Testowanie automatyczne I - Wprowadzenie

Wykład 1

Czym jest testowanie Automatyczne "Testing is the process of exercising or evaluating a system or system component by manual or automated means to verify that it satisfies specified requirements." [IEEE 83a]

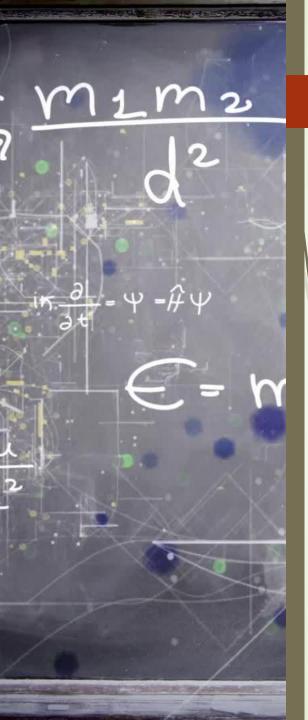
"Software testing is the process of executing a program or system with the intent of finding errors." [Myers]

"It involves any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results." [Hetzel]



Czym jest testowanie Automatyczne

- ► ISQB
 - Using purpose-built software tools to control and set up test preconditions
 - Executing tests
 - Comparing actual outcomes to predicted outcomes



Czym jest testowanie Automatyczne

Jest to testowanie przy wykorzystaniu narzędzi które w sposób autonomiczny odtwarzają scenariusz testowy, przeprowadzają asercje i publikują wynik przeprowadzenie testów.

Czemu testujemy Automatycznie

- Optymalizacja procesu wytwarzania oprogramowania
 - Po wprowadzeniu zmian developer lub tester może zweryfikować czy jego zmiana nie spowodowała regresji
- Wymagany element sprawnie działającego CI/CD
 - System jest podawany testom automatycznym przed każdą publikacją produkcyjną i/lub po niej
- Ujednolicenie wykonywania testów by uniknąć ludzkich błędów
 - Człowiek może ignorować mniej ważne kroki lub zachowania np. w walidowanej komunikacji pojawiają się 2 spacje zamiast jednej przed słowem klucz.

Czemu testujemy Automatycznie

- Testowanie na szerszej ilości środowisk
 - Aplikacja mobilna testowana na wersji Android 7,8,9
- Wykonanie trudnych/drogich testów do przeprowadzenia manualnego testowania
- Zwiększenie Zaufania do dostarczonego oprogramowania

- Wymóg jednoznaczności wyników
 - Test który odbywa się na tej samej wersji oprogramowania w tym samy środowisku testów powinien zwrócić ten sam wynik za każdym razem.
 - Jeżeli testy nie mają jednoznacznych wyników oznacza to:
 - Zaniżenie zaufania do testów przez zespół
 - Zwiększenie kosztu testowania przez:
 - Potrzebę powtarzania testów
 - Manualne testowanie przypadków
 - Dodatkowy koszt analizy wyników
 - Spowolnienie w wydaniu wersji

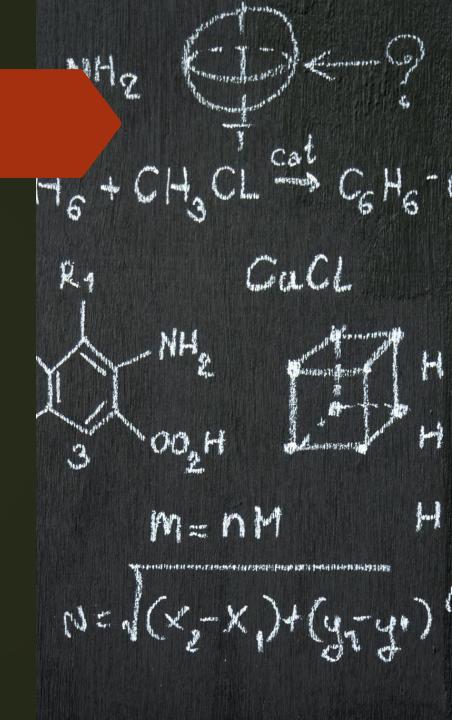
- Wymóg jednoznaczności wyników
- Testowanie automatyczne jest kosztowne w wszystkich stadiach swojego życia.
 - Wynika to z:
 - Kosztu powstania zalążków
 - Kosztu integracji w obecny proces
 - Kosztu wykonywania testów
 - Kosztu utrzymania istniejących testów
 - Kosztu regularnej kontroli jakości testów
 - Nawet zakładając że projekt który jest testowany się nie zmienił to środowisko odbywania się testów mogło ulec zmianie

- Wymóg jednoznaczności wyników
- Testowanie automatyczne jest kosztowne w wszystkich stadiach swojego życia.
- Czas egzekucji testów i ich skala
 - Testy wykonujące się wiele godzin mogą nie nadawać się do specyficznych użyć np. weryfikacji codziennej paczki produkcyjnej
 - Czym więcej testów mamy tym większa szansa pojawienia się nie jednoznaczności wyników i wzrost kosztu utrzymania

