Igor Malovitsa

Contacts

Email: igor.malovitsa@gmail.com

Skype: igor.malovitsa

Discord: @wizord.tv

About myself

I'm a software engineer with 13+ years of commercial experience in IT industry.

I prefer problem solving over blaming.

I like to work in teams and discuss design decisions.

I know math and not afraid to use it.

Skills

Programming languages: Rust, TypeScript, Python, C#, SQL, Perl;

Good experience with reverse engineering and understanding low-level systems;

Deep involvement in client communication;

Mentoring people in Rust programming & software engineering;

Conducting technical interviews for Software Engineers;

Solid knowledge of version control systems: git, svn, TFS;

Experience with frameworks: Node.JS, .NET, Django, bottle, Qt5;

Knowledge of basic *nix tools – bash, sed, grep, vim...;

Can read and apply academic papers for good use;

Fast learning and flexibility;

Languages

English (Upper-intermediate)

Russian, Ukranian (Native speaker)

Employment experience

2016-04 – 2024-09 Senior Software Developer, DataArt

2015-07 – 2015-09 Senior Software Developer, Cốc Cốc

2012-08 – 2015-03 Software Developer, GlobalLogic

2012-03 – 2012-08 Sysadmin

2010 – 2012 WebDev freelancing

Project highlights

Codec for ASTERIX protocol

https://github.com/freeasterix/freeasterix

Description: ASTERIX is a network protocol used by air radars to communicate information about surveillance.

Technologies used: Rust, Wireshark, Python

The protocol codec together with a test suite, documentation and python bindings was implemented in three weeks.

Responsibilities: All of them

Reverse-engineering a binary image format

Description: Mischief is a program for drawing, built on top of vector representation. Since the progam was discontinued, there is a need to understand the format.

The format implemented a custom compression algorithm based on LZMA+Arithmetic coding, and bit packing, which made reverse engineering a bit challenging.

Technologies used: OllyDbg, x86 assembly, and a steady hand

Implemented a GPS clock with ESP8266

Description: This was a pet project to dip my toes into embedded development.

This embedded project was entirely in Rust.

Implemented a custom async NMEA protocol parser.

Implemented a custom MAX7219 driver.

Experimented with MAX7219 representation and timings.

Custom blockchain using Substrate protocol

Description: The customer requested to implement a distributed ledger for Carbon offsets.

Technologies: Rust, Substrate, Node.JS, React.JS, RabbitMQ

Team size: 10 people

Responsibilities:

- Implemented a custom fungible token system with access control
- Implmented KYC regulated requirements on the blockchain
- Implemented a benchmarking tool for blockchain throughput
- Contributed to marketplace backend and UI implementation

Travel company front page and booking

Description: a well known travel company has a rich front page which requires continuous support and upgrades. Major tasks included implementing an OAuth2 integration for corporate clients, media-rich elements on the front page, analyzing loading times and speeding up the loading time.

Discovered and fixed a bug which increased customer retention by 3%.

Designed and implemented a button which increased app conversion by 15%.

Team size: 7 people

Technologies used: React, Next.js, Contentful, Phraseapp, Docker, Angular, SCSS, Node.JS, WebPack, OAuth2, Ruby, Rails

Responsibilities:

- communicating with the client
- designing interactive elements
- feature implementation
- code review
- writing unit tests

Deal Management System

Description: An equity investment company supports their own internal deal management system. Deal management is done in part via a set of rich web applications, which require continuous upgrades and support. Web apps complexity is ranging from simple CRUD interfaces to rich data visualisation and planning.

Team size: 8 people

Technologies used: C#, ASP.NET MVC, IIS, RavenDB, Node.JS, React, Webpack, bootstrap, Knockout.JS, Angular, HighChart

Responsibilities:

- communicating with the client
- designing web UI
- feature implementation
- debugging
- code review
- writing unit tests

Internal Browser Interfaces

Description: The goal of the project is to provide a rich, responsive and pleasant user experience in the Browser. Browser interfaces included a custom media player, download manager, settings, home page and more. Browser features included rich integration with media sites, acceletated browsing and downloads, custom input IME, and more. All of those features require a fine-tuned, responsive UI to attract and retain a large urerbase.

Technologies used: Node.JS, HTML5, React, Webpack

Responsibilities:

- communicating with the client
- designing UI
- feature implementation
- debugging
- performance optimization
- code review
- writing unit tests

Web interface for business logic simulation

Description: an equity firm has complex logic written entirely in XLS files, and the goal was to automate, scale and speed up the process of simulating business logic for profit optimization.

The project was split between SPA client written in React.JS and a C# server that processed business logic

Team size: 10 people

Technologies used: C#, ASP.NET MVC, IIS, Node.JS, React, Webpack, bootstrap

Responsibilities:

- UI team lead
- communicating with the client
- feature implementation
- debugging
- code review
- writing unit tests

Emergent Data Viewer

Description: The goal of the project was to implement a native crossplatform application to view detailed reports produced by a portable device and allow the user to route the data to different locations.

Project duration: 18 months; Team size: 6 people

Technologies used: C++, Qt5, SOAP

Responsibilities:

- feature implementation
- developing Qt5/QML UI
- debugging
- implementing scripts for CI
- writing unit tests
- code review
- writing design documents
- documenting project architecture

Backup and Recovery Web Console

Description: The goal of the project was to create a simple web-interface and API for a powerful backup solution. The application provided realtime data updates and transaction safety. Web interface had a good coverage of underlying backup features.

Project duration: 14 months; Team size: 8 people

Technologies used: Python, JavaScript, HTML5, SASS, ExtJS, jasmine, REST

Responsibilities:

- API design
- developing web UI
- developing API backend
- debugging
- writing unit test
- implementing scripts for CI
- writing deployment scripts
- code review
- communicating with clients

Media recording system

Description: The goal of the project was to record media from distributed sites. Application used a distributed storage system. The recording had to be done automatically on pre-set schedule. The application was controlled using a simple web interface.

Project duration: 8 months; Team size: 3 people

Technologies used: Python, Django, REST, JSON, JavaScript, jQuery, SASS, ffmpeg

Responsibilities:

- API design
- feature implementation
- product deployment
- developing web UI
- developing backend
- database migration
- communicating with clients

Education

Certified Google Cloud Developer (expired)

2020 From Nand to Tetris / Part I, Coursera

2016 Machine Learning, Coursera

2014 Cryptography I, Stanford University Online Course

2006 – 2012 V.N. Karazin Kharkiv National University, Physical and Technical Faculty, master’s degree (experimental nuclear and plasma physics).