Iterative and Incremental Validation of Simulation Conceptual Models

Erkin Çilden, Ahmet Sezer, M. Haluk Canberi
STM Defense Technologies
Engineering and Trade Inc.
Ankara, Turkey
{erkin.cilden, asezer, hcanberi}@stm.com.tr

Halit Oğuztüzün
Middle East Technical University,
Department of Computer Engineering
Ankara, Turkey
oguztzn@ceng.metu.edu.tr

ABSTRACT

Model validation is a technically and managerially challenging task for simulation projects. One-shot validation of conceptual models is not practical for most simulation system projects due to the inherent logical complexity of individual models, interdependencies between models, and the diverse viewpoints of the stakeholders.

Fidelity, in particular, is an issue that permeates these characteristics. The required level of fidelity is largely determined by the expectations of the users and sponsors about the simulation system, e.g. the kinds of analyses that can be accomplished. Tacit assumptions that are implicit in the stakeholder viewpoints and underlie all modeling activities gradually become explicit through model validation activities. Moreover, harmonizing assumptions as they become explicit and clearly spelled out requires negotiations, which take multiple rounds of exchange. Therefore, stakeholders with diverse viewpoints often struggle to align on rules, boundaries, methods and constraints, and assumptions throughout the validation process.

Therefore, model validation, by nature, is a multiple-stakeholder activity, that requires a specifically defined process per project, so that all aspects of the activity, varying from stakeholder management to application of technical validation methods, can be executed in gradually increasing levels of agreement and harmony.

In this paper, we argue that the difficulties mentioned above can be relieved by the establishment and application of iterative and incremental approaches for model validation activities throughout a project. Specific practices have to depend on the specifics of the project, such as model fidelity requirements, availability of field data for validation activities, accessibility of subject matter experts, etc. Nevertheless, a simulation system development project can highly benefit from a custom-defined iterative and incremental process for model validation, inspired by corresponding software development principles.

ABOUT THE AUTHORS

Erkin Çilden obtained his B.S., M.S., and Ph.D. degrees in Computer Engineering from Middle East Technical University, and currently is a group leader and senior software engineer at STM. His research interests include multi-agent systems, reinforcement learning, and simulation software design.

Ahmet Sezer has a B.S. and M.S. in Mechanical Engineering from Middle East Technical University and currently is a group manager at STM. His expertise includes simulation modeling (weapon systems, flight dynamics) and accreditation.

M. Haluk Canberi has a B.S. in Electrical and Electronics Engineering from Hacettepe University and an MBA from Hacettepe University. He is currently the director of the RF and Simulation Systems Directorate at STM. His technical expertise includes avionics and simulation systems.

Halit Oğuztüzün is a professor in the Department of Computer Engineering at Middle East Technical University (METU), Ankara, Turkey. He obtained his BS and MS degrees in Computer Engineering from METU in 1982 and 1984, and his Ph.D. in Computer Science from The University of Iowa, Iowa City, IA, USA in 1991. His recent research activities include distributed simulation and model-driven engineering.

Iterative and Incremental Validation of Simulation Conceptual Models

Erkin Çilden, Ahmet Sezer, M. Haluk Canberi
STM Defense Technologies
Engineering and Trade Inc.
Ankara, Turkey
{erkin.cilden, asezer, hcanberi}@stm.com.tr

M. Halit Oğuztüzün
Middle East Technical University,
Department of Computer Engineering
Ankara, Turkey
oguztzn@ceng.metu.edu.tr

INTRODUCTION

A simulation conceptual model is an integral part of the simulation software, on which more than one expertise is joined. It consists of software components, thus requiring the application of a few software engineering practices. On the other hand, it is an imitation of a real-life phenomenon, which requires a thorough knowledge of the phenomenon itself, which is usually expertise in that subject matter. Usually, in addition to these design and implementation specialties, other roles also exist that have something to say about the simulation conceptual model, stakeholders such as the sponsor of the project, the end-user of the final product, the quality assurance experts, etc.

The simulation conceptual model is the common ground on which all stakeholders and developers communicate. This paper follows the definition of Pace which interprets the conceptual model from the simulation developer's perspective and involves any information that describes a simulation developer's concept about the simulation and its pieces (Pace, 2000). Therefore, the terms "conceptual model" and "simulation conceptual model" are used interchangeably in this paper. Related literature states that the model validation process would better be integrated with model development activities, and a notion of iteration is required. Starting from the model requirements, the validation process triggers arguments on the conceptual model and its fidelity. Repeated loops of design, development, verification, and validation are known to promise mature simulation conceptual models and related implementations earlier.

In this paper, we argue that validation practices could take advantage of the best practices from iterative and incremental software development methods, especially agile methods. Separation of validation activities from design and development activities may help the model validation team be isolated from model development concerns, therefore be more focused on validation concerns, such as calibration of fidelity. Obviously, model validation is a multiple-stakeholder activity, and existing agile practices have already addressed fruitful ways to involve stakeholders in the decision-making process. From our experience, the application of agile practices for conceptual model validation may introduce a number of advantages, as well as some implementation challenges.

