# Support to strengthening the higher education system in Azerbaijan

Twinning project ENI/2018/395-401

**Mission Report**

**Short-Term Mission on Activity 2.4 Improve study programmes in the priority areas in pilot universities to incorporate learning outcomes and inform and raise awareness about these achievements**

**(June 3 – 7, 2019)**

## 1. Name and Function of the Expert:

Full name of experts

| | | |
|---|---|---|
| **Prof.** | **Pierre** | **Collet** |

Signature

## 2. Objective and Tasks of the Mission:

The mission is carried out within the framework of:

COMPONENT 2: **PILOT STUDY PROGRAMME IN PRIORITY AREAS ARE IMPROVED TO BE MORE STUDENT-CENTRED**

Activity 2.4 Improve study programmes in the priority areas in pilot universities to incorporate learning outcomes and inform and raise awareness about these achievements

Benchmarks for this activity are:

- Minimum 12 study programmes revised and updated to incorporate learning outcomes;
- Adjusted methodological compendium;
- Dissemination events to raise awareness are organised

## 3. Time schedule of mission:

| Date and Time | Activity |
|---|---|
| Monday 3rd of June 2019 | - Briefing with RTA and Component Leader II.<br>- Meeting with the members of the Working Group for IT specialties and employers in MoE. |
| Tuesday 4th of June 2019 | Deskwork in the Ministry of Education |
| Wednesday 5th of June 2019 | Deskwork. Revising the state study programmes in Computer Engineering and Computer Sciences |
| Thursday 6th of June 2019 | Deskwork. Revising the state study programmes in Computer Engineering and Computer Sciences |
| Friday 7th of June | Debriefing with the Higher Education Department about results of the mission |

## 4. Relevant Background Information/State of Affairs regarding the mission

A meeting was organised by Lisa Bydanova and her team on Monday June 3rd 2019 with representatives of local companies, ANAS and Azerbaijan Universities:
**(the list of invited people and institutions can be provided by Lisa Bydanova)**

A debate started on the question "Are you satisfied with the students who graduate from universities, and if not, what do you suggest to improve the level of the students to match the skills you need?"

What comes out of this is that it seems that communication between universities and companies could be improved. The impression of the expert is that universities teach the students without enough interaction with companies, so the result is that the knowledge, experience and skills of the students who come out of Azerbaijan's higher education system are not consistent with what companies need.

## 5. Achievement of the Expected Results

A new version of the State Standards with the learning outcomes and suggested revised syllabus has been produced, both for Bachelor degrees in Computer Science and Applied Computer Science, but also for Masters degrees in Computer Science and Applied Computer Science.

The mission contained mostly a deskwork; 1 meeting with the members of the Working Group for IT specialties and employers in MoE was organised on Monday and a debriefing with the MoE staff on the mission's results took place on Friday. Because of Ramadan holidays it was not possible to schedule more meetings with relevant staff from universities.

Considering the results of all meetings, the suggestion for modifications in current SSHE in Computer Science include:

1) more courses on applied computer science
2) more practical work through projects, internship, etc.;
3) introduction of internships and apprenticeship.

The last point is linked to the nowadays demand for experienced software programmers by the companies.

## 6. Unexpected Results

N/A

## 7. Issues Left Open After the Mission

It could be considered to implement the new recommended curriculum progressively, taking the time from 3 up to 5 years (cf. below).

## 8. Recommendations (including recommendation for future missions)

Extensive study of the national curricula both for Computer Science and Computer Engineering show that:

- **For Bachelor degrees :**
    - Out of 36 teaching modules, 14 are on General Knowledge (Culture General Competences), and 22 focus on Professional competences), but all in all, only 6 are courses on practical Computer Science (PC 2, 12, 14, 15, 18, 21).
    - Only 6 modules out of 36 on Computer Science (*i.e.* 17%) is not enough computer science in a Computer Science curriculum.
    - Then, most courses on Computer Sciences are not applied courses, but theoretical courses.

- **For Master degrees:**
    - None of the 44 modules contain Computer Sciences.

This confirms the summary of the main complaints by the companies that were underlined during the meeting with employers:

- Current internships are not adapted to what companies need: the students are not skilled enough when they come for an internship, so no real work can be done with them.
- There is a real difference between expectations from the companies and the real knowledge and skills of the students.
- For many students, adequate knowledge come from outside the university curriculum, through self-learning.
- Education is too general, the level on specific contents is too poor.
- Software development is underdeveloped compared with Information Technology (IT).

The above mentioned suggest that the state standard for Computer Science study programmes needs to be updated.

### 8.1 Remarks on the curriculum content, both in Bachelor and Master degrees

Before proposing a curriculum on Computer Science, it is very important to understand well that Computer Science is not a science such as Mathematics or Biology.

#### *8.1.1 Programming* ***is a skill***

There is a great difference between knowledge (what you acquire by learning) and know-how (what you acquire by practice).

