ON IMPACT IN SOFTWARE ENGINEERING RESEARCH ANDREAS ZELLER, CISPA HELMHOLTZ CENTER FOR IT SECURITY

DAGSTUHL WORKSHOP "SE FORSCHUNGSMETHODENTRAINING" MARCH 2, 2020





ANDREAS ZELLER: KEY FACTS

- PhD in 1997 on Configuration Management with Feature Logic
- 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

Since 2001 in Saarbrücken, Germany (Saarland University + CISPA)



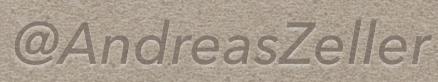


ANDREAS ZELLER: KEY FACTS

 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

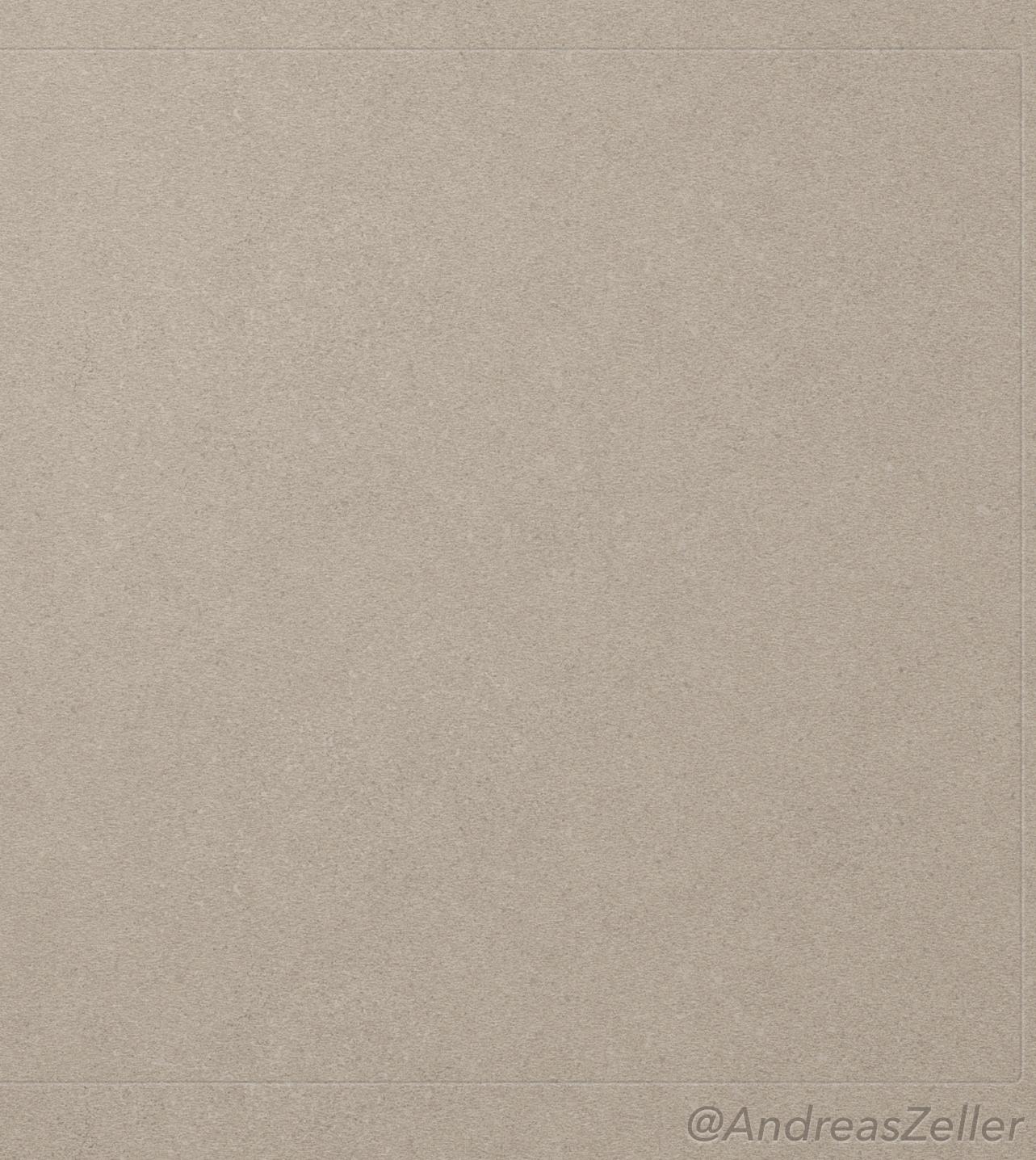
• Since 2019, Faculty at CISPA Helmholtz Center for Information Security

- 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?

