

Fundamentals of Software Development (4331604) - શિયાળું 2023 ઉકેલ

Milav Dabgar

January 20, 2024

પ્રશ્ન 1(a) [3 ગુણ]

Define Software and explain its characteristics.

જવાબ

સૉફ્ટવેર એ પ્રોગ્રામ્સ, સૂચનાઓ અને દસ્તાવેજીકરણનો સંગ્રહ છે જે કમ્પ્યુટર સિસ્ટમ પર કાર્યો કરે છે.
મુખ્ય લક્ષણો:

કોષ્ટક 1. સૉફ્ટવેર લક્ષણો

લક્ષણ	વર્ણન
અમૂર્ત	શારીરિક રીતે સ્પર્શ કરી શકાતું નથી
તાર્કિક	વ્યવસ્થિત અભિગમ દ્વારા બનાવાયેલ
ઉત્પાદિત	પરંપરાગત રીતે ઉત્પન્ન નહીં, વિકસિત
જટિલ	અંતર્ગત જટિલ માળખું ધરાવે છે

મેમરી ટ્રીક

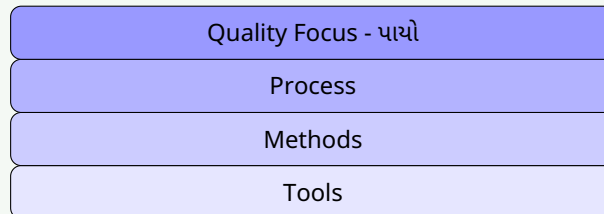
“અમૂર્ત તાર્કિક ઉત્પાદન જટિલતા”

પ્રશ્ન 1(b) [4 ગુણ]

Write a note on Software engineering – A layered technology.

જવાબ

Software Engineering એ સ્તરીય ટેકનોલોજી તરીકે માળખાકીય છે જ્યાં દરેક સ્તર આગામી સ્તરને આધાર આપે છે.
સ્તરીય માળખું:



આકૃતિ 1. Software Engineering Layers

કોષ્ટક 2. સ્તર વર્ણન

સ્તર	હેતુ	વર્ણન
Quality Focus	પાયો	ગુણવત્તાયુક્ત ઉત્પાદનો પહોંચાડવાનો ભાર
Process	ફ્રેમવર્ક	સોફ્ટવેર ડેવલપમેન્ટ કેવી રીતે કરવું તે નક્કી કરે છે
Methods	તકનીકો	પ્રવૃત્તિઓ કરવાની વિશિષ્ટ પદ્ધતિઓ
Tools	સ્વચાલન	પદ્ધતિઓને આધાર આપતું સોફ્ટવેર

મેમરી ટ્રીક

“ટૂલ્સ મેથડ્સ પ્રોસેસ ક્વોલિટી”

પ્રશ્ન 1(c) [7 ગુણ]

Explain Software Process framework and umbrella activities.

જવાબ

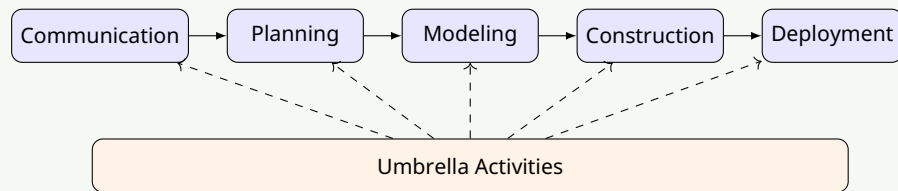
સોફ્ટવેર પ્રોસેસ ફ્રેમવર્ક સોફ્ટવેર ડેવલપમેન્ટ માટે મુખ્ય પ્રવૃત્તિઓ અને umbrella પ્રવૃત્તિઓ સાથે માળખું પ્રદાન કરે છે.
ફ્રેમવર્ક પ્રવૃત્તિઓ:

કોષ્ટક 3. ફ્રેમવર્ક પ્રવૃત્તિઓ

પ્રવૃત્તિ	હેતુ	મુખ્ય કાર્યો
Communication	આવશ્યકતાઓ સમજવી	હિસ્સેદારો સાથે વાતચીત, આવશ્યકતા એકત્રીકરણ
Planning	રોડમેપ બનાવવો	અંદાજ, શેડ્યૂલિંગ, જોખમ મૂલ્યાંકન
Modeling	બ્લુપ્રિન્ટ બનાવવા	વિશ્લેષણ અને ડિઝાઇન મોડેલ્સ
Construction	સોફ્ટવેર બનાવવું	કોડિંગ અને ટેસ્ટિંગ
Deployment	વપરાશકર્તાઓને પહોંચાડવું	ઇન્સ્ટોલેશન, સપોર્ટ, ફીડબેક

Umbrella પ્રવૃત્તિઓ:

- **Software project tracking:** પ્રગતિ નિરીક્ષણ અને ગુણવત્તા નિયંત્રણ
- **Risk management:** સંભવિત સમસ્યાઓ ઓળખવી અને ઘટાડવી
- **Quality assurance:** ધોરણો પૂરા થાય તેની ખાતરી કરવી
- **Configuration management:** ફેરફારોને વ્યવસ્થિત રીતે નિયંત્રિત કરવા
- **Work product preparation:** ડિલિવરેબલ દસ્તાવેજો બનાવવા



આકૃતિ 2. Process Framework

મેમરી ટ્રીક

“કોમ્યુનિકેશન પ્લાનિંગ મોડેલિંગ કન્સ્ટ્રક્શન ડિપ્લોયમેન્ટ” “ટ્રેક રિસ્ક ક્વોલિટી કન્ફિગરેશન વર્ક”

પ્રશ્ન 1(c OR) [7 ગુણ]

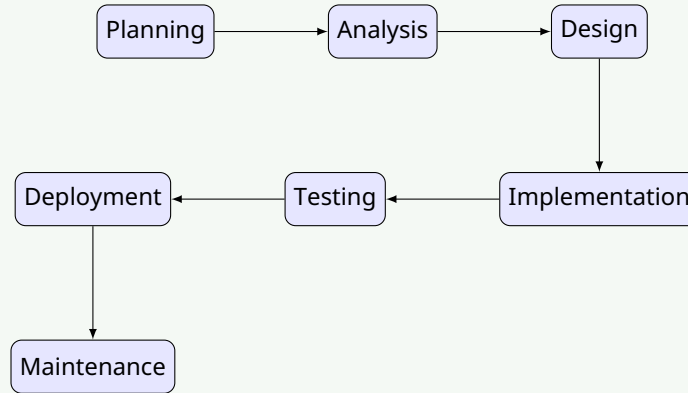
Define SDLC and explain each phase.

