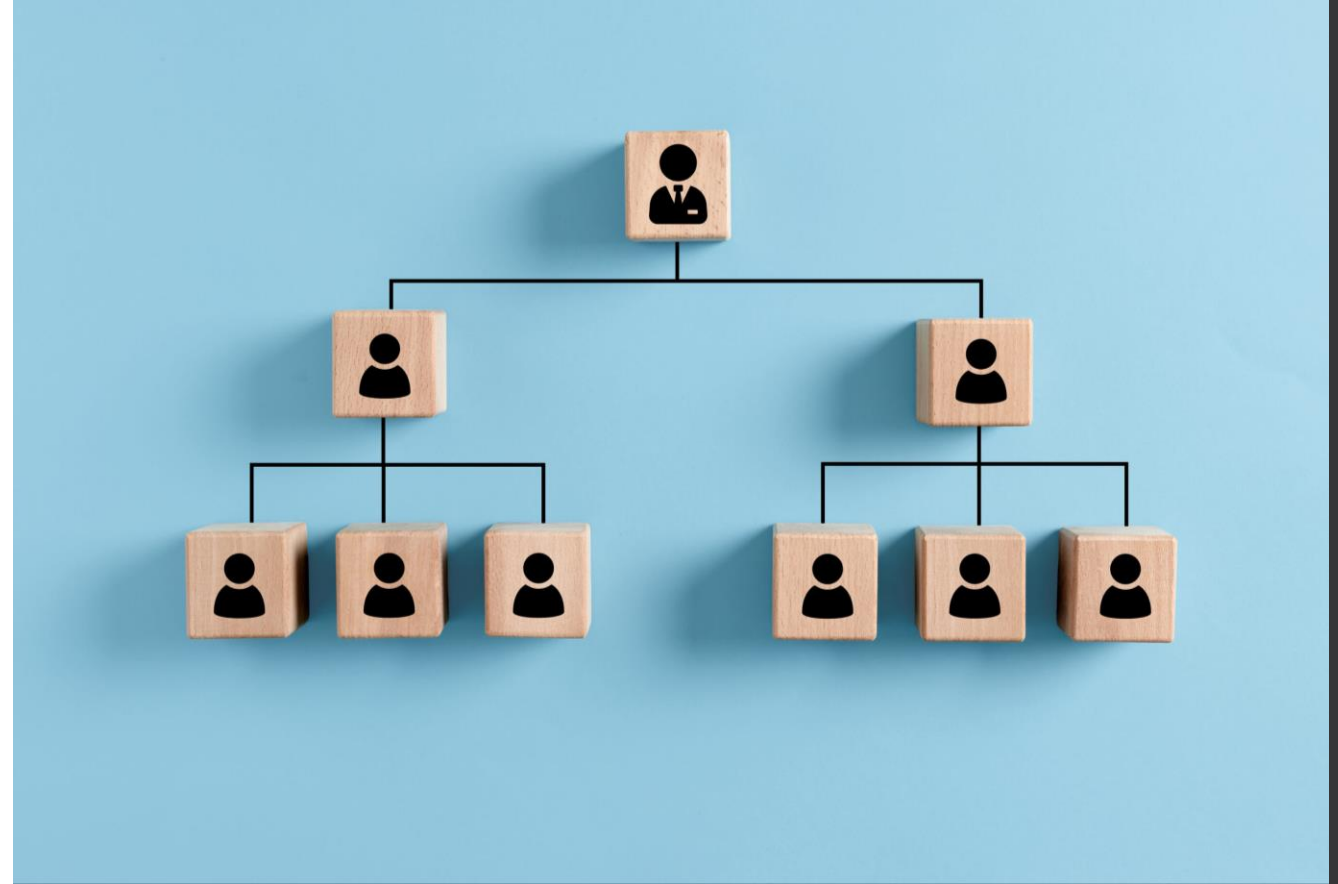# TEAM TOPOLOGIES

What will we cover?

- What is Team Topologies?

- Team Types

- Key Concepts

- Team types

- Implementation of Team Topologies

- Challenges

WHAT IS TEAM TOPOLOGIES ?

- Organize for a **fast flow value**.

- **Team-of-teams** organization.

