#### ON IMPACT IN SOFTWARE ENGINEERING RESEARCH

ANDREAS ZELLER, CISPA / SAARLAND UNIVERSITY

ICSE NEW FACULTY SYMPOSIUM GÖTEBORG, MAY 29, 2018

@AndreasZeller

## **ANDREAS ZELLER: KEY FACTS**

- PhD in 1997 on Configuration Management with Feature Logic
- Since 2001 professor at Saarland Informatics Campus (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 on Friday

@AndreasZeller

### 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

### VARYING PERSPECTIVES

- Programming Languages folks miss the intellectual challenge
- Formal Methods folks miss elegance and challenge
- Industry folks miss usefulness and applicability

### WHAT MAKES IMPACTFUL RESEARCH?

• How did your work make the world a better place?

### **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"



 $\Delta_1$ 

 $[\Delta_7:\mathsf{T}] = \mathsf{T}$ 

### FEATURE LOGIC: LESSONS LEARNED

- You can get plenty of papers accepted
  - even if you miss the problem
  - even if you do not evaluate
- "Modeling for the sake of modeling"
- Enabled much of my later work, though

## DDD (1994-1999)

- During PhD, programmed a lot
- Debugging was hard!
- Built the DDD debugger GUI with my student
   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

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

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

 $\begin{cases} (c'_{v}, c'_{x}) \\ dd'(c'_{x} \setminus \Delta_{i}, c'_{x}, 2) \\ dd'(c'_{v}, c'_{v} \cup \Delta_{i}, 2) \\ dd'(c'_{v}, c'_{v} \cup \Delta_{i}, c'_{x}, \max(n - 1, 2)) \\ dd'(c'_{v}, c'_{x} \setminus \Delta_{i}, \max(n - 1, 2)) \\ dd'(c'_{v}, c'_{x}, \min(2n, |\Delta|)) \\ (c'_{v}, c'_{x}) \end{cases}$ 

