
DataArt

Introduction of Quality Assurance

Testing in a nutshell

DataArt

Competencies:

- Guarantee quality of a product
- Finding defects (bugs)
- Preventing bugs
- Work with docs

Couple of words about documentation

DataArt

1. Test cases;
2. Test plan;
3. Check list;
4. Test suite;
5. Bug reports.

Test plan

DataArt

Test plan – main document for application testing. It describes testing strategy and testing approaches. It contains following:

•Title, author, version control, history of changes

•Table of contents

•Introduction. Short description of application and requirements

•Functionality that will be and that will not be tested

•Types of testing

•Testing documents

•Hardware, software and tools

•Entry and exit criteria

•Suspension criteria and resumption requirements

•Responsible people

•Schedule and risks

•Approvals

•Appendixes

Some facts about test cases

DataArt

- Test cases can be based on requirements (specifications, communication with customer, mails) or existent functionality
- Used for requirements coverage, providing more quality for less time
- The source for reporting and QA
- Usually automated test scripts base on test cases
- Allows to organize team work

New test case. What to start with?

DataArt

1. Learn requirements and pick out all possible cases including negative cases
2. Check the genuineness of the test case
3. Test case should describe an atomic independent functionality
4. Don’t use passive voice; Test cases should not contain tough language and be ambiguous
5. Avoid using redundant steps
6. Review your test cases

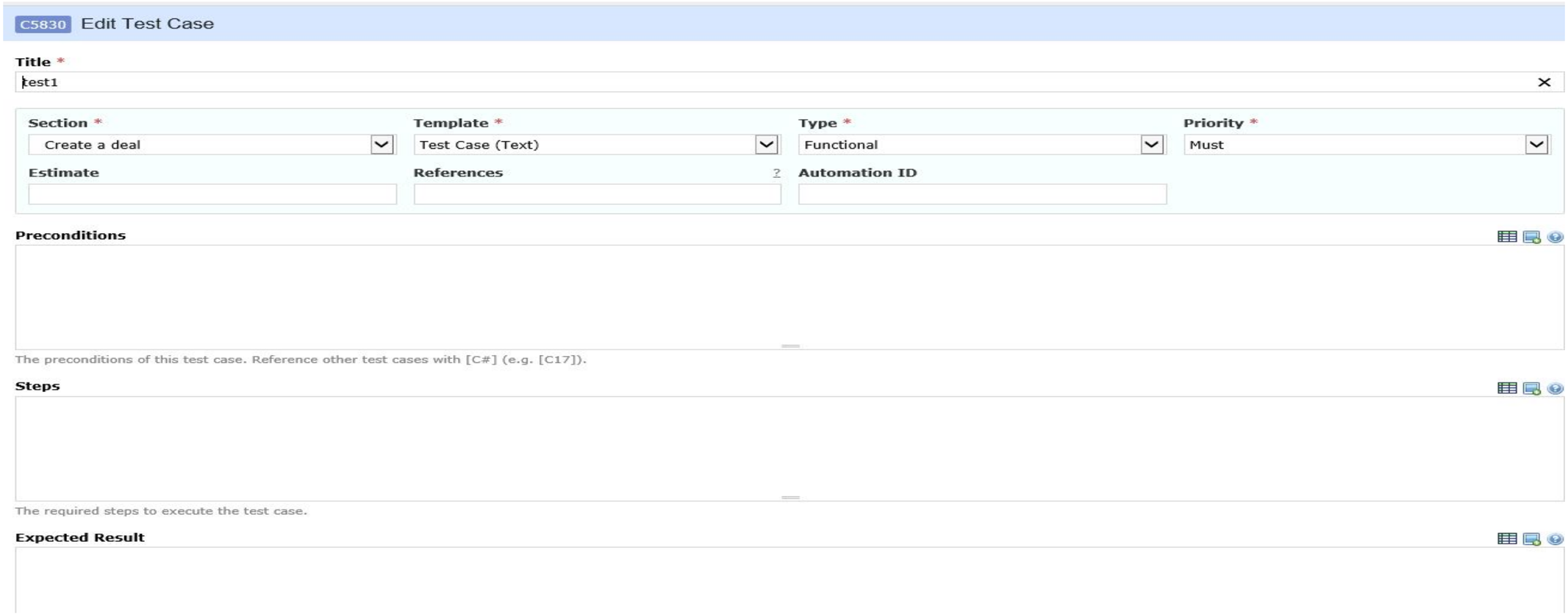
New test case. Let’s focus on attributes

