SoftServe
Ability. Agility. Advantage.

Test Cases Overview

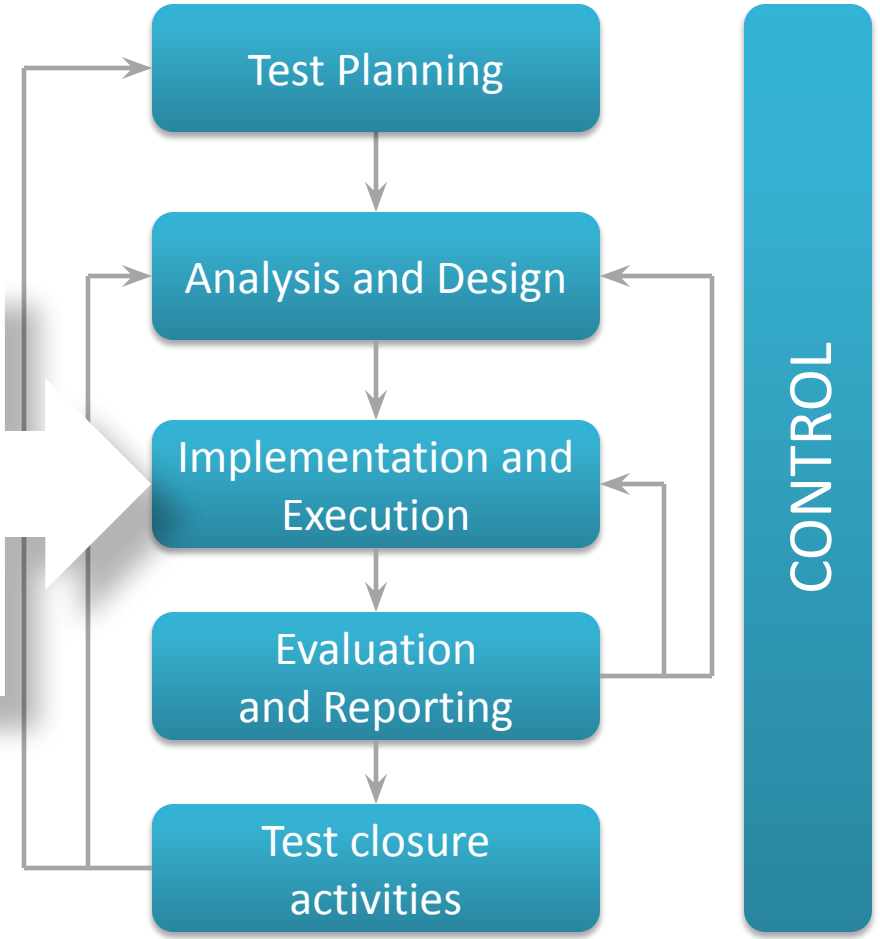
Prepared by:
QMO March 2013

Agenda

- Fundamental Test Process
- Test Case Definition / Structure
- Test Case Implementation
- Test Case Specification
- Test Case Management tools
- Best Practices

Fundamental Test Process

…The purpose of the Implementation and Execution phase is to organize the test cases in procedures and / or scripts and to perform the physical test in the correct environment.



According to Standard Glossary of Terms used in Software Testing

Test Planning and Control

- Determine the scope and risks and identify the objectives of testing
- Determine the test approach (techniques, test items, coverage, etc)
- Implement the test policy and/or the test strategy
- Determine the required test resources (e.g. people, test environment, PCs)
- Schedule test analysis and design, test implementation, execution and evaluation
- Determine the exit criteria

- Measure and analyze the results of reviews and testing
- Monitor and document progress, test coverage and exit criteria
- Provide information on testing
- Initiate corrective actions
- Make decisions

Test Analysis and Design

- Review the **Test Basis ***
- Identify **Test Conditions **** based on analysis of **Test Items ****
- Design the tests using Test Design Techniques
- Evaluate testability of the requirements and system
- Design the test environment set-up and identify any required infrastructure and tools

*** Test Basis** – all documents from which the requirements of a component or system can be inferred (the documentation on which the test cases are based).

**** Test Items (Test Conditions)** – an item or event of a component or system that could be verified by one or more test cases, e.g. a function, transaction, feature, quality attribute or structural element.

Test Implementation and Execution

- Develop and prioritize our test cases, using the techniques
- Create test suites from the test cases for efficient test execution
- Implement and verify the environment

- Execute the test suites and/or individual test cases, following test procedures
- Log the outcome of test execution
- Report discrepancies as incidents
- Repeat test activities as a result of action taken for each discrepancy

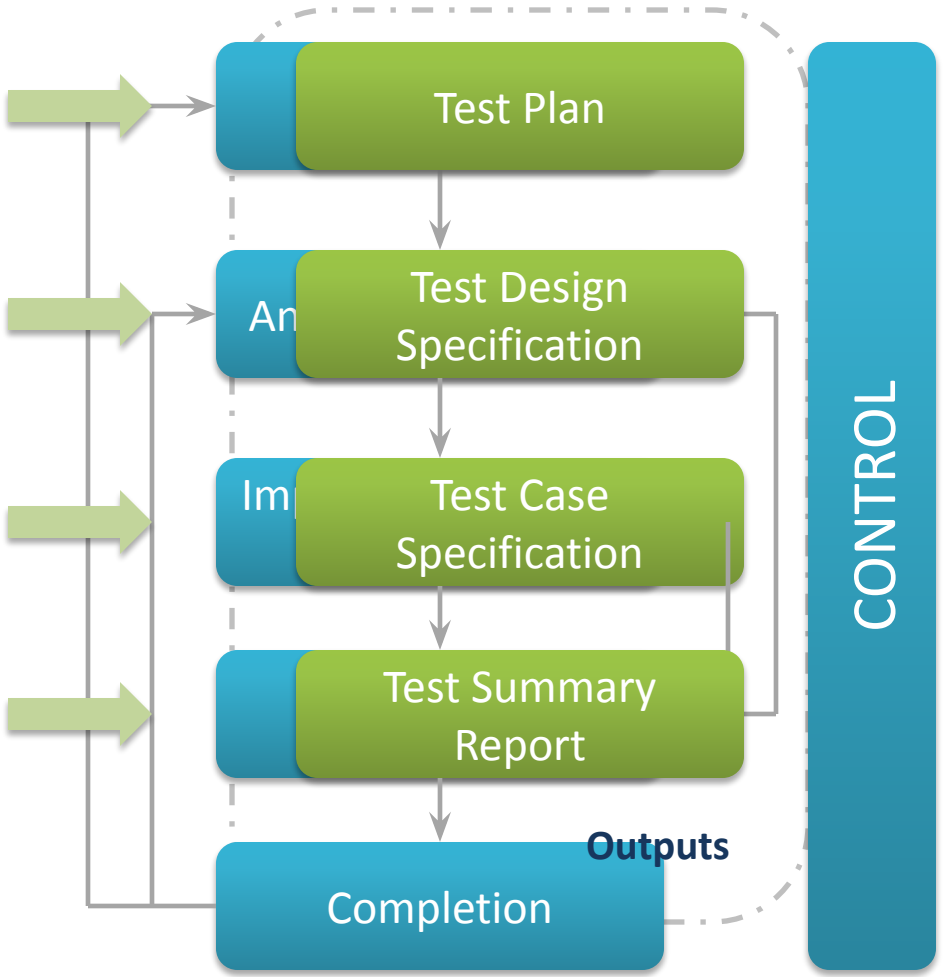
Evaluation and Reporting

- Check test logs against the exit criteria specified in test planning
- Assess if more tests are needed or if the exit criteria specified should be changed
- Write a test summary report for stakeholders

Test Closure Activities

- Check which planned deliverables we actually delivered
- Finalize and archive ‘testware’
- Evaluate how the testing went and analyze lessons learned

