COMPUTER SCIENCE IN THE NETHERLANDS

REFLECTIONS ON THE STATE OF THE ART

THEO D'HONDT
SJOUKE MAUW
WIM VAN PETEGEM

QANU
Catharijnesingel 56
PO Box 8035
3503 RA Utrecht
The Netherlands

Phone: +31 (0) 30 230 3100
E-mail: support@qanu.nl
Internet: www.qanu.nl

Project number: Q0745

CONTENTS

PREFACE **4**

PROCESS IN WRITING THIS REPORT **5**

The NVAO Programme Assessment 5

Working Method of the Assessment Panel for the State-of-the-Art Report 5

Topics for the State-of-the-Art Report 5

COMPUTER SCIENCE IN THE NETHERLANDS: STATE OF THE ART IN 2019 .... **7**

1. Student intake and capacity 7

2. Anticipating new developments in computer science 8

3. Professional skills 10

4. Diversity in student populations and staff 11

5. Innovation in education 12

6. Final project 14

CLOSING REMARKS **15**

APPENDIX: PANEL MEMBERS NVAO ASSESSMENT **16**

PREFACE

Close to a year ago, most of the individuals listed at the end of this document met at QANU's headquarters in Utrecht to get acquainted with one another and with the task lying ahead: forming a panel to assess how computer science is taught at Dutch universities. Three of the panel members authored this state-of-the-art document; it is no coincidence that we were the only ones to visit all—or nearly all—of the ten sites. The only native Dutch member works and lives at the other extremity of the BeNeLux, while the other two are strictly south-of-the-border. This gave us a unique perspective even though all three of us are grounded in teaching variations on computer science.

The reviewing process was clear: for each of the sites a cycle was established, the most fascinating part of which was the actual visit and the interaction with faculty and students. This gave us insight far beyond what resulted from the reading of reports and theses. And not surprisingly, we ended up acquiring a cross-cutting view of computer science programmes in the Netherlands. This document is the result of applying our experiences to the six concerns shared by the ten departments.

This document may have been compiled by three persons, but it is the result of efforts made by many more people. The ten departments with their faculty members, students and alumni are to be lauded for hosting the visits of the panel. They were remarkably open to a process which is after all of an inquisitive nature. As a panel we tried to act as critical friends, and the feeling persists that we succeeded in that. We also thank the many other members of the panel, with special mention for the student representatives: without exception, they participated in a spirit of productivity. Last but not least, we congratulate Peter Hildering and Mark Delmartino for their expert assistance: without their support this would have been a much more difficult undertaking.

Prof. Theo D'Hondt
Panel chair

16 June 2020

PROCESS IN WRITING THIS REPORT

The NVAO Programme Assessment

This state-of-the-art report is a direct product of the NVAO assessment of the computer science programmes in the Netherlands. In 2019, the computer science programmes at both the bachelor's and master's level were assessed at ten universities in the Netherlands (Leiden University, Delft University of Technology, University of Utrecht, Eindhoven University of Technology, Open Universiteit, University of Amsterdam, Vrije Universiteit, Radboud University, University of Groningen and Twente University). This cluster assessment was aimed at reaccreditation of these programmes by the Accreditation Organization of the Netherlands and Flanders (NVAO). A panel of international peer reviewers (see Appendix) was installed by the participating programmes to perform these assessments. These panel assessments have now been finalized, and the findings presented as recommendations to the NVAO.

In addition, the NVAO assessment panel was asked by the programmes to review the state of the art of computer science education. The state of the art report is not part of the NVAO assessment, but an independent assignment for the benefit of the ten participating universities, taking into account the topics provided by the universities.

Working Method of the Assessment Panel for the State-of-the-Art Report

Core panel members Prof. Theo D'Hondt (chair, 10 visits) and Prof. Wim Van Petegem (10 visits) and Prof. Sjouke Mauw (9 visits) agreed to write the state-of-the-art report. The panel was assisted by Peter Hildering, secretary and project manager at QANU.

After the completion of the ten site visits, the QANU project manager approached the participating programmes to provide potential topics for a state-of-the-art report based on the discussions between the panel and programme during the accreditation process. The resulting list of topics was reviewed by Informatica Platform Nederland (IPN) on behalf of all programmes, and summarized into six topics. This set of six topics was presented to the panel by the QANU project manager. The panel agreed to base their State of the Art Report on this set of topics (provided below).

The core panel met on 20 April 2020 to discuss its findings on the six topics. After this meeting, the panel wrote a draft report with chapters on each of the six topics, and discussed this draft in a meeting on 11 May 2020. Afterwards the report was finalized and approved by all three participating panel members on 16 June 2020.

