

From DevEx Disaster to Delight

How to Champion a DevEx Revolution in Your Organization

DevEx disasters...

Bad DevEx

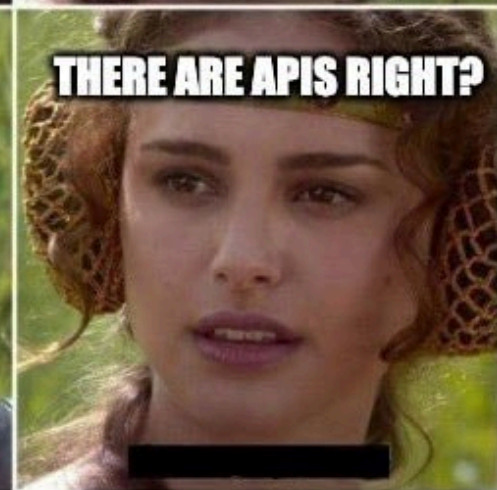
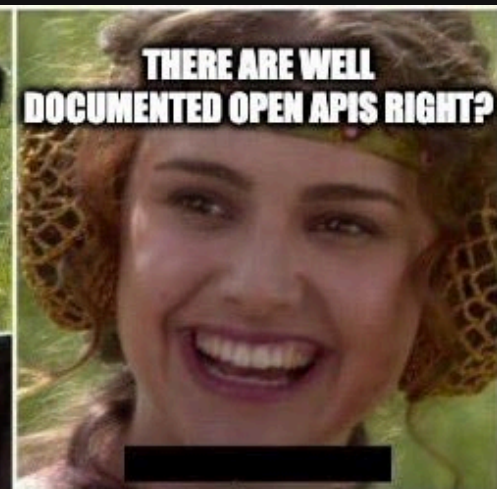
Common examples

- Poorly documented features (or bugs)

Bad DevEx

Common examples

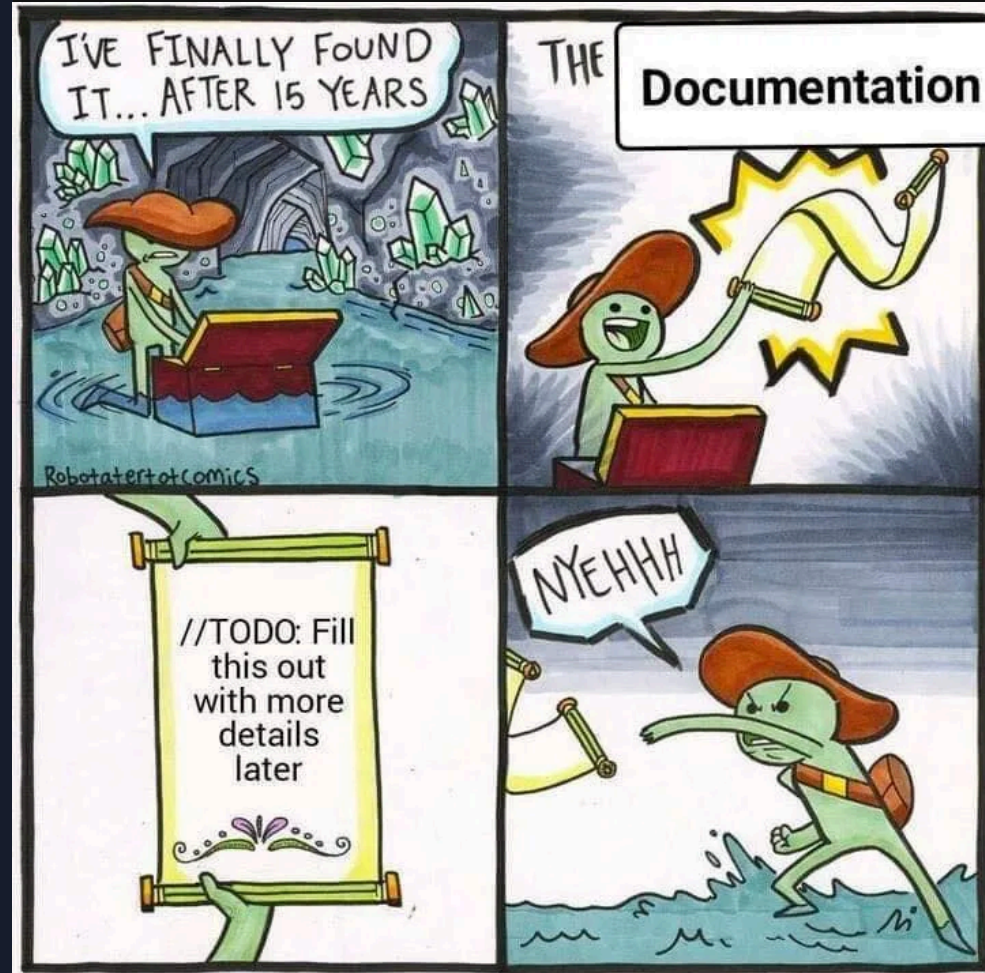
- Poorly documented features (or bugs)
- Missing OpenAPI spec (or even APIs)]



