ISSN 2347 - 3983
Volume 8. No. 9, September 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter23892020.pdf
https://doi.org/10.30534/ijeter/2020/23892020

MDA Transformation Process from predictive project management methodologies to agile project management methodologies

Ibrahim Hamzane, Abdessamad Belangour, Mouad Banane, and Allae Erraissi
Laboratory of Information Technology and Modeling, Hassan II University, Faculty of sciences Ben M'Sik, Casablanca, Morocco
Emails: hamzane.ibrahim@gmail.com, belangour@gmail.com, mouadbanane@gmail.com, erraissi.allae@gmail.com

ABSTRACT

Project management models are constantly evolving over time, among the most widely answered models we find Agile models or else called "Adaptive" and "Predictive" models. A study by the Standish group confirms that Agile projects achieve the desired result three times more than projects that are carried out by conventional (predictive) Methodologies. Thus, many organizations tend to apply the new project management model or migrate from the standard model to the agile one. In this article, building on previous research, we'll build on the principles of MDA to define a metamodel of Agile and Predictive Methodologies, and then we'll effect a transformation that will help organizations transform project management.

Key words: Project Management, Agile Project Management, Predictive Project Management, MDA, Metamodeling, Transformation, ATL, Digital transformation.

1. INTRODUCTION

Project management plays a key role in achieving goals by following plans and expectations. Agile or "Adaptive" and Waterfall or "Predictive" are two of the most common project management frameworks that organizations use to execute projects. While both have similar characteristics, one is drastically different from the other, each having their own strengths and weaknesses depending on the project. According to the report of the Standish group [1] only 9% of IT projects managed in agile from 2002 to 2012 suffered a failure against 29% of failure on projects managed in predictive, and 42% of Agile projects were successful against only 14% for projects managed in Predictive. Aside from statistics, we also find the advancement of technologies that promote work with Agile Methodologies such as Cloud, Big Data, IAAS, PAAS, etc. Thus, a migration to agile has become necessary to stay the course.

The first publications on "agile" Methodologies appeared in 1991 with the development of the RAD method by James Martin. Other Methodologies then emerged such as the DSDM method, the Cristal family of methodologies, development driven by TDD ", etc. While all "agile" Methodologies are based on common principles and values, they stand out, however, in relation to the management practices and instruments that they mobilize. Thus, in the literature, we find many studies on Agile methodologies (Adaptive methodologies) given the current trend [2], [3] and [4]. Unlike traditional project management methodologies which appeared long before which are based on prediction (risk, plan, dates, constraints, etc.).

The objective of this chapter is to expose the two project management metamodels (Agile and predictive) then carry out a transformation of predictive methodologies towards Agile methodologies which will help organizations in their digital transformations [31] and finally validate our model by calculating the execution time of the transformation according to a case study.

2. BACKGROUND OF PROJECT MANAGEMENT

2.1 Predictive methodologies

In the context of project management, the classic or traditional approach is the most common. Also called the "waterfall method" or "V cycle", it is based on the treatment of the different phases of a project in a sequential manner: specification and analysis of needs, design or general planning, development and production, test and corrections, and finally, delivery. All these steps depending on each other, a task can only be started when the previous one has been validated.

The predictive nature of the traditional management process has the advantage of providing comprehensive project planning from the start: clearly defined objectives, fixed deadlines and deadlines, and a precise budget. This method also promotes knowledge transfer. This will motivate the team members. Indeed, it should be noted that this approach calls for a team with specialized resources.
This involves the production of a significant amount of documents and other communication media that will be accessible to all stakeholders.

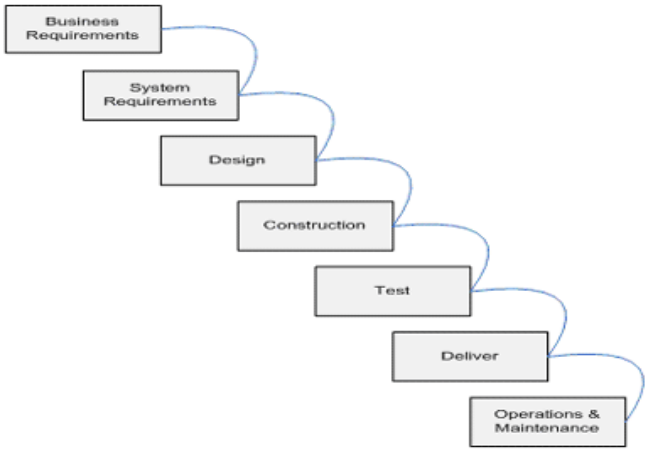

Figure 1: Predictive project management example methodology.

2.2 Agile methodologies

The term "agile" defines a project management approach that takes the opposite of traditional predictive and sequential approaches of the V-cycle or waterfall type. The very notion of "project management" is called into question in favor of "product management". In order to reason more "product" than "project". After all, the objective of a project is to give birth to a product.

