Software, Engineering and Experimentation

Guilherme Horta Travassos
Universidade Federal do Rio de Janeiro
COPPE/PESC
CNPq Researcher, **ISERN** Member
ght@cos.ufrj.br
www.cos.ufrj.br/~ght

Agenda

Software Systems Perspective
Science, Engineering and Software
Current Steps
The ESE Group at COPPE
Final Remarks

Software Systems

used largely by people other than system developers

users may be from different background, so a proper user interaction must be designed

Portability and heterogeneity are key

They must be thoroughly verified, validated and tested before their operational use

Software is everywhere…

Software Systems Evolution

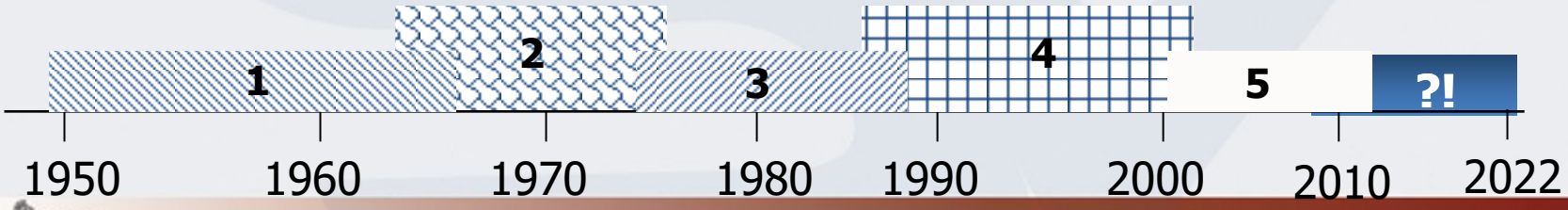
Early years
Custom Software
Standalone
Batch

Second Stage
Multi-user
Real-time
Database
Product Software

Third Stage
Distributed Systems
Embedded "intelligence"
Low cost hardware
Consumer Impact

Fourth Stage
Powerful desk-top systems
Object-oriented technologies
Expert systems
Artificial neural networks
Parallel computing
Network computers

Fifth Stage
Multi-skilled, geographically distributed development
Componentry (reuse and recycling)
Development and evolution models, including biological analogies
Interdependence among design, business, and evaluation
Agile software manufacture
Empowering the domain expert (vs. maintaining integrity)
Non-scripting development languages



Adapted from PRESSMAN, R. S. , 1994, "*Software Engineering: A Practitioner's Approach*", *European Edition, McGraw-Hill.*

Some Characteristics of Software Systems

Software can not be manufactured (in the classical sense)

X

Software costs are concerned with its engineering

Some Characteristics of Software Systems

Software doesn't "wear out", but it deteriorates

Hardware

X

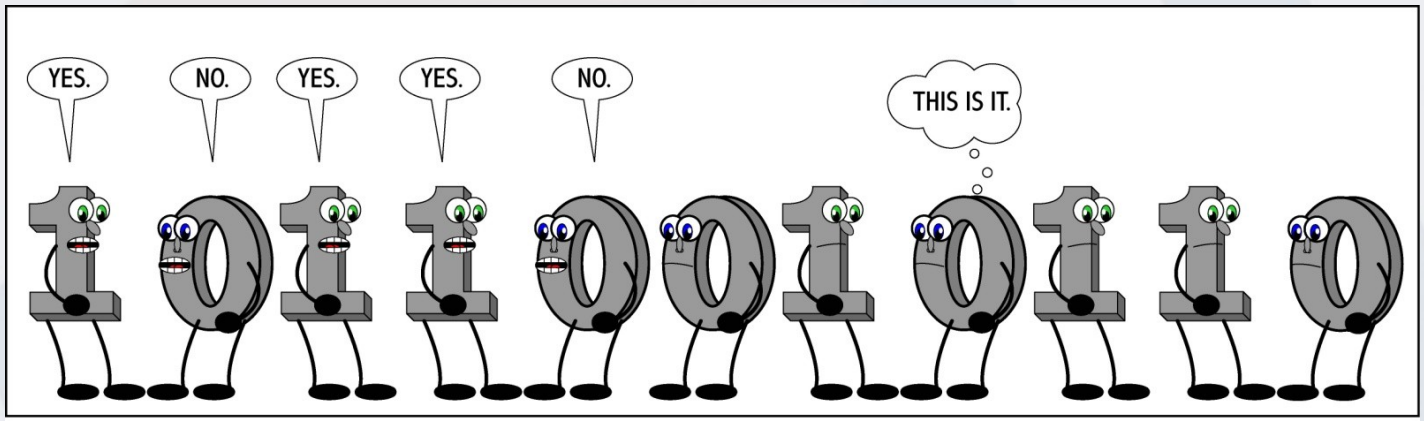
Software

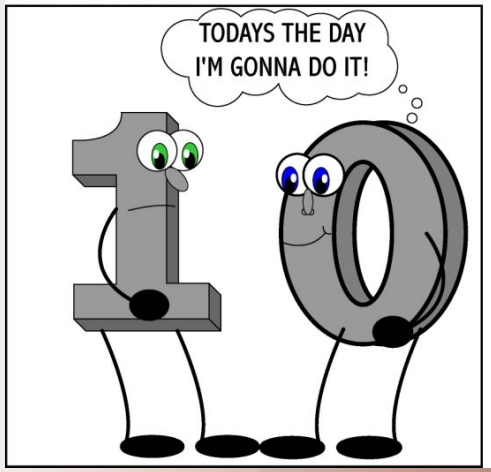
Some Characteristics of Software Systems

Usually custom-built rather than assembled from existing (high quality) components

