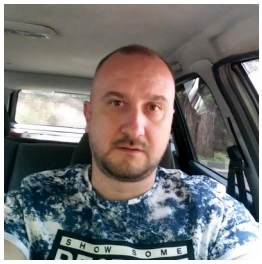
Resume

Python/FullStack Software Engineer, Software/Solution Architect



Contacts:
Ivan Zakrevsky, 1976 y.b.
+38068 400 64 28
Skype: industrialnet
E-mail: emacsway@gmail.com
Www: https://emacsway.github.io/about/
Blog: https://emacsway.github.io/
Open Source (primary): https://bitbucket.org/emacsway
Open Source (secondary): https://github.com/emacsway

About me:
An XP and DDD expert.

My main interest is to make the code clean, simple, reliable, readable, modifiable and maintainable; the implementation of new features rapid and cheap; the curve of cost of code change asymptotic (instead of exponential); the team skilled, successful, and having high self-motivation; the experience shared; the risks low.

My approaches are XP, TDD, DDD, SOLID, Design Patterns, Refactoring, Designing Through Refactoring, Clean Code & Clean Architecture, and others.

Also I'm an Open Source developer and an adherent of KISS principle. My philosophy is: "Do it shortly, clearly".

My trend is:

> “I like my code to be elegant and efficient. The logic should be straightforward to make it hard for bugs to hide, the dependencies minimal to ease maintenance, error handling complete according to an articulated strategy, and performance close to optimal so as not to tempt people to make the code messy with unprincipled optimizations. Clean code does one thing well.” (Bjarne Stroustrup, inventor of C++ and author of The C++ Programming Language.)

Education:
1994-1999 - The Dnepropetrovsk State Technical University of Railway Transportation

Open Source Projects:
Python:
* "rope" (contributor) - a python refactoring library. https://github.com/python-rope/rope
* "SQLBuilder" (owner) - a lightweight Python SQLBuilder. Can be integrated with Django.
https://bitbucket.org/emacsway/sqlbuilder
* "cache-tagging" (owner). Cache-tagging allows you easily invalidate all cache records tagged with a given tag(s). The package supports Django.
https://bitbucket.org/emacsway/cache-dependencies
JavaScript:
* Store.js (lead contributor) - a super lightweight implementation of Repository pattern for relational data and aggregates. The library allows you to use Domain-Driven Design (DDD) on client-side as well as reactive programming. https://github.com/emacsway/store (canonical repo: https://github.com/joor/store-js-external).

Articles:
Articles in English:
* "Implementation of Repository pattern for browser’s JavaScript" https://emacsway.github.io/en/javascript-and-repository-pattern/
* "About my experience of using Django Framework" https://emacsway.github.io/en/django-framework/
* "Design of Service Layer" https://emacsway.github.io/en/service-layer/
* "About problems of cache invalidation. Cache tagging." https://emacsway.github.io/en/cache-dependencies/
* "How to quickly develop high-quality code. Team work." https://emacsway.github.io/en/how-to-quickly-develop-high-quality-code/
* "Why I prefer Storm ORM for Python" https://emacsway.github.io/en/storm-orm/
Articles in Russian:
* "Реализация паттерна Repository в браузерном JavaScript" https://emacsway.github.io/ru/javascript-and-repository-pattern/
* "О моем опыте использования Django Framework" https://emacsway.github.io/ru/django-framework/
* "Проектирование Сервисного Слоя" https://emacsway.github.io/ru/service-layer/
* "О проблемах инвалидации кэша. Тегирование кэша." https://emacsway.github.io/ru/cache-dependencies/
* "Почему я выбираю Storm ORM для Python" https://emacsway.github.io/ru/storm-orm/

Bibliography:
Architecture & Design:
* "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides
* "Pattern Hatching: Design Patterns Applied" by John Vlissides
* "Patterns of Enterprise Application Architecture" by Martin Fowler, David Rice, Matthew Foemmel, Edward Hieatt, Robert Mee, Randy Stafford
* "Refactoring: Improving the Design of Existing Code" by Martin Fowler, Kent Beck, John Brant, William Opdyke, Don Roberts
* "Clean Code: A Handbook of Agile Software Craftsmanship" by Robert C. Martin
* "Code Complete" by Steve McConnell
* "Domain-Driven Design" by Eric Evans
Programming languages:
* "Learning Python" by Mark Lutz
* "Python Tutorial" and "Python HOWTOs" (official)
* "JavaScript: The Definitive Guide" by David Flanagan
* "JavaScript Patterns" by Stoyan Stefanov
Databases:
* "PostgreSQL" by Korry Douglas, Susan Douglas
Algorithms:
* "Algorithms and Data Structures" by N.Wirth
Compilers & Parsers:
* "Compiler Construction" by N.Wirth
Methodologies:
* "Extreme Programming Explained" by Kent Beck
Operational system:
* "Unix and Linux System Administration Handbook" by Evi Nemeth, Garth Snyder, Trent R. Hein, Ben Whaley

* "The Linux® Kernel Primer: A Top-Down Approach for x86 and PowerPC Architectures" by Claudia Salzberg Rodriguez, Gordon Fischer, Steven Smolski
* "Digital computers and microprocessors" by Aliyev / "Цифровая вычислительная техника и микропроцессоры" М.М.Алиев

