

# Unix Shell

We can do better now

We could do better since the 70s ...  
and we didn't

# About Me

- [Ilya Sher](#)
- Bash user – since around 1996
- DevOps & Programmer – professionally since 2001
- Working on Next Generation Shell – since 2013

# What's wrong with the Unix shell?

- Programming language
- User interface

# Programming Language

- Syntax
- Error handling
- Lack of structured data

Being addressed by multiple modern shell projects ... differently from NGS

# User Interface

The Unix shell is stuck in telegraph-like communication paradigm:

*Send Text – Receive Text*





# User Interface – how we got here?

Same communication paradigm: *send text – receive text*

- Telegraph
- Teleprinter (aka teletype, aka TTY) →
- Teleprinter - connected to computer
- Video Display Unit

Incremental improvements without conceptual breakthrough



# User Interface – how we got here?

Technological breakthrough – VT52 in 1974/1975

Cursor movement support.

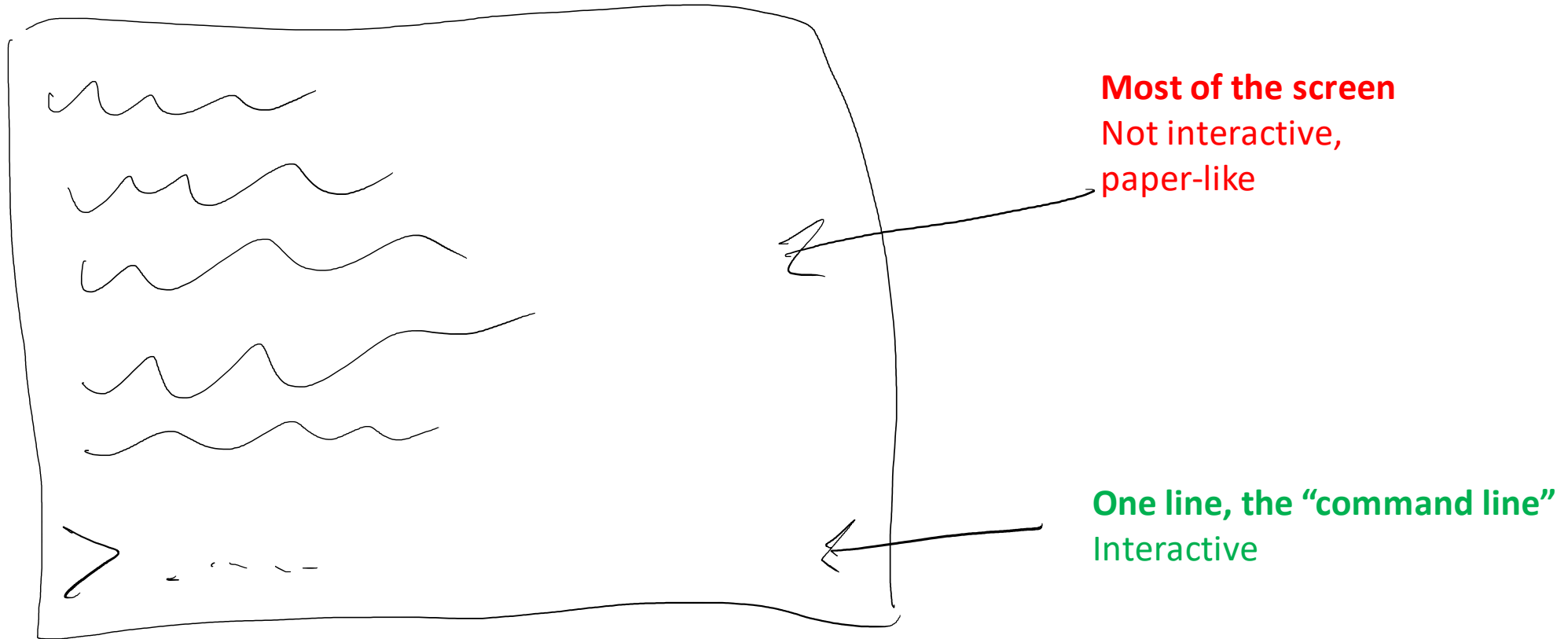
Reaction:

- [Bill Joy](#) releases [vi](#) in 1976
- Unix shell – ignores this capability until this day (mostly)