- again and again?
- Usefulness can someone make money with it?
- Innovation is the *delta* in any of these metrics

• Intellectual challenge - was it hard, or could anyone have done this? • Elegance - is your research specific to a context, or can it be reused





IMPACT OUTSIDE OF SE

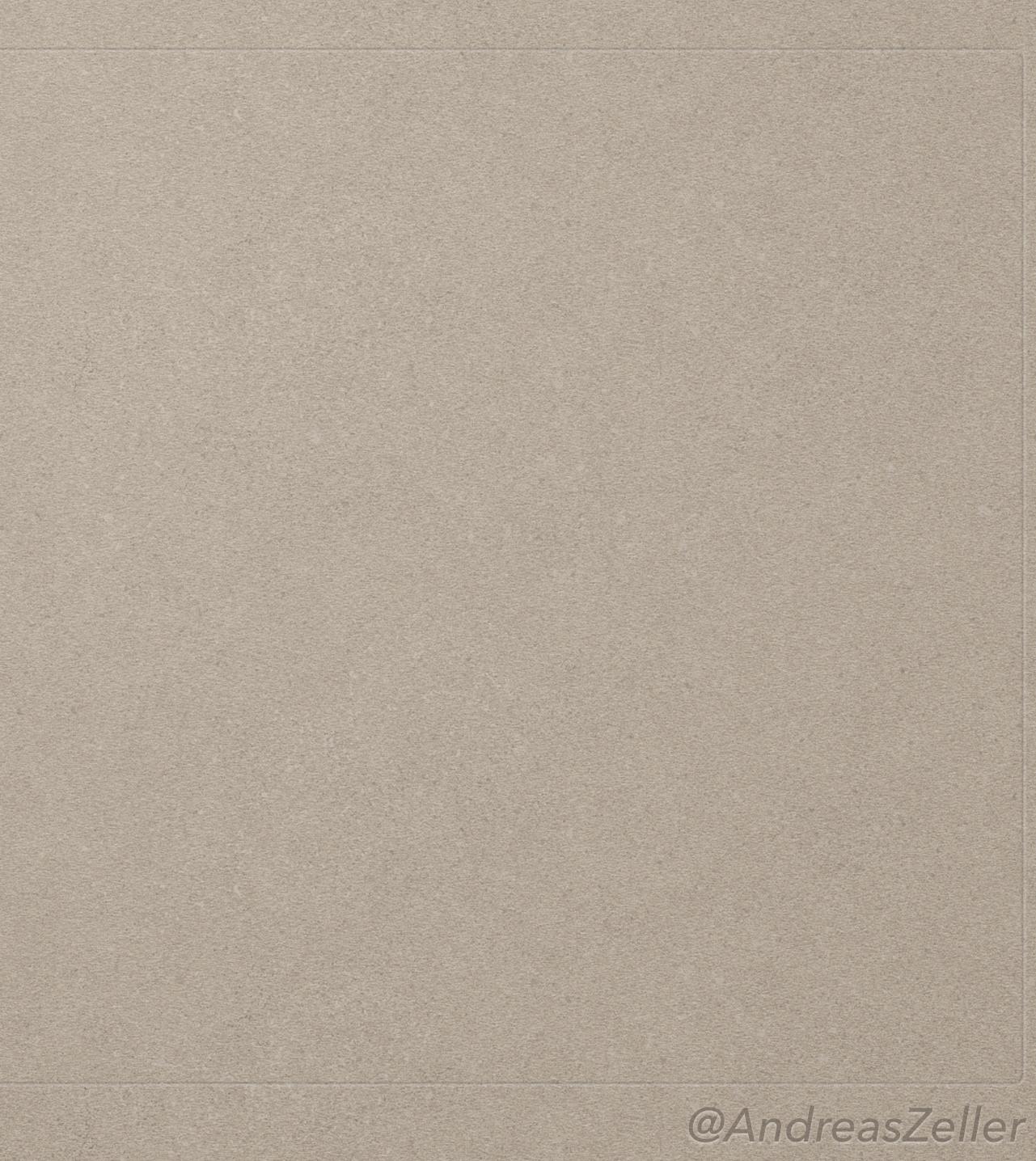
- Programming Languages folks miss the intellectual challenge
- Formal Methods folks miss elegance and challenge
- Industry folks miss usefulness and applicability
- Far too often, we recluse in our private bubbles







