

where **user experience** and
software engineering meet

Andrew J. Ko



dub

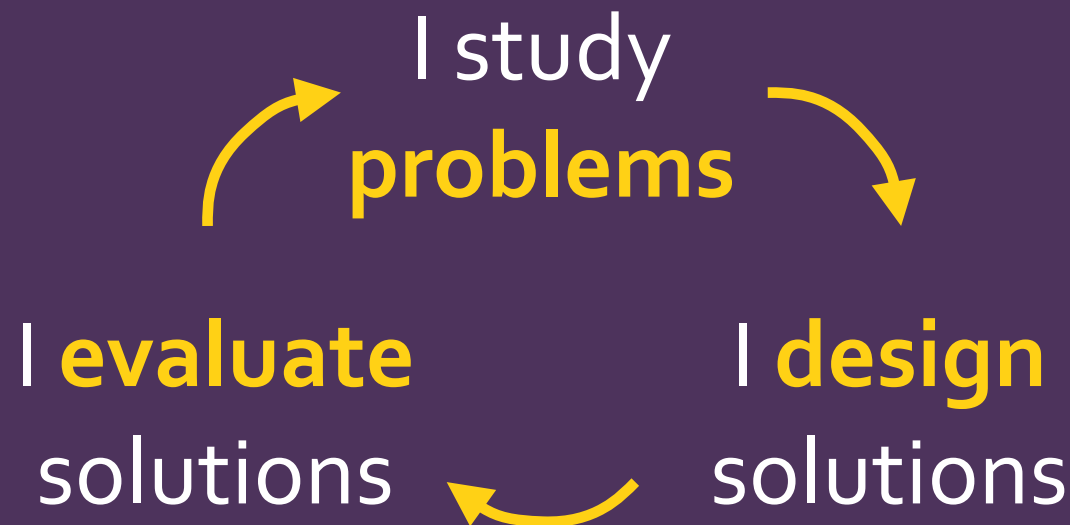
in **method**, I'm an **HCI** researcher

I study
problems

I **evaluate**
solutions

I **design**
solutions

in **method**, I'm an **HCI** researcher



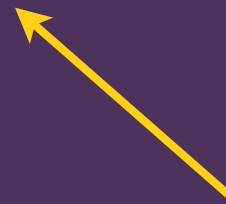
in **topic**, I'm an **SE** researcher

I want to make it **easy** to
create **useful, usable** software
that **empowers**
and **enriches**
users' lives

human-computer interaction



software engineering



me

human-computer interaction



how do we get the
right design?
(the lives of users)



software engineering



how do get the
design right?
(the lives of developers)

user-centered software evolution



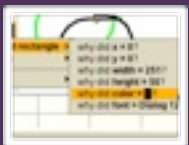
how do we get the
right design right?

talk outline



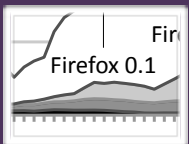
why is software evolution **difficult**?

a study of information needs at Microsoft



how can **tools** help software evolution?

debugging with the Whyline



how can **users** help software evolution?

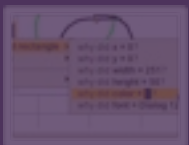
recent work in leveraging the crowd

talk outline



why is software evolution **difficult**?

a study of information needs at Microsoft



how can **tools** help software evolution?

debugging with the Whyline



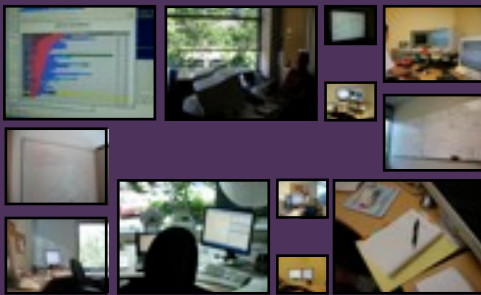
how can **users** help software evolution?

recent work in leveraging the crowd

information needs at Microsoft

with the Human Interactions in Programming group at Microsoft Research

observed **25 hours** of coding and bug fixing, in the role of “new hires”



357 pages of handwritten notes

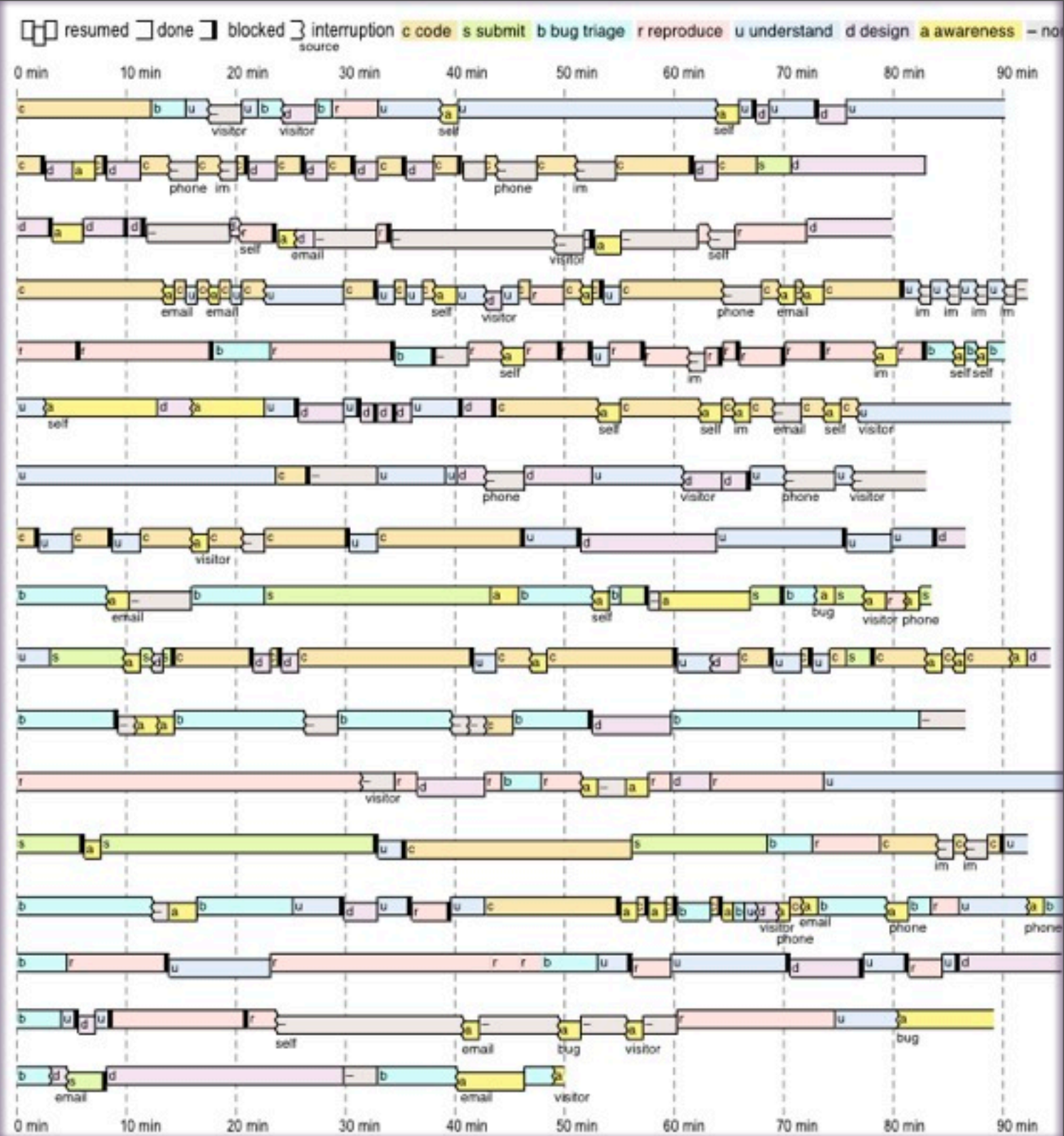


4,231 events in an spreadsheet

Name	Shoulder	Time	Estimated Event	Goal	Status	Tag	Source
Peira	Wider	2:43 PM	P: So her's watching video				
Peira	Wider	2:43 PM	N: You're not taking to me, but on research too, so the better not have to				
Peira	Wider	2:43 PM	P: So the to research did not work most of the time				
Peira	Wider	2:43 PM	N: Both for implementation				
Peira	Wider	2:43 PM	P: The product team will be stuck, and I will be necessary				
Peira	Wider	2:43 PM	N: Oh, so going to send you a new bug				
Peira	Wider	2:43 PM	P: Great, I can't find time				
Peira	Wider	2:43 PM	[Peira leaves, and explains to me how research is used occasionally, but I don't really account for the whole picture]				
Peira	Wider	2:43 PM	P: So that's it				
Peira	Wider	2:44 PM	P: You can get your research group the product build is a waste of ...	FRANCE	DONE		
Peira	Wider	2:44 PM	P: So what do I have here ... bug with crash			RESEARCH	REFERENCE
Peira	Wider	2:44 PM	I don't really know what's standard, but usually, there's a crash, and we can report, but we do have a stack dump, and we load it into the debugger			SITUATIONS	REFERENCE
Peira	Wider	2:45 PM	2:45 PM	But they put the method in the bug report file, so whenever I see crash logs including a CSR method, I immediately send it CSR and say, "hey, look at this"			
Peira	Wider	2:45 PM	P: Is a pretty old task				
Peira	Wider	2:45 PM	Yes, then run these automated tests that are very intensive testing, and they check a build, and I go for 2 weeks, and we get bugs from a very old build that I don't know [Peira leaves] so I can't take it on				
Peira	Wider	2:45 PM	Peira had a meeting at 2:30. Peira has for the time, take a picture of the screen with product build, and he sends me the template he referred to in our interview				
Peira	Wider	3:00 AM	3:00 AM	One of the main tools is Outlook. Great also for code review, like a lot of files, but they've been deemed to be a success. I guess. I very check in whenever a code review, so we're in the flow	FRANCE		
Peira	Wider	3:00 AM	This bug ... I was going to know why I change to the control that we write				
Peira	Wider	3:00 AM	3:00 AM	Yes, you have found and fixed, and you're in the control that we write			

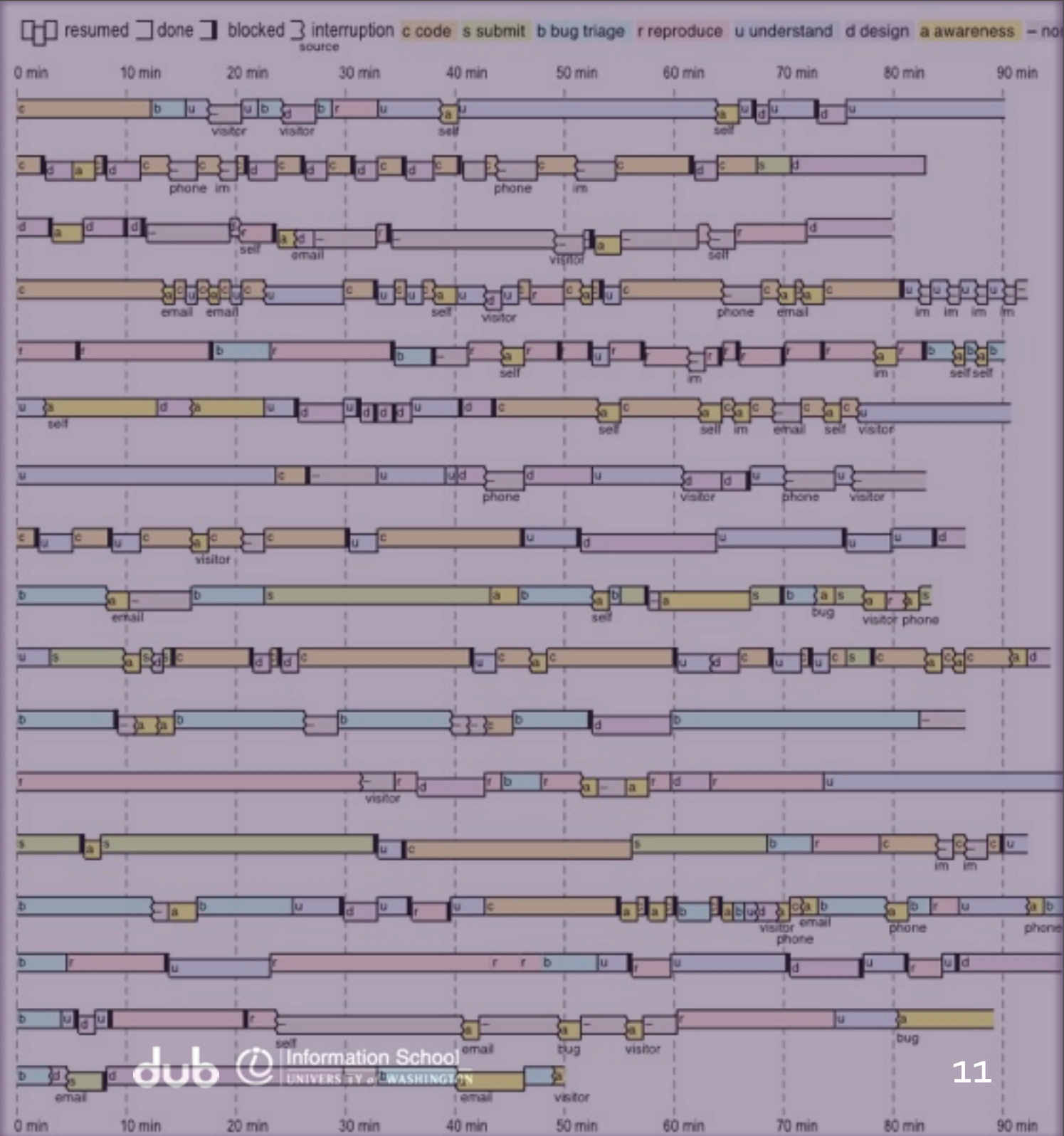
Pseudo-initial title	Product years on team	Work months in phase
A	Communication tool	Investigated two bugs
SDE 2	end users and dev	12 in bug fixing
B	17 developers	6 in development
SDE 1	developers	6 in bug fixing
C	Hard at work	5 in bug fixing
SDE 3	developers	5 in bug fixing
E	Service packs	Reproduced failures
SDE 2	end users	2 in bug fixing
F	Build automation	Diagnosed build failure
SDE 5	end users	6 in development
G	service packs	Prepared build config
SDE 1	end users	6 in development
H	mobile devices	Read unfamiliar code
SDE 1	end users	1 in bug fixing
I	discussion boards	Trigged 7 bug reports
SDE 1	end users	6 in bug fixing
K	SQL server	Wrote feature code
Lead 5	end users	5 in development
L	MS Office	Trigged w/ coworker
SDE 1	end users	2 in bug fixing
M	Encryption	Investigated two bugs
SDE 1	end users	2 in bug fixing
N	DRM	Prepared check in
SDE 4	end users	5 in development
R	Communication tool	Trigged and fixed bugs
SDE 3	end users	1 in bug fixing
T	Office application	Investigated two bugs
SDE 1	end users	5 in bug fixing
U	Office application	Waited for build & test
SDE 3	end users	12 in bug fixing
V	Content protection	Debated fix with team
Lead, < 1	end users and devs	5 in bug fixing

