EDG’s Development Lifecycle

A summary of testing and development recommendations from the tac

Scope

This presentation is meant to be a unified vision for EDG's future development and testing strategy

This presentation summarizes the findings from:

- The White Paper: "Internal Review of Current EDG Testing Practices"
- The White Paper: "Independent Review of EDG Test Programs"
- Recommendations from the Tiger Team looking at EDG tradecraft
- Recommendations from recent EDG management offsite

In addition it will show EDG's vision for:

- The automated test suite, DART
- The software development and collaboration tool suite by Atlassian

Apologies Up Front

Reasons the pitchforks are about to come out

- Introduces new process
- Increase in documentation
- More work up front
- It is ***change***

Reasons the pitchforks should go away

- Less work in the long run
- Assists junior developers in learning EDG tradecraft
- Protects our tools
- Green check marks are wonderful

Vocabulary

There are a few items we will be mentioning by name often:

Atlassian Products – (Replaced TeamForge)

- *Confluence* – Knowledge Management
- *Jira* – Project Management and Issue Tracking
- *Bamboo* – Continuous Integration
- *Stash* – Source Control using Git

Other Products

- DART – Automated testing suite replaced ERGOSTAR
- Git – Distributed version control system replaced SVN
- ServiceNow – Replacement for IMIS
- Tool Pedigree database (TPD) – Database being compiled based upon recommendations of all Tiger Teams

Git

Git should be used for all projects

Two repositories for each project

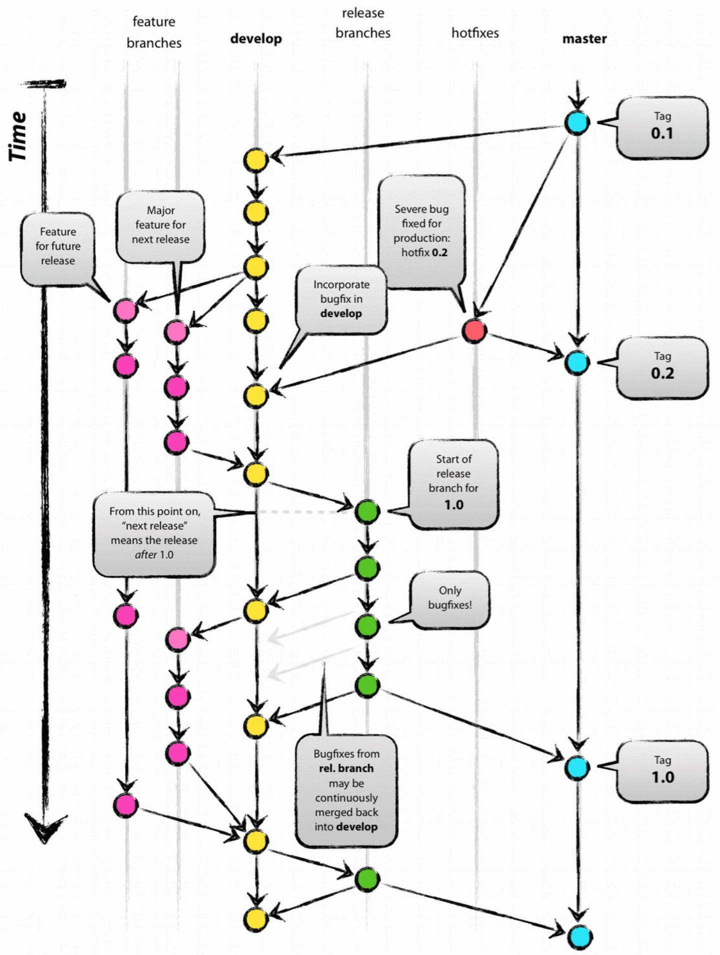
- One for code (EDG only)
- One for tests (shared with COG)

Two long lived branches

- ***Master*** – Official releases
- ***Develop*** – Stable branch

Three short lived branch types

- ***Feature*** – development done here
- ***Release*** – code freeze for testing
- ***Hotfix*** – immediate fixes to master



feature branches
develop
release branches
hotfixes
master
Time
Tag 0.1
Major feature for next release
Feature for future release
Severe bug fixed for production: hotfix 0.2
Incorporate bugfix in develop
Tag 0.2
Start of release branch for 1.0
From this point on, "next release" means the release after 1.0
Only bugfixes!
Bugfixes from rel. branch may be continuously merged back into develop
Tag 1.0

