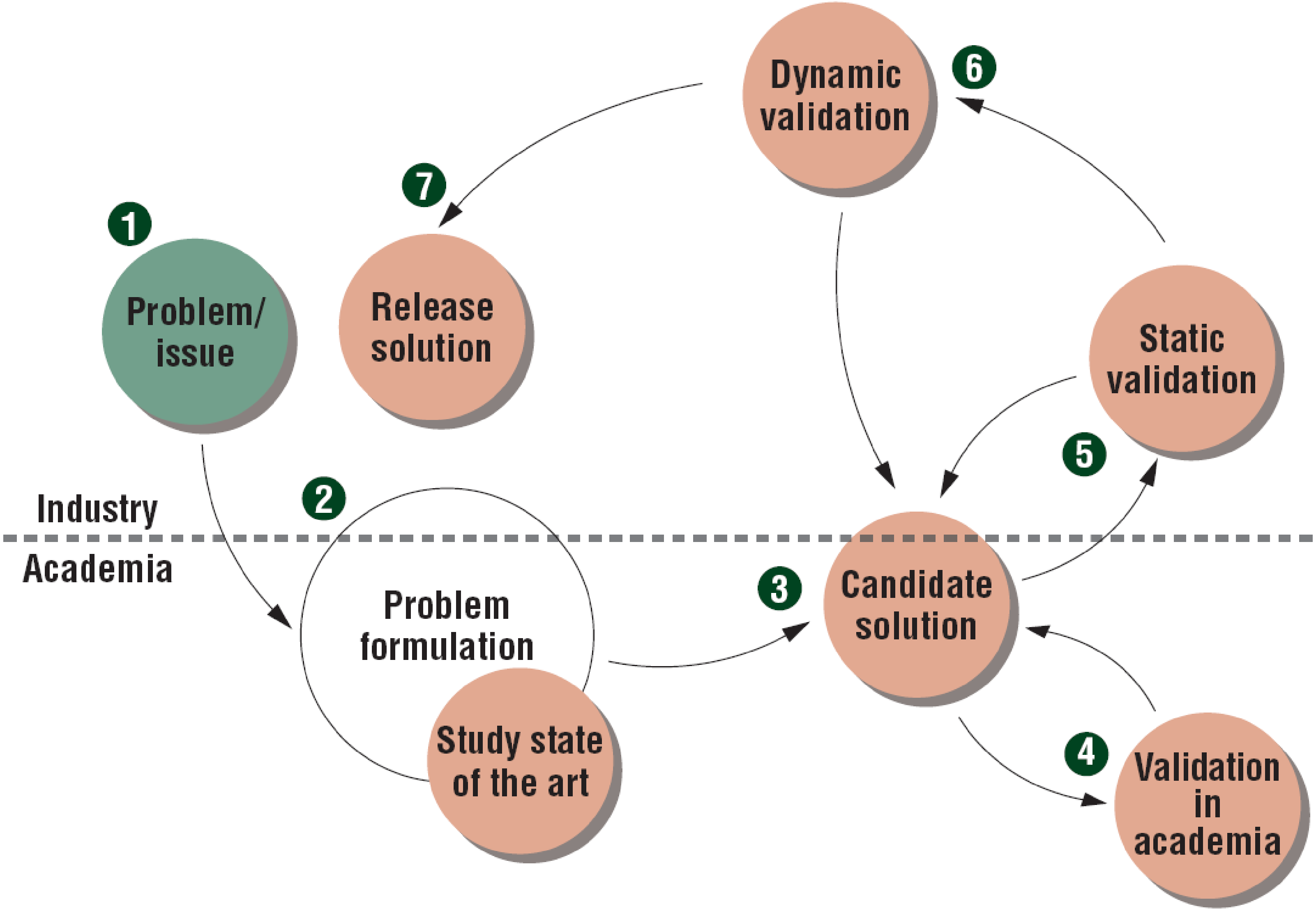
Support session Case Study

Our way of doing research: knowledge exchange

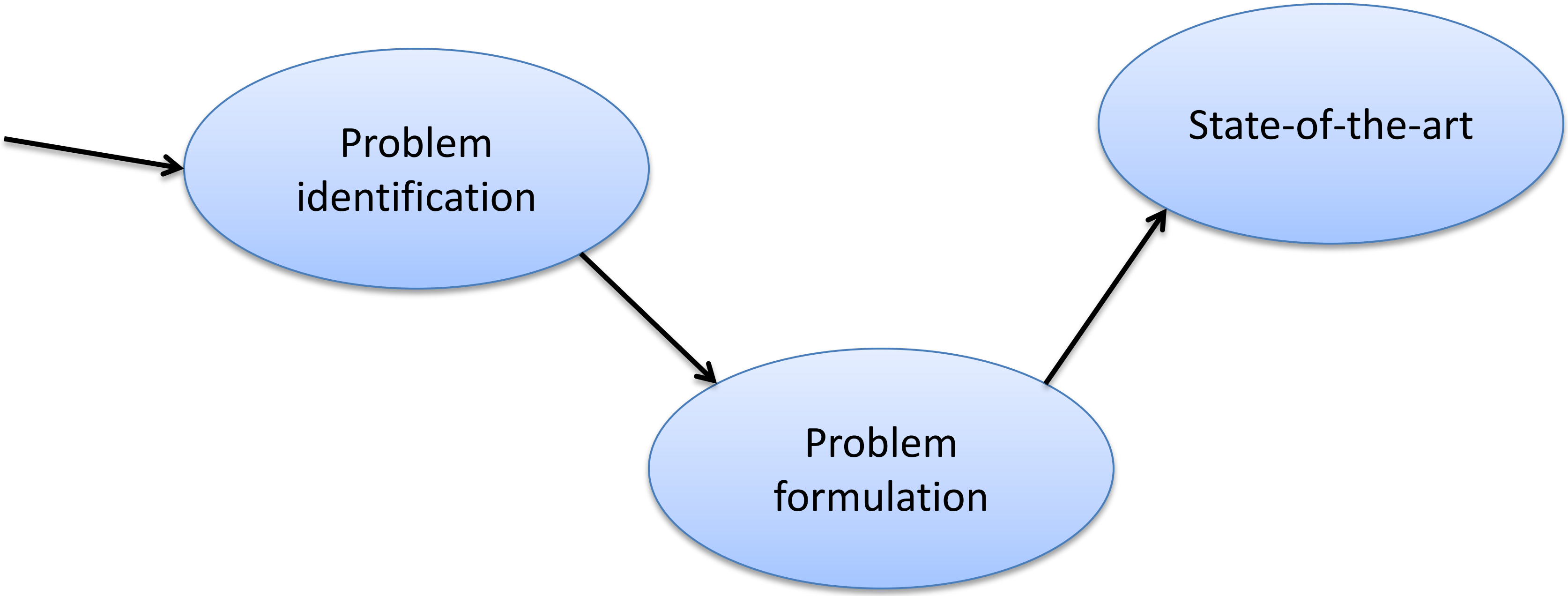


1. Problem/Issue

- Interviews to capture needs
- Process assessment
- Problems prioritized by industry

2. State-of-the-art and problem formulation

- State-of-the-art study



3. Candidate solution

- A solution to the problem formulation or part of it is proposed based on literature and own inventions in close collaboration with industry

4. Academic validation

- Experiment with students
- Check applicability of solution
- Refine solution
  - Low cost
  - Low risk

5. Static validation in industry

- Offline validation – refinement/tailoring
  - Interviews
  - Workshops
  - …
- Refine solution based on feedback

