

# Testautomatisering

FitNesse

- FM:
  - Lab 3, utvärdering
  - Lab 3 – kort snack
  - FitNesse
  - Snabb kik på hur vi startar FitNesse
  - FitNesse Installation

**Idag**

- Lab3 - Snabbutvärdering
  - 1. Hur många timmar har du lagt?
  - 2. Hur många ytterligare timmar kommer du lägga?
  - 3. Svårighet: För **Lätt** / **Lagom** / För **Svår**
  - 4. Lärdommar - "Jag ser...":
    - Ingen praktisk nytta (-)
    - Viss praktisk nytta (**0**)
    - Stor praktisk nytta (+)
  - 5. Labbformatet:
    - Dåligt(-)   -   Varken eller(**0**)   -   Bra(+)

**Snabbutvärdering**

- Lab 1 + 2

- De dåliga nyheterna är att jag inte hunnit rätta färdigt dem
- De bra nyheterna är att jag är klar med AD-eleverna nu, så jag räknar med att hinna med lab 1-3 på onsdag.
- Lab 3 kommer gå snabbare att rätta (Peppar, peppar)

**Lab 1 + 2**

- Kort Lab 3 snack

**Kort om labben**

- Hur mycket skall man ha förstått?
  - Förstå de grova dragen i processen
  - Vilket verktyg är bra för vilken uppgift
  - Vilket perspektiv har vi när vi jobbar med de olika verktygen

## Lab 3

- Hur mycket skall man ha förstått?
  - Ju mer ni programmerar, desto bättre testare kommer ni bli.

**Lab 3**

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

**Lab3**



- Copy + Paste

- Duplicerad kod -> svårare underhåll
- Utöver detta: vi kan missförstå vad koden gör

- Vad är intressant med BDD?
  - Vi får en levande specifikation
  - Vi får regressionstester

# • 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

- Refactoring

- Målbilden är en liten refactoring var ~5:e min
- Och en lite större refactoring för varje Feature alt. varje dag.
- Spara till i slutet av en Sprint => Ingen refactoring

**Lab3**

- Var kommer ni in i detta
  - Agile: Whole team
  - Cucumber Features + Scenarios
    - => Bra om ni har förståelse för hela processen

**BDD**

- Om testramverk

**BDD**

- FitNesse kommer vara det sista test ramverket vi tar upp i den här kursen.
  - Mitt råd: Välj något av dessa ramverk och försök utforska det djupare.
  - Kan vara en konkurrensfördel
  - Ger er tid till annat: T.ex. utforskande tester

**FitNesse**

- Den bistra sanningen

- 1 vecka/ramverk => bara intro
- Ni kommer glömma snabbt om ni inte på något sätt håller kunskaperna levande



- Men!

- Kan ni sitta 1-2h/vecka och experimentera/utforska => Ni blir bättre över tid
- Kan ni sitta 1h/dag => Ni kommer ganska snart bli bekvämma med ert verktyg

**FitNesse**

- FitNesse

**FitNesse**

- FitNesse

- Wiki
- WebServer
- TestRamverk

**FitNesse**

- Test i FitNesse visas som en Wikisida
- Test i FitNesse skapas och editeras på en WikiSida
- Test i FitNesse körs på wikisidan, via en test-knapp

**FitNesse**

- Acceptance Test Framework
- Dvs. vi jobbar med User Stories, Scenarios, Acceptanskriterier, eller dylikt.
- Vi betraktar SUT utifrån.

**FitNesse**

- Används generellt sett inte med Cucumber
- Däremot används det med fördel tillsammans med något Unit Test Ramverk
  - Ex. xUnit eller Rspec

- 

**FitNesse**

- Om något känns oklart – fundera på ifall det går dra någon analogi med Cucumber

**FitNesse**

- En snabb översikt
  - SUT (System Under Test)
  - Fixture (Slim)
  - WikiSida med ett eller flera Test i olika Tabell

**FitNesse**



- En snabb översikt
  - Fixture (Slim)
  - En väldigt tunn "Översättare" mellan FitNesse och SUT
  - Ex. skulle vi kunna ha kolumnerna guess och mark? i en tabell.
    - Hur vet FitNesse var den skall skicka anropet med guess?
    - Genom Slim! (Eller Fit – en annan, äldre fixture).
  - (Tänk på hur era Scenario Outlines såg ut förra veckan – Det finns tydliga paralleller med Cucucumber)

