Project – Shop of the future

INTRODUCTION

Supermarket chains like Walmart in the US, Albert Heijn and Jumbo in the Netherlands are constantly monitoring and improving profit levels by decreasing costs and/or increasing turnover. You can think of a common approach like cutting costs in procurement. For example, Walmart has been a main driving force behind the introduction of RFID tagging when it started demanding its usage from suppliers. The costs lowered as the overall intake scanning procedure became shorter[1]. Also attracting customers by either offering good value for money or giving extra service to customers, while maintaining a higher price level is commonly seen as a method to optimize profits.

In this learning task we will focus on the extra service offering ("Shop of the future") of a fictitious supermarket chain. It is inspired by an existing "Shop of the future" by Metro Group (Tönisvorst). It is located at Tönisvorst (near Venlo) in Germany. See for exampe http://www.youtube.com/watch?v=eefUwLHLGWo Until 2013 it was possible to apply for a free guided tour. The picture on the front page is taken from this website and depicts a device used for scanning 1D bar codes.

GOAL

The high level goal of this learning task is:

- The development of a fully functional **application** constituted of distributed components.
- The correct application of a rather formal software engineering approach: **RUP**. You are expected to be able to understand, judge, and use (parts of) this methodology.
- The ability to **advise** a customer on basis of technological and/or financial feasibility arguments.

DESCRIPTION OF CASE

A large supermarket chain has requested your development group to construct a Proof of Concept (POC). The purpose of the POC is to test whether some new emerging technologies are usable in a shopping and buying setting. The system must contain the following components:

- A **web application** on which customers can register themselves and compose a shopping list.
- A shopping cart/scanning aid on a **mobile device**. After the user logs in, the shopping list is retrieved. The user can consult this list and scan either 1D (classic barcode) or 2D (such as QR) codes from the packaging of goods. The scanned items are then either excluded (preselected on shopping list) or included (not previously selected on shopping list).
- At the exit the customer pays for his/her products. Scanned products are obtained from the scanning aid, and **a receipt is calculated** and presented to the customer. Customers are checked at random (suggested by the system) by rescanning the goods and comparing this with the alleged scan results of the customer.

---

1 Less time implies a small saving. Informally, a small saving repeated many times is a huge saving in the end. Formally, this is called economics of scale. Large retail stores are aware of this principle and use it abundantly in their procurement of goods.

Furthermore, the supermarket chain also wants to obtain more insight into the technical and financial feasibility of two major technologies in the context of their realm of operation. The first subject of interest is a suitable contactless **payment** (NFC) method. Consult for example http://open-nfc.org/wp/editions/android/ on emulating NFC. Current products are for example Google Wallet, Vodafone/Visa Smartpass and Rabobank MyOrder. The second subject is pushing **personalized advertisements** whenever a customer is in reach of the wireless network in the store. This can be solved by pushing messages to newly entered listening devices or pulling advertisements from the device itself. In a broader context this mechanism is related to the concept of 'nearables', see http://en.wikipedia.org/wiki/Nearables, usually based on Bluetooth low energy technology.

You are expected to experiment with[2] and report on these technologies without the requirement that they are functionally interwoven with the earlier part of the POC.

On your documentation you should always make a distinction between your prototype (PoC) and the final system that delivers the full functionality.

Finally, there are no constraints related to choice of implementation technology. What mobile & web platform you choose is up to you. Obviously, these choices have to be motivated.

EXPECTED MODE OF OPERATION

- You are obliged to use the **RUP methodology** during the execution of the project. Every RUP activity has its typical artifacts and/or deliverables. As RUP is iterative by nature, you will spend a considerable amount of time on improvement. It is quite possible that you construct and test some code representing functionality that is still being developed.
- The **deliverables** section below describes the *minimal* set that must be present at the end of the project. Any non-compliance to this rule implies an automatic project failure. The project group and/or individuals will be given clear instructions on how to proceed.
- Setting the right **priorities** is of paramount importance. For example, spending a large amount of your time on a flashy website is clearly a wrong choice. Be sure to identify the basic functionality of all the constituent parts and start with this. Extra functionality could be added in a later stadium.
- It should be clear from your documentation which **functionality** is included in the POC and which functionality is **not** included in the POC but is desired in a deployable version.

This might be one of the lasts projects you will encounter during your period as a student. Your professional skills, technical and non-technical, are expected to meet some standard. This standard can be formulated as "Behave as if this a real project for a real customer". This implies certain behavior summarized below.

