

Service delivery management application / Fortune 500 Co.

A Java-based Enterprise Web Application with its own open source frameworks. It enables end-to-end management of service delivery for a Fortune 500 company.

Overview _

Client:

Hybrid cloud data services company
> 10000 employees
USA

Business case:

- + Resource management
- + Revenue recognition
- + Handling of field support

Industry _

- + IT Services
- + Data storage
- + Cloud

Services _

- + Software outsourcing
- + Team augmentation

Project type _

- + Web

Technology _

- | | |
|--|----------------------|
| + JEE | + JQuery |
| + Groovy | + JQuery Mobile |
| + Oracle DB | + HTML, CSS |
| + Microsoft SQL Server | + Velocity templates |
| + SAP Hana | + Weblogic |
| + Hibernate | + Maven |
| + ETL | + Jenkins |
| + Infinispan | + Sonar |
| + JGroups | + Nexus |
| + Microsoft EWS | + GitLab |
| + Custom OSS frameworks for web dev (jWic, appkit) | + Zabbix |
| | + Grafana |
| | + Selenium |
| | + RallyDev |
| | + ServiceNow |

Description _

The Enterprise Web Application empowers end-to-end service delivery management application through several key features, such as:

- + opportunity assessment,
- + project management,
- + resource management (resource dispatch, resource skills, logistics, 3rd party service providers),
- + status tracking,
- + revenue recognition,
- + time entry,
- + invoicing,
- + financial information tracking,
- + field support handling

On top of these, it integrates into the corporate applications ecosystem and enables business process optimization and workflow automation (until now, both were not possible with other enterprise software solutions).

Challenges _

- + Business complexity and application size (over 750.000 lines of code).
- + Dynamic and fast paced business environment with changing needs and requirements.
- + Taking over responsibilities from enterprise legacy systems.
- + Complex integration with enterprise applications (SAP, Oracle ERP, MS Exchange, Business Intelligence systems, Enterprise data warehouses) and other external systems (Office365, Google Maps, SMS providers).
- + Need for high availability and scalability.
- + Large user base (thousands of users) across all continents.
- + Need for support with low SLAs, data security, legal requirements, corporate standards compliance.
- + Very high quality standards because of application's broad reach and visibility.

Solutions _

- + Modular architecture: it enabled us to tackle the complexity, with a focus on easy maintenance through constant refactoring and code clean-ups.
- + Agile development process: it allowed us to adapt quickly to business needs. An "Open to change" philosophy was enabled by loose module and component couplings, multiple abstraction layers, dependency injection and defined extension points.
- + Strong Business Analysis team: they worked closely with field SMEs to document, improve and execute the transition from legacy systems to our application.
- + Strong DevOps team: they worked closely with other application groups and designed solutions for integrating the application into the corporate ecosystem.
- + Clustering and Load Balancing: enabled high availability and scalability, together with the ability to have non-disruptive updates.
- + Customer support team & strict policies to deal with data protection and legal requirements. The implementation of these policies was periodically audited by our team and customer representatives.
- + Continuous integration system & multi-step QA (incl. automated regression testing, manual testing, business validation, multiple testing- and staging-environments): allowed us to meet and exceed client's quality standards.