

How to have your DevOps cake and eat it, too!

Bill Shubin

Senior Sales Engineer



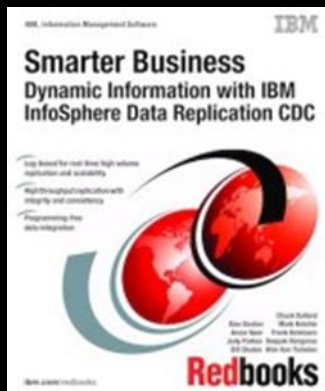
Visit us in
Booth
#515

Meet Your Speaker



Bill Shubin

Sales Engineer



Co-authored IBM
Redbooks
publication on
InfoSphere Data
Replication CDC

Seasoned Sales Engineer with 20+ years in technical pre-sales. Specializes in data integration, DevOps, and modernization across enterprise systems.

Expertise

- DevOps (IBM i, LUW), CI/CD
- Data Integration & Replication (ETL, CDC)

Career Highlights

- Rocket Software: DevOps advisor, CI/CD advocate
- Couchbase: NoSQL evangelist for modern web/mobile
- IBM: Technical sales for InfoSphere Information Server

"Helping enterprises modernize legacy platforms—without sacrificing what already works."

✉ wshubin@rocketsoftware.com 📞 626.625.3572

🔗 linkedin.com/in/billshubin



Agenda

In today's session, we'll cover:

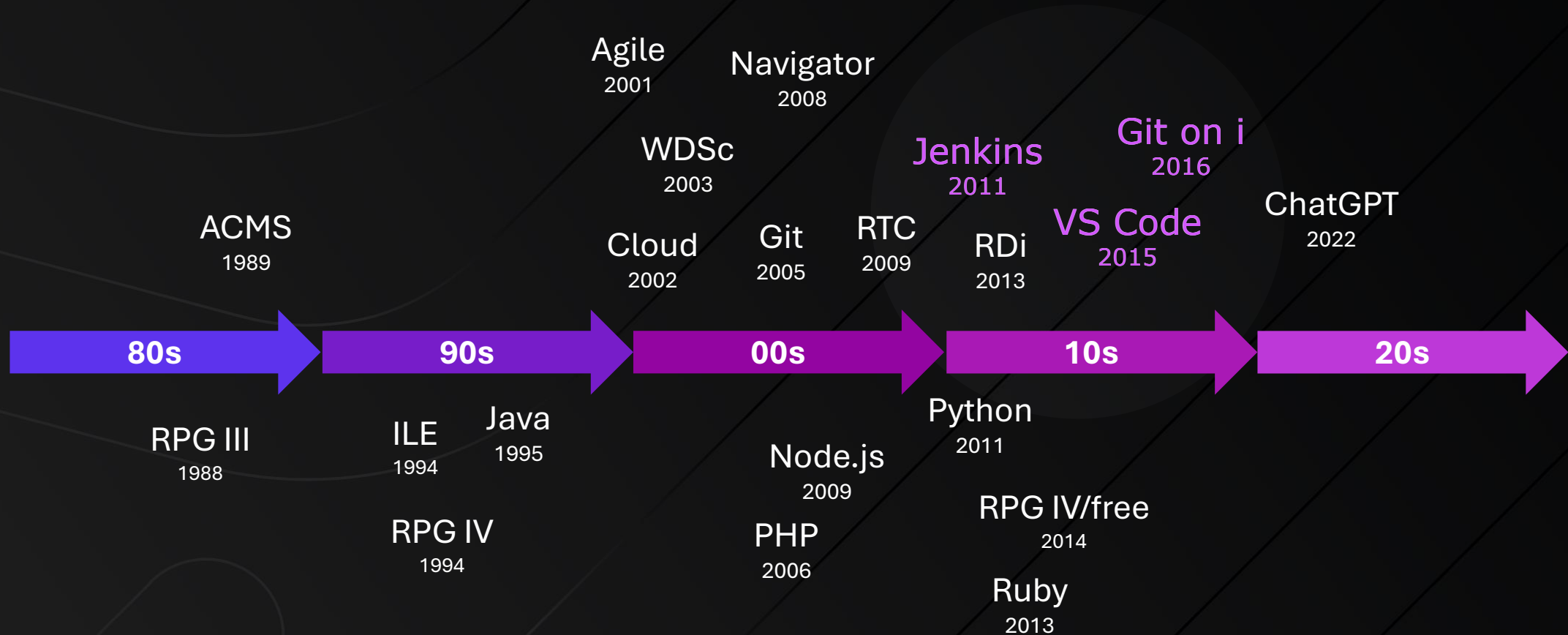
1. Best practices for populating Git with IBM i source code
2. Testing new 3rd-Party tools without disrupting what works
3. Empowering developers using both 5250 and modern tools
4. Real-world examples of what works — and what doesn't
5. Recommendations & Takeaways
6. Q/A

