C U R R I C U L U M   V I T A E

of

Jacques de Hooge Msc
born on October 3rd 1957
The Hague, Netherlands

Contact data

Burgemeester le Fèvre de Montignylaan 206
3055 NH Rotterdam

phone: +3110 - 4183398
          +316-53901502

email: info@geatec.com

Education

Atheneum B

Delft University Electronics / Information Theory, a.o. with subjects in the area of compiler design, operating systems, databases, real time control systems, numerical mathematics, stochastic signal processing, image processing and artificial intelligence.

Several courses in IT didactics at the Nederlands Opleidings Instituut voor Informatica (NOVI, Dutch Training Institution for Computer Science).

Three year curriculum "Gestalt Development Training voor Individual, Talent and Profession" at Psychological Advice Center Hoogenraad-Oberman.

Profession

Active in technical and scientific software design since 1982.
Via own company GEATEC engineering sinds 1987 (www.geatec.com).

Activities

All working activities described in this CV were performed for diverse customers via my own company GEATEC engineering. Should you want to know more about the projects mentioned in this CV and my role in them, please contact me. Extensive references on request.

Competences

To build, as design-lead cooperating with team of developers, reliable, maintainable and efficient technical software, following requirements, within planned time and budget, and with pleasure for those involved in the process.

Writing a comprehensible, compact yet concise requirements analysis.

Fluency in English, both conversational and in documents. Validated level: native speaker.

Providing in-house training concerning software-software development: Requirements and domain analysis, Functional and technical design utilizing original or of the shelve patterns expressed in e.g. UML, C++ and STL, Python, C# or JAVA. Systematical, automated and modular testing. Design, simulation, realization and committing of industrial real time systems.

Practical, experience based quality management in software development: Setting up a flexible development cycle geared toward intensive cooperation, but bringing together personal and unique skills and contributions of each developer. Short cyclic delivery of intermediate results and continuous adjustment in a rapidly changing environment. Early simulation of system and context to uncover hidden requirements in order to reduce risks. Code reviews, automated unit- en regression tests.

Solving technical bottlenecks (performance, reliability, real time behavior, dynamical behavior of controls, speed, accuracy and stability of technical and scientific calculations)

Research

Developing a mathematical method and software for non-invasive 3D reconstruction of epicardial potentials from measurements with skin electrodes. Writing the mathematical and technical parts of a number of publications on this subject (www.geatec.com/publications).

Hobby's

Astronomy and giving lessons and lectures about that subject (www.sterlicht.nl - in Dutch). horse-riding, playing and incidentally teaching jazz piano.

Open source development ((www.transcrypt.org, www.qquick.org, www.josmith.org, https://github.com/JdeH)

Work activities, reversely chronological

Open source development, 2014 - now

Development of the Transcrypt Python to JavaScript compiler (www.transcrypt.org).
Development of several other open source packages ((https://github.com/JdeH) .

HAGA Teaching Hospital Cardiology dept., 2013 - now

Research concerning reconstruction of epicardial potentials based on non-invasive
measurement.

Contributions to several scientific publications.

Development of the required mathematical and 3D graphics software.

Tools & Platforms: C++, Python/NumPy/SciPy/Cython, Dicom, OpenGL,
Fenics/Dolfin, GMSH, Windows, Linux.

Rotterdam University of Applied Sciences, Technical IT dept., 2012 - now

Lecturer on the subjects Linear Algebra, C++, Robotics, Artificial Intelligence,
Statistics, Image Processing, Real Time Analysis, Theoretical IT a.o..

Fugro Geoservices, Leidschendam, 2012 - now

Development of an Internet application to make available on-line the results of
vibration measurements in building construction.

Data-acquisition and storage in a database. Making this data available graphically in
a web application incl. user, privilege and configuration management.

Tools & platforms: Python, C#, HDF5, SQLServer, Django, ASP .NET MVC.

LIAG Architects and Building Advisors, 2012 - now

Building and maintaining the company website www.liag.nl.

Tools & platforms: Python, Django, MySQL.

