Proceedings Paper:
Kuusinen, Kati, Petrie, Helen orcid.org/0000-0002-0100-9846, Fagerholm, Fabian et al. (1 more author) (2016) Flow, Intrinsic Motivation, and Developer Experience in Software Engineering. In: Proceedings of the International Conference on Agile Software Development:Agile Processes in Software Engineering and Extreme Programming (XP 2016). Lecture Notes in Business Information Processing . Springer

Flow, Intrinsic Motivation, and Developer Experience in Software Engineering

Kati Kuusinen[1], Helen Petrie[2], Fabian Fagerholm[3], and Tommi Mikkonen[1]

[1]Tampere University of Technology, Tampere, Finland
`kati.kuusinen@tut.fi, tommi.mikkonen@tut.fi`

[2]University of York, York, United Kingdom
`helen.petrie@york.ac.uk`

[3]University of Helsinki, Helsinki, Finland
`fabian.fagerholm@helsinki.fi`

Abstract. Software developers are both users of development tools but also designers of new software systems. This dual role makes developers special users of work-related software. To increase the understanding of developers as users and to evaluate the ability of common measurement scales to address developer experience, we conducted a survey measuring developers' flow state, intrinsic motivation and user experience. Scales used were the Short Dispositional Flow Scale, items from the Intrinsic Motivation Inventory, the Short AttrakDiff-2, and our own DEXI scale. 57 developers from 25 countries responded and results indicate that intrinsic motivation and autotelic experience are significant predictors of developers' UX whereas hedonic, pragmatic, and general quality are not. In addition, developers' needs are characterized by efficiency, informativeness, intuitiveness, and flexibility of the tool.

Keywords: Software development; User experience; Developer experience; Development tools; Integrated development environments; Human factors

1   Introduction

Software engineering (SE) is a professional human activity that demands numerous skills and qualities from developers. Technical skills are needed to create the code that builds the software, while social skills are needed to be able to collaborate with other developers and to communicate with stakeholders. SE is an endeavor which builds complex systems that realize user and business requirements in technologically sophisticated manners. Considering the challenges of SE, the user experience (UX) of developers is an area that has been very little studied. Developers are users of multifaceted development tools such as integrated development environments (IDEs). Yet little is known about how these tools support developers in their demanding activities and the nature of their UX with such tools.

IDEs are commonly used tools in SE, and are applications used for composing, compiling and debugging program code [1]. IDEs also manage dependencies among different packages and modules, control builds, and provide linking to other tools such as those for requirements management or test environments. Consequently, IDEs

play a major role in making developers productive and feel comfortable during their daily activities. Yet despite their important role, little is known about how these tools support developers and the nature of UX with such tools. While it may be overreaching to conclude that happy developers are better at their work [2], both happiness and motivation have been connected with raised productivity [3]. Mood influences developers' performance on programming tasks [4], and happiness has been found to have productivity benefits [5].

Although qualities of both developers and development work have been studied, developers have rarely been investigated as users of development tools. As developers are users of IDEs, all that is true of any user according to UX definitions (e.g. [6]), should apply also to developers. However, the dual role of developer as both users of systems and developers of systems makes them special: besides being IDE users, developers should be able to understand the human user to be able to fulfill their needs with the software under development. A concept of developer experience (DX) has been suggested to address the particularities to SE [7]. The concept of DX is influenced by the concept of UX [7]. Moreover, DX consists of aspects related to cognition, affect, and intention and an understanding of the concept should help practitioners in improving development environments with respect to developers' needs, perceptions and feelings [7].

In this paper we address DX in terms of the experienced state of flow, intrinsic motivation (IM) and UX. Our goal is to determine the core concepts and predictors of DX related to IDE usage in order enable improvement of IDEs to improve developers' IM towards their work and their ability to experience flow (deep, focused, rewarding concentration) during their work. Our assumption is that these factors both make developers' work more enjoyable and increase their productivity. To this end, we conducted a survey of developers' experiences of software development using a particular IDE, Qt Creator. We used the Short Dispositional Flow Scale (SDFS-2) [8], parts of Intrinsic Motivation Inventory (IMI) [9], and a UX scale consisting of the Short AttrakDiff-2 [10] and our own DEXI scale. We ran multiple linear regression analyses to investigate whether these scales can significantly predict developers' ratings of overall UX (OUX) and the IDE's ability to fulfill their needs (need fulfillment score, NFS). Moreover, we address the impact of perceived choice of Qt Creator since it often is the employer who decides which tools are used. Finally, we present best qualities and areas for improvement in the IDE as assessed by the respondents. Our contributions include increased understanding of developers as users, and core UX concepts related to DX and developers' needs related to IDEs.

