Carnegie Mellon University

Copyright 2015 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

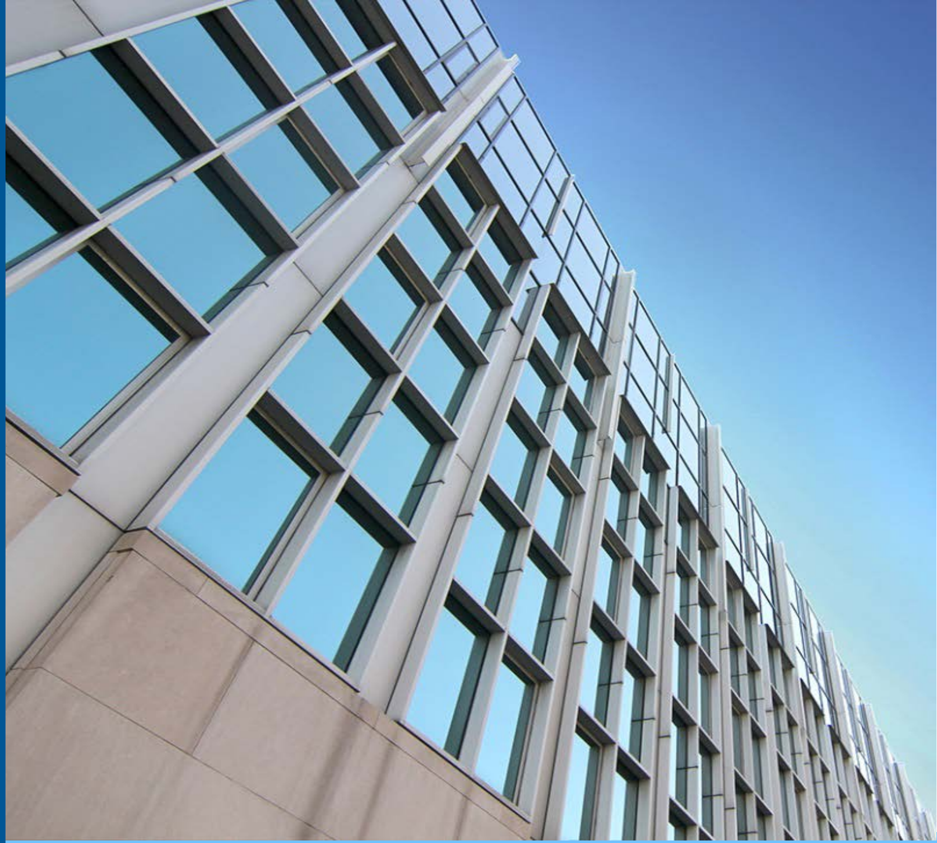
NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

DM-0002231

What DevOps is Not!

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Hasan Yasar
C. Aaron Cois

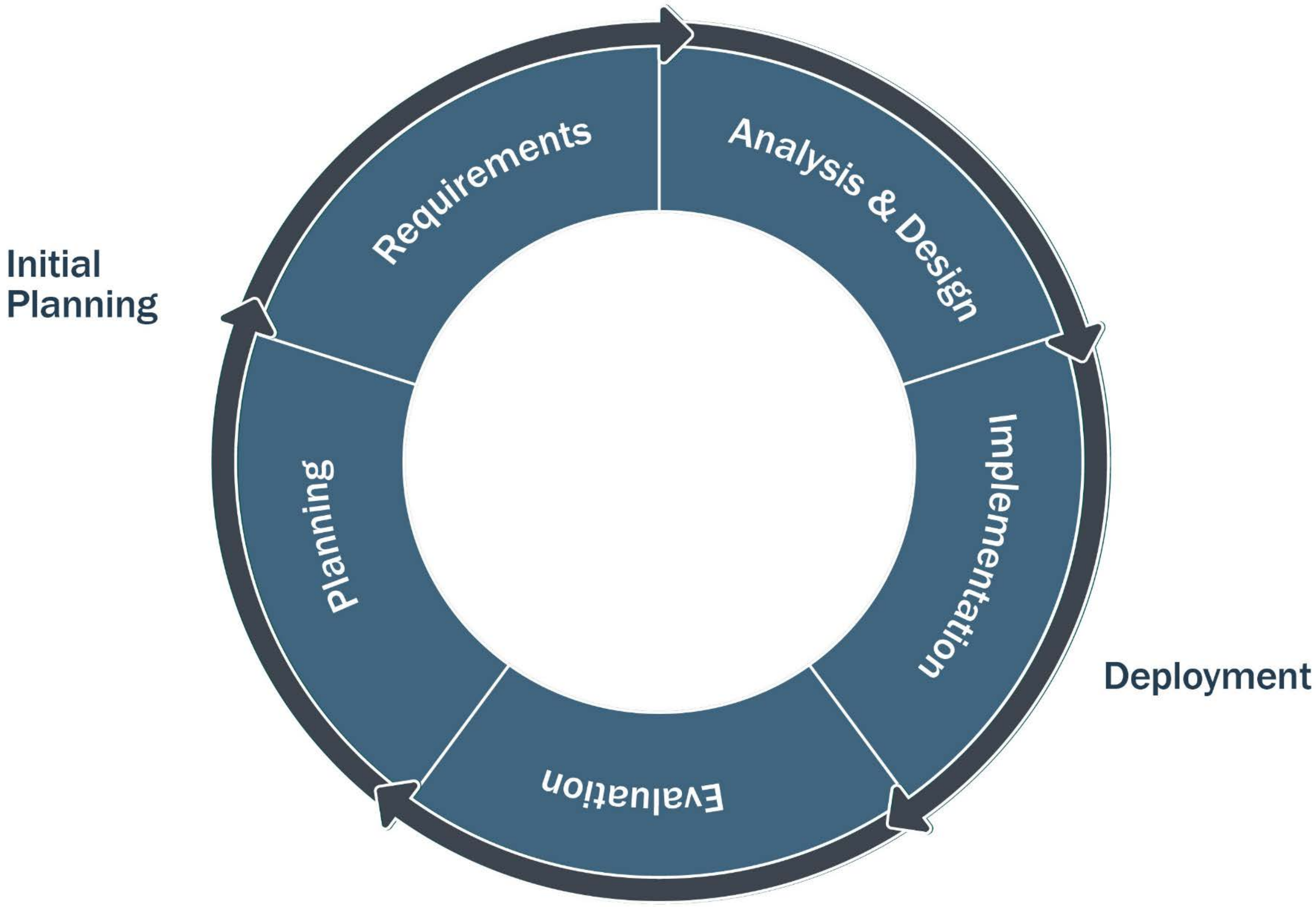


DevOps is not
A FAD

Organizations adopting DevOps are deploying code **30x more frequently** with **50% fewer failures.**

DevOps is not
NEW

DevOps is Agile



DevOps is an Extension of Agile Thinking

Agile

Embrace constant change

Embed Customer in team to internalize expertise on requirements and domain

DevOps

Embrace constant testing, delivery

Embed Operations in team to internalize expertise on deployment and maintenance

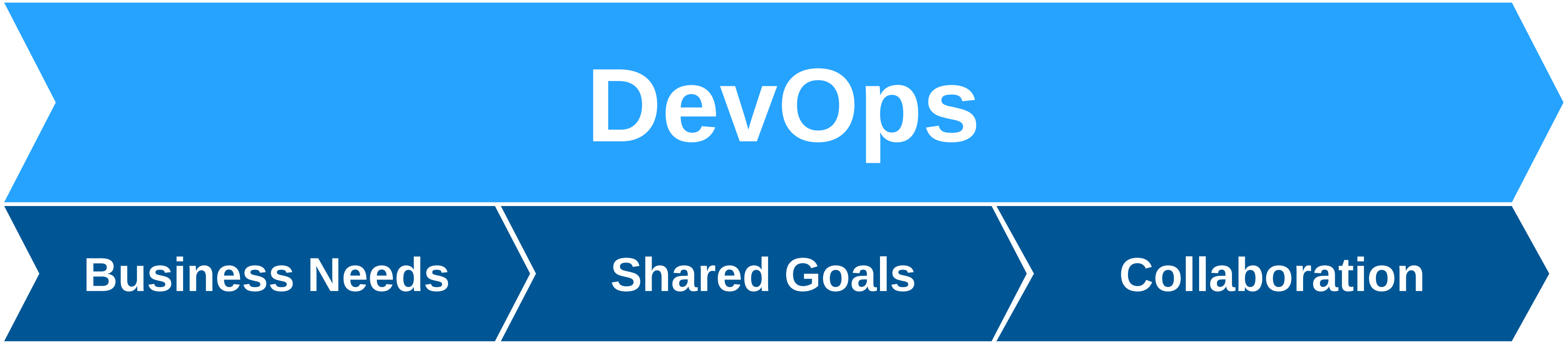
Polling Question 2

Would you like more information about the relationship between Agile software development and DevOps?

1. Yes
2. No

DevOps is not

TOOLING

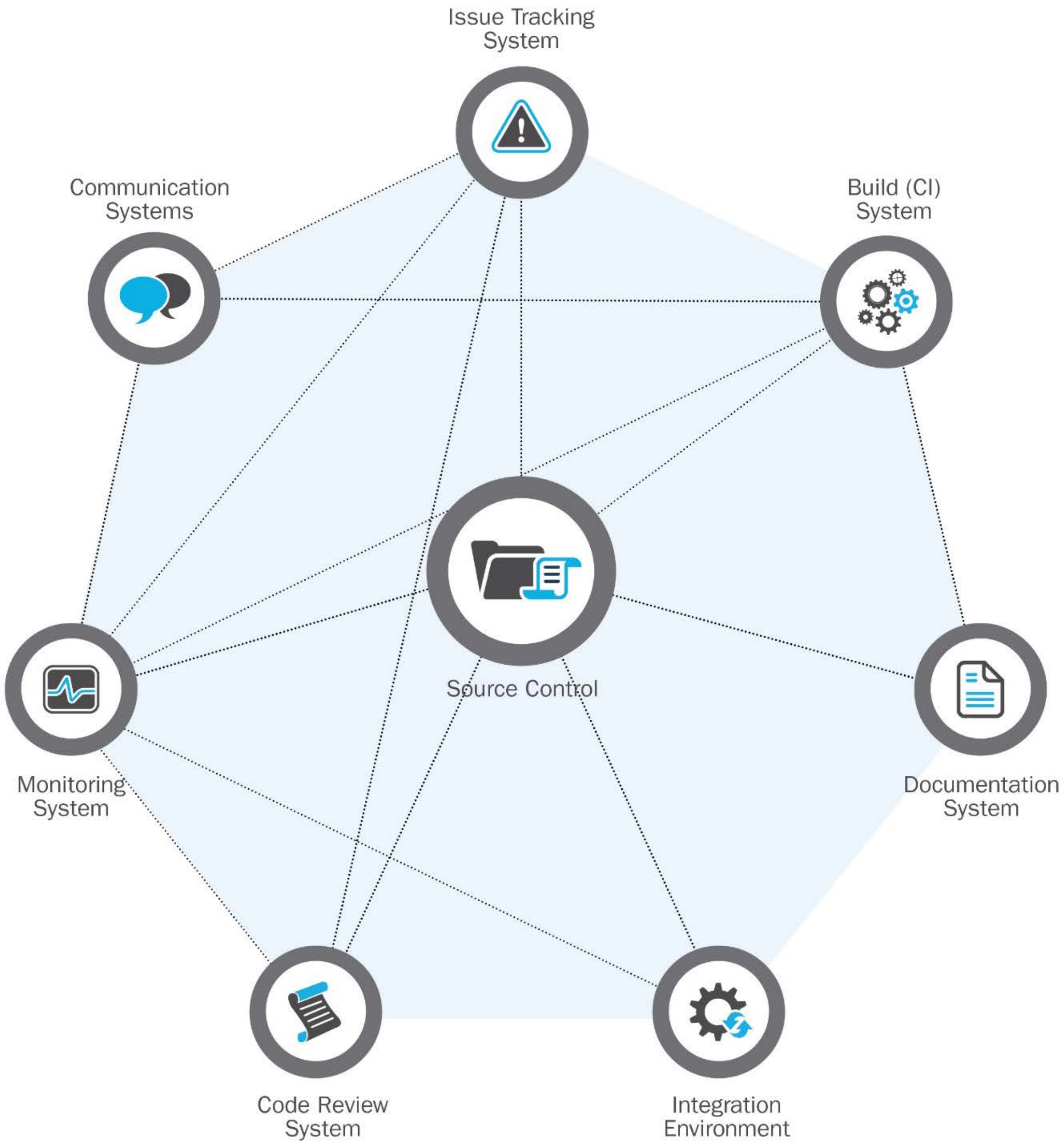


DevOps

Business Needs

Shared Goals

Collaboration



Many tools can help you achieve your DevOps goals…

But **don't get distracted!**

Integration and communication, even among tools, is key.

