ACROSS
Autonomous Control for a Reliable Internet of Services

ICT COST Action IC1304

Autonomous Control for a Reliable Internet of Services (ACROSS)

Modeling fault behavior in evolving complex open software environments

ACROSS MC meeting, October 13, 2016, Bilbao, Spain

Tihana Galinac Grbac

UIP-2014-09-7945
EVOSOFT

hrzz
Hrvatska zaklada za znanost

cost
EUROPEAN COOPERATION IN SCIENCE AND TECHNOLOGY

Modern networks and software

- Software become central part of the modern network
- It should run on any hardware, serve to many users, satisfy their complex communication needs and deliver proper ICT service, effectively and efficiently
- Modern software has to be flexible on network context, information context, communication context, ...
- Modern network should provide **reliable and robust** ICT services (resistant against system failures, cyber-attacks, high-load and overload situations, flash crowds, etc.)

Key problems with software evolution

- More and more software systems tend to evolve towards complex software systems (e.g. IoS)
- Interconnection of peripheral systems over distributed network into system of systems (IoT)
- Key problems become:
  - Can we develop foundations on software behavior?
  - How can we measure software behaviour in network?
  - Can we predict and simulate software behaviour in network?
  - How to manage complex software system?
  - Are we able just by observing properties of system parts to predict and model its overall behaviour?

EVOSOFT: UIP-2014-09-7945
hrzz
Hrvatska zaklada za znanost

Relation to ACROSS

- Reliability and availability service chains will very much depend on their structure
- knowing the appropriate statistical fault distribution would enable more systematic approach for automated guidance for creation of reliable software chains
- Interesting is to model the underlying processes that generate distributions and how they influence the statistical fault distributions
- Context awarness based on system structure and measurements on software abstract levels

EVOSOFT: UIP-2014-09-7945
hrzz
Hrvatska zaklada za znanost

Previous studies on fault distributions

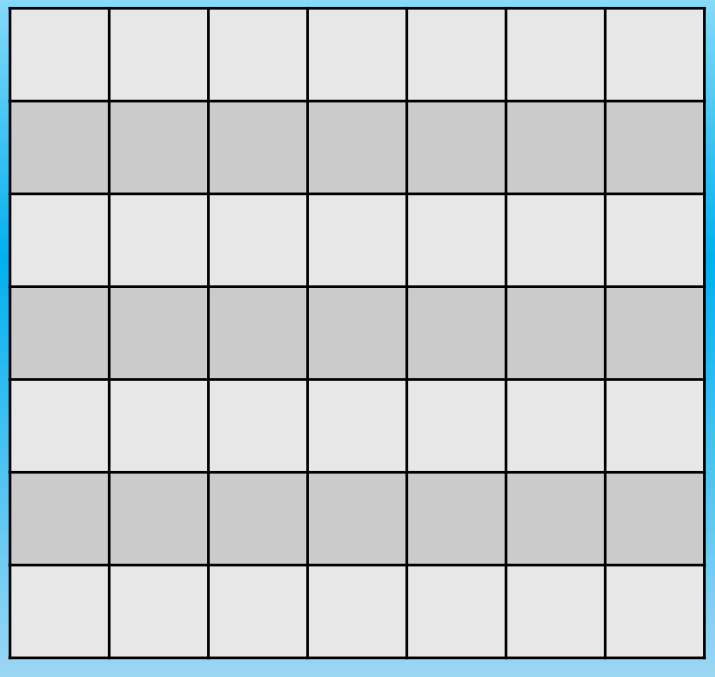
- Empirical studies on fault distributions
- Analytical studies on fault distributions
- Industrial versus open source

EVOSOFT: UIP-2014-09-7945
hrzz
Hrvatska zaklada za znanost

System verification and reliability

- Number of levels of abstraction
- Global properties of system and local properties describing component behaviour
- Imposible to derive simple rules from local properties towards global properties*