Suppose you want to teach students how to ride a bicycle. If you teach 100 hours of courses on explaining how a bicycle is made, (the materials to be used for wheels, the mechanical constraints on spokes, chain links, pedals, brakes, etc…); 100 hours on the physics of cycling, (inertia, centrifugal force showing at what angle you must lean in the turns, the gyroscopic forces of the wheels, etc), and 100 hours of courses by a professional cyclist *to show* students how to sit on a bicycle, how to pedal, how to ride in a straight line, how to turn, how to brake,... After these 300 hours of courses on “learning how to…”, the students will have learned everything on bicycle making, bicycle riding physics, bicycle riding techniques (with demonstration), but it is not sure that many students would in the end actually manage to ride a bicycle.

### *8.1.2 Computer Science **is an art***

Computer Science is not only a skill, but it is also an art. One of the major world-class professor in Computer Science is Prof. Donald Knuth. His fundamental books on Computer Science are called: “The Art of Computer Programming”. The first volume came out in 1968 and the latest one is supposed to be published in November 2019. Prof. Donald Knuth was awarded the Turing Award (= Nobel prize in Computer Sciences) for this work in 1974. Donald Knuth’s most famous quote is: “Beware of bugs in the above code; I have only proved it correct, not tried it” (cf. Donald Knuth’s wikiquote page).

There are many ways to write the same program: some programs are more “elegant” than others, some are more efficient, more robust, more secure, but choices must always be made when programming, because programming is a creative process and therefore, it qualifies as an art.

Understanding and establishing officially that Computer Science is both a skill an art is very important for creating a good curriculum in Computer science, because even if you could teach a pure science such as Mathematics only thanks to theoretical courses, it is **impossible to teach Computer Science without practical courses**.

### *8.1.3. How to teach a skill and an art*

Teaching a skill and an art requires many practical lessons in the study programme. Because Computer Science is a difficult art, **a lot of time to teach it must be devoted to practical courses**.

### *8.1.4 Organisation of courses*

In France, a study programme for a total of 200 students doing a bachelor’s degree at university would comprise 4 different type of classes:

- ***Lecture course (LC):*** given in a large auditorium in front of the whole 200 students, to teach theory, algorithmics, notions, concepts. This is “top-down teaching”. The students must take notes and learn the contents explained in the Lecture Course (very similar to what is currently done in Azerbaijan curricula).

- ***Integrated course (IC):*** given in a large auditorium, in front of the whole 200 students, these are lecture courses that include examples. After explaining a concept, the teacher stops and shows an application of the theory on the blackboard by doing an exercise, or implementing an algorithm on a computer, shown to all by using a video projector. These “Integrated Courses” mix theory and exercises.

- ***Supervised work (SW):*** given in an “exercise room”, to a maximum of 40 students. If the whole student cohort is 200, this means creating 5 groups of 40 students. The groups can be taught by different teachers but must have the same contents (needs coordination between the different teachers). The teacher gives exercises to solve to the students, such as writing an algorithm on an exercise book. The students have 5 to 10 mn to solve the exercise, during which the teacher can answer questions, give explanations. At the end of the 5 to 10 mn, a student is selected to go to the blackboard to show what he has done. The teacher helps him in front of everyone to correct his mistakes. All this is done on paper, not on a computer.

- ***Practical work (PW):*** given to a maximum of 20 students, in a computer room containing 20 computers, meaning that for 200 students, 10 groups of students must be created. Each student is in front of a computer. He/ she has to write a small program to implement an algorithm seen during a supervised course, because as for cycling, you need to “practice on a real bicycle to learn how to ride a bicycle”.

For each topic, a senior teacher is identified as responsible for the topic. He/she gives the Lecture and Integrated courses, and, if several groups of Supervised / Practical work are needed, he/she coordinates the different teachers who will teach the supervised / practical sessions.

Preparation time is greater for Lecture courses than for Integrated courses than for Supervised work than for Practical work. Therefore, the current practice in France is to pay standard hours for Supervised Work. Integrated courses count (on the payroll) as 1.25 Supervised work hours because they require more time for preparation. Lecture courses count (on the payroll) as 1.5 Supervised Work hours for the same reason. Because during Practical Work, students are supposed to work on their own (teachers are only there to answer their questions), Practical work is only paid 0.75 Supervised Work hours.

Therefore, a non-tenured (cf. below) teacher doing 10h LC + 10h IC + 10h SW + 10h PW will be paid 10*1.5+10*1.25+10*1+10*0.75 = 45 hours.

Typically, senior teachers get more LC and IC than junior teachers, who will be doing SW and PW, meaning that senior teachers are paid “more” for the same number of face to face hours than beginning teachers. In all universities, though, permanent tenured staff get paid the same for Practical work than for Supervised work, so the special 0.75 PW rate being applied for non-tenured staff. Then, x1 rather than x0.75 for tenured staff only applies if the teacher is not in overtime (192h/year, as French Associate/Professors teach 50% of their time = 192h/year).

Finally, courses in France are done in 2 hours sessions, as it is considered that 1h is too short to start doing something interesting (the time for the students to get in the room, start the computers if in Practical Work session, excitement towards the end of the course, a 1h00 session results in 45mn “real” work, which is very short to explain things. In practice, the French 2h is a bit too long[1]. It could be recommended to put in place a duration of 1h30 for teaching sessions. It is also a convenient duration as it is compatible with 3 ECTS topics, which typically corresponds to ~ 30h hours (therefore 20x 1h30 sessions); however the number of face to face hours may vary.

