Group: xchen22@student.sdu.dk, dily22@student.sdu.dk, sebaj17@student.sdu.dk

Heading:

[CX: Chen's Comment]
[DK: Khai's Comment]
[SJ: Sebastian's Comment]

1 Section 1 : Name

SWE topic 1

A description and classification of the different stakeholders

Primary stakeholders

Core stakeholders - *These stakeholders have high power, high urgency, and high legitimacy.*

- **Product owner** is responsible for managing all the stakeholders (he represents the interests of everyone with a stake in the project) and he is also responsible for the long term return on investment (ROI). Should by classified as a core stakeholder, which means he has big authority and influence on the project same as legitimacy and urgency.
- **Development team** are the one who deliver the final product/software. The developers are at the heart of the entire project as they help give it a vision and pave the way for the project to be done and become successful.
- **Client** is individual or organisation that purchase the product/software.

Dependent stakeholders - *high legitimacy and urgency*

- **Customers** are people who will use the product/software to purchase an items. They decide whether the product will success or not. Customers have an interest in quality item products, service and low price. That's why they should be classified as dependent stakeholders.
- **Legal team** protects our company/development team from lawsuit troubles. Client might request some features that are against the law, than that's legal team job to discuss it with client to find a solution.

Dominant stakeholders - *high power and legitimacy*

- **Investors** are significantly impacted by the associated concern and its performance.

Discretionary stakeholders - *high legitimacy, low power, and low urgency*

- **Domain provider** will provide us domain on which our product/software will run.

Secondary stakeholders

- **Suppliers** provides products to the client.
- **Customer support** helps customer in case of need, for example: refund a purchase; correct invalid personal/address data during purchase, ...
- **Competitors.** The only competitors that we had were other software companies, but after client made a contract with us, we don't need to worry about them anymore. The reason is because client specifically came to us with his demands for a product. The client's competitors does not affect our product/software development.

]

A description and classification of the different users (End Users, Super Users, and System User)

Who will be the user base? What is in it for them?

We categorised the users in the userbase into 3 main groups, namely the **End Users, Super Users** as well as the **System Users**. End users are those users who use the system only, super users refer to those users who are often the administrator or have additional rights/ features and system users are those 3rd party services that utilise the system for their functions.

In this project, we have the following in our user base:

- **End Users:** Customers (Those buying pots and pans)
  - Household Families
  - Restaurant Owners
- **Super Users:**
  - Development Team (Ensure the system is working fine and desirable for its client)
    - Developers (Update and maintain the system)
    - Product Owner
    - Non Developer Team (E.g. Business engagement team)
  - Client Team (Using the system to sell the pots and pans)
    - Logistic Team (Keep track of inventories)
    - Staff (Ordinary Admin)
    - Managers (Ordinary Admin + Administrator Rights)
    - Account Manager (Access and maintain the accounts)
    - IT Department (Daily maintenance and troubleshooting for the system)
- **System Users:** (Utilising the system by making system calls to achieve their functions)
  - Payment Gateway
  - Email Server (For Support + Communication + Account)
  - Cloud Server (Storing of data)

Who will maintain the platform on a daily basis and offer support to end users?

Staff of client IT Department to log in and ensure system is working before shift each day.
Staff and account managers should offer support to end users
Client team to liaise with Development team should they need any assistance

Which internal and third party systems will interact with our system?

- Payment Gateway
- Time System
- Email Server
- Security System
- Cloud Server

A prioritized backlog containing all found requirements, both functional and non-functional in a format of your choice

The main product vision is We want to build a system where our client can keep track of their inventory. What is in stock, what needs to be ordered from our suppliers, and a initial version of a shopping basket.

