UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Introduction to Database Design

Basi di Dati

Bachelor's Degree in Computer Engineering
Academic Year 2024/2025

Stefano Marchesin

Intelligent Interactive Information Access (IIIA) Hub
Department of Information Engineering
University of Padua

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

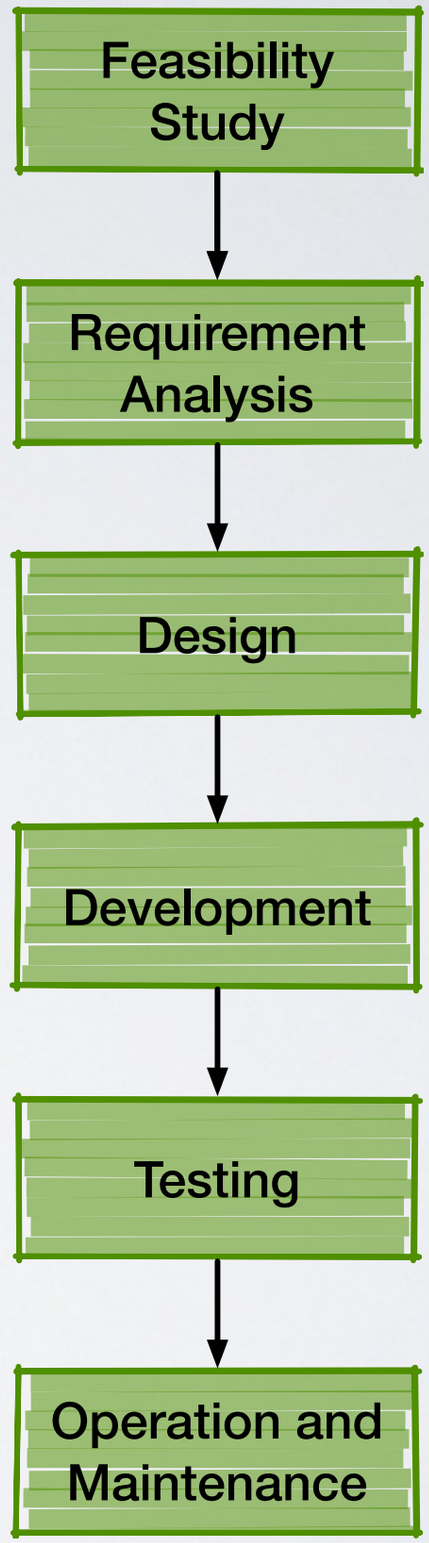
Outline

- Life cycle of an information system
- Requirement analysis
- Database design
- Conceptual, logical, and physical design

Software Lifecycle

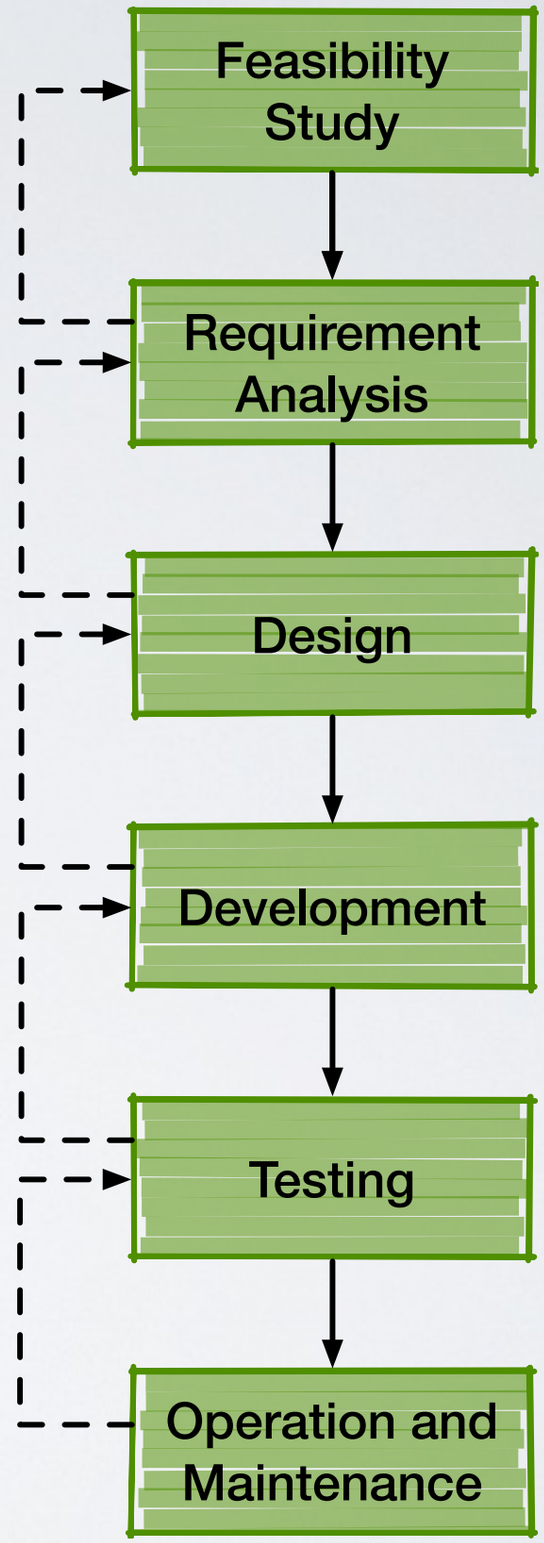
Information System Life Cycle

- Feasibility Study
- Requirement Analysis
- Design
  - data design
  - application design
- Development
- Testing
- Operation and Maintenance



