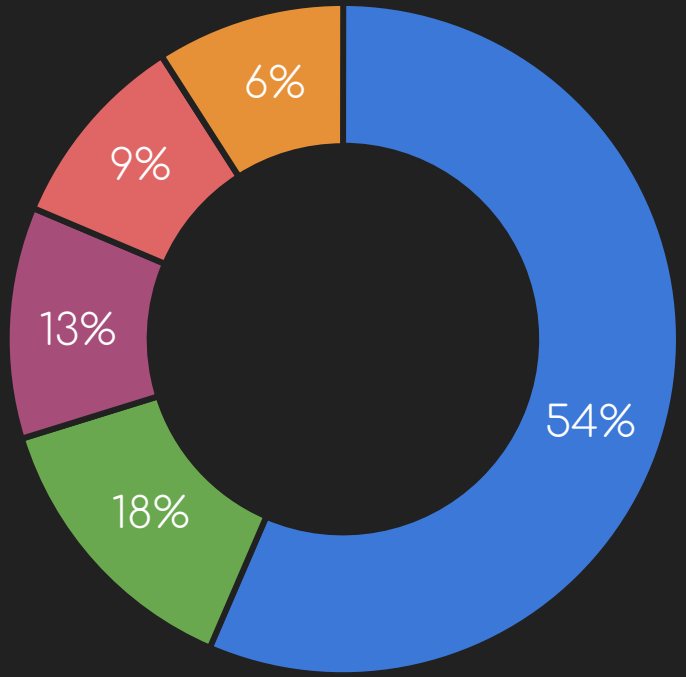
How To Build Future-Proofed Applications

Masking Technology
hello@masking.tech
linkedin.com/company/maskingtechnology

Masking Technology

Biggest challenges of software development

- Integrating different systems and technologies
- Understanding user requirements
- Finding and hiring right developers
- Continuous performance management
- **Adapting to changing client requirements**



(GoodFirms. (2024, November 8). Remarkably Useful Stats and Trends on Software Development)

Future-Proofed Applications

Adaptable, without heavy refactoring and breaking beyond repair.

