ON IMPACT IN SOFTWARE ENGINEERING RESEARCH

ANDREAS ZELLER, CISPA HELMHOLTZ CENTER FOR IT SECURITY

DAGSTUHL WORKSHOP "SE FORSCHUNGSMETHODENTRAINING"
MARCH 2, 2020

@AndreasZeller

ANDREAS ZELLER: KEY FACTS

- PhD in 1997 on Configuration Management with Feature Logic
- Since 2001 in Saarbrücken, Germany (Saarland University + CISPA)
- Four 10-year impact awards 2009–2017 (for papers 1999–2007)
- ACM Fellow in 2010
- ERC Advanced Grant in 2011
- SIGSOFT Outstanding Research Award in 2018

ANDREAS ZELLER: KEY FACTS

- Since 2019, Faculty at *CISPA Helmholtz Center for Information Security*
- Roughly equivalent to a Director at a Max Planck Institute
- Devoted to groundbreaking fundamental research in IT Security
- Seven funded PhD positions, minimal teaching obligations
- Awe-inspiring colleagues + students, great team work

– I am a minority

WHAT IS IMPACT?

WHAT IS IMPACT?

- *How do your actions change the world?*
- Often measured in citations, publications, funding, people, …
- All these are indicators of impact, *but not goals in themselves*
- We want to make the world a better place
- Gives meaning and purpose to our (professional) life

WHAT MAKES IMPACTFUL RESEARCH?

- *Intellectual challenge* – was it hard, or could anyone have done this?
- *Elegance* – is your research specific to a context, or can it be reused again and again?
- *Usefulness* – can someone make money with it?
- Innovation is the *delta* in any of these metrics

IMPACT OUTSIDE OF SE

- *Programming Languages* folks miss the intellectual challenge
- *Formal Methods* folks miss elegance *and* challenge
- *Industry* folks miss usefulness and applicability
- Far too often, we recluse in our *private bubbles*

MY PATH TO IMPACT

MY PATH TO IMPACT

- Life can only be understood backwards; but it must be lived forwards (Søren Kierkegaard)

CONFIGURATION MANAGEMENT WITH FEATURE LOGIC (1991–1997)

- Topic defined by my PhD advisor Gregor Snelting
- Idea: Formally describe variants and revisions with *feature logic*
- "A unified model for configuration management"

