

Test Automation

Índex

- Introducció
- *Scripting*
- *Automated comparison*
- Altres

Índex

- **Introducció**
- *Scripting*
- *Automated comparison*
- Altres

Introducció

- Què és?
 - El procés **d'automatitzar** les proves de **regressió**.
 - Avantatges:
 - **Reduir l'esforç** necessari per fer test
 - **Augmentar el nombre de tests** a executar en un temps limitat (pot estalviar fins un 80% del temps de test manual)
- Quins aspectes que defineixen la qualitat d'un *test case* pot millorar la automatització?
 - **Efectivitat** en la detecció de defectes
 - **Exemplaritat** (testejar més d'una cosa a la vegada)
 - **Economia** (esforços i recursos per executar-lo)
 - **Mantenibilitat** i expansió (quan es modifica el SW)

Nota: Efectivitat i exemplaritat només depenen de les habilitats del tester

Introducció

- Avantatges:
 - Executar tests de **regressió**
 - **Més** test en **menys** temps (o més sovint)
 - Genera més **seguretat** en el SW
 - Executar test **impossibles** d'executar **manualment**:
 - Proves on-line amb 200 connexions, dades d'una base de dades, atributs interns d'una GUI, events generats en una GUI, etc ...
 - Millorar la **precisió**:
 - Executant tasques repetitives i avorrides (entrar inputs, textos, etc) que són propenses a error.
 - **Consistència** en/i la **repetibilitat** del test
 - Executar el test **exactament** igual cada cop
 - **Reutilització** del test

Introducció

- Desavantatges
 - **Esperances no realistes** (el *automated testing* no és la panacea!)
 - Porta fàcilment a fer **mals tests**:
 - Es **preferible millorar** l'efectivitat del **test** que no la de l'automatització (*"Automating chaos just gives faster chaos"*)
 - Creure que l'automatització descobrirà nous defectes
 - El **defectes** es solen trobar en la **primera execució** d'un test. El *regression testing* només assegura que cap efecte secundari ens tornarà a produir un **nou error** (i per tant serà un error **menys probable**).
 - **Manteniment no sempre és fàcil** (canviar el **SW** pot significar **canviar el test**)
- *Missatge: L'automatització no sempre és bona!*

Introducció

- Es pot automatitzar el **disseny** d'un test case?
 - La **imaginació** i habilitat d'un dissenyador **humà** no es poden substituir, pero ...
 - Les **eines** d'automatització dels **test inputs** poden ajudar. N'hi ha de tres **tipus**:
 - Code-based
 - Paràmetres per executar un path, ...
 - Interface-based
 - En una GUI es poden visitar tots els objectes sensibles (botons, caselles), visitar tots els links d'una pagina web, ... (tècnica anomenada *monkey test*)
 - Specification-based
 - A partir de llenguatge natural (o quasi) poden generar inputs i els outputs corresponents: valors límit, particions equivalents, ...

Que suggereix el darrer punt? Es poden automatitzar les proves de caixa negra!!!

Introducció

- Limitacions del automated testing:
 - **NO susbtitueix el manual testing !!!!!!!**
 - Hi ha **testing** que **MAI** es podrà realitzar de forma **automàtica**
 - **Només** s'hauria d'**automatitzar** el manual testing que **s'executa molt**
 - NO s'hauria d'automatitzar:
 - Test **que s'executa rarament**
 - SW **volàtil** (que canvia molt, sobretot la seva interfície)
 - SW fàcilment verificable per un humà i difícilment per una màquina. **Exemple?:**
 - Imatge, esquema de colors, estètica, sò ...
 - **Interacció amb hardware** (sobretot quan no podem controlar perfectament l'estat del hardware)
 - Si el SW és **inestable** → **Manual testing!**
 - Manual testing (85% errors) / Automated testing (15%) (en aquest cas hem de considerar el JUnit com *manual testing*)
 - **No millora l'efectivitat** del test (no troba més ni millors errors)
 - Les **eines NO són intel·ligents**:
 - Un usuari pot veure que el comportament del SW és incorrecte tot i que els outputs siguin els esperats

Índex

- Introducció
- ***Scripting***
- *Automated comparison*
- Altres

Scripting

- L'automatització és fàcil amb *scripting*, però és avorrida i pesada
- Avantatges de *l'scripting* detallat:
 - Testers executen sempre el **mateix test**
 - **Errors** de SW més **fàcils** de **reproduir**
 - **Qualsevol** pot **executar** el **test** (programadors, testers, ...)
 - Fàcil **d'automatitzar**
- Desavantatges:
 - Difícil de **crear**
 - Molt **texte/instruccions redundant**
 - És *verbose*
 - **Avorrit d'executar** (el tester no fa gran cosa excepte mirar)
 - **Overhead** de feina
 - **Esforç** per crear el script
 - La **informació** està documentada **dos cops** (a la **descripció** del test i al **script**)
 - ➔ Informació hauria de ser legible per humans i per l'eina de test.

