PROJECT-REPORT

Modern Software Development

IV1303

Abir Khoulani | 810420 | abirk@kth.se

Adam Chahoud | 980910 | achahoud@kth.se

Hiba Qutbuddin Habib | 990821 | hibaqh@kth.se

Hibatallah Belhajali | 951128 | hibbel@kth.se

Klara Lindemalm | 990930 | klindema@kth.se

Rasmuss Per Brodin | 980129 | rpbrodin@kth.se

CINTE

VT-2023

Assignment 1

format 1:

req 1

Description of the requirement:	A product name or serial-code should provide all information about a product, including name, product description, serial-number, stock and price.
Rationale:	This is necessary in order to complete a purchase at the register and help customers with product information.
Reference:	Not needed for this requirement
Source:	The customer

req 2

Description of the requirement:	The database should contain reasonable information about the transactions like transaction number, date and time, name of product, quantity and staff name or nr. And this database has to be modified automatically after every buy operation.
Rationale:	This function will be used by the company's accountant in order to be able to calculate all income and take care of the financial situation of the company
Reference:	Not needed for this requirement
Source:	The customer

req 3

Description of the requirement:	The program should support functionality of manually adding or removing items in the database. .
Rationale:	We want to have a seasonal stock where we add and remove products often.
Reference:	Not needed for this requirement
Source:	The customer

format 2:

req 1

As a [checkout staff] i want to [be able to quickly get information regarding price, name and serial-code by either writing the product name or scanning the serial-number so that I can sell the product or help customers with information about the product]

req 2

As a [company's accountant] I want to [be able to retrieve all information about the transactions in the database system] so I can [take care of the financial situation, review financial accounting and internal controls then write the audit report]

req 3

As a [logistics staff] i want to [be able to change the stock of items in the database in an administrative view and remove items that have expired or no longer available for sale] so i can [update the database manually and maintain relevant stock information about the assortment in store]

format 3:

req 1

User Case ID: 1
Use case Title: Sale info
Actors: System, customer, checkout staff
Flow of Events: Create the database and create queries that extract the required information and a program that uses this to display the information.
Alternative event:
Pre-condition: The product has to be available in the database.
Post-condition: Possibility to update the shopping cart and sum up the total amount for the purchase.
Assumptions:
Priority: 2

req 2

User Case ID: 2
Use case Title: transactions info
Actors:System, customer, floor staff , company's accountant
Flow of Events: Create the database and a view that extracts all information about requested products. Followed by a program that is easy to use that displays the information retrieved in the database in a readable fashion.
Flow of Events: Create the database which contains the transactions information and a view that extracts all information about a buy operation. Followed by a program that automatically adds transactions to the database after a new buy.

Alternative event:
Pre-condition: A buy operation has to be performed
Post-condition:
Assumptions:
Priority: 3

req 3
User Case ID: 3
Use case Title: update database
Actors:System, customer, logistics staff
Flow of Events:
Alternative event: Create the database and implement a program where the database can be easily manipulated.
Pre-condition: The person needs to be logged in to the system and also have administrative authority listed on their sales id.
Post-condition:
Assumptions:
Priority: 1

format 4:

req 1

- General Requirement Description:
    - Requirement ID: 01
    - Requirement title: Sale Information.
    - Requirement description: A product name or scanning a product should provide all information about a product, including name, product description, serial-number, stock and price.
    - Rationale: This is necessary in order to complete a purchase at the register.
    - Event/Use Case ID: 1
    - Intended user: The staff working in the store.
    - Specific user who stated the Request: The customer.
- Requirement Evaluation Data:
    - Business value: This requirement can reduce errors in transactions, such as incorrect prices. This can help improve customer satisfaction and reduce the risk of losses for the business.
    - Priority: 2
    - Acceptance criterion: Querying the database with either name or code will return a table with all information stated.
    - Fit criterion: Technical staff should be able to query the database correctly with 10 minutes of training.
- Other Description Data:

  - Preliminary Implementation Plan: Create a model of the database, create the database and create predefined queries that can be used.
- Requirements Reporting Data:
- Requirements Management Progress Data:
- Requirements Completion Data:
- Port Implementation Data:

req 2

- General Requirement Description:
  - Requirement ID: 2
  - Requirement title: Transactions
  - Requirement Description: The database should contain reasonable information about the transactions like transaction number, date and time, name of product, quantity and staff name or nr. And this database has to be modified automatically after every buy operation.
  - Internal/External Requirements: TAX rate is calculated rightfully given the transaction.
  - Intended User: The company accountant
  - Specific user who stated the Request: The customer
- Requirements Evaluation Data:
  - Business Value: Profitable
  - Priority: 3
  - Acceptance Criterion: Querying the database within a specific time period will retrieve information about the purchases performed within that period.
  - Fit Criterion: The customer should be able to retrieve transactional data from the database within a given time period in less than 20 minutes of training.
- Other Description Data:
- Requirements Reporting Data:
- Requirements Management:
  - Preliminary Implementation Plan: With a model of the database create a table of tractions and implement queries that retrieve information from a specific time period.
- Requirements Management Progress Data:
- Requirements Completion Data:
- Port Implementation Data:

req 3:

- General Requirement Description:
  - Requirement ID: 03
  - Requirement title: update database
  - Requirement description: The database should support functionality of manually adding or removing items.
  - Requirement type: Internal req
  - Rationale: We want to have a seasonal stock where we add and remove products often.
  - Event/Use Case ID: 3
  - Related to Requirement: 1,2
  - <u>Non-functional requirements:</u>

  - Intended User: logistics staff
  - Specific User who stated the req: The customer
  - <u>Customer Satisfaction:</u>
  - <u>Customer Dissatisfaction:</u>
  - <u>Reference document:</u>
- Requirements Evaluation Data:
  - Business Value:
    Accuracy
    Efficiency
    Improved decision-making
  - Other value:
    Flexibility
  - Priority: 1
  - Acceptance criterion: The database must provide a user-friendly interface that allows authorized users to manually add or remove items from the system.
  - fit criterion: Logistics staff should be able to add or remove product from the database correctly with 10 minutes of training.
- Other Description Data:
- Requirements Reporting Data:
- Requirements Management:
  - Preliminary Implementation Plan:Create a model of the database, create the database and create predefined queries that can be used.
- Requirements Management Progress Data:
- Requirements Completion Data:
- Port Implementation Data

Compare the different requirement formats with respect to the following:

- Detail of information

The requirement format differed in detail of information. Format 2 was shortest and most concise while format 4 was more ingoing and required a lot more information. As we are in the planning stages of the project we weren't able to meet all the required fields of the firth format. The first and third format required some information but not nearly as much as the forth.

- Level of coverage of the information communicating the requirement

A clear difference was seen between the availability of information in different formats. The first and second formats were quite easy to fill out, they did not cover a large part of the information, as it was more extraneous information in them. The third one was more comprehensive and covered a lot more, we needed to think deeper to answer it. Format four is completely comprehensive, so we could not answer some questions, because of lack of information, and we need to start with the project to answer.

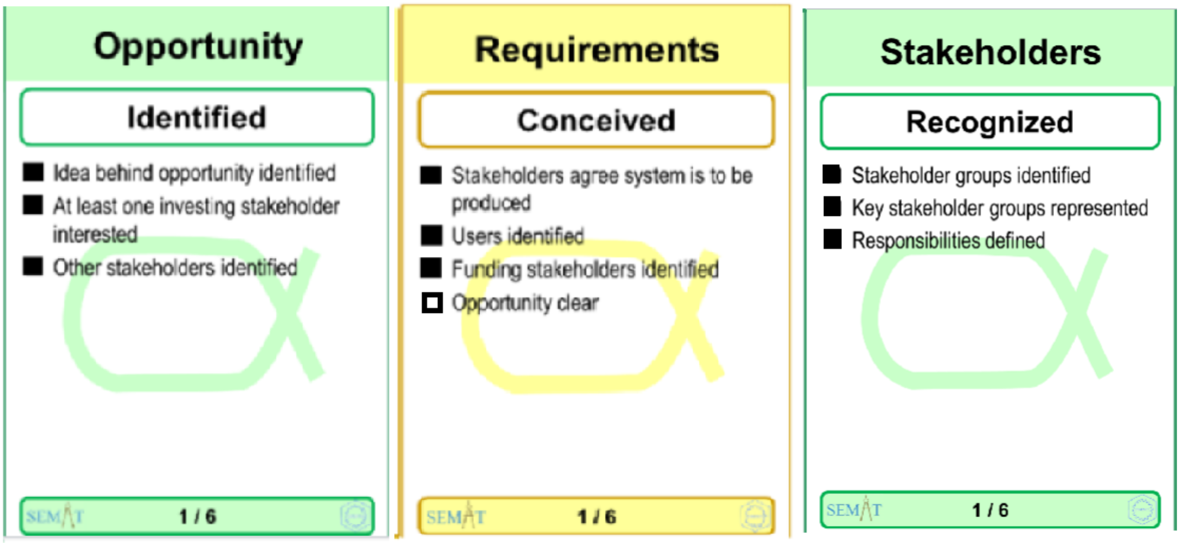
- Level of documentation effort

the documentation effort required in format 1 could be considered basic as it only gives us basic information, the second format seems like a written version of an oral requirement for which the documentation effort also seems low, format 3 on the other hand gives us enough information to start working on the project and effort of documentation is considerably higher than format 1&2, format 4 gives all the available and then some the effort here is very high.

- Difficulty to write the requirements

There was a clear difference in difficulty to write the requirements with the different formats. The easiest was format 2 which contained minimal information and it was clear what was necessary to correctly describe the requirement. The second easiest format was format 1 also with minimal excess information. Format 3 was more difficult as more excess information was required and some parts required more thought to write, for example assumptions. The most difficult format was format 4 that had very detailed information, many parts and required many aspects to be considered before it could be written. This format could also not be finished before the requirement has been implemented and tested making it more difficult as well.

