

Modelling the Alignment Between Agile Application Development and Business Strategies

Karolis Noreika^[0000-0003-4062-2149] and Saulius Gudas^[0000-0002-7835-5149]

Institute of Data Science and Digital Technologies, Vilnius University,
Akademijos 4, Vilnius
{karolis.noreika, saulius.gudas}@mif.vu.lt

Abstract. Enterprise application software (EAS) needs to be aligned with the long-term goals of the organization to fully support business operations. Agile software development management tools like Atlassian “Jira” do not ensure required integrity between the strategic business objectives and lower-level items (themes, initiatives, epics, user stories, tasks, etc.). The aim of the paper is to describe a model-driven approach of Agile software tools enhancement with integrity checking functionality. The complexity of the problem lies in transforming the declared business strategies into well-defined structures suitable for model-driven development methods. The first task is mapping required business strategies to well-defined application design models using MODAF concepts and products. The content of Agile concepts is defined using the concept of capability and related MODAF models. In the next step, the interactions between the concepts in the dynamic Agile hierarchy are formalized using management transaction (MT) concept. The advantage of our approach is the usage of well-defined frameworks to identify the content of Agile concepts and their interactions in the application development to ensure alignment. This is proved by providing an example of improved usability of “Jira” concepts and Agile software management hierarchy.

Keywords: Agile development, MODAF, Management Transaction, Business and IT alignment, Model-driven development

1 Introduction

The model-driven enterprise application software (EAS) development focuses on the content of causal links of the business domain. EAS must be aligned with the long-term goals of the organization and fully support business operations. This ensures that the business operates efficiently and provides a competitive advantage over its competitors. Businesses regularly review and define strategic business objectives to set long-term goals and support vision and mission statements. Various information systems are considered an integral part of business operations and ensure that a business can operate in its business area. However, continuous changes in the business environment, various compliance and legal requirements from government authorities require constant im-

provement of existing business processes. These reasons, together with technical novelties, require making changes to existing and create new information systems. EAS supports business processes over the big scope for interlinked business units and their processes. The development of such systems requires deep business process knowledge. Managing development changes via EAS projects is an integral part of the business development area in most enterprises. By having extensive knowledge and experience working with the “Jira” package, it was observed that the link between strategic business objectives and Agile management items like themes, initiatives, epics, and user stories is not fully provided in “Jira”.

Standish Group, a leading project management statistics provider, shares publicly available results from 2011 to 2015 that successful projects only constituted up to 31% [1]. The definition “successful” included the time, scope, and budget constraints and also that the stakeholders are satisfied with the outcome. Project management researches by KPMG, AIPM and IPMA [2] indicate that only up to 44% of EAS projects are likely to be delivered that meet the original goal and business intent, only up to 36% on budget and up to 30% on time. Although Agile methods are becoming more popular, this means that business and IT alignment is still an important field of research.

Agile methods for software development management are extremely useful in constantly changing business conditions where the environment is ambiguous and the scope of the requirements is highly likely to change over the period of the project. It is proven by multiple researches, like the “14th State of Agile” report, that when using Agile methods, the projects are more successful with business value delivered in 46% and customer/user satisfaction is achieved in 45% of Agile projects [3]. Although Agile methods improve the success of software development projects, using only Agile methods proves not to be enough to make sure development done during information systems projects is contributing to strategic business objectives. Applying Agile methods for software development project management is an important part of the success together with the proper usage of software development management tools. In this field, “Jira” from Atlassian is a clear market leader with 67% of respondents from the “14th State of Agile” report indicating it as the primary tool to manage Agile-based software development management projects and 78% recommending it for other professionals.

The levels in the Agile management hierarchy from themes to user stories are also known as TIES, standing for themes, initiatives, epics, and user stories [4]. But the links between TIES structure elements and also the strategic business objectives are currently defined by human interaction and communication between project managers, Agile leaders, and business managers. Additional definition of coordination between business and IT management is required to ensure business and IT alignment. This is also not resolved by following the traditional requirements traceability approach [5]. Although it provides some structured way to map the requirements to business needs and project objectives, it still does not capture the causal knowledge of the domain. Discovering causal knowledge is essential to manage the complexity of business and IT alignment.

We propose an approach to support the links between strategic business objectives and themes, initiatives, epics, and user stories using the concept of management transaction and enterprise architecture frameworks.

