Bsc. Thesis Luuk Tijhaar

”Agentic AI in dike design”

July 12th 2025

Witteveen + Bos

UNIVERSITY OF TWENTE.

Bsc. Thesis Luuk Tijhaar

"Agentic AI in dike design"

July 12th 2025

Supervisors:

Dr. Ir. Anouk Bomers
(University of Twente)

Ir. Koen de Jong
(Witteveen+Bos)

Ir. Richard Debets
(Witteveen+Bos)

Bachelor thesis for Civil Engineering
Collaboration of the University of Twente & Witteveen+Bos
Photo front page: Van Wijk n.d.

Preface

Before you lies my BSc thesis, which I submitted in support of my Bachelor's degree in Civil Engineering and Management. This research was conducted in collaboration with the University of Twente and Witteveen+Bos. My graduation study has been fascinating, I learned a lot about myself and programming, and I would like to thank everybody who has helped me accomplish this goal.

To start, I would like to thank my daily supervisors at Witteveen+Bos, Koen de Jong and Richard Debets. Koen, thank you for your enthusiasm and help in developing the idea behind the thesis. During the thesis, you were always a great help and put a lot of effort into answering my questions despite your full agenda, and you were a great sparring partner. As for Richard, thank you for your excellent advice on programming and your knowledge of academic writing and justifications. You put in a lot of effort, and I appreciate it a lot.

Secondly, I would like to thank Anouk Bomers from the University of Twente. Anouk, I know I have not been the ideal student, and my work has not always been of the highest quality. However, thanks to your effort and extensive feedback (which is precisely what I needed), I have made significant improvements since I started writing my proposal. Additionally, I would like to thank you for your advice and guidance.

I would also like to thank my family and especially my parents. You have always been there for me and supported me through thick and thin. I am privileged to have such loving and supportive parents.

Lastly, I would like to thank my colleagues from Witteveen+Bos who made time available to assist me with my research: Lennart Stelling, Louise Guichelaar, Jetze Twijnstra, Patrick Hannema, Selina Holtrop, and Maya Drouin.

I hope you find the thesis and its outcomes interesting, and I wish you a good read.

Luuk Tijhaar
Enschede, July '25

Summary

Over the past few decades, dike design has evolved from a technical discipline into a multidisciplinary approach that incorporates various social and environmental aspects. Additionally, the requirements for dikes have changed as they have shifted from being flood probability-based to flood risk-based. The new approach has complicated the calculations and overall process of dike design, leading to long design processes which span several years. This is an expensive process that requires extensive calculations, stakeholder participation, and numerous permits. Automation is required to improve the efficiency of this process. Since AI represents a paradigm shift in automation applications, this study aims to explore the application of agentic Artificial Intelligence (AI) in dike design by developing a proof-of-concept system that designs a dike with a focus on macro-stability and stakeholder integration. This Agentic AI aims to replicate the decision-making process in dike design and independently design a dike.

The central research question of this study is: "What is the added value of agentic AI in automating and optimising dike design, with regard to efficiency, accuracy, and flexibility?"

To address this question, the research is structured around three subquestions focusing on:

1. The structure of the system architecture.
2. The design and functioning of individual agents.
3. The evaluation of the developed model.

Agentic AI refers to systems in which multiple specialised agents collaborate to perform complex tasks. These agents are typically powered by Large Language Models (LLMs), enabling them to interpret inputs and generate suitable outputs. Some agents are equipped with specialised tools for specific operations, while others rely on extensive contextual information embedded in their prompts.

The architecture of the agentic AI system consists of four agents. The stakeholder agent identifies and ranks stakeholders while simultaneously setting demand for the dike design. The other three agents together form a design loop with a supervisor agent that determines the flow of the design loop, a dike design agent that assigns dimensions and assesses macro-stability, and a demand-check agent that verifies the design against the system's set demands. The stakeholder agent and the dike design agent are built on established theories, and the dike design agent utilises a metamodel for macro-stability assessment.

The model evaluation utilises Covello & Merkhofer's risk assessment framework to evaluate the model; this framework assesses six criteria, and one is added for this specific model. The extra aspect is flexibility. Methods to assess the model of these criteria include an expert elicitation, analyses of the model's results, and comparing a design from the model to an actual dike, in this case, the Grebbedijk. The model scores well in terms of flexibility, completeness, practicality, and effectiveness. Average on accuracy and acceptability, and low on logical soundness, which includes transparency and consistency. Transparency is a common issue for AI systems due to the complexity of their internal processes. Since the model relies on an LLM, the quality of the results depends on the LLM; when the LLM is inconsistent in its results, the model also becomes inconsistent. This is the biggest flaw in the model.

Based on the current study, it is recommended to continue developing Agentic AI by researching the performance of various LLMs, LLM settings, and different structures, such as an agentic workflow, rather than an agentic AI system. These factors have a significant impact on the performance of such models. Agentic AI systems are beneficial in the initial design phase of projects due to their time efficiency, but are limited in consistency and therefore in actual decision-making.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Context

Traditional civil engineering was focused on technical solutions. A design was considered good when it was a stable and solid construction. Over the years, the design process has evolved from technical disciplines towards more environmental, social, and ecological disciplines. This paradigm shift is also noticeable in dike design (Brugge, Rotmans, and Loorbach 2005). Initially, the focus was on high and solid dikes; however, it has now shifted towards optimisation while ensuring safety. This regards optimisation, considering both technical aspects and stakeholder integration, while limiting space, environmental, and ecological impacts, to ensure an economically viable design. This multifunctional dike design approach has been explored by Van Loon-Steensma and Vellinga (2014). Additionally, the requirements from Rijkswaterstaat in the Netherlands for dikes have shifted from a flood probability approach towards a flood risk approach (Rijkswaterstaat 2017).

For the new flood risk approach, dike designers need to create a failure budget, which is a budget that sums the chances of failure for every failure mechanism, ensuring that the sum of these chances equals the maximum chance of failure of the dike. Some of the failure mechanisms that should be considered are: overtopping, piping, macro-stability, and outer slope erosion (Rijkswaterstaat 2017). These new rules regarding dike design complicate the design process with complex and optimised design requirements requiring extensive and precise calculations (Rijkswaterstaat 2017).

Simultaneously, the design process has become more complex by considering other factors surrounding the design. Stakeholder integration has become important; every civilian or organisation can oppose big infrastructure projects, causing significant delays in the design and execution process. The effectiveness of stakeholder integration is, for example, explored in Begg et al. 2017. This study examines the integration of stakeholders in projects in Germany and England. The main conclusion is that participation occurs, but within pre-set boundaries, and is limited by the community's financial constraints.

Because the Netherlands is densely populated, the chance that houses or infrastructure are built next to or on dikes is significant. All stakeholders have their opinions, but not everything can be considered and included in the final design. To ensure that important stakeholder preferences are included in the design, stakeholders must be ranked in order of importance, giving priority to one stakeholder over another. The primary stakeholders in these projects are Rijkswaterstaat and the local water board. Both have an extensive list of standard demands for the dike regarding aesthetics, safety, and spatial and environmental factors. In addition to the most important stakeholders, numerous local stakeholders are directly affected by such a project, as well as various interest groups with a vested interest. Due to its complexity and social approach, stakeholder analysis and integration have remained a mainly human task with limited automation.

1.2 Research Objective

Over the last few decades, dike design has become an extensive process that requires a significant amount of time and money. To improve the efficiency of this process, the goal of this research is to develop, analyse, and test a method for automating two aspects of dike design. One technical aspect is macro-stability, and another social aspect is stakeholder integration in the project. This involves developing an autonomous Agentic AI framework and specialist agents which utilise Large Language Models (LLMs) related to the separate disciplines that are included in the system, the resulting model is a proof of concept idea. It will be evaluated in this research to discover whether this type of system can contribute to efficient and reliable dike design.

The study will investigate how specialist agents can replicate real-life dike design processes and whether the quality of the results these agents produce accurately reflects the work done by real-life experts. The ultimate goal of this research is to assess and estimate the added value of agentic AI in the context of dike design.

1.3 Research Questions

The main research question guiding this project is:

> **"What is the added value of agentic AI in automating and optimising dike design, with regard to efficiency, accuracy, and flexibility?"** With added value referring to the potential contribution of agentic AI to improving the efficiency, quality, or decision-making process in dike design, agentic AI offers added value when it can enhance the dike design process.

To address this overarching question, the project comprises three specific research questions:

1. **What is an efficient system architecture for agentic AI in dike design?**

   This question focuses on the setup of the agentic AI model, including the interactions between agents, the model's flow, and the actions it can take. This prepares for the implementation and collaboration of individual agents within the framework.

2. **How can specialist agents for macro-stability and stakeholder integration be developed using existing engineering models and stakeholder theory?**

   This question focuses on the development of agents based on existing theories in the areas of macro-stability and stakeholder integration. To test the framework, it needs to be evaluated.

3. **What are the advantages and disadvantages of using agentic AI in dike design?**

   Here, the results and processes of the agentic system are analysed, evaluated, and compared to traditional engineering designs.

1.4 Reading guide

This thesis presents a detailed investigation into the development and evaluation of an agentic AI model for dike design with an emphasis on macro-stability and stakeholder integration. The methodology in Chapter 2 introduces agentic AI and expands on the methods for developing an agentic AI framework, specialised agents, and the framework and methods of model evaluation.

The resulting framework and agents from the method are presented in Chapter 3, the results. The result contains the actual framework and agent as well as the results from the model evaluation. The discussion in Chapter 4 examines the assumptions and potential weaknesses of the method and model. In Chapter 5, the conclusion presents an elaboration of the interpretation of the results and answers the research questions. Finally, in Chapter 6, recommendations are made, accompanied by a call to action, including suggestions for further research and the application of these models in the field of civil engineering.

2 Methodology

This methodology outlines the methods for constructing an agentic AI system that can independently design a dike, as well as the method for evaluating the same model based on the Covello and Merkhofer (1993) framework. The overall method is shown in the flowchart in Figure 1.

Initially, this method emphasises the development of the overarching agentic framework, which outlines its construction through the integration of existing frameworks, namely LangGraph, as well as the specified system requirements. Utilising these elements, the framework is built according to defined steps and subsequently tested for functionality.

Resulting from the system architecture, the agents that are part of the flow are constructed by first setting functional requirements, building and testing the agent on functionality by comparing it to the functional requirements, which is done per agent.

Finally, the model is evaluated by performing an expert elicitation, presenting the model's results, and comparing model performance to an actual dike in the Netherlands. Using Covello and Merkhofer's (1993) risk assessment framework, the system is evaluated based on seven categories.

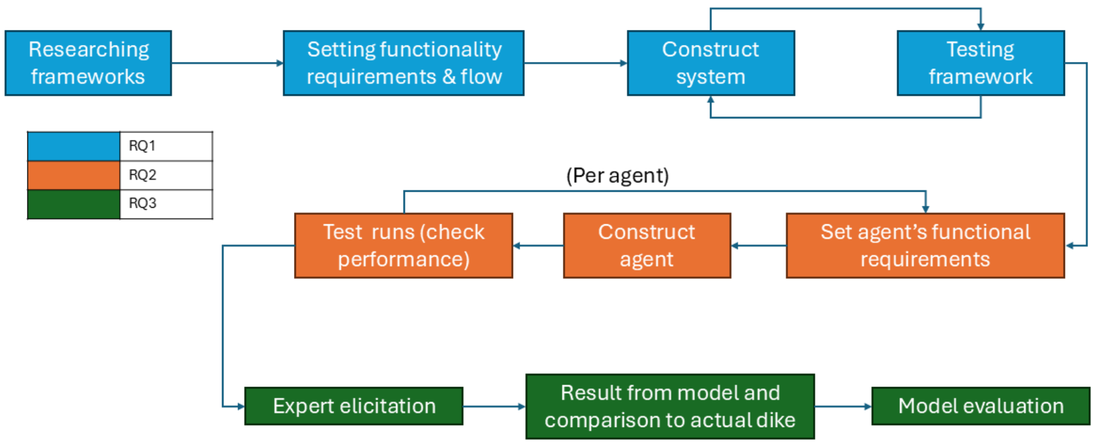

Figure 1: The framework of the methodology for constructing an agentic AI system.

2.1 System Architecture

To start, what is agentic AI? Agentic AI is a system consisting of specialist agents using LLMs that have access to special tools that ensure the agents can perform complex tasks. The collaboration of these agents should enable the system to perform several complex tasks and so imitate complex processes. Examples of applications can be found in healthcare and finance (Acharya, Kuppan, and Divya 2025) and show transformative potential in decision-making, automation and human-AI collaboration.

In the pursuit of developing an agentic framework, it is crucial to first comprehend the various existing frameworks and fundamental components that comprise agentic AI. This study undertakes a systematic analysis and evaluation of several frameworks to explore the possibilities within this domain. One notable framework analysed is LangGraph 2025, which leverages analysers to execute web research within an agentic paradigm. Similarly, the Mishra 2025 framework advances a booking system by deploying a supervisor agent.

For the purpose of this framework, LangGraph is utilised due to its straightforward graph structure, robustness as evidenced by Hosseini and Seilani (2025), and the abundance of practical examples. Moreover, the selection of this framework was advised by Koen de Jong 2025. From examining these frameworks, this research identifies a clear need for an agentic AI framework, which serves as the foundation for further development in this study. However, to effectively apply agentic AI to dike design, a thorough understanding of the dike design process is required.

To gain an understanding of the current process for dike design, colleagues from Witteveen+Bos were consulted, along with the dike design guidelines from Deltares and Rijkswaterstaat (van Bree et al. 2018; Rijkswaterstaat 2017). The dike design guidelines provide a clear overview of how dikes are designed and what factors need to be considered. Considering only macro-stability and stakeholder preferences, the simplified process is outlined below (Hannema 2025; Twijnstra 2025):

1. Project start
2. Stakeholder identification initiated
3. Write a participation plan
4. Set demands
5. Design loop
   (a) Design dike
   (b) Check the dike with demands
6. Design is finished

Using the knowledge of the agentic AI system and the dike design process, the demands for the overall agentic AI system can be established. These demands should encompass the system's general functional requirements. The demands are the following:

- The system can independently make decisions and design a dike, incorporating macro-stability, dike dimensions and accounting for stakeholder preferences.
- The system replicates the dike design process, including stakeholder assessment and repeated assessment.
- The system must enable precise and efficient communication protocols between agents, adopting standardised messaging formats to ensure interoperability and prevent data loss during the dike design process.
- The system must support iterative cycles, enabling continuous refinement and optimisation of design solutions through successive iterations based on feedback loops between agents.
- The system must be flexible in its inputs and parameter ranges while providing reliable outputs.
- The system should provide transparency in its internal flow and thought processes by displaying intermediate results and justifying its decisions.

