

Opleiding : Software testautomatisering

Praktijkcursus - 4d - 28u00 - Ref. OTT

Prijs : 2430 € V.B.

★★★★☆ 4,1 / 5

Het automatiseren van softwareontwikkelingstesten is een kostbare maar winstgevende investering. Ontdek de organisatorische/financiële aspecten, best practices en tools voor het automatiseren van de verschillende soorten tests: unit tests, integratietests, functionele tests en performancetests.

Pedagogische doelstellingen

Aan het einde van de training is de deelnemer in staat om:

- ✓ Een testrepository gebruiken en koppelingen naar gerelateerde tools instellen om een testcampagne te automatiseren
- ✓ Een statische analyse van code en testdekking uitvoeren en deze integreren in een softwareproductieketen
- ✓ Het gebruik van dummy-, stub- of simulacrumobjecttechnieken en het opzetten van een continue integratieserver
- ✓ Functionele testautomatisering implementeren op fat clients en het web
- ✓ Prestatietests automatiseren, met de introductie van een test om de drempelwaarde voor denial of service te bepalen

Doelgroep

Ontwikkelaars, software kwaliteits- en testmanagers, testautomatiseringsingenieurs, projectmanagers.

Voorafgaande vereisten

Goede kennis van programmeren in Java.

Praktische modaliteiten

Praktisch werk

Theoretische sessies worden afgewisseld met praktisch werk.

Opleidingsprogramma

DEELNEMERS

Ontwikkelaars, software kwaliteits- en testmanagers, testautomatiseringsingenieurs, projectmanagers.

VOORAFGAANDE VEREISTEN

Goede kennis van programmeren in Java.

VAARDIGHEDEN VAN DE CURSUSLEIDER

De deskundigen die de cursus leiden zijn specialisten op het betreffende vakgebied. Zij werden geselecteerd door onze pedagogische teams zowel om hun vakkennis als hun pedagogische vaardigheden voor elke cursus die zij geven. Zij hebben minstens vijf tot tien jaar ervaring in hun vakgebied en oefenen of oefenden verantwoordelijke bedrijfsfuncties uit.

BEOORDELINGSMODALITEITEN

De cursusleider beoordeelt de pedagogische vooruitgang van de deelnemer gedurende de gehele cursus aan de hand van meerkeuzevragen, praktijksituaties, praktische opdrachten, ...
De deelnemer legt ook van tevoren en naderhand een test af ter bevestiging van de verworven kennis.

1 Inleiding, herinneringen aan het softwaretestproces

- De rol van testen in het ontwikkelingsproces.
- Tests: unit, functioneel, enz.
- De verschillende testmethoden.
- Testproces en teststrategie.
- Instrumenten en methoden die in verschillende stadia worden gebruikt.

Praktisch werk

Installatie van enkele gereedschappen.

2 Testbeheer automatiseren

- De dekking van vereisten door tests beheren. Begrip van dekking en granulariteit.
- Ontwikkelingsproces: organisatie van testsuites en casecreatie.
- Moet een test worden geautomatiseerd? Criteria om rekening mee te houden?
- Voorbereiden op automatisering.
- Samenstelling van de testpopulatie.
- Testontwikkeling en -verificatie (Beoordelen)
- Uitvoering, registratie van afwijkingen. Opstellen van incidentenrapport volgens IEEE.
- Anomaliebeheerders. Geautomatiseerde aanmaak van anomalieën.
- Resultaten van testuitvoering analyseren. Testconsolidatie.

Praktisch werk

Gebruik van een testrepository. Opzetten van bruggen naar gerelateerde tools (TestLink/Squash/Quality Center of andere, etc.).

3 Automatisering van eenheidstesten

- Organisatie en best practice voor unit testen.
- Automatiseringscriteria.
- Unit testen: Tests Gedreven Ontwikkeling.
- Het meten van codedekking: dekking van structurele tests, dekking van instructies en takken.
- Statische codeanalyse: toolgebaseerde analyse van niet-uitvoerende broncode (coderingsregels): Checkstyle, Cobertura.
- Automatisering met een configuratiebestand.
- Dynamische codeanalyse: dekking van instructies, vertakkingen, predicaten, enz.
- Automatisering met een dekkingsanalysetool.
- Organisatie van unit tests, peer programming, peer testing.
- Frameworks gebruiken: testscripts beheren, testgegevens beheren, resultaten ophalen.

