FIBO™ Build, Test, Deploy and Maintain Methodology

Dennis E. Wisnosky and the FIBO Leadership Team

This document is not intended to be the definitive guide for FIBO Build, Test, Deploy and Maintain. It is not the compendium of all material. Rather it references the most current URLs containing detailed instructions, files, etc.

Table of Contents

FIBO[1] Build, Test, Deployment and Maintenance Methodology

1.0 Introduction

The Financial Industry Business Ontology (FIBO) is a family of RDF/OWL ontologies developed as an open standard by the EDM Council. In order for FIBO to become a widely accepted standard to be implemented in many systems across the industry, the Council is testing and demonstrating FIBO ontologies as they move through a rigorous process and become available to the industry. Important activities making this happen include:

- A FIBO community including ontologists, vendors, enterprise architects, business architects, technical architects, programmers, database specialists and subject matter experts from among the financial institutions, regulators and their consultants and service providers.
- An infrastructure including a platform and a methodology where the FIBO community collaborates in a structured process based on "Continuous Improvement"[2] Agile principles and Model Driven Architecture™[3].
- Working within a defined process model that embodies automatic testing, and results in FIBO being accepted Internationally as a Standard.
- Collaborating with the Object Management Group (OMG) in publishing FBIO as an OMG specification.

A highly specialized FIBO Leadership Team (FLT) consisting of subject matter experts in financial services, project and standards process management, ontology and architecture - oversees all aspects of FIBO. In order for FIBO to gain the acceptance necessary for its continued success, FIBO has the participation of contributors with a wide variety of skills and backgrounds in FIBO Content Teams (FCT) and Proof of Concept Teams (FPOCT) made up of domain experts, regulators, tooling vendors and others.

FIBO follows the model of other successful open source projects. In these projects, material is accessed and adapted for a person, or an organization's, own use. The most successful (i.e., reusable and valuable) adaptations are fed back into FIBO for others to use.

FIBO is viewed as being very similar to the build of a complex, open-source software system. Ontologies are treated in the same way as "sources" like Java or C++ with the same challenges such as managing dependencies and merging differences. Modern software engineering principles are therefore being applied to the process of building FIBO ontologies including Test Driven Development (TDD), Continuous Integration and Continuous Improvement. Additional

[1] FIBO is a Trademark of the Enterprise Data Management Council (EDMC)
[2] Including "Continuous Integration", "Continuous Delivery" and "Continuous Deployment"
[3] Model Driven Architecture is a Trademark of the Object Management Group (OMG)

principles specific to ontology development include adherence to modeling policies and best practices, testing for logical consistency and testing for deductive closure.
These practices are highly dependent on automation to merge Build, Test, Deploy, and Maintain into a continuous process. This is the path that FIBO follows, wherein:

Build is the process of fleshing out FIBO as both UML and RDF/OWL models using OMG and W3C Standards, respectively. FIBO Builds are maintained privately in a GitHub repository and publicly here. All FIBO products are generated or derived from FIBO RDF/OWL. There are more than 300 total Ontologies in 30 Domains and Subdomains.

Test includes unit tests, integration tests, stress tests and instance data tests that are automatically run against the ontologies and related artifacts regularly to ensure conformance to requirements. The FIBO Process Team (FPT) uses Jenkins. to manage testing of FIBO builds.

Deployment for FIBO began when an initial FIBO Conceptual Ontology (BCO) began to be formed in 2008 by a team of Subject Matter Experts. FIBO is being used in the industry today. Examples can be found here. At this writing, FIBO deployment includes popular OWL formats, SKOS, a Natural Language Glossary, UML conforming to the OMG Semantic Modeling for Information Federation (SMIF) specification, Visual OWL, and Linked Data Fragments. Within the industry, FIBO could be used as the basis for a database, be integrated with reasoners, be offered as a web service, or within a service, or some other particular architecture.

Maintenance is continuous in a FIBO GitHub Development Repository and published as changes are made by FIBO Content Teams (FCT's.) Production FIBO is the tested and vetted and is published quarterly. Links to the FIBO Infrastructure are here. FIBO automation, described in detail in this document, extends maintenance to all FIBO participants in the industry for both fixes and enhancements.

The FIBO is a grand vision; it is a standard that will be used across the financial industry for decades. It will grow and adapt as the years go by and user demands change. It is therefore crucial that the FIBO Build, Test, Deploy and Maintain process have clear and unambiguous goals expressed below:

For Build:

- Securely and unambiguously manage all FIBO content
- Automatically perform needed conversions into the FIBO publication formats
- Document and measure quality metrics for FIBO publications

For Test:

- Execute a testing process automatically from GitHub with results documented: "tests include confirming uniform use of documented patterns, e.g., hygiene tests (verifying conformance to published style guidelines, searching for undefined references, etc.) and logical tests (conformance to FIBO policies with respect to OWL)"
- Ensure that these unit tests (including SPARQL queries and representative OWL individuals) are executed on as many vendor platforms as possible, enabling vendors to contribute to

the further Build of FIBO and allowing them to demonstrate their technology and how FIBO can be applied to the community and the industry as a whole

For Deploy:

- Create a demonstration platform where FIBO can be shown with several standard datasets
- Maintain a documentation platform, where documentation for various audiences is automatically generated from the ontologies and optionally also from the datasets (showing visualizations of both ontology classes as well as individuals)
- Work with early adopters in financial institutions and regulators that have agreed to provide sample deployments showing how FIBO can be integrated into real processes
- Create a documented set of deployment best practices based on this experience

For Maintain

- Provide for life cycle configuration and issue management
- Encourage deployment teams in the industry to provide feedback through a formal infrastructure
- Use automated tools to rationalize or refactor FIBO as necessary

2.0 FIBO Ecosystem

Achieving these objectives uses a FIBO ecosystem with the characteristics shown in the table:

2.1 Collaboration Requirements

FIBO demands a process that allows practitioners of varied backgrounds from the finance industry, their consultants and service providers to view, review, and contribute to FIBO. The standard is free, open and available to financial institutions, regulators, ratings agencies and other players in the industry.

FIBO cannot be developed by the heroic efforts of a single person or even of a single dedicated team. It requires review and contribution from the whole community. The FIBO infrastructure therefore allows for the dissemination of the ontologies as well as a means for accepting feedback.

2.2 Quality/Testing Requirements

FIBO will be used as a 'best practice' reference throughout the industry. This means that FIBO publications will be held to a high standard of scrutiny for quality. A job of the FIBO Leadership Team (FLT), defined below, is to enforce high standards of quality for the published models, ontologies and specification documents including strict configuration management. FIBO Build, Test, Deploy and Maintain is done in a wide variety of tools governed by strict EDMC guidelines. These guidelines are well documented, and, to the extent possible, supported by automated tools. The FLT is able to determine whether a submission satisfies the FIBO quality requirements through the use of an open source utility known as Jenkins, which is described below. Jenkins is automatically linked to JIRA, a de facto standard issue management tool, also described below.