The system can independently make decisions and design a dike, incorporating macro-stability, dike dimensions and accounting for stakeholder preferences. This demand covers the general expectation of the functionality of the system. The system can design a dike while accounting for macro-stability, providing dimensions of the dike and considering stakeholder preferences and demands; The output of the model should be a safe dike that incorporates stakeholders' preferences.

The system replicates the dike design process, including stakeholder assessment and repeated assessment. To accurately imitate the dike design process, the choice to replicate this process has evolved over time and seems to be the most effective because these are real-life established methods. This process was explained earlier in this chapter and includes a feedback loop that utilises trial and error to optimise the design in terms of material usage, space, and the inclusion of stakeholder preferences.

The system must enable precise and efficient communication protocols between agents, adopting standardised messaging formats to ensure interoperability and prevent data loss during the dike design process. This demand covers the general communication means of the system. The agent must be able to communicate using the system's communication means, which store intermediate results and relevant information for all agents, and prevent data loss during the design process.

The system is required to support iterative cycles, enabling continuous refinement and optimisation of design solutions through successive iterations based on feedback loops between agents. This means the system should be able to assess the design and provide feedback to itself to improve and optimise the dike design via feedback loops and the ability to incorporate this feedback in the design by adapting the dike dimensions while ensuring a reliable and safe design.

The system must be flexible in its inputs and parameter ranges while providing reliable outputs. One of the main advantages of applying agentic AI is its ability to interpret context and translate this to input parameters for the system. The system should be able to interpret the system input and translate it into design parameters.

The system should provide transparency in its internal flow and thought processes by showing intermediate results and accounting for its decisions. A main issue of AI applications is the lack of transparency in its internal processes and assumptions. This makes most AI applications a black box system, which makes their results difficult to validate. To prevent this as much as possible, the system must account for its steps and decisions as much as possible.

These general system demands can be divided and transformed into demands for the framework and the separate agents resulting from the general structure. These demands must encompass both functionality and transparency, as well as accountability. For the framework, the inter-agent communication, as well as system flow and LLM usage optimisation, are covered. This results in the following demands for the system's framework. These are general demands, but they should ensure optimal collaboration among the specialised agents resulting from the framework.

- The framework must replicate the simplified dike design process as stated earlier in System Architecture by imitating every step of this six-step process.
- The framework must provide an effective communication means, optimising inter-agent communication, and allow for data storage.
- The LLM usage of the system should have a set structure in which LLMs are utilised concerning inputs as well as outputs.
- The system's flow must be transparent and traceable.

The framework that meets these demands supports the use of individual agents as well as their collaboration in terms of communication and optimal LLM utilisation.

2.2 Agents

Based on the system architecture results, four agents are distinguished, each with its own set of background information and functional requirements. These agents are the stakeholder agent, the supervisor agent, the dike design agent and the demand checker agent. Additionally, the system's communication relies on the state, a JSON data structure that is passed from agent to agent while only being updated, which gives it the function of both a means of communication and a database.

2.2.1 Stakeholder agent

The first agent in the agentic AI system is the stakeholder agent. The goal of the stakeholder agent is to establish stakeholder preferences that should be incorporated into the design. This ensures a more optimal design than when the design only relies on macro-stability. Since dike design is only assessed in 2D for this application, stakeholder preferences are limited to 2D dike design in this study. Nevertheless, these stakeholder preferences can still be applied. For 2D design and only considering macro-stability as a failure mechanism, in this study, the following main standard preferences are assumed (Stelling 2025):

1. The dike is safe.
2. The dike is as small as possible.
3. The ditch behind the dike is as deep as possible.
4. The dike is as cheap as possible.

These are the main preferences that need to be integrated into the model and can be adopted as standard demands. However, every project has special stakeholders or circumstances that need to be taken into account, and therefore, static stakeholder demands are not a sufficient representation of stakeholder preferences in the model. In a typical project, experts from Witteveen+Bos follow a specific flow: first, stakeholders in the project are identified and ranked in terms of their importance, power, and interests. Based on these rankings, the stakeholders are classified by participation level. These levels are collaborate, involve, consult and inform. For these levels, Witteveen+Bos writes a participation plan that outlines various participation methods for each category.

Next to stakeholder participation, the agent should be able to consider and integrate stakeholder demands and wishes into the design. Input for these demands can be input in an Excel sheet in the code or via the system prompt. The system should be able to filter demands relevant to 2D dike design and transform them into SMART demands. Only without the time-bound aspect, because that is not relevant to this system, as time is not a factor in the system.

So this results in the following functional requirements for the agent:

- The agent should be able to identify relevant stakeholders for a scenario.
- The agent writes a participation plan for the involved stakeholders.
- The agent can identify relevant demands and wishes for the 2D dike design.
- The agent converts demands to SMART demands, without the time-bound aspect.

2.2.2 Supervisor agent

The supervisor agent directs the flow of the design loop of the dike. Its main function is to determine what the next step in the design process is and to evaluate the current state of the process. The supervisor must be able to decide on the most efficient flow of the system. The supervisor has three options to direct the flow to:

1. The dike design agent.
2. The demand checker agent.
3. The finish of the process.

The supervisor's task is to call either one of these options and provide a message to the agent or explain why the finish is called. This message provides instructions for the agent on what needs to be changed or specifically assessed. This led to the following functional requirements for the supervisor agent:

- The agent can independently direct the flow of the design loop.
- The agent can provide other agents with specifics of the task.
- The agent can assess when an optimal design is made.

2.2.3 Dike design agent

The dike design agent's purpose is to design a dike in terms of dimensions and assess its stability, while also incorporating feedback from the supervisor agent and the demand checker agent. The agent uses a macro-stability metamodel that takes dike dimensions as input and returns a Factor of Safety (FoS). For this research, the dike is considered safe at a minimal FoS of 1.3. Efficiently using this metamodel is the main task of the dike design agent. Additionally, the agent should be able to receive, interpret, and effectively implement feedback in the design.

To assess macro-stability, the model utilises a metamodel for macro-stability (Guichelaar 2024). This model is trained on data from 10,000 dikes from d-stability. The main theories behind the d-stability model are Bishop's Method and the Limit Equilibrium Methods. From the Bishop's method, the FoS is derived; the formula for the FoS is shown in Equation 1. This is the core theory behind the data on which the metamodel is trained.

$$\text{FoS} = \frac{\sum \text{resisting moments}}{\sum \text{driving moments}} \tag{1}$$

Some design factors are standardised, such as the outer slope or the inner berm slope. The standardised dimensions are shown in Table 1. These dimensions are common for existing dikes. By standardising these dimensions, the accuracy of the metamodel is increased (Hastie, Tibshirani, and Friedman 2009).

Table 1: Standard dimensions in the macro-stability metamodel (Guichelaar 2024).

Dimension	Value
Outer slope	1:3
Inner berm slope	1:20
Ditch slope	1:3
Bottom width ditch	0.5m
Thickness clay cover	1.5m

The dimensions on which the macro-stability metamodel is trained are formulated in Table 2 with the range of the dimensions and their units. Within these ranges, the metamodel can assess the dimensions and their effect on stability accurately (Guichelaar 2024). However, in some circumstances, one or more of these dimensions can be provided as input to the overall model, allowing the agent to consider some dimensions as preset. These dimensions are commonly the thickness of the cohesive layer (i.e., clay layer underneath the dike) or the dike height.

Table 2: Dimensions which the metamodel is trained for (Guichelaar 2024).

Dimension	Minimal value	Maximal value	Unit
Crest width	3	10	m
Inner slope	+/- 2.14	4	-
Dike height	3.5	6	m
Water height	3	5.5	m
Thickness cohesive layer	0	8	m
Inner berm length	0	15	m
Distance ditch-dike toe	0	5	m
Depth ditch	0.5	1.25	m
Dike core	clay	sand	-
Factor of Safety	+/- 0.77	+/- 2.68	-

To incorporate the model into the agent, the agent must adapt to the input dimensions of the metamodel, enabling the metamodel to evaluate different designs based on their stability scores. The agent knows what needs to be changed or improved based on a message from the supervisor agent, the demands key of the state or based on an insufficient stability score. When the agent considers the dike stable and sufficient, the agent should redirect the flow to the supervisor agent.

This leads to the following functional requirements for the dike design agent:

- The agent can assign the dike dimensions based on its input and provided parameter ranges.
- The agent uses the macro-stability metamodel to assess macro-stability.
- The agent can consider feedback and implement it in the design.
- The agent only considers stable dikes (i.e., FoS > 1.3) as acceptable.
- The agent can explain its decisions and is transparent in its logic and strategies.

2.2.4 Demand checker Agent

The demand checker agent is a critical feedback body of the model. The demand checker agent assesses the dike based on the previously set demands and wishes from the stakeholder agent. The demand checker agent assesses the extent to which each demand and wish is implemented in the model and whether this is sufficient to comply with the set of demands. The agent should specify what needs to be improved to meet the demand. It should also verify the implementation of the wishes in the design and estimate the relative costs of the design.

The demand checker agent has the following functional requirements:

- The agent should be able to interpret the dike design.
- The agent should be able to assess the implementation of demands and wishes in the design.
- The agent should be able to provide feedback on the design.
- The agent should estimate the costs of the design.

2.3 Model evaluation

Model evaluation in AI and proving the results of AI models can be challenging. AI models are often black-box models that lack transparency and usually lack hard, proven theory or objective right or wrong answers. To undertake this evaluation, the framework presented in Chapter 6 of Covello and Merkhoger (1993) is utilised, encompassing six key criteria: logical soundness, completeness, accuracy, acceptability, practicality, and effectiveness. To also evaluate the adaptability of the model, the flexibility aspect is added to the evaluation. These aspects of the review will be explained after the evaluation methods are described. These include an expert elicitation, analyses of model results, and a comparison to the design of an actual dike, to assess the design of the agentic AI model against the results of current design methods.

The expert elicitations, conducted with experts from Witteveen+Bos within the fields of geotechnics, environmental management, and AI, were used, as well as an associate professor in hydraulic engineering from the University of Twente. The expert elicitation consists of a general presentation that explains the general model, its structures and its agents. The general presentation focuses on logical flow and outputs. The model's functionality is demonstrated through a few pre-run examples, after which experts can try the model for themselves by providing input. After this, the experts received an evaluation form to assess the model in five different categories: flexibility, accuracy, logic, efficiency, and expected added value.

To analyse the results of the model, the logic of the flow is examined to check the consistency and logic of the intermediate steps taken by the agents, which also concerns the transparency of these steps and the completeness of the information included in them. The final results of the model are important to analyse for accuracy. This is measured by: (1) the inclusion of stakeholder preferences and demands, and (2) optimisation of the design; the resulting dike is considered optimised when a FoS between 1.3 and 1.35 is reached. For this analysis, three scenarios are considered. One random scenario, one extreme case that is suboptimal for stability, featuring a maximum cohesive layer, a high dike, and a sand core. And one situation involves a complicated stakeholder issue due to conflicting interests.

To verify whether the system produces similar results to the design of an actual dike, the system receives an input scenario that corresponds to an actual dike designed by Witteveen+Bos, specifically the Grebbedijk (located near Wageningen) on the Nederrijn. For the comparison between the actual design and the result of the model, the similarities and differences will be analysed. There is no single correct or incorrect answer regarding dike design, as it is not objectively determined. However, since the actual dike is designed based on established methods, similarities are assumed to be good, and differences are assumed to be deviations.

Logical soundness requires that the AI model operates with internal consistency and clear reasoning. Each component of the system must use logical or established processes and methodologies that are transparent and easily understood by users and stakeholders. This involves clearly articulating how each agent within the AI system makes decisions and processes information, ensuring that its reasoning and decision pathways are traceable and rooted in coherent logic. Transparency is further enhanced by making assumptions and decision-making processes explicit, allowing stakeholders to understand how the model integrates various inputs and arrives at specific outcomes, thereby fostering trust and confidence in the system. To evaluate this aspect, both the results of the expert elicitation and the results of the model are used.

Completeness, within the context of Covello and Merkhofer's risk assessment framework, involves thoroughly addressing both technical and social aspects, as these are essential for a detailed and meticulous dike design. It entails identifying the appropriate parameters and variables to ensure that all significant factors impacting the design process are taken into account. For the evaluation of this aspect, the model's results for regular runs and the actual dike are used.

Accuracy in this context requires the model outputs to reliably reflect the integration of stakeholder preferences and the optimisation of the factor of safety, which should be above but as close as possible to 1.3. To evaluate accuracy, test runs and comparisons to the actual dike are used; primarily, optimisation and stakeholder integration are employed to assess accuracy.

Acceptability is gauged through stakeholder and expert evaluation, focusing on trust in the model and transparency of its operations, assumptions, and limitations. For the acceptability assessment, the results of the expert elicitation are used.

Practicality is assessed by the model's ease of use and integration into existing workflows, supported by user-friendly interfaces and efficient data handling capabilities. The judgment on the practicality of the model is based on expert elicitation from the efficiency and quality categories, together with a comparison of using the metamodel separately or within this system.

Effectiveness considers the model's contribution to informed decision-making in dike design, particularly its ability to balance technological requirements with stakeholder integration and preferences. This also involves evaluating the model's role in facilitating adaptive and iterative design improvements. Effectiveness is evaluated as a category of the expert elicitation, and based on this, effectiveness is assessed.

Flexibility in an agentic AI model is crucial for adapting to changing scenarios and incorporating new data with minimal effort. A flexible model integrates updates and adjusts seamlessly, ensuring consistent performance across different contexts. Evaluating flexibility involves assessing the model's adaptability and identifying areas for improvement to maintain robust functionality. High flexibility enhances the model's reliability and broad applicability, which are crucial for dynamic applications such as dike design. This criterion underscores the model's capability to meet evolving demands efficiently. Assessing flexibility is based on the expert elicitation, the result of the model for both normal flow and the actual dike.

Applying this comprehensive framework enables a structured and critical evaluation of the AI model, facilitating its refinement and ensuring alignment with professional and community expectations. Detailed results from these criteria will be discussed separately to illustrate the model's capacity against each evaluation metric. The evaluation rubric can be found in Appendix C. To effectively evaluate model performance and identify trustworthy experts, expert elicitations are conducted. This provides expert opinions on the agentic AI model.

3 Results

The results contain the production of the previously described methods. The first section outlines the structure and functions of the overall framework of the system. This includes the agents in the system and their flow, the inter-agent communication, the system in- and outputs and prompting. In the methods demands for the system were set, whether these demands comply is explained separately.

