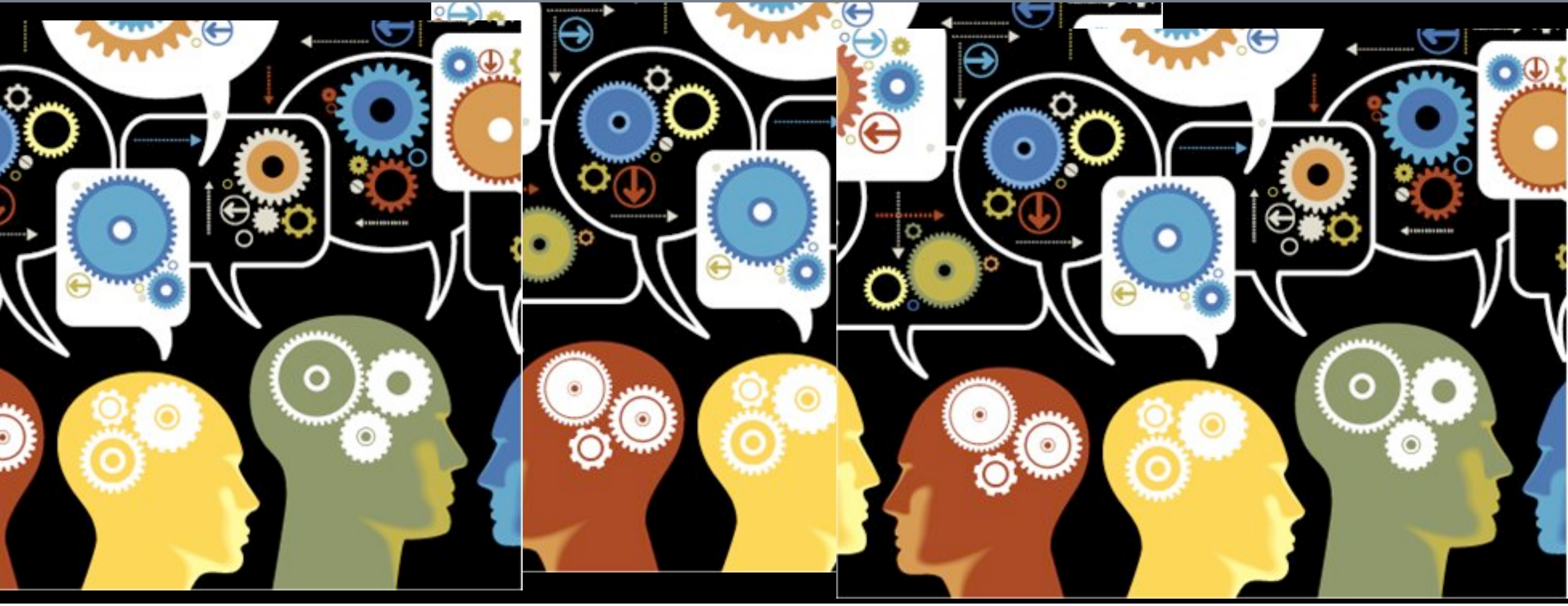
DevOps → Technical Agility

Dominic Delmolino

What is DevOps?

