Defect Tracking with JIRA Tracker

Presenter: An Nguyen

Introduction

- Your role
- Your background, experiences
- What do you want from this course

Course Objectives

At the end of the course, you will have acquired sufficient experience to:

- Acknowledge the defect life cycle
- Acknowledge defect characteristics
- Know how to submit/update/keep track a defect with Jira
- Report defect

Agenda

- JIRA tracker:
  - Overview
  - Defect Tracking Workflow
  - Defect Characteristics
  - JIRA Usage
- JIRA practice

References

- http://cscv-portal.vdc.csc.com/iqms/Service Lines/Development/Test/Guidelines/GUID-Defect Tracking Guidelines.doc
- http://www.atlassian.com/software/jira/docs/v3.8.1/
- Using JIRA course

Assessment Disciplines

- Class Participation + Assignment: 100%
- Final Exam: no
- Course duration: 3 hours

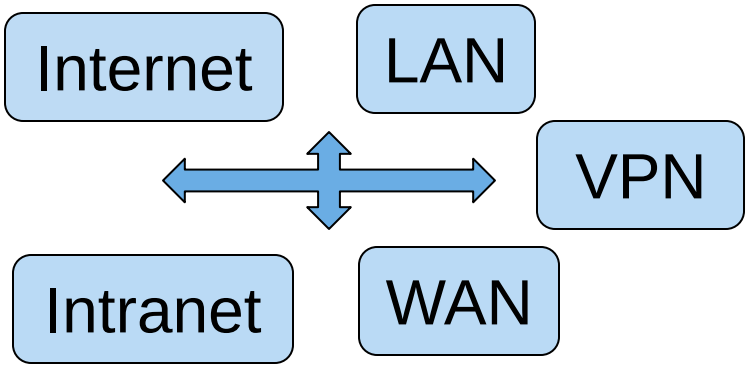
Course Administration

- In order to complete the course you must:
  - Sign in the Class Attendance List
  - Participate in the course
  - Accomplish the assignment
  - Provide your feedback in the End of Course Evaluation


Bug Tracking

JIRA Tracker

- Overview
  - Web-based tracking application
  - Work easily in different kinds of network:



  - Browsers to work with Jira

  - Deployed in many projects
  - Apply email notification

JIRA Tracker

- Introduce Jira tracking system
- Defect Characteristics

JIRA - Defect Characteristics

Some standard fields are applied to all kinds of projects

- R: required when issue is submitted
- U can be edited, updated by Project Lead, Implementer, QC Lead

Field	Definition	Auto	Submitter	Project Lead	Implementer	QC Lead
ID	Unique identifier for issue. Used as the primary reference indicator.	R				
Summary	The descriptive summary of the issue. The summary should give enough information about the issue to identify the effect and possible cause of the problem. Example: “Pressing the Apply button on the configuration dialog causes a system crash” For change request from customer, specify its ID here		R	U	U	U
Created	The date that issue is created	R				
Updated	The date that issue is updated	U				
Components	Components of product that issue belongs to		R	U	U	U
Affects Version/s	Designates the build number in which the issue was discovered. The list of build number is defined by project		R			
Fix Version/s	Designates the build number in which the issue was resolved. The list of build number is defined by project (can leave blank)			U	U	
File Attachments	All file attachments for the issue		U	U	U	U
Image Attachment	All screenshot attachments for the issue		U	U	U	U
Environment	Designates the detailed test environment in which the issue was discovered.		R			

JIRA - Defect Characteristics

Field	Definition	Auto	Submitter	Project Lead	Implementer	QC Lead
Severity	The severity of the issue Pre-defined list: Cosmetic: The product is not working as it was designed, but this is a cosmetic or trivial problem that has little or no business impact on the customer. Minor: Any Jira that affects enhanced program features from working as designed, a work around is available Major [default item]: Any Jira that affects standard program functionality from working as designed, a work around is available Fatal: Any Jira that stops standard functionality of program from working as designed, no work around is available		R	U	U	U
Detected At (list)	The activity to find the Jira. The pre-defined list as below: Requirement Review Design Review Code Review Test Review Other Reviews Unit Test Build Test UAT *Refer to examples of "**Detected At"** at the end of this table*		R	U	U	U
Defect Type (list)	Pre-defined list: Wrong [default item] Missing Extra NA (for not an Jira)		R	U	U	U

