Development and Verification of Rule Based Systems - a Survey of Developers

Valentin Zacharias

FZI

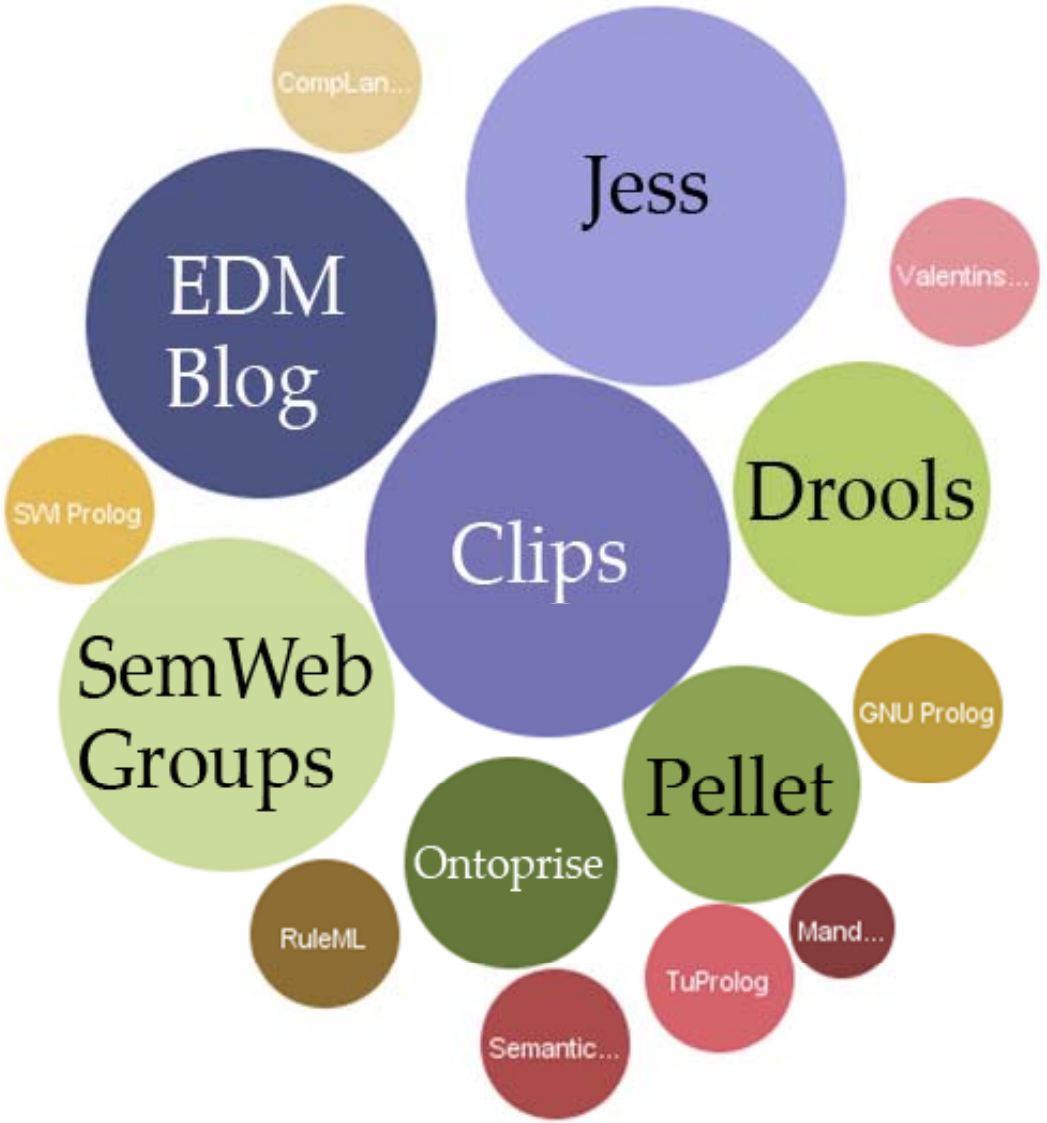
Karlsruhe, Germany

Results from an Online Survey about Rule Base Development

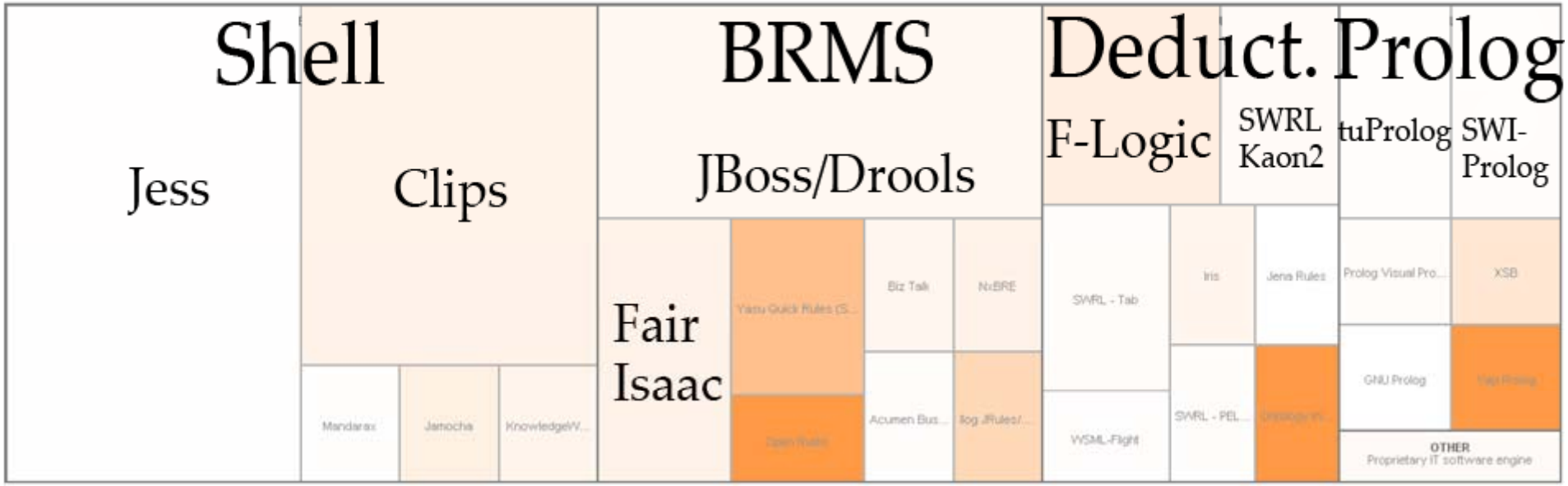
Agenda

- Participants
- Development Methodology
- Tools
  - Development
  - Debugging
  - V&V
- Comparison to OO Development & Issues Hindering Dev.
- Comparison to last surveys (from 18 years ago)
  - Hamilton, D., Kelley, K.: State-of-the-practice in knowledge-based system verication and validation.
  - O'Leary, D. In: Design, Development and Validation of Expert Systems: A Survey of Developers.

76 Participants



Languages and Systems?

