



STUDIJŲ KOKYBĖS VERTINIMO CENTRAS

VILNIAUS UNIVERSITETO
INFORMATIKOS STUDIJŲ PROGRAMOS (612I10002)
VERTINIMO IŠVADOS

EVALUATION REPORT
OF *INFORMATICS* (612I10002)
STUDY PROGRAMME
at VILNIUS UNIVERSITY

Grupės vadovas: Team leader:	Prof. Jukka Paakki
Grupės nariai: Team members:	Prof. Rolf Backofen
	Prof. Jerzy Marcinkowski
	Vida Juozapavičienė
	Lukas Jokūbas Jakubauskas

Išvados parengtos anglų kalba
Report language - English

DUOMENYS APIE ĮVERTINTĄ PROGRAMĄ

Studijų programos pavadinimas	<i>Informatika</i>
Valstybinis kodas	612I10002
Studijų sritis	Fizinių mokslų studijų sritis
Studijų kryptis	Informatika
Studijų programos rūšis	Universitetinės studijos
Studijų pakopa	Pirmoji
Studijų forma (trukmė metais)	Nuolatinė (4m.)
Studijų programos apimtis kreditais	240 ECTS
Suteikiamas laipsnis ir (ar) profesinė kvalifikacija	Informatikos bakalauras
Studijų programos įregistravimo data	Lietuvos Respublikos švietimo ir mokslo ministro 1997 m. gegužės 19 d. įsakymu Nr. 565

INFORMATION ON EVALUATED STUDY PROGRAMME

Title of the study programme	<i>Informatics</i>
State code	612I10002
Study area	Physical Sciences
Study field	Informatics
Kind of the study programme	University Studies
Study cycle	First
Study mode (length in years)	Full-time (4years)
Volume of the study programme in credits	240 ECTS
Degree and (or) professional qualifications awarded	Bachelor of Informatics
Date of registration of the study programme	19 of May 1997, under the order of the Minister of the Ministry of Education and Science of the Republic of Lithuania No.565

©	Studijų kokybės vertinimo centras
	The Centre for Quality Assessment in Higher Education

CONTENTS

CONTENTS	3
I. INTRODUCTION	4
II. PROGRAMME ANALYSIS	5
1. Programme aims and learning outcomes	5
2. Curriculum design	6
3. Staff	11
4. Facilities and learning resources	13
5. Study process and student assessment.....	14
6. Programme management	15
III. RECOMMENDATIONS	16
IV. SUMMARY	17
V. GENERAL ASSESSMENT	18

I. INTRODUCTION

The subject of this evaluation is an undergraduate curriculum in *Informatics* taught in the Department of Computer Science, Faculty of Mathematics and Informatics, Vilnius University. It is one of nine Bachelor study programmes offered by this Faculty (this number including other curricula in Computer Sciences – Software *Engineering* and *Bioinformatics*).

The curriculum is 8 semesters (4 years, 240 ECTS credit points). The degree awarded is "Bachelor of Informatics". The *Informatics* study programme splits, after four semesters, into two tracks, *Computer Science* and *Mathematical Computer Science*. Only the full-time mode of studies is offered.

The previous external assessment of this study programme, by an international group of experts, took place in 2006. The result of the assessment was positive, and the reviewing panel concluded that they “did not explore any major problems in the study program”.

The current procedure of the external evaluation of Vilnius University first-cycle study programme *Informatics* was initiated by the Centre for Quality Assessment in Higher Education of Lithuania which selected and appointed the external international evaluation Review Panel consisting of its Chair, professor Jukka Paakki (University of Helsinki, Finland), professor Rolf Backofen (University of Freiburg, Germany), professor Jerzy Marcinkowski (University of Wrocław, Poland), Vida Juozapavičienė (employer representative – social partner, Lithuania), and Lukas Jokūbas Jakubauskas (student representative – Lithuania).

For the evaluation, the following documents have been taken into account:

1. Law on Higher Education and Research of Republic of Lithuania;
2. Procedure of the External Evaluation and Accreditation of Study Programmes;
3. General Requirements of the First Degree and Integrated Study Programmes;
4. Methodology for Evaluation of Higher Education Study Programmes.

The basis for the evaluation of the study programme is the Self-Evaluation Report (referred to as the SER) prepared in 2013, its annexes and the site visit of the Review Panel to Vilnius University on November 27th, 2013. The visit included meetings with different groups: the administrative staff of the faculty, the staff responsible for preparing the self-evaluation documents, teaching staff, students and social partners. The Review Panel evaluated various support services (classrooms, laboratories, library, computer facilities), examined a sample of students' work, and various other materials. We also visited

some actual classes. At the end of the visit preliminary general conclusions of the visit were presented to the Head of Department teaching the study programme. After the visit, the Review Panel met to discuss and agree the content of their final report, which represents the agreed views of the Review Panel.

