

where **user experience** and
software engineering meet

Andrew J. Ko



dub

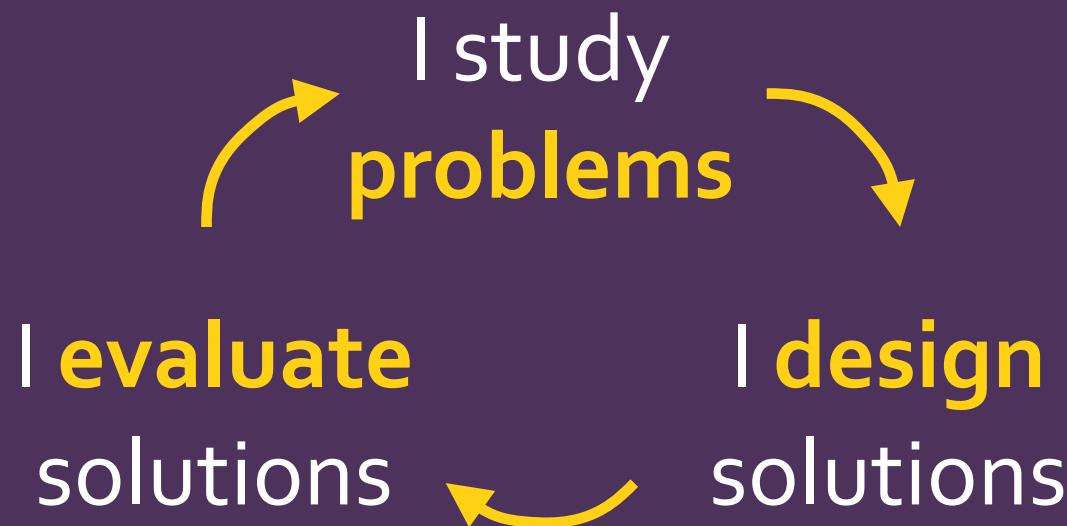
in method, I'm an HCI researcher

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 we get the
design right?

(the lives of developers)

user-centered software evolution



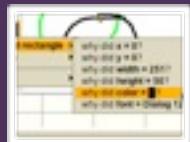
how do we get the
right design right?

talk outline



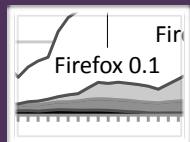
why is software evolution **difficult?**

a study of information needs at Microsoft



how can **tools** help software evolution?

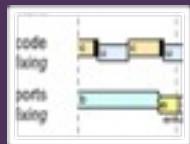
debugging with the Whyline



how can **users** help software evolution?

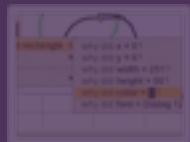
recent work in leveraging the crowd

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	Proceedings	Time	Estimated Event	Start	End	Scope	Notes
Pasha	Vistor	2:43 PM	P: So he's watching closely.				
Pasha	Vistor	2:43 PM	P: You know, looking & stuff, he's on research too, so the other half has it G...				
Pasha	Vistor	2:43 PM	P: So he's watching closely most of the time.				
Pasha	Vistor	2:43 PM	P: But then he's also... I mean, he's got his own research thread, which will be stuck, and it will be necessary.				
Pasha	Vistor	2:43 PM	P: So he's got his own research thread, which will be stuck, and it will be necessary.				
Pasha	Vistor	2:43 PM	P: No, I'm going to send you in New Eng...				
Pasha	Vistor	2:43 PM	P: I don't know if I have it right.				
Pasha	Vistor	2:43 PM	[Pasha laughs], and I'm gonna do the first assignment to the linked repository, but I don't really account for the whole picture.]				
Pasha	Vistor	2:43 PM	P: So there's a...				
Pasha	Vistor	2:44 PM	P: You can tell your research guys that product mode is a waste of ...			Project	Project
Pasha	Vistor	2:44 PM	P: So what do I mean here... bug with respect.			Spec/Trade	Spec/Trade
Pasha	Vistor	2:44 PM	P: I don't really know what's important, but usually, there's a crash, and we can't... sorry, but we don't have a stack dump, and we insert it into the debugger.			Situations	Situations
Pasha	Vistor	2:45 PM	P: But they put the method in the bug report file, so whenever I see bugs like [mentioning a C# method], I immediately scroll to C#PR and say, "Hey, look at this."			Ref/Source	Ref/Source
Pasha	Vistor	2:45 PM	P: It's a pretty cool tool.				
Pasha	Vistor	2:45 PM	P: See, then run these automated tools that are very intensive testing, and they obviously a crash, and it goes for 2 weeks, and we get bugs from a very old bugs tool [überfeste] [Überfeste] [überfeste] [überfeste] [überfeste].				
Pasha	Vistor	2:46 PM	P: [Pasha had a meeting at 2:45 so I think him for the time, take a picture after dinner with product studio, and he sends me the template he referred to in the interview]				
Papa	Jeff	9:05 AM	J: 9:05 AM One of the main goals is Chalktalk. Used also for code reviews, like a 40% off time, but they've been downed to 24 hours. I guess, every check-in undergoes a code review, so we're in drags now.			Project	Project
Papa	Jeff	9:05 AM	J: This [bug]. I was going to review this change to the controls that we wrote.				
Papa	Jeff	9:05 AM	J: 9:05 AM in 24hrs, you have blood and sweat, and you do, so I have to point. That's				

