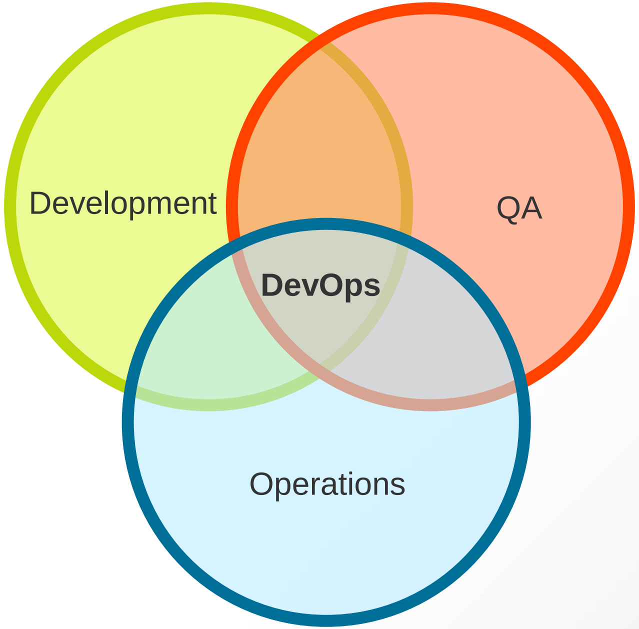
HOW TO ACHIEVE DEVOPS FROM STANDALONE WEB METHODS

Dave Pemberton
Global Consulting Services
webMethods Practice Manager UK & Nordic

DEVOPS
WHAT IS IT?

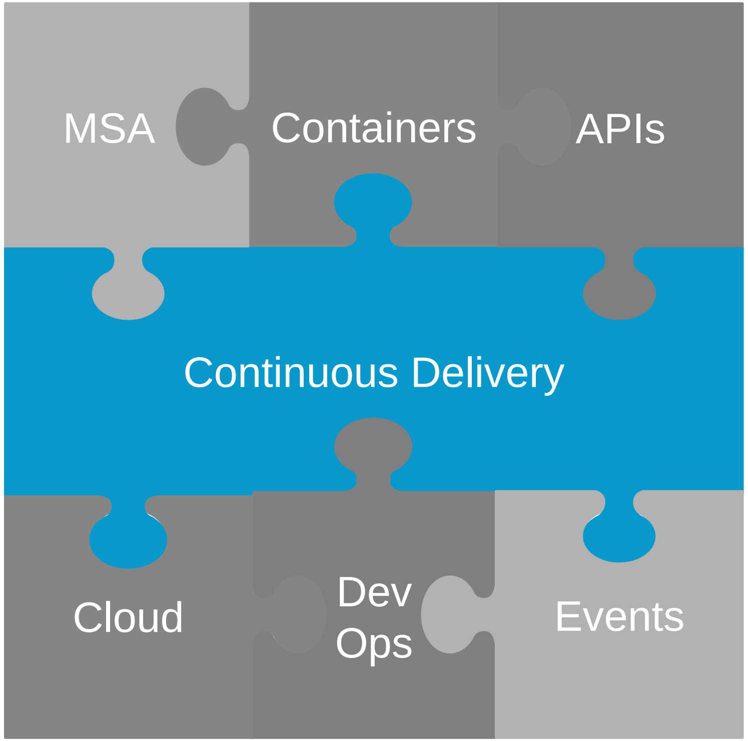
- DevOps is the practice of IT Operations and Development participating together in the entire software lifecycle, from design through the development process to production support
- Devops is a philosophy, cultural change and paradigm shift.

- Devops is not a tool!
- Devops is not a Job description or role!
- **It's the way we work and behave**



WHY DEVOPS

WHY DEVOPS
SOME OF THE COMPETITION



MSA

Containers

APIs

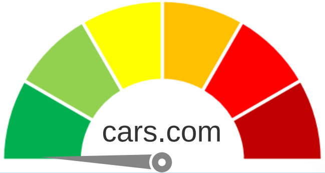
Continuous Delivery

Cloud

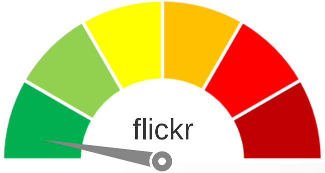
Dev Ops

Events

300 deployments per **year**



10+ deployments per **day**



50+ deployments per **day**

11.6 deployment every **sec**

WHY DEVOPS
CUSTOMER EXAMPLES

Walgreens Boots Alliance

- **Implemented a complete DevOps CI/CD deployment pipeline**
- Previously a deployment took an entire weekend with shift working
- Following a 80-150 page script
- Manual
- Error Prone
- Reduced time to 15 minutes!
- Increased stability and reliability

British Army

- **Agile development and DevOps is firmly ingrained in the culture**
- Agile development processes are important because of the ever changing requirements
- Devops continuous processes in place to automate onerous deployment processes
- Considerably increases agility, and accuracy of delivery

DEVOPS
FOUNDATIONS

AGILE

- Deliver Fast
- Deliver Often
- Deliver Right

BUILD AUTOMATION

- Reliability
- Consistency
- Predictability

DEPLOY AUTOMATION

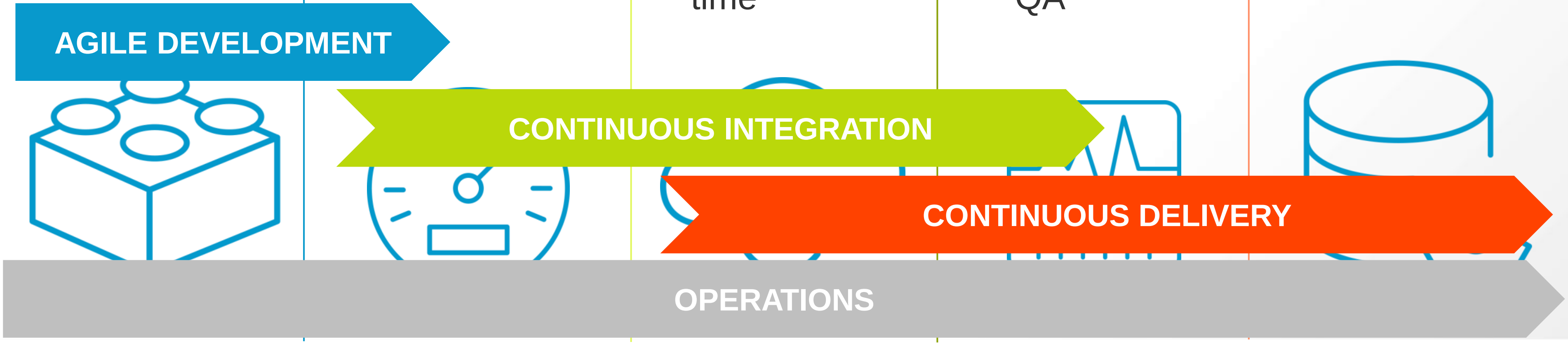
- Reduce Time
- Reduce Errors
- Reduce down time

TEST AUTOMATION

- Improve Quality
- Accelerate QA

AUTOMATED PROVISON

- Reliability
- Reduce Risks



AGILE DEVELOPMENT

CONTINUOUS INTEGRATION

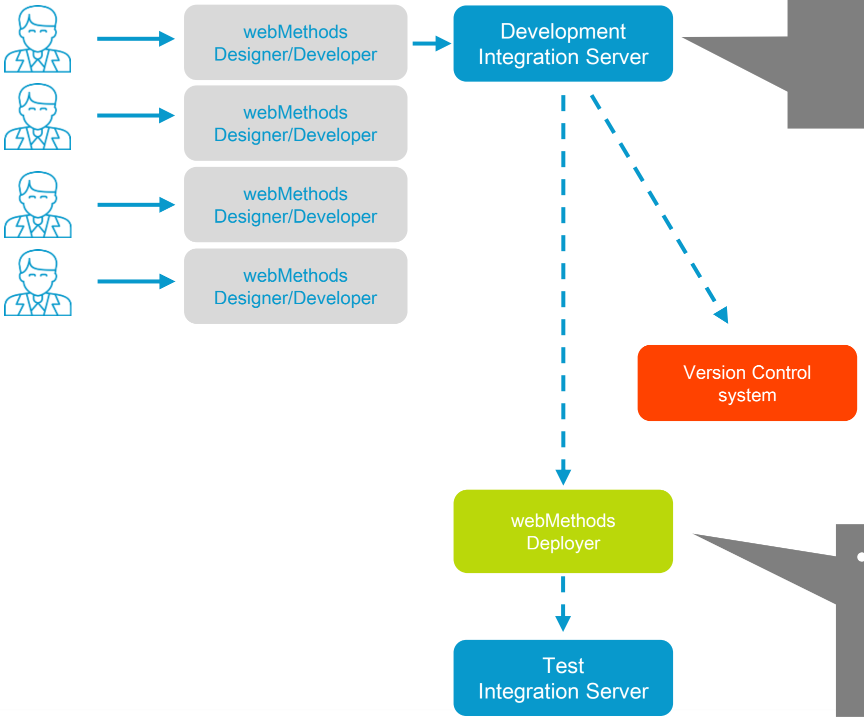
CONTINUOUS DELIVERY

OPERATIONS

WEBMETHODS DEVELOPMENT

WEBMETHODS DEVELOPMENT
THEN AND NOW

webMethods (Centralised)