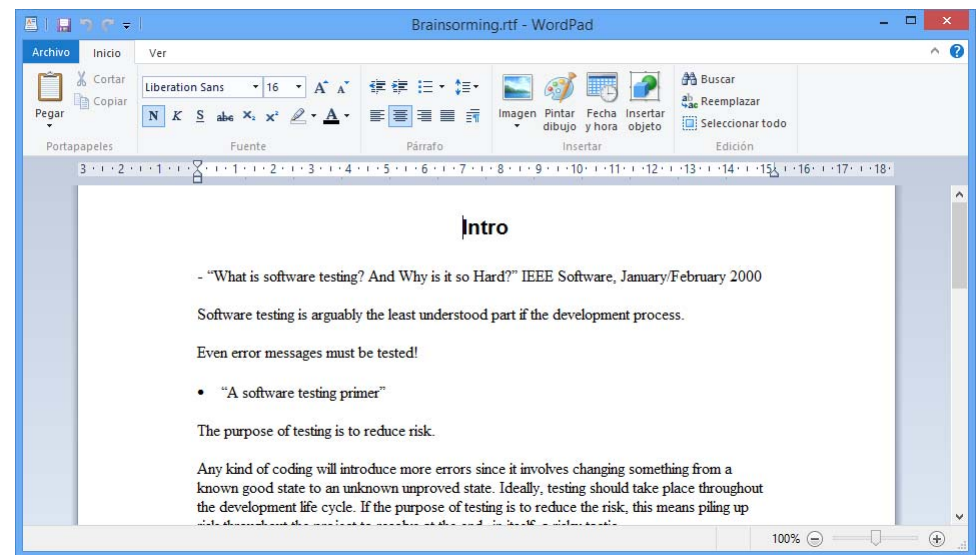
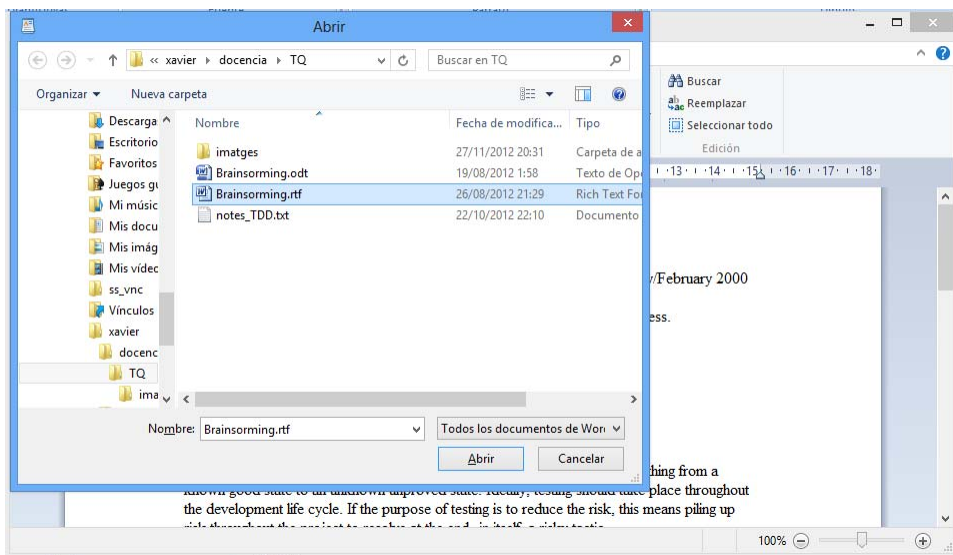
