

# Experiences of Using, Teaching and Supporting Erlang/OTP

Commercial Users of Functional Programming  
Tallinn, Estonia, September 24, 2005



Erlang Training and Consulting Ltd  
[www.erlang-consulting.com](http://www.erlang-consulting.com)

Francesco Cesarini  
[francesco@erlang-consulting.com](mailto:francesco@erlang-consulting.com)

---

## Contents

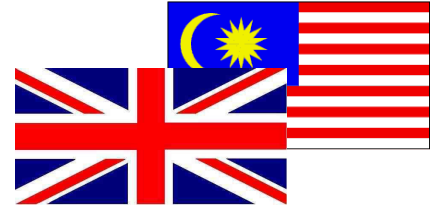
---

- Who Are We?
- Why Choose Erlang/OTP?
- Who Chooses Erlang?
- Developing
- Testing
- Supporting
- The Future



## Who are we?

- **Erlang Group**
  - Erlang Training and Consulting Ltd
  - Erlang Systems Ltd
  - Erlang Systems Sdn Bhd & Erlang Protocols Sdn Bhd
- **Global Provider of Erlang/OTP Services**
  - Training
  - Short-term Consultancies
  - Contracting and Recruitment
  - Product Development & Joint Ventures
  - Support and Hosting of Services
- **Goal: A one-stop shop for all your Erlang needs**

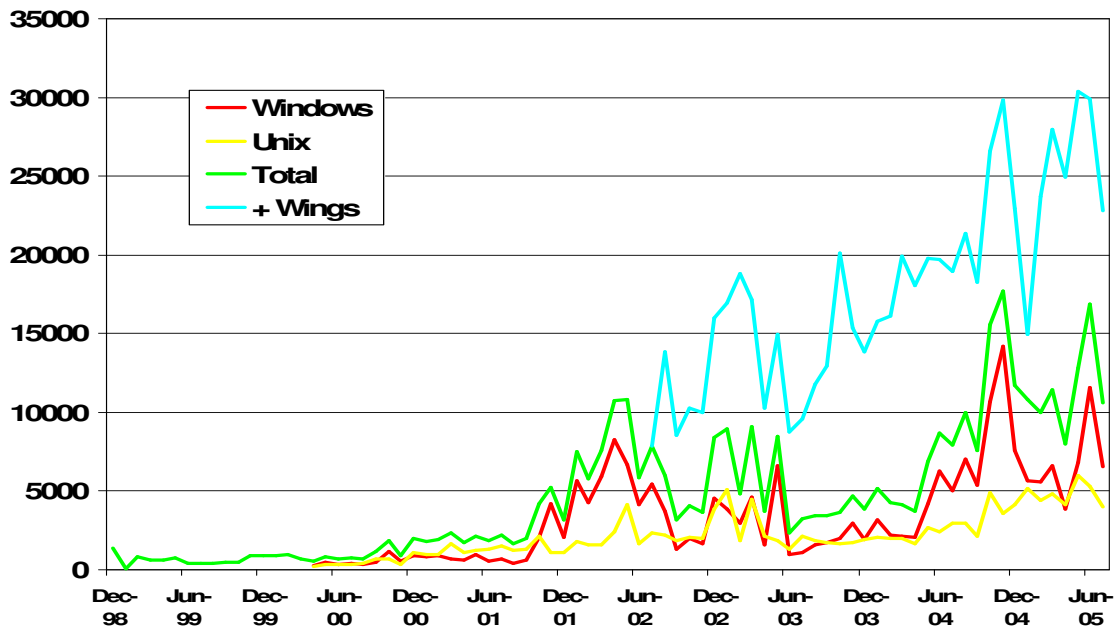


## Who are we: Some of our clients...



- Ericsson (Worldwide)
- T-Mobile (UK), WAP, SMS, IN services
- Teba Bank (South Africa), Commercial Banking
- Vail Systems (USA), Internet Enhanced Telephony
- IT University (Sweden)
- IdealX (France), Open Source Software Solutions

## Who are we: Some of our clients...



## Why Choose Erlang/OTP?

- **Developed by the CS Lab at Ericsson in the late 80s**
  - *After three years of prototyping and iterations, you would expect them to get it right!*
- **Four to ten times less code than C or Java**
- **Has achieved Nine Nines availability in live networks**
- **Released as Open Source in December 1998**

## Why Choose Erlang/OTP?

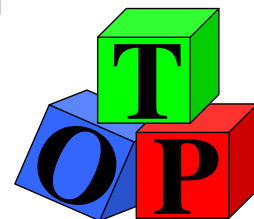
What if Skype had been implemented in Erlang?

- Declarative, with a high abstraction level
- Handles massive concurrency & message passing
- Distribution in heterogeneous networks
- Simple but powerful error handling mechanisms
- Hot code loading
- OS / Hardware independent

## Why Choose Erlang/OTP?

What is the Open Telecom/Transaction Platform?

- Erlang is just a programming language
- For large scale software design you need
  - Ready built applications and libraries
  - Design principles
  - Methodology
  - Tools
- These issues were addressed in 1993
  - Merged with Erlang Development in 1995 to form OTP



## Who Chooses Erlang?

No budget or marketing to spread the language means users come across Erlang through other channels...

