Naval Information Warfare Center PACIFIC

TECHNICAL REPORT 3343
April 2024

Capturing Model-based Risk Management in an Agile Scrum Environment

Francesca V. Mirabile
Parker C. Suggs
Jeff H. Waters
NIWC Pacific

Naval Information Warfare Center (NIWC) Pacific
San Diego, CA 92152-5001

This page is intentionally blank.

TECHNICAL REPORT 3343
APRIL 2024

Capturing Model-based Risk Management in an Agile Scrum Environment

Francesca V. Mirabile
Parker C. Suggs
Jeff H. Waters
NIWC Pacific

Administrative Notes:
This report was approved through the Release of Scientific and Technical Information (RSTI) process in July 2023 and formally published in the Defense Technical Information Center (DTIC) in April 2024.

NIWC Pacific
San Diego, CA 92152-5001

NIWC Pacific
San Diego, California 92152-5001

P. M. McKenna, CAPT, USN
Commanding Officer

M. J. McMillan
Executive Director

ADMINISTRATIVE INFORMATION

The work described in this report was performed by the Command & Control (C2) Technology and Experimentation Division (specifically Branch Codes 53629 and 53622) of the Command & Control and Enterprise Engineering (C2E2) Department, Naval Information Warfare Center (NIWC) Pacific, San Diego, CA.

Released by
Emily Nguyen, Division Head
Command & Control (C2) Technology and Experimentation

Under authority of
Scott Crellin, Department Head
Command & Control and Enterprise Engineering (C2E2)

ACKNOWLEDGMENTS

We would like to acknowledge the support of our colleagues and the leadership in our branches, division, and department.

Editor: MGK

EXECUTIVE SUMMARY

This paper describes a reusable approach for managing risk based on model-based systems engineering (MBSE) in a modern agile scrum environment. Software systems are often built with agile teams based on short sprints of user-defined tasks, with continuous customer feedback incorporated. Although the agile process is intended to address some forms of risk, software development is complex and error-prone. Can we do better? Many forward-thinking organizations develop models of systems to focus on simpler representations to design and test concepts before expending the time and cost of building the system. How do we model risk with MBSE? How do we integrate MBSE risk management with agile risk management?

The authors address these questions based on key principles of risk management, agile, and model-based software environments to derive an integrated approach that is simple, reusable, practical, and organic. The approach addresses four key steps of risk management:

1. Identifying the risk or opportunity.
2. Analyzing the risk and its impacts.
3. Responding by developing mitigation or enhancement tasks.
4. Integrating risk monitoring and management into daily processes.

Using the Unified Architecture Framework (UAF), the authors develop a reusable template for representing risk in an MBSE model. This report presents the template, the design of the data schema and entry forms, and a process for using them. The goal is to ensure that risk is defined, communicated, and managed organically and seamlessly during the entire development process.

The authors present the approach in the following sections:

- **Section 1 – Introduction and Definitions:** Defining Risk, Opportunity, MBSE, and Agile
- **Section 2 – Principles of MBSE-Driven Risk Management in an Agile Scrum Environment:** Identifying guideposts of consistency and truth to inform the approach
- **Section 3 – Approach Solution:** A template and schema for defining and representing risk in an MBSE environment
- **Section 4 – Integrating Risk Management into Agile Scrum Process:** How the template fits into and can drive the Agile Process
- **Section 5 – Conclusion**

Keywords: Risk Management, Agile, Model-Based System Engineering, MBSE, Software Development

This page is intentionally blank.

ACRONYMS

ACS	agile core services
DoD	Department of Defense
DoDAF	Department of Defense Architecture Framework
JSON	javascript object notation
JSON-LD	javascript object notation - linked data
MBSE	model-based systems engineering
NAVWAR	Naval Information Warfare Systems Command
NIWC Pacific	Naval Information Warfare Center Pacific
REST	representational state transfer
ROME	Risk Opportunity Management E-template
UAF	Unified Architecture Framework
URL	uniform resource locator
VAULTIS	visible, accessible, understandable, linked, trustworthy, interoperable, and secure
W3C	World Wide Web Consortium

This page is intentionally blank.

CONTENTS

FIGURES

TABLES

This page is intentionally blank

1. INTRODUCTION AND DEFINITIONS

1.1 INTRODUCTION

When engaged in any new project or activity to reach a goal, there is always a risk of failure. It's human nature to overlook or minimize such risk in the enthusiasm and optimism of a new project [1]. While plans are made, schedules detailed, costs fine-tuned, and technical solutions specified, the risks get little attention. Who wants to focus on the negative? Who wants to be the one to say the project has issues—splash cold water on the team's forward progress—and perhaps stop the project before it ever gets started? Yet software projects are notoriously susceptible to failure [2]. Every day, we learn about security breakdowns, data leaks, ransom attacks, cost overruns, schedule delays, and user dissatisfaction. Ignoring risks doesn't seem to be a wise decision. But what is the appropriate way to address and manage risk in a software project without killing momentum and morale? Is there a way to integrate risk management into the everyday software development process without extra effort and in a manner that is positive, encouraging, motivating, and reaffirming for our team and leadership?

In this paper, the authors describe a positive approach to integrating risk management into the popular agile scrum development process utilizing model-based system engineering (MBSE). This introductory Section 1 begins with definitions and goals. Section 2 outlines the key principles underlying these definitions of risk, opportunity, MBSE, and Agile Scrum. In Section 3, the authors describe an approach that aligns with the principles. In Section 4, the authors discuss integrating risk management into the agile scrum process and then provide a conclusion in Section 5.

Along the way, we'll consider key questions. What is risk? What can we do to lessen, avoid, or eliminate risk? Who should be responsible for tracking and managing the risks? Is there a way to manage the risks as part of our daily work? How do we represent risk in our MBSE efforts? And how do we utilize the Agile Scrum framework to define, manage, and mitigate the risks? These are questions that all participants in a project will want to consider and answer to improve their chances of success.

1.2 RISK

Risk has been part of human history since the very beginning. From cave dwellers taking physical risks when hunting for survival to the advent of trades through navigation, risk evolved from being just physical to having a financial quality to it. As humanity has evolved, the definition and understanding of risk have evolved, and it has become possible to separate physical risk from financial risk [3].

Risk has evolved and continues to be part of our daily lives. When we get on a plane to reach a beautiful place or get in our car and know that we can go at speeds, we have brakes to help us get to our destination safely. Therefore, life is about taking reasonable risks, avoiding them, and mitigating them without always being aware—without this, we would not progress. Risk is part of our lives and is a natural component of a project. Risk management is how we handle the natural risks in our project.

