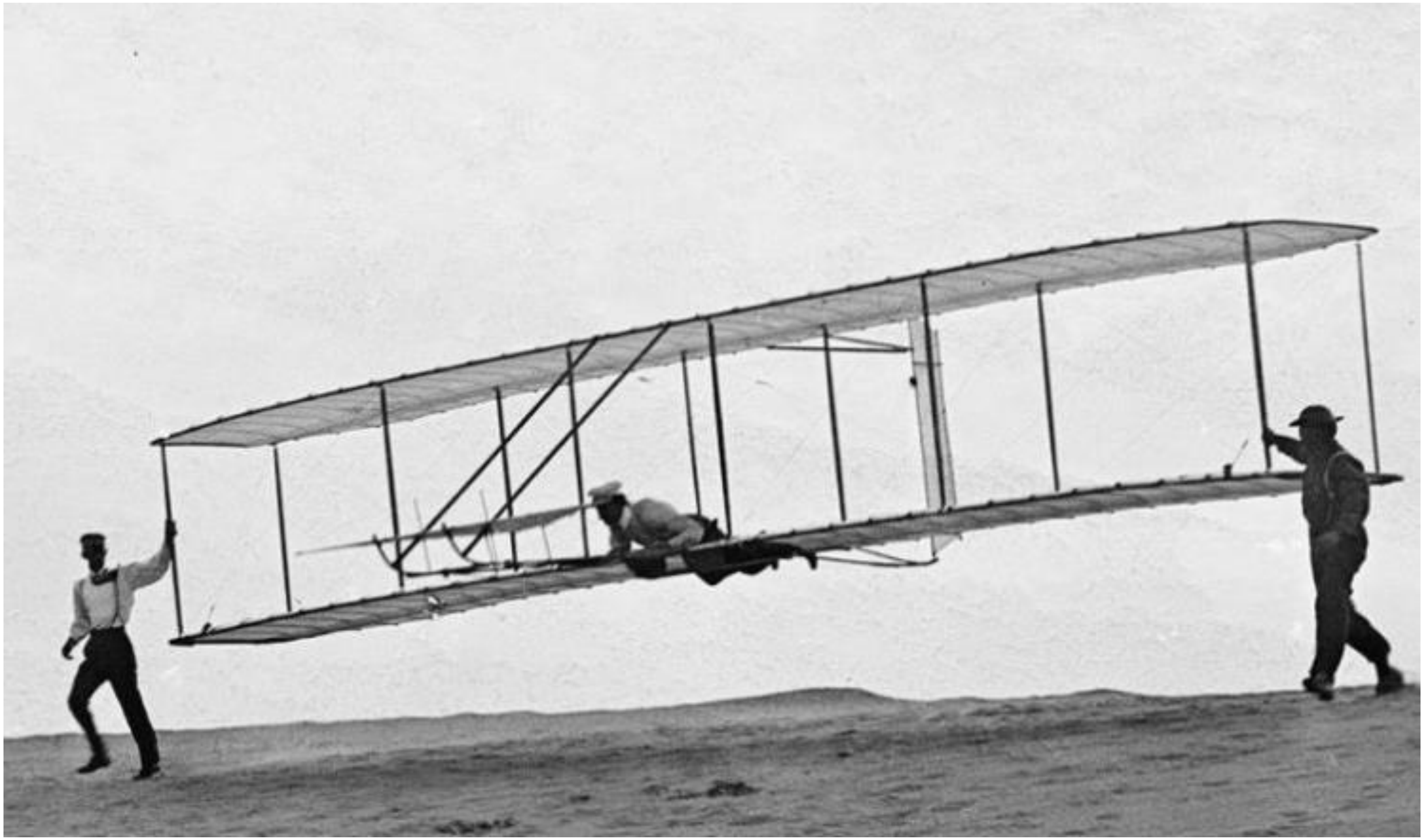


TestHuset

…IN A QUEST FOR BUGS

IEEE 829 / ISO 29119

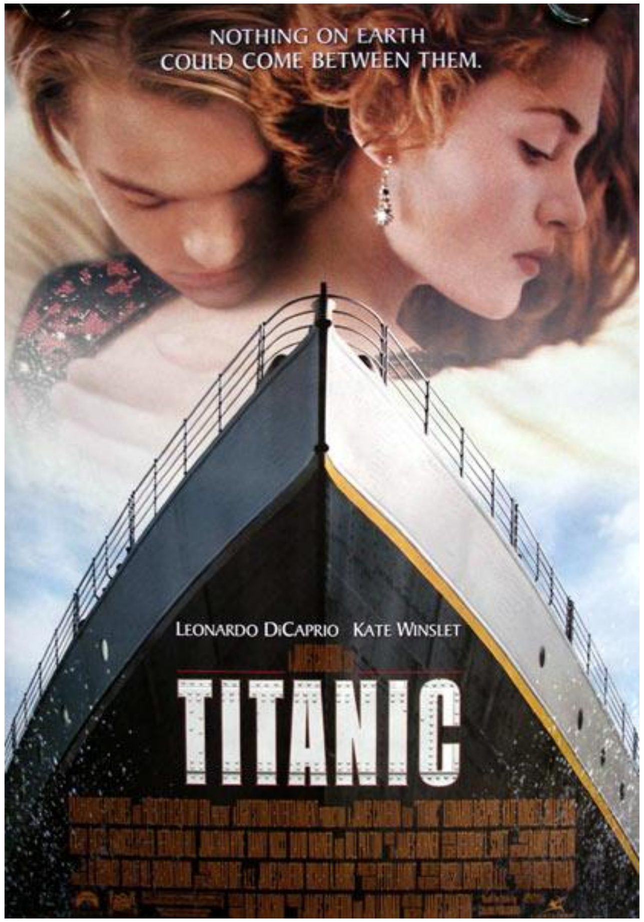
EARLY TEST…



