blueguava

RELEASING HIGH QUALITY DIGITAL PRODUCTS

The four pillars of professional quality assurance to ensure a successful product launch

Contents

Foreword

Ensuring that software leaves the development workshop in tip-top shape is something that all developers and software development companies can agree on. It is a vital element of every software release and will have direct impact in the adoption of the product.

In this eBook, we aim to speak to both those who are well-versed in the world of quality assurance, as well as those who need an introduction to this discipline. We hope to share with you what a day in the life of a BG tester looks like, how to create test plans and strategies, and conduct and how based on our lessons learned, we implement a four-pillar QA approach that supports the software from planning to release and beyond.

By sharing knowledge about the importance of quality assurance, we hope to raise awareness amongst professionals and business leaders that QA cannot be circumvented or handled as an afterthought to ensure that the software launch runs smoothly and remains as a reliable solution throughout its lifecycle.

You might be wondering; how can we speak of these experiences with such high confidence?

As partners to world-renowned companies, we have tested market-leading streaming services, ensuring defect-free and high-quality releases for the last ten years. The pressure to deliver was immense, given the fact that the software was available in 50 countries across three continents and serviced millions of people worldwide. Thanks to our extensive experience, wide range of testing capabilities, and our arsenal of testing devices, we succeeded every time, becoming one of their most cherished strategic partners.

How Our QA Team Evolved

Our QA team started with only one member when we started providing professional QA services. While we certainly don't recommend setting up an in-house QA team with just a single person, the Blue Guava QA team's road to success was both educational and phenomenal.

Today, that original person is our QA team lead and our team has grown by 900%. We have a team of ten enthusiastic, passionate, and vastly experienced testers that offer QA services to our clients. Let's see a quick example of how the increased team size (as well as the expertise of our team members) translates to faster deliveries:

With the introduction of support global clients (more about these in the next chapter), the implementation of regression tests accelerated quite visibly. Based on the needs of our customer's application, the team introduced multiple shifts to cover the activities necessary to ensure that the application and its components were performing as expected. The application testing had two main goals, to ensure the functional quality of the application and to ensure that the corresponding analytics provide the expected data. Previously, a single regression test took up to three days: two days for the functional part and one day for the analytics part. Today, a regression test takes only 24 hours. One team member starts the functional element while another begins the analytics part. The person working in the night shift will continue and finish the test with any leftover functional pieces.

Everyone in the team possesses core knowledge about every platform we test on – both the functional and analytics elements. This shared knowledge allows us to seamlessly rotate and work around schedules and resources to ensure there is no impact to the work as workloads start to increase