1. A common definition of risk is "the possibility of something bad happening" [4]. To be more specific, a traditional definition of a risk has four components:
2. An event (i.e., something that can happen).
3. Something with some likelihood of happening (i.e., more than a remote possibility, perhaps 10 to 30% if mitigation steps aren't taken).
4. Something that would negatively impact our ability to achieve our goals.

5. Something we can and should diminish, avoid, or prevent [5].

The definition is infused with uncertainty. Which event? How likely? What exactly would be the impact? The first observation about this traditional definition of risk is its focus on negativity (i.e., the bad things that might happen). However, because the likelihood of the event is unknown, it may not happen, and perhaps the opposite will happen. For example, if the risk is the possibility of a loss of investment funding, then perhaps the opposite is possible: an increase in investment funding, if we manage the risk well. A second observation is that the mitigation techniques are tasks designed to avoid or prevent something negative, but perhaps the tasks could be designed to enhance the opposite—something positive. The uncertainty of the event and its likelihood of occurring suggest an opportunity to flip the event in our favor, creating an opportunity.

1.3 OPPORTUNITY

Opportunity is defined generally as "the possibility of doing something that you want to do," that is, something positive to achieve our goals, something favorable [6]. The definition can be considered quite parallel with risk: 1) an event (something that can happen); 2) something with a likelihood of happening (more than a remote possibility); 3) something that would positively impact our ability to achieve our goals; and 4) something we can and should encourage, take advantage of, and nurture [7]. The definition has similar uncertainty to risk, but from a positive perspective.

The alignment of the definitions of risk and opportunity, considering one to be the flip side of the other, suggests one can take a positive approach to how to think about and manage risk by viewing risk as an opportunity. Opportunity is motivating, empowering, exciting, and rewarding, whereas risk is discouraging, limiting, sobering, and penalizing. Risk is something to avoid; opportunity is something to pursue. Focusing too much on risk can result in an overly defensive approach at the cost of a lack of offense and lost opportunity [8]. Perhaps risk mitigation techniques can be rebranded as opportunity techniques, and risk identification can be reconsidered as opportunity identification. This is not merely a semantic rebranding because we do not settle for simply preventing or avoiding a negative event; we actively address and pursue the opposite positive events. Can we use an opportunity-based approach to positively and organically integrate risk management into our software development?

1.4 MODEL-BASED SYSTEMS ENGINEERING

Spreadsheets, documents, and even in-house-developed tools have been traditionally used to document the intent, analyses, trade-offs, and decisions made through the systems engineering process. This simplified approach works if the systems are simple. But complex systems cannot be easily represented in this way. The possibility of introducing manual errors or propagating errors using this ad hoc approach is too great for complex systems.

MBSE is an effective way to manage a complex product's development, requirements, overall design, analysis, validation, verification, and risk details [9]. At its most basic, a "model" is a simplified version of reality, focused on one or more key characteristics of a complex system [10]. A model helps us to see, explore, and understand those characteristics. Systems engineering is, in lay terms, the art and science of building machines to provide useful, functional, and desirable capabilities effectively and efficiently [11]. Consequently, model-based systems engineering is the art and science of using models to see, explore, and understand systems as an integral part of building them [12]. Systems can be complex, time-consuming, and expensive to build, and they are prone to common and uncommon errors, dissatisfied users, and unfulfilled and unforeseen requirements. As such, creating simplified, focused versions of our systems can help us find solutions and avoid

problems more quickly, with less time and cost. Modeling and simulating can help us identify gaps, explore user needs to better define requirements, and explore options. Further, MBSE enables projects to become more agile because of the sharing and management of project information. The MBSE methodology often aims to increase the quality and efficiency of system definition, architecture, and integration by providing a single authoritative source of truth where the system is represented in an unambiguous way and making understanding the system more accessible to members of the entire project. MBSE benefits include visibility into aspects that are impacted by change during the lifecycle of the project. Additional benefits of MBSE include engaging others and leveraging knowledge across the project [13].

Currently, MBSE is a focus of Navy efforts to improve system engineering [14]. The Unified Architecture Framework (UAF) is the next evolution of the Department of Defense Architecture Framework (DoDAF) [15]. UAF contains architecture modeling elements for representing risk, risk ownership, mitigation techniques and their impacts, and relationships to the rest of the architecture. In this report, we explore how one might use these simplified and focused representations of risk and opportunity to more easily communicate and manage risk among the stakeholders.

1.5 AGILE AND SCRUM

Agile, as originally defined in the Agile Manifesto, is a software development style focused on people (over processes and tools), working software (over comprehensive documentation), customer collaboration (over contract negotiation), and responding to change (over following a plan) [16, 17]. This does not mean there cannot be documentation, processes, tools, and plans; however, those artifacts do not take priority over people and flexibility. The overall point is that the software development team should build customer products more directly for and with the customer, rather than creating and relying on contracts, plans, processes, tools, and documentation. Agile was a reaction against the traditional software development approach ("waterfall"), which was viewed as too sequential, cumbersome, costly, risk-averse, customer distancing, and slow [18].

Scrum is a strategy for implementing Agile, characterized by rapid iterative sprints (e.g., two-week iterations) focused on a list of specific, needed tasks (a "backlog") defined and prioritized by a customer representative ("product owner"), with customer review of product demonstrations and status at the end of every sprint [19]. The focus is on coding rather than documentation, on customer satisfaction rather than detailed requirement fulfillment, and rapid iterations to ensure the team stays on track with customer feedback, expectations, and guidance. Scrum is one technique for implementing Agile principles. Although Agile and Scrum can be easily criticized [20], (such as focusing too much on the developer perspective versus the business perspective; diminishing the importance of other "ilities" such as stability and sustainability; allowing Scrum to re-introduce role-based bureaucracy, scheduled planning, and developer-protective sprints over agile interactions; trivializing the importance of longer-term planning, modeling, and design in favor of near-term working software; and failing to recognize the importance of science-driven user engagement, such as human factors professionals leading a user-centered design effort versus simply appointing or including a customer representative) Agile and Scrum are currently the primary and most popular software development approaches in industry and government.

Now that we have these definitions of risk, MBSE, and Agile Scrum, the question is: How do the characteristics, goals, and components of these concepts fit together to enable a practical, positive, and successful approach to risk management? In the next section, we explore the implied principles of these definitions to help guide our approach.