- Wymóg jednoznaczności wyników
- Testowanie automatyczne jest kosztowne w wszystkich stadiach swojego życia.
- Czas egzekucji testów i ich skala
- Syndrom pestycydu
 - Testowanie tej samej przestrzeni w pewnym momencie może traci wartość i wprowadzać w fałszywe poczucie braku wad w oprogramowaniu

Czym testowanie Automatyczne nie jest

- Nie jest zastępstwem dla testów manualnych
 - Nie ma wyobraźni
 - Nie zastąpi testów Eksploracyjnych



Metodyki wytwórcze a testowanie Automatyczne



Pod warunkiem że mamy nie ograniczony dostęp do zasobów każdy jednoznacznie powtarzalny przypadek testowy jesteśmy wstanie z automatyzować ale czy powinniśmy?



Tak jak testowanie manualne tak jak i testowanie automatyczne podlega restrykcjom modelów wytwórczych oprogramowania. Uproszczona Piramida testowa Mike Cohn

E2E

Integracyjne

Jednostkowe

Uproszczona Piramida testowa Mike Cohn

- E2E (End to End) Kompleksowe testy systemowe skupiające się na interakcjach użytkownika końcowego symulując jego zachowania od początku do końca.
- Testy integracyjne wykonywany w celu wykrycia defektów w interfejsach i interakcjach pomiędzy modułami lub systemami.
- Jednostkowe testowanie najmniejszej testowalnej (zamkniętej) jednostki oprogramowania

Przypadki testowe

Odpowiedzenie na pytanie co zamierzam przetestować ma elementarne znaczenie dla testów automatycznych bo od odpowiedzi na te pytanie zależy nie tylko metodyka testów ale też wybór narzędzia i technika.

Przypadki testowe

- Które przypadki automatyzujemy najczęściej?
 - Najważniejsze
 - Np. zabezpieczające główne ścieżki użycia aplikacji
 - Testy łatwe do zautomatyzowania
 - Testy które zwrócą się nam najszybciej
 - Testy które wykonujemy najczęściej

Techniki Automatyzacji scenariuszy

- Behaviour Driven Testing
 - Z rodziny Behaviour Driven Development
 - Wpierw powstają testy dla funkcjonalności a potem funkcjonalność
 - Scenariusz opisuje zachowania
 - Np.
 - Jeżeli jestem na ekranie produktu,
 - Gdy kliknę dodaj produkt do koszyka,
 - Wtedy Produkt jest dodany do koszyka

Techniki Automatyzacji scenariuszy

- Behaviour Driven Testing
- Keyword Driven Testing
 - Wybieramy słowo klucz które definiuje funkcjonalność lub akcje
 - Np. Dodaj przedmiot do koszyka, sprawdź czy przedmiot jest w koszyku

Techniki Automatyzacji scenariuszy Behaviour, Keyword i step table są to sposoby opisu przypadku testowego i stworzone by nie tylko usystematyzować opis ale również poprawić czytelność i przejrzystość procesu testowania dla zespołu.

Techniki ograniczania ilości przypadków testowych stosujemy niezależnie od sposobu opisu przypadków testowych. Korzystamy z nich po to by zagwarantować pokrycie funkcjonalność przypadkami testowymi.

Lieretura

- Beizer, B. (1990) Software Testing Techniques (2e), Van Nostrand Reinhold: Boston MA
- Black, R. (2009) Managing the Testing Process (3e), John Wiley & Sons: New York NY
- Buwalda, H. et al. (2001) Integrated Test Design and Automation, Addison Wesley: Reading MA
- Copeland, L. (2004) A Practitioner's Guide to Software Test Design, Artech House: Norwood MA

Literatura

- Craig, R. and Jaskiel, S. (2002) Systematic Software Testing, Artech House: Norwood MA
- Fewster, M. and Graham, D. (1999) Software Test Automation, Addison Wesley: Harlow UK
- Graham, D. and Fewster, M. (2012) Experiences of Test Automation, Pearson Education: Boston MA
- Kaner, C., Bach, J. and Pettichord, B. (2002) Lessons Learned in Software Testing, John Wiley & Sons: New York NY
- Kaner, C., Padmanabhan, S. and Hoffman, D. (2013) The Domain Testing Workbook, Context-Driven Press: New York NY
- Wiegers, K. (2002) Peer Reviews in Software, Pearson Education: Boston MA
- Weinberg, G. (2008) Perfect Software and Other Illusions about Testing, Dorset House: New York NY
- Smilgin, R. (2016) Zawód Tester, Wydawnictwo Naukowe PWN SA
- Cieciura, M. (2009) Wybrane Problemy Społeczne i Zawodowe Informatyki, Wydawnictwo Sowa
- RAJIV, C. (2018) Software Testing A Self-Teaching Introduction, Mercury Learning and Information
- Arnon A. (2020) Automatyzacja testów, Wydawnictwo Naukowe PWN SA