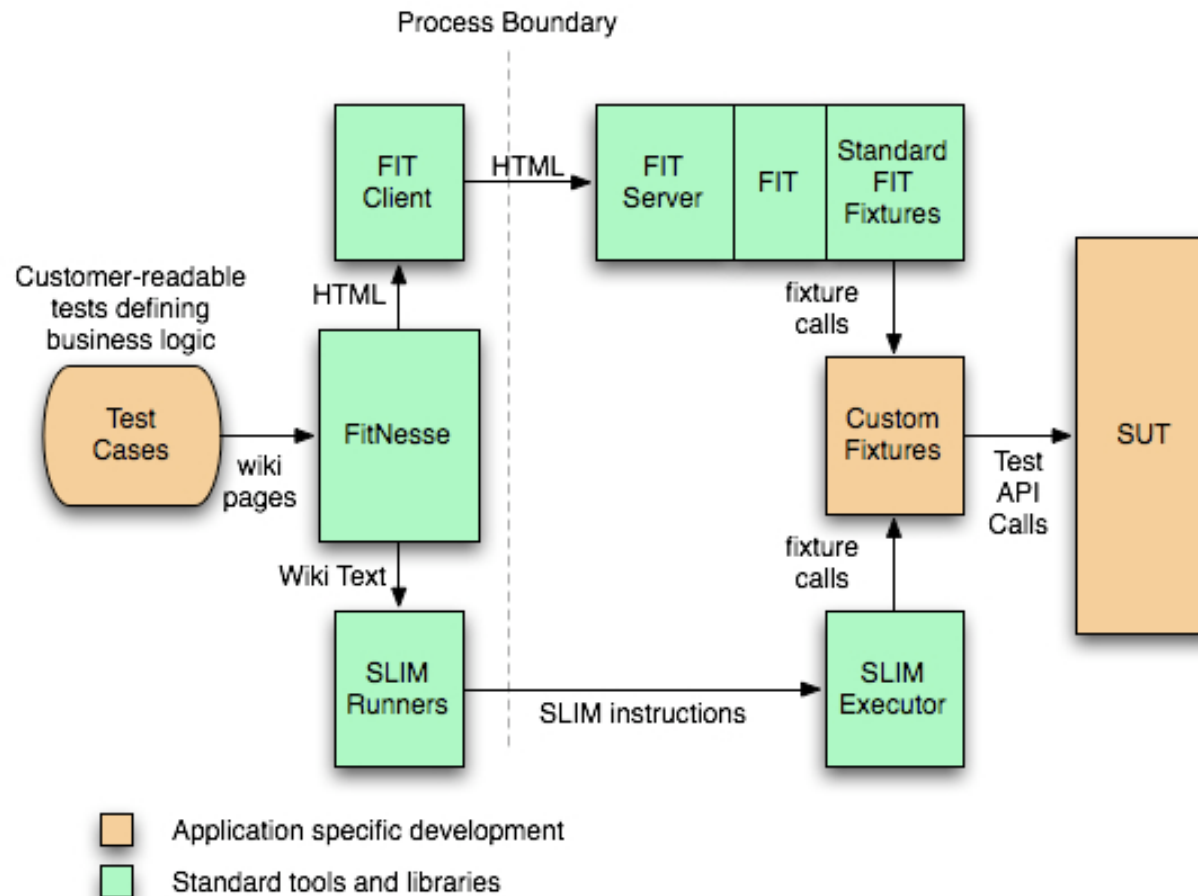
**FitNesse**

- En snabb översikt

- FitNesse "guess" -> Slim.guess -> SUT.guess  
|  
v
- FitNesse "mark?" <- Slim.mark <- SUT.mark

**FitNesse**

## FitNesse Test Systems.



# FitNesse

- Exempel på tabell/wikisida:

**FitNesse**



[FitNesse](#) > [UserGuide](#)

## TwoMinuteExample

Failure Navigator

< of 1 >

Tests Executed OK

Test

Tools

**x Assertions:** 5 right, 1 wrong, 0 ignored, 0 exceptions (0.005 seconds)

[▶ Precompiled Libraries](#)

[Expand All](#)

[Collapse All](#)

[A One-Minute Description](#)

### An Example FitNesse Test

If you were testing the division function of a calculator application, you might like to see some examples working. You might want to see what you get back if you ask it to divide 10 by 2. (You might be hoping for a 5!)

