INTRODUCTION TO Azure DEVOPS

Ranjit Karni
VPark Innovations
Ph No: 967-697-6662

History and How it Started

- Before the current evolution of DevOps, there was no direct connection between development and operations.

- Developers were responsible for building applications and operations had the responsibility of implementing those solutions.

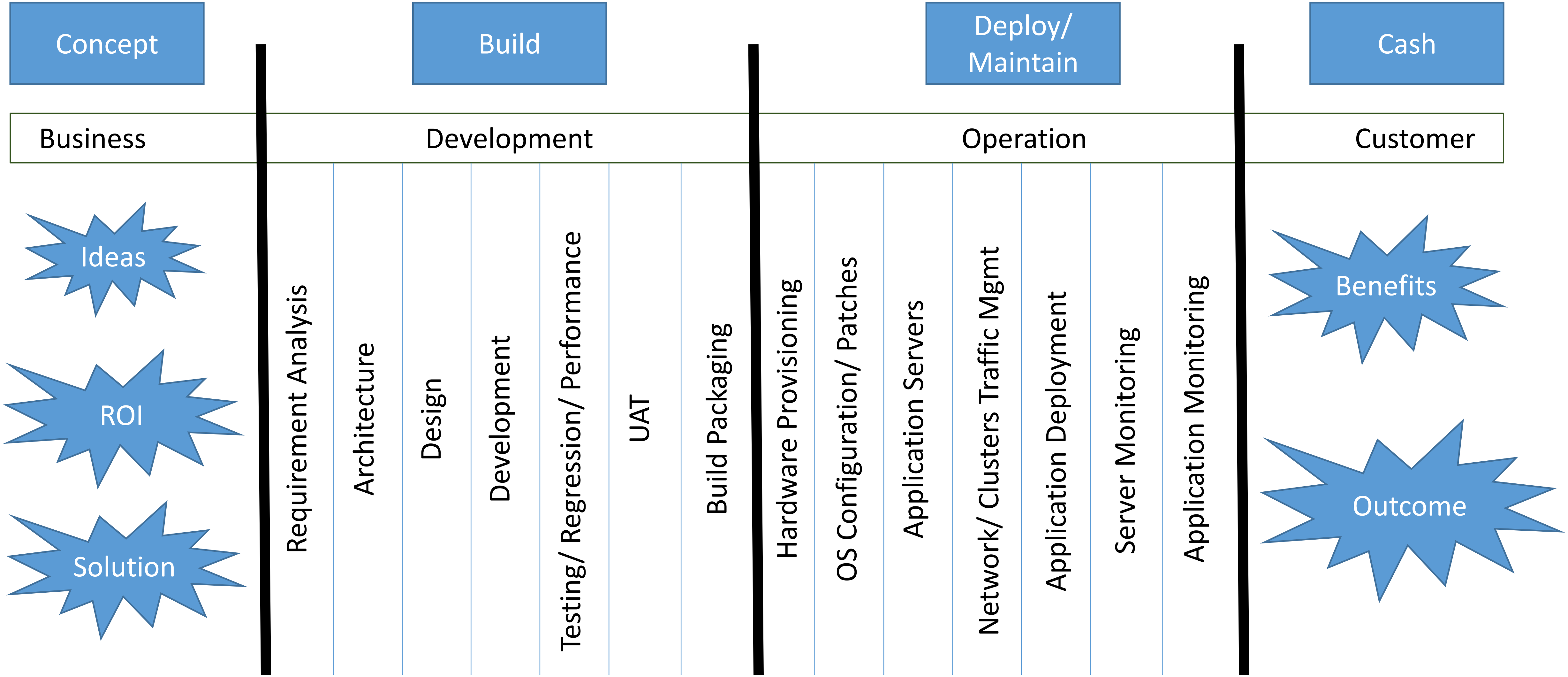
- There was no formal method of communication between the two teams other than when problems arise during deployment.

- Around 2007, a movement started in Europe that was based upon the idea that there needed to be a direct connection between developers and operations, not as a specific position, but as a meld of the two into a connected process flow.

Prerequisites to become a Dev Ops Engineer

- “There’s no formal career track for becoming a DevOps engineer. They are either developers who get interested in deployment and network operations, or sysadmins who have a passion for scripting and coding, and move into the development side where they can improve the planning of test and deployment.
- Either way, these are people who have pushed beyond their defined areas of competence and who have a more holistic view of their technical environments.”

