

Fundamentals of Software Development (4331604) - Summer 2025 Solution

Milav Dabgar

May 17, 2025

પ્રશ્ન 1(a) [3 ગુણ]

Give IEEE definition of software. Write one example of each for application and system software.

જવાબ

IEEE વ્યાખ્યા: સોફ્ટવેર એ કમ્પ્યુટર પ્રોગ્રામ્સ, પ્રક્રિયાઓ, નિયમો અને સંલગ્ન દસ્તાવેજુકરણ અને ડેટાનો સંગ્રહ છે.

ઉદાહરણો:

કોષ્ટક 1. સોફ્ટવેર પ્રકારો અને ઉદાહરણો

સોફ્ટવેર પ્રકાર	ઉદાહરણ	હેતુ
એપ્લિકેશન સોફ્ટવેર	Microsoft Word	વર્ડ પ્રોસેસિંગ અને દસ્તાવેજ બનાવટ
સિસ્ટમ સોફ્ટવેર	Windows 10	હાર્ડવેર સંસાધનોનું સંચાલન કરતી ઓપરેટિંગ સિસ્ટમ

- એપ્લિકેશન સોફ્ટવેર: ચોક્કસ કાર્યો પૂર્ણ કરવા માટે અંતિમ વપરાશકર્તાઓ માટે રચાયેલ પ્રોગ્રામ્સ
- સિસ્ટમ સોફ્ટવેર: કમ્પ્યુટર હાર્ડવેરનું સંચાલન અને સંચાલન કરતા પ્રોગ્રામ્સ

મેમરી ટ્રીક

“Apps help Users, Systems help Hardware: તફાવત યાદ રાખવા માટે મેમરી ટ્રીક.”

પ્રશ્ન 1(b) [4 ગુણ]

Write a short note on data dictionary.

જવાબ

ડેટા ડિક્શનરી એ એક કેન્દ્રિય રિપોਜીટરી છે જેમાં સિસ્ટમમાં વપરાતા ડેટા તત્વોની વ્યાખ્યાઓ અને લાક્ષણિકતાઓ હોય છે.

કોષ્ટક 2. ડેટા ડિક્શનરીના ઘટકો

ઘટક	વર્ણન
ડેટા નામ	ડેટા તત્વ માટે અનન્ય ઓળખકર્તા
ઉપનામો (Aliases)	વપરાયેલ વૈકલ્પિક નામો
વર્ણન	હેતુ અને અર્થ
ડેટા પ્રકાર	ફોર્મેટ (પૂર્ણાંક, સ્ટ્રિંગ, વગેરે)
લંબાઈ	કદ મર્યાદાઓ
મૂલ્યો	માન્ય શ્રેણી અથવા સેટ

- હેતુ: વિકાસ ટીમમાં ડેટાના ઉપયોગમાં સુસંગતતા સુનિશ્ચિત કરે છે
- ફૂથદા: અસ્પષ્ટતા ઘટાડે છે, સંચાર સુધારે છે, ડેટા વ્યાપ્તાઓને પ્રમાણિત કરે છે
- ઉપયોગ: સિસ્ટમ ડિજાઇન અને ડેટાબેઝ બનાવટ દરમિયાન સંદર્ભિત

મેમરી ટ્રીક

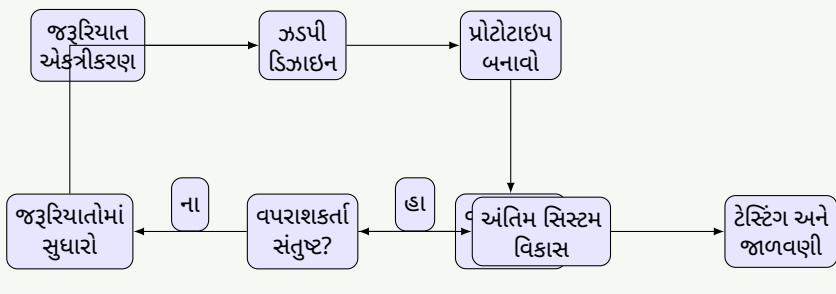
“Dictionary Defines Data Clearly: ડેટા ડિક્શનરીનો હેતુ.”

પ્રશ્ન 1(c) [7 ગુણ]

Explain prototype model with figure.

જવાબ

પ્રોટોટાઇપ મોડેલ એક પુનરાવર્તિત અભિગમ છે જ્યાં જરૂરિયાતોને વધુ સારી રીતે સમજવા માટે પ્રારંભિક તબક્કે કાર્યકારી મોડેલ બનાવવામાં આવે છે.



આકૃતિ 1. પ્રોટોટાઇપ મોડેલ

લાક્ષણિકતાઓ:

કોષ્ટક 3. પ્રોટોટાઇપ મોડેલ તબક્કાઓ

તબક્કો	પ્રવૃત્તિ	આઉટપુટ
જડપી ડિજાઇન	મૂળભૂત આર્કિટેક્ચર	પ્રારંભિક ડિજાઇન
પ્રોટોટાઇપ બિલ્ડ	કાર્યકારી મોડેલ	ટેસ્ટ કરી શકાય તેવી સિસ્ટમ
વપરાશકર્તા મૂલ્યાંકન	પ્રતિસાદ સંગ્રહ	જરૂરિયાતોમાં સુધારો

- ફૂથદા: પ્રારંભિક વપરાશકર્તા પ્રતિસાદ, ઘટાડેલું વિકાસ જોખમ, વધુ સારી જરૂરિયાત સમજ
- ગરફાયદા: અપૂર્વાનુભૂતિ વિશ્લેષણ તરફ દોરી શકે છે, ગ્રાહક પ્રોટોટાઇપને અંતિમ ઉત્પાદન તરીકે અપેક્ષા રાખે છે
- શ્રેષ્ઠ માટે: અસ્પષ્ટ જરૂરિયાતોવાળા પ્રોજેક્ટ્સ

મેમરી ટ્રીક

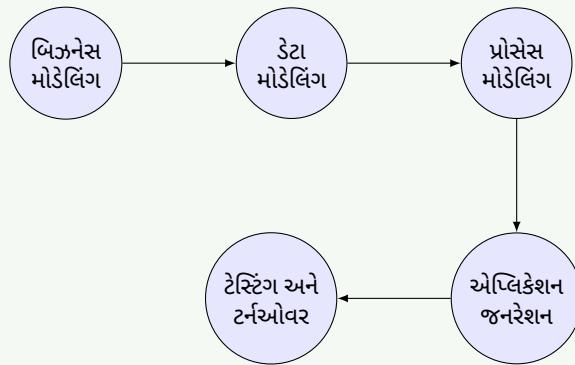
“Prototype Proves Possibilities: પ્રોટોટાઇપિંગનો મુખ્ય લાભ.”