17 developers
hard at work
across 25 hours



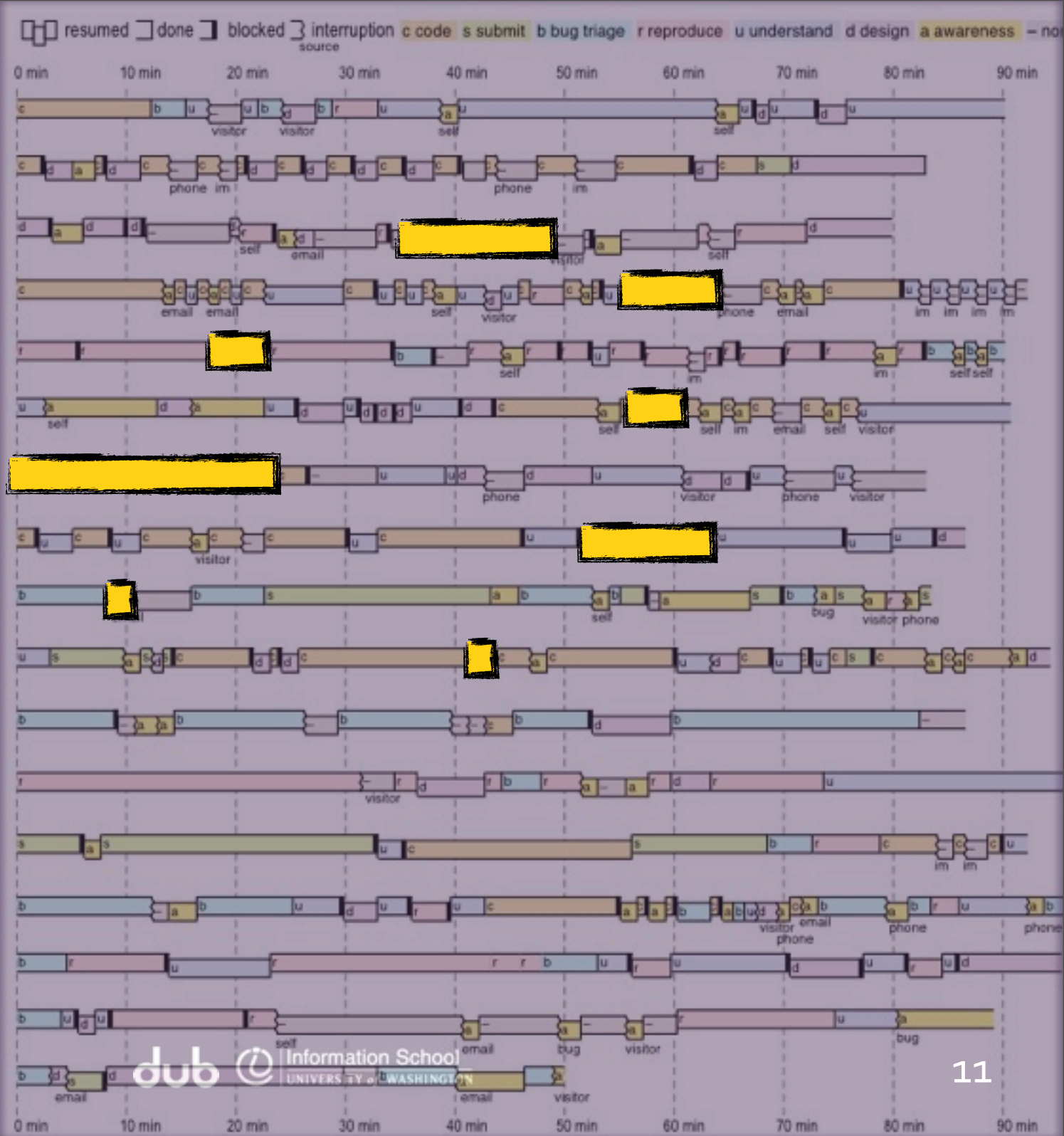
8 activities

- writing code
- submitting code
- triaging bugs
- reproducing a failure
- understanding behavior
- reasoning about design
- maintaining awareness
- non-work activity



8 activities

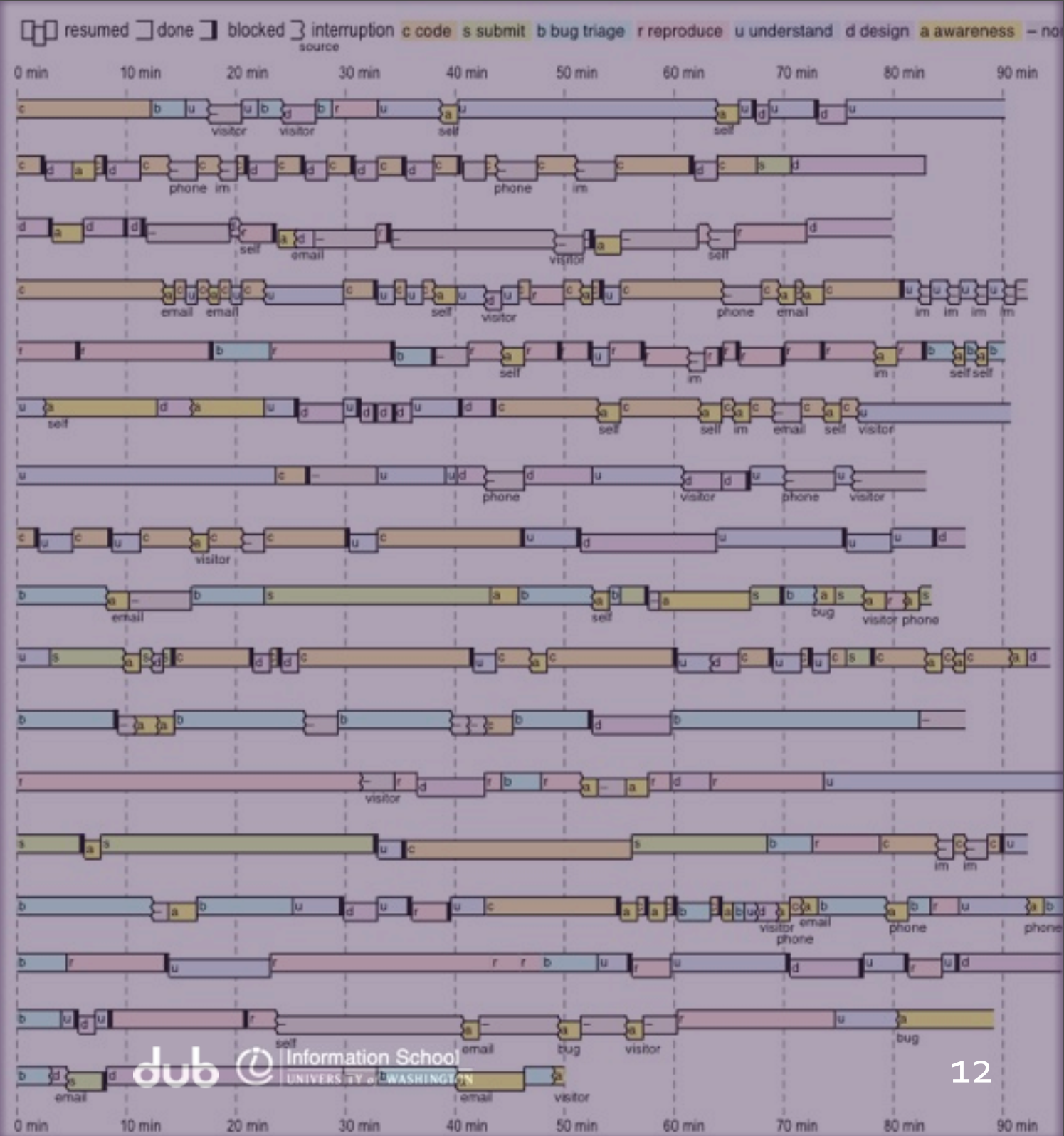
- writing code
- submitting code
- triaging bugs
- reproducing a failure
- understanding behavior
- reasoning about design
- maintaining awareness
- non-work activity



Pseudo-initial title	Product	Work
years on team	customers	months in phase
A	end users and devs	fixed two bugs
SDE, 2		12 in bug fixing
B	Development tool	Fixed generated code
SDE, 2		1 in development
C	File metadata	Interpreted spec
SDE, 3	developers	6 in bug fixing
D	end users	face to face conversation
SDE, 4		8 in development
E	developers	phone call
SDE, 2		2 in bug fixing
F	developers	instant message
SDE, <1		6 in development
G	end users	e-mail alerts
SDE, 1		6 in development
H	end users	bug report change alerts
SDE, 1		1 in bug fixing
I	end users and devs	task avoidance
SDE, 4		6 in bug fixing
K	end users	getting blocked
Lead, 3		5 in development
L	end users and devs	meetings
SDE, 10		2 in bug fixing
M	end users	task completion
SDE, 3		2 in bug fixing
N	end users	Data processing tool
SDE, 4		5 in development
R	end users	Communication tool
SDE, 3		1 in bug fixing
T	end users	Office application
SDE, 1		5 in bug fixing
U	end users	Office application
SDE, 3		12 in bug fixing
V	end users and devs	Content protection
Lead, <1		5 in bug fixing

9 reasons for switching tasks

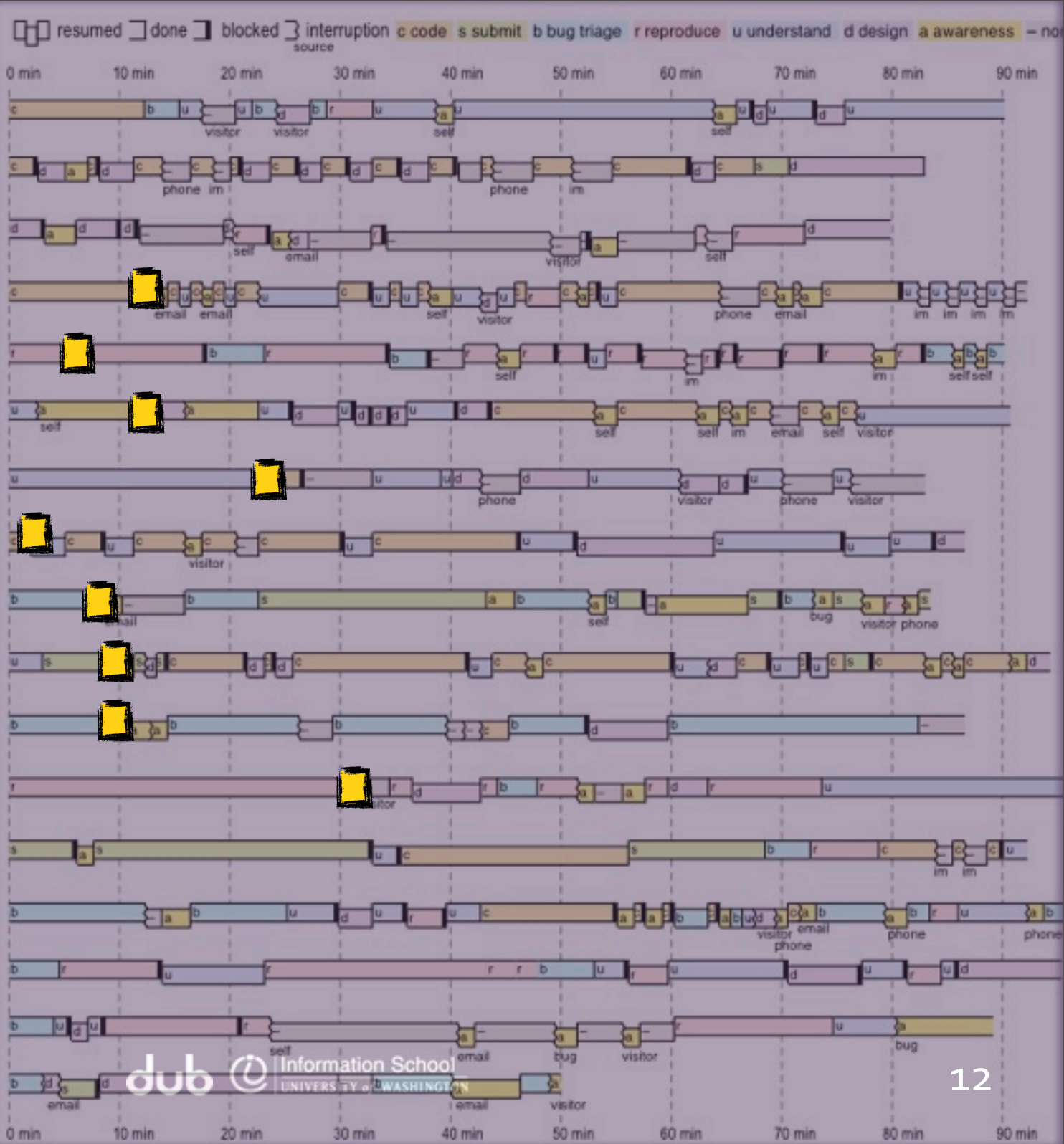
- face to face conversation
- phone call
- instant message
- e-mail alerts
- bug report change alerts
- task avoidance
- getting blocked
- meetings
- task completion



