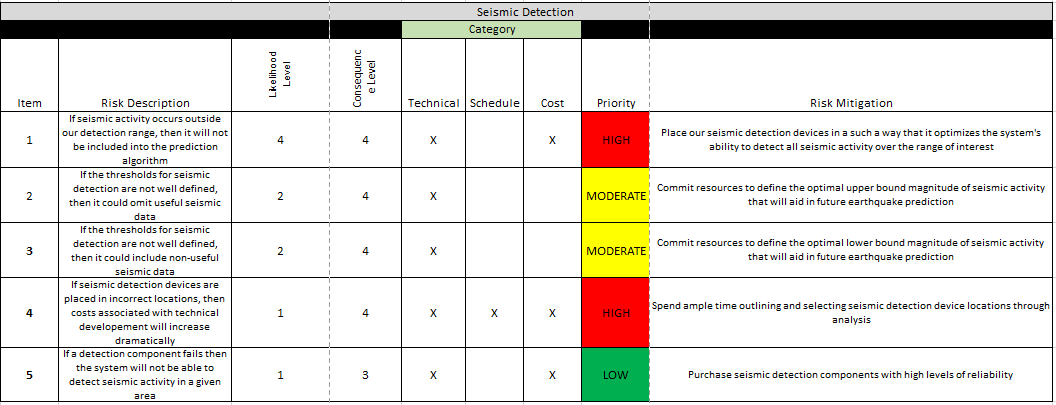
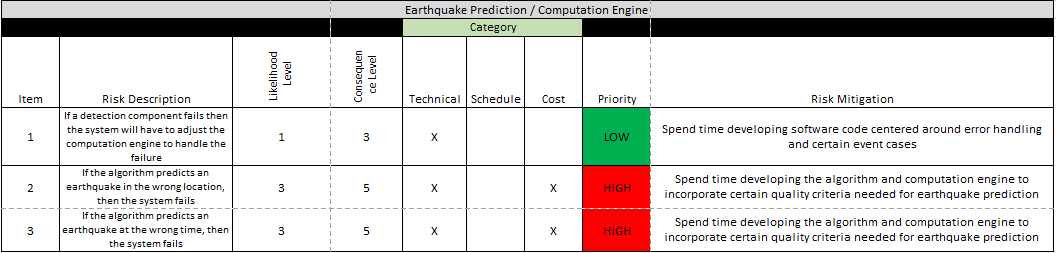
**Homework 3 - Risk and TPMs**

**Group 2 - Gabrielle Davenport, Stephen Jason, Ryan Patton, Mike Thiemet**

**SYS\_ENG 5101**

**Risk Tables**





The system functionalities that were assessed for risks are impacted by the following system requirements:

1. The system shall provide accurate information
2. The system shall detect seismic waves (P-waves and S-waves) greater than 2.0 in magnitude.
3. The system shall interface with GPS technology in order to quantify the movements of tectonic plates
4. The system shall store data provided by data inputs.
5. The system shall allow stored data to be retrieved and analyzed by computation engines.
6. The system shall analyze the stored data to predict the next earthquake.
7. If the system predicts an earthquake of a <certain magnitude>, the system shall send a notification to specified parties.

Detailed Risk Explanations

If seismic activity occurs outside our detection range, then it will not be included into the prediction algorithm. This risk would primarily be technical as it sets the range within which the system will extract useful data. The likelihood of this risk occurring was given a 4 rating because most earthquakes are low magnitudes on the Richter Scale and are for the most part not worth tracking. The risk was given a high priority due to the need for a reliable system. This risk will be mitigated by placing our seismic detection devices in such a way that it optimizes the system’s ability to detect all seismic activity over the range of interest. This will help mitigate the chance of earthquakes falling further away from “hot” areas going undetected.

If the thresholds for seismic detection are not well defined, then it could omit useful seismic data. This risk would primarily be technical as it clarifies what is necessary for a functioning system. Considering the value of the system is driven from the inclusion of every source of seismic data possible, the data must be plentiful. The likelihood of this risk occurring was given a 2 because various methods are already in place to define earthquakes that should be easy to emulate in our system. The priority of this risk was moderate because while it is still a very important aspect of the system, the value in the system does not lie in its ability to accurately define the seismic activity as much as it relies on the time that the earthquakes are predicted by. This risk will be mitigated by committing resources to define the optimal upper bound magnitude of seismic activity that will aid in future earthquake prediction.

