e-Informatica Software Engineering Journal, Volume 18, Issue 1, 2024, pages: 240105, DOI: 10.37190/e-Inf240105

Measuring End-user Developers' Episodic Experience of a Low-code Development Platform

A Preliminary Study

Dongmei Gao*, Fabian Fagerholm*
*Department of Computer Science, Aalto University
dongmei.gao@aalto.fi, fabian.fagerholm@aalto.fi

Abstract

Background: As low-code development platforms (LCDPs) are becoming a trend, understanding how end-user developers think and feel as they work with such platforms is important. Particularly, assessing experiences during episodes of use can contribute to overall experience throughout long-term use.
Aim: This paper aims to understand end-user developers' episodic experience when they are building an application on a low-code platform and to provide guidance on how such experiences can be measured.
Method: We designed the Episodic Developer Experience Questionnaire for LCDPs based on prior literature and refined it through expert Delphi sessions. The instrument contains 10 individual experience items, capturing various aspects of episodic experience. We further validated it through remote online tests on an LCDP.
Results: The results showed significant differences in the relationships between items describing aspects of overall experience and items describing perceptions of tool quality and task difficulty. Programming expertise also affected end-user developers' episodic experience.
Conclusion: The study illustrates the design of questionnaire-based experience assessment in the context of development and identifies the importance of separating personal experience from assessment of tasks and tools since tool quality and task difficulty do not necessarily influence experience straightforwardly.

Keywords: developer experience, episodic experience, low-code development platforms, experience measurement, software engineering, human-computer interaction

1. Introduction

Developer experience (DX) refers to the subjective experiences arising from the involvement in software development [1, 2]. Understanding and improving DX can contribute to better support and increased well-being for software developers and may ultimately lead to higher levels of creativity and productivity [3, 4]. Developer motivation and happiness – factors of a positive experience – are also regarded to be correlated with the success of a project [5, 6].

Software developers have often been understood as having expert programming knowledge, while end-user developers [7] may have different kinds of domain knowledge but

Submitted: 15 Jun. 2023; Revised: 15 Dec. 2023; Accepted: 5 Feb. 2024; Available online: 1 Mar. 2024

are not specialised in software development. End-user developers therefore need software development tools that allow them to generate software programs or applications using domain concepts without needing to delve into the details of the implementation. Recently, Low-Code Development Platforms (LCDPs) have emerged in the industry to enable domain experts without programming knowledge to build applications [8, 9]. LCDPs offer a variety of ready-made components and features that allow making applications easily, usually by manipulating visual objects. Because they allow rapid expression of ideas, even developers experienced with textual programming languages may prefer to use LCDPs to save effort and time for certain kinds of tasks. Given the diverse levels of knowledge among developers, some LCDPs are designed to support both inexperienced and expert developers. They aim to enable developers of all levels to achieve their goals and to give them a positive developer experience.

LCDPs are still new and it is unclear how widespread their use will become. Some recent studies have discussed the conceptual understanding of LCDPs [10, 11], their characteristics [8], and challenges and opportunities associated with their use [12]. To understand how LCDPs could fit into the tool landscape of developers at different skill levels, it is important to understand how they affect DX.

DX can be observed on different time scales, varying from a developer's earliest awareness of a tool to use over a long period of time [2]. In this study, we focus on episodic experience to understand how developers feel after a single session of LCDP use where they focus on a bounded task, such as developing an individual application feature, in a given time. Their experience may change over time as they use an interactive product. They may be excited to use a new product for the first time, or attracted by its appearance, then become anxious by the complexity of the interaction during use, or satisfied because of the intuitiveness and efficiency of the functionality, and finally develop a complex experience about the product after a longer period of use.

Measuring and tracking episodic experience can yield insights into how the long-term experience has formed. It can also inform iterative design improvements to the platform. Our study design works by eliciting episodic experiences through naturally-bounded sessions of work that can be completed within a given period. Particularly, each session has a small set of goals and utilises a particular set of tool features, which we believe is an important unit of observation to inform the understanding of DX and assist in improving the design of LCDPs and other development tools.

To facilitate the comparison of experiences of LCDP use, we propose EDEQ-LCDP, a questionnaire instrument that captures important dimensions of developers' episodic experience. The intent is to provide an instrument that can be used rapidly after a development session to give an indication of the contents and quality of the developer's experience. The development of the questionnaire also illustrates how measuring DX is different from measuring UX in two senses: in terms of what experiential content or characteristics are relevant to the context of development rather than the context of use, and in terms of the object of assessment, which can vary between the experience itself and an assessment of a tool or product. We seek to disentangle the experience from assessments of the tool and the task and discuss the challenges of doing so.

The aim of the present study is therefore to provide an understanding of end-user developers' episodic experience while performing application-building tasks on an LCDP so that the development of measurement instruments for episodic developer experience can be better guided. We formulated the following research question:

RQ: How can end-user developers' episodic experience of low-code development platforms be measured?

The paper also contributes to the EDEQ-LCDP instrument as a basis for episodic developer experience measurement. To our knowledge, the instrument proposed in this paper is the first to focus specifically on the episodic experience of LCDPs. We take some steps towards examining the construct validity of the questionnaire as well as its content validity for practical and conceptual purposes, while properties of statistical generalisability are of secondary importance. While our study is performed using a particular LCDP, the instrument is not tied to any of its special features, and could, bearing limitations in mind, be used in tasks with other LCDPs as well.

2. Related work

DX has been defined as "a means for capturing how developers think and feel about their activities within their working environments" [1]. Fagerholm and Münch [1] proposed a conceptual framework for developer experience with three main dimensions – cognition (perception of development infrastructure), affect (emotional state about work) and conation (the value of one's own contribution). An extended discussion of the three dimensions was provided in a study of measuring DX toward the use of a Deep Learning (DL) platform [13]. Here, "cognition" is the rational basis provided by the DL platform, "affection" is explained as the emotional state, and "conation" is about the developer's tendency to use the DL platform voluntarily.

Transferring from the concept of user experience (UX), Fagerholm and Münch [1] conceptualised DX as experiences arising from the context of development, with impacts for individual developers and teams of developers, but also for organisational and product outcomes such as process adherence, productivity, and the quality of the end product. Thus, DX is a concern for developers themselves and for the organisations they work in and has implications for the customers and end-users of the software they produce.

Time seems to have an impact on the importance people attribute to different qualities of the experience with interactive products [14]. *Anticipated experience* refers to evaluations that occur before use, *immediate* or *momentary experience* is related to in-the-moment impressions, *episodic experience* emerges during a specific time-bounded duration, and *accumulated experience* (also known as cumulative experience) happens after long-term usage. Fagerholm's theoretical framework also categorises developer experience into immediate, episodic, and cumulative experience [2].

Episodic experience has drawn attention in various contexts: for example, research on learning to become a coach has shown that it can affect the way that individuals perceive what they know, and have the potential to influence their perception of future learning [15]. Marti and Iacono [16] recorded users' experiences during four weeks of using a product, showing that their episodic experience changed over time. Despite the crucial importance of usability in the product's initial acceptance, aspects of reliability, motivation, comparison with other products, change in behaviour and touch points – how the product communicates with the user, for example by notifications and alerts – are even more crucial for a user to resonate with a product and value it in the long term.

Greiler et al. [17] proposed an actionable conceptual framework to better understand and improve DX. They suggest putting the framework to work using an Ask-Plan-Act process: make problems visible by qualitative or quantitative means, determine the area to

be improved, and make continuous and small improvements. A survey instrument could be utilised in such an improvement cycle.

To facilitate the development of a DX instrument for the LCDP context, we should first understand more about the factors influencing DX, learn more about the characteristics of end-user developers as a particular group of LCDP users, and finally describe current research on LCDPs themselves. We discuss these topics in the following sub-sections.

2.1. Factors influencing developer experience

While some attempts have been made to construct survey instruments to assess developer experience (e.g., the DEXI scale [18]), there is currently a lack of instruments designed to measure DX in specific settings and for particular categories of experience. Previous studies have aimed to identify metrics or factors that impact developer experience. Kuusinen [19] conducted a survey considering a particular cross-platform IDE and found characteristics such as efficiency of use, flexibility, informativeness, intuitiveness, and reliability to be appreciated by developers. While developers in this study concentrated mainly on the pragmatic qualities of the tool, they also related experiences of hedonic aspects, such as affection for the tool, and expressions of feeling at home with it and that it has a good atmosphere.