But First... *Things To Consider Before We Start*

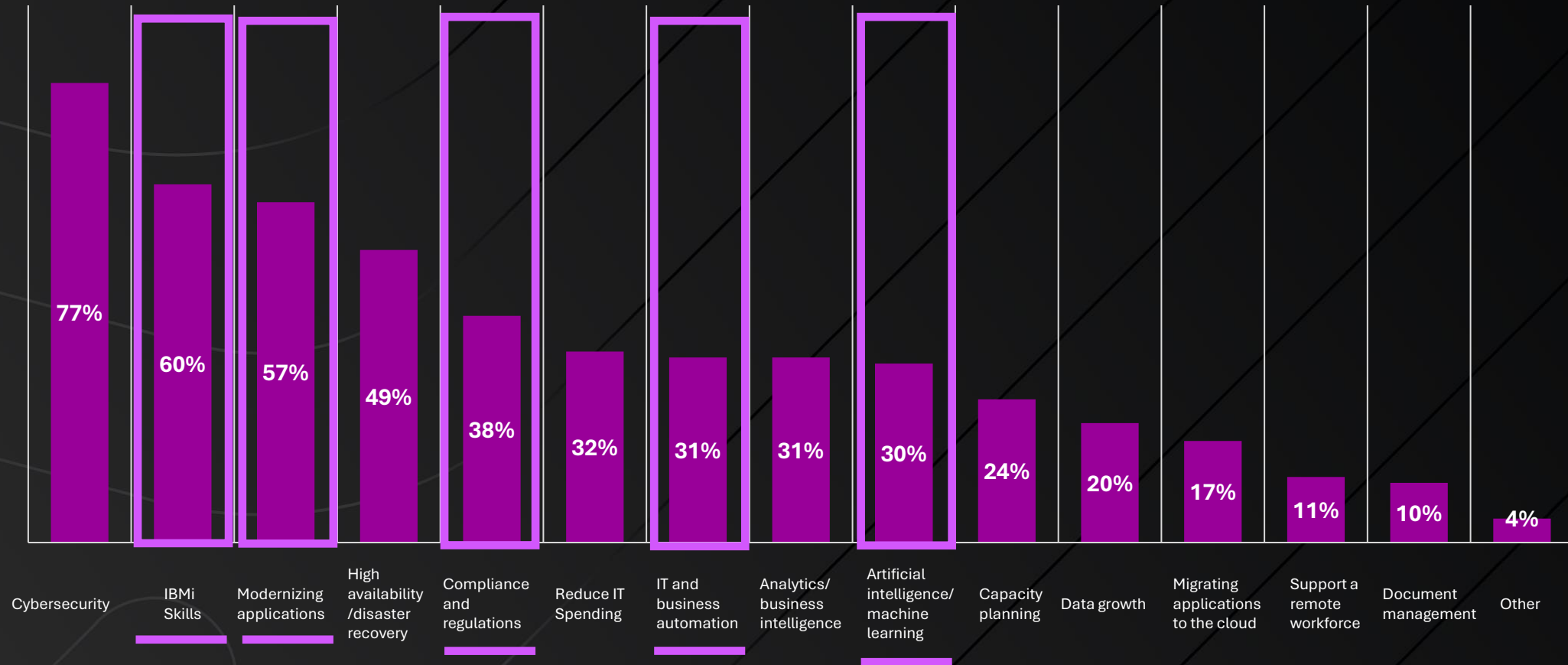
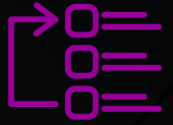
Are you using a
DevOps product?

