Introducing A Functional Language At Work

A Developer's Guide

@robertkluin

FP.
How do I love thee?

Code is easier to reason about?

Fewer surprises?

Decrease in defects?

Reduced development time and effort?

Basically, it is better in every way.

Right???


Danielle Griscti


Pixabay user 34356

Background

We were very friendly to new tech

Open to anything that could be better

Including beta technologies

Our CTO encouraged pushing the limits and being aggressive

It was who we were

Lesson 1

Scala

@robertkluin

Brace yourself…

This is a sad, sad tale

Context

Project

- Critical Project
- Tight timeline
- Good fit for FP, and scala in particular

Context

Team

- Relatively newly formed team
- Two highly skilled team members
- Several less-experienced team members
- Most had some familiarity with FP
- New to Scala

There was one very vocal supporter of Scala

Most of the team was indifferent

Little practical experience,
but that's OK

The Scala fan went off the Scala deep end

The Scala indifferent wrote mostly functional code

The junior devs wrote java


KOMATSU
SD23
USAF

Everyone wrote different stuff

The lack of consistency and variety in coding styles within libraries proved challenging—even with the fans

SBT was a necessity

No one liked the SBT build tooling

Despite being JVM based,
it did not fit into our infrastructure

It ended in pain and disagreement

This was not an issue with Scala

It was an issue with

```
us ++ scala
```

, err

```
us ::: scala
```

Takeaways

It was **too flexible for our** level of experience, size, corp **culture**

We picked a terrible project,
given our circumstances

This spooked some management about FP


Alexas_Fotos

How is FP better than the alternative?

If you’re selling the benefits,
don’t ignore the costs

We didn’t intentionally ignore the costs

Falschfahrer / Wikipedia

We thought flexibility would help

We expected JVM to help

We thought being able to write Java would help

History

“Pythonic”

Easy to identify

One “obvious” way to do things

“Pythonic” was ingrained
into **our culture**

We were writing distributed systems

Within some teams, we realized OOP practices lead to many, many defects


HOME
HD
CLIP

Hard to debug defects

We started naturally writing functional code

Bugs could take hours or weeks to trace down

This is what brought us
to LambdaConf

Lesson 2

golang

@robertkluin

We introduced golang
almost by mistake

Prototype that ran away on us

It spread like wildfire

Everyone was redoing stuff in golang

It was new, it was exciting, …

it was premature

No one was shipping stuff

(Note: we did eventually ship stuff)

Takeaways

The lack of flexibility helped us

Golang is crazy opinionated

Its tooling enforce code style

It gave us a “Pythonic” feel

It is limited, in many ways

Its type system is limited

The lack of flexibility helped us.

It **fit our culture**

We should have shipped
one thing first!

Doing too much all at once
turned out to be a very risky idea

This strained many parts of the org

It was extremely high risk

Lesson 3

Elm

@robertkluin

Pushed by management

Full disclosure, that was me

Opinionated

Outstanding tooling

Shipped full new product within 1 month

Eliminated bugs the other stacks had been chasing for months

The defect rate bottomed out

Proved massive refactor was not only possible, but safe!

The product was a huge value gain,
but non critical path

It was also isolated

Takeaways

Like golang, it **fit our culture**

A management champion
was useful for getting started

The value was immediately proven

Starting with an isolated, ancillary product helped

Sadly, being replaced now

The **promoters moved on**to other companies

Aside 1

The Little Rascal who snuck in Haskell

@robertkluin

Used Haskell to create a runnable API spec

Documented the expected behaviors of the system

Made experiments possible

Documented the API

Lesson 4

Clojure

@robertkluin

The immutable data model
was an awesome fit

Completely unnatural in other languages

Had to provide “native” feeling client libraries for consumers

Built a prototype that matched the need

They offered to build and maintain support into the ecosystem

Takeaways

Team is focused purely
on proving value

And, hitting deadlines
with minimal defects

Not trying to make it
a company standard

They worked with inf & ops

Aside 2

The Language Doesn't Matter

We had numerous attempts
to introduce favorite languages

They never worked out

Because we focused on the tech,
not the value

Start small

Demonstrate the value

Demonstrate fitting into your ecosystem

Pick a low-risk area / project

Find a management champion

Go slow, be careful going all-in

Learn first!

Help your Ops / Infrastructure Eng staff

Plan for hiring

Encourage usage of functional paradigms in non-functional langs

HAMLIN'S
WIZARD OIL

THE GREAT MEDICAL WONDER.

There is no Sore it will Not Heal, No Pain it will not Subdue.

PLEASANT TO TAKE

MAGICAL IN ITS EFFECTS.

HAMLIN'S
BLOOD AND LIVER PILLS

For Liver Complaint, Constipation,

AND ALL

Disorders of the Stomach and Digestive Organs.

PREPARED AT THE LABORATORY OF

HAMLINS WIZARD OIL COMPANY, CHICAGO, ILL.

Promotion, evangelism, and education are the biggest factor of success

This is true of all tech we introduced