Test Development Driven (TDD) i Behaviour Development Driven (BDD)

Consultor: Roman Roset roman@uoc.edu

1. Objectius d'aquesta unitat

En aquesta unitat l'objectiu és conèixer una tècnica que potser ja estem utilitzant d'alguna forma però que no ens havien dit el nom: el TDD. No només serà aquest l'objetiu, sinó que veurem com podem fer TDD amb JavaScript i quines eines tenim. I ja està? Doncs potser que anem un pas més enllà i aprofitem les lliçons apreses d'aquells que fa anys que hi treballen, i veurem dos coses més: la relació que hi ha entre les Històries d'Usuari i el TDD i que en les històries d'usuari és solen definir les proves d'acceptació amb termes més pròxims al BDD (un product owner no es fixa en allò que no pot funcionar, per exemple, comprovar que el camp està en blanc, sinó que es fixa en el comportament que acceptarem).

Teniu ganes de començar, oi? Doncs, anem per feina!

1. TDD from starting from user stories – a top-down style

Us presento un fantàstic post per a que comprengueu per què definim les proves d'acceptació en les User Stories, i com aquestes han de dirigir la programació que fem:

- http://exceedhl.wordpress.com/2007/08/02/tdd/

Fixeu-vos especialment en aquests punts:

- Què és un test d'acceptació, i com a partir del test d'acceptació fem els altres test.
- Quan agafem una HU si analitzem amb detall el test d'acceptació, comprovarem si realment es invesT.
- No us fixeu encara amb les eines, les veurem més endavant.
- Especial atenció als pros i contres.
- Per cert, l'exemple usa BDD... noteu la diferència entre Test i Comportament? És una qüestió de semàntica bàsicament... ja ho veurem.

2. Un bon tutorial de TDD en javascript que hi ha en la Xarxa

A continuació seguirem un tutorial de molta qualitat que hi ha en Internet sobre l'us de TDD:

1. Introducció a TDD: http://www.etnassoft.com/2011/02/10/tdd-en-javascript-1/
2. Test Frameworks: http://www.etnassoft.com/2011/02/13/tdd-en-javascript-2/

Fixar-se en el següent:

- La estructura teòrica de un TDD
- Què és la refactorització
- El framework Qunit, és el que jo us proposo. Fixeu-vos que depèn de jQuery, però també podem usar zepto (fixeu-vos en els test de backbone).

3. Seguim amb el mateix tutorial, però ara ens parla de BDD

Seguirem amb el mateix tutorial, i ara ens parla de BDD. El que m'ha agradat és el següent comentari:

Por lo general, cuando hablamos de TDD (incluyendo esta serie de artículos), no nos referimos a la versión más purista de la metodología sino a su evolución inteligente elaborada a partir de sus críticos

Anem a veure el tutorial:

3. http://www.etnassoft.com/2011/02/15/tdd-en-javascriptiii-parte/

Fixar-se en:

- El BDD és el mateix però usant un vocabulari diferent: *en lugar de hacer un test unitario, realizamos un **análisis del contexto**. En lugar de escribir métodos que comiencen con la palabra 'test', los comenzamos con un **'debería' (should)**. En este sentido, los frameworks basados en BDD ofrecen una sintaxis ligeramente modificada a los que vimos en TDD incorporando este vocabulario más enfocado a una especificación que a una verificación.*
- El framework que ha triat per usar.

4. Testejar aplicacions MVC amb BDD

Finalment uns tutorials a mode d'exemple sobre com testejar aplicacions fetes amb MVC i programant amb la tècnica de BDD. Ostres! Quina casualitat! Els tenim amb backbone i Jazmine.

- http://tinnedfruit.com/2011/03/03/testing-backbone-apps-with-jasmine-sinon.html
- http://tinnedfruit.com/2011/03/25/testing-backbone-apps-with-jasmine-sinon-2.html
- http://tinnedfruit.com/2011/04/26/testing-backbone-apps-with-jasmine-sinon-3.html

Bé, però la veritat és que no hi ha un clar favorit sobre com fer les proves. L'important es fer-ho amb TDD o BDD i amb algun framework on puguem fer les proves tan automàtiques com sigui possible.

