

The Impact of the PSP on Software Quality:

Eliminating the learning effect threat through a controlled experiment (and beyond)

Fernanda Grazioli
Leticia Pérez

Diego Vallespir
Silvana Moreno

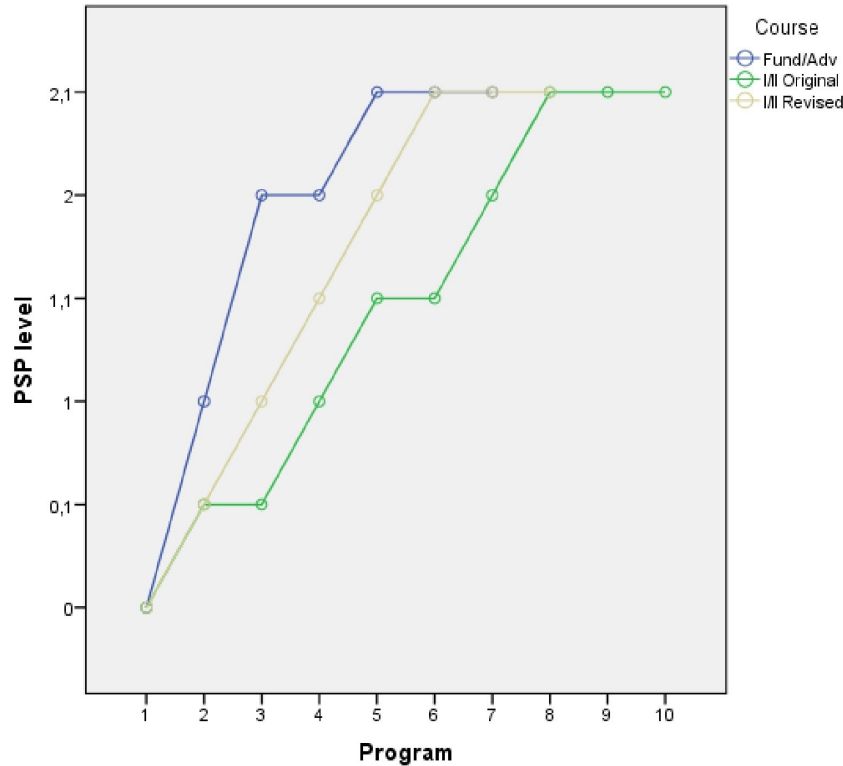
Agenda

- Introduction
- Experiment Design
- Results
- And Beyond
- Other things that we are doing

What do we know about PSP results?

- Statistical analysis of the evolution of the results showing that during the course the engineer improves his/her performance
 - Then it can be statistically inferred that the **PSP is responsible for the quality improvement**
 - But, in fact, this is only one possible reason
- Possible reasons for the improvement
 - Learning the domain of the PSP course exercises
 - Repeat the programming recording data
 - Etc.

One Prior Work



- A Cross Course Analysis of Product Quality Improvement with PSP
Grazioli and Nichols
TSP Symp. 2012
- An Analysis of Student Performance during the Introduction of the PSP: An Empirical Cross Course Comparison
Grazioli, Nichols and Vallespir
TSP Symp. 2013

But repetition still has a chance of be an important factor in the improvements

Our Research

Research question:

Are the performance improvements observed in the PSP courses due to the introduction of the phases and techniques of the PSP or to programming repetition?

We want to observe changes in product quality

Our Experiments

- We designed and performed controlled experiments with software engineering undergraduate students at the Universidad de la República.
- The students performed the exercises from the PSP for Engineers I/II course without applying the PSP techniques.

Experiment Setup - Measures

- We use three measures to evaluate the quality of the products:
 - Defect density in compile
 - Defect density in unit test
 - Total defect density

Experiment Setup - Hypotheses

- A statistical hypothesis is an assumption about a population parameter. Hypothesis testing refers to the formal procedures used in experimentation to accept or reject statistical hypotheses
- The hypotheses aim at knowing if comparing a developed program to another one developed previously, the software engineer improves his performance in any of the aspects mentioned
- So, we compare programs by pairs to find if the changes in each performance dependent variable are statistically significant

Experiment Setup - Subjects

- Computer science students in the final years (4th or 5th year of the career)
- With at least four courses on programming
- The students participate in the experiment in order to obtain credits. So, they are motivated
- They don't know they are taking part in an experiment
- They know that the data they collect will be used in research work

Experiment Setup - Materials

- PSP 0 and PSP0.1 process scripts
- Requirements of PSP programs 1 to 8 (from PSP for engineers course I/II)
- PSP tool for data collection (MS Access)

Experiment Setup - Experiment Design

- 22 students (12 during 2012 and 10 during 2013)
- Each one developed 8 programs (same programs for each student and developed in the same order)
- Repeated measure design
 - Several measures are taken for the same subjects
- PSP0 used in the first program
- PSP0.1 for the other programs
- We are only collecting data and not introducing PSP phases or techniques (reviews, design, PROBE, etc.)

