

# The Experimental Process

---

## (01 OPJIU) Empirical Methods in Software Engineering

<http://softeng.polito.it/EMSE/>



**SoftEng**  
<http://softeng.polito.it>

Version 1.6.0  
© Marco Torchiano, 2017






This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

To view a copy of this license, visit

<http://creativecommons.org/licenses/by-nc-nd/4.0/>.

You are free: to copy, distribute, display, and perform the work

Under the following conditions:

-  **Attribution.** You must attribute the work in the manner specified by the author or licensor.
-  **Non-commercial.** You may not use this work for commercial purposes.
-  **No Derivative Works.** You may not alter, transform, or build upon this work.
  - For any reuse or distribution, you must make clear to others the license terms of this work.
  - Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

---

# THE EXPERIMENTAL PROCESS

Wohlin et. al. 2000

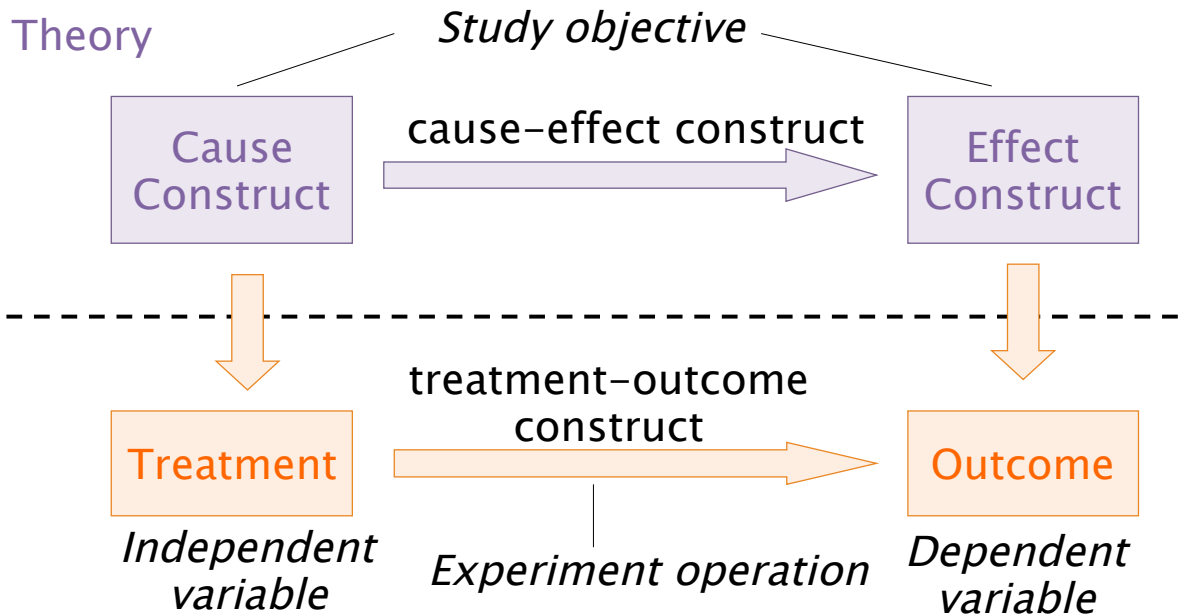
## Empirical method

---

- From observation find an explanation
- Formalize it into a theory
- Formulate an hypothesis
- Test it with a study

# Experimental principles

---



## Observation

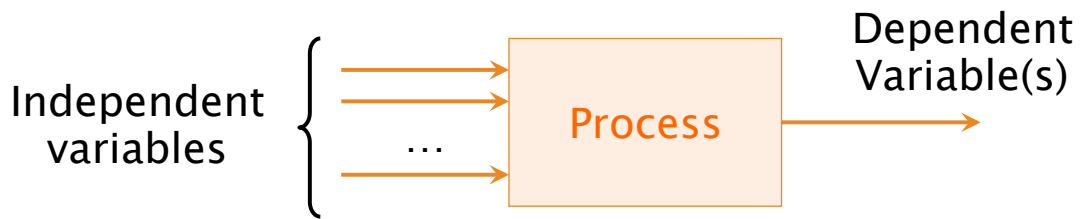
# Glossary

---

- **Construct (conceptual)**
  - ♦ Broad concepts or topics of study
    - Abstract
    - Not directly observable
    - May be complex (have multiple parts)
  - ♦ E.g. quality, productivity, skill
- **Variable (Operational Construct), Measure, Metric**
  - ♦ Precise definition
  - ♦ Procedure to measure

# Glossary

---



- Dependent (output, response) variable:
  - ◆ Quantities observed in the study
  - ◆ E.g. LOC/day
- Independent (input) variable:
  - ◆ Quantities controlled and monitored
  - ◆ E.g. years of experience, development method

# Glossary

---

- Factor
  - ◆ An input variable whose effect on the output we want to study
  - ◆ E.g. development method
- Treatment (Level)
  - ◆ A particular value of a factor
  - ◆ E.g. upfront design vs. incremental design

# Glossary

---

**Subject** performs a **task** with an **object**

- ◆ Experimental unit of observation
- The treatment may be applied to:
  - ◆ Task
    - E.g. Develop using a given *methodology*
  - ◆ Object
    - E.g. Requirement with a given *notation*
  - ◆ Subject
    - E.g. Developer with a particular *skill/training*

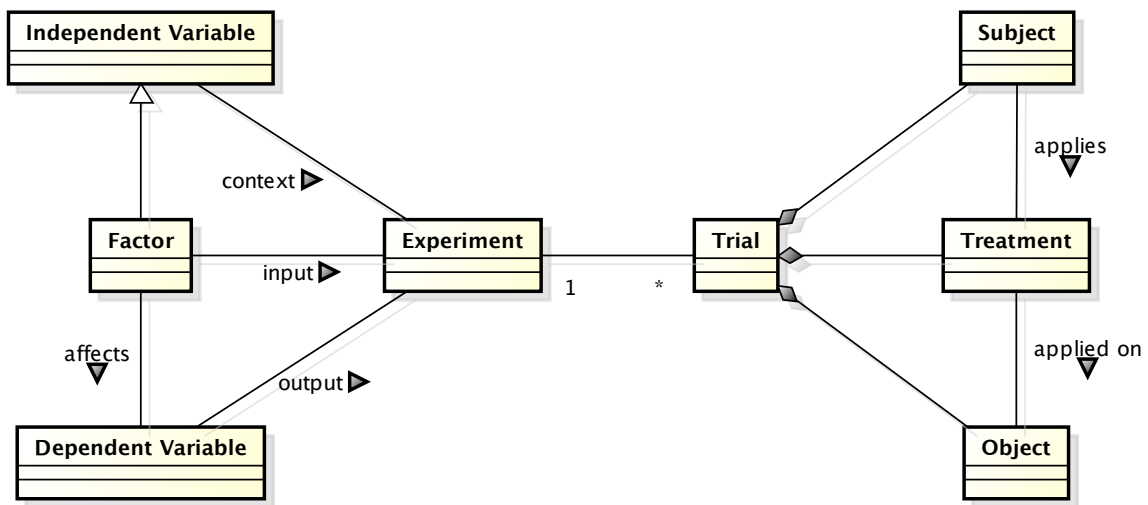
# Glossary

---

- Trial (experimental unit)
  - ◆ A combination of (Subject, Task, Object, Treatment)
  - ◆ Subject + Treatment
    - Task and Object counted as part of the treatment
    - Object: Software artifact
- An experiment typically involves several trials