2 Related Works

2.1 Agile Software Management Tools and Frameworks

Many software tools are developed in the market to support the Agile software project management process. One of the leaders is Atlassian’s “Jira” tool [3]. “Jira” has many customization options, including setting up dashboards and enabling additional add-ins. However, the current state of “Jira” software only supports the themes, initiatives, epics, and user stories structure. It does not provide clear practices for making sure the links between different levels of Agile hierarchy are established properly based on business context. It is not ensured that each user story, epic, initiative, and theme links to one of the strategic business objectives. Links are determined by experts working on project delivery. There is no verification to ensure that i.e. each user story is linked and integrated properly to the theme on the highest level of the Agile project management hierarchy.

Microsoft “Azure DevOps” is also one of the most popular Agile software project management tools, mainly due to its vast capabilities for managing software development itself – i.e. CI\CD, code writing, pipelines, etc. However, the structure for Agile software project management in Microsoft “Azure DevOps” is simple and does not contain the business context knowledge in any formal way [6]. Microsoft “Azure boards” – a part of Microsoft “Azure DevOps” uses the structure of epics, features, user stories, and tasks.

Big amount of informational units needs to be handled and coordinated during the Agile software development project to achieve the strategic business objectives. As an example, information elements distribution from a real-world project is presented in Table 1.

Table 1. Agile software project management levels

Hierarchy level	Agile concept	Description	Number of objects
1	Strategic business objective	Short statement explaining long term business goal and having link to vision and mission of the company.	1
2	Theme	High level objective with time constraint and usually monetary goals, that allows to achieve strategic business objective.	1-5
3	Initiative	Specific activity to contribute to achieving goals on the theme level.	5-15
4	Epic	Functionality description to achieve goals of linked initiative.	>300
5	User story	Task representing business need or problem.	>3000

Like the work breakdown structure (WBS) used in traditional project management [7], projects managed using Agile methods have their own definitions that represent the hierarchy of tasks in an EAS project. Agile software project management professionals and vendors use various naming for the elements in the hierarchical structure. The lowest level of granularity usually is defined by the concept of “user story”. The intent behind using user stories is to provide not too detail and also not too wide description of the business problem. Using it, both IT and business representatives can have a medium to share ideas that would lead to the solution of a problem. User story concept was originated by Connextra in the United Kingdom and popularized by Cohn, M. [8]. There is research done on the benefits and limitations of using user stories [9, 10]. User stories should span no longer than the single development iteration (1 to 4 weeks).

The next level in the Agile software project management hierarchy is epic. The definition of epic varies from “a user story that is bigger than the development capacity for the development iteration” [8], [11], or just “a group of related user stories” [12] to a less definite but more end result-oriented definition like in [13], stating that “epics are really large stories that align with the product vision and provide the next level of product detail for senior management”. Epics should span no longer than a calendar quarter or up to 12 weeks.

Going further up from the most detailed level of Agile software project management of user stories and later epics, the next level is defined as initiatives. Atlassian, the vendor of various tools for software development management, including “Jira” tool, states that an initiative is “a collection of epics that drive toward a common goal” [14]. Other sources indicate that initiative is about broad focus and significant impact to company’s performance and initiatives “provide business context for decision-making and help you navigate the course of your organization” [12]. Initiatives provide both business and IT perspective on achieving business goals and should not span more than 1 year.

Theme (also called feature) is the top level in Agile software project management. It can be defined as “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” [15]. Theme provides a convenient way to indicate that a set of stories have something in common, such as being in the same functional area” [13] or just “large focus areas that span the organization” [14]. But themes are business objectives translated into more specific means to achieve those business objectives and can be considered as the level between initiatives and strategic business objectives.

Scaled Agile frameworks like SAFe [16] or LeSS [17] do not formally describe the internal links between their own equivalents of the TIES structure elements, therefore we propose an approach to solve this gap.

2.2 Enterprise Architecture Modelling

Enterprise architecture (EA) frameworks are used for conducting enterprise analysis, design, and implementation of relevant information systems necessary to execute business strategies. It helps organizations to go through business and technology changes. Enterprise architecture frameworks (MODAF, DODAF, NAF, and others) provide a set

of principles, guidelines, and models to ensure the alignment throughout the entire enterprise from a business and technology perspective. An approach to business and IT alignment in an Agile environment using concepts from Scrum, Scaled Agile frameworks and ArchiMate was presented in [18].