webMethods Designer/Developer

webMethods Designer/Developer

webMethods Designer/Developer

webMethods Designer/Developer

Development Integration Server

Version Control system

webMethods Deployer

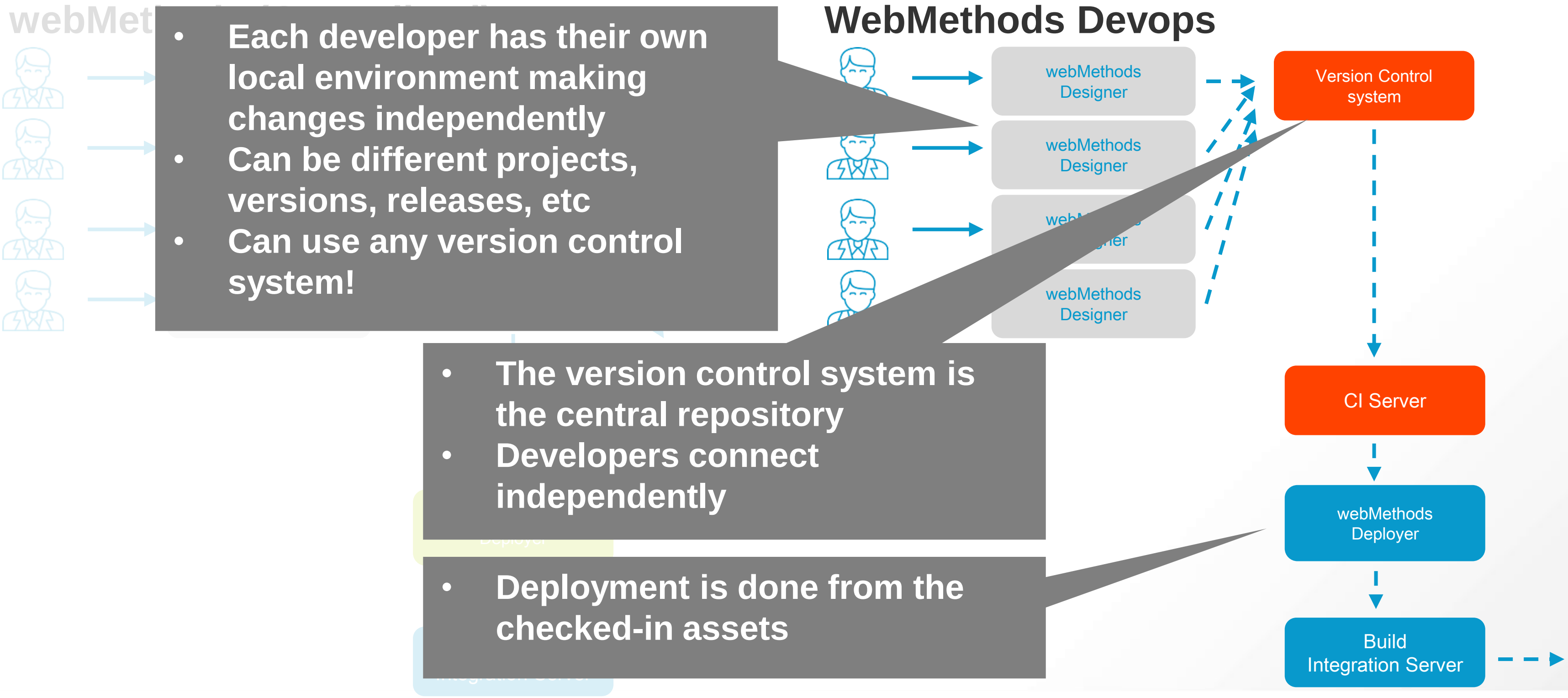
Test Integration Server

- One Code base shared and changed by all developers concurrently

- Limited version control support
  - Subversion
  - ClearCase
  - Visual SourceSafe
- Infact WmVCS was deprecated

- Development Server is not a reliable source and version control bypassed for deployment

WEBMETHODS DEVELOPMENT
THEN AND NOW

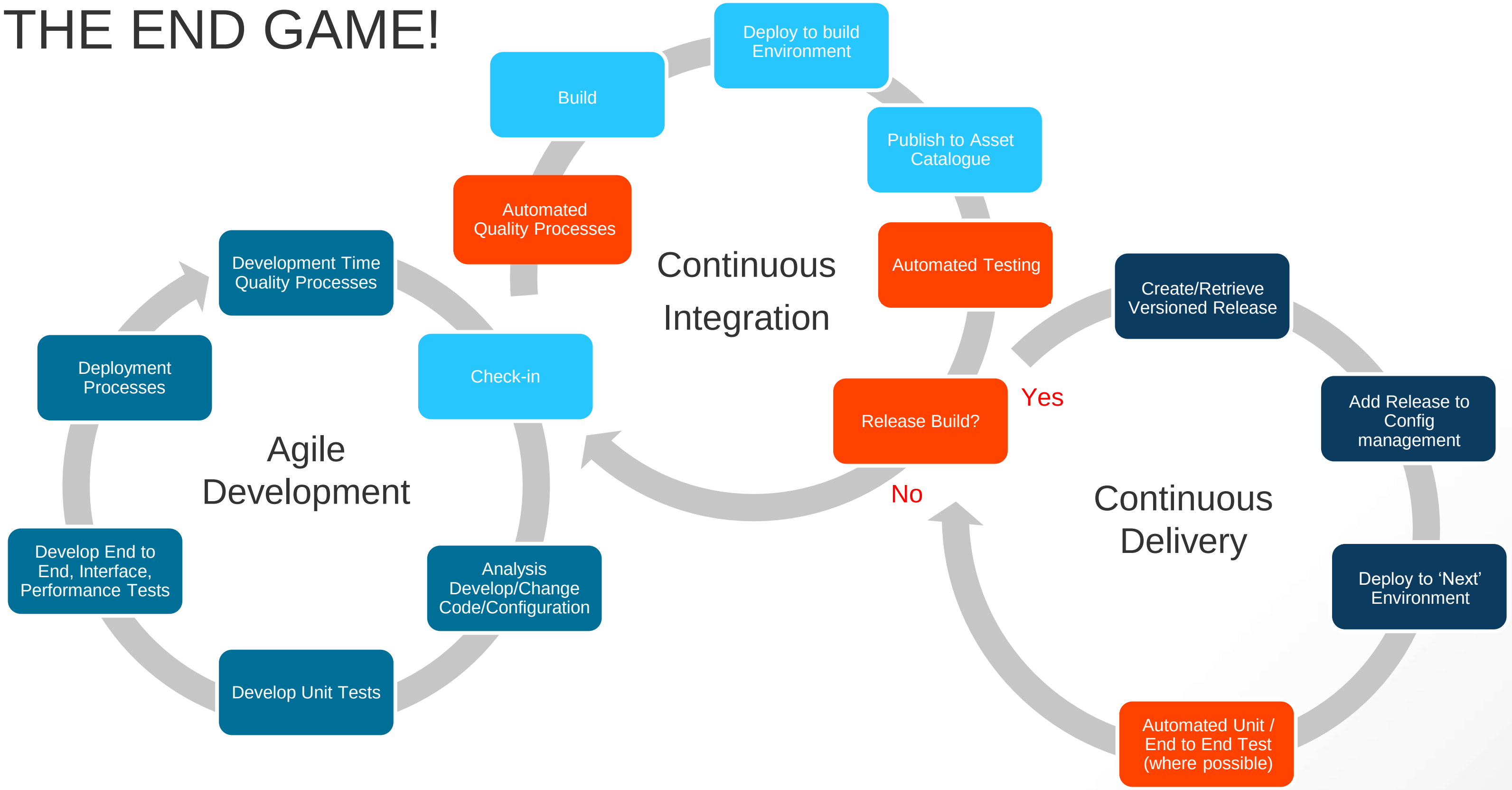


- Each developer has their own local environment making changes independently
- Can be different projects, versions, releases, etc
- Can use any version control system!

