INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 20, Issue: 4, Month: August, Year: 2025
Article Number: 6807, https://doi.org/10.15837/ijccc.2025.4.6807

CCC Publications

The Impact of Human Factors on Software Development Processes Applying the Agile and Waterfall Methodologies: A Case Study Using Real Data

Š. Sielskaitė, D. Kalibatienė

Šarūnė Sielskaitė*
Department of Information Systems, Faculty of Fundamental Sciences
Vilnius Gediminas Technical University, Lithuania
Saulėtekio al. 11, Vilnius, 10223 Vilniaus m. sav., Lithuania
*Corresponding author: sarune.sielskaite@vilniustech.lt

Diana Kalibatienė
Department of Information Systems, Faculty of Fundamental Sciences
Vilnius Gediminas Technical University, Lithuania
Saulėtekio al. 11, Vilnius, 10223 Vilniaus m. sav., Lithuania
diana.kalibatiene@vilniustech.lt

Abstract

The software development process (SDP) contains many variables depending on the methodology chosen. Software development methodologies such as Agile or Waterfall can specify how and in what sequence software is developed. However, the SDP remains a hot topic among practitioners and academics, who seek to clarify the key aspects that influence the successful implementation of the SDP. As the SDP is a dynamic and knowledge-intensive business process (BP), one of its most impact components is the human factor (HF). It is difficult to predict how the HF impacts the sequence and results of SDP implementation. This paper proposes a new approach to investigate the impact of the HF on the SDP through the perspectives of different software development methodologies. The advantages and novelties of this approach include modelling HF uncertainties through their fuzzification in SDP activities, modelling different SDP methodologies using a case-handling approach, and simulating a dynamic, case-based SDP with real HF-related data collected from several IT organizations. The results show that the impact of the HF on SDP performance differs across the Waterfall and Agile methodologies. The results also allow researchers and practitioners working on software development projects to familiarize themselves with the impact of the HF on different SDPs, and to assess the degree of development risk associated with the SDP depending on the methodology chosen.

Keywords: software development process, human factors, Agile, Waterfall, fuzzy logic.

1 Introduction

The software development process (SDP) is a multifaceted and dynamic undertaking that plays a pivotal role in shaping the technological landscape. This intricate procedure involves a series of systematic steps, from the inception of project requirements to the delivery of a functional software product. The human factor (HF) can influence various business processes. It is very difficult to predict the extent to which HF aspect will influence the predicted results, and different subject areas may respond differently to the HF. The importance of human error is emphasized in aviation and in the safety of remotely piloted merchant ships [25, 34]. Beyond the technical aspects, the HF significantly influences the course and outcome of software development. The intricate interplay between developers, project managers, end-users, and other stakeholders introduces a layer of complexity that extends beyond code and algorithms. The HF encompasses communication, collaboration, creativity, decision-making, and adaptability, all of which are integral to the success of a software development endeavor. Understanding the profound impact of human elements on the SDP is essential in order to foster effective teamwork, manage expectations, and ultimately deliver software solutions that not only meet technical specifications, but also align with human needs and expectations. Fuzzy logic aids in interpreting uncertainties by providing a mathematical framework that allows for the representation of vague and imprecise information, enabling more flexible and realistic decision-making in complex systems. For example authors in [24] developed model that efficiently optimizes marketing project portfolios under uncertainty by using fuzzy rules. This approach helps managers select the best marketing initiatives to maximize returns within acceptable risk and budget limits.

The main SDP activities include analysis, design, development, testing, and maintenance, each of which are performed by humans [15]. Humans create and use software products, and are at the center of the dynamic, ongoing, intricate, and chaotic SDP. This is the main reason why understanding the impact of the HF on the SDP is significant. As the SDP is a dynamic and knowledge-intensive business process (BP), one of its most impactful components is the HF. Moreover, it is difficult to predict how the HF impacts the sequence and results of SDP implementation.

This research aims to contribute to a deeper analysis of the impact of the HF on the SDP by comparing different software development methodologies (SDMs) and how the impact of the HF varies with changes in the sequencing of the activities in the SDP. The main gap in existing approaches is lack of real historical project data and very small number of studies and experiments to have a clear vision of HF issues within software development. The main strength of this research is that real SDP data is used for experimentation and the validation of the proposed approach. The main contributions and strengths of this research are as follows: 1) An analysis of the impact of the HF on a Waterfall SDP, 2) An analysis of the impact of the HF on an Agile SDP, 3) A comparison between the Waterfall and Agile SDP methodologies, 4) The incorporation of real data into the experiment.

Main research questions are formulated as follows: how HF impact knowledge-intensive and dynamics BPs and how different sequence of tasks does influence SDP with HF aspect. The main novelty of this article is the proposed hybrid case-based approach with the application of CMMN SDP modelling and simulation, enriched with the human factor modelling component using ANFIS. Additionally, historical data from real SDP projects are used to verify the proposed method.

This paper is structured as follows: section 2 describes preliminaries; section 3 reviews the literature on the SDP; section 4 describes the approach taken by this research; section 5 presents a case study and the results of an experimental study by using SDP modeling and simulation; and, finally, sections 6 and 7 conclude the paper.

2 Preliminaries

The SDP encompasses the procedures involved in creating a software product, covering essential phases such as gathering requirements, design, implementation, testing, and maintenance [4].

The Agile methodology presents an iterative project management and software development approach that prioritizes swift and efficient value delivery to clients. Agile principles involve collaborative work with end users, self organizing cross functional teams, adaptive planning, evolutionary development, early delivery, continual improvement, and the flexibility to adapt to changing needs [22]. The Waterfall methodology is introduced and described as a sequential and linear approach to software development. The phases of the Waterfall methodology involve gathering requirements, design, implementation, testing, and maintenance, emphasizing the importance of completing each phase before progressing to the next [27]. Case Management Model and Notation (CMMN) is renowned for its adaptive case management approach, offering decision-making support through suggestions while keeping individuals in management roles. Unlike workflow-based methods that use a control flow, case management focuses on providing information about the process case to workers [12].

Agent-based simulation is a computational modeling technique widely used in various fields, including social sciences, economics, ecology, and, notably, the domain of software engineering and software development. Agent-based simulation involves modeling entities, known as agents, and simulating their interactions within a specified environment to observe emergent behaviors and patterns [13].

A fuzzy set is one in which each of the elements have degrees of membership, such as set A in the example below [23]:

$$A = \{(u, \mu_A(u)) \mid u \in U, \mu_A : U \to [0, 1]\} \tag{1}$$

where A is in the universe of discourse U, and $\mu_A(u)$ is a membership function (MF) of u in A. The value $\mu_A(u)$ represents the grade of membership of u in A, and is interpreted as the degree to which u belongs to A.

Different types of MF shapes can be found in the literature, as presented by Choi and Rhee [8]. In this research, trapezoidal [13] MF shapes are applied since they offer more flexible representations than triangular functions.

3 Related Works

Almost every dynamic BP is influenced by the HF, with Ruiz and Salanitri [28] noting its critical role in effective software development and SDP success. Guveyi, Aktas, and Kalipsiz [15] further explored this by analyzing various HFs that impact software quality, categorizing them as personal, interpersonal, and organizational factors. They highlighted key influences like education, experience, motivation, and job satisfaction, while acknowledging that specific HF impacts on SDP outcomes were not fully examined.

Authors in [20] examined social and human factors affecting productivity in software development teams and project management, emphasizing commitment as a key factor influencing project success and cost. They found a strong correlation between collaboration, team cohesion, and SDP capabilities, underscoring the need for integrated teamwork for project success. Limitations of their study include result generalization, limited study details, and a small number of relevant measurements.