X

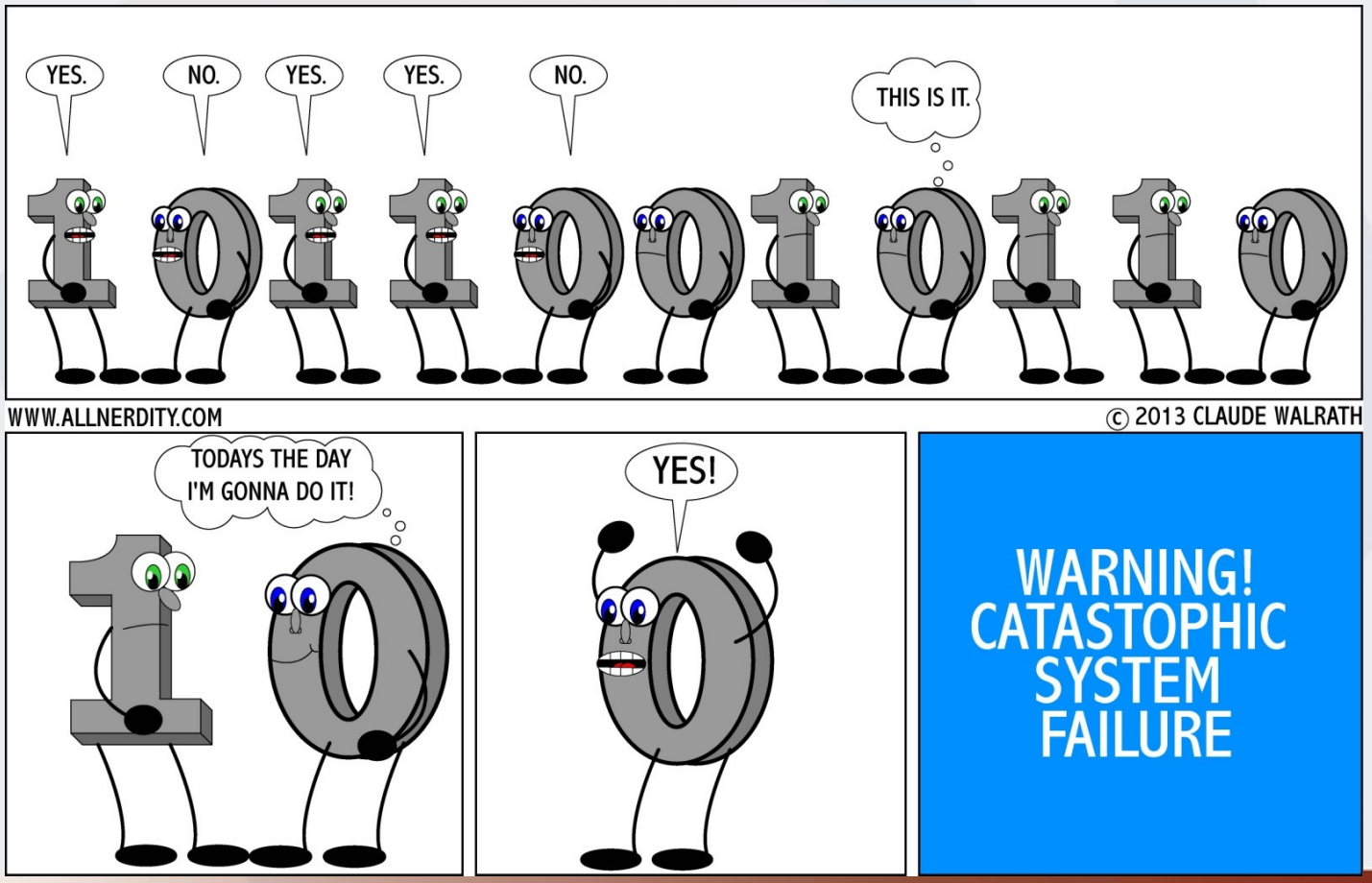
Some Characteristics of Software Systems





Some Characteristics of Software Systems

All software systems can fail…



Some Characteristics of Software Systems

- Top 10 software failures of 2014 (COMPUTERWORLDUK): http://www.computerworlduk.com/galleries/infrastructure/top-10-software-failures-2014-3599618/
  - Amazon 1 penny price glitch
  - UK airspace closed
  - Toyota Prius recalled over software glitch
  - Heartbleed security flaw uncovered
  - US National Grid Gas Company blew $1 billion
  - Emergency numbers go offline for six hours
  - Apple forced to pull iOS8 update (phones unable to make calls)
  - iCloud hacked
  - Air India forced to divert Boeing 787 flight
  - Delivery of F-35B stealth fighters delayed

Why ?

- A full list of evidence at http://catless.ncl.ac.uk/Risks/

...

Software Engineers Reality...

Software systems construction does not follow a smooth pathway…

Software Systems Construction

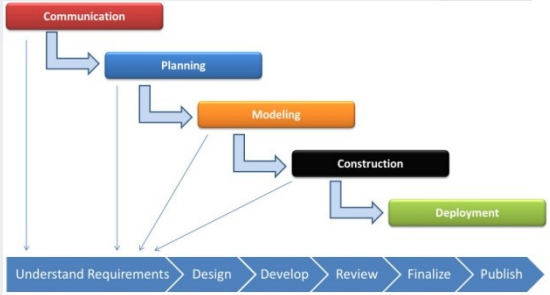
In general, it follows a Software Development Process specifying:

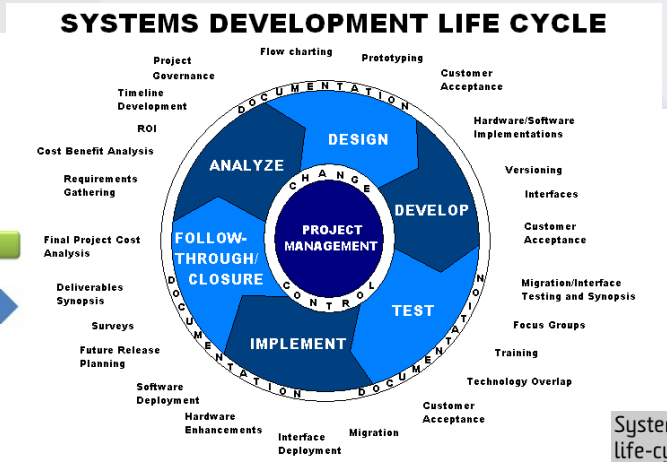
the adopted software life-cycle and paradigm
the software technologies (methods, tools) to be used throughout the development time
who participates (roles) and when
the management, quality and verification, validation and testing plans

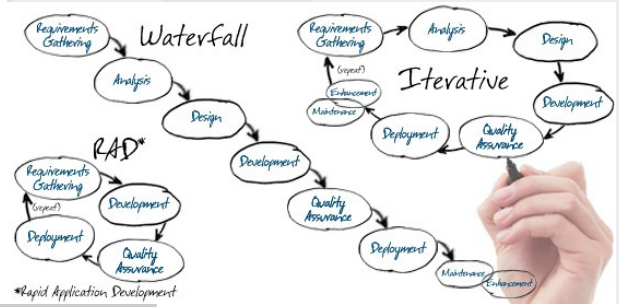
It defines how multiple developers should communicate and cooperate

