

ARCHITECTURE EVOLUTIONS

G. Molines

2020-2021





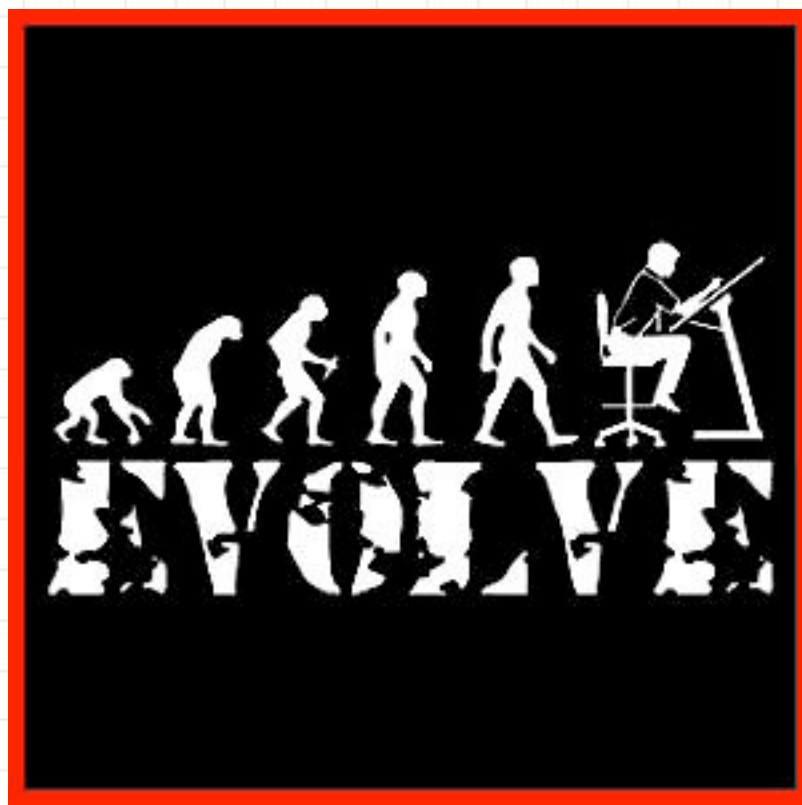
TIME EVOLUTION

Architecture's life

Can you define the architecture at the beginning of the project and be done with it?

Yes? congratulations!





Why?

- Legal
- Finance
- Competition
- Technology aging
- Maintainability
- “Fix the mess” (aka pay now or pay later)

And of course...

- Evolving requirements

But why ???

- Because you have a **problem**....
- Example: VTT move from DB2 to MongoDB

→ Notion of architectural smell

How

- No scientific method
- Avoid big-bang
- Validate each change
- → it has a cost!
- But keeping defects around also has a cost
 - Bad news doesn't age well

