Smart City Software

Simonofski, Anthony; Sokolvak, Daria; Serral Asensio, Estefania

Publication date:
2022

Document Version
Peer reviewed version

Smart City Software: A Review of Development Methodologies and Modelling Languages

Chapter · January 2022

DOI: 10.1007/978-3-031-07478-3_3

3 authors, including:

Anthony Simonofski
University of Namur
40 PUBLICATIONS **340** CITATIONS
SEE PROFILE

Estefanía Serral
KU Leuven
86 PUBLICATIONS **860** CITATIONS
SEE PROFILE

Smart City Software: A Review of Development Methodologies and Modelling Languages

Anthony Simonofski[1][0000-0002-1816-5685], Daria Sokolvak[2], and Estefanía Serral[2][0000-0001-7579-910X]

[1] University of Namur, Rue de Bruxelles 61, 5000, Namur, Belgium
[2] KU Leuven, Oude Markt 13, 3000 Leuven BELGIUM.
anthony.simonofski@unamur.be

Abstract. Smart cities aim to improve citizens' quality of life using technology. Due to their size, smart city projects often rely on dedicated software that is built using development methodologies and modelling languages. This paper aims at discovering what software development methodologies and modelling languages are being used in a Smart City context as well as revealing the drivers behind these choices. To do so, a literature review and six semi-structured interviews with practitioners in Belgian smart cities were conducted. The results demonstrate that there are various software development methodologies (Waterfall Model, Agile, Scrum, AUP, Hybrid Agile Methodology) and modelling languages (UML, SysML, BPMN, DSML, informal modelling) applied in a smart city context. This paper contributes to the understanding of the current state of software development methodologies and modelling languages within the smart city context. Moreover, by taking a closer focus on Belgian Smart Cities, it sheds light on the concrete state of practice and highlights the drivers and challenges associated with each approach.

Keywords: Smart City, Software, Modelling Language, Software Development.

1 Introduction

During the last few years, we have observed an escalating tendency of a substantial number of people shifting towards living in the urban areas (Caragliu *et al.* 2011; Gaur *et al.* 2015). This growing number of city residents pushes city authorities to find ways for sustainable management of the increasing amount of organizational, technical, physical, and social concerns that emerge due to the high concentration of people. Rapid urbanization brings major challenges to the administration and general infrastructure of the city. These are a deficit of resources, falling into decay infrastructure, price fluctuations and scarcity of energy, troubles with human health, global environmental issues, and demand for improvement of economic potential (Nam & Pardo 2014). Despite this, Smart City projects can improve these challenges and create economic opportunities and social advantages for citizens.

There is no unified and standard definition of a Smart City. Common for any Smart City remains the aim to deliver effective services to people, regulate and improve infrastructure, foster cooperation of economic stakeholders to promote progressive and innovative solutions in private and public sectors (Albino *et al.* 2015). The following objectives are being obtained via the deployment of two vital elements that fuel Smart City: Information Technology (IT) and human capital (Ahvenniemi *et al.* 2017; Hollands 2008). Smart Cities are complex systems composed of tons of digitalized and interconnected operations and ecosystems. This complexity requires a Smart City system to be designed, managed, and governed properly. In the heart of the Smart City, there are numerous devices and sensors that are being interconnected via high-speed network and together they form the Internet of Things (IoT) and generate an enormous amount of valuable data (Gaur *et al.* 2015). Data, consequently, boosts the creation of modern services for citizens, companies, and public administrations (Zanella *et al.* 2014). To properly configure, manage, and monitor IoT devices used in city infrastructure, the assistance of Smart City software is in the center of importance.

