

# Tests Management in CALIPSOneo: A MDE Solution

Alberto Salido, Julián Alberto García García, José Ponce, Javier Jesús Gutierrez

Software Engineering and Early Testing Group, Universidad de Sevilla, Sevilla, Spain  
Email: [alberto.salido@iwt2.org](mailto:alberto.salido@iwt2.org), [julian.garcia@iwt2.org](mailto:julian.garcia@iwt2.org), [iosepg@us.es](mailto:iosepg@us.es), [javierj@us.es](mailto:javierj@us.es)

Received 3 March 2014; revised 2 April 2014; accepted 10 April 2014

Copyright © 2014 by authors and Scientific Research Publishing Inc.  
This work is licensed under the Creative Commons Attribution International License (CC BY).  
<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

Testing in Software Engineering is one of the most important phases although, unfortunately, it cannot be always successfully fulfilled due to time constraints. In most cases, the development phase takes more time than it was estimated, entailing negative effects on the testing phase. The delay increases even more in Research and Development (R + D) projects, where the real time to execute tasks is more difficult to control. Model Driven Engineering (MDE) offers a solution to avoid testing costs without affecting the execution quality of the applied test. This paper presents a practical overview of a Model Driven Testing (MDT)-based methodology and its impact on CALIPSOneo project, which was carried out in liaison with Airbus Defense and Space and, particularly, with the Product Lifecycle Management (PLM) department.

## Keywords

Model-Driven Testing, Model-Driven Engineering, Product Lifecycle Management

---

## 1. Introduction

Test phase is a key point in software development. A good definition of tests will ensure the created software quality. Unfortunately, Software Engineering does not include for this proposal any specific methodology that ensures the expected results of this phase. Taking into account the authors' experience, test phase is usually avoided when the project undergoes any delay, what reduces the amount of tests executed as well as their quality.

In the particular case of Functional Tests, people who do not participate in the requirement elicitation, for instance programmers, frequently describe them. Consequently, they sometimes approve some functional tests that do not satisfy the initial needs previously defined.

Model-Driven Testing [1] is one of the initiatives applied when looking for a solution to these aspects. This

paradigm can help producing the development phase systematically and independently, by decreasing the number of generated tests.

Model-based testing is application of model-based design for designing and optionally also executing artifacts to perform software testing or system testing. Models can be used to represent the desired behavior of a System Under Test (SUT), or to represent testing strategies and a test environment.

A model describing a SUT is usually an abstract, partial presentation of the SUT's desired behavior. Test cases derived from such a model are functional tests on the same level of abstraction as the model. These test cases are collectively known as an abstract test suite. An abstract test suite cannot be directly executed against an SUT because the suite is on the wrong level of abstraction. An executable test suite needs to be derived from a corresponding abstract test suite. The executable test suite can communicate directly with the system under test. This is achieved by mapping the abstract test cases to concrete test cases suitable for execution. In some model-based testing environments, models contain enough information to generate executable test suites directly. In others, elements in the abstract test suite must be mapped to specific statements or method calls in the software to create a concrete test suite.

This work presents the aforementioned idea in the context of NDT methodology (*Navigational Development Techniques*) [2], but it is analyzed in a specific use case: CALIPSOneo project. This project has a special behavior and it is defined by means of PLM (*Product Lifecycle Management*) solutions [3].

CALIPSOneo is an ambitious project based on PLM (Product Lifecycle Management). Its main objective is the design of a new PLM methodology to conform to a PLM collaborative design and the required development of the software that satisfies that concept. This methodology allows defining, simulating, optimizing and validating the aeronautical assembly processes in a 3D virtual environment before these processes are implemented in a real shop floor. CALIPSOneo is subdivided into three individual subprojects, in order to effectively manage the work needed to complete it. These subprojects are: MARS (Automated Shop-Floor Documentation Updating System), PROTEUS (Process Structure Generation and Use) and ELARA (Generalization to Assembly Oriented Authoring Augmented Reality).

These subprojects are independent and teams involved in each of them also differ. However, subprojects have to be coordinated and they have to be correctly integrated because they have common actors who demand common functionality.

This paper is structured as follows. After this introduction, Section 2 analyzes the context of Test Driven Engineering. Then, Section 3 offers an overview of CALIPSOneo project and its methodological context and Section 4 studies the real application of our solution to that project. Finally, this paper states the conclusions obtained and presents our future work.