- Wikipeter -

Extendable

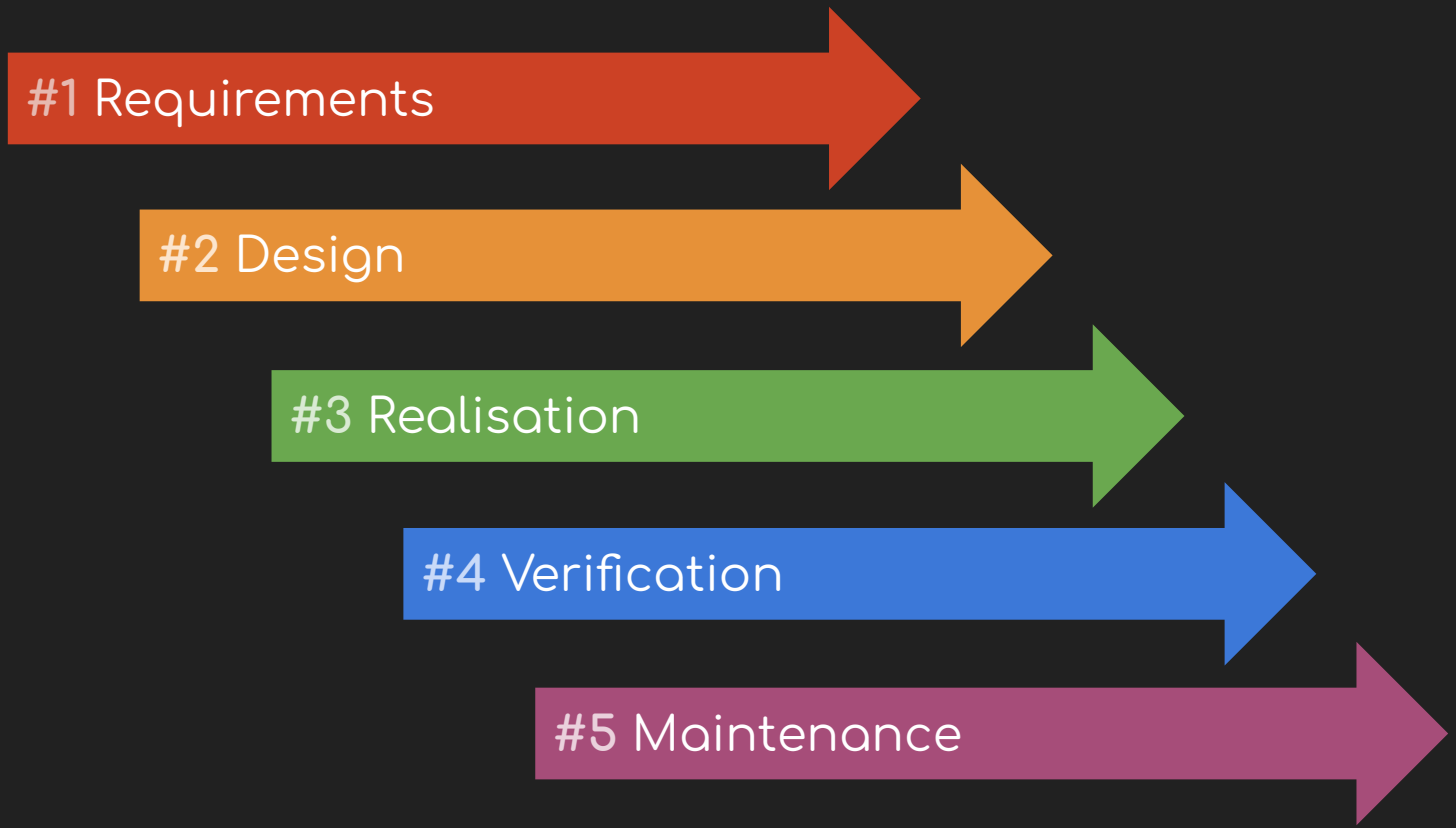
Modifiable

Reducible

Time & cost

Productivity

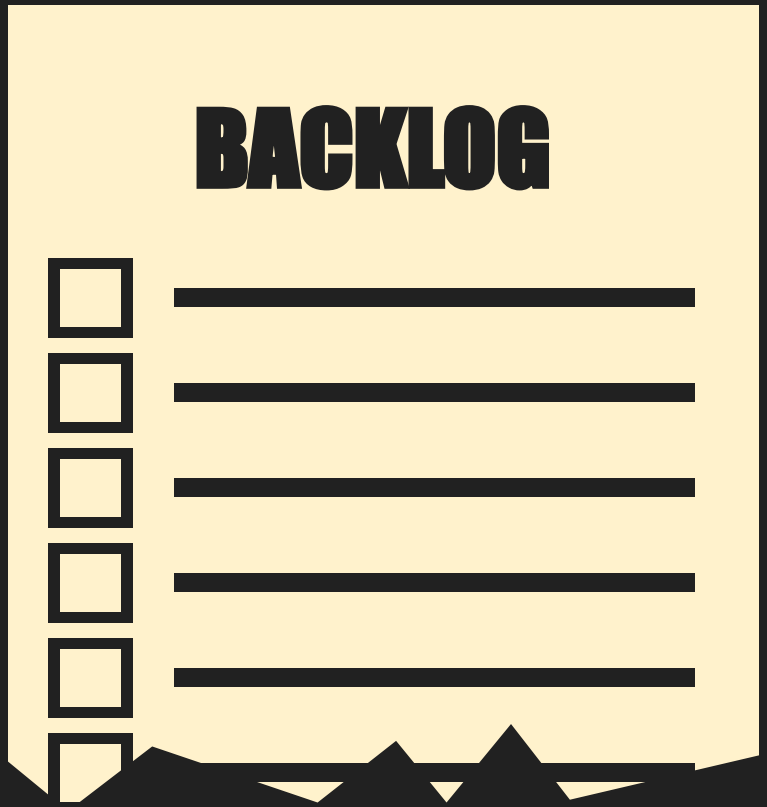

5x


#1 Requirements
#2 Design
#3 Realisation
#4 Verification
#5 Maintenance

#1

Keep it Small

LESS is MORE


BACKLOG

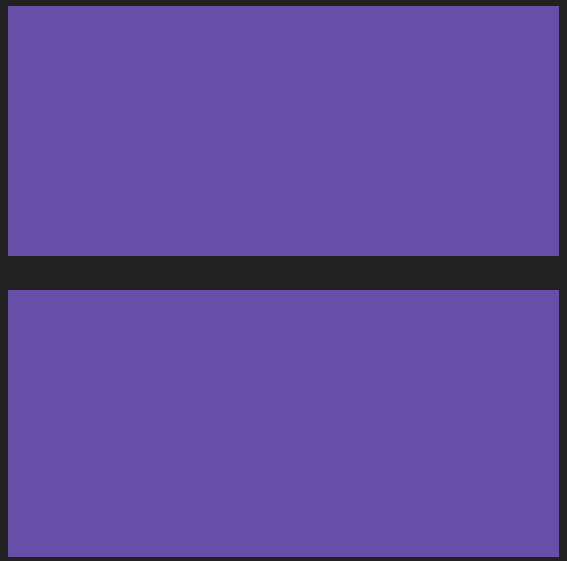
- Must haves
- ~~Should haves~~
- ~~Could haves~~
- ~~Would haves~~

Acceptance criteria

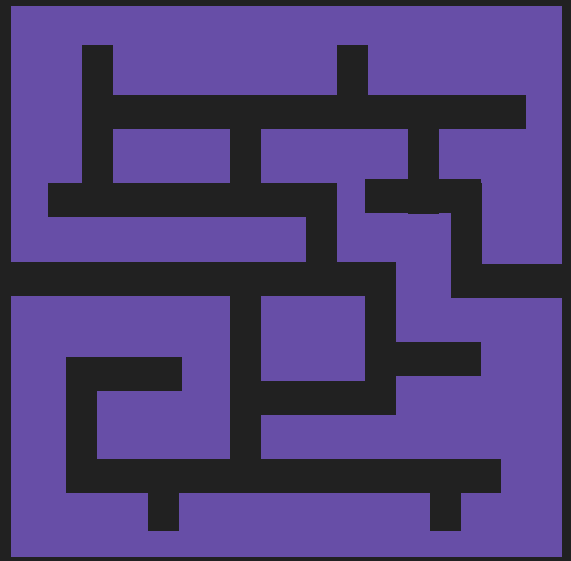
- [ ] Understandable
- [ ] Value > Effort
- [ ] Value > Risk
- [ ] Stable