In [FitNesse](#), tests are expressed as tables of **input** data and **expected output** data. Here is one way to specify a few division tests in [FitNesse](#):

eg.Division		
numerator	denominator	quotient?
10	2	5.0
12.6	3	4.2
22	7	3.142857142857143~3.14
9	3	3.0<5
11	2	4<5.5<6
100	4	[25.0] expected [33]

# FitNesse

- Grundpelaren för att köra test:
- Wikisida/Test Table
  - (Generellt Decision Tables - Dvs: varje rad i en table skall svara ja eller nej på en fråga - i det här fallet blev resultatet det vi förväntat oss i fem fall av 6)
- Annorlunda uttryckt: Vi har ett antal input-fält och ett fält med förväntat utfall



[FitNesse](#) > [UserGuide](#)

## TwoMinuteExample

Failure Navigator

< of 1 >

Tests Executed OK

Test

Tools

**x Assertions:** 5 right, 1 wrong, 0 ignored, 0 exceptions (0.005 seconds)

[Precompiled Libraries](#)

[Expand All](#)

[Collapse All](#)

[A One-Minute Description](#)

### An Example FitNesse Test

If you were testing the division function of a calculator application, you might like to see some examples working. You might want to see what you get back if you ask it to divide 10 by 2. (You might be hoping for a 5!)

In [FitNesse](#), tests are expressed as tables of **input** data and **expected output** data. Here is one way to specify a few division tests in [FitNesse](#):

eg.Division		
numerator	denominator	quotient?
10	2	5.0
12.6	3	4.2
22	7	3.142857142857143~3.14
9	3	3.0<5
11	2	4<5.5<6
100	4	[25.0] expected [33]

# FitNesse

- Vi kan ordna flera Test Tables i en Test Suite
  - Kan vara en modul i SUT
  - Eller systemet som helhet

**FitNesse**





[FitNesse](#) > [SuiteAcceptanceTests](#)

## SuiteWidgetTests

slim

Suite

Tools

variable defined: `TEST_SYSTEM=slim`

### Contents:

- [Contents Tests Include](#)
- [Contents Tests Include With Help](#)
- [Contents Usage](#)
- [Set Up](#)
- [Set Up Link Test](#)
- [Should Not Be Able To Include Parent Page](#)
- [Simple Widgets ...](#)
- [Test Backwards Search Widget](#)
- [Test Broken Wiki Word Link](#)
- [Test Classpath Containing Variable](#)
- [Test Classpath Preceded By Define](#)
- [Test Collapsible Sections](#)
- [Test Comment](#)
- [Test Contents](#)
- [Test Contents Filters](#)
- [Test Contents Help](#)
- [Test Contents Properties](#)
- [Test Contents Regraced](#)
- [Test Contents With Recursion](#)
- [Test Email Widget](#)
- [Test Evaluator](#)
- [Test Existing Wiki Word Link](#)
- [Test Expression](#)
- [Test Gt Sub Page Widget](#)
- [Test Gt Sub Page Widget In Alias](#)

# FitNesse

- När vi kör en Suite så kommer alla testerna i Suiten köras
- Ett sida ingår i en Suite om den är markerad att vara en testsida och är en sida som är ordnad under aktuell Suite i Wiki-hierarkin.

- Denna Suite råkade ha en helt platt struktur

**FitNesse**



[FitNesse](#) > [SuiteAcceptanceTests](#)

## SuiteWidgetTests

slim

Suite

Tools

variable defined: `TEST_SYSTEM=slim`

### Contents:

- [Contents Tests Include](#)
- [Contents Tests Include With Help](#)
- [Contents Usage](#)
- [Set Up](#)
- [Set Up Link Test](#)
- [Should Not Be Able To Include Parent Page](#)
- [Simple Widgets ...](#)
- [Test Backwards Search Widget](#)
- [Test Broken Wiki Word Link](#)
- [Test Classpath Containing Variable](#)
- [Test Classpath Preceded By Define](#)
- [Test Collapsible Sections](#)
- [Test Comment](#)
- [Test Contents](#)
- [Test Contents Filters](#)
- [Test Contents Help](#)
- [Test Contents Properties](#)
- [Test Contents Regraced](#)
- [Test Contents With Recursion](#)
- [Test Email Widget](#)
- [Test Evaluator](#)
- [Test Existing Wiki Word Link](#)
- [Test Expression](#)
- [Test Gt Sub Page Widget](#)
- [Test Gt Sub Page Widget In Alias](#)

# FitNesse

- Men vi kan ordna en Suite efter t.ex. Feature eller vad som helst annat vi kan tänka oss
- Wiki-strukturen ger oss stor frihet när det kommer till hur vi strukturerar våra test.

**FitNesse**

- När vi editerar ett test så gör vi det med Wiki-syntax
- Detta kommer kännas igen hyfsat om någon av er editerat wiki sidor innan.



