







1) Which editor do you use for Erlang development?

		Percentage	Responses
Eclipse		11.9	25
Emacs		49.7	104
NetBeans		1.9	4
SciTe		0.9	2
Vim		18.7	39
Other		16.7	35















2) Do you use special Erlang plugin/mode with your editor (Erlide, ErlyBird, Distel, Emacs mode, etc.)?

(all results shown)

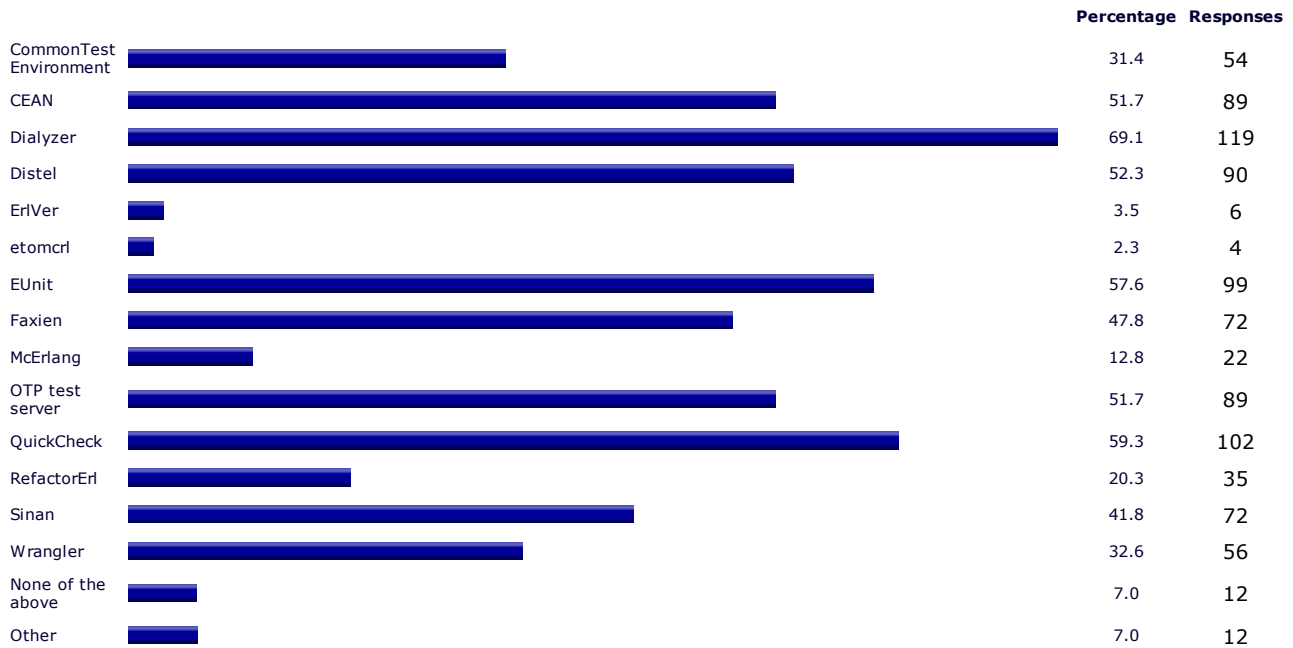
- Emacs mode
- Distel
- Yes, Emacs mode
- yes, Erlang mode for emacs
- Distel
- Distel, Erlang emacs mode
- no
- Emacs erlang-mode
- Yes, Erlide
- no
- erlang-mode for Emacs
- emacs mode
- erlang-mode
- no
- textmate erlang bundle
- no
- Emacs mode
- Yes, Erlide.
- emacs-mode
- Yes, custom
- Erlide
- Emacs mode
- Distel
- erlide, emacs, distel
- No
- no
- erlide
- distel
- none
- erlide
- emacs mode
- NO
- yes, vim-erlang integration package ;))
- Erlang mode
- Emacs mode, sometimes Distel
- Erlang-mode for Emacs
- Emacs
- Yes, distel and emacs mode
- Erlide, Emacs mode
- Emacs mode.
- Emacs Erlang mode ,and Flymake mode (eflymake for erlang)
- distel
- Erlware-mode
- Emacs mode delivered by OTP
- No
- ErlyBird
- patched emacs-mode from erlware
- No
- NO
- Emacs mode
- Emacs mode, distel