જવાબ

SDLC (Software Development Life Cycle) એ સોફ્ટવેર એપ્લિકેશન્સ વિકસાવવા માટેની વ્યવસ્થિત પ્રક્રિયા છે.
SDLC તબક્કાઓ:

કોષ્ટક 4. SDLC તબક્કાઓ

તબક્કો	હેતુ	મુખ્ય પ્રવૃત્તિઓ	ડિલિવેરેબલ્સ
Planning	અવકાશ નક્કી કરવો	શક્યતા અભ્યાસ, સંસાધન ફાળવણી	પ્રોજેક્ટ પ્લાન
Analysis	આવશ્યકતાઓ એકત્રિત કરવી	આવશ્યકતા સંગ્રહ, દસ્તાવેજીકરણ	SRS દસ્તાવેજ
Design	આર્કિટેક્ચર બનાવવું	સિસ્ટમ ડિઝાઇન, ડેટાબેસ ડિઝાઇન	ડિઝાઇન દસ્તાવેજો
Implementation	કોડ લખવો	પ્રોગ્રામિંગ, યુનિટ ટેસ્ટિંગ	સોર્સ કોડ
Testing	ગુણવત્તા ચકાસવી	સિસ્ટમ ટેસ્ટિંગ, બગ ફિક્સિંગ	ટેસ્ટ રિપોર્ટ્સ
Deployment	સોફ્ટવેર રિલીઝ કરવું	ઇન્સ્ટોલેશન, યુઝર ટ્રેનિંગ	લાઇવ સિસ્ટમ
Maintenance	ચાલુ સપોર્ટ	બગ ફિક્સ, એન્હાન્સમેન્ટ્સ	અપડેટેડ સિસ્ટમ



આકૃતિ 3. SDLC Phases

મેમરી ટ્રીક

“પ્લાન એનાલિસિસ ડિઝાઇન ઇમ્પ્લિમેન્ટેશન ટેસ્ટિંગ ડિપ્લોયમેન્ટ મેઇન્ટેનન્સ”

પ્રશ્ન 2(a) [3 ગુણ]

Describe advantage disadvantage of prototype model.

જવાબ

Prototype Model વિશ્લેષણ:

કોષ્ટક 5. Prototype Model ફાયદા/નુકસાન

ફાયદા	નુકસાન
વહેલો ફીડબેક વપરાશકર્તાઓ તરફથી	સમય વાપરતું ડેવલપમેન્ટ પ્રોસેસ
ઓછું જોખમ નિષ્ફળતાનું	ખર્ચમાં વધારો પુનરાવર્તન કારણે
બહેતર સમજ આવશ્યકતાઓની	Scope creep થઈ શકે છે

મેમરી ટ્રીક

“વહેલો ઓછું બહેતર વિરુદ્ધ સમય ખર્ચ સ્કોપ”

પ્રશ્ન 2(b) [4 ગુણ]

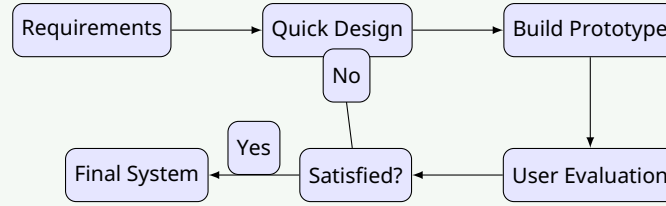
Explain Prototyping Model and justify when to use with example.

જવાબ

Prototyping Model વિકાસ પ્રક્રિયાની શરૂઆતમાં સોફ્ટવેરનું કાર્યશીલ મોડલ બનાવે છે.
ક્યારે ઉપયોગ કરવો:

કોષ્ટક 6. ઉપયોગના ઉદાહરણો

સ્થિતિ	ઉદાહરણ	જસ્ટિફિકેશન
અસ્પષ્ટ આવશ્યકતાઓ	ઓનલાઇન શોપિંગ કાર્ટ	યુઝર ઇન્ટરફેસને સુધારવાની જરૂર
નવી ટેકનોલોજી	મોબાઇલ બેંકિંગ એપ	શક્તિ પરીક્ષણ જરૂરી
યુઝર ઇન્ટરેક્શન જટિલ	ગેમિંગ એપ્લિકેશન	યુઝર અનુભવ ચકાસણી જરૂરી



આકૃતિ 4. Prototyping Process

મેમરી ટ્રીક

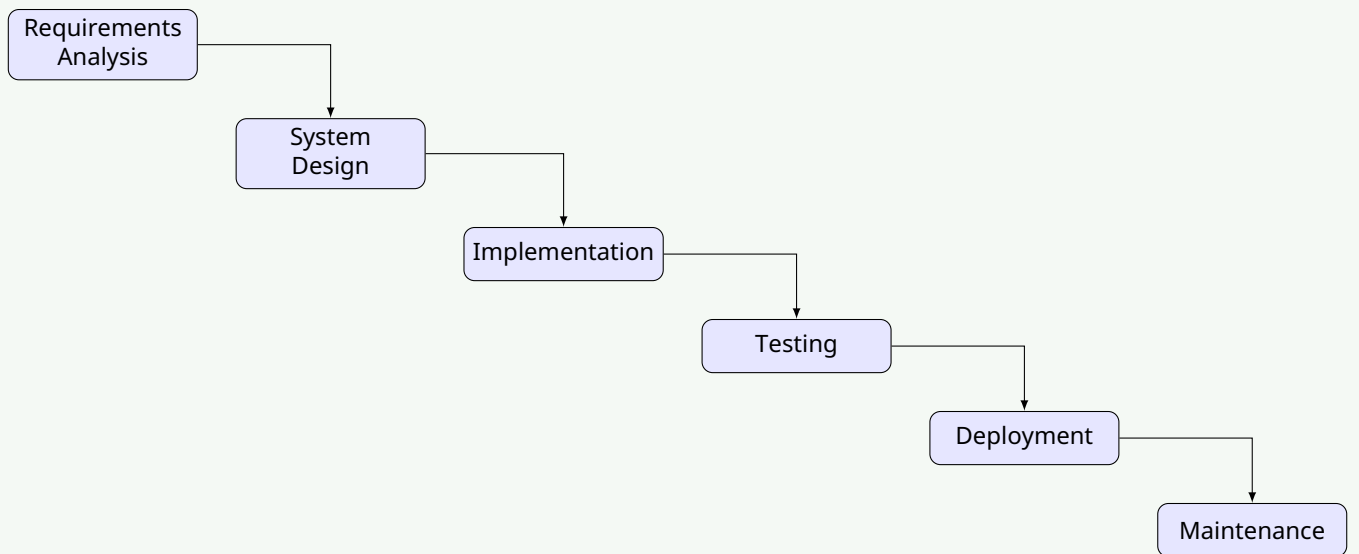
“આવશ્યકતા ઝડપી બિલ્ડ યુઝર સંતુષ્ટ ફાઇનલ”

પ્રશ્ન 2(c) [7 ગુણ]

Sketch and discuss (I) Waterfall model & (II) Incremental Model.

જવાબ

(I) **Waterfall Model:** રેખીય ક્રમિક અભિગમ જ્યાં દરેક તબક્કો આગલા તબક્કા પહેલાં પૂર્ણ થવો જોઈએ.