Output Documentation

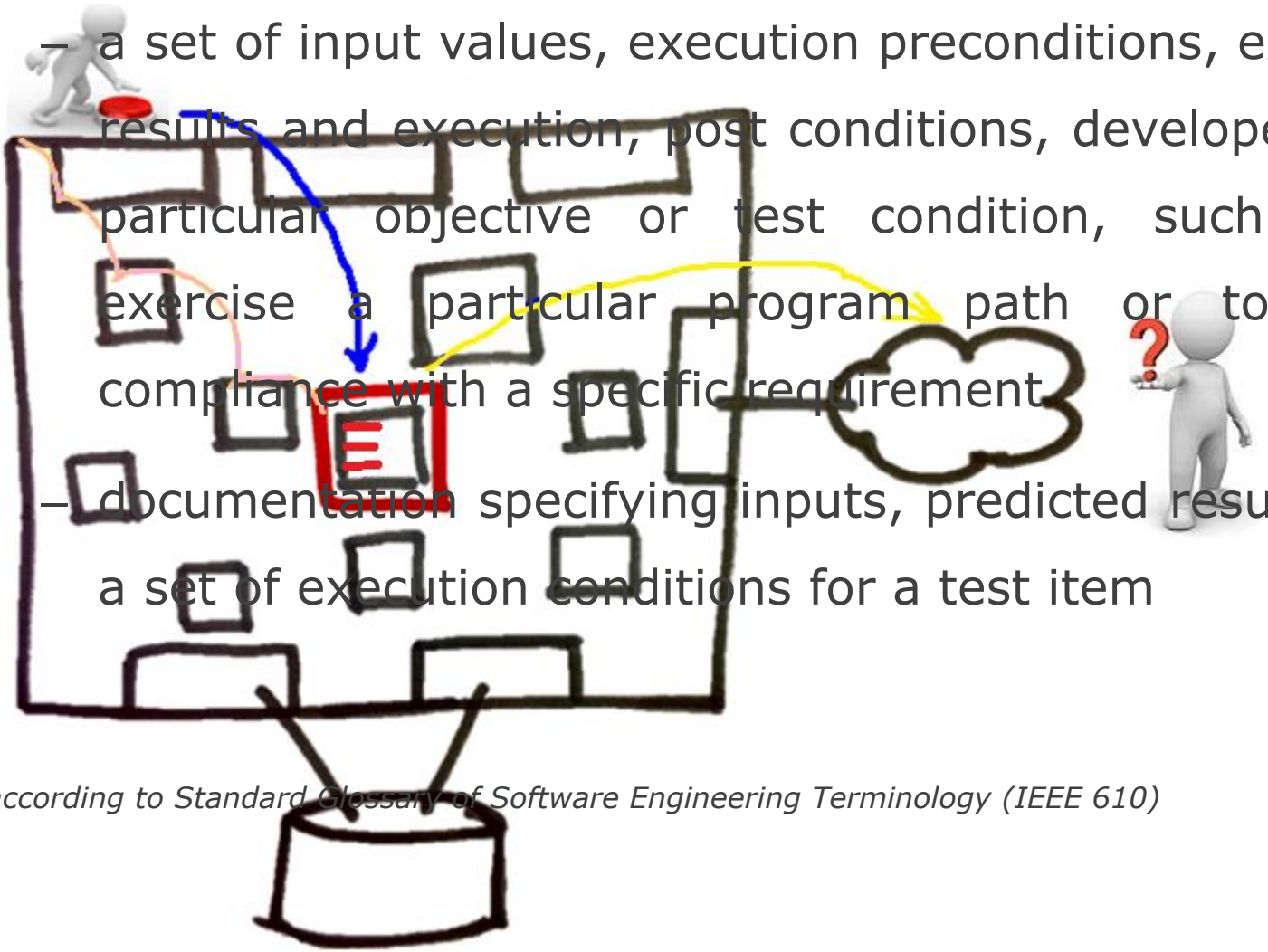


TEST CASE DEFINITION / STRUCTURE

- Test Case - is …
- Test Cases mission…
- Test Case consists of …

Test Case – is …

- 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
- documentation specifying inputs, predicted results, and a set of execution conditions for a test item



** according to Standard Glossary of Software Engineering Terminology (IEEE 610)*

SoftServe
Ability. Agility. Advantage.

Test Cases mission

Test cases creation…

…makes you think about specific usage scenarios, looking for places forgotten by everybody (like implicit requirements);

…helps you detect bugs early, since errors in code can be prevented before the coding is done, also new test ideas will be generated;

…makes your work effective, test cases are prepared before actual implementation, so when it is done, you are ready to go and test product quickly and efficiently;

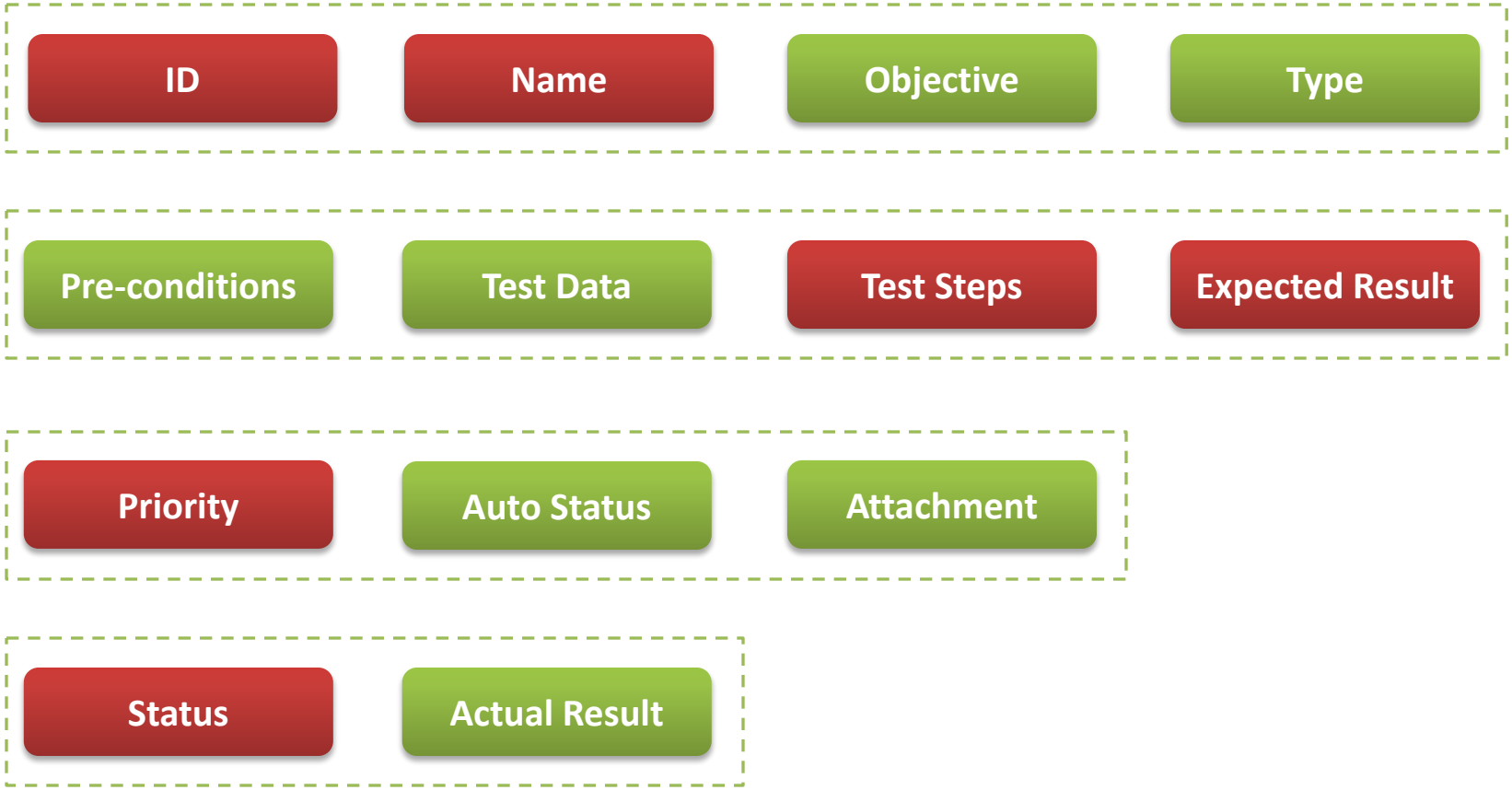
…improves your testing coverage, since cover requirements thoroughly;

…comes as evidence of testing work;

…helps newcomers to familiarize with application.

SoftServe
Ability. Agility. Advantage.

Test Case consists of…



**** Test Case structure might vary depending on particular project needs.***

Test Case Structure

- Identification of the test case
- It should be unique across Test Case Specification
- Can consist of numbers or/and letters

  Example: *1, 2, 3, etc; UR.001, UR.002, etc*

- Short name of test case which briefly indicates what will be verified

- Describes the functionality/actions that test case validates/does
- It should be detailed enough to understand purpose of test case

- Reflects the relative importance of the test case taking into consideration different aspects
- Can be presented by words or numbers

  Example: *High, Medium, Low, etc;*

  Major, Minor, Trivial, etc;

  1, 2, 3 (where 1-the most important, and 3-the least important), etc

Test Case Structure

- Reflects the type of test case depending on what kind of testing is covered by particular test case
  Example: *GUI, Functional, System, Performance, etc.*

- Indicates whether test cases is candidate for automation taking into consideration different aspects or not
- Also indicates whether test cases is automated already or not
  Example:
  Candidate – test case is recommended for automation, but it is not automated yet
  Not Candidate – test case is not recommended to be automated and should be run manually
  Automated – test case is automated and during the next execution can be run automatically etc.

- Defines conditions that should be met before test case can be executed
- Usually pre-condition field lists data/actions which should exist/be done in system and links to appropriate test cases/test functions which can setup required pre-conditions

- Step by step instructions on how to carry out the test case
- There should not be missed or redundant steps!