- distel
- Yes, Emacs/XEmacs mode
- Emacs mode
- Distel, Emacs mode, Wrangler, CEDET, ECB, Esense
- distel, erlang mode
- Erlide
- no
- Emacs mode, Distel, Wrangler, ECB, CEDET, Esense.
- Erlide
- Erlide
- Erlide
- no
- yes
- Emacs mode
- distel
- Emacs mode
- Distel, Erlware Emacs mode, Wrangler
- Emacs mode, Distel
- Emacs mode
- OTP emacs mode
- distel
- Emacs,Distel
- well, none of them work decently, therefore not.
- Distel
- Erlide, Emacs mode
- Erlang mode & Distel
- Emacs Erlang Mode
- Distel, plan to use Wrangler
- Yes: Distel, Emacs mode.
- No
- No
- Distel
- No
- distel, emacs mode, flymake
- Erlang Textmate Bundle
- no
- There is nothing on the market that would really help me in editing

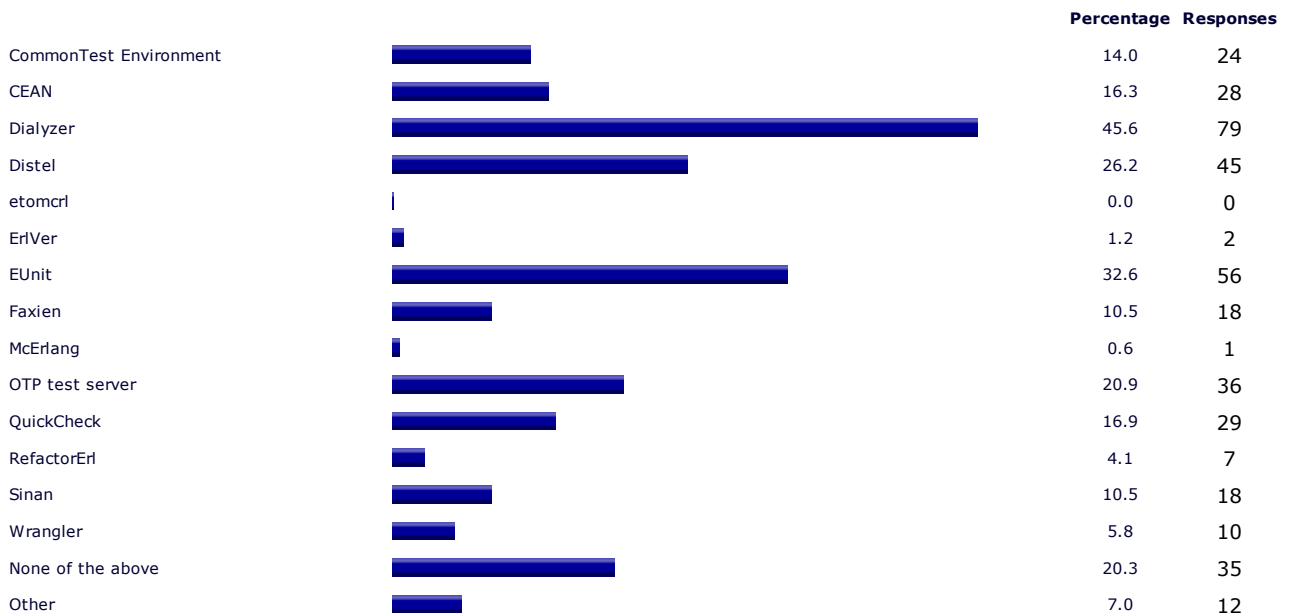
3) **On what platform is the development done?**