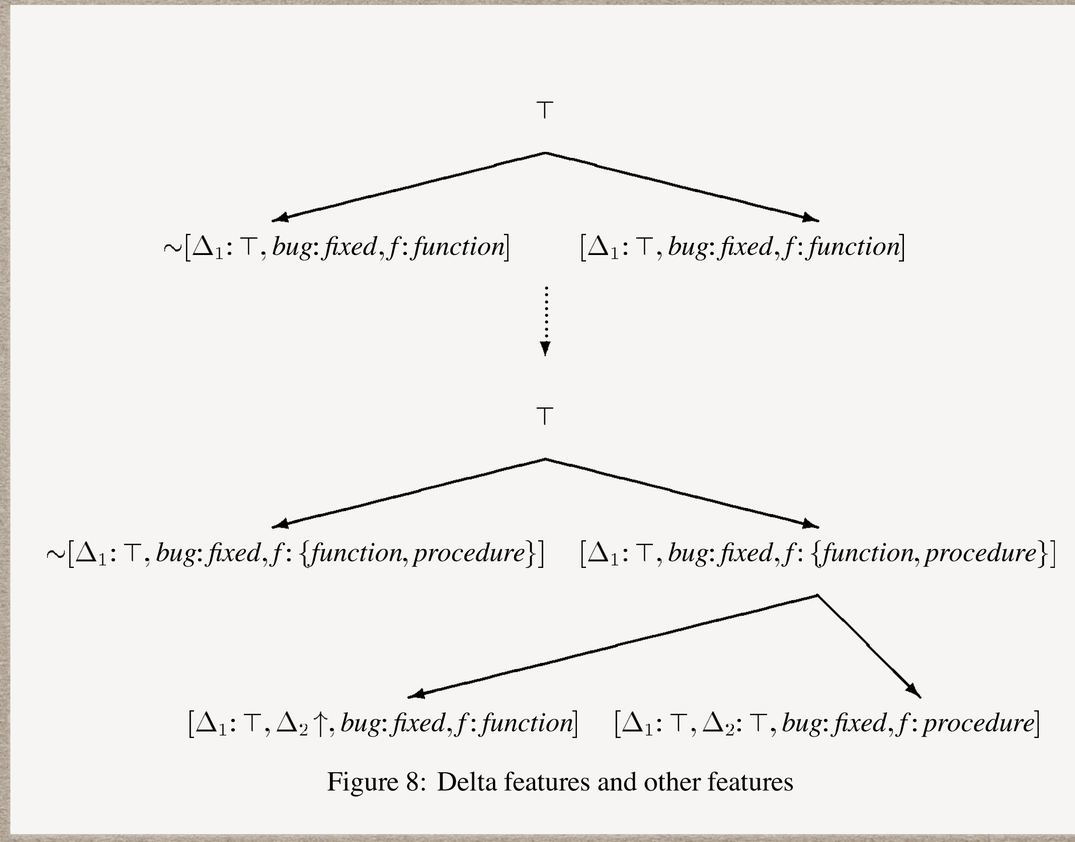


Figure 8: Delta features and other features

FEATURE LOGIC: LESSONS LEARNED

- **(None)**

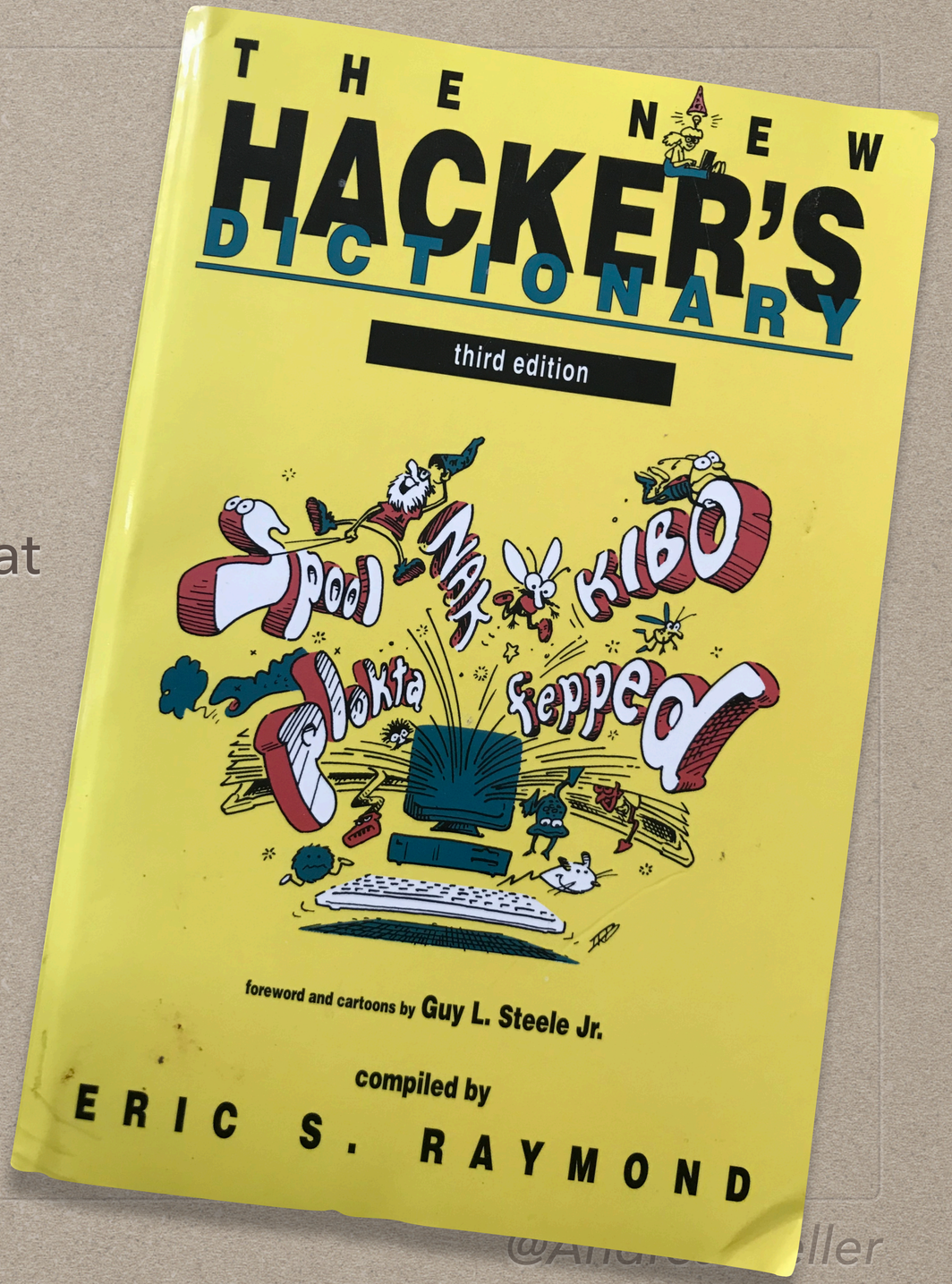
– did everything wrong

FEATURE LOGIC: LESSONS LEARNED

- You can get plenty of papers accepted
  - even if you miss the problem
  - even if you neither prove nor evaluate
- “Modeling for the sake of modeling”
- Enabled much of my later work, though