The Review Panel was truly impressed by the fact that no translator was needed during the visit. All the meetings were held in English and all the staff we met and all students we had an opportunity to talk to were fluent in English.

II. PROGRAMME ANALYSIS

1. Programme aims and learning outcomes

The understanding the Review Panel has after reading the Self-Evaluation Report (SER), analyzing the study programme after talking to all the interested groups (the people responsible for the programme, to the teaching staff, students and social partners) is that the philosophy of the study programme is consistent with the tradition of computer science studies offered by mathematical departments, with strong emphasis on fundamental theoretical subject and on understanding of the basic concepts, and – in consequence – with technology subjects receiving relatively less time than it would be normal in the tradition of technical studies. This model has proved worldwide to produce graduates who are attractive for the labour market, able to follow the fast evolution of technologies and capable not only to produce code but also to think and solve problems.

This philosophy is correctly reflected in the name of the programme and uniformly understood by all the interested parties (students, employers, the Department) but not always correctly reflected in the official definition of the programme aims, which are: *“to prepare highly-qualified broadly-educated computer scientists able to design, create and maintain the software:*

- *To build formal information models of various application areas or use the existing ones for the achievement of the predefined goals;*
- *To implement a software project independently or in a (multicultural) group by modern software development tools and technologies;*
- *To evaluate the software system from the points of view of usability, correctness, performance and security; ;*
- *To install, maintain and update skillfully the hardware and software.”*

This is much too generic, and while being so generic still not really consistent with the programme learning outcomes and the curriculum design which both – correctly – put a lot of weight on the fundamental computer science subjects and (relatively) less on the development tools and technologies. Unlike the programme aims the learning outcomes of the programme are relatively clear and consistent with the curriculum design (albeit still not specific enough to be really measurable). The document “*Study Programme Description*” identifies three “*General competences*” and eight “*Subject competences*” that jointly constitute the expected outcome of the study programme.

Information about the Programme is available on the Web, on the website of the Department (<http://mif.vu.lt/lt2/studijos/programos/studiju-programos/bakalauro-studijos/informatika>).

2. Curriculum design

The *Informatics* curriculum (both tracks) is in computer science, and this is reflected in the way it is composed. About 15% of subjects (measured by the ECTS points) are mathematical subjects (including probability and statistics, mathematical modeling and mathematical logic). About 62 % are courses in computer science subjects (evenly divided between theoretical and technological subjects), about 15 % are projects, practices and preparation of the Bachelor thesis and the rest are the courses in English and in informatics law. This design meets the Lithuanian legal requirements.

The study programme splits, after four semesters, into two tracks, *Computer Science* and *Mathematical Computer Science*. But otherwise the programme is very rigid: there are almost no optional subjects (total of 20 ECTS credit points, not including the “General University Subjects”). This may be only partially explained by the Lithuanian legal regulations, which set an upper limit on the volume of the optional subjects in the curriculum, as the legal upper limit is 25% of the courses (measured by the ECTS credit points), while in the analysed programme only about 8% (or 14%, if we include the “General University Subjects”) of modules are optional. This design choice, while being a standard one in Lithuania, is very unfortunate, because:

- Leaving the students the possibility of choosing their courses turns them into active participants of the process. Choosing courses is an important exercise in taking responsibility for one's decisions and so by itself contributes to an important learning outcome.
- A curriculum with a lot of electable courses can in a natural way evolve, following the very fast development of this particular field, and can easily accommodate courses which for some reasons can be offered occasionally, or even once, but not on regular every year basis, for example guest courses given by visiting teachers or by industrial partners.

In the current structure of the CS/IT curricula offered by the Faculty, with several related rigid bachelor (BA) programmes (which then split again), the decision which curriculum to choose needs to be made by a student before he/she is really prepared to make it, which is objectively unnecessary as the differences between the programmes of the first two years of studies are not always justified by the different programme aims of the curricula.

It follows from the discussions the Reviewing Panel had, that all the interested parties apart from the department leaders and some of the teachers would like to see much more choice and flexibility in the programme. Some of the social partners for example, declare that they would be interested to offer new courses which, we understand, would be very attractive for students. But obviously such courses will not be offered every year, and in the current structure they cannot be built into the study programme. Also the students expressed a very clear view that rigidity is one of the main drawbacks of the curriculum and some of them made the huge effort to overcome the unnecessary bureaucratic barriers and individualize their study programme.

The view of the Department leaders, as far as we understand it, is that teaching a curriculum with a lot of choice would be too expensive compared to the limited resources of the Department, as it would inevitably lead to teaching in very small groups (actually – as we learned from the alumni – the real choice the students have is sometimes even smaller than it would follow from the programme description, as the Department not always can really offer all the promised electable courses). The Reviewing Panel does not share this point of view.