- Yes
- No

Language and Tools Evolution



Top priorities and concerns*



Agenda

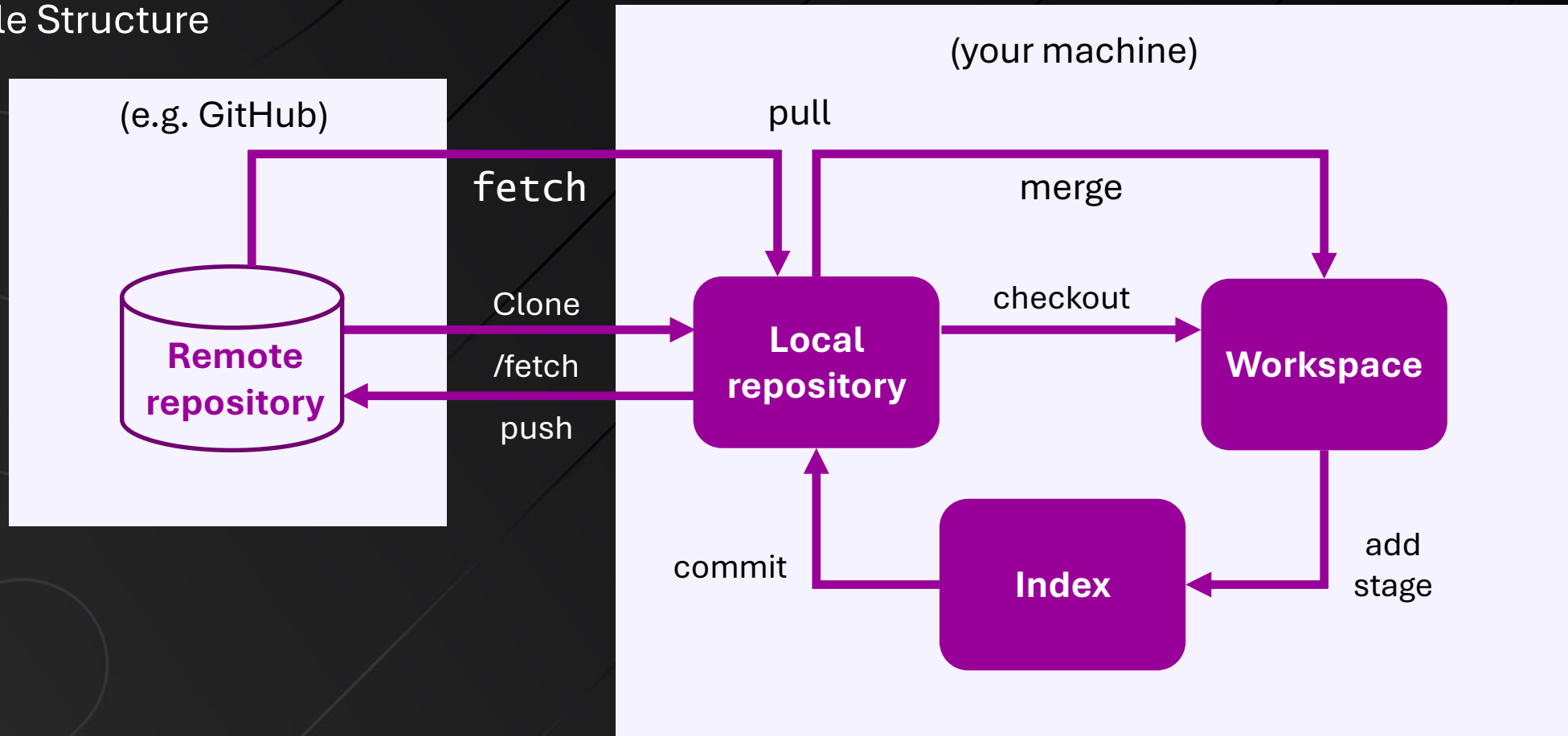
In today's session, we'll cover:

1. Best practices for populating Git with IBM i source code
2. Testing new 3rd-Party tools without disrupting what works
3. Empowering developers using both 5250 and modern tools
4. Real-world examples of what works — and what doesn't
5. Recommendations & Takeaways
6. Q/A