## 2. Model-Driven Testing

Everyday, Model-Driven Testing is becoming more standardized to perform software tests. One option to successfully run this task is to focus it on the standard UTP [4], according to the methodology introduced by Dai [5] which deals with the use of U2TP to transform system models in UML into Testing Models by means of transformation queries Query/View/Transformation-QVT.

As the previous figure represents, Independent Platform Models (IPM) can be transformed into Specific Platform Models (SPM), getting the code from the systems derived from SPM through another transformation step. Once the system has been designed, it can be transformed into System's Test (IPT). The quality team will enrich these tests to provide additional information. Next, the SPM, which contains specific information about the implementation of the system, can be translated into Specific Platform Test (SPT), using the previous information from IPT and the knowledge of the quality team. Finally, these test are generated into code for testing the system, ensuring the quality of the test and covering all the requirements of the project.

U2TP allows concepts to develop requirements and tests models for black-box tests [6]. The profile introduces four groups of logical concepts, covering the following aspects: Test Architecture, Test Behavior, Test Data and Time consumed. All these concepts define together a modeling language to visualize, specify, analyze, build and document a testing system.

There are some studies on methodologies that modify the UML metamodel providing the testing oracle with additional information from the sequence tests diagram: As a sequence diagram, it belongs to a relevant scenario that must be populated. Such diagram is fulfilled with information on the expected result and the initial state [7].

However, the Test Driven Models concept has been applied to many contexts. [8] presents a comparative study which analyzes how some proposals cope with the systematically generation of software tests, functionally tests, in this case. Some of these proposals start with requirements models to become functional tests models.

Despite the proposal we present along this paper, we have found another [9] based on Figure 1. It begins with a functional requirements model, that consists in an instance of a metamodel defined, and QVT transformations outline the functional tests. This proposal constitutes the starting point of our work, which will be then adapted to CALIPSOneo project.

### 3. CALIPSOneo

The next paragraphs offer an overview of CALIPSOneo project, as well as the way the test phase has been approached.

#### 3.1. Project Overview

CALIPSOneo (*Advanced Aeronautical Solutions Using PLM Processes & Tools*) is a project leader by Airbus whose main goal is to identify a working methodology that may allow engineers to define, simulate, optimize and validate aeronautical assembly processes on a 3D environment, before being executed in a real assembly line. This takes place through an integral process of requirements recollection, and the customization and use of the existing PLM software available. The definition of this methodology has been based on the design and assembly process of the Fan Cowl which is the code name of the piece that covers the airplane engine. Figure 2 shows graphically the situation of this piece.

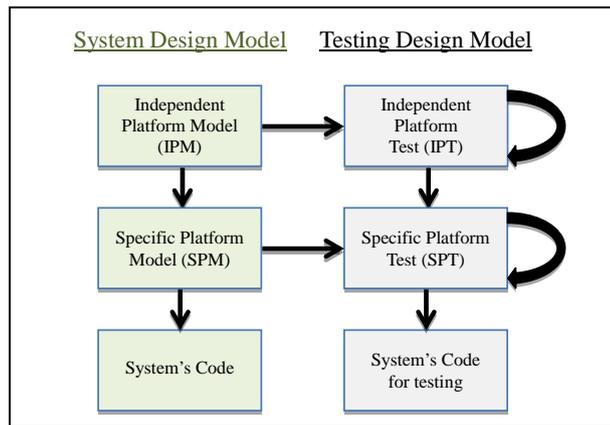


Figure 1. Proposed used in the U2TP profile.

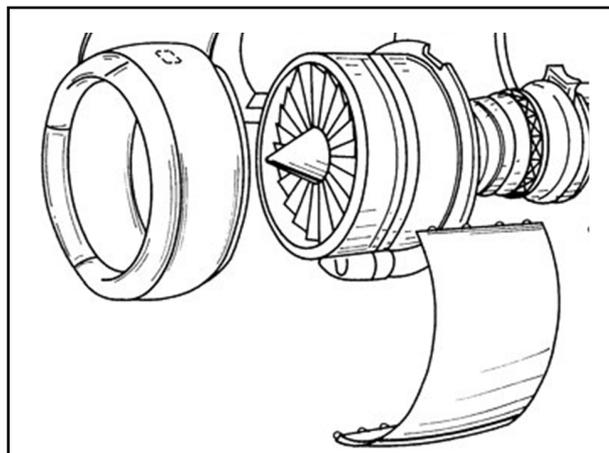


Figure 2. Fan cowl overview.