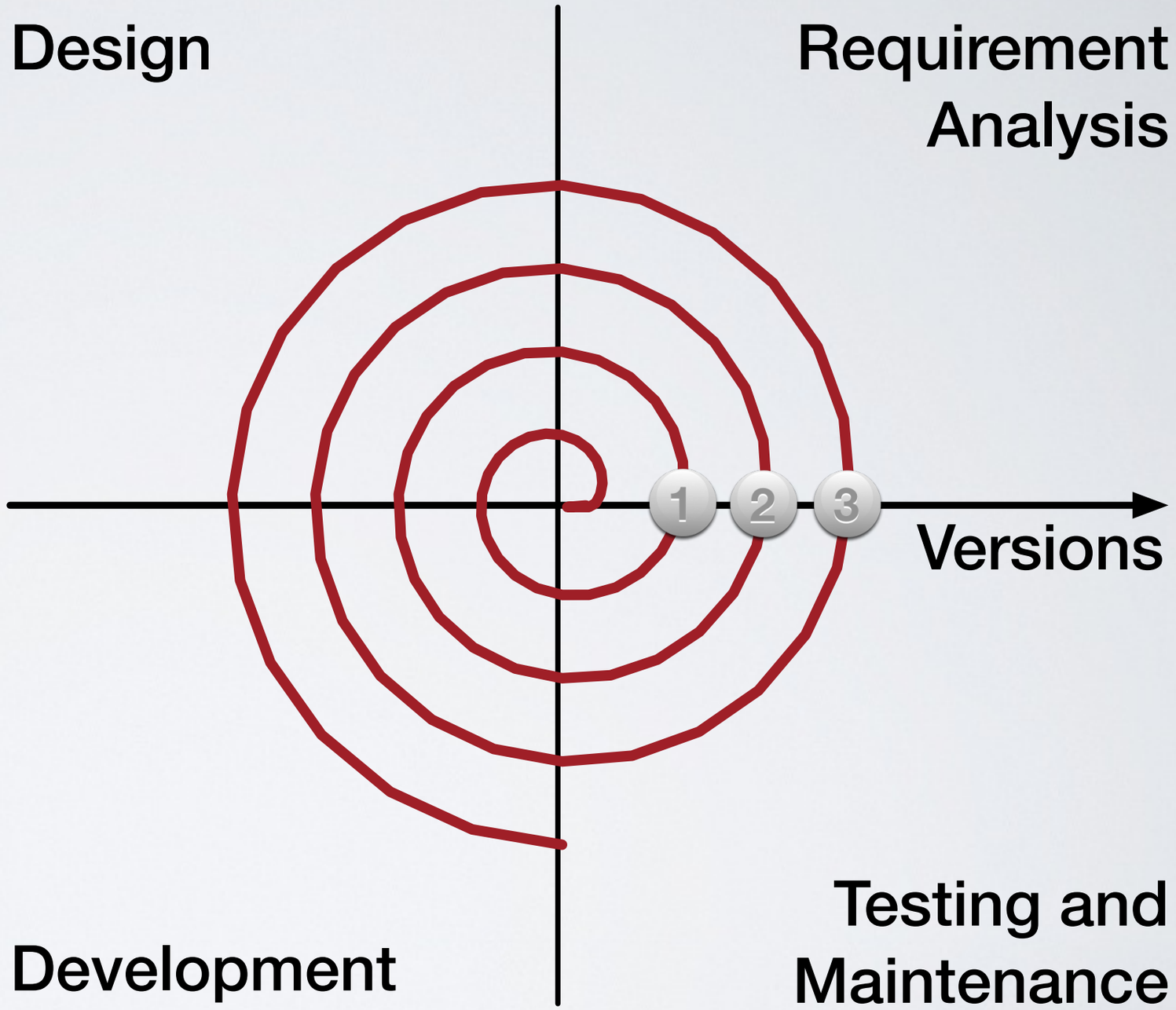
Information System Life Cycle

- Feasibility Study
- Requirement Analysis
- Design
  - data design
  - application design
- Development
- Testing
- Operation and Maintenance



Information System Life Cycle

- Feasibility Study
- Requirement Analysis
- Design
  - data design
  - application design
- Development
- Testing
- Operation and Maintenance



