CCS 2017
Cancún, México

#ECC17

Major Transitions in Information Technology

Sergi Valverde
University Pompeu Fabra
Complex Systems Lab (Barcelona)

@svalver

upf. Universitat Pompeu Fabra Barcelona
IBE CSIC upf.
Gobierno de España · Ministerio de Economía, Industria y Competitividad
Unión Europea
Fondo Europeo de Desarrollo Regional "Una manera de hacer Europa"
Agencia Estatal de Investigación
SANTA FE INSTITUTE

Goal

AN EVOLUTIONARY THEORY OF INVENTION AND INNOVATION

Technological Change is…

(1) A physical process of interaction with the world

- Useful technology has adaptive value
- Environmental constraints

ground stone tools: hafted adzes left and right, grooved and hafted axe in center

Technological Change is…

(2) A social process of interaction with other people

- Not all inventions have adaptive value
- Imagination exceeds the boundaries of rationality
- Society selects inventions according to cultural goals

Wind-Harnessing Bike (2005)

Patterson Sailing Bicycle

Cultural Phylogenies

BEHAVIORAL AND BRAIN SCIENCES (2016), Page 1 of 65
doi:10.1017/S0140525X14001356, e0

Norenzayan et al.

The cultural evolution of prosocial religions

Nevertheless, we caution against rushing to embrace analytical techniques imported from genetic evolution – used to reconstruct species phylogenies – to cultural evolution. Cultural evolution is in some crucial respects unlike genetic evolution (see discussion in sect. R3.1). Species, for example, are not subject to intergroup competition that creates massive and directed horizontal transmission of only some traits.

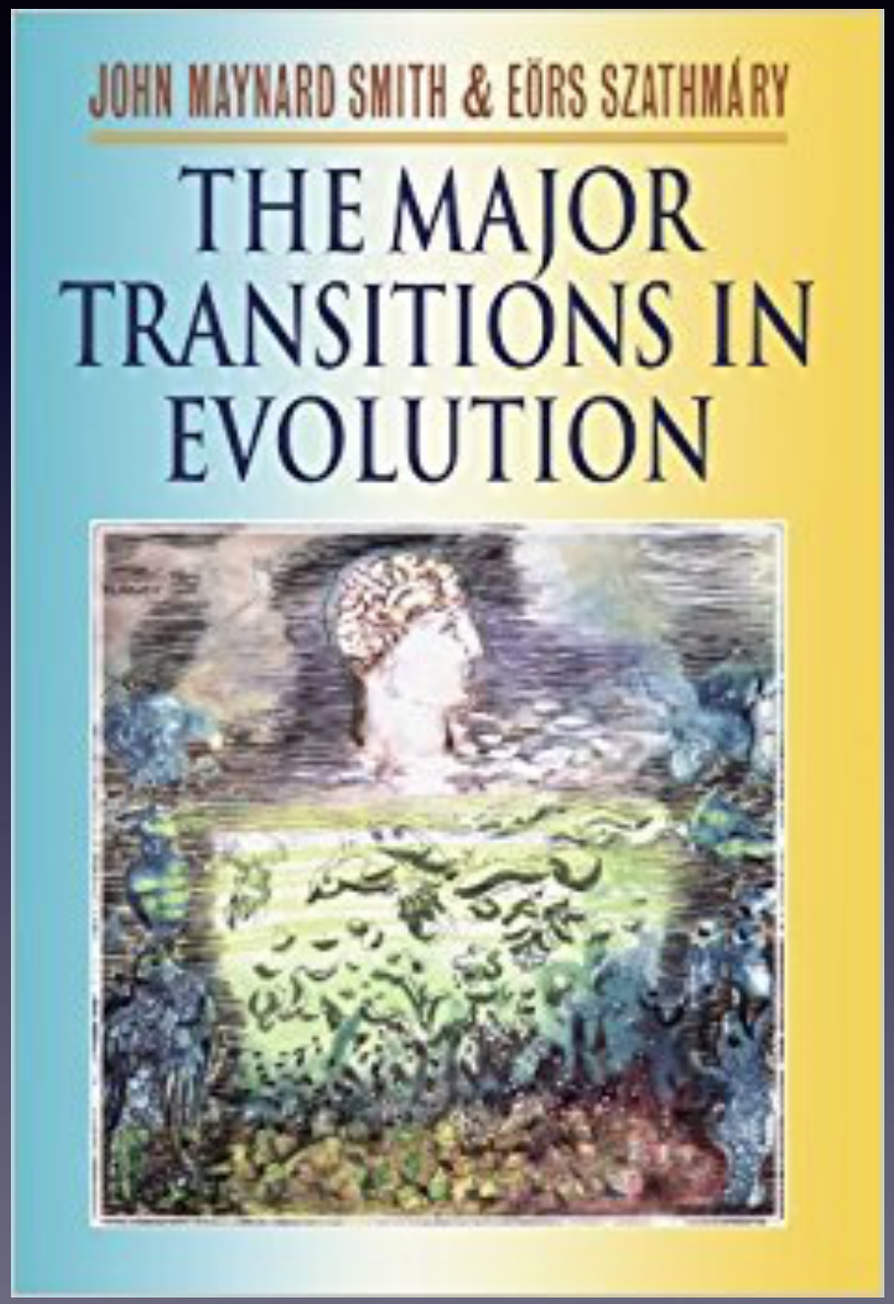
The Pleasures and Perils of Darwinizing Culture (with phylogenies)

Russell D. Gray, Simon J. Greenhill and Robert M. Ross
Department of Psychology
University of Auckland
Auckland
New Zealand
Correspondence to: rd.gray@auckland.ac.nz

To appear in *Biological Theory*, Volume 2 (4).

Abstract
Current debates about "Darwinizing culture" have typically focused on the validity of memetics. In this paper we argue that meme-like inheritance is not a necessary requirement for descent with modification. We suggest that an alternative and more productive way of Darwinizing culture can be found in the application of phylogenetic methods. We review recent work on cultural phylogenetics and outline six fundamental questions that can be answered using the power and precision of quantitative phylogenetic methods. However, cultural evolution, like biological evolution, is often far from tree-like. We discuss the problems reticulate evolution can cause for phylogenetic analyses and suggest ways in which these problems can be overcome. Our solutions involve a combination of new methods for the study of cultural evolution (network construction, reconciliation analysis, and Bayesian mixture models), and the triangulation of different lines of historical evidence. Throughout we emphasize that most debates about cultural phylogenies can only be settled by empirical research rather than armchair speculation.

Origin of Novelties



Major Transitions

Evolution of Replicating Molecules

Evolution of Chromosomes

Evolution of DNA

Evolution of Eukaryotes

Evolution of Sex

Evolution of Multicellularity

Evolution of Eusociality

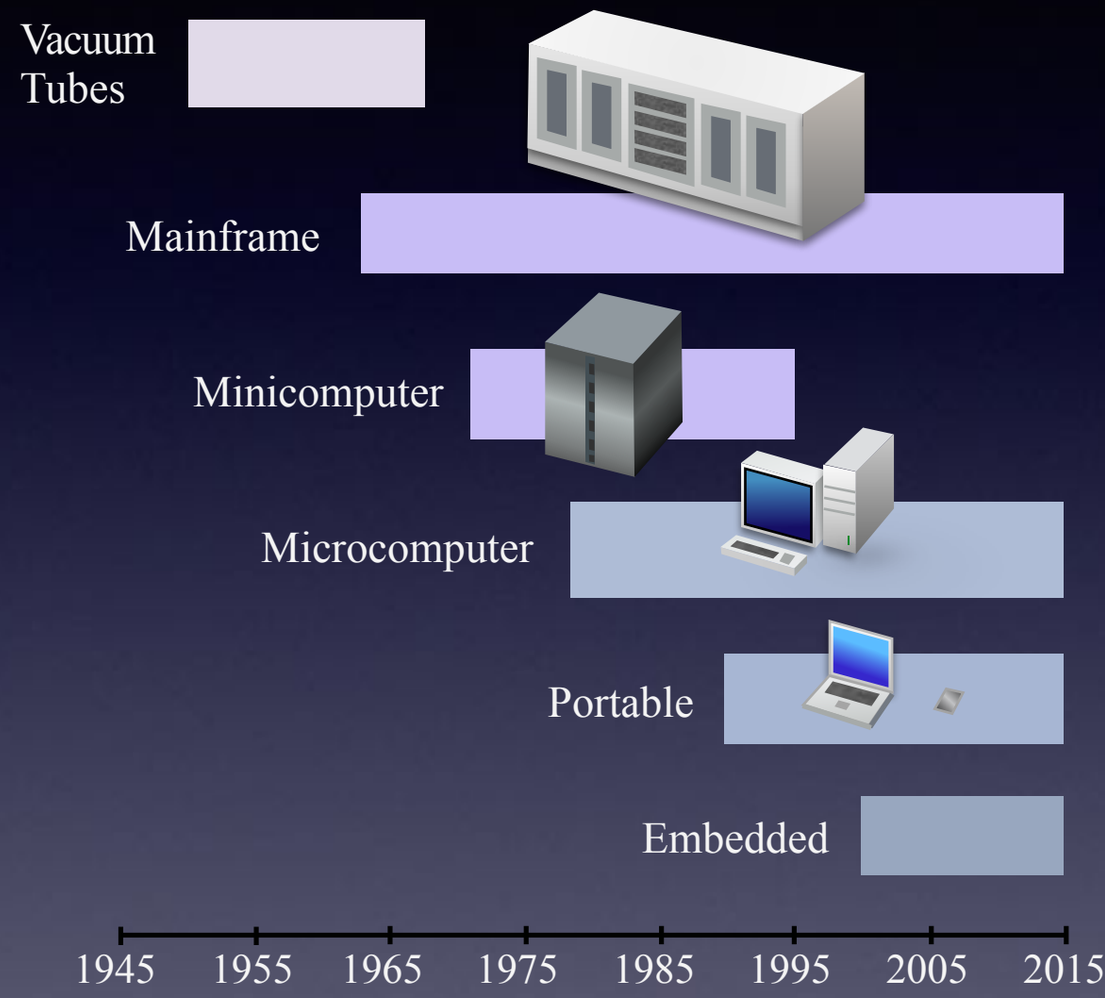
Evolution of Languages and Culture

Properties of Major Transitions

- Smaller entities join to form larger entities.
- The smaller entities' reproduction becomes dependent on the larger entity.
- Differentiation of roles with the larger entity.
- Conflict between elements within the larger entity occur.
- New ways of transmitting information arise.

Technological Transitions

- Study technology to better understand who we are and where we go
- Not all technologies are of equal importance.
- Times and places in the past when a large number of novel artefacts/behaviours appeared together or in rapid succession.



Cultural Evolutionary Tipping Points in the Storage & Transmission of Information (2013)

Bentley, R. A. & O'Brien, M. J.

Major Transitions in IT

Network models of cultural transmission might provide explanations for periods of rapid technological change

PHILOSOPHICAL TRANSACTIONS B

rstb.royalsocietypublishing.org

Review

Cite this article: Valverde S. 2016 Major transitions in information technology Phil. Trans. R. Soc. B 371: 20150450. http://dx.doi.org/10.1098/rstb.2015.0450

Accepted: 18 May 2016

One contribution of 18 to a theme issue 'The major synthetic evolutionary transitions'.

Subject Areas: evolution

Keywords: transition, software, innovation, technology, networks, evolution

Author for correspondence: Sergi Valverde e-mail: sergi.valverde@upf.edu

Major transitions in information technology

Sergi Valverde[1,2,3]

[1]Complex Systems Lab, Universitat Pompeu Fabra, Dr Aiguader 80, 08003 Barcelona, Spain
[2]Institut de Biologia Evolutiva (CSIC-UPF), Passeig Marítim de la Barceloneta, 37, 08003 Barcelona, Spain
[3]European Centre for Living Technology, Ca' Foscari University of Venice, 30124 Venice, Italy

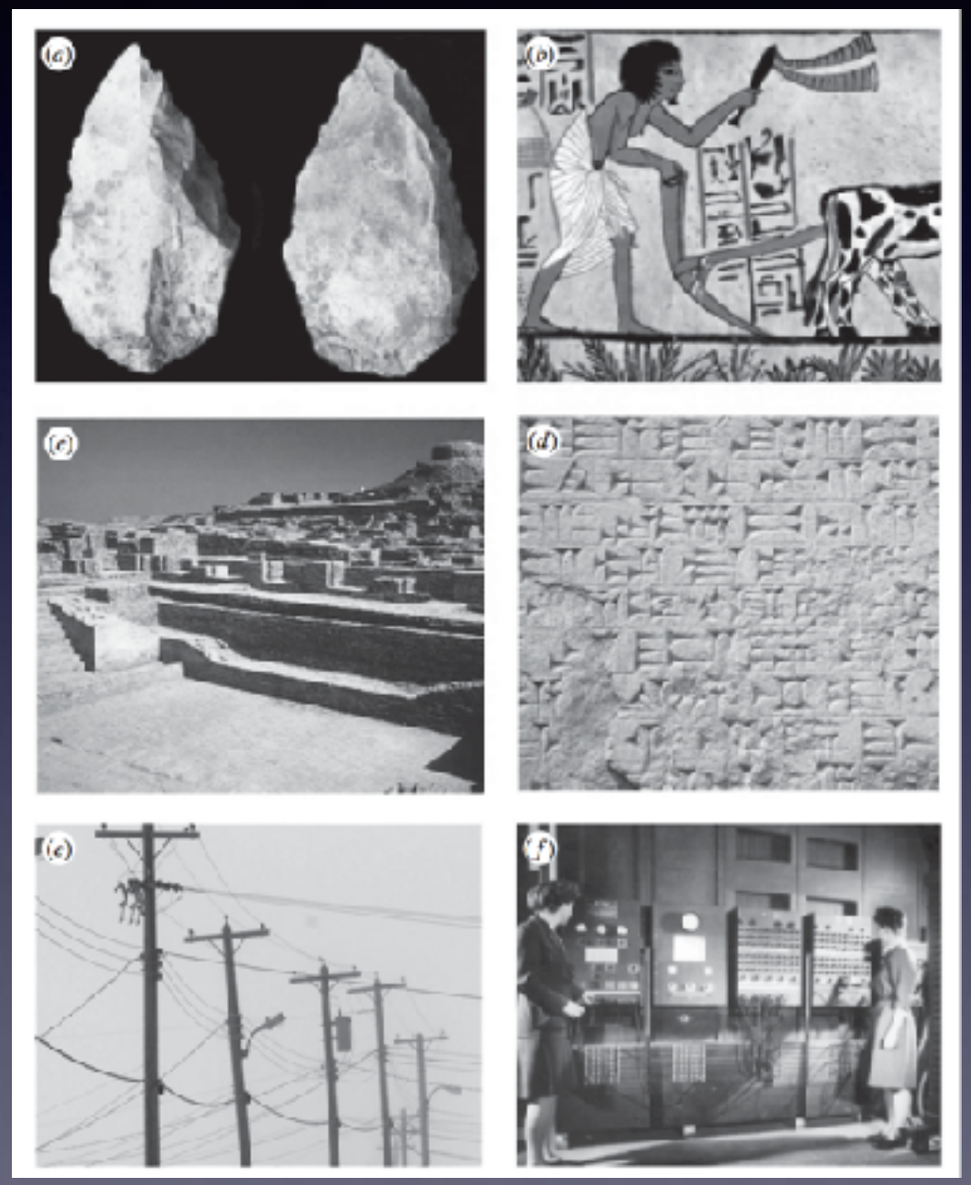
When looking at the history of technology, we can see that all inventions are not of equal importance. Only a few technologies have the potential to start a new branching series (specifically, by increasing diversity), have a lasting impact in human life and ultimately became turning points. Technological transitions correspond to times and places in the past when a large number of novel artefact forms or behaviours appeared together or in rapid succession. Why does that happen? Is technological change continuous and gradual or does it occur in sudden leaps and bounds? The evolution of information technology (IT) allows for a quantitative and theoretical approach to technological transitions. The value of information systems experiences sudden changes (i) when we learn how to use this technology, (ii) when we accumulate a large amount of information, and (iii) when communities of practice create and exchange 'free' information. The coexistence between gradual improvements and discontinuous technological change is a consequence of the asymmetric relationship between complexity and hardware and software. Using a cultural evolution approach, we suggest that sudden changes in the organization of IT's depend on the high costs of maintaining and transmitting reliable information.

This article is part of the themed issue 'The major synthetic evolutionary transitions'.

1. Introduction

Technology is a powerful agent of change that allowed us to expand our capacities beyond what any other species ever did [1]. The development of external forms of changing our external world, while becoming more and more independent of its uncertainties, has been key for our success. The presence of a symbolic mind and the capacity of propagating non-genetic information by means of language co-evolved with a rapidly improving technological potential. In this context, although other species have been capable of a limited use and construction of tools, none shares with us the cumulative nature of our technology [2].

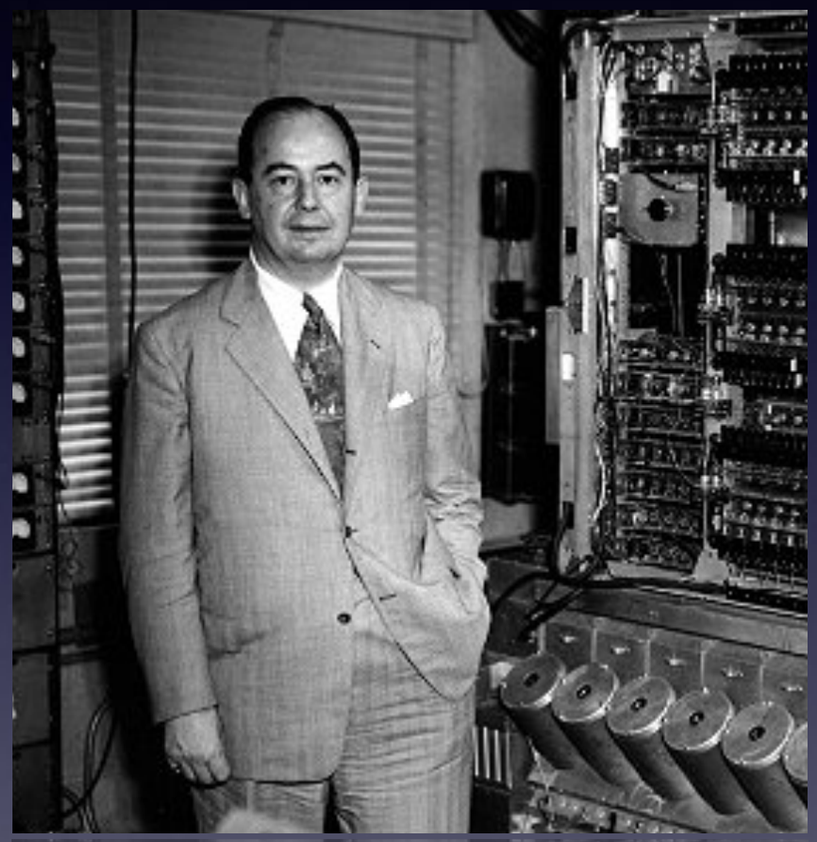
Several key events mark our historic and prehistoric record of artefacts and inventions. Not all innovations are, however, of equal importance. Only a few



Major Transitions in Information Technology (2016) Valverde, S.