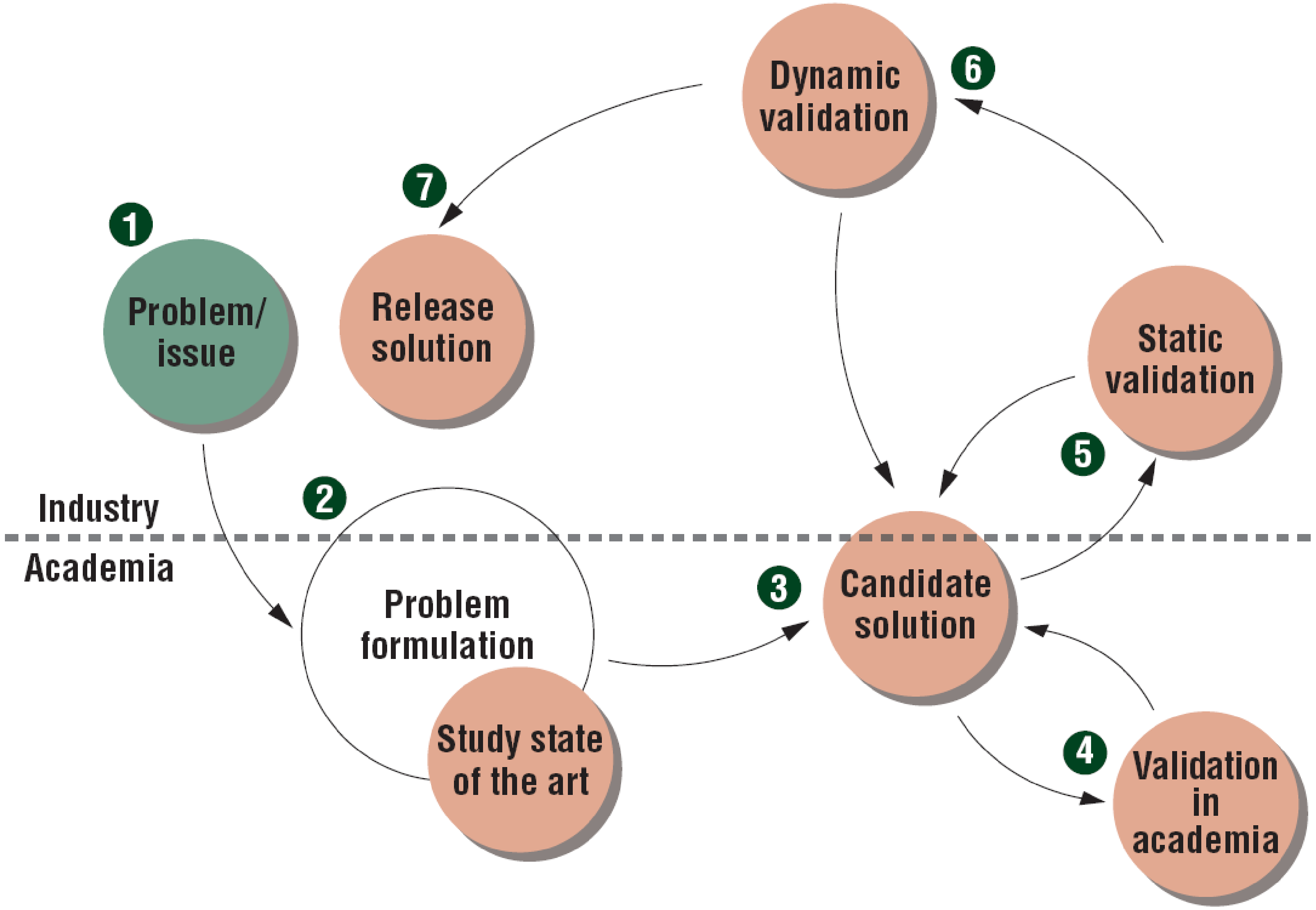
6. Dynamic validation in industry

- Pilot project – real use, but limited
- Evaluate real usage
  - But limit risks and costs
- Scalability, usability, usefulness

7. Release

- Two aspects
  - Academically: publications
  - Practically: Released for wider use – in organization – outside organization

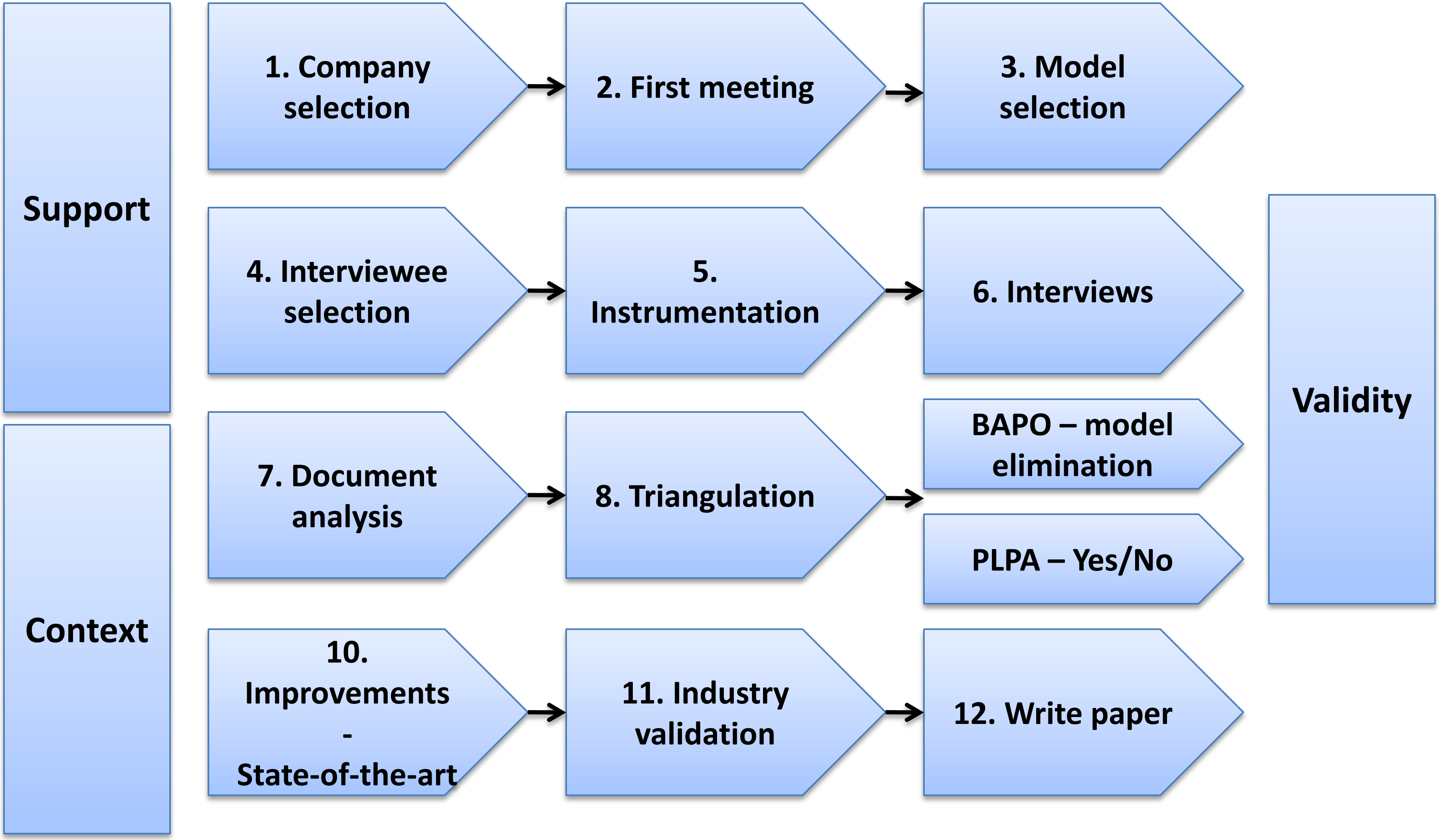
Our way of doing research: knowledge exchange



