Personal

Name:
Eric van Orsouw
Email:
eric@whitemagic.it
Cell:
+31 6 57200060
Date of birth:
December 12, 1966
Residence:
's-Hertogenbosch

Skills

C, C++, Java, C#, OO, UML, Hatley&Phirbai, HTTP, Rest, DVB, TVA, XML, XSD, JSON, JSON Schema, Cassandra, Zookeeper, Ansible, Qt, Wamp, Lucene, Visual Studio, Netbeans, CLion, IntelliJ, Jenkins, Gitlab-CI, Assembler, RS232, I^2C, CAN, TCP/IP, Jira, SVN, Git, scrum-master, requirements, Windows, Linux, Unix, Office, Docker, Robot framework, OpenCV

Hobbies

- Working with wood
- 10KM running
- Race cycling
- 3D printing
- Electronics
- Coding

Eric van Orsouw

Education, courses

Polytechnic Electronics **1985-1990**
HTS Den Bosch

AI – MOOC Stanford 2019, Java **EE** – Infosupport 2016, **Scrummaster** – Zilverline 2014, **MCD Web Development** – zelfstudie 2013, **Projectmatig werken** – Schouten Nelissen 2009, **Embedded Systems Architect** – ESA 2006, **OO Design/Analysis** – Datasim 2001, **Hatley&Phirbai** – ASML 2000

BHV – G4S 2016, **VCA-VOL** 2020

About me

Perfectionistic but pragmatic software engineer with an eye for usability in general. I like to think out-of-the box and strive towards solutions that can easily be reasoned about. I am proficient in working out ideas and requirements and have demonstrated experience in writing performant and maintainable code. I can perfectly operate alone with little input but also enjoy working as part of a team. My personal interest is to discover mechanisms over specials and software that functions predictably in degrading or resource-constrained environments.

Experience overview

Besi **2022-present**
Roles: Senior developer
I work in a small software team to develop and improve a machine that separates chips, inspects and sorts them into trays.

OMRON **2020-2021**
Roles: Senior developer, Team lead
I work with a small software team to take the automation of production lines to the next level.

de JONG DUKE **2018-2019**
Roles: Senior developer
For a next generation commercial coffee machines I design and implement an online API for a new GUI in C++. I also introduce new toolchain within the embedded team.

AXON **2017-2018**
Roles: Senior developer
I design and implement the next generation C++/Qt based scalable MPEG transport stream multi-viewer for the broadcast industry.

SeaChange **2008-2017**
Roles: Software Architect, Lead developer, Team lead
As an architect I analyze the impact of new requirements and describe the required changes to the system. As a developer I cooperate in the ground-up development of Java/NOSQL backend.

P.T.S. PHILIPS NXP ASML Delem **1999-2007**
Roles: Senior developer, Team lead
Outsourcing period with development in C/C++.

SPC/Company **1991-1998**
Roles: Developer, Senior developer
C/C++/assembler development of vision-based inspection software.

Senior Developer **Besi (2022-present)**

Besi makes back-end machines for the semiconductor industry. I work in a small team that develops and improves a machine called device-sorter-2 which is part of a chip-sawing-line. This machine inspects and sorts separated chips from a carrier into a tray using 2 concurrent pick-and-place heads. Development is primarily in C#. On the side, I work out the software aspects for new planning features at both device and machine level in documentation and prototypes.

Senior Developer, Team Lead **Omron (2020-2021)**

Omron Manufacturing has production lines to produce various electronic products. To remain competitive, much of the work is automated. Over the past decades, the emphasis has become more and more on software. Together with a small team, I bring consistency and uniformity in existing software.
One specific automation tool I worked on was called pick-to-beamer that assists an operator, using a camera and projector, in picking the right parts during assembly. Keywords here are simplicity and easy-of-use. In addition, I write various requirement documents for both new and existing software components.
Development is primarily in .Net Core and C#.

Senior Developer – Rest API **De Jong Duke (2018-2019)**

De Jong Duke develops and produces B2B coffee machines and strives towards continuous improvement. My task is the development of an online API on top the current control software to facilitate the outsourcing of a new GUI. I collect requirements and formally document these. I assure that the API is generic and free of machine specific elements. I implement the API in C++ and integrate these into the existing control software. To ease external development, I create an API stub-application that allows all relevant machine aspects to be simulated. During my period I help the team to depart from Rhapsody and introduce a new toolchain using C++11, CMake, Gitlab and Docker. This allows the developers to more efficiently work together as a team.

Senior Developer - Multiviewer **Axon (2017-2018)**

Axon wants an improved and scalable multi-viewer for the compressed domain that had to execute both on Windows and Linux and be written in C++/Qt. My task is to create a requirements document and implement the multiviewer as part of a 2-man team. I collect and document requirements and after approval by review. I designed the new multi-viewer to be fully on-the-fly configurable and controllable via json files and rest interfaces. For debugging I introduced a mechanism where at various places in the source, metrics are recorded. The result of one-year development was a drop-in replacement multiviewer that performed beautifully, was scalable and ready for new features. The metrics interface was a hit and provided useful insight during debugging, troubleshooting and automated integration tests.

