

Demystifying "Developer Experience" ...

and why it matters

"Bad developer experience?"

"You know it when you see it..."

Bad DevEx

Common examples

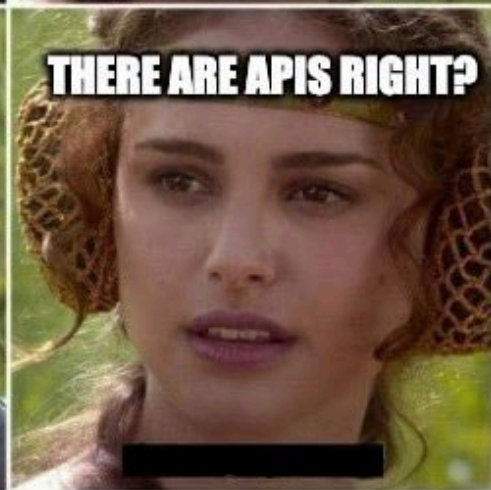
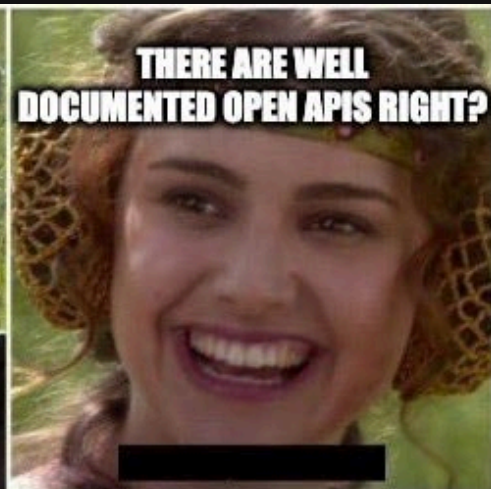
- Poorly documented features (or bugs)



Bad DevEx

Common examples

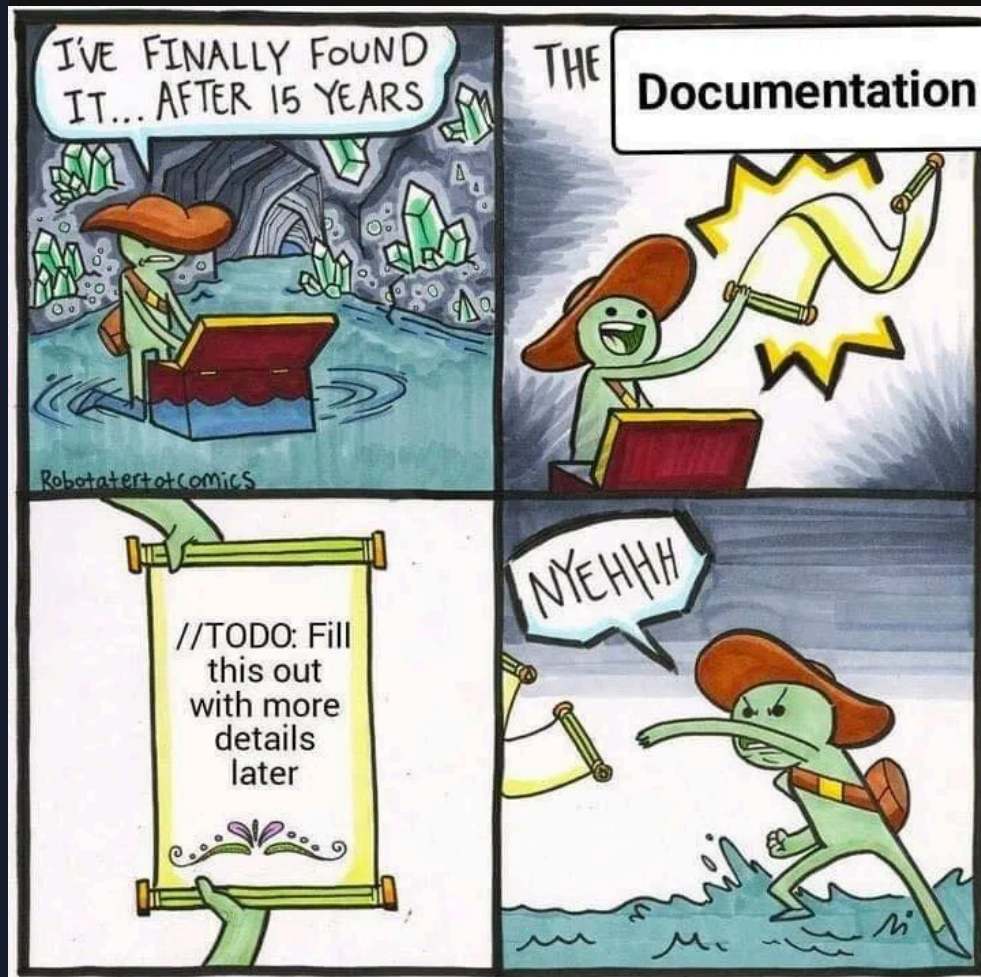
- Poorly documented features (or bugs)
- Missing OpenAPI spec (or even APIs)