Alpha Cards



Opportunity

Identified

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

SEMAT 1 / 6

Requirements

Conceived

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

SEMAT 1 / 6

Stakeholders

Recognized

■ Stakeholder groups identified
■ Key stakeholder groups represented
■ Responsibilities defined

SEMAT 1 / 6

- Opportunity state Identified
  This state is cleared as an idea has been identified and there is an investing stakeholder (the customer). We have identified other stakeholders to be companies using registers and wanting to easily check information regarding their products.
- Requirements state Conceived
  All items for this state are checked with the exception of “opportunity clear”. This could also be checked depending on the definition, we have an opportunity but the state is only identified, not yet made clear.
- Stakeholders state Recognized
  All checklist items have been checked for this state.

Result

The group successfully completed this assignment and learned valuable information about requirement formats and Alpha cards.

Assignment 2

Test cases (req1):

- **Test Case ID** 1.1
- **List of modules** Requirement 1
- **Input** name of product.
- **Output** Information containing, name, serial-code, price and stock.
- **Environmental needs** The cashier program
- **Inter - case dependencies** No dependency.
- **Special procedural need** No special procedural

Test cases (req1):

- **Test Case ID** 1.2
- **List of modules** Requirement 1
- **Input** serial-code of product.
- **Output** Information containing, name, serial-code, price and stock.
- **Environmental needs** The cashier program
- **Inter - case dependencies** No dependency.
- **Special procedural need** No special procedural

Test cases (req1):

- **Test Case ID** 1.3
- **List of modules** Requirement 1
- **Input** name of product/serial-code that does not exist.
- **Output** text informing the user that the product does not exist.
- **Environmental needs** The cashier program
- **Inter - case dependencies** No dependency.
- **Special procedural need** No special procedural

Test cases (req2):

- **Test Case ID** 2.1
- **List of modules** Requirement 2
- **Input** Time-interval. Date and time interval
- **Output** All transactions within the given time interval placed in the store
- **Environmental needs** The cashier program
- **Inter - case dependencies** No dependency.
- **Special procedural need** No special procedural

Test cases (req2):

- **Test Case ID** 2.2
- **List of modules** Requirement 2
- **Input** Run a transaction of one product.
- **Output** The database should be updated,
- **Environmental needs** The cashier program

- **Inter - case dependencies** No dependency.
- **Special procedural need** No special procedural

Test cases (req3):

- **Test Case ID**3.1
- **List of modules** Requirement 3
- **Input** Add a product with given serial number name price, ect
- **Output** Updated database with increased stock of the given item
- **Environmental needs** The cashier program
- **Inter - case dependencies**No dependency.
- **Special procedural need** No special procedural

Test cases (req3):

- **Test Case ID** 3.2
- **List of modules** Requirement 3
- **Input** Remove product with given serial number name price, ect
- **Output** Updated database with decreased Stock of the given item
- **Environmental needs** The cashier program
- **Inter - case dependencies** No dependency.
- **Special procedural need** No special procedural

Estimation time for the requirement:

Agile (Planning poker)

Requirement 1:
8 h for the group to fulfill the requirement of req 1.

Requirement 2:
10 h for the group to fulfill the requirement of req 2.

Requirement 3:
7 h for the group to fulfill the requirement of req 3.

Traditional

Req 1:

- Create database 70 lines of code
- Create program that interacts with the database, including:
    - Sub system that communicates with the database 100 lines of code

170 total lines of code 10 h for the group

Req 2:

- Create database 70 lines of code
- Create a program that interacts with the database, including:
    - Sub system that communicates and manipulates the database 110 lines of code

180 total lines of code 10.5 hours

Req 3:

- Create database 70 lines of code
- Create a program that interacts with the database, including:
    - Sub system that manipulates the database 110 lines of code

