Developer Experience…

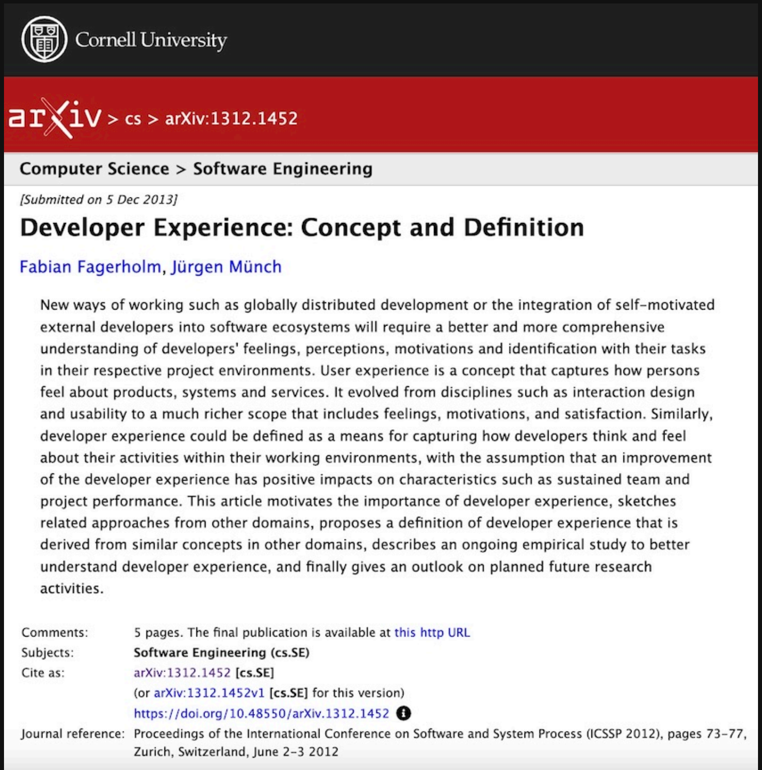
Central to DevOps Success

@IAmJerdog

What is Developer Experience?

DevEx is *more* than just your parent's DevOps lifecycle

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 ℹ
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.

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 International Conference on Software and System Process (ICSSP), Zurich, Switzerland, 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 International Conference on Software and System Process (ICSSP), Zurich, Switzerland, 2012.

Point of clarification

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

Workshop some new terms

Workshop some new terms

"DevOps Experience"

Workshop some new terms

"DevOps Experience"

"DevExOps"

Workshop some new terms

"DevOps Experience"

"DevExOps"

"DevOpsEx"

Workshop some new terms

"DevOps Experience"

"DevExOps"

"DevOpsEx"

?????

Jeremy Meiss

Co-Founder, DevEx Consultant

DevEx Institute

https://DevEx.Institute

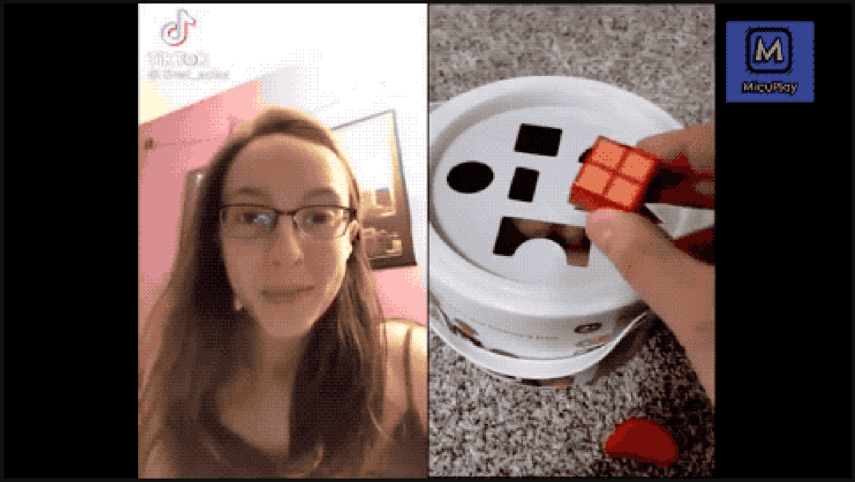
DevOpsDays Kansas City Organizer

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."*

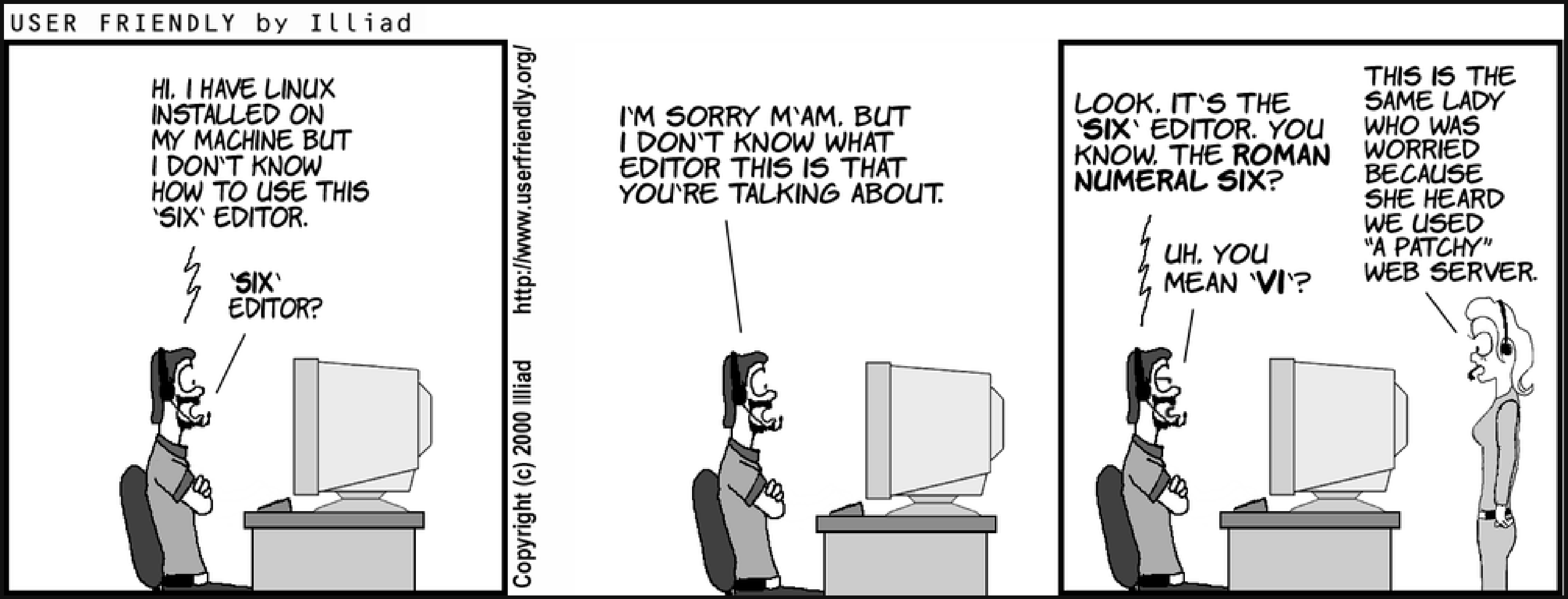
-Jessica West, Co-Founder, DevEx Institute