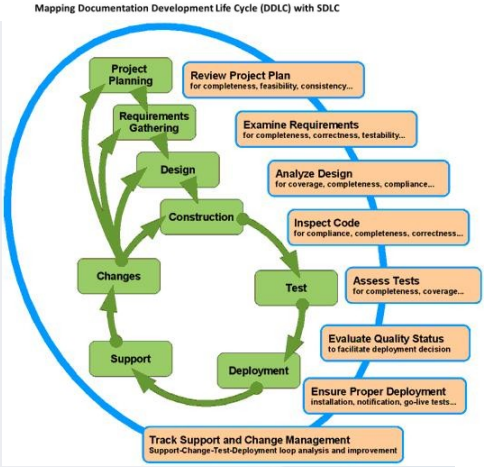
Software Systems Construction

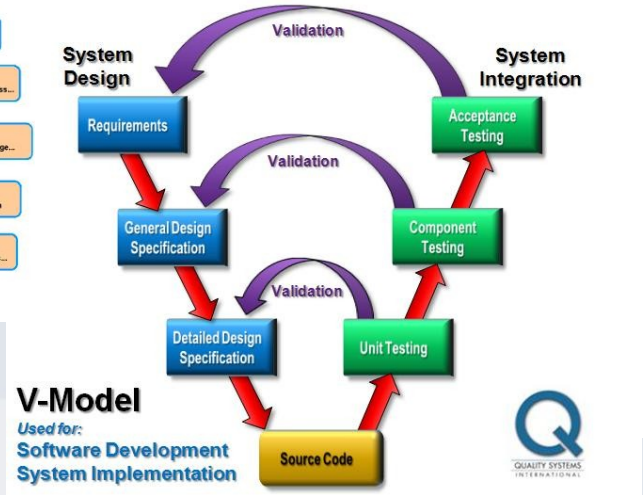
Some life cycle shapes…

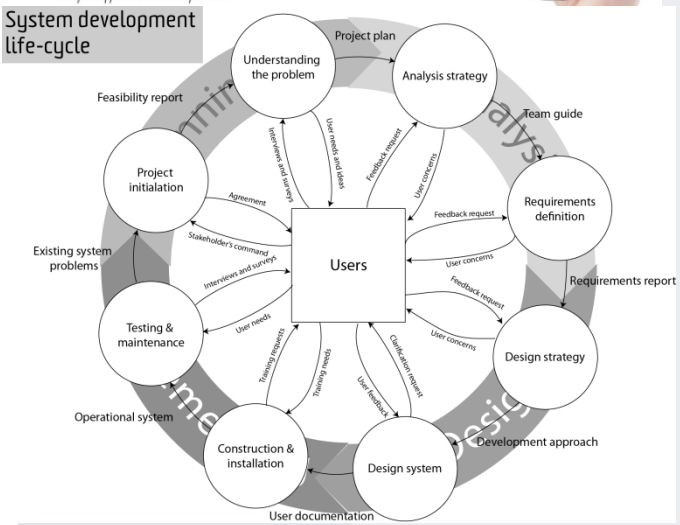












Software Engineers Reality...

Software Development requires communication and collaboration between developers and stakeholders…

not easy to guarantee communication and collaboration...

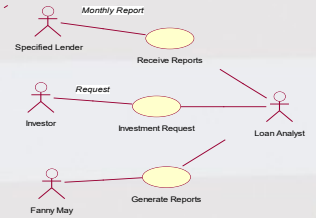
Software Development Communication Perspectives

REQUIREMENTS

Loan-Arranger Requirements Specification – Jan. 8, 1999

Background

Banks generate income in many ways, often by borrowing money from their depositors at a low interest rate, and then lending that same money at a higher interest rate in the form of bank loans. However, property loans, such as mortgages, typically have terms of 15, 25 or even 30 years. For example, suppose that you purchase a $150,000 house with a $50,000 down payment and borrow a $100,000 mortgage from National Bank for thirty years at 5% interest. That means that National Bank gives you $100,000 to pay the balance on your house, and you pay National Bank back at a rate of 5% per year over a period of thirty years. You must pay back both principal and interest. That is, the initial principal, $100,000, is paid back in 360 installments (once a month for 30 years), with interest on the unpaid balance. In this case the monthly payment is $536.82. Although the income from interest on these loans is lucrative, the loans tie up money for a long time, preventing the banks from using their money for other transactions. Consequently, the banks often sell their loans to consolidating organizations such as Fannie Mae and Freddie Mac, taking less long-term profit in exchange for freeing the capital for use in other ways.

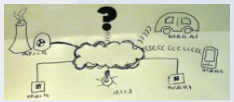


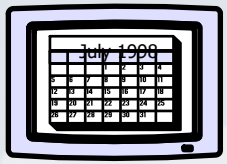
FORMAL

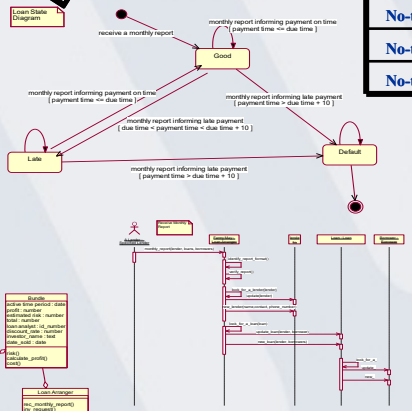
Scalene Triangle:

{<x,y,z>: (x != y) ^ (x != z) ^ (y != z)}

TEST CASES			
CLASS	X	Y	Z
Scalene	3	4	5
Isosceles	5	5	8
Isosceles	3	4	3
Isosceles	4	7	7
Eqüiláteral	2	2	2
No-triangle	1	2	3
No-triangle	5	1	4
No-triangle	3	5	2

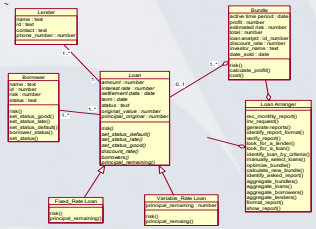






Tacit requirements

AD-HOC



Problem Domain

Solution Domain

Computer Domain

SOURCE CODE

Travassos, G.H. (2014). Software Defects: Stay Away from Them. Do Inspections!. In: 9th International Conference on the Quality of Information and Communications Technology (QUATIC) . Guimarães, Portugal. http://dx.doi.org/10.1109/QUATIC.2014.8

Software Engineers Reality...