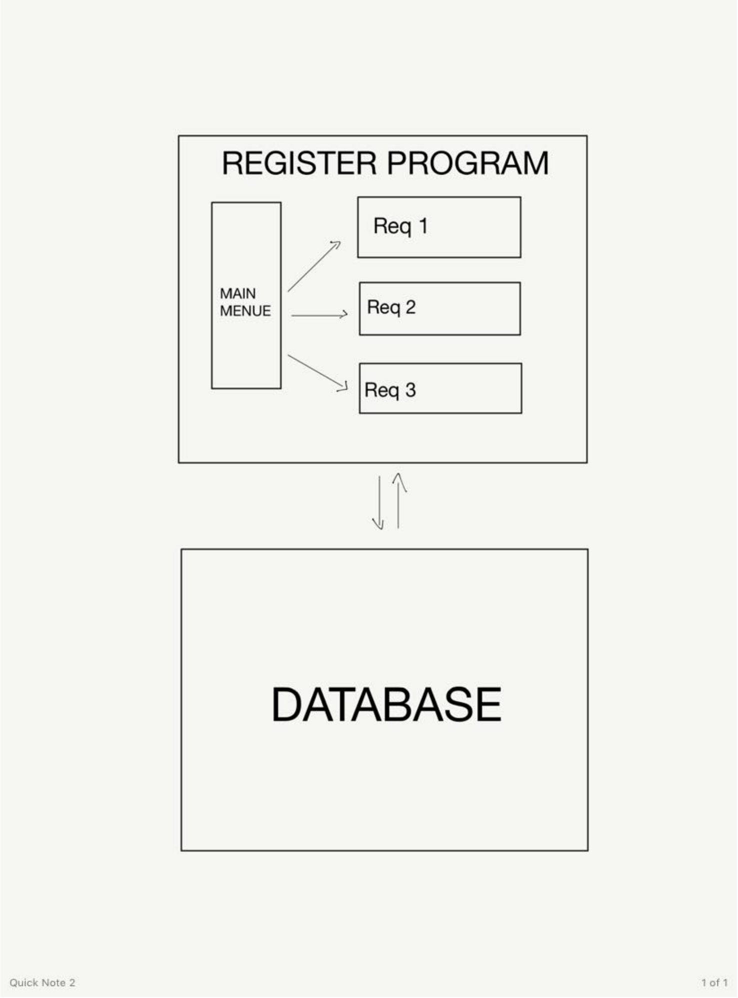
180 lines of code 10.5 hours

Compare the two approaches

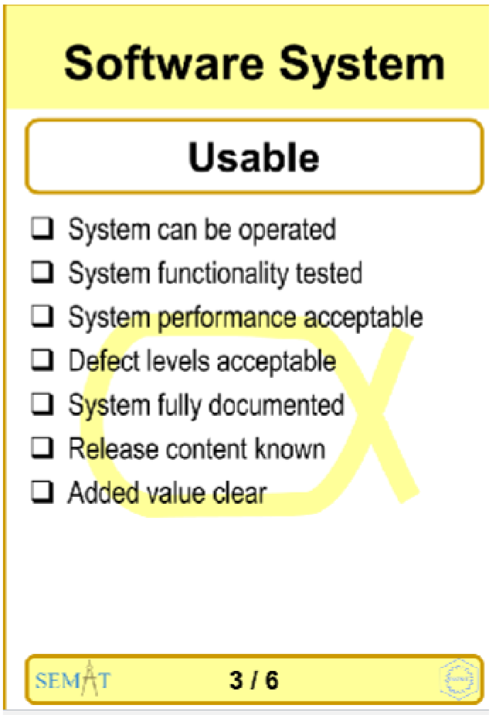
The group felt like the agile way of estimating the work time was more efficient and fun. If we had historical data to back up work time per LOC and previous knowledge of how to estimate LOC in a project the traditional way could get more accurate results. Since we did not have this it felt more like blind guesses rather than estimations.

Design the system

Quick design of how the system will work.



Status of Software System Alpha

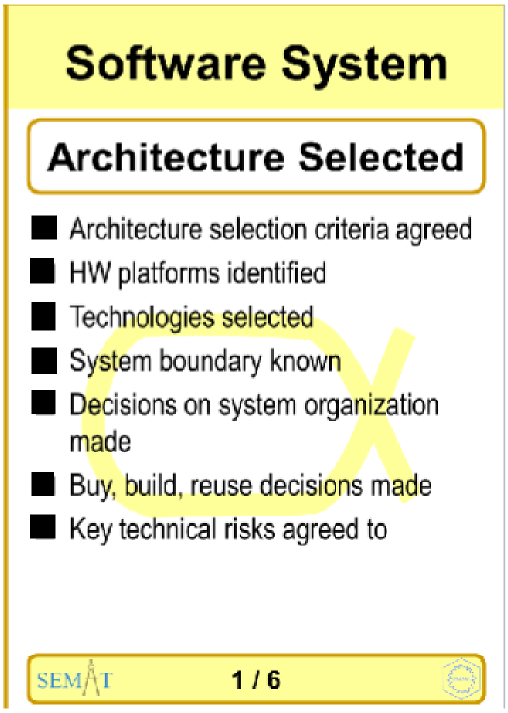


Software System

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

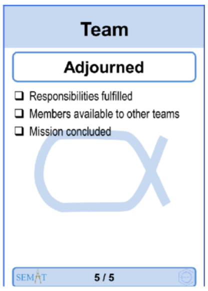
Architecture Selected

- ■ 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

The current state for the Software System Alpha is Architecture selected. Everything regarding the software has been decided but the implementation has not yet started so naturally the software system can not be in the Usable state yet.

Status of Team Alpha

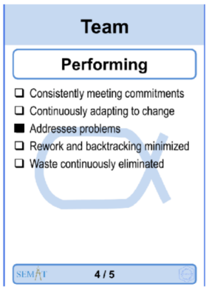


Team

Adjourned

- ☐ Responsibilities fulfilled
- ☐ Members available to other teams
- ☐ Mission concluded

