

Tools @



**klarna**

Simpler   Safer   More Fun



Tobias Lindahl

## Code Management

The Problem  
The Solution  
The Analysis  
The Results  
The Metrics  
The Details

The Graphs

...more

## Tools @ **klarna**



Simpler Safer More Fun

...more

...more

...more

...more

...more

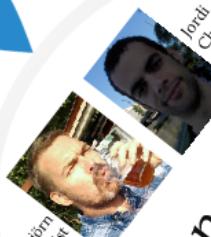
David Evans



## Testing

Unit Testing  
Continuous Integration  
GUI Testing  
Acceptance Testing

Jordi Chacon



Torbjörn Törnkvist

## Translation "polish"

The Problem  
The Solution  
Using GoTest  
Background  
The Problem  
About Peacock  
DEMOS TEST!



Tobias Lindahl

# Code Management

## The Problem

- Organically grown code
- Active application structure
- No clear cuts in functionality

## Our solution

- Move existing modules to new applications.
- Group applications in clusters (layers).
- Fix dependencies and cluster dependencies.
- Refactor code to satisfy constraints.
- Profit!

## The Analysis

- Find dependencies between modules.
- Sort them from applications and layers
- Write them down in a file
- Make extremely cool graphs

## The Action

- Generate shell scripts for
- Creating new application structure
- Moving the code
- Patching incisor paths, Makefiles, etc

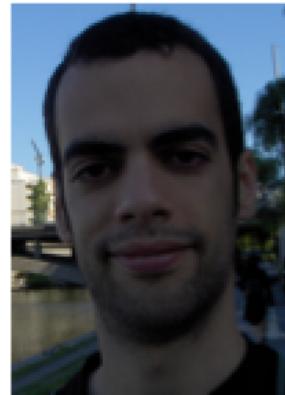
## The Graphs



Torbjörn  
Törnvist



Jordi  
Chacon



# Translation "POlish"

Background

Using GetText

The Problem  
is causing increasing headache!

The Solution



About POlish

- POlish is a ui-layer developed by Klamo
- Web-GL built on top of gettext to easily access PO-Files
- Provides a simple and friendly interface for translators
- Log in via OpenID
- Keep track of what is untranslated
- Allows contributors to search for terms
- Extreme correctness of translations
- Started with a simple command
- Has been used for the last three months

DEMO TIME!  
<http://www.klamo.com/polish/>

New features  
coming... or not

- REST Interface
- Translation memory
- Spell checker
- Online service
- git@github.com:cent/pnish.git

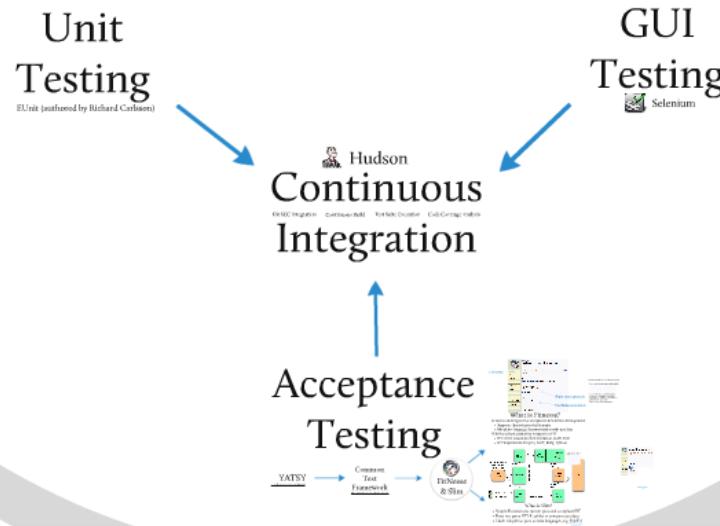
David  
Evans



Jia  
Wang



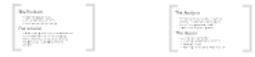
# Testing





Tobias Lindahl

## Code Management



The Graph



## Tools @ klarna

Simpler Safer More Fun



Thunkit  
...the most efficient way to work with your application's UI. It's a visual editor for building and testing user interfaces.

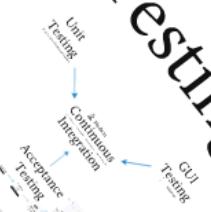


David Evans



Jia Wang

# Testing



Jordi Chacon  
Torbjörn Törnkvist

# Translation "polish"



Poedit  
...a fast, simple and reliable internationalization and localization tool for translators and developers.

POEdit  
...a fast, simple and reliable internationalization and localization tool for translators and developers.

Demo Trans



Background  
The Problem  
Using GetText  
About Poedit  
Demo Trans



Tobias Lindahl

# Code Management

## The Problem

- Organically grown code
- Active application structure
- No clear cuts in functionality

## Our solution

- Move existing modules to new applications.
- Group applications in clusters (layers).
- Fix dependencies and cluster dependencies.
- Refactor code to satisfy constraints.
- Profit!

## The Analysis

- Find dependencies between modules.
- Sort them from applications and layers
- Write them down in a file
- Make extremely cool graphs

## The Action

- Generate shell scripts for
- Creating new application structure
- Moving the code
- Patching incisor paths, Makefiles, etc

## The Graphs



# The Problem

- Organically grown code.
- Ad-hoc application structure.
- No clear cuts in functionality.

# Our solution

- Move existing modules to new applications.
- Group applications in clusters (layers).
- Put constraints on cluster dependencies.
- Refactor code to satisfy constraints.
- Profit!

# The Analysis

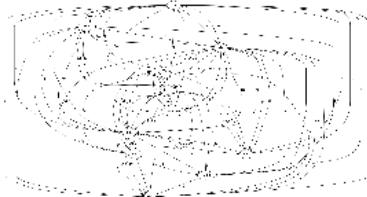
- Find dependencies between modules.
- Divide into applications and layers
- Write boring statistics to a file
- Make extremely cool graphs

# The Action

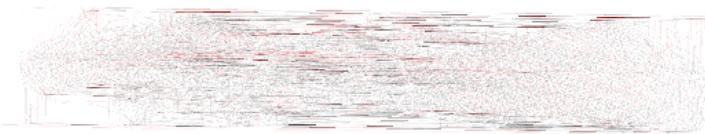
- Generate shell scripts for
  - Creating new application structure
  - Moving the code
  - Patching include paths, Makefiles, etc