Bad DevEx

Common examples

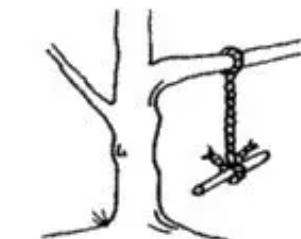
- Poorly documented features (or bugs)
- Missing OpenAPI spec (or event APIs)
- Downloading documentation... as a PDF, or access-gated



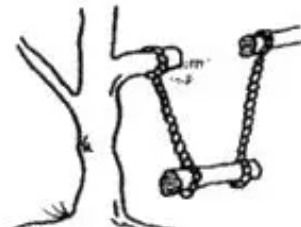
Bad DevEx

Common examples

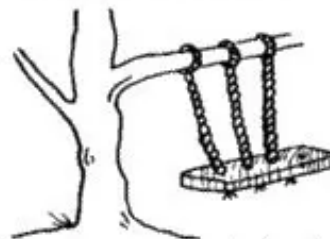
- Poorly documented features (or bugs)
- Missing OpenAPI spec (or event APIs)
- Downloading documentation... as a PDF, or access-gated
- Missing examples... of *anything*



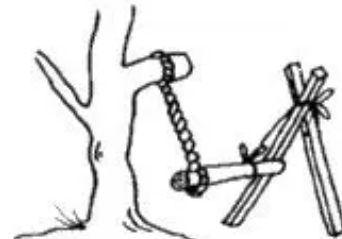
What the user asked for



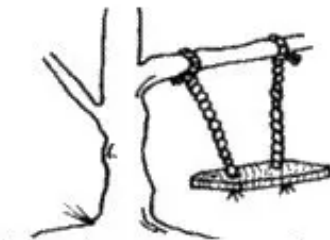
How the analyst saw it



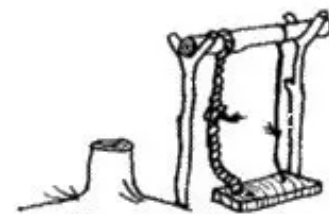
How the system was designed



As the programmer wrote it



What the user really wanted



How it actually works

Bad DevEx

Common examples

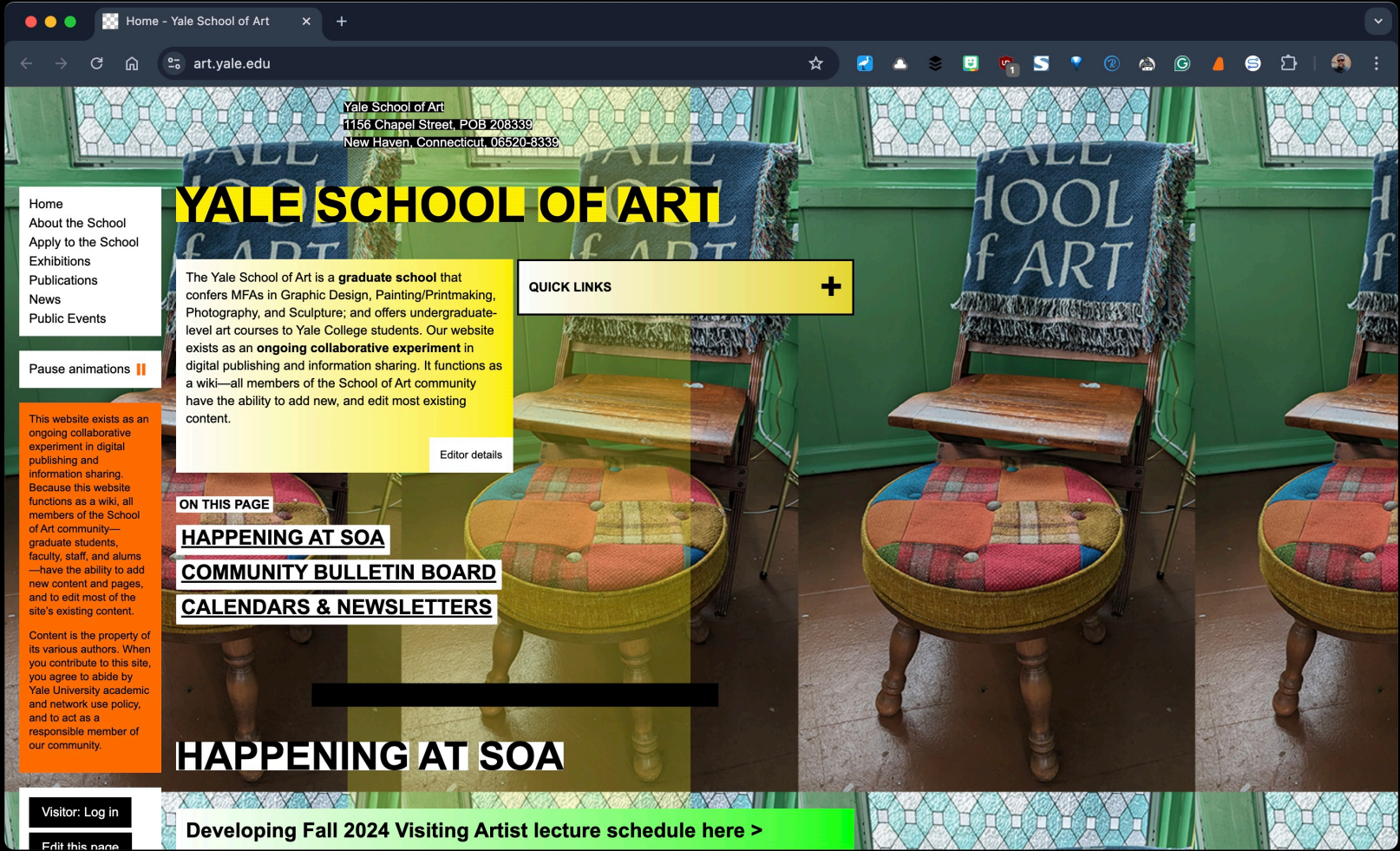
- Poorly documented features (or bugs)
- Missing OpenAPI spec (or event APIs)
- Downloading documentation... as a PDF, or access-gated
- Missing examples... of *anything*
- “CI as Magic 8-Ball”