The importance of developers' motivation for work has been widely emphasised in software engineering [20]. Intrinsic factors, such as autonomy and sense of achievement, have been found to be essential motivators of software developers [5]. Graziotin et al. [21] investigated factors associated with developer unhappiness, finding several factors to negatively influence happiness, such as being stuck in problem-solving, time pressure, bad code quality and coding practices, mundane or repetitive tasks, imposed limitations on development, and personal issues. Storey et al. [4] found that developers' job satisfaction can in turn impact their perceived productivity. To improve developer satisfaction, they identified several factors: how managers manage, and whether developers can effectively use their skills and believe their work has an impact. Linberg [22] found a large gap between developers' definition of job success and traditional definitions. Developers might remain satisfied with their jobs even if the project seems like a failure by others. Motivation is thus a complex dimension of experience that may be very personal and not only be driven by external outcomes.

Previous research has presented numerous factors to describe developer experience in different contexts. Bobkowska [23] built a model to explain the intuitiveness of software engineering techniques using UX concepts. It includes four perspectives: cognitive processes (users' familiarity with the technique), motivations (using the technique to solve a problem), actions (actions lead to good or poor results) and emotions (positive or negative attitudes). The model attempts to explain how episodic experiences with the techniques build up into cumulative experience and a perception of intuitiveness. Here, the episodic experiences are related to specific actions taken within the frame of a particular technique, and those experiences then accumulate and are generalised into cumulative experiences. The latter may then include the notion of intuitiveness regarding the technique. However, intuitiveness may also be the starting point: early experiences influence motivation to learn, they influence affective attitudes towards the technique, and they moderate the results of learning and actions, which then set the direction for the development of further experiences.

Despite the many existing studies on cognitive (cf. [24]) and affective (cf. [25]) aspects involved in software development, how to measure episodic DX, such as what dimensions

to measure to capture the salient features of developers' experiences as compared to users' experiences, remains an open question.

2.2. End-user developers

End-user development (EUD) has been defined as "the set of methods, techniques, tools, and socio-technical environments that allow end users to act as professionals in those ICT-related domains in which they are not professionals, by creating, modifying, extending and testing digital artefacts without requiring knowledge in traditional software engineering techniques" [26]. In the context of EUD, the term end-user developer is also widely used to describe people who have neither the skills nor the interest to customise a system in the way that software professionals do [27].

Cotterman and Kumar [28] defined an end user as "any organizational unit or person who has an interaction with the computer-based information system as a consumer or producer/consumer of information". They mapped out three dimensions across which the role of users can vary: operation, development, and control. Operation refers to tasks and actions necessary for operating a computer system, development refers to tasks related to the development of such a system, and control refers to the decision-making authority to acquire, deploy, and use resources necessary for operation and development. The role of a user may then involve more or less of the activities across these dimensions, and according to this taxonomy, end users share some traits with persons whose primary role is to develop a system.

Today, the boundary between development and non-development tasks in the roles of employees who are not primarily hired as software developers can be considered somewhat fluid because of the flexible ways in which many software products can be customised or controlled programmatically after they have been deployed to users. In the case of LCDPs, this fluidity goes even further. End-users of LCDPs become end-user developers when they begin to develop their software applications using LCDPs, shifting their role across the development dimension of Cotterman and Kumar's taxonomy.

Most importantly, we assume that end-user developers range from professional developers to people who are not experienced with or specialised in conventional software development (so-called citizen developers) [29]. As far as we know, the term citizen developer first appeared alongside LCDP in a 2014 Forrester report [30]. Since then, the term has been widely embraced by LCDP vendors and organizations as their potential user groups [12]. The promise of LCDPs is that they allow citizen developers to build their software applications without the help of professional developers [11, 29] while professional developers can also benefit from LCDPs, for example by being more productive and reducing time spent on repetitive tasks. The low-code platforms on the market today are also increasingly focused on freeing people from heavy, monotonous underlying development tasks and focusing more on the realisation of ideas.

The two terms EUD and citizen developer thus have a significant overlap, with the latter originating in industry and focusing attention on the role of an employee in relation to the IT function of an organisation [29], and the former being a more long-standing and established term in research which we perceive as subsuming the other term. We have chosen to use the term EUD in this paper because of its longer history, broader and more elaborate definition, and because we do not investigate the aspects of citizen development that relate the citizen developer and the applications they produce to the larger IT infrastructure and governance in an organisation. We consider the end-user

developer as a person who has two roles, being an end-user of the product and a developer of an application at the same time, a definition that shares many important aspects of the citizen developer term and is consistent with terminology used in human-computer interaction and software engineering research (see, e.g., [26–29]).

2.3. Low-code development platforms (LCDPs)

In recent years, Low-Code Development Platforms (LCDPs) have emerged as a specific kind of EUP tool that reduces the effort of implementing simple applications. LCDPs can provide developers with the means to generate and deliver business applications quickly with minimum effort in terms of installation and configuration of the development environment, training, and implementation of the project [31]. They contain a set of components and features for programmers and non-programmers. LCDPs are primarily aimed at developers with a limited programming background [11]. These end-user developers rely on LCDPs to achieve their creative ideas without going through conventional development cycles to build applications from scratch as traditional developers do. In addition, faced with the ever-changing customer needs in the software business, firms often adopt LCDPs for fast delivery with a minimum of hand-coding, quick setup and deployment. They also address the ability of LCDPs to test business ideas within days or weeks [30]. Thus, professional developers (professionals with an education or career in software development) can benefit from LCDPs as well, for example by being more productive [29].

Waszkowski [31] pointed out three main features of LCDPs: databases, business processes and user interface. They simplify the development process, reduce the learning costs and visualise the coding environment to allow developers to spend less time on coding and focus on their goals. Although there is a varied range of different types of LCDPs, some typical functionalities and features can be identified [11]. Two major structures of LCDPs have been described: UI to Data and Data to UI [11], denoting the flow of design from either the user interface or the data model. Sahay et al. [11] list four LCDP characteristics that have a large impact on DX: 1) *interoperability*, by creating standards and a friendly ecosystem in a domain to mitigate issues caused by closed sources; 2) *extensibility*, allowing developers to customise capabilities to reduce the design constraints; 3) the *learning curve*, which is still less intuitive for end-user developers; and 4) *scalability*, such as running on the cloud to manage intensive computing [11].

The term no-code is closely related to low-code. It is sometimes used as a slight variation of low-code [10, 29] or a synonym to refer to low-code development practices [8]. As we can see in the market, LCDP vendors define their products as no-code or low-code tools according to their business goals. The distinction of them is still ambiguous and practitioners also use these two terms interchangeably. In this paper, we view the no-code platform as a subset of a low-code platform. End-user developers can use the basic functionality of the low-code platform without writing any code but are still required to write several lines of code to get access to more advanced features.

3. Research approach

In this paper, we present the development of a new questionnaire instrument that aims to identify important influencing factors of developers' episodic experience of LCDPs. The instrument we designed aims at end-user developers including both professional developers

and citizen developers. The purpose of developing the instrument is primarily to structure the phenomenon of developer experience in the context of LCDPs, to provide further understanding of it, and to provide guidance on how such experiences can be measured. To a lesser extent, we are also interested in the questionnaire instrument itself for purposes of measuring DX in an LCDP context, but we do not aim to fully validate the instrument for general use.

The process of developing the questionnaire and providing a preliminary validation of it involved three stages, as shown in Figure 1. First, we analysed related literature to collect potentially relevant question items and constructed a preliminary questionnaire based on these items (Stage 1). Second, we used a Delphi study [32] to gather opinions from a panel of experts, helping us to capture important nuances and refine the questionnaire (Stage 2). In the third stage, we ran a task-based test in which 19 developers were asked to use an LCDP in individual sessions (Stage 3). After each task, they filled in our questionnaire, providing us with data to further analyse it. The result of this research process is the final questionnaire proposed in this paper.