પ્રશ્ન 1(c OR) [7 ગુણ]

Explain RAD model with advantages and disadvantages.

જવાબ

RAD (રેપિડ એપ્લિકેશન ડેવલપમેન્ટ) પ્રોટોટાઇપિંગ અને પુનરાવર્તિત વિકાસ દ્વારા જડપી વિકાસ પર ભાર મૂકે છે.



કોષ્ટક 4. RAD ફાયદા વિ. ગેરફાયદા

ફાયદા	ગેરફાયદા
જડપી વિકાસ	કુશળ ડેવલપર્સની જરૂર
પ્રારંભિક વપરાશકર્તા સંડોવણી	મોટા પ્રોજેક્ટ્સ માટે ચોગ્ય નથી
ઘટાડેલો ખર્ચ	વપરાશકર્તા પ્રતિબદ્ધતાની જરૂર
વધુ સારી ગુણવત્તા	જો મેનેજ ન કરવામાં આવે તો તકનીકી જોખમો

- મુખ્ય લક્ષણાં: સ્વચાલિત સાધનો અને 4GL પ્રોગ્રામિંગનો ઉપયોગ કરે છે
- સમયરેખા: સામાન્ય રીતે વિકાસ માટે 60-90 દિવસ
- ટીમ: નાની, અનુભવી વિકાસ ટીમો

મેમરી ટ્રીક

“RAD Rapidly Accelerates Development: RAD નો મુખ્ય ઘાલ.”

પ્રશ્ન 2(a) [3 ગુણ]

Give the full form of following: SQA, FTR, RAD, BVA, GUI, DFD

જવાબ

કોષ્ટક 5. સંક્ષોપો અને પૂર્ણ સ્વરૂપો

સંક્ષોપ	પૂર્ણ સ્વરૂપ
SQA	Software Quality Assurance
FTR	Formal Technical Review
RAD	Rapid Application Development
BVA	Boundary Value Analysis
GUI	Graphical User Interface
DFD	Data Flow Diagram

મેમરી ટ્રીક

“Software Quality And Formal Technical Reviews Rapidly Analyze Development, Boundary Value Analysis Guides User Interface, Data Flow Diagrams”

પ્રશ્ન 2(b) [4 ગુણ]

Define agile methodology. Discuss agile principles.

જવાબ

વ્યાખ્યા: Agile એ પુનરાવર્તિત સોફ્ટવેર વિકાસ અભિગમ છે જે સહયોગ, સુગમતા અને કાર્યકારી સોફ્ટવેરની જડપી ડિલિવરી પર ભાર મૂકે છે. મુખ્ય Agile સિદ્ધાંતો:

કોષ્ટક 6. Agile સિદ્ધાંતો

સિદ્ધાંત	વર્ણન
પ્રક્રિયાઓ કરતાં વ્યક્તિઓ	લોકો અને સંચાર પ્રાથમિકતા છે
દસ્તાવેજુકરણ કરતાં કાર્યકારી સોફ્ટવેર	કાર્યકારી સોફ્ટવેર એ પ્રાથમિક માપદંડ છે
ગ્રાહક સહયોગ	સતત ગ્રાહક સંડોવણી
ફેરફારોને પ્રતિસાદ	કઠોર યોજનાઓ કરતાં અનુકૂલનક્ષમતા

- પુનરાવર્તન કાળ: સામાન્ય રીતે 2-4 અઠવાડિયા (sprints)
- ડિલિવરી: વારંવાર કાર્યકારી સોફ્ટવેર રિલીઝ
- ટીમ માળખું: કોસ-ફંક્શનલ, સ્વ-વ્યવસ્થિત ટીમો

મેમરી ટ્રીક

“Agile Adapts And Advances: Agile મુખ્ય ફિલસ્ફૂરી.”

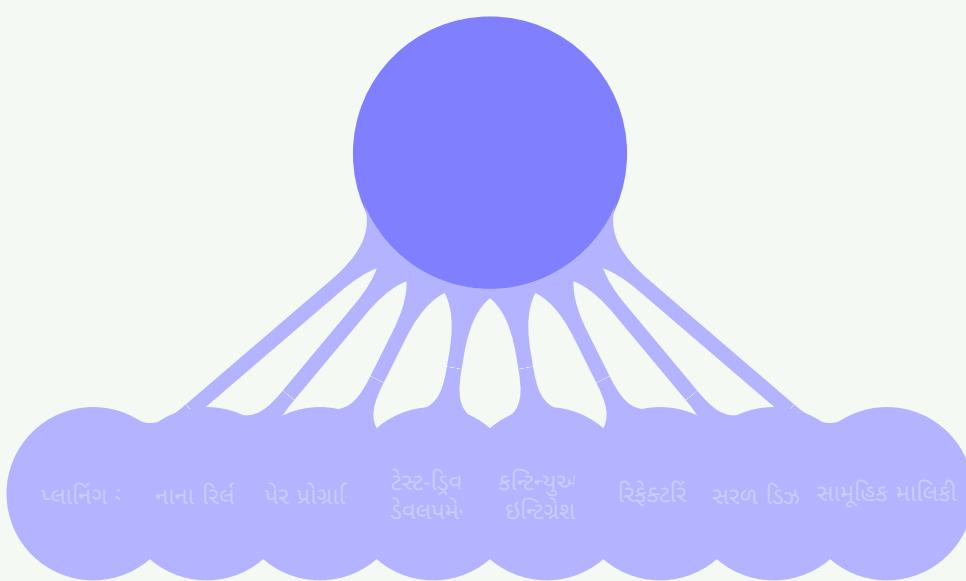
પ્રશ્ન 2(c) [7 ગુણ]

Explain XP model with its advantages and disadvantages.

જવાબ

XP (એક્સ્પ્રીમ પ્રોગ્રામિંગ) એ એજાઇલ મેથડોલોજી છે જે એન્જિનિયરિંગ પ્રેક્ટિસ અને ગ્રાહક સંતોષ પર ભાર મૂકે છે.

XP પ્રેક્ટિસ:



આકૃતિ 3. એક્સ્પ્રીમ પ્રોગ્રામિંગ પ્રેક્ટિસ

ફાયદા અને ગેરફાયદા:

કોષ્ટક 7. XP ફાયદા અને ગેરફાયદા

ફાયદા	ગેરફાયદા
ઉચ્ચ કોડ ગુણવત્તા	અનુભવી પ્રોગ્રામરોની જરૂર
જડપી પ્રતિસાદ	ગ્રાહક ઉપલબ્ધ હોવો જોઈએ
ઘટાડેલા બજ્ઝ	કોડ કેન્દ્રિત, ઓછું દસ્તાવેજુકરણ
સુગમતા	ખર્ચ અંદાજ કાઢવો મુશ્કેલ

- મુખ્ય પ્રથા: પેર પ્રોગ્રામિંગ કોડની ગુણવત્તા સુનિશ્ચિત કરે છે
- ટેસ્ટિંગ: સ્વચાલિત ટેસ્ટિંગ સાથે ટેસ્ટ-ફર્સ્ટ અભિગમ
- ગ્રાહક ભૂમિકા: ઓન-સાઇટ ગ્રાહક સતત પ્રતિસાદ પ્રદાન કરે છે

મેમરી ટ્રીક

“eXtreme Programming eXcels through Practices: XP ચોક્કસ પ્રથાઓ પર આધાર રાખે છે.”