Topics for the State-of-the-Art Report

The programmes requested the panel to reflect on the state of the art of academic computer science education in the Netherlands, focusing on topics that concern the computer science programmes in the Netherlands as a whole. In doing so, the panel was asked to pay attention to the following questions:

1. Compared to five years ago, the student intake in computer science programmes has increased significantly. This will lead to a large increase in the student population throughout the programmes. To provide these students with a high-quality academic education, sufficient staffing of the computer science departments throughout all universities is of great importance. Is there enough capacity on a national level to safeguard the quality of the academic computer science programmes?
2. According to the panel, are there any topics or developments that are currently underappreciated in the curricula of the computer science programmes on a national

level, such as computer security? How could these topics be embedded within the programmes?

3. The professional skills of a computer scientist are important. Is enough attention being paid in the computer science programmes to these skills, and if necessary, how could this be improved?
4. What does the panel think of measures to control the size, composition and diversity of the student populations and staff within the programmes? This concerns balances in national, European and non-European students and staff, male and female students and staff, and inclusivity of the population in general.
5. How could the programmes benefit from new didactic concepts and teaching methods, such as blended learning, MOOCs, project/problem-centered learning, and to what extent could this be expected to lead to improvements in the teaching quality, upscaling with regard to the large student population, reduced teaching load and accessibility? And are there any teaching methods that the panel thinks are essential to academic computer science education (such as a master-apprentice approach), regardless of their teaching load?
6. What differences in size and shape of the final projects of the programmes did the panel detect, and how do these differences relate to the different goals and didactic concepts of the programmes?

In the remaining part of this document, the panel will discuss its findings from the site visits on these topics. Each of the six questions will be treated in a separate chapter.

COMPUTER SCIENCE IN THE NETHERLANDS: STATE OF THE ART IN 2019

1. Student intake and capacity

Relative to five years ago, the intake of computer science students in the Netherlands has significantly increased. This occurred across the board for all programmes assessed by this panel, but with a remarkable variation depending on local circumstances. In the coming years, this will lead to a substantial spike in the student population and hence require local organizations to scramble in order to meet growing needs. The successive waves of additional first-generation students are now starting to impact all bachelor's and master's years, and are certain to require further mitigation in the near future.

Ensuring a sustained high quality of university-level education demands a substantial growth of capacity in both teaching and research, the presence of the latter being crucial to the former in any academic setting. Faced by the inevitable delays —measured in years— needed to identify and recruit new faculty members, programme managements are being forced to develop strategies that cover a longer term than what they are used to. Moreover, they find themselves competing for academic talent with kindred colleagues —both academic and professional, national and international— and this talent is unfortunately extremely scarce. The panel has been able to establish that all programme directors are aware of these facts and are indeed considering measures to strengthen their academic staff.

However, it is not clear to what extent higher-level university management realises the seriousness of the situation. The panel was surprised, for instance, by the Dutch government's recent reluctance to allow programmes that run the risk of being overwhelmed to establish a numerus fixus. An admission process of numerus fixus with mandatory entrance requirements is being considered by several bachelor's programmes, but has been implemented by only 2 of the 10 computer science programmes. This can be attributed to the above-mentioned hesitation at the government level to impose a numerus fixus on programmes whose graduates are in high demand by industry and society. The panel is convinced that a numerus fixus is necessary to prevent significant capacity problems and considers it unlikely that it will lead to an additional shortage of computer science graduates in the Netherlands as a whole, given that a large part of the expansion follows from the increase in international students who are less interested in remaining in the Netherlands after their studies. Furthermore, allowing a numerus fixus in only a select number of universities or programmes gives them an unwanted advantage, as it allows them to select high-level students. The same goes for differences in selection between related programmes at the same university, such as data science or AI, almost all of which currently impose a numerus fixus.

The intake of students is rising in all programmes, but varies strongly with circumstances. The presence or absence of a separate bachelor's or master's programme in Artificial Intelligence or Data Science seems to be the prime reason for this variation. Delft University of Technology offers a bachelor's programme in Computer Engineering ("Technische Informatica") that actually incorporates these separate disciplines, and they have been hit by the strongest increase in student numbers. They have found no other solution than to impose a numerus fixus. Their sister university in Eindhoven, although very similar in nature, seems to have been saved from a similar extreme student population growth by the presence of a bachelor's programme in Data Science. The universities in Amsterdam, Groningen, Nijmegen, and Utrecht offer bachelor's programmes in both Computer Science and Artificial Intelligence. They provide a less clear-cut case, conceivably impacted by the presence of a numerus fixus for Artificial Intelligence (and not for Computer Science). In Amsterdam, the failure to merge the programmes of the two universities has had an unpredictable effect on student populations, and this also obscures the interplay between Computer Science and Artificial Intelligence.