Experiment Setup - Environment

- The students perform the assignments individually in their houses
 - But they are permanently monitored by a tutor.
 - There is a constant feedback with the tutor (mail, phone calls or meetings)
 - Students are not in a time-limited classroom, so in this way the time records and the amount of defects found are not biased by the available time of class.

2012 Experiment Analysis

	2	3	4	5	6	7	8
1							
2							
3							
4							
5							
6							
7							

- We made hypothesis testing with comparing program by program
- For example: defect density in UT is statistical better/worse in program 2 or program 5

2012 Experiment Analysis

	2	3	4	5	6	7	8
1							
2							
3							
4							
5							
6							
7							

- We made hypothesis testing with comparing program by program
- For example: defect density in UT is statistical better/worse in program 2 or program 5
- To present the results we use cells in gray (not evidence), red (evidence of deteriorations), green (evidence of improvement)

2012 Experiment Results

- Basically, we did not find improvement
- But
 - We need more subjects
 - We want to compare our experiment with data from the PSP courses

Comparing with PSP Course

- We compared our results with PSP for Engineers I/II courses
 - We want to know whether the quality improvement is because of the PSP practices or because of other characteristics
- PSP levels in the course

Group	Program assignments
PSP0	1-2
PSP1	3-4
PSP2	5-8

Results - Hypotheses test

- In a context of few samples and repeated measures the most suitable statistical hypotheses test is the Wilcoxon signed-ranks test
- We used the 2-tailed Wilcoxon test because we do not know a priori if the dependent variables will increase or reduce their values
- In the results
 - Each cell contains the p-value (2-tailed) of the Wilcoxon test.
 - The cells in green or red indicate that the null hypothesis has been rejected ($p \leq 0.05$).
 - The green ones indicates improvement
 - The red ones indicate the opposite
 - The gray cells indicate that it has not been possible to reject the null hypothesis.

Defect Density Tests

PSP Course

COMPILE

Level	PSP1	PSP2
PSP0	p=0.000 , d=0.7	p=0.000 , d=1.4
PSP1		p=0.000 , d=1.0

UNIT TESTING

Level	PSP1	PSP2
PSP0	p=0.001 , d=0.5	p=0.000 , d=1.0
PSP1		p=0.021 , d=0.4

TOTAL DEFECTS

Level	PSP1	PSP2
PSP0	p=0.000 , d=0.9	p=0.000 , d=0.7
PSP1		p=0.072

Experiment

Group	progs3-4	progs5-8
progs1-2	p=0.04 , d=0.3	p=0.001 , d=0.6
progs3-4		p=0.296

Group	progs3-4	progs5-8
progs1-2	p=0.000 , d=0.9	p=0.001 , d=0.7
progs3-4		p=0.012 , d=0.4

Group	progs3-4	progs5-8
progs1-2	p=0.000 , d=0.6	p=0.000 , d=0.7
progs3-4		p=0.961

Results Sum Up

- Since the experiment does not change the level of PSP used (PSP0.1 from program 2 to 8) the results of this experiment indicate that the programming repetition:
 - Do not continuously improve defect density in compile and UT
 - The defect density in UT deteriorates in the last four programs
- The experiment contribute to the elimination of an important threat to the validity of different experiments performed with the PSP
- Besides, this experiment shows that without the **adequate practices** the quality of software cannot be improved by the simple reason of the programming learning effect