Test Case Structure

- Shows how the system must react based on the test steps
- Expected results should be mentioned only for test case objective!
- *"Verify", "Correctly", "Successfully"* words are forbidden for expected results! Exact behavior of the system, which is going be verified, should be mentioned:

Example:

Incorrect: Verify "TestUser" user is created -> It is not understandable how to verify it

Correct: "TestUser" user appears in the list of users

- Lists data which is used while test case execution
- Can be presented in this field directly or via link to attached files
- Data should be accurate!

- May contain files which can be used while test case execution

Test Case Structure

- Shows the result of test case execution to indicate whether behavior of the system meets expected results of test case or not

  Example:

  Pass – expected results of test case and behavior of the system match

  Fail – expected results of test case and behavior of the system do not match

  Blocked – test case was unable to be executed due to some reasons (e.g. blocker issue, etc)

  Skipped – test case was untested since it wasn't planned to be executed this time, etc.

- Shows the actual output of the system. This field is used when actual behavior of the system doesn't meet expected results of test case

Test Case: Priority

Software testers may prioritize their test cases in order:

- to reduce the cost of regression testing, so that those which are more important, by some measure (e.g. if the time limits means…), are run earlier in the regression testing process
- to increase a test suite's rate of fault detection, thus allowing developers to fix severe faults earlier in software development process

Prioritizing test cases can be done by considering:

- high critical functionalities, which are the parts of Core test cases or new functionality, implemented in build/release
- modules containing more bugs, more complex or more dependent
- areas which are highly accessed by the customer/end users
- positive test cases
- risk analysis

Test Case: Automation Status

It is impossible to automate all testing!

Good Candidates for Automation are:

- Repetitive test cases that run for multiple builds
- Test cases that run on several different configurations (hardware/OS)
- Frequently used functionality that introduces high risk conditions
- Test cases that tend to cause human error
- Test cases that require multiple data sets
- Test cases that are impossible to perform manually
- Test cases that take a lot of effort and time when manual testing

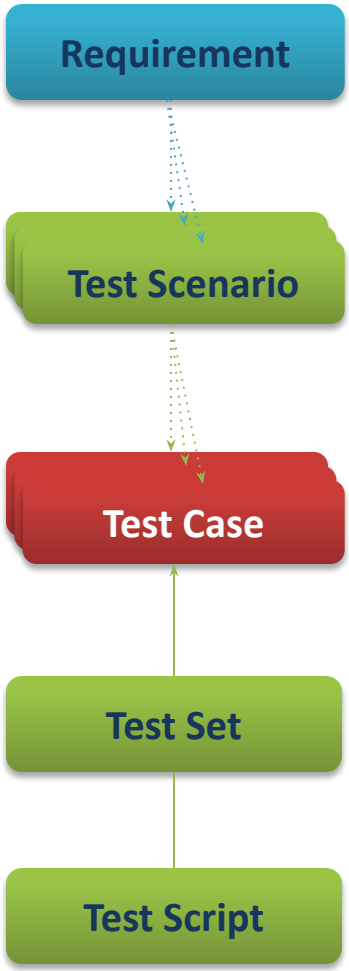
Bad Candidates for Automation are:

- If automation efforts are few times higher than manual execution
- Test cases that are only performed a few times
- Some test steps cannot be automated

TEST CASE IMPLEMENTATION

- Test Case Implementation Flow (http://www.youtube.com/watch?v=-b3Pj2IU5FI)
- Test Case Implementation Example
- Test Data Preparation

Test Case Implementation Flow



Test Case Implementation Example

Requirement : 18-55 patient should be able to post a request, which should be processed within 1 hour. If patient is 45-55 female with more then 1 child – request should be processed within 30 minutes.

Test Design

Requirement:	Scenario:	Test Case:	Objective:
Patient should be able to post a request.	Verify request can be posted by patient within valid age range	Verify request sending by female within valid age range	This test verifies that information dialog appears on 'Submit' action and request appears in 'Woman' category.
		Verify request sending by male within valid age range	This test verifies that information dialog appears on 'Submit' action and request appears in 'Man' category.
		Verify request sending by female within invalid age range	This test verifies that warning dialog appears on 'Submit' action. Request is not created in 'Woman' category.
		Verify request sending by male within invalid age range	This test verifies that warning dialog appears on 'Submit' action. Request is not created in 'Man' category.
		Verify request sending by patient with age which contains non-acceptable chars	This test verifies that error dialog appears on 'Submit' action.
		Verify request sending by patient without age defined	This test verifies that error dialog appears on 'Submit' action.
	Verify prerogatives for request posted by female	*<Test Case Name>*	*<Test Case Objective>*
	Verify prerogatives for request posted by patient with children	*<Test Case Name>*	*<Test Case Objective>*

Test Case Implementation Example

Test Case

Test Case ID: 0001	**Test Case Name:** Verify request sending by female within valid age range	**Status:** Pass
Test Type: Functional	**Author:** <First and Last Name>	**Creation Date:** 03/17/2013
Automation: Automated	**Priority:** High	**Disposition:** Reviewed-Completed

Objective:
This test verifies that information dialog appears on 'Submit' action and request appears in 'Woman' category.

Pre-Conditions:
- 'Reception' system is started;
- 'Patient Request' form is opened;

If not, run Test ID: 0156 to start.

	Test Steps:	Expected Results:
1	Select <Sex> option in 'Sex' section	
2	Set <Age> value in 'Age' field	
3	Click 'Submit' button on the form	a. 'Patient Request' form is closed; b. info message appears in dialog window: 'Your request has been successfully sent.'
4	Click 'OK' button in a pop-up window	Information dialog is closed.

Post-Conditions:
Request is available in the list of 'Patient Requests' in 'Woman' category.

Test Data:
<Sex>: female
<Age>: "18", "19", "35", "54", "55"

Attachment(s):

SoftServe
Ability. Agility. Advantage.

Test Data Preparation

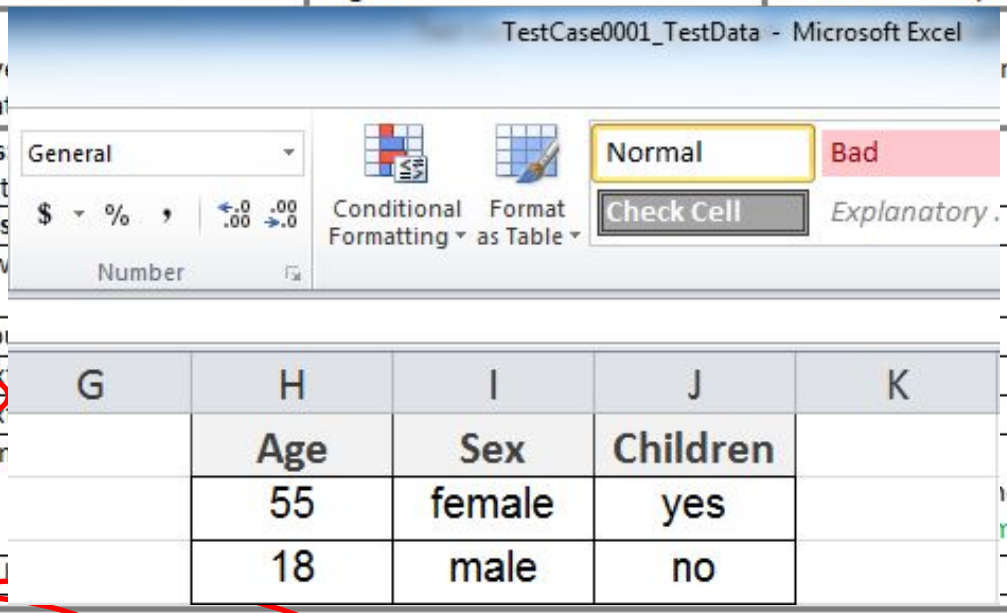
Test Case ID:	Test Case Name:	Status:
0001	Verify 'Patient Request' can be successfully submitted	Pass
Test Type:	**Author:**	**Creation Date:**
Functional	<First and Last Name>	02/03/2013
Automation:	**Priority:**	**Disposition:**
Automated	High	Reviewed-Completed

Objective:
This test case v… …n' page, filled with correct dat…

Pre-Conditions
'Reception' syst…

	Test Steps
1	Click 'New…
2	Se… #1 inpu…
3	Se… #2 (Sex…
4	Se… #3 (Sex…
5	Click 'Subr…
6	Click 'OK' …

…ndow: 'Your …nitted.'



G	H	I	J	K
	Age	Sex	Children	
	55	female	yes	
	18	male	no	

Test Data:
#1: TestCase0001Data.xls/Sheet1/Age
#2: TestCase0001Data.xls/Sheet1/Sex
#3: TestCase0001Data.xls/Sheet1/Children

Post-Conditions:
Request is added to the list of 'Patient Requests'.

