Impact of Developer Experience in the outcome of Software Projects

Anders Nylund

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.
Espoo TBA

Supervisor

Prof. Pirjo Professor

Advisor

Dr Alan Advisor

Aalto University
School of Science

Aalto University
School of Science

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the master's thesis

Author Anders Nylund

Title Impact of Developer Experience in the outcome of Software Projects

Degree programme Computer, Communication and Information Sciences

Major Software and Service Engineering **Code of major** SCI3043

Supervisor Prof. Pirjo Professor

Advisor Dr Alan Advisor

Date TBA **Number of pages** 20 **Language** English

Abstract

Your abstract in English. Keep the abstract short. The abstract explains your research topic, the methods you have used, and the results you obtained.

The abstract text of this thesis is written on the readable abstract page as well as into the pdf file's metadata via the \thesisabstract macro (see above). Write here the text that goes onto the readable abstract page. You can have special characters, linebreaks, and paragraphs here. Otherwise, this abstract text must be identical to the metadata abstract text.

If your abstract does not contain special characters and it does not require paragraphs, you may take advantage of the abstracttext macro (see the comment below).

Keywords Developer Experience, Software Projects

Aalto University
School of Science

Aalto-universitetet, PB 11000, 00076 AALTO
www.aalto.fi
Sammandrag av diplomarbetet

Författare Anders Nylund

Titel Impact of Developer Experience in the outcome of Software Projects

Utbildningsprogram Computer, Communication and Information Sciences

Huvudämne Software and Service Engineering **Huvudämnets kod** SCI3043

Övervakare Prof. Pirjo Professori

Handledare TkD Alan Advisor

Datum TBA **Sidantal** 20 **Språk** Engelska

Sammandrag

Sammandrag på svenska. Try to keep the abstract short. Abstract explains your research topic, the methods you have used, and the results you obtained.

Nyckelord Nyckelord på svenska, temperatur

Preface

I want to thank Professor Pirjo Professori and my instructor Dr Alan Advisor for their good and poor guidance.

Otaniemi, Date to te announced

Anders Nylund

Contents

Thesis dictionary

DX	Developer Experience
UX	User Experience
IDE	Integrated Development Environment

1 Introduction

1.1 Motivation

Currently sections Introduction and Motivation are intertwined. It should be made more clear that what should go where.

A common theme in the thesis could be answering the question "Is Developer Experience something worth investing on in software companies?". I.e. is there any apparent reward in putting money in better tools or learning new technologies?

After all learning new technologies is not about solving new problems, but it's about solving the same problems more efficiently, faster, easier i.e. with a better developer experience.

Developer Experience is what is felt by the developer while trying to achieve a goal i.e. completing a project

Personal opinion: For me developer experience means having a low friction and easy setup with my own development environment. I like to have an environment that is lightweight, fast, and easy to use. It should have a short cycle of feedback i.e. when making a change to the source code it should be immediately reflected in the output. This might be the reason why I like to develop for the web, as the tools are often quick and have a fast feedback cycle. The environment should perform tasks automatically as building, reporting errors.

React and the tools supporting it are a great examples of excellent developer experience. The tools are intuitive and guide the developer in making the right things.

Developers in software projects are in a crucial role when considering the success of software projects.

The same way as User Experience (UX) is considering the user of a system or tool, Developer Experience (DX) can be seen as the experience that developers have as users of a system. Here the system however includes the tools, frameworks, processes that the developer is the user of when developing software.

Developer experience has not been studied that much previously. There has been a doctoral thesis published about the topic recently [2]. In this thesis

DX has been studied previously, but research on it is still lacking the connection to practical applications. This is one the biggest motivators for this thesis, as the topic is novel and there is huge potential in improving software development processes, and thereby also potential to improve the outcome of the projects.

There is possibly huge value that can be gained from studying Developer Experience and learning about how it works. A better experience can help organizations

Currently a quick search with the keyword *"Developer Experience"* on google gives as a result mostly articles on how framework and library authors should consider their users (developers) experience with using the product (tool, library, framework). However, DX is something more and includes also the feelings and perceptions of the developers. In some research the term *Developer Experience* with the abbreviation of DE^x is used, and in some other research the term *Programmer eXperience* and abbreviation *PX* is used [3] [7].