Some years ago, several analyses were conducted on PLM methodology, concluding that there are many advantages for Airbus Defense and Space [10]. The knowledge obtained from such analyses help CALIPSOneo team compile the different requirements for the project, the management plans for the requirements and requirements' track matrix to meet the needs achieved for the PLM business. Therefore, CALIPSOneo covers both, the design of a new PLM methodology, adjusted to a collaborative PLM solution, and the development of prototypes that satisfy this concept.

CALIPSOneo is divided into three sub-projects with the aim of managing effectively the work to be performed in order to finish the project. This enables the project scope to be managed in a more efficient way, as the different crews are working on the needed tasks to run the project. The three sub-projects are: MARS (*Automated Shop-Floor Documentation Updating System*), PROTEUS (*Process Structure Generation and Use*) and ELARA (*Generalization to Assembly Oriented Authoring Augmented Reality*).

ELARA project's goal is to design a software solution that makes developers work easier by means of Augmented Reality. The expert can visualize the task to perform in 3D on the real image of the work zone, using the tablet PC video camera. This 3D information comes from iDMU (*industrial Digital Mock Up*) [11] and the data obtained from MARS system. The final result will be an industrial prototype that will be used in the fan cowls assembly line for Airbus A320neo.

PROTEUS is responsible for identifying the main structure that interconnects products, processes and resources, also called PPR, and creating iDMU information. Nowadays, people who design products and people performing assembly operations work in different environments. Nevertheless, as the product is a strong requirement for the assembly process, any change it undergoes will influence the team who is working on such process. PROTEUS merges both areas through the Manufacturing Hub; a database that stores information on the products and resources created by the design team, and shows it updated to the processing team.

Finally, MARS system uses the iDMU information PROTEUS generates, elaborating Work Instructions automatically, for workers to perform correctly their job. MARS is strongly connected with PROTEUS, adding the necessary documentation to carry out assembly tasks. MARS, will generate the shop-floor information, even as 2D or 3D document.

Despite each project previously described has well defined objectives and a clear working scope, it is necessary to emphasize that a great integration level is demanded along its development, in order to guarantee that all information among the systems is always correct and coherent.

### 3.2. Methodological Environment

NDT methodology [2] has been used to get the three CALIPSOneo sub-projects' goals. This methodology is covered over the MDE paradigm, but for the testing phase, it uses an extension totally focused on the proposal suggested by [9], previously referred in Section 2.

Currently, NDT controls and defines a set of metamodels for each phase of the software development lifecycle: Viability study, Requirements, Analysis, Design, Implementation, Testing and Maintenance. In addition, it has a set of QVT transformation rules that makes it possible to generate one model from the others systematically. This implies a lower cost for the software development.

However, it is necessary to have a tool that automates the whole process, in order to get all the potential of NDT and make it practical and useful to the enterprise environment. For this aim, NDT has a set of tools called NDT-Suite<sup>1</sup> [12], which, in turn, is composed of the following tools: 1) NDT-Profile, that defines UML profiles in Enterprise Architect<sup>2</sup> for each NDT metamodel; 2) NDT-Quality [13], that allows measuring, automatically, the quality of use of this methodology at each of the software lifecycle phases and checking the correct track of the MDE rules defined in NDT; and 3) NDT-Driver [14], that mechanically follows all QVT rules defined in NDT.

In the last years, NDT has evolved and it actually offers an effective and global work frame for the practical application of NDT, called NDTQ-Framework<sup>3</sup>. This work frame covers a set of processes that takes place in software development processes, quality processes, testing processes, management and security processes. NDTQ-Framework is based on different reference models, for instance, CMMi (*Capability Maturity Model In-*

<sup>1</sup>Available on <http://www.iwt2.org>.

<sup>2</sup>Available on [www.sparxsystems.com.au](http://www.sparxsystems.com.au).

<sup>3</sup>This paper presents, on detail, NDTQ-Framework, but can be found more information on <http://www.iwt2.org>.

tegration) [15] and ITIL (*Information Technology Infrastructure Library*) [16] [17]. Different standards, such as ISO 27001, ISO 9001:2008, UNE EN 16602 and ISO 14000 certify its application to real projects.