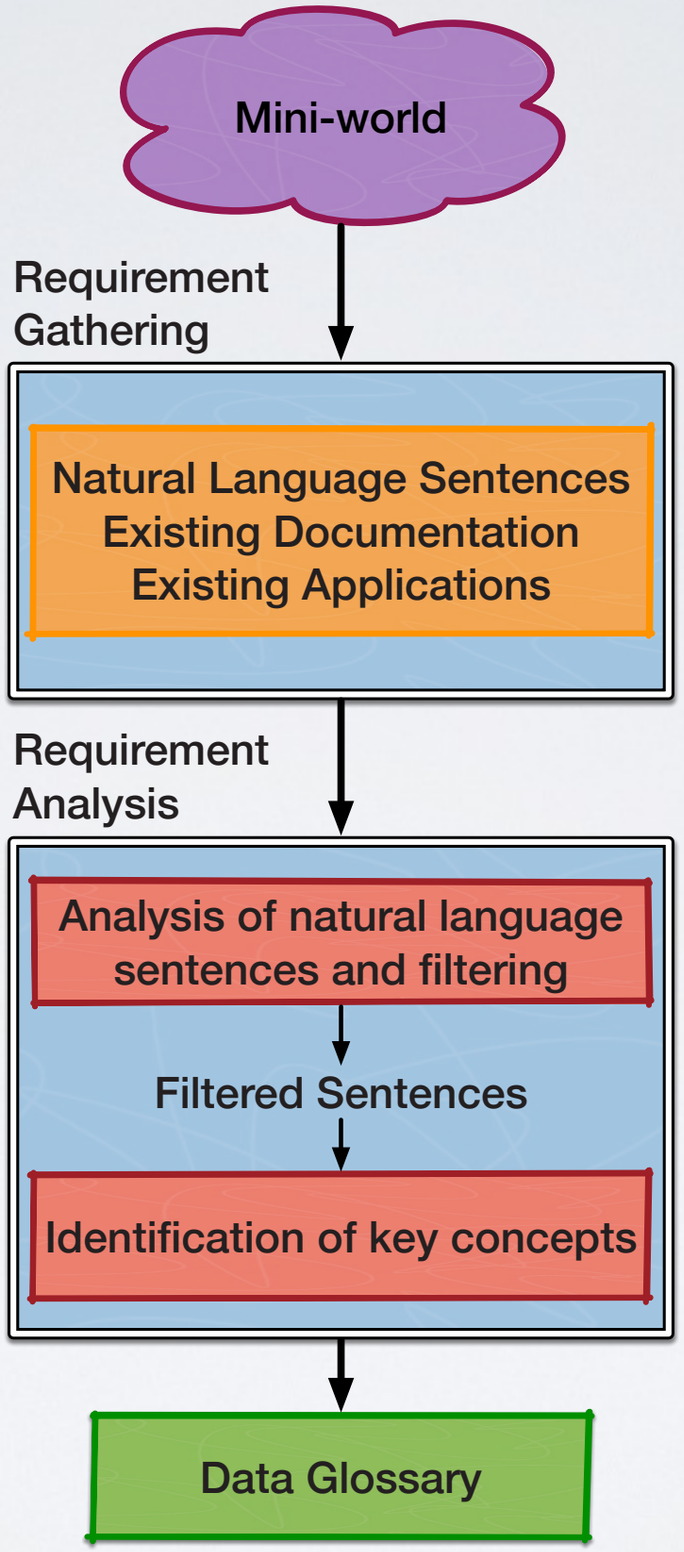
Requirement Analysis

Requirements

Requirements are the set of **features** that a **system** must **have** to **comply** with its **purpose**

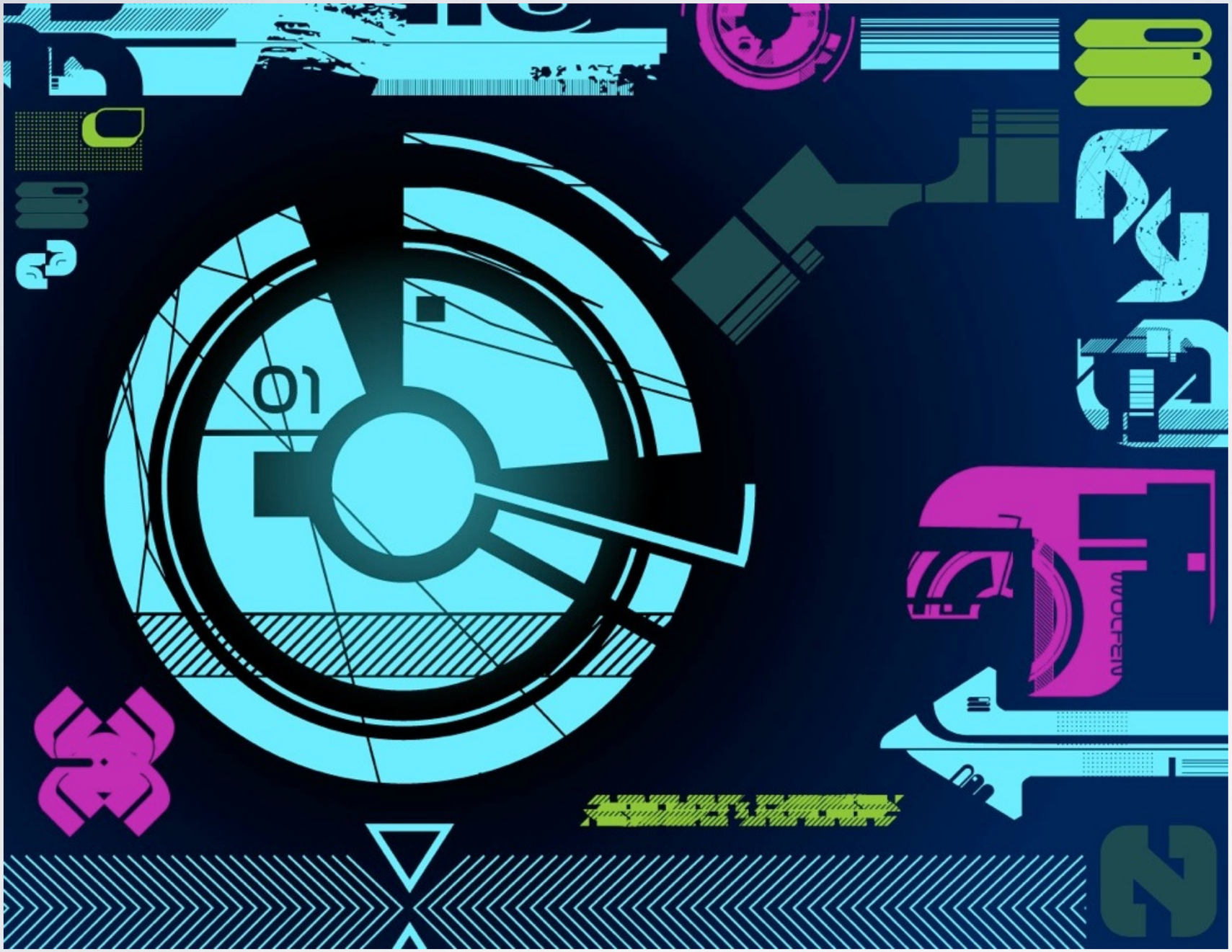
- Functional requirements
  - what the system must do, e.g. printing the invoice
- Non-functional requirements
  - the way in which a system must do something, e.g. printing in less than a minute
- Other constraints
  - general requirements set by stakeholder, often defined ahead, like using Linux

