



Speed in CI/CD: Get Faster to Get Better

Dan Zentgraf, Director, Solutions Architecture - Liquibase

Today's Presenter



Dan Zentgraf
Director, Solutions Architecture

What This Talk is About

- Speed is Good
- Acceleration is Better
- Pushing to Get Faster Makes You Better



Speed is Good



Fast Things Tend to Be More Awesome



We Like Speed

- We like taking less time to get places
- We like getting things sooner
- We hate waiting
- We get frustrated when we are slowed by blocked work
- We have tons of cultural sayings on the benefits of speed

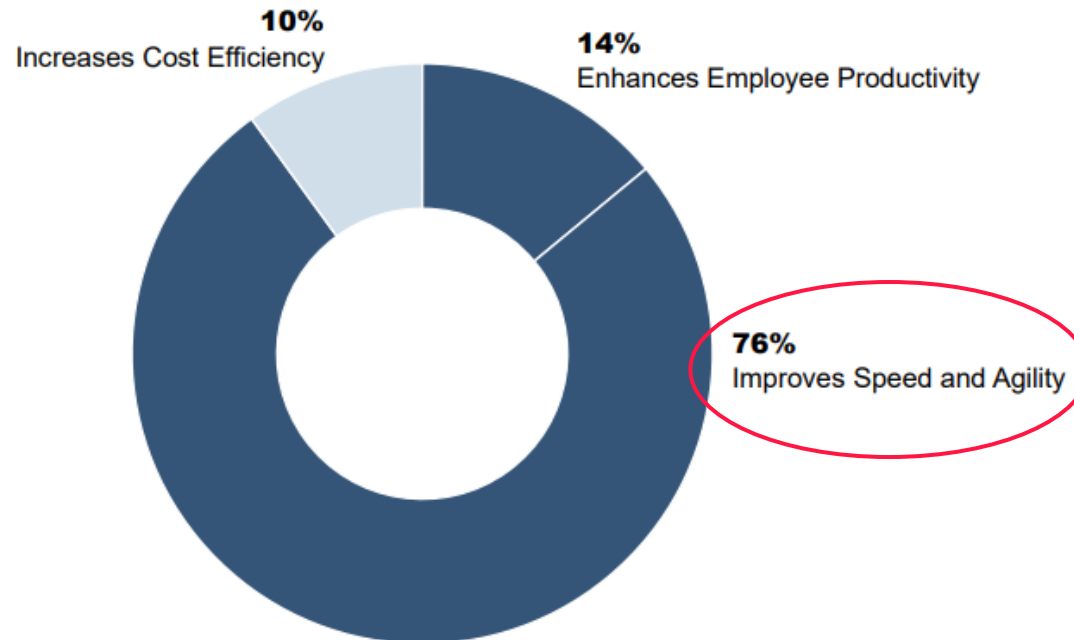


Business Uses Technology to Achieve Speed

Automation for Speed and Productivity

Automation investments driven by desire to increase speed and enhance productivity

Top Value Drivers for Automation Investments in 2020



n = 49

Source: 2020 Gartner Emerging Technology Roadmap for Midsize Enterprises



- POLL -

What do you think is the greatest benefit of CI/CD?

- Increased speed of delivery
- Shortened development cycle
- Increased release frequency
- Reduced complexity of dev lifecycle
- Reduction in bugs post deployment

Key Benefits of CI/CD

Technical Benefits of CI/CD	
Increased speed of delivery	74%
Shortened development cycle	68%
Increased release frequency	59%
Reduced complexity of dev lifecycle	50%
Reduction in bugs post deployment	40%
Reduced time to complete QA feedback loop	37%



Acceleration is Better

The *PURSUIT* of more speed is better than just keeping speed you already have

Race to Commercial Jet Aviation

- 1947 – Boeing 377

- One of the last piston-engine propeller airliners
- ~60-100 passengers
- Cruising Speed – 262 knots (201 mph)
- Range ~3,600 nautical miles

