Test Execution

May 2016, 2018

softserve

Agenda

- *What is build?*
- *Versioning, Version Control Systems*
- *Test Execution process*
- *Test Execution in Zephyr for Jira*
  - ✔ Test Cycle
  - ✔ Test Execution
  - ✔ Test Case
  - ✔ Test Execution Progress

ft**serve**

Build

Version Control System

What is build?

Build is a version of a program.
The term is usually used in reference to a program that is still in development and is used only by development company.

Who	DEV
When	• A certain point in development has been reached • The code has been deemed ready for testing or release
Why	To create an application that can be tested
How	Generated by source code compilation along with necessary files

Compilation it's a process of translating source code from a high-level programming language to a lower level language (e.g., assembly language or machine code).

When the build is finished, it is often stored as a single package and is marketed under a version number.

softserve

Versioning

Software versioning is the process of assigning either unique **version names** or unique **version numbers** to unique states of computer software.

Versioning is used to:

✔ Keep a pulse of progress for any software development cycle

✔ To describe program history

✔ Keep up with competitors

softserve

Versioning model examples

Sequence-based identifiers	Date of Release identifiers
Each software release is assigned a unique identifier that consists of one or more sequences of numbers or letters	Each software release is assigned a unique identifier that contains the day of release

✔ numbers: Adobe Photoshop 5.5
✔ numbers and letters:
  1.0b2 (beta)
  1.0rc1 (release candidate)

✔ Ubuntu Linux: Ubuntu 11.10, was released in October, 2011
✔ Wine: Wine 20040505, was released in 5th of May, 2004

Different software producers use different schemes to denote releases of their software.
since version numbers are human-generated, not computer-generated, there is nothing that prevents arbitrary changes that violate general guidelines

softserve

Version Control System

A **Version Control System** (or **Revision Control System**) is a system that tracks incremental versions (or revisions) of files and, in some cases, directories over time.
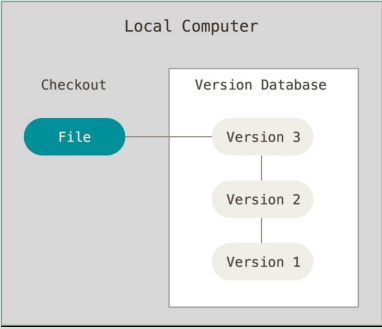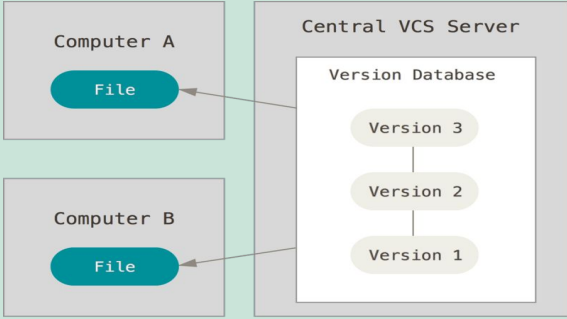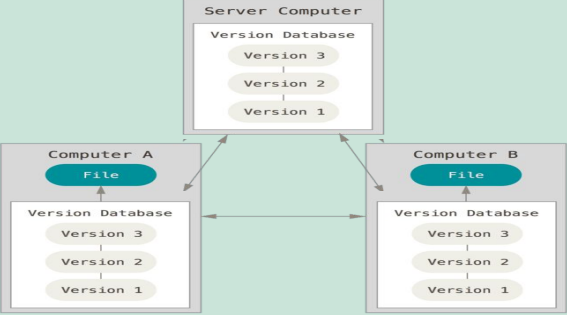
Problems resolved by Version Control System:

✔ Communication with team via email about updates

✔ Making updates directly on production server

✔ Accidentally overwriting files, which can never be retrieved again

Version control systems are essential for any form of distributed, collaborative development.

softserve

Comparison of version control software

Local Version Control Systems (VCS, RCS)	Centralized Version Control Systems (CVS, Subversion or SVN, Perforce)	Distributed Version Control Systems (Git, TFVC, Mercurial, Bazaar or Darcs)
local VCSs had a simple database that kept all the changes to files under revision control on Local Computer.	These systems have a single server that contains all the versioned files, and a number of clients that check out files from that central place. For many years, this has been the standard for version control.	This is systems where clients don't just check out the latest snapshot of the files: they fully mirror the repository. Thus if any server dies, any of the client repositories can be copied back up to the server to restore it. Every clone is really a full backup of all the data.