17 developers
hard at work
across **25 hours**

E	Service packs	Reproduced failures
SDE, 2	Developers	2 in bug fixing

Visual Studio

Windows Vista

G service packs

Service packs end users mobile devices

Mobile devices

discussion boards

SDE educational

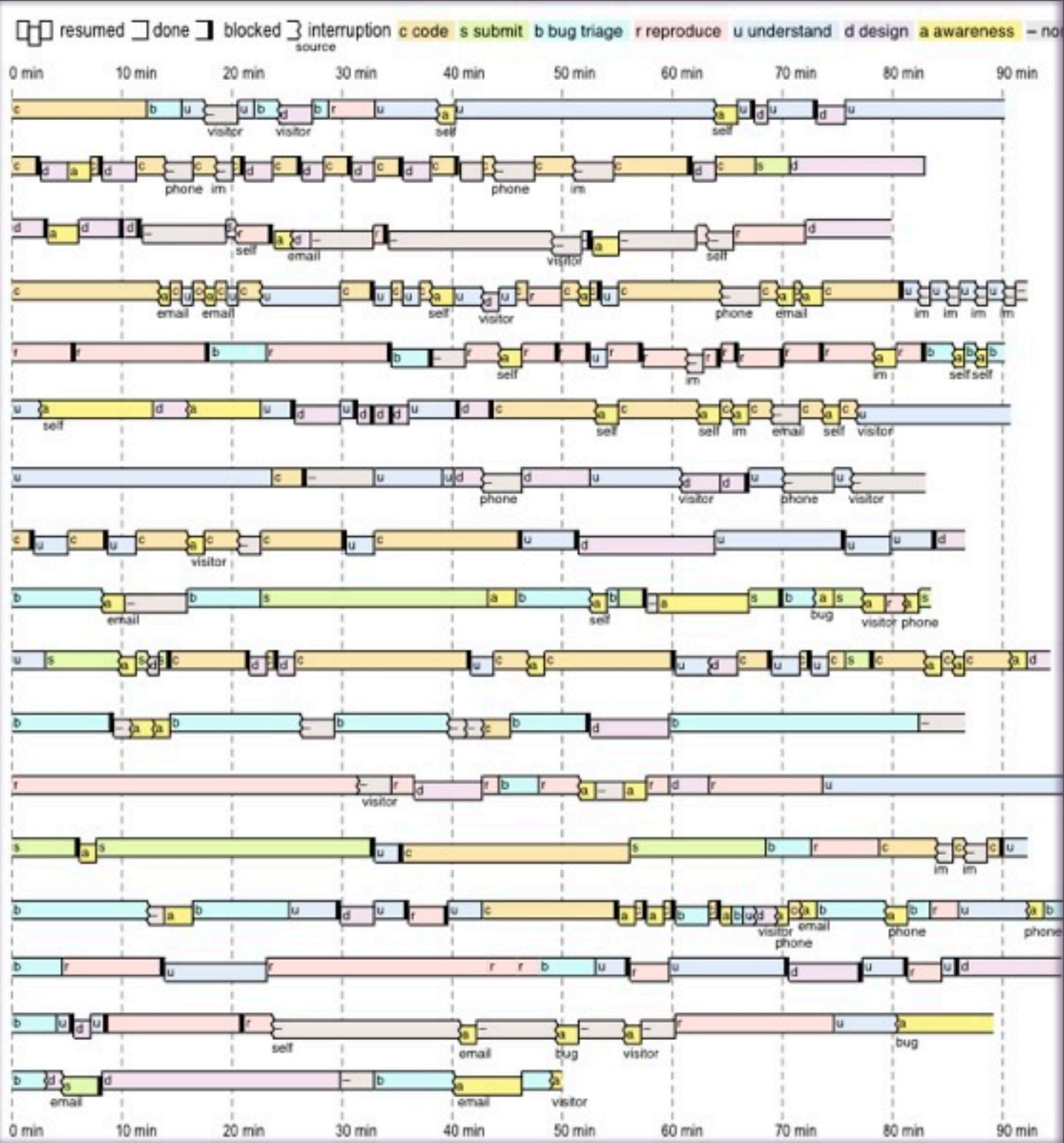
K SQL server

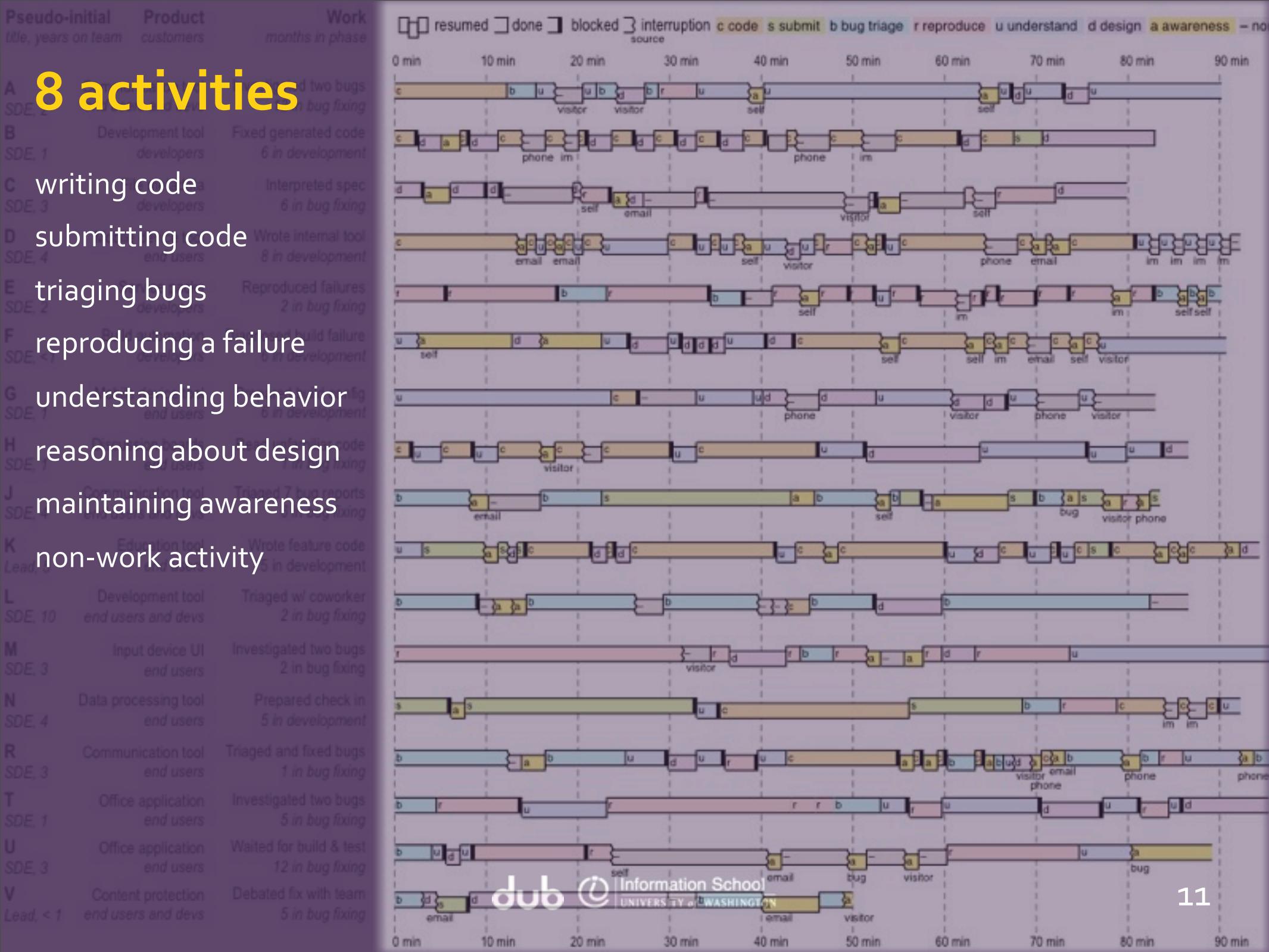
MS Office