Pseudo-initial title	Product years on team	Work months in phase
A	end users and devs	12 in bug fixing
B	Development tool	Fixed generated code
C	File metadata	Interpreted spec
E	end users	6 in bug fixing
F	end users	6 in development
G	end users	6 in development
H	end users	1 in bug fixing
I	end users and devs	6 in bug fixing
K	end users	5 in development
L	end users and devs	2 in bug fixing
M	end users	2 in bug fixing
N	end users	5 in development
R	end users	1 in bug fixing
T	end users	5 in bug fixing
U	end users	12 in bug fixing
V	end users and devs	5 in bug fixing

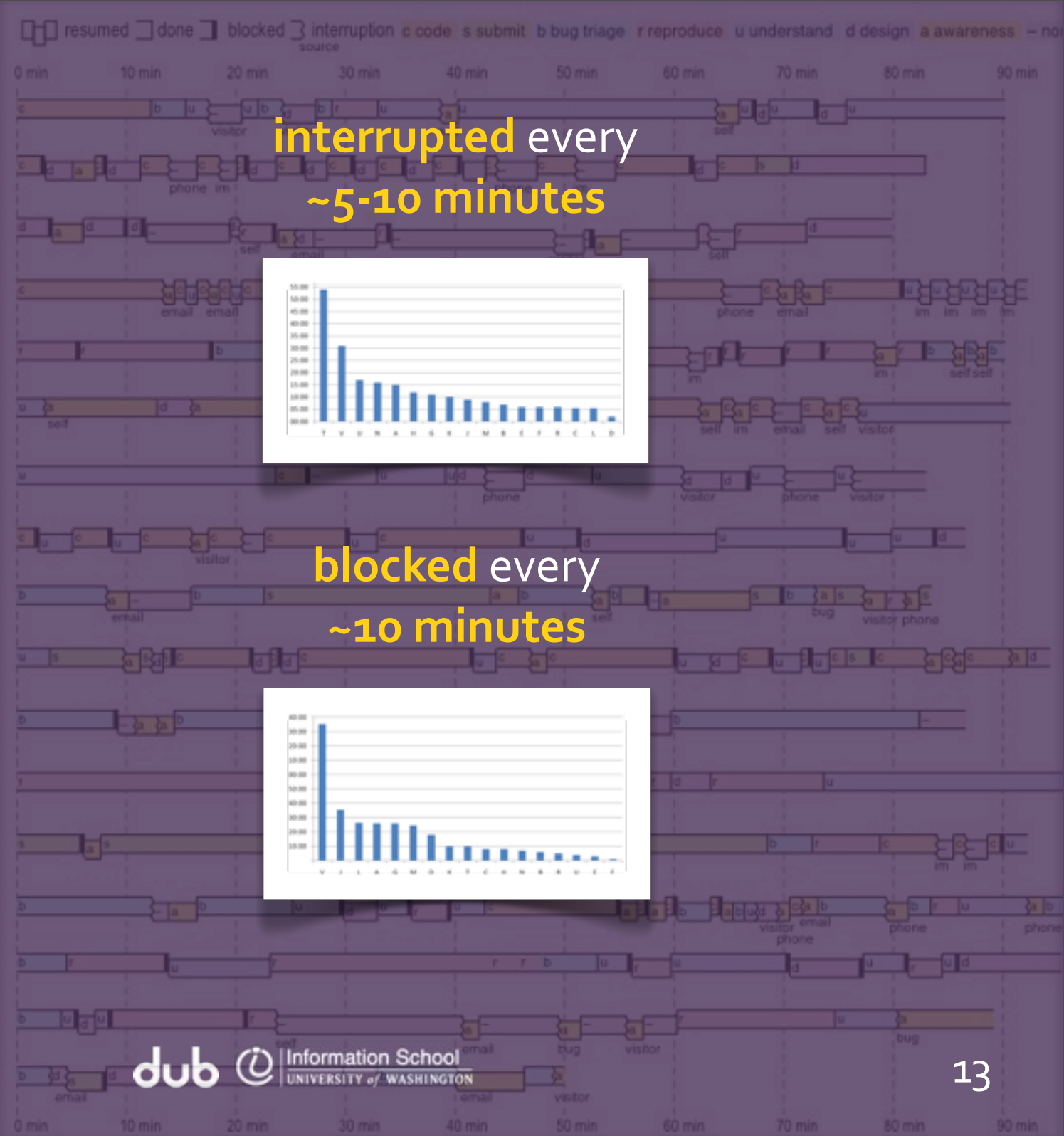
9 reasons for switching tasks

- face to face conversation
- phone call
- instant message
- e-mail alerts
- bug report change alerts
- task avoidance
- getting blocked
- meetings
- task completion



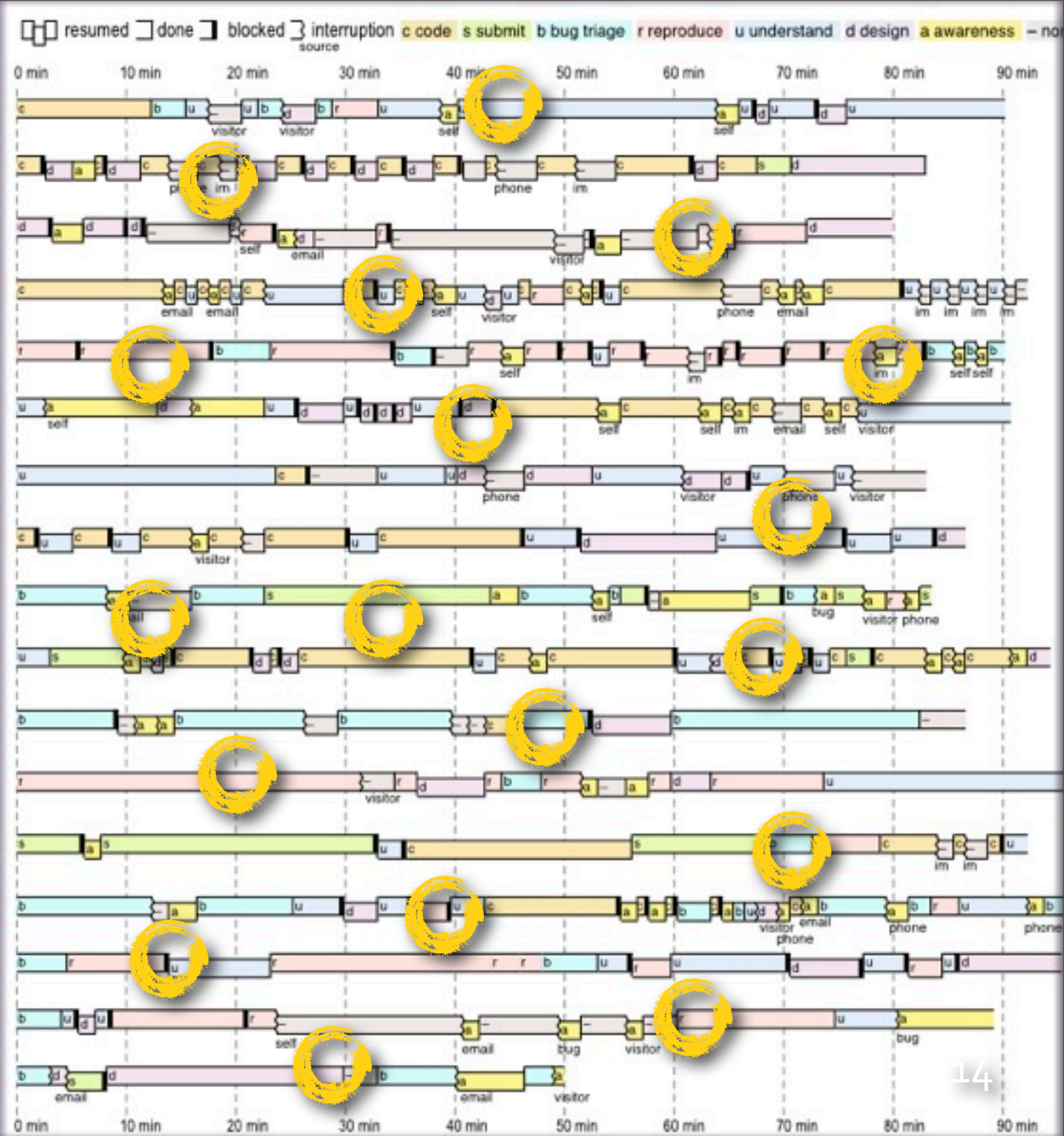
Pseudo-initial title, years on team	Product customers	Work months in phase
A SDE, 2	end users and devs	Investigated two bugs 12 in bug fixing
B SDE	Development tool	Fixed generated code development
C SDE	File metadata	Interpreted spec development bug fixing
D SDE	Mobile device tool	Wrote internal tool development
E SDE, 2	Service packs developers	Reproduced failures 2 in bug fixing
F SDE, <1	Build automation developers	Diagnosed build failure 6 in development
G SDE, 1	Mobile device tool end users	Prepared build config 6 in development
H SDE, 1	Discussion boards end users	Read unfamiliar code 1 in bug fixing
J SDE, 4	Communication tool end users and devs	Triaged 7 bug reports 6 in bug fixing
K Lead, 5	Education tool end users	Wrote feature code 5 in development
L SDE, 10	Development tool end users and devs	Triaged w/ coworker 2 in bug fixing
M SDE, 3	input device UI end users	Investigated two bugs 2 in bug fixing
N SDE, 4	Data processing tool end users	Prepared check in 5 in development
R SDE, 3	Communication tool end users	Triaged and fixed bugs 1 in bug fixing
T SDE, 1	Office application end users	Investigated two bugs 5 in bug fixing
U SDE, 3	Office application end users	Waited for build & test 12 in bug fixing
V Lead, < 1	Content protection end users and devs	Debated fix with team 5 in bug fixing

software development work is highly fragmented



Pseudo-initial title	Product years on team	Work months in phase
A	Communication tool	Investigated two bugs
SDE, 1	end users	2 in bug fixing
B	Development tool	Fixed generated code
SDE, 1	end users	1 in bug fixing
C	File metadata	Interpreted spec
SDE, 1	end users	6 in bug fixing
D	Mobile device tool	Wrote internal tool
SDE, 4	end users	6 in development
E	Mobile device tool	Diagnosed failures
SDE, 2	end users	2 in bug fixing
F	Mobile device tool	Diagnosed build failure
SDE, 4	end users	6 in development
G	Mobile device tool	Prepared build config
SDE, 7	end users	6 in development
H	Discussion boards	Read unfamiliar code
SDE, 1	end users	1 in bug fixing
I	Communication tool	Triaged 7 bug reports
SDE, 4	end users and devs	6 in bug fixing
K	Education tool	Wrote feature code
Lead, 5	end users	5 in development
L	Development tool	Triaged w/ coworker
SDE, 10	end users and devs	2 in bug fixing
M	Input device UI	Investigated two bugs
SDE, 3	end users	2 in bug fixing
N	Data processing tool	Prepared check in
SDE, 4	end users	5 in development
R	Communication tool	Triaged and fixed bugs
SDE, 3	end users	1 in bug fixing
T	Office application	Investigated two bugs
SDE, 1	end users	5 in bug fixing
U	Office application	Waited for build & test
SDE, 3	end users	12 in bug fixing
V	Content protection	Debated fix with team
Lead, < 1	end users and devs	5 in bug fixing

next, we looked for **information** that developers sought to get their work done ...



21 information needs observed (by frequency)

what have my coworkers been doing?

what code could have caused this behavior?

have resources I depend on changed?

what code caused this program state?

how do I use this data structure or function?

did I make any mistakes?

what is the program supposed to do?

in what situations does this failure occur?

is this problem worth fixing?

why was this code implemented this way?

what's statically related to this code?

what are the implications of this change?

5 information needs least often satisfied

	% unsatisfied
what have my coworkers been doing?	
what code could have caused this behavior?	36%
have resources I depend on changed?	
what code caused this program state?	61%
how do I use this data structure or function?	
did I make any mistakes?	
what is the program supposed to do?	15%
in what situations does this failure occur?	41%
is this problem worth fixing?	
why was this code implemented this way?	44%
what's statically related to this code?	
what are the implications of this change?	

3 were **debugging** related ...

what have my coworkers been doing?

what code could have caused this behavior?

have resources I depend on changed?

what code caused this program state?

how do I use this data structure or function?

did I make any mistakes?

what is the program supposed to do?

in what situations does this failure occur?

is this problem worth fixing?

why was this code implemented this way?

what's statically related to this code?

what are the implications of this change?

reproducing,
diagnosing,
and **scoping**
failures were
the most time-
consuming
activities

2 were **design** related

what have my coworkers been doing?

what code could have caused this behavior?

have resources I depend on changed?

what code caused this program state?

how do I use this data structure or function?

did I make any mistakes?

what is the program supposed to do?

in what situations does this failure occur?

is this problem worth fixing?

why was this code implemented this way?

what's statically related to this code?