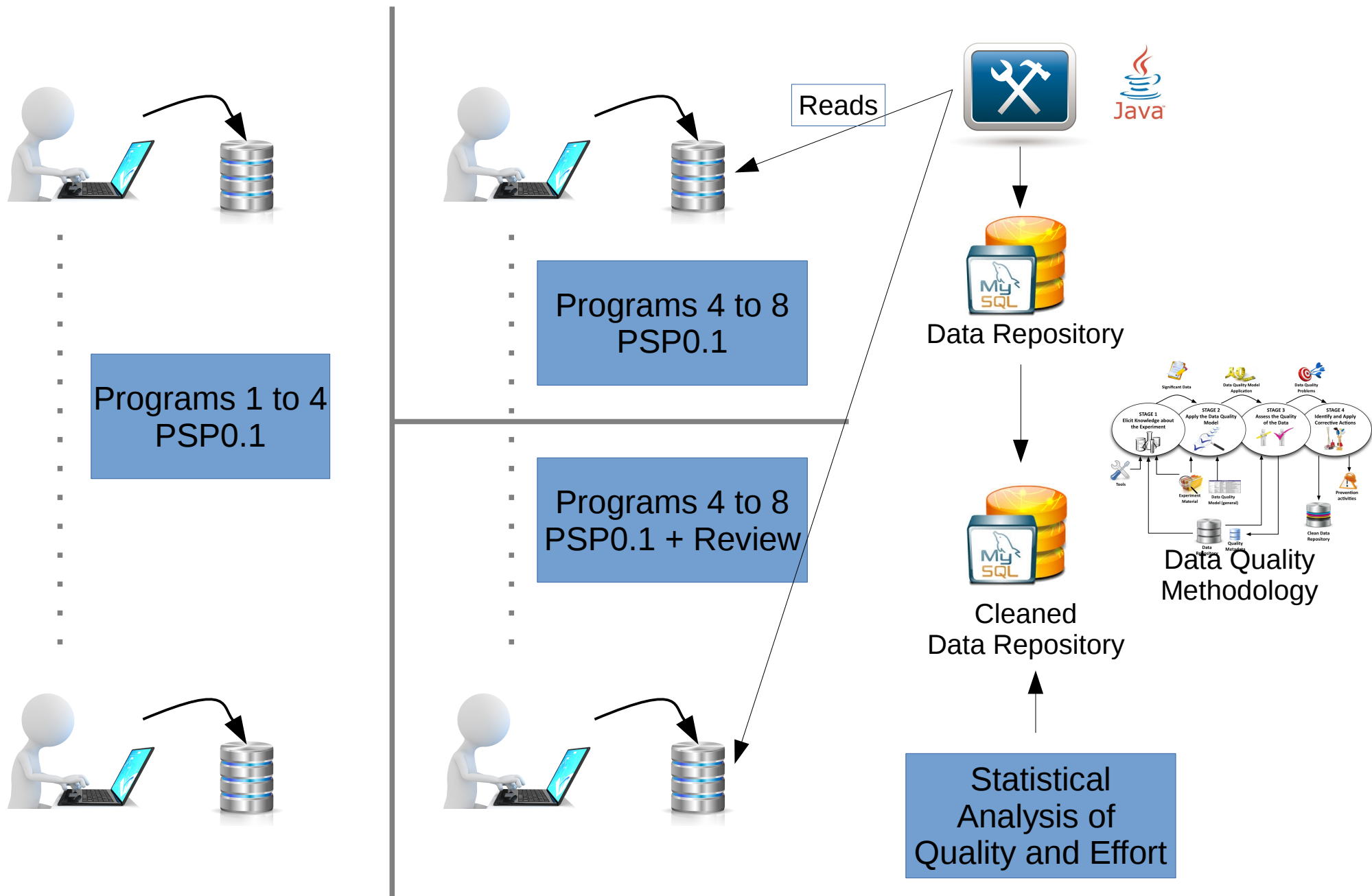
And Beyond

- We want to know the effects on quality and effort of personal reviews and detailed design
- By designing and executing experiments
 - Isolating the techniques
- By analyzing more PSP data
 - With the techniques embedded in PSP
- By analyzing TSP data
 - With the techniques embedded in TSP

Personal Reviews Research

- Mapping study (ongoing research)
 - There are few articles that study personal reviews outside the PSP
 - Why?
 - We don't know why this technique is not study so much, but we have some ideas
 - It is not common to register defects injected and removed at this level
 - Maybe, the software engineering community is not really aware of the importance of this simple technique