The second section of the results covers the structures and the theories behind the agents. Every agent has its own flow and functions that are explained. Just as for the overall system, the set functional requirements for every agent are addressed.

In the third section the results of the model evaluation are elaborated on. This includes the Covello & Merkhofer risk assessment frameworks results as well as the processed results of the expert elicitation and results from the model.

3.1 Agentic Framework

3.1.1 Structure

The goal of the agentic framework is to create a system in which agents can communicate with each other and are directed to each other to optimise the result of the task based on the system's input. This section is relevant for Research Question 1, "What is an efficient system architecture for agentic AI in dike design?". All applicable elements will be explained in this section, along with the system's flow and the demands it must comply with from the methodology.

The system consists of four agents that communicate via the state. An overview of the framework can be found in Figure 2. Figure 2 shows the structure and general information flow of the system. Essentially, the system functions as follows: the system is initiated with a prompt from the user that provides the system with a dike design scenario. The stakeholder agent then translates this scenario into SMART demands on the dike, which are inserted into the state. The state is passed on to the supervisor agent, and the supervisor initiates a design loop between the dike design agent and the demand check agent. When a dike is designed, the demand checker agent assesses the dike based on the set demands and wishes and provides feedback on the design. The supervisor agent determines when the demands and wishes are sufficiently incorporated into the design. When not adequately implemented, the design loop will start over; when sufficiently implemented, the system reaches its end, and the final design is the output.

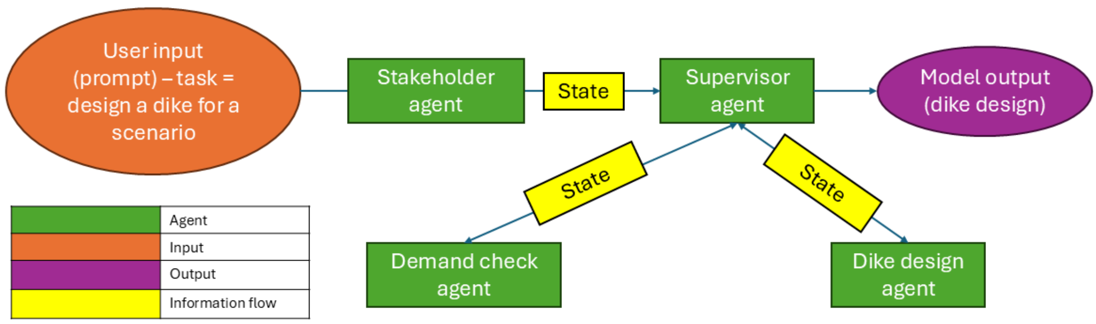

Figure 2: Conceptual framework of the agentic architecture.

The structure comprises the general communication means of the system, as well as how the system directs the flow, and how the prompts and output of the agents are typically structured.

The communication means of the system is the state. The state is a data structure with defined keys that consist of a set type of information. In this framework, the state has eight data keys. These are the following:

1. next: the next agent in the system. Defined as a string, which is a datatype that contains textual information.
2. messages: all inter-agent messages. Defined as a list of messages, defined as strings that contain textual information.
3. demands: the demands that are set for the dike. Defined as a list of SMART demands (defined in Stakeholder agent).
4. wishes: preferences for the dike that are not required. Defined as a list of strings that contain textual information.
5. stakeholder analysis: the performed stakeholder integration plan. Defined as a list of strings that contain textual information.
6. demand check: the results of the demand checker agent. Defined as a list of demand checks (defined in Demand checker agent).
7. dike dimensions: a database of the dike dimensions. Defined as a list of dike dimensions data structure (defined in Dike design agent).
8. dike agent logic: the logic behind the decisions of the dike dimension agent. Defined as a list of strings that contain textual information.

The state keys are often structured as a list, which makes the state a database. When agents are called, the state is the input of the agent. When specific keys are required for an agent, they can be entered in the prompts. This is the means of communication. The state and the definitions of the state keys can be found in Appendix B. The system routing is performed using the LangGraph graph structure, which defines nodes (in this system, the agents) and edges (connections between nodes). LangGraph automatically passes the state to the next node in the system. For routing in the supervisor node, the agent can decide which agent to use for itself. The agent utilises LangGraph's Command function to redirect the system's flow to the agent, as this design loop lacks a predetermined structure.

The agents' inputs are prompts. Prompts are texts that contain tasks and the context of those tasks, using specific state keys and data entries from the state. Since prompts can be very sensitive as pointed out in Salinas and Morstatter (2024). The prompts are structured as markdown prompts. Markdown prompts utilise markdown syntax to effectively organise and enhance the presentation of textual information in a clear and readable manner. This lightweight markup language enables the incorporation of headers, lists, and emphasis, thereby improving the readability and comprehension of complex instructions or tasks. Commonly used in documentation and AI interactions, Markdown enables users to structure content systematically, facilitating better engagement and understanding. By using elements like bold and italics for emphasis, Markdown prompts help communicate tasks and instructions efficiently within various computational contexts.

The output of the agent is similar to the state structure. The agent's output has defined keys with a predefined response structure. This way, the output becomes predictable and can be programmed with. The state is updated based on certain output keys. The output is structured via the with_structured_output function, which allows for a predefined output in a TypedDict structure. It is essential to have a strict output which ensures all output keys are used.

3.1.2 Functional requirements

This section verifies whether the demands outlined in subsection 2.1 are met and whether the framework provides a sufficient foundation for the system. The first demand:

The framework must replicate the simplified dike design process as stated earlier in System Architecture by imitating every step of this six-step process.. The dike design process, as described in subsection 2.1, consists of 6 steps. Which do comply with the framework in Figure 2. The project start is the input scenario for the system (step 1). The stakeholder identification (step 2), the participation plan (step 3), and the demands (step 4) are covered by the stakeholder agent and the supervisor agent. The dike design agent and the demand checker agent perform the design loop (step 5). The final step is the presentation of the design (step 6). Since all steps are integrated, the system complies with the demand.

The framework must provide an effective communication means, optimising inter-agent communication, and allow for data storage. The system utilises a state structure with specific entry keys that ensure that the correct information is stored in the right place. The state is passed from agent to agent, allowing agents to extract relevant information from it. The state is an effective means of communication and stores all relevant intermediate results.

The system has a logical information flow and logical actions. When the system test runs, the flow seems logical, the set first steps are performed, and the supervisor alternates between the dike design agent and the demand checker, often in a logical sense; however, since the LLM can decide on the flow, this is not a consistent flow. In general, the system's flow and logic are sufficient, although not entirely consistent.

The LLM usage of the system should have a set structure in which LLMs are utilised concerning inputs as well as outputs. The system has a prescribed LLM usage utilising markdown structure for prompting, and for every LLM, there is a structure defined with keys the LLM has to fill in. This leads to effective LLM utilisation.

The flow of the system must be transparent and traceable. To ensure transparency, the LLMs explain their decisions. Additionally, to visualise the flow of the system, the name of the agent is printed for every agent that is called.

3.2 Agents

3.2.1 Stakeholder agent

The stakeholder agent consists of two LLMs and a few supporting functions for visualisation and hard-coding to prevent inconsistency of the LLM. The structure of the agent is shown in Figure 3. As can be seen, the agent's input is the scenario, which is also the input of the overall system. The scenario is inserted in the prompt of the first LLM. The stakeholder identifier is the LLM that identifies at least 20 stakeholders for the project, which is based on the reasoning of the LLM itself and an example from an actual project (Twijnstra 2025). Then all these stakeholders get assigned a power and an interest score. Importance is classified as follows:

- 0-4: low power or interest.
- 4-7.5: medium power or interest.
- 7.5-10: high power or interest.

The input prompt of the agent consists of the scenario from the user and information on the power-interest classification and participation categories. The prompt of the LLM can be found in Appendix A1. The identified stakeholders, combined with the power and interest scores, are the output of the LLM. Based on these power and interest scores, a participation category is assigned to the stakeholders. These are the following:

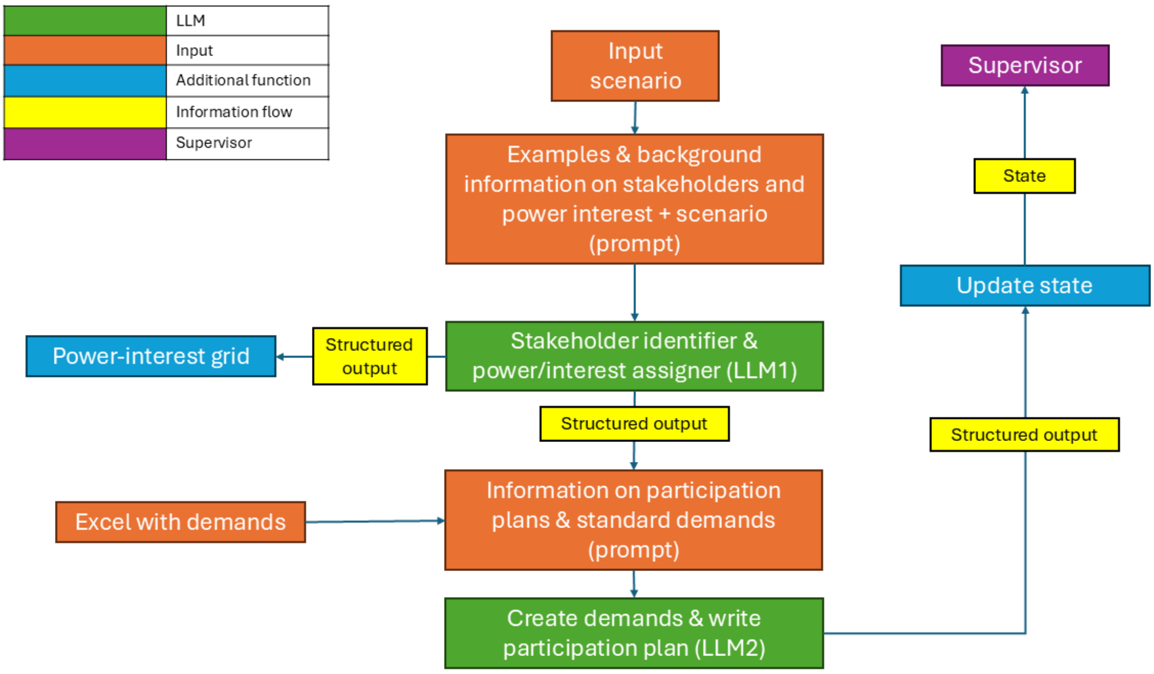

Figure 3: The structure of the stakeholder agent.

- Inform: low/low, medium/low, low/medium.
- Consult: medium/medium, high/low, low/high.
- Involve: high/medium, medium/high.
- Collaborate: high/high.

Next to the division of participation categories, the agent also plots a power-interest grid based on the output of the LLM. An example of the power-interest grid is shown in Figure 4.

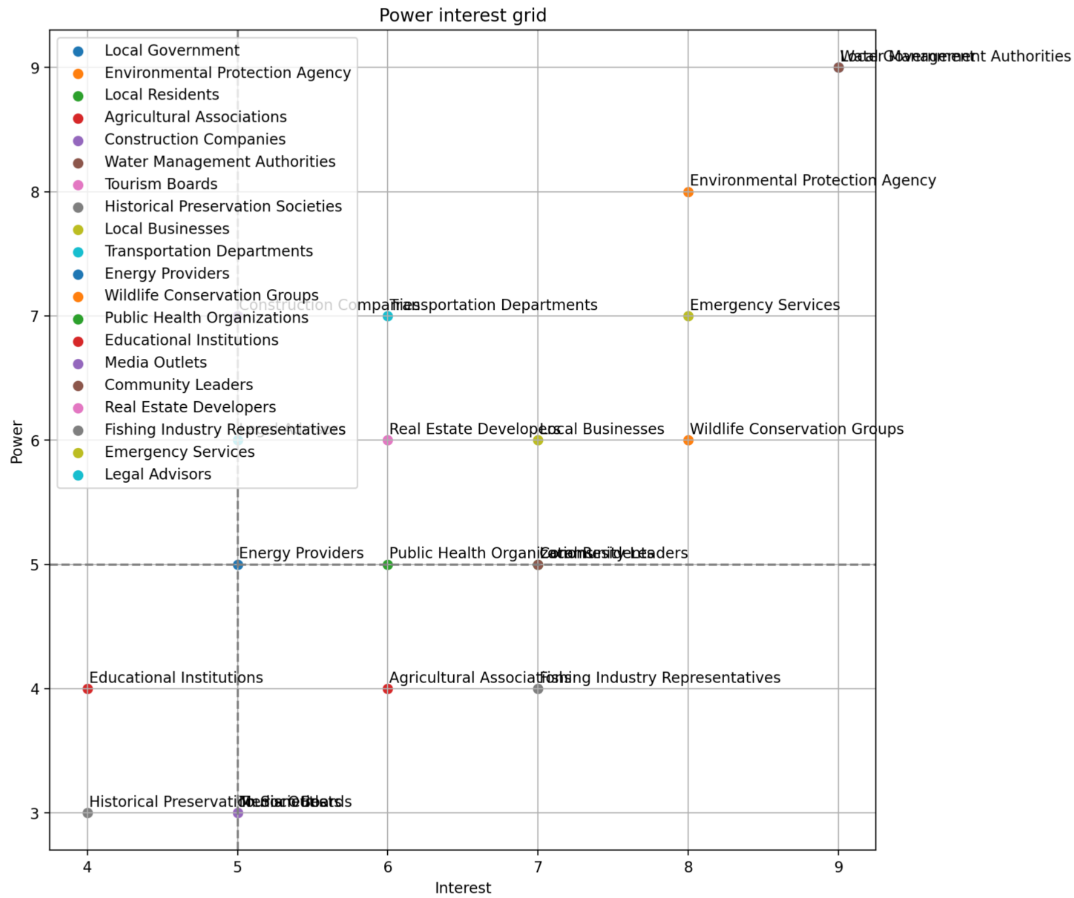

Figure 4: Example of a power-interest grid generated by the stakeholder agent.

The second LLM of the agent uses the stakeholders and participation categories to write a participation plan. This is based on background information on participation plans and the LLM's processes; the LLM assigns an approach to involve stakeholders in every category. Additionally, the agent can receive an Excel file as input and assign SMART demands and wishes to the dike design. The time-bound aspect is omitted because it is irrelevant for this model, and only demands and wishes for 2d design are included, since only these can be considered. These demands can be based on the optional Excel file, as well as the general preferences of stakeholders. These wishes are included in the prompt. The prompt can be found in Appendix A2. The output of the LLM is the participation plan, a list of SMART demands, and a list of wishes. The wishes also include optimisation factors such as minimal dike width and height, or maximising ditch depth. To check whether the agent complies with the demands set in Chapter 2, every demand is elaborated on in the rest of the section.

