Lecture 18:- 03-05-18

1. Usibility:-

- Interface (NON arch)
- User friendly layout (NON arch)
- Response time
- Screen resolution.

2. Performance:-

- response time
- Efficiency
- Requirements fullfilled
- user ~~adaptiveness~~ expectness.
- Predictibility.

3. Availability:-

- if application crashed/down then in how much time it will recover.
- Faults
- 24/7
- Network down

Quality Attributes

these senario generate availability

Source of stimulus : from where the error occured. jahan se generate hua can be external or internal.

Stimulus: e.g. response delay. jo occure hua hai.

Environment: system condition
کسی ماحول میں اپنی ایپلی کیشن کو چلانا ہے

Archifact: part or entire system reason why our system has error
human error bi system error hai.

Response:
fault identify, notify the user and ~~manag~~ manage that is response.

Response Measure:

Source → Stimulus → Environment

internal
External
(user)

Fault
crash
omission
Timing response

Normal
degrad

Artifact

process
storage
processor
communication

Response

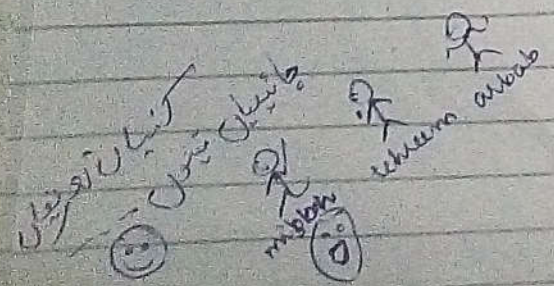
Record, notify
Disable, Continue

Response measure

Repaire time } (fixing time)
+ availability
available
+ { degrad time interval }

(down and then up your website this time is called degrad time Interval.)

1 - available دوبارہ



08-05-18

Lecture 19:-

System Attributes (architectural)

- Security
- Usability
- Availability
- performance

Usability:-

→ Interface
→ Response time
→ Interaction b/w user & interface

Availability:-

→ crash and then repair and degrad time is availability.
→ how much time application is available.
* in degrad time most complex things are happening here.
* Normal :- Successful execution

Performance:-

→ Response time
→ Predictibility
→ efficiency
→ how functionality is divided. (Architectural)
→ communication/new component (Arch)
→ Algorithm selection (loops, n (NON) complexity ↓, reduced)
→ Algorithm Implementation

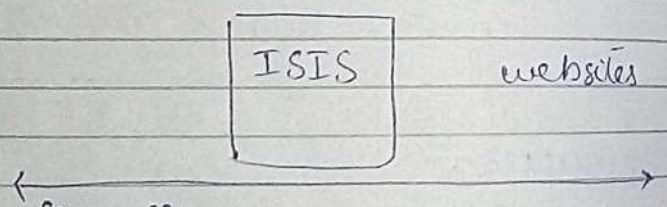
(Architectural)
e.g. code optimization.

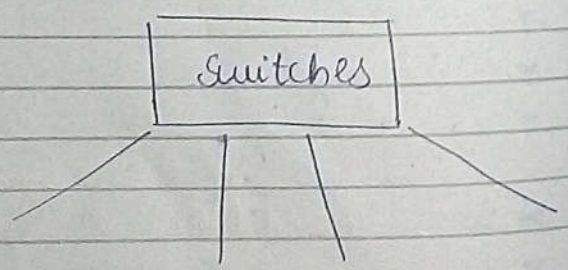
Security:-

- Authentication
- Login (user id / password)
- Authorization (permissions)
- SQL Injection
- Hacking
- cross domain request
- Direct Browsing (not in code, flag set in server)
- Dependency injections.

normal web
(Deep web
Black web)

Firewall





Some configuration handled here (emails, sites blocking)

which request will be handled this thing handled.

authorization & authentication that will be done with token based identification.

- Token generated on first time login and that token return to the client using session.

Api – Application programming Interface.

- Amazon web services AWS Devops.
- some sites give on request api.
- cc Authentication
  - Source identification (system IP address).
  - Token based.

Attributes case senario: for availability.