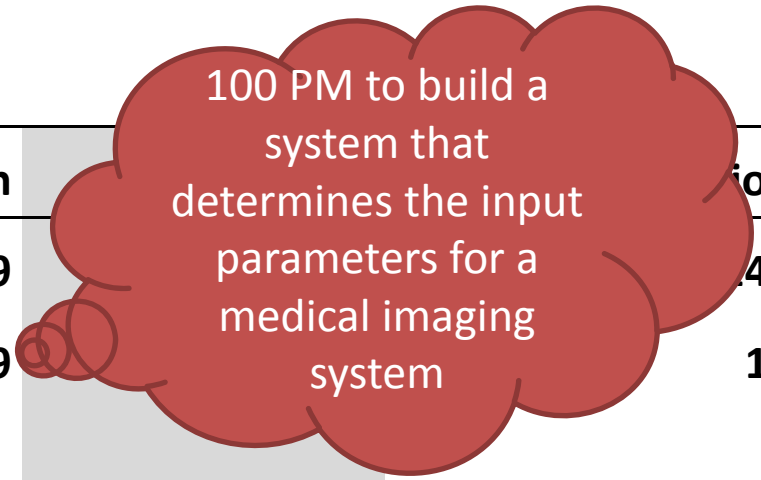


Size

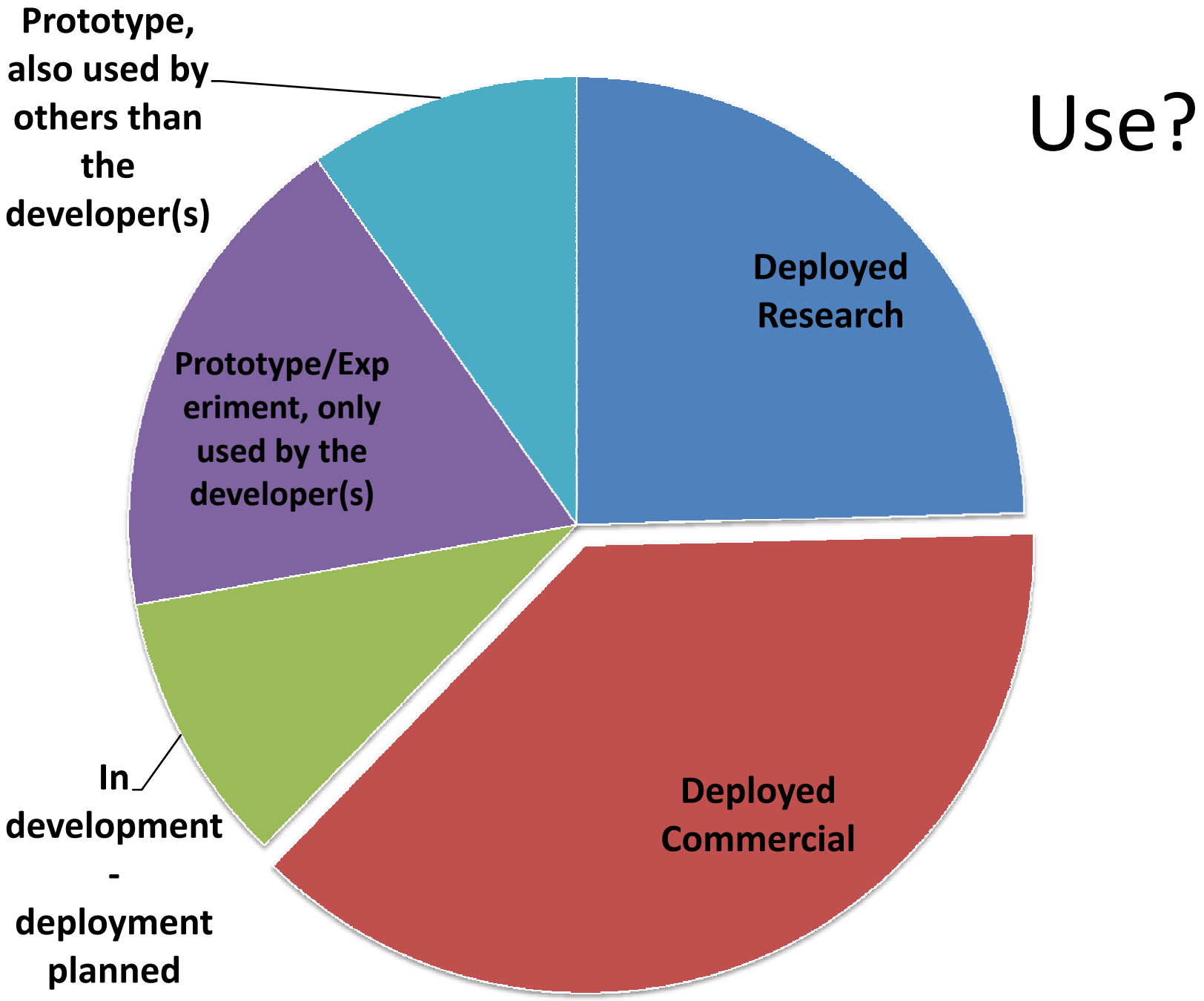
	Mean	Median	Standard Deviation
PM for entire software	**59**	**15**	**148**
PM for rule base	**9**	**5,5**	**15**
Number of rules	**1969**	**120**	**8693**
Size of average rule	**9,3**	**5**	**17**
Size of largest rule	**24**	**11**	**39**
Rule developers	**3**	**2**	**4**
Other software developers	**3**	**1**	**8**
Domain experts that created rules	**1,5**	**1**	**2**
Domain experts as consultants	**1,9**	**1**	**2,5**
Domain experts for V&V	**1,7**	**1**	**2,4**
Others	**0,6**	**0**	**1,6**

Size

	Mean		ion
PM for entire software	59		48
PM for rule base	9		15
Number of rules	1969	120	8693
Size of average rule	9,3		17
Size of largest rule	24		39
Rule developers	3	2	4
Other software developers	3	1	8
Domain experts that created rules	1,5	1	2
Domain experts as consultants	1,9	1	2,5
Domain experts for V&V	1,7	1	2,4
Others	0,6	0	1,6

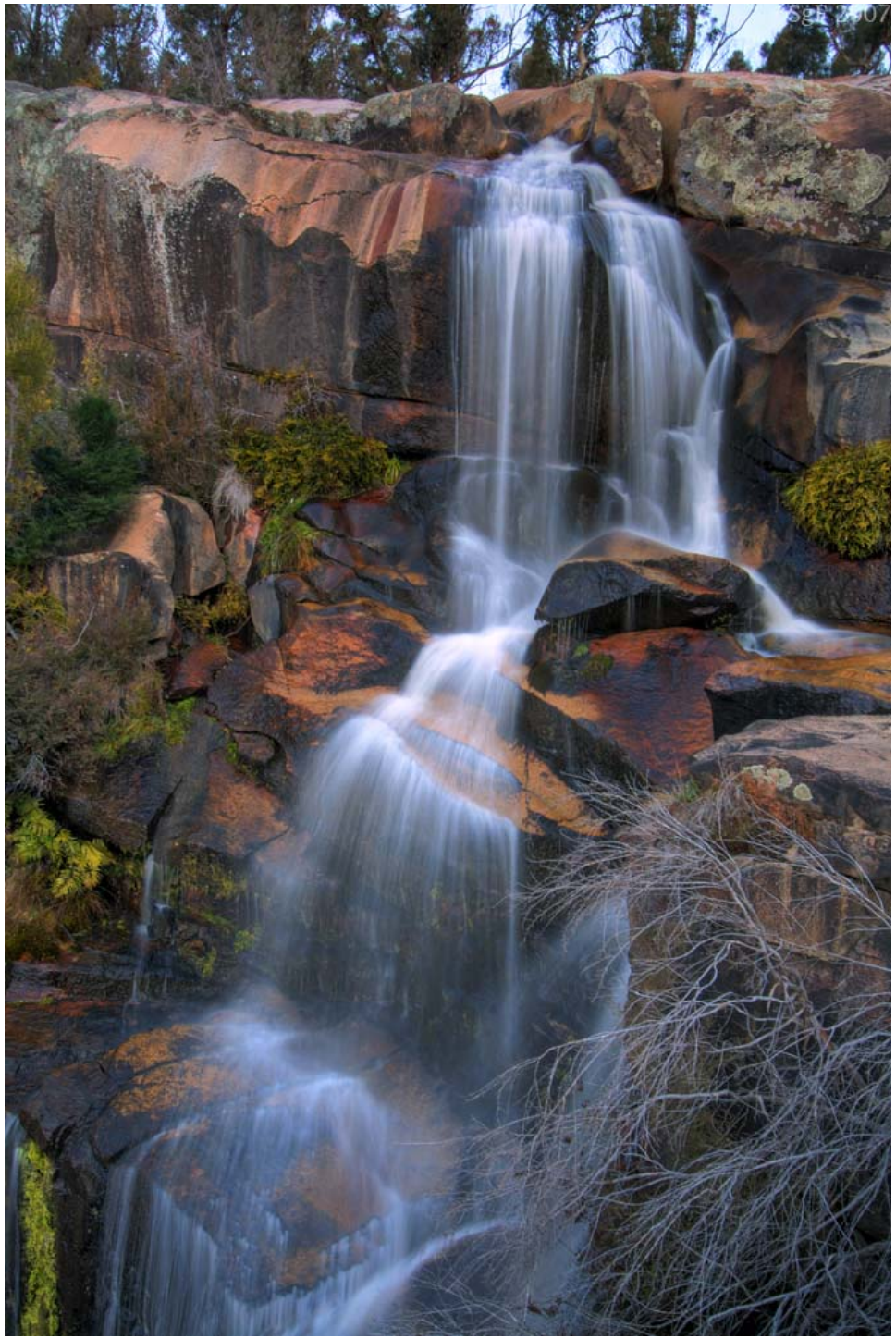


Use?

