infopulse

Case for **IT Services company**

Decomposition of the Batch Grid system of a Core Banking Solution_

Industry: Software & Hi-Tech, Banking

Location: Nordics

Employees: 20 000+ FTEs

Client Background

One of the largest IT services companies and a multinational supplier of computer software, IT services, and innovative solutions to enterprise customers across the EU and worldwide.

Business Challenge

Our reliable client, a large IT services company, reached Infopulse to assist with the modernization of their customer's core banking solution. The financial services company wanted to enhance the effectiveness, maintainability, and security of their Batch Grid system. As the project was large-scale and long-term, different internal teams were involved in the solution development. The system itself lacked consistency and generated a range of dependencies between other systems responsible for regular batch processes (e.g., End of Day).

As a result, the customer had to deal with the following challenges:

- An increase in maintenance issues of the core banking solution.
- Releases of new features were complicated due to multiple development branches that had to be merged and tested individually before each release.
- The solution structure was prone to defects, with over ten systems responsible for the Core Batch Grid (CBG) system code. The tight coupling in the system made it difficult to handle versioning and feature toggling.

Solution

Upon conducting the detailed analysis, Infopulse experts came up with solutions that could address the discovered system issues and validated their effectiveness at the PoC stage:

- Iterative splitting would allow a sequential separation of systems, thus safeguarding from system failures.
- Integration of standalone system tests into the build pipeline would improve the work quality and speed up a feedback loop for all development teams.
- A transactional framework would help separate business logic from a technology platform.
- An enhanced development workflow would enable the distribution of duties between development teams, thus increasing their autonomy, reducing conflicts, and improving focus on functionality.
- A performance validation test suite would allow for faster and more accurate performance testing.
- The introduction of common architecture development standards helps decrease and prevent technical debt.

When we implemented and deployed all the above solutions to production, we turned a tightly coupled monolith application into a platform. This crucial change resulted in the following:

- Each subsystem codebase was reduced to business logic that related only to the subsystem data.
- Subsystems were placed into separate Git repositories.
- By introducing APIs integration, we implemented contracts between subsystems, allowing developers to change integrations via consensus, provided by pull request functionality.
- Shifting to the weak code ownership allowed individual teams to work independently on end-to-end functionality and control their workflow.

