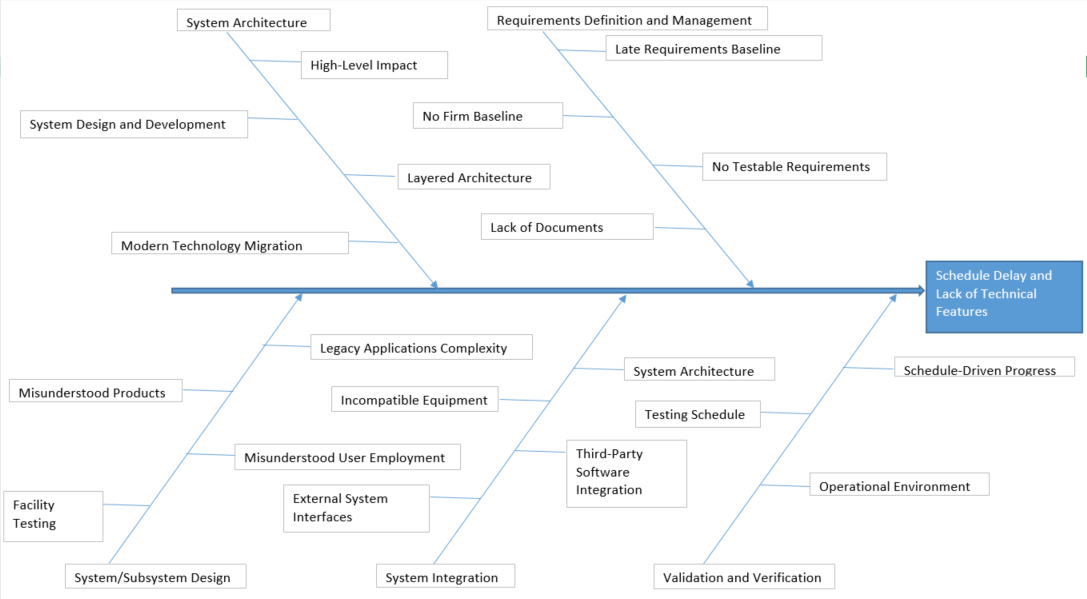
# Analyze

## Process Models for Determining Potential Root Causes

With the problem defined and data inputs captured in the measurement phase, the problem may now be understood at a level where an in-depth analysis could yield benefits for LM, the stakeholders, and the TBMCS. In determining what caused TMBCSs progress to be delayed by years, a variety of causes were investigated and analyzed by cause-and-effect diagrams. The cause-and-effect diagram shown in Figure 3-1 details a high-level breakdown of key factors leading to schedule delays and lack of technical features in the end-product.



*Figure 3-1. Cause-and-Effects Diagram (Fishbone Diagram).*

The Fishbone Diagram above lists off 19 key causes contributing to the main process improvement problem, third-party software integration. The key causes are broken down into their 5 major divisions for cause contribution: system architecture, requirements definition and management, system/subsystem design, system integration, and validation and verification. In Table 2, each process input from the Fishbone Diagram is combined in a relational matrix with each of the 5 cause contribution divisions. By doing so, a wider system perspective can be observed showing how each process input affects the system as a whole. The process inputs can now be compared even in divisions where their impact may not be directly impacted.

The Comparative Analysis seen in Table 2 directs attention to problem diagnosis to refine the ranges or batches where defect(s) source(s) emerged. The difficulty in applying this to the TBMCS system lies in the lack of documentation supporting what explicitly would define a defect. The TBMCS did not necessarily produce an abundant defects throughout its lifecycle but failed to define what software upgrades would be needed to achieve the features desired by the government’s organizations.

Table 1 Comparative Analysis

|  |  |  |  |
| --- | --- | --- | --- |
| Problem statement: | Schedule Delays and Overdrawing Budget | | |
|  | **Is** | **Is not** | **Differences and changes** |
| What | Software bugs and integration complexities | Legacy system baseline to improve upon | Mutual agreement on which software applications to utilize from beginning |
| Where | Mostly in the virtual software space with some hardware interfaces | Defects most noted in lack of functionality in testing and operations | Isolating software bugs before testing on hardware, allow for separate testing for each software integration |
| When | Defects at the design stage, mostly stemming from stringent measures on what would be considered defects | Defining what qualifies as defects more |
| Size | Defects scattered throughout whole system, ultimately affecting lofty operational testing requirements/goals | More consistent testing measures and procedures |

Breaking down Table 2 shows no single defect nor issue produced the problems brought on by the TBMCS. No indication was given that the output of code was specifically causing defects other than issues stemming from the integration of various software products which would not necessarily be defects but would cause delayed schedules. The Is/Is Not Comparative Analysis may not be the most applicable for this system because of its heavy reliance on software functionality. The analysis, however, does highlight how the TBMCS could not produce definitive results to even highlight definitive measures on what would qualify as a defect. The legacy system was supposed to work out all of the kinks in the software development and reduce defects. The main schedule delays happened long before operational testing, when the software was still being implemented and hadn’t yet been tested.

## Methods for Evaluating Potential Root Causes

Many methods and statistics exist to aid in find the problem’s root cause. The sample size of the data would need to be evaluated to determine the significance under evaluation. By testing the means, the data would receive validation in its normality to validate the null hypothesis. A t-test and Z-test could then make inferences about the population mean, with the population variance unknown. The mean, variance, confidence intervals, and proportions would validate the hypothesis generated.

The approach would start with a Cause-and-Effect Relational Matrix shown in Table 2 to narrow down specific root causes showing a high likelihood of the input being a root cause. The matrix is by no means a comprehensive final decision on the root cause given its imperfection but should guide the selection of a root cause.

Table 2 Cause-and-Effect Relational Matrix

The results from the Relational Matrix above show each process input impacting the system at a percentage between 4.60% and 6.02%. Since the variety of the process inputs span a medley of issues within the TBMCS system, it would be expected to not have a couple dominant inputs with large percentage impact. Surprisingly, testable requirements received a low percentage tally because its impact was not widely felt across the whole system, specifically system/subsystem design, and system integration. Modern technology migration’s percentage score of 6.02% scored high due to two high scores in the same areas testable requirements lacked, system/subsystem design and system integration with all three other divisions receiving mediocre ratings around 5. The Cause-and-Effect Relational Matrix demonstrates a plethora of factors contributed to problems in the TBMCS ultimate output.

While the ratings of the importance to the customer may be debated without direct input from the customer, analysis of what the customer wanted from their customer can guide the process for determining the ratings. The government wanted improvements over their legacy system and was not concerned as much with lapses in providing some technical documentation. Requirements definition and management and validation and verification were rated the highest because they both affected the functionality of the end product the most. System/subsystem design received an 8 rating because the government wanted their system/subsystems to contain certain features but showed little desire to get involved with the technical design and means for achieving features they desired. System integration received a 7 rating because the design would be based off a fully-integrated legacy system so they took for granted the ease with which the new system may be integrated with all of the inputs. The lowest rating, system architecture, received a 6 because the government’s involved organizations did not need a complete new architecture to merely upgrade features.

## Identification of Root Causes

From the Cause-and-Effect Relational Matrix, the root cause would appear to be Modern Technology Migration from the System Architecture. The software requested by the government used was antiquated and did not align with the legacy system’s software.