EUROPEAN ACADEMIC RESEARCH
Vol. IX, Issue 12/ March 2022

Impact Factor: 3.4546 (UIF)
DRJI Value: 5.9 (B+)

ISSN 2286-4822
www.euacademic.org

DEVOPS and Agile Methodologies in Brazil's Largest Urban Microcredit Project: Applications and Benefits

FILIPE DE PAULA SILVA[1]
CÉSAR AUGUSTO BORGES DE ANDRADE[2]
JOÃO PAULO ABREU MARANHÃO[3]
RAFAEL T. DE SOUSA JR.[4]

Abstract

This article proposes a model of continuous deployment processes, for an Urban Microcredit Project guided by agile methodologies, unifying development and operations in a single area. An analysis is carried out of the credit market driven by the growing digital transformation of banks who increasingly need agile processes to remain competitive in the market. The article emphasizes a model that can guide the breaking of paradigms of legacy systems and processes that make it difficult to adapt microcredit to the new market scenario. Thus, it is possible to highlight the benefits that the new processes can deliver in order to add value to one of the largest urban microcredit products in Brazil.

Keywords: *DEVOPS, Agile Methodologies, Case Study, SCRUM*

[1] **Filipe de Paula Silva** received his bachelor degree in computing in 2008 from the Pernambuco University, Recife, Brazil. He received degree in Specialization in Software Management and Development in 2022 from the University of Brasilia (UnB), Brasília, Brasil. He has experience in Computer Science, with an emphasis on artificial intelligence, data mining techniques and Agile Methodologies. He currently works at the in a large bank in the northeast region of Brazil, occupying the position of Systems Analyst, acting in several projects for your institution ORCID: 0000-0002-4184-0511

[2] **César Augusto Borges de Andrade** received his bachelor degree in data processing in 1997 from the Mackenzie Presbiterian University, São Paulo, Brazil, and his M.Sc. degree in systems and computing in 2013 from the Military Institute of Engineering (IME), Rio de Janeiro, Brazil. Currently, he is a Ph.D. student at the Graduate Program in Electrical Engineering at the University of Brasilia (UnB), Brazil, researching on machine learning applied to malicious software detection systems. ORCID: 0000-0001-5776-2119

[3] **João Paulo Abreu Maranhão** received his bachelor degree in telecommunications engineering in 2003 and his M.Sc. degree in systems and computing in 2014 both from the Military Institute of Engineering (IME), Rio de Janeiro, Brazil. Currently, he is a Ph.D. in Electrical Engineering at the University of Brasilia (UnB), Brazil, researching on multidimensional signal processing and machine learning applied to network intrusion detection systems.
ORCID: 0000-0003-0632-6434

[4] **Rafael T. de Sousa Jr.** (Senior Member, IEEE) received the bachelor's degree in electrical engineering from the Federal University of Paraíba (UFPB), Campina Grande, Brazil, in 1984, the master's degree in computing and information systems from the Ecole Supérieure d'Electricité–Supélec, Rennes, France, in 1985, and the Ph.D. degree in telecommunications and signal processing from the University of Rennes 1, Rennes, in 1988. He was a Visiting Researcher with the Group for Security of Information Systems and Networks (SSIR), Ecole Supérieure d'Electricité–Supélec, from 2006 to 2007. He has worked in the private sector from 1988 to 1996. Since 1996, he has been a Network Engineering Associate Professor with the Electrical Engineering Department, University of Brasília (UnB), Brazil, where he is currently the Coordinator of the Professional Post-Graduate Program on Electrical Engineering–Cybersecurity (PPEE) and supervises the Decision Technologies Laboratory (LATITUDE). He is Chair of the IEEE VTS Centro-Norte Brasil Chapter (IEEE VTS Chapter of the Year 2019) and of the IEEE Centro-Norte Brasil Blockchain Group. He is currently a Researcher with the Productivity Fellowship Level 2 (PQ-2) granted by the Brazilian National Council for Scientific and Technological Development (CNPq). His professional experience includes research projects with Dell Computers, HP, IBM, Cisco, and Siemens. He has coordinated research, development, and technology transfer projects with the Brazilian Ministries of Planning, Economy, and Justice, as well as with the Institutional Security Office of the Presidency of Brazil, the Administrative Council for Economic Defense, the General Attorney of the Union and the Brazilian Union Public Defender. He has received research grants from the Brazilian research and innovation agencies CNPq, CAPES, FINEP, RNP, and FAPDF. He has developed research in cyber, information and network security, distributed data services and machine learning for intrusion and fraud detection, as well as signal processing, energy harvesting and security at the physical layer.
ORCID: 0000-0003-1101-3029

1. INTRODUCTION

Banking institutions have pioneered the adoption of computing in their financial services and processes. This scenario was driven by the globalization of the economy and the need for greater security and control of banking transactions. The market needed online financial services with geographically distributed economic movement so that users could have access to banking services regardless of their location.

In Brazil, according to Fonseca, Meirelles and Diniz (2010, p.8), one of the first banks to invest in computerization, in 1979, was Banco Federal Itaú, which created Itautec. Banco do Brasil had computers in its facilities since 1966, but Itaú was one of the pioneers in developing real-time transactions. Soon after came the ATMs with automated and online financial transaction services. Furthermore, banks and financial institutions were big investors in geographically distributed networks that would soon become part of today's internet.

All this initial digital transformation required a large investment in the security and reliability of banking systems. This scenario generated a need to apply standards for the development of robust software and the use of extremely mature programming languages, as any failure in the application's business architecture or software bug could bring great financial losses to banking institutions and customers.

The new digital transformation of the banking system is driven by the recent business model of financial startups and versatile digital banks that were already born applying modern agile processes and software development, delivery and operation methodologies. This generated a great need for traditional banks to also adapt to the new digital age, also leading to several of their products, already adapted to old technological paradigms, to have their processes changed.

According to FEBRABAN (2021), national banks increased their investments in technology by 48% in the year 2019 and another 7% in the year 2020 (Figure 1), having an accumulated growth of 55%. It should be noted that the growth in expenses and investments has been mostly concentrated in hardware and software (Figure 2). This proves the growing efforts by these institutions to adapt to the market. The Bank where this study was applied invests so that all of its products follow the technological market, attracting startups in its innovation hub.

