



Lady Bug

Author: Mira Kajko-Mattsson

Reviewers: Barry Myburgh, Paul McMahon, Cecile Peraire, Maria Augusta Nelson

Lady Bug is a private kindergarten. So far, it has managed to administer all its business with the aid of simple tools such as MS Word, Excel, PowerPoint and pen and paper. The information they handled dealt with children, their parents, children's preferences, health and allergies, kindergarten activities, finances, employees, and the like.

Recently, Lady Bug has discovered that using these tools has become difficult and cumbersome. Because they had the same information in several places, they had difficulties keeping up with changes and were starting to notice inconsistencies. All this had a negative impact on parent satisfaction and obstructed the management of the kindergarten's core business.

One instance has been particularly devastating. A boy, Peter, suffered from gluten allergy. On the first day of Peter's arrival at kindergarten, this information was registered in the main office Word file but not in Peter's binder record. One day, a new employee, Mary, unknowingly served Peter bread with gluten. She did not have access to the main office Word file, only to the binder. Peter became very ill for about two months. His parents had to stay at home with him and the Lady Bug reputation was badly damaged.

To improve their business operation, Lady Bug decided to acquire a new system that would support their business management. The system was to be developed by a small company called *Administrator AB*. The company had five employees and they focused on developing simple information systems. Despite the fact that developing kindergarten management systems was new to them, they decided to accept the project. It was a great opportunity to get established in a new market. A new project called *Pippi Longstocking* was started.

To date, *Administrator AB* has worked on the project for two months. Their work has, however, progressed very slowly. It seems that no one at Lady Bug has time to communicate with Administrator's development team. So far, they have had only two meetings. During the first meeting, Lady Bug's manager, Ann, presented the problems, requirements for the system and listed future system users. All except the kitchen personnel should be able to work with the

1. As a user (kindergarden manager), I shall be able to record information about the children.
2. As a user (kindergarden manager), I shall be able to create personnel schedules.
3. As a user (kindergarden employee), I shall be able to access information about the children's parents.
4. As a user (kindergarden manager), I shall be able to record all the planned kindergarden activities.
5. As a user (kindergarden employee), I shall be able to record all the finalized kindergarden activities.
6. As a user (kindergarden manager), I shall be able to record menus – list of dishes that have been served and that are going to be served.
7. As a user (kindergarden manager), I shall be able to record information about children's health.
8. As a user (kindergarden employee), I shall be notified about children's health problems.
9. As a user, I shall be able to see which staff member took care of which children at a particular point in time.

Figure 1. Ann's initial requirements

system. Because Lady Bug staff were very busy, she did not wish to involve them in any meetings with Administrator. She insisted that her support would be enough.

During the second meeting, *Administrator* contacted Ann to validate the requirements and to agree on their prioritization. Figure 1 shows the initial nine requirements. Andrew, the project manager, presented the requirements to Ann who confirmed that they were correct. Ann, however, could not provide any feedback on how to prioritize them. To her, they were all of the same priority. She left this to Andrew and his team instead. Neither did Ann specify any other requirements concerning the constraints, system quality, and technologies to be used. Why should she? She had no idea that she, as an acquirer, had the right to request all this. The only request that she provided was that the system should be run on Windows 7.

By the end of the second meeting, Ann and Andrew wrote a contract. Because Ann did not have a full picture of all the requirements, Andrew suggested that Lady Bug pay for the time spent on system development. It would not pay for delivered functionality. Andrew assured Ann that this was the best solution. Ann would be free to change requirements at any time and decide when to stop development. It was agreed that Lady Bug would pay 50 000 USD for the first three months of work. Such a solution was of course optimal for Andrew and his team. They were very glad that Ann did not bother how they would work. This gave them freedom to make their own decisions. Neither did they inform Ann that standard systems with similar functionality already existed on the market. Why should they? They would then not win the development project.

Back at the office, the team consisting of Paul, Robert, Richard, and Patricia started working on the system. Paul and Robert have been working together at *Administrator* for about a year, but Robert and Patricia were new to the team. Robert came from the aerospace industry where he worked for 15 years, while Patricia had just finished her Masters degree in Software Engineering.

The team first decided to implement the system in Java and Access database. Then they started elaborating the requirements, which generated many questions. For instance, what kind of information should be recorded about the children and their parents? What types of activities do kindergartens have today? The developers were desperate to get answers. However, they did not

dare bother Ann with questions. She was not the type of person that one contacts spontaneously. However, Paul's daughter was in kindergarten and he had a rough idea about what information might be needed.

The team outlined a plan and decided on the priority of the requirements. They determined that the system would be implemented using the "database repository" architectural style. The functionality was to be implemented in different applications sharing data via a central repository. Because it was difficult to know what data such a central repository would include, they decided that they would evolve it while implementing the requirements.

Instructions

The above-described scenario is an imagined outline and excerpt of a simple project story. Its purpose is to provide a basis for exercising the Kernel by testing the ability to analyze the scenario and evaluate the project status by using the Kernel Alphas. The purpose is not to provide a fully-fledged methodologically and technologically advanced project description.

Your task is the following:

- Read through the scenario description.
- Discuss the scenario in group and commonly resolve any problems with understanding and inconsistencies.
- Determine the tools to be used in your evaluation effort (Alpha checklists versus Alpha cards)
- Evaluate the project using the Alphas. There is no specific order of Alphas to be chosen for the evaluation. You are free to choose any alpha in any order as long as you feel comfortable with your choice and feel that you are making progress.
- While evaluating the state of each alpha, the following steps are recommended:
 - For each alpha:
 - Choose the alpha to be evaluated.
 - Allow each group member to study the alpha and first select the state on his/her own.
 - Allow all the group members to present their individual estimates.
 - Discuss the estimation results and resolve disagreements, if any.
- Review the agreed state of each Alpha and discuss the overall condition of the project.