SEMAT 5 / 5

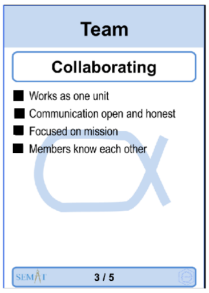


Team

Performing

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

SEMAT 4 / 5

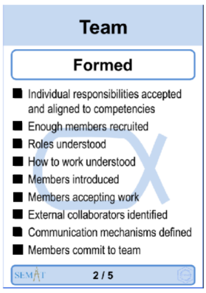


Team

Collaborating

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

SEMAT 3 / 5

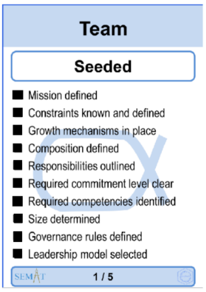


Team

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

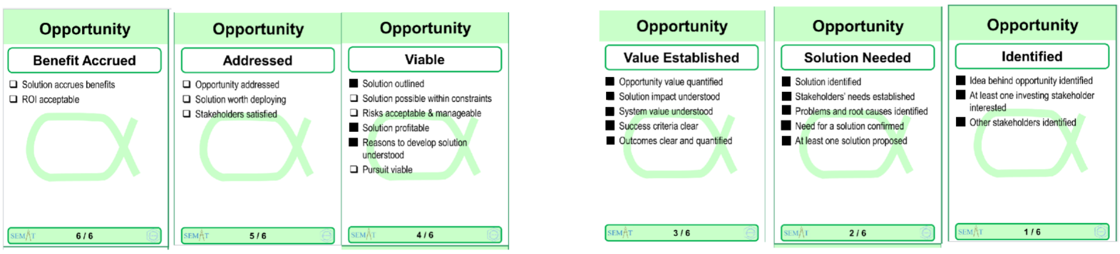
Seeded

- ■ 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

The current status for the team is collaborating. We have not yet started to write code and so we can not check all the requirements for the performing state.

progress of the status states



Opportunity
Benefit Accrued
☐ Solution accrues benefits
☐ ROI acceptable
SEMAT 6 / 6

Opportunity
Addressed
☐ Opportunity addressed
☐ Solution worth deploying
☐ Stakeholders satisfied
SEMAT 5 / 6

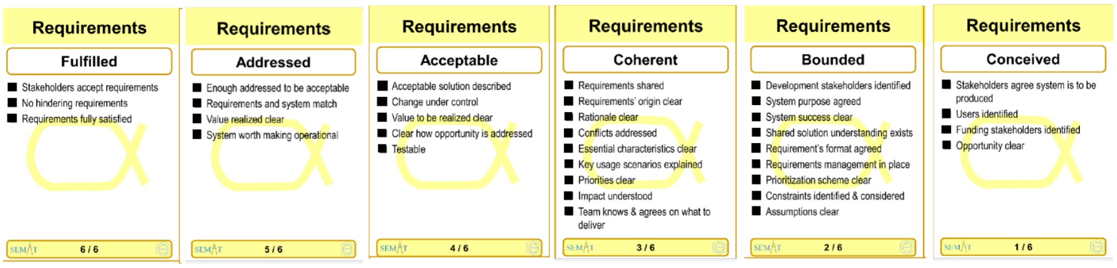
Opportunity
Viable
■ Solution outlined
☐ Solution possible within constraints
☐ Risks acceptable & manageable
■ Solution profitable
■ Reasons to develop solution understood
☐ Pursuit viable
SEMAT 4 / 6

Opportunity
Value Established
■ Opportunity value quantified
■ Solution impact understood
■ System value understood
■ Success criteria clear
■ Outcomes clear and quantified
SEMAT 3 / 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
Identified
■ Idea behind opportunity identified
■ At least one investing stakeholder interested
■ Other stakeholders identified
SEMAT 1 / 6

The opportunity alpha is currently in the Value Established state. The reason we cannot proceed to the Viable state is because we have become unsure if the system can be delivered within the time constraint.



Requirements
Fulfilled
■ Stakeholders accept requirements
■ No hindering requirements
■ Requirements fully satisfied
SEMAT 6 / 6

Requirements
Addressed
■ Enough addressed to be acceptable
■ Requirements and system match
■ Value realized clear
■ System worth making operational
SEMAT 5 / 6

Requirements
Acceptable
■ Acceptable solution described
■ Change under control
■ Value to be realized clear
■ Clear how opportunity is addressed
■ Testable
SEMAT 4 / 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
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
Conceived
■ Stakeholders agree system is to be produced
■ Users identified
■ Funding stakeholders identified
■ Opportunity clear
SEMAT 1 / 6

The requirements alpha is in the fulfilled state. We were only given three requirements from the customer so it was not difficult to plan the system so that they would be fulfilled and agreed by the customer.

Result

All alphas have progressed and the assignment was completed in a satisfactory manner.

Assignment 3

Evaluating the Scenario presented using Alfa cards. Presented with the group understanding of the text piece for each state followed by the evaluation of the group.

Opportunity Alpha
Understanding: With an understanding that all team members are in agreement that a possible solution is attained, however no risk evaluation has been accomplished so they do not feel confident to check the opportunity's viable state.
Evaluation: The provided reasoning is valid and we agree with Cynthia that a reasonable action for this is to check and assess the risks before moving forward to the viable state.

Stakeholders Alpha
Understanding: Sam differs in the planning poker game and the team members agree that the last check item isn't attained "The responsibilities of the stakeholder representatives have been defined!"
Evaluation: In our minds this is a reasonable argument and crucial to define before moving to the next state so that everything is clear from both parties perspectives and facilitates the project during the process with defined representatives from the stakeholder perspective.

