

# FLOWING<sup>®</sup> extrategy

---

**Un Approccio Frameworkless  
per Sviluppare la tua Single  
Page Application**



# Antonio Dell'Ava

**frontend developer**

background in interaction design

loves cycling



# OUR PEOPLE

ideato e extrategy ora sono Flowing: una community inclusiva di persone che condividono passione per l'innovazione e bisogno costante di evoluzione. Abbiamo cocreato insieme questa nuova realtà, e insieme a chi condivide i nostri valori continuiamo a farla crescere giorno per giorno.



—



# Frameworkless movement

<https://github.com/frameworkless-movement/>

—

Peace & Love...  
non odiamo i  
framework

—

**Prendere decisioni  
tecniche  $\neq$  Usare il  
framework più di  
moda**



A diagram showing the components of a frontend. A large, light green oval is centered on the page. Inside this oval, the word "Frontend" is at the top. Below it, the words "Interfacce" and "Application" are positioned on the left and right sides respectively. The entire diagram is set against a white background with a light green gradient.

**Frontend**

**Interfacce**

**Application**

**i regret this decision**





**UX**



**Business  
Needs**



**Non Functional  
Requirements**



**Sviluppo del  
Progetto**

# Non Functional Requirements

quanto durerà il software?

in quanto tempo deve essere appreso?

qual è il “ritmo” dell’interazione?

da chi deve essere utilizzato? che competenze ha?

---

# Come legarli al progetto?

## Design System

FLOWING<sup>®</sup>  
extrategy

---

# Errate aspettative

“Per fare il design system devo avere una pattern library”

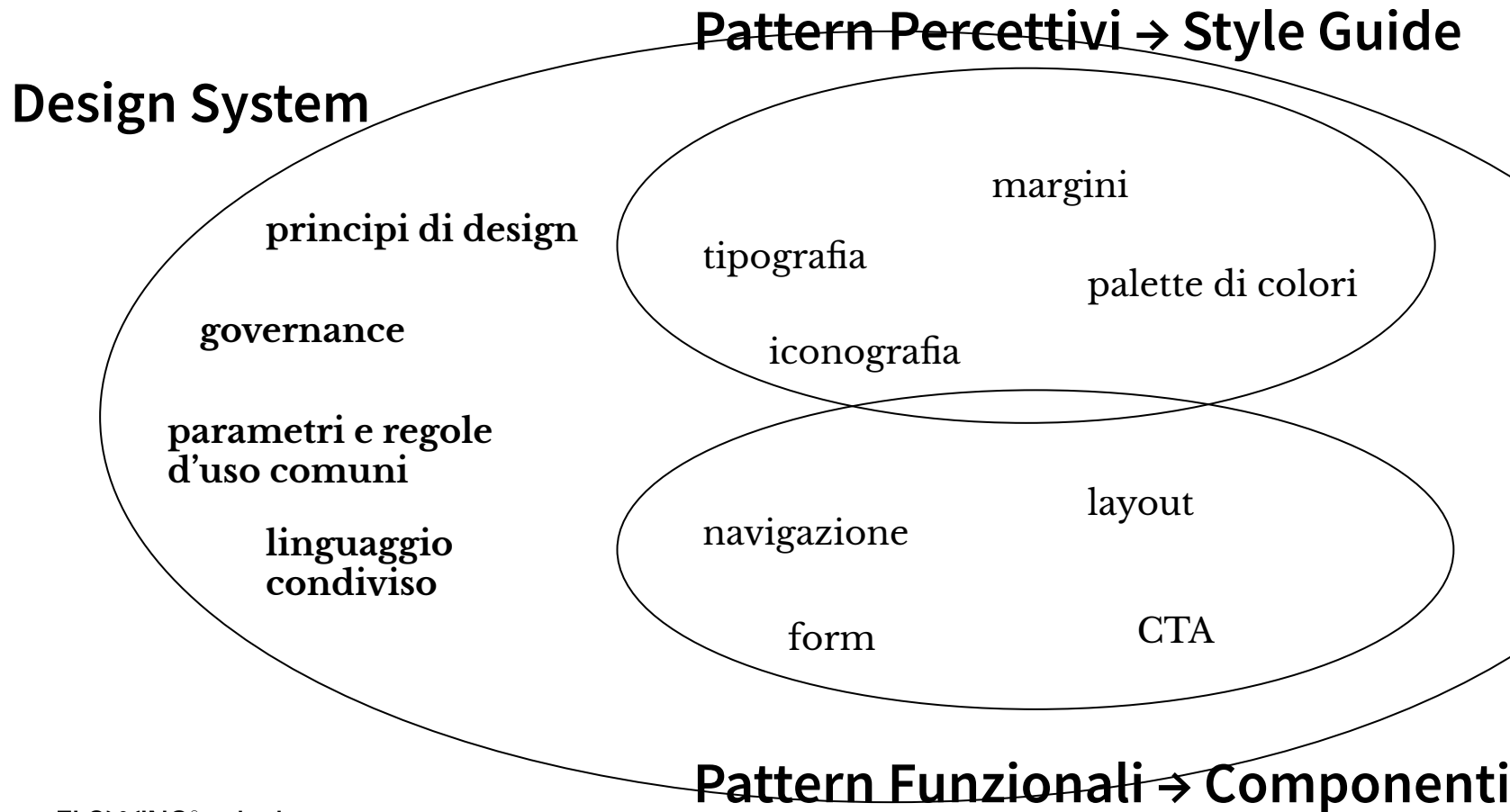
“Il design system è UX”

---


# The true story

“ho una letteratura di interfacce che rispecchia i n.f.r., abbiamo lavorato per avere un linguaggio comune, quindi un vocabolario dei componenti mi sarà utile”





