



# 30 Years of Ruby

*OSS Vision*  
*Ruby Association*

Yukihiro "Matz" Matsumoto  
@yukihiro\_matz



# Lessons from the History



# I Love Programming Languages



# Language Geek



# Trivia



What is the First Programming Language?



# FORTRAN (1954)



# Plankalkül (1942)





# Konrad Zuse



Z3



- Assignment
- if-statement
- Loop



- Structure
- Subroutine
- Array



- Assertion
- Exception
- Goal-directed Execution



1972 Spec Released



# 1998 First Compiler Implemented



**Idea itself is not Enough**





# Provide Useful Implementation



# Grow with Users



# What is the Simplest Programming Language?



Syntactically



Lisp, FORTH, APL (1960s)



None Popular Now



“

*If a language is simple, our software will be more complex, because total amount of complexity is constant*

”

Larry Wall (-ish)



**Simple is (not always) Best**





Since Human Mind is Complex



# Follow the Mental-Model



# Other Notable Languages



PL/I



Ada



Complexity is our Enemy



# Creeping Featurism



# Linear Complexity is Better





Than Exponential Complexity



# Bigger (not always) the Better



# Perl



# Perl1.0 (1986)



# Perl5.0 (1993)



# Perl5.38 (2023)



# 2000 "We are Bored" Event



361 RFCs





# Perl6



# Raku



# Boring Pitfall



“

*we are f\*\*ed unless we can come up with something that will excite the community, because everyone's getting bored and going off and doing other things*

”



# The Biggest Enemy of OSS



Or Any Project Driven by Motivation



OSS need to keep moving forward



# We Need to Attract the Community





# Second System Syndrome



When we re-create the system, we often  
become too ambitious



to Solve Everything all at Once



"Scrap & Build" is a Bad Idea



We Should Evolve Gradually



# Ruby



# I Once Followed Perl6 Idea



# Ruby2 (2003)





# RFCs



I was too lazy to bookkeep



# Ruby1.9 (2007)



- YARV VM
- M17N



# Ruby2.0 (2013)



- Keyword Argument
- Refinement
- Module#prepend



# Ruby3.0 (2020)



- "Real" Keyword Argument
- Pattern Matching
- JIT
- Heap Compaction





Ruby4.0 (2027/2030)?



- Package System ?
- Steering Committee ?



# The Future



# Lessons



Idea itself is not Enough



Simple is not always Best



Bigger not always Better



# Boring Pitfall





# Motivation Matters



# Second System Syndrome



We Need to Progress Step by Step



# Survival



# How to Survive



# How to Keep Ruby Alive for Next 30 Years



# Assumptions



# Human Being Survive





# Programming Survive



# We Program in Languages



# Upon Those Assumptions



We Need Community



# Community needs new Blood



# Attract Young Programmers



# Job & Money



# Keep Ruby Attractive for Companies





How Attractive?



- Reliability
- Productivity
- Efficiency



To be Reliable



We need Community Assets



Tools, Gems, etc.



Textbooks, Tutorials, Schools, etc.



# Use-cases, Blog Posts, Shared Experiences



To be Productive





# Improve Ruby the Language



# Improve Supporting Tools



Rubocop, RubyMine, Ruby-lsp,...



To be Efficient



# Improve Ruby implementation(s)



# Encourage Investment to Ruby



If Ruby is 10% Faster,



It will Decrease Backend-cost by 10%





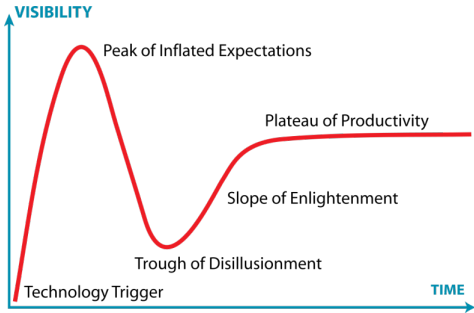
We Need more Investment



This is the Last Chance



# Hype Cycle



# Hype Cycle



Only Paranoid can Survive



# Ruby is Wonderful



Come, Join Us



Create the Better Future





Otherwise,



We will be forced to use...



# Tipping Point



Only Paranoid can Survive



We will Keep Moving Forward



We will Lengthen Our Stride



We will Create Value



# As a Community





Come, Join Us



Create the Better Future



# That's the Power of Ruby



Thank you