To develop a successful software system, a suitable and appropriate methodology should be chosen (Qureshi 2012). The term methodology corresponds to the systematic approach to a process, following a predefined plan and appropriate management that guarantees that planning is being respected. Any software development methodology is associated with the set of steps required to be executed with the aim of delivering the software or piece of software as a result. These steps are: requirements, design, coding, testing, roll-out, and maintenance (Qureshi 2012). During this software development lifecycle multiple models are being constructed (Elaasar & Conallen 2013). Specifically, different modelling techniques are employed in the requirements gathering stage of software development methodology and in the design step responsible for specifying how the information system will be implemented (Qureshi 2012). In general, modelling is an irreplaceable part of any system and serves a set of purposes: it helps to simplify reality, facilitate communication among stakeholders, and have different views on the system (Elaasar & Conallen 2013). To perform modelling there is a need for modelling languages that will determine: how the models must be created, their elements, notation, syntax, and semantics (Al-Fedaghi & Alahmad 2018).Smart City software is not an exception: It also requires a suitable software development methodology and modelling languages that support it.

Many papers have already discussed characteristics, advantages, and challenges of different software development methodologies (Vijayasarathy & Butler 2016) and modelling languages (Evensen & Weiss 2010). However, the investigation of existing studies has shown that there is a lack of literature dedicated to the description and comparison of different software development methodologies and modelling languages for a specific Smart City context. Therefore, the goal of this paper is to dig deeper into these two vital components of smart city software. More specifically, this paper is focused on answering the following three Research Questions (RQ):

- RQ1: What software development methodologies and modelling languages have been used or suggested to be used in Smart City projects?
- RQ2: What advantages do these methodologies and modelling languages bring?

– RQ3: What software development methodologies and modelling languages are now used in Belgian Smart Cities and what is the reason behind these choices?

2 Methodology

2.1 Literature Review

Since our first objective is to discover the current state of software development methodologies and modelling languages in Smart City context, literature review was considered a proper method to give an answer. To find articles that describe software development in Smart Cities we started with several combinations of the obvious search terms: *software development Smart City*. Based on this term the following queries were executed: *(software OR development OR projects OR process OR application OR method OR methodology) AND "Smart City"*. To extract specific information about modelling languages, we executed the following search query: *(design OR modelling) AND Smart City*.

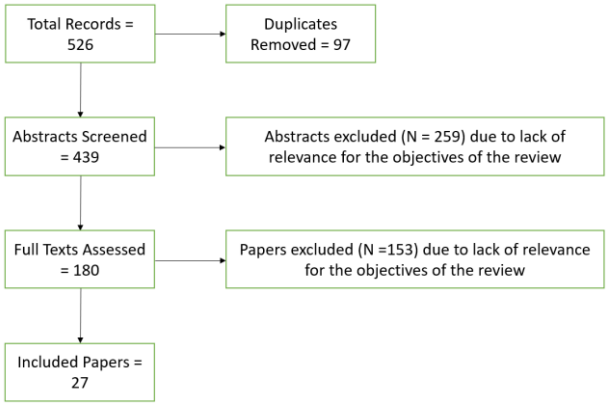

Fig. 1. Literature Review Methodology

The following inclusion criteria were developed: the study is written in English; the study is published between 2000-2020, to ensure that the article is recent and up to date; the study is cited in other research papers; the study cites other research papers; the study is written in a consistent and academic way; the study is aligned with the purpose of the literature review: it describes software development methodologies and/or modelling languages used in Smart Cities and reasons behind this choice. The most heavily used database was Google Scholar since it gives an extensive overview of the existing publications in multiple databases. Complementary searches were performed on other databases such as IEEE Xplore, Scopus and Web of Science. The procedure to select a study to be added to the paper is the following: first check the title and read the abstract. If the abstract of the paper is related to the purpose of the literature review, proceed with a full read of the paper; otherwise, discard it. If after full critical reading

all inclusion criteria are met, the study is selected and utilized in the paper; otherwise, it is discarded. In total, as highlighted in Figure 1, 27 papers were finally selected. 14 articles were selected for software development methodologies and 13 were selected for modelling languages. These papers can be found in the Appendix of this paper.

2.2 Interviews

To understand the use in practice in Belgium, we performed semi-structured interviews. This type of interview comprises a set of open-ended questions that tackle the topic the research aims at investigating. This research is considered beneficial since the open-ended nature of the questions identifies the focus of the interview yet provides room for a more detailed discussion between interviewer and respondent. The fact that the interview is not strictly structured permits to discover personal opinions and experience of the respondent (Guest *et al.* 2006). Six people participated in the interviews. Four of them were representatives of Smart City of Ghent (Belgium) and are employees of the organization named District09. District09 was defined by the respondents as the ICT provider for Ghent city. Other two respondents are employees of Digipolis Antwerpen, the IT partner of the city of Antwerp (Belgium). Table 1 summarizes the background of respondents in terms of Smart City they are related to and position occupied.

