No Code
Low Code
Go Code

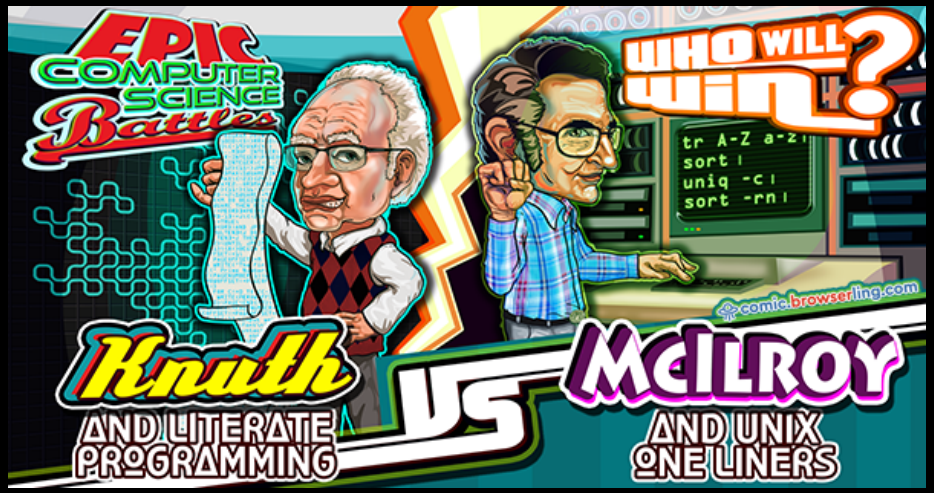
Embracing the code that isn't there

John Gosset | 2024-05-16

KISS

Knuth vs. McIlroy

- Programming Pearls: A Literate Program (1986)
- "Read a file of text, determine the n most frequently used words, and print out a sorted list of those words along with their frequencies."



McIlroy's Low Code Solution

```
tr -cs A-Za-z '\n' |
tr A-Z a-z |
sort |
uniq -c |
sort -rn |
sed ${1}q
```

"Knuth has shown us here how to program intelligibly, but not wisely. I buy the discipline. I do not buy the result. He has fashioned a sort of industrial-strength Fabergé egg—intricate, wonderfully worked, refined beyond all ordinary desires, a museum piece from the start."

Keep it Simple, Superhero

One of my most productive days was throwing away 1000 lines of code.

— Ken Thompson

The cheapest, fastest, and most reliable components are those that aren't there.

— Gordon Bell

Everything should be made as simple as possible, but not simpler.

— Albert Einstein

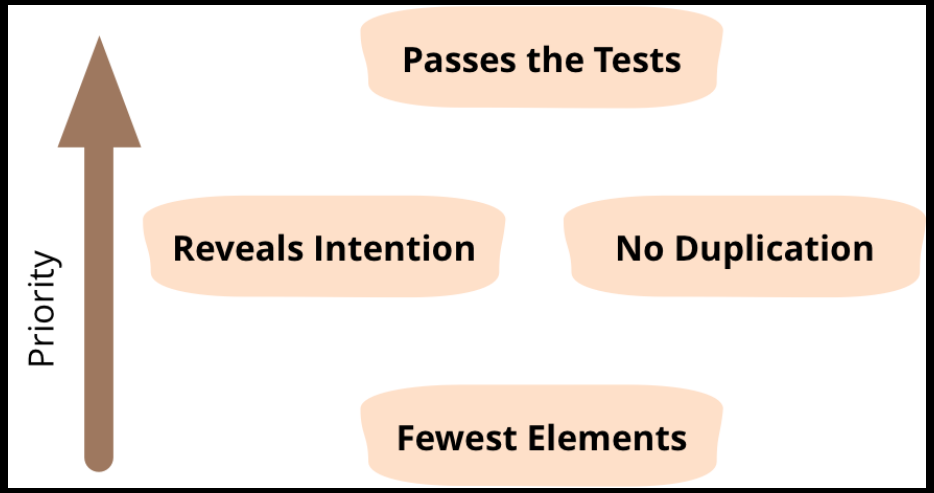
~~Simplify, Simplify,~~ Simplify

— Steve Jobs

Go was born to simplify

"Back around September 2007, I was doing some minor but central work on an enormous Google C++ program, one you've all interacted with, and my compilations were taking about 45 minutes on our huge distributed compile cluster. ... At this point I asked myself a question: Did the C++ committee really believe that was wrong with C++ was that it didn't have enough features? ... I started another compilation, turned my chair around to face Robert, and started asking pointed questions. Before the compilation was done, we'd roped Ken in and had decided to do something. We did not want to be writing in C++ forever..."

Kent Beck: 4 Rules of Simple Design



Low Code?

Definitions: IBM

Low Code

> “Low-code is a visual approach to software development that enables faster delivery of applications through minimal hand-coding.” (IBM)
>
> (Ref)

Definitions: AWS

Low Code

> "Low code or no code is a general phrase for development platforms and tools designed to help business users design and develop applications. Professional developers also use low code or no code tools for repetitive tasks that do not require conventional software development."
>
> (Ref)

Definitions: No One Owns "Low Code"!