#2

Keep it Simple



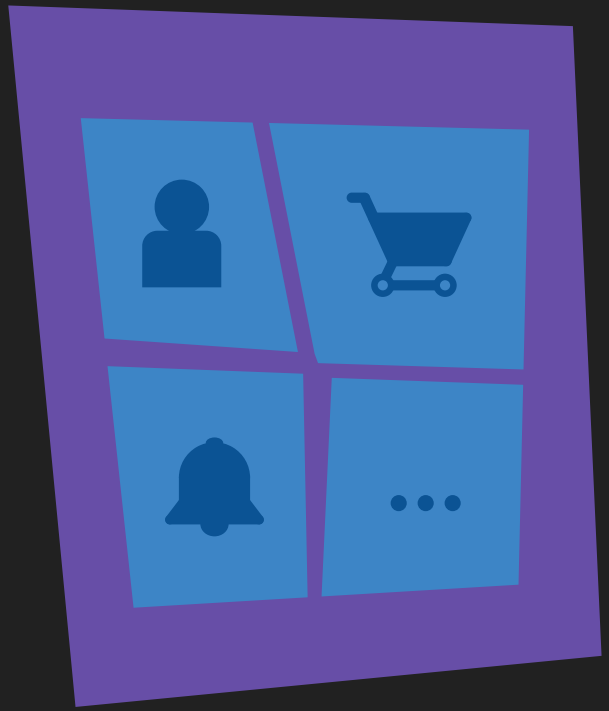
VS



Simple

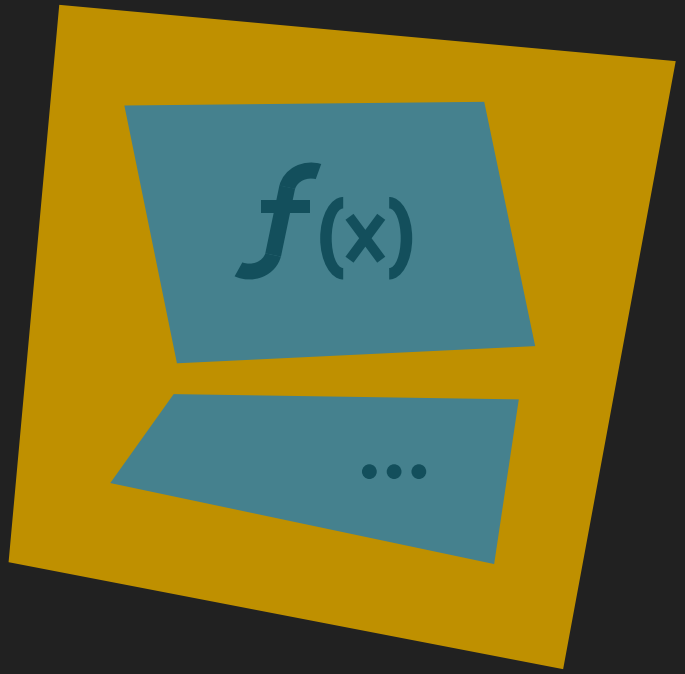
Complex

Good
architecture
is fundamental



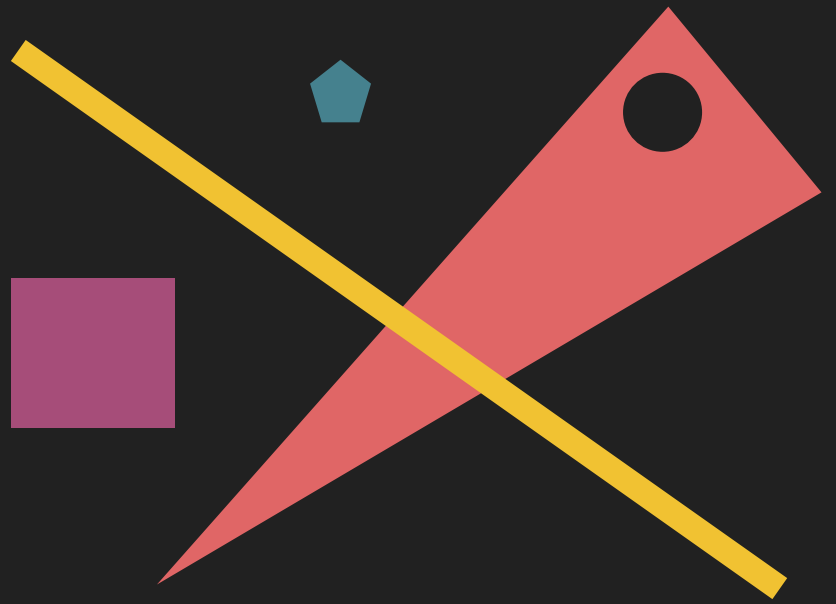
Strong
modularization
is crucial

Strong
componentization
is essential

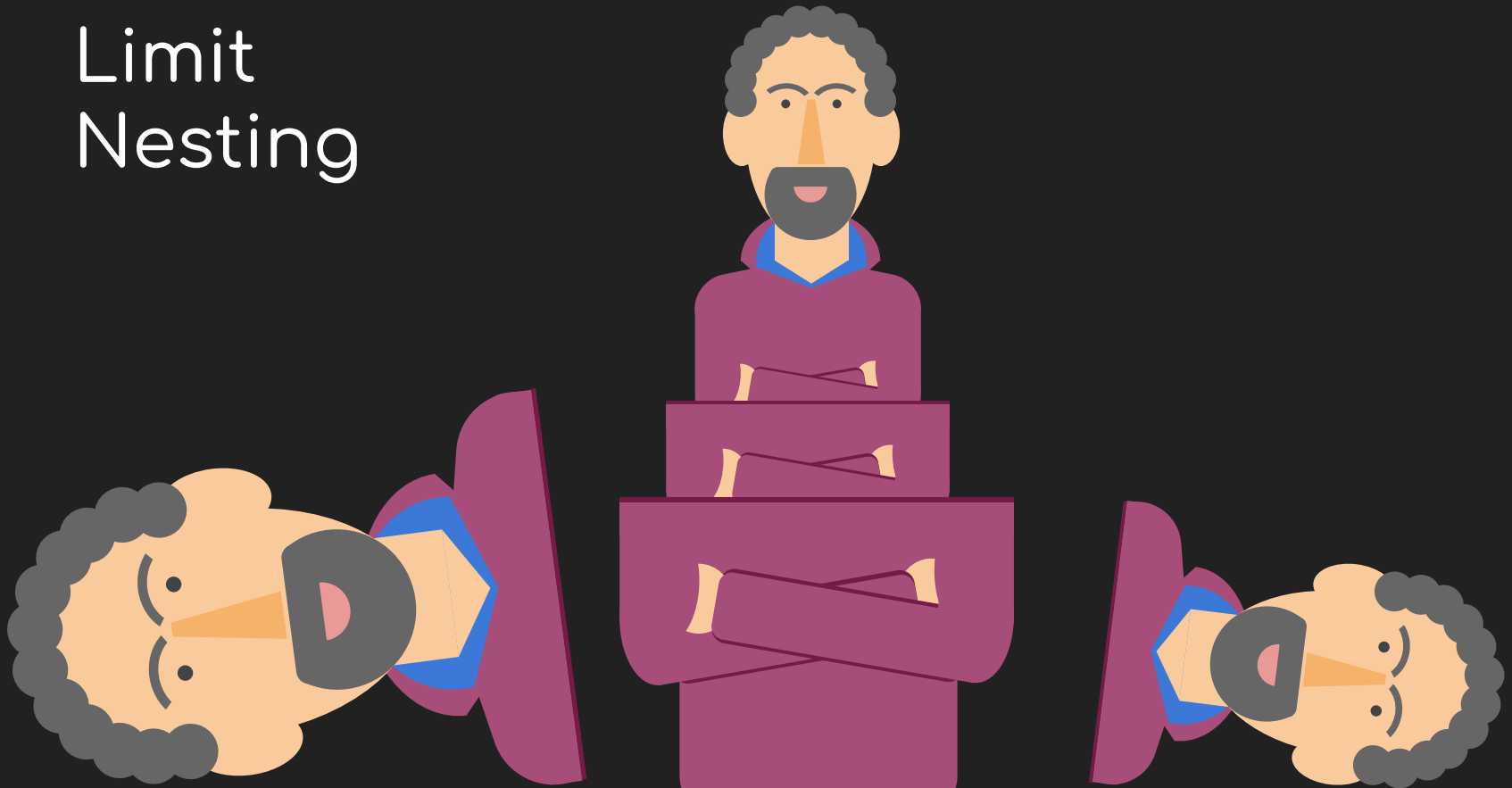