DevEx includes every interaction a developer/ops practitioner has with systems, tools, and processes



The evolution of the IDE

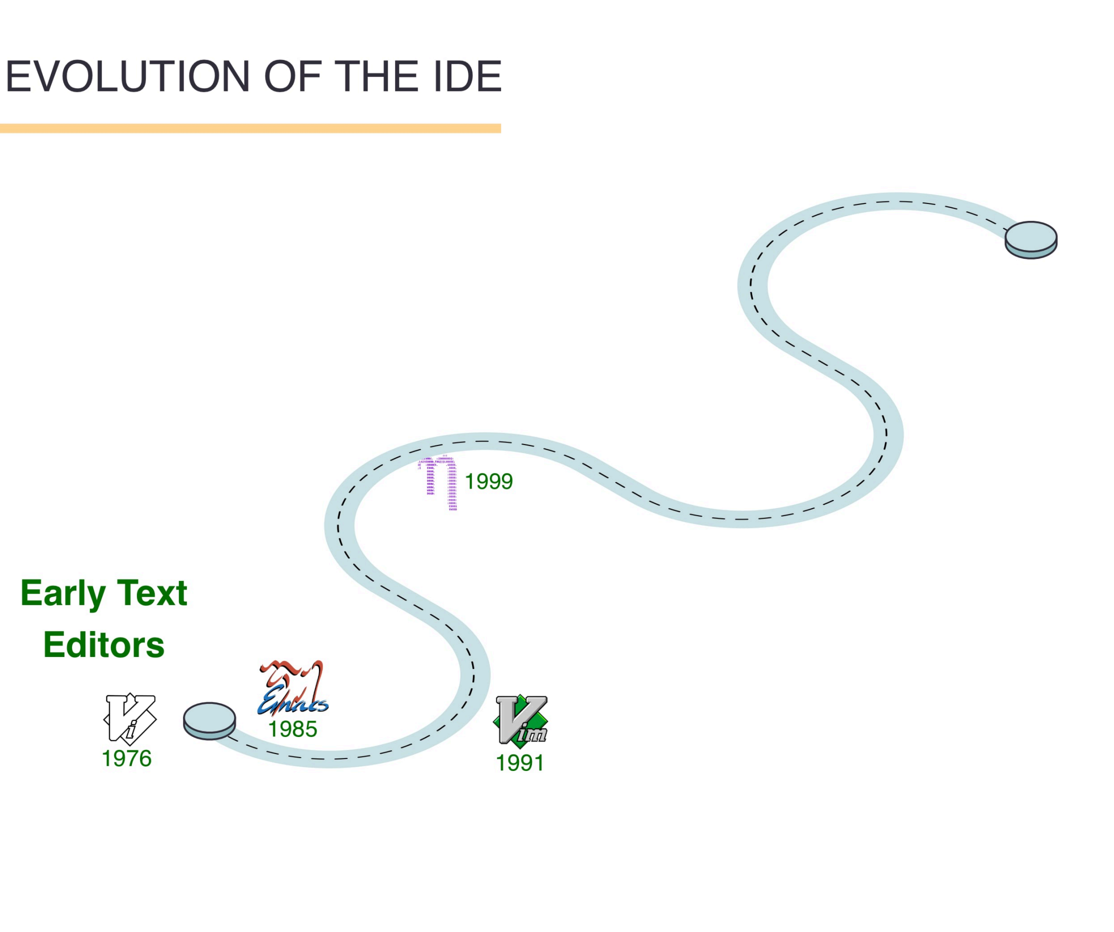
Early text editors


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"