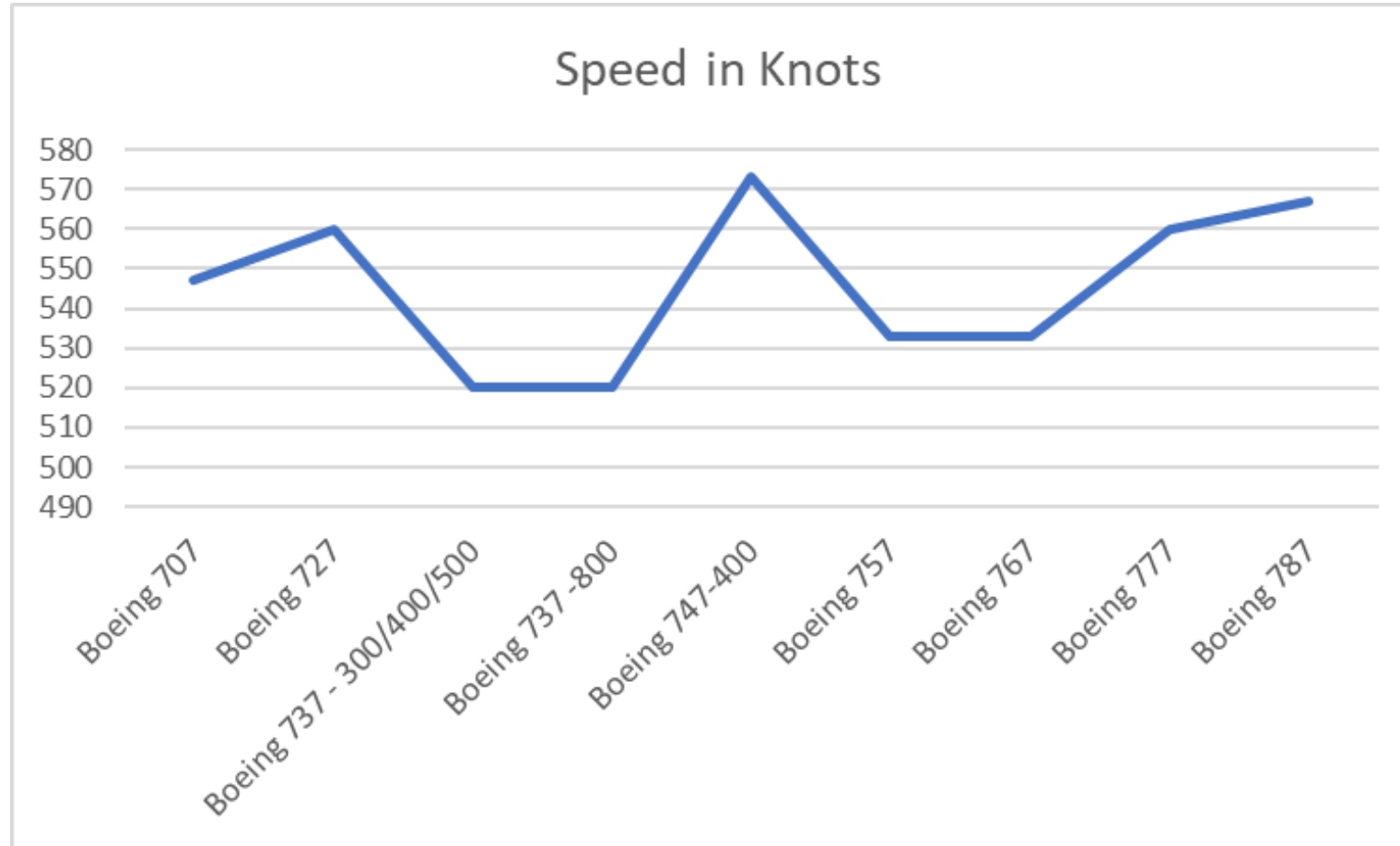


- 1957 - Boeing 707

- ~189 passengers depending on configuration and variant
- Cruising speed of 525 knots (~604 mph)
- Range ~3,200 - 4,800 nautical miles



Commercial Jet Aviation 60+ Years Later



You Can Always Find a Way to Pursue Speed

- **If planes are what they are, how do you accelerate if you are an airline?**
 - Boring plane –all 737s (first versions in 1966)
- **Agility**
 - Standardized plane
 - Move crew around quickly
 - Ability to swap planes on routes
 - Simplified maintenance
 - Faster, simplified boarding
- Only consistently profitable major US carrier for 47 years running
 - (as of end of 2019)





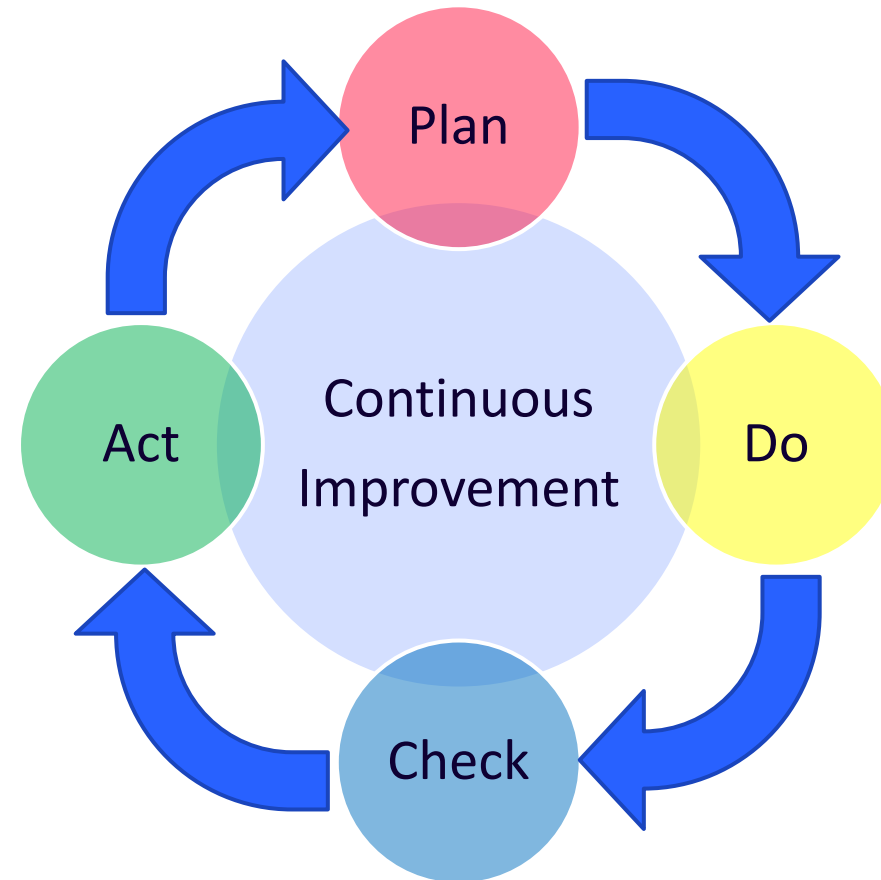
Pushing to Get Faster Makes You Better

Speed Provides a Vision and Goal



The Other CI – Continuous Improvement

- Intrinsic to Agile / DevOps
- Achieve better results without backsliding elsewhere
- Improved Throughput
- Improved Quality