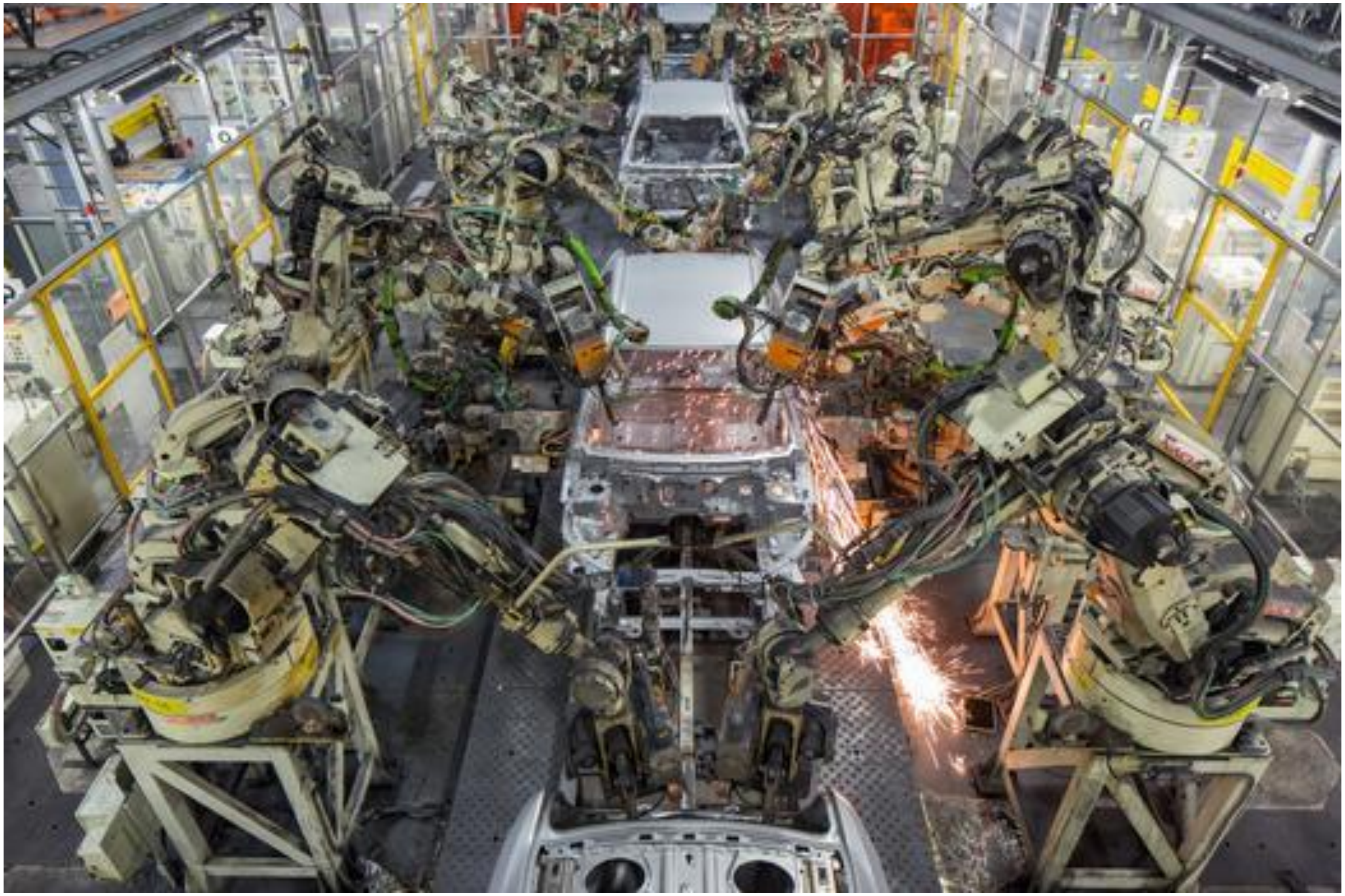
…TO FAIL FAST AND STAY SAFE

https://www.simulationinformation.com/education/early-history-flight-simulation

BUILD THE RIGHT PRODUCT



BUILD THE PRODUCT RIGHT



http://toyotanews.pressroom.toyota.com/releases/toyota+500k+corolla+line+off+tmmms.htm