Software Development demand software technologies, however...

not enough evidence regarding software technologies...

Some Software Technologies Pitfalls...

As it has been commented by Forrest Shull (Keynote at ICGSE, 2012):

Requirements Elicitation: 30 studies covering 43 different techniques over 20 years of research

Dieste, O., Juristo, N., and Shull, F. "Understanding the Customer: What Do We Know about Requirements Elicitation?" IEEE Software, vol. 25, no. 2, pp. 11-13, March/April 2008.

SW Process Capability/Maturity Models: 61 studies; 52 process models.

von Wangenheim, C., Hauck, J., Zoucas, A., Salviano, C., McCaffery, F., and Shull, F. "Creating Software Process Capability / Maturity Models," IEEE Software, vol. 27, no. 4, pp. 92-94, July / August 2010.

Distributed SW Development: "Few of the models from our review were evaluated…"

Prikladnicki, R., Audy, J. L. N., and Shull, F. "Patterns in Effective Distributed Software Development," IEEE Software, vol. 27, no. 2, pp. 12-15, March / April 2010.

SPL Testing Techniques: 60% of papers describe "solutions or conceptual proposals," while "just a few" report experiences from real development environments.

da Mota Silveira Neto, P.A.; Runeson, P.; do Carmo Machado, I.; de Almeida, E.S.; de Lemos Meira, S.R.; Engstrom, E.; , "Testing Software Product Lines," Software, IEEE , vol.28, no.5, pp.16-20, Sept.-Oct. 2011.

Some Software Technologies Pitfalls...

And also observed in some of our investigations:

Cost Estimation Models: 11 studies (including 2 replications) using different data sets. No evidence about feasibility of models nor possibility of aggregation

Kitchenham, B. ; Mendes, E. ; Travassos, G. H. (2007). Cross versus within-company cost estimation studies: A systematic review. IEEE Transactions on Software Engineering, v. 33, p. 316-329, 2007. http://dx.doi.org/10.1109/TSE.2007.1001

Model based Testing: from 85 selected papers (representing 71 approaches), 27% are speculative, 45% just present simple using examples, 15% show proof of concepts, 5% report some experience and 8% have been experimented.

Dias Neto, A. C. ; Subramanyan, R. ; Vieira, M. E. R. ; Travassos, G. H. ; Shull, F. .(2008) Improving evidence about software technologies: A look at model-based testing. IEEE Software, v. 25, p. 10-13, 2008. http://dx.doi.org/10.1109/MS.2008.64

Testing Stop Criteria: 74 criteria (3 repeated) resulting in 108 variations. Most of them regard software reliability. Others are specific. Just 27% have been evaluated, without evidence about their feasibility (no context indication)

Vidigal, V., Travassos, G. H. (2013). A quasi -systematic review on Testing Stop Criteria. WAMPS 2013.

Some Software Technologies Pitfalls...

And also observed in some of our investigations:

Agility Characteristics and Agile Practices: More relevant characteristics to introduce agility in software processes are concerned with communication, understandability and adaptation (not with agile methods). The agile practices Presence of Client and Planning Poker are not relevant. However, Continuous Integration and Backlog are highly relevant.

De Mello, R.M.; Silva, P.C.; Travassos, G.H. (2014).
Agility in Software Processes: Evidence on Agility Characteristics and Agile Practices. SBQS 2014.

Estimation of Software Testing Effort: There is no consensus about software testing and what can be considered effort regarding it. Therefore, current models and factors are not generically adequate and to use one or another model is risky.

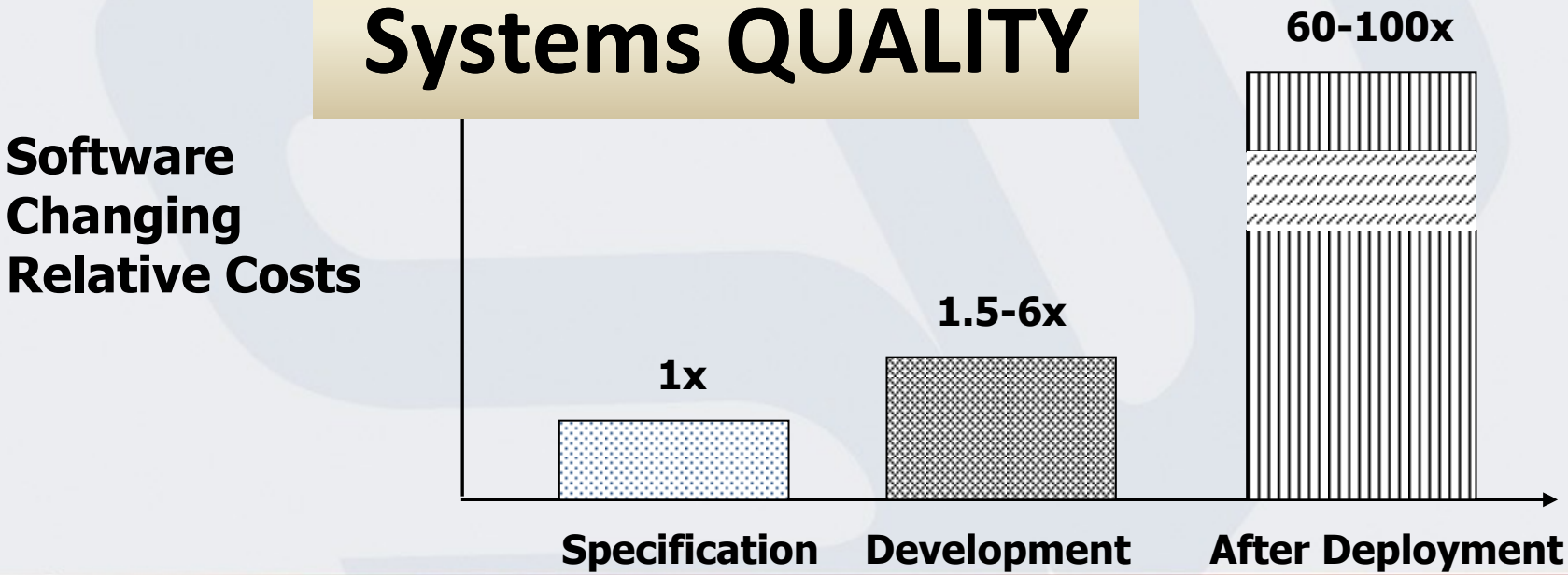
Souza, T.S.; Ribeiro, V. V.; Travassos, G.H. (2014).
Software Testing Estimation Effort: Models, Factors and Uncertainties. CACIC 2014 (in press)

Software Systems: related persistent problems

We struggle to build high reliability and quality software

However, our ability to support and enhance existing software is still threatened by poor design and insufficient resources

LACK OF Software Systems QUALITY



Software Engineers Reality...