Almost every academic and practitioner agrees that the HF is a crucial part of the SDP. Capretz [7] discussed software as a product that is created by humans, observing that humans are more complicated and less predictable than software. Thus, to predict the SDP with the HF as a variable becomes difficult and requires deeper analysis. The author suggests that software professionals should recognize that the people involved in the SDP are as important as the processes and the technology itself. Studies featuring internal empirical experiments are emphasized by Capretz as one of the keys to product improvement.

Pirzadeh [26] investigated HF impacts on SDP from development lifecycle and software management perspectives, focusing on Waterfall methodology and splitting HF influences into development and management phases. The findings indicate that requirement engineering is the most studied software management phase and that developers are the primary human role in SDP. However, the study suggests a lack of research on how various human roles dynamically impact process performance, highlighting an ongoing emphasis on technical aspects over human factors, which significantly affect SDP success and quality.

Amrit, Daneva & Damian [2] emphasized that the HF is very important in software development. For this reason, the authors performed a citation analysis among research that analyzes the impact of the HF in software development. They concluded that the study of the HF in software development could be considered a subfield of empirical software engineering, and therefore shares methodological issues with social and behavioral sciences. The main flaw in this analysis is that the amount of research that the authors relied upon was somewhat small (featuring only the 20 most commonly cited articles/books related to the HF in software development) and covered only one decade (2001–2010).

Other authors have analyzed the impact of the HF on only a specific stage of software development. For example, Gonçalves et al. [14] investigated the impact of the HF in the context of software testing – a key process that ensures a reliable, high quality product but that requires a lot of human work. The final quality of software can be impacted by the HF, but the specific goal of this research was to identify the HF (cognitive, operational, and organizational) present in the test process and to define the influence of HFs during its execution. The authors concluded that the HF creates both motivational and demotivational aspects. The results show the dissatisfaction of some professionals with various cognitive, operational, and organizational aspects. Thus, the HF must be observed, understood, and treated as the focus of a software factory, and should be qualified and managed by a management team that values intellectual capital.

Machuca-Villegas et al. [21] described an instrument for measuring perceptions of the social and human factors that influence the productivity of software development. The main problem area that the authors addressed was the notion that social and human factors are not explicitly addressed in any manner that may facilitate their identification and adoption in order to propose strategies for improving the productivity of the software development team. To address this issue, the research sought to assess these factors from the standpoint of software development professionals. The authors concluded that the development and evaluation of an instrument used to measure perceptions of the social and human factors that exert influence on the productivity of the software development team, and which could then be used by development teams, was necessary. The main disadvantage of this research is its lack of results that might illustrate how this instrument impacts the productivity of software development from the perspectives of social and human factors.

In a further tertiary study, Dutra, Diirr & Santos [10] analyzed and summarized HFs and their influence on both software engineering development teams and agile software development teams. In their research, the authors detailed all HFs that influence software development. The HFs were then divided into groups based on the results of their influence (for example: influencing a team member, influencing a team, influencing an organization, etc.). As a result, the authors identified 101 HFs that influence software development activities from different perspectives. The main disadvantages and limitations of this research are its descriptive validity, theoretical validity, and interpretive validity.

Capretz & Ahmed [6] took personality types as HFs and tried to measure the impact of these HFs on software development. They divided personality types into different groups and analyzed how those HFs influence different SDP roles. The authors concluded that no single personality type fits the wide spectrum of tasks that encompass the engineering of software. Instead, better software results come from the combined efforts of a variety of mental processes, outlooks, and values.

Other authors [33] stress the importance of HF in generative design, showing that incorporating ergonomic data from anatomical scans enables algorithms to produce functionally optimized, user-centered designs. This approach highlights the value of aligning design workflows with user needs, especially in fields like physical therapy, and calls for clear methodologies and intuitive tools to integrate human-centered principles. Prioritizing HF in generative design, the study concludes, leads to innovative solutions that more effectively meet user needs and drive advancements across various fields.

Authors in [25] underscore the critical role of HF in aviation safety. Through the analysis of incident and accident reports, a predictive model was developed to anticipate fatalities based on the accident's cause. By integrating data on contributory causes, flight phases, damage sustained, and mortality, the study applied the HFACS taxonomy to correlate HF with accident outcomes. The proposed machine learning demonstrated results, affirming the importance of HF policies in mitigating accidents and failures within the highly regulated aviation industry. The research aims to enhance safety standards by identifying key HF contributing to fatal accidents, guiding future investments and procedural improvements.

Table 1: Comparison of related works

Reference no. (year of publication)	**Problem**	**Main goal/aim**	**What is the HF in the SDP?**	**How does the HF influence the SDP?**	**Method**	**Results and conclusions**	**The most influential social or human factor**
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
15 (2020)	Any factor affecting people will directly affect software quality and success.	Reveal what factors affect humans.	At each stage of the SDP, human knowledge, intelligence, and experience affect software product quality.	Due to the HF across the entire SDP, software quality depends on human behaviors.	Systematic literature review (SLR).	Personal factors (experience and education) are the most important category of HFs.	Human knowledge, intelligence, and experience.
20 (2021)	Software companies need to measure their productivity	Establish a set of measures related to the social and human factors that influence productivity in software development teams.	Social and human factors are of particular importance because they impact the results of software projects and are considered important elements affecting costs.	Personal aspects and human activities represent an opportunity to improve productivity.	SLR, survey and mapping.	The HF impact could be different depending on different SDP methodologies.	Commitment, collaboration, team cohesion capabilities, and SDP experience.
7 (2014)	A lack of studies and empirical experiments to understand HF issues within performance-oriented teams, collecting data and creating insights to improve overall SDP.	Emphasize the problem and stimulate researchers towards analyzing the impact of the HF impact on the SDP.	The software product depends on human activities, such as problem-solving capabilities, cognitive aspects, and social interaction.	The HF is a make-or-break issue that affects most software projects.	Review.	Lack of study of the impact of the HF on the SDP.	Problem-solving capabilities, cognitive aspects, social interaction.
26 (2010)	The SDP is a human-centered activity that highlights the impact of the HF and requires performance from different perspectives.	Identify and characterize the HFs influencing the SDP when applying different methodologies.	The SDP is a human-centered activity.	Humans play different roles in the SDP. This has an impact on process performance and success.	SLR.	- Lack of primary and secondary studies on the HFs related to the SDP. - HF impact depends on SDP methodologies.	Teamwork, communication, virtual teamwork, human resource management.
2 (2014)	Suggest that software engineering researchers draw on reference disciplines (i.e., Information Systems) and borrow well-established theories.	Conduct a citation analysis and identify the main theories in the HF.	Software development has been characterized in essence as a human activity where the HF plays a critical role.	The growing importance of HFs in software development research is clearly evidenced by the ICSE 2014 conference entirely devoted to HFs ("Social Aspects of Software Engineering").	Citation analysis (bibliometric analysis)	Main theories regarding the HFs in software development to come from a field related to behavioral and social sciences.	N/A