Table 1. Interviewees' information

City	**Position**
Ghent	Business Analyst
Ghent	Business Analyst
Ghent	Innovation Lead
Ghent	Innovation Lead
Antwerp	Solution Architect
Antwerp	Enterprise Architect

The type of qualitative data analysis that we have selected for this research is thematic one. Thematic analysis was performed according to (Braun & Clarke 2014) and comprised six sequential steps: data familiarization, initial code generations, searching for themes, reviewing themes, defining and naming themes and producing the report. We started from transcribing recorded interviews into textual format. After this, interview transcripts were carefully read and meaningful units, recurrent ideas and main issues in data were highlighted. After thorough review of data, we performed initial coding by selecting relevant parts of text and assigning them a code name that captures the sense included in it. Coding executed has an inductive nature meaning that codes emerge progressively from the data content and no codebook was developed in advance (Medelyan, 2020). After the first round of coding was finished, the procedure was repeated to refine the code names and arrive with polished codes. Based on the acquired set of codes, we aimed at searching for themes by sorting initial codes into potential themes and sub-themes. The outcome of this step was a set of candidate

themes, sub-themes and code associated with them. Next, themes and sub-themes were reviewed and refined making sure that they indeed capture what they were supposed to. Themes and sub-themes were then defined and renamed if necessary, highlighting the actual meaning stored in them and their differentiating characteristics. The coding scheme containing overarching themes, sub-themes, distinguishing characteristics, and example from transcript can be found in the supplementary material online[1].

3 Results

3.1 Methodologies and Modelling Languages in Smart City projects

Based on our literature review related to software development methodologies, it was possible to identify three types of methodologies that were used or suggested to be used in different Smart City software projects: Traditional (Waterfall), Agile (General Agile, Scrum and AUP) and Hybrid Agile Methodology. Waterfall Model is characterized by sequential development, defined and clear requirements, testing only taking place after coding, documentation creation after each phase and test team involvement during the testing stage only. Agile is a framework that manifests client's satisfaction through continuous frequent delivery of functional software, welcomes changing requirements and encourages constant collaboration among motivated individuals. Table 2 illustrates the software methodologies that are being applied for different Smart City projects, the types of projects they were executed in, and the benefits (reported in the selected papers) of using them.

Table 2. Software Development Methodologies

Methodology	Applications	Benefits
Waterfall	Prototype of the smart office system for IOT-based security system	• easy to understand; • easy to coordinate; • structured documentation; • minimal resources ; • easy implementation; • quality of development process; • minimized overhead; • better cost estimation;
Agile	• InterSCity Environmental monitoring system microservices-based platform (Custom Agile) • Smart City Bus Application (Scrum) • Blockchain- based software (Scrum)	• time and cost savings; • less documentation; • feedback from end-users; • faster implementation; • transparency; • lightweight development process; • rapid delivery of value; • self-organizing team;

[1] https://zenodo.org/record/5887609#.Yeq08f7MI2w

	• e-Rakorev Smart City Application (AUP)	• acceptance of changing requirements; • earlier data visualization.
Hybrid-Agile Methodology (HAM)	Smart City Procurement	• allows to estimate time and budget; • improved planning; • resource sharing; • reduced cost of misunderstanding and poor communication involvement of users; • increases acceptation; • educates citizens in project-related topic; • strengthens citizens' interest in the city.

After investigation of modelling languages, we identified four modelling languages that were applied in different Smart City software development projects: UML, SysML, BPMN, DSML. Table 3 illustrates the software modelling languages that are being applied for different Smart City projects, the types of projects they were executed in and their reported benefits.