JIRA - Defect Characteristics

Field	Definition	Auto	Submitter	Project Lead	Implementer	QC Lead
Defect Category (list)	Pre-defined list: Requirement - Wrong Requirement - Missing Requirement - Ambiguous Design - UI Design - Database Design - Component Design - Architecture Implementation - Error Handling Implementation - UI Implementation - Logical Defect Implementation - Standard Implementation - 3 Party Defect Implementation - Setup Script Test - Documents Test - Scripts Performance Other		R	U	U	U

JIRA - Defect Characteristics

Field	Definition	Auto	Submitter	Project Lead	Implementer	QC Lead
Injected At (list)	The engineering workflow at which the defect injected Requirement Design Implementation Test Others *Refer to examples of "**Injected At**" at the end of this table*		U	U	U	U
Submission Source	Where the issue comes from. The pre-defined list is as below: Customer Onshore Offshore [default item]		R			
Associated Test Case	The test case that issue occurs		R			U
Description	The detailed description of the issue that is used for its general technical discussion. The description should include the effects and the use case/steps to reproduce it. If there are any test scripts or other tools necessary to reproduce the issue they should be referenced here.		R			
Comments	Capture the all the discussions on the issue		U	U	U	U
Issue Types	Pre-defined list: New Feature Improvement Defect Task Sub Task		R	U	U	U

JIRA - Defect Characteristics

Field	Definition	Auto	Submitter	Project Lead	Implementer	QC Lead
Status	Pre-defined list: Open Reopened Resolved Verified Closed					
Resolutions	The current standing of the Jira in the resolution process. The pre-defined list as below: Unresolved Fixed Cannot Reproduce Do Not Fix Need more info Cannot Fix Duplicate Pending Outside Support			U	U	
Assigned to (list)	Who (Implementer) will resolve the Jira (fix defect or implement the change request, new feature) or who will provide more info or who will review Jira and make final decision. Set by Project Leader when reviewing new Jira. The list is defined by project.		U	R	U	U
Cause Category	This is category of root cause which is helpful for further analysis by the project or organization. For pre-defined list, please refer to Defect Cause Category in later section.			U	R	
Defect Root Cause	This is description of root cause of the defect. Defect root cause is expected to be analyzed and identified by the implementer, who addresses the defect (it may require working with other team member(s) to analyze and identify correct root cause), as the defect addressed. Some root cause are obvious but others require the answers to a number of why questions. Fishbone technique is sometime needed to identify the right root cause.			U	R	

Defect Category

Defect Category	Description
Requirement	Note: In case the defect is detected from testing: the behavior is correct as per requirement specifications, but is determined as incorrect at the time of testing. Beside fixing the defect from code, requirements also need to be changed
Requirement – Wrong	Fail to make correct requirement specification.
Requirement – Missing	Fail to make adequate requirement specification.
Requirement – Ambiguous	Fail to make clear requirement specification.
Design	Note: In case of the defect detected from testing: the behavior is correct as per requirement specifications, but is determined as incorrect at the time of testing. Beside fixing the defect from code, architecture/design also need to be changed
Design - UI	Fail to make correct, adequate, clear UI design
Design - Database	Fail to make correct, adequate, clear database design
Design – Component	Fail to make correct, adequate, clear component design
Design – Architecture	Fail to make correct, adequate, clear architecture
Implementation	
Implementation - Error handling	The software fails to detect an error in the data or in their operation. The software fails to defend itself against bad input and bad treatment by other parts of the system or user. The software fails to recover from errors
Implementation - UI	Fail to implement UI correctly, clearly
Implementation - Logical defect	Fail to implement coding logic (algorithm, functions etc.), business logic …
Implementation - Standard	Fail to follow coding standard.
Implementation - 3 Party Defect	The 3rd party component fails to work correctly or work awkwardly when integrated with projected software
Implementation - Setup Scripts	Fail to develop correct installation scripts