Reference no. (year of publication)	**Problem**	**Main goal/aim**	**What is the HF in the SDP?**	**How does the HF influence the SDP?**	**Method**	**Results and conclusions**	**The most influential social or human factor**
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
14 (2017)	Software testing is a key process that ensures a reliable, quality product and requires a lot of human work, where the final quality of the software can be impacted directly by several factors.	Identify the HFs (cognitive, operational, and organizational) present in the testing process and define the influence of these factors during their execution.	In software factories the HF represents the central investment capital to achieve productivity and quality.	The influence of HFs in the work environment is directly related to the quality of the product being developed and its productivity. Increase in efficiency and human effectiveness is the primary factor responsible for building in the development of software products.	Quantitative methods and survey techniques.	- The dissatisfaction of some professionals with various cognitive, operational, and organizational aspects. - HFs must be observed, understood and be the focus of a software factory.	Cognitive, operational and organizational aspects
21 (2021)	Social and human factors are not explicitly addressed in any manner that may facilitate their identification and adoption to propose strategies for improving the productivity of the software development team.	Assess the HF from the standpoint of software development professionals.	Social and human factors play an important role in software engineering.	Social and human factors may affect the productivity of the software development team.	Survey-based research.	Created an instrument that proves that social and human factors exerted an influence on the productivity of the software development team.	Perceptions.
10 (2021)	The HF influences individuals, the development team, and the software project activities.	Summarize HFs and their influence on software development teams.	The term “HF” represents a person’s physical or cognitive features, or social behavior.	Different HF types can influence the SDP in different aspects (project team size, project tracking, delivery time, etc.).	A tertiary study where Matic analysis is used to examine the resulting data.	The identified HFs and their influences can be considered most significant by IT organizations, researchers, and academics in SE practice.	A person’s physical or cognitive features, social behavior, sentiments or attitudes, communication, motivation, and collaboration.
6 (2010)	People are more complicated and less predictable than computers, thus the complexity of personality entails intricate dynamics that ultimately become an integral, yet often overlooked, part of software development.	To discern connections between personality traits and the process of software development.	Personality types as HFs: extroversion, introversion, sensing, intuition, thinking, feeling, judging, and perceiving.	Different types of personalities could be a good fit for one or another SDP role but may not fit other roles at all.	Research.	- No single personality type fits the wide spectrum of tasks that encompasses SE. A broad range of personality types is beneficial to SE. - Better SDP results come from the combined efforts of a variety of mental processes, outlooks, and values.	Extroversion, introversion, sensing, intuition, thinking, feeling, judging, and perceiving.

Reference no. (year of publication)	Problem	Main goal/aim	What is the HF in the SDP?	How does the HF influence the SDP?	Method	Results and conclusions	The most influential social or human factor
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
33 (2022)	Little research has been applied directly to the problem of ergonomics in generative design.	Develop a model for approaching generative design development work, oriented around human factors (ergonomics), and describe a case study from the PRIME-VR2 research project in which an algorithmic workflow utilized user scan data and 3D-printing technology to generate bespoke versions of a standard controller device.	The working distinction for this paper is that the hHF information (e.g., ergonomic, or psychological factors) forms the basis by which a human-centered design approach can be formulated.	HF in this research is described as ergonomic action.	Research.	- Discrete HF information can be fused with a generative design algorithm to create a framework and workflow for HCD-based generative design. - HF become a more important area of design and the use of discrete data and design intelligence becomes more common within generative design use.	Human anatomy and usability.
25 (2023)	Aviation demands have increased over the years; while safety standards are very rigorous, managing risk and preventing failures due to HF, thereby further increasing safety, requires models capable of predicting potential failures or risky situations.	To propose a model capable of predicting fatal occurrences in aviation events such as accidents and incidents, using as inputs the human factors that contributed to each incident, together with information about the flight.	In this research HF is analyzed in aviation sector from security perspective.	Many aviation accidents have been caused by human error. From this perspective further analyze was developed.	Model capable of predicting fatal occurrences in aviation events.	- Significance of HF. - Validation of predictive models. - Guidance for safety improvement.	Human errors.

From Table 1, we can see that the impact of the HF on the SDP is a topic that is under consideration among many researchers. In the literature review, we considered different perspectives from which we analyzed research. Firstly, we looked at problems that different authors raised within the same domain. All of these problems were related to the impact of the HF on different areas of the SDP: quality [15], productivity [20, 21], performance [7, 26], and testing [14].

Several main aspects are visible from the literature review, including the impact of software development methodologies [20, 26] and the quality of the product [14, 15]. Software development methodologies are themselves also influenced by the HF because of intensive human involvement in the SDP. In Agile methodologies, for example, the HF has the most significant impact on the SDP. The authors that emphasized software development methodologies in their research mostly analyzed commitment, collaboration, team cohesion, SDP capabilities and experiences [20], teamwork, communication, virtual team working, and human resource management [26]. From a quality perspective, the main HFs analyzed were experience, education [15], and cognitive, operational, and organizational aspects [14].

Although the different authors in Table 1 analyzed the impact of the HF on the SDP from different perspectives, they all agreed on one thing – the HF affects SDP productivity. For example, Guveyi, Aktas & Kalipsiz [15] observed that any factor affecting people will directly affect software quality and success. Pirzadeh [26] also pointed to the SDP as a human-centered process, which is why the impact of the HF on the performance of the SDP from different perspectives is highlighted.

Multiple authors have developed different approaches to researching the same or very similar problems (Table 1 Column 3). For example, some [2, 7] have aimed to systematize the impact of the HF in software engineering by performing systematic literature reviews. Other authors have tried to measure the impact of the HF [10, 21], while yet more have sought to establish a set of HFs that impact the SDP [6, 14, 15, 20, 26].

To identify and measure the impact of the HF on the SDP, it is first necessary to understand what the HF is. Various authors have presented different descriptions of the HF itself (Table 1 Column 4). Guveyi, Aktas & Kalipsiz [15] defined the HF as representing important elements that impact the results of software projects and affect their costs. Other authors [18] use the HF concept to represent a person's physical or cognitive features, or social behavior. Some authors [6] also classify HFs into types (i.e., extroversion, introversion, etc.) in order to assess which personality types are most suitable for a given software engineering task. In summary, the HF concept is multifaceted and can be viewed from various perspectives, whether perceived as representing personality types, parts of human behavior, or the features that describe a person.

Another important aspect that was analyzed in the papers in Table 1 is the question of how the HF influences the SDP. From the table, it can be seen that the HF impacts the SDP, but the forms of this influence are different. For example, some authors [6] emphasize that different types of personalities could be a good fit for one or another SDP role while not fitting others. This view is shared by other authors [15], who note that different types of HF could have different impacts on the SDP according to their roles. Findings relating to the impact of the HF on the SDP (Table 1 Column 5) can be summarized through the main software product features: quality [14, 15], costs [20], performance [26], productivity [14, 21], and delivery [10].

Column 7 in Table 1 presents the HFs that most strongly influence the SDP in each paper as follows: experience and education [15]; commitment and collaboration [10]; team cohesion, capabilities, and experience [20]; cognitive aspects of problem-solving capabilities and social interaction [7]; teamwork, communication, virtual team working, and human resource management [26]; and communication and motivation [10]. Most authors agree that the HF impacts the SDP, but this can vary according to HF type [6], software development methodology [10, 20, 26], and roles [6].

HF impact in different areas is the topic under analysis by other authors [25, 33]. HF can be considered not only as a human characteristic, but also as anatomical features [33] or human errors [25]. It also affects the processes of another subject area and can change the predicted results.

From this literature review, a primary set of HFs for future research was outlined as follows: motivation, collaboration, experience, education, communication, and commitment.

4 Materials and Methods

In this section, we describe our fuzzy and case-handling based dynamic SDP modeling and simulation approach. This consists of the following main elements: the SDP model, initial data preprocessing, ANFIS-based inference, CMMN SDP modelling, the simulation of the SDP model.