Redundant tooling is worse than no tooling at all.

Polling Question 3

Would you like more information on shared goals in DevOps?

1. Yes
2. No

DevOps is not

A PRODUCT

DevOps is About Culture and Quality

Early involvement of experts

Ops = experts in maintainability and deployability

Complete engagement

Don’t bring Ops Engineers in as consultants – make them first-class team members with same success criteria as devs

Break down organizational silos

Enable and require constant communication

Without a Collaborative Culture, You Don’t Have DevOps

Ask yourself:

Do your Devs know **exactly** what **actual** production looks like?

Does Ops know how Devs package a build?
Is it **consistent**?

Can both Dev and Ops collaborate on server configuration and apply it automatically to both **development and production environments**?

Without a Collaborative Culture, You Don’t Have DevOps

Ask yourself:

Do business analysts **know the cost** of feature addition or modification?

Can project managers measure project status **at any point in time**?

Can the customer measure project status **at any point in time**?

DevOps is not

ONE SIZE FITS ALL

DevOps Requires Customization to Meet Your Unique Needs

Example: How should I configure my CI server?

Want 90% test coverage?

Fail the build if code base is < 90% covered

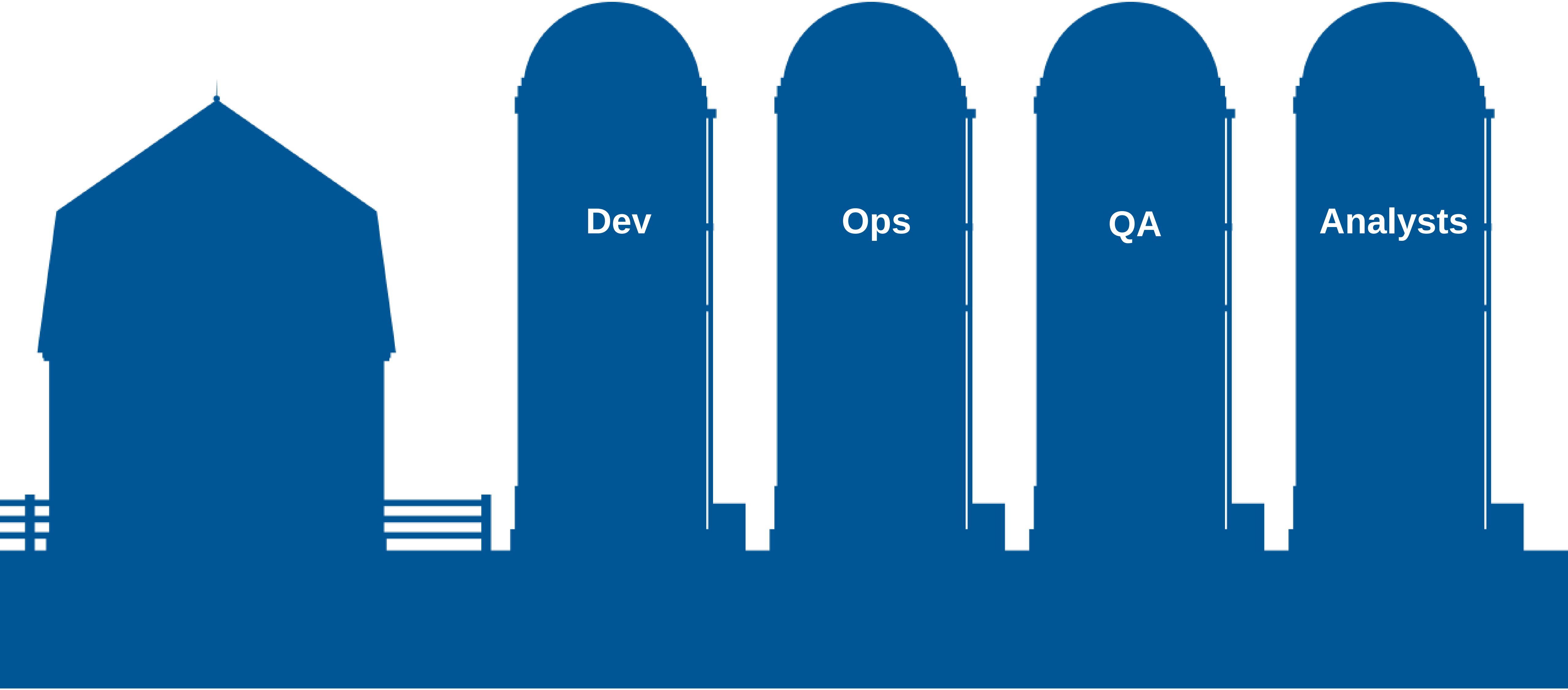
Want all DB queries < 2sec?

Test them, and fail the build otherwise

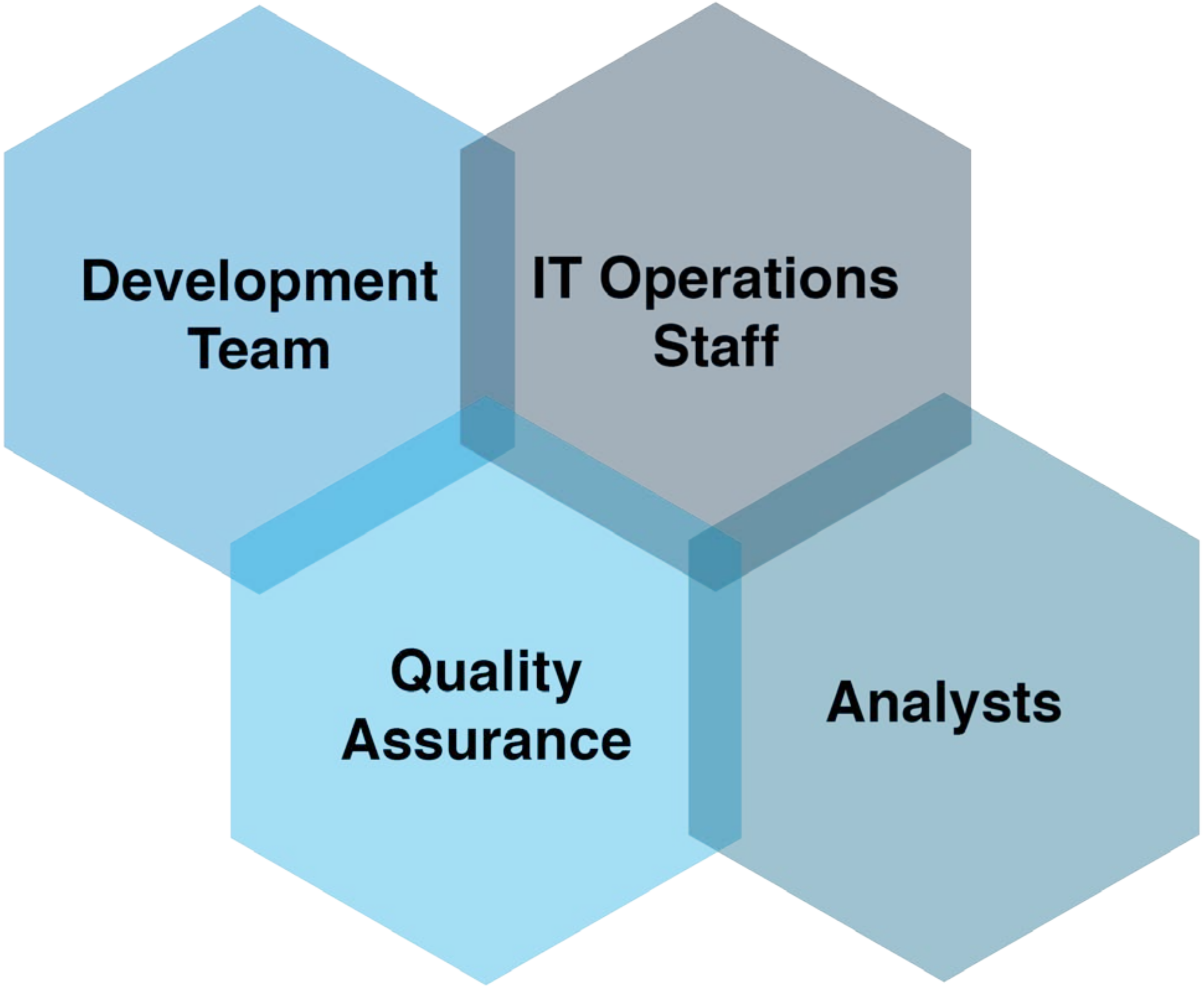
What does **quality** mean to your organization?

