

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

g'day.

i've been making *software* for 25+ years, on 3 continents, as projects of all sizes and levels of impossibility. Seen many *different* terminals, languages, designs, projects, customers, teams, organisations and cultures, and have met even more different *attitudes*.

So i found that the more interesting and difficult part is.. the *people*, while all the technicalities are somehow solvable. Software is just a limited and twisted way of *communicating knowledge* between people, through space and time - and it's people who cause anything. Making software is a game of *cooperation*, now vs future, exercise in *correct thinking*, where all the participants are actually *translators* - and *Trust* is the most important feature.

And i found i also successfully *mediate* between the extremes - western and eastern thinking, technical & applicational, experts & newbies, gears & art, real & virtual.. i put *passion* in my work and do *care* for it. i do trust people.. and thus weigh them. i work With, not just For. i fight For, not Against. i do work, not time. And put toys at github.com/svilendobrev.

So.. i make (software) projects from ideas, people and software. Be it possible or not, regardless what has to *change* - software, organisation, people, or.. me. Because life *IS* change.

Shortage of people? No. Inefficient thinking, still trying to store human knowledge about reality in primitive languages. Reality.. is very complex. Complexity is fought with structure. Structure, in the field of knowledge, means languages - specific for the notion, domain, product, project, and people.

i love idea-generating, *nice* software, mentoring, team making, steep changes, cultural diversity, organisational building, improvisation. i can help to *efficiently* make *effective* software, and tune the community and *culture* to allow for it. i make *languages* that talk the patterns of the task. And *attitudes* that empower people. So one can start solving the real issues.

Hands-on software leader, advisor, mentor, "playing-trainer", architect, CTO, hacker... programmer.

An Impossible Dream project? What about?

More about me, software and Life at www.svilendobrev.com.



az@svilendobrev.com

Personals

- now in *Varna, Bulgaria*. Bulgarian and Australian citizenship
- Fluent Bulgarian, Russian, English
- born 1970 Bulgarian, family+kids; sworn non-smoke mountaineer

Specialities and interests

- the *Team* as live, self-tuning instrument for *translating knowledge* into software; leadership; proximity + communication patterns, de-hierarchization, *organisational* patterns and health
- *mentoring, teaching*, inside-trenches organisational analysis and coaching... and learning
- *knowledge* and technology management, audit + analysis - of code, workflows or organisation; philosophy of software
- methodologies, *agile* CrystalClear vs CMMi/formal process; *outsourcing* + distributed global development; mental gears, *cultural* differences, pair-programming, context-switching, *attention* to detail, selective stupidity
- *design of languages*; using general languages as carriers of higher-level application/*domain specific* ones; *generating* code; *self-interpreting* objects; grammars, compilers, interpreters, virtual machines; interfaces, protocols, frameworks and aspects
- *requirements* and *usability* engineering, use-cases; user interaction and interfaces, analysis perspectives - technical + people, end-user + organisational; man-machine systems
- *testing* as art; version, configuration and project management, lifecycles, estimation, negotiation, risk and resistance-points; documentation workflows, traceability, trackability; licensing
- *business analysis* in many domains – software itself, media, content/metadata, accounting, HR, customs, betting, CAD/CAM, mechanics, physics, banking, hardware, maps, graphics, DTP, simulation
- *modelling, architecture, flows* - of data, code, events; *analysis; optimisation*; efficient vs effective; *persistency*, databases, messaging, *bi-temporal* data+code, *semantic* relations, sync/async and distributed processing
- languages: *python* /13y, C/C++ /23y, java /3y, javascript php ruby, asm; yaml xml html css; perl make sh; prolog...more
- platforms: *linux* /10y, android, web, desktop, sun, hp, dos; apache; many compilers, 80x86 assembler, dos extenders
- paradigms: object-oriented, functional, event-, rule-, dependency-driven, markup, behavioral,..
- persistency: couchdb, sql - sqlite postgresql mysql; rdf, sparql, key-value, non-relational; corba