This page is intentionally blank

2. PRINCIPLES OF MBSE-DRIVEN RISK MANAGEMENT IN AN AGILE SCRUM ENVIRONMENT

A "principle" is a stable, consistent truth or belief about people or the world that, because it is true, is general, does not change, and can be used as a guidepost or measure for determining a successful approach [21]. Considering the driving principles underlying software development, risk, MBSE, and Agile Scrum, our goal is to observe and consider the principles and the relationships among these principles to determine if we can form an approach to risk management that is positive, organic, useful, and productive. From the discussion in the Introduction and Definitions section above, a number of principles can be defined:

1. **Desire to be positive:** When developing systems, people (e.g., developers and sponsors) naturally want to succeed and focus on the positive to create, build, and improve new capability, not focus on the negative and slow the process [22].
2. **Software development is prone to errors:** Software development is known to be complex and error-prone, with a significant history of failed or flawed systems, costly overruns, and dissatisfied customers [23].
3. **Risk management tends to be ignored, postponed, and perceived as negative:** Risk management is recognized as important, but it tends to be diminished or ignored, appearing as a negative drain on positive development efforts [24].
4. **Risk equals opportunity:** Opportunity is the positive end of the risk spectrum (i.e., the likelihood that something good will happen) [25].
5. **Software development is complex:** When developing systems, the complexity of systems and system development can increase the likelihood, obscurity, and impact of errors and make the challenge of finding, understanding, and communicating solutions more difficult [26].
6. **Simplicity and focus of models help manage complexity:** Creating models (i.e., representations simplified to focus on a given characteristic of a complex system) can be a low-cost, rapid, more understandable, and easier-to-communicate method of exploring, developing, and testing new capabilities before, during, and after building them [27].
7. **MBSE applies the benefits of modeling to manage software development complexity:** MBSE creates models to manage software development complexity and is recommended for the development of government systems [28], [14].
8. **Traditional software development ("waterfall") overemphasized the focus on design artifacts, specifically tools, processes, requirements, and planning documentation:** When organizations develop systems, there is a tendency to focus on secondary development artifacts, e.g., the tools, processes, requirements, and planning documentation. In the spirit of superior design upfront, the traditional focus was on defining and establishing a complete definition of what was to be built and documenting it, then building to the documents, rather than flexibly and rapidly adapting to emerging or dynamic customer requirements [29].
9. **Modern Agile software development emphasizes building working software:** Agile consists of four general principles—prioritizing people, customer focus, working software, and flexibility—which can be implemented with different approaches. The focus is on developer, stakeholder, and user collaboration, iteratively building and demonstrating

working software, focusing on the customer's needs, and being flexible and adaptable to requirements change [30].

10. **Dissatisfied customers:** When organizations lack focus on the customer and lack flexibility, they produce software prone to cost overruns, missed requirements, and dissatisfied customers [31].
11. **Organizations favor Agile structures, such as Scrum:** Organizations want to be Agile but tend to desire a more well-defined, structured approach to Agile, such as Scrum, which can be implemented as an easily measurable, repeatable, and rote process [32].
12. **Scrum is structured Agile:** Scrum intends to achieve Agile in an organizationally acceptable manner by specifying a specific, repeatable, and structured process, as evidenced by epics of work consisting of short (usually 2-week) iterative development sprints. Each sprint is composed of a list of tasks or user stories (i.e., a backlog, prioritized by the product owner). At a sprint planning session, the individual tasks are assessed for difficulty and then assigned to developers and tracked with a task management system (e.g., Jira). At the end of the sprint, the accomplished tasks are demonstrated as working software [33].

These principles overlap in six major ways, which can help guide our modern risk management approach: simplicity, deconstruction, positivity, customer involvement, flexibility, and prototyping speed. Concepts and approaches that are *simple* are easier to understand, communicate, and pursue. *Deconstruction* takes complex systems and decomposes them into simpler, interacting component parts. Framing the concepts from a *positive* perspective ensures we maintain morale, do not miss opportunities, and maintain momentum. *Customer involvement* ensures we are building and validating the right system with relevant usability. *Flexibility* means we are allowing for ongoing change in customer requirements, technology, solutions, and development needs. Finally, *prototyping speed* means visibility and proof that we are building a product that meets needs rapidly and enables exploration, innovation, and new requirement elicitation by using an early version of the final product.

3. APPROACH SOLUTION

Risk management is the process of understanding and managing risk proactively, minimizing negative risks, and optimizing opportunities [34]. Risks and opportunities are identified, analyzed, prioritized, treated, and monitored. Below are steps for risk management:

1. **Identify the risk or opportunity** – This can happen at any time during the project's life cycle. This step occurs when uncertainty is detected and potential risks or benefits that could affect a project negatively or positively are documented.

2. **Analyze the risk or opportunity** – This step involves the assessment of the likelihood and impact of the risk or opportunity and identifying ownership, response strategies, and artifacts. At this point, the choice may be to escalate, share, or accept the risk or opportunity as-is. To escalate the risk or opportunity is to request the risk or opportunity be managed at a higher level in the organization. To share the risk is to shift ownership to a third entity that is better suited to respond. To accept the risk or opportunity as-is is to decide to take no action and simply accept whatever happens.

3. **Respond** – This step involves implementing tasks designed to mitigate the risk or enhance the opportunity.

4. **Monitor and Report** – This step involves observing the risk or opportunity, being aware in real-time of the status, and communicating effectively with stakeholders.

These steps describe the general approach to risk and opportunity management; next, we will describe specific steps to integrate this general approach into the Agile Scrum daily processes.

3.1 RISK AND OPPORTUNITY: MBSE MODEL TEMPLATE

We need a simple, reusable, and natural approach for implementing risk management with MBSE and Agile Scrum that aligns with and supports the derived principles. We begin by developing an MBSE risk/opportunity model template, including an associated data schema and supporting data entry forms.

Following the MBSE principle that simple, focused models can help manage complexity, we should consider modeling risk management. Using the UAF, let’s consider a reusable template, shown in Figure 1, for representing risk and opportunity in an MBSE environment. Fortunately, the UAF provides modeling concepts (e.g., risk), activities (e.g., mitigation), resources (e.g., roles, systems), and relationships (e.g., ownership, satisfaction) that we can use to create our model. Following the additional principles that risk can be viewed as opportunity and that developers tend to want to focus on the positive, we should consider incorporating opportunity into the template. With these principles in mind, we present the reusable Risk Opportunity Management E-template (ROME), shown below in Figure 1. The goal is to ensure that risk and opportunity are defined, communicated, and managed organically and seamlessly during the entire development process.