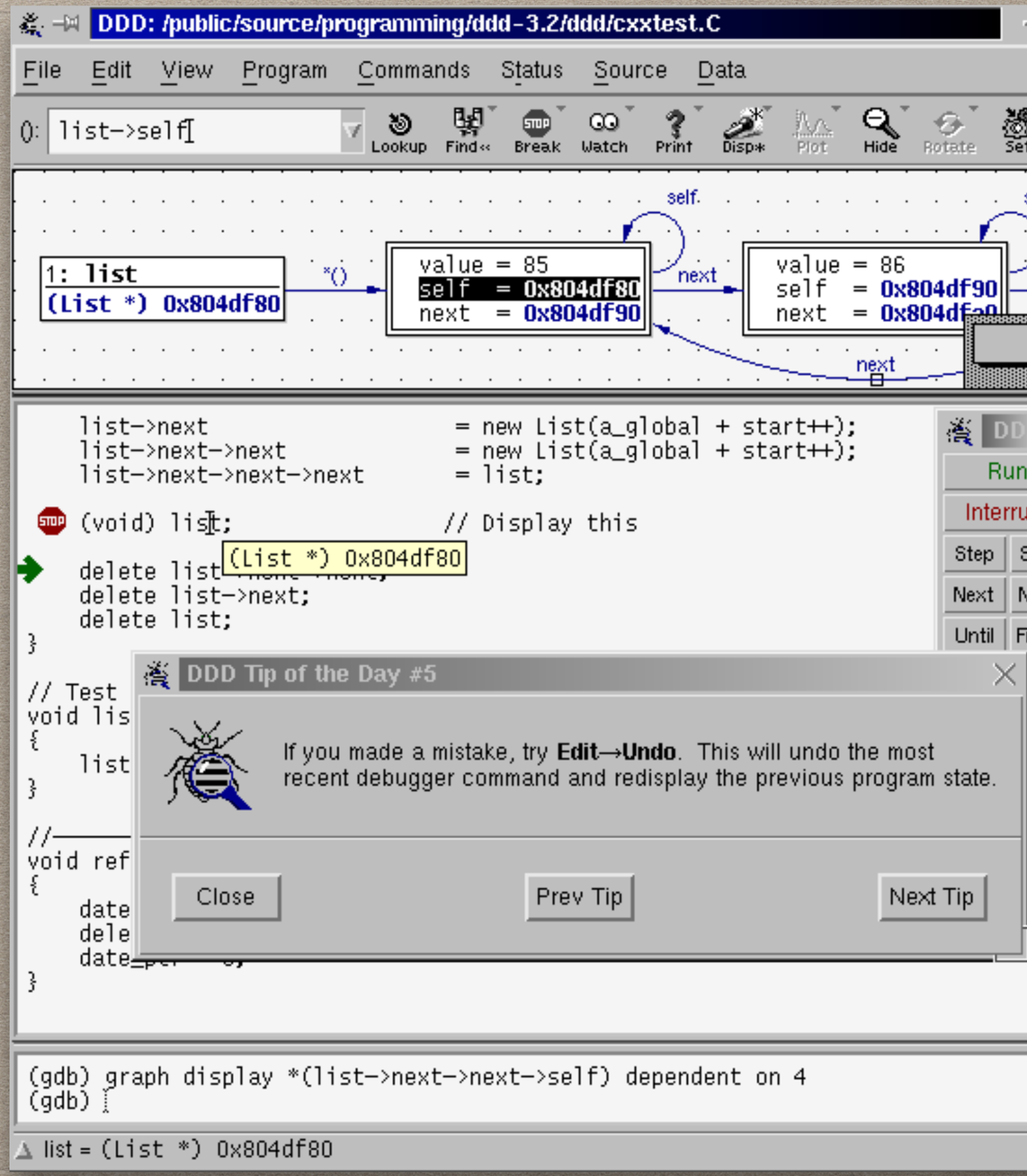
WHAT TO DO AFTER PHD

- During PhD, found standards and topics at German IT companies disappointing
- Academia seemed good alternative
- Socialized by open source development



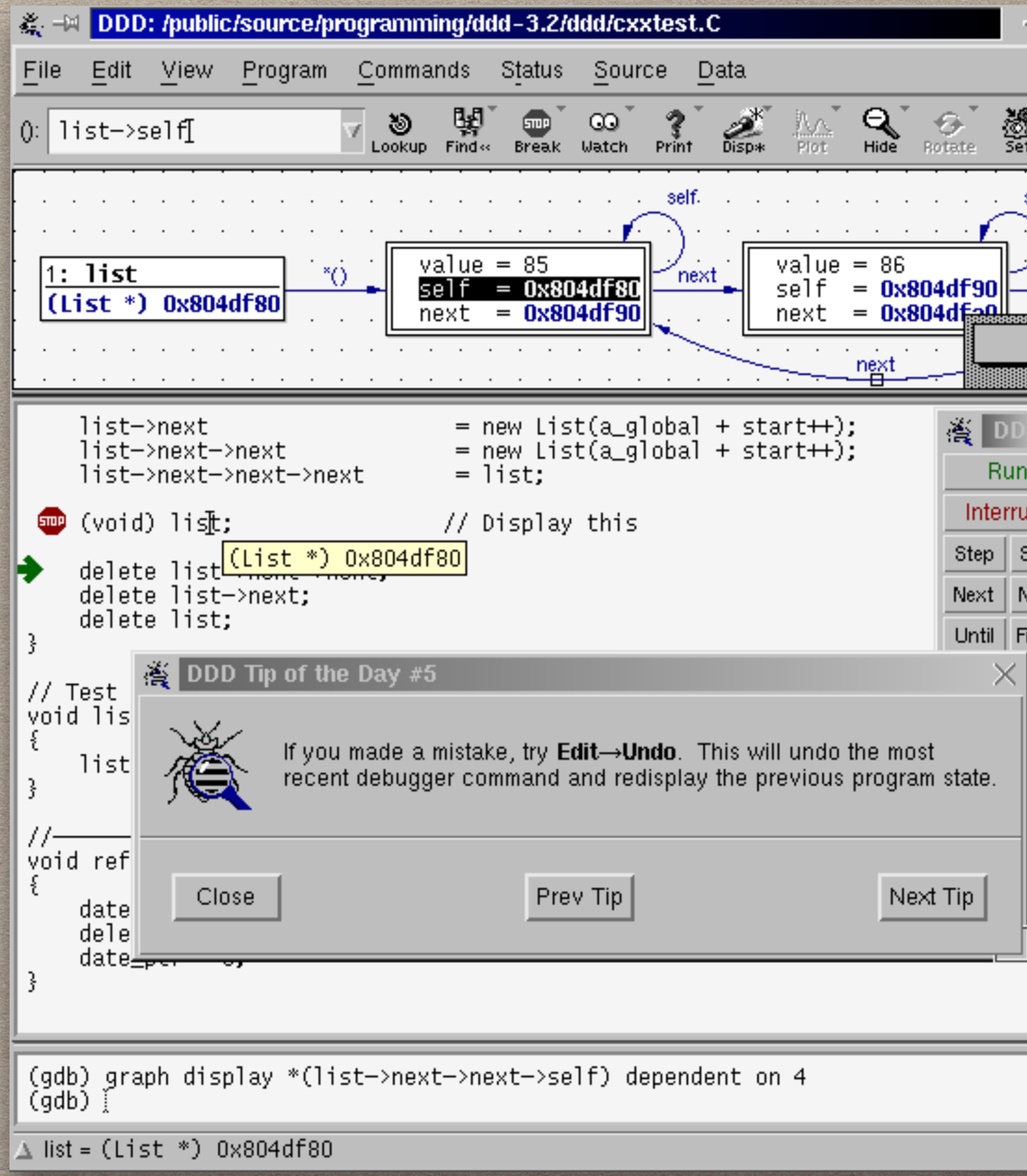
DDD (1994-1999)

