Software Design and Architecture

 Based on Requirements

Everything must be based on requirement of stakeholder – business requirement

Architecture is forced on non-functional requirement

Translate business requirement to the right architectural design decision

If stakeholder values maintainability 🡪 Layer pattern

If stakeholder values performance 🡪 Layer pattern won’t be a good solution

Design or architectural pattern expresses a relation between a problem and a solution.

### Rationale, Rationale, Rationale

It is important for reader to understand the reason why I made a specific decision

Make your assumptions explicit and add them to the description

### 3. Don’t Repeat Yourself (DRY)

Duplication is evil, we shall not repeat yourself

### 4. Slice the Cake

### it is called in **Agile development** the best description for being Slice the cake.

Design will iterative in vertical slices

Prototype each slice to confirm it works

### 5. Prototype

We prototype the design

Validate our assumption

Gives the quick fail which is good

### Quantify

1. Measure the size of each reused part
2. Measure the size of the entire software product
3. Calculate the ratio of reused parts, which is the sum of reused parts divided by (2

### Focus on the Boundaries and Interfaces

Focus on boundaries and interfaces

Separate the concerns

### The Perfect is the Enemy of the Good

Nothing is perfect, there is always something forgotten