Technologies

Java 8/11

A coding language

Hazelcast

An in-memory data grid

Spring Framework

A Java application platform

Flyway

A database migration tool

Oracle Database

A persistence storage

jOOQ

A light database-mapping library and typesafe SQL querying in Java

Narayana

A transactional manager for JTA

IBM MQ

A Java message service

Prometheus

A monitoring tool

Logstash

A logging tool

Logback

A logging tool

Automated testing tools

ArchUnit

JUnit

Mockito

Cucumber

DbUnit

Development and CI deployment tools

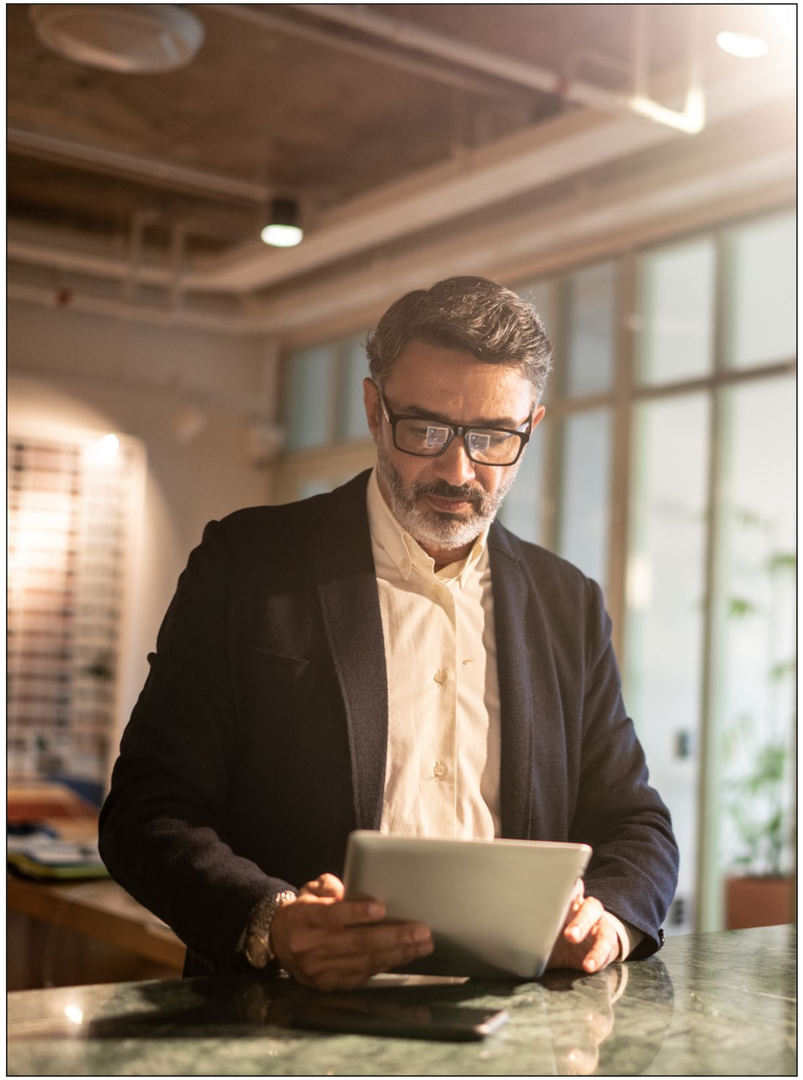
Jenkins

Docker

Maven

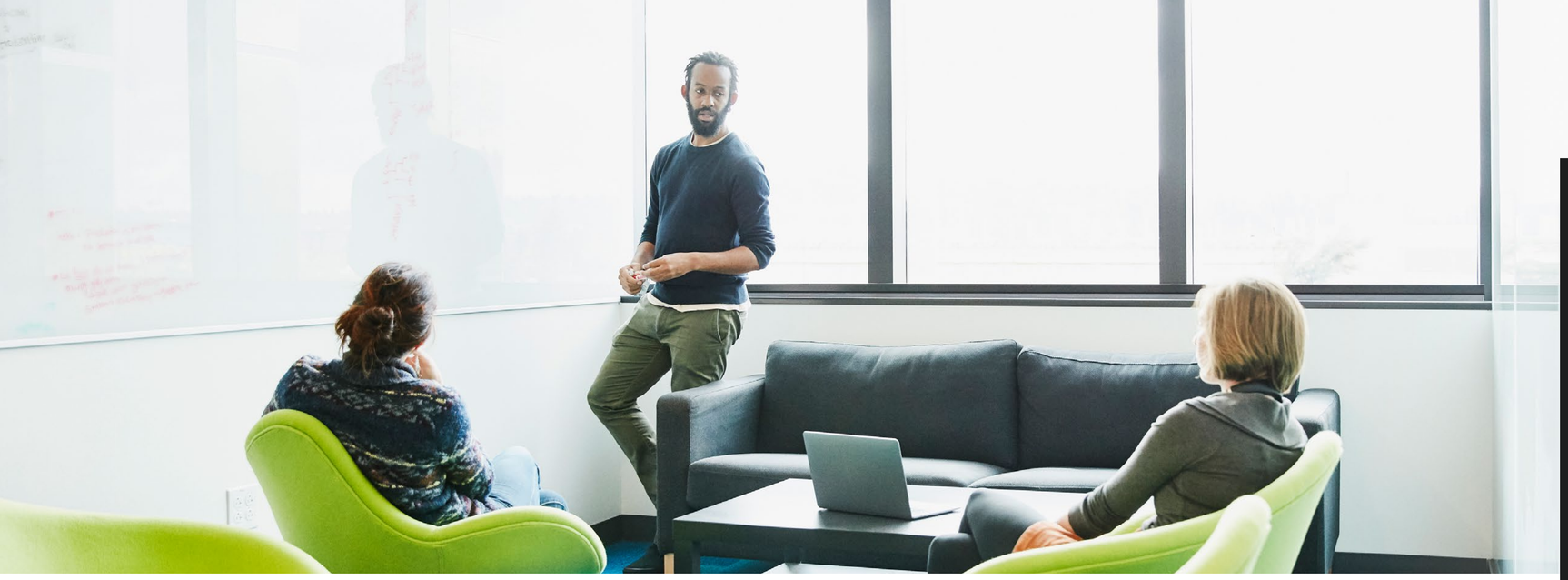
Python

Git (Bitbucket)



Business Value

- Business domain functionality was distributed into subsystems, thus providing independent maintenance.
- Modifications in one business logic component will not affect others, which lowers the number of defects, makes them easier to detect, and greatly improves the defect resolution time.
- The code duplicates were eliminated by introducing the abstraction layer between a business logic layer and a persistence layer, thus simplifying the product development process.
- Faster time to market and higher quality of new products.
- Due to the distribution of systems across separate repositories, the build time was significantly reduced, enabling faster feedback on changes and decreased costs on changes implementation.
- Common development standards established a unified framework for software developers. The overall efficiency of the development team greatly improved.
- The weak code ownership allows teams to decide how to implement the framework on their level, enabling technologies and process improvements.
- An employee engagement level increased as per development managers.



About Infopulse

Infopulse, part of the leading Nordic digital services company Tietoevry, is an international vendor of services in the areas of Software R&D, Application Management, Cloud & IT Operations, and Cybersecurity to SMEs and Fortune 100 companies across the globe. Founded in 1991, the company has a team of over 2,300 professionals and is represented in 7 countries across Europe and the Americas.

Infopulse is trusted by many established brands, such as BICS, Bosch, British American Tobacco, Credit Agricole, Delta Wilmar, ING Bank, Microsoft, Offshore Norge, OLX Group, OTP Bank, SAP, UkrSibbank BNP Paribas Group, Vodafone, Zeppelin Group, and others.

For more information, please visit **www.infopulse.com**

Contact us

PL +48 (221) 032-442

DE +49 (69) 505-060-4719

US +1 (888) 339-75-56

UK +44 (8455) 280-080

UA +38 (044) 585-25-00

BG +359 (876) 92-30-90

BR +55 (21) 99298-3389

info@infopulse.com