- During PhD, programmed a lot
- Debugging was hard!
- Built the DDD debugger GUI with Dorothea Lütkehaus
- Welcome change from formal work



DDD (1994-1999)

- DDD was among the first dev tools with a “professional” GUI
- Downloaded by the tens of thousands
- Adopted as a GNU project: Street credibility with developers
- Impact through *usefulness*

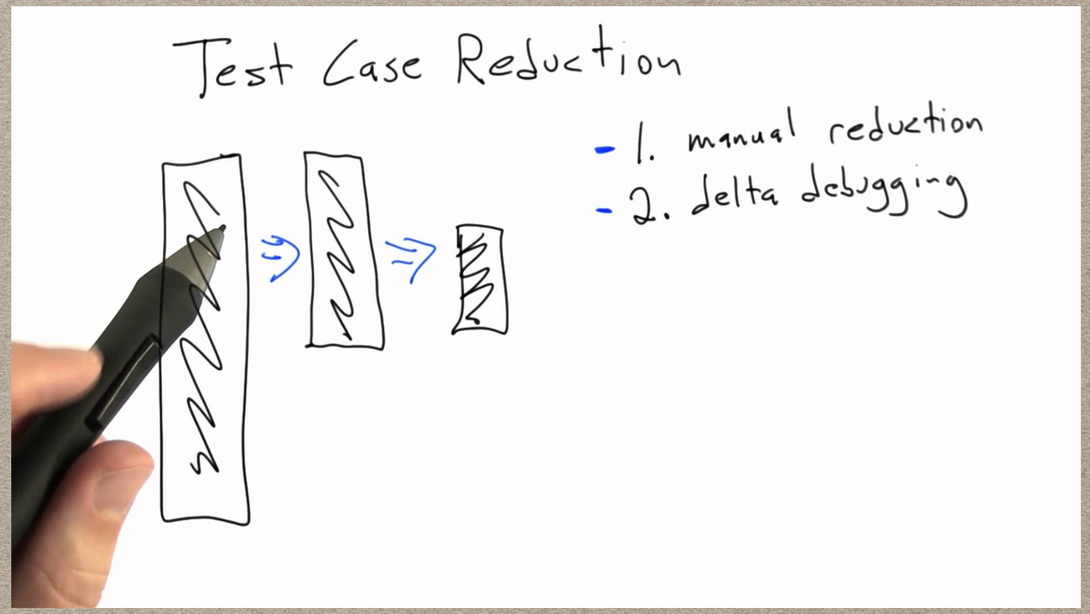


DDD: LESSONS LEARNED

- **Work on a real problem** – *"real" as in "real world", not "real papers"*
- **Assume as little as possible** – *make things fit into real processes*
- **Keep things simple** – *complexity impresses, but prevents impact*

DELTA DEBUGGING (1999–2003)

- After PhD, looking for new topic
- Delta Debugging brought together debugging and version control
- Isolate failure causes through repeated experiments



DELTA DEBUGGING (1999–2003)

- Delta debugging was a bomb
- Easy to teach + understand
- 7 lines of algorithm
  (and 25 lines of Python)
- Spent two years on these