Table 3. Modelling Languages

Language	Application	Benefits
UML	• e-Rakorev Smart City Application • Intelligent cloud-based car parking service • Location aware mobile services for a Smart City	• common clarity about system operation; • effective communication among stakeholders; • guarantee that the correct system is being developed; • support of the validation; • traceability from requirements to low-level design.
SysML	• Integrated Smart City System	• unambiguous; • enforces consistency; • prevents attempts at making incompatible connections; • improves precision and efficient communication.
BPMN	• Bike Sharing System • Report broadcasted by police to citizens via the Telegram channel	• standardization; • ease of use; • rich set of symbols; • no specific software required; • improves business processes.
DSML	• Smart Service System • Citizen Mobility System	• increased modelling productivity; • allows better coverage of the target domain; • understood more easily by modelers and model users;

		• use of expertise to share knowledge within the development team.

3.2 Methodologies and Modelling Languages in Belgium: State-of-Practice

Software Development Methodologies

All the respondents, both from Ghent and Antwerp, confirmed that their organizations try to be as Agile as possible and use Agile methodology most of the time. However, later participants explained that the development process is not completely Agile, even though it is called this. The requirements stage is completed in the Traditional Waterfall way, meaning that all the requirements are very detailed, clear, and consistent. The reason for this, as explained by our participants, is that Smart City projects are complex projects with many stakeholders involved and they are dependent on grants, funding, subsidies and on multiple governmental organizations that provide their support. When a proposal about the solution is created, it should include detailed, clear requirements and estimations to be present to relevant parties for approval. Business analysts from Ghent Smart City specified that when the new project initiation is received, they are responsible for conducting the whole research about the project and for producing a report containing all the strictly specified requirements, and a clear estimation of the timeline and budget needed. Antwerp representatives have also admitted that in the proposal that is being generated for their projects all requirements are very detailed, clear and are gathered and specified in Traditional Waterfall manner. The development itself, on the other hand, is being performed in Agile way with continuous incremental delivery. When asking about the reason for choosing this hybrid methodology, all respondents explained that the Traditional part of methodology applied to requirements is, first, required due to multiple stakeholders' dependency. Some Smart City representatives stated that, even though this way of dealing with requirements is necessary in Smart City projects, this method is also beneficial since it allows to avoid extra cost and time spending on modification of the solution.

Modelling Languages

All respondents mentioned that there is no standard modelling approach undertaken for software projects in their organizations and the selection of the models to be created depends on the business analyst responsible for their production and the characteristics of the project itself. Business analysts, as described by our respondents, are quite free to choose the way to deal with visualization of the requirement and design of the system. There are business analysts who create many different visualizations, while some business analysts barely create any models at all. When asking if the respondent find this unstructured modelling approach sufficient, we got an answer that it develops models that are clear, intuitive and understandable and this is far more important than sticking to strictly defined modelling languages. Therefore, practitioners focus more on understandability within their organization through ad-hoc modelling languages than one understandability in a broader context. Business analysts also underlined that there is no demand for standard UML diagrams since they require constant update. On the

other hand, another business analyst from Ghent told us that he personally does not create any specific models. Business analyst, nevertheless, admitted that he can develop a Context Model sometimes. It was described as a useful model since it provides a useful overview of external factors and the system's interaction with them. Business analyst also explained that there is no specific modelling language applied for Context Models creation and named the approach used for this an "informal way" of models' generation. In favor of this modelling technique respondent stated that this approach is understandable, simple and allows for creativity. When investigating the challenges that are associated with certain modelling techniques, it was noticed that some of the respondents associate UML with complexity, maintenance that arises from changing requirements and time consumption. One of the participants said that: "We avoid UML models that require updates, maintenance, probably explanation even, we try to be high-level. Of course, different business analysts do this the way they consider it to be relevant. But no standards for this and no desire to spend time, effort, and money on structural modelling". One respondent also explained that BPMN-based models can sometimes be too complex or lack certain features. Therefore, the Service Blueprint is added as an extra model in this case. In general, participants tend to highlight that they are trying to avoid complex modelling, keep models clear, intuitive, lightweight and understandable. According to one of the respondents, "this is more important than the fact that I have to use strict analyst models". During one of the interviews, respondents suggested that focus should be shifted from the design stage, where these models are created, to the actual development and emphasis should be there. One more challenge that was announced and is not related to the specific modelling language, is the fact that advice about tooling is not given within the organization.