Attachment(s):
TestCase0001Data.xls

Test Data Preparation

Test Data on Tester's disposal ☺

Test Case ID: 0001	**Test Case Name:** Verify 'Patient Request' can be successfully submitted	**Status:** Pass
Test Type: Functional	**Author:** <First and Last Name>	**Creation Date:** 03/17/2013
Automation: Not automated	**Priority:** High	**Disposition:** Reviewed-Completed

Objective:
This test case verifies that 'Patient Request' can be successfully created on 'Request Registration' page, filled with correct data and closed on 'Submit' action.

Pre-Conditions:
'Reception' system is started

	Test Steps:	Expected Results:
1	Click 'New Patient Request' icon	
2	Set age info in 'Age' field	
3	Set sex info in 'Sex' field	
4	Set children info in 'Children' field	
5	Click 'Submit' button on the form	a. 'Patient Request' form is closed; b. Info message appears in dialog window: 'Your request has been successfully submitted.'
6	Click 'OK' button on info dialog	Information dialog is closed.

Test Data:
<none>

Post-Conditions:
Request is added to the list of 'Patient Requests' either 'Woman' or 'Men' category based on input.

Attachment(s):
<none>

Pros
- time saving during test cases designing;
- time saving for experienced tester in specific area during test cases execution (not always, since some test cases require complex inputs, queries, etc.);
- important bugs can be found.

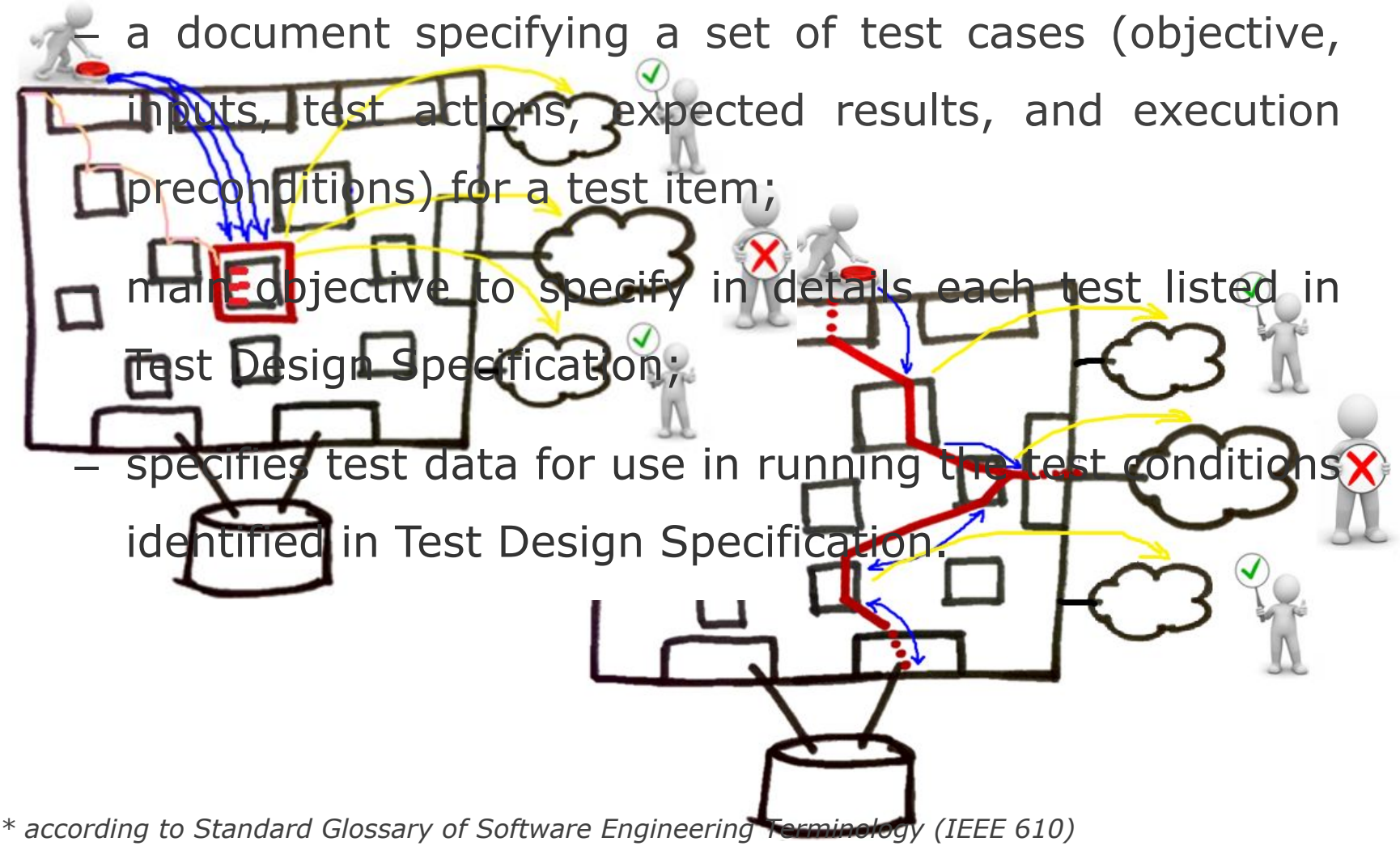
Cons
- time consuming for non-experienced tester in specific area during test cases execution;
- hard to entail issue due to chaotic inputs.

TEST CASE SPECIFICATION

- Test Case Specification - is …
- Test Case Specification consists of …

Test Case Specification – is …

- a document specifying a set of test cases (objective, inputs, test actions, expected results, and execution preconditions) for a test item;
- main objective to specify in details each test listed in Test Design Specification;
- specifies test data for use in running the test conditions identified in Test Design Specification.



** according to Standard Glossary of Software Engineering Terminology (IEEE 610)*

SoftServe
Ability. Agility. Advantage.

Test Case Specification consists of …

- Test Case Specification ID
- Test Items
- Input Specifications
- Output Specifications
- Special Procedural Requirements
- Inter-case Dependencies
- Environmental Needs

**** Test Case Specification might vary depending on particular project needs.***

Test Case Specification

section covers

- Unique "short" name for the test case
- Version date and version number of the test case
- Version Author and contact information
- Revision history

identifies

- the items or features to be tested by test case. References for source documents (Requirements Specification, Mock-ups, Users Guide etc) can be provided in the section as well

identifies

- all inputs required to execute the test case (Data, Tables, Human Actions, Conditions, Files etc)
- It is also acceptable to simplify the documentation process by using tables for elements, steps and values

Test Case Specification

identifies

- all outputs required to verify the test case
- outputs can also be simplified using tables as noted above and may even be included in the same table as the associated input to further simplify the documentation and improve its usefulness

consists of

- Hardware
  - » Configurations
  - » Limitations
- Software
  - » System (Operating systems, Compilers, Tools)
- Other Application
  - » Mix of applications
- Other
  - » Facilities
  - » Training

Test Case Specification

identifies

- Any special constraints on the test case(s)
- Special approach (in needed) for executing test case(s)
- Focus on key elements such as:
  - » Special Setup
  - » Operations intervention
  - » Output location and identification
  - » Special wrap-up

