++ASSIGNMENT 4

4

**DMAIC PLAN / STATUS**

CSE 6329 -- SOFTWARE MEASUREMENT AND QUALITY ENGINEERING

Professor Dennis J. Frailey

**Fall, 2017**

|  |  |
| --- | --- |
| NAME(s) | ID Number(s) |
| **Ridham Kothari** | **1001446086** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Grading Comments (student – do not write inside this box)** | | | |
| **<total goes here>** | 1. **CTQs (Critical to Quality) (10 points)** | | |
| **/ 5** |  | |
| **/ 5** |  | |
| **<total goes here>** | 1. **Process Flow and Swim Lane Diagram (20 points)** | | |
| **/ 8** | **Identify All Organizations** |  |
| **/ 8** | **Identify All Process Steps** |
| **/ 4** | **Other Info** |
| **<total goes here>** | 1. **Root Cause Analysis (25 points)** | | |
| **/ 10** | **Technique 1** |  |
| **/ 10** | **Technique 2** |
| **/ 5** | **Root Cause(s)** |
| **<total goes here>** | 1. **Causal Model (20 points)** | | |
| **/ 15** | **Identify All Possible Causes** |  |
| **/ 5** | **Relationships Make Sense** |
| **<total goes here>** | 1. **Most Important Root Causes / Flow Diagrams (15 pts)** | | |
| **/ 5** | **Cause 1** |  |
| **/ 5** | **Cause 2** |
| **/ 5** | **Cause 3** |
| **<total goes here>** | 1. **Recommendations (10 points)** | | |
| **/ 6** | **Good Recommendations** |  |
| **/ 4** | **Legibility, Correct English, etc.** |
| **<total>** | **Grand Total** | | |

**DMAIC Plan / Status**

This document consists of our DMAIC plan and the results obtained so far, so as to provide a status report on execution of that plan.

**DEFINE**

**Charter (from management):**

**Business Problem:** The customers are complaining that there are increasing numbers of failures in our newer products and that correction of software failures is too slow. GAMMA, one of our most important customers, is also one of the ones complaining the most about this problem, although we are losing other customers because of this. We must correct this in order to satisfy our customers, especially GAMMA.

**Goal:** Determine the causes of the slow response and higher failure rate and correct them. Reduce the response time by at least 50% and reduce failure rate to what it has traditionally been in the past.

**Scope:** The entire business process of the company may be affected by this. No part of the company is off limits.

**Timeline:** We must resolve this by the end of the year.

**Resources:** We have obtained the assistance of several UTA students who have taken a course that covered the appropriate methods. We want them to develop a more complete DMAIC plan.

**Definition (from team assigned to solve the problem)**

**The problem:** Slow response to software failures and higher failure rates.

**The customers:** Several, notably GAMMA Corporation.

**Voice of the customer:** See memo from GAMMA. Customer quality requirement is software that runs properly, with minimal failures, and rapid response to correcting the software when it does fail.

**CTQs:** Measurable attributes that are critical to quality for this customer.

| CTQ #1 | How Measured | Why it is Critical |
| --- | --- | --- |
| Slow response to software failures. | The time taken to respond of failures and fixing it. | Delay in response affects the customers business operations. |

| CTQ # 2 | How Measured | Why it is Critical |
| --- | --- | --- |
| Increase in number of failures in new software product. | The number of time software has failed after the delivery of the product. | Number of failures affects the operation efficiency. |

**Target process(es) to be improved:** All processes in the company. Note: we will begin by analyzing the customer support process but any part of our organization’s process may be subject to improvement based on findings from our analysis. Specific processes needing improvement are defined as a result of the analysis.

**Project Targets:** Reduce response time by at least 50%; reduce failure rate.

**MEASURE**

**Process Flow:**

The process flow in IPC organization of the software maintenance process (customer response process) contains detailed steps as follows:

**ROLE OF EACH ORGANIZATION/PEOPLE:**

**GAMMA Corporation:**

* **Girisha Gotti**

**Job Title:** Information Processing Manager

**Job Description:** Responsible for providing computing and networking services to all GAMMA employees.

**IPC Organization:**

* **James Johnson**

**Job Title:** Customer Representative

**Job Description:** IPC’s representative to some of its major customers, including GAMMA, Delta, and Zeta Corporation.

* **George Wilson**

**Job Title:** Maintenance Clerk

**Job Description:** Review incoming problem reports, prioritize them, combine similar reports that probably require the same fix, and ship updates back to the customer representative.