- Word of Mouth
- Surfing the Web
- Open Source Applications
- Universities
- Ex-Ericsson employees / Companies using Erlang
- Seeds (and bombs) in comp.lang.functional

## Who Chooses Erlang?

“ Given the insistence of the members of the team to use the best language for specific purpose, we have code that is written in

- Prolog (enumerating device drivers, goal seek)
- LISP (classical AI stuff, inference)
- Erlang (server, communication and distributedness)

The management is worried about our choice of three language and has strongly suggested a relook at this strategy. The External Consultant (Subject Matter Expert) has suggested that... the long term success and reach of the product will be based on complete implementation in C++...”

An anonymous post on the Erlang mailing list August 25, 2005

## Who Chooses Erlang?

*It's dangerous to be right when the decision makers are wrong.*

Voltaire

- Someone championing Erlang is rarely enough
- You need management on your side
- You need marketing on your side
- You need people who can argue with “Subject Matter Experts”

## Who Chooses Erlang?

*In a company where management is positive towards Erlang, not only are the staff positive, but they also want to work with it!*

Lennart Öhman

- Open minded technical decision makers
  - Too few of them around!
- Marketing when approached on the right level
- People already in trouble with nothing to lose
- People “ordered” to use Erlang
- People using the right tools in the correct domain

## Developing

- **1995 PC Week Study of software projects shows that**
  - 16% successful
  - 53% operational (but less than successful)
  - 31% cancelled
- **Butler Group 1997 on large software projects states**
  - That 5 out of 6 large projects fail
  - In >\$300M companies, 9 out of 10 large projects fail
- **Erlang is no exception, disaster projects happen**
  - Blame has never been put on Erlang!

## Developing

*It is all about technology. The rest is just about communication between people.*

Anonymous

- **Most companies now use agile methodologies**
- **One should always start with a prototype**
  - Make mistakes on a small scale, not in a production project
  - Prove that your ideas work, don't just assume it
- **Never underestimate the need of training**
- **Set aside time for code reviews, refactoring, stress tests and benchmarking**

# Developing

- **How do people get away with choosing Erlang?**
  - *Provide Solutions to problems, not a tools to solve a task*
  - *Say it is a prototype, get it in production and wait for the rewrite in Java + to fail*
  - *Success is no guarantee if management is not behind you*
- **Do they benefit from its full potential?**
  - *Erlang is easy to learn, OTP is not*
  - *Software upgrade/release handling is not easy*
- **Professional Services**
  - *Code reviews result in a 40-50% reduction in code*
  - *Code reviews result in 100 - 500% increase in performance*
  - *Experienced people and prototypes result in less design errors*

# Testing

“No documents or instructions on how to install the system, Gating Issue”

Real World TR

- **Until 2001, lack of adequate training for test**
  - *Outcome still is very organisation dependent*
  - *Have had to develop two sets of exercises*
- **Lack of documentation**
  - *Designers will read the code, testers won't*
- **Lack of methodology**
  - *Organisations used existing “non-Erlang” processes*
- **Geographically spread organisations**

# Testing

- **Iterative design and test works best**
  - *Each release includes new functionality and bug fixes*
- **Test driven development is a good match**
  - *Erlang Group uses it for in-house projects*
- **Erlang test server**
  - *Success story with the OTP and the AXD301*
  - *Heavy when used in small projects*
- **Ongoing research**
  - *Testers rarely follow research trends and results*
  - *Must be introduced by management*
  - *Long live the Dialyzer!*
  - *Need to make the gap between research and testing smaller*

# Supporting



Since cut-over of the first nodes in BT's network in January 2002, only one minor fault has occurred, resulting in 99,9999999% availability.

*"As a matter of fact, the network performance has been so reliable that there is almost a risk that our field engineers do not learn maintenance skills"*  
says Bert Nilsson, Director  
NGS-Programs Ericsson

Ericsson Contact, Issue 19 2002



## Supporting

Not learning maintenance skills is not a risk, but a fact!

- **Until 2001, there was no specialised training**
    - *With AXE10 switches, engineers got 6 months training*
    - *Today, they are lucky if they get a few weeks*
  - **Small dev. teams vs. large support organisations**
  - **Not necessarily software related issues**
    - *Hardware problems*
    - *Networking problems*
    - *Configuration problems*
  - **Erlang related outages**
    - *Too often directed back to third line support / development*
    - *Much more can (and should) be done in pre-emptive support*
- 

## The Future

Ericsson does not know how to market programming languages!

- **Erlang Foundation**
    - *Nothing is happening*
    - *Running it on a volunteer basis is not enough*
    - *Needs to be financially driven and funded*
  - **Distribution**
    - *Sell packaged CDs, printed manuals*
    - *Online Branded Sales of T-Shirts, Coffee Mugs, Pins, Bags, etc*
    - *Partner up with “The Big Boys”*
-

# The Future

Ericsson has no interest in marketing Erlang...

- **Erlang sites must be professionally maintained**

- *Kept up to date*
- *Look and feel*
- *Availability*
- *Usability*
- *Cannot rely on volunteers alone*

- **Branding**

- *Professional marketing material*
- *Presence at Conferences / Trade Shows*
- *Target right people at right level*



# The Future

- **Universities**

- *Channel funds in lectures and projects*
- *Summer School of Functional Programming*
- *Pay for Guest Lectures*

- **Erlang Support Licenses**

- *Create a model to generate revenue to fund critical key elements to boost Erlang usage*
- *Revenue generated will be ploughed back into the community*
- *Support is the only way to gain mainstream acceptance*

- **Our Goal: Expand Erlang Community**

- *We can not rely on organic growth alone*

# Contents

---

- **Who Are We?**
- **Why Choose Erlang/OTP?**
- **Who Chooses Erlang?**
- **Developing**
- **Testing**
- **Supporting**
- **The Future**