RELATED WORK

Validation of Simulation Conceptual Models

It is a challenging task to define, apply, and orchestrate the methods and processes to decide that a conceptual model will execute correctly when implemented, and represents its source phenomenon properly. The challenge originates from the vagueness of the terms "correct execution" and "proper representation." Fortunately, various methods exist to verify the model implementations and validate the conceptual models for correct execution and proper representation, respectively. More formally, model verification, or simulation verification, is defined as "ensuring that the computer program of the computerized model and its implementation are correct" and model validation is defined as "the substantiation that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model" (Sargent, 2020).

Verification of simulation models has a long history in concurrence with the software testing literature (Fairley, 1976). Typically, there are two ways to verify a model. Static testing is concerned with the structural correctness and content integrity of the simulation software, and methods like structured walkthroughs, static analyses, and source code reviews can be used. Dynamic testing, on the other hand, deals with the run-time behavior of the model. Various methods exist for dynamic testing, such as tracing, animation, etc (Sargent, 2020).

In principle, simulation verification is well-founded if the conceptual model is known to be valid. However, this is rarely the case, since the validity of a model almost always remains a subject of debate among stakeholders throughout the simulation development. Meanwhile, a verified implementation is usually a prerequisite for simulation conceptual model validation, as well. Therefore, the model validation requires additional attention, as there are even more vague terms in the aforementioned definition of model validation, such as “satisfactory range of accuracy” and “intended application of the model.” The problem here is that it is usually impossible to define a quantitative or qualitative representation of “satisfactory” and “intention.” This problem is already known in the literature and has been extensively studied for decades (Balci, 1994; David, 2013; Law, 2008).

What makes model validation difficult to manage is that validity is strongly tied to the notions of credibility and confidence, which involve judgements of the stakeholders that contain subjective elements. In order to better cope with this challenge, the related literature differentiates the concept and the operation of a model. Conceptual validation of a model is concerned with whether the theories and assumptions underlying the model abstraction are correctly and reasonably set for the intended use of the model, while operational validation focuses on the behavioral accuracy of the model with respect to its intended use. There is an intermediate activity, referred to as “computational/computerized verification,” which focuses on the correct implementation of the conceptual model (Sargent, 2020). In practice, however, the conceptual model is the ground on which the whole model validation and verification activity is constructed. In other words, at the end of the day, if there is no conceptual model on which all stakeholders hand-shake, it is not meaningful to discuss operational validity or computational accuracy. Therefore, any model validation activity inherently addresses its conceptual validation.

In both cases, a model’s intended use is extremely critical for the validation effort, in other words “a model is only validated with respect to its purpose” (Robinson, 1997). A model’s intended use is not precisely delineated most of the time, especially where many models exist in the simulation interacting with each other. Even at the initial stages of the project, an agreement on the intended use drives the fidelity of the model, which is an important characteristic that guides the subsequent design steps.

The fidelity of a simulation conceptual model is the expected uncertainty in the model implementation (Chaudhari et al., 2022). Model fidelity is directly affected by its intended use, and its level is very challenging to define and agree on. Therefore, most of the time, fidelity requirements are described at a higher level, since otherwise, it is highly probable that requirements do not align with the intended use, or are conflicting with each other. For this reason, it may not be realistic to define model validation requirements upfront, due to the vague nature of model fidelity.

Validation as a part of the Simulation Model Development Process

A simulation model is a multiple-aspect abstraction of reality. A model should be designed by taking into account at least three aspects of abstraction: domain, usage, and computation. For example, a radar model, when completed, is a fusion of expertise on the radar domain, user’s intentions, and expertise on the capability and limitations of available digital technologies, as a bare minimum. Depending on the context, domain, intended use, and limitations, the number of aspects is likely to increase. If the radar model is requested to be affected by atmospheric conditions, knowledge of environmental modeling would be necessary, which would be a new aspect of the radar model. The number of models, their interrelations, and computational complexity may be a burden if the simulation has some performance requirements, such as real-time or faster-than-real-time execution, which is another aspect of the radar model concerning software architecture. The sponsor of the project may dictate some resource constraints for radar model development, such as limited access to referent data. Even additional internal stakeholders, such as owners of different projects, may have concerns about the reusability of the radar model within their projects. So, at least three stakeholders should be working on the same model for validation, which is a task having its own technical, social, and managerial challenges.

Therefore, model development activities are essentially of multi-stakeholder nature due to the multiple-aspect abstraction phenomenon. Consequently, model validation activities turn out to focus on elevating stakeholders’ confidence, while maintaining the credibility of the models throughout the project. That is why it is advised to conduct model validation activities (together with simulation verification activities) as an integral part of the project development lifecycle (Balci, 1994, 2012). There are various methods for model validation (and verification), but selecting and tailoring methods needs to be carefully orchestrated (Roungas et al., 2018). Nevertheless, the best practices in the literature guide us to extend our simulation development practices with model validation activities, forming a unified model generation process (Fonseca I Casas, 2023).

Iterative and Incremental Software Engineering

Generations of software engineering researchers and industry leaders have been studying effective ways of developing software. Different sets of rules and collections of best practices for software development are available in the literature for decades, referred to as a "software development lifecycle." Managing software development is becoming more and more challenging since software systems are getting bigger in size, more complex in behavior and more adaptable to change. Consequently, software development processes have evolved to include more agile, iterative, and incremental practices which are empirically proven to better cope with the complex nature of this problem (Papatheocharous & Andreou, 2014; Serrador & Pinto, 2015).

