PNSQC

OCTOBER 10-12 2022

RICHARD ROBINSON
SMART PIPELINES

ANNUAL
TWARE
ALITY
FERENCE
-12, 2022

RICHARD ROBINSON

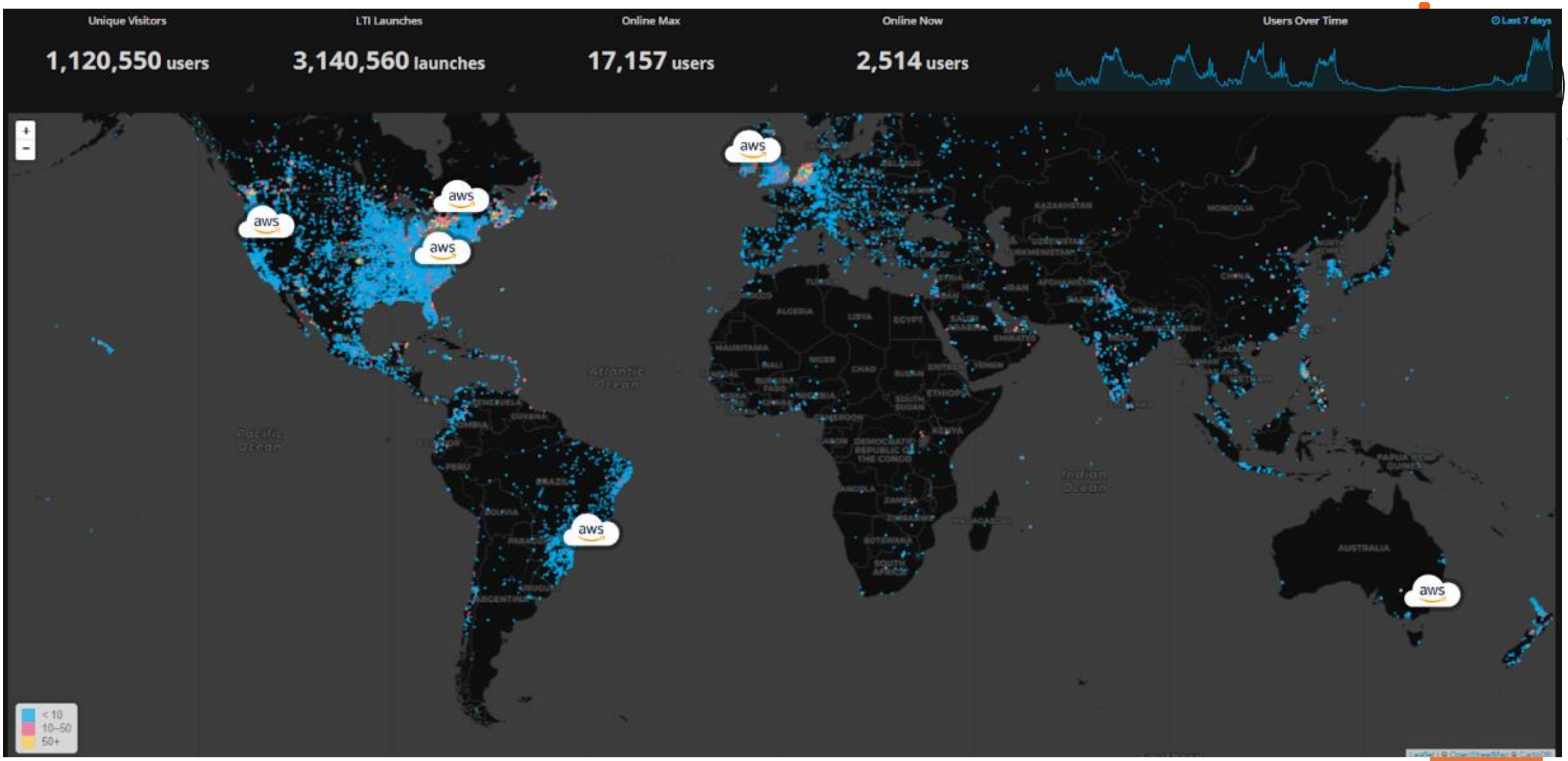
SMART PIPELINES

40TH ANNUAL PACIFIC NW SOFTWARE QUALITY CONFERENCE OCT. 10–12, 2022

RICHARD ROBINSON

SMART PIPELINES

PACIFIC NW SOFTWARE QUALITY CONFERENCE
OCT. 10–12, 2022