A so-called "traditional" approach generally expects from the client a detailed and validated expression of the need for entry into the project, leaving little room for change. The realization lasts the time it takes, and the appointment is made with the customer for the recipe. This tunnel effect can be very harmful and conflicting, there is often a phase difference between the initial requirement and the application carried out.

This refers to the validated specifications and the contract. Some projects end in pain (especially in the context of a standard package contract) at the risk of compromising the customer relationship. In addition, it is not uncommon for certain requested functionalities to turn out to be useless in the end, while others, discovered along the way, could have given more value to the product.

A 1994 survey by the “Standish Group” (admittedly controversial, like all surveys that deal with a sensitive subject) made the following observation : “31% of IT projects are stopped in the process, 52% only end in at the cost of significantly exceeding time and budget while offering less functionality than was requested; only 16% of projects can be considered successful. ".

This same survey, repeated in 2008, shows a success rate of 35%, which is rather positive but remains very low. The problem remains unresolved. Among the chess reasons, come first:

- Lack of end-user involvement: 12.8 %.
- Changes to specifications during the project: 11,8 %.

On the other hands, the Agile approach proposes to considerably or even completely reduce this tunnel effect by giving more visibility, by involving the client from the beginning to the end of the project and by adopting an iterative and incremental process. She considers that the need cannot be fixed and, on the contrary, proposes to adapt to changes in the latter. But not without a minimum of rules:

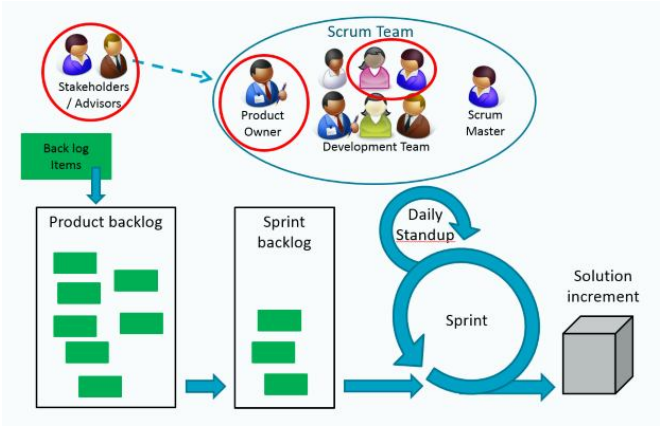

Figure 2: Agile project management with Scrum.

3. RELATED WORK

In the organization, all processes can be subject to disruption. We are talking in this manuscript of a hazard or a need for change to characterize all types of disturbances. A hazard is defined in the Larousse dictionary as “an unforeseeable and most unfavorable turn taken by events and linked to an activity, action”. These are events that affect all the perspectives of the process approach presented in the previous chapter. Each of these perspectives describes an important aspect of the process models and can be an object of change throughout its life cycle [5]. In the work of [6], according to a study he conducted on the work of [7] and [8], managing uncertainty and responding to change has become a primary concern of organizations. While the need to manage change is not new to a business, it is more so is the almost constant increase in the frequency with which these changes need to be made. This explains the growing interest in research that focuses on agility. Indeed, in order to respond to changes, agility with its two dimensions (i.e. organizational and technical) makes it possible to act on all perspectives of the process: (i) technical agility acts on the informational perspective with the resolution syntactic and semantic differences and guarantee company interoperability, (ii) organizational agility makes it possible to act at the level of

the organizational perspective with a mobilization and dynamic restructuring of resources and (iii) the two dimensions (i.e. organizational and technical) act on the functional, operational and behavioral perspective with a dynamic reconfiguration of the process.

Before the publication of the agile manifesto in 2001, some agile Methodologies were already applied in software design. We will present the most frequently cited [9,10]. These Methodologies are said to be differentiating and therefore have different characteristics, but all meet the main objective of agility: customer satisfaction. The practices constituting these Methodologies can in theory be used anywhere, individually or in conjunction with others [11] and therefore offer a malleable character.

The Rapid Application Development or RAD method is the first agile method of project management that completely breaks with the previously classic cascade development cycle. This new development cycle is termed iterative, incremental and adaptive. This method was published by James Martin [12]. The RAD method recommends individual and then collective evaluation phases of the product code before quickly integrating it into the project. We also find pairs of programming phases for the most strategic parts, allowing both to increase the ability to detect errors, but also to increase creativity.

PMoreover, the RAD method introduces the role of facilitator. This person, external to the development team, provides a perspective on the project. Depending on the progress of the project, whether technically, economically or strategically, this client representative can make decisions in order to reorient the project objectively. The facilitator is also responsible for proposing solutions to the problems reported by the design team.

The RAD method in version 2 offers greater flexibility as to the size of the design groups in order to optimize human resources according to the nature of the project. It also gives greater consideration to communication between design actors by setting up facilitation and reporting groups. These groups are responsible for evaluation and validation. Finally, version 2 offers the possibility of offering reduced functionality deliverables in order to further shorten the duration of development cycles.