2.3 Language Requirements

Some of the most exacting requirements are on the language(s) in which the FIBO ontologies are published. FIBO is published in a Development version – all of FIBO, vetted and unvetted, and a Production version – vetted by Domain specific SME's. They are all here. Additionally, a FIBO Primer provides a layman's view of FIBO Products.

2.3.1 SMIF

Because FIBO will be used for decades, any language used to express FIBO is required to have the backing of major standards organizations, and be stable. Most importantly, FIBO must be both machine readable, and people readable.

For the first requirement, machine readable, FIBO ontologies are published in OWL 2[4], the World Wide Web Consortium (W3C), an international standards body, recommendation for sharing ontologies on the web. For the second requirement – people readable, FIBO ontologies are published in UML[5]. To meet this dual requirement, the Council is supporting an evolving Object Management Group[6] (OMG) specification named Semantic Modeling for Information Management (SMIF)[7]. SMIF is a specification that describes corresponding OWL and UML models and diagrams. This enables SMIF compliant tools to perform a feature known as 'round tripping'. That is move seamlessly between OWL and UML. This is further described below.

2.3.2 OWL and Corresponding UML

As stated, the standards published by the EDMC will be in a SMIF representation. Thus, any SMIF compliant UML tool should be able to import either the OWL or the SMIF representation, facilitate documentation via diagrams that represent the OWL 2 for non-technical audiences, edit the SMIF representation, and export conforming to the RDF/XML serialization of OWL. The FIBO standards include the normative OWL 2 representation in RDF/XML and normative diagrams in SMIF, in order to facilitate understanding and interoperability with a broad range of tools and repositories.

One of the design requirements of OWL, as an ontology language for the web, is that ontologies in OWL are interoperable with other distributed ontologies. It is possible to transform OWL ontologies, by virtue of their representation in RDF[8], into and from other modeling systems. This makes OWL a good choice for an interoperable ontology language. In particular, highly skilled architects and ontologists can use their own tools to perform high-volume and high-accuracy ontology updates, and transform the result into W3C-standard OWL for sharing with FIBO and related communities.

---

[4] See http://www.w3.org/standards/techs/owl#w3c_all for the set of documents comprising the OWL 2 Language Recommendations. OWL is the "the Web Ontology Language"

[5] See http://www.omg.org/spec/UML/ FIBO is using UML and OWL 2 in concert with an evolving OMG spec known as SMIF "Semantic Modeling for Information Federation"

[6] http://www.omg.org/spec/UML/

[7] https://github.com/ModelDriven/SIMF/blob/master/RevisedSubmission/SIMF-Submission.docx

[8] https://www.w3.org/RDF/

OWL can be read, processed and produced by a wide variety of tools, both commercial and open source. For desktop review and editing of OWL ontologies, the open source tool Protégé[9] is the tool of choice. Others include the free version of TopBraid Composer, a commercial tool from TopQuadrant.

There are several open source databases and APIs that can read and process queries over OWL, including the OWL-API, RDFLib, Jena and RDF4J (formerly Sesame), as well as databases OpenLink Virtuoso and others. Inferencing capabilities for OWL include open source (Pellet, Hermit) and commercial tools, many with free editions (OntoText's GraphDB, SPARQLverse). Commercial database products include Anzo (Cambridge Semantics), AllegroGraph (Franz Inc), MarkLogic, and Stardog (Stardog Union), Oracle and IBM. Adaptive, No Magic and TopQuadrant are providing web-based access to FIBO for search and graphical exploration. Wizdom Systems, Inc. can provide an LEI Resolver Service based on FIBO.

FIBO must accept contributions from a wide range of tools, and it is imperative that the FIBO curators be able to tell what has changed from one version to a proposed next version. The best understood and most successful way to achieve this is through tools that can provide text-level diff and merge capabilities. This means that FIBO is published in a format that is stable, i.e., small changes in an ontology result in small (localized) changes in the representation. The EDM Council therefore provides a stable, predictable serialization of RDF as the standard storage format for its ontologies. "Stable" and "Predictable" mean that small changes in an ontology result (for the most part) in small, localized changes in the files. This allows source control systems like git to automatically merge modification from multiple users with a minimum of human intervention. The Council makes available an open source "RDF Toolkit" so that FIBO users can test the integration and publication of FIBO in their own environments.

2.3.3 Glossary

Generated from the OWL, the FIBO glossary is a list of terms defined in FIBO, including their generated definition, natural language definition and OWL properties. As with all of FIBO, there is a Production version of the Glossary, which includes definitions of terms that have been vetted by the FIBO content teams, as well as a Development version that is vetted and unvetted FIBO.

2.3.4 Data Dictionary

Generated from the OWL, the FIBO Data Dictionary is in an Excel Workbook format. The Data Dictionary includes the Class, the human and machine generated definitions and the properties. The Data Dictionary can be downloaded and opened in any application that can read .xlsx.

2.3.5 Vocabulary

Generated from the OWL, The FIBO vocabulary is distributed using the Simple Knowledge Organization System (SKOS). It is a machine-readable dictionary of the FIBO terminology in a variety of RDF formats.

[9] http://protege.stanford.edu/

3.0 FIBO Infrastructure

3.1 EDMC - Business Ontology

FIBO development began in 2008 at the onset of the Global Financial Crisis. Thirty Finance Industry Domain Business Ontologies were developed by the Council's Head of Semantics and Standards and leading various Subject Matter Experts within the industry. These Domains and their status is shown here as are the names and schedules of FIBO Teams described herein.
It is from this list that FIBO Content Teams (FCT) described in Section 2.3, select a particular Domain to enter the EDMC standards process. In this process, each FIBO is tested in its OWL format for mathematical correctness, is repeatedly reviewed by both Subject Matter Experts (SMEs) and ontologists, is documented, published as an EDMC standard and is offered to the Object Management Group to become an OMG specification. This EDM Build – Test process is outlined in Figure 1. Figure 2 Shows the FIBO Deploy - Maintain process. Appendix A shows an expanded version of these diagrams.