Bad DevEx

Common examples

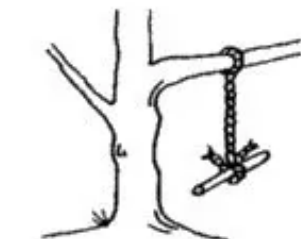
- Poorly documented features (or bugs)
- Missing OpenAPI spec (or even APIs)
- PDF documentation... or access-gated



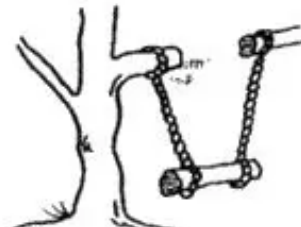
Bad DevEx

Common examples

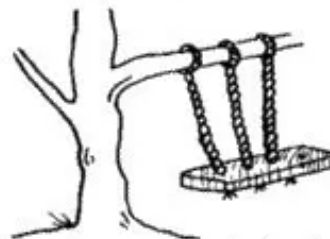
- Poorly documented features (or bugs)
- Missing OpenAPI spec (or even APIs)
- PDF documentation... or access-gated
- Missing examples... of *anything*



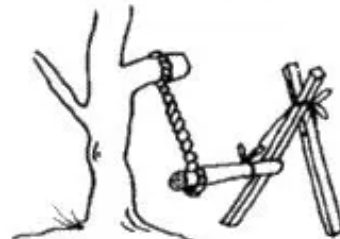
What the user asked for



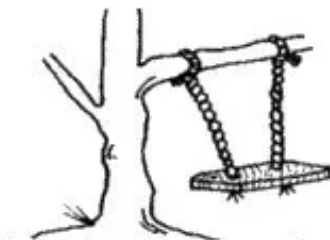
How the analyst saw it



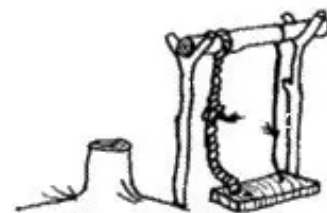
How the system was designed



As the programmer wrote it