* **Melinda Shah**

**Job Title:** Maintenance Manager

**Job Description:** Resolves the priority disputes, maintain necessary programing resources, handles the work of configuration control board(CCB).

* **Sharleen Jefferson**

**Job Title:** Maintenance Programming Manger

**Job Description:** Supervises the programmers and assigns them to work on specific problem report repairs. Also sits in CCB.

* **Narayan Bhat**

**Job Title:** Maintenance Programmer

**Job Description:** Analyze problem reports, identify the source of the problem, and make the require repairs.

* **Wendy Stottlemeyer**

**Job Title:** Maintenance Test Team Leader

**Job Description:** To perform system level tests, regression test and to prepare final update packages.

* **Shivani Patel**

**Job Title:** Quality Assurance Manager

**Job Description:** Supervises QA staff and serves on both software development CCB and software maintenance CCB.

* **Rachel Wallace**

**Job Title:** Software Development Manager

**Job Description:** Supervises all software development for new products.

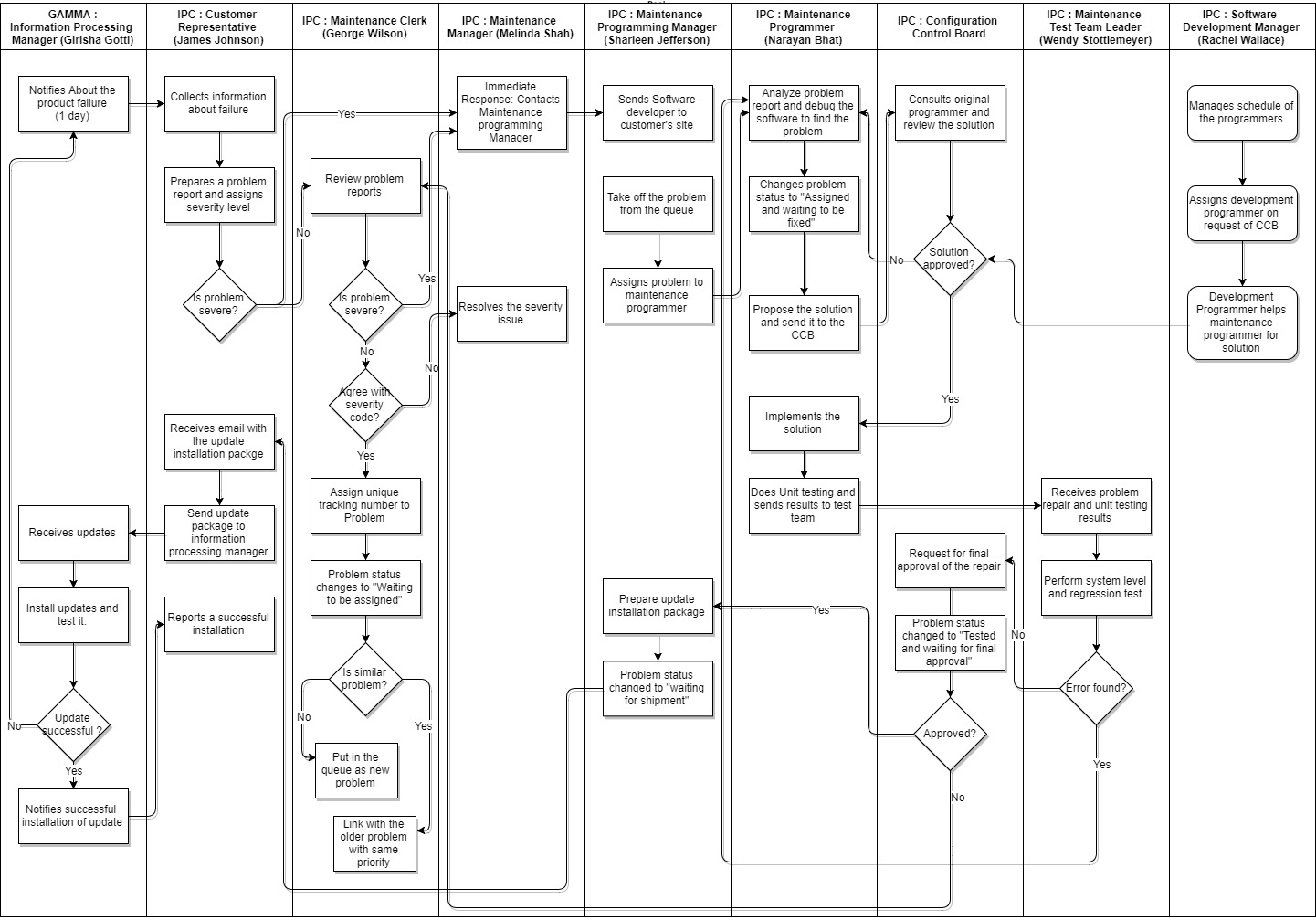
**PROCESS FLOW OF IPC ORGANIZATION:**

1. Girisha Gotti (Information Processing Manager of GAMMA Corporation) notifies Customer Representative (James Johnson) of IPC about failure in software within one business day after encountering it.
2. James Johnson gathers required information about failures, assigns severity level, complete a problem report and sends to maintenance clerk (Geroge Wilson) to assign module queue to a problem.
3. If James found a problem severe enough to halt customer’s business operations, he speaks directly with the maintenance manager (Melinda Shah).
4. Melinda Shah refers Sharleen (Maintenance programming manager) and she sends software developer to the customer’s site immediately and resolve the problem.
5. If problem is not severe, George Wilson assign each problem unique tracking number and mark “waiting to be assigned” status and gives priority to each problem based on its severity and put in the queue of the day.
6. If George finds problem like earlier problem, he links it to the previous problem with the same priority and status.
7. Programming staff takes off the problem from the queue by first come first serve within priority group and programmer is assigned to fix the problem.
8. Problem status has been changed to “Assigned and waiting to be fixed”.
9. When a repair has been assigned to programmer, they assess the problem and presents plan to fix it in front of Configuration Control Board (CCB), and CCB approves it, after reviewing possible problems.
10. After approval from CCB maintenance programmer implements proposed solution and perform unit test on it. Problem status has been changed to “Fixed and waiting for regression test”.
11. Test group does regression test and system level test.
12. If software fails the test, problem has been put in the queue of the day. Problem status has been changed to “Waiting to be assigned”.