Git Concepts



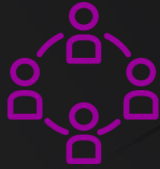
Sample Structure



Best Practices for Populating Git with IBM i Source Code

- Organize source code for easy version control
- Separate logical layers: Source, objects, deployment
- Structure repositories
 - Source physical files, stream files, hybrid by module or function
- Automate synchronization between IBM i and Git

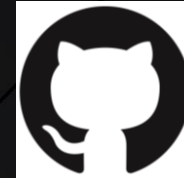
Git Tools for the Job



Git is distributed



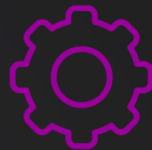
Get a Git server



Any editor + CLI will do



Get an editor with Git integrated



Utility tools



Tortoise Git



Comparison Tool

Adopting IBM i and Git

Managing Git with the IBM i

- Develop on the IBM i -> Sync with Git
- Develop on Git -> Sync with the IBM i
- Develop on Git and IBM i concurrently -> Bi-directional Sync

Experiment and innovate

Move at a pace that makes sense to you!

Agenda

In today's session, we'll cover:

1. Best practices for populating Git with IBM i source code
2. Testing new 3rd-party tools without disrupting what works
3. Empowering developers using both 5250 and modern tools
4. Real-world examples of what works — and what doesn't
5. Recommendations & Takeaways
6. Q/A

Testing New 3rd-Party Tools Without Disrupting What Works



Static Code Analysis



Test Automation



Today, a new generation of programmers who use VS Code can create and deploy code for the IBM i platform without knowing anything at all about the IBM i platform itself.

*Alex Woodie, IT Jungle,
Jan 25 2023*

- Easy to install
- Easy to use
- Microsoft – but free and open source
- Cross-platform
- Every company already has it

Popular project management and issue-tracking tool

- Track IBM i Work: Manage RPG/COBOL tasks, bugs, and requests.
- DevOps Integration: Connect with DevOps tools for automated workflows.
- Audit & Compliance: Link issues to code changes for traceability and reporting.

Modern Open-Source Integrations



DevOps developers can continue to use Git for version control – but still adhere to DevOps processes and maintain compliance.



Plugins for Continuous Integration and Continuous Delivery purposes to build and test your software projects.



Plugin streamlines IBM i development with integrated editing, version control, and deployment in one modern IDE.



Integration with DevOps tools enabling automation from ticket creation to code promotion.

Testing 3rd Party Tools without Disruption

- Use a sandbox or a non-production environment
- Clone the process and run in parallel
- Use feature flags or isolate test workflows
- Communicate clearly with teams during testing