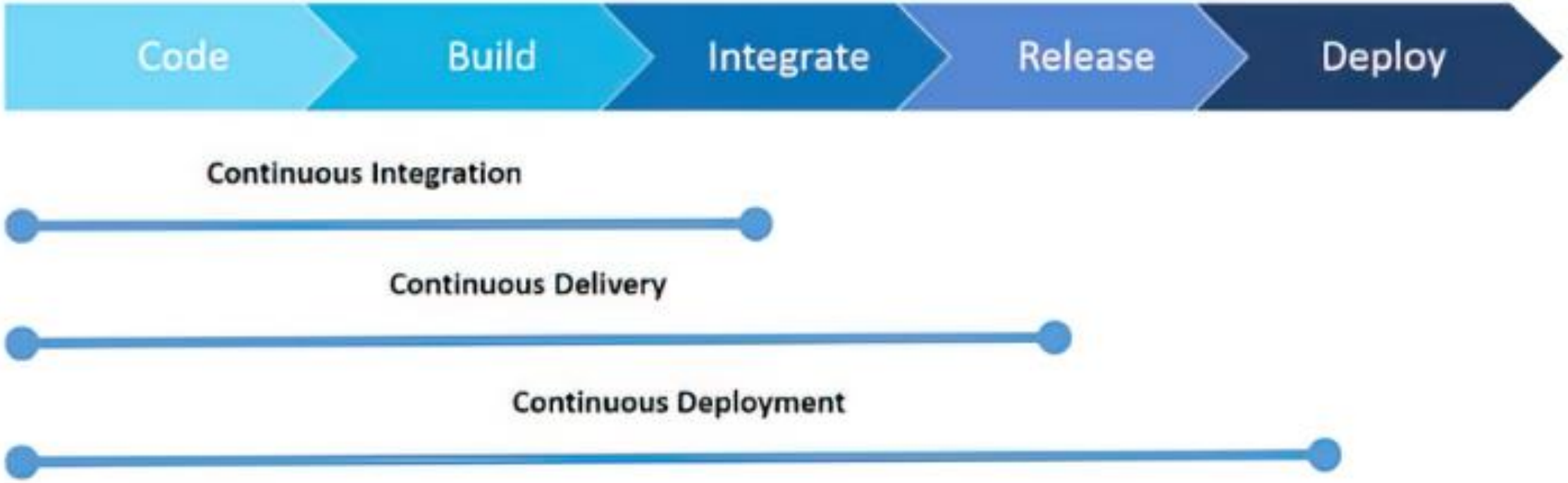


RICHARD ROBINSON

SMART PIPELINES

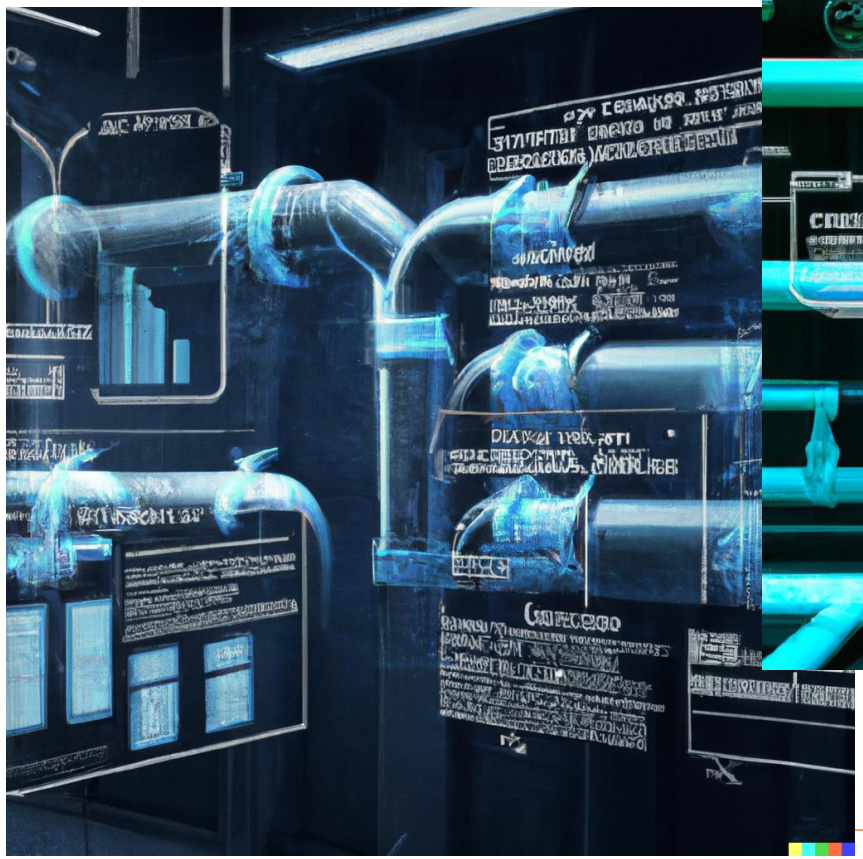
PACIFIC NW SOFTWARE QUALITY CONFERENCE
OCT. 10–12, 2022

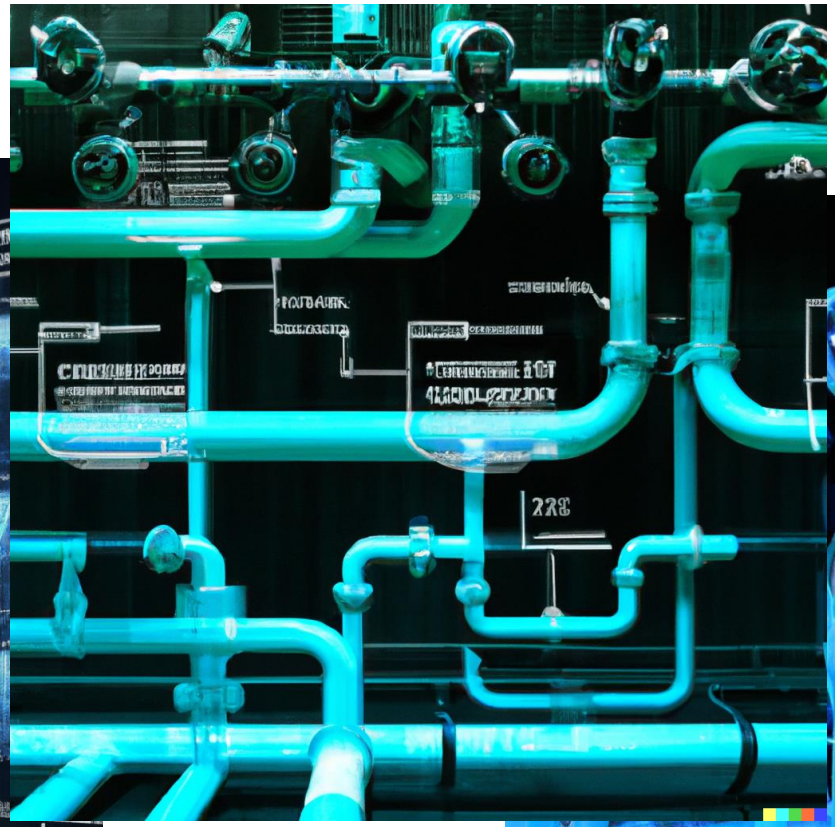
CI/CD Pipelines



Source: https://www.saviantconsulting.com/blog/difference-between-continuous-integration-continuous-delivery-and-continuous-deployment.aspx