softserve

Version Control System

Generally speaking, source control tools provide some or all of these features:

✔ Check out, check in
✔ Merging
✔ Project control
✔ Change tracking
✔ Difference checking
✔ History

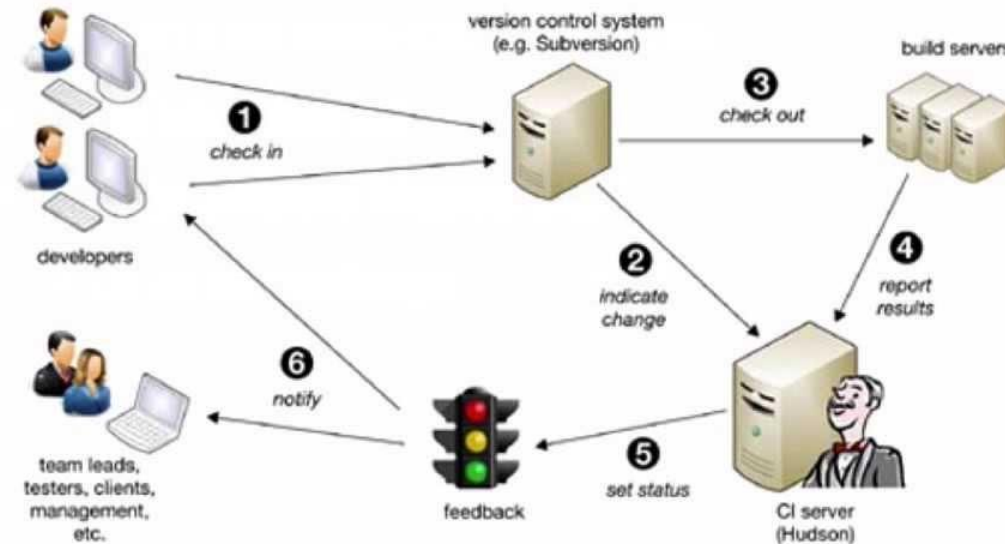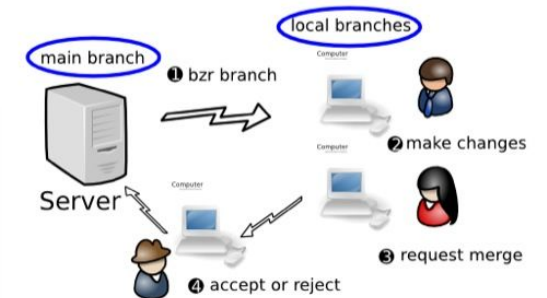Image source: http://www.methodsandtools.com/tools/tools.php?hudson

softserve

Typical environments

Development

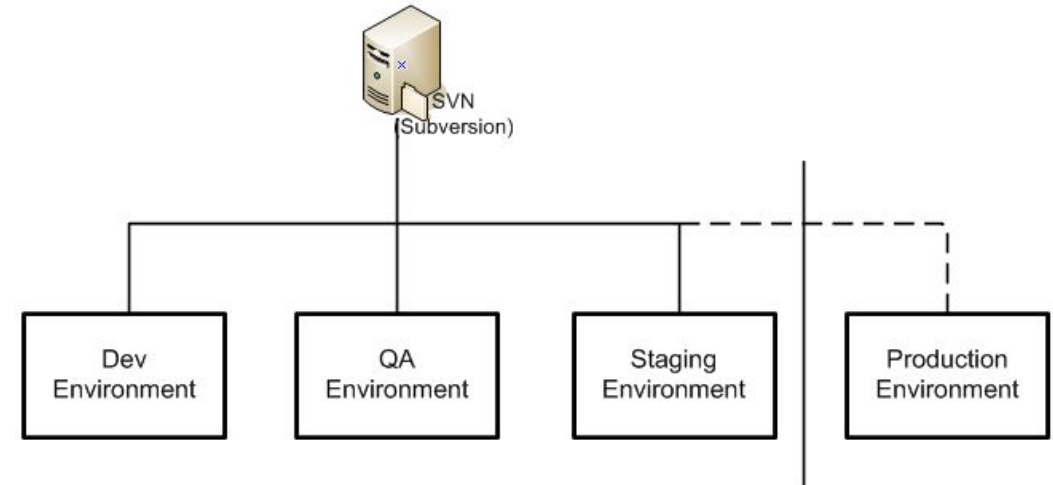Where programmers work, Unit tests happen here