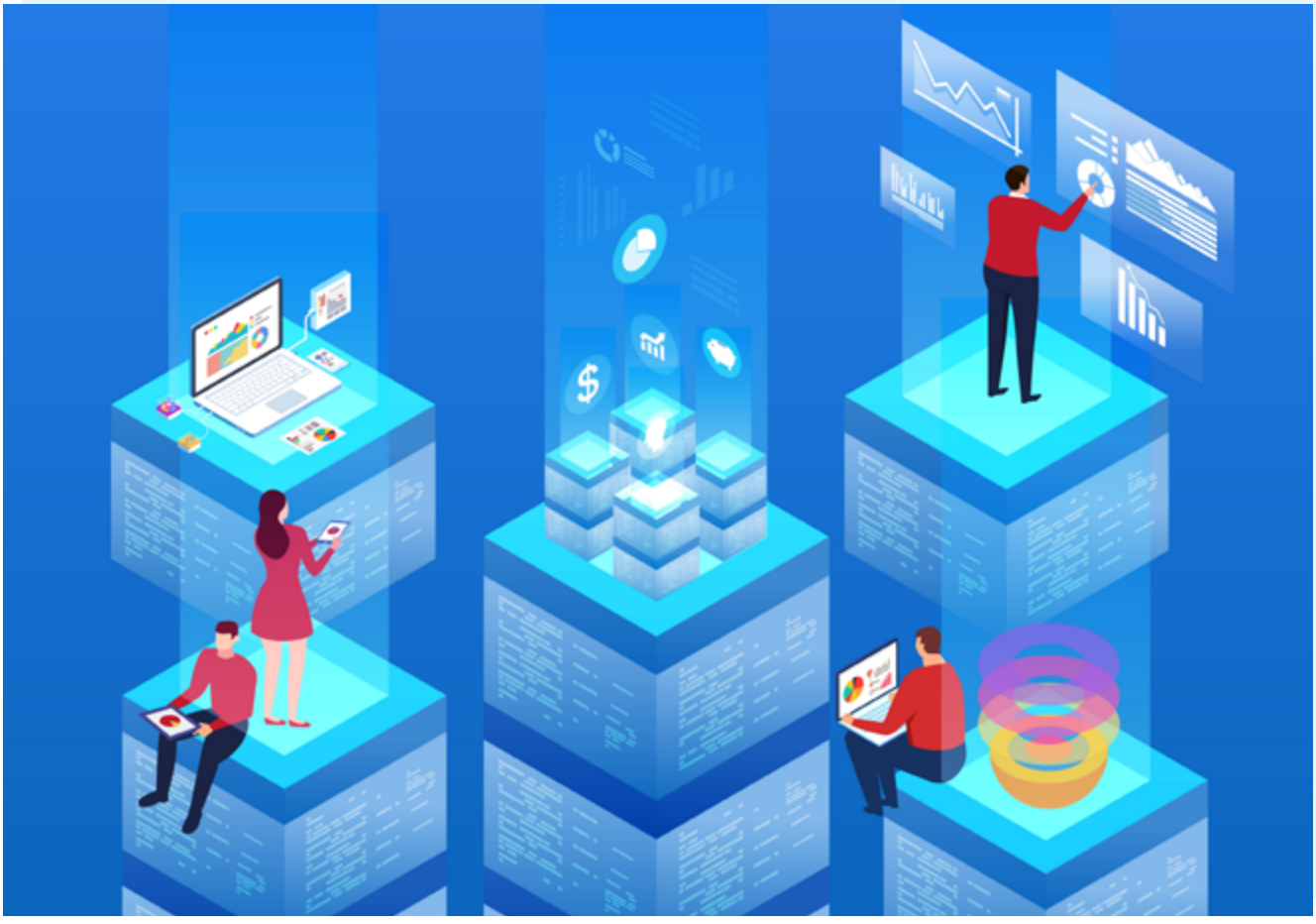
On the other hand, we make sure that beyond this "jack of all trades" approach, we provide each team member the opportunity to specialize in one particular platform, learning and mastering it in depth until they become subject matter experts. This methodology allows us to work on almost any platform commonly used around the world and address complex use cases that may be platform specific.

QA Lab

We are dedicated to ensuring the best possible product launches for our partners, and as such, we always aim to work with the highest industry standards in mind. Over the last ten years, we have tested and ensured the successful rollout of a worldwide video streaming service. This could not have been possible without the cutting-edge lab environment we designed and built for this purpose, coupled with the industrial certifications we have been awarded that recognize the excellence of our work.

Emulators can be advantageous when starting an in-house QA team, as they provide the primary purpose of replicating real devices' functionality while imitating user actions and recreating operational behaviors. However, testing on real, physical devices is the best and only way to test the app in the same environment. While the similarities are close to the real deal, it is impossible to make an application perform the same way within an emulator environment as it would perform on a real device. Emulators just cannot predict every environmental factor, feature, or user action.

> One of our most significant advantages is that we work and test on real, tangible devices – never emulators. It guarantees that digital products are tested, and the bugs are identified on those same platforms that consumers will hold in their hands while they use the app.



> Over the years, we have amassed a large collection of devices. Our lab contains devices representing almost all the platforms and versions for iOS and Android, most of the big TV brands (LG, Samsung, Apple) and we also have Chromecast, ROKU, Amazon Fire TV devices among others.

Other steps we have taken to ensure our high quality QA services derives from the certificates we have accumulated and secured. If your company has an in-house QA team or you would like to create one yourself, one of the priorities should be acquiring those necessary certifications that instill trust and convey professional services.

- **ISO/IEC 27001:2014 (2018, 2019, 2020)**
- **GDPR compliance (2019)**
- **ISTQB-Certified Testers**

Acquiring the ISO/IEC 27001 means the company possesses a sound model for security management systems, informing clients and partners that the way the company stores and manages data on their behalf is secure.

ISO27001 also affirms that the physical offices, employees, and systems comply with data laws and regulations. This means that ISO27001-certified companies adhere to best practices regarding security standards, right from the onboarding process to how clients use their software.

In addition, having an ISTQB-certified workforce means that all our QA testers possess general knowledge about every software testing base, as well as specialized knowledge on a variety of software testing areas (mobile application testing, acceptance testing, etc.). It is an affirmation that our testers are internationally recognized professionals in QA.

