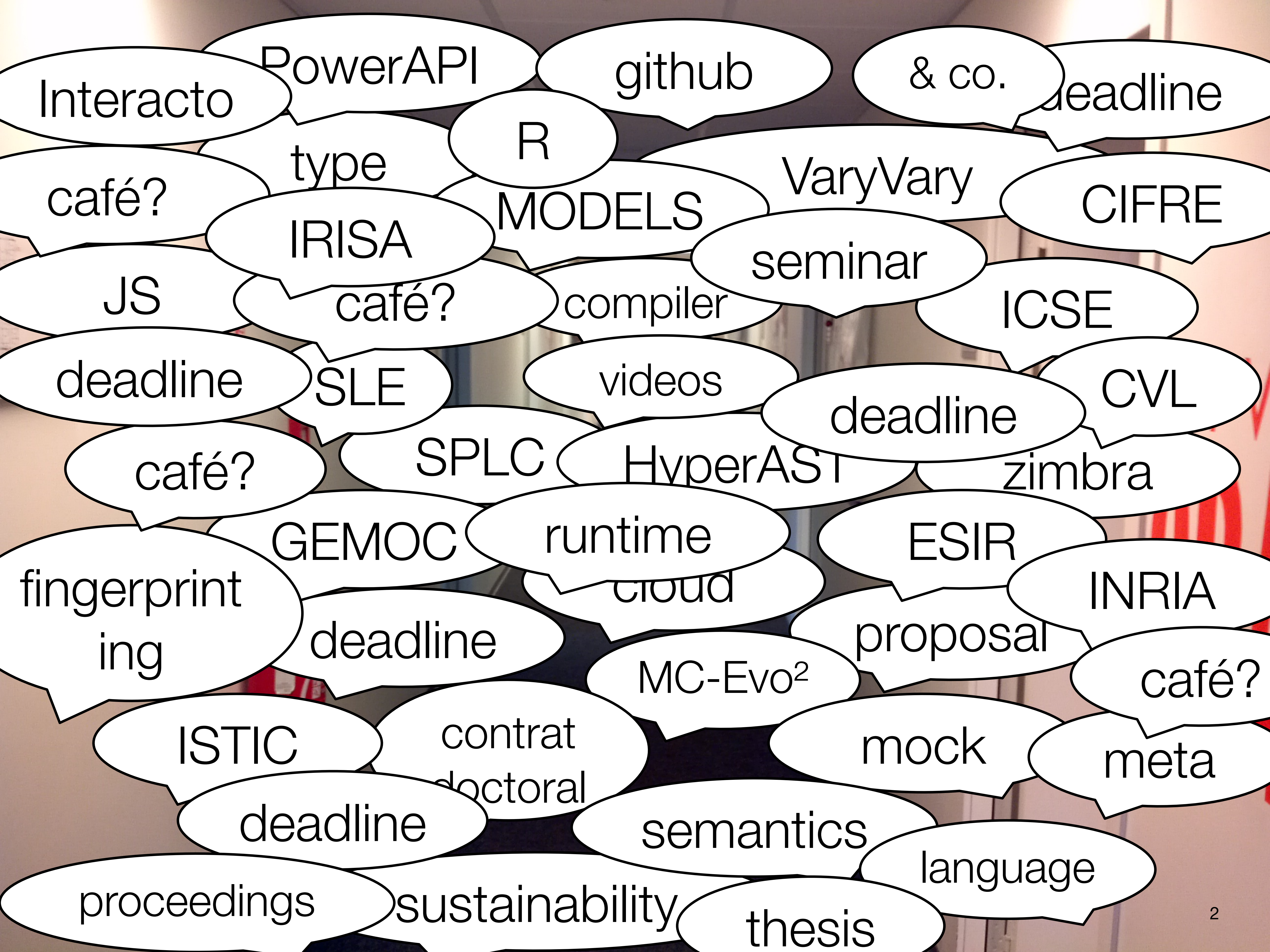
Brief introduction about the scientific method



Interacto

PowerAPI

github

& co.

deadline

café?

type

R

MODELS

VaryVary

CIFRE

IRISA

JS

café?

compiler

seminar

ICSE

deadline

SLE

videos

deadline

CVL

café?

SPLC

HyperAST

zimbra

GEMOC

runtime

cloud

ESIR

fingerprint
ing

INRIA

deadline

MC-Evo^2

proposal

café?

ISTIC

contrat
doctoral

mock

meta

deadline

semantics

language

proceedings

sustainability

thesis

HANDGUN BALLISTICS FOR BIG GAME!

Beautiful and relevant science

Science

- Long history of scientific methods
  - From ancient Greece to post modernism
- Karl Popper (1902 – 1994)
  - Rejects the possibility of induction
  - Introduces the notion of falsifiability
    - observe
    - express hypotheses that are falsifiable
    - Test to corroborate the hypothesis
    - determine a scope of validity
  - Strong emphasis on case studies and experiments

The Scientific Attitude

- **Communalism:** knowledge should be accessible for all people
- **Universalism:** everyone should have the right to contribute
- **Disinterestedness:** science should be objective and not ruled by special interests
- **Originality:** the results should be new
- **Skepticism:** scientists should be open to criticism.

Science in every-day situations

What does it mean to have a scientific attitude to things?

Some suggestions:

- You are objective. Specifically, you base your judgements on observations and verified facts
- You realize to what extent you and everyone else can be biased by your/their perspective
- You are curious and want to know facts
- You have some knowledge of scientific methodology and try to apply it

Science in research projects

Science is an activity with possibly different objectives:

- Exploratory research
- Testing-out research
- Problem-solving research

Exploratory research

- Research on a new problem about which little is known
- The problem may come from any part of the discipline; it may be a theoretical research puzzle or have an empirical basis
- The research work will need to examine what theories and concepts are appropriate, developing new ones if necessary, and whether existing methodologies can be used
- It obviously involves pushing out the frontiers of knowledge in the hope that something useful will be discovered.

Testing-out research

- Research on the limits of a previously proposed generalization
- This is often termed the ‘null hypothesis', which we are bringing evidence to ‘overthrow’ - i.e. to show is inadequate
- We can try to answer questions like: Does the theory apply on polyglot software systems? In new technology industries? At another scale?
- Make an original contribution and improve (by specifying, modifying, clarifying) the important generalizations

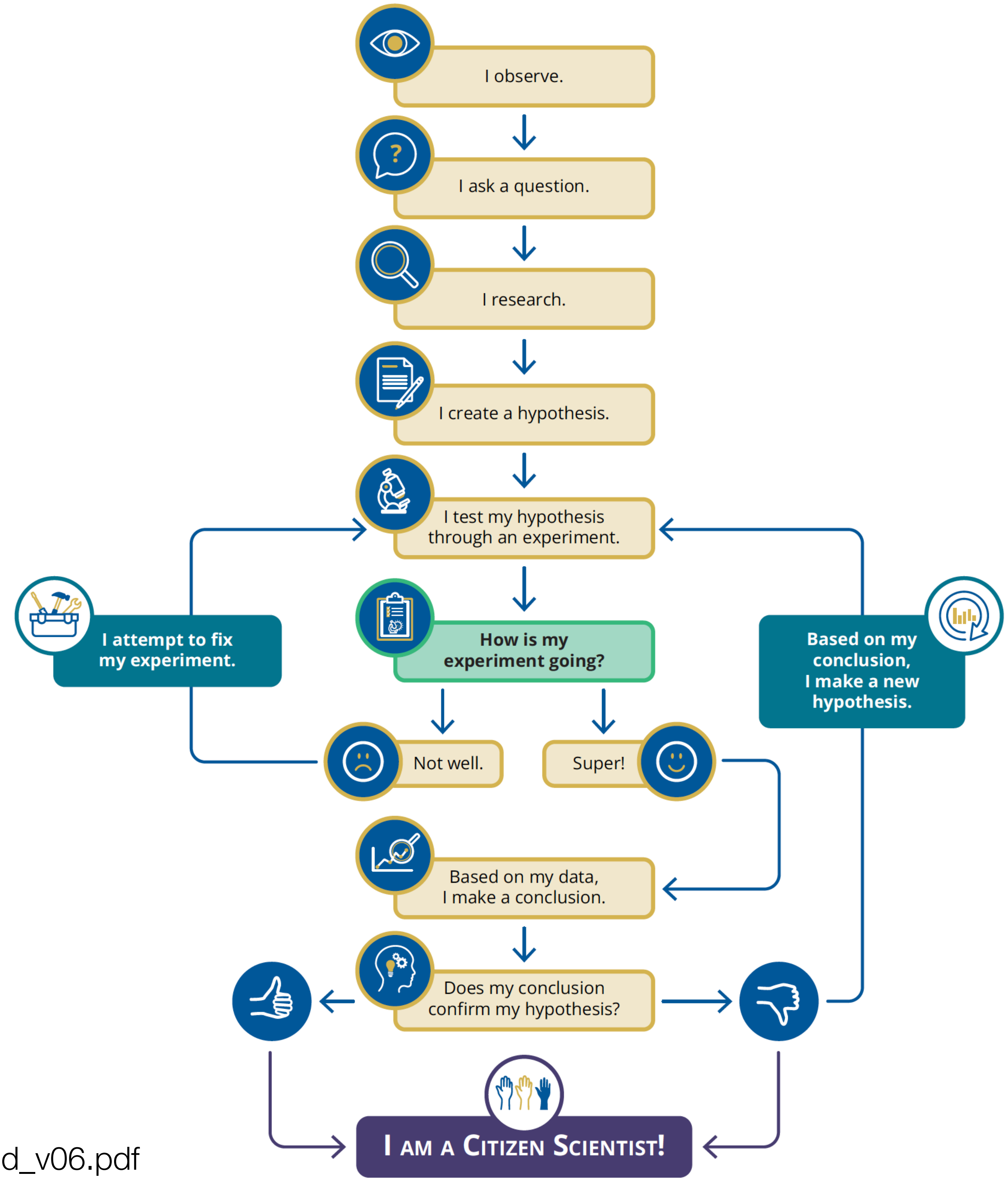
Problem-solving research

- Research starting from a particular problem in the real world, and bring together all the intellectual resources that can be brought to bear on its solution
- The problem has to be defined and the method of solution has to be discovered
- The person working in this way may have to create and identify original problem solutions every step of the way
- This will usually involve a variety of theories and methods, often ranging across more than one discipline since real-world problems are likely to be ‘messy’ and not soluble within the narrow confines of an academic discipline

Inductive vs. Deductive Reasoning

- **Inductive reasoning** is the act of making generalized conclusions based off of specific scenarios
- **Deductive reasoning** is the act of backing up a generalized statement with specific scenarios
- Most scientific work eventually combines both

Experiment studies – The big picture



I observe.