The rest of the paper is structured as follows: the next section presents the background and related work followed by the research methodology. Then the results section presents the linear regression analyses on the scales' ability to predict OUX and NFS, the impact of perceived choice on DX, and the core qualities of IDEs. We discuss our results and threats to validity. Finally, we present concluding remarks.

2   Background

Motivation and flow. One of the current influential theories of motivation is self-

determination theory developed by Deci and Ryan [11]. They distinguish between intrinsic (IM) and extrinsic motivations (EM). IM refers to engaging in a task because of it is inherently pleasurable and satisfying, whereas EM refers to engaging in a task because of its outcomes, the task is used as a means to lead to the outcome [11]. In contrast, flow refers to a state of concentration so focused that it amounts to absolute absorption in an activity [12]. Applicable to both work and leisure [13], flow builds on IM and internal reward over the achievement rather than on external goal or recognition. Its effect can be characterized as being totally focused on a particular task at hand, so that the person becomes fully immersed in a feeling of energized focus, full involvement, and enjoyment in the process of the activity. While immersed, three conditions have to be met to achieve a flow state [14]: 1) One must be involved in an activity with a clear set of *goals and progress*; 2) The task at hand must have clear and immediate *feedback*; 3) One must have a good balance between the perceived *challenges* of the task at hand and their own perceived *skills*, so that there can be confidence in one's ability to complete the task at hand.

User Experience. Commonly, UX is understood as subjective, context-dependent, and dynamic [15]. It is affected by user's expectations, needs and motivation, as well as system characteristics such as purpose and functionality, and the context of use including physical, organizational and psychological aspects [6]. The hedonic-pragmatic model of UX divides user experience into a hedonic or non-utilitarian dimension and a pragmatic or instrumental dimension [16]. Hassenzahl [16] further divides the hedonic into two sub-dimensions of identification and stimulation while the pragmatic/instrumental dimension relates to usability and usefulness.

Software Engineering. The core of software development is writing program code that constructs the running software; this demands the ability to concentrate and work alone for many hours [17]. Moreover, programming work requires a logical mind and the ability to pay attention to details [17]. Developers need to be analytical, capable of making decisions, independent, creative, tenacious, and be able to tolerate stress [18]. Although programmers tend to be introverted, sensing, and thinking [17], social skills are crucial in their work: developers' interpersonal and communication skills have been considered even more important than their technical skills for project success [19]. Due to the complex nature of software development, specialized tools are used. One of the most general tools that are used to create programs is an IDE, which offers numerous features. A sophisticated IDE, extended with plugins, may manage dependencies among different packages and modules, control complex builds, and provide linking to other tools such as requirements management or test environment. Thus, the IDE acts as an interface between the developer and the computing infrastructure that is needed for creating, configuring, and managing complex applications as well as their source code and build environment.

IDEs have two main productive goals: increasing developer speed and reducing the number of errors made by developers [1]. As IDEs are a main tool in software development, they also play a major role in making developers productive and comfortable in their work. Moreover, IDEs are a key aspect in developer experience (DX), a concept that encompasses developer's perceptions of their work and phenomena related to it such as cognitive, motivational, affective, and social aspects. For example, memory overload is a limiting factor for programmers, especially for beginners who have not yet developed strategies to relieve it [20]. Modern

development environments provide many aids to programmers, but the same challenges are still present. Cognitive factors also concern larger structures in software development, such as methods and processes, but research on this aspect is scarce.

Developer experience and motivation. The concept of DX aims to provide an intuitive abstraction of the huge variety and quantity of human factors that influence developers and the outcomes of SE [7]. While UX considers the context of use of a system, DX considers the context of software development, including aspects beyond software tools, such as development processes, modeling methods, and other means of structuring SE tasks. Some of these aspects are embedded in tools such as IDEs while others are part of organizational practices. The software development activity and environment differ in significant ways from other information-intensive activities and environments. For example, software development requires a nested understanding: developers use software to build further software that is to be used by users to accomplish their particular tasks. Also, developers frequently configure and extend their tools, in effect continuously developing both the development environment and the end product at the same time.