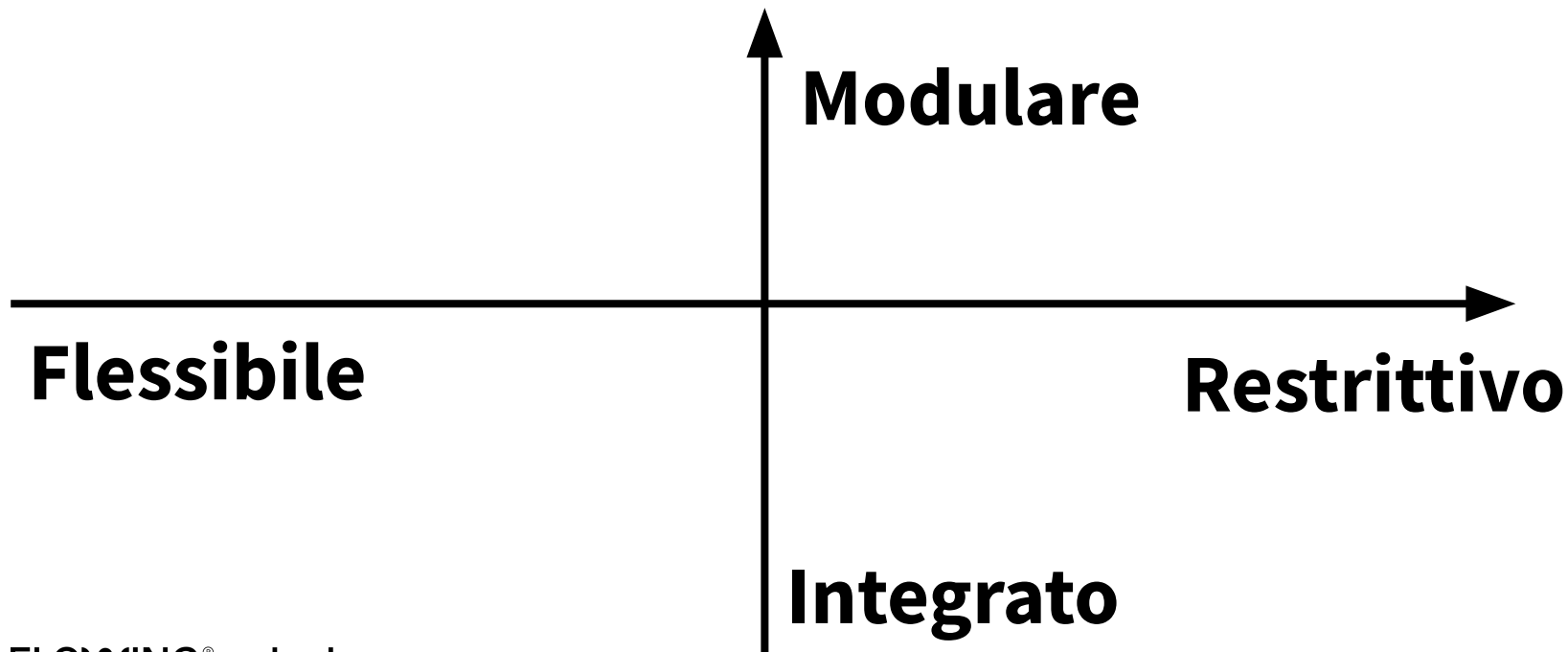
# Processo

1. Principi
  2. Parametri e regole d'uso comuni
  3. Governance
  4. Pattern
  5. **Pattern Library**
- 

# Design System: Principi

- 3..5 frasi
- tutto il team coinvolto
- Esempio errato: **“completezza e piacevolezza”**
- Esempio corretto: **“Seguire le seguenti priorità: prima completezza delle informazioni di contesto poi piacevolezza estetica”**
- la “prova del 9” è chiedersi “il contrario del mio principio potrebbe essere valido per un altro design system?”

# Design System: Parametri e regole d'uso comuni



# Design System: Governance

- Quale è il team? Livello di delega?
- Quali sono gli strumenti?
- Come si rendono visibili i risultati? Canali?
- Quali sono i riti?



# Design System: Design Pattern

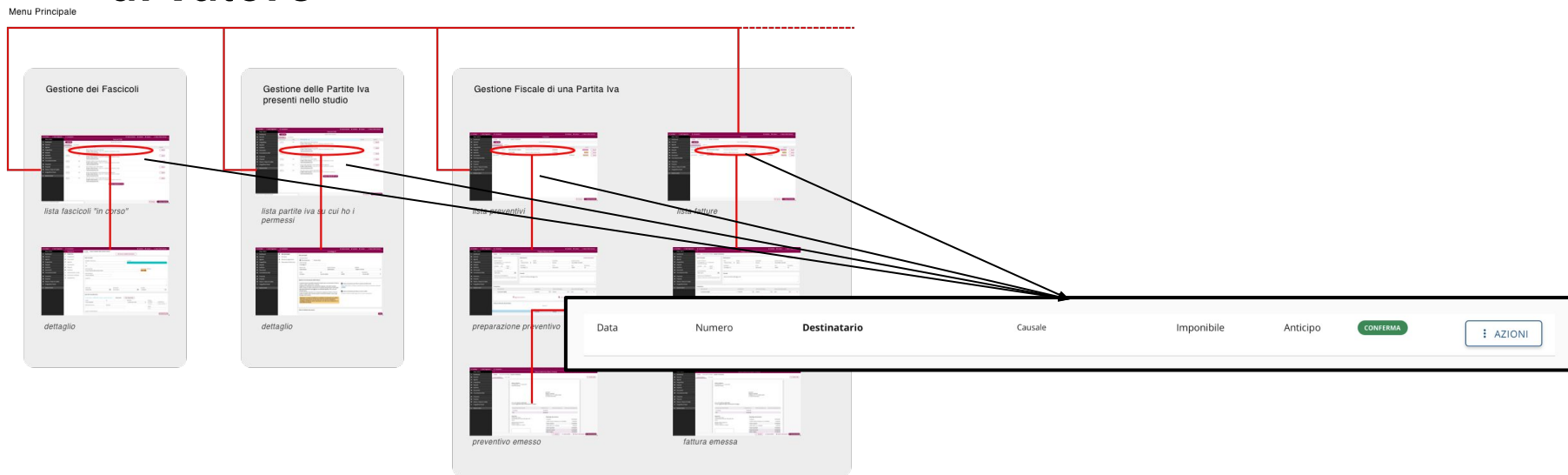
→ [def] una soluzione progettuale  
riusabile per un problema ricorrente

→ Come? Processo induttivo da

**UI+User journey a Design Pattern**

# Interface inventory

→ fa emergere quanto un pattern è centrale nella consegna di valore



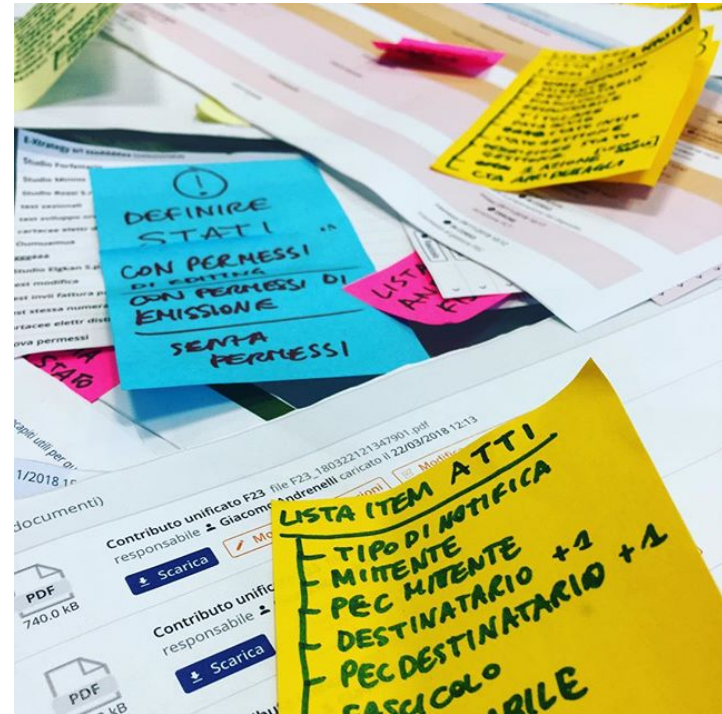
# 1. Visualizzare il flusso

- stampare un flusso di interfaccia
- taggare i componenti in funzione dello scopo che hanno



## 2. Definire il Pattern e la Struttura

si ritagliano e si  
raggruppano elementi  
simili definendo la  
specificità e i contenuti



### 3. Battezzare e documentare il pattern e varianti

#### Lista documenti contabili

- Data
- Numero
- Destinatario
- Causale
- Imponibile
- Anticipo (solo per preventivo)
- Stato
- Azioni
  - ◆ Riporta in bozze (solo per preventivo)
  - ◆ Accetta preventivo (solo per preventivo)
  - ◆ Incasso (solo fattura)
  - ◆ [...]