$$dd(c_{✔}, c_{✘}) = dd'(c_{✔}, c_{✘}, 2)$$

$$dd'(c'_{✔}, c'_{✘}, n) =$$

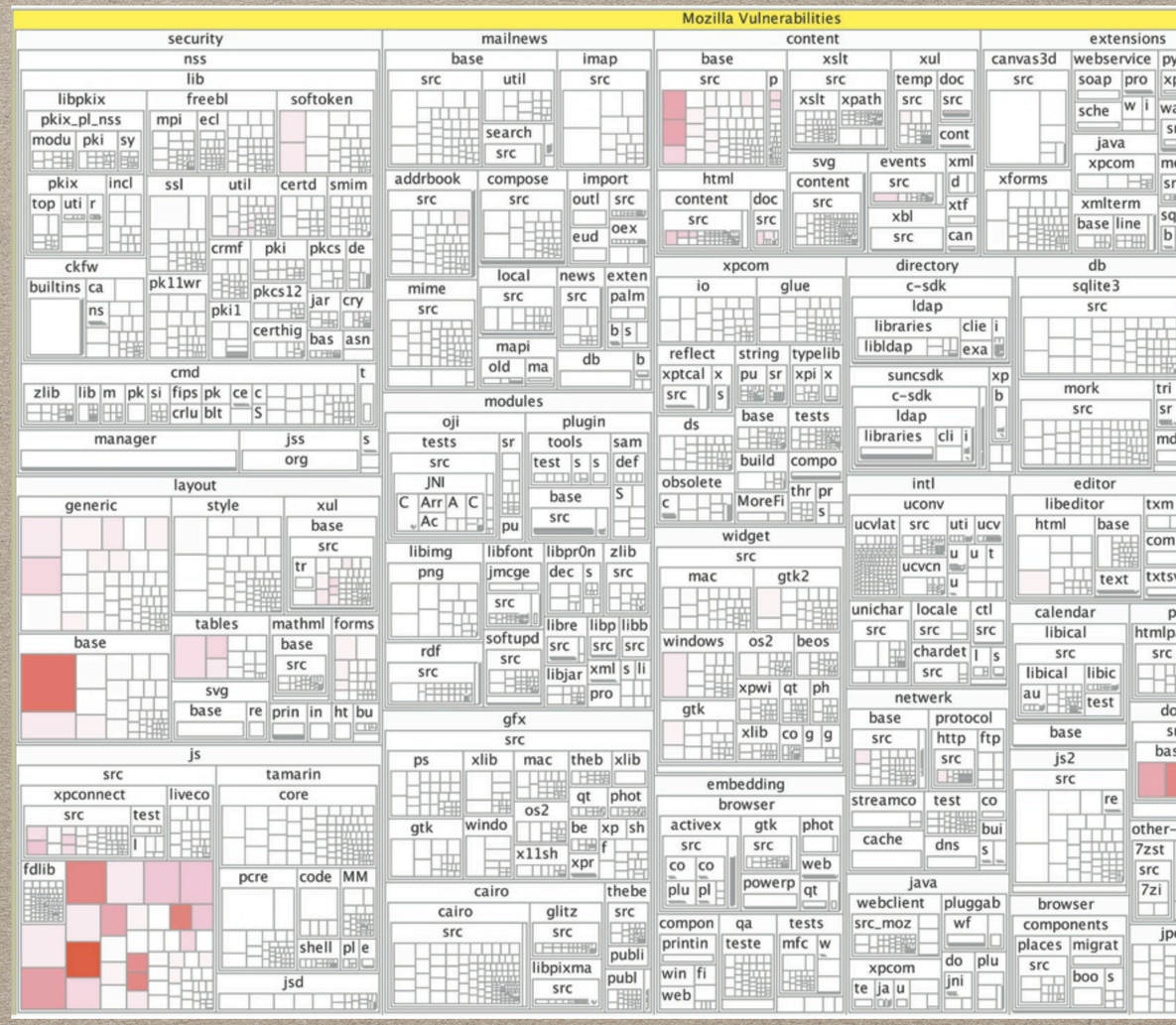
$$\begin{cases} (c'_{✔}, c'_{✘}) & \text{if } |\Delta| = 1 \\ dd'(c'_{✘} \setminus \Delta_i, c'_{✘}, 2) & \text{if } \exists i \in \{1..n\} \cdot test(c'_{✘} \setminus \Delta_i) = ✔ \\ dd'(c'_{✔}, c'_{✔} \cup \Delta_i, 2) & \text{if } \exists i \in \{1..n\} \cdot test(c'_{✔} \cup \Delta_i) = ✘ \\ dd'(c'_{✔} \cup \Delta_i, c'_{✘}, \max(n-1, 2)) & \text{else if } \exists i \in \{1..n\} \cdot test(c'_{✔} \cup \Delta_i) = ✔ \\ dd'(c'_{✔}, c'_{✘} \setminus \Delta_i, \max(n-1, 2)) & \text{else if } \exists i \in \{1..n\} \cdot test(c'_{✘} \setminus \Delta_i) = ✘ \\ dd'(c'_{✔}, c'_{✘}, \min(2n, |\Delta|)) & \text{else if } n < |\Delta| \text{ (“increase granularity”)} \\ (c'_{✔}, c'_{✘}) & \text{otherwise} \end{cases}$$

DELTA DEBUGGING: LESSONS LEARNED

- Work on a real problem — *Why debug? We build correct software*
- Assume as little as possible — *Version control? tests? Never heard of it*
- Keep things simple — *25 lines of Python is probably excessive*
- **Have a sound model** — *DD was my version model reborn*

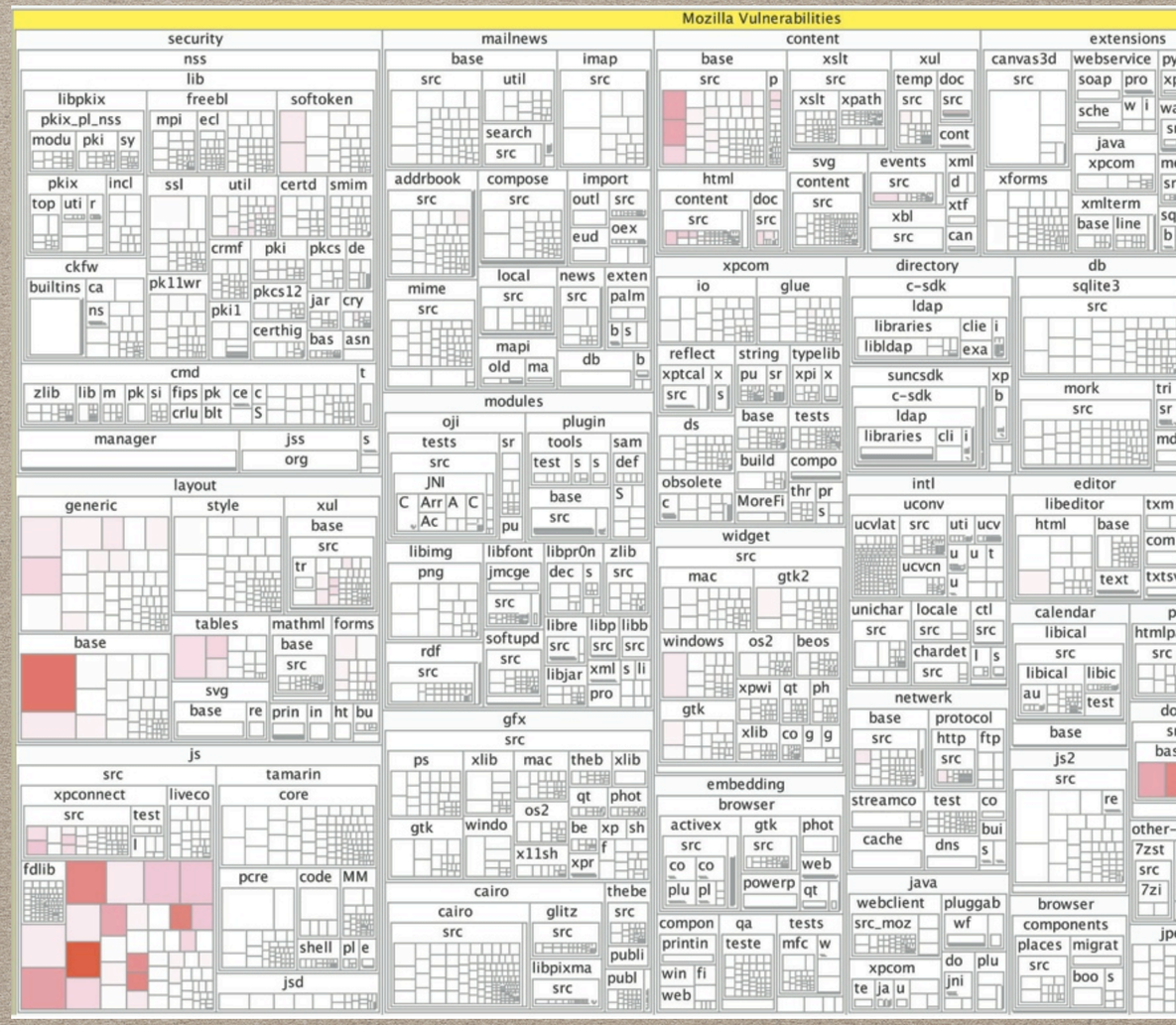
MINING SOFTWARE ARCHIVES (2003–2010)

