S-SCRUM—Methodology for Software Securitisation at Agile Development. Application to Smart University

Sergio Claramunt Carriles, José Vicente Berná Martínez, Jose Manuel Sanchez Bernabéu, and Francisco Maciá Pérez

Abstract The use of agile methodologies during software development is a common practice nowadays, mainly because they facilitate the delivery of value to the client and contribute to the viability of the project. However, security is an aspect that can hardly be contemplated when focusing on the development of functionalities. In the agile development team, responsibilities are diluted in the team and the individual competence of the members has to be relied upon. This paper proposes to extend the SCRUM methodology with new processes, artefacts, and roles to generate Security SCRUM (S-SCRUM). This methodology contemplates the guarantee of security in any project that uses it and claims the figure of the security expert as an indispensable figure in the development of large-scale software. As part of the proposal, the methodology has been used in a real project being developed by nine Spanish universities, Smart University, demonstrating its usefulness and contribution to both agility and system security, facilitating the delivery of secure value increments.

Keywords Security SCRUM · Agile · Secure development · Security expert

S. C. Carriles · J. V. B. Martínez (✉) · J. M. S. Bernabéu · F. M. Pérez
University of Alicante, Carretera San Vicente del Raspeig S/N, 03690 San Vicente del Raspeig, Alicante, Spain
e-mail: jvberna@ua.es

S. C. Carriles
e-mail: sergio.claramunt@ua.es

J. M. S. Bernabéu
e-mail: jms.bernabeu@ua.es

F. M. Pérez
e-mail: pmacia@dtic.ua.es

P. P. Joby et al. (eds.), *IoT Based Control Networks and Intelligent Systems*, Lecture Notes in Networks and Systems 789, https://doi.org/10.1007/978-981-99-6586-1_3

1 Introduction

In 2022, the University of Alicante, together with eight other public universities, obtained a grant from the UniDigital Plan [1] for the development of a new Smart University platform that would provide all Spanish universities with a platform, based on open source, capable of capturing, storing, processing, and exploiting the data sources produced by the different digital ecosystems of a campus. This new platform will have to be public, it will be exposed like any other service to malicious eyes and, above all, it will have to be scalable to offer services to a potential user community of hundreds of thousands [2]. Today, agile approaches to software development are widely used, as these approaches allow value to be delivered quickly and consistently.

However, in the context where the platform will have to exist, regardless of its functionalities, security will be one of the biggest challenges [3]. This is why the design and implementation of security mechanisms and systems to guarantee the confidentiality, integrity, and availability of the service must form part of and be integrated into the development of the platform [4]. But in a project developed from scratch through an agile methodology that includes change as part of the value, it is very difficult to design security in advance [5]. One of the artefacts used during agile development is user stories. Through them, we define the functionality expected by the user, but they rarely (if ever) define aspects concerning quality requirements such as security aspects [6].

In this work we propose, within the agile methodology, to include security as part of the iterations, so that, without being part of the specification, but through a specific process added to SCRUM [7], this property is guaranteed in the system. This new artefact has been called "security stories" and the resulting methodology is called Security SCRUM or S-SCRUM.

The rest of the article is organised in the following sections: Sect. 2 presents the adaptation of the development methodology to our work context; Sect. 3 shows how the methodology is applied in the development of the Smart University platform and the results it has generated; Sect. 4 presents the contributions and lessons learned; and Sect. 5 finally presents the conclusions and future work.

2 Security SCRUM

In an agile development environment, it is common to use methodologies such as SCRUM. This organisation allows product delivery to move forward quickly and always ensure the delivery of value to the customer. Figure 1 illustrates the typical SCRUM development cycle.

