# TIMP-CE Roadmap

Overhauling the entire risk model process is unrealistic, disruptive, and prone to errors. We should insert changes in steps, in modular ways such that improvements to the process and risk models is **incremental** and **measurable**.

The main stages of TIMP Risk Management we should focus on include:

* Data gathering
* Data integration
* Model run
* Validation
  + data validation (prior to integration)
  + results validation (model outputs)
* Result rollup/presentation

#TODO: Fill out more details about each stage.

## Toward measurable model improvements

Any improvements to a process need to be measurable – ideally we can devise a single metric, or a set of consistent metrics that can measure risk model performance, and which can be compared through time.

In addition, model performance metric needs to be able to be validated with collected data.

This set of model metrics need to be agnostic to the model implementation details, and capable of convincing SME of a model’s quality.

Examples:

* How about classification based metrics such as precision/recall?
  + They would work for certain risk types where classification data is available. But it’s not ideal since adverse events are in general sparse, and our risk models is required to output LOF. We can attach a threshold-based classification stage after LOF and derive classifications. But ideally the performance metric can also be a probability-based metric.
* What about using amount of predicted rupture events to measure model performance?
  + This metric cannot be validated by data, so it’s a bad metric.

### Timeline for model performance

1. Use one specific threat as a starting point, work with the risk engineer to understand current model flow and performance measurement (Ian suggests SSWC with Brian Patrick).
2. Decide on an approach, taking into account of available data types to do validation with.
3. Implement and validate – can we apply this to previous years’ models/data combinations?
4. SME approval/implement in process?

**(1-3) Aimed for September currently**

We can take a threat-by-threat approach, before making a generalized method to reduce difficulty and data requirements. Prioritize performance measurement only for threats that inform assessment, for more measurable impact.

Overall game-plan:

1. Pick a threat
2. Make something
3. Iterate??
4. Profit???

## Incremental improvements focused on Validation

A huge time sink is the data input validation stage. Several things we could do:

1. Inherit previous year’s validation dataset if no changes occur – as time goes by, the model coverage improves rather than starting from 0.
2. When there are new cases (e.g. new code label, or segment with combination of characteristics), focus validation on these set. Essentially, replace random sampling during validation with targeted sampling to save iteration time and increase test case coverage.
3. Integrate database and validation procedures with Foundry – to improve model and data tracking

The current data integration and validation stage has the following functional steps:

1. Risk engineers check the factors their specific risk depends on have the correct data and is entered according to their interpretation of the model (i.e. scale point assigned correctly). This is done for random sampling of 250 pipeline-segments, with “ground-truth” calculated manually in excel.
2. After the data is entered into MarinerDB, and the model runs complete, the model results need to be validated.
3. In this step, the assumption is that the MarinerDB data is already cleaned and correct. 25k random samples (pipeline segments) are picked from the database, and their associated risks are calculated by the risk engineers “manually” in excel sheet.
4. These manual calculations are compared to the automated model outputs. Whenever mismatches occur, these are addressed between the risk engineers and Jackson.
5. This process continues until all 25k samples have matching results with the engineers’ calculations.

### Why do the engineers need to do 250 manual input data validation every time?

The model may change every year, assigning different qualitative factor to the same value. The data may be entered incorrectly, etc.

### Why is 25k random sampling required every year?

The model may change every year, Jackson and the engineers might have different interpretation of the model. Edge cases that were unexpected or unseen before might occur.

### Can we make the validation process more automated and robust?

There are definitely opportunities for this. Take the 25k random sampling points for example. Every year a different sample is taken, this is likely not the same set as the previous year’s, even though for some percentage of the pipelines’ model/data change might not be present. There is duplicate work here from year to year.

The reason for this validation process is it’s hard to anticipate all cases, and the hope of the random sampling process is it might catch some of these unexpected cases. In theory the set of all the past year’s random samples should contain a bigger and bigger proportion of all possible cases. This should be taken advantage of.

We can analyze the possible causes for a pipeline segment’s model output to be different from year to year.

Suppose in year 1, modelX is calculated as y = A1 + B1 + C1, where y can be considered as LOF of type X, and A1, B1, and C1 are risk factors.

The types of data and modeling changes that can impact modelX calculations from year to year include:

* **Substitution**. Ex: In year 2, the name of A1 has changed to A1'. But the model and the meanining of the risk factor doesn’t change, in this case nothing needs to be doen to re-use year1’s test case.
* **Numerical**. Ex: For some of the time-dependent factors, the scaling needs to advance. So if say B1 is in fact B1(t), and has value based on 5 years pipe age in year 1, it would advance to that of 6 in year 2.
* **Modeling**.
  + Factor changes: In year 2, the calculation or derivation of A1 has changed. This might be due to:
    - If the collection method A1 has changed (i.e. qualitative scale), then year1 test cases’ modelX values cannot be re-used.
      * This can also include the case where new code list values are introduced in year2.
    - If the caclulation of A1 has changed due to a substitution or numerical change in its component, then year1 test cases can still be added after propapgating these adjustments.
  + Pipeline characteristics changes: A specific pipeline segment test-case from year1 might not exist in year2 anymore, due to change in the combination of its characteristics used to define that segment previously. Here the test case cannot be re-used.

Substitution changes is handled in data collection by engineers and Steven/Uriel. Numerical advances is done in Jackson and Gordon’s code. Previous test cases where these are the only impacting data/modeling changes can be re-used here.

A potential solution would involve better tracking of model and risk factor changes from year-to-year, and keeping a repository of every year’s test cases. Currently 25k samples is big for Excel, therefore better data solution is needed. Foundry seems like a good solution for all these problems, but there is organizational difficulty to implement a Foundry-based solution. Steps we can take: 1. See if we can run the risk model from MarinerDB in Foundry 2. Start keeping validation sample database in Foundry 3. For next year’s model, keep track of the amount of substitution, numerical, and modeling changes – for this solution to make sense, the amount of modeling changes shouldn’t be a majority of changes.

## References

* TIMP Risk model [road map](https://pge-my.sharepoint.com/:b:/r/personal/ixh8_pge_com/Documents/Microsoft%20Teams%20Chat%20Files/RIsk%20model%20road%20map_final_June2021.pdf?csf=1&web=1&e=yePQc4)
* [Data sources presentation slides](https://pge-my.sharepoint.com/:p:/p/s3lg/Ec3AUFhrkw1Bke6Zf86sa44BGPXI5i9Y9MKDzVwEvkjapg?e=1dYRqR) describes the risk data collection and validation process.