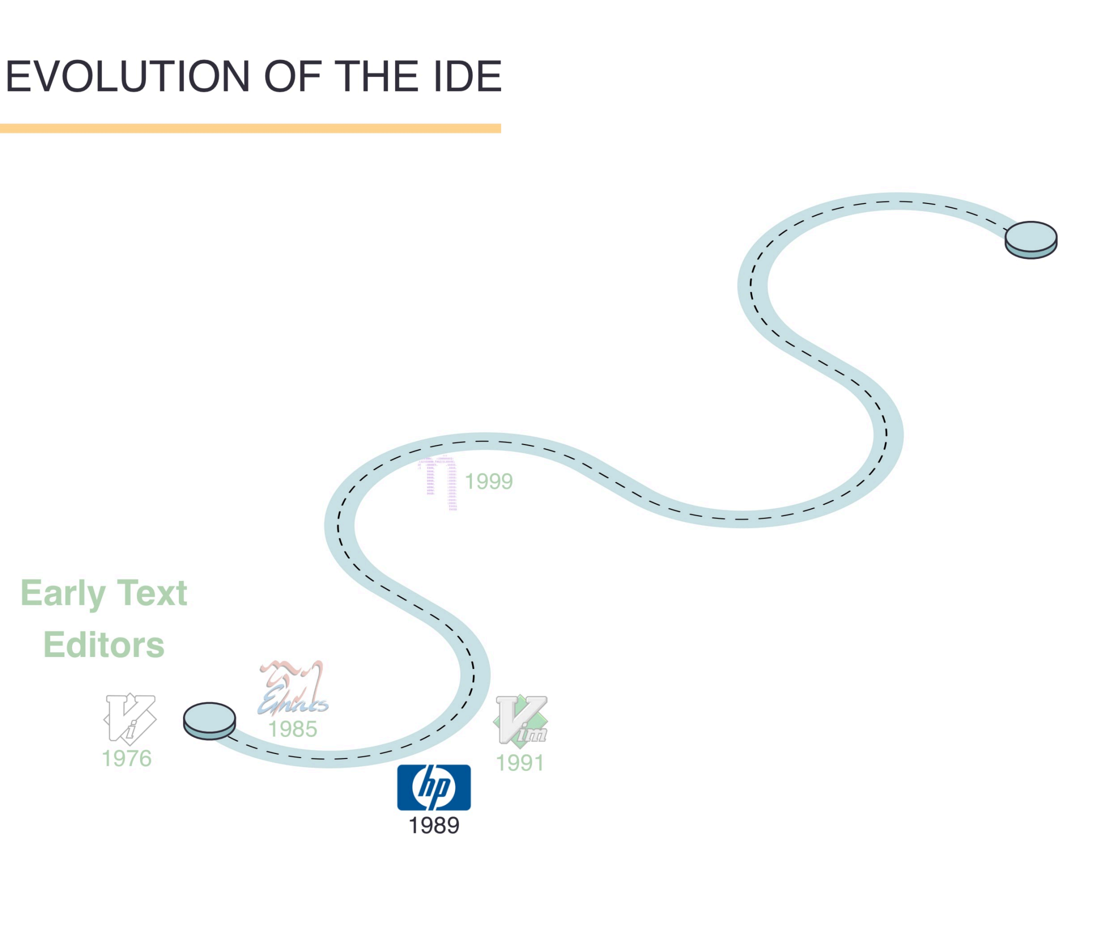
The evolution of the IDE

Early text editors



The evolution of the IDE

First plug-in IDE

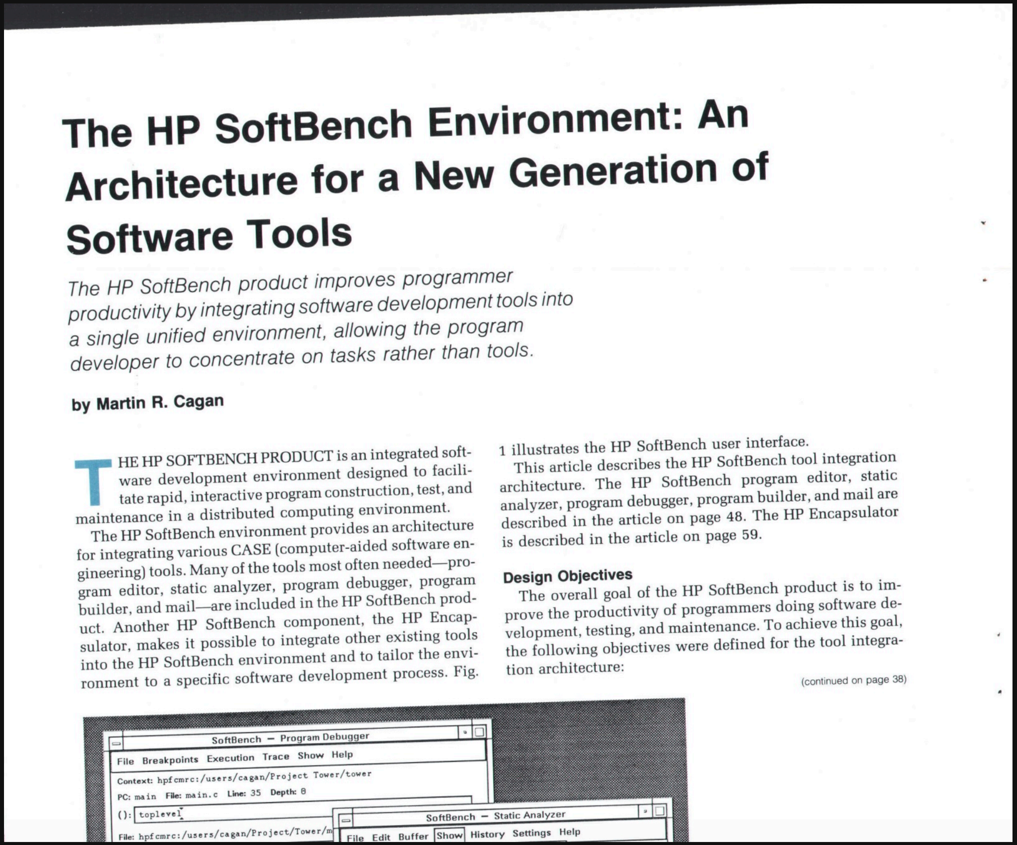


EVOLUTION OF THE IDE

The evolution of the IDE

First plug-in IDE

HEWLETT PACKARD
Installing and Customizing SoftBench Products
HEWLETT PACKARD
Installing and Customizing SoftBench Products
HEWLETT PACKARD
C and C++ SoftBench User's Guide
HEWLETT PACKARD
Installing and Customizing SoftBench Products
HEWLETT PACKARD
Installing and Customizing SoftBench Products
HEWLETT PACKARD
Installing and Customizing SoftBench Products
HEWLETT PACKARD
C and C++ SoftBench User's Guide
HEWLETT PACKARD
Installing and Customizing SoftBench Products
HEWLETT PACKARD
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
SoftBench — Static Analyzer
File Edit Buffer Show History Settings Help

```
Context: hpfcmrc:/users/cagan/Project Tower/tower
PC: main   File: main.c   Line: 35   Depth: 0
(): toplevel
File: hpfcmrc:/users/cagan/Project/Tower/m
```

