**Adriana Sejfia**

**HW1**

**Foundations of Software Engineering**

**SWEN-610**

**Questions and Answers**

***1.5.*** Consider the seven software categories presented in Section 1.1.2. Do you think that the same approach to software engineering can be applied for each? Explain your answer.

***1.5.*** The seven software categories although broad are in the end software categories, meaning they have something in common. I think that basic principles of software engineering and each stage of the software process would be applicable for each of the categories. That being said, each of these categories have also distinguishing characteristics. Some categories are more safety-critical than others, i.e. scientific/engineering software, while others require faster delivery than others i.e. web-based apps. These distinguishing characteristics would sometimes require spending more time in planning than construction, or testing than construction or the other way around. Therefore, in general, it could be said that the same approach can be applied to all of the categories, but each category, more specifically, requires for that approach to be tailored to some extent.

***4.4.*** Provide three examples of software projects that would be amenable to the incremental model. Be specific.

***4.4.*** Three examples of projects that could be developed using the incremental model are the following: a) Media sharing app, i.e. sharing pictures, videos, and photos to a group of users of that app. The app could develop functionality incrementally, i.e. first delivering only the photo-sharing functionality, and then adding the other two.

b) Streaming and downloading movies, similar to Netflix, with the option of downloading the movie as well. In fact, the next increment of Netflix could be downloading the movies.

c) Research paper sharing software that has the following functionalities: sharing papers, citing from a paper, and searching based on tag values. Each of the functionalities could be delivered incrementally, without impacting negatively the rest of the functionalities, as long as the sharing increment is the first one to be delivered.

***4.9.*** Provide three examples of software projects that would be amenable to the component-based model. Be specific.

***4.9.*** In my opinion software projects that could be amenable to the component-based model are the ones that have commercialized functionalities, i.e. have been in the market for a longer period of time. Yet these project still require a degree of uniqueness due to differing requirements and environments. Three examples of software projects that could be developed using the component-based model, based on that criteria, are:

a) Health Information Systems – there are a lot of components available in the market that when added up could result in a health information system, such as patient information systems, pharmaceutical operations software and administration software.

b) Project Management Software – components that divide tasks, manage progression and enable communication can be found and integrated to create a project management software.

c) Financial Management Software – could be created by integrating components that deal with finance management,

***5.2.*** Describe agility (for software projects) in your own words.

***5.2.*** I think that the best way to describe agility is being versatile to changes and building your whole process and project based on that premise.

***5.3.*** Why does an iterative process make it easier to manage change? Is every agile process discussed in this chapter iterative? Is it possible to complete a project in just one iteration and still be agile? Explain your answers.

***5.3.*** An iterative process offers more flexibility in moving between the stages and it offers more possibilities to check the functionality of the software against the requirements. Hence, software engineers can introduce changes, compared to a non-iterative process, earlier on during a process, making it less costly time-wise and resource-wise. Pretty much all of the agile processes discussed are iterative in one way or another, either in a general way, i.e. process, or more specific way, i.e. functionalities delivered. Theoretically, it would be possible to deliver an agile process on the first iteration if there would be no changes in requirements requested from the customer, or no changes in the market. In practice however, this may never happen.

***7.1.*** Since a focus on quality demands resources and time, is it possible to be agile and still maintain a quality focus?

***7.1.*** Agility doesn’t mean you necessarily sacrifice quality; it actually means you perform more check-ups during the way to see that the quality requirements are being met, which I think, leads to better quality of the end product.

***7.3.*** Describe the concept of separation of concerns in your own words.

***7.3.*** Separation of concerns refers to breaking up a software project to the smallest pieces possible, but not further than that. That is, each ‘concern’ should be able to deliver a ‘separate functionality’ of the software, with as little to no help from other concerns.

***8.1.*** Why is it that many software developers don’t pay enough attention to requirements engineering? Are there every circumstances where you can skip it?

***8.1.*** Because often times software engineering is equated to coding, and everything else is considered as a waste of time. I think there are no circumstances where you could actually skip requirements engineering; there are just different approaches you could take to go through this stage. If you decide to go straight to coding, then the requirement engineering phase would be done while you’re coding in an informal manner, because one way or the other you have to accommodate the requirements made by the customer.

***8.9.*** Develop a complete use case for one of the following activities:

a. Making a withdrawal at an ATM.

b. Using your charge card for a meal at a restaurant

c. Buying a stock using an online brokerage account

d. Searching for books (on a specific topic) using an online bookstore.

e. An activity specified by your instructor.

***8.9.***

b. Using your charge card for a meal at a restaurant

Precondition: The customer has ordered a meal at a restaurant.