I ask a question.

I research.

I create a hypothesis.

I test my hypothesis through an experiment.

How is my experiment going?

Not well.

Super!

I attempt to fix my experiment.

Based on my data, I make a conclusion.

Does my conclusion confirm my hypothesis?

Based on my conclusion, I make a new hypothesis.

I AM A CITIZEN SCIENTIST!

https://www.nsf.gov/news/classroom/images/Scientific_Method_v06.pdf

Beautiful and relevant science in the area of software engineering

Have ideas, develop prototypes, formalize contributions, talk, read and publish papers about languages, models, product lines, architecture, systems, testing, cloud, web, security & privacy, sustainability…

Science and SE

- Inductive research: Working from specific observations in real settings to broader generalizations and theories
  - Field studies and replications, analyze commonalities
- Scalability and practicality considerations must be part of the initial research problem definition
- Researching by doing: Hands-on research. Apply what exists in well defined, realistic context, with clear objectives. The observed limitations become the research objectives.
- Multidisciplinary: other CS, Engineering, or non-technical domains

Lionel Briand, Useful Software Engineering Research: Leading a Double-Agent Life.
Keynote at ICSM'11

Science and SE

- Making a conscious effort to understand the problem first
  - Precisely identify the requirements for an applicable solution
  - More papers focused on understanding the problems
- Better relationships between academia and industry
  - Different models
  - Mutually beneficial setting where software development is an object of study
  - Exposing PhD students to industry practices

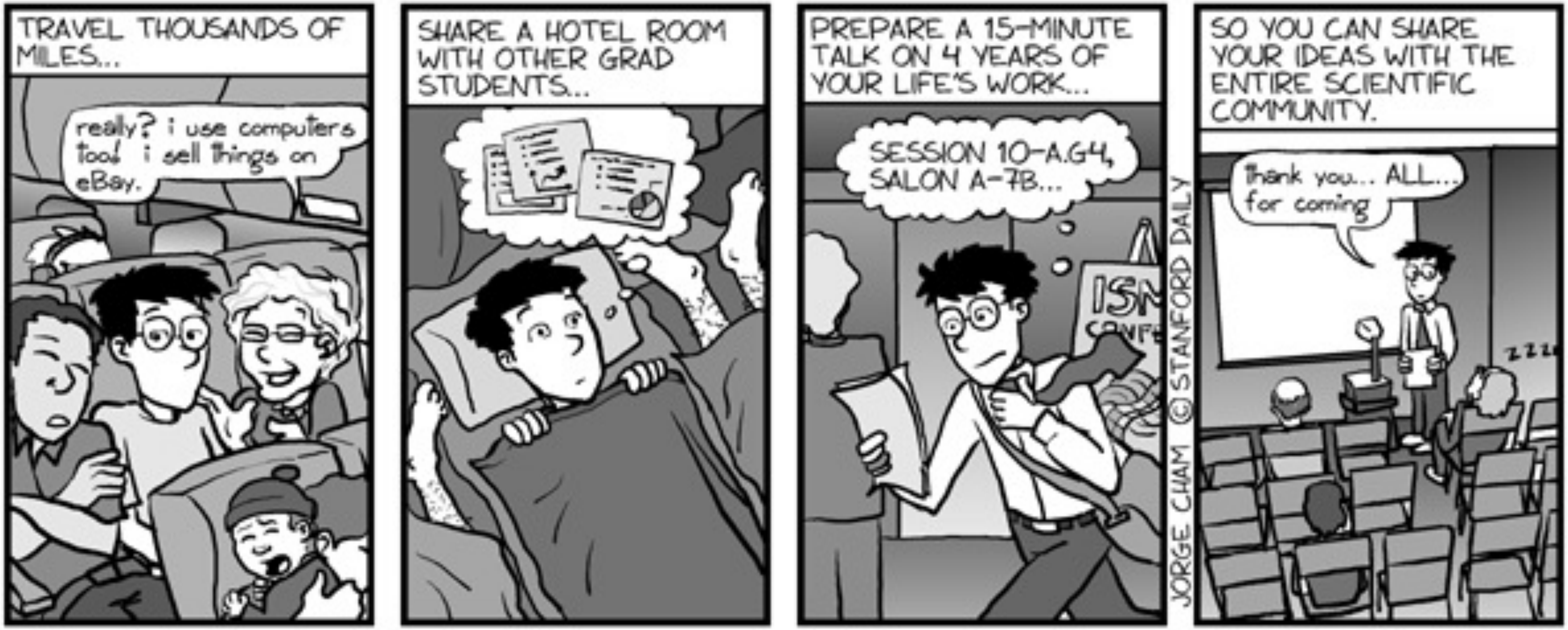
Lionel Briand, Useful Software Engineering Research: Leading a Double-Agent Life.
Keynote at ICSM’11

Experiment

- Formalize the problem and its scope
  - Probably the most difficult part of our research
- Build prototypes
  - Very important software development activity
  - Necessary to tune and validate the solution
- Experiment it on some known problems
  - Select case studies
  - Be domain specific

Write

- Write often, share often
  - iterative
  - long process
- Learn how to write
  - read good papers
  - be critical
  - learn the patterns
  - read each others' work


TRAVEL THOUSANDS OF MILES...
really? i use computers too! i sell things on eBay.
SHARE A HOTEL ROOM WITH OTHER GRAD STUDENTS...
PREPARE A 15-MINUTE TALK ON 4 YEARS OF YOUR LIFE'S WORK...
SESSION 10-A.G4, SALON A-7B...
ISM CONF
JORGE CHAM © STANFORD DAILY
SO YOU CAN SHARE YOUR IDEAS WITH THE ENTIRE SCIENTIFIC COMMUNITY.
Thank you... ALL... for coming
zzz

phd.stanford.edu

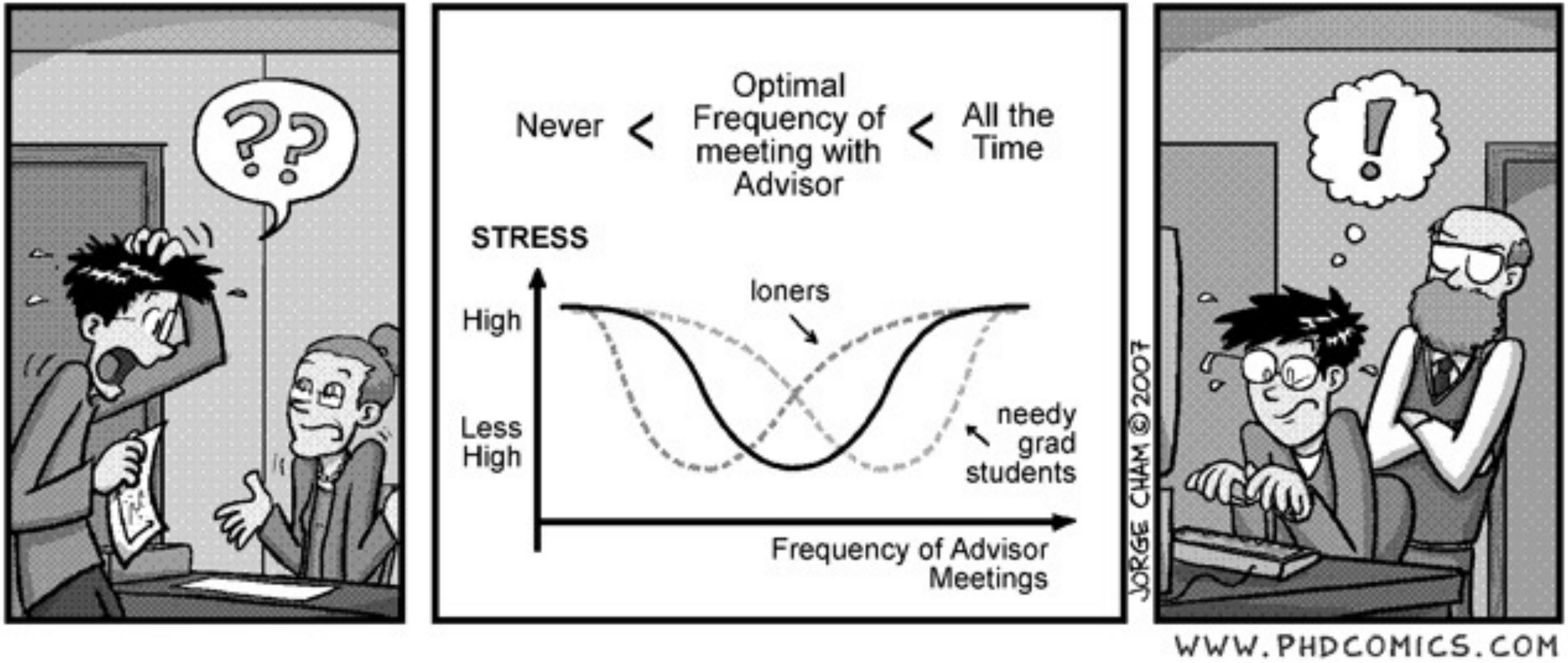
Supporting the research: Projects

OUR FUNDING AGENCY REQUIRES US TO WRITE A REPORT ON OUR ACTIVITIES.
IT'S A LOT OF WORK, BUT DON'T WORRY. I'M A STRONG BELIEVER IN DIVISION OF LABOR.
CAN YOU WRITE THE WHOLE THING BY MONDAY?
HOW IS THAT DIVISION OF LABOR?
I DIVIDE AND YOU LABOR.
JORGE CHAM © 2011
WWW.PHDCOMICS.COM

Supporting the research: Reviews

PROF. JONES
TIPPITI
TIPPI...
OH, HEY, CECILIA! I'VE BEEN MEANING TO GIVE YOU SOME CONFERENCE PAPERS TO REVIEW...
DOH!

phd.stanford.edu/


??
Never < Optimal Frequency of meeting with Advisor < All the Time
STRESS
High
Less High
loners
needy grad students
Frequency of Advisor Meetings
JORGE CHAM © 2007
!
WWW.PHDCOMICS.COM

EXPERIMENTAL SCIENCE

Why conducting experiments, studies?

