PS Software Engineering

NC-vvv-yyyy November 7, 2001

ImmoWarrior

Report

Andrea Geierspichler, Michael Pober, Roman Gross, Christian Koidl, Gunnar Ruhs

Department of Computer Science
University of Salzburg

Correspondence to:
Universität Salzburg
Institut für Computerwissenschaften und Systemanalyse
Jakob–Haringer–Straße 2
A–5020 Salzburg
Austria

ImmoWarrior

Chapter 1

The Project

To learn the how-to's of "Software Engineering" we simulate a real life situation in the pro-seminar. A client, represented by the seminar-leader, comes to our software-development-firm and gives us an assignment.

The initial information is: A company that auctions real estate wants us to develop a software-solution for their process of collecting and representing information. (Code: "Immowarrior). Potential users of this system are: the ones, who gather and enter the information and the ones, who want to read it. Those readers can either be persons, or other systems. This leads to two different scenarios:

- Immowarrior sends the information to an other computer system by using XML
- Immowarrior displays the information for a person - what requires us to design an appropriate user-interface.

Specially required by the client is, that the software is to be easily maintainable and that no one has to install client-software to be able to use Immowarrior.
For the first phase of the project, we ought to concentrate on the basic-functions of the software, but we should keep the possibility in mind to add expansions, developed in co-operation with our group.

This is the requirement profile for our project. The given deadline is the end of January 2002

Chapter 2

A CRC Session

What is a CRC-session?

The CRC-session was invented by Ward Cunningham and Kent Beck in 1989 and then popularised by Rebecca Wirfs-Brock.CRC stands for Class, Responsibility and Collaborator.It is an informal technique for object oriented analysis and design, used for visualising the structure and behaviour of object oriented systems. Basically standard index cards are used to represent classes, responsibilities and colaborabtions with other classes, they are collected on the frontside. On the backside descriptions of the classes and attributes are witten down. A class is an abstraction of something from the problem domain. Responsibilities are tasks that can be ascribed to particular classes and a collaborator is another class that you require to talk to in order to carry out your own responsibilities. The goal is to define and to structure the problem domain and to represent the problem in an object oriented way.

Organising a CRC-session

The number of people is important, too few and it becomes difficult to mentally change gear, too many and you are not all involved and the productivity is cut by more disagreements. Five is the ideal number. If there are more people, one solution is to have the extra people be present strictly as observers. The group should be composed of developers, domain experts and an object oriented faciliator. Try to avoid having an excessivly dominant participant - no bosses either. From the system requirements decide on a simple set of scenarios that you want to try out.

Starting the session

The first step is to extract classes from the problem domain. One useful tool is to find all the nouns and verbs. The nouns are a good key to what classes are in a system and the verbs show what their responsibilities are. With this information you can start a brainstorming session. Now analyse which classes are important for the project. You do not need to find all classes and responsibilities. The scenario will make them more obvious later on. In a CRC-session subclasses and superclasses are also defined. This can be done at any time it becomes obvious. Attributes of the class do not really have to be defined right now. This will be necessary when you get to the design phase.

Defining Use-Cases

Each person should take the responsibility for being a class. Select a scenario, identify and allocate new classes and responsibilities as the scenario unfoldes. Each person has to focus on their own class. The list of classes will grow and then shrink as the group filters out the good ones. Afterwards try out some speacial scenarious like exceptions. Later on, when getting to the design phase consider folowing things: Target environment, language choice of supporting software components, performance requirements.

Perform Use-Case Scenario Testing

1. **Call out a new scenario.** In this phase of the CRC-session the scenario is discussed and described in basic
2. **Determine which card should handle the responsibility.**
3. **Update the card whenever necessary.** If a card needs to be updated one of two situations has arisen: The responsibility needs to be added to an existing card or a new card needs to be created.
4. **Describe the processing logic.** Now the bussines logic for the responsibilities should be described step by step and the rules for the system should be recorded.
5. **Act out the collaboration between the different classes.**

Advantages of CRC-Modelling

1. The analysis is done by people who understand the problem domain (experts)
2. The future-users are actively involved in defining the model for the program.
3. CRC-modelling can easily be explained to a group of people.
4. The CRC-session leads directly into class-diagramming.

Things you have to keep in mind

- Start with simple scenarios.
- Take the time to select meaningful class names.
- Take the time to write a description of the class.
- If in doubt, act it out!
- Layout the cards on the table to get a feeling for system structure.
- Be prepared to be flexible.

Appendix A

Protocolls

A.1 1^{st} Lession - October, 25^{th}, 2001

Basical information about the course and division into groups.

A.2 2^{nd} Lession - October, 31^{st}, 2001

First meeting with the client and CRC-session.

1. Groupleader: Christian Koidl
2. Participating Groupmembers: Andrea Geierspichler, Gunnar Ruhs, Roman Gross, Michael Pober
3. Topics talked about: Analysis of the problem and basic class-definition
4. Results:
   (a) The client wants the possibility to obtain relevant information on offers via the internet or the mobile phone.
   (b) The information should be sent to the customer either automatically by the system (push) or on individual request (pull).
   (c) Clients as banks, authorities and private persons get only one account.
   (d) The system should provide an access-control-system.
   (e) Private persons receive their username/password by e-mail.
   (f) Private persons have to pay provision for publishing their information
   (g) There should be the possibility to include different multimedia-elements (pictures, movies, sounds,...).
5. List of possible classes: Session, User Interface, Account, Permission, Information Storage, Real Estate, Editor, Private Person, Prospective Buyer

6. New assignments: Refining the problem-statement. Discuss and act out possible use-cases. Think about the user-interface.
7. Open questions:
   (a) Should permanent accounts be available for privat persons?
   (b) Do private persons have to pay a fee although the object is not sold?
   (c) Is it necessary to have an account to request information?