

IREK JOZWIAK

HASKELL PROGRAMMER & INTERACTION DESIGNER

About me

BUSINESS-FOCUSED

My ultimate goal is to help you make money. I perceive programming as a means of achieving a bigger goal, not an activity on it's own. I'm not a guy who does the nine-to-five in his own little field.

AGILE

I use Agile Software Development to ensure that you can prove your business ideas as quickly as possible. I solve your problems. I try hard not to implement imagined features.

RELIABLE

I use the Haskell programming language and Test-Driven Development to make certain that your product will work—both in terms of producing correct results and fulfilling user expectations.

TRANSPARENT

When I work, I don't hide knowledge and I don't promote myself. I build teams. I don't care about political games. I'm into building great products.

Quick overview *(details on the following pages)*

SKILLS

Haskell, business analysis, functional programming, interaction design, Test-Driven Development, full cycle development, SQL, Java, LDAP, TeX, Agile Software Development, object-oriented programming, Ruby, Linux, shell, parsers, networking, architecture design, team building, pair programming, eXtreme Programming, SCRUM, Javascript, AJAX, jQuery, User Interface design, emacs, algorithms and data structures, logic, lambda calculus, design for simplicity, enterprise software, liaising between technical and non-technical staff, web applications, integration, deployment, unit testing, coaching developers, usability, Domain-Specific Languages, learning quickly.

EXPERIENCE

2010–	Webapp, Haskell, Javascript, Agile, analysis, architecture, TeX	Telecommunication	Munich, Germany
2009–2010	Webapp, Haskell, Javascript, analysis, TeX	Building industry	Poland
2005–2008	Webapp, Java, SQL, Javascript, Agile, analysis	Investment Banking	London, UK
2004–2005	Webapp, Java, SQL, Agile	Molecular Biology	Poland
2004–2005	Webapp, PHP, SQL, OOP, Agile	e-commerce	Poland
1999–2005	C, SQL, FreeBSD, Linux, TCP/IP	Telecommunication	Poland

Considering hiring me?

EMAIL

irek@humane-software.com

CALL

+48 606 915 011

CHECK

www.humane-software.com

SEE RECOMMENDATIONS

linkedin.com/in/irekjozwiak

Experience details

HUMANE SOFTWARE

Owner & Haskell Developer

May 2010–Present (3 years 4 months)*

Our customer, Global Access GmbH, is an internet connectivity and colocation provider. We have developed billing, traffic, monitoring and cloud solutions. We integrated well with the on-site staff and came to work together as one team. The work involved creating a simple enterprise architecture capable of running services that handle different concerns in a decoupled manner. This in turn allowed us to gradually refactor software written in PHP and Ruby towards a unified, Haskell-based architecture. The software is largely stateless and thus easy to scale if necessary. Acceptance tests ensure correct functioning. We initially wrote unit and system tests in HUnit, but are now moving towards QuickCheck. We also developed elaborate deployment tests, dictated by the need to deploy the system to over 20 machines that were inaccessible from outside (giving us no possibility to fix glitches).

Technologies used: Haskell, Javascript, (atto)parsec, enumerators, QuickCheck, HUnit, Ruby, SQLite3, TeamCity, nginx, Linux, SNMP, VMWare, automated virtualisation, emacs, eXtreme programming, pair programming.

FIRMUS

Haskell Developer & Interaction Designer

July 2008–July 2010 (2 years)

I developed a web-based system for price-modelling and invoicing. Thanks to a good relationship with the customer and small size of the company I was able to employ my ideas virtually without any restrictions. Some of these were quite experimental, yet successful:

- The pricing model was described by a business person using a simple DSL, which included testing facilities. The business code was written using TDD.
- Instead of multi-field, multi-screen forms, the user interface was based on big text areas (one per task) into which the user typed what he or she would normally write on paper. Instead of complex workflows, parsers interpreted this natural input and provided real-time feedback.
- The price and invoice were generated on the basis of natural language input, modeled on how non-technical users would communicate with each other. Following just a few simple rules would make the parser happy.
- Invoices were beautifully typeset using pure TeX.
- All this was achieved in approx. 4300 lines of Haskell, including tests.

Despite this sounding like a technical/idea overkill, the software was actually well-received by its users. The user interface differed from other contemporary applications, but it was very similar to what the users were used to, and this mattered to them more. It was also a great opportunity to test ideas that would have never been considered in a corporate setting.

Technologies used: Haskell, Parsec, TeX, Javascript, AJAX.