To empirically show the benefits, the relevance of your proposal

- “Is my tool usable while the title of my paper is ‘A usable approach for...’”

- “Is my new algorithm better than existing ones? In which cases?”

- etc.

=> i.e., to assess the claims you wrote in your introduction

Why conducting experiments, studies?

To study, measure, understand some objects, phenomena, practices, etc.

- “Is there parts of UML that are never used?”
- “Do developers use try/catch in JUnit tests? Why? Has this practice any negative impact?”
- etc.

=> i.e., to have arguments to motivate your work

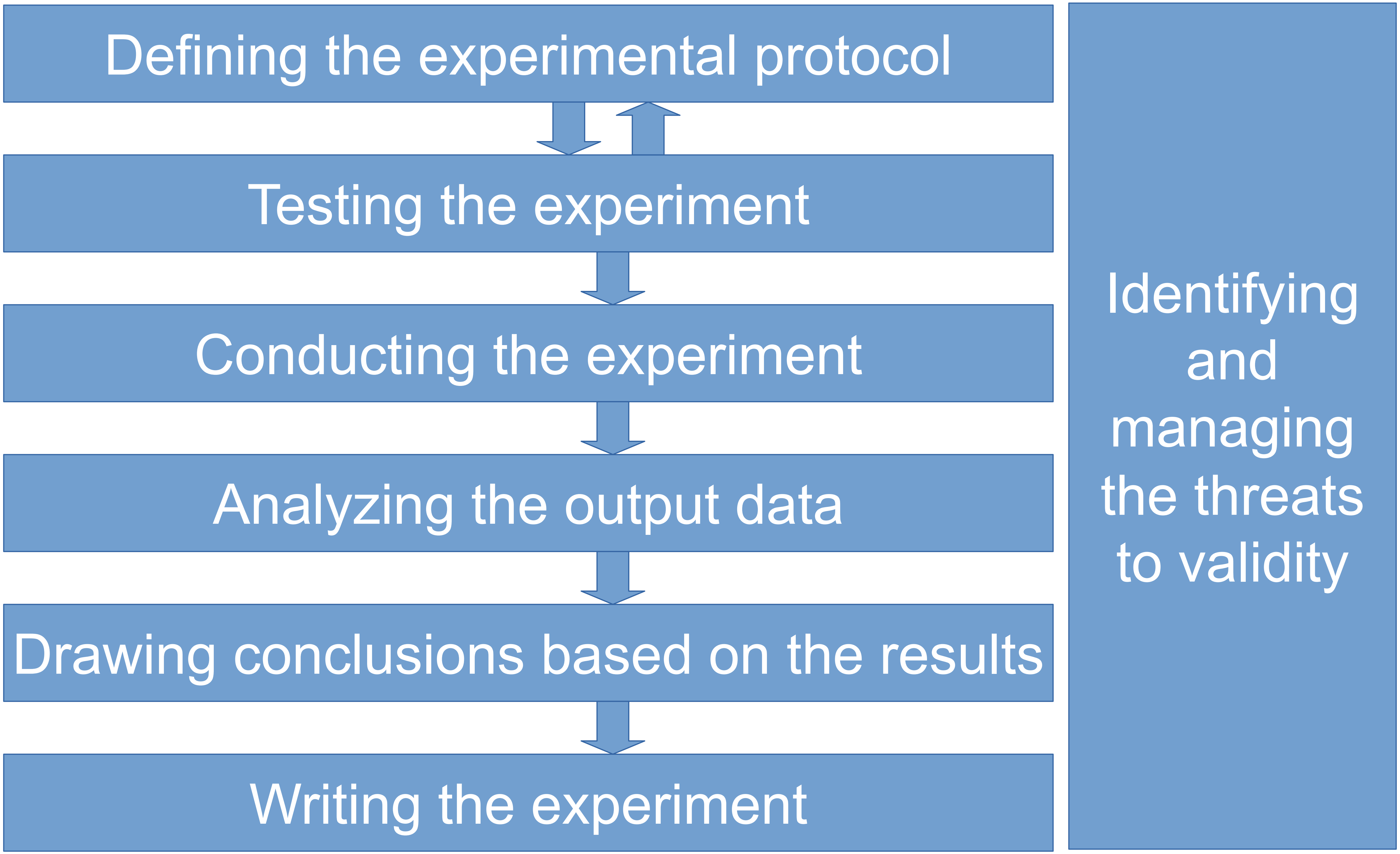
Why conducting experiments, studies?

To have your paper accepted in top-tier conferences/journals

ICSE, ASE, FSE, ISSRE, ISSTA, ICST, MODELS, SLE, SPLC, OOPSLA, PLDI, ICSME, S&P

TOSEM, TSE, JSS, SoSyM, IST, ESE, STVR

Experiment studies – The big picture



Defining the experimental protocol

Testing the experiment

Conducting the experiment

Analyzing the output data

Drawing conclusions based on the results

Writing the experiment

Identifying and managing the threats to validity

Defining the experimental protocol

What is the **goal(s)** of the study?

Example: “*The goal of our study is to investigate the relation between classes participating in antipatterns and their change- and fault-proneness [...].*”

F. Khomh, M. D. Penta, Y.-G. Guéhéneuc, G. Antoniol, *An exploratory study of the impact of antipatterns on class change- and fault-proneness*, Empirical Software Engineering, 2012

Formulate the goal(s) of your study in one or several **research questions**. Example:

RQ1: “*What is the relation between antipatterns and change-proneness?*”
RQ2: “*What is the relation between antipatterns and fault-proneness?*”

[...]

Defining the experimental protocol

What are the **objects** of the study?

The data you will analyse, aka "*context*"

The objects you will analyze

"*The context of this study consists in the change history and issue-tracking systems of four Java systems.*
[…]
The four systems have ***different sizes*** *and belong to* ***different domains*** *[...]*.

Arguments that motivate the selection
The goal is to avoid threats to validity
(e.g., data not representative, you analysed your code, only one system analysed)

Defining the experimental protocol

What **variables** are you going to measure during the experiment?
- **Dependent variables**: the output metrics you will measure to discuss your research questions.
“***Change-proneness*** *refers to whether a class underwent at least a change between release k and the subsequent release k+1.*
[…]
Fault-proneness *refer to whether a class underwent at least a fault-fixing change between releases k and k+1.*
[...]
We compute the odds ratio indicating the likelihood of an event to occur.“

Other examples: error rates, execution times

Defining the experimental protocol

What **variables** are you going to measure during the experiment?

- **Independent variables**: the input metrics you will use to compute the dependent variables.

“*Our independent variables are the number of classes participating in the 13 antipatterns.*”

Defining the experimental protocol

For experiments that involve humans

- carefully design and describe the tasks they will do

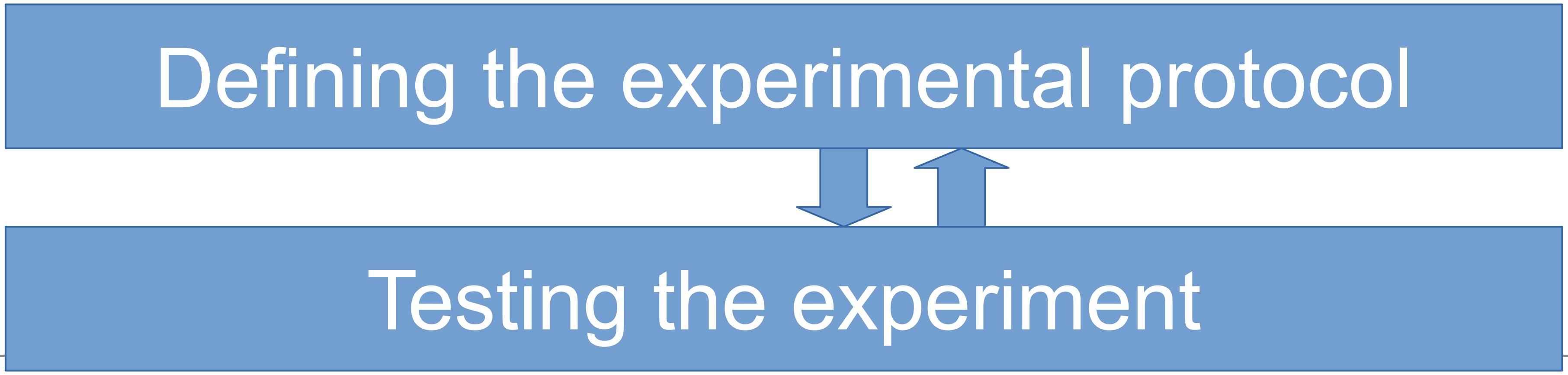
“Each subject performed four successive tasks. This number of tasks was defined to limit the duration of the experiments on one subject to around 20 minutes. […] These tasks are defined as follows: [...]”

- carefully select and describe the population

“*These subjects were composed of researchers (66 %), PhD students (14 %), industrials (11 %), research engineers (4 %), and others (5 %). They claimed to be expert (41 %), proficient (39 %), competent (13 %), advanced beginner (4 %), and novice (4 %) in MDE.*”

- One-shot experiment

If your experiment fails, not possible to re-execute it with the same human subjects without introducing a threat to validity



Defining the experimental protocol

Testing the experiment

Train your experimental protocol

- Select one or several training data sets

  Software systems, code, small set of humans

- Apply your protocol on the training data sets

  The goal is to debug your protocol

  (e.g. fix bugs in your tool, precise tasks to do by the subjects)

- Do not reuse these data sets in your study and analyses!