What the user really wanted



How it actually works

Bad DevEx

Common examples

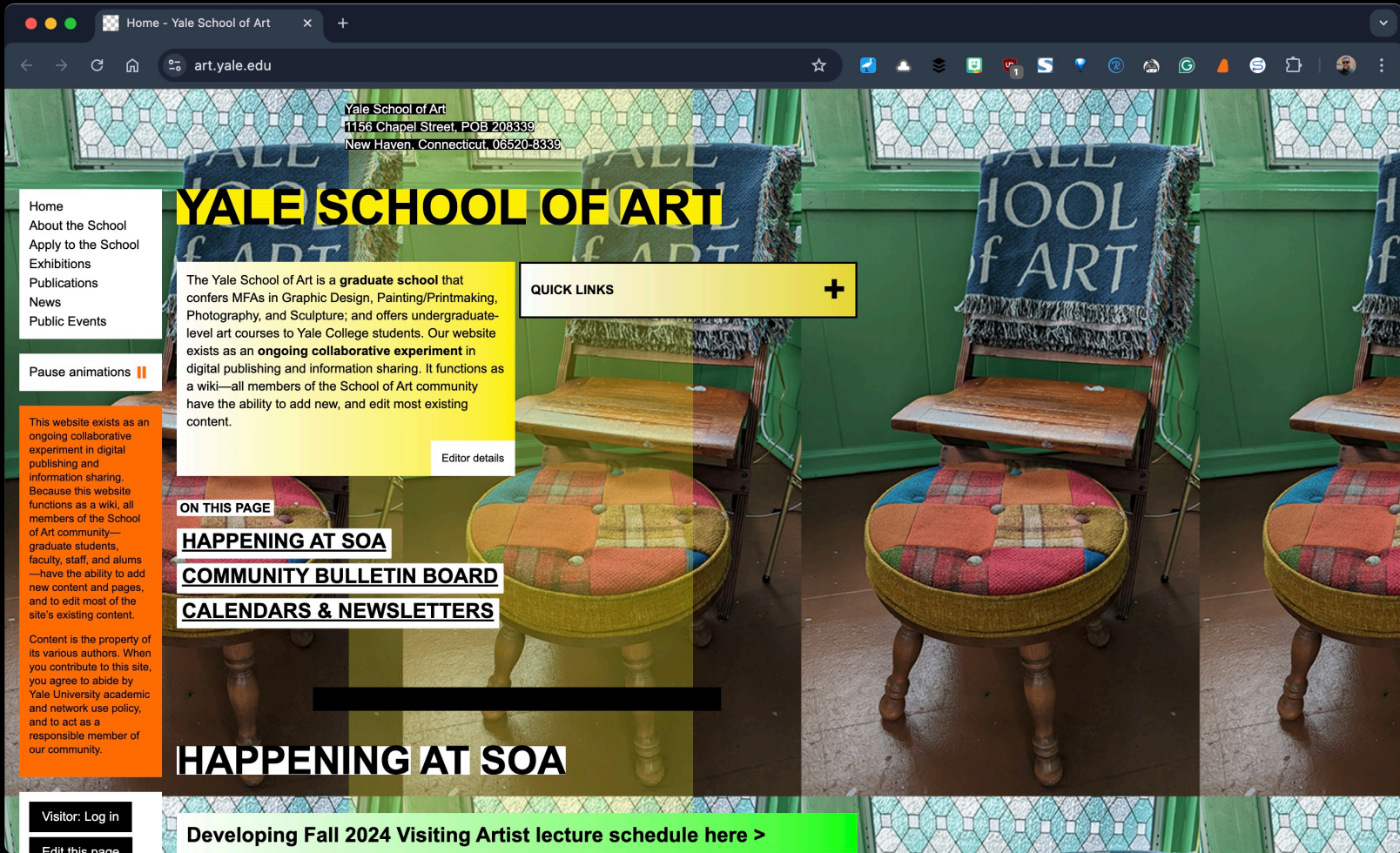
- Poorly documented features (or bugs)
- Missing OpenAPI spec (or even APIs)
- PDF documentation... or access-gated
- Missing examples... of *anything*
- “CI as Magic 8-Ball”



Ramiro Berrelleza 
@rberrelleza

Long time ago, in a galaxy far away, I worked at a team where our CI environment was so different from local or production, that the only realistic option way to validate a change was in prod. So we would commit the change, rerun CI jobs until they were green, deploy to prod, and then monitor the logs for about 1 hour. If no major errors were logged after that you were good to go 😊