BNP PARIBAS INVESTMENT BANK

Agile Developer & Interaction Designer

July 2007–August 2008 (1 year 2 months)

I worked as a Java developer at the Fixed Income Bond Static Data Repository. Using MQ and SQL replication, the system obtained bond information from multiple systems worldwide (including Bloomberg), worked out the quality data, and merged it into a uniform format. The information was then populated to multiple front-office and settlements systems within the bank using XML over MQ. It was also presented via a web interface to employees in different departments, including traders. The system employed extensive test automation on the unit level, system level, as well as on the functional level, using FIT tests written by Business Analysts. The project was run using a mixture of Scrum and eXtreme Programming practices, including pair programming, Test-Driven Development, refactoring and continuous design and planning. I took part in the estimation, planning, design and development of the functionality. I also created stories for the user interface. I designed most of the user interface-related parts of the features, often creating prototypes to better understand the functionality and to maximise usability, readability and robustness. I have also acted as a Scrum Master.

Technologies used: Core Java, MQ, Hibernate, Oracle, WebWork, JUnit, JMock, JSUnit, Velocity testing, FIT, AJAX, CSS, D/HTML, DOJO, Tomcat, Javascript, design patterns, eXtreme Programming and Scrum, Ant, IntelliJ IDEA, JProfiler, Linux.

* As of August 2013

BNP PARIBAS INVESTMENT BANK

Agile Developer & Interaction Designer

October 2005–July 2007 (1 year 10 months)

I worked as a Java developer for an Operational Risk management system. It was a group-level project to be used by all business lines world-wide. It was a greenfield project that allowed the bank to comply with Basel II requirements. The system allowed analysts to manage operational risk based on historical and foreseen data, and was set to eventually provide capital limits for traders. I took part in all stages of the full development cycle, including meetings with the customer and gathering requirements, analysing and estimating requirements, creating user stories, developing working and maintainable code, and automating testing. I also acted as a release manager. The project used eXtreme Programming practices such as pair programming, continuous design and integration, refactoring and Test-Driven Development. I frequently designed user interface features to meet new requirements. I pro-actively implemented usability principles throughout the application. I created a strategy and infrastructure to migrate acceptance tests to Watij. I coached developers to familiarise them with eXtreme Programming practices.

Technologies used: Core Java, J2EE, Hibernate, Spring, JDBC, Oracle, WebWork, servlets, JUnit, DynaMock, JSUnit, Watij, FIT, AJAX, CSS, D/HTML, JSP, Velocity, DOJO, WebSphere, Javascript, Business Objects, design patterns, eXtreme Programming, Linux, Ant, IntelliJ IDEA.

POZNAN UNIVERSITY OF TECHNOLOGY

Analyst Developer

October 2004–June 2005 (9 months)

WebMobis is a platform that allows computational biologists to analyse proteins and visualise them. It stores computation results in SQL database. It is a web-based application built in accordance with enterprise software industry standards. It communicates with other platforms (using HTTP and e-mail), as well as with its own computational engine, to provide analytical data. As an analyst and a Java developer in the project, I created the architectural framework for the application. I took part in all stages of the development cycle, such as gathering and analysing requirements, developing working and maintainable code, automated testing and deployment. I coached developers to familiarise them with eXtreme Programming practices.

Technologies used: Core Java, Tomcat, PostgreSQL, JDO, Tapestry, servlets, JUnit, JMock, XML, sockets, Ant, Eclipse, UML, Linux, SQL, Object-Oriented Programming, design patterns, eXtreme Programming.

KOLPORTER INFO

Software Designer (hands-on)

February 2004–September 2004 (8 months)

I created a flexible e-commerce platform, which allowed the company to enter the online shopping market. I was a hands-on designer, taking part in all stages of the full development cycle, ie. gathering requirements, analysis, development of the code and automated testing. I also lead the team on the IT side. The application was made using Objected-Oriented Design, design patterns, tiers, etc. I was responsible for introducing the agile development process. I also lectured on SQL.

Technologies used: PHP5, Java techniques, PostgreSQL, UML, FreeBSD, SQL, PHPUnit, Object-Oriented Programming, design patterns, eXtreme Programming.

VARIOUS COMPANIES

Developer & Network Administrator

1999–January 2004 (4 years)

I have worked for various companies and organisations as a developer and network administrator. I developed various web-applications using SQL and low-level modules in C, including a kernel module for FreeBSD. I have administered large ISP networks based on both hardware routers as well as FreeBSD and Linux servers. I have created security policies for these networks and servers.

Technologies used: SQL, ANSI C, PHP, LDAP, FreeBSD, Linux, TCP/IP.