*We could do better since this point*

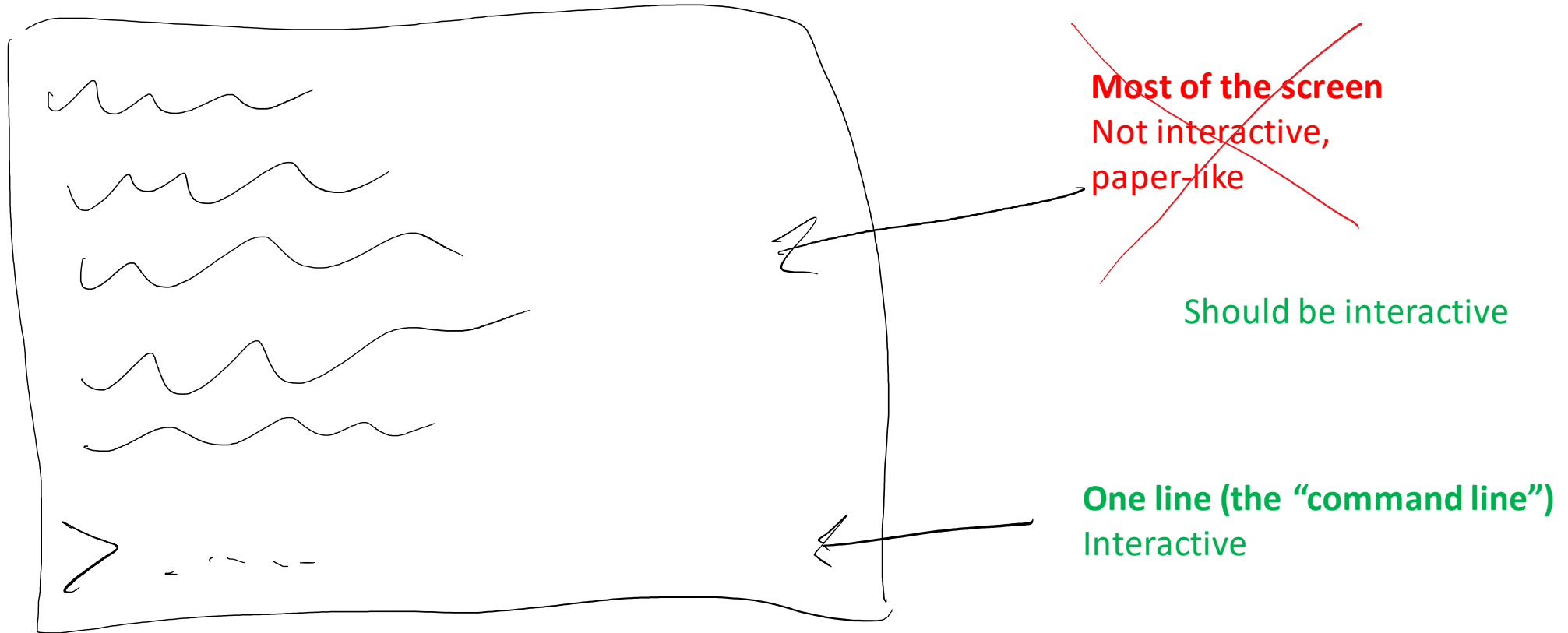


# User Interface - “Interactive Shell”





# User Interface - “Interactive Shell”



# User Interface – Objects on the Screen

- type = file, id = /Users/blah/file.txt , name = file.txt
- type = AWS::CodePipeline, id = ..., name = ProjectXyzPipeline

