Svilen Dobrev *making software - for people, by people*

g'day.

i've been making *software* for.. long time, around the world, as projects of all sizes and levels of impossibility. Seen many *different* terminals, languages, designs, projects, customers, teams, organizations and cultures, and *attitudes*.

So i found that.. the interesting and difficult part is the *people*. Technicalities are solvable. Software is just a limited, twisted way of *communicating knowledge* between people, through space and time. It is people who cause anything. Making software is a game of *cooperation*, exercise in *correct thinking*, all participants are *translators*, and *Trust* is most important thing.

az@svilendobrev.com

Seems i like to *mediate* the extremes - western or eastern, experts+newbies, gears+art, real+virtual. i put *passion* in my work and *do care* for it. i trust people.. and thus weigh them. i work With, not For. i fight For, not Against. i do Work, not Time. i like ideating, improvisation, mentoring, team- or organization- building, steep changes, persistence, diversity. To *efficiently* make *effective* software, i push for attitudes that empower people, tuning the *culture* to allow it.

Reality.. is very complex. Complexity is tamed with structure. Structure means *languages* - specific for the notion, domain, product, project, task, people. So one can dive in the real issues.

i make (software) projects from ideas, people and software. Be it possible or not, regardless what has to *change* - software, organization, people, or.. *me*. As life is change is life.

So.. *Hands-on* software mentor, “playing-trainer”, leader, architect, CTO, analyst, hacker.. programmer.. Friend? Possible. More about me, software and life at **www.svilendobrev.com**.

An Impossible dream project? What about?

Personals

- now in Varna, Bulgaria. born 1970 Bulgarian, +Australian citizen
- fluent Bulgarian, Russian, English ; with family+kids; non-smoker; mountains, bicycle, e-foil, sea lover

Specialties and interests

- the *Team* as alive, self-tuning instrument for *translating knowledge* into software; *organisational* patterns, cultures and health ; leadership and empowerment; proximity and communication patterns, de-hierarchisation
- *mentoring*, *teaching*, *learning*, inside-trenches organisational analysis and coaching ; philosophy of software
- *requirements e*ngineering, knowledge and technology management, audit and analysis of code/ workflows/ organisation
- *methodology*, *agile* vs CMMi process; global teamwork; *cultural* differences; mental gears/contexts, selective stupidity
- *usability* engineering, use-cases; user experience/ interaction/ interfaces; human-machine systems and perspectives
- *testing* as art; version and *configuration* management; documentation workflows, traceability, trackability; licensing
- project management, life-cycles, estimation, negotiation, risk and resistance-points, strategy mapping
- *business analysis* in many domains – software itself, media, content/metadata, finance/accounting/banking, HR, chips-design, bureaucracy, betting, CAD/CAM, mechanics, physics, hardware, maps, graphics, audio, pre-print, simulation
- *making languages*; using general languages as carriers of higher-level *specific* ones; *generating* code; *self-interpreting* objects; grammars, compilers, interpreters, virtual machines; protocols, frameworks and aspects
- *modeling*, *architecture*, *flows* - data, code, events, *time*; analysis; optimisation; efficient-effective; persistency, databases, bitemporal data+code, durable execution, messaging, semantic relations; sync/async; client-server/ standalone/ distributed
- languages: *python 25y*, C/C++ 25y, javascript 7y, java 3y, graphql 3y, html/css 25y; perl make sh; many more.. from prolog to erlang/clojure to various assemblers and machine languages
- toolkits/frameworks: django flask fastapi react nodejs express ; qt wx gtk kiwi; sqlalchemy; many more
- platforms: linux, android, web, desktop, dos; apache; many compilers, assemblers.. ; docker
- paradigms: object-oriented, functional, event-, rule-, dependency- driven, markup, behavioral, ..
- persistency: SQL- sqlite postgresql, xtdb/ bitemporal, couchdb, mongo; rdf/ sparql, key-value; API-design, graphql/ rest/ ..
- tools of the trade: vim, linux, vcs: bzr/git/hg/svn/cvs; e2e testing, selenium; Jira, wikis in general
- my *own* python crash course (~90+ people in last 10 years), and many DIY tools - github.com/svilendobrev
- www as runtime env; (micro)service vs library; search, accessibility, multi-linguality; digital signatures, smartcards
- devOoops.. or more. Get job *done* - software, hardware, or psychology; fix the dripping tap, broken chair, drivers or paperwork; hack, negotiate, *find* a way, *fight* for it.. It's all one jazz.. *Engineering*
- *reverse engineering*; search, anonymity; protections; social nets, digital identity; Do-It-Yourself, repairing.. stuff
- reflecting, exchanging ideas and opinions; artistic design; applying knowledge across fields; making toys and playing
- *caring* about what i do. With extreme attention to detail – which assumes freedom and no noise... and nothing fake..