[FitNesse](#) > [UserGuide](#)

## TwoMinuteExample

Help text:

Tags:

Spreadsheet to FitNesse

FitNesse to Spreadsheet

Format



Insert Template



wrap



rich text



plain text

```
!*< Hidden
!define TEST_SYSTEM {slim}
*!
[[A One-Minute Description][OneMinuteDescription]]
!1 An Example !-FitNesse-! Test
If you were testing the division function of a calculator application, you might like to see some examples working. You might
In FitNesse, tests are expressed as tables of '''input''' data and '''expected output''' data. Here is one way to specify a fe
|eg.Division|
|numerator|denominator|quotient?|
|10        |2          |5.0    |
|12.6      |3          |4.2    |
|22        |7          |~3.14  |
|9         |3          |<5     |
|11        |2          |4<_<6  |
|100       |4          |33     |

This style of FitNesse test table is called a [[Decision Table][Slim.DecisionTable]], each row represents a complete scenario
!3 Running our test table: Click the Test button
Before we do another thing, let's run this test table. See the little blue and white '''Test''' button in the upper-left, just
Ah, color! In the green cells, we got back the expected values from our code. When we divided 10 by 2, we expected and got bac
What about red? A cell turns red when we get back a different value than what we expected. We also see two values: the '''expe
```

# FitNesse

- Lägg märke till att tabellen är väldigt likt hur vi skapade tabeller i cucumber scenarios.



- I den här kursen kommer vi köra FitNesse lokalt
- Det finns dock ingenting som hindrar att man lägger upp en FitNesse server på t.ex. ett företags server.
- (Vi kan också styra rättigheter för olika användare, etc).

**FitNesse**

- Wikis är byggda för att det skall vara relativt enkelt för vem som helst att med ganska lite övning editera dem.
- Detta gör det till ett utmärkt verktyg för samarbete över de traditionella gränserna (Precis som Cucumber, f.ö.)
- Det gör det också till ett utmärkt vertyg för gradvis förfining av våra test / test suites

**FitNesse**

- Sidtyper
- Varje sida har en rad egenskaper (du kommer åt dem genom att klicka på tools-knappen – eller genom att lägga till .properties på adressen).
- En sida kan t.ex. vara Test, Suite, Ignored, etc.

**FitNesse**

- Wikisyntax:
- Se `/FitNesse.UserGuide.QuickReferenceGuide`

**FitNesse**

- Källor:

- De främsta källorna för information som behövs denna vecka är
- Snabbguiden: `/FitNesse.UserGuide.QuickReference`
- Användarguiden: `/FitNesse.UserGuide`
- ...mer om slim imorgon
- Ni kommer åt dessa antingen via er lokala installation eller fia [fitnesse.org](http://fitnesse.org)

**FitNesse**

- Snabbguide FitNesse

**FitNesse**

- Dagens Uppgift:

## Ladda hem FitNesse

- fitnessse-standalone.jar från [fitnesse.org](http://fitnesse.org)
- (+ ev. java? Kräver java v.6+: kontrollera med "java -version" )
- starta FitNesse med "java -jar fitnessse-standalone.jar" – på en rad.
  - (För Virtual Machine - starta med: "java -jar fitnessse-standalone.jar -p 8080" – på en rad)

**FitNesse**

- Dagens Uppgift:

Ladda hem FitNesse

- Troubleshooting: Se installations-sidan
- Om ni har en webbserver eller något liknande igång – se till att ni startar FitNesse på en annan port, med växeln, -p. Se föregående sida för exempel.

**FitNesse**



- Dagens Uppgift:

## Efter installation av FitNesse

- Öppna en browser och Gå in på: localhost
  - (För Virtual Machine: localhost:8080)
- Gå in på Two-Minute Example från startsidan och se till så att du kan köra testet från denna sida.

**FitNesse**

- Dagens Uppgift:

## Efter installation av FitNesse

- Prova att bygga lite enkla Wiki-konstruktioner mha QuickReferenceGuide (För att kunna skriva några test behöver vi dock veta något om slim)
- Testa att göra följande: Formattera text (bold, italic, etc), Skapa En tabell, Skapa en helt ny undersida med en länk tillbaka till föräldern, Lägg till en timestamp för när sidan senast modifierades
- Du kan editera vilken sida som helst genom att skriva "?edit" efter adressen.

**FitNesse**

- Dagens Uppgift:

Efter installation av FitNesse

- Vid tid över: Lab 3, eller Gå igenom FitNesse tutorial ([localhost/FitNesse.UserGuide](http://localhost/FitNesse.UserGuide))

**FitNesse**

- FitNesse + Ruby

**Imorgon**

...

Fin