

Article

Causal Interactions in Agile Application Development

Saulius Gudas and Karolis Noreika * 

Institute of Data Science and Digital Technologies, Vilnius University, LT-08412 Vilnius, Lithuania;
saulius.gudas@mif.vu.lt

* Correspondence: karolis.noreika@mif.vu.lt

Abstract: The Agile approach and tools are popular for the management of Enterprise Application Software (EAS) development. This article focuses on the issue of inconsistency between strategic business objectives and the functionality of the software developed. Agile management tools lack the functionality of EAS project activities coordination. This article aims to rethink Agile project management using the causal modelling approach. A causal model of Agile project management using a management transaction (MT) concept was developed. The notion of the space of processes was used to identify the MTs location along the axes of aggregation, generalization, and time and to formalize their interaction specifications. Taxonomy of the coordination meta-types and types was developed using the identifiers of the MTs. The modified Agile activities hierarchy was developed, and vertical and horizontal causal interactions between Agile activities were identified. This modified Agile management model helps to consistently track the integrity of EAS project content. Complexity indicators were introduced to evaluate the EAS project complexity and their average and normalized values are presented. Additional attributes in the Agile management tool Jira are proposed. Monitoring mismatch between strategic business objectives and development activities content helps to improve the success of EAS projects delivery.

Keywords: agile management; causal modelling; management transaction; coordination; management control

MSC: 37M05



Citation: Gudas, S.; Noreika, K.

Causal Interactions in Agile

Application Development.

Mathematics **2022**, *10*, 1497. [https://](https://doi.org/10.3390/math10091497)

doi.org/10.3390/math10091497

Academic Editor: Jan Awrejcewicz

Received: 24 February 2022

Accepted: 28 April 2022

Published: 30 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Using Agile methods for business management and software development management emerged in 2001 when the Agile manifesto was formulated [1]. Agile frameworks such as Scrum, XP, ScrumBan, and SAFe gained popularity over the years [2] because they provide a way to adapt to change and potentially reduce the costs of projects. However, the success rates of project delivery are still poor and only 19% are considered successful in terms of agreeing to time, budget, scope, and customer satisfaction. There are several reasons for such rates, including lack of people's competency, alignment between business and IT, and a constantly changing business environment. This research is focused on the formal definition of the coordination in the hierarchy of the Agile activities (theme, initiative, epic, and user story).

User stories are used in Agile methods as a general practice to capture business needs. However, user stories do not fully express the link between the requirements for the Enterprise Application Software (EAS) and the strategic objectives of the enterprise. EAS is a type of software that is used by multiple users in an enterprise (such as ERP, CRM systems, etc.). In order to provide additional context information about a user story, such concepts as theme, initiative, and epic are used. Together they form the TIES hierarchy [3]. Traditional Agile project management involves monitoring and managing the interactions between these levels of Agile concepts by project managers (human activities). Modified Agile management aims to monitor the interaction of Agile concept levels with virtual tools

by providing the project or program manager with a dashboard of unmatched interactions, semantics, and other attributes.

Causal modelling is used to investigate the vertical and horizontal interactions of the Agile management activities (theme, initiative, epic, and user story). The management transaction (MT) framework is used as a predefined internal structure for modelling any Agile activity. To ensure that work performed on lower levels of the Agile hierarchy contribute to strategic business objectives, any two adjacent levels in the Agile hierarchy must be defined as management transaction. This allows ensuring that causal links are defined and information is transferred properly throughout the Agile hierarchy.

The remainder of this article is structured as follows: Section 2 elaborates on Agile software management tools and frameworks, methods from business management, and business and IT interaction modelling fields used for EAS project management. Section 3 presents the traditional Agile project management hierarchy, causal modelling paradigm, management transaction concept, and introduces modified Agile management hierarchy. In Section 4 causal modelling in EAS development management is presented. Section 5 contains the formalization of the Agile activities and their management through the taxonomy of coordination in the space of processes. Section 6 contains the description of interaction content modelling of the Agile activities. In Section 7, the semantics of vertical and horizontal interactions are presented and EAS project complexity indicators are introduced. Finally, in Section 8, the method described in previous sections is applied in the Agile project management tool Jira and the causal modelling approach is demonstrated to ensure that work performed on lower levels of the Agile hierarchy contributes to strategic business objectives.

2. Related Works

2.1. Agile Software Management Tools and Frameworks

Many software tools support the Agile software project management process. Atlassian Jira software is one of the leaders [2]. Jira has many customization options, but the current functionality of Jira only supports the definition of the TIES structure activities. It does not provide any formalities to ensure that the links between the different activities in the levels of the Agile hierarchy are properly identified and justified in the business context. This means that one cannot verify that every lower-level TIES activity is linked and integrated properly to the theme on the highest level on the Agile project management hierarchy. These vertical links are only defined by EAS project experts.

The Agile Scrum framework is the most popular Agile framework to ensure EAS project delivery success. However, this is mostly a team level (up to 10 people) framework. Scaled Agile frameworks such as SAFe [4] or LeSS [5] are used for enterprise-level Agile project management. However, these frameworks lack the tracing logic of the internal links between their own activities.

Research of the application of the modified Agile methods is receiving attention from various business domains, such as manufacturing, logistics, and education. The modified value stream mapping Agile framework for earthmoving equipment manufacturing business was presented in [6]. Other areas include application in logistics where an Agile framework for a logistics area company was proposed [7] and for improving the educational process [8,9].

2.2. Business and IT Alignment Using Business Management Methods

Business and IT alignment methods that are based on a business management approach are focusing on communication and leadership. Strategic management is the ability to communicate decisions from the top leadership throughout the hierarchy of departments and employees. Strategic management is defined as “the process of assessing the corporation and its environment in order to meet the firm’s long-term objectives of adapting and adjusting to its environment through manipulation of opportunities and reduction of threats” as stated in [10].

Coordination problems are caused by interdependencies between two or more tasks. Coordination is defined as “integrating or linking together different parts of an organization to accomplish a collective set of tasks” [11]. The theory of coordination conceptualization in the IS domain based on a neurobiological perspective was presented in [12].

Business management methods rely on the communication between the employees in an enterprise. However, it is not ensured that strategic objectives information is sufficiently transferred through all the levels of the organization and eventually as software development tasks to ensure business and IT alignment.

2.3. Business and IT Interaction Modelling Methods

Business and IT capabilities alignment was firstly researched in [13] and the Business and IT strategical alignment model (SAM) was introduced. SAM consists of four domain alignment perspectives (business and IT strategy, organization and information systems infrastructure, and processes). Although SAM is aiming to identify causal dependencies between business activities and IT-based activities, it is still fuzzy and focuses mostly on the business management part of the problem. SAM is the early attempt to identify causal dependencies between business activities and IT-based activities.

Another well-known method for business and IT alignment is Guidelines Regarding Architecture Alignment (GRAAL). GRAAL provides a collection of concepts and relations among them [14]. Several definitions of alignment were identified in [15] based on the SAM, i.e., business alignment, two types of cross-domain alignment, intellectual alignment, IT alignment, and operational alignment.

Object Management Group (OMG), the leading organization in business modelling practices and methodologies, provides and supports a group of methods under the Business Process Model and Notation (BPMN) and Decision Model and Notation (DMN) categories [16]. Model-driven architecture (MDA) and model-based engineering (MBE) approach for software development is consistent with these modelling methods.

However, the aforementioned methods are not adapted to be used for business domain management processes and are not able to capture the causal knowledge of business management interactions. The causal links in the business domain processes are very complex. The domain models created using these notations are simplified empirical projections of real-world processes that do not take into account the complexity of the process management interactions. A new method is required to evaluate the complexity of management interactions content and utilize the flexibility of Agile methods.

3. Preliminaries

3.1. Traditional Agile Project Management Hierarchy

Traditional Agile management hierarchy TIES includes these activity types: strategic objective identification, themes specification, initiative definition, epic specification, and user story specification. It is based on the iterative development model [17] and by completing a set of lower-level activities, a higher-level activity is completed.

The lowest level of granularity in Agile software projects usually is the user story. User stories intend to provide an abstract, but quite detailed description of the business problem and value for the customer. The user story concept was popularized in [18]. The timeframe when the user story should be completed should be no longer than a single development iteration (1 to 4 weeks).

The lower level in the Agile software project management hierarchy is epic. As defined in [19], “epics are really large stories that align with the product vision and provide the next level of product detail for senior management”. Epics should be completed within a calendar quarter or up to 12 weeks.

Going further up in the Agile hierarchy is initiative. An initiative is “a collection of epics that drive toward a common goal” as defined in [20]. Initiatives provide a perspective for business and IT on achieving business goals. It should be completed in a year or less.

Theme (also called feature) is the top-level activity in Agile software project management hierarchy TIES. It is “an aggregation of user stories to show business value delivered and to help with prioritization as well as show planned product delivery at a high level” [21]. Strategic business objectives are translated into themes as more specific means to achieve those business objectives. The theme is the transition level between strategic business objectives and initiatives.

