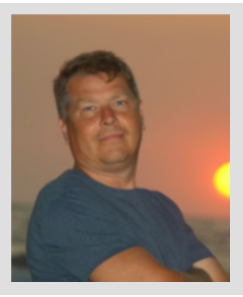

Valentin VĂLCIU

Sr. Software Developer/
System Architect

https://www.linkedin.com/in/axiac

https://stackoverflow.com/users/4265352/axiac

https://github.com/axiac

https://axiac.ro

Summary

Software development revolves around problem-solving. I enjoy identifying challenges and discovering effective solutions.

Programming languages are not important, and frameworks are even less so; they come into vogue, shine, and then fade away. What truly matters is the application architecture and design. These elements determine an application's longevity, its adaptability to changes, and the speed at which it can be developed or modified when necessary.

A thorough understanding of the business integrated into the software product is more important than the choice of language or framework and should guide the decisions.

* **Programming languages:** I am familiar with about a dozen, to varying extents, and there are a few more that I learned but have since forgotten (or became out of fashion). I can learn a new language more quickly than it takes to get familiar with the business.
  In alphabetical order: C, C++, CSS, HTML, JavaScript, jq, PHP, Ruby, SQL, TypeScript, Unix shell script (Bash), etc.
* **Favourite language:** Ruby
* **Languages that I want to learn:** APL, Rust
* **Frameworks:** I don't know any. I can learn a framework if necessary (it happened in the past), but specialising in or being confined to one isn't a good approach.

Areas of expertise

- Software Development
- Software Architecture and Design
- Code Quality
- Domain Driven Design
- Clean Code

Industries

- Computer Software
- E-commerce

Work experience

Cognizant Softvision

2016-present

Sr. Software Developer

Develop software based on the customer's needs, create architectures, designs, diagrams, documentations & workflows, train junior developers, evaluate candidates.

Int'l Outsourcing Group

2010-2016

Sr. Software Developer

Been involved in maintaining and enhancing a legacy e-commerce website. Developed new features and integrations with external systems. Designed and developed a new internal system to collect, normalise and centralise the information about chargebacks.

Lunatech

2002-2010

Software Developer

Developed web games, desktop games for Windows, Nintendo DS games, dynamic websites, newsletters.

Phenomedia Romania

1996-2002

Software Developer

Developed Windows applications, static web sites, web games and desktop games for Windows.

Education

University of Bucharest, Romania
Faculty of Mathematics
1991 - 1996

- BSc in Computer Science

Languages

Romanian	●●●●●
English	●●●●○
French	●○○○○

Skills

Problem Solving	●●●●●
Critical Thinking	●●●●●
Adaptability	●●●●●
Communication	●●●●○
Team Work	●●●●○

Outside of work

* Contribute to OSS - fixed a long-standing bug in Standard PHP Library, added a new feature to Homebrew, fixed bugs and improved features in Backstage, etc.
* Write code to solve puzzles on **Advent of Code**
* Solve math puzzles on **https://www.mscroggs.co.uk**
* Solve puzzles of various kinds and cross-word puzzles.
* Ranked in top 3 on several company internal programming contests (2016-2022).
* Read a lot of technical articles about software development.
* Stay up to date with the latest changes and improvements of the languages, frameworks and technologies that I use.

Outstanding Technical Achievements in OSS contributions

* Fixed a long-standing bug in Standard PHP Library — https://github.com/php/php-src/pull/1880
* Implemented Homebrew command `brew outdated` for casks — https://github.com/Homebrew/brew/pull/2309
* Integrated repository phpunit-mock-objects back into phpunit keeping its history intact, as if they weren't ever split — https://github.com/sebastianbergmann/phpunit/issues/3103#issuecomment-387645672

Selected relevant project experience

Web application for a medical company

2024-now

The project is a web application to be deployed in hundreds of medical offices worldwide. It is built as a new implementation of an existing application, incorporating a new UI design, improved backend, modern technologies and enhanced workflows.

My responsibilities as member of the team

* Examine the behaviour and implementation of specific features from the previous application version, document the findings, and design software components and workflows to implement them in the new application.
* Implement new features, fix bugs.
* Review other developers' code, providing suggestions for improvement and alignment with project rules and practices.

Self-assigned responsibilities

* Teach developers how to write clean, maintainable code.
* Mentor junior developers, helping them learn best practices in software development.
* Organise a weekly session to discuss technical challenges related to the project and deliver presentations on technologies and methodologies aimed at enhancing developers' knowledge and improving code quality.

Technologies: TypeScript, Node.js, React, Testing Library, MongoDb, Kafka, micro-services, micro-frontends.

Internal platform for developers

2022-2024

The project is an in-house platform designed for the developers.

The initial team's assignment was to build a POC using the Backstage framework, set up test and production environments in Cloud Foundry, create CI/CD pipelines to support development, build and deploy the code in these environments and document the process.

After the POC, the team's activity included:

* Onboard five feature teams to evolve the POC into an MVP; train them on the knowledge and processes established during the POC.
* Set up new test, pre-production, and production environments in AWS (Cloud Foundry proved to not be suitable for the MVP and the final product).
* Create and maintain CI/CD pipelines in Azure DevOps for development and deployment in the new test, pre-production, and production environments.
* Establish guidelines to enable dozens of independent teams to develop features seamlessly; document and enforce these guidelines (when possible) in tools and configurations.
* Implement customer-specific processes and document their implementation and usage.
* Document the project structure, code, configuration and workflows.
* Develop libraries to be used by feature teams and document these libraries.
* Train developers of the new feature teams on the established coding practices and workflows and how to use the Backstage functionalities correctly.
* Review other developers' code, providing suggestions for improvement and alignment with project rules and practices.
* Coordinate development activities where feature teams' work overlaps.
* Host bi-weekly meetings to share updates on development progress, issues, challenges, upgrades, and plans with feature teams.
* Compose and distribute a monthly newsletter to keep everyone in the area informed about the product's progress and near-future plans.
* Mentor junior developers, helping them learn best practices in software development.