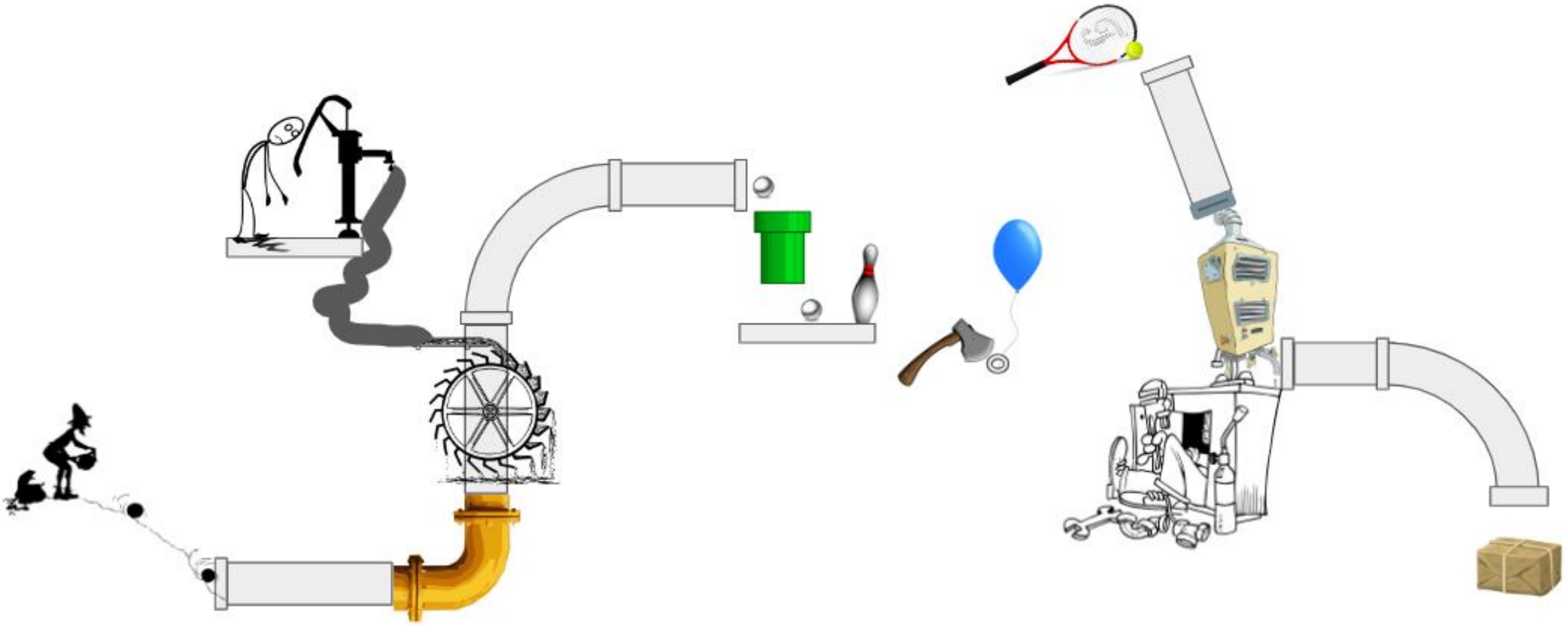
Better Quality Pipelines





- RICHARD ROBINSON
- SMART PIPELINES

A “Rube Goldberg” Pipeline



Potential Pipeline Issues

- Unreliable pieces

- Pieces requiring manual effort (human toil)

- Inconsistent pieces

- Brittle, requiring constant fixing/manual intervention, workarounds, etc.

Improving a Pipeline (an example)

Steps to move forward:

1. **Understand all the pieces** and each step involved in the deployment
2. Create a **checklist** to ensure that the manual and brittle process could be performed without missing key steps to keep supporting the business by releasing new features and fixes until automation could be built and other improvements implemented
3. **Eliminate downtime** - investigate and employ techniques and tools so each step could be performed and deployed without downtime
4. Need **representative test environments** to test the whole process internally and build automation against

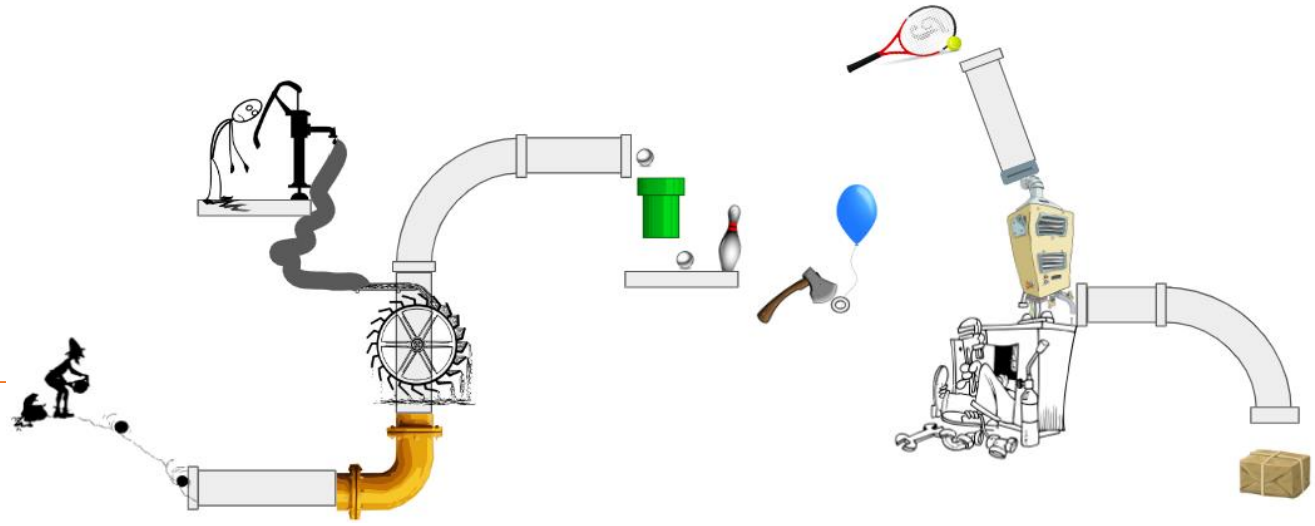
Checklists

In today's modern world of complexity whether in surgery, building modern skyscrapers, flying an airplane, or building and deploying software the volume and complexity our human brains are dealing with needs a level of assistance to consistently get it right. **"…the volume and complexity of what we know has exceeded our individual ability to deliver its benefits correctly, safely, or reliably."**

"The Checklist Manifesto: How to Get Things Right" - Atul Gawande

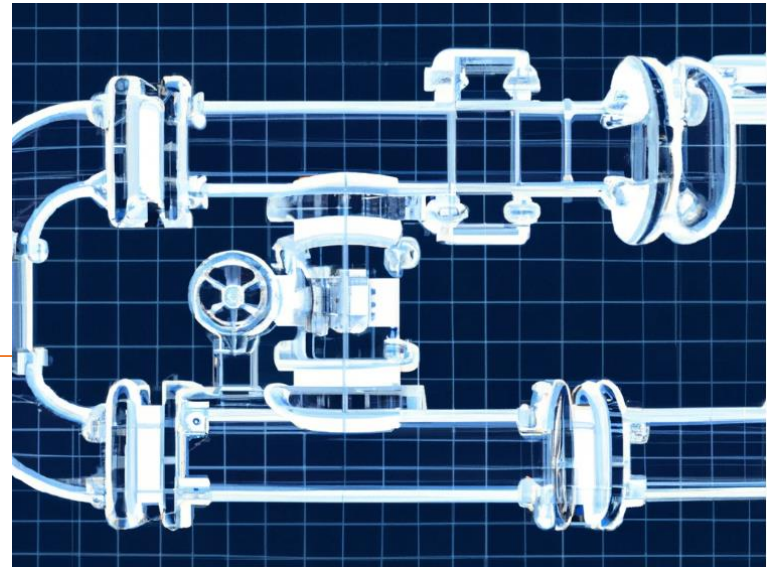
Checklists – Improving Pipelines

- Document **which steps** are done today
- **Proper sequencing** of steps if needed
- What level of **detail?**
  - Don’t always need every detail and specific piece
  - Sometimes just high-level reminders of key steps is sufficient
  - Link to additional detail as needed



Checklists – Improving Pipelines

- Helps ensure **proper execution** even with current pipe
- The pipeline starts to take shape creating a kind of **blueprint/map** of what needs to be replaced/worked on
- More easily **communicate** plans, progress, and ROI
- **Satisfaction** as the team turns the checklist items that are automated a different color while retaining them in the checklist or just completion