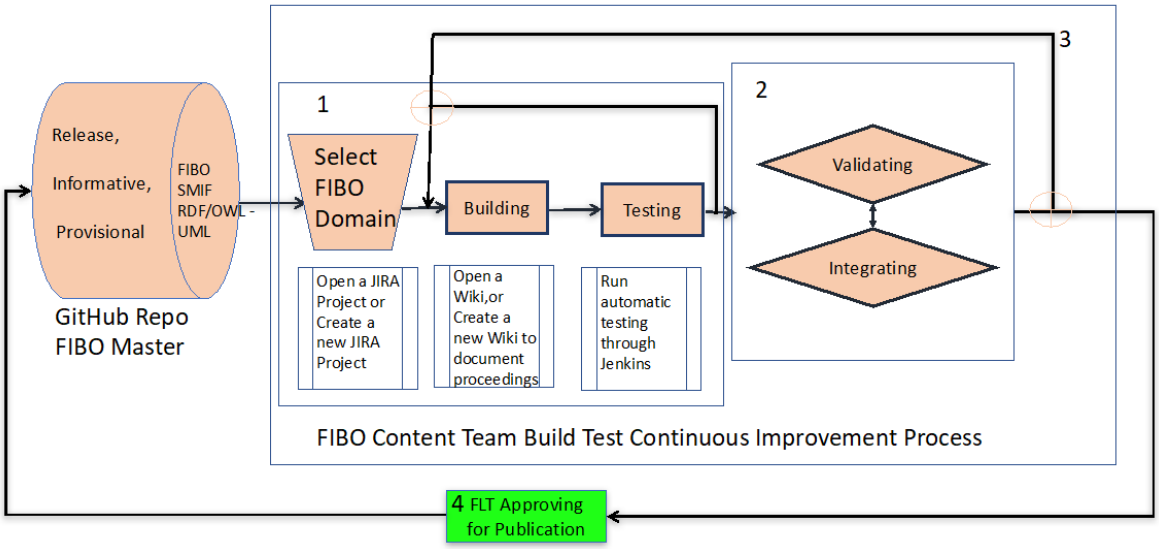


Figure 1 - FIBO EDMC Build Test Process

The FIBO GitHub Repository contains FIBO in 3 maturity levels:

- Informative models are ones that have been considered by a development team, but have been explicitly rejected. They are included in FIBO sources because they include references, without which FIBO would fail the basic referential consistency tests.
- Provisional ontologies were developed in the early days of FIBO, or have been developed by FIBO Content Teams but have not been vetted or tested to the level of release.
- Release models have undergone extensive unit and integration testing, and have passed the most rigorous tests for completeness, consistency and correctness.

FIBO Content Teams may select their Domain or Domains of interest from the GitHub Repository. FCTs meet at least weekly in a standard meeting format that is described later in this document. Building is accomplished using a combination of OWL and UML. Testing is through

specific Hygiene protocols developed by the FIBO Process Team (FPT) and using off-the-shelf reasoners. Because FIBO is a single integrated model, it is critical that the content built and tested by individual FCTs is supportive, not destructive of other FCT's. The FIBO Leadership Team then approves content.

As shown in Figure 2, FIBO content is Deployed and Maintained in a highly automated publishing process managed by the EDMC developed RDF ToolKit. As changes are made in Release and Provisional and Informative they are published as FIBO Development with caveats that its contents cannot be guaranteed. This is 100% of FIBO. FIBO Production is published along with FIBO Development at the end of each quarter.

Both Development and Production are published in all languages and formats described here.

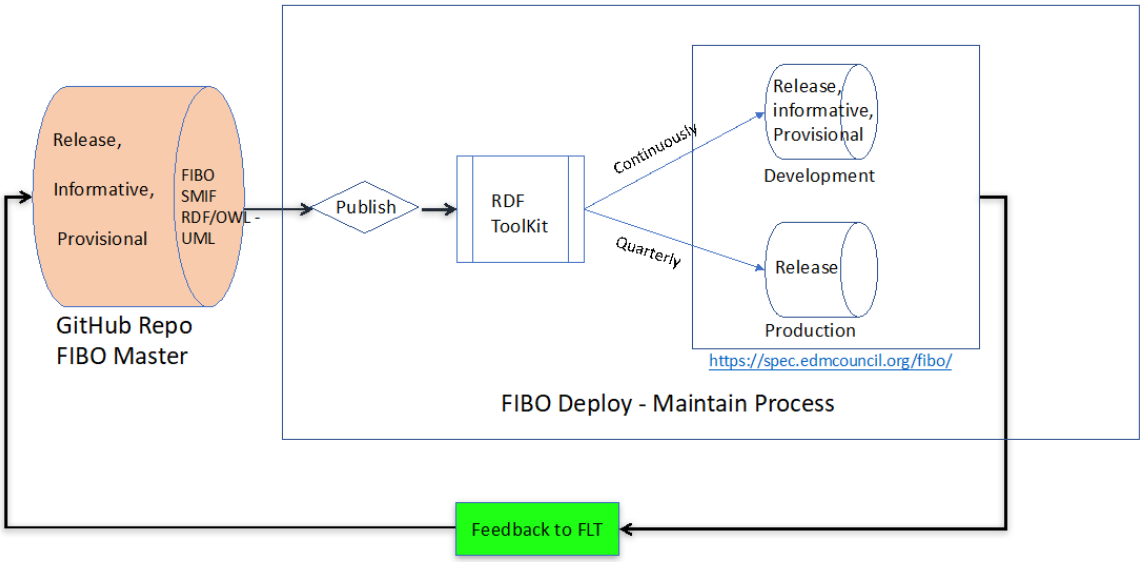


Figure 2 - FIBO EDMC Deploy - Maintain Process

This process is managed by state of the art collaboration tools within a complex infrastructure of both people and machines. The machine component of the infrastructure is described here and in section 4.

The people component of the infrastructure consists of FIBO Teams:

- A FIBO Leadership Team (FLT)
- One or more FIBO Content Teams (FCT) operating simultaneously
- One or more Proof of Concept Teams (FPOCT) operating simultaneously
- Supporting these teams, are:
    - A FIBO Process Team (FPT) working to establish and maintain the FIBO infrastructure
    - FIBO Vendor Teams (FVT) providing FIBO test suites using their COTS tools

- Additionally, Banks, Regulators and Universities from time to time form FIBO Implementation Teams. The Council asks that these organizations contribute their content back to FIBO

3.2 FIBO Teams

This section elaborates on the FIBO process, including the machine and person components of the infrastructure. The interdependence of these teams cannot be over emphasized. There is clear overlap between and among the teams that build, test, deploy and maintain their FIBOs. As a practical matter, at any given point in time, a person working on FIBO can expect to contribute his or her expertise to at least two FIBO teams, each of which has a distinct responsibility. For the current FIBO Team meeting status, go here.

3.2.1 FIBO Leadership Team (FLT)

FLT members include ontologists, architects, subject matter experts and standards process specialists. The role of the FLT includes:

- Develop new and utilize existing standards, frameworks and guidance
- Build a strong FIBO roadmap and strategic framework
- Establish and maintain a FIBO methodology document
- Establish an accessible repository for all stages of FIBO
- Oversee FIBO governance and configuration
- Adopt and adapt current research and innovation
- Adjudicate changes in GitHub and JIRA
- Establish and mentor all other FIBO Teams

In addition, the FLT meets in person at OMG technical meetings offering FIBO content to OMG, from time to time at other venues, and in weekly scheduled GoToMeeting sessions.

