ĐỀ 2 AGILE SCRUM

**Request 1**

Scrum master : A Scrum Master chiefly acts as a coach and facilitator to the Scrum Team

BA: bridges the gap between external stakeholders and the development team, interpreting business requirements into understandable development tasks to match a final software product with the expected business value

Developer team : internal stakeholder , He writes, debugs and executes the source code of a software application.

Designer team : internal stakeholder A UI/UX designer’s job is to create user-friendly interfaces that enable users to understand how to use complex technical products

Tester team : Internal stakeholder. The role of a tester is to test out products for bug and provide reports to the project teams about any issues or improvements that the product may require

**Request 2 :** User stories in product backlog

As an administrator, I want to see more health related data in our employee profile in order to manage them better and give them proper job

As a team member , I want to log-in through/via my email account in order to safe time while logging in with no need to fill in the password

**Request 3 -** Project Management System – level 1

1. Project Initiation – level 2

1.1 Define project scope and objectives – level 3

1.2 Identify stakeholders and their requirements– level 3

1.3 Create project charter– level 3

- Define project scope and objectives – level 4

1.4 Define project team roles and responsibilities– level 3

1.5 Conduct project kickoff meeting– level 3

2. Sprint 0: Planning and Setup– level 2

2.2 Create product backlog and prioritize features– level 3

2.3 Define sprint cycles and timelines– level 3

2.4 Set up development environment and tools– level 3

2.5 Define Definition of Done (DoD) – level 3

2.6 Develop release plan– level 3

3. Sprint 1– level 2

3.1 Sprint planning meeting– level 3

3.2 Develop user stories for sprint – level 3

3.3 Define sprint backlog and tasks– level 3

3.4 Develop software architecture and design– level 3

3.5 Implement core functionalities of DTS– level 3

3.6 Conduct daily scrum meetings– level 3

3.7 Test and validate sprint deliverables– level 3

3.8 Conduct sprint review and retrospective– level 3

4. Sprint 2– level 2

4.1 Sprint planning meeting– level 3

4.2 Develop user stories for sprint– level 3

4.3 Define sprint backlog and tasks– level 3

4.4 Implement additional functionalities of DTS– level 3

4.5 Conduct daily scrum meetings– level 3

4.6 Test and validate sprint deliverables– level 3

4.7 Conduct sprint review and retrospective– level 3

5. Sprint 3– level 2

5.1 Sprint planning meeting– level 3

5.2 Develop user stories for sprint– level 3

5.3 Define sprint backlog and tasks– level 3

5.4 Implement remaining functionalities of DTS– level 3

5.5 Conduct daily scrum meetings– level 3

5.6 Test and validate sprint deliverables– level 3

5.7 Conduct sprint review and retrospective– level 3

6. Sprint 4: Release and Deployment– level 2

6.1 Conduct final sprint planning meeting– level 3

6.2 Test and validate final release– level 3

6.3 Conduct final sprint review and retrospective– level 3

6.4 Deploy DTS to production environment– level 3

6.5 Conduct post-release testing and monitoring– level 3

6.6 Close out project– level 3

Request 4

A – 4

C – 2

B – 3

Start 0

D – 5

E – 4

H– 5

F – 2

I – 9

G – 6

End

path 1 : Start A -> B -> C-> G-> End 4+3+2+6 = 15weeks

path 2 : Start D -> B -> C-> G-> End 5+3+2+6 = 16weeks

path 3 : Start D -> E -> F-> G-> End 5+4+2+6 = 17weeks

path 4 : Start D -> H -> I -> End 5+5+9 = 19weeks => critical path , 19weeks are the minimum duration to complete that deliverable

Explain: To reduce the overall project schedule by 3 weeks, we will focus on reducing the amount in the critical path. But after being reduced to 16 weeks , there is 1 more path that is exceeding this number which is path 3 – 17 weeks. On the other hand, path 3 and critical path both have task D . Obviously , we will have something to do with task D under crashing method

Crashing: In this method , we can try to shorten the duration of some of the critical path activities by adding additional resources or working overtime.

Solution1 : we can add more workers to task D to complete it earlier than 3 weeks

Solution 2: we can force the team to work overtime in task D so that it can be completed earlier than 3 weeks.

Solution 3 : reward the team if they can complete task D earlier than 3 weeks

* Crashing will increase cost .

Request 5

BAC ( Budget at completion) – 5000$

DAC – 7 months

After 5 months

AC ( actual cost ) - 6500$

PV ( Planned value ) = 5000/7 x 5= 3571$

EV (earned value) = 5000 x 80% = 4000$

CV ( cost variance) = EV – AC = 4000-6500 = -2500 < 0 => over budget

SV ( schedule variance) = EV – PV = 4000 – 3571 = 429 >0 => ahead of schedule

CPI ( cost performance index ) = EV/AC = 0,6 < 1 => over budget

SPI ( schedule performance index ) = EV /PV = 4000/3571 = 1,12 >1 => ahead of schedule

EAC ( estimate at completion ) = BAC/CPI = 5000/0,6 = 8333 $

EDAC ( estimate duration at completion ) = DAC/SPI = 7 / 1,12 = 6,25 months

So this project is ahead of schedule but over budget

Behind schedule -> apply fast tracking method, crashing method , reduce the project scope

Over budget -> cut down on human resource , reduce the project scope

On schedule

On budget

Ahead of schedule

Under budget

DE 1 ITERATIVE INCREMENTAL

Request 1 – ĐỀ 1

Project manager: Internal stakeholder, planning and organizing, managing tasks, budgeting, controlling costs and other factors. Everything they do helps make sure the project can be completed on time and on budget, and more importantly, profitable.

BA: bridges the gap between external stakeholders and the development team, interpreting business requirements into understandable development tasks to match a final software product with the expected business value

Developer team : internal stakeholder , He writes, debugs and executes the source code of a software application.

Designer team : internal stakeholder A UI/UX designer’s job is to create user-friendly interfaces that enable users to understand how to use complex technical products

Tester team : Internal stakeholder. The role of a tester is to test out products for bug and provide reports to the project teams about any issues or improvements that the product may require

2 Request 2 Project scope statement

Project in-scope item :

* Our team will develop an website Project management system PMS to support project managers, team members and other project stakeholders on following management area scope, cost , quality, communication, schedule, issue
* This website will have 4 main releases which respectively provide the system functionalities for 4 roles: system admin, team member, customer , project manager . 4 main releases will include about 30 usecase

Project deliverables :

* project schedule
* project charter
* project management plan (scope, time, quality, resource, communication, risks, stakeholder, procurement )
* test plan,
* code package for module 1-2-3,
* meeting minutes,
* project final report

Out of scope item :

* This project will not include payment to external vendors

REQUEST 3

Request 3:

1. Software Project Management System

1.1 Phase 1: Initiating (level 1)

1.1.1 Create Project Charter (level 2)

1.1.1.1 Solidifying the project scope (level 3)

1.1.1.2 Determine resources needed (level 3)

1.1.1.3Define project SMART goal and deliverables (level 3 )

