* Architectural Model

Architectural Model is an artifact that captures some or all of the design decisions made by the model

It is presenting abstract idea and documentation of those design decisions.

* Ambiguity

A model is ambiguous if it is open to more than one interpretation

* Accuracy

A model is accurate if it is correct, conforms to fact

* Precision

A model is precise if it is sharply exact

# Viewpoints

Logical Viewpoint

Physical Viewpoint

Deployment Viewpoint

Concurrency Viewpoint

Behavioural Viewpoint

Consistency in viewpoints →

contain overlapping design decisions ∴ scope for inconsistency

1. Direct Inconsistency →
2. Refinement inconsistency →
3. functional vs nonfunctional
4. Static vs dynamic
5. Dynamic vs dynamic

Dynamic means those components that can be changed
Static cannot be changed

# Architectural Analysis Goals

- Completeness
- Consistency (across views)
  - Name
  - Interface (return types)
  - Behaviours
  - Interaction
  - Refinement (higher & lower level models)
- Compatability (across literature)
  - style
  - reference architecture
  - architectural standards
- Correctness

**RIBIN**

# Architectural Concerns

Structural

Behavioural

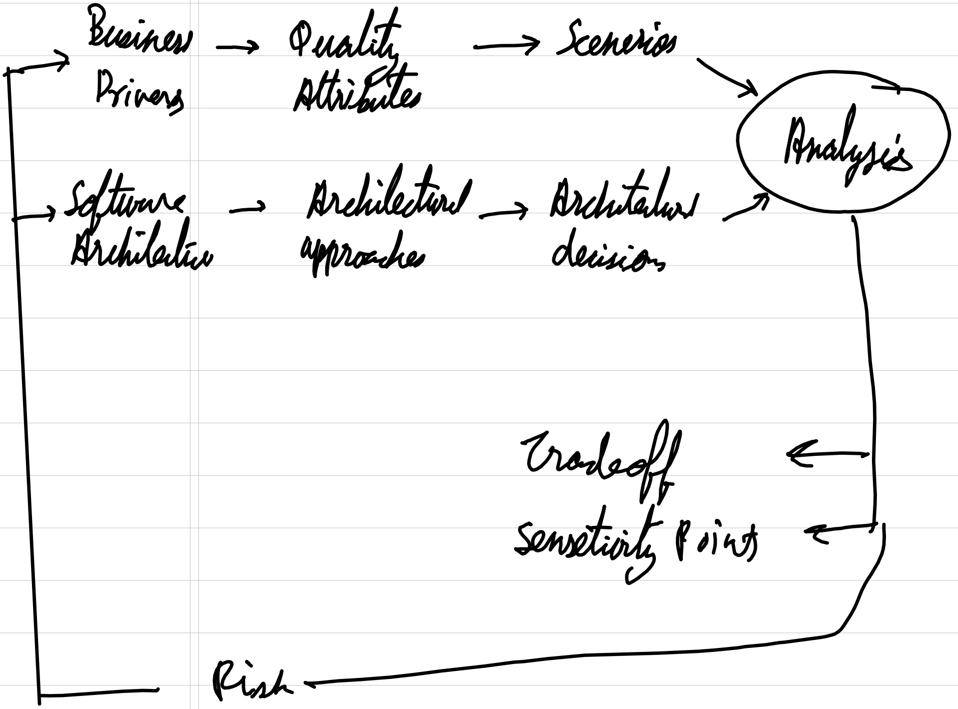
Interaction

Non-functional

# ATAM analysis

Modificability
Security
Performance
Reliability

Quality Attributes



# ATAM

1. Who are stakeholders
2. what are business goals
3. what is critical functionality?

} Business Drivers

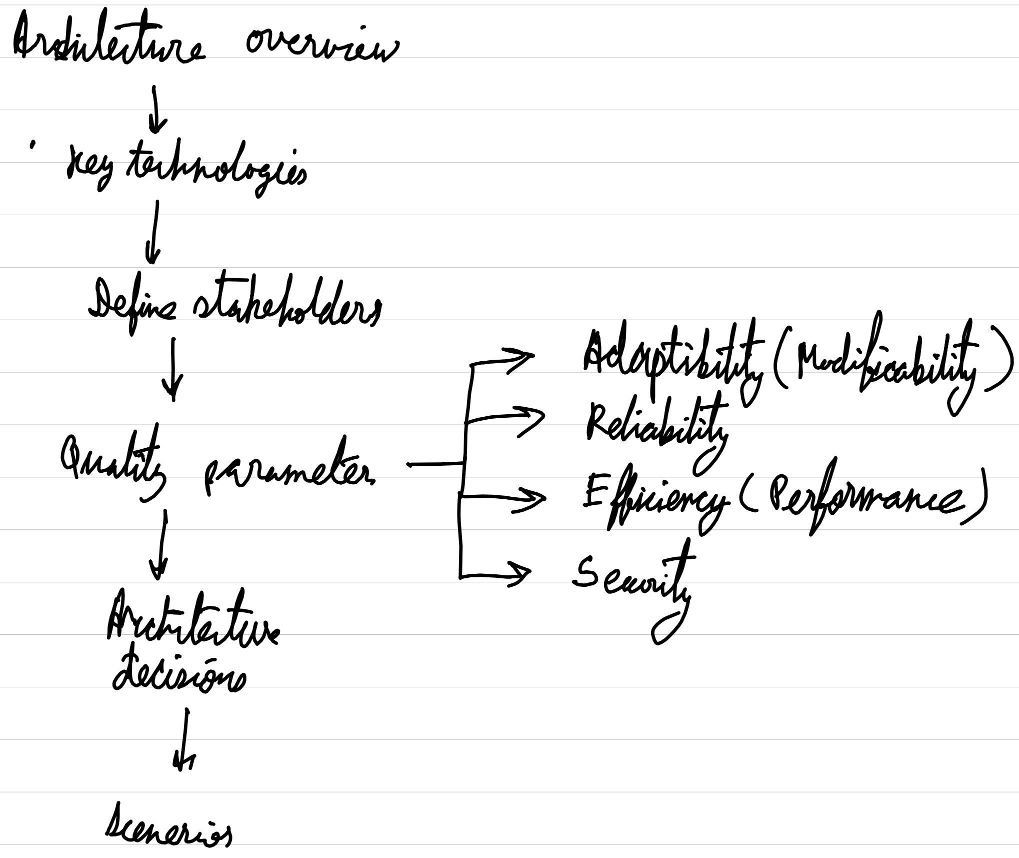
what are different use case?
(20 - 20k) scale
(Black friday Sale)