# The Graphs

Original application structure



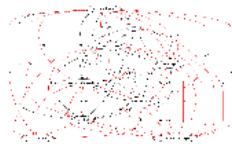
New improved(?) application structure



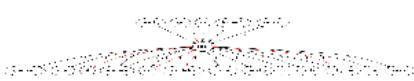
Layers to the rescue



but some bad dependencies remain



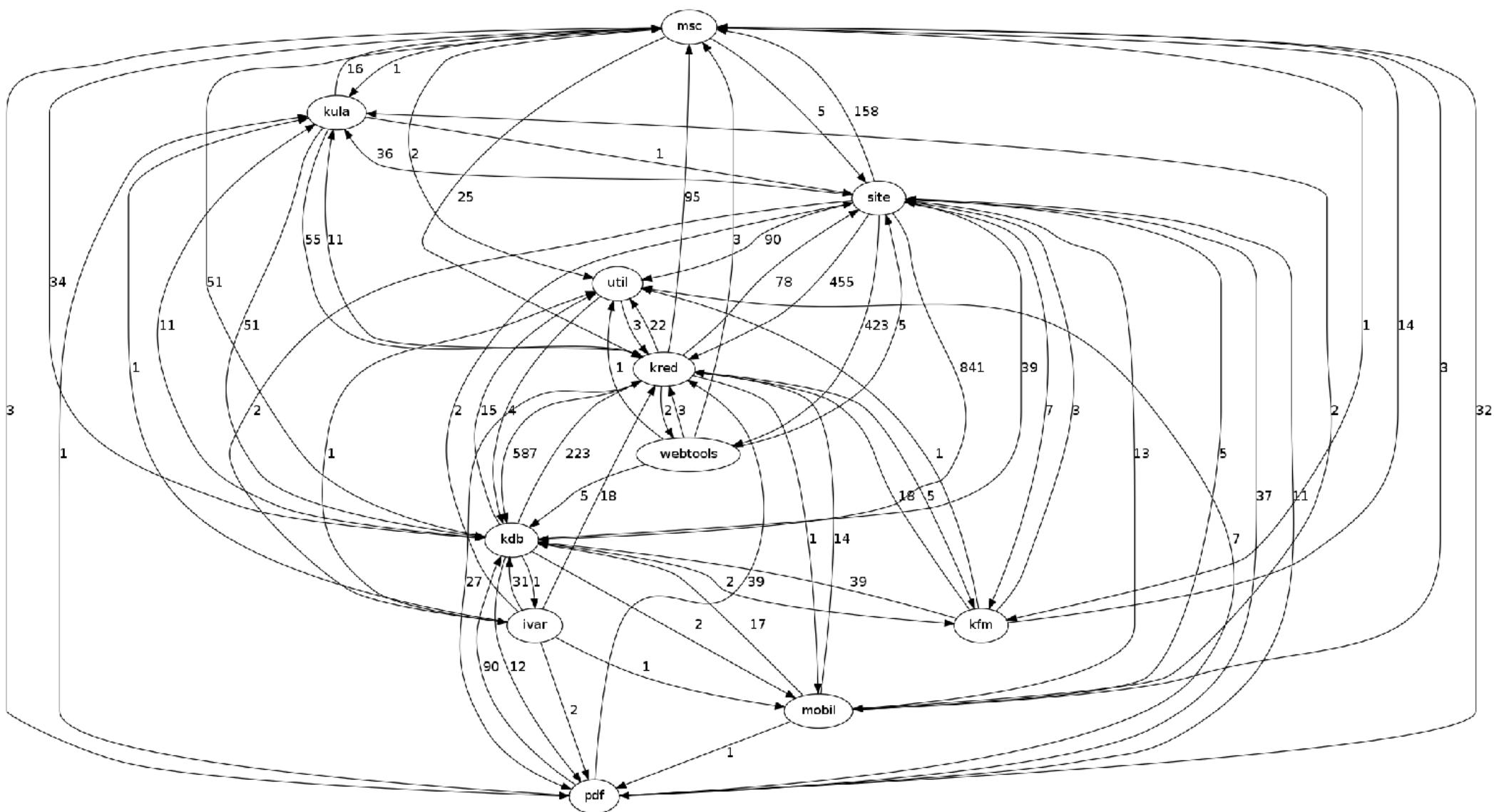
and then to the policy application



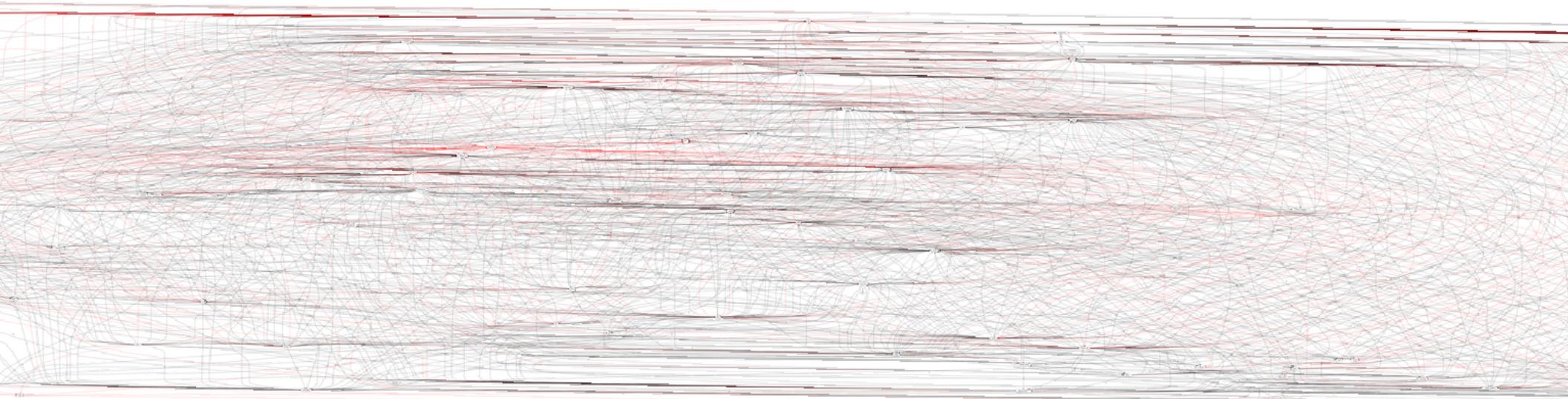
Zoom in on Business Layer



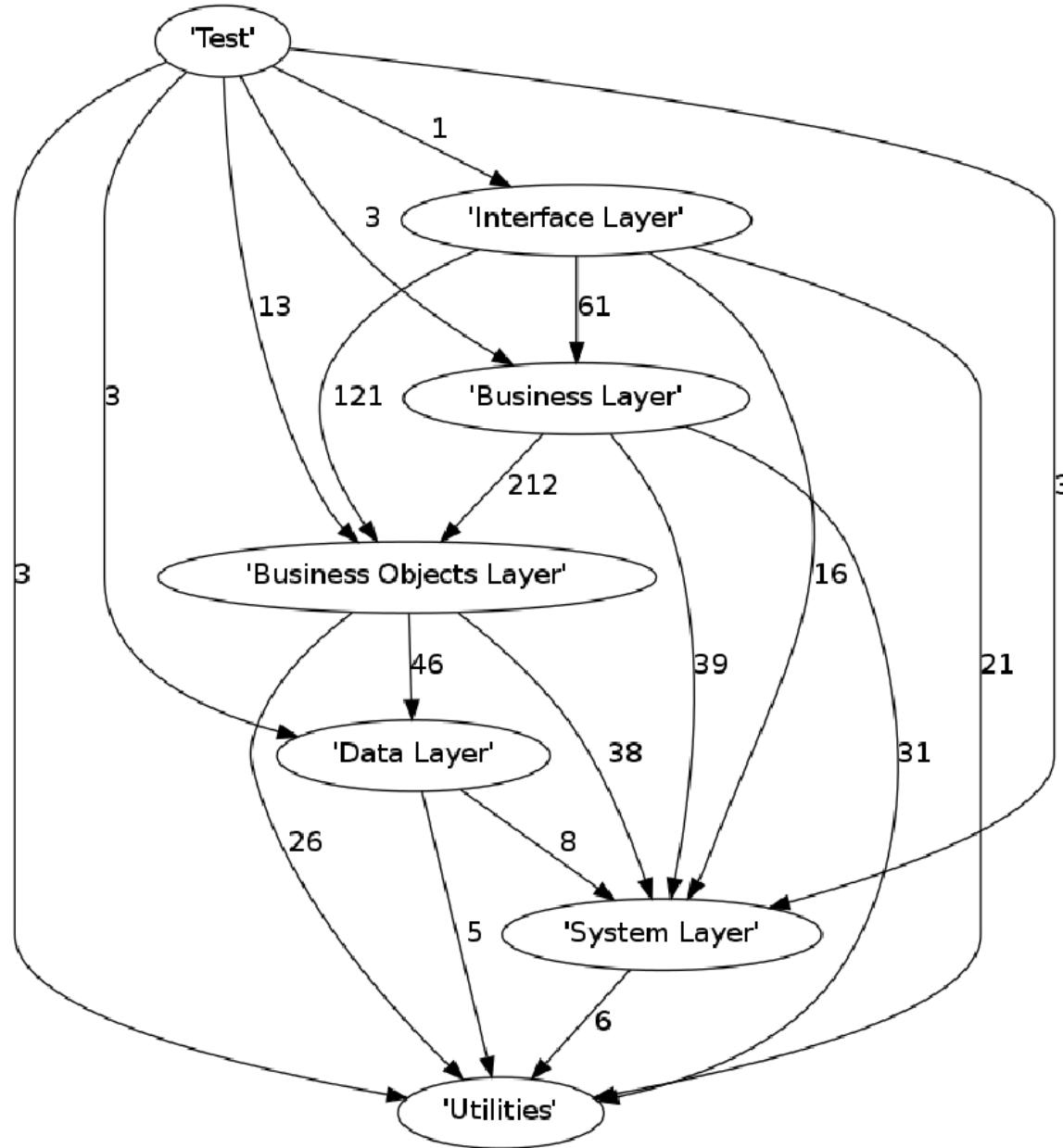
# Original application structure



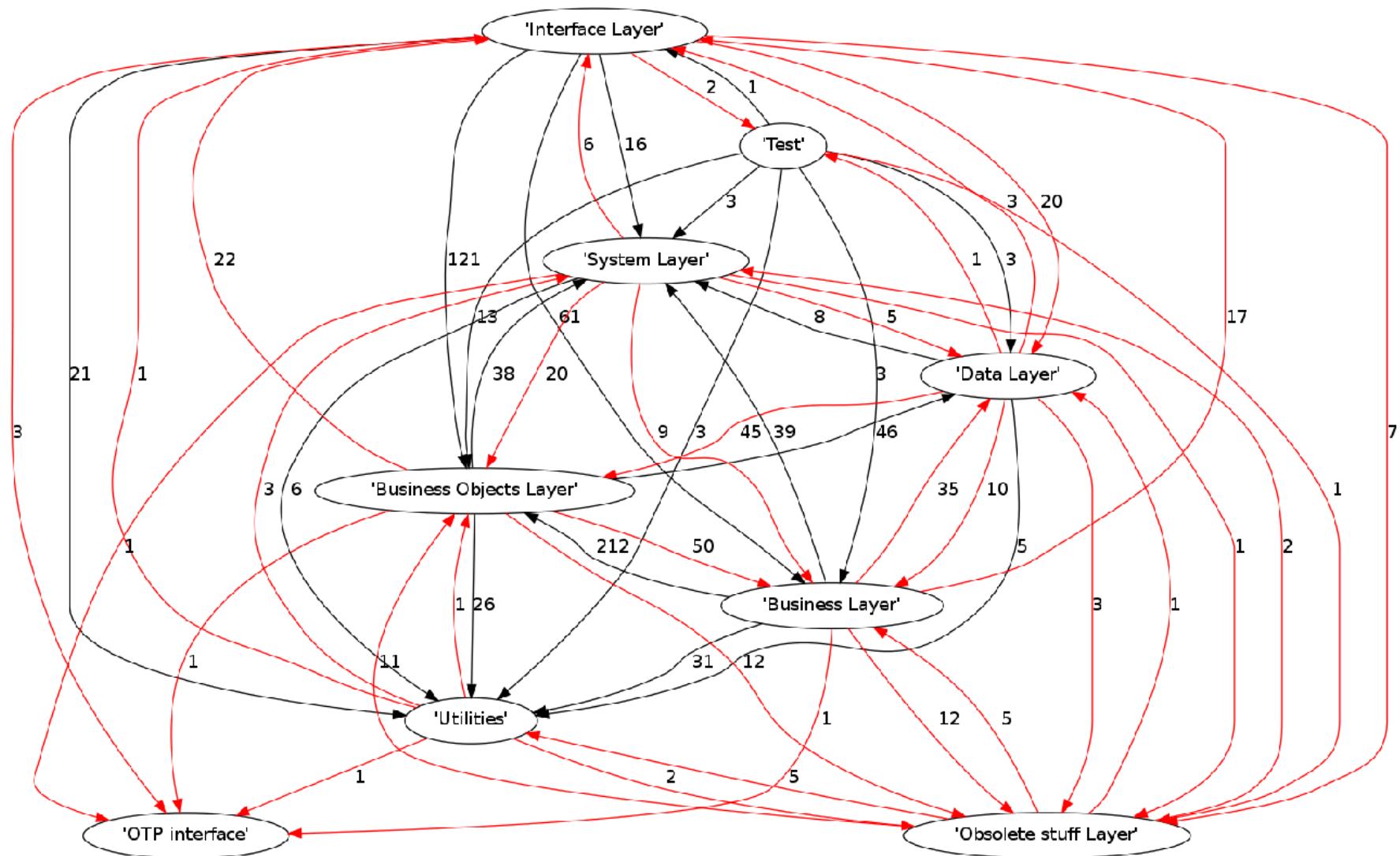
# New improved(?) application structure



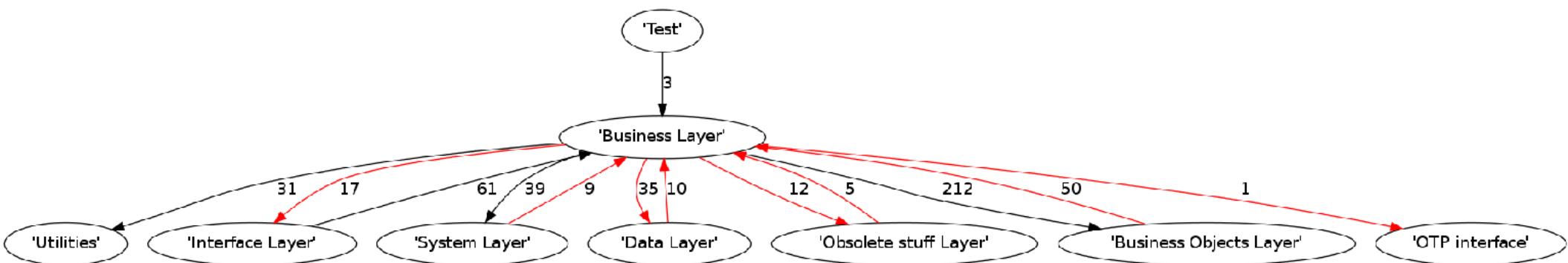
# Layers to the rescue



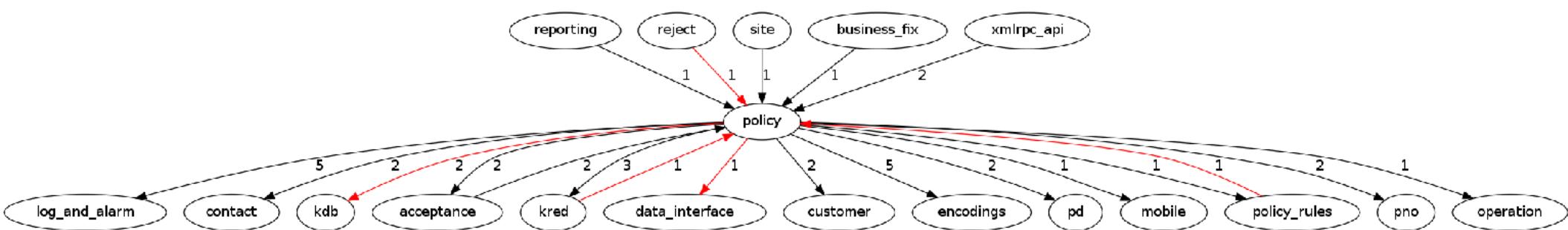
but some bad dependencies remain



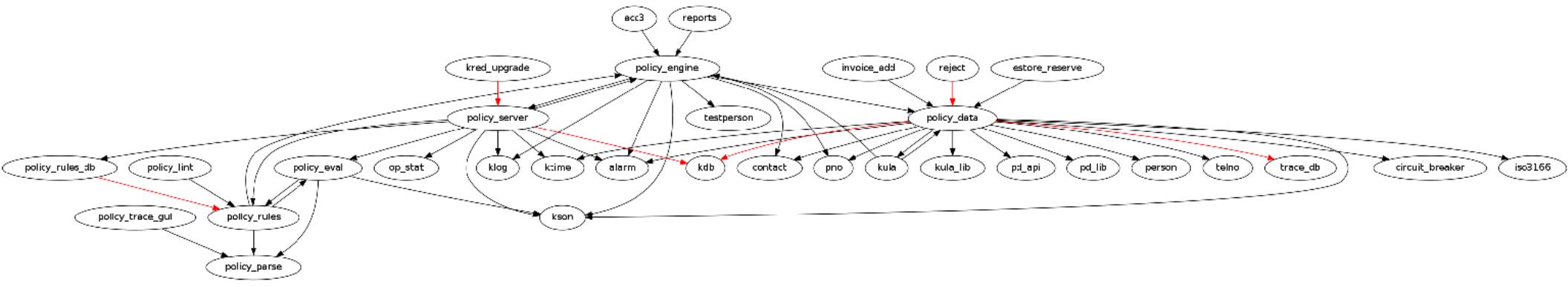
## Zoom in on Business Layer



and then to the policy application

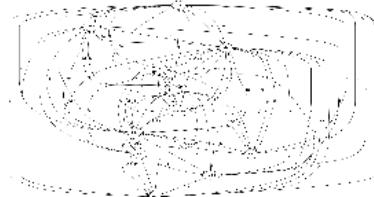


expand to see how the actual modules dependencies look like

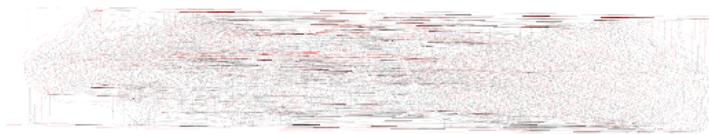


# The Graphs

Original application structure



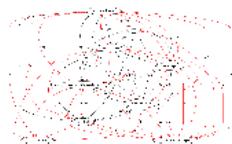
New improved(?) application structure



Layers to the rescue



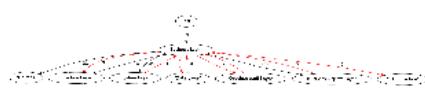