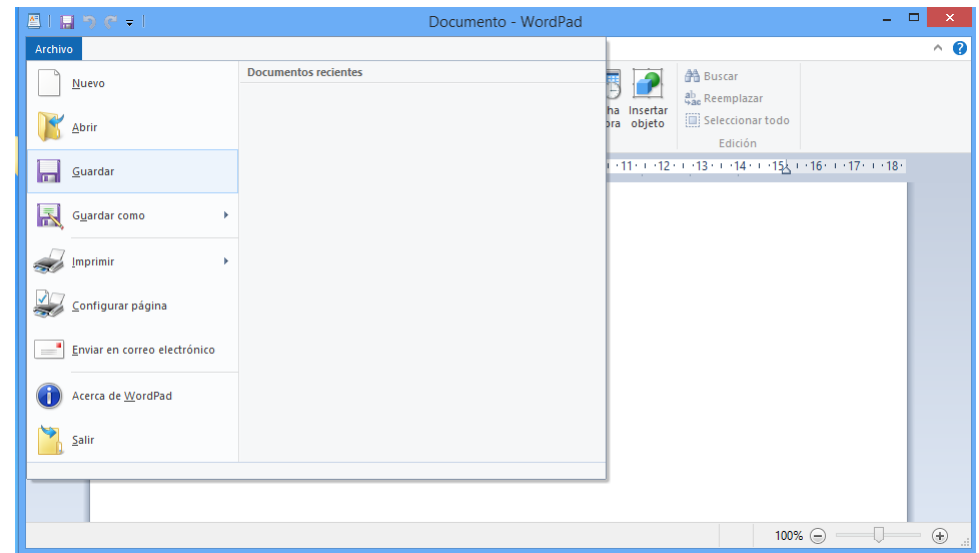
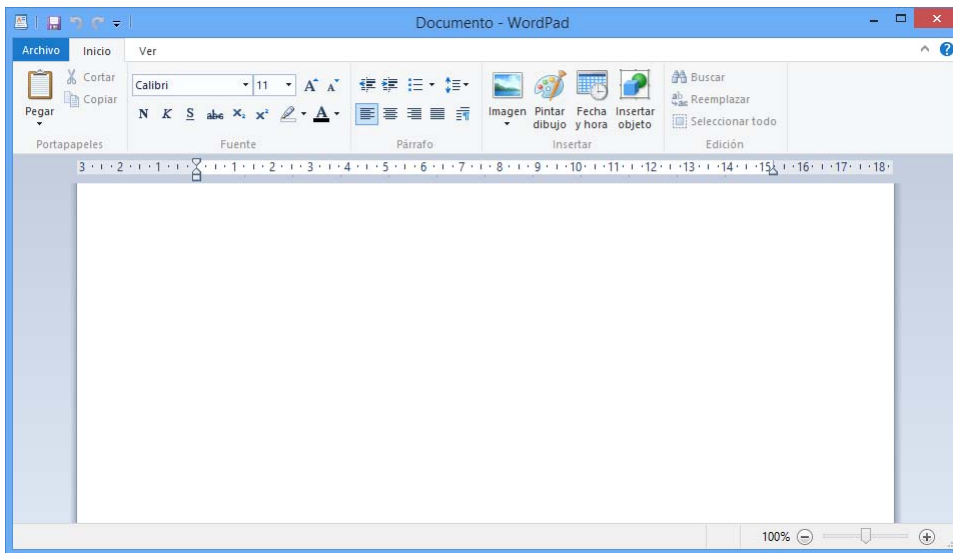
Tipus

