

Customer

Silver Star Agencies BV is an agency company which handles sea going vessels into several ports at Europe and the Baltic.

Challenge

To support its business a custom application is used for registration of tasks, times, cargo, crew, stores, bunkers and vessels details. This application (further named PortMaster) is an existing application that needed to be completely re-developed to support nowadays needs.

The request for developing custom framework was non-trivial due to necessity to combine the best from Windows Communication Foundation and Devexpress XP objects to achieve simplicity of definition of distributed system entities and their implementation.

Solution

Investigation of the Devexpress components' code was done to get the knowledge about their internal work. Further the investigation results were proved by implementation some series of prototypes that demonstrated benefits and disadvantages of using various approaches with minimal costs spent on it.

Moreover, several improvements have been developed for this framework, such as: custom validation rules applied on the entities; security component that enhances the system security. It was possible to implement due to the flexible component-oriented architecture where best-practices and design patterns were used, such as: dependency container, abstract factory, adapters and so on.

Interim results

Currently we are working on implementation of the service performing MS exchange folders synchronization. This can be achieved by combination of the best from the following platforms: .NET and unmanaged C++ coding. First allows us to develop reliable service engine and user interaction through .NET framework along with power of WCF for inter-process communication. The second provides all power of working with system-level components (such as: MAPI) to perform low-level operations with the data. We are looking forward to the integration of the exchange service with a system.

To build those systems we were using the best from development process, like: unit tests, integration servers (for continuous building and automated tests run), and functionality prototyping (allowed us to significantly improve code quality and mitigate technological risks that might appear on the project).