# Experimental Process Concepts

---



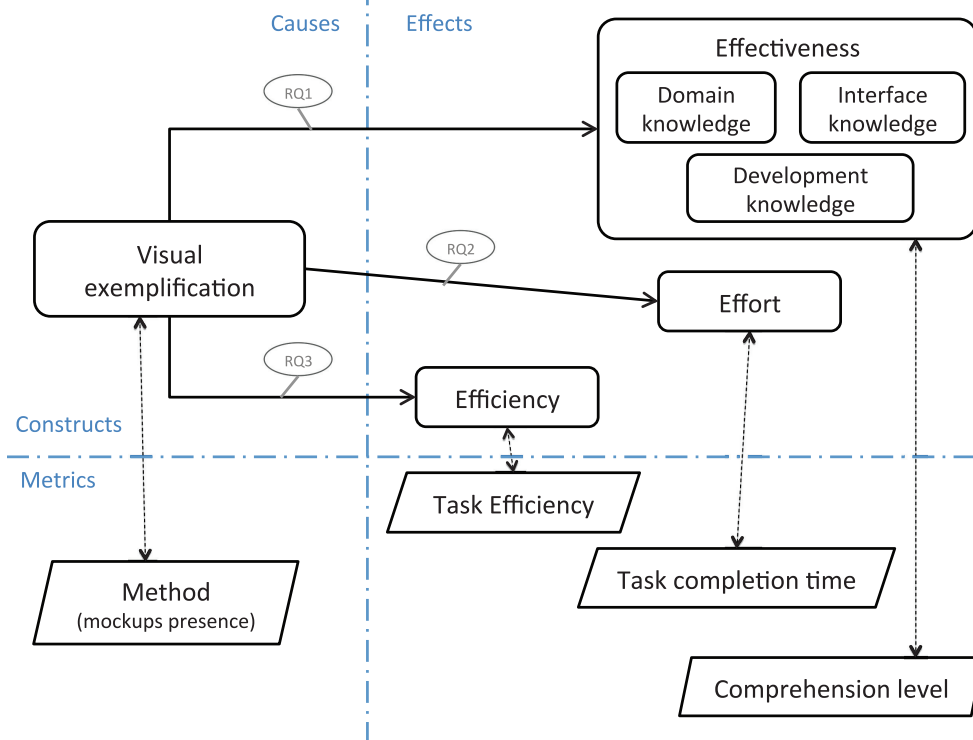
## A running example

---

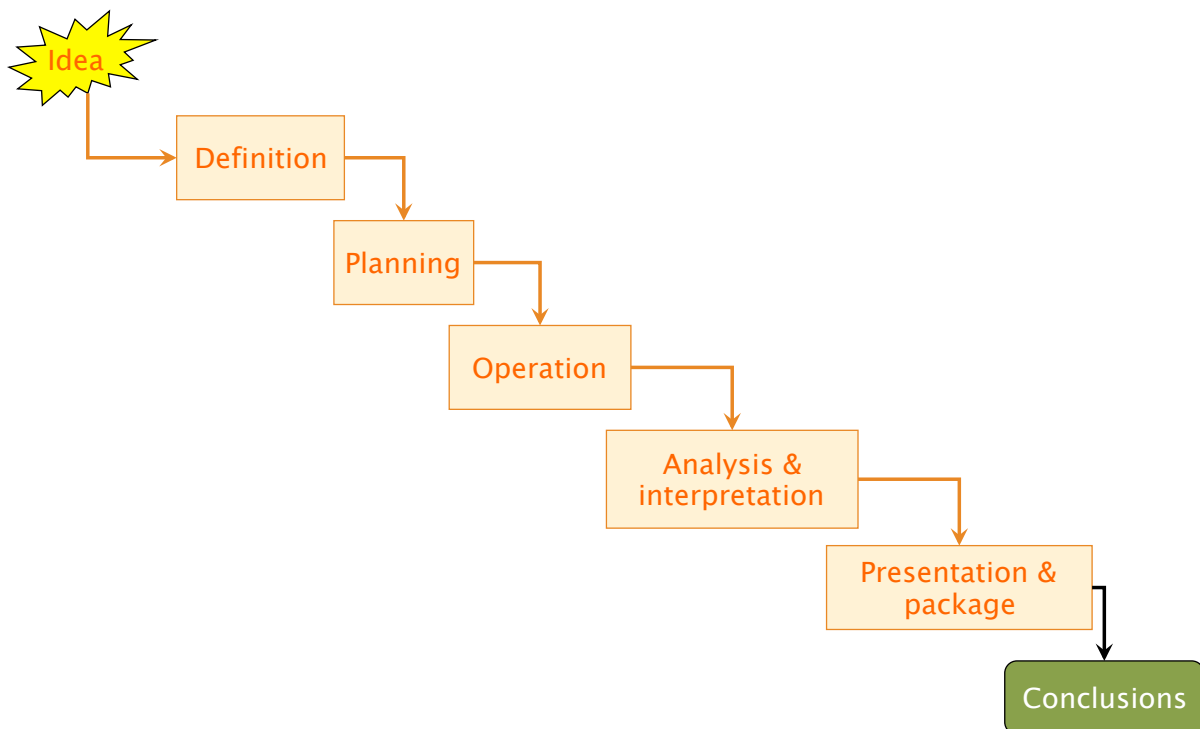
Ricca et al., **Assessing the Effect of Screen Mockups on the Comprehension of Functional Requirements**. ACM Transactions on Software Engineering and Methodology, 24(1), 38pp, 2014

<http://doi.acm.org/10.1145/2629457>

# Conceptual model



# Experimental process



# Experimental process steps

---

- Definition
  - ◆ Goals and objectives of the study
- Planning
  - ◆ Define context
  - ◆ Formulate hypotheses
  - ◆ Identify input and output variables
  - ◆ Design the study
  - ◆ Analyze threats to validity

# Experimental process steps

---

- Operation
  - ◆ Preaparation
  - ◆ Execution
  - ◆ Data validation
- Analysis and interpretation
  - ◆ Data understanding
    - Descriptive statistics, EDA
  - ◆ Possible data reduction
  - ◆ Hypothesis testing
  - ◆ Results interpretation



# Experimental process steps

---

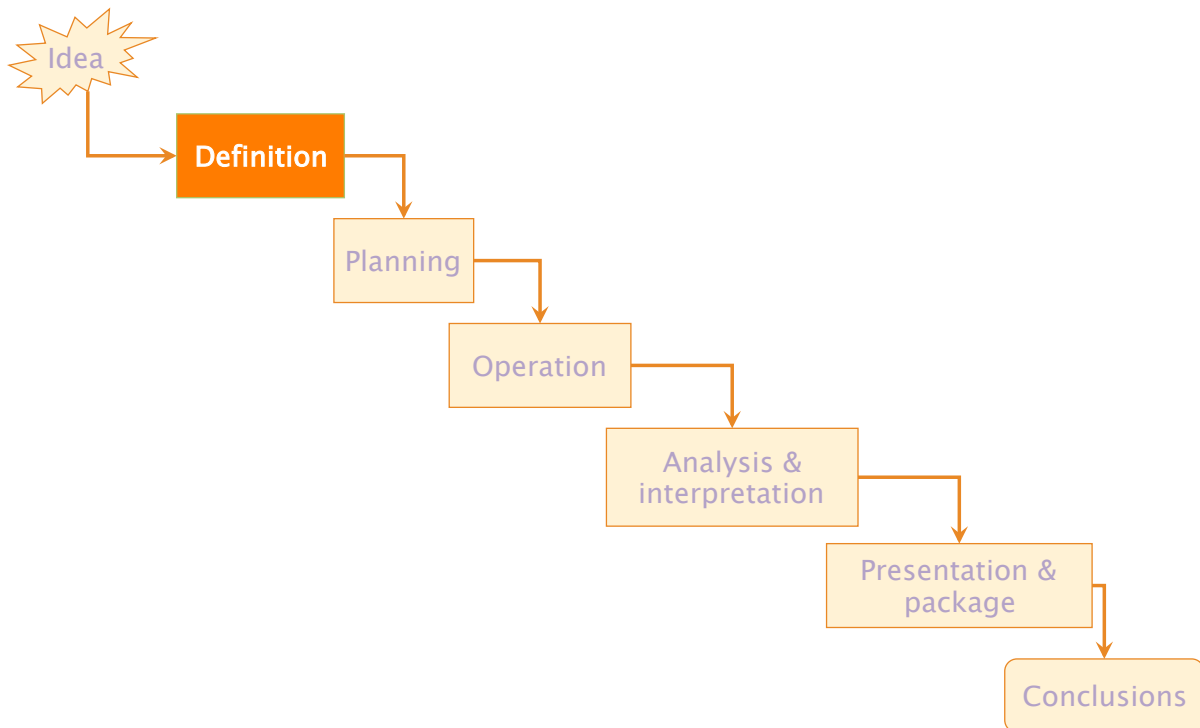
- Presentation and package
  - ◆ Document results
  - ◆ Prepare lab-package to enable replications
  - ◆ Sum up lessons learned

---

## EXPERIMENT DEFINITION

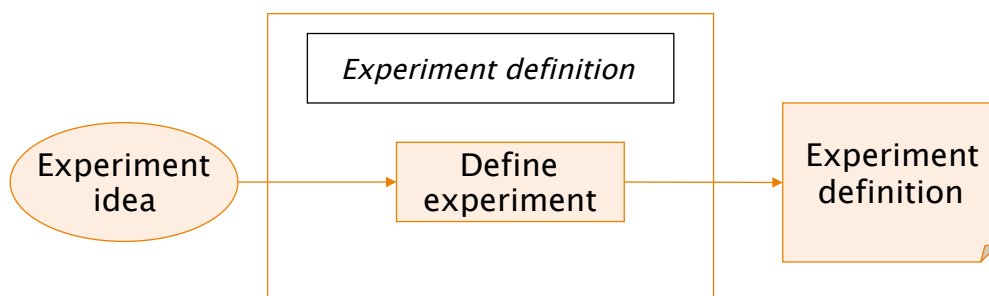
# Definition

---



## Experiment definition: overview

---



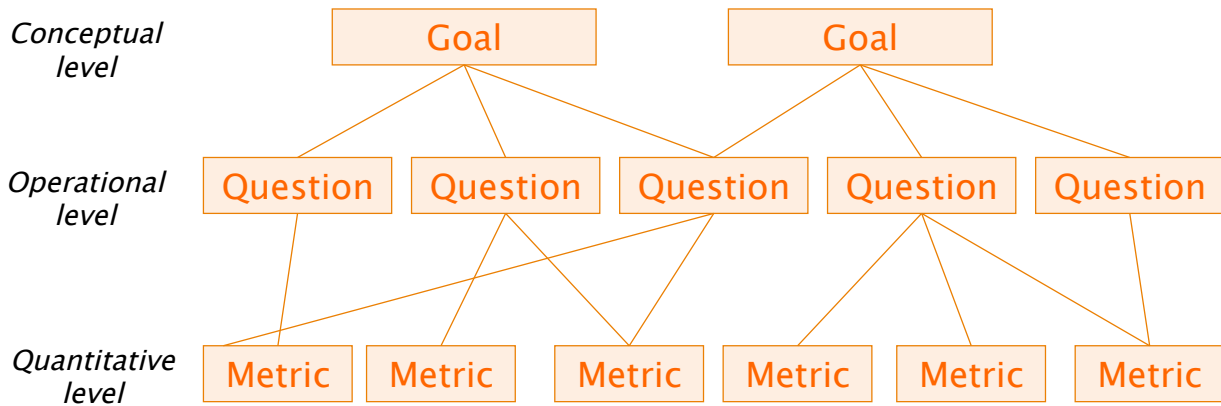
The definition determines the foundation of the experiment (*what* and *why*).

At this level, hypotheses should be clear but not formally described

# Goal-Question-Metric (GQM)

---

- Research approach



[Basili94b][Solingen99]

## Goal definition template

---

Analyze

Objects(s) of study

for the purpose of

Purpose

with respect to their

Quality focus

from the point of view of

Perspective

in the context of

Context

# Goal definition examples

Object of study	Purpose	Quality		
		focus	Perspective	Context
Product	Characterize	Effectiveness	Developer	Subjects
Process	Monitor	Cost	Modifier	Objects
Model	Evaluate	Reliability	Maintainer	
Metric	Predict	Maintainability	Project manager	
Theory	Control	Portability	Corporate manager	
	Change	Comprehension	Customer	
			User	
			Researcher	

## Goal definition example

*“Analyze the PBR and checklist techniques for the purpose of evaluation with respect to effectiveness and efficiency from the point of view of the researcher in the context of M.Sc. And Ph.D. students reading requirements documents”*

Regnell et al. Are the Perspectives Really Different?

