

# Content Management Web Application (CMWA)

## SharePoint 2010 Intranet Portal

The **Skyce Software Solutions** team is proud to present its contribution in designing and developing a Content Management Web Application, a very challenging project which allowed us to prove that our professional approach and innovative skills can keep up with an international customer demands.

## Challenge



### About customer

The customer is a German based subsidiary of a known automaker which is part of the world's top 5 largest automobile manufacturers with over **80,000**<sup>1</sup> employees and millions of vehicles sold in more than **190**<sup>2</sup> countries.

### Business needs

This CMWA is meant to provide all the required tools to allow the customer monitoring over **42,000** car model configuration variations (grouped by different characteristics) in terms of: new releases, specification updates and planning discount campaigns for car sales.

<sup>1</sup> Data available at end of year 2011

<sup>2</sup> Data available at end of year 2011

## In brief

The main characteristics of this project exposed in a brief view:

- Industry: Automotive
- Country: Germany
- UI language: German
- Type: Web-based
- Main platform: Microsoft SharePoint 2010
- Start date: January 2011
- End date: April 2012
- Total effort: 50+ man-months

## Customer

Relevant business facts for this client:

- Place in world's top 5 largest automobile manufacturers
- Over 80,000 employees in more than 190 countries
- Millions of vehicles sold world-wide

## Features

The CMWA functionality is defined by the following capabilities:

- Unified tools for managing tens of thousands of items in single or bulk editing modes
- Assistance spread all over entire hierarchy based on workflows, scheduled jobs and e-mail notifications
- Discount campaigns can be defined and resized on the fly
- Integration with SharePoint 2010, Active Directory and other content management modules
- Extended sorting and filtering possibilities

Basic car components like: engines, gear boxes, cushion and others are maintained from dedicated application screens.

CMWA offers also an **approval workflow** which pushes up the hierarchy detailed e-mail notifications regarding updated car models (offering a clear view of the old and the new values at specification field level) and/or newly imported car configurations.

**Discount campaigns** may be defined at car model category level (like for an entire compact or limousine class) or at specific car models level (e.g. the diesel version of a 4-doors compact with automatic gearbox).

Any running or future-planned campaign can be enriched on the fly with new models as also specific models can be excluded at any time, the entire system readjusting itself transparently.

### Integration

CMWA integrates with other existing content management modules from customer's infrastructure (Active Directory integration is also required) and receive once per day over 42,000 updated items representing car model configuration variations.

### Flexibility

This amount of car models should then be managed (by single or multiple item selection) in a centralized yet flexible way to edit each descriptive field. Because the CMWA is accessed from multiple locations, the concurrent editing protection mechanisms are required.

### Performance

Despite the huge amount of data processed daily basis (imports cumulated with average expected user data editing) and its multi-user access characteristic, the overall performance of CMWA should remain high and fluid.



## Highlights

Aspects nominated by the customer as being high level goals:

- Unitary, ergonomic with a modern look UI
- Excellent performance for data related operations
- High configurability level for all application components
- Extendible architecture

## Solution



The project took over 50 man-months to complete and was updated by many change requests during its roadmap.

The **Agile** software development method combined with very efficient object-oriented programming practices, like **Inversion-Of-Control** and **Dependency-Injection**, allowed our team to be flexible and efficient against every feature re-design and keep the project's delivery on time and on budget.

### Technologies under the hood

The best results may be achieved only by using the best tools and most appropriate professional skills. The CMWA was not an exception for this rule and below is a brief view of the most relevant technologies our team bound together for the success of this project.

#### Software

- Microsoft **SharePoint 2010 Enterprise** (with publishing infrastructure enabled) integrated with Active Directory.
- Microsoft SQL Server 2008 R2.
- Microsoft Windows Server 2008 R2 (as Hyper-V machines).
- Microsoft Visual Studio 2010 with **Team Foundation Server** integration. All the code behind was written in C#.
- **Telerik UI Controls** (customer choice for the UI look-n-feel).
- VPN access to allow any-time testing on our staging machines.

## Technology

The technology footprint of this project consists mainly in:

- Microsoft SharePoint Server 2010 integrated with Active Directory
- Microsoft SQL Server 2008 R2
- Microsoft Windows 2008 Server R2 with Hyper-V
- Microsoft Visual Studio 2010 with Team Foundation Server
- ASP.NET and C#
- Telerik .NET UI controls
- AJAX / jQuery / CSS
- Custom Web-Services
- Custom Web-Parts

## Involvement

A list of tasks our team successfully completed:

- System architecture and business analysis
- SharePoint, Web-Service and T-SQL development
- Testing
- Performance optimization
- Maintenance

- Various **performance measuring tools** for SharePoint, SQL Server and Visual Studio.
- **Compliance check tools** for CSS and HTML.

## Hardware

- HP ProLiant ML360 G6 Servers based environments for virtual machines.
- **Multi-CPU/Multi-Core** configurations with up to 48 GB of RAM.
- 4 x Hard Disks running at 10,000 rpm with **local and geographic redundancy**. Automatic daily back-ups for development environments (including source code and databases).
- Power source (including UPS stations) and **Internet access redundancy**. The Internet connections are supplied by different providers.