The Flow – Concept To Cash



Concept

Build

Deploy/ Maintain

Cash

Business

Development

Operation

Customer

Ideas

ROI

Solution

Requirement Analysis

Architecture

Design

Development

Testing/ Regression/ Performance

UAT

Build Packaging

Hardware Provisioning

OS Configuration/ Patches

Application Servers

Network/ Clusters Traffic Mgmt

Application Deployment

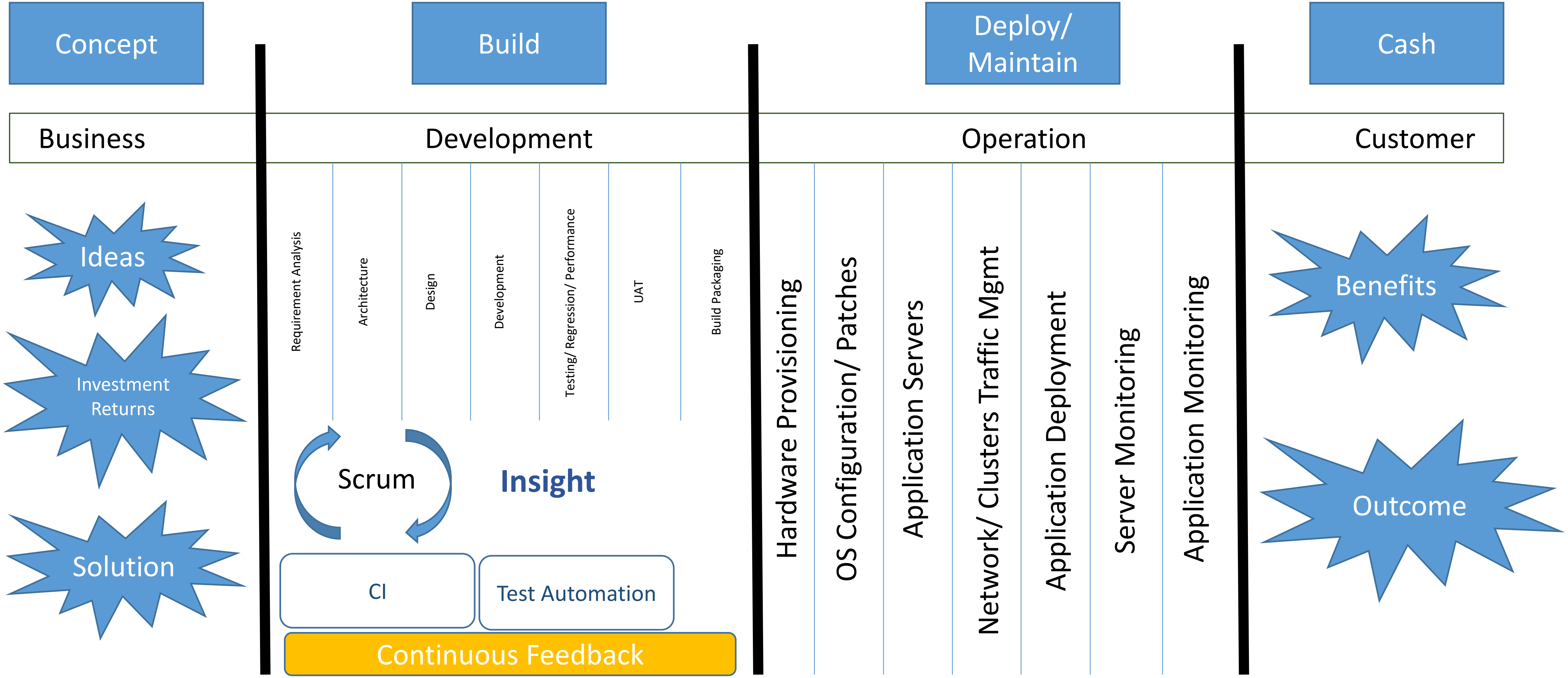
Server Monitoring

Application Monitoring

Benefits

Outcome

The Flow – Concept To Cash



Concept

Business

Ideas

Investment Returns

Solution

Build

Development

Requirement Analysis

Architecture

Design

Development

Testing/ Regression/ Performance

UAT

Build Packaging

Scrum

Insight

CI

Test Automation

Continuous Feedback

Deploy/ Maintain

Operation

Hardware Provisioning

OS Configuration/ Patches

Application Servers

Network/ Clusters Traffic Mgmt

Application Deployment

Server Monitoring

Application Monitoring

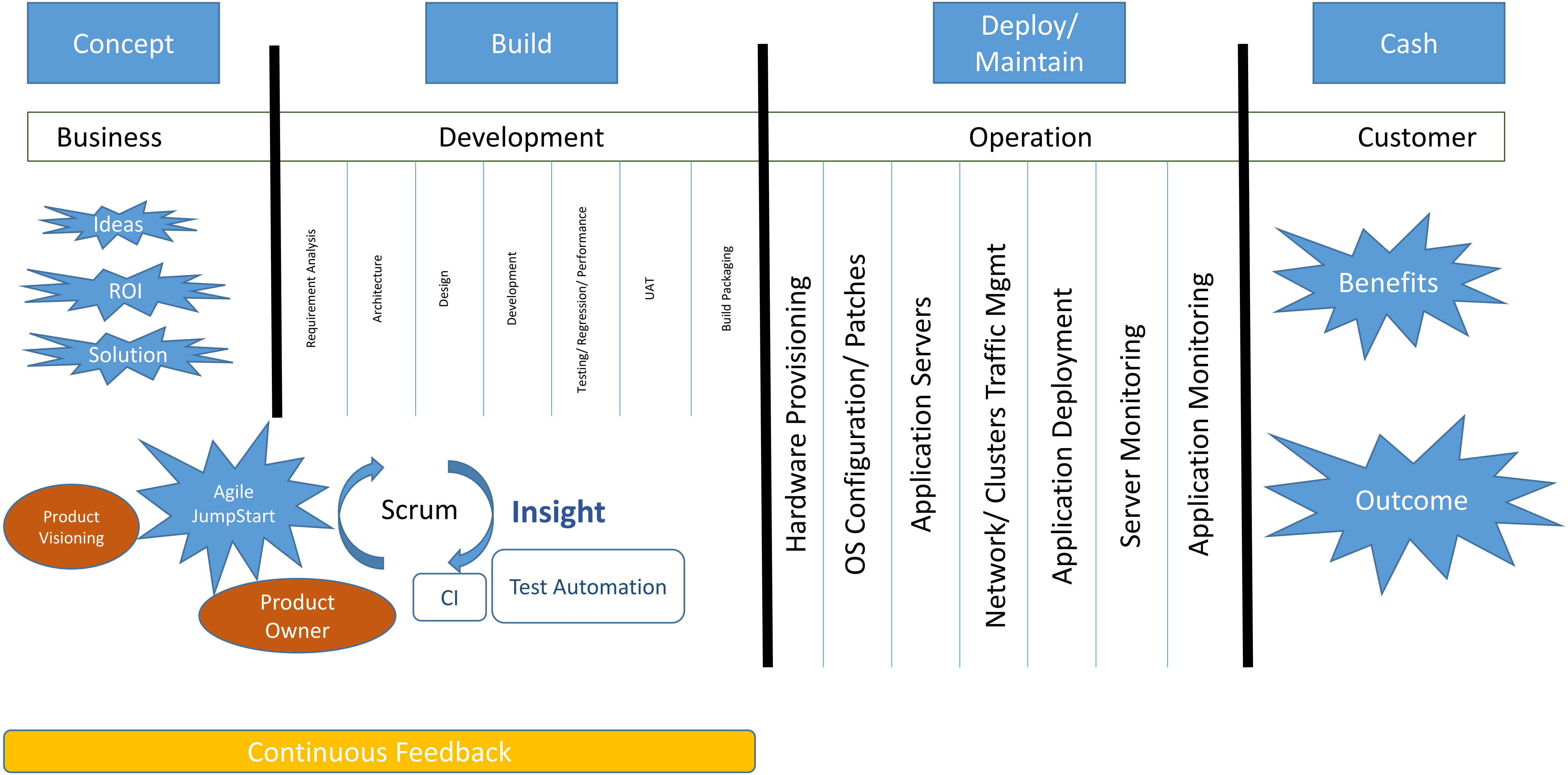
Cash

Customer

Benefits

Outcome

The Flow – Concept To Cash



Concept

Build

Deploy/ Maintain

Cash

Business