IT Transition

1. When we learn how to use computers

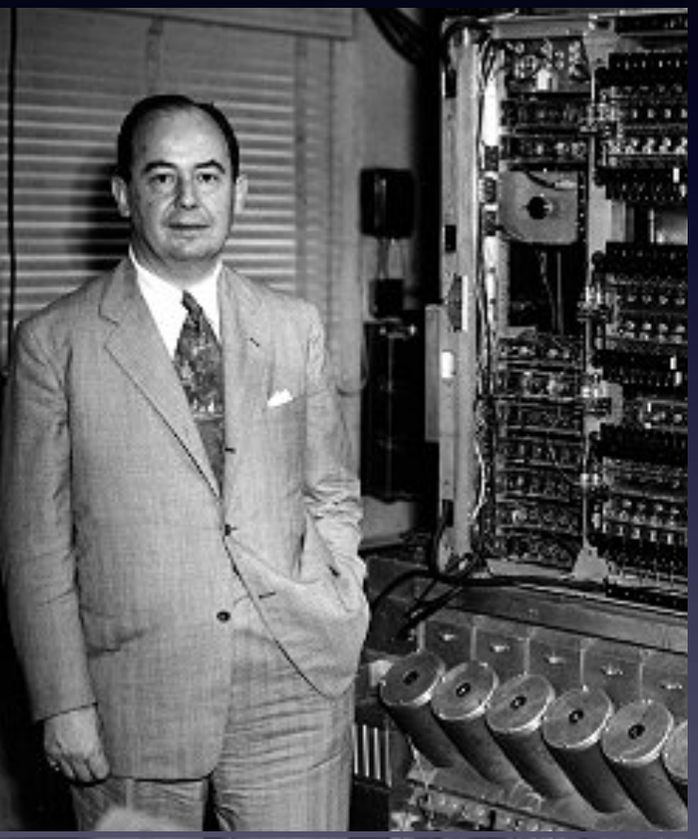


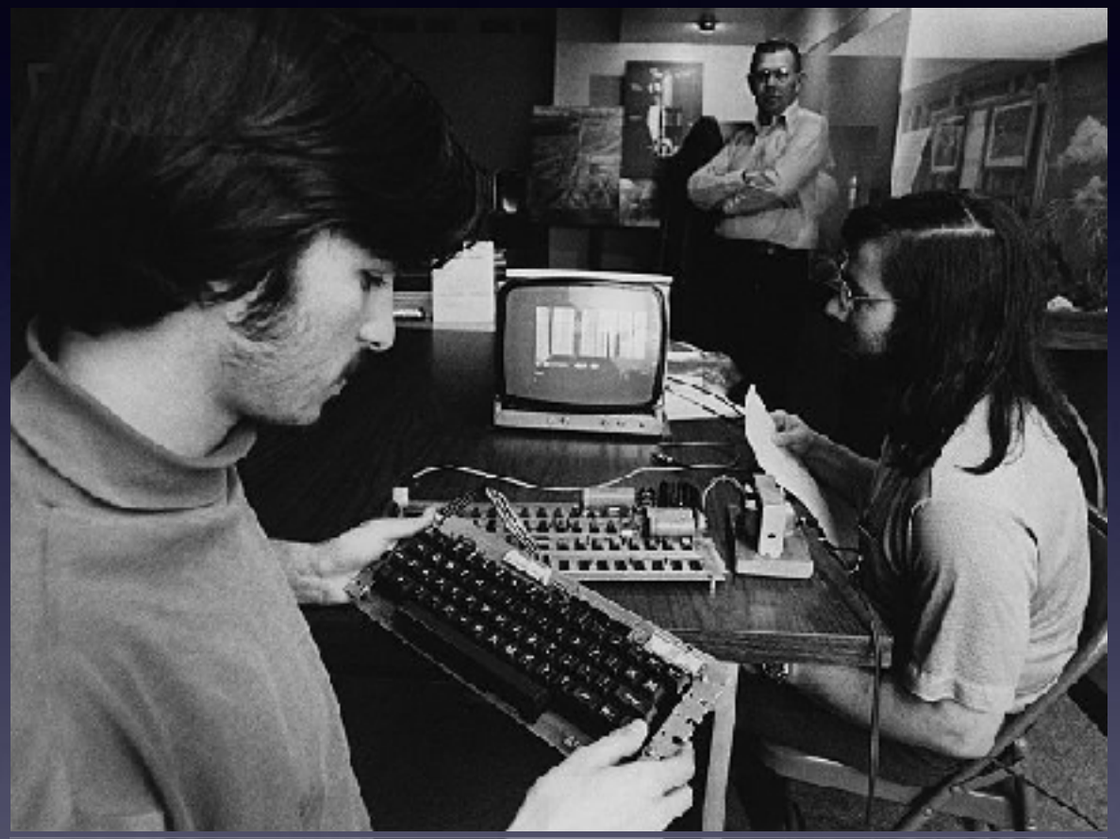
John von Neumann

First Draft of a Report on the EDVAC (1945)

IT Transition

2. When computers are for everyone

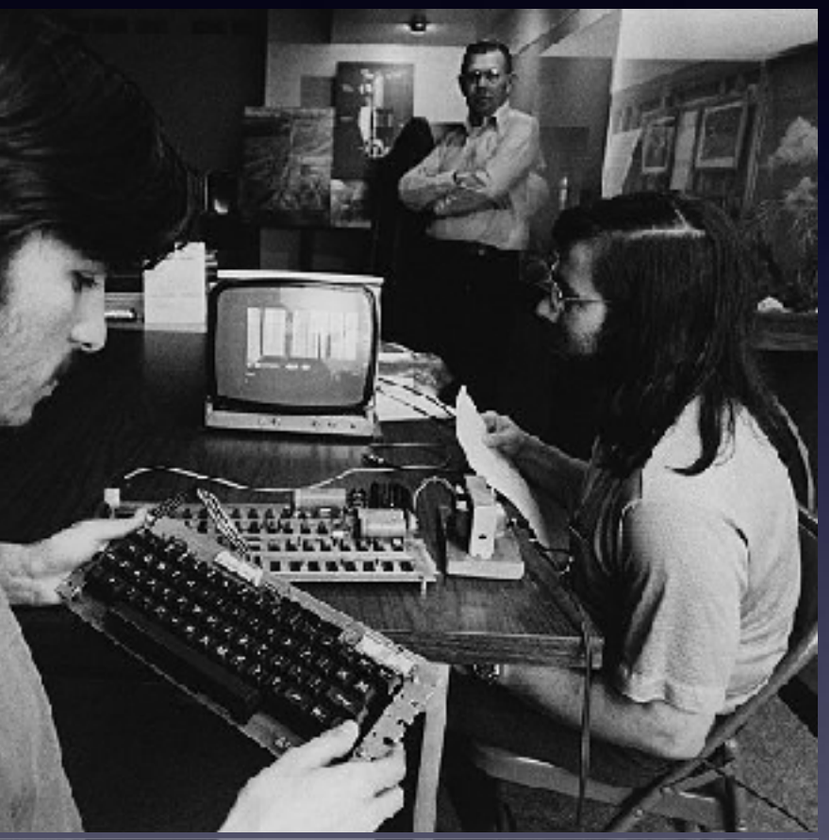




Steve Jobs & Steve Wozniak at
the Homebrew Computer Club

IT Transition

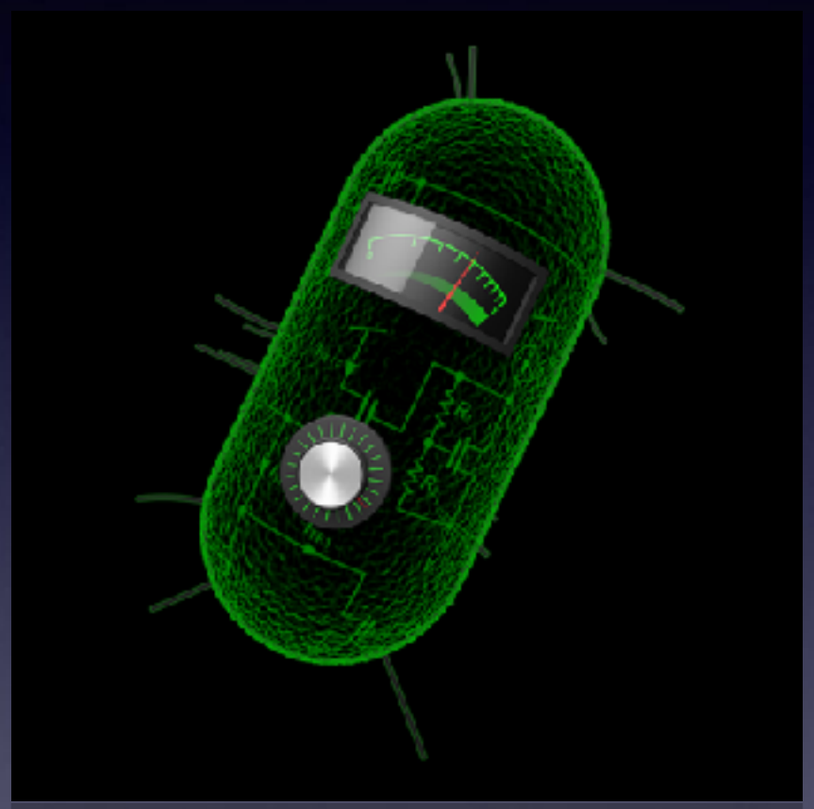
3. When communities exchange free information



Richard Stallman

The Next Transition ?

4. When biology meets programming

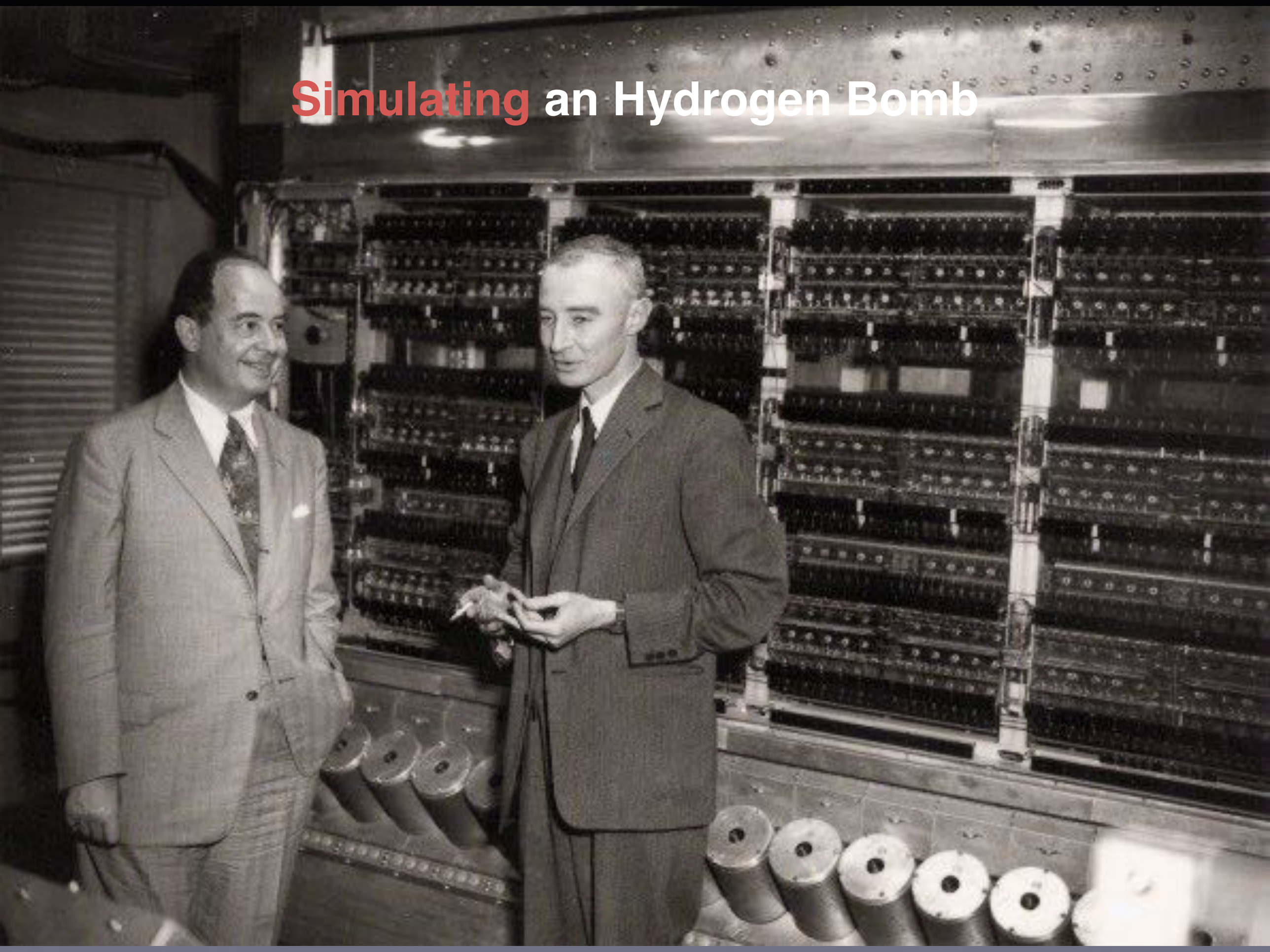


Software Crisis of Synthetic Biology (2015)
Valverde, Porcar, Pereto, Solé

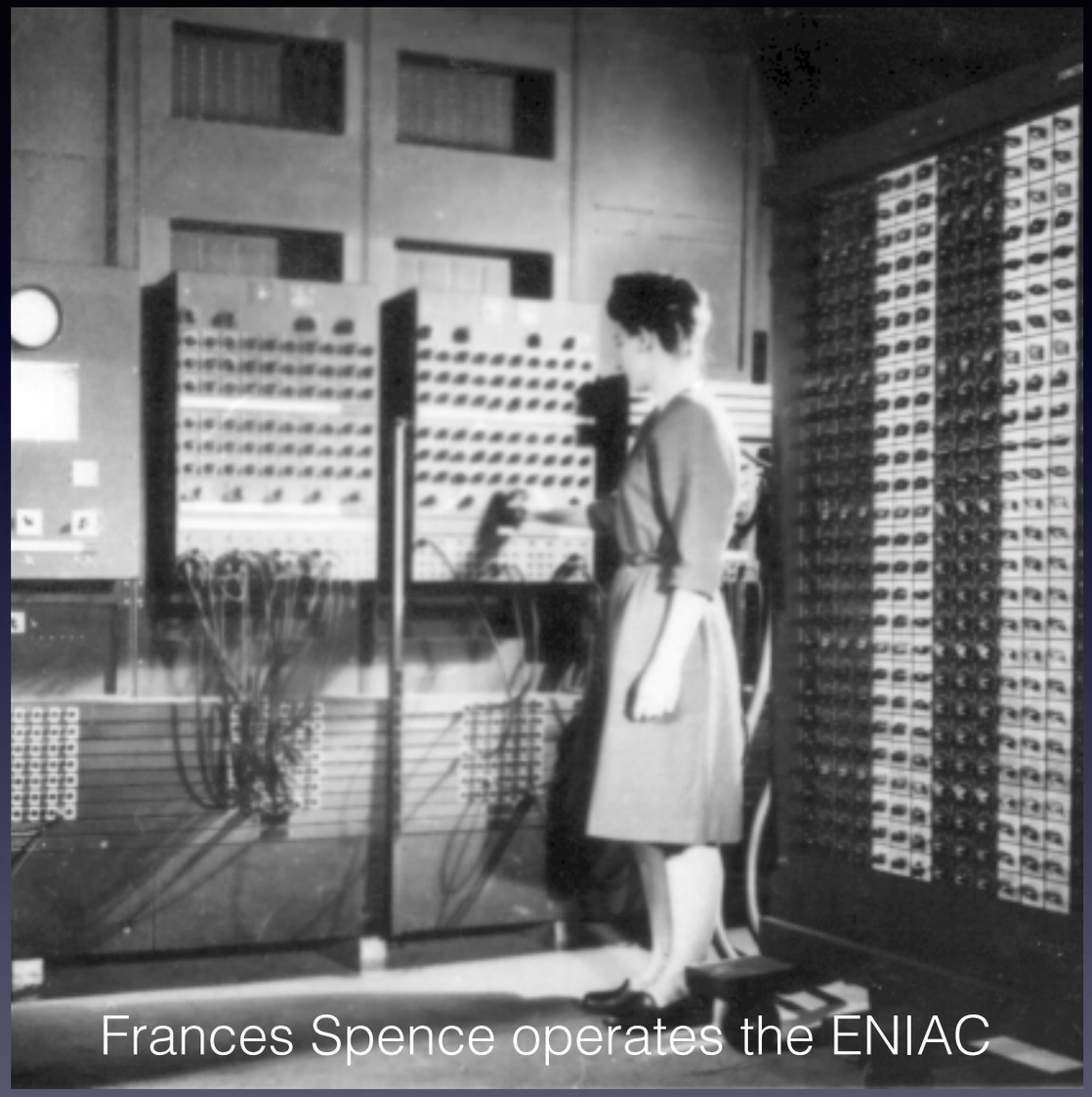
I.