Apart from the structural problem described above, there are three other major issues concerning the curriculum design:

1. One of the recommendations of the previous assessment, that took place in 2006, was replacing some of the courses in mathematics and physics by courses on fundamental subjects of informatics, such as algorithms, complexity theory, formal languages and automata theory. This was done, but there is a strong impression of a lack of reflection concerning the learning outcomes of the new courses. For example:

– **Algorithm Theory.** Concepts of mainly historical importance, like primitive recursive functions or Post canonical system are covered, which do not contribute significantly to the understanding of any modern ideas. Also, for some reason, three different proof theoretic paradigms are taught: Hilbert-style calculus for propositional logic, sequent calculi, and resolution method. The only reason to teach any

proof theory at an undergraduate level is to illustrate the fact that the concept of proof is formalized, but clearly one paradigm is enough for that (one reason to teach about resolution could be its application in logic programming, but **Logic Programming** is only an optional subject). On the other hand, the notions which are cornerstones of the worldview that should be a result of a solid theory/algorithms oriented Computer Science (CS) education, which are Rice theorem (saying that there are severe limitations on the possibility of automatic verification of programs), or Cook-Levin theorem, are not present in the curriculum. Also, while a lot of time is devoted for courses in theoretical computer science, still no combinatorial undecidable problems seem to be shown.

– **Mathematical Logic.** Again, topics of marginal importance are covered, which could maybe find their place at specialized seminars, but are useless as a part of undergraduate curriculum. The examples include: *Sequent predicate logic calculus. Minus-normal calculus. Intuitionistic logic. Semantic tableaux method. Sequent calculi of modal logics K and S4. Equivalent formulas. Mints theorem. Tableaux method for modal logics K and S4. Classification of temporal logics.* It is not at all clear whether the importance of modal logics justifies having them as an obligatory topic in an undergraduate CS curriculum, but even if one thinks that they are important then one such logic should be selected and taught. Obviously, none of the learning outcomes of the *Informatics* curriculum requires that the student can understand a difference between logics K and S4. Also, none of the learning outcomes requires teaching about proof systems for modal logic – this is a topic known probably by just a handful of people worldwide, and the understanding of the Reviewing Panel is that it was an initiative of one of the teachers to include his own research topics in the syllabus. Not only this is rarely correct to do so, in the context of an undergraduate curriculum, but also the research topics themselves are quite far from being central for modern logic or theoretical computer science. The time wasted for the numerous irrelevant – from the point of view of the learning goals – topics could be spent for example for showing some real applications of modal logics in computer science, for example by playing with some LTL model checker.

– **Cryptography and Information Security** course is taught during the 5th semester. It covers, among other things: *Public key cryptography: encryption and digital signature schemes. Knapsack, RSA, Rabin, ElGamal cryptosystems, cryptanalysis of special cases. Digital signature schemes: RSA, ElGamal, DSS, Rabin. Security issues.* But the course **Algorithm Analysis** is taught during 6th semester, and it is only then when some basic complexity theory notions (including the notion of NP) are introduced, which are necessary if mathematical ideas behind the public key cryptography are to be seriously talked about.

– **Automata and Formal Languages course (5th semester)**. Half of the semester is spent for teaching about formal grammars. The only reason to devote so much time to this topic would be to use this later for the Compiling Methods course. This was actually the original reason why formal grammars found their way into the CS curricula. But Compiling **Methods** is a course for another track (and it probably should not be obligatory anyway – the topic was central for CS in the 1960s, possibly also important in the 1970s but since then it has faded in importance).

– **Algebra and Geometry** course (1st and 2nd semesters). The notions of: “*Vector spaces: basis and dimension, transition matrices; subspaces: spanning, sum and intersection of subspaces; matrices of linear transformations, kernel and image*” come in the end of the second semester, after “*systems of linear equations and matrices: Gaussian elimination, Kronecker-Capelli’s theorem Cramer’s rules, matrix operations, rules of matrix arithmetic, invertible, triangular, symmetric matrices.*” This order makes just no sense. Another question is what are the advanced notions of linear algebra (except of the aforementioned, also the notions of Eigenvalues and eigenvectors are included) good for in CS curriculum. On the other hand the abstract algebra notion of group seems to be barely mentioned and the notion of field is not mentioned at all. This is wrong, as some abstract algebra background is needed for many computer science applications.

– **Error correcting codes**. The syllabus includes: “*linear algebra over the finite fields. The linear codes. Generating and control matrices. The decoding algorithms – 6 hours*”. How is it possible to teach that in 6 hours to people who have no background in abstract algebra? As we already observed, the only abstract algebra notion mentioned during the Algebra **and Geometry** course (1st and 2nd semester) is the notion of group. There is nothing about fields, in particular anything at all about finite fields.