A Day in the Life of a QA Tester

The accelerated software development cycle puts pressure on QA as well. In software testing, just as in any other territory, challenges may occur while keeping pace with the market.

Although it might sound surprising, these are not necessarily technical problems. The top challenges emerging daily in a QA team's life would be lack of adequate communication in the group or other units, insufficient information, unrealistic schedules, and repetitive regression cycles.

The best way to visualize and understand these challenges leads through a rundown of a typical day in our own QA team. It takes a great deal of time and energy to reach an ideal state of testing, but in the end, it is truly worth it.

Before the reform that eased and streamlined work for our testers was implemented, the team faced difficulties, inconveniences, and generally suboptimal conditions for doing their job. For instance, some of our testers were swamped in tickets and were overloaded with work, while others barely received any, leaving them without tasks for days.

> Yet, the issues were more extensive than that. For example, the priority list can change multiple times during the workday, sometimes on a whim without prior notification. This led to confusion and inefficiencies in our team. This problem becomes increasingly challenging if you are working on deliverables that are time zone sensitive and you have to ensure the team members are available and deliver in the partner's work day. While our solution was provided quickly and efficiently in the form of work shifts to cover the entirety of our partner's working hours, with the haphazardly changing priorities, it became increasingly more challenging to track what was needed at the time and what its status was.

To ensure a solid framework for our testers, we successfully convinced our partner to assemble and send a daily email to our team at the end of their workday. From that point forward they compiled a list of the most important tasks for that day, providing direction for the workflow and giving us the chance to organize our efforts more efficiently. The agreement between us was that there could be deviations from this priority list only in emergency cases. In any other situation, we all stick to the plan laid down in that daily email.

Another difficulty was the necessity to carry out regression tests on every new build that we had received. It put a lot of pressure on our testers, who were sometimes bombarded with multiple new builds in a single day – this resulted in a time crunch. Our recommended solution was that regression tests. We suggested that regression tests should only be conducted on builds that already reached the "market candidate" stage. Unsurprisingly, when they agreed, and we enacted this new policy together, considerable amounts of time and resources were saved.

We wholeheartedly recommend one last agenda item to every QA team leader or manager: don't take your entire team to meetings! It's completely unnecessary. Let them test and work on their tasks while only the QA team leader attends meetings. Afterward, they can quickly and efficiently disseminate all that information to the rest of the team without interrupting their workflow.

In the table below, we have summarized the main challenges mentioned in our example and the corresponding solutions which helped our team provide an optimized outcome.

Challenge	Solution
Multiple priority changes during the workday.	A defined priority list is assembled and sent to the QA team providing direction to the day's workflow.
Organize daily work for a team whose members work in shifts and different time zones.	We introduced three shifts to fully cover the working hours of our partners. (Covered a timespand starting from 3AM until 6PM Eastern Time)
Regression tests on every build we have received.	Conduct regression tests only on Market Candidate builds to save time and resources
Every QA team member was invited to most of the meetings.	Only the QA team leader attends meetings and conveys information to the rest of the team.

It is also important to mention that testing with the agile methodology yields the best results. Agile software development goes hand-in-hand with agile testing – given the fact that testing is also an integral part of the entire SDLC.

As such, a QA team working in agile will be almost like any other developer team; the two teams will be working in tandem, defining and implementing sprints to deliver and test the software or its components. Due to this strong collaboration, testing is involved throughout the entire development cycle, instead of leaving QA to the last moments before release.

By involving testing this early, bug-hunting and elimination will speed up considerably, and we can avoid time-consuming fixes later that could potentially delay the release. As a result, fewer immediate hotfixes are needed, as the method is guaranteed to reduce the number of bugs that make it to production.

Agile testing ensures that quality becomes the driver of progress. As testing becomes an integral part of the SDLC, the concept of testability is also integrated into code and design.

You can read about the exact benefits and way of operation later, as we discuss the four phases or pillars of professional QA services.

The Importance of a 360° Product Review

Without proper quality assurance, no product succeeds. When a company is in a rush to launch its product as soon as possible – or when it performs the first load testing after launch – software failures are bound to happen. If they are released in such a state, products will not achieve their full potential, and they certainly won't meet consumer needs or requirements.

A QA team that is available 24/7 to perform a 360° review of your digital product is a surefire way to drastically improve your software delivery life cycle. It is also an excellent way to free up company resources that could be used to prepare for launch via marketing, instead of just remaining stressed and reactive, performing bug triage even in the last couple of days leading up to launch.