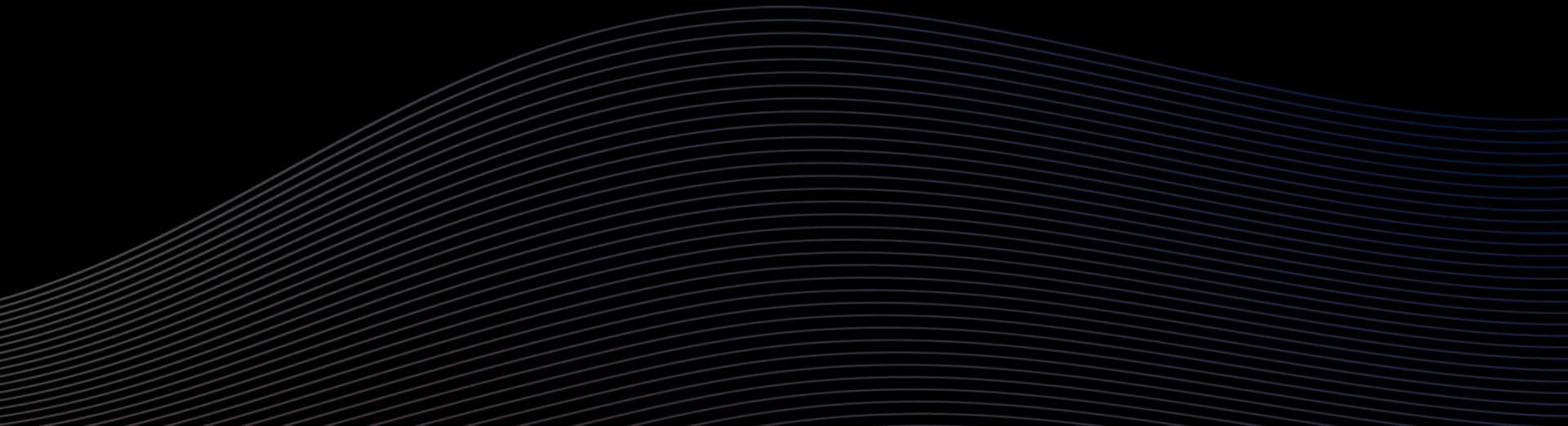
#### Lista documenti contabili: variante preventivo

Data	Numero	Destinatario	Causale	<u>Imponibile</u>	<u>anticipo</u>	Stato	Azioni
							<div>Riporta in bozze Accetta Rifiuta Fattura</div>