Business knowledge is captured as models using Enterprise Architecture frameworks, but special attention should be dedicated to also capturing the causal knowledge [19]. Causal knowledge is a “description of causal links among a set of factors . . . which provides a means for organizations . . . how best to achieve some goal” [20]. Causal modelling is suitable for discovering causal dependencies in various real-world domains. The causality-based approach to EA development (e.g. under MODAF) introduced in [19] links three modelling perspectives: causal modelling (real world do-main modelling), structural modelling (EA framework, using MODAF products as an example) and meta-modelling (causal meta-models of EAF). Based on the same causality-based approach, we construct an Agile business and IT alignment method, linking three modelling perspectives: causal modelling (Agile process modelling), structural modelling (EA framework, using MODAF products as an example) and meta-modelling (causal meta-models of interaction in Agile hierarchy).

Two levels of enterprise causal knowledge modelling introduced in [21]. The first level is the presentation of the discovered causation using the Management Transaction (MT) framework. At the second level, a deep knowledge structure of MT is revealed in a more detailed framework called the Elementary Management Cycle (EMC). We use only MT framework as a unified component of causal knowledge (deep knowledge) for an enterprise management model. MT is relevant to some enterprise goal (G) and captures knowledge on enterprise process (P), feedback loop (F, P), informational input flow (A), informational output flow (V) and management function (F): $MT = (P, F, A, V)$. The conceptual structure of MT is presented in Fig. 1.

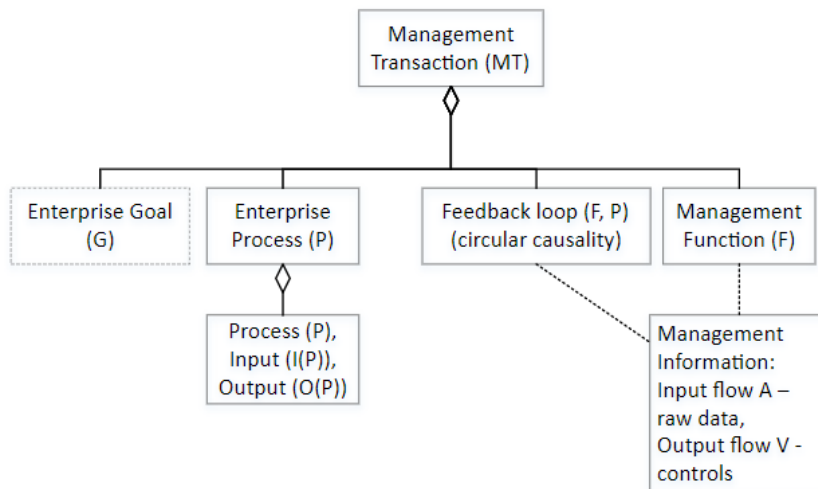


Fig. 1. Conceptual structure of Management Transaction

British Ministry of Defence Architecture Framework (MODAF) is an Enterprise Architecture framework that is used in model-based engineering (MBE) of complex systems. It uses the concepts of views (Strategic, Operational, System, Acquisitions, Technical Standards, and All views) together with a set of models (products) that help model the entire enterprise and its operations. MODAF is the most widely known and established framework. One of the key concepts is the “capability” concept. A capability is described by MODAF as “a specification or classification of the ability of the enterprise” [22]. The strategic views are focused primarily on capability management. Capabilities change over time, but the main idea is that capabilities enable the organization to act and succeed in the preferred domain based on the skill set and abilities the enterprise has. Capability is the main concept from the MODAF that we believe is missing in Agile software project management tools to ensure the alignment of strategic business objectives to the Agile software project development tasks.

3 Mapping of Agile Software Management Hierarchy to Enterprise Architecture Framework Concepts

Business strategy execution is a complex process of acting on the defined strategic plan in an effort to achieve strategic business goals that are derived from the vision and mission statement of the organization. It is an ongoing activity during the whole lifespan of the enterprise. When the business execution process goes from strategic objectives to themes in the Agile management TIES context, market researches are done to ensure the planned themes are what the market is expecting from the enterprise. Also, enterprises evaluate their capabilities, strengths to ensure these are utilized in order to achieve strategic objectives. Typically in the business execution process achieving a strategic objective takes effort from many various business functions like marketing, logistics, etc., and very often – IT. New or current improved IT solutions directly make an impact to achieve strategic objectives as customers need service on-demand with fewer approvals or any other delays in their experience.