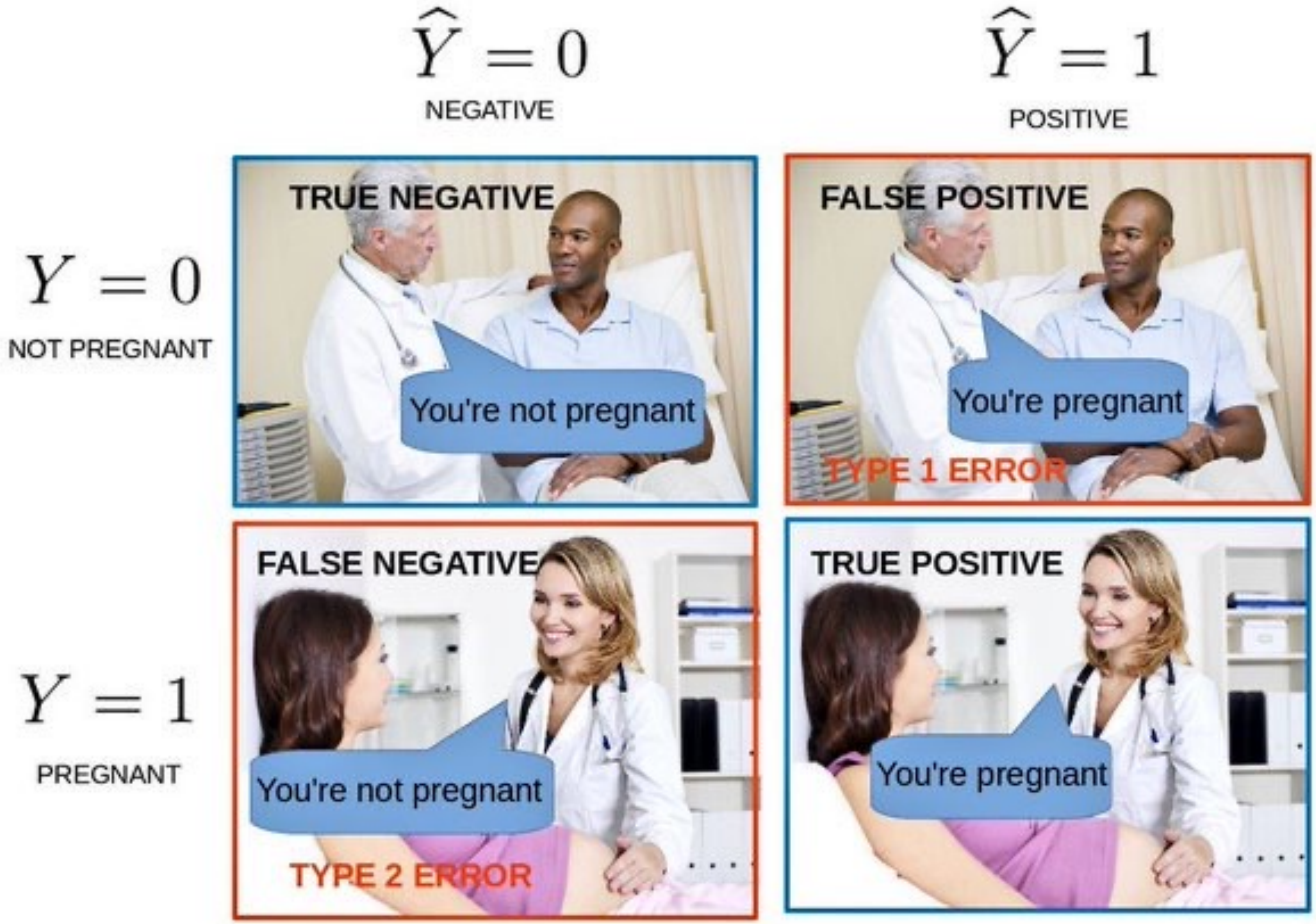
  Instead, write in the paper that you train your protocol.

Conducting the experiment

Well, just run your protocol (fingers crossed)

Analyzing the output data

True/False negative/positive



Analyzing the output data

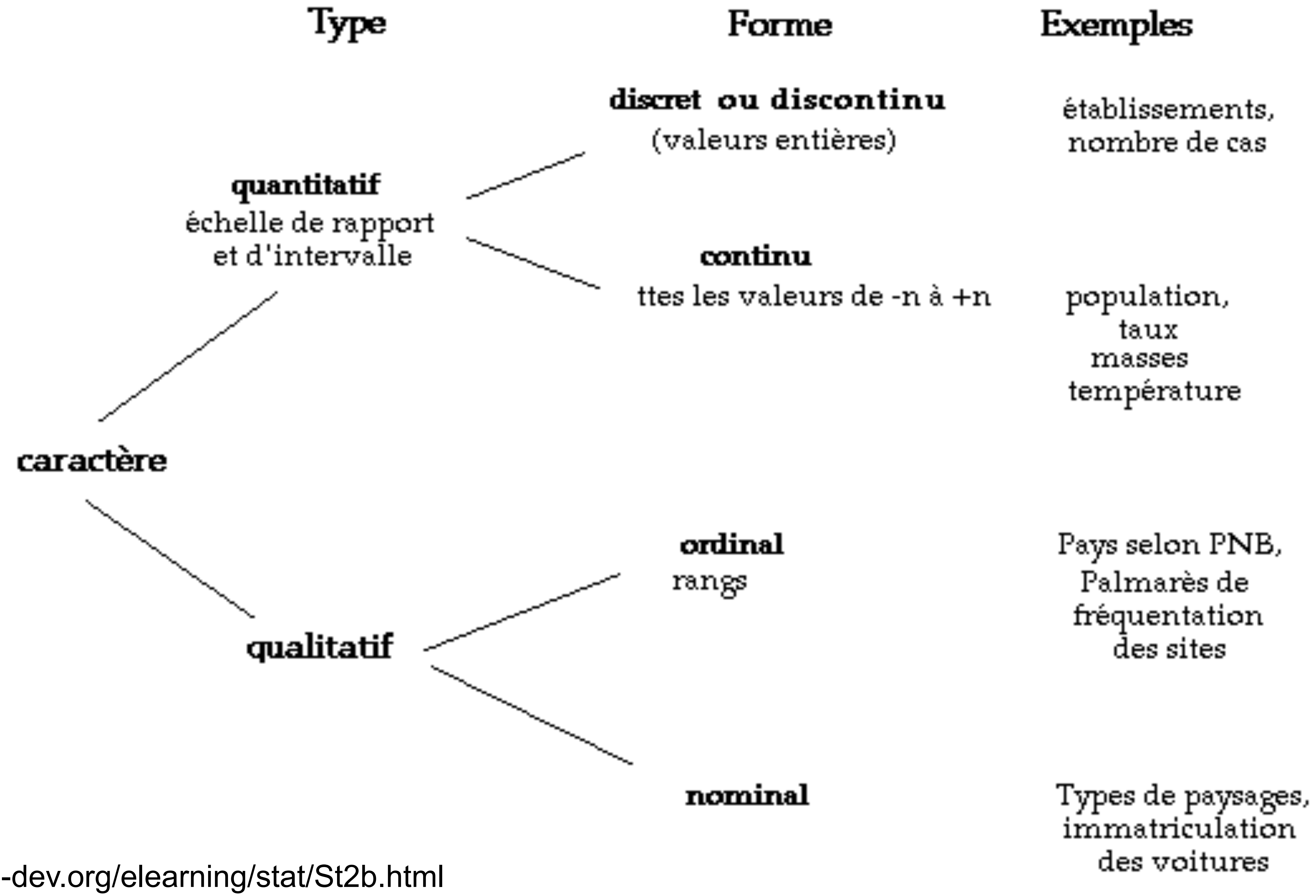
Statistics, statistics, and statistics

Normal distribution, t-test, mean, standard deviation, p-value, significance, confidence level, Spearman’s rank-order correlation, Pearson’s correlation, Mann-Whitney test, odds ratio, Fisher’s exact test, …

Have to find the statistical methods to use to analyse your results

Analyzing the output data

Identify the kinds of data to analyse...



Type

Forme

Exemples

caractère

quantitatif
échelle de rapport
et d'intervalle

discret ou discontinu
(valeurs entières)