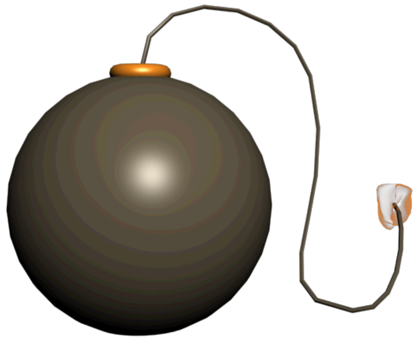
KEEP THE PRODUCT RIGHT

FEEDBACK_

MARKET ADOPTION? _

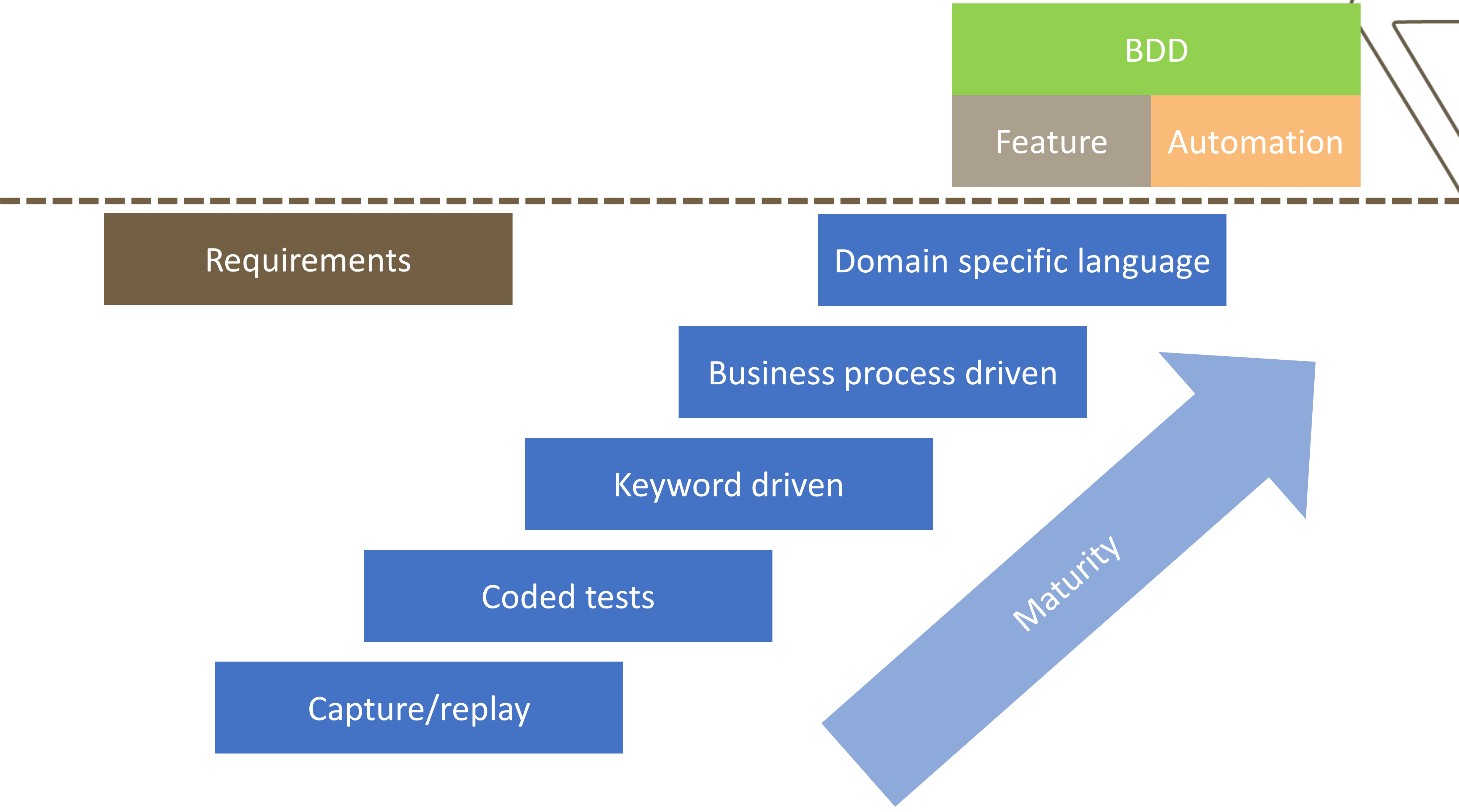
PRODUCT PERFORMANCE? _

ACCIDENTS, DISASTERS
AND BAD LUCK _



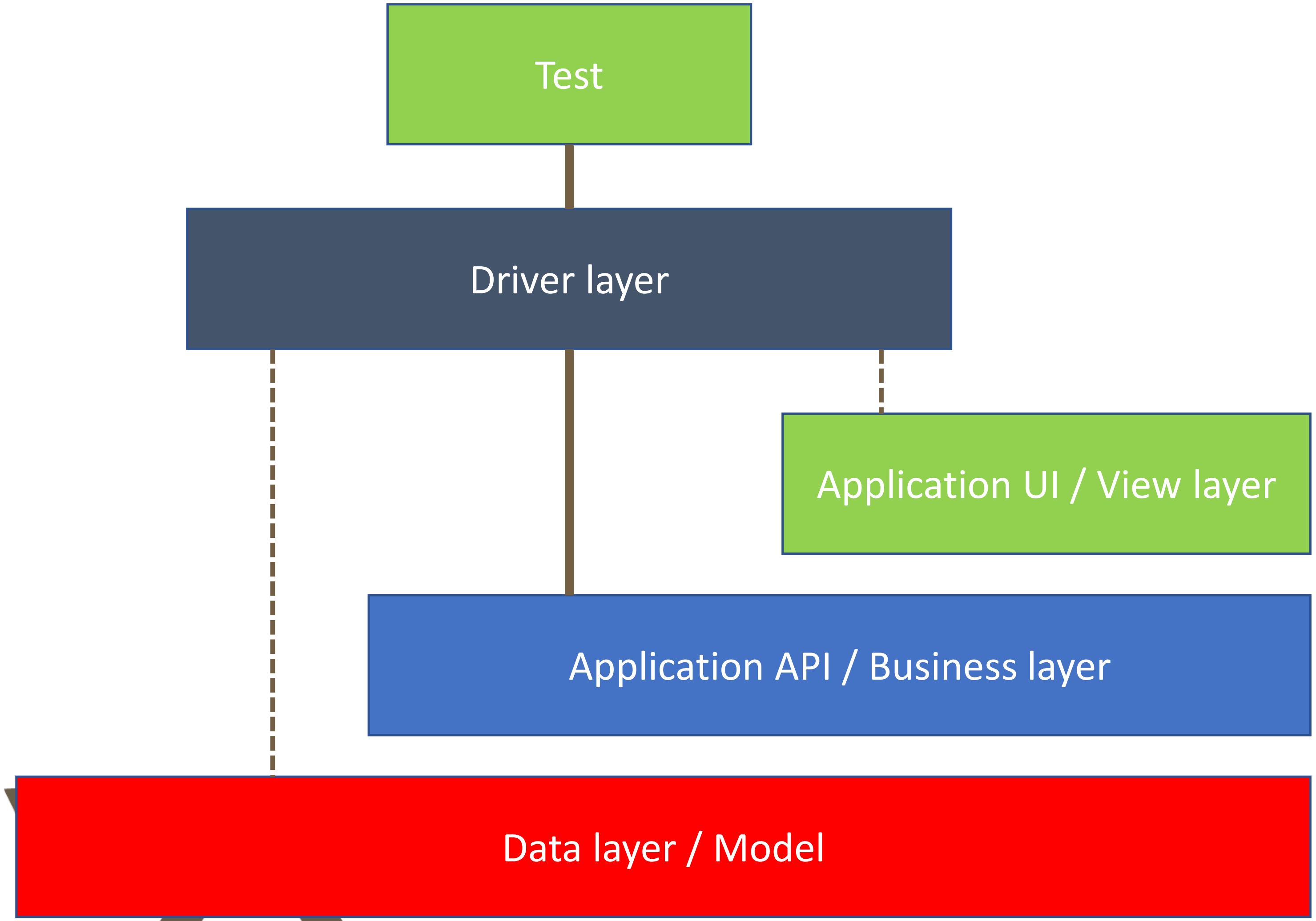