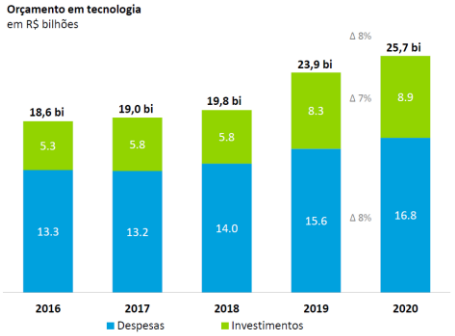

Figure 1. Budget in banking technology
Source: FEBRABAN (2021, p. 6)

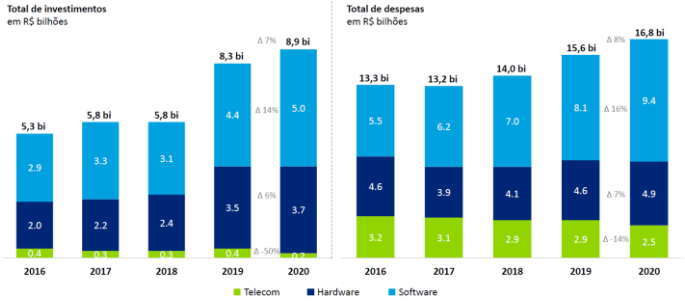

Figure 2. Investments and bank expenses
Source: FEBRABAN (2021, p. 8)

Among its various banking services and products there is the Urban Microcredit Project. This is one of the largest regional development microcredit programs in the country. The project occupies an extremely competitive market that is increasingly being shared with new fintechs that already offer new digital modalities and facilities. The counterpoint is that the Project still suffers from the deficiencies of the institution's traditional processes, and tries to meet the rapid changes in the market without a modern agile process and without a less bureaucratic and continuous software delivery system. How to change this scenario?

The article presents a case study of the implementation and application in the Microcredit project of a continuous integration, delivery and implementation model based on DEVOPS and guided by the main agile methodologies adopted in the market. Automated continuous delivery is one of the main pillars in ensuring the reduction of bureaucracy in the processes for implementing changes and improvements required by the business. Facilitating several daily deliveries in production by automating tests,

breaking the barrier of formal processes and division of the operation and development area is a priority.

Necessary modifications will happen so that these processes can be adapted to the institution's legacy as it has several systems and tools that are already in great disuse in the market and that suffer from problems in adapting to new software standards. Not least, a lot of software is partially modernized with parts that are still legacy and difficult to change.

This article consists of six sections: section 2 presents the theoretical foundation with the main concepts of Agile Methodologies and DEVOPS. Section 3 presents the related works. Session 4 describes the case study and how it was applied. Section 5 describes the proposed process. Section 6 presents the evaluation of the process and, finally, section 7 presents the final considerations.

2. THEORETICAL FOUNDATIONS

Various authors and researchers in the areas of management and information technology collaborated so that agile methodologies were easily adapted and absorbed by the market. With this phenomenon, there was a great need for other consecutive technology and management processes to adapt to the flow of deliveries that the agile model had. Thus, less bureaucratic processes of development and implementation were developed in order to overcome the difficulties of communication and relationship between the areas of development and operations, giving rise to DEVOPS.

2.1 Agile Methodologies

According to Fowler M. and Highsmith J. (2001), Agile Methods must have values and principles as agreed on in the Agile Manifesto. These values include: valuing interaction between individuals over processes and tools, working software over abundant documentation, direct customer collaboration, cutting down on contract bureaucracy.

Not least, the list of principles defined in the Agile Manifesto are the basic concepts for the formation of new methodologies or agile processes and are enumerated in 12 principles according to the Manifesto.

Table 1. Twelve principles of the agile manifesto

1	The highest priority is to satisfy the customer, through early and continuous delivery of value.
2	Accept requirements changes, even at the end of development. Agile processes adapt to changes, so that the customer can take advantage of competitive advantages.
3	Deliver running software frequently, on the scale of weeks to months, preferably in shorter time periods.
4	Business-related people and developers must work together on a daily basis throughout the course of the project.

5	Build projects around motivated individuals. Giving them the environment and support they need, and trusting them to do their job.
6	The most efficient and effective method of conveying information to and within a development team is through a face-to-face conversation.
7	*Functional* software is the primary measure of progress.
8	Agile processes promote a sustainable environment. Sponsors, developers and users must be able to maintain constant steps indefinitely.
9	Continuous attention to technical excellence and good design increases agility.
10	Simplicity: the art of maximizing the amount of work that didn't need to be done.
11	The best architectures, requirements and designs emerge from self-organizing teams.
12	At regular intervals, the team reflects on how to become more effective, then adjusts and optimizes their behavior accordingly.

Source: (FOWLER; HIGHSMITH, 2001)

The main objective of agile teams is to provide the greatest possible value of a software product to the client (MARTIN; MARTIN, 2009, p. 37). An agile team works hard to keep the software structure flexible so that when requirements change, the impact on the system is minimal (HIGHSMITH, 2001).

Although several methodologies emerged prior to the Agile Manifesto in 2001, they have adapted to follow the Manifesto's core principles. These include among others Extreme Programming (XP), Scrum, and Kanban.

Agile can be added to several other processes and encourages new practices and tools. For Variani (2017) “Agile paved the way for DEVOPS, diverting the direction of waterfall model software development methodologies and moving towards a continuous development cycle”.

2.2 DEVOPS

Traditional banking institutions still use the old model of segregating software development and operations teams. This separation ends up generating conflicts between the objectives of the teams that should work with the same purpose. The development team seeks quick deliveries to provide new features and bring value to the customer, while the operations team seeks stability in the production environment, avoiding unavailability of services, meeting demand and maintaining correct capacity. The DEVOPS methodology encourages the coexistence of development and operations teams in a single environment. In addition, it does not segregate professionals from these areas, giving all the responsibility for developing the operation to everyone.

One of the issues with DEVOPS is that there is no strict definition of the process. According to Sato (2015, p. 7), DEVOPS is a cultural and professional movement that encourages the end of segregation between development and operations teams with a focus on automation, collaboration, sharing of tools and knowledge. We can also consider that "It is a phenomenon in expansion, but whose adoption at the level of organizations is

still at an embryonic stage, requiring further research in order to clarify the benefits, costs and barriers to adoption." (SOUSA, 2019, p. 7).

The process is fully adaptable to the environment in which it will be implemented and can be used with different Agile Methodologies, as it is agnostic. It should be taken into account that DEVOPS is a methodology that has characteristics derived from the Agile Manifesto. This is one of the reasons why it can be easily executed with the main agile methodologies in the market. Encouraging rapid response to change and team collaboration by giving everyone ownership of the code and all responsibilities shared equally make teams self-managing. This feature is present in both DEVOPS and SCRUM, EXTREME PROGRAMMING (XP), KANBAM.

DEVOPS encourages several production deliveries per day. Which speeds up product changes to remain competitive in the market. These small deliverables are easier to test and lower the risk of introducing bugs into production. XP shares the same feature of continuous integration and deliveries in small ready-made slices that are more easily manageable and testable (PRESSMAN, 2006, p. 61).

DEVOPS, as it does not strictly follow a standard, has variations in the representation of the software lifecycle. This cycle, according to Joby (2019), is commonly divided into 5 phases: Development, Continuous Testing, Continuous Integration, Continuous Implementation and Monitoring. Each of these phases uses specific tools, methods and processes, but they can still adapt to the tools used in the environment in which it is deployed. The focus of the entire process is on continuous delivery, as it is through this that the customer can see the value of the entire process. The other phases work so that the delivery runs as little risk as possible and is designed with success and agility. The more successful deliveries happen, the more the customer is satisfied and the higher the final product quality.