A second driver in student intake has been the switch to English language bachelor's programmes in some universities. This switch has facilitated a substantial influx of international students to the bachelor's degree in Computer Science, at least in the six universities that offer an English language programme. This has undoubtedly been a significant factor for student numbers in the technical universities. However, of the three universities that maintain the bachelor's programme in Dutch (apart from the Open University), there is a marked difference between Utrecht (8% increase over 5 years) on the one hand, and Leiden (54% increase over 5 years) or Amsterdam (UvA) (33% increase over 5 years) on the other. The relative success in Amsterdam is attributed to the attraction of the capital combined with the preference of some students for their native language. The limited interest of female students in Computer Science continues to present a challenge. Even though all programmes make a manifest effort to improve the gender ratio, to the point of mobilizing female faculty members as role models, the results are meagre. Adding insult to injury, all Artificial Intelligence bachelor's programmes boast much higher relative numbers of female students. Internationalization and diversity will be considered in section 4.

Several (but not all) programmes have taken the lead in anticipating growing and evolving student populations. Some, admittedly under extreme pressure such as in Delft, have resorted to innovative and original procedures to manage large numbers of students in a given class. Quantitative information accumulated over time and extrapolated to the future is said to be crucial for the efficient allocation of limited resources, human or otherwise. Most programmes have already proceeded with substantial recruitment initiatives for new staff, although constrained by university-level policies and the significant turn-around time between posting an academic job opening and the successful candidate's start date. It is remarkable in a positive sense how most programmes are impacted by the culture shift in diversity, language and nationality of new faculty members. However, a notable area of concern that needs to be addressed is gender balance, which is not always sufficiently attended to. Another is the involvement of the research faculty in the bachelor's programmes, which was below par in some universities. The need for improvement is, not surprisingly, more pronounced in the Dutch-language programmes.

Growth is manifest in most aspects of the various programmes, and student teaching assistants are one of the less obvious ones, albeit not the least important. The panel noticed that most programmes have trouble scaling up their teaching assistant staff without losing a certain degree of quality. Again, this seems due to a shortage of candidates rather than to a shortage of funding. Another area in which it is impossible to speed up adaptation is infrastructure. In multiple places the panel was shown a building site or even to an empty lot as evidence that the intent was far ahead of the actual realization. It was unable to guess at the interference with other growth programmes such as Artificial Intelligence, but presumes this has an additional impact.

Note that the Open University is not addressed in this section: the panel did however appreciate it offering a university-level programme in Computer Science to an audience whose recruitment follows substantially different rules from those applicable in the other 9 universities.

2. Anticipating new developments in computer science

Based on its overview of the programmes it reviewed, the panel found a number of new developments in computer science that deserve additional attention in the curricula in the coming years. Even though the computer science discipline is based on strong and stable foundations, it has an inherently dynamic nature. New developments are fueled by an interplay of many factors, such as theoretical insights, technological progress, novel use cases, advances in other disciplines and societal developments.

Over the past few years, new developments have quickly found their way into academic Computer Science programmes. One example is the increasing importance of computer security, which has led to security-oriented master's programmes or at least dedicated security courses at various Dutch universities. Security awareness and at least a basic level of security knowledge should be expected from every student who graduates in Computer Science nowadays.

A more recent development concerns the explosion of interest in machine learning. This has quickly become a standard tool for developers and scientists alike, competing with conventional computing paradigms. The market demand for specialists in this domain is high, and scientific developments go fast. Educational programmes will have to follow this trend.

Strongly linked to the development in machine learning is the growing importance of data sciences, including fields like text mining, modelling, visualization and decision-making. Because of the interdisciplinary nature of this field, universities will have to make a choice about whether data sciences will be integrated into the Computer Science programmes or formed into an independent study programme. The former will be most suitable to a more technical approach to the subject, while an interdisciplinary approach would justify a separate programme. This will depend heavily on the local situation, such as embedding in existing research programmes. In any case, Computer Science master's programmes could offer topics related to data sciences, such as neuro-linguistic programming, visualization and enterprise modelling.

