

# Agile QA Process

## Intro

Since Agile frameworks conquered software development - they changed a lot: our teams structure, our understanding of hierarchy and specialisation, delivery schedule, requirements approach, and many other things.

As part of this process [Quality Assurance](#) (QA) as well had to be redefined and reimagined (in part - because Agile actually does not have an explicit view or guidelines for this subject). Fortunately, some teams did it really well by embracing the freedom Agile provided and tailoring/improving QA process constantly according to their needs.

I was lucky enough to be part of couple of such "QA successful" teams. So in this document I would like to share my knowledge about Agile QA: how do we test and do overall Quality Assurance in context of Agile.

## Couple of notes before we begin

**First of all**, from my previous experience I believe that QA activities are a part of usual Agile team responsibilities (so for the most time there should be no specific QA "team") and most of QA work for new development is done within the same [Iteration](#). Which also means that QA effort should be fully included into [User Story](#) (US) estimation during [Planning](#).

**The second point** I would like to make is that QA process will differ depending on type and size of the software being created. But the overall strategy in many cases will be very similar.

**Third point** is the one about the "shift" of QA in Agile from being the activity belonging to specific "owners" to spreading into entire Agile Team and beyond. Everyone, including [Product Owner](#) (PO), Scrummaster and Team members now have to mind Quality. In practical sense this manifests in e.g. concept of [Definition of Done](#) (DoD) on different stages of the process, and also in [Acceptance Criteria](#) (AC) (which are a mix of

Requirements and QA activities).

And the **point #4** is about Developers and Testers. Though Agile in many contexts (e.g. in Scrum) dismisses specific roles by referring to Development Team members simply as ... "cross-functional Team members", in reality too often these roles are evident and "played" by different people.

Of course, when need be, Developers can test and Testers can write (at least some) code. But nevertheless we usually have these two distinctive groups in the Team. If we take a 9-member Scrum team as an example, we will probably find there one or two members who are clearly and primarily Testers (and maybe one Business Analyst, but this is completely another story).

## Overall QA Process

Here I'd like to list some points regarding overall QA approach in the Project context, while later I will concentrate more on the User Story level.

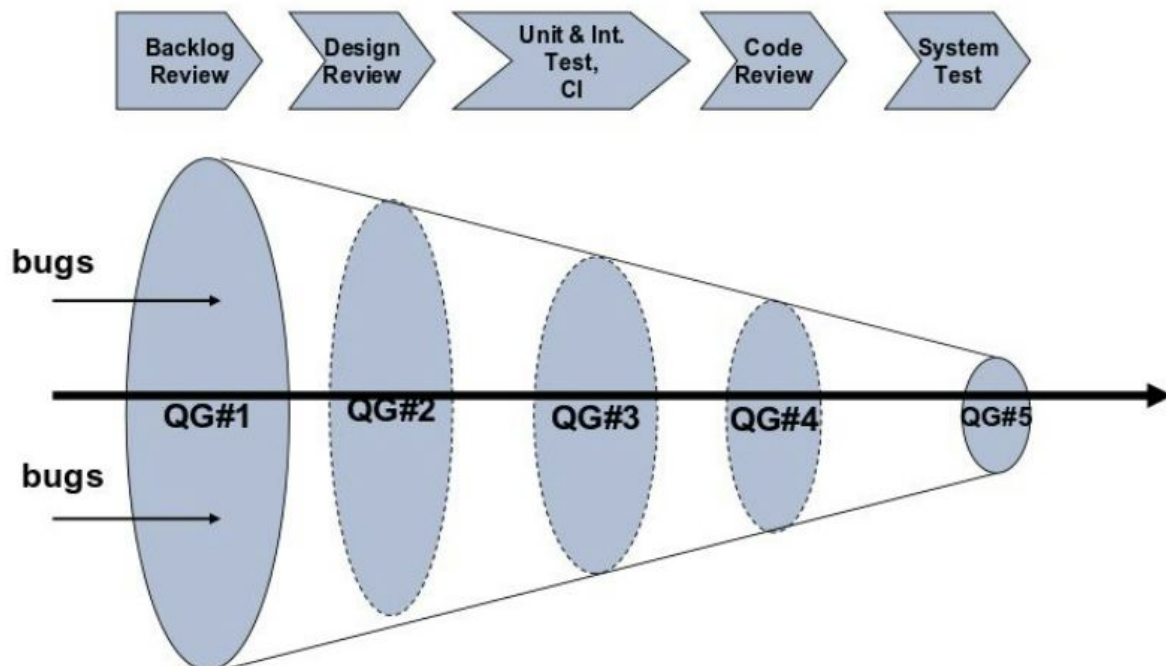
So, from my experience:

1. QA processes in Agile team have to be aligned with [Agile Manifesto](#) and [Principles](#), which also means that Product has to be "release ready" in the end of Iteration.
2. QA activities in Agile project start from very beginning and are [spread through almost all the development stages](#) and [Team roles](#).
3. All available types of testing can (and should if applicable) be used, but the most important for Agile is **automation of everything**: Unit tests, [Integration](#) tests, Functional tests, [Regression](#) tests, [E2E](#) tests, AC, etc.
4. QA metrics are widely used and extremely helpful: [Test coverage](#), [Defect density](#), number of Defects, etc.

## Assurance vs. Control

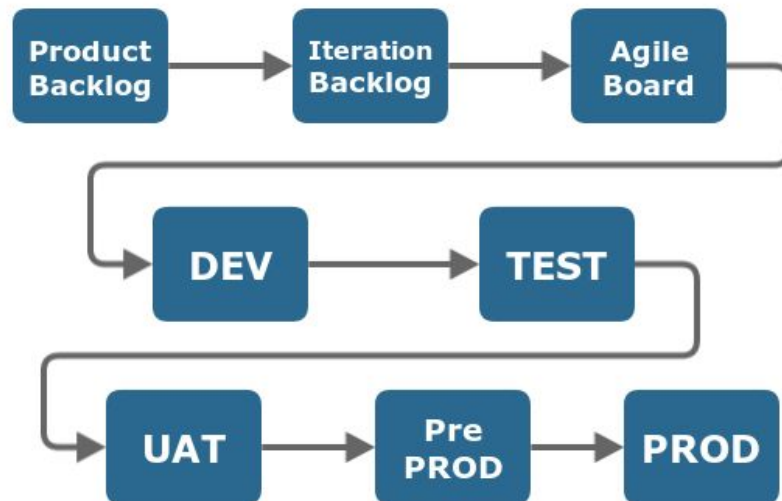
The old question of “QA vs. Quality Control (QC)” is also actual, and the answer is pretty much the same as in non-Agile context: QA is a failure **prevention** while QC is a failure **detection**.

The following image from [Scrum Alliance presentation](#) illustrates on Agile process timeline how good QA dramatically reduces number of Defects for QC phase:



# QA process for User Story

Usually in Agile project each User Story will “travel” the following route:



- US is “born” as an idea in some stakeholder’s head and later is documented by Product Owner in [Product Backlog](#);
- after US is accepted into the Iteration - it appears in [Iteration Backlog](#) (in physical world it “manifests” into one or several tasks on [Agile Board](#));
- when a Team member starts working on this particular US (usually - in their local Development environment) - this US migrates through different stages on the Board;
- after work on the Development environment is done - the code is pushed to [Continuous Integration](#) (CI) environment, and after that (when CI is “all green”) - to Testing environment;
- and after being fully tested and “green” - US can be [baselined](#) and shown to PO (however, in complex cases these activities can be postponed until after [UAT](#));
- from Testing environment US moves to UAT and [Pre-Production](#), and then finally - it is deployed to Production environment.

## User Story QA for every Role

As already mentioned earlier, QA in Agile is a process spread across every phase and affecting almost every role.

In this section I will provide a list of QA activities which different roles should be performing during the lifespan of generic US to ensure its quality. Needless to say, for every specific US these lists have to be tailored.

### Product Owner

Overall when we think about Quality Assurance from PO's side, their role in it is pretty much covered by two ("simple to say and hard to achieve") things: knowing the domain on the highest possible level and understanding stakeholders' goals.

When PO "is fluent in" domain and stakeholders - they create high value Backlog, grasp the questions, understand the implications, can quickly evaluate options from different perspectives of domain model and overall be a perfect source of adequate requirements.

Besides knowledge, PO has to be available (which theoretically is a "given" in Agile, but so often so not true in real life) and empowered to make decisions. Not having these latter two aspects are actually the major QA risk (big red flag!) in Agile project.

That's why there are not so many actual QA steps required from PO in context of specific USs development, because all major workload is done "beforehand and at the backstage".