REF: HP Journal, June 1990 edition

The evolution of the IDE

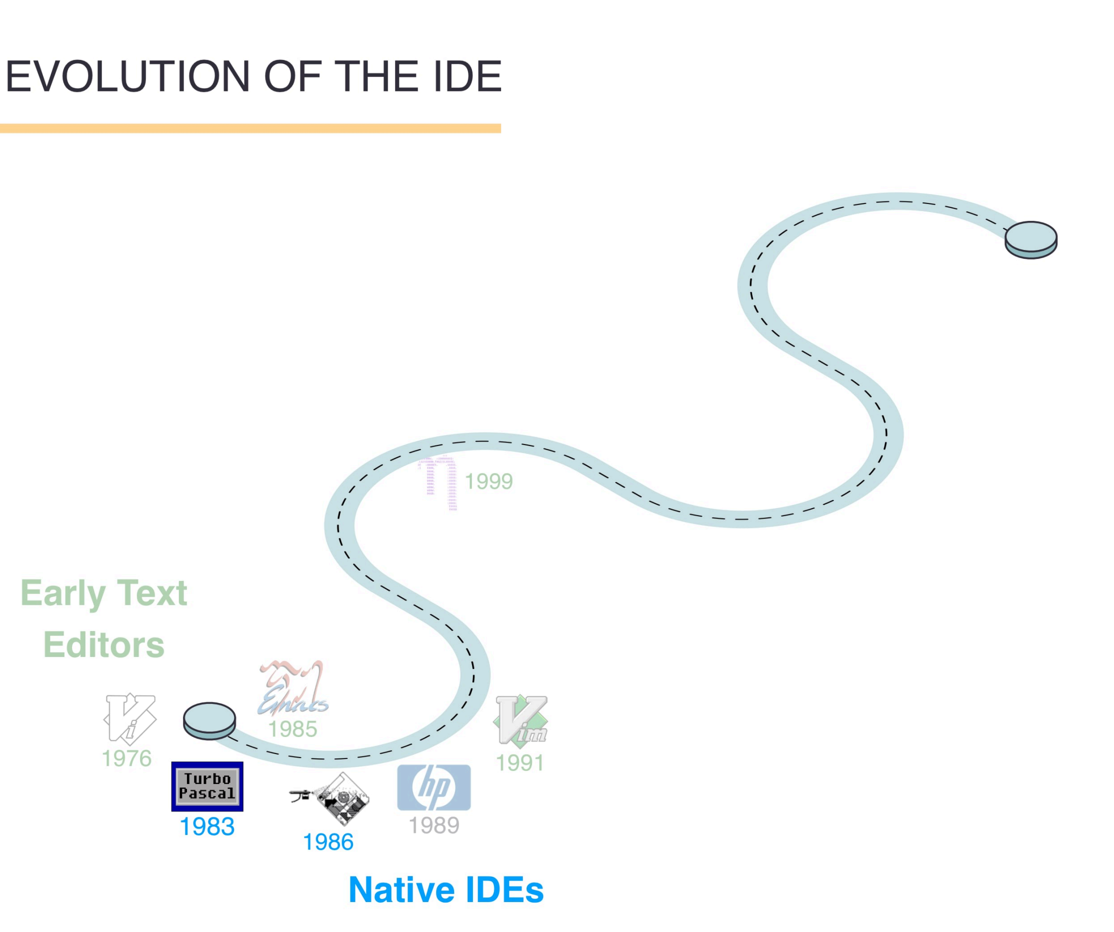
Early reviews

> *"...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.