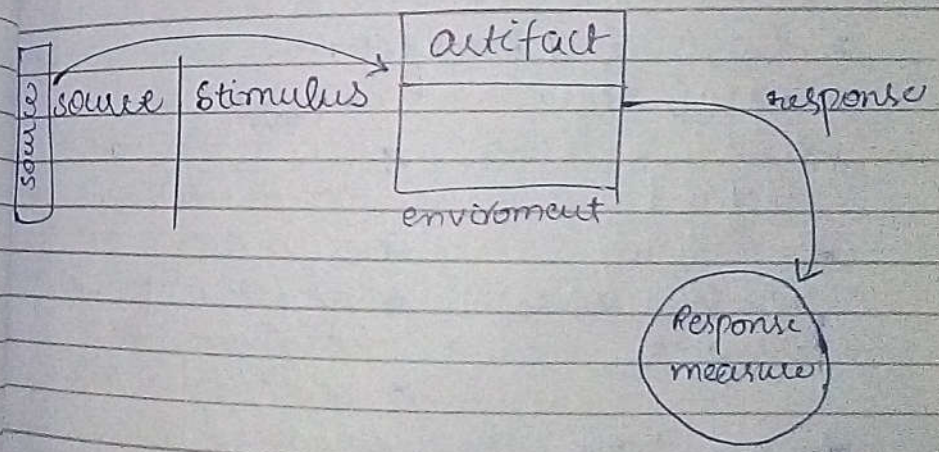
* Source of Stimulus:- due to which that stimulus happend e.g. server down.
* Stimulus :- The action which happend due to some error that is stimulus e.g. crashed.

* Enviroment :-
  normal
  degated
* Artifect
  jahan pe asar aye.
  e.g. memory leakage
* response
* Response time measure.

System Quality attributes & Business Quality attributes :-
  case senarios.

lecture 20:-

source	Developer
Stimulus	UI changed
Artifact	code
Enviroment	Design Time
Response	Modification Done
Response measure	2HRS.



Business Outliers

1. Time to market
2. Cost & benefit
3. projected lifetime of the system
4. Target Market
5. Integration with legacy system.

1. Time to Market:-

کونسے ٹائم پر ایپلیکیشن کو مارکیٹ میں لانا ہے
this is very important for a company for success and to compete the market.

2. Cost & benefit:-

- project
  - timeline defined
  - one client/ specific to customer
- product.
  - in which you do marketing.
  - each one use.
  - Investment, profit based on market
  - beneficial or not.

* when your are doing work project: developing phase.
* when project completed: product

check that how much you invest in product and in near future what profit you will gain.

3. projected lifetime of the system:-

patches => 1.1, 1.2, 1.3...

versions => 1, 2, 3, ...

e.g BANKS, DOS
(changes) step by step (Desktop etc)
windows changed one by one.

4. Target market:-

- are for products
- mostly are website.
- per user
- a specific organization or may be the ~~ter~~ normal user will be the targeted market.

5. Integration with legacy system:-

Business Intelligence
Databases (Qlikview).

ORACLE
SQL Server
Access
Excel
FQA DATA

} legacy systems

Lecture 21:- 15-05-18

Acheiving Quality attributes

1. Tactics:-

Design Pattern
Architectural pattern
Architectural strategies

" is a design decision that influences the control of system quality attribute response. "

Design pattern:-

Singleton : one instance can ucali
static/

Abstract Factory : polymorphism
FASEADE :
Proxy :

Architectural pattern:-

MVC, Peer to peer

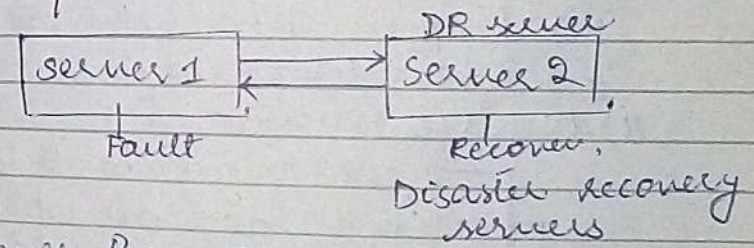
Architectural strategies:-

when multiple pattern of Design & architecture will be combine that is called strategy.

1. Availability Tactics:-

(i) Fault Detection
- Exception Handling

(ii) Fault Recovery
- Rollback
  e.g Database inserting data in application.
  on complete commit that.
- Application restart if crashed
  if system crashed forcefully restart.
- State synchronization
  In windows we get different points to recover window.