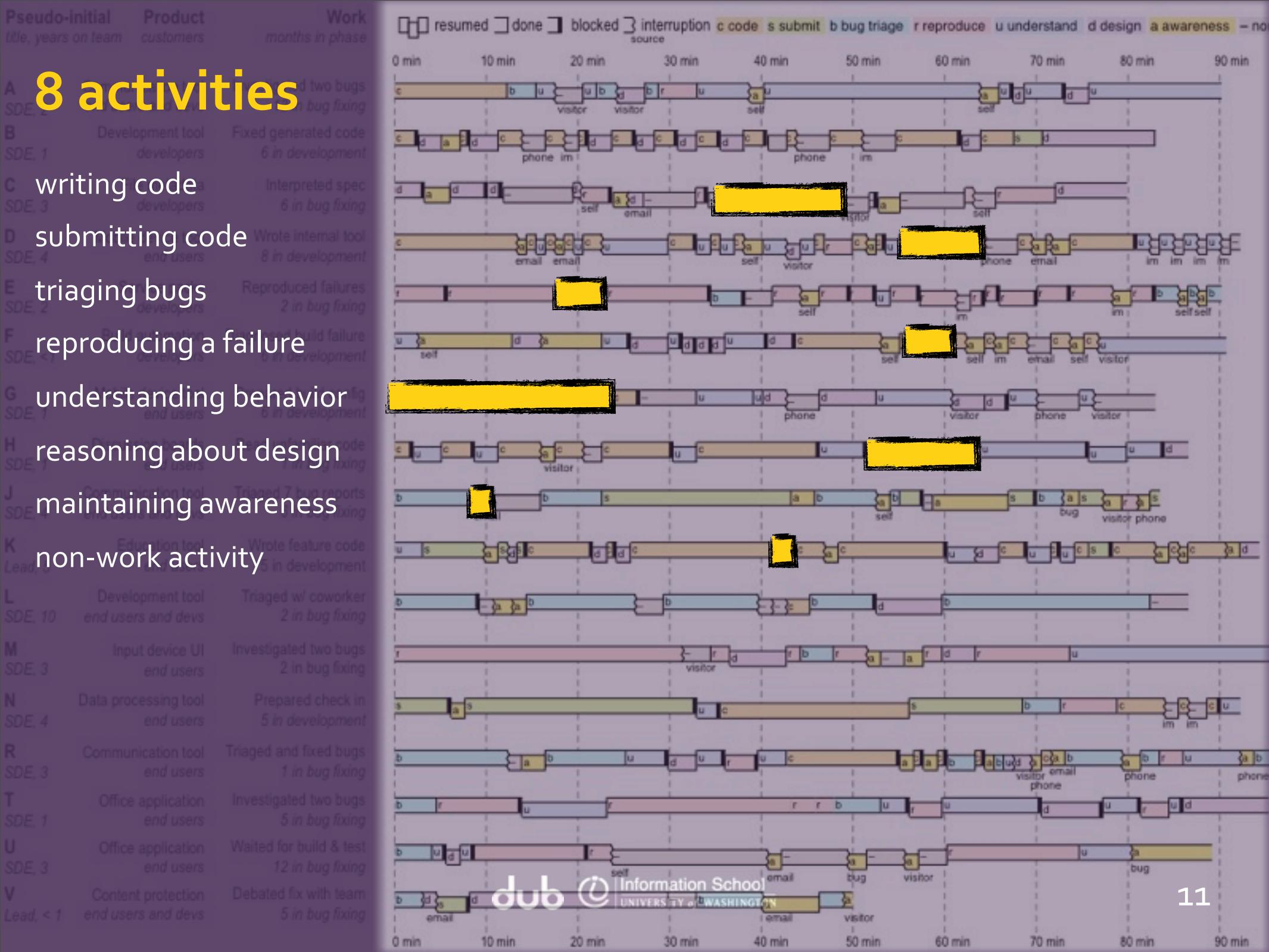
Encryption

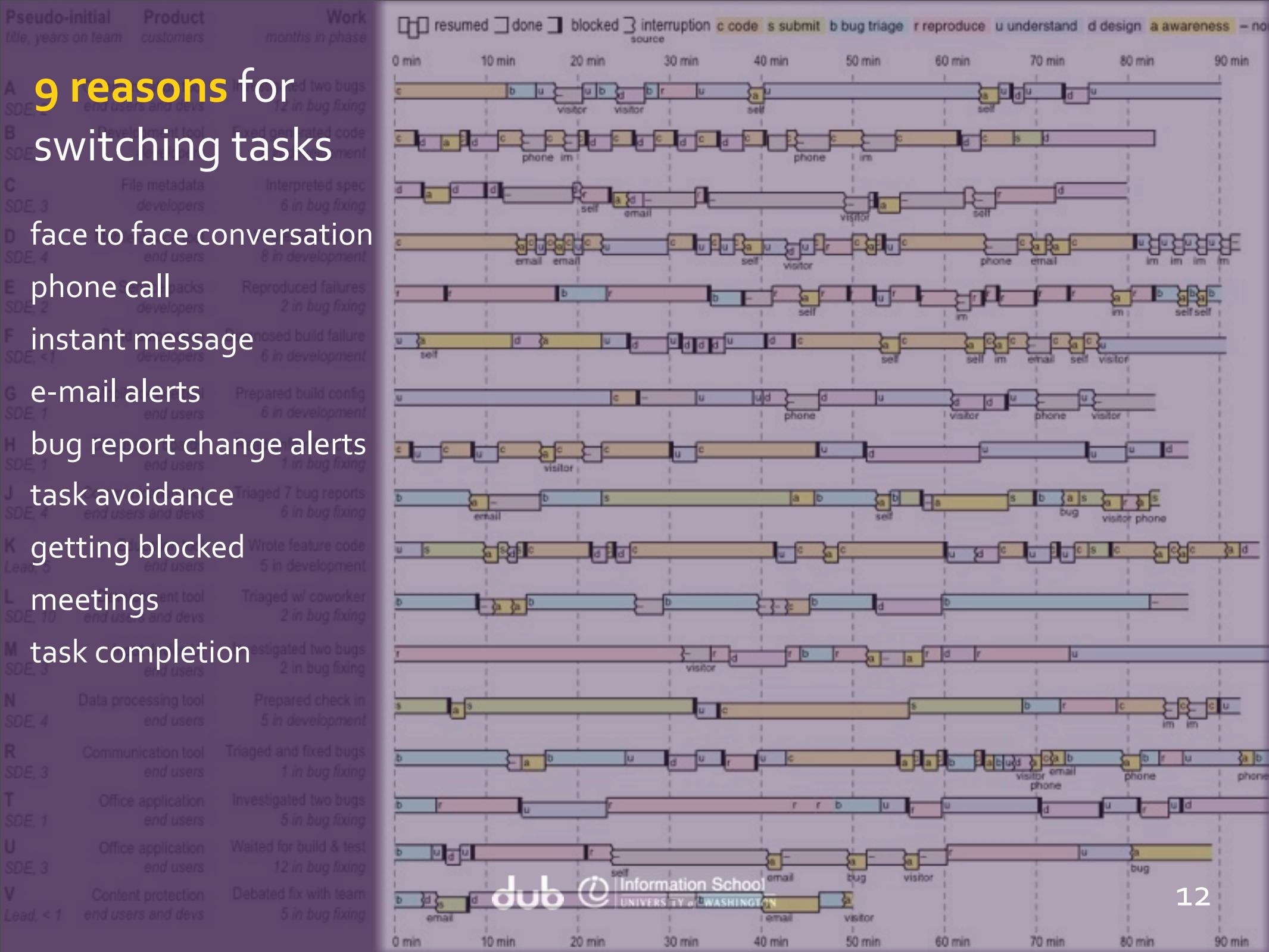
Encryption DRM

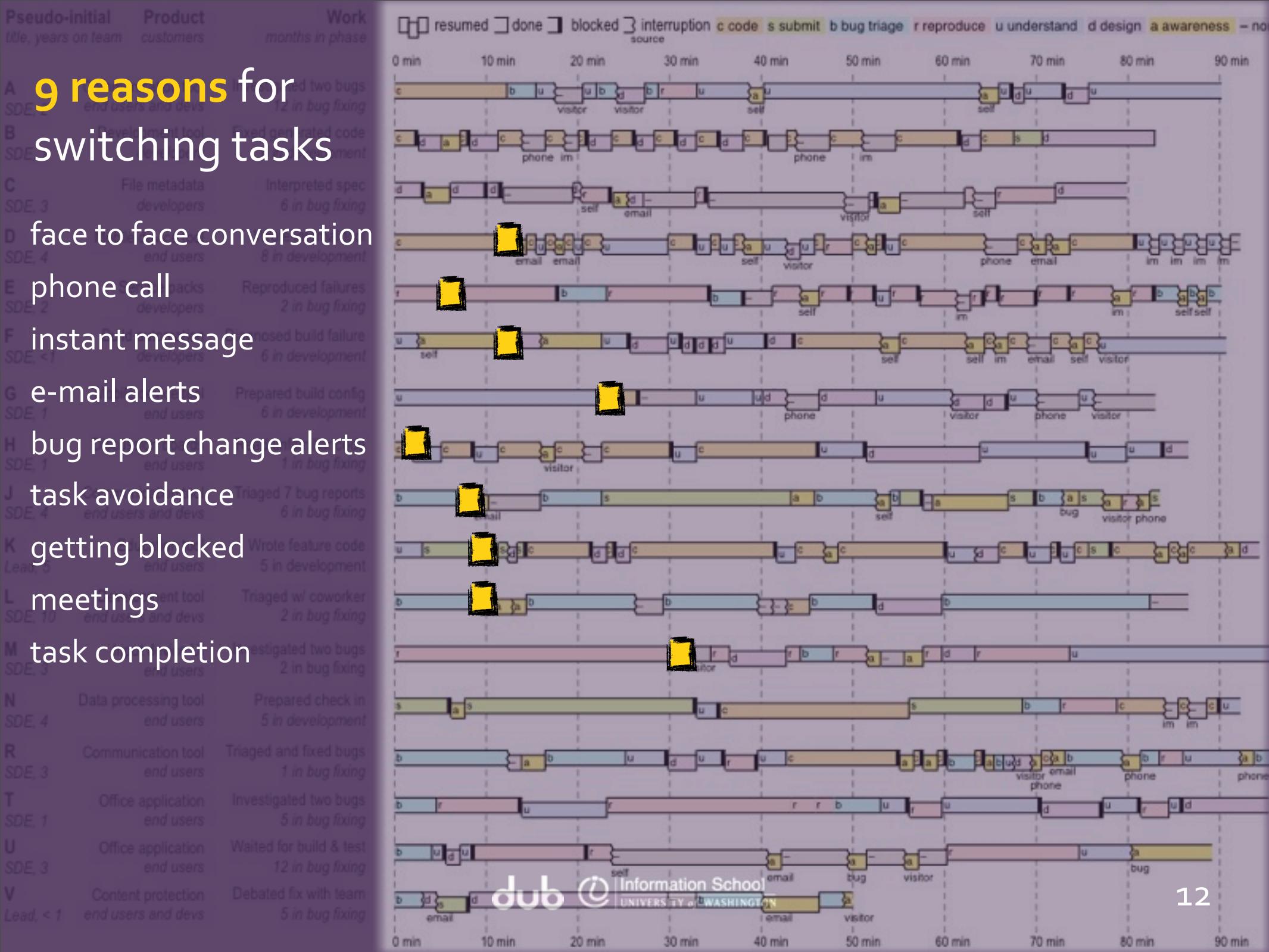
SDE, 3 DRM end user

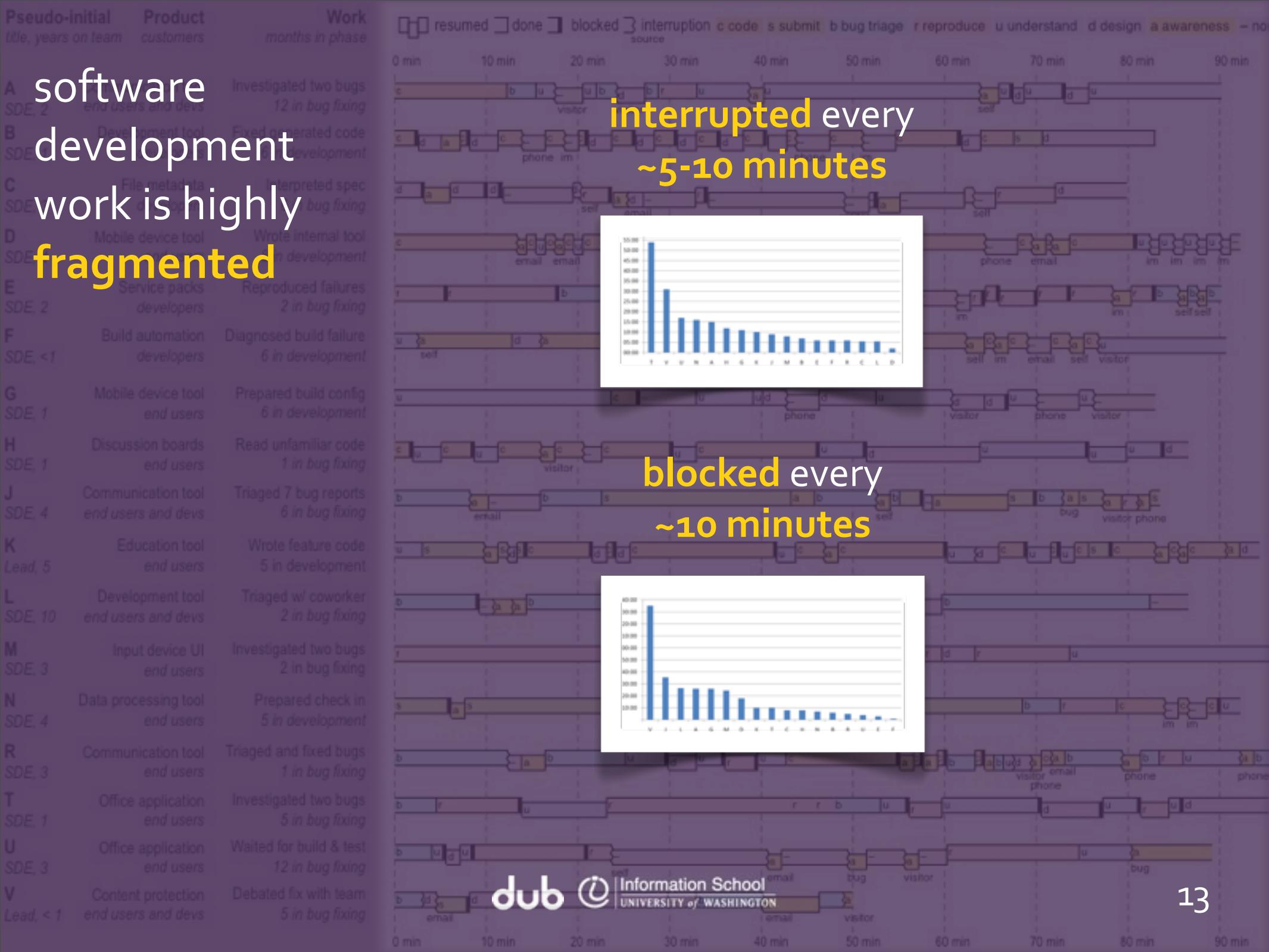


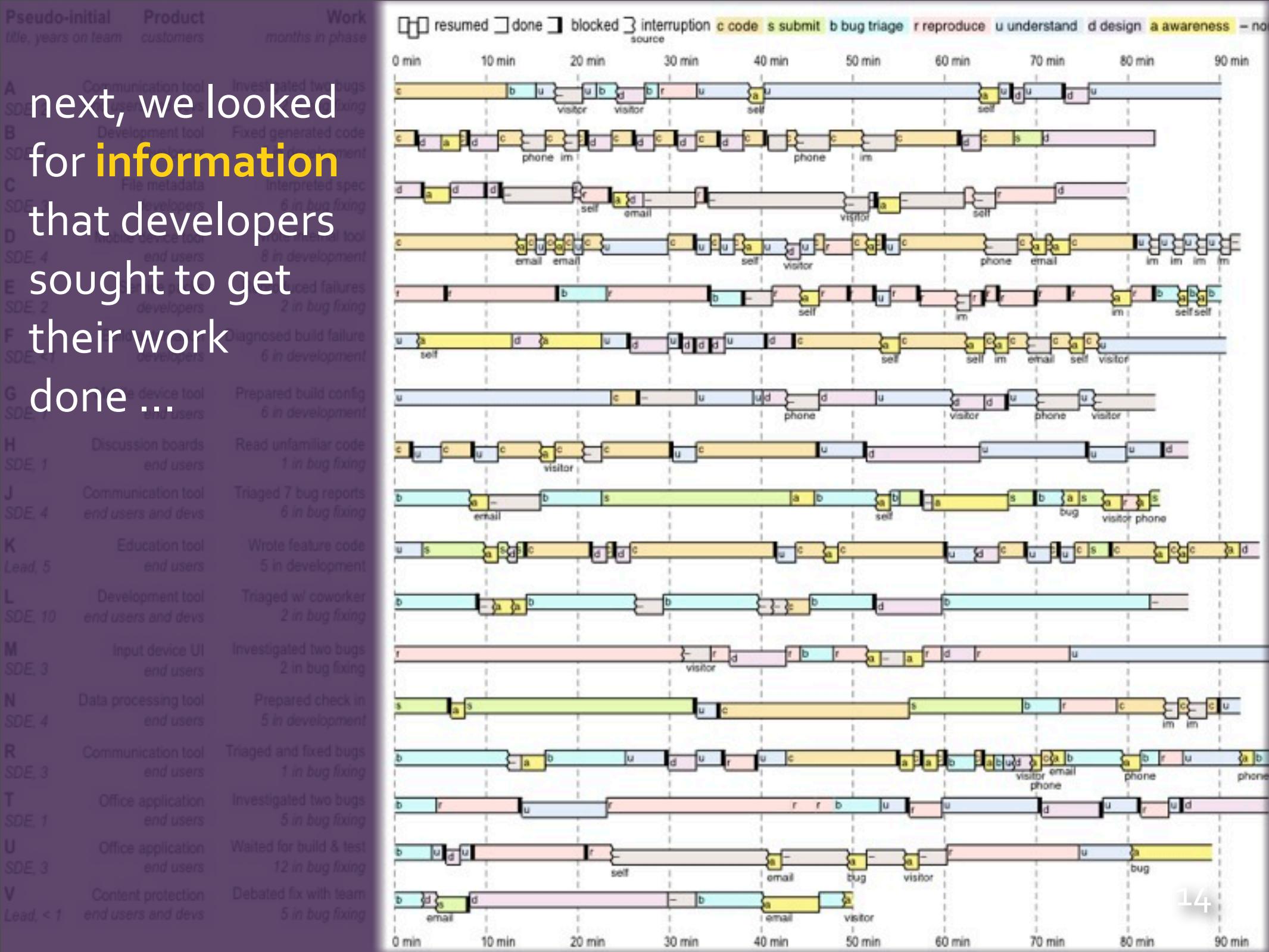




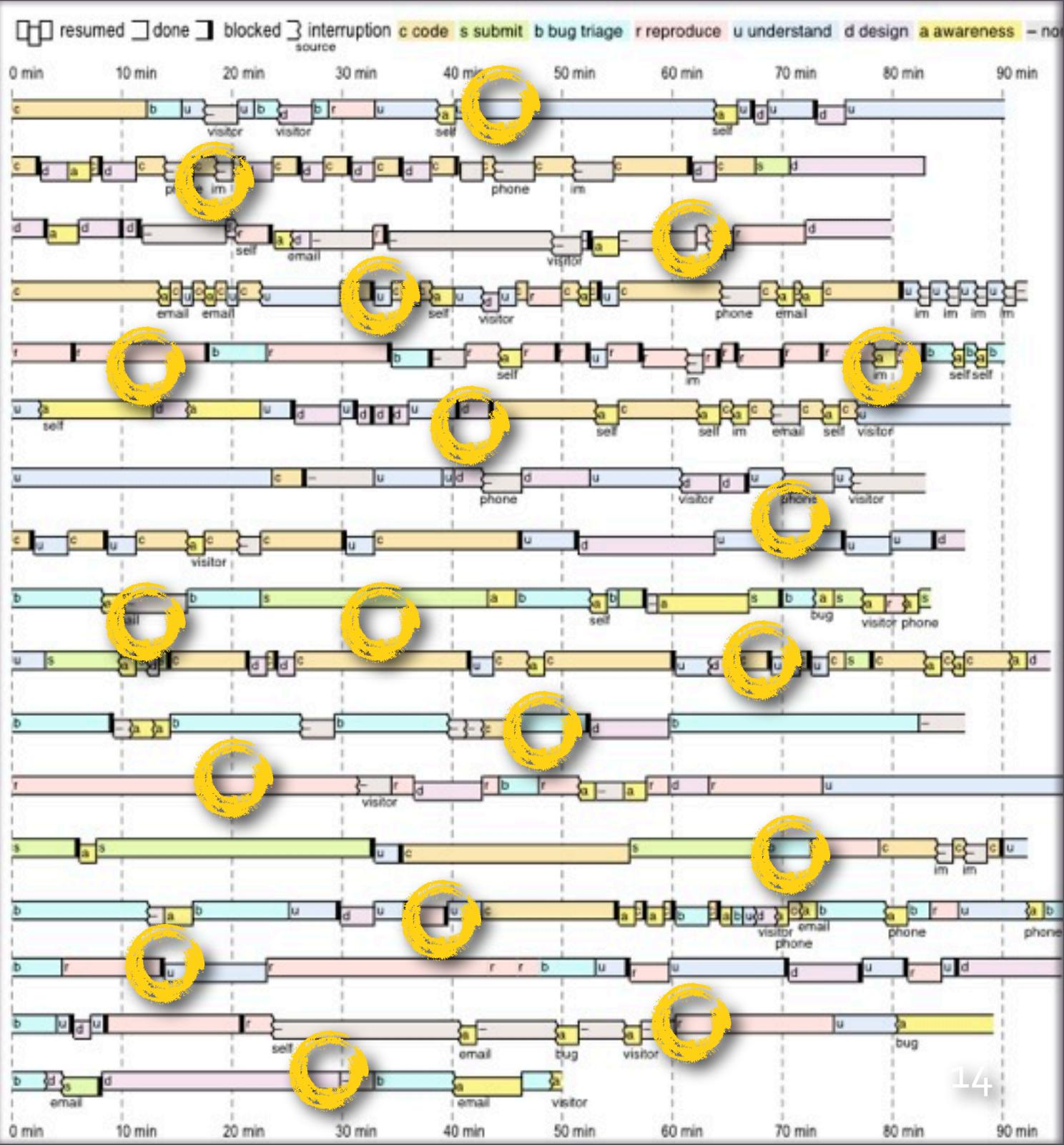








next, we looked
for **information**
that developers
sought to get
their work
done ...