Development

Operation

Customer

Ideas

ROI

Solution

Requirement Analysis

Architecture

Design

Development

Testing/ Regression/ Performance

UAT

Build Packaging

Hardware Provisioning

OS Configuration/ Patches

Application Servers

Network/ Clusters Traffic Mgmt

Application Deployment

Server Monitoring

Application Monitoring

Benefits

Outcome

Product Visioning

Agile JumpStart

Product Owner

Scrum

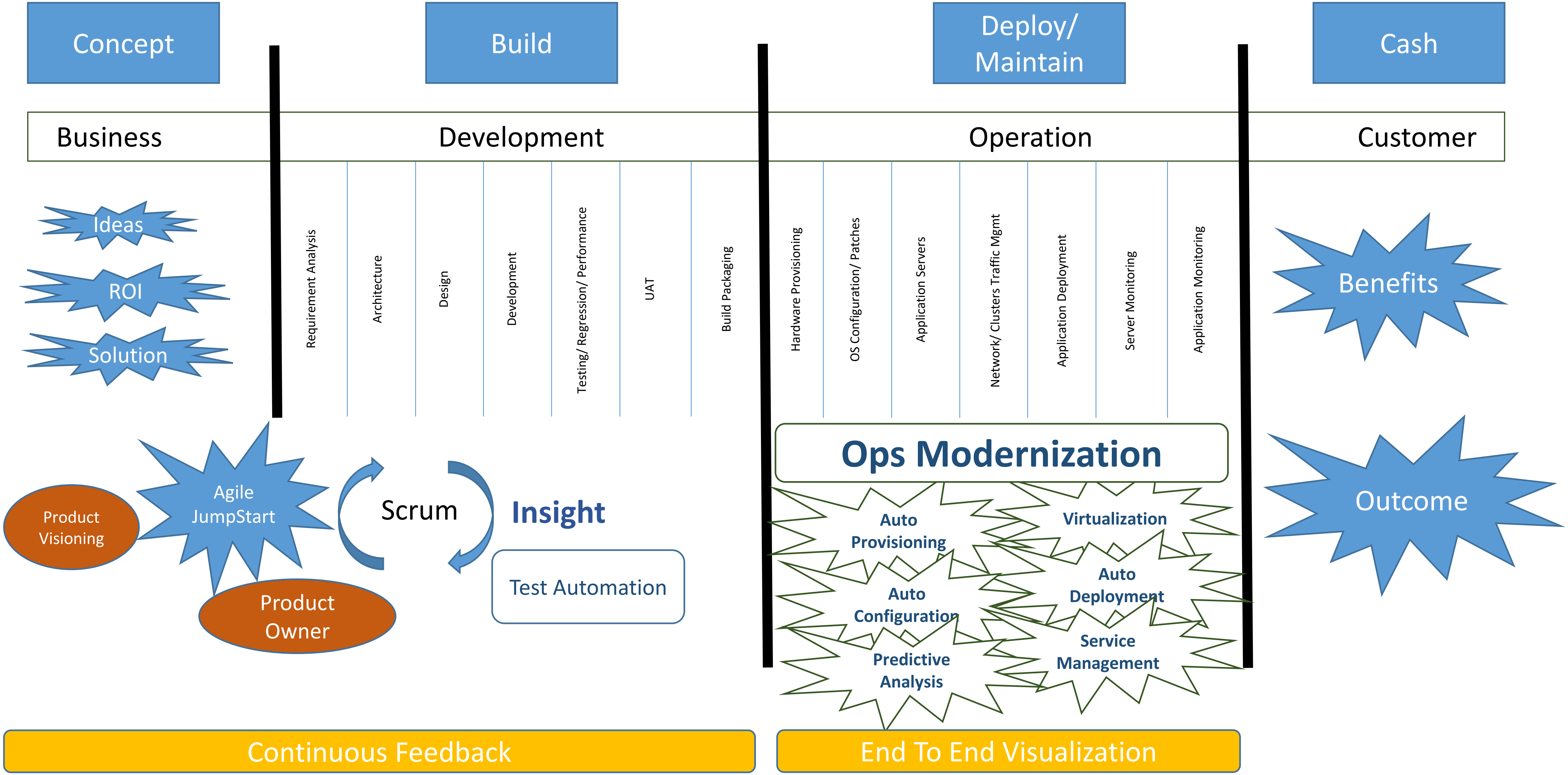
Insight

CI

Test Automation

Continuous Feedback

The Flow – Concept To Cash



Concept

Build

Deploy/ Maintain

Cash

Business

Development

Operation

Customer

Ideas

ROI

Solution

Requirement Analysis

Architecture

Design

Development

Testing/ Regression/ Performance

UAT

Build Packaging

Hardware Provisioning

OS Configuration/ Patches

Application Servers

Network/ Clusters Traffic Mgmt

Application Deployment

Server Monitoring

Application Monitoring

Benefits

Outcome

Product Visioning

Agile JumpStart

Product Owner

Scrum

Insight

Test Automation

Ops Modernization

Auto Provisioning

Virtualization

Auto Configuration

Auto Deployment