- The version control system is the central repository
- Developers connect independently

- Deployment is done from the checked-in assets

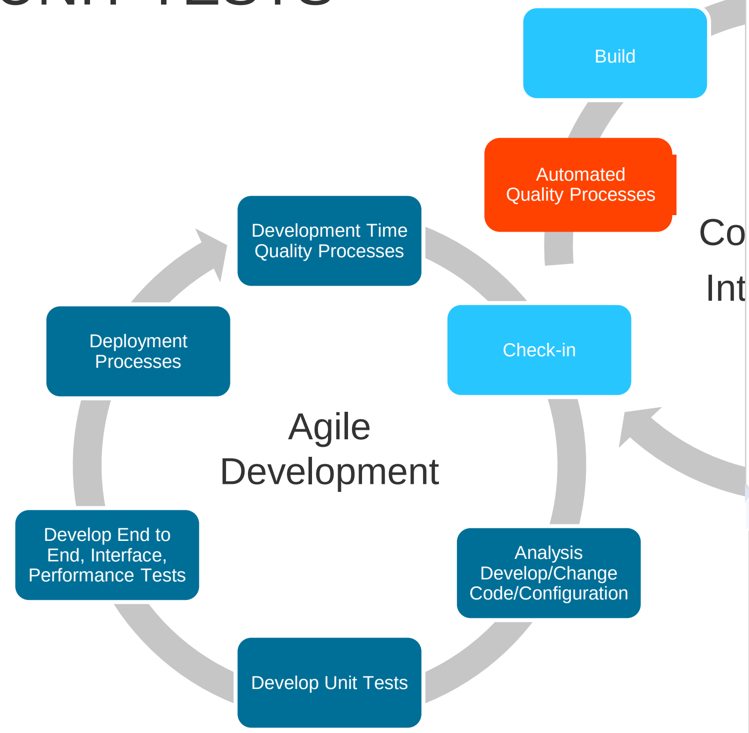
CONTINUOUS LIFECYCLE
THE END GAME!

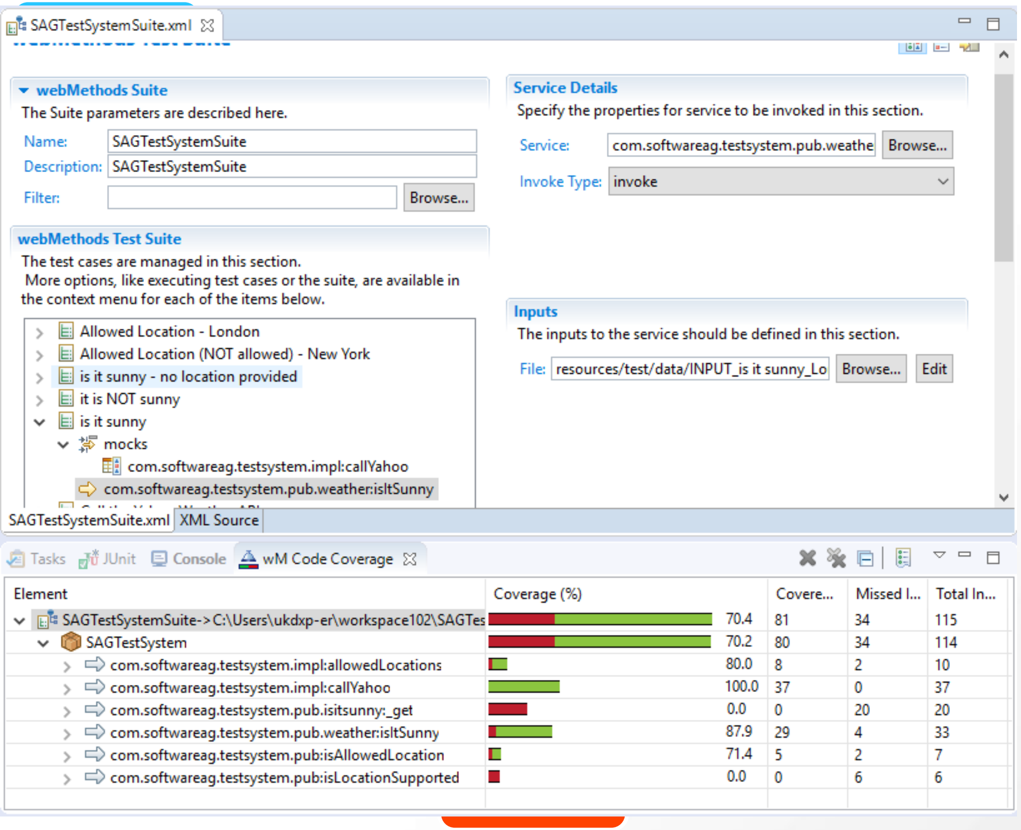


Build
Deploy to build Environment
Publish to Asset Catalogue
Automated Quality Processes
Automated Testing
Continuous Integration
Development Time Quality Processes
Check-in
Deployment Processes
Agile Development
Release Build?
Yes
No
Create/Retrieve Versioned Release
Add Release to Config management
Continuous Delivery
Deploy to 'Next' Environment
Develop End to End, Interface, Performance Tests
Analysis Develop/Change Code/Configuration
Develop Unit Tests
Automated Unit / End to End Test (where possible)

WITH THE DEVOPS EDITION

CONTINUOUS LIFECYCLE
UNIT TESTS


Build
Automated Quality Processes
Development Time Quality Processes
Deployment Processes
Check-in
Agile Development
Develop End to End, Interface, Performance Tests
Analysis Develop/Change Code/Configuration
Develop Unit Tests
Co
Int



webMethods Suite

The Suite parameters are described here.

Name: SAGTestSystemSuite

Description: SAGTestSystemSuite

Filter:

webMethods Test Suite

The test cases are managed in this section.
More options, like executing test cases or the suite, are available in the context menu for each of the items below.

- Allowed Location - London
- Allowed Location (NOT allowed) - New York
- is it sunny - no location provided
- it is NOT sunny
- is it sunny
  - mocks
    - com.softwareag.testsystem.impl:callYahoo
  - com.softwareag.testsystem.pub.weather:isItSunny

Service Details

Specify the properties for service to be invoked in this section.

Service: com.softwareag.testsystem.pub.weathe

Invoke Type: invoke

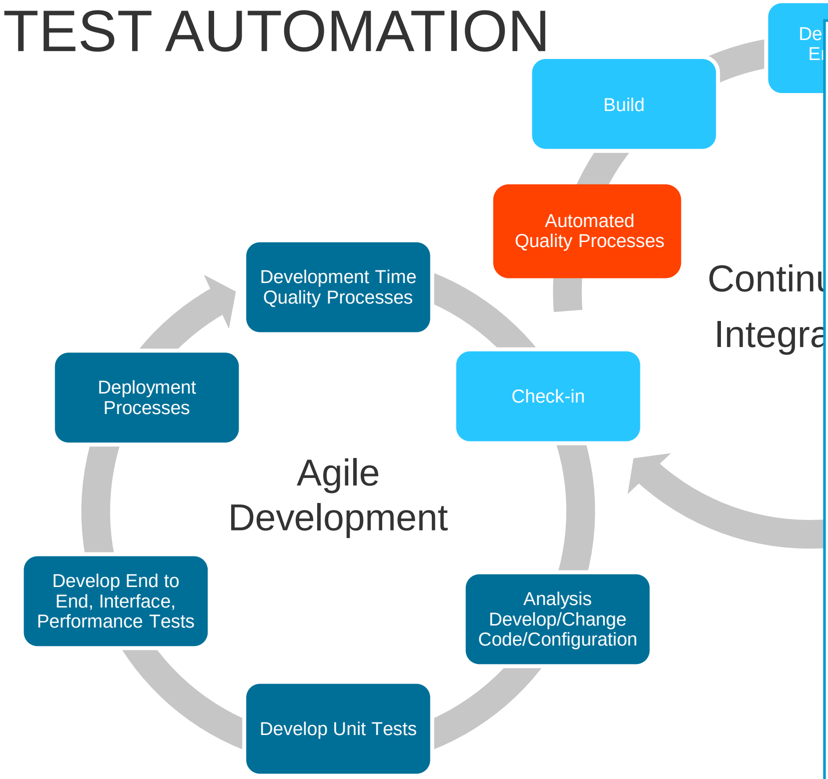
Inputs

The inputs to the service should be defined in this section.

