Data Management Plan

Citizen Development: Potential challenges of applying low-code and using SCRUM.

Study program: Master Digital Business Administration

Course: DR 4 – Master Thesis

Supervisor: Prof. Dr. Nikolaus Obwegeser

Co-Supervisor: Prof. Dr. Sebastian Höhn

Author: Pascal Steiner

Basel, 13th of June, 2022

Bern University of Applied Sciences

1. Data Management Plan

This Data Management Plan describes the methodological approach to data collection and analysis of the master thesis "Citizen Development: Potential challenges of applying low-code and using SCRUM". The research question can be divided into the following two sub-questions:

1. What are the challenges related to low-code development by the end user?
2. To what extent is the SCRUM method suitable for structuring this type of software development?

2. Research design

Since very little research can be found on this topic at present, the focus of the work was to generate new results. For this purpose, an explorative approach was chosen and interviews with 10 persons were conducted. The semi-structured interviews allowed, depending on the course of the conversation and the knowledge of the interview partner. The problem-centered interview according to Witzel was applied. (Witzel & Reiter, 2012) and then evaluated based on the qualitative content analysis according to Kuckartz (2012).

2.1 Data Collection

The interviews took place between April and May 2022 and lasted an average of 33 minutes. The longest interview lasted 1 hour and 8 minutes and the shortest 16 minutes. Prior to the start of data collection, the research was registered on Aspedictet. Beforehand, a test interview was conducted to check the technical infrastructure, length and type of introduction, and sound quality. In connection with contacting and scheduling the interview, interview participants received an initial written explanation of the research question and an estimate of the time required to complete the interview. The initial assumption was between 30 and 60 minutes. All interviews took place via Microsoft's Teams video telephony application in digital format. For all recordings, the consent of the interview participants is available.

2.2 Interview guide

The interview guide contains an introduction in which the research question and organizational aspects were explained. At the beginning of the interview and for the elicitation of individual characteristics, the current job title and previous experience in low-code software development were asked (see Table 1).

The semi-structured procedure according to Witzel (Witzel & Reiter, 2012) made it possible that the interviewees could speak freely and at the same time were guided towards the problem by the basic structure in the form of core questions.

The content of the questions was based on the four overarching thematic clusters:

- Differentiation between Citizen Development and classical software development
- Challenges of Citizen Development
- General assessment of SCRUM
- Assessment of the compatibility of SCRUM and Citizen Development

2.3 Formation of the sample

A total of 18 potential interview partners were requested, ten of whom agreed to participate. The decisive factor in determining the participating interview partners was that they already had experience in the area of low-code software development. By means of a keyword search for "low-code", it was possible to identify people on the LinkedIn platform who had already engaged in the topic. Four people responded positively to the direct request for an anonymized interview. From these interviews, six additional interviewees emerged through referrals from other interviewees (see. Table 1). The test interview (Interview 1) was also transcribed, but was not used for further analysis.

Table 1: list of interview partners

Interview	Professional Background	Experience with low-code development
2	Expert Software Engineer	3 Years
3	Solution Sales Manager	8 Years
4	Business Analyst	10 Years
5	Microsoft 365 Architect	7 Years
6	Head Application Plattform	2 Years
7	Business Engineer	3 Years
8	Computer Scientist	2 Years
9	Digital Coach	7 Years
10	Product Owner	20 Years
11	Solution Advisor	1 Year

Note: Testing interview (1) is not included in this listing because it was not used for further analysis.

2.4 Reflexion of the data collection

In retrospect, the data collection in the form of expert interviews can be considered successful. The guidelines developed in advance were effective in guiding the interviewees to the research question. The results can therefore be considered as subjective perceptions and ways of processing of the persons interviewed (Witzel & Reiter, 2012). Only one interview showed a strong deviation in duration. Since the interviewee only gave very short answers, the interview with a duration of 17 minutes was short compared to the rest of the interviews. The longest interview lasted over an hour. Since all of the interviews conducted provided insights, all 10 interviews could be considered in the evaluation.

3. Data analysis

3.1 Analysis of data material

Transcripts were created based on the recorded interviews. To ensure anonymization, names and company name of the employer were anonymized with and replaced with the generalizing words "interviewee" and "company".

1. What are the challenges related to low-code development by the end user?
2. To what extent is the SCRUM method suitable for structuring this type of software development?

3.2 Qualitative content analysis according to Kuckartz

In the context of Kuckartz's qualitative content analysis (Kuckartz, 2012), information was filtered from the text and deductive determinate categories were formed. The technique is well described and it allows for iterations of deductive and inductive category and code formation to remain open to unanticipated information. Initially, the focus of category formation was on identifying advantages, disadvantages, and recommendations for the Citizen Development and SCRUM dimensions. As a result of the realization that the statements in these dimensions often could not be clearly evaluated, it was decided in a further version to omit this dimension of the evaluation. This was done in favor of a clearer structure consisting of Citizen Development and SCRUM. The code formation was then based on an inductive procedure, so that codes and sub-codes could be created for each category. In a next step, sub-codes that had too little data material or correlated too strongly with others in terms of content were combined or redefined.

3.3 Category and code system as a basis for analysis

In the following, the deductively derived main categories are presented. Following this, the codes and sub-codes were developed inductively over several iterations, described and underlaid with anchor examples. The detailed description on the level of codes and sub-codes can be found in the separate code book.

Table 2: Overview of the main categories

Main category	Definition
Citizen Development	These include statements that can be related to the use of low-code technology by the end user in an organization.
SCRUM	The category includes all statements that relate to the use of SCRUM or assessments of SCRUM

Note: table 3 contains the classification and description of the categories

Limit of the method

In forming the categories, codes and sub-codes, it was challenging to make the delimitations so precisely that there would be no overlaps. As described in Chapter 3.2, the original approach of classifying the advantages, disadvantages and recommendations was therefore discarded in favor of a thematic classification.

References

Kuckartz, U. (2012). *Qualitative Inhaltsanalyse. Methoden, Praxis, Computerunterstützung*.

Witzel, A., & Reiter, H. (2012). *The problem-centred Interview*.

Table 3: Main-Categories, Codes and Sub-Codes

Main Category	Sub-Category	Codes	Sub-Codes
Citizen Development	Citizen Developer	Willingness to learn and motivation Need for affinity and existing knowledge Missing knowledge Time investment	
	Enablement	Learning culture Empowerment Team composition Training User exchange	
	Coordinator	Center of competence Internal contact Overarching strategy Lack of efficiency Dependency on IT	
	Classical IT Topics	Application field Conscious limitation Criticality of the applications Documentation Governance Interfaces Life-cycle management Ownership Prerequisites Testing	
SCRUM	Impact of SCRUM	Application of SCRUM SCRUM effort SCRUM risk management SCRUM standardization and quality	
	Impact on SCRUM	SCRUM roles	Business analyst Product owner SCRUM master Software developer
		SCRUM events	SCRUM daily SCRUM review SCRUM retro
		SCRUM artifacts	SCRUM backlog SCRUM estimations SCRUM sprint

Note: table 3 contains the classification into categories, codes and subcodes. The detailed code descriptions are shown in the separate codebook.