5. Exemple pràctica amb Sencha:

No deixeu de llegir aquest post:

http://www.sencha.com/blog/automating-unit-tests/

Utilitza els següents frameworks per a fer BDD:

- PhantomJS: El phantom serveis per emular l'us d'un navegador. D'aquesta form, donada una pàgina podem dir que executi events d'elements d'aquesta pàgina, anant a buscar els elements a través del seus Ids en css.
- JSLint: S'utilitza per (millor mirar http://www.jslint.com) reportar qualitat en el codi JS.
- PhantomLint: és una extensió de phantomJS per a que cada fitxer del projecte el faci passar pel JSLint.
- Jasmine: Un tester BDD!!!!! Per a Javascript!!. .. i per a què necessitem el PhantomJS? Doncs per que el Jasmine pugui fer les proves emulant que és un usuari.

Espero que amb aquest minitutorial, qui vulgui fer BDD no tingui cap problema!!!

Optimitzant una aplicació de mòbil amb tecnologia Web (WPO)

Consultor: Roman Roset roman@uoc.edu

1. Objectius d'aquesta unitat

En aquesta unitat es veurà alguns exemples de tècniques per optimitzar les aplicacions webs per a mòbils.

La pregunta clau és: cal optimitzar una aplicació web per a mòbil que sį'instal.la tota en local com una aplicació d'escriptori? La resposta en general és que no cal preocupar-se tant de l'optimització comparat amb la mateixa aplicació si es serveix des de web.

Totes les optimitzacions que es descriuen, en general, van dirigides a aplicacions web que es serveixen des de servidor. I em de ser nosaltres que analitzem si una possible tècnica d'optimització ens pot anar bé o no per a la nostra aplicació. Per exemple, està comprovat que el temps de resposta de l'event 'touch' és molt més ràpid que l'event 'click'. Ara bé, l'optimització d'emportar-se les pàgines fent pre-fetch i emmagatzemant-les en local, en general, no ens caldrà en una aplicació d'escriptori, donat que quan s'instal·la, ho fa amb totes les pàgines. Però i si la aplicació carrega dinàmicament d'un servidor les dades o inclús fragments de pàgines? Doncs llavors, sí que val la pena aquesta optimització.

El tema d'optimitzar no ha de ser gratuït. Cada cop que optimitzem el que fem en general és augmentar la complexitat de codi i reduir la mantenibilitat de l'aplicació. Per tant, cada

optimització ha d'estar controlada i justificada. Per tant, si usem tècniques d'optimització s'espera una justificació analitzant l'abans i el després empíricament.

Així mateix, l'optimització de codi per aplicacions web, i en especial per a mòbils és actualment un tema per gent experta, donada la complexitat d'elements que hi ha en joc. Per tant, us recomano que guardeu aquesta unitat per aplicar un cop ja heu superat la fase d'aprenentatge, doncs serà en aquell moment que s'entendran la majoria de problemes que apareixen i les tècniques que es poden aplicar.

1. Per entrar en matèria

Per entrar en matèria us deixo un vincle cap l'últim congrés mundial important sobre velocitat en aplicacions web mòbils:

- http://velocityconf.com/velocity2012/

Per a la majoria de sessions hi ha penjada la presentació. El que es interessant d'aquest congrés, és que es poden veure els diferents temes a tractar sobre l'estat d'art actual:

1. Automation strategies
2. Optimizing mobile performance
3. JavaScript speedups
4. Metrics and monitoring

Aquí teniu un recull de les presentacions:

- http://velocityconf.com/velocity2011/public/schedule/proceedings

Si llegiu alguna presentació tingueu en compte que aquest congrés no es acadèmic, sinó d'empresa. Les presentacions parles d'optimització, però també intenten vendre el producte de la seva empresa.