When the business execution process goes from themes to initiatives, business function managers are evaluating the relevant capabilities and distribute the high-level tasks to appropriate department leaders. In the case of the IT department, this could be reducing costs through “sun setting” a list of legacy applications by either implementing features in more modern EAS or by optimizing the business process so that it does not rely on the legacy system. Once moving through the business strategy execution process from initiatives to epics this will be units of a smaller work effort like replacing a feature in some old legacy software with more modern tech stack tools. A single task to contribute to is usually called a user story and reflects the business need from the stakeholders perspective. In the example case, this may be just moving some software component to a new tech stack.

The described business strategy execution process could be presented as a BPMN diagram (Fig. 2)

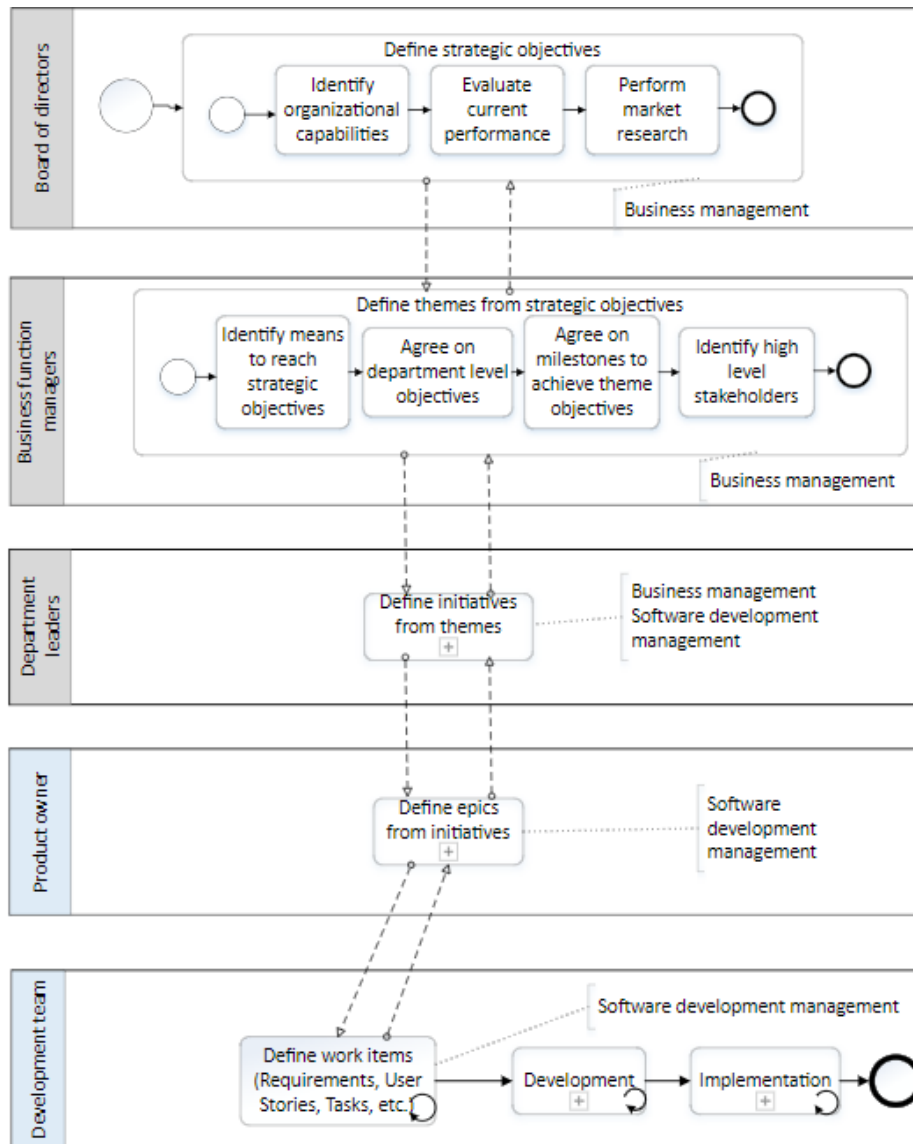


Fig. 2. Business strategy execution process

“Development” in Fig. 2 means the software development activities: engineering, coding, and testing the EAS or its components. “Implementation” in Fig. 2 means following the established procedures to put the tested solution (the result of development) to production environment that is readily available for end-users. The message flows between separate lanes indicate the interaction of information between different levels of Agile hierarchy and contain a feedback loop as presented in Fig. 1. Agile project

management process can also be described as fuzzy. Agile project management practices and tools do not provide a structured and formalized approach to define interactions in the TIES hierarchy. Mapping of business strategy execution to well-defined (structural) application design models is carried out using MODAF products: Strategy view models (StV-2 Capability Taxonomy, StV-4 Capability Dependencies, StV-5 Capability to Organization Deployment Mapping).