It is also not clear to which of the learning outcomes the module **Chaos theory and fractals** is supposed to contribute.

2. The most important learning outcome of a curriculum in the tradition of mathematical computer science studies are the of “problem solving” competences. In the case of the evaluated program they are defined as "General competences 1 and 2": “*Ability to analyze and organize the information. Ability to apply the knowledge in practice.*” The possibility of achieving these learning outcomes depends not that much on the contents of the courses as on the way the courses – in particular the mathematical courses – are taught. And in particular it depends on the way the "exercises" (the teaching form that is not a "lecture") are used. The teacher (preferably the lecturer) should, after each lecture, prepare for the students a list of tasks to solve at home. They should know this list some days in advance, and

measures should be taken that when they come to the "exercises" they are prepared, which means that each of the students can solve at least some of the tasks. Then the "exercises" should be "an active form of teaching" – the students should present their solutions on the blackboard, preferably without any help from the teacher, who should only, from time to time, comment on the solutions.

The way mathematical subjects are taught in the institution we visited is very much different from the described above. The Reviewing Panel visited some "exercises", talked a lot to students about the way the "exercises" are organized, and concludes that they are just another form of lecture: it is mainly the teacher who is speaking. The only difference between lecture and "exercises" is contents rather than form: during lectures "theory" is taught, while during "exercises" the teacher shows the students how to solve simple tasks. Sometimes, after giving a couple of examples, he invites one of the students to the blackboard and asks him to solve another one, which does not differ too much from the previous ones. But this is only about repeating examples and hardly constitutes an active form of teaching.

The above view of the way the "exercises" are organized was confirmed by the teachers during the meeting the Reviewing Panel had with the teaching staff.

It needs to be added here that the above critical remarks do not apply to classes organized in the form of computer labs. The Reviewing Panel visited also some of such classes and the overall impression was good. In particular we appreciate the effort made by the teacher teaching the labs of the **Algorithms and Data Structures** course.

3. There is a huge amount of overlap between the study courses. For example the syllabus of the course **Informatics fundamentals I** (1st semester) contains:

Sorting. Insertion sort, Bubble sort, Heapsort, Quicksort. Searching. Linear and binary search. Hash method. Collisions. Linear probing method.

Syllabus of the course **Informatics fundamentals II** (2nd semester) contains:

Graphs. Internal representation. Neighbourship matrix, relation lists. The shortest path problem for the weighted graph. Dijkstra 's algorithm. Analysis and evaluation. Internal and external sorting. External merge method. Analysis and evaluation of complexity. Balanced trees. Balance criterion. AVL trees. Inserting in AVL tree. Basic transformations (left and right simple and double rotations). Searching in AVL tree. (...) Two linked lists. Decks. Trees, binary search trees. Basic operations.

And syllabus of the course **Data Structures and Algorithms** (3rd semester) contains:

Sorting, internal sorting, quicksort Merge sort, von Neuman algorithms, external sorting, formalism and procedures Abstract data types, stack, queue, examples, programming of stack and queue by using array and pointer structures binary search trees (...) AVL trees (...).

This means that up to half of the Data **Structures and Algorithms** course is just a repetition of earlier courses. Some of the topics taught during the Informatics **fundamentals II** are also repeated at the Combinatorics **and graph theory** course.

The above list is not exhaustive – more examples of notions which are introduced twice or more are: Euler paths (**Graph theory, Discrete mathematics, Algorithm analysis**), finite automata (**Discrete mathematics, Algorithm Theory**). Kruskal and Prim algorithms (**Graph theory, Algorithm analysis**). An absolutely basic Dijkstra shortest path algorithm is taught four times (**Informatics fundamentals II, Graph theory, Algorithm analysis, Artificial Intelligence**).

The ubiquity of overlap in the curriculum (or at least in its theoretical part) was confirmed by the students. One of the examples that the Reviewing Panel learned about during the meeting with the students is the **Artificial Intelligence** course which, according to the students is "80 % a repetition". Also the teachers themselves did not deny that the curriculum is full of overlap, and saw it as unavoidable. "I know that this algorithm was taught about before" – one of the teachers told us – "but I need to present it again, because I want to be sure that the way it is explained is detailed enough".

Apart from the points 1-3 above one minor issue is that not all the learning outcomes listed as "*Subject competences*" can be achieved in the current curriculum. For example, no functional programming language is covered by the courses syllabi of one of the tracks (and none is really taught – as the Reviewing Panel confirmed while talking to the alumni) while the learning outcome 6.1 is the ability "*to write software by using selected programming system (imperative, object-oriented, functional, logic) taking into consideration the needs of the applied field*". This clearly fails also because **Logic Programming** is only an optional subject. It must be added however, that most of the outcomes defined by the eight "*Subject competences*" are covered by the syllabi of the courses, and can be probably achieved, at least to some extent.