DevOps is not

SILOED


Dev
Ops
QA
Analysts

DevOps Breaks Down Silos



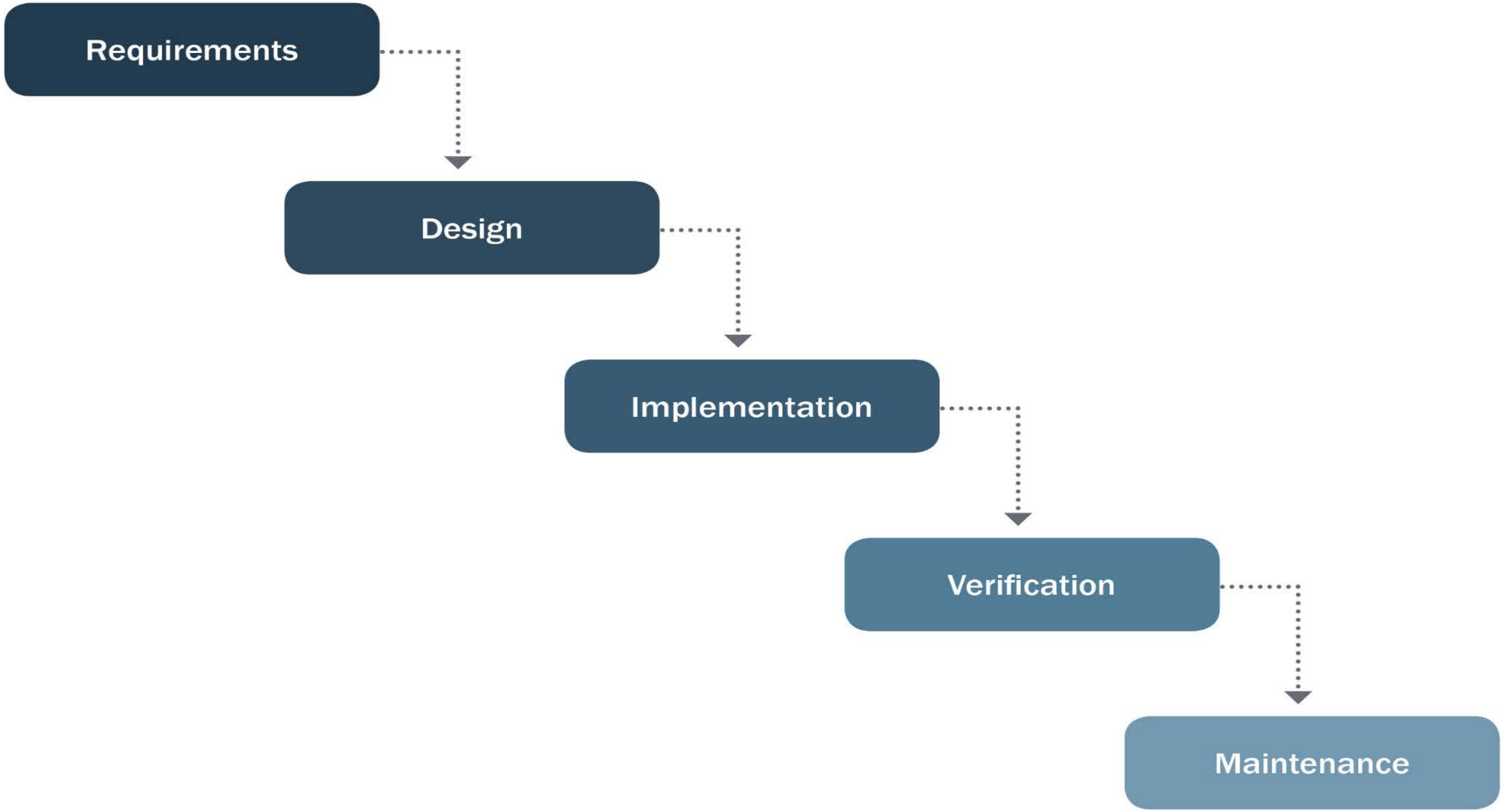
Development Team

IT Operations Staff

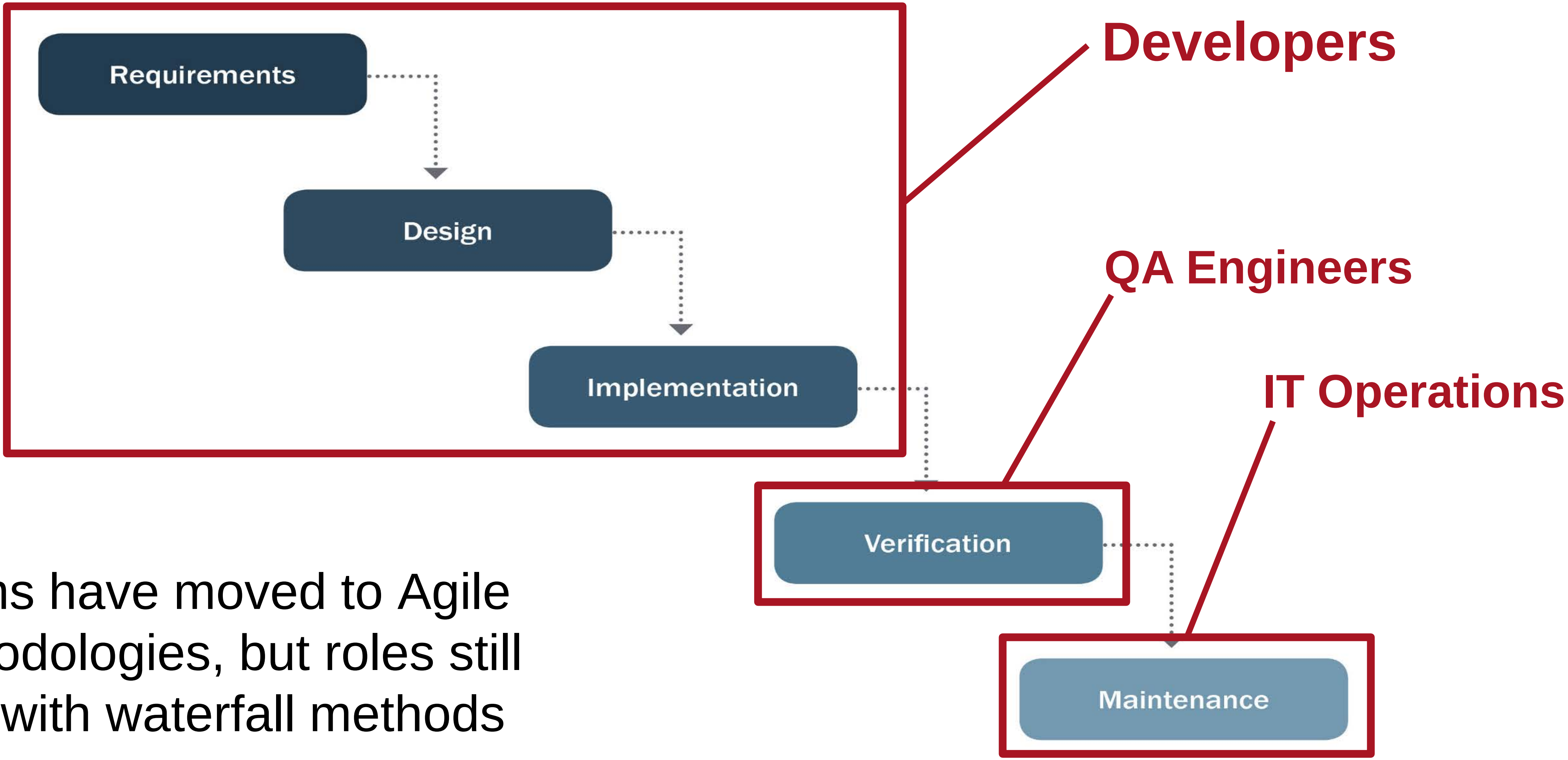
Quality Assurance

Analysts

Waterfall


Requirements
Design
Implementation
Verification
Maintenance

Organizational Silos Are Waterfall, Not Agile



Teams have moved to Agile methodologies, but roles still align with waterfall methods

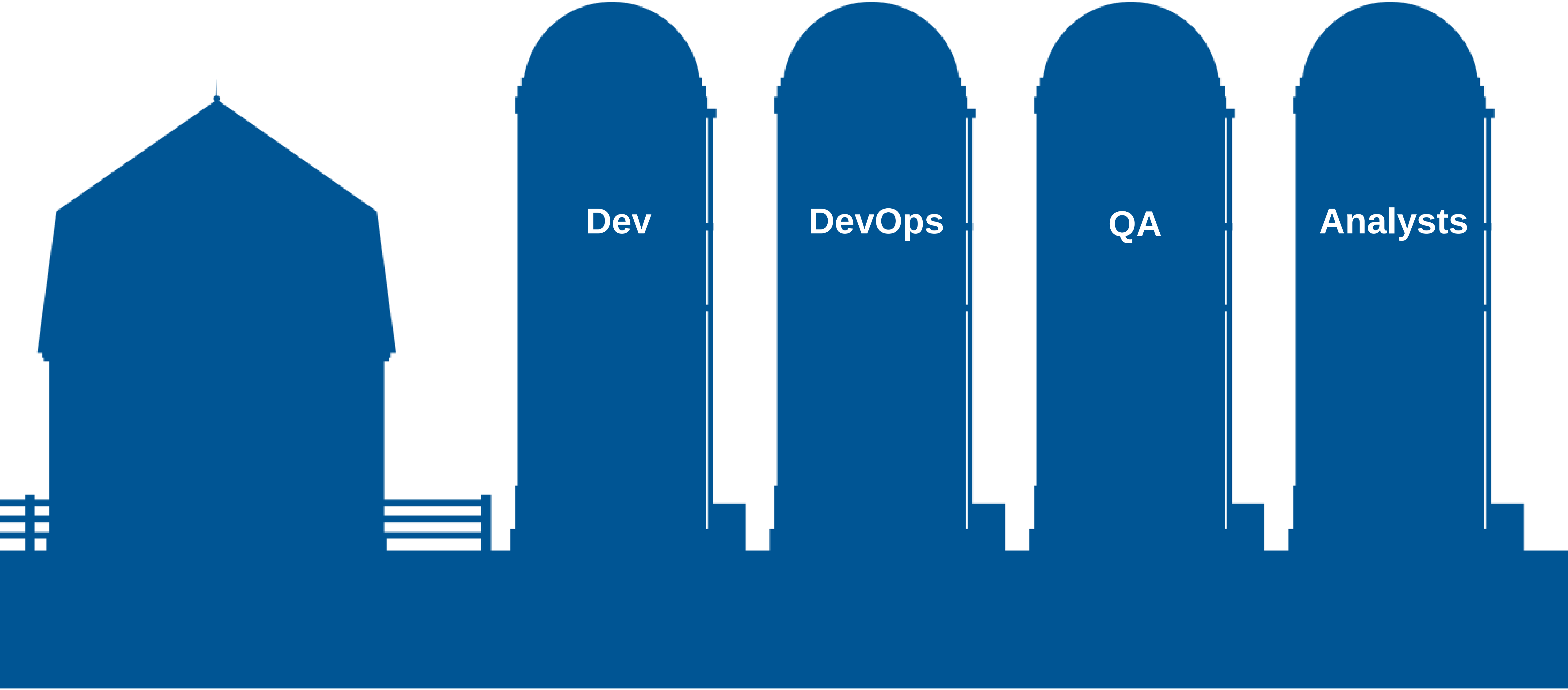
DevOps is not

AN ORGANIZATIONAL UNIT

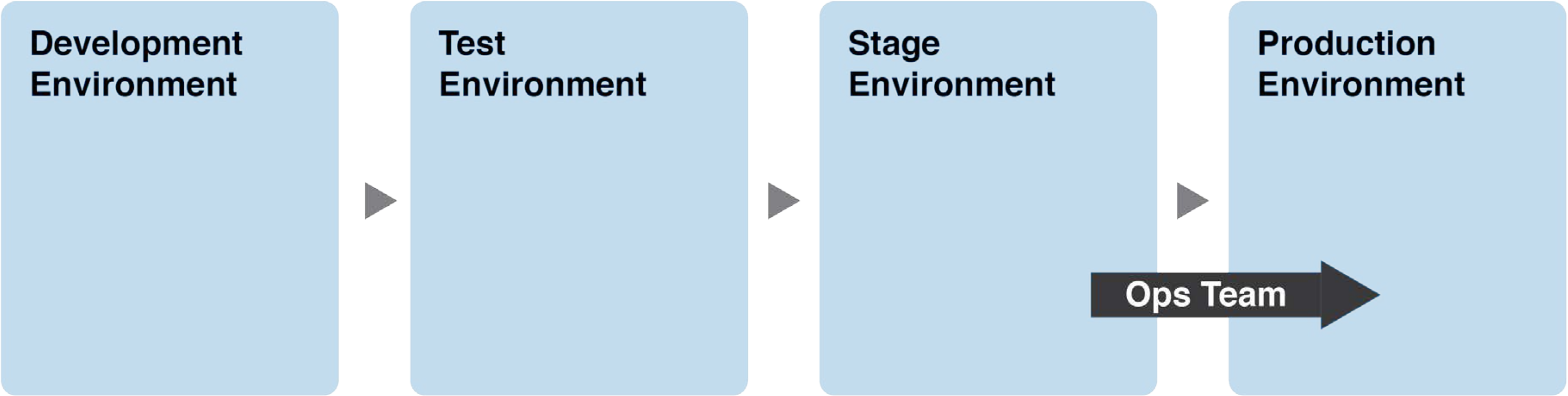
Polling Question 4

Do you think “DevOps Engineer” is a valid role?