but some bad dependencies remain



and then to the policy application



Zoom in on Business Layer





Tobias Lindahl

# Code Management

## The Problem

- Organically grown code
- Active application structure
- No clear cuts in functionality

## Our solution

- Move existing modules to new applications.
- Group applications in clusters (layers).
- Fix dependencies and cluster dependencies.
- Refactor code to satisfy constraints.
- Profit!

## The Analysis

- Find dependencies between modules.
- Sort them from applications and layers
- Write them down in a file
- Make extremely cool graphs

## The Action

- Generate shell scripts for
- Creating new application structure
- Moving the code
- Patching incisor paths, Makefiles, etc

## The Graphs





Tobias Lindahl

## Code Management

The Problem  
The Solution  
The Analysis  
The Results  
The Metrics  
The Details

The Graphs

The Data

## Tools @ **klarna**



Simpler Safer More Fun

Businesses need  
differentiation  
and the right  
tools to succeed.  
That's what  
Klarna offers.

Think fast  
act slow  
and always  
innovate.



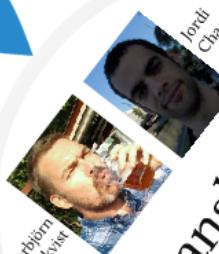
David  
Evans



Jia  
Wang

# Testing

Unit Testing  
Continuous Integration  
GUI Testing  
Acceptance Testing



Jordi  
Chacon  
Torbjörn  
Törnkvist

# Translation "polish"

The Problem  
The Solution  
Using GoTest  
Background  
About Peacock

The Problem

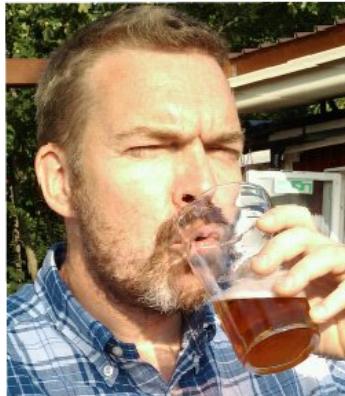
The Solution

Background

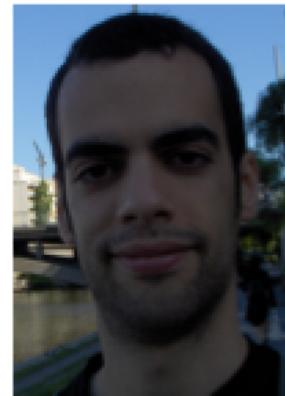
Torbjörn  
Törnkvist

Jordi  
Chacon

Torbjörn  
Törnvist



Jordi  
Chacon



# Translation "POlish"

Background

Using GetText

The Problem  
is causing increasing headache!

The Solution



About POlish

- POlish is a ui-layer developed by Klamo
- Web-CLI built on top of gettext to easily access PO-Files
- Provides a simple and friendly interface for translators
- Log in via OpenID
- Keep track of what is untranslated
- Allows translators to search for terms
- Ensures correctness of translations
- Started with a simple command
- Has been used for the last three months

DEMO TIME!  
<http://www.klamo.com/polish/>

New features  
coming... or not

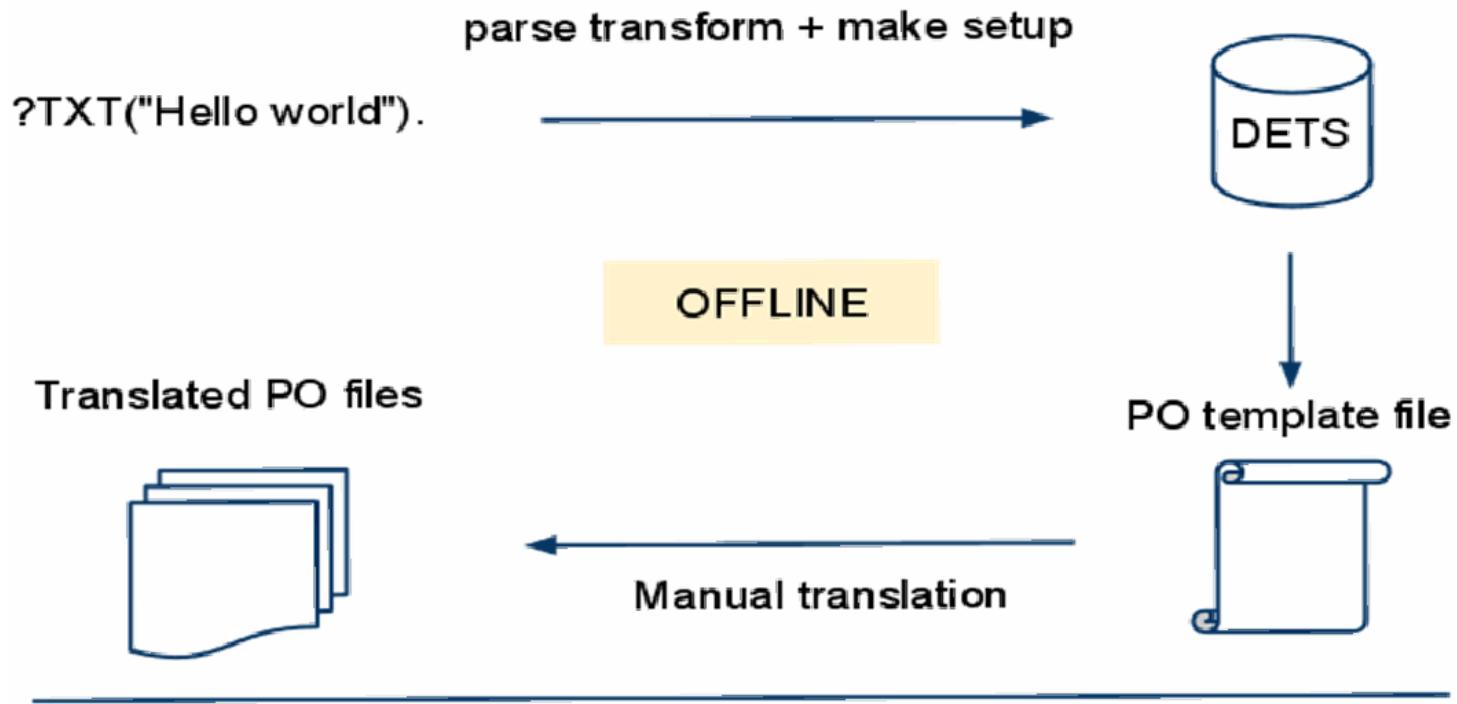
- REST Interface
- Translation memory
- Spell checker
- Online service
- git@github.com:cent/pnish.git