- In the early 2000s, open-source version repositories became available
- Stephan Diehl saw an opportunity for visualization and approached me
- Quickly expanded into data mining
- Tom Zimmermann: our MSc student
- Work of a research team



MINING SOFTWARE ARCHIVES (2003–2010)

- Our 2004 paper was the first ICSE paper on mining software archives
- Handful of competing groups; instant hit
- MSR now a conference on its own
- Paper has ~1300 citations so far
- Impact at Microsoft, Google, SAP…



MINING SOFTWARE ARCHIVES (2003–2010)

- We are now after the gold rush
- Data still exciting (if you have some)
- Few new insights on old data
- Get out of a field when too crowded

Figure 2: Color-coding keys by their defect correlation; (red = strong). The five strongest correlations are highlighted.

MINING SOFTWARE REPOSITORIES: LESSONS LEARNED

- Work on a real problem — *Empirical research is core field of SE*
- Assume as little as possible — *simple parsers for multiple languages*
- Keep things simple — *essence of 2004 paper is one line of SQL*
- Have a sound model — *retrieval, precision, recall, etc, etc*
- **Keep on learning** — *statistics, data mining, machine learning*

FUZZING AND TEST GENERATION (2012-)

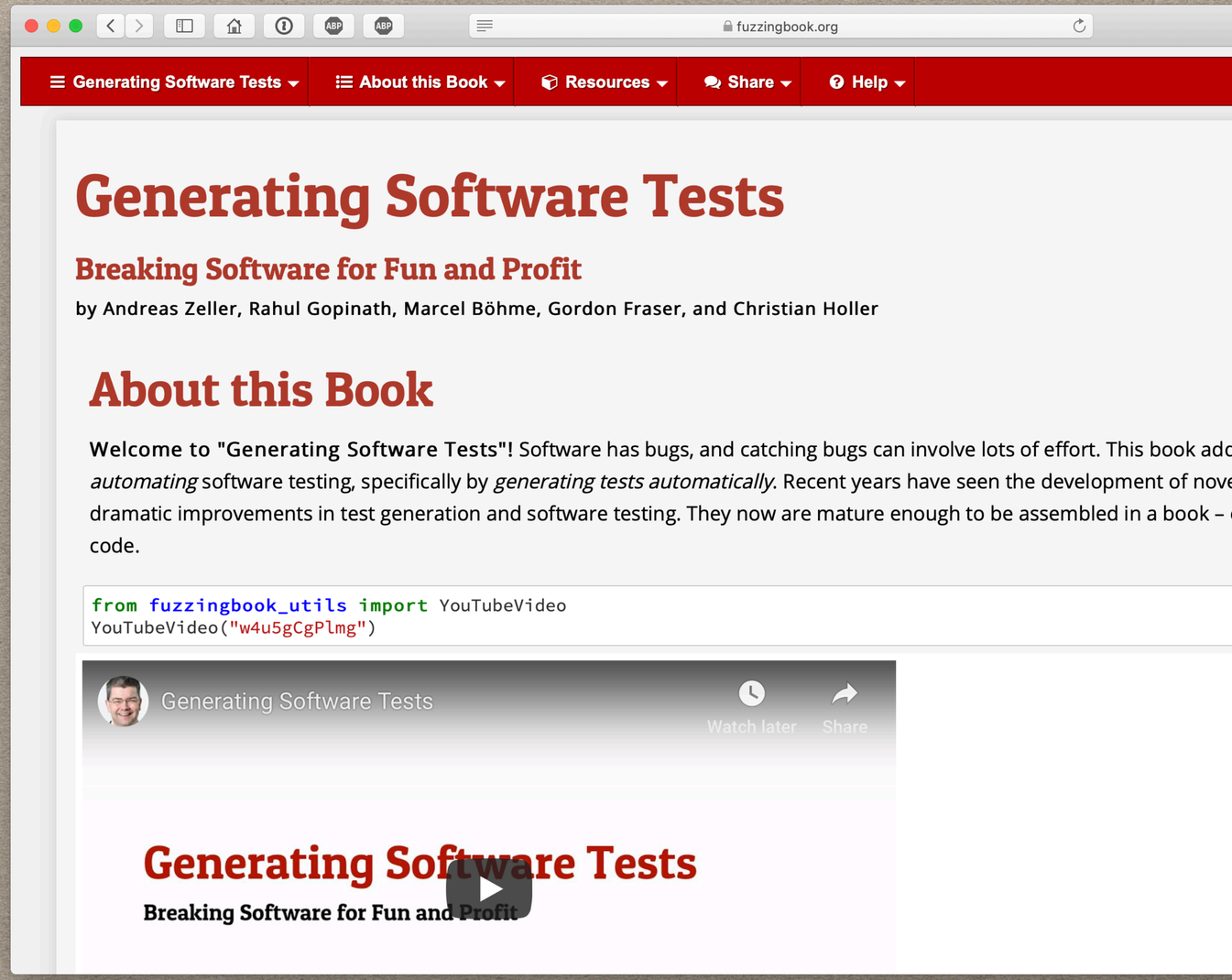
- In 2012, ran LangFuzz: a grammar-based fuzzer for JavaScript
- Found 2,600+ JavaScript bugs so far
- Work on grammar inference + more grammar-based testing
- Aim: build the best fuzzing framework ever

```
http://user:password@www.google.com:80/command?foo=bar
       &lorem=ipsum#fragment
http://www.guardian.co.uk/sports/worldcup#results
ftp://bob:12345@ftp.example.com/oss/debian7.iso
```

```
URL ::= PROTOCOL '://' AUTHORITY PATH
        ['?' QUERY] ['#' REF]
AUTHORITY ::= [USERINFO '@'] HOST [':' PORT]
PROTOCOL ::= 'http' | 'ftp'
USERINFO ::= /[a-z]+:[a-z]+/
HOST ::= /[a-z.]+/
PORT ::= '80'
PATH ::= /\/[a-z0-9.\/]*/
QUERY ::= 'foo=bar&lorem=ipsum'
REF ::= /[a-z]+/
```