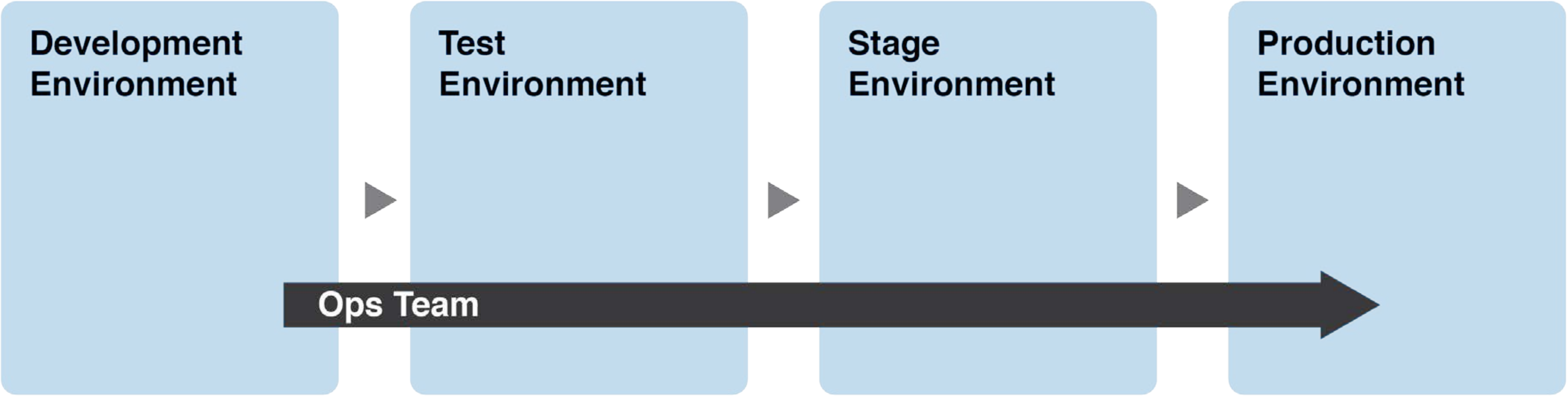
1. Yes
2. No


Dev
DevOps
QA
Analysts

DevOps is Systematically Shifting Left Ops Concerns



DevOps is Systematically Shifting Left Ops Concerns



Development Environment

Test Environment

Stage Environment

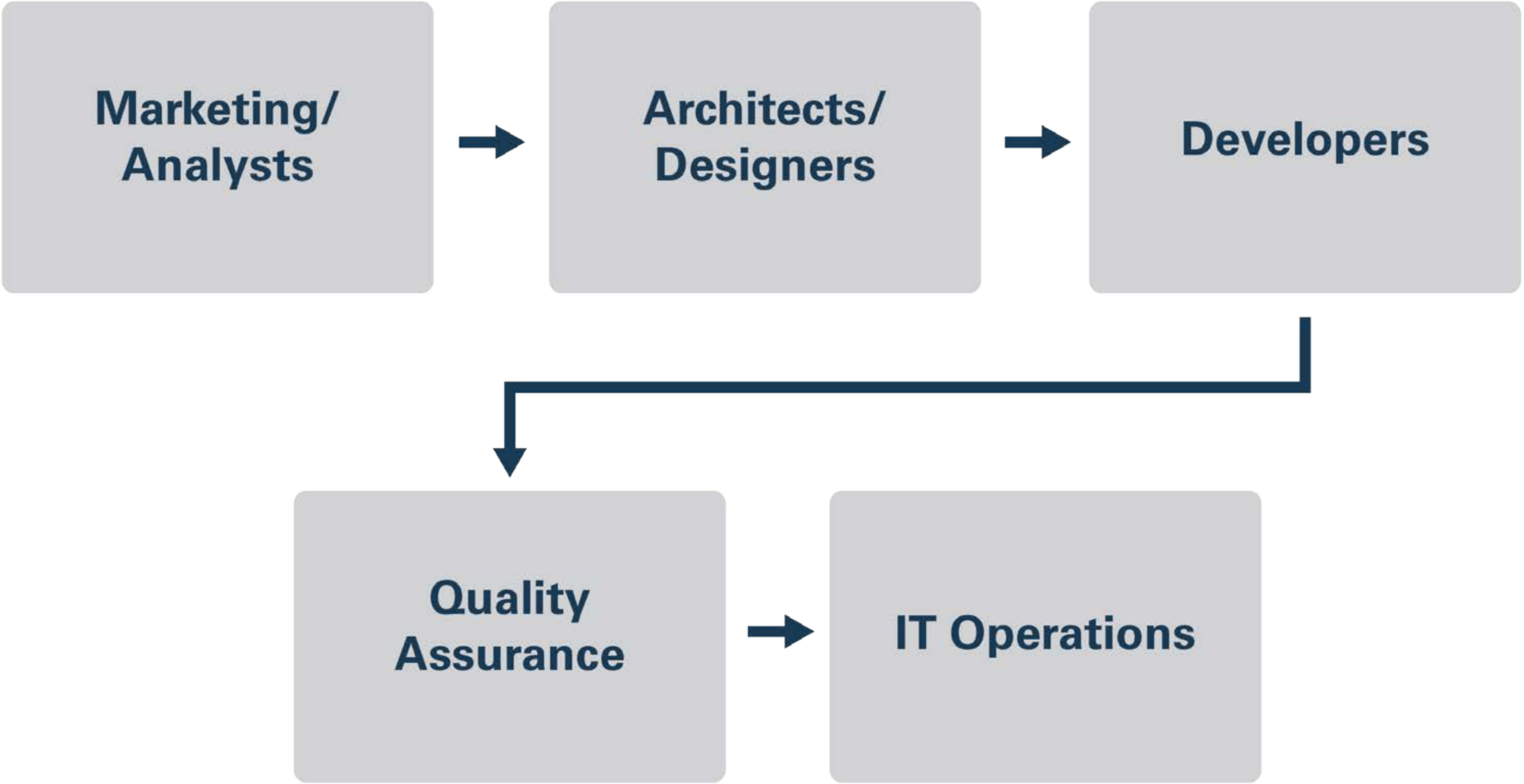
Production Environment

Ops Team

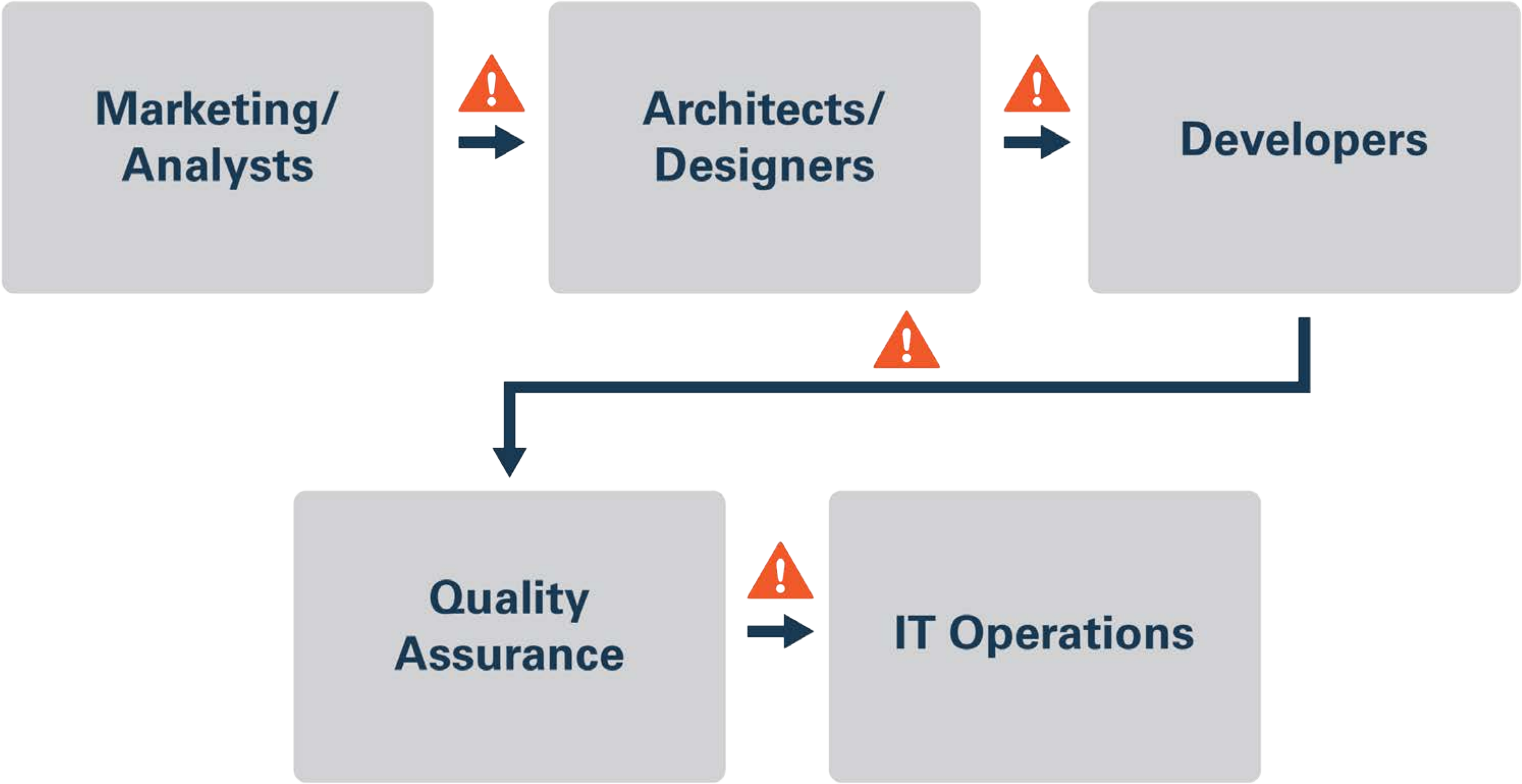
DevOps is not

JUST ABOUT DEV AND OPS

Silos Create Many Transitions Throughout the SDLC



Every Transition of Your System is a Risk



Marketing/ Analysts

Architects/ Designers

Developers

Quality Assurance

IT Operations

Software Projects Are Complex

Scalabilty

Business Constraints

Infrastructure

Deployment

Legal Issues

User Requirements

Networks

Performance

Maintenance

Market Needs

Functional Requirements

Updates

Budgets / Timelines

Programming

Technical Documentation

