

Kod szkolenia: **360TEST**

Tytuł szkolenia: **Testowanie 360 - wszystko o testowaniu**

Dni: 5

Opis:

Adresaci szkolenia:

Szkolenie jest adresowane do początkujących testerów, którzy chcą poznać całościowy obraz pojęć i technik związanych z testowaniem oprogramowania

Cel szkolenia:

Przedstawienie praktycznych zagadnień związanych z testowaniem Przedstawienie rodzajów testowania Przedstawienie technik i narzędzi związanych z testowaniem funkcjonalnym, wydajnościowym, bezpieczeństwa i zapewnieniem jakości kodu

Mocne strony szkolenia:

Mocną stroną szkolenia jest rozbudowana część warsztatowa, która pozwala na przedstawienie praktycznych zagadnień związanych z testowaniem oraz rozmaitych pułapek czyhających na testerów w codziennej pracy. Obejmuje ona następujące zagadnienia

- Zaprojektowanie rodzajów testów dla przykładowych aplikacji
- Zaprojektowanie przypadków testowych dla przykładowych aplikacji
- Diagnozowanie problemów przy pomocy narzędzi programistycznych w przeglądarce internetowej: Firebug, Chrome Developer Tools
- Praca z aplikacją Java na serwerze Linux - testowanie i diagnozowanie problemów
- Selenium jako narzędzie do automatyzacji testów interfejsu
- wykorzystanie Selenium IDE jako narzędzia do prototypowania i usprawniania pracy testera
- RobotFramework i Selenium jako framework do automatyzacji typowej aplikacji internetowej
- Test usługi SOAP przy użyciu SoapUI
- Test usługi REST przy użyciu SoapUI
- analiza zagrożeń - przykład
- wykorzystanie narzędzia Findbugs+find-sec-bugs/PMD+GDS-Security-Rules do analizy statycznej kodu
- wykorzystanie checklist CIS oraz narzędzia Lynis do analizy konfiguracji
- narzędzie OWASP Dependency Checker i victi.ms - analiza komponentów
- test penetracyjny przykładowej aplikacji internetowej z wykorzystaniem OWASP

ZAP/Burp Suite oraz narzędzi z zestawu narzędzi Kali Linux

- Apache JMeter - wykorzystanie do automatyzacji testów wydajnościowych aplikacji internetowej
- Jenkins jako system ciągłej integracji
- SonarQube jako narzędzie do badania jakości kodu źródłowego
- Findbugs i PMD - wykorzystanie gotowych i zbudowanie własny reguł
- Junit i TestNG - narzędzia do testów jednostkowych Java
- Nunit - narzędzie do testów jednostkowych .NET
- Mocha - narzędzie do testów jednostkowych JavaScript

Wymagania:

Biegłe posługiwanie się komputerem, znajomość podstaw pracy w systemie Windows oraz pracy z aplikacjami internetowymi. Wskazana znajomość systemu Linux na poziomie użytkownika aplikacji.

Parametry szkolenia:

5*8 godzin (5*7 godzin netto) wykładów i warsztatów (z wyraźną przewagą warsztatów).

Program szkolenia:

1. Wprowadzenie

- techniki testowania
 - testy na poziomie kodu źródłowego (whitebox)
 - testy aplikacji typu blackbox i greybox
 - wprowadzanie błędów do środowiska (ang. fault-injection) jako technika badania niezawodności
- rodzaje testów
 - testowanie funkcjonalne
 - testy нефункционалне (definicje)
 - testy zgodności z ... (standardami, prawem, ...)

2. Projektowanie przypadków testowych

- metody projektowania przypadków testowych
 - klasyczne przypadki testowe (pozytywne i negatywne)
 - testy zastępujące specyfikację (Behaviour Driven oraz Specification by example)
 - testy sterowane danymi (Data Driven)

3. Na co zwracać uwagę przy testowaniu

- kryteria akceptacji - co klient chciał a pewnie zapomniał o tym napisać
- ważne parametry: wydajnościowe, bezpieczeństwa

4. Testowanie w projektach internetowych

- architektura typowej aplikacji internetowej
 - architektura wielowarstwowa
 - SOA i pojęcia związane z integracją usług sieciowych
- piramida testów

- co testujemy
 - testy za pomocą interfejsu GUI
 - testy integracyjne usług
- rodzaje testów
 - testy manualne
 - testy automatyczne
- testowalność aplikacji internetowych - co zrobić by aplikacje testowało się łatwiej
 - procedury wytwarzania i wymagania dla dostawców
 - mechanizmy zwiększające testowalność aplikacji
- 5. Praca z aplikacją internetową
 - podstawy pracy przez SSH
 - podstawy pracy z Linuksem
 - podstawy diagnozowania problemów
 - czytanie logów
 - narzędzia developerskie w przeglądarce internetowej
 - Podstawy wykorzystania SQL
- 6. Testy automatyczne przy pomocy narzędzi automatyzujących
 - omówienie Selenium IDE jako narzędzia do prototypowania testów automatycznych
 - omówienie Robot Framework jako narzędzia do automatyzacji testów
 - omówienie HP QTP (Unified Functional Testing)
- 7. Testy usług SOAP i REST
 - omówienie SoapUI
 - omówienie Postman
- 8. Testowanie bezpieczeństwa
 - na czym polega testowanie bezpieczeństwa
 - kontekst ochrony danych osobowych i innych zasobów
 - standardy bezpieczeństwa: OWASP
 - Czy OWASP Top Ten jest dobrym zbiorem wymagań bezpieczeństwa?
 - metodyka
 - cykl bezpiecznego tworzenia oprogramowania
 - etapy prowadzenia testów penetracyjnych
 - wprowadzenie do modelowania zagrożeń
 - analiza statyczna kodu
 - analiza komponentów
 - weryfikacja konfiguracji - checklista
 - testy penetracyjne
 - narzędzia do testów penetracyjnych: OWASP ZAP, skipfish, Burp Suite
 - interpretacja i weryfikacja raportu z testów
 - narzędzia zwiększające produktywność testera
 - poza testowaniem: zarządzanie podatnościami, reagowanie na incydenty
- 9. Testowanie wydajnościowe
 - etapy prowadzenia testów wydajnościowych
 - analiza i planowanie
 - przygotowanie skryptów

- korelacja
- praca z różnymi technologiami aplikacji
- parametryzacja
- modularyzacja
- programowanie i skryptowanie testów
- narzędzia do testów wydajnościowych
 - JMeter
 - HP Load Runner

10. Testy jednostkowe (białoskrzynkowe) dla testerów

- rodzaje testów białoskrzynkowych
 - testy w technologii Java lub .NET
 - testy technologii Front (Javascript)
- narzędzia do badania jakości testów jednostkowy
 - pokrycie kodu testami
 - wprowadzanie błędów jako metoda badania jakości testów jednostkowych
- badanie jakości kodu źródłowego
 - SonarQube
 - Findbugs i PMD - efektywne wykorzystanie, predefiniowane zestawy reguł dla aplikacji różnych typów
- narzędzie do ciągłej integracji kodu (Continuous Integration)

11. Testowanie użyteczności

- Testowanie użyteczności
 - testy intuicyjności
 - zasady użyteczności
- Testy dostępności zgodnie z wymaganiami dla systemów z domeny publicznej i samorządowej
 - poziomy zgodności WCAG
 - testy dostępności manualne
 - testy dostępności przy pomocy narzędzi automatycznych