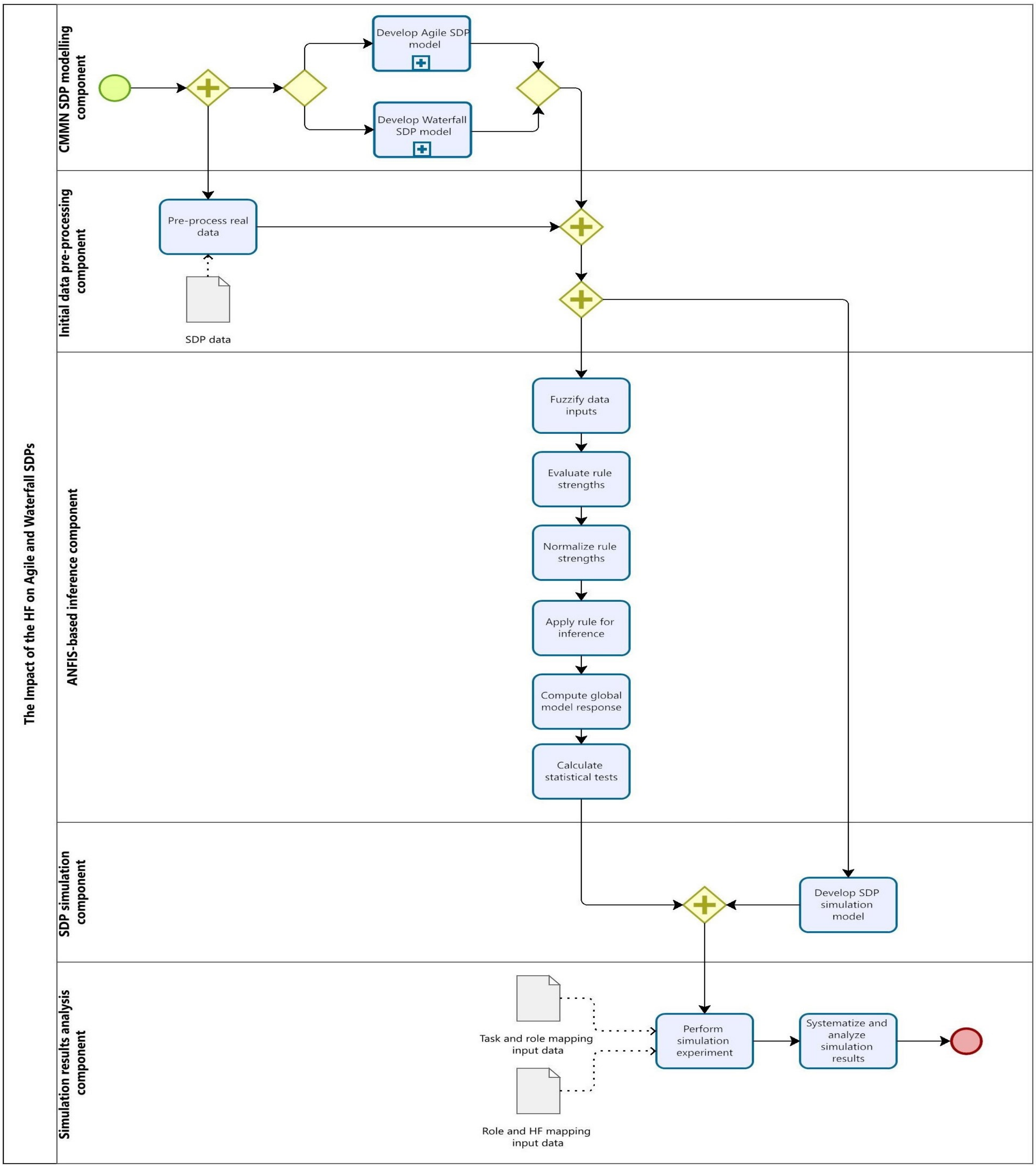

Figure 1: A BPMN depicting the impact of the HF on Agile and Waterfall SDPs.

Real data was first collected from software development projects, containing task information, duration, responsible resources, and deviations from estimated times. This data, incorporating HFs, was then loaded into ANFIS to perform fuzzification, rule evaluation, normalization, inference, and model response computation, followed by statistical tests. Concurrently, CMMN SDP models for Agile and Waterfall methodologies were created and transformed into CMMN SDP simulation models implemented in a prototype. Simulation was conducted using ANFIS-trained data and CMMN models, with input data organized into task-role and role-HF mapping tables, and results were then systematized and analyzed (see Fig. 1).

4.1 The SDP Model

In this research, the two most popular software development methodologies (i.e., Waterfall and Agile) are used to investigate the impact of the HF on the SDP. These two methodologies were chosen for this research as they are based on completely different principles, featuring a different sequence of activities, a different approach to changes in the environment and requirements, a different approach to the role of the HF in the SDP, etc. Agile is a flexible, iterative approach that prioritizes individuals, interactions, working solutions, and adaptation to change through iterative cycles known as sprints, and is widely used in frameworks like Scrum, Kanban, and Extreme Programming [9, 11, 30]. Agile's benefits include enhanced customer satisfaction, faster delivery, and adaptability to change. In contrast, the Waterfall model follows a linear, phase-based process—ideal for projects with fixed requirements—where each stage (requirements, design, implementation, testing, maintenance) must be completed before moving on [3]. Though Waterfall promotes discipline and predictability, it is less adaptable to change and less suited for complex projects.

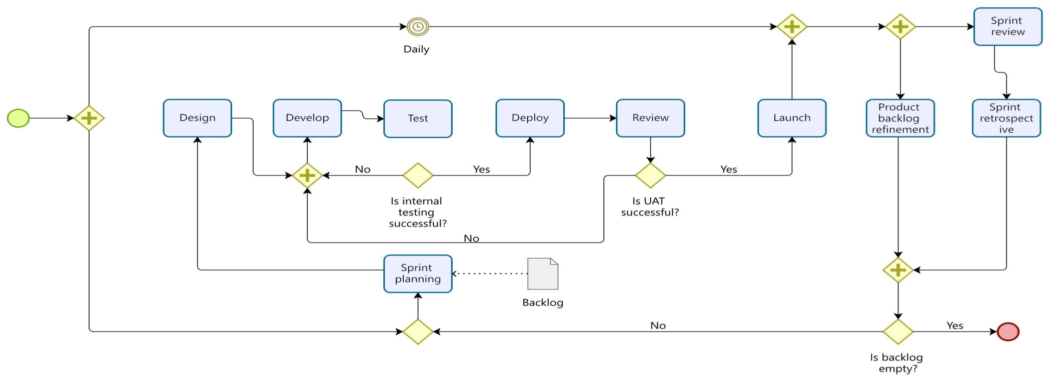

Figure 2: A BPMN depicting an Agile SDP [29]

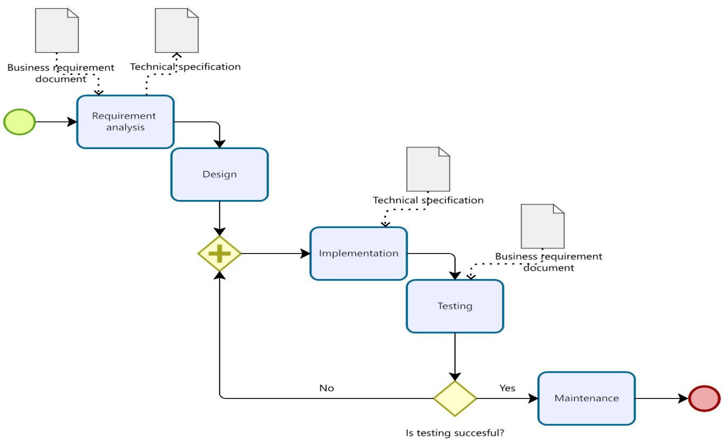

Figure 3: A BPMN depicting a Waterfall SDP.

The differences between Waterfall and Agile software development methodologies have been extensively analyzed in the literature. The primary distinction is that Waterfall is a conventional, linear approach characterized by sequential phases: requirements gathering, design, implementation, testing, and maintenance, making it suitable for projects with stable requirements [5, 31]. In contrast, Agile values iterative development, emphasizing adaptability, collaboration, and customer involvement, with key features like iterative cycles and frequent feedback that allow for responsiveness to changing requirements [16, 30]. These methodologies are often compared in terms of flexibility, with Agile offering a more dynamic approach than Waterfall's rigid structure. Furthermore, studies have examined their impacts on project success, team dynamics, and stakeholder satisfaction, enriching our understanding of their implications across various development contexts [1]. The input of this step is formed of aspects of the Agile and Waterfall methodologies: main activities and data, sequence rules, and other relevant information.The output of this step is BPMN-based Agile and Waterfall SDP models.