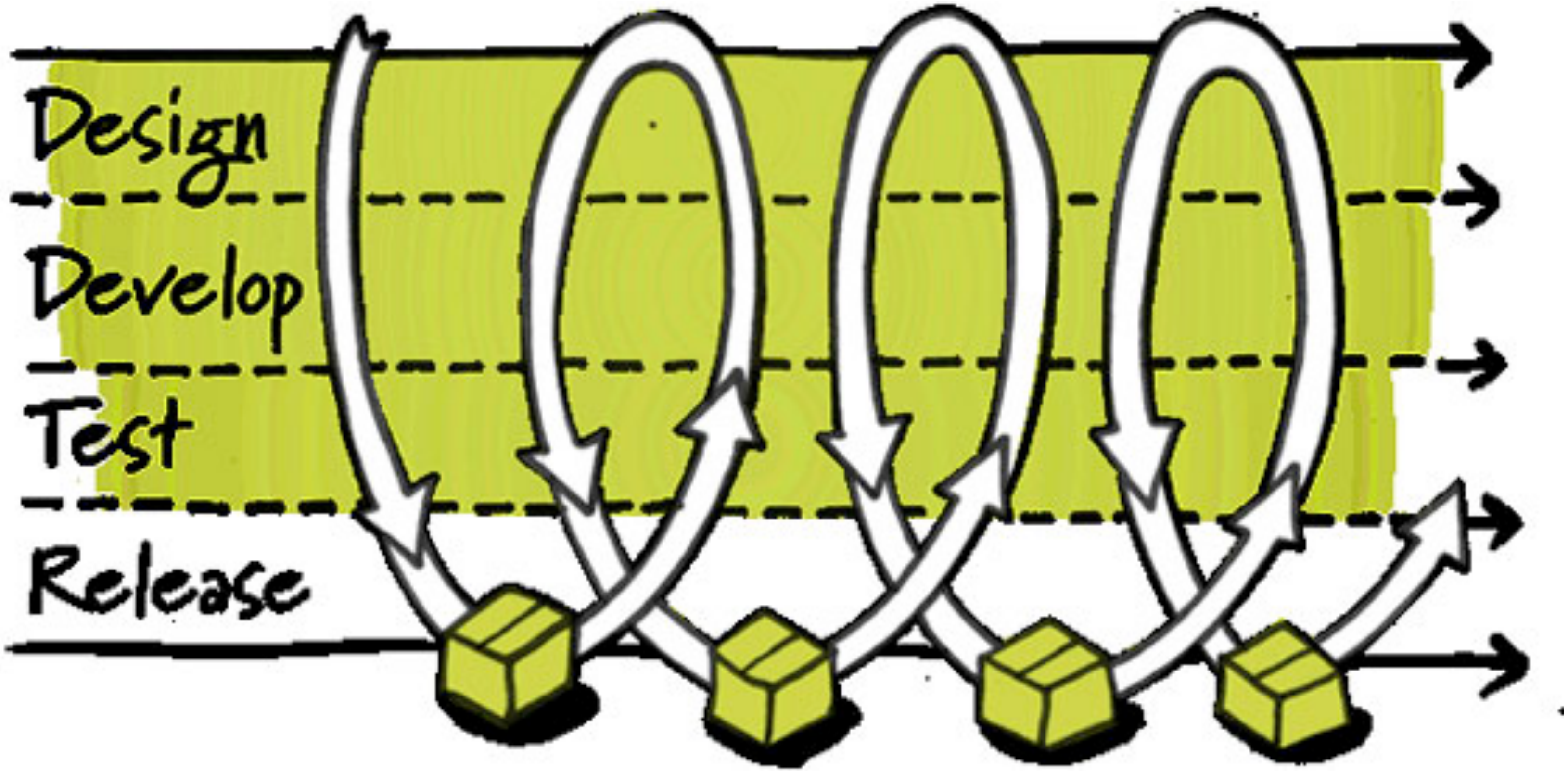


DevOps is the term for the practical technical, cultural and organizational techniques used to facilitate the practices of ***Agile Software Development*** *from* ***Requirements thru Release****.*

2015 Federal CASIC Workshops
Computer Assisted Survey Information Collection

Agilex

Agile Development = Rapid Feedback Cycles



How quickly can I test this so I can get feedback on what I need to do next?