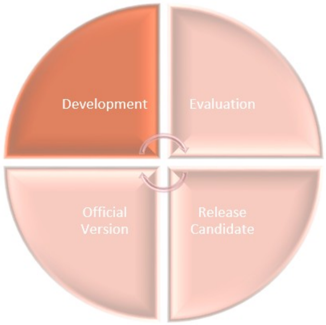
Four Phases of a Project

The wall between development and evaluation should be extremely low

Migration between the four phases should be controlled via Bamboo to ensure what code is where

A memorandum of understanding between EDG and its customers must be crafted so that customers can be trusted with evaluation copies

Development



During this phase the development team is in its most isolated state, and is focused purely on the requirements on hand

Tasks during this phase:

- Development of code base (***by a team***)
- Creation of unit tests
- Code reviews
- Documentation of capabilities in TPD

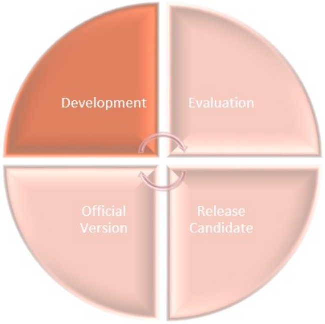
Development during this phase shall:

- Be done in feature branches
- Be focused on one thing at a time

AED: Adopt the ‘Git Flow‘ workflow described previously

ESD: Recommend contracts follow a similar workflow (or at least vocabulary)

The Development Te



The development team should be broken into three categories

Development

- Lead Developer – Project lead
- Additional Developers (always at least one)

Testing

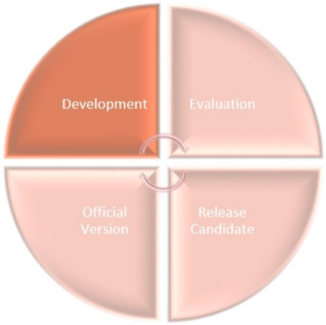
- Lead Tester – Doesn’t answer to Project lead, keeps the ‘I’ in IV&V
- Additional Testers (if necessary)

Project Support

- Branch Chief
- Systems Integrator

AED: Development will be conducted in teams; testers shall be equal partners in development so they can fully understand the operation

Unit Testing



“The primary goal of unit testing is to take the smallest piece of testable software in the application, isolate it from the remainder of the code, and determine whether it behaves exactly as you expect”

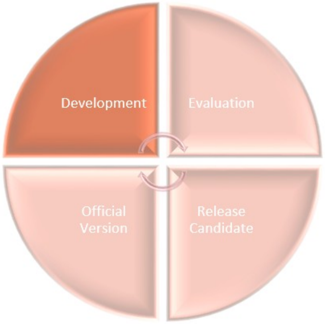
Unit tests should test:

- *The Good* - expected values
- *The Bad* - extreme, but acceptable values
- *The Ugly* - out of range values

Unit testing is more work up front, but saves time and produces better code

AED: Developers should incorporate unit testing during development

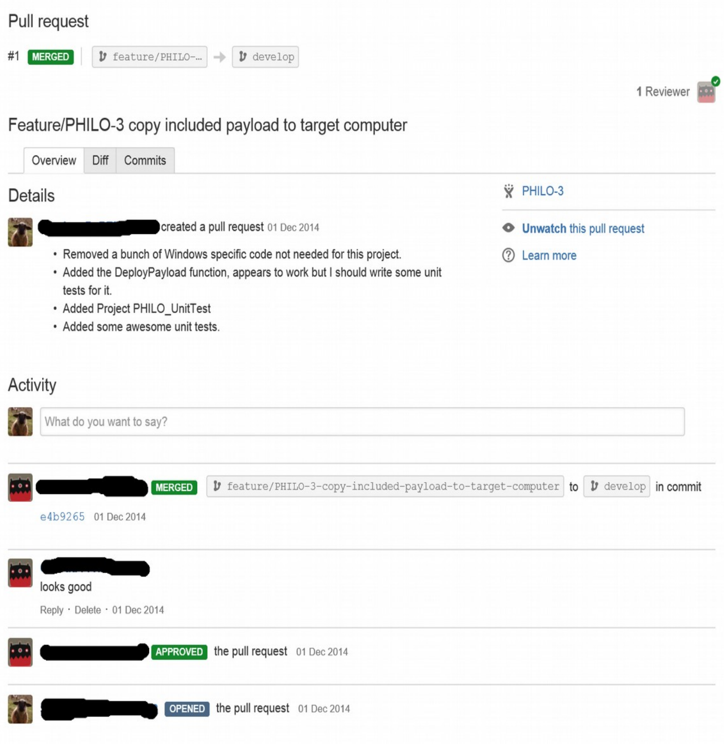
Code Reviews via Stash



Conduct frequent code reviews earlier while developing

What is a code review / pull request?