Extreme programming or XP is an agile method of software production-oriented project management, invented by Kent Beck, Ward Cunningham and Ron Jeffries, used since 1996, and documented in 1999 [13]. It is based on five values:

- Communication between all stakeholders.
- Simplicity.
- Numerous retrospectives.
- Courage.
- Respect.

The practices of this method are the application of very fast cycles, with little functionality to do. As soon as the tasks are completed, they are implemented in the software, thus creating continuous integration. We also find a very strong involvement of the client or his representative who actively participates in decision-making. Sometimes the design team wants to have the customer or their representative directly on site [14]. The XP method also advocates the use of Planning Poker, a card game to aid communication and planning [15, 16]. This card game makes it possible to estimate the complexity and duration of the design tasks to be carried out while putting players in confrontation with the estimates of others. Regarding the design team, it should be small, i.e. made up of less than 20 members. They never have to work overtime to be efficient all the time and can practice pair programming. As in the RAD method, this practice consists of positioning two designers next to each other and having them write the code together. In the XP method, it is advisable to practice programming in pairs as often as possible.

The Scrum method is an agile method of project management associating a work process made up of short iterations called “sprints” at the end of which deliverables are proposed as well as practices applied throughout these sprints. This method was initially theorized by Takeuchi and Nonaka in [17] under the name of "rugby approach" by analogy with a rugby team which would advance united to advance the ball. The rugby approach method was initially intended for the industrial sector and proposed an iterative and multidisciplinary production [17]. It was in 2001 that Schwaber and Beedle described the method called Scrum still used today [18].

The method consists in writing usage scenarios, called user stories, in which the designers project themselves as potential users The Scrum method is an agile method of project management associating a work process made up of short iterations called “sprints” at the end of which deliverables are proposed as well as practices applied throughout these sprints. This method was initially theorized by Takeuchi [17] under the name of "rugby approach" by analogy with a rugby team which would advance united to advance the ball. The method consists of writing usage scenarios, called user stories, in which the designers project themselves as potential users. During this sprint, the team will work on carrying out tasks to meet the uses of user stories. So it's about transforming the client's needs into conceptual intentions and then into design tasks. While a sprint typically lasts one to four weeks, each day the design team should have a meeting, called a daily scrum. During this meeting, the designers quickly discuss their progress and possible problems. When all user stories are complete or the sprint is nearing its end, designers hold a meeting with customers, or their representative, called a sprint review, in which they present deliverables showing the completed user stories. It can be a demonstration, a visual, a table or any visualizable element. The product backlog is updated, while yet another sprint is ready to start over. The designers do another meeting, called the Sprint Retrospective, in which they list the positives and negatives of the sprint just completed, with the goal of improving the sprint ahead. Each sprint, therefore each iteration makes it possible to increase the functionality of the project, and the numerous meetings make it possible to offer a strong adaptability, while improving group cohesion, mutual awareness and knowledge sharing [19, 29, 21].

4. METAMODEL OF PREDICTIVE METHODOLOGIES

4.1 Predictive methodologies

Predictive Methodologies focus on analyzing and planning for the future in detail and take into account known risks. In extremes, a predictive team can report exactly what features and tasks are planned throughout the duration of the development process. Predictive Methodologies rely on effective analysis of the initial phase and if this goes very badly, the project may have difficulty in changing direction. Predictive teams will often set up a change control board to ensure that only the most important changes are taken into account [30].

4.2 Meta-model of predictive methodologies

Figure 3 presents the proposed meta-model for predictive methodologies.