—

# Pattern Library?

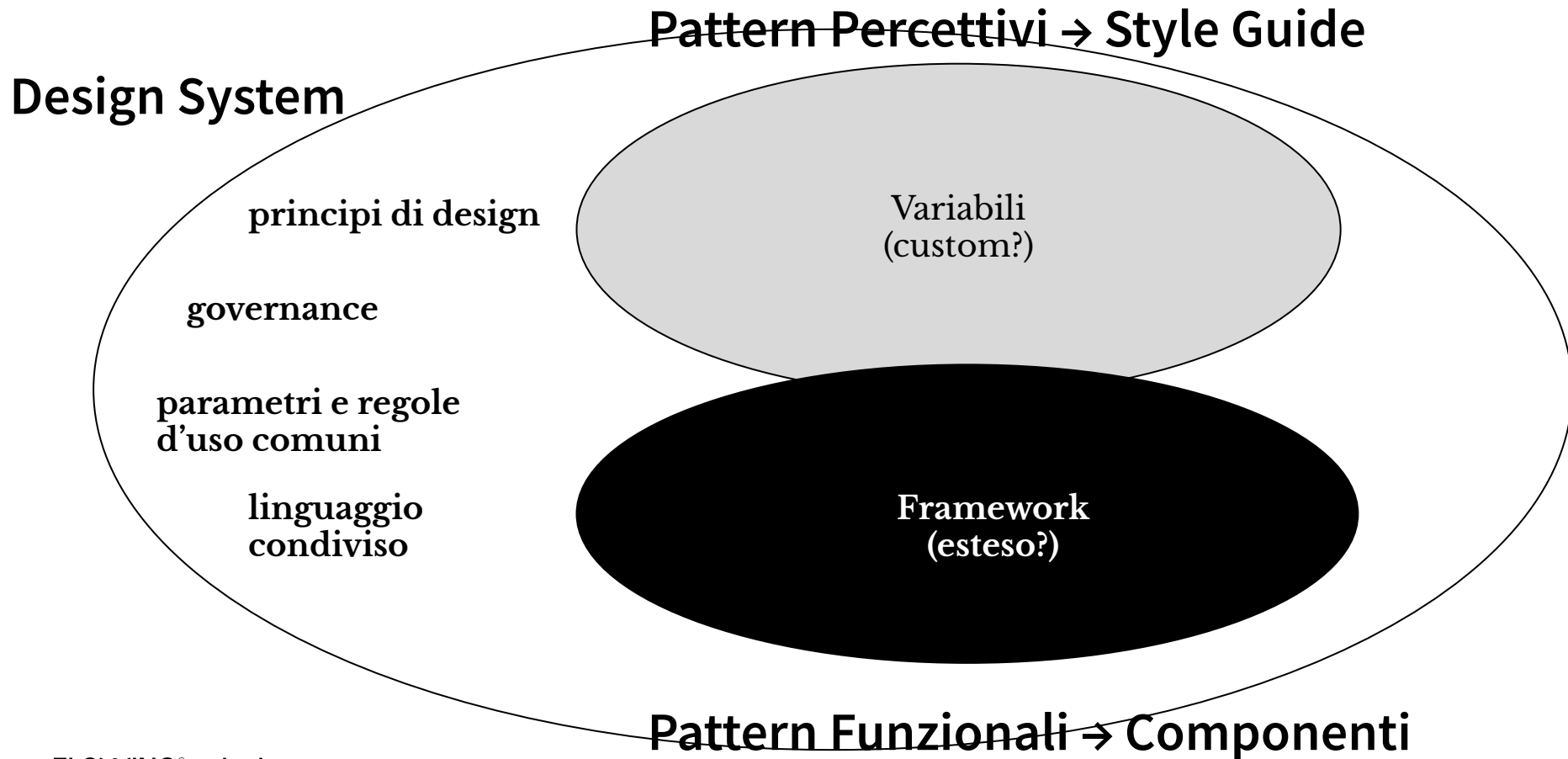


—

# Framework

==

una pattern library  
pronta



# Alcuni Esempi

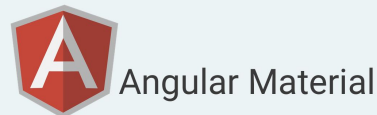
## CSS only



## Integrabili

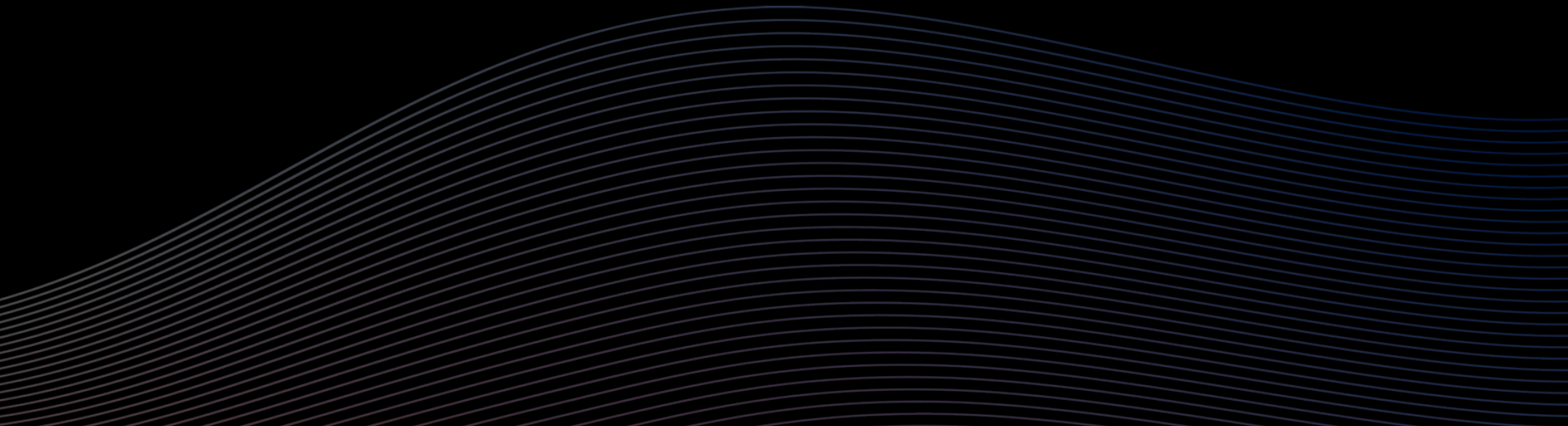


## Integrati



—

**È per forza un male  
usarli?**



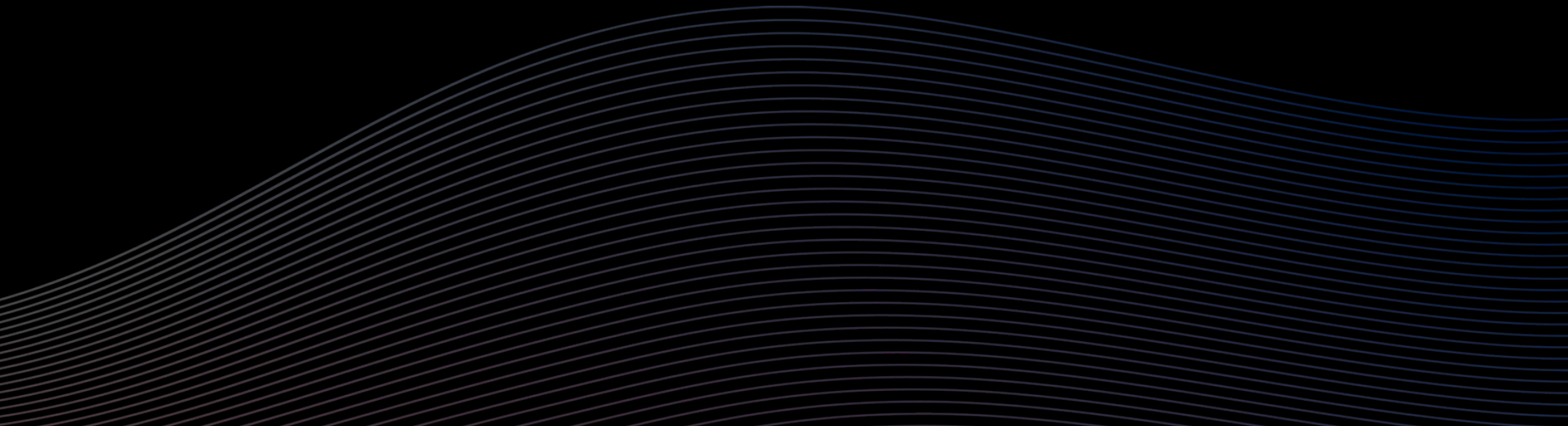
---

# Pitfalls

**FLOWING<sup>®</sup>**  
extrategy

—

# 1. Tradeoff



Framework UI

**PRO**

Velocity

**CONTRO**

Specificità

Debito Tecnico

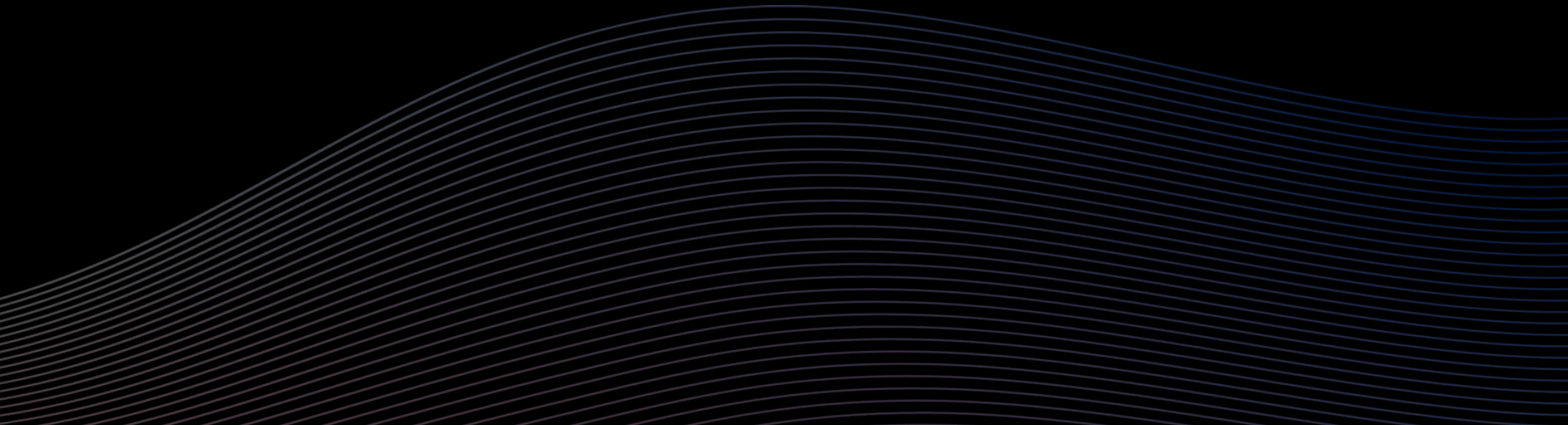
FLOWING® extrategy



	Avventato	Prudente
Scelta	<i>“Non abbiamo tempo per progettare”</i>	<i>“Preferiamo accelerare e siamo consapevoli delle conseguenze”</i>
Involontario	<i>“Cosa è la UX?”</i>	<i>“Ora sappiamo come avremmo dovuto farlo”</i>

—

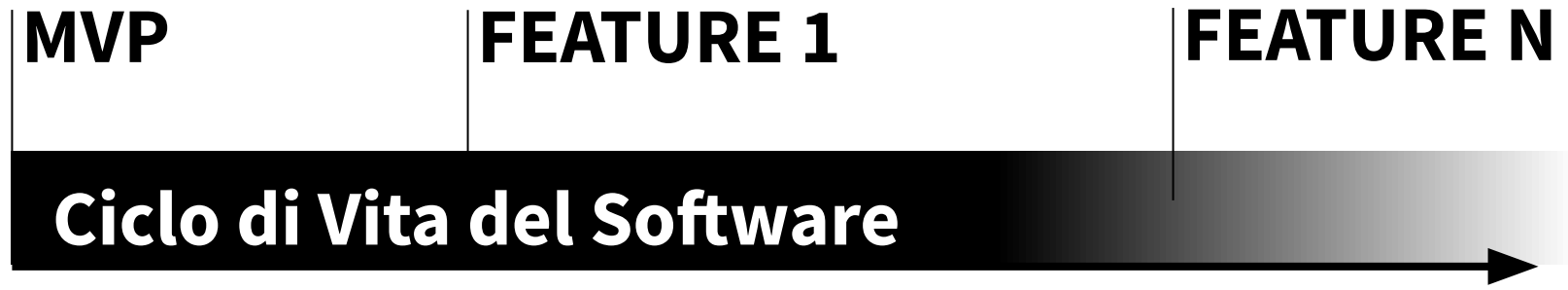
## 2. Il Software soddisfa i bisogni nel tempo