SPLE Case study

- Your mini-research project
- Process assessment in industry
- Identify improvement potential
  - Propose solutions

Process


Support
1. Company selection
2. First meeting
3. Model selection
4. Interviewee selection
5. Instrumentation
6. Interviews
Validity
Context
7. Document analysis
8. Triangulation
BAPO – model elimination
PLPA – Yes/No
10. Improvements - State-of-the-art
11. Industry validation
12. Write paper

1. Company selection

2. First meeting

- Commitment
- Input for model selection
- Interviewee selection

3. Model selection

- # Products
- Large scale reuse between products
- Common platform

- Not a clear line for when to use what

4. Interviewee selection

- Explain what you need (and why)
  - More is always better

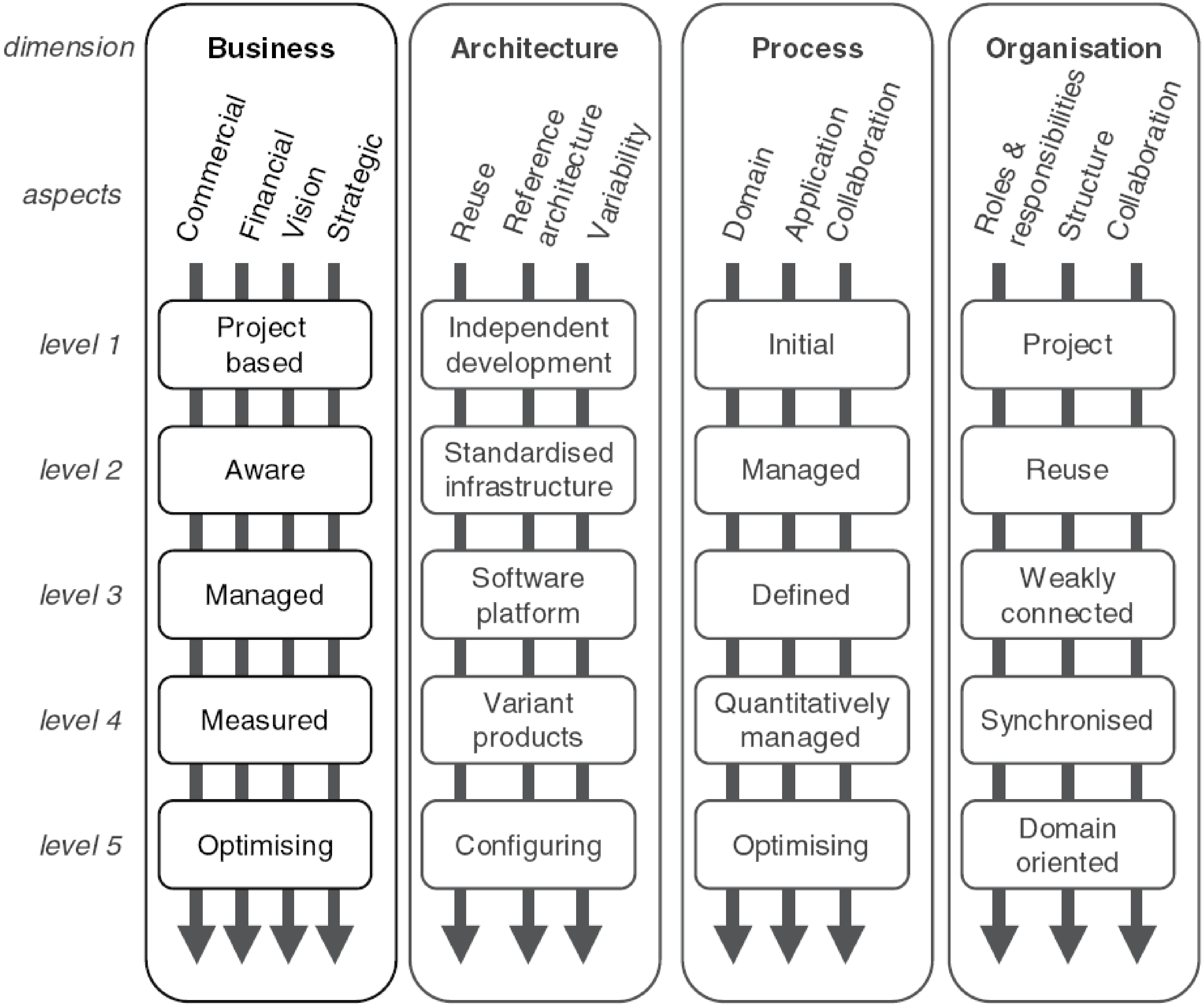
- Regardless of model (BAPO or PLPA) you should cover
  - Business
  - Architecture
  - Process
  - Organization

5. Interview instrument

- Questions
  - How long is the interview
  - How many questions can you cover?
  - 20-30 questions per hour

- Focus on asking about **what** not how

- All questions to all roles?
  - Alignment
  - Might not seem relevant for interviewee



dimension	Business	Architecture	Process	Organisation
aspects	Commercial Financial Vision Strategic	Reuse Reference architecture Variability	Domain Application Collaboration	Roles & responsibilities Structure Collaboration
level 1	Project based	Independent development	Initial	Project
level 2	Aware	Standardised infrastructure	Managed	Reuse
level 3	Managed	Software platform	Defined	Weakly connected
level 4	Measured	Variant products	Quantitatively managed	Synchronised
level 5	Optimising	Configuring	Optimising	Domain oriented

- **Main criteria** are essential for product line development and have to be fulfilled:
  - The business unit develops more than one product.
  - Products have common features.
  - Products have common qualities.
- **Inclusion criteria** indicate that product lines already exist:
  - The same part of software is used in more than one product.
- **Supporting criteria** apply if a business unit has problems that the PLA addresses:
  - The business unit has quality problems.
  - The business unit has complexity problems.
  - The business unit expects increasingly differentiated products.
- **Exclusion criteria** rule out an economically advantageous product line:
  - There is an immature, instable market for the products.
  - There is technological change.
  - The software is small; optimization will not be profitable.
  - The software development effort is negligible. It would be better to focus on other improvements.
  - New product development is too seldom.
  - The business unit develops specific, commissioned custom products.
- **Additional information** is useful data that cannot be assigned to one of the preceding criteria:
  - the competitive situation

Traceability

Question	Model element
Is product quality an aspect considered in the architecture?	BAPO-A: Product quality (Level 3-)
…	….

6. Interviews

- Too many interviewers is frightening
    - One asking… One taking notes…
    - Tape recorder
    - Assure them it’s anonymous

6. Interviews

Terminology and defs

- The language at companies is different from what you read in your papers
  - Be clear and explicit
  - Prepare yourself
  - What is a SPL in other words?