The TIES hierarchy with more details is presented in [22]. The TIES structure is used in the Agile project management tool, Jira. Jira is the most popular tool for Agile project management [2]. Other popular Agile frameworks such as SAFe [4] use similar concepts for their own activities hierarchy that can easily be mapped to the TIES structure. The Agile activities hierarchy TIES is implemented in the Jira tool and was chosen for practical purposes: to enhance the existing environment for automatic tracing the vertical integrity from themes to user stories. TIES hierarchy helps to semantically link the business requirements with the information system requirements starting from a user story level to a strategic objective as presented in Figure 1.

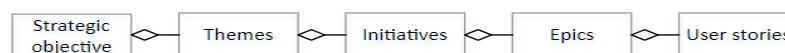


Figure 1. Agile management hierarchy (traditional), $\diamond$ aggregation relationship.

In Figure 1 the hierarchy of key Agile concepts is depicted: strategic objectives consist of themes, each theme consists of some set of initiatives, each initiative consists of a set of epics, and epics are decomposed into a set of user stories. Traditional Agile project management involves monitoring and managing the interactions between these levels of Agile concepts by department, program, or project managers (human activities).

The internal structure of the Agile activities during development iterations (sprints in Scrum [23]) is not structured from a causal modelling perspective and therefore it is impossible to model it. The specifics of the Agile activities are not fully considered, as the traditional Agile hierarchy usually is perceived as a tree structure [22] that does not provide the specifications of the vertical feedback loop information between the Agile activities. The experts need to define the integrity of vertical links between the Agile activities. For these reasons, a new method is required that would specify the informational content of interactions between the levels of Agile activities and can validate how complete the EAS project tasks interaction specification is. This would provide an overview of informational gaps in the EAS project context.

The problem of linking the strategic objectives to the TIES structure is not further discussed in this article. This link was investigated using the enterprise architecture framework MODAF in [24]. MODAF is the most widely known and established framework and it was used to model the strategic objectives as enterprise vision and capability taxonomy models. This allowed for gradually linking the modelled contents of the strategic business objectives to themes with their own set of MODAF models.

3.2. Causal Modelling Paradigm

Real-world processes modelling methods are classified according to the modelling paradigms explained in Figure 2. External modelling is based on the black-box approach, where the internal details of the system are not clear, and the model is specified as an input-process-output. Internal modelling involves a deep prior knowledge of the causal relationship of an area, the behavior of its elements, and internal causal relationships [25]. Grey box modelling partially knows the internal structure of the domain and the causal relationships. The qualitative similarities and differences between the grey box, white box, and black box modelling paradigms are explained in Figure 2.

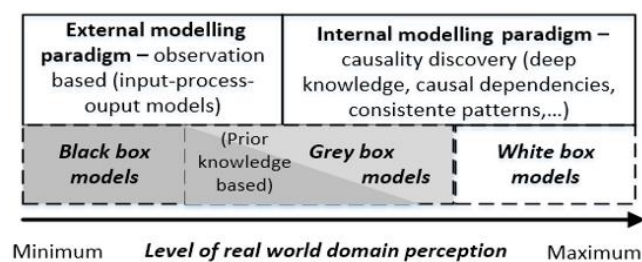


Figure 2. Different real-world modelling paradigms.

In this study, we used a causal modelling approach to Agile project management that is specified as a modified Agile activities hierarchy. Causal knowledge is a type of knowledge, next to declarative, procedural, and relational knowledge [26]. Causal knowledge is also defined as a “description of causal links among a set of factors ... which provides a means for organizations ... how best to achieve some goal” [27]. Causality is as important a theoretical concept as the concept of force in Newtonian physics according to [28,29]. A general theory of causation developed in [30,31], which underlies the theory of causal nets (TCN) developed in [32]. The understanding of causality in systems modelling can be different and depends on the nature of the knowledge used. The awareness of the theory of specific domain causality is the prerequisite for discovering regularities (consistent patterns, and causal knowledge) in a given domain. Two levels of enterprise causal knowledge for software engineering needs were introduced in [33]. The management transaction (MT) framework was used as the first level for presenting the discovered causation. At the second level, the Elementary Management Cycle (EMC) was revealed as a deep knowledge structure of MT in more detail. The causal modelling approach is suitable for discovering causal dependencies in various real-world domains.

3.3. Management Transaction Concept

We used the management transaction framework as a unified component for identifying causal knowledge (deep knowledge) of enterprise management dependencies. management transaction (MT) = (P, F, A, V) is related to some enterprise goal (G), captures the knowledge on a management function (F), enterprise process (P), informational input flow (state attributes A), and informational output flow (controls V). MT includes a feedback loop between F and P composed of A and V flows. A conceptual management transaction model is presented in Figure 3, estimating the enterprise goal (G), which is not specified explicitly in the MT definition.

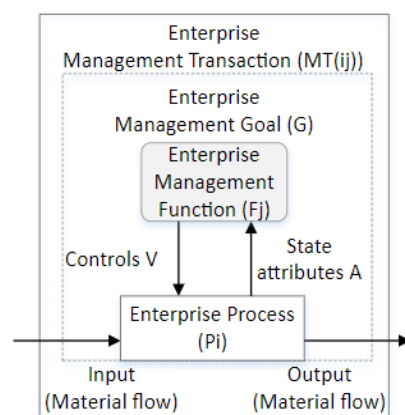


Figure 3. Conceptual model of management transaction.

3.4. Modified Agile Management Hierarchy

Management activity interactions are necessary to ensure that effort performed on EAS development projects contributes to achieving enterprise strategic objectives. Inter-

actions can involve different types of management functions, such as business strategic management (defines how the business is run, and the policies complied to) and software development management, which defines the aspects of application software development: technology stack, IT systems, software development patterns used, etc.

The particularities of the modified Agile hierarchy:

- Classifies management functions by type;
- Identifies vertical interactions and horizontal interactions between Agile project activities (theme, initiative, epic, and user story) (Figure 4);
- Any interaction between Agile activities is considered as an interaction of two complex structures (defined as MTs).

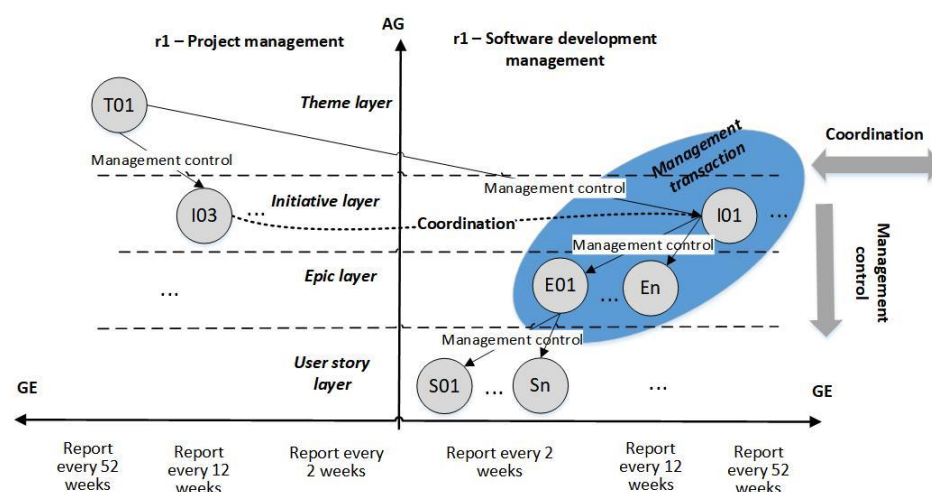


Figure 4. Example of the modified Agile management hierarchy (created by author).

Horizontal interactions are called coordination, vertical interactions are of a more complex nature and are called management control.

Execution of business strategy takes place at the level of theme depicted as T01. Two types of management functions are interacting: r1, project management; and r2, software development management. I03 is a r1 type initiative that defines the assignment to develop application software for the initiative I01 (r2 type). The I01 initiative belongs to the Agile software development management function. A coordination link is required between the initiatives I03 and I01 to ensure strategic objectives are achieved that are represented by theme T01. This means a specific set of information must be transferred between initiatives I03 and I01.

Other lower-level Agile activities E01, ..., En, and S01, ..., Sn are the software development management tasks as described previously in this article. This situation was modelled using the concept of space of processes which is explained in more detail in Section 5.1. In Figure 4 the GE axis refers to different generalization levels of information (project state reporting) linking to different time periods: 2, 12, or 52 weeks. The GE axis also indicates different levels of information generalization, e.g., report on the GE level “Report every 52 weeks” includes more general information compared to lower GE level “Report every 12 weeks”. The timeframe was described as a repeating process, meaning information must be shared every 2, 12, or 52 weeks due to the nature of continuous development that the Agile process promotes to ensure alignment between software development activities and strategic business objectives.