Iterative and incremental approaches, and recently agile methodologies, aim to deliver high-quality software by non-traditional practices such as repetition of short development cycles, customer/user involvement, incremental delivery, and responsiveness to changing user requirements (Matharu et al., 2015). Although specifics of the applied methodology vary, frequent delivery and stakeholder involvement are common features of any iterative and incremental approach.

Stakeholder Involvement in Software Engineering

It is accepted that a software development project is never solely a technical activity. It involves various communication and coordination challenges, such as the interaction of team members among themselves, as well as negotiations with the stakeholders outside the development team. Modern software development practices blend technical methods with social practices, favoring the collective agreement on expected artifacts over individual technical outcomes.

Stakeholder involvement is a crucial aspect of software engineering projects, ensuring that their needs and expectations are addressed effectively. Evidently, the successful delivery of software engineering projects is directly related to the involvement of stakeholders in the decision-making process within the various stages of the project (McManus, 2004). Even in the very early stages of a software development project, stakeholder involvement is critical (Hujainah et al., 2018; Khan et al., 2022). Even for non-technical aspects, such as software project risk management, stakeholder involvement is known to be critical (Vrhovec et al., 2015).

Agile software development practices positively impact software project success, by encouraging interaction with relevant stakeholders. To achieve this, establishing effective communication channels, conducting regular stakeholder meetings, and incorporating feedback throughout the development process are crucial. Since today's software systems are complex and evolving, responsiveness to changes in user requirements via stakeholder involvement is practiced and known to be effective (Matharu et al., 2015).

Stakeholder Involvement in the Validation of Conceptual Models

As pointed out earlier, simulation conceptual model validation is a multiple-stakeholder activity, tightly coupled with the notion of model confidence. Model confidence refers to a decision-maker's belief in the expected outcomes when using a model in a specific situation. "Convincing other stakeholders to accept the results of these tools can be a challenge, and indeed, the adoption of models and simulations at the level of system development still lags the pace of the underlying computational and application advances. The acceptance of models and simulations remains largely a function of the subjective preferences of engineers and other stakeholders" (Chaudhari et al., 2022). Different aspects of the process, namely model-related, modeler-related, and stakeholder-related factors, need to be orchestrated toward a mature, agreed upon, and usable conceptual model.

Although no commonly agreed upon recommended practices exists regarding how to align model development activities with stakeholder involvement, promising attempts exist such as linking simulation modeling to the systems engineering process (Van Der Zee, 2012), direct stakeholder involvement in the model development process (David, 2013; Voinov & Bousquet, 2010), and argument-based approaches (Saam, 2019).

ITERATIVE AND INCREMENTAL CONCEPTUAL VALIDATION PROCESS

Problem Definition

As we discussed earlier, any activity related to model validation has the potential to impact the conceptual model in question, including verification methods and operational validation techniques. Consequently, the validation process is subject to frequent changes, making it challenging to define upfront criteria. This is due to the unique focus of each conceptual model, and when these models are employed together, their emergent behavior may differ from expectations, necessitating iterative adjustments to model fidelity as well as model interfaces and shared data such as environment. Thus, treating conceptual model validation, simulation verification, and simulation validation as sequential tasks is generally not recommended, as the model development team initiates design decisions while involving other stakeholders as critical decision-makers in a repetitive and multi-stakeholder process.

These challenges mirror those encountered in software development lifecycles, where freezing requirements in advance is impractical, feature details evolve during development, and stakeholders beyond developers play significant decision-making roles.

Recommended Practice

Thus, it might be useful to transfer some practices from iterative and incremental software development lifecycle rules into simulation validation activities. To do so, we argue that the following recommended practices might help:

- *Separate model development and model validation processes*: To ensure a clear distinction between model development and model validation, treat them as separate projects or processes. While aligning their flow, keep them as independent of each other as possible by managing them as different teams. Establish mechanisms for triggering tasks and managing interactions between the two processes, while maintaining their separation. This separation allows each team to focus on their specific objectives and responsibilities while still collaborating effectively.
- *Change the mindset*: Shift the perception of model validation from being merely a series of method applications, discussions, and meetings attached to the development process. Instead, consider model validation as a work product the model validation team delivers, involving all relevant stakeholders. By considering it as a work product, it becomes easier to promote a more dedicated approach toward validation and foster a sense of ownership and responsibility within the model validation team.
- *Define the validation process*: At the beginning of the project, collaborate with stakeholders to define the validation process in enough detail. Engage in discussions to clarify the steps, activities, and expectations involved in the process. Document the decisions and establish rules and guidelines to ensure consistency and clarity. While the rules can be adjusted based on best practices throughout the project, it is essential to manage changes in a controlled and deliberate manner. A well-defined validation process enables transparency, accountability, and effective communication among validation team members.
- *Embrace an iterative and incremental approach*: Opt for an iterative and incremental approach in the defined validation process, preferably leveraging agile practices, such as Scrum (Schwaber & Sutherland, 2020). By breaking down the validation activities into smaller increments, you can achieve frequent deliveries and demonstrate progress at regular intervals. This approach allows for the early identification of issues and facilitates a more flexible and adaptive validation process. This also encourages continuous improvement and collaboration among team members.
    - *Define the delivered work product*: Clearly define the expected outcome or work product of the validation process. This could be in the form of specific validation reports, metrics, or other tangible deliverables that demonstrate the validity and quality of the simulation. Since the validation methods vary, select the deliverables accordingly. Ensure that the defined product is measurable and can be evaluated against predetermined criteria. Having a clear understanding of the desired deliverable enables the validation team and stakeholders to align their efforts toward achieving the intended objectives.
    - *Establish a "progress" measure*: Create a mechanism to measure the progress within the validation process. Define key metrics or indicators that reflect the advancement of validation activities. This allows for effective monitoring and evaluation of the validation process, providing insights into its overall status and identifying areas that may require further attention or improvement. Establishing a progress measure enables the validation team to track their achievements and make data-driven decisions to drive the process forward.
    - *Define how to trigger corrective actions*: Determine specific triggers and actions to be taken in case the validation process fails to progress as expected. Identify thresholds or milestones that,