This ensures the transforming of the declared business strategies to well-defined structures, starting with the capability concept. This way, the content of Agile concepts (Theme, Initiative, Epic, User story) is defined using the concept of capability and related MODAF models (Table 2).

Table 2. Alignment of Agile and MODAF concepts

Agile concepts	MODAF products and concepts	
	View	Elements
Theme	StV-1	Capability
	StV-2	Capability dependence
Initiative Epic	StV-6	Operational Node
	OV-2	Operational Activity
		Operational Activity Flow
Epic User story, Task Change request Bug	OV-5	Operational Activity
		Operational Performer
		Operational Role
		Operational Activity Flow

Agile management concepts mapping to MODAF products from Table 2 is required to specify the internal structure (content) of Agile concepts using StV-1, StV-2, StV-6 models. This will help to reveal the integrity of the interactions between the Agile process levels.

The focus of the paper is on the level of “Department leaders” as in Fig. 2, where the actual business and IT alignment is happening. This requires additional coordination between business management and software development management activities as displayed in Fig. 3

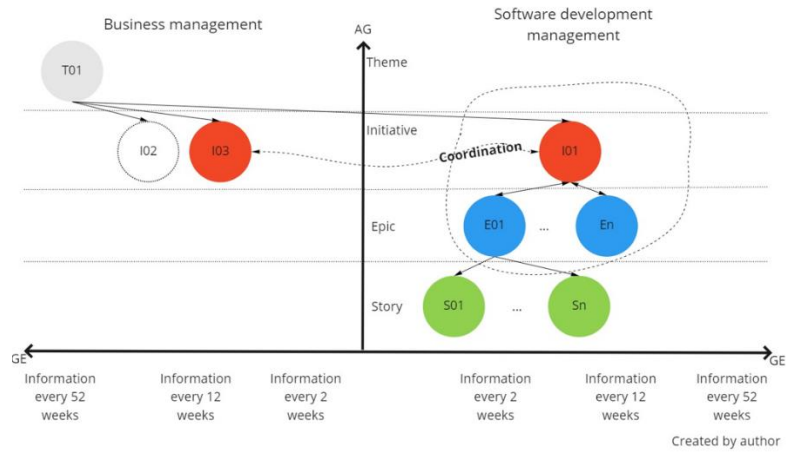


Fig. 3. Business management and software development management coordination relationship

Business strategy execution is happening on the theme level depicted as T01. I03 is a business management initiative that defines assignment I01 for enterprise application development. A coordination link is required between the business management process in initiative I03 and the Agile software development management process in initiative I01. Agile activities I01, E01..En, and S01..Sn are the software development management task hierarchy TIES as described previously in this paper.

To further detail the content of coordination link between Agile concepts initiatives I03 and I01 a mapping to management transaction concepts is required. MODAF products and concepts column in Table 2 shows that we can model $MT = (P, F, A, V)$ in more detail. MT is causality based framework representing the grey/white –box approach which specifies the causal dependencies of the problem domain.

Agile concepts hierarchy is a static structure in terms of its components: themes, initiatives, epics, and user stories, and it does not represent the dynamics of the system. In our approach enterprise architecture is also considered as a static structure that details Agile concepts in this step of transformation. However, MT is a dynamic structure, capable of representing internal structure and interactions of items:

- a) MT represents an internal structure of the Agile concepts (Theme, Initiative, Epic, User Story), if concepts are defined as self-managed activities;
- b) MT represents the internal structure of the interactions between adjacent Agile hierarchy levels (Theme – Initiative; Initiative – Epic; Epic – User Story), if interactions between adjacent Agile hierarchy levels are considered as self-managed activities. In this case content of interactions (feedback loop) between adjacent Agile hierarchy levels is revealed.

In this paper we only investigate application b) of MT. We consider initiatives and themes are highly abstract and they need to be mapped to capability concept. Capability concept from MODAF is sufficient to represent the fuzzy information from the theme and strategic objective level. The mapping is presented in Table 3.