A developing field with a potentially high impact is quantum technologies. This field is on the crossroads between physics and computer science. The practical and theoretical consequences of achieving quantum supremacy, which happens when quantum computers can practically solve tasks that were considered intractable for classical computers, are manifold. Computer Science students will have to be aware of these developments, and dedicated master tracks in this direction could be developed. Topics specific for computer science could include quantum algorithms and their complexity, post-quantum cryptography, and quantum-based cryptographic algorithms.

Given the global impact of the corona pandemic that is in effect at the time of writing this document, society will need specialists in all disciplines who can contribute to prevention and mitigation. From the perspective of Computer Science, this will include the fields of Digital Health and Bioinformatics. These fields relate to traditional computer science disciplines, such as modelling, information engineering, robotics, user interfaces, and data analysis. New developments, such as contact tracing and electronic patient data, come with serious privacy concerns. Teaching programmes may consider integrating various aspects in a joint Digital Health master's programme.

In order to properly perform their future role in society, all computer science students should be made aware of the relation between their discipline and society. Societal responsibility should be embedded in the DNA of every computer scientist. This relates not only to ethical aspects of the study and application of information technology, but also to aspects such as sustainability, legal considerations, economic impact, privacy, etc. It will have to find its way into AI-related courses (ethical aspects of AI, explainable AI), security (ethical hacking, privacy), user interfaces (human aspects), and data processing (GDPR).

Computer Science departments will have to regularly reflect on the current contents of their study programs in relation to new developments in computer science and its relation to society. Updates can range from enhancing courses to the development of new, possibly interdisciplinary, study programs. The departments might want to anticipate such developments by hiring researchers in new fields such as those mentioned above.

3. Professional skills

In all programmes, the panel could confirm that some attention was paid to professional skills. In most of them, the learning outcomes specify that the students should acquire professional skills in some way. In all of the self-evaluation reports, the panel found mention of skills such as lifelong learning, critical thinking, presenting, writing and communicating, co-operating and collaborating in teams, managing projects or being able to use a project-based approach, problem-solving, acting ethically, establishing links with the community and/or being sensitive to temporal and social aspects. All of these qualify as professional skills, i.e. necessary skills for newly graduated computer scientists when starting their professional career: they enable them to act as a professional.

The way programmes implement these skills in the curriculum varies. Sometimes explicit courses on professional skills (or a subset of them) are offered, for instance with regard to ethics in computer sciences, presentation skills, or project management. In these cases, the students are assessed by the teacher or course coordinator, who is an expert on this topic. In the majority of cases, however, the development of professional skills is integrated in other courses. This can be in mainstream Computer Science courses, or in project or group work (including the final project). Sometimes some sort of training for a particular skill set is provided, but it can also be the case that students are acquiring them via practice. A variety of assessment formats is used: the course teacher might be responsible, or the course team is expanded to include an expert on the particular skill(s) who performs the assessment with them; professional skills are assessed with a pass/fail or are graded; etc. The panel does not express a preference for one or other modality here. Based on what it has seen in nearly all programmes, it does suggest that professional skills and their development within the curriculum deserve greater visibility. This could be implemented by defining a clear learning pathway on professional skills spanning the whole programme and making its content explicit in each phase of the curriculum. A clear learning pathway promotes transparency for the students and allows them to see where they can expect to be trained for which skill, and enables a curriculum-wide analysis of skills education within a programme. This applies to skills education in general, encompassing academic skills.

A particular professional skill that the panel would like to mention separately is entrepreneurship. It is striking to see how many students in Computer Science are either professionally active in the field or even running their own company parallel to their studies. Although the panel appreciates this entrepreneurial spirit among students, it points out possible drawbacks for study efficiency: many of these students take longer (much longer) than expected to finish their studies. It is a matter of finding the right balance here. Most probably this will require tailor-made solutions taking into account individual circumstances, although a more universal policy for student-entrepreneurs might be helpful as well (at the level of the university or even at a government level).

The panel also wants to address the perceived finality of a bachelor's programme in Computer Science: should it prepare students for a seamless continuation of their studies into a master's programme, or should it be a self-contained programme that directly leads to the labour market, which badly needs computer scientists? Here an interesting tension arises when it comes to professional skills versus academic skills: to what extent should students be trained in both of them over the course of the bachelor's and master's programmes? The panel cannot give a clear-cut answer: some programmes try to combine both ambitions, but there are also programmes that mainly focus on the academic approach, especially in the more fundamentally research-focused and technical universities. In order to get a clearer view on this question, the programmes need more and better data on their alumni. In too many cases the programmes were not able to provide quantitative data about where their bachelor graduates end up: are they (self-)employed, or do they continue with a master's programme, and if so, do they proceed in Computer Science or in other directions? The situation is slightly better with data on the whereabouts of alumni from the master's programmes, but even there, the data are not always conclusive. The panel strongly recommends that the programmes increase their efforts to keep track of their alumni and assemble sufficient and insightful data on a regular basis as an instrument in their quality assurance cycle, and also for profiling purposes.