#### *8.1.5 Distribution of LC, IC, SW, PW*

Some topics are more applied than others, meaning that the distribution between LC and IC (theoretical courses) and SW and PW (to improve student’s skills) can be different. However, some rules are observed: typically, there is no less than 33% SW and PW compared to LC and IC, even in the most theoretical topics.

For example, in the University of Strasbourg (France), the “Graph theory” (purely theoretical topic) has 20 LC (lecture course in front of all students) + 14 SW (exercises session for all students to practice graph theory on concrete exercises); but this is an exception (we try to apply most computer science).

On most Computer Science topics, however, there is typically only 50% to 33% LC or IC, compared with SW and PW, so this means 50 — 66% practice in groups of 40 students (exercises on paper) or 20 student groups (practice on computers).

As an example, in Strasbourg:

- Data Structures and Algorithms (important 6 ECTS topic) have 20 LC, 22 SW, 12 PW
- Databases have 12 LC, 14 SW, 10 PW
- Computer architecture has 8 LC, 10 SW, 12 PW
- Web programming has 12 LC, 15 PW (no SW on paper)
- Logic and Logic programming has 18 LC, 22 SW, 6 PW
- Software development techniques (less theoretical) has 14 IC, 16 PW

According to Muriel Penicaud (current French Minister of Labour) and Stanislas Dehaene, (Professor at Collège de France, member of the French Academy of Sciences, President of the National Scientific Committee for Education) : “Cognitive sciences show that we learn more and more deeply, by regularly putting to practice what we learn. There is an intrinsic relationship between theory and practice, between the brain and the hand”.

As a conclusion, it could be recommended that the study programme contains **at least 50% exercises and practical work (SW + PW)** compared to theoretical LC + IC.

[1] At the French – Azerbaijani University (UFAZ), where I’m teaching it was adopted 1h30 sessions, which seem to be a really good duration.

### *8.1.6 Evaluation*

The 2012 French board of University Presidents (CPU) stated that all courses should be evaluated on at least 3 moments (In computer science, this is quite simple to implement):

1. a mid-term exam (possibly online, or during a practical work session, or a standard exam)
2. a homework mini-project (that can be done by groups of 2 or 3 students)
3. a final exam.

The homework mini-project is essential as it is a hands on practical project, that requires the students to actually program on their computers. Assigning them to be done to groups of 2 or 3 students develop the organisation skills of the students, their self-learning. They force students to practice.

The mini-project should be handed back with a small 4-5 pages report that also develops writing skills.

In specific topics such as English, practical evaluation can be done through an oral presentation of the group of students in front of the class, with power point-like slides to develop presentation skills, which will be very useful in their later professional life.

It could be recommended that **at least 3 grades per course,** with no grade counting for more than 50% of the total grade for the module, and with one grade evaluating a mini-project (2-3 students) with a report or a presentation in front of all to develop collaborative and presentation skills.

## 8.2 Remarks on the specificity of master's degrees requirements

### *8.2.1 Master's degree to be necessarily backed on a research laboratory/institute*

In France, all University courses (even at bachelor's degree) must be based on research.

In practice, bachelor's degree levels are not much linked to research, because the students do not yet have the skills necessary for that. However, Master's degrees **must be done in correlation with research** done in a university research laboratory.

For each Master's degree, there is a group of common courses, common to all Master's degrees of the University, but if there are 3 research laboratories in Computer Science in a University, each research laboratory must have the opportunity to teach their research specialty in a specific Master's specialty.

For instance, in the University of Strasbourg, we have one Master's degree, that is given in *5 specialties* (*Data Sciences and Complex Systems, Imaging and 3D*, *Science and Engineering of Networks*, *Internet and Systems*, *Science of Software Engineering* and *Management of Computer Science Projects*).

The specialty courses of the *Data Sciences and Complex Systems* Master's specialty are done by researchers from 2 research teams: Complex Systems and Theoretical Bioinformatics team and Data and Knowledge Sciences team.

The specialty courses of the *Imaging and 3D* Master's specialty are done by researchers from the Computer Graphics and Geometry team and the Images, Learning, Geometry and Statistics team. And so on…

This link between researchers and master's student is essential, as it gives the opportunity to researchers to teach the latest available knowledge to master's students, and to train future PhD candidates for the research teams.

This also means that at master's degree level, teachers are not 100% teachers; they are **50% teachers and 50% researchers that all have a PhD**.

It could be recommended that all universities that offer master's degrees in Computer Science should have **<u>at least a research laboratory in Computer Science</u>** to make this link between master's students and researchers.

### *<u>8.2.2 Internships</u>*

Because of the importance of practical work, the last term of a master's degree is devoted to **a 20 weeks internship** in a company. 20 weeks is a long period, so typically, courses stop by the end of February, and internships start in the beginning of March, until the end of August, and the Master's jurys take place on the last days of August.

**All internships topics must be validated by the head of the master's specialty**. The topic of the internship must be directly linked to the topic of the master's specialty. For example, an internship in a company to create a database is not valid for a master's degree specialty in imaging.