# Goal definition example

---

Analyze	Objects(s) of study <b>PBR and checklist techniques</b>
for the purpose of	Purpose <b>Evaluation (and comparison)</b>
with respect to their	Quality focus <b>effectiveness and efficiency</b>
from the point of view of	Perspective <b>the researcher</b>
in the context of	Context <b>M.Sc. and Ph.D. students reading requirements docs</b>

# Goal definition example

---

*analyze the use of stereotyped UML diagrams, with the purpose of evaluating their usefulness in Web application comprehension for different categories of users. The quality focus is to ensure high comprehensibility, while the perspective is both of Researchers, evaluating how effective are the stereotyped diagrams during maintenance for different categories of users, and of Project managers, evaluating the possibility of adopting the Web modeling technique WAE in her organization, depending on the skills of the involved developers. The context of the experiment consists of two Web applications (objects) and four groups of subjects: research associates, students from an undergraduate course, and students from two graduate courses.*

▪ Ricca et al. How Developers' Experience and Ability Influence Web Application Comprehension Tasks Supported by UML Stereotypes: A Series of Four Experiments

# Goal definition example

---

Analyze	Objects(s) of study stereotyped UML diagrams
for the purpose of	Purpose evaluating their usefulness
with respect to their	Quality focus comprehension
from the point of view of	Perspective researcher and project manager
in the context of	Context research associates, undergraduate students graduate students

# Goal definition example

---

- *Analyze* the use of screen mockups for the purpose of understanding their utility with respect to the effectiveness in comprehending requirements and the effort and the efficiency in performing comprehension tasks from the point of view of the requirements analyst, developer, and customer in the context of students reading two requirements specification documents for desktop applications.

# Goal definition example

---

Analyze	Objects(s) of study the use of screen mockups
for the purpose of	Purpose understanding their utility
with respect to	Quality focus effectiveness in comprehending requirements and the effort and the efficiency in performing comprehension tasks
from the point of view	Perspective the requirements analyst, developer, and customer
in the context of	Context students reading requirements specification for desktop applications

<http://softeng.polito.it>

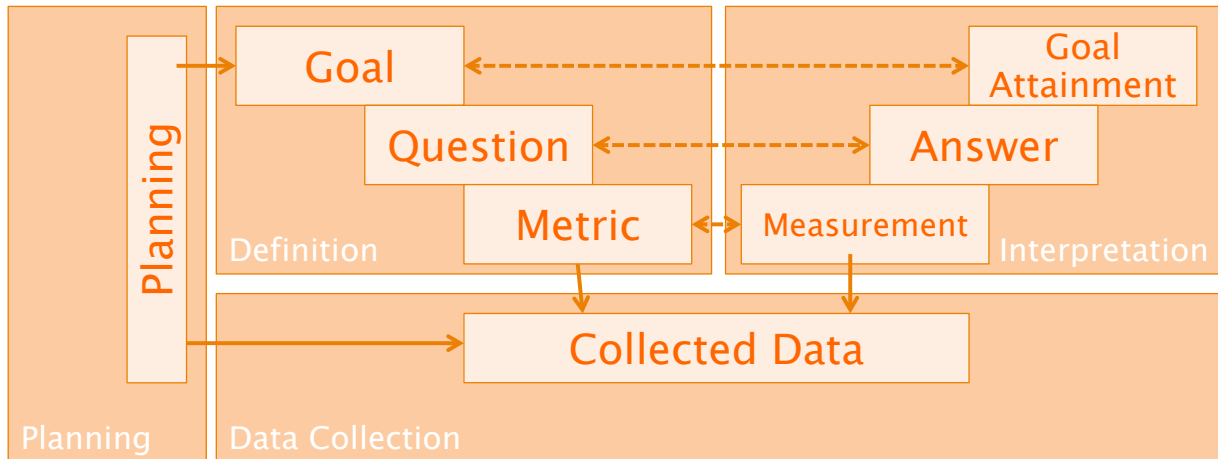
## Example questions

---

- RQ1. Does the requirements comprehension *effectiveness* vary when use cases are provided in conjunction with screen mockups?
- RQ2. Does the *effort* required to complete a comprehension task vary when use cases are provided in conjunction with screen mockups?
- RQ3. Does the *efficiency* in performing a comprehension task vary when use cases are provided in conjunction with screen mockups?

# GQM in perspective

---



## Questions

---

- Refinement of goals to a more operational level
  - ◆ By answering the question one should be able to conclude whether the goals has been achieved
- Expected answers can be formulated as (high level) hypotheses
- Questions may focus on different aspects of the goal



# Example – Goal

---

- Goal: reliability
  - ♦ **Analyze** the product and process
  - ♦ **For the purpose** of characterizing
  - ♦ **With respect to** reliability and its causes
  - ♦ **Form the point of view of** the software development team
  - ♦ **In the context of** project A

# Example – Questions

---

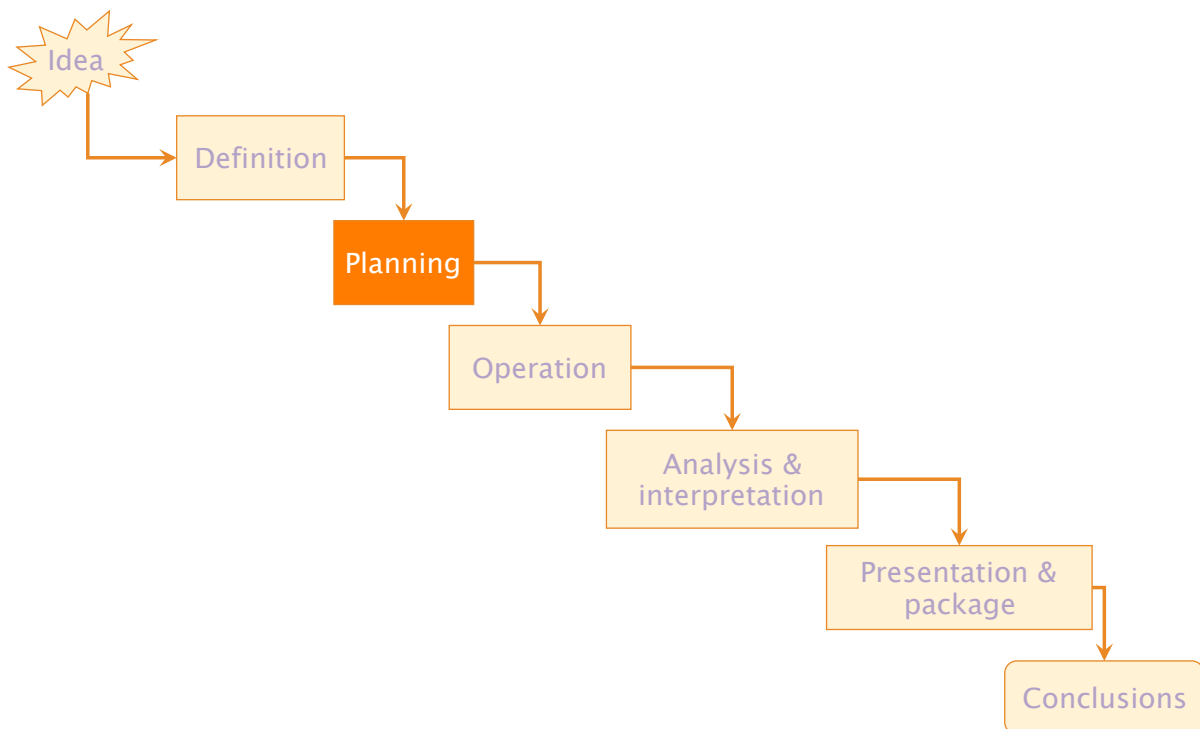
- Product definition
  - ♦ Does the sw adhere to coding standards
  - ♦ What is the complexity of sw?
- Quality
  - ♦ What is the distribution of failures?
  - ♦ What is the distribution of faults?
  - ♦ What was the distribution of failure handling effort?
  - ♦ What is the relationship between code reviews and reliability?

---

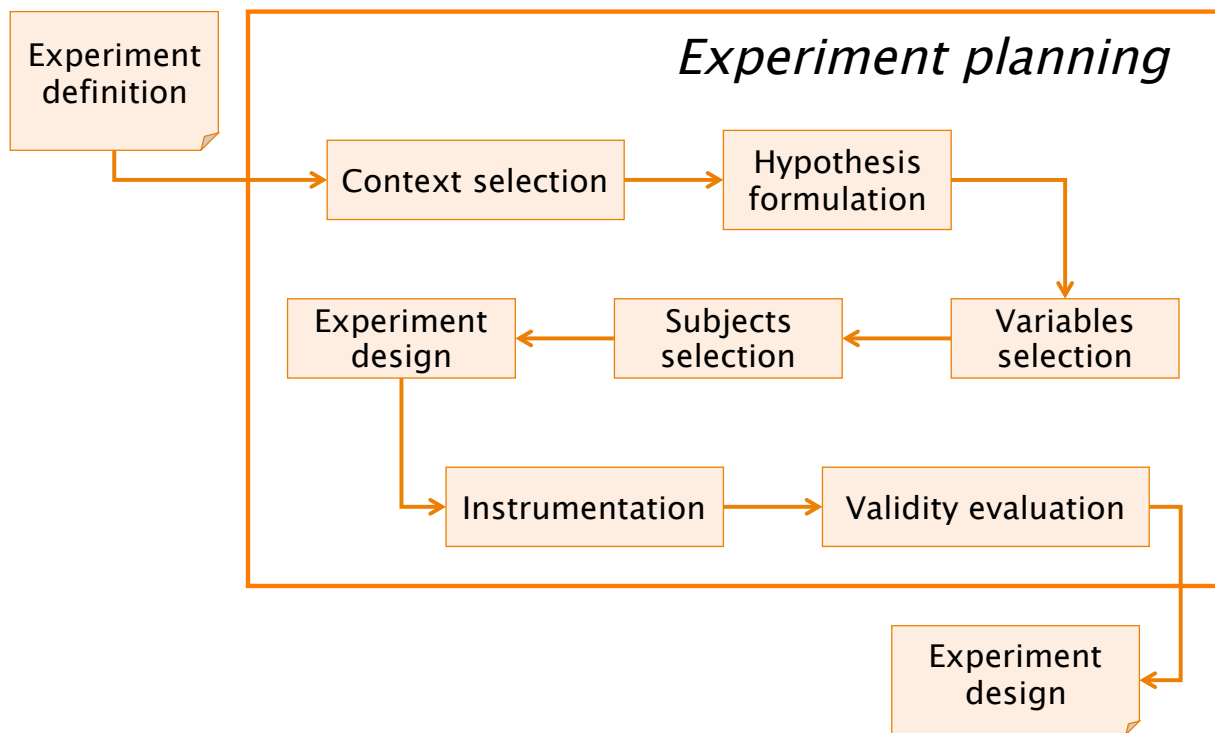
# EXPERIMENT PLANNING

## Experiment planning

---



# Experiment planning: 7 steps



## Context selection

- The content of the experiment can be characterized according to 4 dimensions:
  - ♦ Offline vs. Online
  - ♦ Student vs. Professional
  - ♦ Toy vs. Real
  - ♦ Specific vs. General

