River birch root, Johnston Mill Nature Preserve

What The Hell Just Happened?

Better Bugs Make Better Programs

bug?
defect?
error?
fault?
flaw?
problem?

problem:
questionable property of program run

failure:
observable incorrect program behavior

defect:
incorrect program *code*

Better **Problem** Reports

Help Make Better Programs

Level -1: "why thing no go?"

https://www.schoolhouserock.tv/Interjections.html

Level 0 Structure

"I upgraded a library in my project **and** now I cannot connect to Datomic."

https://imgs.xkcd.com/comics/existential_bug_reports.png

Level 1 Structure

report element	contains
action	I upgraded foo 3.3 to foo 3.4
expected	connect returns a connection
actual	connect throws

One-Shotting

Problem reporting is an iterative process.

Level 2 Structure

report element	**contains**
environment	OS, libs, all processes
~~action~~ *history*	I upgraded foo 3.3 to foo 3.4 in process A
expected	connect to process B returns a connection
actual	connect throws
evidence	verbatim observations, e.g. error and stacktrace
impact	any info to support prioritization

Good Reports

concise:
everything necessary to cause the problem
...and nothing else

precise:
details, not shorthand
observations, not hypotheses

I have made this longer than usual because I have not had time to make it shorter.

https://quoteinvestigator.com/2012/04/28/shorter-letter/

Problem reporting usually becomes a collaborative process.

Good Collaboration

sympathetic

positive

trusting

curious

precise

Common Mistakes

theorizing, esp. assuming the failure is the defect

suppressing information

omitting information*

misreporting

leaping into action

*not actually a mistake. Why?

It is a capital mistake to *theorize* before you have all the evidence. It biases the judgment.

https://en.wikipedia.org/wiki/Sherlock_Holmes

Example Mistakes

theorizing	connect throws -> Datomic bug
suppressing info	eliding stacktrace on connection failure
omitting info	no report of dependencies
misreporting	accidentally reporting data from a different environment
leaping into action	reconfiguring the system to try to get connection to work

Avoiding Mistakes

theorizing	just don't
suppressing info	just don't
omitting info	learn about the domain
misreporting	double checking extra eyeballs
leaping into action	just don't

Clojure/Datomic Specifics

most problems have small REPL reproductions

work from a consistent basis

provide doc strings for every var

name data, not just functions

minimize dependencies

minimize number of processes

Consistent Basis

```
(def conn (d/connect client {:db-name "solar-system"}))
(def db (d/db conn))

;; where did this data come from?
(d/q '[:find (pull ?e [*])
       :where [?e :data/source]]
     db)

;; how many objects are there?
(d/q '[:find (count ?e)
       :where [?e :object/name ?n]]
     db)

;; largest radius?
(d/q '[:find (max ?radius)
       :where [_ :object/meanRadius ?radius]]
     db)
```

one db value

https://github.com/cognitect-labs/day-of-datomic-cloud/blob/master/tutorial/aggregates.repl

Names and Docs for Data

```
(d/q '[:find (pull ?e [*])
       :where [?e :data/source]]
     db)
```

```
(def all-data-sources
  "Returns all attributes for all :data/sources"
  '[:find (pull ?e [*])
    :where [?e :data/source]])

(d/q all-data-sources db)
```

There Will Be a Test

problems are not bugs

use a template

describe, don't theorize

trust and be curious

be concise and precise

References

Simple Made Easy (InfoQ video)

The complete Sherlock Holmes Canon (text)

Debugging with the Scientific Method (YouTube)

Stewardship Made Practical (YouTube)

Computers Can Be Understood (text)

@stuarthalloway (me)

Brumley Nature Preserve