3.2.2 FIBO Content Teams (FCT)

FIBO Content Teams are formed when industry leaders decide that one or more of the FIBO Domains in FIBO Development, can support a use case important to their organization. This starts the Process.

FCTs are designed to build FIBO with content that has been developed by SMEs from the Finance industry, tested to the highest requirements published by the EDMC and delivered to OMG. This includes SMIF compliant RDF/OWL - UML models and documentation. Because of this, and the fact that several FCTs will be working simultaneously, it is necessary that each FCT operate under identical guidelines. In between these meetings, Ontologists and other experts prepare for SME reviews.

FCTs are composed of at least five people mentored by at least one member of the FLT. The Team leaders should be from within the FCTs FIBO Domain and include: an SME who understands the idea of ontology, an ontologist who understands the Finance Domain, a qualified super ontologist, or an SME from within the Domain who is also an ontologist. Broad FCT membership is encouraged from throughout the industry including at least two competitors to the Team leaders firm.

The role of an FCT includes:

- Develop Use Cases or User Stories, that can be satisfied by one or more of the FIBO Domains:
  - These will be submitted to the FLT and published on the EDMC website
  - They are to be used by the FCT that created them, or by FPoCT teams (see below) to organize proofs of concept
  - Provide Test Data for the Use Cases
- Follow the FIBO Development Process to update the model:
  - Using a standard Git Tool, download the desired Domain(s) from GitHub
  - Use SMIF UML and the RDF/OWL models in the standards based tool of choice to modify the existing Domain(s) to suit the objectives of the Use Case; This includes:
    - Review and validate pre-existing class hierarchies, properties, and restrictions editing and extending each as needed to address both the Use Cases and the integrity of the Domain
  - Carry out SME Reviews[10] on the resulting model
    - Render restrictions describable as refinements and/or reuse of properties
    - Render existing classes and properties and disjoints and inverses
    - Edit definitions
    - Harvest additional synonyms
    - Review and edit provenance metadata ensuring that all references to other namespaces in other relevant ontologies are referenced appropriately. E.G. LCC in OMG.
- Create spreadsheets or other business accessible formats to explicate terms, definitions, synonyms, and metadata, etc. reports for off line review
- Ensure that the model loads in Protégé and CCM and run the modified RDF/OWL model through Pellet and Hermit as Jenkins jobs
- Document progress in the FCT Wiki
- Adjudicate issues in EDMC JIRA
- When the FIBO is ready for the Release Maturity Level, build required publication documentation

3.2.3 FIBO Proof of Concept Teams (FPoCT)

FPoCTs are established from time to time by EDMC members to use one or more FIBO Domains in a real world environment that can test the feasibility, or to prove the cost benefit of using FIBO in a particular use case for an operator or for a regulator. As with FCTs, the most important aspect of an FPoCT is a leader with a Use Case or User Story. These can be generated by the FPoCT team itself, or taken from other proposed Use Cases or User Stories as published by the EDMC. An FPoCT can form beginning with any stage FIBO.

[10] Subject Matter Expert (SME) reviews are intended to capture and /or validate the business semantics of the FIBO ontologies. These are carried out using any FIBO Products as well as ancillary materials appropriate to the FCT.

A Use Case or User Story may begin with a discussion that no FIBO content is supportive, for example. Then the FPoCT actually would create a new FIBO. At the opposite end of this possibility, a Use Case or User Story in advance of starting a FIBO Implementation Team would begin with one or more existing FIBO. PoCs (Proof of Concepts) are carried out by member organizations to demonstrate how FIBO can be used in their business. While FPoCTs are users of FIBO, not developers of FIBO, it is the intention and expectation that FPoCTs use the FIBO Build, Test, Deploy and Maintain process and tooling as appropriate and that FPoCT meetings follow the standard FIBO FCT Meeting process.

In contrast to the activities of the FLT during FIBO development, it is not necessarily expected that the data in a FPoC will be unclassified and readable by the public or even by other EDMC members. The deliverables should include at least a lessons learned document, outlining strengths and shortcomings of FIBO, and in the latter case, recommendations for improvement, and issues that can be acted on by a content team or the FLT.

If appropriate, a demonstration or video of the PoC is a good deliverable.

3.2.4 OMG Task Force (OMG/TF)

The OMG process involves establishing a Finalization Task Force (FTF) when a draft set of documents on a particular new standard is approved by the OMG Architecture Board. It is the responsibility of the FTF to adjudicate issues, in OMG JIRA, that are surfaced during a public comment period. Issues that are not resolved by an OMG/FTF and new issues that arise over time, based on usage, may be resolved via the FTF or are transferred, once the specification is finalized, to a Revision Task Force (RTF). When the OMG AB is satisfied that the issues have been addressed, the standard is proposed for adoption by the broader OMG membership, at which point it becomes a formal specification

3.2.5 FIBO Implementation Teams (FIT)

FITs are created by the operators with the sole desire to solve particular business problems. FITs extend FIBO BCOs into FIBO Operational Ontologies (FOO). These FOOs are executable designs that solve a particular business problem of the operator.
A FIT will be staffed mainly from a member organization, but the FLT may be asked to provide mentoring advice to the FIT at all stages of development, from use case identification to identifying an appropriate FIBO to use, to the development of an appropriate FOO. Lessons learned by supporting FITs in this way will be fed back in to the core FIBO ontologies and documentation.

3.2.6 FIBO Process Team (FPT)

The FIBO Process Team is responsible for establishing and maintaining the FIBO Infrastructure. This includes management of web servers for FIBO Namespaces, GitHub/Jenkins management and how they are referenced in EDMC JIRA. In particular, the FPT is responsible for liaison with all vendors, helping them to set up Jenkins slaves for FIBO testing. While the FIBO Content Teams are primarily responsible for the development of the tests themselves, the FPT is responsible for the mechanism of automatic test jobs, making sure that these tests run correctly on the Jenkins servers, and the FIBO content build and publication process is optimized.

The FPT includes members from other FIBO teams who represent both user requirements and particular technical expertise in one or more of the infrastructure platforms/systems. The FPT is in constant communication with OMG and EDMC webmasters to insure synchronization of all applicable websites.

3.2.7 FIBO Vendor Team (FVT)

Vendors represent a broad user community, for themselves and for their clients, through their expertise in platforms, technological insight, and research that intersects with domain depth applied to real-world business problems.

To participate in the FIBO Continuous Integration and Improvement lifecycle, as a designated FIBO Vendor Team, a vendor must provide a Jenkins slave platform for testing FIBO use cases as part of the EDM-Council's Jenkins-JIRA infrastructure so that the Council's Jenkins server can schedule generic and vendor-specific test jobs directly to vendor products on vendor hardware. Vendors are also urged to make a FIBO 'sandbox' available for their customers to exercise FIBO.

