

Chapter 1

The Arcadia of Innopolis



One aspect that strikes foreign visitors when they first come to the campus is the impressive real estate. A common impression of new visitors, since the early days, is that it had to exist for at least 20 or 30 years, while indeed it is only a few years old. What happened before the construction of the campus? Innopolis already existed as a small office on two floors in a business center in Kazan downtown. This chapter presents some memories from this Arcadia where everything had a beginning.

1.1 The Early Days of Innopolis

Arcadia in literature is a physical or metaphorical place associated to harmony with nature. The term is derived from a Greek province and over centuries developed into a poetic idea of an idyllic vision of unspoiled wilderness. We like to adopt here this concept when referring to the early days of the vision of Innopolis University and the vast land where the city was later built, at least how it looks like in summertime. The reader will give us this concession at the opening of the curtain. Figure 1.1 shows one of the many pictorial representations of such an ideal idyllic place and time.

In practice, when one of the authors (M. Mazzara) visited Kazan for the first time, in December 2013, the site where the city of Innopolis has now settled was only a vast land covered by snow. Literally, a greenfield in the hills of Tatarstan, or better to say a *whitefield*.

Great plans and visions were on the paper, but it would have been hard to predict that such a vision could actually have been implemented to such an extent in this short time. Even for the most visionary and optimistic person. But indeed it has been. The city is there; you can see it already coming when you cross the Volga bridge on the M7 highway. The major Technopark is visible on the right, on top of the hill. This, together with the beautiful Volga landscape, creates a feeling of



Fig. 1.1 Dream of Arcadia, Thomas Cole (1838). Public domain, via Wikimedia Commons

anticipation on the new visitors. You can share these sensations when you bring newcomers to the city by car, from the airport. The expectation is growing little by little all along the trip and the landscape transformation: from a modern hi-tech city like Kazan, to the idyllic countryside of the region, to then turn again into a modern scenario where you would not expect it.

In the early days (2013–2015), the first operational center of the university was based in Kazan, where a small group of about 30 people started to develop the concept and the early educational programs: the first idea of bachelor program and the inherited Software Engineering master program developed together with CMU university. The first bunch of 14 master students spent 1 year in Pittsburgh, USA (2013–2014), in order to graduate in the program and being trained for then delivering the program in Kazan, as local teaching assistant. Some professors were hired and sent for training to the USA too. This was a massive logistical and financial operation that allowed the university to start up in an effective manner, getting also the attention of the national media and worldwide.

In the small operational center of Kazan, the core of the first bachelor program was designed with long discussions in the only meeting room initially present there. At the time, one of the key partners of the university was ETH, the polytechnique school of Zurich, and one of the visiting professors was Bertrand Meyer,¹ presently an emeritus professor of ETH and still visiting professor of Innopolis. Professor Meyer was instrumental in the startup phase of the organization, as well as David

¹ https://en.wikipedia.org/wiki/Bertrand_Meyer.



Fig. 1.2 First international meeting of SE lab

Vernon² who was also visiting professor and the main consultant of the provost of the time, Tanya Stanko. These two prominent figures had a strong influence in the creation of the early BS and MS programs.

Given the presence of Meyer, and as a natural consequence, while the Master in Software Engineering was inspired by CMU (and later the Master in Network and Security was designed together with the University of Amsterdam), the bachelor program got many elements of inspiration from the Swiss Program.

Figure 1.2 shows one of the early meetings of the Software Engineering lab in November 2014. Figure 1.3 represents a genuine discussion between Bertrand Meyer and Salvatore Distefano, another one of the early visiting professors (now at the University of Messina), about the application of design by contract [11] to non-functional requirements.

During those months, many pedagogical discussions had to be put on the table, with frictions and discussion sometime, as it can happen in every human environment. We were trying to develop a program that would offer all the classic foundations of the program presented by top universities, but at the same time offering some element of uniqueness, some kind of cherry on top of the pie. The solution to this problem appeared relatively early studying the program of ETH. However, it was not exempt from troubles.

² [https://en.wikipedia.org/wiki/David_Vernon_\(professor\)](https://en.wikipedia.org/wiki/David_Vernon_(professor)).



Fig. 1.3 Dinner discussion

It is worth mentioning the decision about this point of uniqueness among all the other pedagogical issue. This choice costed an enormous amount of time of discussion and sometime the (we believe, unjustified) dissatisfaction of students. Eiffel was chosen as the first programming language to be thought to our students, the introductory language. This was the cherry on the pie we were looking for, this element of uniqueness. Among other supporting elements, this was exactly the same choice made by ETH (at least until 2016).

1.2 Eiffel as First Programming Language

The choice of Eiffel and design by contract as programming and methodological tools for first year bachelor has been long discussed and at times opposed inside the university. Ex-post, we can confidently affirm that it was the right choice. However, it is worth here spending some time to motivate the decision.

Which programming language is better to start studying for the beginners? There is no single correct answer to this question, no common agreement on this, and debate in some cases is vibrant. There is no global consent between programmers, teachers, academics, and business people. The reason is simple: the question is incorrectly formulated. For comparison, imagine that the question is not about programming languages but about human languages: what is the best foreign language to learn? Clearly, there is no single correct answer. The answer depends on such factors as the student's mother tongue, his cultural background, his objectives and motivation (professional or leisure?), and others.