File: resources/test/data/INPUT_is it sunny_Lo

Element	Coverage (%)	Covere...	Missed I...	Total In...
SAGTestSystemSuite-> C:\Users\ukdxp-er\workspace102\SAGTes	70.4	81	34	115
SAGTestSystem	70.2	80	34	114
com.softwareag.testsystem.impl:allowedLocations	80.0	8	2	10
com.softwareag.testsystem.impl:callYahoo	100.0	37	0	37
com.softwareag.testsystem.pub.isitsunny:_get	0.0	0	20	20
com.softwareag.testsystem.pub.weather:isItSunny	87.9	29	4	33
com.softwareag.testsystem.pub:isAllowedLocation	71.4	5	2	7
com.softwareag.testsystem.pub:isLocationSupported	0.0	0	6	6

CONTINUOUS LIFECYCLE
TEST AUTOMATION



De
Er

Build

Automated Quality Processes

Continu
Integra

Development Time Quality Processes

Check-in

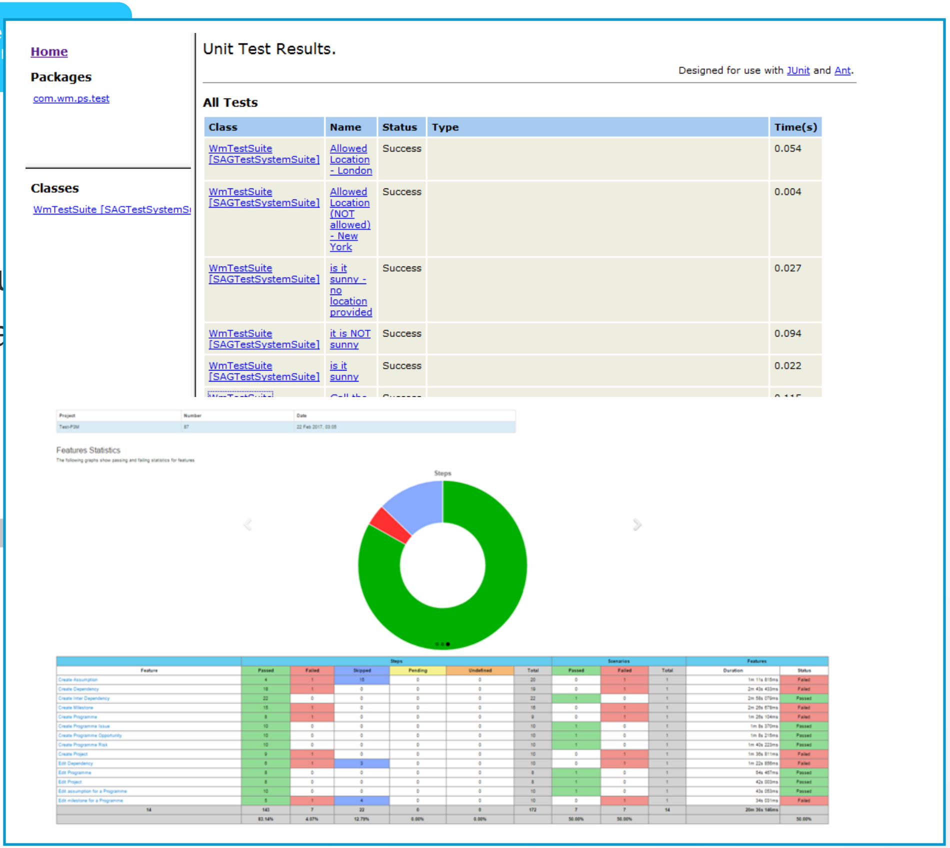
Deployment Processes

Agile Development

Develop End to End, Interface, Performance Tests

Analysis Develop/Change Code/Configuration

Develop Unit Tests



Home

Packages

com.wm.ps.test

Classes

WmTestSuite [SAGTestSystemS

Unit Test Results.

Designed for use with JUnit and Ant.

All Tests

Class	Name	Status	Type	Time(s)
WmTestSuite [SAGTestSystemSuite]	Allowed Location - London	Success		0.054
WmTestSuite [SAGTestSystemSuite]	Allowed Location (NOT allowed) - New York	Success		0.004
WmTestSuite [SAGTestSystemSuite]	is it sunny - no location provided	Success		0.027
WmTestSuite [SAGTestSystemSuite]	it is NOT sunny	Success		0.094
WmTestSuite [SAGTestSystemSuite]	is it sunny	Success		0.022

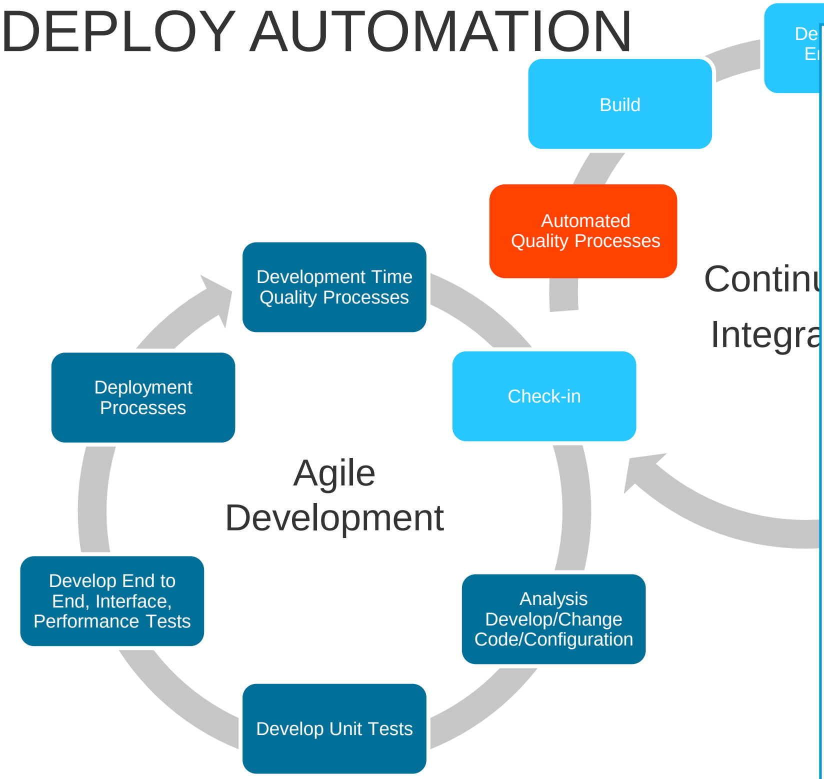
Project	Number	Date
Test-PSM	87	22 Feb 2017, 03:05

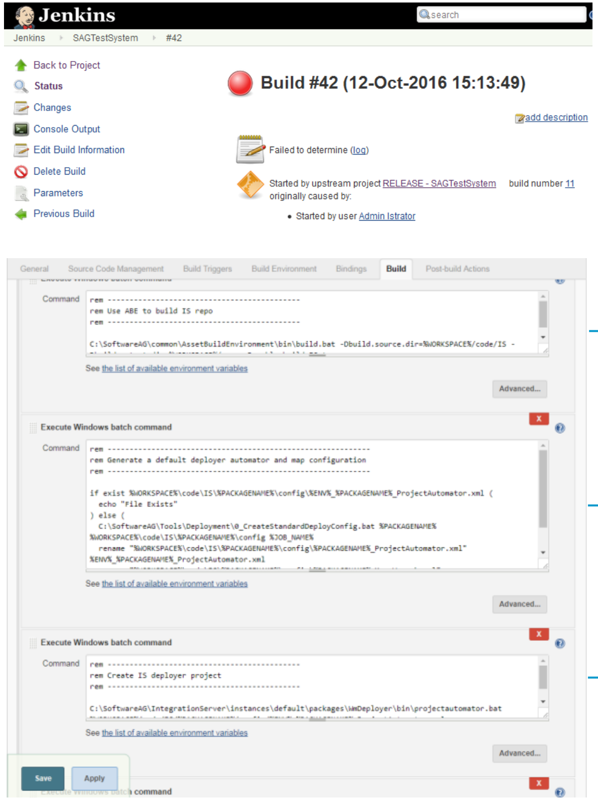
Features Statistics

The following graphs show passing and failing statistics for features

	Steps						Scenarios			Features	
Feature	Passed	Failed	Skipped	Pending	Undefined	Total	Passed	Failed	Total	Duration	Status
Create Assumption	4	1	15	0	0	20	0	1	1	1m 11s 815ms	Failed
Create Dependency	18	1	0	0	0	19	0	1	1	2m 43s 433ms	Failed
Create Inter Dependency	22	0	0	0	0	22	1	0	1	2m 58s 079ms	Passed
Create Milestone	15	1	0	0	0	16	0	1	1	2m 26s 678ms	Failed
Create Programme	8	1	0	0	0	9	0	1	1	1m 26s 104ms	Failed
Create Programme Issue	10	0	0	0	0	10	1	0	1	1m 8s 370ms	Passed
Create Programme Opportunity	10	0	0	0	0	10	1	0	1	1m 8s 215ms	Passed
Create Programme Risk	10	0	0	0	0	10	1	0	1	1m 40s 223ms	Passed
Create Project	9	1	0	0	0	10	0	1	1	1m 36s 811ms	Failed
Edit Dependency	6	1	3	0	0	10	0	1	1	1m 22s 856ms	Failed
Edit Programme	8	0	0	0	0	8	1	0	1	54s 467ms	Passed
Edit Project	8	0	0	0	0	8	1	0	1	42s 003ms	Passed
Edit assumption for a Programme	10	0	0	0	0	10	1	0	1	43s 053ms	Passed
Edit milestone for a Programme	5	1	4	0	0	10	0	1	1	34s 031ms	Failed
14	143	7	22	0	0	172	7	7	14	20m 36s 146ms	
	83.14%	4.07%	12.79%	0.00%	0.00%		50.00%	50.00%			50.00%