6. Interviews

- Be on time
- Welcome the person, present yourself
- Explain purpose
- Explain what the data will be used for
  - Assure anonymity
- Ask questions
- Have a _very_ open-ended question in the end (things missed?)
- Thank them!
- Take 10 after an interview and summarize

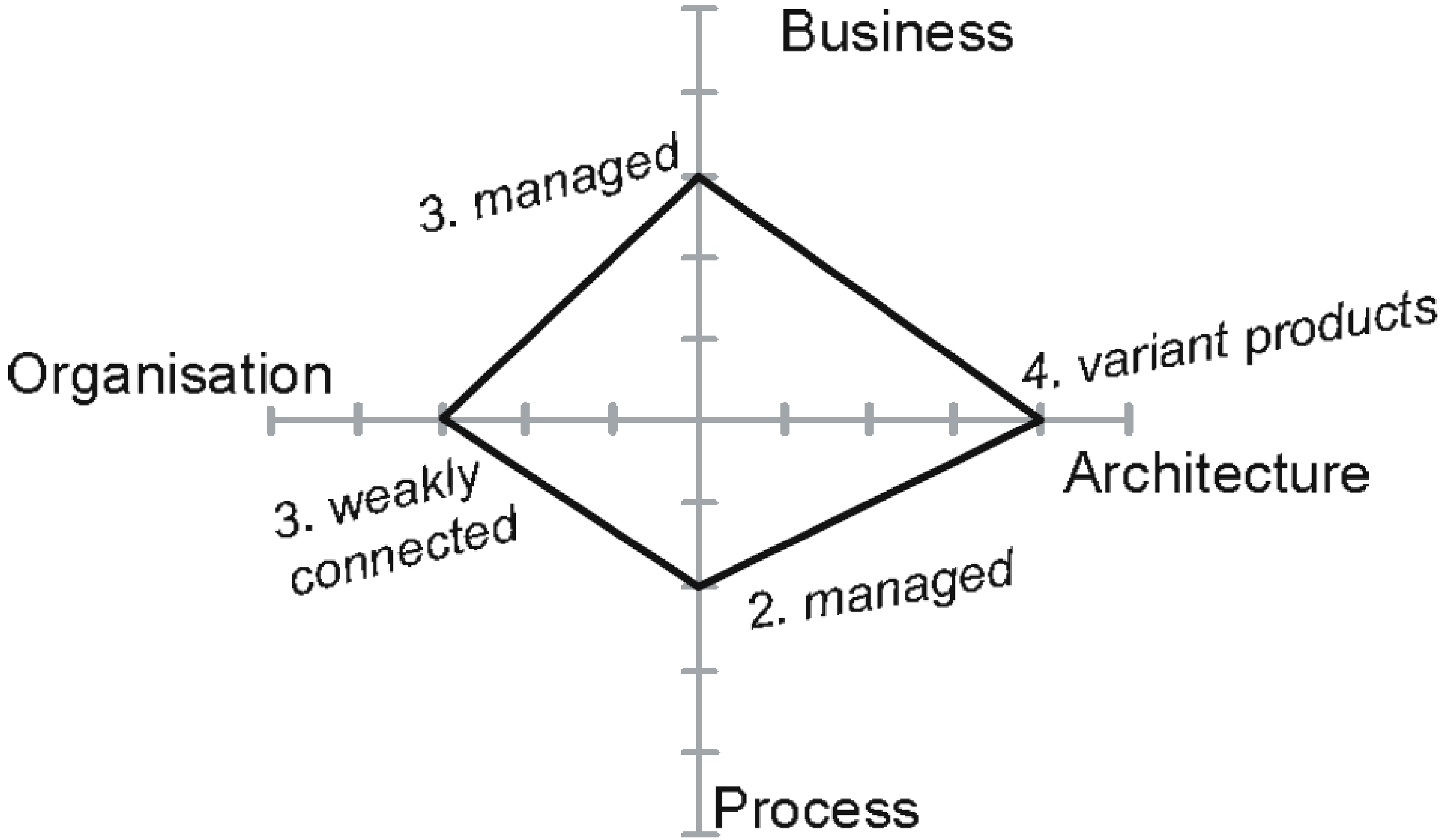
7. Document analysis

- Double check interviews
- New information

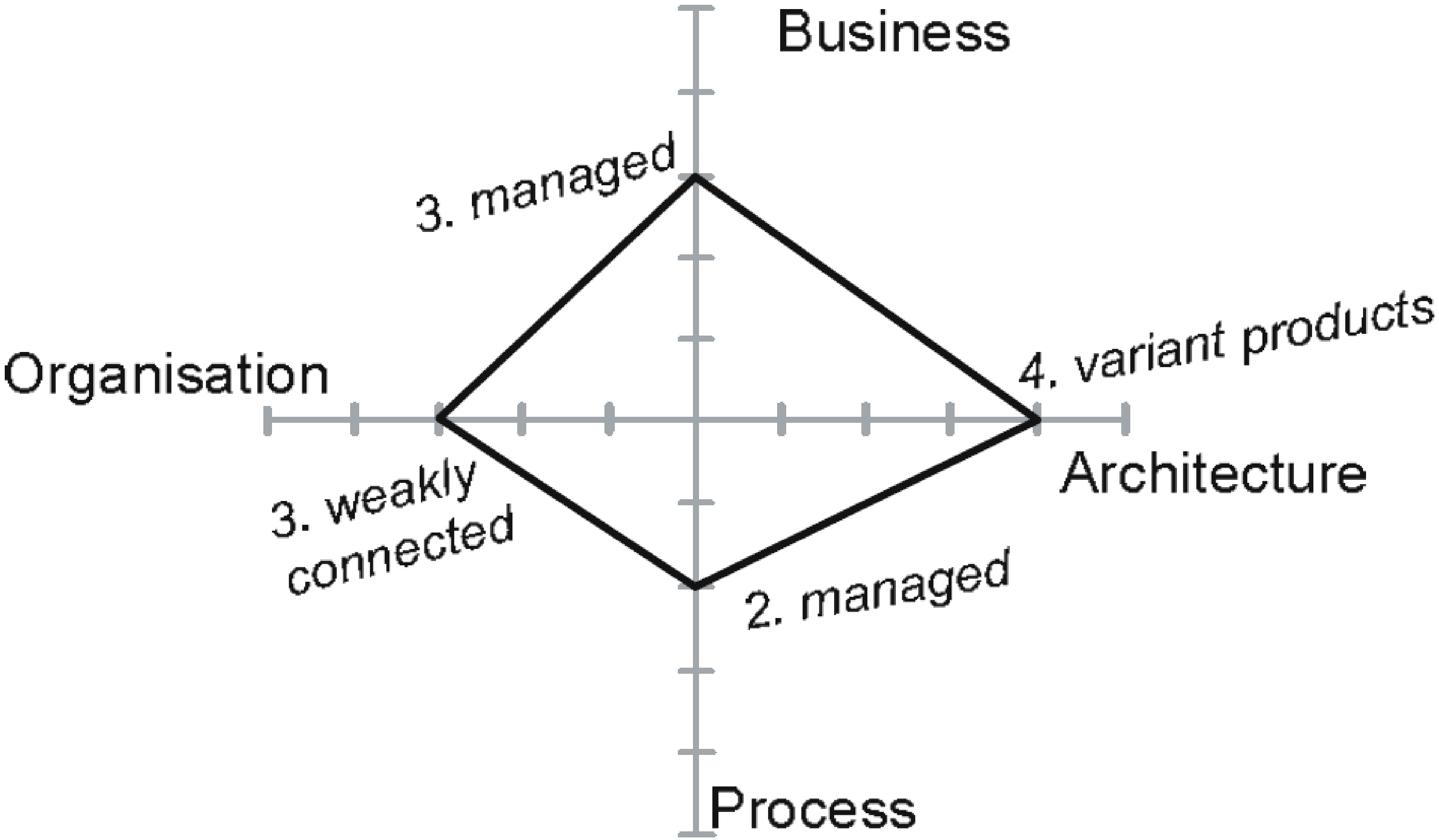
8. Triangulation

- Roles
- Interviews and documents

9. Model elimination



10. Improvements BAPO



10. Improvements PLPA

- Yes
  - Suggest transition
- No
  - Why not
  - What are the obstacles
  - What needs to change to enable a SPL approach?