Motto-s and thoughts

- Find a *friend* to be your senses.
- With time, places *change* people. Other way happens noticeably only while *coming in* - or *switching on.*
- Software is a (twisted) way of *communication*. Sometimes to yourself from Tuesday.
- *Languages* are your tools. Make them if no suitable ones.
- One can't make decent *tool*/thing if never has been *user* of it.
- If you make something usable by idiots, only idiots will use it. Do not try to outsmart the user.. who might be you.
- Software is about *people*. Association is a great thing - follow your common intuitional sense.
- Trust is the *essence* of software. Machines *trust* 100%, but people.. don't. How much trust or distrust can you handle?
- Software is made of decisions (and so is life, but most decisions are materialized and don't look as such).
- www is *global* village, anyone at *near-zero* distance.. but never zero. And one may need that warm touch - or good kick.
- In making software, the doable - sellable - wanted - needed, rarely overlap. Finding the dream spot may ruin illusions.
- First rule of Leadership: it's all *your* fault.
- Manager's choice: change *People* or.. *Change* people.
- Don't always believe yourself. But always believe *in* yourself.
- There's no "cannot" - only "do not want". And, there's no "have to" - only "want to".
- Search for (people-who-bring) questions. They make life interesting. Not answers - many industries+institutions do that.

Experience

python re-engineering contract - parcelHive, Varna-Sofia, 2025

• manufacturer of remote-controlled lockers - couriers, vending.. Re-working legacy overgrown python+js software on kiosk-like machines, ~25 variants, into coherent and maintainable form. FSM workflows as source-of-truth, server-side-rendered UI, capability+lifecycle-based layers of devices, storage, i/o (firmware too). Digital Twin prototyping. 1-2 people.

CTO / Chief Architect - eCollect.org, Varna-Sofia, 2021-2024

• receivables fintech scaleup. Team 10 remote; in-house node.js+mongo+vue ~250K js, event-sourcing DDD microservices. "*No coding, just other 90%*". Develop and mentor team, educate management and other depts. Find dev and system bottlenecks, "buy" time+resources to fix. Streamline processes, code reviews, releasing-CI/CD, UX studies; be product owner - stop "everybody-tells-tech-do-this-now", sort Jira, gitlab. Talk software to clients. Pitch tech to investors.. Acquired
• coding outside main system: extract+visualize js event-flow-maps, python ; few "paypage" projects, node.js+mongo
• after acquiring, lead a remake for huge volumes - durable execution /temporal.io, bitemporal data /xtdb, python.

CTO, mentor, team lead, architect, main coder - Resolve+FleetPal, Varna, 2017-2021

• small startup with friends. About dutch dental-insurance B2B workflows, then american truck-maintenance housekeeping. Doing most hats.. machinery, product road-maps, frontend and UX, methodology, workflows, team of 3-7. Max efficiency-per-teammate.. python, django, graphql-instead-of-drf, React+Blueprint SPA via browser-editor generative framework. Mentoring 2 younger colleagues into better persons. (~ 90k python + 90k javascript and up)

software expert and methodologist, leader of software group - ZMDI / IDT, Varna, 2014-2017

• branch of german fab-less chipmaker. Caring for 2 software teams (8 ppl); represent software side in complex projects; mentor people of various backgrounds; push improvements in culture, methodology, technology; hands-on - embedded or applications, assemblers to python; tools for requirements engineering, chip-description and testing; watch functional safety.

python mentor, coach, analyst - Strypes.eu, Sofia, 2014/contract

• python courses for 20+ people, beginners to advanced. Noticed fast-growth pains.. did a study, interviewing trenches and up, on ways of work, culture, tech, relations, pains.. and suggested changes for healthier structure, process, attitudes, culture.

python team Mentor - Ericsson, Athlone/Ireland, 2013/contract

• dozen new people, most straight from the uni... to become programmers.. and a real software Team.. in about 4 months. IMO.. Succeeded. As long as proper mentality gets in, the rest is.. persistence. (update 2019: they are all seniors now)

architect-maker - Meeter, Varna-Tokyo, 2012-2013

• distributed personal places+chat - couchdb as messaging. Store data as-is, let it flow, react on changes. Focus on user. A friend did iphone, i did all else: http-rpc-with-memory/replication, back-end, android+web apps (python, java, web).