One of the pillars of DEVOPS is to have knowledge disseminated, avoiding islands of specialization. This feature is one of the most common found in several agile methodologies on the market. Self-managing teams share responsibility for the software product equally among all members (JOBY, 2019, p. 33).

3. RELATED WORKS

Among the works that address the challenge of implementing agile methodologies in financial institutions with legacy processes and systems, one that stands out is that of Fernandes and Junior (2020), which addresses the critical success factors of technology projects managed through agile models. In addition, the author highlights the transition from traditional project management approaches – focusing on the PMBOK – to agile using the

merging of tools and processes contained in SCRUM and KANBAN and focusing on projects related to digital channels that directly impact on the client. The objective was to have faster returns and with demands that needed quick planning and response. The author focused on the importance and functionality of the main roles existing in agile teams and the Critical Success Factors (FCS) of technology projects in a large financial institution, as agile processes are more focused on functional and fast delivery of a product responding to changes in the rapid adaptation of the business to the environment.

According to Ahmad, Soomro and Brohi (2014), agile project management provides an approach that not only offers agility, but also has project concepts with deliverables and applied management. In addition, the authors use several applied concepts, such as XP, SCRUM and Unified Process, to generate discoveries through comparisons that will give future directions to be explored.

The dynamics of agile makes the work of Maia and Brito (2019) also a highlight to be taken into account in this study. According to the authors, the agile methodologies governed by the agile manifesto – which were initially conceived for software development and management – show their great capacity to insert themselves in challenging environments and with rigid processes, as in the case of civil construction. The authors propose the automation of various processes in the field in order to provide more agile services aligned to the global market. Civil construction is an interesting point to be compared with old financial institutions that have several legacy and rigid processes because of the risks involved in changes and innovations and the strict safety and efficiency regulation imposed by control entities.

Another very important work in the area was that of Rutenberg (2017), which addresses the application of agile methodologies in a software development approach at scale within a large financial institution. The author proposed improvements in the software development process by applying the best practices of agile methodologies for prioritizing backlog items according to market analysis data and customer needs. He also defined a list of issues and improvements based on agile practices to drive software development as responsive as possible to rapid market changes. With these interventions, the author hopes that the development processes will be oriented to business value based on more accurate estimates, giving autonomy for the development team to know the entire, end-to-end process. Also aiming to protect against staff turnover, the author estimates a reduction in the integration time of new team members so that the capacity to meet demands is minimally impacted by their changes. Thus, Rutenberg promotes a result that impacts business with responsiveness to change, reduced risk and more added value by delivering the right product at the right time.

Mindful that financial institutions have many legacy systems, one of the related works, which is no less important, is that developed by Schaller (2016), which involves, in addition to the issue of large-scale legacy systems, the challenges of applying DEVOPS against the backdrop of the organizational culture that is consequently guided by security assumptions and explores the issue of large-scale automation of this type of system. The author points out the typical challenges faced by this approach found in the literature when thinking about a scenario of legacy systems and infrastructure. Another very important point which the author emphasizes is the issue of anti-patterns commonly implemented in old legacy systems, as it characterizes a great challenge to modern practices and agile standards which DEVOPS has under its umbrella. The author also reviews similar case studies applied to large and old institutions that either contributed to computational expansion, such as Hewlett Packard (HP), or were one of the pioneers in adopting design standards, process automation, large-scale development, aiming at humanity's most challenging goals, such as The National Aeronautics and Space Administration (NASA).

The work by Feitosa and Ferreira (2020) also explored the difficulties of effectively conceiving agile processes within a large financial corporation that was already exploring this concept, but which was struggling with the long duration and excessive complexity of projects. The authors also faced necessary changes in the scope of deliveries in a way that was difficult to control because of the action of regulatory agencies that required changes or adjustments without much time for preparation. To the detriment of the organization's adversities, their work presents possible mitigating actions for the implementation of an agile methodology in scale and prepared to deal with inclusions of unplanned work in a closed development scope.

Another work that highlights the complementation of agile methodologies with the union of development and operations is the one by Podgornik and Gibertoni (2020), which addresses the experience of observing the need to have an agile infrastructure team , in addition to having an agile development team. The authors noted that this scenario enabled the combination of agile processes in a way that resolved any segregation between teams, transforming them into a single agile team, dividing all responsibilities equally and giving more freedom so that faster changes were better accepted and did not cause a chaotic environment in application development. The work obtained results of deliveries with higher quality and less susceptible to errors or serious problems. Despite not being in a legacy or financial environment, the study addresses the union of agile processes with DEVOPS bringing a complete agile cycle.

By condensing agile methodologies and DEVOPS on a legacy architecture that underwent constant evolution to adapt to digital

transformations, the authors were able to observe the great difficulties involved in adapting new processes and automation tools to modernized systems, but still with vast characteristics that needed major adaptations for this type of approach to be viable. The organizational culture of corporations with this type of systems also makes it difficult to implement agile methods and modern tools for managing processes that are less bureaucratic and more focused on delivering value quickly and responsively. As such, this entire digital transformation tends to be a major challenge in the legacy and financial ecosystem.

This work differs from the aforementioned ones because it applies a case study on formatting an agile management process added to automation tools contained within the DEVOPS framework, customized for systems with legacy parts and in a financial environment of a product specialized in high-circulation urban microcredit. This product is offered by a bank that needs extremely fast deliveries to maintain its competitive edge and keep up with the digital transformation. Modifications to agile and DEVOPS were necessary so that the flows detailed here were feasible for the adapted activities.

4. Case Study: DEVOPS and Agile Methodologies in an Urban Microcredit Project.

The case study is just one of many ways to do social science research, experiments, surveys, historical research, and archival information analysis (ROBERT, 2003, p. 19). This type of study comes from a tradition of medical and psychological research, in which it refers to a detailed analysis of an individual case that explains the dynamics and pathology of a given disease. This method assumes that knowledge of the phenomenon studied can be acquired from the intense exploration of a single case (GOLDENBERG, 1997, p. 33).

This modality of study was carried out within the Urban Microcredit Project of a large bank for 6 months, starting in October 2021, and continues to be implemented and improved until the time of writing of this article. The project identified frequent urgent demands to sustain the product's regional leadership, but the information technology area was finding it more difficult to meet the requested changes every day. This was because, in addition to having many legacy systems, agile practices and continuous delivery were not an organizational culture. All these factors served as inputs for the empirical experience of this research. With each methodology or process implemented, the results were consequently observed with generation of value and mostly positive feedback.

Data collection and feedback served as inputs to improve the application of the methodologies and processes detailed here. Suggestions for improvement originated from the entire team involved in the activities impacted by the practices of this research. The implementation was carried out iteratively and incrementally in such a way that the employees were gradually adapting and without offering much resistance to innovation. Many of the practices had to be based on the knowledge base of the bank's Corporate University in order to overcome some bureaucratic stages of change. This also ended up giving strength to those most resistant to accept the changes, as these, as suggested by this research, had similarities with several processes studied and practiced in the professional development materials offered in courses that are prerequisites for a career at the institution.