12:39 AM · Aug 3, 2024



Yale School of Art
1156 Chapel Street, POB 208339
New Haven, Connecticut, 06520-8339

YALE SCHOOL OF ART

- Home
- About the School
- Apply to the School
- Exhibitions
- Publications
- News
- Public Events

Pause animations

This website exists as an ongoing collaborative experiment in digital publishing and information sharing. Because this website functions as a wiki, all members of the School of Art community—graduate students, faculty, staff, and alumni—have the ability to add new content and pages, and to edit most of the site's existing content.

Content is the property of its various authors. When you contribute to this site, you agree to abide by Yale University academic and network use policy, and to act as a responsible member of our community.

The Yale School of Art is a **graduate school** that confers MFAs in Graphic Design, Painting/Printmaking, Photography, and Sculpture; and offers undergraduate-level art courses to Yale College students. Our website exists as an **ongoing collaborative experiment** in digital publishing and information sharing. It functions as a wiki—all members of the School of Art community have the ability to add new, and edit most existing content.

[Editor details](#)

QUICK LINKS

ON THIS PAGE

[HAPPENING AT SOA](#)

[COMMUNITY BULLETIN BOARD](#)

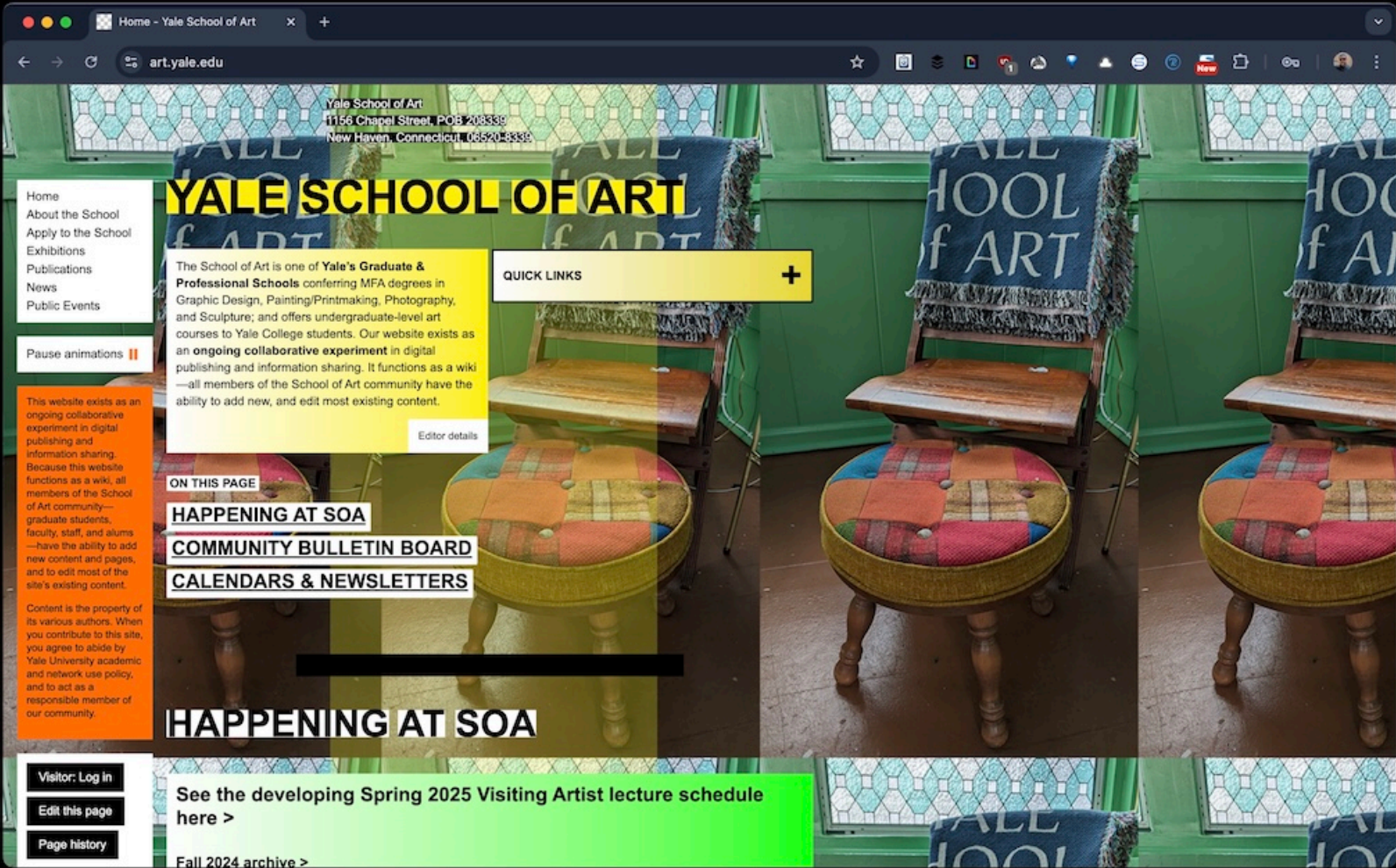
[CALENDARS & NEWSLETTERS](#)