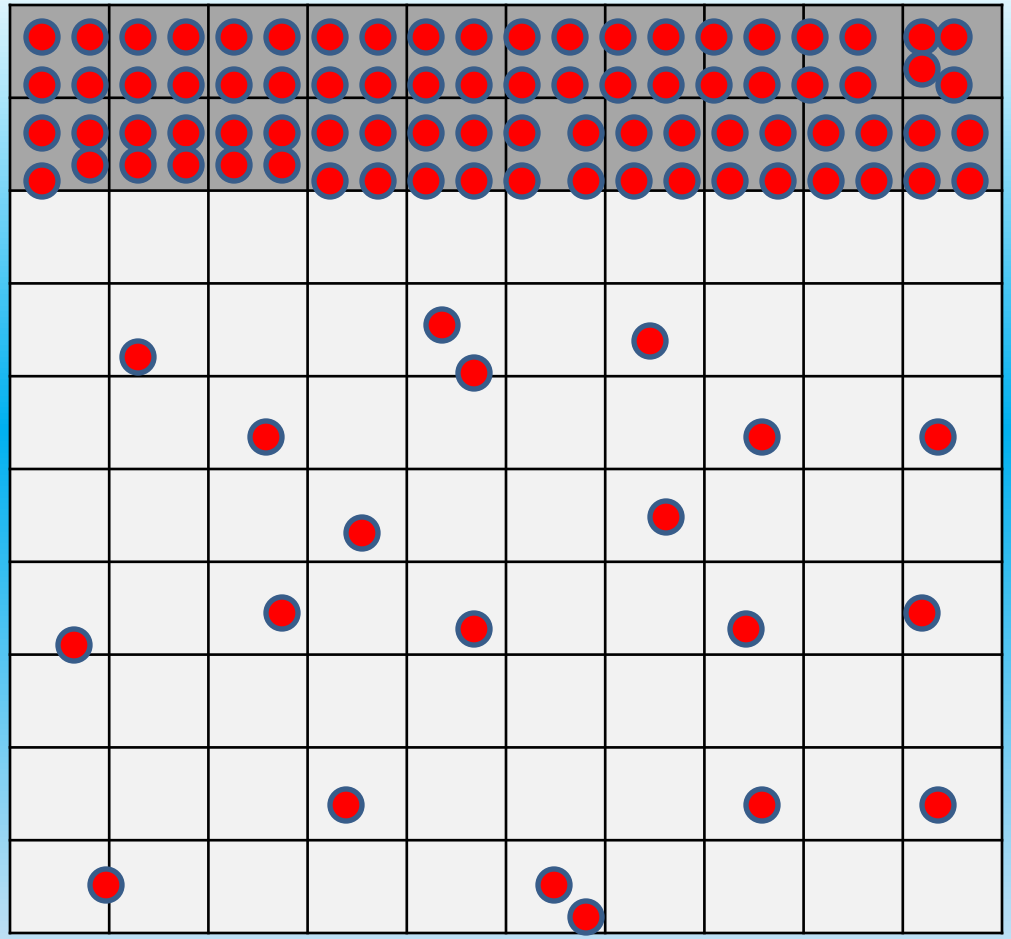
System and system components



Source: Complex software systems : Formalization and Applications -
Work done in EU project GENNETTEC: GENetic NeTworks: Emergence and Complexity

