Lukáš Zapletal
lukas.zapletal@liberix.cz

Testing Driven Development

Test Driven Development

- TDD je způsob vývoje softwaru
- testy se píší před samotným kódem
- spouštění testů je automatizované
- krátké cykly mezi spouštěním
- častá refaktorizace (psaní kódu, který se zahodí)
- podpora xUnit

TDD cyklus

- a) napište test
- b) učiňte ty nejnutnější kroky, aby se to zkompilovalo a spustilo (nový test selže)
- c) napište kód
- d) znovu spusťte test
- e) refaktorujte a zlepšujte výsledky, opakujte

TDD

- testy postupně přibývají
- nové změny kódu mohou znefunkčnit staré
- programátor má před sebou vždy seznam testů, které musí zprovoznit
- malé krůčky
- pomocné nástroje mohou zachycovat historii spouštění testů, kontrolovat úspěšnost testů před vložením nového kódu do repozitáře, late commit a podobně

TDD

- malé krůčky jsou lepší než velké
- krátkodobé cíle jsou lepší než dlouhodobé
- díky TDD lze programovat flexibilně a modulárně
- časté přidávání kódu do repozitáře je klíčové (v řádech hodin)
- psaní testů a opakovaná refaktorizace je daní
- stejně tak trable v případě databází

Continuous integration

- každá, byť jen drobná, funkční změna by se měla hned commitnout do repozitáře
- před tím musí projít všechny testy
- jakmile jediný test selže, programátor nemůže změnu promítnout (udělat commit)
- existují nástroje pro podporu CI
- zlepšuje to flexibilitu týmu
- malé krůčky jsou lepší než velké skoky
- v pátek odpoledne není stres

TDD a CI

Následuje ukázka TDD metodiky v prostředích IntelliJ IDEA a Eclipse