Introducing A Functional Language At Work

A Developer’s Guide
Benefits of FP
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.
You
Right???
Background Story
We were very friendly to new tech
Our CTO encouraged pushing the limits and being aggressive
Open to anything that could be better
Including beta technologies
It was who we were
Scala
Brace yourself...
This is a sad, sad tale
Context

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

● Relatively newly formed team
● Two highly skilled team members
● Several less-experienced team members
● Most had some familiarity with FP
● New to production 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
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 SBT
Despite being JVM based, it did not fit into our infrastructure
It ended in pain and disagreement
Takeaways
This was not an issue with Scala
It was an issue with
us ++ scala, err
us :::: scala
It was **too flexible for our** level of experience, size, corp **culture**
We picked a terrible project, given our circumstances
A hidden mistake...
We didn’t intentionally ignore the costs
We thought flexibility would help
We expected JVM to help
We thought being able to write Java would help
Our Culture
“Pythonic”
Easy to identify
One “obvious” way to do things
“Pythonic” was ingrained into our culture
Why were we interested in FP?
We were writing distributed systems
Within some teams, we realized OOP practices lead to many, many defects
Hard to debug defects
We started naturally writing functional code
Bugs could take hours or weeks to trace down
We had real problems to solve
We actively wanted a solution
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)
But it was a success!
Takeaways
The lack of flexibility helped us
Golang is crazy opinionated
Its tooling enforces code style
It gave us a “Pythonic” feel
And, it is limited, in many ways...
Its type system is very simplistic
The lack of flexibility helped us
It fit our culture
An observation
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
But, we learned from that mistake
Pushed by management
Full disclosure, that was me
Opinionated
Outstanding tooling
Shipped full new product within 1 month
Eliminated bugs we had been chasing for months in the prior stack
The defect rate bottomed out
Proved a 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
Starting with an isolated, ancillary product helped
The value was immediately proven
A management champion was useful for getting started
Sadly, it has been replaced now
The promoters moved on to other companies
Promotion and evangelism is critical
The Little Rascal who snuck in Haskell
Used Haskell to create a runnable API spec
Documented the expected behaviors of the system
Made experiments possible
Documented the API
Got a larger group interested in Functional Programming
Rapidly built a prototype that solved the business problem
The immutable data model was an awesome fit
They offered to build and maintain support into the ecosystem
Immutability was completely unnatural in other languages
Had to provide “native” feeling client libraries for consumers
Takeaways
Team is focused purely on proving value
And, hitting deadlines with minimal defects
They worked with inf & ops
They actively try to ensure it is not about the language, but the value.
Manage risks
Pick a low-risk area / project
Go slow, be careful going all-in
Demonstrate the value
Learn first!
Encourage usage of functional paradigms in non-functional langs
Plan for hiring
Be mindful of “disruption”
Demonstrate fitting into your ecosystem
Help your Ops / Infrastructure Eng staff
Find a management champion
The Language Doesn’t Matter
We had numerous attempts to introduce favorite languages
Focusing on the tech, not the value, rarely worked out
Your culture is key
Pick solutions that can fit into your culture
Languages and tooling can shape and alter a company’s culture
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 & LIVER PILLS

For Liver Complaint, Constipation, AND ALL
Disorders of the Stomach and Digestive Organs.

PREPARED AT THE LABORATORY OF
HAMLIN'S WIZARD OIL COMPANY, CHICAGO, ILL.
Promotion, evangelism, and education are critical for success.