Agile values and principles with Xray

Agile follows a set of values and principles that allow teams to collaborate and have more agility while producing better, working software more tailored to customer needs.

Manifesto for Agile Software Development

Back in 2001, a group of people wrote a manifesto to make better software development. Agile software development is backed up by 4 values and 12 principles depicted in the Manifesto for Agile Software Development, quoted next.

*We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:*

Individuals and interactions *over processes and tools*
Working software *over comprehensive documentation*
Customer collaboration *over contract negotiation*
Responding to change *over following a plan*

*That is, while there is value in the items on
the right, we value the items on the left more.*

There are several frameworks following Agile values/principles, including Scrum, Kanban, among other ones. However, Nowadays, Agile is used in broader scope as a way to bring agility in some context.

The key for having agile teams is having people at center of your process and providing them the means to work effectively together. Also allowing them to learn and adapt, both to their needs and to customer needs.

Being Agile requires commitment and focus, working as a team on what matters, keeping communication always present and delivering value often.

Transparency, continuous feedback, continuous improvement, technical excellence and ownership are all important to make sure the team produces the right product, the right way.

Therefore, how can a tool support these values and these principles?

Individuals and interactions over processes and tools

Tools and processes don't solve your problems just by themselves. Anyone trying to sell you a tool or some amazing processes as being a silver bullet isn't telling you the whole truth.

Individuals, interactions through communication and collaboration are way more important.

However, this does not mean that a tool or that some process can't help you out. Tools can assist you; however, they should not put you in a straitjacket. They should give the means so that you can use them in the way that works best for your team. This is the approach that Xray follows: you can follow whatever test process you want, more or less strict, using more or less entities and using them in a way that meets your needs.

That's why Xray provides some reference guidelines and some usage options but it does not tell you how to use it in a standardized way. This gives you freedom to adapt, so you can be more agile.

Having individuals and interactions in mind, with Xray, all team members (testers included) are invited to collaborate on any issue being handled, no matter if it's a Story, a Bug, a Test or a Test Plan.

In fact, Xray does not differentiate a tester from any other role within the team; they're equal team members that have assigned or self-assigned tasks and that can easily collaborate with each other.

Working software over comprehensive documentation

Agile team focus is to provide working software (i.e. deliver value), as frequently as possible. Therefore, writing long and comprehensive documentation upfront may not be the best approach as you may be documenting features that will change substantially during your software development life cycle. Most of these documents are never read and get outdated a few moments after you start development.

So, instead of doing extensive documents, Agile teams will write the necessary documentation, the one that actually delivers value.

One way of doing documentation that provides real value is by writing test cases that validate our SUT.

Tests are somehow living documents that explain how the system is built and how it should behave. Tests at lower levels of the Test Automation Pyramid are more focused on technical aspects, somehow similar to the left-side quadrants of the Agile Testing Quadrants; these tests are mostly used to support the team. On the other hand, other types of tests, such as E2E/UI based tests, provide more customer oriented test cases (i.e. real usage scenarios).

All tests implicitly provide some sort of documentation on how the system should work. Automated tests, as long they are written in a clear way, are living, executable documents that constantly describe and validate the system. Xray provides the support for many different automation frameworks based reports , which in turn allow you to choose whatever language plus framework combination to implement automated tests.

TestNG unit JUnit ROBOT FRAME WORK/ cucumber

Selenium/WebDriver

▶ Execution Details

Test Description

None

Test Details

Test Type:	Generic
Definition:	Robotframework-Webdemo.Login Tests.Valid Login.Valid Login

Results

Context	Error Message	Duration	Status
Open Browser To Login Page		5 sec	PASS
Input Username		1 sec	PASS
Input Password		1 sec	PASS
Submit Credentials		3 sec	PASS
Welcome Page Should Be Open		1 sec	PASS
Close Browser		562 millisec	PASS

One of the automated frameworks natively supported is Cucumber (or Gherkin based frameworks in general) which is used in BDD context. Documentation "produced" by BDD is simple, overall clear and yet focused on real usage scenarios.