10. Improvements

- Risk
  - – Big/small improvement
- Cost
  - Initiation threshold – education, rework etc.
- Benefit

State-of-the-art

- Literature

References

- References to research findings are an essential part of any research paper
    - The references should be used to strengthen your argument – and to show that you have done your homework
- Usually you summarize the research finding in your own words and then cite the source
- Example:
    - Disciplined CM practices have shown to decrease defect rates by 10% in a case study by Svensson et al [2], in a company of similar size to the one in this assignment.
- Always read the paper you reference

References cont.

- Always acquire the original article (no pre/off-print)
- Check “trustworthiness” – peer reviewed?
  - In what conference, workshop, journal is it published?
  - Is the source peer-reviewed?
  - Peer-review implies some level of quality/trustworthiness of the work

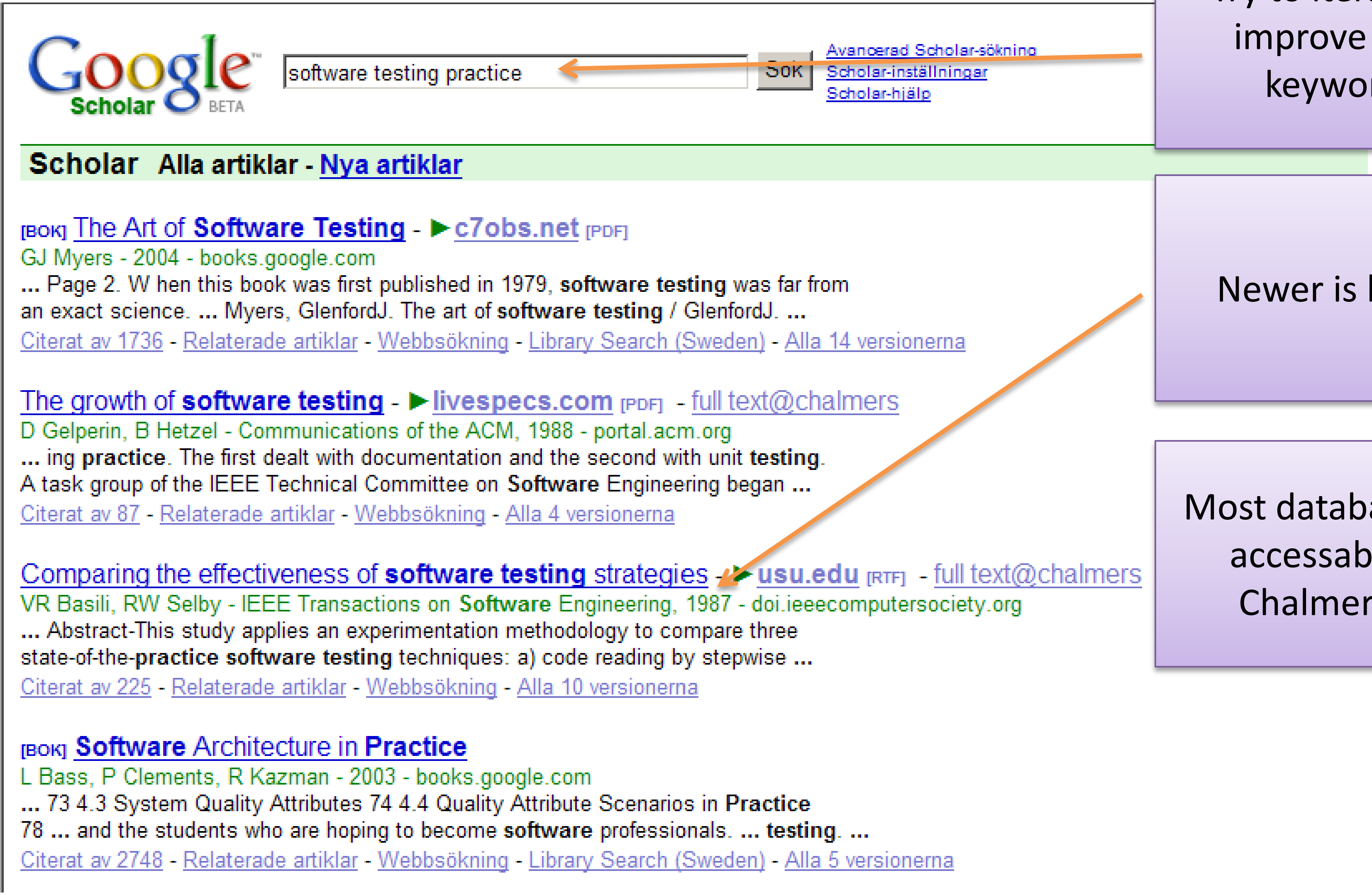
No Wikipedia

An example of finding a paper – and a process

- How do you go about finding research literature?
  - Search – keywords
  - Check trustworthiness
  - Scrutinize findings
    - Read abstract
    - Read conclusions
    - Read full paper
  - Use the finding

Google scholar
IEEE Explore
ACM digital library
www.engineeringvillage.com

Searching



software testing practice

Scholar **Alla artiklar** - **Nya artiklar**

[BOK] The Art of **Software Testing** - ▶ c7obs.net [PDF]
GJ Myers - 2004 - books.google.com
... Page 2. W hen this book was first published in 1979, **software testing** was far from an exact science. ... Myers, GlenfordJ. The art of **software testing** / GlenfordJ. ...
Citerat av 1736 - Relaterade artiklar - Webbsökning - Library Search (Sweden) - Alla 14 versionerna

The growth of **software testing** - ▶ livespecs.com [PDF] - full text@chalmers
D Gelperin, B Hetzel - Communications of the ACM, 1988 - portal.acm.org
... ing **practice**. The first dealt with documentation and the second with unit **testing**. A task group of the IEEE Technical Committee on **Software** Engineering began ...
Citerat av 87 - Relaterade artiklar - Webbsökning - Alla 4 versionerna

Comparing the effectiveness of **software testing** strategies - ▶ usu.edu [RTF] - full text@chalmers
VR Basili, RW Selby - IEEE Transactions on **Software** Engineering, 1987 - doi.ieeecomputersociety.org
... Abstract-This study applies an experimentation methodology to compare three state-of-the-**practice** **software testing** techniques: a) code reading by stepwise ...
Citerat av 225 - Relaterade artiklar - Webbsökning - Alla 10 versionerna

[BOK] **Software** Architecture in **Practice**
L Bass, P Clements, R Kazman - 2003 - books.google.com
... 73 4.3 System Quality Attributes 74 4.4 Quality Attribute Scenarios in **Practice** 78 ... and the students who are hoping to become **software** professionals. ... **testing**. ...
Citerat av 2748 - Relaterade artiklar - Webbsökning - Library Search (Sweden) - Alla 5 versionerna

