Testautomatisering

BDD, Exempel

Idag

- FM:
  - Regexp
  - Lab 3
  - BDD
  - Kod-exempel
  - Handledning?
- EM:
  - Handledning

Regexp

- The first rule of regexp club…
- Om vi med rimliga medel kan undvika regexp: Undvik!

Regexp

- Regexp
- Om vi med rimliga medel kan undvika regexp: Undvik!
- Kodunderhåll
  - Komplexitet
  - Få har en djup förestående för regexp

- Regexp
- validates_format_of :email, :with => /^(|(([A-Za-z0-9]+_+)|([A-Za-z0-9]+\-+)|([A-Za-z0-9]+\.+)|([A-Za-z0-9]+\++))*[A-Za-z0-9]+@((\w+\-+)|(\w+\.))*\w{1,63}\.[a-zA-Z]{2,6})$/

Regexp

Regexp

- Regexp
- Djup förståelse ej centralt för utveckling/test
  - Dyker upp i test relativt ofta
  - Dyker upp i script-språk (ruby, PHP, perl, etc.) med jämna mellanrum
  - Dyker upp i kompilerade språk (C#, Java, etc.) mer sällan.
  - För de allra flesta – en väldigt smal delmängd räcker

Regexp

- Kraftfullt verktyg för vissa problem
  - Är [användarinput] en giltig mailadress?
  - Skriv en metod => 20-30 rader kod?
  - Regexp => 1 rad

- Är konvention i andra fall
  - cucumber step definitions

Regexp

- Om du inte hamnar i en sits där du dagligen använder regexp: ha ett humm om hur det funkar.

Regexp

- Crash course

Regexp

Regexp

- /^ $/
- Markerar början och slut
- /^text$/ matchar ”text”

Regexp

- Specialtecken:
- `^ $ \ [ ] . ? * + ( ) |`
- Om vi behöver matcha på något av ovanstående, använd \ innan
- 2\+3 matchar ”2+3”

Regexp

- Specialtecken:
- ^ $ \ [ ] . ? * + ( ) |
- [] matchar ett tecken motsvarande innehållet i dessa brackets
- [öÖ]vrigt matchar ”övrigt” eller ”Övrigt”
- [A-Za-z0-9] matchar alfanumeriskt tecken
- [0-9_] matchar ”0”-”9” eller ”_”

Regexp

- Specialtecken:
- ^ $ \ [ ] . ? * + ( ) |
- [] matchar ett tecken motsvarande innehållet i dessa brackets. Obs! ändrat beteende för specialtecken
- [\+0] matchar ”\”,”+” eller ”0”
- ^innuti brackets: ”utom: ”
- [^”] matchar alla tecken utom ””””

Regexp

- Specialtecken:
- ^ $ \ [ ] . ? * + ( ) |
- . Catch all. Matchar alla tecken utom radbrytning
- . matchar ett tecken, vilket som helst, t.ex. ”¤” eller ”4” (matchar ej ”\n” eller ”\r”)

Regexp

- Specialtecken:
- ^ $ \ [ ] . ? * + ( ) |
- ? Optional character
- /^colou?r$/ matchar ”color” eller ”colour”

Regexp

- Specialtecken:
- ^ $ \ [ ] . ? * + ( ) |
- * repetition, 0 eller fler gånger
- B[a-z]*D matchar ”BD”, ”BaD”, ”BabcD”, etc.
- Watchout: greed
- /^c*$/ matchar ”” i ”cccc”

Regexp

- Specialtecken:
- ^ $ \ [ ] . ? * + ( ) |
- + repetition, 1 eller fler gånger
- [0-9]+ matchar ”3”, ”1248”, men inte ””

Regexp

- Specialtecken:
- ^ $ \ [ ] . ? * + ( ) |
- {X,Y} repetition, X till Y gånger
- [0-9]{2,4} matchar ”32”, ”1248”, men inte ””, ”3”, ”14567”

Regexp

- Specialtecken:
- ^ $ \ [ ] . ? * + ( ) |
- () gruppering + referens
- ([A-Z][a-z]+^s) matchar t.ex. ”Cat”

Regexp

- Specialtecken:
- ^ $ \ [ ] . ? * + ( ) |
- | eller
- (a1)|(b1) matchar ”a1” eller ”b1”

Regexp

- Specialtecken:
- Characters: \w \d \s \t \n \r
- \w: word-character (bokstäver, ”-”)
- \d: digit
- \s: white space
- \t: tab
- \n: new line
- \r: carriage return
- \w\s\d matchar en bokstav följt av mellanrum, följt av en siffra

Regexp

- Specialtecken:
- `^ $ \ [ ] . ? * + ( ) |`
- /^I should see ”([^”]+)”$/ matchar ”I should see ” följt av ett eller flera tecken innanför ”, som ej innehåller ” och ger oss en referens till det som matchats innanför parenteserna.