The overall project can be depicted by the following equation:

$$\mathrm{SDP}_i^j = \{(\mathrm{Task}_i, H_i, \mathrm{Resource}_i, \mathrm{HF}_i) \mid i = 1, \ldots, n\} \tag{2}$$

where: $Task_i$ – all assignments that should be completed in order to move to another assignment or result; H_i – actual time for task execution; $Resource_i$ – competence that can perform a task; and HF_i – human factor of a task executor.

4.2 ANFIS-based inference

ANFIS is used for fuzzy inference in this research because it combines the adaptability of neural networks with the interpretability of fuzzy logic, effectively modeling complex nonlinear relationships. This adaptive neural network framework [17] uses fuzzy IF-THEN rules (e.g., Equation (2)) with tailored membership functions to generate specified input-output pairs:

$$R_i : \text{if } (x_1 \in A_1^{(i)}) \wedge \ldots \wedge (x_n \in A_n^{(i)}) \text{ then } f_i = a_i^T \cdot x + b_i \tag{3}$$

where $x \in \mathbb{R}^n$ a vector of inputs characterized by an appropriate MF, and (a_i), (b_i) are the coefficients of linear Takagi–Sugeno consequents. Training ANFIS involves determining parameters related to both the premise (input parameters) and the consequences (output) using an optimization algorithm. The primary structure of ANFIS encompasses five layers.

In Layer 1, inputs $(\mathrm{x}_i), \ldots, (\mathrm{x}_n)$ are fuzzified using trapezoidal MFs with adaptive parameters: (c_1), *(c_2)* , *(c_3)* , *(c_4)* . The trapezoidal function was chosen because of its flexible representation compared to triangular functions: it allows for a flat plateau in the middle, which can better capture certain types of fuzzy relationships [18].

Layer 2 evaluates the rule strength:

$$w_i = \prod_{j=1}^{n} \mu_i(x_j) \tag{4}$$

Layer 3 normalizes the strengths of all rules:

$$\bar{w}_i = \frac{w_i}{\sum_i w_i} \tag{5}$$

Layer 4 applies the rule R_i to obtain the output f_i

Layer 5 computes the global model response:

$$f = \sum_i \bar{w}_i \cdot f_i \tag{6}$$

The input of this step is output data from the previous step – initial pre-processing data. Real data were uploaded to the ANFIS-based system. The impact of the HF on the SDP can be represented as:

$$\mathrm{HF}(f_1, f_2, \ldots, f_n)$$

where f_i represents a specific attribute of a resource. In this study, we highlight key resource attributes such as motivation (*f1*), experience (*f2*), and availability (*f3*). Consequently, a linguistic variable within the HF encompasses linguistic attributes f_i – such as motivation (*f1*), experience (*f2*), and availability (*f3*) – whose value set T can be further segmented into meaningful crisp intervals. The linguistic meaning of these attributes corresponds to human linguistic attribute levels, categorized as *Excellent* (5), *Good* (4), *Moderate* (3), *Low* (2), or *None* (1). The outputs of this step are a set of functions that show the dependencies of each HF on deviation from estimated time.

4.3 CMMN SDP modelling

Case Management Model and Notation (CMMN) is a standardized graphical notation developed by the Object Management Group (OMG) to represent case management procedures, enhancing understanding and communication of complex, flexible processes. It visually captures the flow of activities, dependencies, and decision points, making it particularly useful in nonlinear and adaptive industries. CMMN facilitates stakeholder collaboration by providing a clear method for representing procedures. Inputs for this process include elements from Agile and Waterfall methodologies, while the output consists of CMMN-based Agile and Waterfall SDP models.

4.4 The simulation of the SDP model

Agent-based modeling and simulations (ABMS) are highly effective for replicating dynamic processes involving human actors by mimicking human behavior and accounting for psychological factors, workflow sequencing, and evolving conditions that influence BP outcomes. Often called multi-agent modeling, this approach conceptualizes decision-makers as agents whose activities are simulated within a framework of predefined rules. ABMS is particularly suitable for representing phenomena that emerge from multi-agent systems, making it applicable across various fields, such as traffic detection. Creating an agent-based model involves three key stages: 1) identifying and creating agents, 2) defining their interactions with one another and the environment, and 3) specifying the simulation environment [19]. When applied to simulating the SDP and illustrating its dynamic execution through CMMN, ABMS effectively captures the behavior of case executors and SDP roles throughout the implementation stages. The simulation inputs include ANFIS-generated data, CMMN SDP models, and input data, while the outputs are the simulation model and results based on different CMMN SDP models.

5 A Case Study

5.1 Experiment Data

Data for the experiment was collected from various IT organizations of differing sizes and complexities, ranging from small private organizations to a large international corporation, ensuring an objective study independent of the organization type. The main criteria for data collection included estimated and actual task execution times for each employee (see Table 2) and expert evaluations based on three human factors: motivation, experience, and availability (see Table 3), allowing us to calculate the average task deviation for each employee.

Table 2: Snapshot of collected data

Task No.	Name of task	Actual	Estimated	Deviation
X1	Y1	182	258	71%
X2	Y2	5.5	10	55%
X3	Y3	6.5	6.5	100%
X4	Y4	95	136	70%
X5	Y5	10	14.5	69%
Average deviation of tasks for employees				95%

Deviation was calculated as the ratio of actual to estimated time (see Table 2). If this ratio exceeds 100%, then actual time exceeded estimated time; on the contrary, if it is less than 100%, then actual time was lower than estimated time. Experts and stakeholders of every IT project under investigation were asked to evaluate their employees based on the HFs of motivation, experience, and availability by assigning a score from 1 to 5 as follows: *Excellent* (5), *Good* (4), *Moderate* (3), *Low* (2), or *None* (1). Experts were chosen by criteria that it would the most suitable and related person to project team. The final column in Table 3 presents the average deviation of tasks for each employee from Table 2.

Table 3: Snapshot of summarized data for all employees

Employee	Motivation	Experience	Availability	Average deviation of tasks for employees
Z1	2	2	5	298%
Z2	2	4	5	125%
Z3	4	3	4	217%
Z4	3	5	1	98%
Z5	4	4	2	195%

The analyzed HFs were assessed to ensure that the experimental data did not correlate (Table 4).

Table 4: Correlation of input data

	Motivation	Experience	Availability
Motivation	1	064688592	-0.12146834
Experience	064688592	1	-0.38768911
Availability	-0.12147	-0.38768911	1

In order to process the obtained data using ANFIS, the percentage values of deviation were transformed to metrics from 1 to 5 (i.e., *None* (5), *Low* (4), *Moderate* (3), *High* (2), or *Very high* (1)) by applying the rules in Table 5.

Table 5: Rules of splitting the data

Average deviation of tasks for employees	Metrics for ANFIS
$(+\infty; 200)$	1
[200; 155)	2
[155; 125)	3
[125; 100)	4
$[100; -\infty)$	5

A final ANFIS data snapshot is presented in Table 6. All data for ANFIS were concluded from 31 rows.

Table 6: Final data snapshot for ANFIS

Motivation	Experience	Availability	Deviation
2	2	5	1
2	4	5	3
4	3	4	1
4	5	1	4
4	4	2	2

5.2 Modelling the SDP with CMMN

The development of the CMMN model for simulating the SDP aimed to represent all pertinent tasks and allocate them based on the respective stages of the process.

