Top Five Challenges to Software Projects in Palestine

By Huthaifa Afanah

For the past 15 years, many have considered the software industry in Palestine to be promising. I, however, take this claim to be only partially true. The software industry will fall short of its potential if we fail to identify the issues that are holding back its growth in the developing Palestinian economy and do not take corrective measures very soon. This article will outline the top five reasons for the failure of software development projects in Palestine. Some will consider my judgment too harsh, others will criticize me for having omitted important aspects, and I admit that in some cases I do not have the solutions for these problems. But if we acknowledge what is wrong, we can work together to try to remedy the situation.

1. Poor Software Engineering Practices

In general, Palestinian software-company owners, senior developers, consultants, and software project managers possess higher education degrees and international experience in the software development domain. However, when software-engineering knowledge is to be applied to practice, the track record is less impressive. Local software vendors can choose agile software development approach to develop software systems through collaborative efforts of cross-functional teams, providing flexible and continuously developing responses. Alternatively, they may utilize the traditional waterfall model, where in sequential manner a software program moves through various stages that follow a clear order, moving from the identification of requirements through design, implementation, testing, and maintenance. In Palestine, complex and national-level software projects have been initiated with improper requirement analysis, inadequate testing, insufficient understanding of end-user needs, and lack of long-term software-extensibility planning. Over the last ten years, we have simply not witnessed enough progress in the application of software-engineering practices in the local software industry.

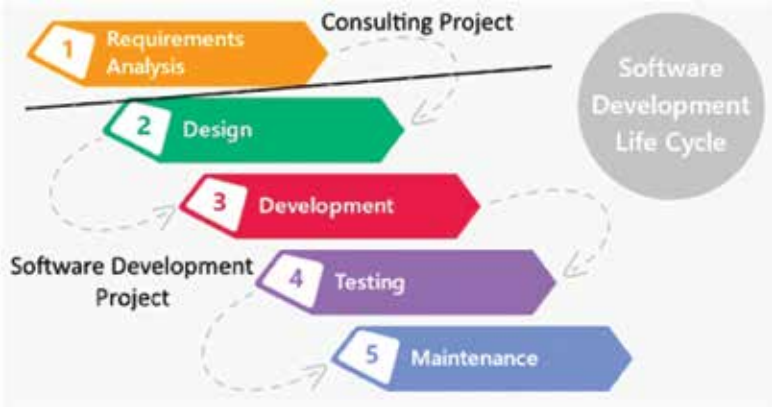
In 2010, for example, I arranged for the first Scrum training which was successfully carried out in Palestine: Coached by the finest Scrum trainers in the world, around 30 software engineers were graduated as Certified Scrum Masters, having gained skills that enable them to successfully cooperate with global teams and work collaboratively on agile-based software development. But since then, no company has asked to conduct another training course (e.g., training for Scrum Master, Scrum Developer, or Scrum Product Owner), nor has any software-development company sent staff to pursue further training in this area. However, agile development is on the rise throughout the world.

> The local software industry must have a greater appreciation of software-engineering best practices.

2. Inappropriate Procurement and Contracting Practices

As the Palestinian government and related organizations (e.g., municipalities and semi-governmental organizations) are the large clients in Palestine, these institutions will be the focus of this point in my discussion. Here, procurement is generally handled either directly by the governmental entities or through donor-related bodies such as foreign consulting firms or NGOs. Whereas procurement and contracting procedures vary widely, they constitute more often than not fixed-cost contracts that fit a waterfall approach; none are flexible enough to support the use of the agile development methodology.*

Using a fixed-cost contract can work, but it is contingent on the vendor being given the ability to estimate correctly the project costs (e.g., effort and money). In order to be able to deliver what is needed on time and within budget, requirements must be clear, well-defined, and agreed upon among all parties. In the local Palestinian market, in many cases, terms-of-reference (TOR) documents do not cite enough technical detail, and the requirements



are not clear enough. Moreover, it has become common practice to ask the software vendors to carry out the gathering and analysis of information to determine requirements under the same contract, which inevitably leads a project to fail.

In fact, as they are searching for software solutions, many clients have no clear understanding of when they must procure services as opposed to goods; for example, when customer-relationship-management (CRM) systems are procured and implemented for ten organizations of different sizes, even when they are offering similar services. Customization requirements in such cases necessitate procurement of both goods and services.

My company was invited to participate in a tender where a donor-funded NGO, which was committed to applying international procurement practices, asked to implement within 5 months a business-process management solution for 22 of its beneficiary organizations. The sponsor considered this project a tender for a procurement of goods, and software vendors were asked to provide a single price tag in the form of $X for all services rendered. Given the scope of the project, neither a simple procurement-of-goods approach nor a one-price-for-all tag was feasible. However, the sponsor team insisted that this was the way to go. Most vendors did not argue the point, and the sponsor received several bids. Almost two years later, the project was still overrun and the quality of the implemented solutions not up to standard. Nevertheless, the sponsor has applied the same approach to other software projects!

3. Poor Project and Program Management Practices

When a building collapses while it is under construction, investigators will usually categorize the cause under one of two main scenarios or a combination thereof: Either a foundational issue caused the collapse, whereby the building's initial design was flawed; or the execution was flawed, whereby an incident or a series of incidents led to the building's collapse. The same logic applies to software project failures. Software projects do not happen in isolation. Palestine is a developing country under occupation, and software projects are usually part of larger development programs that may target the government, independent public institutions, community-based organizations, municipalities, or the private sector. One of the most common mistakes in software design is poor planning for the interdependence between different projects in the same or aligned programs.