Changer, Leidschendam, 2011 - 2013

Upgrading Internet application and SOAP web services to enable customers to draw
up construction diagrams on-line in order to facilitate customized fabrication and
delivery of roof gutters.

Tools & platforms: Python, Django, MySQL, C#, ASP .NET MVC.

Legal360, Leidschendam, 2009 - 2012

Development of software to automatically classify judicial verdicts.

Design of a parameterized document generator for judicial documents.

Fugro-Jason GeoScience, Leidschendam 2000 - 2010

Setting up and monitoring a 5 year software quality improvement track, subdivided into a number of projects. Managing these projects. Goals a.o.: making applications suitable for automated testing, providing in-house developed class libraries with a stable, consequent API, converting Linux applications into multi-platform applications, improving developer documentation.

Proposing and implementing organizational change to improve coordination between the development teams involved in building software to detect, visualize and model oil and gas reservoirs. At the start each team was strictly focused on its own partial application. For user interfacing, algorithms and data model technical coordinators were recruited to strengthen lateral connections and consistency between the partial applications. This e.g. promotes GUI uniformity and compatibility of the utilized data models. Besides that, consulting specialists from all teams, work was restructured from version centered to project centered enabling innovations spanning multiple versions.

Introduction of a structured development method in which quality management through early testing and peer reviews are a fixed element. High-level designs are drawn up in UML and made available on-line to all developers through hyperlinks from automatically generated code documentation. Knowledge sharing is stimulated by presentations from developers scheduled regularly.

Providing in-house training, ranging from collective courses in the area of projectwise IT development , via UML, programming and Design Patterns to individual coaching of developers. With individual coaching, designs of new modules were drawn up together with their developers, which led to a significant improvement of the designs (and increased job satisfaction for those involved).

Building a GUI for an oil and gas flow simulator, utilizing a GUI library with built in support for the Observer Pattern, developed on personal title as an open source project (Eden, Event Driven Evaluation Nodes).

E-marketing company, 2008 - 2009

Providing several in-house courses for software developers, a.o. requirements analysis, functional design, connecting relational databases to object oriented software. Evaluation and advice with regard to several projects in the area of E-marketing.

Engineering company specialized in climate installations, 2008

Providing in-house courses object oriented design and programming in Java.

Dutch Educational Institution for IT (Nederlands Opleidingsinstituut Voor Informatica, NOVI), Amsterdam, 1988 - 2007

Drawing up courses Object Oriented Design and C++, and teaching them for many times to employees of a long list of large and small companies in the industry, financial services and government sectors.

Through the years the nature of these trainings changed. At first IT development managers wanted to get acquainted with the overall characteristics of object

oriented development (ROOS workshop). Later on programming courses became more important. In the end the accent shifted from programming to design.

Drawing up C++ exam specifications for the EXIN foundation (Institution for certification of Dutch IT exams), chairmanship of the Dutch C++ exam committee.

Baan, Ede and an offspin specialized in HRM, 2000 - 2005

Realizing of two systems for Competence Management. The first one had a companywide relational HRM database as its backbone. Security and privacy played an important role in this case. Data had to be protected in a layered fashion to prevent a wildgrowth of nearly equal competence profiles. This lead to uniformity in the use of profiles in e.g. assessments. Still in some special cases managers needed the freedom to deviate from the standard profiles under controlled circumstances, since local conditions and requirements could be different. This necessitated a quite extensive system of user privileges and integrity checks.

The second system was distributed and Web enabled. It was characterized by a strong Office look-and-feel, so that it instantly felt familiar to any manager without training.

TNO Advanced Chassis and Transport Systems, 2002

Providing an in-house training C and C++ programming of embedded systems.

TNO WT, 2001

Providing an in-house training C++ for simulation and visualization.

Borealis (BP subsidiary) Antwerp and Hessenatie Antwerp and ECT Europoort, 1995 - 2000

Requirements analysis, design, realization and commissioning of computer controlled cranes utilizing both PLC's and industrial PC's.