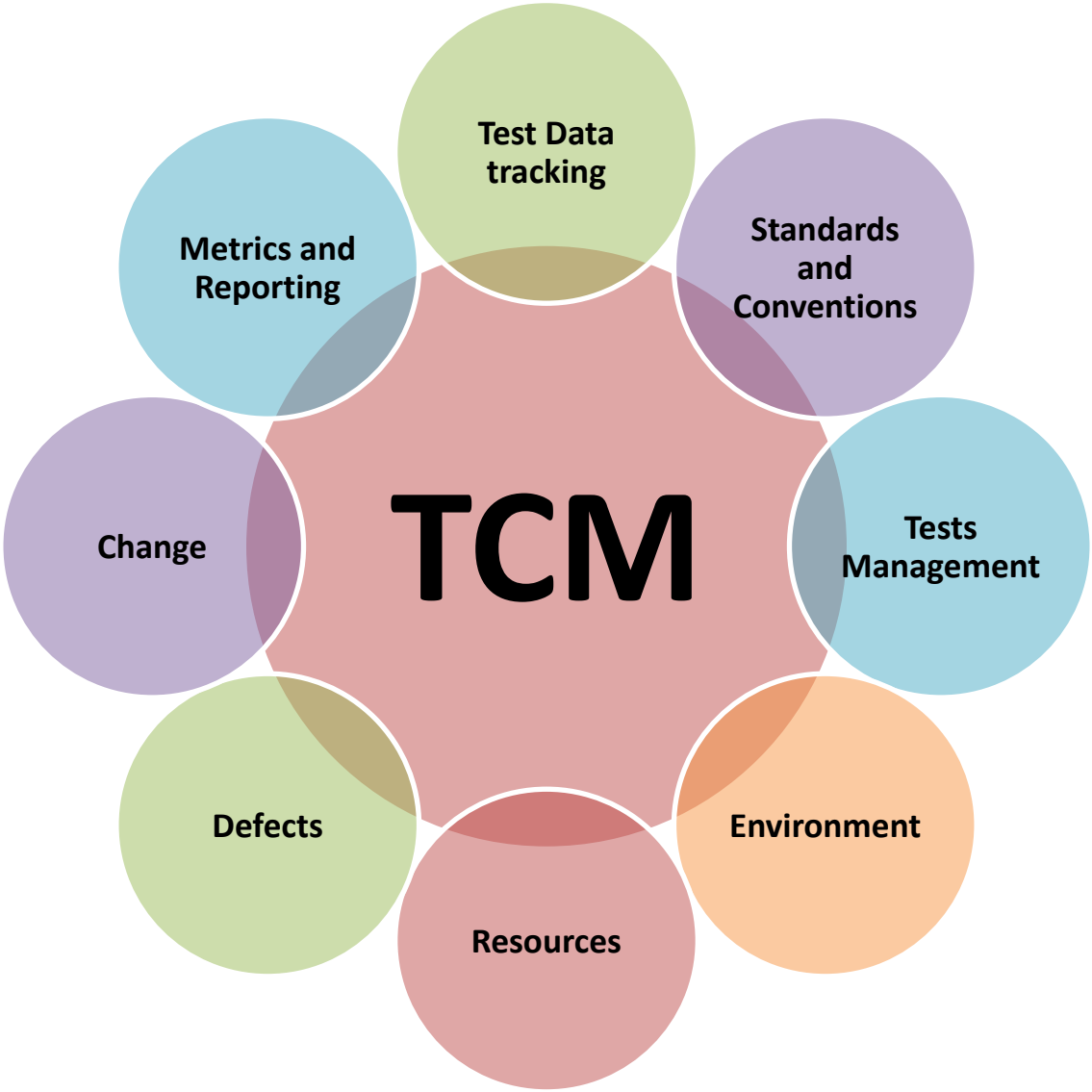
identifies

- Any prerequisites for test cases
- It is also recommended that the relationship of test cases be documented at both ends of the relationship.

TEST CASE MANAGEMENT TOOLS

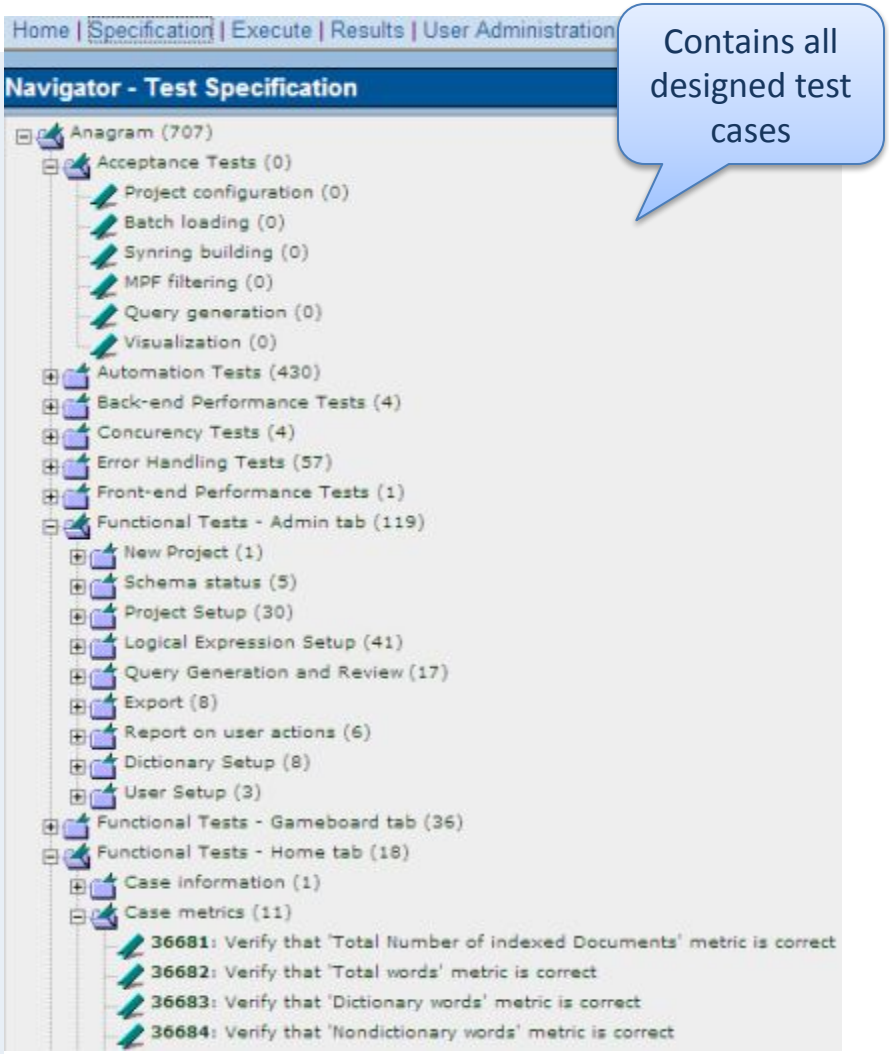
- TCM Tools assets
- Test Link
- MS Excel
- *Test Cases Examples*

Test Case Management Tool

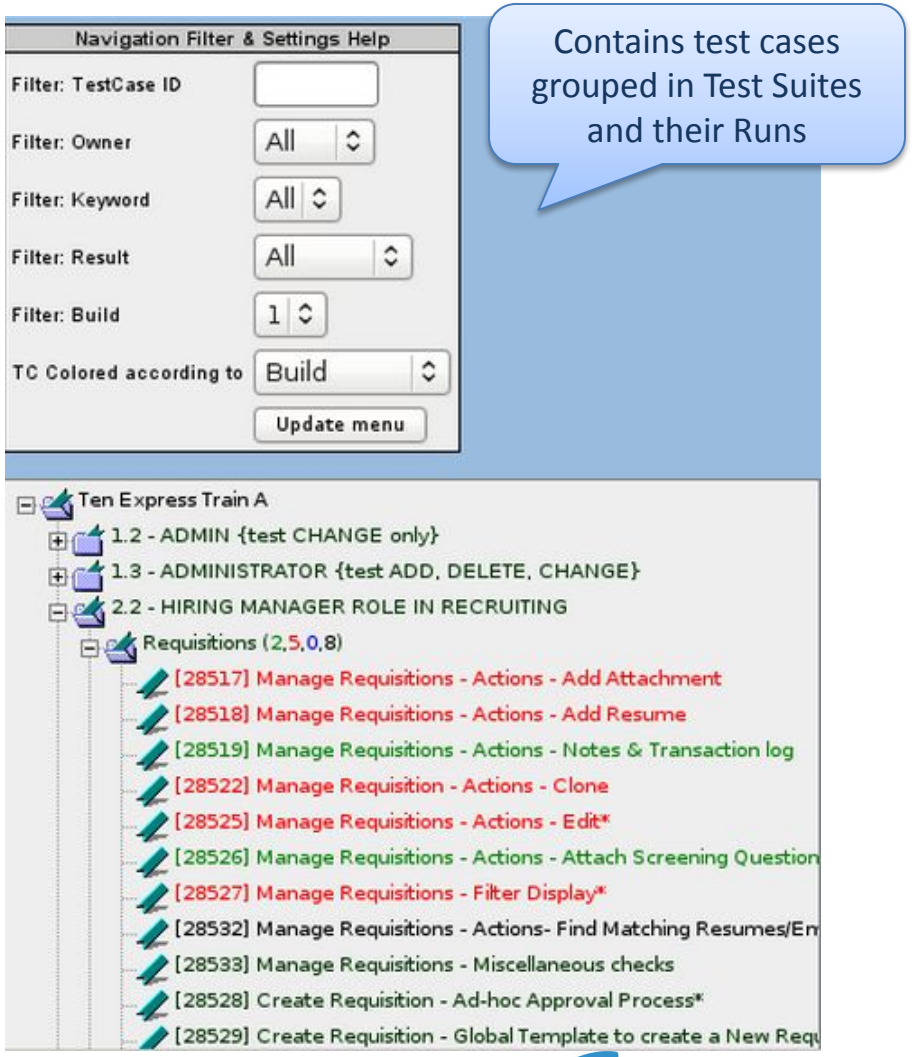


SoftServe
Ability. Agility. Advantage.