Alumni might also be an interesting resource for the programmes in a totally different way as well because they form direct links to industry or society at large. The panel did see alumni appearing as work field representatives on advisory boards, as mentors (sometimes co-supervisors) of final projects, as role models in orientation activities, even as teachers of specific courses. It welcomes these ways of collaboration and observes significant potential in further enhancing and strengthening them: this provides ample opportunity for students to develop their own skills when exposed to peers who are successful in their professional careers. Furthermore, involving alumni in select, and possibly newly created, lecture material could certainly alleviate the shortage of staff and expertise.

On the other hand, the panel also heard the voice of some students for whom contact with the outside world should be more in proportion to the contact with the academic research in their own institution. Again, a delicate balance needs to be found. According to the panel's observations, the programmes generally succeed in that endeavour, although here and there it identified points requiring improvement, such as teachers from industry without a PhD or teaching qualifications and no involvement in research, but with full academic responsibilities. Admittedly, this was often the case when a pragmatic, temporary solution was needed to accommodate growing student numbers. All in all, the panel supports continuing all efforts to carefully involve alumni and the professional field at large in the programme.

As a last remark, the panel noticed the role study associations ('studieverenigingen') often have in contributing to the development of professional skills, for instance in the organization of orientation activities, guest lectures, business events, mini-symposia, scientific conferences, etc. That should not necessarily come as a surprise after reading the above observations. Computer Science students are indeed quite entrepreneurial, and conversely the work field badly needs them, so it should be easy to establish good relationships. The panel can only applaud this co-operation, which is definitely beneficial to both sides. However, it encountered some study associations that were also involved in organizing specific courses or professional skills training. It questions whether this should be their sole responsibility. It recommends that the programmes engage the study associations in a dialogue in order to recognize and share responsibilities with regard to professional skills. Having said this, it appreciates the professional behaviour of the students in the study associations and is impressed by the level achieved in their professional skills.

4. Diversity in student populations and staff

The explosive increase in the number of computer science students within the Netherlands is putting a strain on the study programmes, especially in relation to the teachers' workload and the availability of sufficient infrastructure. However, this increase is also producing a qualitative change in the student population. The increasing inflow of non-Dutch and non-EU students leads to an increasing variety in the students' cultural and educational background. On the one hand, this enriches the student population as a whole, but on the other hand it implies a number of challenges for the study programmes.

An example of such a challenge mentioned by some programmes during the site visits is the decreasing cohesion between the students, sometimes even observed as a segregation of various groups (e.g. Dutch, EU, non-EU students). As a consequence, some students did not feel at home or that they were treated equally. This also limited the opportunities for the students to optimally benefit from the multicultural learning environment. Related to this is the diversity in language skills, which is sometimes expressed in the form of communication and cooperation problems. Another challenge is the variety in prior knowledge and skills of students from different educational backgrounds. In practice, this requires some students to spend a lot of time catching up, while others feel they lose time because the teachers adopt a slower pace. These problems were especially observed by

Computer Science master students in programmes that admitted students from non-Computer Science bachelor's programmes.

The universities have implemented a variety of measures to prevent or mitigate such challenges. We will briefly discuss some examples below.

In general, the universities have developed a clear profile and consistently communicate this to prospective students. During meetings with the students at the different universities, it became obvious that most of them had made a well-motivated choice for their current study programme. They seemed to be well aware of the differences between the various Computer Science programmes in the Netherlands. Differentiating factors mentioned by them include the fundamental vs. applied orientation of the programme, or whether the programme specializes in a sub-topic, such as security. Other factors, such as the atmosphere of the campus and the scale of the programme were also mentioned.

Even though most Computer Science programmes in the Netherlands are taught in English, there is still a market for Computer Science bachelor's programmes in Dutch (e.g. at the University of Amsterdam or Utrecht). A university's choice to offer such a bachelor's programme will not only result in fewer students – and thus fewer capacity problems – but also in a more homogeneous student population.