établissements,
nombre de cas

continu
ttes les valeurs de -n à +n

population,
taux
masses
température

qualitatif

ordinal
rangs

Pays selon PNB,
Palmarès de
fréquentation
des sites

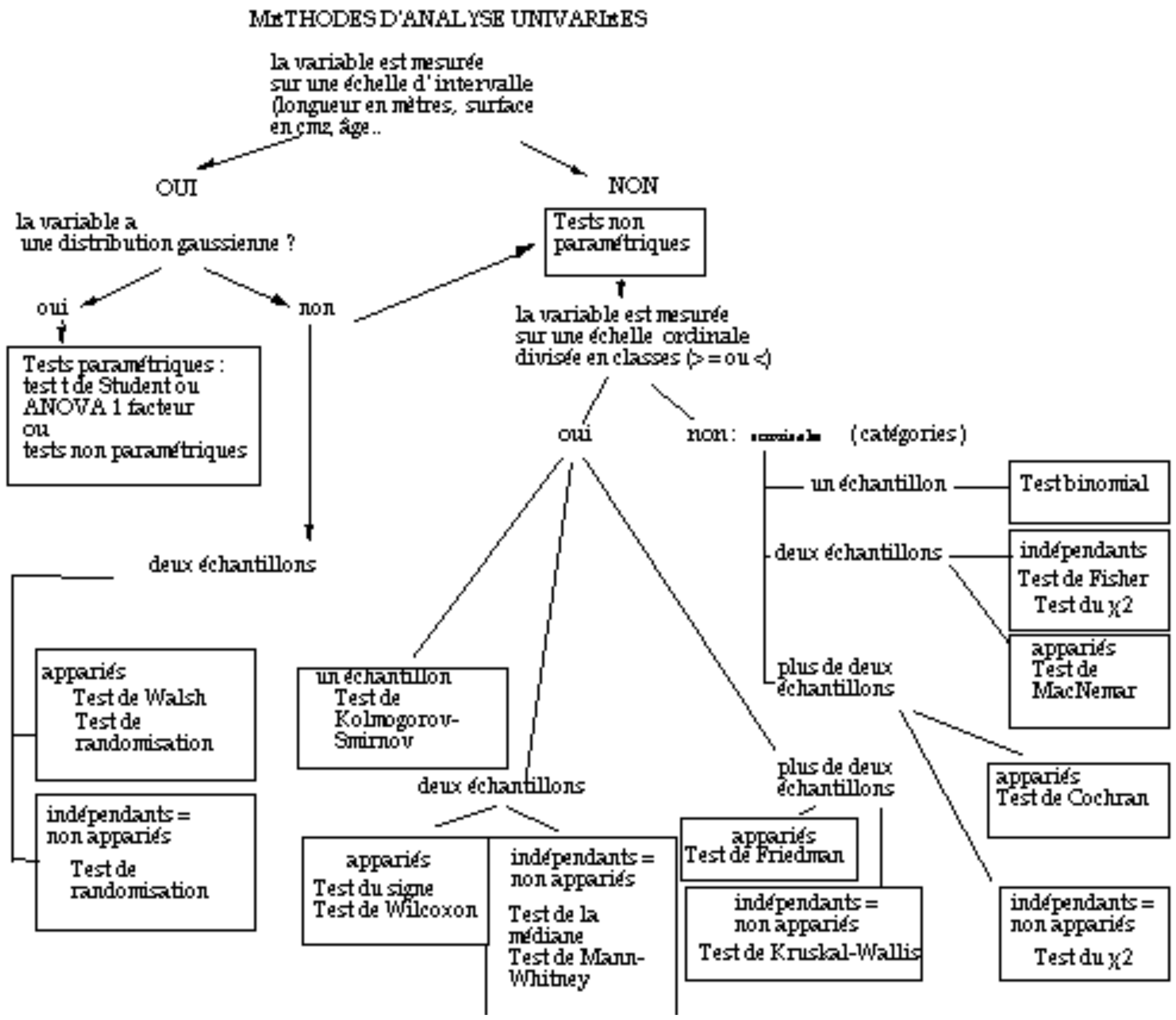
nominal

Types de paysages,
immatriculation
des voitures

http://www.cons-dev.org/elearning/stat/St2b.html

Analyzing the output data

… to then identify the statistical tests to use?



MÉTHODES D'ANALYSE UNIVARIÉES

la variable est mesurée sur une échelle d'intervalle (longueur en mètres, surface en cm2, âge..

OUI

la variable a une distribution gaussienne ?

oui

Tests paramétriques : test t de Student ou ANOVA 1 facteur ou tests non paramétriques

non

deux échantillons

appariés
Test de Walsh
Test de randomisation

indépendants = non appariés
Test de randomisation

NON

Tests non paramétriques

la variable est mesurée sur une échelle ordinale divisée en classes (> = ou <)

oui

un échantillon
Test de Kolmogorov-Smirnov

deux échantillons

appariés
Test du signe
Test de Wilcoxon

indépendants = non appariés
Test de la médiane
Test de Mann-Whitney

plus de deux échantillons

appariés
Test de Friedman

indépendants = non appariés
Test de Kruskal-Wallis

non : [illegible] (catégories)

un échantillon
Test binomial

deux échantillons

indépendants
Test de Fisher
Test du χ2

appariés
Test de MacNemar

plus de deux échantillons

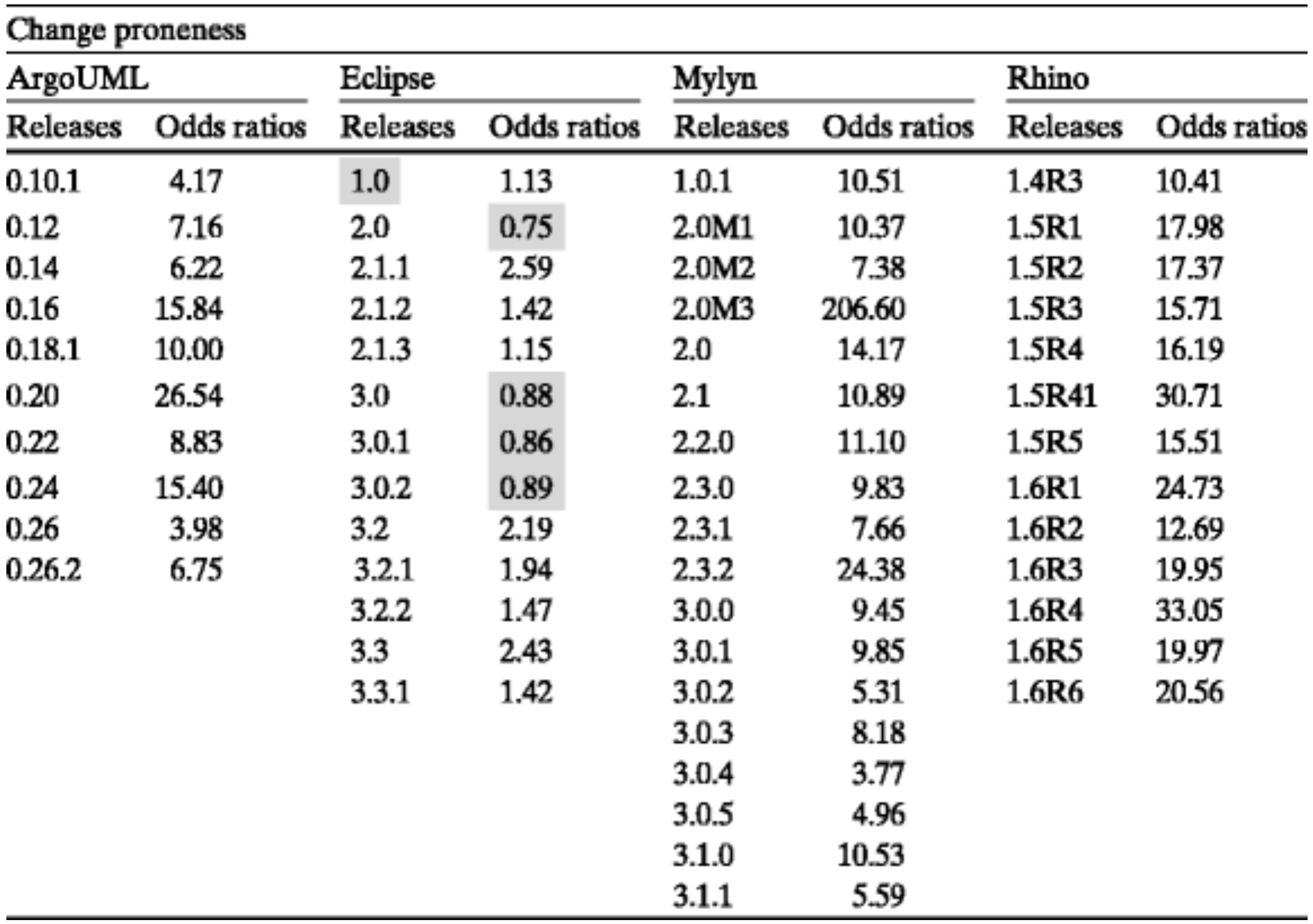
appariés
Test de Cochran

indépendants = non appariés
Test du χ2

Analyzing the output data

To finally describe the results

RQ1: Change-proneness odd ratios. Releases where Fisher's exact test did not show significant differences are highlighted in gray; odd ratio < 1 are also highlighted in gray