The internship must be described by a half page description of what the student must do, and how he/she will be supervised (i.e. weekly meetings with the company supervisor, etc.).

The internship must contain a real work from the student. In order to make sure that everything goes right, **2 supervisors** follow the student's work during the 20 weeks:

- an **academic supervisor (**typically a teacher from the master's specialty) and
- a **company supervisor**, who works with the student in the company.

The University supervisor visits the student in the company at least twice during the 20 weeks of the internship. He verifies that everything is progressing well and that the internship will not be a failure (important because this internship counts for a semester).

These internships are very profitable to the student (an opportunity to have a hands-on experience in a real company) and to the company (an opportunity to hire someone who works in the company for 20 weeks, so who will know the company well).

At the end of the internship, the student must hand back **a 20 to 30 pages report** describing what he/she has done. For each student, **a defence of 20 to 30 mn** is organised at the end of August. The student shall present orally what he/she has done in front of a jury, consisting in at least 2 teachers of the master's specialty, together with the academic supervisor of the student and the company supervisor of the student.

As a conclusion, it could be stated that **internships** (involving real work, supervised by a university teacher and a company tutor) are very important as they **create a strong link between universities, companies and students.** They **boost immediate employment** after a bachelor's or a master's degree and support **development of practical skills.**

### 8.3 Remarks on teacher's requirements

In France, all permanent academic staff holds at least:

- for Associate Professors: a PhD
- for full Professors: a PhD and an 'HDR - Habilitation à Diriger des Recherches' (which is an Accreditation to Supervise Research). It is a kind of "super" PhD that is a summary of around 10 years of research, with at least several recent (less than 3 or 4 years) publications in international refereed journals and the supervision of at least 1 PhD student who defended successfully his PhD.

Associate Professors and Full Professors are hired on a 50% teaching and 50% research basis. Typically, Lecture Courses and Integrated Courses are preferably given to full Professors, while Supervised Work and Practical Work are given to Associate Professors.

In some higher education institutions that do not teach master's degrees and have only bachelor's degrees, it is possible to hire Professors from the High Schools to give Supervised and Practical Work. These High-School teachers teaching in higher education institutions (PRAG, in French, for PRofesseur AGrégé) hold at least a master's degree, and possibly a PhD. The difference is that because these teachers are not researchers, they do 100% teaching (so one PRAG gives as many hours as 2 Associate Professors or Full Professors).

It could be recommended that **permanent / tenured Associate or Full Professors all hold a PhD in Computer Science.** In Engineering schools where only bachelor's degrees are taught, it is possible to hire teachers with only a master's degree, but to give Supervised Work or Practical Work courses.

It is also important to consider **increasing the pay of Associate Professors and Professors to attract the most skilled staff for academic positions**. There is a strong demand for computer scientists in the industry. Currently, good computer scientists are paid much more if they work for companies than if they work at Universities. This makes it difficult to recruit skilled teachers at University.

### 8.4 Remarks on apprenticeships

In Europe, the unemployment rate of young people is the lowest in countries that implement apprenticeship.

Apprenticeship is a different way to implement internships. Typically, all Universities have at least a Master specialty that is given under the modality of apprenticeship. The way it is done is the following:

- Before the year starts, the student must find a company that will hire him/her as an apprentice.
- As for an internship, the subject of the apprenticeship must be validated by the head of the master's degree.
- In 1st year, the students attend University for 2 days per week and work in the company for 3 days per week, all year long.
- In 2nd year, the students attend University for 1 week per month and work in the company for 2 weeks per month, all year long.

At the end of the 2 years apprenticeship, the company knows the student very well, and the student knows very well the company. Therefore, if the company chooses to hire the student, he/she will be efficient immediately, because of experience in the company for 2 years.

This creates a valuable link between students and companies, and between Universities and companies, which was a complaint of the companies that were attending the meeting at the beginning of my expertise.

As an incentive to companies to hire an apprentice, the government does not apply any employer taxes to the salary of the student in apprenticeship.

This is an investment from the government to maximise employment of young adults immediately after their studies.

Apprenticeships replace internships in Bachelor and master's degrees evaluation. Each student has a University and a Company tutor, as for internships.

Each year, the student must hand back a 30 to 50 pages report and make a 20 mn presentation in front of a jury (made of at least 2 teachers of the curriculum plus the University tutor and the Company tutor).

For more details on French apprenticeship-based Master's degrees, here is the web page of the French Ministry of Education (in French) describing the advantages and modalities of this process:

https://www.alternance.emploi.gouv.fr/portail_alternance/jcms/recleader_6113/decouvrir-l-alternance

In conclusion, it could be stated that **introducing apprenticeship-based Computer Science or Computer Engineering Master's degree** will dramatically **increase employability** of young engineers.

## 8.5 Recommendations on the content of a bachelor's degree in Computer Science

During the meeting with the companies, some of them complained that the students did not know about modern computing languages, and they gave as an example that they were taught the PASCAL language, that they did not know about object oriented programming, about artificial intelligence (neural networks)...

PASCAL is a language that came out in 1970 and that no companies are using it anymore. Teaching of PASCAL was abandoned in France in the beginning of the 1990's. As for neural networks (Artificial Intelligence), the PERCEPTRON came out in the late 50s; hence it could be considered teaching Artificial Intelligence neural networks in Azerbaijan too. The fact that PASCAL is still taught, and that neural networks or artificial intelligence are still not taught means that the contents of the curricula must be seriously renovated.