TestHuset

BLACK BOX AUTOMATION



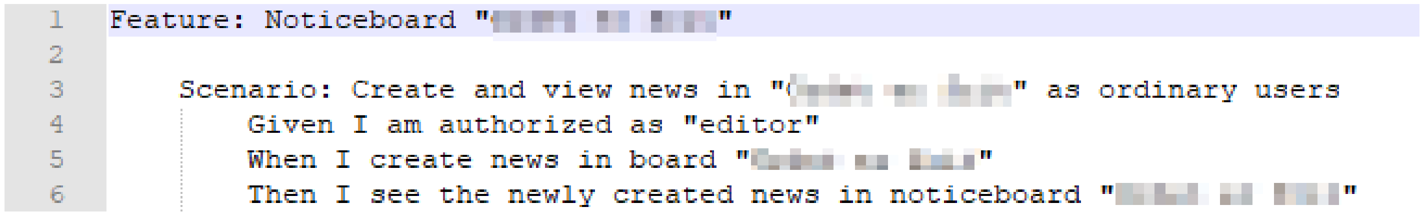
TestHuset

TEST AUTOMATION



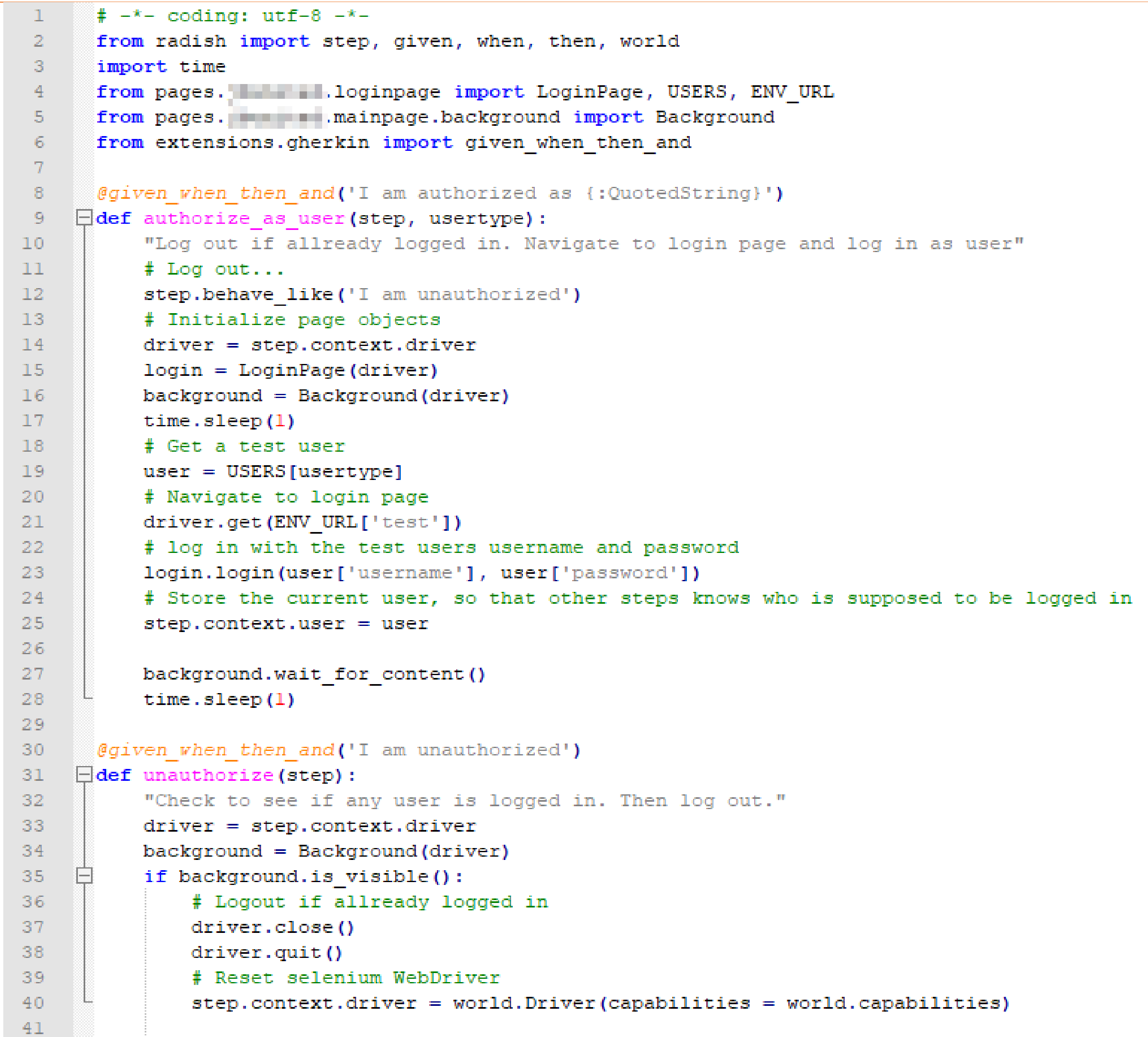
TestHuset

SCENARIO



```
Feature: Noticeboard "[redacted]"

    Scenario: Create and view news in "[redacted]" as ordinary users
        Given I am authorized as "editor"
        When I create news in board "[redacted]"
        Then I see the newly created news in noticeboard "[redacted]"
```