- From IT space. Applicable to **all knowledge work**.

Team Topologies principles

1. **Fast flow value**. Decouple teams to be **quick**.

2. Be mindful of **team size**. **Small teams, high trust**.

3. **Cognitive / mental load**. Try to ease teamwork.

4. **Organizational architectures** for the system.

5. **Conway's Law**. Communication pathways & architecture.

6. Expect your software to become **obsolete very soon**.

KEY CONCEPTS
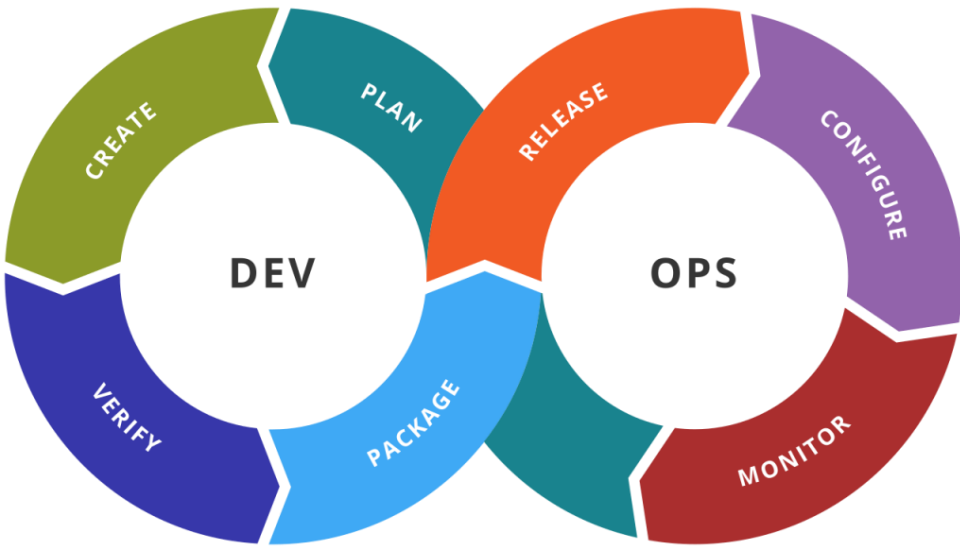
DevOps combines practices and tools to deliver applications and services faster

# Origins of Team Topologies

- Start 2008 influenced by **DevOPs**.

- Eliminate **IT bottlenecks**

- Example of the new methods: continuous delivery.

- **New tools**, more **variety** of options

- Work more **quickly**

- **Effective development and teams working together**

Principles vs team types

- Focus on **team types** and renaming teams.
  - Understand **underlying purposes and principles**.
    - Rationale behind creating platform groupings.
    - Purpose of the different team types.
  - Driven by **flow** and lowering **cognitive load**.
    - Less cognitive load more efficiency
    - A flow-based approach is effective and humane.

Principles of Team Topologies vs team types

Complicated-subsystem team

Enabling team

Stream-aligned team

Platform team
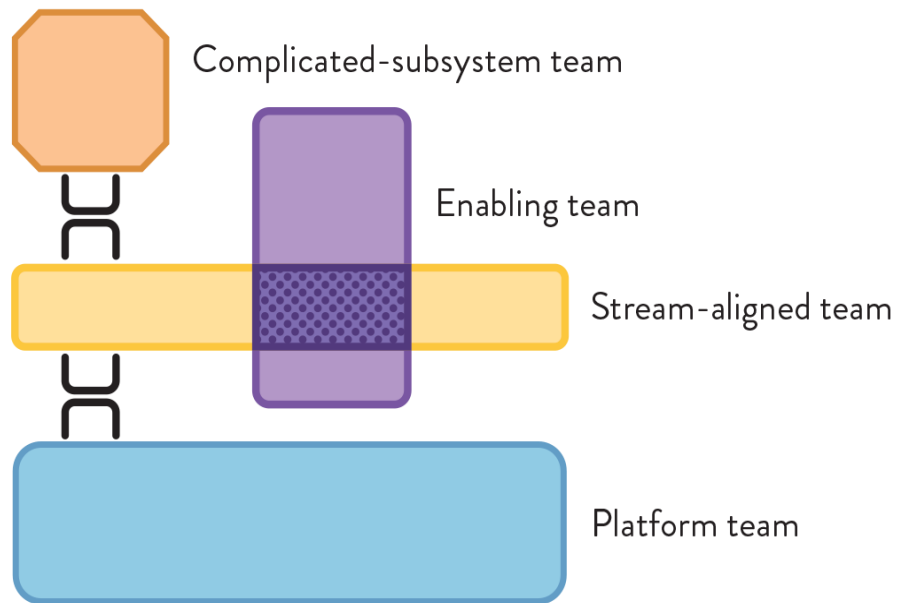
- **Only** team types
  - o **Misinformed**
  - o **Counterproductive**
    - Manager implement without necessary changes
    - Not value stream approaches
      - o Maximize value
      - o Minimize costs
- No reorientation, no flow.