# Hypothesis formulation

---

- Two hypotheses must be formalized
  - ♦ **Null** hypothesis  $H_0$ 
    - no real underlying trends or patterns in the experiment setting
  - ♦ **Alternative** hypothesis  $H_a$ 
    - There exists real underlying trends or patterns in the experiment setting
- ♦ If we investigate the existence of a pattern, the null hypothesis must state that no patterns exist

## Scientific method (reminder)

---

- Conjecture (P)
  - ♦ Administration of treatment has influence on some feature
- Consequence (Q)
  - ♦ We observe a difference in terms of some feature

If P, then Q

# Falsification (*modus tollens*)

---

- We aim at detecting  $\sim Q$ 
  - ♦ The opposite of the consequence
  - ♦ We test the null hypothesis
  - ♦ If verified we can conclude the conjecture is false
- Aiming at verifying Q: is **wrong**
  - ♦ Provides no insight on the conjecture
  - ♦ *Affirming the consequent* fallacy

## Example

---

- Question
  - ♦ Do code reviews affect quality?
- Conjecture (P)
  - ♦ Code reviews reduce defects
- Hypothesis – alternative (Q)
  - ♦ (When code reviews are applied) we observe fewer defects than when they are not applied
- Hypothesis – null ( $\sim Q$ )
  - ♦ We observe no difference in terms of defects when code reviews are applied or not

# Example

---

- Outcome of the experiment
  - ◆ We confirm the null hypothesis ( $\sim Q$ )
    - We conclude that code reviews do not reduce defects ( $\sim P$ )
    - $\sim Q \Rightarrow \sim P$
    - The conjecture has been falsified
  - ◆ We reject the null hypothesis
    - We are more confident that it is likely that code reviews reduce defects ( $P$ )
    - ~~$Q \Rightarrow P$~~
    - The conjecture has been corroborated

# Hypotheses example

---

- $H_{c0}$ . The presence of screen mockups *does not significantly improve* the comprehension level of functional requirements.
- $H_{t0}$ . The presence of screen mockups *does not significantly improve* the time to accomplish a comprehension task.
- $H_{e0}$ . The presence of screen mockups *does not significantly affect* the efficiency of the comprehension task.

# Variable selection

---

- Independent variables
  - ◆ Variables that we can control
    - Treatment
  - ◆ Variables that we can monitor
    - Context and domain
    - Possible confounding factors
- Dependent variables
  - ◆ Allow measure the effect of treatments
    - Sometimes they cannot be measured directly
    - Use of proxies

## Example variables

---

Variable	Type	Description	Scale
Method	Indep.	whether requirements are augmented with screen mockups	Nominal $\in \{S, T\}$
Application	Indep.	experimental object used in task	Nominal $\in \{AMICO, EasyCoin\}$
Lab	Indep.	order of the experiment unit within the experiment for the participant	Ordinal $\in \{1, 2\}$
Experiment	Indep.	participants' profiles and experiment	Nominal $\in \{UniBas1, UniGe, PoliTo, UniBas2\}$
Comprehension level	Dep.	comprehension achieved by a participant on the functional requirements	Ratio $\in [0, 1]$
Task completion time	Dep.	time spent by a participant to complete a comprehension task	Interval $\in (0, \infty)$
Task efficiency	Dep.	task time efficiency	Ratio $\in [0, 600]$
Source	Dep.	(main) source of information used to perform comprehension task	Nominal $\in \{G, K, UC, UCD, S\}$

# Subjects selection

---

- Subjects are selected from a population:
  - ♦ Probability sampling
    - Simple random, systematic, stratified
  - ♦ Non probability sampling
    - Convenience, quota
- Related to the level of generalization of the experiment

# Subjects selection

---

- General principles
  - ♦ The larger the variation of the population is, the larger is the sample size needed
  - ♦ Analysis of data may influence choice of sample size: consider how to analyze data since design stage



# Subjects Example

---

	<b>UniBas1</b>	<b>UniGe</b>	<b>PoliTo</b>	<b>UniBas2</b>
<b>Date</b>	Nov. 2009	Dec. 2009	Apr. 2010	Dec. 2010
<b>Location</b>	Univ. Basilicata	Univ. Genova	Poly. Torino	Univ. Basilicata
<b>Degree</b>	Computer Science	Computer Science	Computer Engineering	Math/Telecom
<b>Level</b>	Undergrad	Undergrad	Graduate	Graduate
<b>Year</b>	2nd	3rd	2nd	1st
<b>Semester</b>	1st	1st	2nd	1st
<b>Number</b>	33	51	24	31

## Design

---

- Experiment = series of trials
  - ◆ Number of factors and treatments determines design type and data analysis
- Memento:
  - ◆ Design and interpretation of results are closely related: the choice of design affects the analysis and vice versa

# Design

---

- General design principles
  - ◆ Randomization
  - ◆ Blocking
  - ◆ Balancing

# Randomization

---

- Used to
  - ◆ average the effects of a factor that may otherwise be present
  - ◆ select representative subjects for the population they come from
- Applies on
  - ◆ allocation of the objects
  - ◆ allocation of the subjects
  - ◆ order the tests are performed

# Blocking

---

- Used to eliminate the undesired effect of a (confounding) factor we are not interested in
- Blocks are built separating by factor  
E.g.:
  - ♦ Block 1 : subjects with experience
  - ♦ Block 2 : subjects with no experience
- Blocks studied separately
- Effects between blocks not studied

# Balancing

---

- Experiment design is balanced when treatments are assigned so that each treatment has equal number of subjects
  - ♦ Viceversa also subjects should have a burden as far as possible similar
  - ♦ Desirable because it both simplifies and strengthens the statistical analysis of data, but not necessary

# Design types

---

- For each combination of number of factors and levels, different experiment design solutions
  - One factor with two treatments
  - One factor with more than two treatments
  - Two factors with two treatments
  - More than two factors each with two treatments

## Fully randomized design

---

- The levels of the primary factor are randomly assigned to the experimental units.
- Balance
  - ♦ Same number of replications for each level

# Fully randomized design

---

- Model
  - ♦  $Y_{ij} = \mu + T_i + \text{error}$
  - ♦ Where
    - i level of the main factor
    - j replication (subject) for that level
- Estimates
  - ♦  $\mu = \bar{Y}$ 
    - the average of all the data
  - ♦  $T_i = \bar{Y}_i - \bar{Y}$
- Hypothesis
  - ♦  $H_0: T_i = T_j \Leftrightarrow \bar{Y}_i = \bar{Y}_j \Leftrightarrow \mu_i = \mu_j$

## 1 Factor, 2 Treatment Levels

---

- Most typical and simple case
  - ♦ One of the treatments can be “absence of”
    - Does the introduction of a technique affects some output variable?
- Example
  - ♦ Factor: design notation
    - Treatment 1: UML
    - Treatment 2: UML w/stereotypes

## Randomized – 1 F, 2 T

---

- Each subject is randomly assigned to one of the two treatments
  - ♦ Balancing
  - ♦ Comparison of subjects in two groups
- Example of hypothesis
  - ♦  $H_0: \mu_1 = \mu_2$
  - ♦  $H_1: \mu_1 \neq \mu_2, \mu_1 < \mu_2 \text{ or } \mu_1 > \mu_2$
  - ♦ Where:
    - $\mu_t$  = mean of dependent variable for subject that received treatment t

## Randomized – 1 F, >2 T

---

- Each subject is randomly assigned to one of the treatments
  - ♦ Balancing
  - ♦ Comparison of subjects in groups
- Example of hypothesis
  - ♦  $H_0: \mu_1 = \mu_2 = \dots = \mu_n$
  - ♦  $H_1: \mu_i \neq \mu_j$  for at least one pair (i,j)

# Randomized – 1 F, >2 T

---

Subjects	Treatment 1	Treatment 2	Treatment 3
1	X		
2		X	
3			X
4			X
5	X		
6		X	

$\mu_1$                        $\mu_2$                        $\mu_3$

## Full factorial design $L^k$

---

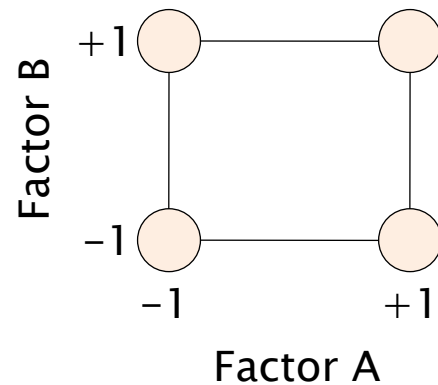
- All possible combination of factor levels are observed
  - ♦ Number of factors:  $k$
  - ♦ Number of levels per factor:  $L$
  - ♦ Replications per trial:  $n$
  - ♦ Sample size:  $n * L^k$
- Balance
  - ♦ Same number of replications for each level

# Full factorial design $2^2$

- Typically factors with  $L=2$  levels  $2^k$ 
  - ♦ Levels +1, -1

Trial	Factor A	Factor B
1	-1	-1
2	1	-1
3	-1	1
4	1	1

Design matrix



## Interaction

- The effect of a combination of two factors together
- Example: coffee
- Factors:
  - ♦ Putting sugar
  - ♦ Stirring
- Outcome
  - ♦ Sweetness

	No stir	Stir
No Sugar	0	0
Sugar	0	1



# Orthogonality

---

- The design matrix
  - ♦ has columns that are all pairwise orthogonal
  - ♦ all the columns sum to 0
- Eliminates correlation between the estimates of the main effects and interactions.
- Full factorial design are orthogonal