Developer motivation is as another important factor in SE. The majority of studies on motivation in SE report that developers are distinct from other occupational groups with respect to motivation [3]. "The work itself" is the most commonly cited motivator, but there is a lack of detail regarding what aspects of the work is motivating, how motivational processes occur, and the outcomes of motivating developers [3, 21]. Investigations also show the importance of considering affective aspects of SE. The presence and variation of developers' emotions over time has been documented [22]. Programming is influenced by mood [4], and happiness has been found to have productivity benefits [2]. This underlines the importance of considering affective aspects both for purposes of well-being and outcomes.

3 Method

Our research goal is to increase understanding of DX. We aim to clarify how flow, IM, and UX are intertwined in software development. This will enable improvement of development tools to better support developers' ability to experience flow in their work and to enhance developers' IM towards their work. Our hypothesis is that these factors make developers' work more enjoyable and increases their productivity.

In this paper we address the following research questions:

1. Can we predict the developers' overall UX with the IDE and its ability to fulfill their needs from their sense of flow in their work and their IM?
2. Can we predict the developers' overall UX with the IDE and its ability to fulfill their needs from their assessment of the practical, hedonic, and general quality of the IDE?
3. What kind of impact does perceived choice have on developers' assessments?
4. How do developers describe the best qualities of the IDE and those that need improvement in relation to UX vocabulary?

We conducted a survey measuring developers' self-reported experiences of software

development activities when using Qt Creator, a cross-platform IDE including a code editor, graphical user interface editor, compiler, visual debugger, and version control. Our survey consisted of the following three scales: 1) the Short Dispositional Flow State Scale (SDFS-2) [8] used in its entirety, 2) parts of the Intrinsic Motivation Inventory (IMI) [9] including questions related to interest/enjoyment, perceived competence, effort/importance, and perceived choice, and 3) a UX scale consisting of the short version of the AttrakDiff-2 (SAD-2) [10] used in its entirety and our own Developer Experience Scale (DEXI). The scales, except DEXI, were selected because they are widely used and validated. They are also short enough to be combined in a single survey. DEXI was created to address characteristics of software development.

Respondents also rated the overall UX (OUX) of the IDE and its ability to fulfill their needs (NFS) as follows: 1) OUX: "How would you rate the overall user experience of Qt Creator?" (from 1 = bad to 7 = good). 2) NFS: "How well does Qt Creator respond to your needs?" (ranging from 1 = not at all to 7 =completely). We also asked respondents to describe the qualities of the IDE on two open-ended questions: 1) "In your opinion, what are the best qualities of Qt Creator?" and 2) "How could Qt Creator better support your development work?". Finally, we collected demographic information, including the country they were based in, age, experience of software development (in years), experience of using Qt Creator, developer role, size of the organization they are working for, their operating system and target platforms, and used license type of Qt Creator.

Dispositional Flow State Scale (SDFS-2). We measured the frequency with which developers experience different dimensions of flow during software development activities with Qt Creator using the Short Dispositional Flow State Scale (SDFS-2) [8], with Likert items (from 1 = never to 7 = always). The SDFS-2 measures nine dimensions of flow, each with one item (Table 1). In addition to the SDFS-2 items, an additional item measured the experience of frustration: "I feel frustrated".

Intrinsic Motivation Inventory (IMI). Since the original IMI is long and redundant, we used a commonly used shortened version [23] (Table 2) with selected items from the following IMI subscales: interest/enjoyment (the actual self-report measure of IM), perceived competence, effort/importance, and perceived choice. Framing of the question and assessment scale was according to the IMI. Thus, the question was as follows: "*For each of the following statements, please indicate how true it is for you, using the following scale*" (from 1 = not at all true to 7 = very true).