Checklists – Pipeline Deployment Example

Phase	Item	State	Internal - 1	Internal - 2	Internal - 3	Internal - 4	Internal -	Staging	AU	EU	SA	CA	NA	OR
Pre-Upgrade	Clear out auto courses	AUTOMATED	Complete	Complete	Complet	Complete	Complet	Complete	Complet	Complete	Complete	Complete	Complete	Complete
	Populate /partners_global_mapping/bongo_api_base_url key in consul		Complete	Complete	Complet	Complete	Complet	Complete	Complet	Complete	Complete	Complete	Complete	Complete
	Reset DB schema for new column		Complete	Complete	Complet	Complete	N/A	Complete	N/A	N/A	N/A	N/A	N/A	N/A
Upgrade-Predeploy	DB Snapshot	AUTOMATED	N/A	N/A	N/A	N/A	Complet	Complete	Complet	Complete	Complete	Complete	Complete	Complete
bongo_229_predeploy.rb	Update schema	AUTOMATED	Complete	Complete	Complet	Complete	Complet	Complete	Complet	Complete	Complete	Complete	Complete	Complete
	Update sentry version in consul to 22.9	AUTOMATED	Complete	Complete	Complet	Complete	Complet	Complete	Complet	Complete	Complete	Complete	Complete	Complete
Upgrade	Adjust branch mapping and masking to the appropriate environment(s)		Complete	Complete	Complet	Complete	Complet	Complete	Complet	Complete	Complete	Complete	Complete	Complete
Microservices	bongo-lti (env by env fine)		Complete	Complete	Complet	Complete	Complet	Complete	Complet	Complete	Complete	Complete	Complete	Complete
	data-layer (env by env fine)		Complete	Complete	Complet	Complete	Complet	Complete	Complet	Complete	Complete	Complete	Complete	Complete
	Enable Insights for the X-Ray Default group (one per region)		Complete	Complete	Complet	Complete	Complet	Complete	Complet	Complete	Complete	Complete	Complete	Complete
Upgrade	Merge bongo repo changes (or re-run circleci build for internal environments or masked in prod)	AUTOMATED	Complete	Complete	Complet	Complete	Complet	Complete	Complet	Complete	Complete	Complete	In Progres	In Progress
Front-end and back-end	Merge bongo-ui repo changes (or re-run circleci build for internal environments or masked in prod)	AUTOMATED	Complete	Complete	Complet	Complete	Complet	Complete	Complet	Complete	Complete	In Progres	In Progres	In Progress
	Wait for periscope and codedeploy to finish	AUTOMATED	Complete	Complete	Complet	Complete	Complet	Complete	Complet	Complete	Complete	In Progres	Not Starte	Not Startec
	Verify php basic functioning (can rollback in codedeploy within 5 min)		Complete	Complete	Complet	Complete	Complet	Complete	Complet	Complete	Complete	In Progres	Not Starte	Not Startec
Microservices	bongo-integration (needs to deploy to all regions simultaneously, but could go anywhere)		Complete	Complete	Complet	Complete	Complet	Complete	N/A	N/A	N/A	N/A	Not Starte	Not Startec
Verification	Confirm launch from integrated partner		Complete	Complete	Complet	Complete	Complet	Complete	Complet	Complete	Complete	In Progres	Not Starte	Not Startec
	Admin panel	AUTOMATED	Complete	Complete	Complet	Complete	Complet	Complete	Complet	Complete	Complete	In Progres	Not Starte	Not Startec
	checksum/builddate of npm - alt-`	AUTOMATED	Complete	Complete	Complet	Complete	Complet	Complete	Complet	Complete	Complete	In Progres	Not Starte	Not Startec
	General LTI launches working	AUTOMATED	Complete	Complete	Complet	Complete	Complet	Complete	Complet	Complete	Complete	In Progres	Not Starte	Not Startec
	Try out creation of all 4 assignment types	AUTOMATED	Complete	Complete	Complet	Complete	Complet	Complete	Complet	Complete	Complete	In Progres	Not Starte	Not Startec
	Video Analysis - including content analysis with metrics over time	AUTOMATED	Complete	Complete	Complet	Complete	Complet	Complete	Complet	Complete	Complete	In Progres	Not Starte	Not Startec
	Confirm general link and deep link launch from open source partne	AUTOMATED	Complete	Complete	Complet	Complete	Complet	Complete	Complet	Complete	Complete	In Progres	Not Starte	Not Startec
	Processing of videos	AUTOMATED	Complete	Complete	Complet	Complete	Complet	Complete	Complet	Complete	Complete	In Progres	Not Starte	Not Startec
	Combining videos	AUTOMATED	Complete	Complete	Complet	Complete	Complet	Complete	Complet	Complete	Complete	In Progres	Not Starte	Not Startec
	LTI 1.3 stuff is setup -- set up new institution		Complete	Complete	Complet	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Not Startec
	LTI 1.3 stuff is setup -- confirm ALB has new pieces (/lti13/login, /lti	13/launch, /lti13/j	Complete	Complete	Complet	Complete	Complet	Complete	Complet	Complete	Complete	In Progres	Not Starte	Not Startec
	Activity Overview performance test		N/A	N/A	N/A	N/A	N/A	N/A	Complet	N/A	N/A	N/A	Not Starte	Not Startec