TestLink: Specification/Test Suites



Contains all designed test cases



Contains test cases grouped in Test Suites and their Runs

TestLink: Pros and Cons

- Test Cases to Requirements traceability
- Keywords classification
- Defects creation and linkage to bug tracking system
- Test Suites creation and run on different configurations
- Filtering abilities

- Non-flexible test cases creation and update possibilities
- Poor test execution reports
- Poor test data management
- Impossibility to edit test case while execution
- Test cases versioning is not supported
- Poor performance

MS Excel: Specification/Test Suites

One of the possible representations of test cases in Excel:

Test Cases in Excel

	ID	Priority	Test Case	Pre-Conditions	Test Data	Test Steps	Expected Results	Post-Conditions	Status	Actual Result	Defects	TOTAL	PASS	FAIL
2	**Scenario 1: Verify request can be posted by patient within valid age range**											0006	4	2
3	0001	High	Verify request sending by female within valid age range	- 'Reception' system is started; - 'Patient Request' form is opened; If not, run Test ID:0156 to start.	<Sex>: female <Age>: "18", "19", "35", "54","55"	1. Select <Sex> option in 'Sex' section 2. Set <Age> value in 'Age' field 3. Click 'Submit' button on the form 4. Click 'OK' button in a pop-up window	**3.** a. 'Patient Request' form is closed; b. info message appears in dialog window: 'Your request has been successfully sent.' **4.** Information dialog is closed.	Request is available in the list of 'Patient Requests' in 'Woman' category.	pass					
4	0002	High	Verify request sending by male within valid age range	- 'Reception' system is started; - 'Patient Request' form is opened; If not, run Test ID:0156 to start.	<Sex>: male <Age>: "21", "22", "50", "64","65"	1. Select <Sex> option in 'Sex' section 2. Set <Age> value in 'Age' field 3. Click 'Submit' button on the form 4. Click 'OK' button in a pop-up window	**3.** a. 'Patient Request' form is closed; b. info message appears in dialog window: 'Your request has been successfully sent.' **4.** Information dialog is closed.	Request is available in the list of 'Patient Requests' in 'Men' category.	fail	2. Error meesage is provided that Age value is out of valid boundaries.	DE1234			
5	0003	Medium	Verify request sending by female within invalid age range	- 'Reception' system is started; - 'Patient Request' form is opened; If not, run Test ID:0156 to start.	<Sex>: female <Age>: "0", "1", "17", "56","99"	1. Select <Sex> option in 'Sex' section 2. Set <Age> value in 'Age' field 3. Click 'Submit' button on the form 4. Click 'OK' button in a pop-up window	**3.** a. 'Patient Request' form is opened; b. warning message appears in dialog window: 'You are not allowed to send a request due to your age.' **4.** a. warning dialog is closed; b. 'Patient Request' is closed.	No new requests were added to the list of 'Patient Requests' in 'Woman' category.	pass					
6	0004	Medium	Verify request sending by male within invalid age range	- 'Reception' system is started; - 'Patient Request' form is opened; If not, run Test ID:0156 to start.	<Sex>: male <Age>: "0", "1", "20", "66","99"	1. Select <Sex> option in 'Sex' section 2. Set <Age> value in 'Age' field 3. Click 'Submit' button on the form 4. Click 'OK' button in a pop-up window	**3.** a. 'Patient Request' form is opened; b. warning message appears in dialog window: 'You are not allowed to send a request due to your age.' **4.** a. warning dialog is closed; b. 'Patient Request' is closed.	No new requests were added to the list of 'Patient Requests' in 'Men' category.	pass					
7	0005	Medium	Verify request sending by patient with age which contains non-acceptable chars	- 'Reception' system is started; - 'Patient Request' form is opened; If not, run Test ID:0156 to start.	<Sex>: female, male <Age>: "-1", "$", "65th", "1~!@#$%^&*", "2()_+}{:"?", "3><,./';][", "4-1`40V =$"	1. Select <Sex> option in 'Sex' section 2. Set <Age> value in 'Age' field 3. Click 'Submit' button on the form 4. Click 'OK' button in a pop-up window	**3.** a. 'Patient Request' form is opened; b. error message appears in dialog window: "Age' cannot include alpha or special chars. Please, check provided age info and try again.' **4.** a. error dialog is closed; b. 'Patient Request' form stays opened.	No new requests were added to the list of 'Patient Requests'.	pass					
8	0006	Medium	Verify request sending by patient without age defined	- 'Reception' system is started; - 'Patient Request' form is opened; If not, run Test ID:0156 to start.	<Sex>: female, male <Age>: <blank>	1. Select <Sex> option in 'Sex' section 2. Set <Age> value in 'Age' field 3. Click 'Submit' button on the form 4. Click 'OK' button in a pop-up window	**3.** a. 'Patient Request' form is opened; b. error message appears in dialog window: "Age' cannot be empty. Please, provide age info and try again.' **4.** a. error dialog is closed; b. 'Patient Request' form stays opened.	No new requests were added to the list of 'Patient Requests'.	fail	2. New request is created without age info.	DE1235			
9	**Scenario 2: Verify prerogatives for request posted by female**													

SoftServe
Ability. Agility. Advantage.

MS Excel

- Many organizations use **Microsoft Excel** to create and manage Test Cases. Test Case Specifications are stored in repositories

- Excel can be easily customized (e.g.: additional fields can be added)
- It is easy to track changes (and versions)
- Important fields/notes/etc can be highlighted (e.g.: masked in different colors, bolded etc)
- Test cases and execution results are stored in the same place, so it is easy to make updates (even while execution)

- Test cases included in Excel document cannot be updated simultaneously
- No automatic linking: requirement – test case, issue – test case
- Excel limits in number of steps (long text is not fully visible in cell)

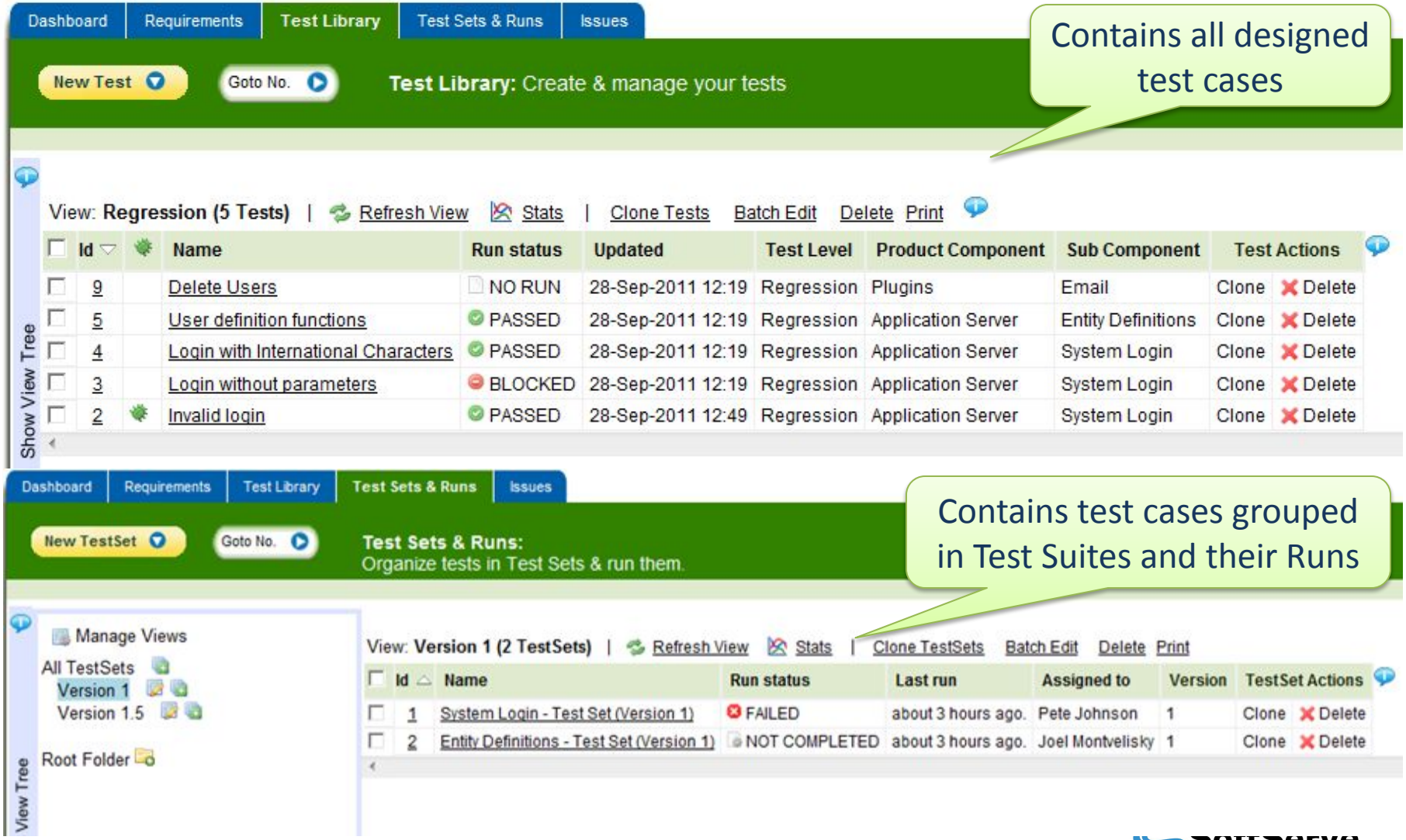
SoftServe
Ability. Agility. Advantage.

PractiTest

- **PractiTest** – the complete & lightweight SaaS solution for QA Management, SaaS Test Management, Issue Tracking and Requirements

- **5 key features of PractiTest**
  - Flexible Hierarchical Views
  - Fields and Workflow Customization
  - Intelligent Anti-Duplication Bugs Mechanism
  - Bugs submitting by Email
  - Super-Fast Bug Review

PractiTest: Test Library



SoftServe
Ability. Agility. Advantage.