Personal Reviews Experiments



Statistical Analysis of Quality and Effort

- Effort is easy to calculate
- However, quality is not
 - Of course students performing reviews will find less defects in UT per KLoc
 - But, we do not know if the density of defects in the final program is less or not (I'm sure it is, but this is not enough)
- We can analyze
 - Individual improvements as we did in the experiment
 - Comparing the two groups (with and without reviews) with the limitations mentioned for quality analysis

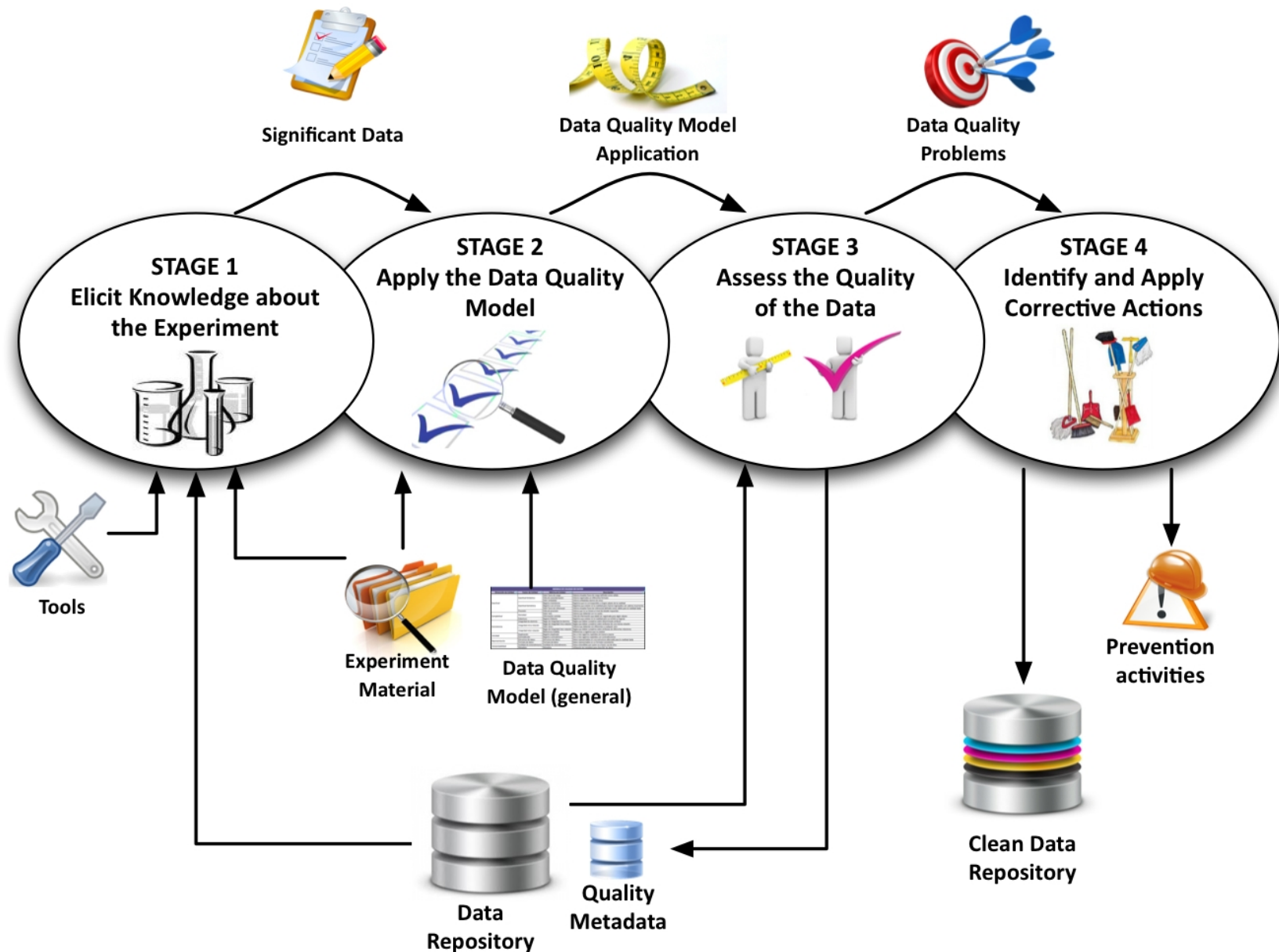
How Can We Know the “Final” Quality

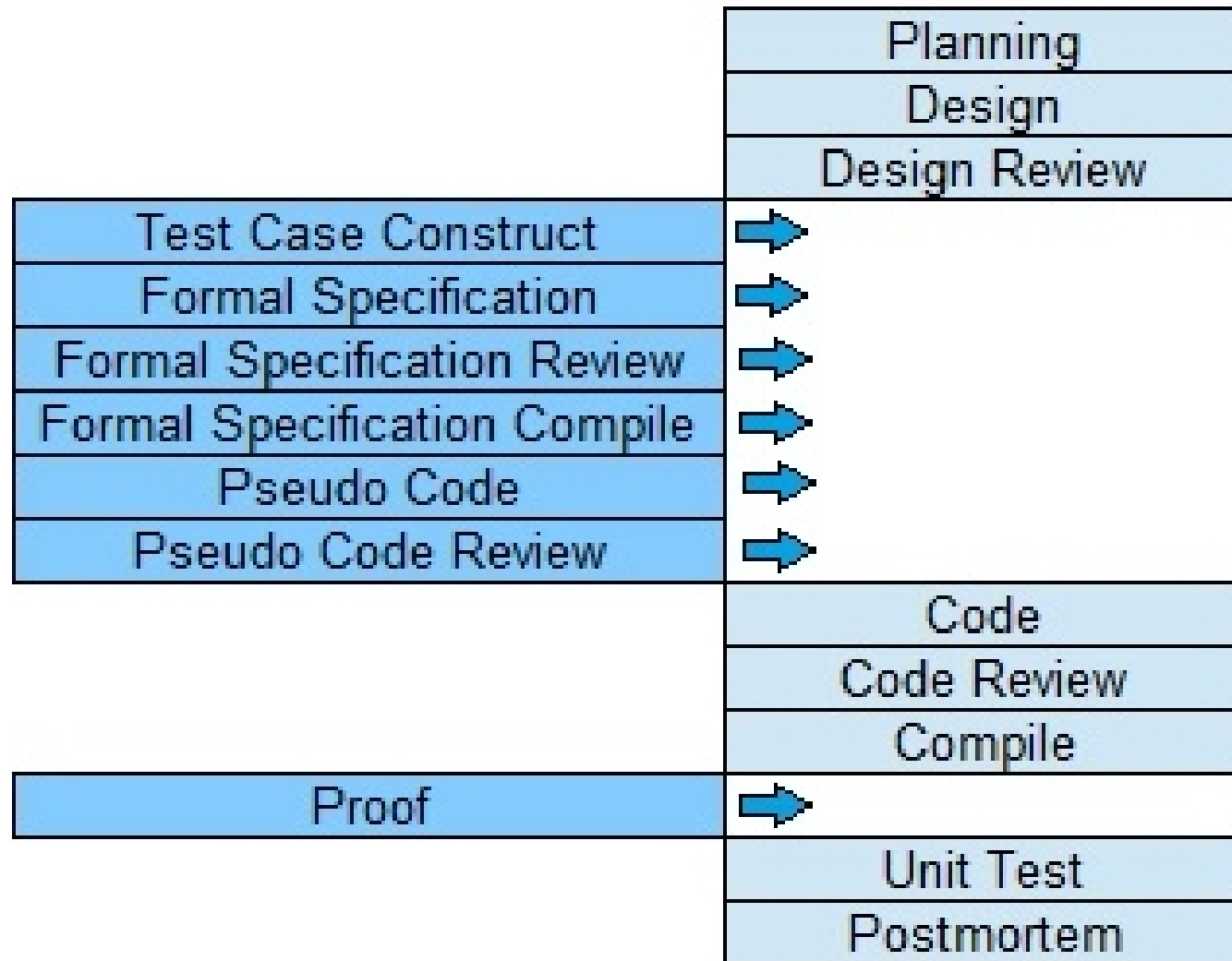
- Executing some predefined tests
 - We need to change the requirements of the PSP programs a little bit, defining some specific interface with the “user”
 - Low cost for the researchers
- Executing reviews on the programs
 - High cost for the researchers
- Changing the programs 1 to 8 in order to have more dependencies between them
 - Analyze the defects after release

Other Things that We are Doing

- Measuring the data quality of PSP data
- An adaptation of PSP that incorporates verified design by contract (PSP_{VDC})

Data Quality Methodology





Questions

Diego Vallespir

dvallesp@fing.edu.uy

Grupo de Ingeniería de Software

Universidad de la República

Uruguay