

## Complex Testing of ERP System for Automating Work of Car Dealer Networks

### Customer

The company developed the ERP system that would allow substituting the current solution for automating business processes of the company's branches.

<b>Company</b>	<i>Car dealer, car service provider</i>
<b>Country</b>	<i>Russia</i>
<b>Business Domain</b>	<i>Auto Business</i>
<b>Services Rendered</b>	<ul style="list-style-type: none"> <li>✓ <i>Requirements testing</i></li> <li>✓ <i>Prototypes testing (functional and technical)</i></li> <li>✓ <i>GUI and usability testing</i></li> <li>✓ <i>Functional testing</i></li> <li>✓ <i>Load testing and stress testing</i></li> <li>✓ <i>Automatic testing</i></li> </ul>
<b>Communication Model</b>	<i>Rendering services in quality assurance for the software developer, independent testing and quality control</i>
<b>Duration</b>	<i>2 years</i>
<b>Labor Efforts</b>	<i>20 man-years</i>

### Project

The aim of the project is to develop an office work information system that would automate the following business processes of the customer's branches (*Enterprise Resource Management*):

- managing all company's resources (human and non-human)
- rendering various services to company's clients
- sales process
- supporting communication with customers (internal CRM system)
- supporting working relations and financial operations of the customer with various private individuals and legal bodies that are involved in the life cycle of the whole car service's office work (spare parts suppliers, dealers, car manufacturers, etc.)
- managing extensive number of reference data content
- managing a compound security complex and adjusting data access levels

The main aim in developing a new web system was to replace a similar system that did not prove to be appropriate, being not easy to use, complicated, and too expensive in managing and making changes in terms of dynamic business processes modifications in the customer's production, as well as changes in the legislation of the Russian Federation, and other external factors.

To meet these requirements, the development was performed involving open-source solutions (freeware components and software) and using new effective technologies and architectural solutions.

### Challenge

The main A1QA tasks were the following:

- to arrange a process of quality control that would fit the most to the specific character of the project, from the very beginning of software design and development

- to plan and perform a set of activities aimed at software quality control
- to develop testing strategies and perform effective quality control activities (e.g. full requirements examination, and automation) in order to decrease the cost and shorten the period for development of the system
- to arrange the process of effective communication within a large project team

## Solution

To provide high quality of the end product, testing must begin in the beginning stage of the project's life cycle. Before launching the project, specialists that were most skilled in testing complex applications with complicated business logic arranged and fully documented regulations for the work of a quality control project team. Apart from that, various requirements testing methods were analyzed: critical analysis, examination, testing a business model by overlaying it with a testing model that is being developed following the requirements set. Each of the methods above was successfully implemented during all the period of the system's development, depending on the type of a requirements testing task.

During the life cycle, the quality control project team performed the following operations:

- **Testing the Requirements.** The testing was held on the stage of their confirmation before the start of the project design, which helped avoid logic defects in the planned functionality, decreased the risk of additional changes in the architectural solutions caused by such defects at the product realization stage and the cost of these changes.
- **Testing the Functional Prototype.** The testing helped detect logical defects as well as mismatches in the application's structure and navigating system, create a well organized and clearly structured system with perfect user interface. It would subsequently allow end users to master the new system promptly and help them avoid extensive additional training and reduce support costs.
- **Testing Technical Prototypes (Functional Testing and Load Testing).** The testing helped choose the most stable architectural solution out of two solutions, that would satisfy the customer's requirements.
- **Functional Testing and GUI Testing.** The testing included different testing types on different stages: module testing, testing the integration of system components, testing access rights and rights management system (this would allow creating a flexible system that would be adjustable for the customer's different needs and changing processes), testing the system's usability, etc.
- **System Load Testing.** The testing allowed performing diagnostics of the system's work on-load and eliminate performance problems on early development stages.
- **Automates testing.** The testing was used for saving time while testing invariable parts of the application as well as parts that required large time expenditures and regression functional testing after developing new system components. The testing allowed saving time during stabilizing the project and define the stability of the product's release versions.
- **Organizing Acceptance of the System.** The acceptance was organized on the customer's side. As a result, acceptance testing was completed on the due date and showed positive results.

The customer's QA team was mainly involved in holding acceptance testing and further implementation of the product.

## Cooperation with Project Team

The project team was divided into three teams:

- business analysis team
- development team
- testing team

Arranging effective cooperation between all the teams within the project was the key factor in providing the project's success. During the stage of testing the requirements and describing the testing model, the closest cooperation was established with the business analysis team, which formed system requirements. Apart from arranging the process of managing requirements defects, the issues of requirements precision, consistency, and completeness, as well as their ability to correspond with the

customer's expectations and the company's business processes were discussed regularly. Suggestions on improving the requirements and the format of the requirements, as well as the content and, therefore, functionality planned to be implemented.

During the stage of testing the system's functionality, all the issues concerning requirements and arrangement were discussed using email, MSN, and phone conversations involving all three teams. The most effective solutions on eliminating errors were drawn up.

During the acceptance stage, an infrastructure for performing acceptance was developed for the testing team on the customer's side (set-up systems builds, lists of testing scripts for performing acceptance, and consulting by the A1QA testing team on the system's functionality).

## Technologies Used

- JMet, Zabbix
- Selenium, JUnit
- FireFox 2.0+ / IE 6.0 +
- Rational ClearQuest (including Rational ClearQuest Web)
- Borland Caliber RM
- Mercury Test Director

## Results

Depending on a phase of the project, the results of the team's activity can be divided into several parts:

### Requirements Development Phase

- Using the cooperation of requirements testing teams and implementing the most effective testing strategies applied for different types of requirements before the start of the development, some serious defects were detected and eliminated while testing documentation and prototype. This helped reduce working efforts for the development and time spent on developing the project, and therefore avoid possible expensive modifications.
- The customer highly appreciated the ability of A1QA specialists to promptly analyze requirements, understand complicated and specific business processes, and effectively cooperate with the project team remotely.

### Prototyping Phase

- Load testing and functional testing of the technical prototype allowed comparing two architectural solutions and choose the one that met the stability and performance requirements the most. As a result, the risk of modifying the system in case of choosing the wrong architecture was minimized.
- For a functional prototype, the A1QA team suggested GUI and usability testing (ease of navigating and interface consistency). After performing the listed activities, common recommendations on GUI design were elaborated, and the development was performed in a single pattern. As a result, a clear and easy-to-use GUI was developed.

### Active Development Phase

- Complex approach to testing allowed detecting system's functionality, performance, and interface on early stages and eliminate them promptly. As a result, the customer obtained a stable system satisfying all their requirements.
- Due to timely complex of activities aimed at integrating system components on the requirements level, building and integrating separately implemented functional system components was performed in a short space of time and with minimal number of issues arisen.

### Acceptance Phase

- Detailed acceptance scripts prepared by the A1QA specialists allowed the testing team on the customer's side to effectively join the testing process and successfully perform the acceptance on the customer's equipment.
- Precise planning, coordination and cooperation of two remote QA teams during the testing process helped perform the acceptance within the planned time limit.

## Organizing the Quality Control Process

- The effectively built process of managing defects and queries for requirements change helped effectively control all changes and, accordingly, total amount of project implementations, which helped make the project more manageable.
- Detailed testing scenarios in the testing scenarios management system helped increase the team during peak periods in order to perform comprehensive testing, with minimized risks of disorganizing the working process.