The evolution of the IDE

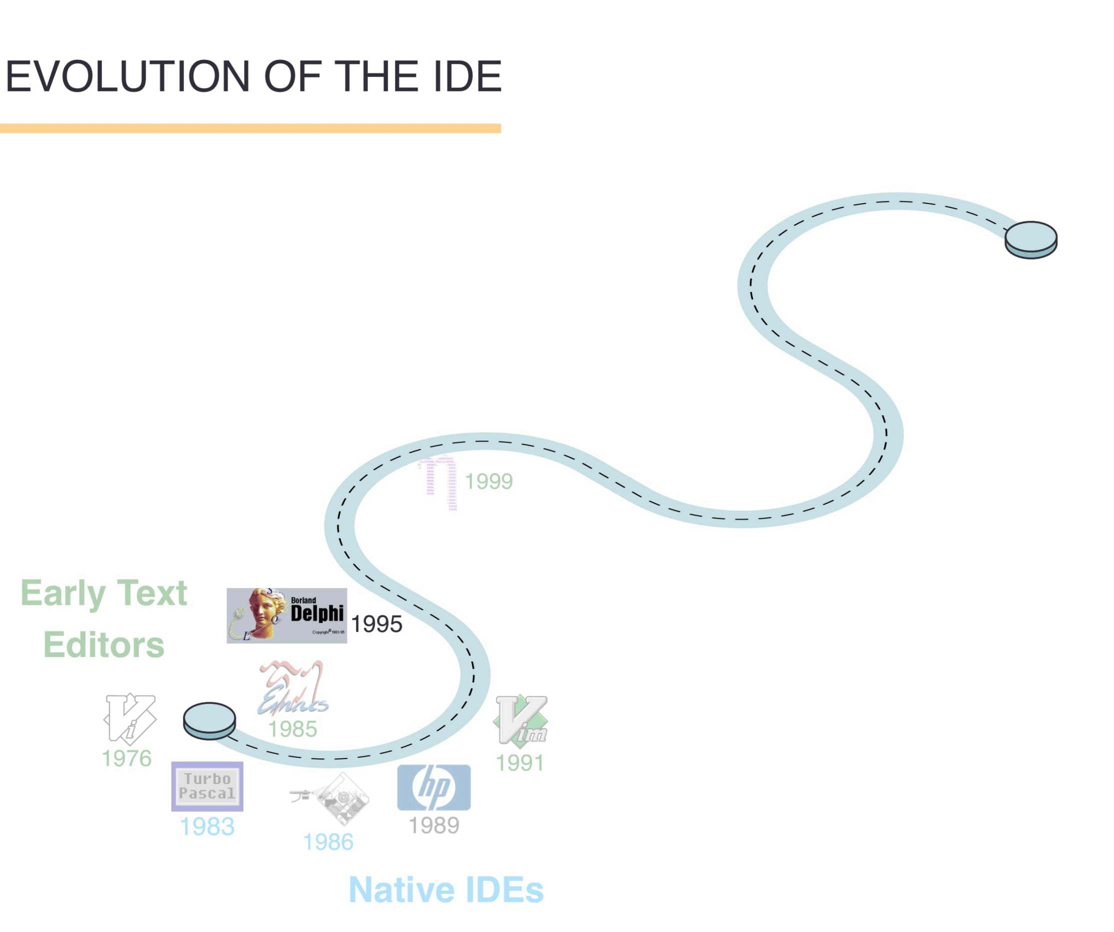
Native IDEs in the 1980s



EVOLUTION OF THE IDE

The evolution of the IDE

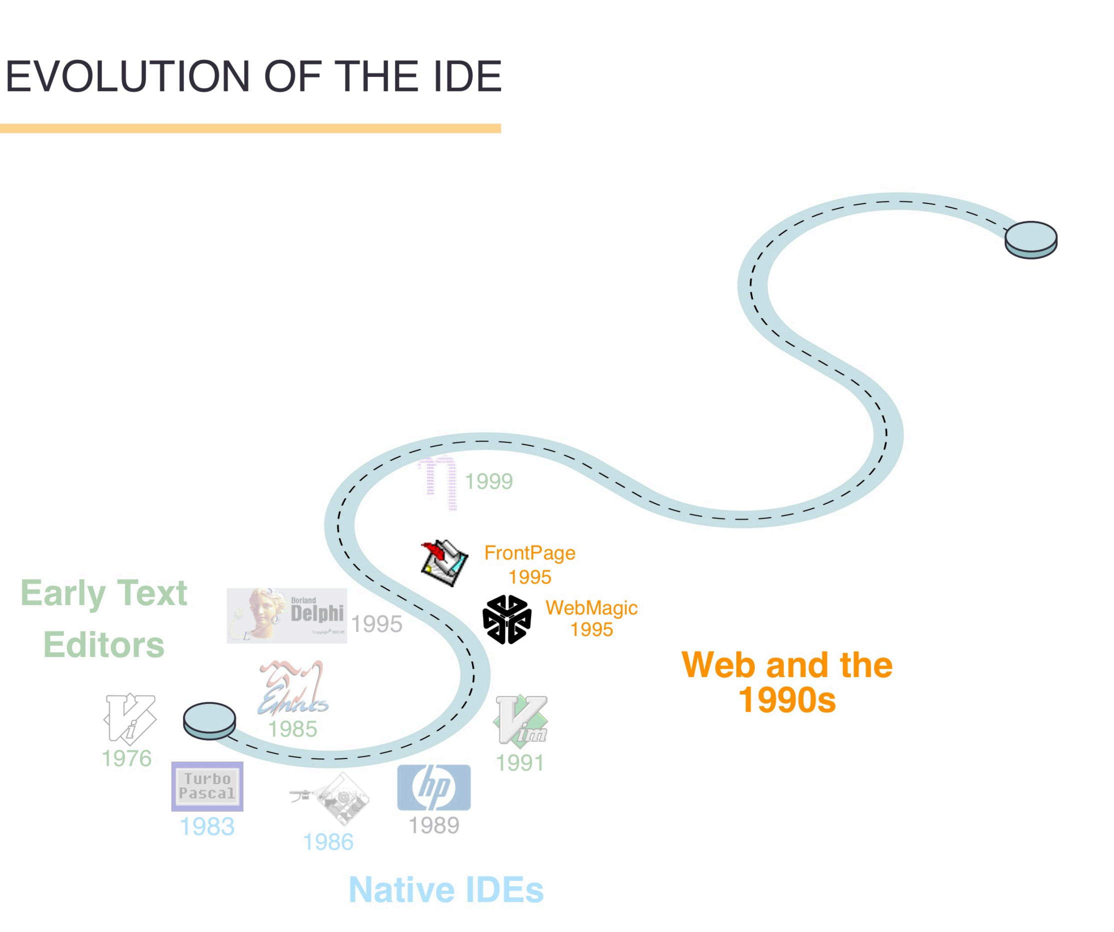
Cross-platform in the 1990s



EVOLUTION OF THE IDE

The evolution of the IDE

The Web and the 1990s



EVOLUTION OF THE IDE

Early Text Editors

1976

1983

1985

1986

1989

1991

Native IDEs

1995

FrontPage 1995

WebMagic 1995

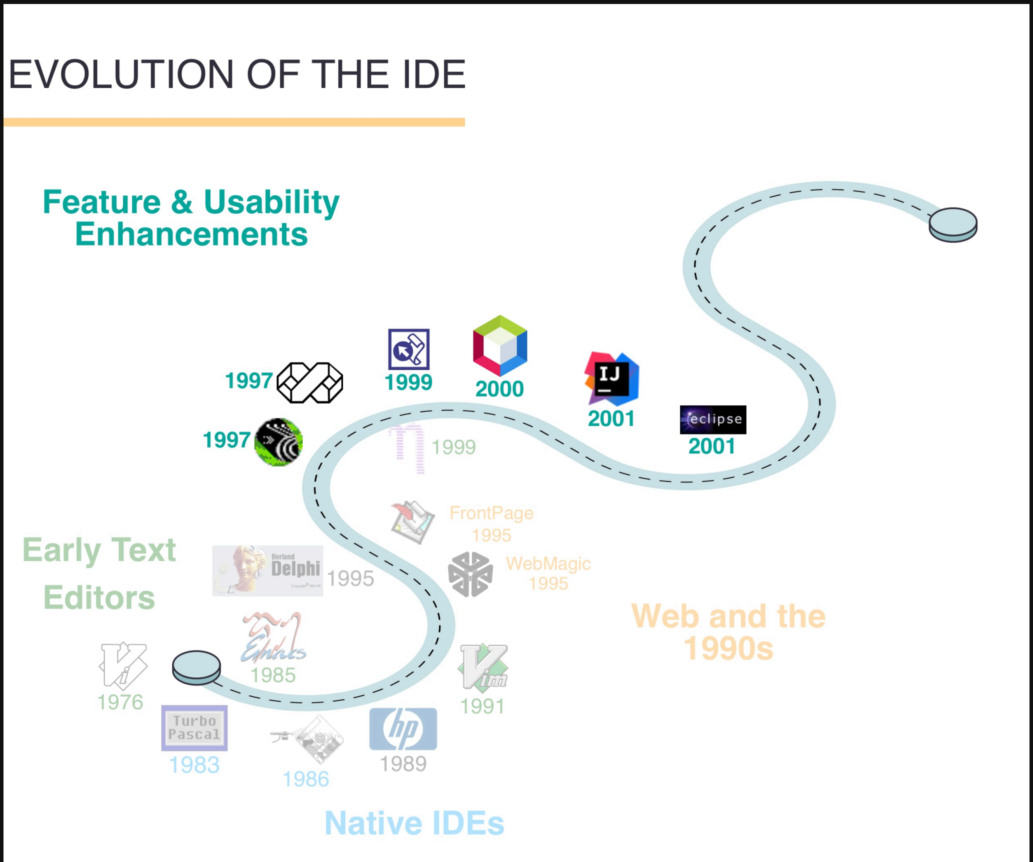
1999

Web and the 1990s

The evolution of the IDE

Feature & Usability Advancements

(Late 1990s to 2000s)