4 Discussion

Overall results of literature review led us to the conclusion that there is no specific methodology that fits all the Smart City software development projects. This finding is in line with the conclusions previously made in the literature related to the general software development projects (Andrei et al., 2019; Awad, 2005; Vijayasarathy & Butler, 2016). However, Hybrid Agile Methodology seems to be the most popular one in the Smart City projects performed by the interviewees. This makes sense as this method combines the benefits of the two other methods while addressing some of their challenges or limitations. With respect to the modelling languages, literature review results introduce four modelling languages implemented in multiple Smart City software projects: UML, SysML, BPMN and DSML. The findings with respect to the benefits demonstrate that all of the modelling techniques facilitate communication among stakeholders. UML leads to evasion of costly remarks and gives common clarity of the system, while SysML is unambiguous, scalable, consistent and allows to detect errors in early stages. BPMN is concluded to be easily understandable and facilitates involvement of the stakeholders. DSML, finally, is beneficial mainly due to its ability of customization for certain purposes, reliability, and reusability. One more important finding is that UML tends to be considered as time-consuming, expensive, effort demanding and can sometimes provoke miscommunication. When looking for

modelling languages applied in Smart City projects in Belgium, some interviewees mentioned that did not always use models. This, consequently, leads us to assume that the modelling part could have been skipped in these projects due to the drawbacks mentioned above. As we can see from the results, all the modelling languages introduced by Smart City representatives serve the facilitation of communication among various parties which again supports the arguments illustrated in former studies (Elaasar & Conallen, 2013).

Even though this paper provides a broad overview of software development methodologies and modelling languages in a Smart City context, this paper also has several limitations. The literature that refers to methodologies or modelling techniques in a Smart City context almost never explains the reasoning behind. Therefore, the answer to the second research question is more general and the link between the methodology or modelling technique chosen and the reasons behind this is missed. We tried to overcome this limitation by explicitly searching for the specific outcomes of the identified methodologies and modelling languages. The general lack of existing literature about the topic in the Smart City context shows that more research needs to be performed in this area. In addition, this study considers a limited number of interviewees (six people) of two Belgian Smart Cities, Ghent and Antwerp. This may have not allowed us to see a broader picture of possibilities and experiences. As such, in future work we suggest increasing the sample size to make results more representative. Moreover, we suggest making purposive sampling more accurate and direct the interviews towards more specific profiles. For example, when there is an aim to discover modelling techniques, it is suggested to consider business analysts, designers or modelers as a rich source of information.

5 Conclusion

This paper aimed at discovering software development methodologies and modelling languages applied in various Smart City projects, the reasoning behind their selection and have a closer look at the Belgian Smart Cities' experience with respect to these terms. From the literature study we discovered that Traditional (Waterfall), Agile (General Agile, Scrum, AUP) and Hybrid Agile are the methodologies applied for different Smart City projects. With respect to the modelling languages, we found UML, BPMN, SysML and DSML executed. Through the interview with Belgian Smart City representatives, we explored that they apply Traditional and Mixed methodology and execute UML, BPMN, ArchiMate, DSML and Informal modelling techniques. We also demonstrated benefits of the methodologies and modelling techniques derived from the literature and illustrated the experience of Belgian Smart Cities with respect to their approaches. While the limited number of respondents limits the generalizability of the findings, the paper provides valuable insight about possible approaches applied in Smart City context and sheds light on how Belgian Smart Cities manage software projects with respect to methodologies and modelling. Most importantly, it highlights justifications of the choices made and challenges faced by these Smart Cities. Further research is needed to discover software development methodologies and modelling

languages implication in other Smart Cities and investigate how Smart City context influences the choice.

References

Ahvenniemi, H., Huovila, A., Pinto-Seppä, I., & Airaksinen, M. (2017). What are the differences between sustainable and smart cities? *Cities*. doi:10.1016/j.cities.2016.09.009

Al-Fedaghi, S., & Alahmad, H. (2018). Integrated modeling methodologies and languages. In *ACM International Conference Proceeding Series*. doi:10.1145/3164541.3164605

Albino, V., Berardi, U., & Dangelico, R. M. (2015). Smart Cities: Definitions, Dimensions, Performance, and Initiatives. *Journal of Urban Technology*, **22**(1), 3–21.