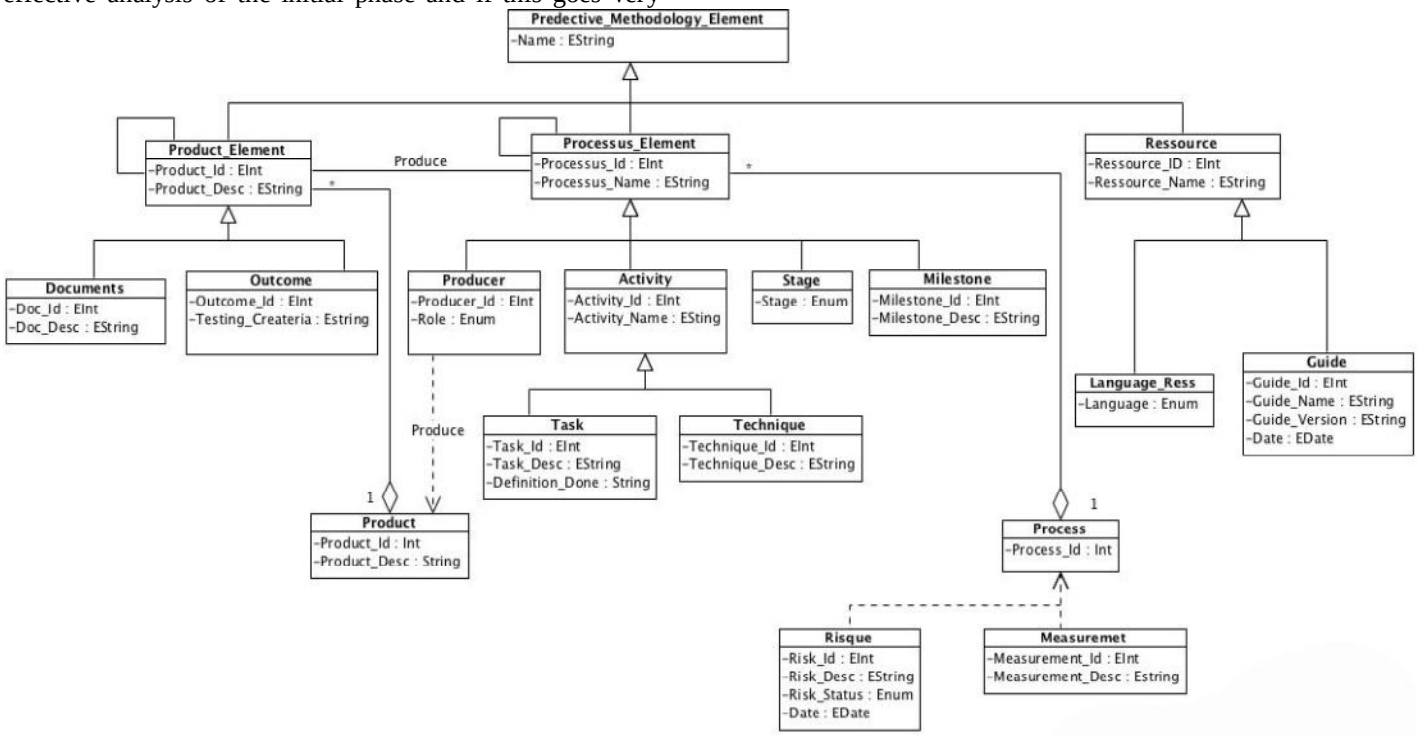

Figure 3: Proposed meta-model of predictive methodologies.

4.3 Meta-model description

Below the description of the proposed meta-model for predictive methodologies:

- **Process Element**: Elements of methodology transforming inputs into outputs. The relationship between one or more elementary processes generates the process., (as example: Management practices, process performance steps)
- **Activity**: A basic abstract process element that models a functionally consistent operation that can be performed by one or more producers, (as example: Practice (WBS, Gant ...), Activity, Task, Technique)
- **Task**: The smallest manageable unit of work., (as example: Write unit tests, model use cases, Write code)
- **Technique**: Specific means to achieve a goal., (as example: Interview, workshop reflection)
- **Measurement**: Specific means to carry out measurements on the processes, (as example: Baseline comparison, Cost control …)
- **Risk**: An essential element when it comes to predictive methodologies, (as example: Risk level, risk status …).
- **Producer**: Someone or something that performs a unit of work either directly or indirectly (i.e. creates, evaluates, iterates, or maintains)., (as example: Project Manager, tester, Sponsor, tool).
- **Stage**: Represents managed time intervals or time points, within a project, i.e. the temporal aspect of the process., (as example: Sequence, phase, part of project).
- **Milestone**: These are the milestones of the projects, (as example: Milestone, decision point, etc).
- **Product Element**: Any artifacts developed and produced during the project. This item is equivalent to ""Work", (as example: Version, UML models, technique).
- **Activity**: A deliverable is a product element that provides a description and definition of the packaging of other product elements, and can be delivered to an external party., (as example: Release, technical report).
- **Documents**: An identifiable resource provided by the development team that supports the overall software development process., (as example: Software Architecture Document, specifications, etc.).
- **Outcome**: The result is a product element that provides a description and definition of the non-tangible product elements. A key difference between results and artefacts is that results are not candidates for harvesting

as reusable assets., (as example: Practice, discipline, field, plan).

- **Resource**: Methodological elements used in the realization of the project., (as example: Standard reports, expert documentation).
- **Language**: Vocabulary (i.e. a set of terms) combined with a set of grammatical rules used to produce one or more items of products., (as example: Modeling language, Implementation language, Database Language).
- **Guideline**: Rules and guidelines on the appropriate use of a given methodological element., (as example: Standard, methodological guidelines).

5. METAMODEL OF AGILE METHODOLOGIES

5.1 Agile methodologies

Development methods exist on a continuum ranging from adaptive to predictive. Agile methods are on the adaptive side of this continuum. One of the keys to adaptive development methods is a "rolling wave" approach to schedule planning, which identifies milestones but leaves flexibility in the path to reach them, and also allows the milestones themselves to change. Adaptive methods aim to adapt quickly to changing realities. When the needs of a project change, an adaptive team also changes. An adaptive team will have a hard time describing exactly what will happen in the future. The later a date, the vaguer an adaptive method will be about what will happen on that date. An adaptive team cannot report exactly what tasks it will perform next week, only the features it plans for next month. When asked about a release six months from now, an adaptive team might be able to report only the release's mission statement, or a statement of expected value versus cost [30].

