

AKO SME ZAVIEDLI PORIADOK VO VÝVOJI?

Obsah

- Úvod
- Čo nás naučili v škole
- Ako sme narazili
- Špecifiká vývoja vo VNET-e
 - a firmách, kde sa vyvíja pre interné potreby
- Oprášenie teórie, akčný plán a vyslobodenie
- Rekapitulácia

Vyštudoval som FIIT STU

□ Učili ma, že:

- sa používajú rôzne modely vývoja
- programovaniu predchádza analýza a návrh (špecifikácia, diagramy...)
- existujú unit-testy (a mnohé iné)
- treba mať jasný plán, verifikovať, validovať
- treba robiť code-review
- na dvojsemestrálny projekt v 7-člennom tíme treba mať 7 rôznych manažérov a 140 strán dokumentov :)

Zamestnal som sa vo VNETe

- Teóriu ovládam, už len trochu praxe
 - zistiť ako to robia skúsenejší
 - osvojiť si praktické postupy

- Škola mieni, firma mení
 - zistil som, ako veci nerobiť
 - „antivzory“ v praxi

Antivzor I - Testy

- Zbytočnosť
- Čas sú peniaze, úloh je veľa, tak aby to odsýpalo
- Poklikal som a beží to
- Čo sa môže na tomto kúsku kódu pokazit'?!
 - Je to iba interný projekt, neletíme do vesmíru.
 - Interpretované jazyky sú nebezpečné
 - nefunkčný kód nemusí spôsobiť problém, pokiaľ sa vetva nevykoná

Antivzor II - Dokumentácia

- Netreba, za to nás neplatia
- Kto by to vôbec čítal?
- Stačí poznačiť ako to spustiť, prípadne čo robiť, keď sa to zosype
- Pár komentárov nás predsalen nezabije, hlavne pri „dočasných“ riešeniach

Antivzor III - Barličky

- Jasně, přidám tam „ifko“:
 - veď je to len „dočasne“
 - nebudeme predsa strácat' čas systémovým riešením
- Veľmi sa s tým netrápte, hlavne jednoducho
- Beží to v screen-e, chalani, keby to náhodou spadlo, stačí znova spustiť

Antivzor IV - Vývoj vs. produkcia

- Zmením to rovno „na živej“, ved' sú to len tri riadky
 - aspoň to hneď „otestujem“
 - bude to skôr nasadené, ved' už na to čakajú
 - do GIT-u to dám hneď potom (samozrejme „zo živej“, nebudem to predsa písať ešte raz)

Antivzor IV - Vývoj vs. produkcia

```
-rw-r--r-- 1 lulo      lulo      29653 Jun 19  2015 ind.php
-rw-r--r-- 1 root      root      33739 Jul 19  2016 index-new.php
-rw-r--r-- 1 lulo      lulo      22098 Apr  1  2014 index.bak
-rw-r--r-- 1 www-data  www-data  33323 Apr 18 10:43 index.php
-rw-r--r-- 1 root      root      33735 Jun 30  2016 index.php.backup
-rw-r--r-- 1 root      root      28198 Jun 18  2015 index.php.bak
-rw-r--r-- 1 root      root     859596 Mar  7  2013 index2.html
-rw-r--r-- 1 root      root      26310 Feb  4  2015 index_.php
```

Antivzor V - Replikovateľné prostredie

- Inštalovateľné balíky/projekty sú práca navyše
- Zaznačené závislosti netreba, ešte by ich potom bolo treba aktualizovať
 - u mňa to predsa beží a na serveri som to vyladil tiež
 - nový kolega nech sa s tým potrápi, nebudeme ho predsa platiť za nič
- Hlavne nenasadzovať všetko rovnako, nech to máme pestrejšie

Všetko má svoje dôvody

- Neplačme nad rozliatym mliekom, obviňovaním sa to nevyrieši
- Nevieme, čo viedlo k daným rozhodnutiam

- Hanba nie je robiť niečo zle, ale uspokojiť sa s tým a nechcieť to zmeniť

Špecifické prostredie

- Nie sme „software house“
- Vyvíjame pre interné potreby, nie pre zákazníka
 - to zvädza ku kompromisom
- Vyvíjame podporné systémy, nie produkt
 - dnes už nie celkom pravda
 - zastrešujeme zároveň operatívne činnosti
- Firma rastie a s ňou aj potreby

Bojový plán

- Neudržateľnú situáciu bolo treba stabilizovať
 - v ideálnom prípade aj zvrátiť
 - hrozilo prepuknutie demotivácie
- Oprášiť a aplikovať teoretické poznatky
 - a doplniť o „rady starších“ a skúsenejších
- Nasadiť nové technológie a „nadbehnúť si“

Bojový plán - Git(Lab)

- Nie je to len úložisko na prenos a archiváciu
- Master nie je jediná vetva
- Merge môže ísť aj requestom
 - a schváliť mi ho môže aj kolega
- Issues tvoria praktický to-do list a čiastočnú dokumentáciu
- Continuous Integration
 - testy, deployment

Bojový plán - Code review

- Pomocou merge request-ov v GitLab-e
- Zároveň knowledge-sharing
- Dokážeme odhaliť chyby v neotestovanom kóde

- Automatické nástroje (Flake8 - Python linter)
 - prvotná kontrola, ešte pred commitom
 - zvýšenie kvality existujúceho kódu

Bojový plán - Replikovateľné prostredie

□ Inštalovateľné balíky

- setuptools, PBR
- setup.py, requirements.txt

□ Zjednotené nasadzovanie

- /opt
- virtualenv
- process manager a server - Circus, uWSGI/Gunicorn

Bojový plán - Testy

- Aspoň na nové projekty
- Maximalizovať pokrytie
- Napísať hneď, neskôr = nikdy
- Mať aspoň základné („lucky-path“) testy
 - aj pri deklaratívnych ORM modeloch aspoň inštanciovanie triedy
- Mocking
 - šetrí čas, znižuje zložitosť testovania
- TDD - zatiaľ nie striktne

Bojový plán - Dokumentácia

- reST - Python native, prípadne Markdown
- Aspoň README
 - popisom projektu, základným použitím/spustením
- Dokumentácia v kóde
 - aspoň pre API
- Napísať hneď, neskôr = nikdy

Bojový plán - Kontajnery (Docker)

- Stabilizácia legacy projektov
- Zdieľanie s kolegami
- Continuous Integration
 - Testovacie prostredie
 - Build process
- Nasadenie do cloudu

Bojový plán - Monitoring / logovanie

□ Sentry

- centralizované logovanie
- traceback vždy po ruke na analýzu
- možnosť pridať vlastné dáta (session)
- koniec posielania screenshotov
- konfigurovatelné notifikácie

Bojový plán - Open Source komunita

- Firemný profil na GitHub-e
- Propagácia
- Príspevok do komunity
- Motivácia robiť veci poctivo

Bojový plán - Manažment kvality

- Také naše malé „ISO“
- Identifikujeme nedostatky
- Kontinuálne hľadáme možnosti zlepšenia
- Učíme sa nové veci
- Nerobíme veci bezhlavo - diskutujeme

Zhrnutie

- Nie vždy sú akademické vedomosti nepoužiteľné
- Treba hlavne chcieť, samé sa veci nezmenia
- No pain, no gain

Otázky?

Čo..?

Kedy..?

Prečo..?

Kam..?

Kde..?

Ako..?

Odkiaľ..?

Koľko..?

Ďakujem za pozornosť :)