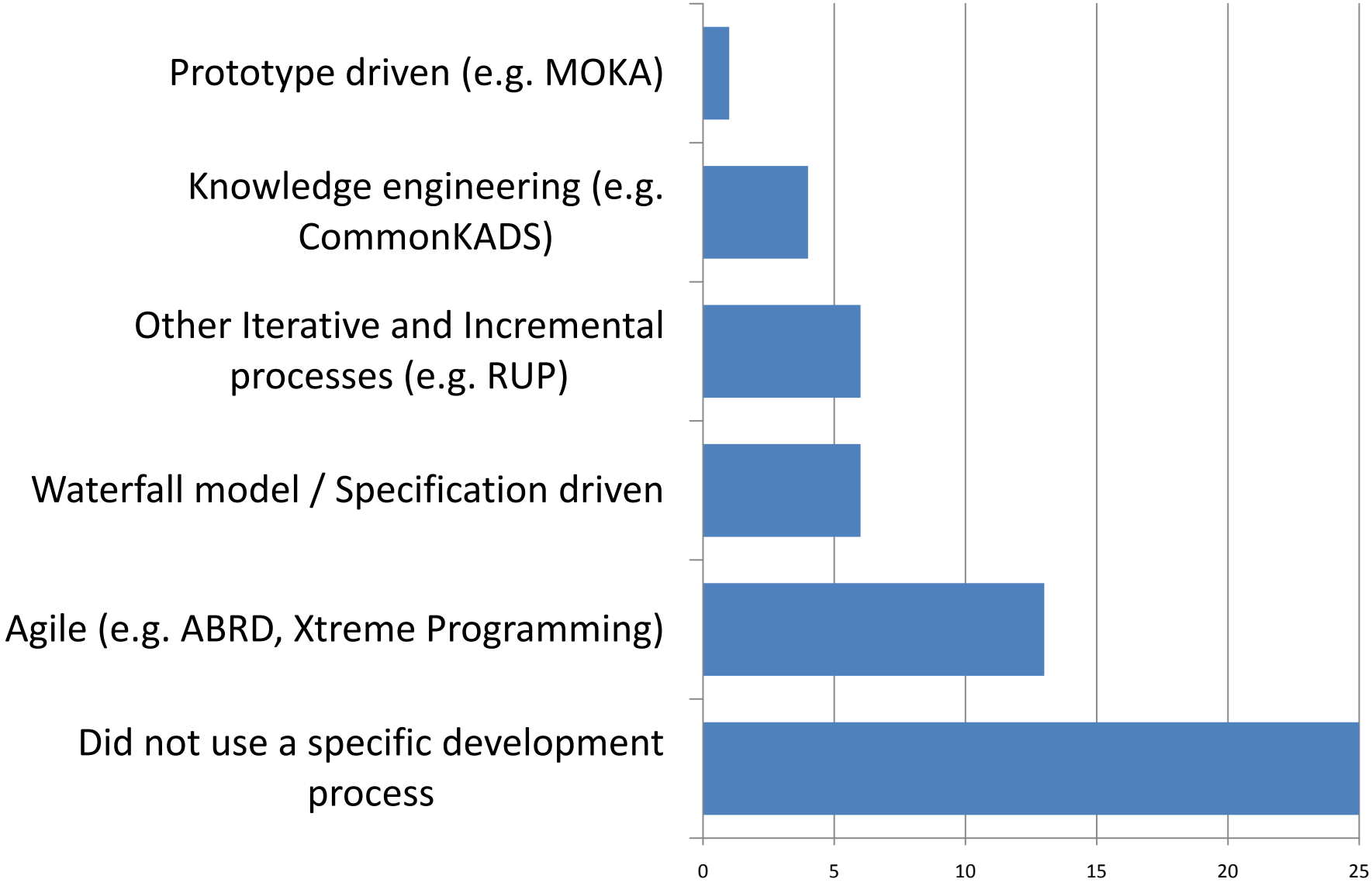

Prototype, also used by others than the developer(s)
Deployed Research
Prototype/Experiment, only used by the developer(s)
In development - deployment planned
Deployed Commercial

Methodologies

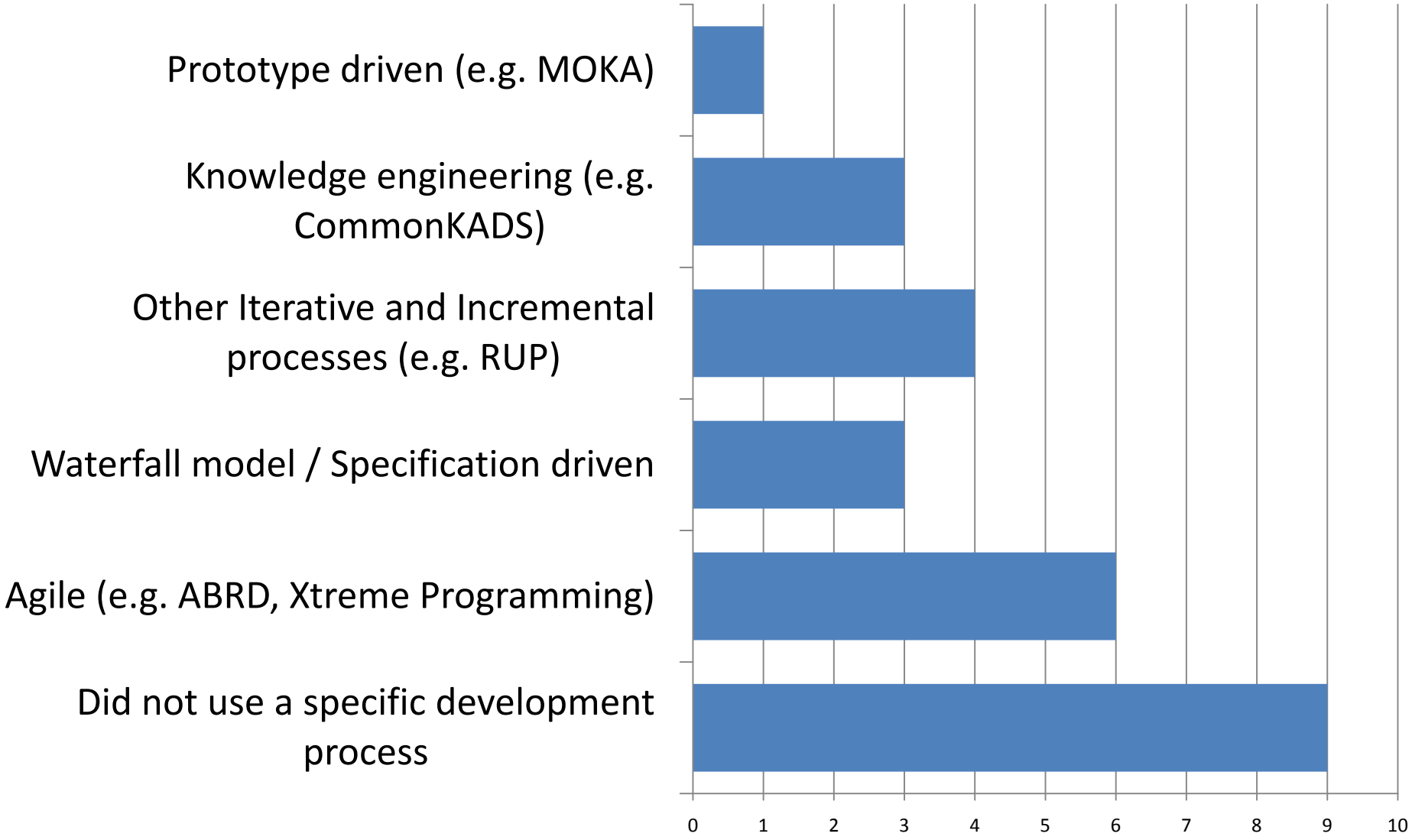
“CommonKADS is *the* methodology to build rule based systems”



