GEMINI SOLUTIONS

Fundamentele testării software
Instrumente de management

Alina Pacurar

Quality assurance vs. Software Testing

GEMINI SOLUTIONS

- **Software Testing:**

Testarea software este un proces planificat care este utilizat pentru a identifica corectitudinea, integralitatea, securitatea si calitatea software-ului.

- Testarea se face in general pentru a demonstra ca software-ul face ceea ce spune ca face, precum si ca software-ul nu face ceea ce nu ar trebui sa faca.
- Scopul testarii/testerului este de a localiza defecte si de de a se asigura ca defectele sunt fixate.

Quality assurance vs. Software Testing

GEMINI SOLUTIONS

- **Quality Assurance (QA):**

- este un proces planificat si sistematic de a evalua calitatea unui produs, in scopul dezvoltarii unui produs de calitate. Scopul este de a oferi o asigurare că produsul indeplineste asteptarile de calitate clientilor. Se ocupa de asigurarea calitatii si prevenirea bug-urilor care apar într-un produs în curs de dezvoltare.
- notiunea de QA este asociata cu activitati cum ar fi masurarea calitatii procesului utilizat pentru a dezvolta un produs, imbunatatirea proceselor si prevenirea defectelor.
- Se compune din procedurile de audit si de raportare legate de dezvoltarea si testarea software.

Software Testing Life Cycle (STLC)

GEMINI SOLUTIONS

STLC defineste etapele / fazele prezente în testarea software-ului. Cu toate acestea, STLC variaza practic in functie de urmatoarele:

1. Capriciile Managementului
2. Software Development Life Cycle

In general, STLC cuprinde urmatoarele faze:

Modele de testare

GEMINI SOLUTIONS

Ciclul de dezvoltare/viata al produsului software se desfasoara conform unor serii/iteratii de etape bine definite. Se pot distinge urmatoarele modele:

- Modelul cascada
- Modelul V
- Modelul spirala
- Modelul iterativ si incremental
- Modelul agil

Acestea sunt intr-o continua evolutie si schimbare si drept urmare ciclurile de dezvoltare/viata ale produselor software tind sa dezvolte particularitati specifice.

Modele de testare – Modelul agile

Tine cont de **realitatea obiectiva** (Clientii nu pot defini de la început foarte clar si irevocabil cerintele)

- lucreaza impreuna cu clientii
- in acest moment este cel mai bun model pe care il avem în industriile non-critice si care permite o dezvoltare rapida a software-ului. Poate fi aplicat în aproape orice proiect în care o schimbare survenita pe parcurs sau un bug descoperit mai tarziu nu implica pierderi cu consecinte mari sau fatale

Modele de testare – Modelul agile

Metodologia scrum in modelul agile.

Test case, Test scenario, Test Plan

GEMINI SOLUTIONS

- In industria software, “scenariile” de test se numesc **test cases**.

Un test case se defineste printr-un set de intrări de testare si un set de iesiri care sunt conforme cu asteptarile.

- **Scenariile de test** reprezinta cerintele de testare grupate în functie de functionalitatea unui modul. Astfel un test scenario contine in general mai multe test cases.
- Un **test plan /** plan de testare este un document care descrie domeniul de aplicare, abordarea, obiectivele, resursele, și efortul necesar pentru testare software. Identifică elementele care urmează să fie testate, elemente a nu fi testat, cine va face testarea, tipul de testare aplicat, criterii de admis / respins ale produsului, programul de testare.

Ce este un Bug?

GEMINI SOLUTIONS

Bug - O problema, eroare, defect, posibilitate nedocumentata, greseala intr-un sistem software care împiedica sistemul sa functioneze asa cum este de asteptat.

Nu toate defectele software sunt cauzate de greseli în cod. O sursa raspandita de defecte costisitoare sunt lacunele si neclaritatile la nivel de cerinte, de exemplu, cerintele "ascunse" sau incomplete pot sa rezulte într-un set de erori introduse inca in faza de proiectare de catre designerul programului.

Test sessions & reports

GEMINI SOLUTIONS

- **Test session** - Interval de timp in care se executa o suita de teste pe o versiune a produsului software; pe parcursul unei sesiuni de testare se pot crea test cases noi, verifica defecte, raporta noi probleme/imbunatatiri.