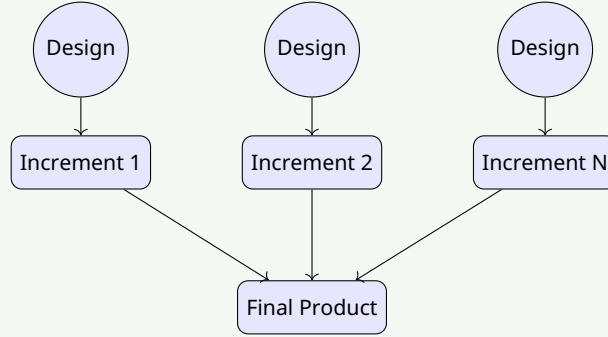


આકૃતિ 5. Waterfall Model

કોષ્ટક 7. Waterfall લક્ષણો

લક્ષણો	વર્ણન
ક્રમિક	એક સમયે એક તબક્કો
દસ્તાવેજીકરણ આધારિત	ભારે દસ્તાવેજીકરણ
યોગ્ય	સ્પષ્ટ આવશ્યકતાઓ માટે

(II) Incremental Model: નાના increments માં વિકાસ જ્યાં દરેક increment કાર્યક્ષમતા ઉમેરે છે.



આકૃતિ 6. Incremental Model Concept

કોષ્ટક 8. ઉપયોગ સરખામણી

લક્ષણ	Waterfall	Incremental
લવચીકતા	ઓછી	વધુ
જોખમ	વધુ	ઓછું
ડિલિવરી	પ્રોજેક્ટના અંતે	બહુવિધ ડિલિવરીઓ

મેમરી ટ્રીક

“વોટર એકવાર પડે, ઇન્ક્રિમેન્ટ બહુવિધ બનાવે”

પ્રશ્ન 2(a OR) [3 ગુણ]

Describe advantage and disadvantage of Incremental Model.

જવાબ

Incremental Model વિશ્લેષણ:

કોષ્ટક 9. Incremental Model ફાયદા/નુકસાન

ફાયદા	નુકસાન
વહેલી ડિલિવરી કાર્યશીલ સોફ્ટવેરની	કુલ ખર્ચ વધુ હોઈ શકે
સરળ ટેસ્ટિંગ નાના increments ની	સિસ્ટમ આર્કિટેક્ચર સમસ્યાઓ
ઓછું જોખમ વહેલા ફીડબેક દ્વારા	મેનેજમેન્ટ જટિલતા વધે છે

મેમરી ટ્રીક

“વહેલી સરળ ઓછું વિરુદ્ધ કુલ સિસ્ટમ મેનેજમેન્ટ”

પ્રશ્ન 2(b OR) [4 ગુણ]

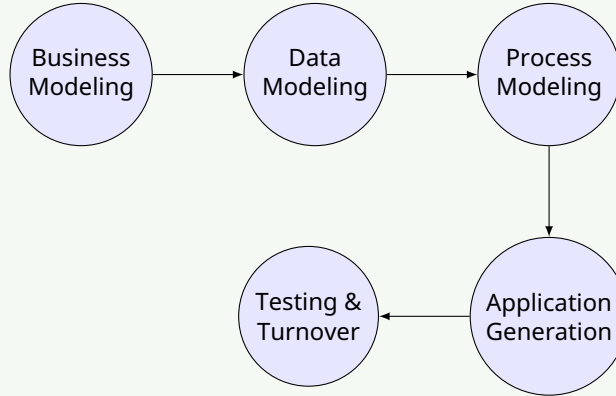
Write concept of Rapid Application Development (RAD) and explain it.

જવાબ

RAD યોજના અને ટેસ્ટિંગ કરતાં ઝડપી prototyping અને ત્વરિત ફીડબેક પર ભાર મૂકે છે.
RAD ઘટકો:

કોષ્ટક 10. RAD પ્રક્રિયા

તબક્કો	અવધિ	પ્રવૃત્તિઓ	આઉટપુટ
Business Modeling	ટૂંકી	માહિતી પ્રવાહ નક્કી કરવો	બિઝનેસ આવશ્યકતાઓ
Data Modeling	ટૂંકી	ડેટા ઓબ્જેક્ટ્સ નક્કી કરવા	ડેટા મોડલ્સ
Process Modeling	ટૂંકી	પ્રોસેસિંગ functions નક્કી કરવા	પ્રોસેસ વર્ણનો
Application Generation	ટૂંકી	ટૂલ્સ વાપરીને બનાવવું	કાર્યશીલ એપ્લિકેશન
Testing & Turnover	ટૂંકી	ટેસ્ટ અને ડિલિવર કરવું	ફાઇનલ સિસ્ટમ



આકૃતિ 7. RAD Flow

મેમરી ટ્રીક

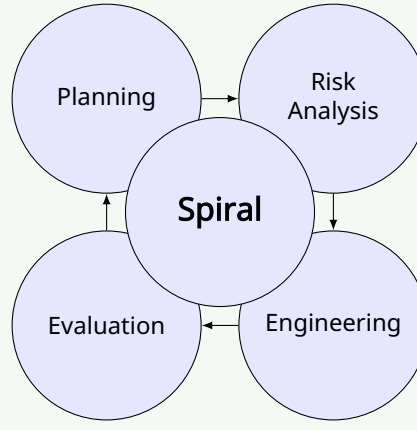
“બિઝનેસ ડેટા પ્રોસેસ એપ્લિકેશન ટેસ્ટિંગ”

પ્રશ્ન 2(c OR) [7 ગુણ]

Design and describe Spiral Model and give advantage and disadvantage.

જવાબ

Spiral Model પુનરાવર્તક વિકાસને વ્યવસ્થિત જોખમ વિશ્લેષણ સાથે જોડે છે.



આકૃતિ 8. Spiral Model Quadrants

Spiral ચતુર્થાંશ:

કોષ્ટક 11. ચતુર્થાંશ વિગતો

ચતુર્થાંશ	પ્રવૃત્તિ	હેતુ
Planning	લક્ષ્ય સેટિંગ	આવશ્યકતાઓ અને અવરોધો નક્કી કરવા
Risk Analysis	જોખમ મૂલ્યાંકન	જોખમો ઓળખવા અને ઉકેલવા
Engineering	વિકાસ	ઉત્પાદન બનાવવું અને ટેસ્ટ કરવું
Evaluation	ગ્રાહક મૂલ્યાંકન	પરિણામો મૂલ્યાંકન અને આગલા iteration ની યોજના

ફાયદા વિરુદ્ધ નુકસાન:

કોષ્ટક 12. Spiral ફાયદા/નુકસાન

ફાયદા	નુકસાન
ઉચ્ચ જોખમ પ્રોજેક્ટ્સ સારી રીતે હેન્ડલ થાય	જટિલ મેનેજમેન્ટ જરૂરી
મોટી એપ્લિકેશન્સ માટે સારું	નાના પ્રોજેક્ટ્સ માટે મોંઘું
ગ્રાહક સામેલ આપા દરમિયાન	જોખમ વિશ્લેષણ કુશળતા જરૂરી

મેમરી ટ્રીક

“પ્લાન રિસ્ક એન્જિનિયર ઇવેલ્યુએટ + ઉચ્ચ સારું ગ્રાહક વિરુદ્ધ જટિલ મોંઘું જોખમ”

પ્રશ્ન 3(a) [3 ગુણ]