Customer collaboration over contract negotiation

BDD is an approach where technical and non-technical users collaborate together to have clear understanding of requirements and their acceptance criteria, by writing concrete usage examples/scenarios in natural language that can be later on executed by an automation engine. These examples describe the intended behavior of the system, in a way that can be easily understood by all "stakeholders".

Xray has native support for Cucumber and other Gherkin based frameworks (e.g. Behave, Calabash, SpecFlow), allowing customers to collaborate with the product/development team by writing and reviewing Scenarios.

Test Details

Type:	**Cucumber**
Scenario Type:	**Scenario Outline**
Scenario:	Given I have entered <input_1> into the calculator
	And I have entered <input_2> into the calculator
	When I press <button>
	Then the result should be <output> on the screen

```
Examples:
  | input_1 | input_2 | button | output |
  | 20      | 30      | add    | 50     |
  | 2       | 5       | add    | 7      |
  | 0       | 40      | add    | 40     |
  | 4       | 50      | add    | 54     |
  | 5       | 50      | add    | 55     |
```

But besides supporting BDD, Xray also promotes collaboration on any work item, including user stories, bugs, and all testing related entities (e.g. Tests, Test Executions, Test Plans). Since Xray uses issue types as a way to abstract testing related artifacts, you can leverage the built-in collaboration features, among others, that Jira provides for any issue in general.

Activity

All	**Comments**	Work Log	History	Activity	Transitions

∨ ⬤ Administrator added a comment - In 1 hour   🔗 ✏ 🗑

This plan should cover the risky areas, namelly the core operations related with basic arithmetic operations.

Responding to change over following a plan

Agility means being able to quickly adapt as changes arise because customers can continuously provide feedback in order to have a more tailored product /system that addresses their needs.

Instead of having a well-defined plan built at the start, taking into account a list of highly detailed requirements defined in advance, Agile teams iterate and deliver value frequently based on customer feedback that constantly is used as feedback loop to build a better, more suitable product.

If working in a Scrum context, teams iterate on sprints and can use Test Plans and/or Test Executions on a per sprint level or even use Sub-Test Executions to manage their work, in a visible way, in the scope of the sprint.

But being able to respond to change, means that Tests can be written at any moment and added to respective Test Plan(s).

In fact, if you're implementing more and more automated tests as a means to add additional coverage, they can be dynamically added to Test Plans along with their results, as soon as you start running them in your CI/CD pipeline. Thus, you don't have to plan ahead what exact list of test cases you'll need. You may assess the risks (at the start and during development), enumerate them in your Test Plan and use them to create Tests and execute them at any moment.

Test Plan with automated tests for sum operation

Edit | Comment | Trigger Bamboo Build | More ▾ | Start Progress | Resolve Issue | Close Issue | Admin ▾

Details

Type:	▣ Test Plan	Status:	**REOPENED** (View Workflow)
Priority:	⌃ Major	Resolution:	Unresolved
Affects Version/s:	None	Fix Version/s:	v3.0
Component/s:	None		
Labels:	None		
Sprint:	Sprint 1		
Test Count:	6		

Description

Risks/sensible areas to cover:

- addition operation in basic mode
- addition operation in scientific mode
- handling of negative numbers

Tests

≡ Test Plan Board | + Create Test Execution ▾ | + Add ▾

Overall Execution Status

5 PASS 1 TODO

	Key	Summary	Requirements	#Test Executions	Issue Assignee	Latest Status
▶	CALC-1202	CanAddNumbers	CALC-1178	61	Administrator	PASS
▼	CALC-1180	Cucumber Test As a user, I can calculate the sum of 2 numbers	CALC-1178	39	Administrator	PASS

Key	Summary	Environment	Status
CALC-4369	Test Execution for cucumber Execution	None	PASS
CALC-2958	Test Execution for cucumber Execution	None	PASS
CALC-2095	Test Execution for cucumber Execution	None	PASS
CALC-2093	Test Execution for cucumber Execution	None	PASS
CALC-2061	Test Execution for cucumber Execution	None	PASS
CALC-2055	Test Execution for cucumber Execution	None	PASS
CALC-2036	Test Execution for cucumber Execution	None	FAIL
CALC-1918	Test Execution for cucumber Execution	None	PASS

Scrum values

Focus

Everyone focuses on the work of the Sprint and the goals of the Scrum Team.

