

1. Your team is receiving a lot of issues. Lots of them ending up on different teams, lots of them as not a bug, could not reproduce, duplicate. But majority of issues is not taken look at as developers are busy with features. QA does not take issues as priority as development is busy anyway.

3 of the possible/probable root causes of the problem	3 possible reasonable solutions	1, ideally 2 or more non-trivial problems you may encounter when implementing the solution
There is no proper programme/tool where is possible to register issues and category issues. As a result, different workers register the same issues, because they can't see others register issue.	Implement tool which will help in managing and tracking issues. Also, it should have some functionality for categories issues. This will be great help for people who register issue, because they can check (automatically) if there exist already similar issue.	The problem may be the cost of tool. Also, there might be problem that staff would not use the tool properly, and they still register duplicate issues.
Technical Support is not working properly and is not trying to solve problem by themselves. And register all the issues which even are not bug. (they have no proper knowledge)	To train staff and explain who to solve the issues that can be solved by themselves. Also, this will help staff to better understand software functionality and to distinguish what is bug and what is not.	
Lack of staff.	Add team members.	

2. Your acceptance tests run 24 hours in CI. You cannot use CI for verification of emergency fixes as it takes too long to get relevant results from it.

3 of the possible/probable root causes of the problem	3 possible reasonable solutions	1, ideally 2 or more non-trivial problems you may encounter when implementing the solution
Acceptance test is executing during week-days.	If Acceptance tests should be planned not to execute working hours.	
Product was not tested properly before Acceptance test. All High or Medium defects should be find and correct before Acceptance test.	Before acceptance test product should be thoroughly tested. Which means that important defects should be find and corrected before Acceptance test.	

3. Other teams (the same teams over and over again) often break your tests in CI in master and future-release branches. Most of those test failures catch real problems. You spend a lot of time on failing test investigation and teams must often solve problems not-long before the release.

3 of the possible/probable root causes of the problem	3 possible reasonable solutions	1, ideally 2 or more non-trivial problems you may encounter when implementing the solution
Team is oriented on deadlines and merge buggy code in the master branch.		
Team does not test code before merging.	Pre-flight builds	

4. QA Engineers do not know what exactly to verify with bug fixes, they sometimes spend unnecessary too much time on the verification (they are retesting what automated tests already tested and/or figuring out for a long time what to test) and sometimes also do not catch some bug which they could and should caught and the bug gets it to production.

3 of the possible/probable root causes of the problem	3 possible reasonable solutions	1, ideally 2 or more non-trivial problems you may encounter when implementing the solution
Unqualified team members. team consist of junior QA Engineers and they have no experience.	Training of QA Engineers. Hire the qualified and experience staff.	Resource problem.
There is no leader who controls QA Engineers works and deadlines.	Someone should be responsible for working of QA engineers and should control their work.	
Communication problem between team members.	Teambuilding activates.	

5. You are a QA lead of team of 5 QA Engineers. All 5 of them are much more interested in automation than work on customer reported issues and there are many customer complaints that their problems take too long to get some attention.

3 of the possible/probable root causes of the problem	3 possible reasonable solutions	1, ideally 2 or more non-trivial problems you may encounter when implementing the solution
as all team member are more interested in automation maybe it is problem why customer reported issues takes long time to solve.		We may have to hire additional testers, or fire some of the actual ones.

There are no strict deadlines for solving issues.	customer reported issues should be assigned between 5 QA Engineers and there should be strict deadlines.	This may cause demotivation in team members.
Tasks is not divided into team members.		

6. One developer in your team likes to give very detailed code reviews to others with lots of comments and often insists on the comments being fixed even when other developers whose code is being reviewed, do not think those comments need to be fixed.

3 of the possible/probable root causes of the problem	3 possible reasonable solutions	1, ideally 2 or more non-trivial problems you may encounter when implementing the solution
Coding standard is not defined.	Define coding standard.	One side will always be disappointed.
Relationship between team members and this developer.	Choose teammember carefully.	It will be really hard to complete team.
No lead developer to solve the dispute.	Assign a lead developer.	

7. You are working in a big company with hundreds of developers. Mainline branch is very often broken, often needs to be locked to fix failing the tests and that blocks integration of new changes.