4.1. The application of agile methodologies

The banking institution has a vast library of software process knowledge available to all employees, but still applies the use of the RUP in most processes as a standard. Despite being customizable to the bank's needs and having some agile features, the Unified Process already suffers from problems with some stages of the process that have massive documentation and large bureaucracies to be overcome. Many roles and many iterations through the RUP phases are required. Faced with this, the microcredit project entered the SCRUM implementation process to improve responses to changes. An exclusive area for the project was created with dedicated IT and the customer always available. With the approach and reduction of bureaucracy in communication, the customer was able to see the return of value faster.

The development team, with the customer always close by, began to self-organize to ensure deliveries. Islands of knowledge began to disappear as all members of the IT team worked together to ensure delivery and ended up alternating between the different phases of the software process.

With the presence of the Covid-19 pandemic, the technology market is experiencing excessive growth in the region, and several outsourced jobs have gained large turnover of personnel. Due to bureaucratic and contractual issues of a large financial corporation, the Bank and the project did not have much flexibility to retain professionals with competitive job offers. This fact led the bank's staff to implement some XP techniques along with the SCRUM process, making the latter more focused on project management, and giving the other a mission to facilitate the exchange of knowledge. New contractors began programming in pairs with employees and contractors already experienced in the project's software products. This technique, which belongs to XP, accelerated the training of newcomers, faster integration of the team and showed that an agile methodology is efficient to adapt even to scenarios resulting from a global pandemic.

SCRUM's daily meetings, in addition to helping the team to manage itself by exchanging information about what was done, what is being done and blockers, helped with team integration, particularly with newcomers to the project. XP's practice of refactoring, along with pair programming, has helped newcomers get to know parts of products already being modernized and transformed.

Along with the SCRUM process, a panel was inserted in the production room so that KANBAN could also be integrated into the other processes. This helped the team to better visualize the agreed delivery priorities and record the progress of each task available on the board. The Product Owner was able to visualize the evolution of each delivery, improving their expectations and reducing the anxiety of waiting, as they have constant feedback on what was being done. This greatly reduced the customer coming to the IT team for news of the progress of the agreed tasks.

As the Microcredit Project uses the constitutional regional development funds to make part of the loans, at the end of each month, an accounting close is carried out for all operations that used money from this fund and which are outstanding. There is a need for accountability to the Federal Government, the Central Bank and the Ministry of Economy on how this money was used and the status of each loan that used this money. This work mobilizes the business area to work within IT to audit all data related to operations that used resources from the constitutional fund. Thus, in many cases, programmers worked practically all day in pairs with the business professional to validate all the data.

Upon observing this scenario, it was suggested to insert the business team in a special SCRUM sprint at the end of each month so that everyone would be responsible for the deliveries. The business area was also included in the five SCRUM ceremonies bringing the business product backlog into IT during that period. Sprint Planning, daily meetings, review and retrospective meetings also became part of the business team's routine during this period.

This adaptation led to a streamlined delivery of FNE loan reports, as the teams, when unifying at that time, began to exchange knowledge where the professionals' work would converge. Much of the process was improved and the IT team was able to plan better each time this sprint happened.

The result of all this implementation of Agile Methodologies brought observable value to the customer, as expectations began to be met more quickly. Several smaller turnkey deliveries started to happen more frequently and the project profit was noticeably improved. Customer satisfaction was quickly observed and the team's synergy in the deliveries led to the advertising team's greater confidence in investing more in the dissemination of new features.

Although deliveries improved, the team's workload still remained very high because of some bottlenecks that needed to be resolved. As the bank still maintained the old development structure segregated from the operation, the implementations in production still suffered from excessive bureaucracy at each delivery, which led programmers to go beyond all the documental part to achieve the implementation of a new functionality together with infrastructure professionals. Many stages of approval for new deployments needed to be overcome and many interventions by managers were necessary to overcome the suspicion that more frequent deliveries would not destabilize production. How could this be overcome?

4.2. The application of DEVOPS in Urban Microcredit

Upon identifying that the systems implementation process was no longer coherent with other agile processes, the delivery of systems in production began to undergo improvements. The large bureaucracy in the operations area required an excessive effort from the development team to overcome this obstacle. Some automation tools were already present in the project, but it still didn't integrate correctly and they always needed human intervention to execute all the steps. Thus, DEVOPS was suggested as a way to improve the timing of deliveries so that Urban Microcredit can maintain its leadership in such a competitive market.

Digital transformation lent several new technologies to the microcredit scheme that needed to be included in the product. The introduction of the PIX payment system was one of the most anticipated features for the project. Despite already operating at the Bank, the project had its own account for loan operations and all functionalities are managed by systems independent of the Bank. But the market has not stopped and has already adapted to the scenario brought about by the COVID-19 pandemic, as prospecting for the sale of microcredit loans would be limited in places where the on-site visit of loan agents would be impeded by the lockdown. Agents validated the data and provided proof that the borrower was really the person the documentation showed.

As a result, alternatives for facial biometrics, video analysis and other technologies emerged that the microcredit project needed to implement so that loan sales were minimally impacted. DEVOPS came as an answer to streamline the delivery of these new features.

4.3. Continuous integration

As the project already used automation tools, but did not have an integration between them, many of the DEVOPS processes were only applied to what already existed. The Bank has JENKINS as build automation for applications, but it was not programmed for a continuous integration process.

All phases were initiated by human intervention. JENKINS is a tool capable of executing almost all phases of DEVOPS. It is open source, compatible with most languages on the market, has a very active community on the internet committed to plugin development and is very customizable. It is usually deployed on a server that is available to run the integration as soon as a new feature or fix is ready and has been integrated into the final application.

The institution also uses IBM Rational Team Concert (RTC) as a version control tool. The branch structure already follows the market standards for integration of changes in different stages. With that, it was already possible to create triggers for merge events in key branches to start processes in JENKINS.

Faced with this first scenario, a new application was chosen at the beginning of its development in the C# language to carry out the first steps of continuous integration. A project was created in JENKINS and the plugin for monitoring changes in RTC branches was installed. The institution has a centralized repository of approved libraries for use in its applications under development and in the generation of builds. This has been configured to be accessed from the entire intranet. It was configured in JENKINS to perform the application's build by getting all the dependencies configured in the application's configuration file at the centralized repository address through the NUGET dependency manager. In this way, the use of the correct dependencies will be guaranteed when building the application.

NUGET is a tool that can be included in .NET and .NET CORE projects with the function of sharing and consuming reusable and useful code. This sharing takes place through code packages compiled from DLLs and binaries hosted in repositories spread across the internet or intranet of organizations. Both versioning and indexing are controlled and customized by the tool so that other projects can be customized to reuse shared code. Thus, it is an essential tool so that continuous integration can be automated, because in the automated build the development and operation team will not need to worry about managing the supply of cataloged libraries used by each specific construction of a software product.