13. If passes, it is taken to the CCB for final approval.
14. For final approval, CCB refers maintenance programmer who is making the repair and with the development programmer who originally wrote that part of the software.
15. After final approval, Sharleen prepares the update installation package. During this, status of the problem changes to “waiting for shipment”.
16. Customer representative (James Johnson) sends prepared installation package to Girisha Gotti to install on the systems at GAMMA Corporation.

**CASUAL FACTORS:**

* Multitasking is not allowed for programmer even if he is capable enough because of old company policy which was instituted by a former vice president of the company.
* Every time unit testing is being done from scratch by maintenance programmer because unit test developed by the original programmer are not being saved.
* To resolve the dispute about priority of the problem repair, Maintenance manager (Melinda Shah), Maintenance programming manager (Sharleen Jefferson) and customer representative (James Johnson) must meet in person which takes number of days to match their schedules. This causes delay in overall progress of problem repair. This can be avoided if we use different approach like online meetings.

**SWIMLANE DIAGRAM:**



**Data to Collect:**

* Interviews with key people involved in the process described above. Interviews are intended to extract their perspectives on the process as well as numeric data, where available.
* International Products Corporation Data relevant to this problem (from IPC Data Report)

**Collect Data:** Interview results have been collected and are summarized in various supplementary files.

**ANALYZE:**

**Root Cause Analysis:**

Root cause analysis is examining a business process to find out where it is going wrong. The root cause analysis techniques as discussed in class are as follows:

* Five Whys: Keep asking WHY until you find the root cause.
* Cause and Effect Charts: Categorize cause and then subcategorize them to get a better understanding of what causes are most likely to be causing the problem.
* Fault Tree Analysis: Top down, deductive failure analysis in which an undesired state of a system is analyzed using Boolean logic to combine a series of lower level events.
* Matrix Diagram: A tool allowing a team to identify the presence and strengths of relationships between two or more lists of items by providing a compact way of representing many to many relationships.

From above mentioned techniques, I have used Five Whys and Fault Tree Analysis to find out the root causes of the problem.

After analyzing Professor Frailey’s report on interviews, the three major Root Causes considered are as follows:

* High preference is given to development over maintenance.
* Significant flaw in Company’s programming policy.
* Company doesn’t have any policy to save test cases/code.