Requirements Gathering and Analysis

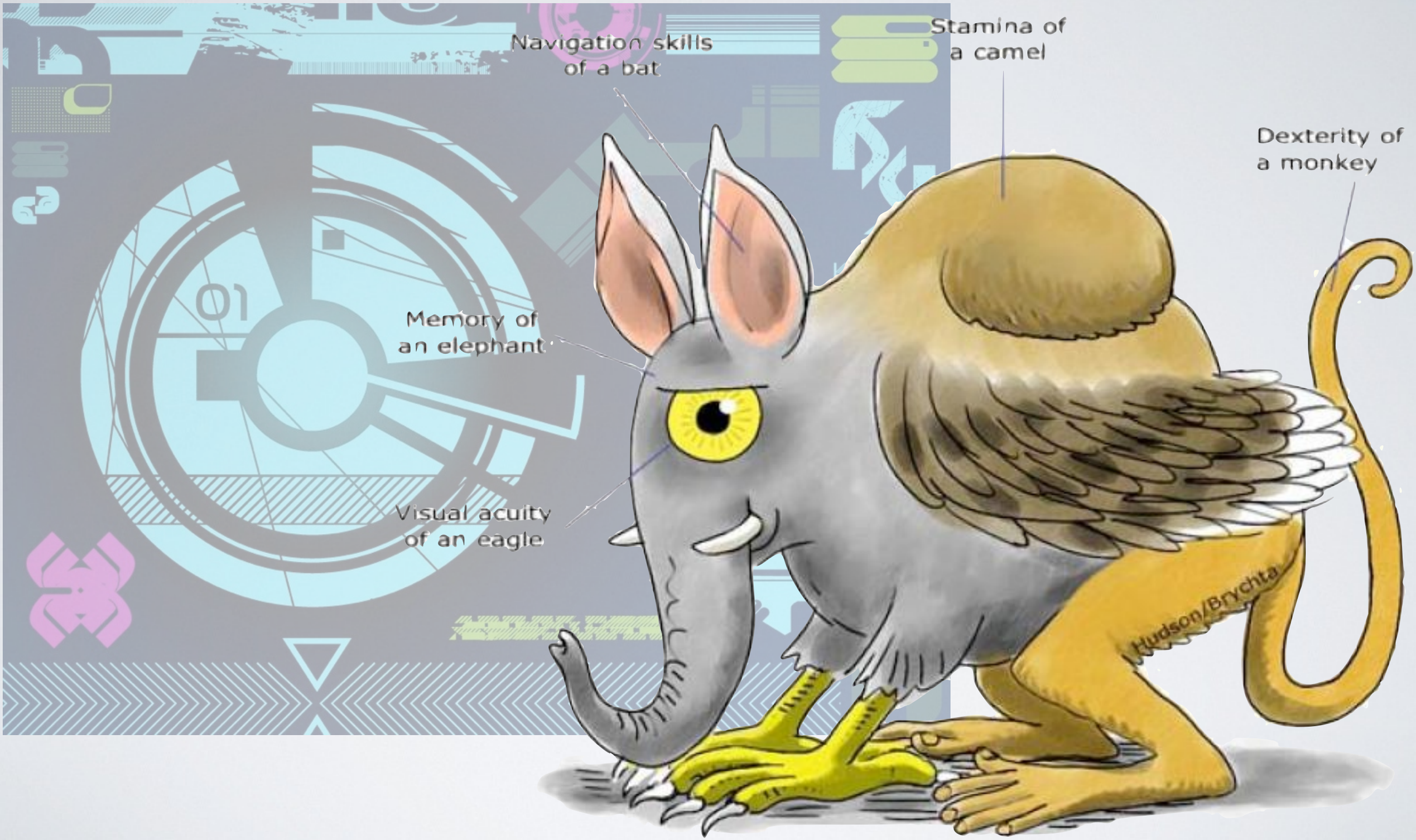


Users

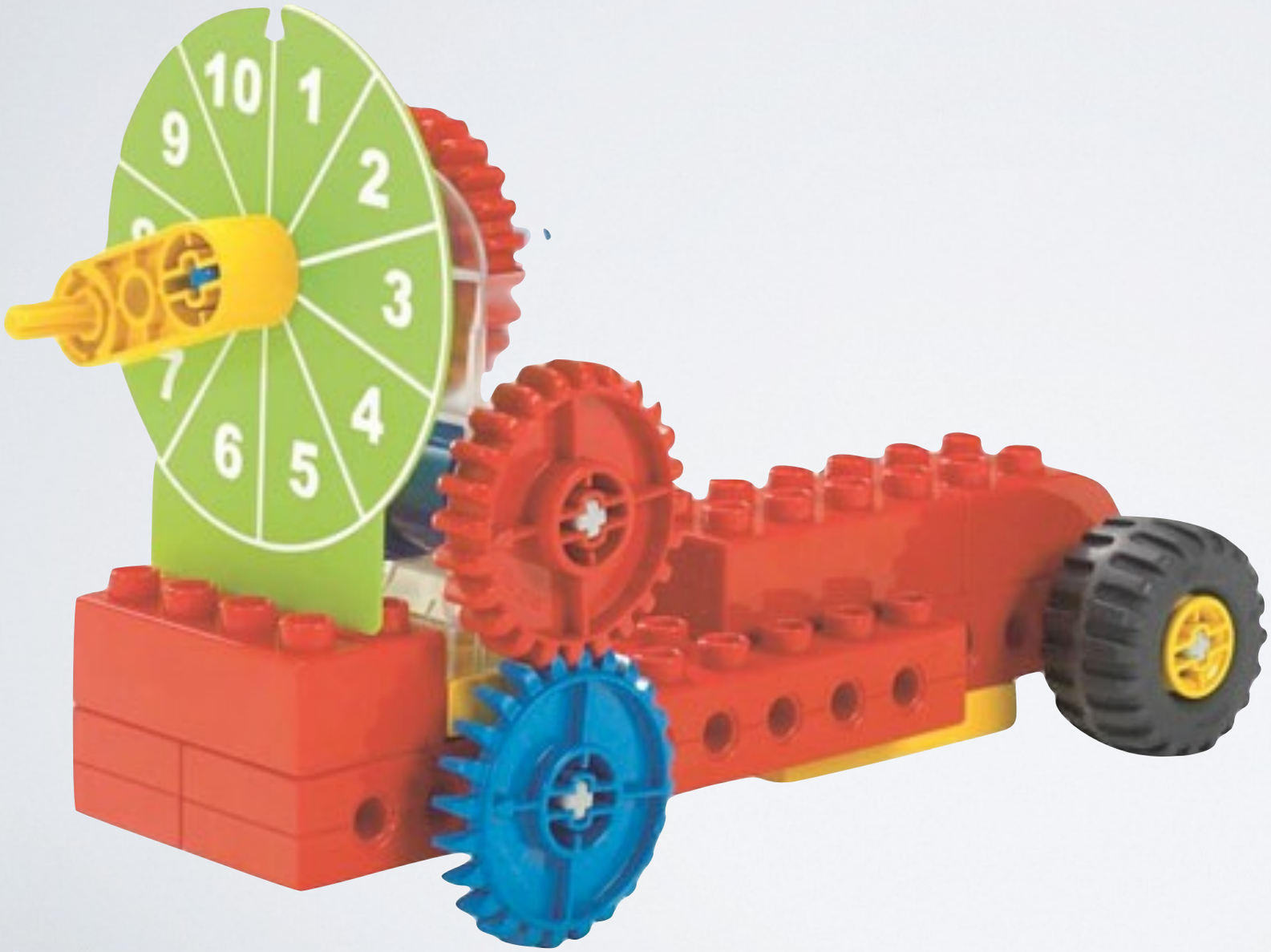
Users



Users


Navigation skills
of a bat
Stamina of
a camel
Dexterity of
a monkey
Memory of
an elephant
Visual acuity
of an eagle
Hudson/Brychta

