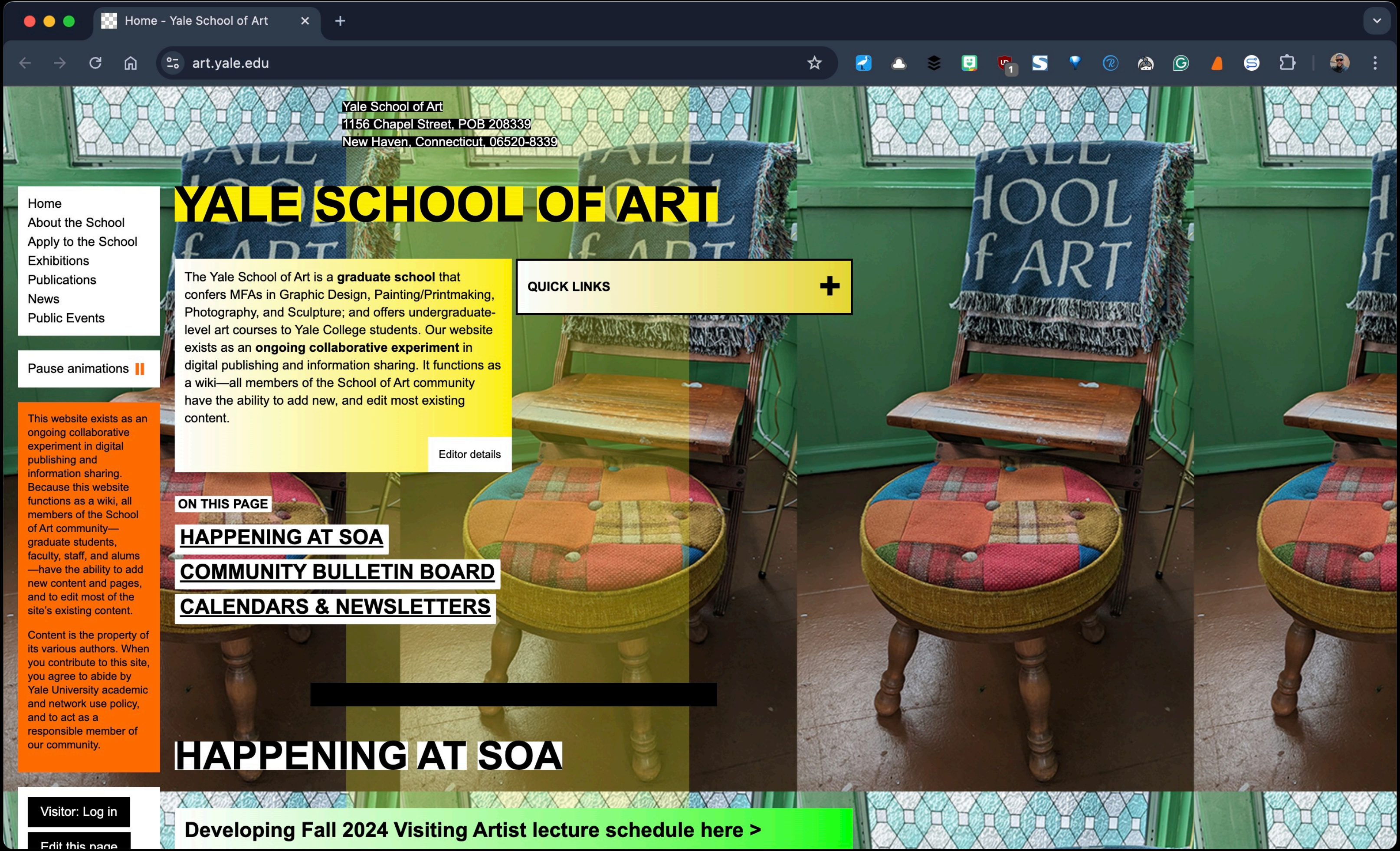
Developer Experience...

Central to DevOps Success

What is "Developer Experience" (DevEx)?



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

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 alums—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

YALE SCHOOL OF ART

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

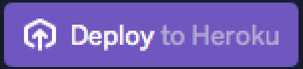
COMMUNITY BULLETIN BOARD

CALENDARS & NEWSLETTERS

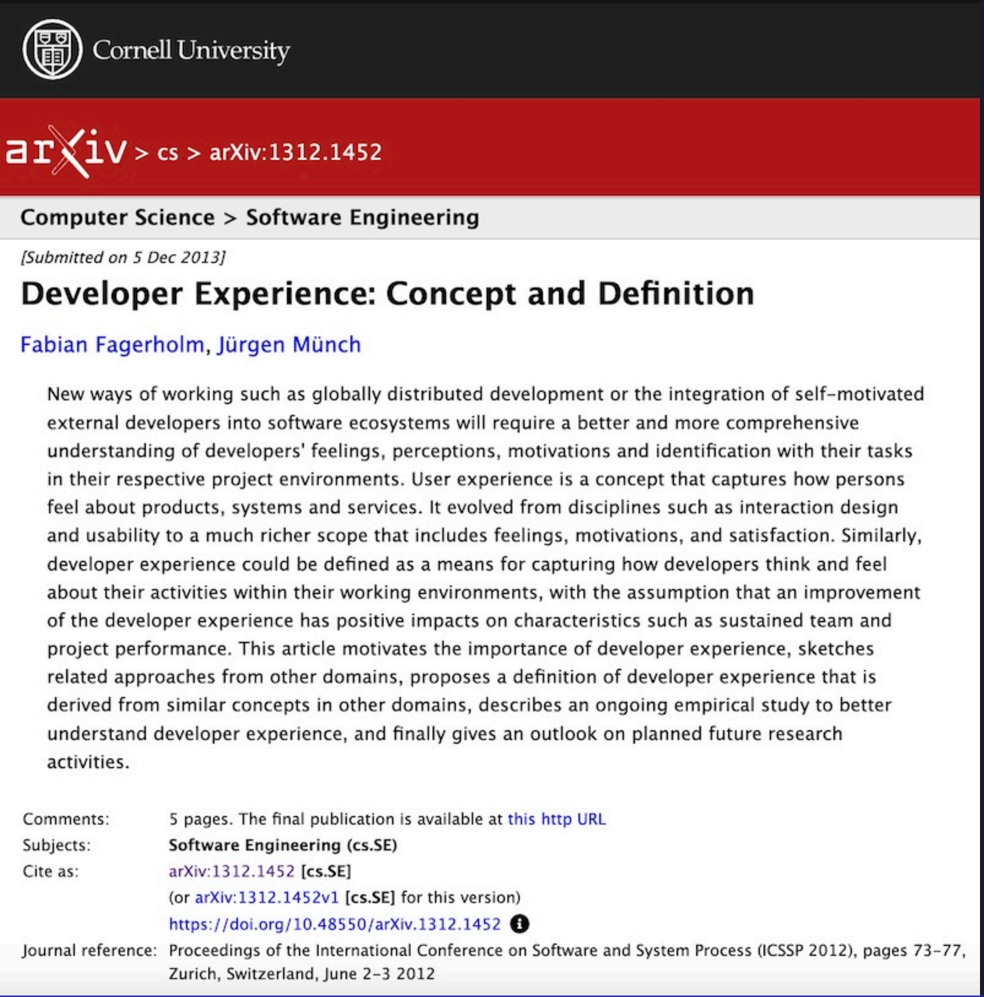
HAPPENING AT SOA

Developing Fall 2024 Visiting Artist lecture schedule here >

```
git push heroku main
```



DevEx isn't new


Cornell University

arXiv > cs > arXiv:1312.1452

Computer Science > Software Engineering

[Submitted on 5 Dec 2013]