This organisation prioritises, as we have said, the user story (UH), the functionality, and the increase in value for the owner. Based on the UH defined by the owners, a backlog is created, which in turn is used to generate a sprint backlog by the scrum master [8] and the developers, which will finally be executed in the sprint. However, in

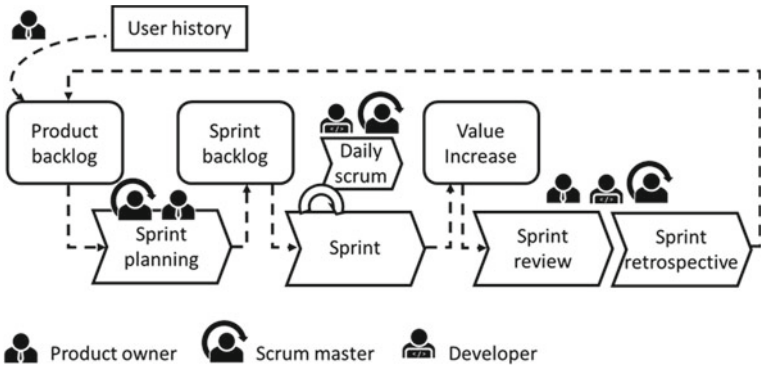

Fig. 1 General SCRUM methodology diagram

this methodology, only the aspects contemplated in the user story are implemented and, therefore, security, not being part of the user stories provided by the owner, may remain unimplemented or at least not receive the main focus of interest during development.

In our proposal, what has been done is the modification of the SCRUM methodology, adding a securitisation process, where the increment provided together with its integration is analysed and refined from a security point of view, and the necessary implementations are added, or it is added as a new user story. The SCRUM methodology is as illustrated in Fig. 2.

This methodology, which we have called Security SCRUM (S-SCRUM), includes a new specific profile in the development that of the security expert. The development is still focused on the functionality, but once it is finalised and integrated, it is proposed that this expert analyses the security requirements related to the new increment, and proposes, through user security stories, the implementation needs on the software.

The aim is to maintain the agile nature of the SCRUM development methodology, but to add a mechanism to ensure that security is taken into account, without interfering with development.

2.1 The Role of the Security Expert

The proposal considers explicitly adding the figure of the security expert to the development chain. The security expert is a specialist in vulnerabilities and their harmful effects on the software, and his or her particular perspective, aligned with functionality and focused on security, is essential to provide the software with the necessary quality [9]. By adding a specific profile, developers are relieved of the task of analysing and implementing security, as is the case with other profiles such as system administration.

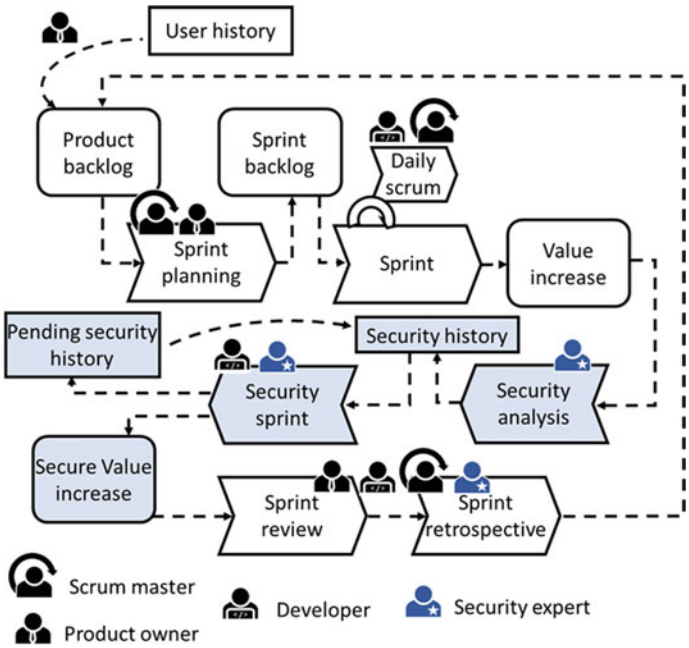

Fig. 2 S-SCRUM methodology diagram

The security expert profile ensures that a security expert will analyse the newly generated increment and detect security gaps and issues and generate security stories for implementation.