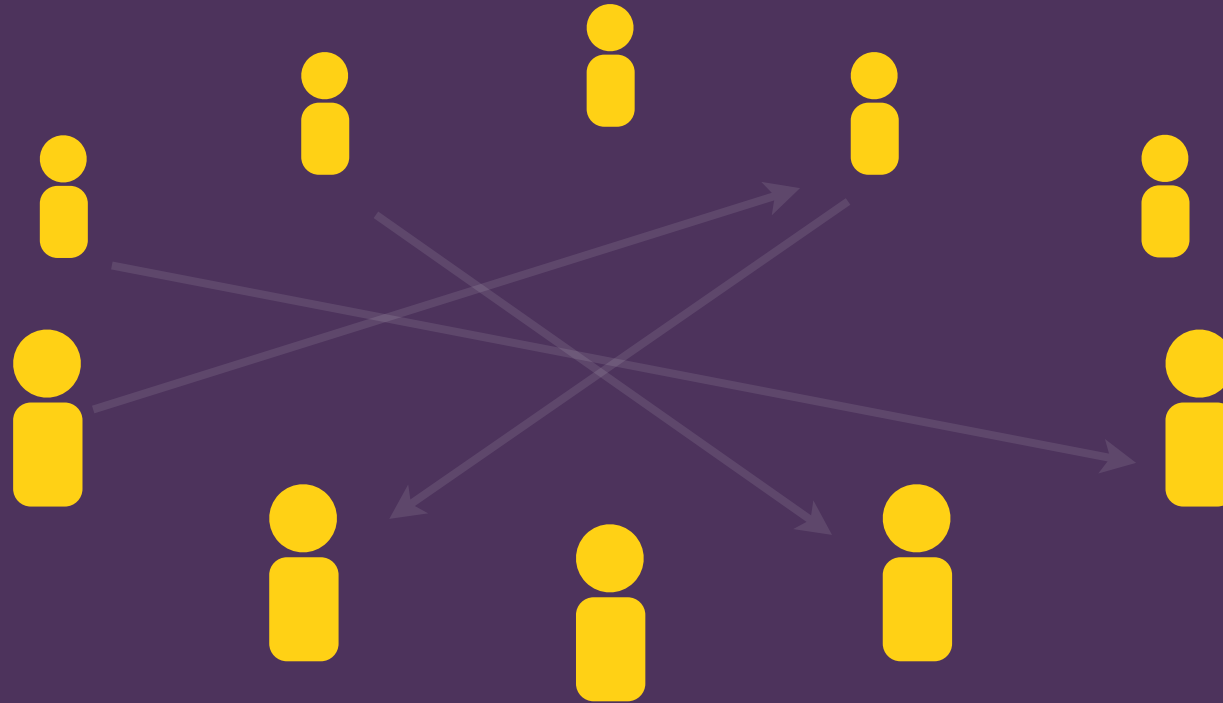
what are the implications of this change?

knowing what
software
should do was
rarely known

software development is tacit

plans and specifications are **unwritten**





software quality depends highly on the quality of **human communication** and **cognition**

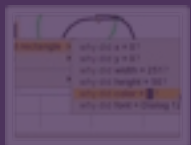
these human activities are **faulty and unreliable**

talk outline



why is software evolution **difficult**?

a study of information needs at Microsoft



how can **tools** help software evolution?

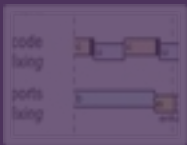
debugging with the Whyline



how can **users** help software evolution?

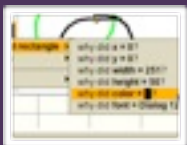
recent work in leveraging the crowd

talk outline



why is software evolution **difficult**?

a study of information needs at Microsoft



how can **tools** help software evolution?

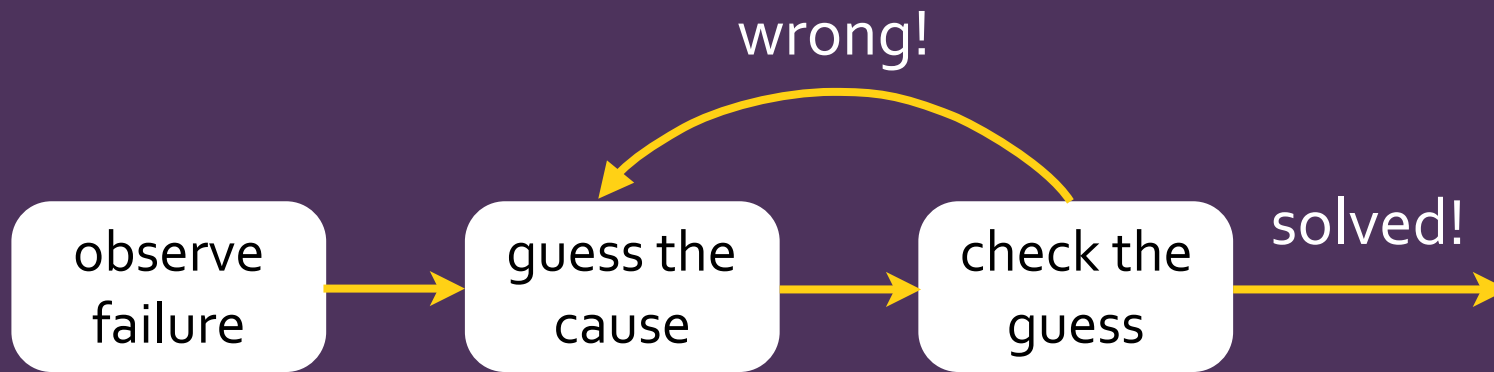
debugging with the Whyline



how can **users** help software evolution?

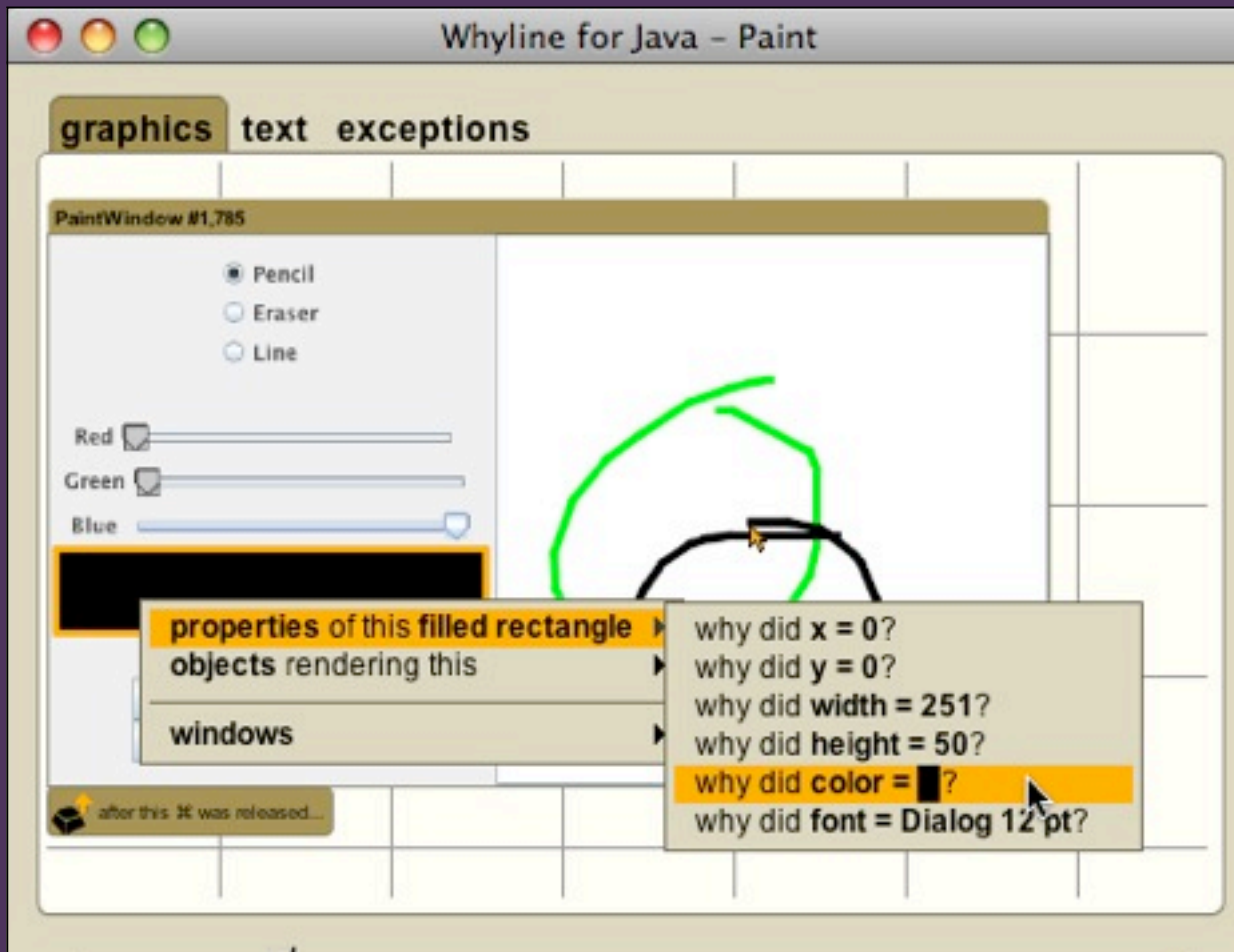
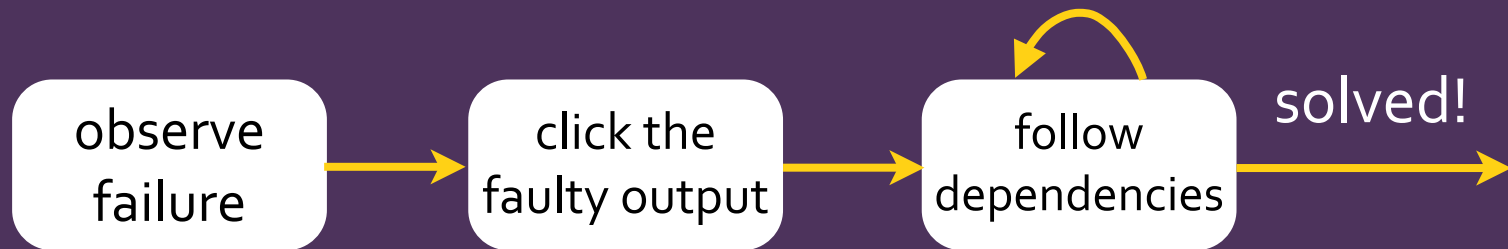
recent work in leveraging the crowd

why is **debugging** such a challenge?



Whyline

a Workspace for Helping You Link Instructions, Numbers, and Events

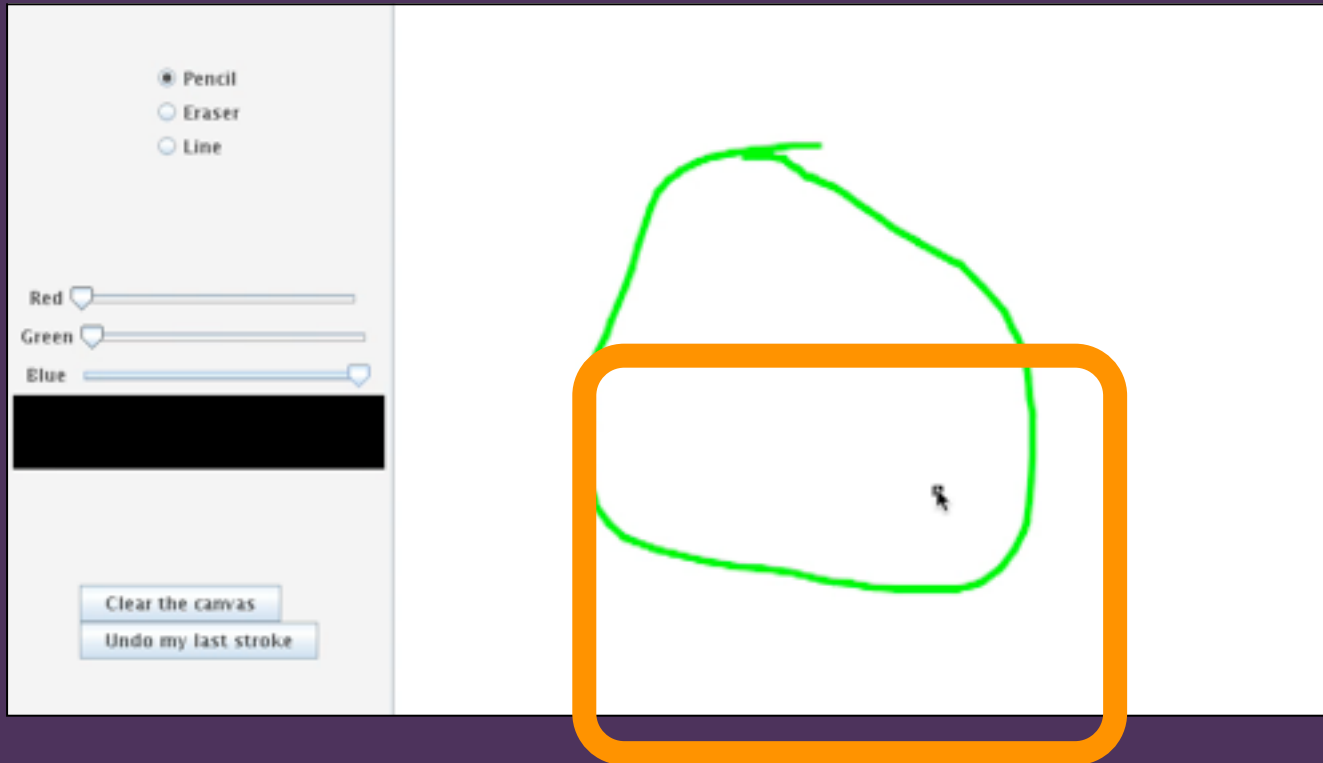


an example ...

