

Quality Cruising

Making Java Work for Erlang

Erik (Happi) Stenman

Introduction

Introduction

I will talk about **automated testing**, and how to make **Java** do that work for you.

Introduction

I will talk about **automated testing**, and how to make **Java** do that work for you.

Introduction

I will talk about **automated testing**, and how to make **Java** do that work for you.

But first a short recap for those of you who weren't here last year or don't remember my talk about [Kreditor](#).

Kreditor Europe AB

- Founded in December 2004.

Kreditor Europe AB

- Founded in December 2004.
- Bring trust to Internet shopping, by providing old style billing through hi-tech solutions.

Kreditor Europe AB

- Founded in December 2004.
- Bring trust to Internet shopping, by providing old style billing through hi-tech solutions.
- More than 1700 Internet shops connected.

Kreditor Europe AB

- Founded in December 2004.
- Bring trust to Internet shopping, by providing old style billing through hi-tech solutions.
- More than 1700 Internet shops connected.
- The company vision:

Kreditor Europe AB

- Founded in December 2004.
- Bring trust to Internet shopping, by providing old style billing through hi-tech solutions.
- More than 1700 Internet shops connected.
- The company vision:
 - “Be the coolest company in Sweden.”

The Implementation



The system is a combination of **fault tolerant servers**, lots of **protocol- and glue-code**, **persistent storage**, and a web-UI.

Some implementation details

- The system is built in-house from scratch using **LYME** (**L**inux, **Y**aws, **M**nesia, and **E**rlang).
- We have a distributed system with multiple servers to provide a fault tolerant, high availability solution.
- We aim for 5 nines availability, in a setting where we introduce new features in the system every week (almost every day).
- The problem fits **Erlang** really well.

Process

Process

- We have have an informal agile process.

Process

- We have have an informal agile process.
- We have very short time to market, for simple changes the time from idea to finished integration can be less than one hour.

Process

- We have have an informal agile process.
- We have very short time to market, for simple changes the time from idea to finished integration can be less than one hour.
- It is crucial for us to have an automated comprehensive test suite.

Process

- We have have an informal agile process.
- We have very short time to market, for simple changes the time from idea to finished integration can be less than one hour.
- It is crucial for us to have an automated comprehensive test suite.
 - With a framework that works.

Process

- We have have an informal agile process.
- We have very short time to market, for simple changes the time from idea to finished integration can be less than one hour.
- It is crucial for us to have an automated comprehensive test suite.
 - With a framework that works.
 - Which is **used**.

Process

- We have have an informal agile process.
- We have very short time to market, for simple changes the time from idea to finished integration can be less than one hour.
- It is crucial for us to have an automated comprehensive test suite.
 - With a framework that works.
 - Which is **used**.
 - **Always**.

Process

- We have have an informal agile process.
- We have very short time to market, for simple changes the time from idea to finished integration can be less than one hour.
- It is crucial for us to have an automated comprehensive test suite.
 - With a framework that works.
 - Which is **used**.
 - **Always**.
- Enters Yatsy and CruiseControl.

Yatsy

- Yet Another Test Server – Yaws compatible
(Yatsy is Swedish for Yatzee – testing is a bit like a dice game.)

Yatsy

- Yet Another Test Server – Yaws compatible
(Yatsy is Swedish for Yatzee – testing is a bit like a dice game.)
- Why (yet) another test server?

Yatsy

- Yet Another Test Server – Yaws compatible
(Yatsy is Swedish for Yatzee – testing is a bit like a dice game.)
- Why (yet) another test server?
 - The “released” OTP-test server is from 2004.

Yatsy

- Yet Another Test Server – Yaws compatible
(Yatsy is Swedish for Yatzee – testing is a bit like a dice game.)
- Why (yet) another test server?
 - The “released” OTP-test server is from 2004.
 - It isn't really open source.

Yatsy

- Yet Another Test Server – Yaws compatible
(Yatsy is Swedish for Yatzee – testing is a bit like a dice game.)
- Why (yet) another test server?
 - The “released” OTP-test server is from 2004.
 - It isn't really open source.
 - We just couldn't get it to work.

