
Programming Simplicity



See Also

References [https://guitarvydas.github.io/
2024/01/06/References.html](https://guitarvydas.github.io/2024/01/06/References.html)

Blog <https://guitarvydas.github.io/>

Blog [https://publish.obsidian.md/
programmingsimplicity](https://publish.obsidian.md/programmingsimplicity)

Videos [https://www.youtube.com/
@programmingsimplicity2980](https://www.youtube.com/@programmingsimplicity2980)
[see playlist “programming simplicity”]

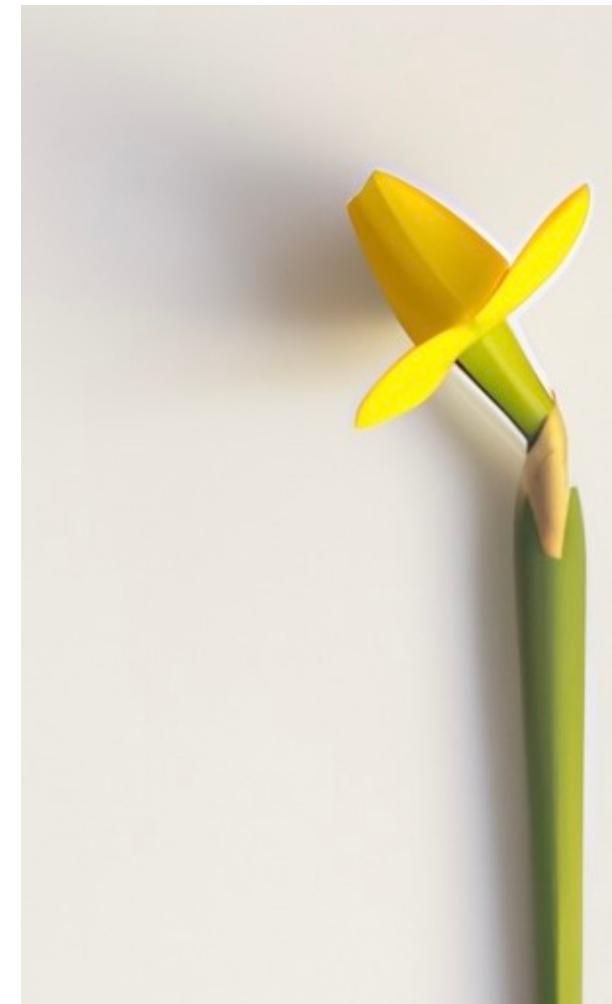
Discord <https://discord.gg/Jjx62ypR>

X (Twitter) [@paul_tarvydas](#)

More writing (WIP): [https://leanpub.com/u/
paul-tarvydas](https://leanpub.com/u/paul-tarvydas)



Simplicity



Simplicity is the lack of nuance.

Simplicity for whom?

- Users?
- Developers?
- Analysts?

“Nuance” means different things in different situations, hence, “simplicity” means different things to different people.

MVI Engineering (Minimum Viable Implementation) is very different from Production Engineering.

UX

User eXperience.



DX

Developer eXperience.



AX

Analytic eXplanation.



End-user Simplicity

- No options.
- Tactile hardware.
- Easy to use and to interconnect.
- Build and Forget†.
- Asynchronous†.

Developer Simplicity

- Computer tools for decreasing development turn-around time
- Ignore “efficiency”.
- Build and Forget. Weakly coupled - easy to rearrange†.
- Asynchronous†.
- MVI† first. “Efficiency” is achieved later by handing-off to Product Engineers for refactoring.

Analytic Simplicity

- Abstraction.
- Notational brevity, e.g. Greek letters and mathematical symbols instead of words or phrases.
- Strongly coupled†.
- Synchronous†.

† *Build-And-Forget, Strong Coupling, Weak Coupling, Asynchronous, Synchronous, MVI* - see elsewhere.