(iii) Fault Prevention
a. Removal for service
b. Transactions
c. Process Monitor

a. Removal for services:-
if your application/service is using by 3rd party and fault occur due to that service cut of that service.

b- Transaction
if Fault occured then rollback.

c- Process monitor
all running processes monitored
If there is a chance of occuring fault notifications start.

3. Modifibility Tactics:-

- "How fast a system incoopeate new changes is called modifibility"

(i) Localize modifications.
(ii) Prevent Ripple Effect
(iii) Daffer binding Time.
(iv) Intermediary layers.

(i) Localize modification:

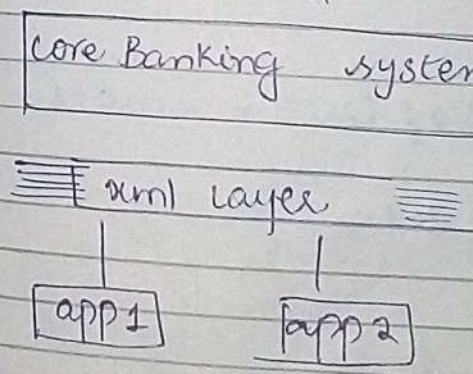
- Generalized module
- limit possible options
- Anticipate expected changes
- make module configurable and generalized for different requirements.
- possible actions / uses / options include (unknown user product / for Business purpose)
- Forecast the changes in near future

(ii) Prevent Ripple Effect:-
↳ continuous responses against some action.

to prevent this remove dependencies

- syntax of Data & Services
  e.g. infinite loop (star's example).
- Semantics of Data & services
- sequence of Data & controls
- Interface Identity.

(iv) Intermediary layers:-
web services, API's, XML layers. } open source function location communicate ~~to~~ two apps. data pass flow.



Lecture 22:-

27-05-18

Ripple Effect:-

Dependencies:-

Data

Checkintime HH:mm:ss

Checkout Time

Reports changed by changing the formats.

* need to reduce the dependencies to get less ripple effect and if Ripple application runs fast.

1. Syntax of Data & Services
2. Semantics of Data & services
3. Sequence of Data & services
4. Interface identity
5. Quality of

2. Semantics of Data & services:-

↳ logic

can decide

3.

flow of Data

if multiple classes data passing

4. Interface Identity:-

which classes implements the interface that are dependent upon that interface.

5. Quality of code & services:-

How you implement & how many functions are used and made by themselves.

(iii) Daffer binding time:-

Static binding - compile time

dynamic binding - Runtime

Daffer binding - add delays

1. Runtime registration
2. Configuration files
3. polymorphism
4. Component Replacement

→ on run time you tell that you need to register.
e.g Printing

→ web.config modules & dlles configured.

→ overloading & overriding

→ changes replaced in one module and that will not effect other

/remaining application.

modifiability increased through this

4. Performance tactics:-

↳ Response time

these tactics controls our response time.

(i) Hardware Resources
(ii)

Availability of resources
usage of resources
→ CPU, datastore, memory & buffers & Bandwidth

e.g. Data streaming applications use the resources.

- control resources
- control events

→ scheduling of resources to do the events.

→ that handled by resources

Types

* FIFO
all are equal
* Fixed priority
Semantic importance
Process on priorities.

↳ fetch data
↳ Process controller
↳ Event Monitor.

e.g. task manager has high priority

* Dynamic Priority Scheduling
request deadline mentioned that in how much time it should be completed.
(based on request deadline priority sets.)

* Round Robin
recyclic time assigned for resource allocation.

e.g Req 1	200ms
Req 2	400ms
Req 3	250ms
Req 4	100ms
100ms	assigned/Quantum time.

5. Security Tactics:-

(i) Registering attacks.
(ii) Detecting attacks
(iii) Recovering from attacks

Lecture 23:- 26-06-2018

Design the Architecture:-

a- Architecture in SDLC
b- Designing architecture
c- Forming Teams.
d- Creating Skeletal system

b- architecture Drivers
Functional
non functional

a- Attribute Driven Design:- (ADD)

Steps

1. choose the module (system/SW) to decomposed
2. Define the module
   a- choose architectural drivers
   b- architectural pattern
   c- module's instenciation → connection of modules.
   d- modules interfaces
   e- Integration of child modules
3. Repeat steps for every module

module :-

Garage door opening system.