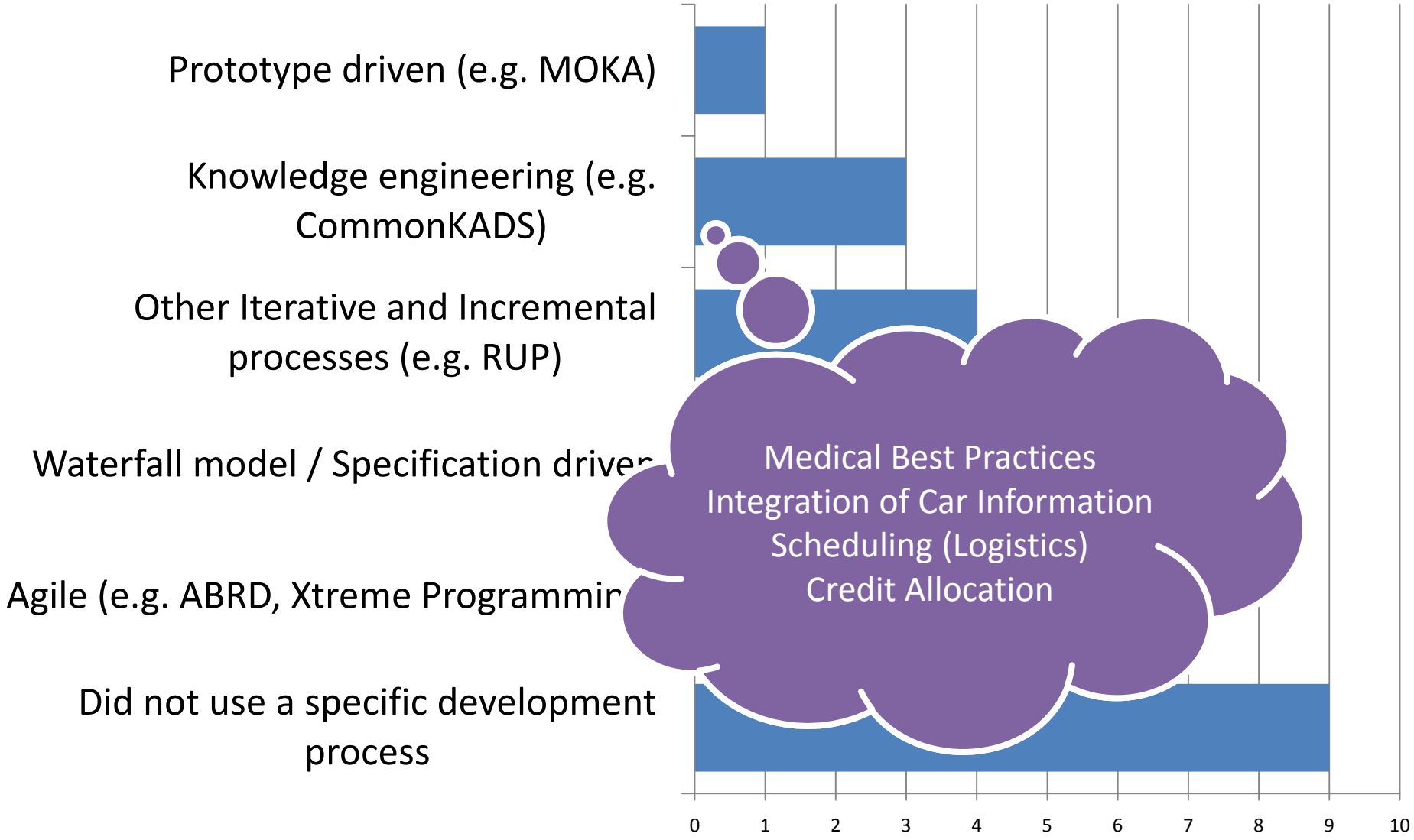
Methodology



Methodology – only 10PM+



Methodology – only 10PM+



Methodology – only 10PM+

Prototype driven (e.g. MOKA)

Knowledge engineering (e.g. CommonKADS)

Other Iterative and Incremental processes (e.g. RUP)

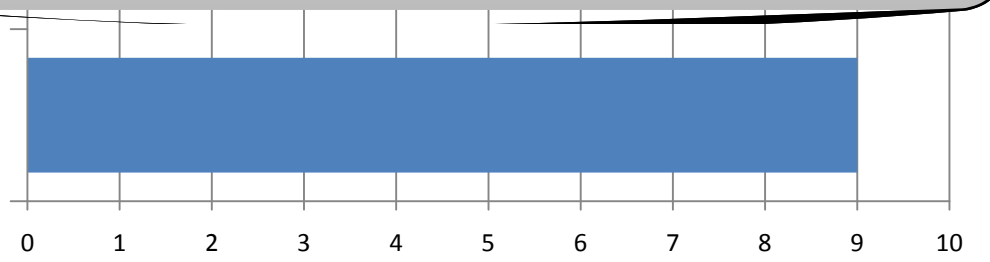
Waterfall model / Specification driven

Agile (e.g. ABRD, Xtreme Programm

Did not use a specific development process

18 Years Ago:

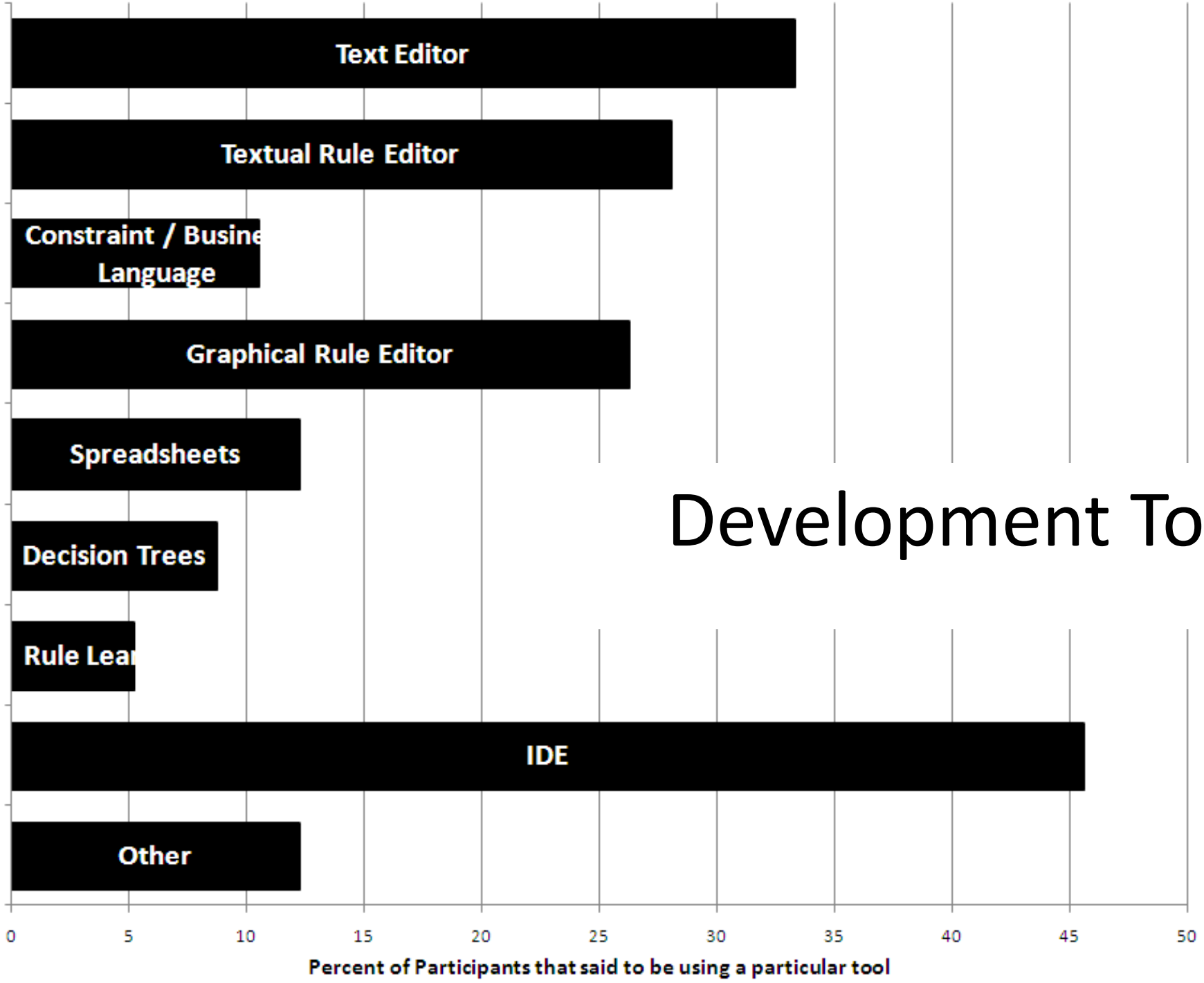
"The most frequent (40%) life cycle model used was the cyclic model [...]. However 22% of the respondents stated that no model was followed"