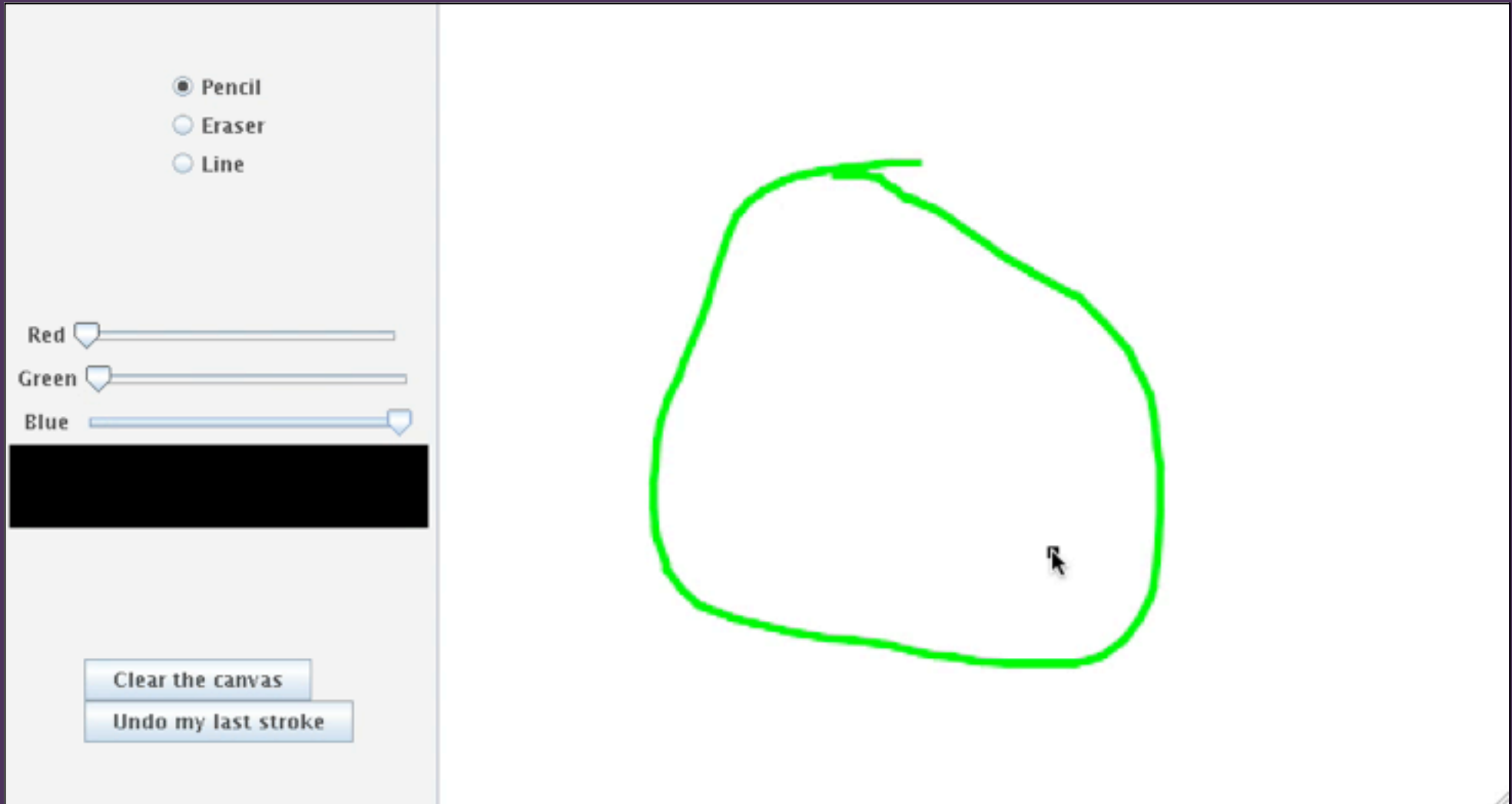
why was the line black?

an example ...

why was the line black?



record the problem



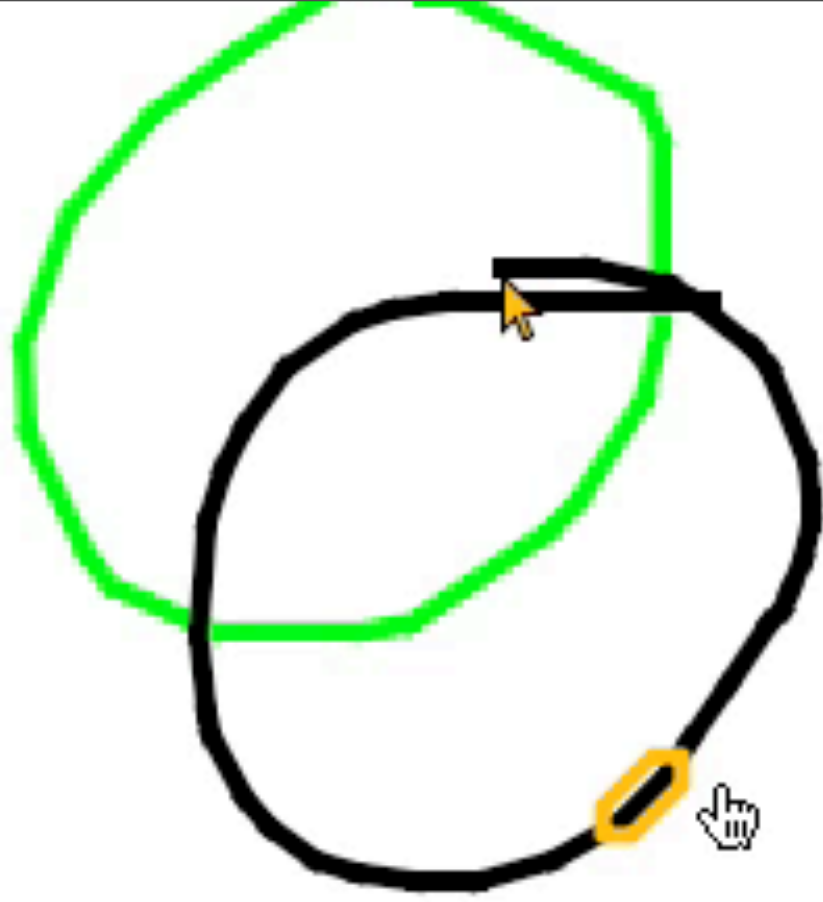
record the problem



Resolving classes (856 remaini

load the recording

why was the line color black?



why was the line color black?

why was the line color black?

graphics text exceptions

PaintWindow #1,785

- Pencil
- Eraser
- Line

Red

Green

Blue

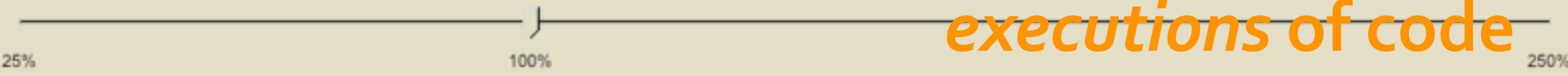
Clear the canvas

Undo my last stroke

code

- properties of this line
- objects rendering this
- windows
- why did x1 = 188?
- why did y1 = 288?
- why did x2 = 176?
- why did y2 = 300?
- why did color = ?**
- why did font = Dialog 12 pt?
- why did stroke = 5.0 pixel stroke?

executions of code
(execution events)



showing all i/o events

before this was released...

why was the line color black?

```

37
38
39 public Rectangle getBoundingBox() {
40     return new Rectangle(minX, minY, maxX - minX, maxY - minY);
41 }
42
43
44 public void paint(Graphics2D g) {
45
46     Stroke oldStroke = g.getStroke();
47     g.setStroke(new BasicStroke(thickness));
48     g.setColor(color);
49
50     for(int pointIndex = points.length - 1; pointIndex >= 1; pointIndex--) {
51
52         Point one = points[pointIndex];
53         Point two = points[pointIndex - 1];
54         g.drawLine((int)one.getX(), (int)one.getY(), (int)two.getX(), (int)two.getY());
55
56     }
57
58     g.setStroke(oldStroke);
59

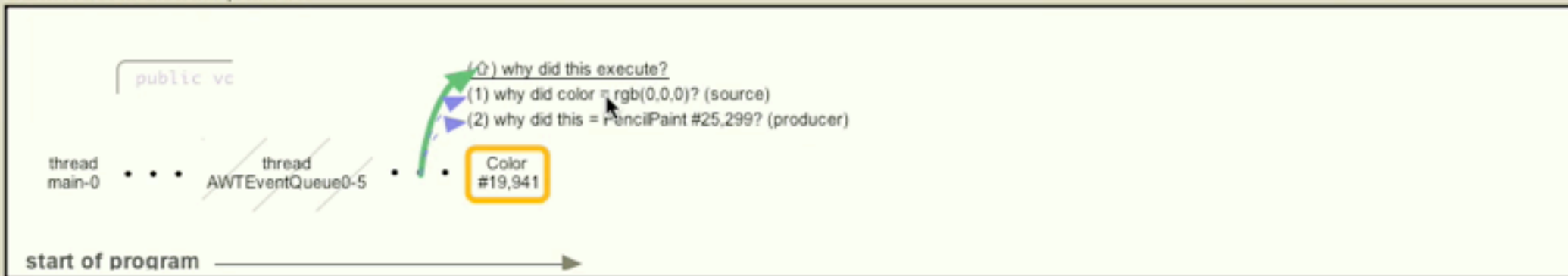
```

PencilPaint #25,299's field color was Color #19,941
 (⌵) why did this execute?
 (1) why did color = rgb(0,0,0)? (source)
 (2) why did this = PencilPaint #25,299? (source)

Q why did color = ■?

A These events were responsible.

← event → event ← in method → in method ← in thread → in thread ⌵ block collapse/expand show threads



why was the line color black?

```
one = points[pointIndex];
two = points[pointIndex - 1];
drawLine((int)one.getX(), (int)one.getY(), (int)two.getX(), (int)two.getY())
stroke.setColor(oldStroke);
```

followup questions about selected event

(↑) why did this execute?

(1) why did color = rgb(0,0,0)? (source)

(2) why did this = PencilPaint #25,299? (producer)

Color

why was the line color black?

PencilPaint #25,299's field color was Color #19,941

(↑) why did this execute?

(1) why did color = rgb(0,0,0)? (source)

(2) why did this = PencilPaint #25,299? (source)

followup
questions about
selected event

why was the line color black?


```

37
38
39 public Rectangle getBoundingBox() {
40     return new Rectangle(minX, minY, maxX - minX, maxY - minY);
41 }
42
43
44 public void paint(Graphics2D g) {
45
46     Stroke oldStroke = g.getStroke();
47     g.setStroke(new BasicStroke(thickness));
48     g.setColor(color);
49
50     for(int pointIndex = points.length - 1; pointIndex >= 0; pointIndex--) {
51
52         Point one = points[pointIndex];
53         Point two = points[pointIndex - 1];
54         g.drawLine((int)one.getX(), (int)one.getY(), (int)two.getX(), (int)two.getY());
55
56     }
57
58     g.setStroke(oldStroke);
59

```

PencilPaint #25,299's field color was Color #19,941
 (🔍) why did this execute?
 (1) why did color = rgb(0,0,0)? (source)
 (2) why did this = PencilPaint #25,299? (source)

selected
 dependency
 highlighted
 in source

Q why did color = ■?

A These events were responsible.

← event → event ← in method → in method ← in thread → in thread 🔍 block collapse/expand show threads

```
public vc
```

thread main-0 • • • thread AWTEventQueue0-5

(🔍) why did this execute?
 (1) why did color = rgb(0,0,0)? (source)
 (2) why did this = PencilPaint #25,299? (producer)

Color #19,941

start of program →

why was the line color black?

```

public Rectangle getBoundingBox() {
41
42 }
43
44 public void paint(Graphics2D g) {
45
46     Stroke oldStroke = g.getStroke();
47     g.setStroke(new BasicStroke(thickness));
48     g.setColor(color);
49
50     for(int pointIndex = points.length - 1; pointIndex >= 1; pointIndex--) {
51
52         Point one = points[pointIndex];
53         Point two = points[pointIndex - 1];
54         g.drawLine((int)one.getX(), (int)one.getY(), (int)two.getX(), (int)two.getY());
55
56     }
57 }

```

PencilPaint #25,299's field color was Color #19,941
 (⚙) why did this execute?
 (1) why did color = rgb(0,0,0)? (source)
 (2) why did this = PencilPaint #25,299? (source)

```

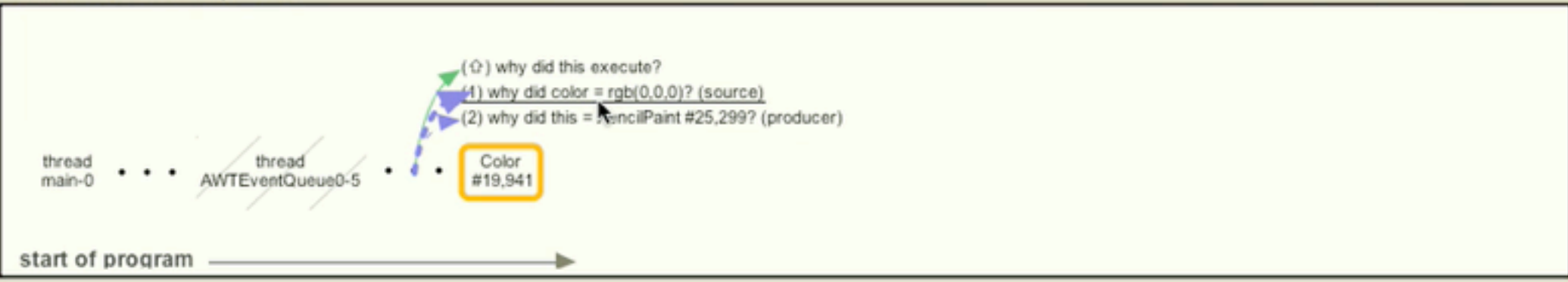
public void paintComponent(Graphics g) {
27
28     public void stateChanged(ChangeEvent changeEvent) {
29
30         objectConstructor.setColor(
31             new Color(
32                 slider.getValue(),
33                 slider.getValue(),
34                 slider.getValue());
35     }
36 }

```

Q why did color = ■?

A These events were responsible.