Invention

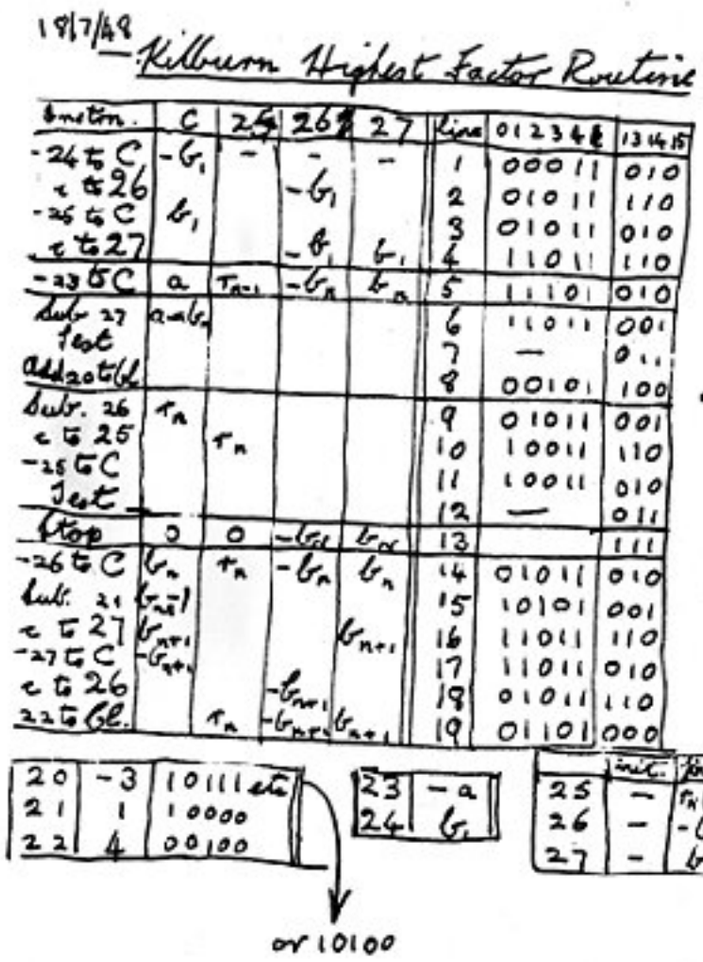
Simulating an Hydrogen Bomb



The Computer and the Program



Frances Spence operates the ENIAC



19/7/49 — Kilburn Highest Factor Routine (amended) —

Instruction	C	25	26	27	Line	0 1 2 3 4 5	13 14 15
-24 to C	$-b_1$	–	–	–	1	00011	010
+ to 26			$-b_1$		2	01011	110
-25 to C	b_1				3	01011	010
+ to 27			$-b_1$	b_1	4	11011	110
-23 to C	a	τ_{n-1}	$-b_n$	b_n	5	11101	010
Subr 27 test	$a-b_n$				6	11011	001
Add 2 to [6]					7	—	011
					8	00101	100 or 000
Subr. 26	τ_n				9	01011	001
+ to 25		τ_n			10	10011	110
-25 to C					11	10011	010
Test					12	—	011
Stop	0	0	$-b_{n+1}$	b_{n+1}	13		111
-26 to C	b_n	τ_n	$-b_n$	b_n	14	01011	010
Subr. 27	b_{n-1}				15	10101	001
+ to 27	b_{n+1}			b_{n+1}	16	11011	110
-27 to C	$-b_{n+1}$				17	11011	010
+ to 26			$-b_{n+1}$		18	01011	110
22 to [6]		τ_n	$-b_{n+2}$	b_{n+1}	19	01101	000

20	-3	10111 etc
21	1	10000
22	4	00100

23	$-a$
24	b_1

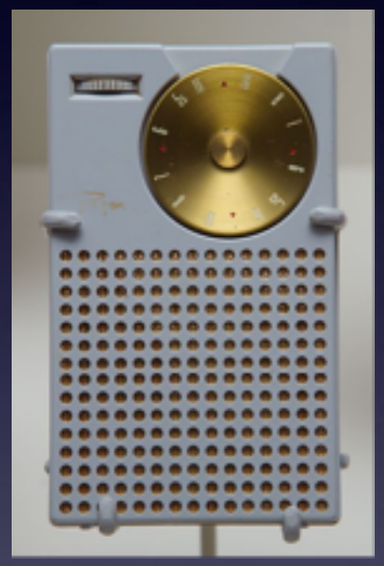
	init.	final
25	—	τ_n (= 0)
26	—	$-b_N$
27	—	b_N

or 10100

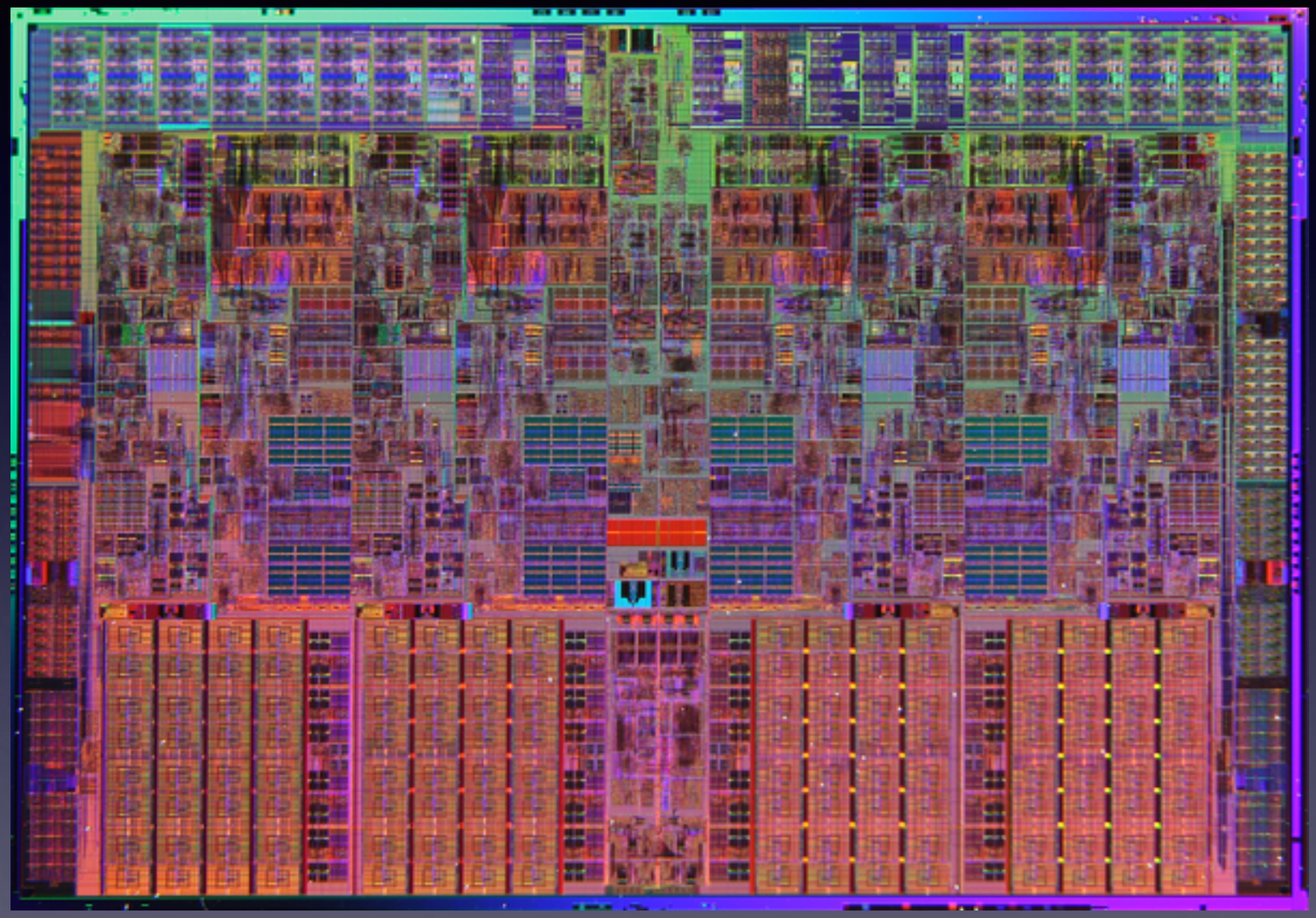
Kilburn Highest Factor Routine (U. of Manchester, 1948)

Stored-program concept: Storage of instructions in computer memory enables it to perform a large variety of tasks

Accelerated Growth



Regency TR-1
Transistor Radio
(4 TI Germanium
Transistors)

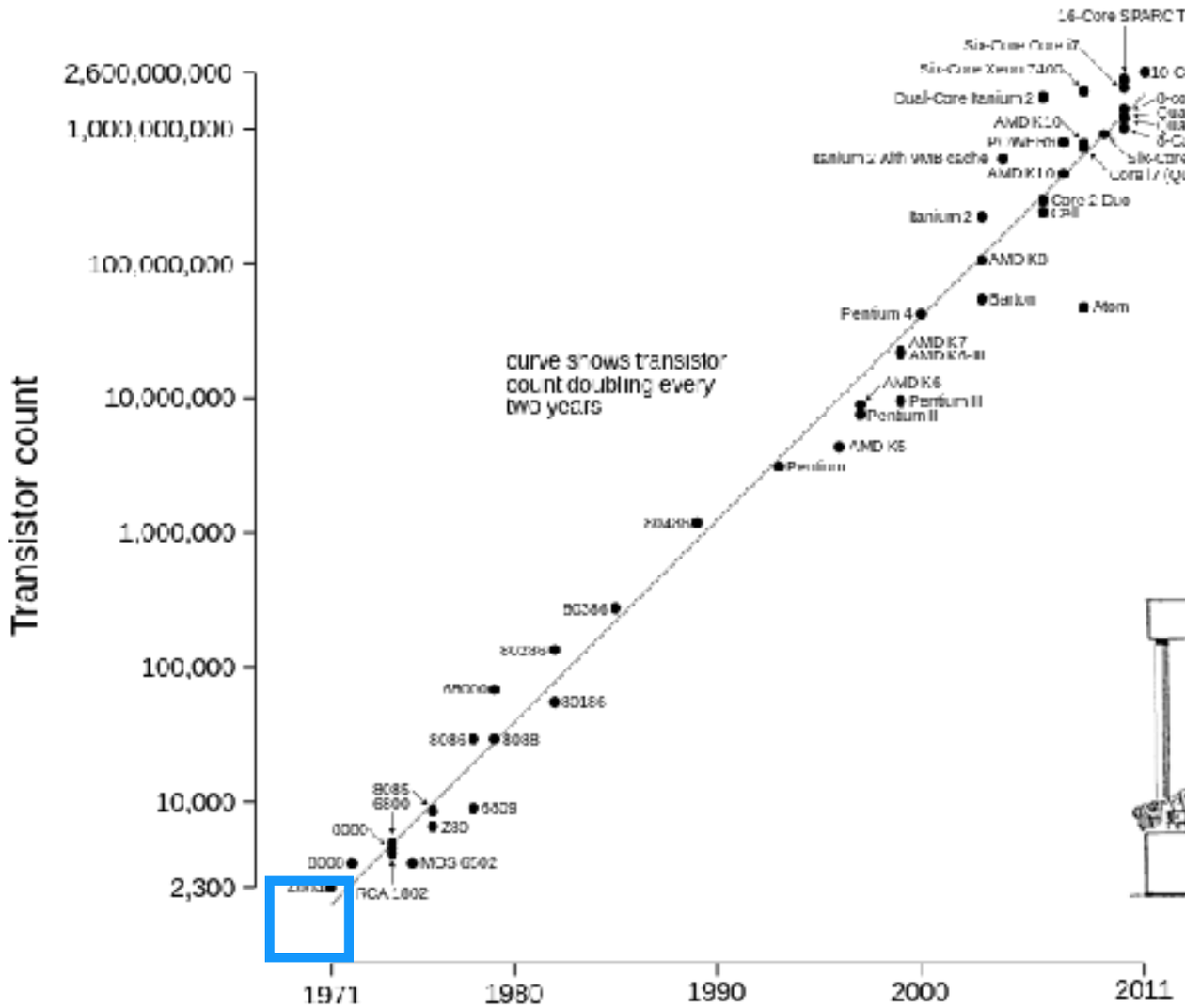


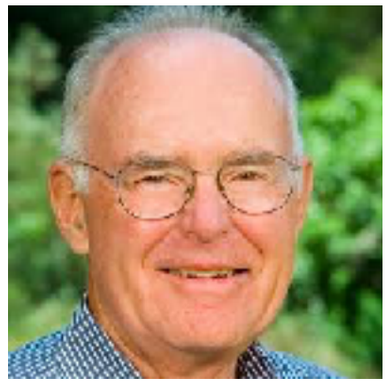
Intel Nehalem EX 8-core CPU (2009)
2,300,000,000 Transistors