RICHARD ROBINSON

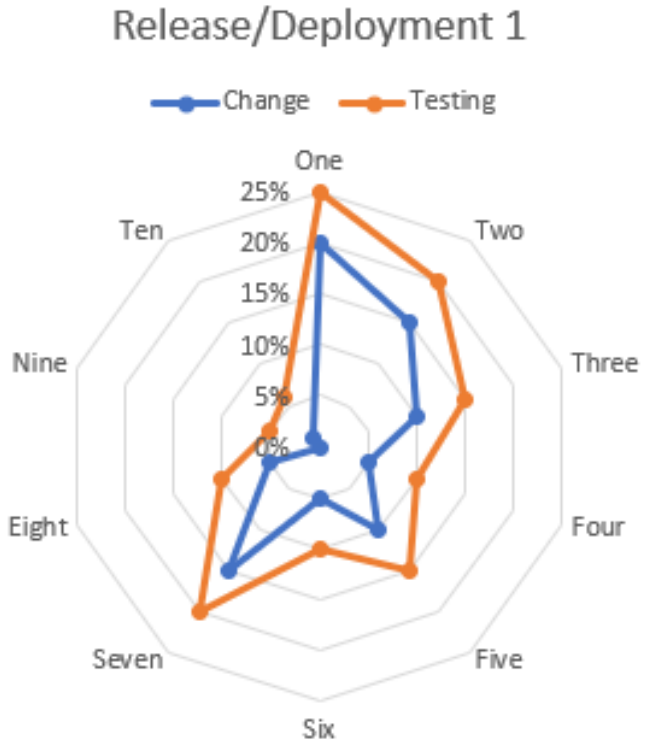
SMART PIPELINES

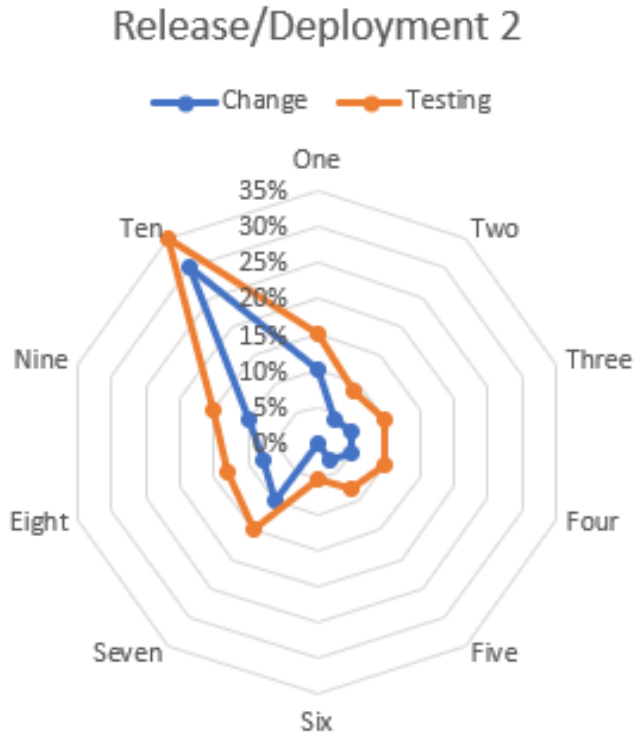
40TH ANNUAL PACIFIC NW SOFTWARE QUALITY CONFERENCE
OCT. 10–12, 2022

Checklists – QA and Testing Example

High-level Area or Consideration	Details/Description
Accessibility	Does this system properly support accessibility with low vision and screenreader support? Also closed captioning capabilities
Performance	Performance tests to ensure that the system's response times and throughput meet the user expectations and meet specified performance criteria or goals.
Scale/Sizing	Horizontal and Vertical scaling considerations
Stress	Pushing the boundaries of performance limits or even just limits of specific fields/types (e.g., numeric limits for integer fields or overflowing string sizes especially when storing in DB tables with columns of certain types)
Security & Privacy	Security tests to determine how secure the system is and if specific security requirements have been met. Could include GDPR, FERPA, HIPAA, data residency restrictions, etc.
Upgrade and Migration	Data preserved after upgrades and properly migrated to any new formats
Stability/Reliability	Tests performed to run the product in a customer-like environment over a period of time to verify that the system remains stable and there are no significant memory/handle/thread leaks or degradation to the system over time.
Usability	Is the system usable without intensive training or use of workarounds? Is it suitable for the target user community? Ease of use and standard UI behavior is good to consider here as well including use of phones, tablets, laptops, etc.
Supportability and Maintainability	Are there logs, debug levels, design docs, troubleshooting guides, API specs, and other things in place so that the product can be supported
Etc.	

Checklists – Tailor to a Release or Focus





RICHARD ROBINSON

SMART PIPELINES

40TH ANNUAL PACIFIC NW SOFTWARE QUALITY CONFERENCE OCT. 10–12, 2022

Eliminate Downtime

- How reliable does it need to be?
- How much downtime is ok?
- How much risk can we tolerate?

"Site Reliability Engineering seeks to balance the risk of unavailability with the goals of rapid innovation and efficient service operations, so that users' overall happiness—with features, service, and performance—is optimized." *Google Site Reliability Engineering.* O'Reilly Media

Eliminate Downtime

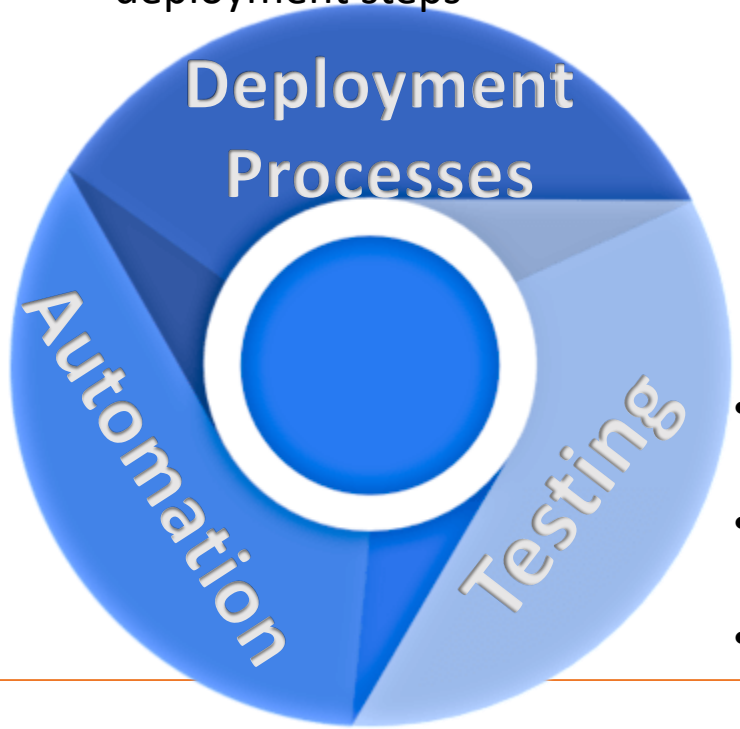
Deployment Strategies and Considerations:

- blue/green or black/red deployment approaches
- Rolling Upgrades
- Canary releases
- Feature Flags
- Backward compatibility
- Rollbacks
- Health and Monitoring
- Testing

Eliminate Downtime

Our journey from 3-4 hours downtime x 6 environments to zero

- Understand each step and build a robust checklist
- Blue/green deployment plans
- Evolve toward better dockerization and cleaner deployment steps



- Automate complex or brittle steps first
- Ensure adequate reviews and good design even for short-term automation
- Continue to build out automation even automating simple/quick steps (no manual steps should be done in production and even simple/quick steps done by humans can be error-prone and time consuming at scale)

- Increase confidence in our manual and automated test suites
- Increase confidence in our manual and automated deployment steps
- Better internal test environments

Internal Test Environments

Mimic production environments

- Same deployment process
- Representative data
- Similar sizes
- Permissions
- Monitoring
- Other aspects… maybe very custom for specific app
- Continue to keep up to date/in sync and improve

