AI for Testing and Testing of AI:

New Research Results (on AI) in Software Engineering

Science! by Infotiv, Gothenburg, 20180516

Robert Feldt

Professor of Software Engineering

Chalmers University and Blekinge Inst of Tech

robert.feldt@chalmers.se

robert.feldt@gmail.com

Who am I?

Professor of Software Engineering (SE) in Sweden. Research is focused on Software Quality/Testing, Human factors in SE, and Applying AI.

Programmer since 38 years & consultant since 26. Sold my first program at age 13.

While doing research in academia I have worked with Tech and Software companies to apply AI to improve Software Engineering.

Main message

AI can be applied in many different ways in Software Engineering (SE)

AI-for-Testing != Testing-of-AI

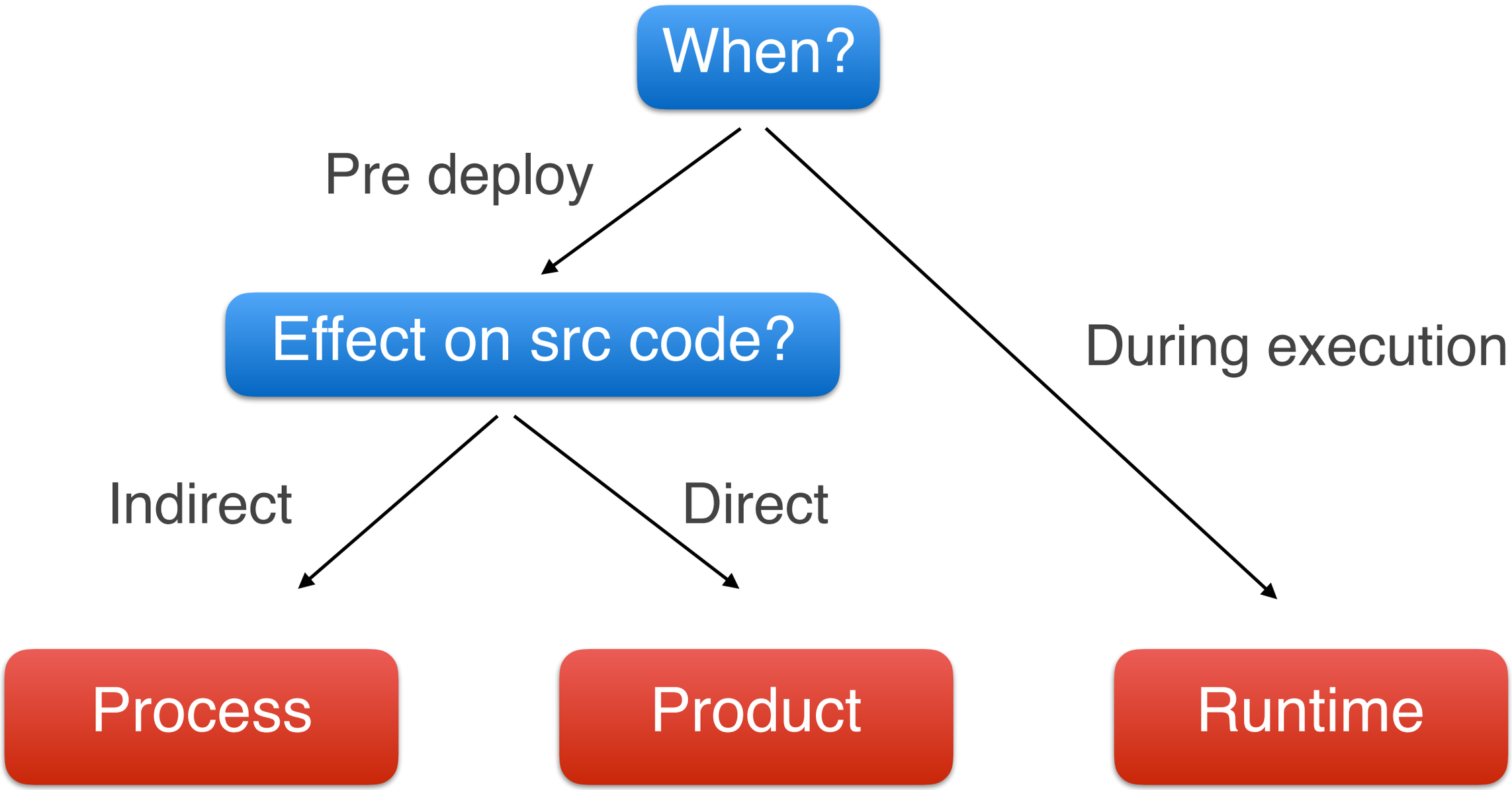
Simple model of AI-in-SE help in analysis/strategy

There is a lot of cool, new Testing & SE Research! :)

AI-SEAL Taxonomy: AI-in-SE Application Levels

- **Point** of AI application?
  - Determines how big an impact the AI and amount of control developers have on SW behaviour.
- **Type of AI** technology?
  - 5 main tribes + supporting technologies
- AI **Automation Level**?
  - From 1 (manual) to 10 (autonomous AI)
- Other and more detailed dimensions, for example:
  - **Shape** of artefact/software?
    - Traditional (Source code or Binary) or AI-specific (ANN)

Point of Application?



Type of AI technology?

So what is AI then?

Moving target definition of AI:

“How to make computers do things which, at the moment, people do better”

— Elaine Rich & Kevin Knight

Type of AI technology?

The Five Tribes of Machine Learning

Tribe	Origins	Master Algorithm
Symbolists	Logic, philosophy	Inverse deduction
Connectionists	Neuroscience	Backpropagation
Evolutionaries	Evolutionary biology	Genetic programming
Bayesians	Statistics	Probabilistic inference
Analogizers	Psychology	Kernel machines

[Domingos2015 “The Master Algorithm”]

Supporting technologies:

Advanced Statistics + Search/Optimisation

AI Automation level?

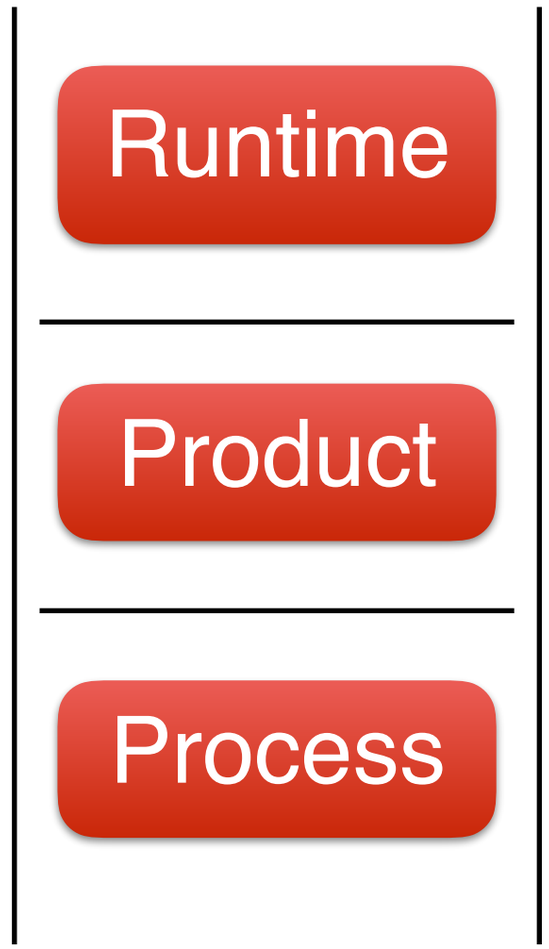
TABLE I
LEVELS OF AUTOMATION OF DECISION AND ACTION SELECTION

HIGH	10. The computer decides everything, acts autonomously, ignoring the human.
	9. informs the human only if it, the computer, decides to
	8. informs the human only if asked, or
	7. executes automatically, then necessarily informs the human, and
	6. allows the human a restricted time to veto before automatic execution, or
	5. executes that suggestion if the human approves, or
	4. suggests one alternative
	3. narrows the selection down to a few, or
	2. The computer offers a complete set of decision/action alternatives, or
LOW	1. The computer offers no assistance: human must take all decisions and actions.

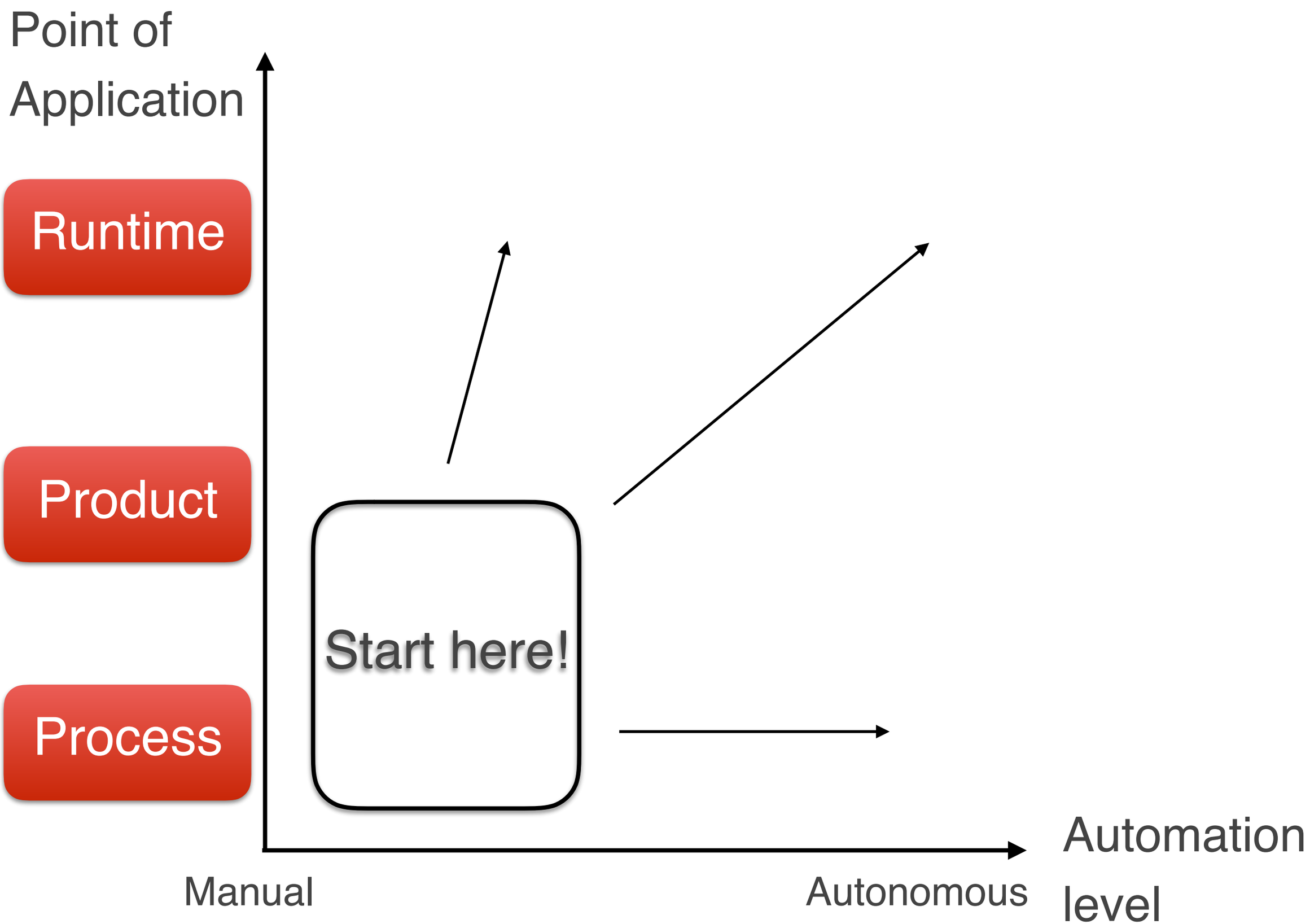
Sheridan1980 from [Frohm2008]

