Task 3 - problem solving

1. Problem

Root causes

1. QA disinterest in actual quality of the product
2. Developers prioritisations - fixing bugs vs new features
3. QA not taking issues seriously - every issue fix should be tested (regressions)

Possible solutions

1. Involvement of QA - they are not doing their jobs :)
    - QA disinterest and laziness
    - Overall develepment process issues
2. Refinement of the issues creation process - avoid useless issues
    - Issue tracking system may not support such tooling
    - QA may not be interested in searching for duplicates or creating reproducer/steps to reproduce
3. Redefine the QA priorities - QA participation in more general scope of development
    - QA disinterest as they are not interested by far
    - Developers disinterest in participation

2. Problem

Root causes

1. CI is running all sort of tests including the ones that are not necessary for a change done in the fix / release
2. The acceptance tests may consist of unrelevant time-wasting tests that are not part of the performance requirements or specification
3. Running whole test suite for an emergency fix may not be possible due to the time aspect

Possible solutions

1. Split the acceptance test suite to the smaller clusters of tests
    - Long tests may be unpossible to split
    - Acceptance test suite may break due to the separation

2. Reduce the time by removing long tasks that are not as important / allow to configure testing specifications
    - Problem stating what exactly should be configurable
    - Easy and transparent configuration may require additional effort
3. Emergency fixes should have a separate CI to be always available in such cases
    - No enough resources for another integration framework
    - May be a resource wasting depending of how often emergency fixes happen

3. Problem

Root causes

1. Other teams do not care about test passing
2. Other teams are not doing code reviews
3. Other teams are not finding their own errors with enough time left before release

Possible solutions

1. Pressure other teams not to commit unstable changes
    - Refusal of the participation from the other teams
    - No tooling or required knowledge in other teams how to avoid such situations
2. Mandatory code review before a feature gets passed to release branches
    - This might result in work priority change
    - The development will take more time
3. Enhance communications between teams to find problems asap
    - Refusal of participations from both sides
    - Time and priorities management prior to the release

4. Problem

Root causes

1. Unskilled engineers
2. Bad working specification
3. Lack of right team management

Possible solutions

1. Organize testing oriented courses to get some experience
    - May be time-consuming and expensive

- A problem with their attitude
2. Create a better specification
   - Might result into a misunderstanding
   - No eligible time and / or people for specification creation
3. Consider hiring more experienced people
   - No people for such position
   - Current testers may take it offesively

5. Problem

Root causes

1. Prioritization of work reponsibilities
2. Enforcement of main mandatory responsibilities for each team member
3. Enhance overall awareness about client's needs for team memebers

Possible solutions

1. Use issue-tracking system that allows for issues to set the priority
   - May still be setting invalid priorities
   - Getting used to a new tracking system takes some time
2. Explain team members the company mission and what their work should prioritize
   - May not consider customer issues as priority
   - Resistence to drop automatization focus as it make their work easier
3. Do a meetup regularly to talk about recent issues possibly including customer
   - Refusal of participation
   - Refusal of actively following and prioritize discussed issues

6. Problem

Root causes

1. Developer is over confident and stubborn
2. Developer is not open to discussion
3. Missing third party view to the problem of whether comments should be fixed

Possible solutions

1. Teach the developer to not pressure other team members
   - He could get insecure/angry after receiving a lot of critic

    - May stop making code reviews
2. Teach the developer soft skills - communication as important part of everyday work
    - May refuse to participate as he thinks he is always right
    - Even more communication may not change his mind on the specific comment problems
3. Require code review from more than one person / review the reviews
    - Duplicates the same work
    - May refuse the other reviewers comments as well

7. Problem

Root causes

1. Small amount of branch levels - separate branches for fixes
2. No testing done before merging changes to the main branch - use CI
3. Review each fix before the actual merge to the mainline production branch