From the causal modelling perspective, any Agile activity is considered as a complex structure (a self-managed activity) and is defined as a management transaction (MT) = (P, F, A, V). Therefore, any interaction between Agile activities is considered as an interaction of two complex structures (defined as MTs).

4. Causal Modelling in EAS Development Management

4.1. Modified Hierarchy of Agile Activities

Analysis of traditional Agile hierarchy from the causal modelling perspective revealed that:

- Different management function types are involved in EAS development;
- There is a causal link between any two adjacent levels activities of the Agile hierarchy;
- Interaction of two activities from adjacent levels of the Agile hierarchy is a complex process with a feedback loop (there is circular causality).

By evaluating these characteristics of a real Agile process, we defined a modified Agile hierarchy. In the modified Agile hierarchy (Figure 5) any activity on each level:

- Is defined as management transaction $MT = (P, F, A, V)$, i.e., themes, initiatives, epics, and user stories have predefined structure;
- Is considered as a white box in mutual interactions in Agile hierarchy including horizontal and vertical interactions.

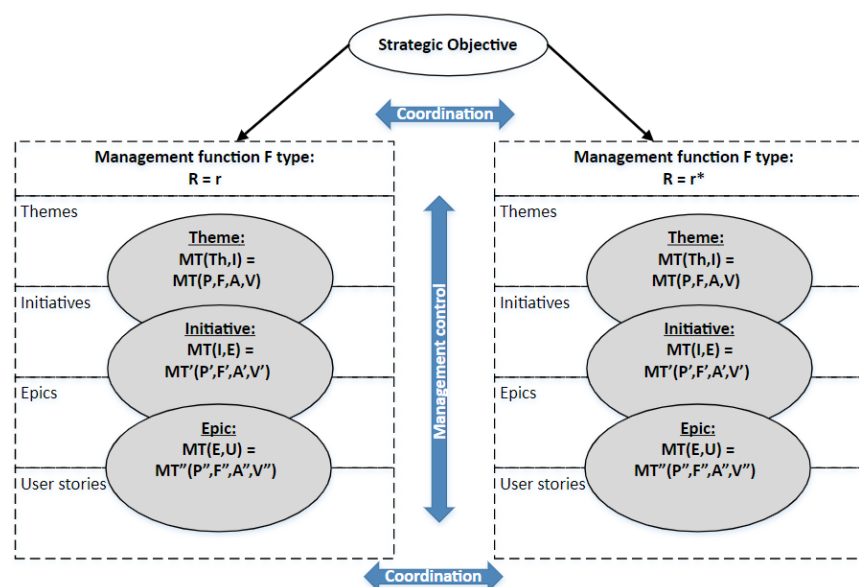


Figure 5. Modified Agile activities hierarchy as management transactions.

Any vertical interaction between activities of different levels (theme, initiative, epic, user story, etc.) is conceptualized as management transaction $MT(P, F, A, V)$, where a higher-level activity is considered as management function (role = F), relevant lower-level activities are considered as processes (role = P); also a feedback loop between higher-level activity (F) and lower-level activity (P) is predefined (flow A of state attributes and flow of controls V). Management transaction is detailed in Figure 3.

For example, the internal interaction in MT (initiative, epic) is as follows: the role of the higher-level element initiative is control function (F), and the role of the lower-level element epic is process (P). Next, the role of epic in internal interaction MT (epic, user story) changes: the role of epic here is F, the role of the user story is P.

Any horizontal interaction between activities of the same level (theme, initiative, epic, and user story) are conceptualized as a coordinating message between two management transactions $MT(P, F, A, V)$ and $MT^*(P^*, F^*, A^*, V^*)$.

A detailed specification of the content of vertical and horizontal interactions in the Agile activity hierarchy is developed. Traditional Agile activities hierarchy is a static structure and it does not consider the dynamics of the information transfers in the management system. On the opposite, MT is a dynamic structure representing the internal structure and content of interactions. The MT can be used to specify the horizontal and vertical interactions between activities at the Agile hierarchy level where:

- Agile activities (theme, initiative, epic, user story) are considered as self-managed activities (Figure 3);
- Vertical interactions at the different Agile hierarchy levels are considered as management control (Figure 5);
- Horizontal interactions are considered as coordination when the coordination message is transmitted (Figure 5).

4.2. Relativity Principle of Assigning Roles

The assignment of the type (P) to the MT element is a relative matter as well as the assignment of type (F). For example, we examined the interaction between initiative (defined as $MT(I, E) = MT(P, F, A, V)$ and epic (defined as $MT(E, U) = MT(P^*, F^*, A^*, V^*)$).

On the level of initiatives in Figure 6, initiative (I) is in the role of management function (F) and epic (E) is in the role of the process (P) in this interaction. A feedback loop (initiative, epic) includes flow A (state attributes of epic) and controls V (impact to epic). A content of flow V is the requirement focused on any element (P^*, F^*, A^*, V^*) of $MT(E, U)$ or on the combination of these elements.

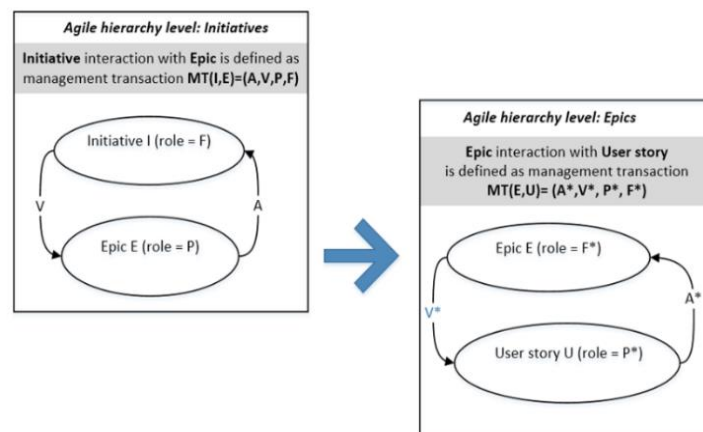


Figure 6. Example: impact of initiative (I) to epic (E) is a content of the control flow V of $MT(I, E)$.

On the level of epics, epic (E) is in the role of management function (F) and user story (U) is in the role of the process (P) in this interaction. A feedback loop (epic, user story) includes flow A^* (state attributes of user story) and controls V^* (impact to user story). Impact of epic (E) to user story (U) is content of flow V^* (controls).

The user story can also be considered as a white box and defined as $MT(U, X) = MT'(P', F', A', V')$, where X denotes any optional lower-level Agile activity.

In the same way, the relative roles are assigned to the MT elements at other levels of the Agile hierarchy. For example, in $MT(Th, I) = MT(P, F, A, V)$, a lower-level activity initiative is considered as a process (P) associated with a higher-level activity theme. The role of activity theme in this interaction is management function (F).

4.3. Vertical Interactions: Management Control

Vertical interaction of activities on any two adjacent levels in the Agile hierarchy (Figure 6) is considered as management transaction ($MT = (P, F, A, V)$). Vertical interaction between activities is considered a self-managed activity MT, with an internal management control loop between activities in two levels of the Agile hierarchy.

Therefore, each interaction of adjacent levels (theme, initiative), (initiative, epic), (epic, user story) is referred to as MT, where the role of the higher Agile hierarchy level activity is management function F and the role of the lower-level activity is process P.

Each specific F is assigned a set of specific P's and a feedback loop is specified: flows A (state attributes of process P) and control instructions V (controls).

An example in Figure 7 represents an identified $MT(I, E)$ between initiative and epic levels. Here, "I1" stands for the id of the initiative, and "E1" for the id of the epic. Type R

stands for the type of management function (i.e., software development management). $R = i$ means that the management function is the same for several adjacent Agile activities. The related Agile activities, epics (E1.1, E1.2, etc.), represent a set of related epics to the initiative I1. Accordingly, user stories (U1.1.1, U1.1.2, U1.n.1) represent a set of related user stories to the initiative I1.

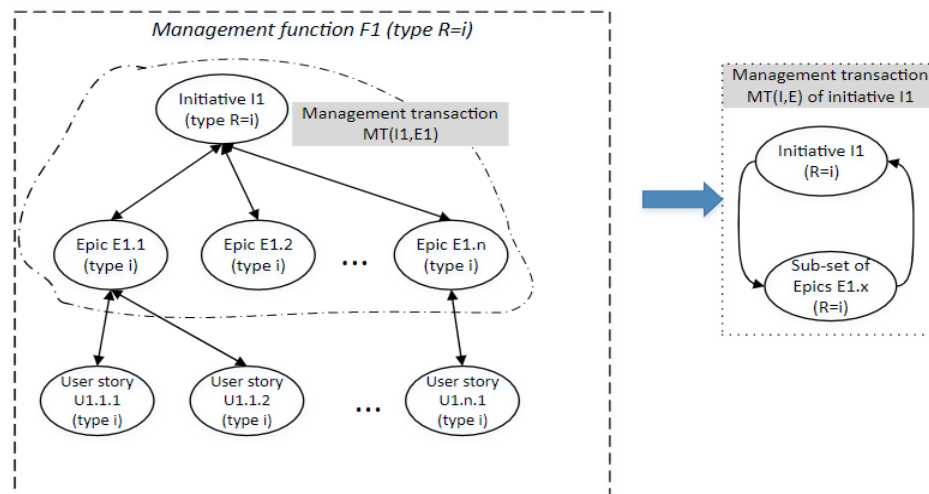


Figure 7. Identification of $MT(I, E)$ between an initiative I1 and set of Epics E1.n.

