

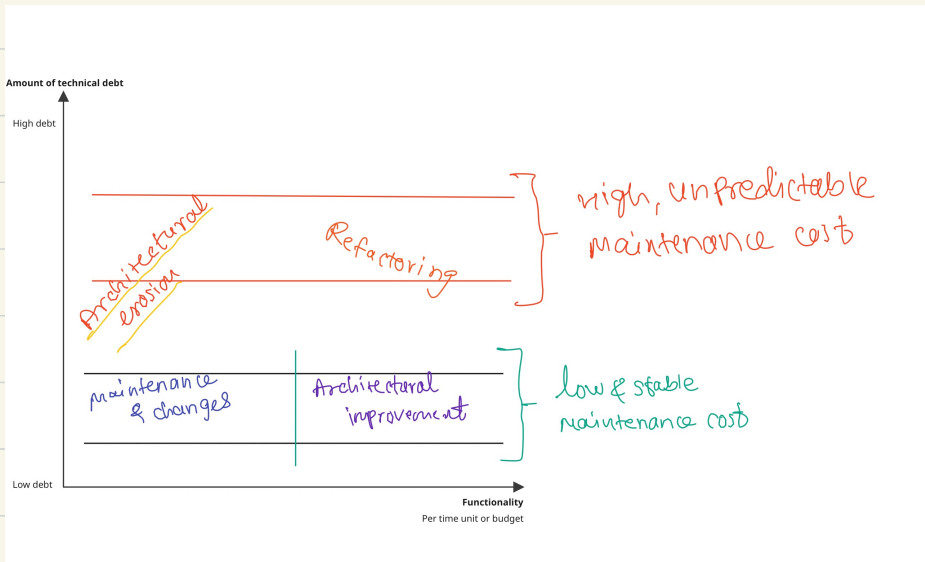
# Sustainability

= lowest possible maintenance & expansion costs

## Effect of Technical Debt

- Types of technical debt
- Architectural erosion
- increasing maintenance cost
- Refactoring

The system is a big ball of mud



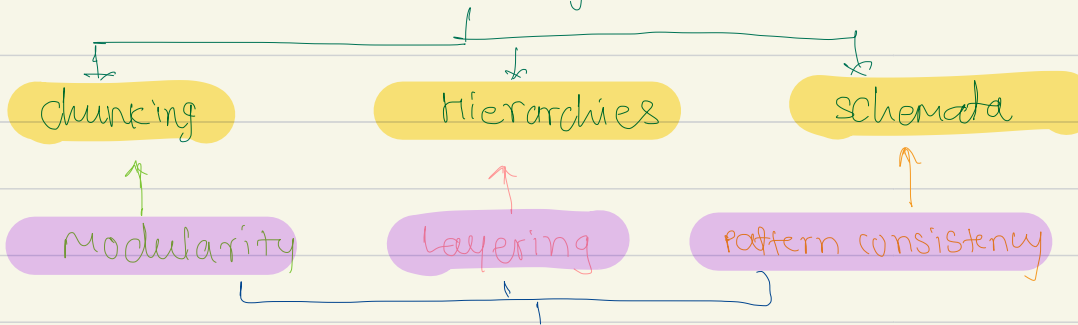
Refactoring is key

# Architecture & Cognitive Psychology

- ⇒ Understand code ?
- ⇒ Solve problem ?
- ⇒ Write code ?