Lack of Software Systems Quality, due...

SOFTWARE SYSTEMS DEFECTS

Society's Dependency on Software Systems

Computers everywhere demand software that have made society highly dependent on software services and facilities

Enormous economic damage and potential human suffering can occur when software systems fail

We need to improve our capacity of engineering software products!

Hardware advancements continue to outpace our ability to build software to tap hardware's potential

Brief History of Engineering

- **Pre-scientific revolution**
  - Procedures usually accomplished by trial and error. Imagination and some reasoning have produced interesting apparels, such as monuments. In the Renaissance first engineers started to systematically ask what works and why. Leonardo da Vinci marked this period.

- **Industrial revolution**
  - Galileo´s publication (Two New Sciences) started the structural analysis by adopting a scientific approach. Steam engines became reality. Rationalists (French) and empirically oriented (British) pushed new engineer schools. Practical thinking became scientific besides intuition, engineers developed mathematical analysis and controlled experiments. University education started. *Professional societies promoted the flowing of information through meetings and journal publications*.

    - http://www.creatingtechnology.org/history.htm

Brief History of Engineering

- **Second Industrial revolution**
  - Electricity and mass production landmarked this period. Chemical, electrical and telecommunications industries flourished, together with Marine, Aeronautic, Control, Industrial engineering. Engineering curricula and graduate schools shown up. Workshops, industrial research and systematic innovations took place.

- **Information Revolution**
  - Research and development in all fields of science after World War II. Engineering research exploded! It has been also *simulated* by Aerospace, microelectronics, computers. Intensive, large scale research has produced bodies of powerful systematic knowledge. Different scientific fields being combined. Evidence based approach is key.

    - http://www.creatingtechnology.org/history.htm

Science, Engineering and Software

- **Science**
  - (knowledge from) the systematic study of the structure and behavior of the physical world, especially by watching, measuring and doing experiments, and the development of theories to describe the results of these activities
  - **a particular subject that is studied using scientific methods**

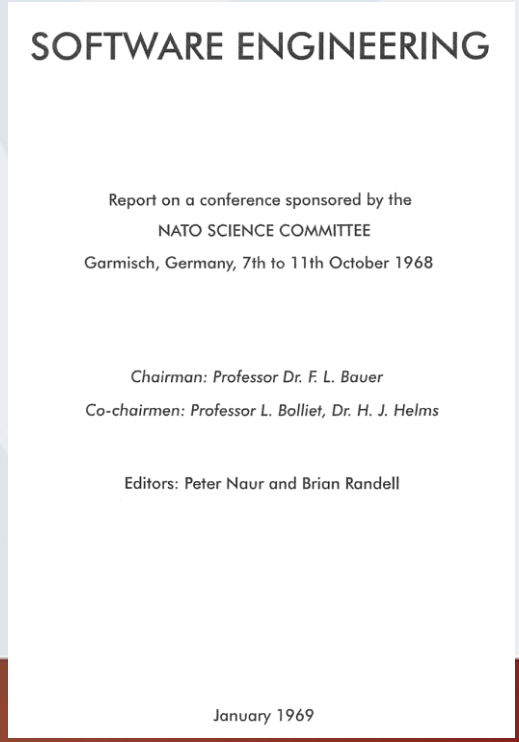
- **Scientist**
  - an expert who studies or works in one of the sciences

- **Engineer**
  - a person whose job is to design or build machines, engines or electrical equipment, or things such as roads, railways or bridges, using **scientific principles**
    - a **civil** engineer
    - a **mechanical/structural** engineer

    a **software** engineer

- **Engineering**
  - the work of an engineer, or the study of this work

http://dictionary.cambridge.org/dictionary/british


SOFTWARE ENGINEERING

Report on a conference sponsored by the
NATO SCIENCE COMMITTEE
Garmisch, Germany, 7th to 11th October 1968

Chairman: Professor Dr. F. L. Bauer
Co-chairmen: Professor L. Bolliet, Dr. H. J. Helms

Editors: Peter Naur and Brian Randell

January 1969

Software Engineering

- It is “a discipline whose aim is the production of fault-free software delivered on time and within budget, that satisfies the client's needs. Furthermore, the software must be easy to modify when the user's needs change.” (Stephen R. Schach, 2008)

- It “is the application of a systematic, disciplined, quantifiable approach to the design, development, operation, and maintenance of software, and the study of these approaches; that is, the application of engineering to software” (SWEBOK, 2004)

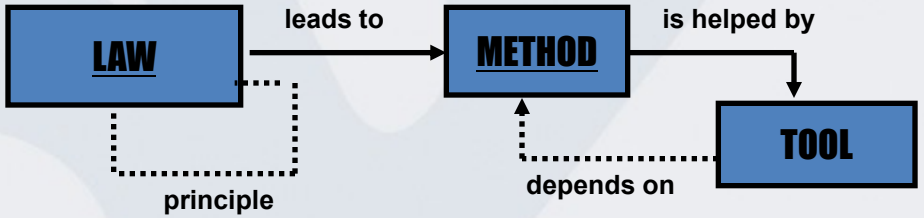
Software Engineering ≠ Software Development

Science, Engineering and Software

- **Engineer**
  - a person whose job is to design or build machines, engines or electrical equipment, or things such as roads, railways or bridges, using **scientific principles**
    - a **civil** engineer
    - a **mechanical/structural** engineer
    - a **software** engineer