UX scales. We used the short version of AttrakDiff-2 (SAD-2) [10]. It contains four items (word-pairs) for both practical (PQ) and hedonic quality (HQ) of UX, and one each for measuring goodness and beauty (general UX quality, GQ). In addition, we formed our own DEXI scale for measuring additional aspects of UX. We selected DEXI items from the following sources: AttrakDiff [16], the dataset of a meta-study of often used UX items [24], and concepts that have been used to describe DX [7]. We used the structure and wording of AttrakDiff in DEXI. We aimed at construct a scale that would be relevant to software development. We selected 5 items (PQ1 – PQ5) measuring pragmatic UX quality (difficult/easy to learn; inflexible/flexible; limited/extensive; uninformative/informative; inefficient/efficient) and 6 items (HQ1 – HQ6) measuring hedonic (non-utilitarian) quality (discouraging/motivating; suppresses/promotes creativity; decreases/increases respect; unenjoyable/enjoyable; separates me from others/brings me closer to others; uninvolving/engaging). One item

Table 1. SDFS-2 scale. Dimensions of state of flow and related survey items**.** [8]

Flow dimensions	SDFS-2 item
Challenge-skill balance	I feel I am competent enough to meet the high demands of the situation
Action awareness	I do things spontaneously and automatically without having to think
Clear goals	I have a strong sense of what I want to do
Unambiguous feedback	I have a good idea while I am performing about how well I am doing
Concentration on task	I am completely focused on the task at hand
Sense of control	I have a feeling of total control
Loss of self-consciousness	I am not worried about what others may be thinking of me
Transformation of time	The way time passes seems to be different from normal
Autotelic experience	The experience is extremely rewarding

Table 2. Selected Subscales and survey items of IMI. [9]

Subscale	Survey item
Interest/enjoyment	I enjoy software development work very much I think software development is a boring activity I enjoy using Qt Creator very much
Perceived competence	I am satisfied with my performance at software development I am pretty skilled in software development I am pretty skilled in using Qt Creator
Effort/Importance	It is important to me to do well in software development
Perceived choice	I use Qt Creator because I have no choice

(GQ-1) measured general quality (not recommendable/recommendable).

Procedure. A web survey was organized with the Qt Company, the provider of Qt Creator. The survey had a front page presenting informed consent statements adopted from World Health Organization's template for qualitative studies [25]. We instructed only those who had been using the IDE to respond, and to respond only once. A global online developer community and Twitter were used to target users of the IDE. Although the survey was distributed globally, the main interest of the IDE provider was in Middle European market. The survey was available for the respondents for four weeks. Participants' median completion time was 9 minutes (M = 17, SD = 31).

Participants. Participants were developers using Qt Creator in their work. In total, 57 developers responded from 25 different countries. Respondents' countries were: France: 8; Germany: 7; Italy: 5; Norway: 4; Austria, Australia, Finland, Switzerland, and United States: 3; Russia and Sweden: 2; Algeria, Andorra, Bulgaria, Brazil, Belarus, Czech Republic, Denmark, Indonesia, India, Iran, Poland, Slovenia, Ukraine, and United Kingdom: 1. The average age of respondents was 35 years (SD: 10). Respondents had on average 8 years (SD: 5) of working experience in software development. 86.0% of respondents had been using Qt Creator for over a year, 12.3% for over a month but less than a year, and 1.8% had used it several times. 42.1% of the respondents considered themselves as front-end developers, 21.1% as back-end developers, 19.3% as architects, and 17.5% considered themselves as other types including either a combination of these roles, or hobbyist, teacher, or researcher. Considering the size of organization where they worked, 22.8% were individual

developers, 19.3% worked for micro businesses (employing ten people or less), 19.3% worked for small companies (over ten but less than hundred employees), 24.6% worked for middle-sized companies (100-1000 employees), and 14.0% worked for large enterprises employing more than 1000 people. Approximately half the respondents (49.1%) used Linux as their primary development platform, while 28.1% used MS Windows and 21.1% OS X. Most of the respondents (91.2%) developed desktop software, 40.3% developed mobile software, and 25.6% developed embedded software (multiple choices were possible on this question). Free software licenses were used by 75.4% of respondents, while the rest (24.6%) used commercial licenses. Demographic variables were not significant predictors of any of the studied variables.

4   Results

Predicting Overall UX and Needs Assessment from Sense of Flow and Intrinsic Motivation. Four multiple linear regressions investigated whether the items of the measures of flow (SDSF-2) and intrinsic motivation (IMI) significantly predicted the respondents' ratings of overall UX (OUX) with the IDE and its ability to meet their needs (NFS).