Try to iteratively improve your keywords

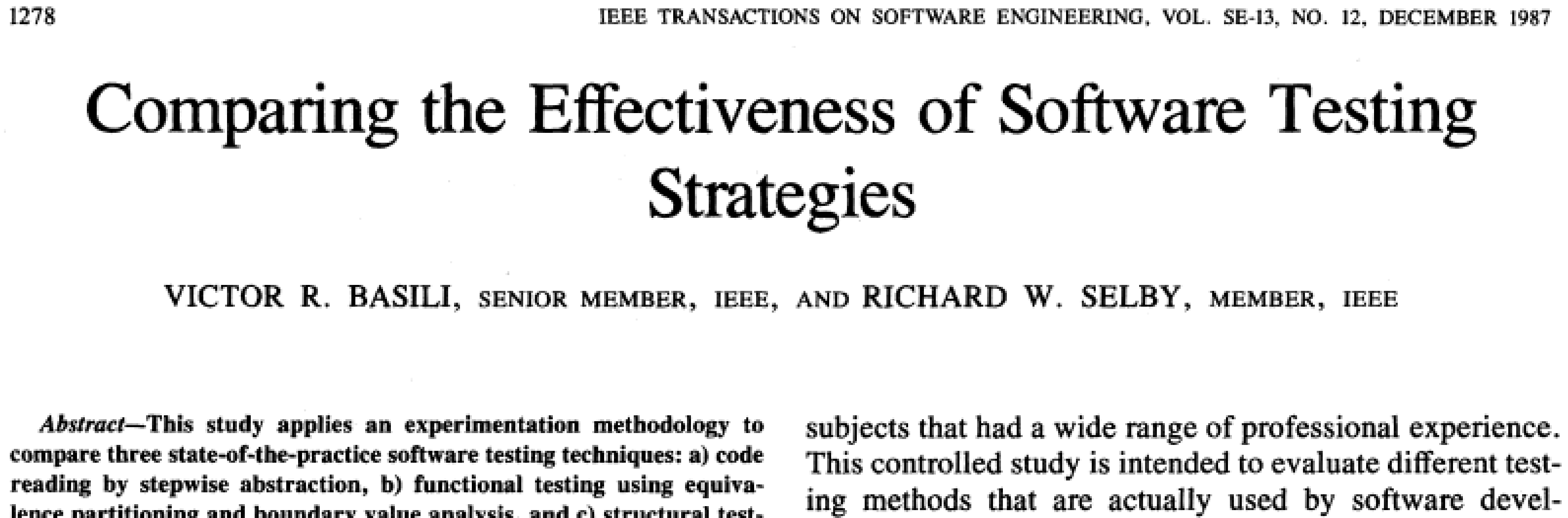
Newer is better

Most databases are accessable on Chalmers IPs

Check trustworthiness
Peer-reviewed?

- Most major conferences and journals are peer reviewed.
  - Is it published in a conference, journal or workshop?

Google it if unsure



1278 IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. SE-13, NO. 12, DECEMBER 1987

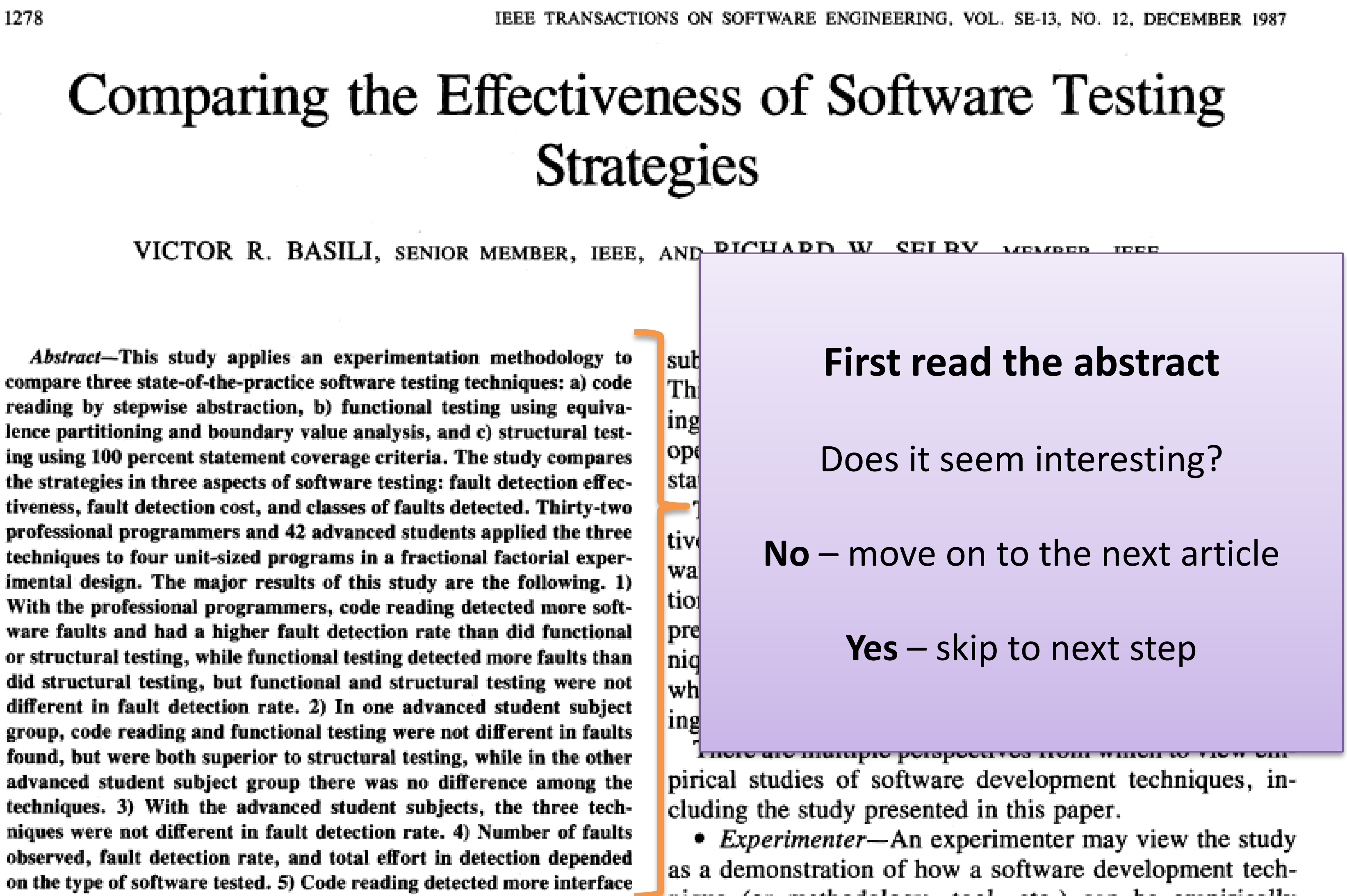
Comparing the Effectiveness of Software Testing Strategies

VICTOR R. BASILI, SENIOR MEMBER, IEEE, AND RICHARD W. SELBY, MEMBER, IEEE

Abstract—This study applies an experimentation methodology to compare three state-of-the-practice software testing techniques: a) code reading by stepwise abstraction, b) functional testing using equivalence partitioning and boundary value analysis, and c) structural test-

subjects that had a wide range of professional experience. This controlled study is intended to evaluate different testing methods that are actually used by software devel-

Scrutinize finding



Comparing the Effectiveness of Software Testing Strategies

VICTOR R. BASILI, SENIOR MEMBER, IEEE, AND RICHARD W. SELBY, MEMBER, IEEE

Abstract—This study applies an experimentation methodology to compare three state-of-the-practice software testing techniques: a) code reading by stepwise abstraction, b) functional testing using equivalence partitioning and boundary value analysis, and c) structural testing using 100 percent statement coverage criteria. The study compares the strategies in three aspects of software testing: fault detection effectiveness, fault detection cost, and classes of faults detected. Thirty-two professional programmers and 42 advanced students applied the three techniques to four unit-sized programs in a fractional factorial experimental design. The major results of this study are the following. 1) With the professional programmers, code reading detected more software faults and had a higher fault detection rate than did functional or structural testing, while functional testing detected more faults than did structural testing, but functional and structural testing were not different in fault detection rate. 2) In one advanced student subject group, code reading and functional testing were not different in faults found, but were both superior to structural testing, while in the other advanced student subject group there was no difference among the techniques. 3) With the advanced student subjects, the three techniques were not different in fault detection rate. 4) Number of faults observed, fault detection rate, and total effort in detection depended on the type of software tested. 5) Code reading detected more interface

sub
Thi
ing
ope
sta
T
tiv
wa
tio
pre
niq
wh
ing

There are multiple perspectives from which to view empirical studies of software development techniques, including the study presented in this paper.