DataArt

1. Unique ID.
2. Author
3. Revision history
4. Priority (critical, major, minor, trivial)
5. Description
6. Preconditions, steps and expected result
7. Post conditions
8. Comments, related requirements and bugs

New test case. Let’s focus on attributes

DataArt



Test suite. Briefly

DataArt

Test suite – batch of test cases, which check certain functionality. For example:

1. User registration
2. Sending messages
3. Removing account

Test suite. Example

DataArt

						Build	1.0.	2.0.
						Test type	AT	AT
						Test date	10.03.2015-12.03.2015	25.03.2015-27.03.2015
						Tester		
						Project Environment	some test by	some test by
						Operating System	Win 7 x 64	Win 7 x 64
						Browser	IE9	IE9
			Project Information				Test Cases Statistics	Test Cases Statistics
Project name	Some project name						Status	Status
Project URL	some.url.com						Passed	Passed
Component	Major functionality						Partially tested	Partially tested
Functionality	Authorization, Registration, Profile						Trivial	Trivial
							Minor	Minor
							Major	Major
							Critical	Critical
							Blocker	Blocker
							Blocked	Blocked
							Not available	Not available
							Not implemented	Not implemented
							Not tested	Not tested
							Total cases	Total cases

Testcase ID	Module	Test type	Testcase header	Steps description	Expected Result	Status	Status
Authorization							
1	Authorization	AT	Checking "Forgot password?" link	1. Go to "XX" page. 2. Press "XX" button. 3. Click "Forget password?" link.	Opens form for inputting E-mail address.	Passed	Passed
2	Authorization	AT	Checking logging	1. Go to "XX" page. 2. Enter E-mail "XXXX@XXX.XXX". 3. Enter password: "XXXXXXX". 4. Press button "log in".	Opens main page. User logged in. Displayed link to profile and user name.	Major	Major
3	Authorization	AT	Logging out	1. Press on user's menu (top right). 2. Click on the link "log out".	User logged out. Displayed main page, user not logged in.	Major	Major
4	Authorization	AT	Sending message	1. Go to "XX" page. 2. Enter E-mail "XXXX@XXX.XXX". 3. Enter password: "XXXXXXX". 4. Press button "log in". 5. Go to user's profile (click on user's name). 6. Press button "Send message". 7. Write text "XXXXX" in summary. 8. Write text "YYYYY" in body. 9. Enter E-mail: "ZZZZZ". 10. Press the button "Send".	Message send. Notification "Message send!" appears.	Critical	Critical