It is also apparent that most results when searching with the term *Developer Experience* gives results about the experience level of a developer, and not the experience of being part

1.2 Research questions and problem

The research problem is finding out *How the developer experience in software projects can affect the outcome.*

The debate throughout the thesis has to be something that makes the reader interested in the topic and engages the reader.

This helps to find the argument of each article and paper that is read for this thesis. It helps the author of this thesis to take a stand when writing some statements. The meaning is not to create some kind of truth that has to be followed. The constant debate throughout the thesis helps to put things in perspective.

One example of debate is "Is Developer Experience something worth investing in?".

Table 1: The research questions

RQ 1	How is Developer Experience defined in software projects?
RQ 2	What aspects of Developer Experience are currently being considered in software projects? What aspects of Developer Experience do developers see as valuable?
RQ 3	Can the results of software projects be improved by investing in a better developer experience?

1.2.1 Alternative research problems:

- "How Developer Experience affects the productivity of developers in software projects"?
- "How the cognitive Developer Experience can affect the outcome of Software Projects". This would allow to restrict the scope of the thesis significantly, as the cognitive Developer Experience takes only into account the *"technical"* parts e.g. Platform, techniques, process, skill, procedures (How developers perceive the development infrastructure?) [3]

There could be some hypotheses that will be tested in the thesis.

1.3 Scope and focus

1.4 Structure of the thesis

This will be finalized later

1. Background and literature review
2. ...
3. ...

2 Background and literature review

Keywords to search background and literature material with:

1. Developer Experience
2. Programmer Experience
3. Happy Developer
4. Unhappy Developer

https://insights.stackoverflow.com/ could be an interesting source of basic facts about programmers around the world, what technologies they are using, what they love and what they dread

This section includes the background and literature review of the topic. The background of the topic should be covered equally from all points of view.

Create a much clearer and better foundation of what software development is, why it is complex etc.

A software project is a project where a group of people share a common goal what can for example be to create a product or service. In a software project there is a developer or multiple developers that have the responsibility of implementing the technical product itself. The developers are the ones writing the executable source code for the program or service, so that it can by it's functions and features achieve the requirements set to it.

Software developers are in a crucial role when considering the success of a software project.

In this thesis the term *Developer Experience* is examined and studied. Developer experience is a rather new concept in the scene of software engineering.

Developer experience can be divided into three different sub areas – cognitive (How do developers perceive the development infrastructure?), affective (How do developers feel about their work?), and conative (How do developers see the value of their contribution?) [3].

2.1 Intrinsic Motivation

2.2 Programmer Experience

Programmer Experience (PX) can be defined as *The result of the intrinsic motivations and perceptions of programmers about the use of development artifacts* [7].

A programmer can be seen as person who gives exact instructions on how a program should behave and function.

2.3 Developer Experience

Developer Experience (DX) is a bigger construct than PX. DX includes also the motivation of developers, and not only the artefacts like the programming environments [7]. Developer Experience is considering more of the social

A developer is a person with a bigger responsibility than a programmer. If a programmer is following instructions, requirements, and guidelines, the developer is also finding out what the instructions, requirements and guidelines should be (find other source than https://devskiller.com/programmer-vs-developer/). Therefore DX is also considering more of the surrounding context than what PX is considering.

The Developer Experience can be divided into 2 different environments, a social and a technical environment [2]. This thesis might focus more on the technical environment.

A common terminology and understanding of Developer Experience is part examining a project's Developer Experience.

2.4 Performance Alignment Work

2.5 Happiness of developers

Happiness of developers have been reported have direct consequences to the themselves, process and the product [4]

2.6 Selection of tools

Perceived choice is a perception of that the choice has already been made. Selecting tools in software development projects is in a crucial role, as it can significantly improve the Developer Experience in software projects.

One study of Integrated Development Environment (IDE), and how it is connected with state of flow, intrinsic motivation, and user experience revealed that if the developers have a high perception of choice, the also are overall more satisfied with the tools [6]. They also concluded that if the selected tools are selected without their input, (they perceive it chosen already), the developers will have a worse developer experience with it, as e.g. their frustration with the tool will be more common.