# Background

- Klarna produces ~50 different PDF documents
- Klarna is serving ~7000 Estores + our own customer care
- We have about ~5000 texts that need to be translated from Swedish to: Norwegian, Finnish, Danish, German, Dutch, English...
- On average, one person working 100% on text changes

# Using GetText

```
-include("gettext.hrl").
```

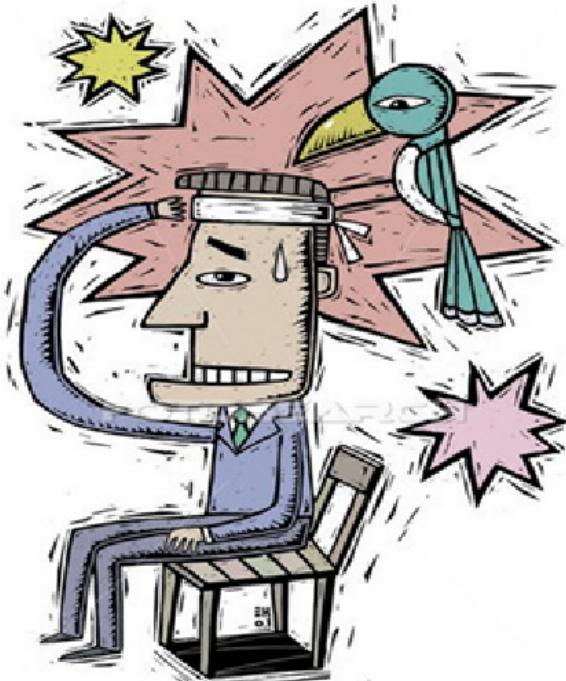


```
-define(?TXT(S), gettext:key2str(S, get(gettext_language))).
```

ONLINE

# The Problem

is causing increasing headache!



- manually editing PO files is a source of problems
- 25.000-line PO files emailed back and forth
- unfriendly for translators to work on these huge po files
- texts always ended up being taken live with only the Swedish version available
- our customers weren't really happy about that...

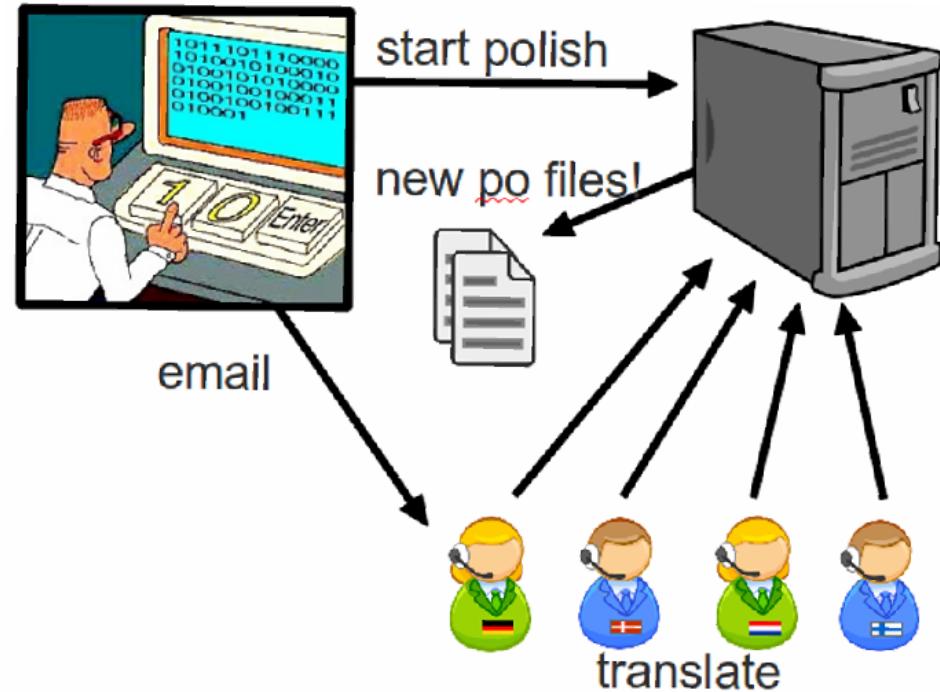
# The Solution

A developer is hacking in his/her branch.

Adds a ?TXT macro containing a new Swedish text.

It should be translated to all languages!

What should the developer do?



# About POlish

- POlish is a utility developed by Klarna
- Web-GUI built on top of gettext to easily access PO-files
- Provides a simple and friendly interface for translators
- Log in via OpenID
- Keeps track of what is untranslated
- Allows translators to search for texts
- Ensures correctness of translations
- Started with a simple commands
- Has been used for the last three months

# DEMO TIME!

<http://www.youtube.com/watch?v=UdhE2YOkBCU>

# New features coming... or not

- REST interface
- Translation memory
- Spell checker
- Online service

`git@github.com:etnt/polish.git`



Tobias Lindahl

## Code Management

The Problem  
The Solution  
The Analysis  
The Results  
The Metrics  
The Details

The Graphs

The Data

## Tools @ **klarna**



Simpler Safer More Fun

Businesses need  
differentiation  
and innovation  
to succeed.  
Klarna is  
driving  
innovation.

Think fast  
act slow  
and  
innovate  
constantly.



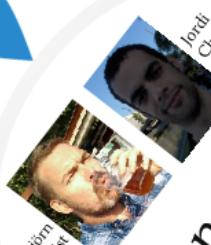
David  
Evans



Jia  
Wang

# Testing

Unit Testing  
Continuous Integration  
GUI Testing  
Acceptance Testing



Jordi  
Chacon

# Translation "polish"

The Problem  
The Solution  
Using GoTest  
Background  
About Peacock  
Demo Test



Torbjörn  
Törnkvist

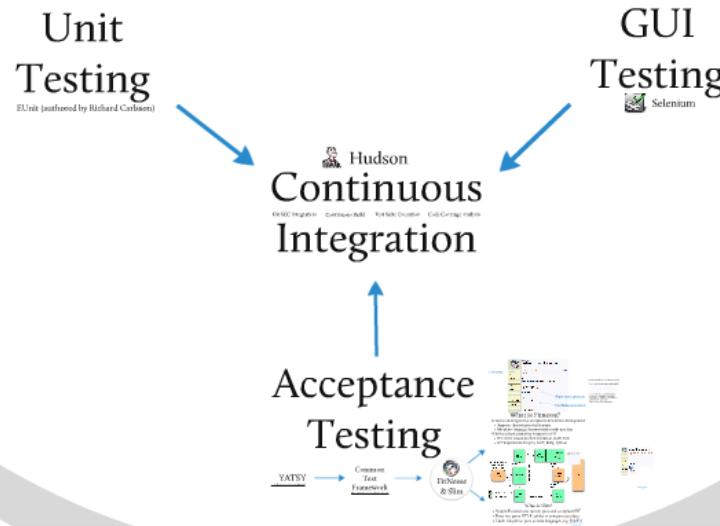
David  
Evans



Jia  
Wang



# Testing



# Unit Testing

EUnit (authored by Richard Carlsson)

# GUI Testing



Selenium

# Hudson Continuous Integration

Git SCC Integration   Continuous Build   Test Suite Execution   Code Coverage Analysis

# Acceptance Testing

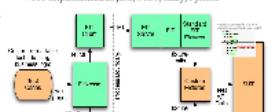
[YATSY](#)

Common  
Test  
Framework



What is FitNesse?  
Framework designed for acceptance test-driven development

- Support Specification by Example
- Mix plain-language documentation with test data
- Wiki-based test authoring wrapper for FIT
- FIT server automates test execution at API level
- FIT implemented in Java, .NET, Ruby, Python



What is Slim?  
Simple FitNesse test runner protocol to replace FIT

- Does not parse HTML tables or compare test data
- Much simpler to run than FitNesse on GlassFish





Hudson

# Continuous

Git SCC Integration    Continuous Build    Test Suite Execution    Code Coverage Analysis

# Integration



# Unit Testing

EUnit (authored by Richard Carlsson)

# GUI

# Testing



Selenium

# Unit Testing

EUnit (authored by Richard Carlsson)

# GUI Testing



Selenium

# Hudson Continuous Integration

Git SCC Integration   Continuous Build   Test Suite Execution   Code Coverage Analysis

# Acceptance Testing

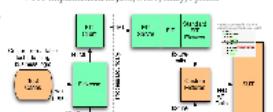
[YATSY](#)

Common  
Test  
Framework



What is FitNesse?  
Framework designed for acceptance test-driven development

- Supports Specification by Example
- Mix plain-language documentation with test data
- Wiki-based test authoring wrapper for FIT
- FIT server automates test execution at API level
- FIT implemented in Java, .NET, Ruby, Python



What is Slim?  
Simple FitNesse test runner protocol to replace FIT

- Does not parse HTML tables or compare test data

• Much simpler to run than FitNesse on GlassFish



# Acceptance

# Testing

YATSY

<http://code.google.com/p/yatsy/>



Common

Test

Framework

[http://www.erlang.org/doc/apps/common\\_test/](http://www.erlang.org/doc/apps/common_test/)



FitNesse  
& Slim





**FitNesse**  
**& Slim**



**Test Button** →

FrontPage.