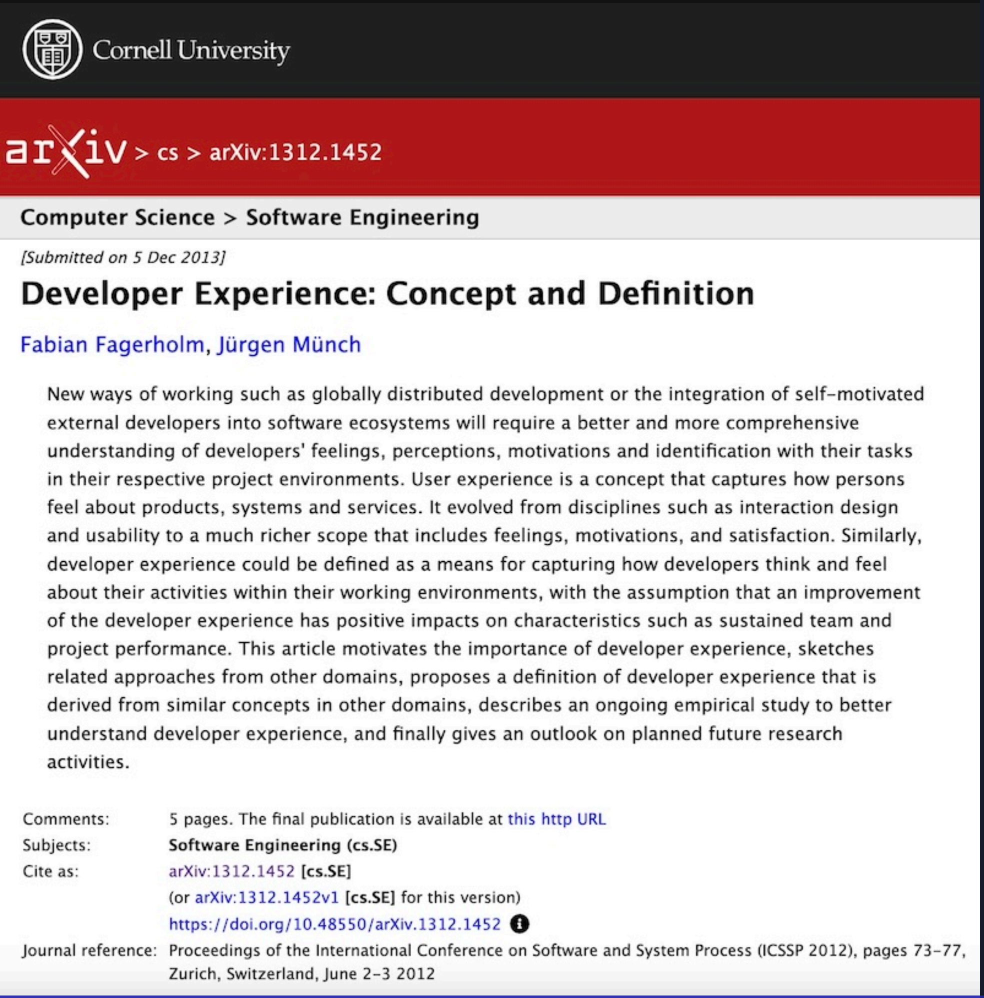
Developer Experience: Concept and Definition

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

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



Cornell University

arXiv > cs > arXiv:1312.1452

Computer Science > Software Engineering

[Submitted on 5 Dec 2013]

Developer Experience: Concept and Definition

Fabian Fagerholm, Jürgen Münch

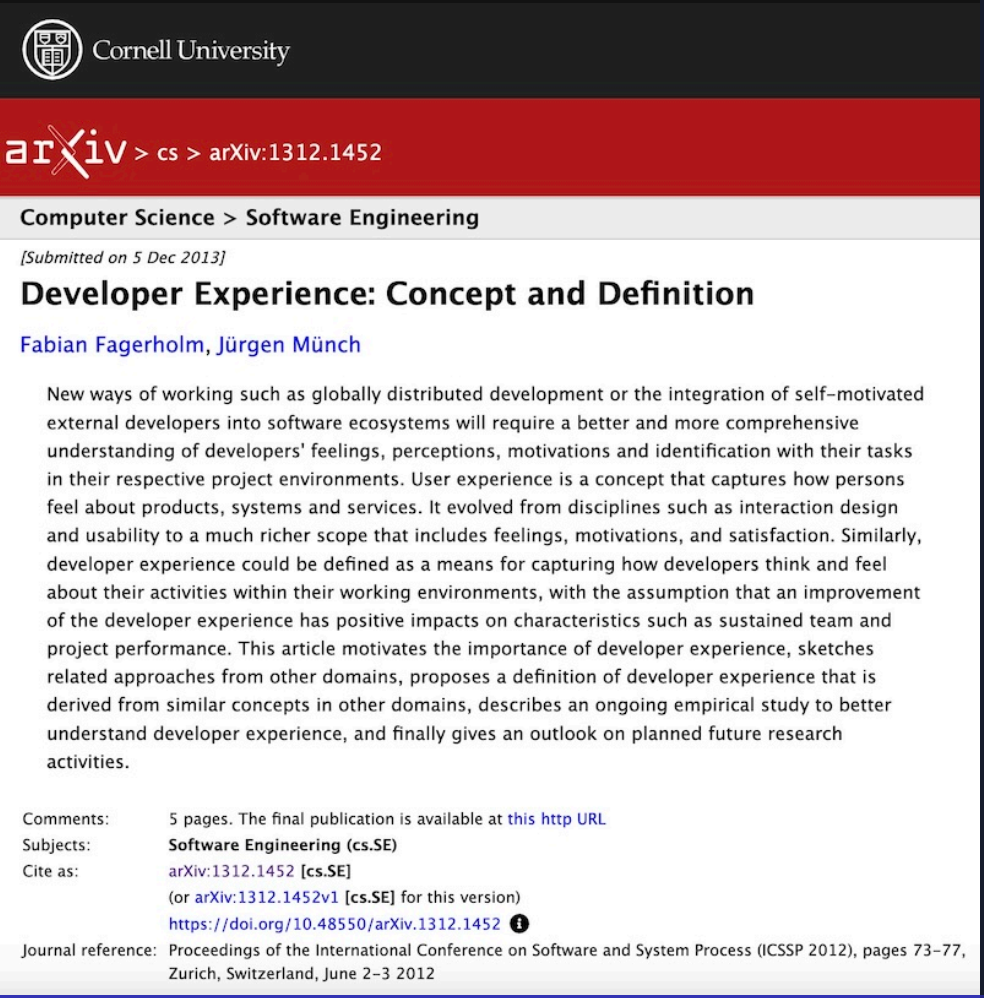
New ways of working such as globally distributed development or the integration of self-motivated external developers into software ecosystems will require a better and more comprehensive understanding of developers' feelings, perceptions, motivations and identification with their tasks in their respective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



Cornell University

arXiv > cs > arXiv:1312.1452

Computer Science > Software Engineering

[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."

Jeremy Meiss

Co-Founder, DevEx Consultant

DevEx Institute

DevOpsDays Kansas City Organizer

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."

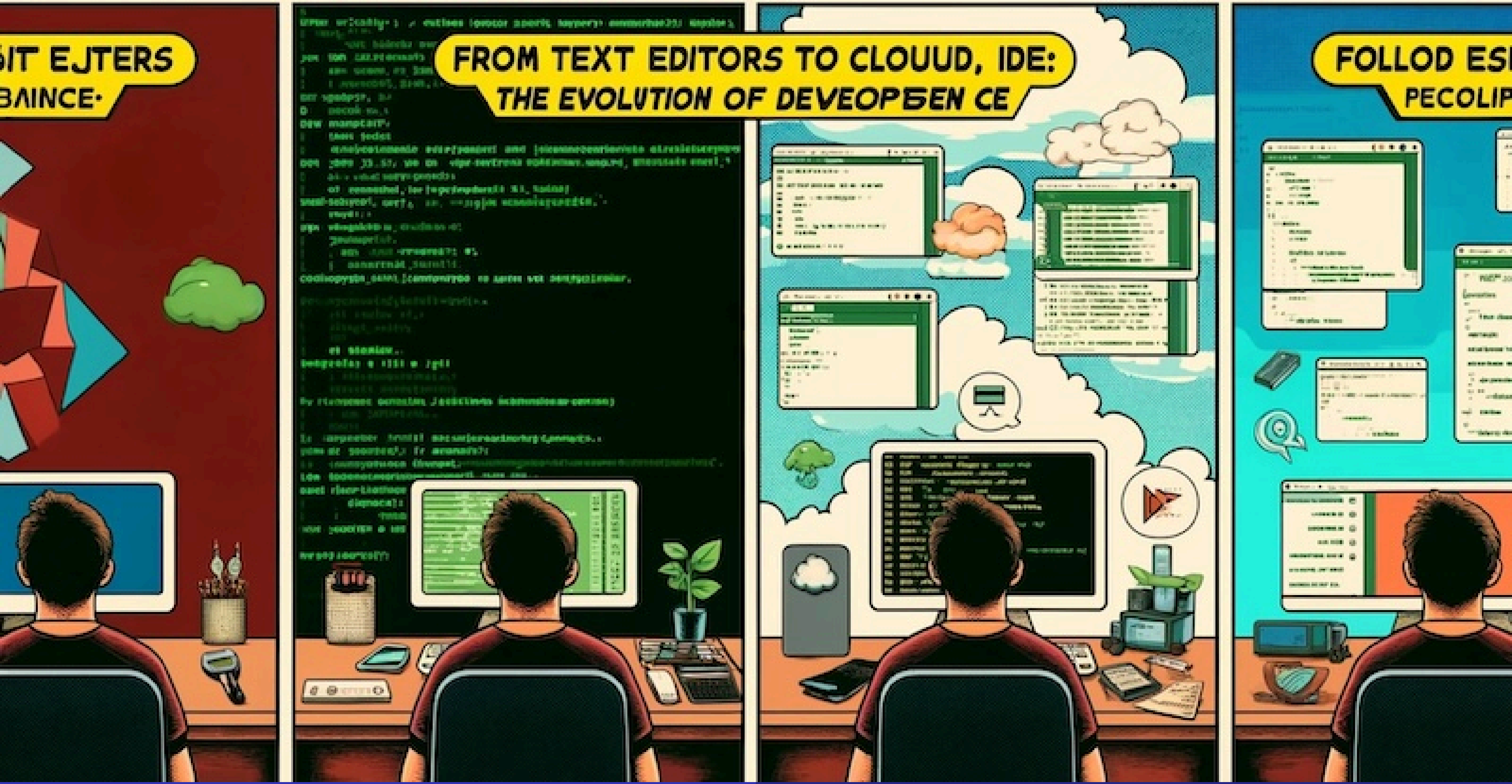
-**Jessica West**, *Co-Founder, DevEx Institute*

