AN ADAPTATION OF EXPERIMENTAL DESIGN TO THE EMPIRICAL VALIDATION OF SOFTWARE ENGINEERING THEORIES

N. Juristo A. M. Moreno
Facultad de Informática
Universidad Politécnica de Madrid
SPAIN

CONTENTS

1. PROBLEM: Validation of SE ideas
2. About validation of Se ideas:
   - When? & How?
   - Implications
   - Alternative ways
3. What is experimentation
4. Kinds of empirical validation
5. Our proposal: A way to perform formal Lab Experiments
6. Some concepts of Experimental Design (ED)
7. Adaptation of ED concepts to SE:
   - Terminology
   - Parameters and Response Variables
   - Example

PROBLEM

Are we sure about the ideas* we use in software development?

⇩

Let's think about the VALIDATION OF SE IDEAS

*idea = concept, paradigm, method, technique, tool, etc.

VALIDATION OF SE IDEAS
When? How?

When are we sure that a proposed idea works?

⇩

After years of use

How do we validate ideas?

⇩

Natural selection

VALIDATION OF SE IDEAS
Implications

- — Ideas have not been validated before being proposed to the community.
- — Industry uses non-reliable ideas when constructing software.

VALIDATION OF SE IDEAS
Other way?

Is there a different way of doing things?

⇩

Yes, through EXPERIMENTATION

Do other scientific and engineering fields validate ideas the same way we do?

⇩

No, thet do it through EXPERIMENTATION

WHAT IS EXPERIMENTATION?

— Experimentation makes the difference between science & engineering and other academic disciplines

— Quantitative study of phenomena

— Test/ideas against reality (Empirical Validation)

KINDS OF EMPIRICAL VALIDATION

— Formal Laboratory Experiments
— Case Studies: Real projects
— Use of Historical Data

OUR PROPOSAL

A way to perform Formal Lab Experiments

An Approach to Empirical Validation of SE Ideas
in the Laboratory

⇩

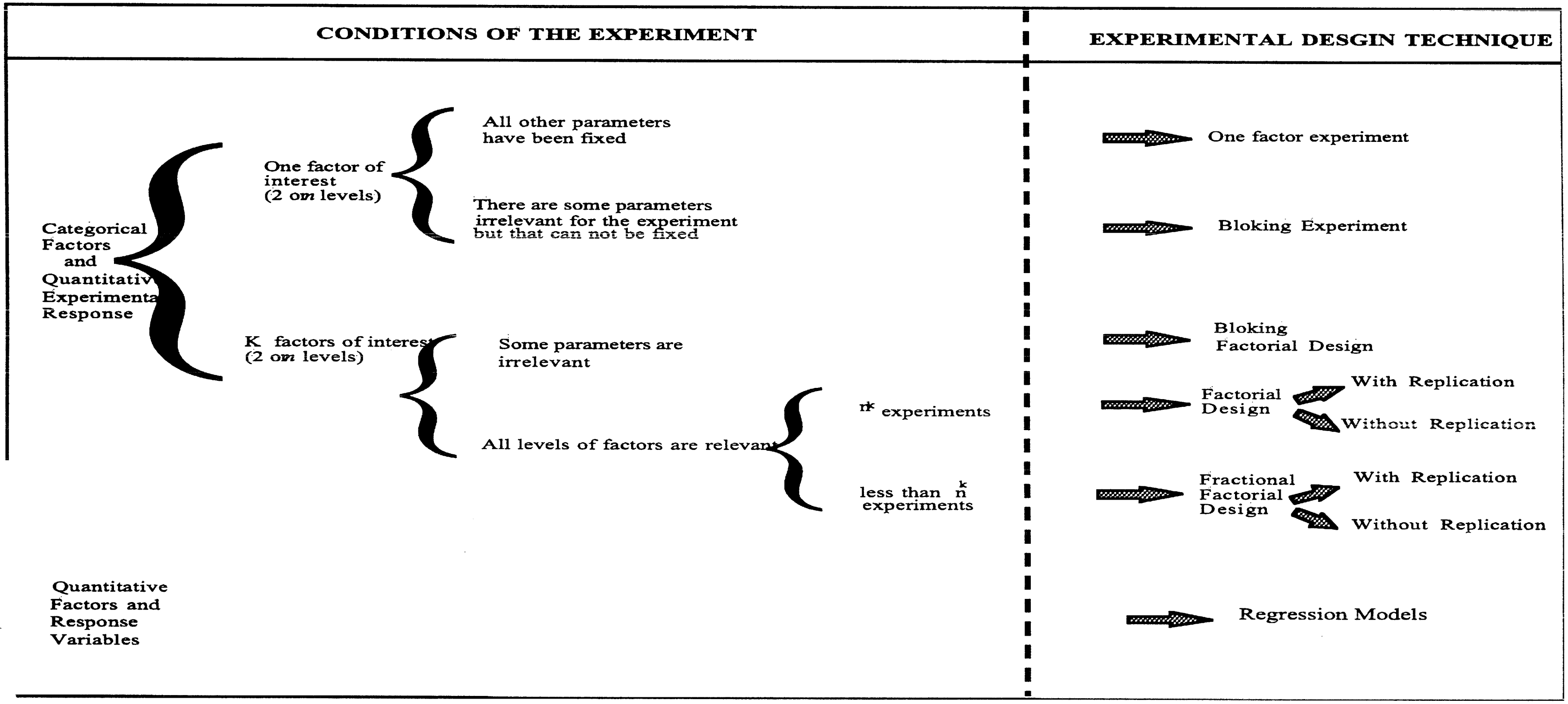
Perform Formal Experiments Using EXPERIMENTAL DESIGN Adapted to SE