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

what have my coworkers been doing?	% unsatisfied
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

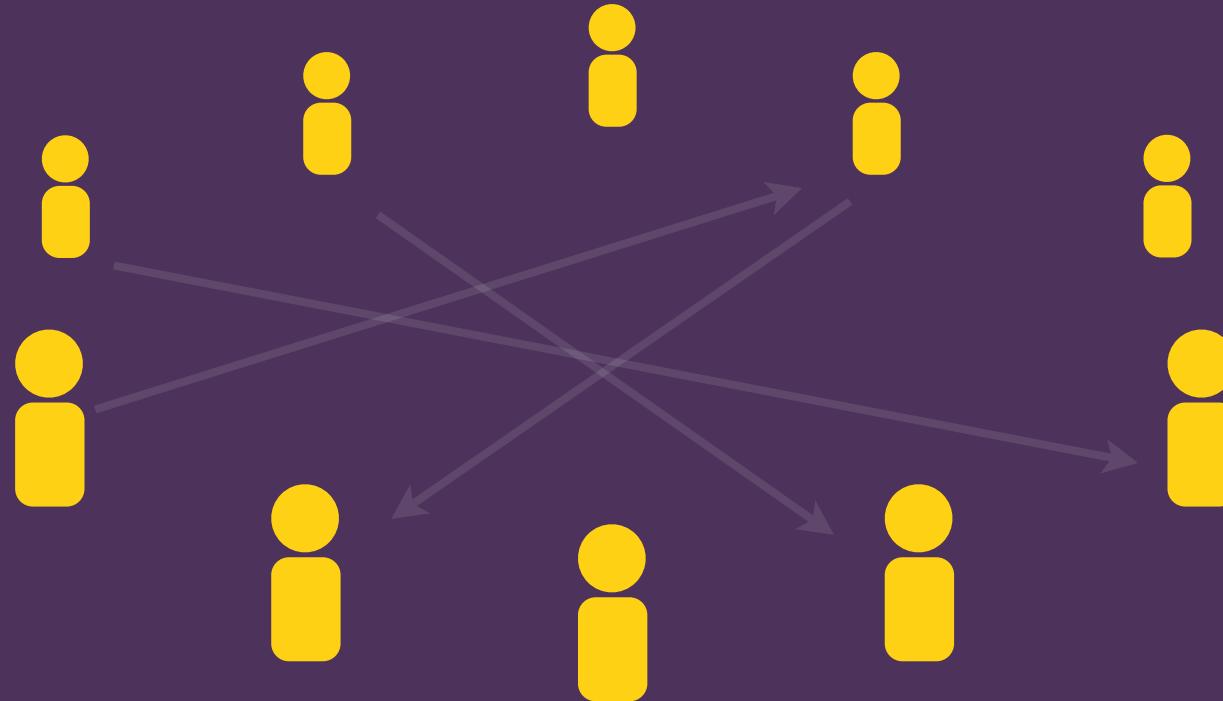
- what have my coworkers been doing?
- what code could have caused this behavior?
- have resources I depend on changed?
- what code caused this program state?
- how do I use this data structure or function?
- did I make any mistakes?
- what is the program supposed to do?
- in what situations does this 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**

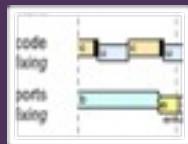




software quality depends highly on the quality of
human communication and **cognition**

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?

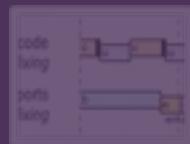
debugging with the Whyline



how can **users** help software evolution?

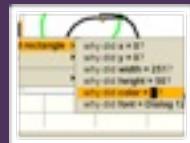
recent work in leveraging the crowd

talk outline



why is software evolution **difficult?**

a study of information needs at Microsoft



how can **tools** help software evolution?

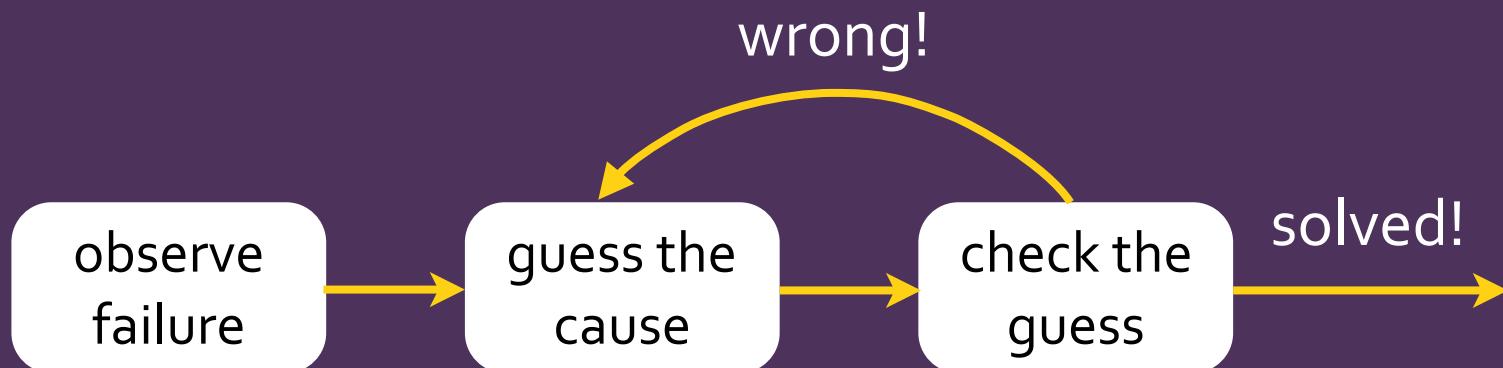
debugging with the Whyline



how can **users** help software evolution?

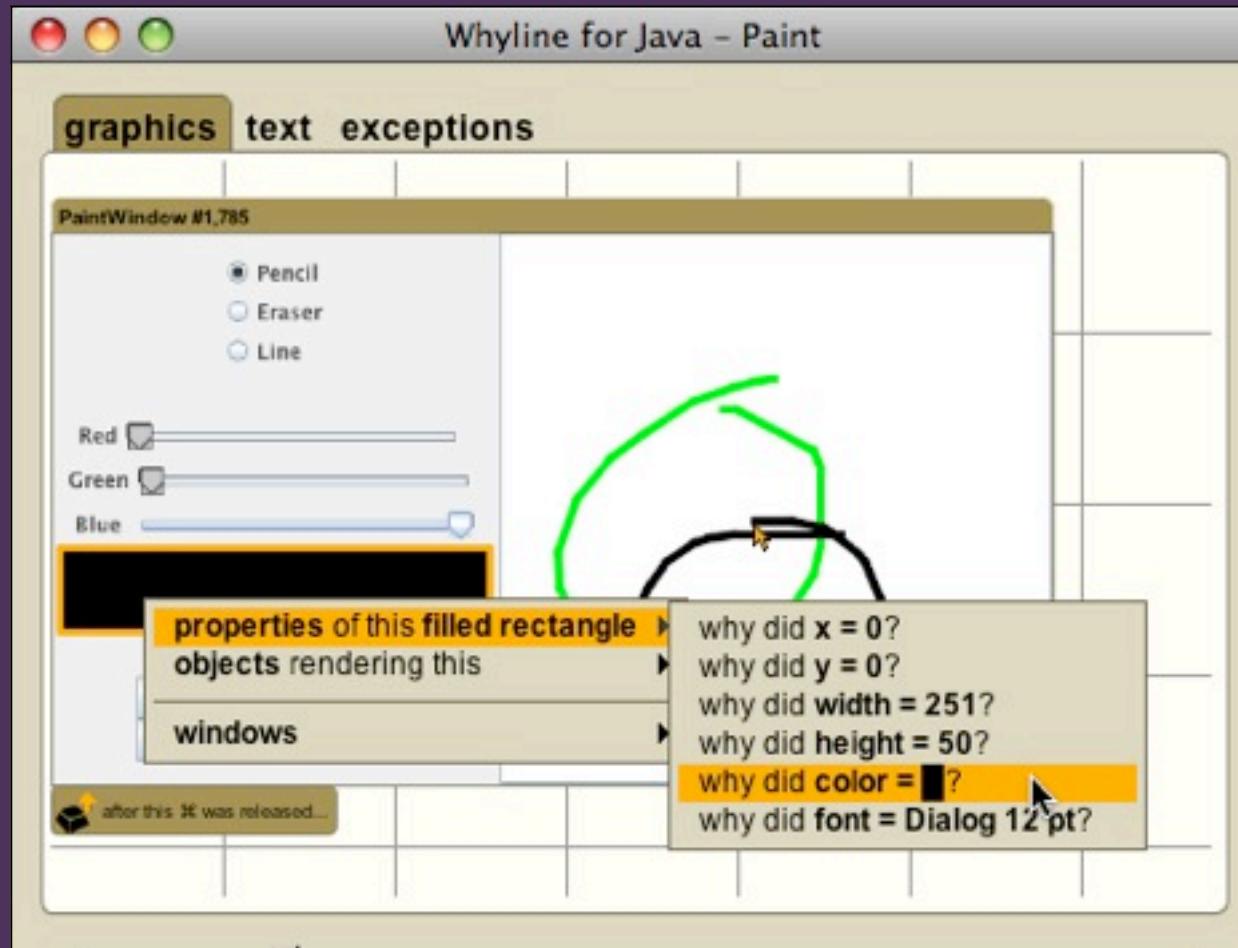
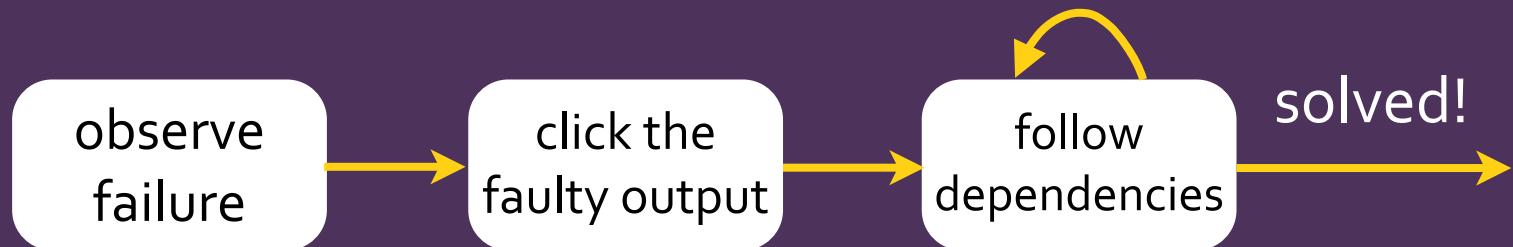
recent work in leveraging the crowd

why is **debugging** such a challenge?