Point of clarification

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

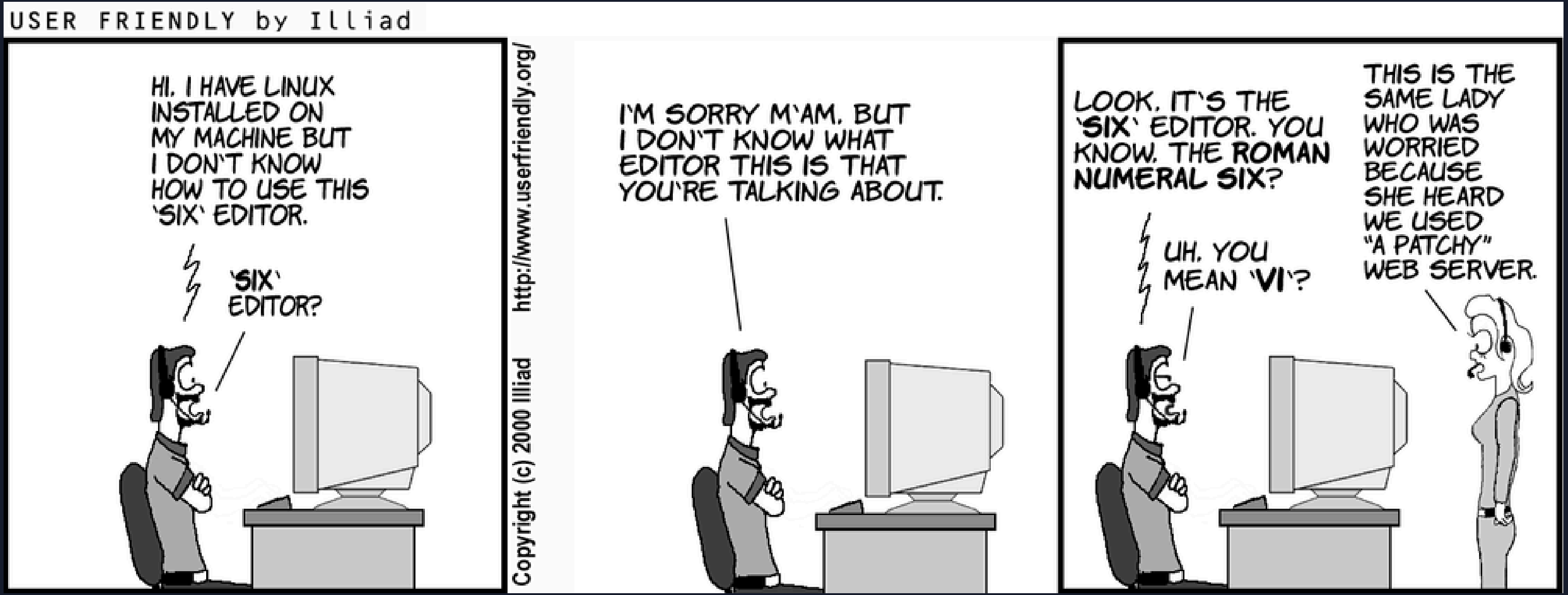
DevEx
A Good Developer Experience
Overal
Satisiency
OVERL SAFISATON
EFFECIENCY
EFFECIENCY
Ocroed
Developner
& ipchnolongy
DEVELOPIER
EFFECIENCY
OOAD
TOOL
EFFEC

DEVELOPER EXPERIENCE
DEVEX
PLANNING
Pe
CODING
DEVEX
CODING
CODING
CODING
PARNING
CODING
TESTING
TESTING
Software Development Lifecycle
Software Deploseycle
Deployition
Seyttware


SIT EJTERS
BAINCE·
FROM TEXT EDITORS TO CLOUUD, IDE:
THE EVOLUTION OF DEVEOPБEN CE
FOLLOD ES
PECOLIP

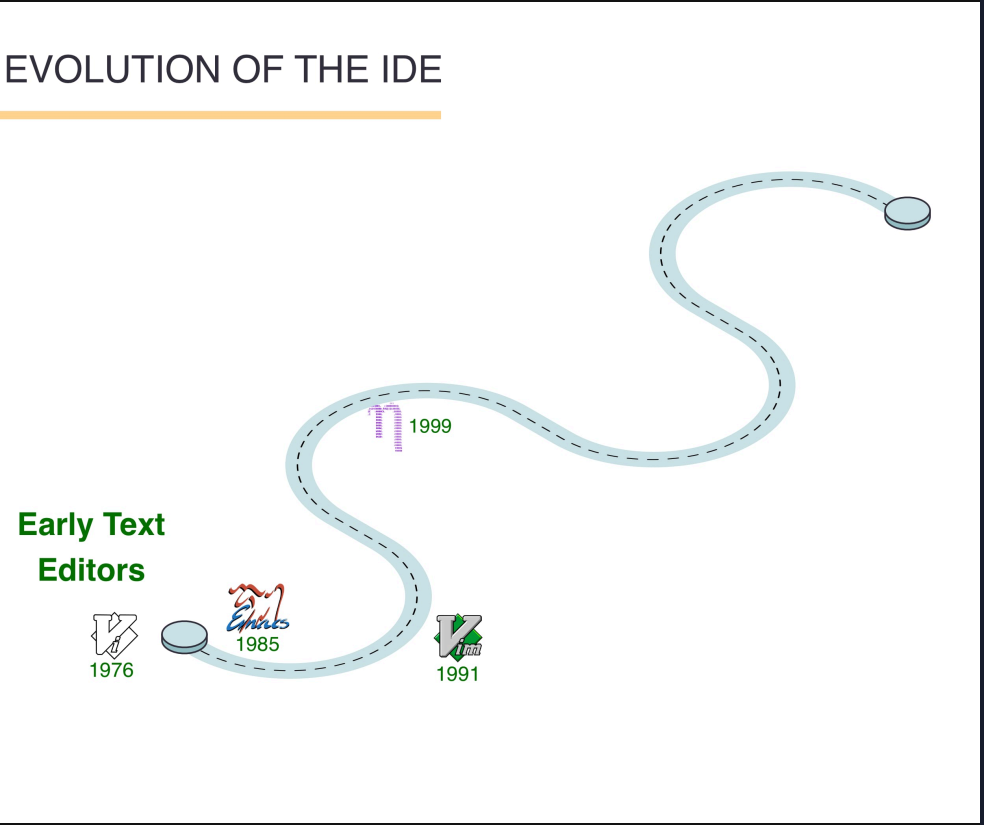
Evolution of the IDE

Early text editors



USER FRIENDLY by Illiad
HI. I HAVE LINUX INSTALLED ON MY MACHINE BUT I DON'T KNOW HOW TO USE THIS 'SIX' EDITOR.
'SIX' EDITOR?
http://www.userfriendly.org/
Copyright (c) 2000 Illiad
I'M SORRY M'AM. BUT I DON'T KNOW WHAT EDITOR THIS IS THAT YOU'RE TALKING ABOUT.
LOOK. IT'S THE 'SIX' EDITOR. YOU KNOW. THE ROMAN NUMERAL SIX?
UH. YOU MEAN 'VI'?
THIS IS THE SAME LADY WHO WAS WORRIED BECAUSE SHE HEARD WE USED "A PATCHY" WEB SERVER.