So, why is it essential to perform this comprehensive review of your product?

The short answer is to ensure a defect-free and reliable launch of your custom software solutions, to free up your development team, giving them more time to work on what matters, and save yourself and your company a ton of headache. Now let's unfold this answer using our strategy and the four main pillars of QA.

The Magic Recipe of Blue Guava's QA Strategy

In general terms, our methodology boils down to the triad of cross-functionality, knowledge sharing, and rotating roles. We intend to train each team member on all devices, enabling the team to be able to test products on virtually any platform who can test products on virtually any platform. This way, they can broaden their horizons while being able to perform testing for the client whenever it is needed, regardless of time zones and shifts. The diversified devices and challenges also make work more exciting.

Let's examine some of the tools that enable the Blue Guava QA team to walk and chew bubblegum at the same time:

1

Daily priorities and goals

Since we are often testing for overseas partners and consequently work in three different shifts, it's hard to run daily standups in Scrum. As we have mentioned before, our solution came in daily priority emails from the client to plan for the next 24 hours. The partner must come to an agreement about the condition that there will be no sudden, last-minute deviations from this email's contents – only in cases of absolute necessity.

2

Follow Up

The QA team leader writes a follow-up email summarizing current priorities, sifting through the importance of previous days to avoid confusion and synergize progress.

3

Reduce Meetings

We have learned that it is usually enough to have only the QA team leader attend meetings with the client. It gives the team more time to test with fewer distractions, and the team leader can convey all essential information after they returned from the meeting.

4

Tri-shift workday

We are working in three shifts (Morning – Afternoon – Night) to cover various time zones. This way, we are available for our overseas partners for the entirety of their business day – and they greatly appreciate that.

5

Real life test cases

We have realized that the best testing outcomes result from test cases 100% based on approved user stories.

6

Functional and analytics regression tests

We perform both a functional and an analytics regression test as the product launch approaches. This is our tried and proven method for ensuring bug-free and reliable rollouts with excellent performance.

7

Regression tests

Our experiences showed that it is far more advantageous and cost and time effective to perform a comprehensive regression test only on the Market Candidate build.

8

Pre-release sanity test

If a regression test was run beforehand and there were no significant changes to the product, we usually perform only a single sanity test before the market release.

9

Hotfix sanity tests

Some cases and issues arise only after launch. To speed up hotfix releases, we also perform sanity tests on those.

The Four Pillars

Our 360° product review can be grouped into four columns or pillars that serve as a foundation for proper QA, supporting our efforts throughout the SDLC and the post-launch period. These four pillars are tied to three chronological phases of a software development project: planning, development, and release, with the addition of a post-release follow-up phase. In the sections below, we will discuss the main objectives and best practices for each.

1. Planning

It is worth keeping in mind that a QA project could begin as soon as the planning for product development starts. These two can run parallel to each other, yielding a series of advantages in the long run.

However, to begin discussing the specifics of planning, first, we must establish the definition of two crucial deliverables that influence progress in this phase: The test plan and the test strategy.

Test Plan

The test plan describes the scope, objective, and approach of a particular QA project, and it usually includes additional information on features that are to be tested, testing techniques, as well as the framework of daily tasks and their specifics. It determines the process of testing, answering questions like what, when, and how to test.

Test Strategy

This framework lays down a set of thorough guidelines that explain how the precise testing is to be carried out. It includes objectives, the format for documenting testing results and tickets, test processes, and anything else that is particularly needed for that specific project. In a nutshell, it describes the general approach and techniques that should be followed during the project.

With these definitions in mind, let's see why it's beneficial to involve QA from the start!

Significant benefits of proper planning

1

A better understanding of the product

With a QA team analyzing technical requirements and user stories, it is undoubtedly beneficial for all parties involved. QA will gain vital knowledge about the product, while the development team will have a deeper understanding of their product just by teaching others about it. Connections will click, and they will come to realizations that help further improve their software.

2

Identifying and eliminating discrepancies

There are usually some slight deviations in design documentation during the early stages of planning and development. With a fresh set of expert eyes that scan and hunt for such discrepancies, the development team will save itself a lot of time and headaches. This translates to fewer delays and more significant savings at the end of development.

3

Reducing the chance of higher severity issues

As a synthesis of the previous two points, the involvement of QA and their help in discovering design deviations, technical flaws, and increasing the development team's understanding of their product ultimately means that fewer high-severity issues will emerge as a result.