Defect Category

Defect Category	Description
Test	
Test - Documents	Fail to make correct, adequate, clear test documents (test cases, test procedures etc.)
Test - Scripts	Fail to develop correct automated test scripts
Performance	Fail to meet specified / expected performance: - Slow echo - Slow run - Poor responsiveness - Etc.
Others	Defects found not belong to above list
Not a Defect	For change request or not an SCR.

Cause Category

Cause Category	Description
Document Unclear/Ambiguous/ Inconsistent	The original document or design did not reflect the client's real need or was not specific enough. Examples: a requirement that failed to fully define user expectation; Test cases are incorrect or not clear; Detail design is incorrect or not sufficient in detail
Lack of Training	The resource was not sufficiently trained to carry out the work or lacked knowledge of the specific tool or environment or of the application. Examples: a VB programmer assigned to a C++ development without training; estimators had no knowledge/skills necessarily leading to poor estimates results
Human Error	A straight mistake or oversight regarding human behaviors/attributes. Examples: a simple coding error due to mis-typing; an error caused by not unit test carefully or not cover all cases in implementation/testing
Failure to Follow Procedure or Process	The correct procedure is in place but was not followed. Examples: Release code without following the Configuration Management process; Not follow design standards; Not follow coding standards; Not unit test
Management Issue	Appropriate management actions not taken e.g., risk and issues not monitored and managed, plans not kept up to date or communicated to team, time scales arbitrarily cut, work priorities changed, resource reassigned.
Poor Procedure or Process	The implemented processes or procedures do not work, or have lead to misunderstandings. An example might be a defective configuration management process that although correctly executed has allowed code to be incorrectly deployed.
Unrealistic Deadlines Imposed	Deadlines have been imposed that have prevented the product from being properly developed, reviewed or tested, in accordance with the processes.

Cause Category

Cause Category	Description
Unrealistic Deadlines Imposed	Deadlines have been imposed that have prevented the product from being properly developed, reviewed or tested, in accordance with the processes.
Lack of Appropriate Tool	The wrong tool set is trying to be used for the work leading to problems. An example might be trying to produce a GUI interface in COBOL.
Inappropriate Environment	The environment used to work in is inappropriate to the type of work. An example could be trying to do a complex design in a crowded noisy office.
Poor Communication	Mis-communication or lack of communication within the project team or between the project team and other stakeholders.
Caused by 3rd Party supplied product	A defect (or problem) was detected in the product (or service) that was supplied by a 3rd Party and being used by the project.
Inadequate/ Improper input from Client	Information and/or requirements received from Client are unclear, confusing or incomplete and further clarification is needed.
Others	

Status

Status	Description
Open	The issue is opened and ready for the assignee to start work on it.
In Progress	The issue is being actively worked on at the moment by the assignee.
Reopened	This issue was once resolved, but the resolution was deemed incorrect. From here issue are either reopened, or are closed.
Resolved	A resolution has been taken, and it is awaiting verification by reporter. From here issues are either reopened, or are closed.
Verified	The defect is verified.
Closed	The issue is considered finished, the resolution is correct. Issues which are closed can be reopened.