# ErlangTriangleDemo [add child]

► *Click here for Technical Implementation details*      [Expand All](#) | [Collapse All](#)

I WANT A FUNCTION THAT RETURNS THE TYPE OF TRIANGLE PRODUCED GIVEN THE LENGTHS OF THREE SIDES (A, B AND C)

The valid types are:

- All sides equal: Equilateral
- Two sides equal: Isosceles
- No sides equal: Scalene

For example:

triangle_type_fixture			
a	b	c	type?
2	2	2	Equilateral
2	2	3	Isosceles
2	3	4	Scalene

Plain text (ignored)

Test Data (executed)

# What is Fitnesse?

Framework designed for acceptance test-driven development

- Supports 'Specification by Example'
- Mix plain-language documentation with test data

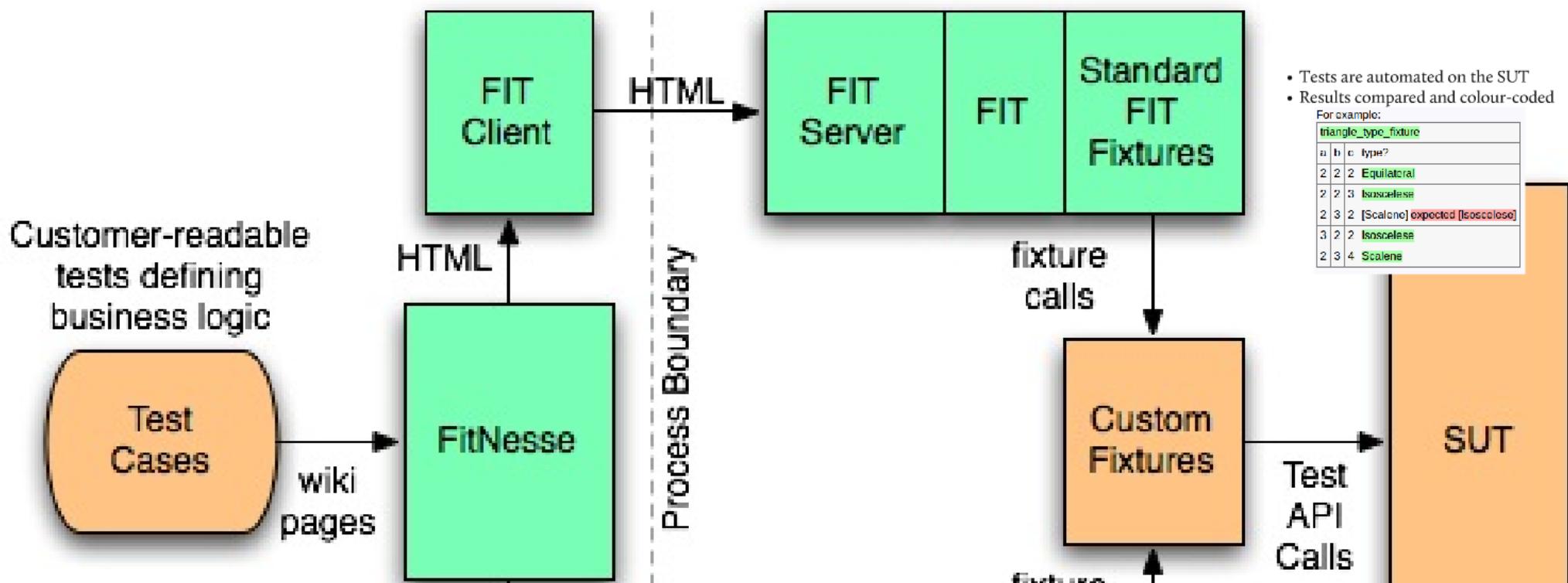
# What is Fitnesse?

Framework designed for acceptance test-driven development

- Supports 'Specification by Example'
- Mix plain-language documentation with test data

Wiki-based test authoring wrapper for FIT

- FIT server automates test execution at API level
- FIT implemented in Java, .NET, Ruby, Python



- Tests are automated on the SUT
- Results compared and colour-coded

For example:

triangle_type_fixture			
a	b	c	type?
2	2	2	Equilateral
2	2	3	Isosceles
2	3	2	[Scalene] expected [Isosceles]
3	2	2	Isosceles
2	3	4	Scalene

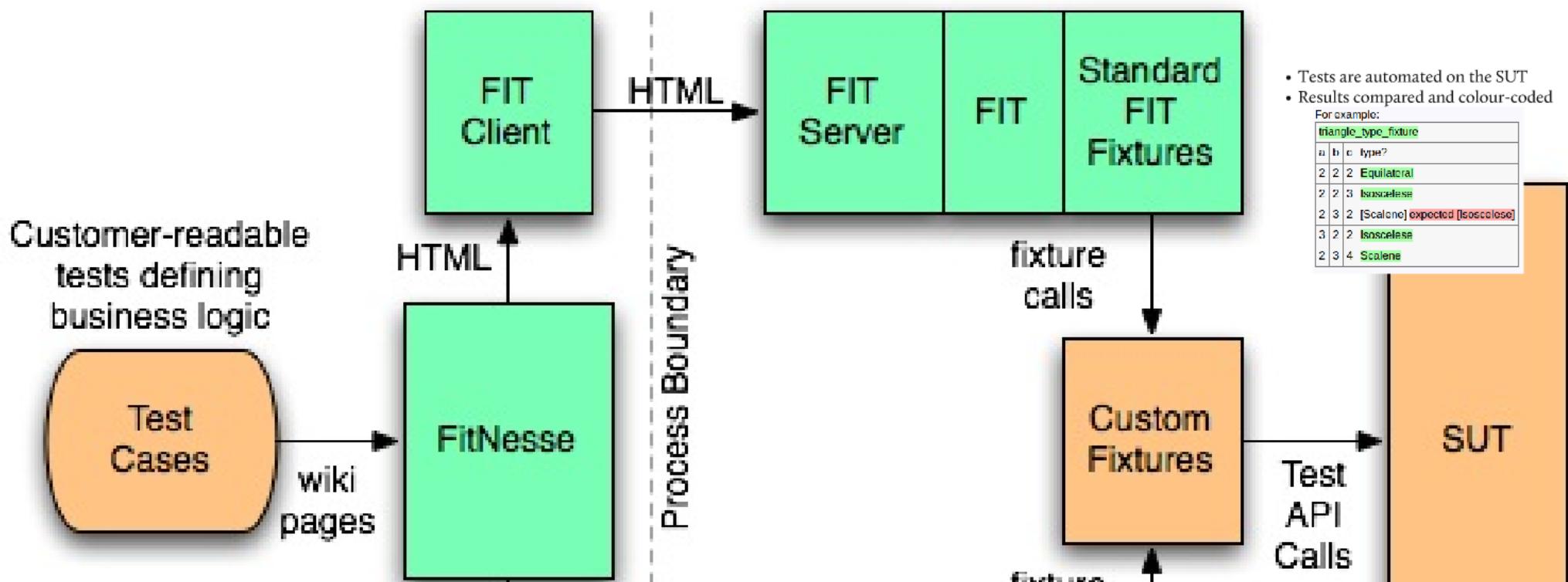
# What is Fitnesse?