પ્રશ્ન 2(a OR) [3 ગુણ]

Define black box testing. Give at least two names of black box testing method.

જવાબ

વ્યાખ્યા: બ્લેક બોક્સ ટેસ્ટિંગ આંતરિક કોડ સ્ટ્રક્ચરના જ્ઞાન વિના સોફ્ટવેર કાર્યક્ષમતાની તપાસ કરે છે, જે ઇનપુટ-આઉટપુટ વર્તણૂક પર ધ્યાન કેન્દ્રિત કરે છે.

બ્લેક બોક્સ ટેસ્ટિંગ પદ્ધતિઓ:

કોષ્ટક 8. બ્લેક બોક્સ પદ્ધતિઓ

પદ્ધતિ	વર્ણન
ઇક્સ્પ્રેસન્સ પાર્ટીશનિંગ	ઇનપુટને માન્ય/અમાન્ય વર્ગોમાં વિભાજિત કરે છે
બાઉન્ડ્રી વેલ્યુ એનાલિસિસ	ઇનપુટ સીમાઓ પર મૂલ્યોનું પરીક્ષણ કરે છે

- અભિગમ: જરૂરિયાતો અને સ્પષ્ટીકરણો પર આધારિત પરીક્ષણો
- ટેસ્ટર જ્ઞાન: આંતરિક કોડ જરૂરી નથી
- ફોક્સ: બાહ્ય વર્તણૂક અને કાર્યક્ષમતા

મેમરી ટ્રીક

“Black Box Behavior Based: ટેસ્ટિંગ ફોક્સ.”

પ્રશ્ન 2(b OR) [4 ગુણ]

Give the full form of CLI. Explain CLI in brief.

જવાબ

CLI: Command Line Interface

CLI લાક્ષણિકતાઓ:

કોષ્ટક 9. CLI સુવિધાઓ

પાસું	વર્ણન
ઇનપુટ પદ્ધતિ	વપરાશકર્તા દ્વારા ટાઇપ કરેલ ટેક્સ્ટ આદેશો
આઉટપુટ	ટેક્સ્ટ-આધારિત પ્રતિસાદ
નેવિગેશન	ફાઇલ/ડિરેક્ટરી કામગીરી માટે આદેશો
કાર્યક્ષમતા	અનુભવી વપરાશકર્તાઓ માટે જરૂરી

- ફાયદા: જરૂરી અમલીકરણ, ઓછો મેમરી વપરાશ, સ્થિરપણેબલ
- ગેરફાયદા: આદેશો શીખવાની જરૂર, નવા નિશાળીયા માટે વપરાશકર્તા મૈત્રીપૂર્ણ નથી
- ઉદાહરણો: Windows Command Prompt, Linux Terminal, DOS

મેમરી ટ્રીક

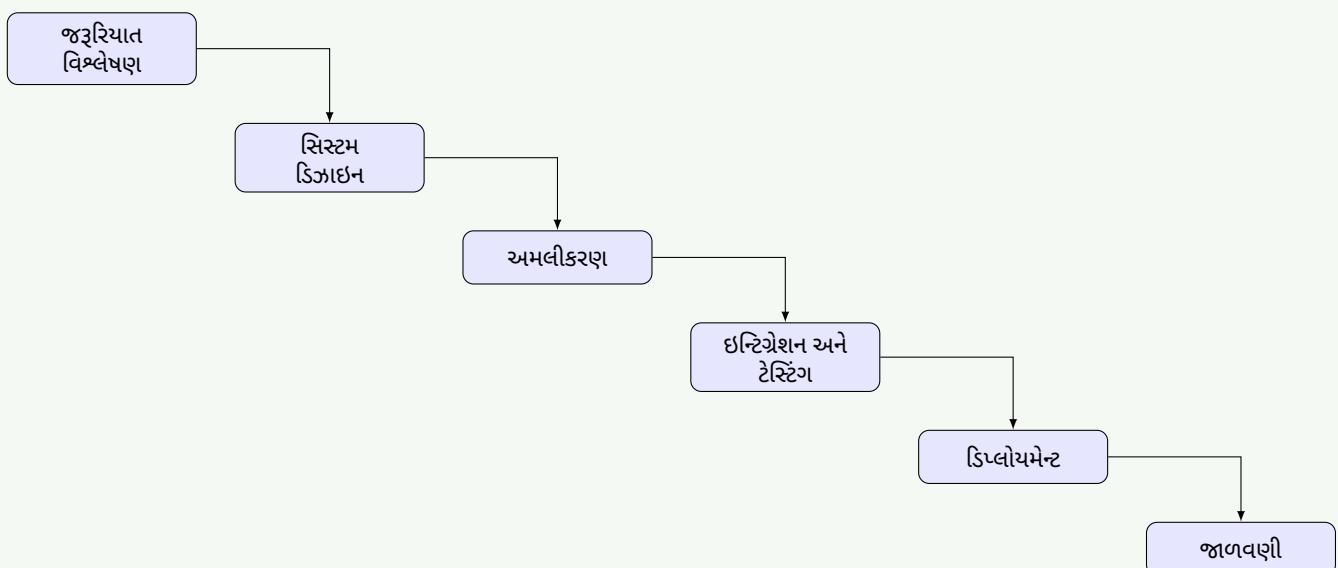
“Commands Lead Interaction: CLI ક્ષયપ્રતિક્ષયા મોડેલ.”

પ્રશ્ન 2(c OR) [7 ગુણ]

Explain waterfall model with neat figure.

જવાબ

વોટરફોલ મોડેલ એ એક રેખીય ક્રમિક અભિગમ છે જ્યાં આગામના તબક્કામાં જતા પહેલા દરેક તબક્કો પૂર્ણ થવો આવશ્યક છે.



