

# **Process - DevOps Testing Process**

Aptean Ltd  
Copyright © 2011-2026.

## Contents

<b>1 Process - DevOps Testing Process.....</b>	<b>1</b>
1.1 Finishing Development.....	1
1.2 Testing.....	1
1.3 Passing Testing.....	1
1.4 Failing Testing.....	1
1.5 Raising DevOps Bugs.....	2
1.6 Priority, Severity, Bug Type and Fault Cross-reference.....	2
1.7 Developer Updates.....	3
1.8 Releasing.....	4

# 1 Process - DevOps Testing Process

The intention of this process is to confirm the process that all should follow so that the reporting of rework reasons can be followed.

## 1.1 Finishing Development

R&D

- update Resolution Notes.
- assigns case to Development Manager.
- Set Sub state to *SCR Done*

Dev Manager

- Assign to release manager to be released

Release Manager

- release to 2ndary test environment.
- Assign to Dev Manager

Dev manager

- Assign to Tester.
- Set Sub-state to *On Test*.

## 1.2 Testing

Testing shall produce a POT/PO2T document in the shared directory  
 \\DGA1FS01OBS\\Test\_Logs\\System\\Customer\\DevOps.

The document shall be placed in a sub-folder named matching the description of the DevOps case being tested.

The document shall be named POT/PO2T followed by the same matching name of the case being tested.

The format is variable, but the appropriate template should be followed.

Template: [File:POT-SFNumber Description v1.1.dotx](#)

Tester:

- Set Sub-state to *Being Tested*

## 1.3 Passing Testing

- Set Sub state to *Ready for Rel.*
- Update Notes.
- Link in POT document
- Assign to Dev Manager

## 1.4 Failing Testing

What to do to the original case

- Set Sub state to *Test Failed*.
- Raise a new DevOps **Bug**.
- Update comments.
- Link in POT document
- Assign to Dev Manager



**Note:** When multiple bugs are found, raise a DevOps case for each bug. Bugs can be consolidated together if this makes sense and they can be done together.

## 1.5 Raising DevOps Bugs

Every error in testing shall be raised as a separate DevOps case:

- The Name shall describe the system, customer/DB and short description of the fault.
- The Description should be described and, if necessary, placed in a document and attached or linked to in the description.
  - ◆ Describe in detail or
  - ◆ Upload a DOCX file or
  - ◆ Add a link to the shared POT document in the shared filesystem.
- Any tags being used by the customer/R&D should be set
- Area - should be set to the system and product.
- Iteration - to the latest iteration for the R&D group.
- The Repro Steps should be noted in detail or linked to the document uploaded/linked.
- The customer must be set
- Priority - the priority of the fix required. Set as per the standards. See table below.
- Severity - the severity of the bug being raised. See table below.
- Bug Type - root cause analysis - see table below.
- Fault - root cause analysis - see table below. Where there are multiple bugs consolidated into one, choose the worst fault.
- Sub State - use the list of values, values defined below
- The parent of this case shall be the User Story being tested.
- Assigned to the development manager, Karthik or Carl.

## 1.6 Priority, Severity, Bug Type and Fault Cross-reference

Priority and Usage

Priority	Description
1	Must be completed now/next
2	Must be completed before the next release
3	Must be completed for final release
4	Nice to have, not essential to complete

Severity and Usage

Priority	Description
1 - Critical	Showstopper.
2 - High	May have a poor workaround. Must be completed
3 - Medium	Impacting the customer, but with workaround
4 - Low	Nice to have, UI issues.

Bug Type Usage

Bug Type	Description
Configuration	
Customization	
Deployment	
Documentation	
Enhancement	
Environment	Has the bug broken or been caused by an environment issue, such as interfacing, filesystem, etc?
Functional	If there is a function simply not working, choose this option
Invalid	
Legacy	If whilst testing functionality, you note that an existing function not related to the development has been broken, choose this.
Localization	
Missed Requirement	If you note that the specification is explicitly missing the requirement, choose this one.
Performance	Functionality works but is unacceptably poor.



Bug Type	Description
Regression	If whilst testing functionality, you note that an existing function related to the development has been broken, choose this.
Security	
Setup	
Third Party	
Usability	
User Interface	Functionality works but implementation of the UI is poor.
Fault Usage	
Fault	Description
Advice Provided to User	
Datafix	
Deployment	If a DevOps bug has been raised by R&D SOLELY for release of product changes, then the bug type should be set to this.
Design Constraint	Bug is as designed, but requirement is not fully met
Design Error	Bug is as designed, but not fulfilling requirement (FS Fault)
Development	Development issue
External Interface	Caused by external system
Hardware/Comms error	
Implementation	Caused by Implementation failure
Missing Function	Function completely missing.
No Fault Found	
Program Error	New bug
Program Error (Rework)	Bug directly with new work.
Revised Requirement	FS change
Unable to Replicate	
Sub-states and their usage	

Sub-state	Who	Description	Action
For Investigation	Tester/PS	Set initially when a bug is raised.	Assign to Dev Mgr
Investigating	R&D	When investigation begins	Assign to R&D
Being Programmed	R&D	Optional - if the development/fix is multiple days.	
SCR Done	R&D	Work complete. Assign back to development manager	Assign to Dev Mgr
On Test	Dev Mgr	When assigned to tester.	Assign to Tester
Being Tested	Tester	When being tested	
Test Fail	Tester	If the issue has failed.	Assign to Dev Mgr
Ready for Rel	Dev Mgr	When the code release is ready	Assign to Rel Mgr
Installed on TST	Rel Mgr	When the update is released to test	Update Hotfix in Release to patch/release version
Installed on PRD	Rel Mgr	When the update is released to production	Update Hotfix in Release to patch/release version
On-hold	Anyone	When the case is put on hold	<b>USE WITH CARE. ENSURE ALL MANAGERS ARE NOTIFIED.</b>

## 1.7 Developer Updates

R&D

- New Bug
  - ◆ Follow similar process for Finishing Development above re:
    - ◊ Sub State
    - Resolved Details
    - ◊ Update



- Bug Type
- Fault
- Comments
- ◊ Assign to Development Manager
- ◆ Existing DevOps Use Case (if assigned to R&D)
  - ◊ Update comments
  - ◊ Assign to Development Manager

 **Note:** If multiple bugs are being fixed by the same release (i.e. in the same program/package), then ALL bugs must be assigned back to the Development Manager and on to the tester. Furthermore, one of these bugs should be considered the master - select one. Add a "Related" link to the other bugs showing that they are related to the first DevOps bug. This will help inform the Development Manager, Release Manager and Tester as to what can be tested and fixed together.

Development Manager

- The original use case and updated bug(s) should be returned to the original tester.

## 1.8 Releasing

Once all associated bugs are fixed, retested and assigned back to the Development Manager, they can all be released to the Release manager for releasing to the next area (TST, PROD, etc) using the standard product-related release procedures.

See also:

- [Creating Release Notes](#)