OUX could be predicted significantly from the SDSF-2 scale (see Table 3). However, only one of the SDSF-2 items was a significant individual predictor, the autotelic experience item, ("the experience is extremely rewarding"). OUX could also be predicted from the IMI scale with two of the items being significant individual predictors, both interest/enjoyment items: "I enjoy software development work very much" and "I enjoy using Qt Creator very much".

Need fulfillment (NFS) could also be predicted from the SDSF-2. Two individual items were significant individual predictors: the autotelic experience item ("the experience is extremely rewarding") and the sense of control item ("I have a feeling of total control. NFS could also be predicted from the IMI scale. Two items were significant predictors: "I think software development work is a boring activity" and "I enjoy using Qt Creator very much".

Predicting Overall UX and Needs Assessment from Practical, Hedonic, and General UX Qualities. When comparing the assessments of quality types, general quality had the highest mean assessment score while the hedonic had the lowest. The difference is statistically significant: The null hypothesis that "the median difference between measurements of PQ, GQ, and HQ, pairwise, is zero" was rejected as follows: between measurements of practical and general quality Z = -3.333, p < .01. between measurements of hedonic and practical quality Z = -4.171, p < .001; between hedonic and general quality Z = -5.590, p < .001. Thus, the GQ assessment was significantly higher than assessment of PQ and HQ.

The overall UX rating, OUX, could be predicted significantly from PQ, GQ, and HQ together (see Table 3). However, only HQ was on the borderline of being a significant predictor.

NFS could be predicted significantly from PQ, GQ, and HQ together (see Table 3). However, none of the quality types were significant predictors.

Table 3 Survey scales/items that significantly predicted OUX and NFS scores. Legend: "UX scales" refers to SAD-2 and DEXI together.

Overall UX (OUX)	Needs Assessment (NFS)
SDFS-2 (F = 3.44, df = 10, p < 0.005)	**SDFS-2 (F = 4.48, df = 10, 46, p < 0.001)**
The experience is extremely rewarding (t = 2.85, p < 0.01)	The experience is extremely rewarding (t = 2.27, p < 0.05)
	I have a feeling of total control (t = 2.80, p < 0.01)
IMI (F = 6.72, df = 8, 48, p < 0.001)	**IMI (F = 9.04, df = 7, 49, p < 0.001)**
I enjoy software development work very much (t = 2.29, p < 0.05)	I think software development work is a boring activity (t = 2.26, p < 0.05)
I enjoy using Qt Creator very much (t = 5.01, p < 0.001)	I enjoy using Qt Creator very much (t = 5.62, p < 0.001)
UX scales (F = 9.80, df = 3, p < 0.001)	**UX scales (F = 6.24, df = 3, p < .01)**
General quality (GQ) (t = .129, n.s.)	General quality (GQ) (t = .48, n.s.)
Hedonic quality (HQ) (t = 2.00, p = 0.05)	Hedonic quality (HQ) (t = .16, n.s.)
Practical quality (PQ) (t = .556, n.s.)	Practical quality (PQ) (t = 1.49, n.s.)

Perceived Choice. The use of work-related tools can be mandatory since often the employer is the one who selects the tools to be used [24]. We measured perceived choice of use with the IMI scale question "I use Qt Creator because I have no choice". It had significant negative correlation with both OUX and NFS (r = -.380, and r = -.370, respectively, p < .01 for both). Thus, developers who perceived high level of choice in use of the IDE assessed OUX and NFS higher than developers who perceived their use of the IDE as mandatory. In addition, developers with low perceived choice enjoyed using the IDE less; there was a moderate negative correlation between perceived choice and the IMI item "I enjoy using Qt Creator very much" (r = -.534, p < .001). Enjoyment on the IMI scale measures motivation and thus we can conclude that developers with lower perceived choice were less motivated towards using the IDE compared to those with high perceived choice. Developers with low perceived choice also felt frustrated more often (r = .519, p < .001). Finally, there was a significant negative correlation between the perceived choice and challenge-skill balance in using the IDE (r = -.296, p < .05)

Since developers with low perceived choice enjoyed using the IDE less than others, we also address here correlations between the motivation towards using the IDE ("I enjoy using Qt Creator very much") and other measures. There was a significant correlation between motivation towards using the IDE and both NFS and OUX ratings (r = .682, and r = .639, respectively, p < 0.001 for both). On the SDFS-2 items, developers who enjoyed using the IDE also experienced a significantly higher sense of control (r = .548, p < .001) and considered the experience significantly more rewarding (r = .539, p < .001). They also felt frustration significantly less (r = -.498, p < .001). In addition, developers who enjoyed using the IDE considered themselves significantly more skilled in using the tool (r = .400, p < .01).