Strong
encapsulation
is vital

Limit Abstractions



Limit Nesting



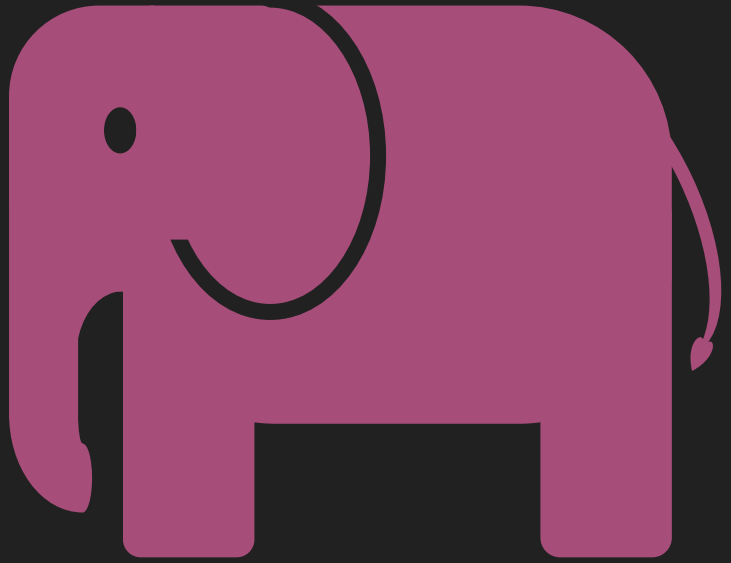
#3

Keep it Elegant

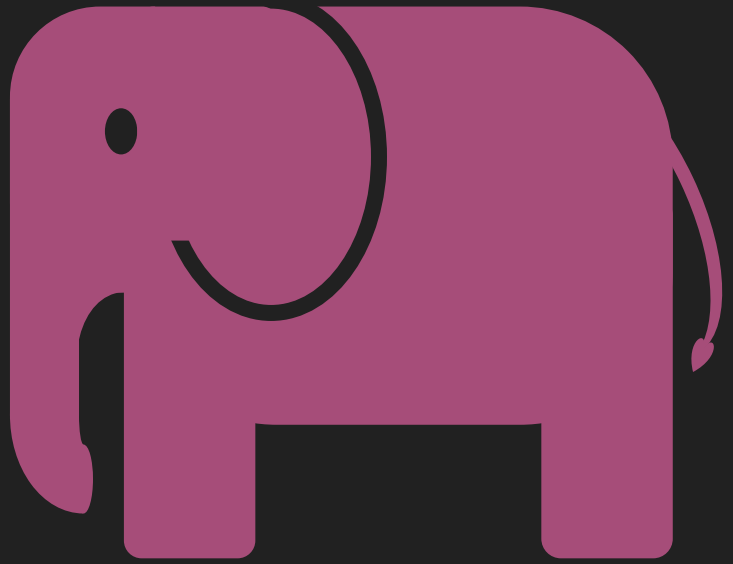
Code is ~~poetry~~

documentation

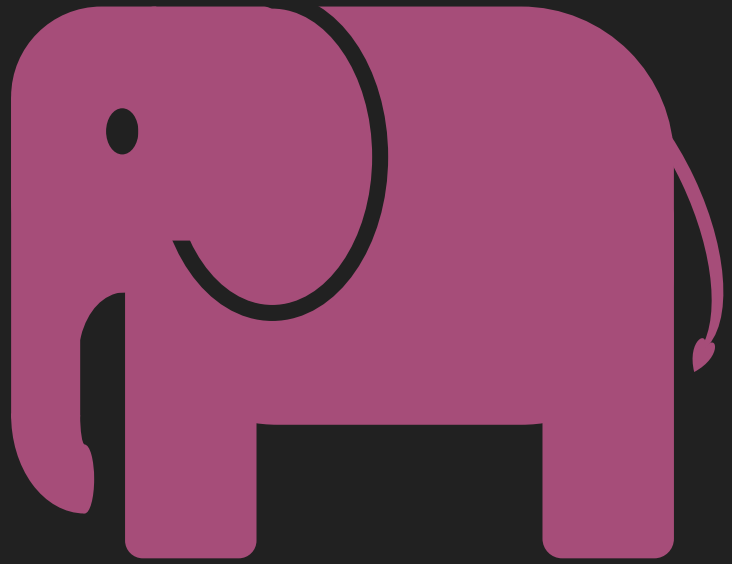
// This is a slide



Dining table



Dining table



Hair dryer

78