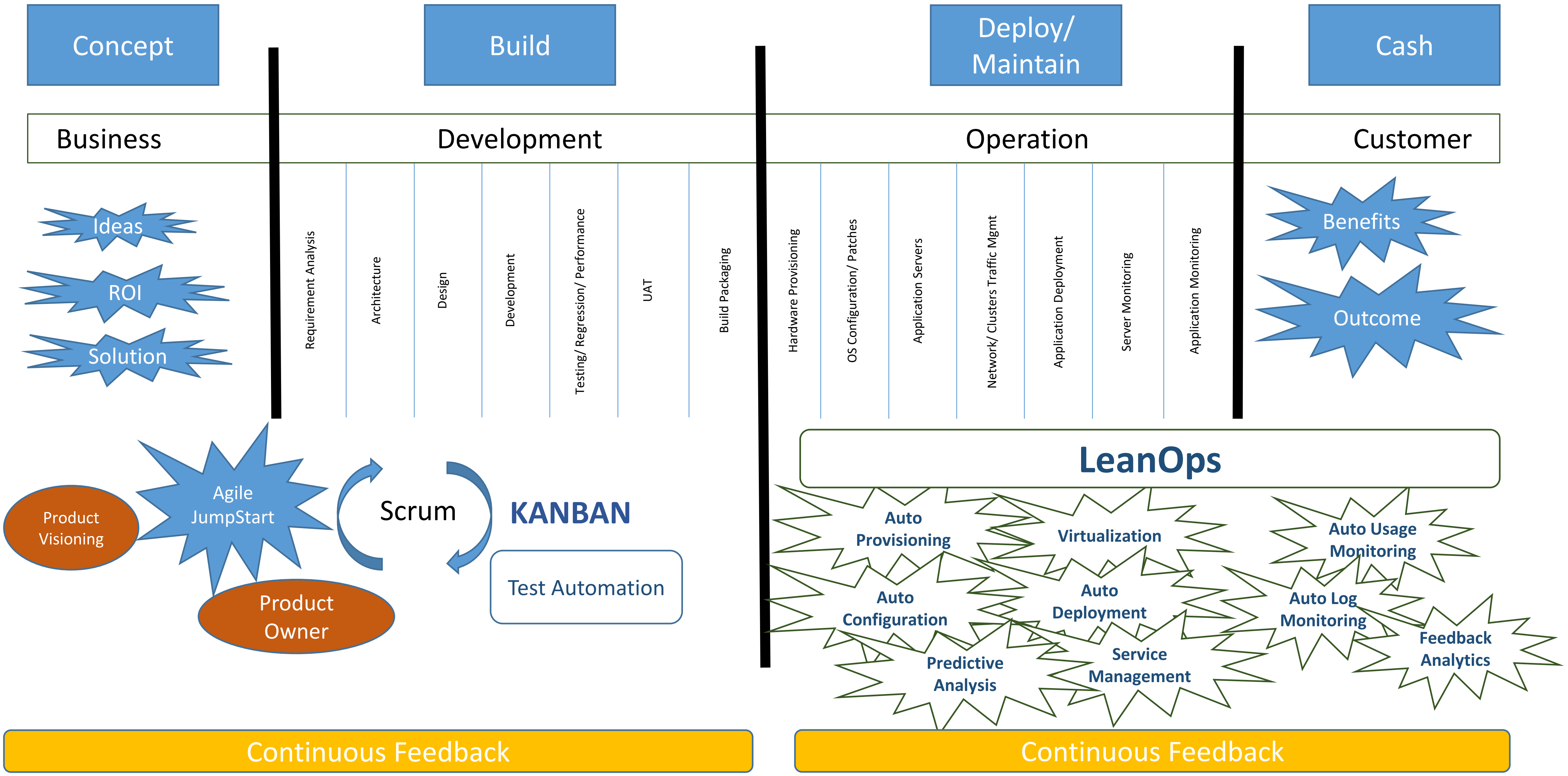
Predictive Analysis

Service Management

Continuous Feedback

End To End Visualization

Value Stream Flow – Concept To Cash



Concept

Build

Deploy/ Maintain

Cash

Business

Development

Operation

Customer

Ideas

ROI

Solution

Requirement Analysis

Architecture

Design

Development

Testing/ Regression/ Performance

UAT

Build Packaging

Hardware Provisioning

OS Configuration/ Patches

Application Servers

Network/ Clusters Traffic Mgmt

Application Deployment

Server Monitoring

Application Monitoring

Benefits

Outcome

Product Visioning

Agile JumpStart

Product Owner

Scrum

KANBAN

Test Automation

LeanOps

Auto Provisioning

Virtualization

Auto Usage Monitoring

Auto Configuration

Auto Deployment

Auto Log Monitoring

Predictive Analysis

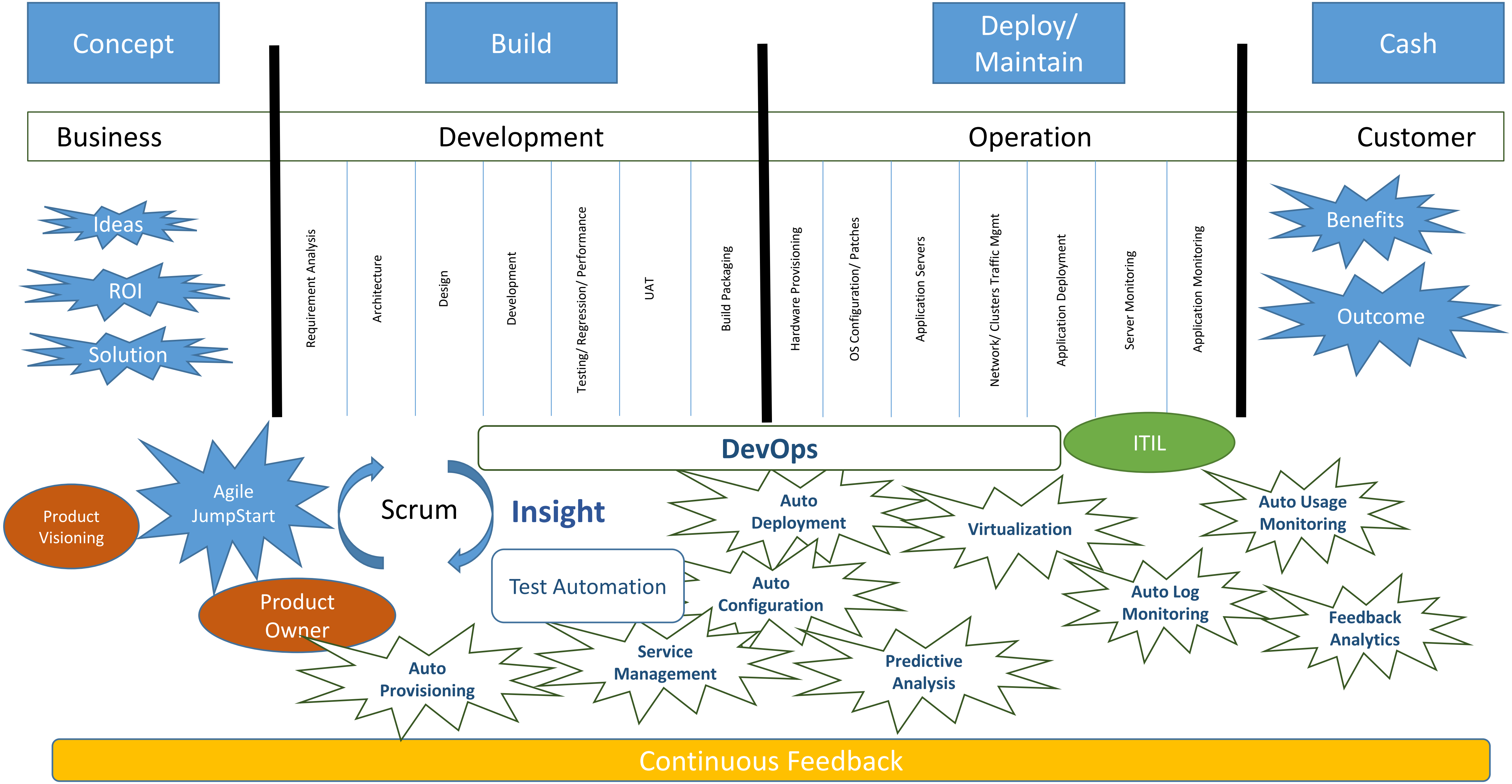
Service Management

Feedback Analytics

Continuous Feedback

Continuous Feedback

The Flow – Concept To Cash



Concept

Build

Deploy/ Maintain

Cash

Business

Development

Operation

Customer

Ideas

ROI

Solution

Requirement Analysis

Architecture

Design