  if not met, indicate the need for corrective measures. Establish protocols for identifying bottlenecks, addressing issues, and initiating appropriate actions to mitigate risks and ensure forward momentum. Proactively addressing deviations from the expected progress helps maintain the integrity and effectiveness of the validation process.
  - *Transfer other useful practices*: Introduce relevant iterative and incremental, or agile practices into the model validation process. For example, early integration involves incorporating validation activities as early as possible in the development lifecycle. Feature-driven development focuses on validating specific features or aspects of the model in a systematic manner. Maintaining a routine helps establish regular validation cycles and checkpoints. Measuring team pace allows for monitoring the productivity and efficiency of the validation team. By adapting these practices, you can enhance collaboration, flexibility, and continuous improvement within the model validation process.
- *Consider project constraints*: Take into account the available project timespan and resource constraints when defining the validation process. Consider factors such as meeting schedules, review timelines, and availability of resources for validation-related activities. Align the validation process with the overall project plan to ensure a realistic and feasible approach. By considering project constraints, you can effectively allocate resources, manage timeframes, and optimize the validation process within the given project context.

PRACTICAL ISSUES: A CASE STUDY

Naval Warfare Effectiveness Analysis Model (GEMED) is an multi-level analysis simulation system in which combat activities of combat platforms can be measured in an integrated synthetic tactical environment (Çilden et al., 2021). Approximately 90 generic system models had been developed by 3 different contractors.

Project Phases and Simulation Conceptual Models

Model requirements were partly driven by the technical specifications; they evolved as an outcome of several meetings with stakeholders (to name a few; end-users, customers, sponsors, domain experts, quality experts) where system requirements and intended use were discussed.

Simulation conceptual models were iteratively developed and documented in the project's Conceptual Model Document, as the primary artifact of the project. Every conceptual model is documented at least by its state transition logic (using UML State Machine Diagrams), its interaction with other models (using UML Component Diagrams with information flows), assumptions and computation primitives. Additional model specific notations were also used, such as pseudocodes, formulas and drawings. The project consisted of two major phases: In the first phase, the primary objective was developing a fully-functional prototype, with a limited number of models lacking comprehensive validation. In the second phase, the focus was on development of all models, their interactions, and their validation.

Although the preliminary revisions of the Conceptual Model Document were released in the first phase, the objective was that all project members can understand and use it for their specific needs in project roles. Moreover, information flow between simulation entities (models), model states and accepted computation primitives (mathematical and physical calculations) were shared with all stakeholders. The Conceptual Model Document had 8 revisions during the project life time, a span of 5 years, including the prototyping phase. All discussions regarding model fidelity and validity of use were captured in this document.

After reaching certain software maturity within the project timeline, the first prototype of GEMED was delivered with the main platform, system, and mission models, together with an intermediate revision of the Conceptual Model Document. This functional prototype was of crucial importance for the project, as the stakeholders were then able to see the models, behaviors, and scenarios in action. The stakeholders were then also able to align the simulation execution with the draft conceptual models before any comprehensive validation activity.

Recommended practice: Separate model development and model validation processes

In a project like GEMED where the model spectrum is very large, it is challenging for all parties to ensure that all the requirements, fidelity needs, and model principles are covered. Upon completion of the functional prototype, all stakeholders were more confident and ready to discuss various aspects of conceptual models, including abstraction levels, features, and computational needs. Therefore, a validation team was established including system engineers from the contractors' side and representatives of the authority of need and the customer.

Depending on the specific needs, domain experts and model designers were also involved from time to time, and contributed to the validation process.

Recommended practice: Change the mindset
In such a team, with multiple authorities and decision makers, repetitive discussions, briefings, and summary sessions are required to align each member's understanding of the current status, team objectives, and plans. Additionally, synchronization, and coordination with development team(s) comes with an additional management cost. For this purpose, a defined validation process helped the team to ensure they are right on track. This helped all members of the validation team to focus on the validation process, and its expected outcomes, instead of other aspects of the whole project, such as user experience, visual details, implementation/design issues.