The agent should be able to identify relevant stakeholders for a scenario. The agent can identify stakeholders based on the examples from the prompt and its trained data. The agent identifies 20 agents ranked by power and interest to create a power-interest grid.

The agent writes a participation plan for the involved stakeholders. Based on the identified stakeholders and their power and interest scores and context on participation levels, the agent writes a participation plan integrating methods to involve the identified stakeholders in varying manners.

The agent can identify relevant demands and wishes for the 2D dike design. The input of the agent is a scenario including special stakeholder needs and an optional Excel file with general wishes. The agent checks whether these demands are relevant for 2D dike design and transforms them into concrete demands and wishes in relation to the design parameters.

The agent converts demands to SMART demands, without the time-bound aspect. The agent uses a structured output, and the demand output is structured as a list of smart demands, in which the LLM needs to specify every element of smart and write a general smart demand. Only time-bound is excluded because it is irrelevant for the model; it continues until an optimal design is reached.

3.2.2 Supervisor agent

The supervisor agent's primary goal is to direct the flow of the design loop within the system. The supervisor agent can call the dike design agent, the demand checker agent and the finish of the process. The supervisor uses a router and the command function from LangGraph to enforce this decision process. The supervisor agent uses a current situation-desired situation structure to evaluate the state of the design and incorporates feedback from the demand checker agent.

The input to the supervisor LLM is a prompt (found in Appendix A3) that consists of its primary task (i.e., determining the system's flow), an example of the dike design process, and the task context. This task context is the current design, logic from the dike design agent, system messages, the demands and wishes, and the output of the demand checker agent. The output of the agent is structured as a "next" key, which determines the point in the system to which the state is directed next, an explanation of its decision, and a list of feedback per aspect of the dike. Feedback is defined by two key elements: a current situation and a desired situation. The output is used to both track the model and provide the other agents with information on what needs to be assessed or improved. The following paragraph assesses the functionality of the supervisor agent, which should comply with the functional requirements outlined in the methodology.

The agent can independently direct the flow of the design loop. The agent can direct the flow of the system without the user interfering. This means the agent can direct the flow independently.

The agent can provide other agents with specifics of the task. The agent provides feedback consisting of a current situation and a desired situation that the agents should try to reach. This provides the agent with a clear understanding of what needs to be done.

The agent can assess when an optimal design is made. The agent can independently determine when it considers a design optimal and let the model go to the end of the system. However, whether this design is optimal will be discussed in the model evaluation (subsection 3.3) of this thesis.

3.2.3 Dike design agent

The dike design agent's goal is to design a dike, assess its macro-stability, and process feedback provided on the design. The agent should decide for itself when a dike is considered "acceptable" to return to the supervisor agent; the agent cannot accept the old design when a new agent call is made. The agent's workflow is as follows: first, it uses an LLM to create a new design. The design is, by definition, not yet acceptable for this round. The agent chooses dimensions for the dike and provides some explanation on its choices. Then the dike is assessed by the metamodel on macro-stability. When a FoS of 1.3 or higher is returned, the dike is considered safe. The LLM reassesses the dike, considering its FoS and dimensions in relation to the demands and wishes. When the LLM determines that every element is sufficient, it returns the state to the supervisor. The flow of the agent is shown in Figure 5.

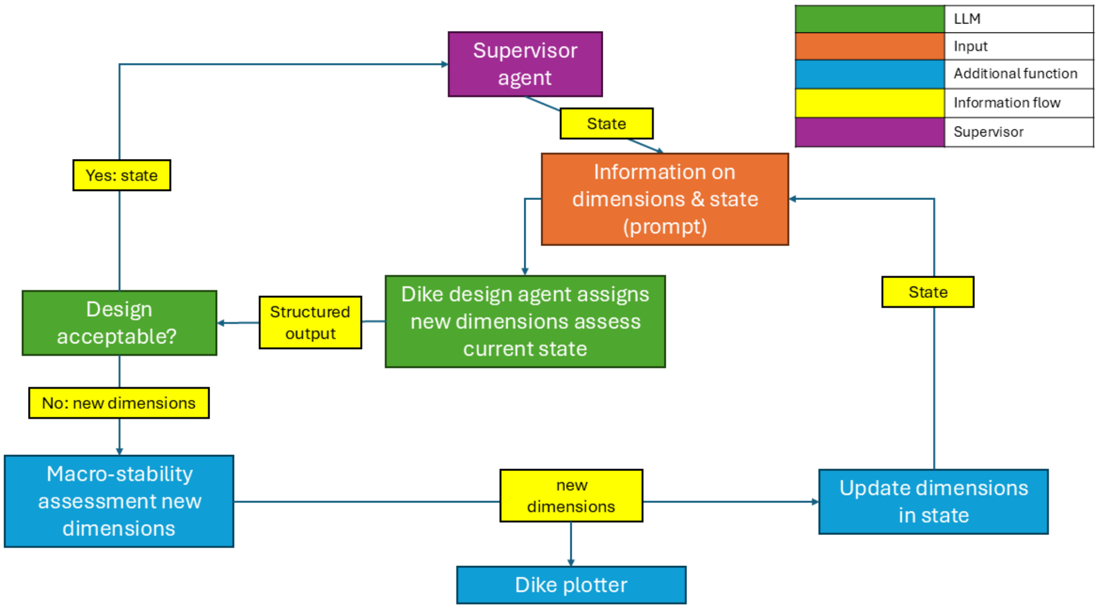

Figure 5: The flow of the dike design agent

The input to the dike design LLM includes the context of the macro-stability metamodel and its parameter ranges, an example design process, a history of previously tried dimensions, and the logic behind the most recent dimensions, as well as outputs from the supervisor and the demand checker agent to provide feedback on the design. The output of the LLM is structured with a key for every dimension that can be altered; these are all dimensions except for water height, which is set at 4.5 meters. This is a dimension the model can't decide on itself, and therefore it is standardised. The dimensions are also transformed into a sketch of the dike; an example of such a sketch is shown in Figure 6. Another output key is "acceptable?", a key with "acceptable" or "retry" as options, indicating whether a new dike needs to be assessed, also provides a reason for the LLM's decision. An additional support for the output is the task for the model to devise a strategy to improve the design before it is implemented, following the LLM Chain of Thought (CoT) theory, which can enhance LLM reasoning and performance (Zhang et al. 2025; Sun, Pan, and Yan 2025). The strategy is structured as follows: first, the model determines its objectives and goals, then it assesses the current condition of the design and proposes actions and methods to achieve the set objectives and goals. The LLM also needs to predict the outcomes of its proposed actions so that it can assess the effectiveness of these actions. When the design is considered acceptable, the newly proposed design is discarded, and the state, including the dimensions of all assessed dikes and the logic behind the agent's decisions, is passed on to the supervisor.

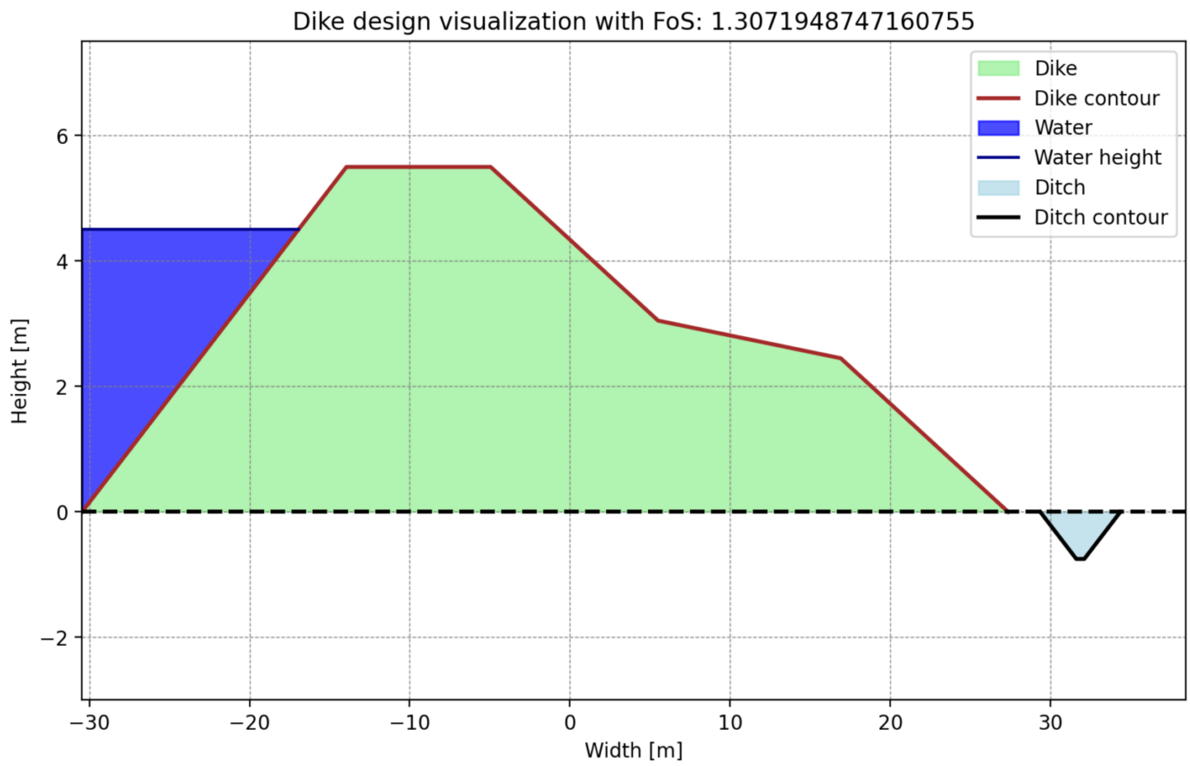

Figure 6: Example of dike sketch

In the methodology, requirements for the agent were set. To verify whether these are integrated into the agent, every demand is reviewed with an explanation of why it is implemented.

The agent can assign dike dimensions. The agent can decide on dimensions to try to implement in the design, except for the water height, which is set at 4,5 meters. The other dimensions the agent can decide on are combined with the input from the different agents. The dimensions are also stored in the state to prevent the agent from looping by using the same dimensions repeatedly. The dimensions assignment functions as expected, and combined with the design strategy, the efficiency of the process is greatly improved. The quality of the results is evaluated in the model evaluation (subsection 3.3).

The agent uses the macro-stability metamodel to assess macro-stability. The output dimensions of the LLM are passed on to the metamodel when the previous design is not considered acceptable. The new dimensions, combined with the set water height of 4.5 meters, are the input for the metamodel. This functions as expected and always returns a FoS.

The agent can consider feedback and implement it in the design. The feedback from both the supervisor agent and the demand checker agent is implemented in the prompt, and the agent can access this feedback. This seems to work as expected, but the quality is evaluated in the model evaluation (subsection 3.3).

The agent only considers stable dikes (i.e., FoS $> 1,3$) as acceptable. The model sometimes has trouble interpreting the FoS correctly and determining whether it is sufficient. To ensure no unstable dikes are implemented, a hard code is implemented to ensure that when the FoS is smaller than 1.3, a new dike is designed; this is also applied to high FoSs. Bigger than 1.5 is retried to prevent unnecessary LLM calls, as the design can logically be improved from 1.5 closer to an optimal design. While not entirely relying on the LLM, the model is not accepted when the FoS is not sufficient.

The agent can account for its decisions. The agent uses a structured strategy to evaluate the dike and strategise the improvement of the dike design, which improves the design process. Next to that, the agent provides a reason for its decisions. This gives a clear overview of the improvements the LLM does and the decisions it makes.

3.2.4 Demand checker agent

The demand checker agent's goal is to verify whether the demands and wishes are integrated into the design. The demand checker must check per demand whether it is included in the design and what needs to be improved. Additionally, the demand checker agent estimates the relative costs of the dike based on cross-sectional area and material costs. For material costs, the same assumed costs are used as in Guichelaar 2024. Lastly, the agent calculates the total width of the dike, including the ditch. The flow of the agent is shown in Figure 7. As can be seen, the costs and design width are first calculated, after which the design is fed back and the output is implemented in the state.

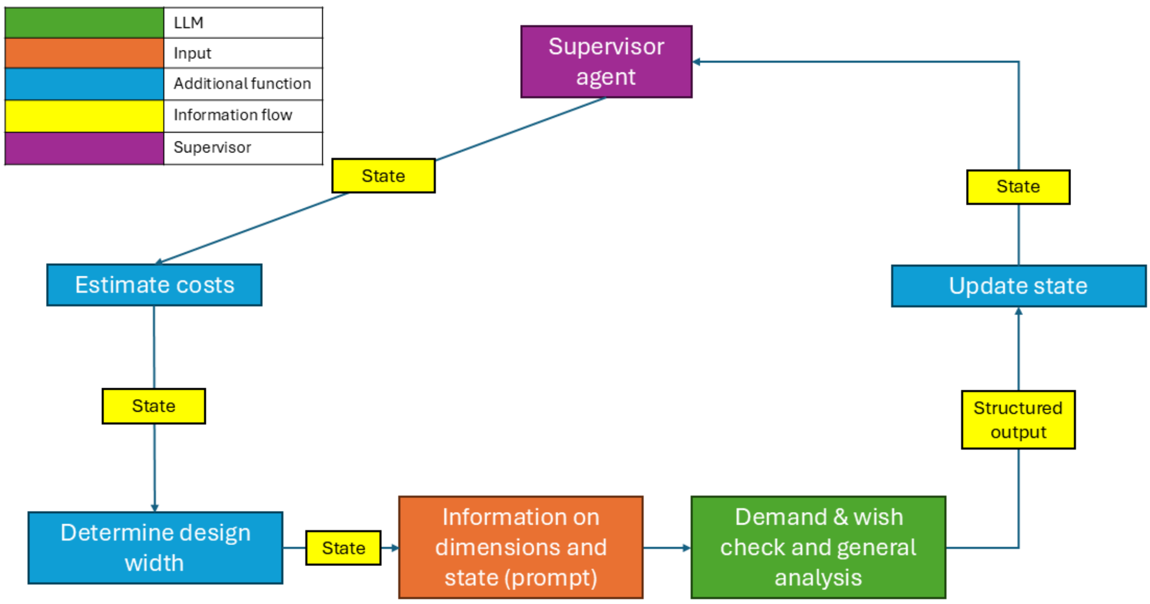

Figure 7: The flow of the demand checker agent

The input of the agent is the latest dike design, and the dike design parameter ranges within the dike can be assessed by the metamodel. Next to the design, the agent receives a task from the supervisor, along with its former actions and the demands and wishes of the state, to be evaluated. The prompt can be found in Appendix A5.