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

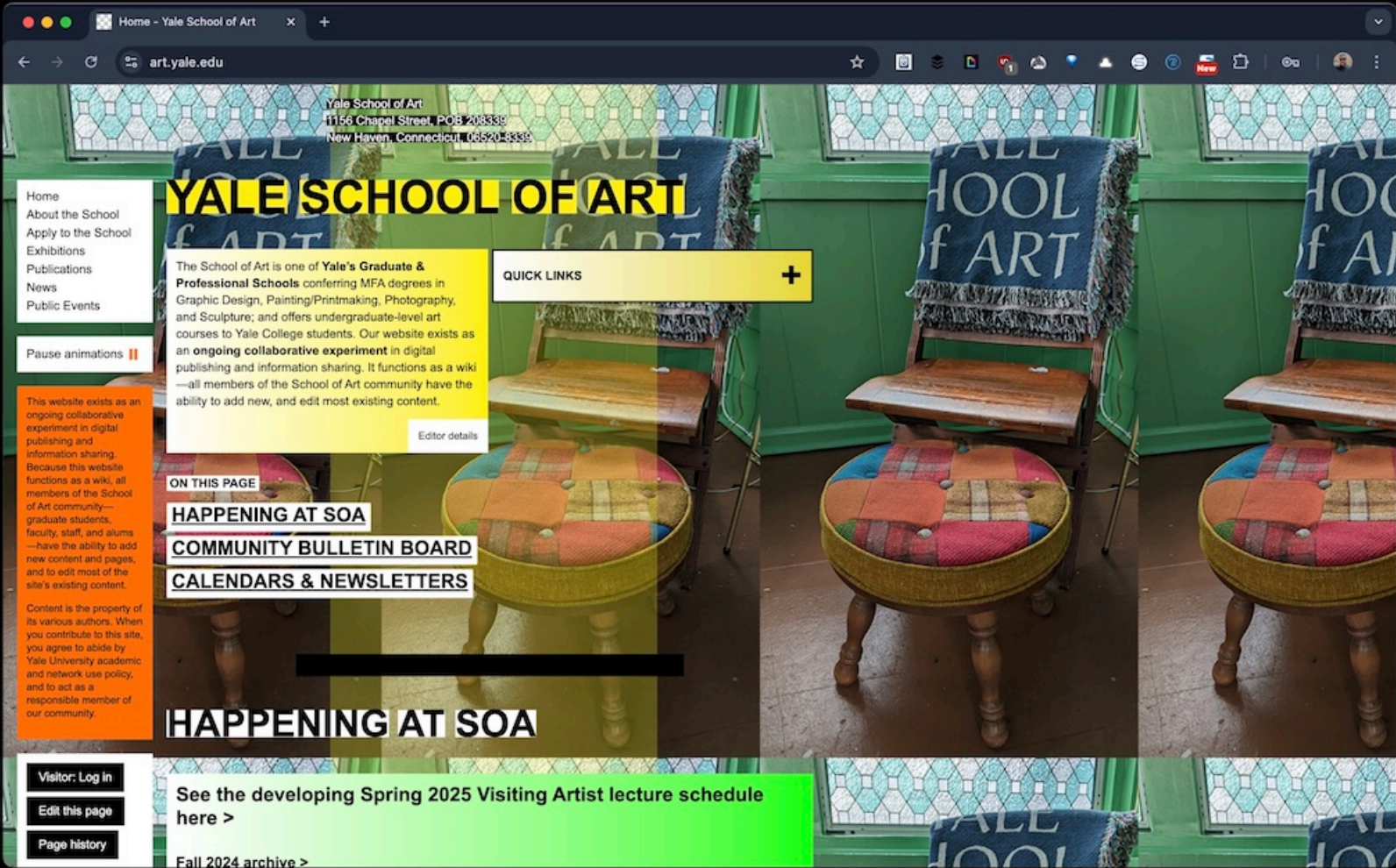
COMMUNITY BULLETIN BOARD

CALENDARS & NEWSLETTERS

HAPPENING AT SOA

Visitor: Log in
[Edit this page](#)

Developing Fall 2024 Visiting Artist lecture schedule here >



- 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 >](#)

What about a good Developer Experience?

```
git push heroku main
```

 Deploy to Heroku

Jeremy Meiss

Director, DevEx & DevRel

OneStream Software

DevOpsDays Kansas City Organizer



A working definition of DevEx

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

-Jessica West, Co-Founder, DevEx Institute

DevEx

A Good Developer Experience



Developer Productivity != Developer Experience

Developer Productivity != Developer Experience

Developer Productivity

- **Focus:** *measured in terms of output, with an emphasis on efficiency and performance.*
- **Context:** *metrics like "time to release", "number of pull requests", or "deployment frequency".*
- **Differs from DevEx:** *doesn't capture the full experience of developers.*

Developer Productivity != Developer Experience

Developer Productivity

- **Focus:** *measured in terms of output, with an emphasis on efficiency and performance.*
- **Context:** *metrics like "time to release", "number of pull requests", or "deployment frequency".*
- **Differs from DevEx:** *doesn't capture the full experience of developers.*

Developer Experience (DevEx)

- **Focus:** *holistic view encompassing all aspects of the developer journey (usability, efficiency, satisfaction, etc.)*
- **Unique:** *integrates elements of UX and productivity, but with a broader scope of psychological safety, community, and feedback loops.*



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](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

"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](#)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

"...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."

BIT EJTERS
BAINCE.



FROM TEXT EDITORS TO CLOUD, IDE: THE EVOLUTION OF DEVELOPMENT



FOLLOD ES
PECOLIP



Evolution of the IDE

Early text editors

USER FRIENDLY by Illiad



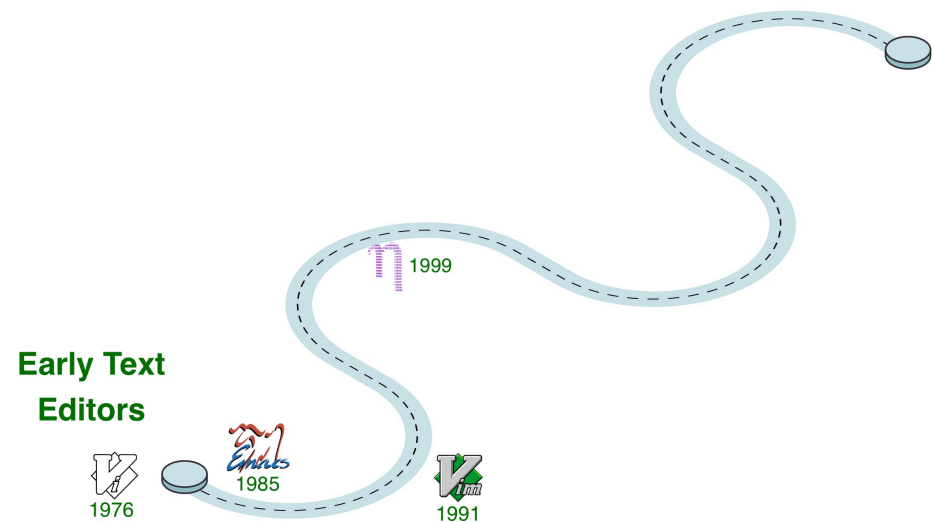
REF: O'Reilly "Learning the vi and Vim Editors"