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

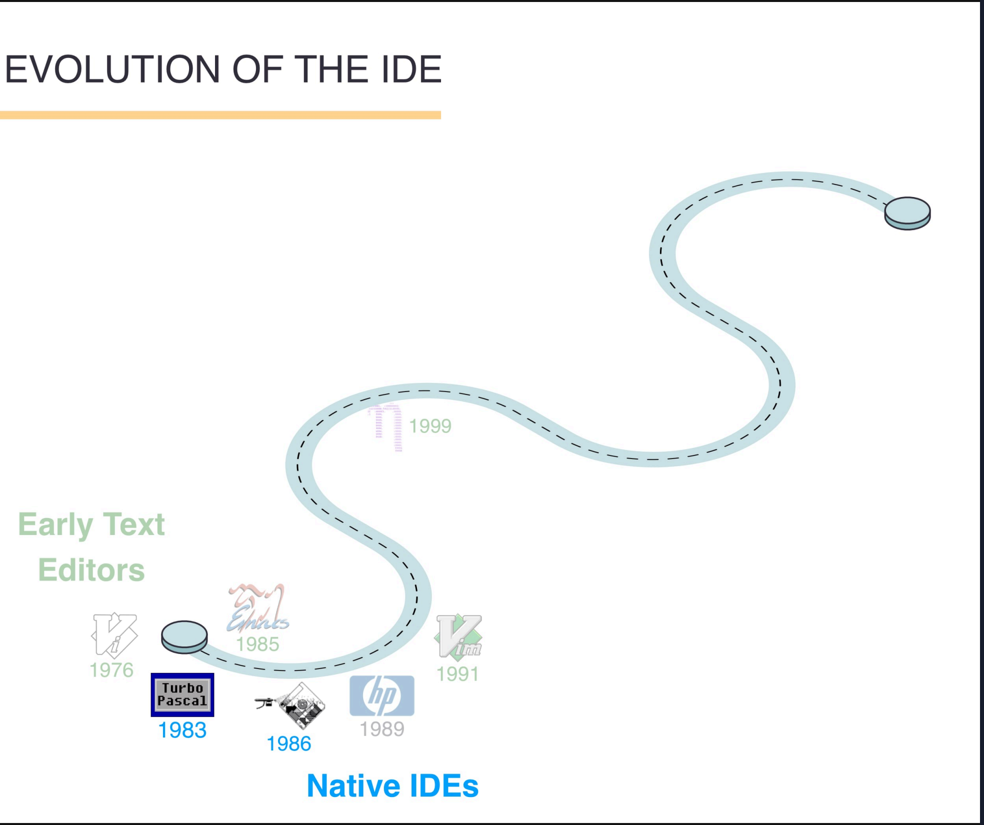


EVOLUTION OF THE IDE

Evolution of the IDE

Early text editors

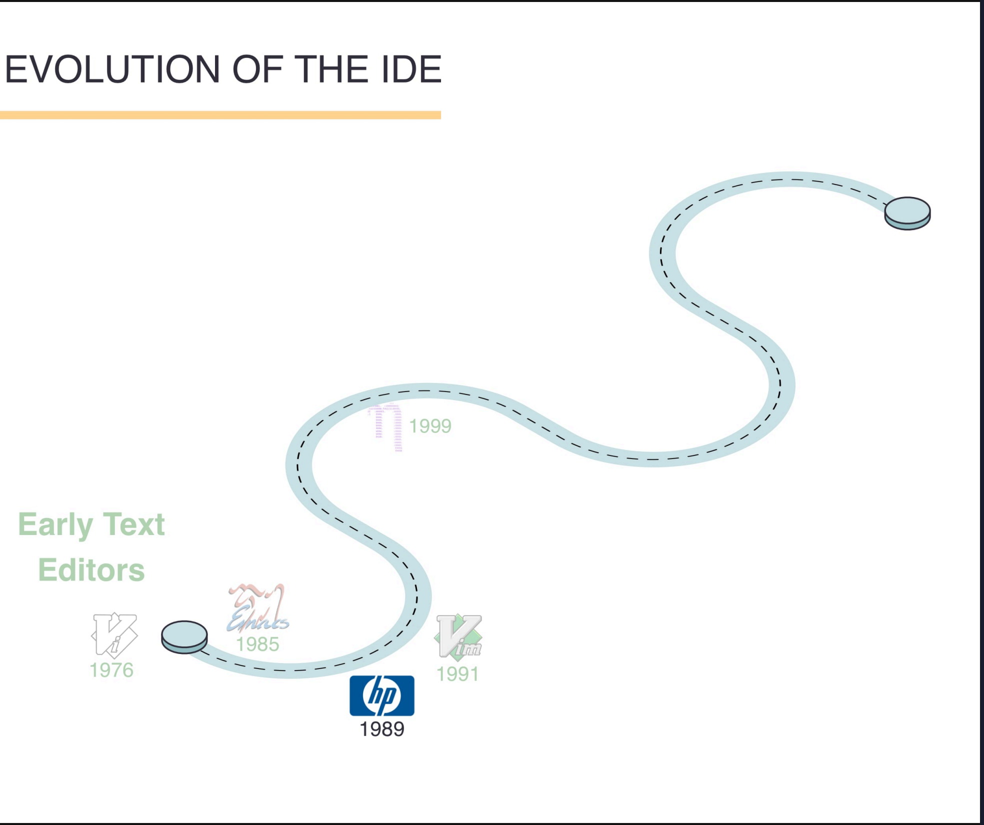
- 1976: Vi
- 1985: Emacs
- 1991: Vim
- 1999: nano


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

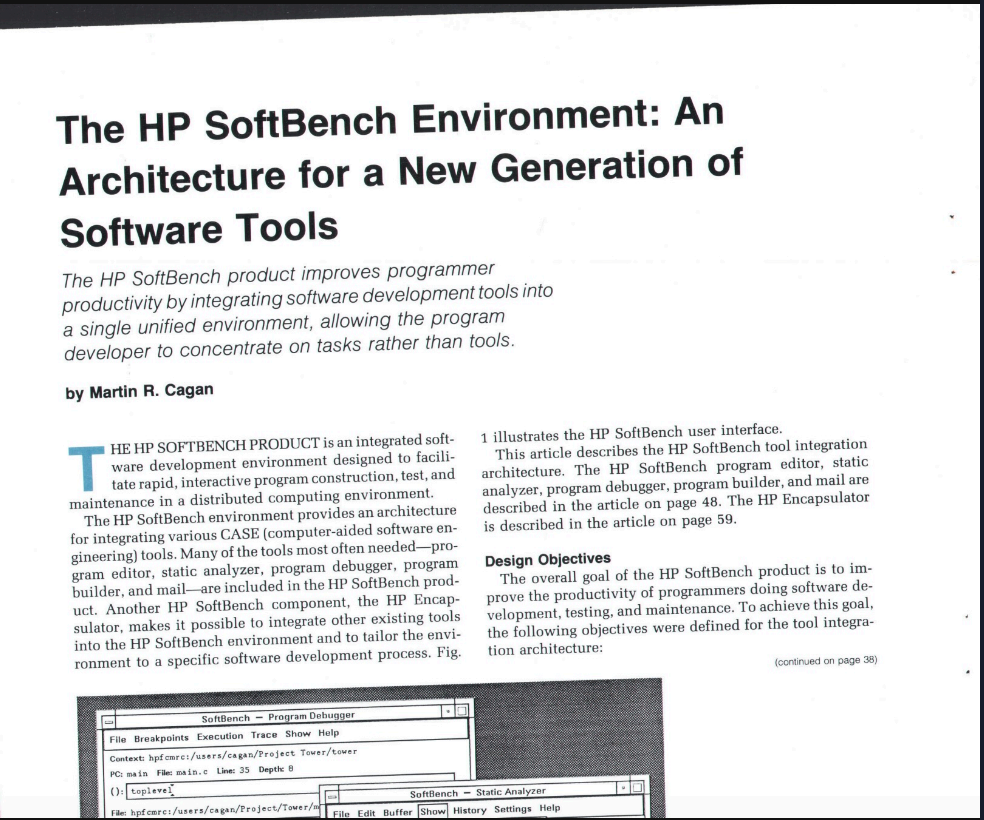
First plug-in IDE

Evolution of the IDE

First plug-in IDE

HP Softbench

HEWLETT PACKARD
Installing and Customizing SoftBench Products
C and C++ SoftBench User's Guide



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)

```
SoftBench — Program Debugger
File  Breakpoints  Execution  Trace  Show  Help
Context: hpfcmrc:/users/cagan/Project Tower/tower
PC: main  File: main.c  Line: 35  Depth: 8
(): toplevel
File: hpfcmrc:/users/cagan/Project/Tower/m

SoftBench — Static Analyzer
File  Edit  Buffer  Show  History  Settings  Help
```