અકૃતિ 4. વોટરફોલ મોડેલ

તબક્કાવાર વિગતો:

કાણક 10. વોટરફોલ મોડેલ તબક્કાઓ

તબક્કો	પ્રવૃત્તિઓ	ડિલિવરેબલ્સ
જરૂરિયાતો	જરૂરિયાતો એકત્રિત અને દસ્તાવેજુકરણ	SRS દસ્તાવેજ
ડિઝાઇન	સિસ્ટમ આર્કિટેક્ચર	ડિઝાઇન દસ્તાવેજો
અમલીકરણ	કોડ વિકાસ	સોર્સ કોડ
ટેસ્ટિંગ	કાર્યક્ષમતા ચકાસણી	ટેસ્ટ રિપોર્ટ્સ
ડિપ્લોયમેન્ટ	સિસ્ટમ ઇન્સ્ટોલેશન	કાર્યકરી સિસ્ટમ
જાળવણી	બગ ફિક્સસ, અપડેટ્સ	અપડેટ સિસ્ટમ

- ફાયદા: સરળ, મેનેજ કરવા માટે સરળ, સારી રીતે દસ્તાવેજ્ઞકૃત
- ગેરફાયદા: અક્કડ, મોંઢ ટેસ્ટિંગ, ફેરફારાને સમાવવા મુશ્કેલ

મેમરી ટ્રીક

“Water Always Flows Downward: વોટરફોલની કમિક પ્રકૃતિ.”

પ્રશ્ન 3(a) [3 ગુણ]

Give one word answer:

જવાબ

કોષ્ટક 11. એક શબ્દના જવાબો

પ્રશ્ન	જવાબ
સૌથી ઓછું કોહેશન છે	કોઇન્સિડેન્ટલ (Coincidental)
સૌથી વધુ કપલિંગ છે	કન્ટ૆ન (Content)
જટિલ પ્રવૃત્તિનો સ્લેક ટાઇમ છે	શૂન્ય (Zero)

મેમરી ટ્રીક

“Coincidental Cohesion, Content Coupling, Critical Zero”

પ્રશ્ન 3(b) [4 ગુણ]

Explain classification of coupling.

જવાબ

કપલિંગ મોડ્યુલો વચ્ચે પરસ્પર નિર્ભરતાને માપે છે. જાળવણી માટે ઓછું કપલિંગ વધુ સારું છે.
કપલિંગ પ્રકારો (શ્રેષ્ઠ થી ખરાબ):

કોષ્ટક 12. કપલિંગ પ્રકારો

પ્રકાર	વર્ણન	ઉદાહરણ
Data	પરિમાણો પસાર થાય છે	પરિમાણો સાથે પદ્ધતિ કોલ્સ
Stamp	ડેટા સ્ટ્રક્ચર પસાર થાય છે	ઓફજેક્ટ્સ/રેકોર્ડ્સ પસાર કરવા
Control	નિયંત્રણ માહિતી પસાર થાય છે	ફ્લેગ્સ/સ્વીચ્યો પસાર થાય છે
External	બાહ્ય ડેટા સંદર્ભ	ગ્લોબલ વેરિયેબલ્સ
Common	શેર કરેલ ડેટા વિસ્તાર	સામાન્ય મેમરી બ્લોક્સ
Content	આંતરિક ભાગમાં સીધો પ્રવેશ	અન્ય મોડ્યુલના ડેટામાં ફેરફાર

- શ્રેષ્ઠ પ્રથા: ડેટા કપલિંગ માટે લક્ષ્ય રાખો
- ટાળો: કન્ટ૆ન અને કોમન કપલિંગ
- ડિઝાઇન ધ્યેય: મોડ્યુલો વચ્ચે નિર્ભરતા ઘટાડો

મેમરી ટ્રીક

“Data Stamps Control External Common Content: કપલિંગ કડકતાનો કમ.”

પ્રશ્ન 3(c) [7 ગુણ]

Define following terms (don't just give the full form):

જવાબ

કોષ્ટક 13. સોફ્ટવેર વ્યાખ્યાઓ

શબ્દ	વ્યાખ્યા
UI	User Interface - સાધન જેના દ્વારા વપરાશકર્તાઓ સોફ્ટવેર સિસ્ટમો સાથે કિયાપ્રતિક્રિયા કરે છે
SE	Software Engineering - એન્જિનિયરિંગ સિદ્ધાંતોનો ઉપયોગ કરીને સોફ્ટવેર વિકાસ માટે વ્યવસ્થિત અભિગમ
PMC	Project Management and Control - સોફ્ટવેર પ્રોજેક્ટસનું આયોજન, દેખરેખ અને નિયંત્રણ
SDLC	Software Development Life Cycle - વિભાવનાથી જાળવણી સુધીના સોફ્ટવેર વિકાસમાં સામેલ તબક્કાઓ
Verification	સોફ્ટવેર નિર્દિષ્ટ જરૂરિયાતો અને ડિઆઇને પૂર્ણ કરે છે કે કેમ તે તપાસવાની પ્રક્રિયા
Validation	સોફ્ટવેર વપરાશકર્તાની જરૂરિયાતો અને હેતુપૂર્વકના હેતુને પૂર્ણ કરે છે કે કેમ તે તપાસવાની પ્રક્રિયા
SRS	Software Requirements Specification - સોફ્ટવેર કાર્યક્ષમતા અને મર્યાદાઓનું વર્ણન કરતો વિગતવાર દસ્તાવેજ

- **Verification:** "શું આપણે ચોગ્ય રીતે ઉત્પાદન બનાવી રહ્યા છીએ?"
- **Validation:** "શું આપણે ચોગ્ય ઉત્પાદન બનાવી રહ્યા છીએ?"
- **મુખ્ય તફાવત:** વેરિફિકેશન સ્પષ્ટીકરણો તપાસે છે, વેલિડેશન વપરાશકર્તા સંતોષ તપાસે છે

મેમરી ટ્રીક

"Users Interact, Software Engineers Plan, Managing Cycles, Specifications Define, Verification checks Requirements, Validation checks Satisfaction, Requirements Specify Software"

પ્રશ્ન 3(a OR) [3 ગુણ]

Explain menu based UI with advantages and disadvantages.

જવાબ

મેનુ-આધારિત UI વપરાશકર્તા પસંદગી માટે શ્રેણીબદ્ધ મેનુમાં વિકલ્પો રજૂ કરે છે.
ફાયદા વિ. ગેરફાયદા:

કોષ્ટક 14. મેનુ UI ફાયદા અને ગેરફાયદા

ફાયદા	ગેરફાયદા
શીખવા માટે સરળ	નિષ્ણાતો માટે ધીમું
ભૂલો ઘટાડે છે	મર્યાદિત સુગમતા
સ્વ-સ્પષ્ટીકરણાત્મક	સ્કીન સ્પેસ વપરાશ

- **માળખું:** વિકલ્પોનું શ્રેણીબદ્ધ સંગઠન
- **નેવિગેશન:** પોઇન્ટ-એન્ડ-કિલ્ક અથવા કીબોર્ડ શોર્ટકટ્સ
- **શ્રેષ્ઠ માટે:** સારી રીતે વ્યાખ્યાપિત કાર્યો વાળી એપ્લિકેશન્સ

મેમરી ટ્રીક

"Menus Make Choices Clear: મેનુ UI નો લાભ."