MY PATH TO IMPACT





MY PATH TO IMPACT

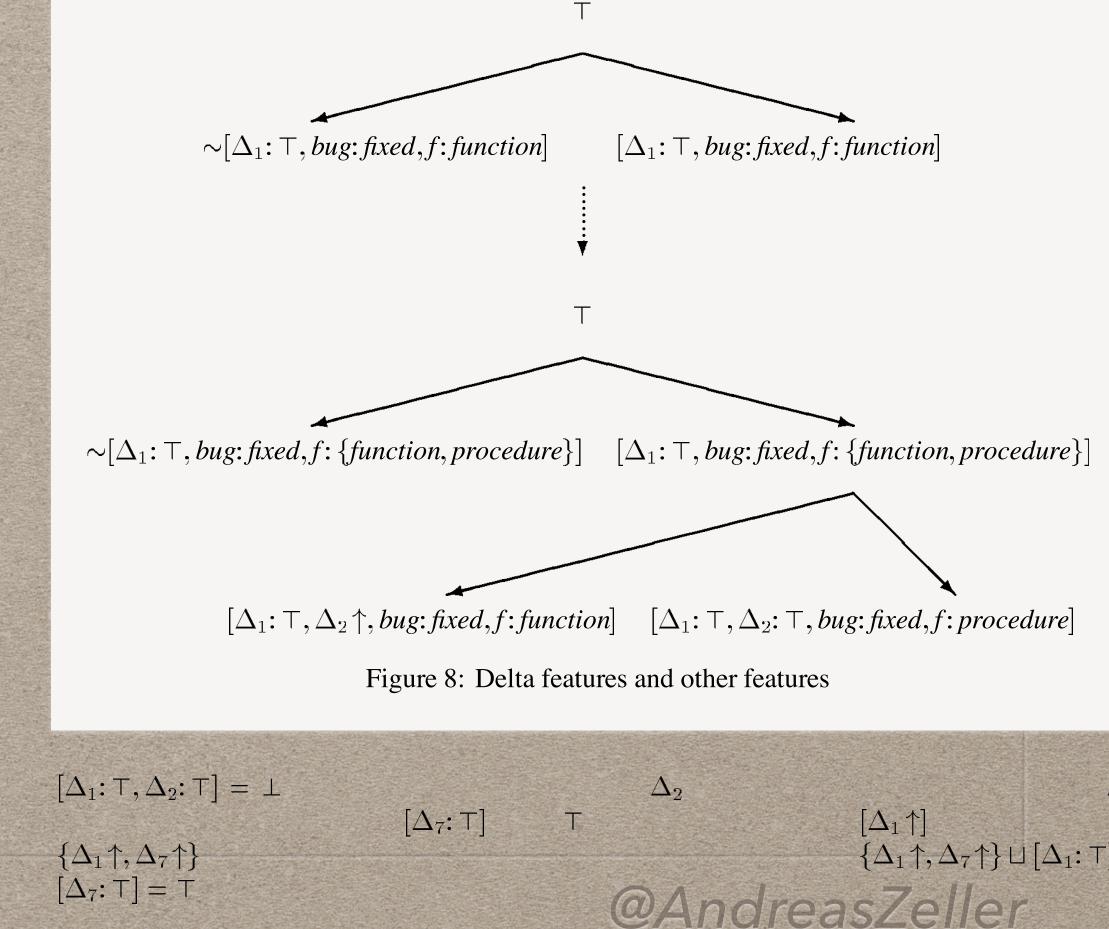
Life can only be understood backwards; but it must be lived forwards (Søren Kierkegaard)





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"