CONTINUOUS LIFECYCLE
DEPLOY AUTOMATION


De
En
Build
Automated Quality Processes
Continu
Integra
Development Time Quality Processes
Check-in
Deployment Processes
Agile Development
Develop End to End, Interface, Performance Tests
Analysis Develop/Change Code/Configuration
Develop Unit Tests



Build #42 (12-Oct-2016 15:13:49)

Failed to determine (log)

Started by upstream project RELEASE - SAGTestSystem build number 11
originally caused by:

- Started by user Admin Istrator

Command

```
rem -------------------------------------------
rem Use ABE to build IS repo
rem -------------------------------------------

C:\SoftwareAG\common\AssetBuildEnvironment\bin\build.bat -Dbuild.source.dir=%WORKSPACE%/code/IS -
```

See the list of available environment variables

Execute Windows batch command

Command

```
rem -----------------------------------------------------------
rem Generate a default deployer automator and map configuration
rem -----------------------------------------------------------

if exist %WORKSPACE%\code\IS\%PACKAGENAME%\config\%ENV%_%PACKAGENAME%_ProjectAutomator.xml (
  echo "File Exists"
) else (
  C:\SoftwareAG\Tools\Deployment\0_CreateStandardDeployConfig.bat %PACKAGENAME%
%WORKSPACE%\code\IS\%PACKAGENAME%\config %JOB_NAME%
  rename "%WORKSPACE%\code\IS\%PACKAGENAME%\config\%PACKAGENAME%_ProjectAutomator.xml"
%ENV%_%PACKAGENAME%_ProjectAutomator.xml
```

See the list of available environment variables

Execute Windows batch command

Command

```
rem -------------------------------------------
rem Create IS deployer project
rem -------------------------------------------

C:\SoftwareAG\IntegrationServer\instances\default\packages\WmDeployer\bin\projectautomator.bat
```

See the list of available environment variables

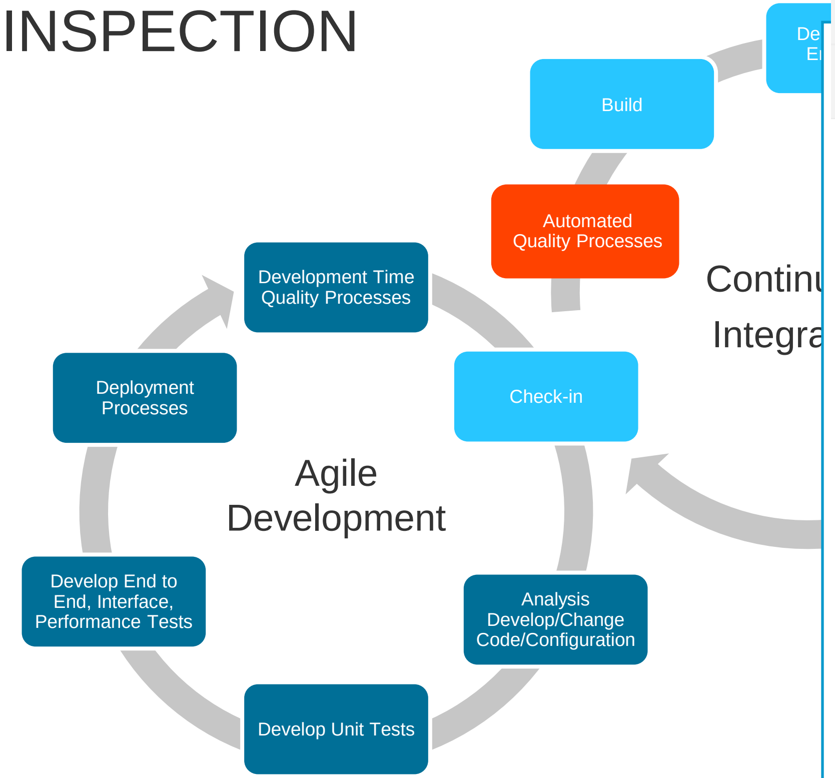
Execute Windows batch command

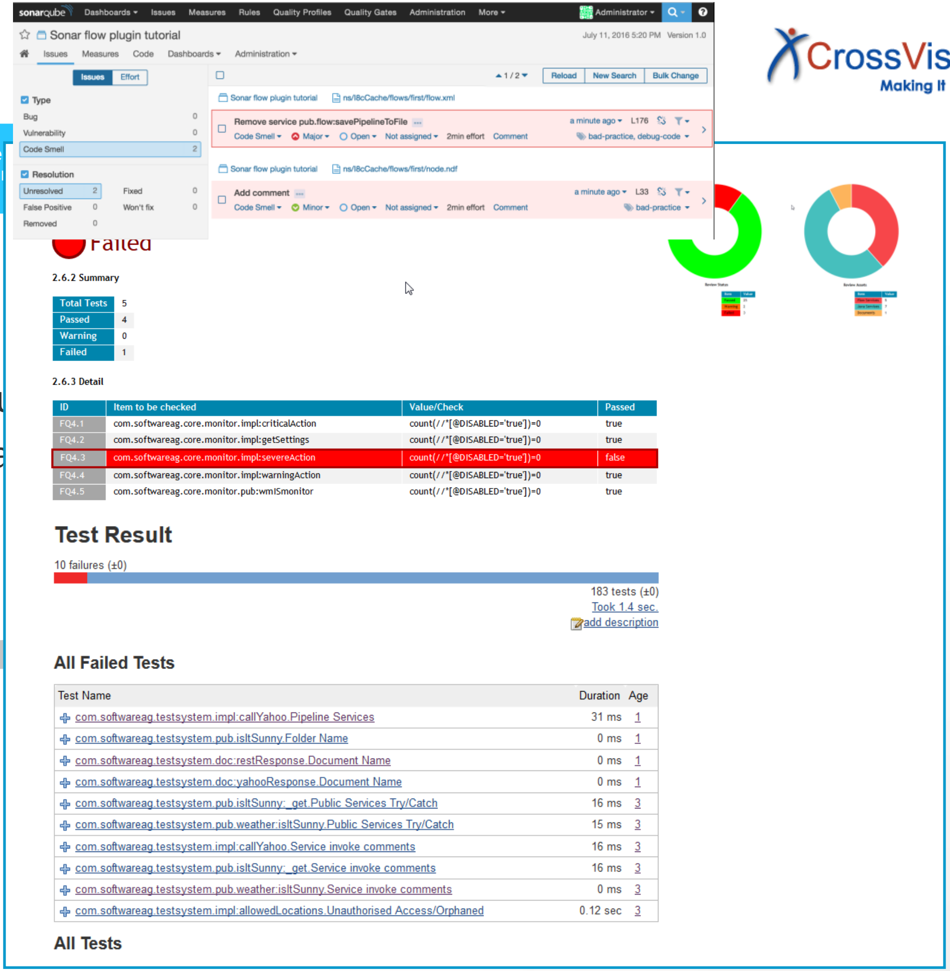
→ **Asset Build Environment**

→ **Project Automator**

→ **webMethods Deployer**

CONTINUOUS LIFECYCLE INSPECTION


De
Er
Build
Automated Quality Processes
Continu
Integra
Development Time Quality Processes
Deployment Processes
Check-in
Agile Development
Develop End to End, Interface, Performance Tests
Analysis Develop/Change Code/Configuration
Develop Unit Tests



Sonar flow plugin tutorial

July 11, 2016 5:20 PM Version 1.0

☑ Type

Bug	0
Vulnerability	0
Code Smell	2

☑ Resolution

Unresolved	2	Fixed	0
False Positive	0	Won't fix	0
Removed	0		

Sonar flow plugin tutorial ns/l8cCache/flows/first/flow.xml

□ Remove service pub.flow:savePipelineToFile — a minute ago L176
Code Smell · Major · Open · Not assigned · 2min effort · Comment · bad-practice, debug-code

Sonar flow plugin tutorial ns/l8cCache/flows/first/node.ndf

□ Add comment — a minute ago L33
Code Smell · Minor · Open · Not assigned · 2min effort · Comment · bad-practice

Failed

2.6.2 Summary

Total Tests	5
Passed	4
Warning	0
Failed	1

2.6.3 Detail

ID	Item to be checked	Value/Check	Passed
FQ4.1	com.softwareag.core.monitor.impl:criticalAction	count(//*[@DISABLED='true'])=0	true
FQ4.2	com.softwareag.core.monitor.impl:getSettings	count(//*[@DISABLED='true'])=0	true
FQ4.3	com.softwareag.core.monitor.impl:severeAction	count(//*[@DISABLED='true'])=0	false
FQ4.4	com.softwareag.core.monitor.impl:warningAction	count(//*[@DISABLED='true'])=0	true
FQ4.5	com.softwareag.core.monitor.pub:wmISmonitor	count(//*[@DISABLED='true'])=0	true