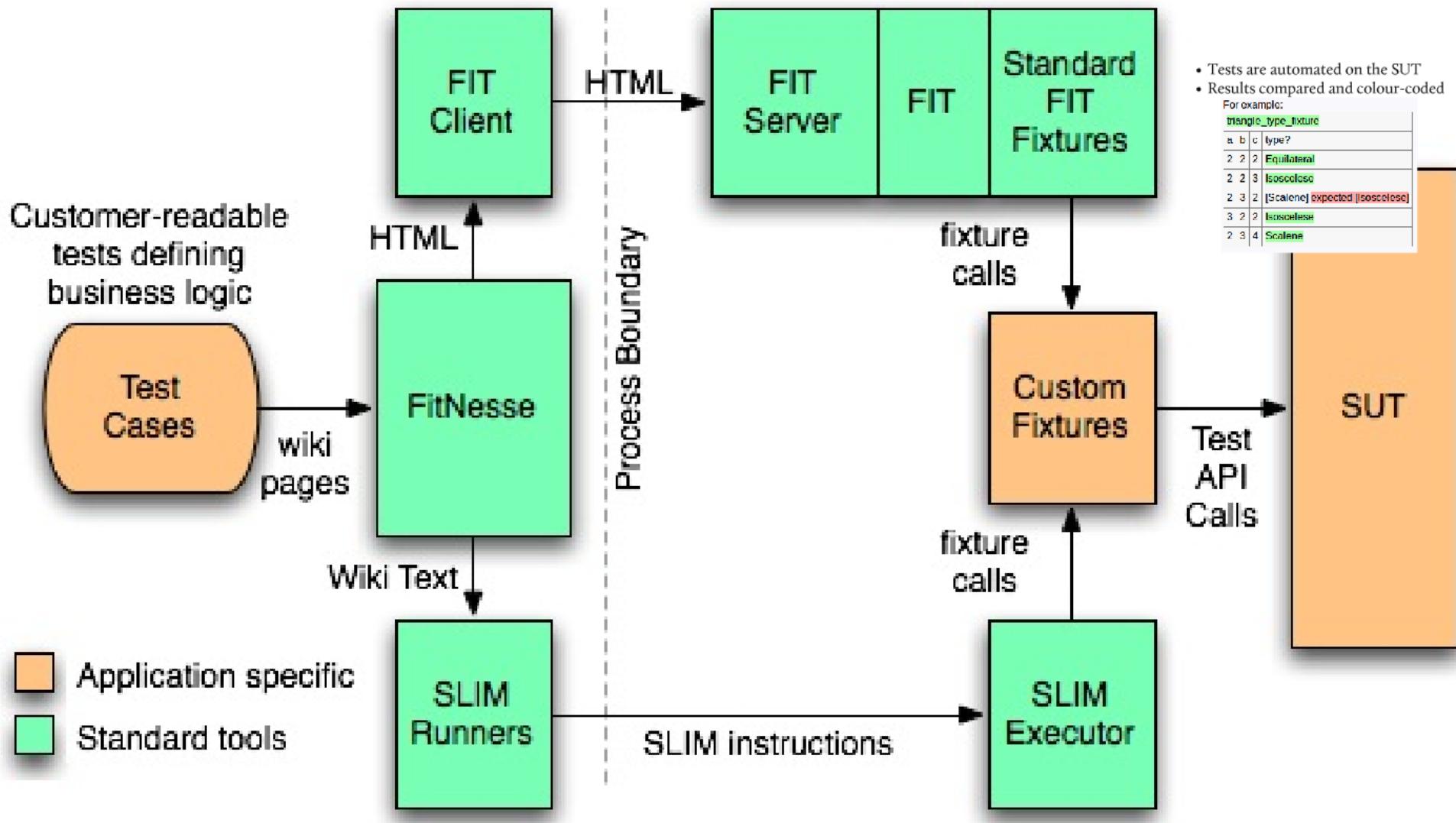
Framework designed for acceptance test-driven development

- Supports 'Specification by Example'
- Mix plain-language documentation with test data

Wiki-based test authoring wrapper for FIT

- FIT server automates test execution at API level
- FIT implemented in Java, .NET, Ruby, Python





- Tests are automated on the SUT
- Results compared and colour-coded

For example:		
<code>triangle_type_fixture</code>		
a	b	c
2	2	2
2	2	3
2	3	2
3	2	2
2	3	4
<code>type?</code>		
Equilateral	Isosceles	Scalene

## What is Slim?

- Simple Fitnesse test runner protocol to replace FIT
- Does not parse HTML tables or compare test data
- Much simpler to port to new languages, e.g. SlimErl

• Erlang implementation of Slim Executor  
• Authored by Antoine Choppin

"This is my first Erlang project, and I do not pretend to know either Fitnesse nor Erlang very well."

<http://code.noulei.com/n/climber/>

# .SlimeRl

- Erlang implementation of Slim Executor
- Authored by Antoine Choppin

"This is my first Erlang project, and I do not pretend to know either Fitnesse nor Erlang very well."

<http://code.google.com/p/slimerl/>



[FrontPage.](#)

# ErlangTriangleDemo

[add child]

Test

Edit

Properties

Refactor

Where Used

Search

Files

Versions

Recent Changes

User Guide

Test History

► Click here for Technical Implementation details

[Expand All](#) | [Collapse All](#)

I WANT A FUNCTION THAT RETURNS THE TYPE OF TRIANGLE PRODUCED  
GIVEN THE LENGTHS OF THREE SIDES (A, B AND C)

The valid types are:

All sides equal: Equilateral

Two sides equal: Isosceles

No sides equal: Scalene



For example:

triangle\_type\_fixture

a	b	c	type?
2	2	2	Equilateral
2	2	3	Isosceles
2	3	4	Scalene

```
emacs@hundhaj
File Edit Options Buffers Tools Erlang Help
%%-----  
%% This fixture code is generated by slim_generator  
%%-----  
-module(triangle_type_fixture).  
  
%% Export Callback Functions  
-export([ new/0  
        , execute/1]).  
%% Export setters and assert functions  
-export([ setA/2  
        , setB/2  
        , setC/2  
        , type/1  
        ]).  
 
```

Two sides equal: Isosceles

No sides equal: Scalene



For example:

triangle\_type\_fixture

a	b	c	type?
2	2	2	Equilateral
2	2	3	Isosceles
2	3	4	Scalene

emacs@hur

```
File Edit Options Buffers Tools Erlang Help
%%-----
%% This fixture code is generated by slim
%%-----
-module(triangle_type_fixture).

%% Export Callback Functions
-export([ new/0
        , execute/1 ]).

%% Export setters and assert functions
-export([ setA/2
        , setB/2
        , setC/2
        , type/1
      ]).

-record(triangle_type_fixture, { a
                                 , b
                                 , c
                               }).

%%-----
%% Callback functions for FitNesse
%%-----
www/1 ~ #+minerals +minerals fixtures/1
```

```
; -- /-
; setC/2
; type/1
] ).
```

```
-record(triangle_type_fixture, { a
                                  , b
                                  , c
                                } ) .
```

```
%%-----  
%% Callback functions for FitNesse  
%%-----
```

```
new() -> #triangle_type_fixture{}.
```

```
%% Setter Functions
```

```
setA(Rec, A) ->  
    Rec#triangle_type_fixture{a = A}.
```

```
setB(Rec, B) ->  
    Rec#triangle_type_fixture{b = B}.
```

```
setC(Rec, C) ->  
    Rec#triangle_type_fixture{c = C}.
```

```
execute(Record) -> Record. %% TODO do initialization
```

```
%%-----  
%% Assert Functions  
%%-----
```

```
type(#triangle_type_fixture{a = A,
```

```
%-----  
%% Callback functions for FitNesse  
%-----  
new() -> #triangle_type_fixture().  
  
%% Setter Functions  
setA(Rec, A) ->  
    Rec#triangle_type_fixture{a = A}.  
setB(Rec, B) ->  
    Rec#triangle_type_fixture{b = B}.  
setC(Rec, C) ->  
    Rec#triangle_type_fixture{c = C}.  
  
execute(Record) -> Record. %% TODO do initialization  
%-----  
%% Assert Functions  
%-----  
type(#triangle_type_fixture{a = A,  
                           b = B,  
                           c = C} = Rec) ->  
    %% TODO  
    R = triangle_type:type(A, B, C),  
    echoString(Rec, R).  
  
%-----  
%% Import some utiltv functions for slimerl
```