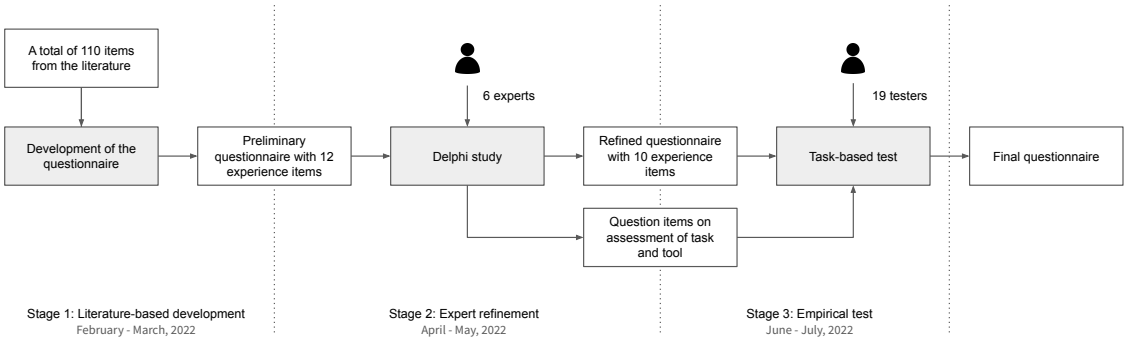

Figure 1. Overview of the research approach

The questionnaire aims to capture end-user developers' episodic experiences in relation to tasks performed using an LCDP. Assessing episodes with a limited number of tasks that involve known features enables assessment of specific parts of an LCDP, which may be helpful when using the questionnaire for design improvements in the tool.

We assume that the degree of familiarity with software development could influence individual developers' experience. Their prior familiarity can influence how much effort and time they need to invest in learning the platform. Therefore, the impact of prior programming background was taken into account in this study by introducing a set of background questions for the participants of the task-based test. Another factor that could influence a developer's experience is the complexity of the task and the environment in which it needs to be conducted. For example, a complex IT environment with many existing applications made using an LCDP presents a more challenging environment because the developer may have to consider existing software when they develop something new. However, in this study, we focus on the individual experience of an LCDP in a bounded development task. The questionnaire instrument does not assess the difficulty of the environment but aims to capture basic information about a developer's personal experience. To differentiate between the experience of the platform and the task, we included question items that asked for assessments of the tool and the task. We discuss the details and output of each stage in the following sections.

4. Literature-based development of a preliminary questionnaire

We developed an initial questionnaire by collecting experience items from existing studies with relevance to DX. Our starting point was a set of factors reported in a multi-vocal literature review of LCDP, which collected factors both from scientific and grey literature [33]. We also searched for articles which presented questionnaire instruments to measure DX or UX using Google Scholar and used items from those questionnaires as potential candidate items for our questionnaire.

4.1. Source literature

Nylund [33] presented factors that improve and worsen DX based on existing literature. The list of 17 DX influencers includes topics such as mood and feelings, project onboarding, and factors related to the software development methodology, such as Agile software development. This study facilitated our search for literature associated with DX by providing a systematically collected set of relevant literature. We expanded this set by a search for DX and UX measurement instruments. We first used Google Scholar with the keywords "developer experience measurement" to find additional candidate items. We also used the keywords "user experience measurement" to extend the search scope. We selected papers that presented questionnaire-based instruments that could potentially be relevant for measuring different aspects of developer experience.

We included items from the Developer Experience Scale (DEXI) [18], one of the first questionnaire instruments that aim to measure the DX of development tools. Its word pairs were obtained from several sources and selected based on their relevance to software development. It contains one item for measuring general quality, five items for pragmatic quality, and six items for hedonic quality. We also collected 20 items from a recent study conducted by Lee and Pan [13] which aimed to evaluate sub-constructs of DX – cognitive, emotional and behavioural – and which itself is based on items from various sources.

We found only a few instruments intended specifically for DX measurement. Because of the commonalities between UX and DX and the greater breadth and depth of UX research, we also considered other widely used UX surveys: the Short Dispositional Flow State Scale (SDFS-2) [34], Intrinsic Motivation Inventory (IMI) [35], and Short AttrakDiff-2 (SAD-2) [36].

4.2. Item selection and categorisation

We initially considered all items from the candidate questionnaires and proceeded to exclude items that did not fit our purpose. We de-duplicated items, removing those that were the same in two or more of the original questionnaires. We retained items that concerned episodic experience and discarded items that were referred only to immediate (e.g., "The appearance is beautiful.") and cumulative experience (e.g., "I will recommend it to my friends.").

Under the episodic experience category, we continued to subdivide the items by cognition, affect and conation to ensure that the instrument would cover the three aspects of the DX framework [1]. To perform this subdivision, we used the following guidelines:

1. *Cognition:* Items related to performing tasks and the efficiency or effectiveness of performing them, or to the usefulness of a tool in helping to perform them.
2. *Affect:* Items that describe emotions in a situation or related to a task or activity.
3. *Conation:* Items that describe the result or value obtained by a developer or that are related to developers' motivation or volition.

We did not assume that the final questionnaire should consist of items that are possible to map exactly to a single dimension in the framework. The purpose of this stage was to ensure a good starting point for the following steps in the questionnaire development.

At this stage, we wanted respondents' responses to focus only on their own experiences with the application development process, avoiding comments and perceptions about the tool or the task. Personal experience is complicated and can be influenced by various factors. Considering the aims of the questionnaire, it is more important to capture developers' notions of the overall experience of making an application than perceptions of the tool itself or the task they have just completed. For example, they might be satisfied with the user interface of the tool but anxious about the slow speed at which they progress in making their application; or they might feel that the platform is hard to use but the task in itself is easy to complete. With our focus being on individual developers' experience, we wanted the questionnaire to capture their assessment of their internal experience and not their observations of the tool or task. Also, focusing on the experience of the process of developing an application makes the questionnaire more general and not tied to any specific features of an LCDP. For these reasons, we removed items that referred to features or characteristics of tools, such as "It has explicit guidelines." and "It has a high level of information."

4.3. Questionnaire format

Since our questionnaire is meant to be used after episodes that may range from minutes to a few hours at most, we wanted a format that is quick to fill in. To that end, we adopted a semantic differential format (polar word pairs), following some existing and widely used UX questionnaires such as AttrakDiff-2 [36]. We also wanted a granular enough scale to let respondents capture nuances between the polar ends of each item. Psychometric literature suggests that more scale points provide better statistical properties for the data [37], but this has to be balanced with complexity for participants. While the issue can be debated, we opted for a seven-point scale based on current psychometric research (see, e.g., Taherdoost [38] for a summary).

The outcome of the preliminary design stage was a set of 12 items. These are available in the supplementary material [39].

5. Delphi study and final questionnaire

Delphi studies aim to improve decision-making by seeking the most reliable consensus from a group of experts [32]. The preliminary questionnaire described above was based on a literature review and our subjective analysis. We sought strengthened confidence in content validity, i.e., that our questionnaire items were consistent with the nature of LCDPs, and whether the wording was understandable in a practical setting. To this end, we contacted prospective experts through our industry collaboration networks and a purposive search on LinkedIn. Lynn [40] suggested using at least three and at most ten panellists for evaluating content validity. We selected and invited six experts with technical and research backgrounds to comment on the questionnaire. All participants were interested in DX and had work responsibilities related to the topic. Three experts work as UX Designers for an LCDP company; one is a principal user researcher in global large-scale research programs;

and the remaining two have rich experience with managing developers in industry practice. The participants are based in the Czech Republic, Finland, France, and the USA.

We held online meetings where we presented the in-progress questionnaire items to each expert in turn. They were free to make any comments and suggestions on the items, and we asked them open-ended questions to gain knowledge of how they perceived the items and why. After each session, we reviewed the opinions collected and refined the questionnaire. The new questionnaire was presented in the next Delphi session. In the last two iterations, the experts had no new remarks about the questionnaire that would have warranted further changes, i.e., we reached a point of saturation.

In the final questionnaire, we added two additional sections to specifically capture perceptions of the tool and the task separately from the respondent's overall individual experience.

The final questionnaire is shown in Table 1. The questionnaire's main instruction is to choose the most appropriate description for one's experience. The word pair items in the first part each capture an aspect of the respondent's experience. The questionnaire does not attempt to assess, for example, the details of the LCDP or the product being developed using it. Parts 2 and 3 are included to provide a way to separately assess aspects of the tool and task. They do not have to be used if this information is not needed. In this paper, they are used for the preliminary validation of the questionnaire.