4.4. Horizontal Interactions: Coordination

Horizontal interaction between the Agile activities is called coordination. Coordination occurs on the same level of the Agile hierarchy between the Agile activities of the same type: themes, initiatives, epics, or user stories.

There are two different cases of horizontal interaction when the interaction is considered as a collaboration of two self-managed activities using coordinating messages:

- A type of management function is the same (in Figure 8);
- A type of management function is different (in Figures 9 and 10), i.e., type i = “business management”, type j = “software development management”.

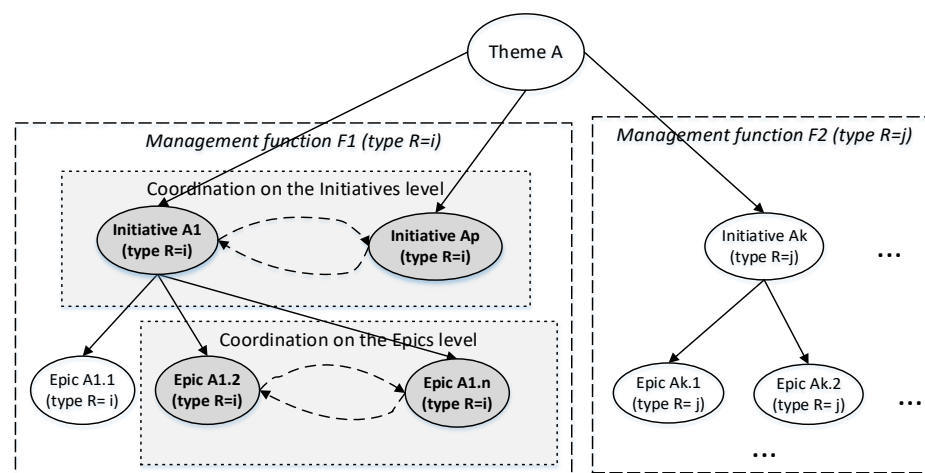


Figure 8. Horizontal interaction (coordination) of the same management function F1 (type $R = i$).

In BPMN, this communication of independent processes when a message is transmitted is called choreography.

The third case (c) is more complicated when horizontal interaction is considered as a management transaction $MT(X, X^*)$, where X and X^* are activities on the same Agile hierarchy level (e.g., themes, initiatives, epics, and user stories) also considered as management transactions (Figure 10).

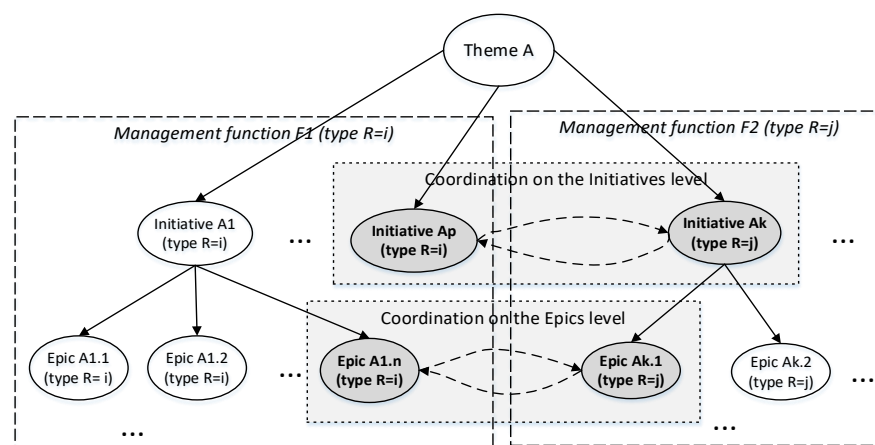


Figure 9. Horizontal interaction (coordination) of Agile activities of different management function types.

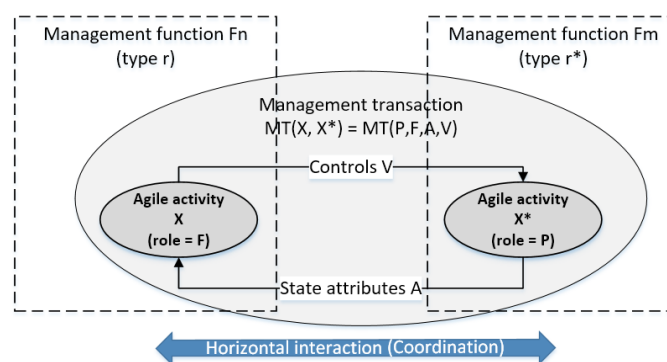


Figure 10. Horizontal interaction conceptualized as a management transaction $MT(X, X^*)$, where X and X^* are activities of the same type.

In Figure 8, “A1”, ..., “Ap”, and “Ak” stands for the identifiers (id) of the initiative, and “A1.1” ... “A1.n”, “Ak.1”, and “Ak.2” for the id of the epic. Type R stands for the type of management function. In the example, the management function R is different (on the left $R = i$, on the right— $R = j$). The related Agile activities, epics (A1.1, etc.), represent a set of related epics to the initiative A1. Coordination is occurring on the same level of the Agile hierarchy within the same management function F1 by coordination message transmission. Coordination on the epic level in Figure 8 represents the software feature development policy impact within the same management function F1. It is decided which specific feature will be built under separate epics and how those components should be reused. Coordination on the initiative level in this example represents the business management impact, i.e., message under which specific initiative a certain system or project will be developed. This is necessary to ensure that different EAS development projects do not interfere with each other, and development efforts are not duplicated.

In case when the management function is different, i.e., in Figure 9, “A1”, ..., “Ap”, and “Ak” stands for the id’s of the initiatives, and “A1.1”, ..., “A1.n”, “Ak.1”, and “Ak.2” for the identifiers of the epics. In this example, the management function type R is different (on the left $R = i$, on the right— $R = j$). In this case, the coordination is occurring on the same level of Agile hierarchy within different management functions (for example, business development management F1 and software development management F2). This example shows the business development coordination with the development of application systems used by end customers. An example of such a case is the launch of a new self-service portal for ordering banking products. On the IT side, i.e., the marketing area may provide some requirements to provide promotions, discounts for the new customers, and the IT development area must implement the requirements. The marketing area needs to prepare

marketing material for the launch of the system and the onboarding of customers. This needs to be coordinated with the IT area, as some instructions may be required.

In the general case, horizontal interaction can be conceptualized as a management transaction $MT(X, X^*)$, where X and X^* are activities of the same type (e.g., theme, initiative, epic, and user story) also considered as management transactions (Figure 10).

This assignment of Agile activity roles F and P inside $MT(X, X^*)$ is consistent with the strategic business and IT alignment model SAM [13], where certain activities are leading due to management decisions in a specific situation.

In the general case it is assumed that the main (leading) activity is X , so its role in the management transaction $MT(X, X^*)$ is F and the role of activity X^* is P .

It should be noted that in turn, activities X and X^* are also considered as management transactions defined as follows: $MT(X, Z)$ and $MT(X^*, W)$, where Z and W are the lower-level Agile activities (of a different type as X , see Figure 7). Management transactions $MT(X, Z) = MT(P'', F'', A'', V'')$ and $MT(X^*, W) = MT(P'', F'', A'', V'')$ are composite; there are internal vertical interactions due to management control.

5. Coordination Taxonomy in the Space of Processes

Coordination is the transferring of information from one coordinating activity (modelled as $MT = (P, F, A, V)$) to another (i.e., $MT' = (P', F', A', V')$) when the content of the coordinated MT' is altered.

We assume that the coordinating MT has the right:

- To transfer the flow of information (attributes, data, requirements, and constraints) from the coordinating MT to the coordinated MT' ;
- To affect the content of the MT' elements (P', F', A', V').

5.1. The Space of Processes

Types of interactions between MT s can be classified based on their mutual position in the Space of Processes (SP) regarding the type of coordinate axes (AG, GE, T): $SP = (AG, GE, T)$. Here: AG is aggregation axis, GE is generalization axis, and T is time axis.

Any $MT = MT(P, F, A, V)$ is represented as a point in the SP with coordinates $MT(i, j, t, r, p)$. The content of the information sent from one $MT(i, j, t, r, p)$ to another $MT'(i', j', t', r', p')$ in the SP depends on the difference of the position of these MT s in $SP = (AG, GE, T)$ (see Figure 11).