← → event event ← in → in method method ← in → in thread thread ⚙ block collapse/expand show threads



why was the line color black?

```
Stroke oldStroke = g.getStroke();  
g.setStroke(new BasicStroke(thickness));  
g.setColor(color);
```

```
for(int pointIndex = points.length - 1; pointIndex >= 1; pointIndex
```

```
Point one = points[pointIndex];
```

```
Point two = points[pointIndex - 1];
```

```
g.drawLine((int)one.getX(), (int)one.getY(), (int)two.getX(), (int)two.getY());
```

because gSlider
was used twice,
ignoring bSlider

```
public void paintComponent(Graphics g) {
```

```
public void stateChanged(ChangeEvent changeEvent) {
```

```
objectConstructor.setColor(  
new Color(  
bSlider.getValue(),  
gSlider.getValue(),  
gSlider.getValue()));
```

```
new Color(  
bSlider.getValue(),  
gSlider.getValue(),  
gSlider.getValue()));
```

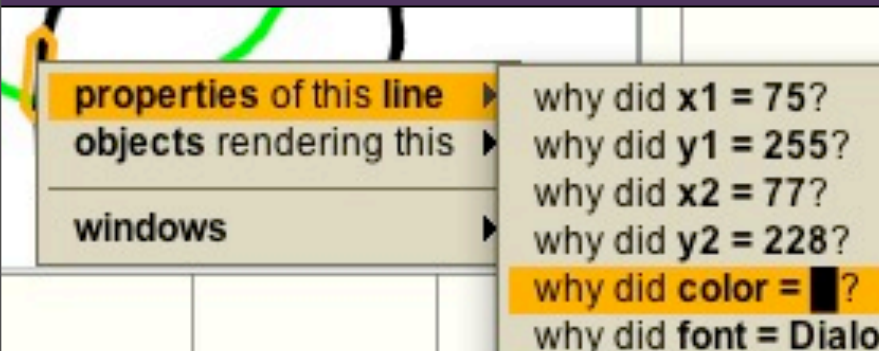
```
bSlider.getValue(),  
gSlider.getValue(),  
gSlider.getValue()));
```

```
gSlider.getValue(),  
gSlider.getValue()));
```

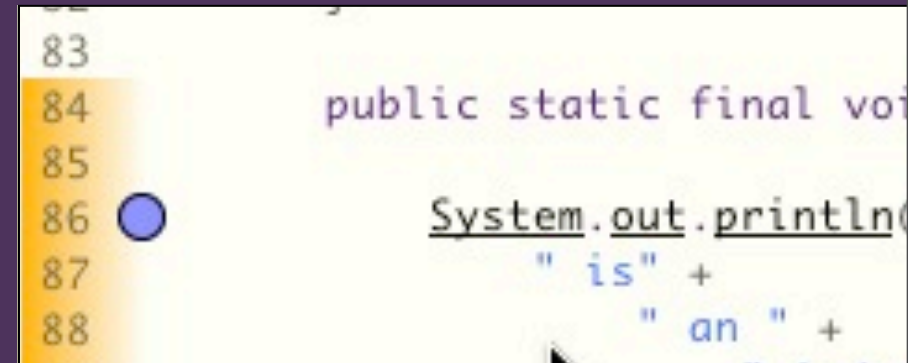
```
gSlider.getValue()));
```

why was the line color black?

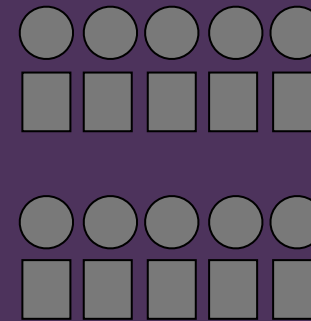
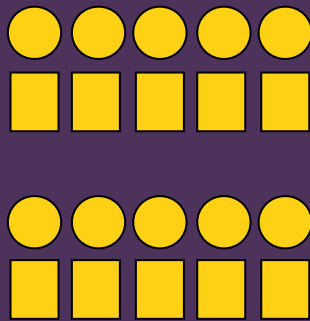
a comparison study



VS



Whyline
group



control
group

both groups had modern IDE features
show declaration, show callers, show references, etc.

subject program

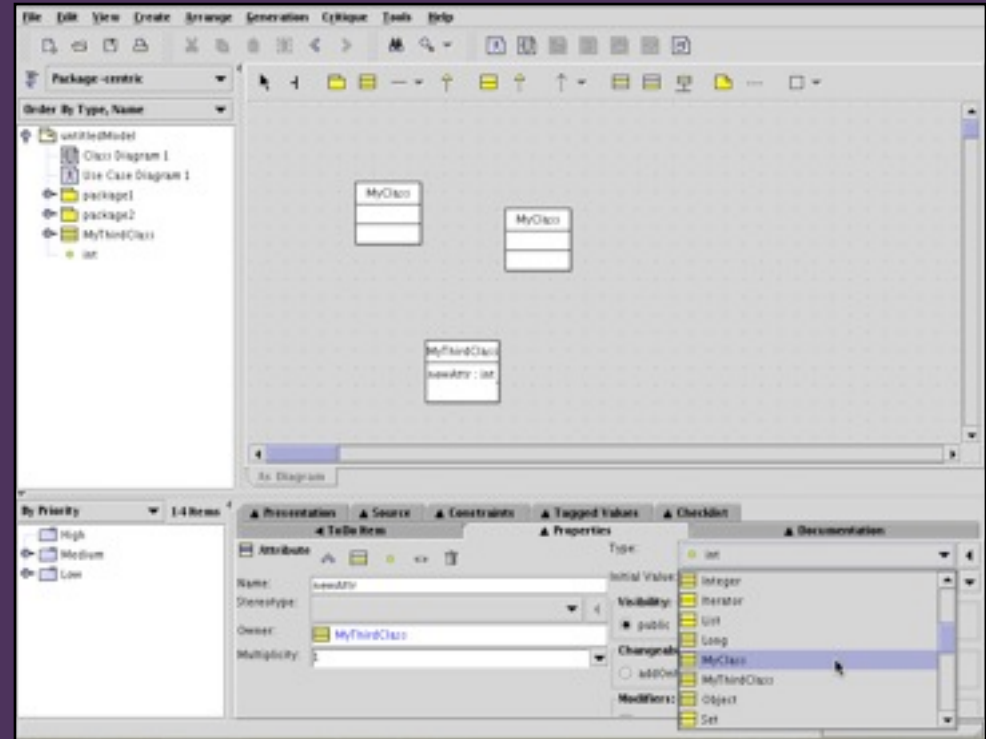
ArgoUML, an open source software design tool

~**150,000** lines of code

22 external libraries

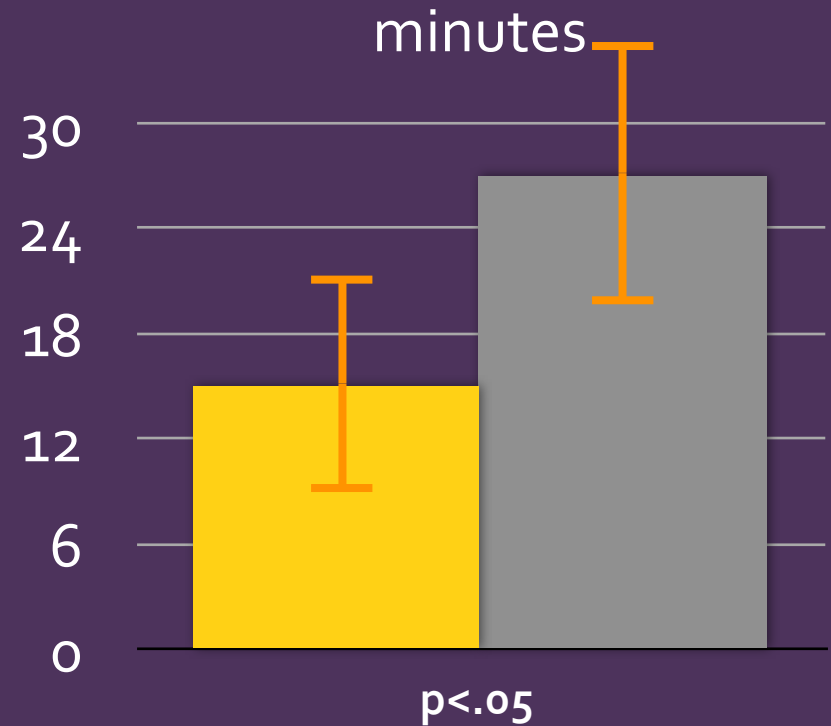
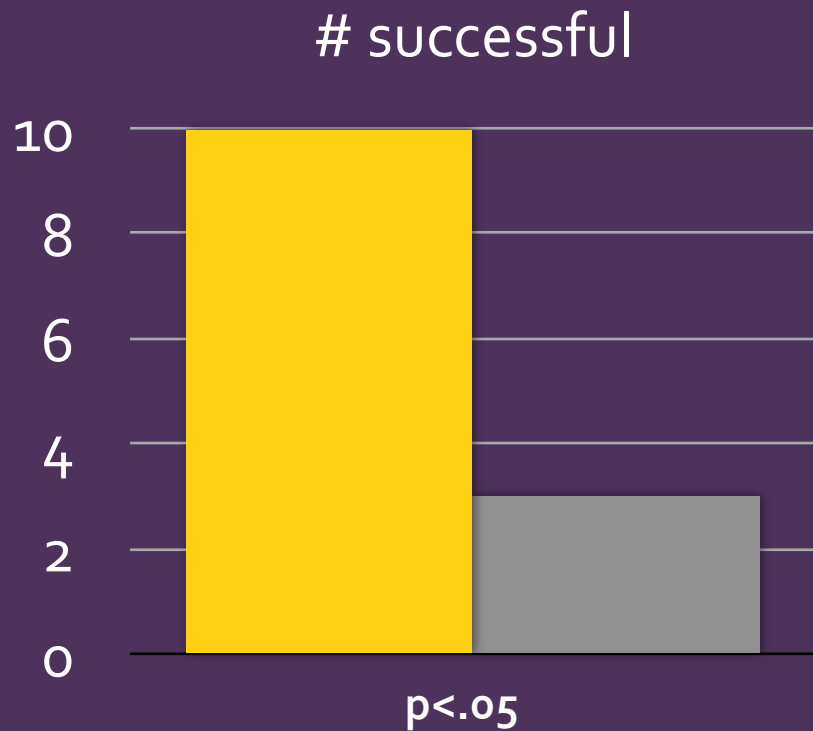
chose **two bug reports** from version 18.1

- one w/ simple fix
- one w/ complex fix



bug 1 results

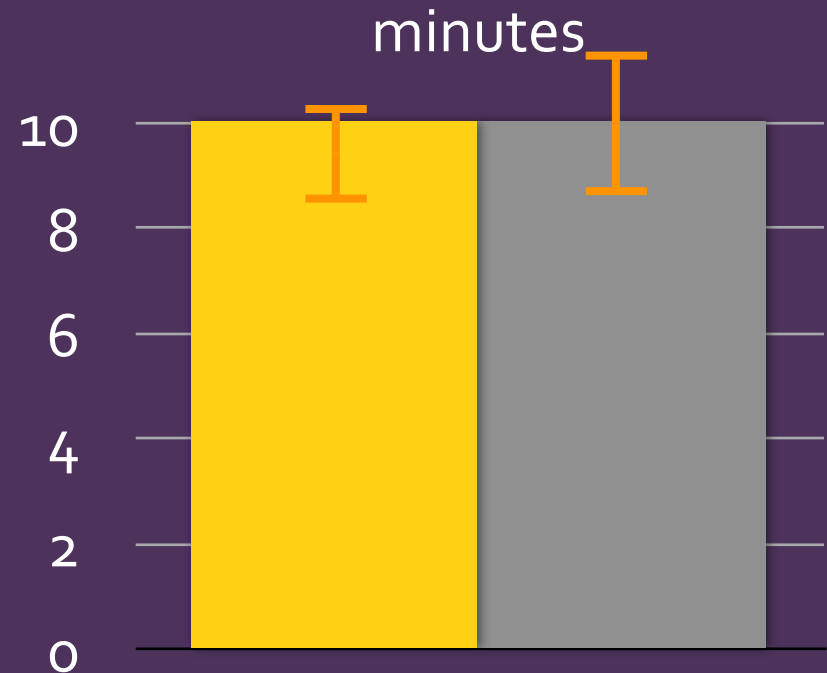
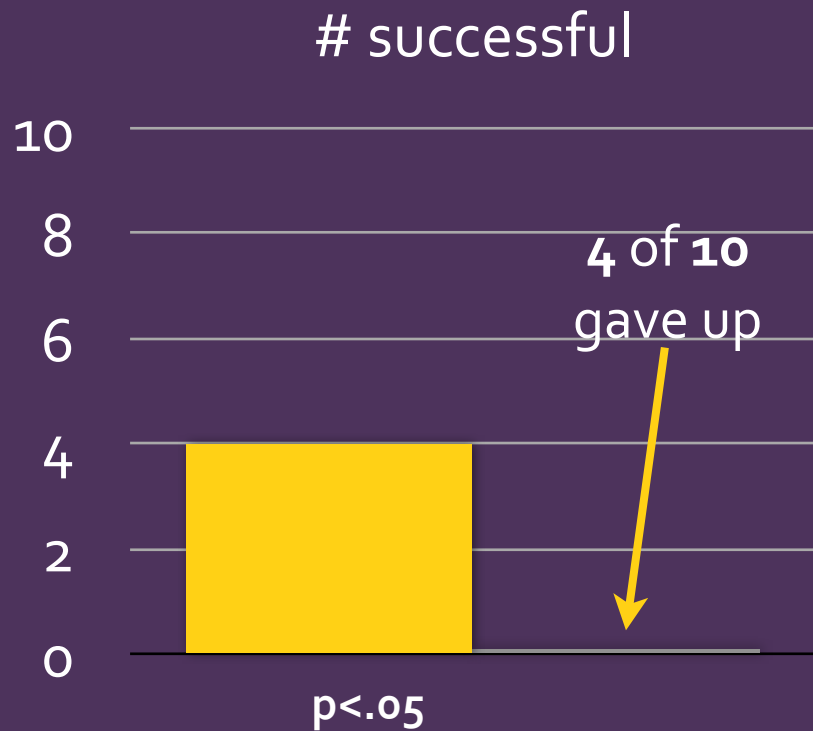
■ whyline ■ control



more successful in **half** the time

bug 2 results

■ whyline ■ control



more successful in the same time

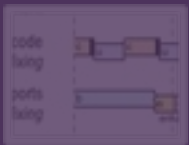
unsolicited quotes from users

“This is great, when can I get this for C?”