Whyline

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

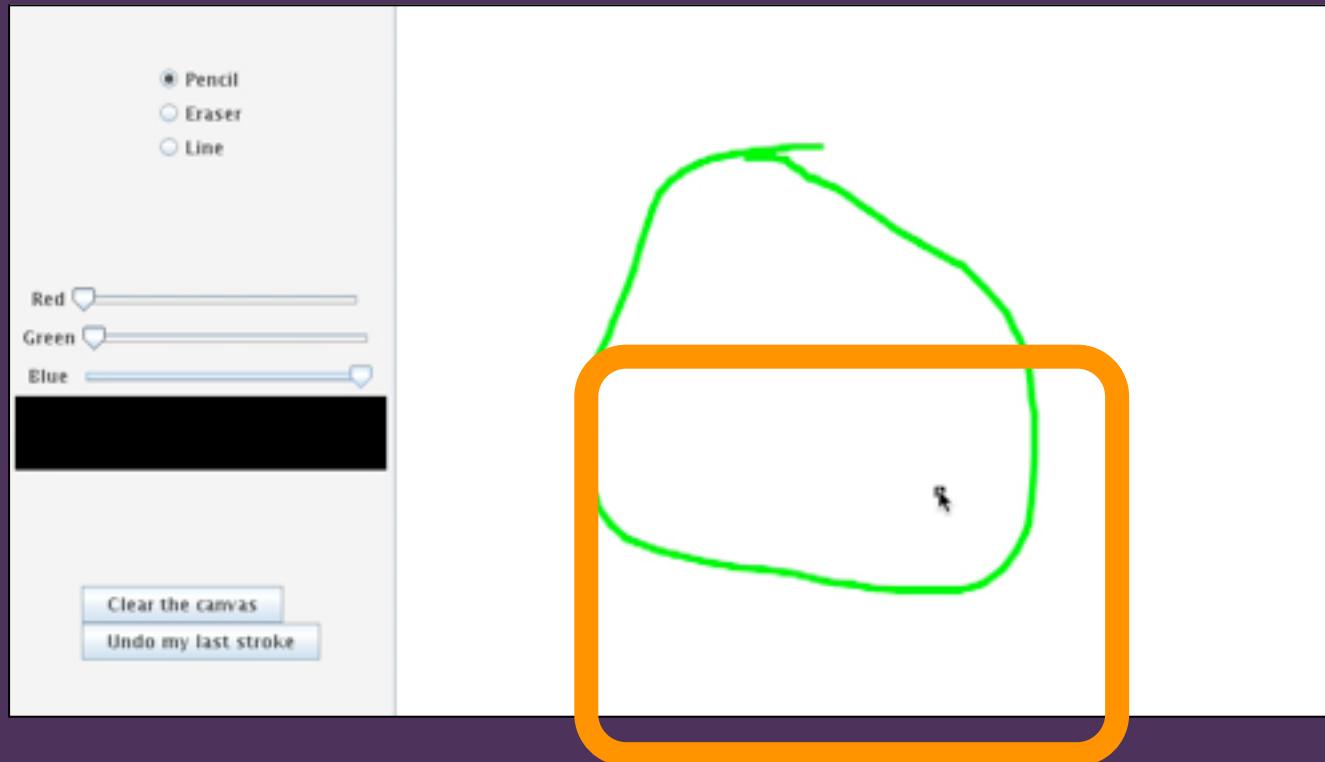


an example ...

why was the line black?

an example ...

why was the line black?

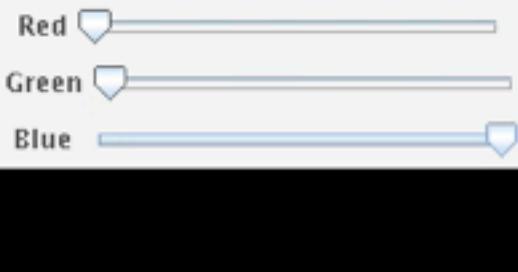


record the problem



25

- Pencil
- Eraser
- Line



[Clear the canvas](#)

[Undo my last stroke](#)

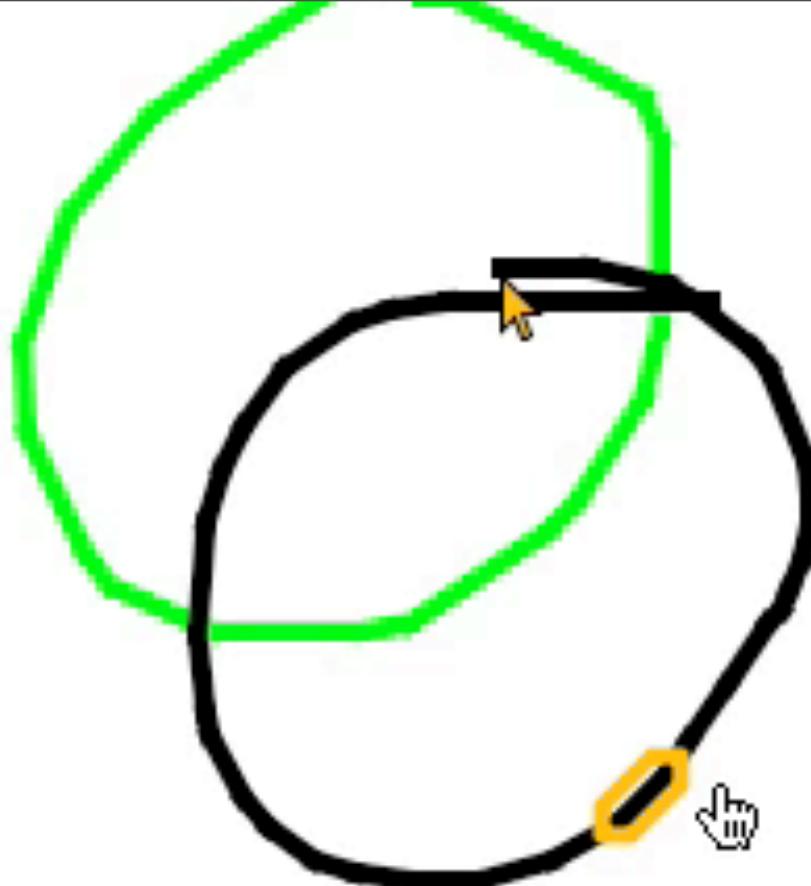
record the problem



Resolving classes (856 remaining)

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

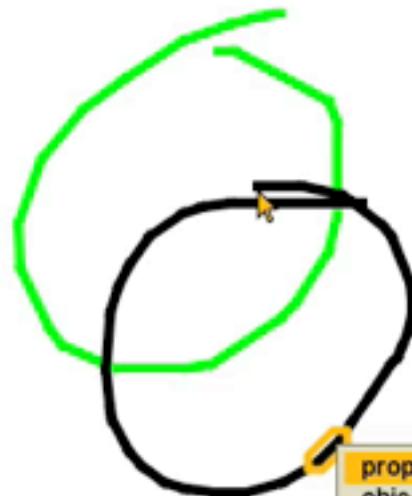
Green

Blue

Clear the canvas

Undo my last stroke

after this  was released...



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)*

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 }
```

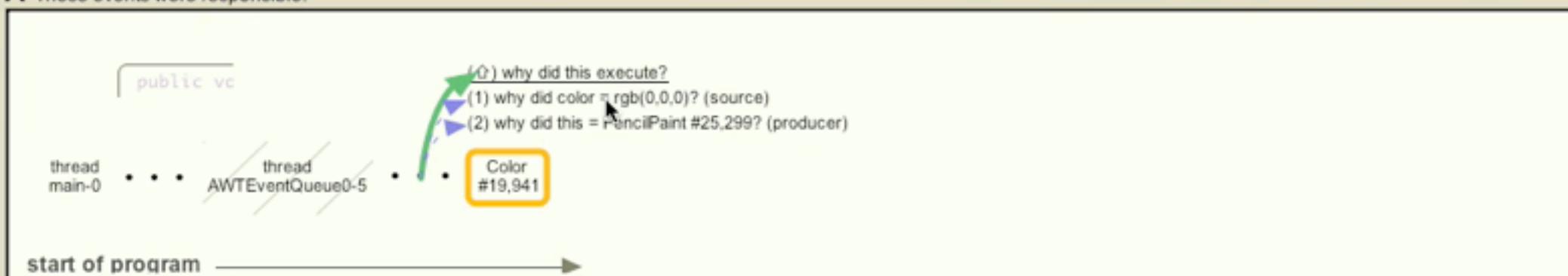
PencilPaint #25,299's field color was Color #19,941
(?) why did this execute?
(1) why did color = rgb(0,0,0)? (source)
(2) why did this = PencilPaint #25,299? (source)

PaintCanvas.java

Q why did color = ?

A These events were responsible.

← → ← in → in ← in → in ↑ collapse/ event event method method thread thread block show threads



why was the line color black?

```
one = points[pointIndex];
two = points[pointIndex - 1];
wLine((int)one.getX(), (int)one.getY(), (int)two.getX(), (int)two.getY())
ke(oldStroke);

```

followup questions about selected event

← →
event event

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

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

```
l : pointIndex--) {
```

```
two.getX(), (int)two.getY());
```

why was the line color black?

The diagram illustrates a Java code editor interface. A central white box contains the code for a class named 'Line'. The 'paint(Graphics2D g)' method is highlighted with a yellow box around its opening brace. Inside this method, several lines of code are underlined in blue: 'oldStroke = g.getStroke()', 'g.setStroke(new BasicStroke(thickness))', 'g.setColor(color)', 'Point one = points[pointIndex]', 'Point two = points[pointIndex - 1]', and 'g.drawLine((int)one.getX(), (int)one.getY(), (int)two.getX(), (int)two.getY())'. A green arrow points from the word 'selected' to the underlined 'oldStroke' line. Another green arrow points from the word 'dependency' to the underlined 'BasicStroke' class name. A third green arrow points from the word 'highlighted' to the yellow-highlighted opening brace of the 'paint' method.

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

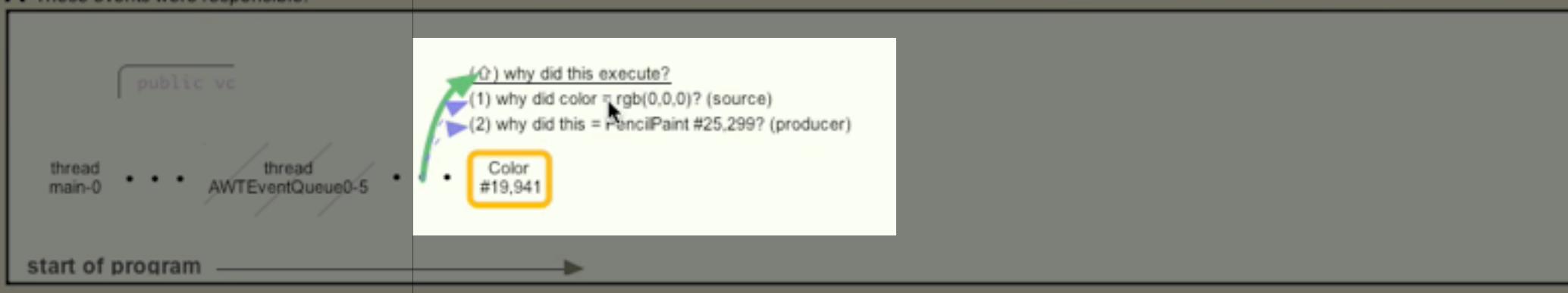
selected dependency highlighted in source

PaintCanvas.java

Q why did color = ■?

A These events were responsible:

					collapse/ expand
event	event	<i>method</i>	<i>method</i>	<i>thread</i>	<i>thread</i>



why was the line color black?

```
public Rectangle getBoundingBox() {
```

PencilPaint.java

```
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 }
```

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) {
```

PaintWindow.java

```
27
28
29     objectConstructor.setColor(
30         new Color(
31             rSlider.getValue(),
32             gSlider.getValue(),
33             bSlider.getValue()));
34 }
```

Q why did color = ■?