It has been impossible to apply NDT methodology directly because of CALIPSONeo project peculiarities. In consequence, we have adapted it for this context, what has provided great results. The modifications carried out are listed below:

On the one hand, as mentioned before, CALIPSONeo is a project consisting of three sub-projects, thus, this aspect has motivated the adaptation of MDE transformation rules of NDT to support the integration and coordination of these projects.

On the other hand, due to CALIPSONeo has delved into a methodological PLM context, it has developed too few software. Therefore, NDT-Quality Profile provides an agile work environment for work teams. Besides, due to the amount of documentation generated on the project, it was necessary to utilize a documental repository based on ECM<sup>4</sup> tools. The methodology offers the possibility of tracking the documents generated as well as requirements.

#### 4. Test Phase and Results

CALIPSONeo uses NDT-Driver tool, one of the supporting tools of NDT, for the testing phase. This way, it is viable to generate, systematically and automatically, the set of system tests from the functional requirements of each CALIPSONeo sub-project.

The execution process of the tests is adapted in accordance with the typology of the project, that is to say, as commented before, that PROTEUS and MARS projects have a methodological nature, whereas ELARA is a software development project.

The validation process in methodological projects comprises three phases. **Figure 3** schematically shows the whole methodological validation process; revision phase, validation phase according to NDT rules, preliminary phase and exhaustive phase.

At the validation phase, the testing team carries out tasks as a technical quality office, reviewing and validating each testing plan (using NDT-Quality) developed by the software's providers in the project. These testing plans are generated, at first, as a draft and they are stored in the documental repository of the project. These quality activities allow the development team to work in parallel and in collaboration with the testing team, reducing time and obtaining exhaustive test plans.

After designing the testing plan (approximately 3/4), the testing team starts with the next phase, the preliminary test phase, where the main objective is to find out the largest number of incidences. At this phase, the testing team has to reproduce each of the activities identified in the methodological document developed within MARS or PROTEUS contexts. For instance, MARS project defines a methodological process to generate Work Instructions, either in Word (to be printed in paper) or in 3DXML (to be rendered in a tablet PC or a computer) documents. These work instructions streamline the task each worker must carry out.

Finally, when the test plan is almost completed (at 90 per cent or more advanced), the testing team performs the exhaustive testing phase with the aim of getting the final acceptance of each methodological proposals defined in MARS and PROTEUS.

Once the last phase of the testing cycle in CALIPSONeo finishes, the testing team writes a testing report with the results, providing the tests typology, the data used and a detailed description of either the incorrect or successful results.

During this phase, and as an added value, the testing team suggests that improvements must be optimized in the methodological process defined.

The application of these testing phases ensures the constant validation of documents as they are developed and it also contributes to implement deliverables quality. This way, the testing phase is boosted up, even being the phase with more cuts over the planning time when projects are out of date.

The testing phase in ELARA project is faced up from a different perspective, but taking into account the methodological bases of NDT. In this case, each testing case is defined using different variables and situations, in which the software must guarantee functionality. For instance, ELARA project has a matching module, which using the tablet PC camera and a virtual model (obtained from the iDMU), synchronizes the real image with the

<sup>4</sup>ECM [18] is the acronym for *Enterprise Contents Management* and the solution used in the CALIPSONeo context was Alfresco. More information is available on the Website.