A small number of modules contain most of the faults

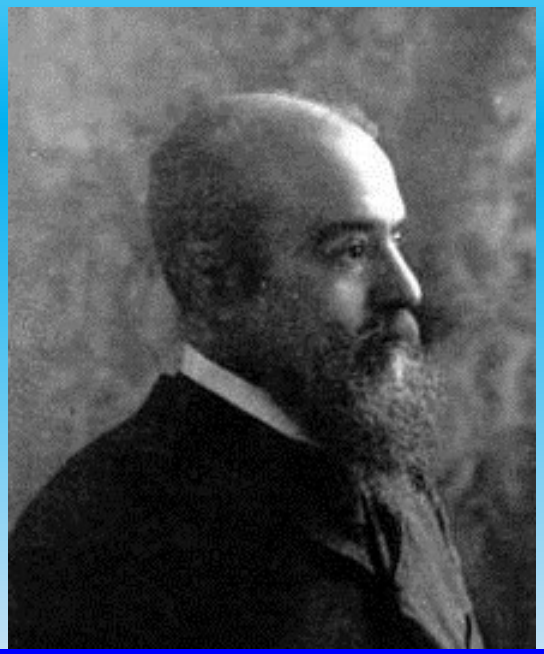
System and system components



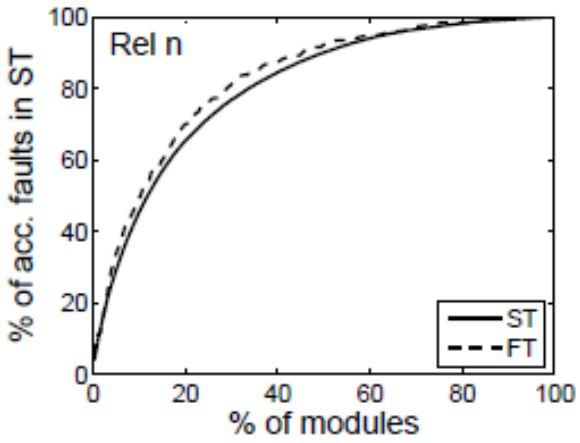
Pareto principle: 80 – 20 rule

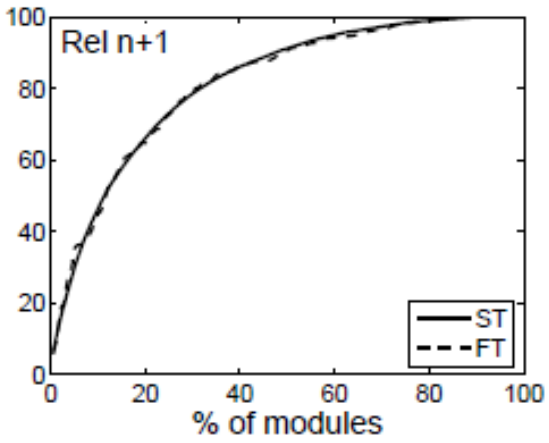
Vilfredo Federico Damaso Pareto

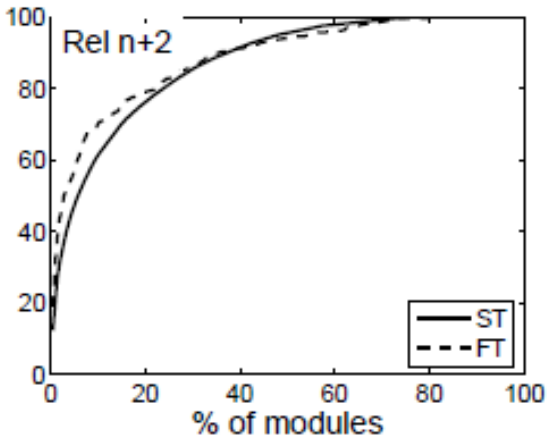
- *1906: 80% of the land in Italy was owned by 20% of the population*
- *Income and wealth among the population follows a Pareto distribution, a* ***power law probability distribution***
- Small occurancies are extremely comon and large occurancies are extremely rare

