Risk Management

A Risk is a "uncertain event or condition that, if it occurs, has a positive or negative effect on a project's object.

- A risk management plan is a document that a project manager prepares to foresee risks, estimate impacts and define the responses to risks.

Risk Management

Reactive Proactive

Reactive Risk management tries to reduce the damage of potential threats and speed an organization's recovery from them, but assumes that those threats will happen eventually.

Proactive Risk Management identifies threats and aims to prevent those events from ever happening in the first place.

Types of Risk -

- Business Risk : Building a product that no one wants or Loosing budgetary commitments.

- Technical Risk: Concerned with quality, design, implementation, interface, maintainence problem.

Project Risk: Concerned with schedule, cost, resources, customer-related issue.

Risk Assessment:

- IT is a process to rank the risks in terms of their damage.
- Determine the average probability of occurence value of each risk.
- Determine the impact for each component based on impact assessment metrix. (severity val).
- Risk Assessment values are determined by multiplying the score for the probability and severity values together.

Impact Assessment Matrix

Probability	Negligible: 1	Marginal: 2	Critical: 3	Catastrophic
Frequent: 5	Med 5	High-10	Very high-15	Very-high-20
Likely: 4				
Occassional: 3				
Unlikely: 2				
Rare: 1				

Risk Control

Risk Control is basically the specific actions to reduce a risk event's probability of happening.
eg: Correct inspection & maintainence of the car reduce the likelihood of mechanical failure.

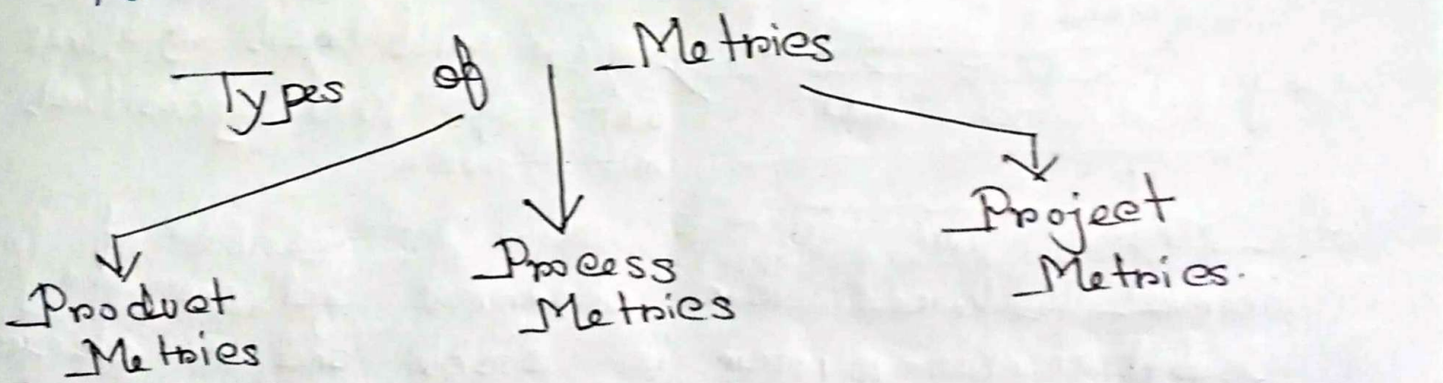
Risk Mitigation

Risk Mitigation is the set of actions to reduce the impact of a Risk Event.
eg: Airbags are used to reduce the impact of an accident on the passenger and drivers.

Risk Mitigation & Control Strategy

- Risk Avoidance.
- Risk Transfer.
- Risk Reduction.
- Risk Monitoring.

Verification	Validation
1) Are you building it right?	2) Have you built the right things?
2) check whether an artefact conforms its previous artefact.	2) check the final product against specification.
3) Done by developer.	3) Done by tester.
4) Concerned with phase containment of errors.	4) Aim is to make final product error free.
5) Methods involves Review, Inspection, Unit testing and integration testing.	5) Involves System testing.
6) Static and Dynamic Activites.	6) Only dynamic.



Reliability: Probability of failure free operation for a given time duration.

- It's an execution event where the S/W behave unexpectedly.

Failure of S/W depends on two factors-
(i) No. of faults being evaluated in S/W
(ii) operational profile of execution of S/W.

Types of Time
(i) Execution Time
(ii) Calendar Time
(iii) Clock Time.

Reliability Metrics

(1) Probability of failure on Demand-
It is a measure of the likelihood that the system will behave in an unexpected way when some demand is made on it.
eg. Safety - Critical System.

(2) Rate of occurence of Failure :— (ROCOF)
A measure of frequency of occurence with which unexpected behaviour is likely to be observed.
eg. ROCF = $\frac{2}{100}$ → S/W will fail 2 times out of 100 operational unit times.