The (quick) history of Computer Science languages is the following:

- **1950: beginning of electronic computers.** (EDSAC, the first real computer: 1948)
- **1950 - 1970: development of the first programming languages.** Fortran (1950), Algol (1958), Cobol (1959), LISP (1968), Pascal, C, Smalltalk. (1st Object Oriented Language) (1970), Prolog (1972), …
- **1990 - now: Apparition of "Modern" languages.** C++ (1985), Java, Python (1991), PHP (1994), Javascript (1995)...

From what I heard; I have the feeling that the computer science taught in Azerbaijan is pre-1990 computer science. I might signify that since the independence of Azerbaijan in 1991, there was no major change in curricula in this field.

Because all Universities are different and have different specialties, I will give here the basis for a solid ***common*** curriculum in Computer Science, that will however leave the possibility for different Universities to teach their specialty topic. Typically, the ***<u>basic</u>*** knowledge that needs to be taught ***<u>in practice</u>*** to Computer Science students is:

- Algorithmics (with exercises and practical work)
- Data Structures (with exercises and practical work)
- ***<u>Programming in</u>*** (not "knowing about") C, but also post-1990 object-oriented languages (Python, C++, Java)
- Web programming front-end and back-end (Javascript, PHP-MySQL)
- Operating systems operation and architecture (Linux)
- Distributed programming (synchronization, sockets, shared memory, RPC)
- Designing, using and programming databases
- Knowing about computer architecture and how to interface programs with devices
- Using networks and developing protocols
- Developing artificial intelligence algorithms
- Parallel programming and with MPI, OpenMP, OpenACC (since 2005, ***<u>all</u>*** computers are parallel, so it is nowadays a necessary skill)

**Recommendation: These courses + University-specific courses on applied computer science should represent 70% of a Bachelor Degree (cf. Annex: New State Standard for Computer Sciences BA & MA for more details). Important courses can be given in parts, such as Algorithmics 1, Algorithmics 2, … so that**

**courses are given over 3 (or 4 years, because in Azerbaijan, secondary studies stop at age 17, not at 18 as in France and other European countries).**

### 8.7 Recommendations on the contents of a master's degree in Computer Science

Because Master's degrees must be linked to a research laboratory and because different universities will typically host laboratories with different research topics, here again, I think it is better to provide some minimal contents for a Master's degree, to be complemented with topics from the research laboratories of the local University, or from specialties from the local professors.

Typically, among the different possible topics, a master's degree in Computer Science should contain at least courses on:

- 3 ECTS: Advanced algorithmics (with exercises)
- 6 ECTS: Advanced programming (with practical work projects)
- 6 ECTS: Advanced distributed computing (with practical work projects)
- 6 ECTS: Collective and Artificial Intelligence (with practical work projects)
- 6 ECTS: Compilation (with practical work projects)
- 3 ECTS: Security (with practical work projects)
- 3 ECTS: Software Certification (with exercises)
- 3 ECTS: Computability and Complexity (with exercises)
- 3 ECTS: Computer aided software proving (with exercises)
- 3 ECTS: Architecture and Web development (with exercises)
- 6 ECTS: Professional life (with project homeworks)
- 6 ECTS: English for Computer Science (obtain C1 level at the end of the course)
- 6 ECTS: Research and Study work (cf. below)
- **27 ECTS:** 4th semester internship (20 weeks from March to August)

Then for a master's degree with specialties on Imaging, computer graphics topics must be added to this core. The same for a master's degree specializing on Deep Learning, or on Networking, or on anything (to be adapted to the capabilities of the different Universities).

Research and Study work: This is a large 150h research project that must be conducted by groups of 2 or 3 students in the topic of the master's degree. This helps students acquire the skills of working together, translate theoretical knowledge into practical realizations, work autonomously, communicate on an innovating work (that they can later put forward in their CV).

**Recommendation: The recommended courses above count for 87 ECTS out of 120 ECTS in total, leaving 33 ECTS for specialty courses in a specific domain or other courses such as Azerbaijan culture. The details of these courses are given in "**Annex_New state standard for Computer Sciences_BA&MA.docx**" file.**

### 8.8 Remarks on the difference between Computer Science and Computer Engineering

As is described in the name, Computer Science is about developing the science of computing. Students are taught more theoretical and less applied topics, because after their master's degree, they should be able to pursue their studies with a PhD.

Computer Engineering students are typically more oriented towards company work at the end of their bachelor's degree or master's degree. Therefore, they have slightly less theoretical topics and more industry-oriented topics such as courses on how to setup a company, know about marketing, economy, business, law, how to understand and write contracts, know about copyrights, patents, and so on.

Computer Engineering degrees typically use more Project-based learning approaches. Quote from https://en.wikipedia.org/wiki/Project-based_learning: *"Project-based learning (PBL) is a student-centered pedagogy that involves a dynamic classroom approach in which it is believed that students acquire a deeper knowledge through active exploration of real-world challenges and problems."*

So the basic courses on computing are the same as those for Computer Science, but more theoretical topics are replaced with more practical and more business-oriented topics.