Deliberate Desynchronization

Technique to **limit synchronization** between teams.

Synchronization **increases cognitive load**.

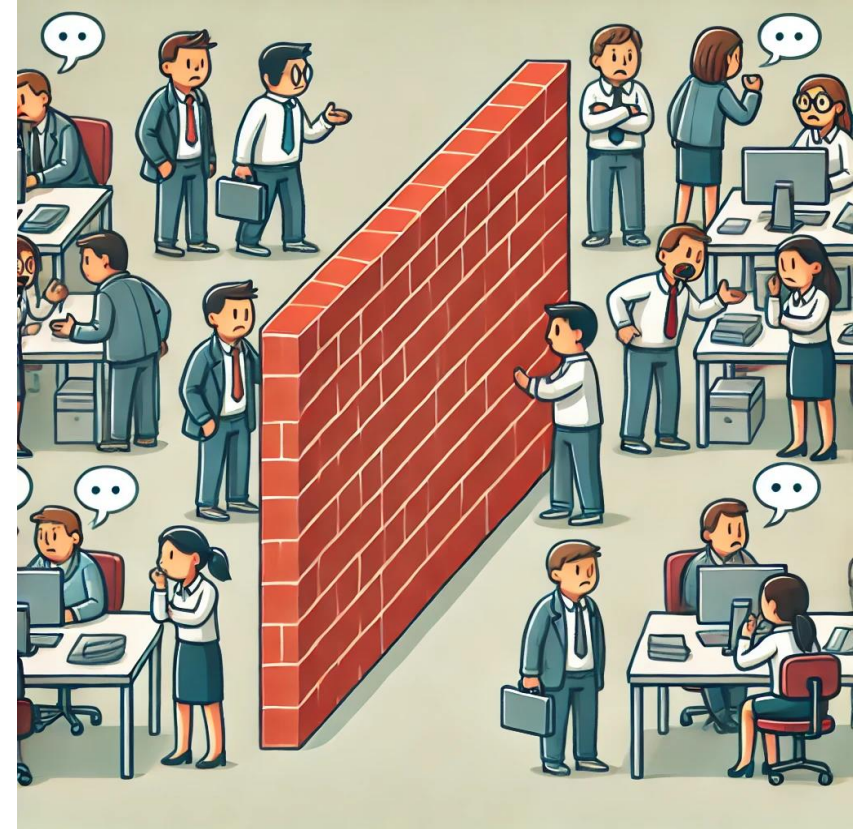If large projects, too many concerns.

Consequences

- Reduces **waiting time**

- **Lowers costs**

- **Value** appears sooner

Encapsulate problem and assign to a team.