Post-condition: The customer pays for the meal through the check card.

Normal Flow:

1. The customer inserts/swipes the charge card

2. The system charges the payment to the charge card.

3. The system produces an invoice.

4. The customer gives the credentials for the invoice.

5. The system checks if the credentials match the ones provided at the issuer.

6. If the credentials match, the system asks for reimbursement from the issuer.

7. The issuer reimburses the system/restaurant.

8.The issuer charges the user for the amount.

9. Use case ends.

Alternative Flow

6a.1. If the credentials don’t match, the customer can try up to three times.

6a.2. When the credentials match, the use case continues to Step 6 of the normal flow.

Exceptions

6.E.1. Credentials don’t match even after three attempts.

6.E.2. Use case ends.

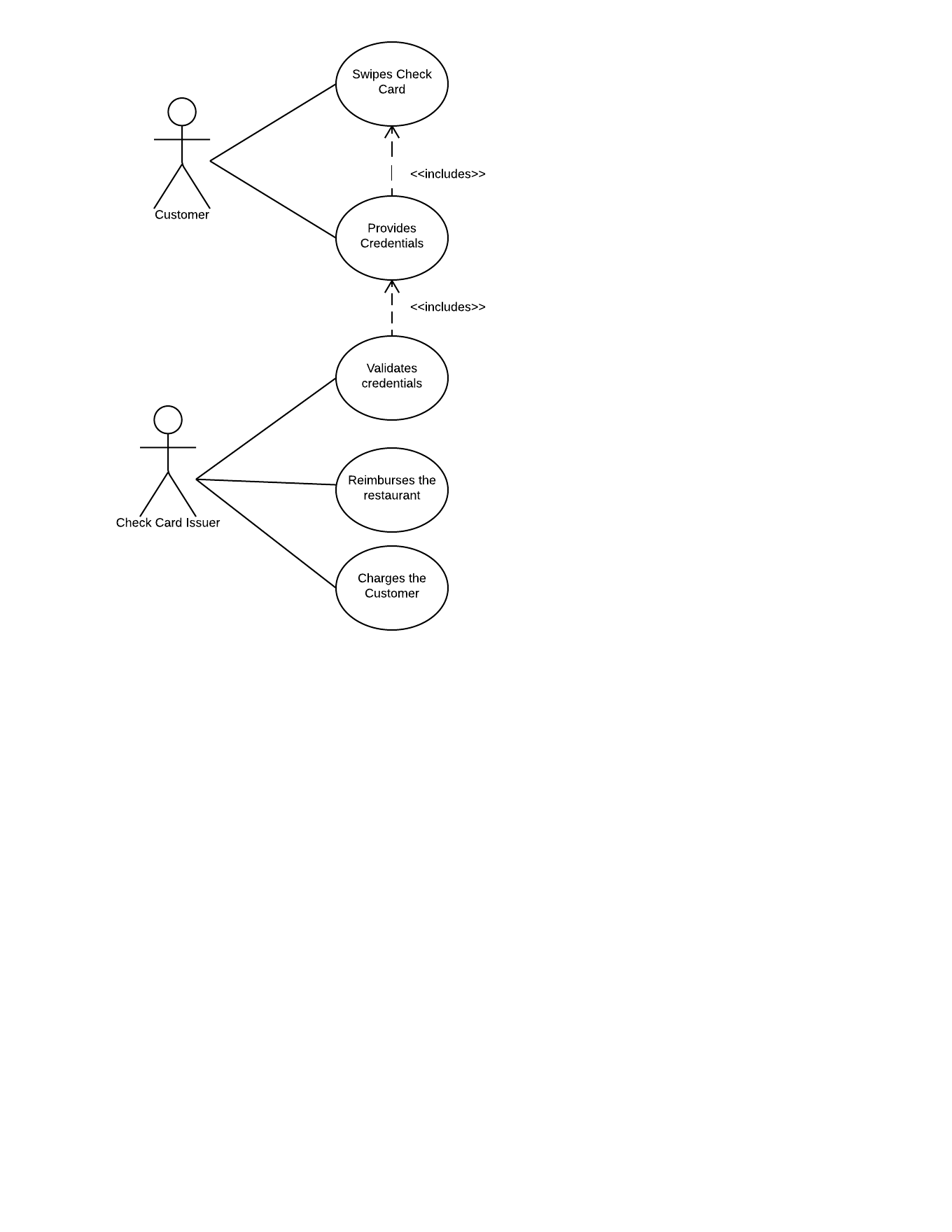


Figure Use Case for Check Card Usage in a Restaurant