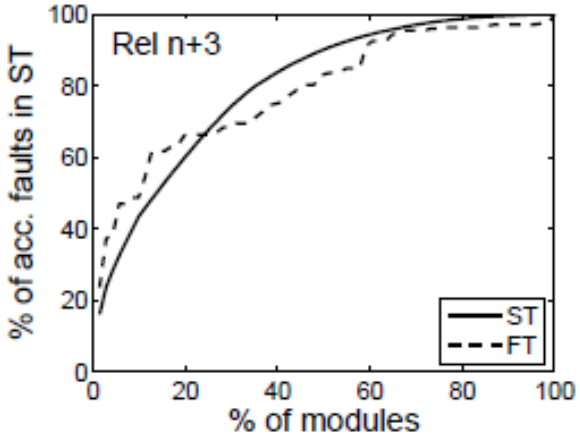


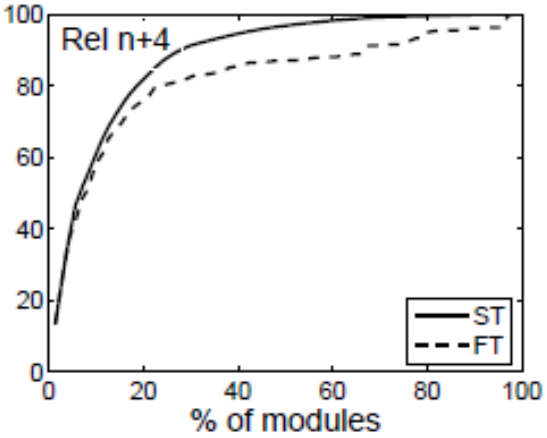
Fault distributions











Empirical studies on fault distributions

- *N. E. Fenton and N. Ohlsson, "Quantitative Analysis of Faults and Failures in a Complex Software System," IEEE Trans. Softw. Eng., vol. 26, no. 8 , pp. 797-814, Aug. 2000.*
- *C. Andersson and P. Runeson, "A Replicated Quantitative Analysis of Fault Distributions in Complex Software Systems," IEEE Trans. Softw. Eng., vol. 33, no. 5, pp. 273-286, May 2007.*
- *T. Galinac Grbac, P. Runeson and D. Huljenic, "A Second Replicated Quantitative Analysis of Fault Distributions in Complex Software Systems," IEEE Trans. Softw. Eng., vol. 39, no. 4, pp. 462-476, Apr. 2013.*

Analytical fault distributions

- All previous principles ultimately depend on the underlying probability distribution
- the fulfillment of a certain empirical principle does not determine the probability distribution uniquely
- The distibutions like double Pareto, Weibull, lognormal, Pareto, and Yule-Simon with power-law in the tail are confirmed

1. *Les Hatton. Power-Law Distributions of Component Size in General Software Systems. IEEE Trans. Software Eng. 35(4): 566-572*
2. *Tihana Galinac Grbac, Darko Huljenić. On the Probability Distribution of Faults in Complex Software Systems, Information and Software Technology, published online first.*

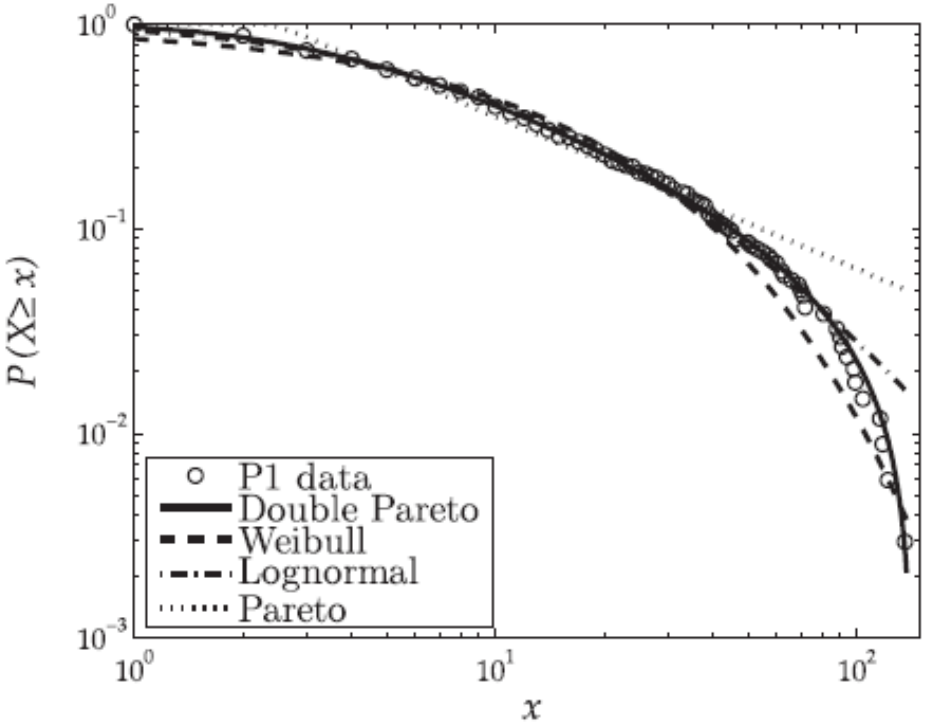
EVOSOFT: UIP-2014-09-7945
hrzz
Hrvatska zaklada za znanost