A These events were responsible.

← →
event event

← in → in
method method

← in → in
thread thread

-collapse/
block

expand/
threads

thread
main-0

thread
AWTEventQueue-5

Color
#19,941

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

start of program →

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(
```

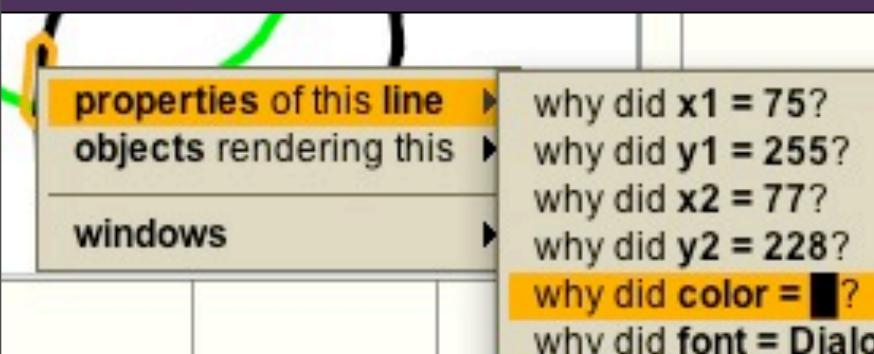
```
                rSlider.getValue(),
```

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

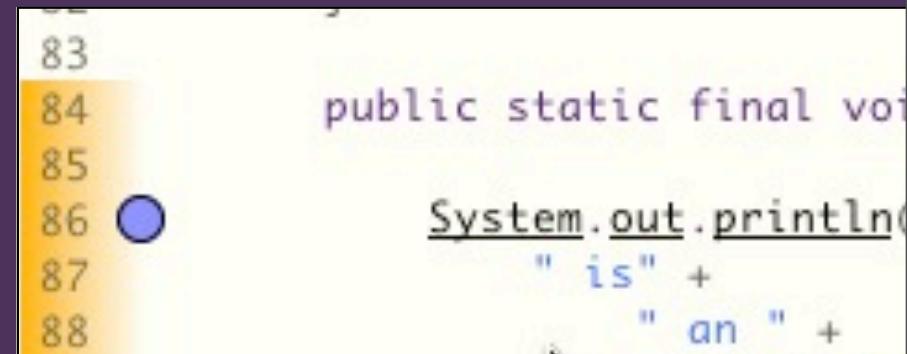
```
                gSlider.getValue()));
```

why was the line color black?

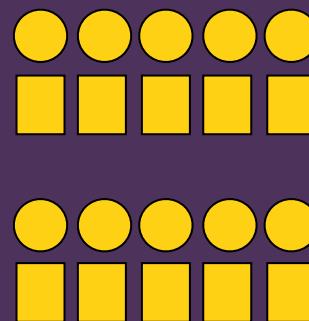
a comparison study



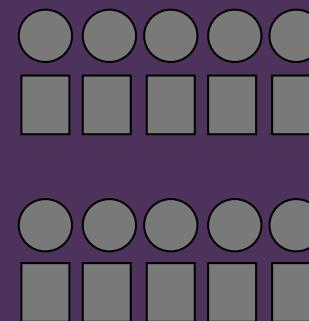
VS



Whyline
group



control
group



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

subject program

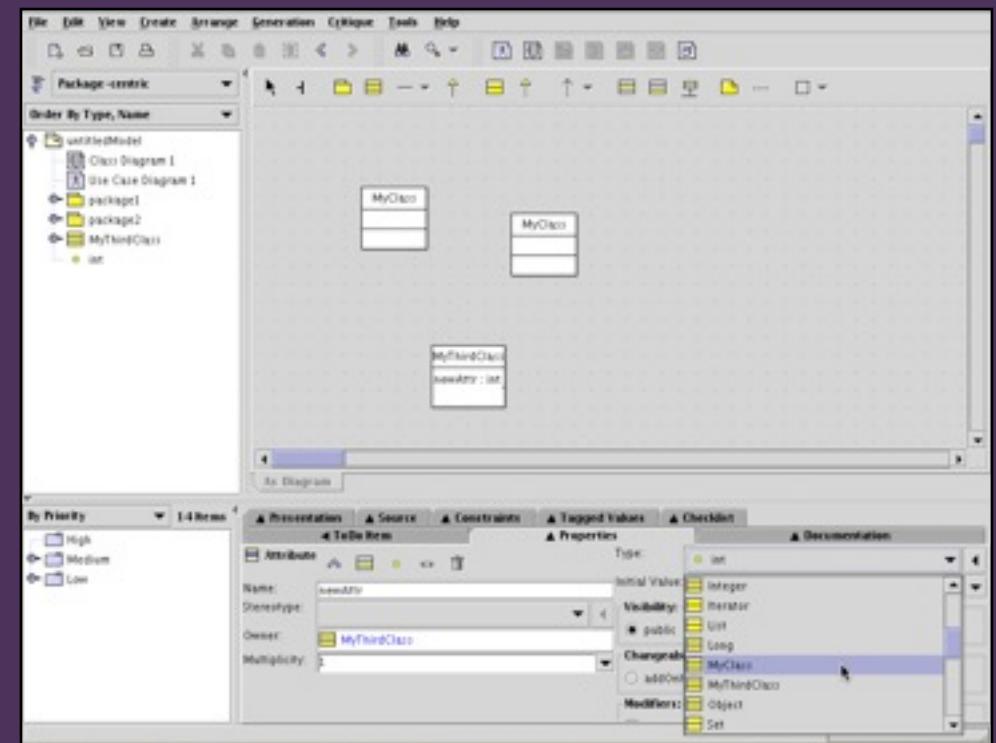
ArgoUML, an open source software design tool

