The Svelte Buzz

🔥

Kanishk Agrawal

Agenda

- What?
- Why?
- Live coding in Svelte
- Shortcomings

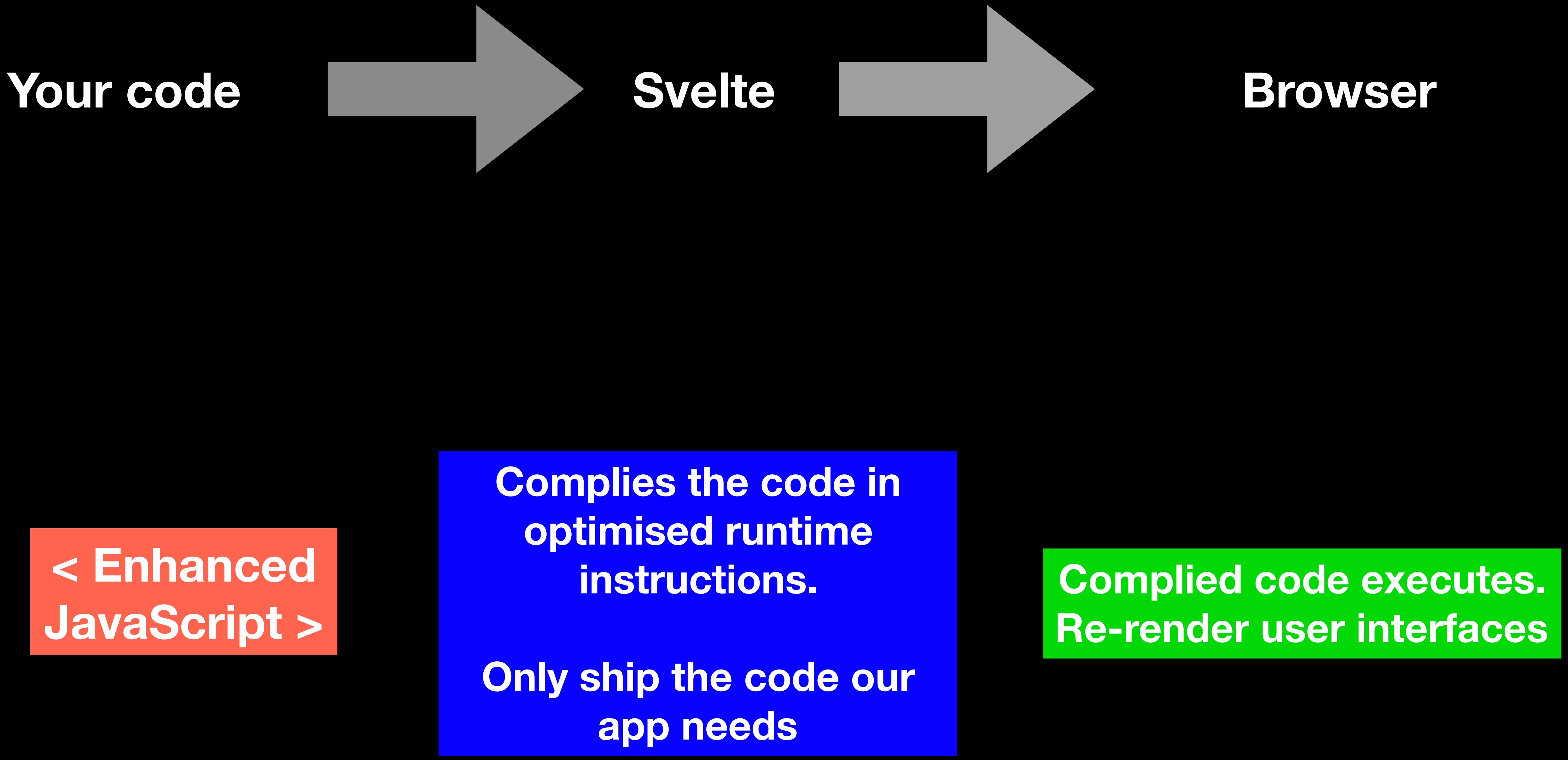
What is Svelte?