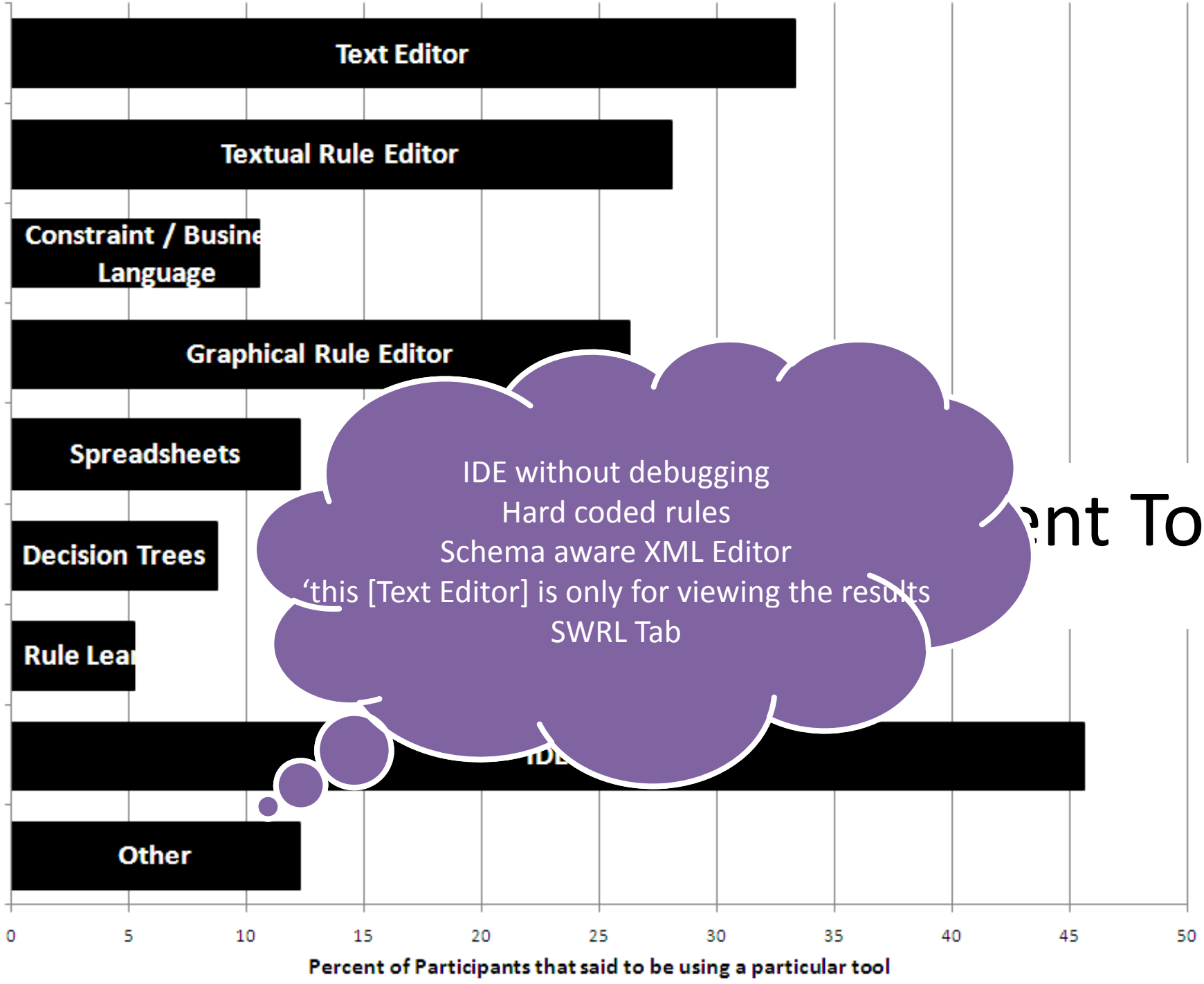
Tools used for Development

"I doubt that manually created rule bases will serve as the basis for the semantic web or other application areas"

Development Tools



ent Tools


Text Editor
Textual Rule Editor
Constraint / Busine
Language
Graphical Rule Editor
Spreadsheets
Decision Trees
Rule Lear
Other
IDE without debugging
Hard coded rules
Schema aware XML Editor
'this [Text Editor] is only for viewing the results
SWRL Tab
0
5
10
15
20
25
30
35
40
45
50
Percent of Participants that said to be using a particular tool

Verification & Debugging

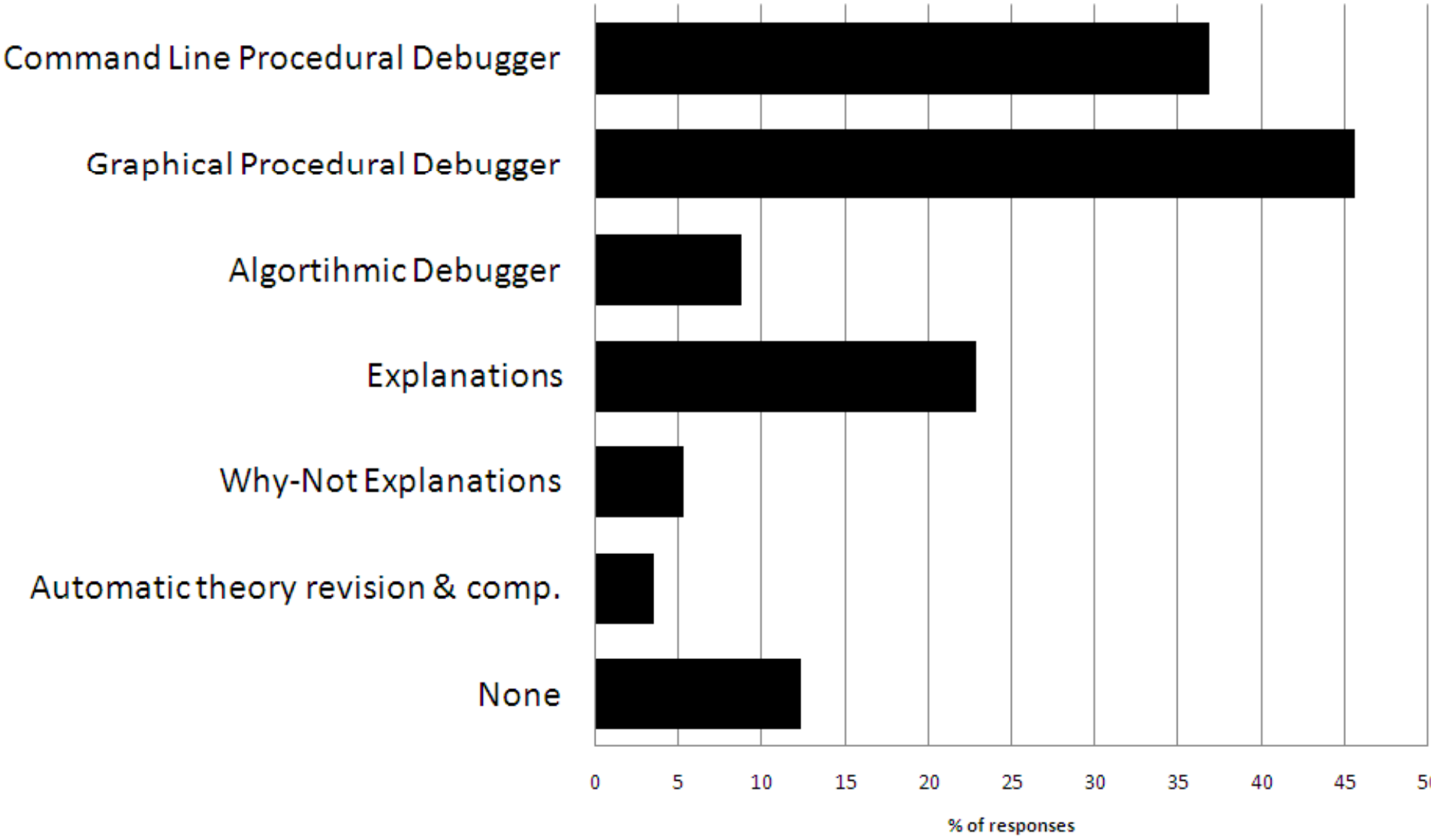
“The correctness of rule bases is ensured with formal verification”