Table 1. Final questionnaire used in the experiment of measuring end-user developers' episodic experience of LCDPs. Parts 2 and 3 are control items, while part 1 constitutes the developer experience instrument

Part 1: Episodic Developer Experience Questionnaire for Low-code Development Platforms (EDEQ-LCDP).			
Think about the session you just completed. Please enter what you consider the most appropriate description **for your experience** during the session.			
1.	Smooth	○ ○ ○ ○ ○ ○ ○	Exhausting
2.	Easy	○ ○ ○ ○ ○ ○ ○	Difficult
3.	Frustrating	○ ○ ○ ○ ○ ○ ○	Enjoyable
4.	Satisfying	○ ○ ○ ○ ○ ○ ○	Dissatisfying
5.	Distracted	○ ○ ○ ○ ○ ○ ○	Concentrated
6.	In control	○ ○ ○ ○ ○ ○ ○	Lack of control
7.	Exciting	○ ○ ○ ○ ○ ○ ○	Boring
8.	Slow	○ ○ ○ ○ ○ ○ ○	Quick
9.	Free to explore	○ ○ ○ ○ ○ ○ ○	Limited
10.	Productive	○ ○ ○ ○ ○ ○ ○	Unproductive
Part 2: How much do you agree with the following statements regarding **the tool**.			
		Completely Disagree – Completely Agree	
1.	The tool was reliable	○ ○ ○ ○ ○ ○ ○	
2.	The user interface was consistent	○ ○ ○ ○ ○ ○ ○	
3.	The tool was quick to respond	○ ○ ○ ○ ○ ○ ○	
Part 3: How much do you agree with the following statements regarding **the task**.			
		Completely Disagree – Completely Agree	
1.	The task was easy to understand	○ ○ ○ ○ ○ ○ ○	
2.	The task was easy to complete	○ ○ ○ ○ ○ ○ ○	

6. Task-based test

To gain insights into how our instrument works when applied to practical settings, we performed a task-based experiment with 19 participants. To identify the complexity and the acceptability of the tasks, we first conducted a pilot study with three participants in which we tested and refined the tasks iteratively. We analysed the resulting data using statistical methods to gain an initial picture of its suitability for field use.

6.1. Participants

One of the ambitions of LCDPs is to enable end-user developers to build their own applications without much effort. The learning cost and effort for professional and non-professional developers are generally not the same. We therefore created a screening survey to obtain two groups of participants, one of which had 14 participants with a programming background and the other consisting of 5 participants without. The entire set of participants included 10 students and 9 company employees. They were all fluent in English and mainly from Asia, Europe, Africa and the USA. Their fields of work or study included: research (6), software development (3), engineering (4), design (3), teaching (1), materials (1), and new media (1). The participants participated on the condition of anonymity and were compensated with a gift card worth 20 €.

6.2. Procedure

The test sessions were performed online, using SAP AppGyver[1], which is marketed as a no-code development platform for building mobile apps for both Android and iOS. As mentioned before, we consider no-code platforms to be a subset of low-code platforms. The LCDP's positioning of itself in the marketplace does not make a clear distinction between low- and no-code. For example, in AppGyver's case, some formula functions require a basic knowledge of programming to read and write, such as data variables and IF-statements. AppGyver provides clear documentation, but we found in our testing that some participants still found it difficult.

Participants shared their screens to allow us to observe their use of the tool. Each session lasted one hour and consisted of three parts:

- **Introduction and guided tour (20 minutes):** We explained the study procedure, what data would be collected and how it would be used and retained, and obtained informed consent following our university research ethics guidelines.
- **Warm-up task (5 minutes):** Since most of the participants had never used LCDPs, they might feel confused or nervous about using a new tool for the first time. We therefore used a warm-up task to help them get familiar with the tool. In this task, participants were instructed to create the user interface of a simple to-do list application that displayed a list of tasks, an input field and a button with the text "Add task".
- **Two independent tasks (35 minutes):** Then, in the main study tasks, participants added simple functions to the user interface of the to-do application created earlier to make it interactive, including the functionalities "Delete task from to-do list" (study task 1) and "Add new task to to-do list" (study task 2). After each of these two tasks, participants filled in our DX questionnaire using an online form.

[1] https://www.appgyver.com/

6.3. Analysis

We first identified differences in how participants felt across tasks and then used the Mann–Whitney U Test and Kendall's Tau to analyse the correlations between individual experience items and various perspectives of tasks and the tool. After that, we used the Intraclass Correlation Coefficient (ICC) to test the consistency among participants in the same group.

6.3.1. Task difficulty vs. developer experience

By comparing the scoring details of the two tasks, we made a simple assessment of the validity of the scoring. Concerning task completion, 12/19 people completed task 1, with 5 of them following the prompts. 5/19 succeeded in task 2, but only 2 of them completed it independently. Table 2 presents the average scores of tool quality and task easiness. Responses to *The task was easy to complete* indicate that task 1 was easier to complete than task 2. This finding is consistent with our intended task design. In contrast, participants considered task 2 to be easier to understand. This can be explained by the learning effect: when doing the first task, participants needed time to understand the task, but due to the continuity and similarity of the tasks, they could be more confident about the second task. In terms of individual experience factors, all experience factors got higher scores in task 1 except for *Satisfying – Dissatisfying*. Participants might be more satisfied about their performance during the process if they solved a complex problem.

Table 2. Average score of tool quality and task easiness in tasks 1 and 2. $N = 19$

Item	Task 1	Task 2
The task was easy to understand.	4.842	5
The task was easy to complete.	4.158	3.211
The tool was reliable.	5.053	5.105
The user interface was consistent.	5.158	4.842
The tool was quick to respond.	5	5.368

6.3.2. Individual experience items vs. the evaluation of the tool and tasks

We ran a Mann–Whitney U Test to compare whether participants who evaluated the platform or the task highly and those who evaluated them low had significantly different experiences. We compared the responses of those developers who assessed these factors on a seven-point scale of 5 or higher and those who assessed them as 3 or lower. Notably, we tested tasks 1 and 2 separately to avoid bias since they were different. However, for the platform, there were no statistically significant differences between those evaluation perspectives (tool reliability, user interface consistency and tool response) in either task. Even if there may be significant differences in one of the tasks, they do not apply to the other. For example, in task 2, those respondents with high experience satisfaction perceived the platform to be highly reliable more often than those with low experience satisfaction, whereas this was not evident in task 1. This indicated the difficulty of the tasks played an essential role in measuring episodic experience.

The same applies to the evaluation of the understanding of the task. None of the experience items were able to distinguish between users who found it easy or difficult to

Table 3. Statistically significant Mann–Whitney U test results between respondent groups Task Completion$_{good}$ and Task Completion$_{bad}$ for all items. n.s. = not significant. $N = 19$

	Task 1		Task 2	
Item	U	p	U	p
Easy – Difficult	8	<0.01	9.5	<0.05
Free to explore – Limited	15.5	<0.05	14	n.s.
In control – Lack of control	15.5	<0.05	8	<0.05
Exciting – Boring	10.5	<0.05	10.5	<0.05
Satisfying – Dissatisfying	32.5	n.s.	5.5	<0.01
Quick – Slow	16.5	<0.05	16	n.s.
Concentrated – Distracted	22.5	n.s.	5.5	<0.01
Enjoyable – Frustrating	22.5	n.s.	0.5	<0.001

understand either task, although four experience items could differentiate users in task 1. As for participants who found the task easy to complete and those who found it difficult, 3 out of 10 word pairs (*easy*, *in control* or *excited*; see Table 3) supported the separation between them. This indicates that when participants rated these items high after performing a task using the LCDP, they were also more likely to find the task easy to complete.

We used Kendall's Tau Correlation analysis to discover which items in each scale were significantly correlated with each perspective.