Team members must know what they have to do, focus on getting it done right and not loose time with other non-essential tasks.

- From specification, to organization, execution and planning, all testing related entities including Test, Test Set, Test Execution, Test Plan, Test Runs can be assigned; users get notified and can track their work (what they have to do and what they have already done). All Xray issue type based entities can also be watched if you want to follow more closely what is happening.

- If you want, you may configure email notifications to get notified upon relevant events, such as knowing that you're assigned some Test Run(s)

Testing related reports, should be automatable, shareable or easily deliverable, so that team members can focus on tasks where they can bring more value.

- Use Jira dashboards with real-time information that can be shared with all your team members

- Create, if really needed, reports as document assets, that can be downloaded, sent and stored (for regulatory/conformance reasons)
  - After having an Xporter report template, you can manually generate a report from the issue screen (e.g. from a Test Plan) but you can also automate the creation of the report to make it happen upon the transition of a workflow or even on a periodic basis.

If your testers/SDETs/developers are implementing automated tests then should be focused on doing the right tests and implementing them; they should not loose time on side activities for having visibility of those results being reflected in Jira/Xray. Therefore, with Xray they can easily submit the automated test results using either one of the free CI plugins or by using the REST API directly. Upon submission, Xray will automatically, if necessary and possible, create Test entities mapping to each automated test. It is also possible to link automated tests to stories (or any "requirement" like issue) right from the automated test code. All this lets your team focus on what matters most.

```
@DataProvider
public Object[][] ValidDataProvider() {
    return new Object[][]{
        {  1, 2, 3 },
        {  2, 3, 4 },  // error or the data itself :)
        { -1, 1, 0 }
    };
}


@Test(dataProvider = "ValidDataProvider")
@Xray(requirement = "CALC-1234", test = "CALC-1")
public void CanAddNumbersFromGivenData(final int a, final int b, final int c)
{
    Assert.assertEquals(Calculator.Add(a, b), c);
}
```

▶ Execution Details

Test Description   ⌄

Test Issue Links (1)   ⌃

tests   ⊙ CALC-1234 As a user, I can calculate the sum two numbers   OPEN

Test Details   ⌃

Test Type: Generic
Definition: com.xpand.java.CalcTest.CanAddNumbersFromGivenData

Results   ⌃

Context	Error Message	Duration	Status
TestAll - calculator (1,2,3)	–	0 millisec	PASS
TestAll - calculator (2,3,4)	java.lang.AssertionError: expected [4] but found [5] at org.testng.Assert.fail(Assert.java:93) at org.testng.Assert.failNotEquals(Assert.java:512) at org.testng.Assert.assertEqualsImpl(Assert.java:134)	1 millisec	FAIL

Openness

The Scrum Team and its stakeholders agree to be open about all the work and the challenges that come with performing the work.

Being open means that everyone should be able to provide feedback, no matter if you're authoring new Test cases or if you're building your Test Plan.

Xray provides the ability for any team member to collaborate on any testing activity, including, and not limited to, Test Runs and Test Plans.

Manual Test As a user, I can calculate the sum of two numbers

Execution Status ▮ **FAIL**   ⇄ ▯ ▯ ▯ ▯ ▯ ▯ ▯ ▯

Started On: **18/Jan/19 11:57 AM** 📅   Finished On: **18/Jan/19 11:57 AM**

Comment	Preview Comment	^

This may be a false negative due to problems with the environment.

✓ ✗

Execution Defects (0)   Create

No def

Activity

All	**Comments**	Work Log	History	Activity	Transitions

∨ ● Administrator added a comment - In 1 hour   🔗 ✏ 🗑

This plan should cover the risky areas, namely the core operations related with basic arithmetic operations.

Respect

Scrum Team members respect each other to be capable, independent people.

Team members work on their assigned tasks and the team trusts them to deliver their work. However, each team member does not work in isolation; the team collaborates in order to successfully complete each task. How? No matter the role, team members can contribute at any moment.