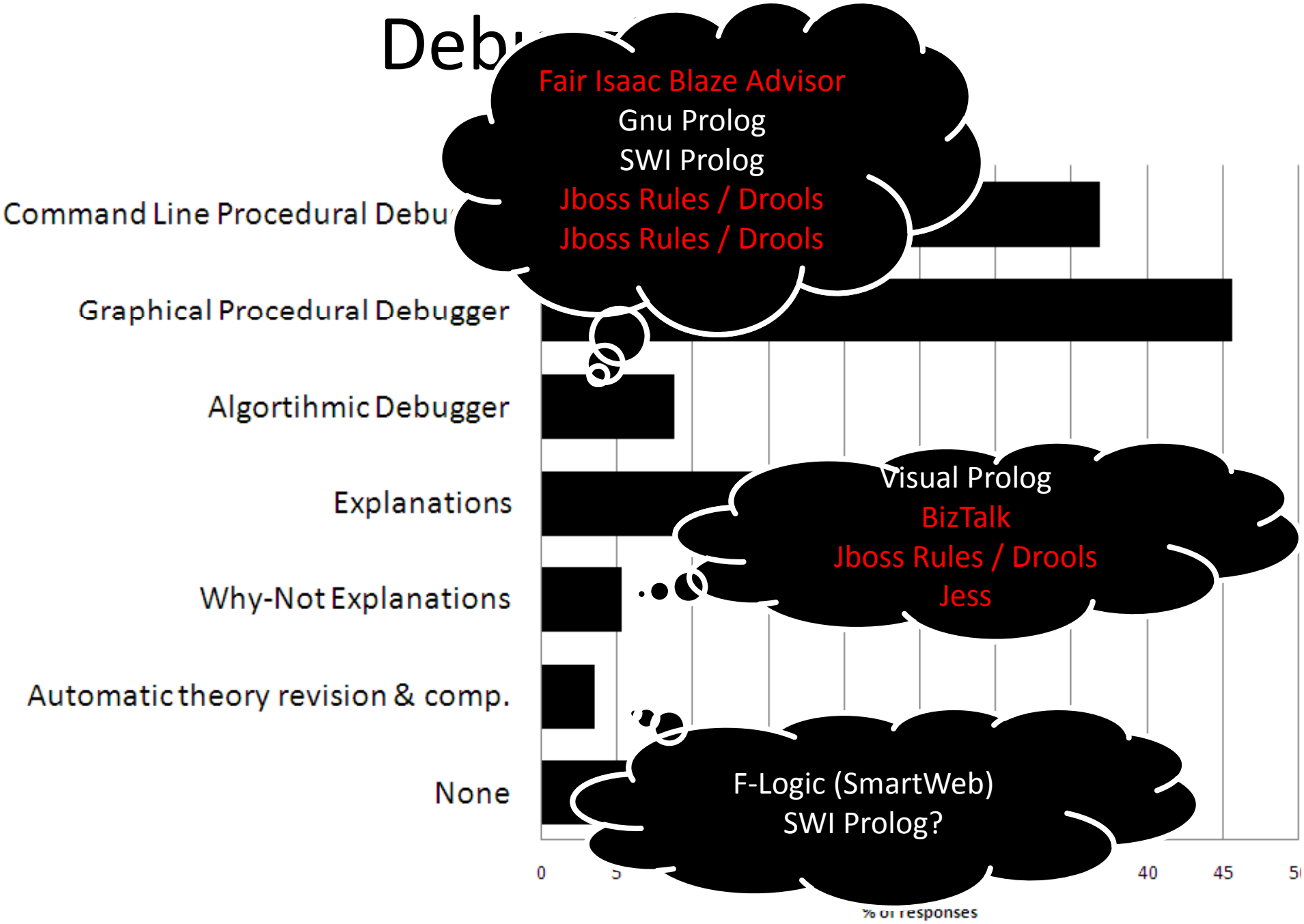
Debugging Paradigms for Rule Based Systems

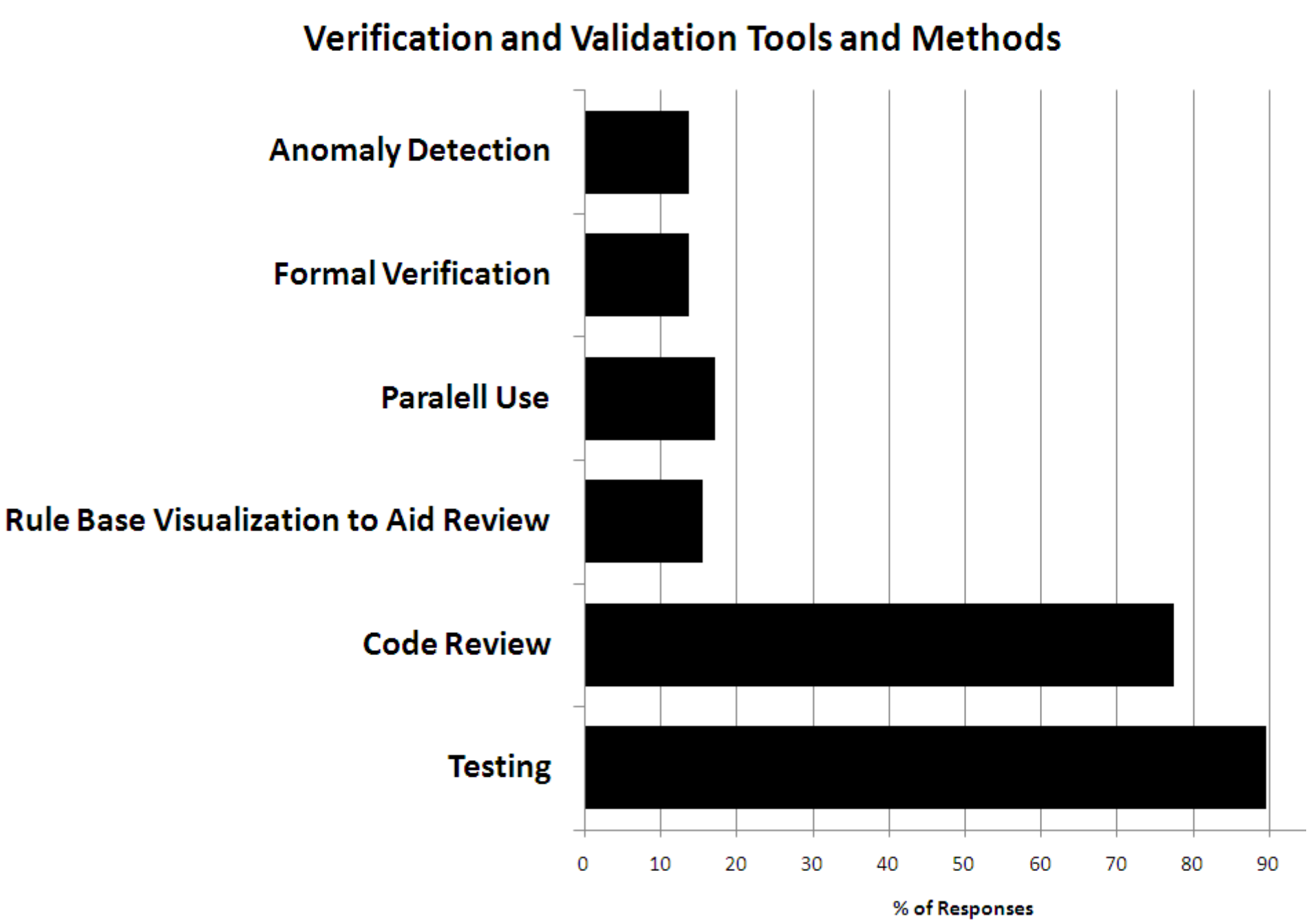
- Procedural Debugging
  - Stepwise execution of inference engine
- Algorithmic Debugging
  - System identifies fault by asking user (oracle) about correctness of results of subcomputations
- Explanations
  - Concise NL or graphical representation of justification for a result
- Why-Not Explanations
  - Explanation also for missing conclusions
- Automatic Theory Revision
  - Automatic correction of rule base

Valentin Zacharias: **The Debugging of Rule Bases, to appear in** Handbook of Research on Emerging Rule-Based Languages and Technologies, IGI Global, Hershey (USA) 2009.

