**Module 8 Test management**

Task #1

Create a risk matrix for testing on a project (see the given project description below)

Insurance company from USA builds a risk-assessment system for analytics team. Epam helps Team composition: 1 PM (B3 onsite), 1 BA (A2 onsite), 1 Key Dev (D3), 5 Devs (D1), 1 QA (L2), 1DQE (L1). Estimated project deadline is June 1, 2023 and is related to org changes in the customer Analytics team.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Impact | | |
|  |  | High | Medium | Low |
| Probability | High |  |  |  |
| Medium |  |  |  |
| Low |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Risk | Probability | Impact | Mitigation |
| Business domain is new for team | High | High | Set of KT sessions, maintain knowledge base |
| New project setup from scratch working from different locatuins (USA and other) | High | High | Common process and visibilityn , Scrum process, creating chats |
| Integration problems with modules after new version upgrade. | High | High | Get the information about integration problems for similar project to avoid them in the future. Get the Salesforce support. |
| Testing schedule is tight. If the start of the testing is delayed since to many software development factors | High | High | Testing is started on early stage from requirement analyzing and testing. Time buffer is added to the schedule for contingencies. discuss various options for the deadline date (not only 1 June) |
| not enough hardware/software/human resources or they deployed with delay | Medium | High | Build the resource plan. Holidays and vacation have been estimated and built into the schedule |
| Defects are found at a late stage of the testing cycle. Such defects are most likely be due to unclear specifications and are time consuming to resolve with extra budget to software development | Medium | High | Add defect management plan to ensure prompt communicationand fixing of issues |
| Scope changes | Medium | Medium/High | Every scope changes it is additional work, time and budget. For every serious changes BA and PM gather all the information and start the re-planning process. It can be scheduler changes, additional resources or overworks (depend on the situation and customer needs) |
| Human factor (for ex -1 BA, 1 QA, 1DQE) | Medium | Low/Medium | Vacation or sick leaves can affect the testing progress if there is only one person is a knowledge keeper for any area. Therefore, build the team and split the work between teammates accordingly. |
| Non-availability and accessibility of test environment | Low | High | Without access to configured test environment, QA won’t be able to complete their work. For mitigate this risk separate Dev ops engineer is assigned to the teams to resolve such type of problems |
| Testing delay to unresolved defect (for ex.-bad source data) | Low | High | During testing, there is a chance that some of the defects may be identified and may become an issue that blocks testing of another functionality. To prevent such situation, the blocker defects should be immediately highlighted in reports, metrics, verbally or by the mail and fixed as a top priority. For example , bad testing data - in such case customer should provide correct data. |
| Human factor (for ex- low level DQE ) | Low | Low/Medium | It needs mentor help. Additional resources can be needed to complete tasks on time. |

Task #2

You have issue tracker log / export file. “bond\_issue\_log.zip”

Please create metrics answering these questions:

1. What is the least reliable component of the system?
2. Is the situation improving over timeline?
3. What weeks were the most dynamic in testing/development?
4. What weeks were the most silent?
5. Suggest a threshold for bug quantity per week (take into consideration their severity)

**1 What is the least reliable component of the system?**

the most bugs found in the model A – 459 (24 % from total)

But the most count of REOPENED bugs in the model C – 199 (25 % from total reopened bugs)

Also, Module C has the most number reopened bugs ( reopened 1 time – 105 bugs, 2 times – 94 bugs)

if we look deeply to the dynamics in the context of bug modules during sprints – we can see that the least reliable component of the system is Module C. It has most of all critical and high bugs. Also the most number of reopened bugs.

**2 Is the situation improving over timeline?**

He we can see dynamic if bug growth (total +fixed). After sprint 23 the situation was improving , but than again after sprint 31 the situation started to worsen, the number of bugs began to grow.

Quality Debt: we can see the number of critical bugs increase more than others

Thus, we can conclude that after 31 sprints, the number of bugs not only increased, but the quality of bugs worsened, that is, the share of critical bugs increased.

1. **What weeks were the most dynamic in testing/development?**

I our data set of bugs we can see next statuses of bugs:

- won't fix

- postponed

- open

- in dev

- fixed

- in qa

- verified

We take into account that each bug goes through its cycle in one sprint.

create a diagram of the number of bugs, not taking into account won't fix and postponed, Because they not included in the capacity of the team.

So the most dynamic in testing/development was week 39. The most count of bugs was 63 except for won't fix and postponed bugs.

1. **What weeks were the most silent?**

So the most silent in testing/development was week 31. The lowest count of bugs was 13 except for won't fix and postponed bugs.

1. **Suggest a threshold for bug quantity per week (take into consideration their severity)**

I recommend to use SQI(Software Quality Index) . For example:

Critical – 250 points

High – 100 points

Normal – 50 points

Low – 10 points

Tiny – 5 points

So. If we don’t take into account severity, Than quantity per week is 66

If we take into account severity, in our case 4932 point its ok for sprint or 4932/83(avg points) = 59.4 or 59 bugs per week.

its count of bugs per week , based on severity column

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| severity | critical | high | normal | low | tiny | total | count of sprints | avg for 1 sprint |
| count of bugs | 191 | 613 | 604 | 250 | 257 | 1915 | 29 | 66.03448276 |
| points/ weight | 250 | 100 | 50 | 10 | 5 | 415 |  |  |
| total weight | 47750 | 61300 | 30200 | 2500 | 1285 | 143035 | 29 | 4932.241379 |