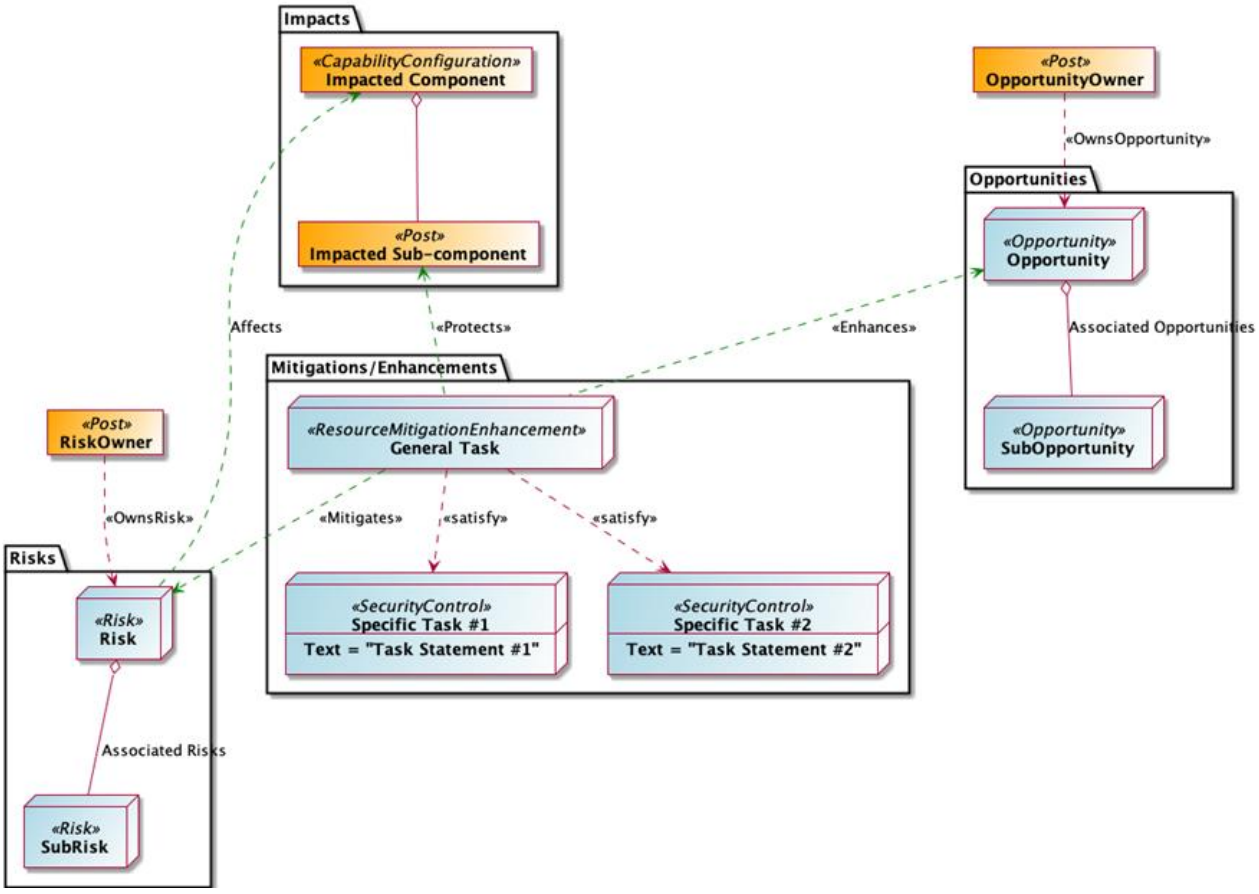

Figure 1. MBSE diagram of risk/opportunity and mitigation/enhancement components.

Modeling risk and opportunity in the architecture provides the advantage of ensuring risk and opportunity management is an integral part of the system, avoids negative connotations, and thus instills a positive risk (opportunity) management mentality. Often, risk management practices may fail because an organization has not yet matured a risk culture. A risk culture involves the awareness and behavior of each organization member towards risk. The more developed the risk culture, the more apt the organization is to handle risk. In addition, associating risk and opportunity within the same structure reinforces a broader perspective on risk management.

Risks and opportunities fit into a defined structure in the model. Stakeholders do not lack guidance, as they are effectively guided through the process. The risk/opportunity model identifies ownership, roles or components affected, resource mitigation and enhancement, and the artifacts and safeguards needed to protect the system from failure. The risk/opportunity template can be reused to provide quality, improve schedules, and reduce costs.

Quality is improved because stakeholders and developers establish mitigation or enhancement strategies quickly, oftentimes as a feature is being designed. Reusing the template also provides a clear structure that helps to identify the substance rather than worrying about the form.

A structure where aspects are easily identifiable saves time for concentrating on the mitigation or enhancement strategy. Modeling risk management helps the project remain on schedule.

The template provides the basic framework for considering risk and answers the following questions:

- **name**: What is the name of this risk/opportunity?
- **description**: What is the risk/opportunity?
- **ownership**: Who "owns" (has the lead responsibility to track and manage) the risk/opportunity?
- **impact**: What parts of the project or system does the risk/opportunity impact?
- **mitigation/enhancement**: What different approaches (sets of mitigation activities) can we employ to minimize the risk or enhance the opportunity?
- **activities**: What are the security controls (operational activities) that satisfy the specific approach, reduce risk, or enhance the opportunity?

By using this template for risks of significant concern, a development team can begin to identify, consider, and document their approach to handling risk.

3.2 RISK AND OPPORTUNITY: DATA SCHEMA AND FORMS

A supporting notional schema for providing a RESTful OpenAPI interface is shown in Figure 2. Representational state transfer (REST) is a popular, scalable architectural style familiar to everyone who has viewed a web page in a web browser, filled in a web form, and clicked on a link. REST principles include:

1. Uniquely identifying resources (things) by their uniform resource locator (URL).
2. Representing the state of those resources as a set of property or values in formats such as HyperText Markup Language (HTML) for humans or Javascript Object Notation-Linked Data (JSON-LD) for machines.
3. Creating, retrieving, updating, and deleting (CRUD) the state of resources by applying to the URLs a small set of corresponding standardized functions (e.g., HyperText Transfer Protocol: HTTP POST, GET, PUT, and DELETE).
4. Representing application state (going forward or backward in the application) by providing hypertext links [35]. OpenAPI is a standard format for representing and documenting the resources (the URLs and the data schema for representing the resource state) and the allowed CRUD operations on those resources [36].

All the substantive information for a given instance of the MBSE model (represented in Figure 1) can be captured using the schema in Figure 2. The benefit of such a representation—a common standardized machine-understandable format—is that it assists with reusability, interoperability, sharing, and ingress/egress of ROME models [37]. Further, the overlap of common data items between events, risks, opportunities, and tasks is handled in the schema in Figure 2. Specifically, since understanding the "Impact" and "nextSteps" (follow-on tasks) is important for any "event," the primary difference between a "RiskOpp" (risk or opportunity) and an event is that an event has happened, whereas a RiskOpp is a potential future event with a given likelihood. Similar to events, "tasks" have impacts and potentially nextSteps, but since they are in the future, they are similar to RiskOpps in that they have a likelihood of happening and have their own potential RiskOpps. Capturing this similarity and recursiveness in the schema (e.g., a task is a type of RiskOpp, and a RiskOpp is a type of event) can seem overly theoretical; however, it allows for reuse of a few simple standardized machine formats and reuse of any associated capabilities built to support these formats (e.g., mapping, persistence, publishing). In addition, the structure avoids duplication of fields and the need to change datatypes simply due to an event occurring in the past, present, or future.