પ્રશ્ન 3(b OR) [4 ગુણ]

Explain classification of cohesion.

જવાબ

કોહેશન માપે છે કે મોડ્યુલની અંદરના તત્ત્વો કેટલા નજીકથી સંબંધિત છે. ઉચ્ચ કોહેશન વધુ સારું છે.
કોહેશન પ્રકારો (શ્રેષ્ઠ થી ખરાબ):

કોષ્ટક 15. કોહેશન પ્રકારો

પ્રકાર	વર્ણન
Functional	એકલ, સારી રીતે વ્યાખ્યાયિત કાર્ય
Sequential	એક ઘટકનું આઉટપુટ આગામીને ફીડ કરે છે
Communicational	તત્ત્વો સમાન ડેટા પર કામ કરે છે
Procedural	તત્ત્વો અમલીકરણ કરુને અનુસરે છે
Temporal	તત્ત્વો તે જ સમયે ચલાવવામાં આવે છે
Logical	તત્ત્વો સમાન કાર્યો કરે છે
Coincidental	તત્ત્વો અવ્યવસ્થિત રીતે જૂથબદ્ધ

- દ્યેય: ફંક્શનલ કોહેશન પ્રાપ્ત કરો
- ડિઝાઇન સિદ્ધાંત: દરેક મોડ્યુલની એક જ જવાબદારી હોવી જોઈએ
- માપન: ઉચ્ચ કોહેશન = વધુ સારી ડિઝાઇન

મેમરી ટ્રીક

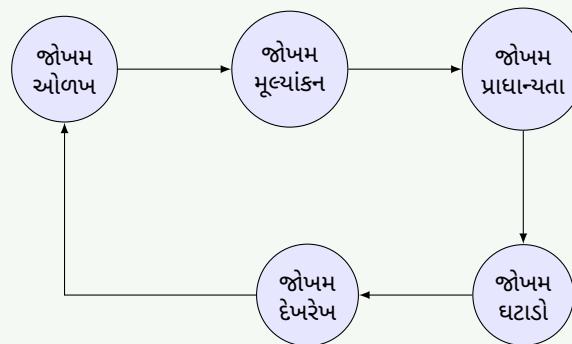
“Functional Sequences Communicate Procedures Temporally through Logical Coincidence: કોહેશન મજબૂતાઈનો ક્રમ.”

પ્રશ્ન 3(c OR) [7 ગુણ]

Define risk. Explain risk management.

જવાબ

જોખમ વ્યાખ્યા: સંભવિત સમસ્યા જે સોફ્ટવેર વિકાસ દરમિયાન આવી શકે છે, જે પ્રોજેક્ટની સફળતા પર નકારાત્મક અસર કરે છે.



આકૃતિ 5. જોખમ વ્યવસ્થાપન પ્રક્રિયા

જોખમ વ્યવસ્થાપન પ્રવૃત્તિઓ:

કોષ્ટક 16. જોખમ પ્રવૃત્તિઓ

પ્રવૃત્તિ	વર્ણન	આઉટપુટ
ઓળખ	સંભવિત સમસ્યાઓ શોધો	જોખમ સૂચિ
મૂલ્યાંકન	સંભાવના અને અસરનું વિશ્લેષણ કરો	જોખમ વિશ્લેષણ
પ્રાધાન્યતા	મહત્વ દ્વારા જોખમોને રેન્ક કરો	પ્રાયોરિટી મેટ્રિક્સ
ઘટાડો	જોખમ પ્રતિભાવોની યોજના બનાવો	મિટિગેશન વ્યૂહરચના
દેખરેખ	જોખમ સ્થિતિ ટ્રેક કરો	અપડેટ થયેલ જોખમ સ્થિતિ

- જોખમ પ્રકારો: ટેકનિકલ, પ્રોજેક્ટ, બિઝનેસ જોખમો
- વ્યૂહરચના: ટાળો, સ્થાનાંતરિત કરો, ઓફ્લાઇન, સ્વીકારો
- સાધનો: જોખમ મેટ્રિસીસ, સંભાવના-અસર ચાર્ટ

મેમરી ટ્રીક

“Risk Requires Careful Planning: જોખમ વ્યવસ્થાપનનું મહત્વ.”

પ્રશ્ન 4(a) [3 ગુણા]

Define: Error, Failure, Test case

જવાબ

કોષ્ટક 17. ટેસ્ટિંગ વ્યાખ્યાઓ

શબ્દ	વ્યાખ્યા
Error	સોફ્ટવેર ડેવલપમેન્ટ પ્રક્રિયા દરમિયાન માનવીય ભૂલ
Failure	અપોક્ષિત પરિણામોમાંથી સોફ્ટવેર વર્તાણૂકનું વિચલન
Test case	ચોક્કસ કાર્યક્ષમતા અથવા સિસ્ટમ જરૂરિયાતને ચકાસવા માટે શરતોનો સમૂહ

- સંબંધ: ભૂલ ખામી તરફ દોરી જાય છે, ખામી નિષ્ફળતાનું કારણ બને છે
- ભૂલ સ્રોત: ડેવલપર ભૂલો, જરૂરિયાતોમાં ગેરસમજ
- ટેસ્ટ કેસ ઘટકો: ઇનપુટ, અપોક્ષિત આઉટપુટ, અમલીકરણ પગલાં

મેમરી ટ્રીક

“Errors Cause Failures, Tests Catch Problems: કારણ અને અસરની સાંકળ.”

પ્રશ્ન 4(b) [4 ગુણા]

Identify any six functional requirements of ATM system.

જવાબ

ATM સિસ્ટમ કાર્યકારી જરૂરિયાતો:

કોષ્ટક 18. ATM જરૂરિયાતો

જરૂરિયાત	વર્ણન
વપરાશકર્તા ઓથેન્ટિકેશન	એકાઉન્ટ એક્સેસ માટે PIN વેરિફિકેશન
બેલેન્સ ઈન્કવાયરી	વર્તમાન એકાઉન્ટ બેલેન્સ પ્રદર્શિત કરો
રોકડ ઉપાડ	વિનંતી કરેલ રોકડ રકમ વિતરિત કરો
ફુંડ ટ્રાન્સફર	એકાઉન્ટ્સ વચ્ચે પૈસા ટ્રાન્સફર કરો
ટ્રાન્ઝેક્શન હિસ્ટ્રી	તાજેતરના ટ્રાન્ઝેક્શન રેકૉર્ડ્સ બતાવો
PIN ફેરફાર	વપરાશકર્તાઓને PIN બદલવાની મંજૂરી આપો

- સુરક્ષા: તમામ વ્યવહારો માટે ઓથેન્ટિકેશનની જરૂર છે
- વેલિડેશન: ઉપાડ પહેલાં પૂરતું બેલેન્સ તપાસો
- લોગિં: ઓડિટ માટે તમામ વ્યવહારો રેકૉર્ડ કરો

મેમરી ટ્રીક

“ATMs Authenticate, Balance, Cash, Transfer, History, PIN: મુખ્ય ATM કાર્યો.”