# Factorial design

---

- Two factors ( $k=2$ )
  - ♦  $\tau_i$  The effect of level  $i$  of factor A
  - ♦  $\beta_j$  The effect of level  $j$  of factor B
  - ♦  $(\tau\beta)_{ij}$  The effect of the interaction between  $\tau_i$  and  $\beta_j$
- Example of hypotheses:
  - ♦  $H_0: \tau_1 = \tau_2 \quad \beta_1 = \beta_2 \quad (\tau\beta)_{ij} = 0$  for all  $i, j$
  - ♦  $H_1: \tau_1 \neq \tau_2 \quad \beta_1 \neq \beta_2 \quad$  at least one  $(\tau\beta)_{ij}$

# Factorial design

		Factor A		
		Treatment A1	Treatment A2	
Factor B	Treatment B1	Subjects 4, 6 $(\tau\beta)_{11}$	Subjects 1, 7 $(\tau\beta)_{21}$	$\beta_1$
	Treatment B2	Subjects 2, 3 $(\tau\beta)_{12}$	Subjects 5, 8 $(\tau\beta)_{22}$	$\beta_2$
		$\tau_1$	$\tau_2$	

## Factorial design – nested

2 factors, two stage nested design

<i>example</i>	Factor A			
	Treatment A1		Treatment A2	
	Factor B		Factor B	
	Treatment B1'	Treatment B2'	Treatment B1''	Treatment B2''
	Subject: 1,3	Subject: 6,2	Subject: 7,8	Subject: 5,4

- Example of hypothesis
  - same as for 2\*2 factorial design

# Factorial design – 3 Factors

---

More than 2 factors,  $2^k$  factorial design

*example*

Factor A	Factor B	Factor C	Subjects
A1	B1	C1	2,3
A2	B1	C1	1,13
A1	B2	C1	5,6
A2	B2	C1	10,16
A1	B1	C2	7,15
A2	B1	C2	8,11
A1	B2	C2	4,9
A2	B2	C2	12,14

- Example of hypothesis
  - same as for  $2 \times 2$  factorial design

---

**SoftEng**  
http://softeng.polito.it

## Randomized blocked design

---

- A co-factor is used as a blocking factor if every level of the primary factor occurs the same number of times with each level of the co-factor.
  - ♦ *Block what you can, randomize what you cannot.*
- Characteristics
  - ♦  $k$  : number of factors
  - ♦  $L_i$  : levels of factor  $i$
  - ♦  $n$  : replications per cell
  - ♦ Sample size:  $n * \prod L_i$

---

**SoftEng**  
http://softeng.polito.it

# Randomized blocked design

---

- Model
  - ♦  $Y_{ijl} = \mu + T_i + B_j + \text{error}$
  - ♦ Where
    - i level of the main factor
    - j level of the blocking factor
    - l replication (subject) for that combination
- Estimates
  - ♦  $\mu = \bar{Y}$ 
    - the average of all the data
  - ♦  $T_i = \bar{Y}_i - \bar{Y}$
  - ♦  $B_j = \bar{Y}_j - \bar{Y}$

## Blocked factorial design $2^k$

---

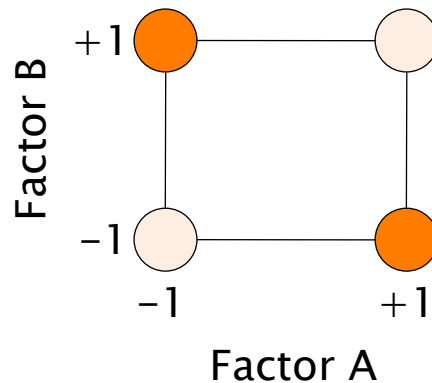
- Each co-factor level occurs the same number of times for each level of any main factor.

Trial	Factor A	Factor B	A*B	Block
1	-1	-1	1	B1
2	1	-1	-1	B2
3	-1	1	-1	B2
4	1	1	1	B1

# Blocked factorial design

---

- Each co-factor level occurs the same number of times for each level of any main factor.



# Latin square designs

---

- One main factor and two co-factors
  - ♦ Allow experiments with a relatively small number of runs
  - ♦ Number of levels of each blocking variable must equal the number of levels of the treatment factor
  - ♦ No interactions between the blocking variables or between the treatment variable and the blocking variable.

## 3 x 3 Latin square

---

- One Main factor
  - ♦ Diagram: Informal, UML, UML w/stereotypes
- Two Co-factors:
  - ♦ Experience: Low, Medium, High
  - ♦ Model size: Small, Medium, Large
- Number of trials:
  - ♦ Fully blocked: 27 ( $= L_{\text{diagram}} * L_{\text{Experience}} * L_{\text{Size}}$ )
  - ♦ Latin square: 9

## 3x3 Latin square

---

		Experience		
		Low	Medium	High
Model Size	Small	Informal	UML	UML w/ stereotypes
	Medium	UML w/ stereotypes	Informal	UML
	Large	UML	UML w/ stereotypes	Informal

# Fractional design $L^{k-p}$

---

- Only an adequately chosen fraction of the treatment combinations required for the complete factorial experiment is selected to be run.
- Number of trials
  - ♦ Full factorial:  $2^3 = 8$
  - ♦ Fractional:  $2^{3-1} = 2^2 = 4$

## Fractional $2^{3-1}$

---

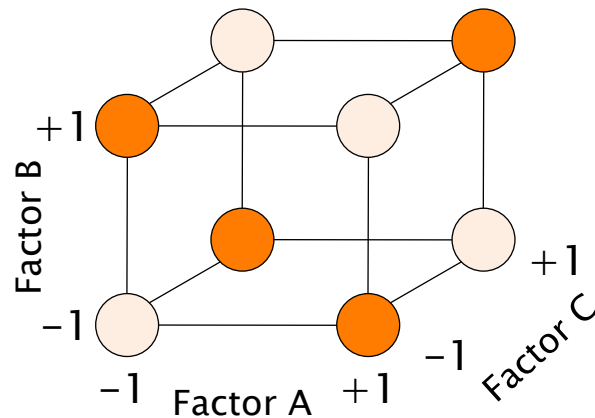
$2^2$  factorial

Trial	A	B	A*B
1	-1	-1	1
2	1	-1	-1
3	-1	1	-1
4	1	1	1

Factor C

# Fractional $2^{3-1}$

- Design generator (generating relation)
  - ♦  $C = A*B \rightarrow$  dark corners
  - ♦  $C = -A*B \rightarrow$  light corners



## Fractional design

$2^{3-1}$  fractional

*Example  
one-half  
fraction of  
the  $2^{3-1}$   
factorial  
design*

Factor A	Factor B	Factor C	Subjects
A1	B1	C2	2,3
A2	B1	C1	1,8
A1	B2	C1	5,6
A2	B2	C2	4,7

- Example of hypothesis
  - same as for  $2*2$  factorial design



# Randomized – 1 F, 2 T

---

	Subjects	Treatment 1	Treatment 2
<i>example</i>	1	X	
	2		X
	3		X
	4	X	
	5		X
	6	X	
		↓ $\mu_1$	↓ $\mu_2$

# Paired designs – 1 F, 2 T

---

- Each subject is assigned to both treatments in two distinct trials
  - ♦ Order must be randomized
  - ♦ Check for individual difference
- Example of hypothesis
  - ♦  $H_0: \mu_d = 0$
  - ♦  $H_1: \mu_d \neq 0, \mu_d < 0$  or  $\mu_d > 0$
  - ♦ Where
    - Defined  $y_{ij}$  as the measure of output variable for subject  $j$  when assigned to treatment  $i$
    - $\mu_d$  is the mean of the individual differences  
 $d_j = y_{1j} - y_{2j}$

# Paired – 1 F, 2 T

---

	Subjects	Treatment 1	Treatment 2
<i>example</i>	1	trial 1	trial 2
	2	trial 2	trial 1
	3	trial 1	trial 2
	4	trial 1	trial 2
	5	trial 2	trial 1
	6	trial 2	trial 1

↓  
 $\mu_d$

# Complete blocked– 1 F, >2 T

---

- Each subject is assigned to **each** treatment
  - ♦ The order is randomized (and balanced)
  - ♦ Comparison of subjects in groups
- Example of hypothesis
  - ♦  $H_0: \mu_1 = \mu_2 = \dots = \mu_n$
  - ♦  $H_1: \mu_i \neq \mu_j$  for at least one pair (i,j)

# Complete blocked – 1 F, >2 T

Subjects	Treatment 1	Treatment 2	Treatment 3
1	Trial 1	Trial 3	Trial 2
2	Trial 3	Trial 1	Trial 2
3	Trial 2	Trial 3	Trial 1
4	Trial 2	Trial 1	Trial 3
5	Trial 3	Trial 2	Trial 1
6	Trial 1	Trial 2	Trial 3

↓
↓
↓  
 $\mu_1$ 
 $\mu_2$ 
 $\mu_3$

## Example design

Table IV. Experiment Design. S is for requirements specification with screen mockups and T without them

	Group 1	Group 2	Group 3	Group 4
First laboratory run	S, EasyCoin	T, EasyCoin	T, AMICO	S, AMICO
Second laboratory run	T, AMICO	S, AMICO	S, EasyCoin	T, EasyCoin

# Instrumentation

---

- Goal of instrumentation :
  - ♦ provide means for performing the experiment and monitor it, without affecting the control of the experiment
- Instruments are of three types
  - ♦ Objects
  - ♦ Guidelines
  - ♦ Measurement tools

## Example instrumentation

---

### 3.4. Procedure

The participants accomplished comprehension tasks using computers equipped with MS Word. An Internet connection was available, while performing the comprehension tasks. We provided the participants with the following material.

- The requirements specification documents in electronic format (MS Word) of EasyCoin and AMICO. In particular, each document contained
  - (i) the system mission, namely, a textual description of both the functionality of the future system and the environment in which it will be deployed;
  - (ii) a UML use case diagram summarizing the use cases of the systems;
  - (iii) functional requirements expressed as use cases specified according to the chosen template. Depending on the experiment design, use cases were or were not complemented with screen mockups;
  - (iv) a glossary of the terms.
- A paper copy of the comprehension questionnaires of EasyCoin and AMICO.
- A paper copy of the postexperiment questionnaire shown in Table VI.
- The training material, which included a set of instructional slides describing the template employed for the specification of the use cases, some examples not related with the experiment objects, and a set of slides describing the procedure to follow in the task execution.

