Manual for User Acceptance Testing

To ensure ease of work and fluid flow of development we prepared this document describing testing and QA process in Actum on Walmark Line project.

What is testing

Testing is part of development process. It is present at all stages of application and development lifecycle. Developers test their code, analysts and project managers test stories, quality assurance is testing if application is functioning according to acceptance criteria and customer is testing if application met his expectations.

Testing is process in which we assert if application is working and looking as designed.

What we should report

Golden rule states that reporting more than less is always better, but you should avoid reporting duplicities. If there is some issue with one part of application on multiple places, one ticket is enough.

Also group issues if they are for same functionality (i.e. validation on two different forms for name are not working) or one part of application (i.e. validation for name and phone number on one form are not working correctly) by using one part of web repeatedly. Only one bug report is enough for this issue.

To make absolutely sure that you are not reporting duplicate you should also look if issue you are reporting is already reported by QA or someone from UAT and make sure that only one person is testing specific feature.

What is bug and what is change request

Bug is anything where application is not working as specified in acceptance criteria, if functionality should change from acceptance criteria or is not in acceptance criteria at all its considered change request.

JIRA

If some bug is found during testing it has to be reported back to the developers. For reporting and tracking we are using JIRA.

You can find it on this URL [http: //actumwalmark.atlassian.net](http://actumwalmark.atlassian.net/) and you will need to use email and password to login in.

JIRA is also used for tracking of development progress and each bug should be linked (field Issue Links) to user story it’s blocking.

Environment

To ensure flow of development and stability of testing environments we work on multiple environments.

TEST – used for testing by QAs from Actum side, this environment is separated because is least stable due to deployment of untested code.

 – **URL:** <https://walmark-line-en-qatest.kentico.dev>

UAT – used for acceptance testing by QAs from Walmark side, infrastructure of this environment should be exactly the same as production

 – **URL:** [https://stage.walmark.eu](https://stage.walmark.eu/)

PROD – is accessible publicly by customers of Walmark, is tested by QAs from Walmark side

Bug reporting

Creating new bug

To create new bug, first you need to click on the “Create” button (this buttons text can vary based on your JIRA language settings) in main menu in header of JIRA. As agreed between Walmark and Actum, bug creator should be only one and in this case it’s Tsvetomir Pavlov.

Create issue window will appear where you need to fill all necessary information so developers are able to fix it.

**Project** – Walmark Line (WML), other projects can have different rules for bug creation

**Issue Type** – Bug for bug and User story for change request

**Summary** – is name of the issue, should be easily readable and quickly describe bug. Example of good summary: “500 after clicking ‘Health index’ button”. Example of bad summary: “Error after clicking button”.

**Description** – needs to contain text description of bug, steps to reproduce and expected result of retest. In this field you can also give developers more information about behavior like browser brand and version, device, URL to page, time of testing etc.

To fill description use formatting provided by JIRA application to enhance readability.

**Priority** – sets on which bug should development team focus first. Bugs with “Highest” priority will be fixed first and bugs with “Lowest” priority will be fixed last.

Priority of bugs is set to “Medium” by default.

**Attachment** – it extremely helps when developers have images showing problem to orient themselves in application and if bug is visual, it’s a necessity.

**Linked Issues** – helps to keep track of what bug is preventing what user story from being finished. Bug almost always “blocks” certain user story. Fill ID (WML-123) of user story to **Issue** field to link them.



**Assignee** – assigns this item to specific person, assign bugs to QA and change requests on PM of project.



After filling all information you can provide, click on “Create” button on bottom of window. It will close this window and redirect you to newly created bug. Where you can make changes to data you already filled in.

Retesting bug

When bug is assigned back to you and is in state “Ready for UAT” It means it can be retested on UAT environment. When you start working on this bug first change its state to “In UAT”, this means it’s currently tested and nobody else will work on it.

Remember read Comments because they can contain important information and check screenshots. Your testing can end in two states:

1. **Bug is fixed** and you can change its state to “Ready for Production” if it was tested on Stage or „To be evaluated“, if it was tested on Production and write comment e.g. „retest on Stage environment OK“.
2. **Bug is not fixed** and you should reopen this bug and assign it back to QA in Actum. Remember to comment why defect still occurs or what was expected, but wasn’t fulfilled.

Example User Story

Let’s take this user story <https://actumwalmark.atlassian.net/browse/WML-47> and try to do an example of testing procedure. First what we need to do is to carefully read through the User Story so we can make a picture of what we are going to test.

**Design**

It is always helpful to see some kind of a static design so we have a starting point of how something should look across different devices.

**Acceptance Criteria**

Probably the most important part of User story is Acceptance Criteria. There should be an exact description of how something should behave and what should the User see. In this example there is stated that Each product thumbnail contains some specific information (Image, Rating, Description etc.) and that these are all loaded from the Product Detail page. Next there is stated that customer can click anywhere on the product thumbnail and it should act as a link to that specific product detail. Note about action stickers is mentioned, but it is also part of another user story.

A very important message is that there should be only 150 characters max in the Product short description.

**Kentico Backend Administration**

First we need to Fill in the necessary information into our Product. Let’s take for example Vitamin C. In Kentico, Navigate to **Products/Product Family/Vitamins and Minerals/Vitamin C**. All the fields that can be edited are found in the **Form** tab on the right side of the page.

**Image** – We can upload the Product Image by clicking on the Upload File button in the Main Image field located in the Walmark Line section.



**Rating** – Ratings themselves are part of a separate user story and can be filled manually in the Reviews custom Module in Kentico. So far submitting the review from the frontend side is not yet implemented.

**Title** – The name of a product can be edited in the Name field located in the General section on the very top of the page. It is also required.

**Products Type** –Choosinga Product type is done by simply choosing one from the dropdown list in the General Section.

**Product Sticker** – Choosing a Sticker is done the same way as the Product type with the exception that we have a special dropdown list in the Walmark Line section.

**Short description** – Description can be filled in the textbox Short Description located in the General section. Mind that according to the notes in User Story it should not exceed 150 characters otherwise the thumbnail can behave not as intended.

Now that we’ve filled in all the neccessary fields the final thing is to assign this product to some category so we can test frontend. To do that simply navigate to any category under the Products item in the content tree. Let’s say we choose Imunity support.

Click on that category and scroll down to the **Pages in this category** section. There we can simply click on **Add page** button and then in the popup windows that opens find our desired product in the content tree hierarchy (in this example Vitamin C) and click **Select**. After that the popup window closes and we should see our product in the section as in the picture below.

When there are more products added in the category you can also rearrange them by clicking the dots icon next to the red bin and drag-and-dropping them wherever you like in the grid.

**Frontend Testing**

Open UAT environment and in the main menu navigate to Imunity support category.

There we can see the product we just added as a thumbnail with all the required items visible. By clicking anywhere in the thumbnail we should get to Product detail page. Next step is to open this page in various different browsers and check that there are no major differences (e.g. some elements overlap others in Firefox).

We should always test against the static design but also pay attention to Acceptance criteria and comments in Jira as there might be some notes that may explain some deviations. Design also provides the tablet and mobile versions of the page so we can test in on those in the same way.