HAPPENING AT SOA

[Developing Fall 2024 Visiting Artist lecture schedule here >](#)

Visitor: [Log in](#)

[Edit this page](#)



- Home
- About the School
- Apply to the School
- Exhibitions
- Publications
- News
- Public Events

Pause animations ||

This website exists as an ongoing collaborative experiment in digital publishing and information sharing. Because this website functions as a wiki, all members of the School of Art community—graduate students, faculty, staff, and alumni—have the ability to add new content and pages, and to edit most of the site's existing content.

Content is the property of its various authors. When you contribute to this site, you agree to abide by Yale University academic and network use policy, and to act as a responsible member of our community.

- Visitor: Log in
- Edit this page
- Page history

Yale School of Art
1156 Chapel Street, POB 208338
New Haven, Connecticut, 06520-8338

YALE SCHOOL OF ART

The School of Art is one of Yale's Graduate & Professional Schools conferring MFA degrees in Graphic Design, Painting/Printmaking, Photography, and Sculpture; and offers undergraduate-level art courses to Yale College students. Our website exists as an ongoing collaborative experiment in digital publishing and information sharing. It functions as a wiki—all members of the School of Art community have the ability to add new, and edit most existing content.

[Editor details](#)

QUICK LINKS



ON THIS PAGE


- [HAPPENING AT SOA](#)
- [COMMUNITY BULLETIN BOARD](#)
- [CALENDARS & NEWSLETTERS](#)

HAPPENING AT SOA

See the developing Spring 2025 Visiting Artist lecture schedule here >

[Fall 2024 archive >](#)

```
git push heroku main
```

 Deploy to Heroku



Developer Experience: Concept and Definition

Fabian Fagerholm, Jürgen Münch

New ways of working such as globally distributed development or the integration of self-motivated external developers into software ecosystems will require a better and more comprehensive understanding of developers' feelings, perceptions, motivations and identification with their tasks in their respective project environments. User experience is a concept that captures how persons feel about products, systems and services. It evolved from disciplines such as interaction design and usability to a much richer scope that includes feelings, motivations, and satisfaction. Similarly, developer experience could be defined as a means for capturing how developers think and feel about their activities within their working environments, with the assumption that an improvement of the developer experience has positive impacts on characteristics such as sustained team and project performance. This article motivates the importance of developer experience, sketches related approaches from other domains, proposes a definition of developer experience that is derived from similar concepts in other domains, describes an ongoing empirical study to better understand developer experience, and finally gives an outlook on planned future research activities.