Moore's Law

Technological performance improves exponentially with time

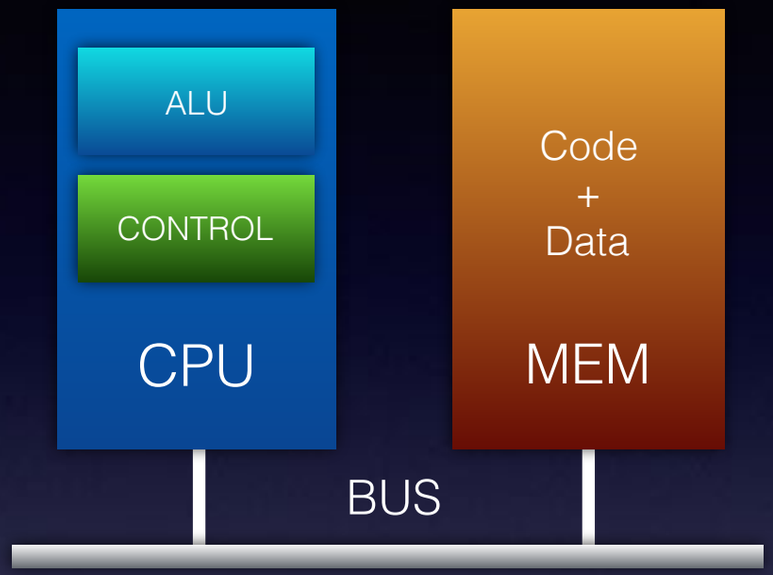
$$x(t) = exp(at)$$

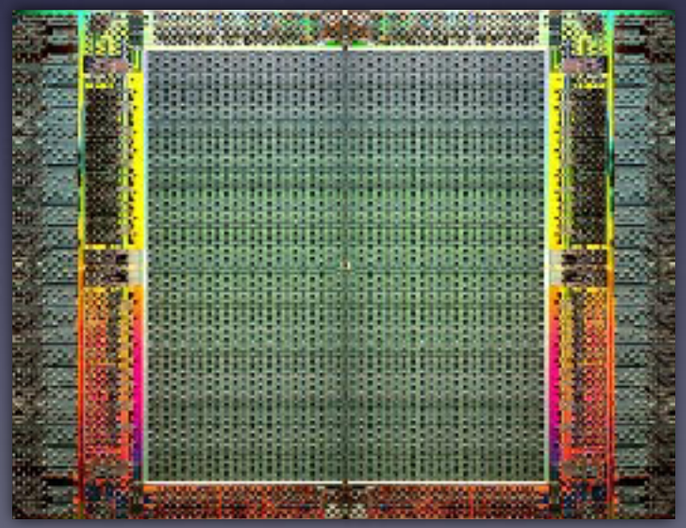


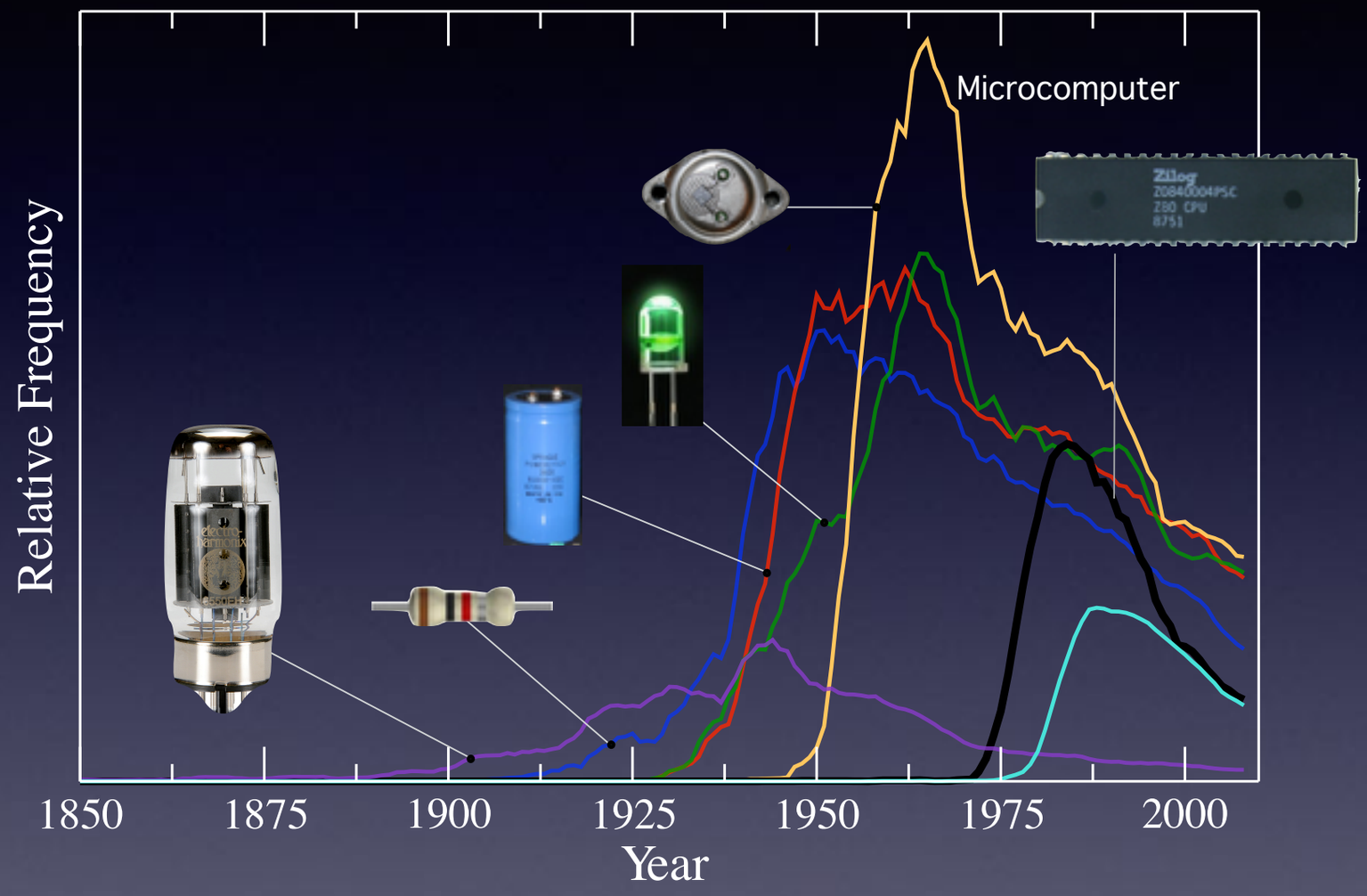


Gordon E. Moore

Modular Hardware



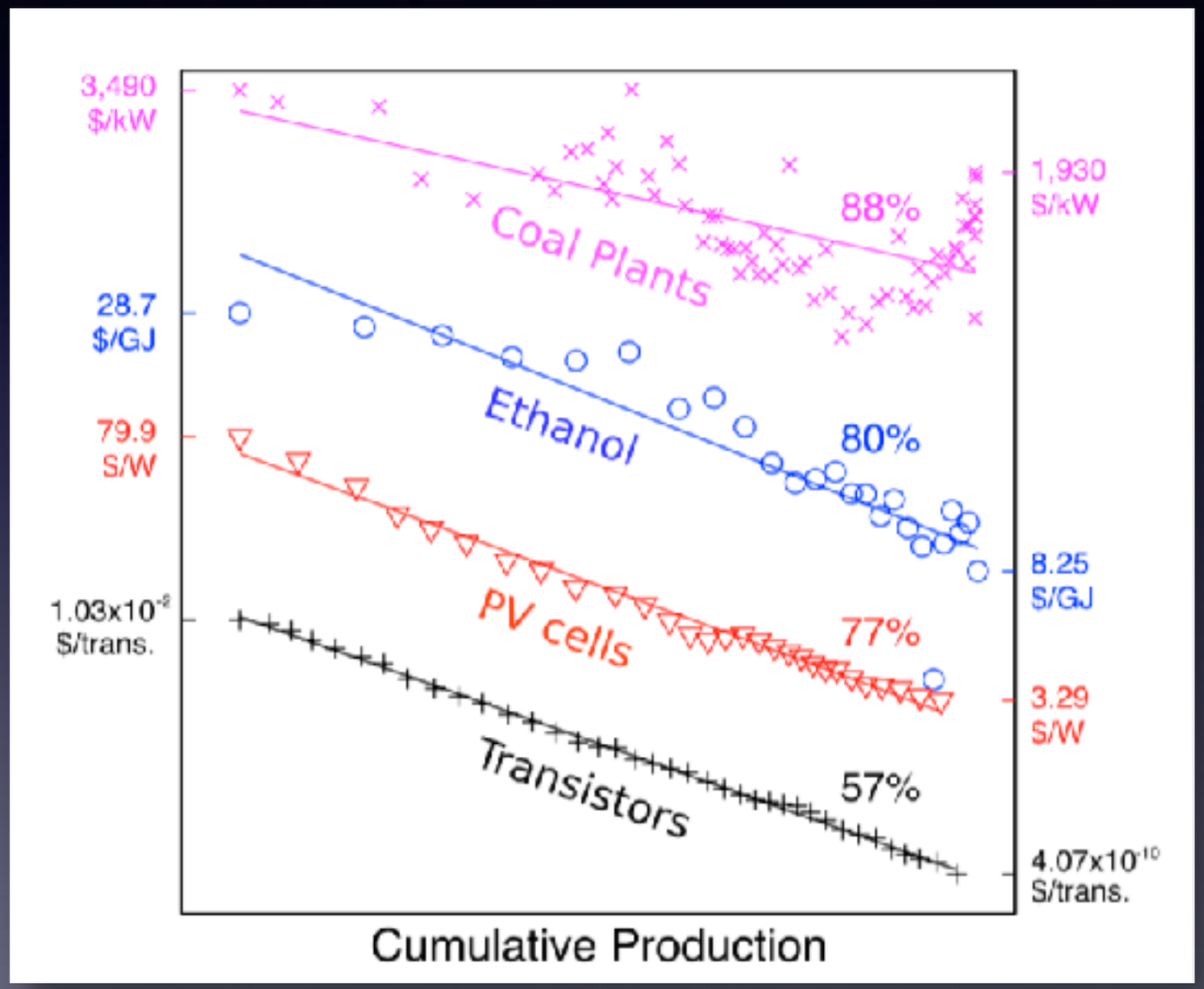




$$p(t) = (\mu + qF(t))(1 - F(t))$$

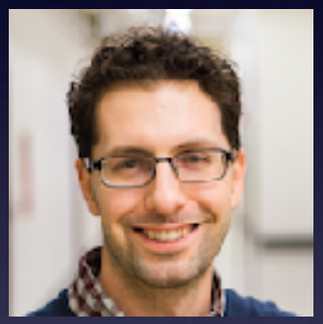
Performance Curves

Rise in efficiency occurs as a by-product of experience



$$C(z) \sim z^{-\alpha}$$

Jessika E. Trancik



James McNerney

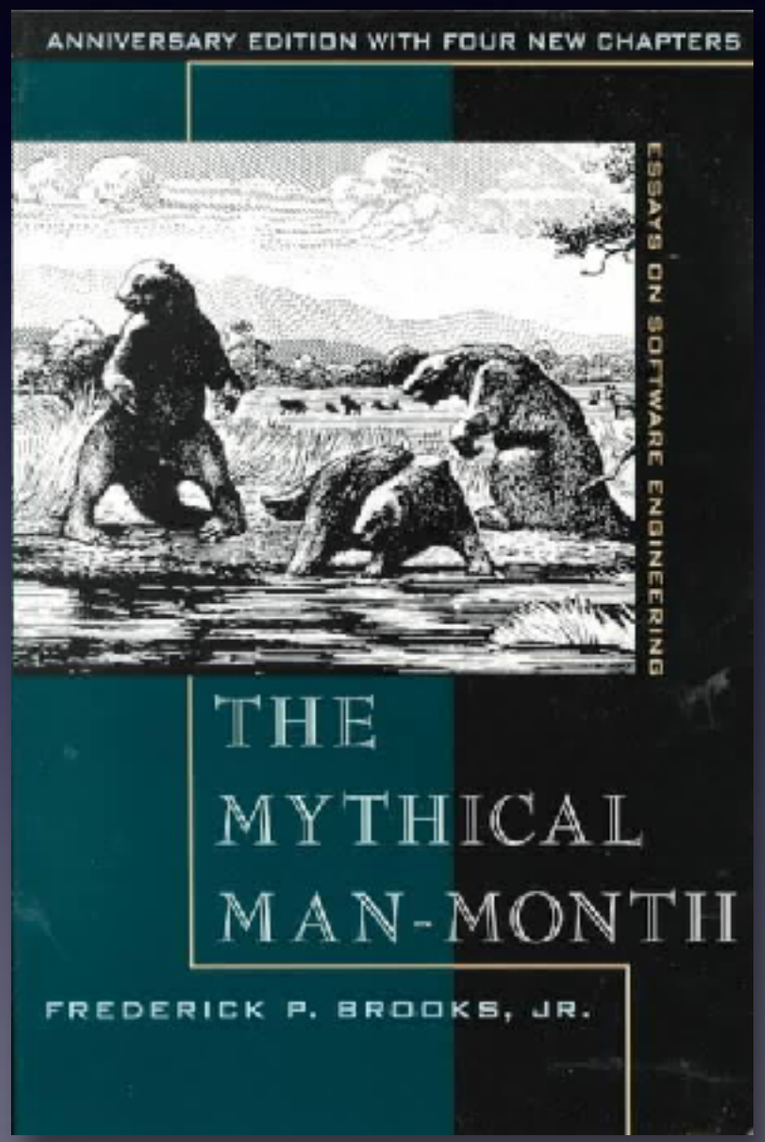
Bela Nagy

Doyne Farmer

Role of design complexity in technology improvement (2011)
McNerney, J.; Farmer, J. D.; Redner, S.; Trancik, J. E.

No Moore's law for Software

"The speed of software halves every 18 months"
- Gates' Law



Fred P. Brooks

Nicklaus E. Wirth

"There is no single development, in either technology or management technique, which by itself promises even one order of magnitude [tenfold] improvement within a decade in productivity, in reliability in simplicity"
- Fred Brooks

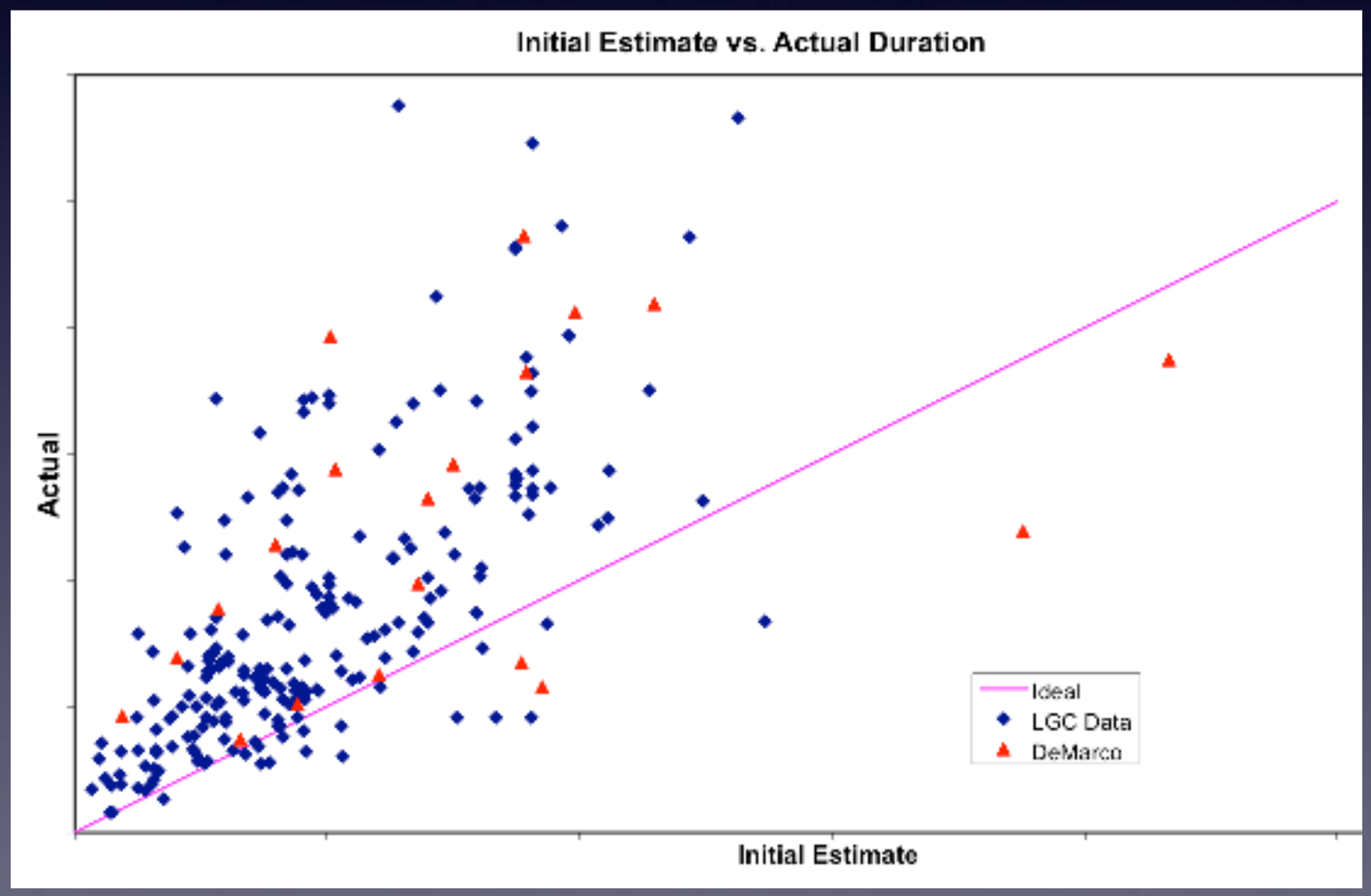
"…tools that are hard to make, and easy to screw up"
- Joe Henrich

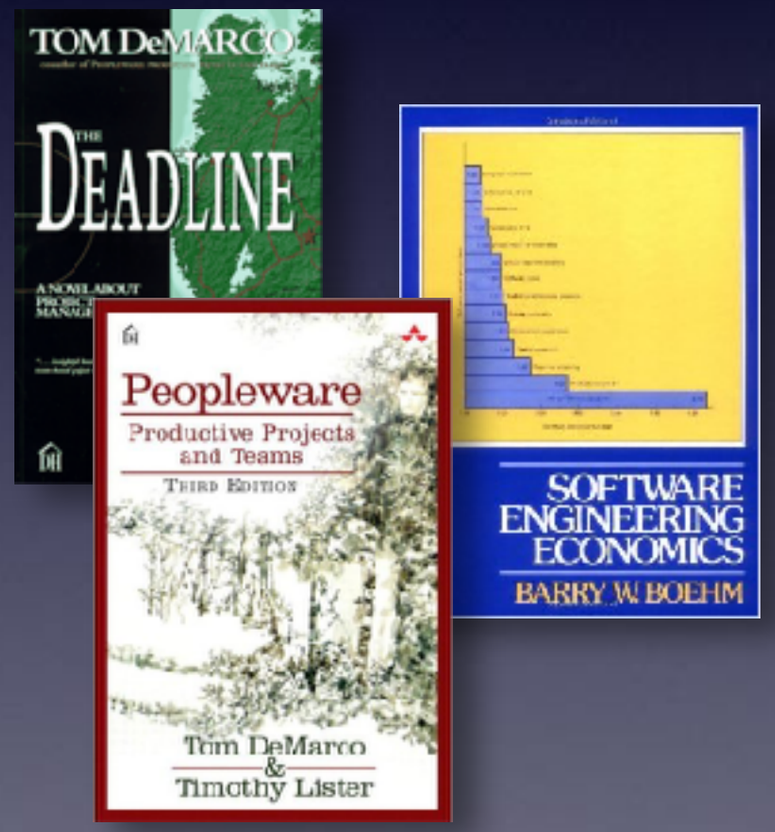
Software Estimation

"Making mistakes is in the nature of being human" - Douglas B. Lenat

"The major problems of our work are not so much ***technological as sociological*** *in nature" - Tom DeMarco*