Regexp

- Experiment:
- http://gskinner.com/RegExr/
- Regexp + text
- Se vad som matchar

Regexp

- Kort övning:
- Hittas på bloggen – dokument/filer under dagens lektion.
- Försök matcha så gott du kan – så går vi igenom det gemensamt sedan
- Sammarbeta om ni vill

- Källa: http://www.regular-expressions.info/quickstart.html

Regexp

- Lab 3

Lab3

- Lab 3
  - Rspec + Cucumber
    - Cucumber till vad?
    - RSpec till vad?
    - Varför växlar vi?
  - BDD
    - BDD – cykeln
    - Vinster?
    - Jämfört med TDD?

Lab3

- Filstruktur
  - /
    - features/ - Cucumber Features
    - specs/ - RSpec Examples
    - lib/ - Implementation
    - bin/ - Executables

Lab3

- Filstruktur
  - /
    - features/
      - support/ <- hjälpfiler för cucumber
      - step_definitions/ <- step definitions för cucumber
    - specs/
      - codebreaker/ <- mirror av lib/
    - lib/
      - codebreaker/ <- klasser som krävs för impl.
    - bin/

Lab3

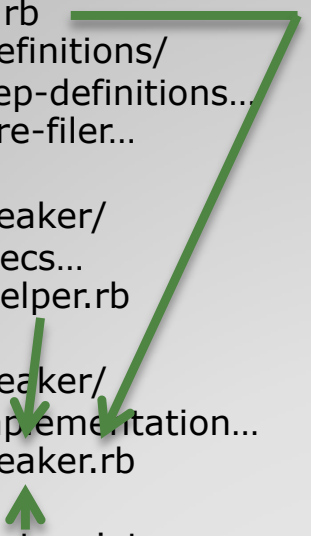
- Filstruktur
  - /
    - features/
      - support/
        - env.rb <- talar om för cucumber var SUT är
      - step_definitions/
    - specs/
      - codebreaker/
      - spec_helper.rb <- talar om för rspec var SUT är
    - lib/
      - codebreaker/
    - bin/

Lab3

- Filstruktur
  - /
    - features/
      - support/
        - env.rb
      - step_definitions/
        - …step-definitions… <- step-definitions för cucumber
      - …feature-filer… <- .feature-filer för cucumber
    - specs/
      - codebreaker/
        - …specs… <- specs för rspec
      - spec_helper.rb
    - lib/
      - codebreaker/
        - …implementation… <- 1 fil/klass för implementationen
      - codebreaker.rb
    - bin/
      - …körbart script… <- körbart script

Lab3

- Filstruktur
  - /
    - features/
      - support/
        - env.rb
      - step_definitions/
        - …step-definitions…
      - …feature-filer…
    - specs/
      - codebreaker/
        - …specs…
      - spec_helper.rb
    - lib/
      - codebreaker/
        - …implementation…
      - codebreaker.rb
    - bin/
      - …körbart script…