Comments: 5 pages. The final publication is available at [this http URL](#)

Subjects: **Software Engineering (cs.SE)**

Cite as: [arXiv:1312.1452 \[cs.SE\]](#)
(or [arXiv:1312.1452v1 \[cs.SE\]](#) for this version)
<https://doi.org/10.48550/arXiv.1312.1452>

Journal reference: Proceedings of the International Conference on Software and System Process (ICSSP 2012), pages 73–77, Zurich, Switzerland, June 2–3 2012

DevEx isn't new

REF: F. Fagerholm and J. Münch, "Developer experience: Concept and definition," 2012 International Conference on Software and System Process (ICSSP), Zurich, Switzerland, 2012.



[Submitted on 5 Dec 2013]

Developer Experience: Concept and Definition

Fabian Fagerholm, Jürgen Münch

New ways of working such as globally distributed development or the integration of self-motivated external developers into software ecosystems will require a better and more comprehensive understanding of developers' feelings, perceptions, motivations and identification with their tasks in their respective project environments. User experience is a concept that captures how persons feel about products, systems and services. It evolved from disciplines such as interaction design and usability to a much richer scope that includes feelings, motivations, and satisfaction. Similarly, developer experience could be defined as a means for capturing how developers think and feel about their activities within their working environments, with the assumption that an improvement of the developer experience has positive impacts on characteristics such as sustained team and project performance. This article motivates the importance of developer experience, sketches related approaches from other domains, proposes a definition of developer experience that is derived from similar concepts in other domains, describes an ongoing empirical study to better understand developer experience, and finally gives an outlook on planned future research activities.

Comments: 5 pages. The final publication is available at [this http URL](#)Subjects: **Software Engineering (cs.SE)**

Cite as: [arXiv:1312.1452 \[cs.SE\]](#)
 (or [arXiv:1312.1452v1 \[cs.SE\]](#) for this version)
<https://doi.org/10.48550/arXiv.1312.1452>

Journal reference: Proceedings of the International Conference on Software and System Process (ICSSP 2012), pages 73–77, Zurich, Switzerland, June 2–3 2012

DevEx isn't new

"New ways of working such as globally distributed development or the integration of self-motivated external developers into software ecosystems will require a better and more comprehensive understanding of developers' feelings, perceptions, motivations and identification with their tasks in their respective project environments."

REF: F. Fagerholm and J. Münch, "Developer experience: Concept and definition. 2012."



[Submitted on 5 Dec 2013]

Developer Experience: Concept and Definition

Fabian Fagerholm, Jürgen Münch

New ways of working such as globally distributed development or the integration of self-motivated external developers into software ecosystems will require a better and more comprehensive understanding of developers' feelings, perceptions, motivations and identification with their tasks in their respective project environments. User experience is a concept that captures how persons feel about products, systems and services. It evolved from disciplines such as interaction design and usability to a much richer scope that includes feelings, motivations, and satisfaction. Similarly, developer experience could be defined as a means for capturing how developers think and feel about their activities within their working environments, with the assumption that an improvement of the developer experience has positive impacts on characteristics such as sustained team and project performance. This article motivates the importance of developer experience, sketches related approaches from other domains, proposes a definition of developer experience that is derived from similar concepts in other domains, describes an ongoing empirical study to better understand developer experience, and finally gives an outlook on planned future research activities.

Comments: 5 pages. The final publication is available at [this http URL](http://this)Subjects: **Software Engineering (cs.SE)**

Cite as: [arXiv:1312.1452](https://arxiv.org/abs/1312.1452) [cs.SE]
 (or [arXiv:1312.1452v1](https://arxiv.org/abs/1312.1452v1) [cs.SE] for this version)
<https://doi.org/10.48550/arXiv.1312.1452>

Journal reference: Proceedings of the International Conference on Software and System Process (ICSSP 2012), pages 73–77, Zurich, Switzerland, June 2–3 2012

DevEx isn't new

"...developer experience could be defined as a means for capturing how developers think and feel about their activities within their working environments, with the assumption that an improvement of the developer experience has positive impacts on characteristics such as sustained team and project performance."