PractiTest: Pros and Cons

- Flexible Test Cases design and management
- Test Cases to Requirements traceability
- Issues creation and linking possibility
- Fields and Workflow Customization
- Powerful test execution reports
- Summary Visualization within Dashboard
- Filtering and *many others*

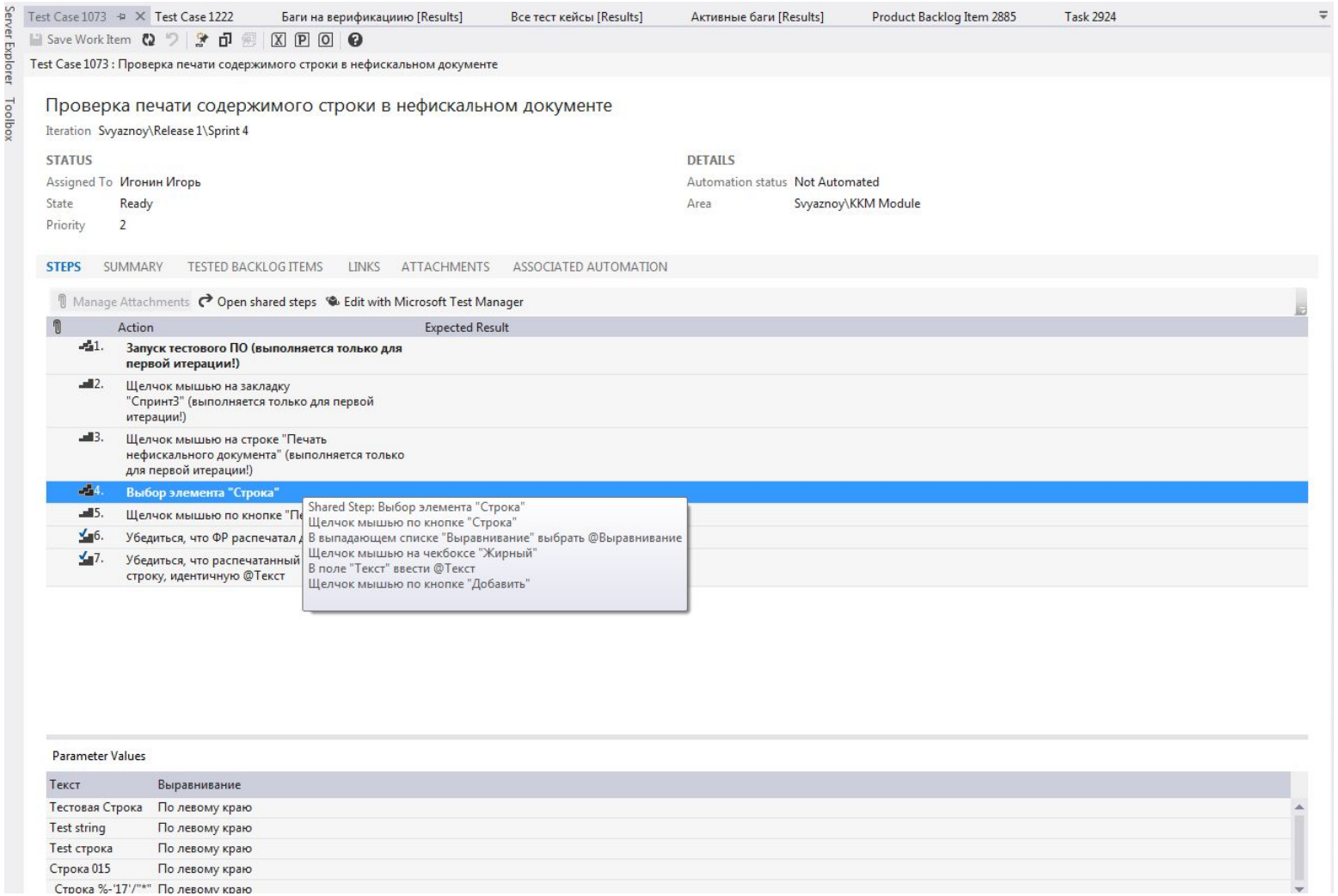
- Absence of On-Premise solution

  PractiTest is SaaS based without ability of in-house hosting
- Not user friendly interface while test steps execution

  e.g.: each test case step represents in separate table
- Impossibility to edit test case while execution

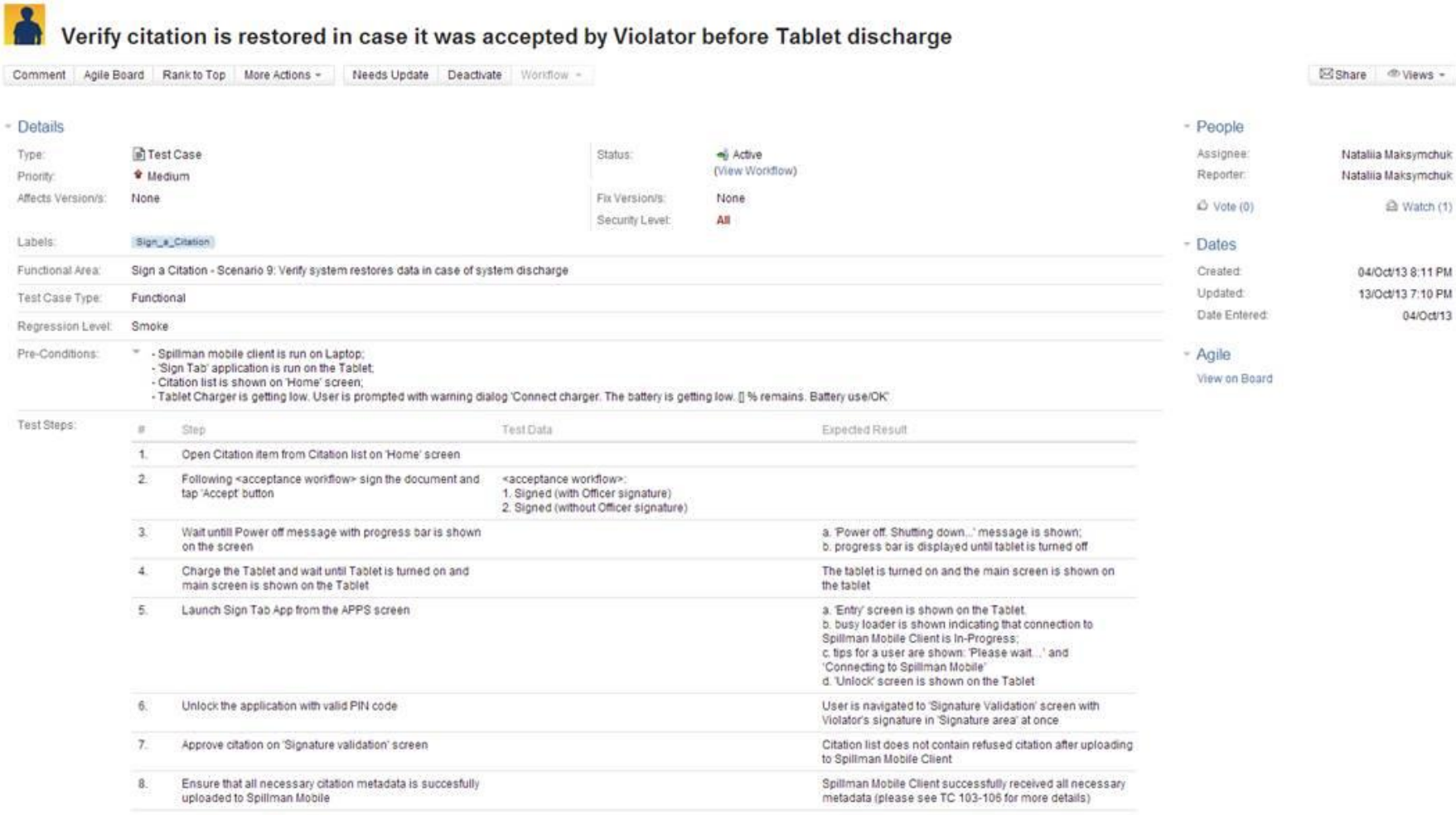
SoftServe
Ability. Agility. Advantage.

Example №1



SoftServe
Ability. Agility. Advantage.

