Resume

Solution/Application Architect, XP coach, Python/FullStack Software Engineer

Contacts:

Ivan Zakrevsky
Skype: industrialnet
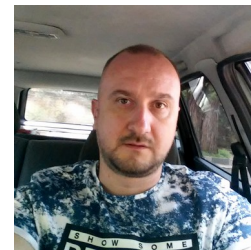E-mail: emacsway@gmail.com

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 expert in Software Design and Agile.

Was participated in development of software for various business domains - B2B and marketplace platforms, social networks, enterprise content management systems, content aggregators, GIS and travel portals, intracorporate portals etc.

My main interest is to make the code clean, simple and maintainable; the implementation of new features rapid and cheap; the curve of cost of code change asymptotic (instead of exponential); the risks low; the team successful.

Real experience of establishing economically successful Agile development. Deep background in XP, TDD, DDD, Refactoring, Evolutionary Design, Clean Code/Architecture, Microservices and others. Extensive bibliography.

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

Education:
1994-1999 - The Dnepropetrovsk State Technical University of Railway Transportation, master's degree
2018-present (up to 2020) - Private Institute of Higher Education "Zaporizhzhya Institute of Economics and Information Technologies", Software Engineering, master's degree

Open Source Projects:
Python:
 - rope (contributor) - a python refactoring library.
 - SQLBuilder (author) - a lightweight Python SQLBuilder. Can be integrated with Django.
 - cache-dependencies (author). Cache-dependencies (former Cache-tagging) allows you easily invalidate all cache records tagged with a given tag(s). The package supports Django.
 - Ascetic ORM (author) - a lightweight Python DataMapper ORM (the project is incomplete).
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.

Articles:
Articles in English:
- "Implementation of Repository pattern for browser's JavaScript"
- "About my experience of using Django Framework"
- "Design of Service Layer"
- "About problems of cache invalidation. Cache tagging."
- "How to quickly develop high-quality code. Team work."
- "Why I prefer Storm ORM for Python"
... all articles in English>>
Articles in Russian:
- "Про Agile на пальцах. Путь к быстрой разработке."
- "Реализация паттерна Repository в браузерном JavaScript"
- "О моем опыте использования Django Framework"
- "Управление логикой приложения. Про MVC, Use Case, Service Layer, CQRS, Event Sourcing и др."
- "Проектирование Сервисного Слоя"
- "О проблемах инвалидации кэша. Тегирование кэша."
- "Почему я выбираю Storm ORM для Python"
... all articles in Russian>>

Bibliography:
The full list of software literature read by me is available by link https://emacsway.github.io/about/bibliography/

Technologies & approaches:
- Programming languages: Python (proficiency, since 2008), JavaScript/TypeScript (advanced), Golang (intermediate), Ruby (elementary), PHP (proficiency, in the past).
- Development environment: Emacs (sometimes VIM, PyCharm).
- Approaches: XP, TDD, DDD, SOLID, OOP, OOD, Design Patterns, DIP, Refactoring, Evolutionary Design (aka Designing Through Refactoring), Clean Code & Clean Architecture, Software Architectural Design, Database Architecture, Low Coupling & High Cohesion, SRP, OCP, GRASP, Microservices etc.
- DB & storages: MySQL, Spatial, PostgreSQL, PostGIS, tsearch2, MongoDB, Redis, Memcached, replication, sharding, analysing, SQL optimization.
- Frameworks: Django (mainly), Flask, also used Aiohttp, Tornado, Twisted, Ruby on Rails, etc.
- JavaScript Frameworks: Dojo, Dojo2, jQuery, AngularJS, Angular2+.
- 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.
- Services: New Relic, Heroku, AWS, Atlassian.
- Others: Docker, Eventlet, GDB, Celery, geographic information system (GIS), Storm ORM, multiprocessing, threading, queue, async, Solr, morphology, business-transactions, high-load, high level concurrency, Remote Facade, DTO, cache invalidation based on tagging, automatic text processing, ACL, social networks integration etc.
- Also, I've a good (5+ years) experience in PHP, Zend Framework, Drupal, but it, mainly, in the past (2004-2009).

Experience:

Feb 2019 - present - Team Lead at a Fitness startup.
- Introduced the best Agile development practices: CI/CD, TDD, Evolutionary Design, Collaborative Development, Code Review and Code Reading, Iterative

planning with Agile tools, XP, Scrum etc.
- Had created a turning point in the qualitative improvement of the development economy.
- Multiple improvements of the reliability. Multiple reduction of the number of bugs.
- Successful dealing with extremely complicated Domain/Application Logic.
- Significant architectural improvements using DDD, EDD, CQRS/ES, Microservices, Clean Architecture.
- Splitting the monolith.
- Working effectively with legacy code
- Practical applying theory of algorithms.
- Drastic improvement in performance optimization.
- Participating in the team building. Interviewing of candidates.
- Performing responsibilities of Solution Architect.

Jul 2017 - Feb 2019 - Solution Architect and XP coach at Business Technologies www.biz-tek.ru.Was participated in development of the enterprise content management system for medical journalism.
- Introduced best practices of Software Design and Agile software development (TDD, XP, CI, CD, DDD, Microservices, Evolutionary Design etc.).
- Keeping the flattened change cost curve of the project. Clean architecture and rapid development. Quick MVP creation.
- The scope of MVP was increased in twice due to high team velocity.
- Almost no bugs due to TDD and Clean Code.
- Created proprietary library for Django model audit which implements Slowly Changing Dimensions (SCD) - Type 4 and supports versioned relationships.
- Applying "Monolith First" principle of Microservices building.
- Technology stack and approaches: Python, JavaScript, PostgreSQL, RESTful API, Django, Django-rest-framework, Angular5+, AWS, Heroku, Docker, Scrum, Jira etc.

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, Evolutionary Design, 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.
- Was performing some responsibilities of Solution Architect.
- Created the high-level library Store.js to handle data on client-side.
- Technology stack and approaches: 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 Qarea on project rebelmouse.com, the best online CMS for social websites.
- Actively used Evolutionary Design, TDD, code review based on catalog of refactorings and catalogs of code smells.
- Debugging on production. Low-level debugging with GDB.
- Backend/SQL high performance optimization. Monitoring with New Relic.
- Implemented social networks integration, polymorphic relations, ACL, statistics, data analytics.
- Technology stack and approaches: 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, a Russian travel portal and FAQ for independent travelers. Technology stack and approaches: 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, integration with geonames services, integration with partner's API, advanced ACL, flexible user notification, variouse Ajax UI widgets, cache dependencies, phased cache, cache cleaning synchronization with DB transactions. Backend/SQL high performance optimization.

Jul 2010 - Jun 2011 - Team Lead at Soft-Ukraine. Developed an intracorporate portals. Technology stack and approaches: Python, JavaScript, Bash, Django, Dojo, jQuery, PostgreSQL, RESTful API, MVVM, dojox.data.JsonRestStore, Ajax UI, Ruby On Rails, Pinax, Debian, Multilingual, etc. DevOps.

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. DevOps. Technology stack and approaches: 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. Technology stack and approaches: Python, JavaScript, PHP, Bash, Django, PostgreSQL, MySQL, Dojo, jQuery, Zend Framework, Pinax, Debian, etc.

Afterword:
"The value of good software design is economic: you can continue to add new functionality quickly even as the code-base grows in size." ("Design Stamina Hypothesis" by Martin Fowler)