Evolution of the IDE

First plug-in IDE

HP Softbench

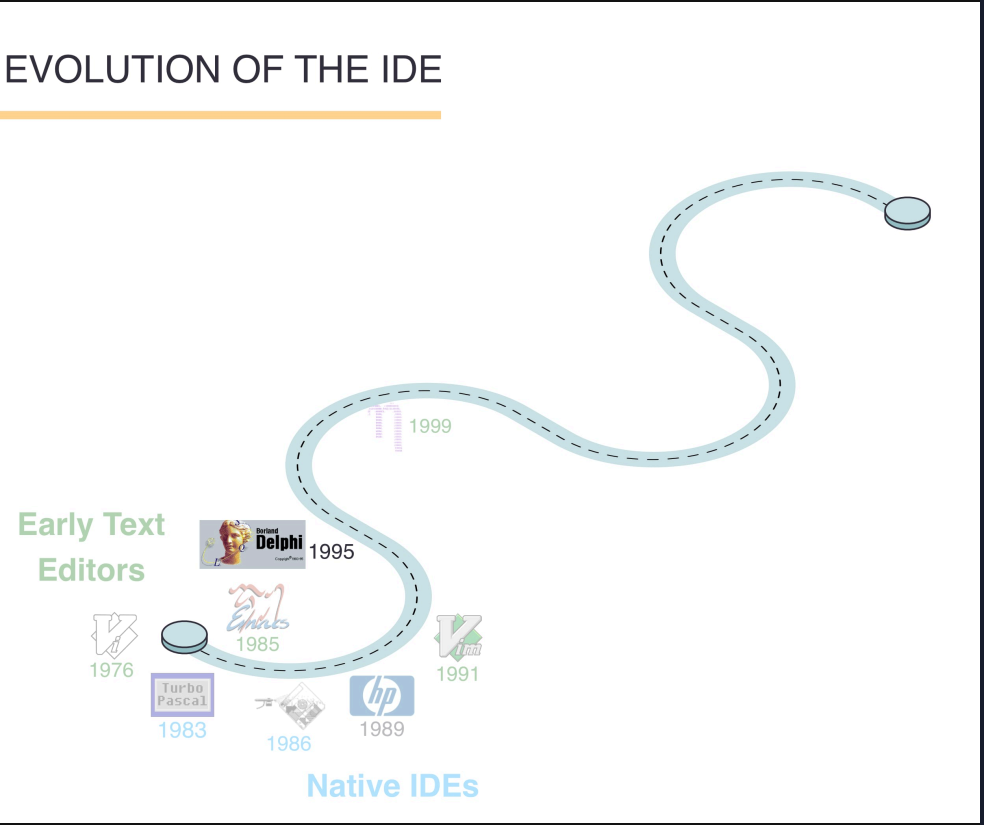
REF: HP Journal, June 1990 edition

Evolution of the IDE

Early Reviews

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

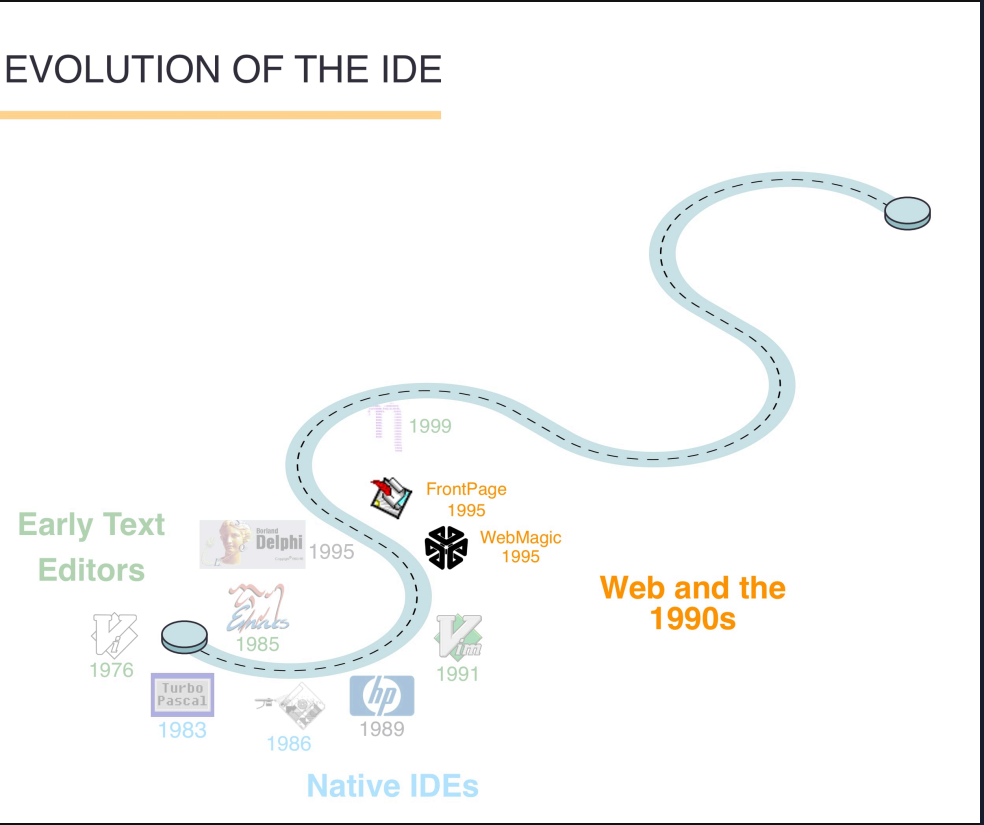
REF: *Computerwoche* ("Computer Week", German counterpart of American magazine *Computer World*), 1995.