From the results, few items had strong correlations with user interface consistency and tool response. Since the LCDP used in this test had already been subjected to user testing and was already a mature product, it was not surprising that these two items were not strongly correlated with differences in individual experience; they may have captured more objective traits of the tool. However, two other items, tool reliability and task completion, reflected strong correlations with a majority of individual experience items:

Tool reliability. Correlations between tool reliability and individual experience items are presented in Table 4. 7/10 (70%) of word pairs correlated with tool reliability in task 1 while 6/10 (60%) had significant correlations with tool reliability in task 2. What is notable is that 4/10 (40%) of all experience items had significant correlations with tool reliability both in tasks 1 and 2. All experience items could predict tool reliability either in task 1 or 2 except for *Smooth – Exhausting* which only showed a strong correlation with the ease of task completion in task 2. In addition to the possibility that it had little to do with the platform and the task, we also speculate that it was the result of a more diverse user understanding of this experience factor.

Table 4. Results of Kendall's Tau correlation analysis between the tool was reliable assessment and individual developer experience items. n.s. = not significant. $N = 19$

	Task 1		Task 2	
Item	τ	p	τ	p
Easy – Difficult	0.434	< 0.05	0.301	n.s.
Free to explore – Limited	0.331	n.s.	0.369	< 0.05
Satisfying – Dissatisfying	0.331	n.s.	0.369	< 0.05
Productive – Unproductive	0.507	< 0.01	0.227	n.s.
In control – Lack of control	0.49	< 0.01	0.479	< 0.01
Exciting – Boring	0.545	< 0.01	0.505	< 0.01
Enjoyable – Frustrating	0.514	< 0.01	0.597	< 0.01
Concentrated – Distracted	0.407	< 0.05	0.509	< 0.01
Quick – Slow	0.418	< 0.05	0.318	n.s.

Table 5. Results of Kendall's Tau correlation analysis between the task was easy to complete and individual developer experience items. n.s. = not significant. $N = 19$

	Task 1		Task 2	
Item	τ	p	τ	p
Smooth – Exhausting	0.298	n.s.	0.399	<0.05
Easy – Difficult	0.583	<0.01	0.363	n.s.
Free to explore – Limited	0.389	<0.05	0.37	<0.05
Satisfying – Dissatisfying	0.312	n.s.	0.644	<0.01
Productive – Unproductive	0.465	<0.05	0.372	<0.05
In control – Lack of control	0.495	<0.01	0.372	<0.05
Exciting – Boring	0.576	<0.01	0.519	<0.01
Enjoyable – Frustrating	0.389	<0.05	0.725	<0.01
Concentrated – Distracted	0.265	n.s.	0.412	<0.05
Quick – Slow	0.475	<0.05	0.478	<0.05

Task completion. 7/10 (70%) and 9/10 (90%) of items significantly correlated with task completion in tasks 1 and 2. All items had significant correlations with task completion in both tasks 1 and 2 (Table 5). It is worth noting that compared to other items, *Satisfying – Dissatisfying* and *Enjoyable – Frustrating* had a stronger relationship ($\tau = 0.644$ and $\tau = 0.725$, respectively) with task completion in task 2. The more participants felt confident completing the task, the more satisfied and enjoyable they felt especially when the task was hard.

With the Mann–Whitney U test and Kendall's Tau correlation analysis, we realized some experience items were changeable across the two tasks. The difficulty of tasks seems to play a great role in measuring developer experience. In further research or validation, a comparison of more tasks of different difficulties together would be advisable.

6.3.3. Programming backgrounds vs. developer experience

Intraclass Correlation Coefficient (ICC) was used to find out if there was consistency in the results of participants with a similar programming background. Table 6 shows the level of consistency in the performance of different groups. According to statistical guidelines [41, 42], ICC values higher than 0.50 are acceptable. Therefore, the experience of participants without programming experience showed consistency in both tasks. Conversely, the participants who had programming experience exhibited inconsistent performance across the two tasks.

Thus, programming experience does make a difference to the episodic experience of the LCDP, but this effect is varied. Our participants' programming expertise, such as

Table 6. Results of Intraclass Correlation Coefficient for each two groups measuring the experience. n.s. = not significant

	Task 1		Task 2	
Group	ICC	p	ICC	p
A: No programming experience	0.582	<0.001	0.626	<0.001
B: <3 years of programming experience	0.093	<0.001	0.185	<0.001
C: ≥3 years of programming experience	0.253	n.s.	−0.160	n.s.
B and C	0.076	<0.01	0.150	<0.001

programming language and skills, was diverse, indicating that the quantitative findings make intuitive sense. For example, one participant was mainly responsible for data analysis using existing libraries while another focused on algorithmic programming. Developers writing operational logic and visual user interfaces also had diverse understandings of workflow. Most of them thought their prior programming expertise did not contribute to their usage of the LCDP. But one participant – a front-end developer – found it easy to do the task because the logic was almost the same. More experiments are needed to discover the relationship between different programming expertise and the experience of using an LCDP.

7. Discussion

In the previous sections, we have described how we created a preliminary questionnaire instrument for measuring episodic developer experience during LCDP use based on analysis of prior literature, how we refined this questionnaire using a Delphi study with experts to obtain a final questionnaire (EDEQ-LCDP), and how this questionnaire was used in a task-based test.

Our research question asks how to measure the end-user developers' episodic experience of a low-code development platform. We designed a questionnaire with 10 items (Part 1 in Table 1) to answer it. Each question item measures a specific aspect of the episodic experience. The items in parts 2 and 3 are used to provide a way to separate personal experience from the assessment of the LCDP and the tasks, and they do not have to be used in practice. In the process of developing the questionnaire instrument, we suggested separating experience from task difficulty and tool usability because experience is quite complicated and affected by various aspects, such as tool, task or even the working environments. The experience may consist of positive and negative assessments independent of tool usability and task difficulty. We tried to free respondents from identifying their precise feelings. Furthermore, we compared our work with related work and discussed the differences between episodic experience and cumulative experience.

7.1. Experience should be set apart from task difficulty and tool usability

The questionnaire we designed covers 10 items to measure developer experience after an episode of completing tasks in an LCDP. All of them are related to developers' thoughts and feelings. We tried to separate the personal experience of an application development process from the task and the tool. During the study, some experts and participants asked us what an item referred to, such as whether *Satisfying – Dissatisfying* meant that the tool was satisfying or that developers felt satisfied with their performance during the process.

We aimed to focus on participants' overall feelings when they were performing the task. Some people might find a tool novel and helpful, but feel dissatisfied with their performance, perhaps because they were unfamiliar with this type of platform or were doubting their ability to perform the task. As a result, their experience of the process was unsatisfactory. We were more interested in their overall personal experience than asking them directly about their attitude to the platform or the task.

Each personal experience factor might be affected by multiple factors. For example, in this study, *Easy – Difficult* did not show a significant correlation with *The task was easy to complete*. One participant gave task difficulty the lowest score but scored their experience

highly difficult. This participant explained that although they personally felt it was difficult to perform the task, they still considered the task to be simple in general and that their lack of expertise was the main reason for feeling it was difficult. If they would have a second chance to do the task, they thought they would perform better. Another participant also found the task easy, but felt nervous about being asked to complete it within a specific time. They stated that given free time, they would look up the documentation carefully.

These examples illustrate the importance of capturing the overall personal experience of development episodes, and keeping this separate from usability assessments of tools. Tool providers usually aim to create a better experience for tool users – in this case, developers using an LCDP. It is not enough to obtain assessments of the product's appearance, functionality, help manuals, etc.; only if developers obtain positive experiences, which may depend on a multitude of personal reasons, and if they feel happy, satisfied, and have achieved their goals using the product, they can become inclined to continue using it and recommend it to their peers.

7.2. Comparing to related work

Our DX instrument was based on prior literature and it bears resemblance to many UX measurement instruments. However, most UX instruments that we are aware of focus on the respondent's assessment *of the product* rather than *of their personal experience*. There are also some existing instruments specifically for measuring DX, but they are not specifically designed for episodic experience. We discuss these issues using a few examples of existing questionnaires below.

The User Experience Questionnaire (UEQ) [43] contains 26 items including six dimensions: Attractiveness, Perspicuity, Efficiency, Dependability, Stimulation, and Novelty. It aims to be quick and easy to apply and is suggested to complement the use of other evaluation methods with subjective quality ratings. In contrast to our questionnaire, it is focused more on an assessment of the *product* than the experience. For instance, the UEQ item *Usual – Leading edge* describes the technology used in the product. *Conventional – Inventive* stresses the novelty of the product. The items thus represent the respondent's personal assessment of the product but do not necessarily reflect an assessment of their personal experience with using the product in a concrete task episode.