3. Staff

There are 36 teachers teaching the students of the Informatics curriculum at the Department of Computer Science, Faculty of Mathematics and Informatics Vilnius University. Out of them there are 30 who hold a PhD degree. All subjects (modules) but 5 (in the case of the Computer Science track) or 3 (in the case of Mathematical Computer Science track) are taught by lecturers without PhD. This means that the legal requirements are satisfied with this respect. In particular the number of the teaching staff is more than adequate to ensure learning outcomes.

Among the 36 teachers about one third are active and internationally visible researchers, about half of the staff are teachers without internationally recognized research record, and about one sixth are teachers with some research record. This is not bad. But on the other hand it may be seen as quite worrying that, among the teachers whom the Reviewing Panel consider to be active researchers, there are two whose research is in probability theory/statistics, two doing computer modeling, and one whose area seems to be physics. While the contents of the curriculum is (correctly) very much oriented towards subjects in algorithms and theory of computer science, there is nobody among the teaching staff who has a publication record in this area. As it was already explained above, in the section "*Curriculum design*" this is unfortunately reflected in the quality of the syllabi concerning the area of algorithms and theory of computer science.

It is worth mentioning that the Faculty seems to defy, at least to some extent, the typical in Eastern Europe culture of academic inbreeding: two of the teachers of the Informatics curriculum have their PhD degrees from French institutions.

As far as the Reviewing Panel was able to understand the rules, the younger teaching staff members are employed for a 5 years period. Then they can apply for a position again, in an open competition. A teacher who wins a competition for a professors position for the third time is tenured. But still, even the tenured teachers are evaluated each 5 years. This system is not bad, except that even teachers who never get promoted to the professor position should have a chance for stabilization after some point. Concerning the evaluation, the basis for it is the number of publications listed by the ISI Web of Science index. This last regulation is independent on the Faculty of Mathematics and Informatics, and is very unfortunate for many reasons. First of all the number of publications hardly can be seen as a proxy of the quality of research. Secondly, it should be understood, both by the University and by the people who are in charge of the higher education in Lithuania, that (i) the ISI Web of Science index is losing – due to impact factors inflating – its usefulness as a tool to measure scientific achievements, and (ii) it is not, and never was, a correct tool to measure scientific achievements in the broad area of computer science, as many of the most prestigious venues of publication in this area are not indexed by Web of Science. At the moment when this Report is being written the best proxy for scientific value of a publication venue in computer science is the service Microsoft Academic Search.

One more issue, which seems to be to some extent typical for Lithuanian higher education institutions, is the aging of the staff. The average age of the teaching staff is about 49 years. No active policy to support younger teachers in their research careers was spotted by the Reviewing Panel. Just the opposite – Review Panel heard that financial incentives are created which encourage younger teachers

to teach more rather than do research. As far as we understood the view of the Head of the Department, he does not see this issue as a worrying one. The Reviewing Panel do.

4. Facilities and learning resources

The buildings of the faculty are adequate for that programme. Renovations have been done in one building, and a new building is planned for the near future.

The faculty is well equipped with computing resources. Recently (2 years ago) a supercomputer with 2000 cores and 600 TB of disk space was bought. This is currently the largest supercomputer in Lithuania. The Reviewing Panel was informed that up to 40% of computing power is sold to companies, which implies that 60% is left for university projects. The faculty invests 200.000 LTL for replacement of equipment every year. The buildings are also well equipped with wireless communication. Furthermore, the students have access to computing services from the faculty. 250MB space seems to be appropriate for each student. Also on the positive side the renovation of several computer rooms (8 new computer classes and 3 to be renovated soon) can be mentioned. The facilities for disabled people should be improved if possible.

The department has two locations with two buildings next to each other in Naugarduko Str. 24 and Šaltinių Str. 1a, and another location in Didlaukio Str. 47. The two locations are reachable by public transport, which however takes some time. The Reviewing Panel estimate this to be roughly 30-40 minutes. However, it can be stated positively that the timetable is organized such that student usually do not need to travel between the two locations on the same day. The same is true for teachers, if they have a course in one location, they will not have lectures in the other locations. The students reassured the Review Panel that there is no problem with the timetable.

The faculty invests between 17.000 and 28.000 LTL per year for the library. The library is well equipped with current computer science literature, albeit there also seem to be some concentration on lecture handbooks. For example, up to 270 copies of some books written by lecturers from the faculty are found in the library. It can also be positively remarked that the library has access to ACM/IEEE digital library, which gives the students the possibility to read many current computer science papers and journals on-line. Students report that most of the material they need are accessible on-line.