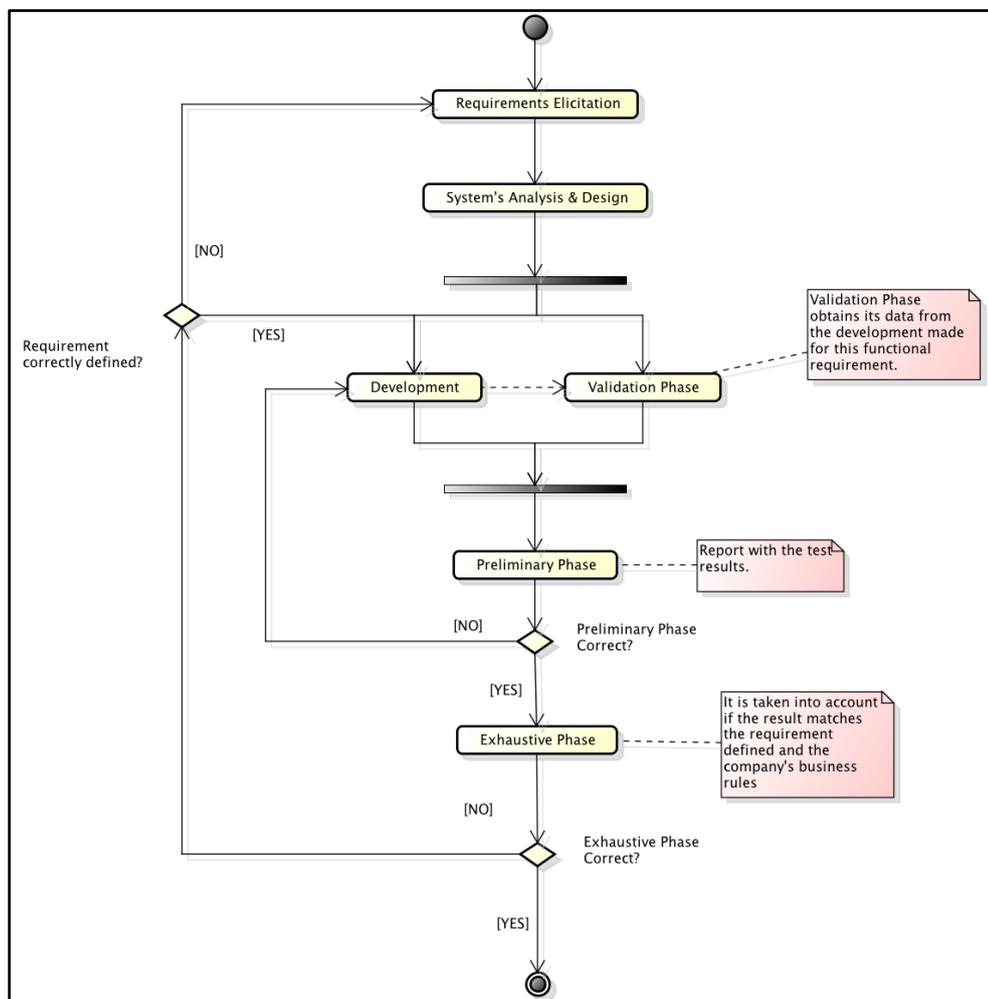


Figure 3. Test phase diagram.

virtual image, helping the shop floor workers perform their assembly tasks dynamically and without errors.

One of the most serious problems we have found out during the test phase of CALIPSOneo project is the low maturity of the diagrams outlined in the analysis and design phases, having as a consequence a test case that does not totally cover all the functional requirements needed. This fact has led the team towards redefining the requirements diagrams and regenerating them again with NDT/Driver.

To conclude, it must be stated that NDT-Suite is efficient in industrial projects, even in Software or Methodological Environment, as it improves the specification of the statement “*What is wanted to be obtained*” and permits its development and validation.

## 5. Conclusions

This paper presents an analysis of the work performed in the context of CALIPSOneo project, by applying Model Driven Engineering to handle the work run through functional tests.

This work provides us with the possibility of validating our research project in a real context. NDT is a proposal applied in many real contexts, but PLM context has several special characteristics that made it attractive to certify the validation of NDT in this environment.

Regarding the executed work, we can conclude that our proposal offers a valid solution, but requires a lot of manual activities from people directing the project and the testing team.

CALIPSOneo allows us to identify a set of requirements to improve our proposal. We have confirmed the importance of determining correctly the system’s requirements to guarantee well-defined tests. This has offered

us a new way to research on any characteristics of NDT-Suite. NDT-Suite includes NDT-Quality, which validates the quality of the artifacts defined by NDT. We have planned to investigate whether we could enrich it to schematize, or even automate the functional revision of requirements for the first phases of the software life-cycle.

Other aspect that must be improved is the concrete definition of PLM context, this way we will raise tools to manage tests.

## Acknowledgements

This research has been supported by MeGUS project (TIN2013-46928-C3-3-R) of the Ministerio de Ciencia e Innovación and NDTQ-Framework project (TIC-5789) of Junta de Andalucía, Spain, and CALIPSOneo Project. This paper has also been supported by the Universia Foundation through its student grant for PhD students.