Technical disciplines in the Netherlands are still male-dominated. Computer Science departments are well aware of this lack of balance and have developed various strategies to attract more young women to technical studies. For example, sending female role models to schools or stressing algorithmic thinking (University of Utrecht). Most universities have university-wide policies to increase the number of female staff. An example is the Eindhoven University of Technology, which also observed an increase in the number of female students due to the development of the Bachelor College approach. Even though there is progress, the developments are proceeding slowly. It is worth noticing that in some specialized programmes, such as for instance AI, Media Technology (Leiden) or Interaction Technology (Twente), the gender ratio is better balanced. An interesting observation is that the increased international student and staff inflow contributes positively to the gender balance.

In order to mitigate the risk of a decreased cohesion among the student population, some universities have proposed language policies to prevent foreigners feeling excluded in a context with native Dutch speakers. Other universities have developed procedures for clustering students of diverse backgrounds for group projects.

Double-degree programmes and student mobility can contribute strongly to internationalization. Surprisingly, international mobility during the study (e.g. through the Erasmus program) is still limited, even though the option is presented frequently to the students.

5. Innovation in education

The majority of courses within the Computer Science programmes support a very traditional way of teaching and learning: lectures given by professors, accompanied by tutorials, project work or practical sessions guided by teaching assistants or PhD researchers. A decade ago this was a suitable model for a Computer Science programme: given the lower number of students at the time, a very personal, almost one-to-one relationship between teachers and students was established. This fruitful collaboration frequently led to a scientific paper covering the students' research (illustrating the high quality in terms of academic outcome of the programmes), or it continued into a PhD research project. In the current situation, however, this model has come under pressure, as the increase in student numbers is not yet matched by an expansion of academic staff. This strains the possibilities for an individual relationship between the students and teachers. Students tend to praise this model, however, and cherish the accessibility of their teachers very much. Of course, they also see the difficulty in keeping up this sort of apprenticeship, and they appreciate all efforts done by the current academic staff to cope with the situation. The panel recognizes that in most programmes several measures have been taken to hire new people with high research competence as well as excellent teaching expertise – not an easy task, as the appealing conditions for computer scientists in the corporate world attract these potential academics as well. It also saw creative approaches, such as involving guest lecturers from outside academia, co-teaching practices (e.g. teaching teams of senior and junior lecturers), and so on. It should also be mentioned that this problem is not limited to Computer Science: in most cases it is a faculty-wide problem. The panel recommends considering very carefully how available resources at that level can be used in a flexible and sustainable way, while keeping in mind a student-centered approach in teaching and learning.

In times of growing student numbers, digital technology can help as well to mitigate the workload of the teaching staff and enable them to re-establish more personal contact with their students. However, the panel was disappointed to see that precisely in the field of Computer Science, very few innovative teaching methods exploiting the potential of digital technology are applied. On the contrary, it sometimes observed hesitation and actually skepticism in both teachers and students with regard to using learning technologies. In view of current trends in society (where digital technology has become a commodity for all) and the rapid rise of distance education fueled by the global corona pandemic mentioned above, the programmes should consider how distance education, online (or blended) learning and flipped classroom methods may be introduced more structurally into the programmes. Of course, boundary conditions for the sound pedagogical use of learning technologies in the courses should be taken into account, like appropriate infrastructure, proper instructional design, adequate support, continuous professionalization of teaching staff and equal access to all. Co-operation with the educational development units at the departmental, faculty or even university level will be of the utmost importance to guarantee the required quality level of education, sufficient efficiency and effectiveness of the measures, and a suitable acceptance rate by teachers and students. These are issues that can be solved: if anywhere, learning technologies should work for Computer Science programmes. And the OU already paved the way: being a distance teaching university, it applies proper digital teaching and learning methodologies in all of its programmes, including its computer science programmes. It combines online with face-to-face education centred around an in-house built learning and working environment, called yOUlearn. This environment offers course-specific sites, with all course contents and exercise materials, but it also has features and tools for course structuring, discussion groups, news, sharing of materials, uploading of assignments, contacts with lecturers, virtual classrooms, collaboration work, information about the programmes, etc. Teachers are constantly informed about these features and can get support to implement the tools when they see an added value in them to make their course more activating. Students also get support via a dedicated set of online services, including access to tailor-made information on their study and study progress, and access to study coach pages where they can learn to learn online. The study centres all over the country serve as physical spaces where students can take exams, and where face-to-face coaching activities can take place. It is worthwhile examining how these blended learning modalities set up in the context of a distance teaching university like the OU can be inspiring for the more traditional on-campus universities.