(3) Mean Time to Failure (MTTF) - A measure of time interval between observed failures.
Useful when system is stable and no changes are being made to it.
↳ indication of how long the system will be operational before failure occurence.

(4) Availability :- Measure of how Likely the system is to be available to use. = $\left(\frac{MTTF}{MTTF+MTTR}\right)^{x100}$ %

MTBF = MTTF + MTTR
Bet^n Failure — To failure — To Response.

Structured Programming

- Top-Down Analysis
- Modular Programming
- Structured Coding

Top-Down Analysis: Problem is broken down into small pieces each one has its own significance. Each problem is individually solved and steps are clearly stated about how to solve the program.

Modular Program: Code is broken down into modules, subprograms or sub-routines.

- based on understanding of top-down Analysis.
- unconditional goto is not preferred as it makes the program flow non-traceable.

Structured Coding:

Structured English

Analyst and designer of the software come up with tools such as structured english.

- Helps to mitigate understanding gap.
- It is nothing but the description of what is required to code.

Like IF-THEN-ELSE,
DO-WHILE-UNTIL.

Decision Table

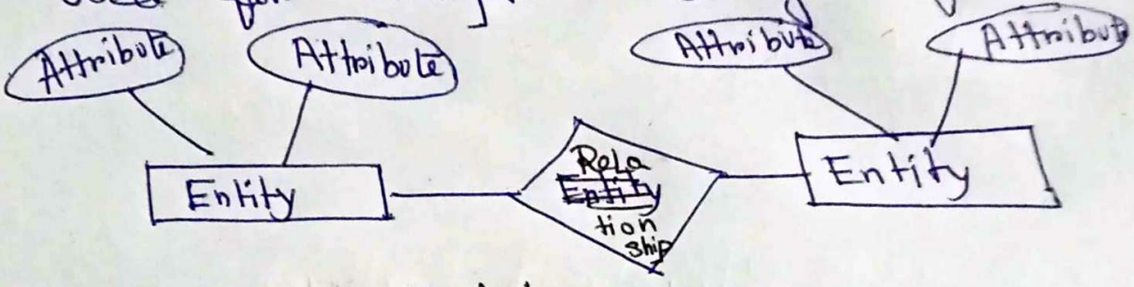
To create the decision table developers must follow basic four steps :—

- Identify all possible condition to be addressed.
- Determine actions for all possible cond^n.
- Create maximum possible rules.
- Define action for each rule.

Conditions / Actions	Rules							
Conditions								
Shocos connected	N	N	N	y	y	y	N	y
Ping is working	N	N	y	y	y	N	y	N
opens website	N	y	y	y	N	N	N	y
Actions								
check Internet Router		X				X		X
Restart web brows					X	X		
Contact ISP	X	X	X				X	

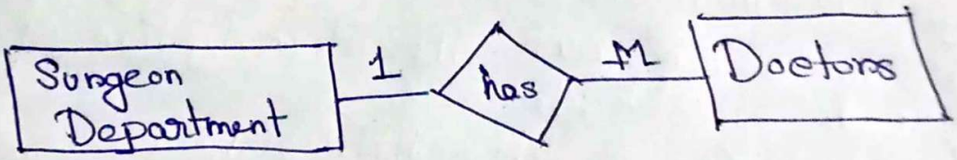
E-R Diagram

Used for conceptual design of database.

