

TDD och BDD

Test-Driven Development
Behaviour-Driven Development



Vad är ett test?

- Ett test är till för att validera att en funktionalitet eller ett system uppfyller krav
- Olika sorters test
 - Enhetstester (Unit Test)
 - Integrationstester
 - Regressionstestning
 - Röktestning (Smoke test)
 - Stresstest
 - Mfl...

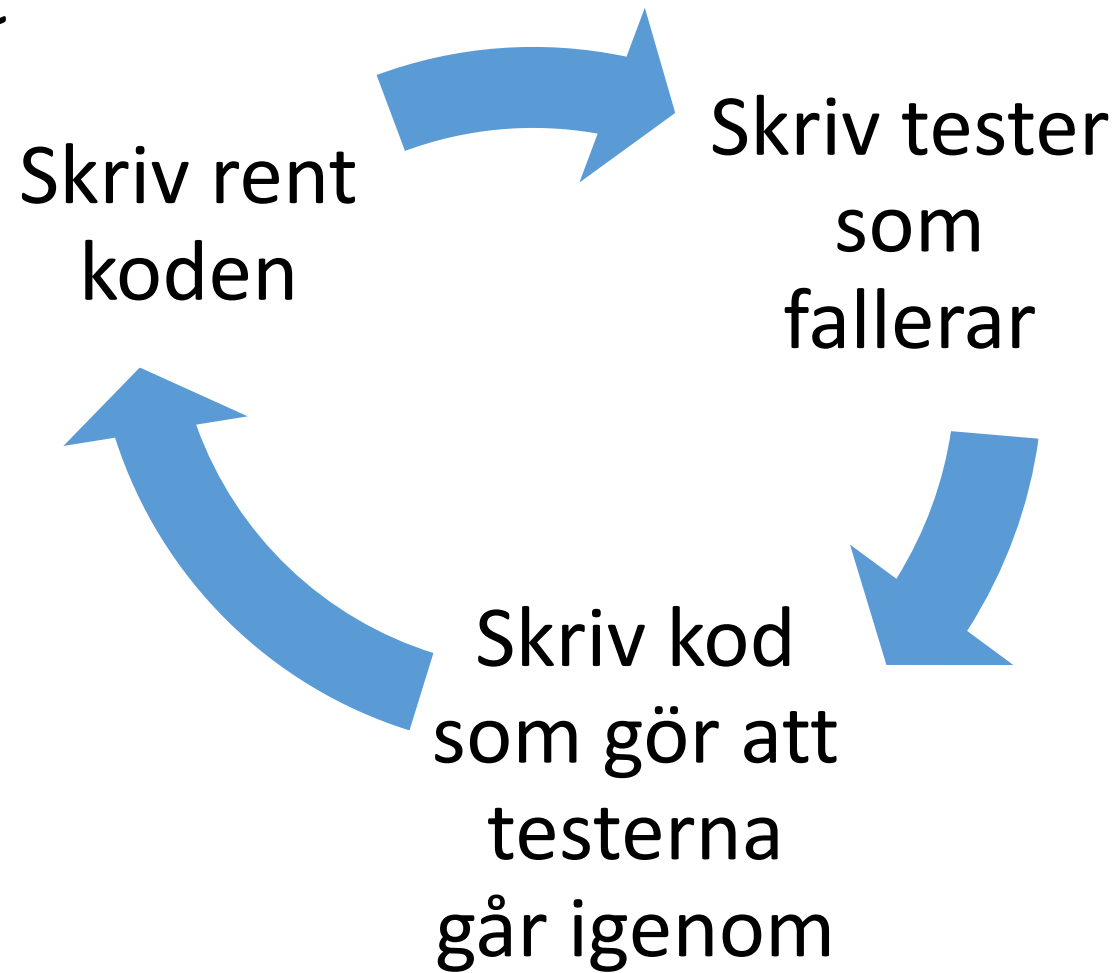
Enhetstestning

- Testa små delar
 - Klasser
 - Metoder
- Skrivs med kod
- Olika ramverk för enhetstestning i C#
 - xUnit
 - NUnit
 - MSTest

Mål

- Målet med både TDD och BDD är att fokusera på krav
- Att skriva kod som endast uppfyller kraven
- Att utgå ifrån tester som testar kraven och önskat beteende
- Att dela upp utvecklingsprocessen i förståelse av krav och implementation av krav

Arbetsflöde



Att skriva testfall

- Läs noga igenom kraven
- Se till att förstå kraven
- Fråga kund/produktägare/kollega om det är oklart
- Identifiera möjliga felfall (lägen där koden som ska testas kan krascha)
- Skriv testet först på papper/i text
- Skriv sedan testet i kod

Att skriva kod som gör att testerna går igenom

- I första steget är det okej att hårdkoda returvärdet
- Nästa varv i cykeln skriv kod som utför det som krävs

Regel:

” Du får inte skriva någon produktionskod om den inte är till för att få ett fallerande testfall att passera”

Att skriva enhetstester (Unit Tests)

Regel

”Du får inte skriva mer av unittestet än vad som behövs för fail.
Kompileringsfel räknas som fail”

Alternativ till TDD?

Hur hade du jobbat om du inte skrivit testerna först?

Ex:

1. Skrivit färdigt en metod
2. Skrivit testfall för metoden
3. Skrivit nästa metod

3 personer per grupp

Diskutera och skriv:

Alternativ till TDD?

Varför är TDD bra?

- Ex: Kommer alltid finnas testfall

3 personer per grupp

Diskutera och skriv:

Varför är testdriven utveckling bra?