To support risk/opportunity capture as a natural part of collecting event information, a traditional event format is extended in the schema in Figure 2 to include impacts, nextSteps (i.e., mitigation and

enhancement tasks), and likelihood. Since responders to events also benefit from event impact suggestions and possible response tasks to be considered, the schema suggests the only real difference between a risk/opportunity and an event is that the latter has not happened yet and may not, so it has a likelihood and a time when the event is expected.

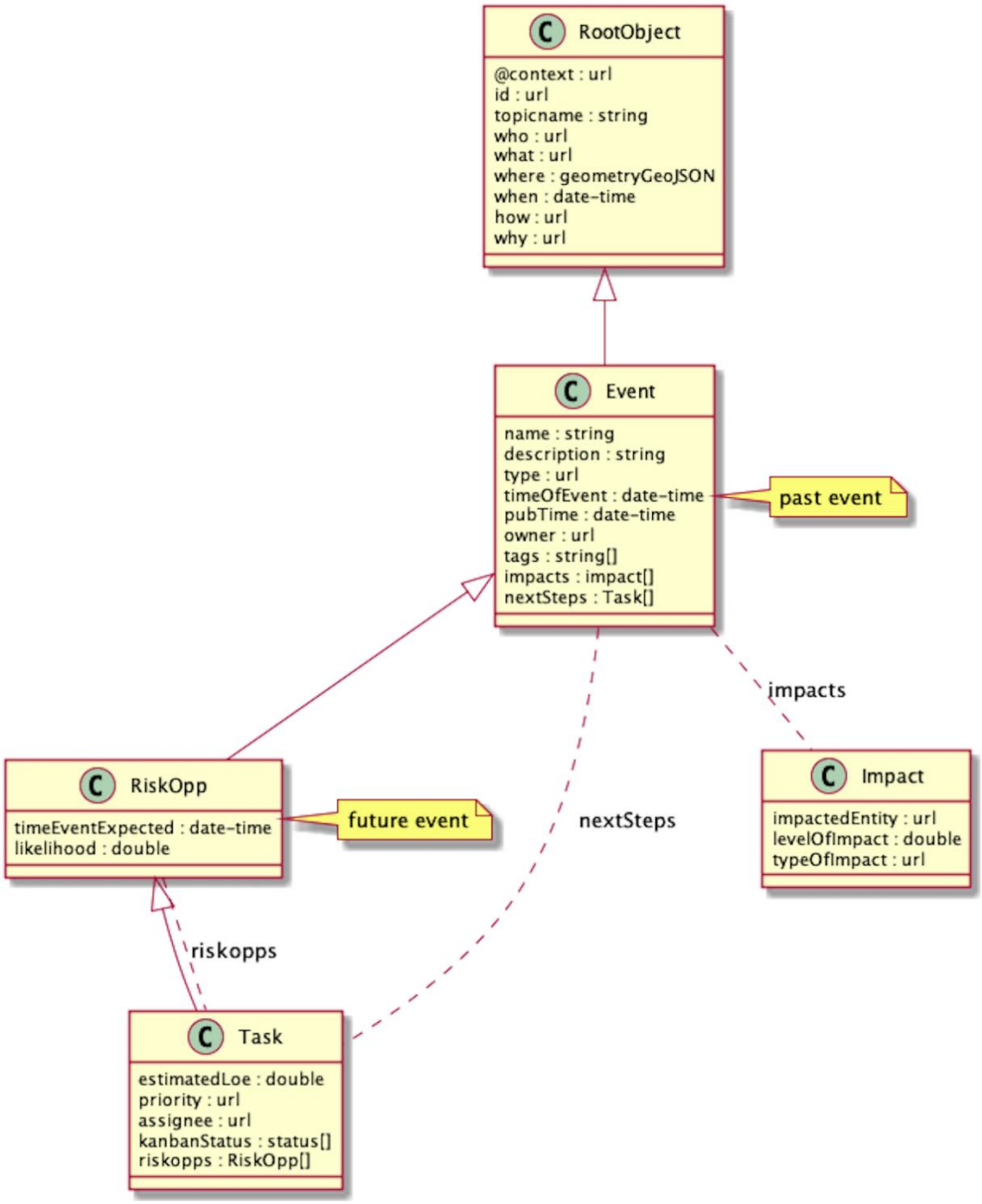

Figure 2. Notional event data schema for representing risks/opportunities in a machine-understandable format.

The schema supports:

1. A javascript object notation - linked data (JSON-LD) representation, a standard from the World Wide Web Consortium (W3C), with the inclusion of the "@context" element in the RootObject [38].

2. Reuse of official standards with the inclusion of the GeoJSON element, a draft standard from the Internet Engineering Task Force (IETF) [39].

3. Linked data principles with the use of URLs instead of conventional strings (the linked data principle is "things, not strings") [40]. We do not go into detail here regarding the important benefits of supporting these standards and best practices; however, the references include links to videos and tutorials on these important topics.

To build the model template (Figure 1), it may require the data from the schema (Figure 2) to be collected from the development team using a form. Three conceptual forms are shown in Figure 3, to represent events, RiskOpps, and tasks, which could become forms in Jira or other task tracking systems and be used in the normal Agile Scrum process organically and seamlessly. As conceptualized, an event is something that happens that one cares about and for which one wants to be notified. An event potentially impacts people or project components positively (opportunity) or negatively (risk). The impact is represented by who or what is impacted, the type of impact, and the positive or negative amount of impact. If there is an impact, then a potential RiskOpp arises. By adding a RiskOpp to the event, one is taken to the RiskOpp form.

