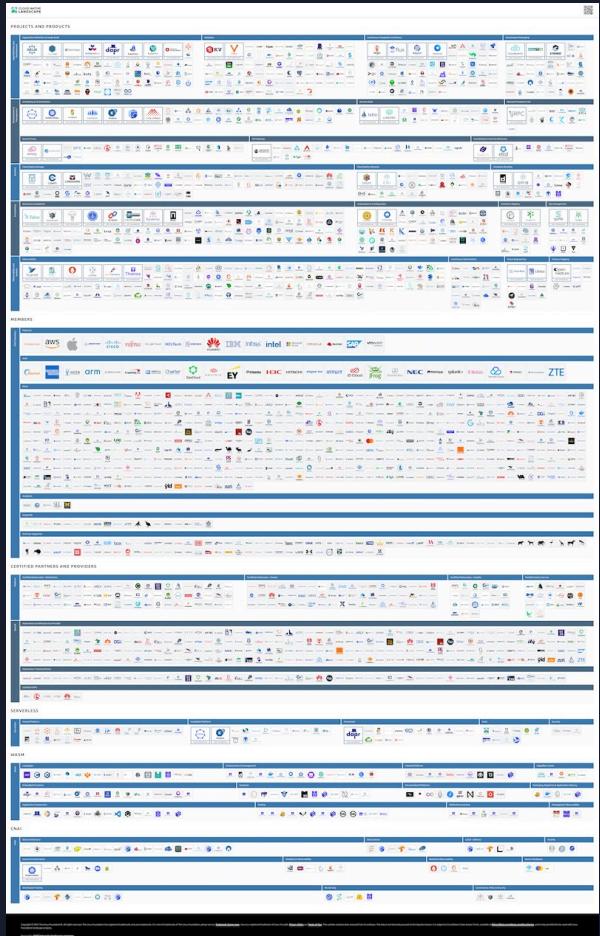


Streamlining DevEx

The Power of CI/CD Standardization and Interoperability



Jeremy Meiss

DevEx / DevRel Consultant
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







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,

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,

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.



[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,

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."

Point of clarification

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

The evolution of the IDE

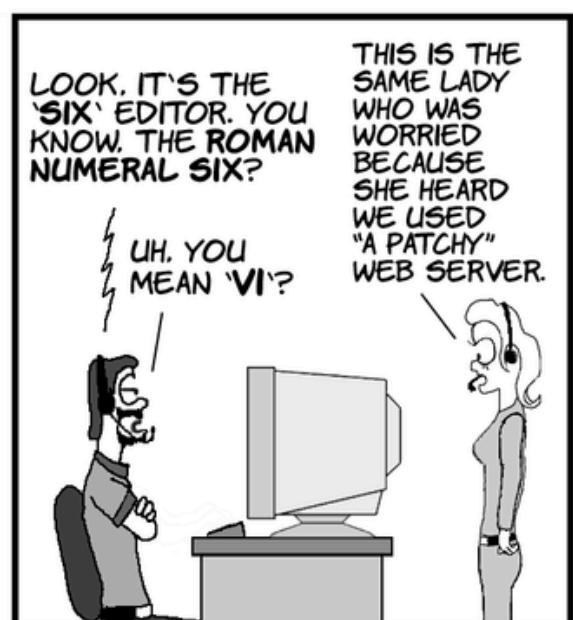
Early text editors

USER FRIENDLY by Illiad



Copyright (c) 2000 Illiad <http://www.userfriendly.org/>

I'M SORRY M'AM, BUT
I DON'T KNOW WHAT
EDITOR THIS IS THAT
YOU'RE TALKING ABOUT.



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

EVOLUTION OF THE IDE

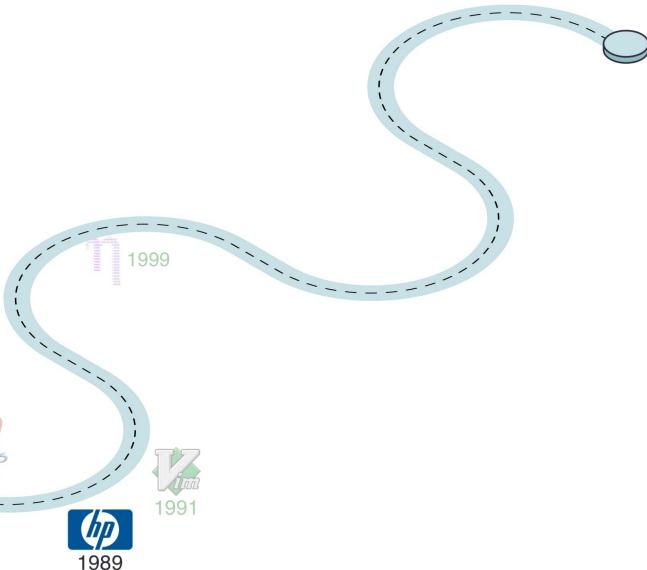
Early Text Editors



Early text editors

- Emacs, 1985
- Vim, 1991
- `nano`, 1999

EVOLUTION OF THE IDE



Early Text
Editors



1976



1985



1989



1991

First plug-in IDE

HP Softbench, 1989



First plug-in IDE

HP Softbench, 1989

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.

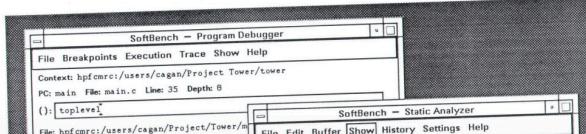
1 illustrates the HP SoftBench user interface.

