

QA/TEST/CI Engineer guide

Csak hogy értelmet nyerjen az eddig átadott információ, kicsit összeszedem a lényeget, hogy mi mit követ és miért.

1: Python – hogy tudjuk, mivel kell dolgoznunk. Eleinte elég a tudást felszednünk

2: ISTQB - Az eddig felszedett tudás minőségét segíti javítani

3: Git – Amikor az ember saját projektek írásába kezd, sokkal jobb, ha egy repo-ban tartja őket. A git tudás szinte mindenhol kötelező és sokkal szebb a CV is, ha a saját project repo szerepel benne. Láthatják az ember fejlődését is.

4: Clean Code: Ha minden ember úgy írná a Pythont, ahogy szerinte jó, akkor eléggé katikus, úgymond spaghetti kódok keletkeznének. Olvashatatlan és logika nincs benne.

<https://hu.wikipedia.org/wiki/Spagettik%C3%B3d>

<https://www.python.org/dev/peps/pep-0008/>

<https://testdriven.io/blog/clean-code-python/>

Jelen tudás birtokában (Python-ISTQB-GIT) nekiállhatunk a projekteknek. A következő linken van pár ötlet. Nyugi, az elején nehéz elkezdni, de ha már eltöltött valaki benne pár órát, biztosan boldogulni fog. Lehetnek az elején benne összezsapott részek, a végeredmény legyen jó.

<https://www.upgrad.com/blog/python-projects-ideas-topics-beginners/>

5: Jenkins – Ha mondjuk már írtunk 1 programot, vagy van egy VM akár Ubuntuval, tesztelhetjük is. Pl számológép esetében írhatunk inputra test esetet. Mi történik, ha nem adunk meg semmit? Mi történik, ha számot betűvel szorzunk? Mi történik, ha nullával? Stb stb. VM esetében csinálhatunk többet is, megnézhetjük az OS fajtáját, a vCPU és RAM mennyiséget. Vagy akár kódot is vizsgálhatunk. Ha PEP8 checkre elfailel a kód amit írtunk, pl túl hosszú, kimarad egy pont, nem jó a behúzás, akkor jelezze egy failed test-el. Kezdetekhez elég is.

6: ROBOT/Selenium – Tudjuk, hogy mind a 2 test framework. A Selenium és a ROBOT is képes Eclipse-ben működni, lehet arra törekedni, hogy ebben tanuljuk meg. De persze elég az is, ha a kurzusokban/videókban használtat tudja az ember, mert ugyan úgy működik. Bele lehet külön projektbe implementálni, hogy Python keywordök segítségével VM-en különböző teszteket hajtunk végre (Robotnál). Vagy Selenium-al csinálhatunk egy hibás regisztrációt valamelyik weboldalon, túl rövid jelszóval. Mondjuk ha már van Jenkins VM, az is 1 tökéletes felület hozzá, a netre sem kell kimenni. Ez közel olyan, mint amit valójában is végeznek automata tesztelés esetében.

7: Opcionális:

Regexp – Ezt ha indokolja a project, érdemes a pythonnal párhuzamosan venni. Vagy inkább közvetlen utána.

Networking – A VM teszteléséhez jól jöhet, ha tudjuk, miért az az IP, miért nem látjuk a VM-et, ha egy másik subneten van, vagy miért nem működik a TV okosító, ha egy netről megy, de másik subneten van. Miért 192.168.1.X az IP, bárhol vagyunk.

Linux – A fentebb használt VM-eknél javarészt kelleni fog. Nem kell rögtön iptables-ben írni egy tűzfalat, elég ha tudja használni. Értem ezalatt a mappák közötti lépegetés, megnyitás, új doksi létrehozása.