

About OSLC PROMCODE

Committee Specification Draft 01 ~ The PROMCODE specification is intended to provide a common interface to exchange project management (PM) data across organizational boundaries.

A software development project is a collaborative activity to be executed in a fixed time period to produce software systems and/or products. A Project entity represents the information of the whole project such as the name of the project, the descriptions of the project, and progress of the whole project. A Project entity also specifies a Metric entity and the unit of size of ScopeItem entities because they should be unique in a project.

Contents

Vision.....	3
Mission.....	4
Values	4
Plans & Reports.....	5
1. Scope	5
2. Work.....	5
2.1. Artifacts.....	5
2.2. Metrics.....	6
3. Risk.....	6
Administrative Information.....	6

DEMONSTRATION ONLY



Organization for the Advancement of Structured Information Standards (OASIS)

Stakeholder(s):

PROMCODE Consortium :

To address the need of systematic sharing of project management information, the PROMCODE consortium was created in Japan with seven member organizations consisting of Nanzan University and six IT service companies which are IBM, Hitachi, Fujitsu, NEC, NRI, and NTT Data. The consortium worked on defining standard information for management of contracted software delivery and published a specification document in 2013 [PROMCODE13][PROMCODE14].

Nanzan University

IBM

Hitachi

Fujitsu

NEC

NRI

NTT Data

Open Services for Lifecycle Collaboration (OSLC) :

The work was built on the OSLC framework and the technical content was based on the OSLC Core specification at that time. As a center of OSLC activities moved to OASIS, some of the core members of the consortium decided to bring its content to OASIS for global standard creation, and created the OASIS PROMCODE project. The main difference of the OASIS project from the work done by the consortium is an extended scope in OASIS and synchronization with the evolution of the OSLC Core specification [OSLCCore3]. The consortium work focused on minimal set of resources considered at that time. When each company implemented the specification, it was recognized that more resources were needed to reflect the general practice used in the industry. In particular, when two collaborating

organizations do not share the management environment, one organization sends to another organization a status of all relevant project information as a report. This is a common practice and the consortium specification did not address that. The OASIS project decided to address that with addition of further resources needed.

OASIS OSLC Lifecycle Integration for Project Management of Contracted Delivery (OSLC PROMCODE) TC

Tom Kamimura :
Chair | Nanzan University

Mikio Aoyama :
Editor | Nanzan University

Yoshio Horiuchi :
Editor | IBM

Tom Kamimura :
Editor | Nanzan University

Shinji Matsuoka :
Editor | Fujitsu Limited

Shigeaki Matsumoto :
Editor | NEC Corporation

Masaki Wakao :
Editor | IBM

Kazuo Yabuta :
Editor | Nanzan University

Hiroyuki Yoshida :
Editor | Nanzan University

Vision

Software is delivered faster at lower cost and with improved quality

Mission

To provide a common interface to exchange project management (PM) data across organizational boundaries

Values

Software: Global software delivery is commonplace today.

Delivery: With ever increasing pressure on the delivery of software projects for faster delivery, lower cost, and improved quality, it is becoming common for software delivery to be performed with collaboration of multiple organizations.

Speed

Cost-Effectiveness

Quality

Collaboration: Effective collaboration between multiple organizations requires activities to be managed and data to be shared across organizational boundaries.

Diversity: The management of software delivery can be highly challenging due to the diversity of the development processes, methods, tools and platforms used by organizations participating in a project [SPMC]. Management data and management practice are usually unique to each organization. Typically, manual operations are performed in exchanging proprietary management data and in coordinating activities, resulting in inefficient, error-prone and inflexible operations.

Data: As the number of organizations involved increases in a software delivery, the need for more systematic and standards-based data sharing and coordination becomes critical.

