

સોફ્ટવેર એન્જિનિયરિંગ (4353202) - શિયાળો 2024 ઉકેલ

Milav Dabgar

November 25, 2024

પ્રશ્ન 1(a) [3 ગુણ]

સોફ્ટવેર ની વ્યાખ્યા આપો અને તેની લાક્ષણિકતા સમજાવો.

જવાબ

સોફ્ટવેર એ કમ્પ્યુટર પ્રોગ્રામ્સ, પ્રક્રિયાઓ અને દસ્તાવેજીકરણનો સમૂહ છે જે કમ્પ્યુટર સિસ્ટમ પર કાર્યો કરે છે.

કોષ્ટક 1. સોફ્ટવેર લાક્ષણિકતાઓ

લાક્ષણિકતા	વર્ણન
અસ્પર્શ્ય	સ્પર્શ કરી શકાતું નથી, માત્ર અનુભવી શકાય છે
વિકસિત	એન્જિનિયર્ડ, મેન્યુફેક્ચર્ડ નહીં
જાળવણીયોગ્ય	સુધારણા અને અપડેટ કરી શકાય છે
વિશ્વસનીય	સતત કામ કરવું જોઈએ
કાર્યક્ષમ	સાધનોનો શ્રેષ્ઠ ઉપયોગ કરે છે

- મુખ્ય મુદ્દો: સોફ્ટવેર = પ્રોગ્રામ્સ + દસ્તાવેજીકરણ + પ્રક્રિયાઓ

મેમરી ટ્રીક

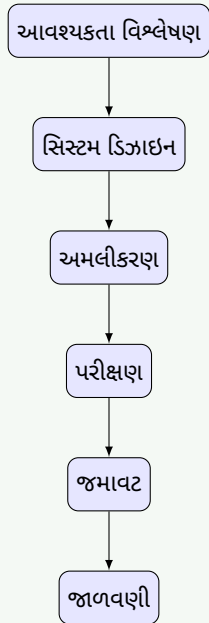
"I Don't Make Reliable Electronics (Intangible, Developed, Maintainable, Reliable, Efficient)"

પ્રશ્ન 1(b) [4 ગુણ]

ક્લાસિકલ વોટરફોલ મોડેલ સમજાવો.

જવાબ

વોટરફોલ મોડેલ એ રેખીય ક્રમિક સોફ્ટવેર વિકાસ પદ્ધતિ છે જ્યાં દરેક તબક્કો પૂર્ણ થયા પછી જ આગળનો તબક્કો શરૂ થાય છે.



આકૃતિ 1. ક્લાસિકલ વોટરફોલ મોડેલ

મુખ્ય લક્ષણો:

- ક્રમિક તબક્કાઓ: તબક્કાઓ વચ્ચે કોઈ ઓવરલેપ નથી
- દસ્તાવેજીકરણ આધારિત: દરેક તબક્કે ભારે દસ્તાવેજીકરણ
- સરળ માળખું: સમજવા અને મેનેજ કરવા સરળ
- નિશ્ચિત આવશ્યકતાઓ: એકવાર શરૂ થયા પછી ફેરફાર મુશ્કેલ

મેમરી ટ્રીક

"Real Systems Include Testing, Deployment, Maintenance"

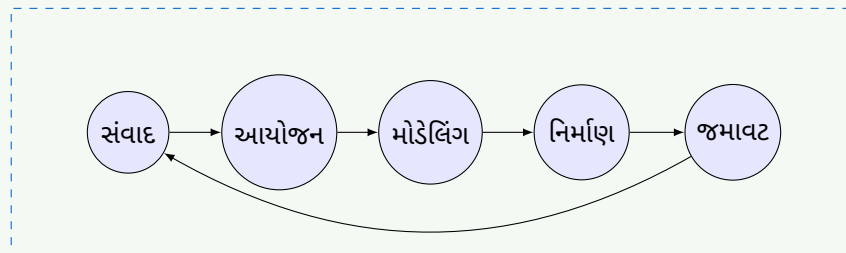
પ્રશ્ન 1(c) [7 ગુણ]

સોફ્ટવેર પ્રોસેસ ફ્રેમવર્ક અને અમ્બ્રેલા એક્ટિવિટી સમજાવો.

જવાબ

સોફ્ટવેર પ્રોસેસ ફ્રેમવર્ક મુખ્ય પ્રોસેસ વિસ્તારો ઓળખીને સંપૂર્ણ સોફ્ટવેર એન્જિનિયરિંગ પ્રોસેસ માટે પાથો પ્રદાન કરે છે.

અમ્બ્રેલા એક્ટિવિટીઝ



આકૃતિ 2. પ્રોસેસ ફ્રેમવર્ક & અમ્બ્રેલા એક્ટિવિટીઝ

કોષ્ટક 2. ફ્રેમવર્ક એક્ટિવિટીઝ વિ અમ્બ્રેલા એક્ટિવિટીઝ

ફ્રેમવર્ક એક્ટિવિટીઝ	અમ્બ્રેલા એક્ટિવિટીઝ
સંવાદ	સોફ્ટવેર પ્રોજેક્ટ ટ્રેકિંગ
આયોજન	જોખમ મેનેજમેન્ટ
મોડેલિંગ	ગુણવત્તા ખાતરી
નિર્માણ	તકનીકી સમીક્ષાઓ
જમાવટ	કન્ફિગરેશન મેનેજમેન્ટ

ફ્રેમવર્ક એક્ટિવિટીઝ:

- સંવાદ: સ્ટેકહોલ્ડર્સ પાસેથી આવશ્યકતાઓ એકત્રિત કરવી
- આયોજન: પ્રોજેક્ટ પ્લાન અને શેડ્યુલ બનાવવું
- મોડેલિંગ: ડિઝાઇન મોડેલ્સ બનાવવા
- નિર્માણ: કોડ જનરેશન અને પરીક્ષણ
- જમાવટ: સોફ્ટવેર ડિલિવરી અને ફીડબેક

અમ્બ્રેલા એક્ટિવિટીઝ પ્રોજેક્ટ દરમિયાન ચાલે છે:

- પ્રોજેક્ટ ટ્રેકિંગ: પ્રગતિ નિરીક્ષણ
- જોખમ મેનેજમેન્ટ: જોખમો ઓળખવા અને નિયંત્રિત કરવા
- ગુણવત્તા ખાતરી: ગુણવત્તા ધોરણો સુનિશ્ચિત કરવા
- કન્ફિગરેશન મેનેજમેન્ટ: ફેરફારો નિયંત્રિત કરવા

મેમરી ટ્રીક

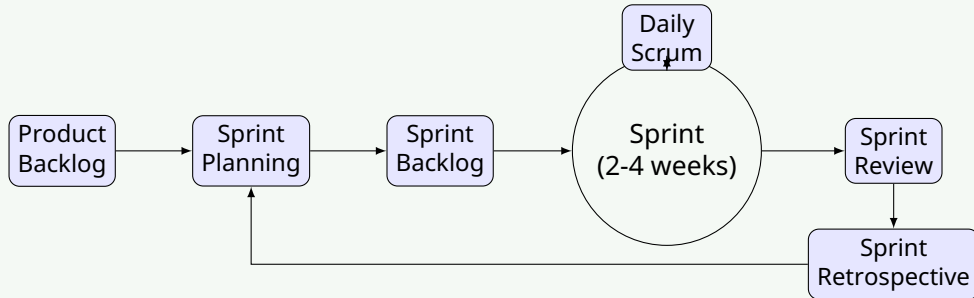
“Can People Make Construction Deploy (Communication, Planning, Modeling, Construction, Deployment)”

પ્રશ્ન 1(c) OR [7 ગુણ]

SCRUM મોડેલ પર ટૂંક નોંધ લખો.

જવાબ

SCRUM એ પુનરાવર્તક અને વૃદ્ધિશીલ પ્રથાઓનો ઉપયોગ કરીને સોફ્ટવેર વિકાસ પ્રોજેક્ટ્સનું મેનેજમેન્ટ કરવા માટેનું એક agile ફ્રેમવર્ક છે.



આકૃતિ 3. SCRUM Process Flow

કોષ્ટક 3. SCRUM ભૂમિકાઓ અને આર્ટિફેક્ટ્સ

ઘટક	વર્ણન
Product Owner	આવશ્યકતાઓ અને પ્રાથમિકતાઓ વ્યાખ્યાયિત કરે છે
Scrum Master	પ્રક્રિયાને સુવિધા આપે છે અને અવરોધો દૂર કરે છે
Development Team	સ્વ-સંગઠિત ટીમ જે પ્રોડક્ટ બનાવે છે
Product Backlog	લક્ષણોની પ્રાથમિકતા આપેલી યાદી
Sprint Backlog	વર્તમાન sprint માટે પસંદ કરેલા કાર્યો

મુખ્ય ઇવેન્ટ્સ:

- **Sprint Planning:** આગામી sprint માટે કામ પસંદ કરવું
 - **Daily Scrum:** 15-મિનિટનું દૈનિક સિંક્રોનાઇઝેશન
 - **Sprint Review:** પૂર્ણ થયેલ કામ દર્શાવવું
 - **Sprint Retrospective:** પ્રક્રિયા પર વિચાર કરવો અને સુધારવું
- ફાયદાઓ:** ઝડપી ડિલિવરી, લવચીકતા, સતત સુધારણા, ગ્રાહક સહયોગ

મેમરી ટ્રીક

“People Sprint Daily Reviewing Retrospectively”

પ્રશ્ન 2(a) [3 ગુણ]

સારા SRS ની લાક્ષણિકતા સમજાવો.

જવાબ

SRS (સોફ્ટવેર આવશ્યકતા વિશિષ્ટતા) દસ્તાવેજ અસરકારક બનવા માટે વિશિષ્ટ ગુણો હોવા જોઈએ.

કોષ્ટક 4. સારા SRS લાક્ષણિકતાઓ

લાક્ષણિકતા	અર્થ
સંપૂર્ણ	બધી આવશ્યકતાઓ સમાવેશ
સુસંગત	કોઈ વિરોધાભાસી આવશ્યકતાઓ નથી
અસ્પષ્ટ નથી	સ્પષ્ટ અને એક અર્થઘટન
ચકાસણીયોગ્ય	પરીક્ષણ અને વેલિડેશન શક્ય
સુધારણીયોગ્ય	જરૂર પડે ત્યારે બદલવા સરળ

- **સંપૂર્ણ:** બધી functional અને non-functional આવશ્યકતાઓ સમાવે છે
- **સુસંગત:** વિવિધ આવશ્યકતાઓ વચ્ચે કોઈ સંઘર્ષ નથી
- **અસ્પષ્ટ નથી:** દરેક આવશ્યકતાનો માત્ર એક જ અર્થઘટન છે

મેમરી ટ્રીક

“Complete Computers Use Verified Modifications”

પ્રશ્ન 2(b) [4 ગુણ]

પ્રોટોટાઇપ મોડેલના લાભ અને ગેરલાભ વર્ણવો.

જવાબ

પ્રોટોટાઇપ મોડેલ આવશ્યકતાઓને વધુ સારી રીતે સમજવા માટે સોફ્ટવેરનું કાર્યકારી મોડેલ બનાવે છે.

કોષ્ટક 5. પ્રોટોટાઇપ મોડેલ - ફાયદા અને ગેરફાયદા

ફાયદા	ગેરફાયદા
આવશ્યકતા સમજણ સુધારે છે	સમયનો વધારે ખર્ચ
વપરાશકર્તા સામેલગીરી	ખર્ચમાં વધારો
પ્રારંભિક ભૂલ શોધ	અપૂર્ણ વિશ્લેષણ
વપરાશકર્તા સંતુષ્ટિ	પ્રોટોટાઇપ મૂંઝવણ

ફાયદા:

- સ્પષ્ટ આવશ્યકતાઓ: વપરાશકર્તા કાર્યકારી મોડેલ જુએ છે
- પ્રારંભિક ફીડબેક: ખર્ચાળ ભૂલો ટાળે છે
- વપરાશકર્તા સામેલગીરી: શ્રેષ્ઠ વપરાશકર્તા સ્વીકૃતિ

ગેરફાયદા:

- વધારાનો સમય: પ્રોટોટાઇપ બનાવવામાં સમય લાગે છે
- વધારાનો ખર્ચ: પ્રોટોટાઇપ માટે સંસાધનોની જરૂર
- સ્કોપ ક્રીપ: વપરાશકર્તાઓ વધુ સુવિધાઓ માંગી શકે છે

મેમરી ટ્રીક

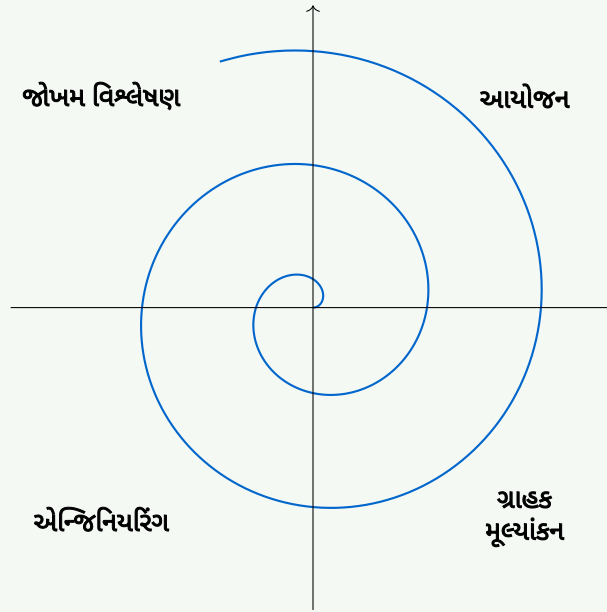
“Better Users Experience vs Time Costs Increase”

પ્રશ્ન 2(c) [7 ગુણ]

Spiral મોડેલ ડિઝાઇન કરો, વર્ણવો અને તેના ફાયદા અને ગેરફાયદા આપો.

જવાબ

Spiral મોડેલ પુનરાવર્તક વિકાસને વ્યવસ્થિત જોખમ સંચાલન સાથે જોડે છે.



આકૃતિ 4. Spiral મોડેલ

કોષ્ટક 6. Spiral મોડેલ તબક્કાઓ

તબક્કો	પ્રવૃત્તિઓ
આયોજન	આવશ્યકતાઓ એકત્રિત કરવી, સંસાધન આયોજન
જોખમ વિશ્લેષણ	જોખમો ઓળખવા અને ઉકેલવા
એન્જિનિયરિંગ	વિકાસ અને પરીક્ષણ
ગ્રાહક મૂલ્યાંકન	ગ્રાહક સમીક્ષાઓ અને ફીડબેક

ફાયદા:

- જોખમ સંચાલન: પ્રારંભિક જોખમ ઓળખ
- લવચીકતા: ફેરફારો સરળતાથી સમાવી શકાય છે
- ગ્રાહક સંડોવણી: નિયમિત ગ્રાહક ફીડબેક

- ગુણવત્તા ફોકસ: સતત પરીક્ષણ અને માન્યતા
- ગેરફાયદા:
 - જટિલ મેનેજમેન્ટ: સંચાલન કરવું મુશ્કેલ
 - વધુ ખર્ચ: જોખમ વિશ્લેષણને કારણે ખર્ચાળ
 - સમય માંગી લે છે: લાંબા વિકાસ ચક્ર
 - જોખમ નિષ્ણાતની જરૂર: જોખમ આકારણી કુશળતા આવશ્યક છે
- શ્રેષ્ઠ આ માટે: મોટા, જટિલ, ઉચ્ચ-જોખમવાળા પ્રોજેક્ટ્સ

મેમરી ટ્રીક

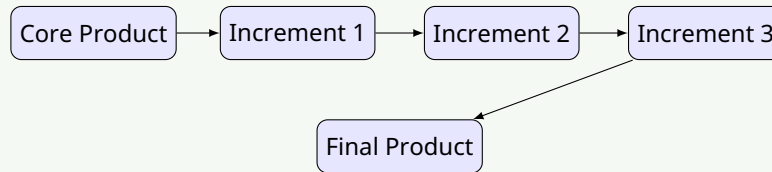
“Plan Risks Engineering Customer”

પ્રશ્ન 2(a) OR [3 ગુણ]

Incremental મોડેલ સમજાવો.

જવાબ

Incremental મોડેલ ઇન્ક્રીમેન્ટ્સ તરીકે ઓળખાતા નાના, વિધેયાત્મક ટુકડાઓમાં સોફ્ટવેર વિતરિત કરે છે.



આકૃતિ 5. Incremental Delivery

મુખ્ય લક્ષણો:

- આંશિક અમલીકરણ: દરેક ઇન્ક્રીમેન્ટ કાર્યક્ષમતા ઉમેરે છે
- પ્રારંભિક ડિલિવરી: મુખ્ય સુવિધાઓ પહેલાં વિતરિત
- સમાંતર વિકાસ: બહુવિધ ઇન્ક્રીમેન્ટ્સ એકસાથે વિકસાવી શકાય છે

કોષ્ટક 7. Incremental મોડેલ લાક્ષણિકતાઓ

પાસા	વર્ણન
ડિલિવરી	બહુવિધ રીલીઝ
કાર્યક્ષમતા	દરેક ઇન્ક્રીમેન્ટ સાથે વધે છે
જોખમ	પ્રારંભિક ડિલિવરી દ્વારા ઘટાડો
ફીડબેક	સતત વપરાશકર્તા ફીડબેક

મેમરી ટ્રીક

“Deliver Functionality Reducing Feedback”

પ્રશ્ન 2(b) OR [4 ગુણ]

RAD મોડેલનો ખ્યાલ લખો અને સમજાવો.

જવાબ

RAD (Rapid Application Development) વિસ્તૃત આયોજન પર ઝડપી પ્રોટોટાઇપિંગ અને ઝડપી ફીડબેક પર ભાર મૂકે છે.

કોષ્ટક 8. RAD મોડેલ તબક્કાઓ

તબક્કો	સમય	પ્રવૃત્તિઓ
Business Modeling	ટૂંકા	વ્યવસાય કાર્યો વ્યાખ્યાયિત કરો
Data Modeling	ટૂંકા	ડેટા આવશ્યકતાઓ વ્યાખ્યાયિત કરો
Process Modeling	ટૂંકા	ડેટાને માહિતીમાં રૂપાંતરિત કરો
Application Generation	ટૂંકા	સોફ્ટવેર બનાવવા સાધનો ઉપયોગ કરો
Testing & Turnover	ટૂંકા	પરીક્ષણ અને જમાવટ

મુખ્ય ખ્યાલો:

- પુનઃઉપયોગી ઘટકો: પૂર્વ-નિર્મિત ઘટકો વિકાસ ઝડપી બનાવે છે
- શક્તિશાળી સાધનો: CASE સાધનો અને કોડ જનરેટર્સ
- નાની ટીમો: ટીમ દીઠ 2-6 લોકો
- સમય-બદ્ધ: કડક સમય મર્યાદા (60-90 દિવસ)

RAD માટે આવશ્યકતાઓ:

- સારી રીતે વ્યાખ્યાયિત વ્યવસાય આવશ્યકતાઓ
- વપરાશકર્તા સામેલગીરી પ્રક્રિયા દરમિયાન
- કુશળ ડેવલપર્સ RAD સાધનોથી પરિચિત

મેમરી ટ્રીક

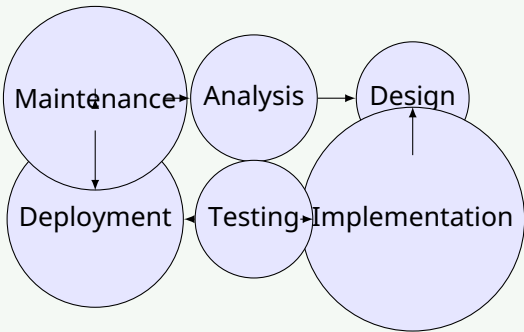
“Business Data Process Application Testing”

પ્રશ્ન 2(c) OR [7 ગુણ]

SDLC વ્યાખ્યાયિત કરો અને દરેક તબક્કા સમજાવો.

જવાબ

SDLC (સોફ્ટવેર ડેવલપમેન્ટ લાઇફ સાયકલ) એ સારી રીતે વ્યાખ્યાયિત તબક્કાઓ દ્વારા સોફ્ટવેર બનાવવા માટેની વ્યવસ્થિત પ્રક્રિયા છે.



આકૃતિ 6. SDLC ચક્ર

કોષ્ટક 9. SDLC તબક્કાઓ વિગતવાર

તબક્કો	પ્રવૃત્તિઓ	ડિલિવરી
Planning	પ્રોજેક્ટ આયોજન, શક્યતા અભ્યાસ	પ્રોજેક્ટ પ્લાન
Analysis	આવશ્યકતા એકત્રીકરણ	SRS દસ્તાવેજ
Design	સિસ્ટમ આર્કિટેક્ચર, UI ડિઝાઇન	ડિઝાઇન દસ્તાવેજ
Implementation	કોડિંગ, યુનિટ ટેસ્ટિંગ	સોર્સ કોડ
Testing	સિસ્ટમ ટેસ્ટિંગ, ઇન્ટીગ્રેશન	ટેસ્ટ રિપોર્ટ્સ
Deployment	ઇન્સ્ટોલેશન, તાલીમ	લાઇવ સિસ્ટમ
Maintenance	બગ ફિક્સ, સુધારાઓ	અપડેટ સિસ્ટમ

તબક્કા વર્ણન:

- **Planning:** પ્રોજેક્ટ સ્કોપ અને સંસાધનો નક્કી કરવા
- **Analysis:** સિસ્ટમે શું કરવું જોઈએ તે સમજવું
- **Design:** સિસ્ટમ કેવી રીતે કામ કરશે તે પ્લાન કરવું
- **Implementation:** વાસ્તવિક સિસ્ટમ બનાવવી
- **Testing:** સિસ્ટમ યોગ્ય રીતે કાર્ય કરે છે તે ચકાસવું
- **Deployment:** વપરાશકર્તાઓ માટે સિસ્ટમ રજૂ કરવી
- **Maintenance:** ચાલુ સપોર્ટ અને અપડેટ્સ

મેમરી ટ્રીક

“People Always Design Implementation, Test Deployment, Maintain”

પ્રશ્ન 3(a) [3 ગુણ]

સોફ્ટવેર પ્રોજેક્ટ્સ મેનેજ કરવા માટેની કુશળતા વર્ણવો.

જવાબ

સોફ્ટવેર પ્રોજેક્ટ મેનેજમેન્ટ માટે તકનીકી અને સોફ્ટ સ્કિલ્સના સંયોજનની જરૂર છે.

કોષ્ટક 10. આવશ્યક પ્રોજેક્ટ મેનેજમેન્ટ સ્કિલ્સ

કૌશલ્ય શ્રેણી	વિશિષ્ટ કૌશલ્યો
ટેકનિકલ	SDLC, સાધનો, તકનીકોની સમજ
નેતૃત્વ	ટીમ પ્રેરણા, નિર્ણય લેવાની ક્ષમતા
સંવાદ	ટીમ અને ક્લાયન્ટ્સ સાથે સ્પષ્ટ વાતચીત
આયોજન	સંસાધન ફાળવણી, સમયપત્રક
સમસ્યા નિવારણ	જોખમ સંચાલન, સંઘર્ષ નિવારણ

મુખ્ય કૌશલ્યો:

- **પીપલ મેનેજમેન્ટ:** ટીમ સભ્યોનું નેતૃત્વ અને પ્રેરણા
- **ટેકનિકલ જ્ઞાન:** વિકાસ પ્રક્રિયા અને સાધનોની સમજ
- **સંવાદ:** ટેકનિકલ ટીમ અને સ્ટેકહોલ્ડર્સ વચ્ચે સેતુ

મેમરી ટ્રીક

“Technical Leaders Communicate Planning Problems”

પ્રશ્ન 3(b) [4 ગુણ]

સોફ્ટવેર પ્રોજેક્ટ મેનેજરની જવાબદારી સંક્ષિપ્તમાં લખો.

જવાબ

સોફ્ટવેર પ્રોજેક્ટ મેનેજર શરૂઆતથી અંત સુધી સમગ્ર પ્રોજેક્ટની દેખરેખ રાખે છે.

કોષ્ટક 11. પ્રોજેક્ટ મેનેજરની જવાબદારીઓ

ક્ષેત્ર	જવાબદારીઓ
આયોજન	પ્રોજેક્ટ પ્લાન, સમયપત્રક, બજેટ બનાવવા
ટીમ મેનેજમેન્ટ	ભરતી, તાલીમ, અને ટીમ મેનેજમેન્ટ
સંવાદ	સ્ટેકહોલ્ડર્સને નિયમિત અપડેટ્સ
ગુણવત્તા નિયંત્રણ	ડિલિવેરેબલ્સ ગુણવત્તા ધોરણો પૂર્ણ કરે છે
જોખમ સંચાલન	પ્રોજેક્ટ જોખમો ઓળખવા અને ઘટાડવા

પ્રાથમિક જવાબદારીઓ:

- પ્રોજેક્ટ આયોજન: સ્કોપ, સમયરેખા અને સંસાધનો વ્યાખ્યાયિત કરવા
- ટીમ નેતૃત્વ: વિકાસ ટીમને માર્ગદર્શન અને સપોર્ટ
- સ્ટેકહોલ્ડર સંવાદ: પ્રગતિ વિશે સૌને માહિતગાર રાખવા
- ગુણવત્તા ખાતરી: પ્રોજેક્ટ આવશ્યકતાઓ પૂર્ણ કરે તે સુનિશ્ચિત કરવું
- જોખમ સંચાલન: પ્રોજેક્ટ જોખમો અને સમસ્યાઓ સંભાળવી

મેમરી ટ્રીક

"Plan Team Communication Quality Risk"

પ્રશ્ન 3(c) [7 ગુણ]

SRS માં આવશ્યકતાઓના પ્રકારોનું વર્ગીકરણ કરો (1) Functional Requirements (2) Non-Functional Requirements.

જવાબ

આવશ્યકતા વર્ગીકરણ વિવિધ પ્રકારની સિસ્ટમ જરૂરિયાતોને વ્યવસ્થિત અને સમજવામાં મદદ કરે છે.

કોષ્ટક 12. Functional વિ Non-Functional Requirements

પાસા	Functional Requirements	Non-Functional Requirements
વ્યાખ્યા	સિસ્ટમે શું કરવું જોઈએ	સિસ્ટમ કેવી રીતે કામ કરવી જોઈએ
ફોક્સ	સિસ્ટમ કાર્યક્ષમતા	સિસ્ટમ ગુણવત્તા લક્ષણો
ઉદાહરણો	લોગિન, સર્ચ, ગણતરી	પ્રદર્શન, સુરક્ષા, ઉપયોગિતા
પરીક્ષણ	કાર્યકારી પરીક્ષણ	પ્રદર્શન પરીક્ષણ

Functional Requirements:

- વપરાશકર્તા ક્રિયાપ્રતિક્રિયાઓ: લોગિન, નોંધણી, ડેટા એન્ટ્રી
- વ્યવસાય નિયમો: વેલિડેશન નિયમો, ગણતરીઓ
- સિસ્ટમ સુવિધાઓ: રિપોર્ટ્સ, સૂચનાઓ, વર્કફ્લો
- ડેટા પ્રોસેસિંગ: CRUD operations

Non-Functional Requirements:

કોષ્ટક 13. Non-Functional Requirement પ્રકારો

પ્રકાર	વર્ણન	ઉદાહરણ
Performance	ઝડપ અને પ્રતિભાવ	Response time < 2 સેકન્ડ
Security	ડેટા સુરક્ષા	Encrypted ડેટા ટ્રાન્સમિશન
Usability	વપરાશકર્તા અનુભવ	શીખવા માટે સરળ ઈન્ટરફેસ
Reliability	સિસ્ટમ વિશ્વસનીયતા	99.9% અપટાઇમ
Scalability	વૃદ્ધિ સંચાલન	1000+ યુઝર્સ સપોર્ટ

મેમરી ટ્રીક

"Performance Security Usability Reliability Maintainability"

પ્રશ્ન 3(a) OR [3 ગુણ]

SRS નું મહત્વ સમજાવો.

જવાબ

SRS (સોફ્ટવેર આવશ્યકતા વિશિષ્ટતા) એક મહત્વપૂર્ણ દસ્તાવેજ છે જે વ્યાખ્યાયિત કરે છે કે સોફ્ટવેરે શું કરવું જોઈએ.

કોષ્ટક 14. SRS નું મહત્વ

પાસા	લાભ
સ્પષ્ટ સંવાદ	બધા સ્ટેકહોલ્ડર્સ આવશ્યકતાઓ સમજે છે
પ્રોજેક્ટ આયોજન	અંદાજ અને શેડ્યુલિંગ માટે આધાર
ગુણવત્તા ખાતરી	પરીક્ષણ માટે પાયો
ફેરફાર વ્યવસ્થાપન	નિયંત્રિત આવશ્યકતા ફેરફારો
કાનૂની સુરક્ષા	કરાર સંદર્ભ દસ્તાવેજ

મેમરી ટ્રીક

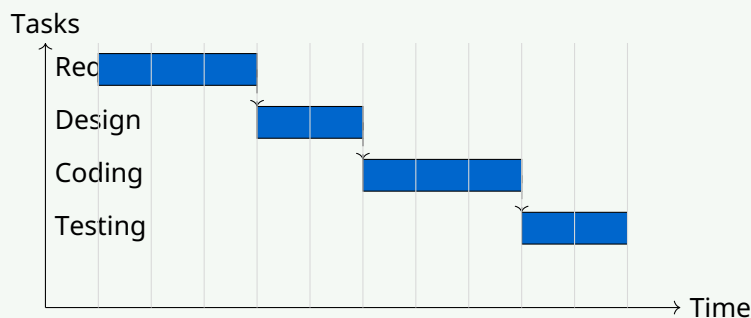
"Clear Planning Quality Change Legal"

પ્રશ્ન 3(b) OR [4 ગુણ]

Gantt Chart સમજાવો.

જવાબ

Gantt Chart કાર્યો, સમયરેખા અને અવલંબન દર્શાવતું વિઝ્યુઅલ પ્રોજેક્ટ મેનેજમેન્ટ સાધન છે.



આકૃતિ 7. સરળ Gantt Chart

કોષ્ટક 15. Gantt Chart ઘટકો

ઘટક	વર્ણન
Tasks	પૂર્ણ કરવાના કામો
Timeline	આડી સમયરેખા
Bars	કાર્ય અવધિ અને પ્રગતિ
Dependencies	કાર્ય સંબંધો
Milestones	મહત્વપૂર્ણ પ્રોજેક્ટ ઘટનાઓ

મેમરી ટ્રીક

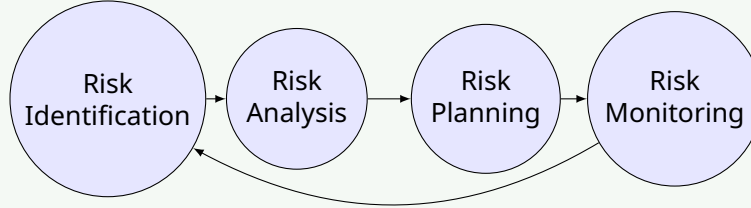
“Tasks Timeline Bars Dependencies Milestones”

પ્રશ્ન 3(c) OR [7 ગુણ]

Risk Management પર ટૂંક નોંધ લખો.

જવાબ

Risk Management એ પ્રોજેક્ટ જોખમોને ઓળખવા, વિશ્લેષણ કરવા અને નિયંત્રિત કરવાની વ્યવસ્થિત પ્રક્રિયા છે.



આકૃતિ 8. Risk Management ચક્ર

કોષ્ટક 16. Risk Management પ્રક્રિયા

તબક્કો	પ્રવૃત્તિઓ	આઉટપુટ
Identification	સંભવિત જોખમો શોધો	Risk list
Analysis	સંભાવના અને અસરનું મૂલ્યાંકન	Risk priority
Planning	પ્રતિભાવ વ્યૂહરચના વિકસાવો	Risk response plan
Monitoring	જોખમો ટ્રેક અને કંટ્રોલ કરો	Updated risk status

Risk શ્રેણીઓ:

કોષ્ટક 17. સોફ્ટવેર જોખમોના પ્રકારો

શ્રેણી	ઉદાહરણો
Technical	ટેકનોલોજી ફેરફારો, જટિલતા
Project	શિડ્યુલ વિલંબ, સંસાધન અછત
Business	બજાર ફેરફારો, ભંડોળ મુદ્દાઓ
External	વેન્ડર સમસ્યાઓ, નિયમનકારી ફેરફારો

Risk પ્રતિભાવ વ્યૂહરચનાઓ:

- Avoid: જોખમ સ્ત્રોત દૂર કરો

- **Mitigate:** સંભાવના અથવા અસર ઘટાડો
- **Transfer:** જોખમ અન્ય સાથે શેર કરો
- **Accept:** જોખમ સાથે જીવો

મેમરી ટ્રીક

"Identify Analyze Plan Monitor (Process), Avoid Mitigate Transfer Accept (Strategies)"

પ્રશ્ન 4(a) [3 ગુણ]

Size estimation માટે મેટ્રિક શું છે? ઉદાહરણ સાથે FP સમજાવો.

જવાબ

Size Estimation Metrics સોફ્ટવેર પ્રોજેક્ટ કદ અને પ્રયત્નોની આગાહી કરવામાં મદદ કરે છે.

કોષ્ટક 18. Size Estimation Metrics

મેટ્રિક	વર્ણન
LOC	Lines of Code
Function Points	Functionality-આધારિત માપદંડ
Object Points	Object-oriented સિસ્ટમ્સ માટે
Feature Points	ઉચ્ચત ફંક્શન પોઈન્ટ્સ

Function Points (FP) વપરાશકર્તા કાર્યક્ષમતાના આધારે સોફ્ટવેર કદને માપે છે.

FP ઘટકો:

- **External Inputs:** ડેટા એન્ટ્રી સ્ક્રીન્સ
- **External Outputs:** રિપોર્ટ્સ, સંદેશાઓ
- **External Queries:** ડેટાબેઝ પ્રશ્નો
- **Internal Files:** ડેટા સ્ટોર્સ
- **External Interfaces:** સિસ્ટમ કનેક્શન્સ

FP ગણતરી ઉદાહરણ: Library Management System માટે:

- External Inputs: 5 (Book entry, Member entry, etc.)
- External Outputs: 3 (Reports)
- External Queries: 4 (Search functions)
- Internal Files: 2 (Book DB, Member DB)
- External Interfaces: 1 (Online catalog)

Simple FP = 5 + 3 + 4 + 2 + 1 = 15 Function Points

મેમરી ટ્રીક

"Inputs Outputs Queries Files Interfaces"

પ્રશ્ન 4(b) [4 ગુણ]

Basic COCOMO મોડેલનો ઉપયોગ કરીને પ્રોજેક્ટ અંદાજ તકનીકો સમજાવો.

જવાબ

COCOMO (COConstructive COst MOdel) સોફ્ટવેર વિકાસ પ્રયત્ન અને શેડ્યુલનો અંદાજ કાઢે છે.

કોષ્ટક 19. COCOMO મોડેલ પ્રકારો

પ્રકાર	વર્ણન	ચોક્કસાઈ
Basic	સરળ કદ-આધારિત અંદાજ	$\pm 75\%$
Intermediate	કિંમત ડ્રાઇવરો સમાવે છે	$\pm 25\%$
Detailed	તબક્કા-સ્તર અંદાજ	$\pm 10\%$

Basic COCOMO ફોર્મ્યુલા:

- $Effort = a \times (KLOC)^b$ person-months
- $Time = c \times (Effort)^d$ months
- $People = Effort / Time$

કોષ્ટક 20. COCOMO Constants

Project Type	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

મેમરી ટ્રીક

“Organic Semi Embedded”

પ્રશ્ન 4(c) [7 ગુણ]

તમારી પસંદગીની સિસ્ટમ માટે Sprint burn down chart તૈયાર કરો.

જવાબ

Sprint Burn Down Chart sprint દરમિયાન બાકી રહેલા કામને ટ્રેક કરે છે. ઉદાહરણ: **Online Shopping System.**
Sprint Goal: User Authentication Module (40 Story Points, 2 Weeks Estimate)



આકૃતિ 9. Sprint Burn Down Chart

Chart Analysis:

- **Green line:** Ideal burn down
- **Red line:** Actual progress
- **પૂર્ણતા:** Sprint સમયસર પૂર્ણ થયું

ફાયદા: વિઝ્યુઅલ પ્રગતિ ટ્રેકિંગ, પ્રારંભિક સમસ્યા ઓળખ, ટીમ પ્રેરણા

મેમરી ટ્રીક

"Track Progress Daily, Identify Issues Early"

પ્રશ્ન 4(a) OR [3 ગુણ]

USE CASE ડાયાગ્રામના ઘટક સમજાવો.

જવાબ

Use Case Diagram વપરાશકર્તા પરિપ્રેક્ષ્યમાં સિસ્ટમ કાર્યક્ષમતા દર્શાવે છે.

કોષ્ટક 21. Use Case Diagram ઘટકો

ઘટક	પ્રતીક	વર્ણન
Actor	Stick figure	બાહ્ય એન્ટિટી
Use Case	Oval	સિસ્ટમ કાર્યક્ષમતા
System Boundary	Rectangle	સિસ્ટમ સ્કોપ
Association	Line	Actor-Use Case સંબંધ
Generalization	Arrow	વારસાગત સંબંધ

સંબંધો:

- **Include:** એક use case બીજાને સમાવે છે (ફરજિયાત)
- **Extend:** વૈકલ્પિક use case વિસ્તરણ

મેમરી ટ્રીક

"Actors Use Systems, Associate Generally"

પ્રશ્ન 4(b) OR [4 ગુણ]

Cohesion અને Coupling સરખાવો.

જવાબ

Cohesion અને Coupling જાળવણીક્ષમતા પર અસર કરતા મહત્વપૂર્ણ સોફ્ટવેર ડિઝાઇન સિદ્ધાંતો છે.

કોષ્ટક 22. Cohesion વિ Coupling

પાસા	Cohesion	Coupling
વ્યાખ્યા	મોડ્યુલની અંદર એકતા	મોડ્યુલો વચ્ચે અવલંબન
ઇચ્છનીય સ્તર	ઉચ્ચ (High)	ઓછું (Low)
ફોકસ	આંતરિક મોડ્યુલ એકતા	આંતર-મોડ્યુલ સંબંધો
અસર	મોડ્યુલ વિશ્વસનીયતા	સિસ્ટમ લવચીકતા

લક્ષ્ય: High Cohesion + Low Coupling = Good Design

મેમરી ટ્રીક

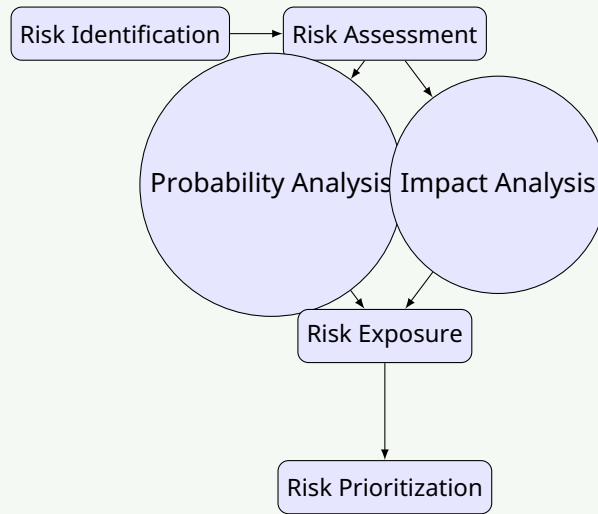
“High Cohesion, Low Coupling”

પ્રશ્ન 4(c) OR [7 ગુણ]

Risk Assessment વિગતવાર સમજાવો.

જવાબ

Risk Assessment મેનેજમેન્ટ પ્રયત્નોને પ્રાથમિકતા આપવા ઓળખાયેલા જોખમોનું મૂલ્યાંકન કરે છે.



આકૃતિ 10. Risk Assessment પ્રક્રિયા

કોષ્ટક 23. Risk Assessment ઘટકો

ઘટક	વર્ણન	સ્કેલ
Probability	જોખમની સંભાવના	0.1 to 1.0
Impact	જોખમ થાય તો પરિણામો	1 to 10
Risk Exposure	Probability × Impact	ગણતરી કરેલ કિંમત
Risk Level	પ્રાથમિકતા વર્ગીકરણ	High/Medium/Low

મૂલ્યાંકન પ્રક્રિયા: 1. Probability Assessment: 0.1 (Very Low) - 0.9 (Very High) 2. Impact Assessment: 1-10 સ્કેલ 3.

Risk Exposure: $P \times I$

Risk Matrix: High (> 4.0), Medium ($2.0-4.0$), Low (< 2.0)

મેમરી ટ્રીક

“Probability Impact Exposure Priority”

પ્રશ્ન 5(a) [3 ગુણ]

કોડ રિવ્યૂ માં Code Inspection તકનીક સમજાવો.

જવાબ

Code Inspection એ ખામીઓ શોધવા માટે કોડની ઔપચારિક, વ્યવસ્થિત તપાસ છે.

કોષ્ટક 24. Code Inspection પ્રક્રિયા

તબક્કો	સહભાગીઓ	પ્રવૃત્તિઓ
Planning	Moderator	શેડ્યુલ, કોડ વિતરણ
Overview	Author, Team	Author કોડ સમજાવે છે
Preparation	Individual	રિવ્યુઅર્સ કોડ અભ્યાસ કરે છે
Inspection	All reviewers	ખામીઓ શોધવી
Rework	Author	ખામીઓ સુધારવી
Follow-up	Moderator	સુધારાઓ ચકાસવા

મેમરી ટ્રીક

“Plan Overview Prepare Inspect Rework Follow-up”

પ્રશ્ન 5(b) [4 ગુણ]

ATM ના ઓછામાં ઓછા ચાર Test Cases તૈયાર કરો.

જવાબ

ATM Test Cases:

કોષ્ટક 25. ATM Test Cases

ID	પરિસ્થિતિ	ઇનપુટ	આઉટપુટ	પરિણામ
TC1	Valid PIN	સાચો PIN	એક્સેસ મંજૂર	Pass
TC2	Invalid PIN	ખોટો PIN	કાર્ડ બ્લોક	Pass
TC3	Withdrawal	રકમ \leq બેલેન્સ	રોકડ વિતરણ	Pass
TC4	Low Balance	રકમ $>$ બેલેન્સ	નકારવામાં આવ્યું	Pass

મેમરી ટ્રીક

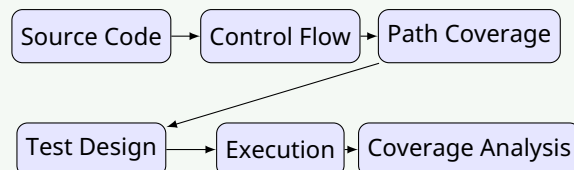
“Login Withdraw Inquiry Change”

પ્રશ્ન 5(c) [7 ગુણ]

White box testing વર્ણવો.

જવાબ

White Box Testing આંતરિક કોડ માળખું અને લોજિક પાથ તપાસે છે.



આકૃતિ 11. White Box Testing પ્રક્રિયા

Coverage Criteria:

- **Statement Coverage:** દરેક સ્ટેટમેન્ટ એક્ઝિક્યુટ કરો
- **Branch Coverage:** બધા if-else પાથ એક્ઝિક્યુટ કરો
- **Path Coverage:** દરેક શક્ય પાથ એક્ઝિક્યુટ કરો
- **Condition Coverage:** બધી શરતો (true/false) ટેસ્ટ કરો

ફાયદા: સંપૂર્ણ પરીક્ષણ, લોજિક ભૂલો શોધે છે

ગેરફાયદા: સમય માંગી લે છે, મોંઘું, કોડ આધારિત

મેમરી ટ્રીક

“Statement Branch Path Condition”

પ્રશ્ન 5(a) OR [3 ગુણ]

કોડ રિવ્યૂ માં Code Walk Through તકનીક સમજાવો.

જવાબ

Code Walk Through એ અનૌપચારિક કોડ રિવ્યૂ તકનીક છે જ્યાં લેખક ટીમને કોડ રજૂ કરે છે.

કોષ્ટક 26. Walk Through પ્રક્રિયા

તબક્કો	વર્ણન	સમય
Preparation	લેખક પ્રસ્તુતિ તૈયાર કરે છે	30 min
Presentation	લેખક કોડ લોજિક સમજાવે છે	1-2 hours
Discussion	ટીમ પ્રશ્નો પૂછે છે	30 min
Documentation	મુદ્દાઓ રેકૉર્ડ કરો	15 min

મેમરી ટ્રીક

“Prepare Present Discuss Document”

પ્રશ્ન 5(b) OR [4 ગુણ]

સોફ્ટવેર દસ્તાવેજીકરણ સમજાવો.

જવાબ

સોફ્ટવેર દસ્તાવેજીકરણ વિવિધ સ્ટેકહોલ્ડર્સ માટે સોફ્ટવેર સિસ્ટમ વિશે માહિતી પ્રદાન કરે છે.

કોષ્ટક 27. દસ્તાવેજીકરણ પ્રકારો

પ્રકાર	હેતુ	પ્રેક્ષકો
User Doc	સોફ્ટવેર કેવી રીતે વાપરવું	End users
System Doc	ટેકનિકલ વિગતો	Developers
Process Doc	વિકાસ પ્રક્રિયા	Project team
Requirements	સિસ્ટમે શું કરવું જોઈએ	Stakeholders

Internal: Code comments, README.

External: User manuals, API docs.

મેમરી ટ્રીક

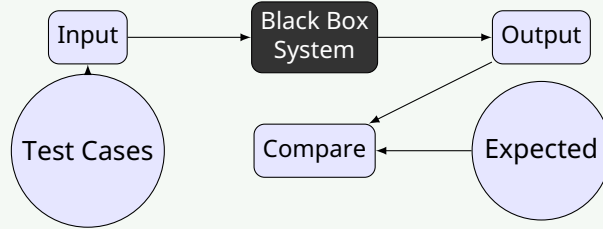
"User System Process Requirements"

પ્રશ્ન 5(c) OR [7 ગુણ]

Black box testing પર ટૂંક નોંધ લખો.

જવાબ

Black Box Testing આંતરિક કોડ માળખાના જ્ઞાન વિના સોફ્ટવેર કાર્યક્ષમતા તપાસે છે.



આકૃતિ 12. Black Box Testing ખ્યાલ

કોષ્ટક 28. Testing તકનીકો

તકનીક	વર્ણન	ઉદાહરણ
Equivalence Partitioning	Valid/Invalid વર્ગો	Age: 0-17, 18-65, >65
Boundary Value	સીમાઓ પર પરીક્ષણ	Age: 17, 18, 65, 66
Decision Table	જટિલ નિયમો	વીમા પ્રીમિયમ
State Transition	સ્થિતિ ફેરફારો	ATM states

સરખામણી:

- User perspective vs Code perspective
- No code knowledge needed vs Programming skills
- Early testing vs Testing after code

મેમરી ટ્રીક

"Equivalence Boundary Decision State"