Structured Architecture = Time savings

Structure-building Processes



Responsibilities

Coupling

Size ratio

Interfaces

Cycle freedom on

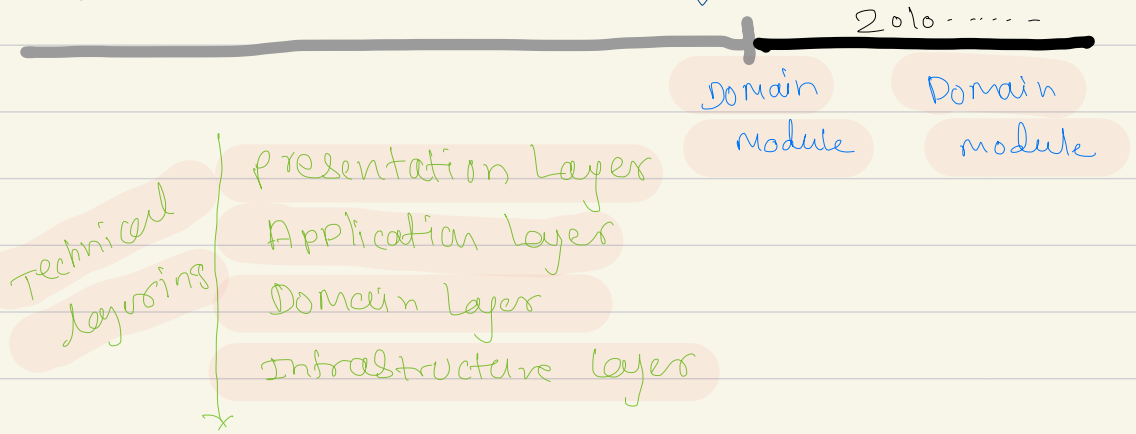
all levels

Uniformity

consistent

patterns

## Layering as an architectural style



- Sonograph tools

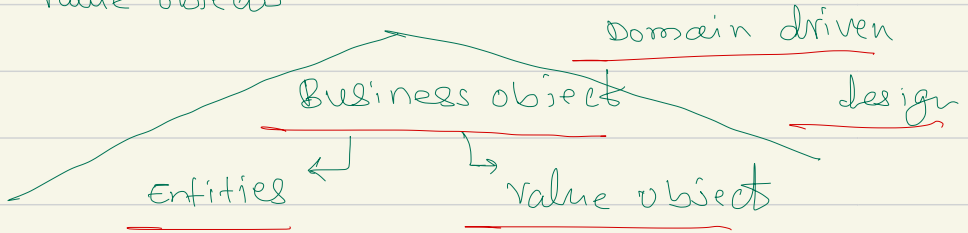
with modularity against technical debts

Modularity

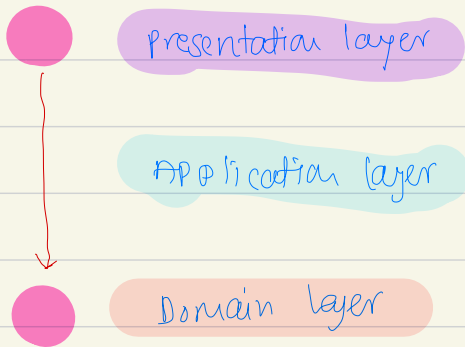
- high cohesion + loose coupling
- Separation of concerns
- Single responsibility principles

## Design patterns

factory pattern is mean to create entities & value objects.

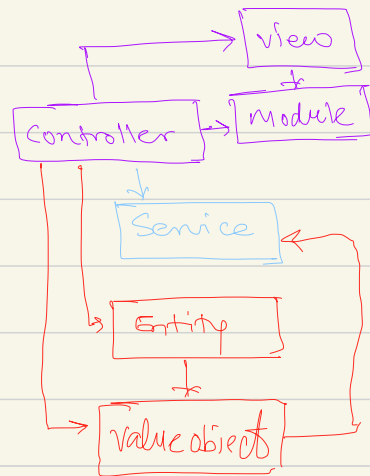


## Layering by patterns



## Layering by pattern

### Domain module



## Cycles create technical Debt

- Helper Methods are considered as utility,
- Helper is a very fuzzy design pattern & the name sounds so harmless.

## Architecture Reviews

### Status Determination

- comprehensive reviews or audits of
  - s/w systems, databases & interfaces
  - development processes
  - structure & processes of the IT organisation
- Broad search
  - interviews with involved stakeholders
  - metrics & analysis tools

# modularity maturity index (mmi)