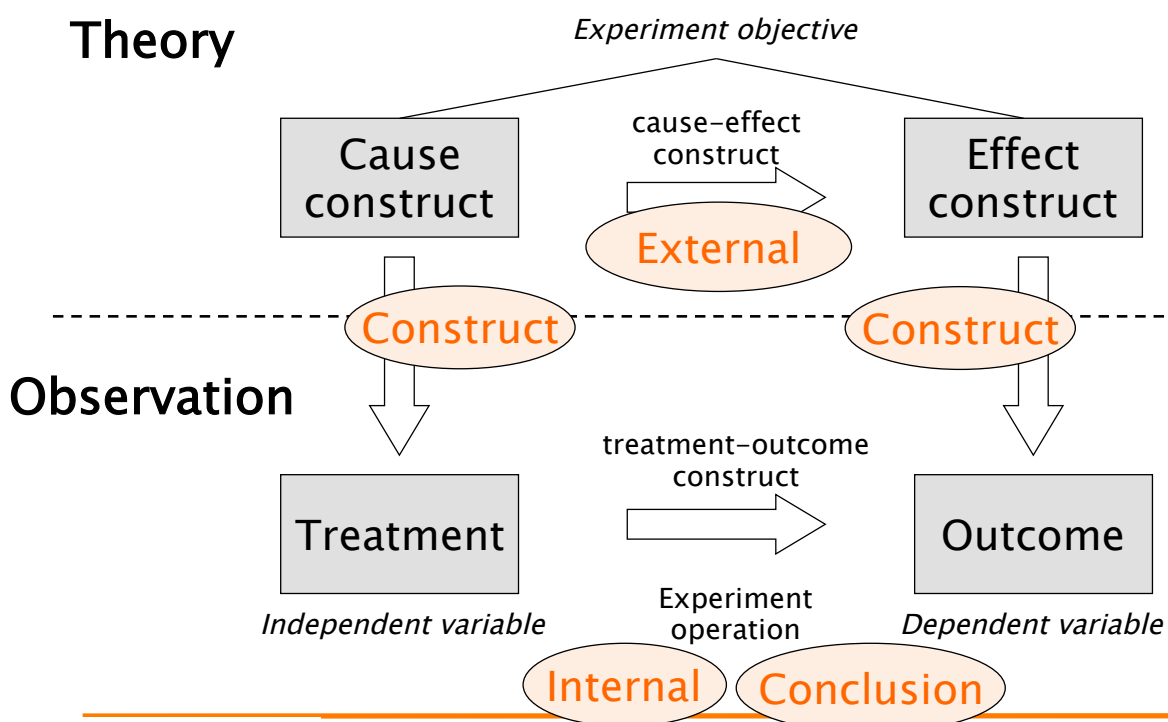
# Validity evaluation

---

- Adequate validity is obtained when results are valid for the population to which we would like to generalize
- **Threats to validity** are limitations to the adequate validity
- There are 4 types of threats:
  - ♦ Conclusion
  - ♦ Internal
  - ♦ Construct
  - ♦ External

# Validity evaluation

---



# Conclusion validity

---

- Conclusion validity
  - ◆ Threats concerning the statistical issues that can affect the ability to draw the correct conclusion about the relationship between treatments and outcome

# Internal validity

---

- Internal validity
  - ◆ Threats concerning issues that lead to indicate a causal relationship, when there is none
  - ◆ The extent to which the behavior observed in the experiment could be due to disturbing factors instead of the treatments

# Construct validity

---

- Construct validity
  - ♦ Threats concerning issues related to the relationship between
    - cause construct and treatment
    - effect construct and outcome
  - ♦ They refer to the extent to which the experiment settings actually reflect the construct under study

# External validity

---

- External validity
  - ♦ Can the result of the study be generalized outside the scope of the study ?

# Validity evaluation

---

- For each threat type, a list of threats is available in [Cook79] and [Campbell63]
- Priority among the threats is a matter of optimization
  
- Possible rank in theory testing:
  - ♦ Internal – construct – conclusion – external
- Possible rank in applied research:
  - ♦ Internal – external – construct – conclusion

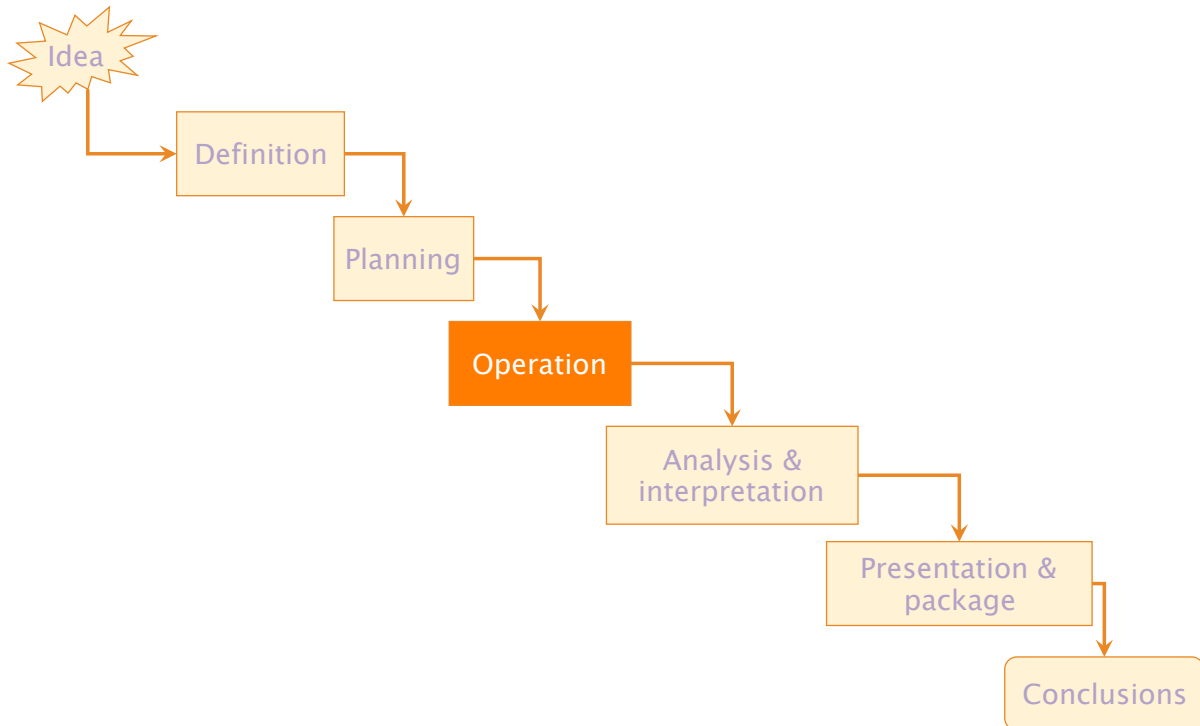
---

## OPERATION



# Operation

---



# Operation

---

- Preparation
  - ◆ Get participants
    - Ethical issues
    - Privacy issues
- Execution
  - ◆ Data collection
- Data validation

# Ethical issues

---

- Professionals
  - ♦ Paid to perform a task
- Students
  - ♦ Recruitment
  - ♦ Experiment as part of a course
    - Is it integrated?
    - Do all participants go through equivalent experience (counterbalanced design)
    - Credits
- Ethical board

# Privacy

---

- In Italy there is quite a strict law:
  - ♦ <http://www.garanteprivacy.it/garante/document?ID=1219452>
  - ♦ Section 7 provides a list of the rights of the subject,
  - ♦ Section 13 details the information to be provided to the subjects

# Privacy

---

- Information to be provided
  - ♦ Purposes and modalities of the processing for which the data are intended
  - ♦ Nature of providing the requested data
  - ♦ Consequences of denial to reply
  - ♦ Entities or categories of entity of data communication and dissemination
  - ♦ Rights
  - ♦ Responsible for the data

## Privacy information

---

- Purposes and modalities of the processing for which the data are intended
  - ♦ The data you provide will be handled for statistical and scientific purposes, aimed at investigating the details of software development. The handling will be carried on by electronic means.
- Nature of providing the requested data
  - ♦ The participation in the investigation is voluntary.
- Consequences of denial to reply
  - ♦ Denying to answer will have no consequence.
- Entities or categories of entity of data communication and dissemination
  - ♦ Personal data collected during the investigation will be shared only among the researchers involved in the project.

# Privacy information

---

- Rights
  - ♦ At any time you will be able to exert your rights with the responsible for the data handling, according to section 7 of D.lgs. 196/2003, which we copy integrally:
  - ♦ ...
- Responsible for the data
  - ♦ The responsible for data treatment is ...

# Execution

---

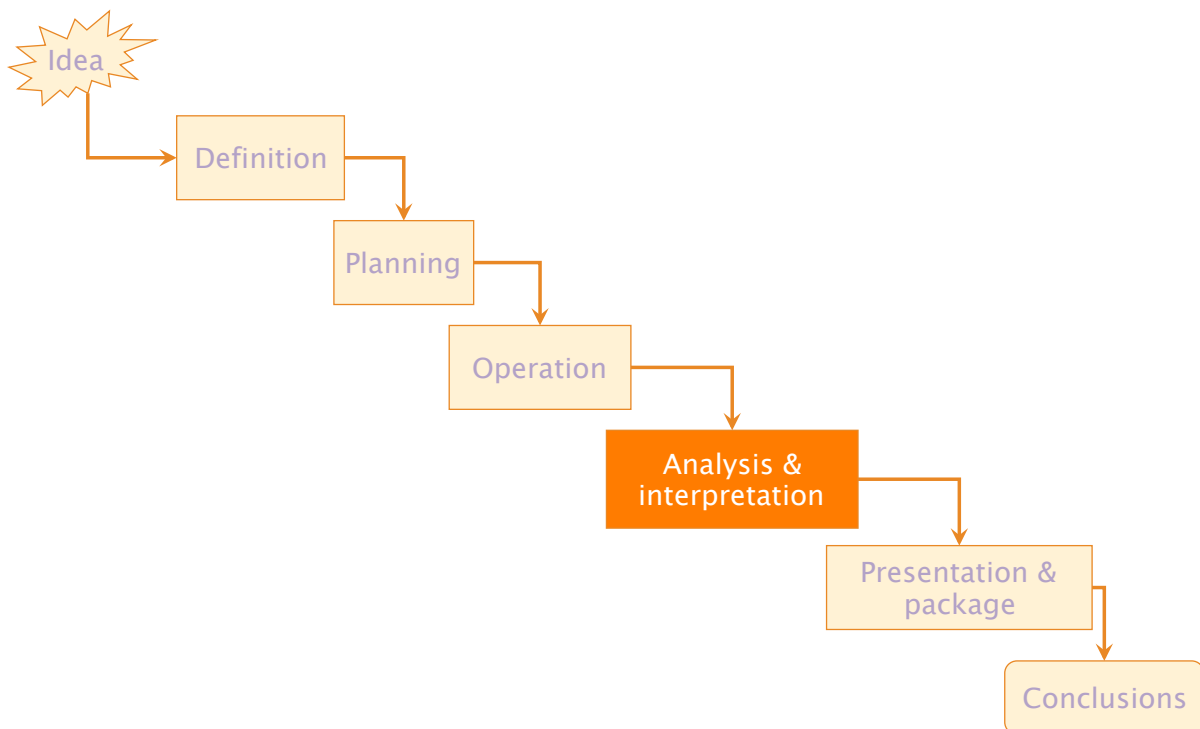
- Data collection
  - ♦ Manually entered by participants
  - ♦ Tool supported
  - ♦ Interviews
  - ♦ Automatic
- Experimental environment