Check-list. Main purposes

DataArt

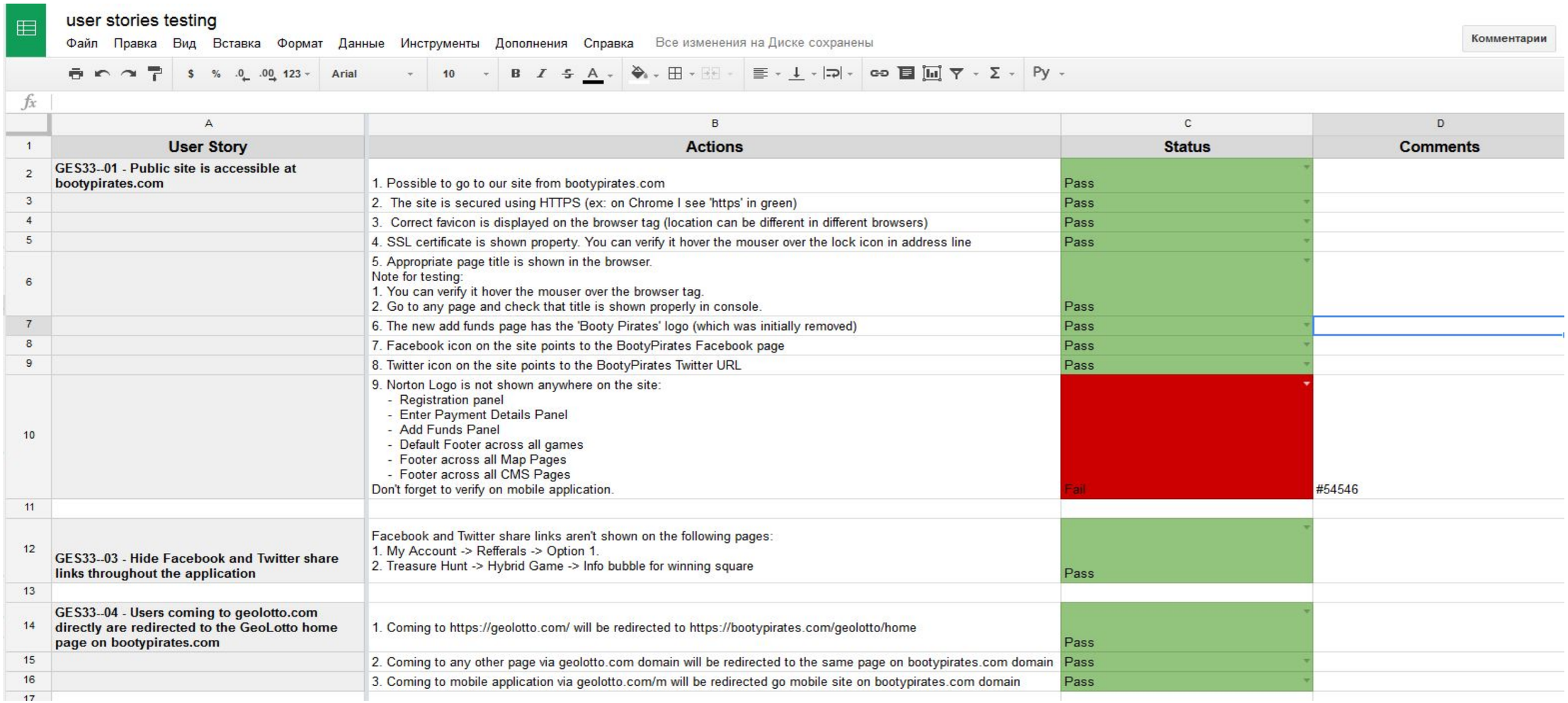
Check-list – list of attributes, applications, characteristics and checks themselves, need for testing.

Mainly used for internal needs. Also:

1. Allows tester not to forget to check something;
2. Expand test coverage;
3. Reduce testing costs;
4. Test control.

Check-list as it is

DataArt



user stories testing

Файл Правка Вид Вставка Формат Данные Инструменты Дополнения Справка Все изменения на Диске сохранены

Комментарии

	A	B	C	D
1	**User Story**	**Actions**	**Status**	**Comments**
2	**GES33--01 - Public site is accessible at bootypirates.com**	1. Possible to go to our site from bootypirates.com	Pass	
3		2. The site is secured using HTTPS (ex: on Chrome I see 'https' in green)	Pass	
4		3. Correct favicon is displayed on the browser tag (location can be different in different browsers)	Pass	
5		4. SSL certificate is shown property. You can verify it hover the mouser over the lock icon in address line	Pass	
6		5. Appropriate page title is shown in the browser. Note for testing: 1. You can verify it hover the mouser over the browser tag. 2. Go to any page and check that title is shown properly in console.	Pass	
7		6. The new add funds page has the 'Booty Pirates' logo (which was initially removed)	Pass	
8		7. Facebook icon on the site points to the BootyPirates Facebook page	Pass	
9		8. Twitter icon on the site points to the BootyPirates Twitter URL	Pass	
10		9. Norton Logo is not shown anywhere on the site: - Registration panel - Enter Payment Details Panel - Add Funds Panel - Default Footer across all games - Footer across all Map Pages - Footer across all CMS Pages Don't forget to verify on mobile application.	Fail	#54546
11				
12	**GES33--03 - Hide Facebook and Twitter share links throughout the application**	Facebook and Twitter share links aren't shown on the following pages: 1. My Account -> Refferals -> Option 1. 2. Treasure Hunt -> Hybrid Game -> Info bubble for winning square	Pass	
13				
14	**GES33--04 - Users coming to geolotto.com directly are redirected to the GeoLotto home page on bootypirates.com**	1. Coming to https://geolotto.com/ will be redirected to https://bootypirates.com/geolotto/home	Pass	
15		2. Coming to any other page via geolotto.com domain will be redirected to the same page on bootypirates.com domain	Pass	
16		3. Coming to mobile application via geolotto.com/m will be redirected go mobile site on bootypirates.com domain	Pass	