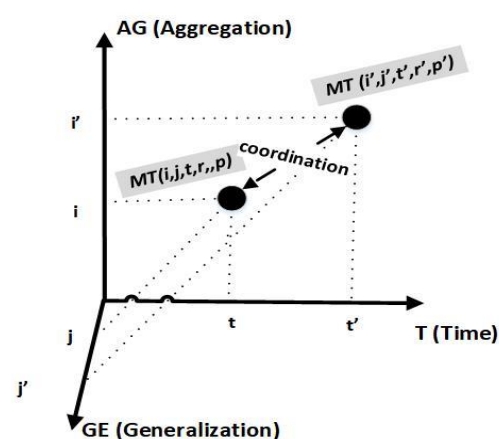


Figure 11. Coordination of MT s in the Space of Processes (AG, GE, T) (abstract case).

The mutual situation of any two MT s in the SP may be as follows [33]:

- The Aggregation axis level is the same ($i = i'$);
- The Aggregation axis level is different ($i \neq i'$);
- The Generalization axis level is the same ($j = j'$);
- The Generalization axis level is different ($j \neq j'$);

- The management function F types are the same ($r = r'$);
- The management function F types are different ($r \neq r'$);
- The process P of different MT and MT' is the same ($p = p'$);
- Processes P of different MT and MT' are different ($p \neq p'$);
- Managed processes are of the same time period ($t = t'$);
- Managed processes are of different time periods ($t \neq t'$).

5.2. Coordination Meta-Types and Types

Taxonomy of coordination meta-types introduced in [33]. MT coordination meta-types are defined by different combinations of indexes r , p , and t values, as classified in Table 1 (here: 1 is “same”, and 0 is “different”). For example, meta-type A denotes interaction of two activities in a single time period t when the same process P is managed, and management function F type R is the same, meta-type C, when management function types are different [33].

Table 1. Coordination taxonomy (meta-types) [33].

Identifiers (t, r, p)	Coordination Meta-Types							
	A	B	C	D	F	G	H	L
Type of management function (r)	1	1	0	0	1	1	0	0
Process identifier (p)	1	0	1	0	1	0	1	0
Time period (t)	1	1	1	1	0	0	0	0

5.3. Types of Coordination of Management Transactions

Based on coordination taxonomy in Table 1, coordination types are classified in Table 2. MT coordination types are defined by combining all possible combinations of indices (i , j , t , r , and p) (here: 1 is “same”, and 0 is “different”).

Table 2. Types of coordination.

Identifiers	Coordination Type															
	A0	A1	A2	A3	B0	B1	B2	B3	C0	C1	C2	C3	D0	D1	D2	D3
i	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
j	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
t	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
r	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
p	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0

Identifiers	Coordination Type															
	F0	F1	F2	F3	G0	G1	G2	G3	H0	H1	H2	H3	L0	L1	L2	L3
i	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
j	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
t	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
r	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
p	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0

Conditional names of coordination types that correspond to the reciprocal arrangement of MTs in the Space of Processes (SP) = (AG, GE, T) are as follows based on [33]:

- If the coordinating MT is on the higher aggregation level than the coordinated MT', such case of coordination is called detailed coordination because the coordinating message being transmitted must be elaborated on (in the opposite direction of axis AG);
- If the coordinating MT is on the higher generalization level than the coordinated MT', such case of coordination is called concretizing coordination because the coordinating message being transmitted must be concretized (in the opposite direction of axis GE);
- Such a case of coordination when the processes of different MT and MT' are different ($p \neq p'$) is called technological coordination.

5.4. Coordination Types in the Same Time Period

Coordination meta-types A, B, C, and D cover different coordination cases (types) in the same time period t when the values of the other MTs identifiers (i , j , r , and p) overlap or differ (Table 2). These types of coordination cover real-time monitoring and management of enterprise activity. Due to the limited scope of the article, we describe in detail only the coordination types A0, A1, A2, and A3 for the same period (where $t = t'$).

Coordination meta-type A ($t = t'$, $r = r'$, $p = p'$) comprises of four coordination types A0, A1, A2, and A3:

1. Coordination type A0 ($i = i'$, $j = j'$, $t = t'$, $r = r'$, $p = p'$): the location of the two MTs in the Space of Processes is identical ($i = i'$, $j = j'$, $t = t'$) and these MTs are associated with the same process ($p = p'$) and management function type ($r = r'$). Type A0 indicates two identical activities (duplication), one of which needs to be eliminated. A0 indicates surplus of activities;
2. Coordination type A1 ($i \neq i'$, $j = j'$, $t = t'$, $r = r'$, $p = p'$): the level of two MTs on the AG axis is different ($i \neq i'$); the values of the other identifiers are the same (overlap); an abstract example of A1 (Figure 12): initiative (AG level $i + 1$) and epic (AG level i);
3. Coordination type A2 ($i = i'$, $j \neq j'$, $t = t'$, $r = r'$, $p = p'$): the level of two MTs on the GE axis is different ($j \neq j'$); the values of the other identifiers are the same (overlap); an abstract example of A2 (Figure 12) initiatives (GE level $j + 1$) and epic (GE level j);
4. Coordination type A3 ($i \neq i'$, $j \neq j'$, $t = t'$, $r = r'$, $p = p'$): the level of two MTs on the AG axis and GE axis is different ($i \neq i'$, $j \neq j'$); the values of the other identifiers are the same (overlap).

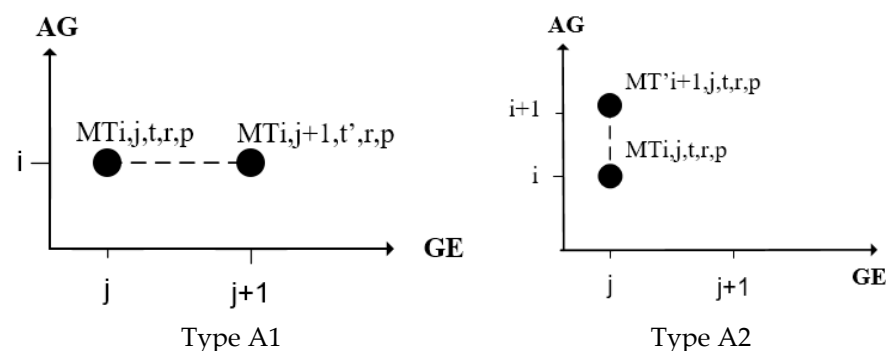


Figure 12. Coordination of types A1 and A2 of MTs in the Space of Processes.

Examples of the coordination type A1 in modified Agile management hierarchy occurring in the scope of one real project:

- Coordination type A1 between theme $Th1(i + 1, j, t, p, r)$ and initiative $I1(i, j, t, p, r)$: theme $Th1(i + 1, j, t, p, r)$ specifies a long term objective: “in 2 years to reduce cost by X amount”; linked initiative $I1(i, j, t, p, r)$: “Stop using system Y, and transfer the features of the system to new system Z”;
- Coordination type A1 between initiative $I1(i, j, t, p, r)$ and epic $E1(i - 1, j, t, p, r)$: the linked epic $E1(i - 1, j, t, p, r)$: “Transfer system Y function N to system Z in 3 months”;
- Coordination type A1 between epic $E1(i - 1, j, t, p, r)$ and user story $U1(i - 2, j, t, p, r)$: the user story $U1(i - 2, j, t, p, r)$: “in 2 weeks to transfer system Y function N component K to system Z”.

5.5. Types of Coordination in Different Time Periods

Coordination meta-types F, G, H, and L (Table 2) cover different cases (types) of MTs coordination in different time periods ($t \neq t'$) when the values of the other MTs identifiers (i , j , r , and p) are the same or different.

In enterprise governance practice, coordination of the activities from different time periods is called:

- Planning or prediction: if the implementation of one or more different MTs are to be performed from time t onwards, in the next period $(t + 1)$;
- Analysis: if one or more different MTs are considered to be performed earlier, from time t back $(t - 1)$.

In all these cases, the coordination should be performed by $MT(i, j, t, r, p)$, which is implemented in the current period t , or by creating an additional coordinating $MT^*(i^*, j^*, t^*, r^*, p^*)$ in period t . Due to the limited scope of the article, we describe in detail only the coordination types H0, H1, H2, and H3 for different periods (where $t \neq t'$):

1. Coordination type H0 ($i = i', j = j', t \neq t', r \neq r', p = p'$): coordination of MTs that have the same controlled process P during different time periods when management function type R is different, and the aggregation and the generalization levels are the same (Figure 13);

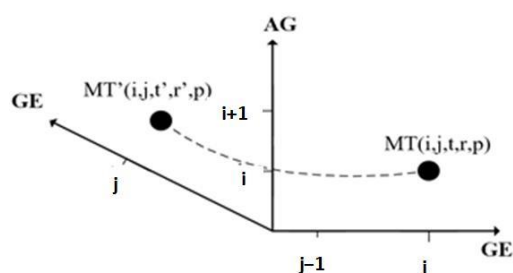


Figure 13. Coordination type H0 in the Space of Processes.