- Focus on incorporation of single development task into source base
- Review new changes to source base by other team members
- Use peer review to exercise EDG best practices
- Document new capabilities in TPD
- Tame the chaos of the merge process



Pull request

#1 MERGED feature/PHILO-... → develop

1 Reviewer

Feature/PHILO-3 copy included payload to target computer

Details

[redacted] created a pull request 01 Dec 2014

- Removed a bunch of Windows specific code not needed for this project.
- Added the DeployPayload function, appears to work but I should write some unit tests for it.
- Added Project PHILO_UnitTest
- Added some awesome unit tests.

Activity

[redacted] MERGED feature/PHILO-3-copy-included-payload-to-target-computer to develop in commit e4b9265 01 Dec 2014

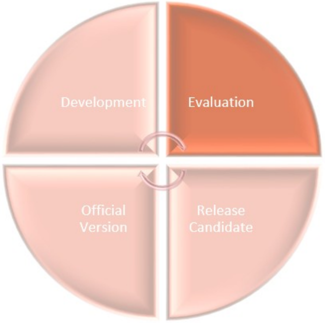
[redacted]
looks good
Reply · Delete · 01 Dec 2014

[redacted] APPROVED the pull request 01 Dec 2014

[redacted] OPENED the pull request 01 Dec 2014

AED: Code reviews should be conducted via Stash before merging into 'develop'
ESD: COTRs should request information to fill in the TPD

Evaluation



During this phase the development team interacts with the testing team and the customers to ensure the product being created fits with the customers' vision

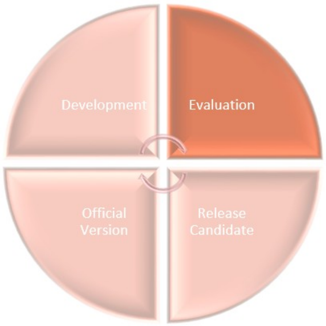
Tasks during this phase:

- Creation of integration test scripts or an integration test plan
- User and developer guide creation
- Demonstration of current functionality
- Refinement of requirements

Evaluation versions are never deployable. If there is a need, a new requirement can be generated

AED and ESD: Pass evaluation versions to the customer earlier to receive feedback often; solicit customer feedback often

Issue Tracking via JIF

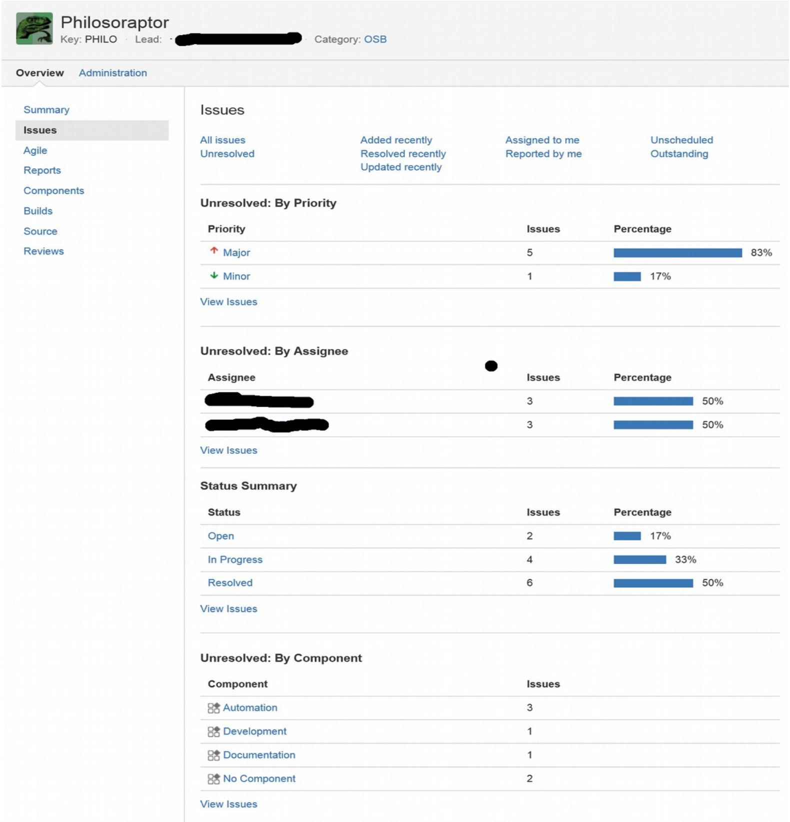


Breaking down requirements into basic tasks makes time estimation more realistic

- Batch tasks into shorter development iterations (Agile)
- Assign tasks to development team members
- Track discovered bugs with new tasks
- Capture reported issues from Customer

Works alongside ServiceNow

- ServiceNow tracks official requirements
- Jira tracks work done to meet those requirements through to delivery



Philosoraptor
Key: PHILO · Lead: [redacted] Category: OSB

Issues

All issues Unresolved	Added recently Resolved recently Updated recently	Assigned to me Reported by me	Unscheduled Outstanding

Unresolved: By Priority

Priority	Issues	Percentage
↑ Major	5	83%
↓ Minor	1	17%

View Issues

Unresolved: By Assignee

Assignee	Issues	Percentage
[redacted]	3	50%
[redacted]	3	50%

View Issues

Status Summary

Status	Issues	Percentage
Open	2	17%
In Progress	4	33%
Resolved	6	50%

View Issues

Unresolved: By Component

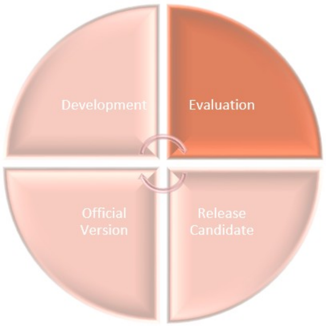
Component	Issues
Automation	3
Development	1
Documentation	1
No Component	2

View Issues

AED: Use JIRA during development to track progress

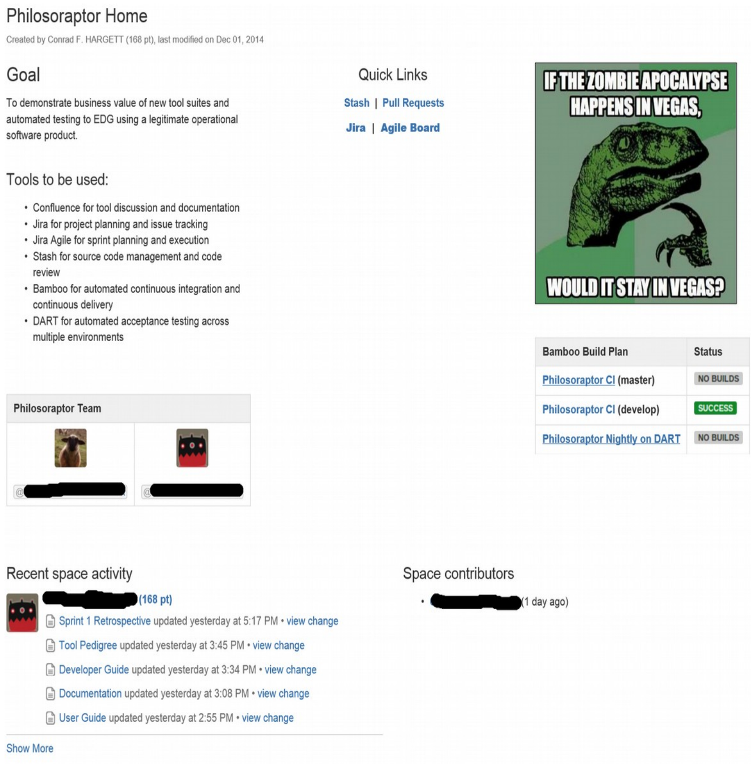
AED and ESD: Use JIRA as an official way for feedback on evaluation versions

Documentation via Confluence



Improve documentation by using Confluence
- Centralize all project knowledge in one location (external to developers' brains!)
- Export Confluence pages to PDF format

Create new project standards for documentation
- User Guide (formal and informal)
- Developer Guide (knowledge retention)
- Tool Pedigree (human readable)



Philosoraptor Home

Created by Conrad F. HARGETT (168 pt), last modified on Dec 01, 2014

Goal

To demonstrate business value of new tool suites and automated testing to EDG using a legitimate operational software product.

