

Desenvolupament d'aplicacions professionals

Qualitat de codi e integració continua

Màster en Enginyeria Informàtica

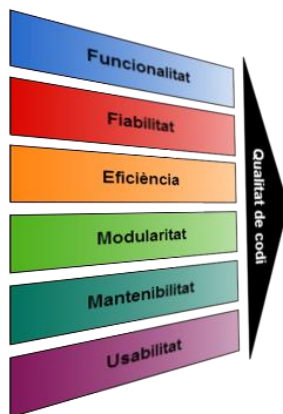
Juan Carlos Martín Capitán

09/01/2015

Introducció a la qualitat de codi i la integració continua

Avui dia, el desenvolupament d'aplicacions software s'ha convertit en grans projectes d'enginyeria, que requereixen gran quantitat de temps i personal de diferents àrees. A més, l'impacte que tenen errors com aquests poden arribar als milions d'euros.

Degut a això, constantment s'han intentat crear i millorar els sistemes i processos pel desenvolupament software, alguns amb més èxit que altres. Però es van adonar que el desenvolupament de software no es podia equiparar a la construcció d'edificis, degut entre d'altres, per la indeterminació dels requisits i els continus canvis d'aquests.

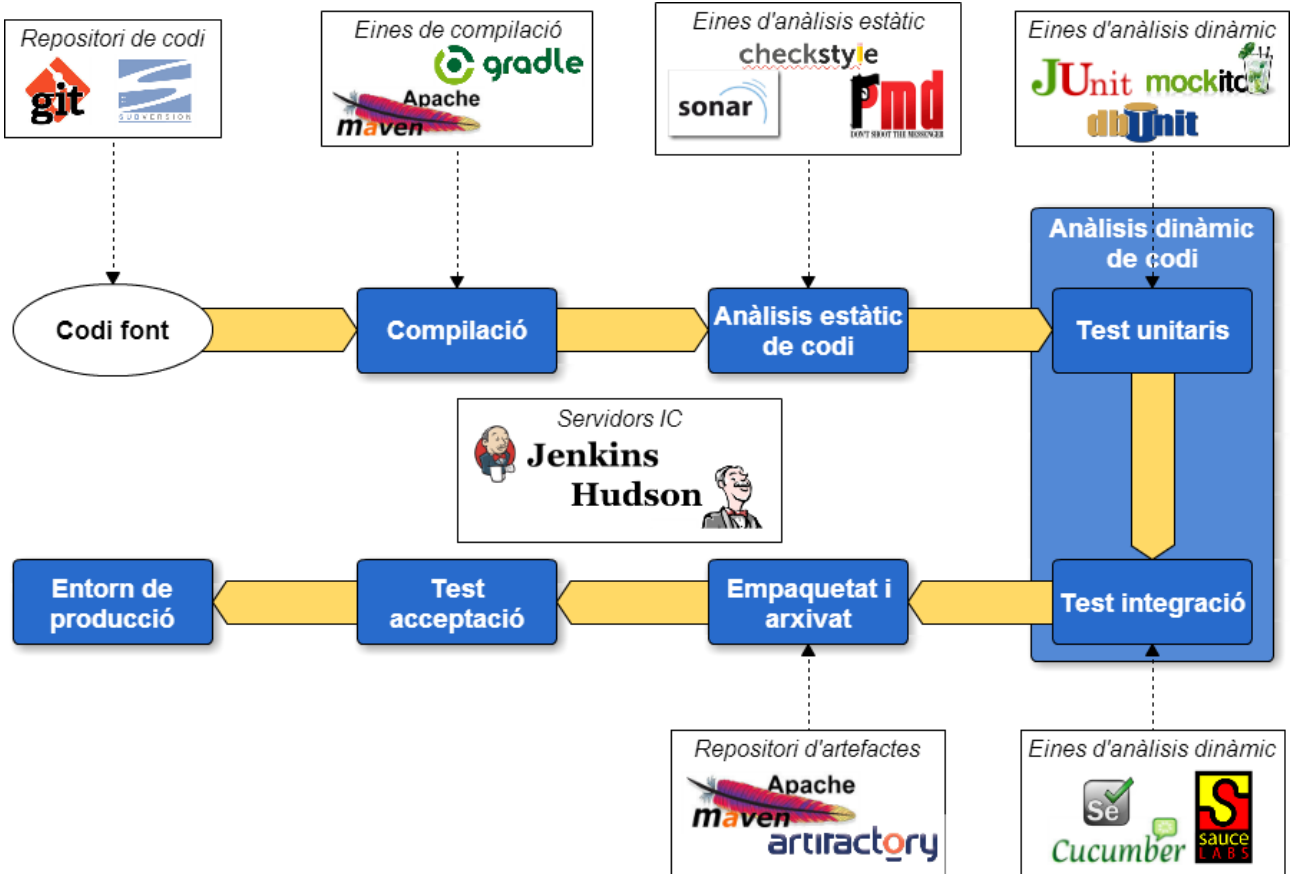


A partir d'aquí van sorgir altres conceptes, per millorar aquets sistemes. Entre ells va sorgir el concepte de **Qualitat de codi** per poder avaluar si un programa estava ben escrit o no. Un concepte que es sustenta sobre 6 pilars: un codi que funcioni (funcionalitat), que no falli contínuament (fiabilitat), que consumeixi pocs recursos i poc temps (eficiència), que pugui dividir-se en petites parts (modularitat), que sigui fàcil de mantenir (mantenibilitat) i que l'usuari sigui capaç d'utilitzar (usabilitat).