With the dependency management tool configured in JENKINS, the next step was to configure the unit tests to be executed during the application's build. Unit tests are those written by the application developer himself/herself. According to Sommervile I. (2011), unit testing is the process of testing program components, such as methods or object classes. Individual functions or methods are the simplest type of component. Your tests must be called for these routines with different input parameters.

Using the XUNIT unit testing tool, developers create unit white box tests to validate every grain of functionality helping to validate their work. By having this feature, unit tests can be easily automated by build automation

tools like JENKINS. Thus, with the XUNIT plugin for JENKINS, it was possible to add the automatic process of executing all unit tests of that application before its build.

If any unit test fails or the application build is not successful, the process can end up stopping if a warning system is not created for the developers. It is possible, after integrating all the merged workspaces into a stream, that compatibility issues of changed dependencies occur in the development process. Sometimes a snippet of code depends on a specific version of a dependency that has been changed by another developer to address another issue with another functionality. This scenario can cause build errors that make continuous integration unfeasible and can be configured in JENKINS so that an email is sent to the developer that caused the problem and/or to the development team so that the impediment can be corrected.

Automatic deployment into a development environment is the next step in this process. This step relies on a centralized application server for JENKINS to perform this deployment. In the organization, Microsoft's Internet Information Services (IIS) is the main web server used and the deployment automation project by JENKINS of a new application in IIS was already configured and ready to use, including the correct plugins. But this process was initiated manually by a developer after merging the changes and build – also manual – of the application. What we did was just add a deployment process call at the end of the build process.

Another point to be noted was the update of the respective application's database when there were script changes. This process was also performed completely manually by the developer in the development environment. In approval and production environments, this update was still performed by a database administrator. In view of this, the .Net Evolve framework was adopted. This tool automates the deployment of script updates in the desired database and has a plugin so that JENKINS can also automate this. Thus, the application implementation process in the development environment was changed to include database updates, if any.

About the RTC version control tool, it was necessary to create a development stream to be the trigger point for the beginning of the continuous integration process. Before, work streams were created with the number of work orders generated for the software factory and in them the entire life cycle of software development and testing was carried out before approval. In this mold, the operations team created a job in JENKINS for these streams each time they were created, and after their life cycle the jobs were deleted. With the development stream, a merge point was created to bring all other development branches together, and with that a fixed job was created and configured with the entire process of the initial continuous integration.

Continuous integration was set up so when any new modifications were merged into the development stream, build steps, automated unit testing, code validation and quality testing, and deployment to the development server were performed. This process gives you the possibility to always test that the development code – which is always used for new branches of work – is healthy and compilable so that developers don't start working on a project with code already addicted to errors from other modifications . It is possible to observe that the software has not yet been integrated with other systems, but it is already possible to prevent several problems originating from the initial stages of the software development cycle and prevent programmers from wasting time adjusting flaws caused by other members of the development team.

4.4. Automated integration testing and continuous delivery

Concerned with the quality control of the source code in development, specialists already used SONARQUBE for this type of validation. This tool performs code quality control by analyzing source code snippets that may contain bugs, duplicate command lines or instruction snippets repetitions, security issues, and infinite loop issues. With that in hand, SONARQUBE was added as the first frontier of automated process testing. One of the advantages of the existing infrastructure is that SONARQUBE already has great maturity in the use and coverage of the main applications of the microcredit project. Therefore, it was only necessary to install some plugins in JENKINS so that the SONAR test projects would already work in an automated way. A few more small validations were added because the institution has naming standards in SQL scripts for table, field, function and trigger names and Sonar needs to consult a base of standards and abbreviations to validate the changes. Thus, the code quality process, already previously matured in the environment, could be assured that it would always be executed and any new modifications to the systems.

With the increase in non-standard code alerts, managers became more aware of standardization charges in the development activity. This happened because programmers initially used this device to validate their codes automatically without worrying too much about what they were developing. Managers then asked that the most used development standards on the market be applied and rigorously encouraged code reuse, as some analysts did not read the code to reuse existing functions and did not care about the code's performance.

Once all previous steps of the continuous integration process have been successfully completed and code quality validation, automated integration testing is the boundary that divides continuous integration from continuous delivery. Because they have features that test the system in a

simulated environment, they can extract functional scenarios that only occur when the software exchanges messages with other systems or simulators. For that, mock and stub tools will also be added to this step.

As the system in question has a web page, which interfaces with the user and exchanges messages with the back-end service, SELENIUM was used to create the tests of the web application's functionality. It also performs tests to validate the application's operation in different browsers, making it possible to track down compatibility and portability issues. In JENKINS, a plugin was installed to simulate a virtual monitor so that, in the integration tests, SELENIUM could perform the WEB page validation actions. Without this plugin, it was not possible to access the website's graphical interface and simulate the user's actions on it, as JENKINS is on a Linux server that does not have a monitor and is only accessible by BASH commands.

After all the steps of successful integration tests, the delivery of the application in the approval environment is done by a JENKINS job that takes the binaries and deploys them on a server ready to make the application available already automated in the matter of orchestrating the services to which application integrates and with all the automated management of load balancing and distribution. After this process, an email is sent to all interested parties so that the approval can be started. After all these processes, the system under approval goes to a new phase.

4.5. Continuous deployment

With the end of the approval period, an email is sent to the DEVOPS team signaling the implementation in production in the next change window. If there were no errors, deployment would take place without any need for human intervention. At the institution, for reasons of session integrity with customers, change windows are created to avoid errors or unavailability of the credit service. The continuous deployment had to be adapted accordingly so that it could be approved by the IT management, as many of the services need session integrity and cannot be allocated to another server at the time of the financial transaction. This could lead to disagreements and serious errors. Thus, all stages of the DEVOPS cycle were met and adapted to the business model of the Microcredit project.

4.6. Characteristics observed in the case study

The Urban Microcredit project was an excellent instance for applying the intended case study because it needed a less bureaucratic process to keep up with market demand. In view of this, together with a theoretical review, it was possible to note relevant events observed in the characteristics of the environment in which the set of processes was inserted.

Table 2. Relevant facts that occurred in the implementation of the processes

1.	Agile methodologies are a new culture that has to be adapted to organizational needs without letting agile principles be modified.
2.	The implementation of agile methodologies brought the need for a gradual adoption so that the team could adapt to the new paradigms and that the principles were not left aside little by little because they seemed to be difficult to be executed.
3.	Agile also brought the need to use the DEVOPS culture to mitigate the segregation between the development team and the operations team, avoiding resistance in the implementation of new versions of the systems.
4.	DEVOPS brings the practice of automation as a culture and requires good training from the team on the tools and plugins used so that the process can be orchestrated.
5.	Management was advised to closely monitor the gradual implementation of the processes so as not to offer resistance for fear of the low quality of software products and fear that automation could destabilize the availability of the production environment.
6.	The development team, already used to making adaptations to legacy systems, did not show any lack of motivation to solve automation difficulties due to lack of compatibility. In the daily SCRUM meetings the resolutions for impediments presented were mostly known by the other experienced team members.
7.	The most resistant managers, after seeing the value generated by the implemented process, were the biggest encouragers for other teams to adopt the same processes.
8.	Many automation tools and plugins for JENKINS are easily available on the internet and easily accessible so that the process is easily replicable in other areas with particular needs.
9.	The testing team was the most overburdened early in the deployment, as they had to automate many processes that were previously performed manually. This generated an initial bottleneck in the flow of releases that required the collaboration of developers to be overcome.