FIBO Vendor Teams are expected to lend their expertise to, and collaborate with, FIBO Content Teams and FIBO Proof of Concept Teams deploying FIBO models to their own platform and tool environments. The FLT is happy to work with any vendor who is willing to provide resources to help determine how that vendor can best contribute to the FIBO community and infrastructure.

Contributions that the vendor provides to FIBO include:

- Testing in real-world scenarios: operational use and test cases outside and beyond those provided by other teams; test suites, structural tests, performance and scalability stress testing; issues, resolutions, and gaps in domain coverage
- Tooling for modeling, visualization, application development and deployment, integration and interoperability
- Data for testing in various business and regulatory scenarios
- Deployment to their own and others' platforms in operational environments, on end-to-end platform ecosystems that may include SaaS, PaaS, and cloud hosting in general
- Compatibility with and conversions between UML, OWL and other modeling paradigms, and the application of reasoning and rules
- FIBO applications and extensions, domain and ontology research and engineering, implementation and deployment stories, possibilities for factoring, and the application of reasoning, rules, and optimization in real-world scenarios
- Free or nominally-priced versions of platform, tools, and related products
- Socialization of FIBO in and among the end-user community
- And, documentation of the above, as appropriate, within the FIBO Build, Test, Deploy, and Maintain infrastructure

4.0 FIBO Process

This section discusses the FIBO Build, Test, Deployment and Maintenance Infrastructure based on GitHub, Jenkins and JIRA being its core components. The simplified EDMC process shown in Figures 1 and 2, and detailed in appendix A, evolves FIBO. To practically participate in this process, it is necessary to study the contents of GitHub, which is the primary repository for all FIBO content including the RDF/OWL files, documentation, use cases, test data, schedules, and more. GitHub manages FIBO which resides on a variety of virtual servers under EDMC namespaces herein referred to as the EDMC Website. At the discretion of the FLT, some of these namespaces are accessible under restricted access for Build, Test, Maintain and here for Deploy.

Jenkins is the primary FIBO Continuous Integration (CI) platform. Jenkins allows automation of the testing process by executing standard test scripts. Go here for details.

Atlassian Confluence is used to document the proceedings of all FIBO Team meetings. Actions are logged as issues in an EDMC JIRA instance as the mechanism for managing FIBO content build and test.

An OMG customization of JIRA is the primary FIBO issue tracking mechanism for OMG issues. A significant challenge is keeping these necessary tools and the data contained within them in synchronization. This is the primary responsibility of the FIBO Leadership Team, but involves all of the other Teams using any FIBO.

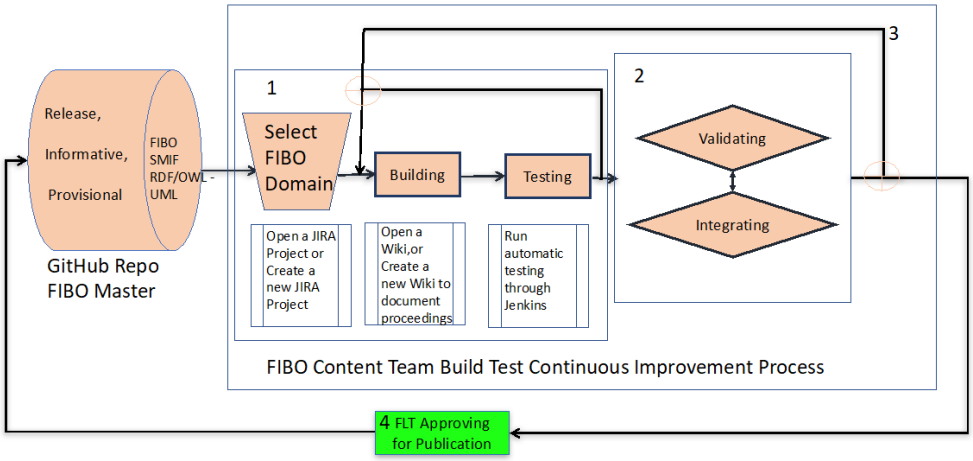


Figure 3 – Relationship of GitHub, Jenkins and JIRA in the FIBO Build-Test Process

The sequence represented by the numbers matches the sequence of what actually happens as a FIBO moves through the steps of Figure 3.

1) A FIBO Content Team is formed as described above, and selects one or more Domains that are in the GitHub Repository at any maturity level. The FIBO is modified as necessary according to the process of Section 3.2.2. Proceedings are recorded in the Team's Wiki. Issues may be discovered and are recorded in EDMC JIRA for management. Some issues are corrected

manually. Use Case data, developed by the FCT, is used to test the FIBO through the automation provided by Jenkins. This loop continues until all issues are resolved.

2) Jenkins testing through reasoners returns no errors meaning that test scenarios provided by the FCT are positively concluded – EDMC JIRA has no unresolved FIBO content issues. The FIBO can be validated with SMEs and integrated with other FCT's.

3) FIBO perhaps has additional issues discovered during validation and integration. FIBO content issues discovered are referenced to EDMC JIRA where they are adjudicated in EDMC GitHub/Jenkins. This loop continues until all relevant JIRA issues are resolved. See the GitHub section below for the discussion on disposition of FIBO Maturity Levels.

4) When all issues of all types are resolved, the FIBO Leadership Team approves the FIBO for publication as Production

4.1 GitHub[11]

The open-source collaboration software Git, and the open cloud-based platform GitHub, have emerged as the de-facto standard for open-source collaboration. GitHub provides an open forum for reviewing and contributing to shared projects, and is very well-suited to the needs of FIBO. This section elaborates on the specifics of the use of GitHub and other infrastructure components for FIBO.

GitHub confers many advantages for a collaborative team. GitHub uses "branches" and "forks" to keep track of proposed and accepted changes in the form of "Pull Requests." Team members can create a fork, modify multiple files, and "commit" them back into their own local "clone" of their "fork." They can then "push" all their "commits" to their fork on GitHub. Any other collaborator can create a clone of that fork as well, or create his own fork and merge his own fork with the original fork at Github, run tests, make comments or propose changes, and create a "pull request" for the "root" fibo repository. The "owners" of that root repository (the FLT) can then accept or reject that pull request. The process is much less error prone than other means of collaboration such as email, thumb drives and unstructured cloud storage because it keeps a running track of all changes. It is a proven method for large, diverse teams to coordinate their work. It will work for non FIBO chartered teams, when anyone can create a fork and make changes in that fork and propose changes to the root repository via Pull Requests. A 15 minute video and this shows how FIBO uses GitHub observing the following policies.

1. FIBO Domains are stored in separate directories in a single Git repository called *fibo*. These directories are named for each of the FIBO Domains, e.g., *fnd*, *be*, *ind*, etc. These directories are siblings in the FIBO repository.
2. Each Domain directory contains a number of directories corresponding to *modules* of that Domain. The names of these directories are lower-case, with multiple words separated by hyphens and no embedded spaces, e.g., ../securities/asset-backed/mortgage-backed-securities.