Software Architect **SeaChange (2016-2017)**

SeaChange has a complex VOD back-office that is sell to operators like BBC and Ziggo to provide video-on-demand functionality to their subscribers. The back-office is very configurable and consists dozens of components and 100+ proprietary interfaces. Due to the relatively small installed base, every customer periodically has new features that have to be fitted. As an architect I am responsible for investigating these requests and come up with a set of modifications to implement the new features. For each request I investigate the solutions options, document the pros, cons and affected components/interfaces. Using reviews I decide the most effective solution. Afterwards, interfaces are unambiguously extended with syntactic and semantic changes and for the implementations, I create a detailed description of new/changed functionality while explicitly leaving out implementation details.

Lead Developer Back-end, Scrum master **SeaChange (2014-2015)**

The SeaChange software-suite includes BMS, a web-based product to manage and productize a VOD catalog. BMS evolved from an ASP-based demo product and needed a complete rehaul to become useful. My task was to provide a new back-end rest API to provide all required metadata and actions on which a new Javascript GUI was developed. I investigated the requests needed by the GUI team to fulfill their (new) need, described the API in terms of mechanisms that could be combined to query arbitrary results. To address backend scalability, I choose a identifier-only relational memory model combined with Lucene for searching and sorting. The result of this endeavor was a flexible API that allowed UI developers to customize arbitrarily and to achieve for example complex scenarios like virtual scrolling through large data sets.

Lead Developer, Team lead - Publisher **SeaChange (2010-2013)**

A critical component in SeaChange's VOD back-office is the publisher that hosts the public API serving millions of client devices (Settop-Box, PC, Tablet). SeaChange already had a SQL-based Publisher written in C# hosting a XML/Rest API and another publisher in C++ hosting a low-bandwidth binary interface. My task is to develop a Java version that is better scalable by using Cassandra and that could replace both existing publishers. I designed the publisher core as a library that provided all functionality using pluggable managers. I wrapped this library by a Rest layer to provide the XML/Rest interface. A separate team wrapped the library

to provide the binary interface. I put a lot of effort into designing mechanisms to overcome the lack of transactions in Cassandra. For example to do predictable and concurrent logging from different publishers. The result was a well performing and scalable publisher that is currently executing at a number of customer's sites.

Lead Developer – MPEG TS Analyzer **SeaChange (2008-2009)**

SeaChange missed a component to validate of MPEG transport streams. The metadata content and deviations needed to be visualized via a web-interface. Deviations should also result in alarms. My initial task was to develop a real-time analyzer that determines timing and content aspects from transport streams and exposes the raw metadata via a TCP interface. As a follow up I was involved in the interpretation of this analyzed metadata into a ready-to-eat form via a C# interface. For performance reasons I choose to use C++ for the analyzer. After creating an MPEG TS decomposition to figure out the relevant information streams I created an OO design. I then diligently used RAII for all classes and introduced a simple buffer allocator all to prevent memory leaks and fragmentation. The C# component I created used simple socket communication to synchronize with the analyzer. Using straight forward sequential code I converted the raw analyzed data into metrics that could be compared against standards and user settings.

The resulting analyzer was extremely fast, easily processing 500Mbit of sustained input on a single core. Additionally, it was very stable and robust allowing it to run 24/7. The C# interface and the associated visualization was functional but failed to meet expectations under heavy load. Specifically, flooding of alarms and its sequential processing needed a redesign. After inception interest from customers dropped and the product was no further discontinued. The C++ analyzer, being a standalone component, however is still actively used for metadata extraction in other SeaChange products.

NXP – Developer – Driver generator **PTS (2006-2007)**

NXP developed a DVB-H receiver chip for reception on digital TV on mobile devices. To verify variations in input and burst-error behavior a test stream generator was needed. My task was to develop such a generator and the analyzer counterpart. A subsequent task was a similar DAB test generator for a software defined radio project. I developed all tools iteratively in C++ separating stream processing as separate re-usable components. The result was a set of flexible script-driven commandline tools.

Philips – Developer – DVB-H/DAB generator **PTS (2006-2007)**

Philips semiconductors has the Hercules chip that formed the heart of analog TVs. My initial task was maintenance work on the very limited 8052-based Hercules core. As I worked on change requests I noted that driver support for the numerous chip variations was tedious and error prone. I suggested the creation of a generator-tool that generates C driver code based on the meticulous excel-sheets describing the chipset's registers and behavior. My task then moved to developing the generator. I created a design in OO/UML and implemented the tool in C++. After that I supported rolling it out in within the department. The resulting tool generated very readable C-code including comment. This significantly reduced the effort to support new chip variations and limited resulting bugs.

ASML– Developer, Team lead **PTS (1999-2000, 2003-2004)**

ASML produces extremely complex wafer stepper machines. During these periods my task was both maintenance work on the component AT (wafer alignment) as well as new development of component FM (Firmware Download) and MI (Metrology Interferometers). I created all designs in Hatley and Phirbai and implementations in C. The resulting code was integrated into the central codebase.

Developer / Senior developer **SPC/Company (1991-1998)**

SPC Company creates mostly vision-based inspection machines. The projects included a library robot, coat-hanger sorting system, blue berry sorter, crop-measuring system and the creation of an HMI library for machine human interaction. I started all projects by creating a functional and technical design using mostly common sense. I coded all software using C, C++ and for vision performance I often resorted to 80x386 assembler. The results of all these projects were functioning machines that were a real joy to work on and to see operating.