AI-in-SE applications have different levels of risk

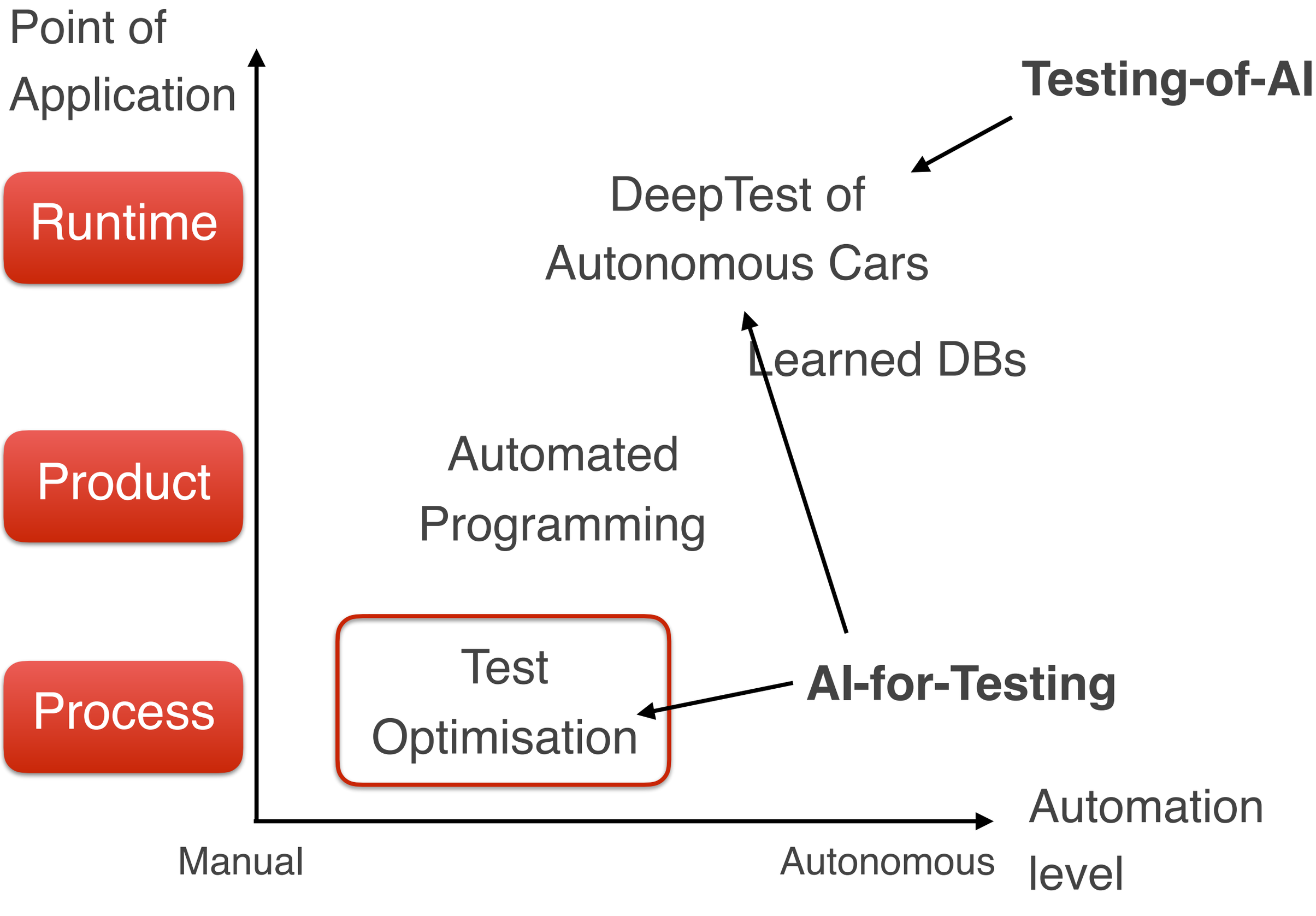
- A ladder of increasing risk:
  - Product more risky than Process
  - Runtime more risky than Product
- Higher levels of automation have higher levels of risk
  - Less time to “reverse” decisions
- Thus:
  - If an AI technology is new to your company, start at low level of automation & at a “lower” point of application.
  - Build more experience then expand “out and up”