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:

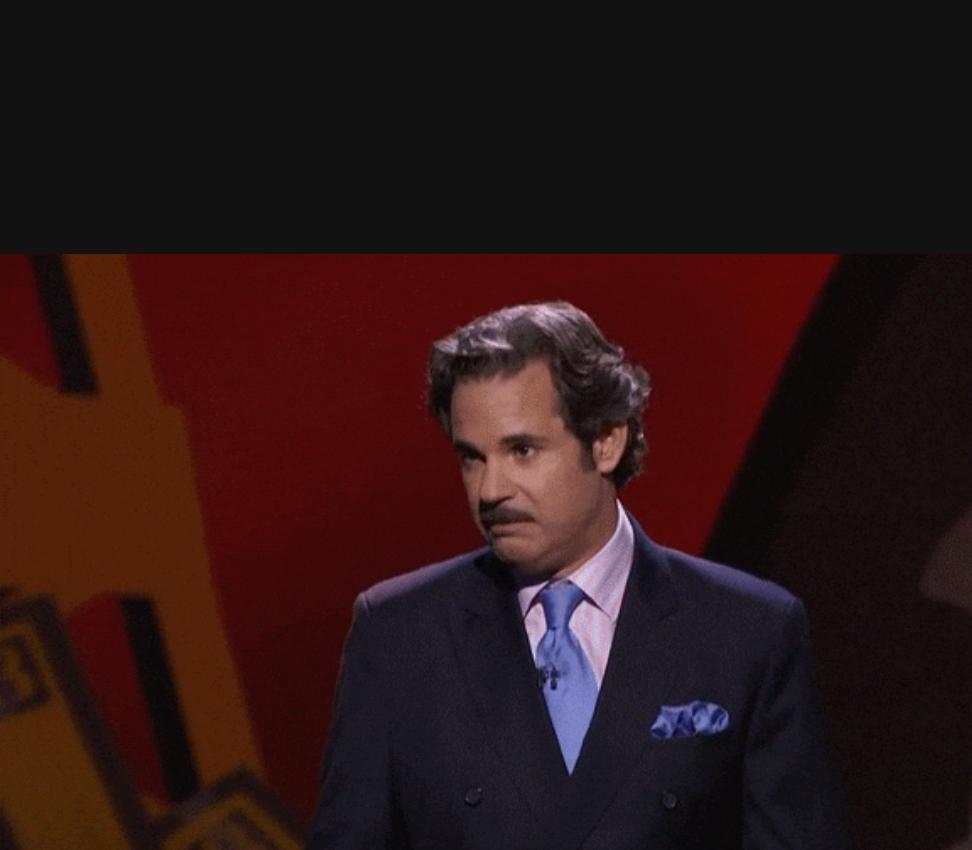
(continued on page 38)



