

Code review és continuous integration toolok

Code review

Egy (nagy)vállalati környezet

- Nagy fejlesztőcsapat, sok fejlesztő
- Rengeteg commit
 - Subversion esetén központi szerver
 - Git esetén elosztottan van mindenkinél egy repó
 - Vagy levelezésben küldik a változtatásokat (l. Linux kernel)
 - Vagy központi szerverre, repóba
 - » Push és pull

Code review

A „Linux-kernel módszer” (részlet kernel levelező listából)

This patch is to the rtl8712_cmd.h that fixes up typo error in comment for word "true".

```
Signed-off-by: Punit Vara <punitvara@gmail.com>
```

```
---
```

```
drivers/staging/rtl8712/rtl871x_cmd.h | 2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)  
diff --git a/drivers/staging/rtl8712/rtl871x_cmd.h  
b/drivers/staging/rtl8712/rtl871x_cmd.h  
index 2514bfd..f204321 100644  
--- a/drivers/staging/rtl8712/rtl871x_cmd.h  
+++ b/drivers/staging/rtl8712/rtl871x_cmd.h  
@@ -436,7 +436,7 @@ struct getrfintfs_parm {  
    * mac[0] == 0  
    * ==> CMD mode, return H2C_SUCCESS.  
- * The following condition must be ture under CMD mode  
+ * The following condition must be true under CMD mode  
    * mac[1] == mac[4], mac[2] == mac[3], mac[0]=mac[5]= 0;
```

Code review

„Linux kernel módszer” folyamata (nagy vonalakban)

- Fejlesztő a helyi fában fejleszt, commitol
- Kész egy változtatása, megformázza, levélben elküldi a központi levelezőlistára
- A felelős fejlesztők **átnézik**, (elsőre többnyire leszólják :-)), majd ha átmegy az ellenőrzésen, beolvasztják a központi fába.

Code review

Központi szerver

- Mindenki lokálisan fejleszti a kódját, összegyűlik „elég” módosítás, felküldi a központi szerverre (*remote*)
 - `git push`
- A központi módosításokat – más módosításait – le tudja tölteni
 - `git pull` vagy `git fetch`, `git rebase`

Code review

Központi szerver

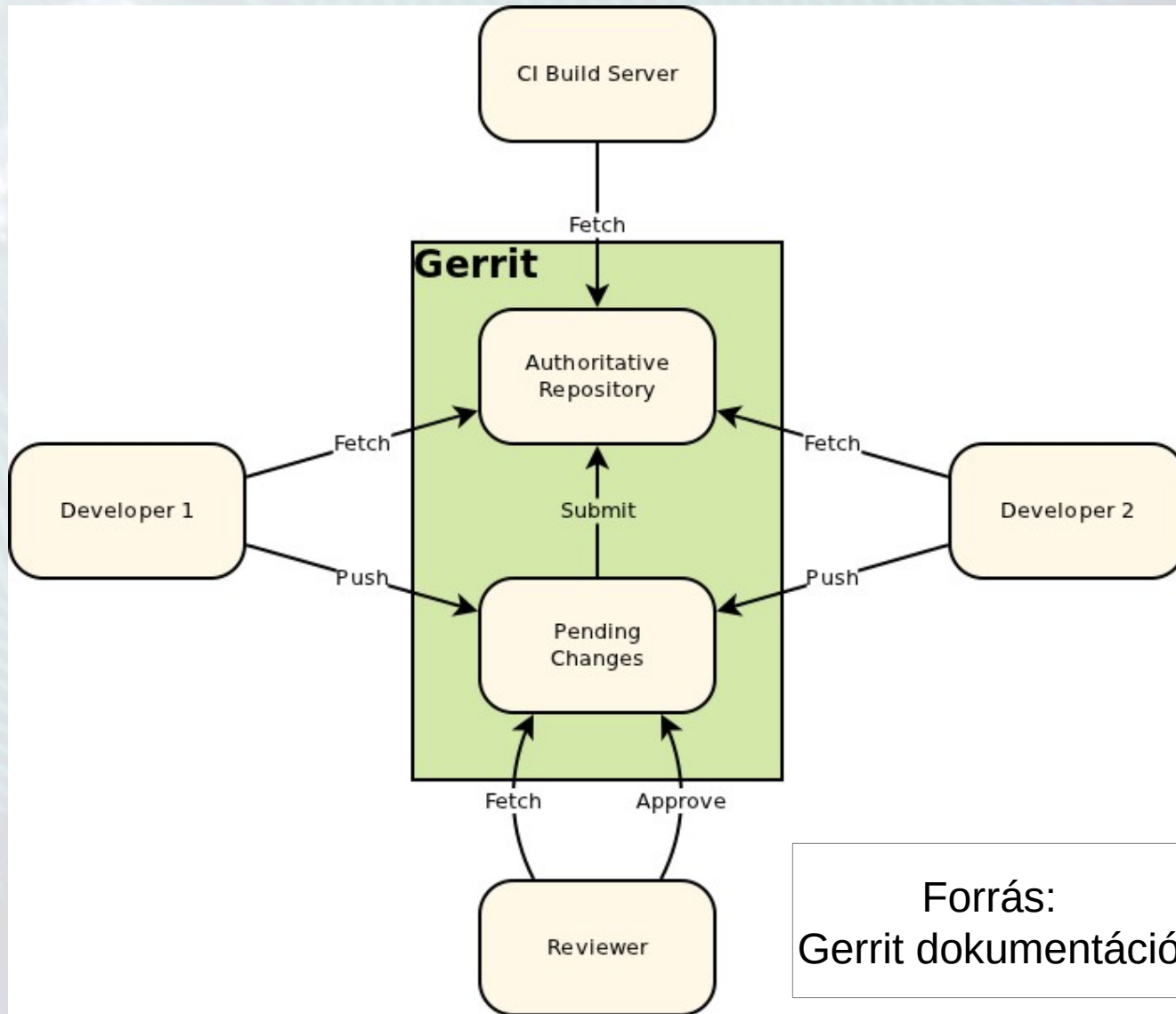
- Hátránya: hozzáféréssel bárki bármit átnézetlenül beküldhet
 - Funkcionalitás elromolhat
 - Coding style sértés
 - Legrosszabb esetben le sem fordul
- Itt is kéne valamilyen review módszer
 - Kódminőség – emberi átnézés
 - Automatizált tesztek
 - Fordítási tesztek