Quick Links

Stash | Pull Requests

Jira | Agile Board

Tools to be used:

- Confluence for tool discussion and documentation
- Jira for project planning and issue tracking
- Jira Agile for sprint planning and execution
- Stash for source code management and code review
- Bamboo for automated continuous integration and continuous delivery
- DART for automated acceptance testing across multiple environments

Bamboo Build Plan	Status
Philosoraptor CI (master)	NO BUILDS
Philosoraptor CI (develop)	SUCCESS
Philosoraptor Nightly on DART	NO BUILDS

Philosoraptor Team	
@[redacted]	@[redacted]

Recent space activity

[redacted] (168 pt)
- Sprint 1 Retrospective updated yesterday at 5:17 PM • view change
- Tool Pedigree updated yesterday at 3:45 PM • view change
- Developer Guide updated yesterday at 3:34 PM • view change
- Documentation updated yesterday at 3:08 PM • view change
- User Guide updated yesterday at 2:55 PM • view change

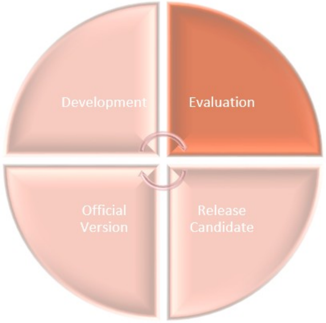
Show More

Space contributors

- [redacted] (1 day ago)

AED: Collaborate on project documentation, increase project visibility to ESD

ESD: Import documentation to Confluence, increase project visibility to AED



Quality Assurance

The decision to cut a release candidate should be a decision between the development team, the testing team, and the branch chief

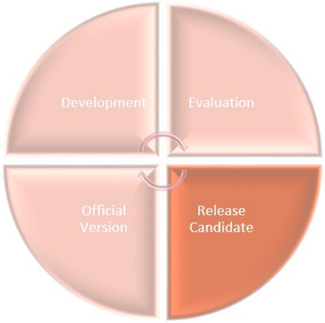
Questions that need to be answered:

- Is there sufficient unit test coverage
- Are there automated integration tests in place
- Is there a test plan in place
- What type of regression testing is needed for this version

Keep in mind that after this step a tool could potentially be deployed with no further changes

AED: Establish a practice of having a sit-down with the project lead, branch chief and testing lead before cutting a release candidate

Release Candidate



A release candidate signifies a code freeze for all requirements and only bug fixes should be introduced to the codebase during this phase

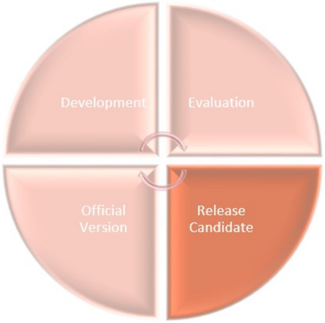
Tasks during this phase:

- Tier One testing
- Acceptance testing by IV&V on bare metal (if required)
- Basic Forensic Testing
- Finalization of all documentation

Operationally deployable only with EDG COPs approval

AED and ESD: Establish a MOU stating that evaluation versions are never to be deployed, and release candidates can only be deployed with COPS approval

Tier One Testing



To alleviate pressure on QRC testing EDG should establish tier one and tier two testing

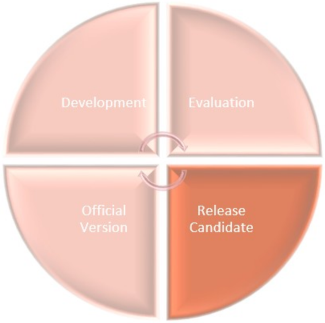
Tier One Testing:

- Testing requirements for the specific environment in mind
- Basic forensics that all tools need to pass

A tool will remain a release candidate until tier one testing is complete

It is up to EDG to ensure that tier one testing is inclusive enough to protect our tools

AED and ESD: Establish a ‘Chinese Menu’ of tests so that our customers can choose what is an immediate test and what is a test for later



The ERB

The engineering review board's role should be expanded to ensure a product meets requirements and is properly tested

Current role of the ERB:

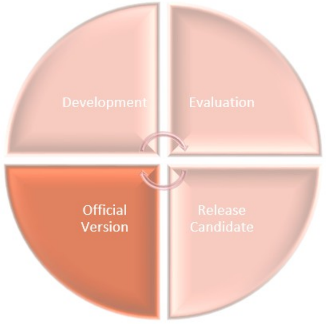
- Ensure the tool meets requirements
- Certify tool for delivery

Expanded role of the ERB:

- Live demo of the tool
- Review tier one testing
- Review automated test coverage
- Ensure TPD has been filled in
- Review of documentation

AED and ESD: The ERB's functionality should be expanded to review the above bullets

Official Version



An official version means that build can be deployed by the customer with no further interaction with EDG. Development doesn’t necessarily stop as new requirements could already be queued up.