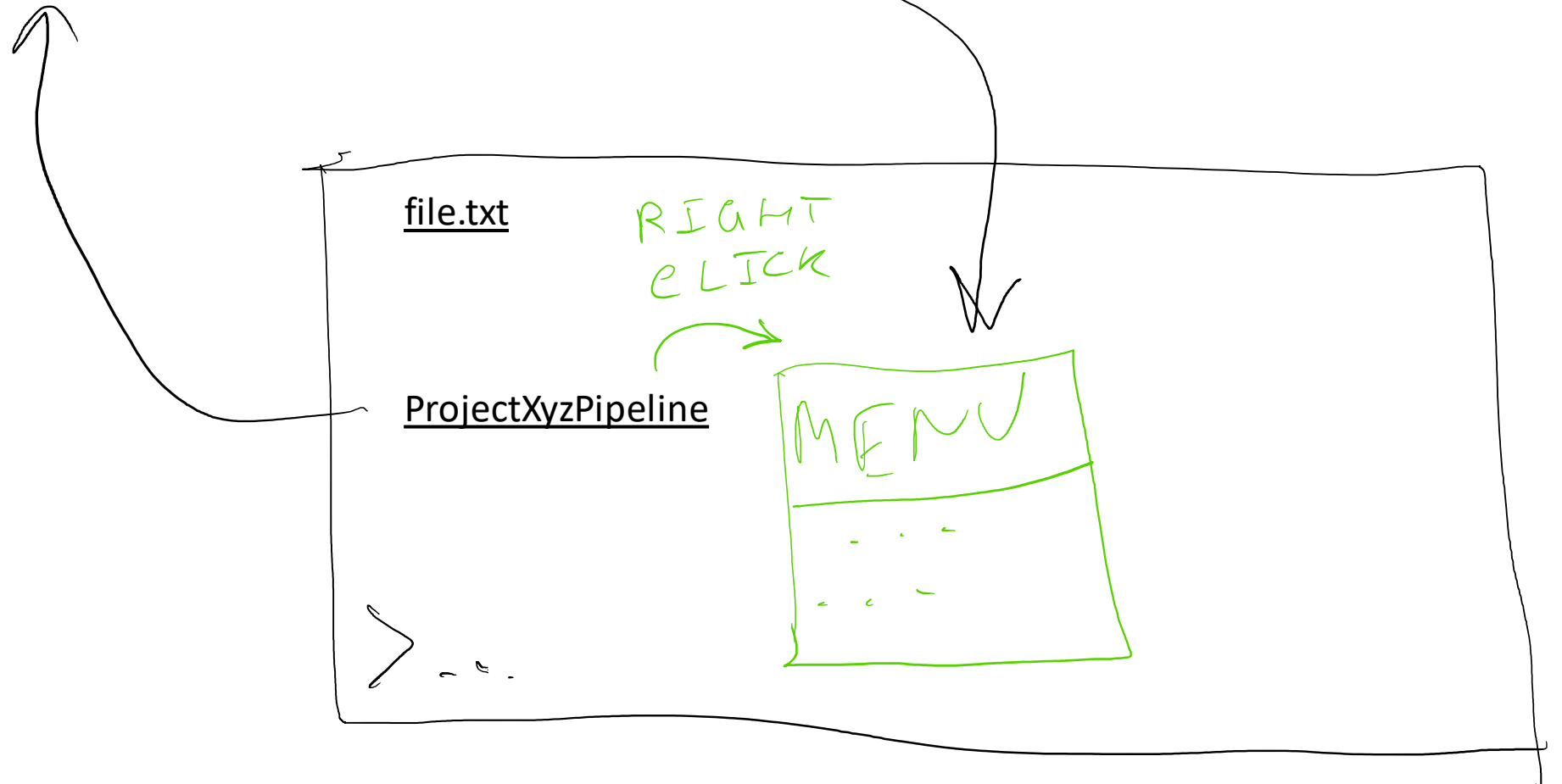
file.txt