- Compiler
- Much like JavaScript
- Truly Reactive
- Write less. Do more
- Inbuilt state management

So what if it’s a compiler?

React / Vue do bulk of their work in browser, Svelte shifts that work into a compile step that happens when you build your app.

No extra code to be shipped



Your code

Svelte

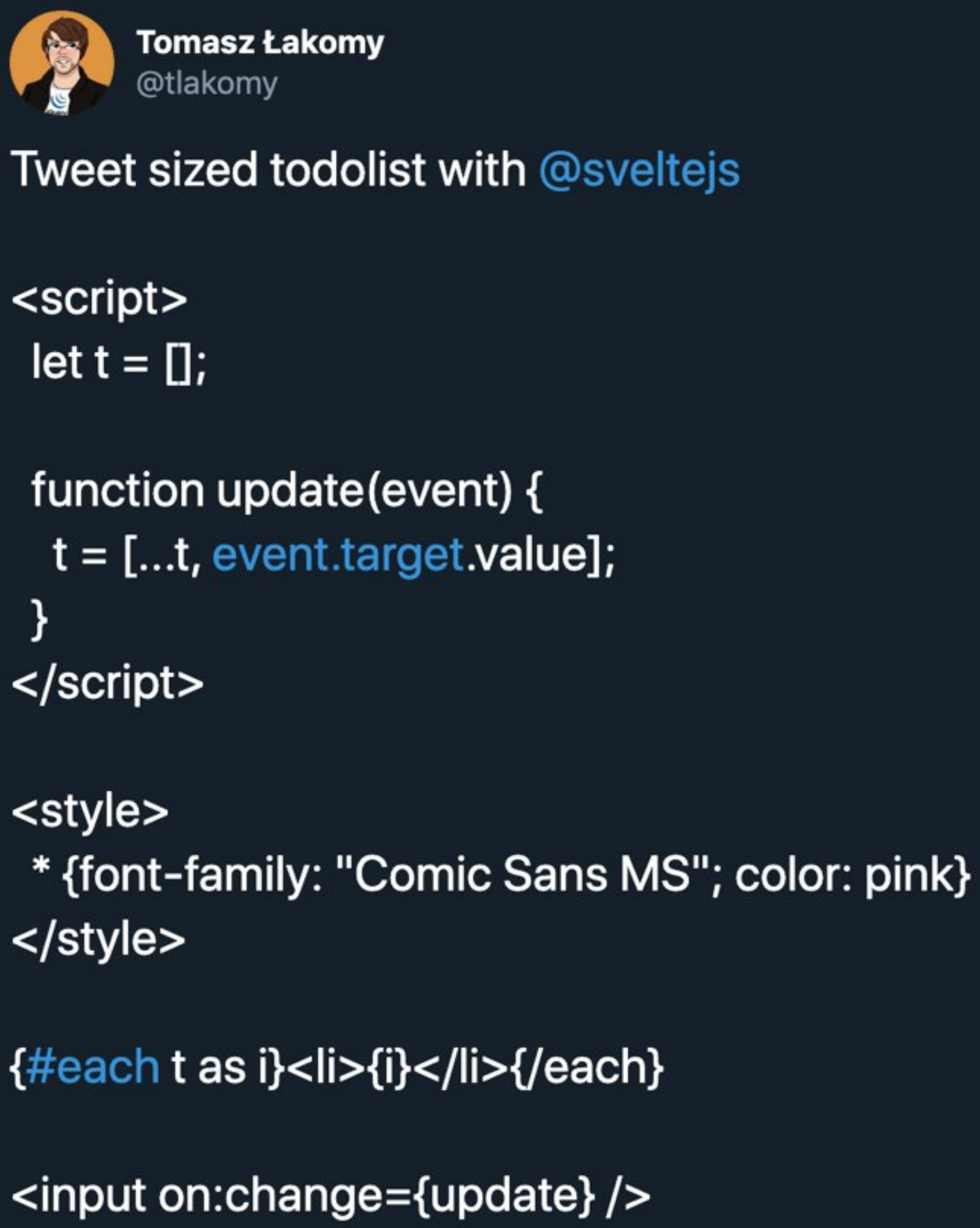
Browser

< Enhanced JavaScript >

Complies the code in optimised runtime instructions.