REF: F. Fagerholm and J. Münch, "Developer experience: Concept and definition. 2012."

Jeremy Meiss

Director, DevEx & DevRel

OneStream Software

DevOpsDays Kansas City Organizer



Turning DevEx disasters into delights



- Overview of DevEx
- The Developer Experience journey
- Why DevEx matters
- DevEx fits all sizes
- Implementing DevEx in your org

A working definition of DevEx

— "...the **journey** of developers as they learn and deploy technology, which if successful, focuses on eliminating obstacles that hinder a developer or practitioner from achieving success in their endeavors."

-*Jessica West, Co-Founder, DevEx Institute*

Point of clarification

- "DevEx" by default focuses on "developer"
- View "DevEx" as a whole of the lifecycle



SOFTWARE DEVELOPMENT LIFECYCLE

SOFTWARE DEVELOPMENT LIFECYCLE

SOFTWARE DEVELOPMENT LIFE

PLANNING

USIG

DESIGN

DESIGN

DEVELOPMENT

DESIGN

DEVELOPMENT

MENANTAMEN

DEPLANNING

DEVELOPMENT

DESIGN

DEVELOPMENT

DEVELOPMENT

MAINTNANCE

MAINTANMENT



TECHNOGE

Cruggle

Crudging

Struggle

Cruggle

Craching

Errur

ERUR

DevEx: Making an impact

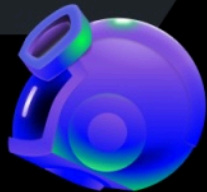
Source: [GitHub](#)

What do we get with better DevEx?

Developers: more productivity and creativity

Teams: better code quality and innovation

Organizations: more retention and profit



Support flow state: fewer interruptions, deeper work, and engaging tasks

50%↑

//significant time for deep work increases productivity

/Flow state

Increase cognitive ease: intuitive processes, understandable code, and easier deployment

50%↑

//intuitive processes increase innovation

/Cognitive ease

Optimize feedback loops: quick responses and code reviews

20%↑

//fast code review increases innovation

/Feedback loops

Home > ACM Journals > ACM Computing Surveys > Vol. 57, No. 1 > A Systematic Literature Review on the Influence of Enhanced Developer Experience on Developers' Productivity: Factors, Practices, and Recommendations

SURVEY | OPEN ACCESS | CC BY



A Systematic Literature Review on the Influence of Enhanced Developer Experience on Developers' Productivity: Factors, Practices, and Recommendations

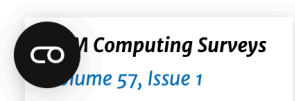
Authors: Abdul Razzaq, Jim Buckley, Qin Lai, Tingting Yu, Goetz Botterweck | [Authors Info & Claims](#)

ACM Computing Surveys, Volume 57, Issue 1 • Article No.: 13, Pages 1 - 46 • <https://doi.org/10.1145/3687299>

Published: 07 October 2024 [Publication History](#)



0 1,948



Abstract

``Developer Experience` != `Developer Productivity``

DevEx factors that impact Dev Productivity

Positive influencers

- Availability of **resources and tools**
- **Relevant expertise** for assigned tasks
- **Minimized interruptions** to maintain developer flow

Negative influencers

- **Code complexity and technical debt**
- **Diverse contexts of tasks**, causing cognitive overload.
- **Lack of standardization**, leading to inefficiencies.