ProjectXyzPipeline

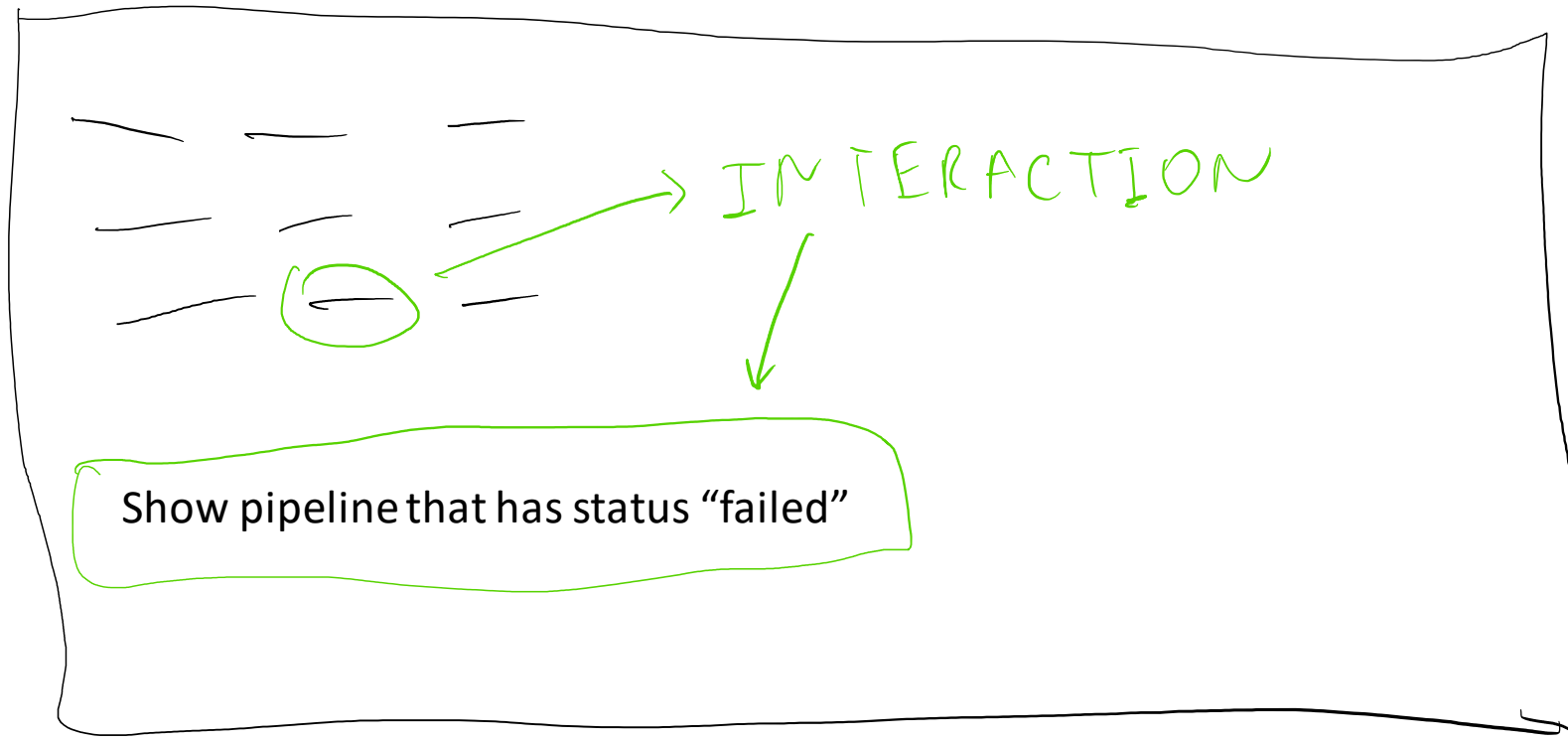
> ...