FEATURE LOGIC: LESSONS LEARNED



- 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

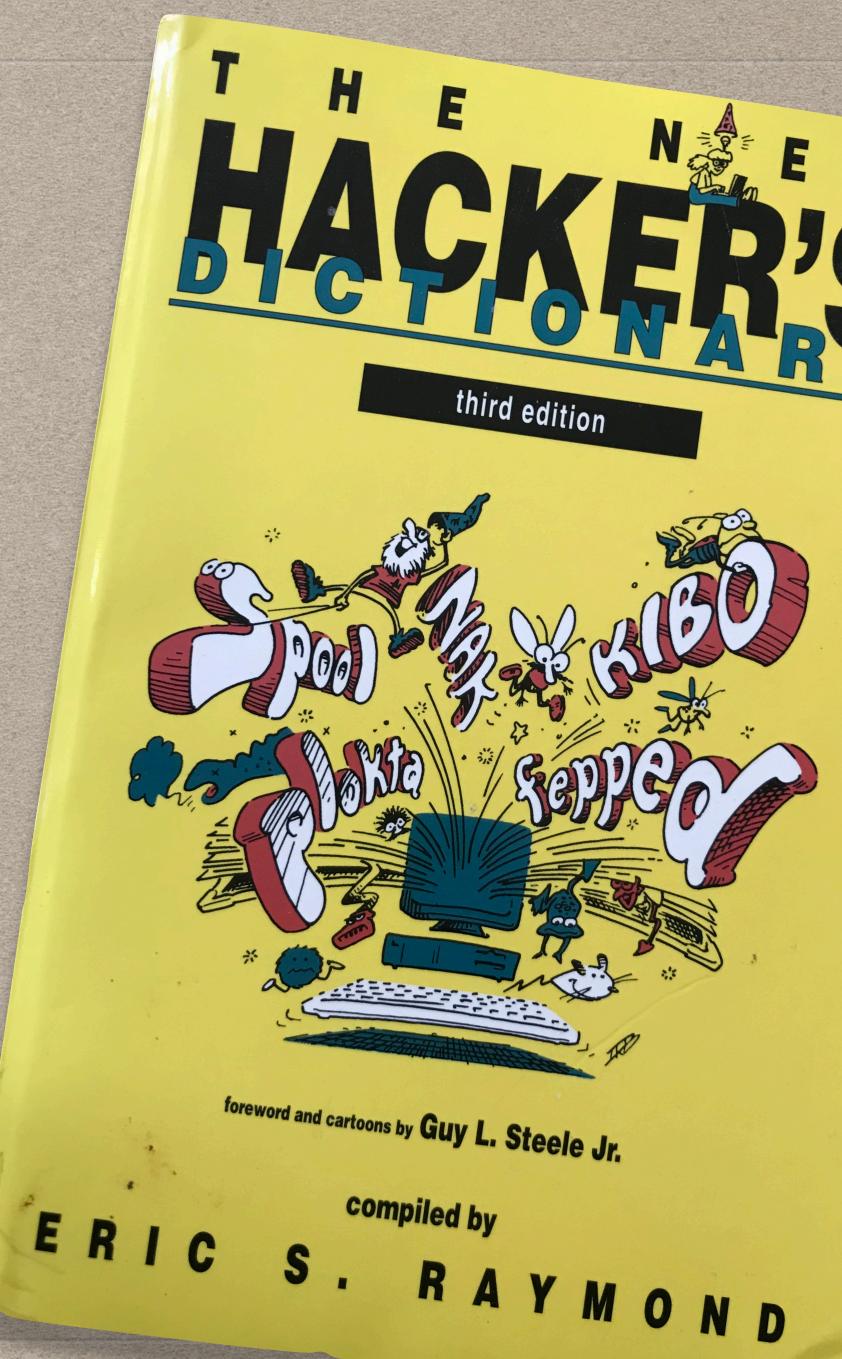




WHAT TO DO AFTER PHD

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





compiled by

WAITON



DDD (1994-1999)

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

🍇 斗 DDI	D: /public/sour	ce/programmi	ing/ddd-3.2/0	ddd/cxxtest	t.C		
<u>F</u> ile <u>E</u> dit	<u>V</u> iew <u>P</u> rog	iram <u>C</u> ommai	nds S <u>t</u> atus	Source	Data		
0: list->	self[⊽ 👸 Lookup	Find« Break	QQ ? Watch Print	Disp* Pi	ot Hide	
1: list (List *) 0x804df80	- V - 50	alue = 85 alte = 0x80 axt = 0x80	14ar80	r ext val sel nex	f = 0:	
list-	->next ->next—>next ->next—>next			t(a_globa t(a_globa			
<pre> weight for the second se</pre>							
}	ǎ DDD Tip (of the Day #5					
// Test void lis { list }		f you made a r recent debuggi	· · ·				
// void ref { date dele	Close		Pre	v Tip		1	
date <u>-</u> }	_pc,,						
(gdb) gra (gdb) ĭ	aph display	*(list—>next	:->next->se	lf) depen	dent on 4		
⊥ list = (Li	st *) 0x8040	df80					
				6	1		

@AndreasZeller



DDD (1994-1999)

- DDD was among the first dev to with a "professional" GUI
- Downloaded by the tens of tho
- Adopted as a GNU project:
 Street credibility with develope
- Impact through usefulness

	A DDD: /public/source/programming/ddd-3.2/ddd/cxxtest.C
	<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>P</u> rogram <u>C</u> ommands S <u>t</u> atus <u>S</u> ource <u>D</u> ata
	0: list->self[v v to contact of the set of
	1: list *() value = 85 (List *) 0x804df80 self next next 0x804df80 next next next next
ools	list->next = new List(a_global + start++); list->next->next = new List(a_global + start++); list->next->next = list;
usands	<pre> weight of the second sec</pre>
	<pre>// Test void lis { list } </pre> If you made a mistake, try Edit→Undo. This will undo the most recent debugger command and redisplay the previous program s
rs	//void ref { date dele date
	(gdb) graph display *(list—>next—>self) dependent on 4 (gdb) [
	▲ list = (List *) 0x804df80
	@AndreasZeller



DDD: LESSONS LEARNED

 Work on a real problem Keep things simple



- "real" as in "real world", not "real papers"

• Assume as little as possible - make things fit into real processes

- 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

Test Case Reduction - 1. manual reduction - 2. delta debugging





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

$dd(c_{\checkmark}, c_{\intercal}) = dd'(c_{\checkmark}, c_{\intercal}, 2)$

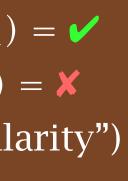
$dd'(c'_{\checkmark},c'_{\checkmark},n) =$

 $(\mathcal{C}'_{\mathbf{v}},\mathcal{C}'_{\mathbf{x}})$ $dd'(c'_{\mathbf{x}} \setminus \Delta_i, c'_{\mathbf{x}}, 2)$ $dd'(c'_{\checkmark},c'_{\checkmark}\cup\Delta_i,2)$ $dd'(c'_{\checkmark} \cup \Delta_i, c'_{\bigstar}, \max(n-1,2))$ $dd'(c'_{\checkmark}, c'_{\ast} \setminus \Delta_i, \max(n-1, 2))$ $dd'(c'_{\checkmark},c'_{\ast},\min(2n,|\Delta|))$ $(C'_{\checkmark}, C'_{\times})$

if $|\Delta| = 1$ if $\exists i \in \{1..n\} \cdot test(c'_{\mathbf{x}} \setminus \Delta_i) = \checkmark$ if $\exists i \in \{1..n\} \cdot test(c' \cup \Delta_i) = \mathbf{X}$ else if $\exists i \in \{1..n\} \cdot test(c' \cup \Delta_i) = \checkmark$ else if $\exists i \in \{1..n\} \cdot test(c'_{\star} \setminus \Delta_i) = \mathbf{X}$ else if $n < |\Delta|$ ("increase granularity") otherwise