In addition, security stories that cannot be implemented during the sprint can be accumulated for the next sprint. This allows security needs to be left unimplemented and noted.

Just as people with knowledge of the frameworks and technologies used are employed during development, in S-SCRUM it is demanded that a dedicated expert, or at least a suitably trained team member, is responsible for the security aspects. In SCRUM the thrust of the product focuses on functionality and on customer value. In S-SCRUM, the aim is that the incremental process should also be safe.

2.2 Security Analysis Process

One of the new processes that has appeared is security analysis. This process is responsible for performing the security analysis on the new incremental value generated and is carried out by the security expert. This process is executed using the Magerit processes [10], dividing the process into several activities as shown in Fig. 3: the creation of the inventory of assets involved in the increment; functional analysis

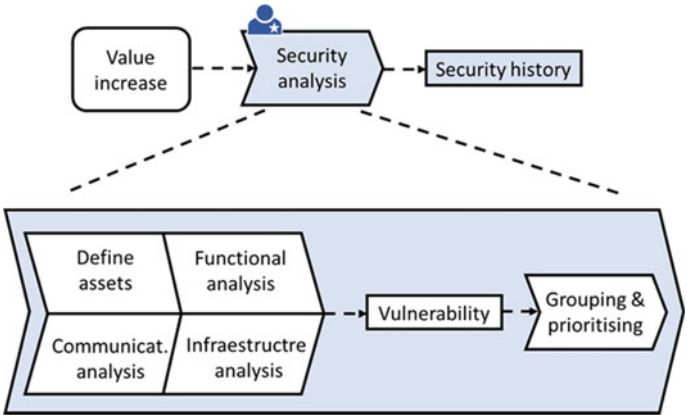

Fig. 3 Internal diagram of the activities of the security analysis process

of the increment; analysis of the communication systems employed by the increment; and analysis of the technologies involved in the increment.

Through these analyses, a list of vulnerabilities are generated, grouped and prioritised. Now, with this ordered information on vulnerabilities, security stories are generated. Each security story reflects the need and intention to address one or more vulnerabilities.

As mentioned above, Magerit is used as the base for the analysis, making use of the assets catalogues, vulnerabilities, and countermeasures. Magerit focuses on the generation of security plans, which can actually be seen as detailed descriptions of security stories. In this case, a security story is generated from the point of view of the security expert, describing only the objective in question, without detailing exactly how it will be implemented. This will be the task of the next process.

The next step is the execution of the security sprint, in which the security stories are materialised. This sprint consists of the implementation of the necessary countermeasures to resolve the vulnerabilities. The security expert, in cooperation with the developers, carries out this action.

Precisely because these user stories can be complex, involve many assets, or even be expected to involve new assets, the security expert can decide to postpone their implementation to the future, leaving them as pending. These security stories will become part of the security stories in the next iteration.

3 S-SCRUM in Smart University

The Smart University [11] project proposes the creation of a system that integrates and centralises all the information coming from the different types of sensorisation devices that the university may have. This information can be visualised, analysed, and processed using AI techniques with the objective of generating information that facilitates decision-making, so that the university is able to manage its resources, infrastructures, and services more efficiently. The platform forms a complex ecosystem of services that should facilitate the use of real-time data, the generation of an Open API for the consumption of historical data or by third-party applications, and a complex system of data representation and exploitation. Figure 4 shows a schematic of the platform architecture.

As can be seen, there are many different technologies and services coexisting on the platform and integrating with each other. All the elements are virtualised using Docker and choreographed through Docker-Compose. Within the platform we can find Nginx proxies, API Rest Node, SQL DB and InfluxDB, Telegraf, Kafka, NiFi, and many other elements.

The development of the platform has been carried out using the S-SCRUM methodology, so that in each iteration the following user stories to be implemented are defined. The implementation has followed an order, from left to right in Fig. 4, of the components. Initially, the user stories were intended to create the basic services to capture and send data to the platform. For this purpose, the data acquisition and its dumping to Kafka, the transmission to InfluxDB, and finally the loading of these data into the FrontOffice in order to be able to offer them to the user are enabled. The following is an example of one of the sprints implemented, as an illustration of the proposed methodology.