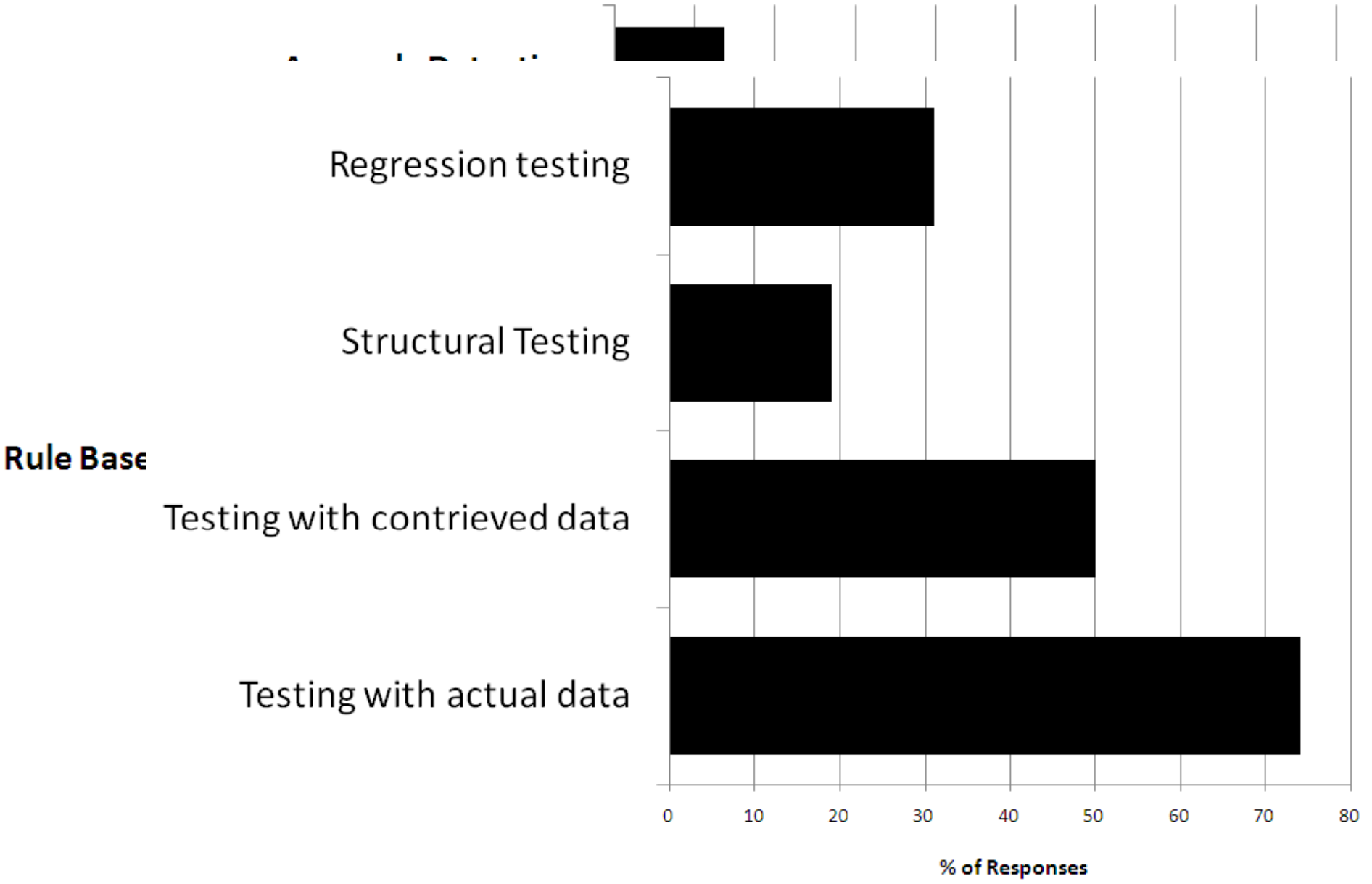
Debugging Tools




Deb
Fair Isaac Blaze Advisor
Gnu Prolog
SWI Prolog
Jboss Rules / Drools
Jboss Rules / Drools
Command Line Procedural Debu
Graphical Procedural Debugger
Algortihmic Debugger
Explanations
Why-Not Explanations
Automatic theory revision & comp.
None
Visual Prolog
BizTalk
Jboss Rules / Drools
Jess
F-Logic (SmartWeb)
SWI Prolog?
0
40
45
% of responses


Verification and Validation Tools and Methods
Anomaly Detection
Formal Verification
Paralell Use
Rule Base Visualization to Aid Review
Code Review
Testing
0
10
20
30
40
50
60
70
80
90
% of Responses

Verification and Validation Tools and Methods



Comparison To ‘Conventional Programming’ and Issues Hindering Development

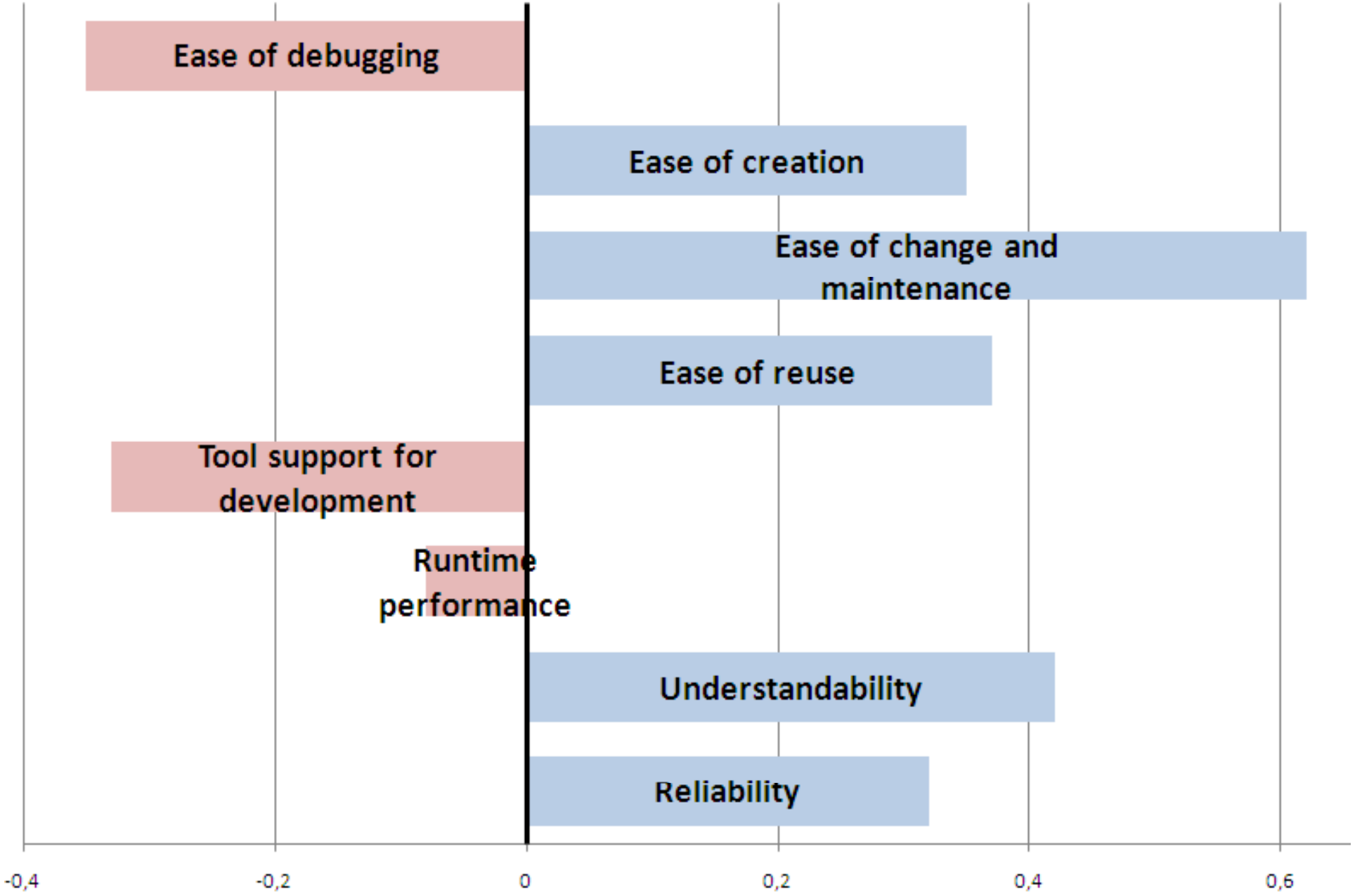
“Runtime Performance and Expressivity are the problems hindering the development of rule based systems”