One more thing. Bugs.

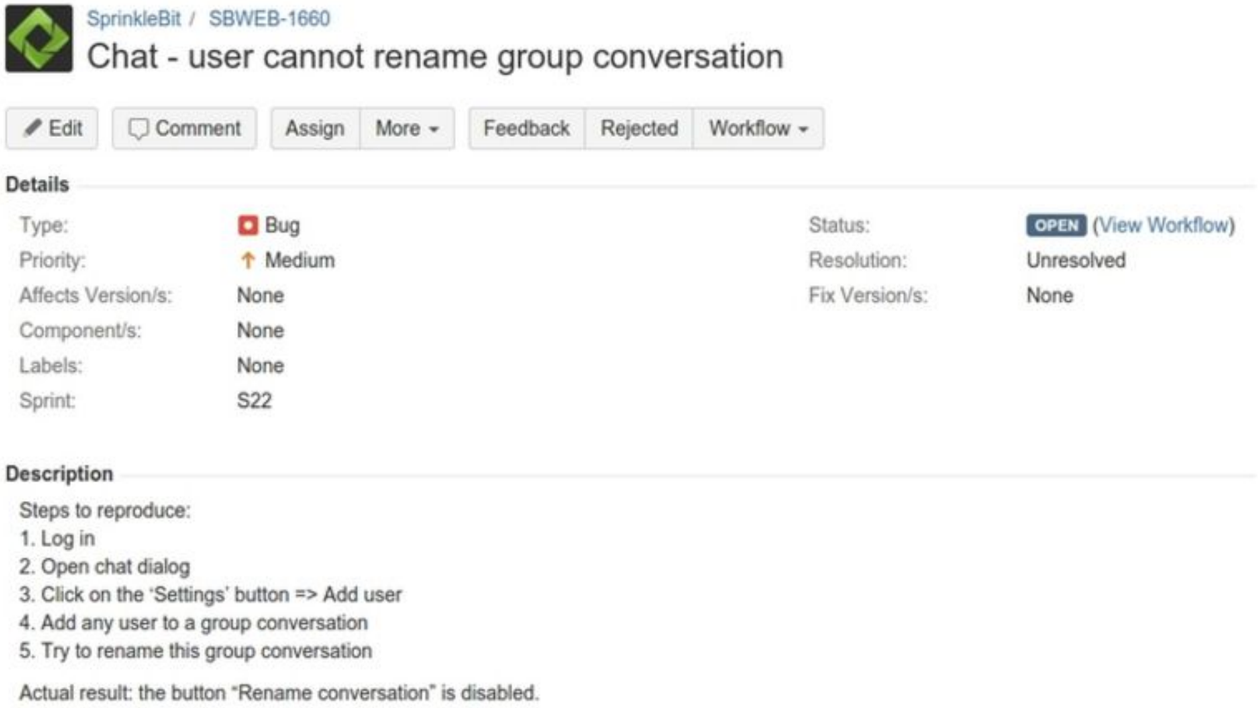
DataArt

Bug is nothing else but program flaw, in other words – defect in software. It can be found while testing software application or product, and usually means difference between expected and actual behavior.

As a rule, such defects show up as a result of error in logic or in coding and result into the failure on unpredicted behavior.