Best Qualities of the IDE and Opportunities for Improvement. Respondents considered efficiency, flexibility, informativeness and intuitiveness the best qualities of Qt Creator and flexibility, informativeness, and reliability required improvement the most (see Figure 1) [26]. Thus, although the IDE was considered both flexible and informative, these were also areas that required improvement the most. It might indicate that these concepts are focal for an IDE. In contrast, developers considered efficiency as one of the best qualities most often (38% of respondents mentioned it),

and it rarely was considered as subject for improvement. However, reliability was rarely mentioned as good quality, whereas 36% of the respondents considered Qt Creator should be more reliable, mostly in terms of stability and faultlessness.

The category of efficiency includes mainly items related to the IDE being fast and efficient to use. Flexibility is the ability of an IDE to respond to developers' needs such as being customizable, scalable, extensive, compatible, or complete. Informativeness was most often related to the presentation of code and text editors, for instance, to intelligent code completion and text highlighting. It was also related to the quality and presentation of information in different built-in tools such as the debugger. Reliability addresses the robustness, stability, faultlessness, and recoverability of the IDE. Intuitiveness is related to the IDE being simple, intuitive, understandable, intelligent, and sensible. Clarity includes such items as clean, unbloated, uncluttered, light, and well-structured. Value was described with the following words: good, great, awesome, best, and free. Aesthetic design was related to the screen layout and the outlook and visual design of the IDE. Empowerment means the ability of the IDE to support developers' work and respect the variety of tasks they have. Finally, approachability was mentioned as creating friendly atmosphere and making the developer to feel at home.

Of the UX qualities identified efficiency, effectiveness and learnability are productivity factors. Ease of use and intuitiveness relate to interaction quality whereas informativeness and reliability relate to information and system quality, respectively. Empowerment, approachability, and aesthetic design are hedonic qualities related to stimulation, appeal and aesthetic quality, respectively. [16]
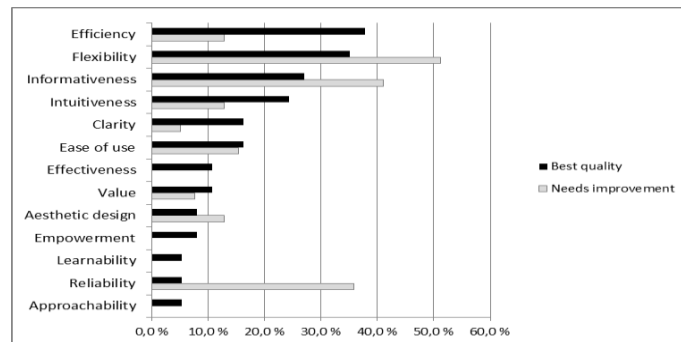

Fig. 1. Best qualities of Qt Creator and those that need improvement as reported by respondents. Percentage of respondents (N = 45) per category.

5   Discussion

Based on the responses of 57 developers from 25 countries, who responded to the survey, research questions are answered as follows:
1. Can we predict the developers' overall UX with the IDE and its ability to fulfill their needs from their sense of flow in their work and their intrinsic motivation?

We found that autotelic experience and intrinsic motivation (IM) towards both software development and the IDE were significant predictors of developers' overall

UX. Need fulfillment could be predicted from the aforementioned autotelic experience item and sense of control and from intrinsic motivation.

2. Can we predict the developers' overall UX with the IDE and its ability to fulfill their needs from their assessment of the practical, hedonic, and general quality of the IDE?

We found that practical, hedonic, and general quality together were significant predictors of overall UX. None of the qualities alone significantly predicted overall UX. However, hedonic quality was on the borderline of being a significant predictor ($p = .05$). Practical, hedonic, and general quality together were also significant predictors of need fulfillment. However, none of the qualities alone was a significant predictor of need fulfillment.