To implement the product backlog we use the method of DEEP, which is described briefly below:

- (D)etails: Backlog item gets closer to being completed or moved into sprint backlog $\Longrightarrow$ item require more detail.
  Backlog items that are low on the priority list
  $\Longrightarrow$ require less detailed explanation.
- (E)stimated: Items high on the requiry list
  $\Longrightarrow$ requires thorough estimation.
  Items that are low on the requiry list
  $\Longrightarrow$ require less thorough estimation.
- (E)mergent: Revise and improve backlog the more information you get from stakeholders and the more knowledge you get about the product. Your backlog is not set in stone: think agile.
- (P)rioritize: When prioritizing keep in mind the value that each item on the backlog will provide.

Product Backlog

The functional requirements defines what a product must do, what its features and functions are. Based on that we present the functional requirements of our product backlog as follows:

1. We need a function that list content of the shopping basket,
2. We need a function which lists the orders in the system,

3. We need a function that creates an order,
4. We need a function that creates users and their orders,
5. API that connects supplier A and B.

[CX: The functional requirements defines what a product must do, what its features and functions are. Based on that we present the functional requirements of our product backlog as follows:

1. The system must be able to list the content of the shopping basket,
2. The system must be able to show the orders in the system,
3. The system must be able to create an order,
4. The system must be able to display login screen
5. User with a valid account must be able to log in with their username and password
6. The system must have an API that can be used by both supplier AB
7. The system must be able to display checkout page
8. The system must be able to update the inventory whenever user checks out
9. The system must be able to calculate the final amount at check out page

]

Product backlog for non-functional requirements.
The nonfunctional requirements are not directly related to the functionality of the system, rather it should define how the system is supposed to perform.
[SJ: Question: The manager wants some kind of discount functionality, but Carsten is not so sure about that - how should we include that and is that supposed to be a part of non-function or functional requirements? It is kind of at the intersection of those two?] Nonfunctional requirements:

1. Client must be able to keep track of their inventory on the website.
2. Client must be able to see what is in stock and what needs to be ordered from our suppliers.
3. Client wants to be able to see an initial version of a shopping basket
4. Client wants an online service (webshop) with corresponding (mobile friendly) website and at some point a dedicated app

]