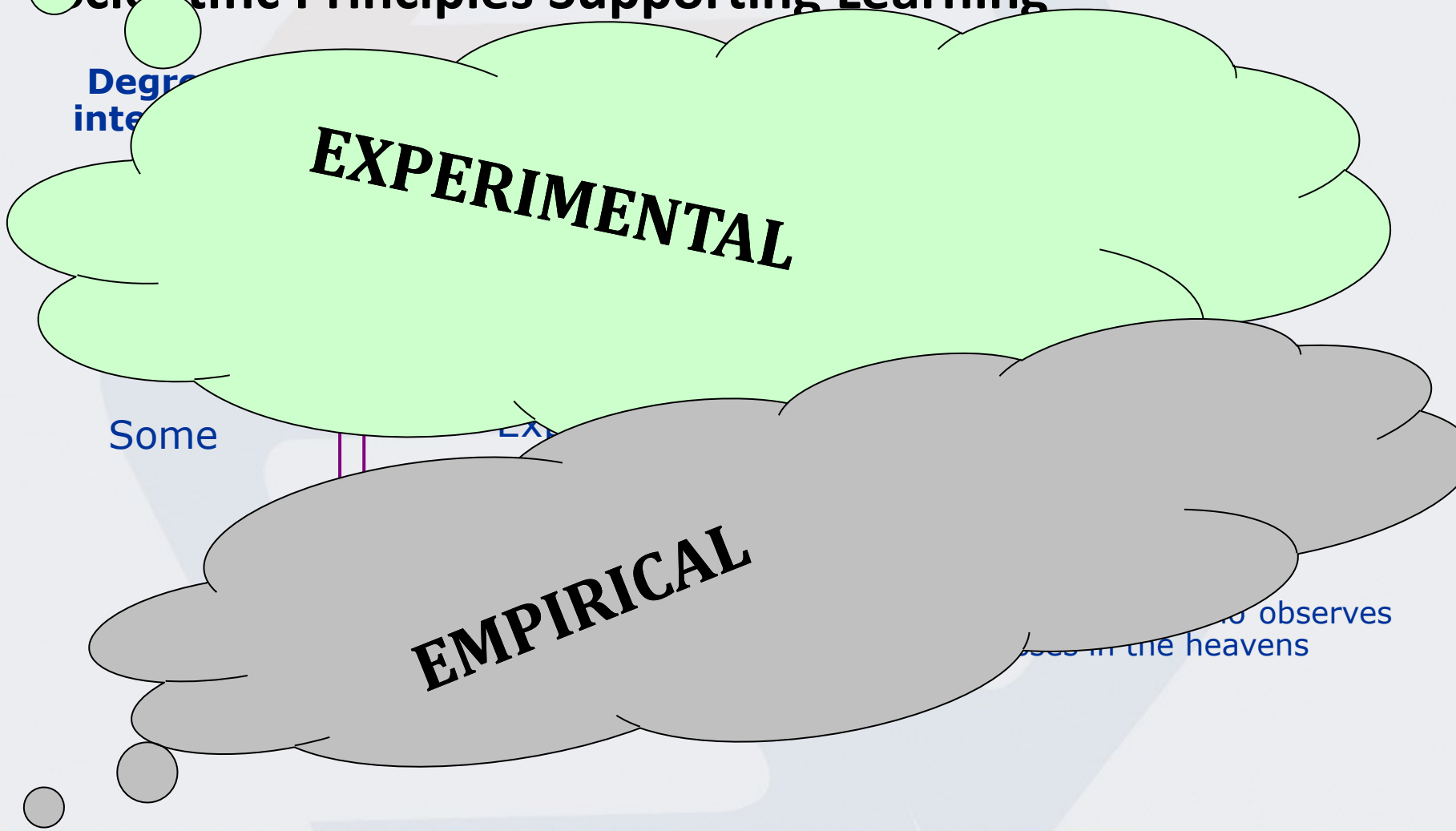
- **Engineering**
  - the work of an engineer, or the study of this work

http://dictionary.cambridge.org/dictionary/british



Science, Engineering and Software

- Scientific Principles Supporting Learning



Science, Engineering and Software

empirical

1: originating in or based on observation or experience <*empirical* data>
2: relying on experience or observation alone often without due regard for system and theory <an *empirical* basis for the theory>
3: capable of being verified or disproved by observation or experiment <*empirical* laws>
4: of or relating to empiricism

experimental

1: of, relating to, or based on experience or experiment
2*a* **:** serving the ends of or used as a means of experimentation <an *experimental* school> *b* **:** relating to or having the characteristics of experiment **:** tentative <still in the *experimental* stage>

http://www.merriam-webster.com/dictionary

Science, Engineering and Software

empirical

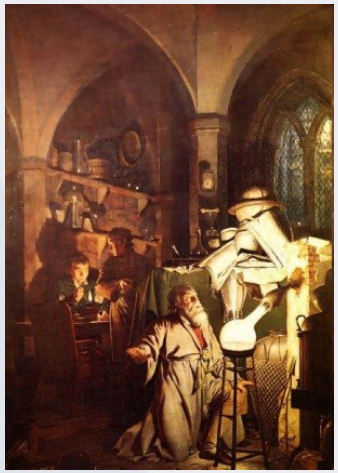
1: originating in or based on observation or experience <*empirical* data>
2: relying on experience or observation alone often without due regard for system and theory <an *empirical* basis for the theory>
3: capable of being verified or disproved by observation or experiment <*empirical* laws>
4: of or relating to empiricism

ntal

em·pir·i·cism
noun \im-ˈpir-ə-ˌsi-zəm, em-
1
a : a former school of medical practice founded on experience without the aid of science or theory
*b : **quackery**, **charlatanry***

http://www.merriam-webster.com/dictionary/empiricism?show=0&t=1322528059

Science, Engineering and Software
The “Quack” World



The **philosopher's stone** (Latin: *lapis philosophorum*) is a legendary alchemical substance said to be capable of turning base metals (lead, for example) into gold (*chrysopoeia*) or silver. It was also sometimes believed to be an elixir of life, useful for rejuvenation and possibly for achieving immortality. For many centuries, it was the most sought-after goal in Western alchemy. The philosopher's stone was the central symbol of the mystical terminology of **alchemy**, symbolizing perfection at its finest, enlightenment, and heavenly bliss. Efforts to discover the philosopher's stone were known as the Magnum Opus.

From http://en.wikipedia.org/wiki/Philosopher%27s_stone

In folklore, the **silver bullet** is supposed to be the only kind of bullet for firearms that is effective against a werewolf, witch, or other monsters. Sometimes (not always) the silver bullet is also inscribed with Christian religious symbolism, such as a cross or the initials "J.M.J" (Jesus, Mary & Joseph).

The term has been adopted into a general metaphor, where "silver bullet" refers to any straightforward solution perceived to have extreme effectiveness. The phrase typically appears with an expectation that some new technological development or practice will easily cure a major prevailing problem.