Features, Specs och körbart script refererar till codebreaker.rb

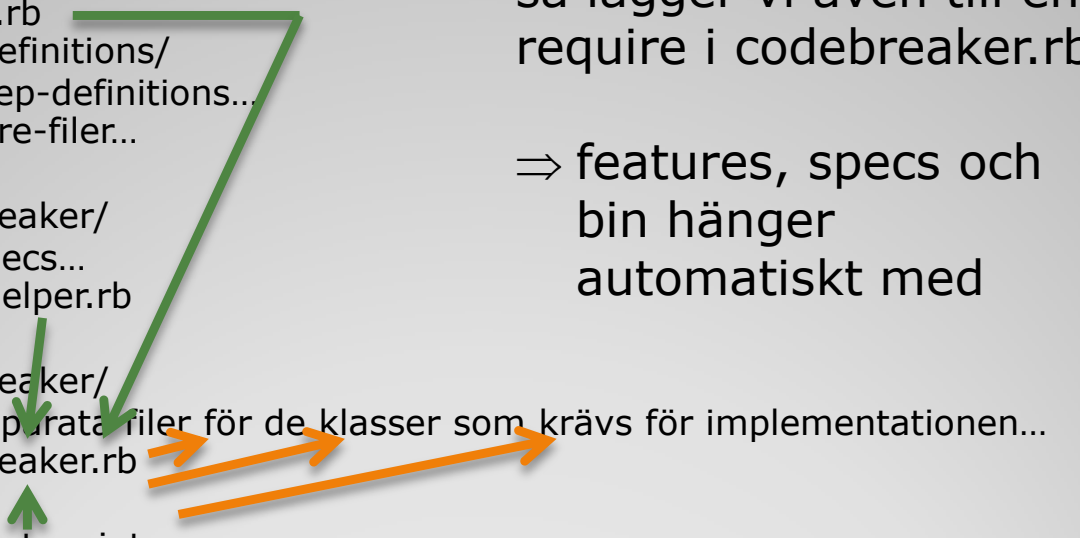
Lab3

- Filstruktur
  - /
    - features/
      - support/
        - env.rb
      - step_definitions/
        - …step-definitions…
      - …feature-filer…
    - specs/
      - codebreaker/
        - …specs…
      - spec_helper.rb
    - lib/
      - codebreaker/
        - …separata filer för de klasser som krävs för implementationen…
      - codebreaker.rb
    - bin/

codebreaker.rb refererar till de klasser som krävs för implementationen.

Lab3

- Filstruktur
  - /
    - features/
      - support/
        - env.rb
      - step_definitions/
        - …step-definitions…
      - …feature-filer…
    - specs/
      - codebreaker/
        - …specs…
      - spec_helper.rb
    - lib/
      - codebreaker/
        - …separata filer för de klasser som krävs för implementationen…
      - codebreaker.rb
    - bin/
      - …körbart script…



När vi lägger till en ny fil i implementationen, så lägger vi även till en require i codebreaker.rb

⇒ features, specs och bin hänger automatiskt med

Lab3

Output

```
~ $ ./mastermind
Welcome!
Enter guess:
> 1234
Mark: ++-
Enter guess:
>
```

```
~ $ ./mastermind
Welcome!
:
>
```

Lab3

Lab3

- Output
- Utskrift till konsolen är extremt intressant att testa
- Det är dock ett integrationstest (eller rent av systemintegrationstest)
- mastermind <-> kommandotolk

- Output
- Utskriftsbeteendet är dock centralt för MasterMind => vi vill fånga det i enhetstester
- Hur gör vi detta?

Lab3

- Output

```
Class Output
  def write(msg)
    puts msg
  end
end
```

- puts ”Welcome!” => @output.write(”Welcome!”)
- double(’output’).should_receive(:write).with(”Welcome!”)

Lab3

- Test Doubles
- Test Doubles är flitigt använda när vi är beroende av komplexa objekt
- Ex: Databas-kopplingar, Mail-tjänster, Nätverksanrop, etc, etc.

Lab3

- Test Doubles

- De är också ett smidigt sätt att se till att vi inte testar andra klasser/objekt än de vi faktiskt vill testa

Lab3

- Komplexitet
- Enkelt program -> Oväntat komplext
- …~1 månad
- Enkel blog: ramverk + Faktor 10-20?

Lab3

- Copy + Paste
  - Duplicerad kod -> svårare underhåll
  - Utöver detta: ngt som inte fungerar

Lab3

- Vad är intressant här
  - Vi får en levande specifikation
  - Vi får regressionstester

Lab3

- Refactoring
  - Det viktigaste steget i den här processen
    - …sett ur ett längre perspektiv
  - Snabbare utvecklingstakt
    - Naiv lösning + tester => profit
  - Förmåga att lägga till nya features
  - Snabbt identifiera och rätta buggar
  - Överföra kunskap

Lab3

- Var kommer ni in i detta
  - Agile: Whole team
  - Cucumber Features + Scenarios

BDD

Exempel

- Ett kort exempel

Nästa vecka

- FitNesse

- ...

Fin