Key Themes of Developer Experience

1. *Developer proficiency & growth*

- Align tasks with expertise and skill level
- Focus on skill development, mentorship, structured work
- Provide challenging but meaningful tasks

Key Themes of Developer Experience

1. Developer proficiency & growth
2. *Work environment & productivity flow*
 - Minimize interruptions, unnecessary context-switching, distractions
 - A healthy physical and virtual work environment
 - Give autonomy over work, tools, decision-making

Key Themes of Developer Experience

1. Developer proficiency & growth
2. Work environment & productivity flow
3. ***Collaboration & communication***
 - Effective team collaboration
 - Provide clear, accessible, relevant information
 - Encourage psychological safety and supportive team culture

Key Themes of Developer Experience

1. Developer proficiency & growth
2. Work environment & productivity flow
3. Collaboration & communication
4. *Code & tooling quality*
 - High-quality, maintainable, well-documented codebases
 - Intuitive, reliable, well-integrated tools and APIs
 - Automation and developer-friendly tooling investment

Key Themes of Developer Experience

1. Developer proficiency & growth
2. Work environment & productivity flow
3. Collaboration & communication
4. Code & tooling quality
5. ***Process & standardization***
 - Balance structured process and developer flexibility
 - Standardization that supports, not hinders, productivity
 - Steadily evolving technical ecosystem with right resources

Making the Case for Developer Experience

Making the case...

Responding to the challenges

1. Acknowledge the concern
2. Focus on the positive
3. Use data and examples
4. Offer a path forward
5. Emphasize collaboration

Making the Case... Challenges you'll likely face

Budget constraints

Challenge:

"We'd love to improve developer experience, but we just don't have the budget for it right now. These kinds of initiatives are expensive."

Response with benefit:

"I understand the concern about budget. However, investing in DevEx isn't just an expense; it's an investment in our team's efficiency and ultimately, our bottom line. For example, by streamlining our onboarding process, we could save ``X`` hours per new developer, which translates to ``Y`` dollars. Let's explore some low-cost, high-impact options we could pilot."

Making the Case... Challenges you'll likely face

Lack of Awareness

Challenge:

"Developer experience? Isn't that just about giving developers nicer keyboards, beanbag chairs, foosball, and artisan coffee? We have more pressing issues to deal with."

Response with benefit:

"That's a common misconception - it is much broader than perks. It's about creating an environment where developers can do their best work. Poor DevEx can lead to frustration, slow development cycles, and even developers leaving the company. Let me share some data that shows the link between DevEx and key metrics like productivity and retention."

Making the Case... Challenges you'll likely face

Resistance to change

Challenge:

"We've always done things this way. Why change now? New tools and processes just add complexity and slow us down."

Response with benefit:

"I understand the hesitation - change can be uncomfortable. But the goal here isn't to add complexity; it's to remove it. These improvements are designed to make our work easier and more efficient in the long run. Let's try a small pilot project with a few volunteers and see how it goes. We can gather feedback and adjust as needed."



DevEx for All

Small Teams - The Agile Advantage

Focus on lightweight tools, processes to promote collaboration and knowledge sharing





DevEx for All

Scaling to Enterprise

Create a specific DevEx team to drive initiatives, or prioritize internal DevEx community for sharing best practices and collaboration.

KEY: Internal developer portal for centralized resources and tools.

DevEx for All

Common Ground

1. Developer feedback
2. Continuous improvement
3. Automation



Practical Implementation

Turning DevEx into Reality

Turning DevEx into Reality

1. *Start Small, Win Big*

- Identify a pain point
- Define success

Example: Streamline onboarding



Turning DevEx into Reality

1. Start Small, Win Big
2. *Focus on feedback*
 - Multiple channels
 - Act on the feedback



Turning DevEx into Reality

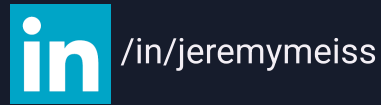
1. Start Small, Win Big
2. Focus on feedback
3. ***Metrics that matter***
 - Measure the impact
 - Communicate results



DevEx is...

"ruthlessly eliminating barriers (and blockers) that keep your practitioners from being successful"

Thank you!



END