629

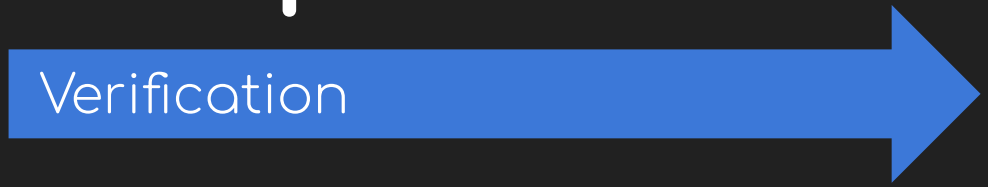
6

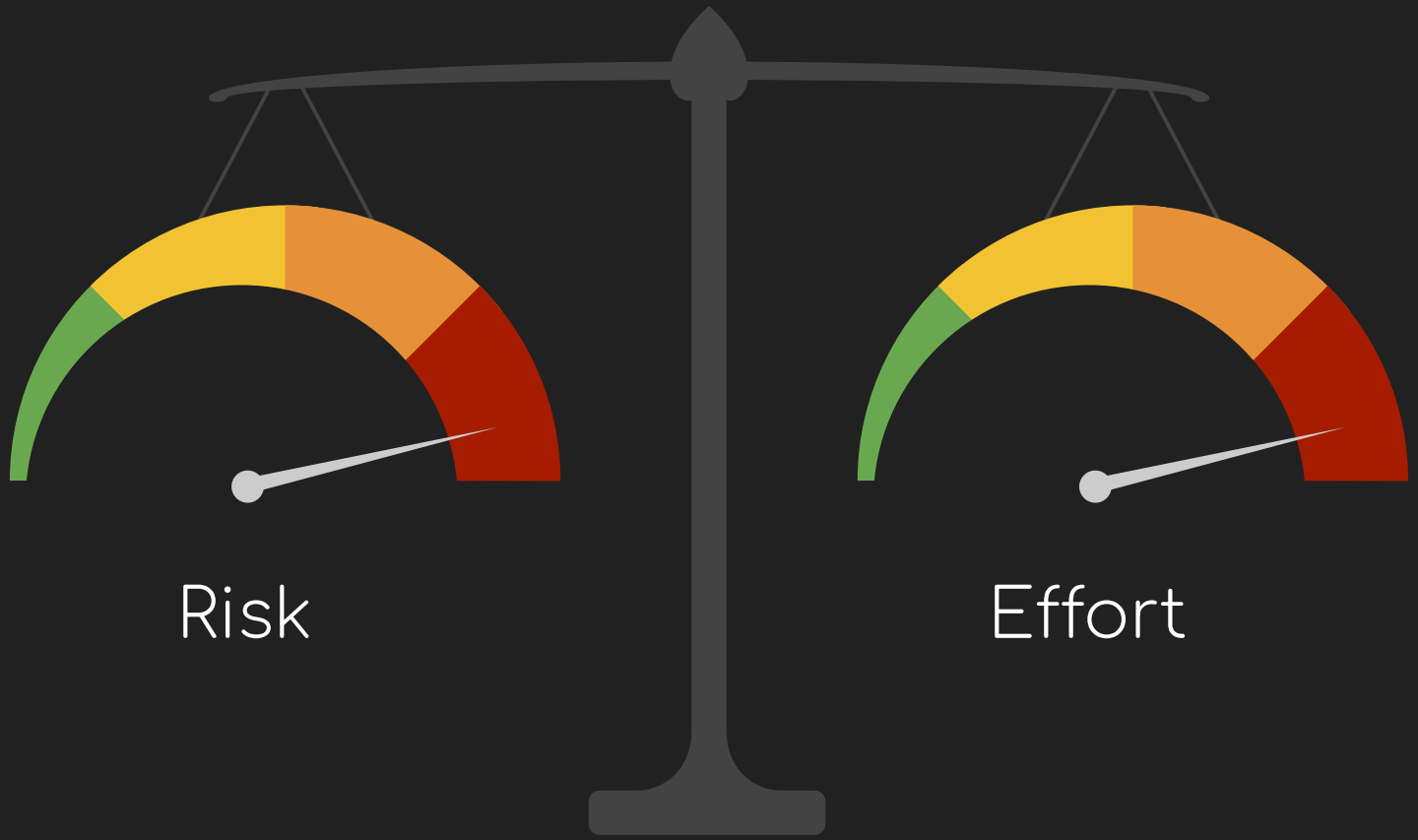
402

32

#4

Keep it Safe




Risk
Effort

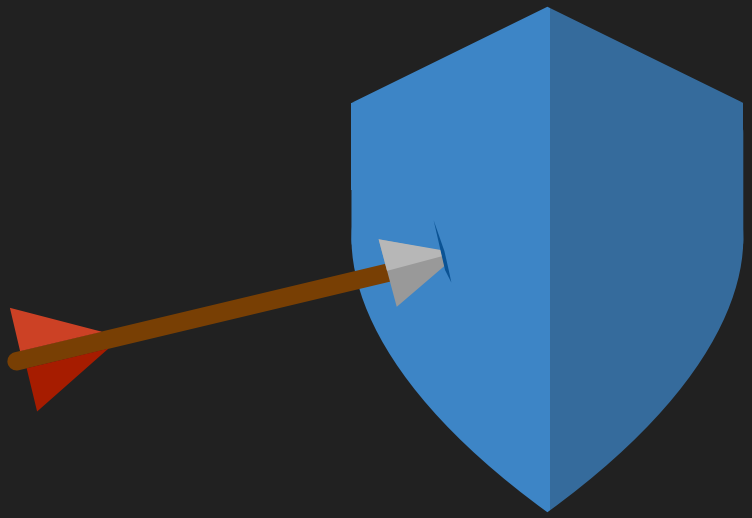
Use case

Misuse case

Ensure correct operation

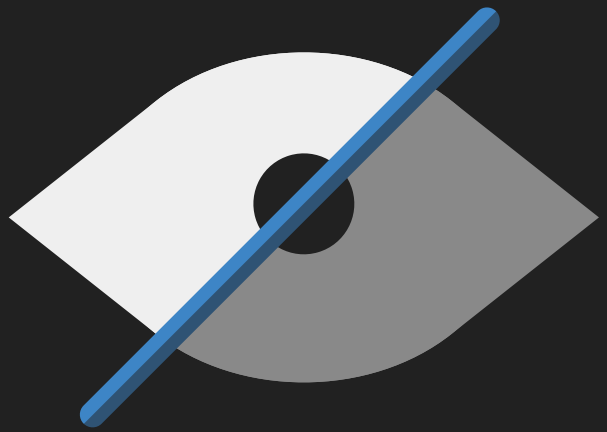
- [ ] User Interface
- [ ] Use Cases
- [ ] Integrations

