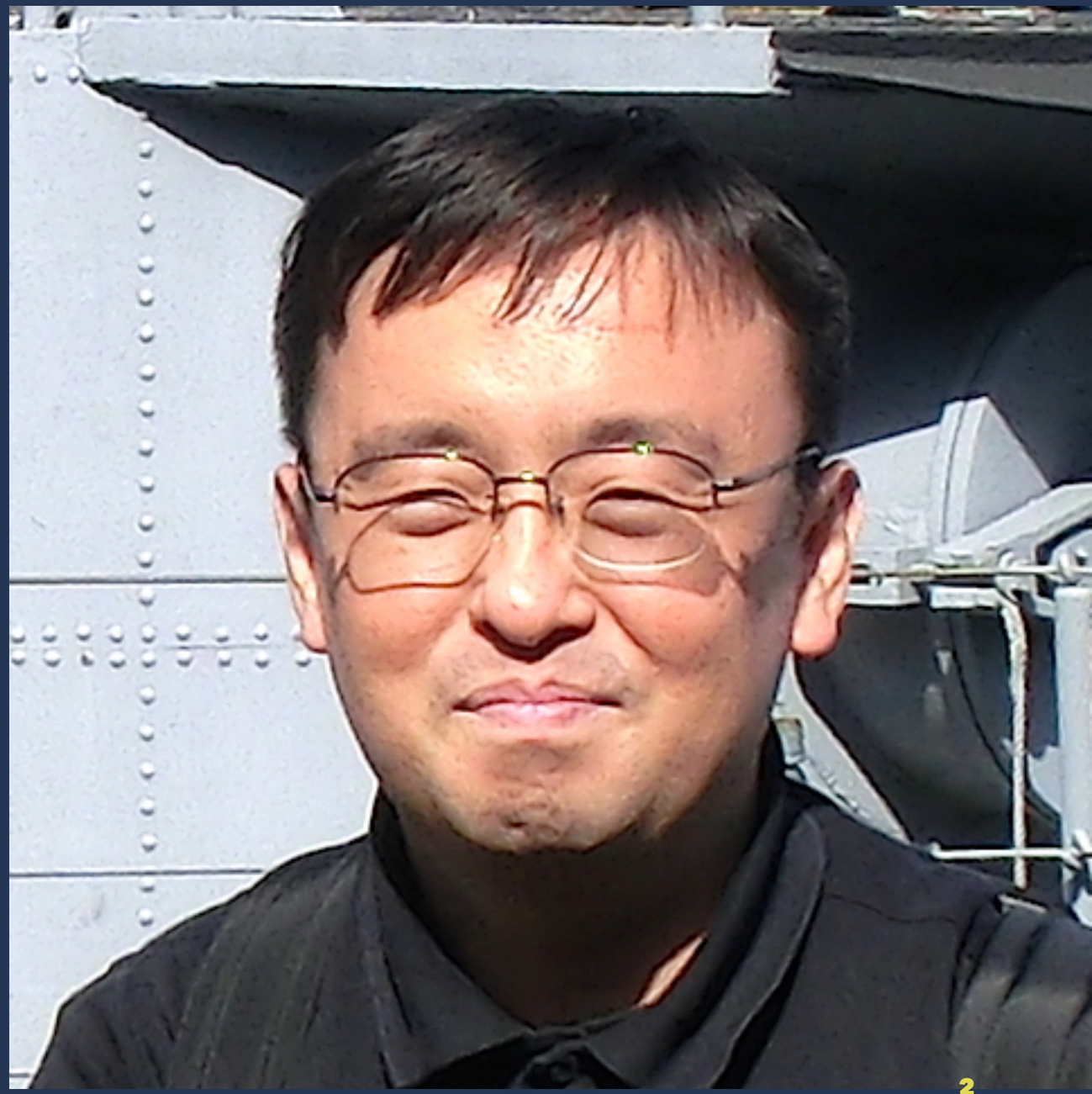
# Writing A New Erlang/OTP Module for Beginners

# Kenji Rikitake

23-MAR-2017
Erlang and Elixir Factory SF Bay 2017
San Francisco, CA, USA
@jj1bdx

- Erlang Factory SF Bay 2010-2016 speaker for seven times, and...
- Erlang and Elixir Factory SF Bay 2017 speaker (8th year!)
- Erlang `rand` module co-creator