Test Result

10 failures (±0)

183 tests (±0)
Took 1.4 sec.
add description

All Failed Tests

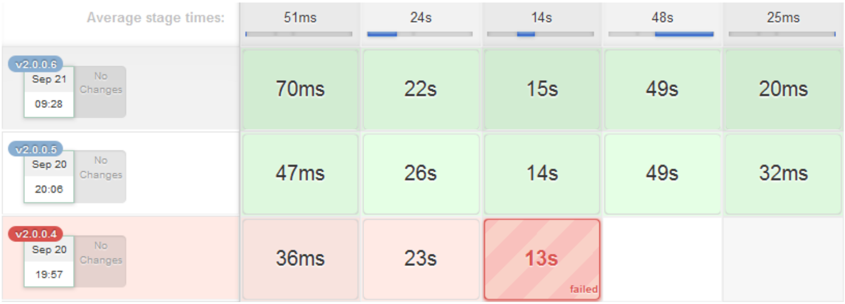
Test Name	Duration	Age
com.softwareag.testsystem.impl:callYahoo.Pipeline Services	31 ms	1
com.softwareag.testsystem.pub.isItSunny.Folder Name	0 ms	1
com.softwareag.testsystem.doc:restResponse.Document Name	0 ms	1
com.softwareag.testsystem.doc:yahooResponse.Document Name	0 ms	1
com.softwareag.testsystem.pub.isItSunny:_get.Public Services Try/Catch	16 ms	3
com.softwareag.testsystem.pub.weather:isItSunny.Public Services Try/Catch	15 ms	3
com.softwareag.testsystem.impl:callYahoo.Service invoke comments	16 ms	3
com.softwareag.testsystem.pub.isItSunny:_get.Service invoke comments	16 ms	3
com.softwareag.testsystem.pub.weather:isItSunny.Service invoke comments	0 ms	3
com.softwareag.testsystem.impl:allowedLocations.Unauthorised Access/Orphaned	0.12 sec	3

All Tests

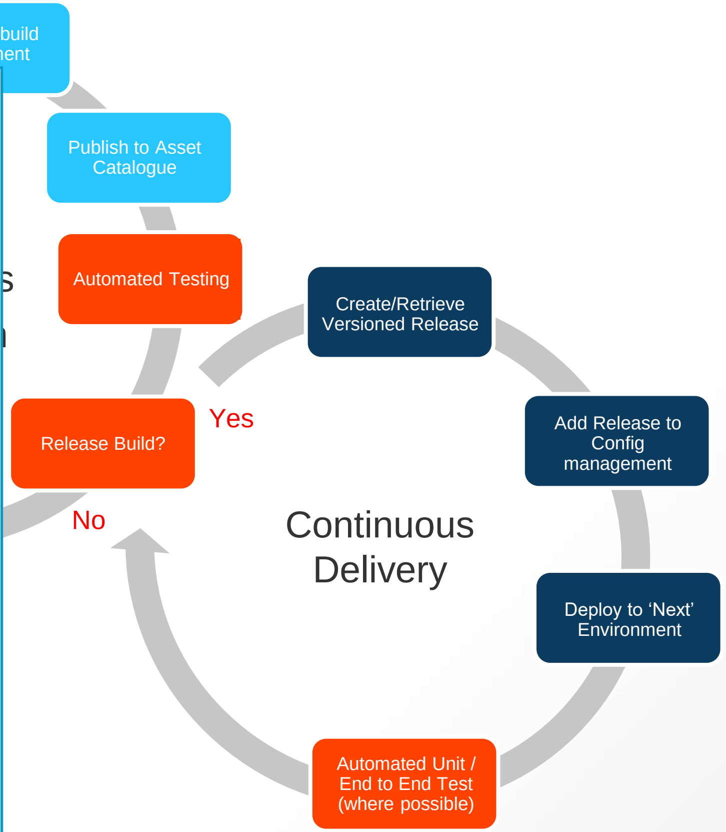
CrossVista Making It Easy

software AG

CONTINUOUS LIFECYCLE
CONTINUOUS DELIVERY

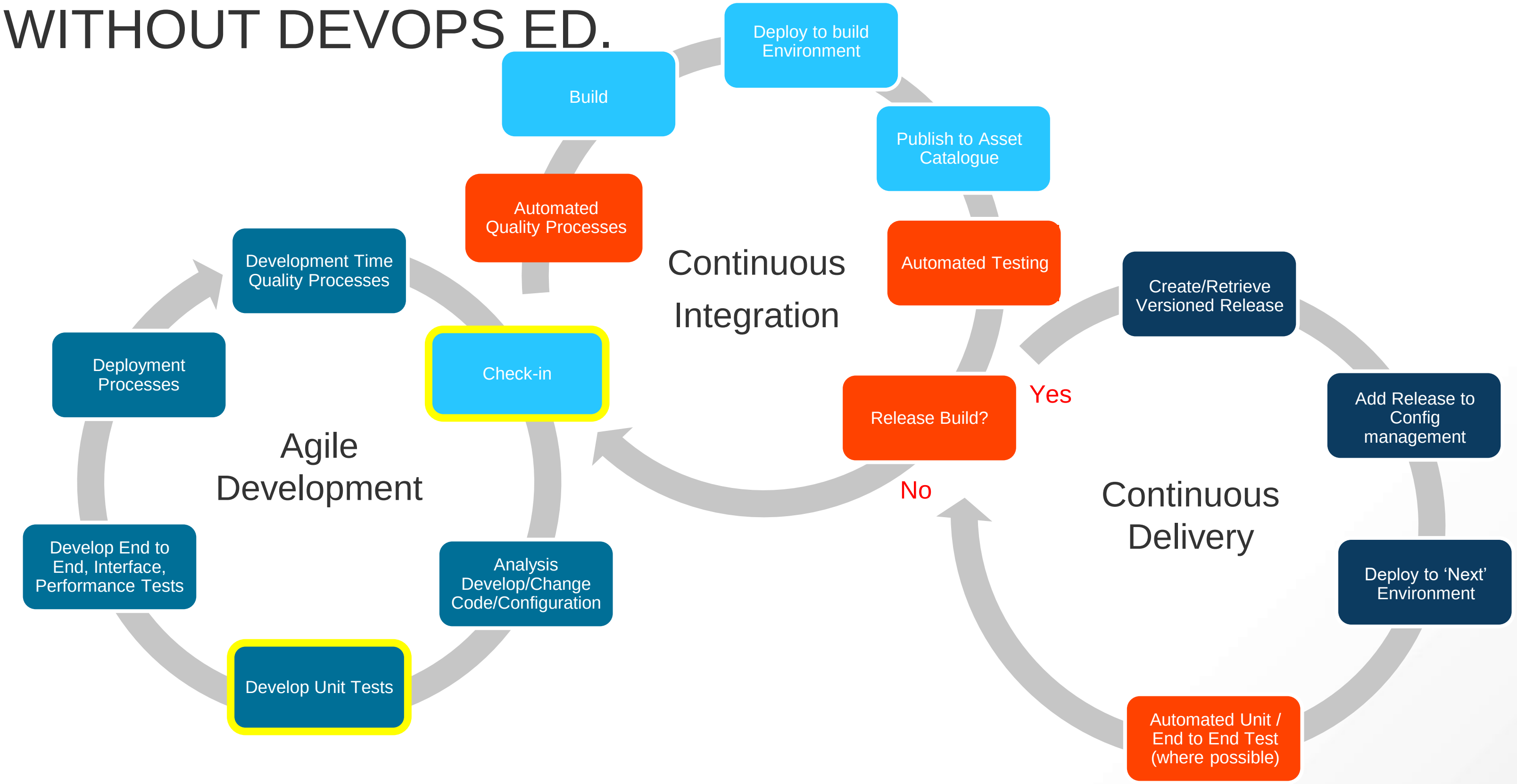


Average stage times:	51ms	24s	14s	48s	25ms
v2.0.0.6 Sep 21 09:28 No Changes	70ms	22s	15s	49s	20ms
v2.0.0.5 Sep 20 20:06 No Changes	47ms	26s	14s	49s	32ms
v2.0.0.4 Sep 20 19:57 No Changes	36ms	23s	13s failed		