~**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

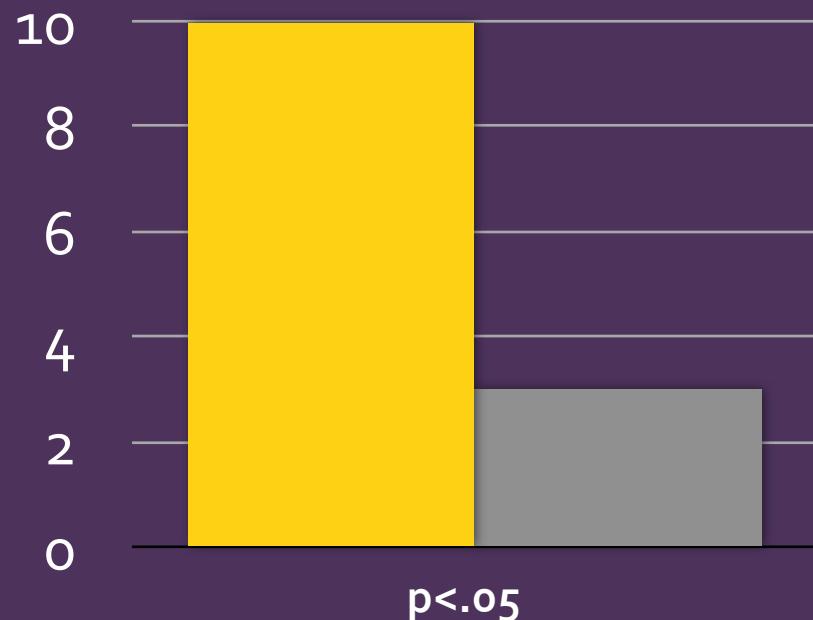


whyline

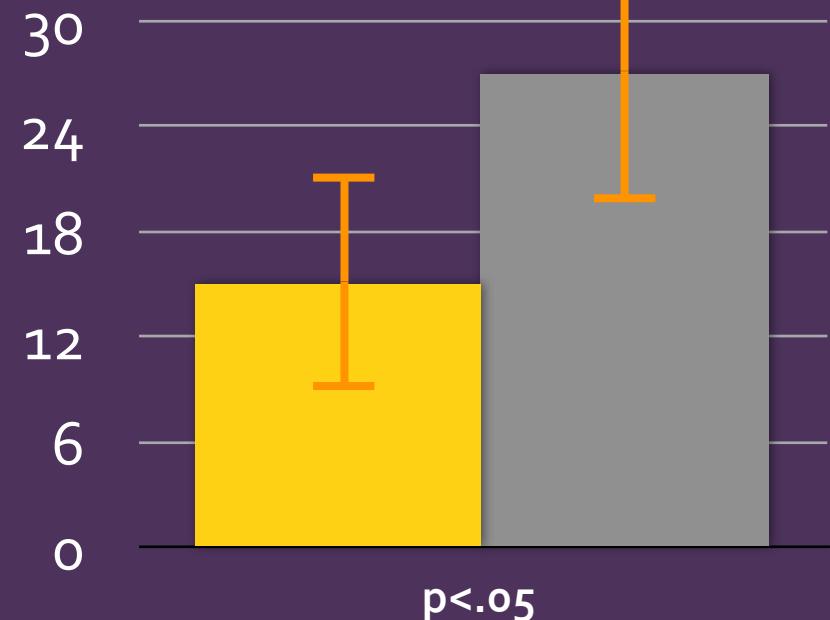


control

successful



minutes



more successful in half the time

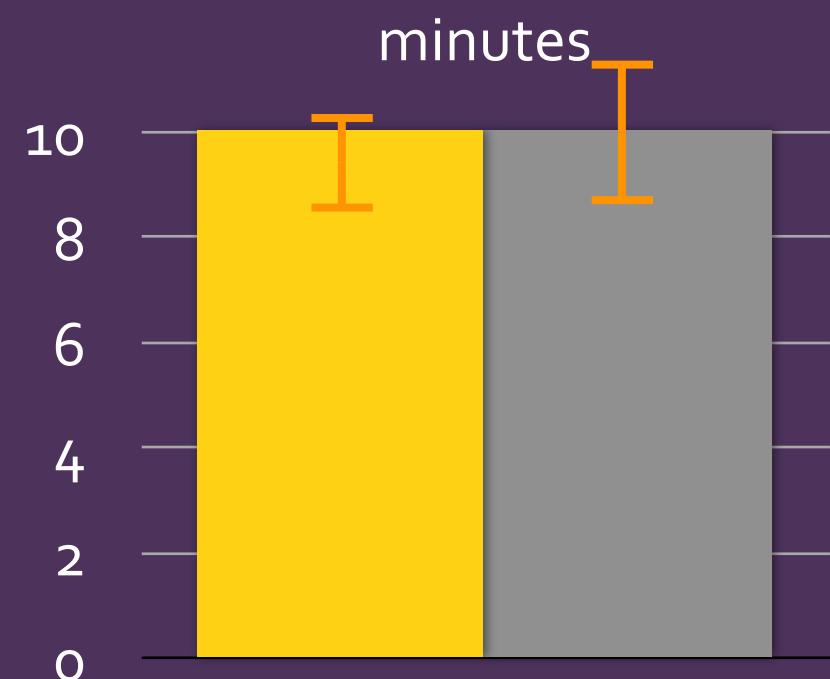
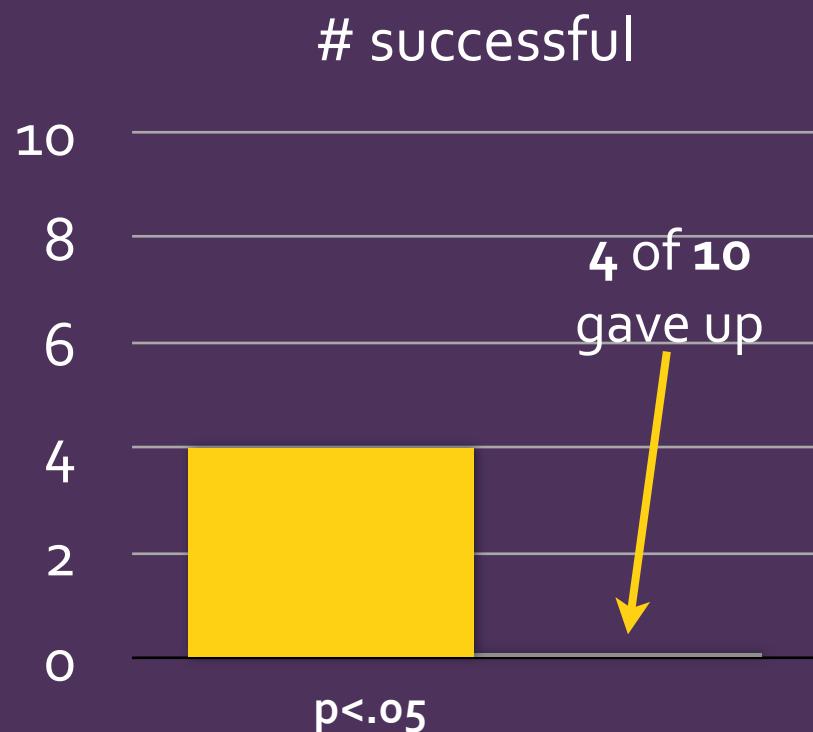
bug 2 results



whyline



control



more successful in the same time

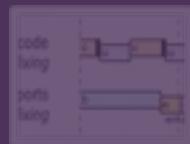
unsolicited quotes from users

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

“My god, this is so cool.”

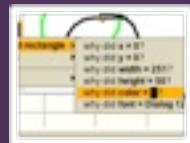
“It's so nice and straight and simple...”

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?

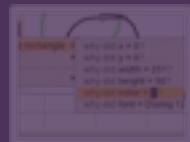
recent work in leveraging the crowd

talk outline



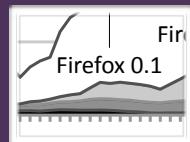
why is software evolution **difficult?**