5.2 Meta-model of agile methodologies

Below the proposed meta-model for agile methodologies:

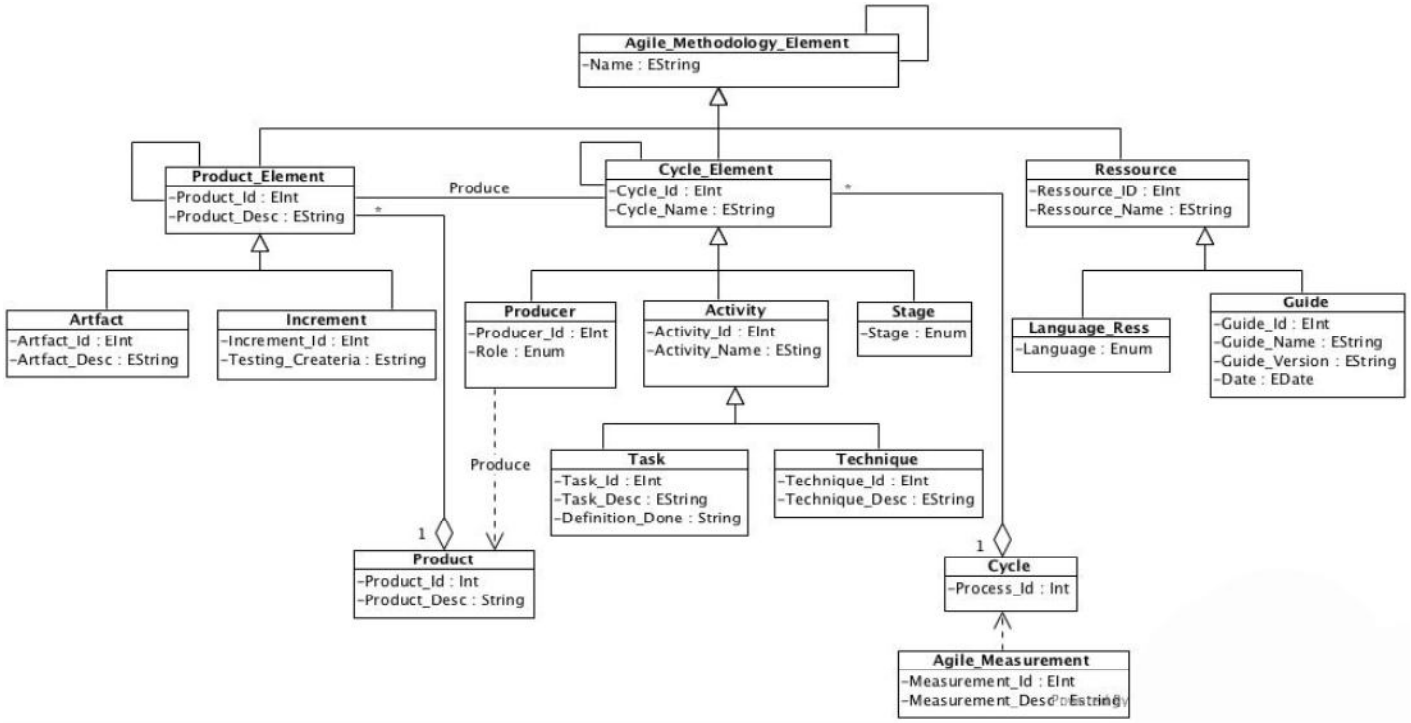

Figure 4: Proposed meta-model of agile methodologies.

5.3 Meta-model description

Below the description of the proposed meta-model for agile methodologies:

• Process Element: Elements of methodology transforming inputs into outputs. The relationship between one or more elementary processes generates the process., (as example: Management practices, process performance steps).

• Activity: A basic abstract process element that models a functionally consistent operation that can be performed by one or more producers, (as example: Practice (pair programming, Refactoring, etc.), Activity, Task, Technique).

• Task: The smallest manageable unit of work., (as example: Write unit tests, model use cases, Write code).

• Technique: Specific means to achieve a goal., (as example: Interview, workshop reflection).

• Measurement: Specific means to carry out measurements on the processes, (as example: Burn Down Chart, Velocity Chart, etc.).

• Producer: Someone or something that performs a unit of work either directly or indirectly (i.e. creates, evaluates, iterates, or maintains)., (as example: Scrum master, team, tester, organization, tool).

• Stage: Represents managed time intervals or time points, within a project, i.e. the temporal aspect of the process., (as example: Iteration, sprint, phase, cycle).

6. TRANSFORMATION

Up till now, we have defined the meta-models of Predictive methodology, and Agile methodology. In this section, we present the transformation rules used to pass from generic meta-models of Predictive methodology to Agile methodology.