Internal Test Environments

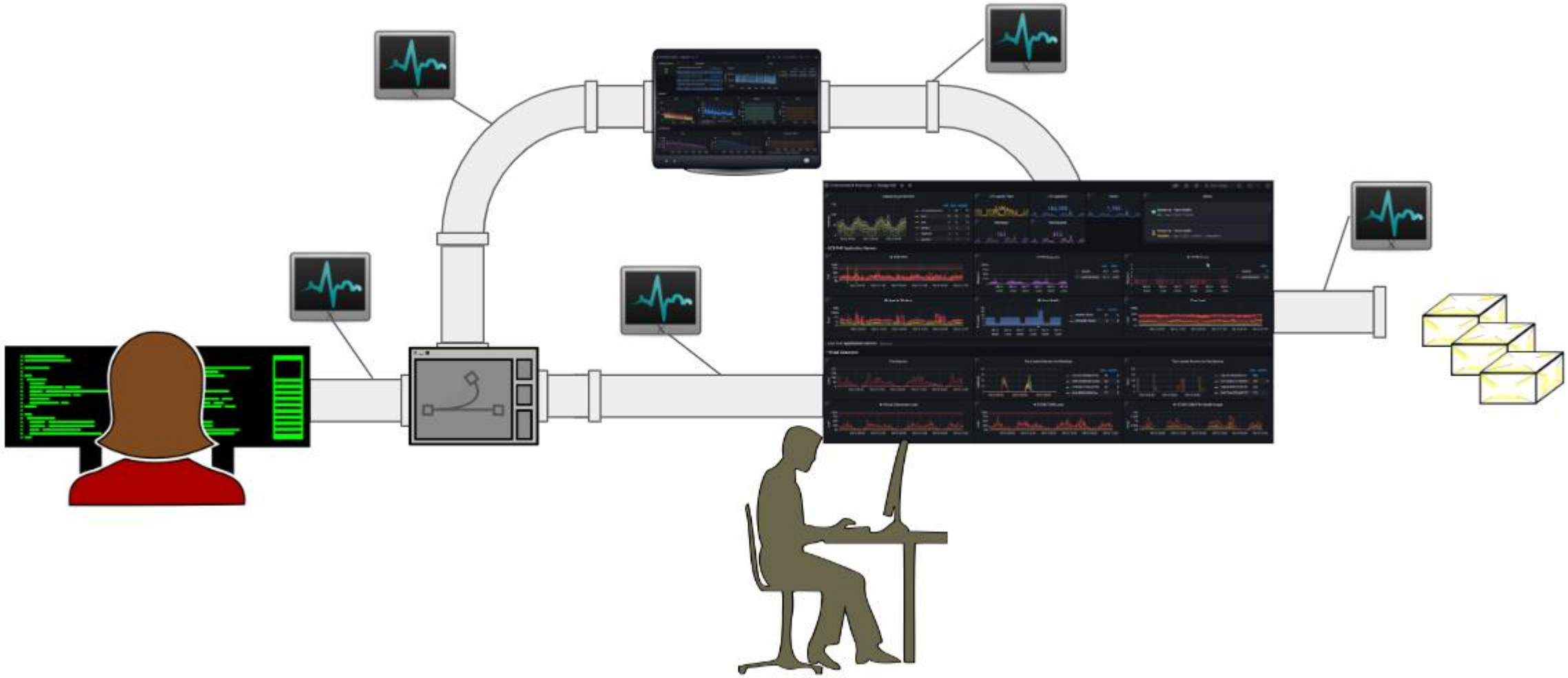
The important principle is to make sure there are internal environments that are close enough to production that steps, checklists, and automation used against those environments will behave the same as in production. This should be considered all the way back to development environments for engineers as much as possible.

RICHARD ROBINSON

SMART PIPELINES

40TH ANNUAL PACIFIC NW SOFTWARE QUALITY CONFERENCE
OCT. 10–12, 2022

Components of a Solid Pipeline



40TH ANNUAL PACIFIC NW SOFTWARE QUALITY CONFERENCE
OCT. 10–12, 2022

Some Components of a Solid Pipeline

- Flexible and Maintainable - need to be able to handle a variety of tools and be able to change, extend, and update
- Automated Scans and Tests - static code analysis, unit tests, functional tests, integration tests, end-to-end tests, specialty tests like perf/load/scale/localization/accessibility tests, etc.
- Monitoring - behavior as software goes through the pipeline and into production. If there is no visibility into the pipe until it comes out the other end, then that just adds risk and surprises into a system that needs to be robust and deterministic. Synthetic usage, regular heartbeats…
- Tooling and Automation – remove human component, enable thorough monitoring while things are in internal environments, traveling through the pipeline, and in production

Flexible and Maintainable

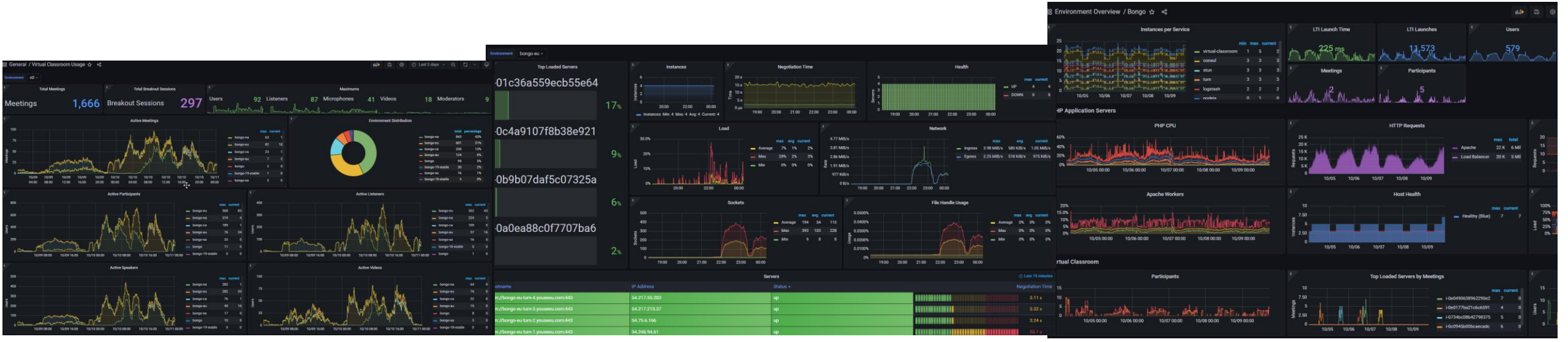
- Needs to handle any code we are going to deploy to production
- Need to plan in updates for all components – owned or dependencies
- Part of ‘done criteria’ is automated deployment process as well as monitoring, automated tests, etc.