First plug-in IDE

HP Softbench, 1989

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

Early Text Editors

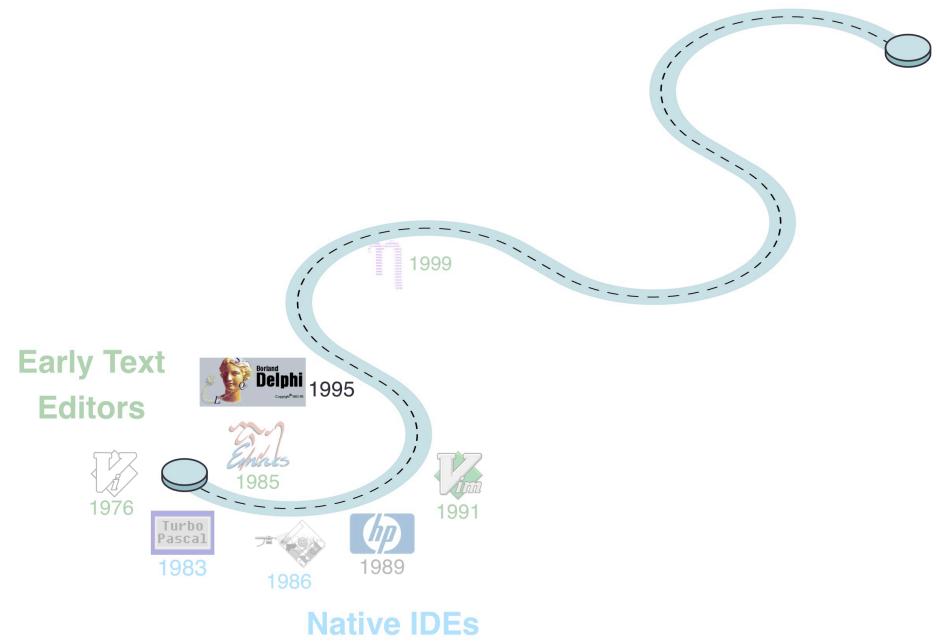


Native IDEs

Native IDEs in the 1980s

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

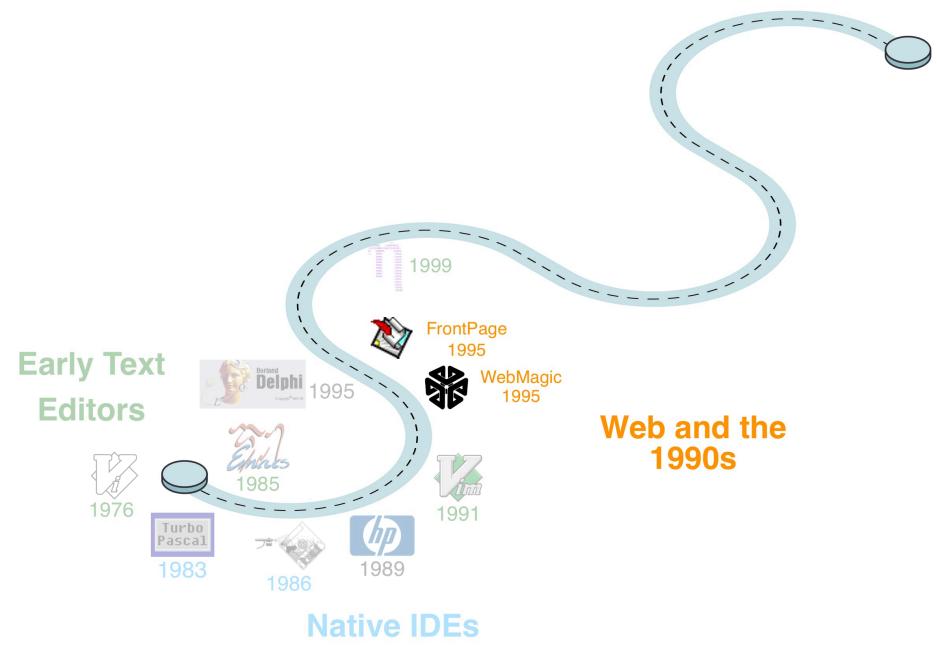
EVOLUTION OF THE IDE



Cross-platform in the 1990s

- Borland Delphi, 1995

EVOLUTION OF THE IDE



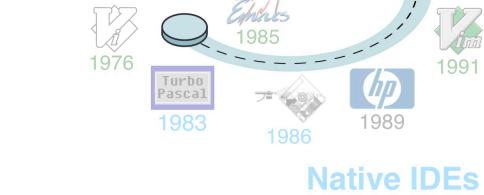
The Web and the 1990s

- Netscape Composer, 1995
- FrontPage, 1995
- SGI's WebMagic, 1995 (story)
- Microsoft FrontPage, 1995

EVOLUTION OF THE IDE

Feature & Usability Enhancements

Early Text Editors

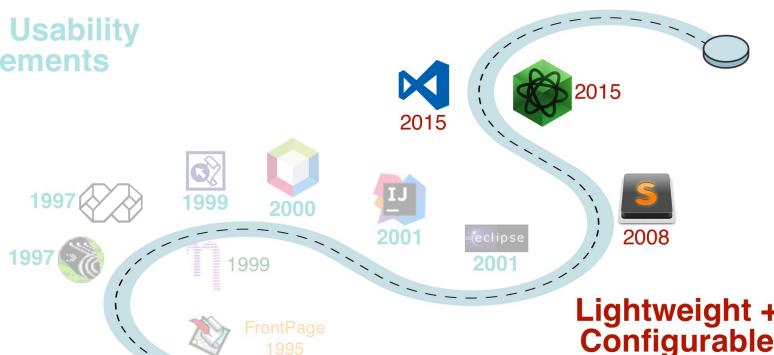


Feature & Usability Advancements (1990s - 2000s)

- Macromedia Dreamweaver, 1997
- MS FrontPage 2000, 1999
- NetBeans, 2000
- IntelliJ IDEA, 2001
- Eclipse, 2001
- MS Visual Studio, 2001

EVOLUTION OF THE IDE

Feature & Usability Enhancements



Early Text Editors

Native IDEs

Lightweight and configurable (2010s - Now)

- Sublime Text, 2008
- Atom, 2014
- VS Code, 2015

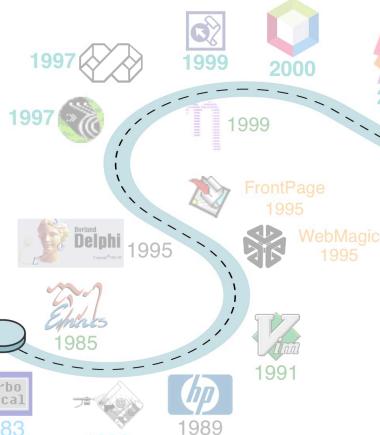
EVOLUTION OF THE IDE

Feature & Usability Enhancements

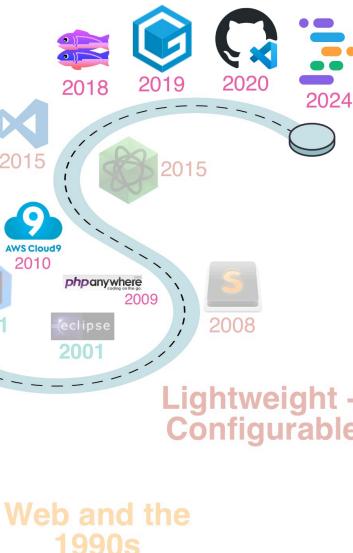
Early Text Editors



Native IDEs



Cloud-based



Cloud-based options

- PHPAnywhere (CodeAnywhere), 2009
- Cloud9, 2010
- Glitch, 2018
- GitPod, 2019
- GitHub Codespaces, 2020
- Google's Project IDX, 2024

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 FTP)
- Multi-language support
- Framework integration
- Pair programming