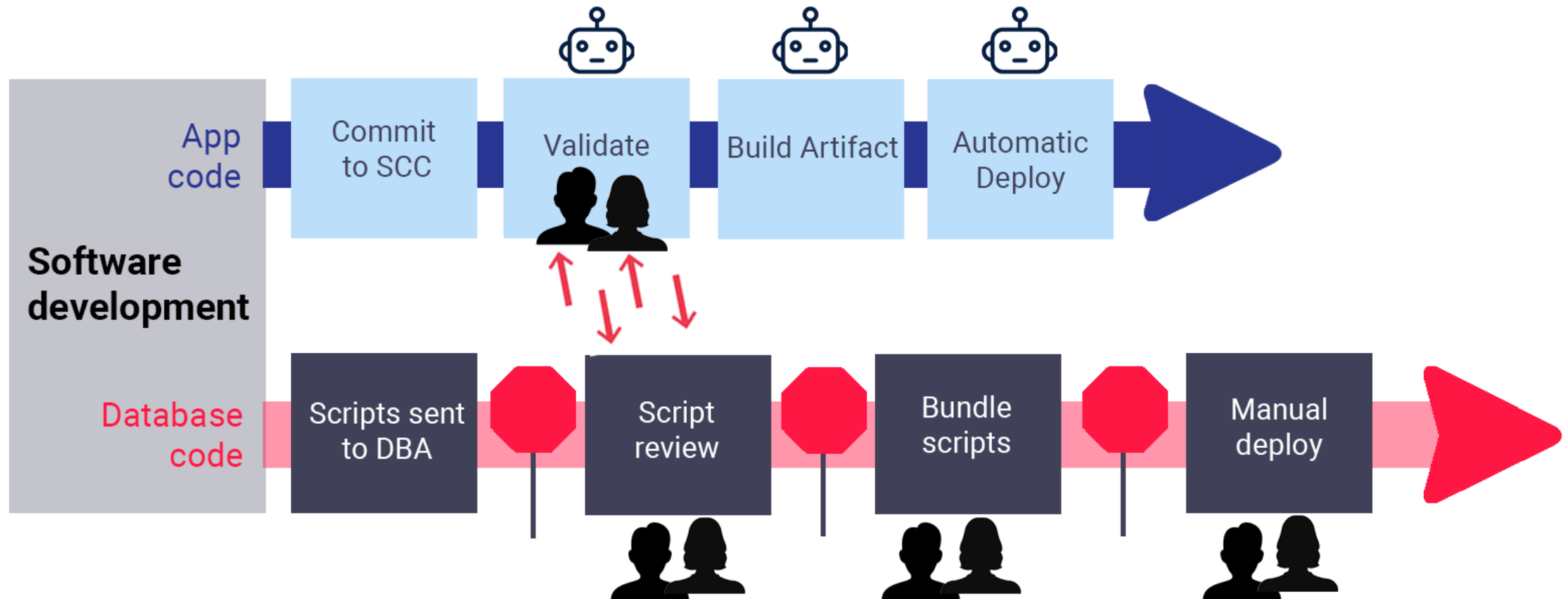


You Must Actually Fix Things

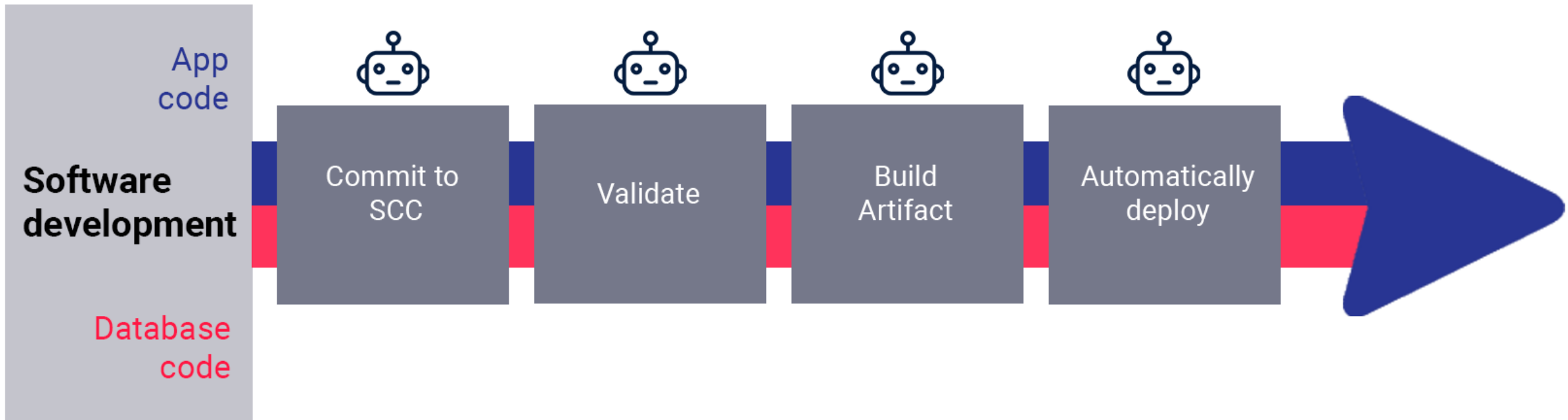
- Simplify - Remove Complexity
- Add Lightness - Remove the Unnecessary
- Automate - Remove Inconsistency
- Monitor and Measure
- Analyze Trends Over Time
- Precise Control
- Uninterrupted Flow - Eliminate Rework