DELTA DEBUGGING: LESSONS LEARNED

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

- Why debug? We build correct software

- Version control? tests? Never heard of it

- 25 lines of Python is probably excessive

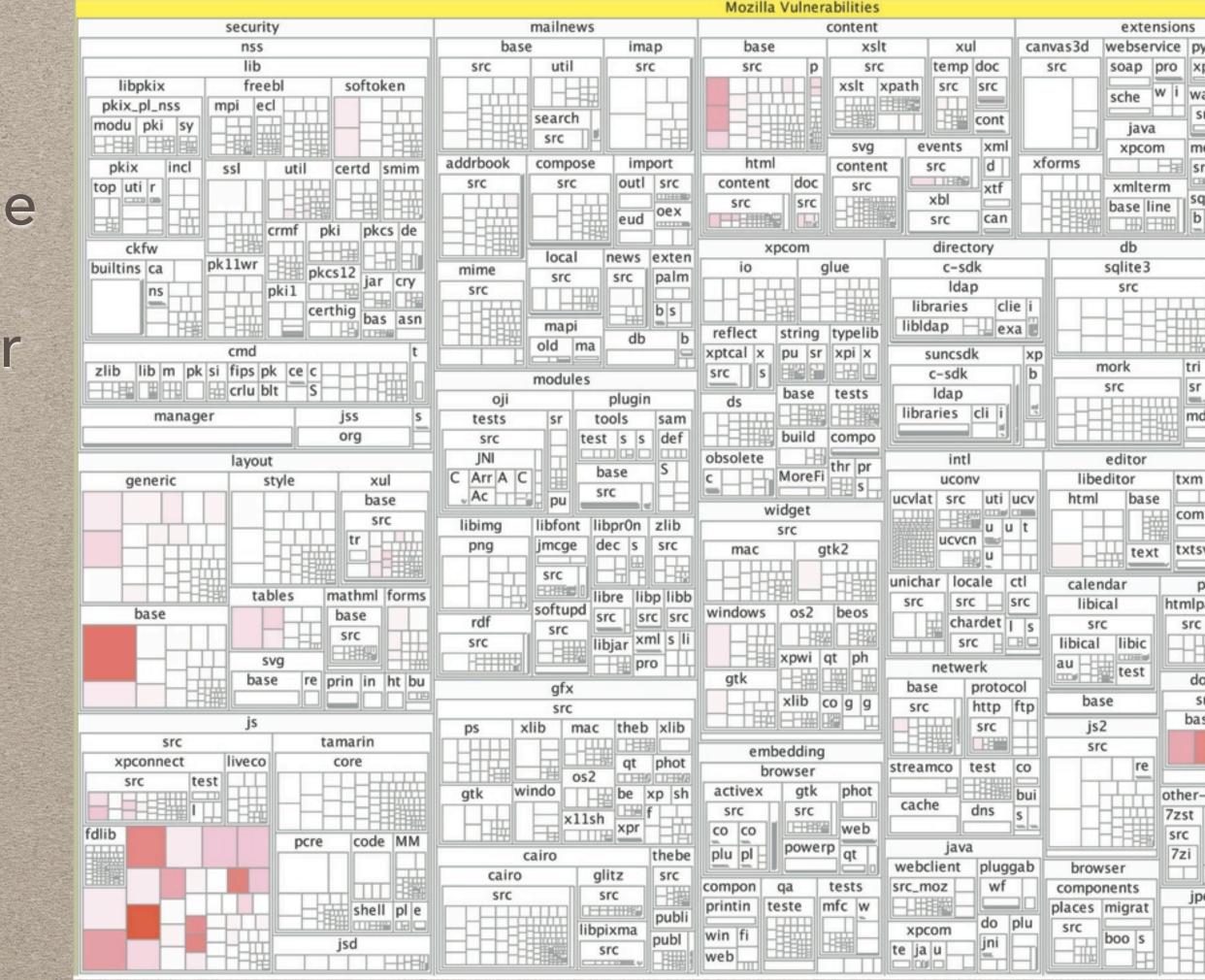
- DD was my version model reborn





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



@AndreasZeller



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...