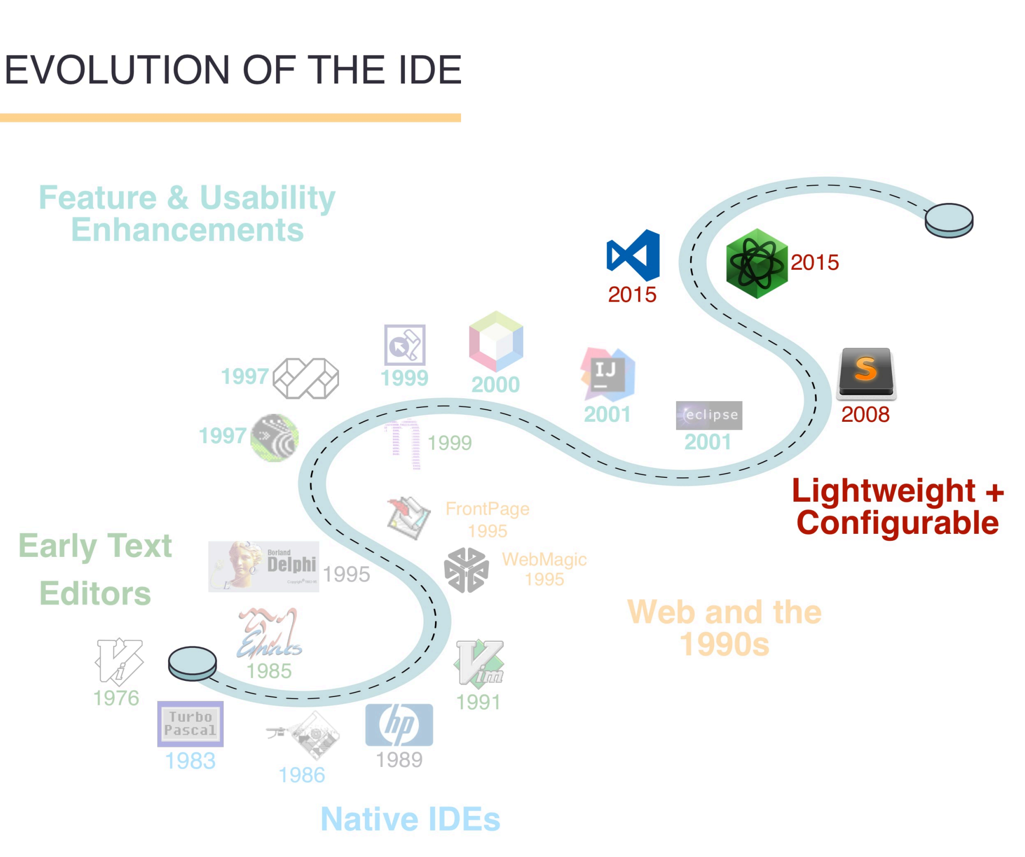


EVOLUTION OF THE IDE

The evolution of the IDE

Lightweight and configurable

2010s to Now



EVOLUTION OF THE IDE

Feature & Usability Enhancements

2015

2015

2008

Lightweight + Configurable

1997

1999

2000

2001

2001

1997

1999

FrontPage 1995

WebMagic 1995

Web and the 1990s

Early Text Editors