Ensure security measures



- [ ] Access Control
- [ ] Content Validation
- [ ] Traceability

Ensure privacy measures

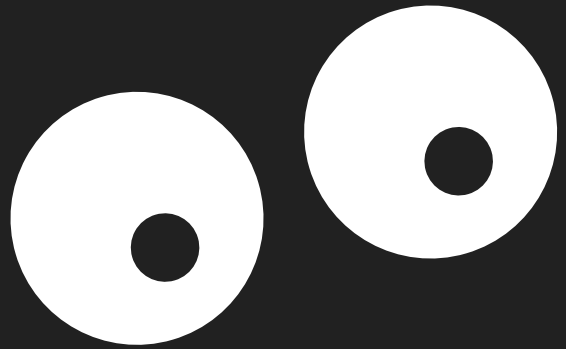


- [ ] Agreements
- [ ] Storage location
- [ ] Data Encryption

Ensure market value

- [ ] User Feedback
- [ ] Competitive Analysis
- [ ] Regulatory Compliance

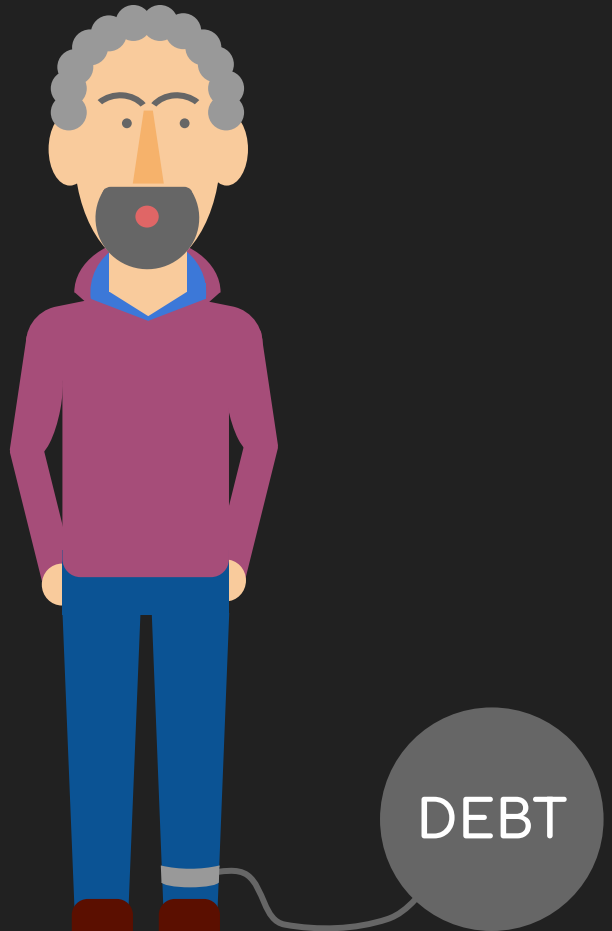
Ensure development continuity

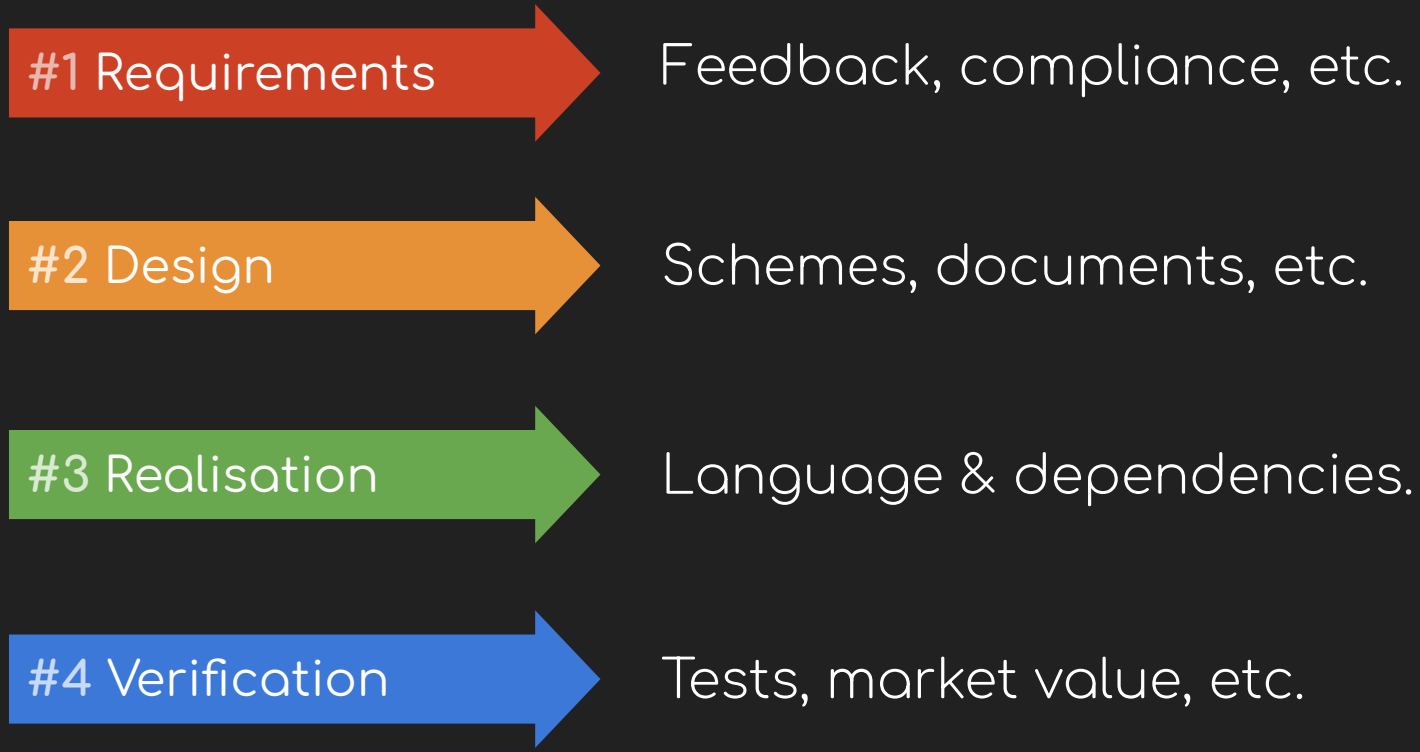


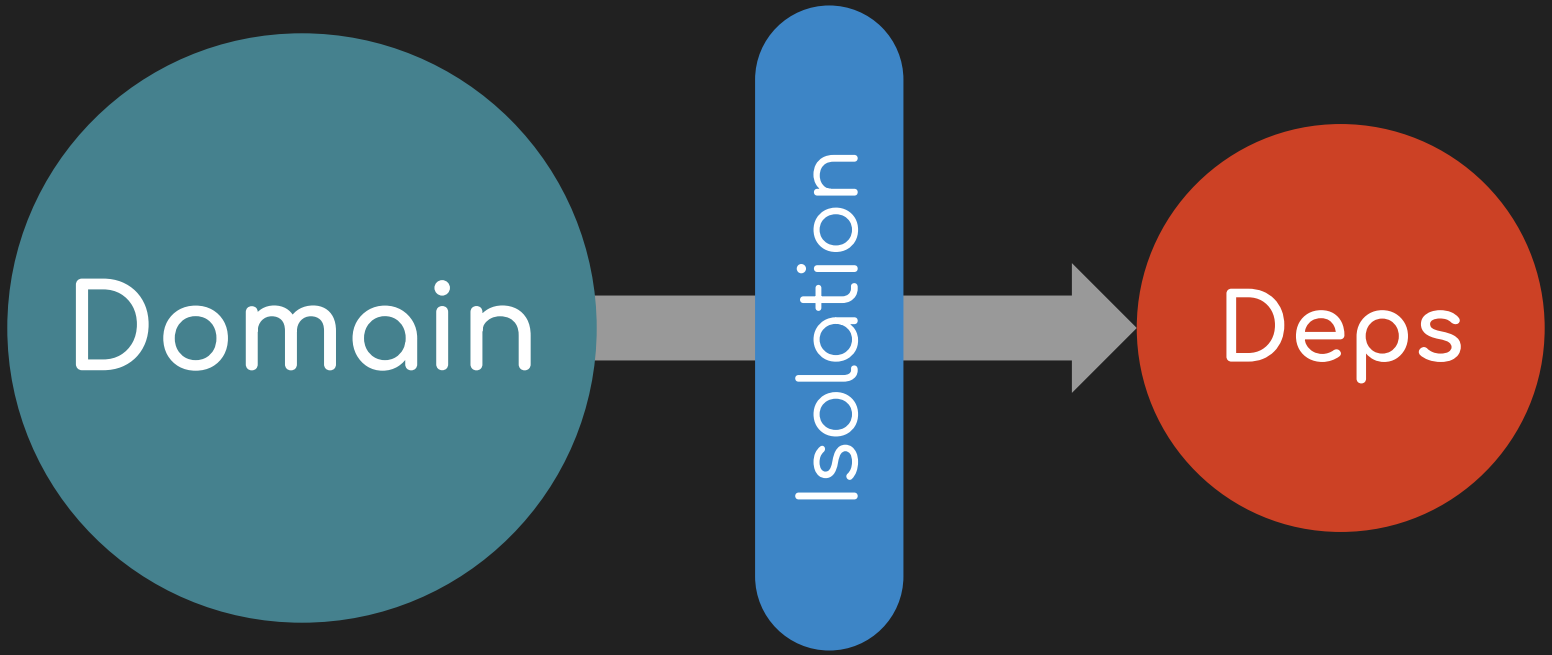
- [ ] Work in pairs
- [ ] Review all work

#5

Keep it Updated


DEBT

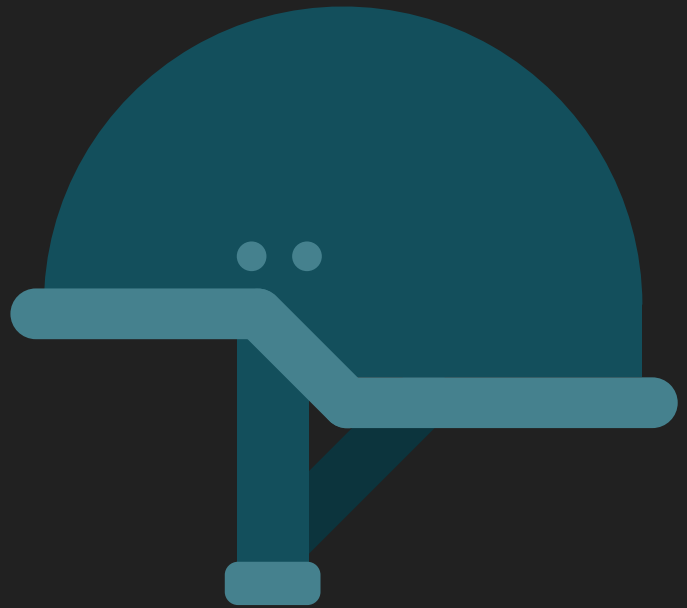

#1 Requirements
Feedback, compliance, etc.
#2 Design
Schemes, documents, etc.
#3 Realisation
Language & dependencies.
#4 Verification
Tests, market value, etc.