> At times, software vendors do not speak out against required procedures that are counterproductive out of fear that their criticism will be counted against them and negatively affect their chances of winning this bid or future bids.

For example, a donor decided to implement unified software for the 14 chambers of commerce in Palestine. One single software was to facilitate information exchange, simplify the interaction between business owners and the staff of the various chambers, and assist decision-makers with future economic planning. But when each chamber office asked for a major modification to its version, not one but 14 different systems were in fact created, defeating the original purpose of the project. A few years later, program management figured out the root cause and the decision was taken to re-engineer and standardize the software services. Whereas the software vendor's unrealistic can-do attitude resulted in a shortsighted approach that lost sight of the overarching purpose, all parties involved had contributed to the failure by underestimating the complexity of the program.

Project management is the glue that binds together all the moving parts. The importance of having a truly specialized project manager who possesses a solid understanding of all project technicalities is frequently underestimated in local software projects. Generally, clients elect someone from their institutional management ranks to handle the project during the pre-contracting and contracting phases; at a later stage, the software vendor is asked to provide a project manager to run daily tasks, which in most cases will be someone whose specialization constitutes a senior technical capacity. In neither case are sufficient project management skills the measure by which the selection can be justified. To compensate for this lack of specialized knowledge, two styles of project management may be employed – and both will lead to the failure of software projects in Palestine. 1. Over-management involves overloading the project with processes, paper work, and lengthy approval cycles on a micro level. The obsession with a rigid process causes delays in the execution and entails enforcing meaningless formalities of communication, e.g., reports, documents, and forms. 2. Reactive management lacks an underlying framework, and ad-hoc procedures dominate the management, preventing projects from achieving their full potential.

4. Lack of Proper Technical Software-Development Capacities

The lack of technical software capacities creates issues that manifest themselves in different sizes and shapes. I believe that there are three main cavities that will lead a software project to failure: First, poor software design and architecture will restrict the software's ability to update and adapt in order to suite the client's needs. Second, poor software quality may result from an absence of quality assurance (QA) across the project's life cycle. Third, it is very important to select the correct technology for the work at hand.

Of a more strategic nature is taking the correct decision about whether to use an off-the-shelf software program or develop a custom software solution from scratch, the typical buy-versus-build debate. Off-the-shelf software-implementation services are not widespread in Palestine, mostly due to the fact that clients worry about the associated high licensing costs. Others claim unrealistically that their business needs are unique and require a custom-built solution. But in many cases, implementing an off-the-shelf package implies less risk and requires far lower total costs than developing custom software.

5. Corruption and Power Monopoly

Corruption is a global, border-crossing problem and not exclusive to the third world. We generally (should) admit that the Palestinian economy is affected by crony capitalism, and (based on my experience in the local market over the last ten years) I dare say that the IT sector is not much different. Corruption and power monopoly can threaten the success of a software project in multiple ways. First, choosing a software vendor based on favoritism does not place enough emphasis on whether the selected vendor is qualified to carry out the project, which may jeopardize a successful outcome. Second and unfortunately, corruption and power monopoly can be reason and "mechanism" for the failure of a software project, when obstacles may be deliberately created to put spikes in the wheel of, for example, a transparency-serving system. The resulting chaos in the absence of effective software and the dependence on paperwork instead serve to create

TOO BUSY TO IMPROVE?

an environment that is suitable for corrupt officials to hide their trails. A plan to implement software that increases transparency, proposes tight controls, and enables effective audits may be conceived as a real threat by some. Third, one of the main corruption threats for software projects is not embezzlement, the accepting of bribes, or a conflict of interests. The main corruption-related threat is the presence of under-qualified staff in decision-making positions in government offices, NGOs, donor organizations, and the private sector. This is the real threat, as it causes projects to fail, wastes millions of dollars, and delays progress – not only of the software sector – for many years.

Among these top five reasons, corruption and power monopoly constitute the problems that the industry itself can do very little to fix. It is the government's role to fight corruption and eliminate power monopolies. But given the complexity of the political situation in Palestine on both internal and external levels, progress on this front may be slow to come, I am afraid.

It is easy to say that things are wrong, but solutions generally are more valuable than criticism. It may be important to consider whether Palestinian software engineers and vendors have thus far accepted the ailing status quo as a substitute for better knowledge. We must educate our clients and workforce on how a software project should run, and we must be bold and honest enough to admit that by failing to insist on proper procedures and analytical groundwork, we create a snowball effect whereby everyone, from clients to software engineers and vendors, will be harmed! After all, "A problem clearly stated is a problem half solved."

Huthaifa Afanah is an entrepreneur and a business-development and IT project-management professional with more than 12 years of experience in leading business operations from inception to execution. He directs operations in new business development and technology project management for clients across the real estate, government, and education sectors in Palestine. His experience in the Palestinian software-development industry climbed from junior application developer to team leader and project manager before he eventually started his own software development firm.

---

* I will not get into the debate on the advantages of waterfall vs. agile systems. Procuring a software project under a fixed-cost contract is common worldwide, although it brings multiple challenges. Software development is an inherently unpredictable process, uncertainty of requirements is unavoidable, and some degree of failure is inevitable in any project.