		Percentage Responses	
Linux/Unix -- Debian		9.9	31
Linux/Unix -- Redhat		6.4	20
Linux/Unix -- Suse		7.0	22
Linux/Unix -- Ubuntu		17.9	56
Linux/Unix -- Other		6.1	19
Macintosh -- Tiger		5.4	17
Macintosh -- Leopard		14.4	45
Macintosh -- Other		0.6	2
Solaris		8.9	28
Windows XP		12.5	39
Windows ME		0.0	0
Windows Vista		3.2	10
Windows -- Other		1.3	4
Other		6.1	19

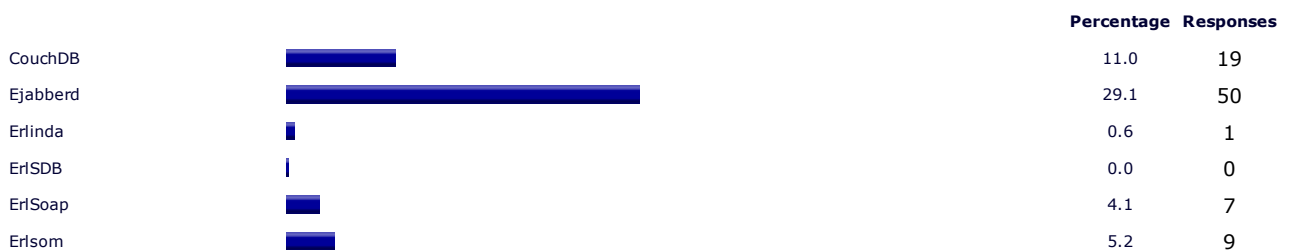
4) **Have you ever heard of the following Erlang tools?**










5) **Have you used the following Erlang tools?**



6) **What Erlang open source applications have you used?**



Erlyweb		15.7	27
OSERL		3.5	6
Yaws		59.9	103
Mochiweb		18.2	31
Tsung		19.8	34
None of the above		15.7	27
Other		8.1	14

7) **Would it help if we generate automated tests for these open source applications?**

		Percentage	Responses
Yes		76.5%	111
No		23.5%	34
Total responses:			145

8) **Do the tools you are using have any shortcomings which make them harder to use?**

(all results shown)

- EUnit is not good if you generate tons of tests, it requires too much memory.
- Yaws API changes every once in a while breaking previous behavior which makes it hard to keep it always up-to-date. Maybe it is because there are no regression tests.
- some of them have mediocre documentation
- No.
- no
- Configuration and usage of such applications simply sucks, at least for people who don't know erlang.
- It would be great if the eclipse plug-ins worked so that I could get the same benefits I get in Java:
- Function argument pop ups
- auto complete
- function description notes
- auto formatting
- good build environment and debugger
- Eclipse does not work on 64 bit Vista
- no
- Being able to explain Quickcheck properties to non-Erlang people.
- Yes
- erlang is a little bit unfriendly to novices. but after passing the first difficult stage of learning it's ok.
- hmmm. well. erlang likes to produce cryptic messages sometimes (on runtime matching errors, for example)
- Documentation is not enough, and need some examples.
- Yes
- no
- Some of them need a lot of configurations to run , and you need to do it manually
- Too much work to set them up
- Yes
- no
- It is difficult to stub my other modules that are not being tested.
- No
- NO
- The Erlang debugger is very basic.
- Generally a lack of tutorials, examples of usage and best practices.
- Not that I have noticed
- too much
- bad documentation
- too much
- documentation
- erl should be able to save "working image" state and have builtin editing.
- Time-consuming to interface with other languages
- It quite difficult to generate/write QuickCheck propties
- NA
- QuickCheck is commercial product
- Generally most tools lack the documentation to use them properly without digging into the source code.
- Haven't used them enough to comment but cursory opinion seems to indicate that test suites need better resource management facilities

9) **Are there any other tools you tried but decided not to proceed with? What were their shortcomings?(Unstable, Not completed, Not resolving my problem, etc).**

(all results shown)