5. Applications and Benefits of Agile Methodologies and DEVOPS in Urban Microcredit Project.

The case study showed that, for the application of methodologies and DEVOPS in an urban microcredit project within a banking institution, a process had to be designed. This process was developed with the intention that other areas and projects within the institution concerned are supported by it. Boundaries are not limited to the institution in question, and the model can be applied to institutions with similar difficulties and challenges. Thus, it is known that banking and finance institutions are commonly structured on modernized legacy systems and rigid processes. However, the technology area has many similar challenges in different business areas that started in the technology world in the same decade or in the next few years.

Financial organizations, despite having similar goals, usually differ in the way they manage information technology. Many were pioneers in this area and are also excellent technology companies. But in order to respond to the ever-faster changes in the market and not lose competitiveness, an agile, customizable and standardized approach can be adopted. Thus, the process model was customized to meet the needs of the changes required by the

project without conflicting with the institution's rules regarding its regulatory limits.

After several meetings with management, adaptation to the institution's standards and validation of the process by the IT audit, the final flow proposed in Figure 2 could be divided into ten phases, with 8 activities and 2 sub-processes:

Sprint Planning – sub-process that brings together the SCRUM ceremonies for sprint planning and the filling of the KANBAN board placed in the production room with the items from the sprint backlog.

Start of Sprint – activity in which the division of tasks is performed for the development team to kick-start activities.

Functionality development in pairs – activity that brings together two team members for the development of functionality using XP's practice in pairs and guided by tests. Tests are developed first.

Test development and automation – activity responsible for developing and automating the tests that will be performed in the phase between integration and continuous delivery.

Merge with the development branch – activity that performs the integration of versioned code on the developer's machine into the development branch of RTC.

Unit tests compilation and execution process – automated task triggered right after the merge of the code in the development branch. This task is started by a trigger at the end of the code merge and has a test success condition so that the main flow of the process continues. In case of non-approved tests, the flow goes back to the development phase so that the necessary corrections are carried out.

Automated Integration Testing – an automated task that orchestrates the delivery of application binary files in an environment that simulates systems integration using mocks and stubs. Furthermore, the services simulate user actions in software with a graphical interface through SELENIUM. This activity depends on the success of the tests so that you can proceed with the main flow of the process. If the tests fail, the functionality goes back to development for corrections to be made.

Automated delivery to the approval environment – automated activity that delivers binaries to the approval environment so that the customer can approve the agreed functionality. This activity is initiated by a trigger and depends on the client's approval to proceed with the main flow.

Deployment in production - automated activity that depends on the customer's approval to be carried out. It is the last step in the DEVOPS process and ensures that one or more functionality has been delivered after all Continuous integration/Continuous delivery (CI/CD) stream approvals.

Sprint Closing Ceremony – sub-process that contains the activities that end the sprint cycle after the agreed period and that makes prompt deliveries of a functional, ready-to-use product. As the DEVOPS flow already performs small ready-made deliveries, the closing ceremony brings together everything that was developed into a product.

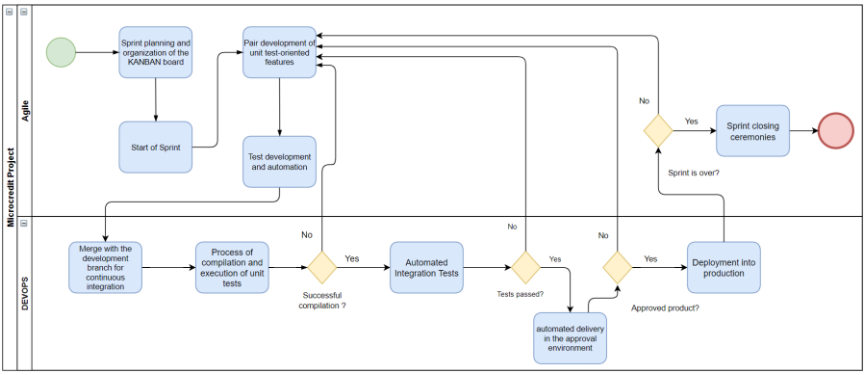

Figure 3. Agile process flow with DEVOPS.

5.1. Sprint Planning Subprocess

This sub-process shown in Figure 3 was customized so that the KANBAN flow was added, as one of the requirements of the audit was that the activities that were being carried out were easily visible in a table. Thus, the customer waiting for the features did not need to interrupt the development team with questions about the progress of deliveries. Not least, the board helped the team in self-management, as it was also used to signal blockers that required the intervention of other team members.

The activities of the sprint planning subprocess, its characteristics and objectives are described below:

Prioritization of requirements to be implemented – Activity with the purpose of prioritizing backlog items for possible candidates to be implemented in the next sprint.

Discussions to understand the functionalities – activity that gives the development team the opportunity to clarify doubts with the P.O. and to, together with the team, understand how such activity can be carried out.

Activity effort estimation – Helps the team to make a more accurate estimate based on past experiences of similar activities. It is an activity that relies heavily on empirical data and on the team's sensitivity.

Definition of the sprint backlog – At the end of the definition and estimation round, here will be selected which activities will fit the team's ability to deliver, taking into account the priority and size of each activity.

Not all prioritized activities can be submitted for the next sprint making it impossible for the team to have enough time to complete it.

Filling and organizing the KANBAN board – The board is displayed in a visible place with colored flags indicating each member of the team. The sprint backlog items are arranged in such a way that anyone watching can have feedback on the progress of the development process and the delivery of features.

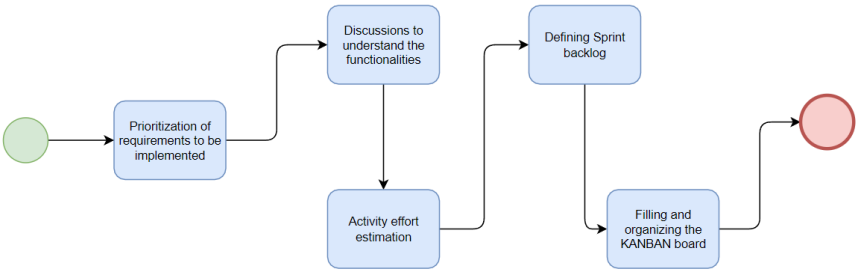

Figure 4. Flow of the sprint planning subprocess.

