

Why Reopen Counts are high in Scrum Development?

Vinod Kumar Chauhan
Associate Quality Engineer
Quality Assurance, HPU Shimla, NOIDA, India
jmdvinodjmd@gmail.com

Abstract— At macroscopic level a user story (US) or bug is reopened due to limitations from four fronts- BA, developer, QA and Environmental issues. But at the root level only developer and QA are responsible when a US or bug is reopened since these are Developer and QA who are responsible for actual implementation of a US or bug and they can control the limitations from BA's end and the environmental issues.

Index Terms— User story reopening, bug reopening.

I. INTRODUCTION

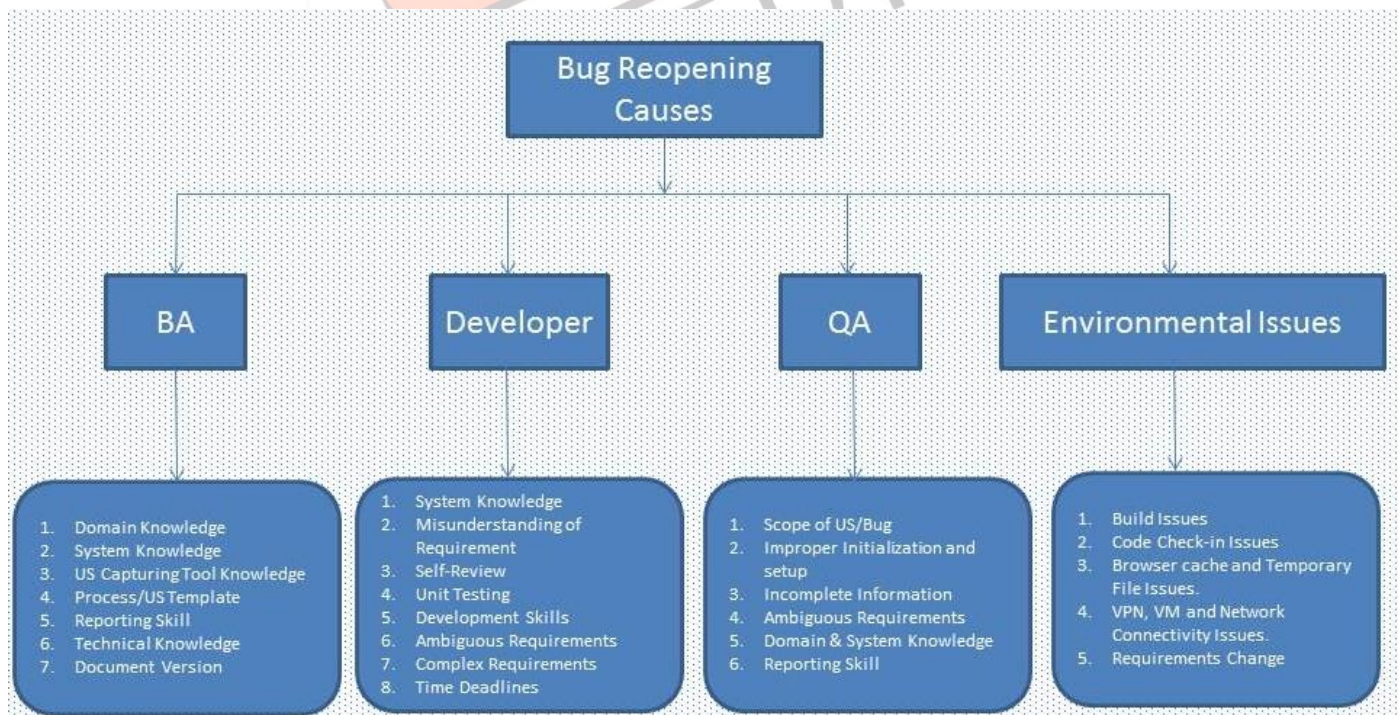
Scrum is an iterative and incremental Agile software development framework for managing software projects and product or application development [1].

User Stories (USs) are a simple way to describe the requirements in the product backlog of a Scrum project. Each user story says what the requirement is and who or what situation it should be applicable to [2]. They are generally maintained using tools like JIRA. A US or bug passes through many stages viz. Open, In Progress, In Development, Ready for Deployment(RFD), Ready for Testing(RFT), Testing and Done respectively. When a US is created, it is in New or Open state; once it is considered for implementation then it is in In Progress state; when a Developer starts work on US then it goes in In Development state; once US is implemented and committed to central repository then it is marked as RFD and whenever a new build is created the changes are present in the build; once build is deployed to the QA environment, US is marked as RFT; when QA starts testing on the US he/she marks it as Testing; now if US is pass then QA marks it Done otherwise US is reopened to fix the issues with the US. Similar is the sequence and treatment for bug also. A US or bug is reopened by the QA to rework on it due to two main reasons, either US/bug is incomplete i.e. done criteria is not fulfilled or due to confusion to QA in testing the US/bug.