When desynchronization goes **too far**

- **Too many** teams **isolated**
  - Mirrors **mini agencies**

- **Fragmented** value
  - Disjointed results
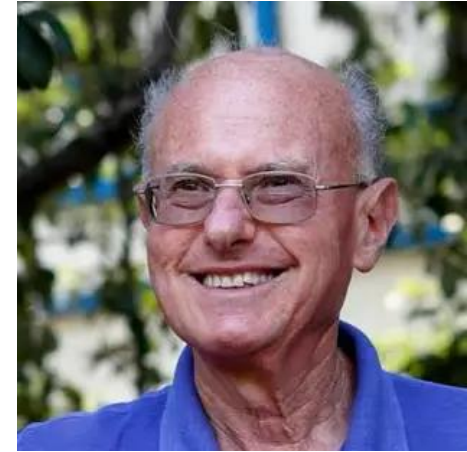 - Duplicated efforts
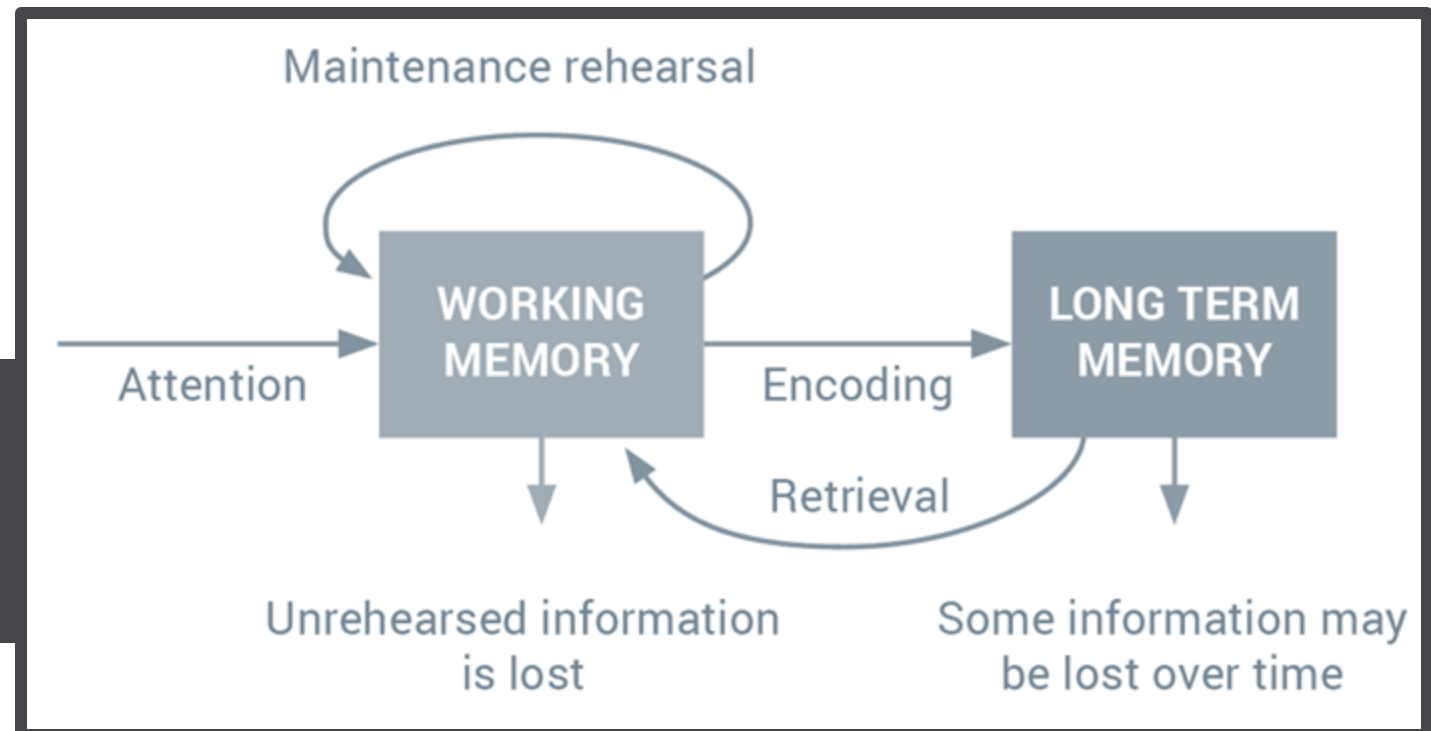 - Gaps in delivery

Deliberate Desynchronization

COGNITIVE LOAD

Coined by **John Sweller** in the 80s:

- Initially geared towards the world of education.

- "Working memory capacity is limited".

- In software, we might take more things into account.

What is it?