Development

Testing/ Regression/ Performance

UAT

Build Packaging

Hardware Provisioning

OS Configuration/ Patches

Application Servers

Network/ Clusters Traffic Mgmt

Application Deployment

Server Monitoring

Application Monitoring

Benefits

Outcome

DevOps

ITIL

Product Visioning

Agile JumpStart

Scrum

Insight

Product Owner

Test Automation

Auto Deployment

Virtualization

Auto Usage Monitoring

Auto Configuration

Auto Log Monitoring

Feedback Analytics

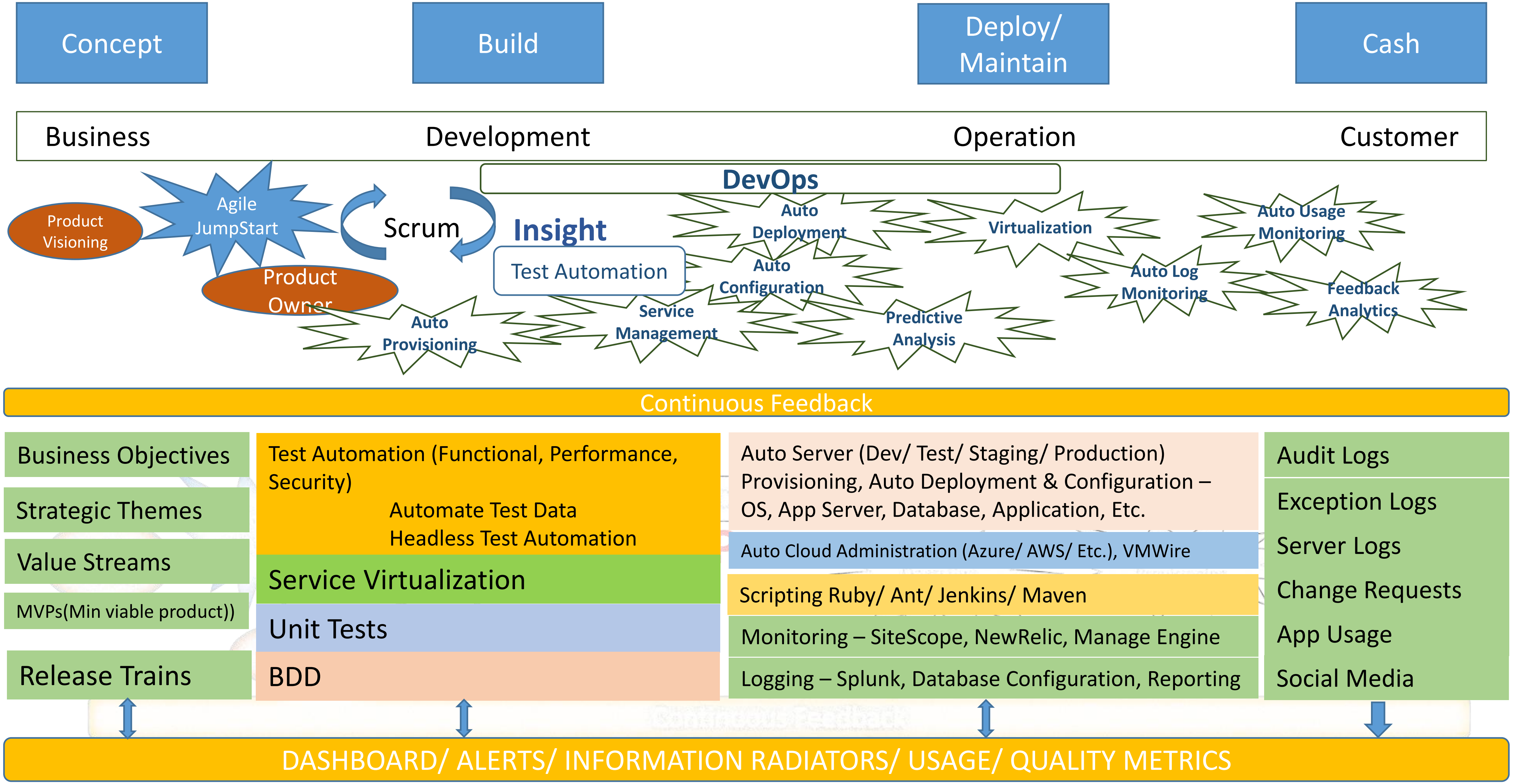
Auto Provisioning

Service Management

Predictive Analysis

Continuous Feedback

Value Stream Flow – Concept To Cash



Concept

Build

Deploy/ Maintain

Cash

Business

Development

Operation

Customer

DevOps

Continuous Feedback

Business Objectives

Strategic Themes

Value Streams

MVPs(Min viable product))

Release Trains

Test Automation (Functional, Performance, Security)

Automate Test Data

Headless Test Automation

Service Virtualization

Unit Tests

BDD

Auto Server (Dev/ Test/ Staging/ Production) Provisioning, Auto Deployment & Configuration – OS, App Server, Database, Application, Etc.