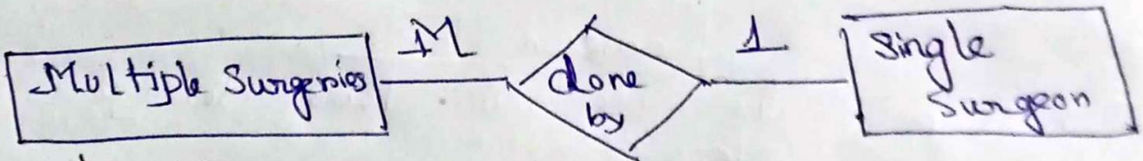


Mapping Cardinality

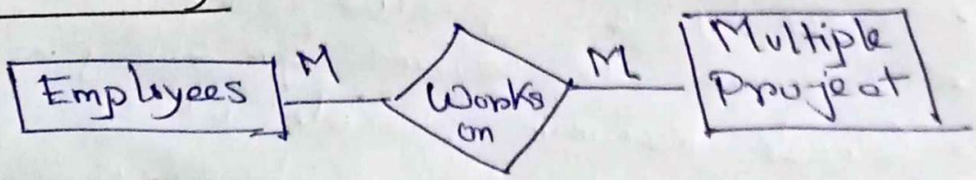
(i) ONE-TO-MANY



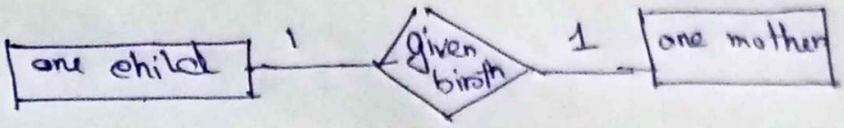
(ii) Many to One



(iii) Many to Many -



(iv) One to One -

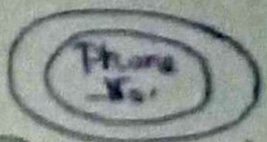


Types of Attributes

1) Key Attributes – which uniquely identifies each entity in

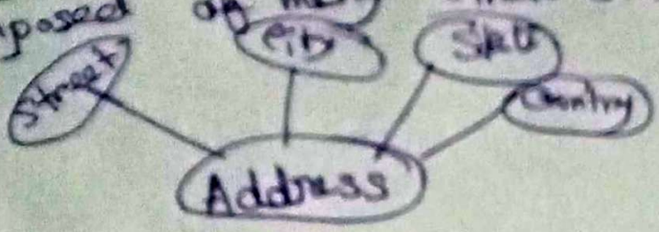
2) Multivalued Attribute

consisting of more than one value



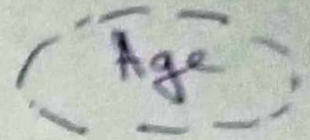
3) Composite Attribute

composed of many other attributes



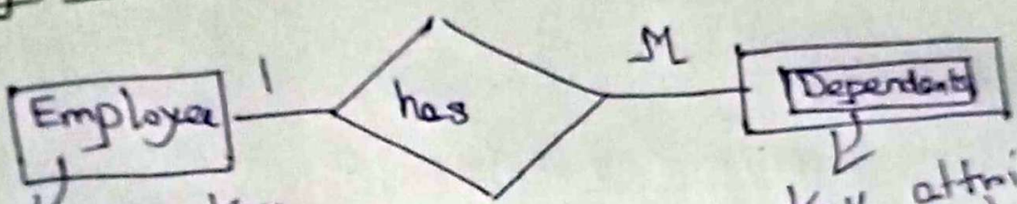
4) Derived Attribute

can be derived from other attribute



Types of Entity

Strong Entity & Week Entity

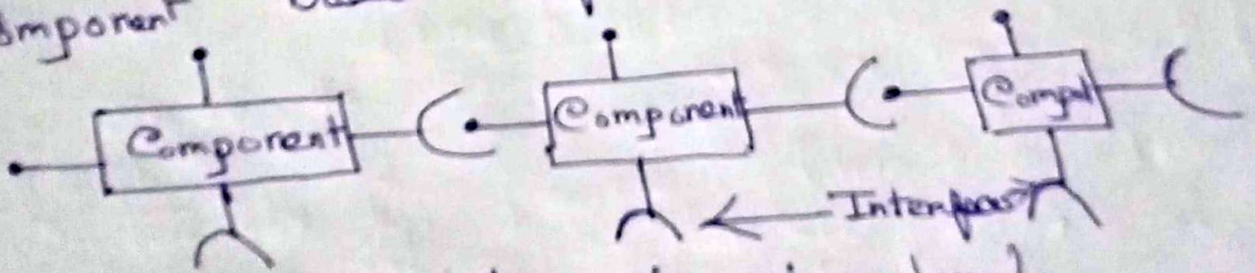


Has primary key attribute.

Key attribute doesn't exist.

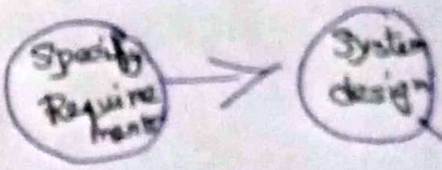
Reuse

There is a whole new vertical which is based on re-use of software component-, is known as Component Based Software Engineering (CBSE)

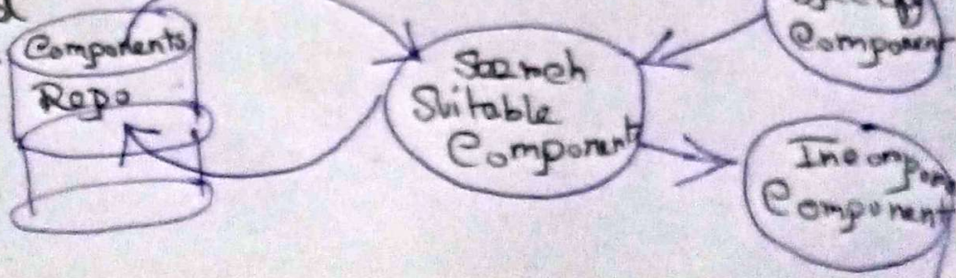


Re-use can be done at various level

- Application level
- Component level
- Modules level



- Keeping req same and Adjusting component
- Keeping comp same & modifying req



Quality Assurance

- Standards
- Reviews & Audits
- Risk Management
- Testing
- Error/defect collection & Analysis
- Change Management
- Education
- Security Management