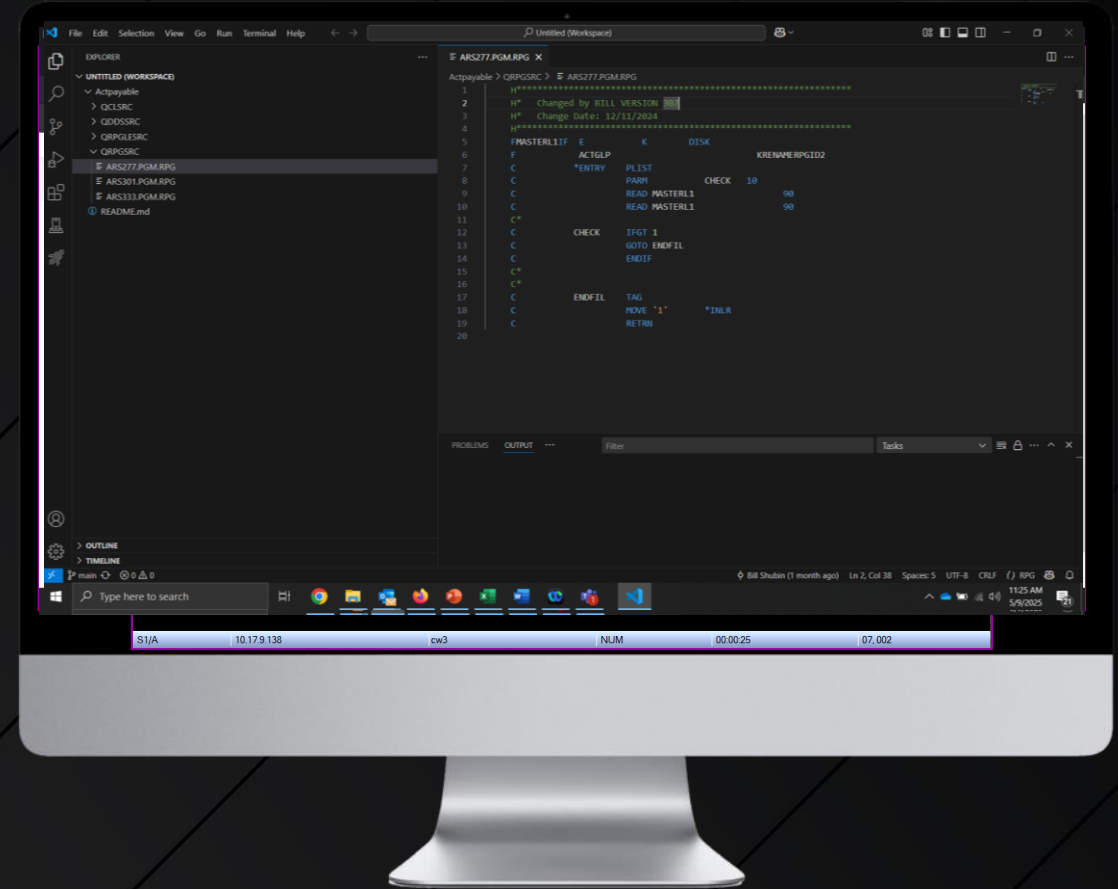
Agenda

In today's session, we'll cover:

1. Best practices for populating Git with IBM i source code
2. Testing new 3rd-Party tools without disrupting what works
3. Empowering developers using both 5250 and modern tools
4. Real-world examples of what works — and what doesn't
5. Recommendations & Takeaways
6. Next steps

Popular IBM i Development IDEs

- Manage traditional 5250 development based on source members
- Support for common IDE's & tools including:
 - RDi
 - VS Code
- Security, Build, Deploy, & Compliance functions built-in



Empowering developers using both 5250 and modern tools

- Respect the power of SEU and PDM — but extend their capabilities
- Encourage hybrid workflows using RDi, VS Code, or web UIs
- Train developers in either classic or modern tools, or both
- Celebrate progress without forcing change

Agenda

In today's session, we'll cover:

1. Best practices for populating Git with IBM i source code
2. Testing new 3rd-Party tools without disrupting what works
3. Empowering developers using both 5250 and modern tools
4. Real-world examples of what works — and what doesn't
5. Recommendations & Takeaways
6. Q/A