Code review

Gerrit

- Nagyvállalatban népszerű code review eszköz
- Google fejleszti, Androidnál alapvetően ezt használják
- Szerepei:
 - Kód áttekintés
 - Központi repó kezelése
 - Commit gyűjtőhely
 - Fejlesztői véleményezés
 - CI toolokkal kapcsolat

Gerrit



Forrás:
Gerrit dokumentáció

Code review

Gerrit

- Fejlesztő távoli repositoryként látja a gerrit-et, push-sal beküldi a változást
 - Ez bekerül az átmeneti területre, de a többiek nem tudják (közvetlenül) letölteni – nincs beolvasztva
 - A többi felhasználó az online felületen átnézi, szavazhat róla (jó minőségű és fordul)
 - +2 review-ban és +1 fordításban → rendben van, bemehet
 - -2 review-ban: vétó, ez semmiképpen sem mehet be
 - Egy változtatás akkor mehet be, ha mindkettő kategóriában a legmagasabb szavazatot (+2,+1) kapja

Code review

Példa

- Egy projekt
- Három felhasználó
 - bela, kebab: commit es review jogokkal
 - wachag: verify jogokkal – most még nem használjuk
- Projekt hasonlóan klónozzható, mint Github esetén
 - Csak a központi repó most a helyi Gerrit szerver lesz
- Git push helyett – mert push-hoz nincs joguk – csak review

Code review

Hiányosságok

- Van, amit a fejlesztő kipróbálás nélkül nem vesz észre
 - Lefordul-e?
 - Helyesen működik?
- **automatizált tesztek**

Continuous Integration

Jenkins

- CI tool
 - Job kezelés
 - Automatizált fordítás vagy teszt
 - » Időzítve, vagy repóba töltéskor
 - » Kézzel futtathatóan
- Különféle külső események (commit, vagy kézi indítás) hatására indítható:
 - Windows batch fájl
 - Linux shell script
 - Maven build
 - Vagy egyéb eszköz – a Jenkins pluginezhető

Jenkins

Példa

- Az előző példában megmutatott projekthez – melyet Gerrit segítségével kezelünk – automatikus build verify-t teszünk hozzá
 - Gerrit oldalról szükséges egy olyan felhasználó, aki olvasni tudja a repót és verify-olhat
 - Jenkins oldalról
 - Gerrit-támogatás konfigurálása – Gerrit trigger plugin
 - Build lépések konfigurálása