- Sinan/Faxien is a promising set of tools, but last time I tried generating/compiling project with it, nothing seemed to work. I also wanted to use yatsy, but it was not clear to me how to configure the whole thing.
- no
- no
- Eclipse plug in did not do what I have asked for in 8. The emacs plug ins are too costly (time/difficulty) for my team to learn in time for them to be used.
- I tried to use dialyzer but it isn't easy to use.
- no
- Unstable and not completed.
- No
- silk
- hmmm. nothing special to tell here ;)
- no
- Erlinda not yet completed
- Common Test has too many wierd "conventions" such as naming of test modules etc.
- Eclipse - unstable
- Faxien and Sinan, they are not stable enough yet.
- No
- NO
- No.
- No, I haven't tried other tools
- No
- ErlyIDE - Not completed, Not resolving my problem, ErlyBird - Not completed, Not resolving my problem
- Tsun- complicate api
- ErIDE, ErlyBird
- Tried to use eunit, but had trouble. Had trouble setting up OTP test server.
- no
- Junit/test Unstable and diffecult to use
- EDTK
- Plugin for Eclipse for Erlang simply did not work for me.
- Erlide is not complete so not using that any more. Haven't tried it in a while though.
- Wrangler, because of lack of time. Erlide - too buggy
- erlide - is active even in an java project







10) **Do you have automated builds and test suites in place? If so, what system are you using?**

(all results shown)





- SCons + SCons Erlang
- erlang-t-build - excellent tool if your project is quite big and you already have a pile of applications that you reuse in new projects.
- no
- Shell scripts
- Yes, we have developed our own build system. Although, we will probably move to use one of the open source ones when they become more stable.
- Yes. ExtremeForge.
- No
- no
- - Company proprietary test system
- ExtremeForge at home
- No. Need some.
- make and erlmake
- Home-grown.
- test partner
- own simple_tests.erl which took after erlang testing server and a bunch of shell scripts/makefile around it.
- OTP test server
- Yes. Home-brewed regression tests.
- OTPBase.
- make & comp
- Yes. A customized lightweight testserver.
- Yes, home made
- no
- OTP base
- CruiseControl, Yatsy

- Yes, clearmake + common test + under development:quickcheck
- no
- Common Test
- No
- NO
- EUnit
- Home grown, ad hoc scripts and applications to, for example, assemble application releases.
- our own
- Yes, Quickcheck
- No
- yes, cruise control
- own framework and Eunit, but we're investigating
- common test
- I use my own CEDET/EDE emacs-lisp script for automated make.
- Built my own.
- not yet
- No, because test suites are counter productive. You tend to chain yourself to your tests rather than rethinking your problem. Automated builds are equally pointless if you don't have a mistaken notion of "testing".
- erlare
- Yes Common Test and our own build tools
- common test
- makefiles
- Not yet
- EUnit
- Yes, OTP Test Server
- Yes, CommonTest.
- Custom
- No
- otp testserver + eunit
- common test
- Gnu autotools, OTP Test Server

11) What methods are you using to test your system?

		Percentage	Responses
Test driven development		23.7	63
Develop – test iterations		33.5	89
Black box testing		19.9	63
Gray box testing		7.5	20
White box testing		13.2	35
Other		2.3	6

12) What testing tools are you using?

		Percentage	Responses
Eunit		18.3	42
QuickCheck		8.3	19
OTP Test Server		11.4	26
Common Test		9.6	22
Manually written functions		40.2	92
Other		12.2	28

13) Would you be interested in trying out the test tools which will result from the protest project?

		Percentage	Responses
Yes		88.7%	118
No		11.3%	15
		Total responses:	133

14) Do you have time and resources to try new tools?

		Percentage	Responses
Yes		65.4%	87
No		34.6%	46
		Total responses:	133

15) **Would you prefer training courses - with hands on exercises - for these new tools?**

		Percentage	Responses
Yes		37.6%	50
No		62.4%	83
Total responses:			133

16) **Would you consider short term consultancies?**

		Percentage	Responses
Yes		31.6%	42
No		68.4%	91
Total responses:			133

17) **Would you be willing to pay for these courses and consultancies?**

		Percentage	Responses
Yes		30.6%	38
No		69.4%	86
Total responses:			124

18) **Imagine that you have three wishes, what Erlang testing applications would you like to create, or improve already existing ones?**

(all results shown)