5. Study process and student assessment

The admission process is typical for the Lithuanian higher education institutions – the competition score is made up of the marks for mathematics, IT and Lithuanian language during school final examination and annual mark for foreign language multiplied by leverage coefficients. General requirements of entering higher educational institutions of Lithuania are published on the website of general admission of Lithuania's Higher Educational Institutions Association. (Detailed information of entering the Faculty study programmes is presented on the website of FMI The requirements for state-funded and not state-funded study places are the same).

The Reviewing Panel did not notice any problems concerning the organization of the study process, except from the fact that not all the courses listed in the study programme as electable are taught each year, which is unacceptable, because the list of electable is part of the study program and as such it constitutes a contract between student and the Department.

According to the SER, an average of 6 students a year participate in Erasmus students mobility program, which means that more than one student in ten takes part in Erasmus during his 4 years study period. This is not bad, also because the students seem to avoid the temptation of going to sunny places for a semester-long vacations (which is one of typical problems concerning Erasmus) and chose very decent universities, mainly in Scandinavia and the Netherlands. Most students say however that – being aware of the possibilities offered by Erasmus – they are not planning to go anywhere, because they already have jobs in Vilnius.

The assessment rules are very precise and are part of the syllabi of the courses. It is remarkable that some of the social partners say that, when inviting students for a job interview, they take into account the student's marks. This is very important, because it means that the marks indeed carry some information. It is also remarkable how much the graduates of the program (or even the students) are wanted by the employers (this is what the Reviewing Panel concluded after meeting the social partners).

University provides the following social support options for the students: incentive scholarship for particularly good study results, social scholarship for students from the needy families or living alone, persons receiving social allowance; students with 45% and lower level of disability; or those below 25 years of age who were granted care or both parents (or one of the parents) are dead. One-time social scholarships are given to students in cases of death of a family member, natural or other disaster,

disease or similar case, and also one-time target scholarship is granted to the students who have achieved good results in sports, cultural and research / public activity.

Disabled students can receive social allowance, and they can study according to individual plans. All the students have a right to get accommodation in Students' Residence. Student Representation Office is involved in numerous activities and invites students to take part in cultural and sports programmes.

6. Programme management

The Informatics BA study programme is administered by the Study Programme Committee which includes teachers of the Faculty, as well as social partners and students' representative. The composition of the Committee has remained quite stable during the assessment period, with representatives of students changing more frequently. Since 2009 professor Rimantas Vaicekauskas is the Chairman of the Committee. Other members of the Committee currently are: prof. dr. Gediminas Stepanauskas, assoc. prof. dr. Rimvydas Krasauskas, assoc. prof. dr. Stanislovas Leonas Norgėla, assoc. prof. dr. Vilius Stakėnas, social partner Bronislovas Dzindzelėta (CC "Omnitel"), social partner Gediminas Mikaliūnas (CC "Alna"), 3rd year student of Informatics Gintarė Budrauskytė.

The Reviewing Panel have the impression that the Study Programme Committee has available all the tools needed to assure high quality of teaching. It follows however from the discussions we had that the Committee does not meet on a regular basis and is not too active. We were, for example, told by one of the Committee members that "teachers who want to change something in the syllabus of their course only need to notify the Committee, which approves everything without even meeting, just by email". This opinion is coherent with the observations we made about the syllabi (see section '*Curriculum design*').

Student feedback is collected, using an online system. The teachers say that in average between 15% and 20% of students of each course give them any feedback (usually only in the numerical form, with no text included). Since each student has in average about 5 or 6 courses each semester this is coherent with what the Reviewing Panel heard from the students – that a student who wants to see his marks in the system has to give feedback for at least one course he took in the last semester and that hardly anybody does anything more than that. The feeling among students is that the feedback system cannot be trusted as an efficient tool of improvement.

Concerning the involvement of the external stakeholders in the curriculum development, the opinions the Reviewing Panel heard from the social partners were mixed. We had a feeling that some of the social partners would like to see more software engineering courses included in the curriculum and that

they are disappointed that their opinion is not taken into account by the Department. But, on the other hand, there were also social partners who expressed a belief that the choice the Department made – namely to have a more mathematically oriented computer science curriculum rather than a software engineering one, serves the needs of the students well and is correct. The Reviewing Panel, while remaining critical about many particular design choices, in general share the point of view and believes that the role of the social partners in this curriculum is just correct – they are listened to, but it is the University that makes the decisions.

In general, the feeling of the Reviewing Panel is that the quality of management is a weak point of this study programme. Neither the leaders of the Department seem to be worried about the issues that the Reviewing Panel sees as worrying (for example the research activity of a younger teachers) nor they seem to believe that improvement is within reach. Many times we heard the Department leaders saying that “this cannot be done within the means we have”.