Evolution of the IDE

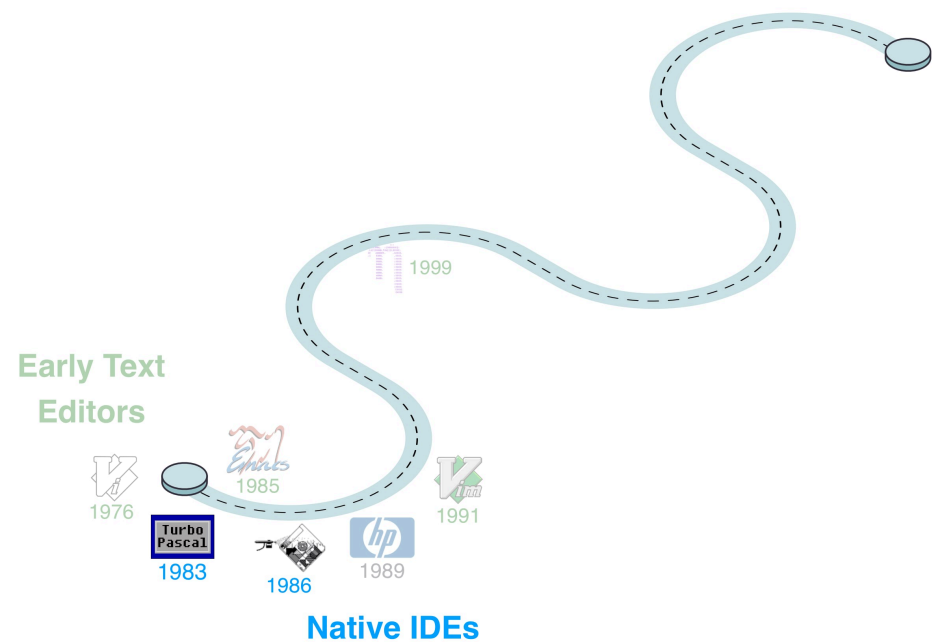
Early text editors

- 1976: Vi
- 1985: Emacs
- 1991: Vim
- 1999: nano

EVOLUTION OF THE IDE



EVOLUTION OF THE IDE

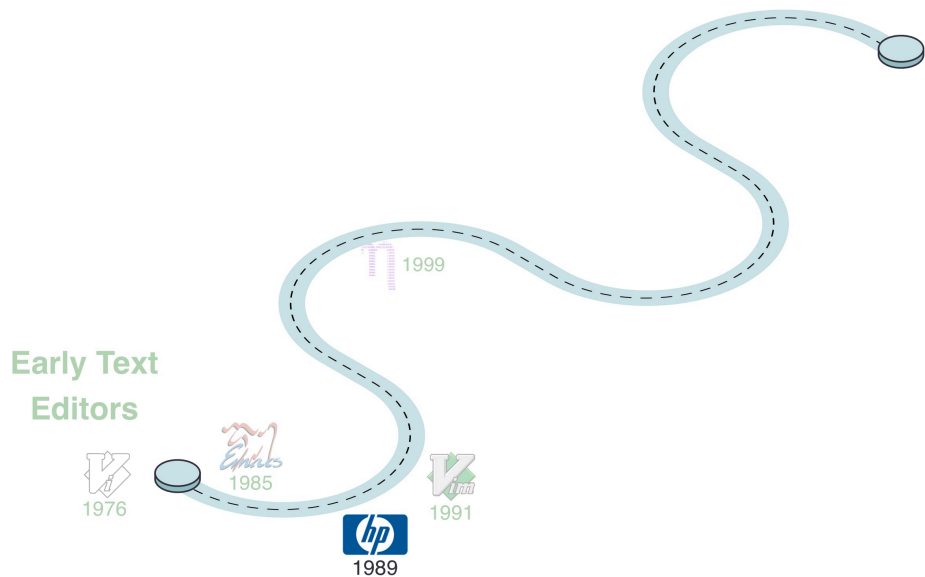


Evolution of the IDE

Native IDEs in the 1980s

- 1983: Turbo Pascal
- 1986: Apple's Macintosh Programmer's Workshop

EVOLUTION OF THE IDE



Evolution of the IDE

First plug-in IDE

HP Softbench

Evolution of the IDE

First plug-in IDE

HP Softbench



Evolution of the IDE

First plug-in IDE

HP Softbench

REF: HP Journal, June 1990 edition

The HP SoftBench Environment: An Architecture for a New Generation of Software Tools

The HP SoftBench product improves programmer productivity by integrating software development tools into a single unified environment, allowing the program developer to concentrate on tasks rather than tools.

by Martin R. Cagan

THE HP SOFTBENCH PRODUCT is an integrated software development environment designed to facilitate rapid, interactive program construction, test, and maintenance in a distributed computing environment.