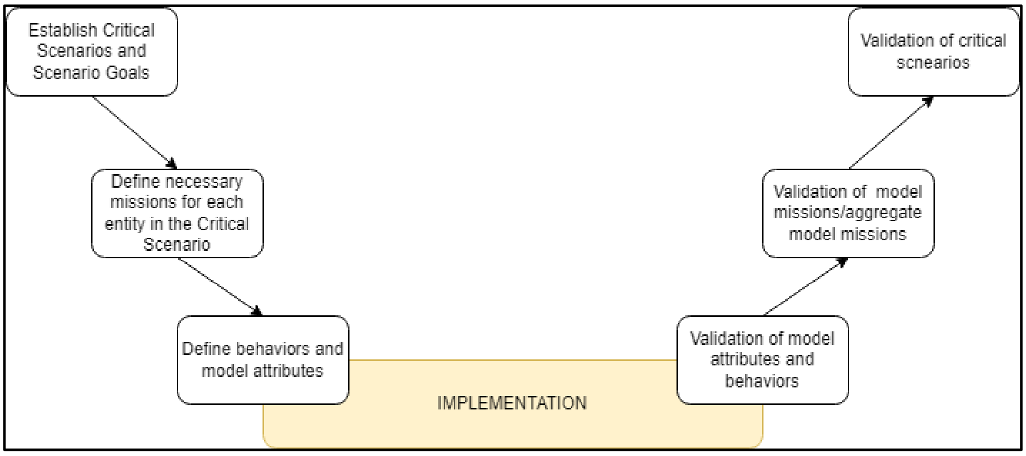

Establish Critical Scenarios and Scenario Goals
Define necessary missions for each entity in the Critical Scenario
Define behaviors and model attributes
IMPLEMENTATION
Validation of model attributes and behaviors
Validation of model missions/aggregate model missions
Validation of critical scnearios

Figure 1 The outline of GEMED defined model validation process, inspired by the V-model.

Recommended practice: Define the validation process
A defined process for model validation was developed and documented, and is shared with and reviewed by all stakeholders. According to this defined process, which is specific to GEMED project needs, validation process starts with the definition of every model attribute according to intended use, following a top down approach. The model features were determined taking into account the higher level task plans and model behaviors within the simulation. Reaching a common agreement on the model features, scenarios were defined as the next step to be used in validation of model attributes. At this level, the process resembles the right hand side of the well-known V-model, where the features, scenarios, and attributes are at the left side of the diagram while the validation activities are at right side (Figure 1).

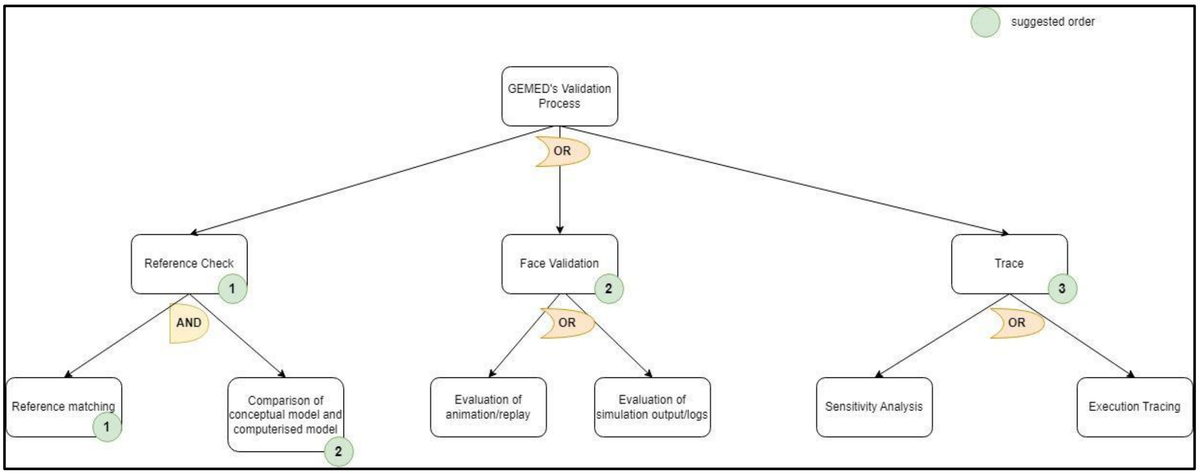

suggested order
GEMED's Validation Process
OR
Reference Check
1
AND
Reference matching
1
Comparison of conceptual model and computerised model
2
Face Validation
2
OR
Evaluation of animation/replay
Evaluation of simulation output/logs
Trace
3
OR
Sensitivity Analysis
Execution Tracing

Figure 2 GEMED model validation flow. Implementation of models is a separate process external to the model validation activities.

Recommended practice: Embrace an iterative and incremental approach
Starting with the second phase of the project, routine validation meetings with the validation team were carried out, trying to keep a constant rhythm. All findings, decisions, tasks were recorded in meeting minutes, and status of each model was tracked and shared with all stakeholders. Validation methods were applied more than once per

model, especially when a major change occurs in model implementation after which all required activities are repeated from the beginning.

Recommended practice: Define the delivered work product

For the validation team, the work product was defined as a number of simulation scenarios per model, specifically designed for model validation, that are capable of demonstration and analysis of all the model features, covering conceptual aspects of the model such as state transitions, behaviors, attributes, interactions, outputs. One important motive for this decision is the repeatability of simulation scenarios, which is extremely useful in automated tests, especially for regression analysis purposes. Moreover, taking a simulation scenario as an anchor, it is easier to stay focused while the set of selected validation activities are being carried out, as summarized in Figure 2.