## References

- [1] Bertolino, A., Marchetti, E. and Muccini, H (2005) Introducing a Reasonably Complete and Coherent Approach for Model-Based Testing. *Electronic Notes in Theoretical Computer Science*, **116**, 85-97. <http://dx.doi.org/10.1016/j.entcs.2004.02.084>
- [2] Escalona, M.J. and Aragón, G (2008) NDT: A Model-Driven Approach for Web Requirements. *IEEE Transactions on Software Engineering*, **34**, 370-390.
- [3] Stark, J. (2011) Product Lifecycle Management: 21st Century Paradigm for Product Realisation. Springer London, London. [http://dx.doi.org/10.1007/978-0-85729-546-0\\_1](http://dx.doi.org/10.1007/978-0-85729-546-0_1)
- [4] Object Management Group (OMG) (2011) UML Testing Profile. <http://utp.omg.org/>
- [5] Zhen, D.R. (2004) Model-Driven Testing with UML 2.0. *Proceedings of the 2nd European Workshop on Model Driven Architecture*, Canterbury, Canterbury, September 2004, 179-187.
- [6] Beizer, B. (1995) Black-Box Testing: Techniques for Functional Testing of Software and Systems. John Wiley & Sons, Inc., Hoboken.
- [7] Lamanca, B.P., Mateo, P.R., de Guzmán, I.G.-R. and Usaola, M.P. (2008) Propuesta para pruebas dirigidas por modelos usando el perfil de pruebas de UML 2.0. *Revista Española de Innovación, Calidad e Ingeniería del Software*, **4**, 36 Pages.
- [8] Escalona, M.J., Gutiérrez, J.J., Mejías, M., Aragón, G., Ramos, I., Torres, J. and Domínguez, F.J. (2011) An Overview on Test Generation from Functional Requirements. *Journal of Systems and Software*, **8**, 1379-1393.
- [9] Gutiérrez, J., Aragón, G., Mejías, M., Domínguez, F. and Cutilla, C.R. (2012) Automatic Test Case Generation from Functional Requirements in NDT. WebRE 2012.
- [10] Más, F., Ríos, J., Menéndez, J.L. and Gómez, A. (2013) A Process-Oriented Approach to Modeling the Conceptual Design of Aircraft Assembly Lines. *International Journal of Advanced Manufacturing Technology*, **67**, 771-784.
- [11] Más, F., Gómez, A., Menéndez, J.L. and Ríos, J. (2013) Proposal for the Conceptual Design of Aeronautical Final Assembly Lines Based on the iDMU Concept. *IFIP International Federation for Information Processing*, **409**, 10-19.
- [12] García-García, J.A., Ortega, M.A., García-Borgoñón, L. and Escalona, M.J. (2012) NDT-Suite: A Model-Based Suite for the Application of NDT. *Web Engineering*, **7387**, 469-472.
- [13] García-García, J.A., Victorio, J., García-Borgoñón, L., Barcelona, M.A., Domínguez-Mayo, F.J. and Escalona, M.J. (2013) A Formal Demonstration of NDT-Quality: A Tool for Measuring the Quality using NDT Methodology. *The 21st Annual Software Quality Management (SQM) Conference*, Not Yet Published.
- [14] García-García, J.A., Cutilla, C.R., Escalona, M.J., Alba, M. and Torres, J. (2011) NDT-Driver, a Java Tool to Support QVT Transformations for NDT. *The 20th International Conference on Information Systems Development (ISD 2011)*, Edinburgh, 24-26 August 2010, 170-176.
- [15] Chrissie, M.B., Konrad, M. and Shrum, S. (2011) CMMI<sup>®</sup> for Development: Guidelines for Process Integration and Product Improvement. Editorial Pearson Education.
- [16] Jong, A. and Kolthof, A. (2008) Fundamentos de ITIL, Volumen 3. Van Haren Publishing, Zaltbommel.
- [17] (2013) Query/View/Transformation. [www.omg.org/spec/QVT/1.1/](http://www.omg.org/spec/QVT/1.1/)
- [18] Gilbert, M.R., Shegda, K.M., Chin, K., Tay, G. and Koehler-Kruener, H. (2012) Magic Quadrant for Enterprise Content Management.