Similarly, AttrakDiff-2 aims to capture product perceptions and evaluations [36]. It consists of items to capture pragmatic and hedonic quality as well as general quality. With items such as *confusing – structured* (pragmatic), *dull – captivating* (hedonic), and *good – bad* (general), it appears to be asking for assessments of different aspects of product quality. The respondent's experiences with the product can be assumed to affect these assessments, but as with the UEQ, what is asked for is an opinion regarding the product, not an account of the respondent's personal experience with using it.

Many other UX measurement instruments follow the same approach. Our instrument takes steps to focus the respondent's attention on their personal experience by instructing them to think about the development episode they just performed and to provide "a description for your experience". While they may factor in aspects of the usability of the tool or the difficulty of the task, this is by intention: the experience of the previous development episode is based on a multitude of factors but what we wish to gain insight into are the contents of the participant's near-term episodic memory of that episode.

Another aspect of the present study is that it focuses on the context of development, where participants are creating new software using a development tool. While DX can be

considered a sub-field of UX, the experience of developers has its own specifics that should be taken into account and that are relevant also to inform the management and organisation of software development in organisations. For example, developers often have particular wishes when it comes to tool choice, such as the extent to which they are customisable according to the developer's wishes, often with scripts or plug-ins that can extend or alter its functionality. They may also have preferences regarding the way a tool allows them to express their ideas, and they may want to know what "goes on under the hood", i.e., they may be concerned about their ability to control the final product when using a high-level tool that hides implementation details. Such concerns are less typical for end-users who are not producing new software but rather work with data such as text or images where the result is what they see on their screen. For these reasons, we have taken the characteristics of development tools and their use in both an organisational and individual setting in our questionnaire.

To our knowledge, instruments for specifically measuring DX are rare. However, some do exist. The DEXI scale is meant specifically for software development as a general questionnaire for measuring DX of a GUI designer tool [18]. DEXI was formed based on frequently used UX questionnaires by selecting items from them that are appropriate for describing aspects of DX. Its *Limited – Extensive* is an example of capturing the customizability aspect of the developer's tools to give them as much freedom as possible. We used *Limited – Free to explore* to characterize this kind of experience; the wording was informed by our expert feedback. DEXI is similar to the UX questionnaires discussed above in that it focuses on the participant's assessment of the product – in the case of DEXI, an IDE. It thus does not address the participant's personal experience directly, but rather incorporates it into the respondent's assessment of the tool. In our view, this has at least practical implications for the use of the questionnaires to inform DX improvements. Whereas DEXI directs the focus towards the tool itself, our questionnaire invites a discussion about the participant's experience in general, which may reveal issues regarding the tool, but could also lead to discovering issues related to the task, the participant's perception of themselves as a developer, or relationships between individual, task, and tool. While it is not guaranteed that a participant will discuss issues beyond the tool, our instrument does avoid directing attention primarily towards the tool. Depending on the purpose, this may or may not be beneficial.

In other work, Lee and Pan [13] constructed a set of items to measure DX of deep-learning platforms based on both the conceptual model of DX [1] and related research on customer experience in fields such as marketing. Their questionnaire uses a different format with statements such as "The interaction between the platform and the developer does not require much mental effort" (cognitive), "The platform is attractive" (affective) and "Developers intend to use the platform" (conative). Respondents indicate agreement on a seven-point scale. Beyond the obvious difference of using a statement format, this questionnaire is much more general than ours and tries to map each item specifically to one dimension of the DX framework. It also appears to mix different levels of experience, with items regarding mainly cumulative and anticipated experience; the conative item here is an example of the latter. Our instrument is more focused specifically on LCDPs.

7.3. Episodic experience vs. cumulative experience

As noted above, previous DX questionnaires are not specific to episodic experience. Instead, the majority of research into DX instruments has focused to a large extent on cumulative experience, with some introduction of anticipated experience.

Studying cumulative experience makes sense when there is reason to believe that enough time has passed for such experiences to form, such as when people have used a product or tool for a long time. The main points of concern with regard to cumulative experience are different from the concerns of the first few times people use a product to solve a problem or accomplish a task. Taking the view that cumulative experience is a function of experiences over several episodes, one can assume that there is an overlap between cumulative and episodic experiences. However, previous research constructing DX instruments hardly distinguishes the two specifically.

Factors such as cost, backward and forward compatibility, version migration paths, and the surrounding developer community are important for the cumulative experience related to development tools, while such factors play a smaller role in the episodic experience. Motivation is another rather different aspect between the two levels of experience. During a development episode, developers seek a solution to a small set of specific problems while they may hope to achieve a bigger goal during long-term usage of the platform, including goals that span beyond an individual piece of software they are developing. For example, people's intention to use an LCDP is to build an application, which is a long-term goal. The goal will be achieved by several episodic goals which are to implement various features of the application. Beyond these, they may have long-term motivations such as advancing in their career. We thus argue that DX questionnaires should be designed with the different levels of experience in mind: general DX questionnaires focusing on cumulative experience have to consider factors beyond the tool, while instruments focusing on episodic experience can take a more narrow set of factors into account.

We believe that motivation also influenced the result of the test in our study. Participants did not have expectations towards this platform since it was their first time using it. They were unlikely to have an emotional attachment to the platform at the outset and were just required to perform the task whether or not it was inherently interesting to them. We therefore did not include general factors such as *Cheap – Premium* (from AttrakDiff-2) or *"Even if the platform price is high compared to other platforms, I think that developers should use the current platform"* (from Lee and Pan). We believe that such general and circumstantial factors are not of primary relevance to episodic experience, but they may work as part of assessments of anticipated or cumulative experience. However, developers, even end-user developers using LCDPs, may not be involved in tool purchase decisions so the cost of acquiring such tools may not be relevant to them.

Concentrated – Distracted is the only item that was strongly correlated with nearly all aspects of the platform in both tasks 1 and 2. Users' attention during an event influences the time, efficiency and completion of the task, therefore influencing their feelings. Concentration on task has been associated with flow experiences in prior literature (e.g., [34]), but it was not significantly correlated with developers' overall experience in assessing DX of a GUI designer tool [18]. Thus, the role of attention in episodic and cumulative developer experiences should be further investigated.

7.4. Insights for further research

This study provides a starting point for further research. Our statistical analysis of the questionnaire responses in our study gives preliminary indications that should be investigated further. The results of the Mann–Whitney *U* Test indicated that *Easy – Difficult*, *In control – Lack of control* and *Exciting – Boring* can differentiate between participant groups with low and high Task Completion Difficulty assessment. Of these three

items, the latter two were also significantly correlated with the ease of task completion (Kendall's Tau Correlation Analysis). In addition, the correlation analysis revealed an additional four items that had similar correlation: *Free to explore – Limited*, *Productive – Unproductive*, *Concentrated – Distracted* and *Quick – Slow*. Task Understanding was strongly correlated with *Productive – Unproductive* and *Quick – Slow*.

A number of items were correlated with the reliability of the platform although we could not successfully differentiate between participant groups based on these items. For example, *In control – Lack of control*, *Exciting – Boring*, *Enjoyable – Frustrating* and *Concentrated – Distracted* are associated with tool reliability. In other words, we might compare the reliability of different LCDPs, and the LCDPs that get higher feedback in these items might be more reliable. User interface consistency and tool response did not show correlations with any experience item. In this study, they could not be predicted by participants' feelings.

Despite the above findings, many open questions remain and there is a need for more research to further test the validity of the questionnaire items. For instance, why do some items perform differently across tasks, and is it related to the difficulty of the task? Why did *Productive – Unproductive* show a significant correlation with the ease of understanding the task? One possibility is that participants could not distinguish between their ability to understand the task and the outcomes they produced. This could have interesting implications for studying DX in general: the memory of a development episode may be distorted by the outcomes, meaning in this case that developers determine their understanding of a task based on whether they were able to produce something tangible, regardless of whether the outcome conformed to what was intended from the outset or not. This would be consistent with prior research showing that, along with the most intense part, the final part of an experience episode has a strong influence on the assessment of the whole episode [44]. Investigating whether interventions in the final part of an episode have an impact on DX, as measured here, could be an interesting avenue for further research.