Only ship the code our app needs

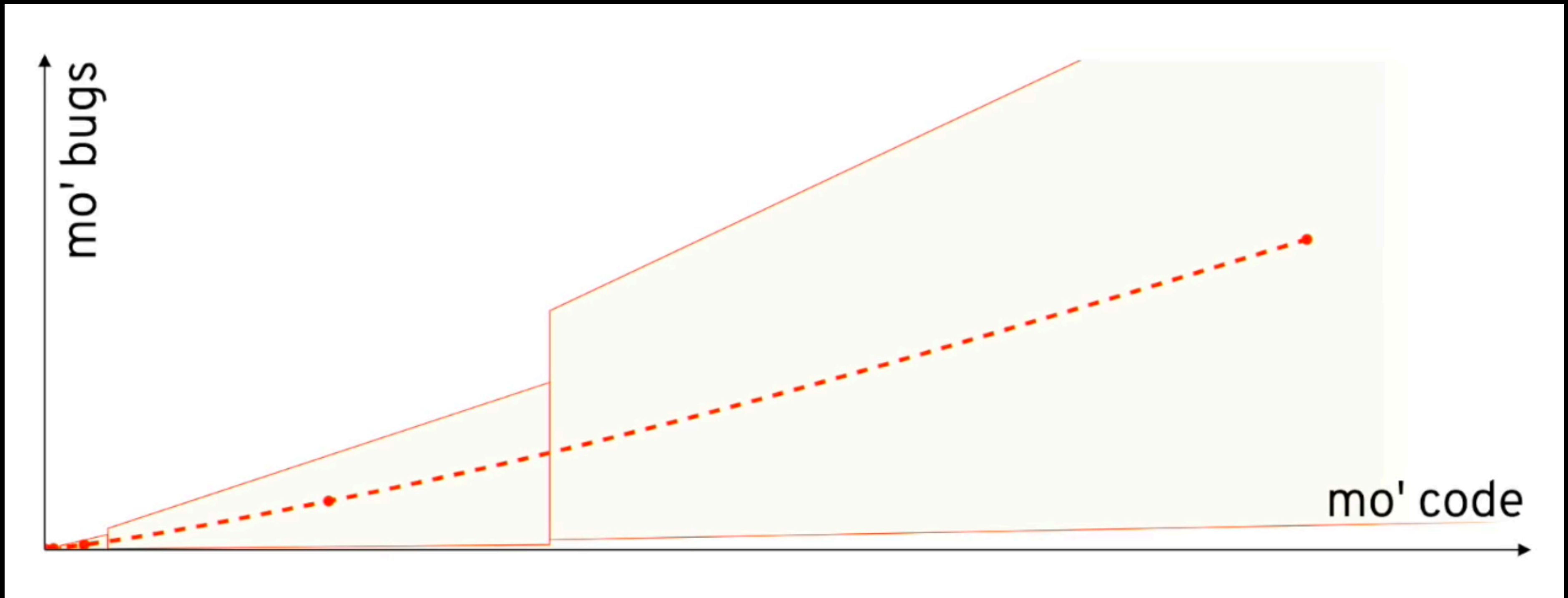
Complied code executes.
Re-render user interfaces