EVOLUTION OF THE IDE

Feature & Usability Enhancements

Early Text Editors



1976



1983



1995



1985



1991



1986



1997



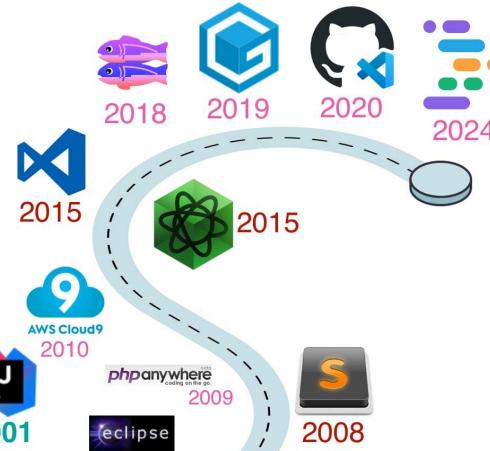
1999



2000

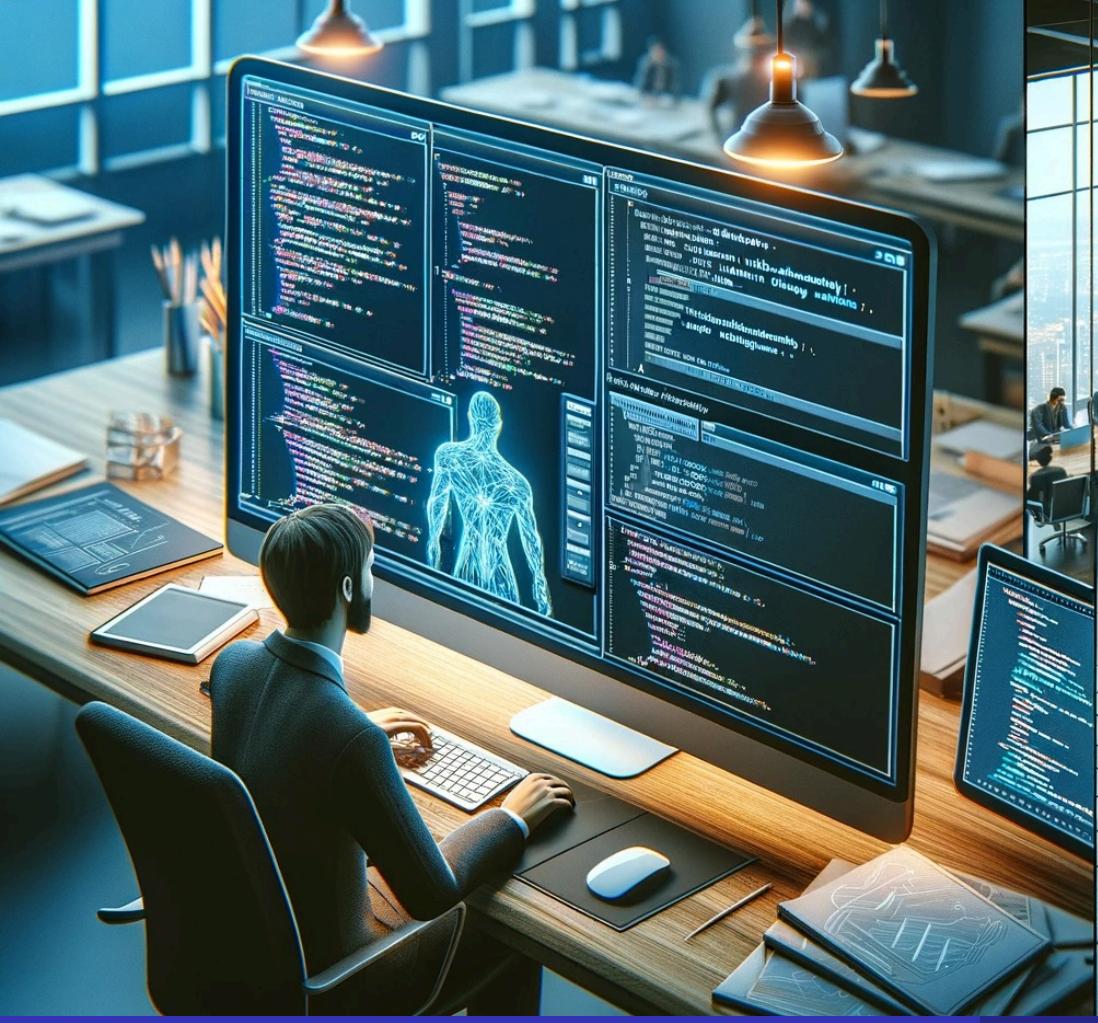
FrontPage
1995WebMagic
1995

Cloud-based



Lightweight + Configurable

Web and the 1990s





CI/CD Standardization

- consistency to development pipelines
- reduces friction
- enhances collaboration

Implementing CI/CD Standardization

Assessment and Analysis

- Thoroughly assess your current CI/CD pipelines
- Identify pain points and bottlenecks
- Analyze specific requirements and constraints

Implementing CI/CD Standardization

Define Standardization Goals

- Define goals and objectives, align with strategy and objectives
- Determine success, like reduced deployment times / error rates

Implementing CI/CD Standardization

Select Tools and Practices

- Choose tools & practices aligned with organization needs, goals
- Establish standard templates and configurations for pipelines
- Enforce coding standards for consistency and readability

Implementing CI/CD Standardization

Utilize Version Control

- Store pipeline configs as code in version control systems
- Implement branching and pull request strategies

Implementing CI/CD Standardization

Automated Testing and Validation

- Integrate automated testing and validation into templates
- Implement code reviews and peer validation early in dev process

Implementing CI/CD Standardization

Documentation and Training

- Create comprehensive docs for processes, configs, best practices
- Provide training to ensure understanding and effective use



Optimizing CI/CD Standardization

Optimizing CI/CD Standardization

Continuous Monitoring & Improvement

- Detect pipeline issues and bottlenecks in real-time
- Establish culture of regular reviews and updating pipelines



Optimizing CI/CD Standardization

Governance & Compliance

- Implement governance policies to enforce pipeline standards
- Validate compliance with industry regulations / internal standards
- Regularly audit and assess adherence to standardized practices



