

Projecte Sméagol

amb canvis petits es poden fer grans coses

(Càtedra de Programari Lliure)

Universitat Politècnica de Catalunya

La UPC

És gran i heterogènia.

El personal TIC de la UPC
està dispers...

geogràficament...

functionalment...

laboralment...

i tecnològicament.

Cal satisfer moltes demandes semblants però no idèntiques.

Habitualment hi ha pocs
recursos disponibles.

Cal una solució!

No Silver Bullet

Cap d'aquestes polítiques
des de dalt és una solució...

el ja s'ho faran,

el despotisme il·lustrat,

la subcontractació dels serveis.

Cap d'aquestes polítiques
des de baix és una solució...

els veïns que s'espavilin,

la postura del Kalimero,

la fugida endavant.

Un petit pas

Incentivar una cultura de col·laboració...

des d'una òptica confederal.

Col·laborem perquè
ens surt a compte
i hi sortim guanyant.

El projecte Sméagol és un
petit pas en aquesta direcció.

Objectius

Avançar en la col·laboració...

aprenent de les comunitats
de programari lliure.

Aprenhere a utilitzar certes
eines col·laboratives.

Aprofitar algunes
idees tecnològiques.

Acabar tenint programari
compartit entre grups.

Adquirir experiència i
transmetre coneixement.

Aconseguir que el treball
sigui divertit i enriquidor.

Estratègia

Buscar un nou projecte...

prou simple (i prou complex),

d'interès per als participants,

que pugui arrapinyar-se
alguns recursos.

Sméagol...

el gestor de reserves
de recursos.

Comencem definint...

una arquitectura,

una primera fita...

(simple i abastable)...

i ens llencem a la piscina!

Metodologies àgils

Les grans especificacions
acostumen a ser irrealis...

i molt cares.

L'ús és el millor
especificador.

És més rentable fer i desfer...

(si es dissenya sota
aquest precepte).

Cal tenir mitjans que ho
permetin...

architectures de programmation
adequades,

mètodes de treball escaients,

eines que ajudin,

equips de treball motivats.

Cicles molts curts
d'especificació, disseny,
implementació i alliberament.

TDD

Test Driven Development

Es tracta d'una de les eines
del desenvolupament àgil...

es determina què
ha de fer un mòdul,

s'escriuen un conjunt de
tests que comprovaran que
ho fa correctament,

s'implementa el mòdul
lluitant contra els tests.

Existeixen eines ad-hoc
per usar aquesta tècnica.

Milloren els problemes
de regressió.

Els tests serveixen com a
exemples i documenten.

Nous errors impliquen
nous tests.

REST

Un model per entendre la semàntica del web...

i poder-la aplicar a les
aplicacions web.

Si seguim el model,
les coses van bé...

l'usuari se sent a casa,

els clients funcionen bé,

el sistema és
escalable i robust,

l'aplicació és part del món.

Conceptes REST

Client/Servidor

Recurs=entitat identificable

URL=identificador d'entitat

1 recurs, N representations

Operacions predefinides: GET, PUT, POST, DELETE

Principis REST

L'estat de l'aplicació i la
funcionalitat està als
recursos.

El significat dels recursos
no varia amb el temps.

Cada recurs té una URL.

Es transmet la representació
d'un recurs.

Un recurs té diverses
representacions negociables.

Els mètodes HTTP ofereixen
la semàntica CRUD.

SOA

Service Oriented Architecture

És un terme molt de moda...

però en realitat és...

un *buzzword*.

SOAP vs. REST

SOAP és un protocol de crides
remotes sobre transport HTTP.

SOAP no estandarditza...

les operations disponibles,

ni el format dels continguts,

ni els identificadors
dels objectes.

SOAP no garanteix
la interoperabilitat.

SOAP repeteix els esquemes
dels RPC, DCOM, CORBA,
Java RMI...

però no treu el suc del
transport HTTP i del web.

Arquitectura

Tres components functionals...

servidor,

client,

agregador.

Per què?

El servidor

És el *back-end*
(p.e. 1 per unitat).

S'hi dialoga mitjançant
una API REST.

1 functionalment,
N implementations
possibles...

però totes es comporten
igual.

Gestiona i controla dades a
tots els nivells...

per això l'anomenem també
l'sméagol.

Pot tenir polítiques pròpies
(permisos, atributs, etc.).

El client

És el *front-end*
(p.e. N per unitat).

Dialoga amb el servidor
mitjançant l'API REST.

1 functionalment,
N implementations
possibles...

però amb el mateix
comportament.

No aporta d'altra
funcionalitat que no
depengui del servidor.

Creació d'embolcalls per a
clients en diferents
llenguatges de programació.

L'agregador

Pot fer d'intermediari entre
clients i servidors
(p.e. 1 per campus).

Permet partager recursos
entre servidors.

Permet als clients tenir una
visió global.

Té informació dels diferents
servidors agregats.

Treballa com un proxy HTTP
i un directori alhora.

Associar-s'hi és voluntari.

Per què?

L'objectiu d'aquesta
arquitectura és...

disposar d'un sistema
modular i flexible,

permettre d'intégrer hi une
existence,

separar presentació i
aplicació.

Les raons d'exploació...

la possibilitat de triar la
tecnologia que implementa
l'arquitectura,

la facilitat per implantar el
sistema sense que calgui
imposar-lo,

un sistema federat en què
cadascú és responsable de la
seva part.

Les raisons techniques...

ens obliguen a tenir una
interfície clara,

els components ben separats
i substituïbles,

i una divisió més fàcil de la
feina a realitzar.

Quant a les raons
estratègiques...

es tracta d'un sistema
flexible,

el model federat s'adequa al
context de la UPC,

potencia la col·laboració
entre diferents unitats,

representa nous reptes
tècnics que dinamitzen la
feina i motiven el personal.

Preguntas?

Gràcies!

<http://creativecommons.org/licenses/by-sa/3.0/es/deed.ca>