SOME CONCEPTS OF EXPERIMENTAL DESIGN

- Proposed early in the 20th century by Sir Ronald Fisher
- It is routinely used by other engineering fields: Chemistry, Agriculture, Pharmaceuticals, ...
- It establishes mathematical foundations to perform experiments, choose the variables of the experiments, collect and analyse data and arrive at conclusions
- The main concept is the idea of statistical significance

SOME CONCEPTS OF EXPERIMENTAL DESIGN

Diferent Techniques for Diferent Situations



ADAPTATION OF ED CONCEPTS TO SE

Terminology

Concept	Description	Application in SE
Experimental unit	Entity used to conduct the experiment	Software projects
Parameters	Characteristic (qualitative or quantitative) of the experimental unit	See
Response variable	Datum to be measured during the experimental unit	See kect table. Note there are no response variables relating to "problem". This is because response variables are data that can be measured *a posteriori*, that is, once the experiment is complete. In the case of SE, the experiment involves development (in full or in part) of a software system to which particular technologies are applied. The characteristics of the *problem* to be solved are the experiment input data, that is, they stipulate how it will be performed. As such, they are parameters and factors of the experiment. However, they are not experiment output data that can be measured and, thus, do not generate response variables.
Factor	Parameter that affects the response variable and whose impact is of interest for the study	Factors are chosen from parameters in table 3. Factors have different values during the experiment
Level	Possible values or alternatives of the factors	Values of factors in table 3
Interaction	The effect of one factor depends on the level of other	Relations between the parameters in table 3, for example problem complexity and product complexity
Replication	Repetition of each experiment to be sure of the measurement taken of the response variable	Repeatability in SE must be based on analogy, not on identity; the different experiments will consist of similar problems, similar processes, similar teams, etc.
Design	Specification of the number of experiments, selection of factors, combinations of levels of each factor for each experiment and the number of replications per experiment	The design will indicate the number of software projects, factors and their alternatives that will be used during experimentation, as well as the number of replications of the experiments, based on analogy.

ADAPTATION OF ED CONCEPTS TO SE

Experiments Parameters

PARAMETERS			
PROBLEM (User need)	PROCESSES of construction employed	PERSONS (team of developers)	PRODUCT
– Definition (poorly/well defined problem) – Need volatility (very/hardly/non volatile need) – Ease of understanding (problem well/poorly/fairly well understood by developers) – Problem complexity – Problem type (data processing, knowledge use, etc.), – Problem-solving type (procedural, heuristic, real-time problem solving, etc.) – Domain (aeronautics, insurance, etc.) – User type (expert novice, etc.)	– Maturity – Description (set of phases, activities, products, etc.) – Relationship between members (definition of interrelations between team members) – Automation (in which phases or activities tools are used) – Risks	– Number of members – Division by positions (no. of software engineers, programmers, project managers, etc.) – Years of experience of each member in development – Experience of each member in the problem type – Experience of each member in the software process applied – Background of each member (discipline of origin) – Type of relationship between members (all in the same building, same town, subcontracts, etc.)	– Type of life cycle to be followed – Software type (OO, databases, real time, expert system, etc.) – Size – Complexity – Architecture/Organization – Hardware platform – Interaction with other software – Processing conditions (batch, on-line, etc.) – Security requirements – Response-time requirements – Documentation required – Help required

ADAPTATION OF ED CONCEPTS TO SE
Response Variable

RESPONSE VARIABLES			
PROBLEM	PROCESS	PERSONS	PRODUCT

	– Schedule deviation
	– Budget deviation
	– Compliance with construction process
	– Products obtained (do they comply with the process stipulations?)
		– Productivity
		– User satisfaction
		  – usability
		  – usefulness
			– Correctness of products obtained (no. of errors, etc.)
			– Validity of the products (compliance with customer expectations)
			– Portability, Maintainability, Extendibility, Performance, Flexibility, Interoperability,…

ADAPTATION OF ED CONCEPTS TO SE
Example of Factorial Design

FACTORS:	Development Paradigm (new/00) Process Maturity (high/low) Problem Complexity (complex/simple)
RESPONSE VARIABLE:	Number of errors detected three months after deployment
ED TECHNIQUE:	Factorial Design
RESULT:	Correctness is better when the new paradigm is used

ADAPTATION OF ED CONCEPTS TO SE

Example of Factorial Design

I	A	B	C	AB	AC	BC	ABC	φ
1	-1	-1	-1	1	1	1	-1	14
1	1	-1	-1	-1	-1	1	1	22
1	-1	1	-1	-1	1	-1	1	10
1	1	1	-1	1	-1	-1	-1	34
1	-1	-1	1	1	-1	-1	1	46
1	1	-1	1	-1	1	-1	-1	58
1	-1	1	1	-1	-1	1	-1	50
1	1	1	1	1	1	1	1	86
320	80	40	160	40	16	24	9	total
40	10	5	20	5	2	3	1	total/8

$SST = 2^2 (C_A^2 + C_B^2 + C_C^2 + C_{AB}^2 + C_{AC}^2 + C_{BC}^2 + C_{ABC}^2 =$

$2(102 + 52 + 202 + 52 + 22 + 32 + 12) =$

$800 + 200 + 3200 + 32 + 72 + 8 = 4512$

A (Development Paradigm): 800/4512 = 18%

B (Process Maturity): 200/4512 = 4%

C (Software Complexity): 3200/4512 = 71%

AB: 200/4512 = 4%

BC: 32/4512 = 1%

AC: 72/4512 = 2%

ABC: 8/4512 = 0%

$Correctness_{\text{paradign = new}}$ = 14+46+10+50 / 4 = 30 = 40 - 10

$Correctness_{\text{paradign = oo}}$ = 22+58+34+860 / 4 = 50 = 40 - 10