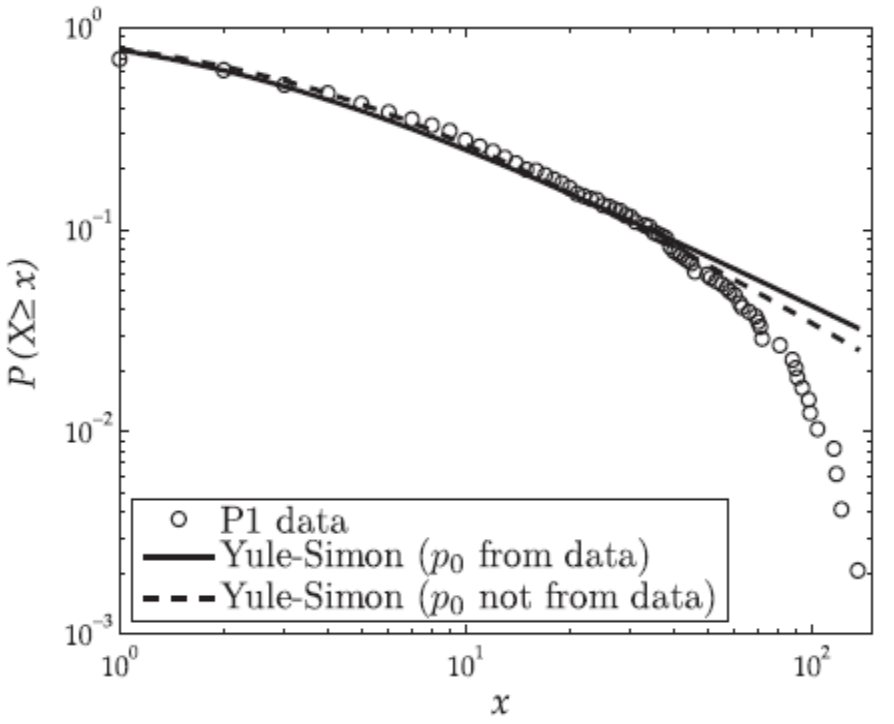
Research questions

- RQ1: How faults are distributed across the software units
- RQ2: Does fault distribution depends on development environment
- RQ3: Is the fault distribution persitant over the system evloution

EVOSOFT: UIP-2014-09-7945
hrzz
Hrvatska zaklada za znanost

Results of analitical distributions fit





Nonlinear regression fit for Pareto, double Pareto, Weibull and Lognormal distribution

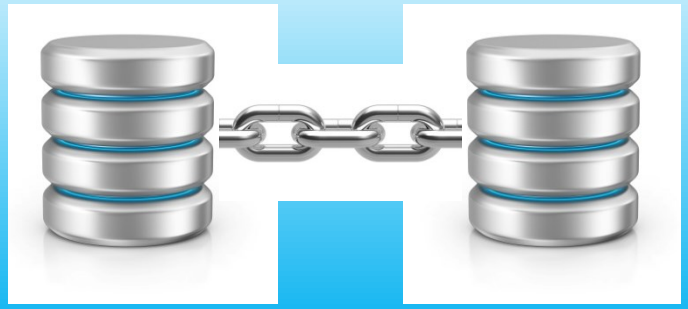
Data selection

- complex enough for network analysis
- evolve over a number of system releases
- Industrial and open access software
- Acess to source code and defect repository
- we selected one industrial telecom core network product from 4G network and two Eclipse plugins: PDE, JDT