To apply all the transformations, we chose the ATL transformation language [22, 23]. We now present extracts from the ATL code that we used to transform the meta-models proposed for Predictive methodology to the meta-model proposed for Agile methodology [32,33]. These defined meta-models present the PIM (Platform Independent Model) level according to the architecture led by the 'MDA' models [24, 25,35,36].

```
rule
Predictive_Methodology_Element2Agile_Methodology
_Element{
   from
     s: Predictive_Methodology_Element
   to
   t: Agile_Methodology_Element(
name<- s.name
      )
}
rule Processus_Element2Cycle_Element {
   from
     s: Processus_Element
   to
   t: Cycle_Element(
Cycle_id<- s.Processus_id,
Cycle_name<- s.Processus_name
      )
}
rule Product_Element2Product_Element {
   from
     s: Product_Element
   to
   t: Product_Element(
Product_id<- s.Product_id,
Product_Desc <- s.Product_Desc
      )
}
rule Ressource2Ressource {
   from
     s: Ressource
   to
   t: Ressource (
Ressource_id<- s.Ressource_id,
Ressource_Name <- s.Ressource_Name
      )
}
rule Language_Ress2Language_Ress {
   from
     s: Language_Ress
   to
   t: Language_Ress (
Language<- s.Language
      )
}
rule Guide2Guide {
   from
     s: Guide
   to
   t: Guide (
Guide_id<- s.Guide_id,
Guide_Name <- s.Guide_Name,
Guide_Version <- s.Guide_Version,
Date <- s.Date
      )
}
rule Producer2Producer {
   from
     s: Producer
   to
   t: Producer (
Producer_Id<- s. Producer_id,
Role <- s.Role
      )
}
rule Activity2Activity {
   from
     s: Activity
   to
   t: Activity (
Activity_Id <- s.Activity_Id,
Activity_Name <- s.Activity_Name
      )
}
rule Stage2Stage {
   from
     s: Stage
   to
   t: Stage (
Stage <- s.Stage
      )
}
rule Task2Task {
   from
     s: Task
   to
   t: Task (
Task_id<- s.Task_id,
Task_Desc <- s.Task_Desc,
Definition_Done <- s. Definition_Done
      )
}
rule Technique2Technique {
   from
     s: Technique
```

```
   to
   t: Technique (
Technique_id<- s.Technique_id,
Technique_Desc <- s.Technique_Desc,
      )
}
rule Documents2Artfact {
   from
     s: Documents
   to
   t: Artfact (
Artfact_id<- s.Doc_id,
Artfact_Desc <- s.Doc_Desc
      )
}
rule Outcome2Increment {
   from
     s: Outcome
   to
   t: Increment (
Increment_id<- s.Outcome_id,
Testing_Createria <- s.Testing_Createria
      )
}
rule Product2Product {
   from
     s: Product
   to
   t: Product (
Product_id<- s.Product_id,
Product_Desc <- s.Product_Desc
      )
}
rule Process2Cycle {
   from
     s: Process
   to
   t: Cycle (
Cycle_id<- s.Process_id,
      )
}
rule Mesurement2Agile_Mesurement {
   from
     s: Mesurement
   to
   t: Agile_Mesurement (
Mesurement_id<- s.Mesurement_id,
Mesurement_Desc <- s.Mesurement_Desc
      )
}
```

Figure 5: ATL transformation code.

7. TRANSFORMATION TESTING

In this section, we will present an experiment on our transformation, this transformation which takes as input the Predective metodology metamodel towards an output which is a model of Agile methodology metamodel, the following figure represents the result of execution time of transformation d 'an IT project managed by predictive methodology, which will be transformed into management by agile methodology.

This transformation was performed by the transformation language ATL [26, 27,34]. To achieve this transformation, we used three instances of projects, a small project called Project_1 (one month of working days), a medium project called Project_2 (two hundred working days), and a large project which is Project_3 (3 thousand of working days). The results of this transformation are presented in the following table:

Table 1: Transformation time for the three projects.

	Project 1	**Project 2**	**Project 3**
Duration (w.d.)	30	200	3000
Transformation time (s)	323	358	391
Duration (w.d.)	30	200	3000

The following figure illustrates the result of transformation time in (s) using the three projects: Project_1, Project_2, Project_3:

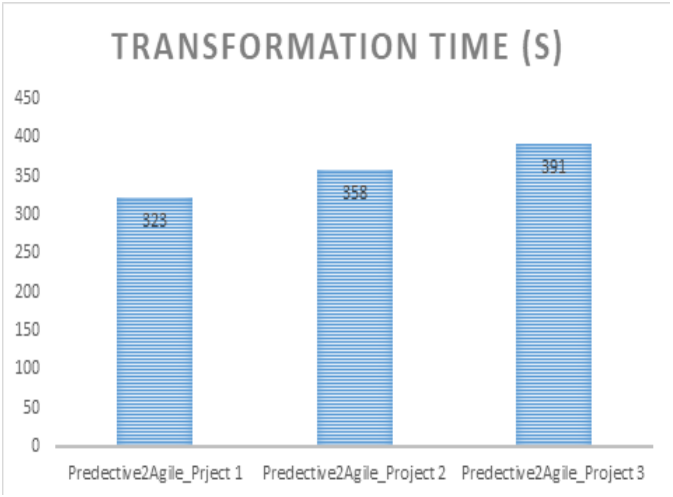

Figure 6: Transformation time for the three projects.