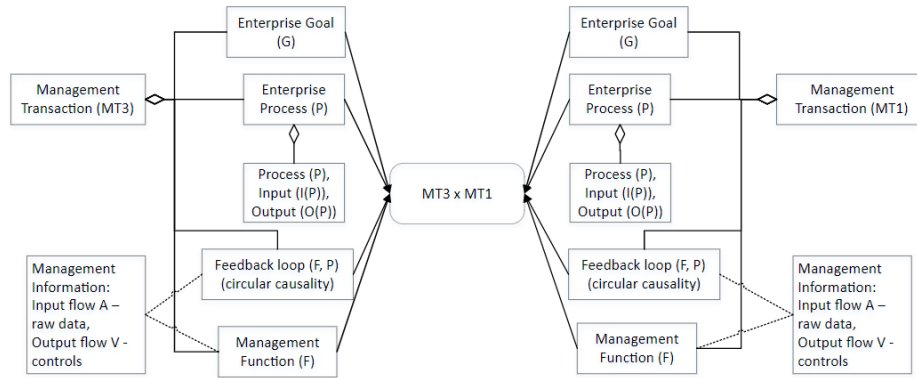
Table 3. Mapping of Agile concepts to MT and MODAF concepts

	Static view	Dynamic view
Agile items mapping to MT elements	MODAF concepts mapping to MT	MT elements
Theme (T): T := G;	StV concepts: Enterprise Phase (EP), Capability (C): (EP, C) := (G)	Goal (G)
Initiative (I): I := MT (F, P, (A,V));	OV concepts: Operational Node (N), Operational Activity (O), Operational Activity Flow (AF): (N, O, AF) := (F, P, (A,V));	Goal (G) Management function (F) Process (P) – (P1, P2, ...,Pn) Information flows (A,V)

The mapping in Table 3 represent the logical links between different levels of Agile hierarchy. They do not represent any software development realization.

Having mapped Agile hierarchy elements to the defined and structured MODAF elements, the coordination relation between different management function elements could be further defined. Each element on the management transaction for business management (MT3) and management transaction for software development management (MT1) can have links with one another. For example, on the business management process management function MT3, the goal could be i.e. ensure faster document signing by enabling digital signing. MT3 management function is coordinating with the enterprise goal (G) of the software development management transaction MT1 enterprise goal (G) ensuring software development management process supports the business management process goal. While developing the solution in the software development management process a feedback loop from MT1 is constantly providing status progress and receiving feedback from the business management process management transaction MT3 enterprise process (P) element.

The scope of all possible combinations of MT3 elements impact to MT1 elements is depicted in Fig. 4 as Cartesian product (MT3 x MT1), but based on our experience usually only a few combinations are used as in this example: MT3(G) <--> MT1(G) and MT3(P) <--> MT1(F, P).



Created by author

Fig. 4. Business management and software development management coordination content

4 Ensuring Coordination Using Agile Software Development Management Tools

To support the coordination requirement between business management and software development management processes, additional information need to be provided in Agile software development management tool like “Jira”. “Jira” has vast customization options for Agile software development project management. “Jira Portfolio” – an Agile software project management governance tool can be used to represent the TIES structure. It contains the WBS structure of the whole project scope and timelines, based on input from users. The example of how “Jira Portfolio” supports the TIES structure is displayed in Fig 5.

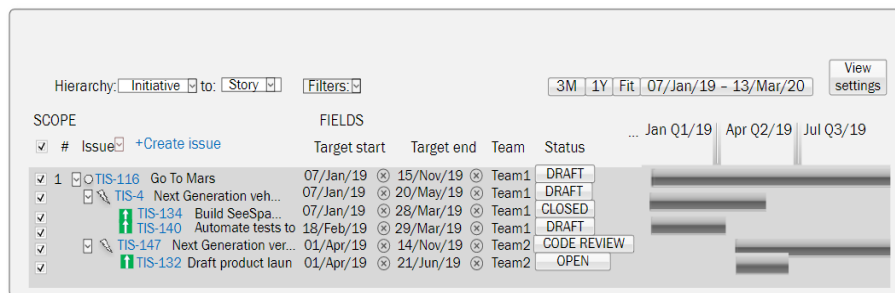


Fig. 5. “Jira Portfolio” example displaying TIES structure

This view contains the necessary information to start the coordination discussions between related business and software development management processes, however it does not contain the causal knowledge, regarding the links between different hierarchy levels of user story, epic or initiative.

The information on the initiative, epic or user story varies based on the level of hierarchy. Initiative has links to related epics, epics have links to related user stories. Typical fields include name, description and other details like creation date, user, etc.

Analyzing the structure of “Jira” views it was observed that there is no designated attribute to provide the causal link between business management and software development management processes. Based on the information in Fig. 2 the links between these levels of Agile management hierarchy needs to be ensured: “board of directors” and “business function managers”, “business function managers” and “department leaders”, “department leaders” and “product owner”, “product owner” and “development team”.