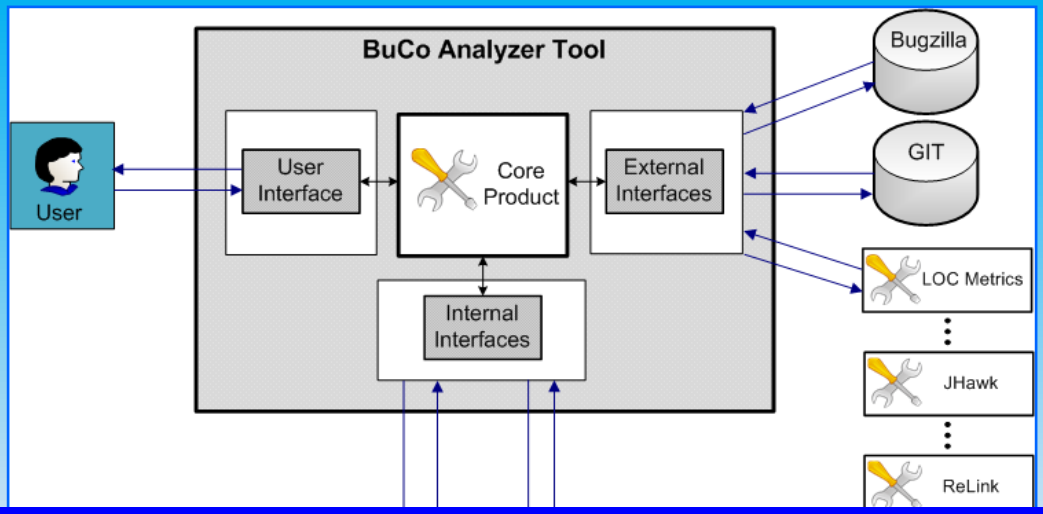
EVOSOFT: UIP-2014-09-7945
hrzz
Hrvatska zaklada za znanost

Data collection

Source code repository
- collection of modules,
Classes, software units



Failure report repository
Repository
- collection of modules,
Classes, software units



EVOSOFT: UIP-2014-09-7945
hrzz
Hrvatska zaklada za znanost

Distribution fit across the studies

Rank	This study	Galinac Grbac et al. [2]	Concas et al. [1]	Zhang [3]
1	Yule–Simon	Double Pareto	Yule–Simon	Weibull
2	Pareto	Lognormal	Double Pareto	Pareto
3	Double Pareto	Yule–Simon	Lognormal	—
4	Lognormal	Weibull	Weibull	—
5	Weibull	Pareto	—	—

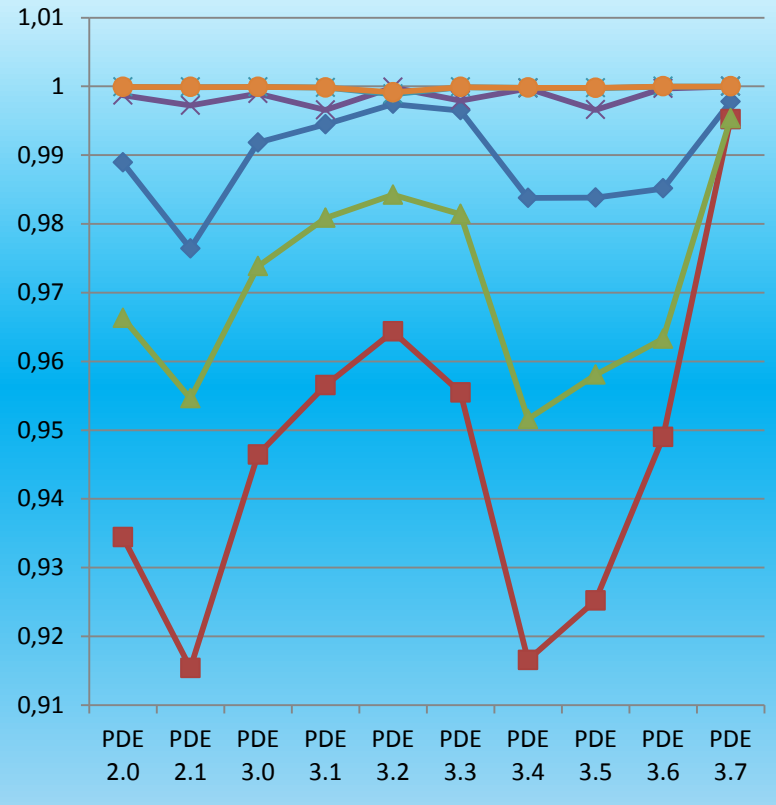
[1] G. Concas, M. Marchesi, A. Murgia, R. Tonelli, I. Turnu, On the distribution of bugs in the Eclipse system, IEEE Trans. Softw. Eng. 37 (2011) no. 6, 872--877.

