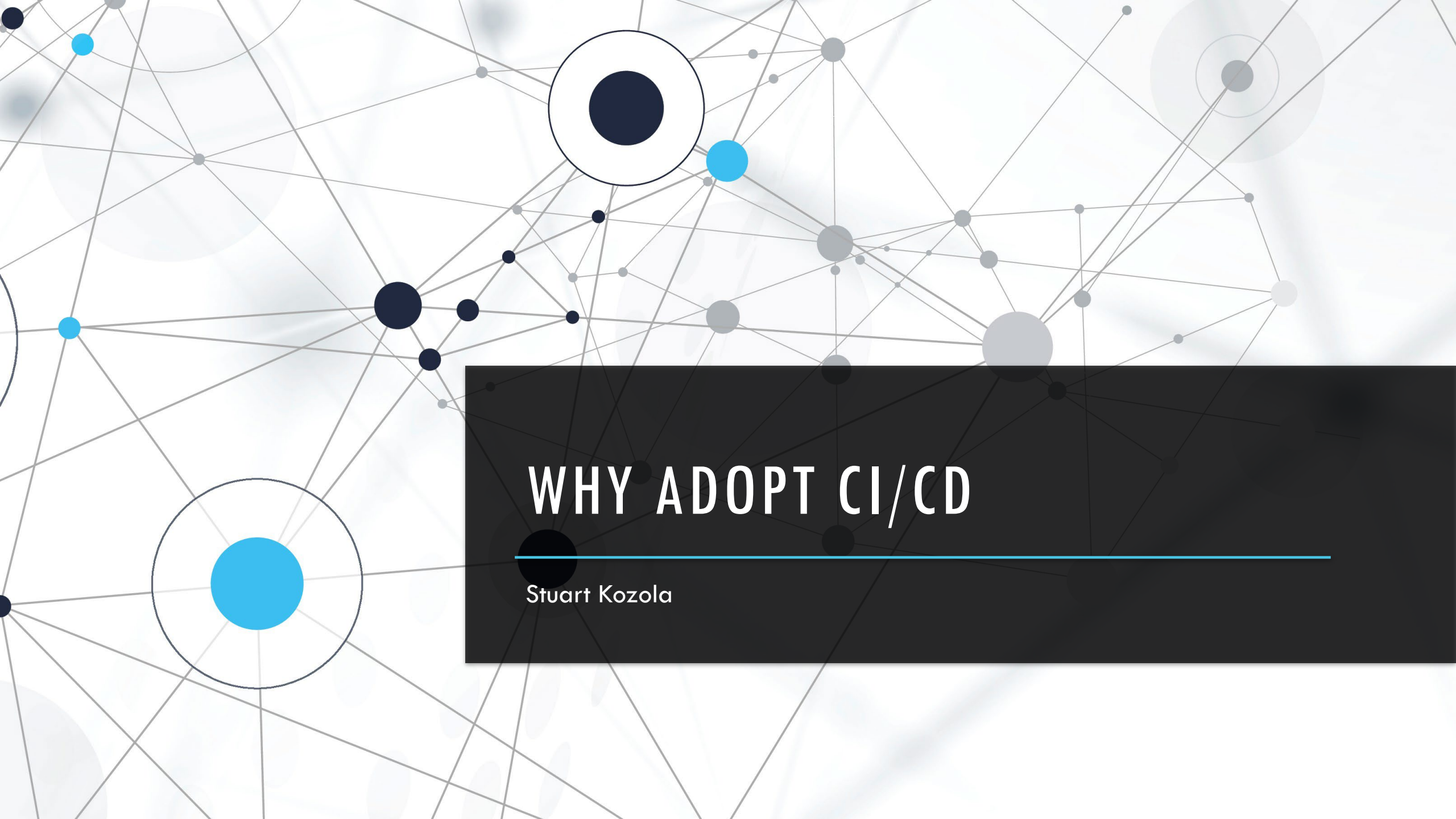


WHY ADOPT CI/CD

Stuart Kozola

AGENDA

Current Challenges with our product development and delivery process

What is Continuous Integration

What is Continuous Deployment

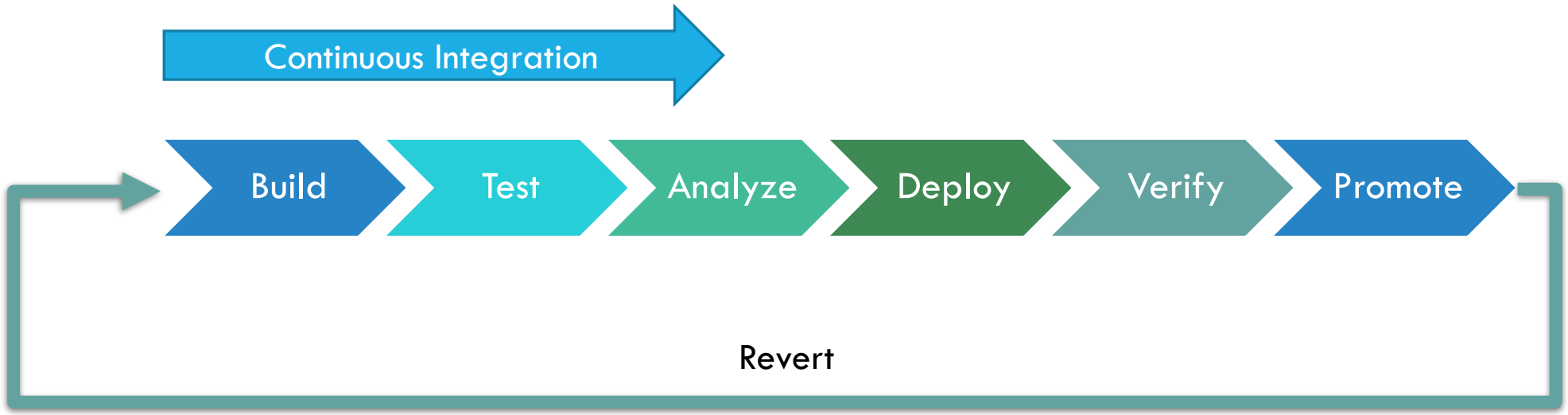
Value of Continuous Delivery

CURRENT CHALLENGES AND DESIRED STATE

	Current State	Pain	Desired State	Practice
 More time innovating (developing software)	Multiple developers Inconsistent code/binaries Manual root cause analysis Inconsistent test coverage Limited visibility into trends	1 **Unable to integrate and automate** software development	Unified and consistent build process Automated packaging of binaries Faster recovery Automated unit/integration testing Full visibility and transparency	CONTINUOUS INTEGRATION
 Increased Quality	Inconsistent releases Slow releases of features Time consuming bug hunts Manual deployments Limited monitoring of issues	2 **Limited visibility and traceability of quality** across the