- toolkits/frameworks: wsgi, django flask werkzeug, oauth; sqlalchemy; qt wx gtk; STL; many more
- tools of the trade: vim, linux, vcs: bzr/git/hg/svn/cvs; unittest,nose,selenium; trac,redmine, wikis in general
- have built and using my *own*: tool sets, coding style, methodology, python clone, persistency layers, UIs, web-framework
- www as runtime env - service vs library; API-design; digital signatures, smartcards; search, accessibility, multilinguality
- DevOps.. much more. Get job *done*, be part of it - software, hardware, or *psychology*; fix the dripping tap, clicking disk, drivers or paperwork; teach, hack, negotiate, *find* a way, *fight* for it.. It's all one jazz.. *Engineering*
- *reverse engineering*; search, anonymity; protections; social *nets*, digital identity
- reflecting, *exchanging ideas*, having opinions, *Do-It-Yourself*
- *artistic* design and layout; apply *knowledge across fields*; making *toys* and playing
- *caring* about what i do. With extreme attention to detail – which assumes freedom + no noise... and nothing fake

Motto-s and thoughts

- Find a *friend* to be your senses.
- *Software* is actually *about people*, not about machines.
- *Languages* are your tools. Make them if no suitable ones.
- One can't make decent tool/thing if has never been *user* of it.
- Do not try outsmart the user.. which can be you.
- If you make something usable by idiots, only idiots will use it.
- Association is a great thing - trust your common intuitional sense.
- Trust is the *essence* of software. Machines *trust* 100%, while people.. don't. How much trust or distrust you can handle?
- www-web makes the village *global*, so anyone is at *near-zero* distance. But never zero. And sometimes one needs just that - warm touch - or good kick..
- With time, places *change* people. Other way happens noticeably only while *coming in* - or *switching on*.
- Any day spent not *learning* anything, is a wasted one.
- Development and Business are the sides of a wall called Users. Drilling across is hard, and the *rope* won't go *through* by only pushing or pulling – do *both, together*.
- Software is a (twisted) way of *communication*. Sometimes to yourself from tuesday.
- Programmers should stop coding Reality into bytes, *grow up* and empower themselves with *Domain Specific Languages*.
- Mastery of someone is measurable by their (self-made) *tools* - material or mental. The deeper the experience, the more the "*self*"-part. Same applies to societies and cultures.. toys, tales, games are tools to pass knowledge.
- First rule of Leadership: it's all *your* fault.
- Manager's choice: change *People* or.. *Change* people.

Experience

software expert and methodologist, leader of software group .bg - ZMDI Varna, 2014-now

- branch of a german microelectronic company. Caring for software teams in Varna and Sofia (8 people) and their capability, relationships and good "health"; representing software perspective in complex projects, mentoring and teaching people from various departments, teams and backgrounds; pushing towards cultural, methodological and technological improvements - group- or company-wide, helping hands-on and giving practical advices. Deeply embedded or application software, from assemblers to python, functional safety, chip description frameworks, methodology, organisation, management.

python tutor - ZMDI Varna, 2014/contract

- branch of a german microelectronic company. Teaching 8+ people, mostly beginners/non-programmers

mentor, coach, analyzer – python, teamwork, organisation – Strypes.eu, Sofia, 2014/contract

- Dutch nearshoring company. Apart of python courses - beginners, intermediate, advanced - for 20+ people.. There were fast-growth pains.. so change was due. Did a study (many interviews, in trenches or management) of ways of work, culture, technology, knowledge management, relations, teaming, pains, funs, etc, and suggested quite a few changes in various aspects towards agile patterns, attitude and culture, then mentored some leaders. Automated testing of one tough-going project, as example of a different culture/attitude/approach to programing as activity

python team mentor – Ericsson software campus, Athlone, Ireland, 2013/contract

- dozen new people, most straight from the uni... to become (python) Programmers.. and a real software Team.. in about 4 months. Impossible? depends.. As long as proper mentality gets in, the rest is.. up to them. Maybe i did it. But we'll know later.. Meanwhile, auditing some core software and translating various APIs into python

architect-maker - distributed personal places+discussion platform - messaging via couchdb. 2012-2013 Varna-Tokyo

- Put data in database. As is. Let it flow wherever, on its own. Relax. React on changes. Fast. Focus on user. Now. (just 2 people, me doing backend, web-app and android-app, other guy doing iphone and related. So - architecture, protocols, project-management and coding - 2 servers in python (17kloc), android native client (12kloc java), messaging via rpc-with-memory over couchdb replication and it's eventual consistency, web/flask/html/css)

...dreaming - a software house-school, for making real projects and learning by doing so - and other way around. 2012 Varna

- wanna learn? have projects? call. It will be interesting. All software making is actually *learning* - just noone admits it.. (didn't work. A chicken-and-egg - eventual students want ready+paid projects, companies with projects want ready teams)

senior pythoneer, someones.com, Melbourne, Australia, 2012 (short contract)