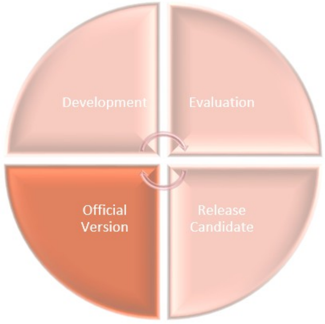
Tasks during this phase:

- Operational use by the customer (hopefully)
- Tier two testing
- Long term regression testing (PSP and patch)
- Generation of new requirements

The tool can be deployed operationally without coordination with EDG

AED and ESD: Rethink requirements so smaller official versions can be delivered more often. I.e. if a tool does three things no reason to hold up two of them

Tier Two Testing



Tier two testing is meant to free up resources for QRC testing. Tests that fall under tier two are the “nice to haves” that bloat testing requirements and delay delivery of a tool

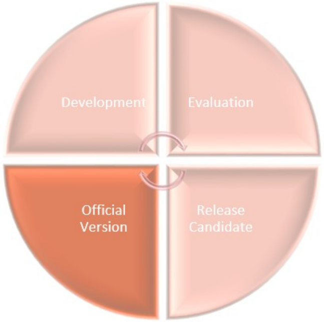
Tier Two Testing Includes:

- Other OS’s the tool may be deployed on
- Other configurations the tool may be used with
- Deeper forensic testing

Issues found here don’t mean a new RC, they are submitted as a new requirement or DR.

AED and ESD: This is a continuation of the ‘Chinese Menu’ for testing and EDG should push for the nice to haves be moved to tier two

Version Control Via Bamboo



Practice continuous integration (CI) to automate building processes and maintain consistency in source base

What is CI?

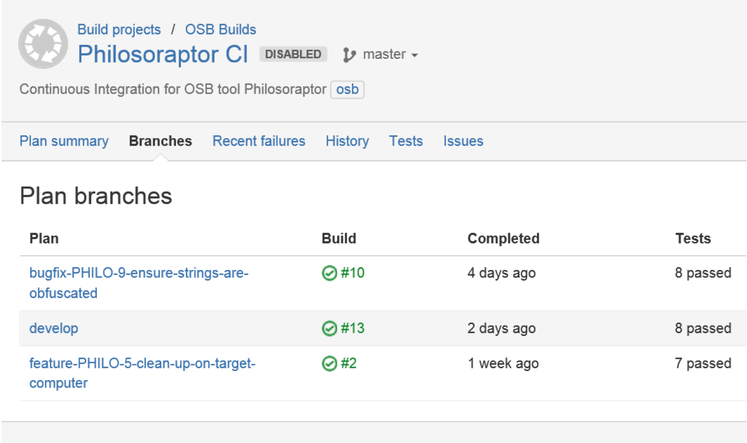
- Compile software on a standardized server (not personal workstations)
- Manage different types of releases with build plans
- Run automated tests to verify correct functionality ASAP (Fail Fast)
- Track test execution results
- All the above ***every time*** the source base changes

Customize Bamboo to automate:

- DART testing
- Electronic Delivery

AED: Create build plans for Release, Release Candidate, Evaluation Copy

ESD: Utilize Bamboo's integration with Dart for continuous testing

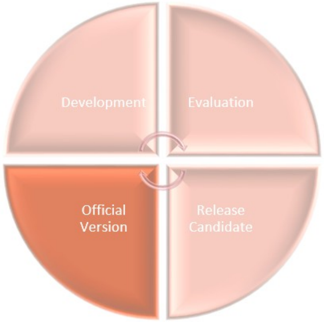


Continuous Integration for OSB tool Philosoraptor osb

Plan branches

Plan	Build	Completed	Tests
bugfix-PHILO-9-ensure-strings-are-obfuscated	✓ #10	4 days ago	8 passed
develop	✓ #13	2 days ago	8 passed
feature-PHILO-5-clean-up-on-target-computer	✓ #2	1 week ago	7 passed