Sprint Backlog(s) - with dates and comments

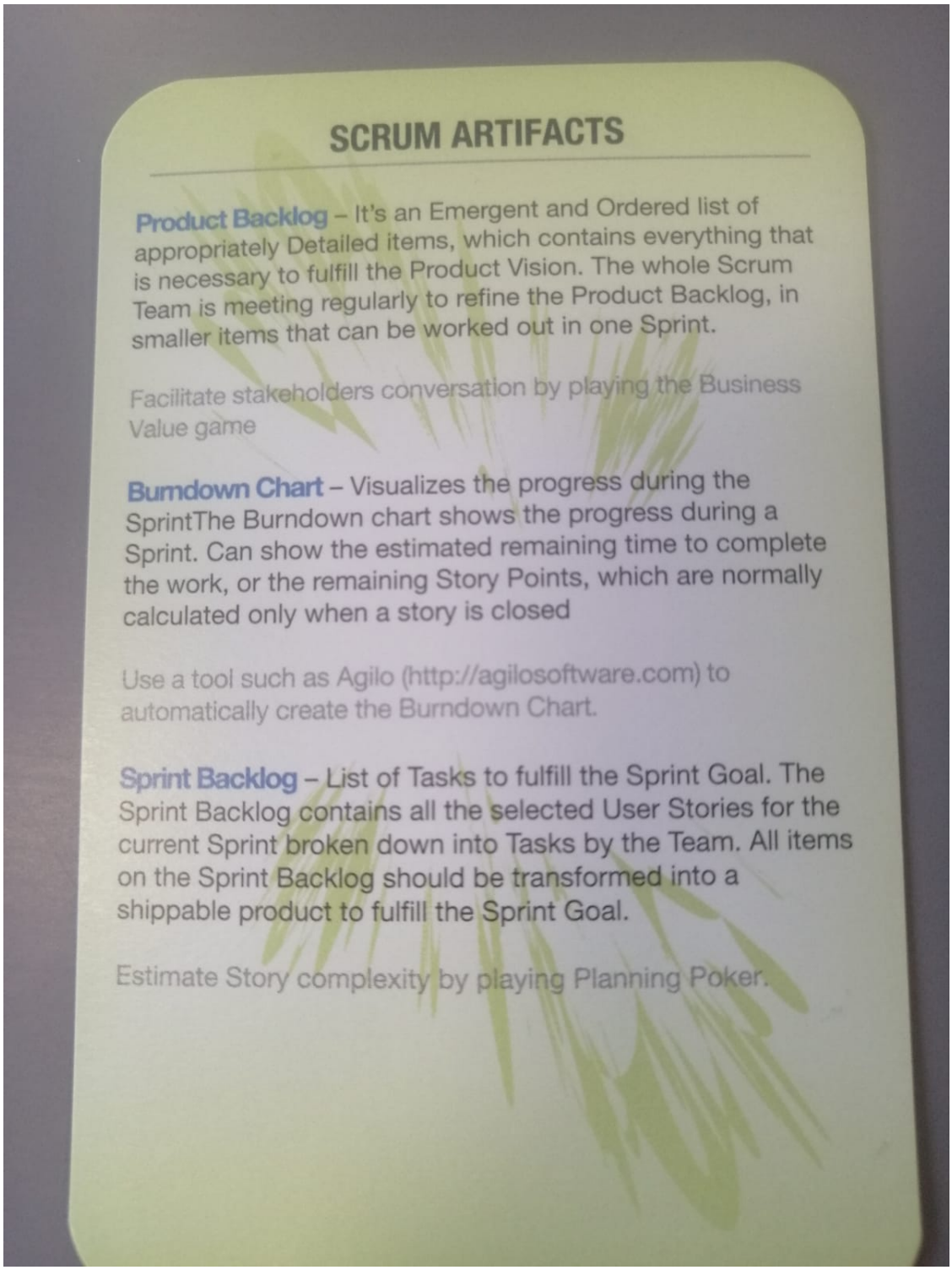
A discussion on how and why you choose to structure your backlog as you did [SJ: Perhaps argue why you prioritized your backlog the way you did]

Quote taken from google: "A product backlog needs prioritization. Items at the top are a higher priority, and items toward the bottom are a lower priority. When deciding which items should be prioritized, consider the value each item will provide."

We choose to structure and prioritise our backlogs based on 2 criteria:
1. Who is the stakeholder raising the request?
2. What values does it bring to the client?

We decide to prioritise client's request first. Among the requests by the client, we will then prioritise those that bring about the greatest value to the client. For example, we will prioritise having a proper payment service before having a discount feature. This ensures that the interest of the client is well taken care of, and each time we work on a request, it will incur the lowest opportunity cost.

Cards with definitions



SCRUM ROLES

Product Owner – Owns the Product Backlog

The Product Owner represents the interests of everyone with a stake in the project (Stakeholder) and he is responsible for the long term **return on investment (ROI)**.

- Doing the right things: order the Product Backlog to generate long term **ROI**