2. Tipus d'optimitzacions

Mireu aquesta pàgina i tingueu-la a mà per quan vulgueu començar a optimitzar les vostres aplicacions:

- http://developer.yahoo.com/performance/rules.html

Llegiu-la amb atenció. En ella trobareu les principals tècniques per reduir l'ample de banda d'una aplicació web. Aquestes tècniques s'anomenen WPO (Web performance Optimization) i són aquelles que, en general, no ens són tan importants si la aplicació s'instal·la en local. Podem distingir dos tipus d'optimitzacions:

- WPO en el servidor web.
    - Gzip components
    - Reduir les redireccions

    - Etc..
- WPO abans de ficar en producció l'aplicació.
    - Minimitzar l'script de javascript
    - Minimitzar el css
    - Optimitzar les imatges
    - Sprites en imatges
    - Estils de css en el top
    - Scripts de javascript en el bottom o a demanda.

La pàgina web que em vist, parla de tècniques de WPO en general, però hi ha un altre tipus d'optimització per a aplicacions web de telefonia mòbil, aquelles que depenen del hardware del dispositiu, i que ens hi podem aprofitar si usem l'html5 i el css3 correcte. Serien les que jo anomeno:

- Optimitzacions dinàmiques d'aplicacions web per a mòbils.

En aquest cas ja no es tracta de produir una versió stàtica del codi html i css minimitzada, amb imatges que ocupin poc, i intentant reduir al màxim l'ample de banda, sinó que, es tracta d'usar el hardware de l'aparell, i els motors gràfics incorporats dintre del navegador per obtenir millors rendiments.

Les següents presentacions parlen d'optimització del rendiment per a mòbils. En aquestes podrem veure els tres tipus d'optimització:

- WPO en servidor
- WPO en client
- Optimitzacions per usar el hardware del dispositiu

Les trobarem aquí:

- http://www.mobilexweb.com/blog/mobile-web-html5-performance-optimization
- http://static.googleusercontent.com/external_content/untrusted_dlcp/www.google.com/es//events/io/2011/static/presofiles/use_page_speed_to_optimize_your_web_site_for_mobile.pdf
- http://www.html5rocks.com/en/tutorials/speed/html5/
- http://www.html5rocks.com/en/mobile/optimization-and-performance.html

No les comento per a que obteniu les vostres pròpies conclusions. Les dues últimes, són, des del meu punt de vista les més interessants, donat que expliquen les tècniques per optimitzar el rendiment amb el hardware.

3. Eines per a automatitzar la optimització estàtica (WPO en el costat del client)

Hi ha diferents eines per separat que optimitzen diferents aspectes del procés. Així per exemple, per minimitzar el css o fer sprites podem fer servir compass, o less. Per minimitzar el javascript podem usar eines com http://developer.yahoo.com/yui/compressor/ . Però hi ha alguna eina que pugui automatitzar tot el procés? Si. La que us proposo que proveu es la més agnòstica, es tracta de HTML5BoilerPlate (HBP) i una versió reduïda per a mòbiles: MobileBoilerPlate (MBP).

Que són els *boilerplate (tasques repetitives)*? No us ha passat que sempre que comenceu una aplicació copieu una espècie d'estructura base per començar on hi ha tots els frameworks que us fan falta i fitxers com d'exemple? Doncs això es un boilerplate.

Una de les coses que podem aprofitar de HMP (són projectes molt amplis i no entraré en detall) que també porta el MBP (MBP és un HMP retallat) és un el script ant que porta per automatitzar les tasques d'optimització.

Podeu llegir aquest document que ho explica:

- http://html5boilerplate.com/docs/Build-script/

No us proposo que ho feu servir de moment, sinó que tingueu aquesta referència per a més endavant per si podeu aprofitar aquest script per a automatitzar l'optimització estàtica de codi. No obstant, us recomano que abans d'usar aquest tipus d'eina prioritzeu la part d'optimització de codi a partir de l'acceleració amb el hardware.

4. Consideracions finals

- No us preocupeu d'optimitzar fins que no veieu la necessitat.
- Prioritzeu les optimitzacions sobre el hardware.
- No us entreteniu en optimitzar a no ser que el rendiment sigui nefast.
- Si optimitzeu feu un anàlisi per què quedi constància en la memòria i sigui un valor afegit.