The same goes for programming languages: no single answer exists. In general, it is easier to answer to the opposite question: “What programming language is better not to start with?”. Experience has shown that teaching a specific language from scratch in order to satisfy a specific and urgent need rarely brings the individual to develop into a great professional. There are, for example, many cases of individuals who improvised themselves as Visual Basic programmers from scratch or moved from FORTRAN or COBOL to Java because of some local business need. Even if they managed to patch the immediate emergency, they rarely developed into great professionals, because they lacked the correct mindset and basic skills required by an experienced professional. Sometimes, such an emergency is inevitable, but developing a quality curriculum for a top-level university requires more care.

Worldwide, examples of good pedagogical approaches for programming are not missing. There are a few preliminary considerations. First, what programming paradigm we want to use? There is a general tendency to prefer the object-oriented programming (OOP) paradigm as starting point since it has proved to develop the right abstraction skills and methodological attitude. This approach, however, is not without its critics: some believe that object orientation may deal too much with design and interface aspects and not enough to algorithmic details and imperative flow structure. According to this view, procedural programming would be better to start, while object orientation should be introduced in advanced courses. Of course, this depends on how the course is taught, but the concern is serious. The school of thought privileging OOP usually concentrates on languages like Java or C# in order to take into account business demand. The school of procedural programming sometime concentrates on purely academic languages, like Pascal, with the benefit of simplicity or widespread languages, like C, offering broader flexibility (and related complexity). There are other paradigms too, for example, the Functional (Lisp, ML, Scala, etc.), which has attracted renewed interest in recent years, and Logic (Prolog).

The second general observation is that any serious computer scientist or software engineer will learn many languages over the course of his career. In particular, almost everyone will learn one or more of the dominant languages such as, today, Java, C#, C, C++, or Python. So the choice of the first programming language is not exclusive of others; rather, it is a preparation for others and should emphasize the development of the skills needed to learn programming. (In fact, an increasing number of students have done some Java or other programming before they even join the university program.)

A related question is: “how much emphasis we want to give on formal reasoning and correctness by construction?”. University is the ideal time of life for learning new things and, at the same time, build the foundations of one’s knowledge and mindset. Establishing a broad and deep basis is also the best way to make sure that students not only receive sufficient initial training to obtain a first job but acquire the extensive long-term intellectual skills to pursue a successive career over several decades: the technologies will change, particularly in such a quickly evolving field as Information Technology, but the principles will remain. As result, a broad school of thought supports the idea that the introductory programming course and the

first programming language should emphasize computer science foundations and formal reasoning at the time of learning the first language, to strengthen a mindset leading to the development of better software. Our experience of teaching Eiffel in Introduction to Programming at IU implies the choice of OOP and an affirmative answer to the second question, and still Eiffel is only one of the possible choices to which these decisions lead. This path is not free of controversy. The experience inherited from ETH Zürich is positive, and the course was well taken by students. We aim at repeating the success in different contexts, though an adaptation phase is necessary and benefits of the approach may not appear as immediate.

1.3 Innopolis Today

Innopolis University is now approaching its tenth year and reached a student population of almost 1000 with 400 employees of which about 25 are faculty members and teaching and research staff up to 200 people. There are plans to reach 2000 students in the incoming years. The university is an integral part of the city, which has now two operational technoparks and two more under construction (at the moment of writing). The flow of students between the university and the companies of technoparks is regular and growing, fulfilling what was, since the beginning, one of the key measures of success: cooperation with industry and supply of highly qualified professionals.

The university has an extensively developed network of international institutions collaborating under different formats: student exchange, Erasmus +, visiting professors, joint PhD supervision, joint projects, and summer internships offline and online. One of the collaborative projects has seen as partners CERN and Newcastle University [2, 4], and collaborative PhD supervisions involve several universities including Toulouse, Nice, and Nantes in France, the University of Southern Denmark, the University of Messina in Italy, and the Brno University of Technology in the Czech Republic. All these activities dramatically supported the internationalization of the project. In turn, the growth and internationalization of the university also helped the development of the city itself bringing professionals, students, and talents from abroad, once the city attracted attention worldwide.

With the prerogative of a selective education free of charge and a constant attention to the internationalization of teaching and research, Innopolis is aiming at exploiting the benefit of the global trends without suffering the risks, such as excessive emphasis on the market aspects of education and the death run to global rankings which would put at risk the fundamental academic values [8].

During the pandemic, the model of Innopolis University did not suffer particularly as those of other countries. Not relying financially on the stream of foreign students, the struggle of the period was moderate and contained, thanks to the ability of management and employees, who all put extra effort to keep things going.

1.4 Innopolis Tomorrow?

While we are writing, Innopolis city is under continuous development and so is the university. Two of the four technoparks presented in the master plan are already operational, and two are under construction. Two new student dorms are also under construction, and they will double the capacity of the four existing buildings. New townhouses and apartments will also be soon available. The plan is, in the incoming years, to increase the capacity from about 1000 to about 2000 students. This significant growth requires an expansion in terms of staff and in particular faculty. The challenges for faculty growth are now different than in 2013, but not easier. The pandemic which started in 2020 changed the market and the perception of people and to some extent their ability or willingness to move. At the same time, many universities which relied mostly on foreign students for their income (e.g., the UK and Australia) are experiencing financial troubles. Innopolis University had a different financial model since the beginning that now appears to be successful and may have an advantage out of the global situation.

In the remaining chapters of this book, we will analyze some of these challenges, in terms of professors recruitment, faculty growth, student enrolment, and university development. We will look at the present and the future of the university and, as a consequence, of the city.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