Scrum Master – Owns the Scrum process

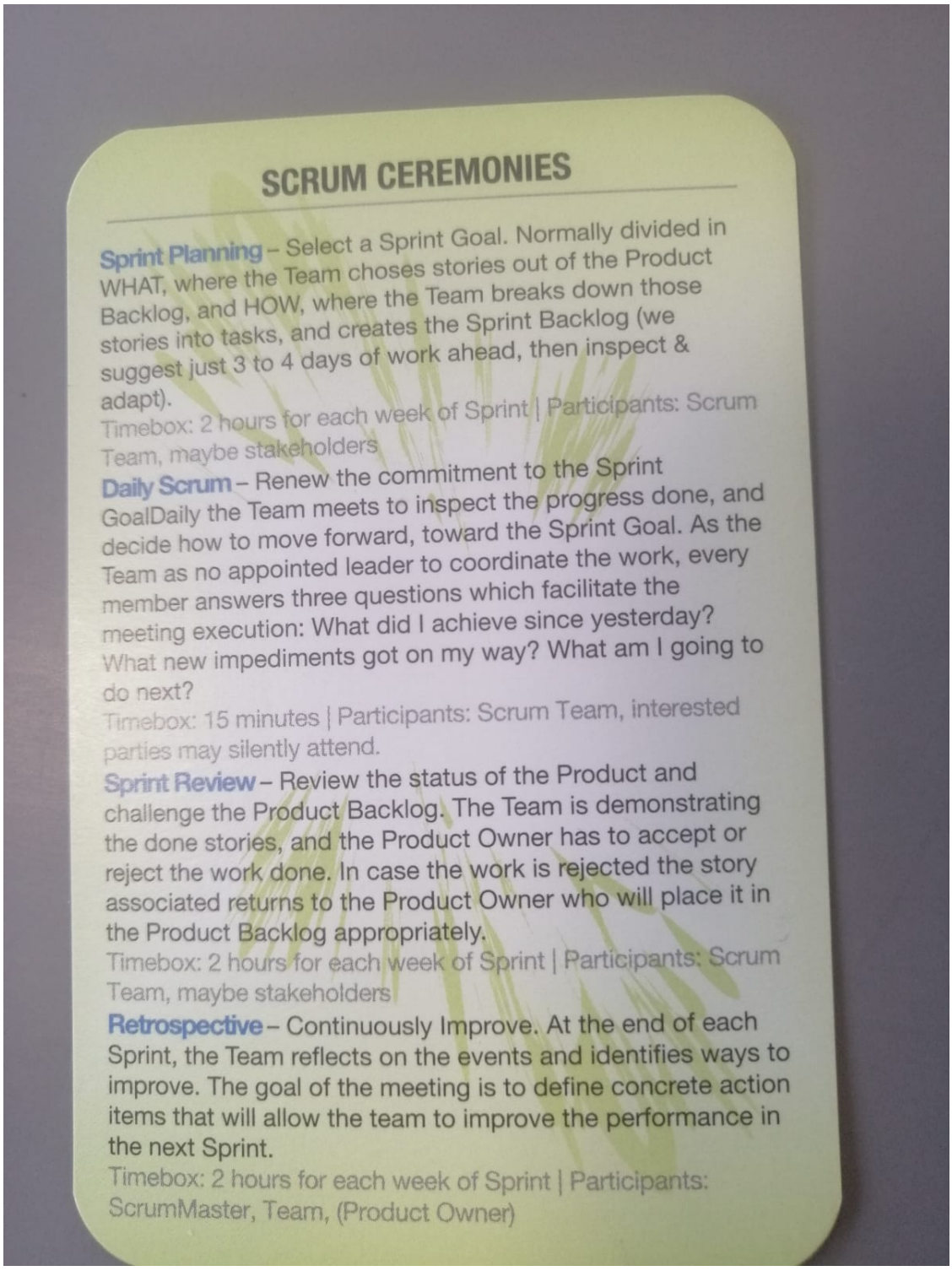
The Scrum Master is responsible for the Scrum process, ensures everybody plays by the rules and also removes impediments for the Team. The ScrumMaster is not part of the Team, and protects the Team from external interferences during the Sprint.

- Doing things fast: coaches the Scrum Team to improve speed, so they will learn faster.
- Remove impediments and facilitate events.

Development Team – Owns the Quality of the Product

The team figures out how to turn the Product Backlog into an increment of functionality within a Sprint. Each team member is jointly responsible for the success of each iteration and of the project as a whole.

- Doing things right: software quality
- Delivery of a "potentially shippable" product increment at every Sprint



[CX: Revised and approved. ✓]
[DK: Revised and approved. ✓]
[SJ: Revised and approved. ✓]

2 Section 2 : Name

Topics: SWE Topic 2

Section 2 Name

```
public f()
{
return 0;
}
```

References