Illustrate importance of SRS

જવાબ

SRS (Software Requirements Specification) એ સોફ્ટવેર ડેવલપમેન્ટ માટે મહત્વપૂર્ણ પાયાનું દસ્તાવેજ છે.
મહત્વ કોષ્ટક:

કોષ્ટક 13. SRS મહત્વ

પાસું	મહત્વ	ફાયદો
કોમ્યુનિકેશન	હિસ્સેદારોની સમજ	સ્પષ્ટ અપેક્ષાઓ
કરાર	કાનૂની સમજૂતી	વિવાદ નિરાકરણ
ટેસ્ટિંગ આધાર	ચકાસણી માપદંડ	ગુણવત્તા ખાતરી

મેમરી ટ્રીક

“કોમ્યુનિકેશન કરાર ટેસ્ટિંગ”

પ્રશ્ન 3(b) [4 ગુણ]

Specify characteristics of good & bad SRS

જવાબ

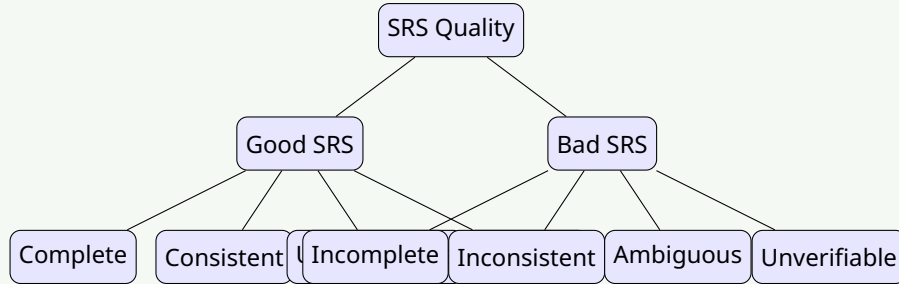
SRS ગુણવત્તા લક્ષણો:

કોષ્ટક 14. સારો vs ખરાબ SRS

સારો SRS	ખરાબ SRS
સંપૂર્ણ - બધી આવશ્યકતાઓ આવરી લેવાયેલ	અધૂરો - આવશ્યકતાઓ ખૂટે છે
સુસંગત - કોઈ વિરોધાભાસ નથી	અસંગત - વિરોધી નિવેદનો
અસ્પષ્ટ નહીં - સ્પષ્ટ અર્થ	અસ્પષ્ટ - બહુવિધ અર્થઘટન
ચકાસી શકાય તેવું - ટેસ્ટ કરી શકાય	ચકાસી ન શકાય - વેલિડેટ કરી શકાતું નથી

વધારાના સારા લક્ષણો:

- સુધારી શકાય તેવું: બદલવું અને જાળવવું સરળ
- ટ્રેસેબલ: સ્ત્રોત અને ડિઝાઇન સાથે લિંક



આકૃતિ 9. SRS લક્ષણો

મેમરી ટ્રીક

“સંપૂર્ણ સુસંગત અસ્પષ્ટ-ન ચકાસી-શકાય વિરુદ્ધ અધૂરો અસંગત અસ્પષ્ટ ચકાસી-ન-શકાય”

પ્રશ્ન 3(c) [7 ગુણ]

Classify Types of Requirements in SRS

જવાબ

સોફ્ટવેર આવશ્યકતાઓને બે મુખ્ય શ્રેણીઓમાં વર્ગીકૃત કરવામાં આવે છે.

(i) **Functional Requirements:** સિસ્ટમે શું કરવું જોઈએ તે નક્કી કરે છે - વિશિષ્ટ વર્તણૂકો અને કાર્યો.

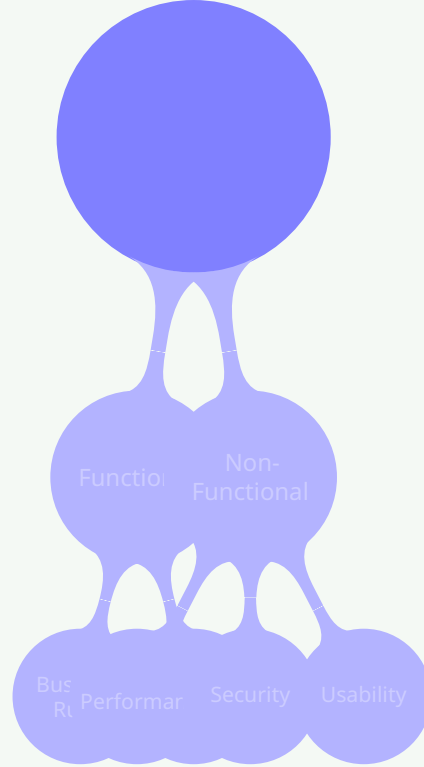
કોષ્ટક 15. Functional Requirements

પ્રકાર	વર્ણન	ઉદાહરણ
બિઝનેસ નિયમો	મુખ્ય બિઝનેસ લોજિક	"આવકના સ્લેબ મુજબ ટેક્સ ગણતરી કરવી"
યુઝર એક્શન્સ	સિસ્ટમ પ્રતિભાવો	"યુઝરનેમ/પાસવર્ડ સાથે લોગિન"
ડેટા પ્રોસેસિંગ	માહિતી હેન્ડલિંગ	"માસિક વેચાણ રિપોર્ટ જનરેટ કરવી"
એક્સટર્નલ ઇન્ટરફેસ	સિસ્ટમ ક્રિયાપ્રતિક્રિયાઓ	"પેમેન્ટ ગેટવે સાથે કનેક્ટ કરવું"

(ii) **Non-functional Requirements:** સિસ્ટમે કેવી રીતે પ્રદર્શન કરવું જોઈએ તે નક્કી કરે છે - ગુણવત્તા લક્ષણો અને મર્યાદાઓ.

કોષ્ટક 16. Non-functional Requirements

શ્રેણી	આવશ્યકતા	ઉદાહરણ	માપદંડ
પ્રદર્શન	પ્રતિભાવ સમય	"પેજ લોડ < 3 સેકન્ડ"	સમય મેટ્રિક્સ
સુરક્ષા	ડેટા સુરક્ષા	"યુઝર પાસવર્ડ એન્ક્રિપ્ટ કરવા"	સુરક્ષા ધોરણો
વિશ્વસનીયતા	સિસ્ટમ અપટાઇમ	"99.9% ઉપલબ્ધતા"	નિષ્ફળતા દરો
ઉપયોગિતા	યુઝર અનુભવ	"ચેકઆઉટ માટે મહત્તમ 3 ક્લિક"	યુઝર મેટ્રિક્સ
સ્કેલેબિલિટી	વૃદ્ધિ ક્ષમતા	"10,000 યુઝર્સ સપોર્ટ કરવા"	લોડ ક્ષમતા



આકૃતિ 10. Requirement Types

મેમરી ટ્રીક

“Functional = શું, Non-Functional = કેવી રીતે”

પ્રશ્ન 3(a OR) [3 ગુણ]

Describe skill to manage software projects

જવાબ

પ્રોજેક્ટ મેનેજમેન્ટ માટે સફળ સોફ્ટવેર ડિલિવરી માટે વિવિધ કુશળતાઓની જરૂર છે.
આવશ્યક કુશળતાઓ:

કોષ્ટક 17. PM Skills

કુશળતા શ્રેણી	વર્ણન	ઉપયોગ
ટેકનિકલ	ટેકનોલોજીની સમજ	આર્કિટેક્ચર નિર્ણયો
નેતૃત્વ	ટીમ પ્રેરણા	સંઘર્ષ નિરાકરણ
કોમ્યુનિકેશન	હિસ્સેદાર ક્રિયાપ્રતિક્રિયા	સ્થિતિ રિપોર્ટિંગ

મેમરી ટ્રીક

“ટેકનિકલ નેતૃત્વ કોમ્યુનિકેશન”

પ્રશ્ન 3(b OR) [4 ગુણ]

Briefly give the Responsibility of software project Manager.