**TECHNIQUE 1: FIVE WHYS**

Fie Whys is an iterative interrogative technique used to explore the cause and effect relationships underlying a problem. The primary goal of the technique is to determine the root cause of a defect or problem by repeating the question “Why?” Each answer forms the basis of the next question.

**ROOT CAUSE 1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **High preference is given to development over maintenance.** | | | | | |
| **Why?** | Maintenance programmer needs to debug the software until he finds the actual problem | | | | |
| **Why?** | | Proper documentation of software not available. | | | |
| **Why?** | | | Programmer who develop the software does not give any time to prepare documentation. | | |
| **Why?** | | | | Company switched to “Agile” methods to develop the software which emphasize more on development. | |
| **Why?** | | | | | CEO James Donohu gives more preference to develop new product rather than maintenance of existing product. |

**ROOT CAUSE 2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Significant Flaw in Company’s programming policy.** | | | | |
| **Why?** | Programmer takes several weeks to understand the problem and the software. | | | |
| **Why?** | | Best programmers are busy most of the time solving other problem which may have lower priority. | | |
| **Why?** | | | Maintenance programming manager works under lot of pressure to allocate work to programmers. | |
| **Why?** | | | | Company does not have specific programming policy. |

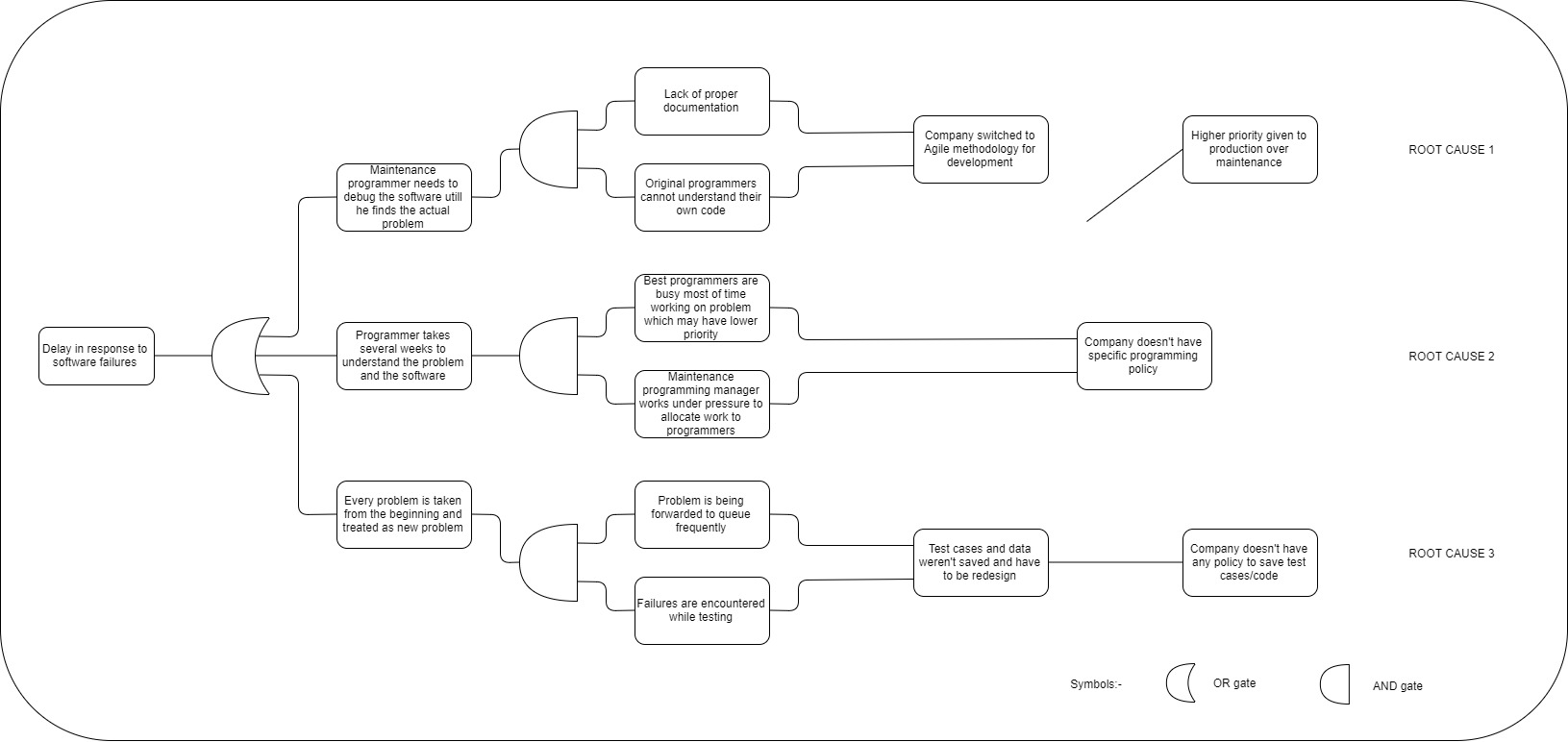
**ROOT CAUSE 3**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Company doesn’t have any policy to save test case/code.** | | | | | |
| **Why?** | Every problem is taken from the beginning and treated as new problem. | | | | |
| **Why?** | | Problem is being forwarded to queue frequently. | | | |
| **Why?** | | | Failures were encountered while testing. | | |
| **Why?** | | | | Test cases and data weren’t saved and have to be redesigned. | |
| **Why?** | | | | | Company doesn’t have any policy to save test case/code. |