2. Coordination type H1 ($i = i', j \neq j', t \neq t', r \neq r', p = p'$): coordination of MTs that have the same controlled process P during different time periods when management function types are different, the generalization levels are different, and the aggregation level is the same;
3. Coordination type H2 ($i \neq i', j = j', r \neq r', t \neq t', p = p'$): coordination of MTs that have the same controlled process P during the different time period when management function types and aggregation levels are different, and generalization level is the same;

Considering the difference between management function types ($r \neq r'$) of MTs, it is necessary to ensure their functional compatibility. This can be performed using a common data warehouse or by coordination from the higher-level MT^* (Figure 14).

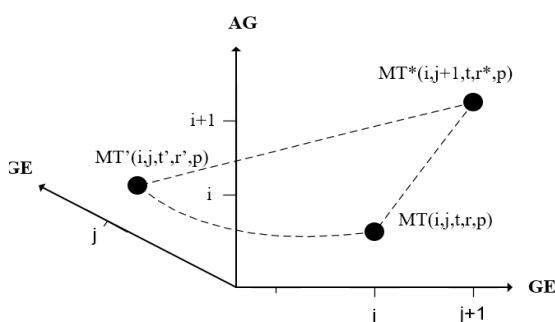


Figure 14. Coordination type H2 in the Space of Processes.

4. Coordination type H3 is a complicated case of coordination in different time periods ($t \neq t'$) when MTs have the same controlled process ($p = p'$), all the other identifiers of these MTs differ (since $i \neq i', j \neq j', r \neq r'$, and $t \neq t'$).

Functional compatibility of different MTs can be performed using a common data warehouse or coordination from the higher-level MT^* (analogous to H2).

6. Interaction Content Modelling

Modelling the content of the Agile activity interaction consists of two cases:

- Content of horizontal interaction between the same type of Agile activities: themes, initiatives, epics, and user stories;
- Content of vertical interaction between the different types of Agile activities: themes and initiatives, initiatives and epics, epics, and user stories.

6.1. Content of Horizontal Interactions

We examined the content of the horizontal interaction types described above (Figures 8–10): message transmission (choreography) between one management function activities (a), between different function activities (b), and a third case (c) where horizontal interaction is performed as a management transaction.

The content of horizontal interaction (coordination) is specified using an example of the initiative (I) horizontal impact on another initiative (I*) depicted in Figures 8–10. A causal model of initiative (I) is defined as $MT(I, E) = (P, F, A, V)$ and the other related initiative (I*) is defined as $MT^*(I^*, E^*) = (P^*, F^*, A^*, V^*)$. Possible content combinations of the coordination message (variants with different semantics) are listed in Table 3.

Table 3. Content variants for horizontal interaction flow C from MT to MT*.

MT(I, E):		MT*(I*, E*) = (P*, F*, A*, V*)		
Flow C	A*	V*	P*	F*
C1	X	-	-	-
C2	-	X	-	-
C3	-	-	X	-
C4	-	-	-	X
C5	X	X	-	-
C6	-	X	X	-
C7	-	-	X	X
C8	-	X	-	X
C9	X	-	-	X
C10	X	-	X	-
C11	X	X	X	-
C12	-	X	X	X
C13	X	-	X	X
C14	X	X	-	X
C15	X	X	X	X

In cases (a) and (b) of horizontal interaction as a choreography (Figures 8 and 9) the content of coordination flow C from initiative (I) can have an impact on the different elements (P*, F*, A*, V*) of coordinated initiative (I*). The instructions received in the message are information on the basis of which the initiative (I*) makes its own decisions, it is not obligatory to follow them.

In case (c) of horizontal interaction as management transaction (management control Figure 10) the content variations in control flow V are the same, but the instructions are mandatory and must be followed.

6.2. Content of Vertical Interactions

Vertical interaction of activities in the Agile hierarchy (interaction between activity from different levels of hierarchy) is considered as management control process defined as management transaction $MT = (P, F, A, V)$ as depicted in Figure 7.

Thus, the causal model of any Agile activity (theme, initiative, epic, and user story) is defined as MT and includes internal elements as follows: management function (F), process (P), and their feedback loop interaction with information flows (A, state attributes; and V, controls). For example, a causal model of the theme is $MT(Th, I) = MT(P, F, A, V)$ and a causal model of initiative (I) is $MT(I, E) = MT(P^*, F^*, A^*, V^*)$.

Vertical interaction content variants are listed in Table 4 by the example of theme impact to the initiative. Variants of the vertical interaction content are defined in the same way as the rest of the adjacent Agile hierarchy levels: initiative—epic, epic—user story.

Table 4. Vertical interaction content variants (V): by example of theme impact to the initiative.

MT(Th, I): Controls (V)	Initiative (I): $MT(I, E) = (P^*, F^*, A^*, V^*)$			
	A*	V*	P*	F*
V1	X	-	-	-
V2	-	X	-	-
V3	-	-	X	-
V4	-	-	-	X
V5	X	X	-	-
V6	-	X	X	-
V7	-	-	X	X
V8	-	X	-	X
V9	X	-	-	X
V10	X	-	X	-
V11	X	X	X	-
V12	-	X	X	X
V13	X	-	X	X
V14	X	X	-	X
V15	X	X	X	X

Examples of the content of V1–V15 impacts from theme to the initiative: the content of V1 is the requirements to the state attributes A* of MT(I, E); the content of V2 is the requirements to controls V* of MT(I, E); the content of V3 is the impact to process P* execution rules of MT(I, E); V4 includes requirements to management function F* (impact to management logic, rules) of MT(I, E); V5–V15 are combinations of V1–V4.

For example, impact V1 in this context imposes restrictions on the delivery date of projects. The impact V2 defines the initiative duration, the requirement that each initiative must be no longer than 9 months. Control flow V3 sets the business rules upon initiative management, i.e., the sequence of initiatives must be defined. The control flow V4 sets the requirements for the duration of the epics (3 months).

Control flow V5 sets requirements for the content of both A* and V* flows of MT(I, E), i.e., forms an impact on the feedback between the initiative (managing function) and the epics that are the controlled processes.

6.3. Model Verification and Validation

We consider the MT framework as a proven method [33,34] for capturing domain causal knowledge. The composition of the MT framework is the basis to verify the modified Agile hierarchy and to validate its application in the enhanced Jira tool.

Model verification definition in [35] states that “model verification <is> the process of assuring that a computerized model is a sufficiently accurate representation of the corresponding conceptual model.”

The interactions of levels between the modified Agile activities in Figure 5 fully match the composition of the management transaction in Figure 3, since it is required that interaction must contain four obligatory attributes: flow A of state attributes, flow V of controls, identification of process P, and management function F. In this way, the modified Agile model is verified as it fully matches the causal knowledge composition.

The process of validation determines whether a behavioral model is sufficiently accurate for Agile product owner needs in the context of tracing the integrity of themes, initiatives, epics, and user stories.

Automatic validation of causal knowledge is also based on the internal structure of the MT framework. The state of an ongoing EAS development project is checked against the defined composition of MTs. Both Tables 3 and 4 serve as validation rules for interactions in EAS projects management. This provides means for automatic validation tools (see Section 8). As an example, the fragment of the database record with the identified gaps of vertical interactions between epic E01 and user stories S01 and S02 is presented in Table 5.

Table 5. Fragment of the database record of vertical interaction evaluation.

Theme	Initiative	Epic	User Story				Validation
			F	P	A	V	
N	N	E01 Technical Tasks (785)	N	Y	Y	Y	S01 Technical architecture review (800) V11
N	N	E01 Technical Tasks (785)	Y	Y	Y	Y	S02 Segregate audit log and transaction log (903) V15

Therefore, the modified Jira tool matches the modified Agile hierarchy in Figure 5.

Here Y stands for existing information required by MT definition, and N for non-existing. Validation is automatically provided using the suggested method.

The results were also checked by the experts. The type of mismatching is identified using rules in Table 4. The advanced Jira tool automatically evaluates the information and presents a relevant error type. Our model, presented in this paper was applied for three EAS case study projects, described in [22] and significant improvement in EAS project deliveries was observed.

7. Semantics of Vertical and Horizontal Interaction

The semantics of vertical and horizontal interaction impact depends on the content of information transferred between the Agile activities. Given formal coordination classification (Section 5), it is possible to introduce the content indicator of EAS project complexity, under assumptions:

Assumption 1. The upper-level $MT(P, F, A, V)$ sends a single flow (impact and control message) to the lower-level MT^* , representing the total MT.

Under Assumption 1, the number of vertical interactions variants between adjacent levels with different content is 15 as listed in Table 4.

An example of vertical interaction content variants in Table 4 only covers the interaction between theme and initiative levels. The rest of vertical interactions are (initiative/epic), (epic/user story) and (user story/optional lower-level activity). Thus, there are four types

of vertical interactions. When combining variants in this way, the number of vertical interactions with different semantics is: $15 \times 4 = 60$.