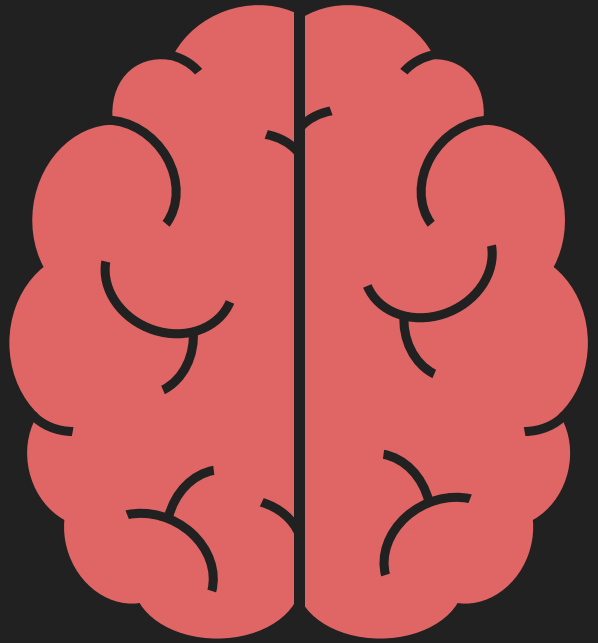

Domain
Isolation
Deps


PLEASE
REMOVE
OBSOLETE
FEATURES

How to apply

Adaptability
requires
discipline





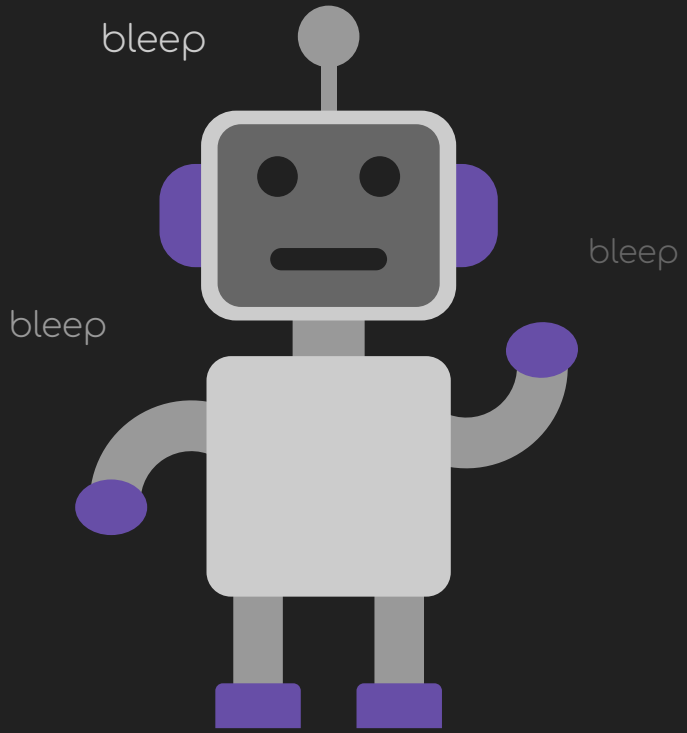
But also a
growth mindset

Alignment

Users

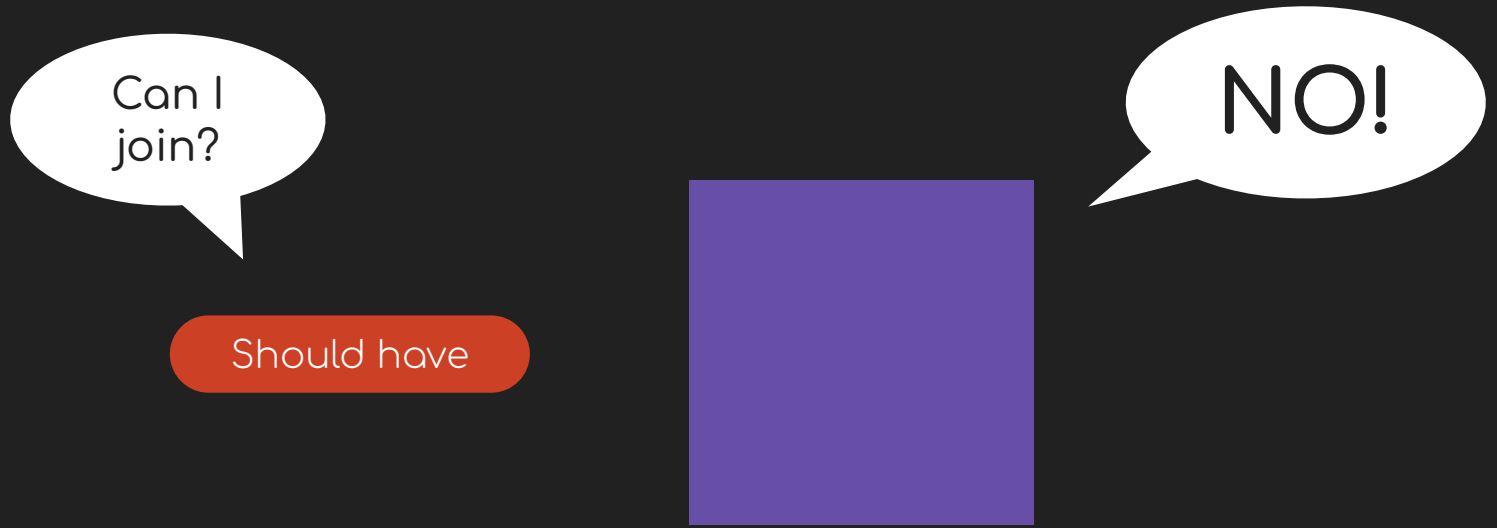
Business

Development