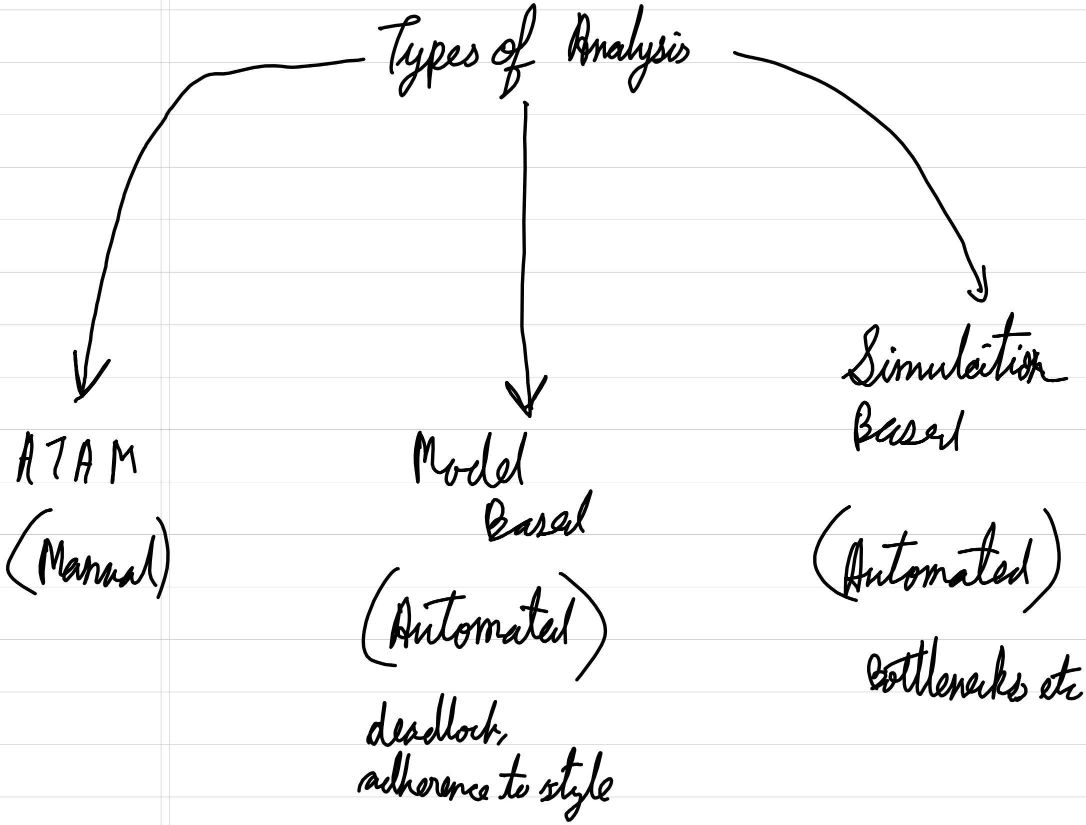
1. Use case scenerios
2. Growth scenerios
3. Exploratory scenerios

} Scenerio analysis

1. Software architecture
2. Architectural decisions

} → find risk


Architecture overview
↓
key technologies
↓
Define stakeholders
↓
Quality parameter → Adaptibility (Modificability), Reliability, Efficiency (Performance), Security
↓
Architecture decisions
↓
Scenerios


Types of Analysis
ATAM
(Manual)
Model Based
(Automated)
deadlock,
adherence to style
Simulation Based
(Automated)
Bottlenecks etc

# Designing Non functional Property

S - Scalability
C - Complexity
A - Adaptivity
R - Reliability
E - Efficiency
S - Security

① Scalability → How much the system can scale
10 users vs 10k users

eg. saving database in memory vs in disk

Distribute data over various parallel processing. Load Balancing, remove Bottleneck

② Complexity → Less complex, the better
Keep component interfaces simple and compact

Complexity increases scope of error, architectural eroson

Seperate the processing from the communication (C2 pattern)

③ Adaptibility → How the system can respond to change, satisfy new requirement

System must be modular, independant components.

Changes in one component should not affect others.

eg layered architecture is less adaptable than event based.

Seperate the concerns into independant modules

Portability must also be taken of
Make connectors flexible

④ Reliability → How reliable the system is
fault tolerance
eg RAID

The availability must be present with minimum downtime

Robustness → unexpected runtime conditions.

Fail gracefully to errors.
Avoid single point of failure
Keep backups.

⑤ Efficiency → Meet performance requirement vs resource usage economy

eg. model runs on CPU vs ~~on~~ GPU

Avoid data transmission wastage. Use the data broadcast carefully.

⑥ Security → The system must be secure to attacks

Intended and Unintended damages.

Seperate levels of privaloges