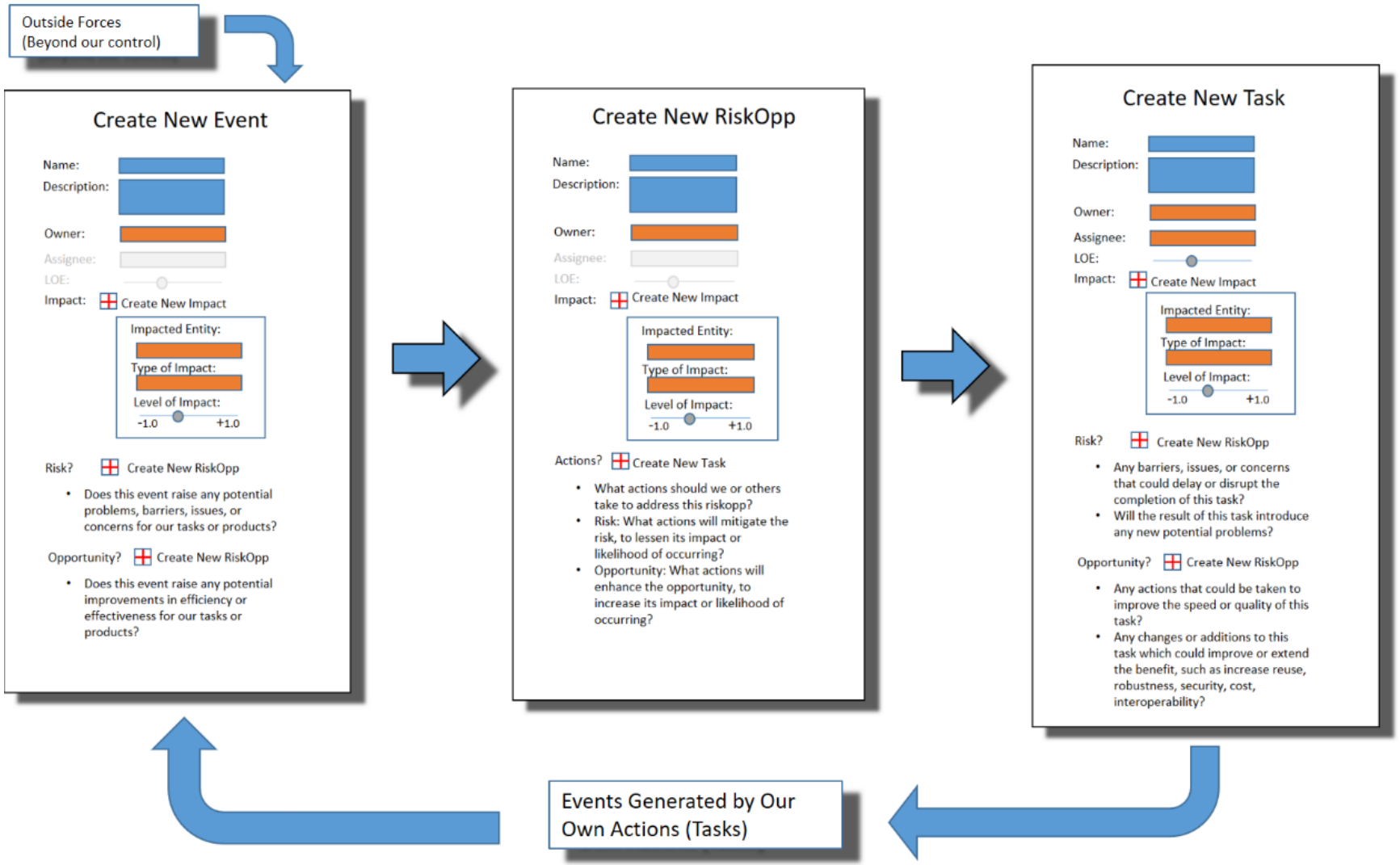

Figure 3. Conceptual risk management entry forms and process flow.

4. INTEGRATING RISK MANAGEMENT INTO THE AGILE SCRUM PROCESS

Details of the RiskOpp can be entered, but now, what will one do about it? What actions will one take to mitigate the risk or enhance the opportunity? Those actions can be represented by creating a new task on the task form, which includes typical information such as task assignee and level of effort. However, because the task is in the future and impacts things positively or negatively, the performance of the task itself and the product produced introduce their own potential risk or opportunity. For example, there is a risk the task will not be completed, an opportunity to complete the task faster, or the potential to avoid the need for the task altogether if events change. Also, the product of the task (e.g., a software product) may introduce risk or opportunity itself. In either case, a new RiskOpp can be added. Finally, the three forms represent a process flow when new events are created, either from forces outside of one's control or because of one's own actions (i.e., tasks), intentionally or unintentionally.

This approach does not add extra processes or meetings; rather, it takes advantage of existing Agile team meetings to discuss risks and opportunities. The path from modeling risks and opportunities to tracking and monitoring is done through the Scrum process. At the beginning of a sprint, when tasking is defined, risks and opportunities should be identified and discussed in relation to the tasks mapped to the backlog. A summary of the major milestones (meetings) of the Agile Scrum process [39] is provided in Table 1.

Table 1. Summary of the major milestones (meetings) of the Agile Scrum process.

Agile Scrum Milestone	Description
Sprint Planning	This is the event when the team identifies all the tasks that will be performed during the sprint. The scrum team(s) and product owner work together to determine priorities and define or redefine the goals for the sprint. During the sprint planning event, the team can identify risks and opportunities associated with the tasks, in addition to identifying all the tasks that need to be worked on to accomplish the sprint's goals.
During the Sprint	This is the time when the teams execute the work to meet the sprint's goals. During the daily scrums, the team collaborates on anything that might impede progress. The goal is to keep productivity at a high level. The daily scrum provides a chance to discuss any important details about risks and opportunities. The status of tasks to reduce risks or enhance opportunities can be included.
Sprint Demonstration and Retrospective	The "working software" built during the sprint is demonstrated on the last day of the sprint to validate the progress and gather feedback from the product owner (customer). The goal of the sprint retrospective is to (1) gather feedback from the development team on how well the sprint went from their perspective, (2) improve the scrum team, and (3) increase the quality of the technical and production processes. Both the demonstrations and the retrospective are times when the opportunities and risks can be discussed and the sprint is completed.

During the planning and execution of the sprint, there are three cases of task, risk, and opportunity data collection:

1. New Task: A task (e.g., a new customer requirement, a risk mitigation task, or an opportunity enhancement task) is added to the backlog and prioritized for a given sprint. Every new task might have links to its own risks or opportunities associated with it.

2. New Risk: Uncertainty is detected and determined to have a negative effect on the project. Identifying risks can happen at any time but can be elicited during planning by asking the following questions of the task assignee and team: "Do you see any barriers, blockers, or challenges to completing the task?" "What do you think we could do to avoid, prevent, or lessen the impact of this barrier, blocker, or challenge?" Details can be gathered in the form of risks, and the actions to avoid, prevent, or lessen can be captured and linked as new tasks, i.e., nextSteps, in response to this risk.

3. New Opportunity: Uncertainty is detected and determined to have a positive effect on the project. Identifying opportunities can happen at any time, but they can be elicited during planning by asking the following questions of the task assignee and team:
    - Do you see any chance for reuse, expansion, extension, or improvement of this task for the benefit of others?
    - What do you think we could do to expand, extend, or improve this task? Details can be gathered in the form of opportunities, and the actions to expand, extend, and improve can be captured and linked as new tasks, i.e., nextSteps, in response to this opportunity.