- a generic social python-backend-integration framework (facebook, twitter, others. Using flask, celery, nose, selenium). in a team of 7, in a place that opposes any change. Done.

hands-on CTO/ Dev-Lead/ architect/ all-thats-needed, Woosha.it; 2010-2012 - risk-it-all startup, Amsterdam+Varna

- redefine *search* and *decision* as paradigms in human terms.. Doing a lot of philosophy, analysis, invention, mentoring; being at same time overall technical leader, Varna-office manager, architect, inventor, mentor, and coder. python+java, linux/, functional and generative code, 2 teams - 3+3 people - in 2 countries. Mentoring a young entrepreneur into systematic business-analysis and software philosophy
- [BrightBird.com](#) - movie search + notification *social* service. Match+import various *metadata*, user patterns, group+classify on various notions. Did info/concept architecture, backend design+coding, data analysis/import, managing teams; (python 10kloc, API, ruby/javascript frontend, 3+3ppl 5months; backend: python/django,celery,whoosh,mysql,virtuoso)
- [Notionery.com](#): wild idea- software Garden to grow *your personal notions* - remember, learn, express yourself, in your own terms - and translate to and from all else, incl. past. Few months of concept digging... *Semantical* web5.. Far fetched ?
- moodyTV: *content-discovery* platform, many metadata sources; shown at IBC2010-Amsterdam. Android thick on-off-line client app in java; python for web-app and desktop/qt app and backend - django,celery,mysql. Personal channels, profiles+suggestions, manual faceted filters, sharing saved filters/searches. (3ppl, 6months, 8kloc python, 20kloc lean java, API+model generation). 2 related android+backend apps
- [facer](#) - API-description language, python (methods, values, clone, inherit, validate; visitors generate tests, docs, clients,..)
- [jgenerator](#) - model-description language, python - schema, types, dialects, generate boilerplates for qt, Java, SAX, sqlite
- [fuselage](#) - tag-based semantic file system, python, over FUSE. Pluggable aspects (parsing or interpreting). Querying and relations between tags/items. Modular architecture, different user-ends or back-ends (2ppl, 2 months, 2kloc python)

mercenary senior dev at Wyplay.com, Marseille, France; 2009 (short contract)

- UI-engine in python, embedded, linux/gentoo, media-player device; medium company, very entangled work / repository structure.. A bit extreme independence - Viva la liberte! - yeah, but without communication, software ain't edible..

owner, Svilen Dobrev Consulting; 2008-... - advisor, methodologist, architect, programmer

- [smok](#) '2011 - python in my language - no more englishizing. Use native bulgarian words instead of english keywords.
- [my site](#) (html+php+css+javascript+python/generation, shared apache, mod_rewrite)
- home network and data infrastructure (5 linux machines and others)
- CSend'2009 for [Bozsoft.biz](#): desktop end-user client for official Customs.bg B2B document flow and exchange protocols - message threading, templates etc (python, 7kloc, 4 months, FSMs, soap, *XML-signatures*, smartcards, windows/gtk)

linuxteam leader, SIStechology.com, Varna; 2003-2008 - team-lead, architect, methodologist, manager, mentor, programmer

- HOR - enterprise Human Resources + payroll management system, 90kloc python, sql, wxpython; 8people 2.5+years; replacing long legacies and representing an *ever-changing*, extremely complicated and entangled legislation reality and relations, expected product life >10 years - hence max flexibility and expressive power, hierarchical domain notions like values and payflows, *bitemporal* code and data, multilingual values, ... made 8 *business and internal languages*. Moving and *Leading it all..* extreme Agile Crystal-Clear methodology; architecting, managing requirements; mentoring, teaching, project-management, inventing, coding, restructuring everything; *fighting..*
- STC- Singapore Turf Club's betting terminals+kiosks renovation - big-money *global outsourcing* effort; 2 separate end-user-applications: 130kloc C++, 15kloc python, 2kloc java, 6people, 2+years; the team spent about 6 months overall onsite in Singapore, and 2 in Germany. CORBA, payments, realtime. Me being core dev, architect, *team* leader-manager, all *relations* to other 3 teams - WN-Konstanz, WN-Singapore, STC; difficult *cultural* mix and communication.. e.g. Requirements gathering through 3 hops, which don't want to talk to each-other.. Lots of future-proofing, refactoring, wrappers and code generation - FSM>C++, Corba>python>python>C++. Documentation flow for text- dia- openoffice-framemaker. Psychologically the most difficult and most rewarding project so far - hey i made a Team, survived crossing 4 cultures, and "end-users said they like your software"! >1000 machines live now..
- linux-and-outsourcing department leader - managed 10 people, lead and architected other 4-7 smaller projects (ERP, financial, betting protocol-simulators, betting apps), mostly python, C++