The output of the agent is a list of checked demands, structured as follows: 1. the demand or wish, and 2. whether it is included. 3. Desired improvements. Per demand, the agent will check its implementation. Additionally, the agent will provide a general analysis of the design. This general analysis is easier for users to understand, as it clarifies the model's flow.

The final part of this section focuses on explaining the agent's compliance with the demands set out in the methodology.

The agent should be able to interpret the dike design. Because the agent has information on the design parameters and their ranges, the agent can sufficiently interpret the design to provide feedback on the design based on stakeholder preferences.

The agent should be able to assess the implementation of demands and wishes in the design. The demand check agent receives the demands, wishes and design from the state in the prompt. So the agent should be able to check the implementation.

The agent should be able to provide feedback on the design. The agent offers feedback on the specific demands and a general analysis of the design. The result is effective feedback on the design.

The agent should estimate the costs of the design. The agent uses a cross-section area*material costs approach. This is only a relative cost indication, meaning that when the cross-sectional area increases, the costs increase. Additionally, sand is less expensive than clay.

3.3 Model evaluation

This section outlines the theories behind the model, its performance, and applicability. For each of the seven aspects stated in the methodology, a paragraph will explain why it received the score of the rubric in Appendix C. For evaluation, the model assessed dikes in several scenarios, which are presented in Appendix D. An expert elicitation was also conducted, including a survey, and the results are presented in Appendix E.

Additionally, the model was tasked with designing a plan for the Grebbedijk area near Wageningen. Some data was available on the Grebbedijk that was inserted into the model. Table 3 shows a comparison of the design from the model to the actual Grebbedijk. Based on the dimensions, the metamodel also assessed the Grebbedijk. As can be seen, the score of the agentic AI design is close to its minimal requirement of 1.3, while the scores from the actual Grebbedijk are higher than the high end of the model. However, the stakeholder assessments and calculations done for this dike are much more extensive than the agentic AI model is capable of, so this does not mean the agentic AI design is better.

Parameter	Agentic AI design	Actual Grebbedijk	Unit
Dike Height	5.7	5.7	meter
Crest Width	4	4	meter
Inner Slope	1:3	1:3 + 1:5	-
Inner Berm Length	2	5	meter
Thickness Cohesive Layer	0	0	meter
Distance Toe to Ditch	2	-	meter
Ditch Depth	1.25	-	meter
Dike Core	clay	clay	-
FoS	1.3143	1.5523	-
Water Height	4.5	-	meter

Table 3: Comparison of design dimensions of the Grebbedijk.

For the logical soundness of the model, consistency and transparency in logic are primarily considered, along with the LLM decision logic in the intermediate steps of the model. The main result of the model is based on the macro stability metamodel with an accuracy of approximately 1% (Guichelaar 2024), which is derived from Bishop's method and the limit equilibrium method. These are established methods for dike design. For stakeholder integration, SMART demands are utilised, along with the main methodology for stakeholder integration from Witteveen+Bos, which serves as a framework for the stakeholder agent (Hannema 2025). These are commonly used practices and provide good stakeholder integration in the design process for such projects. For transparency of the model, the results of the design loop are presented, with a structure that allows the supervisor to assess the current situation and specify the desired situation. Additionally, it justifies its decisions. The dike design agent must also justify its decision to retry or accept the design and outline a strategy to enhance it. When combining these, the agent offers insight into its internal processes and accounts for its decisions to the system's user. The demand checker agent also provides a general analysis of its assessment of the model and identifies areas that need improvement. It also checks per demand and wish whether it is included and what needs to be improved. These intermediate results provide some insight into the internal processes of the LLMs. The model's consistency is doubtful. The results always fall within the considered "acceptable" range (1.3-1.5) of FoS and essentially incorporate stakeholder preferences as much as possible, while ensuring a safe dike with the given input properties, which are transformed into demands. However, the consistency system is mainly dependent on LLMs, which are not always as reliable as desired. For logical soundness, the model scores 2/5 because it is neither entirely transparent nor entirely consistent.

The completeness of the model within its scope is considered in terms of stakeholder integration and dike design, based on a macro-stability assessment. Within these aspects, the model imitates the general design process for a simple dike; it identifies and ranks stakeholders, writes a participation plan, sets demands from the input, and assesses the dike on macro-stability and demand & wishes integration in the design while also using a cost indication and a width calculation. However, since the costs are only an indication, this could be further integrated to provide the actual costs of the dike. For completeness, the model scores 4 out of 5.

The accuracy of the model in terms of Factor of Safety is high, with a maximal deviation of 1% in the metamodel's assessment. The accuracy of the inclusion of demands is quite consistent and functions well; however, the model occasionally fails to interpret numbers accurately. The final result of the model yields an accurate Factor of Safety and consistently provides scores within the range of 1.3-1.5, with most scores falling within the 1.3-1.35 range. Therefore, the accuracy of the model is considered 3/5.

Acceptability: The results from the expert elicitation are important here. Seven experts who did the Microsoft Forms rated the recommendations of the model as follows: This results in an

Table 4: Survey Results from the Expert Elicitation (Appendix D)

Score	**Frequency**
8	1
7	1
6	4
3	1

average score of six, as three reasons were given regarding the unclear theoretical background of the meta-model for the user. The three was due to a lack of transparency in the metamodel, which is outside of the scope of the project and due to a lack of explanation on the metamodel to the expert. With an average score of six from the evaluation session, the model's acceptability is rated as 3/5 due to its intermediate explanations.

For practicality, the model scores well since the input does not need to be filtered, and the LLMs within the model perform a significant portion of the design process independently. There is no required input information for the model to design a dike, and the model returns a stakeholder analysis, dike dimensions, and a sketch of the dike. Based on the results from the forms, for practicality, the model scores 5/5.

For effectiveness, the results from the expert elicitation are essential for the two questions regarding effectiveness. For the first question, "To what extent do you think the agentic AI model contributes to accelerating the dike design process?" the average score for this question is 9, with no significant outliers. The second question, "How efficient is the use of resources (such as time and data) in the implementation of the agentic AI model?" scores an average of 7.9, with one outlier, and is rated as four out of ten due to sustainability considerations, primarily because of the high energy costs associated with LLMs.

For the two questions on flexibility from the expert elicitation. The first question, "How would you assess the flexibility of the agentic AI model to adapt to different design requirements and environmental factors?", scores an average of 7.3, ranging from a 6 to an 8. The second question is: "To what extent can the model easily process changes in the input parameters without significantly affecting the integrity of the output?" Also, scores a 7.3 with scores ranging from a 7 to an 8. That averages towards a 4/5. That gives an overall score of 26 out of 35, which translates to a score from 1 to 10 as 7.4 ($26/35 \times 10$).

4 Discussion

While this system is working quite well, it is crucial to consider its weaknesses and underlying assumptions. A model is an approximation of a real-life situation because the real-life situation is too complex to be modelled accurately, and this is also true for this model. In this section, the assumptions behind the model, as well as its strengths and weaknesses, are explained.

The performance of the model strongly depends on the performance of the underlying LLM. When examining the system's results, a lack of consistency is observed. This also depends on LLM settings, such as temperature (i.e., output randomness) and top-p (i.e., range of options considered). This could influence the model's performance results, especially in larger models that incorporate other failure mechanisms. Still, when the model receives the same input twice, it does not produce the same results.

The model only assesses macro-stability as a failure mechanism. This is only one of the failure mechanisms for dikes. For this model, for example, overtopping, piping, and erosion (Rijkswaterstaat 2017) of the outer slope are neglected. This limits the assessment of the dike that the model produces. This dike would not be reliable in real life because it has not been fully assessed.

One of the main assumptions in the model is that when the FoS is over 1.3, the dike can be considered stable. This simplification does not accurately represent the real-life situation; in reality, these calculations are more complex. However, these kinds of metamodels are reliable and function well within agentic AI systems. When extended to other failure mechanisms, this approach is considered to work efficiently and accurately, and is expected to enhance the design process's efficiency.

Furthermore, the system does not consider external factors. Spatial factors, such as trees or buildings, are not considered; the dike can only be as small as possible, but other space-limited solutions, like sheet piles, are not an option. External loads on the dike are also not considered; these external loads refer to activities such as building roads or houses on the dike.

Next to the LLM, the model relies on the macro-stability metamodel for macro-stability assessment. The application of this metamodel is limited to the water surrounding the dike and the sand or clay within it. The metamodel does not consider spatial factors, such as roads, houses around the dike, and other solutions, including sheet piles. This makes the model only applicable for simple situations where no extra load, limiting spatial factors or other solutions are required.

As with many AI applications, transparency is an issue. The model prints as many intermediate results as possible, but cannot account for every single one of its internal processes. This decreases trust in the results and complicates proving their validity, as there is little hard evidence other than the accuracy of the underlying model. Since transparency is an issue, it is challenging to rely on the accuracy of the results and the justification of internal steps and outcomes. Currently, the trust in AI is limited, but with the advancement of AI systems and methods, this will improve the results and eventually the trust in such systems.

To summarise, while the system does function, it still has its limitations. The model covers only a part of the dike design, and some assumptions regarding stability are made. The model relies on other models, such as a metamodel and an LLM, which both have a significant impact on the results from the framework. However, this proof-of-concept approach for this model demonstrates that agentic AI is capable of tackling complex problems within dike design and civil engineering by integrating technical analysis through tools, as well as incorporating social aspects into the design.

5 Conclusion

This research aimed to explore the added value of agentic AI in automating and optimising dike design regarding efficiency, accuracy, and flexibility. To achieve this, three specific research questions were addressed, each contributing distinctly to the overarching theme.

Firstly, the investigation into an efficient system architecture for agentic AI in dike design revealed that the optimal setup involves clearly defined agent roles and structured interactions. By establishing a reliable means of communication and a clear structure, the framework functions as intended and ensures inter-agent collaboration.

Secondly, the assessment of necessary agents for stakeholder and macro-stability highlighted the critical role of integrating tools like SMART demands and existing methods. Such integration ensured robust stakeholder engagement and a macro-stability metamodel, enabling the system to efficiently account for diverse inputs and scenarios. This agentic approach effectively balances technical accuracy with social considerations, confirming its potential for comprehensive design solutions.

Lastly, the evaluation of the advantages and disadvantages of agentic AI in dike design showed that flexibility was high, enabling swift adaptation to changing conditions and preferences. While other models may struggle with iterative feedback loops and stakeholder involvement, the AI model's architecture facilitated transparent and accountable decision-making. However, challenges such as achieving complete consistency and internal transparency across all scenarios remain, with some results varying in their ability to achieve stakeholder alignment.

In summary, agentic AI proves to be a valuable addition to dike design by significantly enhancing efficiency, precision, and adaptability. However, due to problems related to accountability and consistency, agentic AI is not yet sufficient to replace current procedures; while certain challenges and limitations persist, the system's ability to integrate complex variables and navigate stakeholder dynamics paves the way for more informed and flexible infrastructure solutions. To conclude, the added value of agentic AI is notable in terms of efficiency, flexibility, and independent decision-making, while relying on existing theories.

6 Recommendations

Agentic AI has considerable potential based on this proof of concept. The current application is limited but shows promise in automating complex engineering processes. What additional steps are necessary to fully implement and extend the application of agentic AI in standard practices within the field of civil engineering? To answer that, this section is split into two: first, further research is treated, and after which, the implementation in the field is treated.

6.1 Further research

For further research, it is good to consider the strengths and weaknesses of this agentic AI model. Firstly, the LLM. Many LLMs logically give different results than the used LLM. Further research could identify which LLM performs best in such models, thereby improving model performance. Next to the LLM itself, LLM settings such as temperature (i.e., randomness) are also important. For this study, no sensitivity analysis was conducted by varying these settings. For future research, combining the use of different LLMs and various settings could provide insight into the variation in process and results between these LLMs and their respective settings.

Secondly, integrating other components of dike design into these models would provide a broader application and more reliable results. This model only considers stakeholder integration and macro-stability as failure mechanisms. For future research, integrating other failure mechanisms, such as piping, overtopping, and outer slope erosion, would yield a design that is more reliable, as it also assesses factors beyond macro-stability.

Other technologies could also improve the performance of such models; an LLM could be trained. By enriching these models with specialised datasets on dikes and geotechnical and hydraulic engineering, their ability to provide accurate and integrated information can be significantly enhanced. This will enable AI systems to autonomously make decisions that optimise efficiency, precision, and adaptability in the dyke design process, thereby facilitating technical integration and effective stakeholder communication.

Additionally, comparing an agentic AI model to an agentic workflow would be an interesting approach. The agentic AI model lacks a completely defined flow, whereas the agentic workflow does. Here, the model has less freedom, but the necessity of freedom is doubtful since it increases the chance of errors.

6.2 Applications in the field

In the work field, applying such models for final designs is too early. These models are not reliable enough for that purpose. However, for initial designs, this model could be helpful. Initial designs are more of an estimate anyway, without optimising the design. The initial design needs to be assessed for other failure mechanisms, but it provides a strong basis for the macro-stability aspect. However, since the dike is considered safe when the factor of safety reaches 1.3, it is assumed that the design should also be tested for macro-stability. Since the model lacks hard evidence of stability, this should also be verified using the complex assessment methods employed by Rijkswaterstaat. However, the integration of stakeholder preferences and settings demands the use of the LLM's language skills. This is the speciality of the system, and therefore, the analysis of reports, stakeholder input, and demands is something LLMs could improve this process with nowadays. For these purposes, the LLM is significantly faster and more efficient than humans. For projects with increasing numbers of stakeholders, this could make a significant difference, as the LLM can process large amounts of data extremely quickly.

References

Acharya, Deepak Bhaskar, Karthigeyan Kuppan, and B. Divya (2025). "Agentic AI: Autonomous Intelligence for Complex Goals-A Comprehensive Survey". In: *IEEE Access*. DOI: 10.1109/ACCESS.2025.3532853.

Begg, Chloe et al. (2017). "The role of local stakeholder participation in flood defence decisions in the United Kingdom and Germany". In: *Journal of Flood Risk Management* 11.2, pp. 180–190. DOI: 10.1111/jfr3.12305.

Brugge, R. Van der, J. Rotmans, and D. Loorbach (2005). "The transition in Dutch water management". In: *Reg Environ Change (vol 5)*. DOI: 10.1007/s10113-004-0086-7.