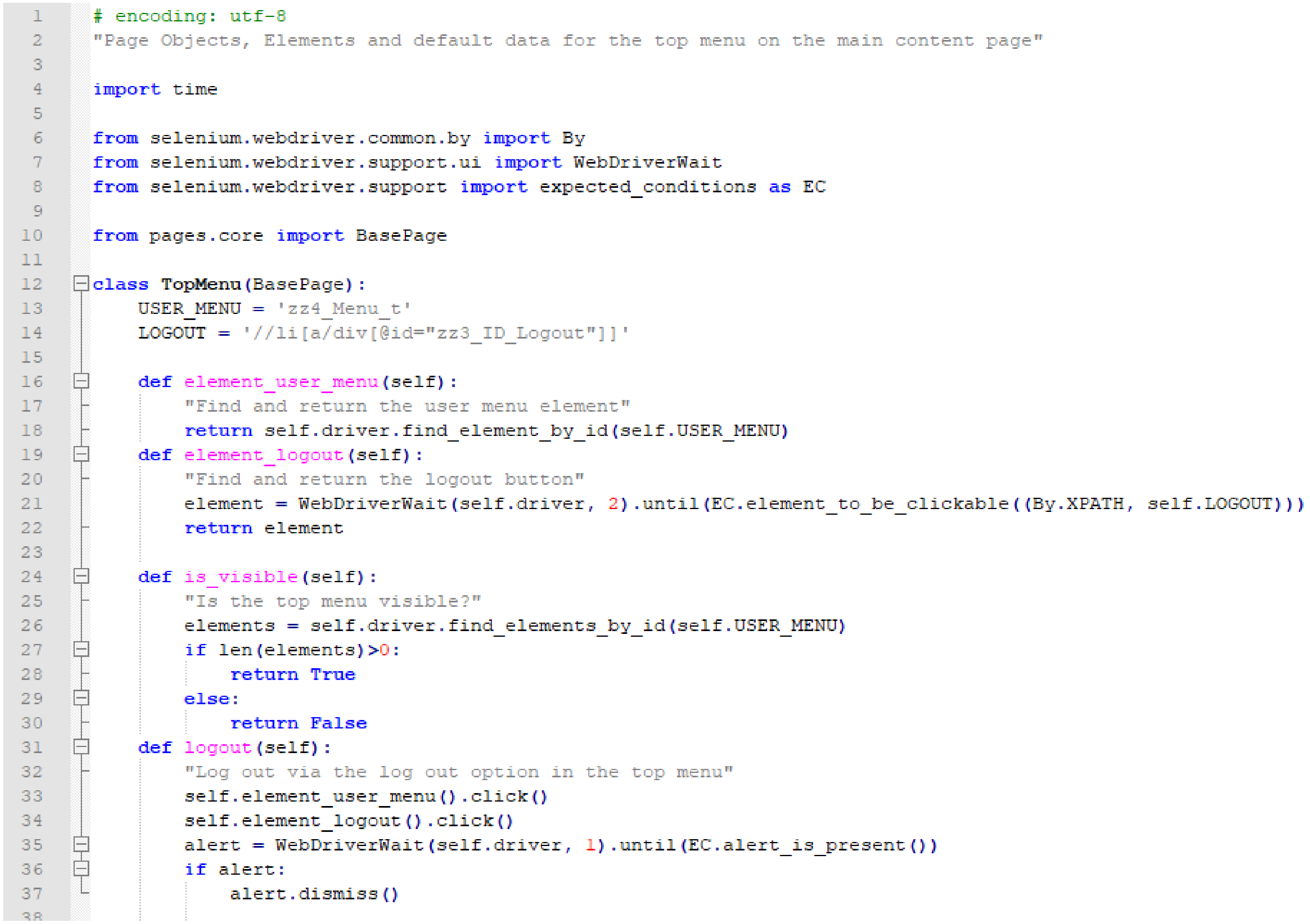
TEST STEPS



```python
# -*- coding: utf-8 -*-
from radish import step, given, when, then, world
import time
from pages.[redacted].loginpage import LoginPage, USERS, ENV_URL
from pages.[redacted].mainpage.background import Background
from extensions.gherkin import given_when_then_and

@given_when_then_and('I am authorized as {:QuotedString}')
def authorize_as_user(step, usertype):
    "Log out if allready logged in. Navigate to login page and log in as user"
    # Log out...
    step.behave_like('I am unauthorized')
    # Initialize page objects
    driver = step.context.driver
    login = LoginPage(driver)
    background = Background(driver)
    time.sleep(1)
    # Get a test user
    user = USERS[usertype]
    # Navigate to login page
    driver.get(ENV_URL['test'])
    # log in with the test users username and password
    login.login(user['username'], user['password'])
    # Store the current user, so that other steps knows who is supposed to be logged in
    step.context.user = user

    background.wait_for_content()
    time.sleep(1)

@given_when_then_and('I am unauthorized')
def unauthorize(step):
    "Check to see if any user is logged in. Then log out."
    driver = step.context.driver
    background = Background(driver)
    if background.is_visible():
        # Logout if allready logged in
        driver.close()
        driver.quit()
        # Reset selenium WebDriver
        step.context.driver = world.Driver(capabilities = world.capabilities)