8. DISCUSSION

The most popular agile methods are RAD [37], XP [38] and Scrum which are based on several communication and coordination support practices. The RAD method advocates the establishment of the role of facilitator whose mission is to make decisions as a "customer representative" as well as to find solutions to the problems of the design team. It also introduces the practice of pair coding. XP builds on collective intelligence by reusing the practice of a facilitator, while promoting the emergence of group cohesion. She also sets up planning poker sessions to estimate the duration of the tasks to be carried out. Finally, Scrum establishes a regular development cycle in which daily meetings take place. These

meetings allow the actors of the collaboration to discuss their progress and their needs. Scrum can also rely on planning poker sessions [28, 29].

We find that these three agile methods have common characteristics and rely on practices easily applicable in one or the other method, making them malleable. The practices used in most agile methods are compatible with each other. [11] proposes the joint use of Scrum and XP, mixing practices such as planning poker, pair code, or the facilitator role.

9. CONCLUSION

We proposed two project management meta-models (Agile and Predictive) which encompass the concepts of project management, then we created a transformation that allows to deduce the Agile model from a Predictive model and we applied the model of transformation on a concrete example of project management.

10. FUTURE SCOPE

The overall objective of this contribution is to help organizations in their digital transformations by using new technologies in the most optimal way, without forgetting that the key element of a transformation will remain the human component which requires a management program. organizational change, we will be working in other areas of IT governance.

REFERENCES

1. Standish Group 2011. The Crisis in Software : The Wrong Process Produces the Wrong Results, The Standish Group Report (2011). www.controlchaos.com/storage/S3D%20First%20Chapter.pdf.C HAOS report
2. E. Mnkandla and B. Dwolatzky, “Agile methodologies selection toolbox,” in Software Engineering Advances, 2007. ICSEA 2007. International Conference on. IEEE, 2007, pp. 72–72.
3. B. Fitzgerald, G. Hartnett, and K. Conboy, “Customising agile Methodologies to software practices at intel shannon,” European Journal of Information Systems, vol. 15, no. 2, pp. 200–213, 2006.
4. “Nokia networks jumped into agile wagon,” ITEA-AGILE, 2006.
5. G. Regev, P. Soffer, and R. Schmidt. Taxonomy of flexibility in business processes. In Business Process Modelling, Development and Support (BPMDS), 2006.
6. W. Zhang, M. S. Branicky, and S. M. Phillips. Stability of networked control systems. In IEEE Control Systems, 2001
7. J. Thomphson and M. P. Heimdalh. Extending the product family approach to support n- dimensional and hierarchical product lines.
8. P. Drucker. The age of discontinuity : guidelines to our changing society. In Harper and Row, New York.
9. LINDVALL, Mikael, BASILI, Vic, BOEHM, Barry, COSTA, Patricia, DANGLE, Kathleen, SHULL, Forrest, TESORIERO, Roseanne, WILLIAMS, Laurie et ZELKOWITZ, Marvin, 2002. Empirical Findings in Agile Methodologies. In : WELLS, Don et WILLIAMS, Laurie (éd.). In : Extreme Programming and Agile Methodologies — XP/Agile Universe 2002. S.l. : Springer Berlin Heidelberg. 2002. p. 197□207
10. ZELKOWITZ, Marvin, 2004. Advances in Computers: Advances in Software Engineering. S.l. : Elsevier. ISBN 978-0-08-047190-7.
11. KNIBERG, Henrik, 2015. Scrum and Xp from the Trenches 2nd Edition. 2. S.l. : Lulu.com. ISBN 978-1-329-22427-8.
12. MARTIN, James, 1991. Rapid Application Development. New York: Toronto: New York: Macmillan Coll Div. ISBN 978-0-02-376775-3.
13. BECK, Kent, 2004. Extreme Programming Explained: Embrace Change. 2. Boston, MA: Addison Wesley. ISBN 978-0-321-27865-4.
14. KOSKELA, Juha et ABRAHAMSSON, Pekka, 2004. On-Site Customer in an XP Project: Empirical Results from a Case Study. In : DINGSØYR, Torgeir (éd.). In : Software Process Improvement. S.l. : Springer Berlin Heidelberg. 2004. p. 1□11.
15. GRENNING, James, 2002. Planning Poker or How to avoir analysis paralysis while release planning. 2002. S.l. : s.n. Disponible à l’adresse : https://wingman-sw.com/articles/planningpoker.
16. HAUGEN, N. C., 2006. An empirical study of using planning poker for user story estimation. In : AGILE 2006 (AGILE’06), AGILE 2006 (AGILE’06). S.l. : s.n. juillet 2006. p. 9 pp. □ 34. DOI 10.1109/AGILE.2006.16.
17. TAKEUCHI, Hirotaka et NONAKA, Ikujiro, 1986. The New New Product Development Game. In : Harvard Business Review. 1 janvier 1986. n° January 1986. Disponible à l’adresse: https://hbr.org/1986/01/thenew-new-product-development-game.
18. SCHWABER, Ken et BEEDLE, Mike, 2001. Agile Software Development with Scrum. 1 edition. Upper Saddle River, NJ : Pearson. ISBN 978-0-13-067634-4.
19. PAASIVAARA, M., DURASIEWICZ, S. et LASSENIUS, C., 2009. Using Scrum in Distributed Agile Development: A Multiple Case Study. In : 2009 Fourth IEEE International Conference on Global Software Engineering, 2009 Fourth IEEE International Conference on Global Software Engineering. S.l. : s.n. juillet 2009. p. 195□204. DOI 10.1109/ICGSE.2009.27.
20. SUTHERLAND, Jeff et SCHWABER, Ken, 2011. The scrum papers: nut, bolts, and origins of an Agile framework. In : Scrum inc. S.l. : s.n. 2011.
21. WEINREICH, Ronny, NEUMANN, Norbert, RIEDEL, Ralph et MÜLLER, Egon, 2015. Scrum as Method for Agile Project Management Outside of the Product Development Area. S.l. : s.n.
22. BANANE, Mouad et BELANGOUR, Abdessamad. Towards a New Scalable Big Data System Semantic Web Applied on Mobile Learning. International Journal of Interactive Mobile Technologies (iJIM), 2020, vol. 14, no 01, p. 126-140.
23. BANANE, Mouad et BELANGOUR: A Big Data Solution To Process Semantic Web Data Using The Model Driven Engineering Approach, International