જવાબ

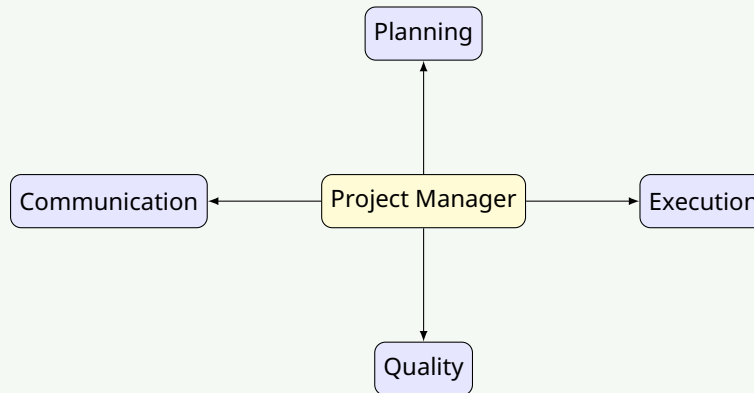
સોફ્ટવેર પ્રોજેક્ટ મેનેજર સમગ્ર પ્રોજેક્ટ લાઇફસાઇકલની દેખરેખ રાખે છે અને સફળ ડિલિવરી સુનિશ્ચિત કરે છે.
મુખ્ય જવાબદારીઓ:

કોષ્ટક 18. PM જવાબદારીઓ

ક્ષેત્ર	જવાબદારી	પ્રવૃત્તિઓ
પ્લાનિંગ	પ્રોજેક્ટ રોડમેપ	શેડ્યૂલ, બજેટ, સંસાધન ફાળવણી
એક્ઝિક્યુશન	ટીમ સંકલન	કાર્ય સોંપણી, પ્રગતિ નિરીક્ષણ
ગુણવત્તા	ધોરણ પાલન	કોડ રિવ્યુ, ટેસ્ટિંગ દેખરેખ
કોમ્યુનિકેશન	હિસ્સેદાર અપડેટ્સ	સ્થિતિ રિપોર્ટ્સ, જોખમ કોમ્યુનિકેશન

વધારાની ફરજો:

- જોખમ વ્યવસ્થાપન: પ્રોજેક્ટ જોખમો ઓળખવા અને ઘટાડવા
- ટીમ ડેવલપમેન્ટ: ટીમ સભ્યોને માર્ગદર્શન અને સંઘર્ષ નિરાકરણ



આકૃતિ 11. PM Functions

મેમરી ટ્રીક

“પ્લાન એક્ટિવિટી ગુણવત્તા કોમ્યુનિકેટ જોખમ ટીમ”

પ્રશ્ન 3(c OR) [7 ગુણ]

Compare PERT chart – Gantt chart side by side.

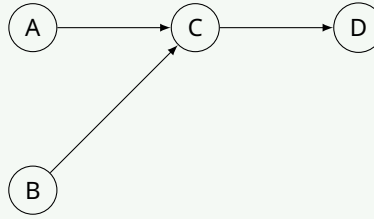
જવાબ

બંને ચાર્ટ પ્રોજેક્ટ મેનેજમેન્ટ ટૂલ્સ છે પરંતુ વિવિધ હેતુઓ સેવે છે અને અલગ લક્ષણો ધરાવે છે.
વિગતવાર સરખામણી:

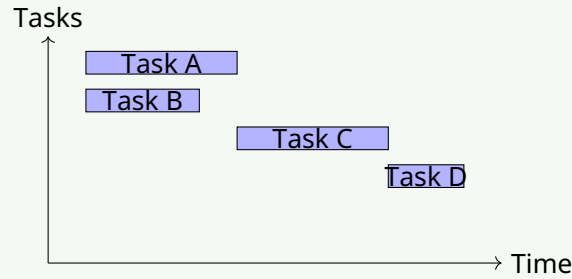
કોષ્ટક 19. PERT vs Gantt

પાસું	PERT Chart	Gantt Chart
હેતુ	કાર્ય અવલંબન દર્શાવવું	પ્રોજેક્ટ ટાઇમલાઇન બતાવવું
માળખું	નેટવર્ક ડાયાગ્રામ	બાર ચાર્ટ
ધ્યાન	ક્રિટિકલ પાથ વિશ્લેષણ	શેડ્યૂલ વિઝ્યુઅલાઇઝેશન
સમય પ્રદર્શન	અંદાજિત અવધિ	વાસ્તવિક તારીખો
અવલંબન	સ્પષ્ટ તીરો	ગર્ભિત જોડાણો
શ્રેષ્ઠ	જટિલ પ્રોજેક્ટ્સ	સરળ શેડ્યૂલિંગ

વિઝ્યુઅલ રિપ્રેઝન્ટેશન:



આકૃતિ 12. PERT Chart ખ્યાલ



આકૃતિ 13. Gantt Chart ખ્યાલ

ક્યારે ઉપયોગ કરવો:

કોષ્ટક 20. ઉપયોગ માર્ગદર્શિકા

સ્થિતિ	PERT	Gantt
પ્રોજેક્ટ પ્રકાર	સંશોધન અને વિકાસ	બાંધકામ, સોફ્ટવેર
અનિશ્ચિતતા	ઉચ્ચ અનિશ્ચિતતા	સ્પષ્ટ કાર્યો
પ્રેક્ષકો	ટેકનિકલ ટીમ	મેનેજમેન્ટ, ક્લાયન્ટ્સ

ફાયદાઓની સરખામણી:

- **PERT:** ક્રિટિકલ પાથ, લવચીક સમય, જોખમ વિશ્લેષણ
- **Gantt:** સમજવામાં સરળ, પ્રગતિ ટ્રેકિંગ, સંસાધન ફાળવણી

મેમરી ટ્રીક

“PERT = પાથ, Gantt = બાર્ડ”

પ્રશ્ન 4(a) [3 ગુણ]

Give steps of Project Monitoring and control process

જવાબ

પ્રોજેક્ટ મોનિટરિંગ વ્યવસ્થિત નિરીક્ષણ અને સુધારાત્મક ક્રિયાઓ દ્વારા પ્રોજેક્ટ ટ્રેક પર રહે તેની ખાતરી કરે છે.
મોનિટરિંગ પગલાં:

કોષ્ટક 21. પ્રક્રિયાના પગલાં

પગલું	પ્રવૃત્તિ	હેતુ
પ્રગતિ ટ્રેક કરવી	વાસ્તવિક વિરુદ્ધ આયોજિત માપવું	વિચલનો ઓળખવા
ગુણવત્તા મૂલ્યાંકન	ડિલિવેરેબલ્સ સમીક્ષા	ધોરણો સુનિશ્ચિત કરવા
પગલાં લેવા	સુધારાઓ લાગુ કરવા	સંરેખણ જાળવવા

મેમરી ટ્રીક

“ટ્રેક મૂલ્યાંકન પગલાં”

પ્રશ્ન 4(b) [4 ગુણ]

Discuss i)Risk Assessment ii)Risk Mitigation

જવાબ

(i) Risk Assessment: સંભવિત પ્રોજેક્ટ જોખમો ઓળખવા અને મૂલ્યાંકન કરવાની પ્રક્રિયા.