- Developers:
    - can interact and leave comments on user stories, on Test cases, on Test Plans
    - review PRs
- Testers:
    - can interact and leave comments on user stories and ask for clarifications, define and review Test cases
    - can define the goals of Test Plans, based on risk assessment, discussed together with the team
- PO:
    - review comments on user stories
    - clarify acceptance criteria on user stories
    - work with team to identify the risks, so a proper Test Plan can be built accordingly

Courage

Scrum Team members have the courage to do the right thing and work on tough problems.

Courage is, first of all, about transparency of the work and its progress, including failures that may happen.

Thus, testing progress can be promptly and automatically evaluated at the Test Execution, at Test Plan or even at user story level.

Details

Type:	🟧 Test Plan	Status:	**OPEN** (View Workflow)
Priority:	⌃ Major	Resolution:	Unresolved
Affects Version/s:	None	Fix Version/s:	v3.0
Component/s:	None		
Labels:	None		
Test Count:	3		

Description

Click to add description

Tests

[≡ Test Plan Board]   [+ Create Test Execution ▾] [+ Add ▾]

Overall Execution Status

3 PASS

Details

Type:	⊡ Story	Status:	**OPEN** (View Workflow)
Priority:	⌃ Major	Resolution:	Unresolved
Affects Version/s:	None	Fix Version/s:	v3.0
Component/s:	None		
Labels:	None		
Requirement Status:	v3.0 - OK		

Description

Click to add description

Test Coverage   +

[Create Test] [Create Sub-Test Execution] [Link Tests]

Version ▾   [None (latest execution) ▾]   All Environments ▾   OK

⌃	OPEN	Unresolved	CALC-3211	Cucumber Test As a user, I can calculate the sum of two numbers	PASS
⌃	OPEN	Unresolved	CALC-3209	Manual Test As a user, I can calculate the sum of two numbers	PASS
⌃	OPEN	Unresolved	CALC-3212	Generic Test As a user, I can calculate the sum of two numbers	PASS

Courage to be receptive to feedback from other team members is also a value of Scrum; Xray promotes this by enabling users to leave comments on any testing entity.

Activity

[All | **Comments** | Work Log | History | Activity | Transitions]

⌄ ⚙ Administrator added a comment - In 1 hour   🔗 ✏ 🗑

I think we need to review these steps to cover a more common scenario.

Courage also means "to be willing to change, if needed". Therefore, users can:

- explicitly abort a Test Run with a specific custom status, to make it more visible
- signal some Test Execution using a specific workflow status while leaving comments on it
- make any testing artifact read-only or writable, using the Jira way of managing permissions

To make sure you're working on the right thing and that it is properly tested, so you can deploy confidently, you may use the Overall Coverage Report to give you real-time insights of real status of the user stories or other items being worked out/tested. You can group your covered items (e.g. stories, epics, bugs, etc) by Priority, by component of the system or by any other field; this gives you

Overall Requirement Coverage Report Switch report ▾

Scope: version; **Version:** v3.0; **Environment:** All Environments ▾ | Component/s ▾ | **Fix Version** v3.0 ▾ | 🗨 How to read this report

| None |
| **Native** |
| **Priority** |
| Status |
| Resolution |
| **Custom Fields** |
| Epic Status |
| Flagged |
| Requirement distinguish |
| UI elements |
| size |

Legend: ■ OK ■ NOK ■ Not Run ■ Unknown ■ Uncovered

None — 41.9% (57) ... % (54) ... 14.7% (20)

core — 75% (6) ... 25% (2)