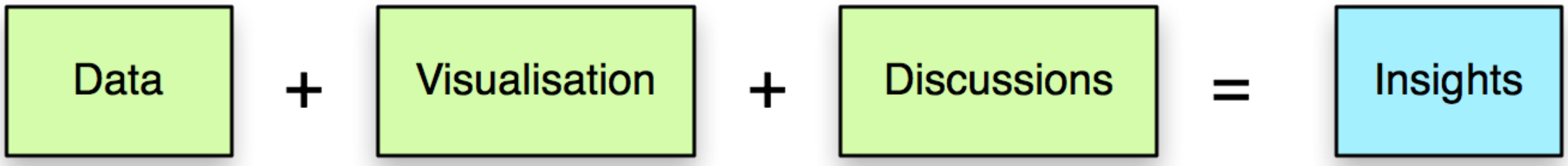


AI-in-SE applications have different levels of risk/gain



AI-in-SE applications have different levels of risk/gain

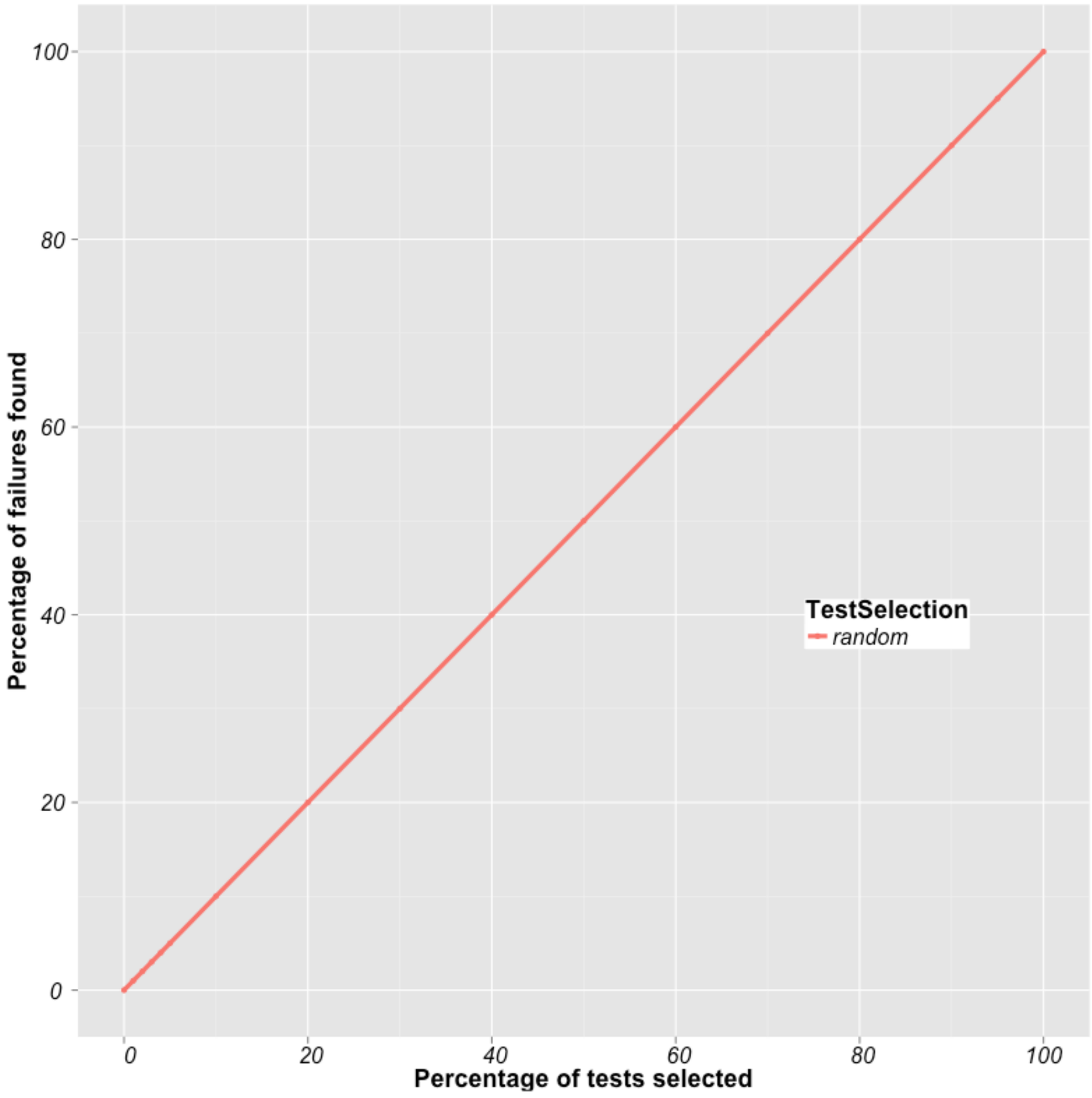



Data
+
Visualisation
+
Discussions
=
Insights

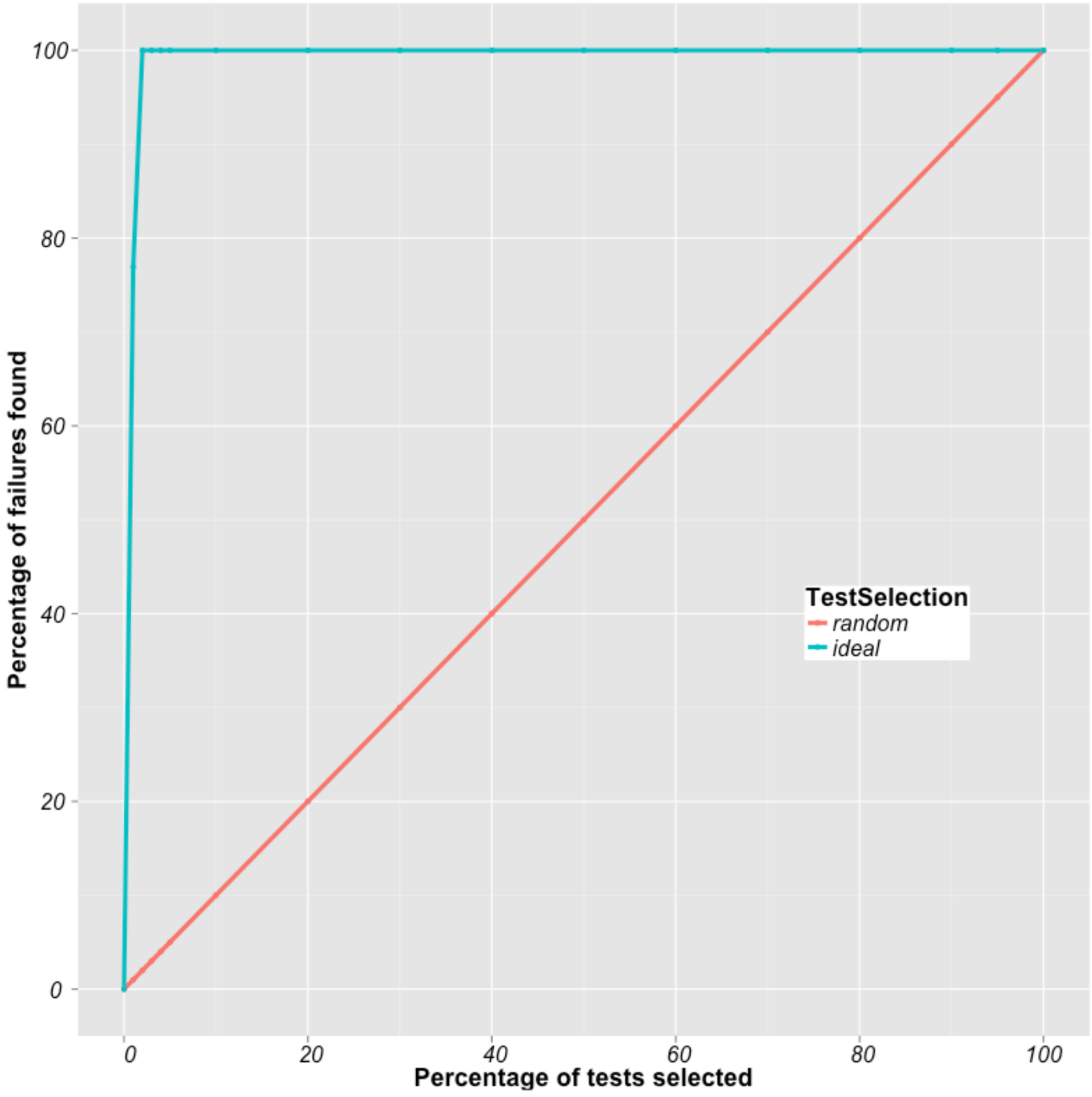
Model

Model++

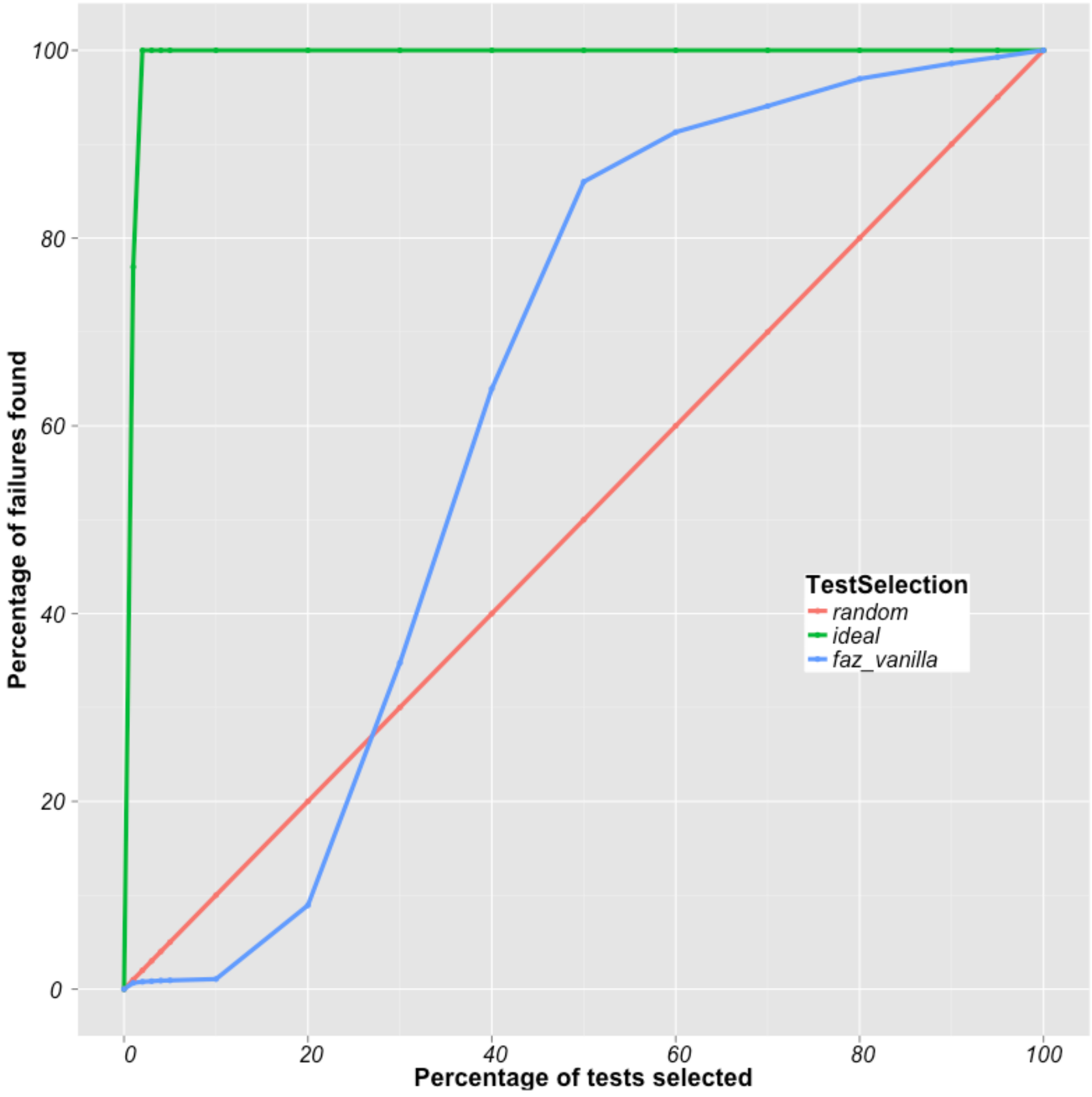
Testing only what is likely to fail



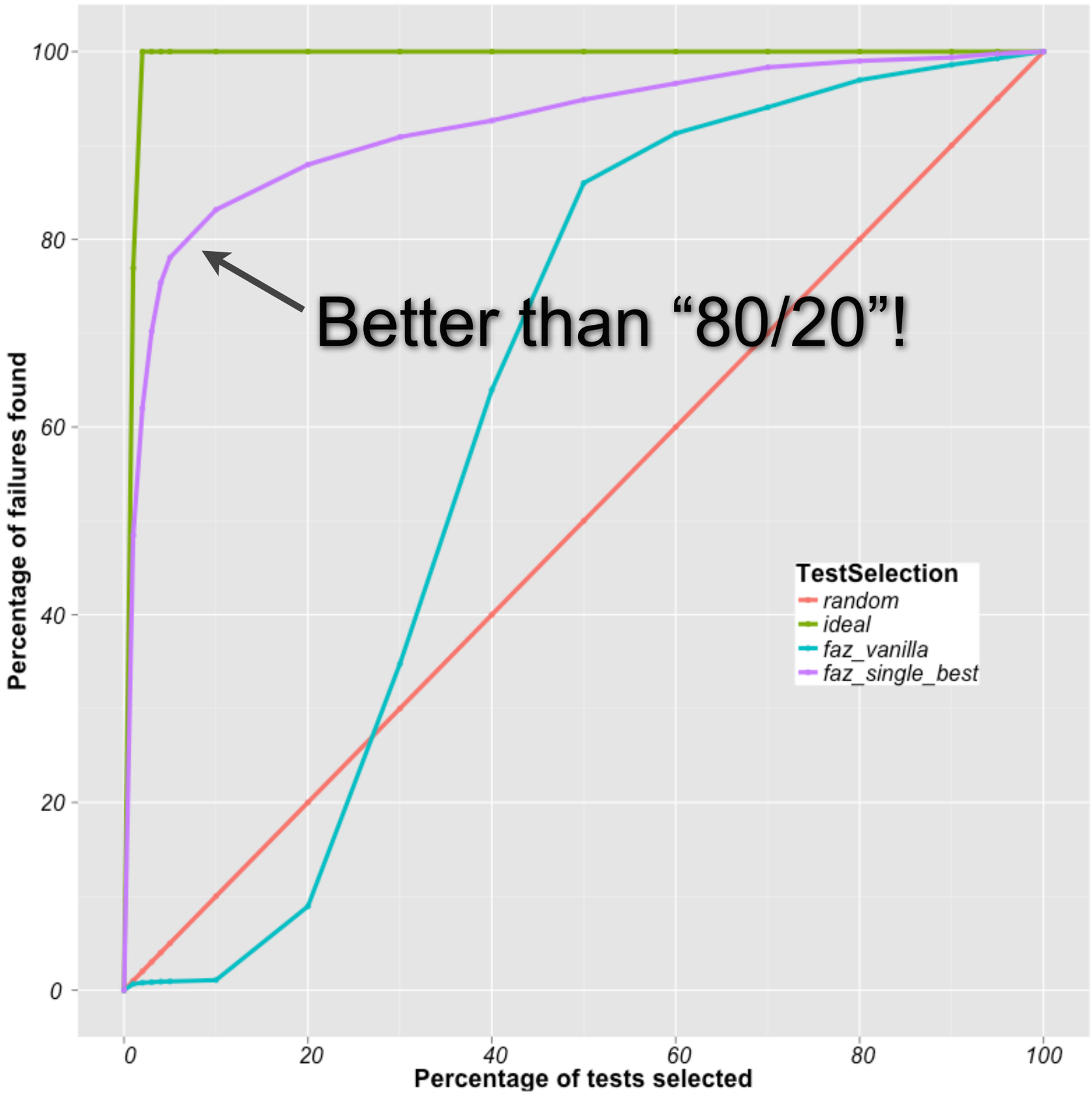
Testing only what is likely to fail



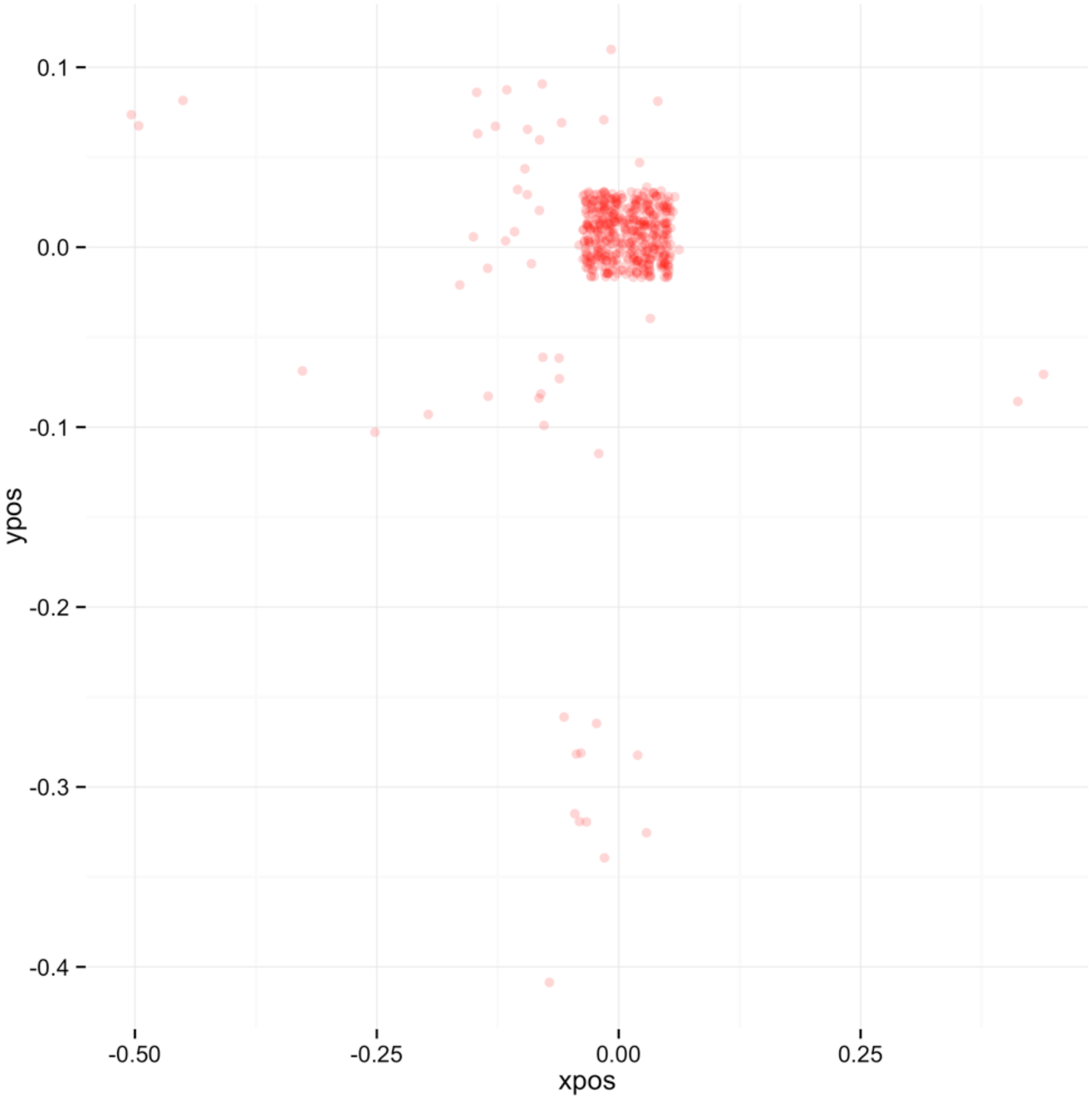
Testing only what is likely to fail



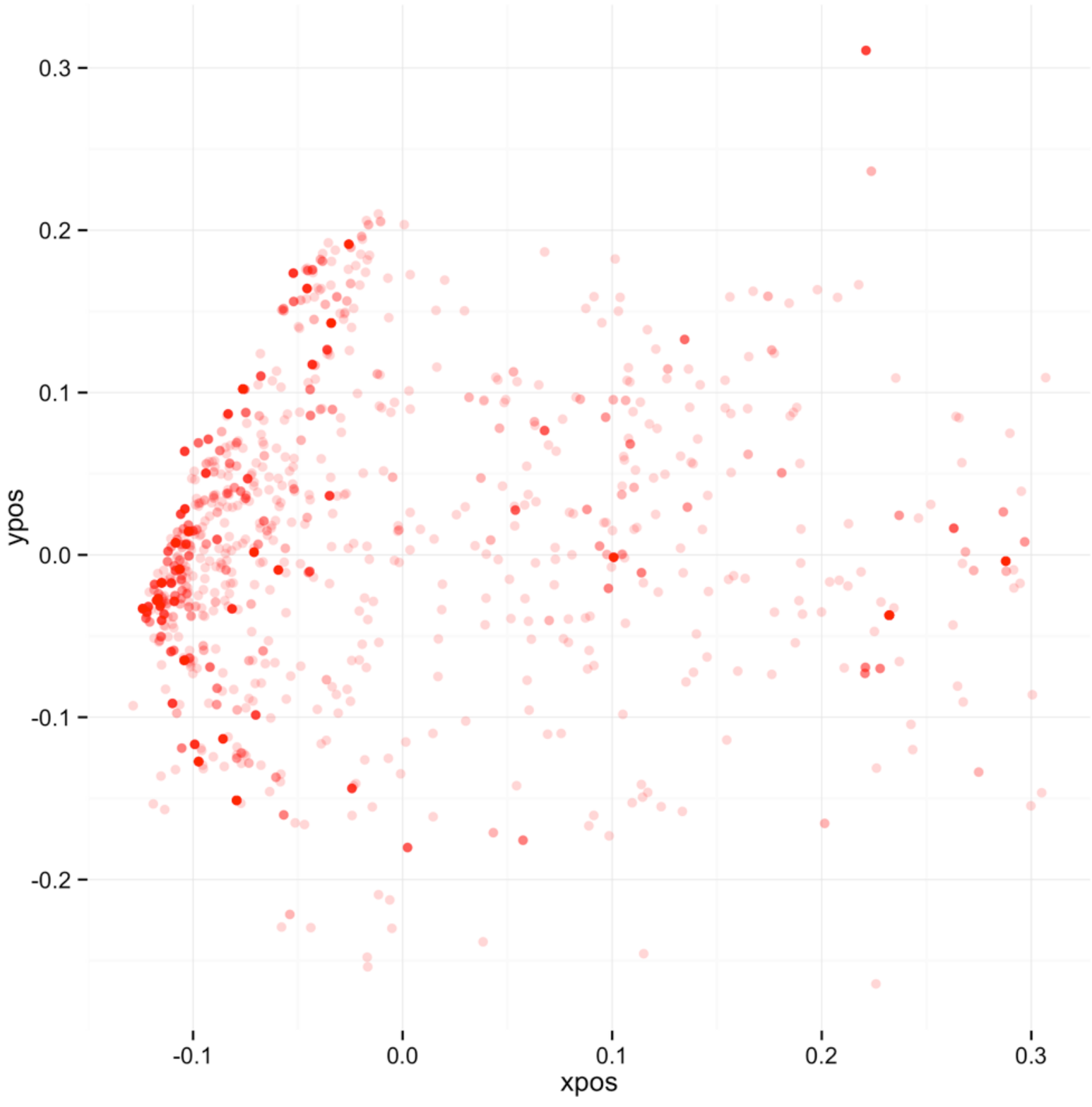
Testing only what is likely to fail



System A