Recommended practice: Establish a "progress" measure

The validation team kept track of two levels of progress: number of successful model features, and number of successful validation scenarios. Each successfully completed validation activity throughout a simulation scenario yields a set of validated features, as well as the validated scenario itself. Both are tangible measures of progress (Figure 3), and it is also possible to repeat the validation process for a given model upon a major change, in order to identify any possible regression.

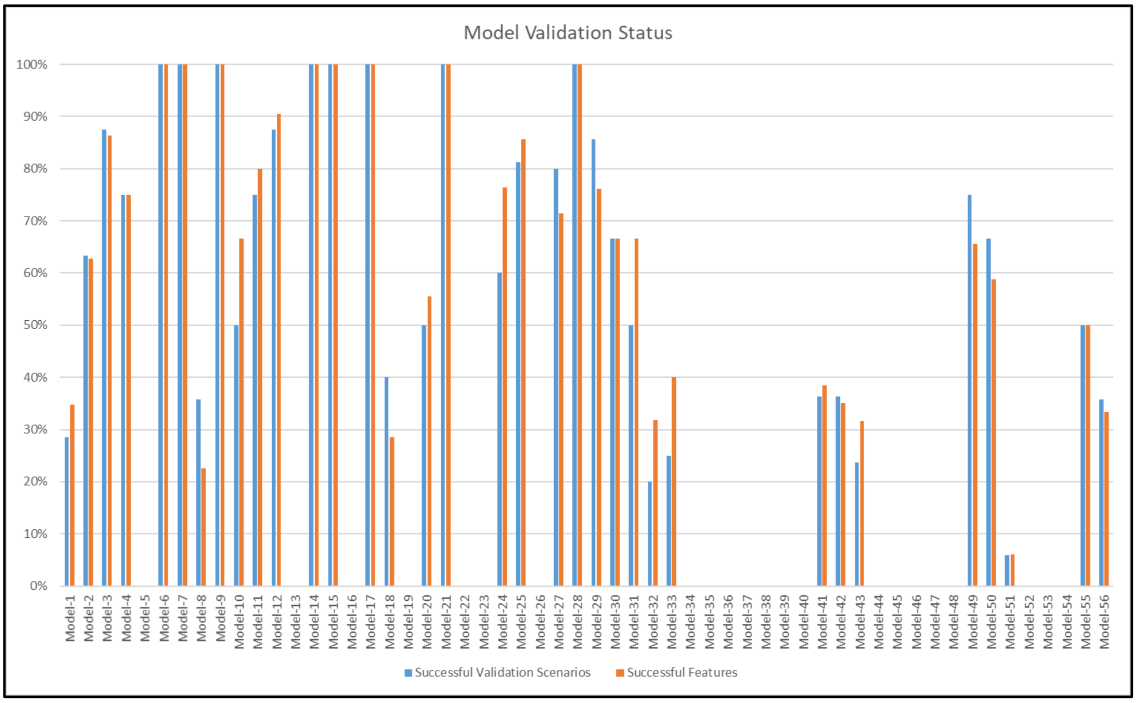


Figure 3 An intermediate snapshot of GEMED model validation measurements, based on the rate of successful validation scenarios and successful features.

Recommended practice: Define how to trigger corrective actions

Most important two flags to trigger corrective actions were (1) falling behind schedule, (2) regression due to a cross-cutting change. It was usually straightforward to define trigger metrics by keeping track of the established progress measure. A major challenge was to re-schedule the validation activities, since participation of multiple institutions required rigorous planning and coordination.

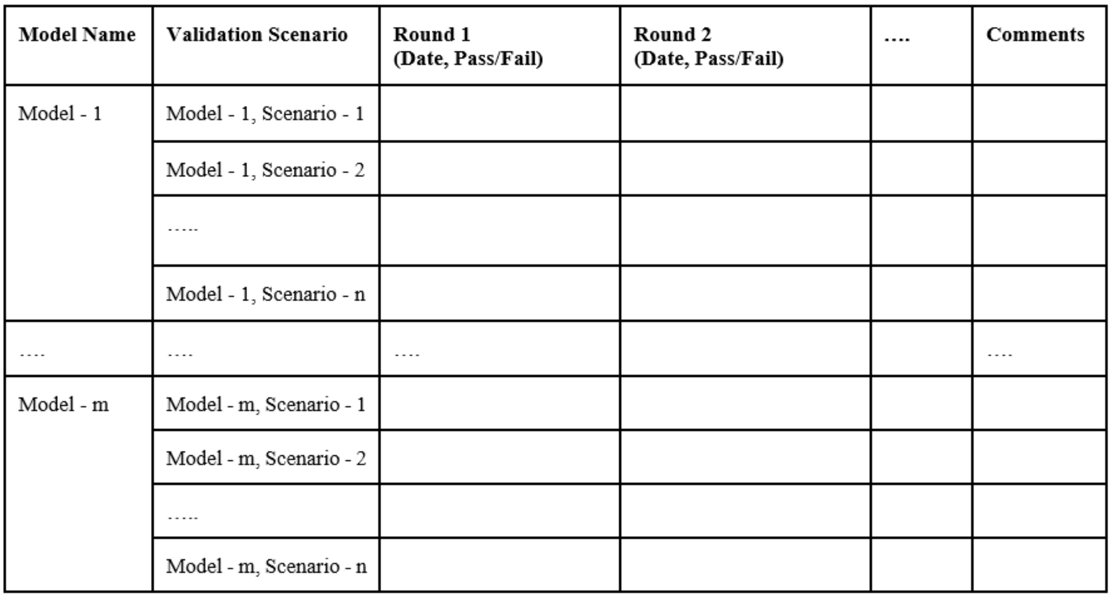
Recommended practice: Transfer other useful practices