- **Test session reports**

- Zona testata si Note detaliate cu privire la tipul de testare efectuat.
- O lista cu toate defectele gasite si intrebari deschise.
- Toate fișierele utilizate sau create pentru a sprijini testarea.
- Mentiuni in ceea ce priveste timpul petrecut pentru executarea testelor vs. timpul petrecut pentru a investiga imbunatatiri ale produslui.
- Grafic cu timpul petrecut pentru: Testarea - crearea si executarea testelor; Investigare / raportare defecte; Configurare sesiune de test sau alte activitati non-testare; Durata sesiunii de test.

Metrici in testarea software

GEMINI SOLUTIONS

S.No.	Testing Metric	Data retrieved during test case development & execution
1	No. of Requirements	5
2	Avg. No. of Test cases written per Requirement	20
3	Total no. of Test cases written for all requirements	100
4	Total no. of Test cases Executed	65
5	No. of Test cases Passed	30
6	No. of Test cases Failed	26
7	No. of Test cases Blocked	9
8	No. of Test cases un executed	35
9	Total No. of Defects identified	30
10	Critical Defects count	6
11	High Defects Count	10
12	Medium Defects Count	6
13	Low Defects Count	8

Defect/Bug Management

- Defect Lifecycle – se defineste prin etapele pe care le parcurge defectul din momentul in care a fost gasit pana cand a fost inchis (rejected/fixed).

- Starile defectelor:

Defecte/Issues

Costuri

Cost Of Defects
Cost US $
100
80
60
40
20
0
2
8
12
25
80
90
Requirement
Design
Code
Test
Prod
Maintenance
Phase

Tipuri de defecte/issues

- Varietatea este definita in functie de cerintele proiectului si de specificul acestuia
- Cele mai des intalnite tipuri sunt: bug, improvement, new feature, task, sub-task, epic.

Defect/issue management tools - Exemple

GEMINI SOLUTIONS

- JIRA - https://www.atlassian.com/software/jira
- Bugzilla - https://www.bugzilla.org;
- Mantis;
- Pivotal tracker;
- Redmine;

Test case management tool

GEMINI SOLUTIONS

De ce avem nevoie de Test Case management tool?

- ne ajuta sa avem date despre calitatea produsului in timp real pe parcursul proiectului;
- sa fim organizati;
- sa putem refolosi testele;
- sa stim in ce directie mergem cu calitatea produsului din timp astfel incat eventuale decizii de corectare sa fie luate la momentul corect.

Test case management tool - Caracteristici

GEMINI SOLUTIONS

- Gestionarea printr-o interfata web a test case-urilor si a sesiunilor de testare
- Versionarea si prioritizarea testelor
- Importarea/exportarea testelor
- Asocierea testelor cu defecte, cerinte si specificatii, si generarea automata a matricii de trasablitate a cerintelor
- Planificarea de campanii de testare si definirea atributelor importante in aceste campanii: responsabil, selectie teste care urmeaza sa fie executate din baza de date, pe ce platforme se testeaza etc
- Baza de date pentru rezultatele campaniilor de testare

Test case management tool - Caracteristici

GEMINI SOLUTIONS

- Suport si integrare cu instrumente de testare automata. De exemplu pentru testarea de performanta se poate programa o suita de teste automate care sa inceapa la o anumita ora si sa intoarca rezultatele direct in baza de date pentru rezultate.
- Definire de metrici de testare si includerea lor in rapoartele de testare
- Integrarea bidirectionala cu un sistem de bug tracking astfel incat atunci cand un test case este failed sa se poate deschide un bug direct din tool-ul de test management si actualiza in sistemul de bug tracking
- Integrarea cu serverul de email pentru trimiterea automata a rapoartelor si a notificarilor

Test case management tool

GEMINI SOLUTIONS

- Testlink
- Zephyr by Atlassian (add-on to JIRA) https://www.youtube.com/watch?v=oKi__ojyCUA
- QA Complete
- Qtest by QASymphony
- Pivotal Tracker

Testarea manuala - pro & cons

GEMINI SOLUTIONS

PRO

- Nu este necesar niciun tool de automation pentru a testa manual un produs software
- Este cea mai primitiva metoda de testare care ajuta la gasirea bug-urilor
- Orice aplicatie se testeaza prima data manual si abia apoi se trece la testarea automata.
- Testarea manuala este necesara in a verifica fezabilitatea testelor automate
- Testarea manuala este o activitate in care testerul are nevoie de rabdare, creativitate & open mind.
- Testarea manuala se face din prisma utilizatorului final

Testarea manuala - pro & cons

GEMINI SOLUTIONS

CONS

- Nu toate scenariile de test pot fi executate manual
- Load si performance testing nu pot fi rulate manual
- Nu se potriveste proiectelor mari cu deadline strans.