Comparison – Question

- How does the rule base and its development process compare to a ‘conventional’ program (created with procedural/object oriented languages) of similar size?
  1. Ease of change and maintenance
     Rule base superior; Comparable; Conventional program superior; Don’t know
  2. [...]

Computer average with:
* Rule base superior = 1
* Comparable = 0
* Conventional program sup= -1
* Ignore ‘Don’t know’

Comparison


Ease of debugging
Ease of creation
Ease of change and maintenance
Ease of reuse
Tool support for development
Runtime performance
Understandability
Reliability
-0,4
-0,2
0
0,2
0,4
0,6

Issues Hindering Development – Question

- What were the most important issues in the development of the rule base?
  1. Rule expressivity – could not (easily) represent what was needed
     Not an issue; Annoyance; Hindered development
  2. [...]

Computer average with:
* Not an issue = 0
* Annoyance = 1
* Hindered development = 2

Issues Hindering Development

	Average	Not an Issue	Annoyance	Hinderance
Debugging	1	12	28	12
Determining completeness	0,76	18	27	6
Supporting tools missing/immature	0,67	26	17	9
Editing of rules	0,66	24	23	6
Determining test coverage	0,65	25	19	7
Inexperienced developers	0,58	31	13	9
Rule expressivity	0,5	33	12	7
Keeping rules base up to date	0,5	30	19	4
Understanding the rule base	0,47	31	19	3
Runtime performance	0,41	35	14	4
Organizing collaboration	0,41	35	14	4

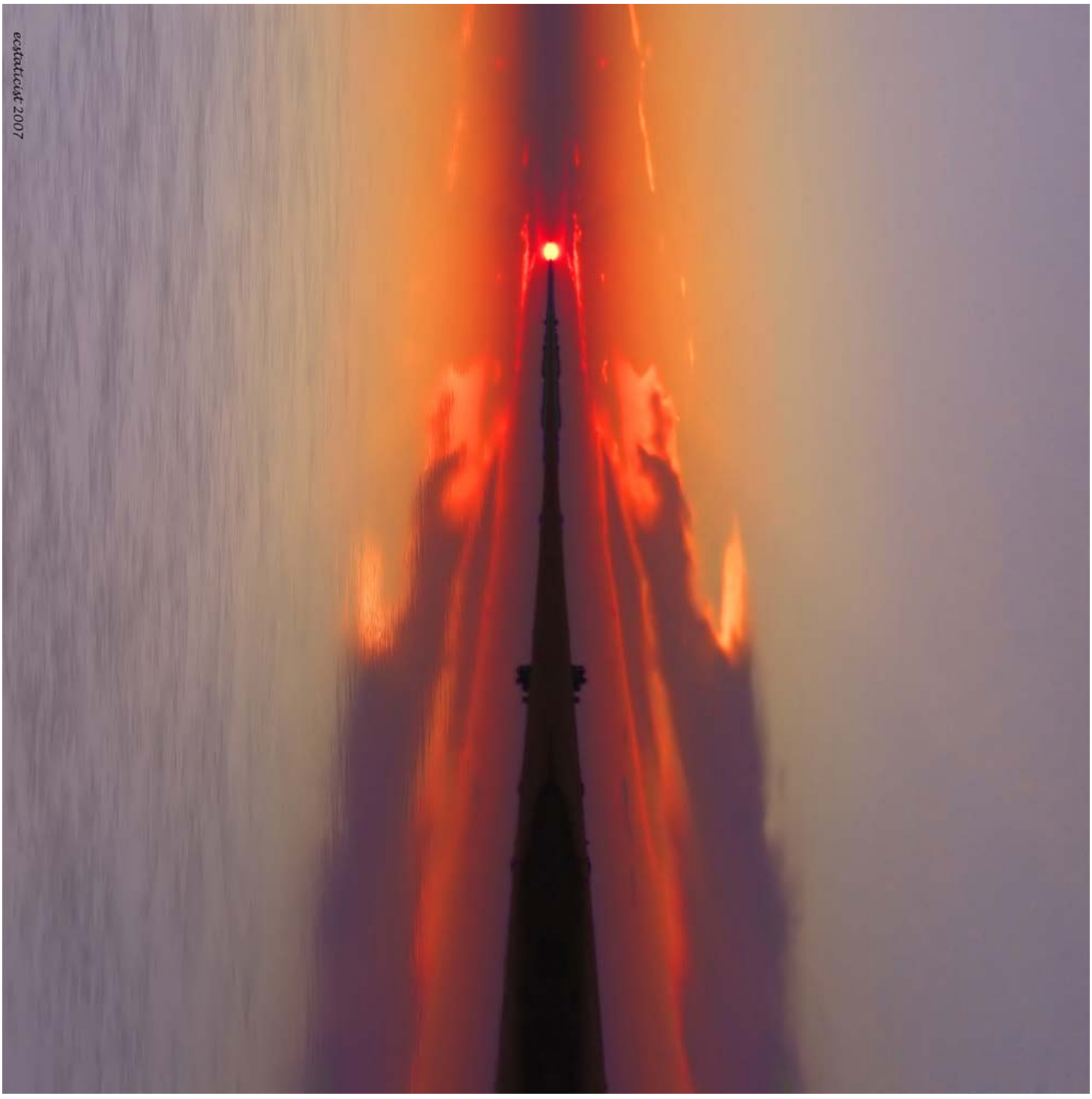
Issues Hindering Development

	erance
Debugging	12
Determining c	6
Supporting too	9
Editing of rules	6
Determining te	7
Inexperienced	9
Rule expressiv	7
Keeping rules	4
Understanding	3
Runtime perfo	4
Organizing col	4

18 Years Ago:

1. **Completeness of Knowledge Base**
2. **Correctness of Knowledge Base**
3. **System does not present possible opportunities to the user**
4. **System is hard to use**
5. **Results difficult to interpret**
6. **Systems presents incorrect opportunities to the user**
7. **Difficult to sequence rules correctly**

Conclusions



Meta Conclusion

„This paper is not a (normal) scientific paper, it summarizes a survey [...]“

- Little empirical data about the challenges facing actual rule base developers
- Little interest in the academic rule community at identifying and tackling practical problems?

Conclusion

- Little academic interest in relevant (in particular agile) rule base development methodologies
- Debugging and finding faults as *the challenge*
- Tool support found wanting, possible motivation for rule interchange
- still most rules created manually, text editors widespread

Thanks for your Attention

Valentin Zacharias
zach@fzi.de
http://vzach.de

Attribution

- Pool Rules by Joe Shlabotnik on Flickr
- Waterfall by Sachman75 on Flickr
- Tools by docman on Flickr
- Bug by Chewy Chua on Flickr
- Obstacle by Guillaume Lemoine on Flickr
- Sun Conclusion by ecstaticist on Flickr
- End by bondywhat on Flickr