Trying to keep a routine and pace was the practices the validation team tried to transfer from agile practices. Due to the more formal nature of the meetings it was not possible to transfer some pace-maker practices such as daily stand-ups, but a weekly routine helped all participants to get ready and prepared for the activities planned for the week.

Recommended practice: Consider project constraints

For GEMED, the number of models to validate was the major constraint that should be taken into account. For this reason, the selected validation methods, their order and preferences are kept as concise as possible (Figure 2).

Model validation activities started roughly 2 years prior to final delivery of the GEMED product. The validation team came together every week, at minimum, and discussed model attributes, scenarios to be studied for model attributes, and the findings from execution of validation scenarios and provided comments and notes to the development team. The main validation dashboard for the validation team was a basic table shown in Figure 4; where GEMED software revision and meeting notes were tracked in meeting notes.



Model Name	Validation Scenario	Round 1 (Date, Pass/Fail)	Round 2 (Date, Pass/Fail)	….	Comments
Model - 1	Model - 1, Scenario - 1				
	Model - 1, Scenario - 2				
	…..				
	Model - 1, Scenario - n				
….	….	….			….
Model - m	Model - m, Scenario - 1				
	Model - m, Scenario - 2				
	…..				
	Model - m, Scenario - n				

Figure 4 The model validation status tracking table

All the findings from the validation meetings recorded, classified and reviewed before they were passed to the development team. More than 300 model feature validation scenarios were produced and after being agreed on the scenario they are automated with use of automation scripts and deployed after every software release in order to provide regression metrics.

Discussion on the Impacts of the Applied Process

- From the company's perspective, the applied process helps the timely delivery of simulation conceptual models (as well as their validated software implementation). It also enables a deeper understanding of stakeholder needs with sound reasoning, thereby elevating all stakeholders' awareness.
- On the end-users' side, the applied process enhances their confidence in understanding the purpose and capabilities of the models during execution. Additionally, end-users gain knowledge of what a model cannot do before accepting it, resulting in time and effort savings. The cooperative experience between stakeholders has a positive impact in that sense.
- Adapting mindsets and convincing stakeholders, reaching a common ground and establishing a shared language, given the diverse professions involved, can be time-consuming.
- Stakeholders, both internal and external, may struggle to consistently keep the common goals in mind, sometimes deviating from the validation track to question unrelated aspects (such as an animation detail of the system/phenomenon), which are usually quite natural requests out of the validation context, but are not necessary for the intended use of the model.
- Maintaining and tracking the set of metrics that indicate the progress of the validation process can be difficult and time-consuming. Tool support, such as tools for traceability, might help.
- The formal acceptance process runs rather smoothly thanks to the mutual understanding that arose from the applied process. They become more focused on higher-level concerns, such as post-execution analysis and mission-level accuracy, rather than getting caught up in model specifics.

CONCLUSIONS AND FUTURE WORK

In this paper, we argue that simulation conceptual model validation activities can benefit from agile software engineering practices, especially in terms of increased stakeholder involvement, better focus on specific objectives, elevated stakeholder ownership, transparency, effective communication, continuous improvement, and the capability of the project team to deal with uncertainties and changes. The related literature already upholds iterative management of model validation, by suggesting to extend the existing system/model development process with the model validation activities. One of the specific recommendations of this paper is to separate model validation and model development processes, and ensure that model validation process is a defined one, on its own.

Another aspect discussed in the paper is that, since the conceptual model is the artifact that is agreed upon with the stakeholders, any model validation activity has the potential to affect the conceptual model. Therefore, even if some referents (i.e. ground truth data to be referred for judging the accuracy of simulation outputs) can be set, it is practically difficult to promise upfront validation requirements, due to the complex nature of model interactions. Changing requirements, together with late stakeholder involvement, are also one of the challenges of the software development lifecycle management, and there is an increasing attention on iterative incremental software engineering practices, especially agile methods, which promise to better cope with these challenges. Our second recommendation is to manage a model validation process with some practices transferred and tailored from agile methods, such as feature driven development, early delivery, setting up a study routine, and measurement of progress.

These recommendations stem from our project specific observations and fusion of knowledge from existing literature on the subject matter. However, it is important to support them with the outputs of further case studies, using more formal approaches such as experimentation, questionnaires, and analysis of long term metrics, which is identified as the future work of the paper.

ACKNOWLEDGEMENTS

This work is supported by the Scientific and Technological Research Council of Türkiye (TÜBİTAK) under Grant No. 115A044. The authors would like to thank Büşra Yılmaz and Fatih Narman for their contributions and key roles in the model validation process throughout the project. The authors are also grateful to the GEMED project team for their patience and effort.