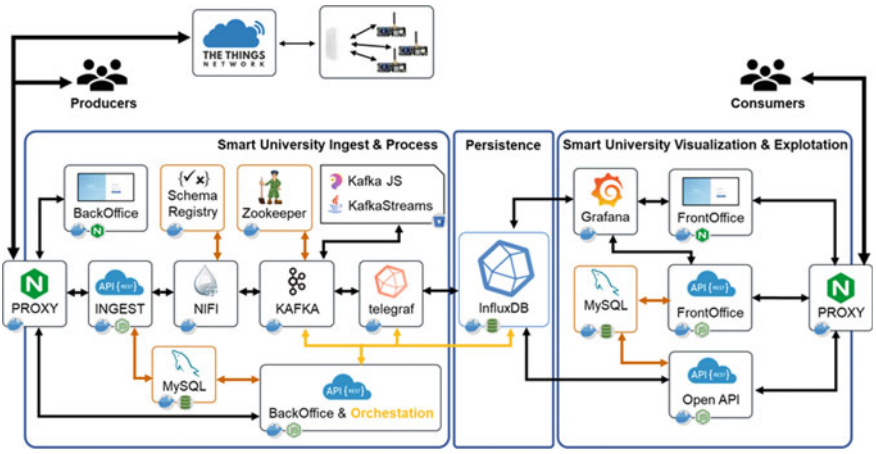

Fig. 4 Internal architecture of the Smart University platform

3.1 Sprint Securitisation—APR—Publish API Rest

One of the project's requirements was the ability to receive data from various sources, through a Rest API offered by the platform, with which customers dump data to the platform to be processed in the Kafka broker and subsequently stored in InfluxDB. In one of the development sprints, it was determined in the user story that the time had come to publish the API Rest for receiving data, in other words, to make it accessible to users and start using it to simulate real-use cases.

At this time, the platform had the elements as shown in Fig. 5, which is divided into two parts, the right part marked as user history, and the left part, marked as security history. On the right side, and as part of the user history of this sprint, a new component, API Rest INGEST (marked as new), would have been added.

When the functionality was completed, it was handed over to the security expert, who analysed the new vulnerabilities generated by this IGNEST component. It was determined that of the most important vulnerabilities found, several were related to secure access to the INGEST resource. This resource was named as ASE1 and added to the asset catalogue. In addition, together with the asset, its detected vulnerabilities were named:

- ASE1v1: Internet exposure of internal services or private use. Italics or bold face are not to be used.
- ASE1v2: Distributed Denial of Service (DDoS) due to excessively large, malformed, or even huge numbers of packets sent to the platform.
- ASE1v3: Lack of centralised monitoring of access to platform components.

This set of vulnerabilities put the availability and confidentiality of the platform services at risk, and therefore countermeasures had to be added to the system. As the vulnerabilities were related to the same assets, they were grouped together for common treatment, and the security expert then defined the security story SH-APR1:

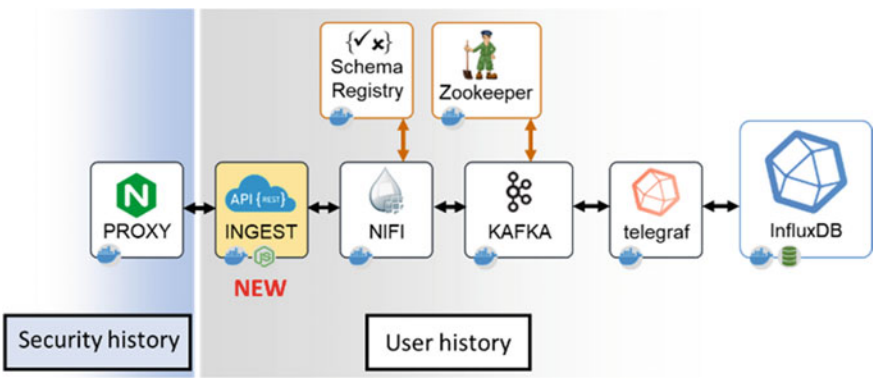

Fig. 5 Security history added to APR user history

Centralise access to the platform through a single point that hides the deployed ports and allows the implementation of traffic control techniques.