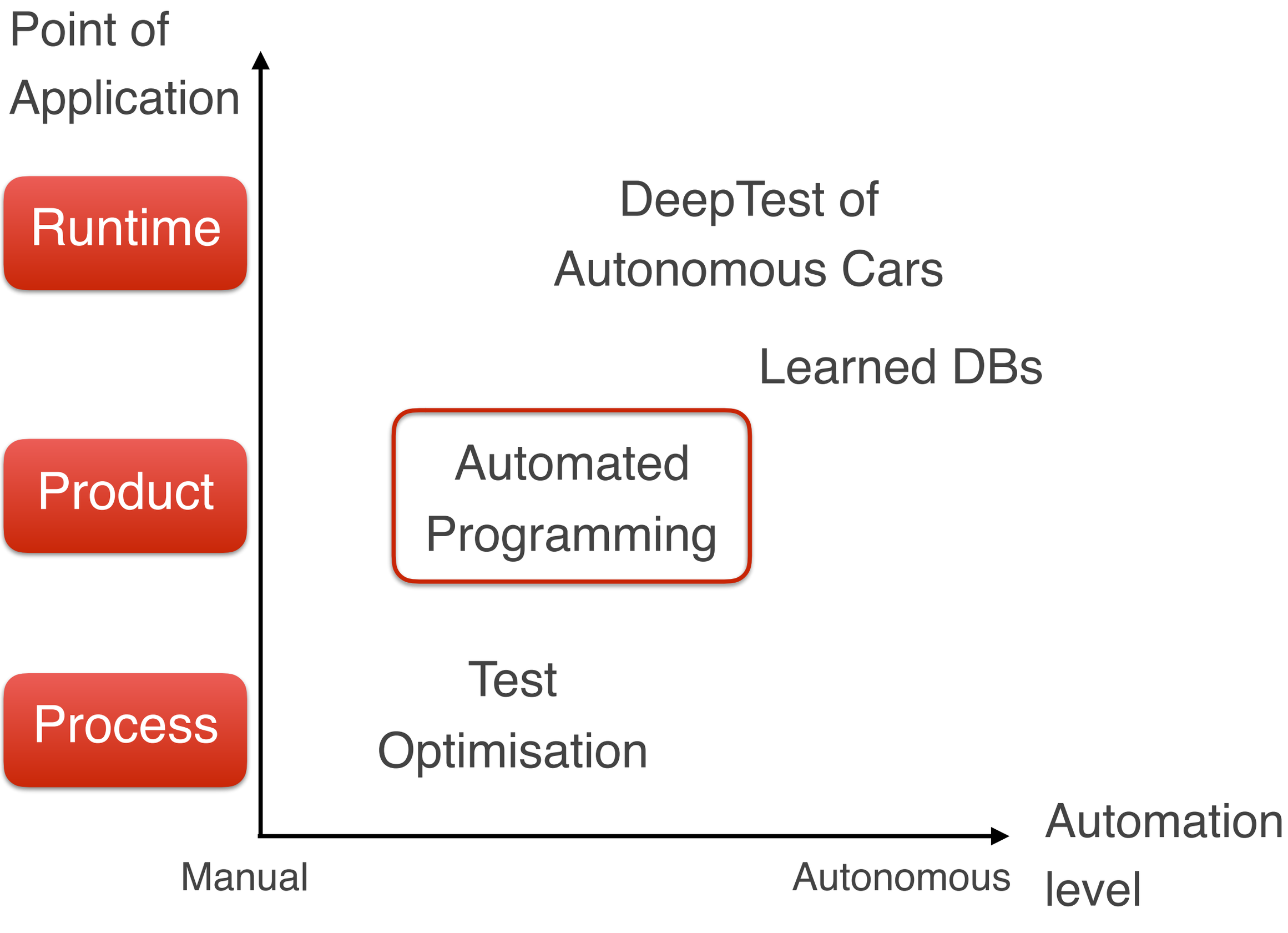
System B



Lessons learned: AI in SE&Test Analytics/Optimization

- Quality of data more important than advanced AI/ML
  - How much data do you have?
  - Do the data represent all important aspects?
- Simple statistical models often almost as good as advanced AI/ML
  - Data often unreliable => simple models give (at least) 80% of value for 20% of complexity
  - Statistical models easier to understand => robust
- Online algorithms almost always worth it => scalability
- Visualising results important for impact, Human + AI > AI
- An AI system is not enough, people need training + understanding to change their behaviour