There has been a study on the Developer Experience of IDEs [5]. However, the study concentrated on the UX of the selected IDE that was studied.

When selecting an IDE it is also important to consider what the other developers in the team or organization is using or what other would prefer to use.

There can be situations when two different developers use a different IDE, and therefore also the experience can be completely different between them. At the most extreme the 2 IDEs are not compatible with each other as their files related to the project are different. An example of this is Eclipse and IntelliJ IDEA as Java IDEs.

In a study of IDEs [5], the survey in the study produced answers that were most pragmatic, but not hedonic. This could show that most of the developers are practical, and not feeling based. This has also been proven [1]. This might also be a reason why Developer Experience has not gotten that much attention yet, as big

part of people in software engineering are *"Introverts"*. Software engineers are also more logical thinkers than feeling based. As Developer Experience is focusing on the feelings and subjective opinions about things, it might be a difficult topic to research.

2.7 Flow state

Flow state is something that many developers want to achieve. For some developers it is really difficult to focus if there are external things that disturb them like sound or something similar. Also, people coming and asking questions might disturb or interrupt the flow state. Therefore many developers are now also trying out remote work where they are not co-located.

[8] studied how an IDE worked in a collaborative environment and it's developer experience.

3 Research material and methods

What material will be used in the research and what methods/methodologies will be used to study the problem. What kind of approach to research will be used in the thesis.

The developer experience can be both short term impulsive, or related to one event in software development, but it can also be a long term experience over a period of time [2]. The research in this thesis will use a longer time-frame of developer experience.

4 Results

Answer the research questions and problem.

4.1 Validity of results

Tässä osassa on syytä myös arvioida tutkimustulosten luotettavuutta. Jos tutkimustulosten merkitystä arvioidaan »Tarkastelu»-osassa, voi luotettavuuden arviointi olla myös siellä.

5 Summary

6 Conclusions

7 Possible references

Development tools

1. Murphy,G.C.,Kersten,M.,Findlater,L.:HowareJavasoftwaredevelopersusingtheElipseIDE? Softw. IEEE 23(4), 76–83 (2006)

2. 17. Muslu,K.,Brun,Y.,Holmes,R.,Ernst,M.D.,Notkin,D.:Speculativeanalysisofintegrated development environment recommendations. ACM SIGPLAN Not. 47(10), 669–682 (2012)

3. Kersten, M., Murphy, G.C.: Using task context to improve programmer productivity. In: Proceedings of the 14th ACM SIGSOFT International Symposium on Foundations of Software Engineering (SIGSOFT 2006/FSE-14), pp. 1–11. ACM, New York, NY, USA (2006)

4. The Impact of "Cosmetic" Changes on the Usability of Error Messages Tao Dong, Kandarp Khandwala May 2019 CHI EA '19: Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems

5. Designing a live development experience for web-components Jens Lincke, Patrick Rein, Stefan Ramson, Robert Hirschfeld, Marcel Taeumel, Tim Felgentreff October 2017 PX/17.2: Proceedings of the 3rd ACM SIGPLAN International Workshop on Programming Experience

6. Are Software Developers Just Users of Development Tools? Assessing Developer Experience of a Graphical User Interface Designer. Kati Kuusinen Conference paper First Online: 23 August 2016

Software Development

1. Capretz, L.F., Ahmed, F.: Making sense of software development and personality types. IT Prof. 12(1), 6–13 (2010)

2. An exploratory study on the influence of developers in technical debt Reem Alfayez, Pooyan Behnamghader, Kamonphop Srisopha, Barry Boehm May 2018 TechDebt '18: Proceedings of the 2018 International Conference on Technical Debt

Soft skills

1. empty

Deveoper mood

1. Graziotin, D., Wang, X., Abrahamsson, P.: Happy software developers solve problemsbetter: psychological measurements in empirical software engineering. PeerJ2(1), e289(2014)

2. Beecham, S., Baddoo, N., Hall, T., Robinson, H., Sharp, H.: Motivation in software engineering: A systematic literature review. IST 50, 860–878 (2008)

3. D. Graziotin, Consequences of unhappiness while developing software, in Proc. 2nd Int. Workshop Emotion Awareness Softw. Eng., May 2017, pp. 42–47.