This user story was implemented as a Docker container that hosts a reverse proxy Nginx (Fig. 5, left part—security history). This reverse proxy should be configured to resolve the detected vulnerabilities:

- Configure a reverse proxy to receive requests from the outside, for this purpose, a Docker container with Nginx is configured with the appropriate services, exposing a single port to the outside 443, properly secured, and directing traffic to the appropriate inside port, as shown in Fig. 6.
- Configure proxy policies to limit the allowed size of data sent, timeouts, source IP restriction (to limit access to authorised stations only), as shown in Fig. 6.
- Configure the log format of this proxy in order to be able to be processed in a monitoring service. The objective is to take advantage of the fact that all the activity will transit through this component in order to have information on all the requests that have occurred, both correct and incorrect, and also information

```
server {
   listen 443 ssl;
   server_name ingest.domain.com;
   cliente_max_body_size ...;

   ssl_certificate ...
   ssl_certificate_key ...

   location / {
      ...
      proxy_pass http://localhost:.../;
   }
}
...
   server_name ingest.domain.com;
     proxy_read_timeout ...;
     proxy_connection_timetou ...;
     proxy_send_timeout ...;
           ...
   location / {
      allow ...;
      allow ...;
      deny all;
      proxy_pass http://localhost:.../;
   }
}
```

Fig. 6 Example Nginx to control traffic

```
...
log_format custom $time_iso8601 | $remote_addr | $request_method | $status |
$request_length | $http_host | $uri;
...
```

Fig. 7 Example Nginx configuration to centralise requests

```
2023-01-25T19:46:50+00:00|193.145.230.15|POST|200|1116|ingest.domain.es|/
2023-01-25T19:46:51+00:00|193.145.230.15|POST|200|1116|ingest.domain.es|/
2023-01-25T19:46:53+00:00|193.145.230.15|POST|200|1116|ingest.domain.es|/
2023-01-25T19:46:57+00:00|193.145.230.15|POST|200|1116|ingest.domain.es|/
```

Fig. 8 Example Nginx configured to generate log information in the chosen format and example of the output produced by the console

on the origin of the requests. For this purpose, the proxy was configured with a treatable format as shown in Figs. 7 and 8.

3.2 Results of Implementing S-SCRUM at Smart University

The project is a live project, which is still under development, so all security aspects are not yet covered. But by using an agile methodology, focused on providing value to the user and in which the security is also carried out, the project guarantees that the increases in value are both functional and secure.

Following the methodology, 17 assets have been inventoried and 105 vulnerabilities have been identified. The SCRUM methodology increases the delivered value, S-SCRUM allows the creation of the asset inventory, vulnerability analysis, and the implementation of security countermeasures, at the same time as the delivery is generated. This also makes it possible to check the proper functionality and validity of the measures provided.

In the catalogue of measures implemented in the platform, we can find many configurations to secure internal communication between containers, encrypt the information stored, protect access to resources and databases, and monitor the general operation of the system, as shown in the panel in Fig. 9.

With a security expert who knows the system and the security measures implanted, as part of the security stories, it is possible to consider the grouping or enhancement of measures already done. Indeed, as the system evolves, it is possible that, at a given moment, a new element will affect other existing assets. It is then when the specialist determines to change, improve, or enhance the measures. Figure 9 shows the result of grouping several monitoring measures and centralising them in a single dashboard. When only one component was monitored, the implementation of dashboards was excessive, especially if we only want to show one or two indicators. But now that

Fig. 9 Capture of the system monitoring dashboard

we have dozens of components with dozens of indicators, it is more than advisable to generate this type of tool.