Braun, V., & Clarke, V. (2014). What can "thematic analysis" offer health and wellbeing researchers? *International Journal of Qualitative Studies on Health and Well-Being*. doi:10.3402/qhw.v9.26152

Caragliu, A., Del Bo, C., & Nijkamp, P. (2011). Smart Cities in Europe. *Journal of Urban Technology*, **18**(2), 65–82.

Elaasar, M., & Conallen, J. (2013). Design management: A collaborative design solution. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. doi:10.1007/978-3-642-39013-5_12

Evensen, K. D., & Weiss, K. A. (2010). A comparison and evaluation of real-time software systems modeling languages. In *AIAA Infotech at Aerospace 2010*. doi:10.2514/6.2010-3504

Gaur, A., Scotney, B., Parr, G., & McClean, S. (2015). Smart city architecture and its applications based on IoT. *Procedia Computer Science*, **52**(1), 1089–1094.

Guest, G., Bunce, A., & Johnson, L. (2006). How Many Interviews Are Enough?: An Experiment with Data Saturation and Variability. *Field Methods*, **18**(1), 59–82.

Hollands, R. G. (2008). Will the real smart city please stand up? *City*, **12**(3), 303–320.

Medelyan PhD, A. (2020). Coding Qualitative Data: How To Code Qualitative Research. *Thematic Analysis Inc.*

Nam, T., & Pardo, T. A. (2014). The changing face of a city government: A case study of Philly311. *Government Information Quarterly*, **31**(S1), S1–S9.

Qureshi, M. R. J. (2012). Agile software development methodology for medium and large projects. *IET Software*, **6**(4), 358–363.

Vijayasarathy, L. R., & Butler, C. W. (2016). Choice of Software Development Methodologies: Do Organizational, Project, and Team Characteristics Matter? *IEEE Software*. doi:10.1109/MS.2015.26

Zanella, a, Bui, N., Castellani, a, Vangelista, L., & Zorzi, M. (2014). Internet of Things for Smart Cities. *IEEE Internet of Things Journal*, **1**(1), 22–32.

Appendix: Selected Papers in the Literature Review

Theme 1: Software Development Methodologies

Alshamrani, A., & Bahattab, A. (2015). A comparison between three SDLC models Waterfall Model, Spiral Model, and Incremental/Iterative model. *International Journal of Computer Science Issues (IJCSI), 12*(1), 106-111.

Balaji, S., & Murugaiyan, M. S. (2012). Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. *International Journal of Information Technology and Business Management, 2*(1), 26-30.

Del Esposte, A. M., Kon, F., Costa, F. M., & Lago, N. (2017). InterSCity: A Scalable microservice-based open-source platform for Smart Cities. In M. Helfert, C. Klein, & B. Donnellan. (Eds.), *Proceedings of the 6th International Conference on Smart Cities and Green ICT Systems (*pp. 35-46).

Fong, S. L., Wui Yung Chin, D., Abbas, R. A., Jamal, A., & Ahmed, F. Y. (2019). Smart city bus application with QR code: a review. In *2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)* (pp. 34-39).

Ghilic-Micu, B., Stoica, M., & Mircea, M. (2014). Collaborative environment and agile development. *Informatica Economica, 18*(2/2014), 32-41.

Ibba, S., Pinna, A., Seu, M., & Pani, F. E. (2017). CitySense: Blockchain-oriented smart cities. In R. Tonelli (Ed.), *Proceedings of the XP2017 Scientific Workshops*.

Kaleshovska, N., Postolov, K., Janevski, Z., Josimovski, S., & Pulevska-Ivanovska, L. (2015). Contribution of scrum in managing successful software development projects. *Economic Development/Ekonomiski Razvoj, 17*(1-2), 175-194.