Testing

Code Review

Data Privacy

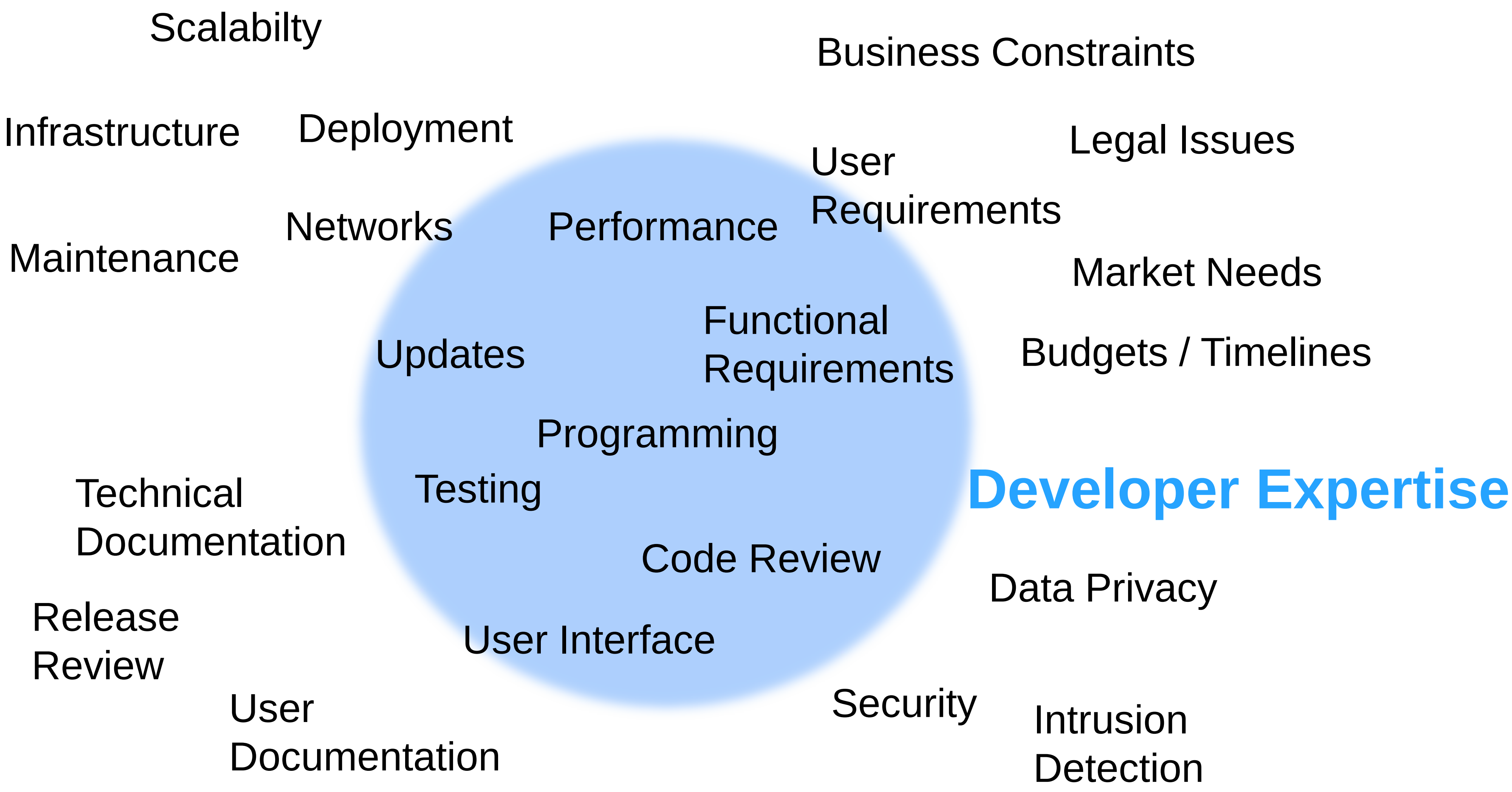
Release Review

User Interface

Security

User Documentation

Intrusion Detection



Scalabilty

Business Constraints

Infrastructure

Deployment

Legal Issues

User Requirements

Networks

Performance

Maintenance

Market Needs

Functional Requirements

Updates

Budgets / Timelines

Programming

Developer Expertise

Technical Documentation

Testing

Code Review

Data Privacy

Release Review

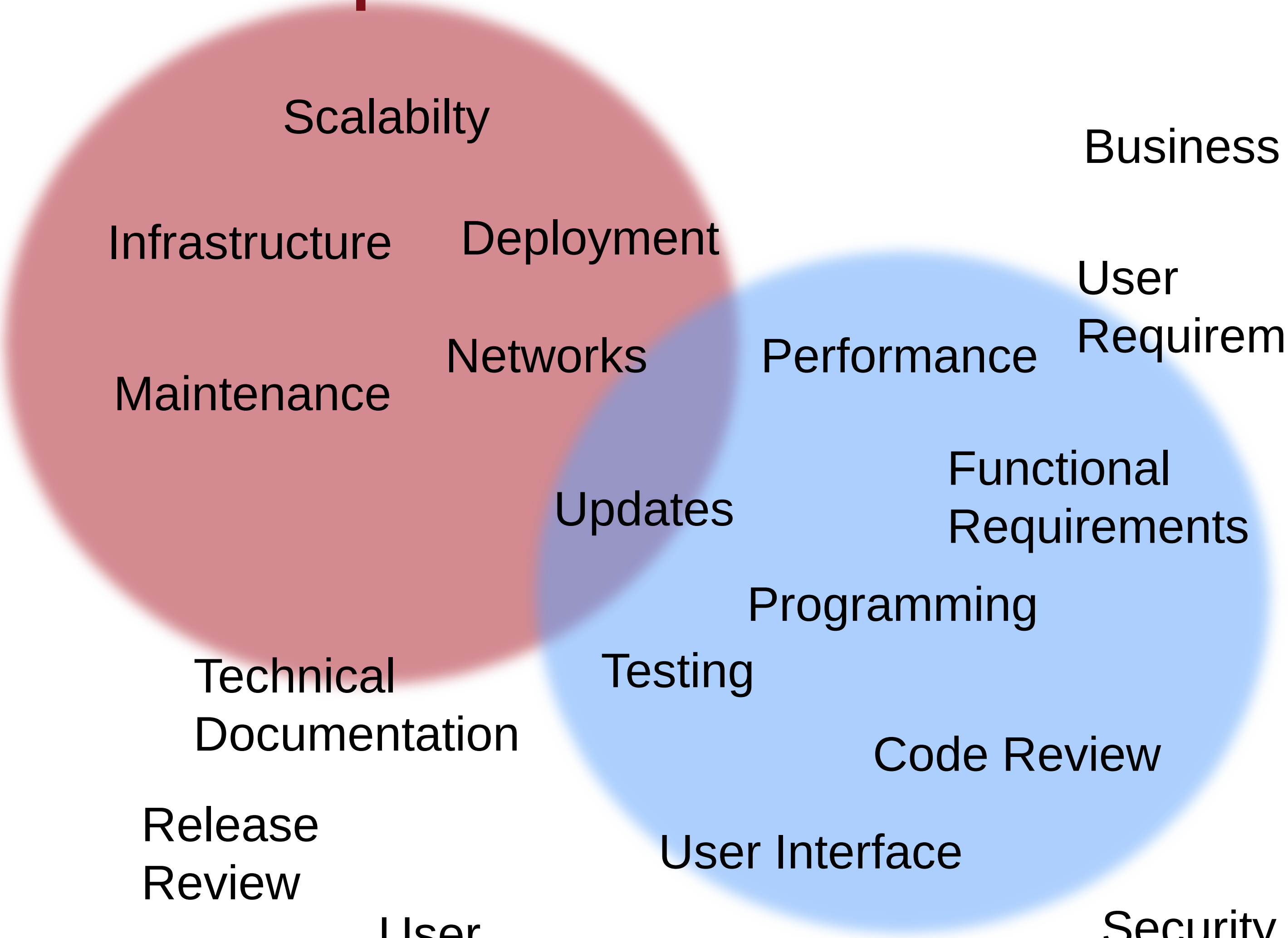
User Interface

Security

User Documentation

Intrusion Detection

IT Operations Expertise



Scalabilty

Infrastructure

Deployment

Networks

Maintenance

Performance

Updates

Functional Requirements

Programming

Testing

Code Review

User Interface

Business Constraints

User Requirements

Legal Issues

Market Needs