Users



Users

Users

Users

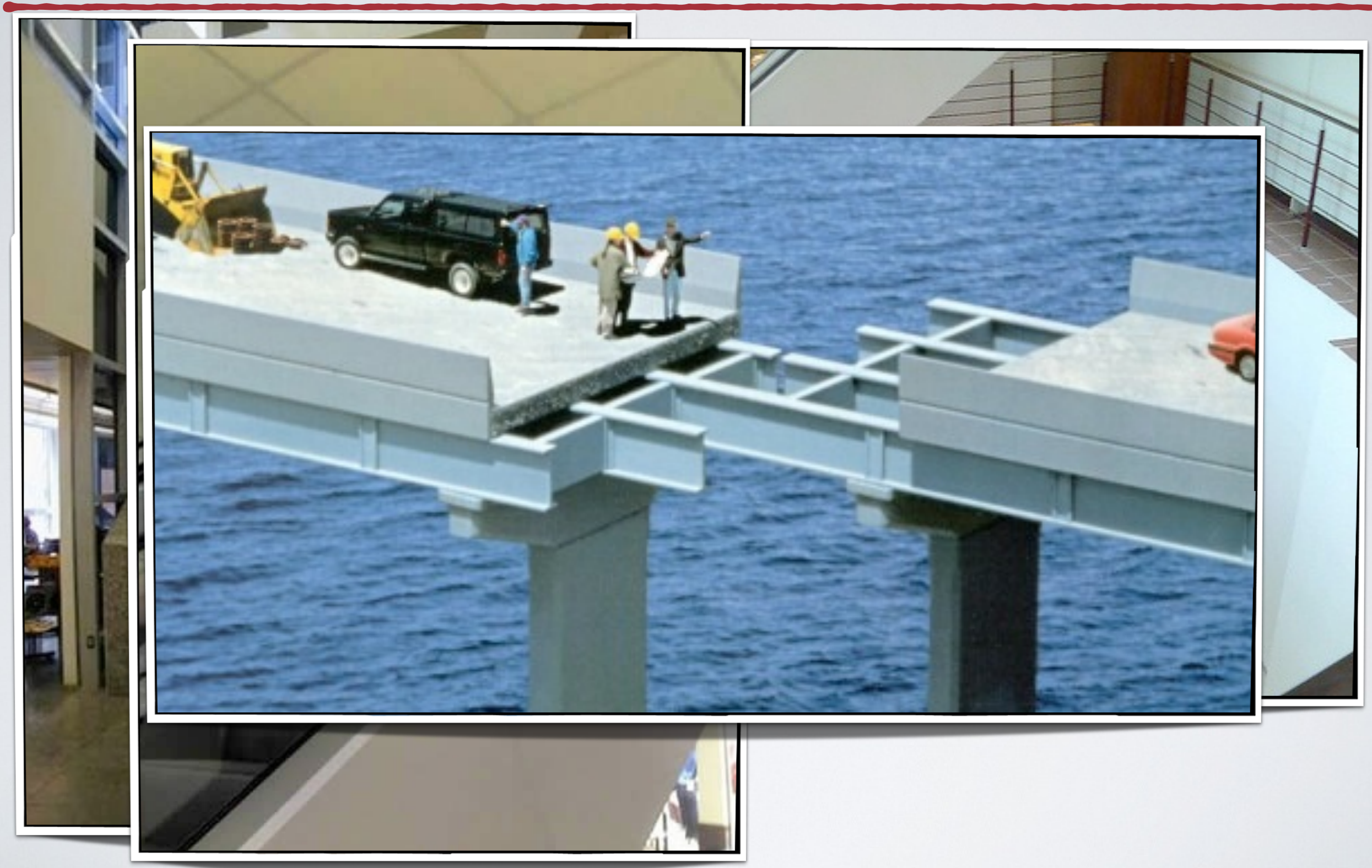
Errors, Errors, Errors

Errors, Errors, Errors

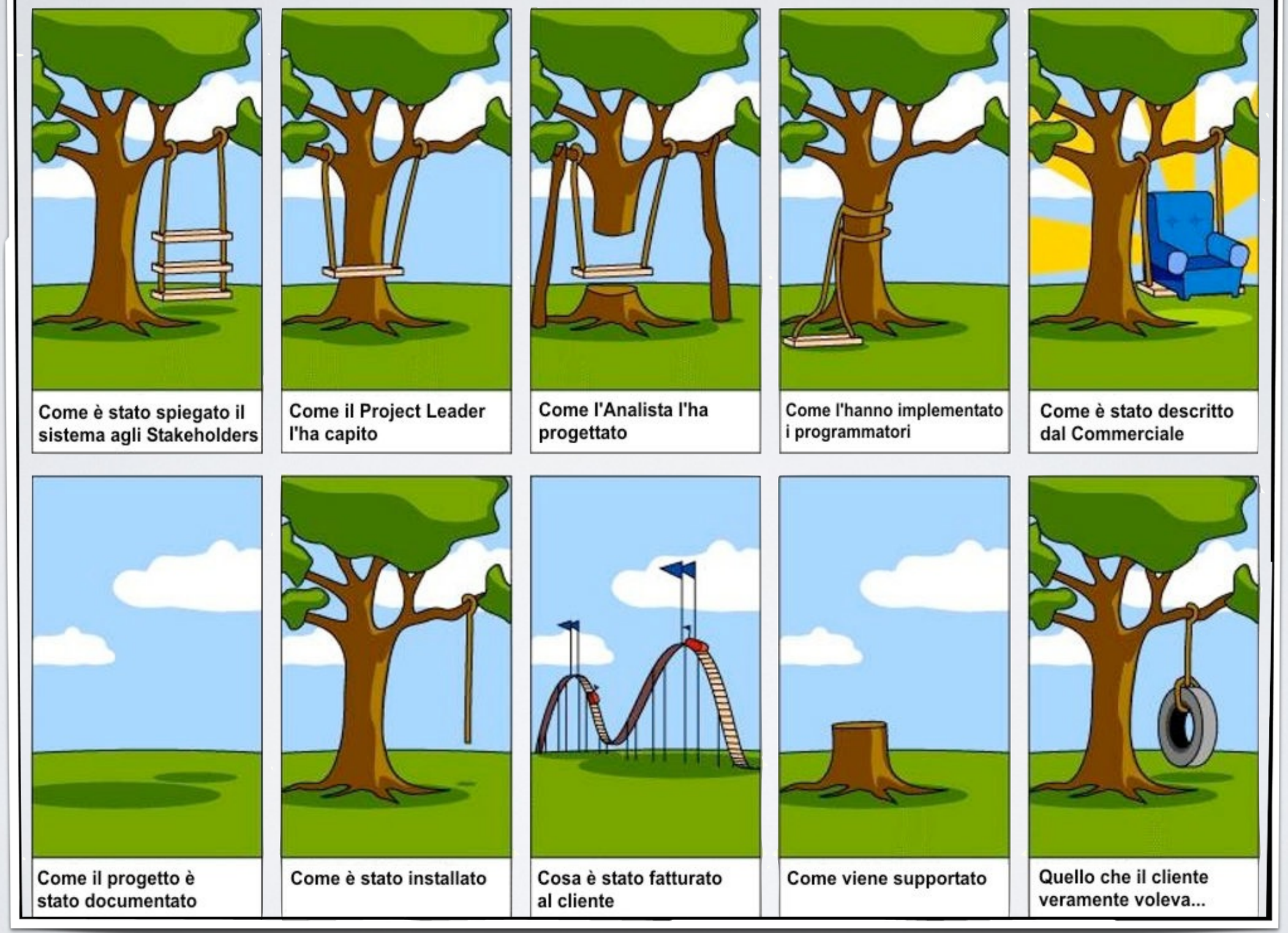
Errors, Errors, Errors

Errors, Errors, Errors

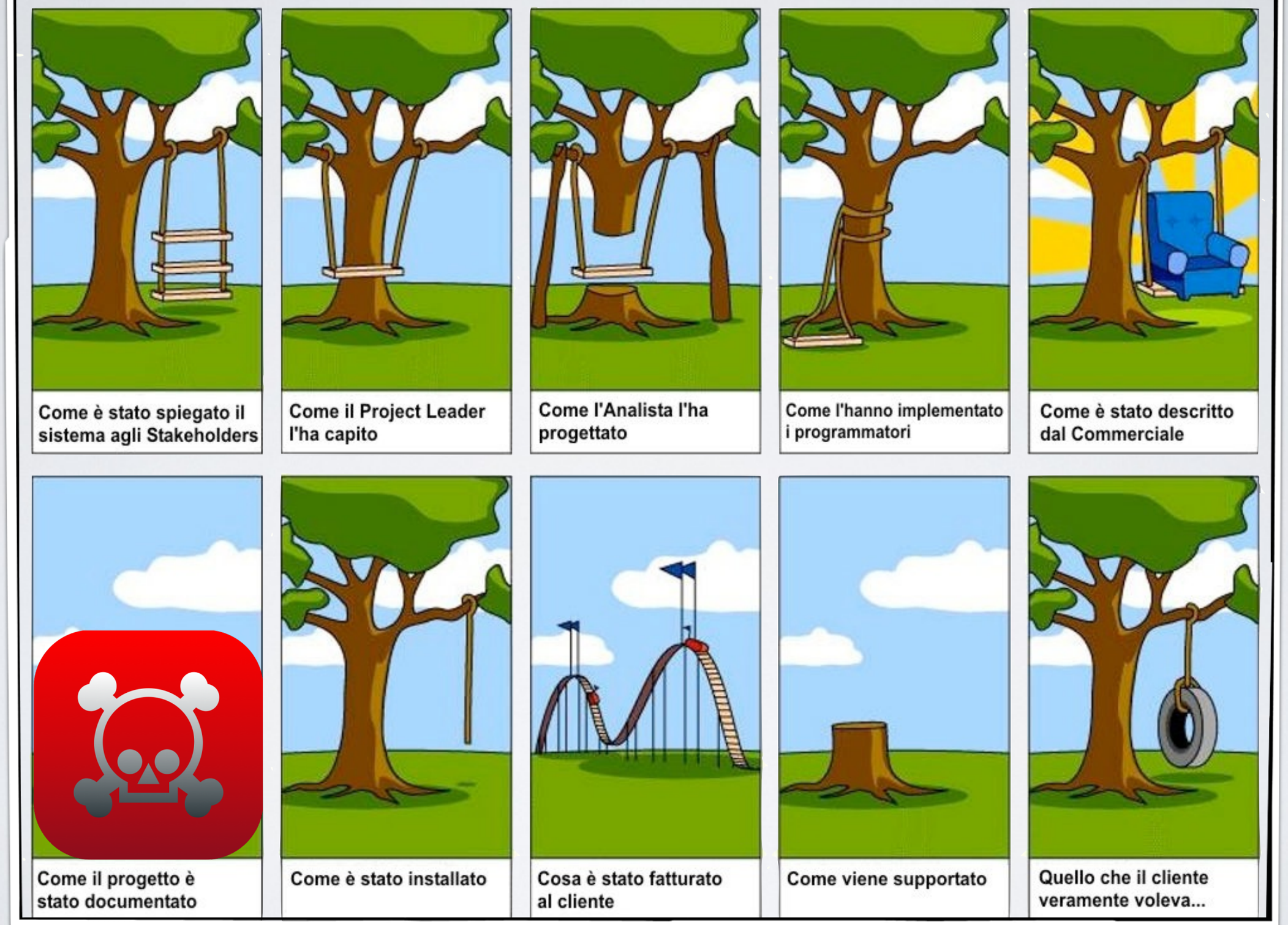