Contraction of the					660 220 22		Selection of the			
		Mozilla Vulnerabilities								
	5	mailnews		content						
	nss		base imap			base xslt				canvas
		lib	src u	ıtil sr	c	src p	src			src
	libpkix	freebl softoken					xslt x	cpath src	src	1
122	pkix_pl_nss mp	pi ecl	Sez	rch					cont	
	modu pki sy		Store St							
							svg	events	xml	
	pkix incl se	sl util certd smim			port	html	content			xform
	top uti r		src	src outl	Src	content doc	src	xbl	xtf	- Herber
				eud	oex	STC STC			can	-199
		crmf pki pkcs de		euu				src		
Sto	ckfw			cal news	exten	xpcom		direct		
	builtins ca pk11wr pkcs12		mime	rc src	palm	io	glue	C-SC	dk	
48.87	ns pkil Jar Cry		src		pann			Ida		
		certhig bas asn			bs		一書譜	libraries	clie	i
		bas asn		api db	b	reflect string	typelib	libldap	exa	ë
1000	cmd t		olo	ma		xptcal x pu sr	xpi x	suncsd	lk b	xp
No. al	zlib lib m pk si fips pk ce c		le i i le			src s		c-sdk		b
8968	crlu blt S		oji plugin		hase	tests	Idap		nIII_	
No.			oji tests s	r tools	sam	ds Dase	CETERRE	· · · · · · · · · · · · · · · · · · ·	cli i	
	manager jss s org		src	test s s	_	build	compo			
		JNI			obsolete	compo	intl			
52021 F		layout	C Arr A C base S		c MoreEi thr pr		uconv		_	
1000	generic	style xul		src	비미		S-			
a Cal		base		u sie		widget				
N N REE		Src		font libpr0n	zlib	Src			💷 u u t	
matrix.			png jm	cge dec s	src	mac g	tk2	ucvc		
- 20			sr		日報				ter .	كالك
ACATO		tables mathml forms		The second secon	p libb		「中田田」		cale ctl	_
1	base	base	sof	tune -	rc src	windows os2	beos	src sr		
		src src	rdf	rc 💷	nl s li				ardet	
NU IN			src	IIII IIII =		xpwi	qt ph	sr	rc Hall	
No.		319		pr	0	gtk	inderroll	netwe	ark	au
Sec.03		base re prin in ht bu	9	Jfx			田田	base	protocol	
				src			cogg	src	http ftp	p 🔔
WUDTH-		js tamaria	ps xlib	mac theb	xlib		-1388 -1-		src	
	Src II	tamarin iveco core			2 phot	embedding				네
1000	src test	iveco core		os2 qt	phot	browser	· · · · · · · · ·		test co	
	sic lest		gtk windo		xp sh	activex gtk	phot	LE	bu	ui
				x11sh		src src			dns s	
	fdlib pcre code MM		cairo thebe		co co web plu pl powerp qt				-	
il and							java			
TRA ST			cairo	glitz	src			webclient	pluggat	
KL2X			src	src	一躍	compon qa	tests	src_moz	wf	co
N.V.N		shell pl e			publi	printin teste	mfc w		do plu	pla
10203				libpixma	publ	win fi	THE	xpcom	jni	1 51
		jsd		Src		web		te ja u		
			(c							

@AndreasZeller





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



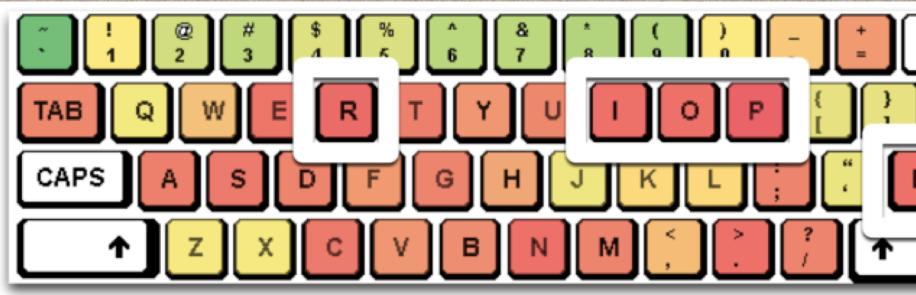
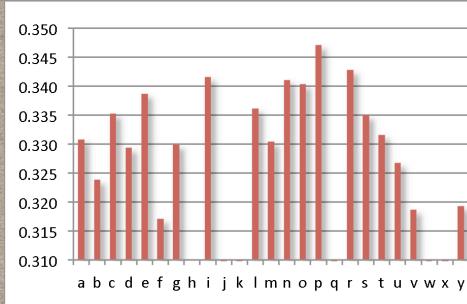


Figure 2: Color-coding keys by their defect correlation; (red = strong). The five strongest correlations are highlighted.







MINING SOFTWARE REPOSITORIES: LESSONS LEARNED

• Work on a real problem • Assume as little as possible • Keep things simple Have a sound model Keep on learning

- Empirical research is core field of SE

- simple parsers for multiple languages

- essence of 2004 paper is one line of SQL

- retrieval, precision, recall, etc, etc

- statistics, data mining, machine learning





FUZZING AND TEST GENERATION (2012-)

- In 2012, ran LangFuzz: a grammarbased 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]+/



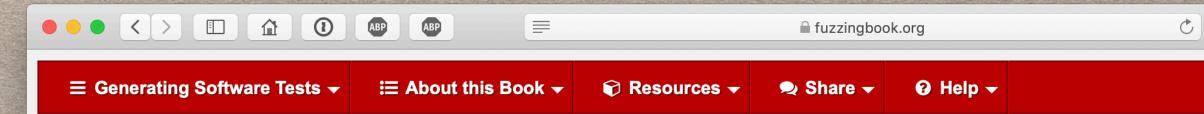




@AndreasZeller

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



Generating Software Tests

Breaking Software for Fun and Profit

by Andreas Zeller, Rahul Gopinath, Marcel Böhme, Gordon Fraser, and Christian Holler

About this Book

Welcome to "Generating Software Tests"! Software has bugs, and catching bugs can involve lots of effort. This book add automating software testing, specifically by generating tests automatically. Recent years have seen the development of nove dramatic improvements in test generation and software testing. They now are mature enough to be assembled in a book code.

from fuzzingbook_utils import YouTubeVideo YouTubeVideo("w4u5gCgPlmg")



Generating Software Tests

Generating Software Tests

Breaking Software for Fun and Profit





FUZZING AND TESTING: LESSONS LEARNED

- 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

- Yes, bugs do exist

- Toss program into black box

- Grammar-based producers

- Grammars and languages

- Constraint solving, search-based testing

- Security starts with SE

- Get your algorithms right first

@AndreasZeller



MORE THINGS I DID (AND DO!)

