

Miért lehetnek sikertelenek a Java EE projektek?

Adam Bien összeszedett [12 okot](#), amelyek miatt a Java EE (és más) projektek sikertelenül záródnak, jobb esetben nem a várakozásoknak megfelelő módon működnek:

1. A tervez jobban ismeri a PowerPoint programot, mint a fejlesztésközöket.
2. Pár DVD kell csak és pár óra elrepül, amíg az alap infrastruktúra (egy alkalmazás- és egy adatbázis szerver) feltelepül a gépedre.
3. Jó pár alkalmazás szerver esetén percekig tart, amíg elindul, s szintén percben mérhet id egy deploy - s ezt naponta tucatnyi alkalommal meg kell tenni.
4. Az alkalmazás szerver reprodukálható hibája lassan - vagy egyáltalán nem javul.
5. Nehéz olyan fejlesztői gépet találni, amelyen az Enterprise fejlesztésközök jól és gyorsan futnak...
6. A tervez imádja a rétegeket és a részekre osztott modelleket, s mire egy entitás eljut a perzisztencia rétegtől a prezentációs rétegig, jó sok osztályt kell példányosítani.
7. Minden szabadon konfigurálható és cserélhető, emiatt hatalmas az XML adattömeg... felmerül a kérdés: valóban szükséges bármit cserélhetre készíteni, ha a projektet elfogadták és élesítették (s ezért soha többé nem nyúlnak hozzá)?
8. A tervez végletekig a vízesés modell híve, esetleg tele van felkapott szakszavakkal és HBR-ekkel - amúgy szakmailag üres.
9. A fejlesztő túlpörgeti magát, komponensek tervezési mintáktól és elegáns megoldásoktól hemzsegnek. Esetleg a laza fejlesztő "csoda, hogy működik" kódot készít.
10. A kezdeti fellángolás után - fleg a hosszúra nyúló tervezés és fejlesztés okán - a résztvevők elvesztik lelkesedésüket.
11. HA, clustering, és a többi használata már egy "vendégkönyv" összetettség projekt esetén is.
12. A kód QA elírásai (például minden getter és setter dokumentálása) feleslegesen lekötik a fejlesztőt és az egekig növeli a fejlesztési költségeket.

Mindez hatványozottan igaz [banki szoftverek esetén](#)... :)