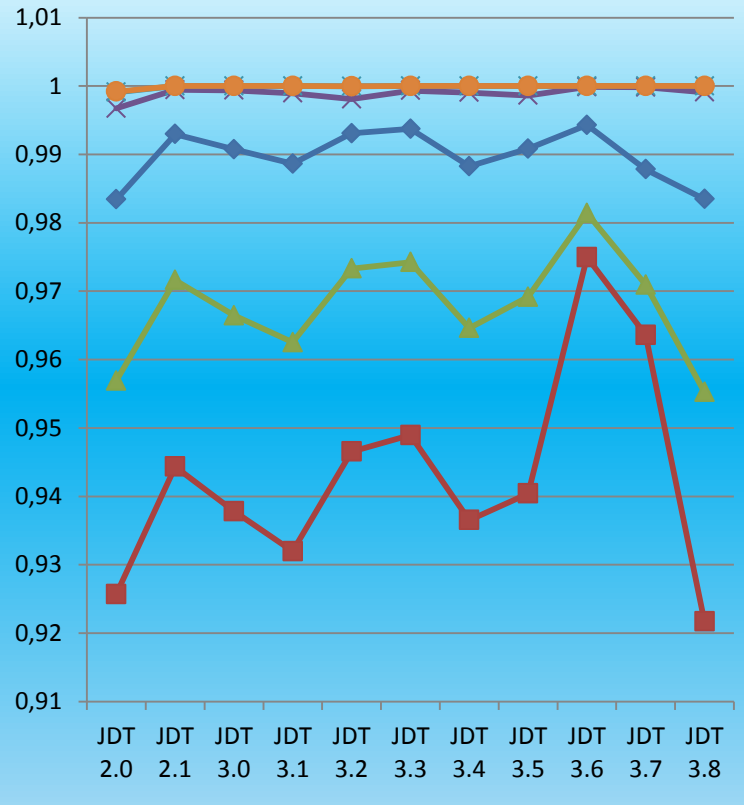
[2] T. Galinac Grbac, D. Huljenić, On the Probability Distribution of Faults in Complex Software Systems, Information and Software Technology 58 (2015), 250-258.

[3] H. Zhang, On the distribution of software faults, IEEE Trans. Softw. Eng. 34 (2008) no. 2, 301-302.