Testarea automata - pro & cons

GEMINI SOLUTIONS

PRO

- Timp scurt de rulare a unui numar mare de teste
- Se evita eroarea umana care apare in procesul de testare manuala

CONS

- Testarea automata necesita investitii substantiala in tool-uri si skills
- Necesita timp indelungat pentru scrierea testelor
- Nu poate inlocui abilitatile intelectuale ale omului in evaluarea UI si UX a unei aplicatii
- Nu poate fi folosita in exploratory testing
- Necesita cunostinte basic de programare

Testarea manuala vs Testarea automata

Cost vs. Timp

Cost Comparison - Manual Versus Automated tests

Automatizarea testarii

GEMINI SOLUTIONS

Bune practici

- Documentarea codului.
- Identificarea metodelor refolosibile si scrierea lor intr-un fisier separat.
- Follow the language-specific coding conventions.
- Mentinerea datelor de test intr-un fisier separat.
- Rularea testelor la intervale prestabilite de timp.

Automatizarea testarii

GEMINI SOLUTIONS

Caracteristicile aplicatiilor din prisma testarii automata

Proiectul software supus testării trebuie sa se afle într-o stare relativ stabila, cu alte cuvinte sa se afle mai degraba într-o faza de dezvoltare de noi functionalitati, sau chiar in faza de intretinere.

Initierea testarii automate in fazele initiale de dezvoltare, inainte ca proiectul software sa fie disponibil unor utilizatori finali, este mai degraba o abordare riscanta, avand sanse mari de anulare si rescriere a *scripturilor* de testare sau chiar de anulare totala a solutiilor de testare automata in care s-au investit timp si bani.

Automatizarea testarii - tools

Cum aleg cel mai bun tool de automatizare pentru aplicatia mea?

Selenium, RobotFramework

TestComplete, Soap UI

Built **in house** tools

Product	Selenium	Katalon Studio	Unified Functional Testing	TestComplete	SoapUI
Available since	2004	2015	1998	1999	2005
Application Under Test	Web apps	Web (UI & API), Mobile apps	Web (UI &API), Mobile, Desktop, Packaged apps	Web (UI &API), Mobile, Desktop apps	API/Web service
Pricing	Free	Free	$$$$	$$$	$$
Supported Platforms	Windows Linux OS X	Windows Linux OS X	Windows	Windows	Windows Linux OS X
Scripting languages	Java, C#, Perl, Python, JavaScript, Ruby, PHP	Java/Groovy	VBScript	JavaScript, Python, VBScript, JScript, Delphi, C++ and C#	Groovy
Programming skills	Advanced skills needed to integrate various tools	Not required. Recommended for advanced test scripts	Not required. Recommended for advanced test scripts	Not required. Recommended for advanced test scripts	Not required. Recommended for advanced test scripts
Ease of Installation and Use	Require advanced skills to install and use	Easy to setup and use	Complex in installation. Need training to properly use the tool	Easy to setup. Need training to properly use the tool	Easy to setup and use

Introducere in Framework-ul Selenium

GEMINI SOLUTIONS

Framework? = guideline (not mandatory rules)

Selenium este un Framework portabil de testare al aplicatiilor web, oferind posibilitatea de a scrie testele automate intr-o multitudine de limbaje (C#, Groovy, Java, Perl, PHP, Python, Ruby si Scala)

- Selenium a aparut in 2004;
- 2007 - Selenium Remote Control (RC)
- 2008 - Selenium Grid
- 2009 - Selenium WebDriver (Selenium 2.0)

Are suport pentru o mare parte din browserele cele mai folosite (Chrome, Firefox, IE, Safari)

Introducere in Framework-ul Selenium

GEMINI SOLUTIONS

De ce Selenium?