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

@AndreasZeller

### 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

					Mozilla	Vulner	abilities						
	mailnews				concent								
	bas	base imap		base		lux fiex		cul 👘	car	ñ			
1b			SPC .	U U O I	1	K	SIC	ø	515	tem	p doc		si
libokia freebl softoken			1111	1111	8		TIT.	THE	issit b	with sto	SFC	-	-
ib libgkis freebl softsken pkix_pl_res mpi eel modu pki sv pkix incl ssl util certd smim top util r top util			1.1.1.1.1.	- F			di F	I HAR	田陽一〇〇	( and			
modu pki sy	1.1.1.1								1 881111				
1111 用始的	湖田	は日本にしては意	L. H1191	SIC	J	-111			svg	events	lens 4	line,	-
pkix incl s	st v	dil cetti smim	addrbook	compose	1.07	port	html	2.2	content	t src	d	xf	0f
top uti r	TIEL	HALF LINE AND	\$FC	SIC	out	SPC	content	doc	SEC	1.08	E. ATT	11	5
TTLOUE .	115	13월 14일 138	1.1.1.1.1	1.1.1.1	Him	- DPX	SITE	2.4		x0l		H	Ę
	crm	f oki lokoside	13743		eva	Junpied		E DA	- Gran	5/C	CAR		E
ckfw	日間にす			the second second	- Contraction	البلبية	10	200/11	St 10.5	dire	ctory		
builtins ca pk1	lwr	ploce 12	mime	10(4)	news-	exten	io	6	due	- C-	sdk.		
12	pk:	h jar cry	sni	500	576	paim	144		1-11-11	L. M	lan .		
	111	certhig Per Th	- Hitest	LLH	1	b(s	1148		- HIM	librarie	s cli	e i 1	
Das aso			1.11980	mapi	1	Calify of	affect	string	typelb	libidap	- ex	3 2	
	cmd	1	LTT TOTAL	old ma	- 00		notcal in:	04 57	beni be	- MINC	udk	30	
zib lib m ok si Nos ok ce c				Sector Printed	47		we list	10000	(CHEC)	6.50	k	161	E
FITH STATISTICS	criu bit	***3		modules			See UC	hase	Dests	idar	1	di fi d	
FILLION	the state of the s	An In		-	pings	1	ds	(1259.3)	CHENG	Ibraries	left fill	16	
care/pr		10 5	tess	91.	10015	Lain	1.日田田	Build	Contrana I		100	m	
-			575	- 08	10 1 1	Oer	checkers	111	compo			8	ß
5	layout		Class S			the pr			19761			H	
geneec	4546	e xul	C ATA L		316	111	2	soreri	1004	UCI INC.	onv r land	-	÷
		0410	and stated as	s. pu	-	1.11	W	idget	1000	ACCOUNT OF	Contraction of the	inter a	
		100	Libimg	(abfont)	libpr0n	2lib	1	SPC	1	1	U.M. N	(T.)	
			phg	jmcge-	dec s	346	mac	9	tk2	Electron of the	UDIN -	1.1	F
			h hate	tre	니면	114	1.444	4	1111	Contraction of the	1961-	111	
	table	a machini forma		Scenal P	Der Di	he lbh	1-43	페	-(11篇)	unichar li	ocare o	100	
base		bue	1,04	Softupd	100 10	or tor	windows	052	beos 1	SIC 1	HC I	eve	
		11 11	101	- 5/10	HC P	al a la	1 1.11	0.114	H LHM	100	hardet	31	
	_	-1e 1200 - 11	SVC	1.2110	libjar *	101 2 10		ALCONG.	or inh	ETTER.	RAC PE	11-1	
	Sig	and and the first first	111114	P-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1		ro		CLUG	State Land	neti	verk		2
一日間	Core .	the prise last list but		- gex			1111	with 1	1202114	base	protos	0	P
- I III Prove and a second second			arc				Contract	0 9 9	SPC	NOLD	ftp.	-	
	p	tion in	ps ja	dib max	c theb	dik t	1.1.005		98. LL	日田田	304		
and and a	in and in the second	Laiterin		1111	41.41	Sec.	emb	redding		12111116	1	:::::	-
APC Text	Neco	Care .	1 FHG	110 012	100	peor.	54	owser		7167400	test	80	Ľ
SPC (REST)	1118		grk w	rindo III.	ilibe	xp sh	REALER	gtk.	phot	in the second se	注制限	bui	
「日子子」	-   日田   -			+114	1000	f L.L.	21C	src .	1	cache	des	5	h
fdib	The second	ness code MM	<del></del>	-10.11	*pr	1-111	10 00	CERES.	mep	_			
	1	pore code para	Givo thebe		plu pl - powerp qt			java					
Contract of the local division of the local		1.1.1	cairo	1	olitz	ME	S.S. Inth	-	Contraction of the	vebclient	plug	ap	
and the second s	The second	T CT CLUDER	SPE		src	-HH	campon	da	cests	9C.mo2	wi	1.1	2
	-	shell ple	1.1.1.1.1	111110	(FIII)	publi	bound D	esce	THE W	口田市	ide l	Since of	P
	in.			000	picenta.	publ	wh fi		ad -	xpcom	100	20	
and the second se		1m			STC .		and the second sec			1010 100 100	- 10 million		

