



Jussi Paalanen työskentelee johtavana konsulttina SYSOPENDIGIA:ssa.

Ajatuksia koodarin arjesta

Koodarin tehtävänä on tuottaa ajoympäristössä ajettavissa olevia sovelluksia, jotka täyttävät asiakkaan tarpeet. Asiakkaan tarpeet on kuvattu tavalla, joka ei ole suorituskelpoisessa muodossa, eikä myöskään automaattisesti muutettavissa sellaiseksi. Siksi tarvitaan tulkkina toimivaa ihmistä.

Tyypillinen ajoympäristö etenkin ns. enterprise-maailmassa on värikäs ja koostuu kaupallisista alustatuotteista, erilaisista sovelluskehityksistä ja kolmansien osapuolten kirjastoista. Ajoympäristö voi vaihdella projektiokohtaisesti ja pelkkä version vaihdos ajoympäristön osajärjestelmässä saattaa vaikuttaa merkittävästi siihen, miten tulee toimia. Koodarin työkalut siis vaihtuvat jatkuvasti; hän ei enää tulkkaa asiakkaan tarpeita aina yhdelle ja samalle symboliselle konekielelle.

Oppiminen onkin tärkeä osa työtä ja sitä tapahtuu sekä horisontaalisesti (uusien asioiden oppimista) että vertikaalisesti (syventävää asiakkoista oppimista). Oppiminen on iteratiivista ja inkrementaalista; tutuistakin asioista opitaan uutta projektien myötä.

Usein on mahdotonta löytää täsmälleen tietyn ympäristön läpikotaisin tuntevia ihmisiä ja heidän kauttaan valmiita ratkaisumalleja. Koodarin rooli onkin vahvasti sellainen, jota Keith Harrison-Broninski mm. kirjassaan "Human Interactions" nimitää "interaction worker" -termillä. Moniosaaja kerää ja jalostaa tietoa, tuottaa uutta ymmärrystä, ja neuvottelee muiden kanssa siihen perustuen. Koodarin täytyy osata tunnistaa riippuvuuksia ja niihin liittyviä kommunikointitarpeita. Organisaation on ymmärrettävä työn luonne ja luotava infrastruktuuri, joka huomioi tällaiset työn piirteet.

Yhtenä haasteista on löytää ratkaisu uuden opetteluun ja asiakkaalle tuottavan työn yhdistämiseksi. Onko koodarin organisaatiolla käytössään jokin systeemi, joka pysyy kehityksessä mukana, löytää toimivat ratkaisut ja organisoi riittävän tietämyksen projektien käyttöön? Vai onko se "ad hoc" -tyyppistä, projekti- ja henkilökohtaista

työtä? Kuinka paljon organisaatio luottaa määrittelelemättömään, työajan ulkopuolella tehtävään, ajankäytön mahdollisuuksien ja harrastuneisuuden sanelemaan opiskeluun?

Koodarin on oltava rehellinen tietämyksensä suhteen. On varmistettava, että sidosryhmille syntyy oikea käsitys siitä, mistä koodari voi lähtökohtaisesti ottaa vastuuta. Hän ei esimerkiksi voi ottaa vastuuta ajoympäristössä olevasta, muiden tekemästä koodista ja dokumentaatiosta. On parempi tunnistaa ja eliminoida riskit esimerkiksi PoC:eilla, kuin törmätä ongelmiin ikään kuin yllätyksenä. Bugit ja "ominaisuudet" ovat osa nykypäivää. Älä anna niiden yllättää itseäsi ja sidosryhmiäsi. Valistuneet sidosryhmät osaavat antaa erittäin paljon arvoa tällaiselle rehellisyydelle.

Mitä enemmän projektissa on lähtökohtaisesti tuntematonta, sitä helpommin otetaan teknistä lainaa ja syntyy refaktoroinnin tarvetta. Myös tämä on haaste koodarille ja hänen organisaatiolleen. Miten projektoitaessa ja asiakkaan kanssa asia ymmärretään ja sovitaan refaktorointityöstä?

Joissakin projektointimenetelmissä, kuten Scrumissa, on mielestäni hienosti ymmärretty nykypäivän koodarityön luonne. Esimerkiksi työ-määräarviointi perustuu väistämättä jonkinlaisiin oletuksiin, jotka puolestaan voivat osoittautua vääriksi. Siksi pyritään varhaisessa vaiheessa havaitsemaan, mikäli oletukset ovat virheellisiä, jotta tilanteeseen päästään nopeasti reagoimaan.

Parhaimmillaan koodarin arki on luovaa ja sosiaalista työtä, jossa odotukset ja tulokset kohtaavat, koska kaikki ymmärtävät ja hyväksyvät työn luonteen. Jokainen pääsee hyödyntämään omia ominaisuuksiaan ja kokemuksiaan, sekä oppimaan uutta. Onnistumisen kokemukset jaetaan toimivassa yhteistyössä tulosta tekevän ihmisjoukon kesken.