Bug’s example

DataArt



SprinkleBit / SBWEB-1660

Chat - user cannot rename group conversation

Edit | Comment | Assign | More | Feedback | Rejected | Workflow

Details

Type:	Bug	Status:	OPEN (View Workflow)
Priority:	Medium	Resolution:	Unresolved
Affects Version/s:	None	Fix Version/s:	None
Component/s:	None		
Labels:	None		
Sprint:	S22		

Description

Steps to reproduce:
1. Log in
2. Open chat dialog
3. Click on the ‘Settings’ button => Add user
4. Add any user to a group conversation
5. Try to rename this group conversation

Actual result: the button “Rename conversation” is disabled.

Bug reports

DataArt

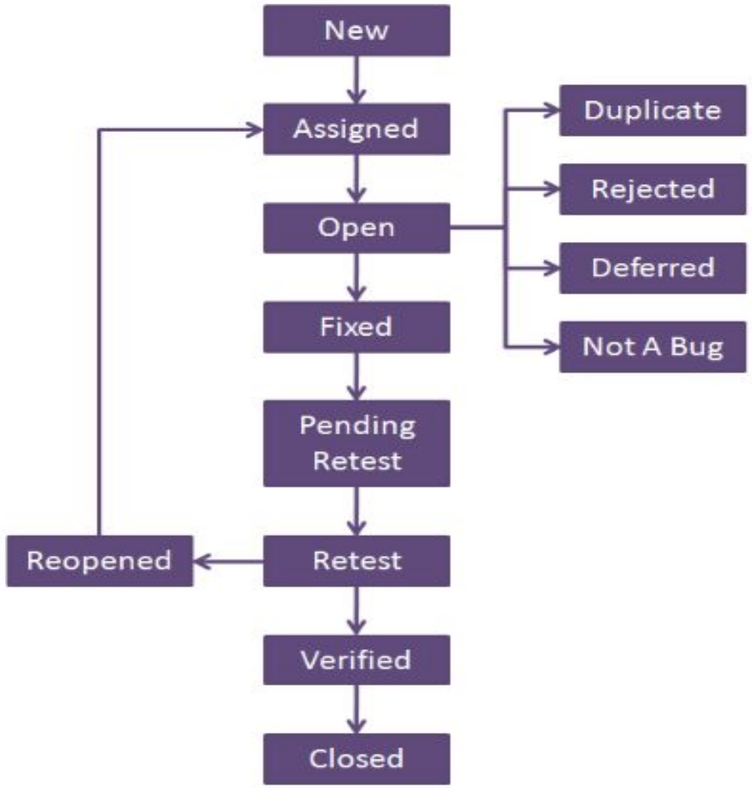
Each bug should be conveyed to the developer. Thus, bug should be reported in a appropriate way. That's why we need documents called Bug Reports.

They should contain following:

1. Defect ID – bug's unique number;
2. Defect description – the summary of the issue;
3. Product version – determines version of a product in which defect is found;
4. Steps to reproduce – includes steps for recreating. Also should contain description for expected and actual results, basing on evidences like screenshots or video recording;
5. Date raised – date of bug reporting;
6. Status – New, Assigned, Open, Retest, Verification, Closed, Failed;
7. Fixed by – This field includes the details of the developer who fixed the defect;
8. Severity – means an impact of the bug on a system (Critical, Major, Minor);
9. Priority – determines the sequence, in which bug will be fixed (Low, Medium, High).

Bug’s lifecycle

DataArt



Bug’s lifecycle

DataArt

1. Finding defect. Status – New

2. Dev team with Project Manager decides whether defect is valid. If not – status Rejected

3. If the defect is not rejected then the next step is to check whether it is in scope. Suppose we have another function- email functionality for the same application, and you find a problem with that. But it is not a part of the current release then such defects are assigned as a **postponed or deferred** status.

4. Then manager verifies, if such was earlier. If yes – status ‘Duplicate’

5. If bug is new, bug is assigned to the developer, who starts fixing it. Status – ‘In Progress’

6. After fixing of bug it’s status is set to ‘Fixed’

7. After this tester starts verifying whether bug is fixed indeed. If such, bug is ‘Closed’. Otherwise, it’s ‘Re-opened’ and re-assigned to a developer

Ar