# User Interface – Plugins

- Plugin Blah: handles AWS::CodePipeline



# User Interface – Record / Replay

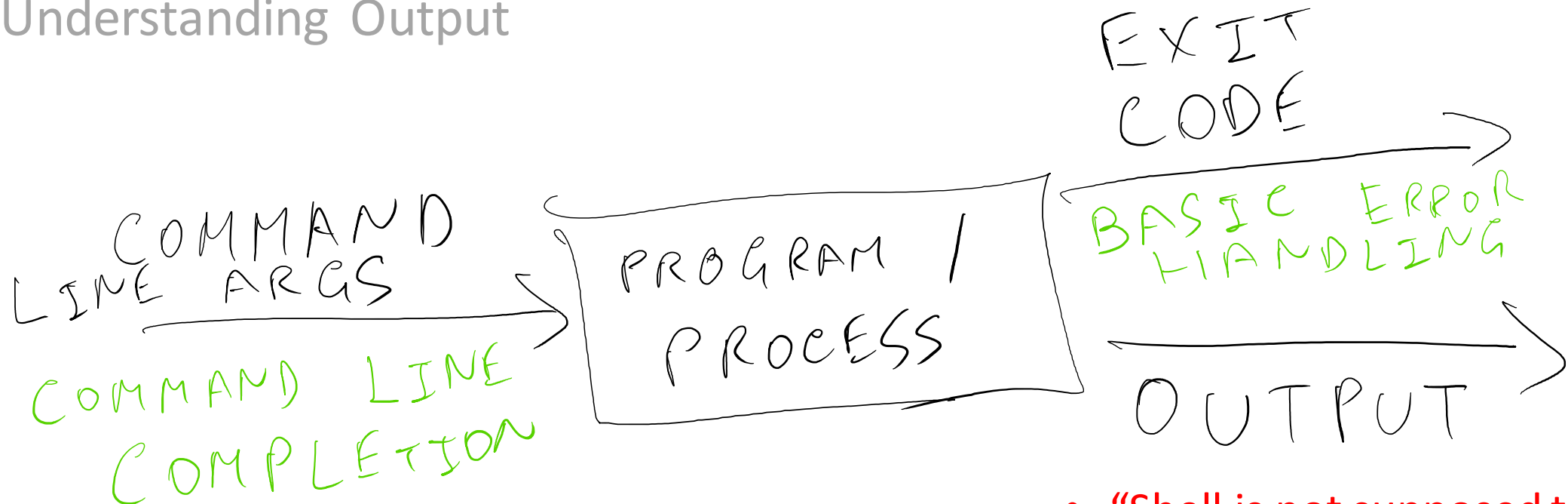


Capture the semantics  
of the interaction

More powerful than just  
history: i-123

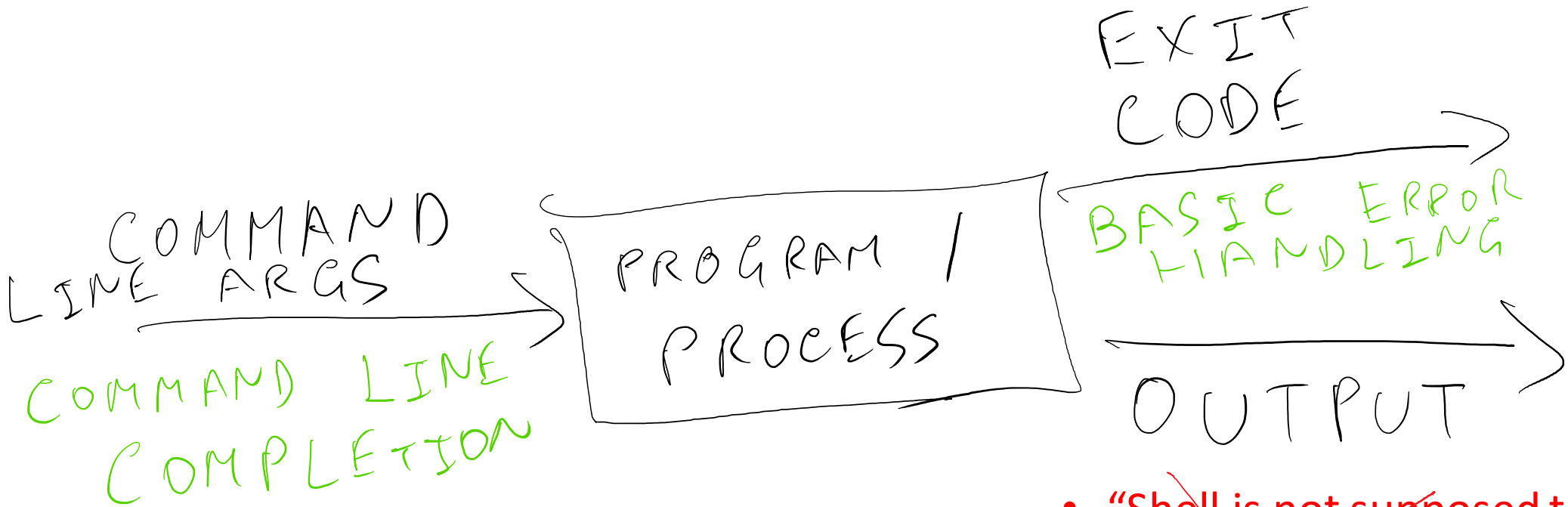
# User Interface – How?

## Understanding Output



- “Shell is not supposed to get into semantics”
- “It’s too much work”

# User Interface – How?



- Semantics – already implemented
- For many programs

- ~~“Shell is not supposed to get into semantics”~~
- ~~“It’s too much work”~~



# UI Vision Summary

- Semantics
  - The more a program "understands" the more powerful it can be.
  - Exit code -> command line arguments -> ???
- Capturing
  - Capture as much as you can
  - Capture the semantics

# Thanks!

- Next Generation Shell (WIP: the discussed UI)
  - <https://ngs-lang.org/>
  - <https://github.com/ngs-lang/ngs>
  - <https://github.com/ngs-lang/ngs/wiki/UI-Design>
- Ilya Sher
  - <https://ilya-sher.org/>