Budgets / Timelines

Technical Documentation

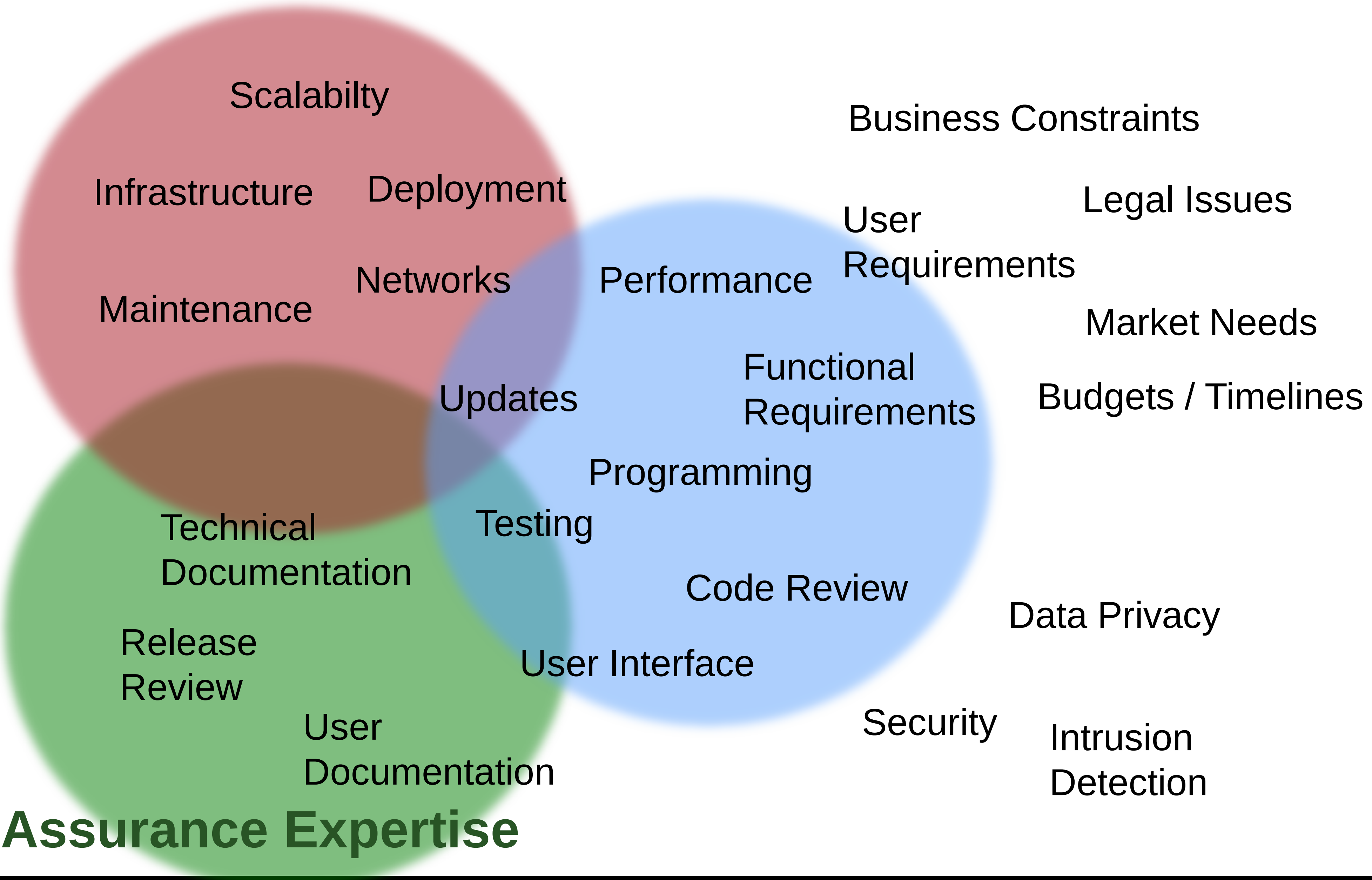
Release Review

User Documentation

Data Privacy

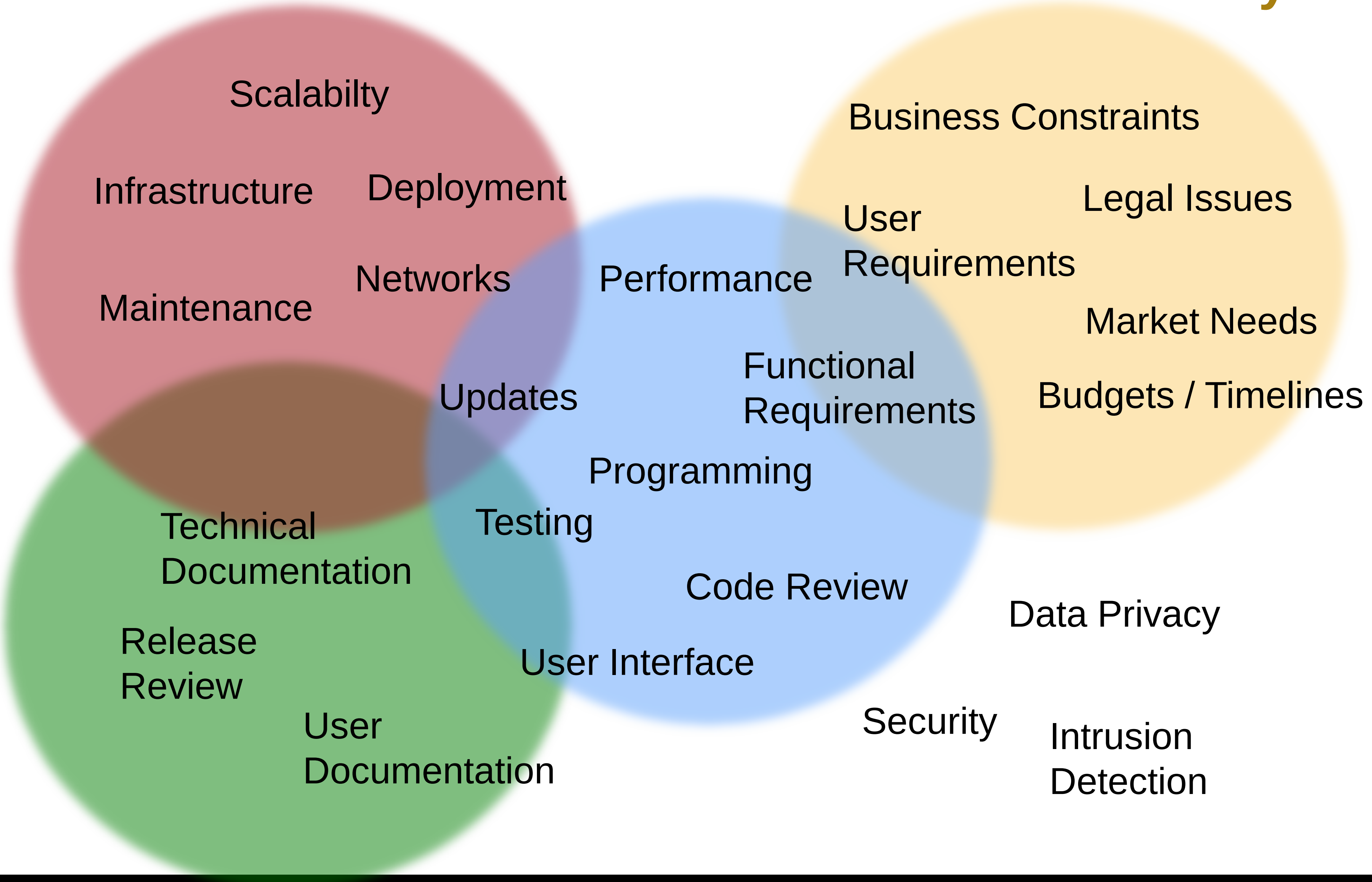
Security

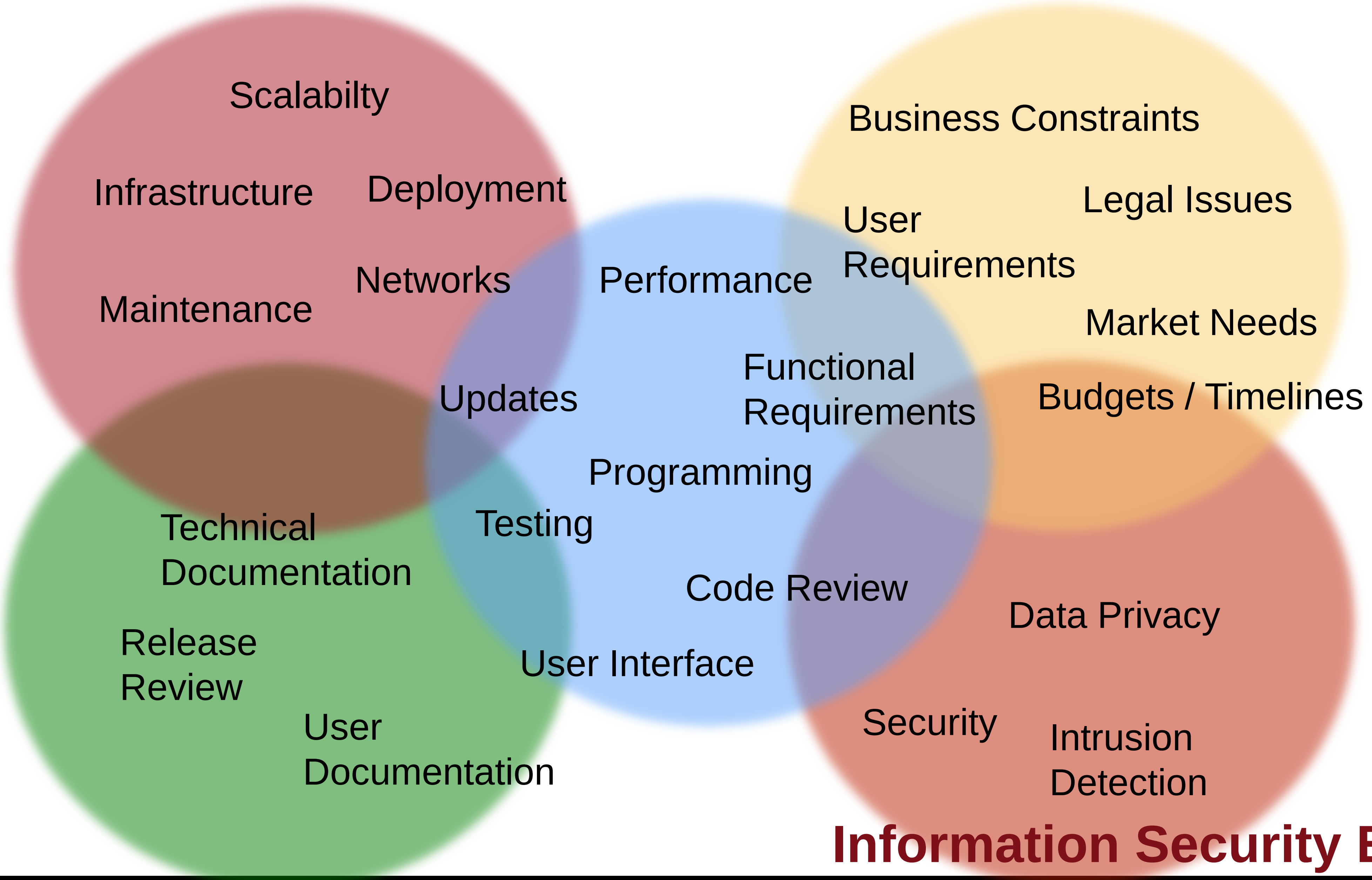
Intrusion Detection



Quality Assurance Expertise

Business Analyst Expertise





Scalabilty

Infrastructure

Deployment

Networks

Maintenance

Business Constraints

User Requirements

Legal Issues

Market Needs

Performance

Functional Requirements

Budgets / Timelines

Updates

Programming

Testing

Technical Documentation

Code Review

Data Privacy

Release Review

User Interface

User Documentation

Security

Intrusion Detection

Information Security Expertise

DevOps is not

REPLACING OPS

Effective Teams Need Dedicated Experts

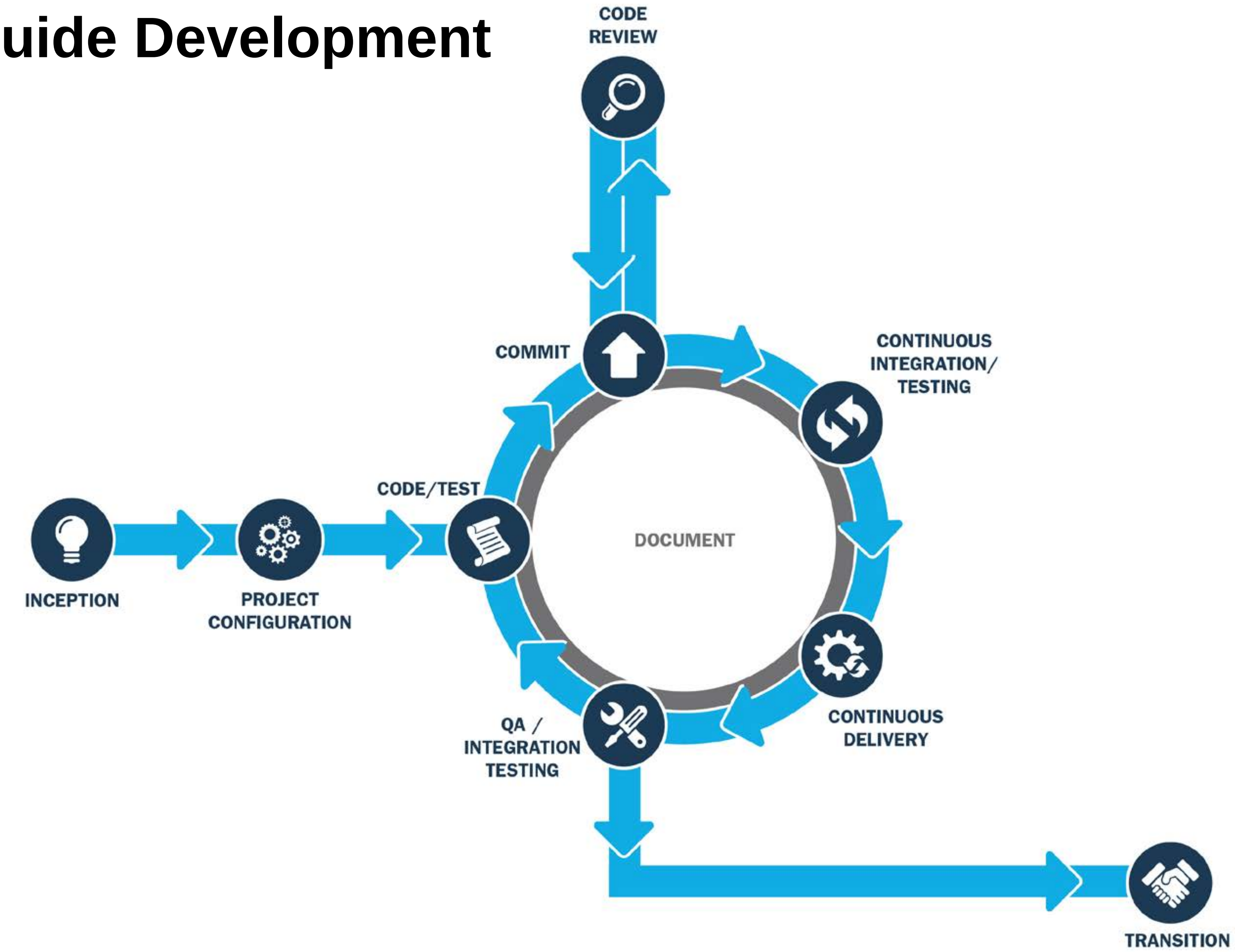
Primary attributes of your system require