“My god, this is so cool.”

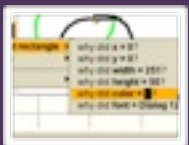
“It's so nice and straight and simple...”

talk outline



why is software evolution **difficult**?

a study of information needs at Microsoft



how can **tools** help software evolution?

debugging with the Whyline



how can **users** help software evolution?

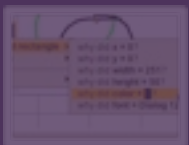
recent work in leveraging the crowd

talk outline



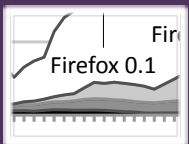
why is software evolution **difficult**?

a study of information needs at Microsoft



how can **tools** help software evolution?

debugging with the Whyline



how can **users** help software evolution?

recent work in leveraging the crowd

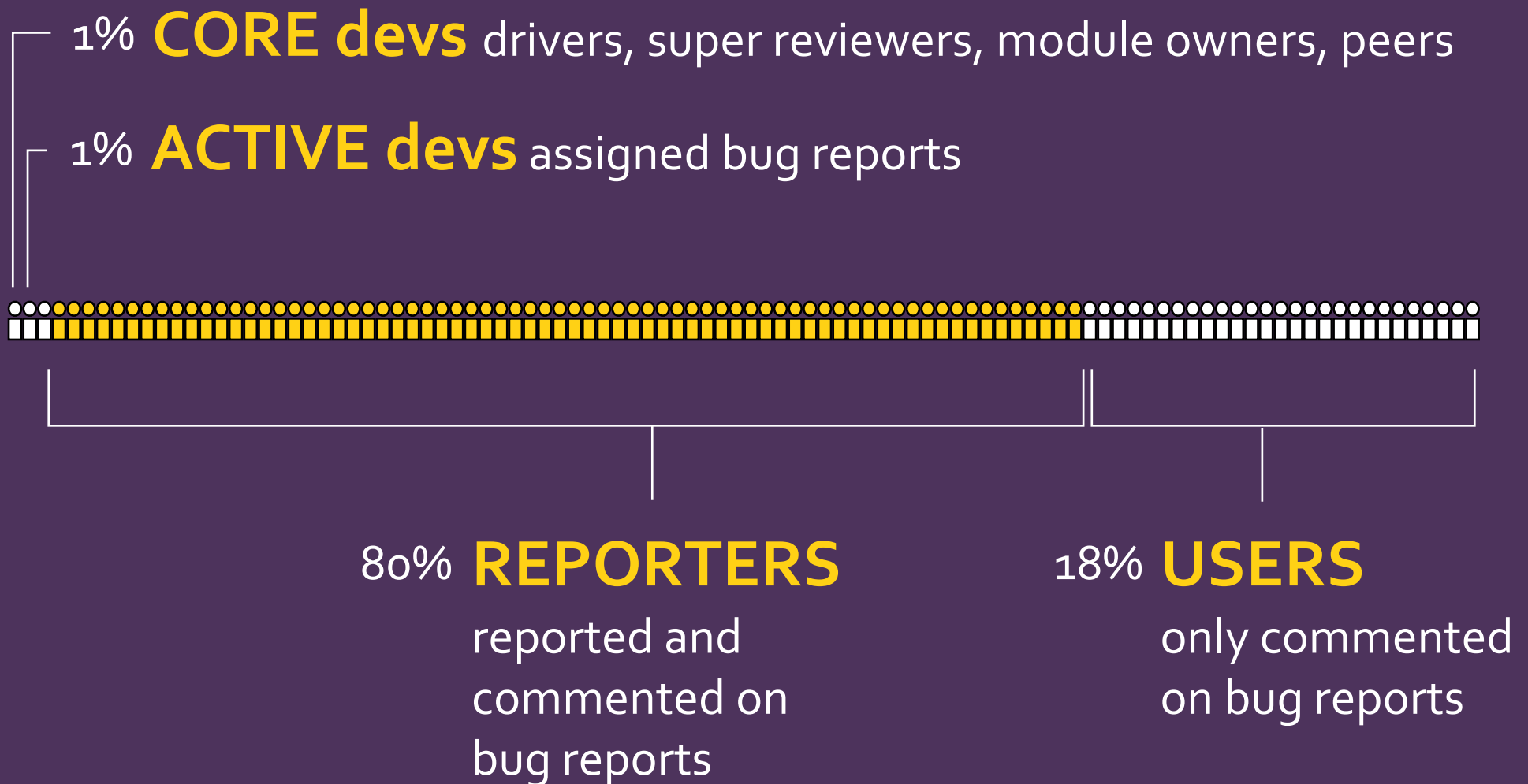
does open bug reporting work?

with my PhD student, Parmit Chilana

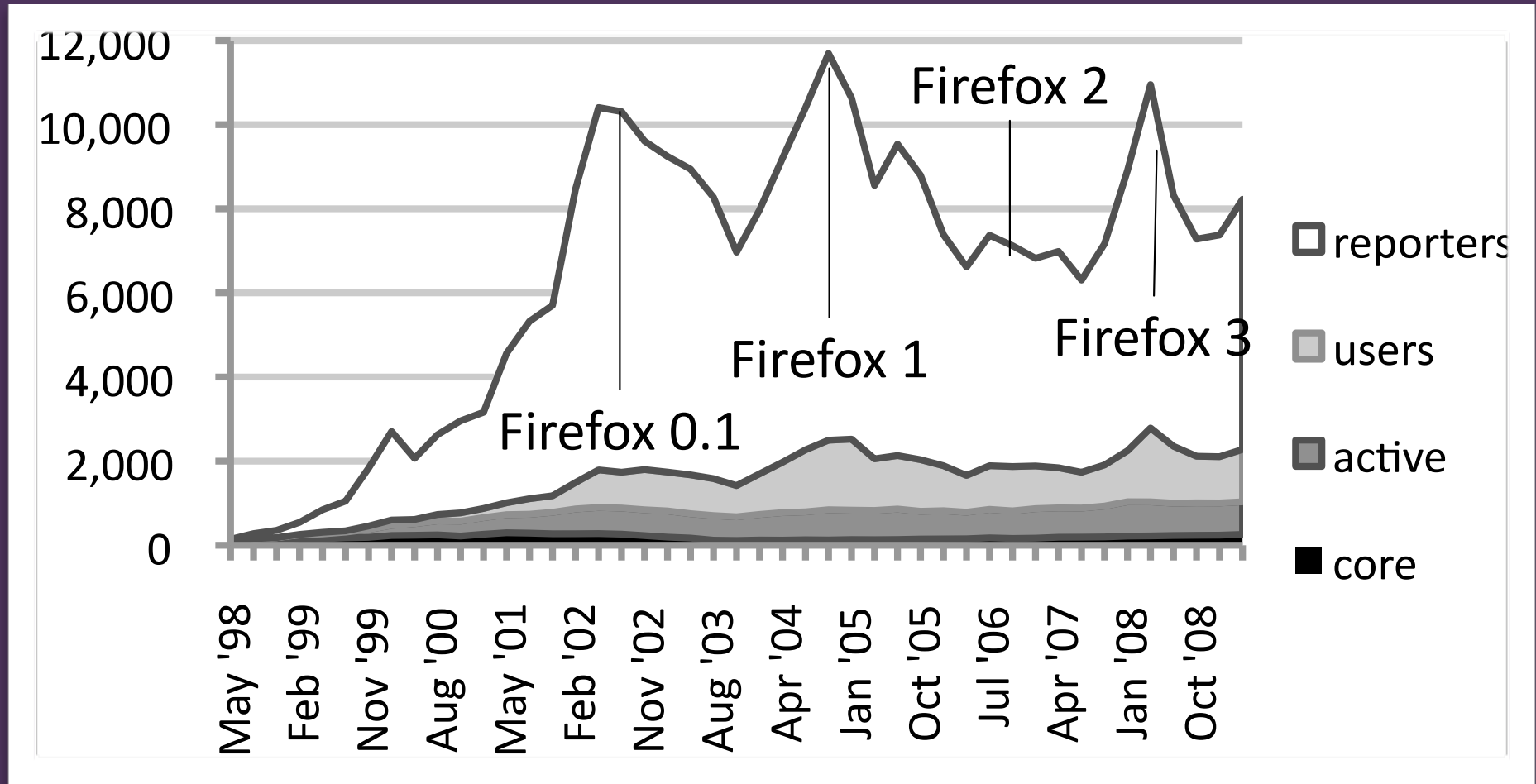
comprehensive analysis of **~500,000 reports** from the Mozilla community

- **quantitative** analysis of to **characterize** bug report resolution trends
- **qualitative** analysis to **explain** bug report resolution trends