Karouw, S., & Wowor, H. (2013). e-Rakorev: Towards governance planning, monitoring and evaluation of urban development for Manado SmartCity. In 2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS) (pp. 47-53)

Kishino, Y., Yanagisawa, Y., Shirai, Y., Mizutani, S., Suyama, T., & Naya, F. (2017). Agile environmental Monitoring Exploits rapid prototyping and in Situ Adaptation. *IEEE Pervasive Computing, 16*(2), 61-71.

Lom, M., Pribyl, O., & Zelinka, T. (2016). Hybrid-Agile Approach in Smart Cities Procurement. *20th World Multi-Conference on Systemics, Cybernetics and Informatics: WMSCI 2016*.

Prasetyo, T. F., Zaliluddin, D., & Iqbal, M. (2018). Prototype of smart office system using based security system. *Journal of Physics: Conference Series, 1013*.

Shankarmani, R., Pawar, R., S. Mantha, S., & Babu, V. (2012). Agile methodology adoption: Benefits and constraints. *International Journal of Computer Applications, 58*(15), 31-37.

Sharma, S., Sarkar, D., & Gupta, D. (2012). Agile processes and methodologies: A conceptual study. *International journal on computer science and Engineering, 4*(5), 892-898.

Vácha, T., Přibyl, O., Lom, M., & Bacúrová, M. (2016). Involving citizens in smart city projects: Systems engineering meets participation. In *2016 Smart Cities Symposium Prague (SCSP)* (pp. 1-6)

Theme 2: Software Modelling Languages

Ardito, C., Caivano, D., Colizzi, L., & Verardi, L. (2020). BPMN extensions and semantic annotation in public administration service design. In R. Bernhaupt, C. Ardito, & S. Sauer (Eds.), *Human-Centered Software Engineering* (pp.118-129).

Calderoni, L., Maio, D., & Palmieri, P. (2012). Location-aware mobile services for a Smart City: Design, implementation and deployment. *Journal of Theoretical and Applied Electronic Commerce Research, 7*(3), 74-87

Chaudron, M. R., Heijstek, W., & Nugroho, A. (2012). How effective is UML modeling? *Software & Systems Modeling, 11*(4), 571-580.

Cognini, R., Corradini, F., Polini, A., & Re, B. (2014). Modelling process intensive scenarios for the smart city. In M. Janssen, H. J. Scholl, M. A. Wimmer, & F. Bannister (Eds.), *Electronic Government* (pp. 147-158).

Dzidek, W., Arisholm, E., & Briand, L. (2008). A realistic empirical evaluation of the costs and benefits of uml in software maintenance. *IEEE Transactions on Software Engineering, 34*(3), 407-432.

Flowers, R., & Edeki, C. (2013). Business process modeling notation. *International Journal of Computer Science and Mobile Computing, 2*(3), 35-40.

Huber, R. X., Püschel, L. C., & Röglinger, M. (2019). Capturing smart service systems: Development of a domain-specific modelling language. *Information Systems Journal, 29*(6), 1207-1255.

Indulska, M., Green, P., Recker, J., & Rosemann, M. (2009). Business process modeling: Perceived benefits. In A. H. F. Laender, S. Castano, U. Dayal, F. Casati, J. P. M. de Oliveira (Eds.), *Conceptual Modeling - ER 2009* (pp. 458-471).

Ji, Z., Ganchev, I., O'Droma, M., Zhao, L., & Zhang, X. (2014). A cloud-based car parking middleware for iot-based smart cities: Design and implementation. *Sensors, 14*(12), 22372-22393.

Karouw, S., & Wowor, H. (2013). e-Rakorev: Towards governance planning, monitoring and evaluation of urban development for Manado SmartCity. In *2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS)* (pp. 47-53).

Muvuna, J., Boutaleb, T., Baker, K. J., & Mickovski, S. B. (2019). A methodology to model integrated smart city system from the information perspective. *Smart Cities, 2*(4), 496-511.

Prasanna A. T. *A Domain Specific Modeling Language for specifying educational games* (Master's thesis) Vrije Universiteit Brussel. Faculty of Science, Master of Science in Applied Science and Engineering: Computer Science, Brussels.

Rosique, F., Losilla, F., & Pastor, J. Á. (2017). A domain specific language for smart cities. *Proceedings, 2*(3), 1-7.