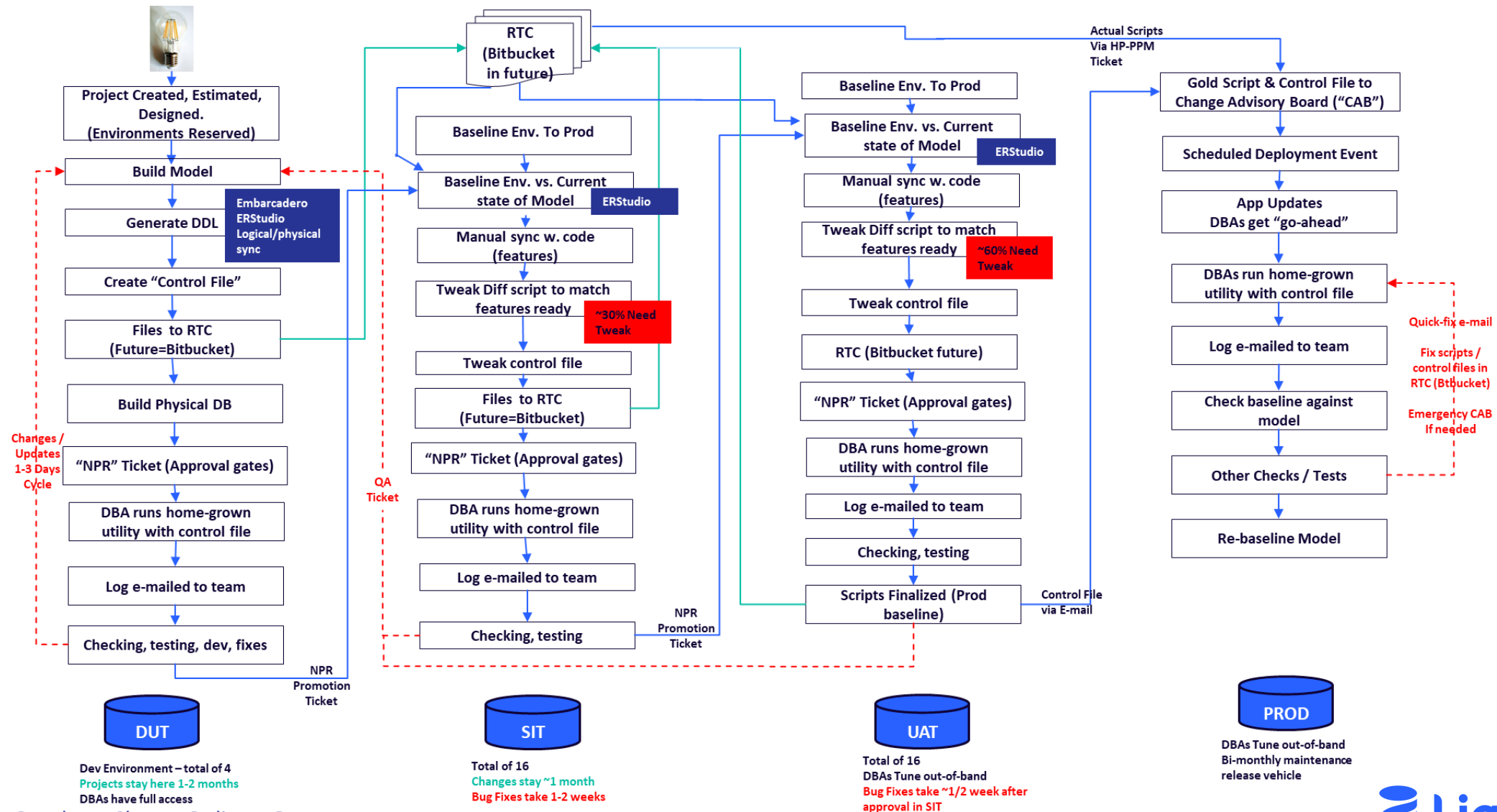
Simplify - Remove Complexity



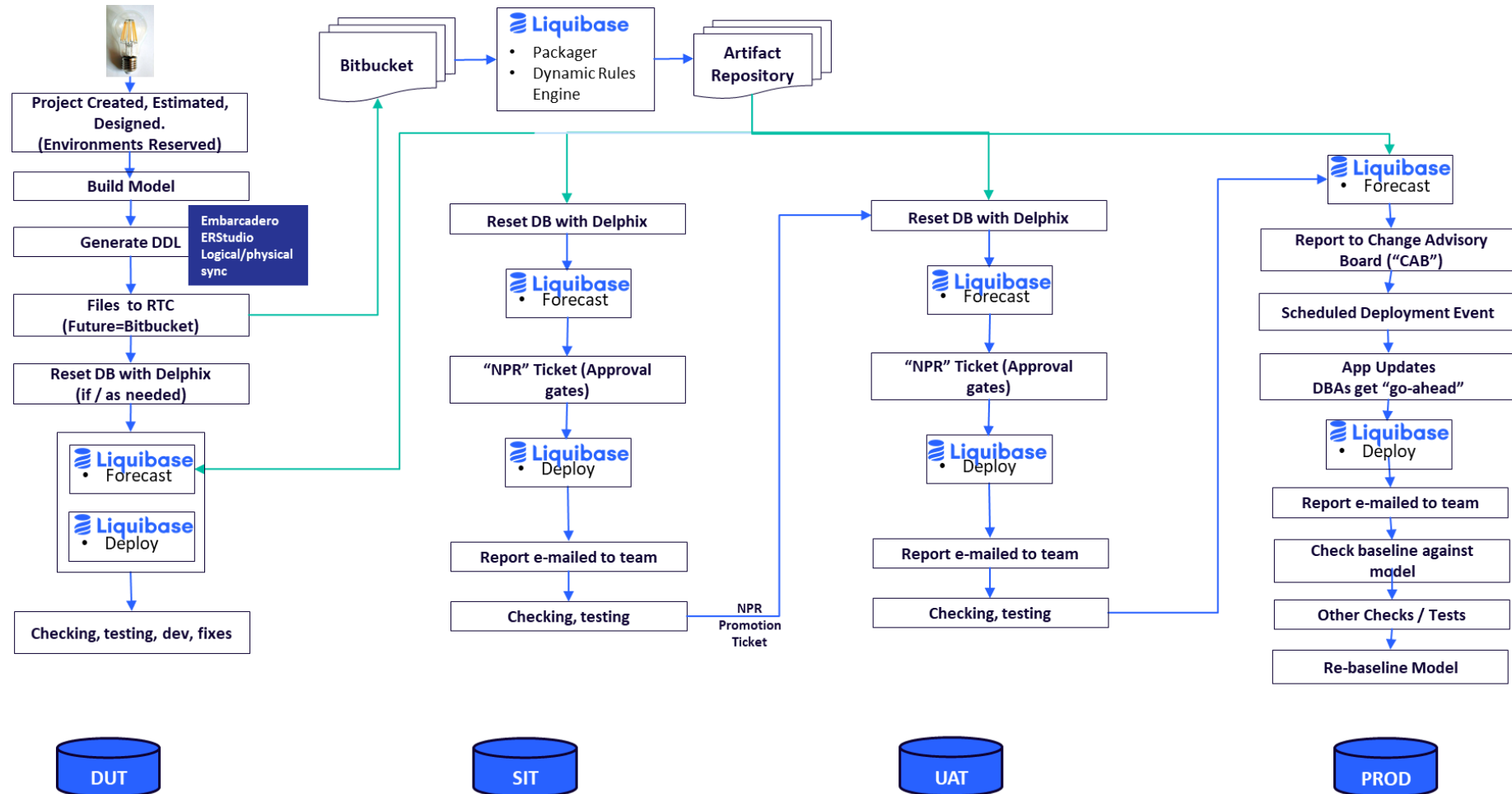
Simplify - Remove Complexity



Add Lightness - Remove the Unnecessary

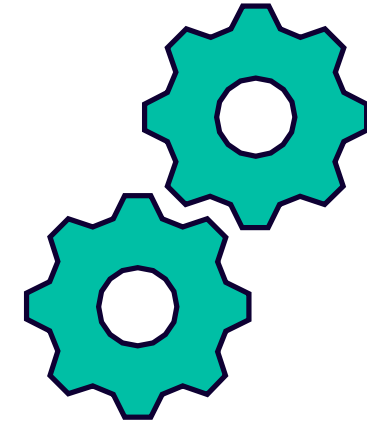


Add Lightness - Remove the Unnecessary



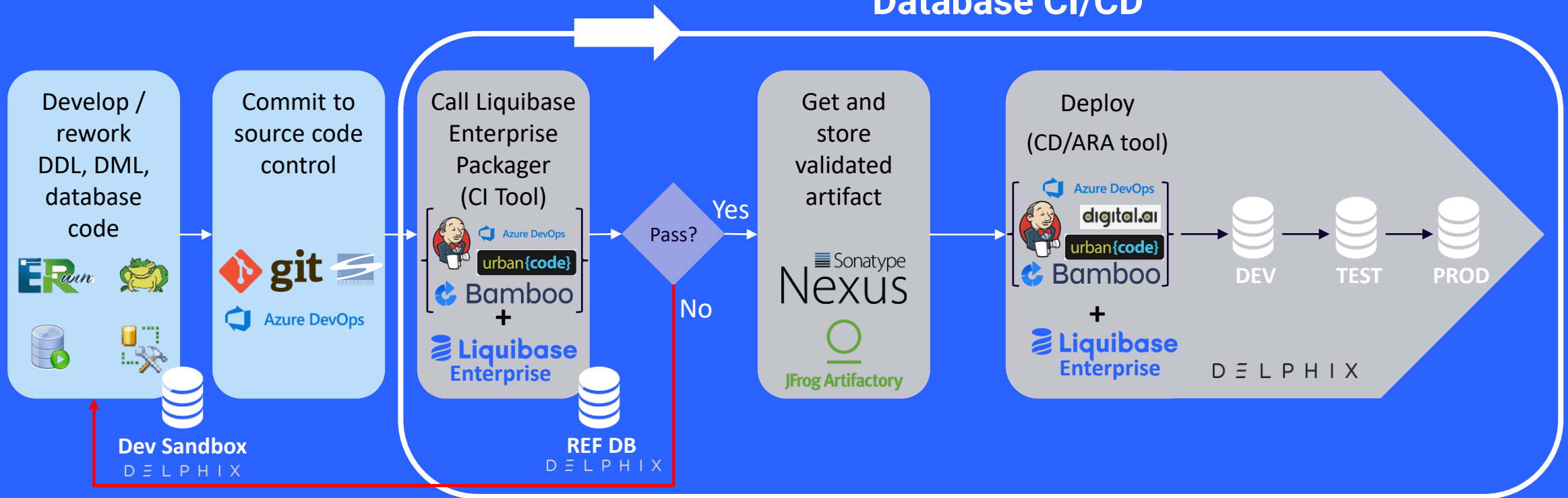
Automate - Remove Inconsistency

- Automation is the fastest way to go fast
- Automation ensures predictable results
- You can't have automation if your process is inconsistent



Automate - Remove Inconsistency

Database CI/CD



Monitor and Measure

- Health
- Impact of Improvements
- Problems
- Decision Support



Image: Johnson Space Center Mission Control, nasa.gov

Monitor and Measure

The screenshot displays the Liquibase HUB interface, which is used for monitoring and measuring database changes. The interface is divided into several sections:

- Header:** Features the HUB BETA logo, a "Take a Survey" button, and a "Log Out" button.