At Borealis, apart from the crane control, an automated storage management system for containers was designed and built. Important was the fact that in practice always also containers were stored and retrieved by hand. So the system was provided with unsupervised learning capabilities. Via feedback through container tags the system learned to know the whole of the container stacking area during movements of other containers. A perturbation in the container stack administration thus got sorted out after a short period of regular operations.

At Hessenatie there was a requirement for a short commissioning time. For this reason the control was completely simulated on a PC in advance. After this the software was converted to run on a PLC.

At ECT an automated control was designed and built for a large number of container cranes at the Maasvlakte. Also in this case simulation in advance was used to shorten commissioning time, but no conversion took place. The executables that first ran on the simulator were copied to industrial PC's on each crane without recompilation and ran flawlessly from the very first moment without any modification. This was remarkable, as earlier controls on similar cranes, written by a 3rd party, took months to commission.

<u>Centric Information Engineering, Nieuwegein, 1999 - 2000</u>

Training 4 shifts of non-IT academics for the job of technical designer. Training was hands-on, with lessons 2 days per week for several months, and 1 day per week to work out the assignments individually.

Since people were needed for production in short term, a compact training schedule was required. This is why 2 shifts were overlaid. While new subjects were explained to one shift, the other shift, in a different room, was busy making the assignments, having assistance now and then.

At the end there was a larger, but still individual, design- and programming assignment. All but one of the participants successfully concluded the training.

<u>Holec Machines and Devices (Holec Machines en Apparaten), Ridderkerk, 1987 - 1998</u>

Introduction of a structured development method for control software running partially on PC's, partially on PLC's. This development method utilized elements of a number of existing methods, in a mix that best fitted the kind of applications at hand and the natural way of working of the developers. Essential for acceptation was that the designers directly benefitted from the method since it enabled simulation in advance. With this the risk and the stress during commissioning were strongly reduced.

Auditing of projects, a.o. EC Delta Terminal and Hongkong International Container Terminals.

At the ECT Delta Terminal auditing was set up for all PLC software. Deliverable was a set of recommendations that has led to increased reliability of the control systems. At the core of the auditing method was the fact that quality was judged according to a number of explicit criteria. This helped focusing attention during the code reviews. Purposeful byproduct of these focused reviews was, that a lot of improvement issues were found, that were not covered by the mentioned explicit criteria.

At Hongkong International Terminals, subject of the audit was software drawn up by multiple parties. This led to recommendations for improvements both of the crane hardware and the control software.

Introduction of Factory Acceptance Testing (FAT). Since accurate real time simulation became a fixed part of the development cycle, it became attractive to familiarize the customer with operation of the control before construction of the controlled system was completed by means of the simulator. This led to early detection of omissions in the requirements, and to delivery of software that performed according to what the customer had meant to specify rather than what was actually in the documents.

Introduction of formal Site Acceptance Tests (SAT). Since behavior of the software during the FAT was adapted until the customer was fully satisfied with it, there was a solid basis for the SAT. Guided by the end result of the FAT and in consultation with the customer a test protocol for the SAT was drawn up. Before the actual SAT this protocol was worked through on site and eventual deviations were corrected. Thus the SAT became a mere formality and commissioning time was reduced strongly.

Providing in-house training courses. These concerned getting people acquainted with the used development and simulation method as well as improvement of the

design style. The latter e.g. led to an improved division of tasks between PC's and PLC's.

Baan, Ede 1997 - 1998

Providing a complete curriculum, 2 days per week for 8 months, to educate 16 talented people without any IT experience upto the level of technical designer. Drawing up and teaching all courses required by that curriculum. First, in consultation with Baan, a profile was drawn up that attendants had to meet before engaging in the training. This was translated into an entrance test by an external educational service.

After this the curriculum was drawn up. First principle was that people got offered a piece of knowledge if it matched concrete questions and needs they had at that instant. This is why the training was structured bottom up. First people learned to program and test. After that the need for design knowledge emerged, so design was the subject. Finally the need for requirements analysis skills surfaced, which was again responded to by a training.