Optimizing CI/CD Standardization

Scaling & Adaptation

- Ensure standardized templates can scale and adapt
- Maintain flexibility to accommodate unique project requirements



Optimizing CI/CD Standardization

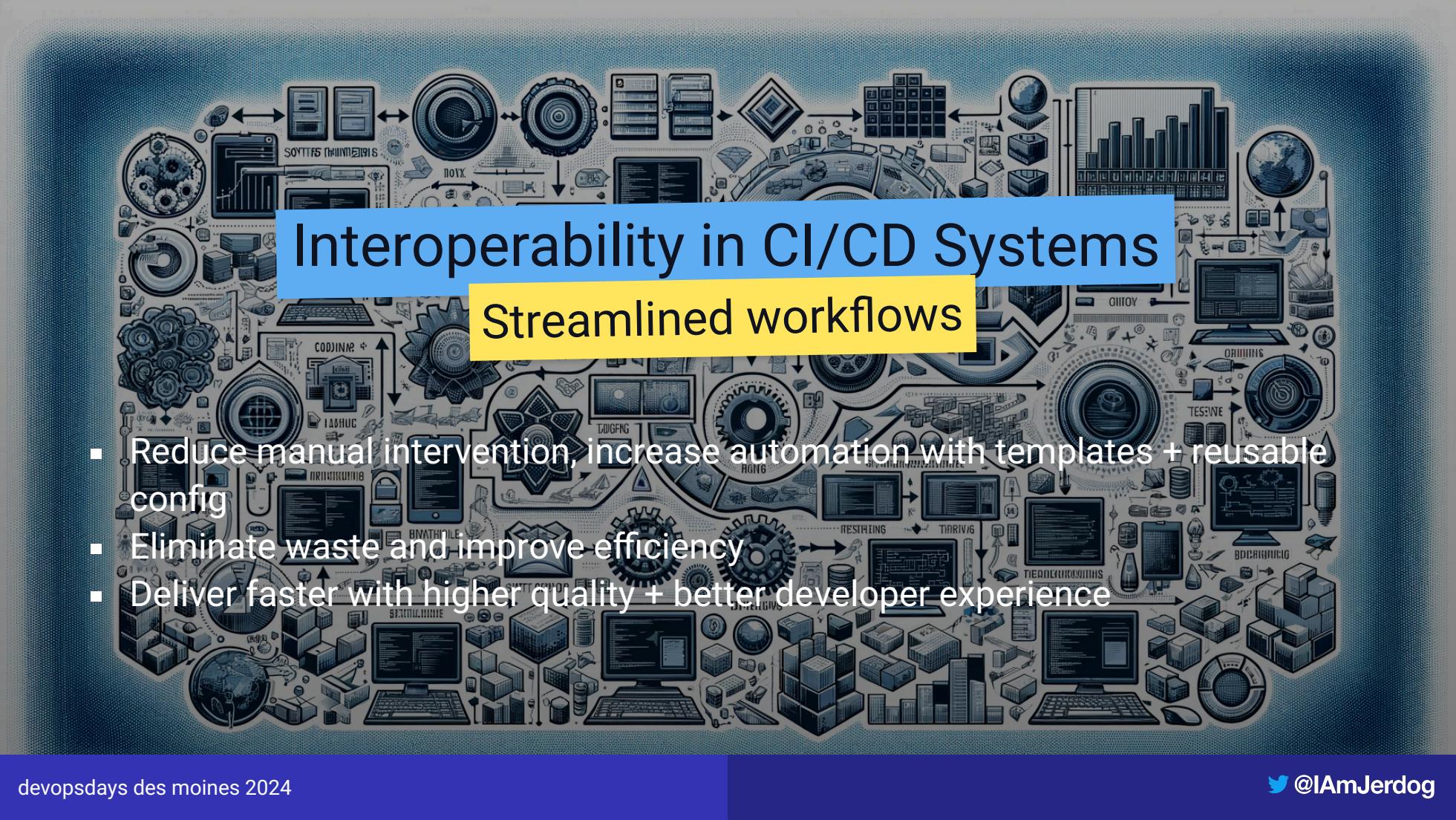
Feedback Loop & Collaboration

- Foster collaborative environments where feedback & contributions encouraged
- Continuously communicate benefits of standardized pipelines & celebrate successes



CI/CD Interoperability

- seamless integration across diverse toolsets
- fosters flexibility in development environments