Possible solutions

1. Introduce more levels of development branches
    - Could result in breaking up the current flow
    - Complications in the development process
2. Introduce CI for fix testing
    - Development can be slower
    - More responsibilities for the developers
3. Specify checklist of requirements of each fix to follow before merge to the mainline branch (CI pass, reviews...)
    - Problem to specify what exactly should the requirements be
    - Development may slow down, as there will be more tasks to do than only push fix

8. Problem

Root causes

1. Lack of understanding of the product (whether developers of manager side)
2. Missing agreement between both sides
3. Lack of the third party view for the problem

Possible solutions

1. Specify the arguments for both sides and come up with agreement through the discussion

    - No enough valid arguments on either of sides
    - Refusal to eventualy come to compromise
2. Include third party independent opinion to the situation
    - It may take long time for the reviewer to understand product priorities
    - Sides may still refuse to follow advices and arguments
3. Discuss the problem with actual customers that use the product
    - Refusal of participation from the customer side
    - Customers may be unsure of what exactly their priorities are

9. Problem

Root causes

1. Not sufficiently defined responsibilities for individual teams in the company
2. Lack of management and knowledge of the organization structure in top level management
3. Lack of proper way for employees to notify management about such problems

Possible solutions

1. Define in proper and well documented way responsibilities for team hierarchies
    - May be problem to define where are clusters of teams for individual responsibilities
    - May be necessary to split / group some of the teams
2. Allow employees to contact high management (directly / undirectly through thier managers) above their concerns
    - Depending on the selected medium - the ammount of requests can be large
    - Prioritazion of concers
3. Prioritize for top management more interaction and learning in the end employees work
    - Employees may be resitant to allow management study their work duties
    - May require higher anticipation and study from the management

10. Problem

Root causes

1. Too small code coverage
2. Too fast development cycle not allowing proper testing
3. Too many new features pushed with every release

Possible solutions

1. Writing more tests - better code coverage
    - Resistance from the actual developers as they need to focus on development
    - May require hiring new people
2. Prolong development cycle to allow testing periods
    - More time spent tesing may require less features delivered with each release
    - May result into customer disatisfaction with the product
3. Testing in the customer environment
    - Customers may be resistant to share their environment specifications
    - Could not be possible to reproduce issues even in the cutomers environment

11. Problem

Root causes

1. Lack of product knowledge in Technical Support
2. Uninterest in product from Technical Support
3. Technical support ability to close issues (invalid closes)

Possible solutions

1. Enhance the Technical support product knowledge through training courses
    - May be problem to find people to teach
    - Time aspect as this can be a long process
2. Enhance Technical Support communications with other teams (unsure whether to close issue)
    - Refusal of contribution from other teams
    - May slow down development (maintainance) process
3. Temporarily prohibit Technical Support teams from closing issues
    - May slow down the process
    - Requirement of other teams to check Technical support decisions

12. Problem

Root causes

1. No reliable dupliation detection
2. No tracking in place of issues / problems
3. Lack of knowledge exchanging capabilities

Possible solutions

1. Start using issue tracking system
    - May be hard to move from current development process state
    - May take time to really take advantage of it
2. Possibly implement own duplication detection for the problems
    - Still there is need to track the problems in one place
    - Lack of knowledge to perform such task which would perform efficiently
3. Enhance communication between DEV and QA, use the same tracking system
    - Cooperation may be difficult as cycles in teams may differ
    - May take more time to decide whether problem really is a duplicate

13. Problem

Root causes

1. He thought that I was familiar with code reviews
2. He assigned the code review by a mistake
3. I was supposed to pass the code review course, but the management forgot

Possible solutions

1. Talk to the team leader, see if this is the regular drill and if he can tell how to do code reviews
    - He tells me to look up for some reviewing conventions on the internet
    - He tells me to look at some other code review reports and follow the patterns
2. Talk to the colleague
    - He acts surprised about my low knowledge and tells me to look it up on the internet
    - He sends me to talk to team leader
3. Talk to the team leader
    - They indeed forgot to prepare the course, I have to look it up on my own
    - There was no course, but team leader said the rules are written in the company documentation pages