4

Earlier bug detection

By involving the QA team from the beginning, bugs will be found early on, and the testing phase will also shorten as QA team members will have fewer issues to track down and report. This leads to faster delivery of the software, and ultimately a reduced amount of time until the product is ready for release.

5

Track changes throughout development

Test cases and project documentation undergo multiple shifts throughout the development and maintenance cycles. Involving testers from the very beginning helps in keeping track of changes made to test cases.

To summarize, with the QA team joining the project early on and helping developers from the start, a company can save a lot of time and money. In other words, it is more time and cost-effective to deal preemptively with as many potential, future issues as possible – problems that could arise from design discrepancies or flaws.

2. Development

Just as you have seen how vital and beneficial it can be to involve QA in planning your digital product strategy, it is even more imperative to have them participate in the development phase. Productive and creative testing can uncover many bugs, effectively shortening the development period and testing. That leads to an on-time launch with a product that is reliable and performs as intended.

> However, leaving QA out entirely, or only involving them after development is officially over, is likely to result in a catastrophic launch. Unfortunately, there are many deterring examples of such scenarios out there.

Bug-hunting tips & tricks

Finding and reporting those pesky bugs that can sour the user experience might seem easy at first but requires time to truly master. We have put together our seven-step best practice guide that's guaranteed to give you new insights on bug-hunting and improve your skills.

1 **Understand the application.**
Having a comprehensive, in-depth understanding of the whole application or module will make your job a whole lot easier. With the necessary knowledge in hand, you can maneuver better and push the app to its limits. Then even the most hidden bugs will come out.

2 **Prepare good test cases before testing.**
The best testing outcomes result from test cases that are 100% based on approved user stories. Ensure your preparation includes writing a good test case that considers all the significant risks of the application.

3

Look for oddities the software shouldn't do.

By this point, you have a great understanding of the software and its features. So, you will surely know when you encounter an anomaly that is not something the software could or should do under normal circumstances.

4

Learn from existing bug reports and try to reproduce them.

Collaborate with testers and developers. Check and discuss each other's bug reports, try to reproduce them on your own, and figure out what might be causing them. While this tip will not find you more bugs to report, it helps you learn more about testing best practices and become more efficient at the job.

5

Do some Monkey Testing.

Just imagine what would happen if you let a curious and intelligent monkey – or an experimenting user – have its way with the app. Now try to mimic that behavior. Perform user tests on the application by providing random, invalid inputs and make it your goal to crash the app. Make sure that you are monitoring how the app behaves and responds to your shenanigans.

6

Defects tend to cluster together

According to the notes about the new build, the issue that you reported has been fixed. Great! But don't forget that neighboring or related areas might have been impacted as well. While checking if that fix is correct, make sure that you check all the associated components!

7

New feature vs. existing features.

Unfortunately, when a brand-new feature is added, some existing ones could malfunction or start behaving weirdly. When checking out the new feature, make sure you also pay attention to the "old" ones and confirm that they didn't break.

Exploratory testing

Testing is not limited to a series of mundane, repetitive tasks, it requires flexible thinking and good perceptive skills and a great deal of creativity to be successful. Exploratory testing can only be performed well if it is done with creative and intuitive flair. As such, exploratory testing is a crucial part of the overall QA process.

> Exploratory testing is most useful when there are few or inadequate specifications concerning the product and its features, or there is significant time pressure on testers. It is also highly effective in the early stages of the SDLC, when requirements are still unclear, and the code regularly undergoes rapid changes. Similarly, it also helps when you do not have time to pre- define and script test cases.

Although it might not seem as informative as it should be, exploratory testing can lead to new knowledge and experience gained about the product. The information you collect during this process will be incredibly beneficial when you start preparing test scripts for regression testing in upcoming development iterations.

> Exploratory testing provides excellent insights into the software and deepens your understanding of it. You also increase your testing expertise, which will generally help you throughout the project and even beyond – in your entire career as a QA tester.

To put it in a nutshell, if you explore during testing, do it with creativity!

Automation: choose your way carefully

Automation can be a double-edged sword, and we urge all QA teams to weigh the pros and cons before deciding on automation.

Of course, the idea of automation might seem advantageous and attractive at first glance. However, it can lead to a series of unforeseen troubles that will make things more complicated instead of streamlined and straightforward.