Errors, Errors, Errors



Bad Practices



Bad Practices



Database Design

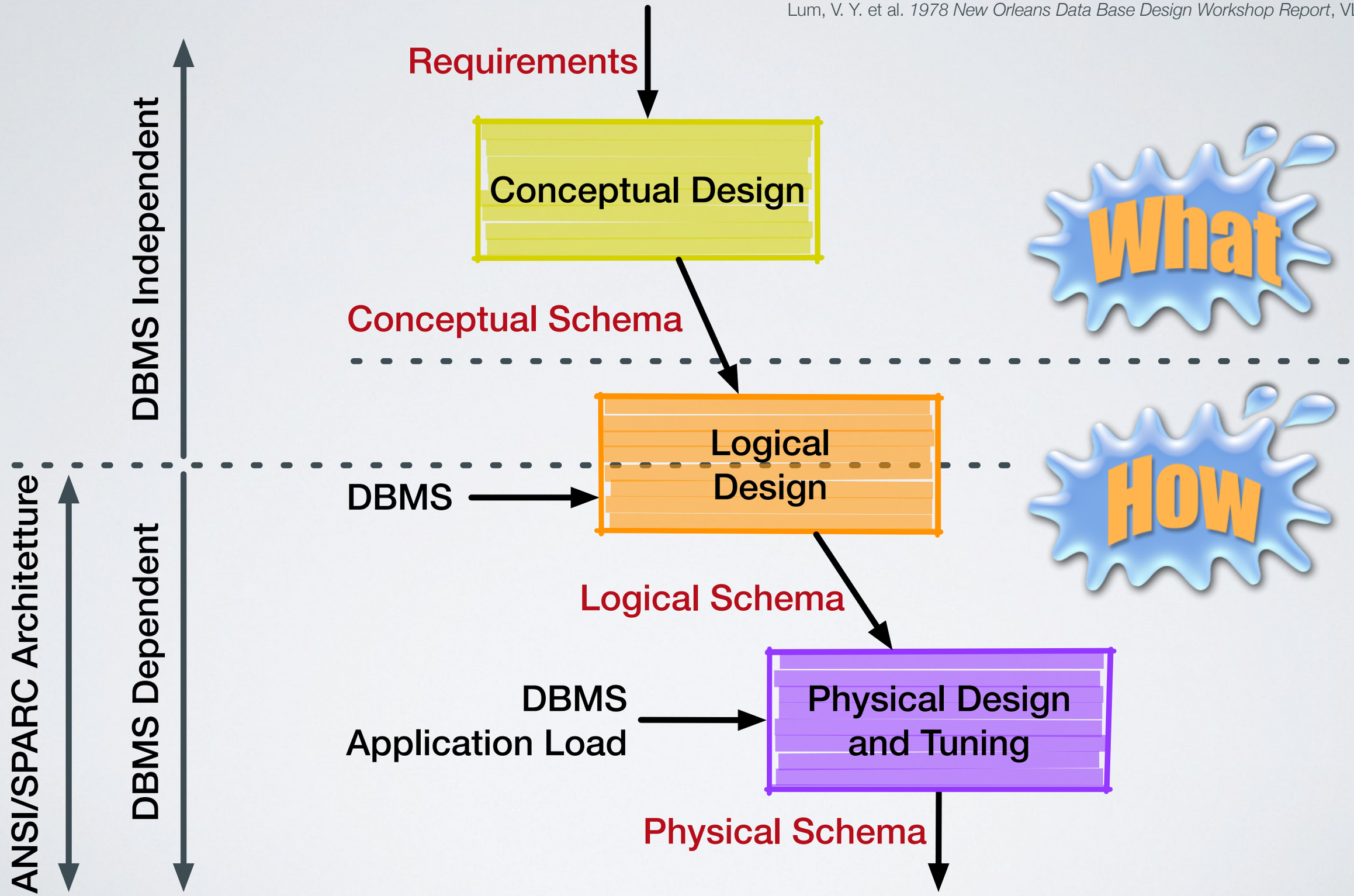
Goals of the Database Design

The **database design** aims at defining the **logical schema** and the **physical schema** (see the ANSI/SPARC architecture) of a database, according to the outcomes of the **requirement analysis**

- **Scenario I**: database design is part of a broader process, i.e. the design of the whole information system
- **Scenario II**: database design is a stand-alone process