Resolution

Jira Resolution	Jira Status						Description	Source Worker	Target Worker
	Open	In progress	Reopened	Resolved	Verified	Closed			
Unresolved	X	X	X				The submitted resolution is the starting one for all new issues reopened issues At this time the Jira as not been assigned to the implementer	Submitter	Project Leader
Fixed				X	X	X	The Jira has been fixed/implemented by implementer and is ready for testing	Implementer Project Leader	Submitter
Cannot Reproduce				X		X	It has been determined that the defect cannot be reproduced.	Implementer Project Leader QC Lead	Project Leader
Duplicate				X		X	It has been determined that this is a duplicate of an existing Jira The Jira ID of the other defect must be specified in Link field	Implementer Project Leader QC Lead	Project Leader
Do Not Fix				X		X	The Jira will not fixed/implemented. The reason can be: the he defect is out of date which means it does not need to fix or Implementer determines that the code is working as designed. It will need to specify the root cause.	Implementer Project Leader	Project Leader

Resolution

Jira Resolution	Jira Status						Description	Source Worker	Target Worker
	Open	In progress	Reopened	Resolved	Verified	Closed			
Cannot Fix	X	X	X	X		X	The Jira cannot be fixed/implemented. The reason could be it is caused by another defect, or fixing it requires a redesign or substantial changes to the module The Jira ID of the relevant defect must be specified in Link field	Implementer Project Leader	Project Leader
Pending outside support	X						The Jira fixing/implementing is outside of the project team and request outside support (belong to customer or 3rd party)	Implementer Project Leader	Project Leader
Future candidate	X						The Jira will be addressed in the next release. All Future Candidate defects must be listed in the Release Notes accompanying the release, under the section “Known Bug”/CR.	Implementer Project Leader QC Lead	Project Leader
Need more info	X						The defect description does not provide sufficient information for the Implementer to action upon	Implementer Project Leader QC Lead	Submitter

JIRA - Defect Characteristics

- Some examples to identify info need to be filled for fields “Detected At” and “Injected At”:
- Example 1: When reviewing test cases, a reviewer sees that the test cases are not very clear. Doing some further checks the reviewer finds that the use cases/requirements associates with these test cases are too ambiguous, then s/he reports that defect as Defected At "**Test Review**" and Injected At "**Requirement**"
- Example 2: When testing a build/product, a reporter sees that the system allows user to enter special characters like !@#$%^&*() ... in a Name field while the requirement states that these characters are not allowed. The reporter also checks the Design document and sees that the Design does not cover that case. Then s/he will report that defect as Detected At "**Build Test**" and Injected At "**Design**“

JIRA - Defect Characteristics

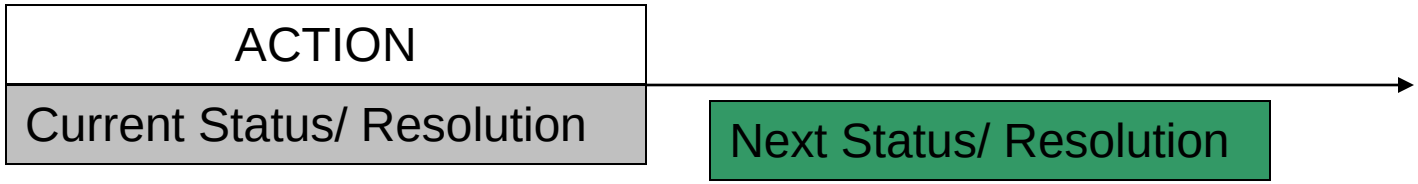
- Example 3: When reviewing code, reporter sees that the code does not set constraint for an Integer variable to 100 as stated in the Requirements and followed by the Design then s/he will report that defect as Detected At "**Code Review**" and Injected At "**Implementation**"
- Example 4: When a developer cross-tests her/his colleague's bug fixed, s/he finds out some issues caused by that fix then s/he will report that issue as Defected At "**Unit Test**" and Injected At "**Implementation**"
- Example 5: Before sending a build to customer, a tester retests the build and finds out a test case was missed, then the tester submitted that issue as Defected At "**Build Test**" and Injected At "**Test**"

What Are Captured And Tracked?