1. Este free and open source
2. Are un numar mare de utilizatori si o comunitate mare de Help
3. Are cross Browser compatibility (Firefox, Chrome, Internet Explorer, Safari etc.)
4. Platform compatibility (Windows, Mac OS, Linux etc.)
5. Suporta multiple limbaje de programare (Java, C#, Ruby, Python, Pearl etc.)
6. Este intr-o continua dezvoltare

Introducere in Framework-ul Selenium

GEMINI SOLUTIONS

Componente:

- **Selenium IDE** - record and playback tool. Este distribuit ca un plugin pentru Firefox si Chrome.
- **Selenium Remote Control (RC)** – este un server care permite utilizatorilor sa scrie teste in limbajul de programare dorit si rularea lor pe o arie mare de browsere.
- **Selenium WebDriver** - are diverse avantaje fata de RC printre care este capacitatea de a comunica direct cu browserul. Foloseste propria capacitate de a automatiza.
- **Selenium Grid** - este folosit pentru a distribui concurent executia testelor pe mai multe platforme si sisteme.

Introducere in Framework-ul Selenium

GEMINI SOLUTIONS

Limitari:

- Selenium poate fi folosit doar in testarea aplicatiilor web-based
- Aplicatiile mobile nu pot fi testate folosind Selenium
- Captcha si Barcode readers nu pot fi testate folosind Selenium
- Rapoartele pot fi generate folosind third-party tools (TestNG sau JUnit).
- Utilizatorul trebuie sa aiba cunostinte basic de programare.

Introducere in Framework-ul Selenium

GEMINI SOLUTIONS

Structura framework-ului

- Un folder “src” (source) care contine test scripts.
- Un folder ”lib” (library) care contine toate librariile si metodele.
- Un folder “class” care contine toate class files (in-case of java).
- Un folder “log” care contine toate fisierele cu log-urile.
- Un fisier/folder care contine toate web element Ids.
- Un fisier care contine URL, environment si login information. (de ce sa le mentinem intr-un fisier separat? URL, Login, si passwords sunt campuri folosite foarte des si se pot modifica la fel de des. In cazul in care le bagam in codul nostru ar deveni destul de greu de facut update de fiecare data atat timp cat ele nu sunt folosite doar intr-un singur fisier.)

Arhitectura Selenium WebDriver

GEMINI SOLUTIONS

1. Selenium Client Library - suporta mai multe librarii cum ar fi Java, Ruby, Python… pentru a suporta mai multe limbaje de programare.
2. JSON (JavaScript Object Notation) Wire Protocol over HTTP - este folosit pentru a transfera date intre client si server in cadrul aplicatiilor web. Comunicarea se face intre servere HTTP folosind REST API. Fiecare browser driver are propriul server HTTP.
3. Browser Drivers - fiecare browser are propriul driver. Driverul comunica cu browserul fara a fi nevoit sa cunoasca logica de functionare a browserului. Cand driverul primeste o comanda, atunci acea comanda este executata pe browserul specific iar raspunsul se returneaza sub forma de HTTP response.
4. Browsers - Chrome, Firefox, IE, Safari...

Selenium WebDriver - Locators

GEMINI SOLUTIONS

Selenium foloseste ceea ce se numeste “locators” pentru a identifica elementele in pagina cu care trebuie sa interactionezi. Cei mai folositi locatori sunt:

1. ID
2. Class Name
3. Name
4. LinkText
5. Tag Name
6. XPath
7. CSS

Selenium WebDriver - Locators

GEMINI SOLUTIONS

Exista diverse modalitati de a gasi/compune si verifica locators printre elementele HTML ale unei pagini web:

- Firebug/ firepath/ cropath
- Developer tools
- Regular expressions

Selenium scaneaza elementele din pagina incepand cu stanga sus. Astfel daca exista elemente care au aceeasi valoare, el o sa-l ia pe primul identificat si le ignora pe celelalte din pagina. Trebuie sa ne asiguram ca in identificarea elementelor folosim atributele care sunt unice.

That’s all folks!

GEMINI SOLUTIONS

Feedback & Learnings & Questions

Knowledge is the Most Beautiful of Awards