FUZZING AND TEST GENERATION (2017-)

- Teaching hands-on fuzzing and test generation
- Uses Python and Jupyter
- Prototype state-of-the-art techniques within *minutes*
- Interactive textbook fuzzingbook.org



FUZZING AND TESTING: LESSONS LEARNED

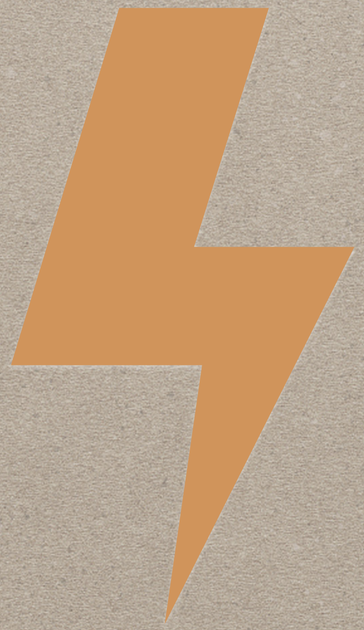
- Work on a real problem — *Yes, bugs do exist*
- Assume as little as possible — *Toss program into black box*
- Keep things simple — *Grammar-based producers*
- Have a sound model — *Grammars and languages*
- Keep on learning — *Constraint solving, search-based testing*
- **Keep on moving** — *Security starts with SE*
- **Build prototypes** — *Get your algorithms right first*

MORE THINGS I DID (AND DO!)

- **Automatic repair** – *Wesley Weimer beat us to it*
- **Automatic parallelization** – *Struggled with complexity*
- **Automatic website testing** – *Built a company for that*

THINGS I STAYED AWAY FROM

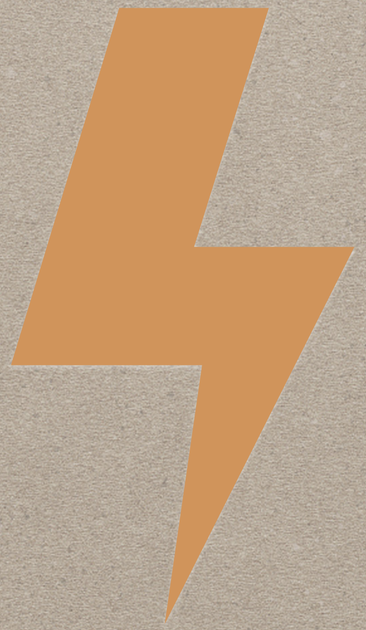
- **Software processes**
- **Formal methods**
- **Modeling**
- **Architecture**



- Work on a real problem
- Assume as little as possible
- Keep things simple
- Have a sound model
- Keep on learning
- Keep on moving
- Build prototypes

THINGS I STAYED AWAY FROM

- **Software processes**
- **Formal methods**
- **Modeling**
- **Architecture**



- What is the problem?
- How can you have impact?
- How do you measure *your* impact?

MEASURING IMPACT

- *How do your actions change the world?*
- Society funds research to take *risks that no one else does*
- Research wants you to take *grand challenges* –
  do not sweat the small stuff; work on the grand stuff
- Saarland University and CISPA expected me to do exactly that
- Worked!