[11] In GitHub repos, FIBO is lower case fibo. This standard is adhered to in this document also

3. Each collaborator creates a *fork* of the fibo repository. A contribution to FIBO is delivered by means of a *pull request* to the fibo repository. The FLT must accept or reject a pull request in its entirety; therefore, requests that entail multiple atomic changes are less likely to be accepted.

The use of GitHub is closely coupled with policies regarding release management in FIBO. FIBO follows the strategy known as "Semantic Versioning". Namely, a version number has three components called Major, Minor and Patch (or Sequence).

FIBO releases are indicated with tags. As we have different maturity levels of FIBO in different branches that each represents a different phase in the life cycle of the FIBO ontologies, we also have different version numbers because we can set tags in each branch. Tags are also used to mark versions or releases as they are submitted to the OMG, these tags have the format "omg-dtc-<yymmss>" (see for more details the document mentioned below). The format for all tags follows the "semantic versioning" scheme as previously stated, with the following format: "<major>.<minor>.<sequence number>", for instance "1.2.3". This enables use of the Follow Your Nose approach to navigation.

4.2 Jenkins

Jenkins is an open source product that can be installed in a master/slave configuration. The Jenkins master server has a web-based interface that can be used to define and monitor "jobs." Jobs can be activated manually, or by many different types of "triggers." One important trigger is a "push" into the GitHub repository that could trigger the execution of many different jobs on the Jenkins server. Another important trigger is the successful or unsuccessful execution of another job that allows for "pipelines" of jobs that can be run in sequence or in parallel.
One Jenkins master server can manage any number of slave servers. A slave Jenkins server has no web based interface, but is basically a job-execution "container" that can be automatically and remotely installed by the Jenkins server on any target host (provided that the Jenkins master has SSH access to the target host[12]). All jobs are defined on the master, but can be "tagged" to only run on specific slaves or on any available slave based on various metrics. In that sense, the Jenkins server is just a job scheduler.

4.3 Confluence Wiki

Confluence is used to store the proceedings of every FIBO meeting, and record Decisions and Policies. Action items that result are automatically shared with JIRA as issues. JIRA is then used to manage issue resolution.

4.4 JIRA

When there is a commit to GitHub, the commit message starts with the JIRA issue number as follows: "<JIRA issue number> <commit message>". Any GitHub repository, including each fork of it, can thus be connected to a central JIRA server so that each GitHub commit ends up in the Activity log of the appropriate JIRA-issue. Also, JIRA and Jenkins are connected to each other, so that each unsuccessful run of a Jenkins job ends up as a comment in the Activity log of all the JIRA-issues that were involved in the scheduling of the job

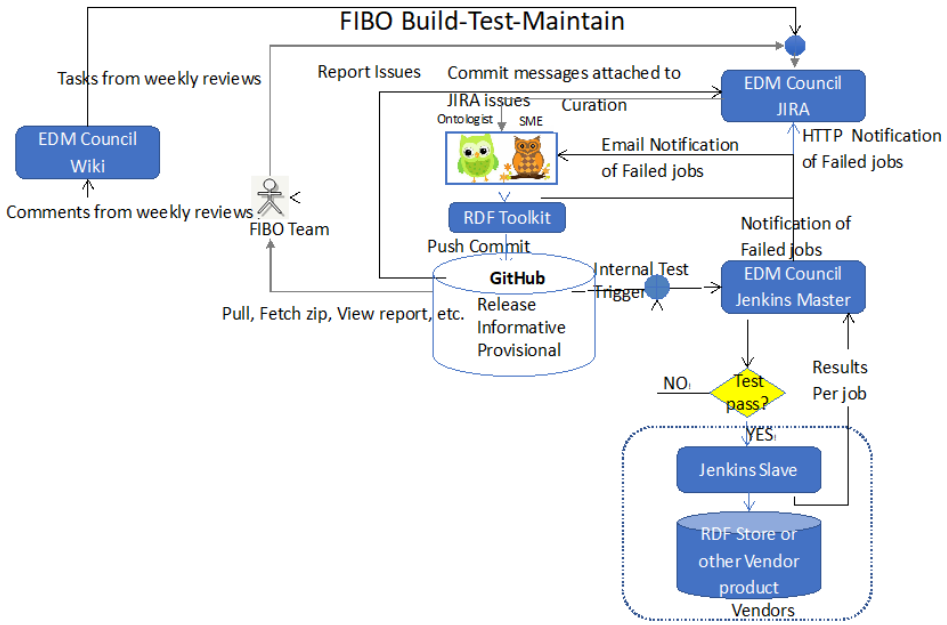


Figure 4 - GitHub/Jenkins-JIRA Collaboration

Figure 4 is a simplified picture of how GitHub, Jenkins and JIRA work together for testing, membership review and reviewer feedback. Appendix A shows a more complete picture. The EDM Council owns and manages the GitHub account, the Jenkins master and EDMC JIRA. Vendors can "plug in" their Jenkins slave server to run as many of the test jobs defined at the master as possible. Each vendor can install their own products on their own slave server, such as an RDF database, where jobs can access that product only when they run on that particular slave. This allows generic and vendor specific jobs to run side by side so that each vendor can opt in to run as many of the FIBO jobs as possible, showing how their product performs and works in the most optimal way. This is true for database vendors and vendors that can generate documentation sites or convert files from OWL to UML or vice versa. Contributing ontologists can work in any tool of their choice, but run their OWL files through a stable serializer to ensure that their files are comparable (e.g., via diff) with those created by their peers.

4.5 Use Cases and User Stories

Critical to the success of FIBO is that it satisfies the business needs of the Finance Industry. These business needs will range from proving regulatory compliance to design or redesign of business processes and IT systems. Understanding and documenting the needs of the user community and how a proposed solution will satisfy those needs has been the bane of engineers in every endeavor of mankind. Diagrams, and other types of flow charts gave way to a structured approach known as "Use Cases" in the 1980s.[13] More recently, usually associated with Agile Software Development, Use Cases may instead be "User Stories."[14] A common theme of both Use Cases and User Stories is that both must contain testing criteria to determine if the user need is satisfied.

[13] Jacobson et al, 1992.
[14] "User Stories". Agile Alliance.

An acceptable Use Case or User Story provides the baseline set of requirements for that Domain/Sub-Domain to graduate from the Provisional to Release maturity level.

For either a User Story or a Use Case to be accepted by the FLT, it must include scripts in SPARQL that perform a viable test on some portion of the emerging model. This is required before an FCT can be commissioned, and prior to an FPoCT. The Use Cases and User Stories will provide the basis for determining whether or not a given concept is needed, whether some set of relationships (properties) are needed, and "when to stop." The FLT will determine whether or not the Use Case or User Story is sufficient to commission an FCT. The choice of presenting a Use Case or a User Story depends on the scope and nature of the business need and the experience of the team.