- *Experimenter*—An experimenter may view the study as a demonstration of how a software development tech-

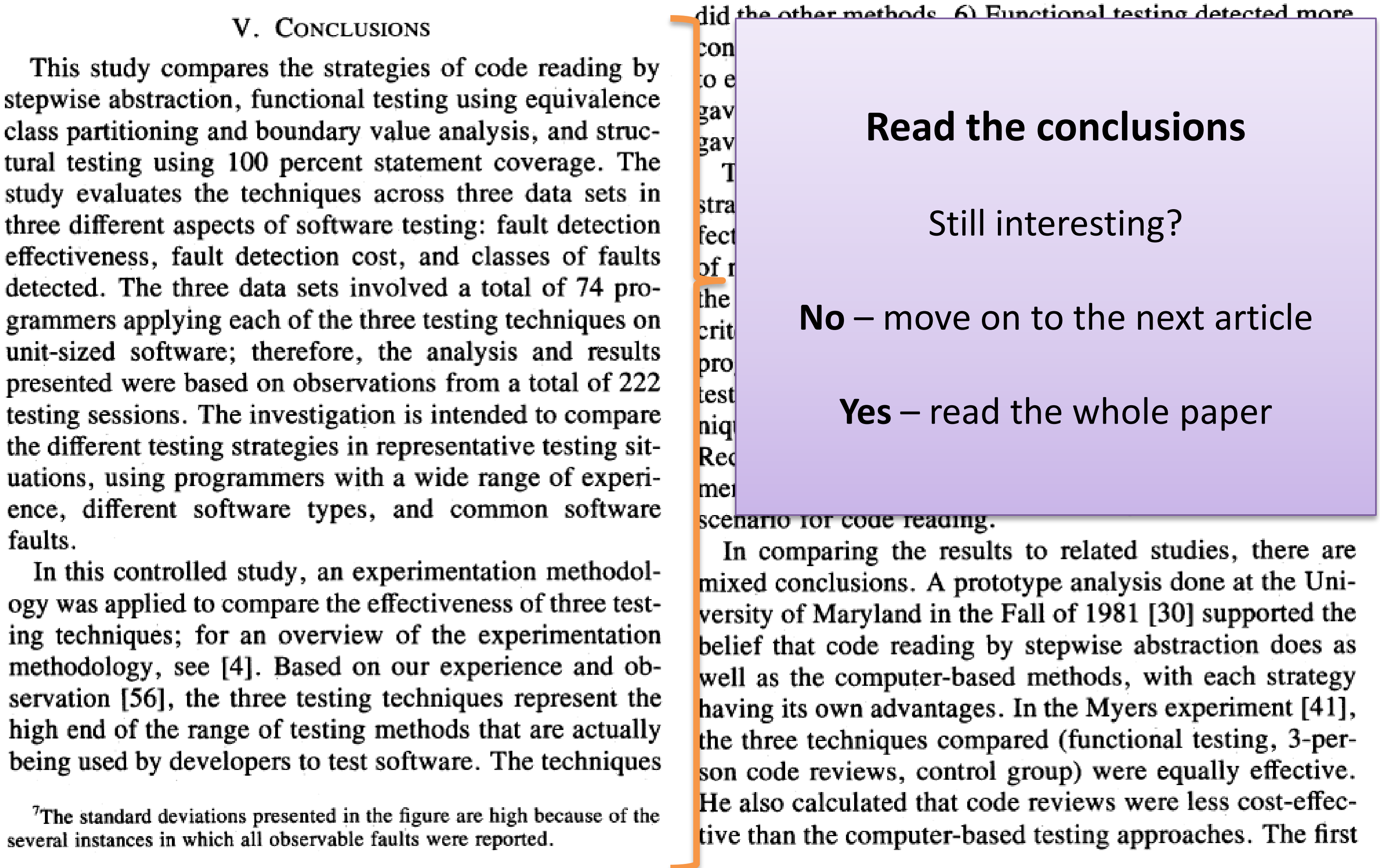
First read the abstract

Does it seem interesting?

No – move on to the next article

Yes – skip to next step

Scrutinize finding



V. Conclusions

This study compares the strategies of code reading by stepwise abstraction, functional testing using equivalence class partitioning and boundary value analysis, and structural testing using 100 percent statement coverage. The study evaluates the techniques across three data sets in three different aspects of software testing: fault detection effectiveness, fault detection cost, and classes of faults detected. The three data sets involved a total of 74 programmers applying each of the three testing techniques on unit-sized software; therefore, the analysis and results presented were based on observations from a total of 222 testing sessions. The investigation is intended to compare the different testing strategies in representative testing situations, using programmers with a wide range of experience, different software types, and common software faults.

In this controlled study, an experimentation methodology was applied to compare the effectiveness of three testing techniques; for an overview of the experimentation methodology, see [4]. Based on our experience and observation [56], the three testing techniques represent the high end of the range of testing methods that are actually being used by developers to test software. The techniques

[7] The standard deviations presented in the figure are high because of the several instances in which all observable faults were reported.

did the other methods. 6) Functional testing detected more
con
to e
gav
gav
T
stra
fect
of r
the
crit
pro
test
niq
Rec
mer
scenario for code reading.

In comparing the results to related studies, there are mixed conclusions. A prototype analysis done at the University of Maryland in the Fall of 1981 [30] supported the belief that code reading by stepwise abstraction does as well as the computer-based methods, with each strategy having its own advantages. In the Myers experiment [41], the three techniques compared (functional testing, 3-person code reviews, control group) were equally effective. He also calculated that code reviews were less cost-effective than the computer-based testing approaches. The first

Read the conclusions

Still interesting?