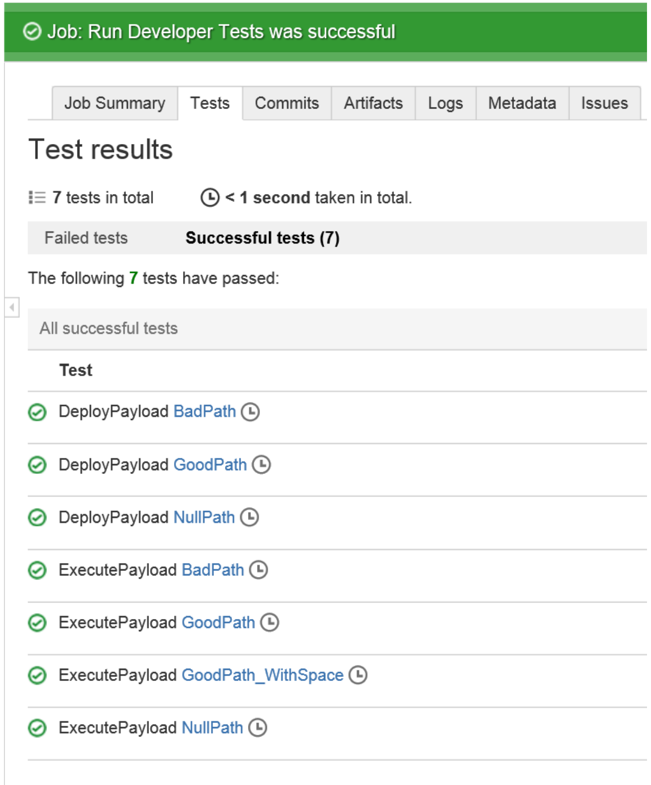
Continuous Testing Via DART



Automate testing across multiple environments using DART

- When applicable (rule of eight)
- Execute unit/developer tests to verify functionality
- Execute acceptance tests to verify requirements
- Collect all test results into Bamboo for easy analysis

A shared repository of dart tests shall be created to make testing easier (the EDG leafbag)



Job: Run Developer Tests was successful

Test results

7 tests in total   < 1 second taken in total.

Failed tests   **Successful tests (7)**

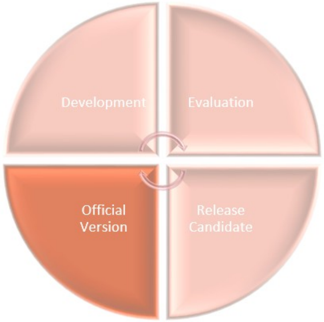
The following 7 tests have passed:

All successful tests

Test
DeployPayload BadPath
DeployPayload GoodPath
DeployPayload NullPath
ExecutePayload BadPath
ExecutePayload GoodPath
ExecutePayload GoodPath_WithSpace
ExecutePayload NullPath

AED: Create repeatable processes for automating tests via DART
ESD: When possible recommend DART to development contracts

New Requirements



The tracking and creating of requirements remains the same for the most part

Current process:

- Draft requirement is generated
- Accepted by the ERB
- Tracked in IMIS (Soon to be ServiceNow)

Recommended changes:

- User stories should be captured in Jira

AED: Ensure user stories are documented in Jira for the team to reference

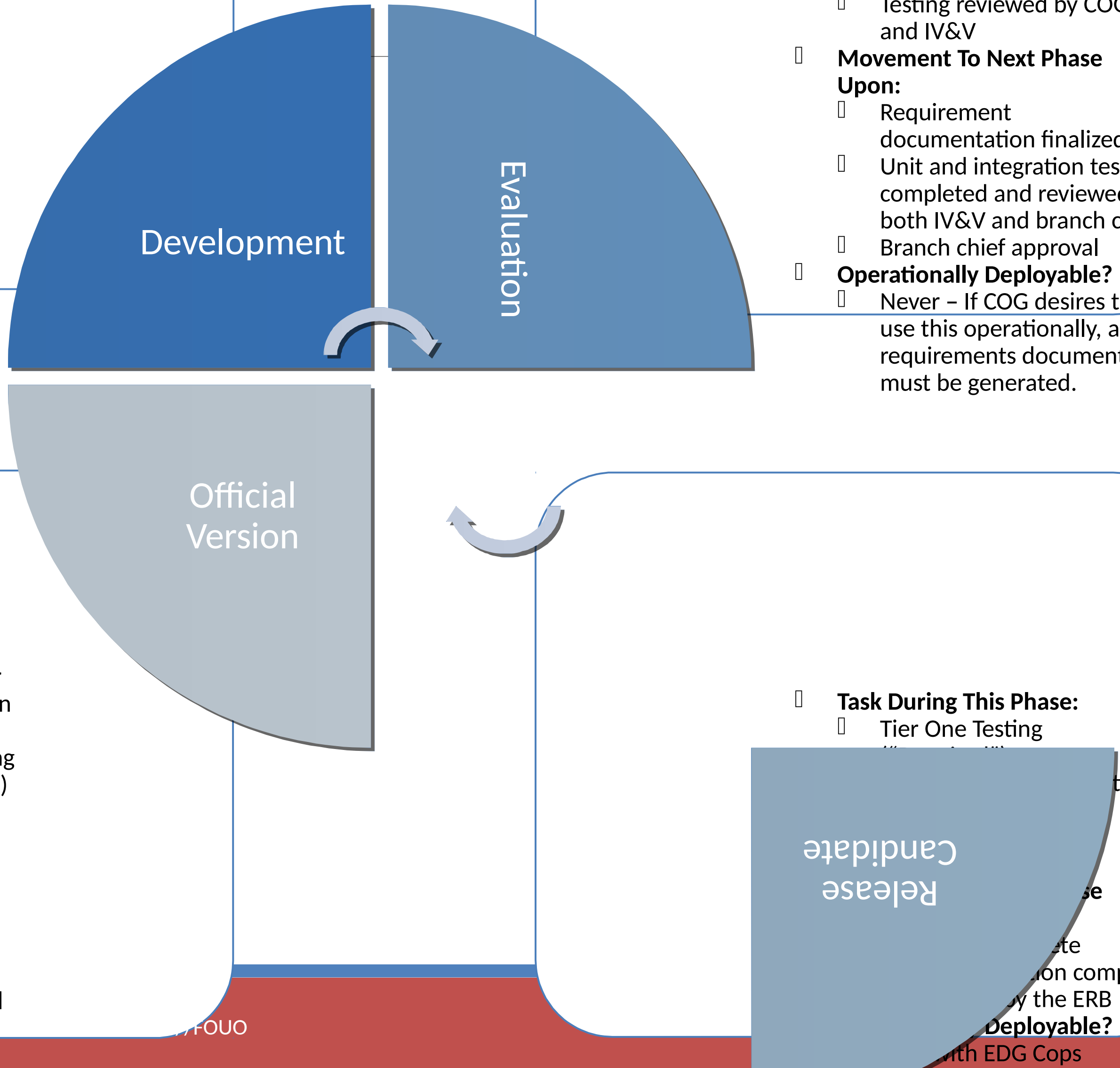
ESD: Ensure user stories passed along to the development team

Next Steps

- Incorporate final comments from branch chiefs
- Get official buy in from front office
- Use these four projects as examples
  - CASCADE
  - IOS Team
  - IMPROVISE
  - PHILOSORAPTOR
- Get a branch into this cycle to use as an example
  - Recommend OSB
- MOU with COG for the development cycle
- Brief EDG All-Hands with overview of this
- Brief branches two at a time to ensure maximum engagement

- **Task During This Phase:**
  - Development of code base
  - Unit tests written
  - Code reviews completed
  - Documentation of capabilities
- **Movement To Next Phase Upon:**
  - Completion of Sprint
  - Satisfactory completion of unit tests
  - Mandatory code review based upon branch best practices
- **Operationally Deployable?**
  - Never

- **Task During This Phase:**
  - Revaluation of tool and requirements by operators
  - Automated integration tests written by IV&V with the assistance of AED and previous testing scripts
  - Code reviews completed
  - Testing reviewed by COG and IV&V
- **Movement To Next Phase Upon:**
  - Requirement documentation finalized
  - Unit and integration testing completed and reviewed by both IV&V and branch chief
  - Branch chief approval
- **Operationally Deployable?**
  - Never – If COG desires to use this operationally, a new requirements document must be generated.



- **Task During This Phase:**
  - Operation use by customer
  - Operational testing done on a continual basis
  - Long term regression testing
  - Tier Two Testing (“Desired”)
- **Movement To Next Phase Upon:**
  - Creation of DR
  - Creation of new requirements
  - Acceptance by the ERB
- **Operationally Deployable?**
  - Deployable with traditional approvals

- **Task During This Phase:**
  - Tier One Testing

ting

se

ete

ion complete

y the ERB

Deployable?

with EDG Cops