- Improve EUnit generally.
Have a free QuickCheck.
- one of the testing tools could be integrated with erlang-t-build
- An open-source testing application, which results to be really usable and open for further studies and developments
- 1. I would like a good debugger/test environment in one editor/IDE
2. I want a build system that works and is easy to deploy
3. Maybe a functional static code analysis tool which shows line/branch hits and performance statistics.
- Equivalent of Nunit
- Open source QuickCheck.
- easy to configure/install continuous integration engine with code coverage analysis and support to eunit, dialyzer and others
- Improve basic functions included in Erlang and the editors rather than making esoteric external applications.
- 1. IDE
2. Better integration of existing tools
- Don't know.
- have a freely available quickcheck for erlang
- Integrate OTP test server with other languages:
 - C++
 - Perl
 - Python
- yes
- It's easy to write write tests for functions, which only transform data and do not have side effects.
But it's too hard to test complex application, when you need isolate part of the code - i.e. from sockets, from another supervisor/worker/bla-bla processes - framework for writing stubs for common behaviors will be appreciated.
- Two another wishes are reserved for further expansion and use.
- I have one idea that I would like to try - it's a possible improvement to QuickCheck but as of yet I havn't had the time to properly evaluate it. No further wishes at this time.
- Load Testing Application , Distributed Testing Application
- Continious Integration Testing. Something that ties together all the various tools. Better documentation of the Common Test environment with lots of examples. Load testing is important. Testing of web UI is also a cumbersome area.
- Test aspects of Sinan and Faxien.
- I love the randomness of quickcheck; here's my three wishes:
 - 1 - an easier to use interface; using developers everyday language (i.e. UML state machines, sequence diagrams, activity diagrams).
 - 2 - a way to quickly and automatically generate a test suite based on the current code; it could also be semi-automatic. This would make this tool easy to deploy.
 - 3 - supervision with statistics on what has actually been tested; too see what functional test coverage my test runs provide.
- * integrated test environment with dialyzer, eunit, xref, etc
- * Code analysers (copy/paste detector, deprecated constructions detector, etc)
- * Make eunit a bit better
- 1 I would really like to have the time/resources to try QuickCheck
2 In erlyweb there is a way to modify already loaded modules. I use this to stub things while testing. I would like to have the time/resources to make that into a product.
- 1. Web-based console tied to the test framework.
2. Tool to interactively refactor test cases when there are changes on the system being tested.
3. Tool that automatically generates test cases.
- Some sweet multcore application for gaming. That can handle insane amounts of players :)
- usability, documented, no bugs
- Web user interface for test
Integration with Subversion
Integration with bugzilla
- * process protocol tester (something like UBF-B + Spin)
* a Dialyzer that works incrementally
- I wish QuickCheck was free and more documented.
- I would like QuickCheck to be free.

- eunit documentation / examples
- OTP server installation / examples
- More features on appmon

- I would focus on writing a better interactive development environment, where it is easier to visualize the message pathways, rather than focus on a false religion of testing.

- We are intested in courses no consultancies as we are not allowed to have constulance at LTE

- 1 wish only: line numbers in the build in erlang exceptions (without parse transforms :-)

- NA student learning Erlang

- The common test environment seems to be a god step in the right direction. However it is lacking in documentaton and experience so all its features are still unknown to me

- 1. more time

- 2. better quickcheck in jungerl

- easy to use test harness (checkout / run test) like cruise control

nicer interface to tests, stats / graphing etc

19) **Thank you for filling out our survey! If you are managed to pique your interest please choose from the following options**

	Percentage	Responses
Do you want a summary of this survey when completed?	45.3	68
Do you want to join the protest project announcement list?	24.7	37
Do you want us to contact you when the first tools are released?	30.0	45

20) **Please give us your contact details:**

1. Name:

2. Email:

3. Phone:

4. Company:

5. City:

6. Country:

21) **What is your role in the organisation?**

	Percentage	Responses
manager	7.7	9
project manager	6.8	8
system architect	12.8	15
developer	41.9	49
tester	2.6	3
researcher	6.8	8
Other	21.4	25
Total responses:		117