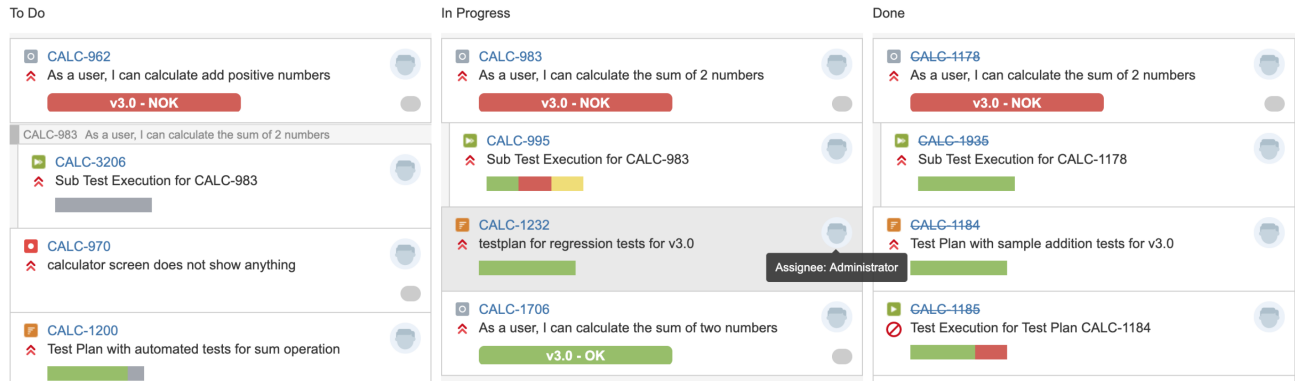**UI**

Scale: 0 10 20 30 40 50 60 70 80 90 100

Commitment

People personally commit to achieving the goals of the Scrum Team.

Test Plans, Test Executions can be assigned to a sprint, so you can track in your active sprint Board. As they're issues, they can be assigned to a certain user, who will be responsible for managing the life cycle of each one of them.

Calculator Board Enhanced with Testing
Sprint 1

🕐 0 days remaining Complete Sprint | Board ▾ | ⌃

QUICK FILTERS: Only My Issues   Recently Updated

To Do

- CALC-962 — As a user, I can calculate add positive numbers — **v3.0 - NOK**
- CALC-983 As a user, I can calculate the sum of 2 numbers
  - CALC-3206 — Sub Test Execution for CALC-983
- CALC-970 — calculator screen does not show anything
- CALC-1200 — Test Plan with automated tests for sum operation

In Progress

- CALC-983 — As a user, I can calculate the sum of 2 numbers — **v3.0 - NOK**
- CALC-995 — Sub Test Execution for CALC-983
- CALC-1232 — testplan for regression tests for v3.0   Assignee: Administrator
- CALC-1706 — As a user, I can calculate the sum of two numbers — **v3.0 - OK**

Done

- CALC-1178 — As a user, I can calculate the sum of 2 numbers — **v3.0 - NOK**
- CALC-1935 — Sub Test Execution for CALC-1178
- CALC-1184 — Test Plan with sample addition tests for v3.0
- CALC-1185 — Test Execution for Test Plan CALC-1184

Calculator / CALC-1232
testplan for regression tests for v3.0

[✏ Edit] [💬 Comment] [Trigger Bamboo Build] [More ▾] | [Stop Progress] [Resolve Issue] [Close Issue] | [Admin ▾]

Details

Type:	📄 Test Plan	**Status:**	**IN PROGRESS** (View Workflow)
Priority:	⌃ Major	**Resolution:**	Unresolved
Affects Version/s:	None	**Fix Version/s:**	v3.0
Component/s:	None		
Labels:	None		
Sprint:	Sprint 1		

Individual Test Runs, which are part of Test Executions, can also be assigned to specific users. Thus, these two levels of assignment (at Test Execution and at Test Run level) provide finer control over the work that needs to be performed.

	Rank	Key	Summary	Test Type	#Req	#Def	Assignee	Status		
☐	1	CALC-3209	Manual Test As a user, I can calculate the sum of two numbers	Manual	1	2	André Rodrigues	FAIL	▶	···
										👤 Assign
										↑↓ Move to...
										🗑 Remove
☐	2	CALC-3212	Generic Test As a user, I can calculate the sum of two numbers	Generic	1	0	Administrator	PASS	▶	···
☐	3	CALC-3211	Cucumber Test As a user, I can calculate the sum of two numbers	Cucumber	1	0	Administrator	PASS	▶	···

While working in the context of a Test Plan or of a Test Execution, users can track their progress and thus are able to evaluate the "progress of their commitment".

Tests

+ Add ▾

Overall Execution Status

1 PASS **1** FAIL **1** EXECUTING

References

- https://agilemanifesto.org/
- https://www.scrumalliance.org/learn-about-scrum/scrum-values
- https://www.neotys.com/blog/the-four-values-of-the-agile-manifesto/
- https://guntherverheyen.com/2013/05/03/theres-value-in-the-scrum-values/