• Automatic repair Automatic parallelization Automatic website testing

- Wesley Weimer beat us to it

- Struggled with complexity

- 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



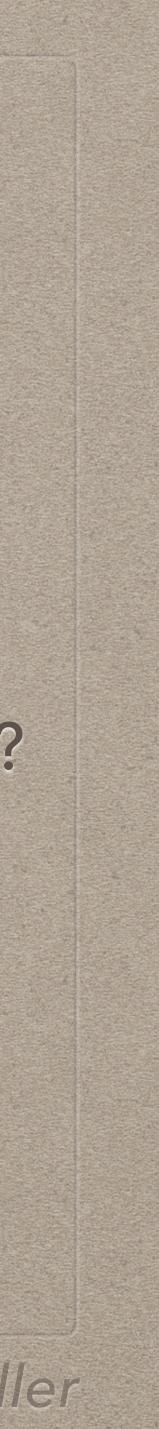


THINGS I STAYED AWAY FROM

- 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

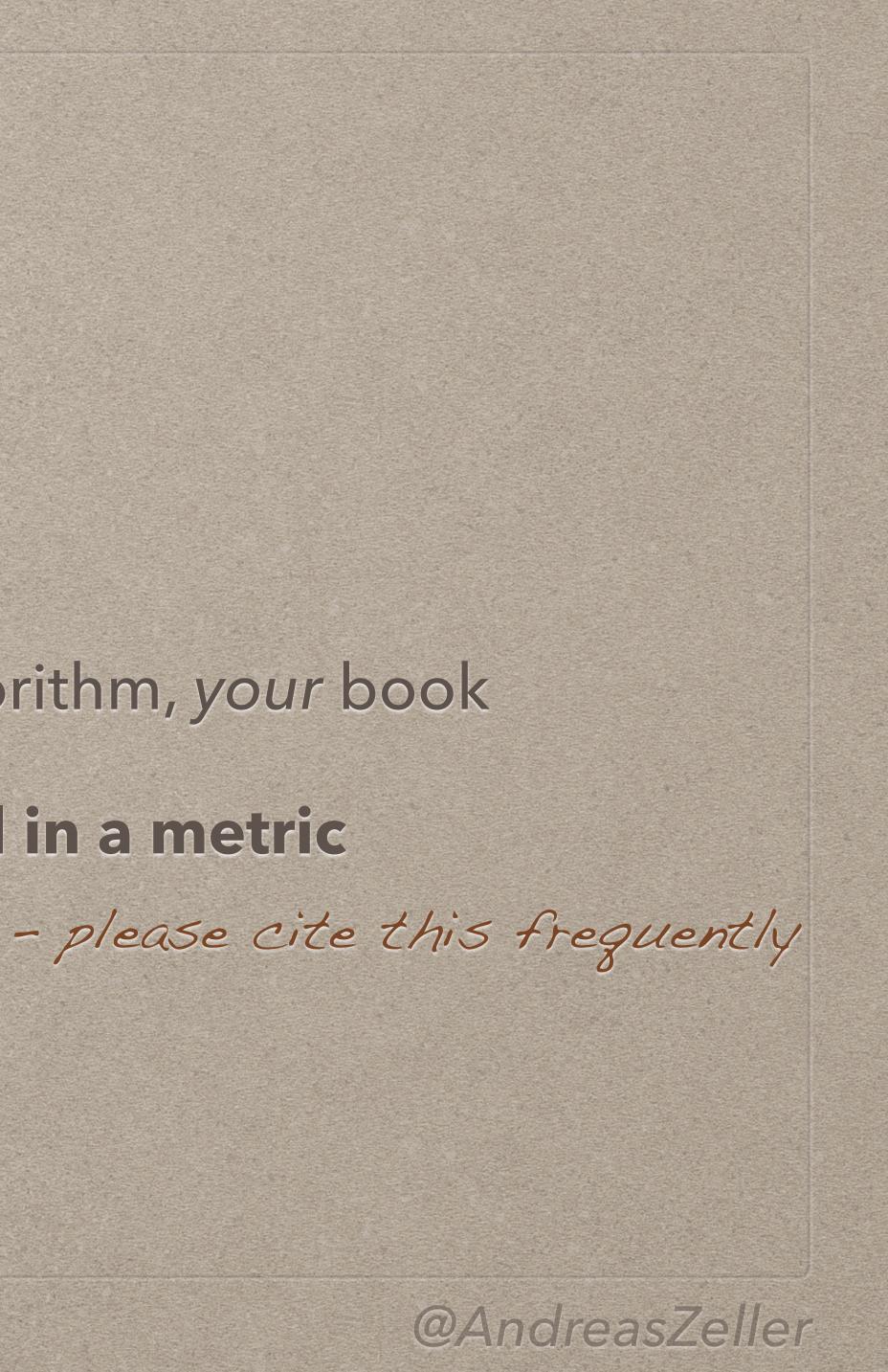
@AndreasZeller



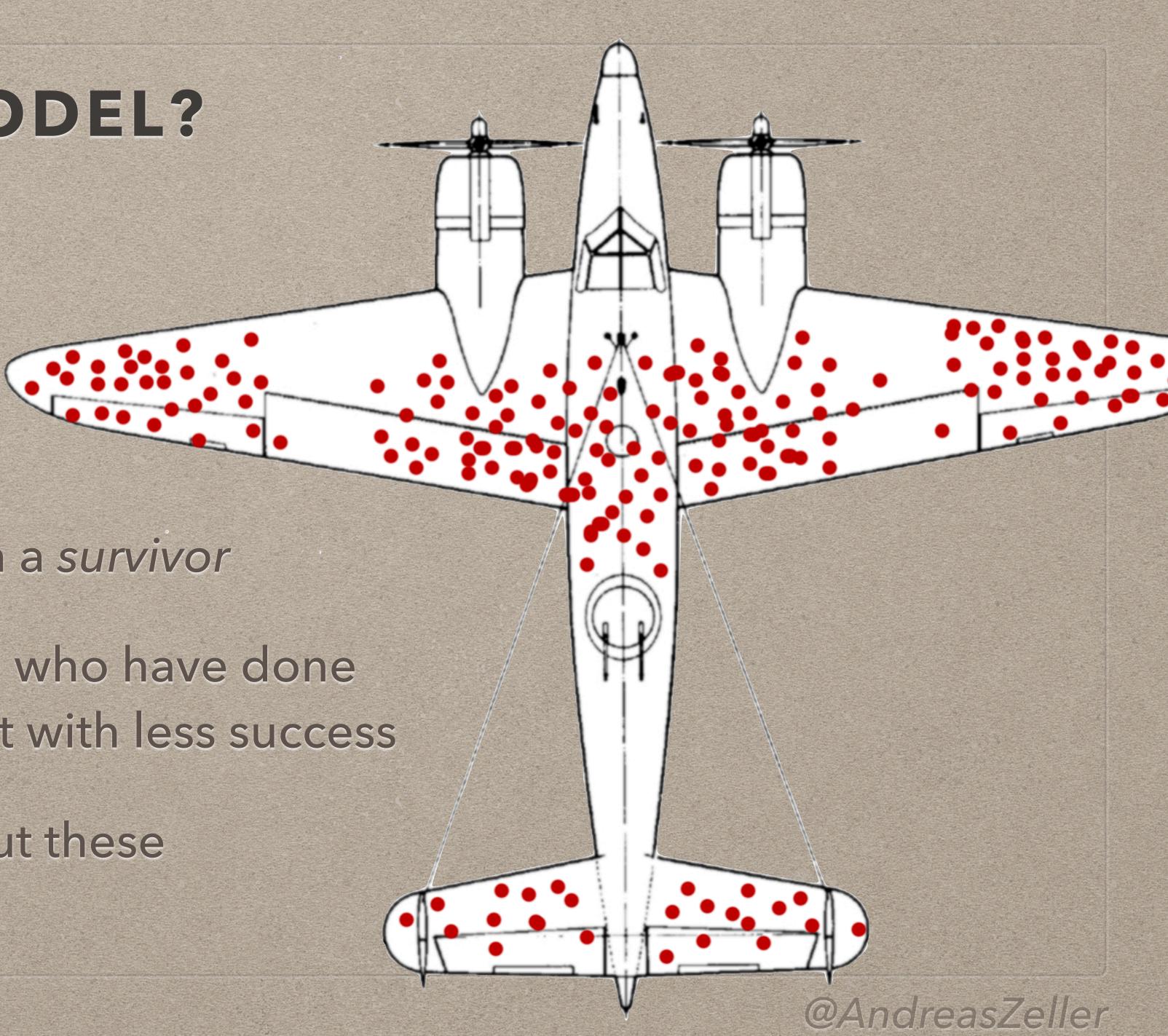
MEASURING IMPACT

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





AMIA 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
- so work on the grand stuff
- If you lack resources, try smarter and harder



Tenure is meant to allow you to take arbitrarily grand challenges -





IMPACT AS A TEACHER

- Teaching can be a great way to multiply your message

- Engage students on topics dear to you



• Not only focus on teaching the standards, but also your research • Teaching your research helps to propagate it and make it accessible

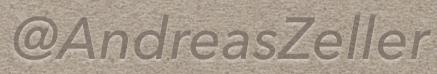




IMPACT WITH INDUSTRY

- 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

- 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

• Getting your technique out as a tool is a great way to have impact!





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

- most satisfying part of your job
- The variety of SE research needs universal problem solving skills
- Find such skills besides good grades

Working with advanced students (MSc, PhD, PostDoc) can be the

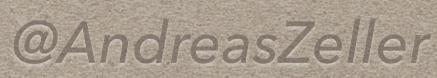




A GREAT ENVIRONMENT

- 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.

My university (Saarland / Saarbrücken) hired me for a tenured position





ON IMPACT IN SOFTWARE ENGINEERING RESEARCH ANDREAS ZELLER, CISPA HELMHOLTZ CENTER FOR IT SECURITY





ON IMPACT IN SOFTWARE ENGINEERING RESEARCH ANDREAS ZELLER, CISPA HELMHOLTZ CENTER FOR IT SECURITY Work on a real problem • Assume as little as possible - make things fit into real processes Keep things simple 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

- "real" as in "real world", not "real papers"

- complexity impresses, but prevents impact

@AndreasZeller