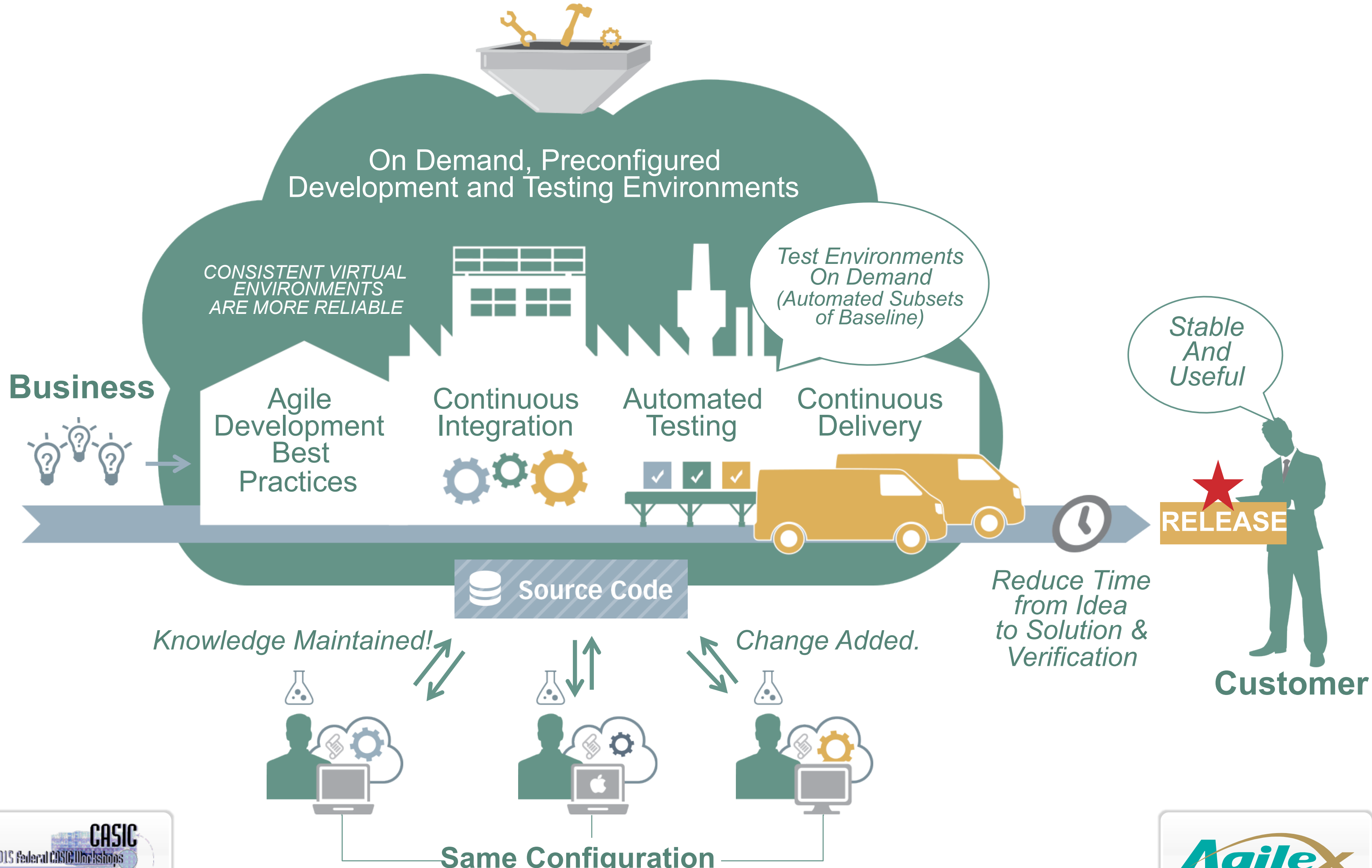
What is Agile Development?

- Value ***individuals and interactions*** over *processes and tools*
- ***Working software*** over *comprehensive documentation*
- ***Customer collaboration*** over *contract negotiation*
- ***Repsonding to change*** over *following a plan*

- Highest priority is to satisfy customer through ***early*** and ***continuous delivery*** of valuable software
- Welcome changing requirements; ***harness change for improvement***
- Deliver working software frequently, from weeks to months, ***preferring the shortest possible timescale***
- At regular intervals, teams reflect on how to become more effective, then ***tunes and adjusts*** accordingly
- Working software is the primary measure of progress
- Simplicity – the art of maximizing the amount of work not done – is essential

Constantly measure whether or not software is delivering value, correct and adjust as often as necessary

What does DevOps look like?



On Demand, Preconfigured Development and Testing Environments

CONSISTENT VIRTUAL ENVIRONMENTS ARE MORE RELIABLE

Test Environments On Demand (Automated Subsets of Baseline)

Business

Agile Development Best Practices

Continuous Integration

Automated Testing

Continuous Delivery

Stable And Useful

RELEASE

Reduce Time from Idea to Solution & Verification

Customer

Source Code

Knowledge Maintained!

Change Added.