If the thresholds for seismic detection are not well-defined, then it could include non-useful seismic data. This risk would include technical aspects because it is simply the opposite end of the previous risk. Ensuring the accuracy of the data is crucial and even though as many sources of data need to be included for possible, the data must be relevant so as not to skew the trustworthiness of the prediction. This risk was also given a low likelihood level at 2 because the data will be previewed beforehand and correlated to how it relates to predicting seismic activity. The associated risk was given moderate priority due to the security of stakeholder confidence in the final product’s data sources. This risk will be mitigated by committing resources to define the optimal lower bound magnitude of seismic activity that will aid in future earthquake prediction.

If seismic detection devices are placed in incorrect locations, then costs associated with technical development will increase dramatically. This risk would be a combination of technical, schedule, and cost. The system cannot afford to go multiple iterations to achieve its desired effects without compromising the final product in some way. This risk was given a very low likelihood level of 1 because seismic detection placements have been studied before and the inacceptable distance away from where the seismic detection devices should be is large given their sensitivity. This risk was given a high priority simply because on the off-chance sensors are placed incorrectly, their data is invalid and cannot be considered in analysis. This risk will be mitigated by spending ample time outlining and selecting seismic detection device locations through analysis. Studying past device locations should provide a very good idea about where to place new ones.

If a detection component fails, then the system will not be able to detect seismic activity in each area. This risk would primarily be cost and technical because additional expenses may be needed to expand the computational function of the system to support the data and analysis and backups may be required. This also falls under a technical risk given the expertise needed to adjust the system’s computational engines. This risk was given a very low likelihood level too because more than one detection components will be placed in the relevant locations and the lifecycle of detection devices is many years. This risk was given a low priority as well because although detection components are needed to detect the earthquakes, the art of purchasing quality detectors and placing the detectors is already an established science. This risk will be mitigated by purchasing seismic detection components with high levels of reliability. Basic research and reviews should guide choosing the optimal detectors for the system.

If a detection component fails, then the system will have to adjust the computation engine to handle the failure. This risk would primarily be technical because it only deals with the function of the computational engine. The likelihood level of this risk was very low at 1 because the risk of detection components failing is low. The priority is also low for this risk because adjusting the computational engine requires only slight software modifications amounting to most likely a couple hours of workarounds in the code. To mitigate this risk, time will be spent developing software code centered around handling errors and certain event case failures. With proper preparation and scenario brainstorming, this risk can be mitigated easily.

If the algorithm predicts an earthquake in the wrong location, then the system fails. This risk would primarily be technical with cost because the earthquake detection location concerns with how the detection data is able to pinpoint the epicenter of the earthquake and the cost of failure is high given the extremely low chances for mess-ups before the system wouldn’t be useful. This risk was given a level 3 likelihood of occurring because the precision of the predictions needs to be extremely high and adjusting the system’s software to retain timely prediction results would be difficult. This risk was also given a high priority because the computational engine’s run-mode is needed to produce the end-result, the prediction. Mitigating this risk means spending time developing the algorithm and computation engine to incorporate certain quality criteria needed for earthquake prediction. The software design will drive the accuracy of the prediction.

If the algorithm predicts an earthquake at the wrong time, then the system fails. This risk would primarily be technical and cost for the same reasons listed for the previous risk. This risk was given a level 3 likelihood rating because it is the inverse of the previous risk. The priority is high for the same reason as the previous risk too. This risk will also be mitigated by spending time developing the algorithm and computation engine to incorporate certain quality criteria needed for earthquake prediction.

If the seismic predictor system needs multiple rounds of $1000+ investments before testing, then the development hours required for a testable system will exponentially increase. The risk is primarily cost as a means of how it ties into schedule. Ideally, the system would not require more than an initial investment and so if it is later determined more money is needed for an operable system, the planning hours for the system were not mapped correctly. The likelihood of this risk is low because once funds have been raised for networking equipment, the only other possible cost could be data acquisitions. The severity of the risk is moderate. If more money is needed for the system due to data acquisitions or scalability it is a good problem to have but if the issue stems from inadequate network funding than the system cannot connect to the vital data inputs. The risk will be managed by deciding on an initial defined list of data sources before ordering the right networking equipment. This risk can be mitigated by initially ordering backup networking parts in case of failures.