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

1% **CORE devs** drivers, super reviewers, module owners, peers

1% **ACTIVE devs** assigned bug reports



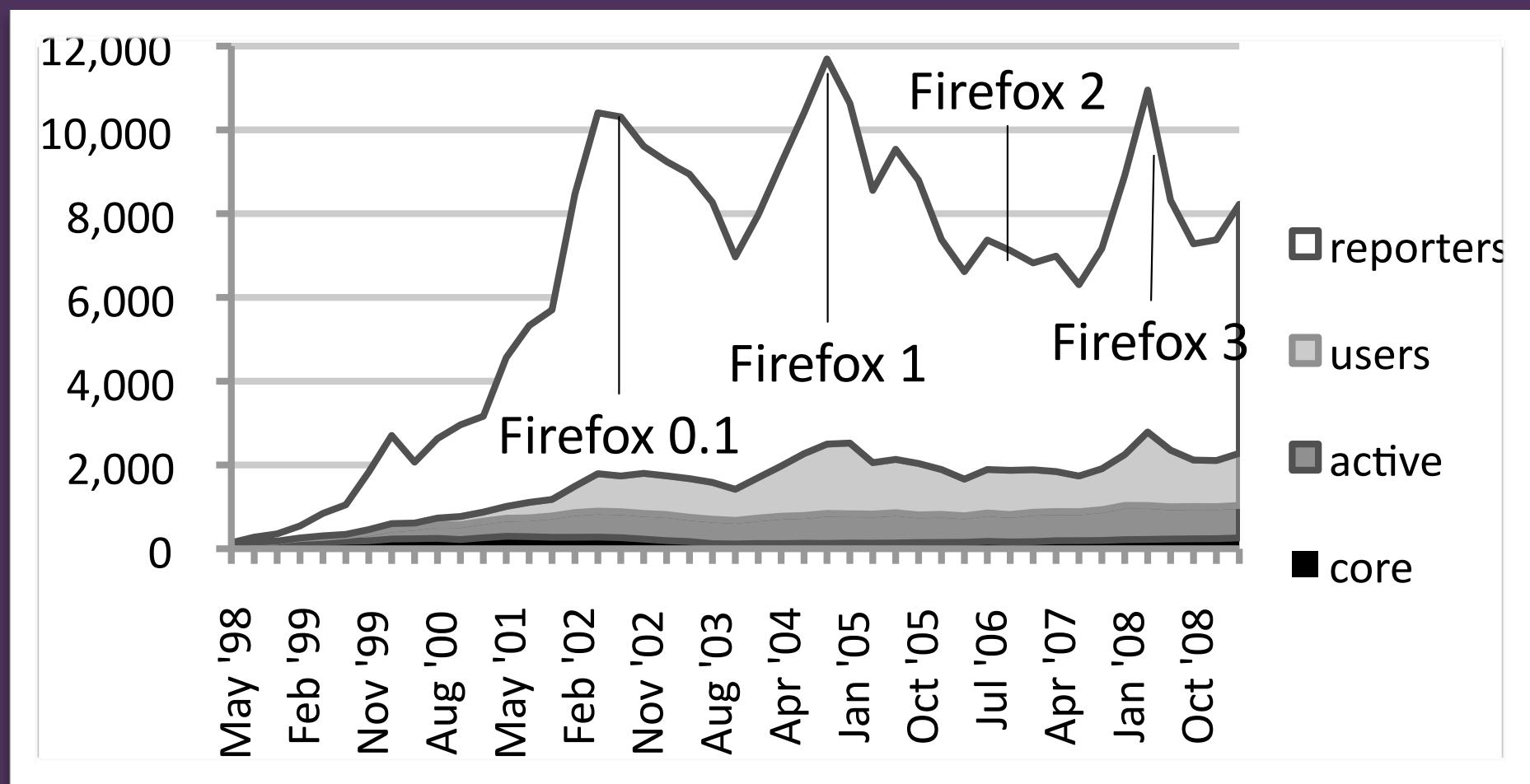
80% **REPORTERS**

reported and
commented on
bug reports

18% **USERS**

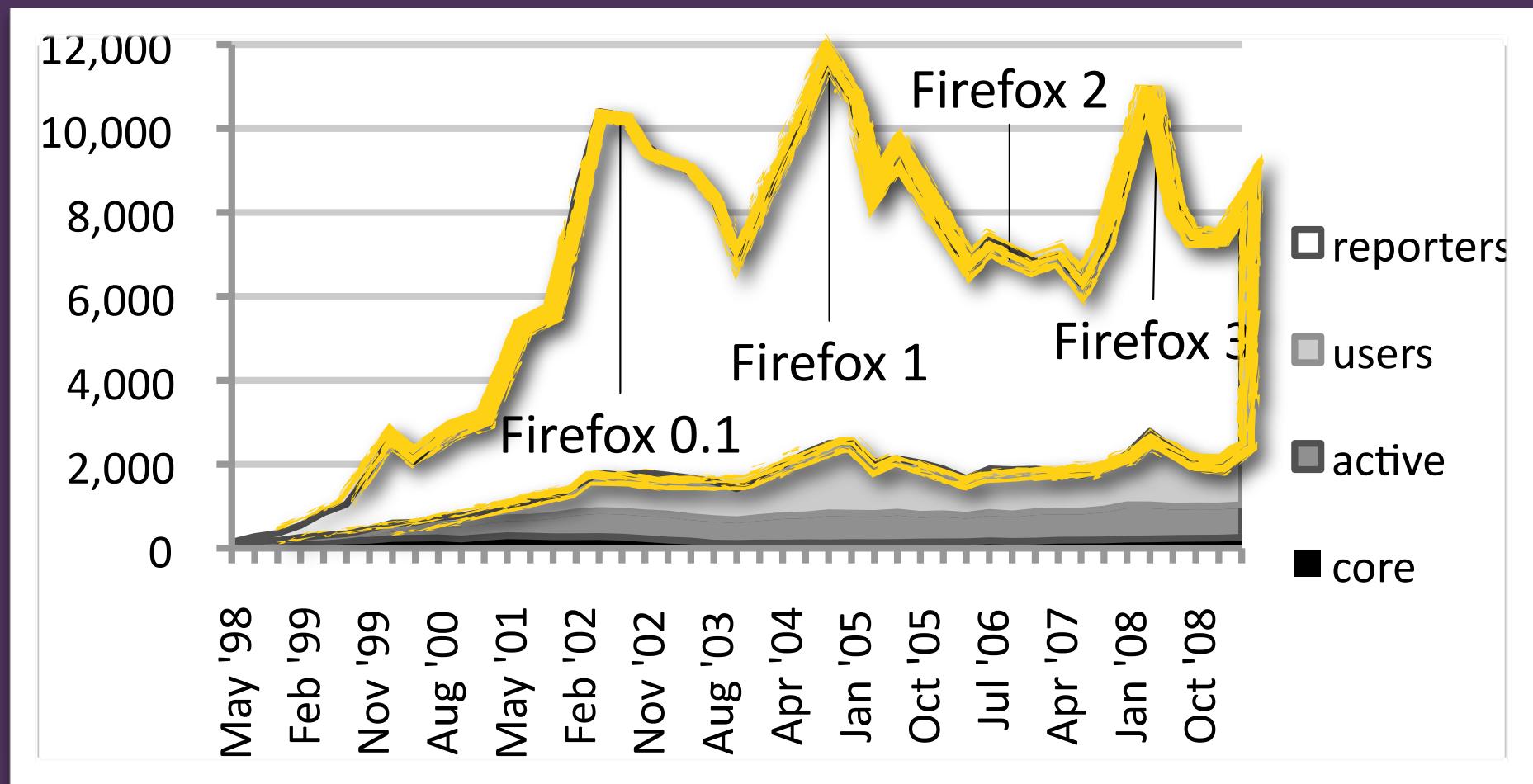
only commented
on bug reports

of comment contributors over time



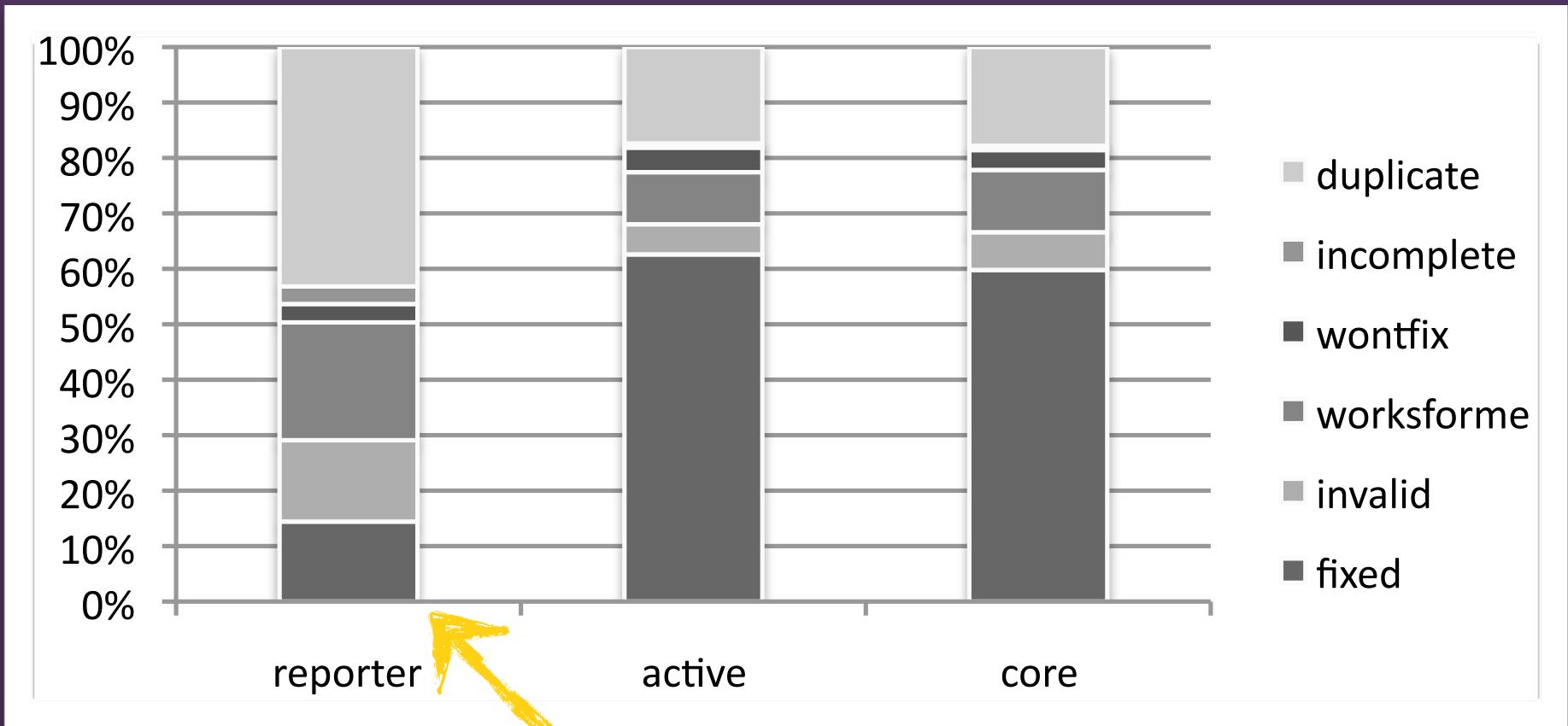
REPORTERs are the most active commenters

of comment contributors over time



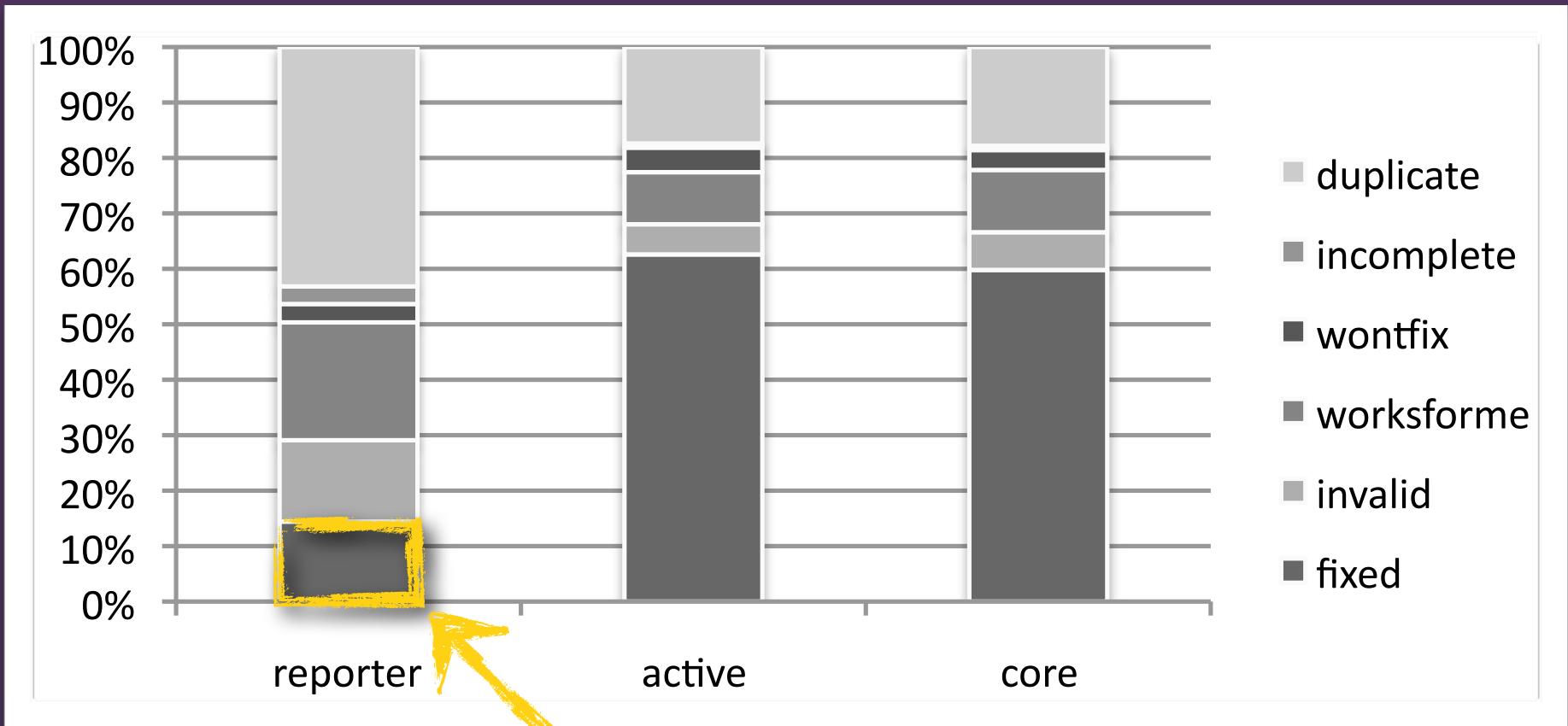
REPORTERs are the most active commenters

resolution by reporter type



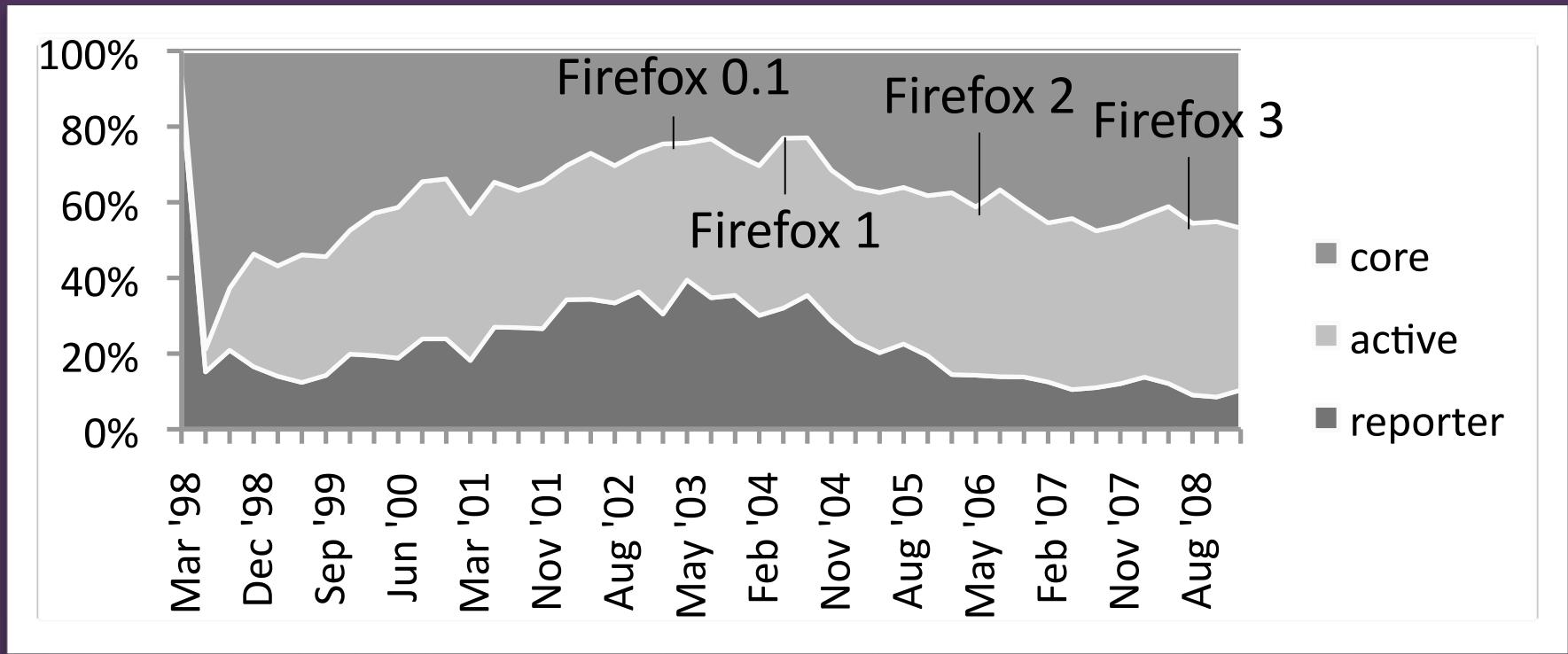
most REPORTER reports are not FIXED

resolution by reporter type



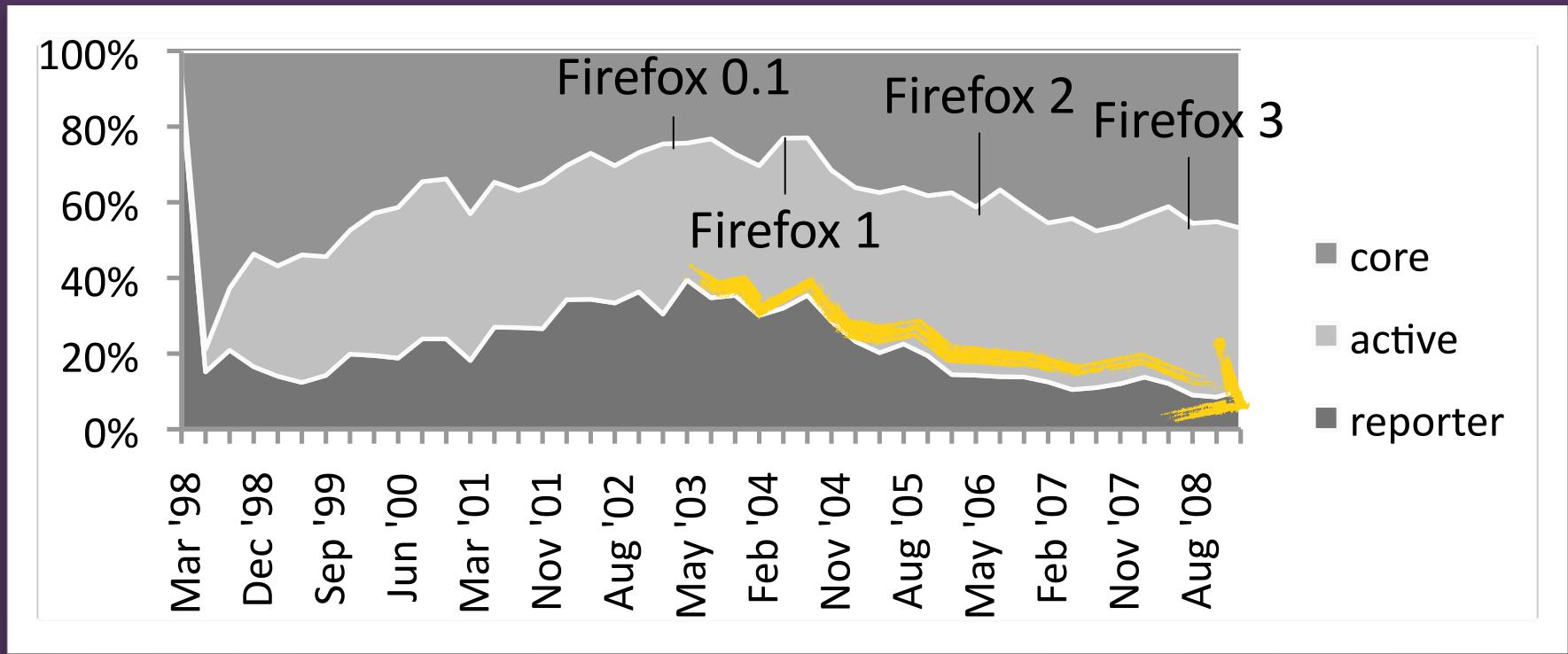
most REPORTER reports are not FIXED

% 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?