Delphi 1995

1985

1976

1991

Turbo Pascal 1983

1986

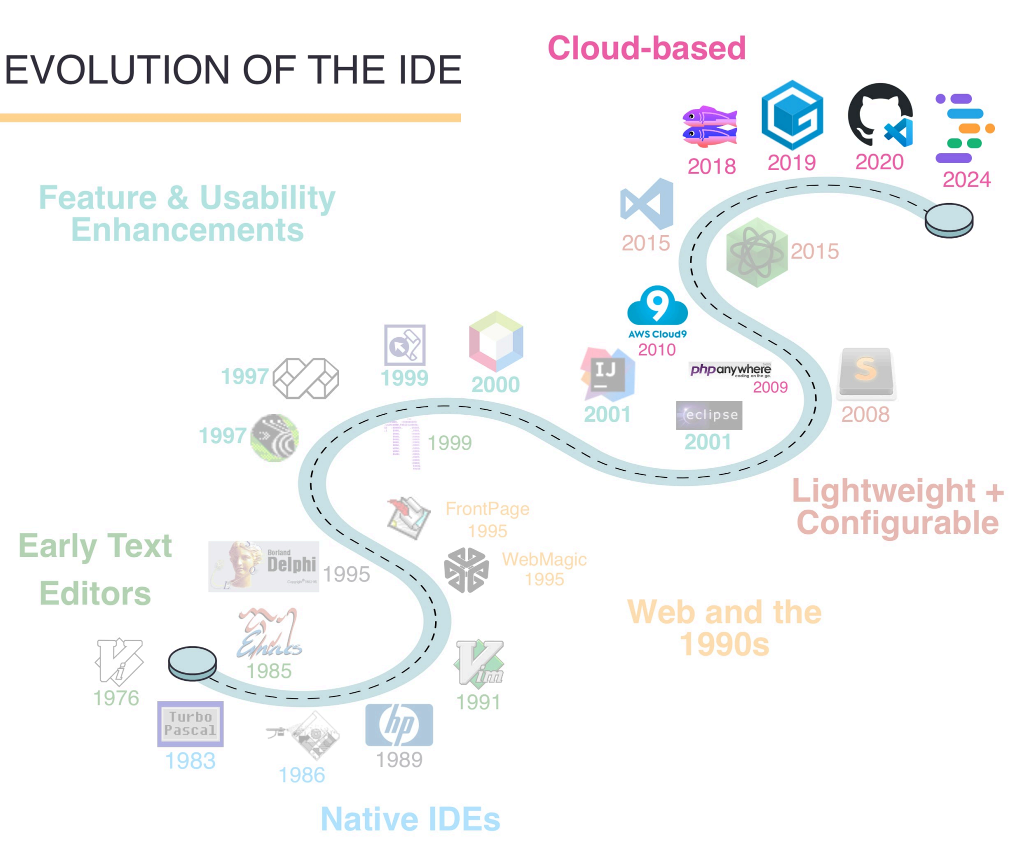
1989

Native IDEs

The evolution of the IDE

Cloud-based options

Now



IDEs are 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.”*

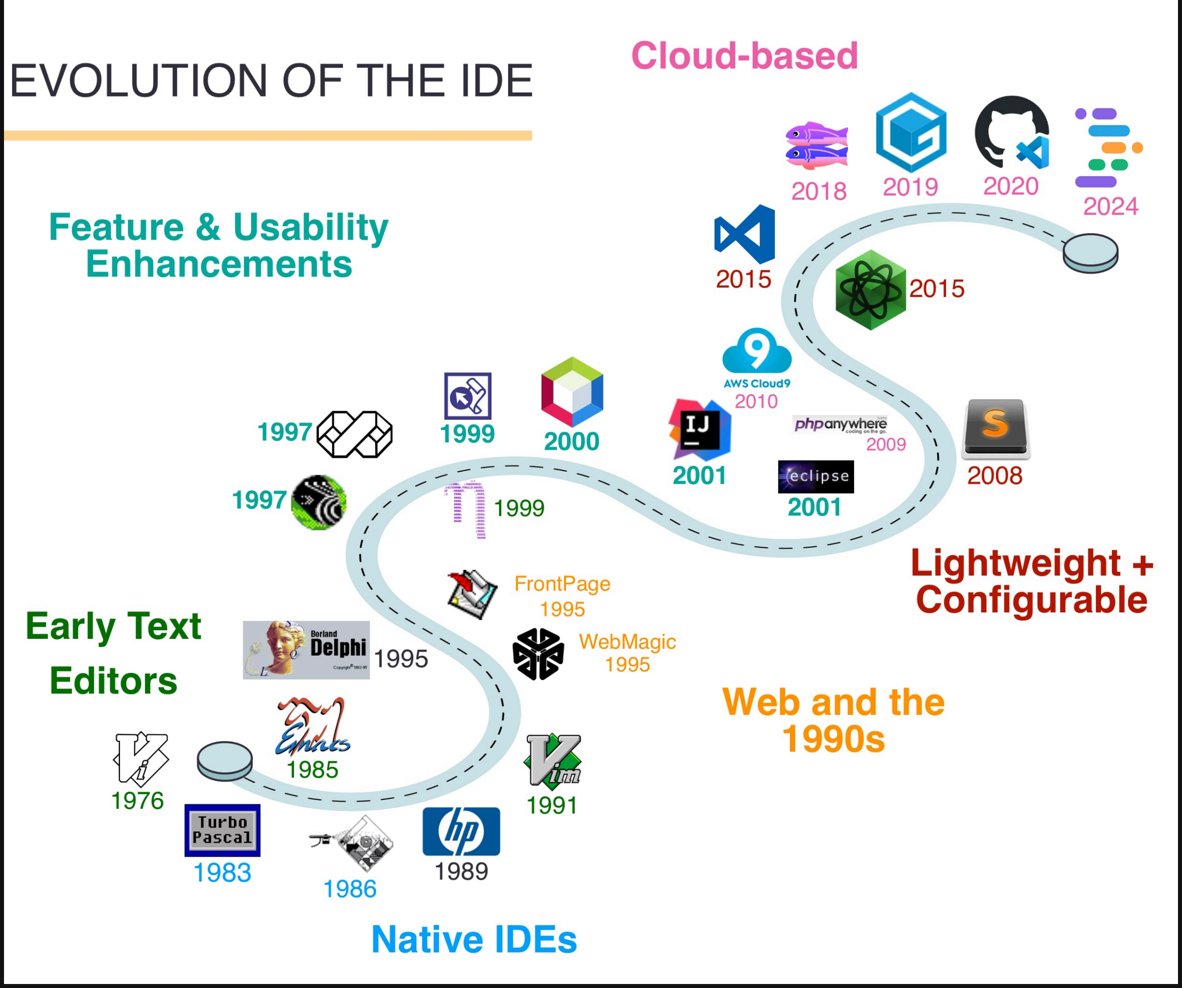
IDEs are a result of DevEx

Things we never knew we needed...