## 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 1200+ citations so far
- Impact at Microsoft, Google, SAP...

					Mozilla Vulne	rabilities			
security	mailnews			content					
755		base		imap	base	833	t ) 3	nd i i	cariva
10	SPC .	001	3/6	SIC 0	511	tem	p doc	57	
libpkis freeb pkix_pl_rss mpi eci modu pki sy	softblea		search src #			KSIT	Koeth Sto	SFC CONT	
mine and and	odil statut izmim	addrbook	compose	Inogeni	Intent	cietas	t Sec	THAT!	xfor
top uti r	mf pki pkcs de	SIC	SIC	evd oex	content doc src src	SIC.	xbl src	E xtf	-
ckfw Little	그네 - 방법 - 영니	local news exten			XPCOM	dire	ctory		
builtins ca pkilwr	mittig srii	mapi	sec palm bs db b	io affect string	olue Cypelib	c-i M librarie libidap	udk an s Cle lexa	R 1	
end	end t			1	NICAL X DU S	r xpi x	Junca	dk	sp
210 IID III DR SI TIDS DR	modules			246 2 24 26 26	利用目的	C-50	k	D	
TTTE WITH DS/CITE/OR	. oji		plugin	ds Dase	Dests	Idap	1.0.1	1.8	
nanager jis js		tests	UT 1	cols sam	-HIMALISE	机田原	ibranes	0	1410
	575	tes	t s s def	build	compo				
layout		Pd hats S			obsolete	intel			
genesic sty	Ac pu arc			C Morel	uconv ucviat sirc uti ucv				
	170	Idomo	abfont In	bpr0n zlib	widger.		111111(-1)	- U - W	1
	tr	ong	proce of	lec s src	mac	gtk2	U.S.V.	CA MAR	÷.
	IT III III I LINN	8-1-1-12	110	THE HE FIELD		口田田	unichar le	scale c	ti .
Land Land	es machimi forms	1.1113	Softund	bre Rbp Rbb	unders Loc3	Dates 1	5/6 3	rcs	10
	5%	edf src	- 51%	biar aml s li			140	andet	
595	FITTER	CT:UTH:	pro	×pwi	ot ph	nete	otrk	100	
hase	ne pris in ht bu		QTX .	and the second second	44k 12	1.001.00	base	protoco	a (4
- HIRNE	1	arc	1000	xip	co g g	SPC	http 3	tp	
В	ps.	xlib mac	theb slib	1 Hereita	104 11		MC	TH	
JTC	Lamarian .		HHH	TH FLAMM	embeddin	g	1 1 Stiffe	1.108	
spoonrect liveco	care	「开始」	110 012	a or proc	browser		niteruico	test i	00
94 AESI		gtk M	vindo	te xp sh	STC STC	phot	cache	des 1	bui
fdib	man lands hild	一番	-140 110	spr in	40 00 Lttt	a web			ala (
14444	pore pose sam		GiVD Thebe			powerp at			_ E
The second secon	100	cairo	0	itz str	Sm. in Chil	and the second second	vebclient	plugg	ab
		SPE		urc – Hill	compon gà	cests	Sc.mos	W	hill.
	shell ple	LITT	ALC: NO	publi	Testes tests	ALC: W	口田市	do a	p p
	110	EEE	- De	urc publ	wn fi	開始	te ja u	194	

## 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.



@AndreasZeller

### 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

# MINING APP ARCHIVES (2014-)

- How do we know an app does what it should do?
- CHABADA checks for mismatches between description and behavior
- Novel usage of NLP; novel app store mining



# MINING APP ARCHIVES (2014-)

- The ICSE paper of 2014 is among most cited
- CHABADA techniques now adopted by Google and Microsoft
- Most of your mobile apps have gone through such an analysis :-)



@AndreasZeller

### MINING APPS: LESSONS LEARNED

- Work on a real problem Yes, there is malware
- Assume as little as possible Descriptions and APIS
- Keep things simple Standard NLP techniques
- Have a sound model Standard NLP methods
- Keep on learning
- NLP, machine learning, recommendation...
- Keep on moving Security starts with SE

@AndreasZeller

## **MORE THINGS I DID**

Automatic repair

- Wesley Weimer beat us to it

- Automatic parallelization - Struggled with complexity
- Automatic website testing
   Built a company for that
- Structured fuzzing

- Langfuzz found 2000+ browser bugs

Automatic sandboxing

- lots of potential in here

## THINGS I STAYED AWAY FROM

- Symbolic techniques
- 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

### 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!
- Push your research and ideas into practice
- Again, shows you what the market wants (and what not)
- Plenty of support available (money, consultancy)

### **IMPACT AS MENTOR**

- Working with advanced students 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 institution (Saarland University) hired me 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

## **SURVIVOR BIAS**

- Researchers with great impact are the selected few who survived academic selection
- What worked for me will not work for most
- Most of us have to struggle with plenty of bad, misguided, short-term career incentives
- Follow incentives until tenured, then set better ones
- Get lucky!

#### ON IMPACT IN SOFTWARE ENGINEERING RESEARCH

ANDREAS ZELLER, CISPA / SAARLAND UNIVERSITY

### LESSONS LEARNED: ON IMPACT IN SE RESEARCH

 Work on a real problem - possibly bursting your bubble Assume as little as possible - immediate impact on adoption Keep things simple - complexity inhibits impact Have a sound model - tools may fade away, concepts persist Keep on learning - learn new stuff and leverage it Keep on moving - do not stay in your cozy SE corner