પ્રશ્ન 4(c) [7 ગુણ]

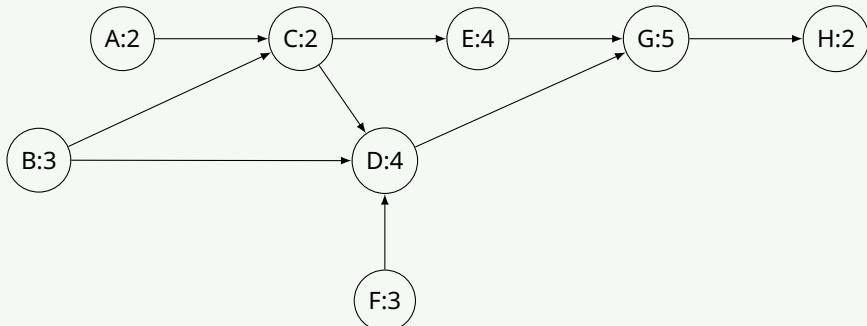
State the use of activity network diagram. Develop activity network diagram for the following system and find the critical path for the same.

જવાબ

એક્ટિવિટી નેટવર્ક ડાયાગ્રામ ઉપયોગો:

- પ્રોજેક્ટ શિજ્યુલિંગ: પ્રોજેક્ટ સમયરેખા નક્કી કરો
- ક્રિટિકલ પાથ ઓળખ: ન્યૂનતમ પ્રોજેક્ટ અવધિ નક્કી કરતો સૌથી લાંબો માર્ગ શોધો
- સંસાધન આયોજન: સંસાધન ફાળવણીને શ્રેષ્ઠ બનાવો

એક્ટિવિટી નેટવર્ક ડાયાગ્રામ:



આકૃતિ 6. એક્ટિવિટી નેટવર્ક ડાયાગ્રામ

ક્રિટિકલ પાથ વિશ્લેષણ:

કોષ્ટક 19. પાથ વિશ્લેષણ

પાથ	પ્રવૃત્તિઓ	અવધિ	ક્રિટિકલ?
A-C-E-G-H	A→C→E→G→H	$2+2+4+5+2 = 15$	ના
B-C-E-G-H	B→C→E→G→H	$3+2+4+5+2 = 16$	હા
A-C-D-G-H	A→C→D→G→H	$2+2+4+5+2 = 15$	ના

- ક્રિટિકલ પાથ: B→C→E→G→H (16 દિવસ)
- પ્રોજેક્ટ અવધિ: 16 દિવસ

મેમરી ટ્રીક

“Networks Navigate Project Paths: નેટવર્ક દાયાગ્રામસનું મહત્વ.”

પ્રશ્ન 4(a OR) [3 ગુણ]

Explain any three requirement gathering activities.

જવાબ

જરૂરિયાત એક્ટ્રીકરણ પ્રવૃત્તિઓ:

કોષ્ટક 20. એક્ટ્રીકરણ પ્રવૃત્તિઓ

પ્રવૃત્તિ	વર્ણન	આઉટપુટ
સ્ટેકહોલ્ડર ઇન્ટરવ્યુ	વપરાશકર્તાઓ અને ગ્રાહકો સાથે સીધી ચર્ચા	ઇન્ટરવ્યુ નોંધો, જરૂરિયાતોની સૂચિ
પ્રશ્નાવલિ	મોટા વપરાશકર્તા જૂથો માટે માળખાગત પ્રશ્નો	સર્વે જવાબો, આંકડાકીય ડેટા
દસ્તાવેજ વિશ્લેષણ	હાલના સિસ્ટમ દસ્તાવેજુકરણની સમીક્ષા	વર્તમાન સિસ્ટમ સમજ

- હેતુ: વપરાશકર્તાની જરૂરિયાતો અને સિસ્ટમ અપેક્ષાઓને સમજવી
- સહભાગીઓ: વપરાશકર્તાઓ, ગ્રાહકો, લોમેન નિષ્ણાતો, ડેવલપર્સ
- દસ્તાવેજુકરણ: SRS દસ્તાવેજમાં રેકૉર્ડ થયેલ તમામ તારણો

મેમરી ટ્રીક

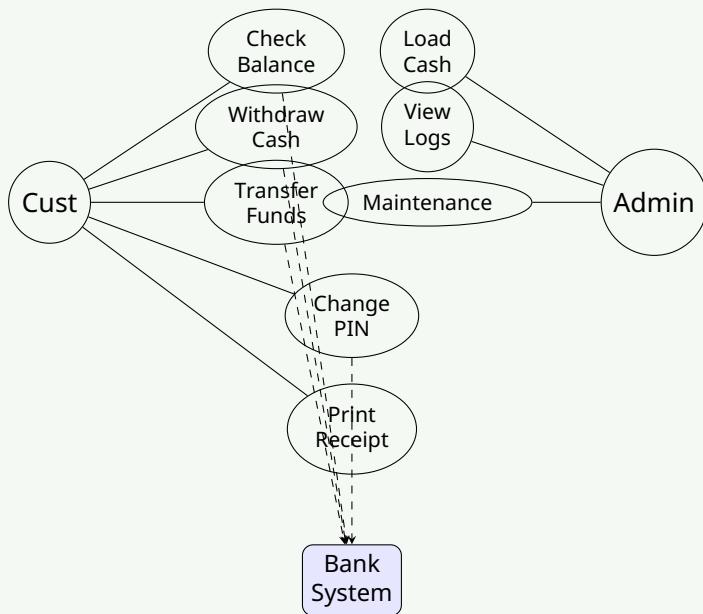
“Interviews, Questions, Documents Gather Requirements: એક્ટ્રીકરણ માટેની તકનીકો.”

પ્રશ્ન 4(b OR) [4 ગુણ]

Develop use case diagram for Bank ATM system.