EVOSOFT: UIP-2014-09-7945
hrzz
Hrvatska zaklada za znanost

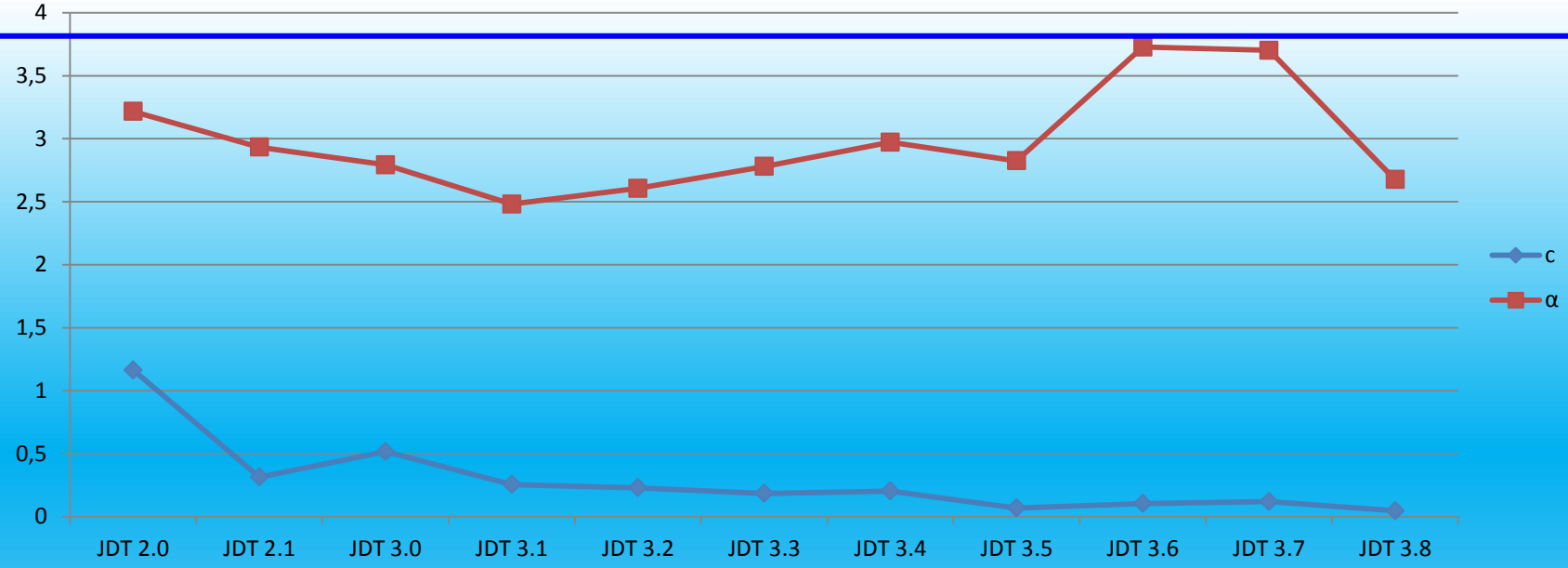
Distributions fit – R^2

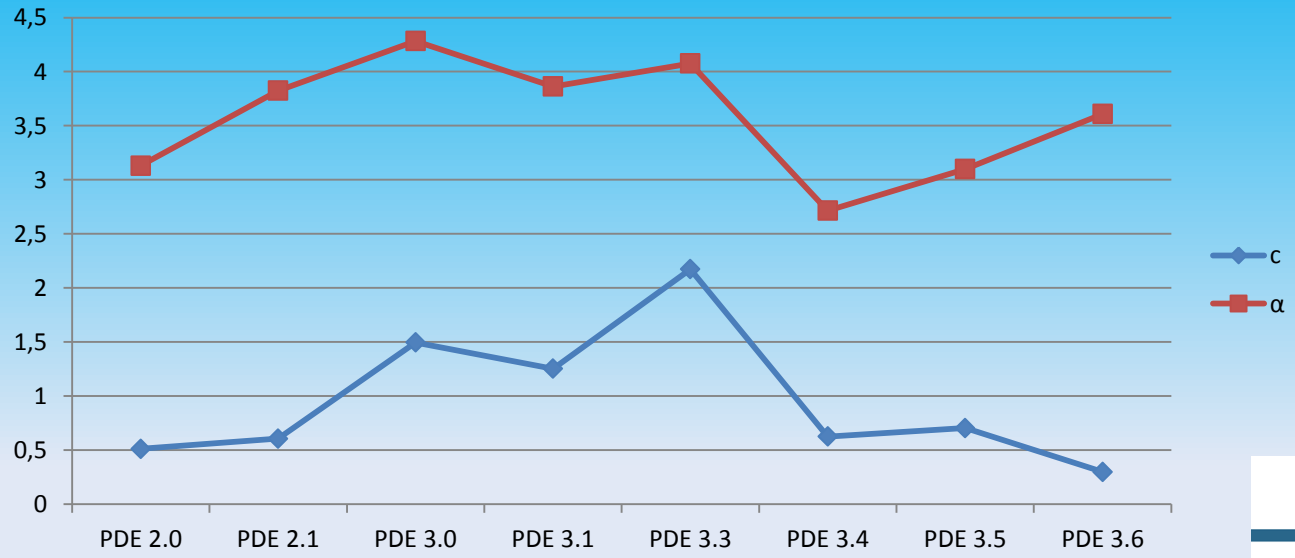




EVOSOFT: UIP-2014-09-7945
hrzz
Hrvatska zaklada za znanost

Evolution of Yule-Simon distribution





EVOSOFT: UIP-2014-09-7945
hrzz
Hrvatska zaklada za znanost

Conclusion

- Yule Simon gives the best fit for all analysed projects
- Yule Simon is similar for projects in system evolution
- But, there are differences betwen parameters in different environments (JDT and PDE projects – Open source Eclipse projects)
- We can reuse Y-S but only between releases in system evolution
- Fitting Y-S with p0 (number of modules with no faults) from the data and p0 not from the data gives almost the same parameters (similarity up to 2 decimal places)
- In environment where there is a lot of software units with no faults Pareto distribution is almost as good as Yule Simon because the tail starts close to ‘0’.
- Simulations aiming to find underlying distributions for generative models and finding simulation model of software fault-behaviour in network over time
- Next step we want to find a model based on Yule process that can explain evolution of faults and other system properties of lage complex systems