The **scope** of this research paper is to analyze the causes of a US/bug remaining incomplete and causes of confusion to QA team which lead to reopening of US/bug and increase of reopen counts.

II. CAUSES OF US/BUG REOPENING

At macroscopic level a US or bug is reopened in Scrum development due to limitations from the four ends: BA, Developer, QA and Environmental issues.



Bug Reopening Causes

A. Business Analyst (BA)

BA is responsible for gathering requirements from the clients and to create USs to implement the requirements. USs are created and maintained using a requirements tracking tool like JIRA, Spira etc. So a BA is responsible for providing information for implementing a US. But sometimes the USs are not written properly by the BA. USs are either ambiguous, lack clarity, contain incomplete information, poor reporting etc. Due to these reasons developer are not able to get the clear picture of requirement which they have to implement. Now if the requirement is not clear to the developer then how he/she would implement the US/bug correctly and what can we expect from him/her. In this situation instead of asking clarification from BA if a developer makes his/her assumption and implements the US then this would confuse the QA and there is a great probability that US/bug is reopened. So here we would discuss different reasons about why USs are not written properly by the BA which ultimately may become the causes of reopening the USs.

1) Domain Knowledge

Having domain knowledge is an essential skill for a BA to work on a project. Domain means some work area like finance, health services, automobile, banking etc. In software, the domain knowledge is knowledge about the environment in which the application runs or operates [3]. Good domain knowledge helps a BA to capture the requirements correctly and to write them with clarity. If a BA lacks domain knowledge then he/she would not be able to capture the requirements properly and then this would impact the implementation of requirements in the project and might lead to reopening of US/bug.

2) System Knowledge

Here system is the system under development. BA should have clear picture about the product and the functionality to be implemented. BA should know about the different users of the system, different application flows and their finer details. If a BA lacks the system knowledge then again he/she would not be able to create the US with clarity and when US is not crystal clear then we could not expect developer to implement that US correctly.

3) US Capturing Tool Knowledge

Better the presentation of US/bug, the easier is the understanding of US/bug and lesser is the time to implement the US/bug. So a BA should possess good knowledge of tool which is used to create the US/bug e.g. A BA working with JIRA should know how to add links, how to attach documents, how to make required text bold or italic etc. so that he/she could better present the US/bug.

4) Process/US Template

The process and US template used for creating US/bug play a vital role in the implementation. If BA is not following good US template for creating USs and he/she is creating it randomly then it would again impact the clarity of US. A user story typically should follow a simple template:

Title (one line describing the story)

Narrative:

As a [role]

I want [feature]

So that [benefit]

Acceptance Criteria: (presented as Scenarios) [4].

5) Reporting Skill

You may have good knowledge of a topic but until and unless you express it appropriately, no one knows about it. The reporting of a US and bug are very important for correct implementation. All the required information and documents must be provided in proper order with a bug and US e.g. a bug should have a screenshot attached to it wherever possible with the defect area as highlighted. This helps the developer to reproduce the bug easily and helps the QA to understand the actual and expected behavior while testing. Steps to reproduce a bug should be clear and the actual and expected behavior should be given properly. The done criteria for a US/bug should be clear and better if written in point wise manner. This helps developer, QA both and reduces the probability of bug/US reopening. So a BA should have good expression power.

6) Technical Knowledge

BA should have technical knowledge to certain extent. He/she should be aware of the technologies and platforms used by the system under development. Since sometimes he/she need to create USs that deal with technical terms so again lack of technical knowledge lead to lack of clarity in US e.g. Suppose a system need to upgrade the application server (say JBoss) from lower version to latest version. Here a BA should be aware of what an application server is and such things.

7) Document Versions

Some USs have some documents with them for their implementation. Now if developer and QA are following different documents then there is probability that US/bug would be reopened. It's the responsibility of a BA that same documents are provided to both developer and QA. Similarly the information present in the US/bug must be kept up to date by the BA.

B. Developer

Developer is the person who actually implements the requirement and thus is the main culprit for reopening of a US/bug. There are a number of reasons for a developer not being able to implement the requirement, these are discussed as:

1) System Knowledge

A developer must have complete knowledge about the system under development, its design, all the flows, all the users etc. System knowledge is one of the major reasons for US/Bug reopening. Most of the times a developer don't have awareness about all the flows and their finer details so he/she applies the fix either at wrong place or not on all required places which lead to reopening of US/bug.