જવાબ

ATM ચુંઝ કેસ દાયાગ્રામ:



આકૃતિ 7. ATM યુઝ કેસ ડાયગ્રામ

યુઝ કેસ વિગતો:

કોષ્ટક 21. યુઝ કેસ સારાંશ

એક્ટર	યુઝ કેસો
ગ્રાહક	બેલેન્સ તપાસો, રોકડ ઉપાડો, ફંડ ટ્રાન્સફર કરો, PIN બદલો
એડમિન	રોકડ લોડ કરો, લોગ જુઓ, સિસ્ટમ જાળવણી
બેંક સિસ્ટમ	એકાઉન્ટ્સ માન્ય કરો, વ્યવહારો પર પ્રક્રિયા કરો

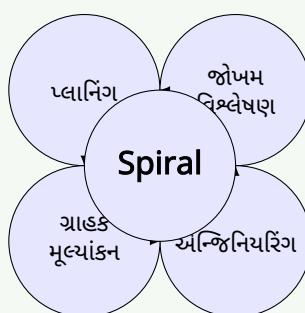
મેમરી ટ્રીક

“Customers Use ATMs, Admins Maintain Systems: એકર્ટર્સ અને ભૂમિકાઓ.”

પ્રશ્ન 4(c OR) [7 ગુણ]

Draw the figure of spiral model. Explain it in brief.

જવાબ



આકૃતિ 8. સ્પાઇરલ મોડેલ ચતુર્થાંશ

સ્પાઇરલ મોડેલ લાક્ષણિકતાઓ:

કોષ્ટક 22. સ્પાઇરલ ચતુર્થંશ

ચતુર્થંશ	પ્રવૃત્તિ	હેતુ
પ્લાનિંગ	ઉદ્દેશ્યો, વિકલ્પો વ્યાખ્યાયિત કરો	પુનરાવર્તન માટે લક્ષ્યો સેટ કરો
જોખમ વિશ્લેષણ	જોખમો ઓળખો અને ઉકેલો	પ્રોજેક્ટ જોખમો ઓછા કરો
એન્જિનિયરિંગ	ઉત્પાદન વિકસાવો અને પરીક્ષણ કરો	કાર્યકારી સોફ્ટવેર બનાવો
મૂલ્યાંકન	ગ્રાહક આકારણી	વપરાશકર્ત્તા પ્રતિસાદ મેળવો

- મુખ્ય લક્ષણ: પુનરાવર્તિત વિકાસ સાથે જોખમ-સંચાલિત અભિગમ
- શ્રેષ્ઠ માટે: મોટા, જટિલ, ઉચ્ચ જોખમવાળા પ્રોજેક્ટ્સ
- ફાયદા: જોખમ વ્યવસ્થાપન, લવચીક, વધતો વિકાસ
- ગેરફાયદા: જટિલ સંચાલન, ખર્ચીન, જોખમ નિપુણતાની જરૂર છે

મેમરી ટ્રીક

“Spirals Plan, Risk, Engineer, Evaluate: ચાર ચતુર્થંશ.”

પ્રશ્ન 5(a) [3 ગુણ]

State TRUE or FALSE for the following.

જવાબ

કોષ્ટક 23. સાચું કે ખોટું

નિવેદન	જવાબ	સમજૂતી
Activity network diagram used to determine critical path	TRUE	પ્રવૃત્તિ નેટવર્કર્સનો પ્રાથમિક હેતુ
In CPM, the shortest path is the critical path	FALSE	સૌથી લાંબો રસ્તો જટિલ માર્ગ છે
Risk avoidance is the best technique to solve risks	FALSE	શ્રેષ્ઠ તકનીક જોખમ પ્રકાર પર આધારિત છે

મેમરી ટ્રીક

“True Networks, False Shortest, False Best”

પ્રશ્ન 5(b) [4 ગુણ]

Identify the differences between traditional model approach and agile approach. (at least 4 differences)

જવાબ

પરંપરાગત વિ. Agile સરખામણી:

કોષ્ટક 24. પરંપરાગત વિ. Agile

પાસું	પરંપરાગત	Agile
પ્લાનિંગ	વ્યાપક અપફન્ટ પ્લાનિંગ	અનુકૂલનશીલ આયોજન
દસ્તાવેજુકરણ	ભારે દસ્તાવેજુકરણ	ન્યૂનતમ દસ્તાવેજુકરણ
ગ્રાહક સંડોવણી	જરૂરિયાતોના તબક્કા સુધી મર્યાદિત	સતત સંડોવણી
ફેરફાર હેન્ડલિંગ	મુશ્કેલ અને ખર્ચાળ	ફેરફાર સ્વીકારે છે
ડિલિવરી	સિંગલ ફાઇનલ ડિલિવરી	વારંવાર ઇન્ફ્રાસ્ટ્રક્ચર ડિલિવરી
પ્રક્રિયા	પ્રક્રિયા-સંચાલિત	લોકો-સંચાલિત

- **પરંપરાગત:** આગાહીયુક્ત, કંબિક અભિગમ
- **Agile:** અનુકૂલનશીલ, પુનરાવર્તિત અભિગમ
- **સુગમતા:** Agile બદલાતો જરૂરિયાતો માટે વધુ પ્રતિભાવશીલ

મેમરી ટ્રીક

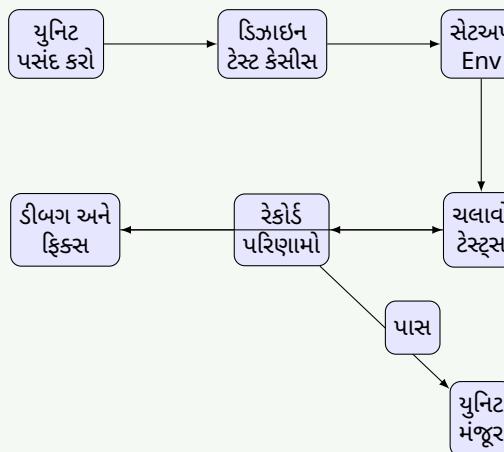
“Traditional Plans Heavy, Agile Adapts Light: મુખ્ય તફાવત ફિલસ્ફૂર્ઝી.”

પ્રશ્ન 5(c) [7 ગુણ]

Define unit testing. Draw the figure of it. Explain the process of unit testing.

જવાબ

યુનિટ ટેસ્ટિંગ વ્યાખ્યા: વ્યક્તિગત સોફ્ટવેર ઘટકો અથવા મોડ્યુલોનું અલગતામાં પરીક્ષણ કરવું જેથી તેઓ ડિઝાઇન સ્પષ્ટીકરણો અનુસાર યોગ્ય રીતે કાર્ય કરે છે તેની ખાતરી કરી શકાય.



