

Interim Report - Ilmarinen

MS-E2177 - Seminar on Case Studies in Operations Research

Terho Mutikainen, Aaro Niinistö, Tuukka Pallonen, Panu Salonen

May 19, 2026

1 Changes in the project objective or scope

Our project is still focused on providing a comprehensive risk assessment model for Ilmarinen which is compatible with the Finnish Solvency Law and other applicable legislation. There has not been any major changes in project objective or scope after the project plan and we are progressing towards the goals set by our client as planned.

2 Project Status

2.a Completed Tasks

We have completed the literature review, where we studied different aspects of CDO risk modelling. First, we looked into how credit rating agencies rate the underlying assets and CDO tranches. Next, we looked into different possibilities for building the model, more specifically Vasicek/One-factor Gaussian copula which seems to be the most suitable approach when considering our input and desired outputs. Also, we got some reassurance for this approach based on the email discussion with one of Ilmarinen's experts who handles CDOs in their daily work. After deciding to proceed with the Vasicek model, we researched further into how different parameters, most notably correlation with market factor, should be estimated according to literature and regulation (Basel 2). Finally, we built the initial version of the model and simulation in Python.

2.b Ongoing Tasks

2.b.1 Checkup with client

Now that we have developed the initial version of the model, we will discuss with the client to confirm that our solution is what they were looking for. We'll

present to them what our model does, get feedback and adjust our model accordingly.

2.b.2 Further development of the model

With the feedback we will adjust the model and continue developing it further. Possible additions would be the incorporation of interest rate change to the valuation of the CDO tranche.

2.c Future Tasks

2.c.1 Model testing

The model needs to be thoroughly tested. We aim to use tests of different granularity to assess the performance of the model. We will use unit testing, stability assessment of the Monte Carlo, sensitivity analysis and finally benchmarking. Unit tests could be test cases with real data from Ilmarinen: we would then compare the output of the model to the realized result of the investment. We will also test the stability of the Monte-Carlo by evaluating the standard deviation of the results per run to see how stable it is. Then end to end sensitivity analysis will be conducted to ensure that parameters that have larger error bounds will not affect the results of the simulation disproportionately.

2.c.2 Model simplification

The final step would be to simplify the model to the format requested by the client. We will run simulations on predetermined CDO compositions and output those into an Excel table.

3 Updated Project plan

While the fundamental objectives and the overarching scope of the project remain closely aligned with the original project plan, the focus of our work has naturally transitioned from theoretical exploration to practical implementation and validation. The original plan emphasized a broad literature review to identify the most suitable modeling frameworks for CDO risk, including the evaluation of various copula structures and recovery rate distributions. Following our research and consultations with Ilmarinen's experts, we have decisively locked in the Vasicek one-factor Gaussian copula model as our core engine and adopted Basel 2 guidelines for correlation parameter estimation. This narrows our methodological trajectory, allowing us to shift away from continuously evaluating alternative structural models and instead focus exclusively on refining this specific framework.

As a result of this methodological certainty, the development phase has progressed efficiently. We have already translated the theoretical mathematical

models into an initial Python-based risk engine, which incorporates the requested Monte Carlo simulations, stochastic Beta-distributed recovery rates, and tranche-specific waterfall logic. Consequently, our immediate project plan is now heavily weighted towards the rigorous evaluation of this computational model. Instead of exploring new risk variables, the upcoming weeks will be dedicated to ensuring stability, conducting sensitivity analyses on key inputs like default probabilities and market correlations, and verifying that the tail-risk outputs align with economic intuition.

Looking ahead, the most critical direction change in our project plan concerns the final phase: model simplification. Originally proposed as a general requirement to ensure the model's usability, this phase is now identified as our primary technical challenge. Because the implemented Python risk engine is computationally intensive, transforming its high-dimensional outputs into a static, user-friendly format for daily operations – such as a spreadsheet table or a polynomial rule set – will require significant effort and careful mathematical approximation. The project timeline remains largely unchanged, ending with a complete final report in mid-May, but our daily efforts will be strictly channeled into bridging the gap between our complex stochastic simulations and the simplified, regulatory-compliant tool requested by the client.

4 Updated Risk Management Plan

At this stage of the project, the main project risks have shifted from broader uncertainty to model calibration, validation, implementation, and simplification risks. In our project plan, one major concern was whether the chosen copula framework would be suitable for modelling CDO risk in the first place. However, following the literature review, client discussions, and initial model development, we have already committed to the Vasicek one-factor Gaussian copula as the core risk engine and implemented an initial Python-based Monte Carlo model with stochastic Beta-distributed recovery rates and tranche-specific waterfall logic. Therefore, the most relevant risks are now associated with the correct estimation of the key parameters, especially probabilities of default, recovery assumptions, and correlation with the common market factor, as well as the numerical stability of the Monte Carlo simulation and the correctness of tranche-loss allocation.

In addition, the final deliverable required by the client is not the simulation engine itself but a simplified spreadsheet-style or rule-based tool, which creates a separate approximation risk. That is, even if the full model performs well, the simplified version may lose too much accuracy or become too difficult to justify in practical risk management. For this reason, our updated risk management approach focuses on the following areas. First, sensitivity analysis for PD, LGD, correlation, and duration-related assumptions. Second, convergence and stability diagnostics for the Monte Carlo estimates, especially the 1-year 97% VaR. Third, verification of the implementation through unit tests and economic

benchmarking. Fourth, parallel validation of any simplified proxy model against the full Python engine before final delivery. This ensures that the remaining work is aligned with both the demands of the model and the client’s practical requirement for a transparent and usable end product.

Table 1: Updated identified risks and corresponding mitigation actions.

Risk	Likelihood	Impact	Effect	Mitigation
Poor internal communication	Medium	Medium	Inefficient division of work, duplicated effort, and delays in testing and reporting.	Regular internal meetings, ownership of tasks, and mandatory short status updates.
Poor communication and collaboration with Ilmarinen	Low	High	Final model may not match the client’s intended inputs, outputs, or usability requirements.	Validating assumptions and interim outputs with the client early, especially before simplification.
Parameter calibration risk	Medium	High	Mis-specified PD, LGD, or correlation assumptions may distort VaR and expected return estimates.	Conducting low/base/stress sensitivity analysis and documenting parameter choices.
Monte Carlo instability	Medium	High	The estimated 97% one-year VaR may vary too much across runs, weakening confidence in the output.	Performing convergence checks, increase simulation count where needed, and reporting simulation uncertainty.
Recovery-rate specification risk	Medium	High	Poor Beta-recovery assumptions may understate or overstate tail losses, especially in stressed scenarios.	Test alternative Beta parameters and stress scenarios, and compare stochastic versus fixed recovery cases.

Risk	Likelihood	Impact	Effect	Mitigation
Model limitation risk	Medium	Medium	The one-factor Gaussian copula may miss concentration effects or more complex dependence structures.	Stating model limitations clearly and use benchmarking plus conservative stress testing where needed.
Complex model cannot be simplified	Medium	Medium	The spreadsheet table or rule set may approximate the Python engine too poorly for operational use.	Validate the simplified proxy against the full model on a scenario grid and use conservative approximation where necessary.
Poor documentation and black box nature of the final model	Low	High	Ilmarinen may find the final tool difficult to audit, maintain, or explain internally.	Maintain a clear, commented codebase and provide a short technical guide for formulas, parameters, and interpretation.
Inadequate personal task management	Medium	Medium	Testing, simplification, and reporting may overrun, reducing the quality of the final deliverable.	Establish clear internal milestones and a clear task dependency map to monitor weekly progress.