Interoperability in CI/CD Systems

Streamlined workflows

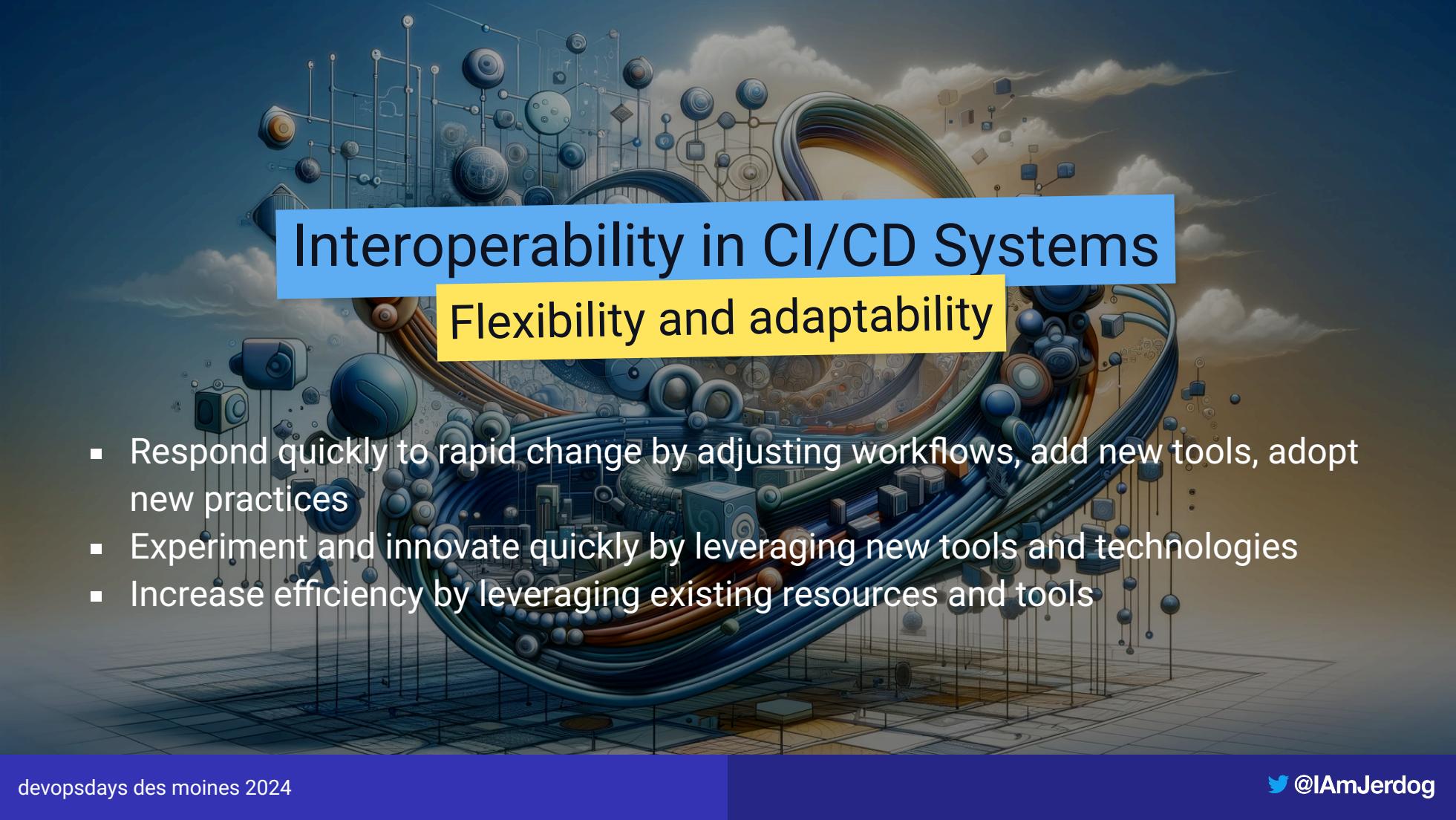
- Reduce manual intervention, increase automation with templates + reusable config
- Eliminate waste and improve efficiency
- Deliver faster with higher quality + better developer experience



Interoperability in CI/CD Systems

Cross-functional collaboration

- Have shared goals and break down silos
- Improve resource utilization and efficiency
- Leverage strengths and expertise of each team