CTO/ lead/ architect/ coder - Woosha.it, Varna, 2010-2012

• moodyTV: *content-discovery* of metadata, personal channels/filters; android,web,qt apps (3ppl 6mo; python, java)
• brightBird: movie-search *social* service. Match/classify metadata, user patterns. (3ppl 5mo; python, sql, sparql)
• notionery: a garden of *your personal notions* - remember, learn, express, in own terms, translate to/from all else incl. past.

linuxteam leader, architect, methodologist, programmer - SIStechnology.com,Varna-Konstanz/Germany-Singapore, 2003-2008

• HOR - enterprise HR+payroll, 90k python sql wxpython; 8ppl 2years; replace legacy, represent *ever-changing*, entangled legislation+relations. Notions+languages for pay-flows+values; multilingual bi-temporal *data and code*,.. Leading it all
• STC - *Singapore* Turf Club's betting terminals+kiosks renovation - *global outsourcing* effort; 2 end-user-apps: C++

python java, 6ppl, 2+years; ~6 months in Singapore, 2 in Germany. CORBA, realtime payments. *Team* leader, core dev, architect, relations to WN-Konstanz, WN-Singapore, STC. Difficult cultural mix.. 3 hops who didn't talk to each-other. Most difficult but rewarding project. My team survived crossing 4 cultures, and "end-users said they like your software"!

senior software engineer - Vintech, Varna, 2002-2003
- topological parametric modeling language + control of mechanical technology - in CAM/CAE plant system; python
- graphical 2D backend API from python into C++; fix + wrap equation-solver cassowarypy

senior software analyst - Petrosys, Adelaide/Australia, 2000-2002
- geophysical maps drawing framework - 2D MVC hierarchy, polygon-processing, C++, python; import from arcSDE
- apply organisational patterns - success in changing company culture - both as technology and people communication

tech lead/ test lead - Motorola, Adelaide/Australia, 1999-2000
- compiler for analog/mixed-signal Verilog-A/AMS, merge and maintain legacy compilers; testing and build-automation; C/C++, sh,perl,make, lex/yacc,pccts; solaris hp linux; full waterfall, requirements tracing, generated documentation
- tasting another language-culture-continent-society - multi-cultural environment in huge global corp...
- *learning* about software made industrially, "pipelines" in CMM4, and *teaching* about software made efficiently

software engineer - Vintech, Varna, 1997-1998
- accounting engine, user terms and languages for reports/ operations, wave-calc. 70k C++ btrieve, 3 ppl. Still used ~2022

sysadmin + software accounting analyst - Biochim Bank, Devnya, 1995-97
- LAN maintenance ; rule-based ledger correctness/conversion, on-the-fly-translation, novell/dos/os2, Clipper, C/assembler

lead programmer - Inmeco & Navel, Varna, 1992-1994
- Missile-launcher simulators - 3D real-time animated simulation of combat for soldier training - real weapon, overhead projecting on huge screen, modeling airplanes, vehicles, trajectory physics; trainer workplace; C++/asm, 3ppl sw/ 8ppl all
- Danube river map+radar/GPS - GUI+map technology; large accurate maps: make, compress, view; i386/penpad, C++/asm

publisher + programmer - Algorithm5, Varna, 1991-1993
- books, papers, cards; enhancing DTP/preprint software - reverse-engineer, patch, make fonts+tools - for cyrillic

cofounder - Specialized Intelligent Systems, Varna, 1990-1992
- with 2 schoolmates: accounting with replication over modems, Clipper, C, asm

other projects:
- curated collection of audio records of tales, songs, theaters from LPs, CDs, radio.. and the machinery behind - record/ cut/ compare/ publish.. some old frozen version at http://gramofonche.chitanka.info
- my site (static html php css javascript python/generation, shared apache, mod_rewrite)
- 2012: idea: as making software *is* learning: software house-school, learn by doing real projects. Didn't work, chicken+egg
- 2012: someones.com, Melbourne/Australia - social-media backend-integration (python, flask, nose, selenium)
- 2009: wyplay.com, Marseille/France - media-player UI, python, embedded; without communication, software ain't edible
- 2008: desktop client app for customs.bg B2B document-flow (python, 4mo, soap, XML-signing, smartcards, gtk/win)
- ~2000: teaching reverse-engineering and analysis - several essays
- university: set-based unification and Prolog interpreter - thesis, 500Loc prolog ; Text editor in C/asm (PC i86)
- school: statistical trends graphical analyzer-visualizer ; simple run-length graphics-compression lib (M6502 Basic + asm)

Education

Technical university of Varna (VMEI), 1986-1990
- M.Sc. in Electrical Engineering / Computer Science
- advancing course: 5 years taken in 4, twice more disciplines - all excellent

Mathematical High college of Varna (2 MG), 1983-1986
- "advancing education" class of 4 students, 7 years taken in 4, gold medal

Contacts

e-mail: **az@svilendobrev.com** -- my site: www.svilendobrev.com -- work and me: svilendobrev.com/rabota
some code: github.com/svilendobrev ; svilendobrev.com/rabota/mysoft.html
cv-resume-biography: **svilendobrev.com/rabota/cv** - **printable-pdf** -- **linkedin.com/in/svilendobrev**

SvD