## Development cycle

Prototyping and iterative design have a reputation for being difficult to manage. We think the keys to handle a good prototyping are: knowing what you want to learn from the prototype, access to rapid prototyping techniques, and end-user involvement in development of the prototype. The benefit is creating a **more end-user-oriented product** faster because the customer is able to ‘feel’ the product earlier and adjust it before refactoring becomes too expensive.

## The analyze phase

The project started with understanding and defining customer’s business and technical needs. All the gathered information was used to build the **project development plan** (including time and budget estimations). The appropriate architecture, technology and interface design are chosen.

Being such a complex application, the customer constantly readjusted CMWA’s business needs by adding new features that naturally branched out from the initially planned ones. For each such feature a new iteration of the development cycle was started to reanalyze the

## Methodology

The Agile methodology accelerates the delivery of high quality custom software keeping low costs

- Prototyping the key features for an earlier look-n-feel of the product
- Iterative design to allow often customer testing and validation
- The end-user is involved during all phases of the process
- Continuous communication with the customer
- Constant recheck of the initial goals to keep up the product’s usability and performance
- Standard compliance tools to ensure the quality

## Infrastructure

Our standard infrastructure for any software project we develop includes:

- Multi-CPU/Multi-Core machines with up to 48 GB of RAM
- Top speed hard drives with geographical and local redundancy
- UPS stations and Internet access by different providers

entire product just to make sure it stays **solid and coherent from the UX<sup>3</sup> perspective**.

Any on the fly added functionalities were meant to integrate with the existing ones or to partly modify them in order to provide the expected behavior. Unfortunately this has also the potential to inject malfunctions in previously working logic.

Many aspects of the CMWA were refactored based on the observations made during the testing phase and concluded mostly in modifying the UI to either offer a **better user experience** by making the UI more intuitive and improving the accessibility to the most needed features or to **redesign the business flow** in order to prevent the user drop in inconsistency data states.

### The implementation phase

In certain cases (e.g., when the sales-bonus feature was too generic described) we created a **high-level prototype** to demonstrate the most critical aspects of the feature. This way, the customer can visualize the future system and make adjustments before it is fully implemented.

One important goal was to have as much as possible of the data-related logic on T-SQL level. This rule allowed us to gain benefits on both performance and security (two pillars of software development). Over 90 stored procedures were written and optimized to push into upper layers no more data than it's really needed being display-ready (with no or minimum post-processing required).

### The improve phase

We apply an iterative, agile approach to development because this ensures **continuous communication** with the customer, repeated reviews, and frequent deliveries. During iterations the existing implementation is thoroughly (re)analyzed and compared with the project's goals.

The overall performance has been constantly improved using **Ajax and caching mechanisms** for UI<sup>4</sup> rendering and T-SQL optimizations which reduced the model **querying time down to 2 seconds** (with 15 querying parameters used).

---

<sup>3</sup> User Experience

<sup>4</sup> User Interface

## Why prototyping?

We think the keys to handle feature prototyping are:

- Knowing what you want to learn from the prototype
- Access to rapid prototyping techniques
- End-user involvement in development of the prototype

The main prototyping benefits are:

- Creating a more end-user-oriented product faster
- The customer is able to 'feel' the product earlier
- The product can be adjusted before refactoring becomes expensive

## Innovation

The most noticeable and creative contribution of our team:

- Extendable bulk editing mechanism for updating characteristics of multiple car models at once. It controls both the UI field rendering and the underlying data. Out-of-the-box support for any new characteristic

## The User Acceptance Test (UAT) phase

As a product's design evolves, usability testing can help to ensure that its UI enables users to realize the benefits of the product. Usability testing is always iterative checks that any design changes your team has made in response to prior tests have successfully resolved the issues.

At the end of development iteration, the User Acceptance Test is scheduled. This is the moment when our latest released features are **tested and validated** by the customer and **helpful feedback is gathered**. This helps us tailor the next development iteration and improve the quality of the final product.

## Result



Beyond the many hours spent improving the initial concept and the development effort, what really matters at the end are the results. Some of the **customer benefits we aimed from the start** and achieved during the process are:

- Over 42,000 items are now managed in a practical and intuitive way. Collaborative work is sustained by workflows, scheduled jobs and distributed access.
- Clear, flexible and responsive UI offering a short learning curve and an exhaustive yet instant feedback from the underlying business logic. Less than 2 seconds to commit UI actions to data layer.
- Complete integration with other software systems, sharing, producing and consuming data along same company processes.
- Professional quality accompanied with a very competitive cost.

## Customer benefits

Some of the benefits we achieved for the customer:

- Scalable but yet performance architecture
- Collaborative working environment
- Clean, flexible and responsive UI
- Short learning curve for the end-user
- Complete integration with other software systems
- Professional quality services at very competitive costs
- A long term partner in our team

## About us

Skyce Software Solutions is a young but dedicated team able to implement a wide range of sophisticated projects at highest possible level.

- Location: Craiova-Romania (Eastern Europe)
- Established: 2010
- Technology: Microsoft .NET based
- Contact: [office@skyce.ro](mailto:office@skyce.ro)