# What I did for OTP

## Wrote Erlang PRNG code
## Tested and published the results
## Put the code into OTP

# This talk is *not* about

## Algorithms
## Random numbers
## Other implementation details

# This talk is about

## Development for Erlang/OTP
## Working with OTP Team
## Gaining support for your code

# Development for Erlang/OTP

# WARNING

## Don't try commiting code only for nurturing your ego or just for your own fame, please

# Why new code?

## Bugfix/security
## New features
## "Fix what I don't like"

# Do you have to change OTP?

**OTP is for *all* Erlang users**
**Who needs new code?**
**Once committed, removal is hard**

# Rationale for `rand` module

- `random` module period is too short and exploitable within <10 hours with a modern desktop computer, so it's basically a security fix
- Fixing API: eliminating the need for initialization
- New features: selecting multiple algorithms, normal distribution generator, jump functions from OTP 20

# Prototyping

## Independent repository
## Common Test and Dialyzer
## Learn erl_docgen

# erl_docgen document example

```
<func>
  <name name="jump" arity="0"/>
  <fsummary>Return the seed after performing jump calculation
      to the state in the process dictionary.</fsummary>
  <desc><marker id="jump-0" />
      <p>Returns the state
          after performing jump calculation
          to the state in the process dictionary.</p>
  <p>This function generates a <c>not_implemented</c> error exception
       when the jump function is not implemented for
       the algorithm specified in the state
       in the process dictionary.</p>
  </desc>
</func>
```

PDF
Top

**Reference Manual**
Version 0.9.2

Expand All
Contract All

**Table of Contents**

Algorithm-dependent state that can be printed or saved to file.

## EXPORTS

**export_seed() -> undefined | export_state()**

Returns the random number state in an external format. To be used with seed/1.

**export_seed_s(X1 :: state()) -> export_state()**

Returns the random number generator state in an external format. To be used with seed/1.

**jump() -> NewS :: state()**

Returns the state after performing jump calculation to the state in the process dictionary.

This function generates a not_implemented error exception when the jump function is not implemented for the algorithm specified in the state in the process dictionary.

**jump(State :: state()) -> NewS :: state()**

Returns the state after performing jump calculation to the given state.

This function generates a not_implemented error exception when the jump function is not implemented for the algorithm specified in the state.

**normal() -> float()**

Returns a standard normal deviate float (that is, the mean is 0 and the standard deviation is 1) and updates the state in the process dictionary.

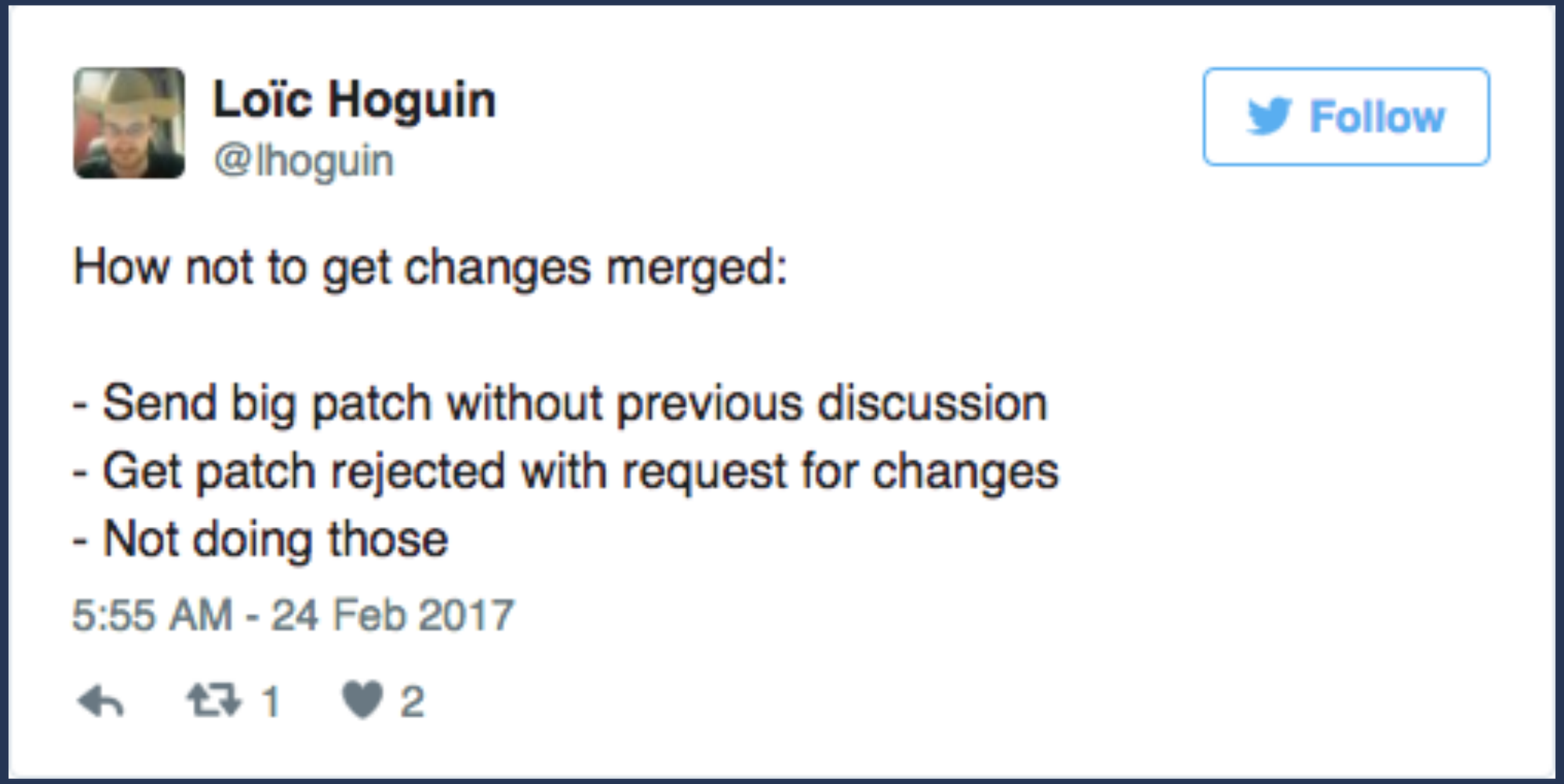
**normal_s(State0 :: state()) -> {float(), NewS :: state()}**