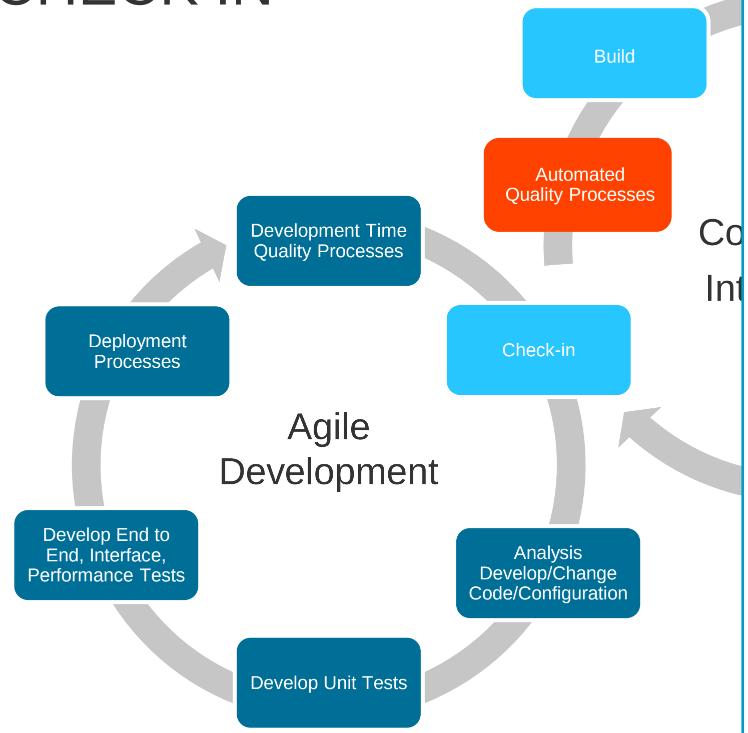

Deploy to build Environment
Publish to Asset Catalogue
Automated Testing
Create/Retrieve Versioned Release
Add Release to Config management
Release Build?
Yes
No
Continuous Delivery
Deploy to 'Next' Environment
Automated Unit / End to End Test (where possible)

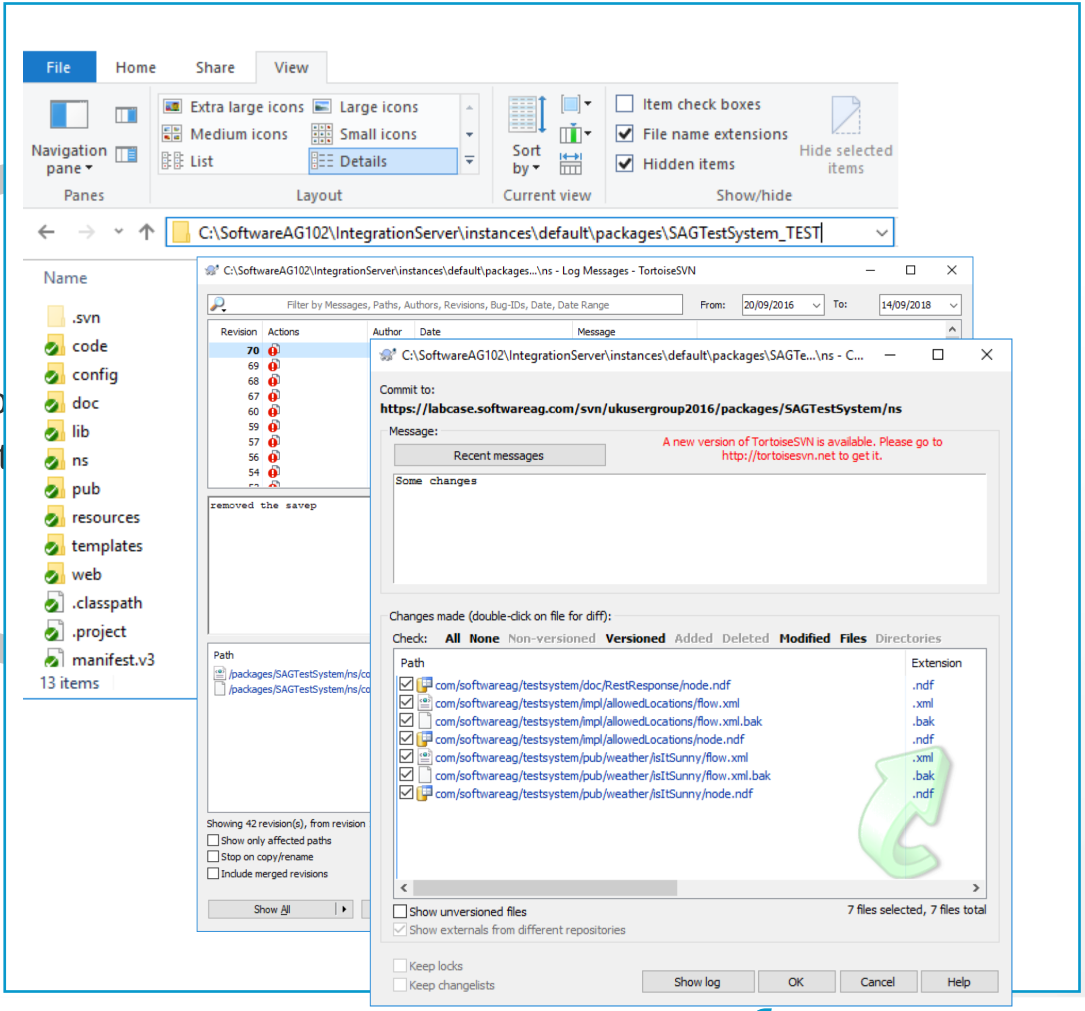
WITHOUT THE DEVOPS EDITION

CONTINUOUS LIFECYCLE
WITHOUT DEVOPS ED.


Build
Deploy to build Environment
Publish to Asset Catalogue
Automated Quality Processes
Automated Testing
Continuous Integration
Development Time Quality Processes
Check-in
Deployment Processes
Agile Development
Release Build?
Yes
No
Create/Retrieve Versioned Release
Add Release to Config management
Continuous Delivery
Deploy to 'Next' Environment
Develop End to End, Interface, Performance Tests
Analysis Develop/Change Code/Configuration
Develop Unit Tests
Automated Unit / End to End Test (where possible)

CONTINUOUS LIFECYCLE
CHECK IN


Build
Automated Quality Processes
Development Time Quality Processes
Co
Int
Deployment Processes
Check-in
Agile Development
Develop End to End, Interface, Performance Tests
Analysis Develop/Change Code/Configuration
Develop Unit Tests



□ Item check boxes
☑ File name extensions
☑ Hidden items

C:\SoftwareAG102\IntegrationServer\instances\default\packages\SAGTestSystem_TEST

Name
.svn
code
config
doc
lib
ns
pub
resources
templates
web
.classpath
.project
manifest.v3

13 items

C:\SoftwareAG102\IntegrationServer\instances\default\packages...\ns - Log Messages - TortoiseSVN

Filter by Messages, Paths, Authors, Revisions, Bug-IDs, Date, Date Range From: 20/09/2016 To: 14/09/2018

Revision	Actions	Author	Date	Message
70				
69				
68				
67				
60				
59				
57				
56				
54				

```
removed the savep
```

Path
/packages/SAGTestSystem/ns/co
/packages/SAGTestSystem/ns/co

Showing 42 revision(s), from revision

□ Show only affected paths
□ Stop on copy/rename
□ Include merged revisions

Show All

C:\SoftwareAG102\IntegrationServer\instances\default\packages\SAGTe...\ns - C...

Commit to:
https://labcase.softwareag.com/svn/ukusergroup2016/packages/SAGTestSystem/ns

Message:

Recent messages

A new version of TortoiseSVN is available. Please go to http://tortoisesvn.net to get it.

```
Some changes
```

Changes made (double-click on file for diff):

Check: **All** **None** Non-versioned **Versioned** Added Deleted **Modified** **Files** Directories