Same Configuration

“Consistent and Repeatable”

CASIC
2015 Federal CASIC Workshops
Computer Assisted Survey Information Collection

Agilex

DevOps Capabilities for Agile Development

- cloud based resources
- workstation and server configuration management
- platform and tool repository
- directory and notification services

- automated testing and continuous integration
- distributed version control
- online code review services
- authorization and access control

- agile lifecycle management
- collaboration support

DevOps Workflow

1

- Amazon Web Services
- VMWare

- Developer Onboarding
  - Issued Physical *or Virtual* Workstation
  - Images built by Operations & Security

2

- Active Directory
- Atlassian Crowd
- OpenLDAP / OpenDJ

- Project admin adds Developer to Project
- *Authorizes access* to Project repositories, wiki spaces and build server

3

- Nexus (project tools)
- Artifactory (project tools)
- Chef
- Puppet
- Git (GitLab, Git Enterprise)
- Subversion

- Developer downloads Project-specific workstation configuration from *binary repository*
- Developer runs *workstation configuration*
  - Auto-installs tools and source code repo

4

- JIRA / Greenhopper
- Rally
- VersionOne
- Confluence
- XWiki

- Developer writes tests and code
- Gets stories from ALM, Adds content to wiki

5

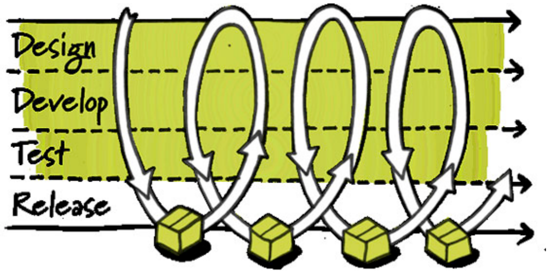
- Maven / Gradle
- JUnit / DBUnit
- Selenium / Geb / Spock
- Amazon Web Services
- VMWare
- Docker

- Developer builds code and tests in *private local environment* (on workstation or in cloud)
- Local environment matches production environment as much as possible

6

- Review Board
- Barkeep
- Confluence
- XWiki

- Developer posts code for online review and commentary by peers and supervisors



7

- Git (GitLab, Git Enterprise)
- Subversion

- After review, Developer *commits and pushes* code to central source code repository for project
- Developer updates tasks and stories

8

- Jenkins / Hudson
- Bamboo
- Maven / Gradle
- JUnit / DBUnit
- Selenium / Geb / Spock
- Git (GitLab, Git Enterprise)
- Subversion
- Chef
- Puppet
- Email Notification
- Amazon Web Services
- VMWare
- Nexus (project tools)
- Artifactory (project tools)

- Build Server polls for and retrieves code changes
  - Builds executables/war files and displays status
  - Notifies developers of failures and need to *revert*
  - Performs builds and tests in succession
- Example build flow:
  - Development build (check for coding errors)
  - Integration build (may include mock services)
  - Functional test build (creation of cloud test VMs)
  - Security test build (i.e. Foritfy scans)
  - Deployment build (results in deployment artifact in binary repository, i.e., Nexus / Artifactory)

Application to Survey Channels and Surveys

Deployment

- Incremental Deployment and Verification

- Deploy “Coming Soon” static page
- Deploy “Countdown clock”
- Deploy “Register to get notified”
- Deploy “Login and create a profile”
- Deploy “Survey capability 1”
- Deploy “Survey capability 2”

- Process acts as a “pipecleaner” to clear potential issues

Testing

- Validate Survey Sequencing and Branching

- Automate tests for all questions and answers
- Automate branch testing
- Automate accessibilty testing

- Tests highlight changes to survey that break sequences or make answer options inaccesible
- Test with every change, maintain green survey

Production Scenarios

- Scalability
  Auto-scale using image deployments

- Zero-downtime Fixes
  Spin up v2, re-direct traffic, spin down v1

- A/B Testing & Sampling
  Multiple versions

- GeoIP traffic sampling
  Like A/B but with directed routing

Overall Benefits to the DevOps Approach

- Rapid, Consistent Deployments
- Continuous Testing, Validation and Verification
- Scalable Deployments
- Zero-downtime fixes
- Traffic-routed testing
- More ways to verify and measure survey options

Thank You

Dominic Delmolino
email: dominic.delmolino@agilex.com
office: 703.889.3800

CASIC
2015 Federal CASIC Workshops
Computer Assisted Survey Information Collection

Agilex