Requirements Alpha
Understanding: Our understanding of the text was that they are in the bounded state as all criteria of the conceived state are attained.
Evaluation: The breakdown of this requirement was quite hard to follow and understand giving their reasoning provided in the text. However, breaking each piece of reasoning down, we believe that the action defined is reasonable as it is important to capture the legacy system requirements first before moving on with anything else regarding the requirement opportunity. Meaning truly understanding what the old system was in use for and what criterias are essential when developing the new system.

Software System Alpha
Understanding: The team is diverse in the state initially as both Sam and Fred raise the Architecture Selected card. However Cynthia argues that the last item on the card hasn't been checked, regarding the technical risks.
Evaluation: As a group we argue that this is a very wise statement that Cynthia has presented, with all software development processes one has to evaluate potential technical risks as they may come at any time during the development of the program. Without a thorough technical risk evaluation significant failures can cause delays and more.

Team Alpha
Understanding: The group in the text are in agreement that they now have to focus to obtain the formed state by setting up individual tasks and focus points for each person in the team.
Evaluation: As a group this evaluation was easy and felt reasonable from the start without any uncertainties. Once a team has a clear goal and good dynamic they can start dividing the project and setting individual focus points for each project member.

Work Alpha

Understanding: The group agrees that they have not achieved the "Initiated state" yet, as the required results are still unknown and the prioritization of work is also unclear. The group has flexibility in planning work and therefore as a response the group must "capture the minimum essential capabilities for the new system".

Evaluation: We only agree with this statement, we are missing a statement from Cynthia but assuming she only agrees we believe this is a valid reasoning they are presenting.

Way of Working Alpha

Understanding: The group agrees that they have not achieved the "Principles Established state" as the stakeholders' representatives are unknown. This leads to the group lacking agreement from stakeholders for the scrum work method and the work management tool. To achieve this state, the group needs to "Get stakeholders assigned and get their agreement on our way of working and tool selection".

Evaluation: When working with a project out evaluation it is vise to maintain close relationship with the stakeholders and get their agreement on the way of working.

Impression of playing a game

- *Good*
    - Everyone gets a change to state their opinion and provides a structured way of working with the project.
    - Good tool to use to engage everyone in the project.
    - Maintains a coherent project where everyone is included.
- *Bad*
    - Without a good understanding of each of the action items the tool can be quite hard to use.
    - When everyone is not involved the game loses its functionality and important tasks can be missed.

Impression of gasping the whole domain of software engineering with the alphas- seven essential things:

- It is a very complex instrument, but when truly using it and understanding each of the seven alphas it provides a useful help tool to maintain a great project structure and list action items as the tool is used throughout the process.
- Opens up to ideas that would not necessarily have come up without the checkpoints of the action cards.

Assignment 4

We choose to implement requirement 1.

Risk identification
The group did not have access to the database modeling software that we would have thought. This will make the implementation of the database more difficult and time consuming. Time was already a risk the group faced as we estimated a collective time of 10 hours to fully implement this requirement.

Pair programming

- good sides
  This way of programming gives a better code quality and reduces the risk of errors, then even if you get an error it is faster to detect it and correct the error by two developers.

  We felt that this method of working helped with knowledge sharing and utilized our skills in an efficient way. It also improved the group morale and encouraged discussion and getting to know the team.

- bad sides
  This way of programming is time consuming as two developers are working on the same code, but the working time is of course not cut in half.

Test first programming

- good sides:
  The positive with test first programming is that it makes the generating of the code more simple by specifying what exactly should the code do.
  Also this process increases the developers confidence in their own code, because they can see that the test actually works and that the code meets the requirement.

- bad sides:
  We felt that this process is time consuming because we have to write the test first before we start to implement the code and this may take longer time.

Refactoring:

- good sides:
  Refactoring makes the structure of the code more organized and it improves readability by eliminate unnecessary parts of the code which improves the code quality.

- bad sides:
  Med refactoring it is a risk of changing the code in a way that makes it harder for other developers to understand it, if we haven’t documented and communicated the changes clearly.

Designing the code first before coding:

- good sides:
  This makes the implementation of the code more simple and organized because we have a clear idea of how the code should look.
  Better planning which helps us to avoid unnecessary time by having a clear idea of what needs

to be done and how long it will take.

- bad sides:
  By designing the code before starting can limit the freedom of developer because he has an initial design he has to respect, and it can be difficult to change the design later after writing a large part of code.

Your ability to estimate the effort required for implementing the requirement

Traditional way	LOC Estimate	LOC Actual	Time Estimate	Time Actual
Create database	70	25	4 hours	1 hours
Create program that uses the Database	100	64	6 hours	3 hours
Total	170	89	10 hours	4 hours

- **Agile way**
  We estimated a total time of 8 hours but completed the requirement in 4 hours.