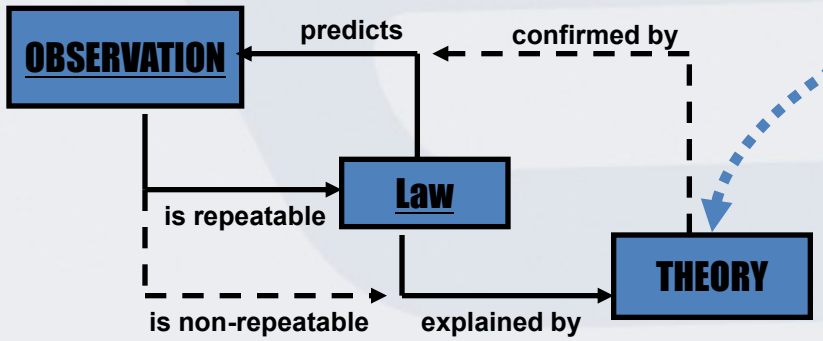
From http://en.wikipedia.org/wiki/Silver_bullet

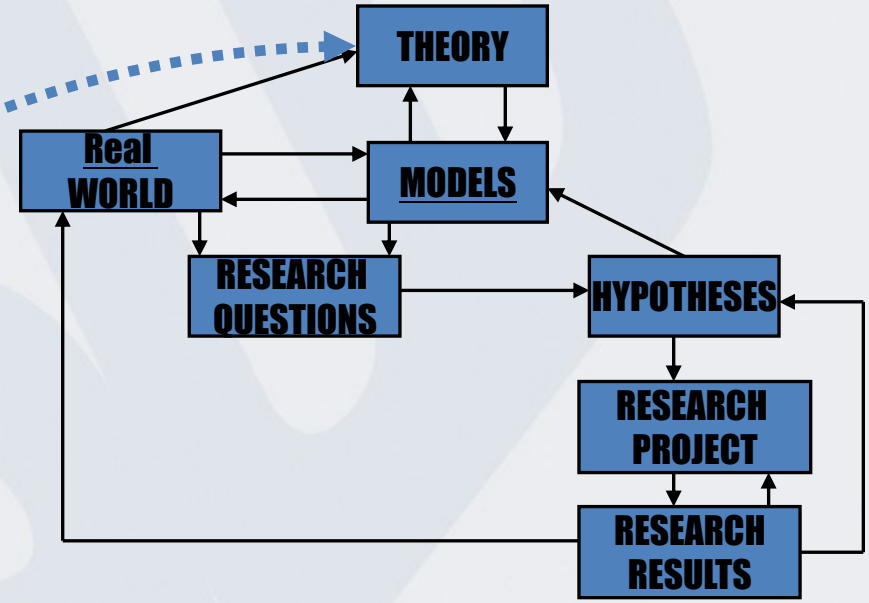
Science, Engineering and Software

- **Science**
  - (knowledge from) the systematic study of the structure and behavior of the physical world, especially by watching, measuring and doing experiments, and the development of theories to describe the results of these activities
  - a particular subject that is studied using the scientific method

- **Scientist**
  - an expert who studies or works in one of the sciences

http://dictionary.cambridge.org/dictionary/british





Science, Engineering and Software

empirical

1: origi

<e

ex·per·i·men·tal·ism

noun \-tə-ˌli-zəm\

: reliance on or advocacy of experimental or empirical principles and procedures; *specifically* **:** instrumentalism:

: a doctrine that ideas are instruments of action and that their usefulness determines their truth

http://www.merriam-webster.com/dictionary/experimentalism

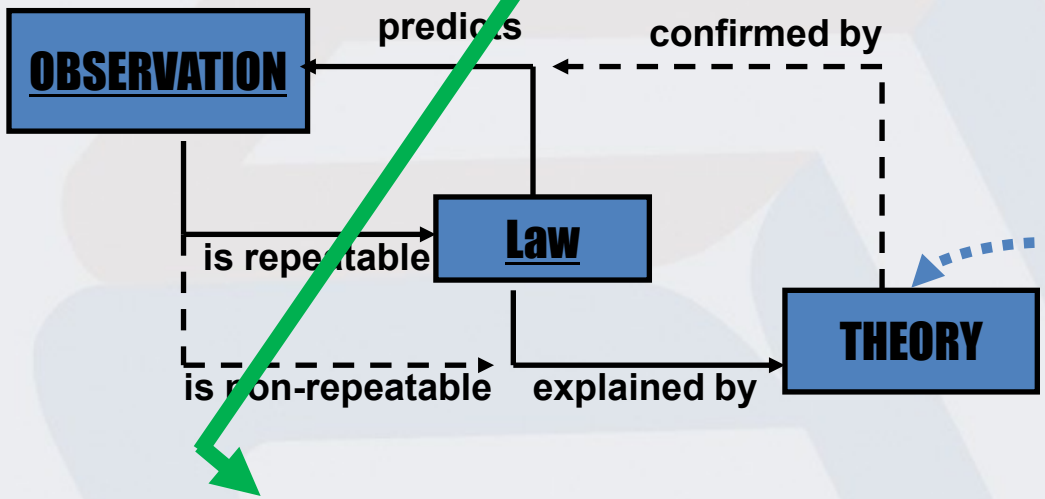
experimental

1 ting to, or based on experience or experiment

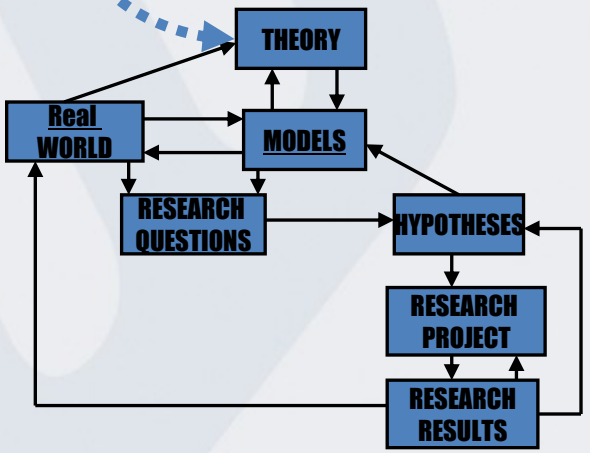
2*a* **:** serving the ends of or used as a means of experimentation <an *experimental* school> *b* **:** relating to or having the characteristics of experiment **:** tentative <still in the *experimental* stage>

Science, Engineering and Software

Empirical



Experimental



Science, Engineering and Software

As it has been recently stated by Mark Harman (keynote at ESEM'12):