Auto Cloud Administration (Azure/ AWS/ Etc.), VMWire

Scripting Ruby/ Ant/ Jenkins/ Maven

Monitoring – SiteScope, NewRelic, Manage Engine

Logging – Splunk, Database Configuration, Reporting

Audit Logs

Exception Logs

Server Logs

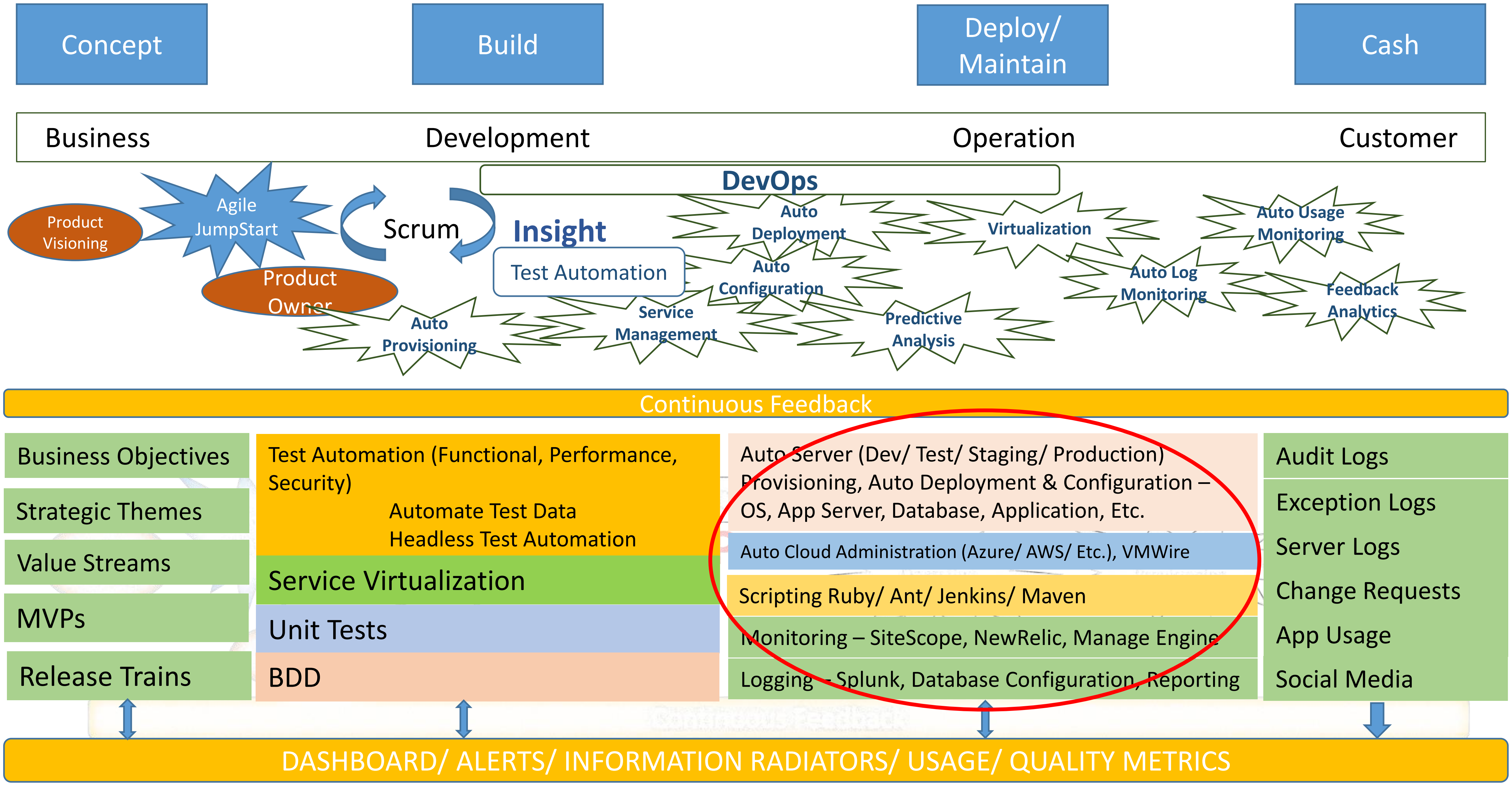
Change Requests

App Usage

Social Media

DASHBOARD/ ALERTS/ INFORMATION RADIATORS/ USAGE/ QUALITY METRICS

Value Stream Flow – Concept To Cash



Concept

Build

Deploy/ Maintain

Cash

Business

Development

Operation

Customer

DevOps

Product Visioning

Agile JumpStart

Product Owner

Scrum

Insight

Test Automation

Auto Provisioning

Service Management

Auto Deployment

Auto Configuration

Virtualization

Predictive Analysis

Auto Usage Monitoring

Auto Log Monitoring

Feedback Analytics

Continuous Feedback

Business Objectives

Strategic Themes

Value Streams

MVPs

Release Trains

Test Automation (Functional, Performance, Security)

Automate Test Data

Headless Test Automation

Service Virtualization

Unit Tests

BDD

Auto Server (Dev/ Test/ Staging/ Production) Provisioning, Auto Deployment & Configuration – OS, App Server, Database, Application, Etc.

Auto Cloud Administration (Azure/ AWS/ Etc.), VMWire

Scripting Ruby/ Ant/ Jenkins/ Maven

Monitoring – SiteScope, NewRelic, Manage Engine

Logging – Splunk, Database Configuration, Reporting

Audit Logs

Exception Logs

Server Logs

Change Requests

App Usage

Social Media

DASHBOARD/ ALERTS/ INFORMATION RADIATORS/ USAGE/ QUALITY METRICS

DevOps main objectives

- Installation of server hardware and OS