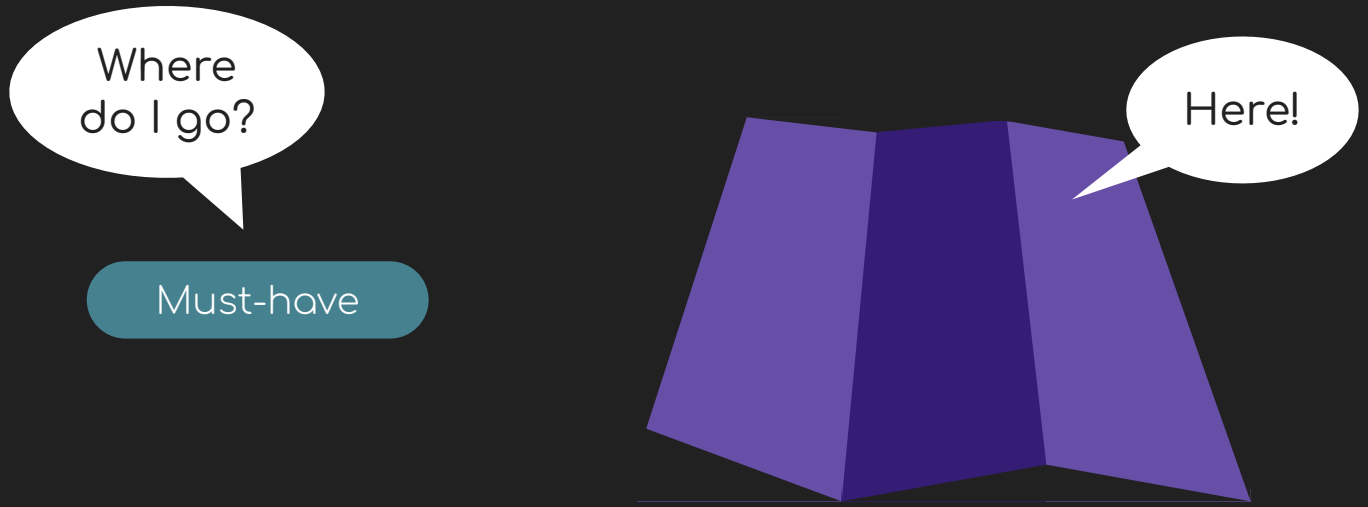


Focus on **Automation**

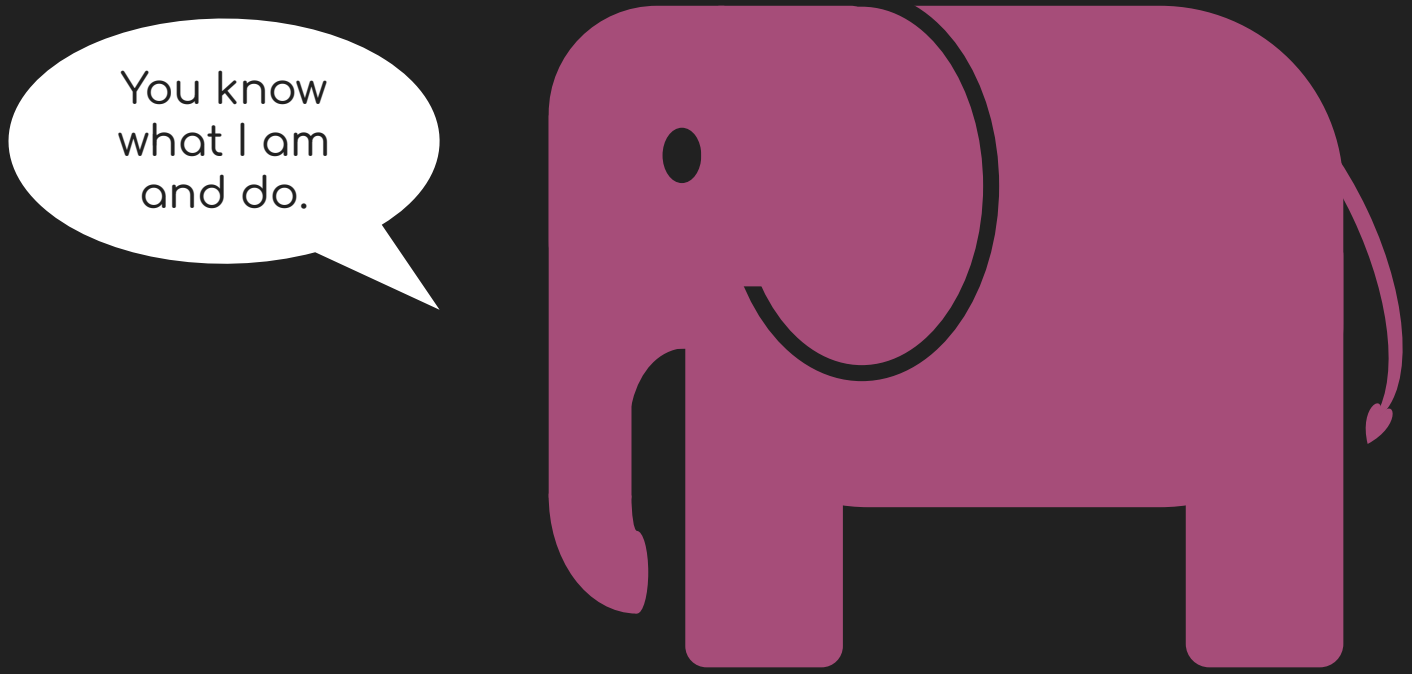
Key takeaways



Strong requirement selection



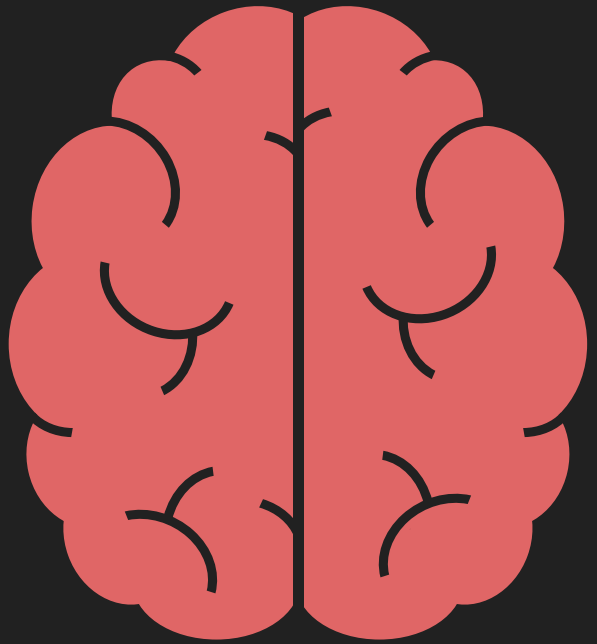
Navigable application architecture



Self explanatory code

Risks mitigated

Debt cleared



Keep
thinking

Thanks!

https://masking.tech