Sharing

Standards

Plans & Reports

Determine the scope, tasks, and artifacts required for projects

A plan is a collection, or a snapshot, of managed entities which is agreed on between an acquirer and a supplier at the project initiation and on the timing when a plan is changed. A Plan entity is a ManagedItemCollection entity which is a collection of planned ScopeItem, WorkItem, and Artifact entities together with targeting Measure entities. | A report is a collection, or a snapshot, of managed items which is created by a supplier for project monitoring. A Report entity is a ManagedItemCollection entity which is a collection of ScopeItem, WorkItem, and Artifact entities together with associated Measurement entities and Measure entities. It represents the situation of the project on a specific date.

1. Scope

Manage the scope of developments

A scope item represents a scope of work from the acquirer's view of a software development contract. It represents a unit of value to be accomplished by the software supplier in the contract. For example, it may represent a function required, a use case in which the software will be used, a requirement which the acquirer expects, or a screen which will provide some concrete functions to the user of the software. A scope item is not an activity and therefore, cannot be started nor ended. A ScopeItem entity has the size property that is used to determine the size of a contract between an acquirer and a supplier. This property is not used to track the progress of work. In this sense, a ScopeItem entity is different from an Artifact entity or a WorkItem entity. An acquirer can use a set of ScopeItem entities as managed units to manage a whole scope of development. Both an acquirer and a supplier also use a set of ScopeItem entities to track the size of development. Attributes plannedSize and actualSize of a ScopeItem are used for the purpose. There should be agreement between an acquirer and a supplier on what kind of ScopeItem entities should be used and on how large each ScopeItem entity is. Change of a ScopeItem entity or its estimated size needs a change of the agreement. A ScopeItem entity can be decomposed into finer grained ScopeItem entities to be used in detailed management. In that case, a coarse grained ScopeItem entity may be used to aggregate a set of finer grained ScopeItem entities.

2. Work

Describe the activities to be performed

A work item describes an activity to be performed in a software development contract. It adds details to the description of the work that is described by a scope item. These details typically include cost, schedule, and resource requirements. The set of all work items in a project forms a work breakdown structure. A WorkItem entity represents the supplier's internal activity. For example, it may represent a development phase such as analysis, design, implementation, or test. It may also represent a finer grained activity such as document writing, reviewing, or coding. A WorkItem entity is a managed unit of activity required to implement a ScopeItem entity or to produce an Artifact entity. Progress of a WorkItem entity is managed by comparing planned and actual dates on which it is started and ended. A WorkItem entity can be decomposed into finer grained WorkItem entities to be used in detailed management. A coarse grained WorkItem entity is used to aggregate a set of fine grained WorkItem entities.

2.1. Artifacts

Describe the work products

An artifact is a work product that is produced in a project such as design documents, source code, test report, and so on. An artifact may be physical or digital. An artifact is produced for a work item or a scope item. An Artifact

entity can be measured using Measure entities, and their measured values may vary at each point of time on a project. The quality of an Artifact entity is managed by comparing targeted and actual Measure entities. An Artifact entity can be decomposed into finer grained Artifact entities to be used in detailed management. A coarse grained Artifact entity may be used to aggregate a set of finer grained Artifact entities.

2.2. Metrics

Measure the artifacts

A measure is an observation of measurable attribute of an artifact. A Measure entity has a metric such as "number of lines of codes (LOC)", "number of found bugs", and so on. It also has a unit which defines the unit of the numbered value, for example, "1" means one line or one kilo lines. A Measure entity must be either targeted by one Artifact entity or observed by one Measurement entity.

3. Risk

Control risks

A risk is a potential problem that must be controlled before it occurs in order to meet the objectives of a project. Failure to control the risk may result in a situation that can negatively impact the project, such as a schedule delay and quality problems. If a risk actually causes the situation to occur, it typically becomes an issue.

Administrative Information

Start Date: 2021-04-28

End Date:

Publication Date: 2021-07-24

Source: <https://docs.oasis-open.org/oslc-promcode/promcode/v1.0/csd01/promcode-spec.html#background-and-motivation>

Submitter:

Given Name: Owen

Surname: Ambur

Email: Owen.Ambur@verizon.net

Phone: