ESSENCE DOCUMENT

Project : DEVELOPER-CONNECTOR

Problem Statement :

A MERN stack based social network web application to connect developers so that they can discuss and get connected with each other to collaborate over various projects.

Objective :

● To make an exclusive social network app dedicated to developers.

● Make them reach other developers and share project ideas , discuss it, like it , view each other's professional profile , projects , see github repos , skill set etc. So that they can later collaborate over projects in various domains.

● To give security to users while login by providing authenticated routes through jwt hashing and salting, etc.

● Keep the U.I. clean.

● To do the state management through redux in order to have alerts , like dislike features , hence making the U.I. and the usability of the app better.

USE CASES:

A use case is all the ways of using a system to achieve a particular goal for a particular
user.Use cases provide a systematic way to organize requirements.

Activity 1&2 : Find actors and use cases & Slice the use cases:

As already explained in SRS our target users are developers across various domains and different age groups.

Use cases and use case slices:

1) Able to register and login

   i)Login

   ii) Register

   iii) Logout

2) Create,update and view profiles

   i) Create and deleted profile

   ii) Add and delete work experience / education

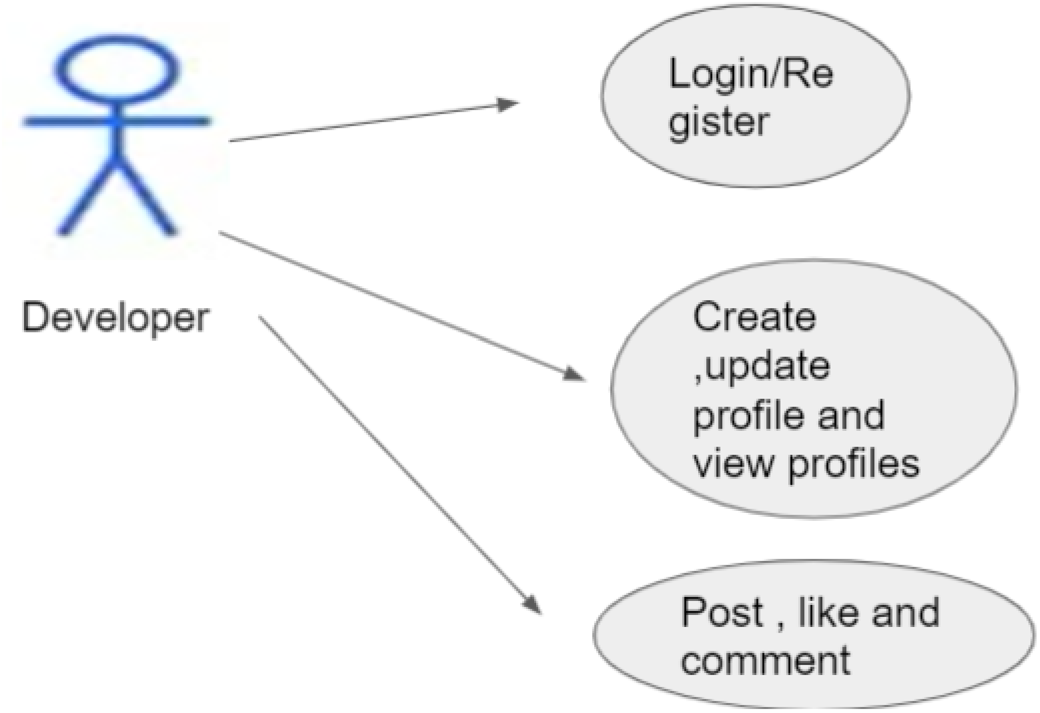
   iii) Able to view other people’s profile

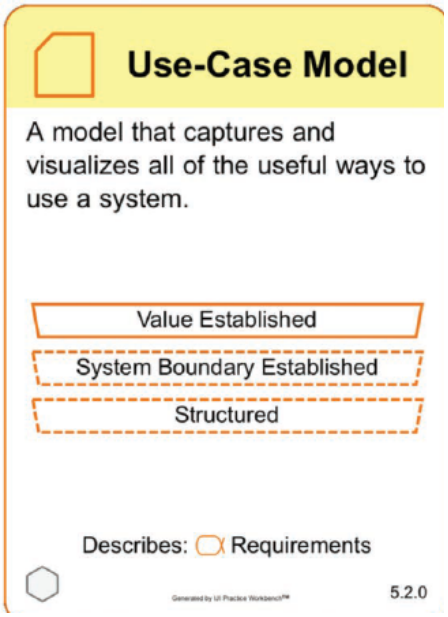
3) Post , like and comment:

i) Post
ii) Like and reply

Work Product:

a) **Use Case Model:**





Use-Case Model

A model that captures and visualizes all of the useful ways to use a system.

Value Established

System Boundary Established

Structured

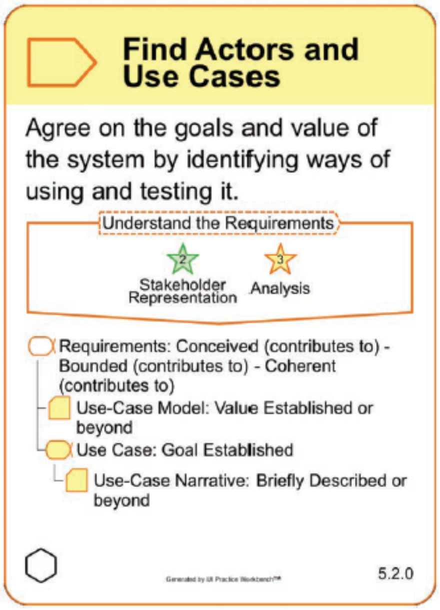
Describes: Requirements

Generated by UI Practice Workbench™

5.2.0

Use case slices are quite straightforward so scope and boundary is clear and use cases are not overlapping so Structured state achieved.

Activity Card:



Find Actors and Use Cases

Agree on the goals and value of the system by identifying ways of using and testing it.

Understand the Requirements

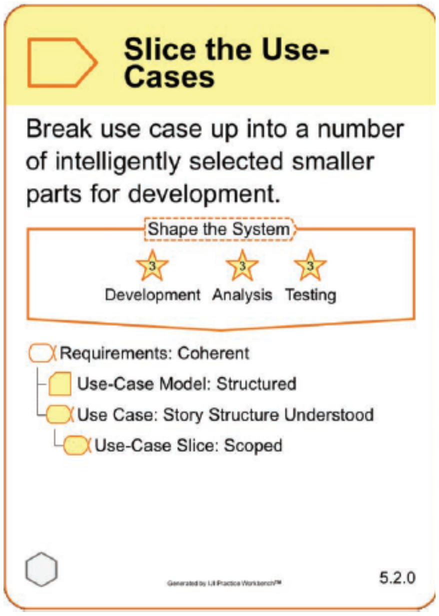
2 Stakeholder Representation

3 Analysis

Requirements: Conceived (contributes to) - Bounded (contributes to) - Coherent (contributes to)

- Use-Case Model: Value Established or beyond
- Use Case: Goal Established
  - Use-Case Narrative: Briefly Described or beyond

Generated by IJI Practice Workbench™ 5.2.0



Slice the Use-Cases

Break use case up into a number of intelligently selected smaller parts for development.

Shape the System

3 Development

3 Analysis

3 Testing

Requirements: Coherent

- Use-Case Model: Structured
- Use Case: Story Structure Understood
  - Use-Case Slice: Scoped

Generated by IJI Practice Workbench™ 5.2.0

After this activity we have achieved mentioned state in our mentioned alphas and work product.

Activity 3: Prepare a use case slice:

In this activity use case narrative is done and test cases are enhanced.

Work Product :

b) Use Case Narrative:

1)Able to register and login:

Basic flow:

1) New users provide name, email , password and confirm password.
2) Old users provide login details that are email/ password.
3) On registering/login is able to view his dashboard.
4) On logout should be logged out.

Alternative Flow:

1)Provide the wrong email id or password.System should not allow login generate alerts.

2)While registering, some fields are missing or passwords and confirm passwords don't match.Then registering shouldn't be allowed.

2)Create , update profile and view profile:

Basic Flow:

1)View a list of developers without or with login.

2)Create a profile with basic information like current professional status, job, bio, skills and github username on login

3)Add education and work experience and delete it.

4) Delete their profile.

Alternative Flow:

1)What if they click on the delete button by mistake , so provide a warning.

2)Giving them options on selecting current professional status.

3)Developers registered first appear first.

3)Post , like and comment:

Basic Flow:

1)Make a post.

2)Like and comment on other people's posts.

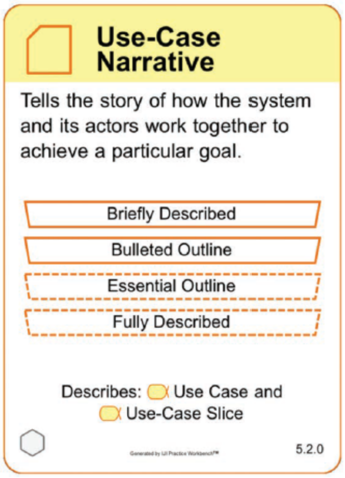
3)Delete your post.

Alternative Flow:

1)Time of posting and commenting getting displayed.

2)Latest post appears first on the post page.

Card



Use-Case Narrative

Tells the story of how the system and its actors work together to achieve a particular goal.

Briefly Described

Bulleted Outline

Essential Outline

Fully Described

Describes: Use Case and Use-Case Slice

Generated by IJI Practice Workbench™

5.2.0

The use cases are essentially outline states as they brief narrative , how system and actor work together and alternative paths are mentioned.

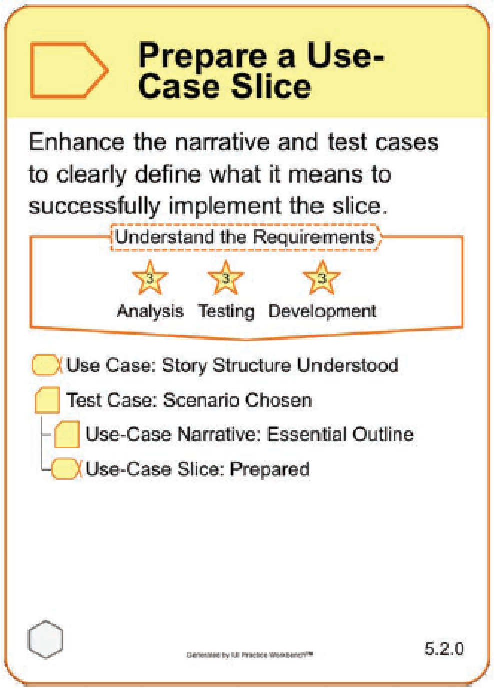
c) Use Case Slice Test Cases:

_ **Use Case Slice Test Cases Work product**

Activity 3: Test a use case slice:

All the use case slice were tested through unit testing on postman and through automation script

Activity card:



Prepare a Use-Case Slice

Enhance the narrative and test cases to clearly define what it means to successfully implement the slice.

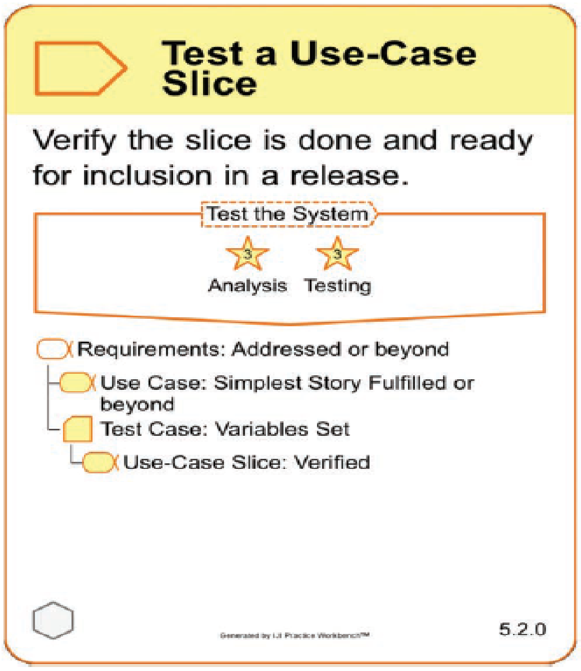
Understand the Requirements

Analysis Testing Development

- Use Case: Story Structure Understood
- Test Case: Scenario Chosen
  - Use-Case Narrative: Essential Outline
  - Use-Case Slice: Prepared

Generated by IJI Practice Workbench™

5.2.0



Test a Use-Case Slice

Verify the slice is done and ready for inclusion in a release.

Test the System

Analysis Testing

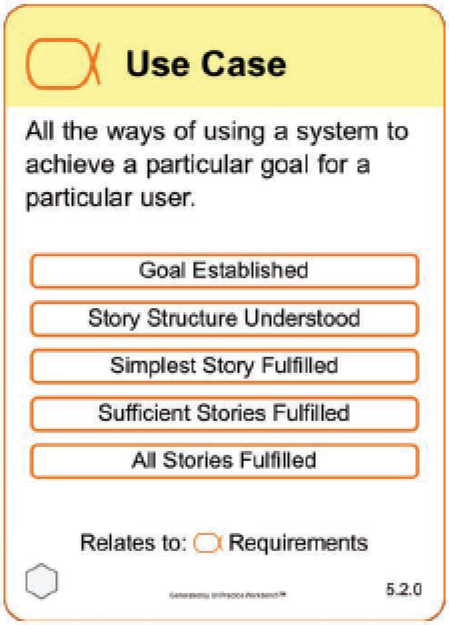
- Requirements: Addressed or beyond
  - Use Case: Simplest Story Fulfilled or beyond
  - Test Case: Variables Set
    - Use-Case Slice: Verified

Generated by IJI Practice Workbench™

5.2.0

After this activity we have achieved mentioned state in our mentioned alphas and work product.

The alphas state achieved:



Use Case

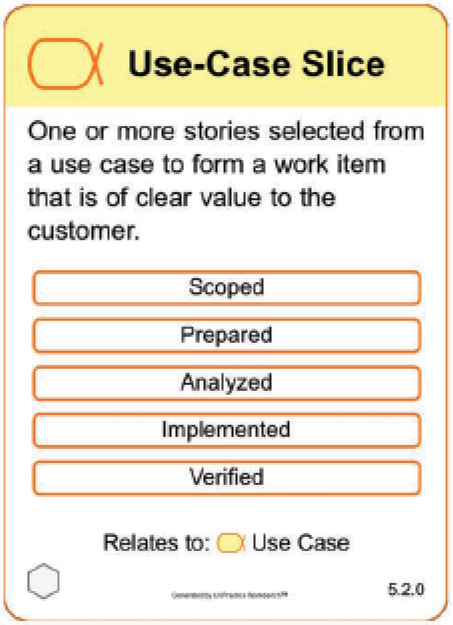
All the ways of using a system to achieve a particular goal for a particular user.

- Goal Established
- Story Structure Understood
- Simplest Story Fulfilled
- Sufficient Stories Fulfilled
- All Stories Fulfilled

Relates to: Requirements

Generated by IJI Practice Workbench™

5.2.0



Use-Case Slice

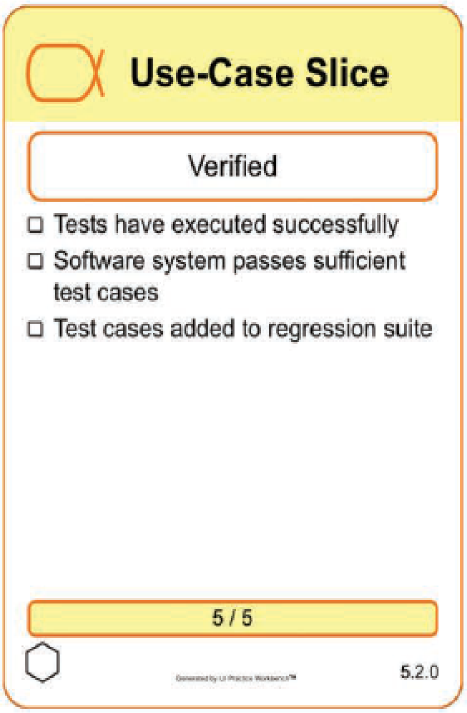
One or more stories selected from a use case to form a work item that is of clear value to the customer.

- Scoped
- Prepared
- Analyzed
- Implemented
- Verified

Relates to: Use Case

Generated by IJI Practice Workbench™

5.2.0



Use-Case Slice

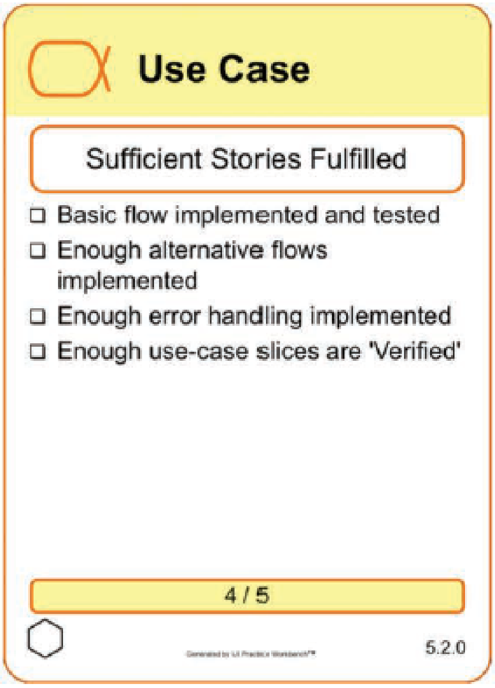
Verified

- □ Tests have executed successfully
- □ Software system passes sufficient test cases
- □ Test cases added to regression suite

5 / 5

Generated by IJI Practice Workbench™

5.2.0



Use Case

Sufficient Stories Fulfilled

- □ Basic flow implemented and tested
- □ Enough alternative flows implemented
- □ Enough error handling implemented
- □ Enough use-case slices are 'Verified'

4 / 5

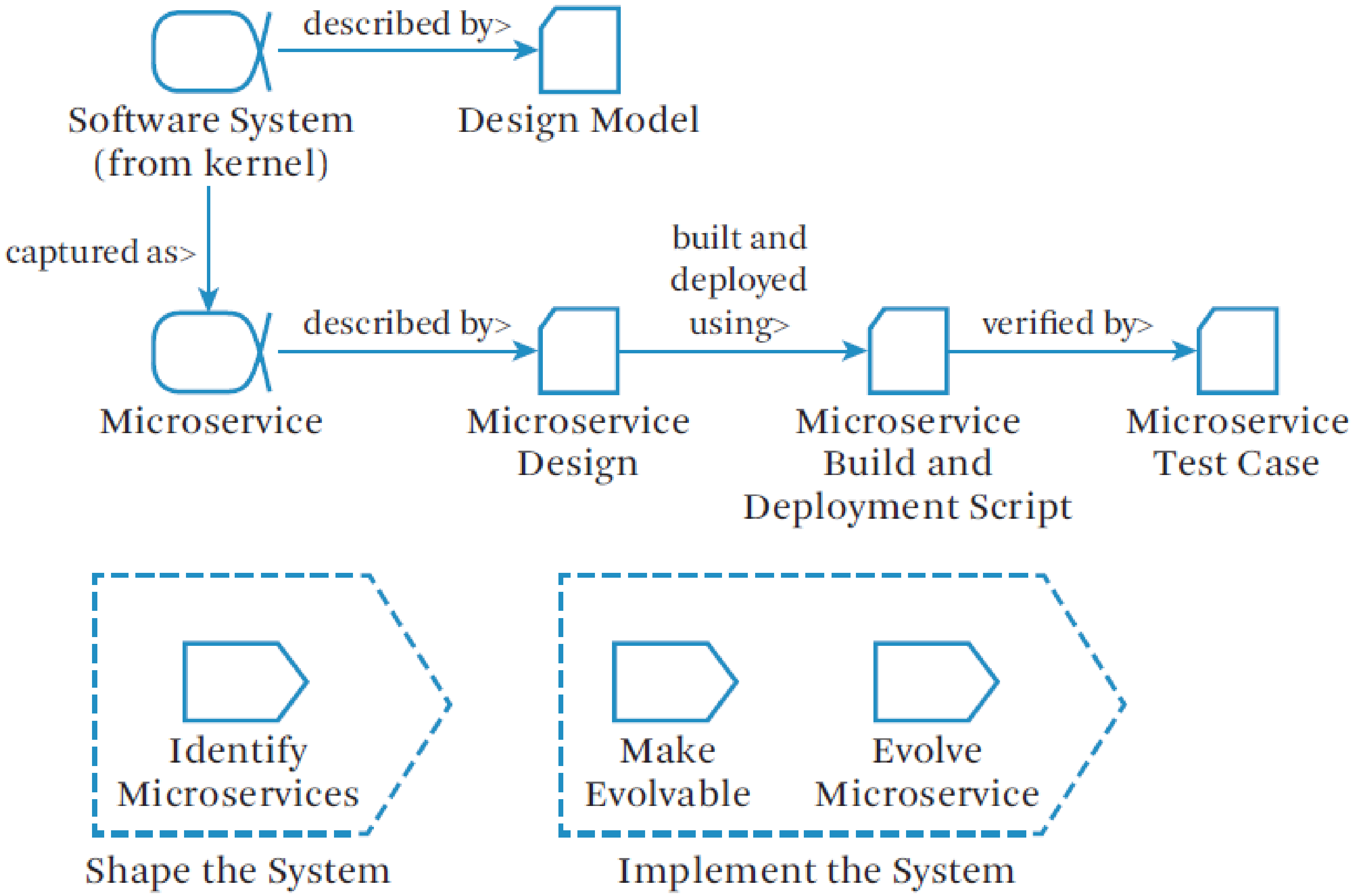
Generated by IJI Practice Workbench™

5.2.0

As we went through the 4 activities we achieved this state.
After 4th activity it is said that use case alpha is in 4th state or beyond.All the use case slice functionality from flow and maximum from alternative flow are tested through postman or automation script.So we assigned 4th state.

KERNEL ALPHA REQUIREMENT: ADDRESSED OR BEYOND. After 4th activity

MICROSERVICES:



Following this:
Activity 1 ,2,3: Identify Microservices, Make microservices evolvable,Evolve Microservice:

Activity 1:

There are 3 database store collections . So there are 2 microservices

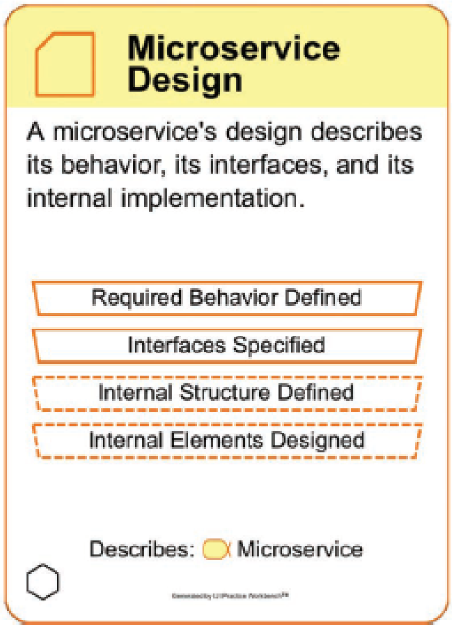
ii) Profile

iii) Posts

Work Product:

Microservice Design and design model: Submitted please refer it.

Cards of work product:



Microservice Design

A microservice's design describes its behavior, its interfaces, and its internal implementation.

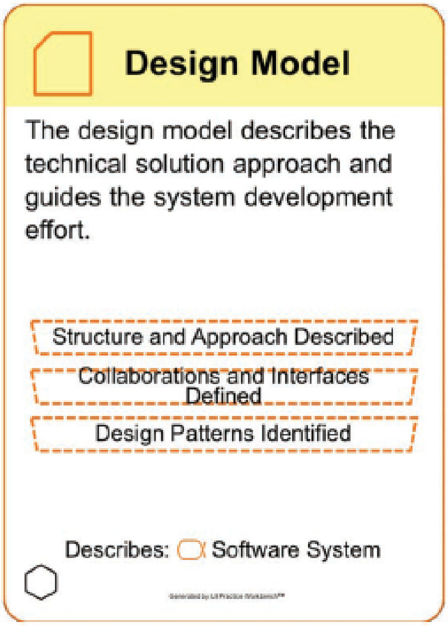
Required Behavior Defined

Interfaces Specified

Internal Structure Defined

Internal Elements Designed

Describes: Microservice



Design Model

The design model describes the technical solution approach and guides the system development effort.

Structure and Approach Described

Collaborations and Interfaces Defined

Design Patterns Identified

Describes: Software System

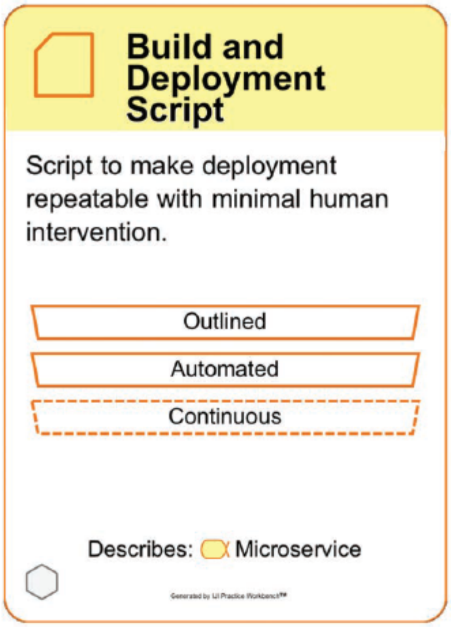
Scope of the microservice has been described in words, the scope of the microservice is
described using interfaces and internal structure of the microservices have been described using 4+1 view model process view and logical view . Hence our work product microservice design has achieved INTERNAL STRUCTURE DEFINED state.(see work product for ref.)

For the design model work product we have defined the structure approach and how microservices are related to each other using interfaces hence 2nd state COLLABORATIONS AND INTERFACES DEFINED is achieved.(see work product for ref.)

Built and deployment script and Microservice test case:

It is there in test and automation script .link: It is there in the readme file.Test case design , readme and automation script (click on this link)

Card:



Build and Deployment Script

Script to make deployment repeatable with minimal human intervention.

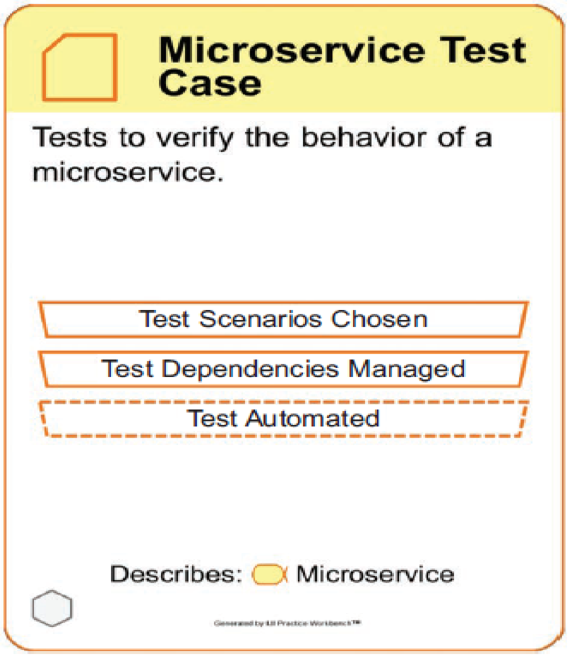
Outlined

Automated

Continuous

Describes: Microservice

Generated by UI Practice Workbench™



Microservice Test Case

Tests to verify the behavior of a microservice.

Test Scenarios Chosen

Test Dependencies Managed

Test Automated

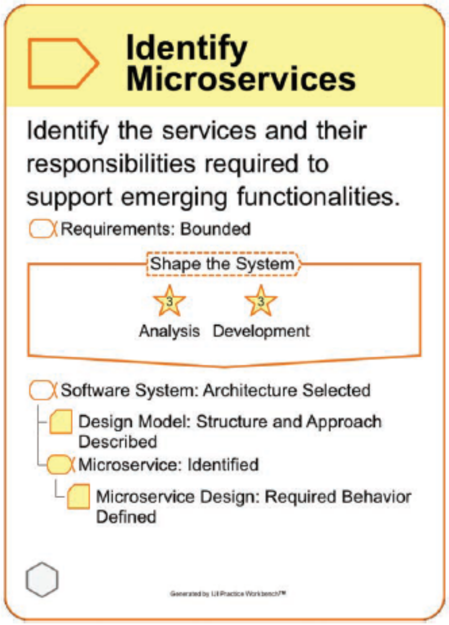
Describes: Microservice

Generated by UI Practice Workbench™

Built and deployment script is outlined as we have steps to it. However we have deployed the whole project at once and individual microservices deployment scripts are not there. Hence not automated.Hence in 1st state achieved.

Test case scenarios are chosen in a shared work product , many of them are automated through mocha and chai library of nodejs.It has been mocked and stubbed.
Mocked” in this case means to create extra test code that simulates the other side of the interface. “Stubbed” means that, instead of simulating the interface, the testers just ensure that the test code will execute without the program crashing.
Yes we have created extra test cases for various mistakes users do.Backend testing scripts and test cases are written. However all the test cases are not automated.
So this work product is in TEST DEPENDENCIES MANAGED STAGE.

After activity 1:



Identify Microservices

Identify the services and their responsibilities required to support emerging functionalities.

Requirements: Bounded

Shape the System

3 Analysis 3 Development

Software System: Architecture Selected

Design Model: Structure and Approach Described

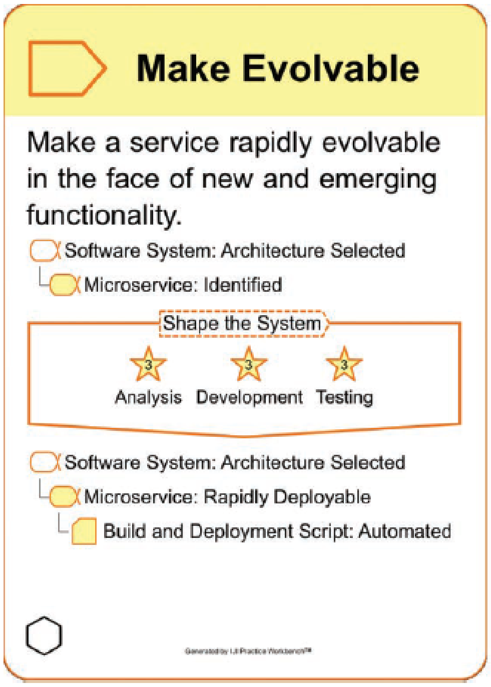
Microservice: Identified

Microservice Design: Required Behavior Defined

Generated by IJI Practice Workbench™

After this activity we have achieved as mentioned state in our mentioned alphas and work product.

Activity 2 & 3: Microservices are made evolvable by developing deployment scripts to build and deploy each microservice individually.
And activity 3 is about incrementally evolving the microservice functionality.



Make Evolvable

Make a service rapidly evolvable in the face of new and emerging functionality.

Software System: Architecture Selected

Microservice: Identified

Shape the System

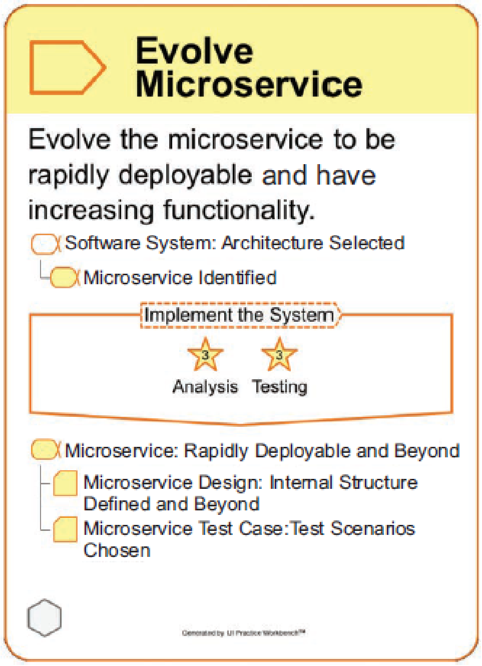
3 Analysis 3 Development 3 Testing

Software System: Architecture Selected

Microservice: Rapidly Deployable

Build and Deployment Script: Automated

Generated by IJI Practice Workbench™



Evolve Microservice

Evolve the microservice to be rapidly deployable and have increasing functionality.

Software System: Architecture Selected

Microservice Identified

Implement the System

3 Analysis 3 Testing

Microservice: Rapidly Deployable and Beyond

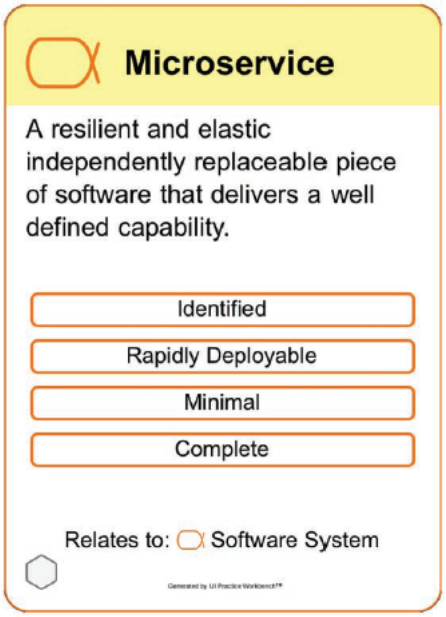
Microservice Design: Internal Structure Defined and Beyond

Microservice Test Case:Test Scenarios Chosen

Generated by IJI Practice Workbench™

Contrasting to what activity gives, the Build and deployment script has achieved outline state.Rest all as mentioned .

The alphas state achieved:



Microservice

A resilient and elastic independently replaceable piece of software that delivers a well defined capability.

Identified

Rapidly Deployable

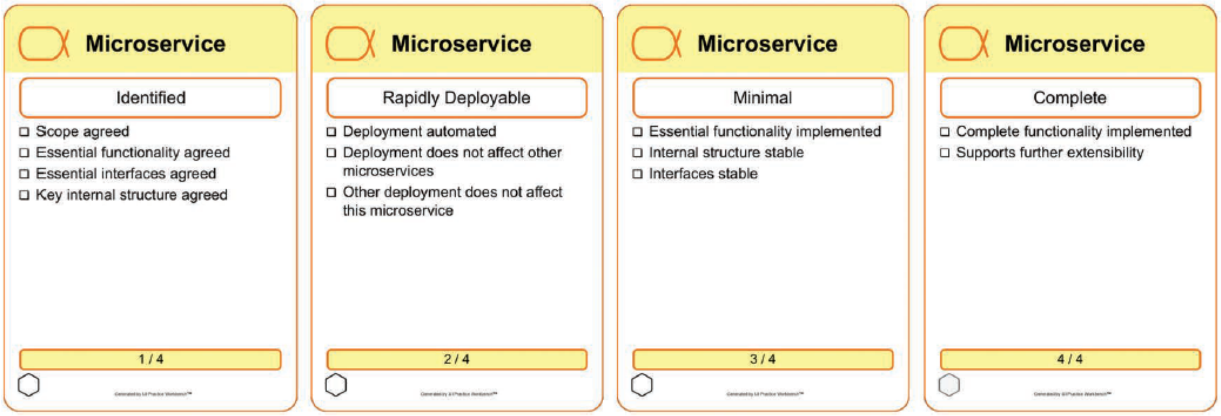
Minimal

Complete

Relates to: Software System

Generated by UI Practice Workbench™

There are 2 microservice alpha



Microservice

Identified

- □ Scope agreed
- □ Essential functionality agreed
- □ Essential interfaces agreed
- □ Key internal structure agreed

1 / 4

Microservice

Rapidly Deployable

- □ Deployment automated
- □ Deployment does not affect other microservices
- □ Other deployment does not affect this microservice

2 / 4

Microservice

Minimal

- □ Essential functionality implemented
- □ Internal structure stable
- □ Interfaces stable

3 / 4

Microservice

Complete

- □ Complete functionality implemented
- □ Supports further extensibility

4 / 4

Microservices were in the identified state after the 1st activity only.(it was justified though work product design doc).
Though our build and deployment script is outlined and not automated in terms of whether we have written a script for individual deployment of microservice alpha or not. However here I will say deployment is

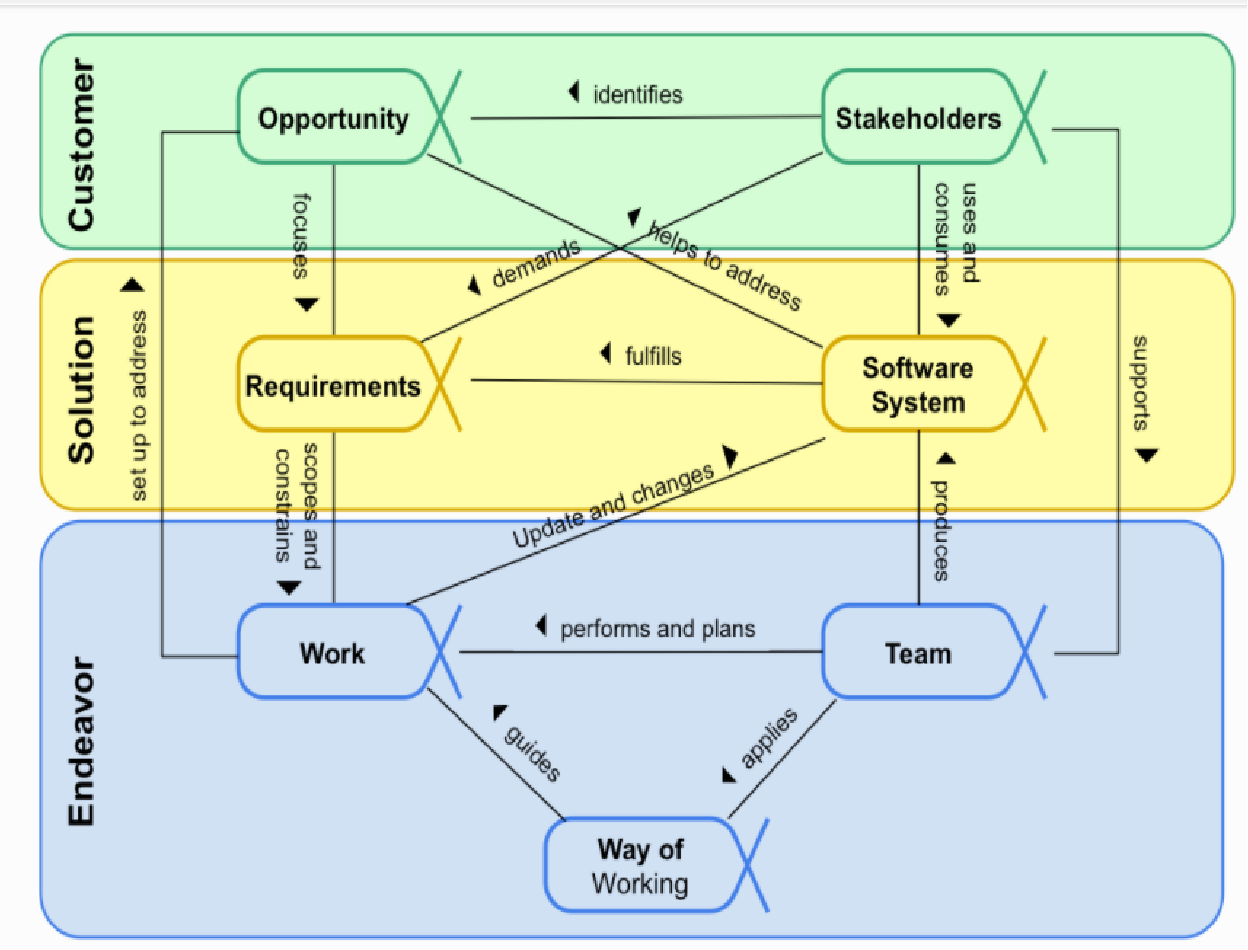
automated as we can run the same script for deployment after adding any functionality.So all functionality implemented hence its in complete state.

The SDLC:

SDLC defines the complete cycle of development i.e. all the tasks involved in planning, creating, testing, and deploying a Software Product. For our project we have used a waterfall model. This model is divided into different phases and the output of one phase is used as the input of the next phase. Every phase has to be completed before the next phase starts and there is no overlapping of the phases.
The sequential phases described in the model are:
1.**Requirement Gathering**: All possible requirements are captured in product requirement documents.
2.**Analysis read**: Using the requirement and based on analysis we define the schemas, models and business rules.
3.**System Design**: Based on analysis we designed the software architecture. We mentioned all the design attributes in software design and architecture document.The architecture used is microservices.
4.**Implementation**: We performed the implementation development of the software in the small units with functional testing. We have mentioned all the test cases in the Test case design document. The code is written in MERN stack. I.e mongodb database , express , nodejs backend and react frontend.
5.**Testing**: We performed testing for all the use case slices and microservices in essence. Hence covering all the requirements in the SRS.
6. **Deployment of system** - We make our application live in a production environment after all functional and nonfunctional testing completed.

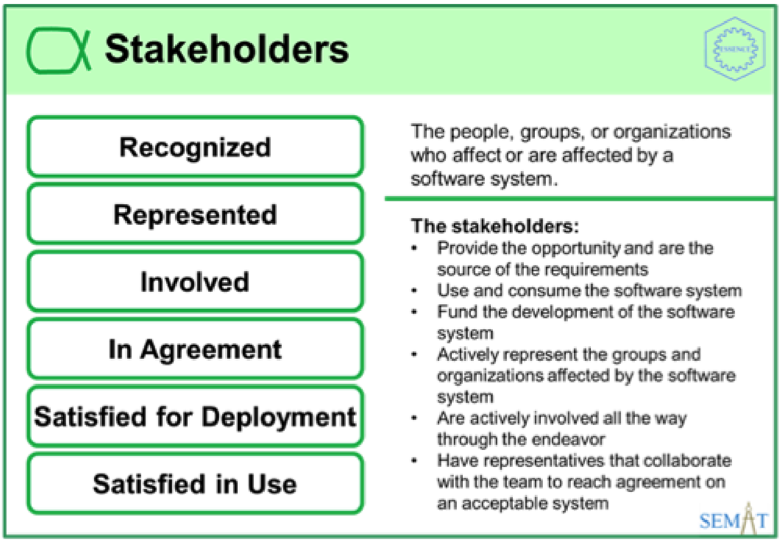
ESSENCE KERNEL:

1) The Alphas:



a)Customer:

i) Stakeholder:



Stakeholders

Recognized

Represented

Involved

In Agreement

Satisfied for Deployment

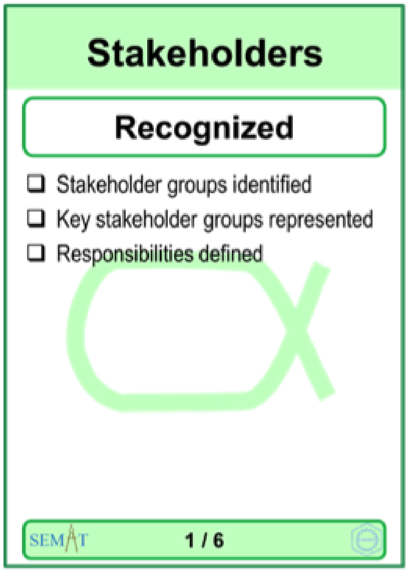
Satisfied in Use

The people, groups, or organizations who affect or are affected by a software system.

The stakeholders:

- Provide the opportunity and are the source of the requirements
- Use and consume the software system
- Fund the development of the software system
- Actively represent the groups and organizations affected by the software system
- Are actively involved all the way through the endeavor
- Have representatives that collaborate with the team to reach agreement on an acceptable system

SEMAT

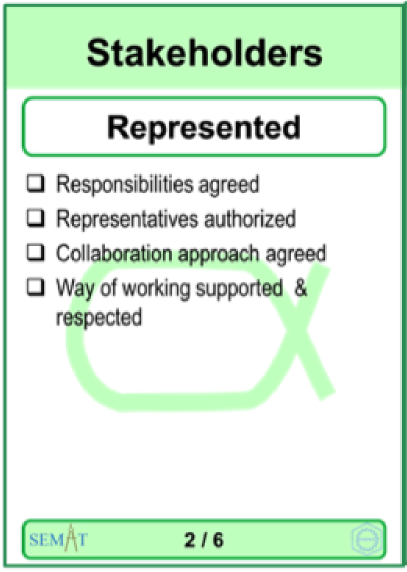


Stakeholders

Recognized

- □ Stakeholder groups identified
- □ Key stakeholder groups represented
- □ Responsibilities defined

SEMAT 1 / 6

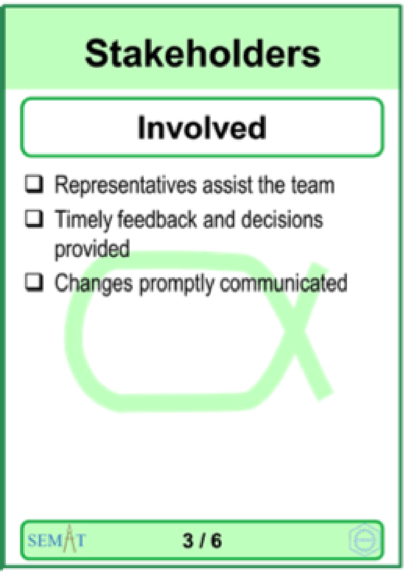


Stakeholders

Represented

- □ Responsibilities agreed
- □ Representatives authorized
- □ Collaboration approach agreed
- □ Way of working supported & respected

SEMAT 2 / 6

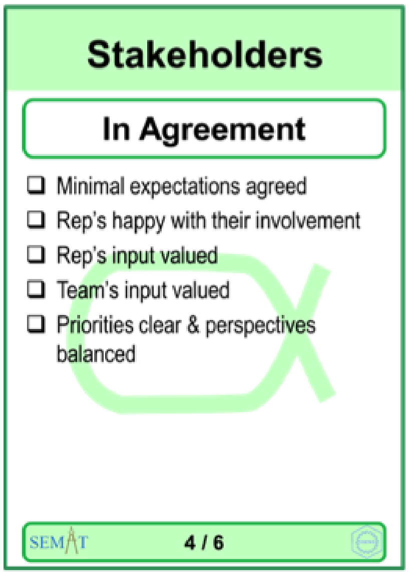


Stakeholders

Involved

- □ Representatives assist the team
- □ Timely feedback and decisions provided
- □ Changes promptly communicated

SEMAT 3 / 6

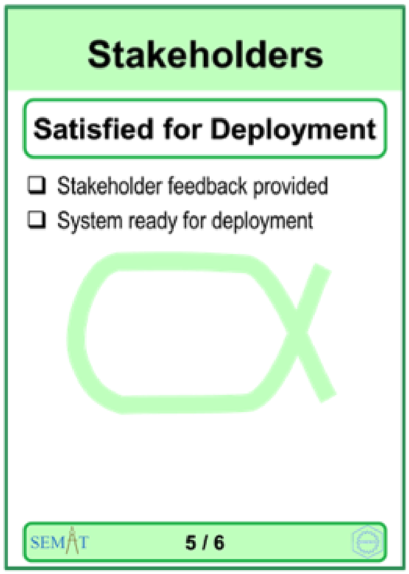


Stakeholders

In Agreement

- □ Minimal expectations agreed
- □ Rep's happy with their involvement
- □ Rep's input valued
- □ Team's input valued
- □ Priorities clear & perspectives balanced

SEMAT 4 / 6

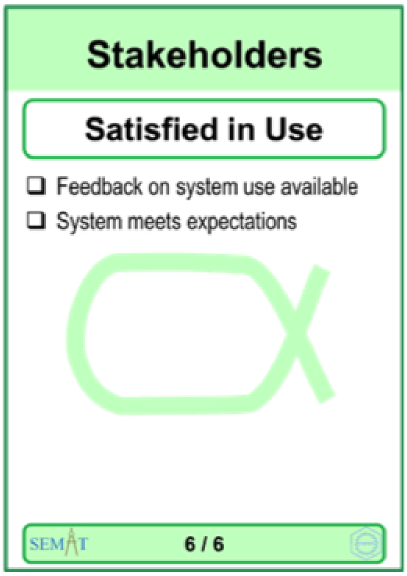


Stakeholders

Satisfied for Deployment

- □ Stakeholder feedback provided
- □ System ready for deployment

SEMAT 5 / 6



Stakeholders

Satisfied in Use

- □ Feedback on system use available
- □ System meets expectations

SEMAT 6 / 6

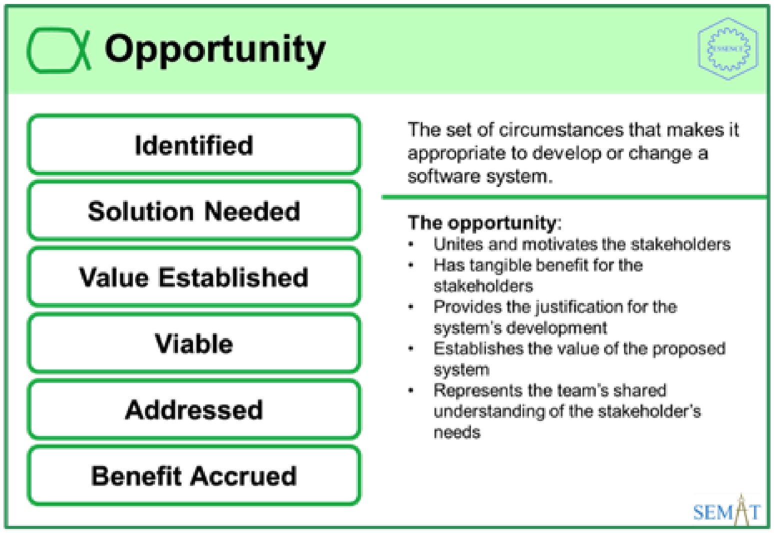
Recognized , Represented and involved: We had identified our target users in SRS only as it's a small project of 2 people there are no investors. We play the role of client as developers are our user and friend who are also developers.
We had decided our responsibilities while making a gantt chart (in SRS for ref).We are represented.we being developers act as customer as well.
All the states of representatives achieved while making a gantt chart.
We were involved and well communicated.

In agreement: Expectations were agreed at the time of SRS.Use cases helped us in thinking from customer perspective.It made us value all the target audience and priorities were clear.

5th and 6th state: All the functionalities are working well so stakeholder satisfaction is there.Feedback was taken before deployment (form target users including us).It is deployment and meets all expectations.

Satisfied in use state achieved.

ii) Opportunities:



Opportunity

Identified

Solution Needed

Value Established

Viable

Addressed

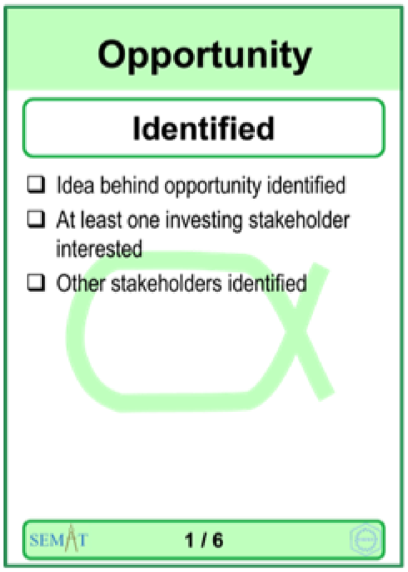
Benefit Accrued

The set of circumstances that makes it appropriate to develop or change a software system.

The opportunity:

- Unites and motivates the stakeholders
- Has tangible benefit for the stakeholders
- Provides the justification for the system's development
- Establishes the value of the proposed system
- Represents the team's shared understanding of the stakeholder's needs

SEMAT

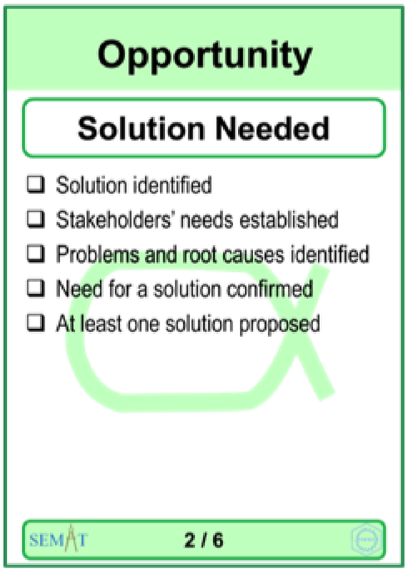


Opportunity

Identified

- ❑ Idea behind opportunity identified
- ❑ At least one investing stakeholder interested
- ❑ Other stakeholders identified

SEMAT 1 / 6

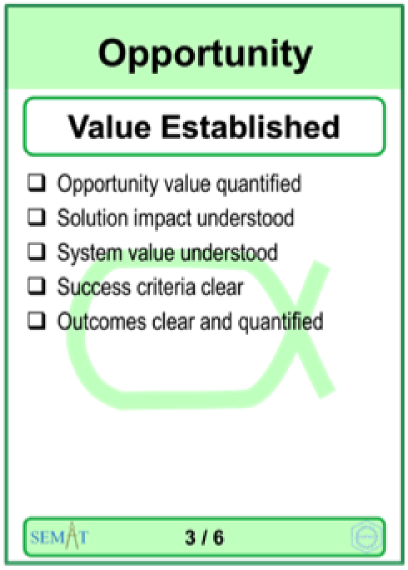


Opportunity

Solution Needed

- ❑ Solution identified
- ❑ Stakeholders' needs established
- ❑ Problems and root causes identified
- ❑ Need for a solution confirmed
- ❑ At least one solution proposed

SEMAT 2 / 6

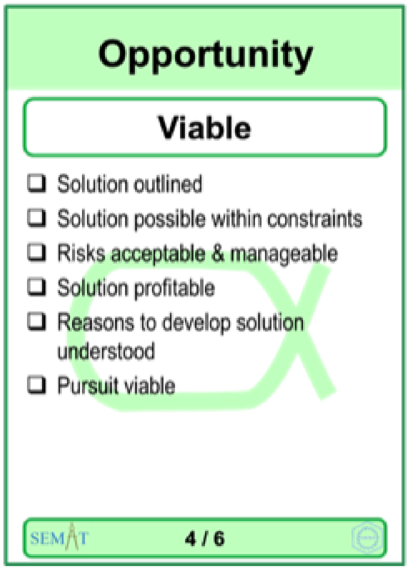


Opportunity

Value Established

- ❑ Opportunity value quantified
- ❑ Solution impact understood
- ❑ System value understood
- ❑ Success criteria clear
- ❑ Outcomes clear and quantified

SEMAT 3 / 6

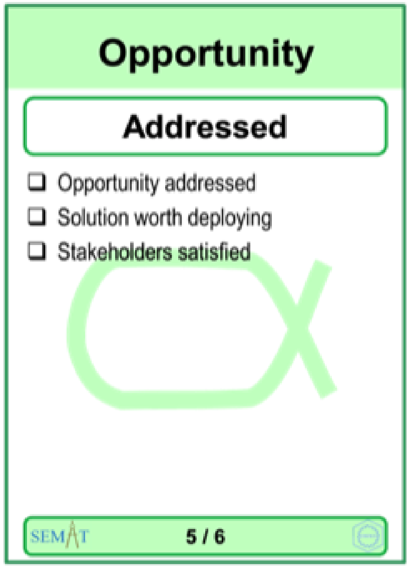


Opportunity

Viable

- ❑ Solution outlined
- ❑ Solution possible within constraints
- ❑ Risks acceptable & manageable
- ❑ Solution profitable
- ❑ Reasons to develop solution understood
- ❑ Pursuit viable

SEMAT 4 / 6

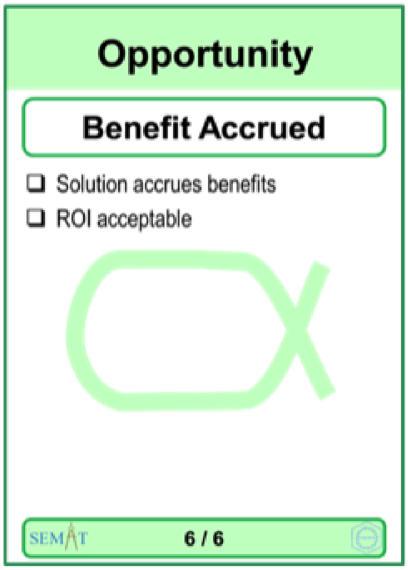


Opportunity

Addressed

- ❑ Opportunity addressed
- ❑ Solution worth deploying
- ❑ Stakeholders satisfied

SEMAT 5 / 6



Opportunity

Benefit Accrued

- ❑ Solution accrues benefits
- ❑ ROI acceptable

SEMAT 6 / 6

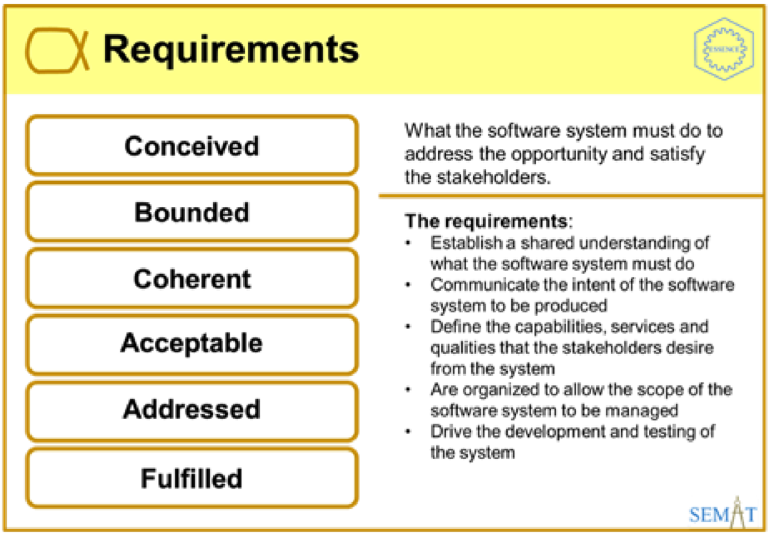
Identified & Solution needed: So first we had a problem statement and objective task where the whole idea was identified , solution needed was confirmed, stakeholders needs established through use cases and solution proposed through SRS and use cases .Stakeholders have already been given last state.

Value established: and viable All the checkpoints passed in making Of SRS and use cases activities.

Addressed: Our solution is deployed ,all promises fulfilled so stakeholders satisfied , hence opportunity addressed.

b)Solution:

i)Requirements



Requirements

Conceived

Bounded

Coherent

Acceptable

Addressed

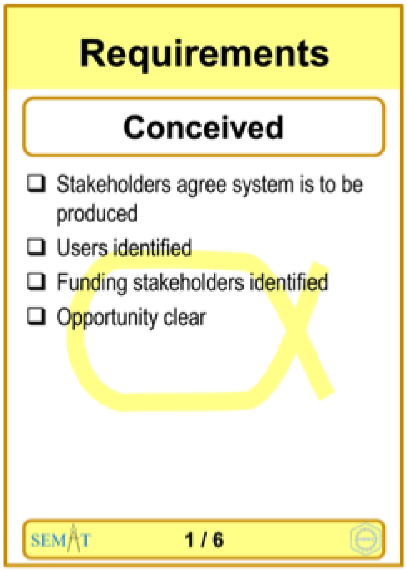
Fulfilled

What the software system must do to address the opportunity and satisfy the stakeholders.

The requirements:

- Establish a shared understanding of what the software system must do
- Communicate the intent of the software system to be produced
- Define the capabilities, services and qualities that the stakeholders desire from the system
- Are organized to allow the scope of the software system to be managed
- Drive the development and testing of the system

SEMAT

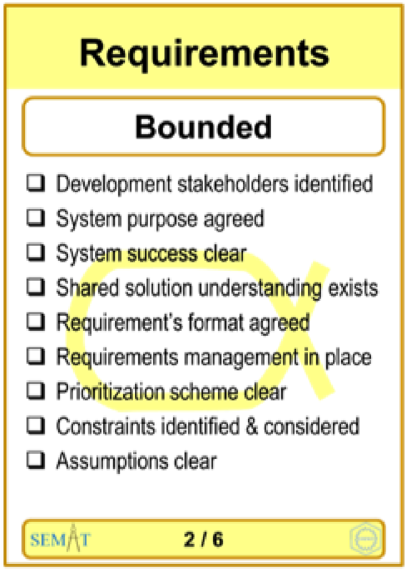


Requirements

Conceived

□ Stakeholders agree system is to be produced
□ Users identified
□ Funding stakeholders identified
□ Opportunity clear

SEMAT 1 / 6

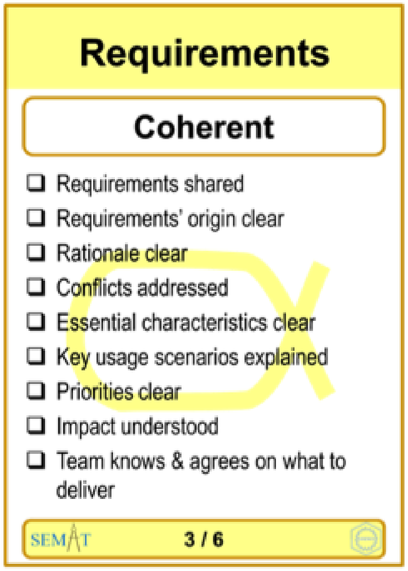


Requirements

Bounded

□ Development stakeholders identified
□ System purpose agreed
□ System success clear
□ Shared solution understanding exists
□ Requirement's format agreed
□ Requirements management in place
□ Prioritization scheme clear
□ Constraints identified & considered
□ Assumptions clear

SEMAT 2 / 6

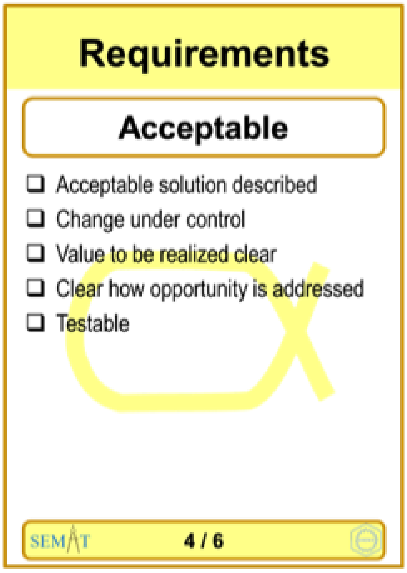


Requirements

Coherent

□ Requirements shared
□ Requirements' origin clear
□ Rationale clear
□ Conflicts addressed
□ Essential characteristics clear
□ Key usage scenarios explained
□ Priorities clear
□ Impact understood
□ Team knows & agrees on what to deliver

SEMAT 3 / 6

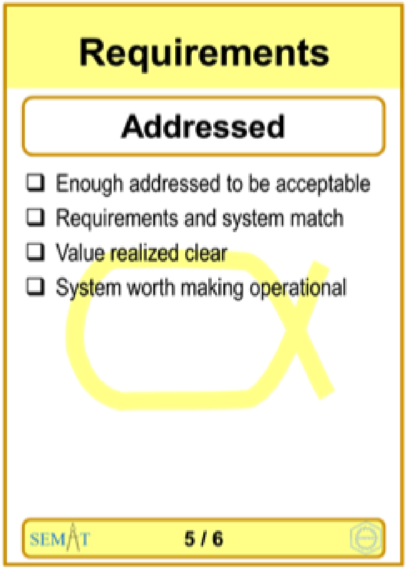


Requirements

Acceptable

□ Acceptable solution described
□ Change under control
□ Value to be realized clear
□ Clear how opportunity is addressed
□ Testable

SEMAT 4 / 6

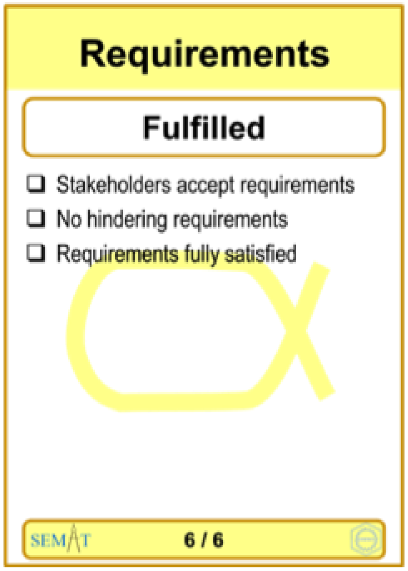


Requirements

Addressed

□ Enough addressed to be acceptable
□ Requirements and system match
□ Value realized clear
□ System worth making operational

SEMAT 5 / 6



Requirements

Fulfilled

□ Stakeholders accept requirements
□ No hindering requirements
□ Requirements fully satisfied

SEMAT 6 / 6

Conceived & Bounded & coherent: Opportunity passed identified stage . Stakeholders cleared all stages.[In the doc above]So conceived done.As SRS was done along with use cases so all remaining check points for bounded done. Coherent state checklist done while doing use cases and SRS and making gantt charts.

Use cases say that after doing the last activity requirement will be addressed or beyond.

Acceptable: Solution was described in use case , SRs and design doc.Test cases was also work product of use case.So done.

Addressed:You can see in the deployed version almost all requirement addressed also through testing..Req and system matching through testing using script and postman.System is deployed so it is worth operationable.

Fullfilled: As all req are fullfilled (through testing) . So its done

ii)Software System

Software System

- Architecture Selected
- Demonstrable
- Useable
- Ready
- Operational
- Retired

A system made up of software, hardware, and data that provides its primary value by the execution of the software.

The software system:

- Is the primary product of any software engineering endeavor
- Is structured, designed and implemented to fulfil the requirements
- Is architected to be maintainable, extensible and testable
- Provides value to its users and other stakeholders
- Should be kept bug free and easy to use
- Can be part of a larger software, hardware or business solution

SEMAT

Software System

Architecture Selected

❑ Architecture selection criteria agreed
❑ HW platforms identified
❑ Technologies selected
❑ System boundary known
❑ Decisions on system organization made
❑ Buy, build, reuse decisions made
❑ Key technical risks agreed to

SEMAT 1 / 6

Software System

Demonstrable

❑ Key architectural characteristics demonstrated
❑ System exercised & performance measured
❑ Critical HW configurations demonstrated
❑ Critical interfaces demonstrated
❑ Integration with environment demonstrated
❑ Architecture accepted as fit-for-purpose

SEMAT 2 / 6

Software System

Usable

❑ System can be operated
❑ System functionality tested
❑ System performance acceptable
❑ Defect levels acceptable
❑ System fully documented
❑ Release content known
❑ Added value clear

SEMAT 3 / 6

Software System

Ready

❑ User documentation available
❑ System accepted as fit-for-purpose
❑ Stakeholders want the system
❑ Operational support in place

SEMAT 4 / 6

Software System

Operational

❑ System available for use
❑ System live
❑ Agreed service levels supported

SEMAT 5 / 6

Software System

Retired

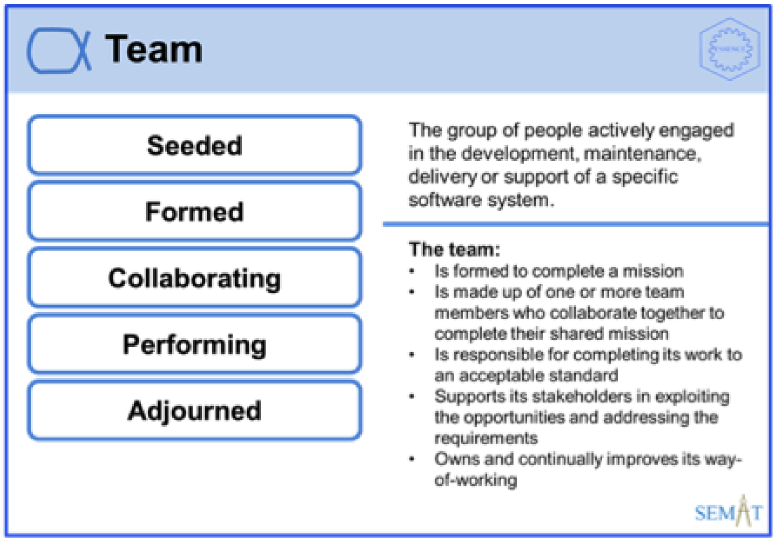
❑ Replaced or discontinued
❑ No longer supported
❑ No authorized users
❑ Updates stopped

SEMAT 6 / 6

Architecture Selected,Demonstrable and Usable: We have used MERN stack for creating this application. M stands for MongoDB which is a database, E stands for Express which is a modular web framework for Node.js, R stands for React which is used for reusable UI components and N stands for Node.js .An executable version of the system is available that demonstrates the architecture is fit for purpose and supports testing. Our application can be operated and all the functionalities have been tested. System performance is acceptable and our system is fully documented.This product will be usable by programmers from all ages along with one with poor eyesights.