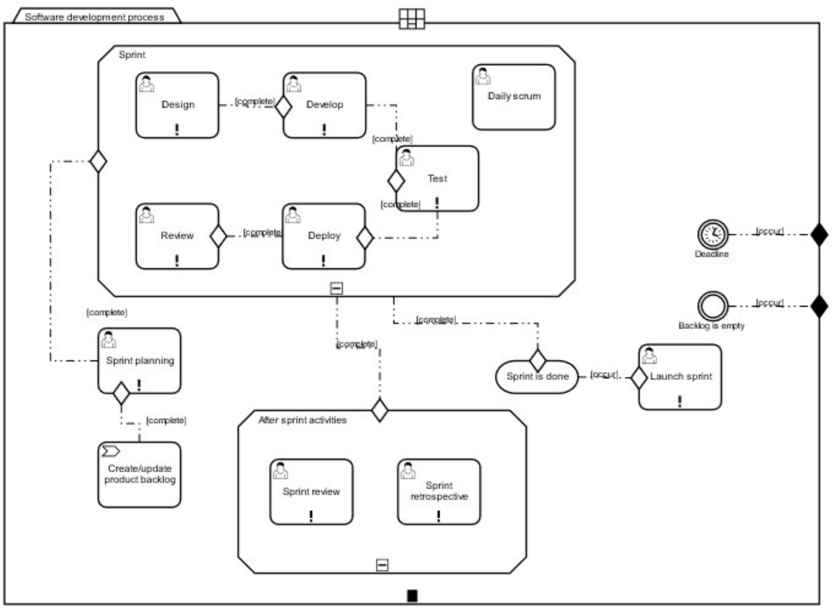

Figure 4: The architecture of a simulation model for an Agile SDP [29].

Figure 4 presents the classical Agile SDP model in CMMN, which is validated with Flowable. The main idea of this model was to represent the activities of the Agile model and their dynamical iterative execution.

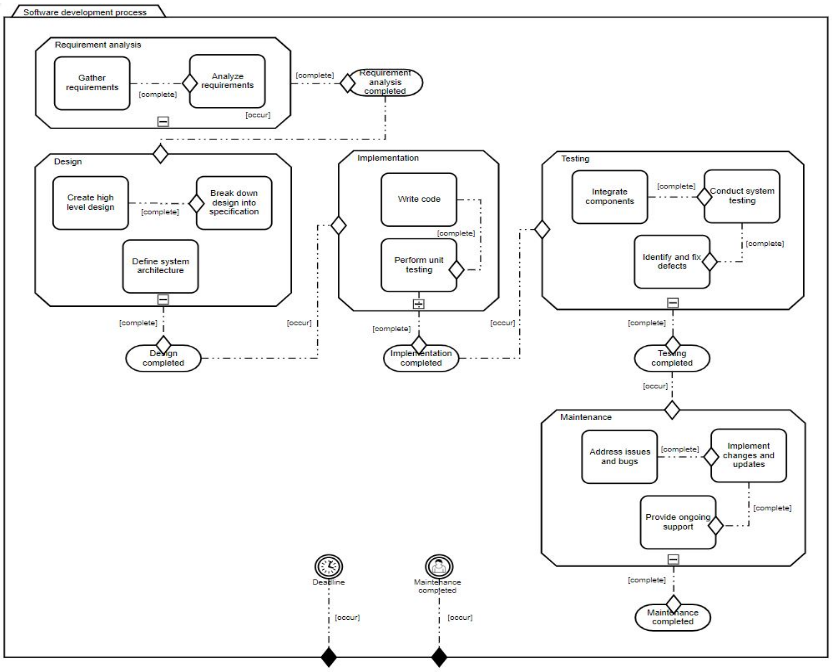

Figure 5: The architecture of a simulation model for a Waterfall SDP.

Figure 5 presents the classical Waterfall SDP model in CMMN, which is validated with Flowable. The main idea of this model was to represent the activities of the Waterfall model and their sequential execution.

5.3 ANFIS in SDP Simulation

HF data was loaded into ANFIS and FIS optimization was then performed. We generated a dataset from real SDP data and assigned possible deviations from the estimated task execution time value to different combinations of HF parameters. Possible deviation values were proportionally assigned in terms of the ratio of estimated to actual time performance. Depending on these values, we added explanations to possible numerical deviations (Table 7).

Table 7: Explanations of possible numerical deviations

Numeric value	Possible deviation value	Explanation
[0; 1.8)	Very high	210% above estimated time
[1.8; 2.8)	High	74% above estimated time
[2.8; 3.8)	Moderate	39% above estimated time
[3.8; 5)	Low	14% above estimated time
5	None	10% below estimated time

Table 8: Hyper parameters for ANFIS

Hyper-Parameters	Description/Value
Fuzzy structure/FIS training data	Sugeno/genfis1
Generation of FIS object	grid partition on the data
MF type (Input/Output)	Trimf/linear
Number of variables (inputs/outputs)	3/1
Optimization method	backpropagation
Maximum number of training epochs	200
Initial step size	0.001
AndMethod/OrMethod/ImpMethod/ AggMethod/DefuzzMethod	prod/probor/prod/sum/wtaver
Data for training/Data for testing	80/20
Number of rules	27

Layer 1, which includes motivation, experience, and availability, serves as the source for fuzzification and membership functions (MFs), with each MF represented as an adaptable node. In Layer

2, we established fuzzy rules by combining all MF pairs, resulting in 27 pairs from the three inputs. Layer 3 incorporated rule strengths as non-adaptive nodes, while Layer 4 focused on defuzzification by applying fuzzy rules through each node. The summation layer (Layer 5) consolidated these into a single, fixed output for the ANFIS network. The Fuzzy Inference Controller integrates the 27 fuzzy rules and MF parameters, refining them using precise data. After optimization, the improved rules and parameters were reintegrated, enabling inference with the optimized values. Following this process, the value of HF in the SDP shows variability after optimization, potentially falling within different linguistic term intervals. Notably, the likelihood of deviation is moderate before optimization and increases afterward, as illustrated in Figure 6.

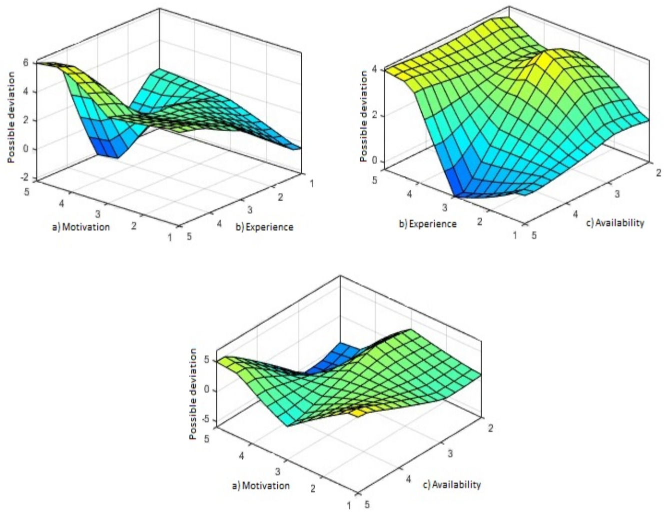

Figure 6: The resulting surfaces of optimized HFs in the SDP – a prediction based on different inputs.

After examining the generated schemas (Figure 6), it becomes apparent that the impact of various HFs on the system is not uniform. Among these factors, experience stands out as particularly influential, exhibiting a heightened sensitivity towards potential deviations. This suggests that individuals' levels of experience play a pivotal role in determining the likelihood and severity of deviations within the system. Conversely, while the motivation factor is also a significant determinant, its effect appears to be comparatively less pronounced. This implies that individuals' motivational states may indeed influence potential deviations, but to a lesser extent than their level of experience. Understanding these nuanced relationships between HFs and deviations is essential for devising effective strategies to mitigate risks and enhance system reliability.

Within this investigation, the Fuzzy Inference System (FIS) employed three input parameters, each characterized by three fuzzy sets. This configuration resulted in a comprehensive rule set comprising 27 rules, forming the basis for constructing the ANFIS model. Some examples of the fuzzy rules are presented in Table 9.

Table 9: Snapshot of fuzzy rules

IF HF1	AND HF2	AND HF3	THEN PD (possible deviation)
Excellent	Good	Good	None
Good	Low	Excellent	Moderate
Good	None	Low	Very high
Low	None	Excellent	Very high
Moderate	Moderate	Good	Low

It is crucial to highlight that the exponential expansion of the partition size within the input space correlates directly with a substantial increase in the number of influential rules. This expansion significantly enhances the speed of system learning and application.