The HP SoftBench environment provides an architecture for integrating various CASE (computer-aided software engineering) tools. Many of the tools most often needed—program editor, static analyzer, program debugger, program builder, and mail—are included in the HP SoftBench product. Another HP SoftBench component, the HP Encapsulator, makes it possible to integrate other existing tools into the HP SoftBench environment and to tailor the environment to a specific software development process. Fig.

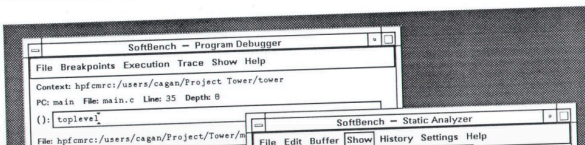
1 illustrates the HP SoftBench user interface.

This article describes the HP SoftBench tool integration architecture. The HP SoftBench program editor, static analyzer, program debugger, program builder, and mail are described in the article on page 48. The HP Encapsulator is described in the article on page 59.

Design Objectives

The overall goal of the HP SoftBench product is to improve the productivity of programmers doing software development, testing, and maintenance. To achieve this goal, the following objectives were defined for the tool integration architecture:

(continued on page 38)





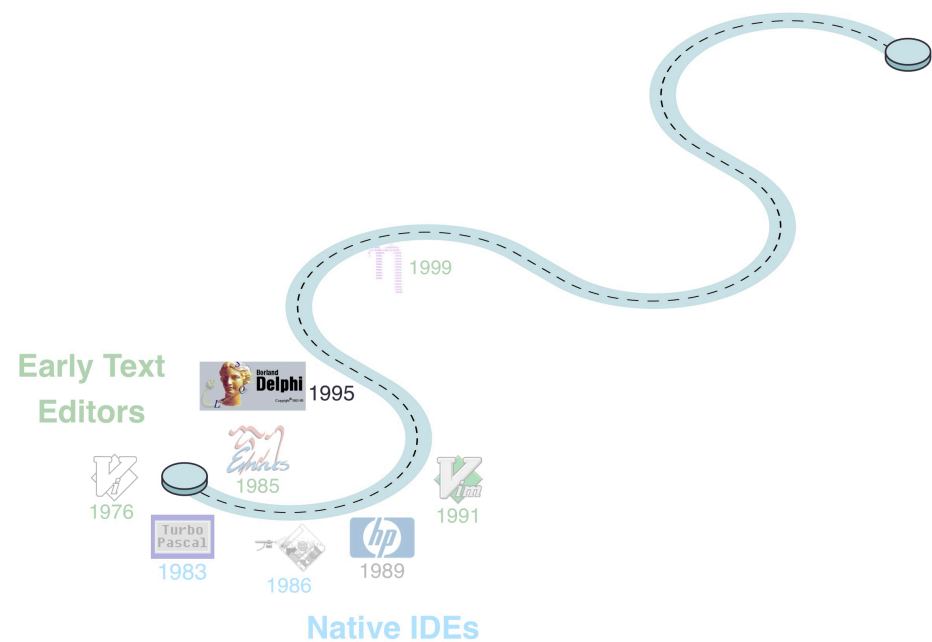
Evolution of the IDE

Early Reviews

"...the use of an IDE was not well received by developers since it would fence in their creativity."

REF: *Computerwoche* ("Computer Week", German counterpart of American magazine *Computer World*), 1995.