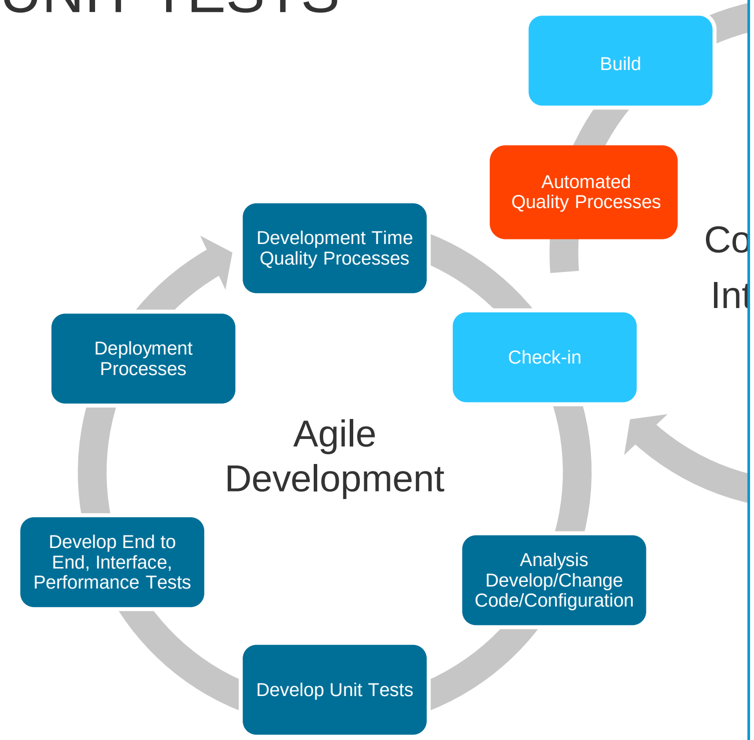
	Path	Extension
☑	com/softwareag/testsystem/doc/RestResponse/node.ndf	.ndf
☑	com/softwareag/testsystem/impl/allowedLocations/flow.xml	.xml
☑	com/softwareag/testsystem/impl/allowedLocations/flow.xml.bak	.bak
☑	com/softwareag/testsystem/impl/allowedLocations/node.ndf	.ndf
☑	com/softwareag/testsystem/pub/weather/isItSunny/flow.xml	.xml
☑	com/softwareag/testsystem/pub/weather/isItSunny/flow.xml.bak	.bak
☑	com/softwareag/testsystem/pub/weather/isItSunny/node.ndf	.ndf

□ Show unversioned files
7 files selected, 7 files total
☑ Show externals from different repositories

□ Keep locks
□ Keep changelists

Show log OK Cancel Help

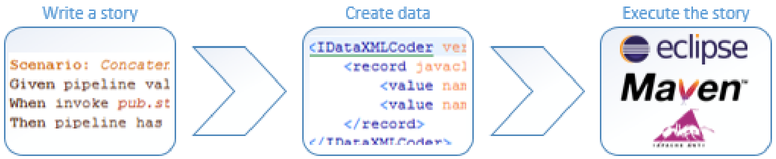
CONTINUOUS LIFECYCLE
UNIT TESTS


Build
Automated Quality Processes
Development Time Quality Processes
Co
Int
Deployment Processes
Check-in
Agile Development
Develop End to End, Interface, Performance Tests
Analysis Develop/Change Code/Configuration
Develop Unit Tests

JUnit 5

jbehave

wm-jbehave



```
Scenario: Concater
Given pipeline val
When invoke pub.st
Then pipeline has
```

```
<IDataXMLCoder ver
  <record javacl
    <value nam
    <value nam
  </record>
</IDataXMLCoder>
```

DEVOPS - DEVELOPMENT
SOURCE CONTROL – WHAT IS YOUR STRATEGY?

SVN

- Developers commit to trunk, day to day changes in trunk
- Trunk is branched and tagged
- Developers work in the release branch and trunk
- Branch tagged and released when ready
- Branch maintained over time.
- Feature Branches

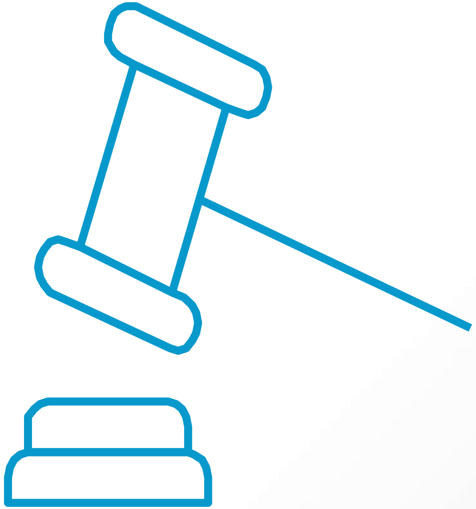
Commit Frequently!

GIT

- Use feature branches (from master) for new features and bug fixes
  - Merge into master using pull requests
    - Reviewed before merge
    - Master branch is high quality/up to date
    - Consistent naming by convention, e.g.
      - users/username/description
      - features/feature-name
- Use release branches:
  - Appropriately named, e.g.
    - /release/20
  - Create bugfix/hotfix branches from release branch, e.g.
    - /bugfix/description
    - hotfix/description

DEVOPS - DEVELOPMENT
DEFINITION OF DONE

- In an AGILE methodology, your definition of done is not just ‘committing the code’. It should include:
  - Creation of suitable Unit Tests
  - Creation of interface tests
  - Creation of end to end tests
  - Creation of performance tests
  - Any deployment/migration scripts
  - …



DEVOPS - DEVELOPMENT
TESTING

- Extend the tests for every feature
- Automate all tests to avoid regression
  - Units
  - Component / Interface
  - End to End
  - Performance
- Using mocking features to avoid the need for the end systems in all environments
- Test all environments as close to production as possible, using mocks if needed!

DEVOPS - DEVELOPMENT
QUALITY IS EVERYONE'S RESPONSIBILITY

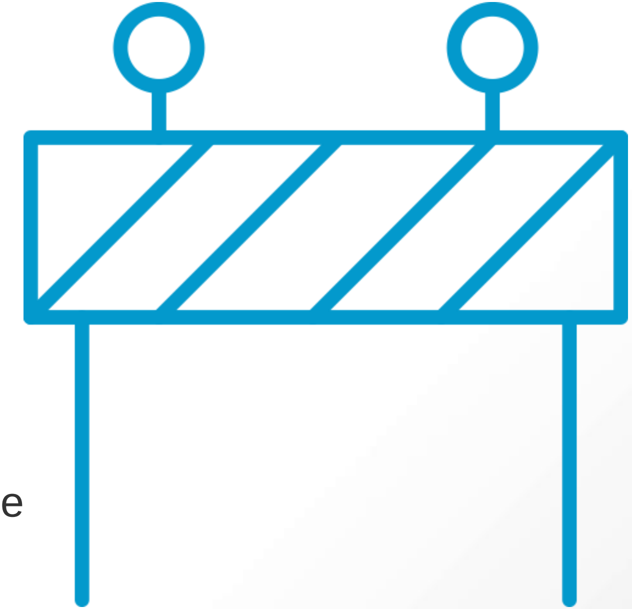
- Just because it works, doesn’t make it right!
- Quality is the responsibility of everyone, not just the QA/reviewer.
- Ensure appropriate standards are in place, and understood/followed
- Implement a continuous inspection process to automate the bulk of the review process in an objective manner automatically, versus a subjective review.

80%

Of the lifetime cost of a software product is spent on maintenance, and maintenance costs have a high variability depending on quality

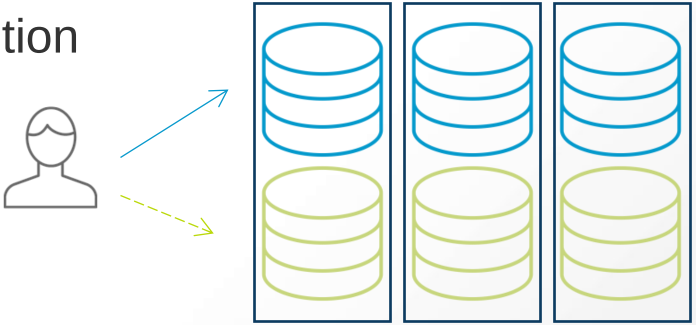
DEVOPS – CONTINUOUS DELIVERY
AUTOMATED PROVISION

- A ‘green field’ deployment guarantees a repeatable process
- Provision a new ‘platform’
  - **Don’t ever assume anything exists!**
  - Automated provision via suitable tools (e.g. puppet/chef/ansible/salt/etc)…
  - Use command central templates to provision the software
  - More recently – You can consider just building new Docker containers
  - If you can’t provision, make sure you ‘tear down’ and level the ‘brown field’ before starting again!



DEVOPS – CONTINUOUS DELIVERY
MICRO OR MONOLITH

- Integration Server
  - Package is a deployment construct
  - Deploy as granularly as possible to avoid the monolith
    - Micro-services approach removes (minimizes) dependencies
    - Manage dependencies at deploy time – it's a deployment issue
    - Use CentraSite Active SOA to understand dependencies if they are complex
- Use a Blue/Green deployment approach for continued operation with no/minimal disruption
- Deploy small
- Deploy often
- Deploy continuously
- Remember 'Feature Toggles!'



DEVOPS – CONTINUOUS DELIVERY
STORE YOUR BINARY ASSETS

- Asset Repository
  - Somewhere to hold your binary assets
    - Build Versions/History
    - Dependencies
  - e.g.
    - Sonatype Nexus
    - JFrog Artifactory
    - Docker Registry
- Deploy from binary into next environment, not from source to guarantee the same binary code in subsequent environments
- When using Containers use the same ‘container’ binary across all environments

DEVOPS – CONTINUOUS DELIVERY
DEV-SEC-OPS

- Security should be part of the same continuous processes, not an after thought!
- Automate security testing like you would unit tests and end to end tests to avoid common security pitfalls.