```

TestHuset

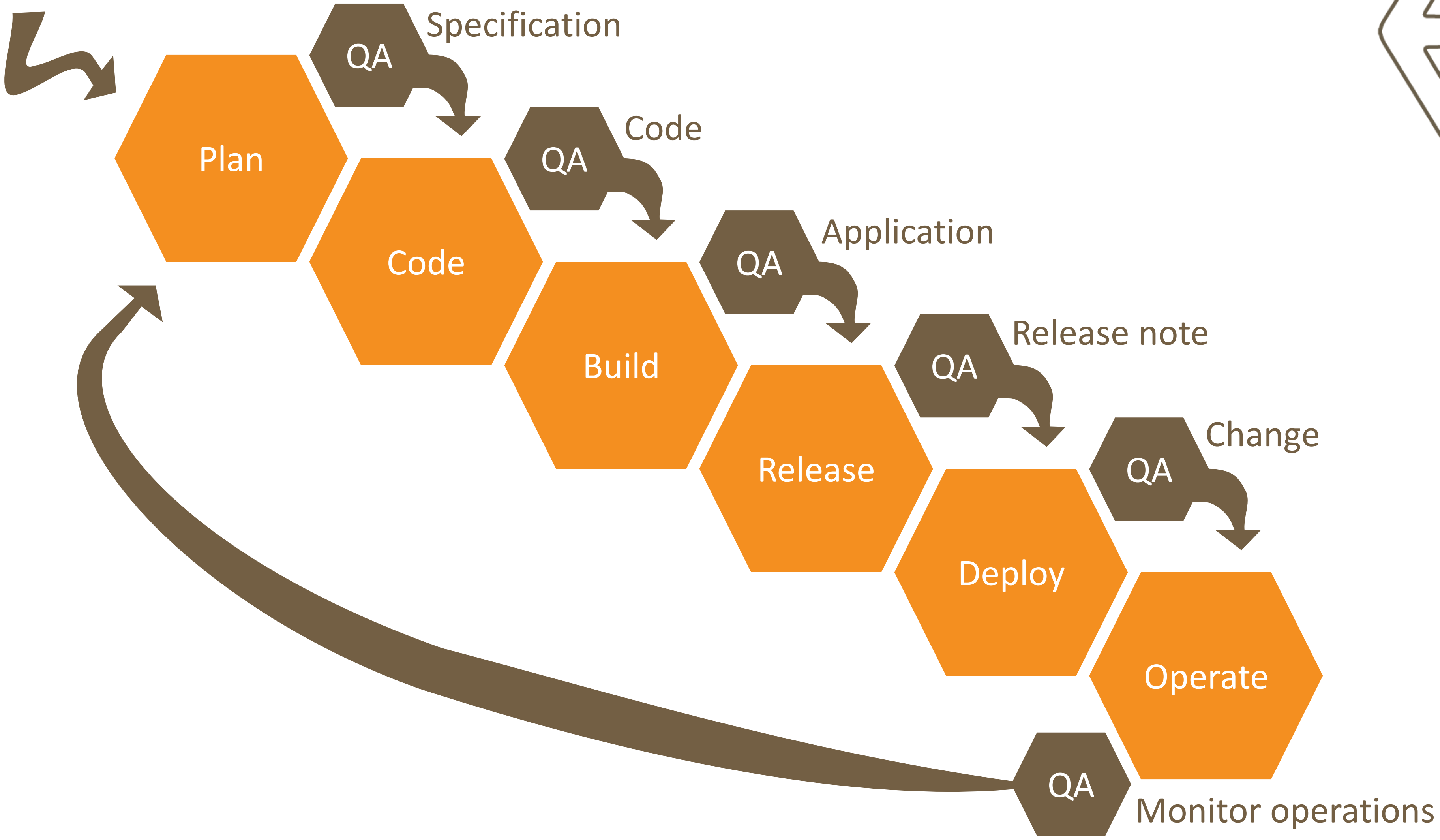
PAGE OBJECTS



```python
# encoding: utf-8
"Page Objects, Elements and default data for the top menu on the main content page"

import time

from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

from pages.core import BasePage

class TopMenu(BasePage):
    USER_MENU = 'zz4_Menu_t'
    LOGOUT = '//li[a/div[@id="zz3_ID_Logout"]]'

    def element_user_menu(self):
        "Find and return the user menu element"
        return self.driver.find_element_by_id(self.USER_MENU)
    def element_logout(self):
        "Find and return the logout button"
        element = WebDriverWait(self.driver, 2).until(EC.element_to_be_clickable((By.XPATH, self.LOGOUT)))
        return element

    def is_visible(self):
        "Is the top menu visible?"
        elements = self.driver.find_elements_by_id(self.USER_MENU)
        if len(elements)>0:
            return True
        else:
            return False
    def logout(self):
        "Log out via the log out option in the top menu"
        self.element_user_menu().click()
        self.element_logout().click()
        alert = WebDriverWait(self.driver, 1).until(EC.alert_is_present())
        if alert:
            alert.dismiss()
```

TestHuset

DELIVERY PIPELINE



TestHuset