Software Development in Disruptive Times

**CREATING A
SOFTWARE
SOLUTION WITH
FAST DECISION
CAPABILITY,
AGILE PROJECT
MANAGEMENT,
AND EXTREME
LOW-CODE
TECHNOLOGY**

JOÃO VARAJÃO, ALGORITMI RESEARCH CENTER,
UNIVERSITY OF MINHO, PORTUGAL

The current pandemic has brought challenges rarely seen before. It has made evident a world that is strongly globalized, capable, and characterized by a high interdependence of resources and means, but that is also fragile and has a high potential for contamination—not only in the physical sense but also concerning information, ideas, processes, and other aspects.

Given the novelty of the situation, one may be tempted to think that this is a unique situation that will soon be overcome, returning eventually to the (apparent) stability that existed previously. However, the reality indicates that this view is, at best, illusory and that we live in an age in which societal fragilities and instabilities will be increasingly evident (optimistically, awareness of them will also become more acute). In other words, crises have always been part of human evolution, and they must be seen as inevitable and recurring realities that need quick and effective responses. The key is to be prepared for them and act accordingly.

As history has so often demonstrated, difficult times enhance society's ability to adapt, and lead to the search for better solutions. Information technologies, which in recent decades have revolutionized the lives of people and businesses—sometimes more or less quietly, sometimes with a bang—are inevitable since they provide cost-effective solutions to the increasingly complex problems of an interconnected and interdependent world. This is easy to understand from a simple example: if, in this pandemic, there had been a global shutdown of the technological infrastructure that supports the Internet, the world would indeed be experiencing a much more complicated and chaotic reality than we are living—and it is already quite hard for everyone.

It is in this context that the software-development project described in this article is worth reporting on, since it involves several disruptive aspects that are fundamental in a world that requires solutions “thought today” to be “made available yesterday.” From the point of market-opportunity awareness to the availability of a fully functional software product, this project took three weeks to complete and involved several state-of-the-art practices and tools: fast decision making; agile project management; and extreme low-code software-development technology.

FROM OPPORTUNITY AWARENESS TO THE DECISION TO GO FORWARD

On March 12, 2020, the coronavirus outbreak was officially declared a pandemic by the World Health Organization. On the same day, the CEO of Quidgest, a medium-sized

From the point of market-opportunity awareness to the availability of a fully functional software product, the project took three weeks to complete and involved several state-of-the-art practices and tools.

software house, identified the pandemic as an opportunity for the company and wrote a plan for creating a web-based software product, together with a preliminary list of requirements (related to monitoring, control, innovation management, and eradication of diseases). This plan was then sent to the sales and marketing teams (both national and international), as well as the research and development teams for feedback. Note that the sales and marketing teams had developed several contacts with government administrations, hospitals, retirement homes, etc., that provided broad insight into the project's market potential.

The response from all teams was quick and positive, so the company decided to go forward with the project starting on March 13 (one day later).

PRODUCT, REQUIREMENTS, AND DEVELOPMENT MILESTONES

The product, named VIRVI – Health Vigilance and Control Software—is presented as “hyper-agile emergency software for global epidemiologic challenges.” As described by Quidgest, VIRVI is “an information system aimed at supporting the monitoring and control of a virus epidemic, like Covid-19, in any country or region, in an emergency timeframe (that is, starting operating in hours). VIRVI is robust, reliable, and capable of continually evolving, forming the basis for good critical management and communication facing virus epidemics.”

As depicted in the timeline in figure 1, VIRVI evolved quickly. On March 12, the project started with an estimated size of 207 function points, which measure software size

FIGURE 1: **PROJECT MILESTONES**

defined as the amount of business functionality that an information system (as a product) provides to users; the complexity level was 941 (according to the company's internal measure that takes into account the number of data tables, data fields, foreign keys, arrays, formulas, forms, form fields, menus, multilanguage, and other aspects, as well as maintenance and documentation requirements). At the product launch three weeks later, the total implemented function points were 1,365, with a complexity level of 4,387.

This shows the high volatility of requirements, regularly updated during the project execution following the evolution of the pandemic and the forecasted market opportunities. After looking at national and international best practices and trends, Quidgest decided to focus its solution on nursing and retirement homes (a priority in fighting the pandemic), national health authorities, hospitals, and health laboratories.

The first fully functional prototype of VIRVI was

made available on March 22, and in the following week, it was presented in four countries, with three formal demonstrations and two leads generated. This garnered feedback from partners and led to the identification of additional required features. Furthermore, it confirmed that it would be possible to respond quickly to each client's particular needs and accommodate new requirements.

Some of the final features of the software are the following: registration and data collection in a centralized information repository; coordination among entities involved in responding to the epidemic, with different access levels; monitoring and decision-making in realtime; availability of open data for external analysis; data visualization, with the provision of indicators, statistics, and maps in realtime; support for official emergency channels; direct communication between citizens/entities and health professionals; and a centralized marketplace for entities and suppliers.

VIRVI was ready to launch April 3, three weeks after the start of the project. The final product has 117 data tables, 506 data fields, 156 foreign keys, 31 arrays, 115 formulas, 106 forms, 867 form fields, and 274 menus.

TEAMS, PROJECT MANAGEMENT, AND DEVELOPMENT TECHNOLOGY

How did Quidgest manage to get a fully functional product in only three weeks, developed in a highly unstable environment with requirements evolving daily and with the described complexity? This section provides some insight.