If the seismic predictor system cannot connect with its sources of data to interpret the raw data, then the resulting predictions will produce rougher estimates and call into question the validity of the prediction(s). The risk of not connecting with sources of data is mostly technical because it revolves around data extraction techniques. The likelihood of this risk is low due to the simplicity of data extraction dealing with seismic predictions. The data for seismic predictions is for the most part quantifiable and any data mining techniques utilizing qualitative data would be a bonus. The severity of the risk is high. The system is essentially useless without the ability to mine data from various sources because of its heavy reliance on combining different sources of data to produce a better overall product. The risk will be managed by designing the system so there are multiple methods in place for analyzing each source of data. The mitigation plan is to bolster the artificial learning capabilities of the system if fewer data sources can be utilized then what was initially thought.

If the seismic predictor system cannot interpret the data inputs accurately then the seismic predictor system’s software design needs restructuring or debugging and more sources of data may need to be added. This risk is primarily technical as it calls out the software design of the system with some of its network interfaces. The likelihood of this risk occurring to meet a two-week prediction time is low to moderate. Two weeks is a bold prediction time for any seismic predictor system and so a system that produces mediocre timing results within a couple days’ notice will be easy but if we hope to meet a two-week cut-off then the risk leans moderate. With all the data available, two weeks is an ambitious but reasonable time frame for prediction if the software’s analysis methods can tie them all in together effectively. The risk does tie into scheduling as knowing if the system produces accurate results will most likely need several full-scale earthquakes to take place before the system’s accuracy can be interpreted and quantified. The risk will be managed by comparing our system to similar systems’ timings, possibly finding a way to utilize a shaker testing unit so the actual earthquake measurements are accurate and thereby eliminating eliminating that as a misinterpretation of the system’s inaccuracy, and ensuring our system is predicting any earthquake across the globe once it is up and running to see as early as possible if changes need to be made to the software. The risk will be mitigated by setting up “dummy”, more basic, software programs to compare its accuracy and timing to verify all the modeling and prediction techniques put into the software are improving the timing and accuracy.

If the seismic predictor system cannot utilize a relevant software technique to predict earthquakes, whether that be by Monte Carlo simulations or another method, then the system will most likely fall behind schedule for manufacturing, assembly, and testing. This risk relies on technical methods to produce adequate software, but the primary concern is schedule. Developing a software program to produce very rough results on the time of the schedule is not the concern if enough assumptions are made but refining assumptions to give as accurate results as possible is more difficult. The system’s goal is not only to produce a prediction but to produce a prediction with assumptions that can be made with confidence, which takes time to analyze and study the effects. The likelihood of this risk occurring is high, but the severity of the risk is low. The severity of the risk is low because as mentioned earlier, even if the risk does occur it will most likely occur in a scenario where it is a matter of refining the accuracy of the software and not so much a matter of having running software. Quick searches yield various methodologies for analyzing earthquakes, but it could be difficult integrating all the various data points to produce well-rounded software and fine-tuning the computational intelligence of the system to learn as more data is inputted. Relying strictly on meeting system requirements, the risk is low but as with most systems if the goal is to improve on system requirements then the schedule and life-cycle hours put into the software has a high chance of falling behind schedule. The risk will be managed by ensuring a certain number of hours have been put into the initial software design so any changes to the software will be minor and the design will be flexible for added data inputs and refinement techniques. Proper preparation will lead to a better overall system initially and more importantly, into the future. This risk could be mitigated by properly setting aside enough time to research the intricacies and details of the software design so each of its desirable feature choices can be made with confidence

If the notification system cannot quickly inform stakeholders of a potential earthquake, then the system will need to utilize an alternative form of communication. This poses primarily a cost risk given that other notification methods can be costly. The likelihood of this issue occurring is low given that we are using a common communication method. There are multiple examples of how this method has been implemented before. This risk will be managed with testing early on to identify if we need to use an alternative form. We can further mitigate the risk by developing this communication at the same time as other components of the system since testing communication functionality does not necessarily require other components to be working.

If the data center cannot store and process data, then the system will not be able to use past data to identify potential earthquakes. This will cause scheduling delays since this is a key component required for the system to go into production. Furthermore, it will be difficult to test other components of the system without the data center. This risk will be managed by centering the development and implementation of other components around the data center to accommodate its needs.