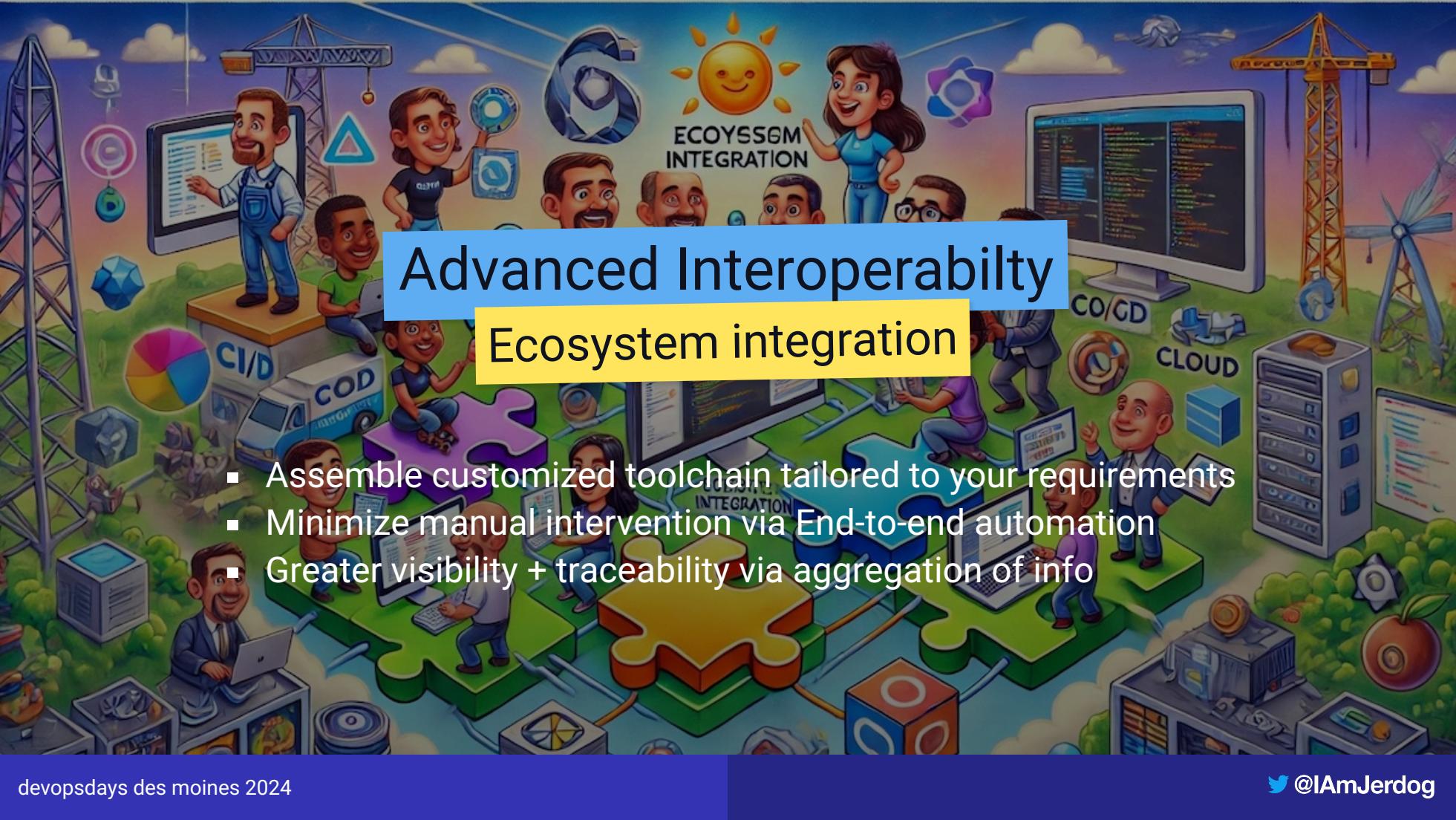


Interoperability in CI/CD Systems

Flexibility and adaptability

- Respond quickly to rapid change by adjusting workflows, add new tools, adopt new practices
- Experiment and innovate quickly by leveraging new tools and technologies
- Increase efficiency by leveraging existing resources and tools





Advanced Interoperability

Ecosystem integration

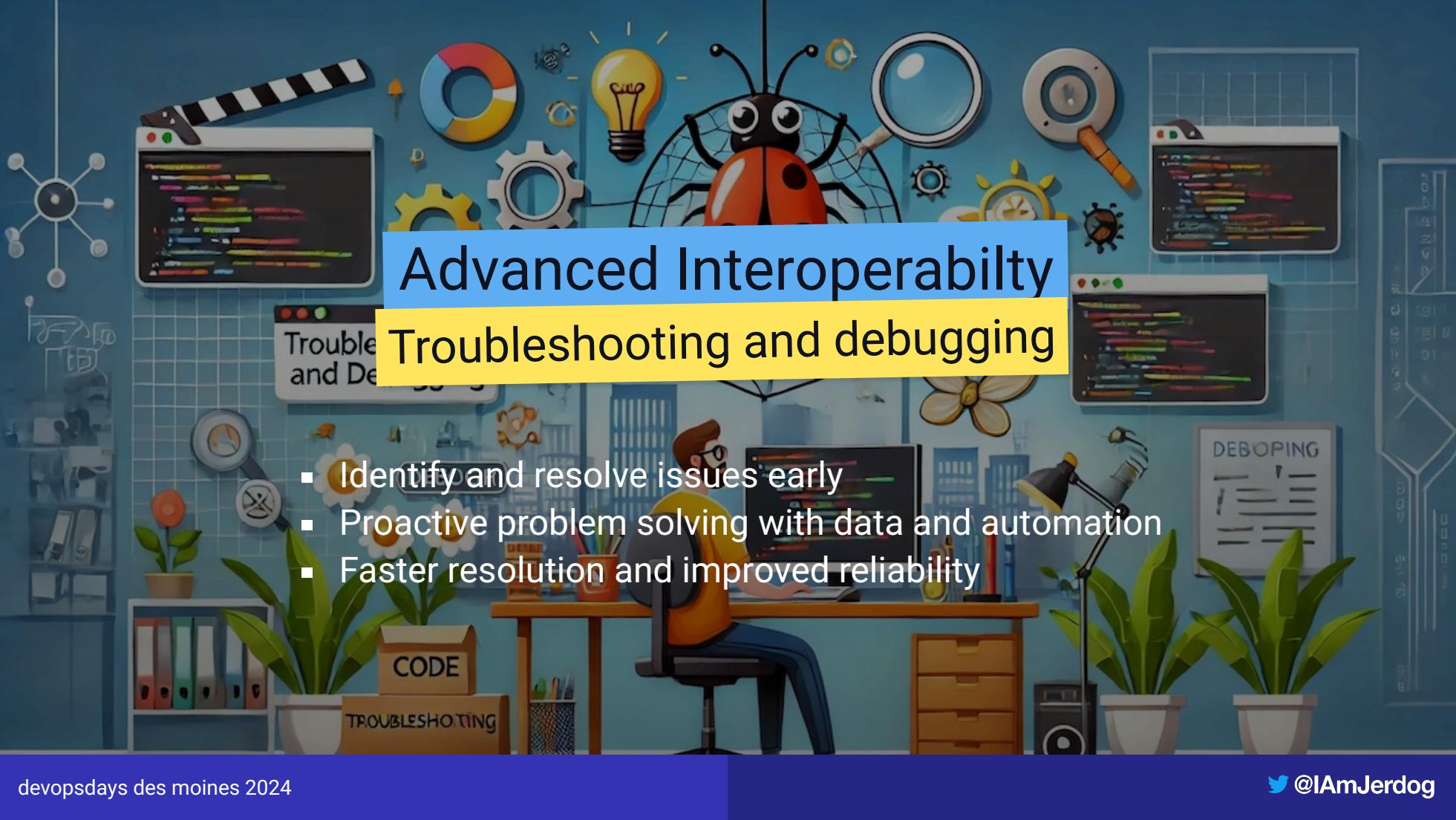
- Assemble customized toolchain tailored to your requirements
- Minimize manual intervention via End-to-end automation
- Greater visibility + traceability via aggregation of info