Evolution of the IDE

Cross-platform in the 1990s

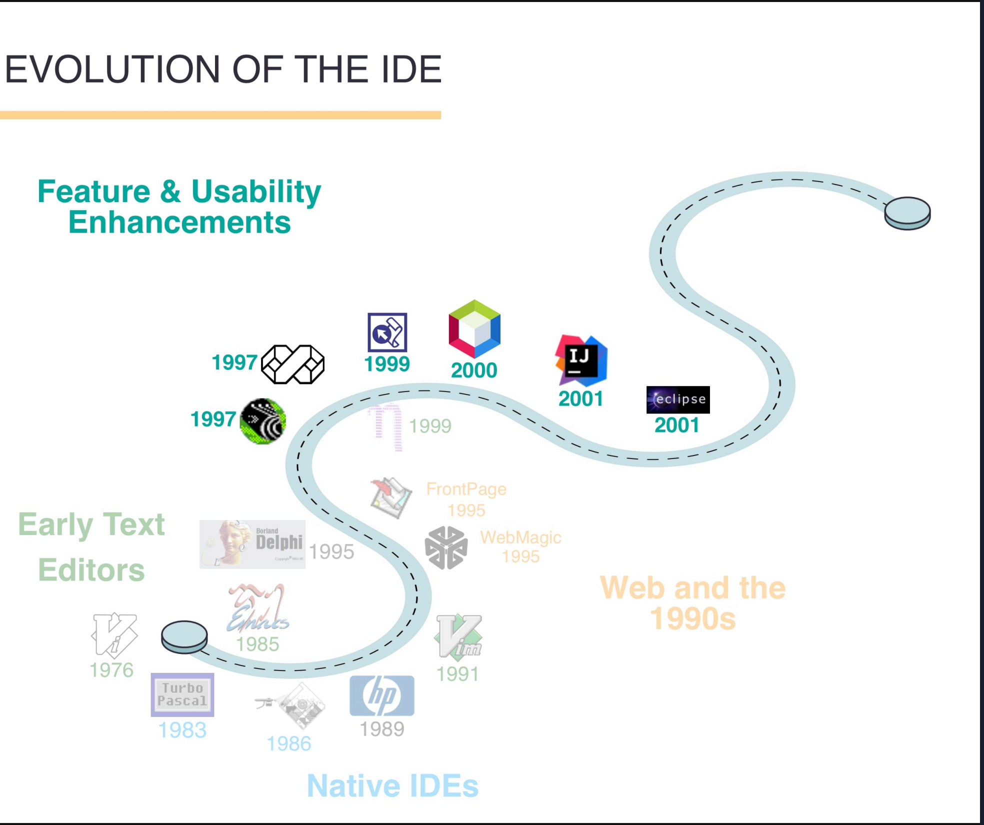
1995: Borland Delphi


EVOLUTION OF THE IDE

Evolution of the IDE

The Web and the 1990s

- 1995: SGI WebMagic
- 1995: Microsoft FrontPage

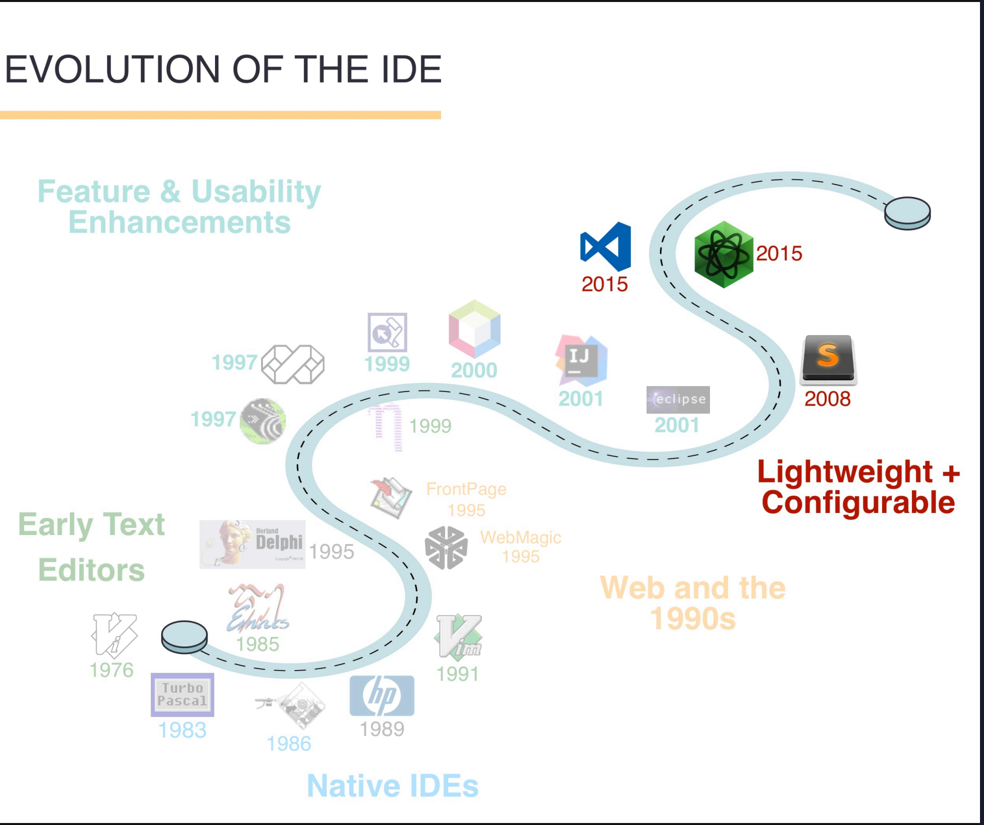


Evolution of the IDE

Features & Usability

Late 1990s to 2000s

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

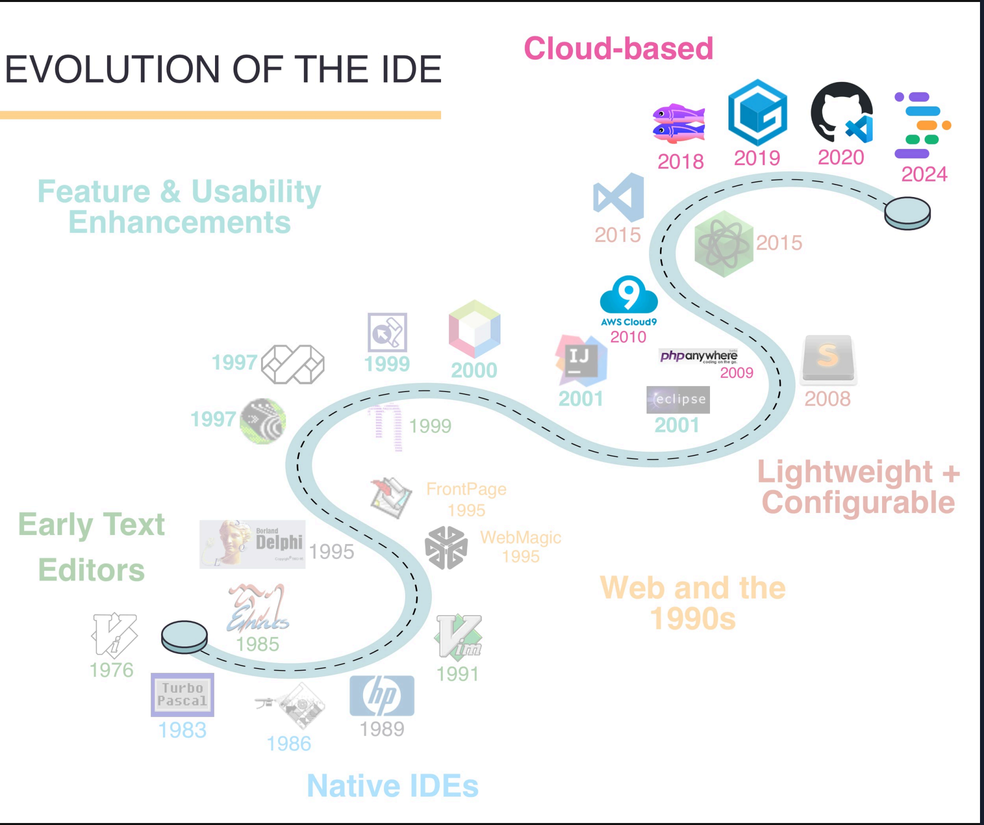


Evolution of the IDE

Lightweight & Configurable

2010s to Now

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



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

A result of DevEx

Things we never knew we needed…

From this:

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

Evolution of the IDE

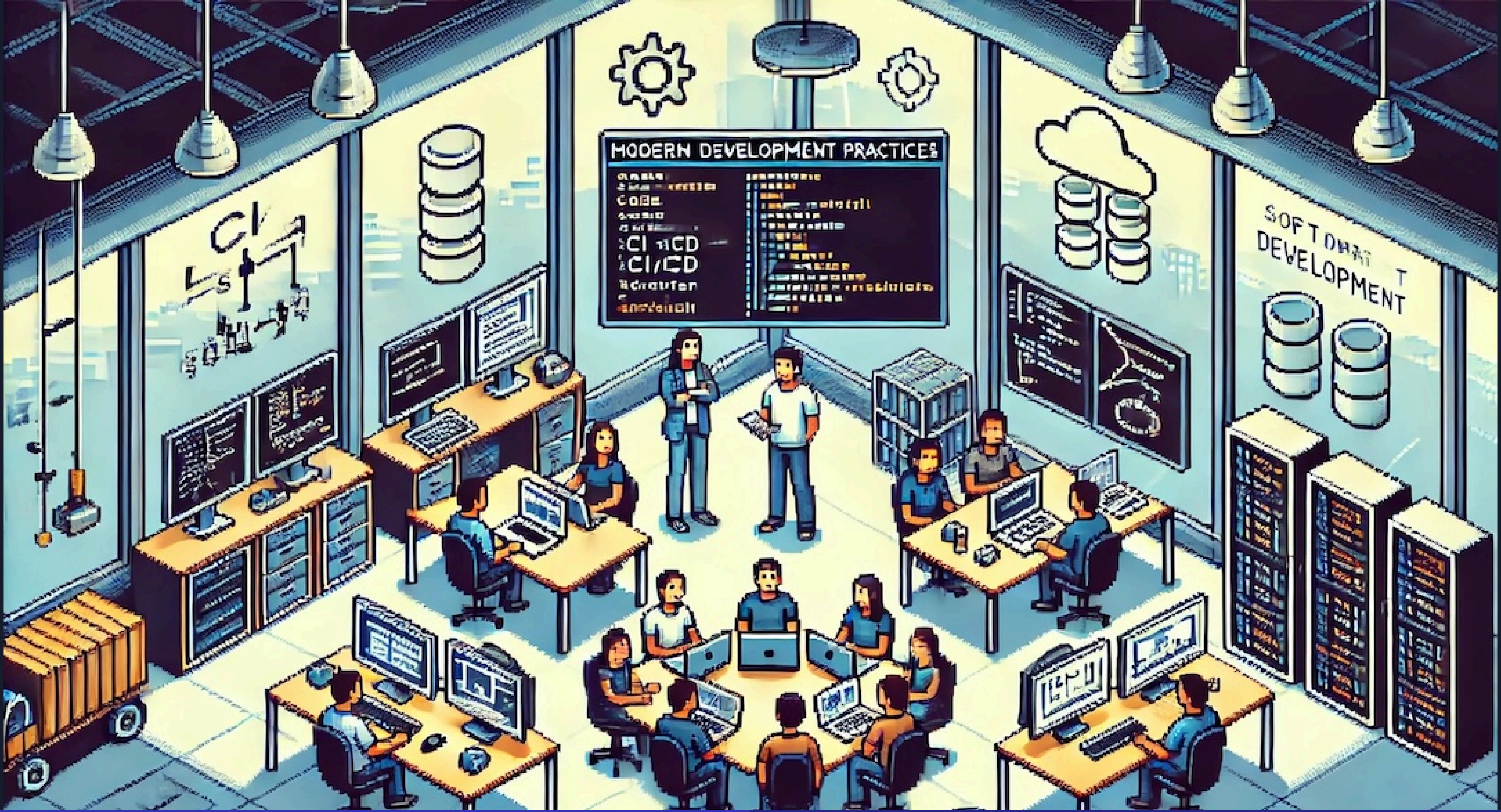
A result of DevEx

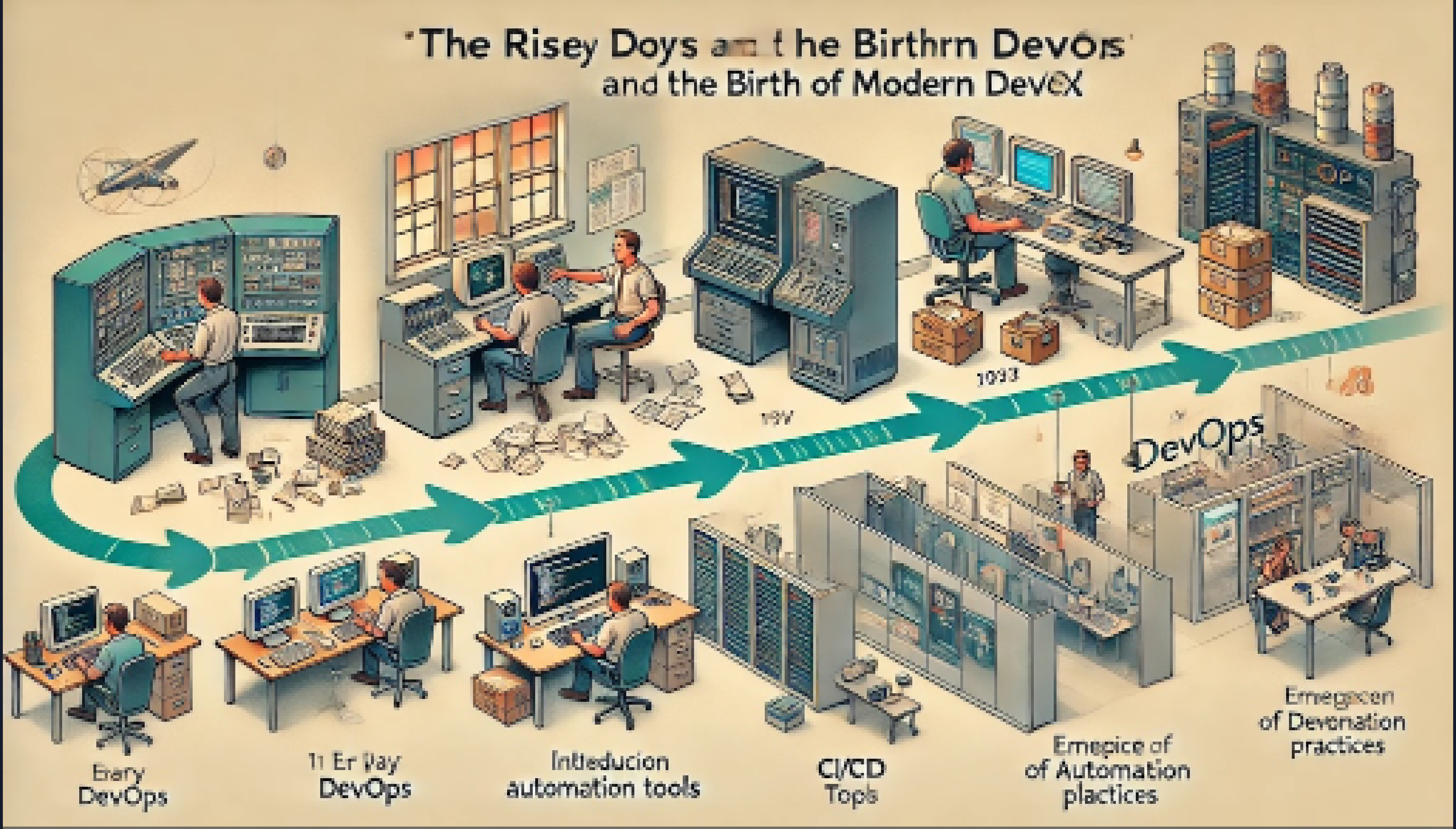
Things we never knew we needed…

To this:

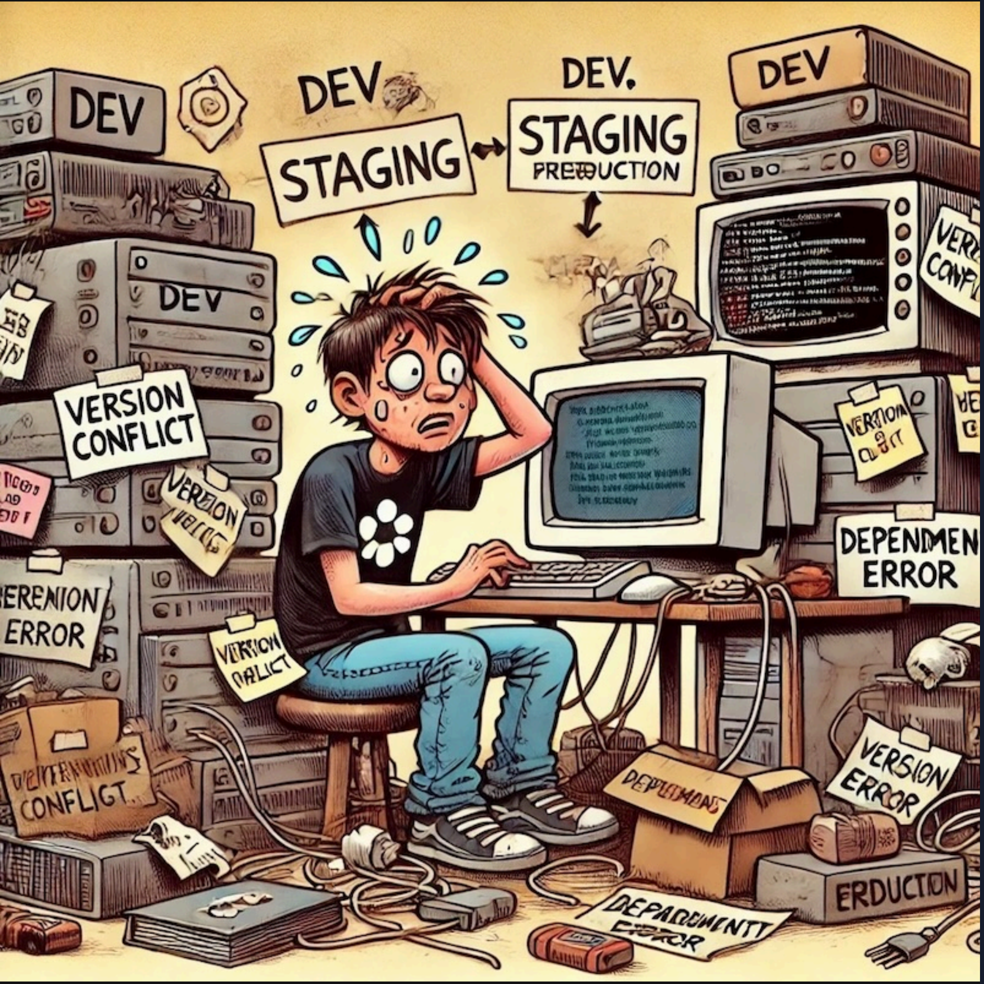
- Code completion
- Code refactoring
- Syntax highlighting
- Debugging
- VCS integration (no more FTPing files around)
- Multi-language support
- Framework integration
- Pair programming

Developer Experience
CD/CI PIPLINES
CID
CEYUD
CCD
1930
DEVOPS
2079


MODERN DEVELOPMENT PRACTICES
CODE
CI/CD
CI/CD
SOFTOWAT T
DEVELOPMENT



'The Risey Doys and the Birthrn Devops'
and the Birth of Modern DevEX
1993
19V
DevOps
Eary DevOps
11 Er Day DevOps
Intleducion automation tools
CI/CD Tops
Emepice of of Automation placlices
Emergcen of Devenation practices



Server Environment Setup

Manual config nightmares

Late 1990s to Early 2000s

Server Environment Setup

Config Mgmt & Containerization

Mid-2000s to 2010s

Server Environment Setup

IaC and DevOps Integration

2010s to Present

HashiCorp Terraform

GitHub Actions

Broader Impact of DevEx

- Deployment pipelines
- Infrastructure as Code (IaC) practices
- Developer Efficiences

What is DevOps?