Tasks, risks, and opportunities are connected. A task can generate a risk or an opportunity. Risks and opportunities can generate tasks that need to be executed to mitigate the risk or enhance the opportunity.

Many Scrum teams utilize an issue tracker, such as Jira, to support the entry and optional linkage of events, risks, and tasks [41]. The authors build on that approach and utilize links to connect tasks, risks, events, and opportunities. The advantages of this approach are that the Agile team does not have to use a new tool and that there are no initially required dependencies between tasks, risks, and opportunities. Furthermore, one item—a task, risk, or opportunity—does not have to necessarily be entered before the other. The issues are independent, but the links provide an integrated look at all relationships among tasks, risks, and opportunities. With this approach, all information is collected in one place: the task management system.

Below are the steps to implement this solution:

1. **Identify:** The project team or stakeholders identify and define any type of risk-related concern, such as events, risks, opportunities, and tasks. The type reflects the class types illustrated in Figure 2, in addition to any project-specific types added by extending the schema.

    Examples of fields associated with one or more of these types include:

    - timeEventExpected: the time when the event is, or was, expected to take place.
    - likelihood: probability of an event occurring.
    - estimatedLoe: the amount of work required to accomplish a task.
    - owner: owner of the risk, opportunity, or task.

- kanbanStatus: task completion status (e.g., NotStarted, InProgress, Done).
- RiskOpp: a risk or opportunity.

When creating a new risk, opportunity, or task, the questions in the form will motivate the user to consider and identify possible positive or negative consequences (risks or opportunities). The questions are written to encourage thinking beyond the immediate circumstances and to consider the project from beginning to end.

Example prompts:

- Are there any challenges or barriers to completing this task? This question would help identify any possible risk.
- Can this task be reused or extended for the benefit of others? This question would help identify any opportunity to be derived from the task.

2. **Analyze:** Once risks and opportunities are identified and entered, strategies for responding and managing the concern need to be developed according to the principles of risk management.

   The team will discuss risks, opportunities, and the possible consequences and plan how to respond to them. If steps (i.e., tasks) to mitigate risk or enhance opportunity are identified, they can be entered in the task management system. Then these responsive mitigation or enhancement tasks can be associated with the risks or opportunities by connecting them with links.

3. **Respond:** A unique task management issue type can represent responses to each risk or opportunity where the associated mitigation or enhancement tasks can be tracked. Tasks that themselves have a risk or an opportunity associated with them can be linked to the original risk or opportunity. A task can have multiple risks or opportunities associated with it. Mitigation or enhancement tasks can be added to the sprint when identified or added to the backlog for later reprioritization and sprint assignment.

4. **Monitor:** Monitoring of the risks and opportunities is executed at the sprint iterations, during the sprint planning, and during retrospective meetings. Risk and opportunity information is represented in the model and linked to the tasks and epics related to it. Mitigation and enhancement tasks that are added to the current sprint can be tracked on the Kanban status board, just like any other task.

The data from the forms can be exported and used to create the MBSE model diagrams by importing into an appropriate MBSE tool or, alternatively, by reading the data into a programming language, generating the diagram layout in an appropriate format (e.g., PlantUML, and then generating the diagram). The specifics of this approach remain the next steps for this ongoing initiative.

This page is intentionally blank

5. CONCLUSION

This paper describes a reusable approach for how to integrate MBSE-driven risk and opportunity management into the agile software development process.

Risk and opportunity management is too often a reactive response to events, an afterthought, where possible opportunities are not investigated or identified, and where a negative connotation is attributed to the entire process.

The authors suggest a practical solution to track risks, opportunities, mitigation, and enhancement strategies by simply using forms in a track management system and integrating the identification, analysis, response, and monitoring of risks and opportunities into the normal Agile Scrum process.

The authors provide this practical approach where everyone on the team is engaged and where risks and opportunities are part of the everyday activities. The approach includes an MBSE template and a process with associated schema and forms for how to model, collect, link, and track risks and opportunities in a natural and positive manner. This approach facilitates an improved, dynamic, and positive risk management culture.

This page is intentionally blank.

REFERENCES

1. Buehler, Roger, Dale Griffin, and Johanna Peetz. "The planning fallacy: cognitive, motivational, and social origins." Advances in Experimental Social Psychology, Vol. 43, Academic Press, 2010. pp. 1–62.
2. Klein, Gary, and James J. Jiang. "Seeking consonance in information systems." Journal of Systems and Software 56.2 (2001): 195–202.
3. Dionne, Georges, Risk Management: History, Definition, and Critique (September 6, 2013). Risk Management and Insurance Review 16, 2, 147–166, 2013. Available at SSRN: https://ssrn.com/abstract=2231635 or http://dx.doi.org/10.2139/ssrn.2231635
4. "risk," Cambridge Dictionary, https://dictionary.cambridge.org/dictionary/english/risk
5. "risk," Glossary, Society for Risk Analysis. Retrieved June 29, 2023. https://www.sra.org/wp-content/uploads/2020/04/SRA-Glossary-FINAL.pdf
6. "opportunity," Collins English Dictionary. Retrieved June 29, 2023. https://www.collinsdictionary.com/us/dictionary/english/opportunity
7. "opportunity," Glossary, Society for Risk Analysis. Retrieved June 29, 2023. https://www.sra.org/wp-content/uploads/2020/04/SRA-Glossary-FINAL.pdf
8. Boehm, B. (2014). Software Project Risk and Opportunity Management. In: Ruhe, G., and Wohlin, C. (eds), Software Project Management in a Changing World. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-55035-5_5
9. Topper, J. Stephen, and Nathaniel C. Horner. "Model-based systems engineering in support of complex systems development." Johns Hopkins APL technical digest 32.1 (2013): 419–432.
10. Shevchenko, N., 2020: An Introduction to Model-Based Systems Engineering (MBSE). Carnegie Mellon University, Software Engineering Institute's Insights (blog), Accessed June 29, 2023, https://insights.sei.cmu.edu/blog/introduction-model-based-systems-engineering-mbse
11. "systems engineering," INCOSE. Retrieved June 29, 2023. https://www.incose.org/about-systems-engineering/system-and-se-definition/systems-engineering-definition
12. Long, David, and Zane Scott. A primer for model-based systems engineering, 2012. http://ccose.org/media/upload/MBSE_Primer_2ndEdition_full_Vitech_2011.10.pdf
13. Cook D., Schindel WD. Utilizing MBSE patterns to accelerate system verification. INCOSE INSIGHT. 2017;20(1):32-41
14. U.S. Navy & Marine Corps, Digital Systems Engineering Transformation Strategy, Washington, DC, 2020. https://nps.edu/documents/112507827/0/2020+Dist+A+DON+Digital+Sys+Eng+Transformation+Strategy+2+Jun+2020.pdf/
15. OMG, OMG UAF Wiki. Retrieved June 29, 2023. https://www.omgwiki.org/uaf/doku.php
16. Beck, Kent, et al., "The Agile Manifesto." (2001): 2009.
17. Flora, Harleen K., and Swati V. Chande. "A systematic study on agile software development methodologies and practices." International Journal of Computer Science and Information Technologies 5.3 (2014): 3626–3637.