કોષ્ટક 22. મૂલ્યાંકન ઘટકો

મૂલ્યાંકન પ્રકાર	પદ્ધતિ	આઉટપુટ
જોખમ ઓળખ	બ્રેઇનસ્ટોર્મિંગ, ચેકલિસ્ટ્સ	જોખમ સૂચિ
જોખમ વિશ્લેષણ	સંભાવના × પ્રભાવ	જોખમ પ્રાથમિકતા
જોખમ મૂલ્યાંકન	જોખમ મેટ્રિક્સ	કાર્ય પ્રાથમિકતાઓ

(ii) Risk Mitigation: જોખમની અસર અને સંભાવના ઘટાડવાની વ્યૂહરચનાઓ.

કોષ્ટક 23. ઘટાડવાની વ્યૂહરચનાઓ

વ્યૂહરચના	વર્ણન	ઉદાહરણ
ટાળવું	જોખમ સ્ત્રોત દૂર કરવો	ટેકનોલોજી બદલવી
ઘટાડવું	અસર ઓછી કરવી	ટેસ્ટિંગ ઉમેરવું
ટ્રાન્સફર કરવું	અન્યને જોખમ સ્થાનાંતરિત કરવું	વીમો, આઉટસોર્સિંગ
સ્વીકારવું	જોખમ સાથે જીવવું	કન્ટિન્જન્સી પ્લાનિંગ

મેમરી ટ્રીક

“ટાળો ઘટાડો ટ્રાન્સફર સ્વીકારો”

પ્રશ્ન 4(c) [7 ગુણ]

Define project risk and how Manage Risk Management it.

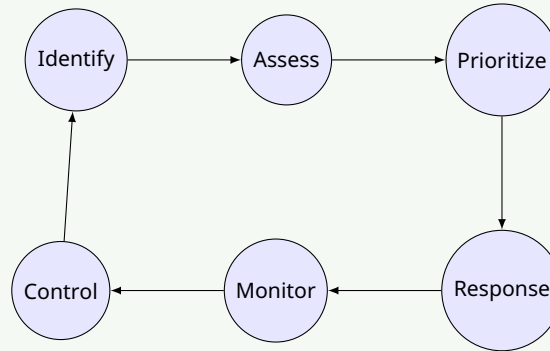
જવાબ

પ્રોજેક્ટ જોખમ એ અનિશ્ચિત ઘટના છે જે, જો થાય તો, પ્રોજેક્ટ લક્ષ્યો પર સકારાત્મક અથવા નકારાત્મક અસર કરે છે.
જોખમ લક્ષણો:

કોષ્ટક 24. લક્ષણો

લક્ષણ	વર્ણન	ઉદાહરણ
અનિશ્ચિતતા	થઈ શકે અથવા ન પણ થાય	ટેકનોલોજી નિષ્ફળતા
પ્રભાવ	પ્રોજેક્ટ પેરામીટર્સને અસર કરે	ખર્ચ, શેડ્યૂલ, ગુણવત્તા
સંભાવના	થવાની શક્યતા	30% વિલંબની તક

જોખમ વ્યવસ્થાપન પ્રક્રિયા:



આકૃતિ 14. Management Loop

જોખમ વ્યવસ્થાપન પગલાં:

કોષ્ટક 25. પ્રક્રિયા વિગતો

પગલું	પ્રવૃત્તિઓ	ટૂલ્સ	આઉટપુટ
જોખમ ઓળખ	બ્રેઇનસ્ટોર્મિંગ	ચેકલિસ્ટ્સ, SWOT	જોખમ રજિસ્ટર
જોખમ મૂલ્યાંકન	સંભાવના/પ્રભાવ વિશ્લેષણ	જોખમ મેટ્રિક્સ	જોખમ રેટિંગ્સ
જોખમ પ્રતિભાવ	વ્યૂહરચના વિકસાવવી	પ્રતિભાવ ટેમ્પલેટ્સ	કાર્ય યોજનાઓ
જોખમ મોનિટરિંગ	સૂચકો ટ્રેક કરવા	ડેશબોર્ડ્સ	સ્થિતિ રિપોર્ટ્સ

જોખમ પ્રતિભાવ વ્યૂહરચનાઓ:

- નકારાત્મક જોખમો: ટાળવું, ટ્રાન્સફર કરવું, ઘટાડવું, સ્વીકારવું
- સકારાત્મક જોખમો: શોષણ કરવું, શેર કરવું, વધારવું, સ્વીકારવું

મેમરી ટ્રીક

“ઓળખો મૂલ્યાંકન પ્રતિભાવ મોનિટર + ટાળો ટ્રાન્સફર ઘટાડો સ્વીકારો”

પ્રશ્ન 4(a OR) [3 ગુણ]

Describe Software design process and explain Design methodologies.

જવાબ

સૉફ્ટવેર ડિઝાઇન આવશ્યકતાઓને વ્યવસ્થિત અભિગમ દ્વારા અમલીકરણ માટે બ્લુપ્રિન્ટમાં રૂપાંતરિત કરે છે.
ડિઝાઇન પ્રક્રિયા:

કોષ્ટક 26. પ્રક્રિયા માળખું

તબક્કો	પ્રવૃત્તિ	આઉટપુટ
વિશ્લેષણ	આવશ્યકતાઓ સમજવી	સમસ્યા વ્યાખ્યા
આર્કિટેક્ચર	ઉચ્ચ-સ્તરીય માળખું	સિસ્ટમ આર્કિટેક્ચર
વિગતવાર ડિઝાઇન	ઘટક સ્પષ્ટીકરણ	ડિઝાઇન દસ્તાવેજો

મેમરી ટ્રીક

“વિશ્લેષણ આર્કિટેક્ચર વિગત”

પ્રશ્ન 4(b OR) [4 ગુણ]

Compare Cohesion and Coupling side by side.