1.1.2 Hold a kick-off meeting (level 2)

1.1.3 Get project charter approval (level 2)

1.1.4 Perform a cost - benefit analysis for the project- - level 2

1.1.5 Create stakeholder Register (level 2)

1.2 Phase 2: Planning (level 1)

1.2.1 Create Project management plan ( level 2 )

1.2.1.1 Create Scope Management Plan (level 3)

1.2.1.2 Create Time Management Plan (level 3)

1.2.1.3 Create Cost Management Plan (level 3)

1.2.1.4 Create Risk Management Plan (level 3)

1.2.1.5 Create Resource Management Plan (level 3)

1.2.2 Meeting with team to discuss about plans (level 2)

1.2.3 Deliver Project Management Plan (level 2)

1.3 Phase 3 Executing (level 1) (Analysis , Design, Code , Test )

Iteration 1 – Level 2

1.3.1 Analysis (level 3)

1.3.1.1 Create Feasibility Report (level 4)

1.3.1.2 Create Use-case diagram and use-case description (level 4)

1.3.1.3 Create Software Requirement Specification (level 4)

1.3.1.4 Perform Requirement validation (level 4)

1.3.1.5 Perform Requirement management (level 4)

1.3.2 Design (level 3)

1.3.2.1 Create Architecture design (level 4 )

1.3.2.2 Create detail design (level 4 )

1.3.3 Implementing (level 3)

1.3.3.1 Implement coding for module 1 : User management (level 4) (usecase 1 to 10 )

1.3.4 Testing (level 3)

1.3.4.1 Perform unit testing (level 4)

1.3.4.2 Perform integration testing (level 4)

**Iteration 2 – level** 2

1.2.1 Analysis (level 3)

1.2.1.1 Update Feasibility Report (level 4)

1.2.1.2 Update Use-case diagram and use-case description (level 4)

1.2.1.3 Update Software Requirement Specification (level 4)

1.2.2 Design (level 3)

1.2.2.1 Update Architecture design (level 4 )

1.2.2.2 Update detail design (level 4 )

1.2.3 Implementing (level 3)

1.2.3.1 Update coding for module 1 : (level 4)

1.2.3.2 Implement coding for module 2 : (level 4)

1.2.3.3 Implement coding for module 3 : (level 4)

1.2.4 Testing (level 3)

1.2.4.1 Perform unit testing (level 4)

1.2.4.2 Perform integration testing (level 4)

1.2.4.3 Perform part of system testing (level 4)

**Iteration 3 – level 2**

1.2.1 Analysis (level 3)

1.2.1.1 Update Feasibility Report (level 4)

1.2.1.2 Update Use-case diagram and use-case description (level 4)

1.2.1.3 Update Software Requirement Specification (level 4)

1.2.2 Design (level 3)

1.2.2.1 Update Architecture design (level 4 )

1.2.2.2 Update detail design (level 4 )

1.2.3 Implementing (level 3)

1.2.3.1 Update coding for module 123 : (level 4)

1.2.3.2 Implement coding for module 4 : (level 4)

1.2.4 Testing (level 3)

1.2.4.1 Perform unit testing (level 4)

1.2.4.2 Perform integration testing (level 4)

1.2.4.3 Perform system testing (level 4)

1.2.4.4 Perform acceptance testing

1.3 Phase 4: Monitoring and Controlling (level 1)

1.3.1 Control scope (level 2)

1.3.2 Track progress (level 2)

1.3.3 Perform Cost control (level 2)

1.3.4 Monitor and control Risk (level 2)

1.4 Phase 5: Closing (level 1)

1.5.1 Create Lesson learn (level 2)

1.5.2 Create Project Final Report (level 2)

1.5.3 Create Project Archive (level 2)

1.5.4 Hold close Project Ceremony (level 2)

Request 4

F – 3

C – 2

A – 3

Start 0

End 0

G– 5

D – 3

B – 5

H – 4

E – 5

I – 3

path 1 : Start A -> B -> C-> F->I-> End = 16weeks

path 2 : Start A -> B -> D-> G->I-> End = 19weeks

path 3 : Start A -> B -> D-> H-> End = 15weeks

path 4 : Start A-> B -> E->H -> End = 17weeks

Path 2 is critical path , 19weeks are the minimum duration to complete that deliverable

Explain: To reduce the overall project schedule by 3 weeks, we will focus on reducing the amount in the critical path. But after being reduced to 16 weeks , there is 1 more path that is exceeding this number which is path 4 – 17 weeks. On the other hand, path 4 and critical path both have task B – 5 days which can be reduced . Obviously , we will have something to do with task D under crashing method

Crashing: In this method , we can try to shorten the duration of some of the critical path activities by adding additional resources or working overtime.

Solution1 : we can add more workers to task B to complete it earlier than 3 weeks

Solution 2: we can force the team to work overtime in task B so that it can be completed earlier than 3 weeks.

Solution 3 : reward the team if they can complete task B earlier than 3 weeks

* Crashing will increase cost .

BÀI 5 đề 1

PV – 5000

EV – 3000

AC – 4000

CV = EV – AC = -1000 > vượt chi phí

SV = EV – PV = -2000> chậm tiến độ

* , reduce the project scope