- Configuration of servers, networks, storage, etc…
- Monitoring of servers
- Respond to outages
- IT security
- Managing phone systems, network
- Change control
- Backup and disaster recovery planning
- Manage active directory
- Asset tracking

Infrastructure As A Code

- With the arrival of tools like Puppet, and Chef, the concept of Infrastructure as Code was born
- Infrastructure as code, or programmable infrastructure, means writing code (which can be done using a high level language or any descriptive language) to manage configurations and automate provisioning of infrastructure in addition to deployments.

DevOps on the Cloud

- What is the relationship between Cloud Computing and DevOps? Is DevOps really just “IT for the Cloud”? Can you only do DevOps in the cloud? Can you only do cloud using DevOps? The answer to all three questions is “no”. Cloud and DevOps are independent but mutually reinforcing strategies for delivering business value through IT

Diff between Manual and Automation setup of Infrastructure

Conventional way	Auto way
Manually setup servers with necessary OS/software's	Automated setup of servers with necessary
Not repeatable for other servers	Easily repeatable for new servers.
Not easily testable	Easily testable. Can be tested even without impacting the real nodes.
Scalability constraints. Time consuming and manual testing needed for each node.	Building a copy of any Chefed up node is very quick. Easy to test and integrate.
Dev != QA != PROD	Dev == QA == PROD

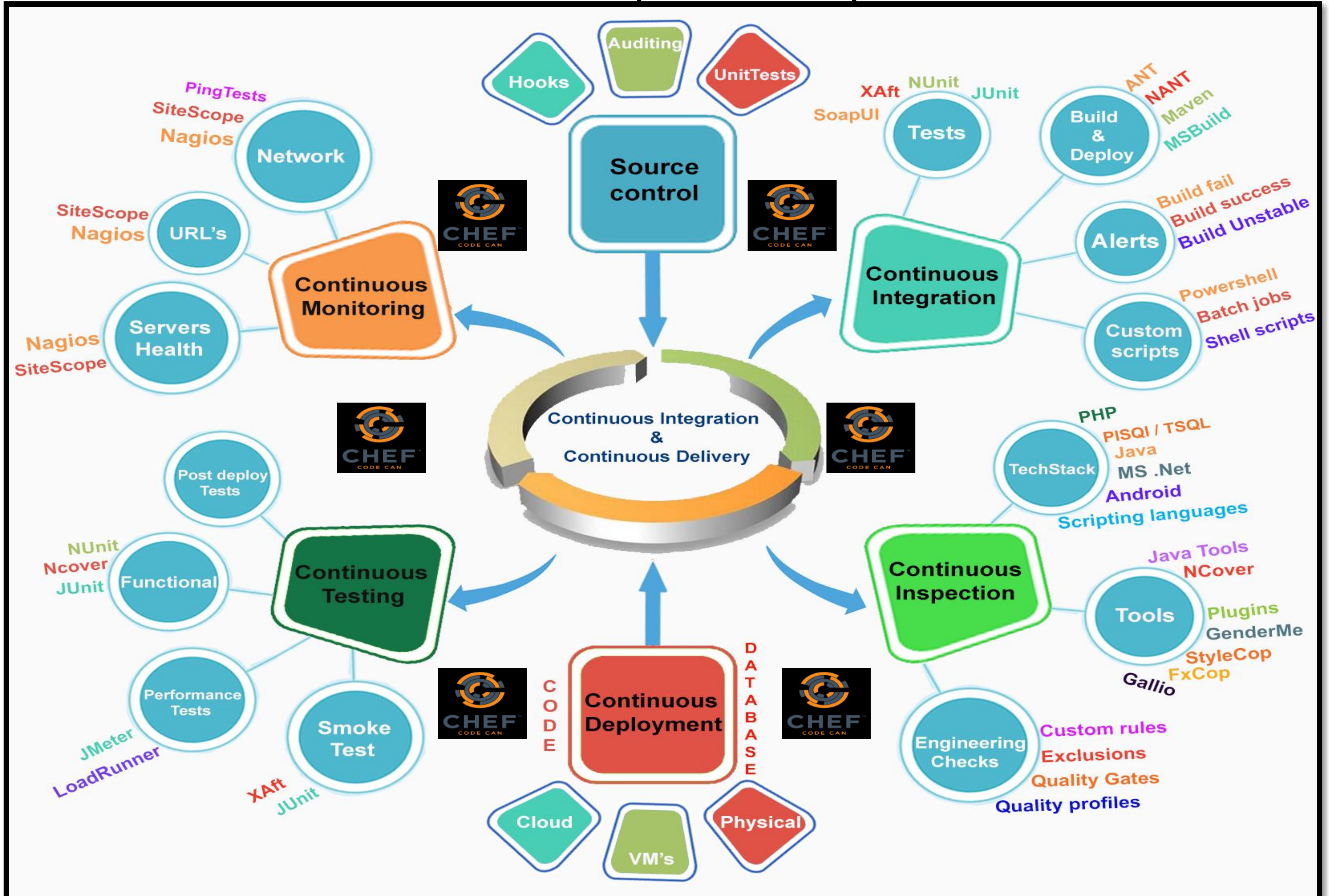
CLOUD COMPUTING and VIRTUALIZATION in DevOps