Automated Scans and Tests

- Static code analysis – code quality, licensing, security scans, etc.
- Various tests run at proper stage of pipeline – especially end-to-end tests as it is deployed into production
- Where in the pipeline to run other tests:
  - Unit tests
  - Functional tests
  - Integration tests
  - Perf tests
  - Load tests
  - Scale tests
  - L10N tests
  - A18N tests

Monitoring

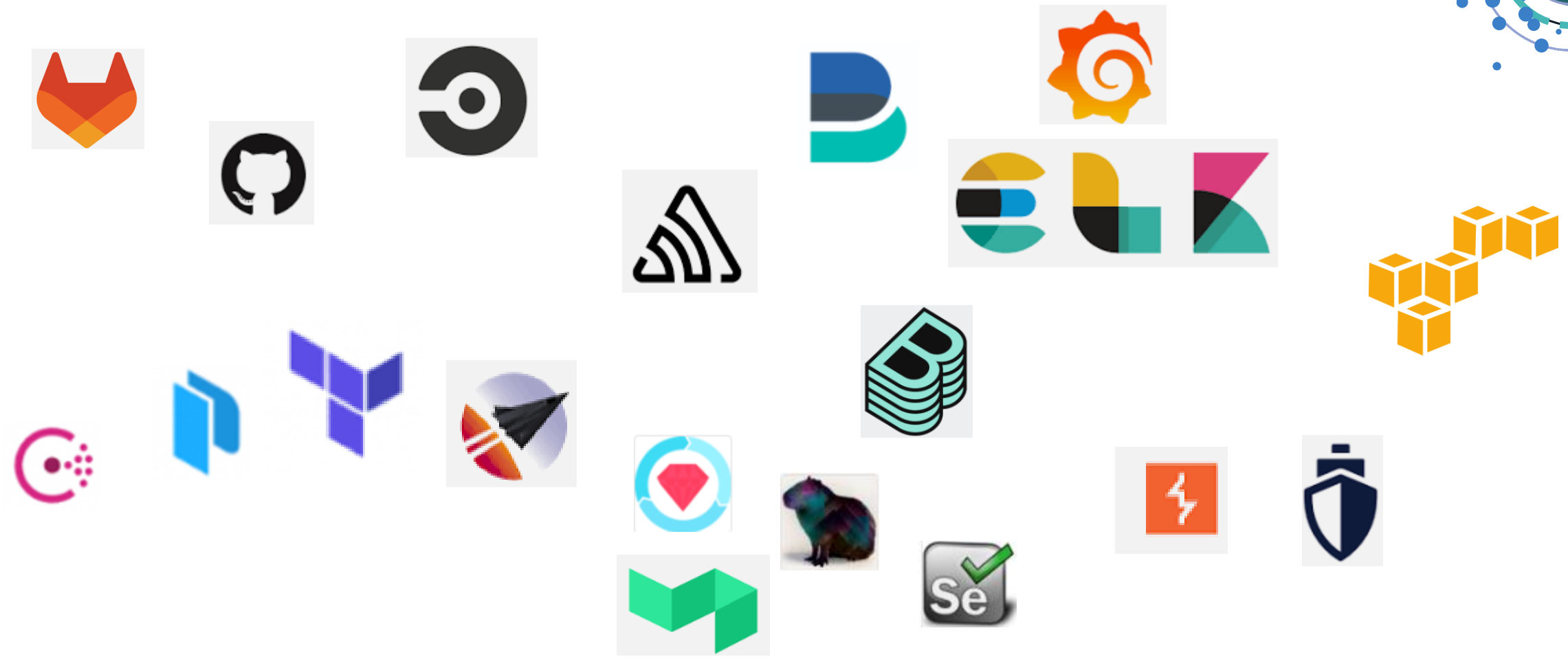
- Need to see how things are behaving and working throughout the pipeline.
- If there is no visibility into the pipe until it comes out the other end, then that just adds risk and surprises into a system that needs to be robust and deterministic.
- Status reporting, Key Health Indicators (KHIs) or Key Performance Indicators (KPIs), Usage reporting, and Auditing
- Synthetic usage, regular heartbeats, other test results



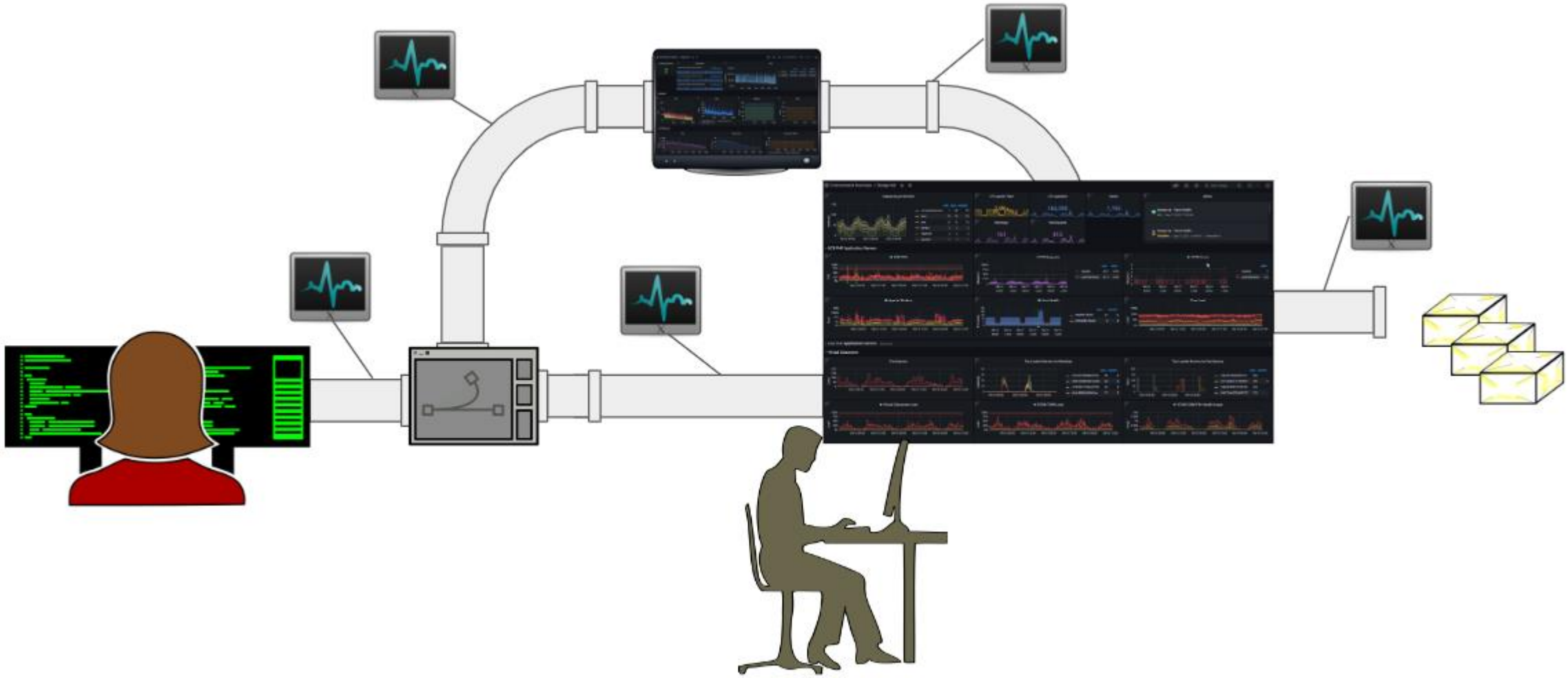
Tooling and Automation

- Enable thorough monitoring throughout pipeline
- Don’t reinvent the wheel – use tools available
- Remove the human variable. Automate everything – especially anything happening in production or representative internal environments