AI-in-SE applications have different levels of risk/gain



Cozy: Generalised Data Structure Synthesis

ICSE'18, May 27–June 3, 2018, Gothenberg, Sweden

Calvin Loncaric, Michael D. Ernst, and Emina Torlak

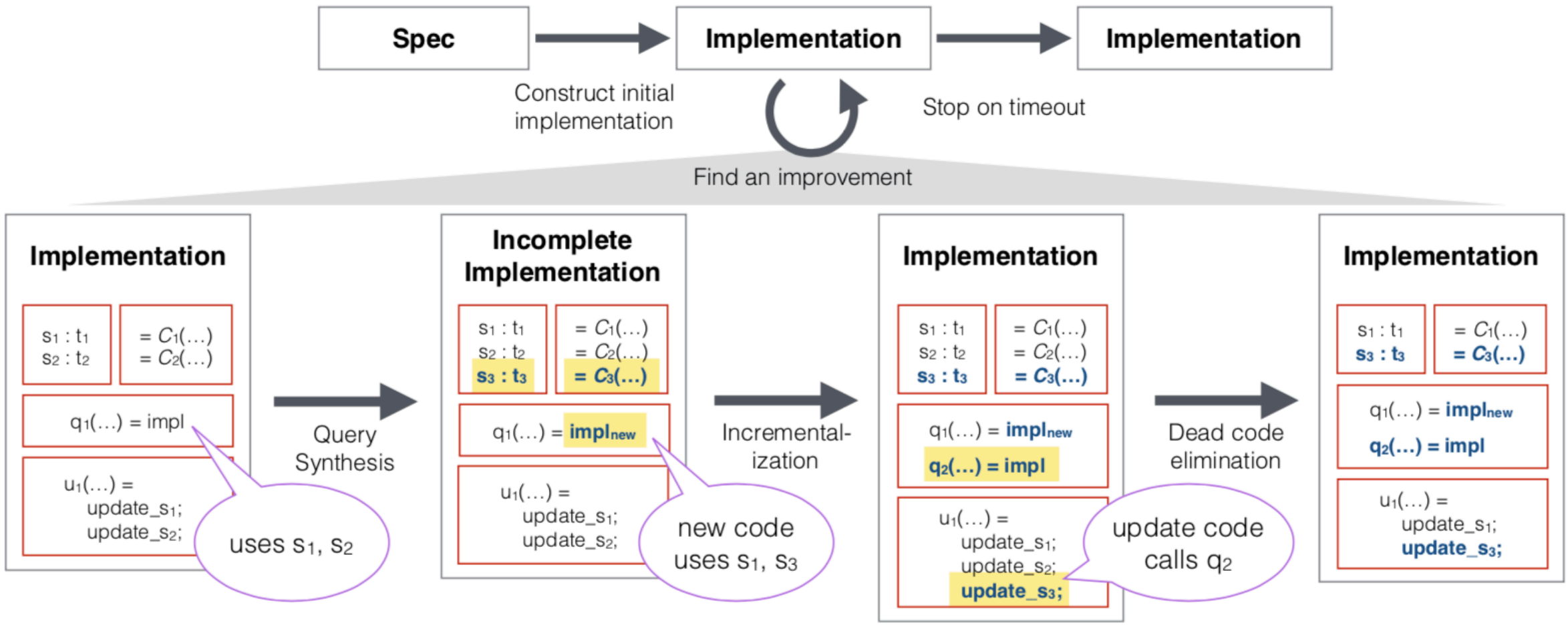


Figure 1: Architecture of Cozy. Each iteration through the loop performs query synthesis, incrementalization, and dead code elimination. Figure 2a shows example input, and Figures 2b and 3 show the corresponding output.

Cozy: Generalised Data Structure Synthesis

Table 1: Programmer effort. LoC measurements do not include comments or whitespace.

Project	Hand-written Span	Commits	LoC	Cozy LoC
ZTopo	1 week	15	1024	41
Sat4j	8 years	22	195	42
Openfire	10 years	47	1992	157
Lucene	13 years	20	68	36

Table 2: Correctness results. ZTopo has no dedicated issue tracker.

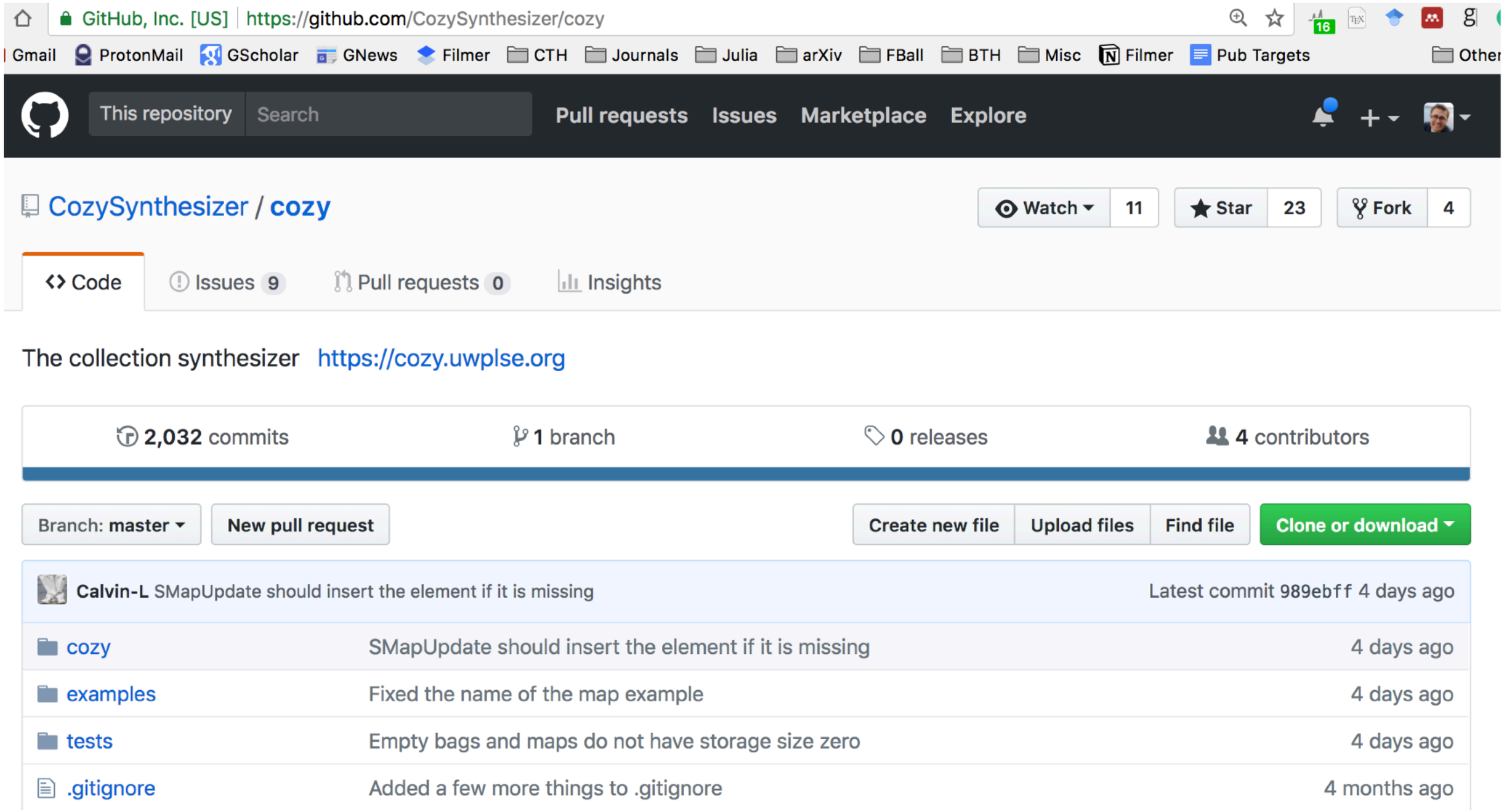
Project	Issues	New defects found
ZTopo	n/a	No
Sat4j	7	No
Openfire	25	Yes
Lucene	1	No

Table 3: Performance results. All times are in seconds.

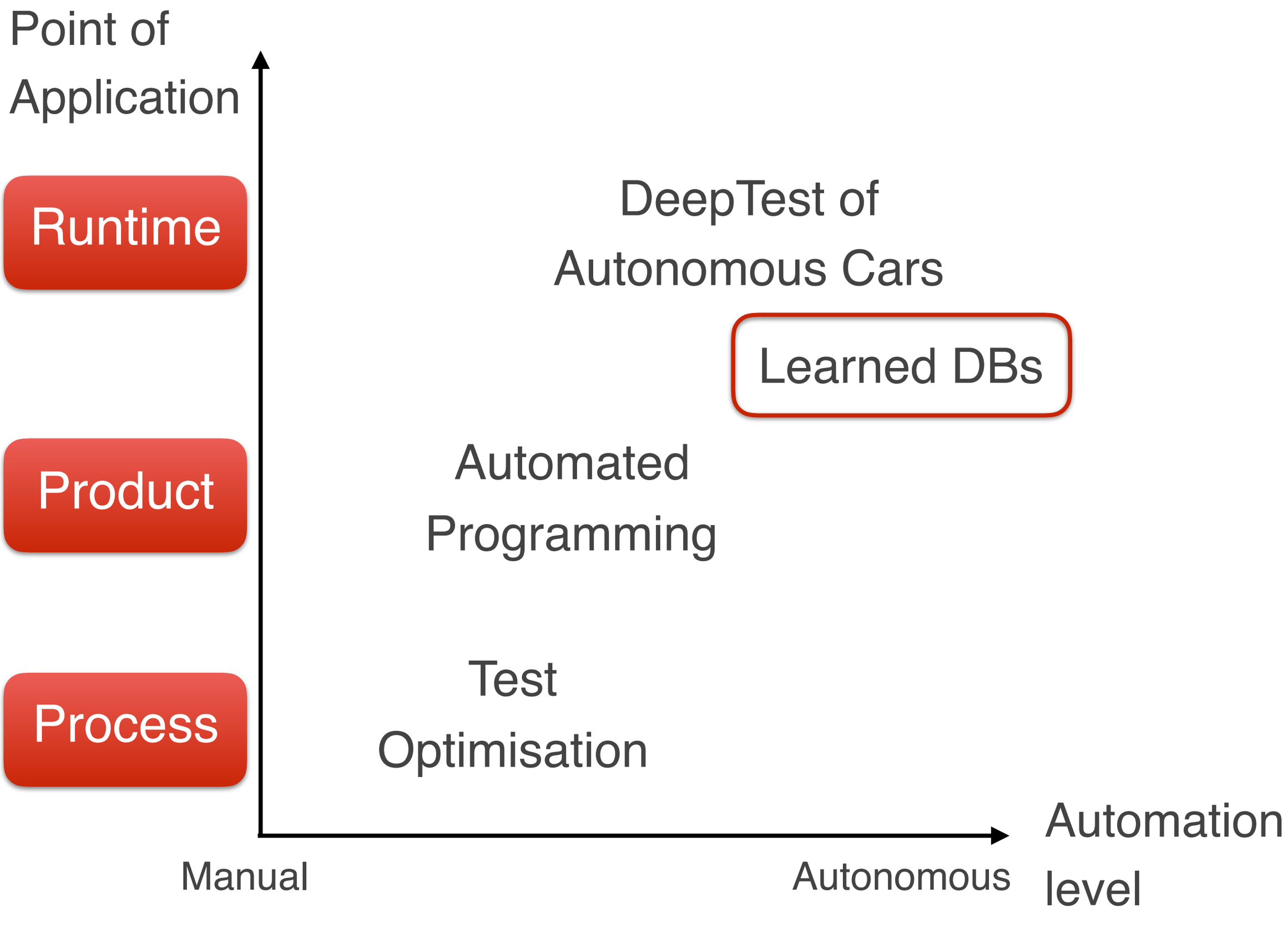
Project	Time (orig.)	Time (Cozy)
ZTopo	5	5
Sat4j	53	61
Openfire	16	15
Lucene	9	9

Cozy: Generalised Data Structure Synthesis

Try it yourself!