Tom DeMarco

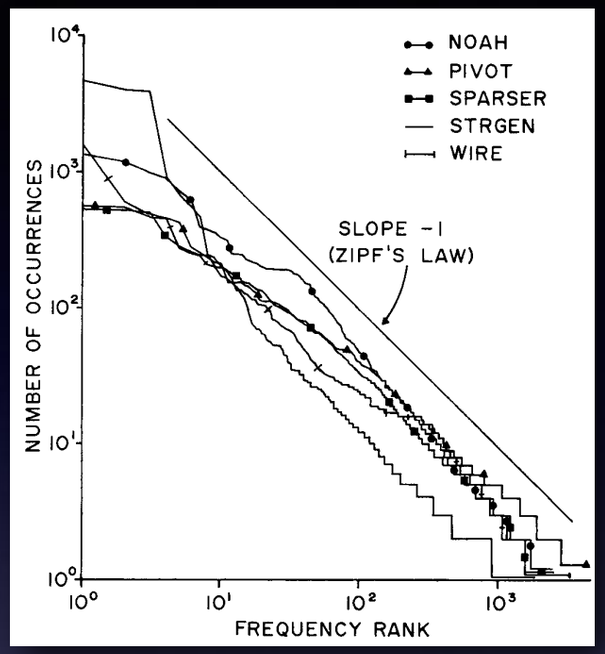




Scaling in Software

1977

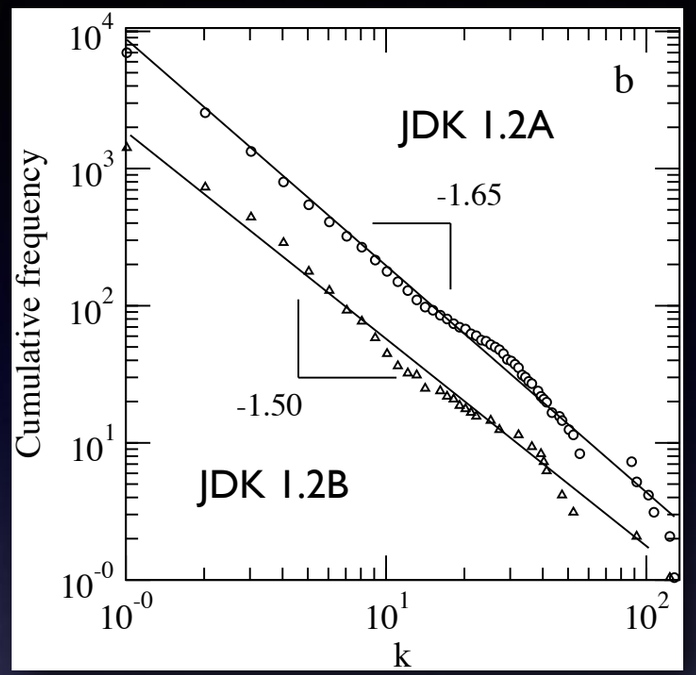
LISP atoms



2002

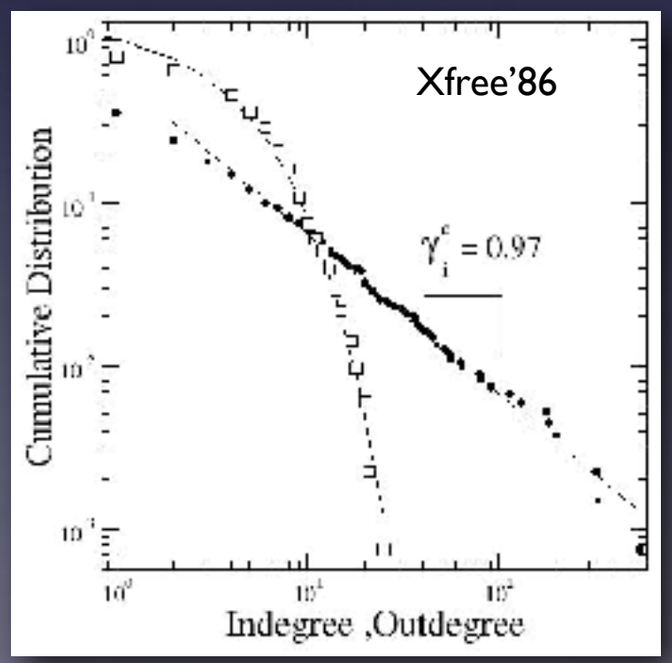
2003

Java, C++ undir



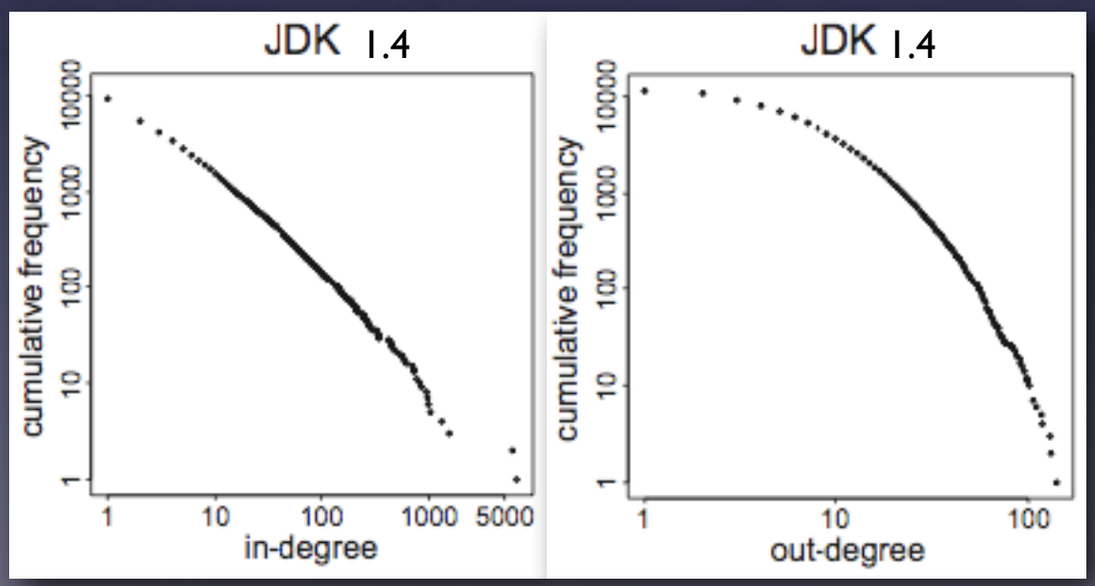
2005

C,C++ dir



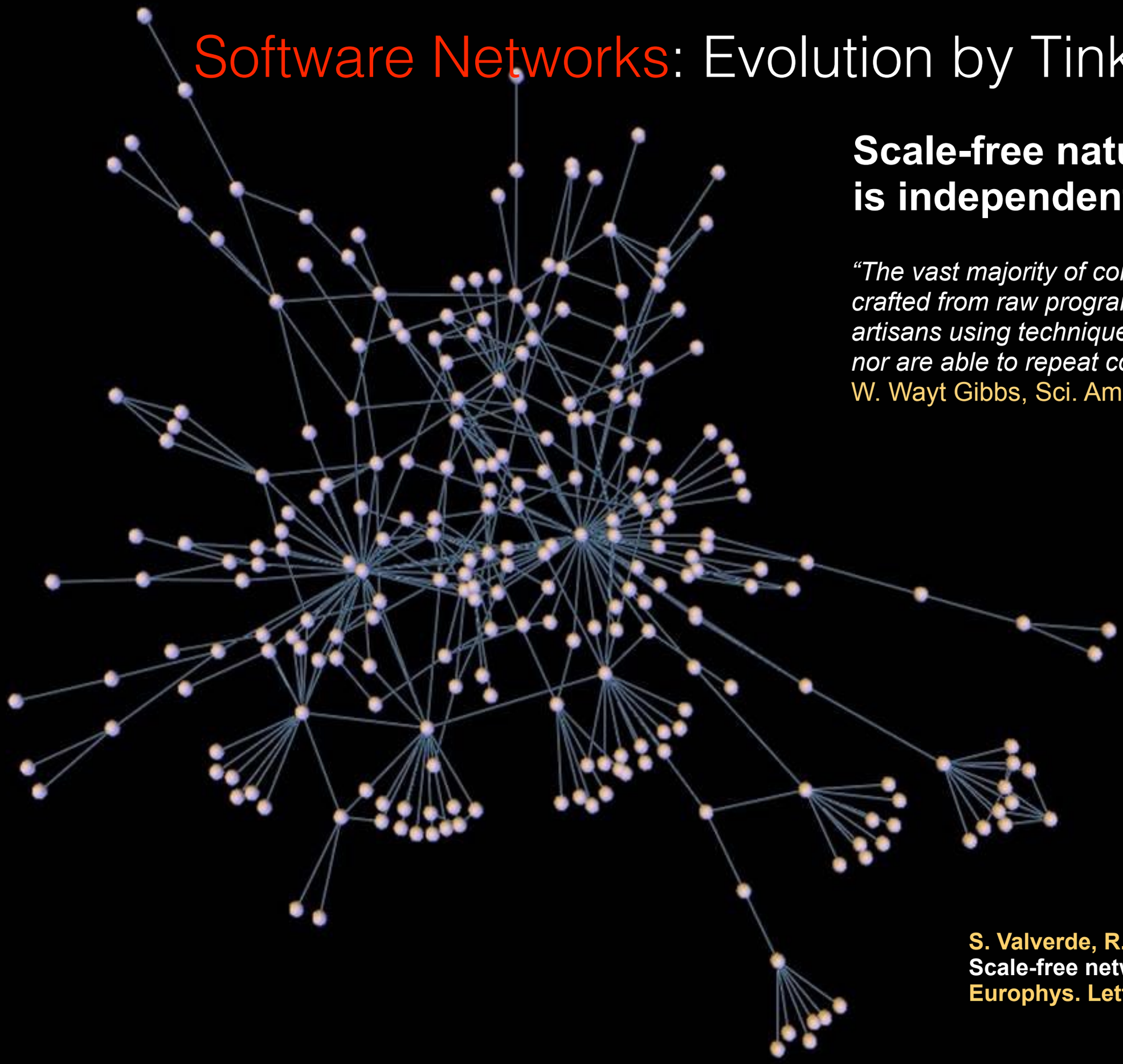
2008

Java dir



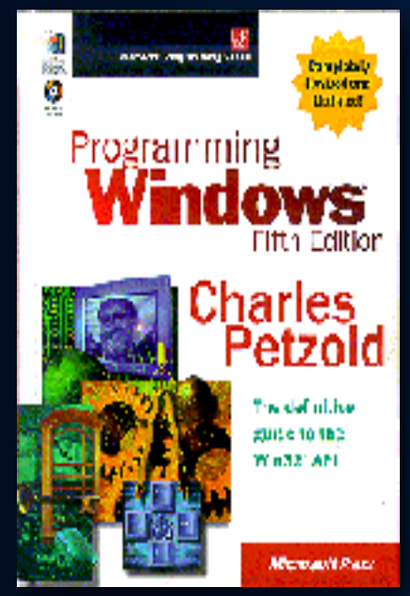
S. Valverde, R. Ferrer and R. V. Solé
Scale-free networks from optimal design
Europhys. Lett. 60 (2002) 512-517

Software Networks: Evolution by Tinkering



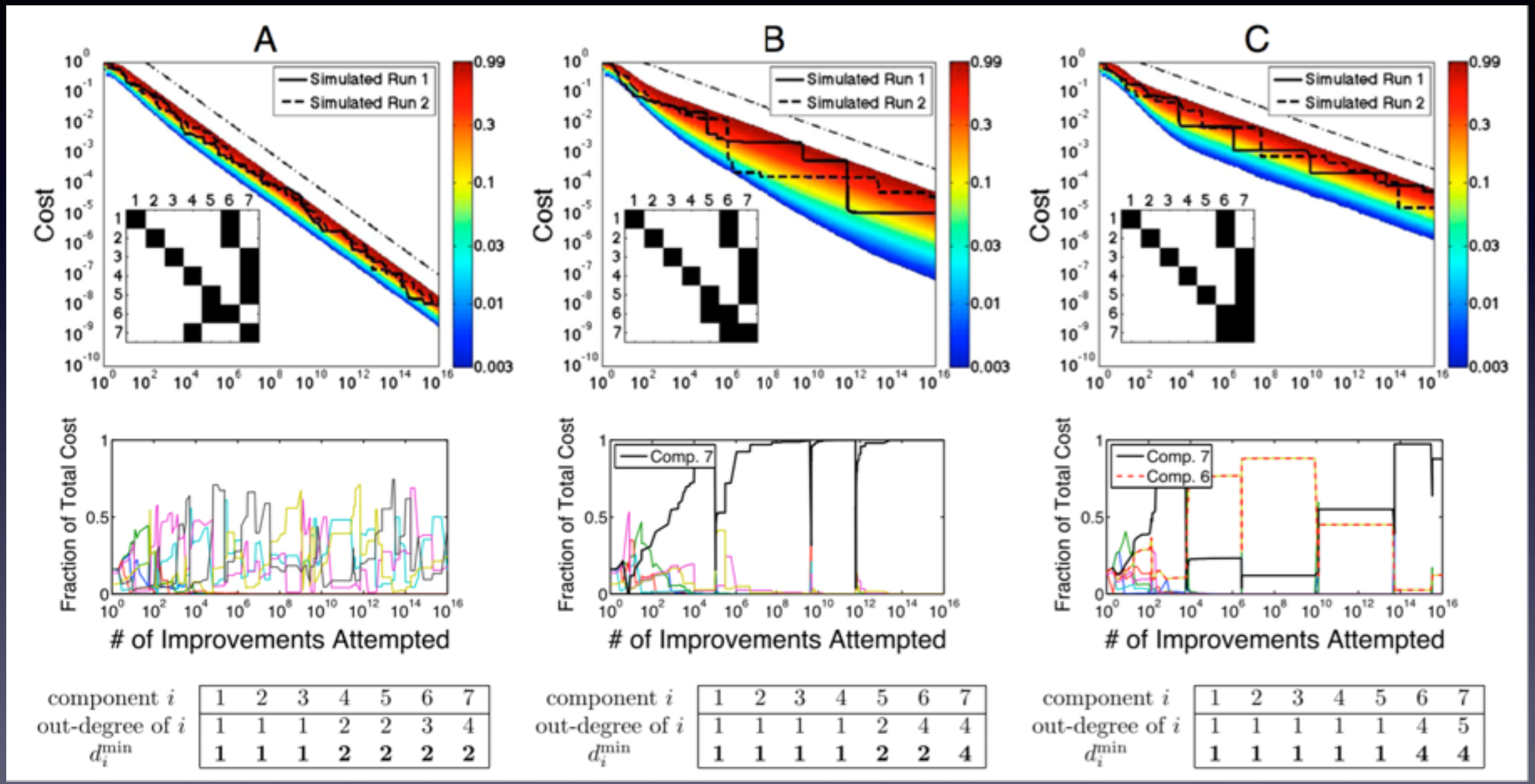
Scale-free nature of Software is independent from Function

"The vast majority of computer code is still hand-crafted from raw programming languages by artisans using techniques they neither measure nor are able to repeat consistently" –
W. Wayt Gibbs, Sci. Am, Sept. 1994



S. Valverde, R. Ferrer and R. V. Solé
Scale-free networks from optimal design
Europhys. Lett. 60 (2002) 512-517

Complexity Bottlenecks



component i	1	2	3	4	5	6	7
out-degree of i	1	1	1	2	2	3	4
$d_i^{\min}$	**1**	**1**	**1**	**2**	**2**	**2**	**2**

component i	1	2	3	4	5	6	7
out-degree of i	1	1	1	1	2	4	4
$d_i^{\min}$	**1**	**1**	**1**	**1**	**2**	**2**	**4**

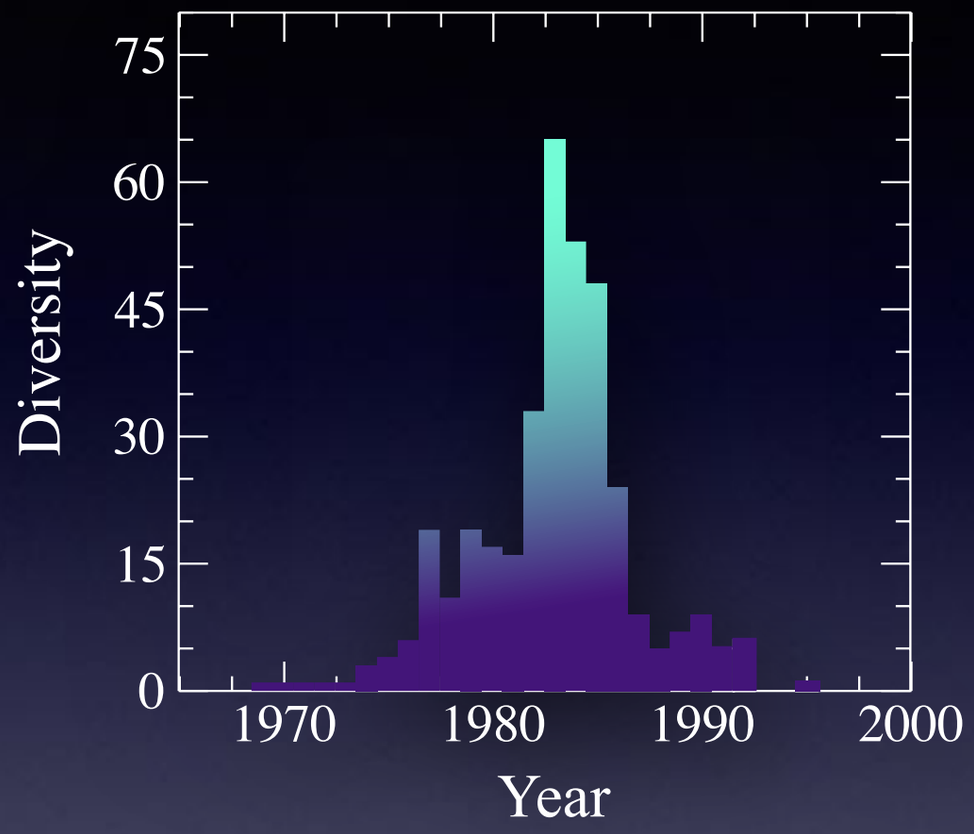
component i	1	2	3	4	5	6	7
out-degree of i	1	1	1	1	1	4	5
$d_i^{\min}$	**1**	**1**	**1**	**1**	**1**	**4**	**4**

Role of design complexity in technology improvement (2011)
McNerney, J.; Farmer, J. D.; Redner, S.; Trancik, J. E.

II.

Application

Personal Computers



III.

Diffusion

The **Evolution** of Programming Languages

COBOL


Sinclair Spectrum 48K
Acorn BBC BASIC Version 2.20
(C) Copyright R.T.Russell 1983
>O.
>L.
10 FOR A=32 TO 255
20 COLOUR A AND 7
30 COLOUR 135-(A AND 7)
40 VDU 32 AND (A=127),A
(A<>127)
50 NEXT
>RUN


python


julia


IN STRONG TYPING WE TRUST
Ada
1983 • 1995 • 2005


BYTE


Java™


R


Lua


Groovy


SQL®

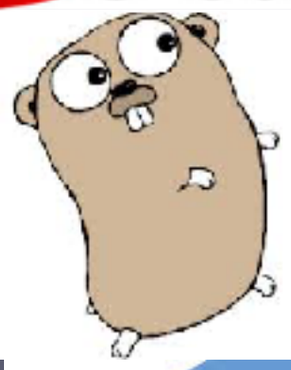

Fortran

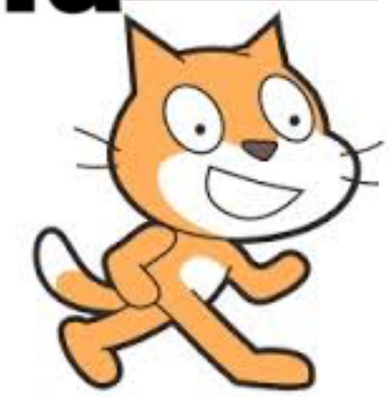

Wolfram
Mathematica


Ruby


I LIKE
APL

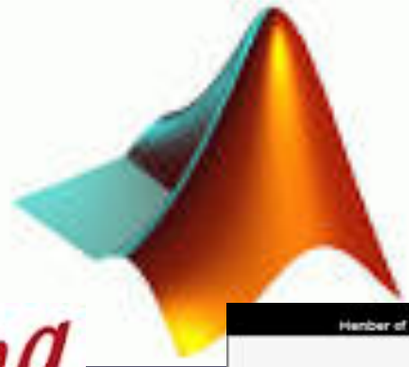

Scala






UNIFIED
MODELING
LANGUAGE


Perl



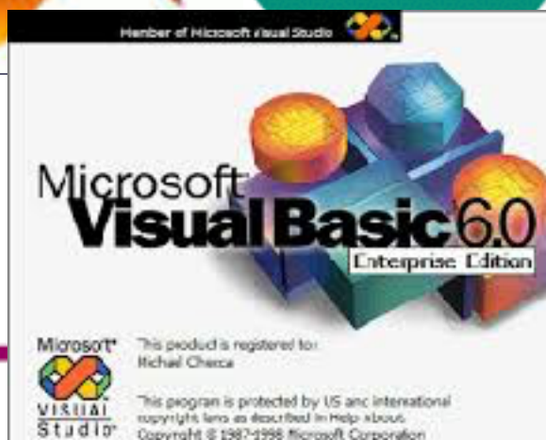

COBOL


C++

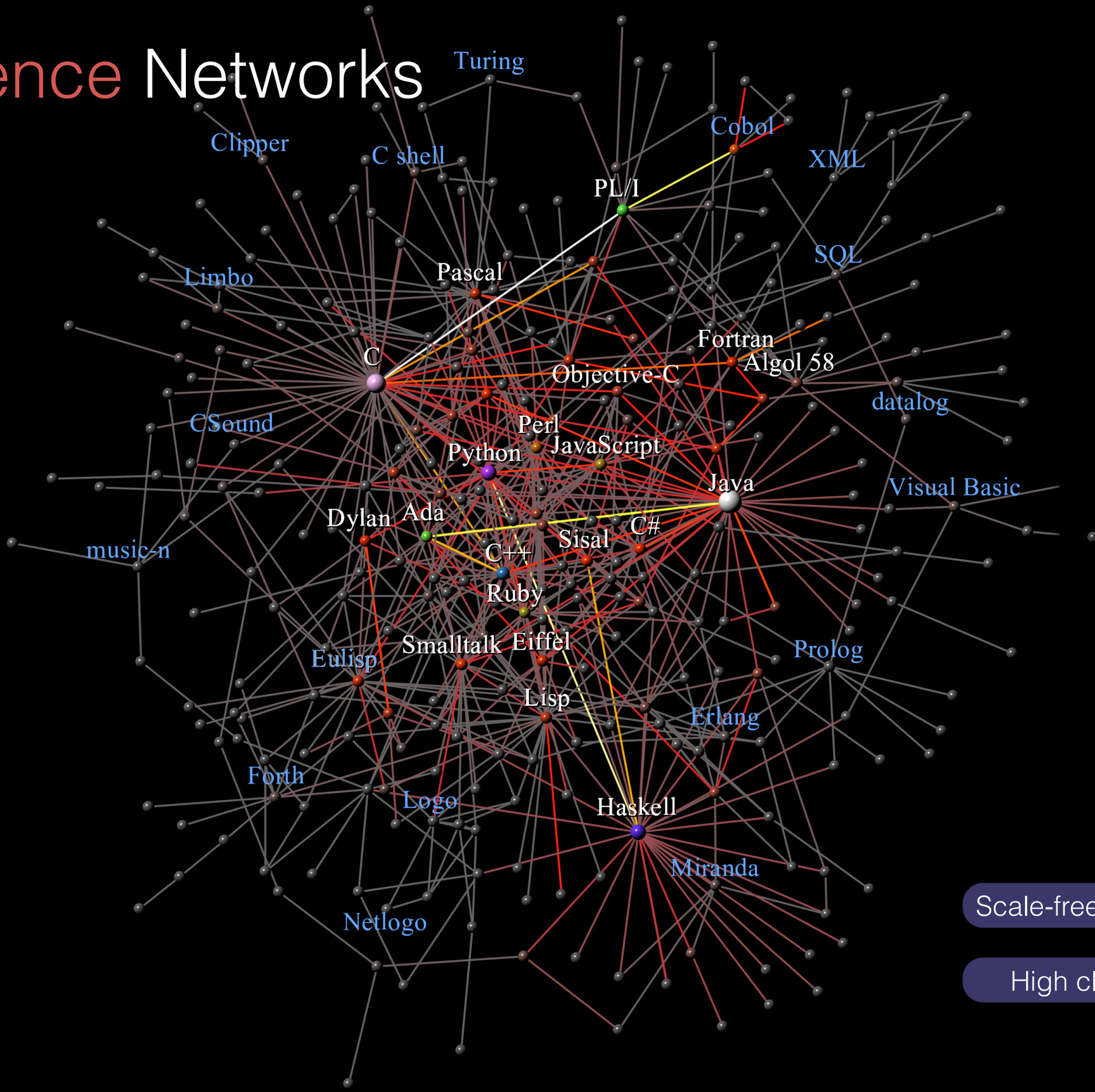

Swift

Eiffel


Erlang
PASCAL


Member of Microsoft Visual Studio
Microsoft
Visual Basic 6.0
Enterprise Edition
Microsoft
Visual Studio
This product is registered to:
Michael Chesca
This program is protected by US and international
copyright laws as described in Help About.
Copyright © 1987-1998 Microsoft Corporation