Advanced Interoperability

Ecosystem integration

Role of community and Open Source

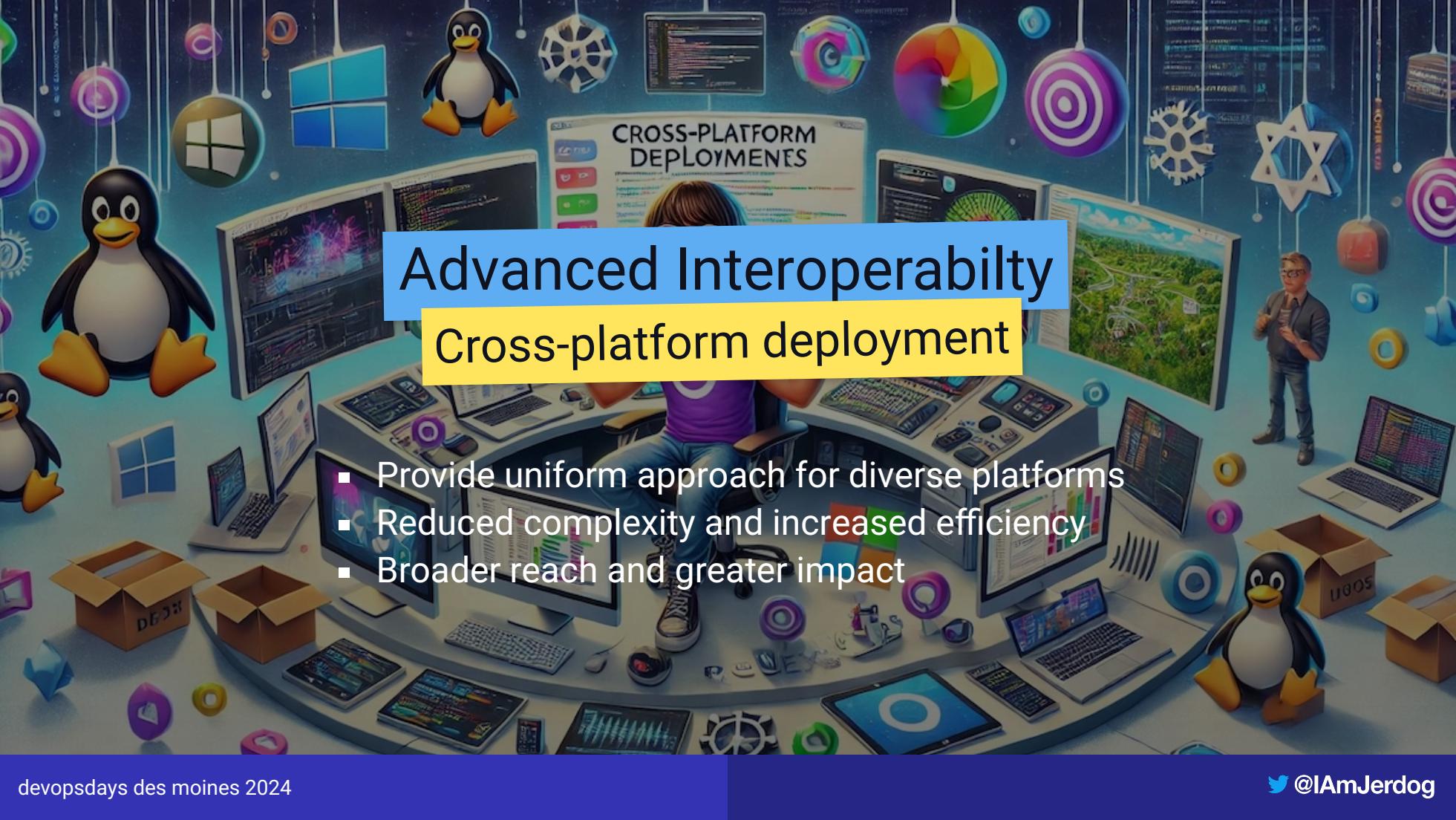
- Address potential interoperability challenges (case studies, submitting fixes)
 - Create external community around your tools, gathering feedback directly



Advanced Interoperability

Troubleshooting and debugging

- Identify and resolve issues early
- Proactive problem solving with data and automation
- Faster resolution and improved reliability



Advanced Interoperability

Cross-platform deployment

- Provide uniform approach for diverse platforms
- Reduced complexity and increased efficiency
- Broader reach and greater impact



CI/CD Interoperability Challenges...

...and Remedies

Challenges	Remedy
Diverse Toolsets & Ecosystems	<i>Prioritize critical dependencies, objectives</i>
Data formats and schema differences	<i>Implement unified data formats</i>
Authentication and Authorization	<i>Standardize methods, integrate governance</i>
Versioning and compatibility testing	<i>Clear versioning policies, regular compatibility testing</i>
Lack of documentation	<i>Prioritize efforts + allocate resources, implement standards + process</i>

DevEx reflects an organization's values



Jeremy (#DevRel & #DevEx)

@IAmJerdog · [Follow](#)



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.

10:57 AM · Jan 25, 2024



26



[Reply](#)



[Copy link](#)

[Read more on X](#)

DevEx is...

...ruthlessly eliminating the barriers (and blockers) that keep your developers (and practitioners) from being successful

-Me

Thank You.



/in/jeremyemeiss



@IAmJerdog



@jerdog



@jerdog@hachyderm.io

END