AI-in-SE applications have different levels of risk/gain



Product/NeuralNet/10 AI-in-SE: Learned DB Indices

The Case for Learned Index Structures

Tim Kraska*
MIT
Cambridge, MA
kraska@mit.edu

Alex Beutel
Google, Inc.
Mountain View, CA
alexbeutel@google.com

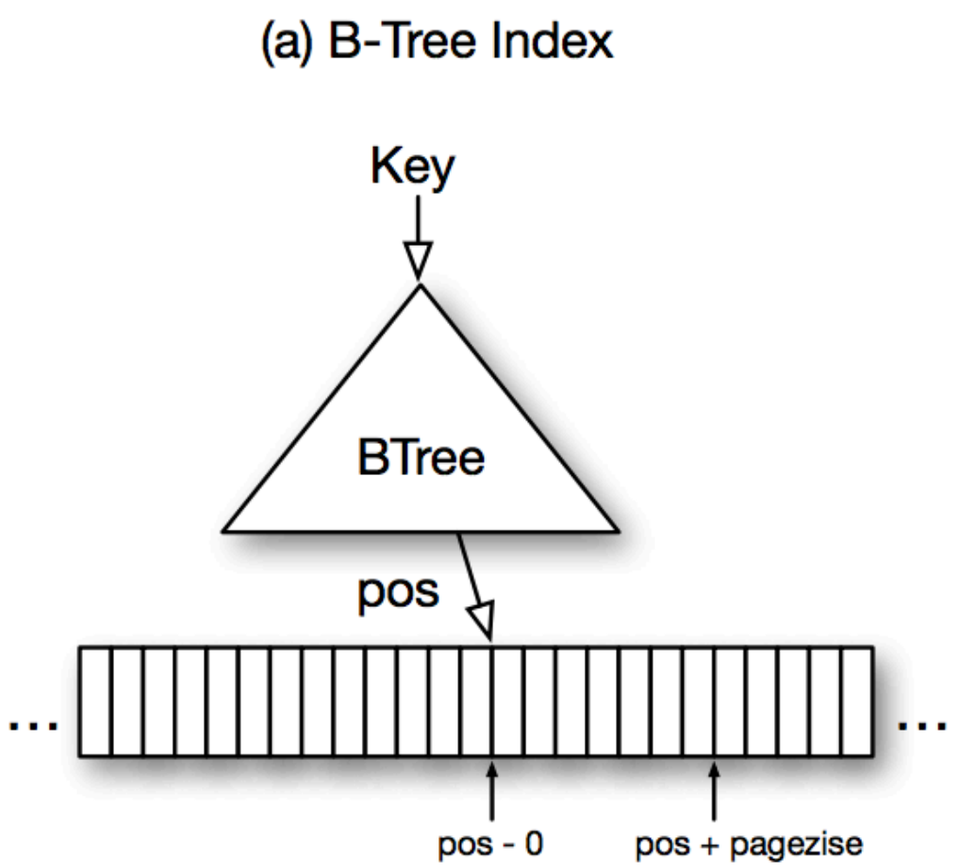
Ed H. Chi
Google, Inc.
Mountain View, CA
edchi@google.com

Jeffrey Dean
Google, Inc.
Mountain View, CA
jeff@google.com

Neoklis Polyzotis
Google, Inc.
Mountain View, CA
npolyzotis@google.com

[https://arxiv.org/pdf/1712.01208.pdf]

Product/NeuralNet/10 AI-in-SE: Learned DB Indices



[https://arxiv.org/pdf/1712.01208.pdf]

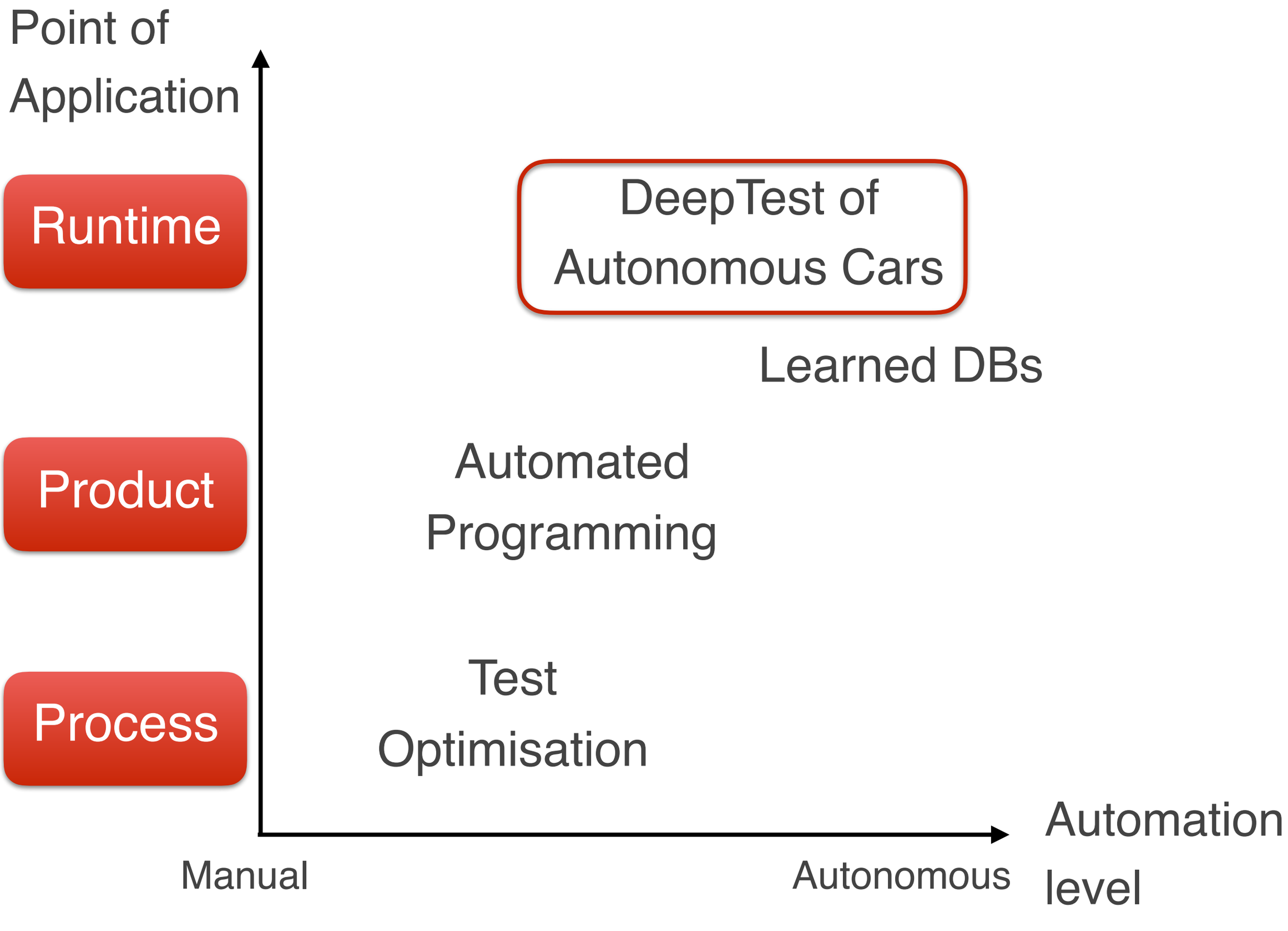
Product/NeuralNet/10 AI-in-SE: Learned DB Indices

Type	Config	Search	Total (ns)	Model (ns)	Search (ns)	Speedup	Size (MB)	Size Savings	Model Err ± Err Var.
Btree	page size: 16	Binary	280	229	51	6%	104.91	700%	4 ± 0
	page size: 32	Binary	274	198	76	4%	52.45	300%	16 ± 0
	page size: 64	Binary	277	172	105	5%	26.23	100%	32 ± 0
	page size: 128	Binary	265	134	130	0%	13.11	0%	64 ± 0
	page size: 256	Binary	267	114	153	1%	6.56	-50%	128 ± 0
Learned Index	2nd stage size: 10,000	Binary	98	31	67	-63%	0.15	-99%	8 ± 45
		Quaternary	101	31	70	-62%	0.15	-99%	8 ± 45
	2nd stage size: 50,000	Binary	85	39	46	-68%	0.76	-94%	3 ± 36
		Quaternary	93	38	55	-65%	0.76	-94%	3 ± 36
	2nd stage size: 100,000	Binary	82	41	41	-69%	1.53	-88%	2 ± 36
		Quaternary	91	41	50	-66%	1.53	-88%	2 ± 36
	2nd stage size: 200,000	Binary	86	50	36	-68%	3.05	-77%	2 ± 36
		Quaternary	95	49	46	-64%	3.05	-77%	2 ± 36
Learned Index Complex	2nd stage size: 100,000	Binary	157	116	41	-41%	1.53	-88%	2 ± 30
		Quaternary	161	111	50	-39%	1.53	-88%	2 ± 30

Figure 4: Map data: Learned Index vs B-Tree

[https://arxiv.org/pdf/1712.01208.pdf]

AI-in-SE applications have different levels of risk/gain



Product+Process/NeuralNet/10 AI-in-SE: DeepTest

Table 1: Examples of real-world accidents involving autonomous cars

	Reported Date	Cause	Outcome	Comments
Hyundai Competition [4]	December, 2014	Rain fall	Crashed while testing	"The sensors failed to pick up street signs, lane markings, and even pedestrians due to the angle of the car shifting in rain and the direction of the sun" [4]
Tesla autopilot mode [17]	July, 2016	Image contrast	Killed the driver	"The camera failed to recognize the white truck against a bright sky" [23]
Google self-driving car [12]	February, 2016	Failed to estimate speed	Hit a bus while shifting lane	"The car assumed that the bus would yield when it attempted to merge back into traffic" [12]