Journal of Scientific; Technology Research. vol. 9, no 02, 2020.
24. Banane, Mouad, and Abdessamad Belangour.;A new system for massive RDF data management using Big Data query languages Pig, Hive, and Spark; International Journal of Computing and Digital Systems 9.2 (2020): 259-270.
25. Erraissi, Allae, and Abdessamad Belangour. Meta-Modeling of Big Data Management Layer. International Journal of Emerging Trends in Engineering Research 7, 7, 36-43, 2019. https://doi.org/10.30534/ijeter/2019/01772019.
26. Banane, Mouad, Allae Erraissi, and Abdessamad Belangour. "SPARQL2Hive: An approach to processing SPARQL queries on Hive based on meta-models." 2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO). IEEE, 2019.
27. Erraissi, Allae. "Using model Driven Engineering to transform Big Data query languages to MapReduce jobs." International Journal of Computing and Digital Systems 9 (2020): 1-9.
28. HAMZANE, Ibrahim, and Abdessamad BELANGOUR. "Implementation of a decision system for a suitable IT governance framework." International Journal of Computer Science and Information Security (IJCSIS) 17.5 (2019).
29. Hamzane, Ibrahim, and Abdessamad Belangour. "A Multi-criteria Analysis and Advanced Comparative Study Between IT Governance References." International Conference on Advanced Intelligent Systems for Sustainable Development. Springer, Cham, 2019.
30. HAMZANE Ibrahim and Belangour Abdessamad, "Project Management Metamodel Construction Regarding IT Departments" International Journal of Advanced Computer Science and Applications (IJACSA), 10(10), 2019. http://dx.doi.org/10.14569/IJACSA.2019.0101029
31. HAMZANE Ibrahim and Belangour Abdessamad, "A Built-in Criteria Analysis for Best IT Governance Framework" International Journal of Advanced Computer Science and Applications (IJACSA), 10(10), 2019. http://dx.doi.org/10.14569/IJACSA.2019.0101026
32. Erraissi, Allae. "Using model Driven Engineering to transform Big Data query languages to MapReduce jobs." International Journal of Computing and Digital Systems 9 (2020): 1-9.
33. Erraissi, Allae, and Abdessamad Belangour. "An Approach Based On Model Driven Engineering For Big Data Visualization In Different Visual Modes." International Journal of Scientific & Technology Research.
34. Erraissi Allae, and Abdessamad Belangour. "Meta-Modeling of Big Data visualization layer using On-Line Analytical Processing (OLAP)." International Journal of Advanced Trends in Computer Science and Engineering 8, No. 4 (2019).
35. Erraissi Allae, and Abdessamad Belangour. "A Big Data Security Layer Meta-Model Proposal." Advances in Science, Technology and Engineering Systems Journal 4, No. 5 (2019). https://doi.org/10.25046/aj040553.
36. Erraissi Allae, et Abdessamad Belangour. « A Big Data Security Layer Meta-Model Proposition ». Advances in Science, Technology and Engineering Systems Journal 4, n° 5 (2019). https://doi.org/10.25046/aj040553.
37. Khan, F. Q., Rasheed, S., Alsheshtawi, M., Ahmed, T. M., & Jan, S. (2020). A Comparative Analysis of RAD and Agile Technique for Management of Computing Graduation Projects. CMC-COMPUTERS MATERIALS & CONTINUA, 64(2), 777-796.
38. Wells, D., & Williams, L. (Eds.). (2003). Extreme Programming and Agile Methods-XP/Agile Universe 2002: Second XP Universe and First Agile Universe Conference Chicago, IL, USA, August 4-7, 2002. Proceedings (Vol. 2418). Springer.