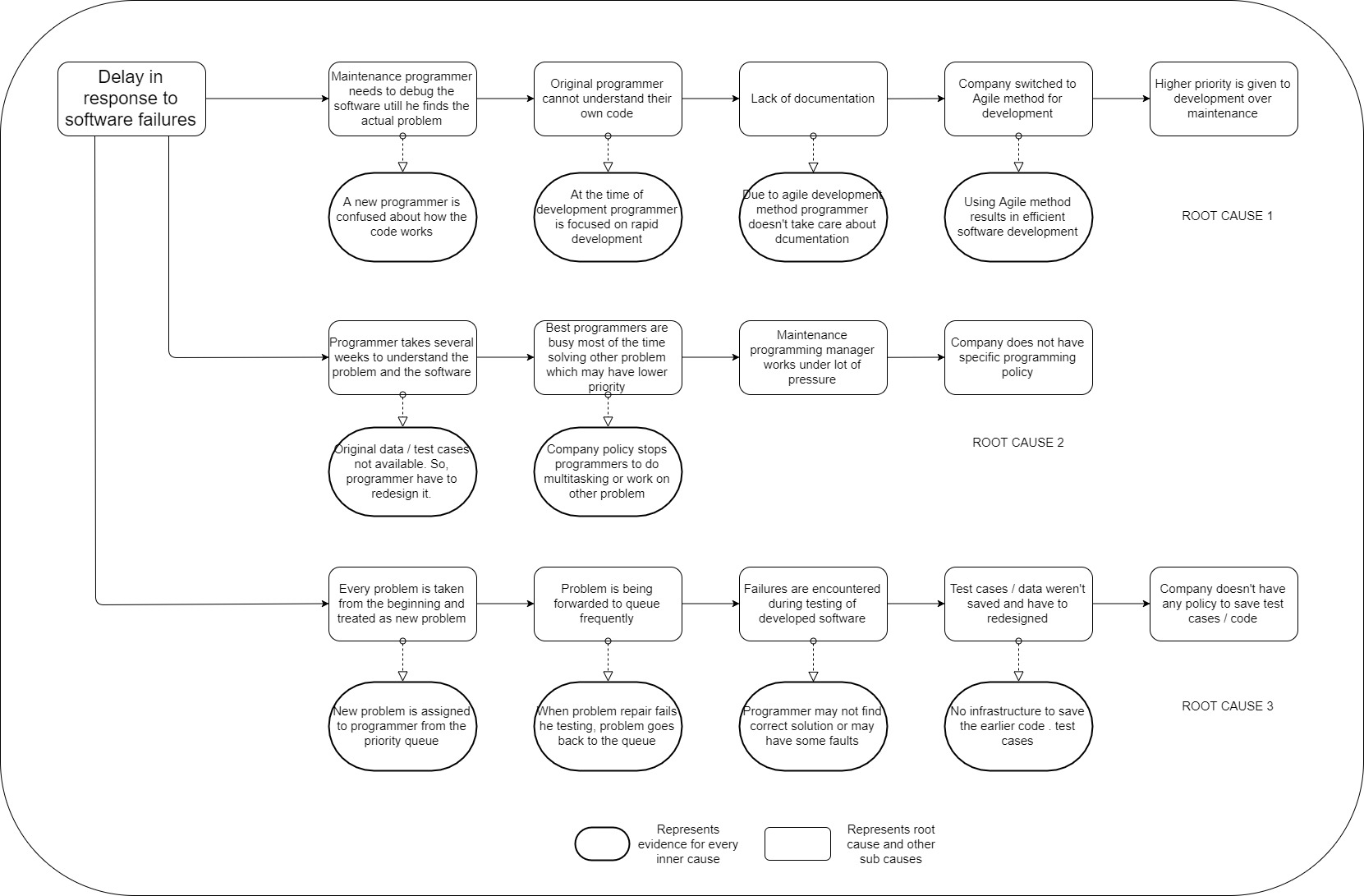
**TECHNIQUE 2: FAULT TREE ANALYSIS**

Fault tree analysis is used for representing all the possible causes in one diagram. [1]

1. Define the undesired event: the primary fault or failure being analyzed.
2. Deduce the event’s immediate causes.
3. Keep stepping back through events until the most basic causes are identified.
4. Construct a fault tree diagram
5. Evaluate your fault tree analysis.

****

Causal Map:



**Collect Data:** To be performed, if necessary, to identify most critical root causes.

**Prioritize Root Causes:** Below is information on each of the three most important root causes, including a flow diagram of each of the three target sub-processes that must be improved to fix the three root causes.

| Root Cause /Target Sub-process 1: |
| --- |
| Root Cause: High preference is given to development over maintenance.  Sub process: Use of Agile methods in development.  Company switched to Agile methodology for software development to improve development time, causes poor documentation of the software. And programmers are not able to explain their own code because they have written the code long time ago therefore maintenance programmers spend long time to understand the software and make wrong assumptions which require rework. |

|  |
| --- |
| Root Cause /Target Sub-process 2: |
| Root cause: Significant flaw in company’s programming policy.  Sub process: Delay in understanding and solving errors by programmers.  According to the policy of company, if a programmer assigned to a particular problem cannot be pulled off to work on another problem or work on two problem simultaneously. So, best programmers often stay busy on problem repair which may have lower priority. |
|  |

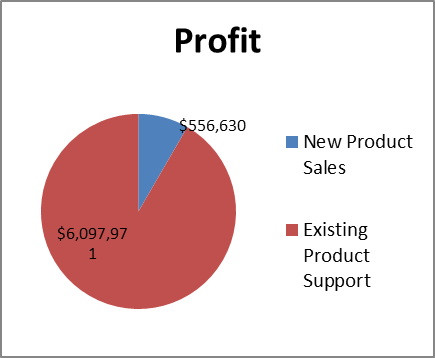
|  |
| --- |
| Root Cause /Target Sub-process 3: |