5.4 SDP Simulation Results

The simulation is performed as described in the model (see Figures 4 and 5). In each iteration, agents are assigned to the relevant SDP tasks that are currently available to them. If no active tasks are identified for a particular agent during the iteration, a message is displayed, and the agent remains idle until the next iteration. Throughout the simulation, the event log entries provide a detailed record for each iteration. This log allows the following to be monitored: the number and nature of tasks identified for a specific agent in each iteration, along with their assigned roles; the initiation of tasks; and the identification of agents that did not encounter any active tasks in the iteration.

The simulation's output data is stored in electronic spreadsheet format and includes information such as the number of iterations, creation time, priority, task details, assigned agent, estimated time, actual time, and deviation. The quantity of stored output data is directly influenced by the input data. The output data from the simulation reveals the number of iterations, specifies which agent performed each task, indicates when the task commenced, provides the estimated time for the task, and records the actual time taken to complete the task. Deviation shows us how the HF of each agent impacts task execution time.

Different simulations were conducted with identical HF calculations and historical project input data according to the Agile and Waterfall SDP models.

Table 10: Results of Agile simulation

Task	**Agent**	**Estimated**	**Actual**	**Deviation**
Create product backlog	Agent 2	2	4.2	0.95
Launch sprint	Agent 5	1	1.74	3.09
Sprint planning	Agent 4	2	2.28	4.33
Deploy	Agent 5	2	3.48	3.09
Design	Agent 1	4	6.96	2.01
Develop	Agent 5	10	17.4	3.09
Review	Agent 5	1	1,74	3.09
Test	Agent 3	3	2.78	6.24
Daily scrum	Agent 5	1	1.74	0.95
Sprint retrospective	Agent 4	2	2.28	4.33
Launch sprint	Agent 5	1	1.74	3.09
Sprint review	Agent 4	2	2.28	4.33
		Amount: 31	**Amount: 48.62**	
		Deviation	157%	

Table 11: Results of Waterfall simulation

Task	**Agent**	**Estimated**	**Actual**	**Deviation**
Gather requirements	Agent 5	5	6.95	3.09
Analyze requirements	Agent 1	20	34.8	2.01
Define system architecture	Agent 2	15	31.5	0.95
Create high level design	Agent 2	10	21	0.95
Break down design into specification	Agent 2	10	21	0.95
Perform unit testing	Agent 4	3	3.42	4.33
Write code	Agent 4	35	39.9	4.33
Integrate components	Agent 4	10	11.4	4.33
Conduct system testing	Agent 3	10	9	6.24
Identify and fix defects	Agent 3	10	9	6.24
Provide ongoing support	Agent 4	30	34.2	4.33
Address issues and bugs	Agent 1	5	8.7	2.01
Implement changes and updates	Agent 4	20	22.8	4.33
		Amount: 183	**Amount: 253.67**	
		Deviation	139%	

As simulations were performed with the same HF input data, the results of deviation from estimated to actual time according to the selected methodology show the impact of the HF. We calculated average deviation by summing estimated and actual times for all tasks and dividing this number by the sum of the estimated time of all tasks to produce a deviation indicator. The results show that the Agile SDP is more affected by the HF because it has a higher deviation indicator on average, with the same resources. Repeated iterations could affect actual execution times and be more strongly influenced by the HF.

6 Discussion

The models presented in this study were created based on their classical concepts. However, it should be noted that in practice, these models may be different, i.e., hybrid. The goal of this study was to apply the proposed methodology to classical SDP models. Nevertheless, when necessary, these models can be modified according to the requirements.

In previous work [29], we simulated Agile SDPs with randomly generated HF data. This demonstrated that the smaller the overall HF indicator, the greater the deviation from the original task evaluation time. Simulations with real data showed us that each HF impacts differently, and this allows us to predict SDP results more accurately.

In addition to motivation, experience, and availability, several other variables could be significant in enhancing the predictive power of future models. For example, factors such as individual cognitive ability, team dynamics, project complexity, and the presence of external dependencies have the potential to influence the accuracy of task estimation. Cognitive ability may influence the manner in which employees process information and make judgments, while team dynamics, including collaboration and communication, may impact overall productivity. The number of interrelated components and the degree of external dependency, including the reliance on third party systems or vendors, may introduce variability in project timelines. The incorporation of these variables into future models would facilitate a more comprehensive understanding of the factors influencing estimation and performance.

The experiment simulated task execution in an agent-based system, assigning tasks to agents based on availability, with idle agents waiting until the next cycle. Event logs captured task assignments, initiations, and idle instances, while output data tracked iterations, task details, assigned agents, estimated vs. actual times, and deviations, revealing human factor impacts on task execution. Simulations for Agile and Waterfall projects, using identical inputs and human factors, highlighted that Agile tasks showed higher deviations from estimates, suggesting Agile projects are more influenced by human factors due to their iterative, dynamic nature.

Overall, this experiment underscores the importance of considering the HF in the SDP, particularly in Agile environments where iterative processes may amplify the effects of human variability on task execution times. These findings contribute to a deeper understanding of how the HF influences SDP dynamics and can inform strategies for mitigating the impact of the HF to improve SDP performance.

The main novelty of this article is the proposed hybrid case-based approach with the application of CMMN SDP modeling and simulation, enriched with the HF modeling component using ANFIS. Additionally, historical data from real SDP projects are used to verify the proposed method, allowing for the examination of how the execution of the process changes, considering the HF.

This research has some limitations and complexities. The main drawback is that the simulation results are based on historical SDP data, with few IT projects. The real historical data used in this study was gathered from potential biases related to the specific organizations or employee roles involved. Nevertheless, for the future works, to address this limitation, we intend to extend the dataset to include a broader range of IT companies/projects and employee profiles, thereby facilitating the development of a more accurate and robust data model. The objective of this expansion is to reduce bias and enhance the statistical significance of future findings. Moreover, this could ensure the reliable verification of the proposed fuzzy and case-based dynamic SDP modeling and simulation approach.

7 Conclusions

An analysis of related works on HF in SDP confirms the importance of studying HF impacts, with authors highlighting key factors like motivation, collaboration, experience, education, communication, and commitment. Accordingly, this research focuses on motivation, experience, and availability as core human impact variables.

The newly proposed fuzzy and case-based SDP modelling and simulation approach allows us to investigate the impact of the HF on the SDP, since its main components are real data pre-processing and the development of both CMMN-based models and a simulation model. The advantages of this are numerous.

First, the newly proposed approach uses real historical project data to predict the impact of the HF on the SDP, producing realistic results. Second, the proposed approach was applied to two different SDP methodologies (Agile and Waterfall), allowing us to understand how the sequence of tasks could impact SDP performance. In summary, the proposed approach allowed us to conduct a deeper investigation and achieve a greater understanding of the impact of HFs on the SDP using different software development methodologies. It also allowed us to assess how the impact of the HF varies when changing the sequencing of SDP activities. The proposed fuzzy and case-based SDP modelling and simulation approach was implemented into a prototype, and experiments were conducted in two ways: with Agile and Waterfall models. The findings indicate that integrating the HF into the SDP simulation model enables us to achieve greater accuracy in predicting actual task execution time and identifying potential SDP risks. The results also show that the Agile methodology is more sensitive to the impact of the HF, while proving that the task execution sequence in the SDP has a meaningful impact on SDP results.

The findings of this paper, though focused on software development, have implications for industries like marketing, product development, and construction, where Agile and Waterfall methodologies are also applied. These methods offer adaptable (Agile) or structured (Waterfall) approaches that can be modified to suit specific industry needs. Agile may be more suitable for customer-driven fields like marketing, while Waterfall could better fit high-risk, controlled environments like construction. However, further research is necessary to adapt these methodologies to the unique challenges of each sector.