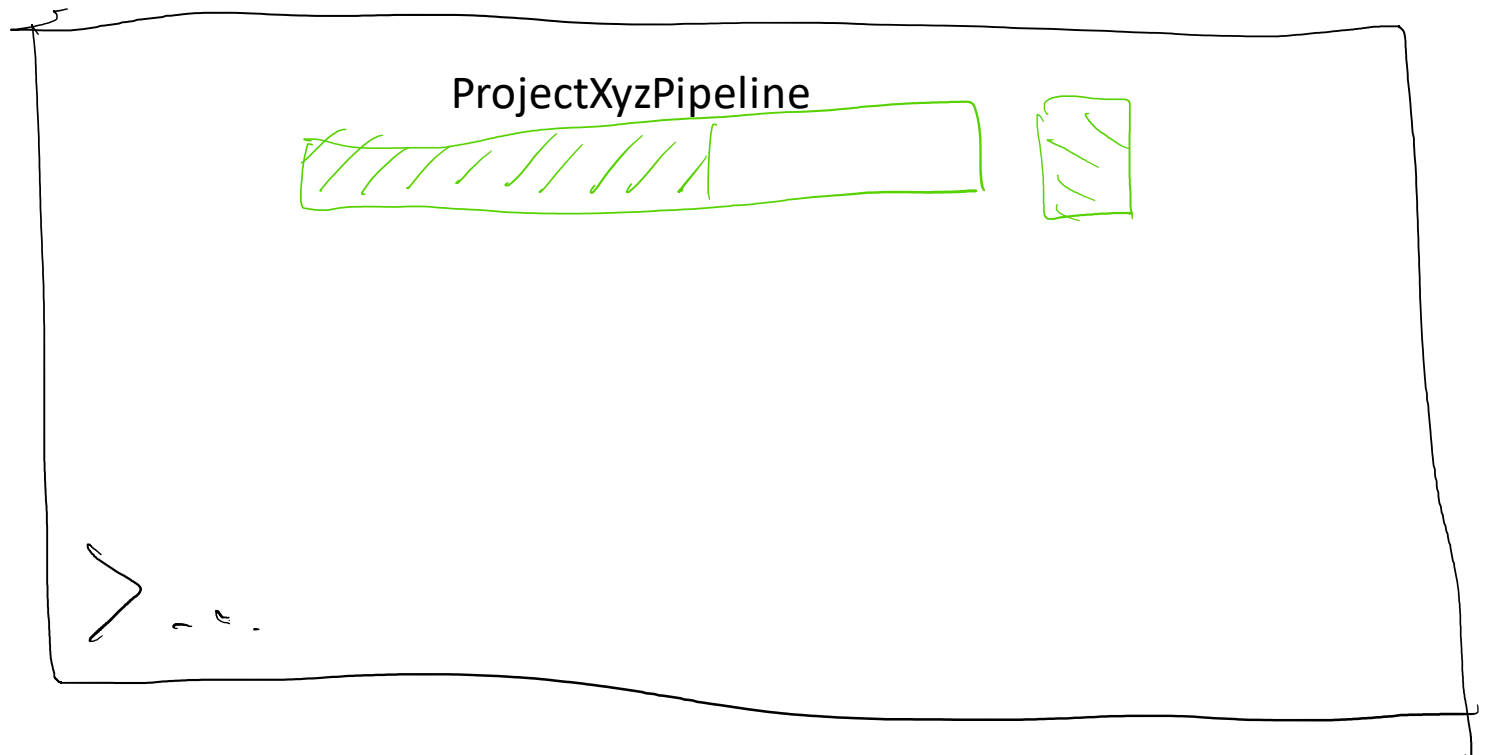
Bonus Slides

# User Interface – Objects of Interest

- Information displayed automatically – ex: CI/CD progress

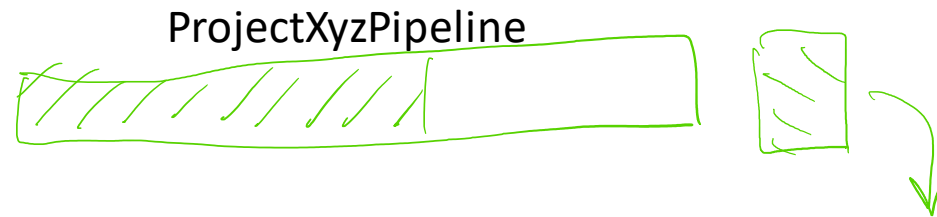
Would be very awkward if implemented in other shells

( like mail notification )



# User Interface – Objects of Interest

- Information displayed automatically – ex: CI/CD progress



Sensible default rules like:

- Show CI/CD runs of pipelines that I created
- Show CI/CD runs that were triggered by my commit

Same but for last errors instead of runs

*This information is shown to you because it matches rule "blah"*

- Edit rule
- Delete rule
- Exclude this object

# User Interface – Objects of Interest

- Information displayed automatically – last error