> the combination of practices and tools designed to increase an organization’s ability to deliver applications and services faster than traditional software development processes

DevOps Principles + DevEx alignment

- Collaboration

DevOps Principles + DevEx alignment

- Enhanced collaboration *via tools and processes*

DevOps Principles + DevEx alignment

- Collaboration
- Communication

DevOps Principles + DevEx alignment

- Enhanced collaboration *via tools and processes*
- Improving communication *via streamlined info sharing and feedback*

DevOps Principles + DevEx alignment

- Collaboration
- Communication
- Shared Responsibility

DevOps Principles + DevEx alignment

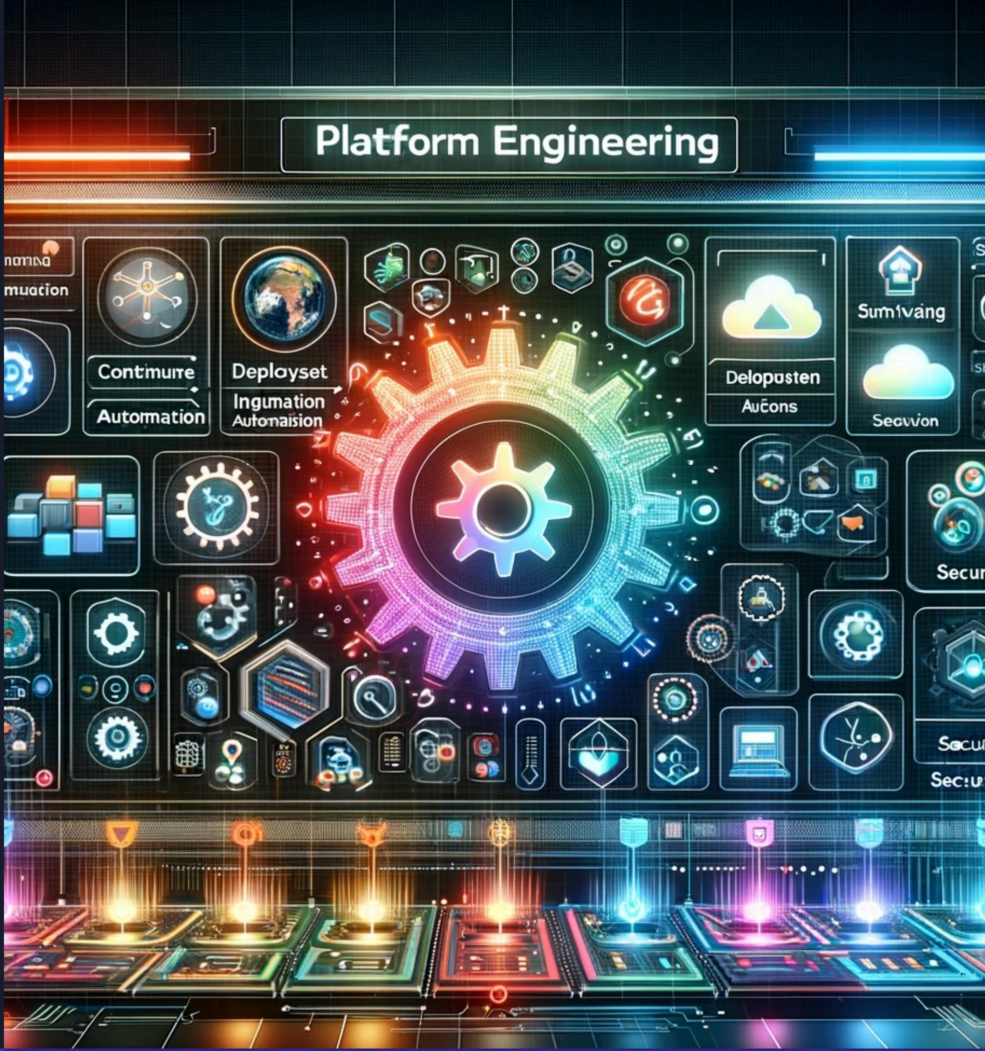
- Enhanced collaboration *via tools and processes*
- Improving communication *via streamlined info sharing and feedback*
- Shared responsibility *by empowering all teams with access and information*

Good DevOps == Good DevEx

- Facilitates smoother transitions between Dev and Ops
- Minimizes bottlenecks with enhanced collaboration
- Ensures feedback loops are efficient and productive
- Enables DevOps principles to take hold within an organization

The Rise of Platform Engineering

- Specific, integrated environments that devs need
- Abstract away infrastructre + backend complexities
- Access to robust, scalable, easy-to-use platforms
- Streamline development processes and reduced setup time



Self-Service Platforms

- Developers empowered with necessary tools
- Leverage automation, templates, policies with agility
- Accelerate development, enhance productivity, foster autonomy

Better Practices for leveling up DevEx

- Empower with the right tools
- Encourage Cross-functional Teams
- Implement Feedback Loops
- Focus on Automation
- Invest in Training and Development

Strategies for Improving DevEx

DevEx reflects an organizational culture

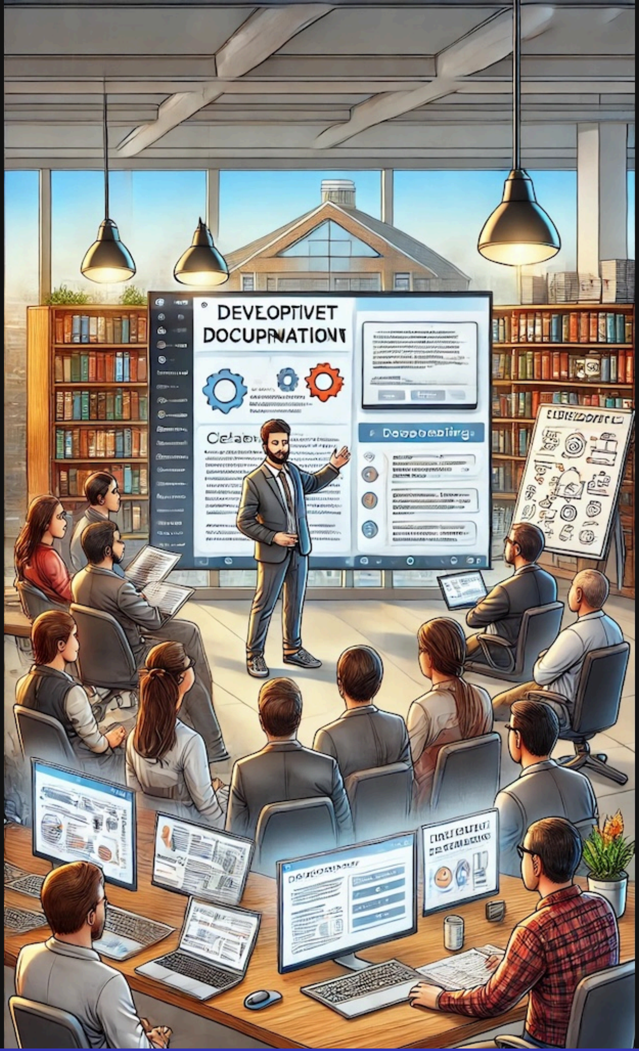
Strategies for Improving DevEx

Improving DevEx in your organization

DevEx initiatives should be modeled from Leadership *FIRST*

Improving DevEx in your organization

1. Foster a positive culture
2. Streamline the workflow(s)



Improving DevEx

Foster a positive culture

1. Clear and concise documentation

- Encourage knowledge sharing
- Create easily accessible resources to reduce toil + empower

Improving DevEx

Foster a positive culture

1. Clear and concise documentation
2. Promote collaboration and communication

- Facilitate code reviews
- Implement comms to foster teamwork + problem solving

Improving DevEx

Foster a positive culture

1. Clear and concise documentation
2. Promote collaboration and communication
3. Champion well-being and growth

- Encourage feedback, up and down
- Recognize achievements
- Create a sense of belonging

Improving DevEx

Streamline the workflow

1. Tools and Automation

- Explore tools which are highly regarded in your field
- Automate repetitive tasks wherever possible

Improving DevEx

Streamline the workflow

1. Tools and Automation
2. Standardize environment setup

- Use config management tools
- Streamline onboarding for all team members

Examples:

JFrog ARTIFACTORY JFrog PIPELINES ATLASSIAN ANSIBLE

DevEx is...

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

Thank you!

@jerdog.dev

/in/jeremymeiss

@jerdog

@jerdog@hachyderm.io

@IAmJerdog

jmeiss.me

END