Day to day presence is required, also during feedback sessions. Make clear agreements on which individual behavior is tolerated. Working at home is fine for a one-off occasion, but it is by no means tolerated as the standard mode of operation.

---

2 This means getting your hands dirty on real code.

DELIVERABLES

DELIVERABLE 1/7 - VISION DOCUMENT

This document contains scope, goal and stakeholders of the project. A business context underpinning the problem description should be present. Be very clear on the scope of the POC, but also make clear what could be possible with a deployable final version.

DELIVERABLE 2/7 - SOFTWARE DEVELOPMENT PLAN (SDP)

This document contains:

- Regarding 'Scope, Goal and Stakeholders', refer to Vision Document.
- List of deliverables.
- Who is responsible for what?
- Activity list ('list of tasks to be done').
- Planning with dependencies between tasks
- Iteration plan: which iteration contains which scope?
- How quality assurance is implemented.
- How configuration management (version control of code and documentation) is implemented.
- List of project risks and countermeasures. Supply real risks, not trivial risks like collective illness.

The aim of the SDP is twofold: it helps the project manager to manage you as a group. Secondly, you tell the customer what will be delivered and how your project approach is. The customer should get the impression from this document that everything is under control. Do you think that a neutral reader of this document will get this impression as well?

DELIVERABLE 3/7 - SOFTWARE ARCHITECTURE DOCUMENT (SAD)

This document should contain the 4+1 view paradigm:

- Use Case View consisting of a use case model and use case description.
- Logical View containing the static parts such as application layers, class diagram and ER diagram.
- Process View containing the dynamic part represented by some important sequence diagrams. It should make the interaction between subsystems clear.
- Implementation View in which the modules/packages, frameworks, libraries and configurations are described.
- Deployment View with a description of topology of the network and which parts contain which software. Also pay attention to security issues.

DELIVERABLE 4/7 - TEST DOCUMENTATION

This document contains several artifacts found in the RUP document suite. Each mentioned document below should be a chapter in this composite document.

- Test Plan (Project plan of the test activity, chosen testing types, etc.).
- Product Acceptance Plan (On what criteria will the POC be accepted?).
- Test Report (Contains analysis and summary of tests (like overall statistics) and actions to be taken on basis of the results).

DELIVERABLE 5/7 – DEPLOYMENT MANUAL

This manual (around 3 pages) should contain:

- Install procedure
- Configuration settings
- What is expected to work and what is known not to work in the POC?

DELIVERABLES 6/7- ADVICE

This should be one document consisting of:

- Implementation plan. What should be taken into account when the system is deployed on a large scale throughout the country? What are the major issues to be dealt with?
- Advice concerning the possibilities of contactless payment, supplemented by some actual code in an appendix.
- Advice concerning the ability to push personalized advertisements, supplemented by some actual code in an appendix.
- Advice concerning ethical aspects. No matter how technologically advanced and functionally rich a system might be, whenever there is any doubt on the ethical aspects of the system, its adaptation will fail. Which

DELIVERABLE 7/7- SOURCE CODE

We nearly forgot to mention it but the source code is of interest as well. You are expected to deliver good quality code (implementing a POC does not mean it is should be hacked code). Points of attention are:

- Structure of the code, including package structure.
- Relevant and concise documentation of the code.
- Mature exception handling.
- Proper logging facilities.

PRESENTATION AND DEMONSTRATION

Every project group must present its results to the complete group of students. This should take not more than 20 minutes and must include a demonstration. Be aware of the fact that every group knows the context of the problem, so keep time spent on the general problem formulation to a minimum and try to focus on the unique points of your solution as this emphasizes the diversity of the implementations. There will be time for questions at the end of the presentation.

ASSESSMENT (GRADE)

The following aspects are taken into account when the overall grade is determined:

- Management: how did the project execute from a RUP and collaboration perspective? Is the document suite complete and relevant? Also, the individual participation (hours spent, work done) in the project is taken into account.
- Requirements: did you understand want the customer wanted? Did you manage to describe this properly? How well are the stakeholders, the goal of the POC, and the deployable system described?
- Architecture: can you describe this and elaborate on the collaboration of the different parts?
- Implementation: solid structure and conciseness of the code. Documentation of code.
- Testing: how thorough was it done, written down and what is the overall impression of the result?
- Advice: quality of advice regarding NFC and pushing of personalized advertisements.
- Presentation and Demonstration: overall impression.

The documentation may be written in Dutch. It will be judged on the use of your language, grammar, overall structure and representativeness.