Data Flow Diagram (DFD)

DFD ek diagram hota hai jo data ka flow dikhata hai system ke andar.

Simple bole toh: data kaha se aata hai, kaha jata hai aur kaise process hota hai.

DFD mein 4 main cheezein hoti hain:

1. External Entity (Rectangle): System ke bahar wale log ya system

2. Process (Circle): Data pe jo operations hote hain

3. Data Store (Open rectangle): Jaha data temporary ya permanently store hota hai

4. Data Flow (Arrow): Data kis direction mein jaa raha hai

Extra (Exam Point):

- DFD system ko easy banata hai samajhne ke liye

- Logical vs Physical DFD ka concept bhi isi se linked hota hai

- Har process ka unique number hota hai (1.0, 2.0 etc.)

Functional vs Non-functional Requirements

Functional: Jo kaam software karega. Example: Login, Search, Payment.

Non-functional: Kaise karega - speed, security, usability.

Simple:

- Functional = features

- Non-functional = quality

Extra:

- Non-functional include: Performance, Reliability, Scalability, Maintainability

- Functional clear hone chahiye before development starts

SRS (Software Requirement Specification)

SRS ek document hota hai jisme software ke saare requirements likhe hote hain.

Simple bole to: Client ko kya chahiye, uska pura likhit version.

Extra:

- Use case, functional + non-functional dono include hota hai
- Developer, tester, client - sabke liye reference hota hai
- IEEE format follow karta hai mostly

Logical vs Physical DFD

Logical DFD: Sirf 'kya' ho raha hai dikhata hai (no technical stuff)

Physical DFD: 'Kaise' ho raha hai, real system ke components ke saath

Simple:

- Logical = Abstract view
- Physical = Actual implementation idea

Extra:

- Logical helpful in initial stages of SDLC
- Physical helpful during design and deployment

Function Oriented vs Object Oriented Design

Function Oriented: System = Functions ka group

Object Oriented: System = Objects ka group (with data + functions)

Simple:

- FO: Structured programming (C)
- OOD: Classes and Objects (C++, Java)

Extra:

- OOD supports reuse, maintenance better

- FO is easy for small projects

- OOD useful in large, scalable systems

SPM (Software Project Management)

SPM ka matlab hai software project ko plan, execute aur control karna.

Goal: Time pe, budget me, aur quality ka software deliver karna.

Extra:

- Covers: Scheduling, Budgeting, Risk Handling, Team Management

- Tools: Gantt chart, PERT, CPM

Risk Management (Reactive vs Proactive)

Risk management ka matlab hai future ke problems ke liye ready rehna.

- Reactive: Problem hone ke baad solution sochna (firefighting)

- Proactive: Pehle se soch ke risk avoid ya control karna

Extra:

- Proactive better hota hai large projects ke liye

- Risk register maintain kiya jata hai

Risk Assessment

Risk assessment mein hum risk identify, analyze aur evaluate karte hain.

Steps:

1. Risk Identify karna

2. Impact aur Probability samajhna

3. Risk priority dena

Extra:

- Tools: Risk Matrix, Risk Register

- Helps in proper planning and control

Risk Control vs Risk Mitigation

Risk Control: Risk ke impact ko control karna ya avoid karna

Risk Mitigation: Risk ka impact kam karna (reduce karna)

Simple:

- Mitigation = Kam karna

- Control = Avoid ya handle karna

Example: Backup lena = Mitigation

CPM (Critical Path Method)

CPM ek project scheduling technique hai jisme critical path nikalte hain

- Critical Path = longest path, jiska delay ka matlab project delay

Extra:

- Helps in finding minimum project duration

- Float/Slack time bhi nikalte hain

Verification and Validation

Verification: Product sahi ban raha hai ya nahi (process check)

Validation: Right product ban gaya ya nahi (actual output check)

Simple:

- Verification = Are we building the product right?

- Validation = Are we building the right product?

Extra:

- V model mein dono hota hai

Types of Testing

1. Unit Testing

2. Integration Testing

3. System Testing

4. Acceptance Testing

Extra:

- Unit = smallest part test

- Integration = module combine kar ke test

- System = full app test

- Acceptance = client check kare

Error Seeding

Error seeding matlab: jaan bujh ke software mein bugs daalna

Purpose: tester kitne bugs detect kar raha hai wo dekhna

Extra:

- Quality check karne ke liye use hota hai

- Real bugs estimate karne mein help karta hai

Cohesion and Coupling

Cohesion: Module ke elements kitne related hain (high chahiye)

Coupling: Modules ek dusre pe kitne dependent hain (low chahiye)

Extra:

- High Cohesion + Low Coupling = Good design

White Box Testing

White box testing mein internal code, logic ko test kiya jata hai

- Developer hi karta hai mostly

Extra:

- Includes: Statement coverage, branch coverage, path coverage

Branch Coverage vs Condition Coverage

Branch coverage: Har if-else jaise decision cover hona chahiye

Condition coverage: Har individual condition (true/false) test hona chahiye

Extra:

- Condition zyada detailed hota hai

Data Flow Testing

White box technique hai jisme variable ke life cycle ka flow check karte hain

- kaaha define ho raha hai, kaaha use ho raha hai, etc.

Extra:

- DU (Define-Use) chains check ki jati hain

Black Box Testing

Black box testing mein software ke input/output test karte hain, bina code dekhe

- Functional testing hoti hai

Extra:

- User perspective se hoti hai

Boundary Value Testing

Black box ka type - input range ke edge pe value test karte hain

Example: Age valid hai 18-30, toh test: 17, 18, 30, 31

Extra:

- Errors mostly boundaries pe hi hote hain

Maintenance Types

1. Corrective - bug fix karna

2. Adaptive - environment change ke hisaab se

3. Perfective - improvement

4. Preventive - future problem avoid karna

Simple: CAP-P

MTBF vs MTTR

MTBF - Mean Time Between Failures (reliability)

MTTR - Mean Time To Repair (maintainability)

Extra:

- High MTBF = Reliable system

- Low MTTR = Quick fix

Reverse Engineering

Existing product ka analysis karke uska design/code samajhna

Example: Competitor ke software ko decompile karke samajhna

Extra:

- Use in legacy systems, no documentation

CASE Tools

CASE = Computer Aided Software Engineering

- Tools jo software development ko automate/help karte hain

Examples:

- Rational Rose, StarUML, Visual Paradigm

- Support design, coding, testing