Maintenance rehearsal

Attention → WORKING MEMORY → Encoding → LONG TERM MEMORY

Retrieval

Unrehearsed information is lost

Some information may be lost over time

How to measure Cognitive Load
(and why it's important)

- Cognitive load is one of the main reasons to use Team Topologies

- Overloading teams can lead to a loss in productivity.

- "**Teamperature"**

- Fostering a **safe** working environment is equally as important.
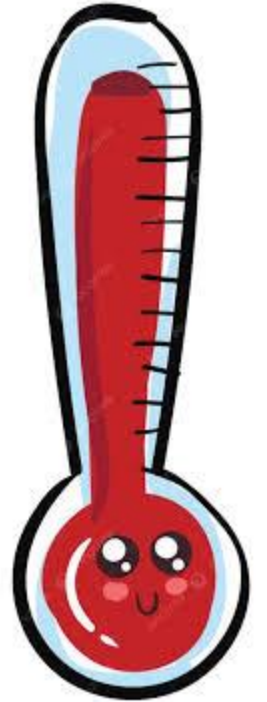
TEAM TYPES

In team topologies

- Stream-Aligned teams

- Enabling teams

- Complicated-Subsystem teams

- Platform grouping & Platform team

Stream-Aligned teams

- Single stream of work

- Small

- Enough skill

- Full spectrum
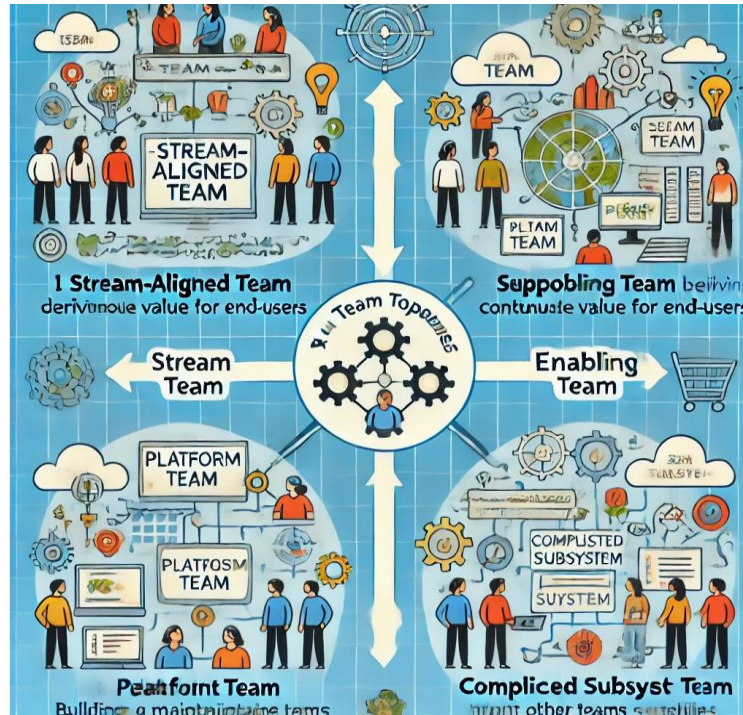
- Help Stream-Aligned teams