Finally, the approach taken here has aimed to examine the instrument to inform the future development of DX questionnaire instruments. To obtain instruments with better validated psychometric properties, the results and insights developed in this paper should be combined with a rigorous instrument development approach. Using factor analysis and correlating candidate instruments with existing DX and UX instruments as well as other ways of obtaining experiential information are possible ways to obtain more strongly validated DX measurement instruments.

7.5. Limitations

The main aim of our study was to develop a questionnaire with a grounding in present DX literature and expert knowledge. The primary concern was how to operationalise the DX concept in the context of LCDPs to capture important experiential factors in development episodes. Thus, we consider *construct validity* and *content validity* to be the most important criteria for the study, i.e., to what extent the questionnaire reflects the concepts it should reflect. Content and face validity have been identified as important for instrument development related to experiential characteristics in other fields [45] and we consider the perspective of experts and practitioners to be crucial for studies on DX measurement as well. Correlational analyses are commonly used to assess construct validity along with incorporating qualitative information from experts and practitioners, as we have done in this paper.

The combination of using prior literature about DX and the Delphi study with experts serves to increase the confidence that the questionnaire reflects prevailing understandings of episodic DX of LCDPs. As we have noted above, prior literature does not, to the best of our knowledge, provide a measurement instrument specifically aimed to measure episodic DX. The literature therefore served to inform the study about potential questionnaire items relevant to DX in general, while our analysis of those items against the DX conceptual framework [1], our knowledge of DX and LCDPs, and the comments from experts in the Delphi study served to refine the questionnaire to capture the desired aspects of episodic DX as well as possible. Our analysis of the task-based test results shows that the questionnaire is understandable by practitioners and can be used to distinguish between different kinds of experiences by developers with different backgrounds. While a single study is not enough to provide construct validity, we nevertheless consider the primary concern of our study to have been adequately addressed. We acknowledge that further work is needed to gain more confidence in the content and criterion validity of the instrument. However, if the questionnaire omits some aspect of the experience that would be relevant for development episodes using LCDPs, adding elements must be balanced against the practical effort required for participants to fill in the questionnaire.

Internal validity is concerned with the extent to which cause-effect relationships are trustworthy within the context of a study. In our case, selection bias and confounding could have influenced the differences in participants' responses.

Prior knowledge is a significant influencing factor when studying episodic developer experience of LCDPs. One of the goals of this study was to discover the impact of prior knowledge on using the LCDP to perform tasks. We attempted to understand whether programming expertise, LCDP expertise, and design expertise have impacts on episodic experience when participants are using LCDPs to perform tasks.

ICC analysis indicates that programming expertise does make a difference to the episodic experience of the LCDP: participants with no knowledge of programming rated the experience items consistently, while those with programming knowledge rated them differently. We observe that our measure of programming knowledge was not detailed enough to discern meaningful patterns among the latter group. Based on the participants' LCDP and design tool expertise, no consistencies were found between participants, regardless of which group they were involved. Due to the lack of a clear definition and control of participants' backgrounds, the correlation between background and experience is not significant. However, by explicitly controlling for and analysing expertise on different levels, we attempted to guard against selection bias and confounders; in other words, prior differences in expertise were accounted for in the study design. Further validation is needed to discover other potential confounders.

A further threat to internal validity is researcher bias: we could have accidentally influenced participants in the study. We took steps to guard against this. In the Delphi study, we presented experts with our in-progress questionnaire and asked them to comment without stating our own opinions. Only after they had provided their feedback did we engage in a discussion where we explained our rationale behind items. In the task sessions, the training task provided participants with an introduction to the most important functions of the LCDP, and they then had to carry out the two other tasks on their own. In cases where they asked for help, we did not immediately solve the problem for them but instead encouraged them to keep exploring until they ran out of time. Only after filling in the questionnaire did we help them complete the task if they still wished to.

Another apparent threat undoubtedly arises from the small sample, the use of only two tasks, and the use of only one LCDP, which raises concerns about *external validity.*

The study groups had one to three participants and the total number of participants (19) is not representative of any larger population of developers. However, we did not aim to validate the questionnaire in a well-defined population. Rather, we focus here on obtaining a questionnaire with good enough construct validity to warrant use in practical settings together with other means of assessment and for further research. The number and nature of tasks also prevent us from drawing far-reaching conclusions about the relationship between task traits and questionnaire items. Both diversified and repeated tasks are necessary to strengthen the validity of the questionnaire for the study of episodic experience.

Finally, the test only covers one LCDP and thus it can be questioned to what extent the results are generalisable to other LCDPs or development tools in general. However, we argue that this aspect is not relevant to the present study. The study design did not rely on any particular aspect of the LCDP used, and the questionnaire does not contain any items that would rely on the characteristics of any particular LCDP. The questionnaire specifically directs participants to report on their personal experience of the session they conducted (i.e., the development episode), not to assess the tool. Thus this should not be a primary validity concern, and we note that many widely used usability and UX instruments have only been validated through years of use and study after their initial introduction. Still, further studies could assess external validity further by varying the LCDP used.

8. Conclusions

In this study, we set out to obtain an understanding of end-user developers' episodic experience when building an application on an LCDP to guide how such experiences can be measured. To this end, we designed a questionnaire with ten experience items to measure the episodic experience of developers using LCDPs after they had completed a specific bounded task. All items were based on prior literature on the characteristics of LCDPs and developer experience. The items were refined based on expert feedback in a Delphi study. We then used a task-based test to obtain data from the actual use of the questionnaire in conjunction with an LCDP. Participants also provided background information on their prior expertise and received a short training session using the LCDP. We analysed correlations between the experience items in our questionnaire and participant assessments of tool reliability, user interface consistency, tool response, task understanding, and task completion, as well as their assessment of prior expertise.

The results show that all experience items in the questionnaire were related to the characteristics of the tool and the difficulty of the task. Perceived task difficulty, tool reliability, and task completion appear to play a role in episodic experience. We emphasise the importance and necessity of separating personal experience from the assessment of tasks and tools. In terms of the impact of developers' background, prior programming experience played an important role in measuring the episodic experience of LCDPs, but a more detailed definition and control of background is needed to verify this conclusion.

The findings in this study can be used to provide insights for further research into developer experience measurement. In particular, we advise instrument developers to develop a detailed understanding of the context of development and to carefully design their instruments for the desired time scale of experience, e.g., anticipatory, momentary, episodic, or cumulative. The questionnaire can be used, taking its limitations into account, for assessing the episodic developer experience of developers using LCDPs. We recommend

that it be used alongside other assessment methods such as think-aloud protocols or post-task interviews. We believe it could prove useful to assess development episodes involving other kinds of development tools besides LCDPs, but further research is required to validate its use for such purposes.

Acknowledgements

We would like to express our sincere thanks to the experts and study participants for their involvement in the research.

References

[1] F. Fagerholm and J. Münch, "Developer experience: Concept and definition," in *International Conference on Software and System Process (ICSSP)*. IEEE, 2012, pp. 73–77.

[2] F. Fagerholm, *Software developer experience: Case studies in lean-agile and open source environments*, Ph.D. dissertation, University of Helsinki, Faculty of Science, Department of Computer Science, Helsinki, Finland, 2015.

[3] D. Graziotin, X. Wang, and P. Abrahamsson, "Are happy developers more productive?" in *International Conference on Product Focused Software Process Improvement*. Springer, 2013, pp. 50–64.

[4] M.A. Storey, T. Zimmermann, C. Bird, J. Czerwonka, B. Murphy et al., "Towards a theory of software developer job satisfaction and perceived productivity," *IEEE Transactions on Software Engineering*, Vol. 47, No. 10, 2019, pp. 2125–2142.

[5] N. Baddoo, T. Hall, and D. Jagielska, "Software developer motivation in a high maturity company: a case study," *Software process: improvement and practice*, Vol. 11, No. 3, 2006, pp. 219–228.

[6] D. Graziotin, F. Fagerholm, X. Wang, and P. Abrahamsson, "What happens when software developers are (un)happy," *Journal of Systems and Software*, Vol. 140, 2018, pp. 32–47.

[7] H. Lieberman, F. Paternò, M. Klann, and V. Wulf, "End-user development: An emerging paradigm," in *End user development*. Springer, 2006, pp. 1–8.