The typical ratios for Computer Science degrees are 70% Computer Science, 30% connex sciences and general knowledge. The ratio for Engineering degrees is different: 60% Computer Science, 20% connex sciences and general knowledge, and 20% Engineering specific topics. Internships are obligatory, as well as projects and (in France) international mobility.

In fact, Engineering topics (centered on business management, marketing, law, etc.) are **added** to standard curriculum because Engineering degrees ***have 6 more ECTS per semester*** than standard Science degrees (= 12 more ECTS / year). Therefore Computer Science represents "only" 60% of the courses in Engineering degrees (the additional courses are typically in humanities concerning setting up companies and business).

In France, international mobility can be:

- 1 term (or 1 year) abroad, or
- 1 internship abroad

### 8.9 Conclusion

Analysis of the current state curricula in Bachelor and master's degrees in Computer Science show that they do not contain enough courses in Computer Science. This explains why companies are not happy with the proficiency of students coming out of Universities.

#### *8.9.1 Analysis of the current bachelor's degree in Computer Science*

Here is a quick analysis of the current bachelor's degree contents.

Color code: ***Brown = problematic***, **Green = ok**, Grey = not related to acquiring skills in Computer Science:

- (CGC1): Scientific understanding of milestones and chronology of Azerbaijan's history,...
- (CGC2): to know fluently the language of the Republic of Azerbaijan (Azerbaijani)...
- (CGC3): to understand the content of speech and writing in one foreign language…

- (CGC4): to know social environment and social heritage relations,...
- (CGC5): to have a scientific understanding of social behavioral forms and consistencies…
- (CGC6): to be knowledgeable about the forms and types of culture, to know the regions…
- (CGC7): basics of economic theory, money-credit and social investment policy,...
- (CGC8): to be knowledgeable about the basics of legal system and law enforcement…
- (CGC9): to be capable of drafting regulatory and legal documents…
- (CGC10): to maintain healthy lifestyle
- (CGC11): to be ready for intercultural dialogue
- (CGC12): to master criticism and self-criticism habits
- (CGC13): to propose new ideas and justify them
- (CGC14): to have an ability to take initiatives and to take on responsibility...
- (PC1): ***To develop programs*** resolving economic and other issues,...
- **(PC2): *to know* the purpose and tasks of computer sciences,...**

- (PC3): ***to know*** the elements of the theory of clusters…
- (PC4): ***to have information about*** curve concept,...
- (PC5): ***to have knowledge about*** complex numbers,...

- (PC6): ***To have knowledge about*** properties of Lebeq-Riman,...
- (PC7): ***To have knowledge about*** Cartesian coordinate system…
- (PC8): ***To be able to*** perform on multilayers and matrices,...
- (PC9): ***to know the basics of*** discrete mathematics, its application,...
- (PC10): ***to know the goals and objectives of*** physics,...
- (PC11): ***to know*** the conventional equations of ordinary differential…
- **(PC12): *to know* algorithms, their properties and their methods,...**
- (PC13): ***to know*** guidelines and normative documents…
- **(PC14): *to know* programming languages and programming methods**
- **(PC15): *to learn* modern programming languages (C++, Java, etc.), ,...**
- (PC16): ***to know*** the events and operations on them (proba, stats)
- (PC 17): ***to know*** the necessary and sufficient conditions of existence of extrema...
- **(PC 18): *to learn the theory of* formal languages that are key areas of algorithmic theory**
- (PC 19): ***to study the basic concepts of*** mathematical logic, axiomatic methods,
- (PC 20): ***to know the methods of*** convergence of the function,
- **(PC 21): *to know how to* organize the database, to learn data models,**
- (PC 22): ***to be knowledgeable about*** the elements of graph theory,

1. The first issue in the current curriculum : the real problematic things ***highlighted in brown*** are that ***all*** PC (Professional Competence) modules but the first one (**"to develop programs**") are "to know, to have information about, to have knowledge about, to be able to, to know the basis of...". There is a very real problem there because as I tried to explain and demonstrate in sections 8.1.1, 8.1.2, 8.1.3, Computer Science is a skill and an art, that can only be learnt by practice, by writing programs or mini-projects on a keyboard and a computer. Once more, if a pizza delivery company hires a cyclist after

he took courses on “learning the theory of riding a bicycle” for 3 years at University, the company will be very disappointed, because even though the applicant will have a Bachelor Degree in riding bicycles, he will fall from the bicycle if he never practiced it during his Bachelor degree.

So in this curriculum, out of 36 modules, only one (PC1) is about developing programs. Unfortunately, I coloured it in grey because… it is not about developing Computer Science programs, but developing programs “resolving economics and other issues”, which is out of scope of computer sciences.

2. The second issue: All companies in the meeting complained about students not knowing much about useful computer science when they hire them (they said that good computer scientists they hired had acquired their skills on their own, out of the University).