EVOLUTION OF THE IDE



Evolution of the IDE

Cross-platform in the 1990s

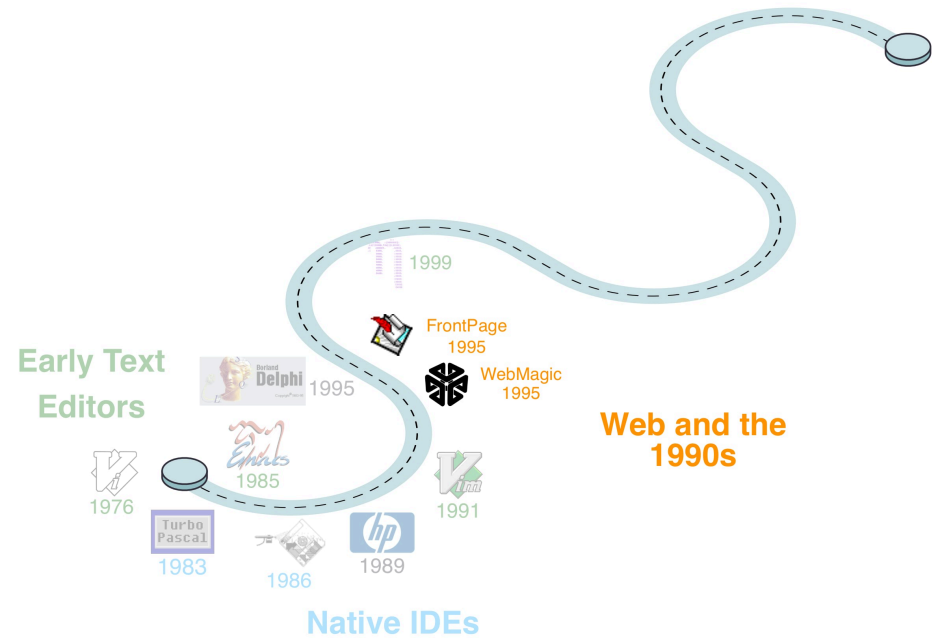
1995: Borland Delphi

Evolution of the IDE

The Web and the 1990s

- 1995: SGI WebMagic
- 1995: Microsoft FrontPage

EVOLUTION OF THE IDE



Evolution of the IDE

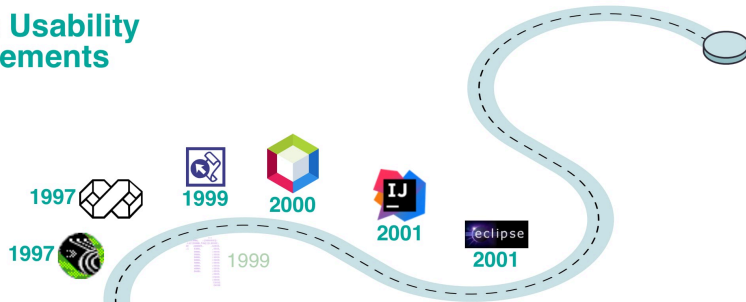
Features & Usability

Late 1990s to 2000s

- 1997: Macromedia Dreamweaver
- 1997: Microsoft Visual Studio
- 1999: Microsoft FrontPage 2000
- 2000: NetBeans
- 2001: IntelliJ IDEA
- 2001: Eclipse IDE
- 2002: Microsoft Visual Studio .NET