It should be considered that at no moment should these monitoring tools be generated as part of the system's functionalities and therefore they would never appear in a user story.

4 Contributions and Lessons Learned

The use of agile methodologies does not mean that not all aspects of development are taken care of. Nowadays, security is an essential dimension in software, as well as performance, efficiency, effectiveness, and even user experience. The SCRUM methodology has proven to generate very valid results in development environments with small and highly motivated teams, but being focused on satisfying the customer, it may neglect the treatment of security. On the other hand, including security from the beginning of the analysis can slow down value generation.

The proposed methodology is an extension of the traditional SCRUM, but with post-delivery processes that ensure that security is well-considered in the new implementation. This process can even be parallel to the implementation of new user stories and should be carried out primarily by a security specialist.

In the methodology, the figure of the security specialist is claimed as a necessary element in software development, as well as performance, efficiency and effectiveness, and even user experience. The SCRUM methodology has demonstrated very valid results in development environments with small and highly motivated teams, but being focused on satisfying the customer, it may ignore the treatment of security. On the other hand, including security from the beginning of the analysis can slow down value generation.

The proposed methodology is an extension of the traditional SCRUM, but marking some processes after the value increase, which guarantee the good contemplation of security in the new implementation. This process can even be parallel to the implementation of new user stories and should be carried out mainly by a security specialist.

In the methodology, the figure of the security specialist is claimed as a necessary figure during project development, as much or more than the figure of a scrum master, for example. Just as a specialist in team management is necessary, so is the figure of a security specialist. This is because only this specialist will be able to analyse security needs in a holistic way, with sufficient awareness and actuality about security, in all dimensions of the application, such as infrastructure, software development, databases, integration, backups, monitoring, or traceability.

While the developer team is focused on building functionalities that guarantee the viability of the project, the security specialist will be focused on making these functionalities secure. The activity of both teams is complementary, with functionality always taking precedence. This avoids paralysis by analysis, or conditioning functionality to security aspects. Although in an agile environment with a certain tendency for the lean paradigm, it is possible to delay decision-making until a functionality is fully clarified.

Another advantage of using a security responsible person is that security is not diluted among the development team [12]. It can happen that because there is no direct responsible person, a security issue is not detected, analysed, and resolved. This leaves an exposed vulnerability in the system. It is necessary to define responsibilities and to delimit the competences of each member of the group. The security officer is therefore the competent member of the team, who ultimately decides on the necessary mechanisms, the timing of their implementation, and the extent to which security levels can be negotiated.

He or she will also be the person to ensure strict compliance with the legal aspects of the functionalities.

Finally, a great advantage is that the methodology includes the entire team in the reviews and retrospectives. This makes the cybersecurity culture flow through the entire project, not just the security specialist, as the team will be able to see the real scope achieved, including the vulnerabilities detected and the countermeasures put in place. This helps training and learning, cybersecurity awareness, and the full team to end up participating in securitisation, directly or indirectly. It can also make it easier for the team of developers, in their daily work, to facilitate or anticipate security measures, paving the way for the expert.

The usefulness and validity of the methodology has been demonstrated through its application in a real project. In this case, a security expert who is part of the development team has been responsible for monitoring vulnerabilities and generating measures. The greatest contribution of the proposal is that security is implemented as the project grows. In other projects where methodologies based on the complete analysis of the system have been used, such as Magerit, security plans are achieved, but they exist afterwards. This means that the system may have been

exposing its vulnerabilities for an undetermined time. Using S-SCRUM, vulnerabilities are discovered at the beginning of the project, where there are only a few elements and therefore fewer vulnerabilities. By being detected and resolved from the beginning of the project, the securitisation process is simpler. And, above all, the process is formalised along with the development methodology, while, without this approach, you have to be confident that each actor in a development will be committed to security. In this approach, there is no need to depend on a developer to perform an activity that is not explicitly assigned to him, all the responsibility is concentrated on the security expert.