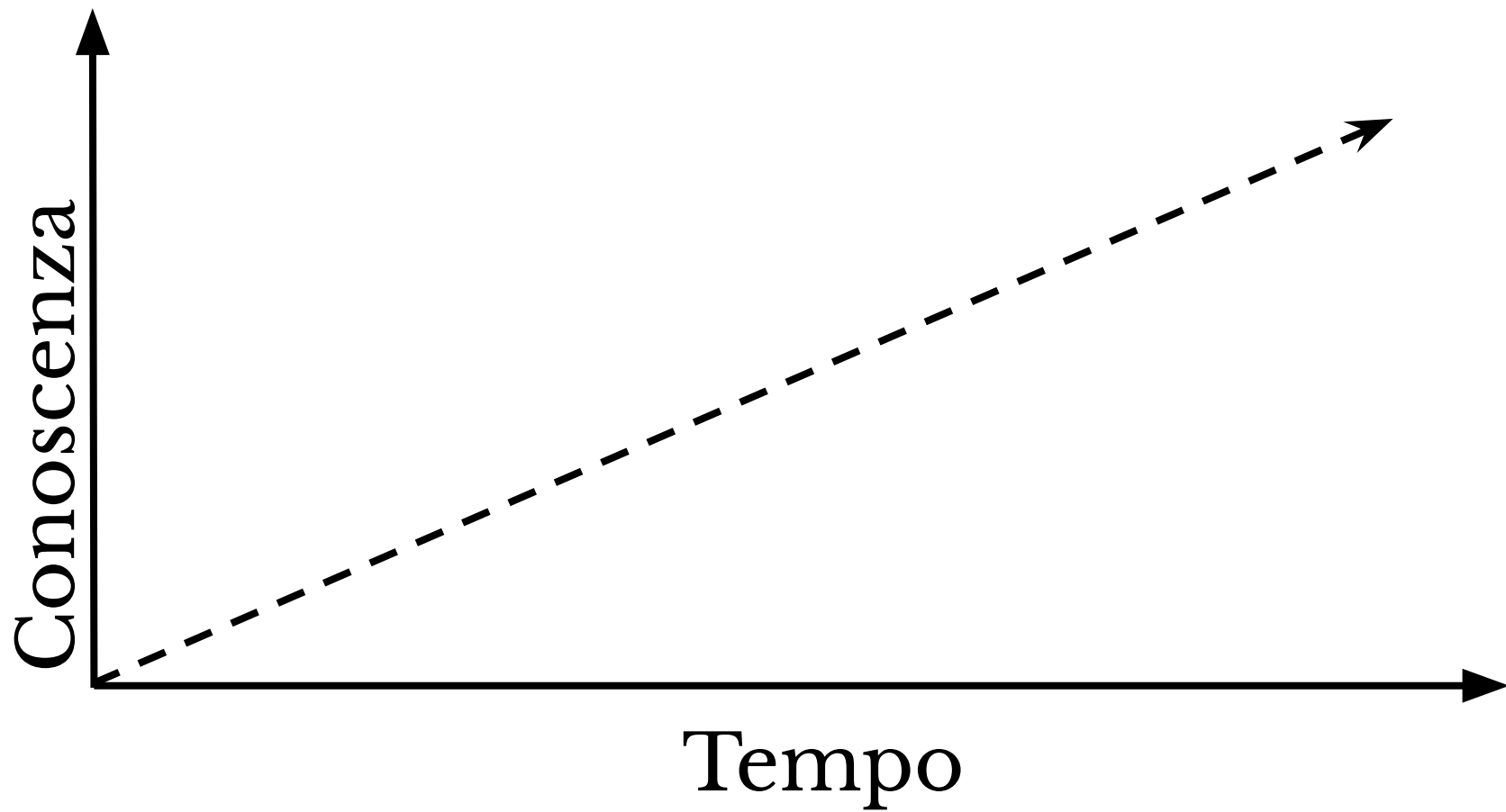
**Ciclo di Vita del Software**

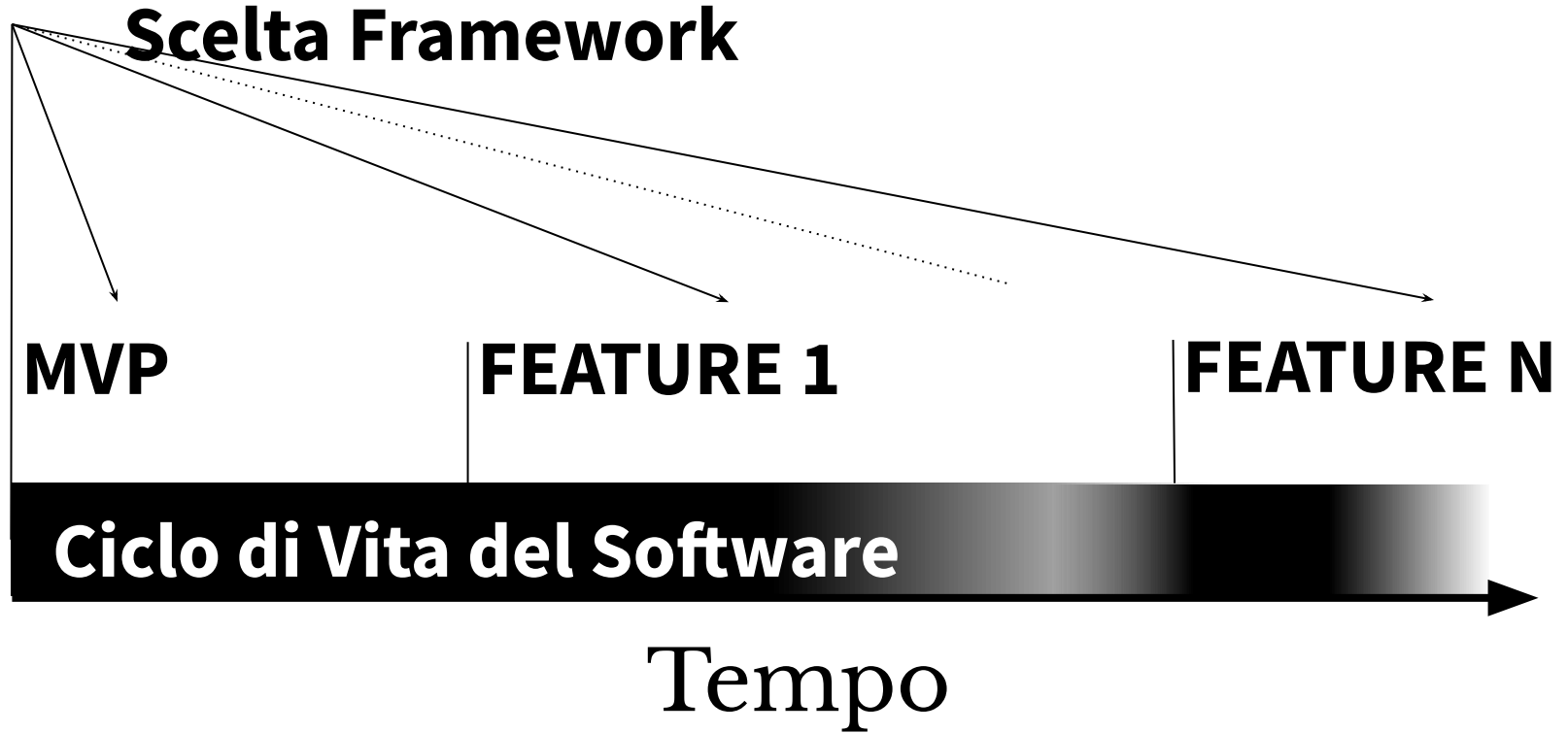


Tempo



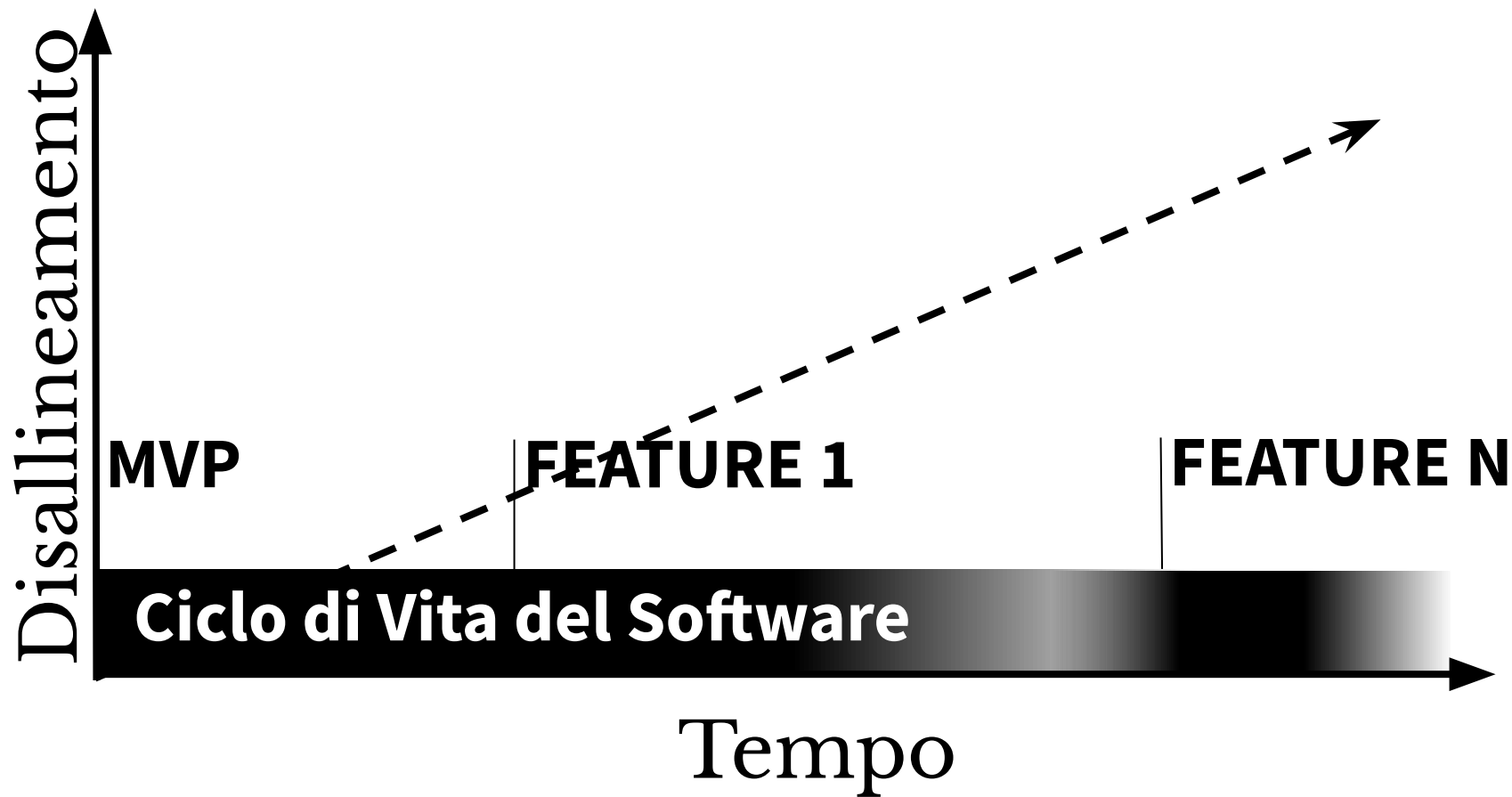
Tempo





—

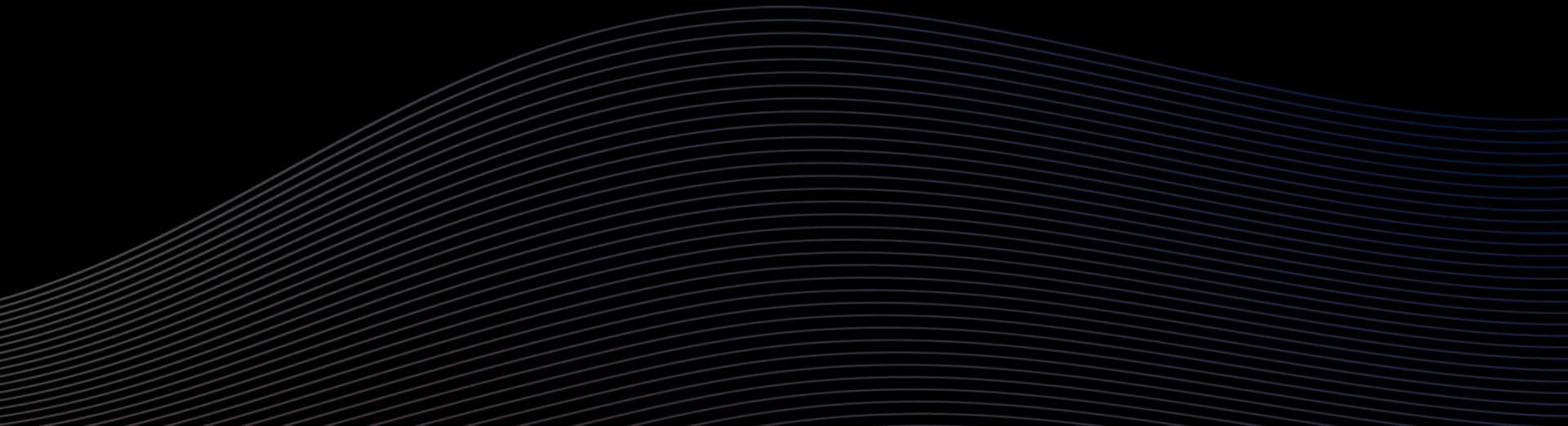
## 4. Il framework non toglie il costo di allineamento





—

# Alternative



# Possibili approcci

1. “Classico”: framework frontend  
react/angular/vue + SCSS
2. **WebComponents**

# Esempio di Web Component

1  €

2  €

3  €

4  €

5  €

6  €

7  €

# Esempio Web Component

```
<html>
  [...]
  <body>

    <mydesignsystem-input-currency
      name="valore-bonifico" num="889271"
      label="valore
      bonifico"></mydesignsystem-input-curr
      ency>
    </body>
  </html>
```

```
import InputCurrency from
  "../InputCurrency/InputCurrency";

window.customElements.define(
  "mydesignsystem-input-currency",
  InputCurrency
);
```