Influence Networks



Scale-free network

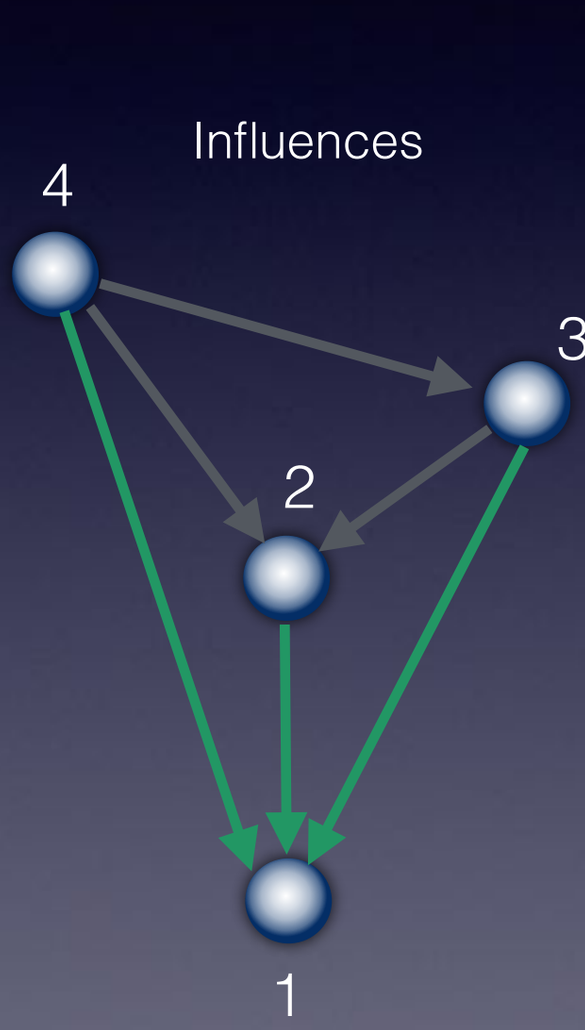
High clustering

WIKIPEDIA

Structural Similarity

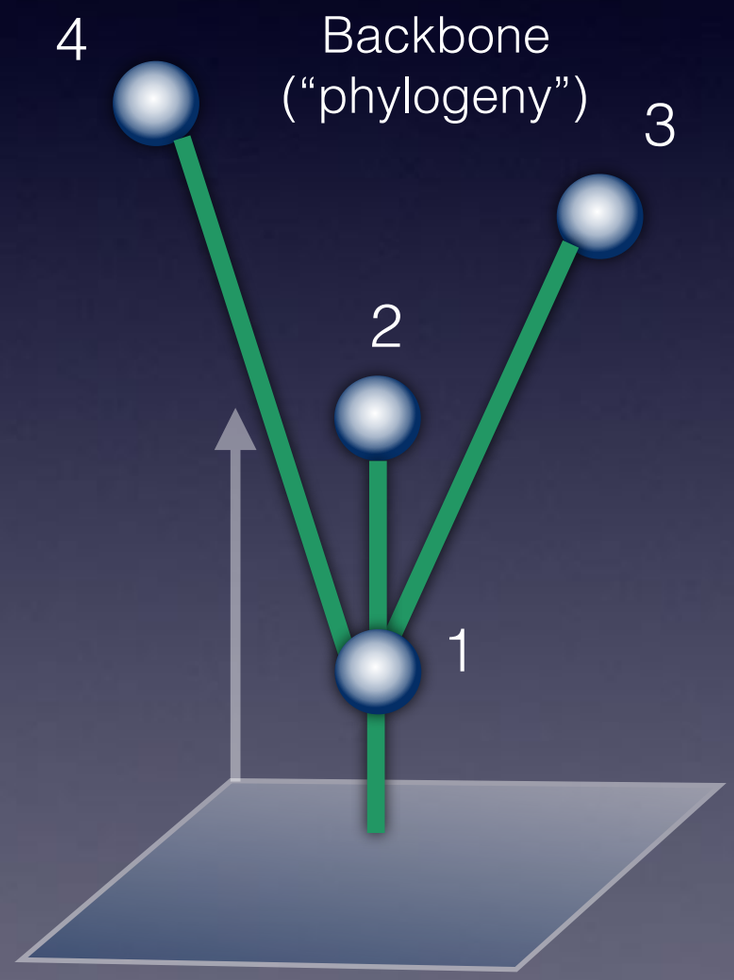
A structural method measures link strength by counting the number of paths visiting any pair of nodes in the influence network.

Derek J. de Solla Price
(1922 - 1983)



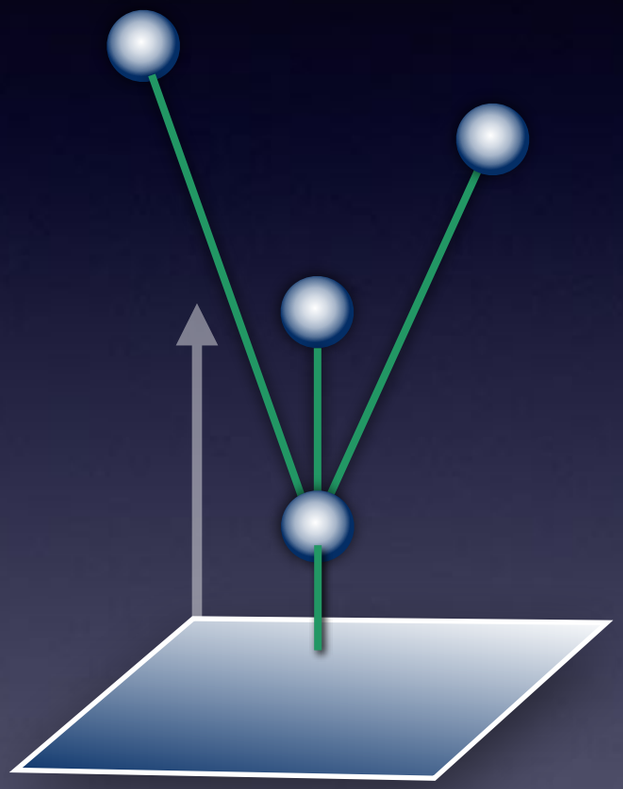
Impact

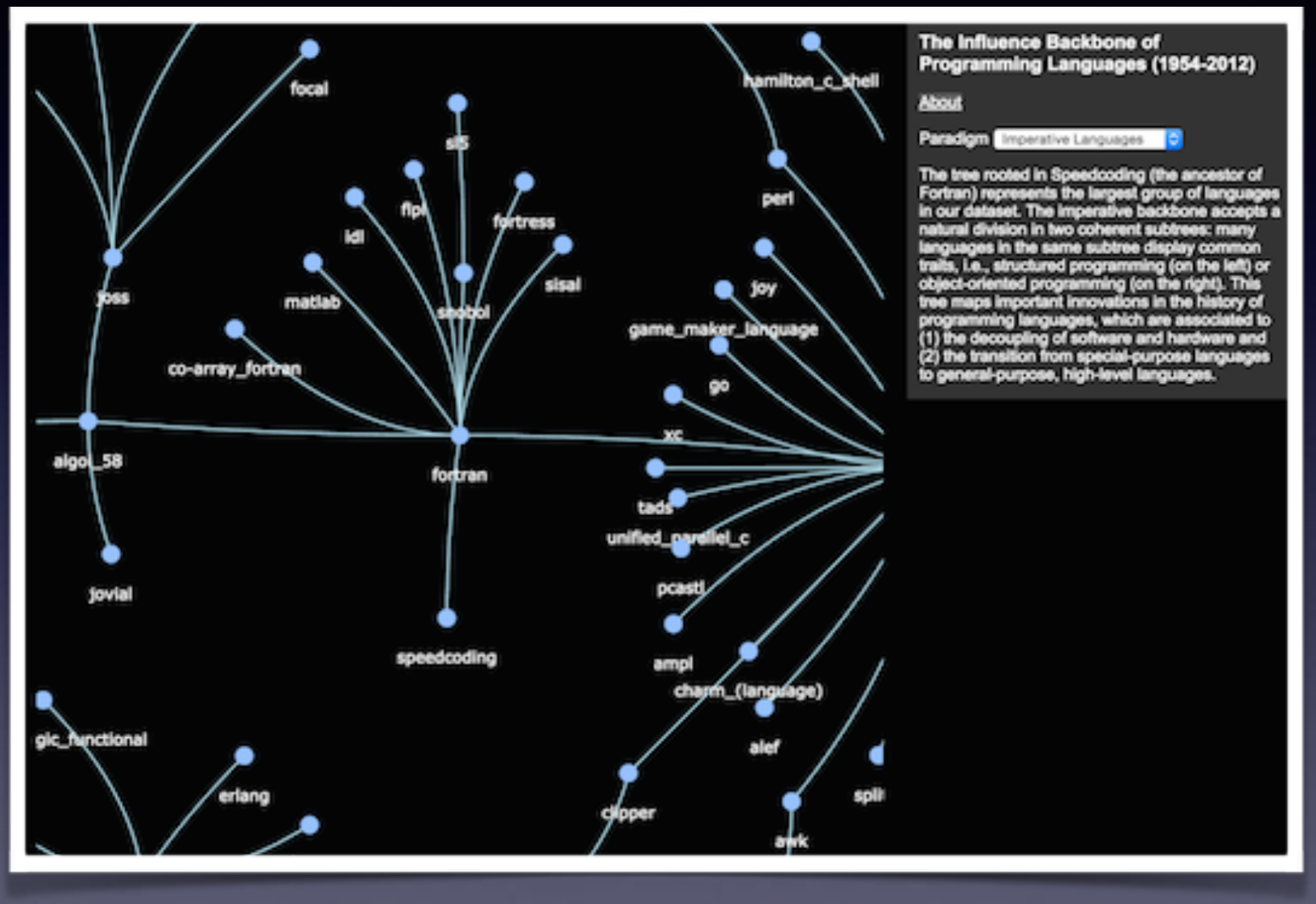
	1	2	3	4
1	0	0	0	0
2	0.3	0	0	0
3	0.38	0.13	0	0
4	0.37	0.2	0	0



$$I_{i\to\alpha} = \sum_{j\in\omega_\alpha - \{i\}} \lambda s_{ij}^{desc} + (1-\lambda) s_{ij}^{anc}$$

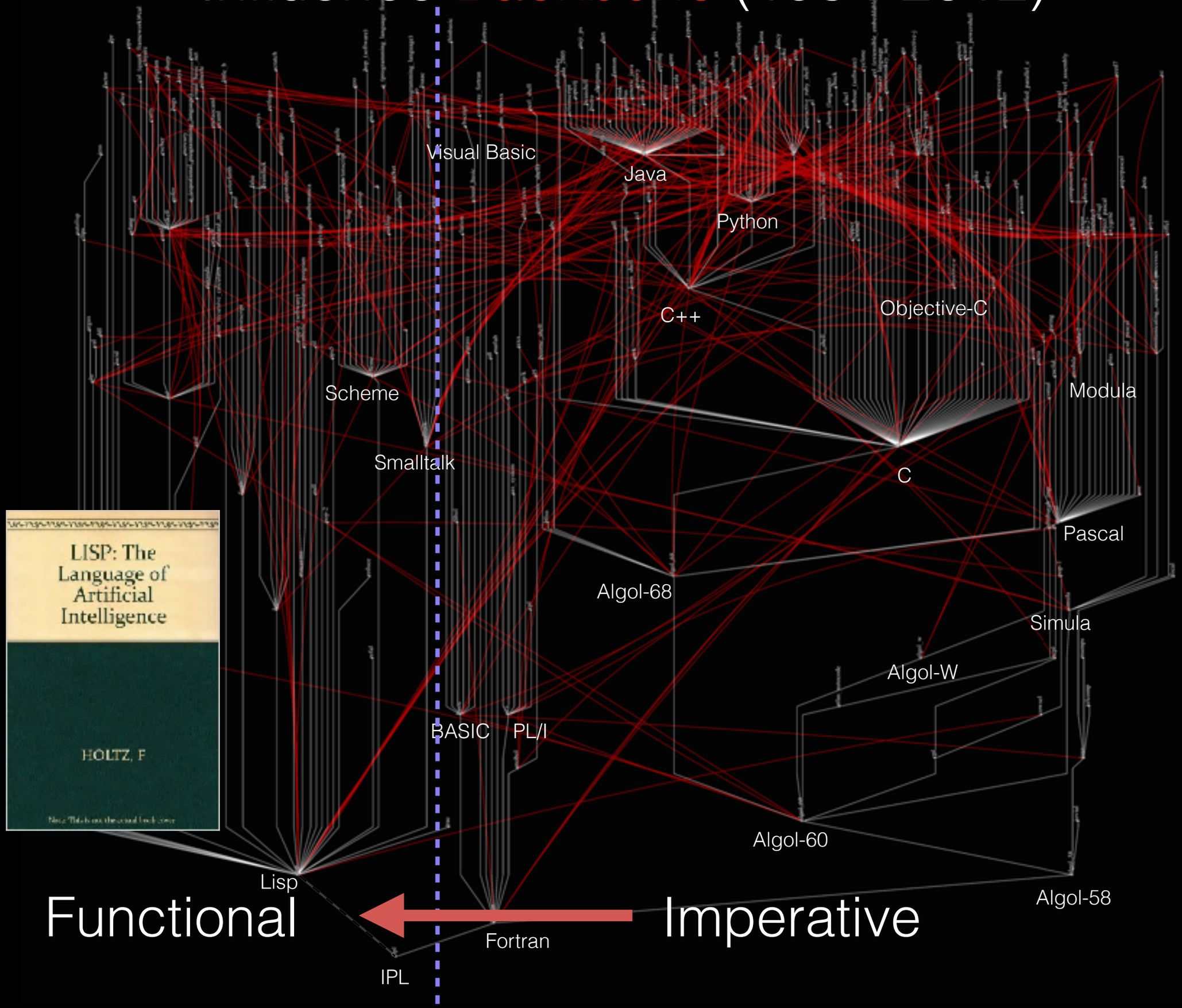
Influence Backbone Maps Vertical Transmission





http://svalver.github.io/Proglang/paradigms.html

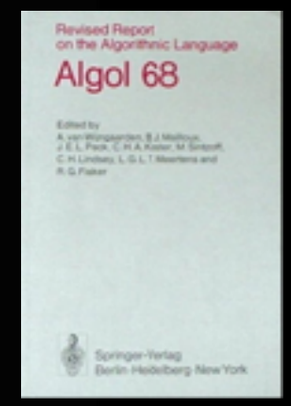
Influence Backbone (1954-2012)



Increasing Horizontal Recombination

Modern:
1970-Today

Early:
1953-1972



Visual Basic
Java
Python
C++
Objective-C
Modula
Scheme
Smalltalk
C
Pascal
Algol-68
Simula
Algol-W
BASIC
PL/I
Algol-60
Lisp
Algol-58
Fortran
IPL

(Valverde & Solé, 2015)