આકૃતિ 9. યુનિટ ટેસ્ટિંગ પ્રક્રિયા

યુનિટ ટેસ્ટિંગ પ્રક્રિયાના પગલાં:

કોષ્ટક 25. યુનિટ ટેસ્ટિંગ પગલાં

પગલું	પ્રવૃત્તિ	હેતુ
ટેસ્ટ પ્લાનિંગ	પરીક્ષણ કરવા માટે એકમો ઓળખો	પરીક્ષણ અવકાશ વ્યાખ્યાપિત કરો
ટેસ્ટ ડિઝાઇન	ટેસ્ટ કેસ બનાવો	બધા કોડ પાથ આવરી લો
ટેસ્ટ સેટઅપ	ટેસ્ટ પર્યાવરણ તૈયાર કરો	ટેસ્ટ હેઠળના એકમને અલગ કરો
ટેસ્ટ અમલીકરણ	ટેસ્ટ કેસ ચલાવો	યુનિટ વર્તાણુક ચકાસો
પરિણામ વિશ્લેષણ	પરિણામોનું મૂલ્યાંકન કરો	ખામીઓ ઓળખો
ખામી ફિક્સિંગ	મળી આવેલી સમસ્યાઓ સુધારો	યુનિટ ગુણવત્તા ખાતરી કરો

- ફાયદા: પ્રારંભિક ખામી શોધ, સરળ ડીબર્ગિંગ, સુધારેલી કોડ ગુણવત્તા
- સાધનો: JUnit, NUnit, એટોમેટેડ ટેસ્ટિંગ ફેમવર્ક
- કવરેજ: ઉચ્ચ કોડ કવરેજ માટે લક્ષ્ય રાખો (સ્ટેટમેન્ટ્સ, બ્રાન્ચીસ, પાથ્સ)

મેમરી ટ્રીક

“Units Test Individual Components Thoroughly: યુનિટ ટેસ્ટિંગનો અર્થ.”

પ્રશ્ન 5(a OR) [3 ગુણ]

Give the full form of the following.

જવાબ

કોષ્ટક 26. પૂર્ણ સ્વરૂપો

સંક્ષેપ	પૂર્ણ સ્વરૂપ
AOA	Activity On Arrow
PERT	Program Evaluation and Review Technique
EVA	Earned Value Analysis
CPM	Critical Path Method
WBS	Work Breakdown Structure
PMC	Project Management and Control

મેમરી ટ્રીક

“Activities On Arrows, Programs Evaluate Review Techniques, Earned Values Analyzed, Critical Paths Managed, Work Broken Structured, Projects Managed Controlled”

પ્રશ્ન 5(b OR) [4 ગુણ]

Explain code inspection.

જવાબ

કોડ ઈન્સ્પેક્શન એ ખામીઓને ઓળખવા અને ગુણવત્તાના ધોરણોની ખાતરી કરવા માટે ટીમના સભ્યો દ્વારા સોર્સ કોડની વ્યવસ્થિત તપાસ છે.

કોષ્ટક 27. ઈન્સ્પેક્શન પ્રક્રિયા

તબક્કો	પ્રવૃત્તિ	સહભાગીઓ
પ્લાનિંગ	ઇન્સ્પેક્શન મીટિંગ સુનિશ્ચિત કરો	મોડરેટર
તૈયારી	વ્યક્તિગત રીતે કોડની સમીક્ષા કરો	બધા નિરીક્ષકો
ઇન્સ્પેક્શન મીટિંગ	તારણોની ચર્ચા કરો	ટીમના સભ્યો
રીવર્ક	ઓળખાયેલ સમસ્યાઓ સુધારો	લેખક
ફોલો-અપ	સુધારાઓ ચકાસો	મોડરેટર

- ફાયદા: પ્રારંભિક ખામી શોધ, જ્ઞાન વહેંચણી, સુધારેલી કોડ ગુણવત્તા
- ભૂમિકાઓ: લેખક, મોડરેટર, સમીક્ષક, રેકૉર્ડર
- ફીક્સ વિસ્તારો: લોજિક ભૂલો, કોડિંગ ધોરણો, જાળવણીક્ષમતા

મેમરી ટ્રીક

“Inspections Improve Code Quality: ઈન્સ્પેક્શનનો હેતુ.”

પ્રશ્ન 5(c OR) [7 ગુણ]

Define white box testing method. Explain different white box testing methods.

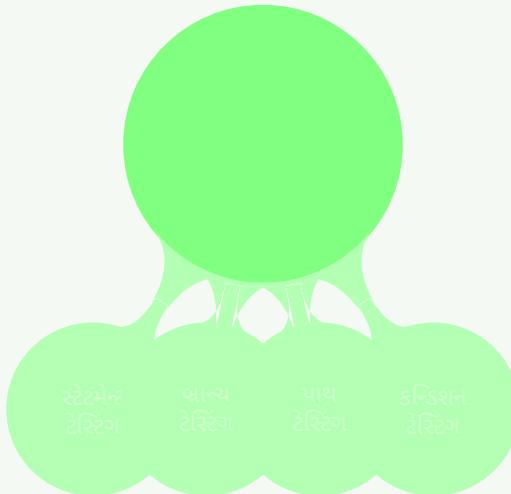
જવાબ

વહાઈટ બોક્સ ટેસ્ટિંગ વ્યાખ્યા: ટેસ્ટિંગ પદ્ધતિ જે સંપૂર્ણ કવરેજ સુનિશ્ચિત કરવા માટે આંતરિક કોડ સ્ટ્રક્ચર, લોજિક પાથ અને અમલીકરણ વિગતોની તપાસ કરે છે.

વહાઈટ બોક્સ ટેસ્ટિંગ પદ્ધતિઓ:

કોષ્ટક 28. વહાઈટ બોક્સ પદ્ધતિઓ

પદ્ધતિ	વર્ણન	કવરેજ ફોક્સ
સ્ટેટમેન્ટ કવરેજ	દરેક નિવેદનને અમલમાં મૂકો	બધા કોડ લાઇન
બ્રાન્ચ કવરેજ	તમામ નિર્ણય પરિણામોનું પરીક્ષણ કરો	જો-તો શરતો
પાથ કવરેજ	તમામ સંભવિત માર્ગો ચલાવો	સંપૂર્ણ અમલીકરણ પ્રવાહ
કન્ડિશન કવરેજ	તમામ શરત સંયોજનોનું પરીક્ષણ કરો	બુલિયન અભિવ્યક્તિઓ



આકૃતિ 10. વહાઈટ બોક્સ ટેસ્ટિંગ તકનીકો

- કવરેજ વિશ્લેષણ: ટેસ્ટિંગની અસરકારકતા માપે છે (દા.ત., એક્ઝિક્યુટેડ સ્ટેટમેન્ટ્સ / કુલ સ્ટેટમેન્ટ્સ)
- ફૂથદા: સંપૂર્ણ ટેસ્ટિંગ, મૃત કોડ ઓળખે છે, ગુણવત્તા સુનિશ્ચિત કરે છે
- ગરફાયદા: કોડ જ્ઞાનની જરૂર, સમય લેતું

મેમરી ટ્રીક

“White Box Sees Inside Code Structure: મુખ્ય ઘ્યાલ.”