2) *Misunderstanding of Requirements*

Another major reason for reopening a US/bug is the misunderstanding of the requirement by the developer. It's very obvious that when a requirement is not clear to the developer then how he/she could implement it correctly. So it's better for a developer to read the US/bug repeatedly until he/she gets the clear picture e.g. suppose FAQ(Frequently Asked Questions) icon has to be added to navigation bar in online shopping website for a consumer. The developer read this and added FAQ icon for anonymous consumer which first time hit the site. So this requirement would be reopened since this has to be implemented for both anonymous and registered consumer. So here developer misunderstood the requirement and due to this he/she could not implement it properly.

3) *Self-Review*

This is another major reason for US reopening. The developers do not review their work after fixing and this is how things skip and US/bug get reopened. A good developer not only review his/her code after fixing the US/bug but also self verifies it in the QA environment before marking US/bug to RFT and this way chances to reopen the US/bug are reduced.

4) *Unit Testing*

A developer is supposed to write unit test cases (like JUnit test cases for JAVA code) to make sure that the code he/she has written is working as expected and this could reduce the probability of US/bug reopening. Developer do not write unit test cases saying that they do not have enough time and this way probability of US/bug reopening increases and on average a developer has to spent more time on US/bug. So in long run, writing unit test cases could reduce the US/bug reopen count and time to implement it.

5) *Development Skills*

If a developer lacks the required development skills then it is definitely going to impact his/her work. A developer should possess skills like comprehension, logic skills, intellect, problem solving ability and passion for coding. These skills help the developer to understand the problem, to visualize all scenarios, implement requirements correctly and keep going. This is one of the major reasons for US/bug reopening.

6) *Ambiguous Requirements*

This situation occurs when a BA lacks in properly documenting the requirement. When a US/bug is not written with clarity then it would definitely going to confuse both developer and QA and it would require extra work to achieve the actual implementation. Many US/bug suffer from ambiguity. Ambiguity means that a developer could interpret the US/bug in more than one way or that multiple persons come to different interpretations. In either case, ambiguous requirements lead to confusion, wasted effort and rework [5] e.g. 'suppose in online shopping, for shopping < \$2000 there is 20% discount and for shopping > \$2000 there is 25% discount.' This is an ambiguous statement because what if total amount is equal to \$2000.

So it's very important that whenever US/bug is ambiguous, instead of making assumptions, a developer must communicate with BA and get the requirement clarified. But if a developer makes his/her assumptions about the US/bug then it is going to confuse the QA and probable to be reopened.

7) *Complex Requirements*

Before starting the implementation of a US/bug by the developer the US/bug should be properly estimated and divided into subtasks. A US with multiple things inside that has more probability of reopening than one with single task. A US should follow the INVEST model coined by Bill Wake (Independent, Negotiable, Valuable, Estimable, Small and Testable). Each US must talk about one type of requirement like each US should be written for one type of user of the system e.g. suppose a US talk about branding of application and functional changes also then it's a complex requirement.

8) *Time Deadlines*

Generally developers have time deadlines to complete their work. This produces a pressure on the developer and he/she is not able to look into the requirements properly and in the hurry to implement the requirement, they miss some points which result into incomplete implementation of the requirements and bound to be reopened by the QA.

C. QA

A QA reopens a US or bug when either the done criteria is failing or due to confusion about US or bug. So here we are going to discuss the different reasons of why a QA get confused and reopens a correctly implemented US/bug.

1) *Scope of US/Bug*

This is one of the major reasons for reopening a complete US/bug by the QA. In such situations QA notices that the US/bug is implemented correctly but the neighboring things are disturbed. Now these could have been disturbed due to the implementation of US or it might be like that even before the implementation of US/bug. Generally a QA should stick to the done criteria of the US/bug and he/she should log a separate bug for the observations in the current US/bug and close the US/bug e.g. suppose in an application print functionality is not working at all. So there is one bug with done criteria that on clicking print button the document should get printed. Developer fixed the bug and when it came to QA, he/she found that on clicking print button the document is getting printed but some of the contents are overlapping. Now what should be done? The done criterion is passing but there is some issue with printed document. Here the QA should close the bug and log new defect for the observations.

2) *Improper Initialization and Setup*

Generally for testing US/bug, an initial setup is required or we can say US/bug has some preconditions. Now if a QA fails to setup the required preconditions for the US/bug then he/she wouldn't be able to test the US/bug and there is a great probability of US/bug being reopened. This is another major cause of US/bug reopening by the QA e.g. in online shopping when a consumer adds five products from five specific categories then he/she would get a discount of \$10. Now when a QA tests this requirement if he/she does not add products from all the five specific categories then he/she won't get any discount because the QA has failed to setup the preconditions. This way US/bug would be reopened although it was complete.

3) *Incomplete Information*

Sometimes a US/bug does not contain all the information required for implementing the requirement e.g. suppose when a US/bug came to a developer for implementation then he/she was having some confusion about the requirement so he/she have had some communication with BA over the requirement and both of them agreed over some change to the requirement. Now if both BA and developer don't update the requirement then when it would come to QA, then the exact picture wouldn't be clear to him/her and there is great probability of reopening the US/bug. So the lesson is that either developer or BA should update the requirement in such situations. Developer should attach mails (which he/she communicated with BA) to the US/bug or should add comments to it as per communication with BA.