Hyperdrive

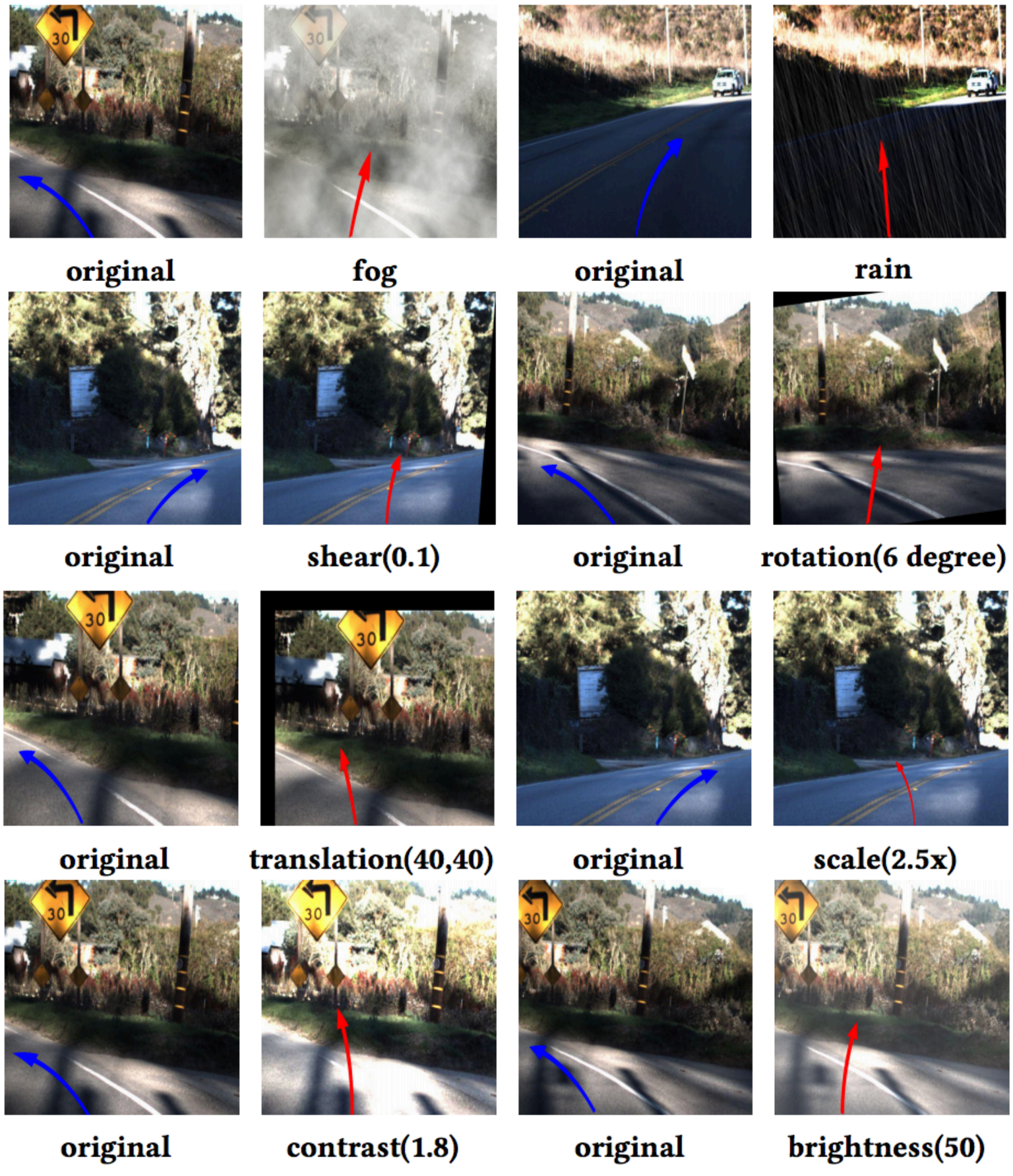
Human Driver Could Have Avoided Fatal Uber Crash, Experts Say

By Ryan Beene, Alan Levin, and Eric Newcomer

22 March 2018, 18:04 CET *Updated on 22 March 2018, 20:27 CET*

- Human driver may have avoided impact: forensic crash analysts
- Self-driving sensors should have detected victim, experts say

Product+Process/NeuralNet/10 AI-in-SE: DeepTest



Neuron Coverage instead of Code Coverage

Neuron Coverage can guide systematic test generation (applying image transformations)

Combining image transformations increases coverage

Over 1000 errors detected in 3 tested models

NNs improved 46% after retraining on generated test images

[https://arxiv.org/pdf/1708.08559.pdf]

Kwiatkowska: Minimal Adverserial Examples

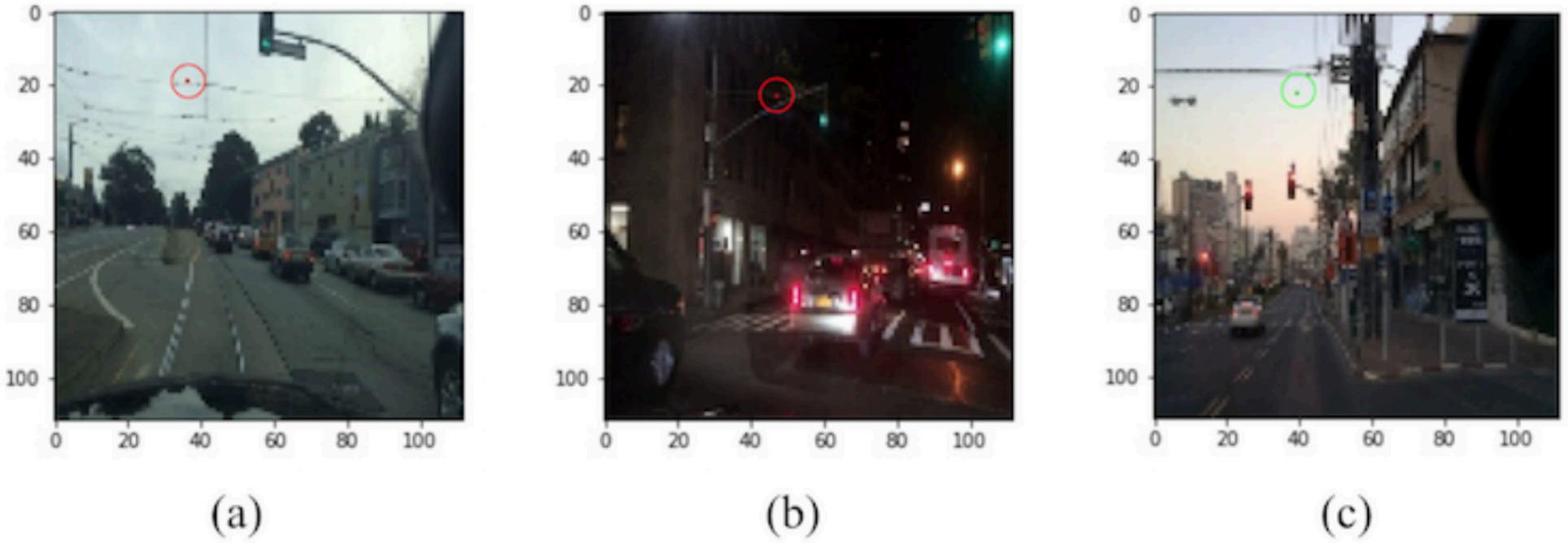


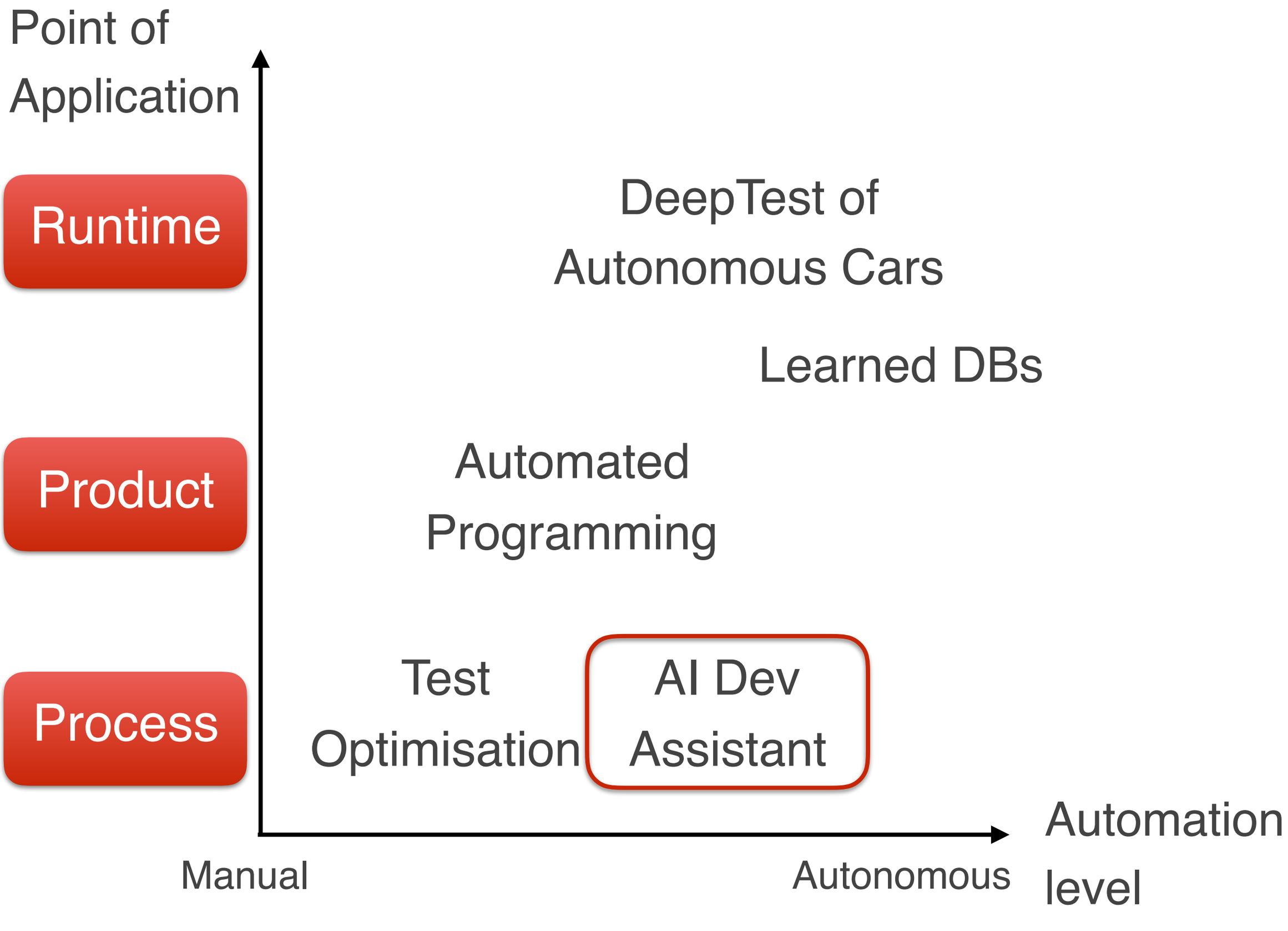
Fig. 4: Adversarial examples generated on Nexar data demonstrate a lack of robustness. (a) Green light classified as red with confidence 56% after one pixel change. (b) Green light classified as red with confidence 76% after one pixel change. (c) Red light classified as green with 90% confidence after one pixel change.