No – move on to the next article

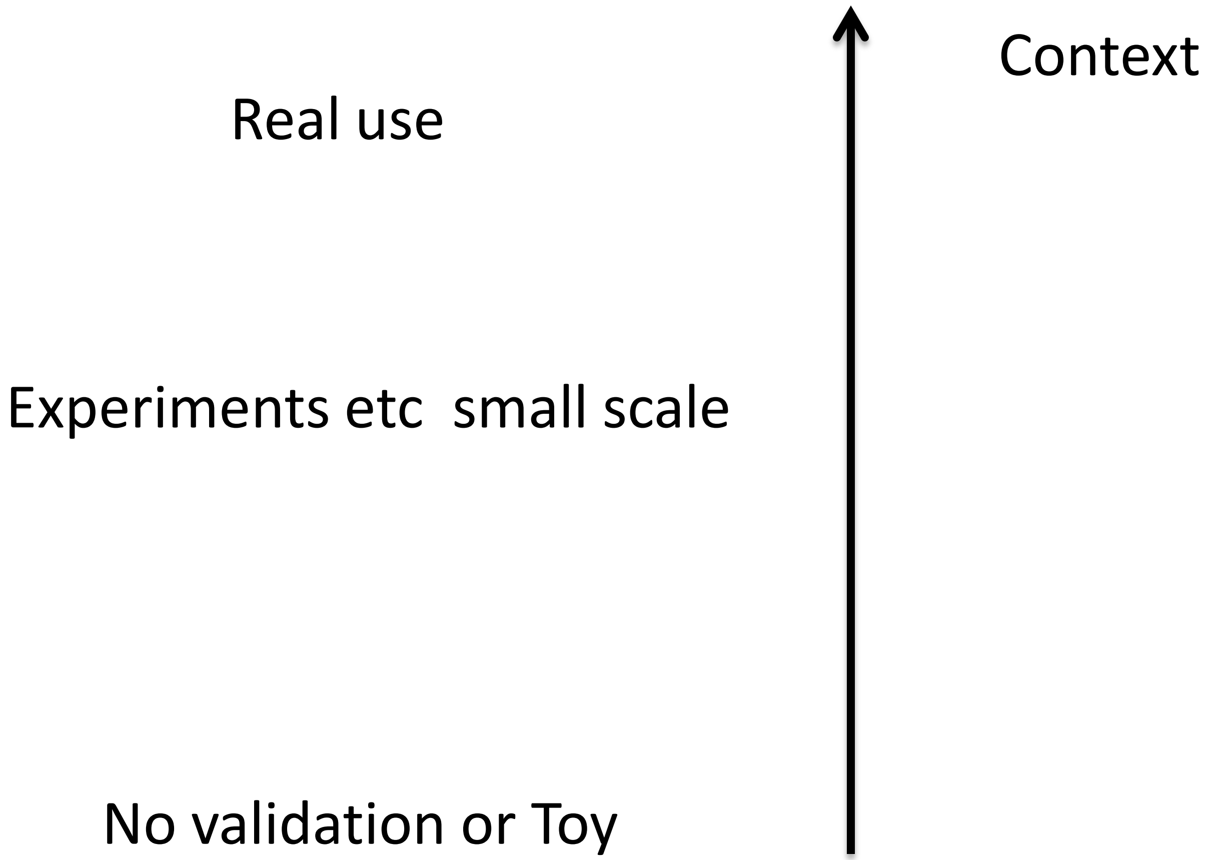
Yes – read the whole paper

Use the finding

- Use it as a reference in your report
  - To strengthen your case
  - We recommend using Perspective based reading as it has been found to be an effective method for finding defects in requirements documents [1].

- Look at the references used in the paper
  - Does any of them seem interesting?
  - Find them

Trustworthiness of evidence



Context

Real use

Experiments etc small scale

No validation or Toy

11. Industry validation

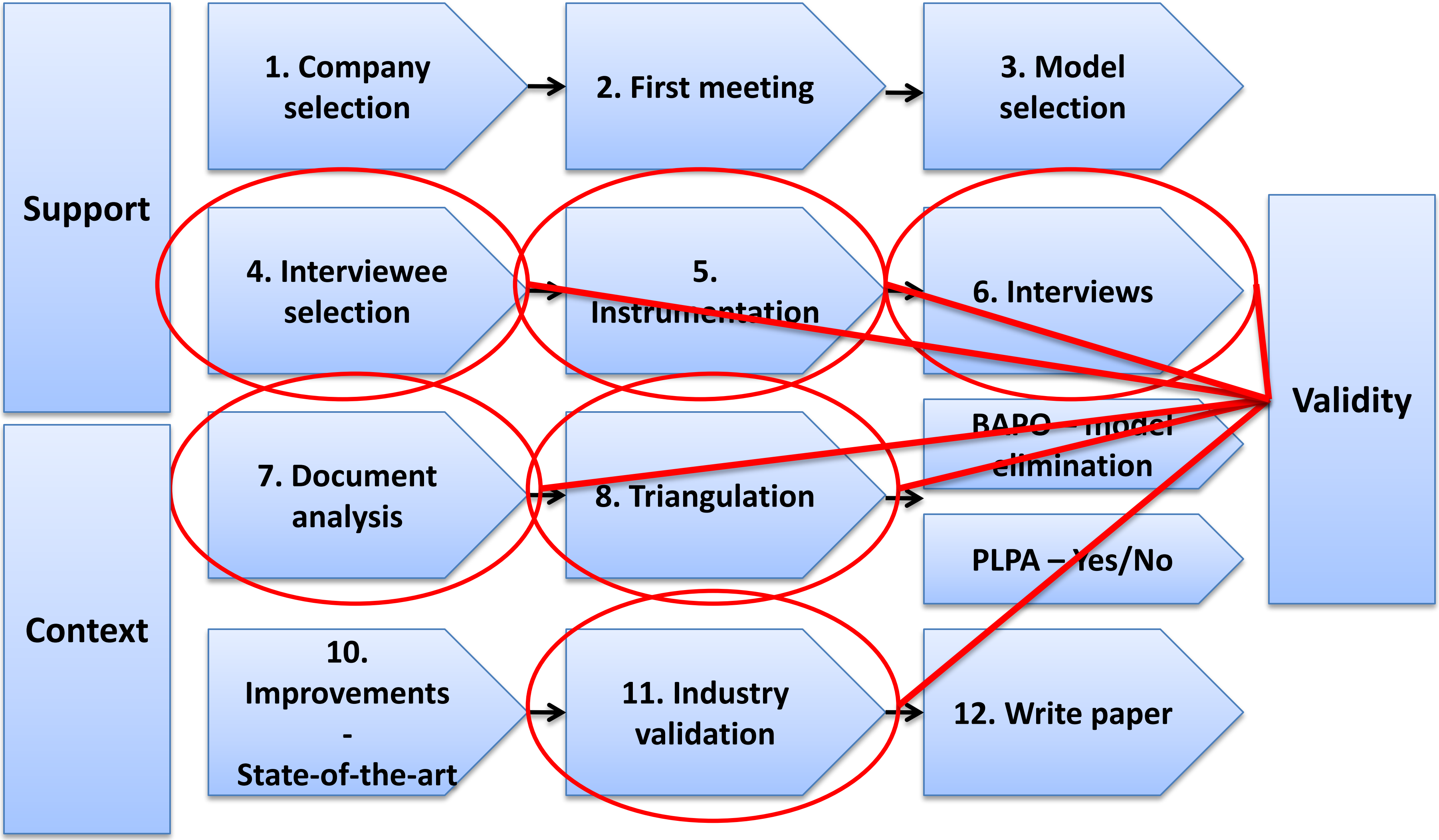
- Present assessment results
  - Based on assessment results and literature
- Present potential solutions
  - Make them understand
  - Make them participate
  - There is nothing wrong if some of your solutions get rejected
    - Document why -> part of report

12. Write paper

Validity

- What is validity – why is it important for you
  - Academic
  - Practical

How validity influence you



Support

1. Company selection

2. First meeting

3. Model selection

4. Interviewee selection

5. Instrumentation

6. Interviews

Validity

BAPO – model elimination

7. Document analysis

8. Triangulation

PLPA – Yes/No

Context

10. Improvements - State-of-the-art

11. Industry validation

12. Write paper

Support

- Problems
- Want more

Problems

- Hard to book interviews
- Champion – your contact
    - Manuscript
        - What we have done and why we are stuck
        - What we need from you

Want more

- Our case study is going really well … but if we could only get [one more interview with …][get access to documentation] … it would add a lot of value and give you better results
- We have now finished our interview study and have interesting results that we would like to come and present to you so you get something back from this case study
  - Assessment results
  - Solutions – basis for discussion

Context