- There's lots of marketing copy by software vendors, but not "one true definition" of low-code
- The original low code… the function
- Can declarative configuration in YAML be "low-code?"
- Core features:
  - Code reuse
  - Working at the right level of abstraction
  - Out of the box functionality
  - Visual modelling (but is this)
  - Empower "citizen developers"

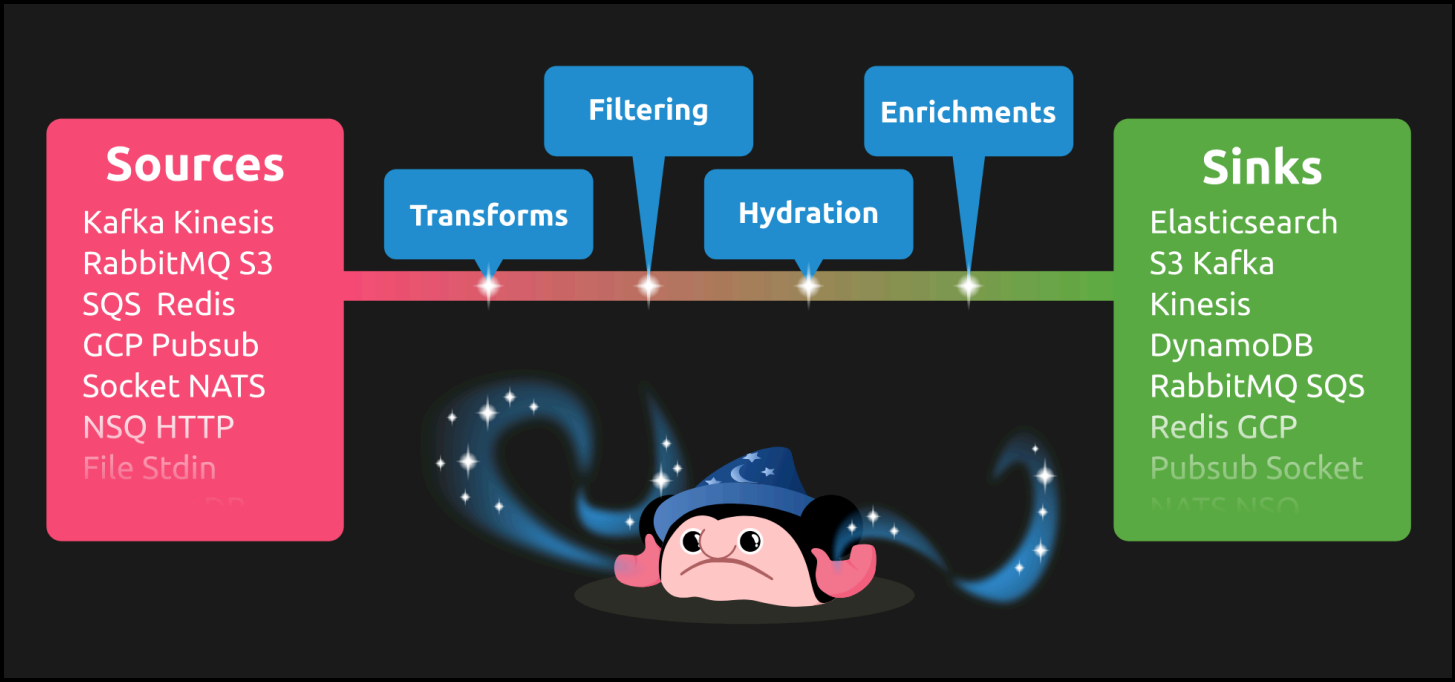
Low Code Tools

- SaaS offerings
  - AWS Workflow Studio
  - Microsoft Power Apps
- Open Source / Self-Hosted
  - Benthos (Go)
  - Telegraf (Go)
  - StackStorm
  - n8n.io
  - node-red
  - Vector

Case Study: Benthos

Benthos

https://www.benthos.dev



Benthos: Basics

```yaml
# config.yaml
input:
  generate:
    mapping: root = "Hello, Benthos!"
    count: 1

pipeline:
  processors:
    - mapping: content().uppercase()

output:
  stdout: {}
```

Benthos: HTTP

```yaml
# config.yaml
input:
  http_client:
    url: https://tradestie.com/api/v1/apps/reddit
    rate_limit: reddit

pipeline:
  processors:
    - unarchive:
        format: json_array

    - mapping: |
        root = this
        root.yolo = true
```

Benthos: Unit Tests

```yaml
# config.yaml
input:
  generate:
    mapping: root = "Hello, Benthos!"
    count: 1

pipeline:
  processors:
    - mapping: content().uppercase()

tests:
  - name: Content must be uppercase
    target_processors: /pipeline/processors/0
    input_batch:
      - content: hello
    output_batches:
      - - content_equals: HELLO
```

Benthos: Custom Plugins (in Go)

```go
func (r *reverseProcessor) Process(ctx context.Context, m *service.Message)
(service.MessageBatch, error) {
  bytesContent, err := m.AsBytes()
  if err != nil {
    return nil, err
  }

  newBytes := make([]byte, len(bytesContent))
  for i, b := range bytesContent {
    newBytes[len(newBytes)-i-1] = b
  }

  if bytes.Equal(newBytes, bytesContent) {
    r.logger.Infof("Woah! This is like totally a palindrome: %s", bytesContent)
    r.countPalindromes.Incr(1)
  }

  m.SetBytes(newBytes)
  return []*service.Message{m}, nil
}
```

Benthos: Studio

In Closing

Takeaways

- Keep it simple
  - Or, as Rob Pike put it: “Less is Exponentially More”
- Prioritize write the interesting code that matters
- Consider low code for simple bridging, enrichment, filtering
- As a developer, you want low code that you can extend

Further Reading

- Knuth & McIlroy
  - Programming Pearls: A Literate Program
  - More Shell, Less Egg
- Rob Pike: Less is Exponentially More
- Kent Beck: 4 Rules for Simple Design
- Benthos
  - Benthos
  - Benthos Plugin example repo (Go)
  - Benthos Studio
  - Stream Processing the Easy Way (with NATS and Benthos)

Thank You!

John Gosset

LinkedIn / GitHub

Slides + Code