The concept of “capability“ in MODAF needs to be used to trace links of individual software development tasks as described in Fig. 2: theme, initiative, epic and user story. And these links of Agile activities are well specified using MODAF products StV-1, StV-2, StV-4, StV-5, StV-6, OV-2, OV-5 as presented in Table 2.

Fig. 6 depicts the view in “Jira” where “1” indicates the current attribute to specify link between related Agile hierarchy elements. However, as this is not sufficient to ensure each task contributes to a strategic business objective, therefore an additional attribute “2” is suggested to be added. The contents of the new field can be provided automatically based on the models in MODAF and the related mapping as defined in Table 2 and Table 3.

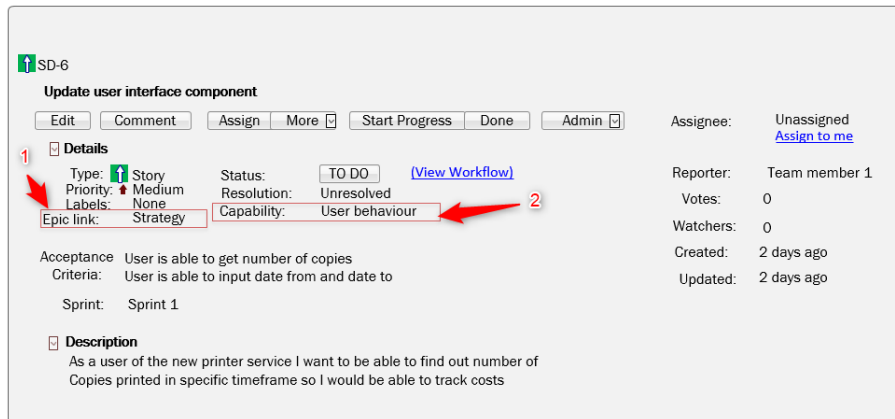


Fig. 6. “Jira” view with additional attribute

5 Conclusions

Business strategy execution is a complex process and the development of EAS is an integral part of it. Various other methods do not ensure that business strategy is aligned with enterprise application development tasks. Therefore, a more formal approach is required to ensure strategic business objectives are achieved while developing enterprise application software. Using only Agile methods and tools for software develop-

ment management proves not to be enough to ensure business and IT alignment. Current Agile software project management tools like “Jira” do not ensure sufficient alignment between the levels of Agile software project management hierarchy and do not provide methods or guidelines to capture the deep causal knowledge behind the levels of Agile software project management hierarchy.

The paper focuses on the mapping of Agile software management hierarchy items (themes, initiatives, epics, user stories) to the views (strategic, operational) of the enterprise architecture framework MODAF to integrate the top-down and bottom-up transitions between layers of Agile software management hierarchy. The causality-based Agile business and IT alignment modelling approach, presented in the paper, interlinks three modelling perspectives: causal modelling (Agile process modelling), structural modelling (EA framework, using MODAF products as an example) and meta-modelling (causal meta-models of interaction in Agile hierarchy). Capability is used here as a key concept for transition to structural modelling of the Agile hierarchy items.

Management transaction (MT) framework is a proven approach to capture causal domain knowledge. Using MT and MODAF Strategy and Operational views, proved to be a sufficient way to capture the strategic capabilities of the enterprise. Using the “capability“ concept allows specifying business strategy in more detail way, tailored for software application engineering.

An example to reveal the content of information required for coordination between business management and software development management processes is provided and this is formalized in 3D view to identify the situation. We proposed a formal visualization of Agile management hierarchy coordination links that are now specified in an abstract process space = AG, GE, time. This helps to identify and segregate the different management function types, i.e. business management and software development management. Agile activities as defined in TIES are now classified based on management functions. This helps to formally define the interaction between different management functions horizontally and between different levels of Agile activities hierarchy (themes, initiatives, epics, user stories) vertically.

Having formally specified the links between business management process and Agile software project management activities helped to detect anomalies in “Jira” attributes list and functionality. We propose a new attribute “capability” that can be automatically defined based on enterprise architecture models from MODAF to ensure correct link throughout the whole business strategy execution process. Furthermore this allows ensuring integrity on the business function level horizontally and on the Agile software project management process hierarchy vertically.