- Left Sidebar:** Contains navigation links for "dzentgraf's Personal Organization", "dzentgraf's Project", "Example01", "Operations", "Changelogs", "Changesets", and "Connections".
- Example01 > Connections:** Shows the project ID (48030cee-9d70-4d55-bf6f-0e41e7da18b0) and a list of connections. The selected connection is "jdbc:postgresql://localhost:5432/randoDEV".
- Example01 > Operations:** Displays a table of operations performed on the project. The table includes columns for Operation Type, Result, Changelog, Start Date, and End Date. The operations listed are updates and a rollback, all performed on 10/1/20.
- Operation Summary:** Provides a detailed view of the selected operation, including Connection Information, Client Information, Operation Parameters, and Runtime Information.
- Changeset Details:** Shows the details of the selected changeset (dan-003), including the author (dan), filepath (hubtest.xml), and duration (00:00:00.34). It also displays the changeset body, which is a JSON object representing the database change log.

Operation Type	Result	Changelog	Start Date	End Date
Update on 10/1/20, 8:39...	Success	hubtest.xml	10/1/20, 8:39 AM	10/1/20, 9:25 AM
Update on 10/1/20, 8:40...	Success	hubtest.xml	10/1/20, 8:40 AM	10/1/20, 9:25 AM
Update on 10/1/20, 9:25...	Success	hubtest.xml	10/1/20, 9:25 AM	10/1/20, 9:25 AM
Rollback on 10/1/20, 9:2...	Success	hubtest.xml	10/1/20, 9:25 AM	10/1/20, 9:25 AM
Update on 10/1/20, 9:27...	Success	hubtest.xml	10/1/20, 9:27 AM	10/1/20, 9:27 AM

Connection Information	Client Information	Operation Parameters	Runtime Information
Name: jdbc:postgresql://localhost:5432/randoDEV JDBC URL: jdbc:postgresql://localhost:5432/randoDEV Date Created: 10/1/20, 8:39 AM	Liquibase Version: 4.1.0 Hostname: DESKTOP-S7AGKRA System User: dan Interface: cli	defaultsFile_liquibaseProLicense: ABwwGgQU***** defaultsFile_changelogFile: hubtest.xml defaultsFile_driver: org.postgresql.Driver	Start Time: 10/1/20, 9:25 AM End Time: 10/1/20, 9:25 AM Duration: 00:00:00.00