Test

For integration, system, and regression testing

Stage

For burn-in and load testing, performance testing, user acceptance testing
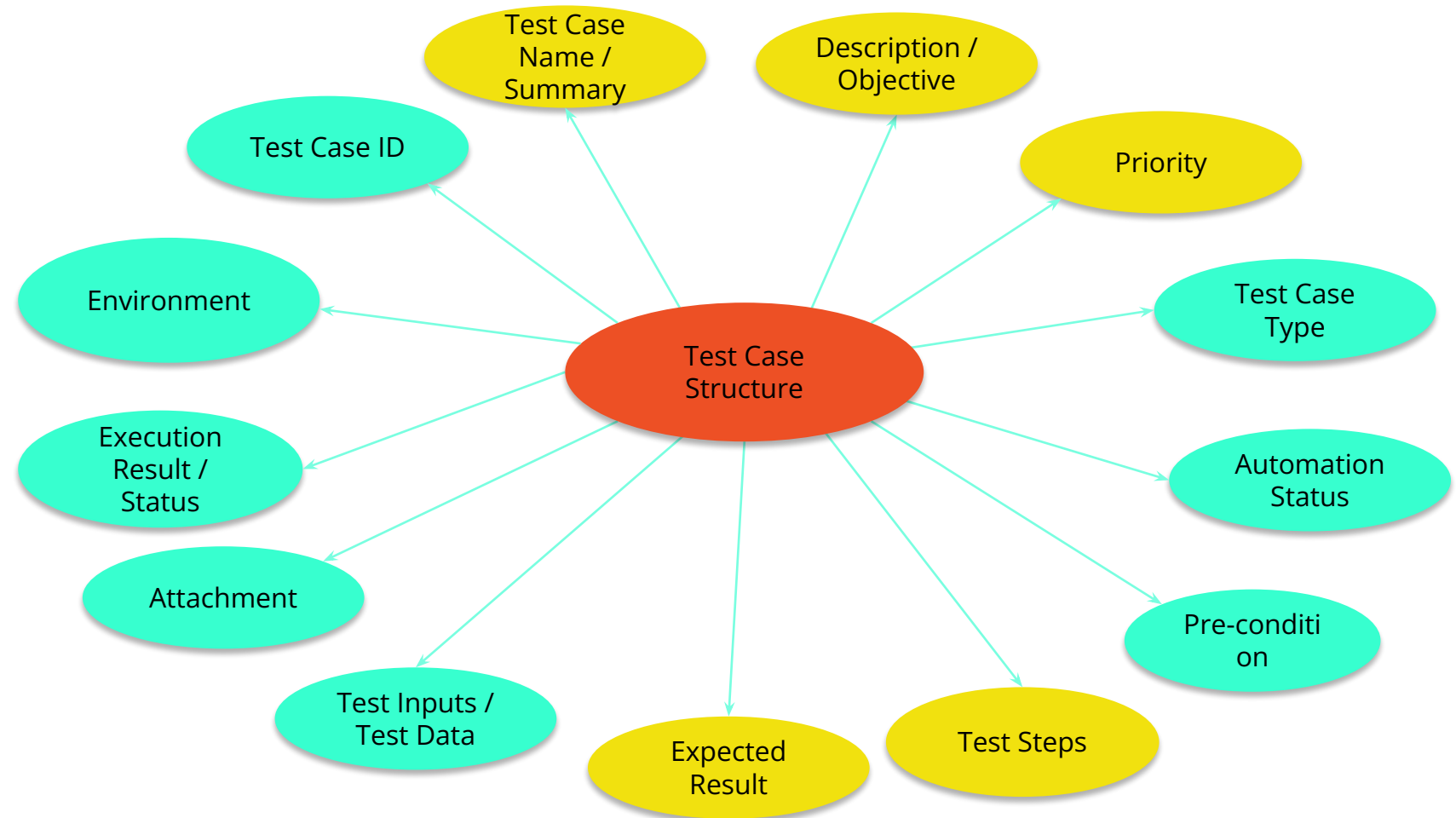
Production

Final deployment environment(s)

softserve

Test Case

softserve

Test Case

- Test Case Name / Summary
- Description / Objective
- Test Case ID
- Priority
- Environment
- Test Case Type
- Test Case Structure
- Execution Result / Status
- Automation Status
- Attachment
- Pre-condition
- Test Inputs / Test Data
- Expected Result
- Test Steps