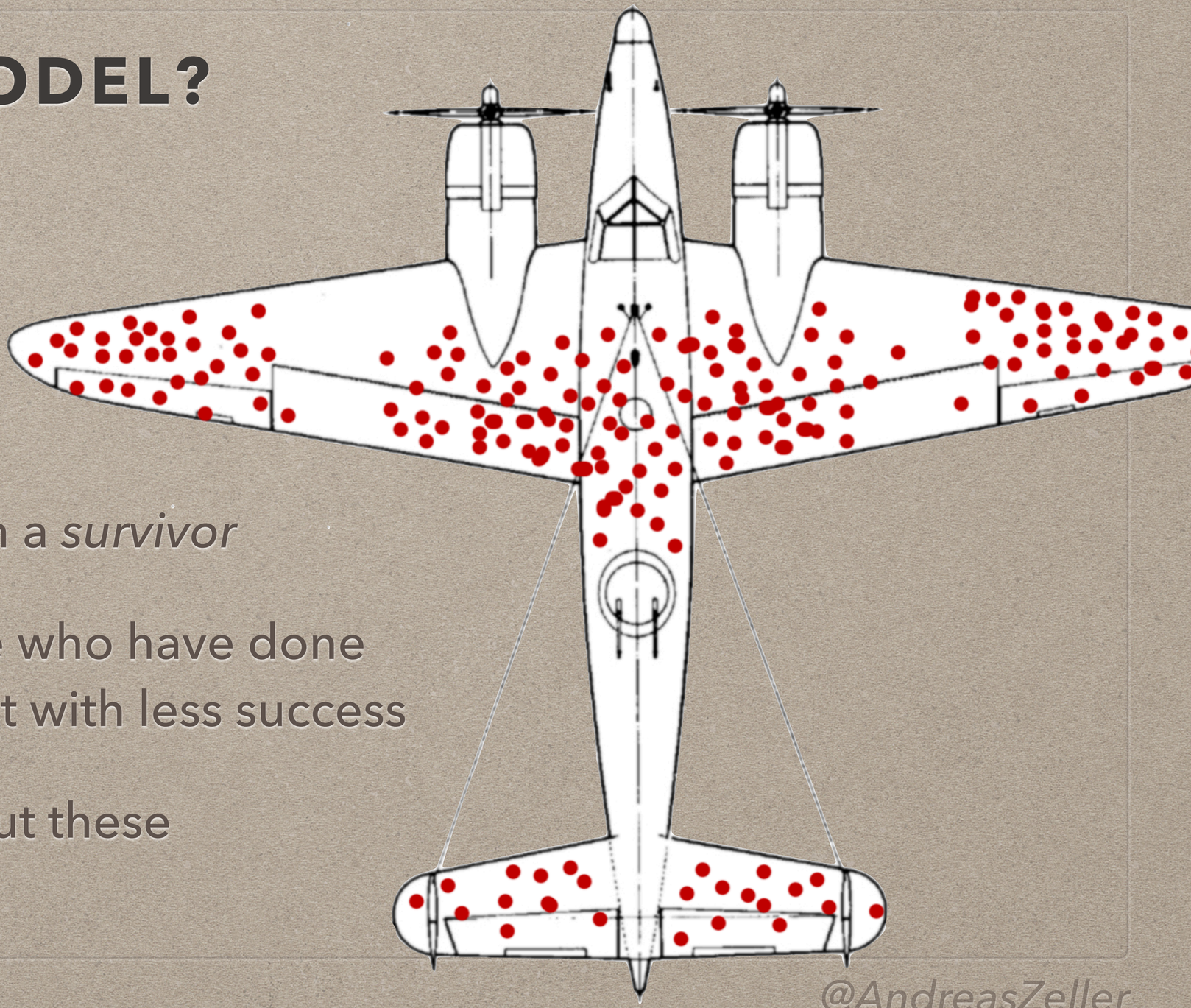
– choose your place wisely

MEASURING IMPACT

- You want to be known for *your* tool, *your* algorithm, *your* book
- **You will *not* be remembered for doing well in a metric**

– please cite this frequently

AM I A ROLE MODEL?



- First and foremost, I am a *survivor*
- There are many people who have done the same or better – but with less success
- We know too little about these

YOUR WAYS TO HAVE IMPACT

IMPACT AS A RESEARCHER

- Society funds research to take *risks that no one else does*
- Research is *risky by construction* –
  you should expect to fail, and fail again
- Tenure is meant to allow you to take arbitrarily *grand challenges* –
  so work on the grand stuff
- If you lack resources, try smarter and harder

IMPACT AS A TEACHER

- Teaching can be a great way to multiply your message
- Not only focus on teaching the standards, *but also your research*
- Teaching your research helps to propagate it and make it accessible
- Engage students on topics dear to you

IMPACT WITH INDUSTRY

- *Do* work with industry to find problems and frame your work
- Do *not* work with industry to solve (their) concrete problems
- Your role as researcher is more than a cheap consulting tool
- Many “research” funding schemes are there to *subsidize* industry

IMPACT THROUGH TOOLS

- Getting your technique out as a tool is a great way to have impact!
- Also allows to check *what actual users need* (and if they exist)
- A tool can have far more impact than a paper
- Funding agencies and hiring committees begin to realize this

IMPACT AS FOUNDER

- Creating a company out of your research can be great fun!
- Allows you to push your research and ideas into practice
- Again, shows you what the market wants (and what not)
- Plenty of monetary and consultancy support available

IMPACT AS MENTOR

- Working with advanced students (MSc, PhD, PostDoc) can be the most satisfying part of your job
- The variety of SE research needs *universal problem solving skills*
- Find such skills besides good grades

A GREAT ENVIRONMENT

- My university (Saarland / Saarbrücken) hired me for a tenured position although I was the candidate with the *fewest publications*
- But they liked the papers, so they hired me
- No pressure or incentives on papers, citations, funding, etc.
- One single expectation: *long-term impact*
- Worked.

ON IMPACT IN SOFTWARE ENGINEERING RESEARCH

ANDREAS ZELLER, CISPA HELMHOLTZ CENTER FOR IT SECURITY

@AndreasZeller

ON IMPACT IN SOFTWARE ENGINEERING RESEARCH

ANDREAS ZELLER, CISPA HELMHOLTZ CENTER FOR IT SECURITY

- **Work on a real problem** – "real" as in "real world", not "real papers"
- **Assume as little as possible** – make things fit into real processes
- **Keep things simple** – complexity impresses, but prevents impact
- **Have a sound model** – causality, retrieval, languages, etc etc
- **Keep on learning** – NLP, statistics, machine learning
- **Keep on moving** – Security starts with SE
- **Build prototypes** – Get your algorithms right first