If the QA team is fully committed to automation, they should take the time to properly investigate what it entails and whether it will yield tangible benefits. It's also important to inform the developers that your team will be using automated scripts to execute tests on certain features.

> Keep in mind that automation may not be applicable for all use cases. In certain cases, only specific processes and sub-processes can be efficiently automated. It is crucial to focus on areas where automation will bring the most value.

Below we have compiled three lists about testing automation. One deals with testing methods that cannot and should not be automated. The second one is a set of recommendations and suggestions to help set up automation if it seems viable. And, the last one is about the testing scenarios that could benefit from automation.

Tests that cannot be automated:

- **Exploratory tests**
- **UI tests**
- **UX tests**

These testing methods are subjective. A UX test is not straightforward enough for a script to properly replicate a human user's behavior. It concerns the user's intuitive, subjective experience and does not follow a simple path that scripts can manage. In other words, forcing automation onto areas where it is not indicated or necessary will only result in additional problems.

All in all, since human creativity cannot be copied yet, current automation methods will be severely handicapped compared to a human tester exploring and interacting with the product inside and out during bug hunting.

How to assess if automation is advantageous:

- Establish if automation is right for the project. Define whether the efforts in automation will yield value.
- Assess the actual cost benefits efficiencies gained from investing in automation
- Focus on test cases that give valuable insight and are related only to the code.
- Always start the task by scouting ahead alone, then ask developers if your plan on automating the testing for that area makes sense.

Scenarios that benefit from automation:

- Automation of the test would be a relatively easy task that can be put together from several, more generalized manual processes.
- The test case would be used for a long time without the need to change, update, or edit it. In this case, the related automation will have a long-life cycle as well, and we can avoid scope creep.
- The comparative cost of automation is lower than the repeated manual implementation of the same test.

If there are any test cases in a QA strategy that fit into one or more of these categories, automating them will reduce costs and save more time for the QA team.

Automation can either help streamline the testing of specific tedious, repetitive but easy processes or make things more difficult. Furthermore, automation might even mean that you are choosing the wrong tools for the task at hand, so it must be carefully considered whether it's worth doing it for that specific thing or not.

Automation may not be the best approach in all cases, sometimes it is better to perform the test manually so spend time evaluating which cases may provide the most benefit.

3. Release

The combined efforts of Dev and QA teams during the planning and development phases root out a considerable number of bugs, eliminating the issues behind most crashes, and generally make the app more convenient and pleasant to use. However, with the release date fast approaching, the software still needed to go through both the finishing touches and proper testing before it can be launched.

There are three key areas where to ensure a smooth rollout, and the delivery of a reliable product which adequately meets the consumer needs and expectations: beta testing, canary releases and field testing.

Beta testing

Beta testing is a very coveted and essential step before a release, as it can add tremendous value to the product. As one of the acceptance testing types, this is probably most commonly known as the 'main' user test or series of user tests before the software launch.

> We can differentiate between two types of beta tests – closed beta or private beta and open or public beta. In closed beta, the product is released to a select group of testers by invitation while open beta is intended for the larger audience where virtually anyone interested in the product can join the testing process.

In open beta, end-users can get a first-hand experience to evaluate customer satisfaction with the product. By letting end-users validate the software early, both developers and QA team members gain a treasure trove of valuable, hands-on feedback about the product and any bugs, crashes, or performance issues.

Open beta testing challenges:

- Shortage of resources (e.g., server capacity)
- Lack of tools used for bug tracking
- Skill level of all participants can lead to unforeseen problems and difficulties
- Collecting and sorting feedback from testers properly

We have collected some of our tried and trusted pre-beta testing preparations below to avoid these issues.

1. Involve the QA team members or at least the QA team lead in the planning meetings.
2. Thoroughly discuss the stages of development with QA.
3. Send out early builds with the finished parts as soon as they are ready, not just the complete application.
4. Set up a way for the QA team to efficiently report and discuss their findings with developers.

The length of a beta test – or a series of tests – dramatically depends on the size of the software and relative complexity. It's usually preferable to plan for at least a few cycles of beta testing before launch, and after the first cycle, the dev and QA teams can figure out roughly how many more cycles needed and how long the entire beta test phase should be.

Canary releases

You've probably heard the song "Canary In A Coalmine" by The Police. A few centuries back, coal miners took caged canaries into the mine as an early-warning system to alert them to the presence of noxious gases. If dangerous gases like carbon monoxide were present, the gases would kill the poor bird before having a fatal effect on the miners, providing a crude early warning system that it was time to immediately exit.