Test Case it's a set of input values, execution preconditions, expected results and execution post conditions, developed for a particular objective or test condition, such as to exercise a particular program path or to verify compliance with a specific requirement.

softserve

Test Case

Test Case ID	BU_001	Test Case Summary	Test the Login Functionality in Banking		
Created By	Mark	Reviewed By	Bill	Environment	Staging

QA Tester's Log	Review comments from Bill incorprate in version 2.1

Tester's Name	Mark	Date Tested	1-Jan-2017	Test Case (Pass/Fail/Not Executed)	Pass

S #	Pre-conditions:		S #	Test Data
1	Access to Chrome Browser		1	Userid = mg12345
2			2	Pass = df12@434c
3			3	
4			4	

Description	Verify on entering valid userid and password, the customer can login.

Step #	Step Details	Expected Results	Pass / Fail / Not executed / Suspended
1	Navigate to http://demo.guru99.com	Site should open	Pass
2	Enter Userid & Password	Credential can be entered	Pass
3	Click Submit	Cutomer is logged in	Pass

softserve

Test Execution

softserve

Test Execution

Fundamental Test Process

Test planning

Test monitoring and control

Test analysis

Test design

Test implementation

Test execution

Test completion

Tasks:

Execute Test Cases

Log the outcomes

Compare Actual and Expected results

Report Defects

Confirmation and Regression testing

Verifying and updating traceability matrixes

soft**serve**

Test Execution

Execute Test Cases

Log the outcomes

Compare Actual and Expected results

Report Defects

Confirmation and Regression testing

Verifying and updating traceability matrixes

✔ Record the IDs and versions of the test item(s) or test object, test tool(s), and testware
✔ Execute the test suites and individual test cases, following test procedures
✔ Compare actual results with expected results
✔ Analyze anomalies to establish their likely causes (e.g., failures may occur due to defects in the code, but false positives also may occur)
✔ Reporting defects based on the failures observed
✔ Log the outcome of test execution (e.g., pass, fail, blocked)
✔ Test corrected software again to ensure that the defect correction did not introduce new defects
✔ Verifying and updating bi-directional traceability between the test basis, test conditions, test cases, test procedures, and test results

softserve

Test Execution Tips

✔ **Pre-Conditions**: Make sure all the pre-conditions are fulfilled before executing test.

✔ **Completing steps in order**: Execute Tests in specified order

✔ **Status**: If the status for a particular step is FAIL, verify if this bug has already been reported. If not, report it immediately. After this continue completing the subsequent steps in the test case. If subsequent steps have dependencies on the failed step and there are no workarounds, then mark them as N/A

✔ **Observations**: Record the observations you've done while completing a particular step

✔ **Thorough completion of the Test Case**: Do not leave any steps not executed. Add observation and set status to N/A

softserve

Test Case Management Tools:

Zephyr for Jira

Test Case Management Tools

Test Case Management Tool – A tool that provides support to the test management and control part of a test process.

- Microsoft Test Manager
- JIRA TCM Solution
- TestLink
- TestLog

✔ Ability to create new and effectively manage existing Test Cases
- TestRail
✔ Ability to track history, Test Case executions, total run time, and estimate workload
- Redmine
✔ Ability to organize and categorize your Test Cases by Product, Component, Test Type, Test Component and
- Fitness
Test Subcomponent
- Borland Silk
✔ Versioning of Test Cases
- Qmetry
✔ Group Test Cases into Test Cycles
- Zephyr
✔ Presence of search and filter capabilities
- Excel
✔ Ability to link Test Cases with requirements, defects and vice versa

✔ Metrics gathering, reports creation, etc.

softserve

Zephyr for Jira

Zephyr for JIRA is an add-on application that augments JIRA 5 and 6, providing cost-effective, highly sophisticated test management capabilities right inside your JIRA.