Code to communicate with the database according to requirement 1

```
import java.sql.*;
import java.util.Scanner;

public class ProductSearcher {
  public static void main(String[] args) {
    // Database credentials
    String url = "jdbc:mysql://localhost:3306/IV1303";
    String user = "user";
    String password = "password";

    // Get user input
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter search term: ");
    String searchTerm = scanner.nextLine();

    // SQL query
    String query = null;
    if (searchTerm.startsWith("100")){
        query = "SELECT * FROM product WHERE product_code =
searchTerm";
```

```
    }
    else{
        query = "SELECT * FROM product WHERE product_name =
searchTerm";
    }

    try {
      // Connect to the database
      Connection conn = DriverManager.getConnection(url, user,
password);

      // Execute the query
      Statement stmt = conn.createStatement();
      ResultSet rs = stmt.executeQuery(query);

      // Check if any results were found
      boolean found = false;
      while (rs.next()) {
        found = true;
        String serialNumber = rs.getString("serial_number");
        String productName = rs.getString("product_name");
        String productDesc = rs.getString("product_description");
        int stock = rs.getInt("stock");
        int price = rs.getInt("price");

        System.out.println("Serial Number: " + serialNumber);
        System.out.println("Product Name: " + productName);
        System.out.println("Product Description: " + productDesc);
        System.out.println("Stock: " + stock);
        System.out.println("Price: " + price);
        System.out.println("------------------------------");
      }

      // Handle case where no products were found
      if (!found) {
        System.out.println("No products found with search term: " +
searchTerm);
      }

      // Close the connection
      conn.close();
    } catch (SQLException e) {
      System.out.println("Connection failed: " + e.getMessage());
    }
  }
}
```

Code to create the database according to requirement 1

```
CREATE TABLE product (
 serial_number VARCHAR(500) NOT NULL,
 product_name VARCHAR(500) NOT NULL,
 product_description VARCHAR(1000),
 stock INT NOT NULL,
 price INT NOT NULL
);

ALTER TABLE product ADD CONSTRAINT PK_product PRIMARY KEY
(serial_number);

CREATE TABLE product_list (
 product_list_id INT GENERATED ALWAYS AS IDENTITY NOT NULL,
 transaction_id INT NOT NULL,
 serial_number VARCHAR(500) NOT NULL,
 price INT NOT NULL,
 CONSTRAINT FK_product_list_transaction FOREIGN KEY (transaction_id)
REFERENCES transactions (transaction_id),
 CONSTRAINT FK_product_list_product FOREIGN KEY (serial_number)
REFERENCES product (serial_number),
 CONSTRAINT FK_product_list_product FOREIGN KEY (price) REFERENCES
product (price)
);

CREATE TABLE transactions (
 transaction_id INT GENERATED ALWAYS AS IDENTITY NOT NULL,
 transaction_date TIMESTAMP(10),
 CONSTRAINT PK_transactions PRIMARY KEY (transaction_id)
);
```

Assignment 5

Solving Pain Points with Requirements Alpha: Scenario 2

Cecile Peraire, Mira Kajko-Mattsson, Barry Myburgh, Maria Augusta Vieira Nelson, Paul E. McMahon

A five-member team has been in charge of developing an online university course management system and has produced the first release. The team has just received the green light from university management to proceed with Release 2.

The functionality in Release 2 would deal with the management of: (1) administrative information, (2) courses, and (3) student performance. It would strongly benefit administrators, faculty members and students, by facilitating their work and communication. It would also benefit university management by decreasing the overall administrative and managerial cost. The management is expecting to see the new system adopted by all at the end of the following academic year.

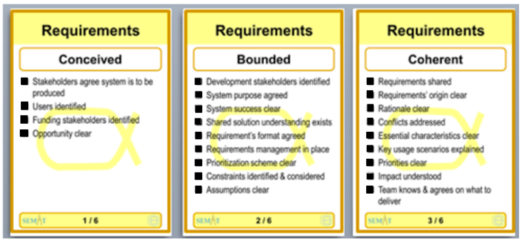
To further understand the university operation, the team members made inquiries about shortcomings of Release 1. They held frequent meetings with users to identify needs for Release 2, and also observed the usage of the new system. This gave them a good understanding of what worked well and what did not. Results of their efforts were analyzed and the improved usage scenarios were derived and documented at a fairly high level. The team's repository was updated to reflect the changes. Users were kept in the loop to validate the scenarios and to identify their relative importance. UI mockups were then created and/or improved for the most important scenarios. It was agreed that details would still have to be elaborated just before implementation.

In addition to the system users, the team also contacted other stakeholders to take their needs into account. For instance, given the university's current growth projection, the team and university management agreed to assume that the system should accommodate up to 5 000 users. They also agreed that any decisions that had constrained the development of Release 1 would also apply to Release 2.

Dealing with different stakeholder groups turned out to be challenging, as they often had different ideas on how things should be handled. For instance, the way the grades were communicated to the students caused disagreement. Faculty preferred the system to notify students about their grades by emails. Management, on the other hand, preferred the students to log in to access their grades. A short discussion helped solve the problem. Sending grades by emails was against the university's policy and the management solution would have to be implemented.