Tomasz Łakomy
@tlakomy

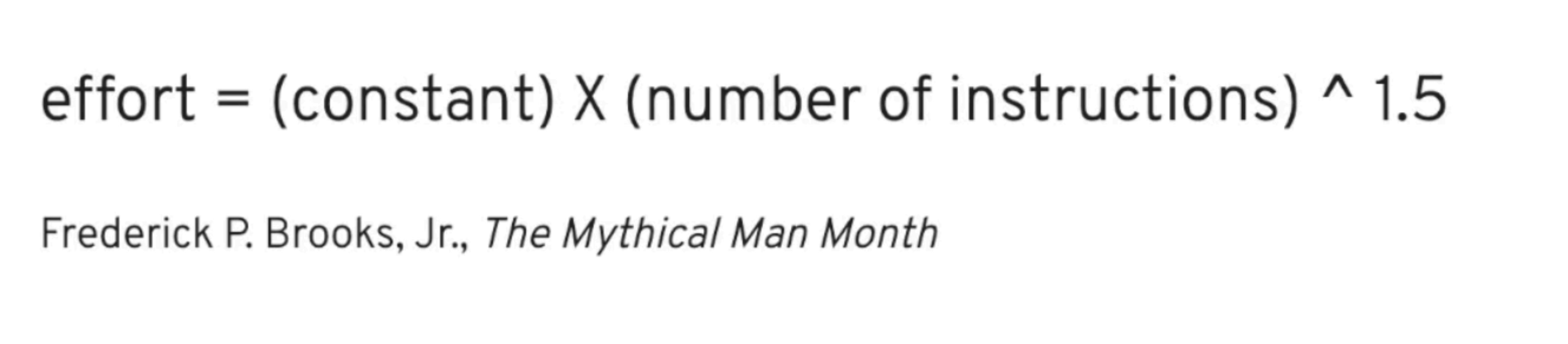
Tweet sized todolist with @sveltejs

```
<script>
 let t = [];

 function update(event) {
  t = [...t, event.target.value];
 }
</script>

<style>
 * {font-family: "Comic Sans MS"; color: pink}
</style>

{#each t as i}<li>{i}</li>{/each}

<input on:change={update} />
```



Source: Code Complete 2 by Steve McConnell

LOC v/s Bugs is quadratic relationship



effort = (constant) X (number of instructions) ^ 1.5

Frederick P. Brooks, Jr., *The Mythical Man Month*

More LOC, more is the effort

Svelte

```
<script>
  let a = 1;
  let b = 2;
</script>

<input type="number" bind:value={a}>
<input type="number" bind:value={b}>

<p>{a} + {b} = {a + b}</p>
```