- Manual scripts (JUnit)
 - Escrits per usuaris **humans**
 - Solen ser **legibles**
 - Serveixen com a **documentació**
- Automated scripts
 - Creats per un **SW**
 - **Difícils** de llegir
 - **Si** són creats perquè **s'ha fet *recording***, tenen una **utilitat limitada**

Recording

- Procés de **gravació** de les **accions** realitzades per l'usuari
- Desavantatges:
 - Solen crear scripts **ilegibles**
 - **Només** són utils si s'han de **re-executar**
 - Molt centrats en **aspectes específics** fàcilment **volatils**:
 - Posició en la pantalla, bitmaps, cadenes de caràcters, ordre en una llista ...
 - Si el **SW canvia** mínimament el script pot fàcilment **deixar de funcionar**.

Recording



Recording

- NO automatizar testing fent recording!!!
 - **Recording NO fa test**, ja que no es fa **cap comprovació** de la sortida (**outcomes**)!!!
 - Si no es comproba el outcome NO es fa automated test, es fa **automated input**.
- Per tant, si volem fer ***automated testing***, hem de fer ***automated comparison***

Outcome Comparison

- Tipus:
 - **Dynamic** comparison
 - Realitzat **DURANT** l'execució del test
 - Com es compara ho decideix el tester (imatges, llistes, element dins una llista, posició en la pantalla, elements en la pantalla ...).
 - **Post-execution** comparison
 - Realitzat **DESPRES** de l'execució del test
- *Automated comparison* **NO** és una tasca trivial
 - Hi ha **moltes coses** a comparar
 - Triar **dynamic** o **post-execution** depèn del **què** es compara i **quan** es compara.
 - Els scripts es tornen complexes
 - Més informació a comparar → Més susceptibles als canvis (del SW i del *test case*)

Scripting techniques

- Tècniques
 - Linear scripts
 - Structured scripting
 - Shared scripts
 - Data-driven scripts

Linear scripts

- És la sortida típica d'un software de recording:
 - Tots els keystrokes, characters, functions keys, posicions del mouse, posició dels clicks, ...
- Pot incloure comparisons
- Avantatges:
 - Des del **primer moment** es pot fer *automating*
 - L'usuari **no cal** que sigui **programador** (d'scripts)
 - Es molt útil per fer **demos!!!**
- Quan utilitzar-lo
 - Per quasi qualsevol **tasca repetitiva**
 - Per automatitzar el **set-up**
 - **Clean-up** del test
 - **Omplir** un **fitxer** o una **base de dades**
 - **Repetir** un **input**
 - Fer **conversions**

Linear scripts

- Desavantatges
 - Crear-lo és un procés **intensiu i laboriós** (típicament de 2 a 10 cops el que costa executar un test manual)
 - Els **inputs** i les **comparisons** estan **hard-coded** en el **script**
 - **No és reutilitzable**
 - **Vulnerable** als **canvis** en el SW
 - Difícil de modificar
 - És **impràctic a llarg termini** i per fer **molts test**.

Structured scripting

- Similar a la programació estructurada:
 - Conté instruccions i **estructures de control**:
 - Seqüències: Instruccions d'execució
 - Seleccions: Decisions (“if”)
 - Interacions: Buclis

Shared script

- Script utilitzat per més d'un test
- Avantatges:
 - **Menys esforç** per **implementar** un test
 - **Manteniment inferior** al d'un **linear** script
 - Elimina repeticions
 - Es **bò** per sistemes **petits** (PC) o per provar una **part petita** d'un SW
- Desavantatges:
 - Molts scripts a utilitzar
 - Manteniment encara relativament alt

Data-driven script

- Les dades es guarden en un fitxer a part
- L'estructura de control es troba en el script
- Avantatges:
 - El mateix script pot servir per molts tests (canviant les dades)

Índex

- Introducció
- *Scripting*
- ***Automated comparison***
- Altres

Automated comparison

- *Verification* és el procés de chequejar els *outcomes* a través d'una *comparison*.
- Tipus: Dynamic i post-execution
- Per què comparar automàticament?
 - Una comparació és probablement una tasca molt automatitzable
 - Sense ***automated comparison*** no tenim ***automated testing***
- *Simple comparison*
 - Comparació EXACTA del outcome i el resultat esperat
- *Complex comparison*
 - Comparació amb diferències esperades: output amb ordre diferent, diferències dins un rang (comparació de floats, ...)
- Quanta informació s'ha de comparar?
 - Molta: Serem molt sensibles a petits (irrellevants?) canvis
 - Poca: Més robust, però menys acurat

Automated comparison

- Filtres:
 - És un pas de **edició** o **transformació**: Eliminació de dades, extracció, conversió, ordenació ...
 - S'apliquen tan als **outcomes** com als resultats **esperats**
- Avantatges:
 - Reutilitzables
 - **Criteri** de comparació **específic**
 - **Outcomes complexes** poden ser analitzats per **eines simples**

Índex

- Introducció
- *Scripting*
- *Automated comparison*
- **Altres**

- Quins tests hem d'automatitzar primer?
 - **Repetitius** i *human error-prone*
 - Testing **funcional** (Què fa el SW)
 - ➔ Test Driven Development
 - Els més **importants**
 - Els que s'executen **més cops**
 - Els més **fàcils d'automatitzar**
- *Do not automate too much too soon (en oposició a “test a little, code a little”)*
- Utilitzar *test-selectors*:
 - Permeten executar diferents tipus de test (*short tests, tests that failed the first time, performance test, load tests, ...*)
- *Designing SW for (automated) testability*
 - De manera **similar** al **TDD**, podem desenvolupar el codi tenint en ment la **manera** en què **farem el test** ➔ En podriem dir ATDD