# Esempio Web Component

```
class InputCurrency extends HTMLElement {  
  
    static get observedAttributes () { [ // attributi osservabili ] }  
  
    connectedCallback () {  
  
        // render  
  
        // bind events  
  
    }  
  
    attributeChangedCallback (name) { //... sync my template }  
  
}
```

# Esempio Web Component

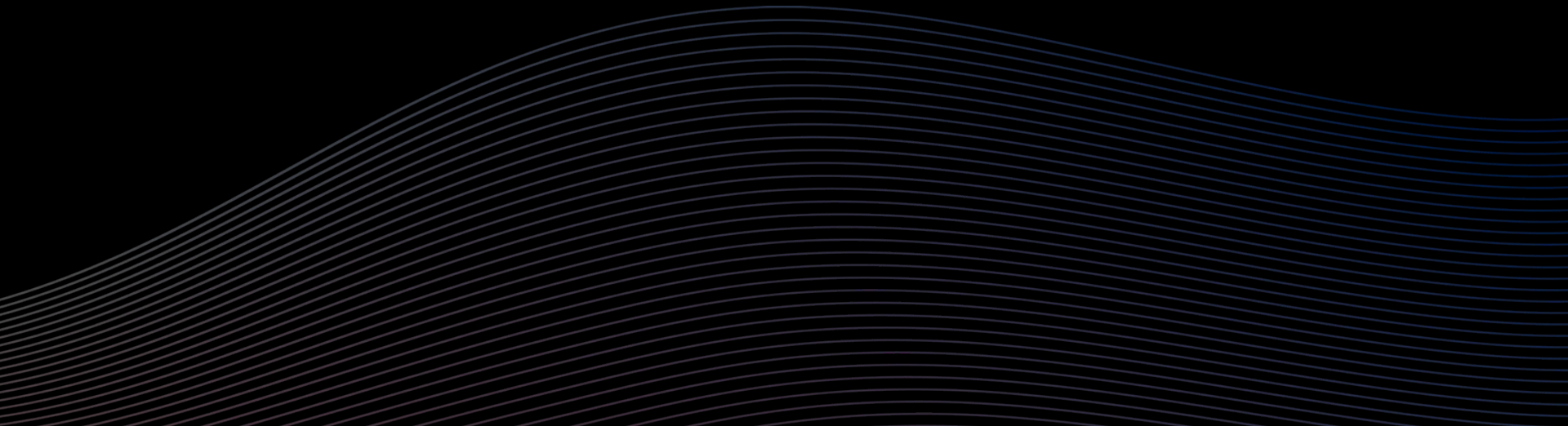
```
const emitChangeValue = (element) => {  
  const event = new window.CustomEvent("value-change", {  
    detail: () => Number.parseFloat(element.value),  
    bubbles: true  
  });  
  
  element.dispatchEvent(event);  
};
```

# **Caratteristiche**

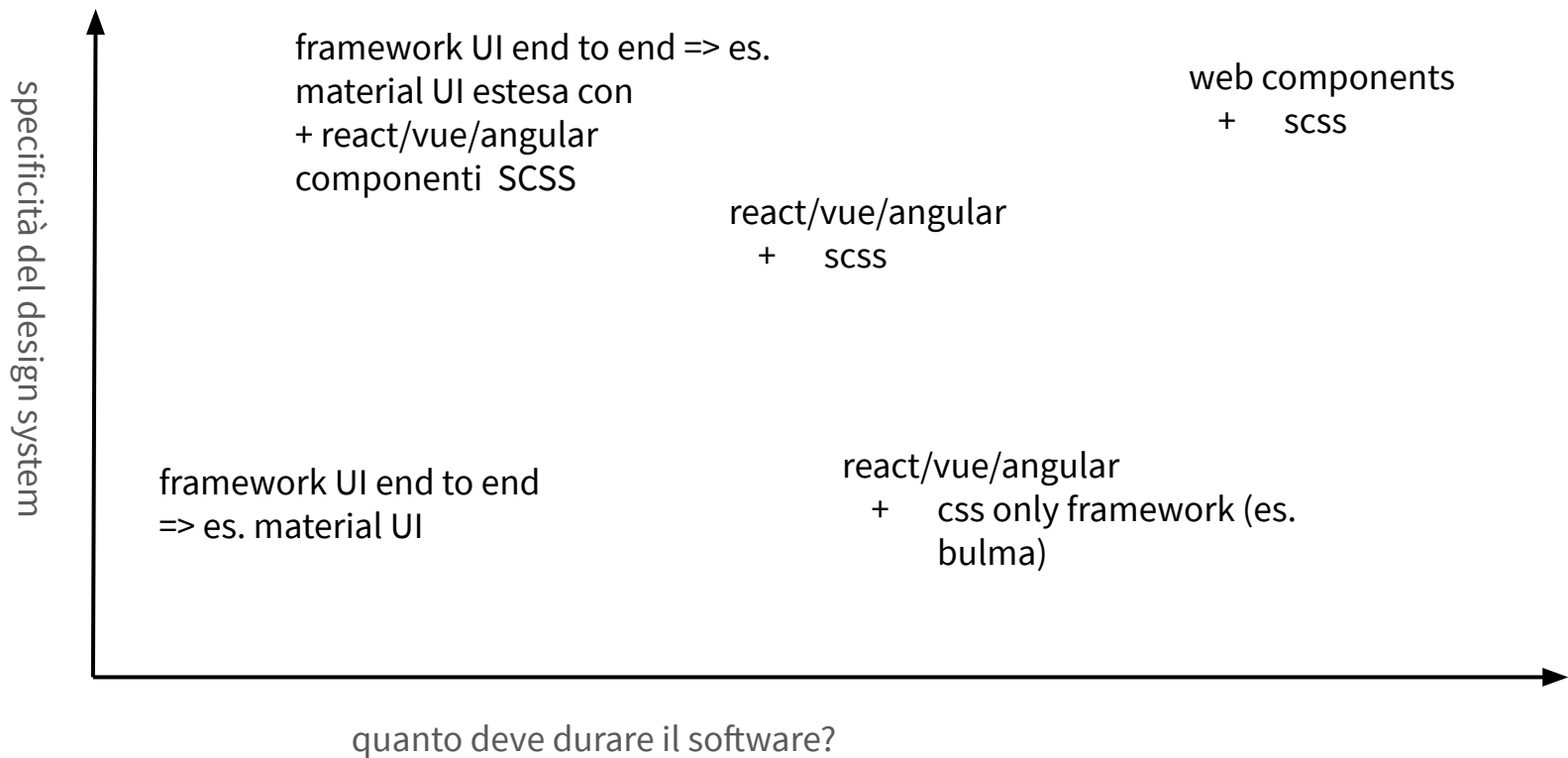
- 1. Costosa**
- 2. Compatibilità**
- 3. Prestazioni**
- 4. Web Standard**

