Kecskés Dániel
Témalabor beszámoló (kivonat)
2018/19 I. félév

ASP.NET Core szolgáltatások készítése DevOps metodika szerint

1 Bevezetés

1.1 A félév menete

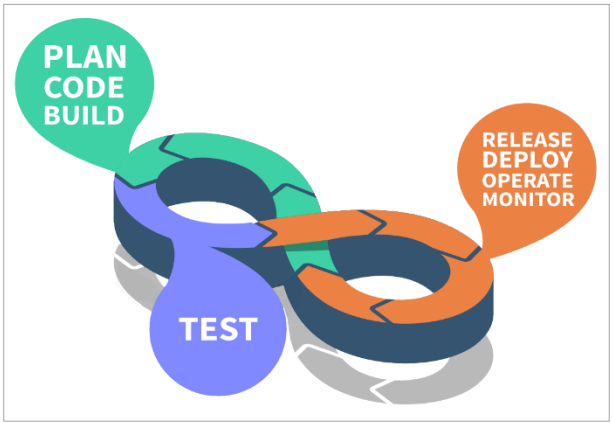
A Témalaboratórium nagyon tanulságos volt számomra, a félév alatt rengeteg technológiát volt szerencsém megismerni a tárgy keretein belül.

A félév első felében megismerkedtünk a DevOps felfogással és metodológiával, hétről hétre újabb és újabb technológiákat próbálhattunk ki. A félév második felében egy önálló projektet kellett elkészítenünk.

1.2 A DevOps

A DevOps mint feladatkör két szakterület találkozásából tevődik össze: a **fejlesztés**ből és az **üzemeltetés**ből, melyek napjainkban egyre inkább összeforrnak. Véleményem szerint ez egy olyan metodológia, ami arra törekszik, hogy minél gyorsabban, tervezettebben és megbízhatóbban tudjunk fejleszteni és üzemeltetni alkalmazásokat, szolgáltatásokat.

Mindezt **folytonos**an kívánjuk elérni, egy olyan folyamatmodell segítségével, mely iterációk sorozatából áll (*1. ábra*).



1. ábra: a DevOps folyamatmodell
(forrás: https://restlet.com/use-cases/api-first/devops/)

1.3 Az önálló feladat

A DevOps feladatcsoporton belül többek között a **Cloud Native** programozás keltette fel leginkább az érdeklődésemet. Cloud Native-nek nevezzük azokat az alkalmazásokat és szolgáltatásokat, melyek kezdettől fogva felhőben való futtatásra lettek fejlesztve. Céljuk, hogy minél jobban kihasználják a felhő adta lehetőségeket, többek között a szinte korlátlan számítási kapacitást, az igény szerinti elérést és a skálázhatóságot.

Az önálló feladatom választásakor a technológiai rálátásom bővítését tartottam szem előtt. Jelen esetben nem egy konkrét végeredmény elérése volt a cél, hanem az ahhoz vezető úton kipróbálni érdekesebb eszközöket, lehetőségeket.

Így esett a választásom az eddig használt JDK helyett a .NET keretrendszerre, Java nyelv helyett pedig **C#**-ra: egy **ASP.NET Core Cloud Native** webalkalmazást készítettem el. Munkám során többek között felállítottam és üzemeltettem egy *AWS Elastic Beanstalk* környezetet, megismertem több Visual Studio tesztelési eszközt, illetve az *AWS Toolkit-et Visual Studio*-hoz, használtam a Swagger keretrendszert a dokumentáció vizualizálására, dockerizáltam .NET Core környezetben és összekötöttem a fejlesztési és kiadási műveleteket egy automatizált, **Continuous Delivery** folyamattal.

A végeredmény egy egyszerűbb, 2+5 dokumentált API-t kiajánló webalkalmazás lett.