dedicated expert team members

E.g. Security, Usability, Deployability

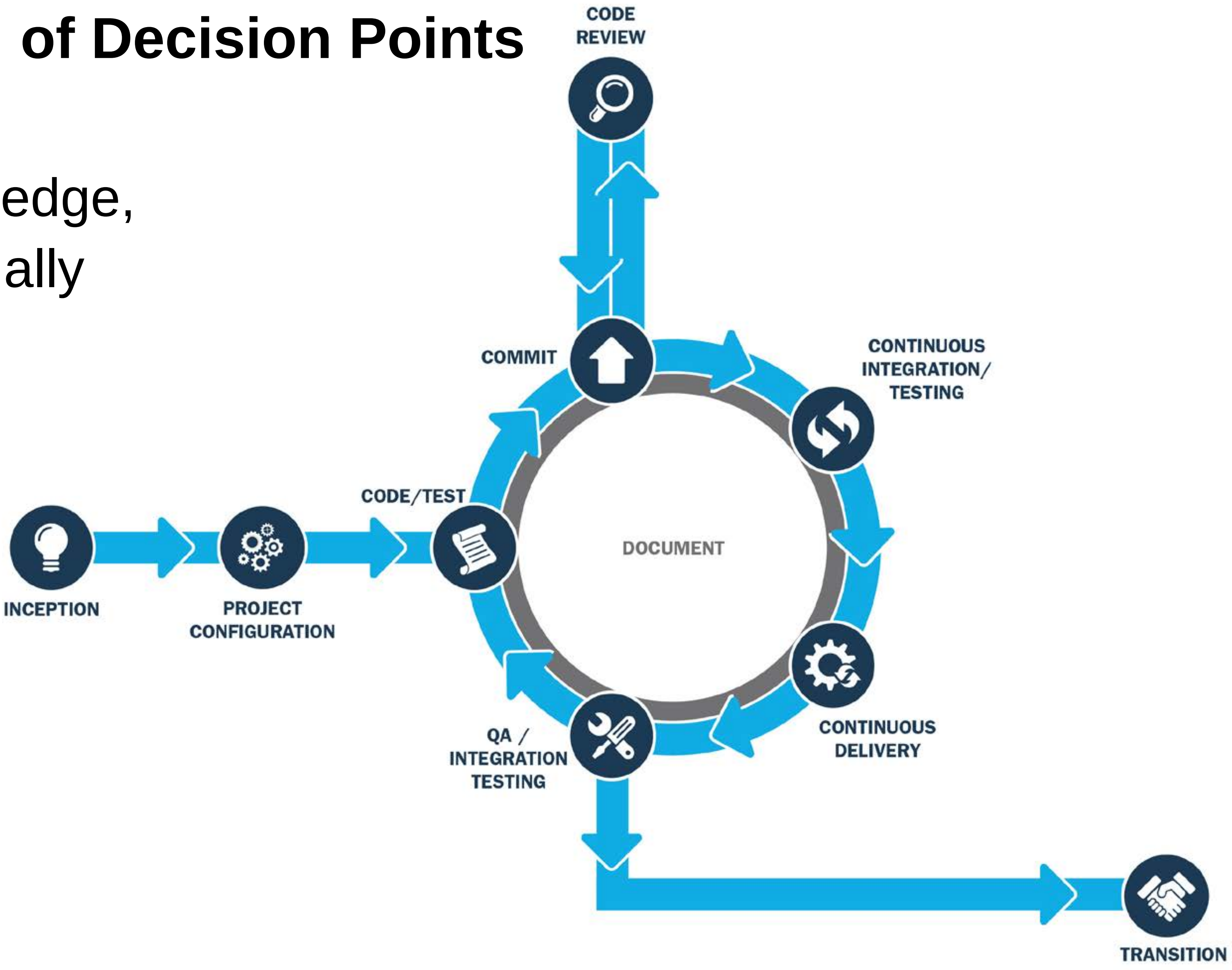
DevOps does **not** mean telling developers to learn / automate operations tasks

Ops Needs to Guide Development



The SDLC is Full of Decision Points

Without Ops knowledge, developers continually make uninformed decisions, causing eventual **risk** or **inefficiency**



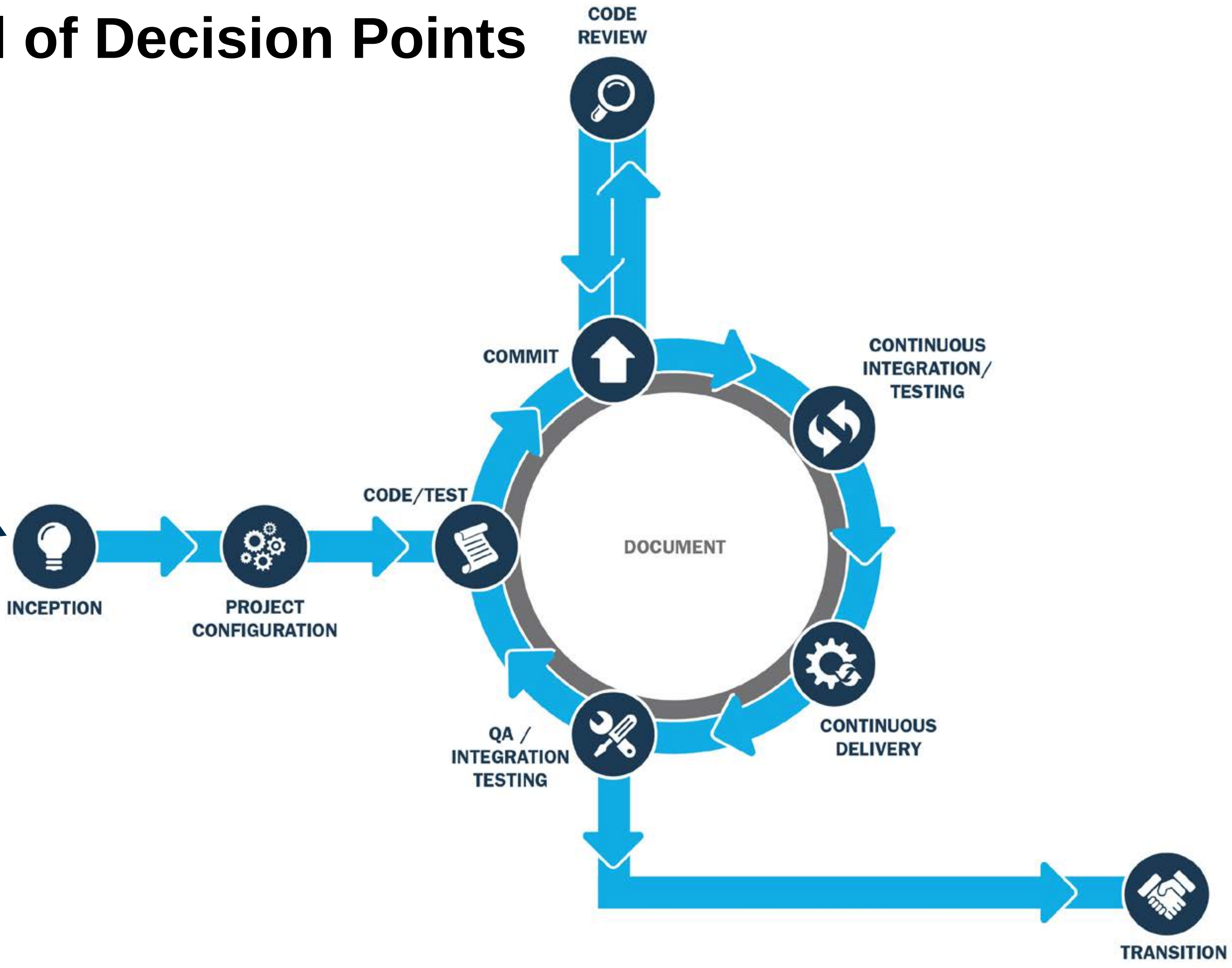
The SDLC is Full of Decision Points

How many users?

Payment model?

Who is the Target Market?

Which regions?



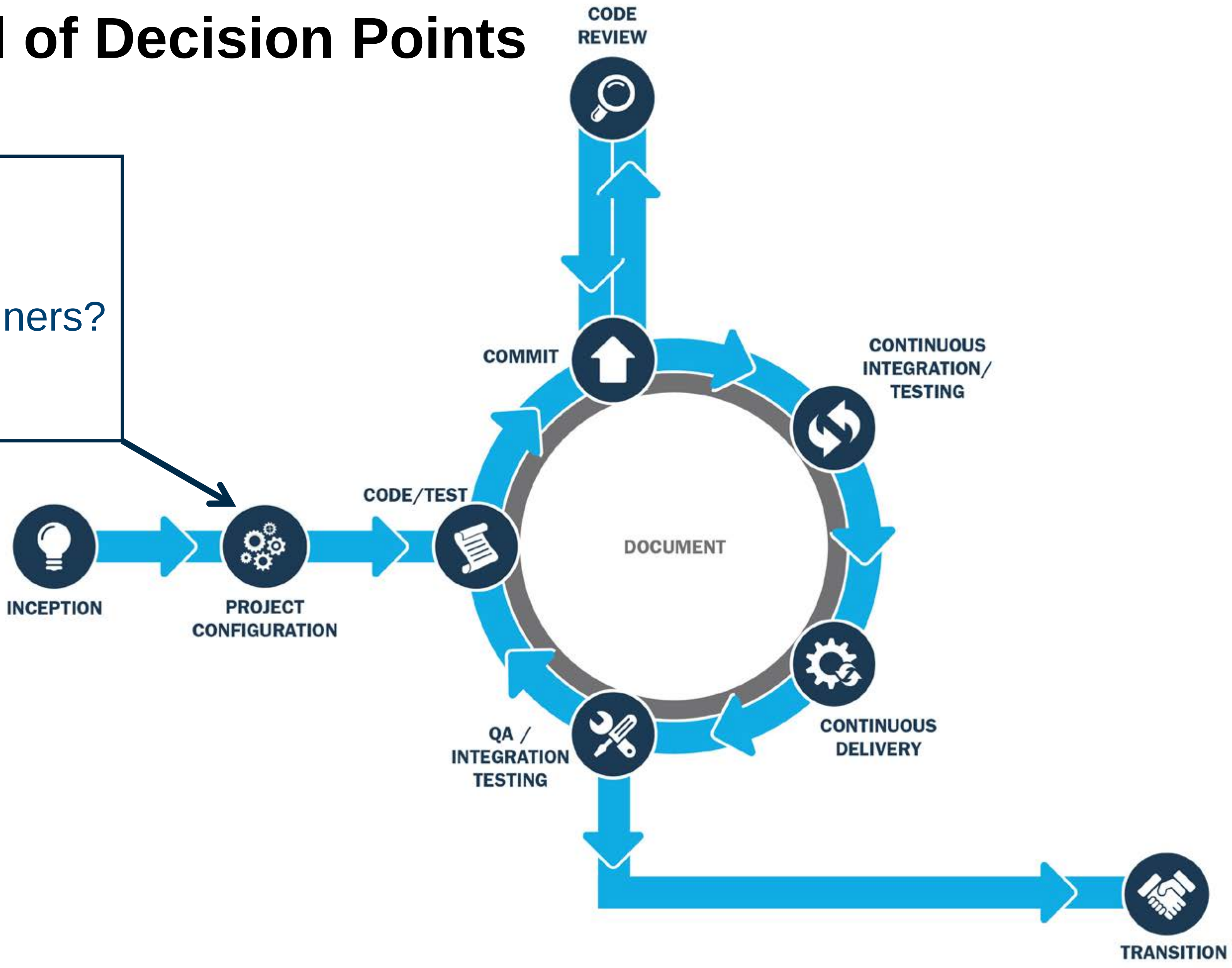
The SDLC is Full of Decision Points

What OS?