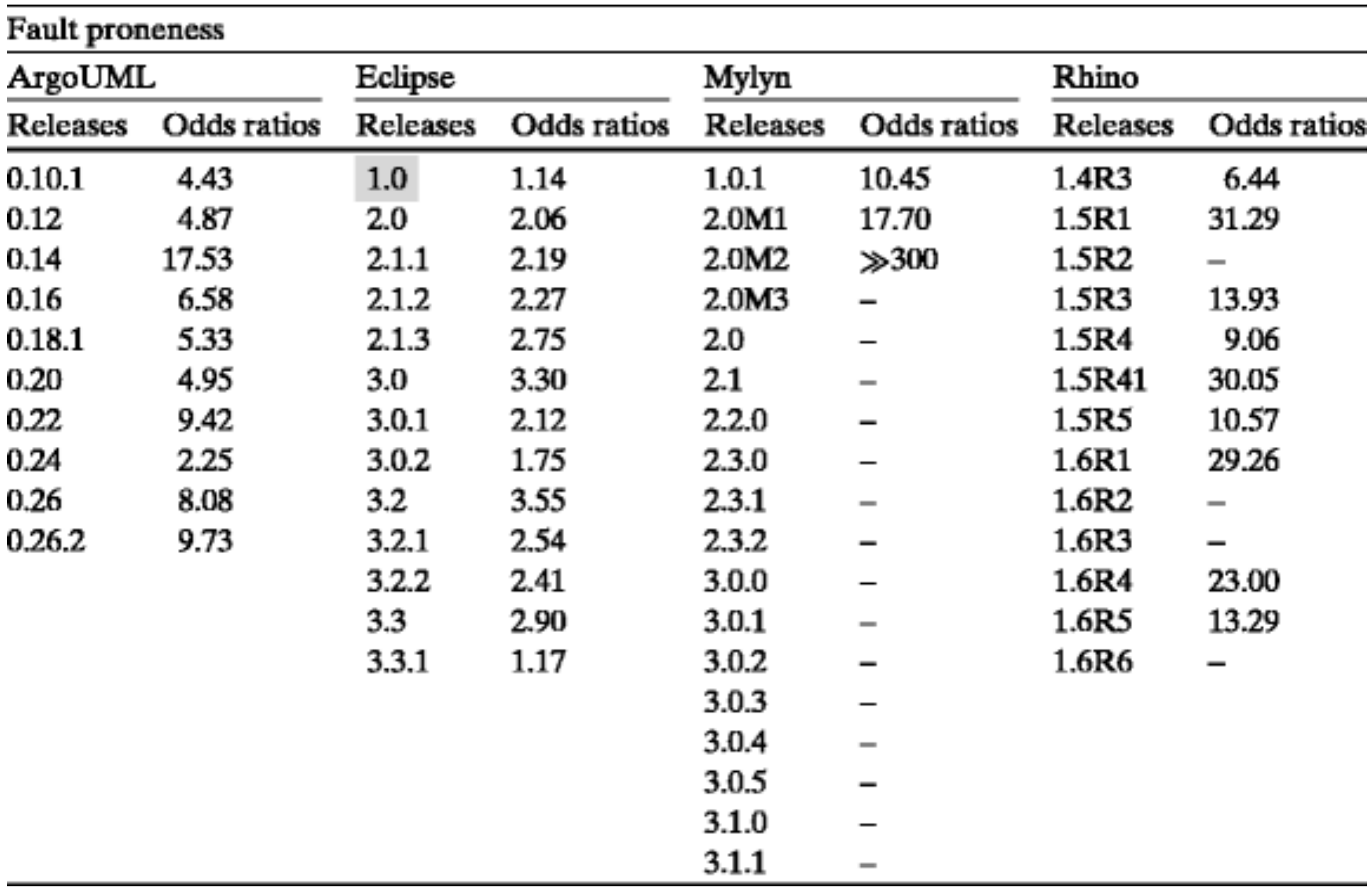


Change proneness							
ArgoUML		Eclipse		Mylyn		Rhino	
Releases	Odds ratios	Releases	Odds ratios	Releases	Odds ratios	Releases	Odds ratios
0.10.1	4.17	1.0	1.13	1.0.1	10.51	1.4R3	10.41
0.12	7.16	2.0	0.75	2.0M1	10.37	1.5R1	17.98
0.14	6.22	2.1.1	2.59	2.0M2	7.38	1.5R2	17.37
0.16	15.84	2.1.2	1.42	2.0M3	206.60	1.5R3	15.71
0.18.1	10.00	2.1.3	1.15	2.0	14.17	1.5R4	16.19
0.20	26.54	3.0	0.88	2.1	10.89	1.5R41	30.71
0.22	8.83	3.0.1	0.86	2.2.0	11.10	1.5R5	15.51
0.24	15.40	3.0.2	0.89	2.3.0	9.83	1.6R1	24.73
0.26	3.98	3.2	2.19	2.3.1	7.66	1.6R2	12.69
0.26.2	6.75	3.2.1	1.94	2.3.2	24.38	1.6R3	19.95
		3.2.2	1.47	3.0.0	9.45	1.6R4	33.05
		3.3	2.43	3.0.1	9.85	1.6R5	19.97
		3.3.1	1.42	3.0.2	5.31	1.6R6	20.56
				3.0.3	8.18		
				3.0.4	3.77		
				3.0.5	4.96		
				3.1.0	10.53		
				3.1.1	5.59		

Analyzing the output data

To finally describe the results

RQ2: Fault-proneness odd ratios. Releases where Fisher's exact test did not show significant differences are highlighted in gray

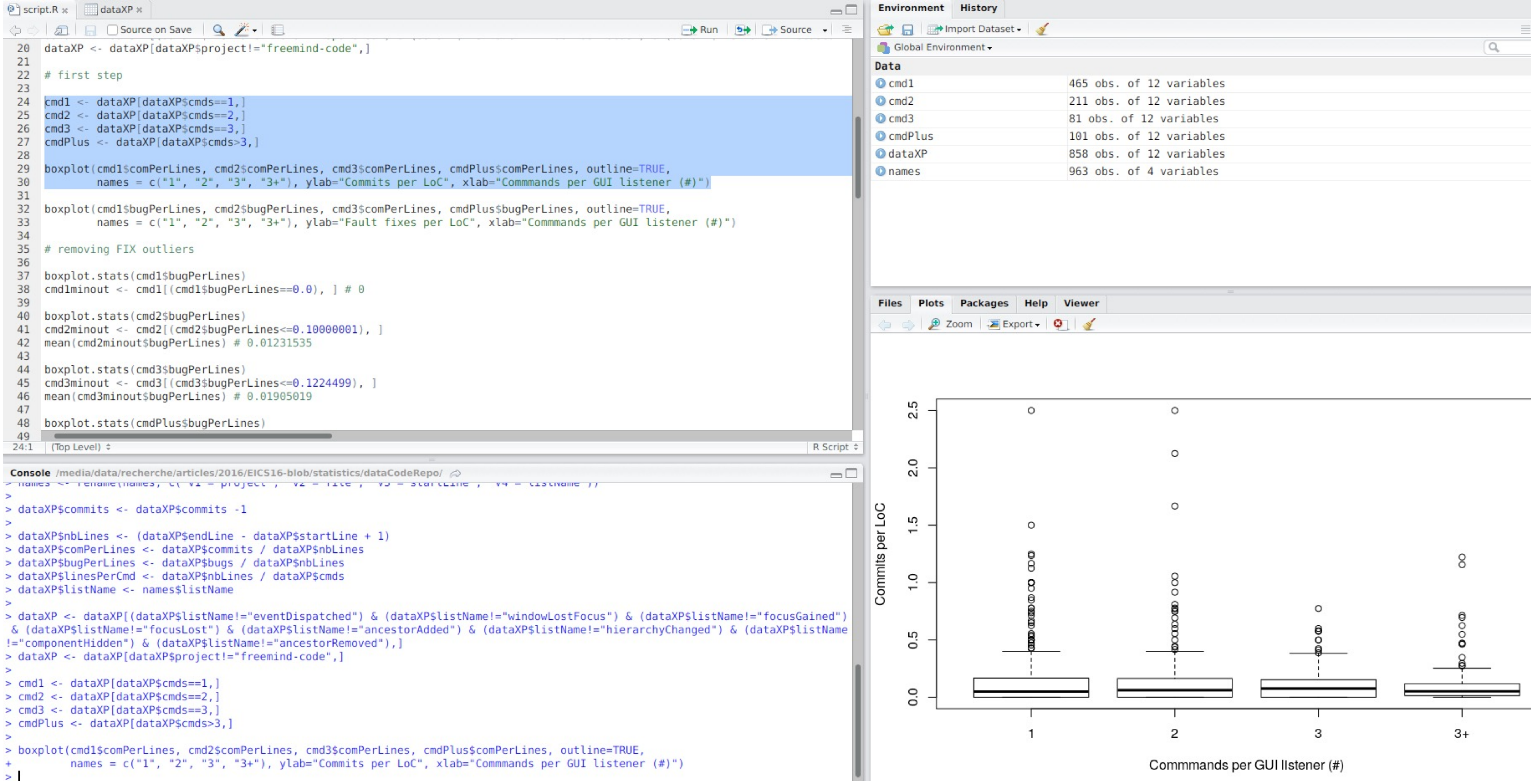


Fault proneness							
ArgoUML		Eclipse		Mylyn		Rhino	
Releases	Odds ratios	Releases	Odds ratios	Releases	Odds ratios	Releases	Odds ratios
0.10.1	4.43	1.0	1.14	1.0.1	10.45	1.4R3	6.44
0.12	4.87	2.0	2.06	2.0M1	17.70	1.5R1	31.29
0.14	17.53	2.1.1	2.19	2.0M2	≫300	1.5R2	–
0.16	6.58	2.1.2	2.27	2.0M3	–	1.5R3	13.93
0.18.1	5.33	2.1.3	2.75	2.0	–	1.5R4	9.06
0.20	4.95	3.0	3.30	2.1	–	1.5R41	30.05
0.22	9.42	3.0.1	2.12	2.2.0	–	1.5R5	10.57
0.24	2.25	3.0.2	1.75	2.3.0	–	1.6R1	29.26
0.26	8.08	3.2	3.55	2.3.1	–	1.6R2	–
0.26.2	9.73	3.2.1	2.54	2.3.2	–	1.6R3	–
		3.2.2	2.41	3.0.0	–	1.6R4	23.00
		3.3	2.90	3.0.1	–	1.6R5	13.29
		3.3.1	1.17	3.0.2	–	1.6R6	–
				3.0.3	–		
				3.0.4	–		
				3.0.5	–		
				3.1.0	–		
				3.1.1	–		

Analyzing the output data

R: statistical computing and graphics

https://www.r-project.org/



```r
dataXP <- dataXP[dataXP$project!="freemind-code",]

# first step

cmd1 <- dataXP[dataXP$cmds==1,]
cmd2 <- dataXP[dataXP$cmds==2,]
cmd3 <- dataXP[dataXP$cmds==3,]
cmdPlus <- dataXP[dataXP$cmds>3,]

boxplot(cmd1$comPerLines, cmd2$comPerLines, cmd3$comPerLines, cmdPlus$comPerLines, outline=TRUE,
        names = c("1", "2", "3", "3+"), ylab="Commits per LoC", xlab="Commmands per GUI listener (#)")

boxplot(cmd1$bugPerLines, cmd2$bugPerLines, cmd3$comPerLines, cmdPlus$bugPerLines, outline=TRUE,
        names = c("1", "2", "3", "3+"), ylab="Fault fixes per LoC", xlab="Commmands per GUI listener (#)")

# removing FIX outliers

boxplot.stats(cmd1$bugPerLines)
cmd1minout <- cmd1[(cmd1$bugPerLines==0.0), ] # 0

boxplot.stats(cmd2$bugPerLines)
cmd2minout <- cmd2[(cmd2$bugPerLines<=0.10000001), ]
mean(cmd2minout$bugPerLines) # 0.01231535

boxplot.stats(cmd3$bugPerLines)
cmd3minout <- cmd3[(cmd3$bugPerLines<=0.1224499), ]
mean(cmd3minout$bugPerLines) # 0.01905019

boxplot.stats(cmdPlus$bugPerLines)
```