18. Abbas, Noura, Andrew M. Gravell, and Gary B. Wills. "Historical roots of agile methods: Where did “Agile thinking” come from?" Agile Processes in Software Engineering and Extreme Programming: 9th International Conference, XP 2008, Limerick, Ireland, June 10–14, 2008. Proceedings 9. Springer Berlin Heidelberg, 2008.
19. Schwaber, Ken, and Jeff Sutherland. "The Scrum Guide, 2020." Accessed June 29, 2023. (2021). https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf
20. N. Ozkan, "Imperfections Underlying the Manifesto for Agile Software Development," 2019 1st International Informatics and Software Engineering Conference (UBMYK), Ankara, Turkey, 2019, pp. 1-6, doi: 10.1109/UBMYK48245.2019.8965504. https://www.researchgate.net/publication/337323304_Imperfections_Underlying_the_Manifesto_for_Agile_Software_Development
21. “principle,” Stevenson, Angus; Lindberg, Christine A., eds. (2010-01-01). "New Oxford American Dictionary." Retrieved June 29, 2023.
22. Wideman, R. Max. Project and program risk management a guide to managing project risks and opportunities. Project Management Institute, Inc., 2022.
23. Firlit, M. “Managing Risk with Scrum,” August 9, 2020, http://org/resources/blog/managing-risk-scrum
24. Boulding, W., Morgan, R., & Staelin, R. (1997). Pulling the Plug to Stop the New Product Drain. Journal of Marketing Research, 34(1), 164–176. https://doi.org/10.1177/002224379703400114
25. West Devon Borough Council, Risk and Opportunity Management Strategy. Retrieved June 30, 2023. https://www.westdevon.gov.uk/article/5156/Risk-and-Opportunity-Management-Strategy
26. Nystedt, Sofia, and Claes Sandros. "Software complexity and project performance." School of Economics and Commercial Law at the University of Gothenburg (1999). https://gupea.ub.gu.se/bitstream/handle/2077/1316/Nystedt.Sandros.ia7400.pdf?sequence=1
27. Schmidt, Douglas C. "Model-driven engineering." Computer-IEEE Computer Society 39.2 (2006): 25.
28. Topper, J. Stephen, and Nathaniel C. Horner. "Model-based systems engineering in support of complex systems development." Johns Hopkins APL technical digest 32.1 (2013): 419–432.
29. Petersen, Kai, Claes Wohlin, and Dejan Baca. "The waterfall model in large-scale development." Product-Focused Software Process Improvement: 10th International Conference, PROFES 2009, Oulu, Finland, June 15–17, 2009. Proceedings 10. Springer Berlin Heidelberg, 2009.
30. Abrahamsson, Pekka, et al., "Agile software development methods: Review and analysis." arXiv preprint arXiv:1709.08439 (2017).
31. Hoda, Rashina, James Noble, and Stuart Marshall. "The impact of inadequate customer collaboration on self-organizing Agile teams." Information and software technology 53.5 (2011): 521–534.
32. Kapitsaki, Georgia M., and Marios Christou. "Where is Scrum in the current Agile world?" 2014 9th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE). IEEE, 2014.
33. Srivastava, Apoorva, Sukriti Bhardwaj, and Shipra Saraswat. "SCRUM model for agile methodology." 2017 International Conference on Computing, Communication, and Automation (ICCCA). IEEE, 2017.

34. Olsson, Rolf. "In search of opportunity management: Is the risk management process enough?" International journal of project management, 25.8 (2007): 745-752.

35. Lange, Kenneth. "The Little Book on REST Services." (2016). http://www.kennethlange.com/books/The-Little-Book-on-REST-Services.pdf

36. The OpenAPI Initiative, “The OpenAPI Specification Explained” Accessed August 14, 2023. https://learn.openapis.org/specification/

37. Sansone, Susanna-Assunta, and Philippe Rocca-Serra. "Interoperability Standards-Digital Objects in Their Own Right" Wellcome Trust 10 (2016): m9.

38. JSON-LD 1.1: A JSON-based Serialization for Linked Data, W3C Recommendation, July 16, 2020. https://www.w3.org/TR/json-ld11/ https://youtu.be/vioCbTo3C-4

39. GeoJSON, Internet Engineering Task Force (IETF), RFC 7946. https://datatracker.ietf.org/doc/rfc7946/ https://youtu.be/B9uZCizEqWs

40. Mountantonakis, Michalis, and Yannis Tzitzikas. "Large-scale semantic integration of linked data: A survey." ACM Computing Surveys (CSUR) 52.5 (2019): 1–40.

41. Akhgarnush, Eljar, Fabian Bruse, and Ben Hofer. "New Project Structure—Agile & Scrum." The Digital Journey of Banking and Insurance, Volume I: Disruption and DNA. Cham: Springer International Publishing (2021): 213-237.

This page is intentionally blank.

INITIAL DISTRIBUTION

84310	Technical Library/Archives	(1)
53224	F. Mirabile	(1)
53225	P. Suggs	(1)
53623	J. Waters	(1)

Defense Technical Information Center
Fort Belvoir, VA 22060–6218 (1)

This page is intentionally blank.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-01-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
April 2024	Final	

4. TITLE AND SUBTITLE

Capturing Model-based Risk Management in an Agile Scrum Environment

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHORS

Francesca V. Mirabile
Parker C. Suggs
Jeff H. Waters
NIWC Pacific

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NIWC Pacific
53560 Hull Street
San Diego, CA 92152–5001

8. PERFORMING ORGANIZATION REPORT NUMBER

TR-3343

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release. Distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This paper presents a novel approach to risk management by integrating model-based systems engineering (MBSE) within an agile scrum environment, aimed at enhancing software development practices. The authors introduce a reusable MBSE template for risk representation, complete with a data schema and entry forms, designed to ensure that risk management is a continuous, integral part of the software development lifecycle.

15. SUBJECT TERMS

risk management, Agile, model-based system engineering, MBSE, software development

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Jeff H. Waters
U	U	U	SAR	38	19b. TELEPHONE NUMBER *(Include area code)* 619-553-3657

This page is intentionally blank.

This page is intentionally blank.

Approved for public release. Distribution is unlimited.

Naval Information Warfare Center (NIWC) Pacific
San Diego, CA 92152-5001