- Suggest new technologies or approaches

- Specialists

Enabling Teams

Complicated-Subsystem team

- Responsible of solving/maintaining a specific complex system

- More efficient than including one specialist in each stream-aligned team

- ¿When do we need to integrate a Complicated-Subsystem team?
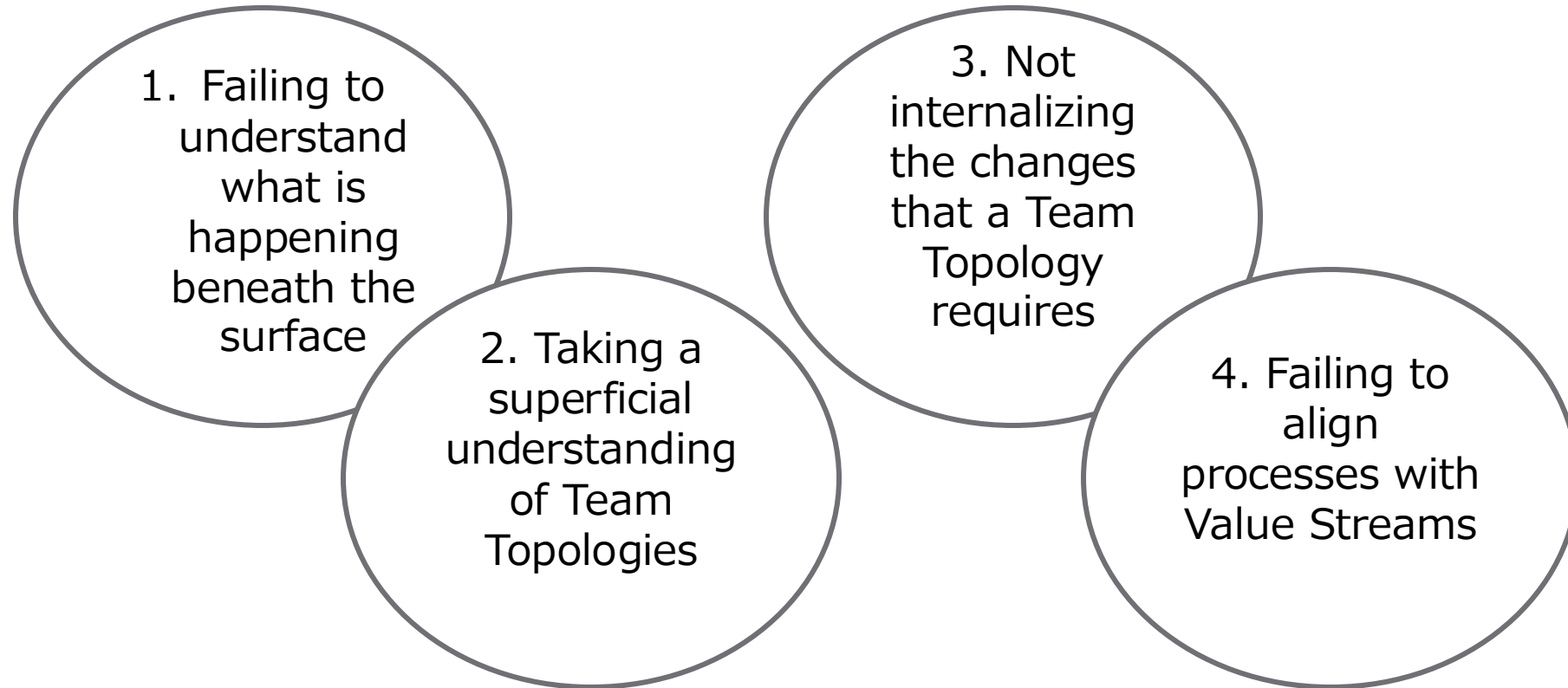
Platform Grouping & Platform Team

Platform Team

- A team responsible for building/maintaining an internal platform

- Propose: Integrate third-party services, near-complete platforms and internal services

- Their commitment is to reduce this cognitive load from the Stream-aligned team

Platform Grouping emerges when multiple Platform Teams are needed to scale and maintain a growing platform efficiently.

Common challenges of the implementation

1. Failing to understand what is happening beneath the surface

2. Taking a superficial understanding of Team Topologies

3. Not internalizing the changes that a Team Topology requires

4. Failing to align processes with Value Streams

Team Topologies implementation: How to get started?

- **"Elephant carpaccio"** exercise.

- **Pilot project**. Evolution & team (dis)engage.

- Create a **friendly FOMO**.

- **Impulse wisdom** and **curiosity**.
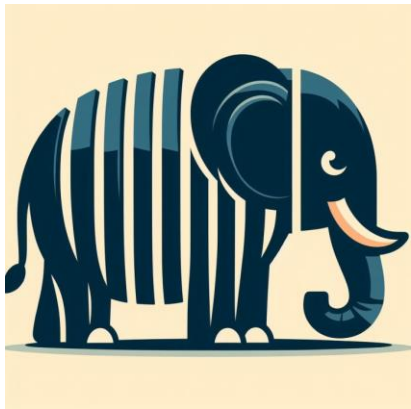
THANK YOU

Daniel Rodríguez Fernández

Francisco Cimadevilla Cuanda

Laura Labrada Campos

Vicente Megido García

Eduardo Pérez Sánchez