Data	
cmd1	465 obs. of 12 variables
cmd2	211 obs. of 12 variables
cmd3	81 obs. of 12 variables
cmdPlus	101 obs. of 12 variables
dataXP	858 obs. of 12 variables
names	963 obs. of 4 variables

```
> names <- rename(names, c( V1 = "project", V2 = "file", V3 = "startLine", V4 = "listName" ))
>
> dataXP$commits <- dataXP$commits -1
>
> dataXP$nbLines <- (dataXP$endLine - dataXP$startLine + 1)
> dataXP$comPerLines <- dataXP$commits / dataXP$nbLines
> dataXP$bugPerLines <- dataXP$bugs / dataXP$nbLines
> dataXP$linesPerCmd <- dataXP$nbLines / dataXP$cmds
> dataXP$listName <- names$listName
>
> dataXP <- dataXP[(dataXP$listName!="eventDispatched") & (dataXP$listName!="windowLostFocus") & (dataXP$listName!="focusGained")
 & (dataXP$listName!="focusLost") & (dataXP$listName!="ancestorAdded") & (dataXP$listName!="hierarchyChanged") & (dataXP$listName
!="componentHidden") & (dataXP$listName!="ancestorRemoved"),]
> dataXP <- dataXP[dataXP$project!="freemind-code",]
>
> cmd1 <- dataXP[dataXP$cmds==1,]
> cmd2 <- dataXP[dataXP$cmds==2,]
> cmd3 <- dataXP[dataXP$cmds==3,]
> cmdPlus <- dataXP[dataXP$cmds>3,]
>
> boxplot(cmd1$comPerLines, cmd2$comPerLines, cmd3$comPerLines, cmdPlus$comPerLines, outline=TRUE,
+         names = c("1", "2", "3", "3+"), ylab="Commits per LoC", xlab="Commmands per GUI listener (#)")
> 
```

Drawing conclusions based on the results

Just answer and comment all the RQs according to the data analysis

RQ1, RQ2: “*classes participating in antipatterns are significantly more likely to be subject to changes and to be involved in fault-fixing changes than other classes.*”

Threats to validity

- Mandatory
- Explain how you managed the possible problems (the threats) that threaten the validity of the experiment
  - So, have to think about it during the whole process of the experiment
- Threats to validity categories
  - Construct validity, internal validity, external validity, conclusion validity, reliability validity, etc.
- Interesting article:

  "*Threats to validity in software engineering research: A critical reflection*", Roberto Verdecchia, Emelie Engström, Patricia Lago, Per Runeson, Qunying Song, Information and Software Technology, Volume 164, 2023.

Threats to validity

- Construct validity threats
  - Relate to the perceived overall validity of the experiments
  - Concern the relation between theory and observation
- Examples:
  - Human subjects already known how to use on of the tools. How did you manage that?
  - How did you manage the tiredness of the subjects?Measurement errors, approximations. Example: git logs may not precisely help in identifying fault-fixes

Threats to validity

- Internal validity threats
  - The phenomena, variables (you may not control) that may affect the results of the study
- Example:
  - “the algorithm layout and the drawing of the UML associations differ from Tool1 to Tool2”

Threats to validity

- External validity threats
  - Concern the possibility to generalise our results.
- Examples:
  - “we studied four systems having different sizes and belonging to different domains.”
  - “we used a particular yet representative subset of antipatterns”

Threats to validity

- Reliability validity threats
  - Concern the possibility to replicate the study.
- Examples:
  - “we studied four systems having different sizes and belonging to different domains.”
  - “The source code repositories and issue-tracking systems of the studied systems are available to obtain the same data”
  - “The raw data used to compute the statistics is on-line”

Empirical software engineering conferences, journals, and references

- Read papers from these conferences/journals to look at the studies
- ESE – journal of empirical software engineering
  - https://www.springer.com/computer/swe/journal/10664
  - https://scholar.google.fr/citations?hl=fr&vq=eng_softwaresystems&view_op=list_hcore&venue=w7tXCm-brrIJ.2016
- ESEM – International Symposium on Empirical Software Engineering and Measurement
  - http://www.esem-conferences.org/
  - https://scholar.google.fr/citations?hl=fr&view_op=list_hcore&venue=TWG2GULnchAJ.2016
- (other top-tier conferences/journals may also have empirical studies)

Keep in mind that

- Empirical studies must be:
  - insightful
  - rigorous and methodological
  - replicable (provides the data sets, the R scripts, readme files, etc.)
- Empirical studies must not:
  - be a demonstration of statistical methods
  - overclaim the results
  - transform, modify data without any relevant explanation

References

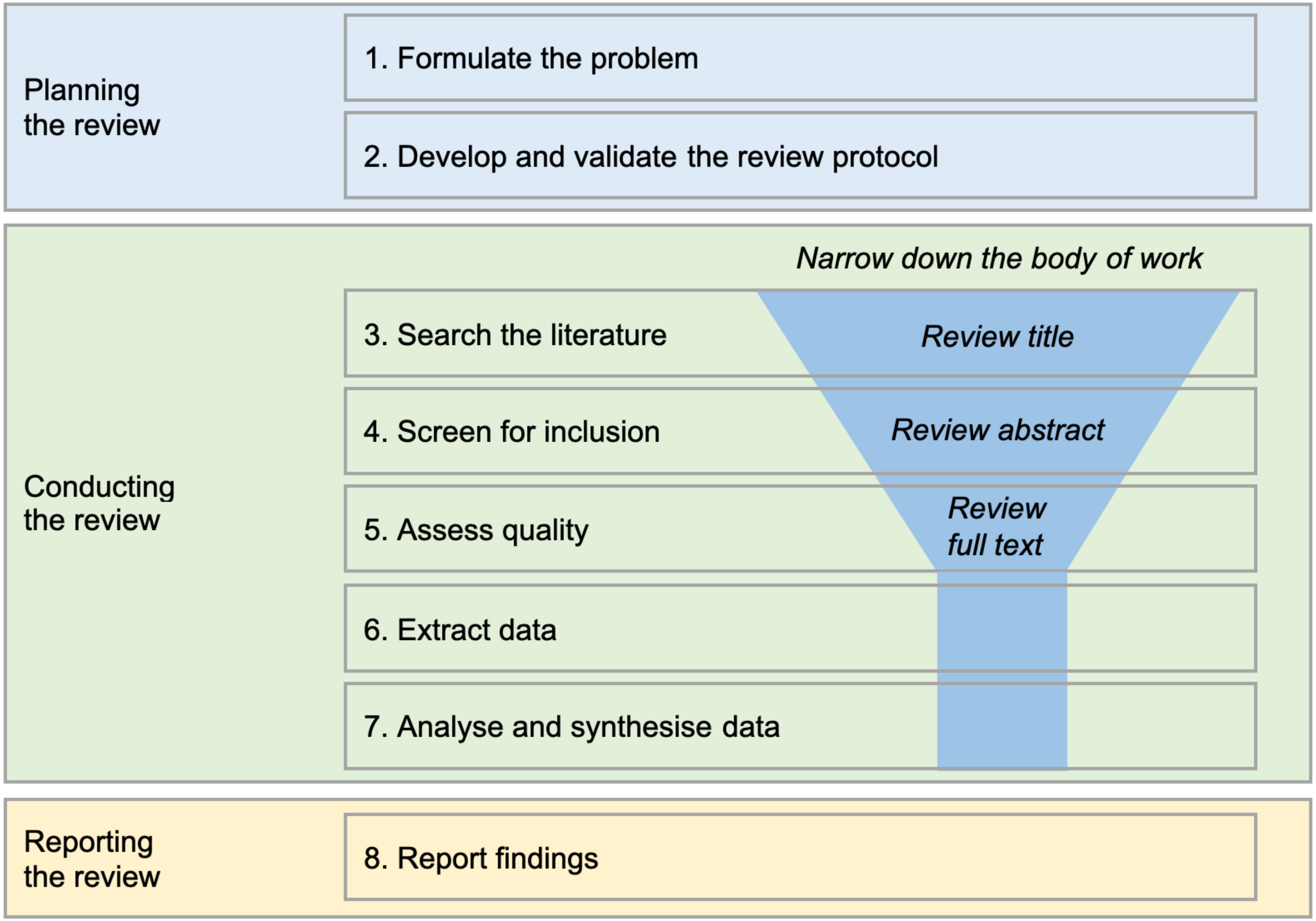
- Guide to Advanced Empirical Software Engineering, Shull et al.
- Handbook of parametric and nonparametric statistical procedures, sheskin
- Experimentation in software engineering, Wohlin et al.
- Empirical Methods and Studies in Software Engineering, Conradi et al
- Preliminary guidelines for empirical research in software engineering, Kitchenham et al.
- The Role of Experimentation in Software Engineering: Past, Current, and Future, Basili
- Five Misunderstandings About Case-Study Research. B. Flyvbjerg

LITERATURE REVIEW

Know your topic!

- A **literature review** is an overview of the previously published works on a topic
- A **systematic review** is essentially a literature review focused on a research question, trying to identify, appraise, select and synthesize all high-quality research evidence and arguments relevant to that question.
- A **meta-analysis** is typically a systematic review using statistical methods to effectively combine the data used on all selected studies to produce a more reliable result.