Some Potential Tools

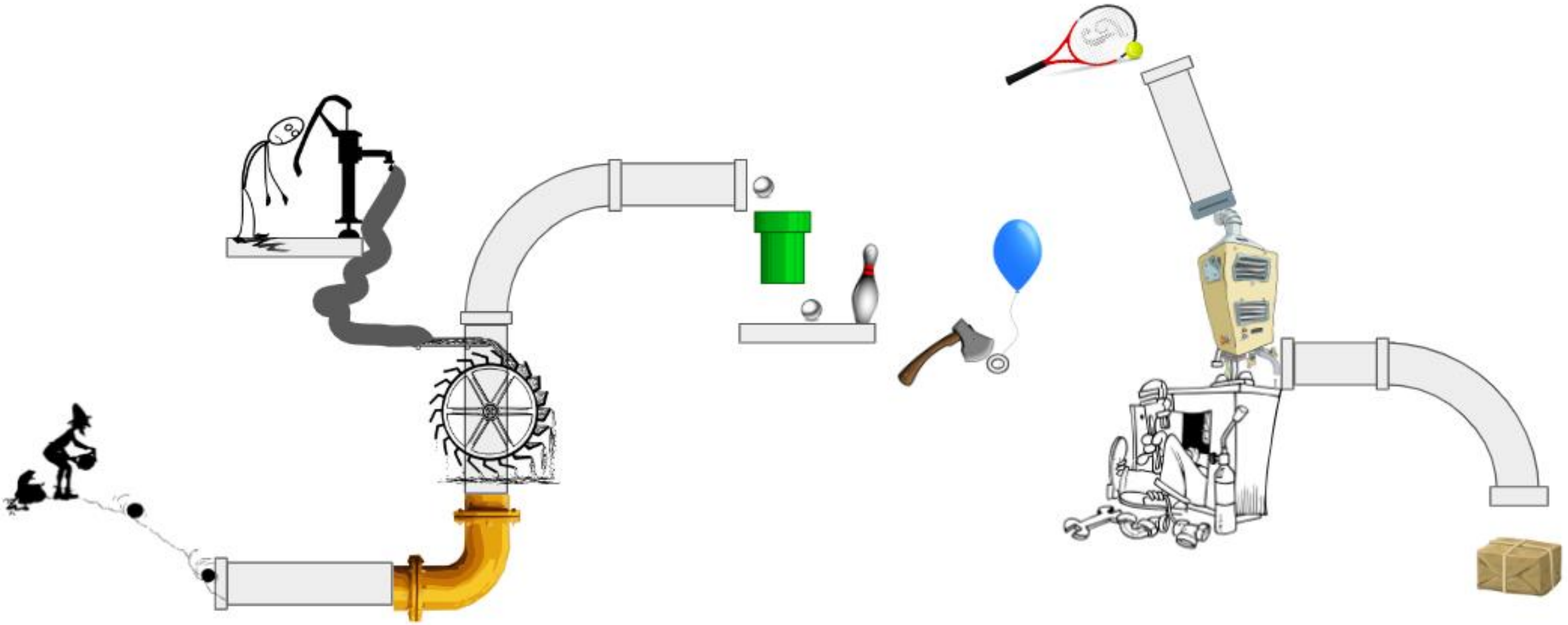


Components of a Solid Pipeline



40TH ANNUAL PACIFIC NW SOFTWARE QUALITY CONFERENCE
OCT. 10–12, 2022

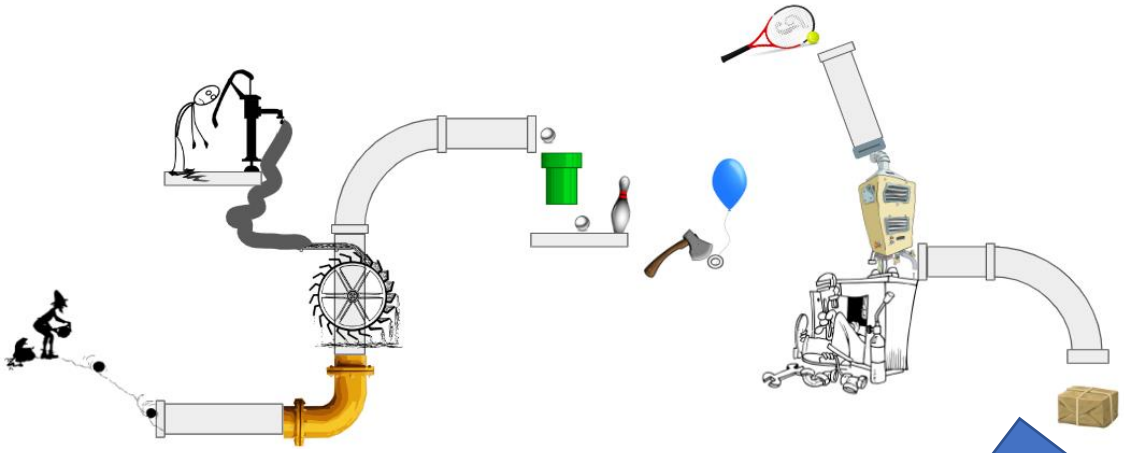
A “Rube Goldberg” Pipeline



Summary/Conclusion

- Checklists – Help ensure immediate quality and fill in the current gaps (w/ automation, tools, expertise, whatever) and provide a blueprint/map of what is needed to work on next
- Incremental Progress - Show ROI and pipeline improvements along the way. Even without all the pieces automated or finished benefits are still realized as each piece itself is solidified and improved.
- Build in Quality, Testing, and Automation Throughout - Risk-based testing approaches, representative internal environments, building confidence in your manual and automated suites, automating all steps, etc.
- Other parts of a solid Pipeline - Various aspects of a pipeline that can be overlooked as we might focus too much on just putting code into it and it plopping out the other side into production. Flexibility, maintainability, monitoring, code scanning, integrated testing, consistency, etc.

All the best in your Pipeline Improvement Journey!



RICHARD ROBINSON

SMART PIPELINES

Questions?

PNSQC

OCTOBER 10-12 2022

Richard.Robinson@bongolearn.com
https://www.linkedin.com/in/richard-robinson-b4a22b2/

THANK YOU