Use Cases and User Stories can be specific or comprehensive. Use Cases use stick figure diagrams to show relationships. User Stories use Story Maps. Reference[15] contains details on common approaches to for each practice.
An example of a specific Use Case follows:

"As a potential shareholder, I want to see which party controls organization X, so that I can assess who appoints the board of directors." This is often called a Competency Question.
A sample use case diagram for this simple use case above might look like this:

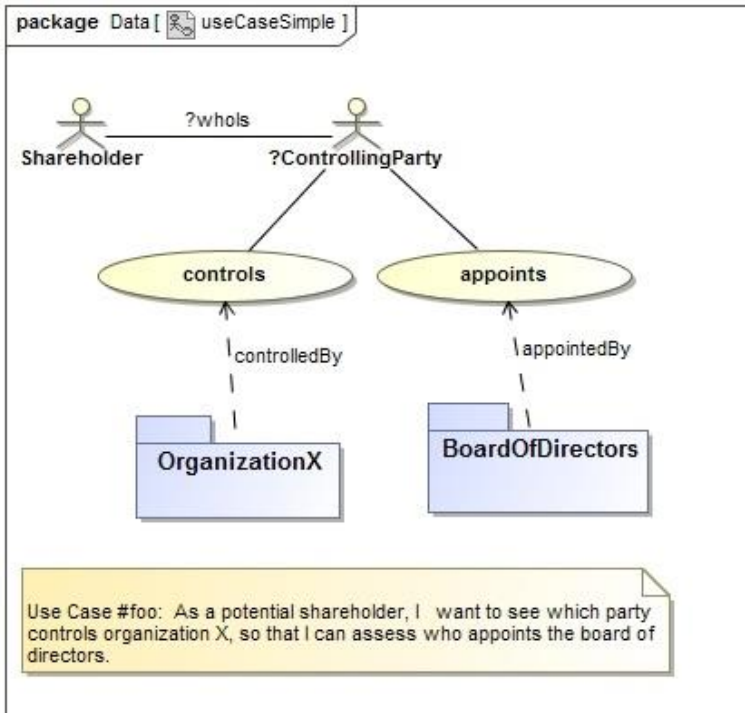


Figure 5 – Sample Use Case Diagram

[15] "User Story And Use Case Comparison". c2.com.

An example of a more comprehensive User Story would be:

"As part of our AML efforts, I need to know which party controls an organization, any entities owned by that organization, and who has a beneficiary relationship to those (sub-) organizations, so that I can determine if certain common money laundering patterns might exist in this organization."

Specific Use Cases and User Stories can be used as drivers for unit tests (see below) during the Modeling Phase, while more comprehensive Use Cases can drive ontology development, e.g., by providing competency questions or driving commonality and variability analysis.
FIBO encourages the collection of Use Cases, especially comprehensive Use Cases, from interested parties in the industry. Contributions of Use Cases require little commitment, but are nevertheless an important and effective way to inform the direction of the FIBO effort.

4.6 Testing

FIBO will adhere to "Behavior Driven Development" (BDD)/"Test Driven Development" (TDD) "Continuous Integration" and "Continuous Improvement" principles applied throughout the FIBO Build, Test, Deploy Maintain process as part of the Continuous Integration Process as previously introduced. See[16] and[17] for additional background on these concepts.
An example of this spiral methodology is shown in the diagram below:

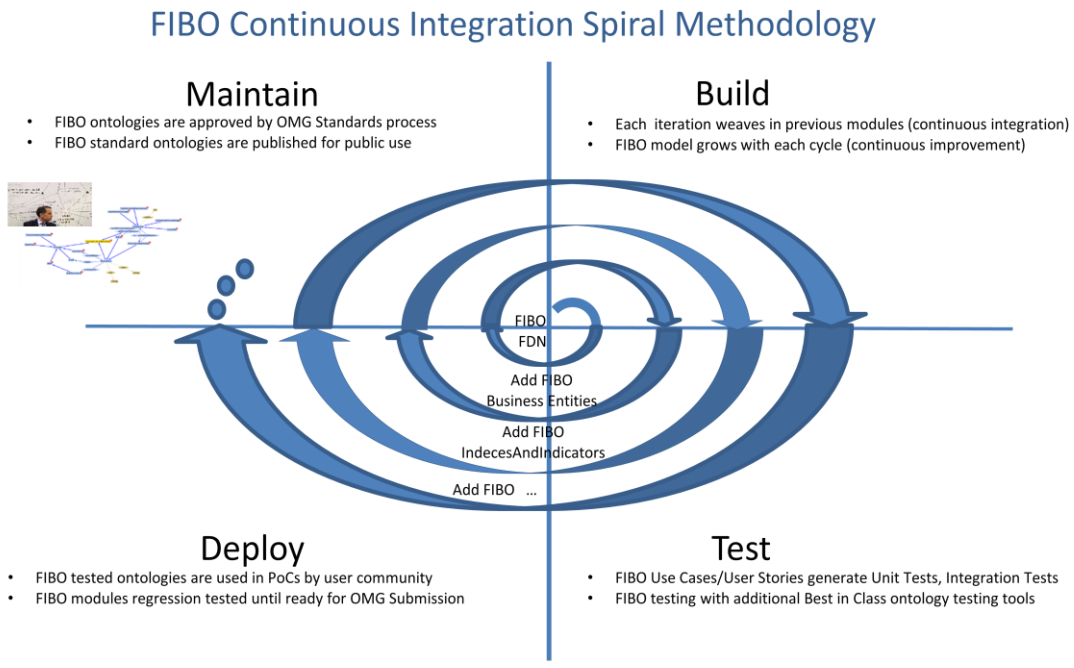


Figure 6 - spiral continuous integration methodology, with testing points highlighted

[16] http://www.slideshare.net/cmungall/ontologies-andcontinuousintegration
[17] http://martinfowler.com/articles/continuousIntegration.html

The FIBO Continuous Integration Cycle includes FIBO model building, testing and validating through integration testing, deploying PoCs to further improve the FIBO modules and ready them for OMG approval, and successfully passing OMG approval, then maintaining the growing community of FIBO modules staged for public use. The testing overlay at each of these stages includes:

4.6.1 Modeling and Unit Testing - Performing unit tests on the FIBO ontologies is critical. Unit testing is a method by which the quality of ontologies can be maintained during the development process. The sources of unit tests are the User Stories and the Use Cases, as previously stated. The test data, the query, and the desired result can all be stored as simple text files, so that conformance to the test can be easily verified.

The Jenkins framework allows for automation of this process. Unit tests will be developed on the submitters workstation, verified in Protégé and when pushed into a repository, Jenkins will run all relevant tests. Any failure to match the desired output will be reported back to the user who pushed that change.