> Fortunately, in the case of canary releases for software, no animals are endangered or harmed in any way. However, developers are spared a catastrophic beta test or final release by first seeing if the app does indeed work as intended.

The concept is quite simple: the software is sent out to a trusted circle of non-QA testers. This sample or package of the current build is then tested, reviewed, validated, and the feedback determines whether the software is stable enough to be more thoroughly beta tested.

Field test

Each country where the software is to be released may have significantly different infrastructure for the network, data transfer, bandwidth, and more, especially if localized versions of the software will happen simultaneously with the main product launch.

> As such, we highly recommend doing field tests, too, even if they are carried out by a third-party provider like a crowd testing service. Performing this type of testing using the local infrastructure and facilities can reveal several country-specific issues or oddities that should be addressed before launch. Testing features that are only available in select countries is a highly recommended process when releasing localized versions.

For instance, when testing for one of our partners, we once encountered an odd issue that could not be reproduced from our office here in Budapest. There was a function of the app that just didn't work at all in the target country, Mexico. No matter how hard we tried, we just couldn't reproduce the same issue here. Testing had to be done in Mexico, and once we involved a third- party there, we finally tracked down and identified the problem related to the country's bandwidth infrastructure.

4. Follow Up

Once the software is released, it is a memorable and uplifting moment in any software development company's story. However, we should never forget the journey that led us to this destination. Developing, testing, and fixing the solution has been a long and arduous process, but now the users are able to have a full experience and enjoy the results of all the hard work.

> In case issues arise, now you have a standard process that will allow you to address any open items remaining after the release and ensure that the product integrity is maintained during future releases.

Users will, more than likely, run into such issues and report them. The developers will have to respond quickly with hotfixes and patches to address those issues. QA will again be involved in testing those builds to ensure that those fixes addressing what's broken do not interfere with or cause malfunctions in other software features and functions.

Let's take a look at the typical activities that occur post-release.

Post-release testing and maintenance activities

In general, post-release testing aims to find, identify, track, and ultimately help developers fix all issues and bugs that seeped through the earlier phases of QA processes. Although the software was thoroughly tested during development, and its release was validated, it might be possible that some aspects were missed, as some defects are likely to only emerge in the production environment.

For instance, while a QA team will likely test the software on a dozen or more devices, there are still hundreds of variants and iterations of those platforms where a few more unforeseen issues could still arise. It's a natural process and it will take a bit of time to deal with the leftover issues. Therefore, it is essential to include the follow-up phase in any QA strategy and developers and QA teams alike should count on this and plan accordingly.

Dealing with production defects from the QA team's side involves the same tasks and challenges that occurred during the previous phases. The difference is that now a vast collection of user data and feedback can pinpoint the source of the issues. Otherwise, those same tasks are about trying to reproduce the problems to seek a pattern within, analyzing user feedback, involving teammates to synthesize a fresh perspective, and working closely with developers whenever possible. All this is done to understand the defect's underlying nature and fix it as soon as possible.

One other significant difference compared to the previous QA process is the more considerable pressure imposed by the time factor. At this stage, we are generally talking about a release schedule for hotfixes. It means that straight after the release, as hotfixes are rolled out every week or even at a quicker pace, more time-sensitive testing is required from the QA team. It's essential to plan for this to avoid undue stress and pressure on team members.

In the end, post-release testing is an iterative process that never truly ends, especially if the software will undergo significant updates in its life cycle. With time, fewer and fewer issues are discovered, with smaller and smaller impacts. QA identifies the problems, and the developers fix them, then QA tests the fix itself.

Hotfix releases

As mentioned above, issues and bugs discovered in the immediate wake of a release need to be fixed quickly and neatly. QA can help developers during the discovery and identification of such defects and during the testing of the particular build that aims to fix the trouble.

Usually, we are talking about the discovery and elimination of production issues in these cases. Problems discovered in the production environment can make the software become unavailable or produce incorrect or unwanted behavior.

Once the hotfix is ready, QA makes sure that it does indeed fix the issue at hand while not creating problems in related parts of the app. At this point, a single sanity test on the latest build is enough to simply check if the product is rational and will behave as intended.

Crash analytics

In cases where we are dealing with post-release crashes, we follow a tried and proven procedure to make sure that the source or cause of those crashes is remedied without delay.

First, we start with a thorough analysis of the crash logs. The second step is to reproduce the crash based on the events documented in relevant crash logs.