I put in grey all modules not related to acquiring Computer Science skills, and in green all modules where students can acquire Computer Science skills.
Out of 64 modules, only 6 concern more or less directly Computer Science (and not maths, physics, or how to maintain a “healthy lifestyle”). This means that only 17% of the modules of a Computer Science bachelor’s degree directly concern Computer Science. It is hence difficult to expect bachelor’s degree students to be good in programming when only 17% of studies directly concern Computer Science (and nearly none concern computer programming).

It is good to have “Culture General Competences” (CGC) modules, but they should be useful for Computer Sciences, such as English (CGC3 is great!), Philosophy of Sciences, Marketing, business-related law, etc. However as regards to “(CGC6): to be knowledgeable about the forms and types of culture, to know the regions…”, it seems irrelevant for a bachelor’s degree of Computer Sciences. This knowledge could be not acquired in general education (high school or earlier). There is no need for so many modules: 38% of CGC modules compared to PC modules, and in PC modules, only 5/22 on Computer Science.

CGC 10 to 14 should have clearly been addressed in primary (CGC10), elementary (CGC 11-12) or high school (CGC 13-14), and not using precious University teaching time in the curriculum.

**As a conclusive remark,** it could be mentioned that with only 17% modules directly concerning computer science, this is hardly a bachelor’s degree in computer science.

*8.9.2 Analysis of the current master’s degree in Computer Science*

- (CGC-1): to improve and to develop one's intellectual knowledge,...
- (CGC-2): search and independent study of new research methods,...
- (CGC-3): to demonstrate one's knowledge and ability in the field of science and...
- (CGC-4): active social mobility

- (CGC-5): to create favourable moral environment in the organization of research...
- (CGC-6): to create favourable moral psychological environment in the organization...
- (CGC-7): to carry responsibility for decisions taken within professional activity...
- (CGC-8): to lead by example with his/her knowledge, skill and personal initiatives
- (CGC-9): to have knowledge and skills in organizing scientific-research...
- (CGC-10): to apply theoretical and practical knowledge in the field of computer sciences
- (CGC-11): to be able to independently obtain new data...
- (CGC-12): to apply the conducted outcomes of researches...
- (CGC-13): to teach ability to conduct experimental researches...
- (CGC-14): to analyse and to critically review obtained scientific, statistic and other data
- (CGC-15): to use efficiently modern information and communication technologies...
- (CGC-16): to present new scientific projects to be implemented...
- (PC-1): to use modern scientific achievements and advanced technologies...
- (PC-2): to select new research methods and to apply them in resolving problems...
- (PC-3): to present the application of the outcomes of scientific-researches...
- (PC-4): to be able to work with innovation technologies...
- (PC-5): to organize experiments in computer sciences...
- (PC-6): to carry out independent analysis by using international experience...
- (PC-7): to present the outcomes of scientific searches...
- (PC-8): to implement practical work by applying modern information technologies...
- (PC-9): to compile and to present, in line with modern requirements, the works using modern information technologies
- (PC-10): to apply modern information-communication technologies in different areas
- (PC-11): to determine general forms and consistent patterns on subject groups
- (PC-12): to apply advanced technologies in the analysis of the issues of computer sciences
- (PC-13): to adapt scientific innovations emerging in the field of computer sciences...
- (PC-14): to lead scientific research work of staff
- (PC-15): to identify new computer science projects, to develop new regulatory technical documents
- (PC-16): to resolve unexpected and complex issues in their professional activity...
- (PC-17): to propose and plan relevant activities and methods,...
- (PC-18): to determine activity or education related issues in a creative manner...
- (PC-19): to select relevant technologies and methods in the course of resolving work...
- (PC-20): to self-critically evaluate personal behaviour in the course of resolving work...
- (PC-21): to present and to justify work and education related problems in Azerbaijani...
- (PC-22): to independently work in complex and unexpected circumstances
- (PC-23): to take responsibility for the strategic activity of organization or groups
- (PC-24): to behave in line with ethical rules in complex situations...
- (PC-25): to evaluate personal and others' needs for continuous training...
- (PC-26): to teach subjects at bachelor level of higher education...
- (PC-27): to obtain relevant scientific and technical information of specialization from textbooks of different areas of computer sciences…

- (PC-28): to communicate his/her knowledge to others through teaching, training...

It could be stated that out of 28 modules, there is not even one which contains a course in Computer Science.

*8.9.3 General conclusion*

By the end of a bachelor's degree, students should know well how to program, but they are not skilled enough to be engineers or to start on a PhD. Therefore, a **master's degree** should **contain advanced courses in Computer Science** to get to the necessary level.

**Recommendation:** Without considering the internship, courses in Computer Science should represent at least 70% of the contents of a Bachelor or a master's degree, not the current 17% for bachelor's degree and 0% in master's degree.

**8.10. Recommendations on the implementation of the recommendations**

The current master's degree curriculum (see "Annex_New state standard for Computer Sciences_BA&MA.docx" file and above) does not support students to acquire the whole range of competences to become engineers or researchers.

This means that the contents recommendation in the present document and in "Annex_New state standard for Computer Sciences_BA&MA.docx" will probably take several years to be implemented, because of the radical changes that are needed and because of the lack of properly trained academic staff. A possible way to improve University teaching would be to *gradually* introduce new courses from the proposed curriculum on 3 to 5 years.