જવાબ

બંને ખ્યાલો મોડ્યુલ ડિઝાઇન ગુણવત્તા માપે છે પરંતુ વિવિધ પાસાઓ પર ધ્યાન કેન્દ્રિત કરે છે.
વ્યાપક સરખામણી:

કોષ્ટક 27. Cohesion vs Coupling

પાસું	Cohesion	Coupling
વ્યાખ્યા	મોડ્યુલની અંદર સંબંધની ડિગ્રી	મોડ્યુલો વચ્ચે પરસ્પર નિર્ભરતાની ડિગ્રી
લક્ષ્ય	ઉચ્ચ cohesion ઇચ્છનીય	નીચું coupling ઇચ્છનીય
ધ્યાન	આંતરિક મોડ્યુલ માળખું	આંતર-મોડ્યુલ સંબંધો
ગુણવત્તા	મજબૂત = બહેતર	નબળું = બહેતર

પ્રકારોની સરખામણી (શ્રેષ્ઠથી ખરાબ):

- **Cohesion:** Functional, Sequential, Communicational, Procedural, Temporal, Logical, Coincidental
- **Coupling:** Data, Stamp, Control, External, Common, Content

ડિઝાઇન પર પ્રભાવ: ઉચ્ચ cohesion અને નીચું coupling વધુ સારી જાળવણીક્ષમતા, પુનઃઉપયોગ, અને ટેસ્ટિંગ તરફ દોરી જાય છે.

મેમરી ટ્રીક

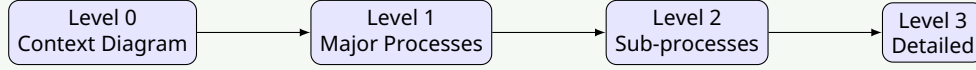
“Cohesion = અંદર મજબૂત, Coupling = વચ્ચે નબળું”

પ્રશ્ન 4(c OR) [7 ગુણ]

Sketch Data Flow Diagram with levels and explain.

જવાબ

ડેટા ફ્લો ડાયાગ્રામ (DFD) ગ્રાફિકલ નોટેશન વાપરીને સિસ્ટમ દ્વારા ડેટા કેવી રીતે ચાલે છે તે બતાવે છે અને વિગતના બહુવિધ સ્તરો ધરાવે છે. DFD સ્તરો:



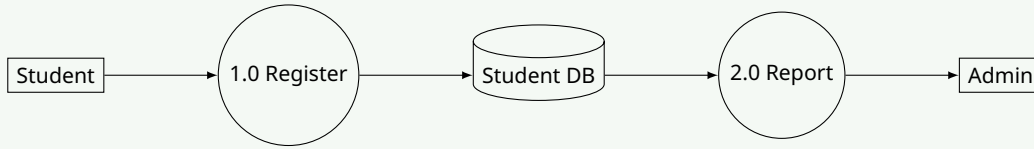
આકૃતિ 15. DFD Levels

સ્તર વર્ણનો:

કોષ્ટક 28. સ્તર વિગતો

સ્તર	અવકાશ	હેતુ	વિગત
Level 0	સંપૂર્ણ સિસ્ટમ	સિસ્ટમ સીમા	એક પ્રોસેસ
Level 1	મુખ્ય કાર્યો	ઉચ્ચ-સ્તરીય પ્રોસેસો	5-7 પ્રોસેસો
Level 2	ઉપ-કાર્યો	પ્રોસેસ વિભાજન	વિગતવાર દૃશ્ય
Level 3+	બારીક વિગતો	અમલીકરણ સ્તર	ખૂબ જ વિશિષ્ટ

ઉદાહરણ - Level 1 DFD:



આકૃતિ 16. Level 1 ઉદાહરણ

ફાયદા: અમૂર્તતા, વિઘટન, ચકાસણી.

મેમરી ટ્રીક

“Context મુખ્ય ઉપ બારીક + પ્રોસેસ એન્ટિટી સ્ટોર ફ્લો”

પ્રશ્ન 5(a) [3 ગુણ]

Give Characteristics of good UI.

જવાબ

સારો યુઝર ઇન્ટરફેસ ડિઝાઇન સોફ્ટવેર સિસ્ટમ સાથે અસરકારક યુઝર ક્રિયાપ્રતિક્રિયા સુનિશ્ચિત કરે છે. UI લક્ષણિકતાઓ:

કોષ્ટક 29. મુખ્ય લક્ષણો

લાક્ષણિકતા	વર્ણન	ફાયદો
સરળ	સમજવામાં સરળ	શીખવાની વળાંક ઘટાડે છે
સુસંગત	એકસમાન વર્તન	અનુમાનિત ક્રિયાપ્રતિક્રિયા
પ્રતિસાદશીલ	ઝડપી ફીડબેક	યુઝર સંતુષ્ટતા

મેમરી ટ્રીક

“સરળ સુસંગત પ્રતિસાદશીલ”

પ્રશ્ન 5(b) [4 ગુણ]

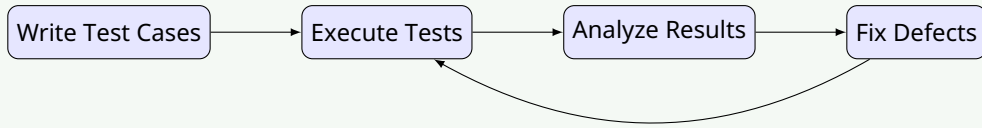
Briefly explain Unit testing

જવાબ

યુનિટ ટેસ્ટિંગ સાચી કાર્યક્ષમતા સુનિશ્ચિત કરવા માટે વ્યક્તિગત સૉફ્ટવેર ઘટકોને અલગતામાં ચકાસે છે.
યુનિટ ટેસ્ટિંગ ઝાંખી:

કોષ્ટક 30. ટેસ્ટિંગ અવકાશ

પાસું	વર્ણન	હેતુ
અવકાશ	વ્યક્તિગત મોડ્યુલ્સ	ઘટક ચકાસણી
અલગતા	અલગતામાં ટેસ્ટ	સ્વતંત્ર ચકાસણી
સ્વચાલન	સ્વચાલિત અમલીકરણ	કાર્યક્ષમ ટેસ્ટિંગ
વહેલી શોધ	વહેલો બગ શોધ	ખર્ચ-અસરકારક



આકૃતિ 17. Unit Testing Cycle

ફાયદા: વહેલી બગ શોધ, કોડ ગુણવત્તા સુધારણા, રિગ્રેશન ટેસ્ટિંગ.

મેમરી ટ્રીક

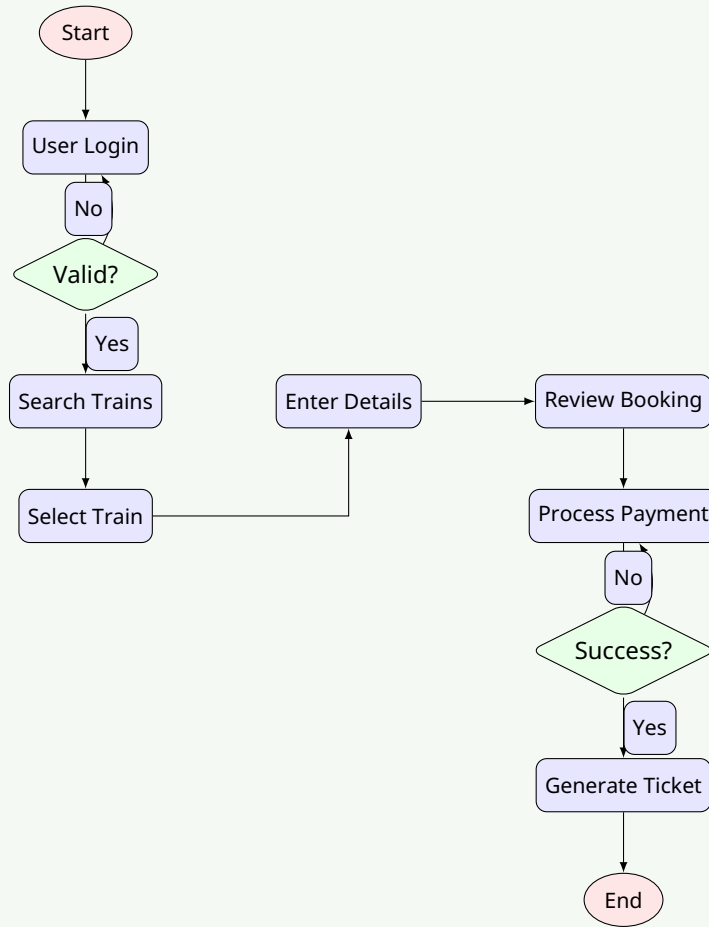
“અવકાશ અલગતા સ્વચાલન વહેલી”