One particular aspect in the further digitization of education is automatic software code evaluation and the prevention of plagiarism. The latter is not only concerned with the traditional way of checking original work in written reports (which is common academic practice nowadays), but also with the specific issue of software code 'sharing' or 'copying'. The panel noticed that compared to the previous quality evaluation round, much more attention is now paid to this topic, a move which it can only applaud. A variety of solutions has been proposed: commercial and proprietary-based, open source and home-made systems are in use or under development. The panel is not in a position to assess the suitability of these tools, although it learnt from the students that they are useful. One recommendation it would like to make is that it might be fruitful to set up an interuniversity collaboration to share ideas, or even complete solutions. Not only will programmes in computer science benefit, but a broader application of the tool(s) to other fields and programmes might be envisioned, making it worthwhile to allocate resources to such an initiative.

A particular asset of the above-mentioned tools is that they provide the students with immediate feedback. The students very much appreciate this feature. In fact, providing and getting feedback is generally an important aspect of any learning activity. In the apprenticeship model of the past, this was inherent to the relationship between teachers and students. In the current situation, however, software tools might take over from basic forms of feedback, but the students will still want some human interaction when it comes to more complex issues. Finding the proper balance between automatically generated feedback and interpersonal guidance is a challenge, but not unfeasible in the field of Computer Sciences where competences are available to implement both successfully.

Concerning feedback, the panel would also like to emphasize that the students showed a great eagerness and maturity to provide feedback themselves on the programme they are part of. The panel discovered an admirable willingness to engage in course evaluations (formative and summative), to discuss general and topical programme issues in sounding board groups for instance, and to participate actively in decision-making processes. Needless to say, the study associations are very instrumental in this respect: they can count on the spontaneous efforts of highly motivated students with sympathy for education in Computer Science, who in turn can motivate their peers in voicing their opinions and concerns. The panel has seen nice examples of how this leads to a permanent dialogue between the students and teachers on improving the quality of teaching and learning in their programmes. Sometimes it was not clear to students how the programme dealt with their feedback. This is not simple, as it affects the next cohort of students: the ones who take the next, improved edition of the course. Nevertheless, the students deserve to see that their feedback is taken seriously and how the programme closes the loop of quality assurance in teaching and learning. All actors, individual teachers, programme management and students (whether or not via the study associations) have a role to play here. And they do.

6. Final project

The panel saw many examples of final projects, for both the bachelor's and the master's degrees. It is therefore in a position to note differences between the programmes, and to check them against the intended didactic methodology and the related learning objectives. The panel observed that in all cases the master's project was essentially research oriented, while the bachelor's project left much more leeway for a professional approach. In a number of cases, the latter consisted of a software engineering project, involving as many as 10 students, sometimes even directly linked to an assignment from an outside agency. It should be noted that the research content was always guaranteed, if necessary by assignments separate from the project.

In the assessment of the final projects and looking at the accompanying assessment forms, the panel frequently encountered a lack of tangible evidence for the actual assessment of the final projects. At the bachelor's level, deliverables are frequently — but not always — limited to design documents, program code and software documentation, while the learning outcomes are much broader. A more extensive rationale accompanying the assessment would be helpful for an outside evaluation, such as the one conducted by the panel. In group projects, the final grade for each individual student is often derived from the overall team assessment with a small variation based on peer review within the team. The panel did not encounter even one instance where this led to abuse. Whether this is an intentional result of the assessment model could not be established. The panel found that at the master's level, the dominant part of the assessment — up to 70% — related to interactions between the supervisors and the student. This was reflected in an evaluation form which, despite frequently being well conceived, was often all too poorly completed by the project assessment committee. The panel used the project thesis — often the single residue — to approximate the quality of the complete project. It has to be said that most of these were adequately graded, and were frequently of an impressive quality, with not a few of them leading to peer-reviewed publications. This also holds for bachelor's projects executed in a more research-oriented vein. Unfortunately, the panel noted that encouraging top students to aspire for this additional recognition often came at the expense of the duration of their studies.

In a number of cases, the panel detected a shortage of attention paid to aspects such as ethics, the academic method or even the academic style of writing. Because of the lack of information mentioned above, it was not possible to determine whether the cause was to be sought with the student, with the supervision or even with the programme. Computer science is a very technical domain, and a project process is often well established at the start, normally at the supervisor's initiative. It seemed difficult however, to avoid having the human aspects diverge from the technical ones. In this respect, it occurred to the panel that the Examination Committee could often act more proactively: in addition to establishing the rules, it could also supervise their implementation.