---

# ANALYSIS AND INTERPRETATION

## Analysis

---



# Analysis and interpretation

---

- Descriptive statistics
  - ♦ Distribution
  - ♦ Central tendency
  - ♦ Dispersion
  - ♦ Visualization
- Data reduction
- Hypothesis testing

## Error types

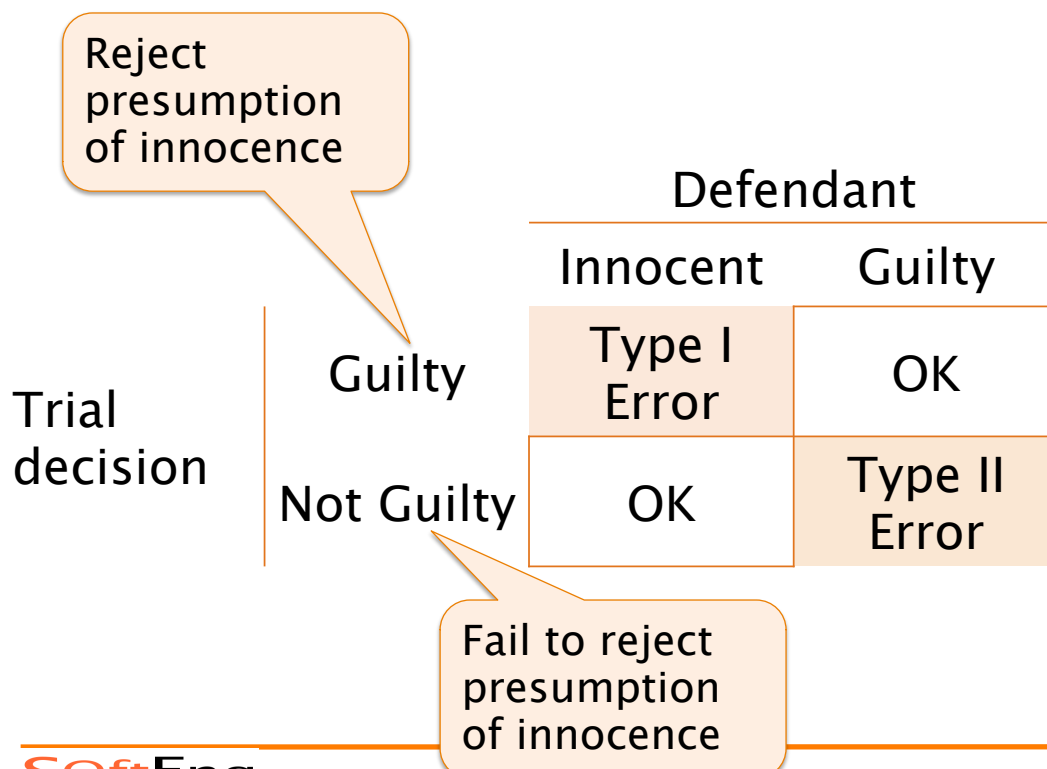
---

- Type I error
  - ♦ When we conclude there is a trend/pattern but actually there isn't
  - ♦  $\alpha = P(\text{reject } H_0 \mid H_0 \text{ is true})$
- Type II error
  - ♦ When we don't see any relation between factors and outcome, but actually there is a trend/pattern
  - ♦  $\beta = P(\text{accept } H_0 \mid H_0 \text{ is false})$

# Error types – Hypothesis testing

		$H_0$	
		True	False
Test decision	Reject	Type I Error	OK
	Fail to reject	OK	Type II Error

# Error types – Justice system



# Power

---

- The power of a test is the probability that the test reveal a true pattern if  $H_0$  is false
- Power =  $P(\text{reject } H_0 \mid H_0 \text{ is false})$   
=  $1 - P(\text{accept } H_0 \mid H_0 \text{ is false})$   
=  $1 - P(\text{type II error}) = 1 - \beta$

# Hypothesis testing

---

- Steps
  - ♦ Fix the significance level ( $\alpha$ )
    - Typically in planning phase
  - ♦ Select the statistical tests
    - Typically in planning phase
  - ♦ Perform the tests
  - ♦ Decide about null hypotheses
    - Reject
    - Fail to reject



# Significance level $\alpha$

---

- What is the acceptable  $\alpha$  level in our study?
  - ♦ Level of confidence:  $1 - \alpha$
- Standard levels:

Significance ( $\alpha$ )	Confidence ( $1 - \alpha$ )
5%	95%
1%	99%

# Test

---

- P-value
  - ♦ probability of obtaining a test statistic at least as extreme as the one that was actually observed, assuming that the null hypothesis is true
- Decision
  - ♦ Reject when  $p\text{-value} < \alpha$
  - ♦ Fail to reject when  $p\text{-value} > \alpha$

# Example – Hypothesis

---

- Conjecture:
  - ♦ coin is “tricky” and disfavors heads
- Consequence:
  - ♦ as a result of a series of tosses the number of heads is smaller than the number of tails.
- Hypotheses
  - ♦  $H_0$ : Heads = Tails = # Tosses / 2
  - ♦  $H_a$ : Heads < Tails
- We assume  $\alpha = 10\%$

# Example – Experiment

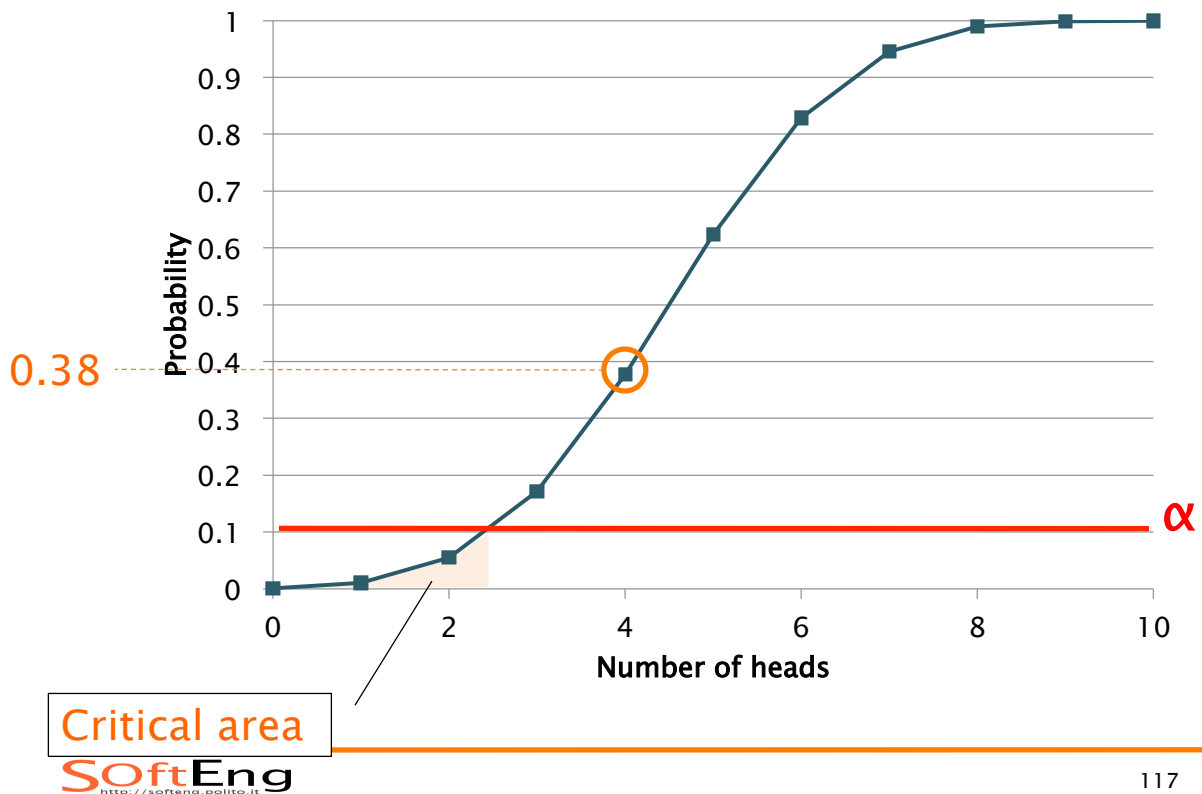
---

- Experiment result: 4 heads in 10 trials



- Assuming  $H_0$  is true, what is the probability of having 4 or less heads in 10 trials?
  - ♦ Binomial distribution
    - Cumulative function