product Lifecyle	Reliable/timely product releases Fast feedback loops from customers Full visibility into quality Automated deployment of products Real-time alerting of issues	CONTINUOUS DEPLOYMENT
 Maximize ROI	Silos: code + tools + teams Manual processes Inconsistent versions Escaped defects Limited insight into failures	3 **High cost reduces investments in the business**	Reliable and repeatable process Automation of everything Versioning of artifacts Continuous quality monitoring Fast innovation	CONTINUOUS DELIVERY

CONTINUOUS INTEGRATION (CI)

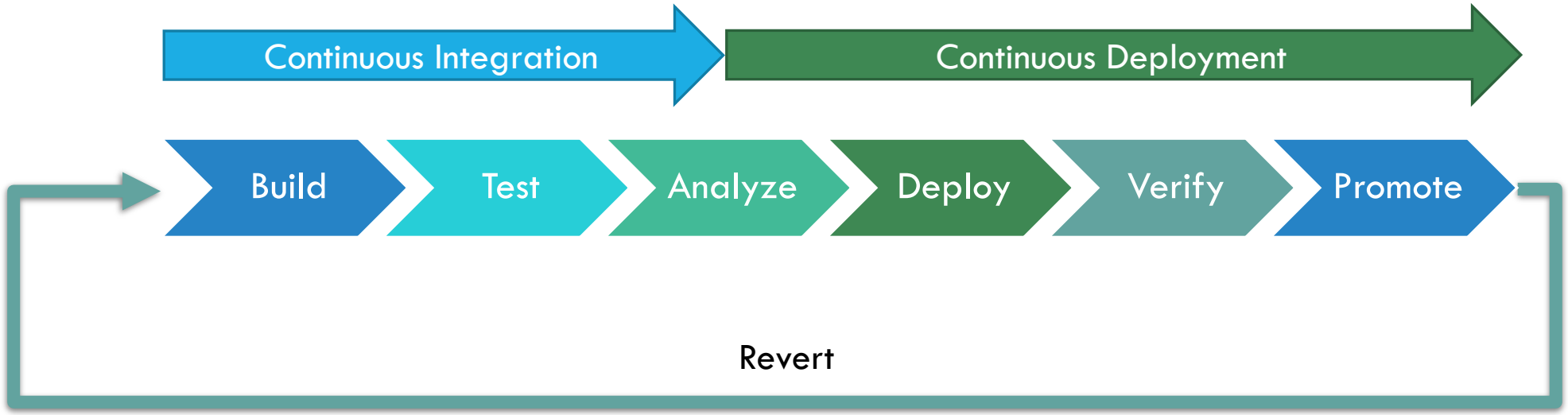
Continuous Integration is the practice of merging all working copies of code changes from multiple contributors into a unified, versioned repository for a single software project. Reliable, maintainable, and high-quality code artifacts are generated through automated builds, automated testing, syntax review, monitoring, and alerting of the build-test-run process of software development.