Changeset Details

<> dan-003

Author: dan | Filepath: hubtest.xml | Duration: 00:00:00.34

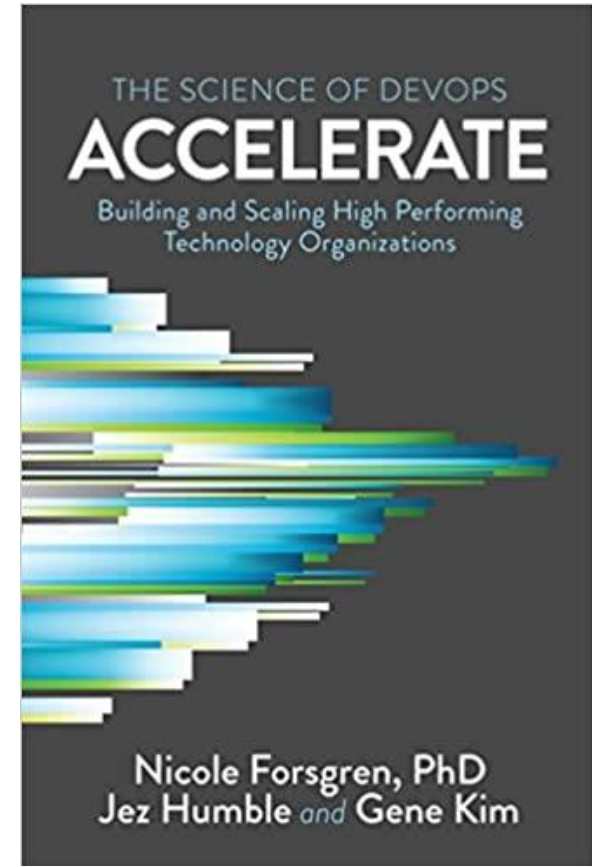
Changeset Body

```
{
  "databaseChangeLog": [
    {
      "changeSet": {
        "id": "dan-003",
        "author": "dan",
        "objectQuotingStrategy": "LEGACY",
        "rollback": {
          "changes": [

```

Analyze Trends Over Time

- Lead Time
- Deployment Frequency
- Change Fail Percentage
- Mean Time to Restore



Analyze Trends Over Time

Averages from actual customer data using DORA KPI metrics¹.


Deployment Frequency

 **3x** deployments/month

Lead Time Reduction

 **95%** delays eliminated

Change Failure Rate

 **94%** errors/month

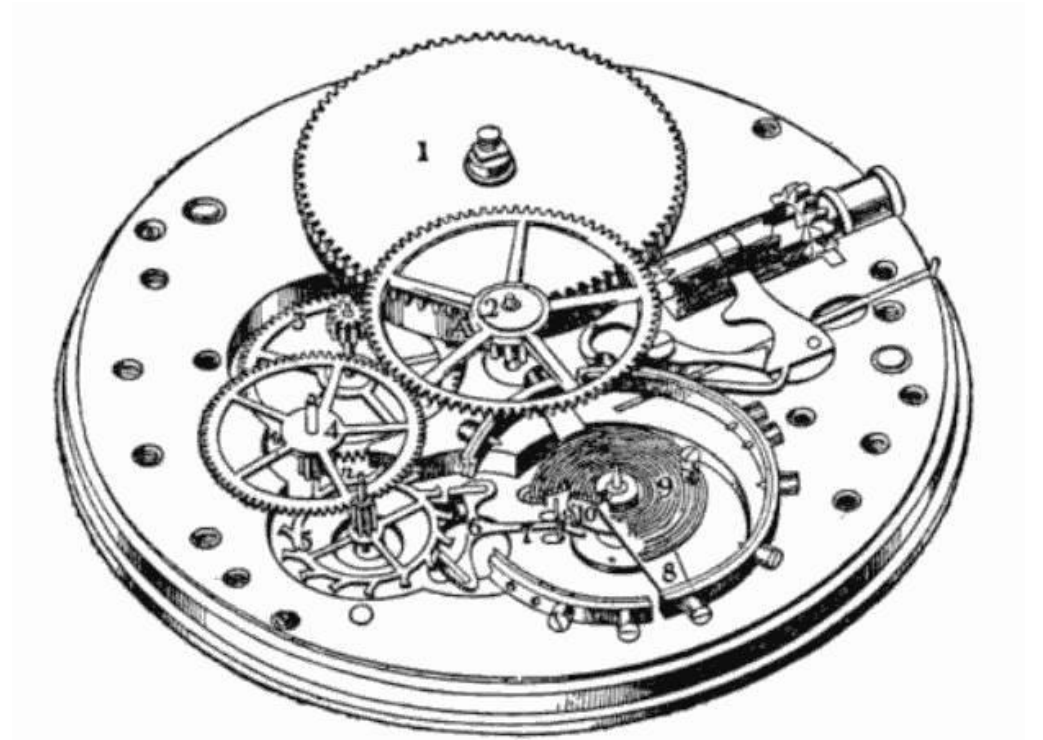
WIP Reduction

 **66%** changes/deployment

¹substituting WIP reduction for MTTR as recovery is not directly related to deployment performance.

Precise Control

- Ability to make very fine adjustments
- Avoid large 'swerves'
- Small errors compound quickly at speed
- Constantly adjust toward expected outcomes



Precise Control

My Projects > GauntletComm > Pipeline2

Columns
Date + 10 more

Filters

Date ↓	Filename	Version	Changeset ID	Labels	Contexts	Author	REF_DB2 Last Sync: 9/24/20, 9:21 PM Package: 9/24/20, 9:21 PM	DEV2 Last Sync: 9/24/20, 9:22 PM Deploy: 9/24/20, 9:22 PM	TEST2 Last Sync: 9/24/20, 2:41 PM Forecast: 9/24/20, 2:41 PM	PROD Last Sync: 9/24/20, 9:24 PM Forecast: 9/24/20, 9:24 PM
9/24/2...	ADD_EMPL...	1	clvx-3 (addColu...	3.0.0006,...		Mike Olivas	✓	✓	—	—
9/24/2...	ADD_EMPL...	1	clvx-2 (addColu...	3.0.0006,...		Mike Olivas	✓	✓	—	—
9/24/2...	ADD_EMPL...	1	clvx-1 (addColu...	3.0.0006,...		Mike Olivas	✓	✓	—	—
10/9/1...			dropForeignKey...	baseline,...		Robert L...	✓	✓	✓	✓
10/9/1...			createTable:Loca...	jira-102, ...		Robert L...	✓	✓	—	—
10/9/1...			addColumn:Ord...	jira-101, ...		Robert L...	✓	✓	—	—
12/21/...			m3rh-86 (addFor...	baseline		Administr...	✓	✓	✓	✓
12/21/...			m3rh-85 (addFor...	baseline		Administr...	✓	✓	✓	✓