EVOLUTION OF THE IDE

Feature & Usability Enhancements



Web and the 1990s

Native IDEs

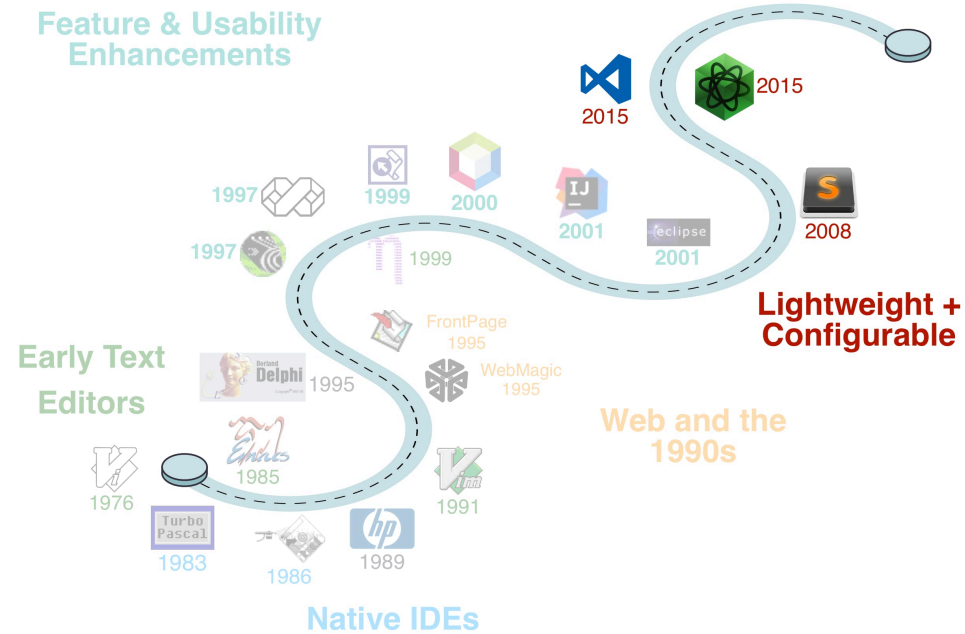
Evolution of the IDE

Lightweight & Configurable

2010s to Now

- 2008: Sublime Text
- 2015: Atom
- 2015: Visual Studio Code

EVOLUTION OF THE IDE



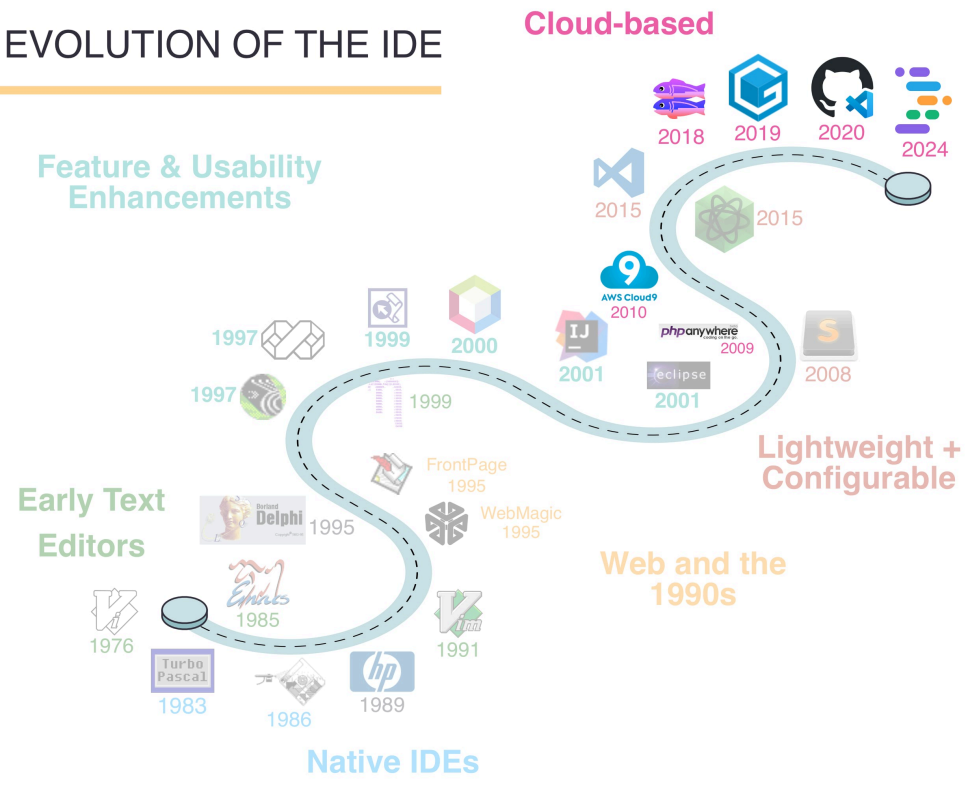
Evolution of the IDE

Cloud-based Options

Now

- 2009: PHPanywhere (eventually becoming CodeAnywhere)
- 2010: Cloud9 (AWS bought it in 2016)
- 2018: Glitch
- 2019: GitPod
- 2020: GitHub Codespaces
- 2024: Google Project IDX

EVOLUTION OF THE IDE



Evolution of the IDE

A result of DevEx

From this:

"...the use of an IDE was not well received by developers since it would fence in their creativity."

Evolution of the IDE

A result of DevEx

Things we never knew we needed...

To this:

- Code completion
- Code refactoring
- Syntax highlighting
- Debugging
- VCS integration (no more FTPing files around)
- Multi-language support
- Framework integration
- Pair programming



1930



Developer Experience

DEVOPS



2079

Key Themes of Developer Experience

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

How does DevEx apply to frontend developers?

DevEx theme ==> Frontend devs

Developer proficiency & growth

Why it matters

- Rapidly changing frameworks
- Critical mastery of JavaScript, CSS, accessibility, performance optimization
- Developers need time and resources to stay up-to-date and improve

Strategies for enhancement

- Continuous learning & experimentation
- Mentorship & pair programming
- Clear career growth paths







DevEx theme ==> Frontend devs

Work environment & productivity flow

Why it matters

- Frequent iterations the norm