Example №2



Verify citation is restored in case it was accepted by Violator before Tablet discharge

Comment | Agile Board | Rank to Top | More Actions | Needs Update | Deactivate | Workflow | Share | Views

Details

Type:	Test Case	Status:	Active (View Workflow)
Priority:	Medium		
Affects Version/s:	None	Fix Version/s:	None
		Security Level:	All
Labels:	Sign_a_Citation		
Functional Area:	Sign a Citation - Scenario 9: Verify system restores data in case of system discharge		
Test Case Type:	Functional		
Regression Level:	Smoke		

Pre-Conditions:
- Spillman mobile client is run on Laptop;
- 'Sign Tab' application is run on the Tablet;
- Citation list is shown on 'Home' screen;
- Tablet Charger is getting low. User is prompted with warning dialog 'Connect charger. The battery is getting low. [] % remains. Battery use/OK'

Test Steps:

#	Step	Test Data	Expected Result
1.	Open Citation item from Citation list on 'Home' screen		
2.	Following <acceptance workflow> sign the document and tap 'Accept' button	<acceptance workflow>: 1. Signed (with Officer signature) 2. Signed (without Officer signature)	
3.	Wait untill Power off message with progress bar is shown on the screen		a. 'Power off. Shutting down...' message is shown; b. progress bar is displayed until tablet is turned off
4.	Charge the Tablet and wait until Tablet is turned on and main screen is shown on the Tablet		The tablet is turned on and the main screen is shown on the tablet
5.	Launch Sign Tab App from the APPS screen		a. 'Entry' screen is shown on the Tablet. b. busy loader is shown indicating that connection to Spillman Mobile Client is In-Progress; c. tips for a user are shown: 'Please wait...' and 'Connecting to Spillman Mobile' d. 'Unlock' screen is shown on the Tablet
6.	Unlock the application with valid PIN code		User is navigated to 'Signature Validation' screen with Violator's signature in 'Signature area' at once
7.	Approve citation on 'Signature validation' screen		Citation list does not contain refused citation after uploading to Spillman Mobile Client
8.	Ensure that all necessary citation metadata is succesfully uploaded to Spillman Mobile		Spillman Mobile Client successfully received all necessary metadata (please see TC 103-106 for more details)

People

Assignee:	Nataliia Maksymchuk
Reporter:	Nataliia Maksymchuk

Vote (0) | Watch (1)

Dates

Created:	04/Oct/13 8:11 PM
Updated:	13/Oct/13 7:10 PM
Date Entered:	04/Oct/13

Agile

View on Board

SoftServe
Ability. Agility. Advantage.

Example №3

Validation Criteria

#	Title	Pre-condition	Action	Expected Result	Automatization Test
1	Verify that home page with first 20 records is displayed	1. User has authority to access Patient Domain. 2. User is logged in OMNI 360 Viewer. 3. OMNI 360 home page is displayed.	1. User verifies that first Patient 20 records are displayed in home page	1.1. System displays first Patient 20 records 1.2. It is possible to filter, change records count and navigate between pages of Patient records.	Yes
2	Verify that Patient domain is visible in Quick Search drop down	1. User has authority to perform Quick Search on Patient Domain. 2. User is logged in OMNI 360 Viewer. 3. Quick Search section is displayed.	1. User clicks on the drop down list of domains	1.1. System displays the list of domains. 1.2. "Patient" domain is displayed.	Yes

Example №4

Verify registration of user after filling in all fields and clicking 'Submit' button	1) Navigate to the registration page 2) Enter 'Email' (Alphabet + numbers + correct email format) 3) Enter 'Password' (Alphabet + numbers) 4) Enter 'First Name' (Alphabet + numbers) 5) Enter 'Last Name' (Alphabet + numbers) 6) Enter 'CAPTCHA' (the same as on CAPTCHA image) 7) Click "License agreement" link 8) Close License agreement page and back to registration tab 9) Check "I agree with License agreement" checkbox 10) Click 'Submit' button 11) Click 'OK' button 12) Check entered email box for received welcome email	3) Password is displayed in asterisks 6) "Correct" message is displayed under the field in green color 7) License agreement page is opened 9) "Submit" button is enabled 10) Registration is successful. Pop-up with following message is displayed: "Thank You just one more thing left to do Click the link in the email to complete the sign up. Check your spam folder in case you dont see it " 11) Pop-up is closed, user is navigated to Landing page 12) Welcome email is delivered from admin@Libreeze.com address

SoftServe
Ability. Agility. Advantage.

BEST PRACTICES

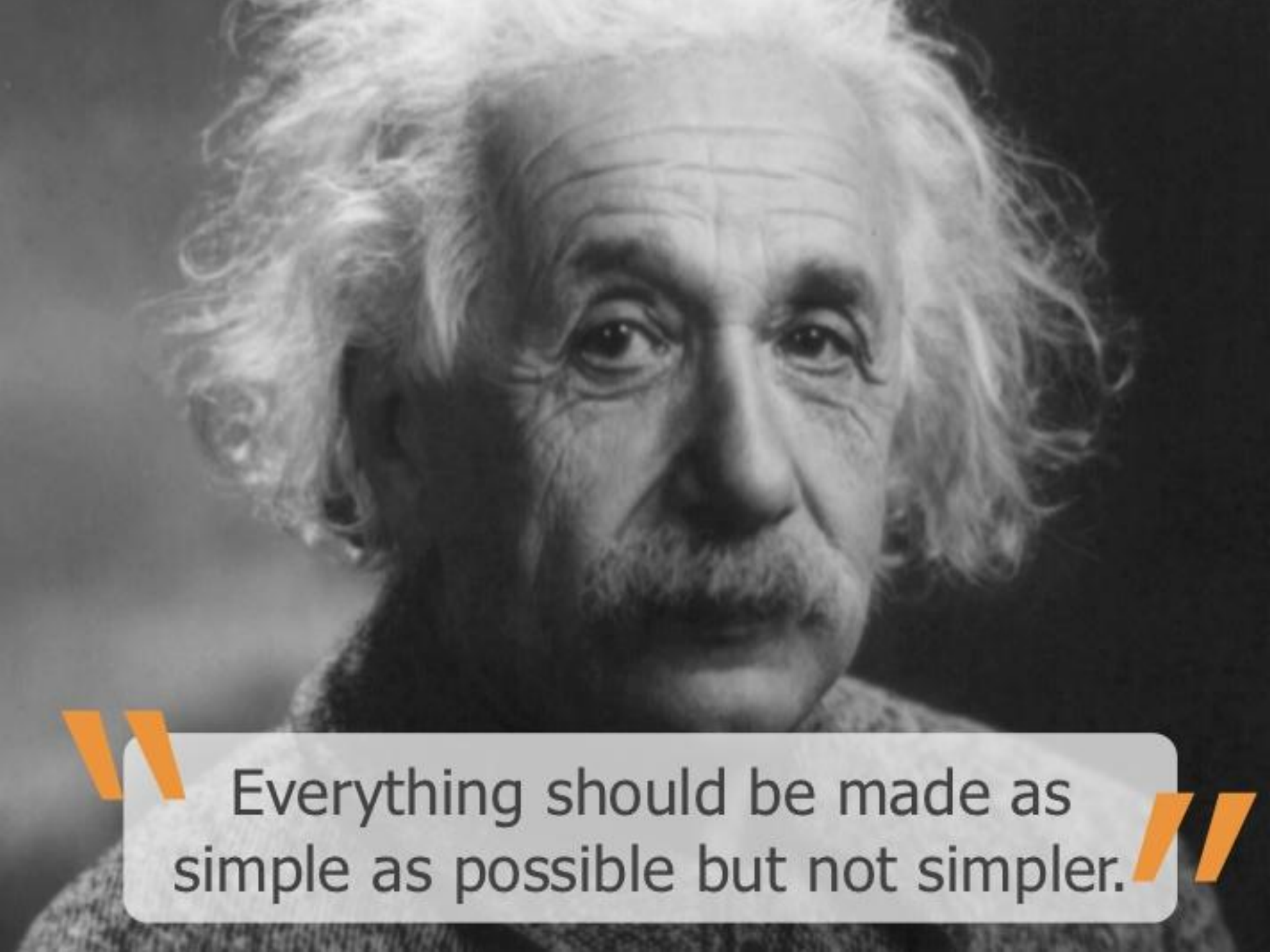
SoftServe
Ability. Agility. Advantage.

Best Practices

- Write test cases for all possible test conditions of test item
- Write test cases with necessary level of detail:
  - Detailed test cases if automation is going to have place
  - Detailed test cases if any legal compliance standards to testing on project
  - Non-detailed test cases (or just test objectives) if test case won't be executed often or step setup is described in another document (link to the document should be provided)
- Write test cases independent and cross-platform:
  - The last test case step should return system to the state before executing this test case
  - It is possible to re-order test cases without additional steps or data setup
  - Tests should be cross-platforms as reasonably possible, working across different devices, platforms, screen resolutions, etc
- Write test cases according to STANDARD TEMPLATE (project convention)

Best Practices

- Write short test cases (up to 10-15 steps)
- Write steps using simple English and general words
- Write ACCURATE test cases
- Write test cases so that expected results are easy to interpret
- Provide TEST DATA if possible or where it makes much sense
- Add reference to bugs and requirements
- Add notes/highlight things if you want to convey / pay attention to some info
- Write detailed SQL queries (it will save time while executing)
- Keep test cases UP TO DATE ☺


"Everything should be made as simple as possible but not simpler."