Praktisch werk

Implementatie van eenheidstests met Junit/TestNG. Integratie in een softwareproductieketen (Ant/Maven, enz.).

PEDAGOGISCHE EN TECHNISCHE MIDDELEN

- De gebruikte pedagogische middelen en cursusmethoden zijn voornamelijk: audiovisuele hulpmiddelen, documentatie en cursusmateriaal, praktische oefeningen en correcties van de oefeningen voor praktijkstages, casestudies of reële voorbeelden voor de seminars.
- Na afloop van de stages of seminars verstrekt ORSYS de deelnemers een evaluatievragenlijst over de cursus die vervolgens door onze pedagogische teams wordt geanalyseerd.
- Na afloop van de cursus wordt een presentielijst per halve dag verstrekt, evenals een verklaring van de afronding van de cursus indien de stagiair alle sessies heeft bijgewoond.

TOEGANGSMODALITEITEN EN TERMIJNEN

De inschrijving dient 24 uur voor aanvang van de cursus plaatsgevonden te hebben.

TOEGANKELIJKHEID VOOR MINDERVALIDEN

Is voor u speciale toegankelijkheid vereist? Neem contact op met mevr. FOSSE, contactpersoon voor mindervaliden, via het adres psh-accueil@ORSYS.fr om uw verzoek en de haalbaarheid daarvan zo goed mogelijk te bestuderen.

4 Integratietesten automatiseren

- Specifieke integratiestrategieën: big-bang, "as we go", incrementeel, etc.
- Bottom-up versus top-down integratie. Gemengde integratie.
- Simulacrum-objecten: stoppers om de aangeroepen functies te simuleren, mocking om een object te vervangen. Frameworks.
- Continue integratie: het opzetten van een configuratiemanager, build builder, beheertools.
- Focus op een softwareconfiguratiebeheerder.
- Automatische storingsmelding.
- Automatische en cyclische uitvoering van softwaretests.
- Focus op een fabrikant.
- Richt u op een continue integratieserver.

Praktisch werk

Gebruik van mock objecten. Het opzetten van een continue integratie server (Jenkins, Hudson of Continuum).

5 Functionele testautomatisering

- Definitie van functioneel testen, non-regressie.
- De test simuleert gebruikersacties op basis van gebruikersinterfaces (GUI's).
- Bevindingen over functionele testautomatisering.
- Testautomatisering via de GUI, via programmeerinterfaces (API).
- Gereedschapsketen, testrobots, scripting (openbare API's).
- Beheer van testveroudering.

Praktisch werk

Het automatiseren van functionele tests op fat clients en op het web. Gebruik van robots en een publieke softwarelaag (API). Implementatie van een automatiseringstool (Katalon Studio met Selenium/HP QuickTest Pro).

6 Automatisering van systeemtests

- De verschillende soorten systeemtests: prestatie, belasting, stress.
- Bevindingen over prestatietesten.
- Prestatietests, definitie van indicatoren.
- Scripts voor prestatietests. Toolimplementatie en -analyse.
- Belastingtests, stresstests: presentatie van belangrijke indicatoren en tools.
- De testdatabase beheren. Contextbeheer.

Praktisch werk

Een belastingstest opzetten.

7 Samenvatting

- Testautomatisering: directe en indirecte kosten.
- Kwantitatieve voordelen van automatisering: volume, modulariteit, enz.
- Kwalitatieve voordelen van automatisering: bespaarde werkprocedures, geen fouten, vertrouwen, enz.
- Het meten van rendement op investering.

KLAS OPAFSTAND
2026 : 2 juni, 15 sep., 8 dec.

PARIS LA DÉFENSE
2026 : 2 juni, 15 sep., 8 dec.