# Where to next with Fitnesse & Slim?

- Use more sophisticated features like variables

THIS SLIGHTLY MORE COMPLEX EXAMPLE CREATES A NEW PERSON, THEN RETRIEVES THEM

Note that in this test we need to refer to:  
► Included page: [CountryCodes \(edit\)](#)

[Expand All](#) | [Collapse All](#)

setup_person								
Nationality	Age	First Name	Last Name	Street	Zip Code	City	Country	pno?
2	30	Jia	Wang	Skt. Eriksgatan 112	54321	Stockholm	209	\$JiaPno

Now lookup this person and check that we stored the correct details and that we concatenate the names and address elements correctly.  
**Pacc:** By default, a new person will not have a Personal Account (pacc).  
**Birth Date:** The first 6 digits of the Pno determine the date of birth of the person.

get_person			
pno	has_pacc_account?	birth_year?	get_address?
\$JiaPno	false	1980	Skt. Eriksgatan 112 54321 Stockholm Sweden

THIS SLIGHTLY MORE COMPLEX EXAMPLE CREATES A NEW PERSON, THEN RETRIEVES THEM

Note that in this test we need to refer to:  
► Included page: [CountryCodes \(edit\)](#)

[Expand All](#) | [Collapse All](#)

setup_person								
Nationality	Age	First Name	Last Name	Street	Zip Code	City	Country	pno?
2	30	Jia	Wang	Skt. Eriksgatan 112	54321	Stockholm	209	\$JiaPno-> 800414 0920

Now lookup this person and check that we stored the correct details and that we concatenate the names and address elements correctly.  
**Pacc:** By default, a new person will not have a Personal Account (pacc).  
**Birth Date:** The first 6 digits of the Pno determine the date of birth of the person.

get_person			
pno	has_pacc_account?	birth_year?	get_address?
\$JiaPno-> 800414 0920	false	1980	Skt. Eriksgatan 112 54321 Stockholm Sweden

- Improve the use of concrete examples to support story specification descriptions
- Build up a body of 'living documentation' : self-checking business rules
- Open-source our fixture generators

## THIS SLIGHTLY MORE COMPLEX EXAMPLE CREATES A NEW PERSON, THEN RETRIEVES THEM

Note that in this test we need to refer to:

► Included page: [CountryCodes \(edit\)](#)

[Expand All](#) | [Collapse All](#)

### setup\_person

Nationality	Age	First Name	Last Name	Street	Zip Code	City	Country	pno?
2	30	Jia	Wang	Skt. Eriksgatan 112	54321	Stockholm	209	\$jiaPno=

Now lookup this person and check that we stored the correct details and that we concatenate the name and address elements correctly.

**Pacc:** By default, a new person will not have a Personal Account (pacc).

**Birth Date:** The first 6 digits of the Pno determine the date of birth of the person.

### get\_person

pno	has_pacc_account?	birth_year?	get_address?
\$jiaPno	false	1980	Skt. Eriksgatan 112 54321 Stockholm Sweden



## THIS SLIGHTLY MORE COMPLEX EXAMPLE CREATES A NEW PERSON, THEN RETRIEVES THEM

Note that in this test we need to refer to:

► [Included page: CountryCodes \(edit\)](#)

### setup\_person

Nationality	Age	First Name	Last Name	Street	Zip Code	City	Country	pno?
2	30	Jia	Wang	Skt. Eriksgatan 112	54321	Stockholm	209	\$jiaPno->[800414-0920]

Now lookup this person and check that we stored the correct details and that we concatenate the names and address elements correctly.

**Pacc:** By default, a new person will not have a Personal Account (pacc).

**Birth Date:** The first 6 digits of the Pno determine the date of birth of the person.

### get\_person

pno	has_pacc_account?	birth_year?	get_address?
\$jiaPno->[800414-0920]	false	1980	Skt. Eriksgatan 112 54321 Stockholm Sweden



# Where to next with Fitnesse & Slim?

- Use more sophisticated features like variables

THIS SLIGHTLY MORE COMPLEX EXAMPLE CREATES A NEW PERSON, THEN RETRIEVES THEM

Note that in this test we need to refer to:  
► Included page: [CountryCodes \(edit\)](#)

[Expand All](#) | [Collapse All](#)

setup_person								
Nationality	Age	First Name	Last Name	Street	Zip Code	City	Country	pno?
2	30	Jia	Wang	Skt. Eriksgatan 112	54321	Stockholm	209	\$JiaPno

Now lookup this person and check that we stored the correct details and that we concatenate the names and address elements correctly.  
**Pacc:** By default, a new person will not have a Personal Account (pacc).  
**Birth Date:** The first 6 digits of the Pno determine the date of birth of the person.

get_person			
pno	has_pacc_account?	birth_year?	get_address?
\$JiaPno	false	1980	Skt. Eriksgatan 112 54321 Stockholm Sweden

THIS SLIGHTLY MORE COMPLEX EXAMPLE CREATES A NEW PERSON, THEN RETRIEVES THEM

Note that in this test we need to refer to:  
► Included page: [CountryCodes \(edit\)](#)

[Expand All](#) | [Collapse All](#)

setup_person								
Nationality	Age	First Name	Last Name	Street	Zip Code	City	Country	pno?
2	30	Jia	Wang	Skt. Eriksgatan 112	54321	Stockholm	209	\$JiaPno-> 800414 0920

Now lookup this person and check that we stored the correct details and that we concatenate the names and address elements correctly.  
**Pacc:** By default, a new person will not have a Personal Account (pacc).  
**Birth Date:** The first 6 digits of the Pno determine the date of birth of the person.

get_person			
pno	has_pacc_account?	birth_year?	get_address?
\$JiaPno-> 800414 0920	false	1980	Skt. Eriksgatan 112 54321 Stockholm Sweden

- Improve the use of concrete examples to support story specification descriptions
- Build up a body of 'living documentation' : self-checking business rules
- Open-source our fixture generators

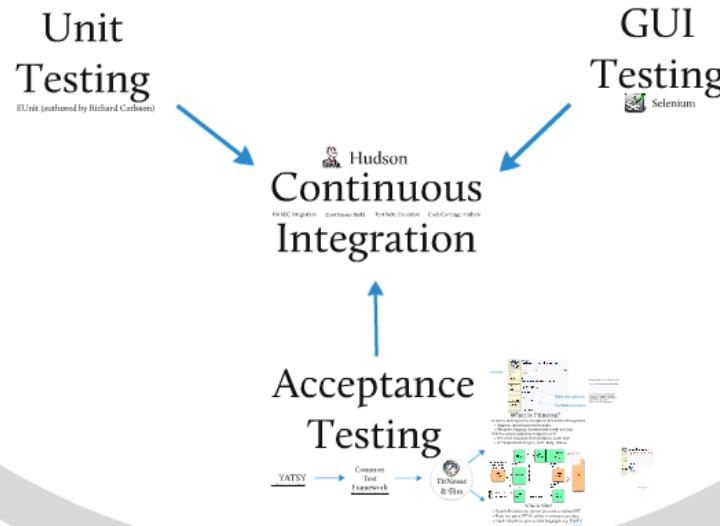
David  
Evans



Jia  
Wang



# Testing



Evolving our toolset is a natural extension of evolving our business.

All our tool choices are open-source.  
All tools we create will also be open-sourced.

# Thanks!

**POlish**: <https://github.com/etnt/polish>  
<http://www.youtube.com/watch?v=UdhE2YOkBCU>

**GetText**: <https://github.com/etnt/gettext>

**SlimErl**: <http://code.google.com/p/slimerl>

**Fitness+Slim**: <http://fitness.org>

POlish demo will be running  
during the break