Term	Description
Test	A test case. This issue-type has default issue workflow and this can be turned on or off. By default, it is off
Test Summary	High level summary and counts of all the tests that have been created in a particular project, grouped in various ways
Test Cycle	A grouping of executed or unexecuted tests. More than one test cycle can exist for a *Version*
Execution	When a test is run and its result or status is recorded
Execution statuses	Pass, Fail, Blocked, WIP (Work In Progress), Unexecuted are default statuses. Custom statuses can be added

softserve

Zephyr for Jira

Major features include:

- ✔ Create, view, edit and clone tests
- ✔ Link to stories, tasks, requirements etc.
- ✔ Plan test execution cycles
- ✔ Execute tests
- ✔ File defects
- ✔ Track quality metrics
- ✔ Create custom dashboards
- ✔ Perform advanced searches using ZQL

soft**serve**
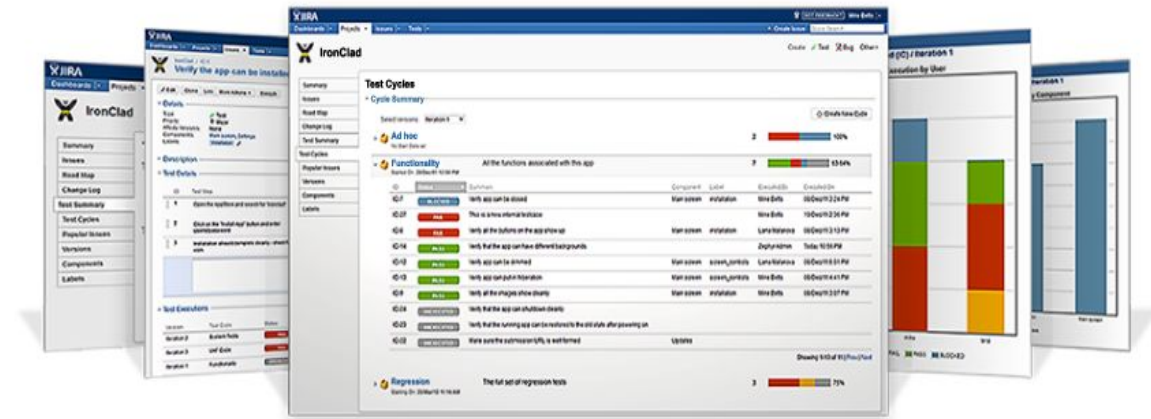
Test Execution in Zephyr

Change Test Execution status from Test Cycle

Test Cycles

Cycle Summary

Select Versions: TCM Test Version 1 ▾   + Create New Cycle

NewCycle
Started On: 17/Oct/14   Build: Environment: 3   100%

Middle Regression
Started On: 01/Oct/14   Build: Environment: 1   20%

ID	Status	Summary	Defect	Component	Label	Executed By	Executed On
TESTTCMA-380	BLOCKED	Creating new user account with cyrillic user name	-	component1		Myroslava Harasym	Today 1:57 PM
TESTTCMA-379	UNEXECUTED	Creating new user account with incorrect user name	-	component1			
TESTTCMA-378	UNEXECUTED	Creating new user account with duplicating user name	-	component1			
TESTTCMA-377	UNEXECUTED	Creating new user account without saving	-	component1			
TESTTCMA-329	UNEXECUTED	Creating new user account and save	-	component1			

softserve

Test Execution in Zephyr

Change Test Execution status from Test Execution

Creating new user account with incorrect user name

← Return to Test Cycst Cyc

⌄ Description

Cloned from another existing test:
This Test case verifies that user account could be created if all fields on 'User Registration' page are filled with correct data; and 'User Registration' is closed on save action

⌄ Test Execution

Execution Status: UNEXECUTED ✎

Defects: ✎

Comment: ✎

⌄ Attachments (Execution) + +

⌄ Test Details

	Test Step	Test Data	Expected Result	Status	Comment	Attachments	Defects	
1	Set 1# value to 'User Name' field	1# "U", "MyTestUser"		UNEXECUTED ✎	✎	+	✎	✎
2	Set 2# to 'Password' field	2# "Password1", "Password2"		UNEXECUTED ✎	✎	+	✎	✎
3	Set 3# to 'Confirm Password' field	3# "Password1", "Password2"		UNEXECUTED ✎	✎	+	✎	✎
4	Click 'Save' button on the page		Just added #1 user is available in the list of the users	UNEXECUTED ✎	✎	+	✎	✎

softserve

Test Execution in Zephyr

Change Test Execution status from Test

Test Execution Progress

Using Test Cycles:

Test Execution Progress

Using Test Summary:

Test Execution Progress

Using Zephyr Dashboard:

softserve

Revision History

Version	Date	Remark	Author
v.1	May, 2016		M. Harasym
v.2	October, 2018	Update according to new ISTQB Standard	V. Ryazhska

softserve

Thank You

softserve