REFERENCES

Balci, O. (1994). Validation, verification, and testing techniques throughout the life cycle of a simulation study. *Annals of Operations Research*, *53*(1), 121–173. https://doi.org/10.1007/BF02136828

Balci, O. (2012). A life cycle for modeling and simulation. *SIMULATION*, *88*(7), Article 7. https://doi.org/10.1177/0037549712438469

Chaudhari, A. M., Rebentisch, E., & Rhodes, D. H. (2022). Confidence in Models and Simulations: A Multi-Stakeholder Analysis. In B. R. Moser, P. Koomsap, & J. Stjepandić (Eds.), *Advances in Transdisciplinary Engineering*. IOS Press. https://doi.org/10.3233/ATDE220636

Çilden, E., Sezer, A., & Canberi, H. (2021). Agent based simulation of naval tactics with effectiveness analysis features. *The Interservice Industry, Training, Simulation, and Education Conference (I/ITSEC) Proceedings*. The Interservice Industry, Training, Simulation, and Education Conference, Orlando, FL. https://www.xcdsystem.com/iitsec/proceedings/index.cfm?Year=2021&AbID=96477&CID=862#View

David, N. (2013). Validating Simulations. In B. Edmonds & R. Meyer (Eds.), *Simulating Social Complexity* (pp. 135–171). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-93813-2_8

Fairley. (1976). Dynamic Testing of Simulation Software. *Proceedings of the 1976 Summer Computer Simulation Conference*, 40–46.

Fonseca I Casas, P. (2023). A Continuous Process for Validation, Verification, and Accreditation of Simulation Models. *Mathematics*, *11*(4), 845. https://doi.org/10.3390/math11040845

Hujainah, F., Bakar, R. B. A., Abdulgabber, M. A., & Zamli, K. Z. (2018). Software Requirements Prioritisation: A Systematic Literature Review on Significance, Stakeholders, Techniques and Challenges. *IEEE Access*, *6*, 71497–71523. https://doi.org/10.1109/ACCESS.2018.2881755

Khan, F. M., Khan, J. A., Assam, M., Almasoud, A. S., Abdelmaboud, A., & Hamza, M. A. M. (2022). A Comparative Systematic Analysis of Stakeholder's Identification Methods in Requirements Elicitation. *IEEE Access*, *10*, 30982–31011. https://doi.org/10.1109/ACCESS.2022.3152073

Law, A. M. (2008). How to Build Valid and Credible Simulation Models. *2008 Winter Simulation Conference (WSC)*, 39–47.

Matharu, G. S., Mishra, A., Singh, H., & Upadhyay, P. (2015). Empirical Study of Agile Software Development Methodologies: A Comparative Analysis. *ACM SIGSOFT Software Engineering Notes*, *40*(1), 1–6. https://doi.org/10.1145/2693208.2693233

McManus, J. (2004). A stakeholder perspective within software engineering projects. *2004 IEEE International Engineering Management Conference (IEEE Cat. No.04CH37574)*, *2*, 880–884. https://doi.org/10.1109/IEMC.2004.1407508

Pace, D. K. (2000). Conceptual model development for C4ISR simulations. *Proceedings of the 5th International Command and Control Research and Technology Symposium*, 24–26.

Papatheocharous, E., & Andreou, A. S. (2014). Empirical evidence and state of practice of software agile teams. *Journal of Software: Evolution and Process*, *26*(9), 855–866. https://doi.org/10.1002/smr.1664

Robinson, S. (1997). Simulation model verification and validation: Increasing the users' confidence. *Proceedings of the 1997 Winter Simulation Conference*.

Roungas, B., Verbraeck, A., & Meijer, S. (2018). *A Framework for Optimizing Simulation Model Validation & Verification*.

Saam, N. J. (2019). The Users' Judgements—The Stakeholder Approach to Simulation Validation. In C. Beisbart & N. J. Saam (Eds.), *Computer Simulation Validation* (pp. 405–431). Springer International Publishing. https://doi.org/10.1007/978-3-319-70766-2_17

Sargent, R. G. (2020). Verification And Validation Of Simulation Models: An Advanced Tutorial. *2020 Winter Simulation Conference (WSC)*, 16–29. https://doi.org/10.1109/WSC48552.2020.9384052

Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide*. https://scrumguides.org/

Serrador, P., & Pinto, J. K. (2015). Does Agile work? — A quantitative analysis of agile project success. *International Journal of Project Management*, *33*(5), 1040–1051. https://doi.org/10.1016/j.ijproman.2015.01.006

Van Der Zee, D.-J. (2012). An integrated conceptual modeling framework for simulation-Linking simulation modeling to the systems engineering process. *Proceedings Title: Proceedings of the 2012 Winter Simulation Conference (WSC)*, 1–12. https://doi.org/10.1109/WSC.2012.6465180

Voinov, A., & Bousquet, F. (2010). Modelling with stakeholders. *Environmental Modelling & Software*, *25*(11), 1268–1281. https://doi.org/10.1016/j.envsoft.2010.03.007

Vrhovec, S. L. R., Hovelja, T., Vavpotič, D., & Krisper, M. (2015). Diagnosing organizational risks in software projects: Stakeholder resistance. *International Journal of Project Management*, *33*(6), 1262–1273. https://doi.org/10.1016/j.ijproman.2015.03.007