References

1. Standish group: Chaos report 2015, https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf, last accessed 2021/06/10.
2. KPMG, AIPM, IPMA: The future of Project management: global outlook 2019 <https://www.ipma.world/assets/PM-Survey-FullReport-2019-FINAL.pdf>, last accessed 2021/07/12.

3. Digital.ai Software, Inc.: 14th Annual State of Agile Report, <https://stateofagile.com/#ufhi-615706098-14th-annual-state-of-agile-report/702749>, last accessed 2021/03/27.
4. Leading Agile, D. Prior.: Agile Planning With Ties With Tom Churchwell., <https://www.leadingagile.com/podcast/agile-planning-ties-tom-churchwell/>, last accessed 2021/06/20
5. Gotel, O.C.Z., Finkelstein, C.W.: An analysis of the requirements traceability problem. In: Proceedings of IEEE International Conference on Requirements Engineering, pp. 94-101. IEEE, USA (1994), doi: 10.1109/ICRE.1994.292398.
6. Microsoft: How SAFe concepts map to Azure Boards artifacts, <https://docs.microsoft.com/en-us/azure/devops/boards/plans/safe-concepts?view=azure-devops&tabs=agile-process>, last accessed 2021/07/17.
7. Project Management Institute: A Guide to the Project Management Body of Knowledge (PMBOK® Guide). 6th edn. Project Management Institute, Inc., USA (2017) .
8. Cohn, M.: User Stories Applied: for Agile Software Development. 1st edn. Addison Wesley, USA (2004).
9. Lucassen, G., Dalpiaz, F., E.M. van der Werf, J.M., Brinkkemper, S.: The Use and Effectiveness of User Stories in Practice. In: Daneva, M., Pastor, O. (eds.), Requirements Engineering: Foundation for Software Quality 2016, LNCS, vol. 9619, pp. 205-222. Springer International Publishing, doi:10.1007/978-3-319-30282-9_14, ISBN 978-3-319-30281-2 (2016).
10. Lucassen, G., Dalpiaz, F., E.M. van der Werf, J.M., Brinkkemper, S.: Forging High-Quality User Stories: Towards a Discipline for Agile Requirements. In: IEEE International Conference on Requirements Engineering (RE), pp. 126-135. IEEE, Canada, (2015) doi: 10.1109/RE.2015.7320415
11. Agile Alliance: Agile Glossary, <https://www.agilealliance.org/glossary/epic>, last accessed 2021/05/10.
12. Kanbanize: How to Structure Work with Themes, Initiatives, Tasks, and Epics in Agile?, <https://kanbanize.com/agile/project-management/tasks-epics-initiatives-themes>, last accessed 2021/05/10.
13. Rubin, K.: Essential Scrum: A Practical Guide to the Most Popular Agile Process. 1st edn., USA (2012).
14. Atlassian: Epics, stories, themes, and initiatives, <https://www.atlassian.com/agile/project-management/epics-stories-themes>, last accessed 2021/05/17.
15. McDonald, K.: Beyond Requirements: Analysis with an Agile Mindset (Agile Software Development Series). 1st edn. Addison-Wesley Professional, USA (2015).
16. Scaled Agile Inc.: <https://www.scaledagileframework.com>, last accessed 2021/08/02
17. Vodde, B., Larman, C.: Large-Scale Scrum: More with LeSS. Addison-Wesley Professional, USA (2016)
18. Noreika, K.: Business Capabilities Utilization Enhancement Using Archimate for EAS Projects Delivery in an Agile Environment. In: Matulevičius, R., Robal, T., Haav, H-M., Maigre, R., Petlenkov, E. (eds.) Joint Proceedings of Baltic DB&IS 2020 Conference Forum and Doctoral Consortium co-located with the 14th International Baltic Conference on Databases and Information Systems (BalticDB&IS 2020) CEUR Workshop proceedings, vol. 2620, pp. 49-56, Estonia (2020).
19. Gudas, S.: Causal Modelling in Enterprise Architecture Frameworks. Informatica, 1-35, doi: 10.15388/21-INFOR446 (2021).
20. Zack, M.H.: If Managing Knowledge is the Solution, then What's the Problem? In: Yogesh Malhotra (eds.) Knowledge Management and Business Model Innovation, Idea Group Publishing, doi: 10.4018/978-1-878289-98-8.ch002 (2001).

21. Gudas, S.: Foundations of the Information Systems' Engineering Theory. Vilnius University, Vilnius (2012).
22. British Ministry of Defence: MOD Architecture Framework (MODAF), https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/36757/20100602MODAFDownload12004.pdf, last accessed 2021/07/15.