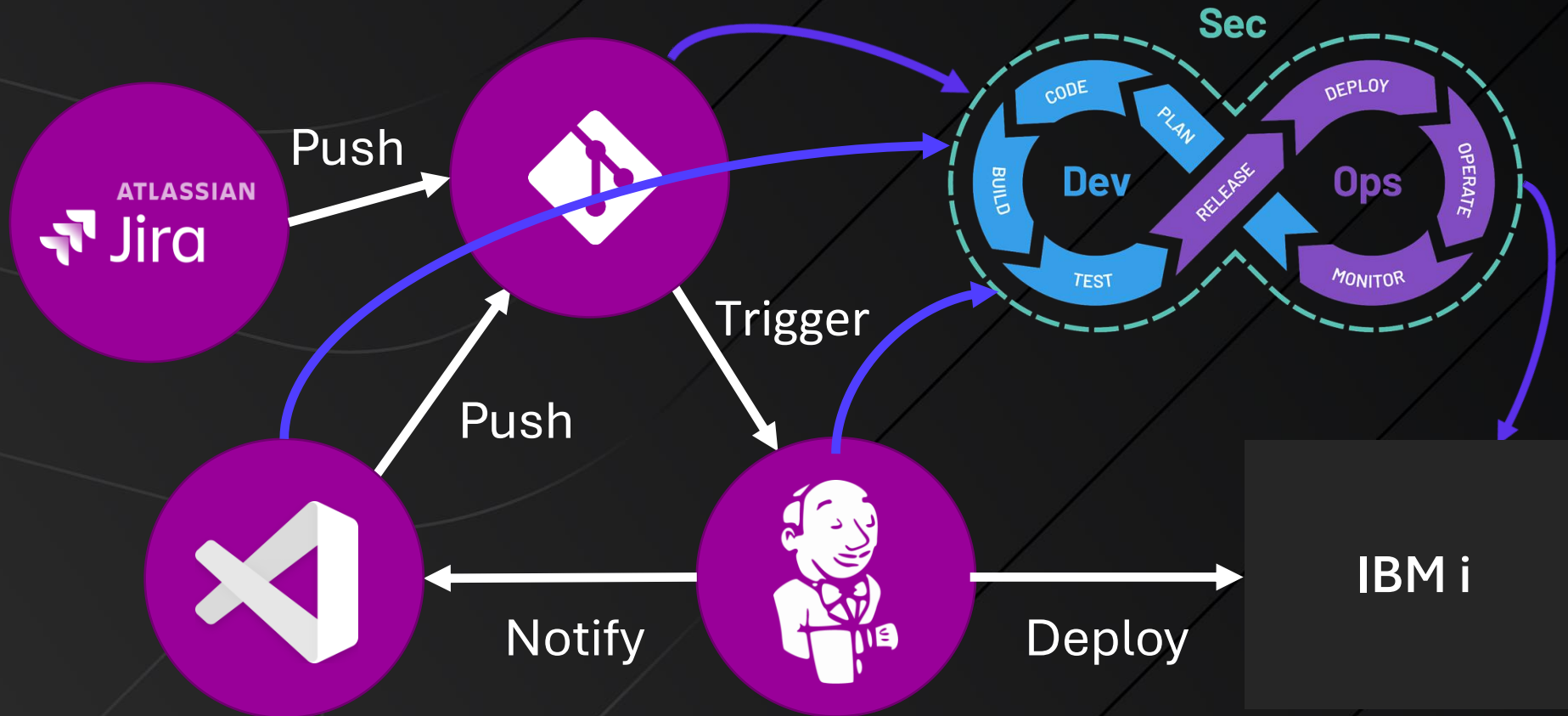
Real-World Examples of What Works

- Always validate your production environment (preferably in an automated fashion)
- Use separate partitions for Development, QA, and Production (when possible).
- You must secure your testing, QA, and production code
- Git + IBM i Integration
- Automated Testing (No CI/CD makes sense without it)
- IBM i DevOps Access Control
- Source Locking with DevOps