- What is Cloud Computing - the practice of using a network of remote servers hosted on the Internet to store, manage, and process data, rather than a local server or a personal computer.
- cloud services can be dived into 3 stacks
- laas - infrastructure as a service - Virtual machines, servers, storage, load balancers, networks.. etc.
- paas - Platform as a service - execution runtime, database, webserver, development tools...etc.
- saas - software as a service - Email, virtual desktop, games, communication.. etc.
- What is the relationship between Cloud Computing and DevOps?

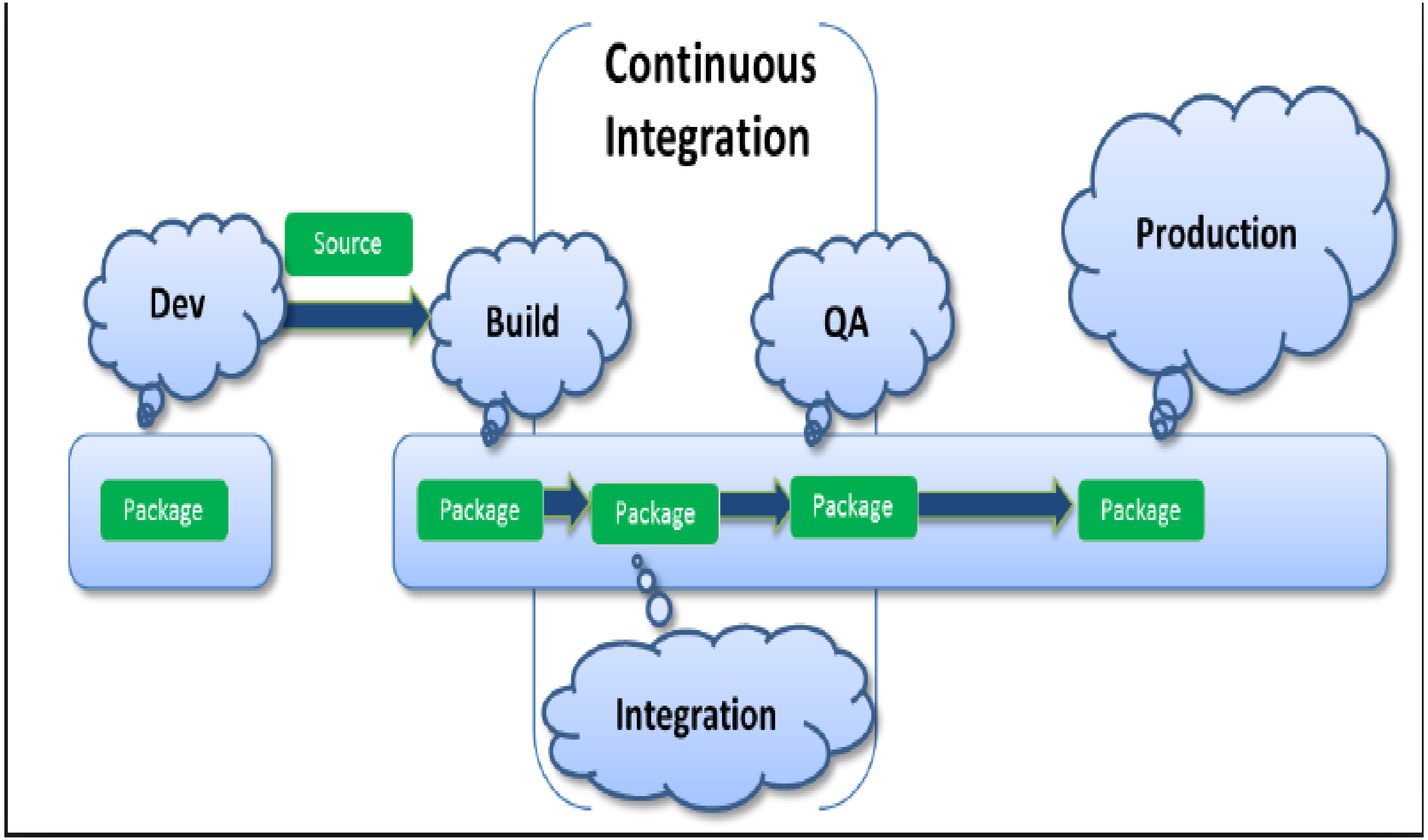
  -- Is DevOps really just “IT for the Cloud”?
  - Can you only do DevOps in the cloud?
  - Can you only do cloud using DevOps?

The answer to all three questions is “no”. Cloud and DevOps are independent but mutually reinforcing strategies for delivering business value through IT.

DevOps Components



DevOps Flow


Continuous Integration
Dev
Source
Build
QA
Production
Package
Package
Package
Package
Package
Integration

Topics to Cover

- Version Control System
- VSTS (CI / CD, Release Management, Dashboards and Reports, Planning)
- Azure Services
- MicroServices Deployment