[8] Y. Luo, P. Liang, C. Wang, M. Shahin, and J. Zhan, "Characteristics and challenges of low-code development: The practitioners' perspective," in *Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. ACM, 2021, pp. 1–11.

[9] M. Tisi, J.M. Mottu, D.S. Kolovos, J. De Lara, E.M. Guerra et al., "Lowcomote: Training the next generation of experts in scalable low-code engineering platforms," in *STAF 2019 Co-Located Events Joint Proceedings: 1st Junior Researcher Community Event, 2nd International Workshop on Model-Driven Engineering for Design-Runtime Interaction in Complex Systems, and 1st Research Project Showcase Workshop co-located with Software Technologies: Applications and Foundations (STAF 2019)*, 2019.

[10] J. Cabot, "Positioning of the low-code movement within the field of model-driven engineering," in *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, MODELS '20. ACM, 2020, pp. 1–3.

[11] A. Sahay, A. Indamutsa, D. Di Ruscio, and A. Pierantonio, "Supporting the understanding and comparison of low-code development platforms," in *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, 2020, pp. 171–178.

[12] F. Khorram, J.M. Mottu, and G. Sunyé, "Challenges and opportunities in low-code testing," in *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, MODELS '20. ACM, 2020, pp. 1–10.

[13] H. Lee and Y. Pan, "Evaluation of the nomological validity of cognitive, emotional, and behavioral factors for the measurement of developer experience," *Applied Sciences*, Vol. 11, No. 17, 2021, p. 7805.
[14] E. Karapanos, J. Zimmerman, J. Forlizzi, and J.B. Martens, "User experience over time: an initial framework," in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2009, pp. 729–738.
[15] B. Callary, P. Werthner, and P. Trudel, "How meaningful episodic experiences influence the process of becoming an experienced coach," *Qualitative research in sport, exercise and health*, Vol. 4, No. 3, 2012, pp. 420–438.
[16] P. Marti and I. Iacono, "Anticipated, momentary, episodic, remembered: The many facets of user experience," in *Federated Conference on Computer Science and Information Systems (FEDCSIS)*. IEEE, 2016, pp. 1647–1655.
[17] M. Greiler, M.A. Storey, and A. Noda, "An actionable framework for understanding and improving developer experience," *IEEE Transactions on Software Engineering*, 2022.
[18] K. Kuusinen, "Are software developers just users of development tools? assessing developer experience of a graphical user interface designer," in *Human-Centered and Error-Resilient Systems Development*. Springer, 2016, pp. 215–233.
[19] K. Kuusinen, "Software developers as users: Developer experience of a cross-platform integrated development environment," in *International Conference on Product-Focused Software Process Improvement*. Springer, 2015, pp. 546–552.
[20] T. Kaltio and A. Kinnula, "Deploying the defined SW process," *Software Process: Improvement and Practice*, Vol. 5, No. 1, 2000, pp. 65–83.
[21] D. Graziotin, F. Fagerholm, X. Wang, and P. Abrahamsson, "On the unhappiness of software developers," in *Proceedings of the 21st international conference on evaluation and assessment in software engineering*, 2017, pp. 324–333.
[22] K.R. Linberg, "Software developer perceptions about software project failure: A case study," *Journal of Systems and Software*, Vol. 49, No. 2–3, 1999, pp. 177–192.
[23] A. Bobkowska, "On explaining intuitiveness of software engineering techniques with user experience concepts," in *Proceedings of the International Conference on Multimedia, Interaction, Design and Innovation*, 2013, pp. 1–8.
[24] F. Fagerholm, M. Felderer, D. Fucci, M. Unterkalmsteiner, B. Marculescu et al., "Cognition in software engineering: A taxonomy and survey of a half-century of research," *ACM Comput. Surv.*, Vol. 54, No. 11s, 2022.
[25] M. Sánchez-Gordón and R. Colomo-Palacios, "Taking the emotional pulse of software engineering – A systematic literature review of empirical studies," *Information and Software Technology*, Vol. 115, 2019, pp. 23–43.
[26] G. Fischer, D. Fogli, and A. Piccinno, "Revisiting and broadening the meta-design framework for end-user development," in *New perspectives in end-user development*. Springer, 2017, pp. 61–97.
[27] B.R. Barricelli, F. Cassano, D. Fogli, and A. Piccinno, "End-user development, end-user programming and end-user software engineering: A systematic mapping study," *Journal of Systems and Software*, Vol. 149, 2019, pp. 101–137.
[28] W.W. Cotterman and K. Kumar, "User cube: A taxonomy of end users," *Communications of the ACM*, Vol. 32, No. 11, 1989, pp. 1313–1320.
[29] M. Hirzel, "Low-code programming models," *Communications of the ACM*, Vol. 66, No. 10, 2023, pp. 76–85.
[30] C. Richardson, J.R. Rymer, C. Mines, A. Cullen, and D. Whittaker, "New development platforms emerge for customer-facing applications," Forrester, Cambridge MA USA 15, Tech. Rep. 16, Jun 2014.
[31] R. Waszkowski, "Low-code platform for automating business processes in manufacturing," *IFAC-PapersOnLine*, Vol. 52, No. 10, 2019, pp. 376–381.
[32] N. Dalkey and O. Helmer, "An experimental application of the Delphi method to the use of experts," *Management Science*, Vol. 9, No. 3, 1963, pp. 458–467.

[33] A. Nylund, "A multivocal literature review on developer experience," Master's thesis, Aalto University School of Science, 2020. [Online]. https://aaltodoc.aalto.fi/items/d822c160-0cc6-41d1-9c29-72470ec2ad13

[34] S.A. Jackson, A.J. Martin, and R.C. Eklund, "Long and short measures of flow: The construct validity of the FSS-2, DFS-2, and new brief counterparts," *Journal of Sport and Exercise Psychology*, Vol. 30, No. 5, 2008, pp. 561–587.

[35] R.M. Ryan, "Control and information in the intrapersonal sphere: An extension of cognitive evaluation theory." *Journal of Personality and Social Psychology*, Vol. 43, No. 3, 1982, p. 450.

[36] M. Hassenzahl, S. Diefenbach, and A. Göritz, "Needs, affect, and interactive products–facets of user experience," *Interacting with Computers*, Vol. 22, No. 5, 2010, pp. 353–362.

[37] D. Alwin, "Feeling thermometers versus 7-point scales: Which are better?" *Sociological Methods and Research*, Vol. 25, No. 3, 1997, pp. 318–340.

[38] H. Taherdoost, "What is the best response scale for survey and questionnaire design; Review of different lengths of rating scale/attitude scale/likert scale," *International Journal of Academic Research in Management*, Vol. 8, No. 1, 2019, pp. 1–10.

[39] D. Gao and F. Fagerholm, "Supplementary materials for the paper 'Measuring end-user developers' episodic experience of a low-code development platform – A preliminary study'," 2023. [Online]. https://doi.org/10.5281/zenodo.7515833

[40] M. Lynn, "Determination and quantification of content validity," *Nursing Research*, Vol. 35, 1986, pp. 382–386.

[41] D.V. Cicchetti, "Guidelines, criteria, and rules of thumb for evaluating normed and standardized assessment instruments in psychology." *Psychological Assessment*, Vol. 6, No. 4, 1994, p. 284.

[42] T.K. Koo and M.Y. Li, "A guideline of selecting and reporting intraclass correlation coefficients for reliability research," *Journal of Chiropractic Medicine*, Vol. 15, No. 2, 2016, pp. 155–163.

[43] B. Laugwitz, T. Held, and M. Schrepp, "Construction and evaluation of a user experience questionnaire," in *Symposium of the Austrian HCI and usability engineering group*. Springer, 2008, pp. 63–76.

[44] D.A. Redelmeier and D. Kahneman, "Patients' memories of painful medical treatments: real-time and retrospective evaluations of two minimally invasive procedures," *Pain*, Vol. 66, No. 1, 1996, pp. 3–8.

[45] J. Connell, J. Carlton, A. Grundy, E. Taylor Buck, A.D. Keetharuth et al., "The importance of content and face validity in instrument development: Lessons learnt from service users when developing the recovering quality of life measure (ReQoL)," *Quality of Life Research*, Vol. 27, 2018, pp. 1893–1902.