Covello, Vincent and Miley Merkhofer (1993). *An Evaluation of the State of the Art. In: Risk Assessment Methods. Springer*. Springer New York. DOI: 10.1007/978-1-4899-1216-9_6.

Guichelaar, Louise (2024). "Design optimization for flood defenses using machine learning." In: *https://resolver.tudelft.nl/40fd11bc-21fb-4704-b961-8329416ebdda*.

Hannema, Patrick (2025). *Conversation on stakeholder participation and standard demands in dike design*. Personal communication: May 1, 2025.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The Elements of Statistical Learning*. Springer New York. Chap. Shrinkage methods, p. 62. DOI: 10.1007/978-0-387-84858-7.

Hosseini, Soodeh and Hossein Seilani (2025). "The role of agentic AI in shaping a smart future: A systematic review". In: DOI: 10.1016/j.array.2025.100399.

Jong, Koen de (2025). *Conversation on preparing agentic AI application preparation*. Personal communication: Februari 2025.

LangGraph (2025). *Research Assistant*. Accessed: April 21, 2025. URL: https://github.com/langchain-ai/langchain-academy/blob/main/module-4/research-assistant.ipynb.

Loon-Steensma, J. M. van and P. Vellinga (2014). "Robust, Multifunctional flood defenses in the Dutch rural riverine area". In: *Natural Hazards and Earth System Sciences*. DOI: 10.5194/nhess-14-1085-2014.

Mishra, Anurag (2025). *Building Multi-Agents Supervisor System from Scratch with LangGraph Langsmith*. Accessed: April 20, 2025. URL: https://medium.com/@anuragmishra_27746/building-multi-agents-supervisor-system-from-scratch-with-langgraph-langsmith-b602e8c2c95d.

Rijkswaterstaat (2017). *Handreiking ontwerpen met overstromingskansen*.

Salinas, Abel and Fred Morstatter (2024). *The Butterfly Effect of Altering Prompts: How Small Changes and Jailbreaks Affect Large Language Model Performance*. DOI: 10.48550/arXiv.2401.03729.

Stelling, Lennart (2025). *Conversation on most common stakeholder preferences in 2d dike design*. Personal communication: May 12, 2025.

Sun, Jun, Yiteng Pan, and Xiaohu Yan (2025). "Improving intermediate reasoning in zero-shot chain-of-thought for large language models with filter supervisor-self correction". In: *Neurocomputing* 620, p. 129219. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2024.129219.

Twijnstra, Jetze (2025). *Conversation on stakeholder analysis in dike design*. Personal communication: May 23, 2025.

van Bree, Bob et al. (2018). *Werkwijzer Ontwerpen Waterkerende Kunstwerken - Ontwerpverificaties voor de hoogwatersituatie*.

Van Wijk, Ronald (n.d.). *Donkere lucht boven dijk*. URL: https://www.ronaldvanwijk.nl/foto/68599-donkere-lucht-dijk/.

Zhang, Weimin et al. (2025). "A sepsis diagnosis method based on Chain-of-Thought reasoning using Large Language Models". In: *Biocybernetics and Biomedical Engineering* 45.2, pp. 269–277. ISSN: 0208-5216. DOI: https://doi.org/10.1016/j.bbe.2025.04.002.

A Prompts

A.1 Stakeholder Agent LLM1 prompt

The prompt of the first LLM in the stakeholder agent is shown below. The prompt instructs the agent to identify and rate the stakeholders in the project.

The prompt:

```
You are a stakeholder identificator for a dike design project.
Identify all relevant stakeholders (about 20) for the scenario: {state['messages']}.
This is an example of stakeholder for another project: {example_stakeholders}.

**Power**
0-4: Low Power
Example: A local community group focused on cultural heritage with no direct
financial contribution or legislative power.
Characteristics: Limited influence on the project decisions; primarily able to voice
opinions but cannot enforce changes.
4-7.5: Medium Power
Example: Regional environmental organizations with advisory influence and moderate
funding capabilities.
Characteristics: Can affect project design through advocacy and engagement but do not
hold ultimate decision-making authority.
7.5-10: High Power
Example: Government regulatory agencies with statutory authority and substantial
funding for the project.
Characteristics: Can approve or veto project plans; major stakeholders in guiding
project direction due to legal jurisdiction and financial ownership.

**Interest**
0-4: Low Interest
Example: Stakeholders from industries not directly impacted by the project, such
as distant tech companies.
Characteristics: Minimal concern for outcomes; project impact is tangential and
does not affect core operations or objectives.
4-7.5: Medium Interest
Example: Residents living near the project who may face indirect impacts such as
aesthetic changes or minor inconvenience.
Characteristics: Interested mainly in neighborhood enhancements and community
benefits; some potential for engagement to voice concerns.
7.5-10: High Interest
Example: Local businesses located along the project area with direct stakes in
traffic flow, accessibility, or economic opportunities.
Characteristics: Highly invested in project success; outcomes could significantly
influence their operations, profitability, and daily activities.

**Participation levels**
Inform: low/low, medium/low, low/medium power/interest.
Consult: medium/medium, high/low, low/high power/interest.
Involve: medium/high, high/medium power/interest.
Collaborate: high/high power/interest.
```

```
Idenfity all stakeholders, rate their power and interest and assign a participation
rate, information on where these fit: {participatie_niveaus}.
These participation rates will later on be used to create a participation plan and
the power-interest rates for a power interest grid, to make the power-interest
grid viewable give all stakeholders slightly different ratings.
```

A.2 Stakeholder Agent LLM2 prompt

The prompt of the second LLM in the stakeholder agent instructs the agent to write a participation plan and to create a list of SMART demands and a list of wishes. The prompt is shown below.

The prompt:

```
You are the start of a dike design project. Your task is to interpert the scenario,
write a participation plan for these stakeholders:
{stakeholder_identification["stakeholders"]}.

Use this as a reference for the participation plan: {participatie_niveaus}
Then, it is your task to check the demands and wishes from the dataframe and
from the state.
The demands need to be included and wishes can be granted.
Make a list of demands which are relevant for the 2d design and transform these
to concrete smart demands.

This is a list of demands from the client, the demands relevant for 2d stability design
need to be included in the answer in the smart demands: {df_relevant}.
Make as much smart demands as you need relevant for 2d dike design, **do not include
ecological, environmental or multifunctional demands (i.e. recreational or roads).
transform the user input to a list of demands, be specific: {state["messages"]}. When
a dike height, thickness cohesive layer or core material is given pass this on clearly
(thickness cohesive layer is clay layer beneath dike).
wishes can also be included in the wishes key. Use this for optimalization specifics,
such as "ensure minimal dike width" or "maximalize ditch depth"
```

A.3 Supervisor Agent prompt

The prompt of the supervisor agent gives the agent context, a task, an example process and inforamtion on its options regarding the flow of the model. It also receives several state keys such as the latest analysis and dike dimensions in order to make its input variable (and thus the results).

The prompt:

```
# Supervisor Agent Instructions
**Role**: You are the supervisor agent for an agentic dike design system.
# System Details
- **Messages**:
    - Track and analyze all communications within the system. This is your communication
    history: {state["messages"]}.
    - Use the dike dimension agent to adjust dimensions and get a new design. Use
    the stakeholder checker to get feedback on the design.
```

```
    - Use the dike dimension agent as often as necessary ensure a Factor of Safety (FoS)
    **absolutely** within this range 1.3-1.5. Make sure its is as close a possible to 1.3.
    - Make sure an optimal design is made by giving feedback on the design by assessing
    current situation and desired situation for:
    1. Factor of Safety,
    2. Dike width,
    3. Dike height,
    4. Other dimension (for example thickness cohesive layer)

Example (for one aspect):
    Current situation: dike height is 5.5 meters.
    Desired situation: dike height is minimal height to minimize space use and
    optimize FoS by lowering height.

- **Dike Dimensions**:
    - Review dimensions (note that the last item in the list is current):
    `{state['dike_dimensions'][-1]}` and decision logic: {state['dike_agent_logic'][-1]}.

- **Demands \& Wishes**:
    - Understand the specific requirements and constraints: {state["demands"]}.
    - Incorporate wishes: {state["wishes"]}.

- **Stakeholder Check Agent**:
    - Evaluate the design against demands and wishes and provide feedback:
    {state["stakeholder_analysis"]} & `{state['demand_check']}
    - *Note*: If keys are empty, the system has just been initialized. Run stakeholder
    analysis when the initial dike design has been made.

## Critical Considerations
- **Stakeholder Agent**:
    - The stakeholder agent always adopts a critical stance regarding the design processes.

- **Design Approval**:
    - Your primary task is to determine when the design meets sufficient criteria.

    ---

## Dike Design Assistant Overview
**Capabilities**: You are a smart dike design assistant that utilizes two specialized
agents.

### Agent Roles
1. **Dike Designer Agent**:
    - *Function*: Adjusts dimensions and evaluates structural stability.
    - *Criteria for Safety*: A Factor of Safety (FoS) of at least 1.3 indicates that
    the dike is safe.

## Design Parameters - stay WITHIN these ranges!!

- **Dike Height**: *(Range: 5 - 6)*
    - *Influence*: Higher dike reduces overtopping but reduces stability.
```

```
    - *Note*: Water height is 4.5 m; ensure an additional 0.5 m to guard against wave
    overtopping.

- **Crest Width**: *(Range: 3 - 10)*
    - *Effect*: Broader crests enhance stability but costs space.

- **Inner Slope**: *(Range: 2 - 4)*
    - *Impact*: Higher values increases stability (expressed as 1: variable slope)
    but costs space.

- **Inner Berm Length**: *(Range: 0 - 15)*
    - Higher value optimizes stability at the expense of space.

- **Thickness Cohesive Layer**: *(Range: 0 - 8)*
    - Critical to stability score; typically immutable. Higher scores diminish dike
    integrity, this means thickness of clay layer beneath dike.
    - *Guideline*: Set to zero if the dike is built on sand. Value is often given in
    input otherwise set to 0.

- **Distance Toe to Ditch**: *(Range: 0 - 5)*
    - Balances stability against spatial costs.

- **Depth Ditch**: *(Range: 0.5 - 1.25)*
    - *Risk*: Depth inversely affects stability.

2. **Stakeholder Agent**:
    - *Function*: Assesses the feasibility of the current dike design and optimizes
    the design.
    - *Supervisor Role*: Must evaluate the feedback and decide whether to initiate
    a new iteration or conclude the design.
    - *Finalizing Design*: Once both agents agree on a design, the process concludes.

---

## Implementation Notes
    - Do not include environmental, ecological or multifunctional factors in the
    design or feedback, this is out of the scope and capabilities of the agents.
    - You are not responsible for deciding or altering dimensions or directly
    assessing stability.
    - Agents possess the authority to modify specific state keys.
    - Focus on enhancing collaboration among agents to refine the design iteratively.
    - Your role involves adding relevant messages to the state to provide context
    and facilitate the agents' tasks.
    - Your role is to decide when the process is finished; the stakeholder agent
    is programmed to always be critical, while you determine when stakeholder
    preferences are sufficiently implemented.
    - The dike design agent is programmed to reach at least 1.3 as a FoS.
    **When the FoS is close to but over 1.3, minimum height = height and thickness
    cohesive layer is considered. Go to finish.**
    - Transition to **FINISH** in the next key upon achieving an optimal design
    after several collaboration cycles (max 24; see the list of messages) current:
```

```
{len(state["messages"])-1}. If extra iterations do not improve the design,
also proceed to **FINISH**.

## Normal Design Process:
1. Call the dike design agent to initiate the design process.
2. Return to the supervisor; the supervisor calls the stakeholder checker
to evaluate the design.
3. Return to the supervisor; the supervisor calls the dike design agent and
provides points for improvement in the agent message key of the structured output.
4. Return to the supervisor; the supervisor calls the stakeholder checker to
evaluate the design.
5. From this moment on, the supervisor must judge whether demands are sufficiently
included in the design. If insufficient, repeat the following loop:
    - Call the design agent to improve the design.
    - Call the stakeholder checker for feedback on the design.
6. When the design is considered optimal, proceed to finish. """
```

A.4 Dike design Agent prompt

The prompt of the dike design agent provide information on the parameter ranges of the meta-model it utilises, it gets a task from the supervisor and feedback from the demand check agent as well as the demands that are set and a history of previously tried dikes with its assessment results. An example process is provided to give the LLM an idea of the flow of the agent.

The prompt:

```
# Dike Dimension Specialist Instructions

**Mission**: Alter dimensions to achieve an optimal dike design.

## Current priority
**{priority}**

## Key Objectives

- **Factor of Safety (FoS)**: Aim for a FoS of at least 1.3; higher is more stable
but also more expensive, make sure its below 1.5. A FoS lower then 1.3 must always
return a "retry" in the overall key. That is also for 1.299 and everything below.
- **Stability Checker**: The dimensions you choose will be assessed on stability
by a tool. Choose carefully.
- **Adjust stability factors**: Adjust stability factors a few times radically to
get a feeling for the impact of factors, previously tried dimensions are shown in
the history: {state["dike_dimensions"]}. When reassessed these do not change, do
not try the samen dimensions again.
- **Minimalize width**: Minimalize dike width by adjusting spatial dimensions
(crest width, inner slope width, dist toe ditch, inner slope) when you need a
lower FoS try lower dimensions for these.
## Design Parameters - stay WITHIN these ranges!!

- **Dike Height**: *(Range: 5 - 6)*
- *Influence*: Higher dike reduces overtopping but reduces stability, make dike
smaller for optimization (also makes dike less wide).
- *Note*: Water height is 4.5 m; ensure an additional 0.5 m to guard against wave
```

overtopping. Min dike height is often given, use this as dike height.

- **Crest Width**: *(Range: 3 - 10)*
- *Effect*: Broader crests enhance stability but costs space, recommended to increase stability when FoS is not satisfied.

- **Inner Slope**: *(Range: 2 - 4)*
- *Impact*: Higher values increases stability (expressed as 1: variable slope) but costs space.

- **Inner Berm Length**: *(Range: 0 - 15)*
- Higher value: increases stability at the expense of space.

- **Thickness Cohesive Layer**: *(Range: 0 - 8)*
- Critical to stability score; typically immutable. Higher scores diminish dike integrity, this means thickness of clay layer beneath dike.
- *Guideline*: Set to zero if the dike is built on sand. Value is often given in input otherwise set to 0.

- **Distance Toe to Ditch**: *(Range: 0 - 5)*
- Balances stability against spatial costs.

- **Depth Ditch**: *(Range: 0.5 - 1.25)*
- *Risk*: Depth inversely affects stability.