Iterative development

Define

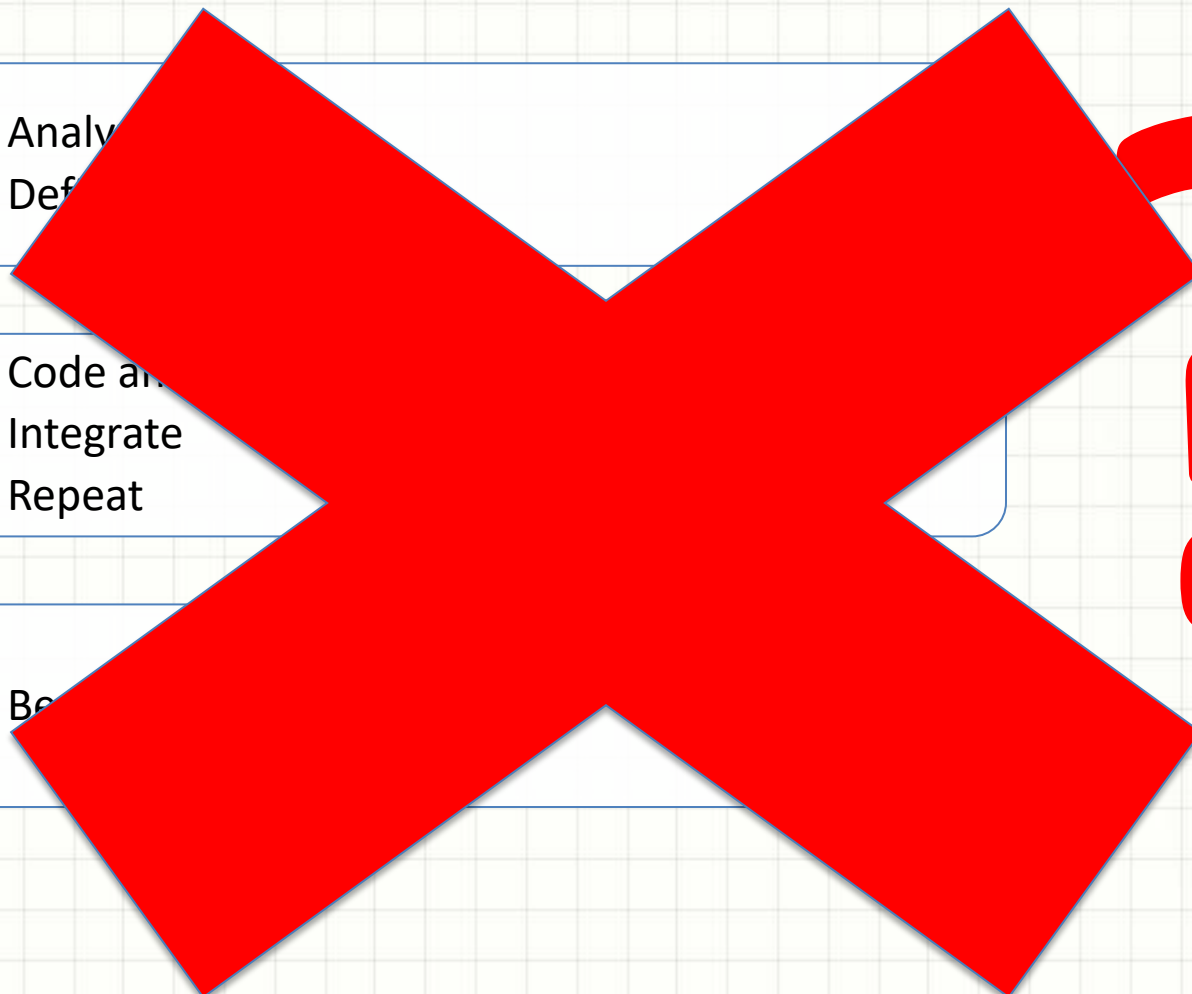
- Analyze
- Define

Iterate

- Code and
- Integrate
- Repeat

Ship

- Be



Iterative development

Define

- Analyze
- Define *Draft* Architecture

Iterate

- Code and Test
- Integrate
- **Question architecture, evolve it**
- Repeat

Ship

- Be done with it... kind of



Techniques

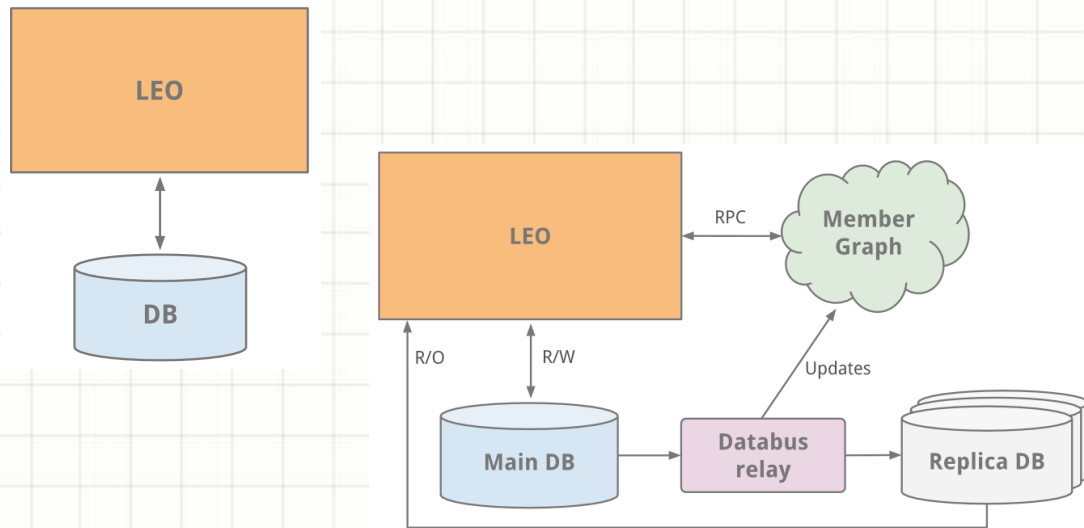
- Deprecation
- API resiliency over time
- Façade
- Rolling replace (or rolling upgrade)



EVOLUTION IN SPACE

Scaling

- Real life example: [Linked In story](https://engineering.linkedin.com/architecture/brief-history-scaling-linkedin)



SOA

<https://engineering.linkedin.com/architecture/brief-history-scaling-linkedin>

Execution platform

- Desktop to embedded in device



- Server to Cloud



THE EVOLUTION OF SOFTWARE ARCHITECTURE

1990's

SPAGHETTI-ORIENTED
ARCHITECTURE
(aka Copy & Paste)



2000's

LASAGNA-ORIENTED
ARCHITECTURE
(aka Layered Monolith)



2010's

RAVIOLI-ORIENTED
ARCHITECTURE
(aka Microservices)



WHAT'S NEXT?

PROBABLY PIZZA-ORIENTED ARCHITECTURE



QUESTIONS?