Ready: The system (as a whole) has been accepted for deployment in a live environment.Installation and other user documentation are available.
Operational: User documentation has been prepared. The system has been made available to the stakeholders intended to use it.All the agreed services levels are supported.

iii.Team



Team

Seeded

Formed

Collaborating

Performing

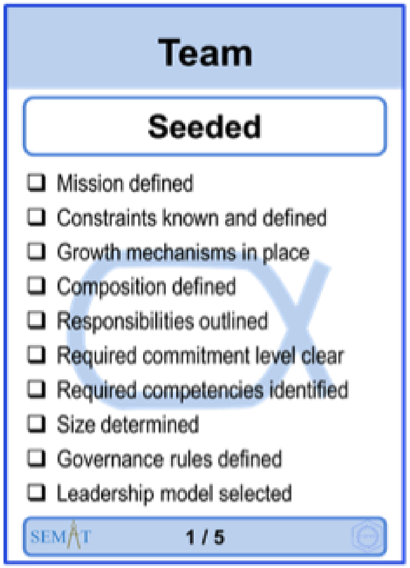
Adjourned

The group of people actively engaged in the development, maintenance, delivery or support of a specific software system.

The team:

- Is formed to complete a mission
- Is made up of one or more team members who collaborate together to complete their shared mission
- Is responsible for completing its work to an acceptable standard
- Supports its stakeholders in exploiting the opportunities and addressing the requirements
- Owns and continually improves its way-of-working

SEMAT

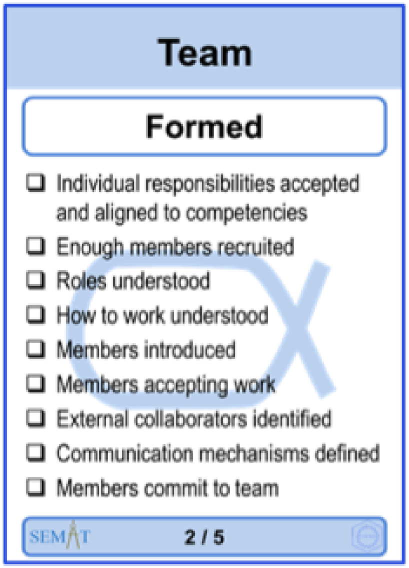


Team

Seeded

□ Mission defined
□ Constraints known and defined
□ Growth mechanisms in place
□ Composition defined
□ Responsibilities outlined
□ Required commitment level clear
□ Required competencies identified
□ Size determined
□ Governance rules defined
□ Leadership model selected

SEMAT 1 / 5

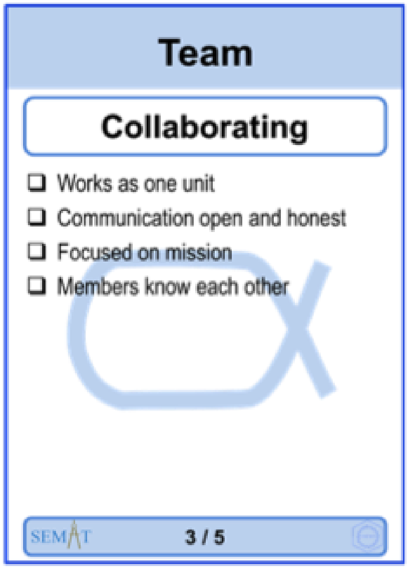


Team

Formed

□ Individual responsibilities accepted and aligned to competencies
□ Enough members recruited
□ Roles understood
□ How to work understood
□ Members introduced
□ Members accepting work
□ External collaborators identified
□ Communication mechanisms defined
□ Members commit to team

SEMAT 2 / 5

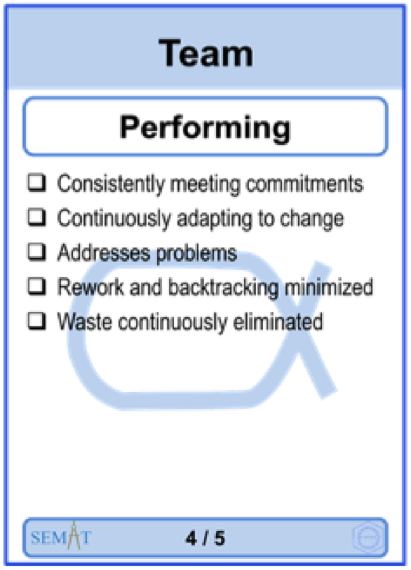


Team

Collaborating

□ Works as one unit
□ Communication open and honest
□ Focused on mission
□ Members know each other

SEMAT 3 / 5

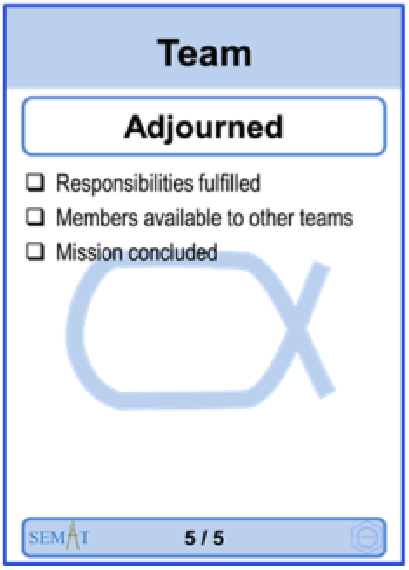


Team

Performing

□ Consistently meeting commitments
□ Continuously adapting to change
□ Addresses problems
□ Rework and backtracking minimized
□ Waste continuously eliminated

SEMAT 4 / 5



Team

Adjourned

□ Responsibilities fulfilled
□ Members available to other teams
□ Mission concluded

SEMAT 5 / 5

Seeded: The mission of the team is defined properly in terms of opportunities and outcomes.We have only 2 members in the team and the responsibilities of each team member is outlined properly.

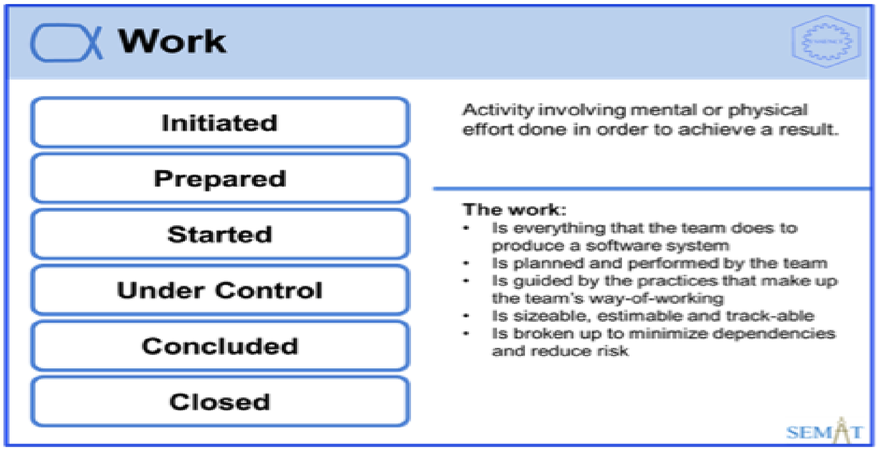
Formed: Every team member understands how the team is organized and what their individual role is.All team members understand how to perform their work.Each team member commits to working on the team as defined.

Collaborating: Both the team members have worked together with dedication and devotion. Both supported each other in every situation.Communication within the team members is open and honest.

Performing,Adjourned: We identified and addressed problems without outside help.Effective progress is being achieved with minimal avoidable backtracking and reworking.The team responsibilities have been fulfilled as the application is working properly without any flaws.

c)Endeavor:

i)Work



Work

Initiated

Prepared

Started

Under Control

Concluded

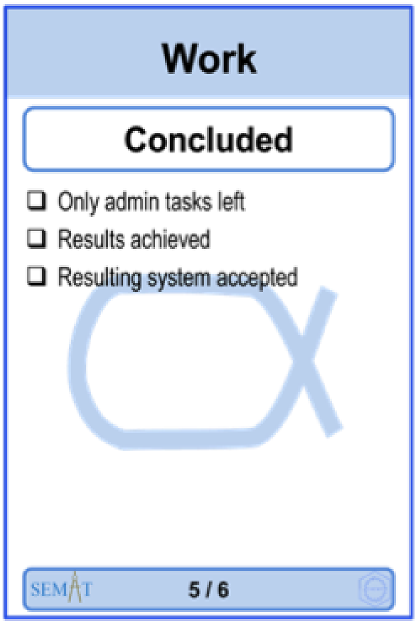
Closed

Activity involving mental or physical effort done in order to achieve a result.

The work:
- Is everything that the team does to produce a software system
- Is planned and performed by the team
- Is guided by the practices that make up the team's way-of-working
- Is sizeable, estimable and track-able
- Is broken up to minimize dependencies and reduce risk

SEMAT

The result required of the work being initiated is clear. The priority order of all the works is prepared properly and we completed it very well. Cost and Effort required to complete this project were estimated properly. The availability of resources were known to both of us.



Work

Concluded

- □ Only admin tasks left
- □ Results achieved
- □ Resulting system accepted

SEMAT 5 / 6

As all the requirements have been fulfilled according to the architecture and design document and SRS .System has been accepted by the stakeholders and team responsibilities are fulfilled (only admin tasks left).Hence work concluded.

iii)Way of Working:

Our team evolves our own way of working alongside with proper understanding of our mission and our working environment. As the work proceeds we continually reflect on our way of working and adapt it as necessary to our current context.

Deployed Link: New tab (powerful-plains-48555.herokuapp.com)