5 Conclusions

This paper has presented an extension of the SCRUM methodology that guarantees a correct implementation of the appropriate securitisation mechanisms in an agile environment. To this end, it is proposed to extend the processes and roles of the traditional methodology with: a new artefact called security story; a new role, the security expert; and two processes, one for analysis and the other for security implementation. This new methodology is called Security SCRUM or S-SCRUM.

The methodology was developed in the project Smart University, during the development of the new platform for smart city environments that is being developed to provide services to several Spanish universities. The methodology is the result of the need for agile development that rewards the delivery of value, but at the same time guarantees the correct securitisation of the systems.

The methodology claims the figure of the security expert as the person responsible for the analysis and decision-making on security mechanisms and measures. A specialised figure is dedicated to this type of issues, because, in large projects, it is necessary to centralise such important work in a perfectly identified figure.

The methodology has been successfully used in the development of the project and allows security to be considered at the same time as development, providing it with the same characteristics as software development. These include agility, the ability to change, and adapt along with the functionalities that appear or are altered. And, above all, to generate the measures that need to be applied because there really are elements that require them.

One of the short-term tasks is the formalisation of the new artefact, the security stories. There is a lot of work in the literature related to user stories and their correct formulation. The main line of work is to take advantage of these proposals to generate a characterisation of these security stories, in order to facilitate the work of the security expert, and at the same time make the result of their generation more standardised. In this way, security stories can be extrapolated from one system to another, as long as similar conditions exist.

In the long term, and following this process of standardisation of security stories, the generation of a catalogue of story patterns is proposed, which would simplify and speed up the work of the specialist. These patterns would allow the specialist to

select from among those that best adapt to his or her needs, once a vulnerability has been detected. And they would establish the best practices and the most recurrent stories in software development.

References

1. Gil JF, Úbeda SS, Carmona RM (2022) Unidigital project: the accessible university of the 21 th century: ındex termtowards the digital transformation of the Spanish University system. In: 2022 ınternational conference on ınclusive technologies and education (CONTIE). IEEE, pp 1–4
2. Ugwuanyi S, Irvine J (2020) Security analysis of IoT networks and platforms. In: 2020 international symposium on networks, computers and communications (ISNCC). IEEE, pp 1–6
3. Prabukusumo MA (2022) Big data analytics for cyber security. Proc Inform Conf 8(15):28–33
4. Stewart F (2004) Development and security. Conflict Secur Dev 4(3):261–288
5. Valdés-Rodríguez Y, Hochstetter-Diez J, Díaz-Arancibia J, Cadena-Martínez R (2023) Towards the integration of security practices in agile software development: a systematic mapping review. Appl Sci 13(7):4578
6. Alsaadi B, Saeedi K (2022) Data-driven effort estimation techniques of agile user stories: a systematic literature review. Artif Intell Rev 55(7):5485–5516
7. Takeuchi H, Nonaka I (1986) The new product development game. Harv Bus Rev 64(1):137–146
8. Ereiz Z, Mušić D (2019) Scrum without a scrum master. In: 2019 IEEE international conference on computer science and educational informatization (CSEI). IEEE, pp 325–328
9. Thomas TW, Tabassum M, Chu B, Lipford H (2018) Security during application development: an application security expert perspective. In: Proceedings of the 2018 CHI conference on human factors in computing systems, pp 1–12
10. Secretaría de Estado de Administraciones Públicas (2012) Magerit v.3: Metodología de análisis y gestión de riesgos de los sistemas de información
11. University of Alicante (2023). UniDigital Smart University Project. Corporate website of the project. Available online https://web.ua.es/es/smart/unidigital/proyecto-smartuni-unidigital.html
12. Beznosov K, Kruchten P (2004) Towards agile security assurance. In: Proceedings of the 2004 workshop on new security paradigms, pp 47–54