3 of the possible/probable root causes of the problem	3 possible reasonable solutions	1, ideally 2 or more non-trivial problems you may encounter when implementing the solution
Developers merged the buggy code in master branch	Pre-flight build. Before merge branch to master branch, first it will be tested and if any test will fail, code will not be merged to master branch.	
Code is not tested before it merges to master branch.		
Merge request feature is not being used.	Use merge request feature.	Conservatism in the company, not wanting to make a change to solve this issue.
	Use trunk branch.	

8. You are a product manager in a team, where many of your developers claim there should be more sustainability work, but you don't agree and think new functionality features will give bigger business value than sustainability work.

3 of the possible/probable root causes of the problem	3 possible reasonable solutions	1, ideally 2 or more non-trivial problems you may encounter when implementing the solution
---	---------------------------------	--

There is quality degradation and developers want to make sustainability work to obtain quality of software.	Consult with another competent person.	Developers might get angry if you decide to stick to the strategy of new features.
There are requirements for new features from customers.		Whatever you decide, you still have the responsibilities. You can't blame the developers for giving bad advice.
There are no sufficient resources for both sustainability work and to add new functionality features.		

9. You are at the top development leadership in a big company and you often hear responsibility among development teams in your company is quite poorly defined and that something must be done about that.

3 of the possible/probable root causes of the problem	3 possible reasonable solutions	1, ideally 2 or more non-trivial problems you may encounter when implementing the solution
There is no position and roles defined in company. Responsibilities and functions is not defined even in working contract.	There should be defined responsibilities and roles for each of the employees. It should be documented.	Someone may be assigned unfitting position/role.
Team members don't want to take responsibilities and be only one responsible person if something is going to bad.	Shared responsibilities. Only one persons should not be responsible for something.	
There is no tool/programme to manage responsibilities to each task/issues/....	Implement program/tool which will help to manage responsibilities and define responsibilities and track.	

10. You are at the top development leadership in a big company which develops new and new versions of the product and you make new release of the product every week. Your product is business critical for your customers and there are often injections with the new releases, which breaks basic functionality for some of your customers. The code coverage is around 40.

3 of the possible/probable root causes of the problem	3 possible reasonable solutions	1, ideally 2 or more non-trivial problems you may encounter when implementing the solution
new release of the product is not tested properly.	Integration testing.	Not being able to afford to hire additional testers.
Lack of testers.	Add testers in the team.	
Lack of time to test the changes.	Automation testing.	

11. You are a QA lead and your employees often tell you it's hard to cooperate with Technical Support to them as the Technical Support has often poor product knowledge. Also, the percentage of issues ending up closed as invalid suggests it might be true.

3 of the possible/probable root causes of the problem	3 possible reasonable solutions	1, ideally 2 or more non-trivial problems you may encounter when implementing the solution
Technical support team members have lack of motivation.	Motivate staff.	May the same problem emerge again.
May be there is new staff and they have no experience and don't know how the product is working.	Training Technical Support team member.	As a QA lead you might be not able to influence the Technical Support team.
Lack of communication and misunderstanding between QA team and Technical Support team.	Analysis problems and issues once together and make it clear what should be reported as an issue.	

12. You are a DEV/QA in a company and you realize many of your colleagues – and you as well - solve the same re-occurring problems over and over again from scratch and that it takes quite a lot of unnecessary time.

3 of the possible/probable root causes of the problem	3 possible reasonable solutions	1, ideally 2 or more non-trivial problems you may encounter when implementing the solution
May be the problem is in architecture, which causes same problem again and again.	Analyze problems and find long-term solution.	To find and implement long term solution will be time consuming and maybe there will be delayed also for new tasks
Lack of the time. As solving problem needs time, there is find only temporary solution. As a result, problem is emerged again and again.	To allocate team whose main task will be to find long term solution and solve these problems.	Lack of staff. As the people needed for solving these problems may be work on other tasks.

13. You are quite a new developer in a team assigned to do a Code Review for your colleague. He just added you to the code review and did not talk to you about that. You are not sure how to do the code review.

3 of the possible/probable root causes of the problem	3 possible reasonable solutions	1, ideally 2 or more non-trivial problems you may encounter when implementing the solution
Communication problem.		

No one in team has time to explain and training new developer.	There should be someone in the team who will be responsible for new developers.	Not being able to afford to put the team in the same room.
The team is not in the same room.	Training new developers and explain standards.	Not being able to afford to assign a person to help the new developers.
	Put the team in the same room if possible	