In the kickoff of the project and during the first two weeks, the team consisted of two full-time system

analysts/developers; one part-time (25 percent) UX (user experience) designer; one full-time project manager; one part-time (50 percent) development-technology expert; five part-time (25 percent) people from the marketing team; and four part-time (25 percent) people from the sales team. After the first two weeks, the team changed to two part-time (50 percent) system analysts/developers; one part-time (50 percent) project manager; one part-time (25 percent) development-technology expert; one part-time (50 percent) public-health consultant; four part-time (5 percent) people from the marketing team; and four part-time (20 percent) people from the sales team. Taken together, this corresponds to about 3.25 FTEs (full-time equivalents) for the development team and nearly 1.84 FTEs for the marketing and sales teams during the three-week development timeline. Considering the total of 1,365 function points for the final product, the average weekly productivity of the analysts/developers/technology expert was nearly 220 function points.

Since there was no conventional enactment of requirements, the project began based only on the “perception” of what features would be important to the product (based mainly on the CEO’s and the company’s experience in the area, as well as a review of the literature and news). The team also had members with training and expertise in the health sector, and several contacts were made with specialists in health care. Later Quidgest’s health coordinator joined the team, promoting the establishment of a voluntary consultancy scheme. The result of all these voices was a daily evolution of the software requirements. In the end, the actual

A **nother ingredient that was crucial to the success of the project was the low-code technology adopted for software development.**

requirements came from a combination of the company's experience, contact with specialists, news from the media, scientific studies, and official reports on the pandemic.

Given the project's characteristics, the company's standard project-management approach was unfeasible in this case since the market window of opportunity was very tight, and the needs of different stakeholders had to be accommodated in the final analysis. Thus, it was necessary to take the agile approach to project management to the extreme because of the need for accepting new requirements almost daily.

The Planner in Microsoft Teams was used to support the process, focusing on teams, tasks, and priorities (established according to the milestones). A Kanban board was created, which could be edited by the whole team, with six states: (1) tasks under analysis, (2) pending, (3) in progress, (4) executed, (5) for testing, and (6) finished. Regular team meetings were held to review the tasks regarding priorities, obsolescence, and clarification of requirements. Another striking aspect of the project was that it was carried out entirely remotely (as of March 16, the company started to work remotely due to the pandemic lockdown restrictions).

Another ingredient that was crucial to the success of the project was the low-code technology adopted for software development. Taking into account the enormous volatility of the requirements and the need to provide fully functional solutions quickly, the challenge was how to achieve this without compromising the quality of the software, the documentation, or further required maintenance.

In this context, the power and usefulness of low-code or extreme low-code technologies become even more evident. These technologies currently include the IBM Automation platform, Zoho Creator, Appian, Mendix, OutSystems, AgilePoint, Google App Maker, Nintex, TrackVia, Quickbase, ServiceNow, Salesforce App Cloud, Microsoft Power Apps, Oracle Visual Builder, and Oracle APEX (Application Express), just to name a few. The distinctive feature of these technologies is their ability to create software applications with minimal hand-coding. For the project, Quidgest used Genio, its proprietary platform. Genio is an extreme low-code (between low-code and no-code) development technology. The development is pattern-oriented, and Genio has code-generation features based on modeling (model-driven engineering).

The chosen technology allowed evolutionary versions of the software to be released on a daily basis. In this way, the development process could be streamlined. Even with the requirements constantly evolving, the next day there could already be a functional solution to support them. The advantages of low-code technology are also fundamental to the maintenance of solutions. The benefits are, perhaps, even more noticeable in this context, by making it possible to support the continuous evolution of requirements without compromising the quality of the software artifacts (including documentation).

Overall, the objectives defined for this project were met despite the high risk caused by the instability of requirements and application scenarios, as well as the urgency of the solution. In a short period of time, Quidgest's project not only gave rise to several sales leads

but also contributed to organizational learning and the visibility of the company.

CONCLUSION

In this project, the challenge was to deploy “software faster than the coronavirus spread.” In a project with such peculiar characteristics, several factors can influence success, but some clearly stand out: top management support, agility (in decision and management), understanding and commitment of the project team, and the technology used. Conventional development approaches and technologies would simply not be able to meet the requirements promptly.

The project described here reflects the demands currently placed on companies in terms of decision and action capacity. It combines market vision and rapid decision-making capacity with action. The company identified an opportunity, defined a project, and decided to move forward, structuring and organizing a team by adopting a different approach to project management—a streamlined agile approach—and adopting a proactive marketing posture. Without technology that supported the rapid development and deployment of software, however, the project could not have been achieved in such a short time—in a context of high instability and rapid evolution of requirements.

A study published by IBM more than ten years ago, “The Enterprise of the Future – Implications for the CIO” (2009), stated, “The enterprise of the future is hungry for change, innovative beyond customer imagination, globally integrated, disruptive by nature, genuine, not

just generous.” These are, more than ever, fundamental characteristics for today’s organizations, to which could be explicitly added, “supported by stable as well as disruptive information technology.” Low-code, extreme low-code, and no-code software development, supported by innovative technologies such as artificial intelligence, certainly have influence in this scenario and are expected to accelerate rapidly toward worldwide adoption as major enablers of digital transformation.

João Varajão is professor of information systems and project management at the Department of Information Systems of the University of Minho, Portugal. He is also a researcher at the ALGORITMI Research Center. His main research interests are in information systems project management and information systems development (addressing, particularly, the success of project management and projects). Before joining academia, he worked as an IT/IS consultant, project manager, information systems analyst, and software developer for private companies and public institutions. He has served on numerous committees of international workshops and conferences. He has been published extensively and serves as editor-in-chief, associate editor, and editorial board member for international journals.

Copyright © 2021 held by owner/author. Publication rights licensed to ACM.