Resume

Oleg Sivokon

<2015-08-02 Sun>

1   Contact Information

Name:	Oleg Sivokon
Birth date:	23, 06, 1978
Cell phone:	+972 (0) 544-563314
Email:	olegsivokon@gmail.com
Address:	Florentin 50, Tel Aviv, Israel

You can find an up-to-date electronic copy of this document at
`https://github.com/wvxvw/resume`

2 Technology Summary

2.1 Languages

ActionScript considerably above average.

C basic familiarity.

Clojure average.

Common Lisp solid average.

Emacs Lisp above average.

Erlang somewhat above basics.

Forth basic familiarity.

Haskell somewhat above basics.

Haxe solid average.

Java somewhat above basics.

JavaScript considerably above average.

Mercury somewhat above basics.

OCaml somewhat above basics.

PHP somewhat above basics.

Prolog somewhat above basics.

Python average.

Ruby basic familiarity.

Scala basic familiarity.

Scheme somewhat above basics.

Shell average.

> *Note that the bar is at different heights in different languages. Thus, for example, I know more about Common Lisp, than I know about JavaScript, but Common Lisp is a bigger language, which attracts more skilled professionals than JavaScript does.*

My estimates may be too shy due to me comparing myself to the data available online, however, my experience is that programmers with less than average knowledge of their trade are underrepresented in such data.

Hopefully, the breadth of my knowledge can compensate for the lack of experience in any particular technology to a certain degree.

2.2 Mathematics And Statistics

J basic familiarity.

Julia basic familiarity.

Maxima basic familiarity.

Octave (Matlab) basic familiarity.

R (S) somewhat above basics.

2.3 Related Technologies

Data representation formats (XML, YAML, JSON, RDF, AMF) above average.

Graph databases significantly above basics.

Network protocols (TCP, UDP, HTTP, FTP) top of the stack - solid average. I have basic knowledge of the protocols below transport level.

Relational databases (SQL) somewhat above basics.

Web stack (HTML, CSS) average.

2.4 Environments

Eclipse solid average.

Emacs considerably above average.

Intellij IDEA average.

Vim basic familiarity.

Visual Studio average.

2.5 Operating Systems

Android basic familiarity.

GNU/Linux solid average.

Mac OS somewhat above basics.

Microsoft Windows average.

2.6 Infrastructure

Build and automation tools Ant, Maven, Gradle, Make, Grunt, Jenkins.

Project management JIRA, Redmine, Track, Mediawiki, Bugzilla.

Testing frameworks Selenium, Flex Unit, JUnit, unittest (Python), 5am, Cucumber.

Version control systems Git (above average), Subversion (average), Mercurial (basic familiarity), Perforce (basic familiarity), Bazaar (basic familiarity).

2.7 Theory

(Finite) Model theory basic understanding.

Complexity theory basic understanding.

Computability theory basic understanding.

Distributed computation (join-calculus, π-calculus) basic understanding.

Objects (description logic, ζ-calculus) solid understanding.

Program semantics average understanding.

Type theory (typed λ-calculus, intuitionistic logic) average understanding.

These estimates relate to theoretical computer science researchers rather than average programmers. I would score higher in my own valuation, if I compared myself to average programmer.

3 Work History

3.1 PowToon Ltd.

Position *Programmer*

Term *January 2014–May 2015*

Accomplishments

- I completely rewrote the major part of the application responsible for generating presentation content from description, making it more efficient in terms of network utilization, making it available to more platforms and improving the overall coding standards.

- I invested heavily into project infrastructure by developing automated builds for a number of sub-projects, setting up and maintaining Jenkins continuous integration server. Reorganizing interaction with version control system and creating a number of automated tests in cooperation with quality assurance department.

- I wrote several in-depth proposals for general improvements in the program I worked on, some of which were later implemented and resulted in improved program performance and customers' experience.

3.2 TransGaming Inc.

Position *Programmer*

Term *April 2013–2013 (less than a year)*

Accomplishments

- I wrote a game portal for "Smart TV" deployment.

- I wrote a tester program, which stress-tested the program performance on "Smart TV".

- I wrote automated build for my project.

- I wrote a sizable chunk of code dealing with usage statistic aggregation using Google Analytics.

3.3 Rounds

Position *Programmer*

Term *December 2010 – August 2011 (9 months)*

Accomplishments

- I rewrote the communication part of the video chat application paying great attention to stability, fault tolerance and error reporting which allowed for better control of application performance and better customer experience.
- In order to accomplish the above, I introduced testing methodologies to the company, I also wrote a testing server, which mocked the application life cycle and usage patterns.
- I also wrote a small natural language processing program which was used to advertise the company's product.

4 Courses

I took several online courses from Coursera, Udacity and edX, which required me to complete programming assignments in Python.

I took a complete series of courses in statistics at edX (three courses), the data scientist toolbox course at Coursera (six short courses), machine learning at Udacity, introduction to algorithms (two courses) at Coursera.

I also volunteered for SCons project, writing an ActionScript plugin for it. (SCons is an extensible build system written in Python).

Two years ago I interviewed for a position of Python programmer at Walla. The company requested that I write a small test project, which I did. The project, although somewhat outdated, is still available for viewing: `https://github.com/wvxvw/intj`.

At the moment, I'm working on two of my pet projects: an intelligent IRC bot, capable of searching the chat history and answering questions about it. This project uses SQLAlchemy to connect to Postgres SQL database as well as Sphinx full text indexing and search engine in order to accomplish its goals.

My other pet project is an intelligent indexing and search for Emacs. It uses Common Lisp to interface with indexing databases (Sphinx at the moment, and potentially Lucene and Swish-e).

I am currently studying in Open University of Israel, mathematics faculty, about to finish my second year.

5 Hobbies

- NLP
- Computational linguistics
- Moral philosophy
- General AI
- Graphs