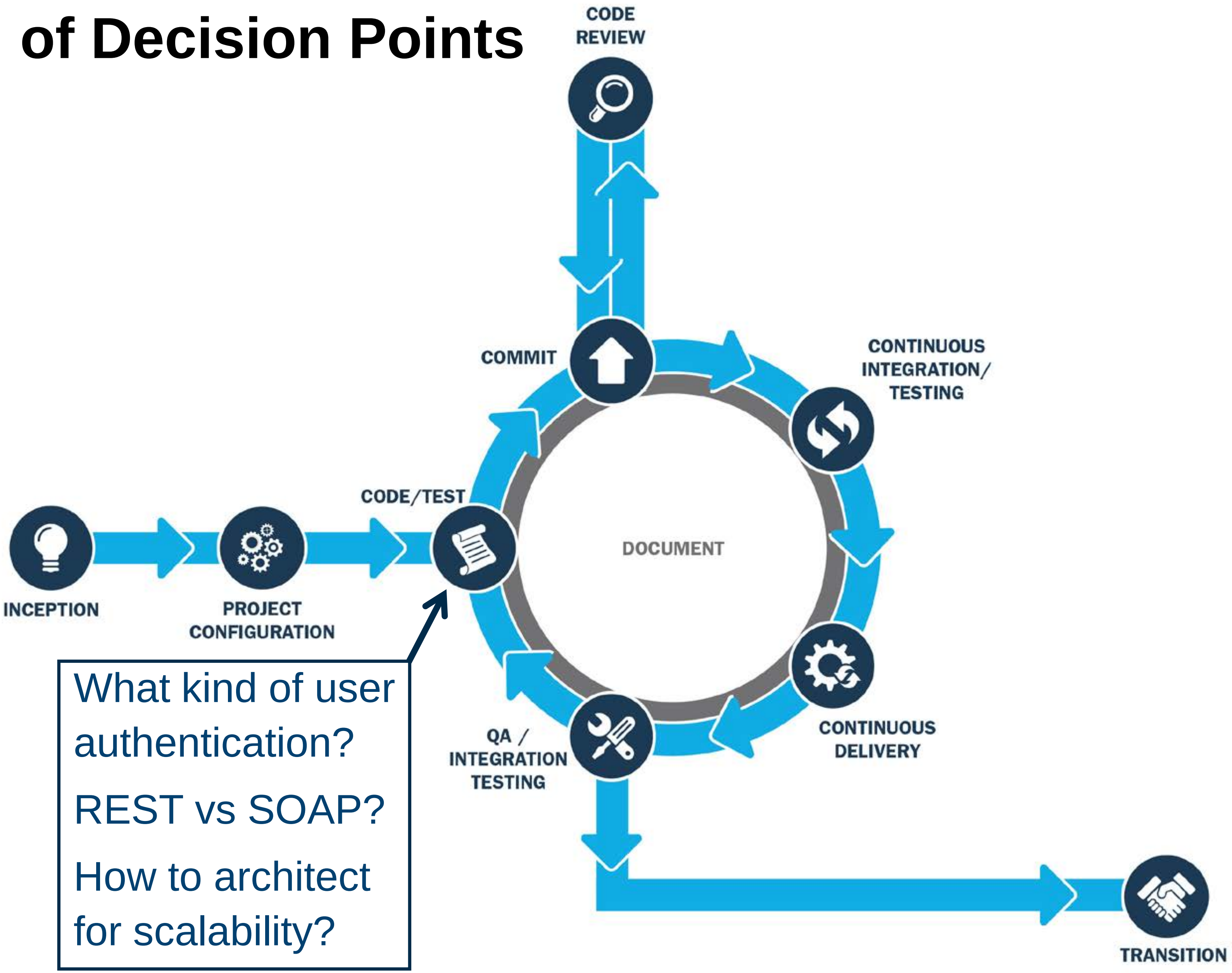
What ports?

VMs or containers?

Entry points?



The SDLC is Full of Decision Points

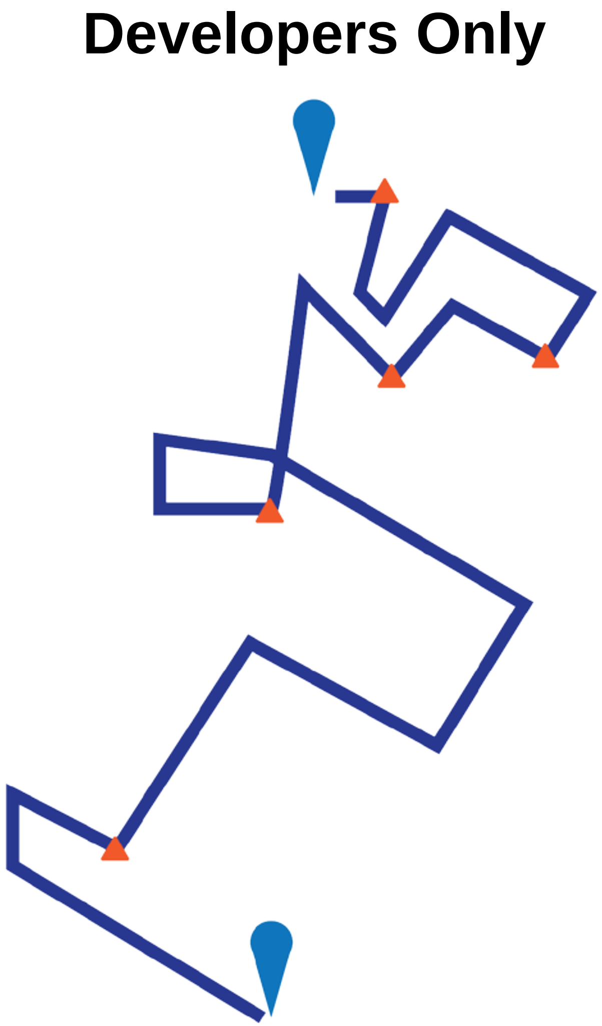


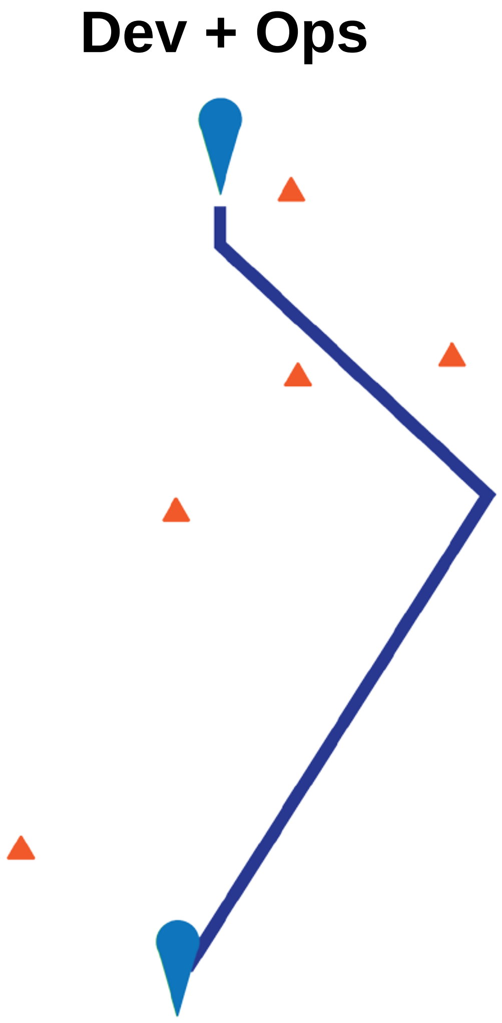
What kind of user authentication?

REST vs SOAP?

How to architect for scalability?

Uninformed Decisions Lead Your Project Down Suboptimal Paths





DevOps is not

CONTINUOUS DELIVERY

Polling Question 5

Does your organization hope to achieve continuous delivery?

1. Yes
2. No

Continuous Deployment

Changes are deployed ASAP into production

Continuous Delivery

Changes are deployed immediately into a *production-like environment*, to ensure that they *could* be deployed into production

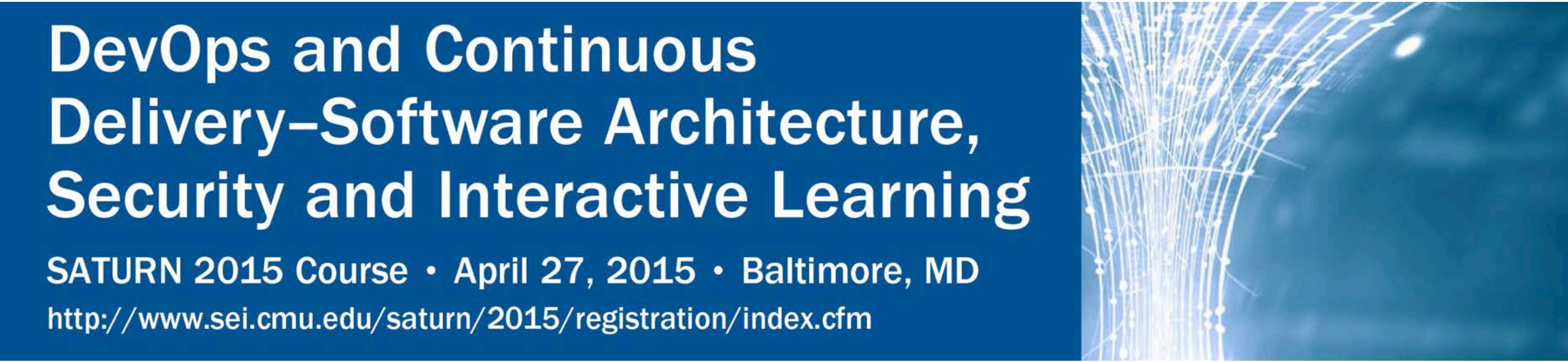
Not Everyone Needs to Achieve Continuous Deployment

Your DevOps goals should be designed around

business needs

Do frequent deployments give you a competitive edge?

If not, what does?



DevOps and Continuous Delivery–Software Architecture, Security and Interactive Learning

SATURN 2015 Course • April 27, 2015 • Baltimore, MD

http://www.sei.cmu.edu/saturn/2015/registration/index.cfm