- Defects from unit test
- Defects from review
- Defects from build test
- Minor change requests
- Resource Requests
- Support Requests
- Tasks

JIRA-Defect Tracking Workflow

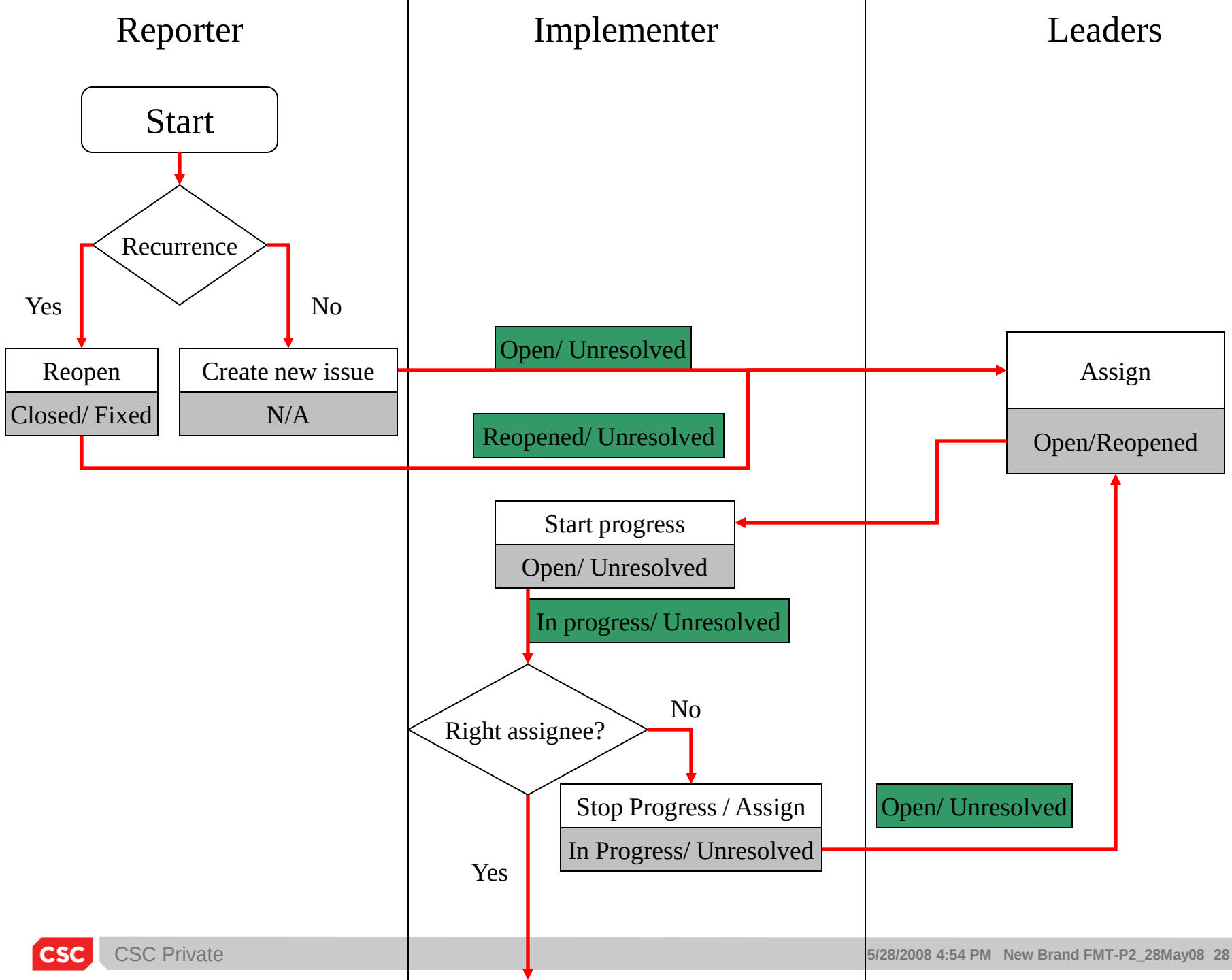
Object in the workflow:

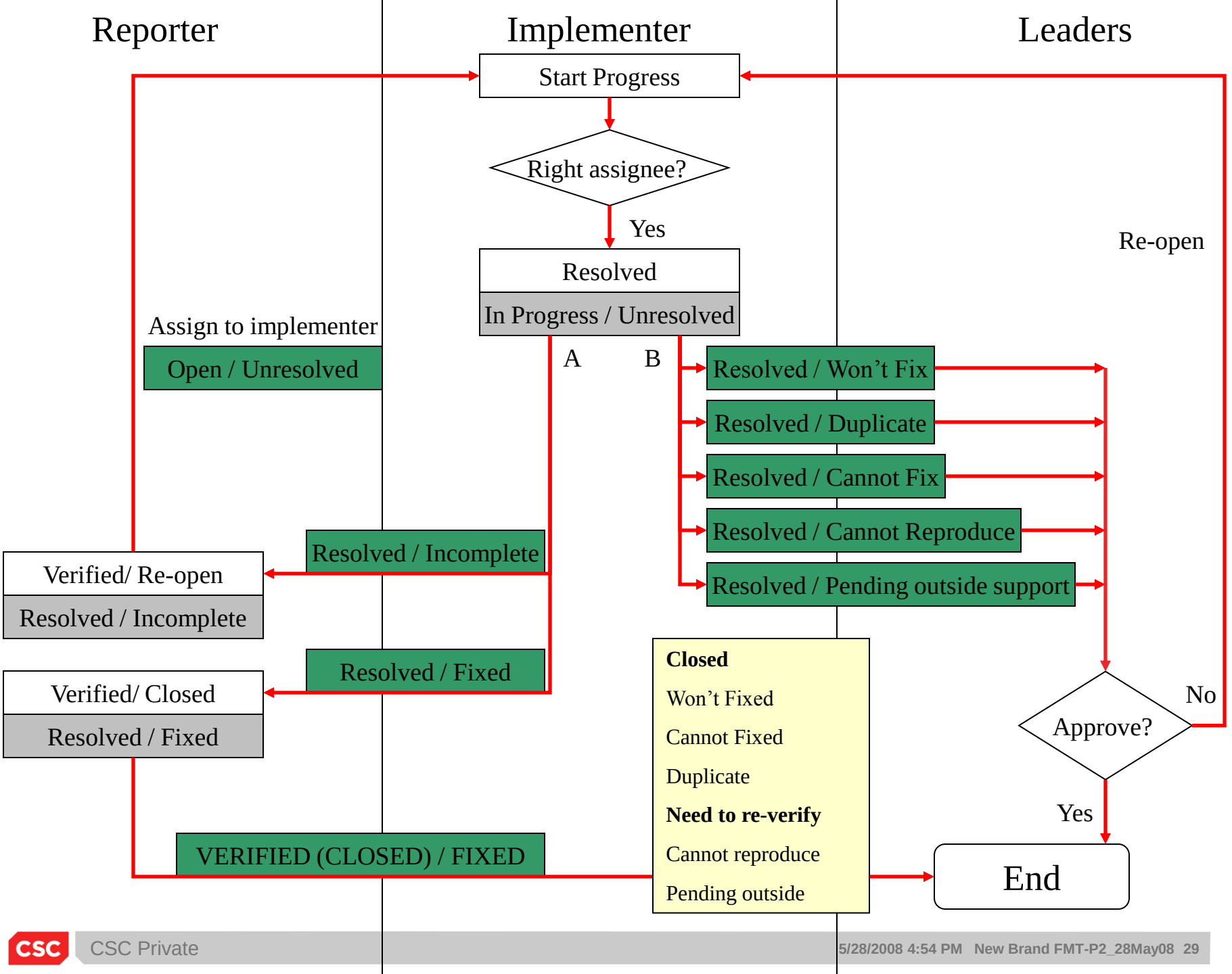


JIRA-Defect Tracking Workflow

Status and available actions:

Step Name (id)	Linked Status	Transitions (id)	Operations
Open (1)	Open	Start Progress (11) >> In Progress Resolved (21) >> Resolved Closed (31) >> Closed	View Properties
In Progress (2)	In Progress	Stop Progress (41) >> Open Assign (51) >> Open Resolved (61) >> Resolved	View Properties
Resolved (3)	Resolved	Verified (71) >> Verified Closed (81) >> Closed Reopen (91) >> Open	View Properties
Reopened (4)	Reopened	Start Progress (101) >> In Progress Resolved (111) >> Resolved Closed (121) >> Closed	View Properties
Verified (5)	Verified	Reopen (131) >> Open Closed (141) >> Closed	View Properties
Closed (6)	Closed	Reopen (151) >> Reopened	View Properties


Reporter
Implementer
Leaders
Start
Recurrence
Yes
No
Reopen
Closed/ Fixed
Create new issue
N/A
Open/ Unresolved
Reopened/ Unresolved
Assign
Open/Reopened
Start progress
Open/ Unresolved
In progress/ Unresolved
Right assignee?
No
Yes
Stop Progress / Assign
In Progress/ Unresolved
Open/ Unresolved


Reporter
Implementer
Leaders
Start Progress
Right assignee?
Yes
Resolved
In Progress / Unresolved
Re-open
Assign to implementer
Open / Unresolved
A
B
Resolved / Won't Fix
Resolved / Duplicate
Resolved / Cannot Fix
Resolved / Cannot Reproduce
Resolved / Pending outside support
Resolved / Incomplete
Verified/ Re-open
Resolved / Incomplete
Resolved / Fixed
Verified/ Closed
Resolved / Fixed
Closed
Won't Fixed
Cannot Fixed
Duplicate
Need to re-verify
Cannot reproduce
Pending outside
Approve?
No
Yes
VERIFIED (CLOSED) / FIXED
End

Using JIRA

- Login to Jira tracker with own username
- Create a new issue (defect)
- Find an issue
- View available issues
- Update issue
- Quick demo with QC role and Developer role

- Q&A
- Break 10 minutes …


PRACTICE

JIRA Practice

- Jira tracker and user accounts (user name/ password):
  - https://vdc-jira.asia.csc.com/jira/browse/TRN
  - Log into Jira system with your network account
  - Select project “Training” to practice below exercises
- Exercise 1: Submit 2 defects to JiraTracker with appropriate required fields, pre-fix your defects with Defect1_YourID/Yourname, Defect2_YourID/Yourname
- Exercise 2: Assign Defect1, Defect2 to you with a comment.

JIRA Practice (cont.)

- Exercise 3: As a developer, follow developer workflow to fix Defect1

- Exercise 4: As a developer, start progress with Defect2 and then you know that you are not an appropriate assignee, so you stop progress to return issue back Open status.

- Exercise 5: As a QC, verify Defect1

JIRA Practice (cont.)

- Exercise 6: As a leader, close Defect1_YourID
- Exercise 7: Find all issues that you have submitted
- Exercise 8: As a leader, move issue type of Defect2 from Defect to Task.

Q&A


THANK
YOU

Revision History

Date	Version	Description	Updated by	Reviewed and Approved By
03/20/2007	1.0	Add Jira section to the original course.	Trang Vo	Nguyet Pham
2009	1.1	Separate Jira and PVCS section	Mai Vo	Nguyet Pham
01/13/2010	1.2	Update practice section	Trang Vo	Nguyet Pham
04/22/2010	1.3	Add revision history, CSC Private in the footer, change cases, spelling	Trang Vo	Nguyet Pham
04/10/2013	1.4	Update Defect Characteristics: Add standard Jira fields and examples Add some image replace for text: Jira Tracker / Jira practice	Lieu Ho	Mai Thi Huynh Nguyen, Anh Thi Phuong Nguyen