Returns, for a specified state, a standard normal deviate float (that is, the mean is 0 and the standard deviation is 1) and a new state.

# Working with OTP Team

# Let me show you some excellent tweets from the author of Cowboy HTTP server

# Tweet from Loïc Hoguin (@lhoguin at Twitter)



# Tweet from Loïc Hoguin (@lhoguin at Twitter)

# Tweet from Loïc Hoguin (@lhoguin at Twitter)

# What to include in a GitHub PR

- What you are going to do with the PR
- How the PR changes the behavior of the code
- What and how the changes will and *will not* affect the other modules and applications
- Common Test cases and type specifications
- Documentation

# What to do when issue a PR

- Report the bug before submitting a bugfix PR
- Choose the right branch
- Separate commits for separate changes
- Make sure that each commit can be compiled
- Make sure that each commit works
- Use `git rebase` so that OTP Team can `git bisect`

# Communication with OTP Team

- Choose the GitHub Issues and private email wisely
- OTP Team are working in Central *European* Time Zone
- In Europe, DST is different from USA/Canada
- OTP Team have their weekends, holidays, and vacations
- Don't expect a prompt reply during non-working hours
- OTP Team members have to handle multiple modules

# How to gain community support for your code

# Promote your code

- Publish ready-to-use modules on GitHub and elsewhere
- `rand` module have *six* related modules released
- Give talks and publish papers on conferences
- Write a PoC and show what the problem is

# Code maintenance and support

- When your code is in OTP you are responsible as the same as the OTP maintainers for the part of the code
- You are expected to contribute a further enhancement or a new feature when they are available
- Old code may be deprecated and removed
- Adding new features should be done *very carefully*

# Licenses for Erlang/OTP

- **Erlang/OTP: Apache License 2 (APLv2)**
- **APLv2: *inconpatible* with GPLv2 and GPLv3**
- **MIT and BSD License code can be easily incorporated**
- **Your code may need to be relicensed to OTP Team**
  - The copyright notice is limited in simple manners


# OTP needs your help

## Your contribution is always welcome

# References

- My WIP document: Writing OTP Modules
  - http://docs.jj1bdx.tokyo/writing-otp-modules/html/index.html
- Source of Writing OTP Modules
  - https://github.com/jj1bdx/writing/otp-modules

# Support for this presentation is provided by


# Pepabo R&D Institute, GMO Pepabo, Inc.

# Acknowledgment


- Dan Gudmundsson - rand module principal developer
- Sebastiano Vigna - Xorshift*/+ inventor
- Erlang Solutions


Thank you
Questions?

Photo credits:

- Title slide: Davide Ragusa, from Unsplash.com
- Kenji Rikitake's face: Yutaka Sakurai and Naoki Sakurai, taken in front of USS Pampanito at Pier 45, San Francisco, CA, USA, March 2015
- "This talk is not about" slide: Markus Spiske, from Unsplash.com
- "OTP needs your help" slide: Matheus Ferrero, from Unsplash.com
- "Thank you" slide: Chris Brignola, from Unsplash.com

(All Unsplash.com photos are licensed under Creative Commons CC0 License)