From that point onwards, the objectives are the same as in any other QA and testing task: document the crash, assign the ticket to the developers, then wait until the fix is ready to be tested.

During testing, the same rules apply as always; has the cause of the crash been eliminated, and has the fix affected any other related areas or features?

Finally, we recommend continual monitoring of the crash logs to see if that particular crash resurfaces or whether a new one emerges after the hotfix.

Monitoring

Finally, an exciting topic of discussion – or debate – is whether the QA Team should continue managing and maintaining the validity of new builds through testing or should an in-house Network Operations Center (NOC) take over their job.

Both groups essentially do the same tasks when monitoring errors after releases or checking errors after hotfix releases related to previous issues. However, NOC has a larger mandate as it also oversees test management, network management, and network monitoring and control.

> In our experience, continued trust in the QA team that has been involved through the product's entire development from planning to release and beyond, providing support to the developers at every step of the turn, will pay off once more. This is because the QA team knows a lot more about the software and its quirks; after all, they have been there from the start and have become experts in the app's functions and operations.

Conclusion

We hope we have provided a window to understand how important proper and professional Quality Assurance strategy is and thatincluding QA standards early in the process can help you achieve a smooth, reliable, and defect-free launch for your digital products.

> All over the market, across many industries where software development occurs, we often see rushed releases or unfinished products being pushed out too early. Even though the media picks up some of these stories – especially if they are about world-renowned, market-leading companies – it seems that not enough organizations take these lessons to heart, leading to repeating the same mistakes. In turn, this results in significant losses on the business side, including customer dissatisfaction, frequent and lengthy downtimes for maintenance and fixes, loss of sales, and a tarnished reputation.

Organizations that integrate QA services early in their development process and do not leave it until the end as an afterthought are poised to realize significant gains, especially if this involvement runs parallel with the development, straight from the planning board.

Let's see a brief but powerful summary of the benefits that great QA can provide, categorized along the four pillars we've discussed:

1

Get involved early

The software design itself will be cleaner from the start, as there will be fewer discrepancies, high-severity issues are discovered sooner, and both developers and QA team members will have a better, more in-depth understanding of the software

2

Get to know the product

As the software is developed, the QA team provides a timely and thorough evaluation of all functions and features. The product is tested to its limits, and consequently, plenty of bugs, performance issues, and crashes are rooted out well before release. A professional and passionate QA team's extensive involvement can help avoid unnecessary delays by getting to a stable market candidate build sooner.

3

Gather feedback fast

Through canary releases, beta testing, and field tests, developers and QA alike will receive a treasure trove of feedback with the launch in sight. Suppose that data is then adequately gathered and documented. Given that the data is adequately gathered and documented, it can ensure that the last significant holdouts of bugs and other issues will be eliminated before launch, averting another set of delays, and guaranteeing a stable and defect-free release.

4

Follow Up

The road to perfection never truly ends. Testing, fixing, polishing, and testing is an iterative process, one that ensures that your released digital product will stay stable and reliable, continuing to meet your users' standards and expectations for years to come.

All in all, consumers greatly appreciate software that provides them with an excellent user experience that is devoid of bugs or crash-inducing issues. If they learn that your company always puts considerable emphasis on the QA process and thus manages to release highly polished applications, your reputation as a stellar software developer will probably make marketing your products easier and ensure a steady stream of sales as well.

So, let's make sure your digital products are released at the highest quality and will continue to meet ever-increasing customer standards. After all, quality is our responsibility, and you are our priority.

Contact us, and we will get your digital products on the road for a smooth, flawless, and high- quality release as soon as possible!

About Blue Guava

Our goal is to become the best long-term partner that any of our clients could wish for. With more than ten years of state-of-the-art software development, streaming, and testing solutions, we have helped market-leader partners increase their revenue and the efficiency of their IT operations while cutting costs and time. Simultaneously, the software products we developed for them streamlined and optimized the streaming experience for millions of their customers across more than 50 countries on three continents.

> At Blue Guava, we believe in exceptional customer service. Our passion is to provide our clients with nothing but the highest quality services that are guaranteed to meet their needs and help them in their quest to produce excellent software solutions.

Our content delivery, content management software solutions, and quality assurance services will help you maximize customer engagement, ultimately empowering your business's customer adoption and retention capabilities.

Through our wide range of testing capabilities as well as our arsenal of platforms to test your software products on, we will get you on track as fast as possible, ensuring the defect-free and reliable launch of your digital products.

Contact Us