Sinclair Spectrum 48K
Acorn BBC BASIC Version 2.20
(C) Copyright R.T.Russell 1983
>O.
>L.
10 FOR A=32 TO 255
20 COLOUR A AND 7
30 COLOUR 135-(A AND 7)
40 VDU 32 AND (A=127),A
(A<>127)
50 NEXT
>RUN


python

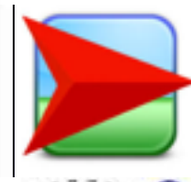

julia


IN STRONG TYPING WE TRUST
Ada
1983 • 1995 • 2005


BYTE
SMALLTALK


Java™


R

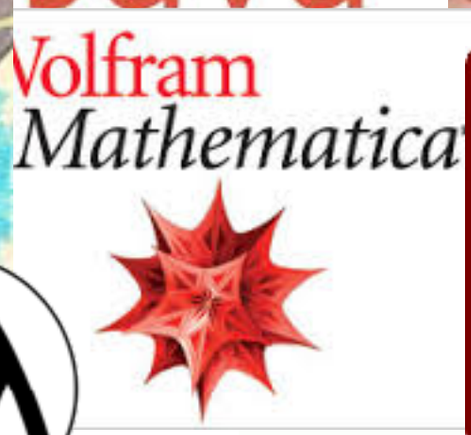



Groovy


Lua

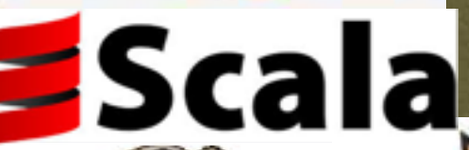

SQL®

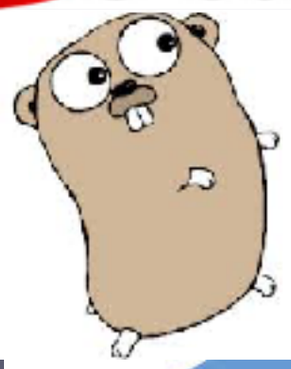

Fortran

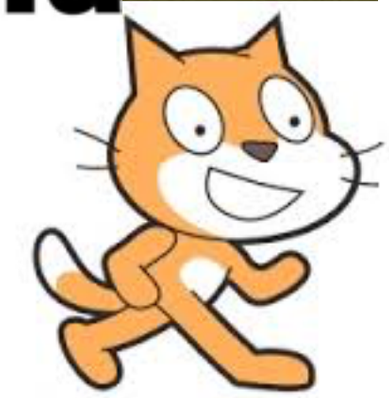

Wolfram
Mathematica


Ruby


I LIKE
APL

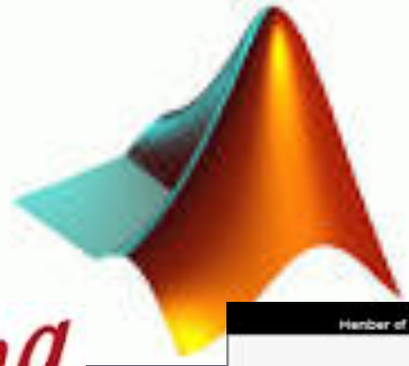

Scala






UNIFIED
MODELING
LANGUAGE


Perl



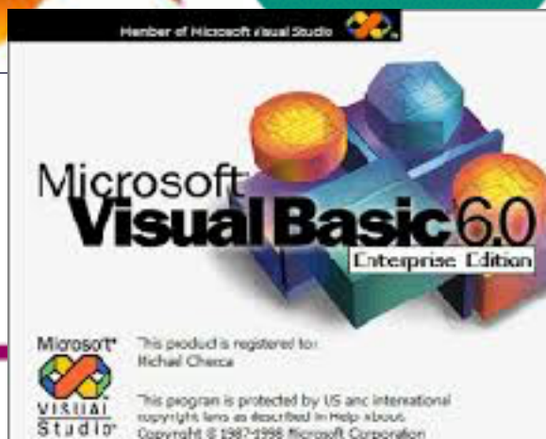

COBOL


C++


Swift

Eiffel


Erlang
PASCAL


Member of Microsoft Visual Studio
Microsoft
Visual Basic 6.0
Enterprise Edition
Microsoft
Visual Studio
This product is registered to:
Michael Chesca
This program is protected by US and international
copyright laws as described in Help About.
Copyright © 1987-1998 Microsoft Corporation


python


Java

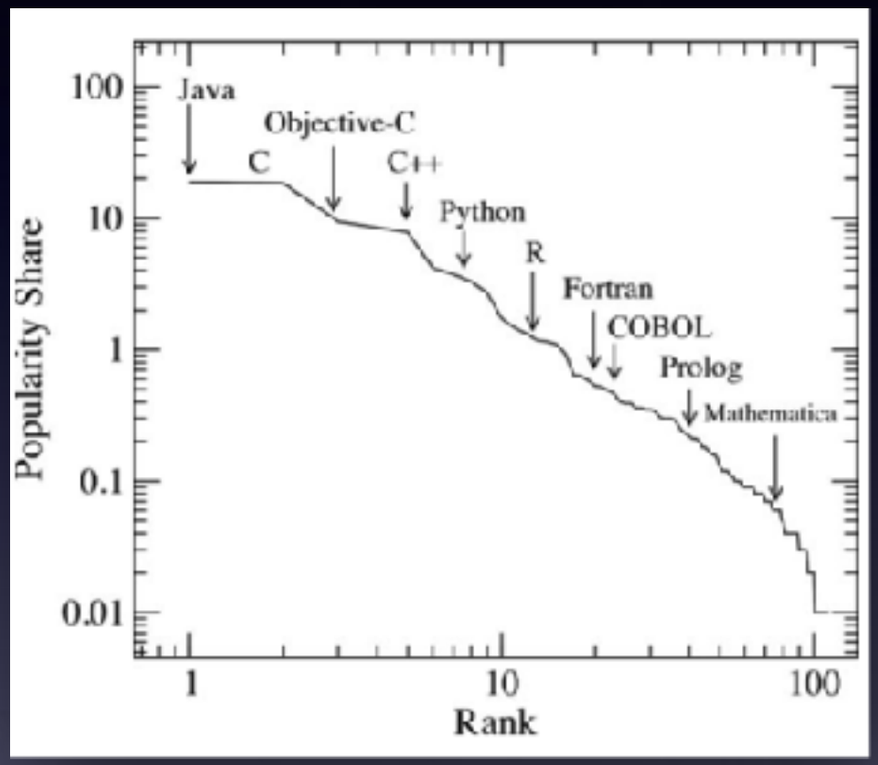

Fortran


COBOL

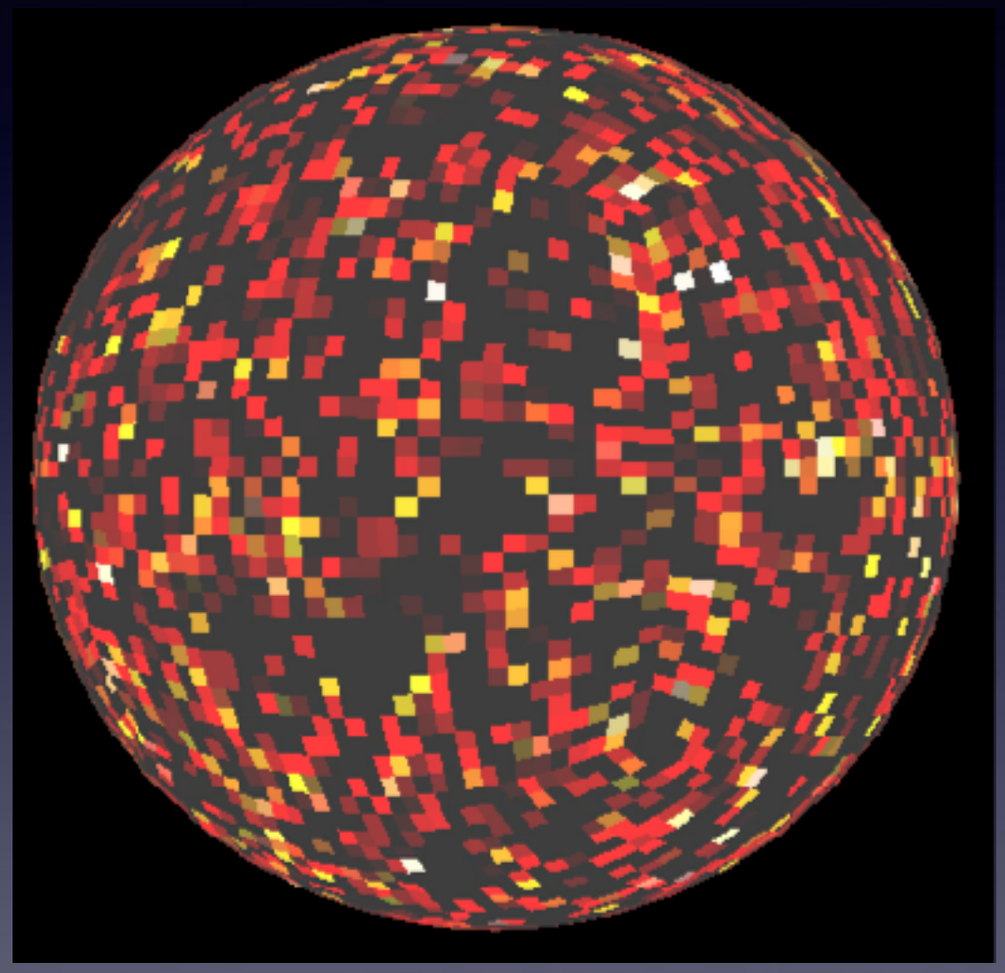

++

Globalisation and Diversity

As globalised communication and increasing returns accelerate, few PLs will survive.



$$f(r) = \frac{A}{r^a}(R + 1 - r)^b$$



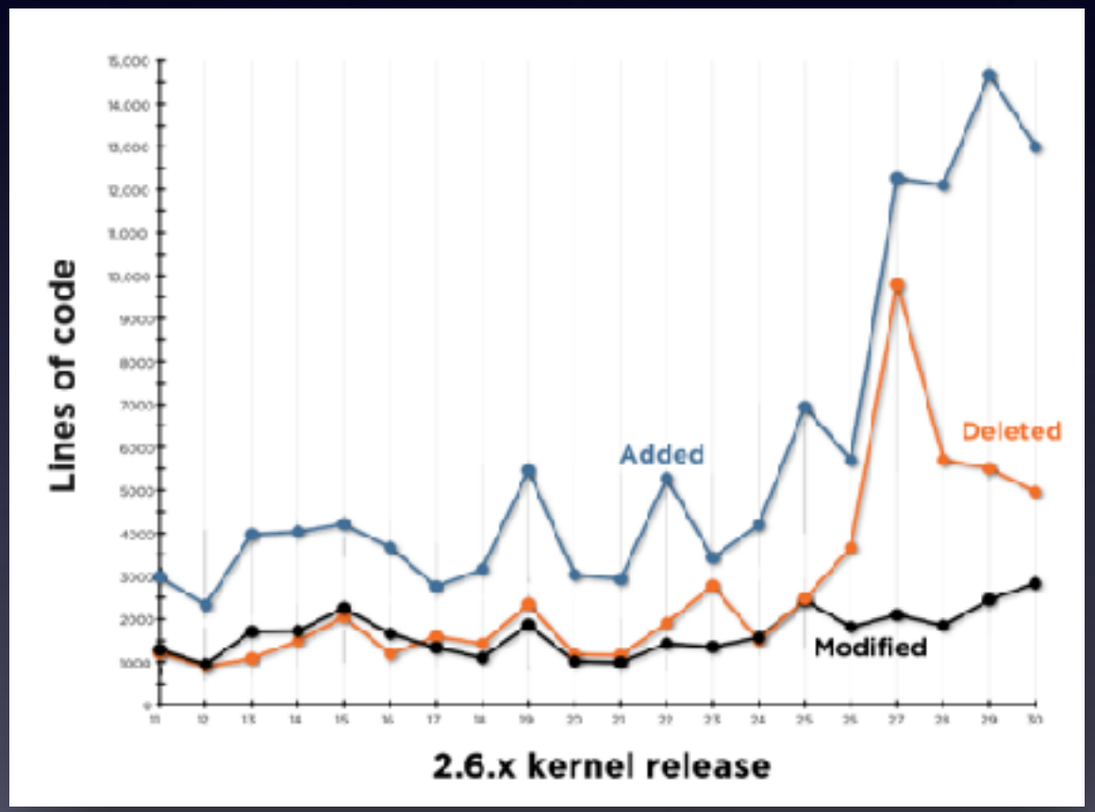
A Cultural Diffusion Model for the Rise and Fall of Programming Languages
S. Valverde & R. V. Solé (2015) Human Biology 87 (3), 88

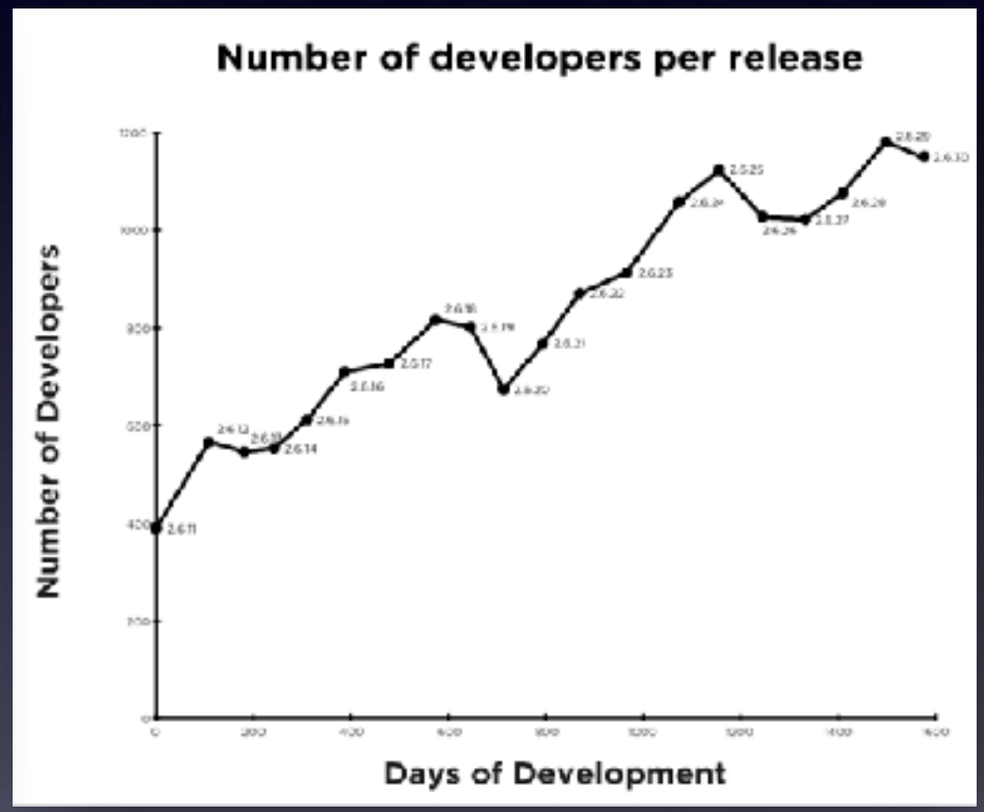
Open-Source Software...

...and Closed-Source Software

The Growth of Linux

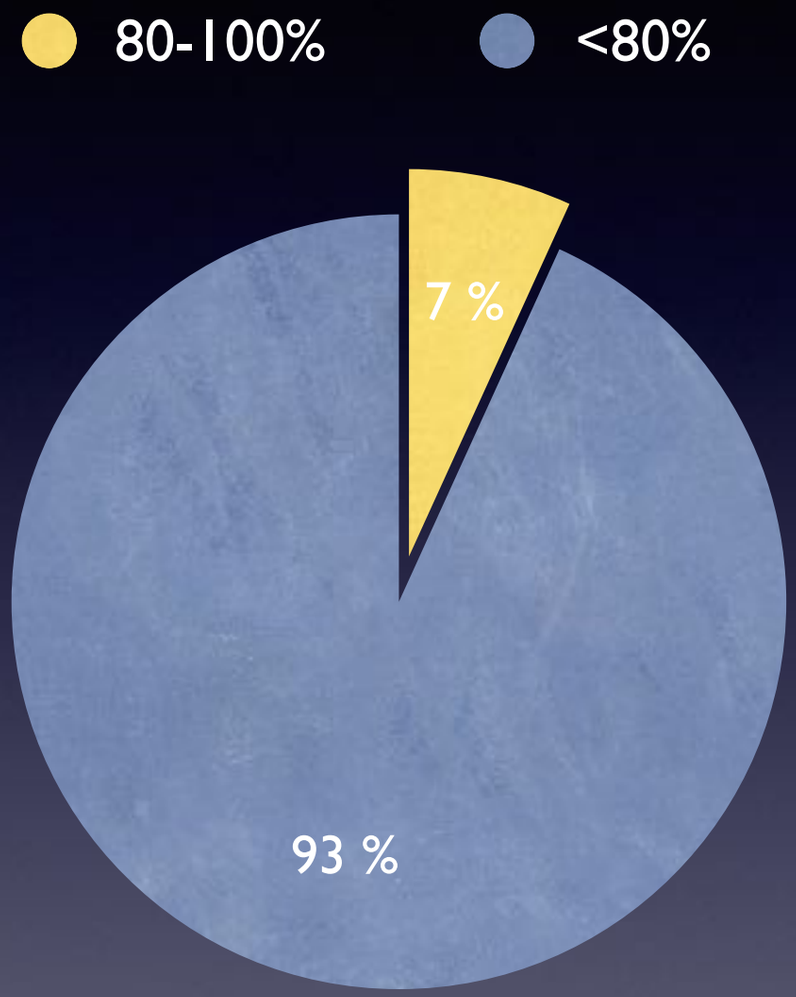
"Computer software develops more effectively when its code is freely accessible to all"