5.2. Sprint Closing Ceremony Subprocess

With the end of an agile management cycle and deliveries of a sprint, it is necessary to make some analysis of the deliverables, processes and results of that work period. Usually a sprint is always accompanied by learning and new difficulties to be overcome by the team and all these events need to be recapitulated and discussed by the members so that all doubts are clarified and knowledge is shared. Efficient agile teams avoid islands of knowledge within the team so that all members can be able to solve any problem. Thus, for regulatory reasons of the financial institution, the activities were customized so that an audit report was developed to be documented and delivered as proof of the product's compliance with the bank's credit and operating standards.

The activities of the sub-process in Figure 4 of Sprint Closing Ceremonies are described below:

Product Presentation – At the end of the sprint, this activity presents the product to the interested parties, reviewing the agreed deliveries and proving the operation of these new features with the bank's current credit rules. Some requirements are revisited to clarify doubts.

Sprint review – Here the characteristics, difficulties and knowledge about each activity delivered in the sprint are discussed. Observation points are added to each activity as a document for future auditing.

Sprint retrospective – The team needs to share what was done well, what can be improved, what they learned, and what can be done differently to

improve deliverables. This data is also condensed in the audit report so that the bank can improve the process as a whole.

Audit report – The bank's audit has the mission of guaranteeing that the credit systems' functionalities are in full compliance with the credit regulations. This avoids customer dissatisfaction and legal issues. The activity condenses all reports and materials prepared in the previous steps of the process and validates with the delivered requirements.

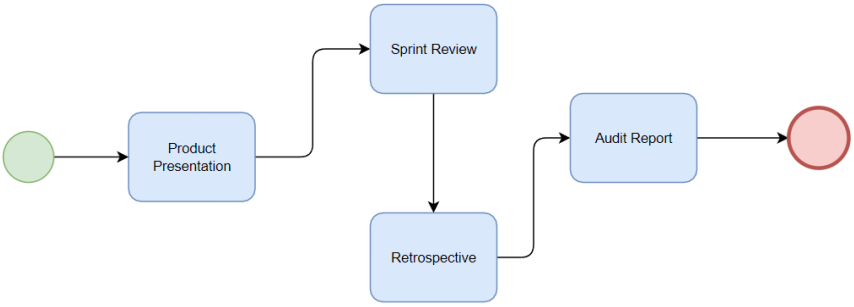

Figure 5. Sprint Closing Ceremony subprocess flow.

6. PROCESS VALIDATION BY ENTERPRISE ARCHITECTURE

A commission for validation and approval of the process was set up by the institution's corporate IT architecture. Comprised of developers and architects with higher education, master's and doctorate degrees, the commission uses the technique of interview questionnaires for the final validation of the process. First, the process was submitted for the IT audit to validate that all process boundaries are in accordance with the institution's principles and rules. After initial approval, the implementation of the process as a pilot in the urban microcredit project was authorized. Thus, fourteen analysts involved in the sector where the project is supported were selected to participate in the training, implementation and validation stages.

Taking advantage of the validation infrastructure, the researcher requested that the review be carried out together and that the questionnaire be modeled in order to better relate the results to the objectives of this work. First, the entire process was presented to fourteen environmental analysts, then the process was adopted and all the infrastructure adaptation was carried out. A small process was carried out to implement DEVOPS in the institution's existing tools while the team was being trained to carry out the adapted agile practices.

The process in Figure 5 consists of orchestrating the various JENKINS jobs that already exist without any orchestration for automatic triggers and treatments. Thus, the process has the following phases:

Development stream creation – The development team must create a stream for merging the tasks performed by the development team.

Job trigger creation – From the development stream, a trigger was created so that, at the end of a merge of a delivery, CI/CD processes are started.

Adaptation and installation of plugins – The corporation already has great proficiency in most of the possibilities of adapting JENKINS so that legacy and modernized systems can be executed at all stages. But some of them were not yet automated and needed human intervention. In this way plugins were selected and developed so that all steps were completely automatic. Thus, tests and builds were possible at all stages of the process for most application modules.

Serialization of CI/CD jobs – The entire job chain is linked so that the entire sequence of integration, delivery and continuous deployment takes place. Alternative paths are created in case of failure in each of the activated triggers. They generate e-mail notifications of the steps being carried out for the entire team, but only the responsible analyst will receive the cause of the error immediately to avoid an uncomfortable environment for the team.

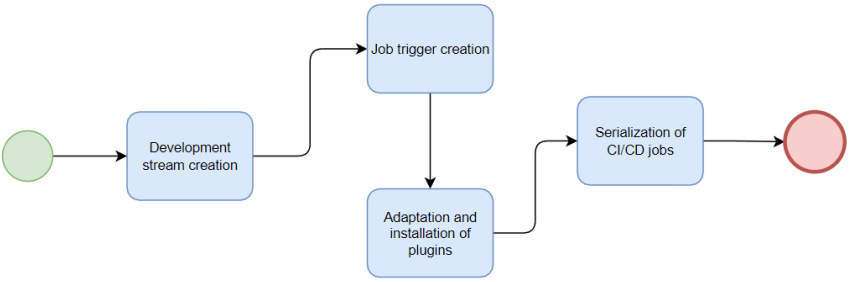

Figure 6. Flow of the DEVOPS implementation process in the urban microcredit environment

Right after the agile training and the execution of the DEVOPS implementation process, the process in figure 2 was established and executed for three months before the interviews and questionnaire response. For reasons of corporate rules, the interviews between the participants were carried out confidentially so that in case of negative feedback, harassment could be avoided. Suggestions were made by some of the interviewees who reported the great work needed to make all the automation promoted by the CI/CD process possible.

The promise of a more automatic future environment that would require less effort from the development team was key for motivating the DEVOPS implementation team. No less important, agile was highlighted as a relevant point due to the ease of adapting the process and quick view of results. A counterpoint was that for this entire process to happen, management must be willing to innovate and reduce bureaucracy in the environment to be applied and believe in the benefits that could come if the process is fully adopted.

The questionnaire in table 3 summarizes the result of the responses applied to the fourteen members of the environment.

Table 3. Results of the process evaluation questionnaire

Question	Disagree	Can't Answer	Agree
The process is consistent with the urban microcredit IT environment.	0%	7%	93%
The process can be well accepted in other areas of the institution.	7%	14%	78%
All areas of the bank are able to carry out the process.	28%	7%	65%
The process is easy to implement.	28%	7%	65%
The process promotes a lighter work environment with fewer repetitive activities.	0%	14%	86%
Documentation is sufficient for learning and executing the process.	7%	7%	86%

7. FINAL CONSIDERATIONS AND FURTHER STUDIES

This study showed a specific case, along with the theoretical review, of the challenges and difficulties of optimizing and designing a software development and operations process included in an environment not only with legacy systems, but also with the organizational structure of a large banking institution and focused on an urban microcredit market niche. The concern with the quality of deliveries was the greatest difficulty to be overcome by the management leaders, as a critical error could generate financial problems for the institution. The doubt over whether the agile management process together with DEVOPS would manage to establish itself in the microcredit project was overcome as the first quality deliveries were implemented in the production environment. The improvement in the response to changes provided by agile practices was soon noticed by all project members, by the client and by the senior management of the environment. Thus, the CI/CD process brought results very quickly, with immediate customer feedback through social networks, mobile app stores and physical service stations.