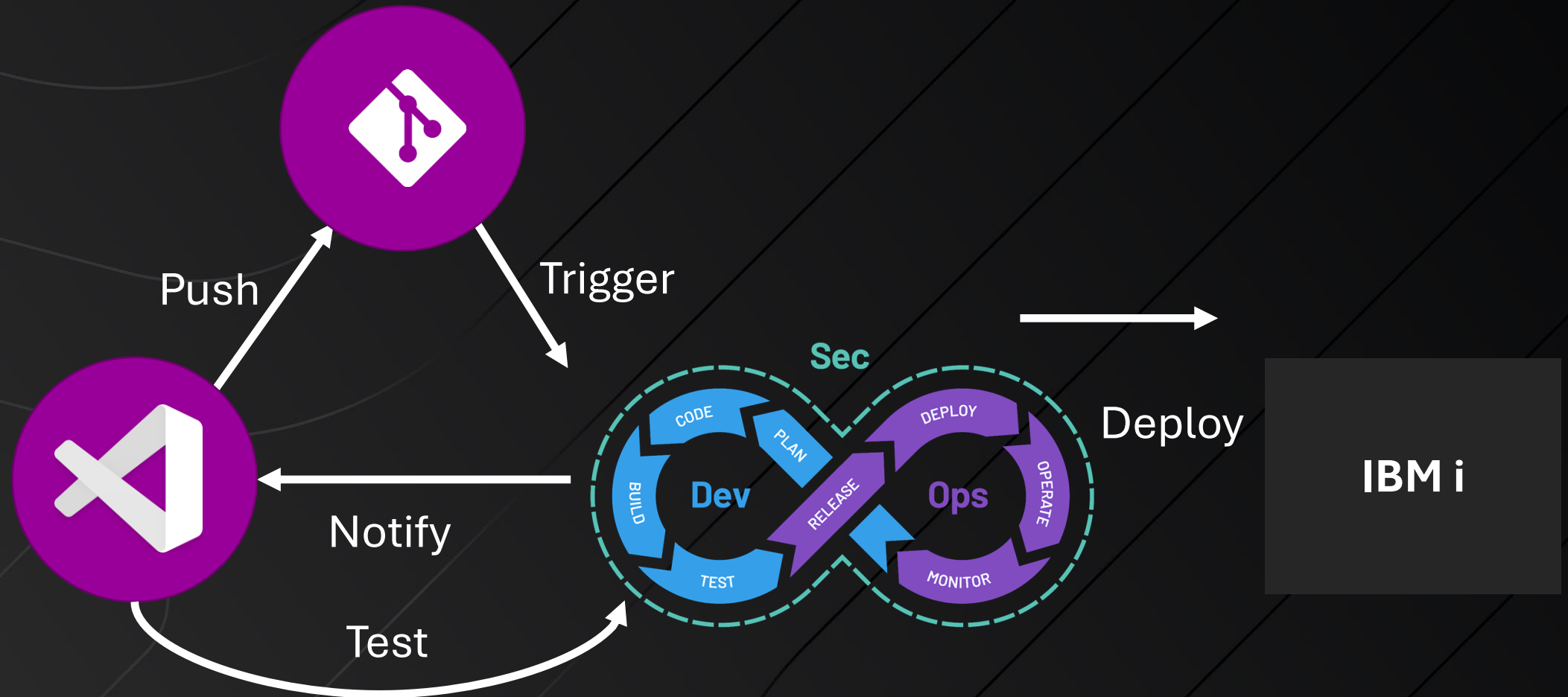
Real-World Examples of What Doesn't Work

- Missing fail-safe rollback process
- No archive of production objects or source
- Allowing promotions without source verification
- Lack of role-based access control and permissions
- Widespread use of *ALLOBJ or *SECADM authorities or emergency overrides

A DevOps Solution Is Recommended...



...and Jenkins or Jira May Not Be Needed



Agenda

In today's session, we'll cover:

1. Best practices for populating Git with IBM i source code
2. Testing new 3rd-Party tools without disrupting what works
3. Empowering developers using both 5250 and modern tools
4. Real-world examples of what works — and what doesn't
5. Recommendations & Takeaways
6. Q/A

Recommendations and Takeaways

- Don't do it all at once – proceed at your own pace
- Agile is easier when using Git
- You don't need Jenkins if you are only developing on the IBM i
- Start using modern IDE like VS Code
- IBM i development plays well in a modern IT environment

Next Steps

- Contact your DevOps vendor
- Download Visual Studio code
- Set up a Git server
- and
- Get started!

Agenda

In today's session, we'll cover:

1. Best practices for populating Git with IBM i source code
2. Testing new 3rd-Party tools without disrupting what works
3. Empowering developers using both 5250 and modern tools
4. Real-world examples of what works — and what doesn't
5. Recommendations & Takeaways
6. Q/A



How to have your cake and eat it too in the DevOps world

Please take the last minute of this session to complete the evaluation. A direct link to the evaluation can be found using the QR code to the right.



Thank You!

info@rocketsoftware.com

www.rocketsoftware.com



Visit us in
Booth
#515

Rocket presentation disclaimer

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON ROCKET'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY ROCKET WITHOUT NOTICE. ROCKET SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM ROCKET (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF ROCKET PRODUCTS OR SOFTWARE.

Rocket's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at Rocket's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Rocket and Rocket logo are trademarks of Rocket Software Inc. ("Rocket Software"), registered in many jurisdictions worldwide. Other product and service names might be trademarks of Rocket Software or other companies. A current list of other Rocket Software-owned trademarks and guidelines of the permitted use is available at <https://www.rocketsoftware.com/company/legal/trademarks>

Other company, product, or service names may be trademarks or service marks of others.

© 2025 Rocket Software Inc. or its affiliates. All rights reserved.