Structured Analysis vs. Structured Design

Page 1 Stub & Driver: Top-Down. Bottom-up. If one level module is ready. but one is not then we make dummy modules for this level to test previous module. Those dump modules are Stub & Driver.. Mutation Testing: value Mutation. Original Mutant Mutation = Killed Mutant/Total Mutant *100% int y = 6; int y=5; Original Mutant Original Mutant if (p>=4) if (p<4) if (a>b) if (a>=b) p=4; p=4; a = a+b; a=a*b; Decision Mutation. Statement Mutation. Test case a=2 b=2. a+b -> 4 a-b a*b a/b a%b 0 4 1 0 Killed Live Killed Killed Mutation Score = 3/4 *100 = 75% Mutation Testing belong to White Box Testing as we make mutant of internal code. White { ① Cooperate based Testing. Box Testing ② Data flow Testing. ③ Mutation Testing. team which Team structure does not use heirenchy -> Democratic. Gama-Testing (■-Testing) -> Acceptance testing other name of 8-Testing is Realise candidate Testing or Final Phase Testing. It is done after α& B Testing. Software Reliability: Software Reliability is a software quality that determines the ability of the software to retain good characteristics over time without the presence of any failure. Software reliabity is talaten is this can be increased or decreased. So, we require Software reliability growth. **Page 2** -MTTF-Mean Time to Failure. MTTR-Mean. Time to Repaire. Estrantz: t1, tz upto : MTTF = $\sum$(ti+1 - ti)/n. 1 t1,t2, t3,.... tn, MTTR = $\sum$(tn-t1)/n. 1 * MTBF = Mean Time between Failure. =MTTF+MTTR what do you mean by 100 hrs MTBF It means that once a failure occures the next failure ts expected to occure after 100 hrs. * Availability: It measures how likely software canbe avaliade for use over a period of time without failure. Availability = (MTTF/MTBF) X100%. Q. what is the availability of software if it's MTBF = 25 days & MTTR = 6 hours. -> MTTF = MTBF-MTTR = 25*24 - 6 = 594 hours. :: Availability = (594/600) *100% = 99% DFD to Structure Chart. 28.10.2024 Data Input. I/P data validity Central FROM User Check Transform OR. O-2 Processing. Input or Invalid Determine Afferent squared root Branch Input final probt. Squared Invalid data. result on valida. Ke Validated data Output or Capture Squared root data Efferent Taxe Captured Display (0-3) Branch Input Data from Capture user data valid data Giet data Validate Data. Invalid data. **Page 3** * Structure chart and DFD is Function oriented Design. Structure Chart: Structure Chart is a chart which can be is derived from DFD. It represent the system in details and determines the no. of modules along with functions of each modules. Structure chart represent Heirenchy Structure of each module * Structured chart is used to determine the no. of modules based on which code will be written. # Short-note SA VA SD. (Structured Analysis vs. Structured Des This two approaches belong to function oriented design. structured Analysis analises users requirement and data flow inside the system. Example of structured Analysis is D Structured Design approach will work after SA. Structured Design is used to determine the no. of modele Ex. Structured Chart. There are two strategies by which we can convert & DFD. into corrosjoonding structured chart. ① Transform Analysis: 3-Parts of DFD: → 1. Input section, →2. Logical Processing → 3. Output section. * Factoring is the process of decomposing one module to multiple module in Structur Chart. o calls with data. calling with control signald →: Arrows are to call a module or jump to a module. A [ ] A Conditional calling B A calling B repeatedly. B Fan-in: no. of ips coming to module. [✓] Fan-out: no. of 0/8. going from module. **Page 4** * Structure chart and DFD is Function Oriented Design. Structure Chart: Structure Chart is a chart which can be is derived from DFD. It represent the system in details and determines the no. of modules along with functions of each modules. Structare chart represent Heirerchycal Structure of each module * Structured chart is used to determine the no. of modules based on which code will be written. # Short-note SA vs SD. (Structured Analysis vs. Structured Design This two approaches belong to function oriented design. Structured Analysis analises users requirement and data flow inside the system. Example of structured Analysis is DFD Structured Design approach will work after SA. Structured Design is used to determine the no. of modules Ex. Structured Chart. There are two strategies by which we can convert a DFD. into corrosjoonding structured chart. ① Transform Analysis: 3-Parts of DFD: →1. Input Section, →2. Logical Processing →3. Output section. * Factoring is the process of decomposing one module to multiple module in Structur Chart. o calls with data. calling with control signaldat →: Arrows cure to call a module or jump to a module. A [ ] Conditional calling. B A calling B repeatedly, B [A] Fan-in: no. of ilps coming to module. [✓] Fan-out: no. of o/p going from module. **Page 5** Documentation External Documentation Internal Documentation. Internal Documentation: This type of documentation is required for Knowing different activities at different

stages of SDLC and this type of documentation is only used by employees at a particular stage and it's not shown to customer. External Documentation: This type of documentation is given to the Customer so that customer can operate the product without any problem. Ex. User's manual. # what are different types of Feasibility studies ① Technical Feasibility, ③ Economical Feasibility, ③ Operationd Feasibility. Leagal Feasibility. (4th)

Short-note SA vs SD: Structured Analysis analyzes user requirements and data flow in the system. Example: DFD. Structured Design works after SA and is used to determine the number of modules (e.g., Structured Chart).

Feasibility Studies: Technical, Economic, Operational, and Legal feasibility are considered. Other techniques include code review, code inspection, and walkthroughs.