Contextual Considerations
- **Current Dimensions**: {state['dike_dimensions'][-1]} and choice justification {state["dike_agent_logic"][-1]}. Judge the design initially on FoS, then on dimensions. FoS of latest dike (the one you assess right now): {state["dike_dimensions"][-1]['FoS']}
- **Supervisor Messages**: Prioritize requests from the supervisor: {state['messages'][-1]}.
Assess whether these are sufficiently implemented in design.
- **Demand check feedback**: {state["demand_check"][-1]}

Dike Core Material

- **Material Type (often given)**:
- *Clay (0)*: Generally more stable but incurs greater cost.
- *Sand (1)*: Always shielded by a protective clay layer.

Decision Protocols

- When state dimensions fail to secure a sufficient FoS (>1.3, <1.5), examine and modify relevant stability factors, a new design will be assessed.
- You should decide when a dike is safe and when supervisor preferences are sufficiently implemented. Safety always has the priority.
- First ensure safety, then optimize to get in the range of 1.3-1.35. Moderately change dimensions when close to range.

Example Process:
1. Check supervisor feedback & demand check.

```
    2. Assess current design based on feedback & demands. (when sufficient return to
    supervisor.)
    3. Think what needs to be changed to comply with said demands & feedback (for example,
    first increase stability to reach FoS, then optimize width).
    4. **Check dimension history to select and eliminate options** and see impact of
    changes. Retrying dimensions that fail does not work.
    5. When a historical dimensions complies or almost complies you can slightly
    change it to come to the final design, optimize by making small changes.
    6. Provide new dimensions based on the design parameters.
    7. Tool checks new dimensions
    8. Start over
---
```

A.5 Demand checker Agent prompt

The demand checker agent's prompt contains information about the ranges of the metamodel as well as the current design and demands from the stakeholder agent.

The prompt:

```
# Stakeholder Analysis checker Instructions

**Role**: Ensure that all demands are included in the dike design decisions.

## Optimization factors.

- **Space (Low Width)**: Stakeholders prioritize minimizing width for spatial
efficiency.
- **Ditch Depth**: Water boards and farmers prefer a ditch as deep as possible
for optimal function.
- **Minimal FoS**: Make sure the FoS is above, but as close as possible to 1.3.
Try to reach FoS between 1.3 and 1.32.
- **Minimal dike height**: Make sure the dike is high enough but not unneccesarily high.

## Dimensions and Parameters (context for options)

- **Dike Height**: *(Range: 3.5 - 6)*
- Higher dikes reduce overtopping but may compromise stability. Water height is 4.5 m;
ensure an additional 0.5 m for wave retention.

- **Crest Width**: *(Range: 3 - 10)*
- Broader width enhances stability.

- **Inner Slope**: *(Range: 2 - 4)*
- Higher slope values boost stability (expressed as 1: variable slope).

- **Inner Berm Length**: *(Range: 0 - 15)*
- Increased length facilitates stability but sacrifices space.

- **Thickness Cohesive Layer**: *(Range: 0 - 8)*
- Influences stability score and resistance; may be intrinsically unalterable.
A higher score may adversely affect dike.
```

```
- **Distance Toe to Ditch**: *(Range: 0 - 5)*
- Greater distances provide more stability, with spatial trade-offs.

- **Ditch Depth**: *(Range: 0.5 - 1.25)*
- Increased depth may reduce stability.

- **Dike Core Material**:
- *Clay (0)*: Offers greater stability but incurs higher costs.
- *Sand (1)*: Economical but always layered beneath clay for protection.

## Evaluation and Output

- **Prepare Power Interest Grid**: Use the ‘StakeholderCheck‘ typedDict to rate
stakeholders on a scale of 1-10, weighing both spatial width and ditch depth importance.

- **Contextual References**:
- Review supervisor feedback: ‘{state["messages"][-1]}‘
- Analyze current configurations: ‘{state["dike_dimensions"][-1]}‘
- Unpack prior stakeholder actions: ‘{state['stakeholder_analysis']}‘
- Confirm compliance with existing demands: ‘{state["demands"]}‘
- Consider stakeholder wishes: {state["wishes"]}

- **Cost Estimation**:
- Estimate total costs (area multiplied by material costs, sand=12€/m2, clay=19€/m2):
{self.costs}
- Overview total width for current configuration: {self.total_width}

## Decision Protocols

When reporting, ensure:
- Balancing stakeholder critiques with engineering mandates.
- Comparing design strategies with anticipated costs and stakeholder satisfaction.
- Seeking optimal alignment with resilience and feasibility aspects.
- Clearly state what you want to be improved (from stakeholder perspective), be
critical.
- Only include feedback on the 2d design and dimensions. No environmental, ecological
or multifunctionality factors.

---
```

B State

The state is the communication means of the system as well as the memory of the system. The state is a TypedDict python structure in where several keys have a set output is a Json structure. The state can be found in Figure 8. The keys have a defined structure for the output. For example the messages key can only add messages to itself.

```
#The state is the database from which the agents retrieve data and which contains messages and variables.
class State(TypedDict):
    next: str
    messages: Annotated[list[AnyMessage], add_messages]
    demands: List[SMART]
    wishes: List[str]
    stakeholder_analysis : List[str]
    demand_check: List[DemandCheck]
    dike_dimensions : List[DikeDimensions]
    dike_agent_logic : List[str]
```

Figure 8: The overall state of the system containing all relevant data for other points of the system or the results.

In the state there are a few nested TypedDict (i.e., TypedDict inside of a TypedDict) that have their own definitions. These can be found in Figure 9, Figure 10, and Figure 11.

```
class SMART(TypedDict):
    """layout for smart demands of a dike. Write a concrete demand in the SMART_demand key using all factors.
    Do not include a timeline since this is irrelevant for this task."""
    Specific: str
    Measurable: str
    Achievable: str
    Relevant: str
    SMART_demand: str
```

Figure 9: The TypedDict structure of SMART.

```
class DikeDimensions(TypedDict):
    dike_height : float
    crest_width : float
    inner_slope : float
    inner_berm_length : float
    distance_toe_ditch : float
    water_height : float
    ditch_depth: float
    thickness_cohesive_layer : float
    dike_core : int
    FoS : float
```

Figure 10: The TypedDict structure of DikeDimensions.

```python
class DemandCheck(TypedDict):
    "Per demand structure whether it is sufficiently included in the design."
    Demand: str
    Included: Literal["Yes", "No"]
    Improvements: str
```

Figure 11: The TypedDict structure of DemandCheck.

C Evaluation rubric

Table 5: Evaluation Rubric for Agentic AI Model for Dike Design

Criteria	1 - Insufficient	2 - Moderate	3 - Sufficient	4 - Good	5 - Excellent
Logical Soundness	Model exhibits inconsistencies and lacks transparency.	Model shows some inconsistencies and limited transparency.	Model is mostly consistent and reasonably transparent.	Model is consistent and largely transparent.	Model is exceptionally consistent and fully transparent.
Completeness	Major factors are entirely missing.	Several factors are not fully addressed.	Most relevant factors are addressed.	Nearly all relevant factors are adequately addressed.	All factors are thoroughly and completely addressed.
Accuracy	Results are inaccurate, outside acceptable margins.	Results are somewhat inaccurate.	Results are reasonably accurate, with minor deviations.	Results are accurate, with minimal deviations.	Results are extremely accurate and consistent.
Acceptability	Lack of trust from stakeholders.	Limited trust from stakeholders.	Reasonable trust, with some concerns.	High trust, with minor stakeholder queries.	Full trust and acceptance from all stakeholders.
Practicality	System is difficult to use and poorly integrates.	System is somewhat challenging to use, with integration issues.	System is reasonably user-friendly and generally integrates well.	System is user-friendly and well integrated.	System is highly user-friendly and seamlessly integrated.
Effectiveness	Model contributes little to improved decision-making.	Limited contribution to decision-making improvements.	Regularly contributes to decision-making improvements.	Significantly contributes to better decision-making.	Exceptionally contributes to enhanced decision-making.
Flexibility	Model does not adapt to new scenarios or changes.	Adaptation to changes is cumbersome and limited.	Model can somewhat integrate changes and adapt.	Model integrates changes smoothly and adapts well.	Model is highly adaptable, integrating changes seamlessly.

D Appendix D: Model results

D.1 Scenario 1

This scenario is randomly generated by the scenario writer in the model. These are the main parameters in the input scenarios:

Parameter	Value	Unit
Dike Height	5.5	meter
Thickness Cohesive Layer	4	meter
Dike Core	Sand	-

Table 6: Main parameters in dike design scenario 1.

To summarise the results; the system designs a dike with a Fos of 1.31 and integrates all demands in the design. The demands were accurately drawn from the scenario.

Outputs

Participation Plan

Collaborate:
Waterschap Drents Overijsselse Delta and Rijkswaterstaat will be involved in strategic decision-making meetings to ensure alignment with national and regional water management policies. Regular meetings will be held to discuss project scope, financial commitments, and design approvals.

Involve:
Local Community and Farmers, Provincie Overijssel, Local Government (Municipality of Olst), Transport and Infrastructure Agencies, and Emergency Services will participate in integral design meetings and dike thinkers sessions. Individual consultations will be organized to address specific concerns related to property and infrastructure.

Consult:
Nature Organizations, Environmental Advocacy Groups, Local Businesses, Construction Companies, Fishing and Aquaculture Associations will be consulted to provide input on environmental studies and design choices. Group discussions will be organized to gather insights and feedback.

Inform:
Historical Preservation Societies, Local Schools and Educational Institutions, Tourism Boards, Local Media, Public Health Organizations, Recreational Clubs, and Water Sports Associations will be kept informed through newsletters, social media updates, and public information sessions.

Demands

Design Height:
Specific: The dike must be designed to a minimum height of 5.5 meters above NAP.
Measurable: Height will be verified through topographic surveys.
Achievable: Design calculations will ensure feasibility.
Relevant: Ensures protection against a once-in-10,000-year flood.
SMART_demand: Design the dike to a minimum height of 5.5 meters above NAP.

Factor of Safety (FoS):
Specific: The dike must have a FoS between 1.3 and 1.5.
Measurable: Stability analysis will confirm FoS.
Achievable: Engineering methods will ensure compliance.

Relevant: Guarantees structural integrity.
SMART_demand: Ensure the dike has a FoS between 1.3 and 1.5.

Core Material:
Specific: The dike core must consist of sand.
Measurable: Material specifications will be documented.
Achievable: Local sand sources will be utilized.
Relevant: Provides drainage and cost-effectiveness.
SMART_demand: Use sand for the dike core.

Cohesive Layer Thickness:
Specific: The cohesive clay layer beneath the dike must be 4 meters thick.
Measurable: Geotechnical surveys will verify thickness.
Achievable: Clay sourcing will be planned.
Relevant: Ensures stability and water tightness.
SMART_demand: Ensure a cohesive clay layer thickness of 4 meters beneath the dike.

Wishes:

Ensure minimal dike width to accommodate maintenance paths.
Maximize ditch depth for effective drainage.
Integrate ecological elements where feasible, such as natural materials and habitats.
Facilitate stakeholder engagement through regular updates and feedback sessions.

Explanation

The current design meets the demands for minimum height, FoS, and cohesive layer thickness. The crest width is towards the wider end but aligns with stability requirements. Further iterations may not significantly improve the design.

Dike Dimensions

Parameter	Value	Unit
Dike Height	5.5	meter
Crest Width	7	meter
Inner Slope	1:3.5	-
Inner Berm Length	12	meter
Thickness Cohesive Layer	4	meter
Distance Toe Ditch	2.5	meter
Ditch Depth	1	meter
Dike Core	Sand	-
FoS	1.3069	-
Water Height	4.5	meter

Table 7: Dimensions of the final dike design of design scenario 1.

D.2 Scenario 2

This scenario involves designing a dike along the river IJssel, above Deventer, towards Terwolde. The dike must be at least 5.3 meters high and includes a clay layer of 2 meters. This area does not solely serve flood prevention but is part of a multifunctional region combining recreational, ecological, and economic interests. To summarise the results; the results effectively integrate the demands and wishes in the design attaining a FoS of 1.36.

Complex Stakeholder Issues: Scenario

Community Leaders and Residents

Concerned about the project's impact on houses and costs but prioritize safety against flooding.

Environmental Organizations

Advocate for the preservation of local ecosystems without disturbance, preferring minimal intervention in natural habitats.

Recreational Advocates

Want the area to remain accessible for recreation, fearing the higher dike may restrict paths for walkers and anglers.

Adjacent Businesses and Cultural Historians

Worry about the impact on infrastructure and access for transportation. Historians want nearby cultural sites protected from potential damage.

Engineers and Safety Experts

Highlight the need for a robust dike design to withstand future sea-level rise and extreme weather, insisting on the utmost protection and safety.

Underlying Complexity

The core issue lies in balancing maximum protection and safety against maintaining environmental values and recreational access. Engineers propose a higher and structurally wider dike, prioritizing safety but potentially compromising accessibility and landscape visibility. They argue this design offers the strongest defense against extreme weather events.

Parameter	Value	Unit
Dike Height	5.3	meter
Thickness Cohesive Layer	2	meter
Dike Core	Clay	-

Table 8: Main parameters in dike design scenario 2

Participation Plan

Collaborate:
Engineering Firms, Safety Experts, Local Government, Water Management Authorities
Design loop 0: Create a shared vision of the project scope and communicate a core message.
Design loop 1: Establish measures and spatial requirements with administrative commitment.
Design loop 2: Secure financial commitments and agree on project decision content.
Design loop 3: Release the design project decision for procedures.

Involve:
Local Residents, Environmental Groups, Adjacent Businesses, Regional Government, Economic Development Agencies, Community Leaders
Design loop 0: Inform stakeholders about the approach and align expectations for plan development.
Design loop 1: Collaborate with stakeholders to address issues affecting project scope.
Design loop 2: Engage stakeholders in discussions about area integration and environmental

studies.
Design loop 3: Shift focus to informing stakeholders about design adjustments and environmental impact reports.

Consult:
Transport Authorities, Tourism Boards, Construction Companies, Public Health Organizations
Design loop 0-3: Consult stakeholders to enrich environmental studies and weigh design choices.

Inform:
Recreational Advocates, Cultural Historians, Agricultural Associations, Media Outlets, Academic Institutions
Design loop 0-3: Inform stakeholders about project progress through newsletters, social media updates, and public information sessions.