English:

* "Friendly meetings with the English language" by Maria A. Kolpakchi
* "English. Reference materials." by Ksenia A. Guzeeva, Tamara G. Troshko

Technologies & approaches:

* Programming languages: Python (primary, since 2008), JavaScript (strong), Ruby (sometimes), PHP (in the past).
* Development environment: Emacs (sometimes VIM, PyCharm).
* Approaches: XP, TDD, DDD, SOLID, OOP, OOD, Design Patterns, DIP, Refactoring, Designing Through Refactoring, Clean Code & Clean Architecture, Software Architectural Design, Database Architecture, Low Coupling & High Cohesion, SRP, OCP, GRASP etc.
* DB & storages: MySQL, Spatial, PostgreSQL, PostGIS, tsearch2, MongoDB, Redis, Memcached, replication, sharding, analysing, SQL optimization.
* Frameworks: Django (mainly), Flask, also used Tornado, Twisted, Ruby on Rails, etc.
* JavaScript Frameworks: Dojo, Dojo2, jQuery, AngularJS, Angular2-5.
* OS: Debian (server), ArchLinux (desktop, server), Ubuntu (desktop, server), Fedora (desktop, server).
* Bug-tracking: Jira (mainly), Trac, Redmine, Pivotal Tracker.
* HTTP servers: Nginx (mainly), Apache2, Lighttpd, Unicorn.
* CVS: Mercurial (mainly), GIT, SVN, Bazar.
* Others: Eventlet, GDB, Celery, geographic information system (GIS), Storm ORM, multiprocessing, threading, queue, async, Solr, morphology, Pinax, business-transactions, high-load, high level concurrency, Remote Facade, DTO, cache invalidation based on tagging, automatic text processing, ACL, social networks integration.
* Also, I've a good (5+ years) experience in PHP, Zend Framework, Drupal, but it, mainly, in the past (2004-2009).

Experience:

* Nov 2016 – Jul 2017 – Senior Full Stack Software Engineer (remote) at jooraccess.com, the global wholesale marketplace where the biggest brands and best retailers do business online. Introduced TDD (and some other practices of XP), DDD, Designing Through Refactoring, code review based on catalog of refactorings and catalogs of code smells, continuous review, monitoring with New Relic. Solved a lot of issues of architecture and design. Improved quality of the codebase. Reduced time of implementation of new features due to the introduced design approaches. Performed some duties of Software/Solution Architect. Created the high-level library Store.js (https://github.com/emacsway/store) to handle data on client-side. Used Python, JavaScript, PHP, Bash, Django, PostgreSQL, jQuery, AngularJS, Dojo, RESTful API, CORS, JWT, Django-rest-framework, CakePHP, Redis, Celery, Pl/SQL, AWS, Ubuntu, Scrum, etc.
* Feb 2013 – Jul 2016 – Senior Backend Developer (remote) at rebelmouse.com, the best online CMS for social websites. Actively used Designing Through Refactoring, TDD, code review based on catalog of refactorings and catalogs of code smells. Debugging on production and low-level debugging with GDB. Backend/SQL high performance optimization. Monitoring with New Relic. Implemented social networks integration, polymorphic relations, ACL, statistics, data analytics. Used Python, Bash, Django, MySQL, MongoDB, RESTful API, Redis, Celery, Memcached, Eventlet, Ubuntu, replication, clustering, document-oriented data storage, kanban, etc.
* Aug 2011 – Mar 2013 – Senior Full Stack Software Engineer (remote) at tripster.ru, the Russian travel portal and FAQ for independent travelers. Used Python, JavaScript, Bash, Django, GIS, jQuery, MySQL, RESTful API, Solr, Nginx, Memcached, Debian, sub-processing, threading, Queue, Async, 3-d SQL Builders, morphology, fast compiled template engine wheezy.template, automatic text processing, etc. Implemented a lot of basic features, social networks integration, HTML5 API History for legacy browsers, geonames services, partner's API, advanced ACL, flexible user notification, variouse Ajax UI widgets, cache dependencies, phased cache (with nocache fragments), cache cleaning synchronization with DB transactions. Backend/SQL high performance optimization.
* Jul 2010 – Jun 2011 – Team Leader at Soft-Ukraine. Developed an online corporative brainstorm system. Used Python, JavaScript, Bash, Django, Dojo, jQuery, PostgreSQL, RESTful API, MVVM, dojox.data.JsonRestStore, Ajax UI, Ruby On Rails, Pinax, Debian, Multilingual, etc. System Administration.
* Oct 2009 – Jul 2010 – Co-founder and CTO at Dstudio. Organized mass production of business card site using Drupal. Developed a city portal using Django and Pinax. Organized infrastructure and team training. System Administration. Used Python, JavaScript, PHP, Bash, Drupal, Django, Pinax, Redmine, Trac, Debian, etc.
* Jan 2004 – Oct 2009 – Zakrevskyi Entrepreneur. Web-development using PHP, Python, JavaScript. Development and maintenance of an industrial business portal. Used Python, JavaScript, PHP, Bash, Django, PostgreSQL, MySQL, Dojo, jQuery, Zend Framework, Pinax, Debian, etc.

Interests:

economy, philosophy, ancient history, ancient culture, SAMBO, mechanics