També va sorgir el concepte de **Integració continua** per automatitzar tasques que fins llavors necessitaven temps i la intervenció humana, amb els error que això pot comportar. Una de les tasques que automatitzava era justament la comprovació de la qualitat de codi. Fent això automàticament es podia realitzar més freqüentment, podent detectar problemes el més aviat possible, i degut a que era més barat la detecció i correcció d'errors a les primeres fases, això també aportava un menor cost de desenvolupament.

Creant una solució d'integració continua

Un dels primers objectius era el desplegament d'un sistema d'integració continua que ens permetés simplificar el desenvolupament en un futur projecte. Però primer era necessari conèixer quines fases són necessàries en un projecte software, des de tenir el codi font, fins al desplegament en un entorn de producció, on els usuaris finals utilitzen l'aplicació. Això s'anomena el cicle d'integració continua:



Una vegada construït aquest esquema i mirant els diferents softwares que existien al mercat, es va poder dissenyar una solució centrada en un servidor d'integració continua **Jenkins** i una eina de qualitat de codi anomenat **Sonarqube**.

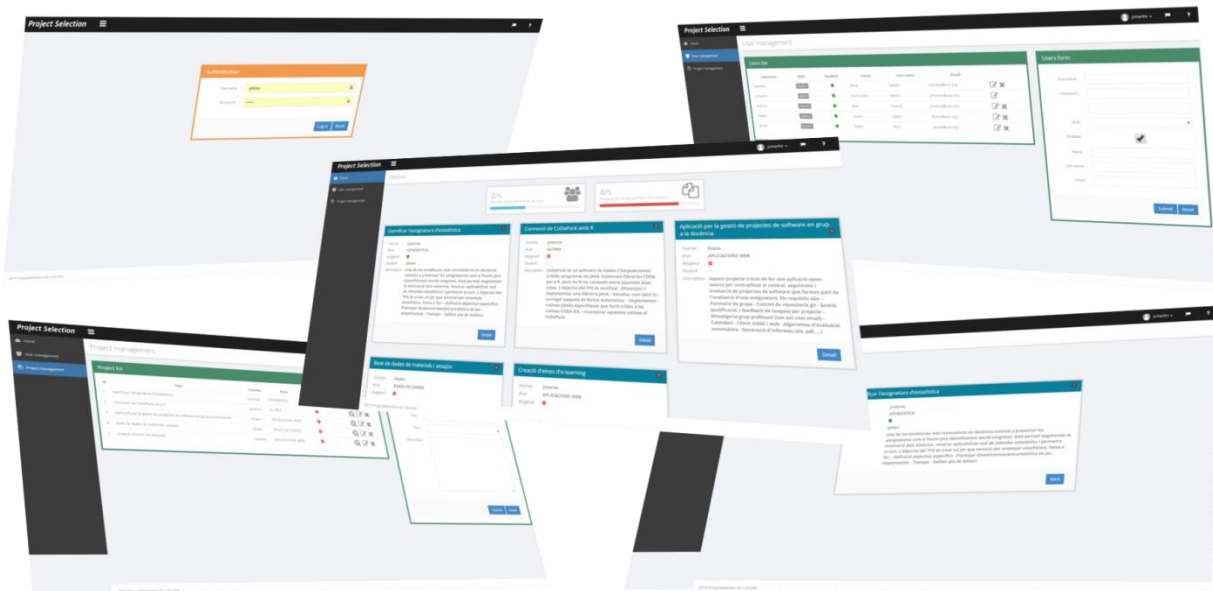


Posant a prova la nostra solució d'integració continua

Una vegada creada la nostra solució d'integració continua, era necessària posar-la a prova. Es va crear una senzilla aplicació web amb la qual es van automatitzar tots els aspectes necessaris per cobrir la part del cicle d'integració continua que ens interessava:

- Configuració d'un repositori de codi per mantenir el codi del programa guardat de forma segura.
- Increment de versió de forma automàtica, per saber en tot moment quina versió s'estava generant.
- Comprovació de la qualitat de codi i mostrar els resultats de forma entenedora amb Sonar.
- Desplegament de l'aplicació totalment automàtic.

El conjunt de l'aplicació era bastant simple, una pàgina de gestió de projectes de final de grau o màster, on els professors podien inserir projectes i els alumnes podien consultar els projectes que hi havia, i assignar-se un d'ells.



Conclusions

Aquest projecte m'ha ajudat a adquirir uns bons coneixements sobre un tema que portava temps interessant-me com es la integració continua. No només utilitzar una petita part, sinó se capaç de fer un complet desplegament, configuració i la utilització completa per fer el desenvolupament d'un projecte.

He pogut comprovar que la integració continua requereix un sistema complet, uns coneixements previs i una configuració prèvia del projecte amb certa complexitat. Després de tot això, ens proporciona una velocitat, fiabilitat i qualitat molt alts en comparació amb un desenvolupament ordinari, i si comparem el que suposa un primer esglau complex amb tot un gran projecte, acaba resultant en un gran benefici a mig-llarg termini pel desenvolupament de software.