Under Assumption 1, the number of horizontal interactions variants on the same level with different content is 15 as listed in Table 3.

An example of horizontal interaction (coordination message) content variants in Table 3 only covers interaction at the initiative level.

The rest of the horizontal interactions include interactions at themes level, epics level, and user stories level. Thus, there are four types of horizontal interactions. When combining variants in this way, the number of horizontal interactions with different semantics is: $15 \times 4 = 60$.

For a typical EAS development project, the average number of different activity types are as follows: 1 theme; 1 initiative for the theme; 10 Epics per initiative; and 240 user stories (24 user stories per epic). For a typical EAS development project, the number of vertical interactions (control messages) with different semantics is (1):

$$Q1(v) = 1 \times 15 + 10 \times 15 + 10 \times 240 \times 15 = 36,165 \quad (1)$$

That is:

- Number of relationships between theme and initiative interactions is 15, as there is only one theme—initiative relationship in the example;
- Number of relationships between initiative and epic interactions is 150, as there are 10 initiative—epic relationships in the example;
- Number of relationships between epics and user stories interactions is 36,000, as there are 10 epics and 240 user stories relationships (2400) in the example.

For a typical EAS development project, the number of horizontal interactions (coordination messages) with different semantics (an indicator of complexity Q1) is (2):

$$Q1(h) = 10 \times 15 + 10 \times 24 \times 15 = 3750; \quad (2)$$

In this case we have only one theme and one initiative; therefore, there are no horizontal interactions at these Agile hierarchy levels.

Assumption 2. Each internal element of a coordinating $MT(P, F, A, V)$ can send coordination effects to each element (P^*, F^*, A^*, V^*) of another $MT^*(P^*, F^*, A^*, V^*)$.

This Assumption 2 applies to both horizontal and vertical interactions in the modified Agile hierarchy.

An example of vertical interaction (management control message) content variants in Table 4 only covers the interaction between theme and initiative levels. Analogically, Table 3 covers the interaction between two MTs in the same hierarchy level horizontally.

The number of variants with different content is 15 for each element (P, F, A, V) of $MT(P, F, A, V)$ when sent to one specific element of another $MT'(P', F', A', V')$.

Under Assumption 2, when one MT element can send an impact to any set of other MT' elements (P', F', A', V') , combinations of (P, F, A, V) and (P', F', A', V') are formed.

The number of horizontal and vertical interactions is calculated as combinations without repetition. In the investigated case:

1. One MT element can send an impact C to any one of the other MT' elements (P', F', A', V') . The number of variants of the content (as listed in Table 3);

$$C_n^k = \frac{4!}{1!(4-1)!} + \frac{4!}{2!(4-2)!} + \frac{4!}{3!(4-3)!} + 1 = 15 \quad (3)$$

2. One MT element can send an impact C to any combination of other MT elements. (P', F', A', V') , number of combinations without repetition is 15;
3. The total number of variants of the content (with different semantics) is $15 \times 15 = 225$ when two Agile activities are coordinated with each other;

4. There are four levels in the Agile hierarchy (theme, initiative, epic, and user story).

On this basis, the project indicator of complexity Q2 for a typical EAS development project in case of horizontal interactions under Assumption 2 can be calculated as follows (4):

$$Q2(h) = 10 \times 225 + 10 \times 24 \times 225 = 56,250; \quad (4)$$

In this case, we have only one theme and initiative and they do not have any horizontal interactions.

In the case of vertical interactions under Assumption 2, the project indicator of complexity Q2 can be calculated as follows (5):

$$Q2(v) = 1 \times 225 + 10 \times 225 + 10 \times 240 \times 225 = 542,475; \quad (5)$$

We believe that the qualitative indicators Q1 and Q2 and their presented average and normalized values can help to evaluate the complexity and effort of communication and coordination in EAS project management in terms of required financing and other resources. The detailed calculations are presented in Appendix A.

8. Ensuring Coordination Using Agile Software Development Management Tools

By ensuring the causal links are defined, an internal system model is constructed. This sets the basis for improving Agile project management tools. We chose the Jira tool by Atlassian, which is the leader in the market [2], to apply the modified Agile method.

Jira uses the attribute “Epic link”, or similar to link together the activities in the TIES hierarchy (Index 1, Figure 15). Index 2 represents the added attribute Capability which we propose to use to link all the related activities in the TIES hierarchy. Capability is a well-defined strategic objective (aspect). The detailed information (Figure 16) appears after clicking on Index 2.

Figure 15. Example of Jira task with existing and proposed attributes.

Theme ID	Initiative ID	Epic ID	User Story ID	Interaction Type	Meaning	Result	Error description
Task-123	Task-436	Task-785	Task-800	V11	Technological coordination	!	No link to strategy el...
Task-125	Task-381	Task-899	Task-903	V15	Business development coordination	✓	<no error>

Figure 16. Additional Jira attributes to support business and IT alignment checking.

The report in Figure 16 is an example of the integrity checking result using the enhanced Jira tool. The interaction content combinations are checked using rules in Table 4 and the coordination result shows what information is missing. The error description is the evaluation of the missing information of the vertical interaction content V1–V14 (see Table 4).

9. Conclusions

A causal modelling approach to the Agile project management was used to rethink the traditional Agile activities hierarchy (theme, initiative, epic, and user story). The traditional Agile hierarchy does not differentiate the management function types, the requirements for the link between any two adjacent Agile hierarchy levels are not defined.

A modified Agile activities hierarchy was developed considering the interaction between two adjacent Agile levels as a complex process with a feedback loop. The conceptual model of this interaction is defined as the management transaction (MT). MT is considered a self-managed activity, which specifies the causal link interaction between two adjacent levels in the Agile hierarchy. Another feature of the modified Agile hierarchy is that the different types of control functions have been identified. Real-life examples provided of the vertical and horizontal interactions between themes, initiatives, epics, and user stories which are based on work experience with the Jira tool.

Vertical interaction of two Agile activities is considered a complex process with a feedback loop as well as horizontal interaction at the same level. The content variants for horizontal interactions (coordination) and vertical interactions (management control) are specified when each variant corresponds to a different combination of elements of the management transaction structure.

The content of interactions includes obligatory elements defined in MT structure: enterprise process (P), input flow A (process P state attributes), output flow V (process P controls), and management function (F). MT is related to some enterprise goal (G).

The taxonomy of coordination meta-types and coordination types are presented. The taxonomy is based on the conceptual structure of MT. It was used to evaluate the state of the project activities (themes, initiatives, epics, and user stories) and the content of causal interactions between them.

The content variants for horizontal and vertical causal interactions are listed and were used to evaluate the state of activities and their interactions in the EAS projects.

The modified Agile hierarchy model verification and validation issues are argued.

The developed content variants show that there are over 3700 interaction content combinations with different semantics. The semantics of vertical and horizontal interaction impact depends on the content of information transferred between the Agile activities. The number of combinations increases with every new user story, epic, initiative, or theme introduced to the EAS project. The vast amount of interaction content combinations proves the complexity of enterprise management in the Agile environment. Therefore, the project management tools should include the proposed additional attributes that would allow monitoring the project state more thoroughly.

EAS project complexity indicators were introduced that can help to evaluate the complexity and effort of communication and coordination in EAS project management in terms of required financing and other resources. The presented average and normalized complexity indicators help to evaluate the complexity of various EAS development projects and compare them.

Current Agile software project management tools such as Jira do not provide guidelines on how to define correct links between levels of the Agile management hierarchy. Given Jira is able to capture the deep causal knowledge as it is suggested in this method, it can significantly increase the capabilities of monitoring the integrity of EAS project activities. This ensures the alignment of each user story, epic, initiative, or theme to the strategic objectives and thus improves the success of business strategy execution.

Based on the results of the enhanced EAS project management tools report, the experts can provide the missing links and content of the TIES hierarchy activities. The report can also indicate that some of the EAS development project activities do not provide value to strategic business objectives thus they should be cancelled.

Author Contributions: Conceptualization, S.G.; Data curation, K.N.; Formal analysis, S.G.; Methodology, S.G.; Software, K.N.; Writing—original draft, K.N.; Writing—review & editing, S.G. and K.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

In this section, the detailed calculations of complexity indicators Q1 and Q2 are presented.

Table A1. Calculations of complexity indicator Q1 and Q2 for vertical interactions.