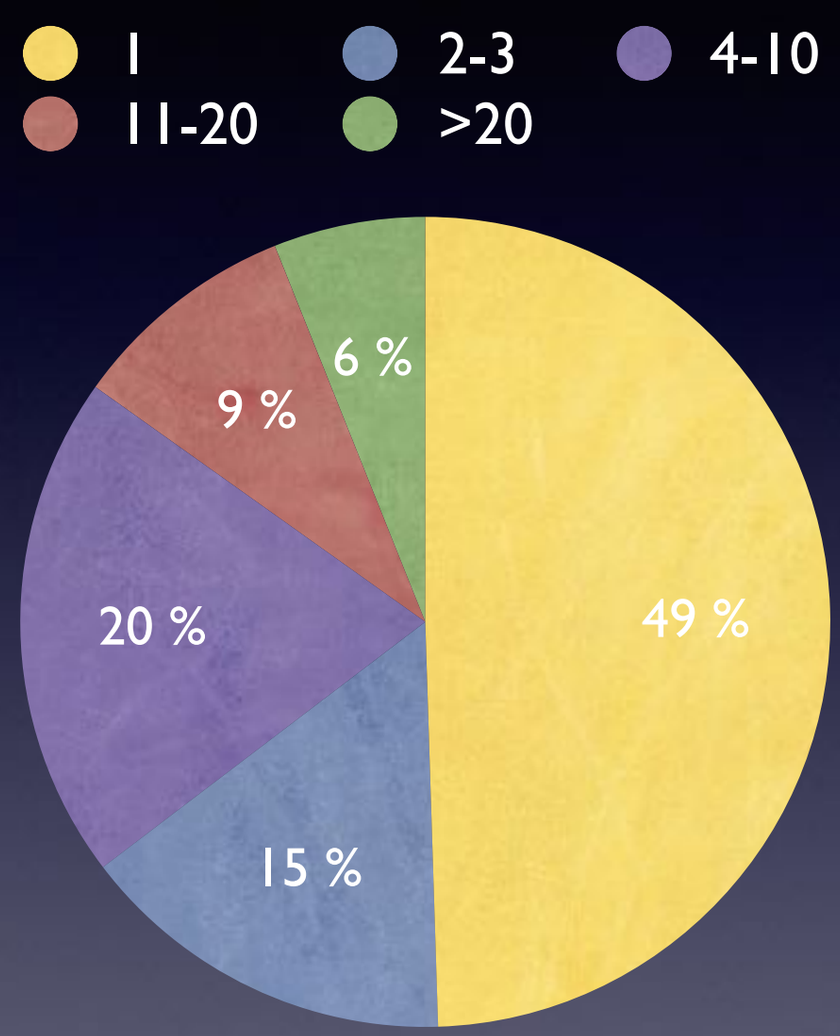
Source: Linux Foundation

Open Source Success



Activity Index

+ at least one file released in the last 6 months



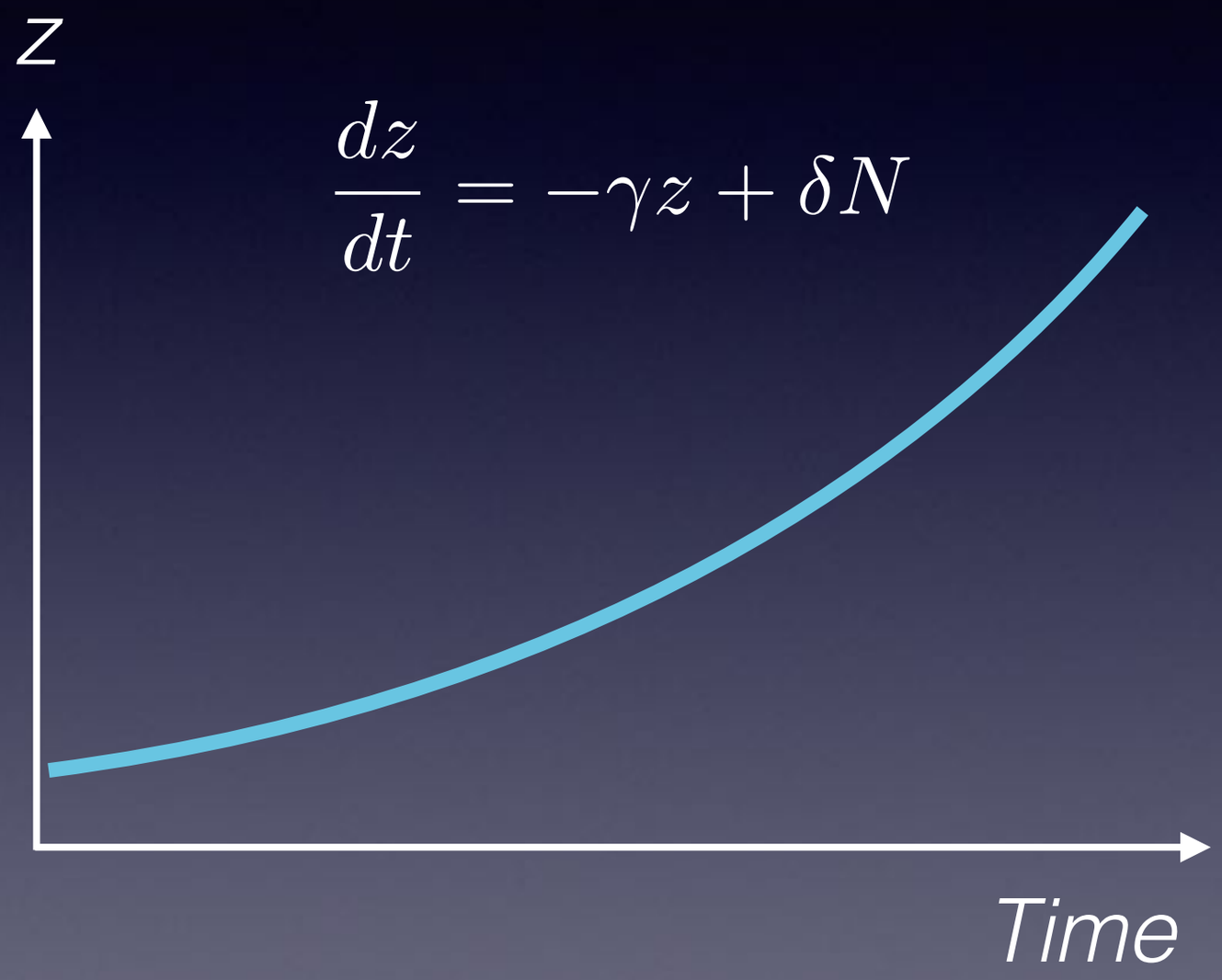
Team Size

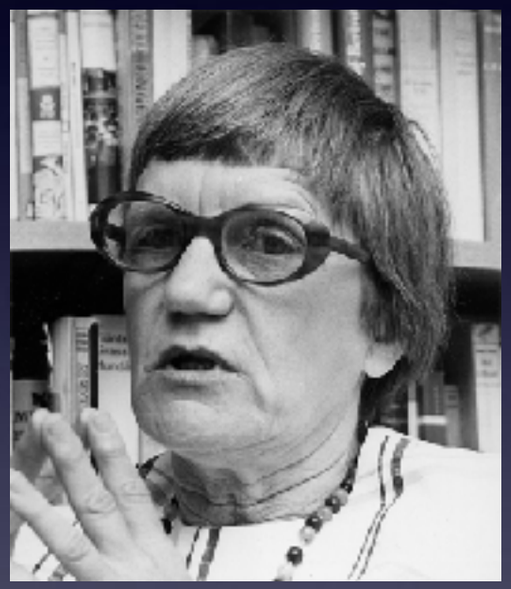
Characteristics of Open Source Projects(2003)
A. Capiluppi, P. Lago & M. Morisio

sourceforge

Population Size and Cultural Complexity

Technological change is itself spurred by
increases in population size



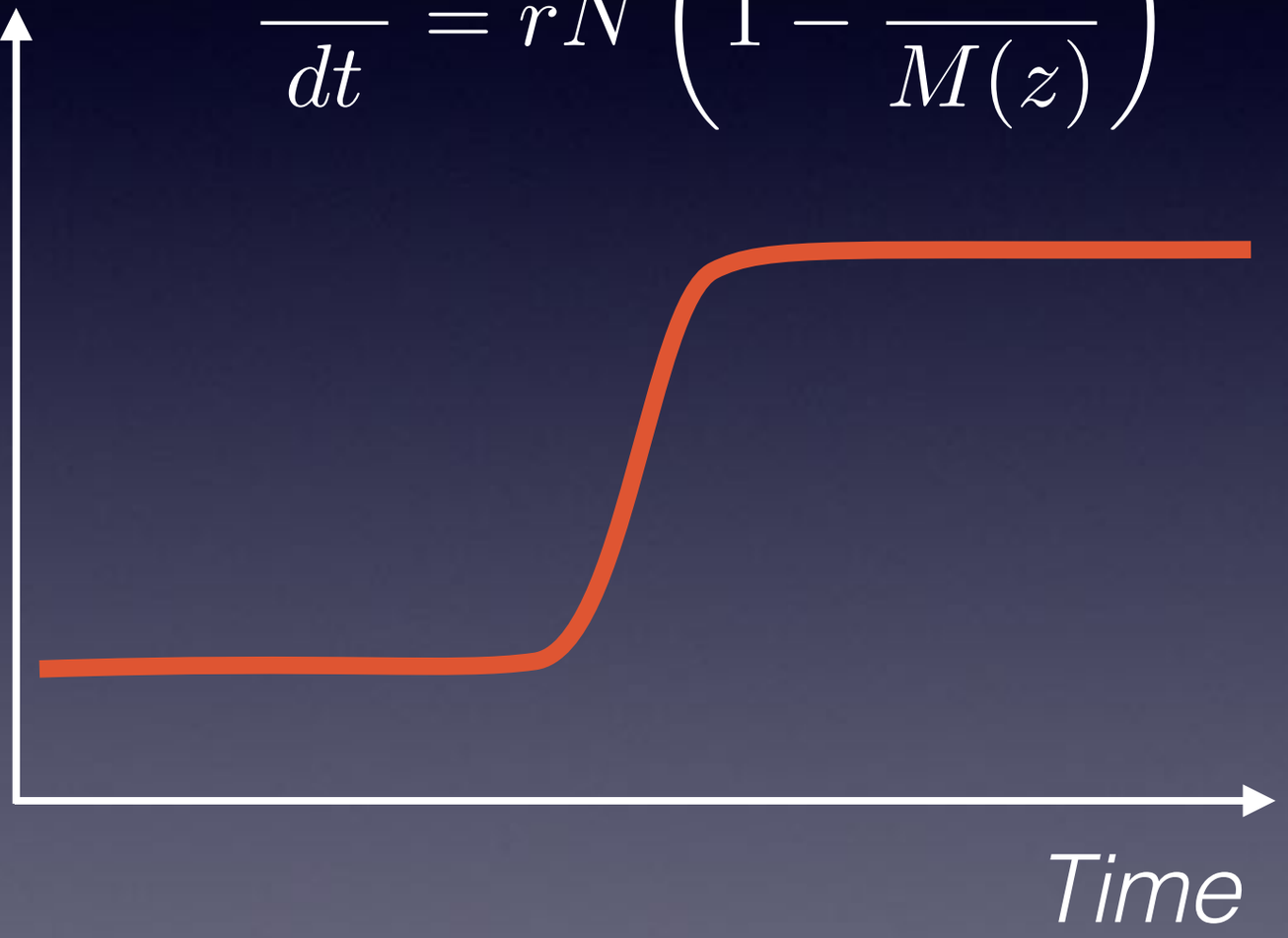


Ester Boserup
(1910-1999)

Population Size and Cultural Complexity

Population size equilibrates with resources at some level mediated by technology

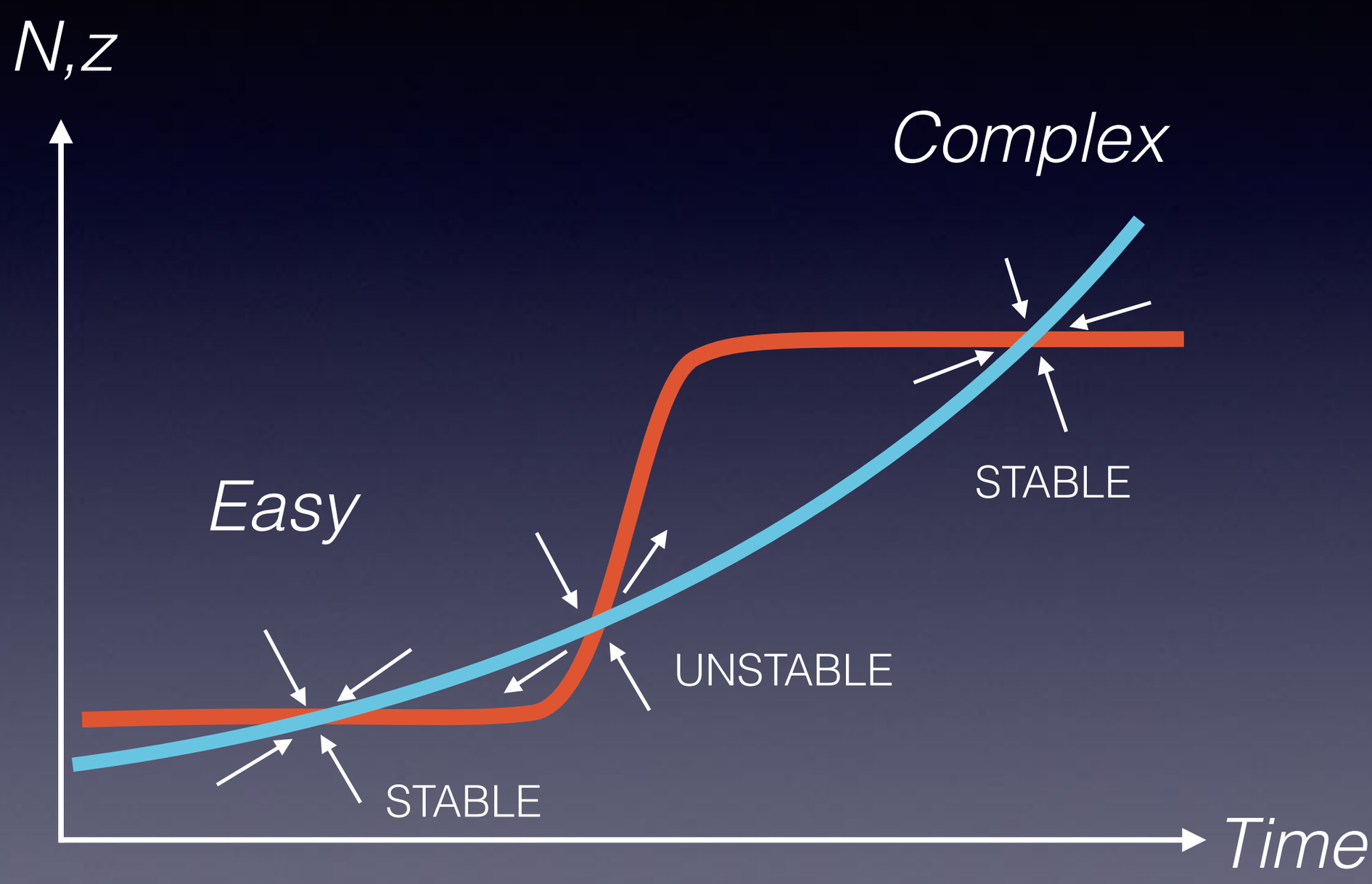
$$\frac{dN}{dt} = rN\left(1 - \frac{N}{M(z)}\right)$$



Thomas R. Malthus
(1766-1834)

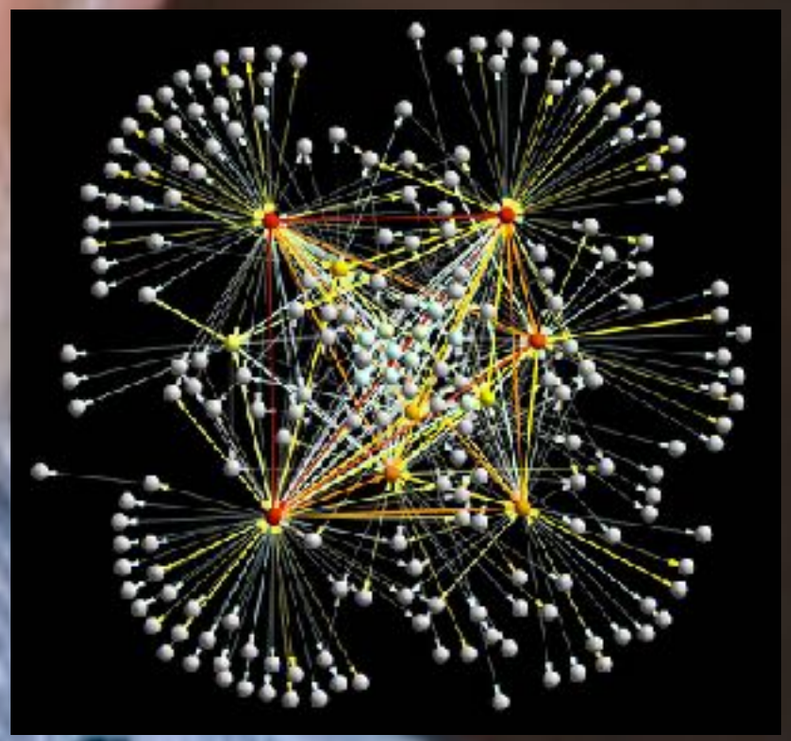
Population Size and Cultural Complexity

Abrupt transitions in technological level are possible



Modeling abrupt cultural regime shifts during the Palaeolithic and Stone Age (2015)

Software Bottleneck



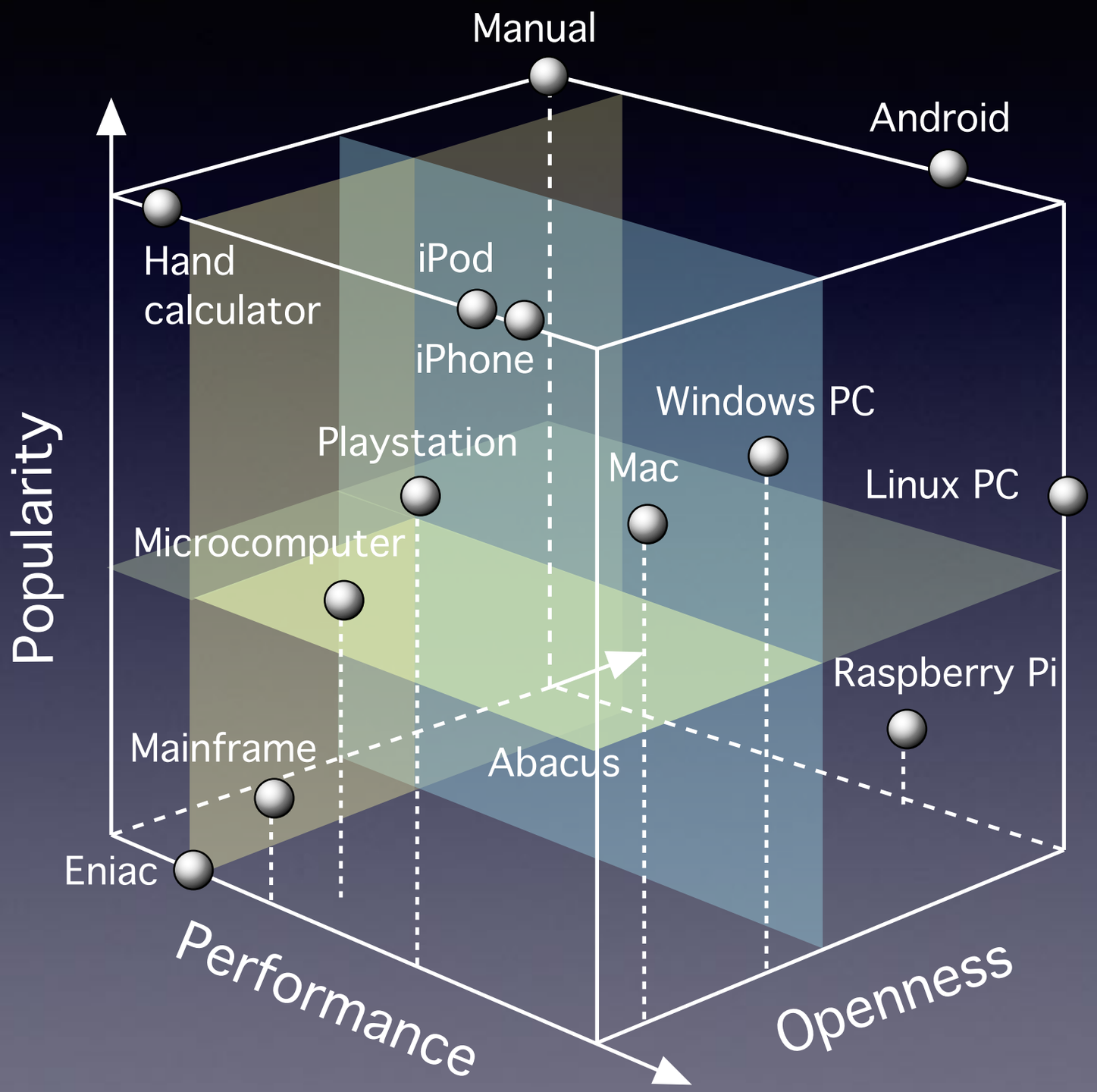
"UNIX is basically a simple operating system,but you have to be a genius to understand the simplicity."

Dennis Ritchie (1941-2011)

Self-Organisation vs Hierarchy in Open-Source Social Networks (2007)

S. Valverde & Ricard V. Solé

Information Technology Evolution Space



Thank You!

Ricard Solé

Jose Lobo

Deborah Strumsky

Marti Rosas-Casals

Bernat Corominas-Murtra

Daniel Rodriguez-Amor

http://complex.upf.es/~sergi