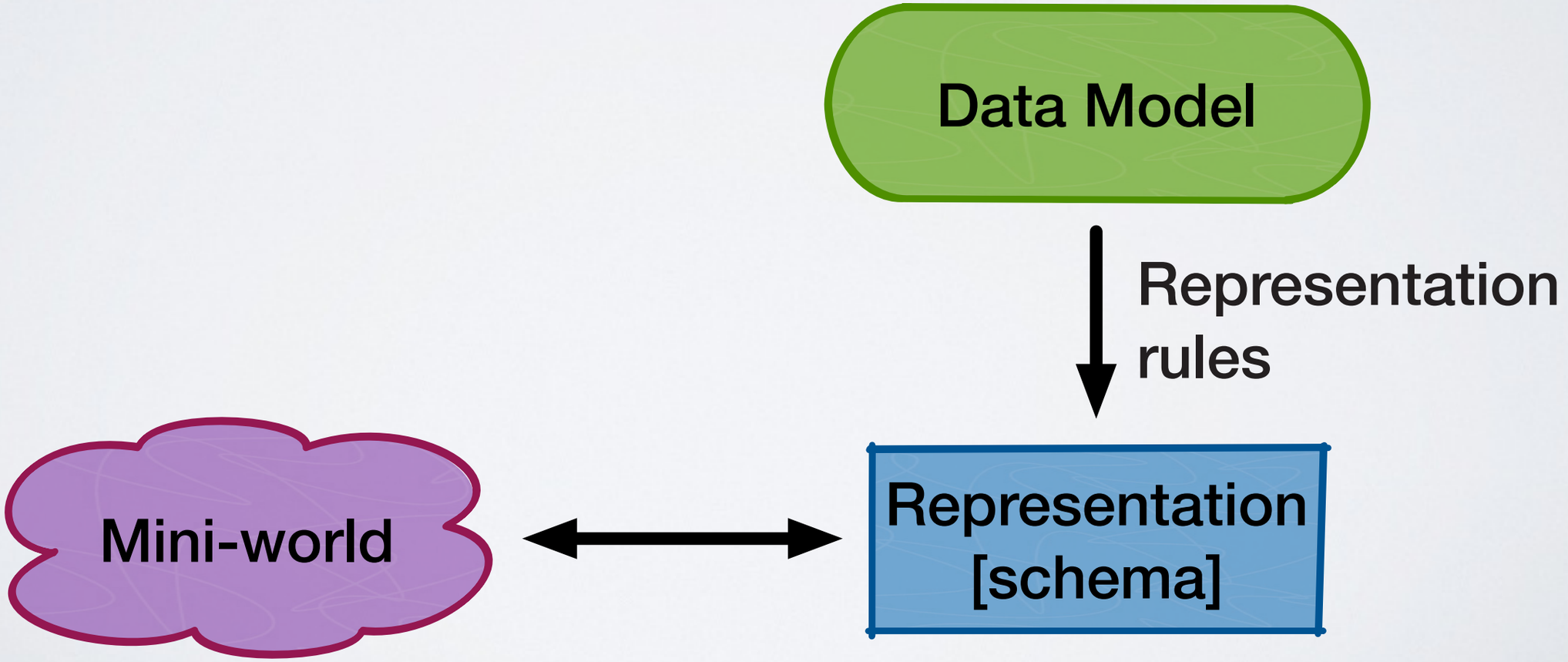
Database Design

Lum, V. Y. et al. *1978 New Orleans Data Base Design Workshop Report*, VLDB 1979



Data Model

A **(data) model** is a set of **symbolic structures** used to describe the **representation** of a **mini-world** of interest.
This representation is called **schema**



Schema and Instance

- A **schema** describes the structure of a database and represents the **intensional** level of a database

- An **instance** consists of the actual data and represents the **extensional** level of a database

Quality of a Schema

- **Completeness**: the schema must represent all the concepts (and their properties) relevant to the mini-world and identified in the requirements
- **Correctness**: the representation structures must be used properly and according to the prescribed semantics
- **Minimality**: each concept must be represented only once
  - you may still have duplications of concepts but they must be carefully motivated and documented
- **Readability**: the schema should be easy to read and self-explaining

Conceptual Design

Representation of the mini-world by means of a **high-level formal model**, integrating all the relevant concepts and **independent from the DBMS**

- **Input**: description of the mini-world resulting from the requirement analysis
- **Output**: conceptual schema plus additional constraints
- **Quality**: completeness; correctness; minimality; readability

Why the Conceptual Design?

- The designer reason to the right level of abstraction, independently from any DBMS and specific applications but focussing just on the mini-world to be represented
- The conceptual schema is understandable also by the stakeholders of the application which are actively involved in its design, reducing the risk of misunderstandings
- The conceptual schema is the most important source of documentation for the application and its subsequent modifications and extensions

Logical Design

Representation of the mini-world by means of **logical structures**, **independent** from **physical structures** and characteristic of a **class of DBMS**

- **Input:** conceptual schema; class of DBMS; estimated application load
- **Output:** logical schema plus additional constraints
- **Quality:** completeness; correctness; efficiency

Physical Design

Representation of the mini-world by means of **physical data structures** specific to a **given DBMS**

- **Input:** logical schema; a specific DBMS; estimated application load
- **Output:** physical schema plus tuning on the specific DBMS;
- **Quality:** correctness; efficiency

Wrap-up on Design

- Conceptual model → conceptual schema
  - Entity-Relationship (ER) model
- Logical model → logical schema
  - Relational model
- Logical model + physical parameters → physical schema
  - Structured Query Language (SQL) and its database-dependent extensions

Further Readings (1/2)

- Batini, C., De Petra, G., Lenzerini, M., and Santucci, G. (2002). *La progettazione concettuale dei dati*. Franco Angeli, Milano.
- Chen, P. P. (1976). The Entity–Relationship Model – Towards a Unified View of Data. *ACM Transactions on Database Systems (TODS)*, 1(1):9– 36.
- Codd, E. F. (1970). A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*, 12(6):377–387.
- Lum, V. Y., Ghosh, S. P., Schkolnick, M., Taylor, R. W., Jefferson, D., Su, S. Y. W., Fry, J. P., Teorey, T. J., Yao, B., Rund, D. S., Kahn, B., Navathe, S. B., Smith, D., Aguilar, L., Barr, W. J., and Jones, P. E. (1979). 1978 New Orleans Data Base Design Workshop Report. In Furtado, A. L. and Morgan, H. L., editors, *Proc. 5th International Conference on Very Large Data Bases (VLDB 1979)*, pages 328–339. IEEE Computer Society, Los Alamitos, CA, USA.

Further Readings (2/2, software engineering)

- Gezzi, C., Jazayeri, M., and Mandrioli, D. (2004). *Ingegneria del software. Fondamenti e principi*. Pearson Italia, Milano, 2nd edition.
- Pressman, R. S. (2010). *Software Engineering: A Practitioner's Approach*. McGraw-Hill, New York, USA, 7th edition.
- Sommerville, I. (2010). *Software Engineering*. Addison-Wesley, USA, 9th edition.

Questions?


WE INTERVIEWED HUNDREDS OF USERS AND TURNED ALL OF THEIR SUGGESTIONS INTO FEATURES.
DilbertCartoonist@gmail.com
Dilbert.com
AS IT TURNS OUT, EVERY USER WE TALKED TO WAS AN IDIOT, AND THEIR DUMB SUGGESTIONS RUINED OUR PRODUCT.
5-7-12 ©2012 Scott Adams, Inc./Dist. by Universal Uclick
IN HINDSIGHT, WE PROBABLY SHOULD HAVE TALKED TO PEOPLE WHO WORK OUTSIDE THIS BUILDING.