3. What kind of impact does perceived choice have on developers' assessments?

We found that perceived choice had a significant negative correlation with both overall UX and need fulfillment. It also had a significant negative correlation with intrinsic motivation towards using the IDE and a significant positive correlation with the frequency of feeling frustrated. Finally, perceived choice had significant negative correlation with challenge-skill balance considering using the IDE.

4. How do developers describe the best qualities of the IDE and those that need improvement in relation to UX vocabulary?

We found that developers considered efficiency, flexibility, informativeness and intuitiveness the best qualities of the IDE whereas flexibility, informativeness, and reliability required improvement the most. Developers described qualities of the IDE with regard to the following practical qualities: productivity and interaction and information quality. System quality represented general quality in developers' descriptions and hedonic qualities were related to stimulation, appeal, and aesthetic quality.

Reflections on Concept of Developer Experience. Fagerholm et al.'s [7] framework of DX addresses the concept in terms of factors related to the perception of development infrastructure, feelings towards work, and the value of the developer's contribution. They relate cognition to the perception of infrastructure, affect with feelings towards the work, and intention (conation) with the value of contribution. In our study, the IDE itself represents the development infrastructure and cognition, affect and intention were addressed with regard to it. Our results indicate that developers also address the infrastructure via intention and affect. Their IM was towards both the use of the IDE and the development work. Some developers described the best qualities of the IDE with affection. In addition, the overall UX assessment of developers seemed to be affected more by the hedonic than pragmatic quality of the IDE since the mean value of the hedonic aspect of UX (HQ) was on the borderline of being a significant predictor of overall UX ($p = 0.05$) whereas the overall UX could not be predicted from the practical or general quality. Thus, our results suggest that Fagerholm et al. over emphasized the cognitive approach of developers towards the development infrastructure. Moreover, the developers' intrinsic motivation also seems to focus on using the IDE while Fagerholm et al. associate it with the developer's contribution. However, the IDE is used to create a contribution and thus our study cannot separate motivation towards development work itself and motivation towards the software under development.

Threats to validity. We studied only one IDE and thus some of our results might

be specific to that. We also had a relatively limited number of respondents (57). In the future, other IDEs and development work in general should be studied. We did not control multiple answering of the questionnaire but asked developers to respond only once. However, we consider the likelihood of multiple answering small. Since the invitation to participate was sent to an online developer community and Twitter, only developers who use those channels could participate, thus limiting the population of developers we sampled from. However, we found no significant difference between developers who were recruited via the online community and those contacted via Twitter. In addition, demographic variables were not significant predictors of any of the studied variables.

5   Conclusions

We have presented results of software developers' sense of flow, their intrinsic motivation (IM) and developer experience (DX) in the context of software engineering. We conducted a survey study on developers using Qt Creator as their development environment. We aimed to clarify how flow, IM, and UX are intertwined in software development. Our final goal is the improvement of development tools to better support developers' ability to experience flow – deep, focused, rewarding concentration in their work – and to enhance developers' IM towards their work. Our hypothesis was that these factors make developers' work more enjoyable and increase their productivity. Our results suggest that IM and sense of flow are significant predictors of DX. IM towards both development work itself and using the IDE significantly predicted DX. Moreover, we found perceived choice of use a significant predictor of both developers' assessment of UX and need fulfillment. Perceived choice also affects developers' IM towards using the IDE and their sense of frustration during development tasks. Thus, developers' motivation is affected both by tool selection and qualities of development tools. Further studies are needed to address their impact on developers' productivity.

Our work examined DX mainly in relation to the key development tool, the IDE. Considering the central role of the IDE in developers' daily activities, it can be expected that results considering developers' experiences while using the IDE play a prominent role also for DX in general. In relation to the concept of DX, our paper contributes to increased understanding of its key factors and its relation to UX, IM, and the flow state experience.

References

1. Muşlu, K., Brun, Y., Holmes, R., Ernst, M. D., & Notkin, D. (2012). Speculative analysis of integrated development environment recommendations. ACM SIGPLAN Notices, 47(10), 669-682.
2. Graziotin D, Wang X, Abrahamsson P. (2014). Happy software developers solve problems better: psychological measurements in empirical software engineering. PeerJ 2(1):e289
3. Beecham, S., Baddoo, N., Hall, T., Robinson, H., and Sharp, H. (2008). Motivation in

software engineering: A systematic literature review. IST, 50, 860–878.