TPM Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Technical Performance Measure (TPM)** | **Source Requirement** | **Quantitative Performance Requirement** | **Current TPM Value** | **Risk of Not Meeting TPM** |
| Percentage of earthquakes predicted within required time frame | 1. (see requirement above) | Within 72 hours of predicted time vs. actual time | 95% within 72-hour window | 1 |
| Percentage difference between earthquake measured on Richter Scale and magnitude measurement of system | 2. (see requirement above) | 2.0 magnitude on Richter Scale | 5% difference | 4 |
| Time improvement increments with number of GPS satellites utilized | 3. (see requirement above) | GPS real-time lag time less than 120 seconds | 5 seconds per satellite added | 4 |
| Number of data sources | 4. (see requirement above) | 5-10 Terabytes of storage allowed by computer(s) | 10 sources or more | 1 |
| Percentage of data relevant and practical for analysis | 5. (see requirement above) | 5-10 Terabytes of storage software supports for analysis | 90% | 2 |
| Accuracy improvements in percentage with each data source added for analysis | 6. (see requirement above) | Less than 60 minutes of analysis for each source of data mined | 2% or more | 2 |
| Number of people reached by notifications | 7. (see requirement above) | Number of devices notified (1,000,000+) | 1,500,000 people confirmed | 3 |

\*Scale for above table:

1 = Very High

2 = High

3 = Moderate

4 = Low

5 = Very Low

For the percentage of earthquakes predicted within required time frame, 95% within 72-hour window, it will be measured by accounting for the time that the computational engine predicted and waiting for the time when the actual earthquake occurs. The software will store values to monitor the prediction times to monitor their relative success. The TPM cannot be 100% within a 72-hour window because of fault locations that are inaccessible and therefore difficult to monitor. The percentage of earthquakes predicted within a time frame should improve over time as the software learns from past data and so it would not be unreasonable to expect 98-99% over the system’s life cycle.

For the percentage difference between earthquake measured on Richter Scale and magnitude measurement of system, 5% difference, it will be measured by consulting other sources’ Richter Scale measures and comparing them to our system’s measure. To monitor the difference, the software will store values for magnitude measurements and if/how they change over time. The percentage magnitudes predicted should incrementally increase over time although the initial runs should be close enough to prove the incremental improvements insignificant. The differential between the measurements is accounted for by considering measurement proximity to the fault line under study and the precision of the instrument used.

For the time improvement increments with number of GPS satellites utilized, 5 seconds per satellite added, it will be measured by comparing past trial times to new trial times with GPS satellites added. To monitor the time improvements, the software will store values relating the number of satellites to computational time. Throughout the life cycle of the system, more satellites would ideally be added to cover fault locations more completely. It is hard to say that each satellite added would improve analysis time at a linear scale, but the average should balance out to an approximately linear relationship. As already mentioned, the measure is limited by linking complications and the effectiveness of each satellite to cover faults equally.

For the number of data sources, 10 sources or more, it will be measured by counting the number of data sources. Further the amount of data gathered from each source can be stored from the software. Monitor the number of data sources will involve pipelining the data flow and defining criteria to validate each data source added is cleared as accurate. The problem with ideally adding more and more data sources throughout the system’s life cycle is that the quality of the data must not suffer significantly. Finding new data sources to mine would not be difficult but qualitative and quantitative evaluation should indicate that the data still adds to the value of the prediction.

For the percentage of data relevant and practical for analysis, 90%, it will be measured by auditing the added data. The audit will include criteria to define the data as relevant. The data sources will need to be monitored to see if any changes to the data sets affect its relevance and, which changes could be considered irrelevant out of the whole. Whole data sets may be added for evaluation but if only 70% of the data in the set is relevant than this needs to be tracked and noted. Adding more and more data does not reduce the value of the system even if only some of the data is relevant but half-hazard adding data sets could lead to sloppy assumptions and untrustworthy results.

For the accuracy improvements in percentage with each data source added for analysis, 2% or more, it will be measured by comparing previous prediction iterations to new iterations with added data. To monitor the accuracy improvements, the timing between the prediction and actual earthquake will need to be stored and evaluated. Two percent is a conservative initial improvement but once the system is operating with sufficient data sources, a 2 percent threshold becomes more reasonable. The accuracy improvements are directly tied to the number of sources of data already evaluated with each run. Given the inherent unpredictability of earthquakes it is difficult to stick with a 2% incremental improvement above 90%.

For the number of people reached by notifications, 1,500,000 people confirmed, it will be measured by notification feedback. In addition to the number of notifications sent out, perhaps metrics can indicate whether the notification was open or at least seen. Second-hand word-of-mouth is more difficult to track which is why the 1,500,000 people confirmed TPM is relatively low for a massive-scale large city earthquake. To monitor the number of people reached by notification, a confirmation button could be added to the notification to confirm the individual received and read the message. The problem with notifications is that just because an individual received and read the message does not mean action will happen as a result and that the individual did not skim over the message without comprehending its possible impact.