Technologies: **Backstage**, TypeScript, Node.js, **Express**, React, Testing Library, shell scripting (Bash), **jq**, PostgreSQL, AWS, Docker, Azure DevOps pipelines.

E-Commerce Platform

2018-2023

The project is an e-commerce cloud platform that supports several hundred websites and various other business units, including mobile and web applications. The backend is built with microservices in AWS, using Java and TypeScript on Node.js. It passed through several Black Friday/Cyber Monday shopping seasons without significant issues.

Worked simultaneously with two teams. One team (named “Foundation”) developed the Node.js framework that powers all Node.js microservices, while the other team (named “Communications”) built the subsystem for sending notifications to customers via email, SMS, and other channels.

My contribution to this project

* Design, develop, and maintain a Node.js framework that powers over 45 microservices written in TypeScript for Node.js (as a member of team Foundation).
* Document the framework, train developers on its usage, and provide support for any issues (team Foundation).
* Design, develop, and maintain a scaffolder microservice that allows developers to create new microservices through cloning, enabling them to create and personalise a new functional microservice (without any specific behaviour) in about 15 minutes (team Foundation).
* Design and implement tools used in CI/CD pipelines to build microservices and manage their semantic versioning; provide support for these tools and pipelines when issues arise (team Foundation).
* Design, implement, refactor, or contribute with code to several microservices developed by the Communications team.
* Design and build tools for the Communications team to create and maintain hundreds of templates (used for customer messages) in multiple languages for dozens of business units (websites).
* Host bi-weekly meetings to keep developers informed about framework updates, plans for future functionalities and improvements and to learn about their needs (team Foundation).
* Review the code of a dozen developers working on microservices using Node.js, TypeScript, and the internal framework, providing suggestions for improvements (team Foundation).
* Teach developers how to write clean, maintainable code.
* Mentor junior developers, helping them learn best practices in software development.

Technologies: TypeScript, Node.js, **Express**, **Handlebars**, Docker, shell scripting (Bash), **jq**, OpenAPI (aka “Swagger”), microservices, REST, AWS, API Gateway, SQS, SNS, SES, Redis, Elasticsearch, DynamoDB, Groovy

Outstanding Technical Achievements

I single-handedly designed, documented, implemented, and maintained a significant change in the code and input data source of the microservice that manages configurations for all business units. This microservice seamlessly handles 300 requests per second, and the entire platform (comprising over 200 websites and numerous background processes) relies on its proper functioning. Over 300 developers contributed daily to the data served by this microservice, with more than 20 PRs per day back in 2020 when the data was stored in a Git repository.

The change involved an incremental migration of the data source, new code to retrieve and integrate the data from both the old and the new data source ensuring that the output remained unchanged, and scripts and procedures to validate the data staged for migration, before committing it to the new data source.

During this critical phase, the microservice underwent a significant overhaul, with approximately 90% of its code replaced in several stages. Its data source was relocated twice, and the data update method was entirely revamped. **Despite these extensive changes, there was no downtime or noticeable issues for customers or developers.**

CRM products based on SugarCRM

2016-2018

CRMs implemented using SugarCRM for a handful of clients

* Configure SugarCRM to align with the client's needs and business-specific workflows.
* Design and implement frontend components for scenarios where the standard SugarCRM components do not meet the client's requirements.
* Design and implement backend components to handle the background processing specific to the client's workflows.

Technologies: PHP, **SugarCRM**, JavaScript, **Backbone.js**, various RDBMSes, shell scripting (Bash, Windows batch files), Vagrant

Outstanding Technical Achievements

Designed and implemented two PHP libraries to interface with a private client's custom system which relied on **DIME**, an outdated communication protocol abandoned 13 years earlier.

Without having access to a live system for testing, I used TDD to design, implement, and test the code. **Upon delivery, it worked flawlessly in production on the first attempt, with no bugs ever reported.**

Social networking website/ E-commerce

2010-2016

Social network website

* Maintain and enhance a legacy PHP website with millions of users (a specialised social network website).
* Implement new backend functionalities and update the frontend page layout.
* Design and implement the integration with a new internal tool for centralised email address processing.
* Design and develop a new internal system and workflows to collect, normalise and centralise the information about chargebacks and generate reports.

Technologies: PHP, MySQL, HTML, CSS, JavaScript, **Smarty**, Symfony components

Outstanding Technical Achievements

While investigating why a daily background workflow was taking over 24 hours to complete, I discovered that the delay was due to several slow SQL queries involving geographical coordinates and trigonometry for distance calculations. This was back in 2011, before MySQL supported spatial data types.

By shifting the computation to PHP, **the script's processing rate increased from approximately 1,000 records per hour to over 11,000 records per hour (more than 10 times faster)**. The speed of the modified code alone increased from 25-30 records per minute to 300-400 records per second (over 600 times faster).

Web games
Desktop games
Nintendo DS games
Websites
Windows applications

1996-2010

Implemented about a dozen of small games, a couple of desktop applications and several websites:

* Web games, in browser, using Macromedia Flash
* Windows games, using Macromedia Director, Blitz Basic, Blitz Max, C++ & DirectX
* Nintendo DS games, using C and C++
* Windows applications, using C++ and MFC
* Websites, using PHP, HTML, JavaScript and CSS

Technologies: PHP, HTML, CSS, JavaScript, **Smarty**, Macromedia Flash, Macromedia Director, Blitz Basic, NintendoDS, C, C++, MFC, Windows API, DAO, ADO, DirectX