| Root cause: Company doesn’t have any policy to save test case/code.  Sub process: Problem is tken from the beginning and treated as a problem.  When problem is being put in the queue, it is considered as a new problem and assigned to a programmer based on priority. Programmer has to rework on problem since test data and code isn’t saved so test engineer needs to write new test cases. If failures are encountered during testing, it is put back in the queue of respective module and treated as new problem. That causes delay in overall progress. |

**IMPROVE:**

**Create Solutions:**

**Solution 1:**

Company should give equal importance to maintenance of the product as new software development. It can be observed from the following diagram of IPC’s annual revenue data that company’s net profit from existing product support is $6,097,971 which is approximately 12 times higher than net profit from new product sales $556,630.



This data also supports that company is making more profit from maintenance work than new product sales.

**Solution 2:**

Company should have efficient programming policy that directs maintenance clerk to assign problem repair to a programmer that utilizes maximum capacity of the programmer. Company should remove old policy that stops programmer to do multitasking even if one is capable enough, that increases efficiency of maintenance programmer.

**Solution 3:**

While development of new software test cases and data should be saved for later use that reduces rework of the maintenance programmer that helps to reduce delay time of problem repair.

**Test the Solutions:** To be performed

**Assess Risks of Implementation:** To be performed

**Create full implementation plan:** To be performed

**Deploy the plan:** To be performed

**CONTROL:**

**Control Plan:** To be performed

**Monitor and Control:** To be performed

**Update training, process documents, procedures, etc.:** To be performed

**Develop Response Plan:** To be performed

REFERENCES:

[1] Fault Tree Analysis - <https://www.smartdraw.com/fault-tree/>

[2] https://en.wikipedia.org/wiki/Fault\_tree\_analysis