But, nevertheless, some of the steps for PO can be anyway mentioned:

### Before US goes into Iteration Backlog

1. Prepare good quality US as a part of Product Backlog.
2. Create comprehensive Acceptance Criteria for this US. The value of good AC cannot be overrated.

### Inside Iteration (before/during US development)

1. Be available to speak with Development Team.
  - a. and investigate if their questions seem to uncover some not-analysed but valuable/potential/risky details.

## After US is developed

1. Be ready to review “done” USs even before [Iteration Review](#).
2. Carefully accept or reject USs before/during Iteration Review and give useful feedback in either case.

## Developers

### Before US goes into Iteration Backlog

1. Make sure you understand the US and AC, and for this to happen -
  - a. ask questions.
2. If US implies significant design changes - discuss them quickly with senior Developers/Tech Leads/Architect and QA Lead so you all can agree on scope of change and scope of testing which will make the estimation more accurate.

### Inside Iteration (before/during US development)

#### Local Development environment

1. Write all necessary Unit Tests.
2. Make local build “green”.
3. Perform [Code Review](#) and make all agreed upon updates.
4. Make sure all AC pass locally.
5. Update all [DevOps](#) scripts and environment settings.
6. Update all related documentation.

## Integration environment

1. Merge local code with trunk/master and commit to CI server.
2. Make sure Integration build is green, and
  - a. fix it if needed.
3. Make sure all automation tests, which are available on CI environment, pass.

## After US is developed

1. If Defects are found - roughly estimate them and get clear decision which ones will be fixed in current iteration (this will depend on their priority, US priority and remaining capacity of the Iteration).
2. If critical Defects are moved to next Iteration (in which case the Story obviously cannot be considered any "done") - create barely enough comments/documentation/notes needed to refresh the context of this US in future.
3. If all critical Defects are being fixed in current Iteration - make sure to go through all previous Developer's QA steps.

## Testers

### Before US goes into Iteration Backlog

1. Make sure you understand what should be done in scope of US.
2. Review whether US contains all AC and if PO is happy with them.
  - a. If not - help create/improve AC and get them approved by PO.

### Inside Iteration (before/during US development)

1. Discuss testing strategy with Developers and PO, plan it and do a rough estimate.
2. Create test suites that include and maybe complement AC.
3. Create all needed automated, integration, performance and UI test cases.

## After US is developed

### Integration environment

1. Make sure that all Unit Tests are done by Developers and Code Review is completed.
2. Make sure that CI build is built and all available there tests (Units/Integration/Automated/etc) are passed.

### Testing environment

1. Make sure the build is correctly deployed to the environment (including DevOps scripts, settings, properties, etc).
2. Perform "build sanity check" test suite and (if found) report Defects.
3. Run all manual and automated test suites which test this particular US and (if found) report defects.
4. Make sure all AC pass.
5. Make sure all performance tests pass.
6. If Defects (due to their size or priority) are being fixed in this iteration - repeat the test scope of "After US is developed" on new build with fixes.
7. After testing is done - include all newly created/updated/good "ad hoc" tests (the ones that make sense to keep for future) into the regular Feature or Integration test suites.
8. Make sure all minor/postponed Defects are registered in issues management tool with all necessary details.
9. If testing was successful and US is moving forward:
  - a. make sure this build was baselined;
  - b. check full DoD on this US.



## UAT and Pre-Production environments

In UAT and Pre-Production stages of the process there actually might be a specific team (or even teams, usually called Operations or just Ops) dedicated to UAT activities.

However, in the light of popular nowadays “DevOps” approach, Development Team members should be anyway very involved with all the stages which features/builds/releases have to go through until new code successfully works on Production.

# How to treat Defects

## In new User Stories

As already mentioned earlier, all the major Defects found during US development inside Iteration should be fixed inside the same Iteration before Demo.

- If major Defects are not fixed, US is considered **not** done in current Iteration (and its US points **not** counted into [Velocity](#)) and most probably is moved to the next Iteration. During the [Retrospective meeting](#) the Team discusses why this happened and how to avoid such situations in future.
- However, some found Defects can be decided not so major by PO (since the Team members might have different understanding than PO of what Defect is actually “major” in the real world) and downgraded in priority. This way US can still be released with those Defects. And those Defects have to be registered in [Issues tracking System](#), added to Product Backlog or [Technical Debt](#) list, and fixed in future (PO decides when).
- Minor Defects are either discarded or also added to Product Backlog or Technical Debt list, and fixed in future.