[https://arxiv.org/pdf/1710.07859.pdf]

ICST Keynote on Verification of Deep NNs:

[https://www.youtube.com/watch?v=OTXEzJnzUV0]

Bonus ICSE paper: Voice-AI as Conversational Assistant



Bonus ICSE: Devy the Voice-AI Developer Assistant

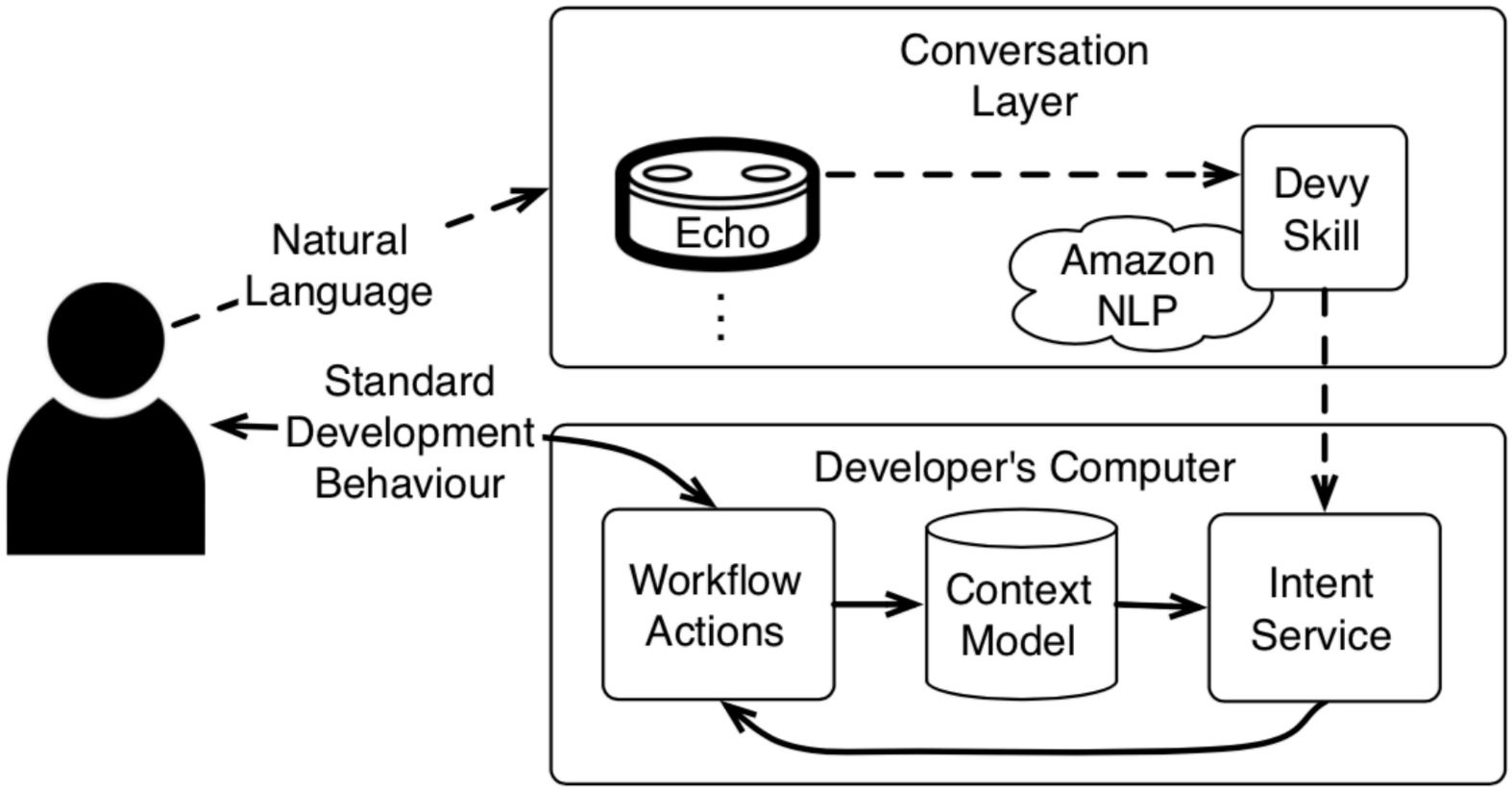


Figure 2: Devy's architecture. A developer expresses their intention in natural language via the conversational layer. The intent service translates high-level language tokens into low-level concrete workflows which can then be automatically executed for the developer. Dotted edges predominantly communicate in the direction of the arrow, but can have back edges in case clarification is needed from the user.

[Bradley et al, ICSE 2018]

Bonus ICSE: Devy the Voice-AI Developer Assistant

Table 1: Manual steps for the common 'share changes' workflow.

(a) Open a web browser for the issue tracker and check the issue number for the current work item.
(b) Open a terminal and run the tests against the changed code to ensure they work (e.g., `npm run tests`).
(c) Open a terminal and commit the code, tagging it with the current work item number (e.g., `git commit -m 'See issue #1223'`).
(d) Pull any external changes from the remote repository (e.g., `git pull`).
(e) Push the local change to the remote repository (e.g., `git push`).
(f) Open the commit in the version control system using the GitHub web interface and open a pull request.
(g) Determine a set of reviewers and assign them to the pull request with the GitHub web interface.

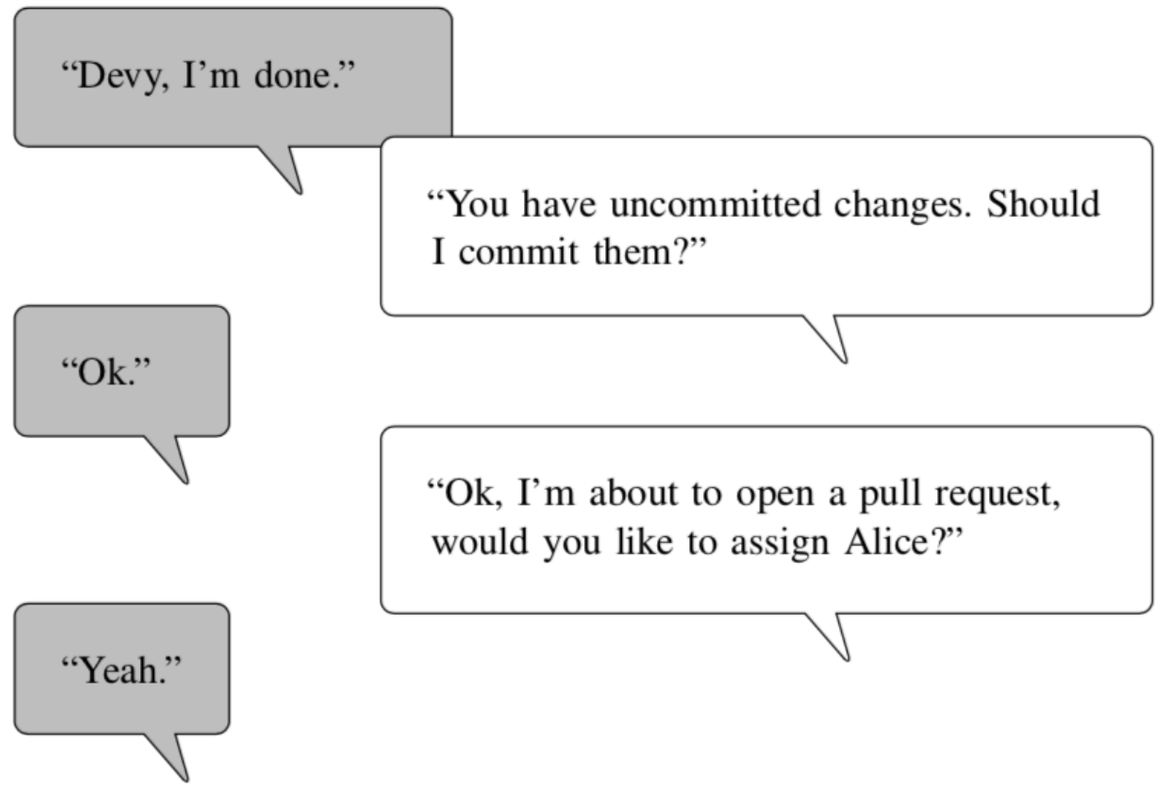


Figure 1: The conversation between the developer and Devy for completing the 'submit changes for review' workflow. The developer's voice commands are given with a grey background, Devy's responses have a clear background. In contrast to Table 2, the context model enables the Devy conversation to be relatively terse, despite the complexity of the workflow.

[Bradley et al, ICSE 2018]

Bonus ICSE Not Testing: Great SE Manager?

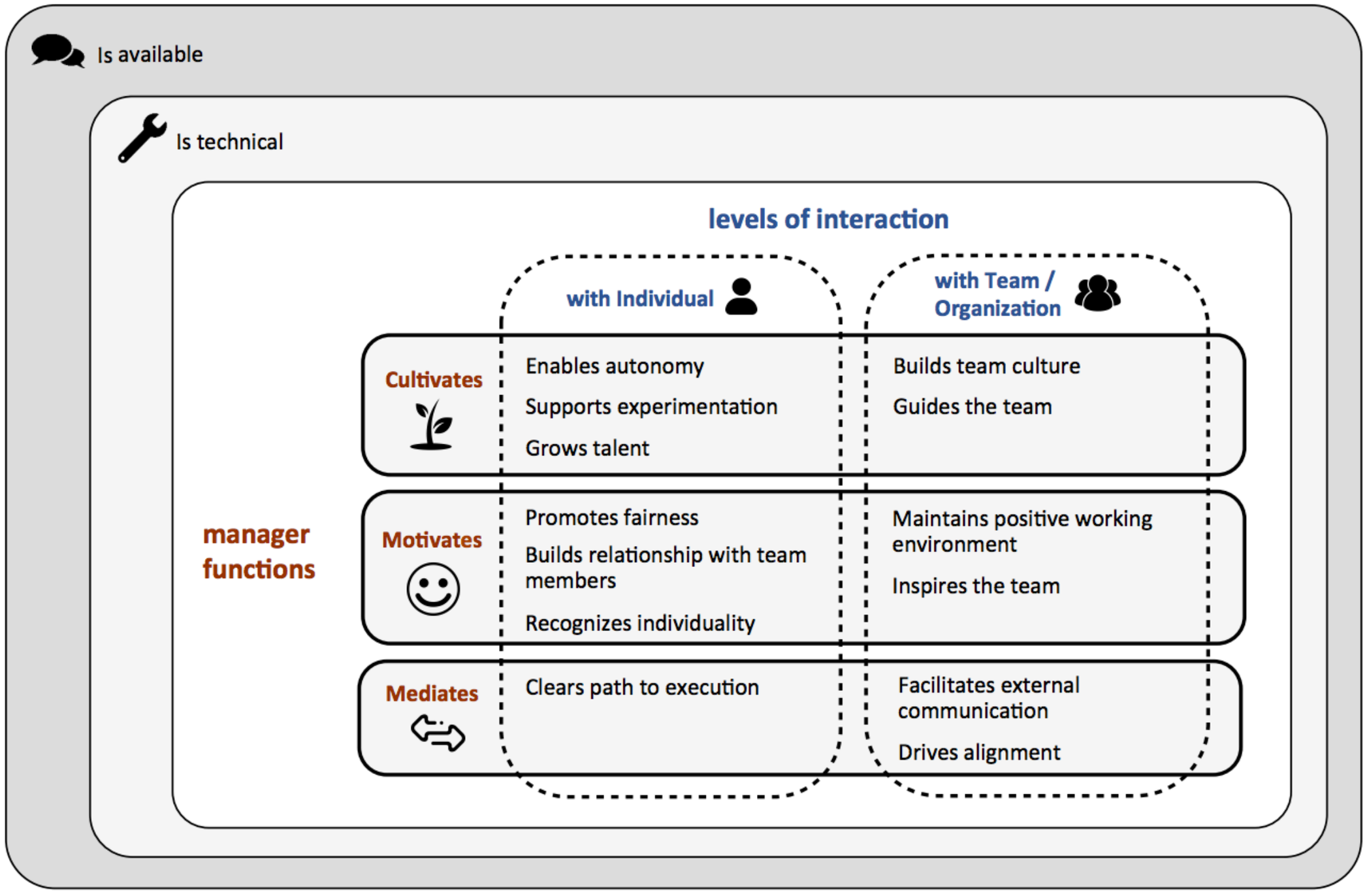


Fig. 1: Conceptual framework for great software engineering managers

[Kalliamvakou et al, ICSE 2018]

Vidare tips/länkar

ICST Keynotes:

1. Facebook's statiska kodanalys och testverktyg
2. DICE/EA om Testing i spelindustrin
3. Kwiatkowska om att hitta AI buggar automatiskt

www.es.mdh.se/icst2018/live/

robert.feldt@chalmers.se Twitter: @drfeldt

http://www.robertfeldt.net