Formula	Theme	Initiative	Epic	User story	Interactions			Q1(v) Average	Q2(v) Average	Q1(v) Normalized	Q2(v) Normalized
					Th-I	I-E	E-U				
Q1(v)	m	n	e	u	$m \times 15$	$n \times 10 \times 15$	$e \times 240 \times 15$	-	-	$Q1(v)/Q1(v)$ average	$Q2(v)/Q2(v)$ average
Q2(v)	m	n	e	u	$m \times 225$	$n \times 10 \times 225$	$e \times 240 \times 225$	-	-	-	-
Typical project	1	1	10	240	1×15 1×225	$1 \times 10 \times 15$ $1 \times 10 \times 225$	$10 \times 240 \times 15$ $10 \times 240 \times 225$	36,165	542,475	1	1
Example project 1	1	1	20	480	1×15 1×225	$1 \times 20 \times 15$ $1 \times 20 \times 225$	$20 \times 480 \times 15$ $20 \times 480 \times 225$	144,315	2,164,725	3.99	3.99
Example project 2	1	2	40	960	1×15 1×225	$2 \times 20 \times 15$ $2 \times 20 \times 225$	$40 \times 960 \times 15$ $40 \times 960 \times 225$	576,615	8,649,225	15.94	15.94

Table A2. Calculations of complexity indicator Q1 and Q2 for horizontal interactions.

Formula	Theme	Initiative	Epic	User story	Interactions			Q1(h) Average	Q2(h) Average	Q1(h) Normalized	Q2(h) Normalized
					Th-I	I-E	E-U				
Q1(h)	m	n	e	u	$m \times 15$	$n \times 10 \times 15$	$e \times 24 \times 15$	-	-	$Q1(h)/Q1(h)$ average	$Q2(h)/Q2(h)$ average
Q2(h)	m	n	e	u	$m \times 225$	$n \times 10 \times 225$	$e \times 24 \times 225$	-	-	-	-
Typical project	1	1	10	240	-	$1 \times 10 \times 15$ $1 \times 10 \times 225$	$10 \times 24 \times 15$ $10 \times 24 \times 225$	3750	56,250	1	1
Example project 1	1	1	20	480	-	$1 \times 20 \times 15$ $1 \times 20 \times 225$	$20 \times 48 \times 15$ $20 \times 48 \times 225$	14,700	220,500	3.92	3.92
Example project 2	1	2	40	960	-	$2 \times 40 \times 15$ $2 \times 40 \times 225$	$40 \times 96 \times 15$ $40 \times 96 \times 225$	58,230	873,450	15.528	15.528

References

1. Manifesto for Agile Software Development. Available online: <https://agilemanifesto.org> (accessed on 20 October 2021).
2. 14th Annual State of Agile Report. Available online: <https://stateofagile.com/#ufh-i-615706098-14th-annual-state-of-agile-report/702749> (accessed on 27 September 2021).
3. Agile Planning with Ties with Tom Churchwell. Available online: <https://www.leadingagile.com/podcast/agile-planning-ties-tom-churchwell/> (accessed on 24 November 2021).
4. Scaledagileframework. Available online: <https://www.scaledagileframework.com> (accessed on 2 December 2021).
5. Vodde, B.; Larman, C. *Large-Scale Scrum: More with LeSS*, 1st ed.; Addison-Wesley Professional: San Francisco, CA, USA, 2016.
6. Zielske, M.; Held, T.; Kourouklis, A. A Framework on the Use of Agile Methods in Logistics Startups. *Logistics* **2022**, *6*, 19. [CrossRef]
7. Tripathi, V.; Chattopadhyaya, S.; Bhadauria, A.; Sharma, S.; Li, C.; Pimenov, D.Y.; Giasin, K.; Singh, S.; Gautam, G.D. An Agile System to Enhance Productivity through a Modified Value Stream Mapping Approach in Industry 4.0: A Novel Approach. *Sustainability* **2021**, *13*, 11997. [CrossRef]
8. Cojocaru, A.-M.; Cojocaru, M.; Jianu, A.; Bucea-Manea-Tonis, R.; Paun, D.G.; Ivan, P. The Impact of Agile Management and Technology in Teaching and Practicing Physical Education and Sports. *Sustainability* **2022**, *14*, 1237. [CrossRef]

9. Saeedi, K.; Visvizi, A. Software Development Methodologies, HEIs, and the Digital Economy. *Educ. Sci.* **2021**, *11*, 73. [\[CrossRef\]](#)
10. Alkhafaji, A.; Nelson, R.A. *Strategic Management Formulation, Implementation, and Control in a Dynamic Environment*, 1st ed.; Routledge: London, UK, 2003.
11. Van de Ven, A.H.; Delbecq, A.L.; Koenig, R., Jr. Determinants of Coordination Modes within Organizations. *Am. Sociol. Rev.* **1976**, *41*, 322–338.
12. Taxen, L.; Riedl, R. Understanding Coordination in the Information Systems Domain. *JITTA* **2016**, *17*, 5–40.
13. Henderson, J.C.; Venkatraman, N. *Strategic Alignment: A Model for Organization Transformation via Information Technology*; Working Paper 3223–90; Massachusetts Institute of Technology: Cambridge, MA, USA, 1990.
14. Van Eck, P.; Blanken, H.; Wieringa, R. Project GRAAL: Towards operational architecture alignment. *Int. J. Coop. Inf. Syst.* **2004**, *13*, 235–255. [\[CrossRef\]](#)
15. Gerow, J.E.; Thatcher, J.B.; Grover, V. Six types of IT-business strategic alignment: An investigation of the constructs and their measurement. *Eur. J. Inf. Syst.* **2015**, *24*, 465–491. [\[CrossRef\]](#)
16. Business Modeling Category-Specifications Associated. Available online: <https://www.omg.org/spec/category/business-modeling/About-business-modeling/> (accessed on 6 December 2021).
17. Larman, C.; Basili, V.R. Iterative and incremental development: A brief history. *Computer* **2003**, *36*, 47–56. [\[CrossRef\]](#)
18. Cohn, M. *User Stories Applied: For Agile Software Development*, 1st ed.; Addison Wesley: Boston, MA, USA, 2004.
19. Rubin, K. *Essential Scrum: A Practical Guide to the Most Popular Agile Process*, 1st ed.; Addison-Wesley Professional: Boston, MA, USA, 2016.
20. Epics, Stories, Themes, and Initiatives. Available online: <https://www.atlassian.com/agile/project-management/epics-stories-themes> (accessed on 17 December 2021).
21. McDonald, K. *Beyond Requirements: Analysis with an Agile Mindset (Agile Software Development Series)*, 1st ed.; Addison-Wesley Professional: Boston, MA, USA, 2015.
22. Noreika, K.; Gudas, S. Using Management Transaction Concept to Ensure Business and EAS Alignment in an Agile Environment. In *Information and Software Technologies. ICIST 2021. Communications in Computer and Information Science*, 1st ed.; Lopata, A., Gudonienė, D., Butkienė, R., Eds.; Springer: Cham, Switzerland, 2021; Volume 1486, pp. 109–120.
23. Schwaber, K.; Sutherland, J. *The Scrum Guide the Definitive Guide to Scrum: The Rules of the Game*, 1st ed.; Scrum.org: Burlington, MA, USA, 2020.
24. Noreika, K. Improving Enterprise Application Software Development Management with MODAF. In Proceedings of the BIR 2021 Workshops and Doctoral Consortium Co-Located with 20th International Conference on Perspectives in Business Informatics Research (BIR 2021), Vienna, Austria, 22–24 September 2021.
25. Francis, B.A.; Wonham, W.M. The internal model principle of control theory. *Automatica* **1976**, *12*, 457–465. [\[CrossRef\]](#)
26. Gudas, S. Causal Modelling in Enterprise Architecture Frameworks. *Informatica* **2021**, *32*, 247–281. [\[CrossRef\]](#)
27. Zack, M.H. If Managing Knowledge is the Solution, then What's the Problem? In *Knowledge Management and Business Model Innovation*, 1st ed.; Malhotra, Y., Ed.; Idea Group Publishing: Hershey, PA, USA, 2001; Volume 1, pp. 16–36.
28. Glymour, C. Review of Making Things Happen: A Theory of Causal Explanation, by J. Woodward. *BJPS* **2004**, *55*, 779–790. [\[CrossRef\]](#)
29. Schurz, G.; Gebharder, A. Causality as a theoretical concept: Explanatory warrant and empirical content of the theory of causal nets. *Synthese* **2016**, *193*, 1073–1103. [\[CrossRef\]](#)
30. Pearl, J. *Causality: Models, Reasoning, and Inference*, 1st ed.; Cambridge University Press: Cambridge, UK, 2000.
31. Pearl, J. Causal inference in statistics: An overview. *Stat. Surv.* **2009**, *3*, 96–146. [\[CrossRef\]](#)
32. Spirtes, P.; Glymour, C.; Scheines, R. *Causation, Prediction, and Search*, 2nd ed.; The MIT Press: Cambridge, MA, USA, 2000.
33. Gudas, S. *Foundations of the Information Systems' Engineering Theory*, 1st ed.; Publishing House of Vilnius University: Vilnius, Lithuania, 2012.
34. Gudas, S.; Lopata, A. Towards internal modelling of the information systems application domain. *Informatica* **2016**, *27*, 1–29. [\[CrossRef\]](#)
35. Malak, R.J., Jr. A Framework for Validating Reusable Behavioral Models in Engineering Design. Master's Thesis, Georgia Institute of Technology, Atlanta, GA, USA, 2005.