Continuous Delivery Pipeline

CONTINUOUS DEPLOYMENT (CD)

Continuous Deployment is the practice of releasing software to end-users as soon as they are ready with automated deployments. CD is enabled through automation of infrastructure, provisioning of systems, verification (security/smoke testing), promoting to production environments, and rolling back from failures to a prior release.

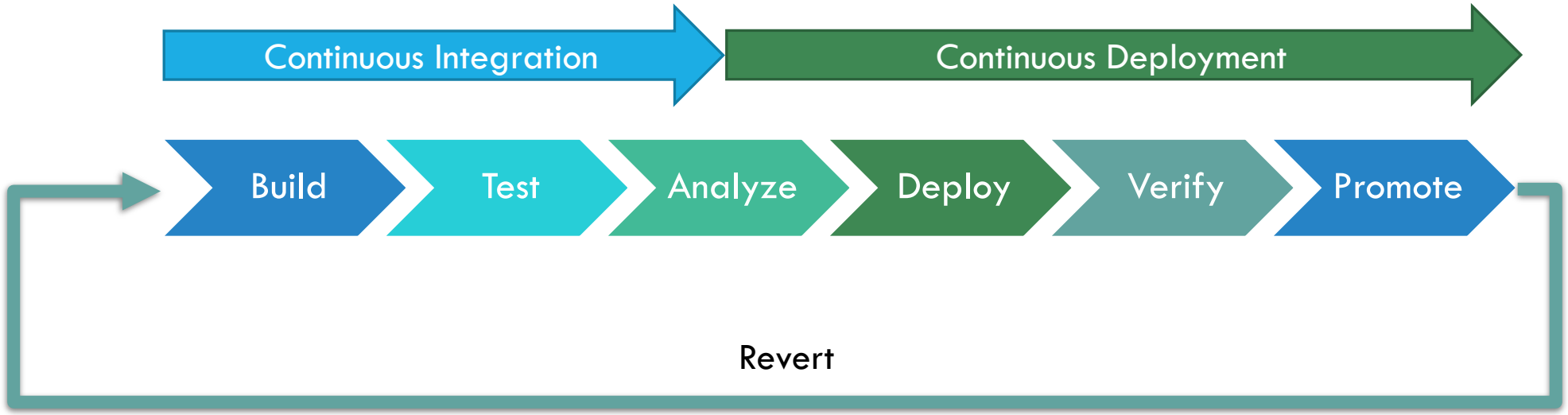


Continuous Delivery Pipeline

CONTINUOUS DELIVERY = CI + CD

Continues delivery is the mindset for delivery of products to end-users in short cycles. Continuous delivery is an engineering paradigm and independent of tools. Continuous Delivery enhances the practices of CI + CD through 8 principles [1].

1. Repeatable Reliable Process
2. Automate Everything
3. Version Control Everything
4. Bring the Pain Forward
5. Build-in Quality
6. "Done" Means Released
7. Everyone is Responsible
8. Continuous Improvement



Continuous Delivery Pipeline

[1] Udacity Cloud DevOps Nanodegree program

BUSINESS VALUE OF CONTINUOUS DELIVERY (CD)

Increase Revenue

- ☑Release value-generating features faster
- ☑Reduce time-to-market
- ☑Innovate more frequently

Reduce Costs

- ☑Reduce unplanned time spent on resolving issues
- ☑Optimize infrastructure utilization

Protect Revenue

- ☑Reduce downtime from deployment crashes/bugs
- ☑Quick undo – revert to prior working state

Avoid Costs

- ☑Reduce escaped defects/bugs in production
- ☑Prevent security holes
- ☑Faster deployments