## Old Defects

All Defects found and not fixed in previous Iterations should be registered in Issues tracking System.

All **major** Defects found and not fixed in previous Iterations should go Into Product Backlog (or some Teams prefer having Defects Backlogs or Technical Debt lists), and for every next Iteration be considered by PO as candidates for Iteration Backlog.

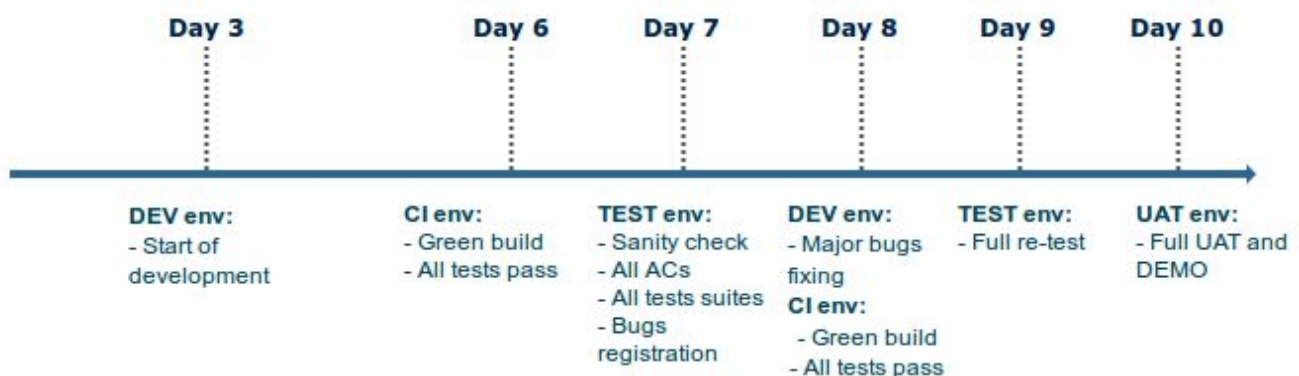
## Urgent Defects

Urgent Defects may appear unexpectedly in the middle of the Team's Iteration, and sometimes fixing such Defects is much more important for the business. In such cases:

1. For Urgent Defect at hand - decide with PO whether interrupt the Iteration for it or fix the Defect inside of Iteration (clearly marking all related tasks as "unplanned" on the Board), and if the latter is chosen - which US to mark down in priority.
2. Make sure such situations do not appear often. Discuss it extensively during next Retrospective meeting. It is question of Product quality. Carefully analyse root causes of every such "appearance" and make sure to remove these "roots". Maybe your Product needs more tests and code reviews? Better formulated AC? Whatever is missing - get it. Urgent major Defects are the symptoms of larger underlying problems, which need to be solved.

## Example

Let's review an example of US being developed during one 10-days Iteration:



- **Day -3** of the Iteration (*not on the diagram*): During [Backlog Grooming](#) meeting US was discussed in details with PO and initially estimated.

- **Day 1** of the Iteration (*not on the diagram*): During Planning meeting US was again discussed, slightly re-estimated, accepted into Iteration Backlog, broken down to specific tasks, and those tasks went to Agile Board.
- **Day 3**: Development tasks for this US were picked up. The same day QA engineer started creating/updating all necessary test cases (since they understand the US and do not need to wait for development finish to start working on tests).
- **Day 6**: Development were done. Code was moved to CI env, built, minor defects fixed, all available on CI tests were made green for this build.
- **Day 7**: Build was moved to Testing environment. All available (automated and manual) tests suites were run, 2 major and 1 minor Defects were found and discussed with PO. Major Defects had to be fixed during current Iteration.
- **Day 8**: In Development environment bugs were fixed, new build moved to CI, and after testing there - to Testing env.
- **Day 9**: Full re-test on Testing env was done. No new major Defects found. build moved to UAT env in the evening. Full automated test suite was started in the evening to run through night.
- **Day 10**: No new major Defects found on UAT. In the afternoon new US was shown to stakeholders during the Demo and accepted by PO.

## Short conclusion

Despite the fact that I needed 11-pages document to describe it, I still think that QA process in Agile is simple and straightforward.

### The key is:

1. to make User Stories the right size,
2. fix Defects inside Iteration, and
3. have really good quality Retrospectives with tangible results.