# Disszertáció

# Tartalomjegyzék

# Bevezető

Probléma: a problémát a munkahelyen fedeztem fel, a jelenlegi munkafolyamatot szeretném optimalizálni autómatizált teszteléssel.   
Röviden a munkám: Product Lifecycle Management szoftvert szabunk testre klienseknek. A szoftver neve Windchill, ami Spring keretrendszerre épülő webes applikáció.   
A probléma pedig ott adódik, hogy a fejlesztéseinket tesztelni kell. Jelenleg kézzel csináljuk ezt. Ez lassú. És valahányszor release-re kerül a sor, újra kell csinálni kézzel minden tesztet, hogy meggyőződjünk róla, hogy minden új funkció az elvártnak megfelelően üzemel és a már meglévő funkciókat nem befolyásolják. Ezt a teszt környezeten egy teszter végzi. Ezen rengeteg idő jelentős részét meg lehetne takarítani, ha a teszteket autómatizálnánk. A másik probléma pedig az, hogy nem lehet elmenteni és újra futtatni a teszteket, mert ugye azt mindig kézzel csináljuk. Itt jön képbe a Selenium. Autómatizálhatjuk a teszteket, szimulálhatjuk a végfelhasználó viselkedését és akárhányszor újra futtathatjuk azokat. Ennek meg van az az előnye is, hogy így megbizonyosodhatunk arról, hogy egy már meglévő funkciót nem rontottunk el egy új fejlesztéssel.

A Cucumber pedig azért hasznos választás, mert így a tesztek olvashatóak lesznek nem csak a fejlesztők, hanem a menedzserek számára is.

PTC Community oldalon publikálhatom a kutatást.

A cég belső portálján is publikálhatom az eredményeket.

Cél: hogy bebizonyítsam az autómatizált tesztek hozzájárulnak a szoftver minőségének javulásához és ezzel időt, energiát és nem utolsó sorban pénzt takaríthatunk meg.

# A kutatott technológiák bemutatása

### Selenium

Page Object Model

Selenium Server – remotely run our tests

### Cucumber

BDD – Behaviour Driven Development

# Szakirodalmi tanulmány

Kiss Tibor küldi

# Megvalósított rendszer bemutatása

Rendszer arhitektúra

- hibák

- sebesség

- jmeter - terhelhetőség

- comparatív összehasonlítás, keresni a piacon

- hogyan lehetne tesztelni azt, hogy a meglévő megy-e még ha hozzáadunk újat

- response idő, érdekesség szempontjából processzor idő

- adatbázis keresés, optimalizálás

- lekérdezések eredményei és időtartama

- eredmények naplózhatósága verziók között, összehasonlítás az előző verziókkal, mi csökken, mi nő, hatékonyság

# Következtetések és tapasztalatok

# Továbbfejlesztési lehetőségek

# Bibliográfia

Bryan Hansen: Automated Web Testing with Selenium and WebDriver Using Java <https://app.pluralsight.com/library/courses/automated-web-testing-selenium-webdriver-java/table-of-contents>

Catalin Tudose: Java: BDD Fundamentals

<https://app.pluralsight.com/library/courses/java-bdd-fundamentals/table-of-contents>