III. RECOMMENDATIONS

1. The management of the programme needs to be more active. The people who run the programme need to believe that improvement is possible and success is achievable.
2. The general principles the curriculum is based on are correct. But the implementation is wrong, at least concerning the theoretical subjects, and major changes are needed in this part. Since none of the teachers of the Department is an active researcher in theoretical foundations of computer science, the Department should possibly look for external advice with this respect. The programme must also be more flexible, with much more optionality built in.
3. The way the theoretical courses are taught needs to be changed so that the students actively prepare for classes, being on regular basis confronted with the situation when they need to overcome some sort of difficulty to solve a task.
4. Ways must be found to increase the research activity of the staff, in particular the younger staff. New, younger, people, possibly from other institutions, need to be attracted. No incentives should be created that lead to overloading of the staff by teaching. This in particular concerns the younger staff members.
5. When assessing the staff, the ISI Web of Science list should not be taken into account, since it is not, and never was, a correct proxy of a quality of a publication venue in computer science. If there is a need of a "parametric" assessment then the ranking lists provided by Microsoft Academic Search are a much more useful tool at the moment.

IV. SUMMARY

The philosophy of the first cycle (Bachelor) study programme in *Informatics*, taught in the Department of Computer Science, Faculty of Mathematics and Informatics, Vilnius University, is consistent with the tradition of computer science studies offered by mathematical departments, with strong emphasis on fundamental theoretical subject and on understanding of the basic concepts, and – in consequence – with technology subjects receiving relatively less time than it would be normal in the tradition of technical studies. This model has proved worldwide to produce graduates which are attractive for the labour market, able to follow the fast evolution of technologies and capable not only to produce code but also to think and solve problems.

This programme is not an exception, at least in the sense that its graduates are indeed in high demand on the job market. This does not mean however, that there would be no room for a lot of improvement. The Reviewing Panel finds the programme management to be its weak aspect. This weakness results, in particular, in many faults in the curriculum design, including many repetitions and some courses covering topics which are not really of central interest in computer science. The Reviewing Panel is also critical about the teaching methods, which do not rely much enough on students own activity. This concerns in particular the mathematical courses (including the ones in theoretical computer science).

Concerning the teaching staff, the feelings of the Reviewing Panel are mixed. On one hand there are many enough active scientists among the staff, particularly for the needs of a first cycle programme. But a closer look reveals that not only just few of the researchers have computer science as their area but also that the staff is aging, that there are very few younger teachers, and that the research quality of the younger staff members does not match the one of the older generation. This means that the outlook, concerning the staff quality, is very worrying.

V. GENERAL ASSESSMENT

The study programme *Informatics* (state code – 612I10002) at Vilnius University is given **positive** evaluation.

Study programme assessment in points by evaluation areas.

No.	Evaluation Area	Evaluation Area in Points*
1.	Programme aims and learning outcomes	3
2.	Curriculum design	2
3.	Staff	3
4.	Material resources	4
5.	Study process and assessment (student admission, study process student support, achievement assessment)	3
6.	Programme management (programme administration, internal quality assurance)	2
	Total:	17

*1 (unsatisfactory) - there are essential shortcomings that must be eliminated;

2 (satisfactory) - meets the established minimum requirements, needs improvement;

3 (good) - the field develops systematically, has distinctive features;

4 (very good) - the field is exceptionally good.

Grupės vadovas: Team leader:	Prof. Jukka Paakki
Grupės nariai: Team members:	Prof. Rolf Backofen
	Prof. Jerzy Marcinkowski
	Vida Juozapavičienė
	Lukas Jokūbas Jakubauskas

**VILNIAUS UNIVERSITETO PIRMOSIOS PAKOPOS STUDIJŲ PROGRAMOS
INFORMATIKA (VALSTYBINIS KODAS – 612I10002) 2014-03-21 EKSPERTINIO
VERTINIMO IŠVADŲ NR. SV4-97 IŠRAŠAS**

<...>

V. APIBENDRINAMASIS ĮVERTINIMAS

Vilniaus universiteto studijų programa *Informatika* (valstybinis kodas – 612I10002) vertinama **teigiamai**.