Demands

- **Design Height:**
  - *Specific:* Design a dike with a minimum height of 5.3 meters.
  - *Measurable:* Height measured from the base to the top of the dike.
  - *Achievable:* Utilize local materials and engineering expertise.
  - *Relevant:* Ensures protection against flooding and extreme weather.
  - *SMART demand:* The dike must be at least 5.3 meters high to provide adequate flood protection.
- **Cohesive Layer Thickness:**
  - *Specific:* Ensure the dike has a cohesive clay layer of 2 meters beneath it.
  - *Measurable:* Thickness of the clay layer measured in meters.
  - *Achievable:* Source clay locally and use standard construction techniques.
  - *Relevant:* Provides structural stability and water resistance.
  - *SMART demand:* The dike must have a cohesive clay layer of 2 meters beneath it for stability.
- **Factor of Safety (FoS):**
  - *Specific:* Maintain a Factor of Safety (FoS) between 1.3 and 1.5.
  - *Measurable:* FoS calculated using engineering models and simulations.
  - *Achievable:* Apply advanced engineering methods and materials.
  - *Relevant:* Ensures structural integrity and safety.
  - *SMART demand:* The dike's design must achieve a Factor of Safety between 1.3 and 1.5.
- **Water Height Difference:**
  - *Specific:* Design the dike to be 0.5 meters higher than the water height.
  - *Measurable:* Height difference measured from water level to dike top.
  - *Achievable:* Adjust design based on water level data.
  - *Relevant:* Prevents overtopping during high water events.
  - *SMART demand:* The dike must be 0.5 meters higher than the water height to prevent overtopping.

Wishes

- Ensure minimal dike width to optimize land use.
- Maximize ditch depth for effective water drainage.
- Incorporate historical site protection measures.
- Facilitate access for adjacent businesses during construction.
- Maintain recreational access paths around the dike.

Explanation

The current design meets the demands and stakeholder wishes effectively. The factor of safety is slightly above 1.3, fulfilling the safety requirement without excessive over-design. Height and crest width are optimized for spatial efficiency, and stakeholder feedback indicates satisfaction with current parameters.

Dike Dimensions

Parameter	Value	Unit
Dike Height	5.3	meter
Crest Width	3.5	meter
Inner Slope	1:3.5	-
Inner Berm Length	12	meter
Thickness Cohesive Layer	2	meter
Distance Toe Ditch	2	meter
Ditch Depth	1.25	meter
Dike Core	Clay	-
FoS	1.3583	-
Water Height	4.5	meter

Table 9: Dimensions of the final dike design for design scenario 2

D.3 Scenario 3

The scenario has a difficult macro-stability aspect. This dikes needs to have the max height, max thickness of cohesive layer and a sand dike core, which all negatively impact stability. To summarise the results; the design effectively integrates all demands and achieves a FoS of 1.34.

Parameter	Value	Unit
Dike Height	6	meter
Thickness Cohesive Layer	8	meter
Dike Core	Sand	-

Table 10: Main parameters in dike design scenario 3

Participation plan

Collaborate

- **Local Government of Enschede**
- **Rijkswaterstaat**

- **Ministry of Infrastructure and Water Management**
- **Water Board**

Approach:

- Regular meetings to discuss project scope and decisions.
- Joint creation of project vision and core message.
- Commitment to measures and space allocation.
- Agreement on financial resources and project decision content.
- Release of design project decision for procedures.

Involve

- **Provincie Overijssel**
- **Environmental Protection Agency**

Approach:

- Integral design meetings and dike thinkers sessions.
- Individual discussions for specific stakeholder concerns.
- Collaboration on environmental studies and area integration.
- Group discussions for multi-stakeholder issues.

Consult

- **Local Residents**
- **Agricultural Landowners**
- **Construction Companies**
- **Emergency Services**
- **Public Health Organizations**

Inform

- **Local Businesses**
- **Transport and Logistics Companies**
- **Tourism Board**
- **Cultural Heritage Organizations**
- **Local Schools and Universities**
- **Media Outlets**
- **Recreational Groups**
- **Local NGOs**

- **Energy Companies**

Approach:

- Information evenings and newsletters.
- Updates via website and social media.
- Regular communication on project progress.

SMART Demands

Demand 0

- **Specific:** Design a dike with a height of at least 6 meters on a clay layer of 8 meters with a sand core.
- **Measurable:** Dike height is 6 meters, clay layer is 8 meters thick.
- **Achievable:** Utilize available materials and engineering techniques.
- **Relevant:** Ensures water safety and structural integrity.
- **SMART demand:** Design a dike with a height of at least 6 meters on a clay layer of 8 meters with a sand core.

Demand 1

- **Specific:** Ensure the dike height is at least 0.5 meters higher than the water height.
- **Measurable:** Water height is 4.5 meters, dike height is 5 meters.
- **Achievable:** Adjust design specifications to meet height requirements.
- **Relevant:** Prevents overflow and ensures safety.
- **SMART demand:** Ensure the dike height is at least 0.5 meters higher than the water height.

Demand 2

- **Specific:** Maintain a Factor of Safety (FoS) between 1.3 and 1.5.
- **Measurable:** FoS is calculated and maintained within specified range.
- **Achievable:** Use engineering calculations and safety assessments.
- **Relevant:** Ensures structural stability and safety.
- **SMART demand:** Maintain a Factor of Safety (FoS) between 1.3 and 1.5.

Demand 3

- **Specific:** Design the dike to be autonomously water-retaining.
- **Measurable:** Dike functions independently without external support.
- **Achievable:** Incorporate water-retaining features in design.
- **Relevant:** Ensures continuous water safety.
- **SMART demand:** Design the dike to be autonomously water-retaining.

Explanation:

The design meets critical safety and structural requirements, as well as stakeholder preferences for minimal space usage while maintaining optimal safety. Further iterations are unlikely to yield significant improvements, thus the process should be concluded successfully.

Dike dimensions:

Parameter	Value	Unit
Dike Height	6	meter
Crest Width	8	meter
Inner Slope	1:4	-
Inner Berm Length	12	meter
Thickness Cohesive Layer	8	meter
Distance Toe Ditch	4	meter
Ditch Depth	0.6	meter
Dike Core	Sand	-
FoS	1.3438	-
Water Height	4.5	meter

Table 11: Final dike dimensions for design scenario 3

D.4 Grebbedijk

This design scenario involves the Grebbedijk from dike post 23+40, near Jachthaven street, to dike post 35+80, near Aan de Rijn street. To summarise the results, the actual dike seems to have a FoS of 1.55 while the design from the system achieves a FoS of 1.31.

Parameter	Value	Unit
Dike Height	5.70	meter
Outer Slope	1:3	-
Cohesive Layer	0	meter
Dike Core	Sand	-

Table 12: Main parameters for Grebbedijk design scenario

Participation Plan for Stakeholders

Collaborate:
Municipality of Wageningen, Rijkswaterstaat, Water Board, Ministry of Infrastructure and Water Management
Approach: Engage in regular strategic meetings to align on project scope and objectives. Jointly develop a core message and communicate it consistently. Secure financial commitments and agree on project decisions. Release design decisions for procedural steps.

Involve:
Local Residents, Provincie Gelderland, Local Businesses, Emergency Services
Approach: Conduct integral design meetings and dike thinkers sessions. Organize individual discussions for specific concerns. Collaborate on environmental studies and design integration. Inform stakeholders about design adjustments and environmental impact reports.

Consult:
Environmental Protection Agency, Construction Companies, Local Transport Authority, Wildlife Conservation Groups, Public Health Department

Approach: Utilize stakeholder expertise to enrich environmental studies. Facilitate discussions on design choices and possibilities. Engage in similar processes as the 'Involve' category.

Inform:

Agricultural Landowners, Local Schools, Tourism Board, Historical Preservation Society, Community Action Groups, Local Media, Recreational Clubs

Approach: Provide updates through newsletters, social media, and public meetings. Address potential disruptions during project execution. Maintain transparency and open communication channels.

Demands

- **Design Height:**
  - *Specific:* Design a dike with a minimum height of 5.70 meters NAP.
  - *Measurable:* Height measured from the surrounding land level of 7 meters NAP.
  - *Achievable:* Utilize existing land elevation data and construction techniques.
  - *Relevant:* Ensures adequate flood protection for the area.
  - *SMART demand:* The dike must be designed to reach a height of 5.70 meters NAP, considering the surrounding land elevation of 7 meters NAP.
- **Factor of Safety (FoS):**
  - *Specific:* Ensure the Factor of Safety (FoS) is between 1.3 and 1.5.
  - *Measurable:* FoS calculated using geotechnical analysis.
  - *Achievable:* Apply industry-standard safety calculations and materials.
  - *Relevant:* Maintains structural integrity and safety of the dike.
  - *SMART demand:* The dike design must achieve a Factor of Safety (FoS) between 1.3 and 1.5, verified through geotechnical analysis.
- **Outer Slope Gradient:**
  - *Specific:* Design the outer slope of the dike with a gradient of 1:3.
  - *Measurable:* Slope gradient measured during construction.
  - *Achievable:* Use standard construction practices for slope formation.
  - *Relevant:* Ensures stability and erosion control.
  - *SMART demand:* The outer slope of the dike must be constructed with a gradient of 1:3, ensuring stability and erosion control.
- **Cohesive Layer:**
  - *Specific:* Incorporate a cohesive layer of 0 meters beneath the dike.
  - *Measurable:* Layer thickness verified through soil analysis.
  - *Achievable:* Utilize existing soil conditions and engineering solutions.
  - *Relevant:* Ensures compatibility with the sandy substrate.
  - *SMART demand:* The dike design must account for a cohesive layer of 0 meters beneath the structure, compatible with the sandy substrate.

Explanation

The current design closely aligns with the desired specifications and stakeholder priorities. FoS is near the ideal range, and spatial efficiency is optimized, addressing core demands. Proceed to finish as further iterations may not yield significant improvements.

Design Dimensions

Parameter	Value	Unit
Dike Height	5.7	meter
Crest Width	4	meter
Inner Slope	1:3	-
Inner Berm Length	2	meter
Thickness Cohesive Layer	0	0 meter
Distance Toe to Ditch	2	meter
Ditch Depth	1.25	meter
Dike core	clay	-
FoS	1.3143	-
Water height	4.5	meter

Table 13: Final design dimensions of the Grebbedijk.

Design actual dike

based on factsheet Grebbedijk dijkvak 3D met assumptions van cohesive layer, crest width, en min ditch properties. Noticeable is then difference in FoS that was drawn from the metamodel. The actual dike seems to be overly stable.

Parameter	Value	Unit
Dike Height	5.7	meter
Crest Width	4	meter
Inner Slope	1:3 + 1:5	-
Inner Berm Length	5	meter
Thickness Cohesive Layer	0	0 meter
Distance Toe to Ditch	-	meter
Ditch Depth	-	meter
Dike core	clay	-
FoS	1.5523	-
Water height	-	meter

Table 14: Design dimensions of the actual Grebbedijk.

E Enquete results

This appendix shows the results from the expert elicitation questionnaire. For every question the answers are produced. As can be seen there is some disagreement on some questions, this is probably due to the fact that trust in AI systems differs from person to person.

1. How do you assess the flexibility of the agentic AI model to adapt to different design requirements and environmental factors?

   Table 15 shows the results from the expert elicitation on question 1. The average score was a 7.3.

Table 15: Results from the expert elicitation survey question 1.

Score	Amount of times rated
8	4
7	1
6	2

2. To what extent can the model easily process changes in input parameters without significantly affecting the integrity of the output? Table 16 shows the results from the expert elicitation on question 2. The average score was a 7.3.

Table 16: Results from the expert elicitation survey question 2.

Score	Amount of times rated
8	2
7	5

3. How do you assess the accuracy of the results presented by the agentic AI model? Table 17 shows the results from the expert elicitation on question 3. The average score was a 6.4.

Table 17: Results from the expert elicitation survey question 3.

Score	Amount of times rated
7	4
6	2
5	1

4. To what extent do you rely on the recommendations made by the model in the design and evaluation of dikes? Table 18 shows the results from the expert elicitation on question 4. The average score was a 6.0.

Table 18: Results from the expert elicitation survey question 4.

Score	Amount of times rated
8	1
7	1
6	4
3	1

5. How well are the logical steps and reasoning followed by the model in generating results? Table 19 shows the results from the expert elicitation on question 5. The average score was a 7.7.

Table 19: Results from the expert elicitation survey question 5.

Score	Amount of times rated
9	1
8	3
7	3

6. How well does the model recognize and address inconsistencies or contradictions in the design? Table 20 shows the results from the expert elicitation on question 6. The average score was a 6.4.

Table 20: Results from the expert elicitation survey question 6.

Score	Amount of times rated
8	2
7	2
5	3

7. To what extent do you think the agentic AI model contributes to accelerating the design process for dikes? Table 21 shows the results from the expert elicitation on question 7. The average score was a 9.0.

Table 21: Results from the expert elicitation survey question 7.

Score	Amount of times rated
10	2
9	3
8	2

8. How efficient is the use of resources (such as time and data) in the execution of the agentic AI model? Table 22 shows the results from the expert elicitation on question 8. The average score was a 7.3.

Table 22: Results from the expert elicitation survey question 8.

Score	Amount of times rated
10	1
9	1
8	4
4	1

9. To what extent do you expect the agentic AI model to be a valuable addition to the dike design process? Table 23 shows the results from the expert elicitation on question 9. The average score was a 8.6.

Table 23: Results from the expert elicitation survey question 9.

Score	Amount of times rated
10	1
9	2
8	4

10. To what degree do you think the insights generated by the agentic AI model will contribute to better decision-making in dike design? Table 24 shows the results from the expert elicitation on question 10. The average score was a 7.1.

Table 24: Results from the expert elicitation survey question 10.

Score	Amount of times rated
10	1
8	1
7	4
4	1

11. What do you consider to be the main improvement areas for the agentic AI model regarding flexibility, quality, logic, efficiency, or added value? Table 25 shows the results from the expert elicitation on question 11. This answers to the open questions are included.

Table 25: Comments and Recommendations for expert elicitation survey question 11.

No.	Comment
1	Everything hinges on how well the arguments are substantiated; are the choices logical and traceable?
2	It should be possible to assign certain parameters a fixed value or range that the model cannot exceed.
3	More accuracy by incorporating multiple variables. More room for creative alternatives. Both are not within your research, but as a recommendation.
4	We discussed: conduct a stakeholder analysis based on location. These will be placed on the stakeholder matrix and correspond with the four levels of participation. These levels can be linked to standard participation methods, allowing you to draft a general participation plan that must still be supplemented by an expert.
5	Logic
6	Integrate the logic of the current dike design process more accurately. The right logic leads to better quality and added value.