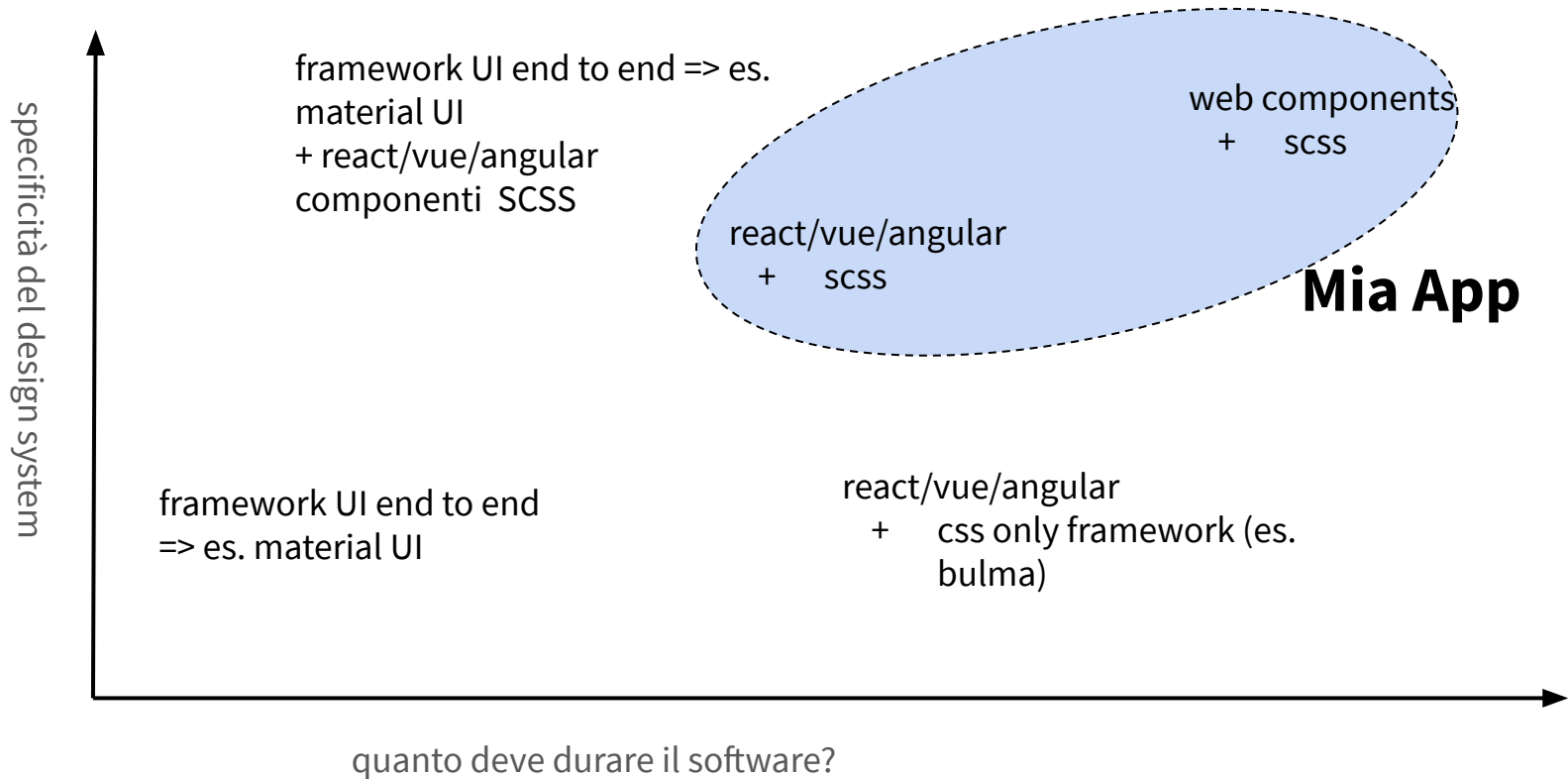
—

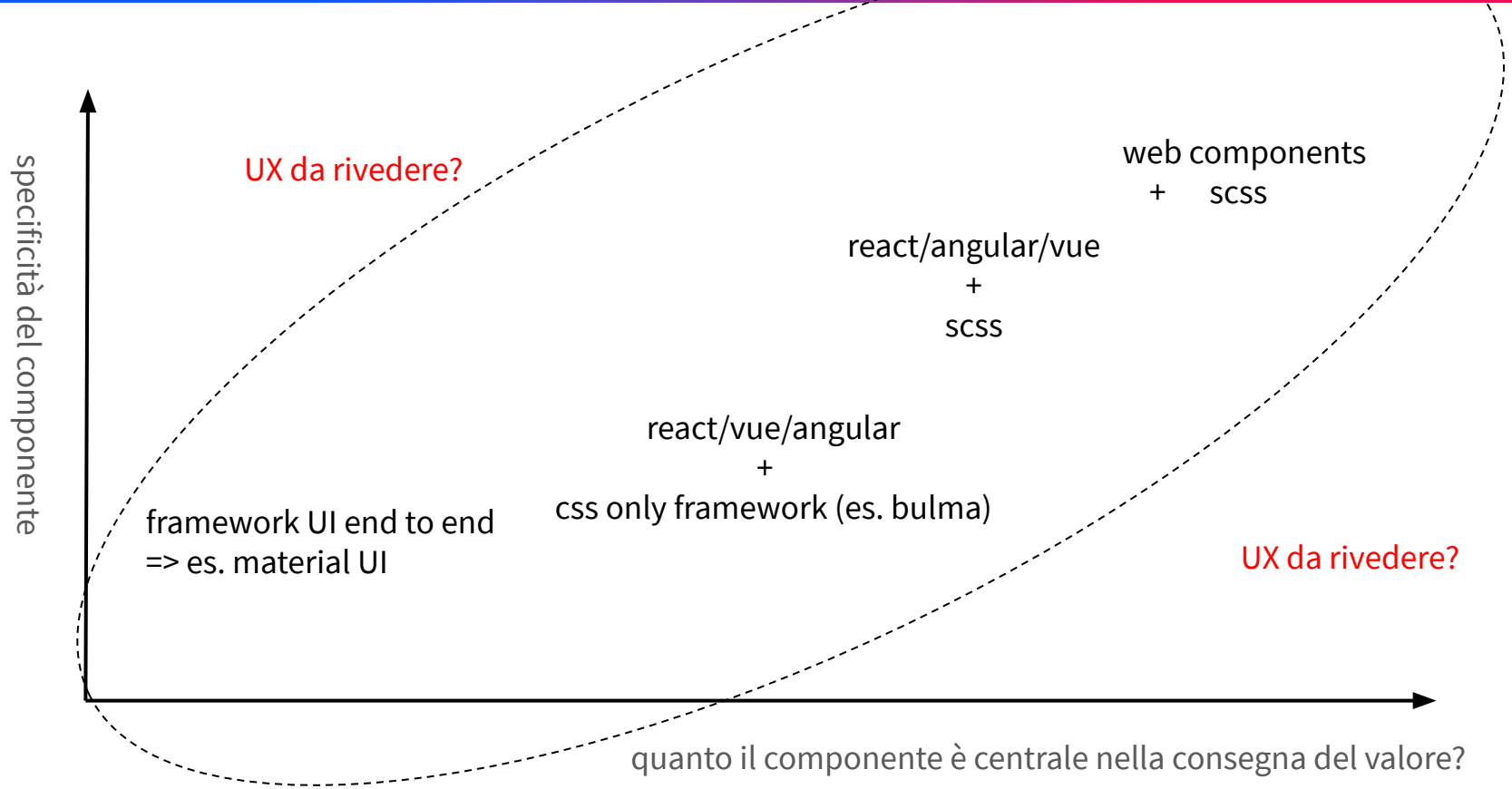
**Scelta Consapevole,  
come si fa?**











---

# Conclusioni

**FLOWING<sup>®</sup>**  
extrategy



**Dan North**

@tastapod

Following



In risposta a [@ygrenzinger](#), [@houssamfakih](#) e [@flowchainsensei](#)

Think of code as surgery to get well. You want minimal surgery, but you don't want crap surgery!

FLOWING® extrategy

# FLOWING<sup>®</sup> extrategy

**Antonio Dell'Ava**

@creativecaos

antonio.dellava@flowing.it