Eil. Nr.	Vertinimo sritis	Srities įvertinimas, balais*
1.	Programos tikslai ir numatomi studijų rezultatai	3
2.	Programos sandara	2
3.	Personalas	3
4.	Materialieji išteklių	4
5.	Studijų eiga ir jos vertinimas	3
6.	Programos vadyba	2
	Iš viso:	17

* 1 - Nepatenkinamai (yra esminių trūkumų, kuriuos būtina pašalinti)

2 - Patenkinamai (tenkina minimalius reikalavimus, reikia tobulinti)

3 - Gerai (sistemiškai plėtojama sritis, turi savitų bruožų)

4 - Labai gerai (sritis yra išskirtinė)

<...>

IV. SANTRAUKA

Vilniaus universiteto Matematikos ir informatikos fakulteto Kompiuterijos katedros įgyvendinamos pirmosios pakopos *Informatikos* (bakalauro) studijų programos filosofija atitinka kompiuterijos mokslų studijų, kurias siūlo matematikos katedros, tradiciją, kai daug dėmesio skiriama fundamentaliai teoriniam dalykui ir pagrindinių sąvokų suvokimui, o tai lemia, kad technologiniams dalykams skiriama mažiau laiko, nei paprastai būtų priimtina pagal techninių studijų tradiciją. Pagal šį modelį visame pasaulyje buvo išugdyti darbo rinkai patrauklūs absolventai, galintys sekti greitą technologijų evoliuciją ir galintys ne tik generuoti kodus, bet taip pat mąstyti ir spręsti problemas.

Ši programa ne išimtis, bent jau tuo aspektu, kad jos absolventai iš tiesų yra itin paklausūs darbo rinkoje. Tačiau tai nereiškia, jog nėra, kur tobulėti. Ekspertų grupės nuomone, programos vadyba yra jos silpnoji grandis. Būtent tai ir lemia daugelį programos sandaros trūkumų, tarp kurių yra daug pasikartojimų, nagrinėjamos tokios kai kurių dalykų temos, kurios iš tikrųjų kompiuterijos moksle nėra pagrindinės. Ekspertų grupė taip pat kritiškai vertina mokymo metodus, kurie nepakankamai pagrįsti

pačių studentų darbu. Tai ypač susiję su matematiniais dalykais (įskaitant teorinius kompiuterijos mokslo dalykus).

Ekspertų grupės nuomonė dėl dėstytojų buvo labai skirtinga. Viena vertus, tarp dėstytojų esama daug gana aktyvių mokslininkų, būtent tokių, kokių reikia pirmosios pakopos programai. Tačiau, kita vertus, tik kelių iš šių mokslininkų pagrindinė kompetencija yra kompiuterijos mokslas; kita problema – personalo senėjimas. Programoje dirba labai mažai jaunesnių dėstytojų, kurių mokslinių tyrimų kokybė nedera su vyresniosios kartos dėstytojų. Tai reiškia, kad personalo kokybės perspektyva kelia didelį nerimą.

III. REKOMENDACIJOS

1. Programos vadovybė turėtų būti aktyvesnė. Už programą atsakingi žmonės turėtų tikėti, kad galima ją pagerinti ir pasiekti sėkmės.

2. Pagrindiniai studijų turinio sudarymo principai yra teisingi. Tačiau pats įgyvendinimas yra neteisingas, bent jau kalbant apie teorinius dalykus, todėl ten ir reikia padaryti pagrindinius pokyčius. Kadangi nėra vienas katedros dėstytojas nėra aktyvus kompiuterijos mokslo srities teoretikas, katedra greičiausiai šiuo klausimu turėtų ieškoti pagalbos iš išorės. Taip pat programa turėtų būti lankstesnė, joje reikėtų suteikti daugiau pasirinkimo laisvės.

3. Reikėtų keisti teorinių dalykų dėstymo būdą taip, kad studentai aktyviai ruoštųsi paskaitoms, reguliariai susidurtų su tokiomis situacijomis, kai, norint atlikti užduotį, reikia įveikti kokio nors pobūdžio sunkumus.

4. Reikėtų rasti būdų, kaip pagerinti personalo mokslinių tyrimų veiklą, ypač jaunesnio personalo. Derėtų pritraukti naujų, jaunesnių asmenų, galbūt iš kitų institucijų. Nereikėtų kurti jokių paskatinimų, kurie sudarytų sąlygas perkrauti personalą dėstomąja veikla. Tai ypač susiję su jaunesniais personalo nariais.

5. Vertinant personalą, nereikėtų atsižvelgti tik į „ISI Web of Science“ platformoje esamą sąrašą, nes tai nėra (ir niekada nebuvo) kompiuterijos mokslui tinkama geros kokybės publikacijų skelbimo vieta. Atsiradus poreikiui atlikti „kriterinį“ vertinimą, šiuo metu daug naudingesnis įrankis būtų „Microsoft Academic Search“ platformoje pateikti reitingavimo sąrašai.

<...>

Paslaugos teikėja patvirtina, jog yra susipažinusi su Lietuvos Respublikos baudžiamojo kodekso¹ 235 straipsnio, numatančio atsakomybę už melagingą ar žinomai neteisingai atliktą vertimą, reikalavimais.

Vertėjos rekvizitai (vardas, pavardė, parašas)

¹ Žin., 2002, Nr.37-1341.