Alongside the training, people did production work from the very first day. So they could put what they just learned in relation to the demands put by reality, and were more interested in expanding their knowledge.

In these 8 months, the group really "gelled". At the end it all came together into a large, practical groupwise assignment, from requirements analysis through design to coding and testing. Eventually 14 people got a fixed contract with Baan.

Traxis, Ridderkerk, 1992 - 1993

Renovation of a package of 26 computational programs in the area of electrical, mechanical and thermal simulation of electrically propelled trains. These programs were all written by professionals in the area of power electronics. The individual computational results were largely correct but the structure and maintainability needed improvement. Incremental improvement starting at the status quo turned out to be undesirable given the bad code quality. Instead, the core computations were dissected from the existing structure and grafted into a new, improved structure. Side effect was that the amount of source code was halved and that a better separation between computations, storage management and user interfacing surfaced. This improved possibilities to e.g. run lengthy computations in unattended batch mode.

Traxis, Ridderkerk, 1991 -1992

Introduction of data lineage management for engineering data. The chosen form of data lineage management makes it possible to trace the source of design decisions in current and past projects and to investigate and interrelate the consequences of alternative design choices.

Starting point was the requirement that the motivation for design choices should always be identifiable, also in retrospective on customer demand. This led to two practical rules in dealing with engineering data:

1. Input data or intermediate results of present or past projects should only be discarded if all concluding documents based upon these data are also discarded.

2. In new projects, conclusions from existing projects can only be copied if also the complete underpinning data that led to those conclusions is included in data for the new project.

Training of employees to work with this form of data lineage management. It was anticipated that people only would accept the new system if it helped them to do their work faster and with less errors. To promote acceptation, the threshold for using the new system was lowered as much as possible by training and by context dependent on-line documentation. In the end these facilities indeed led to acceptation and people handed in a small piece of freedom for the benefits of accountability to the customer and the possibility to understand the motivation for existing designs and learn from them.

Diverse short term development assignments for different customers

Performing quality assessment concerning forecasting software for the Embedded Value of insurance contracts with a large dutch insurance firm. Drawing up an evaluation report with regard to existing software. Advice regarding renovation. Taking part into functional specification and redesign for those parts of the software that were renovated. Performing capacity and performance analysis of the new software. Training developers in Object Oriented Design and programming in Object Pascal.

Developing interface software for magnet strip card readers for Johan Enschedé en Zonen (The company that in that time printed Dutch currency). On specification development of access control software for several customers of this company, a.o. Peekel Productions, the police force, KVGO, Moret Ernst en Young.

Designing and building a GUI for the Heavy Weather Warning System of the Tidal Water Service, a governmental organization.

Designing and building software for calculation and planning of building maintenance and statistical quality assessment. Novum Software.

Designing and building the SoftPlc and Fault Diagnosis Unit ("Black Box") for HMA Power Systems.

Design and building of a grab unloader simulation with visualization. Holec Data Systems.

Drawing up system requirements for an internal transportation system in a steel factory in collaboration with the employees of Techno-Assistance SA for Fabrique de Fer de Mauberge, Temse.

Design and building of image processing software to be used for conference television. Scientific research into use of data compression by means of the two dimensional Fourier Transform, Discrete Cosine Transform and Vector Quantization. Dr. Neher Laboratory, PTT (Dutch Telephone Company).

Providing a management seminar and sequel course, accompanying the move from traditional Cobol to Component Based Development using middleware, Java and IBM Websphere at De Nederlandse Bank (The Dutch central bank).

In-house provision of courses object oriented analysis (OOA, HOOD a.o.), design (OOD, UML a.o.), programming (C++, Python, Java, Object Pascal a.o.),

Component Based Design (Delphi) and Real Time Controls, Design Patterns, Systematical testing in practice at TNO, Delta Lloyd, Ericson, Ordina, CB Direct Insurances, Groningen University, Educom, BSO, Borland Benelux, Gasunie a.o..

Providing for four times the in-company training "Systematical testing in de praktice" at Astra Informatics.

Diverse activities for other companies.