Systematic Literature Review



Planning the review

1. Formulate the problem

2. Develop and validate the review protocol

Conducting the review

Narrow down the body of work

3. Search the literature

Review title

4. Screen for inclusion

Review abstract

5. Assess quality

Review full text

6. Extract data

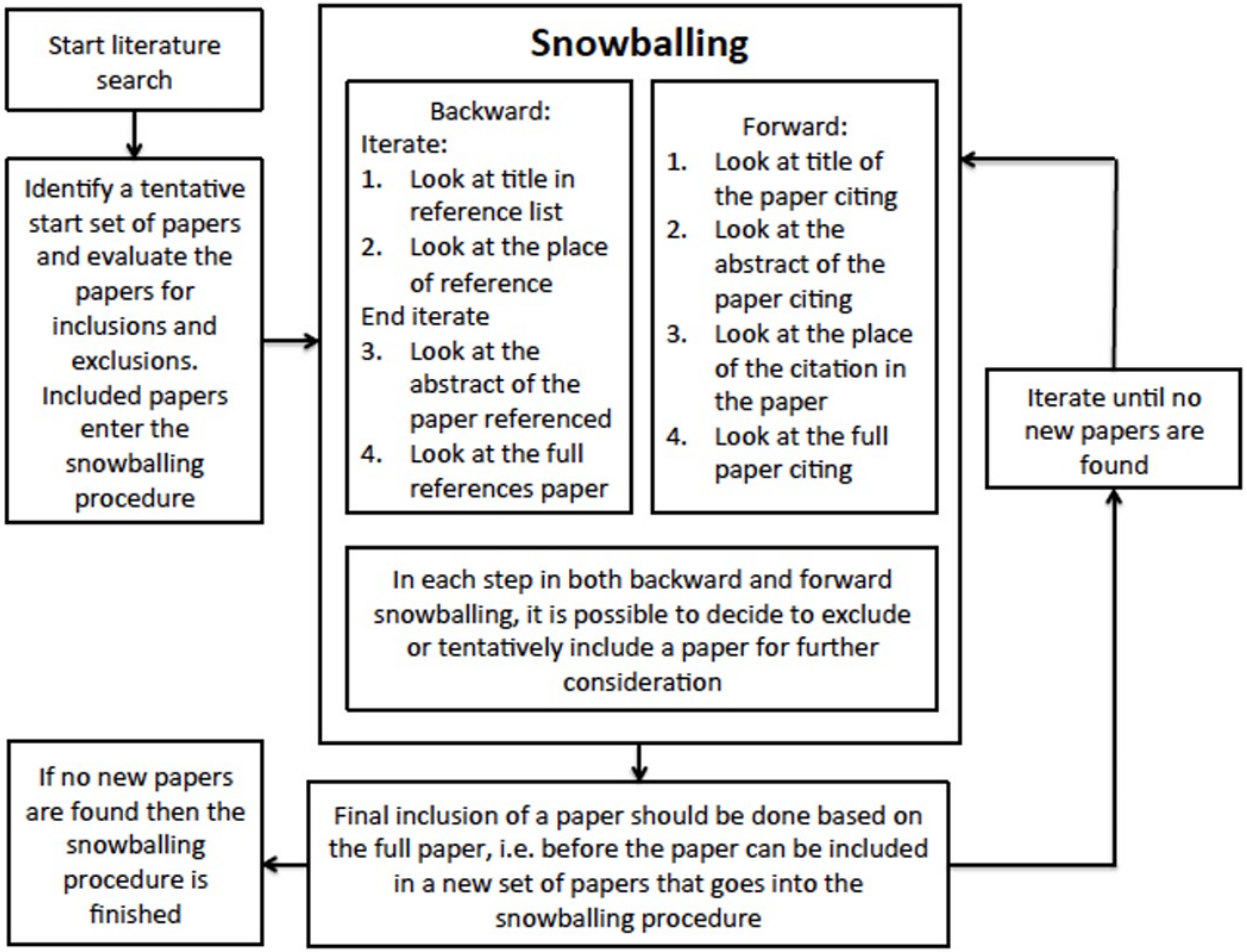
7. Analyse and synthesise data

Reporting the review

8. Report findings

Kitchenham, B. and Charters, S. (2007) Guidelines for Performing Systematic Literature Reviews in Software Engineering, Technical Report EBSE 2007-001, Keele University and Durham University Joint Report.

Snowballing in Literature Review



Start literature search

Identify a tentative start set of papers and evaluate the papers for inclusions and exclusions. Included papers enter the snowballing procedure

Snowballing

Backward:
Iterate:
1. Look at title in reference list
2. Look at the place of reference
End iterate
3. Look at the abstract of the paper referenced
4. Look at the full references paper

Forward:
1. Look at title of the paper citing
2. Look at the abstract of the paper citing
3. Look at the place of the citation in the paper
4. Look at the full paper citing

In each step in both backward and forward snowballing, it is possible to decide to exclude or tentatively include a paper for further consideration

Iterate until no new papers are found

Final inclusion of a paper should be done based on the full paper, i.e. before the paper can be included in a new set of papers that goes into the snowballing procedure

If no new papers are found then the snowballing procedure is finished

Claes Wohlin. 2014. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering (EASE '14). Association for Computing Machinery, New York, NY, USA, Article 38, 1–10. https://doi.org/10.1145/2601248.2601268

Where to find research papers?

- Search engine (Google Scholar)
- Publishers (e.g., IEEE Xplore, ACM DL, SpringerLink)
- National repositories (e.g. HAL)
- Institutional repositories
- Author homepages

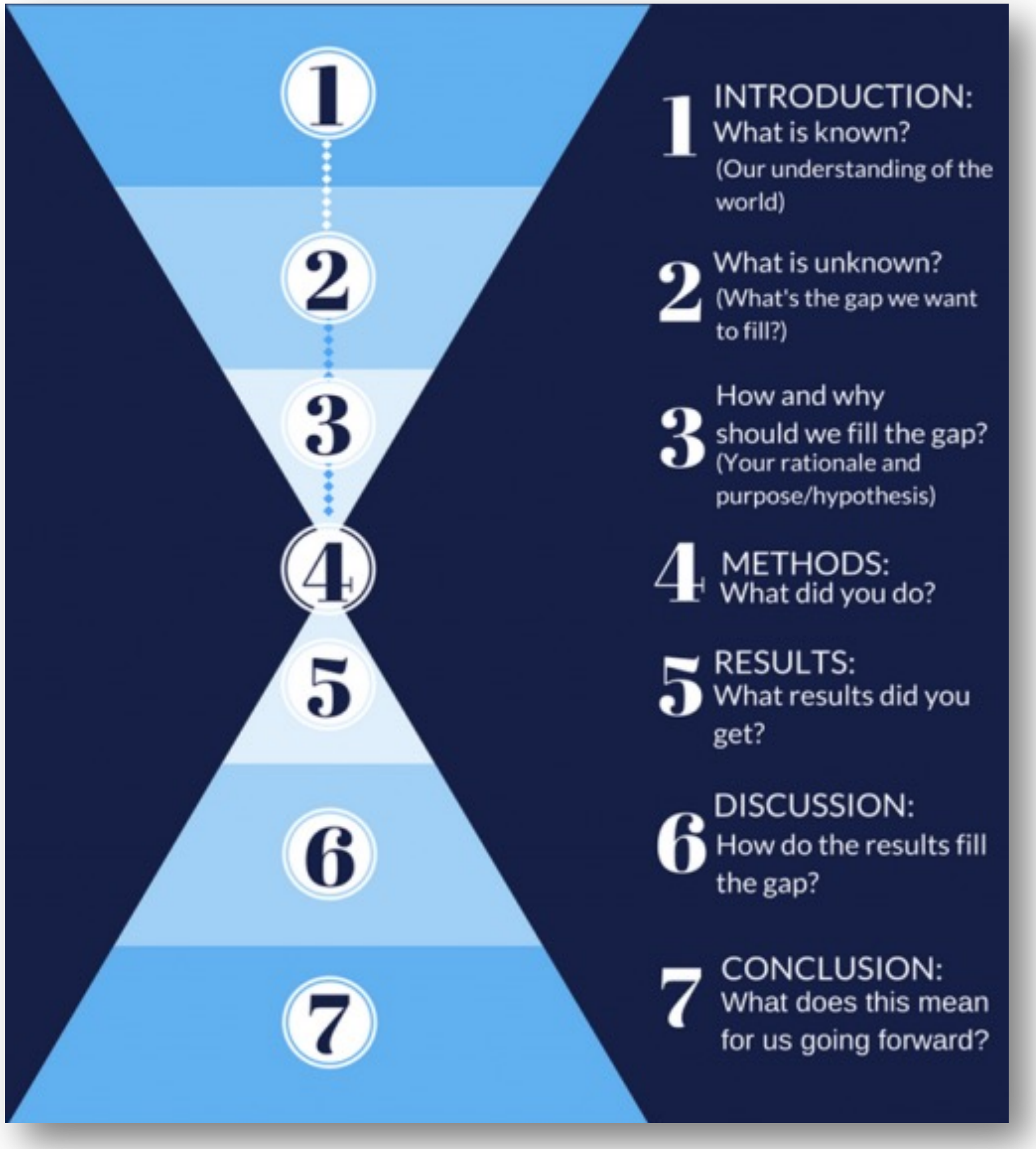
A research paper… Wait, WAT?

- Scientific Magazine:
  - Communication of the ACM, IEEE Computer, IEEE Software
- Journals:
  - General, e.g., TSE, TOSEM, JSS, JoT
  - Domain-specific, e.g., SoSyM, EMSE
- Conferences
  - General, e.g., ICSE, ASE, FSE, SPLASH, PLDI
  - Domain-specific, e.g., MODELS, SLE, SPLC

Note: some journals/conferences are more prestigious than others!

A research paper… Wait, WAT?

- A paper is always well-structured
- Should make systematic the screening
  - Abstract
  - Intro/Conclu
  - Methods/Results



1 INTRODUCTION: What is known? (Our understanding of the world)

2 What is unknown? (What's the gap we want to fill?)

3 How and why should we fill the gap? (Your rationale and purpose/hypothesis)

4 METHODS: What did you do?

5 RESULTS: What results did you get?

6 DISCUSSION: How do the results fill the gap?

7 CONCLUSION: What does this mean for us going forward?

https://blog.wordvice.com/seminar-how-to-write-an-effective-research-paper/

What is expected for SEM?

- A minimal snowballing review

- Draw a general Research Question
  - Existing foundations? technologies? framework?
  - Use in practice? Impact?
- Start from 1 representative paper
- Expand to 4-5 carefully choosen papers
- Integrate additional materials (web, videos, experiments…)
- Organize and report the key findings

What is expected for SEM?

- Presentation:
    - 10" (+5" discussion)
    - Well structured, e.g.
        - Context, RQs, methods, results, conclu