- Focus on creative problem solving
- Bottlenecks, distractions, and interruptions hinder productivity

Strategies for enhancement

- Fast, configurable dev environments   
- Component libraries, design systems  
- Browser DevTools, performance monitoring 









DevEx theme ==> Frontend devs

Collaboration & communication

Why it matters

- Interact heavily with designers, backend engineers, product teams
- Lack of clear communication slows down development
- Responsive behavior, accessibility, animations need structured workflows

Strategies for enhancement









- Strong design-dev collaboration  
- Clear API specs & mocking strategies   
- Shared docs & communication   

DevEx theme ==> Frontend devs Code & tooling quality

Why it matters

- Poor tooling slows down development, increases bugs in production
- Inconsistent code bases lead to hard-to-maintain projects, tech debt
- Focus on accessibility, security, performance ensures high-quality standards

Strategies for enhancement

- Standardized tooling & linting   
- Automated testing & CI/CD    
- Performance & accessibility audits 

DevEx theme ==> Frontend devs

Process & standardization

Why it matters

- Inconsistent processes slow down teams, create friction
- Standardized workflows reduce decision fatigue, cognitive load
- Defined processes for code reviews, versioning, onboarding == efficient, predictable dev cycles

Strategies for enhancement

- Define best practices for code reviews
- Implement versioning & release management
- Project onboarding & documentation

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!



@jerdog.dev



/in/jeremymeiss



@jerdog



@jerdog@hachyderm.io



jmeiss.me



@IAmJerdog

END