Unit Test of the model assures that the model structure is logically sound, and readies it for the subsequent testing of model substance in the Integration Test process.

4.6.2 Validating Through Integration Testing - The substance validation portion of the cycle includes creating instances/individuals of the portions just modeled, generating SQL/SPARQL statements that implement the simple use cases that apply to the recently modeled portion(s), and running those as initial proofs that the content of the model meets desired results. The new portions of the emergent models are the combined with the rest of the model set, and are run with all SPARQL tests, as an 'integration' test, much like system regression tests. The model (or model portions) may then be run through advanced ontology testing tools, such as OOPS! and Oquare[18], as a final validity cycle.

4.6.3 Documenting - By this stage, the model is basically in tested state for this turn of the cycle, and the Use Cases and testing results may be used as part of the FIBO Documentation.

4.6.4 Deploying - The testing results are then made part of the EDMC publishing of FIBO. While responsibility for the assurance of complete testing logically falls on the FIBO Content and the Proof of Concept Teams, any of the other teams which touch the FIBO Model will be responsible for Unit Testing, and Use Case generation of their portions of the emerging model; and they may be called upon by the FLT for further testing activities

[18] OOPS! Ontology Pitfall Scanner: http://oeg-lia3.dia.fi.upm.es/oops/catalogue.jsp ; and OQuaRE http://miuras.inf.um.es/evaluation/oquare/ .

5.0 Roles

There are several roles to play or "privileges" to have within the FIBO Develop, Build, Test, Deploy, Maintain Ecosystem. These roles are largely defined by the FIBO Team structure.

	GitHub Admin	GitHub Contributor	GitHub Viewer	Jenkins Admin	Jenkins Job Admin	Jenkins Viewer	JIRA User
Leadership Team (FLT)	X	X	X	X	X	X	X
Content Teams (FCT)		X	X			X	X
PoC Team(s) (FPOCT)		X	X		X	X	
Vendor Team(s) (FVT)		X	X		X	X	
Process Team (FPT)		X	X				
Industry Implementation Teams (FIT)		X	X				
Industry			X			X	

Disclaimer

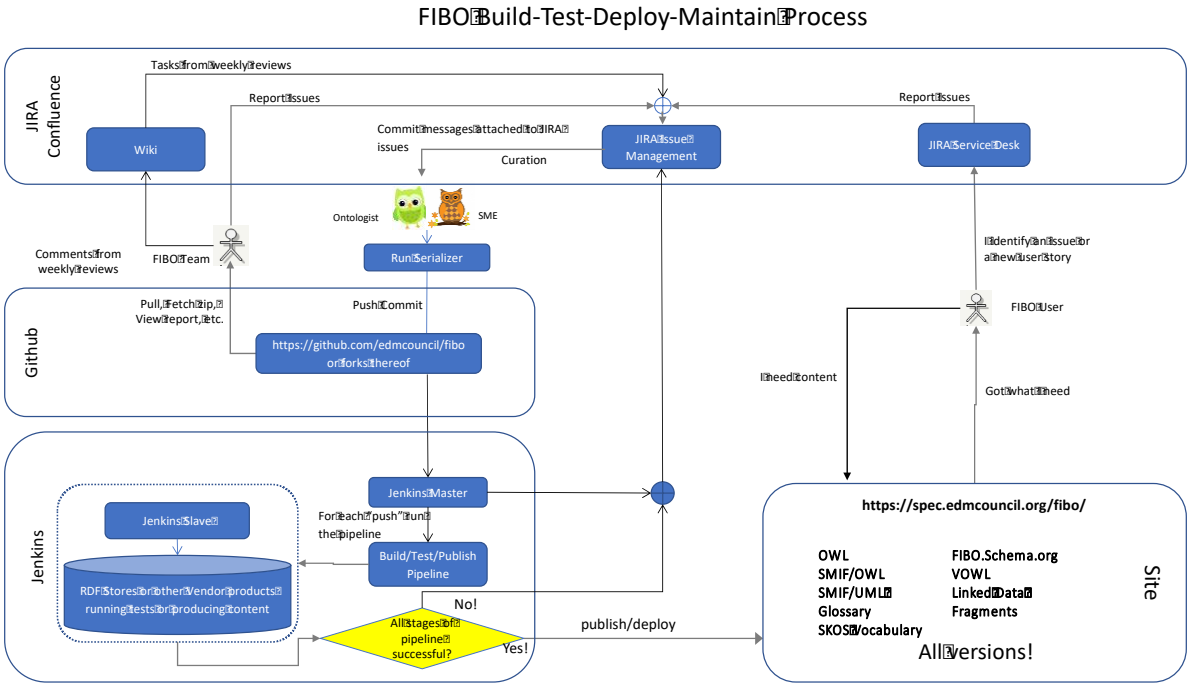
Any opinions, findings, and conclusions or recommendations expressed in this material of those of the author(s) and do not necessarily reflect the views of members of the Enterprise Data Management Council, or any of the organizations affiliated with the FIBO Leadership Team participants.

Acknowledgements

This report is based upon work supported by the Enterprise Data Management Council (EDMC) in cooperation with the following Individuals as members of the FIBO Leadership (FLT).

Michael Bennett,	EDM Council
Cory B. Casanave	Model Driven Solutions
Pete Rivett,	Adaptive Inc.
Elisa Kendall,	Thematix Partners, LLC
David Newman,	Wells Fargo
Dean Allemang,	Working Ontologist, LLC
Richard Beatch,	Bloomberg L. P.
Jim Logan	No Magic
Michael Uschold	Semantic Arts
Jacobus Geluk,	Bloomberg L. P.
Bobbin Teegarden,	OntoAge
Michael Atkin,	EDM Council
Dennis E. Wisnosky,	Wizdom Systems, Inc.

Appendix A


FIBO Build-Test-Deploy-Maintain Process
JIRA Confluence
Tasks from weekly reviews
Report issues
Report issues
Commit messages attached to JIRA issues
Curation
Wiki
JIRA Issue Management
JIRA Service Desk
Ontologist
SME
Comments from weekly reviews
FIBO Team
Run Serializer
Identify an issue or a new user story
Github
Pull, Fetch/zip, View report, etc.
Push Commit
https://github.com/edmcouncil/fibo or forks thereof
FIBO User
I need content
Got what I need
Jenkins
Jenkins Master
Jenkins Slave
For each "push" run the pipeline
Build/Test/Publish Pipeline
RDF Stores or other Vendor products running tests or producing content
No!
All stages of pipeline successful?
Yes!
publish/deploy
https://spec.edmcouncil.org/fibo/
OWL
SMIF/OWL
SMIF/UML
Glossary
SKOS Vocabulary
FIBO.Schema.org
VOWL
Linked Data
Fragments
Site
All versions!

ß