4) *Ambiguous Requirement*

Sometimes the US/bug is ambiguous. A QA can interpret the US/bug in more than one way. In such a situation a QA checks all the possibilities and if any of the scenarios fails then he/she would reopen the US/bug. This is done by the QA to be on the safer side and this is correct also since this is QA who is responsible for the quality of product.

This situation occurs generally when a BA does not write the US with clarity. But when such a US/bug comes to a developer it's the responsibility of the developer that instead of making assumption about US/bug, he/she should have the US/bug updated after proper discussion with BA.

5) *Domain and System Knowledge*

For thorough testing a QA should have domain and system knowledge. Without this knowledge bugs could skip to the next level. Moreover a QA could reopen US/bug even when it is complete. QA should know about all the flows and all the users of the system.

6) *Reporting Skill*

Like a BA, QA should have reporting skill. This is because QA also logs defects and the defects should be reported with clarity, unambiguously. Since if defects are not reported appropriately then first developer would face difficulty in reproducing the defect and secondly he/she might not be able to implement as it was expected. So a QA while reporting defects, he/she should write the steps to reproduce clearly, done criteria of defect should be reported point wise with clarity and screenshots must be attached wherever required. QA should mention the actual system behavior and expected behavior while logging bugs otherwise it would lead to the wastage of time and bug might get reopened.

D. *Environmental Issues and Other Miscellaneous issues*

1) *Build Issues*

Builds are created either by manual process or by automated build tools like Jenkins. Sometimes builds are not created appropriately due to some misses and the build which is deployed to QA environment may not contain the required changes. So some USs/bugs which were supposed to be in build are not present in the build and bound to be reopened by the QA.

2) *Code Check-in Issues*

Sometimes code check-ins are not done properly by the developer so their changes would not be reflected in the build and all such bugs/USs would be reopened by QA e.g. suppose if developer check-in his/her code in wrong repository then his/her changes wouldn't be reflected in the build.

3) *Browser Cache and Temporary File Issues*

Sometimes US/bug is fixed properly and present in the build but still QA reopens that due to caching of old data by the browsers and or due to temporary files on QA's machine. So this problem could be solved if a QA properly clears the browser cache and delete the temporary files before starting testing on a new build.

4) *VPN, VM and Network Connectivity Issues*

VPN, VM and network connectivity issues might create hindrance to the QA testing which may lead to reopening of bug/US e.g. when using VPN if the screen resolution is not set properly then the branding of application could look different to QA.

5) *Requirements Change*

Sometimes requirements change after the requirements are implemented so instead of creating a new US, the old US is reopened to keep track of feature implementation e.g. suppose there was a US to add a navigation bar with five links. It was implemented correctly and closed. Then the requirement is changed by the client. Now a new US could be created to implement the change or old US could be reopened. Generally a new US should be created for this but it depends upon the project and to keep track of feature implementation old US is reopened.

III. CONCLUSION

US/bug gets reopened when it is either incomplete or due to some confusion to QA. Thus the major responsibility for a US/bug getting reopened is of both developer and QA. The limitations from BA's end and due to environmental issues could be controlled by developer and QA. All limitations of BA in creating & updating the US can be handled by the developer through communication with BA and having him/her to keep the US/bug complete and updated. Similarly the environmental issues could be controlled by QA and developer together. Developer should be careful in check-ins, creating builds and should assign the US/bug to QA after self-review in QA environment. QA should take precautions in clearing the browser cache and deleting temporary files.

ACKNOWLEDGMENT

[1] Deepak Babel, Senior Software Engineer-II, Samsung Research Institute (Delhi)

[2] Pankaj Kumar, Senior Software Engineer-I, Samsung Research Institute (Delhi)

REFERENCES

- [1] Wikipedia: "Scrum (software development)", [http://en.wikipedia.org/wiki/Scrum_\(software_development\)](http://en.wikipedia.org/wiki/Scrum_(software_development))
- [2] Wikipedia: "Scrum/User story", http://wiki.openbravo.com/wiki/Scrum/User_story
- [3] TooStep.com: "Domain Knowledge Vs. Analysis Skills: What is more important?", <http://toostep.com/debate/domain-knowledge-vs-analysis-skills-what-is-more-important->
- [4] "The structure of a story", <http://dannorth.net/whats-in-a-story/>
- [5] "Ambiguous software requirements lead to confusion, extra work", <http://searchsoftwarequality.techtarget.com/tip/Ambiguous-software-requirements-lead-to-confusion-extra-work>
- [6] "Writing User Stories Effectively", <http://www.janeve.me/articles/writing-user-stories-effectively>