It shall be stated that the proposed curriculum mostly focusses on ***essential*** Computer Science courses:

- this leaves many ECTS for current courses to be still taught as the curricula are evolving (it is necessary for current teachers to still be able to teach what they are used to teach),
- even when the full proposed curriculum is implemented, it is possible for universities to develop their autonomy and diversity, with:
    - one university developing a master's degree on Network programming because it develops a research team working on Computer Networks,
    - another University developing a master's degree on Computer Graphics because it develops a research team on Computer Graphics,
    - another one on Artificial Intelligence, depending on the priority of the Ministry of Education...

Some principles were successfully experimented at the Franco-Azerbaijani University in Baku (UFAZ), which does implement the recommended courses in "Annex_New state standard for Computer Sciences_BA&MA.docx". Best practices sharing between UFAZ and other universities could eventually be put in place. It could also be envisaged creating textbooks at UFAZ and sharing with other Universities.

**Sum-up of recommendations** (to be implemented gradually):

- **Try to increase the skills of University Professors** (at least a PhD for teaching in Master's degrees) and try to give them a better salary because Computer Science is a very competitive job market (computer scientists with a PhD are likely to have much better salaries in private universities or in companies than in current state universities);

- **Gradually introduce new courses from** "Annex_New state standard for Computer Sciences_BA&MA.docx" in Universities as new skilled professors (if possible, holding a PhD diploma) are recruited.

- **For bachelor's degrees:**

  - Make sure that (within 3 to 5 years), **70% of the courses are about Computer Science** (and not 17% as is currently the case).
  - Make sure that **all courses contain at least 50% applied teaching**, in the form of Supervised Work (exercises sessions, tutorials) or Practical Work (hands on sessions on a computer), with evaluations containing at least one mini-project to be handed back (with a 5-7 pages written report) or a powerpoint presentation in front of the classroom.

  - On the last year, **introduce (or reinforce) 10-14 weeks internships in companies** where real work is done by the students, with University and **company tutors and a 20-30 pages report** and **a defence to give a grade.**

  - Please also note that in France (and in many other countries), secondary studies stop at 18 years old, while in Azerbaijan, secondary studies stop at 17. Therefore at UFAZ, for example, bachelor's degrees are done over 4 years, while it is 3 in France or several European countries.

- **For Master's degrees:**

  - Make sure that (within 3 to 5 years), **70% of the courses are about Computer Science** (and not 0% as is currently the case).

  - Try to develop **research laboratories and research teams** in all Universities so that Master's degree teachers all hold a PhD and are researchers at top level on their topics, that they can then directly teach to Master's degree students, which also means that different universities will teach different Master's degrees.

  - Because these teachers are also researchers, their task should be 50% teaching and 50% research. 50% teaching is a maximum because master’s degree will create the need for research internships and company internships tutoring. Then, if they are researchers, they also need time for their own research and PhD students tutoring.

- The Master’s degree contents of “Annex_New state standard for Computer Sciences_BA&MA.docx” may be common to all universities, with additional specialty courses coming from the research laboratories in Universities.

- Encourage the diversification of Universities specialties towards different topics, so that each University can become the best in one (or several) chosen domains (there are so many different domains in Computer Science, that **one master’s degree cannot encompass all Computer Sciences domains**).
- On the last year, introduce (or reinforce) 20 weeks long internships in companies where real work is done by the students, with University and company tutors and an internship 30-50 pages report and a defence to give a grade.

In France, every 5 years, universities propose to the Ministry of Education a new curriculum (because the field of Computer Science evolves constantly). **New curricula must be approved** by experts at the Ministry of education so that State Diplomas can be delivered by the Universities. This allows both Universities to have different specialties, but also a control by the Ministry to make sure 70% of the courses are about Computer Sciences, with at least 50% applied computer science in each course.

### 8.10 Recommendation for future missions

Yearly missions over 3 to 5 years could be programmed to evaluate the progress recommendations implementation.

## 9. Acknowledgments (if any)

Acknowledgements to Lisa Bydanova and her team at Azerbaijani Ministry of Education for their efficiency and their patience with waiting for this report, that was initially 5 pages long, then 6, then 10, then 20 pages long (+ 33 pages long annex), as I saw the extent of the work to be done.

## 10. Annexes (if any)

- Annex_New state standard for Computer Sciences_BA&MA.docx: File that contains proposed contents for a Bachelor and Master’s degree curriculum in Computer Science and Computer Engineering. This annex also contains snapshots of a standard Bachelor’s degree and a Master’s degree contents at Strasbourg University (validated by the French Ministry of Education).
- Modalities for 1st year's Master's degree in apprenticeship.pdf: File that contains evaluation and functioning modalities for the 1st year of a French Master’s degree in apprenticeship.

- Modalities for 2nd year's Master's degree in apprenticeship.pdf: File that contains evaluation and functioning modalities for the 2nd year of a French Master's degree in apprenticeship.
- Modalities for standard 1st year's French Master's degree.pdf: File that contains evaluation and functioning modalities for the 1st year of a standard French Master's degree in Computer Science.
- Modalities for standard 2nd year's French Master's degree.pdf: File that contains evaluation and functioning modalities for the 2nd year of a standard French Master's degree in Computer Science.