four types of contributors

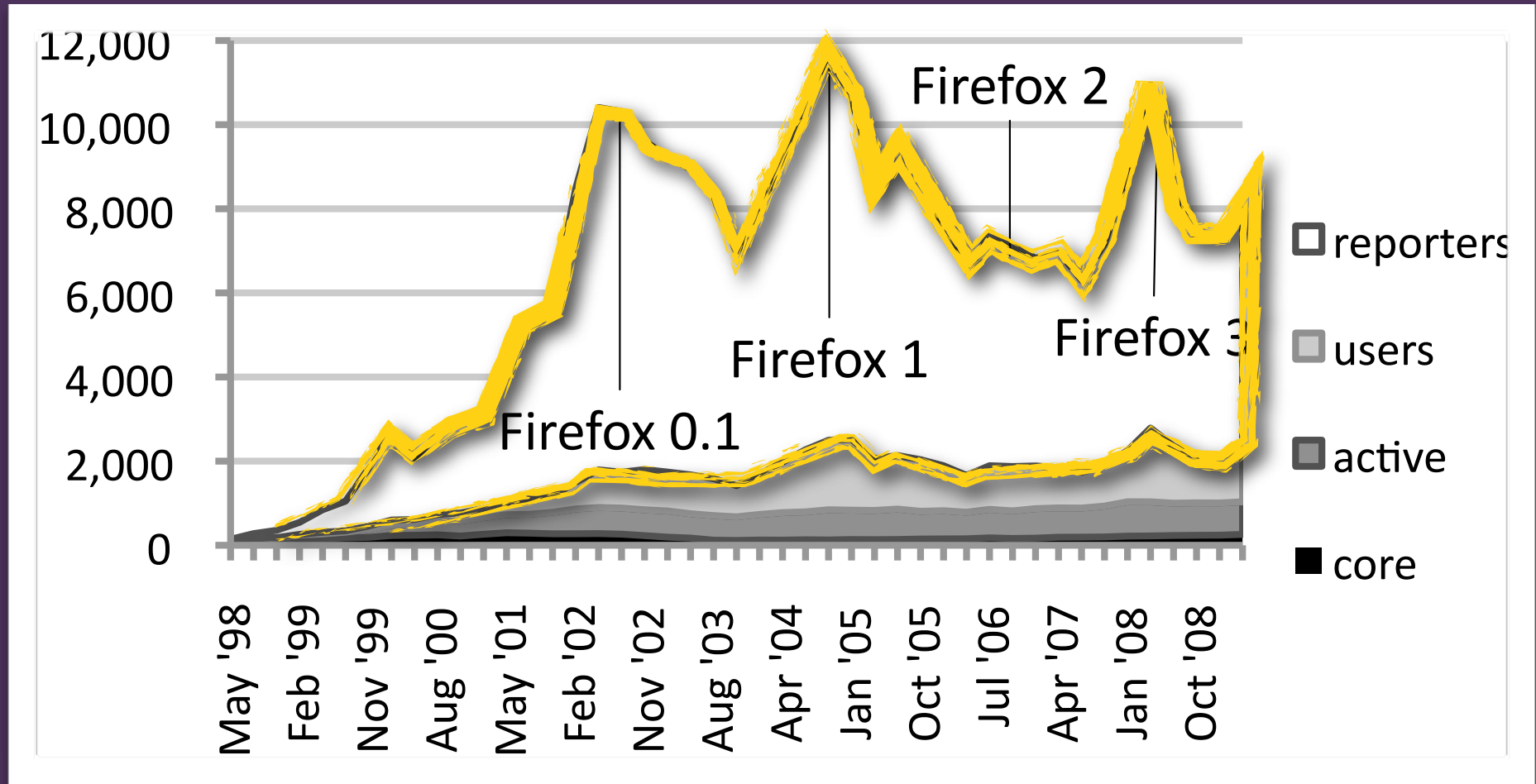


of comment contributors over time



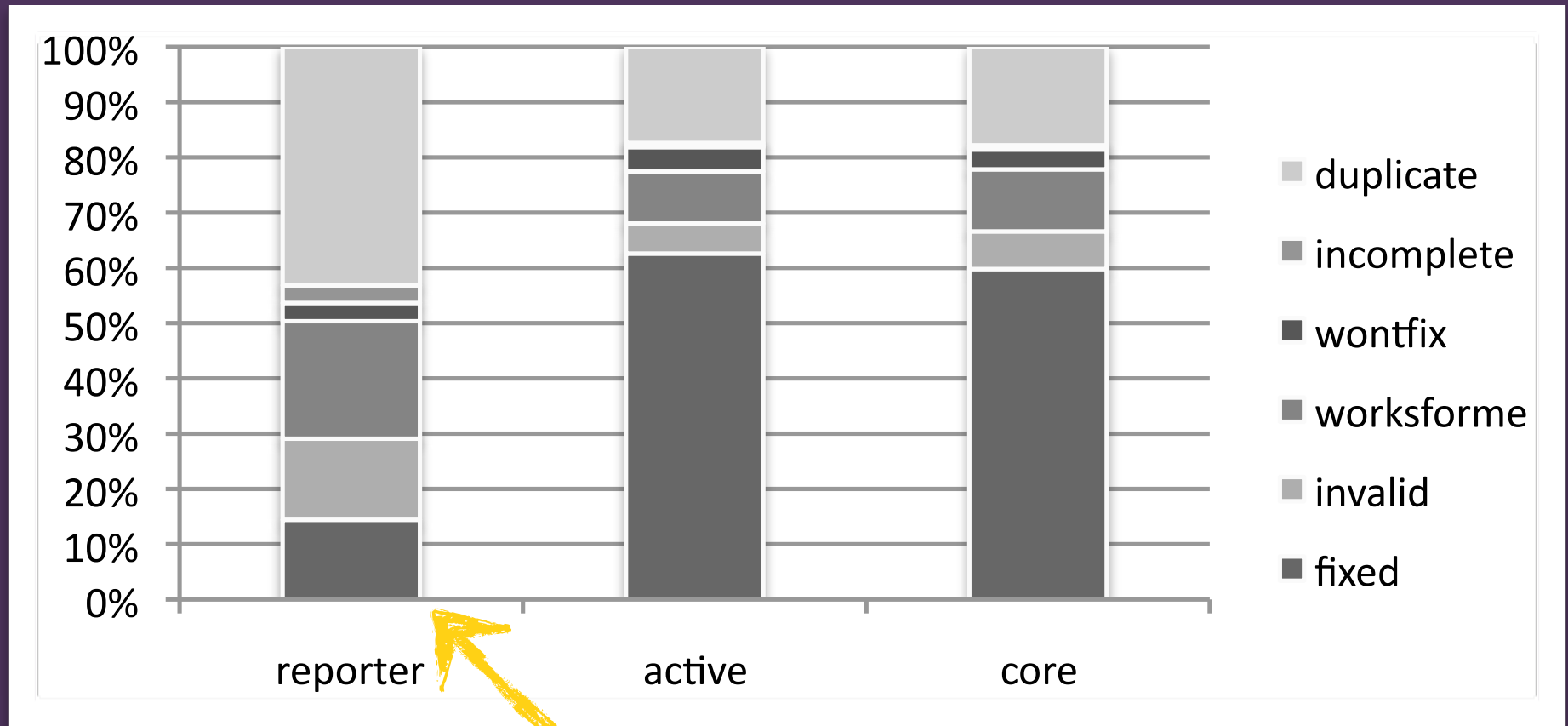
REPORTERS are the most active commenters

of comment contributors over time



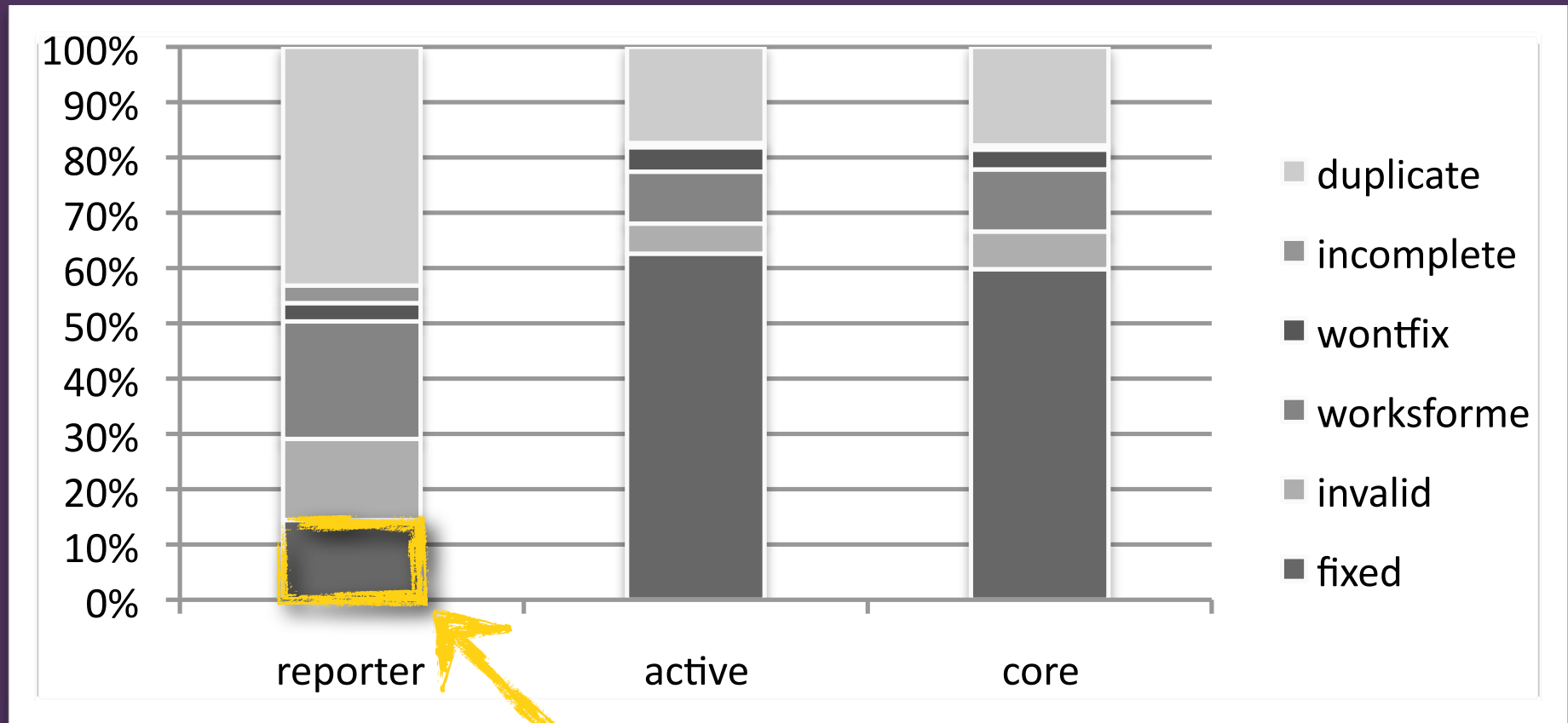
REPORTERS are the most active commenters

resolution by reporter type



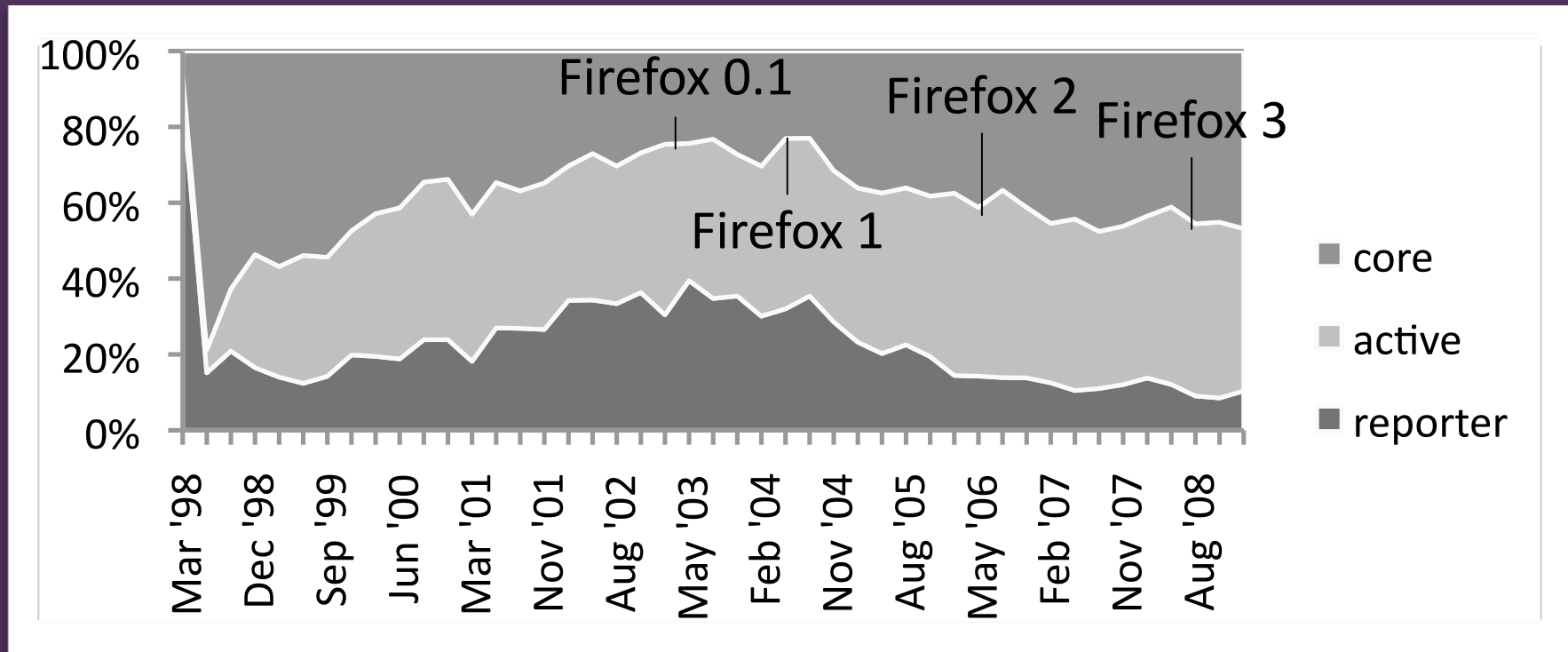
most REPORTER reports are not FIXED

resolution by reporter type



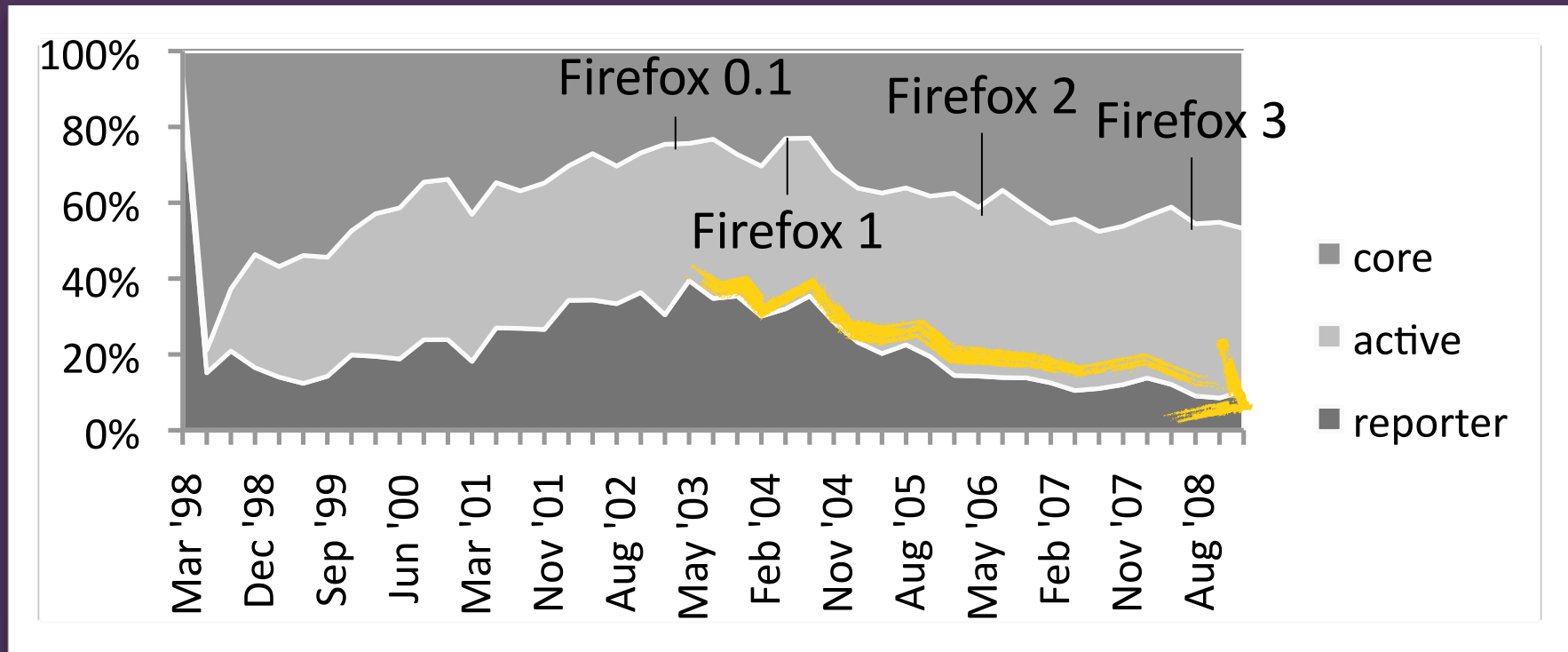
most REPORTER reports are not FIXED

% reports FIXED by each type



REPORTERS have dropped in effectiveness

% reports FIXED by each type



REPORTERS have dropped in effectiveness

why are REPORTERs ineffective?

sampled and categorized **100 reports** of each resolution type...

most REPORTER reports = technical support for **power users' tinkering and using old builds**

rarely provided **static** and **dynamic** context adequate to reproduce problems

reported problems, **resolved shortly after**

is open bug reporting useful?

yes, but ...

- significant overhead to process bad reports
- only a **skewed subset** of users report bugs
- users who report bugs are bad at providing the **static** and **dynamic** context of problems
- **text is a terribly imprecise medium** for expressing this context

ongoing work

enabling **every user** to submit



precise
structured
aggregatable
bug reports
with **zero** training?

three takeaway points

software quality depends highly on the quality of **human communication** and **cognition**

human communication and cognition are **faulty and unreliable**

carefully designed **interactive tools** can compensate for these limitations

questions?