Before, deliveries were done in batches with a large conglomerate of fixes and new features. Often, new features were no longer needed or no longer had the same impact. This was due to the delay in the team's response to changes and not delivering these features quickly and gradually due to the priority value of each demand. In this work it was possible to develop a mixed process of agile methodologies together with a custom CI/CD process to meet

the legal and institutional requirements of the financial institution. Audit processes were included in the process and requirements were met thanks to the effort of the entire team. A small implementation process on the part of DEVOPS was also designed so that the necessary adaptations in legacy or modernized systems were carried out and that the entire JENKINS infrastructure was better utilized. Thus, this sub-process can be a facilitator for new areas of the bank or other institutions to implement and use the process developed in this study.

As a future study, this work contributes so that new processes with an agile base and continuous implementation are developed with a greater depth in specific business models such as urban microcredit. No less important, a future study with additions of new tools emerging in the market and practices that contemplate the new work models of geographically separated teams would be appropriate. Another point initially explored in this work, and which could also be deepened, is the work in pairs with the business team for complex specifications of critical requirements.

Acknowledgments

This work was developed in a laboratory with computational and staff resources supported in part by CNPq - Brazilian National Research Council, Grant 312180/2019-5 PQ-2, Grant BRICS 2017-591 LargEWiN, and Grant 465741/2014-2 INCT in Cybersecurity, in part by CAPES - Brazilian Higher Education Personnel Improvement Coordination, Grant 23038.007604/2014-69 FORTE and Grant 88887.144009/2017-00 PROBRAL, in part by the Brazilian Ministry of the Economy, Grant 005/2016 DIPLA and Grant 083/2016 ENAP, in part by the Administrative Council for Economic Defense, Grant CADE 08700.000047/2019-14, in part by the General Attorney of the Union, Grant AGU 697.935/2019, in part by the Ministry of Health, Grant DENASUS 04/2021, and in part by the Attorney General of the National Treasury, Grant PGFN 01/2021.

REFERENCES

AHMAD G.; SOOMRO T. R.; BROHI M. N.; Agile Methodologies: Comparative Study and Future Direction, **European Academic Research**, Feb 2014, Vol. I.

FEBRABAN. **Pesquisa FEBRABAN de Tecnologia Bancária 2021**. São Paulo2020. Available in: < https://cmsarquivos.febraban.org.br/Arquivos/documentos/PDF/pesquisa-febraban-relatorio.pdf >. Accessed: March 05, 2022

FEITOSA, L.; FERREIRA, W. Desafios da aplicação do ágil escalado em projetos de software: estudo de caso em uma organização financeira. **Revista de Gestão e Projetos**, 2021, p. 195-221. doi: 10.5585/gep.v12i1.17825. Available in: <https://periodicos.uninove.br/gep/article/view/17825>. Accessed: June 19, 2021

FERNANDES, A.; JUNIOR, R. Um estudo dos fatores críticos de sucesso em projetos de TI Gerenciados Por Abordagens Ágeis Em Uma Instituição Financeira. São Paulo: **Revista Gestão & Tecnologia, Pedro Leopoldo**, Oct./Dec.2020, v. 20, n.4, p. 268-283.

Fonseca, C. E. C.; Meirelles, F. S.; Diniz, E. H. Tecnologia bancária no Brasil: uma história de conquistas, uma visão de futuro. São Paulo: **FGV ERA**. 2010, p. 7-23. Available in: <http://hdl.handle.net/10438/13532>. Accessed: May 02, 2021.

FOWLER, M.; HIGHSMITH J. The Agile Manifesto. 2001. Available in: <https://www.drdobbs.com/open-source/the-agile-manifesto/184414755>. Accessed: 03 May 2021.

GOLDENBERG M. A arte de pesquisar. Rio de Janeiro: **Record**; 1997. p. 33.

JOBY, P. P. Exploring DevOps: Challenges and Benefits. **Journal of Information Technology and Digital World (2019)**, Vol.01/ No. 01, p. 27-37.

MAIA E. M.; BRITO A. V.; Aplicação da Metodologia Scrum na Construção Civil, **European Academic Research**, June 2019, Vol. VII.

MARTIN, R. C.; MARTIN, M. Princípios, Padrões e Práticas Ágeis em C#. Porto Alegre: **Bookman**, 2009, p. 30-39.

NERI, M.; MEDRADO, A. L. Experimentando microcrédito: uma análise de impacto do Crediamigo no acesso a crédito. **Revista Econômica do Nordeste**, Volume 41, Nº 01, 2010, p. 1-6.

PODGORNIK, H.; GIBERTONI, D. Um estudo de caso sobre o desenvolvimento de uma aplicação scrum e devops. **Revista Interface Tecnológica**, [S. l.], v. 17, n. 2, p. 244-255, 2020. DOI: 10.31510/infa.v17i2.978. Available in: <https://revista.fatectq.edu.br/index.php/interfacetecnologica/article/view/978.> Accessed: July 20. 2021.

PRESSMAN, R. S. Engenharia de software; tradução Rosângela Delloso Penteado. 6ª ed. São Paulo: **McGraw-Hill**, 2006.

ROBERT, K. Y. Estudo de Caso: Planejamento e Métodos. Porto Alegre: **Bookman**, 2001. p. 19-27.

RUTENBERG, F. G. M. Desenvolvimento De Software Em Escala Utilizando Práticas Ágeis: Um Estudo De Caso Em Uma Instituição Financeira. Santa Catarina: **Universidade do Sul de Santa Catarina**, 2017 Available in: <https://repositorio.animaeducacao.com.br/handle/ANIMA/8986>. Accessed: June 05. 2021.

SATO, D. DevOps na Prática. Entrega de Software Confiável e Automatizada. São Paulo: **Casa do Código**, 2013, p. 7-20.

SCHALLER, Amy E. DevOps transformation challenges facing large scale legacy systems. Utica College: **ProQuest Dissertations Publishing**, 2016 Available in: <https://www.proquest.com/openview/cbc3c0fc15cd39d0d56b92c5169c95bc/1?pq-origsite=gscholar&cbl=18750>. Accessed: May 18, 2021.

SOMMERVILLE, I. Engenharia de Software; tradução Ivan Bosnic e Kalinka G. de O. Gonçalves. 9. ed. São Paulo: **Pearson Prentice Hall**, 2011. p. 144-162.

SOUSA, L. F. R. DevOps: estudo de caso. Coimbra: **Instituto Superior de Contabilidade e Administração de Coimbra**, 2019. p. 7-20 Available in: <http://hdl.handle.net/10400.26/31932>. Accessed: May 17, 2021.

VARIANI, A. DevOps introdução, como surgiu, conceitos e ferramentas. **Alefevariani,** 2016. Available in: <https://alefevariani.wordpress.com/2016/02/17/devops-introducao-como-surgiu-conceitos-eferramentas/>. Accessed: May 18, 2021.