"The essence of science and engineering and their considerable achievements rest upon the careful construction of experiments, from which (often painstaking) observations are made…"

*"…**While** real world empirical observations have an important place in the testing of engineering artifacts in situ and in their final operation context, the first duty of scientist and engineer lies within the realm of pure experimentation, under laboratory conditions, where laboratory control serves as a mechanism for removing selection bias, confounding effect and miss observation."*

Current Steps

- **Experimental Software Engineering**
  - Scientific Knowledge Management and Study Strategies
  - Experimentation Environments and Tools
  - Simulation based studies
  - Evidence based software engineering

To support Research and Development regarding:

- **Software Testing**
  - Integration, Planning and Control, Effort Estimation Models,
- **Software Inspections**
  - Reading Techniques, Checklists, Heuristics Based, …
- **Requirements Engineering**
  - Innovation based paradigm, Effort Estimation Models
- **Search Based Software Engineering and Simulation**
  - Software maintenance (software decay), Risk Analysis, Software Architecture
- **Agility in Software Processes**
  - Software Development Process, Testing Process, DevOps, Data Analytics
- **Context Awareness Software Systems**
  - Requirements, Interoperability, Verification and Validation, Systems of Systems, User Experience

ese.cos.ufrj.br

ESE (**E**xperimental **S**oftware **E**ngineering) represents one of the research and development groups of LENS – Laboratory of Software Engineering of PESC at COPPE/UFRJ. It aims at improving software engineering through applying experimentation (scientific method) for the construction, evaluation and evolution of software technologies (processes, methods, techniques, tools, …). ESE also concerns with the field advance, by researching and proposing models for the planning, execution and packaging of primary and secondary studies in software engineering. ESE group believes these are fundamental activities that will contribute to get software engineering closer to the classical engineering and scientific principles.

Final Remarks

There is no silver bullet!!

There is no philosopher's stone!!

Software Systems Development is not alchemy!

Final Remarks

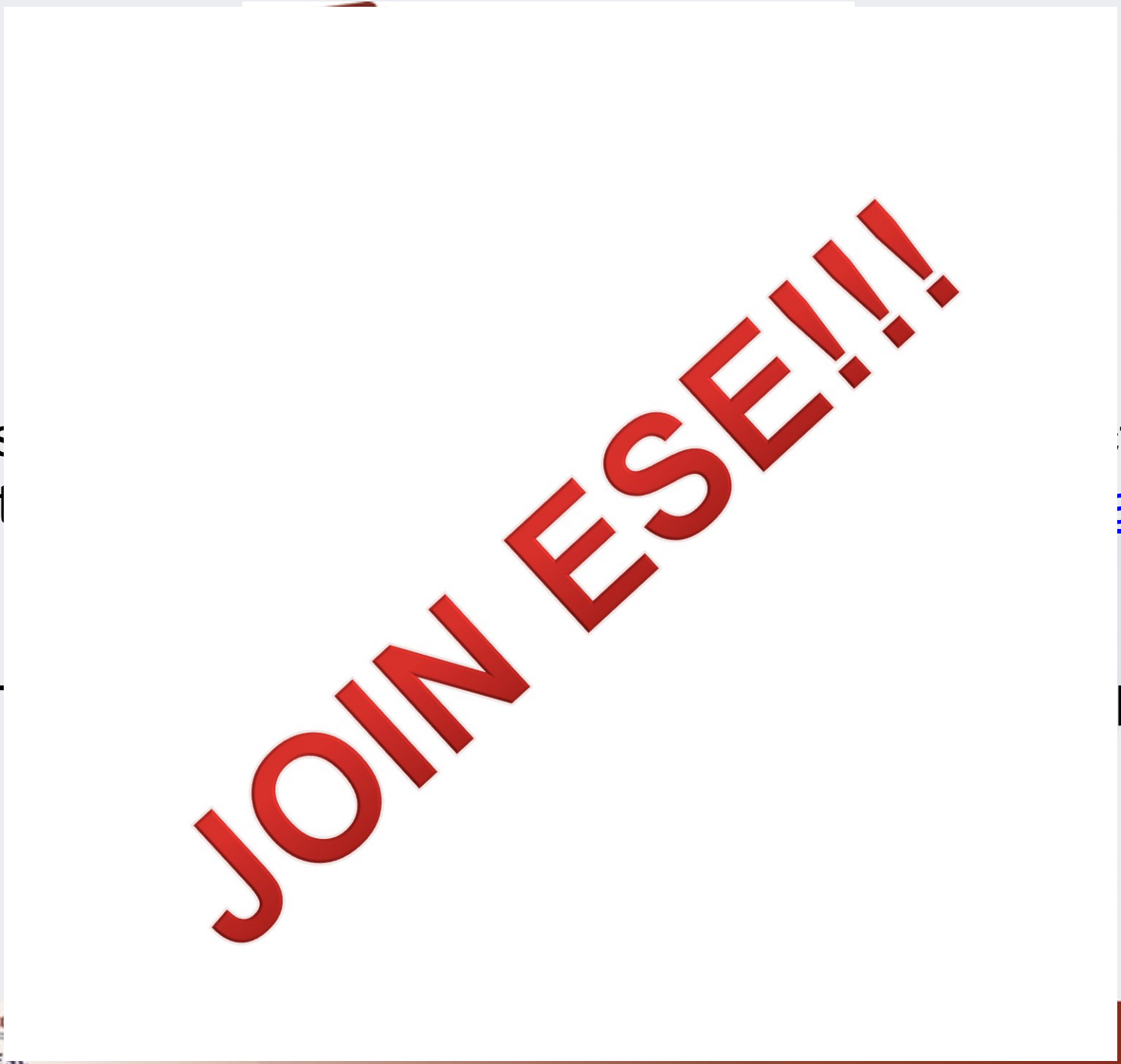
- Software Technology decisions shall be based on evidence.
- Investigations in software engineering share some of the same issues as social science (inspired on…):
  - difficult to collect data
  - non-repeatable
  - difficult to control
- The more we care with the engineering of software systems
  - the more confidence we can have in the quality of our products
  - the better can be our projects
  - the more effective will be our actions

Pos toral

st ao.

-

h!



Software, Engineering and Experimentation

Obrigado por sua atenção.

Guilherme Horta Travassos

Universidade Federal do Rio de Janeiro

COPPE/PESC

CNPq Researcher, **ISERN** Member

ght@cos.ufrj.br

www.cos.ufrj.br/~ght

SEMANA PESC