4. Khan, I. A., Brinkman, W.-P., and Hierons, R. M. (2011). Do moods affect programmers' debug performance? Cognition, Technology & Work, 13(4):245–258.

5. Graziotin, D., Wang, X., Abrahamsson, P., Software Developers, Moods, Emotions, and Performance, IEEE Software, vol. 31, no. 4, pp. 24-27, July-Aug., 2014.

6. Hassenzahl, M., Tractinsky, N. User experience - A research agenda. BIT 25(2), 91-97. 2006

7. Fagerholm, F., & Münch, J. (2012). Developer experience: Concept and definition. Proc. International Conference on Software and System Process (pp. 73-77). IEEE Press.

8. Jackson, S. A., Martin, A. J., & Eklund, R. C. (2008). Long and short measures of flow: The construct validity of the FSS-2, DFS-2, and new brief counterparts. JSEP, 30(5), 561.

9. Ryan, R. M. (1982). Control and information in the intrapersonal sphere: An extension of cognitive evaluation theory. Journal of Personality and Social Psychology, 43, 450-461.

10. Hassenzahl, M., Diefenbach, S., & Göritz, A. (2010). Needs, affect, and interactive products–Facets of user experience. Interacting with Computers, 22(5), 353-362.

11. Deci, E., & Ryan, R. M. (2012). Self-determination theory. Handbook of theories of social psychology, Los Angeles, SAGE. ISBN: 9780857029607

12. Csikszentmihalyi, M. Flow: The psychology of optimal experience. Vol. 41. New York: HarperPerennial, 1991.

13. Csikszentmihalyi, M; LeFevre, J. Optimal experience in work and leisure. Journal of Personality and Social Psychology, Vol 56(5), May 1989, 815-822.

14. Csikszentmihalyi, M.; Abuhamdeh, S. & Nakamura, J. (2005), "Flow", in Elliot, A., Handbook of Competence and Motivation, New York: The Guilford Press, pp. 598–698.

15. Law, E. L.-C., Roto, V., Hassenzahl, M., Vermeeren, A. P.O.S., & Kort, J. (2009). Understanding, scoping and defining user experience: a survey approach. Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI '09). ACM, 719-728.

16. Hassenzahl, M. The Interplay of Beauty, Goodness and Usability in Interactive Products. Proc. HCI. Lawrence Erlbaum Associates, 19, 4 (2004), 319-349.

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

18. Acuna, S. T., Juristo, N., & Moreno, A. M. (2006). Emphasizing human capabilities in software development. Software, IEEE, 23(2), 94-101.

19. Lee, D., Trauth, E. and Farwell, D. 1995 Critical skills and knowledge requirements of IS professionals: A joint academic/industry investigation. MIS Quart., 19, 3, 313-340.

20. Anderson, J. R., & Jeffries, R. (1985). Novice LISP Errors: Undetected Losses of Information from Working Memory. Human-Computer Interaction, 1, 107-131.

21. Franca, A. C. C., Gouveia, T. B., Santos, P. C. F., Santana, C. A., and da Silva, F. Q. B. (2011). Motivation in software engineering: A systematic review update. Proc. Evaluation and Assessment in Software Engineering (EASE), pages 154–163.

22. Shaw, T. (2004). The Emotions of Systems Developers: An Empirical Study of Affective Events Theory. Proc. Computer Personnel Research: Careers, Culture, and Ethics in a Networked Environment, SIGMIS CPR '04, pp. 124–126. ACM

23. McAuley, E., Duncan, T., & Tammen, V. V. (1989). Psychometric properties of the Intrinsic Motivation Inventory in a competitive sport setting: A confirmatory factor analysis. Research Quarterly for Exercise and Sport, 60, 48-58.

24. Sundberg, H.-R. (2015). The importance of user experience related factors in new product development – Comparing the views of designers and users of industrial products. 23rd Nordic Academy of Management Conference, 12-14 August 2015, Copenhagen, Denmark

25. World Health Organization, Informed consent form template for qualitative studies. Available at: http://www.who.int/rpc/research_ethics/informed_consent/enTools.

26. Kuusinen, K. (2015). Software Developers as Users: Developer Experience of a Cross-Platform Integrated Development Environment. Product-Focused Software Process Improvement (PROFES'15), 546-552. Springer International Publishing.