Yatsy

- Yet Another Test Server – Yaws compatible
(Yatsy is Swedish for Yatzee – testing is a bit like a dice game.)
- Why (yet) another test server?
 - The “released” OTP-test server is from 2004.
 - It isn't really open source.
 - We just couldn't get it to work.
- First version hacked together over a weekend by Tobbe.

Yatsy

- Yet Another Test Server – Yaws compatible
(Yatsy is Swedish for Yatzee – testing is a bit like a dice game.)
- Why (yet) another test server?
 - The “released” OTP-test server is from 2004.
 - It isn't really open source.
 - We just couldn't get it to work.
- First version hacked together over a weekend by Tobbe.
 - Released as open source:
<http://code.google.com/p/yatsy/>

Yatsy - example

```
-module(example_SUITE).  
-export([all/1, init_per_suite/1, fin_per_suite/1, init_per_testcase/2,  
        fin_per_testcase/2, simple/1]).  
-include("yatsy.hrl").  
  
all(doc) -> ["Test cases for example."];  
all(suite) -> [simple].  
  
init_per_suite(Config) when list(Config) -> Config.  
fin_per_suite(_Config) -> ok.  
init_per_testcase(_TestCase, Config) when atom(_TestCase), list(Config) -> Config.  
fin_per_testcase(_TestCase, _Config) -> ok.
```

```
simple(doc) ->  
    ["Check that we can get an new example."];  
simple(Config) when is_list(Config) ->  
    [] = example:new(),  
    ok.
```

CruiseControl (CC)

- **CruiseControl** is a framework for *continuous integration*.

CruiseControl (CC)

- **CruiseControl** is a framework for *continuous integration*.
- It is **open source**.

CruiseControl (CC)

- **CruiseControl** is a framework for *continuous integration*.
- It is **open source**.
- It is written in **Java**.

CruiseControl (CC)

- **CruiseControl** is a framework for *continuous integration*.
- It is **open source**.
- It is written in **Java**.
- It automatically

CruiseControl (CC)

- **CruiseControl** is a framework for *continuous integration*.
- It is **open source**.
- It is written in **Java**.
- It automatically
 - checks out the latest version from a repository,

CruiseControl (CC)

- **CruiseControl** is a framework for *continuous integration*.
- It is **open source**.
- It is written in **Java**.
- It automatically
 - checks out the latest version from a repository,
 - does a build,

CruiseControl (CC)

- **CruiseControl** is a framework for *continuous integration*.
- It is **open source**.
- It is written in **Java**.
- It automatically
 - checks out the latest version from a repository,
 - does a build,
 - runs all tests,

CruiseControl (CC)

- **CruiseControl** is a framework for *continuous integration*.
- It is **open source**.
- It is written in **Java**.
- It automatically
 - checks out the latest version from a repository,
 - does a build,
 - runs all tests,
 - and it measures any metric you have automated.

CruiseControl (CC)

- **CruiseControl** is a framework for *continuous integration*.
- It is **open source**.
- It is written in **Java**.
- It automatically
 - checks out the latest version from a repository,
 - does a build,
 - runs all tests,
 - and it measures any metric you have automated.
 - As soon as **anyone** checks **anything** in.

CC – Components

CC consists of:

CC – Components

CC consists of:

- A set of plugins

CC – Components

CC consists of:

- A set of plugins
 - Version control pollers (e.g. `svn st -u`)

CC – Components

CC consists of:

- A set of plugins
 - Version control pollers (e.g. `svn st -u`)
 - Compile and test systems (e.g. ant or make)

CC – Components

CC consists of:

- A set of plugins
 - Version control pollers (e.g. `svn st -u`)
 - Compile and test systems (e.g. ant or make)
 - Publishers (e.g. web site, email, rss, irc)

CC – Components

CC consists of:

- A set of plugins
 - Version control pollers (e.g. `svn st -u`)
 - Compile and test systems (e.g. ant or make)
 - Publishers (e.g. web site, email, rss, irc)
- A build queue

CC – Components

CC consists of:

- A set of plugins
 - Version control pollers (e.g. `svn st -u`)
 - Compile and test systems (e.g. ant or make)
 - Publishers (e.g. web site, email, rss, irc)
- A build queue
- An admin GUI

CC – Components

CC consists of:

- A set of plugins
 - Version control pollers (e.g. `svn st -u`)
 - Compile and test systems (e.g. ant or make)
 - Publishers (e.g. web site, email, rss, irc)
- A build queue
- An admin GUI
- Lots of glue code

The CC loop

- Poll for event
 - A version control update
 - A time has come
- Checkout, build, and run tests
- Gather results
- Publish results

Our CC Setup

- A set of projects
 - commit-[branch], runs on every commit
 - nightly-[branch], runs each night
- CC starts a script
 - svn update
 - make
 - run yatsy
- Reporting
 - Yatsy reports test results through an xml file
 - Reverse engineered from JUnit
 - CC sends email to responsible developer in case of error

Integrating Cover with Yatsy

- There is an application called **Cover** in **Erlang/OTP** which can give you a line by line coverage analysis of a module.
- We have added a module `yatsy_cover` to **Yatsy** which will let you run cover on a test suite.
- If you run **yatsy** with the “cover flag” all beam files are “cover compiled” before the test suite is run.
- **Yatsy** produces a source code file annotated with execution count.

CC-Demo

- Kenneth Lundin*
16.45 Non-Destructive Arrays
Richard Carlsson and Dan Gudmundsson
17.00 How to program efficiently with Binaries and Bit Strings
Per Gustafsson
17.30 Bus transport to the **ErLounge**

There is no conference fee but registration is necessary, however **the EUC is oversubscribed and we are unable to receive further registrations.**

The EUC takes place in Ericsson's conference centre in Älvsjö. Please see directions [here](#).

[Hotell Älvsjö](#) offers rebate rates from November 7 to November 9 for EUC participants. Don't forget to mention "Erlang".

Some [hints](#) from last year's EUC for visitors to Stockholm. We will organise a small sight-seeing tour around Stockholm on the evening of November 7 for foreign visitors.

An Obfuscated Programming Competition will be held in conjunction with the EUC where the goal is to write the most obfuscated Erlang program. More information is available [here](#) and the Jury consists of Joe Armstrong, Richard Carlsson and Jan Henry Nystrom.

The EUC will end traditionally by a kindly sponsored **ErLounge**.

Support our sponsors, they support us!



corelatus

KREDITOR

QuviQ

Sjöland & Thyselius

ERICSSON



Training and Consulting

synapse mobile networks s.a.

Mobile Arts

Marcus Arendt Aktiebolag



Conclusion

- CC gives you *immediate* feedback on the test status of all your branches all the time.

Conclusion

- **CC** gives you *immediate* feedback on the test status of all your branches all the time.
- **Our contribution:**

Conclusion

- **CC** gives you *immediate* feedback on the test status of all your branches all the time.
- **Our contribution:**
 - Open sourced test suite – [Yatsy](#).

Conclusion

- **CC** gives you *immediate* feedback on the test status of all your branches all the time.
- **Our contribution:**
 - Open sourced test suite – [Yatsy](#).
 - Nice simple integration between **CC** and [Erlang](#), including [cover](#) and [dialyser](#), and **EUnit**.

Conclusion

- **CC** gives you *immediate* feedback on the test status of all your branches all the time.
- **Our contribution:**
 - Open sourced test suite – [Yatsy](#).
 - Nice simple integration between **CC** and [Erlang](#), including [cover](#) and [dialyser](#), and **EUnit**.
 - We let [Java](#) do some work for the [Erlang](#) developer.

Conclusion

- CC gives you *immediate* feedback on the test status of all your branches all the time.
- **Our contribution:**
 - Open sourced test suite – [Yatsy](#).
 - Nice simple integration between CC and [Erlang](#), including [cover](#) and [dialyser](#), and EUnit.
 - We let [Java](#) do some work for the [Erlang](#) developer.
- CC is just as valuable as a VCS.

Conclusion

- **CC** gives you *immediate* feedback on the test status of all your branches all the time.
- **Our contribution:**
 - Open sourced test suite – **Yatsy**.
 - Nice simple integration between **CC** and **Erlang**, including **cover** and **dialyser**, and **EUnit**.
 - We let **Java** do some work for the **Erlang** developer.
- **CC** is just as valuable as a **VCS**.
- If you are not using **CC** for your projects, start using it **NOW**.