React

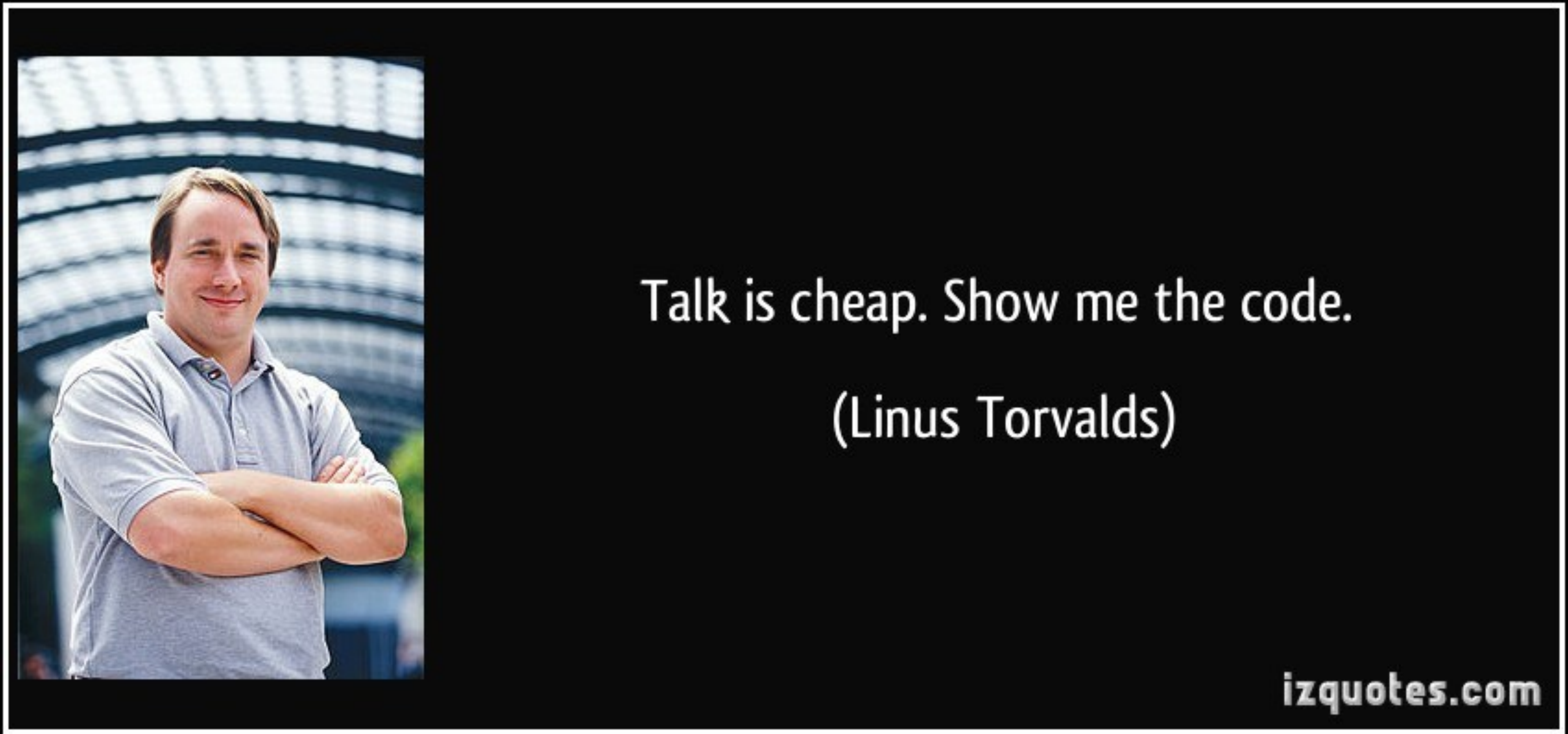
```
import React, { useState } from 'react';

export default () => {
  const [a, setA] = useState(1);
  const [b, setB] = useState(2);

  function handleChangeA(event) {
    setA(+event.target.value);
  }

  function handleChangeB(event) {
    setB(+event.target.value);
  }

  return (
    <div>
      <input type="number" value={a} onChange={handleChangeA}/>
      <input type="number" value={b} onChange={handleChangeB}/>

      <p>{a} + {b} = {a + b}</p>
    </div>
  );
};
```

You write less code

- Means less bugs
- Easier to onboard & understand
- Speedy development
- ~ 30-40% less code than React

Mutations are back!

```
// C++

void counter( ) {

  int count = 0;

  if ( whatever ) {
      count++
  } else if ( whatever2 ) {
      count- -
  } else if ( reset ) {
      count = 0;
  }

}
```



Talk is cheap. Show me the code.

(Linus Torvalds)

React	Svelte
Immutable	Mutable
Functional programming	Procedural / OOP / anything
Library	Compiler (Framework)
Redux	Svelte Store (built-in)
Virtual DOM	No Virtual DOM
CSSTransition / React Spring	Svelte Animation (built-in)
Uses JavaScript	Superset of JavaScript
No templating	Templating (with HTML)
styled-component	Scoped CSS
React Router	Sapper Router
Next.js	Sapper

Things I love about Svelte

- **Scoped CSS**
- **Truly reactive**
- **Enabling Accessibility**
- **Animations & Transitions**
- **Built in State management**

👨‍💻

KanishkAgrawal