Items per page: 10 1 - 10 of 92

Deploy mode: Auto Deploy Forecast Mode: No Forecast Labels: pipeline2

✓ **Deploy changes on DEV2**

System User: automation	Deployment Step: DEV2
System Name: WIN-20E107KB4TN	Database User: system
JDBC URL: jdbc:sqlserver://win-20e107kb4tn:1433;databaseName=GauntletComm_DEV2;integratedSecurity=false	DBMS: Microsoft SQL Server - 12.00.5207(12.0)
Driver: Microsoft JDBC Driver 6.3 for SQL Server	Platform: dev
Schema: dbo	Contexts: pipeline2
Start Time: Sep 24 2020, 9:22:29 PM CDT	Labels: exact
End Time: Sep 24 2020, 9:22:33 PM CDT	Enable row count: local
Total Time: 3 seconds	Stored Logic: validity
	Check: Stored Logic
	Validity: warn
	Action:

Successful Changes

Expand All Details

clvx-1 (addColumn tableName=Employees) ✓

Add column ("Facebook" nvarchar(30)) to table Employees

Change Set Labels: pipeline2,jira-1000,3.0.0006,add_employee_fields.sql

— Collapse Details

ID	Author	Date Executed	Elapsed Time
clvx-1 (addColumn tableName=Employees)	Mike Olivas	Thu Sep 24 21:22:32 CDT 2020	0.009 seconds

Generated SQL

```
ALTER TABLE [dbo].[Employees] ADD [Facebook] [nvarchar](30)
```

Uninterrupted Flow - Eliminate Rework

- First Time Complete and Accurate
- Build Quality In
- Use the pipeline tools and capabilities you have built to ensure problems caught quickly
- Avoid Flow Disruptions



Uninterrupted Flow - Eliminate Rework

The image is a collage of three overlapping screenshots from the Azure DevOps web interface, demonstrating a continuous integration and deployment (CI/CD) pipeline.

- Top Screenshot:** Shows a pipeline named "GauntletComm-PKG" with a status of "SUCCESS". The pipeline graph includes stages: Start, Source Checkout, PackageDB, BuildApp, Package Artifact, Deploy To DEV, Promotion Approval, Deploy To TEST, Promotion Approval, Deploy To PROD, and End. The "Package Artifact" stage is highlighted, showing a duration of 1m 53s and a commit made 3 years ago.
- Middle Screenshot:** Displays the "build_server_pdfs" pipeline, which is also in a "SUCCESS" state. It shows a list of steps with green checkmarks, indicating successful completion. The steps include "Spin up", "Attach", "Check", "Restore", "Install", "Saving", "Create", "Build", and "Persist".
- Bottom Screenshot:** Shows the "Enabling feature flags for Preview Attachment and Grid Views" pipeline. It details the "Linux Job" with a duration of 3m 29s. The job steps include "Prepare job", "Initialize job", "Get sources", "Cmdline", "Nodetool", and "Install dependencies". The terminal output for the "Install dependencies" step is visible, showing commands like "yarn install v1.7.0" and "npm run compile".



QUESTIONS

Be a Part of the Community!

- www.liquibase.org
- Official Liquibase Forum (forum.liquibase.org)
- Stack Overflow (tag: Liquibase)
- Discord Chat
- Reddit: [/r/Liquibase](https://www.reddit.com/r/Liquibase)
- Twitter: [@Liquibase](https://twitter.com/Liquibase)



Dan Zentgraf
Director, Solutions Architecture
dzentgraf@liquibase.com

See Liquibase in Action

- See On-demand Demos: <https://www.liquibase.com/demo>
- Learn about Liquibase Hub: <https://www.liquibase.com/hub>



THANK YOU



Liquibase

Speed in CI/CD: Get Faster to Get Better

It's not about being fast. It's about getting faster

Speed is central to discussions of the benefits of CI/CD. Too often, speed is presented as the goal of CI/CD. That's not entirely true. Speed is exciting, but by itself, it is just expensive. Speed is valuable because of the things that it brings with it. In every instance where people pursue speed, they have to systematically simplify, lighten, and improve the thing they are trying to make fast.

This webinar will dive into what the pursuit of speed brings to DevOps and CI/CD so that you can break through the hype and truly improve your software process. It's not about speed for the sake of speed; it's about eliminating needless complexity and adding smart automation. Now that we're living in a COVID world, more companies are focusing on speed and will leave those that aren't far behind.

Punchline

- All Agile and DevOps assume Continuous Improvement
- Standing pat at a given speed yields stagnation
- A prime driver of Improvement comes from *pursuit* of VELOCITY
- That's why the digital natives have pushed to the point where they can deploy in minutes or seconds



About Liquibase