- resurrected *trust* for outsourcing from WincorNixdorf. covered all relations thereof. Studies over organisational culture and workflows of both companies
- built and lead successful united *self-organising Team+culture* - trust, responsibility, fun & self-discipline
- teaching 3+2 fresh grads into top-devs, Product Manager into Business Analyst; training customer devs at STC onsite
- [static type](#) - python-language for modelling statically typed attributes and structures; validation, optionality
- [dbcook](#) and [timed](#) - python-language to describe persistency of objects and relations, and bitemporal objects/attributes
- [reporter](#) - python-language+engine to describe reports - fields, filters, aggregators, layout, into text, pdf, html, excel, wx
- rights, [sqlalchemyAggregator](#) – python libs for description of access-control, and auto-aggregating SQL columns

senior software engineer, Vintech, Varna, Bulgaria; 2002-2003

- topologic parametric modelling 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 - small, fast company (geophysical maps)

- made a drawing framework - 2D hierarchy, MVC, polygon-processing, C++,python; visualise GIS map-data from arcSDE
- organisational patterns - success in changing company culture - both as technology and people communication

tech lead/ test lead, Motorola, Adelaide, Australia; 1999-2000 - EDA - multi-cultural env in huge global corp...

- new compiler for analog/mixed-signal Verilog-A/AMS, merge and maintain legacy compilers; testing and automate build; C, C++, sh,perl,make, lex,yacc,pccts; sun hp linux; full waterfall, requirements, autogenerate documentation - framemaker
- tasting another language-culture-continent-society
- learning about software made industrially - "pipelines"+ CMM4, and teaching about software made efficiently

software engineer/consultant, Vintech, Varna, Bulgaria; 1997-1998 - small company in a narrow, chaotic market

- made Sceti / Finix - personalizable *accounting* engine, in *human* terms+documents, user *languages* for reports, templating, operations, access-control. wave-calc. 70kloc C++ btrieve novell, 3 ppl. Change human-workflows. Still used.
- prototypes of statewide medical record system, doctor's workplace. "obj-oriented" structural documentation flow

sysadmin/ software+accounting analyst, Biochim Bank; 1995-97 - strategic midsize branch of a big state bank

- accounting software - rule-based correctness, ledger conversion, account-scheme on-the-fly-translation, automation
- LAN maintenance - PCs, novell/dos/win/os2, Clipper, C/assembler

lead programmer, Inmeco & Navel, Varna; 1992-1994 - small international companies, naval & military stuff

- Missile-launcher simulators - 3D real-time animated simulation of combat for soldier training - real weapon, overhead projecting on huge screen, airplane/vehicle models, trajectories, trainer workplace; C++, assembler, physics, 3ppl
- large accurate digital maps: make, compress, view, radar/GPS; i386/penpad
- Danube river map+radar – GUI+map technology; prototype for Gabchikovo channel navigation

publisher+programmer, 1991-93 – many books, papers, cards, art and:

- DTP software+Fonts - reverse-engineer, customize, use various DTP systems, make fonts+tools.. make all work in cyrillic

partner, Specialized Intelligent Systems, Varna 1990-92

- with 2 schoolmates: accounting over modems, clipper, C, assembler

in university..

- set-based unification and Prolog interpreter - thesis, 500Loc prolog ; other C/asm stuff

Education

Technical university 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 - Varna (2 MG), 1983-1986

- "advancing education" class of 4 students, 7 years taken in 4, gold medal

Some community projects

- making and hosting a community forum of [Sladka voda alternative school](#) and supporting the school
- see the code-link below for my open-source software
- do-it-yourself.. the machinery of my site, and my home - www.svilendobrev.com/napravisam
- teaching reverse-engineering/analysis – a few essays – lookup +svd, deadlisting, +fravia

Contacts

e-mail: az@svilendobrev.com

my site: www.svilendobrev.com

work, software, and me: svilendobrev.com/rabota

code, toys and projects: svilendobrev.com/rabota/mysoft.html ; github.com/svilendobrev

cv-biography-resume: svilendobrev.com/rabota/cv - [printable-pdf](#) - [brisk intro](#) -- linkedin.com/in/svilendobrev



SvD