At some point during Release 2, one team member mentioned that a few faculty members were resisting the migration to the new system. They were still managing communication via emails and course assignments and grades via spreadsheets. They had no intention of migrating to the new system in the future. The team decided to interview a few of these faculty members to find out what the problem was. They also organized a short presentation of the functionality to be implemented in Release 2. The new functionality included (1) the management of student deliverables by the faculty members, (2) the ability to grade and provide feedback to the students, (3) the ability for students to view their grades in the system, and (4) the management of course materials by the faculty members. The goal was also to articulate the value of the new system over the value of the wikibased solution.

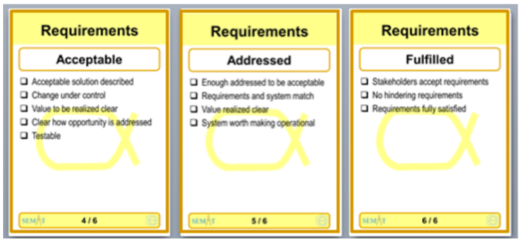
Through the interviews and demonstration, the team realized that the missing features requested by the faculty members were related to the way the new solution computes grades and manages feedback on deliverables. The new system is more restrictive. It does not allow faculty members to associate grading components to each student deliverable and it only supports grades based on points, not on alphabetic symbols.



Requirements
Conceived
■ Stakeholders agree system is to be produced
■ Users identified
■ Funding stakeholders identified
■ Opportunity clear

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

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



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

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

Requirements
Fulfilled
❑ Stakeholders accept requirements
❑ No hindering requirements
❑ Requirements fully satisfied

Summary: The scenario described above is in the coherent state as all the boxes are checked up to this point however the implemented solution is not an acceptable solution therefore the accepted state is not met. The system needs to fulfill the missing features before the solution can be accepted.

Solving Pain Points with Team Alpha: Scenario 2

Maria Augusta Vieira Nelson, Mira Kajko-Mattsson, Barry Myburgh, Cecile Peraire, Paul E. McMahon

A five-member team has been in charge of developing an online university course management system since its first release. The team was formed by the IT director, who carefully chose its members with the purpose of maximizing team productivity. Its composition was based on an optimal mix of personalities with competencies identified as crucial. The team consists of a project manager and four developers, each having the expertise and responsibility in their specific areas such as design, user experience, requirements and database.

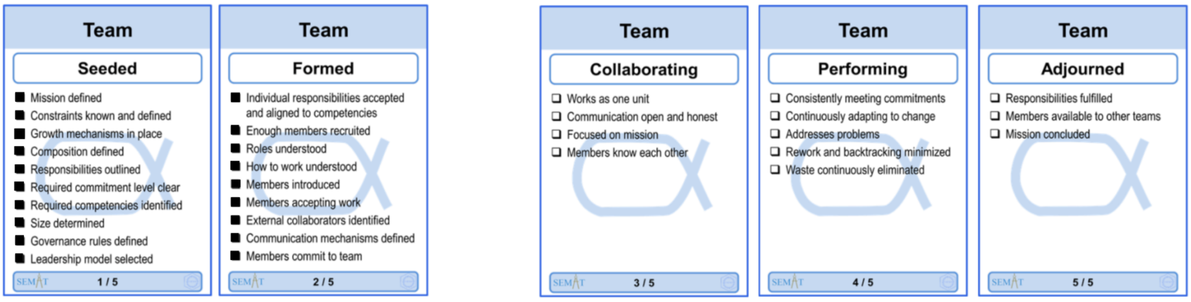
The team feels that its size and composition is satisfactory. The members are confident that they have the required competencies to fulfill their responsibilities. They know that as the system grows in the future, the team might have to be expanded.

The team has started working on the second release. It is very well acquainted with the project's initial needs. It has collectively established its goals, mission and responsibilities. The team members agreed to mainly communicate orally and to document only the most important issues such as requirements, design, problems, test cases and important decisions and events. The team practices a democratic leadership style implying that discussions with more than one possible outcome are discussed to make sure that everyone on the team has a chance to impact the decisions.

Team members have worked well and are committed to the project. Communication is sometimes challenging but each member knows how to conduct his/her own work and is dedicated to doing it. This is how the team succeeded in delivering Release 1.

So far, the stakeholder groups that have been identified are Administrators, Faculty, and Students. Each group has a few representatives who are willing to collaborate with the project team. For Release 2, the team has decided to meet and interview faculty members by visiting them at the university. This would help the team to identify needs for Release 2 and observe the usage of the new system.

After the interviews, two developers were in disagreement about one significant requirement. They shared the conflicting viewpoints with the faculty involved. The faculty agreed with the first developer, and the second developer felt somewhat put out that his opinion did not seem to matter. Since this was not the first time that the second developer's ideas had not been accepted, little by little he stopped communicating with the team.



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

Summary: The scenario above is in the formed state but has not achieved the collaborating state as one of the team players has stopped collaborating and communicating in the project. This due to the arised conflict between the two developers. This is of course very unfortunate and needs to be prevented beforehand as it was further mentioned in the text that this wasn't the first time the developer didn't feel appreciated for his efforts.