પ્રશ્ન 5(c) [7 ગુણ]

Draw activity diagrams of the train reservation system, explain each step.

જવાબ

Activity Diagram યુઝર વિનંતીથી ટિકિટ પુષ્ટિ સુધી ટ્રેન રિઝર્વેશન સિસ્ટમનો વર્કફ્લો બતાવે છે.



આકૃતિ 18. Train Reservation Activity

પગલા-દર-પગલાની સમજૂતી:

- **Login:** યુઝર ઓથેન્ટિકેશન.
- **Search:** રૂટ/તારીખ માટે ટ્રેન શોધવી.
- **Selection:** ટ્રેન અને સીટ પસંદ કરવી.
- **Details:** પેસેન્જર માહિતી દાખલ કરવી.
- **Payment:** ટ્રાન્ઝેક્શન હેન્ડલ કરવું.
- **Ticket:** ટિકિટ જનરેટ અને પુષ્ટિ મોકલવી.

મેમરી ટ્રીક

“લોગિન સર્ચ સિલેક્ટ ચૂક એન્ટર રિવ્યુ પે જનરેટ સેન્ડ”

પ્રશ્ન 5(a OR) [3 ગુણ]

Compare Verification, Validation side by side.

જવાબ

બંને ગુણવત્તા ખાતરીની પ્રવૃત્તિઓ છે પરંતુ વિવિધ પાસાઓ પર ધ્યાન કેન્દ્રિત કરે છે.

Verification વિરુદ્ધ Validation:

કોષ્ટક 31. સરખામણી

પાસું	Verification	Validation
પ્રશ્ન	"શું આપણે સાચું બનાવી રહ્યા છીએ?"	"શું આપણે સાચી વસ્તુ બનાવી રહ્યા છીએ?"
ધ્યાન	પ્રક્રિયાની સાચકી	ઉત્પાદનની સાચકી
પદ્ધતિ	સમીક્ષાઓ, નિરીક્ષણો	ટેસ્ટિંગ, યુઝર ફીડબેક

મેમરી ટ્રીક

“Verification = સાચી પ્રક્રિયા, Validation = સાચું ઉત્પાદન”

પ્રશ્ન 5(b OR) [4 ગુણ]

Define Testing describe any two testing type.

જવાબ

ટેસ્ટિંગ એ તૂલો શોધવા અને તે આવશ્યકતાઓ પૂરી કરે છે તે ચકાસવા માટે સોફ્ટવેરનું મૂલ્યાંકન કરવાની પ્રક્રિયા છે.
બે ટેસ્ટિંગ પ્રકારો:

કોષ્ટક 32. Black Box vs White Box

પાસું	Black Box	White Box
અભિગમ	આંતરિક માળખું જાણ્યા વિના	કોડ માળખા જાણ સાથે
ધ્યાન	કાર્યાત્મક આવશ્યકતાઓ	આંતરિક તર્ક
ટેસ્ટર	યુઝર સ્વીકૃતિ	ડેવલપર યુનિટ ટેસ્ટિંગ

મેમરી ટ્રીક

“Black = બાહ્ય, White = આંતરિક”

પ્રશ્ન 5(c OR) [7 ગુણ]

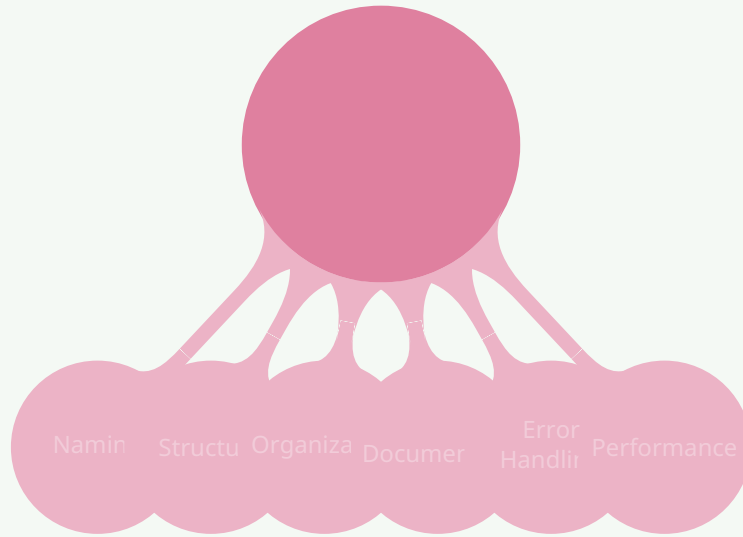
Describe each Coding standards and guidelines.

જવાબ

કોડિંગ સ્ટાન્ડર્ડ્સ એ સુસંગત, જાળવી શકાય તેવા કોડ લખવા માટેના નિયમો અને પરંપરાઓનો સમૂહ છે.

મુખ્ય શ્રેણીઓ:

1. નામકરણ પરંપરાઓ: વેરિએબલ્સ માટે camelCase, ક્લાસીસ માટે PascalCase.
2. કોડ માળખું: સુસંગત ઇન્ડેન્ટેશન, લાઇન લંબાઈ મર્યાદા.
3. ગોઠવણી: એક જવાબદારી, નાના ફંક્શન્સ.
4. દસ્તાવેજીકરણ: હેડર કોમેન્ટ્સ, અર્થપૂર્ણ ઇનલાઇન કોમેન્ટ્સ.
5. એરર હેન્ડલિંગ: ગ્રેસફુલ એક્સેપ્શન હેન્ડલિંગ, લોગિંગ.
6. પર્ફોર્મન્સ: મેમરી લીક્સ ટાળવા, કાર્યક્ષમ એલ્ગોરિધમ્સ.



આકૃતિ 19. Standard Categories

ફાયદા: બહેતર વાંચી શકાય તેવું, સુસંગતતા, જાળવણીક્ષમતા, અને ગુણવત્તા.

મેમરી ટ્રીક

“નામ માળખું ગોઠવણી દસ્તાવેજ હેન્ડલ પર્ફોર્મ રિવ્યુ”