To this:

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

IDE Evolution



Server Environment Setup

Manual configuration

Late 1990s to Early 2000s

Server Environment Setup

Manual configuration

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

Broader Impact of DevEx

- Deployment Pipelines
- Infrastructure as Code (IaC) Practices
- Developer Efficiencies

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

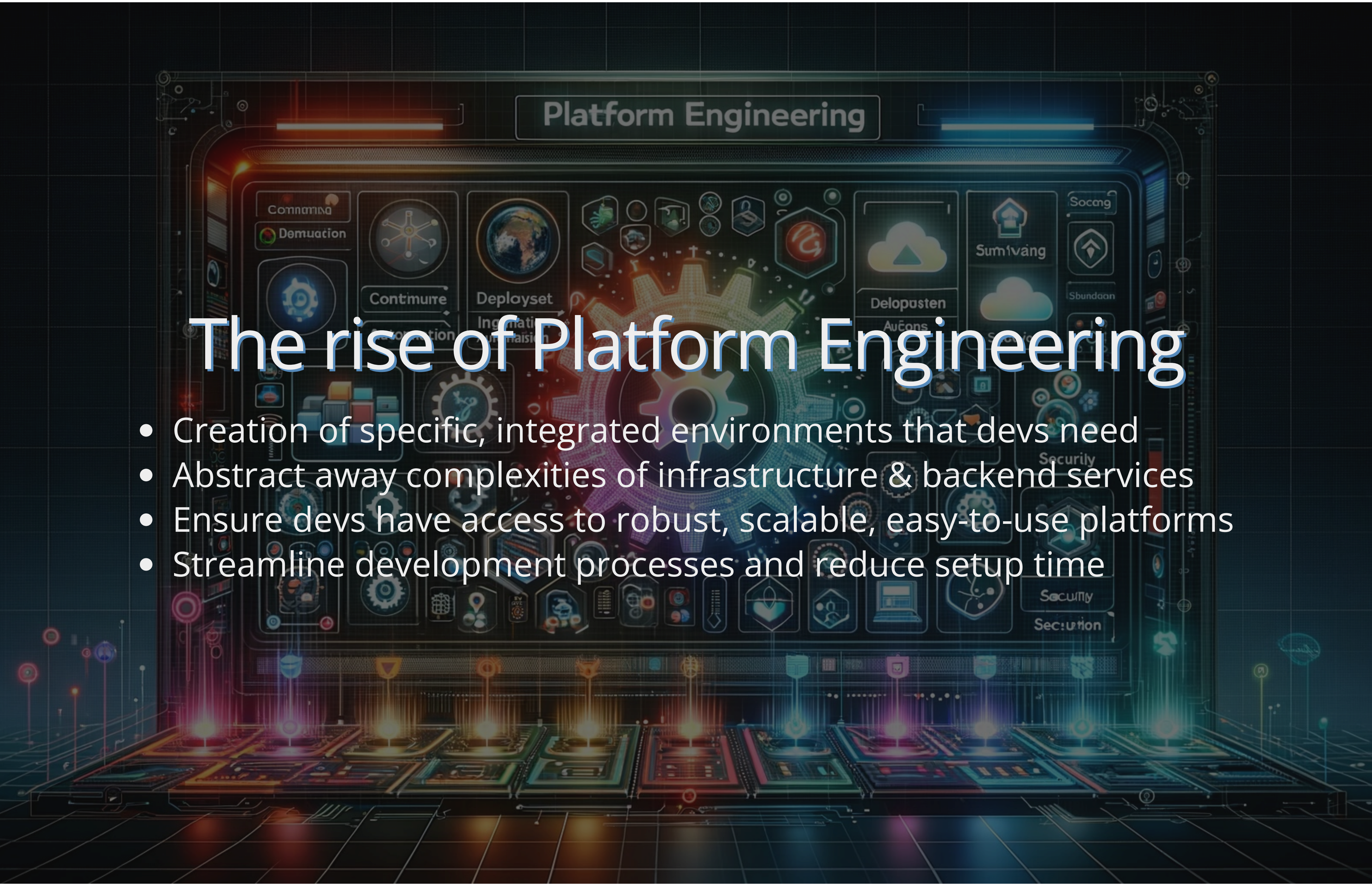
- Enhanced collaboration ***via tools and processes***
- Improving communication ***via streamlined info sharing and feedback***

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***



The rise of Platform Engineering

- Creation of specific, integrated environments that devs need
- Abstract away complexities of infrastructure & backend services
- Ensure devs have access to robust, scalable, easy-to-use platforms
- Streamline development processes and reduce setup time

Self-Service Platforms

- Empower developers to control their needs with necessary tools
- Leverage automation, templates, policies with agility
- Accelerate development, enhance productivity, and foster autonomy and innovation

Better Practices for leveling up your "DevOps Experience"

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

DevEx reflects an organizations values

> *If your company does not already have a process for gathering feedback (internal & external) on your product and/or the tools you use, you will not have a good Developer Experience, and I seriously question your commitment to it.*
>
> *— Jeremy's 'Working' 🇺🇸🇺🇦 (@IAmJerdog)*
> *January 25, 2024*

DevEx is…

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

Thank You.

in /in/jeremymeiss

@IAmJerdog

DEV @jerdog

@jerdog@hachyderm.io