Consequently, the following future research directions are outlined: 1) Extend and classify data from historical projects to produce more accurate results highlighting the impact of the HF and enabling us to predict possible risks to different kinds of projects and organizations, 2) Create a CMMN model that incorporates more varied SDP methodologies and predicts more risks for each specific methodology.

Funding

This research received no external funding.

Author contributions

Conceptualization, Š.S. and D.K.; methodology, Š.S. and D.K.; software, Š.S.; validation, Š.S.; formal analysis, Š.S. and D.K.; investigation, Š.S. and D.K.; resources, Š.S. and D.K.; data curation, Š.S.; writing—original draft preparation, Š.S.; writing—review and editing, D.K.; visualization, Š.S.; supervision, D.K.; project administration, Š.S.; funding acquisition, D.K. All authors have read and agreed to the published version of the manuscript.

Conflict of interest

The authors declare no conflict of interest.

References

[1] Ambler, S. W., & Lines, M. (2012). Introduction to Disciplined Agile Delivery. IBM Press.

[2] Amrit, C., Daneva, M., & Damian, D. (2014). Human factors in software development: On its underlying theories and the value of learning from related disciplines. A guest editorial introduction to the special issue. *Information and software technology,* 56(12), 1537-1542.

[3] Andrei, B. A., Casu-Pop, A. C., Gheorghe, S. C., & Boiangiu, C. A. (2019). A study on using Waterfall and Agile methods in software project management. *Journal of Information Systems & Operations Management,* 125-135.

[4] Bai, Z., Li, Y., Wo´zniak, M., Zhou, M., & Li, D. DecomVQANet: Decomposing visual question answering deep network via tensor decomposition and regression. *Pattern Recognit.* 2021, 110, 107538.

[5] Boehm, B. W. (1988). A Spiral Model of Software Development and Enhancement. *ACM SIGSOFT Software Engineering Notes,* 11(4), 14-24.

[6] Capretz, L. F., & Ahmed, F. (2010). Making sense of software development and personality types. *IT professional,* 12(1), 6-13.

[7] Capretz, L. F. (2014). Bringing the human factor to software engineering. *IEEE software,* 31(2), 104-104.

[8] Choi, B., & Rhee, F. (2009). Interval type-2 fuzzy memberships function generation methods for pattern recognition. *Information Sciences,* 179(13), 2102–2122.

[9] Cockburn, A. (2002). Agile software development. Addison-Wesley.

[10] Dutra, E., Diirr, B., & Santos, G. (2021, September). Human factors and their influence on software development teams-a tertiary study. In Proceedings of the XXXV Brazilian Symposium on Software Engineering (pp. 442-451).

[11] Dyba, T., & Dingsøyr, T. (2008). Empirical studies of Agile software development: A systematic review. *Information and Software Technology,* 50(9-10), 833-859. doi:10.1016/j.infsof.2008.01.006.

[12] Earl, D.J., & Deem, M.W. Monte Carlo simulations. *Methods Mol. Biol.* 2008, 443, 25–36.

[13] Gilbert, N. (2019). Agent-based models. Sage Publications.

[14] Gonçalves, W. F., de Almeida, C. B., de Araújo, L. L., Ferraz, M. S., Xandú, R. B., & de Farias, I. (2017, June). The influence of human factors on the software testing process: The impact of these factors on the software testing process. In 2017 12th Iberian conference on information systems and technologies (CISTI) (pp. 1-6). IEEE.

[15] Guveyi, E., Aktas, M. S., & Kalipsiz, O. (2020). Human factor on software quality: a systematic literature review. In Computational Science and Its Applications–ICCSA 2020: 20th International Conference, Cagliari, Italy, July 1–4, 2020, Proceedings, Part IV 20 (pp. 918-930). Springer International Publishing.

[16] Highsmith, J. (2002). Agile software development ecosystems. Addison-Wesley.

[17] Jang, J. ANFIS: adaptive-network-based fuzzy inference system, 1993; Vol. 23, Issue 3. https://ieeexplore.ieee.org/abstract/document/256541/

[18] Kacker, R. N., & Lawrence, J. F. (2007). Trapezoidal and triangular distributions for Type B evaluation of standard uncertainty. *Metrologia,* 44(2), 117.

[19] Klügl, F., & Bazzan, A. Agent-based modeling and simulation. *AI Magazine,* 2012. https://ojs.aaai.org/index.php/aimagazine/article/view/2425

[20] Machuca-Villegas, L., Gasca-Hurtado, G. P., & Muñoz, M. (2021). Measures related to social and human factors that influence productivity in software development teams. *International Journal of Information Systems and Project Management,* 9(3), 43-67.

[21] Machuca-Villegas, L., Hurtado, G. G., Puente, S. M., & Tamayo, L. M. R. (2021). An Instrument for Measuring Perception about Social and Human Factors that Influence Software Development Productivity. *J. Univers. Comput. Sci.,* 27(2), 111-134.

[22] Manikavelan, D., & Ponnusamy, R. Software quality analysis based on cost and error using fuzzy combined COCOMO model. *J. Ambient. Intell. Humaniz. Comput.* 2020, 1–11.

[23] Marin, M.A., Lotriet, H., & Van Der Poll, J.A. Metrics for the Case Management Modeling and Notation (CMMN) Specification. In Proceedings of the 2015 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists, Stellenbosch, South Africa, 28–30 September 2015.

[24] Mao, J. Optimizing Enterprise Marketing Project Portfolios Using Fuzzy Ant Colony Optimization. *International Journal of Computers Communications & Control*, **2024**, *19*(3), 6458. Available online: https://doi.org/10.15837/ijccc.2024.3.6458.

[25] Nogueira, R.P.; Melicio, R.; Valério, D.; Santos, L.F. Learning methods and predictive modeling to identify failure by human factors in the aviation industry. *Appl. Sci.* 2023, 13, 4069.

[26] Pirzadeh, L. (2010). Human factors in software development: a systematic literature review.

[27] Royce, W. W. (1970). Managing the Development of Large Software Systems: Concepts and Techniques. In Proceedings of IEEE WESCON, 1-9.

[28] Ruiz, M., & Salanitri, D. (2019). Understanding how and when human factors are used in the software process: a text-mining based literature review. In Product-Focused Software Process Improvement: 20th International Conference, PROFES 2019, Barcelona, Spain, November 27–29, 2019, Proceedings 20 (pp. 694-708). Springer International Publishing.

[29] Sielskaitė, Š., & Kalibatienė, D. (2023). On Fuzzy and Case-Based Dynamic Software Development Process Modeling and Simulation Approach. *Applied Sciences,* 13(11), 6603.

[30] Schwaber, K. (2004). Agile Project Management with Scrum. Microsoft Press.

[31] Sommerville, I. (2011). Software Engineering (9th ed.). Addison-Wesley.

[32] Thesing, T., Feldmann, C., & Burchardt, M. (2021). Agile versus Waterfall project management: decision model for selecting the appropriate approach to a project. *Procedia Computer Science,* 181, 746-756.

[33] Urquhart, L.; Wodehouse, A.; Loudon, B.; Fingland, C. The application of generative algorithms in human-centered product development. *Appl. Sci.* 2022, 12, 3682.

[34] Wróbel, K.; Gil, M.; Chae, C.J. On the influence of human factors on safety of remotely-controlled merchant vessels. *Appl. Sci.* 2021, 11, 1145.

https://doi.org/10.15837/ijccc.2025.4.6807
Cite this paper as:

Sielskaitė, Š.; Kalibatienė, D. (2025). The Impact of Human Factors on Software Development Processes Applying the Agile and Waterfall Methodologies: A Case Study Using Real Data, *International Journal of Computers Communications & Control*, 20(4), 6807, 2025.
https://doi.org/10.15837/ijccc.2025.4.6807