Another task for an examination committee, which seems somewhat neglected, is the consistent composition and instruction of final project reviewers. It is in the interest of all concerned parties that reviewers make explicit how they collaborate to produce the final assessment and that they leave a detailed and public trace of their deliberations. Evaluation forms are mostly adequate as a starting point, but they only serve to structure and quantify an assessment process that deserves a qualified judgement in writing by each and every member of the committee.

The panel noted that, for instance, Twente and Eindhoven interact to benchmark the quality of their project assessment. This can be viewed as an extremely meaningful tool for establishing a uniform assessment process across the board. It is certainly open to generalization at a national level, and the panel recommends this.

During the site visit, several projects were assessed that were part of a double-degree programme. These projects were either conducted by students from a Dutch programme at a non-Dutch university in the context of a European agreement, or vice versa. The panel found that sometimes fail-safes were missing and that some programmes had no or only limited guarantees in place for an adequate assessment of final projects according to their own guidelines. An overreaching set of procedures would be beneficial.

Finally, during the course of the site visits, the panel determined that between 10% and 15% of master's graduates continued with a PhD. Most master's programmes are decidedly conceived in a spirit of research, as is often the practice. The curricula are designed to expose students to research in an academic or professional setting, as they should be. They are biased towards the former, however, often leaving limited opening for apprenticeships or final projects conducted in coordination with outside professional agents.

CLOSING REMARKS

With this report, the panel hopes to contribute to the quality cycle of the Computer Science programmes in the Netherlands. For further reflections, it refers to the reports of the development dialogues of the individual site visits, which provide insight into the issues for future development that were discussed with the individual programmes. The panel is confident that the programmes will be able to develop their quality further over the coming years, and it will enjoy seeing the results in six years' time.

APPENDIX: PANEL MEMBERS NVAO ASSESSMENT

- Em. prof. dr. T. (Theo) D'Hondt, emeritus professor in Software Languages and Software Engineering at the Faculty of Sciences and Bioengineering Sciences of Vrije Universiteit Brussel (Belgium) [chair];
- Prof. dr. ir. W.E.A. (Wim) Van Petegem, professor and policy coordinator Learning Technologies at the Faculty of Industrial Engineering Technology of KU Leuven (Belgium);
- Prof. dr. S. (Sjouke) Mauw, professor in Security and Trust of Software Systems at the Department of Computer Science of the University of Luxembourg (Luxembourg);
- Prof. dr. J.J. (John-Jules) Meyer, full professor Computer Science and Artificial Intelligence at the University of Utrecht;
- Drs. L. (Lennart) Herlaar, owner/director at Redbits.nl, a company specialised in software development and IT consultancy, and assistant professor Computer Science at the Faculty of Science of Utrecht University;
- A. (Antonia) Wildvank, owner/CEO at Wildvank, Management en Advies, specialised in IT-management and -consultancy;
- Prof. dr. ir. J. (Jan) Aerts, full professor Visual Data Analysis at the University of Hasselt and associate professor Visual Data Analysis at the faculty of Engineering Science at KU Leuven (Belgium);
- Drs. H.C. (Jeroen) Borst, senior consultant Smart Cities at TNO;
- Prof. dr. P. (Petros) Koumoutsakos, full professor Computational Science at ETH Zürich (Switzerland);
- Prof. dr. ir. J.M.W. (Joost) Visser, Chief Product Officer at Software Improvement Group (SIG) Nederland and professor Large-scale Software Systems at Radboud University;
- Drs. E.A.P. (Ewine) Smits, Manager in Advanced Analytics & Big Data at KPMG Nederland;
- Prof. dr. D.P. (Danilo) Mandic, full professor Signal Processing at the department of Electrical and Electronic Engineering of Imperial College London (United Kingdom);
- Dr. ir. J.C. (Job) Oostveen, Research Manager at the Department Monitoring and Control Services at TNO;
- Prof. dr. B.A.M. (Ben) Schouten, full professor Playful Interactions at Eindhoven University of Technology;
- Dr. ir. N. (Nico) Plat, owner/CEO at Thanos IT-consultancy and architecture;
- N. (Nienke) Wessel BSc, master's student Computing Science and bachelor's student Mathematics and Linguistics at Radboud University [student member];
- E. (Evi) Sijben BSc, master's student Computing Science in the specialisation track Data Science at Radboud University [student member];
- B. (Baran) Erdogan, third-year bachelor's student Computer Science at University of Amsterdam [student member];
- M. (Martijn) Brehm, third-year bachelor's student Computer Science at University of Amsterdam [student member].