# Example – Testing



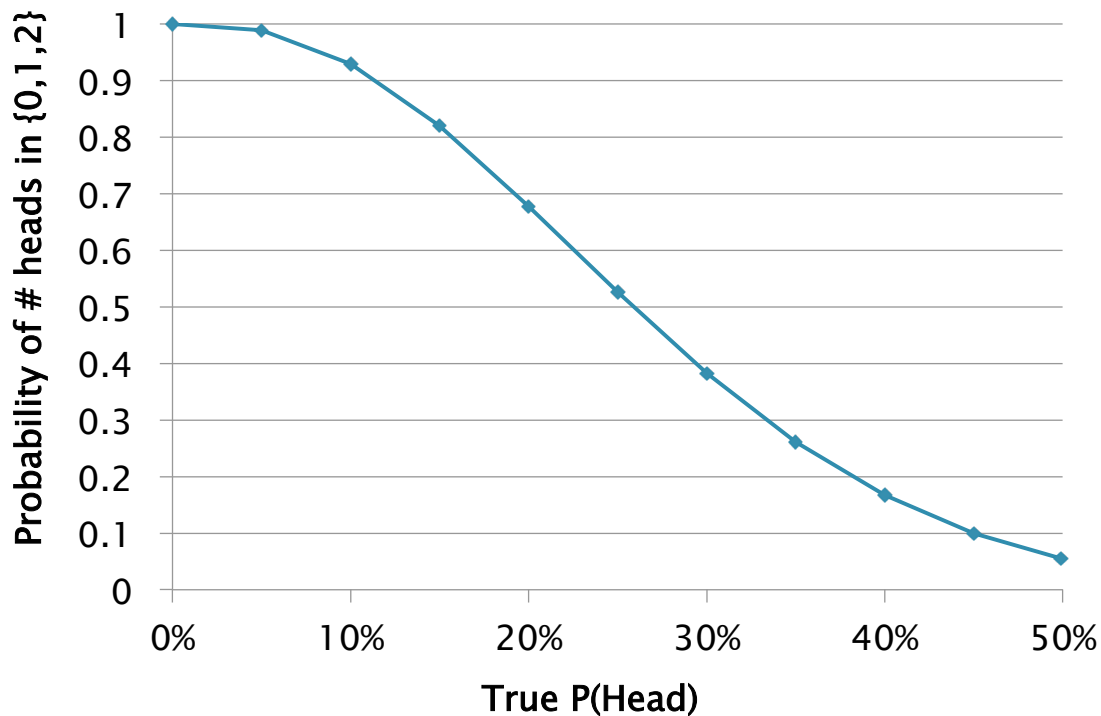
117

# Example – Power

- What is the real capability of the previous experiment to discover a tricky coin?
  - ♦ Power
- Assuming  $H_0$  is false ( $H_a$  is true) what is the probability of rejecting  $H_0$ ?
  - ♦ We reject  $H_0$  if we are in the critical area
    - In the example above: # Heads in  $\{0, 1, 2\}$
  - ♦  $H_a$  is true if  $P(\text{Head}) < 0.5$

## Example – Power

---

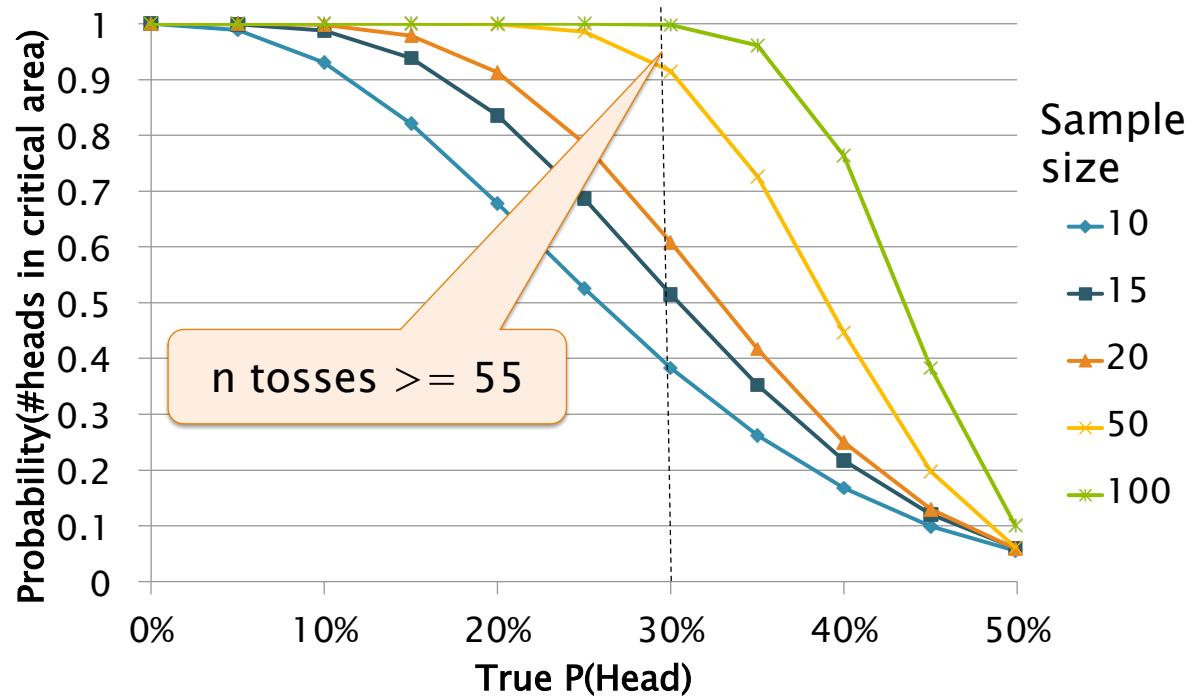


## Example – Power

---

- Let's suppose we suspect that heads show up just 30% of the times
  - ♦  $P(\text{Head}) = 30\%$
- How many trials should we run to have at least 95% of chances to discover such a bias?
  - ♦  $\text{Power} > 0.95$

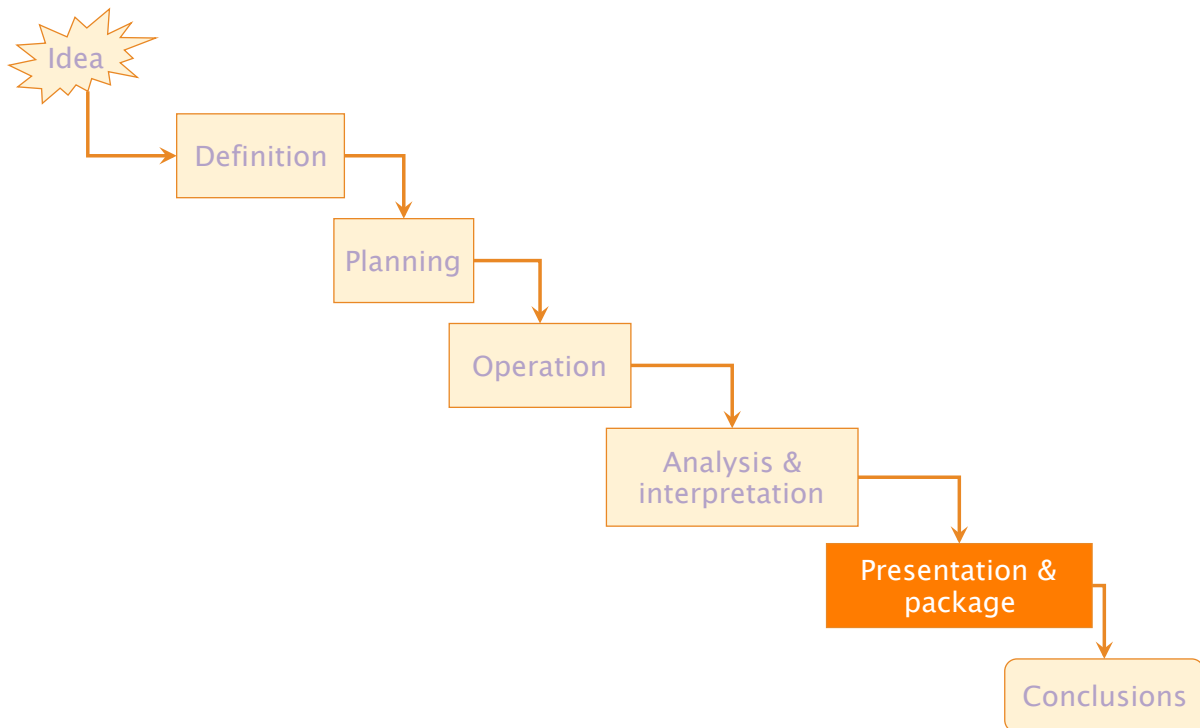
# Example – Power vs. Sample



## PRESENTATION AND PACKAGING

# Reporting

---



# Reporting

---

- Introduction
- Problem statement
- Experiment planning
- Operation
- Data analysis
- Interpretation
- Discussion and conclusions

# APA Guidelines

---

- Abstract
- Introduction
- Method
  - ♦ Design
  - ♦ Subjects/Participants
  - ♦ Apparatus/Materials
  - ♦ Procedure
- Results
- Discussion

# Replication package

---

- “Laboratory packages” (Basili et al., 1999; Shull et al.; 2002; Ciolkowski et al., 2002)
- Fundamental to allow replication
  - ♦ Analysis and goals of the experiment
  - ♦ Motivation for the design decisions
  - ♦ Experimental design, including validity threats and strengths
  - ♦ Context in which the experiment was carried on
  - ♦ Procedure
  - ♦ Analysis methods

# Online databases

---

- Promise



- ♦ <http://promisedata.org/?cat=11>

- Floss metrics



- ♦ <http://melquiades.flossmetrics.org/>

- Sw Lifecycle Empirical DB



- ♦ <http://www.thedacs.com/databases/sled/>

# Online repositories

---

- Zenodo



- ♦ Link to GitHub

- FigShare





# Bibliography

---

- Claes Wohlin, Per Runeson, Martin Host, Magnus Ohlsson, Bjorn Regnell, Anders Wesslen. 2012. *Experimentation in Software Engineering – An Introduction*. Kluwer.
- Basili, Victor; Gianluigi Caldiera, H. Dieter Rombach (1994). "The Goal Question Metric Approach"
  - ♦ <ftp://ftp.cs.umd.edu/pub/sel/papers/gqm.pdf>
- Van Solingen, Rini; Egon Berghout (1999). *The Goal/Question/Metric Method*. McGraw–Hill Education.

# Bibliography

---

- Regnell, Björn, Per Runeson, and Thomas Thelin. "Are the perspectives really different?–further experimentation on scenario–based reading of requirements." *Empirical Software Engineering* 5(4) (2000): 331–356.
- Ricca, et al. "How Developers' Experience and Ability Influence Web Application Comprehension Tasks Supported by UML Stereotypes: A Series of Four Experiments." *IEEE Transactions on Software Engineering*, 36(1) (2010): 96–118.
- Ricca et al., **Assessing the Effect of Screen Mockups on the Comprehension of Functional Requirements**. *ACM Transactions on Software Engineering and Methodology*, 24(1) (2014)