4. On the Unhappiness of Software Developers Daniel Graziotin, Fabian Fagerholm, Xiaofeng Wang, Pekka Abrahamsson June 2017 EASE'17: Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering

Flow state and distractions. Sociability and Social Support

1. Ryan, R.M., Mims, V., Koestner, R.: Relation of reward contingency and interpersonal context to intrinsic motivation: a review and test using cognitive evaluation theory. J. Pers. Soc. Psychol. 45, 736–750 (1983)

2. 27.Oehlberg, L., Ducheneaut, N., Thorton, J.D., Moore, R.J.,Nickell, E. 2006. Social TV: Designing for D istributed, SocialTelevision Viewing. In Proc. Euro iTV'06. (2006), 251-259

3. Leont'ev, A. N. Activity, consciousness, and personality. Prentice-Hall Press, (1978)

4. Perlow, L.A., The time famine: Toward a sociology of work time. Admin. Science Quarterly, 44, (1999), 57-81

User Experience

1. Hassenzahl, M., Tractinsky, N.: User experience - a research agenda. Behav. Inf. Technol. 25(2), 91–97 (2006)

Other

1. Kansala, M. and Tuomivaara, S. (2013). Do Agile Principles and Practices Support the Well-being at Work of Agile Team Members? In Proceedings of the 8th International Conference on Software Engineering Advances (ICSEA 2013), pages 364–367.

2. Dingsøyr, T., Nerur, S., Balijepally, V., and Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. Journal of Systems and Software, 85(6):1213–1221.

3. Journal of Systems and Software Volume 140, June 2018, Pages 32-47 Journal of Systems and Software What happens when software developers are (un)happy Author links open overlay panel Daniel Graziotina Fabian Fagerholm bcXiaofengWangd PekkaAbrahamssone Show more https://doi.org/10.1016/j.jss.2018.02.041

4. Teamwork quality and project success in software development: A survey of agile development teams Author links open overlay panelYngveLindsjørnaDag I.K.Sjøbergab TorgeirDingsøyrbc Gunnar R.Bergersen Tore Dybåa

5. Gass, O., Meth, H., Maedche, A.: PaaS characteristics for productive software development: an evaluation framework. Internet Comput. IEEE 18(1), 56–64 (2014)

References

[1] L. F. Capretz. Personality types in software engineering. *International Journal of Human-Computer Studies*, 58(2):207 – 214, 2003.

[2] F. Fagerholm. *Software Developer Experience : Case Studies in Lean-Agile and Open Source Environments*. PhD thesis, University of Helsinki, 2015.

[3] F. Fagerholm and J. Münch. Developer experience: Concept and definition. *CoRR*, abs/1312.1452, 2013.

[4] D. Graziotin, F. Fagerholm, X. Wang, and P. Abrahamsson. Unhappy developers: Bad for themselves, bad for process, and bad for software product. *CoRR*, abs/1701.02952, 2017.

[5] K. Kuusinen. Software developers as users: Developer experience of a cross-platform integrated development environment. In P. Abrahamsson, L. Corral, M. Oivo, and B. Russo, editors, *Product-Focused Software Process Improvement*, pages 546–552, Cham, 2015. Springer International Publishing.

[6] K. Kuusinen, H. Petrie, F. Fagerholm, and T. Mikkonen. Flow, intrinsic motivation, and developer experience in software engineering. In H. Sharp and T. Hall, editors, *Agile Processes, in Software Engineering, and Extreme Programming*, volume 251. XP 2016, Springer, Cham, 2016.

[7] J. Morales, C. Rusu, F. Botella, and D. Quinones. Programmer experience: A systematic literature review. *IEEE Access*, PP:1–1, 05 2019.

[8] J. Palviainen, T. Kilamo, J. Koskinen, J. Lautamäki, T. Mikkonen, and A. Nieminen. Design framework enhancing developer experience in collaborative coding environment. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, SAC '15, pages 149–156, New York, NY, USA, 2015. ACM.