



A metamodel integration for metrics and processes correlation

Xabier Larrucea, Eider Iturbe

European Software Institute - Tecnalia



Current situation

Nowadays organizations need to **improve** their **efficiency** due to an increase of **global competency** in their markets. In fact they are starting up some **improvement activities** related to **quality models**

Current situation

- **One of the eight elements of software quality management (Humphrey, 2008)**

establish and maintain statistical control of the software engineering process

Improve quality

METRICS

Current situation

- **Software and business processes are intertwined in many organizations but all of them are considered the organizations' engines. In this context metrics are the basis for a statistical process control and they are used to measure organization's behavior and performance (Florac et al., 1999)**

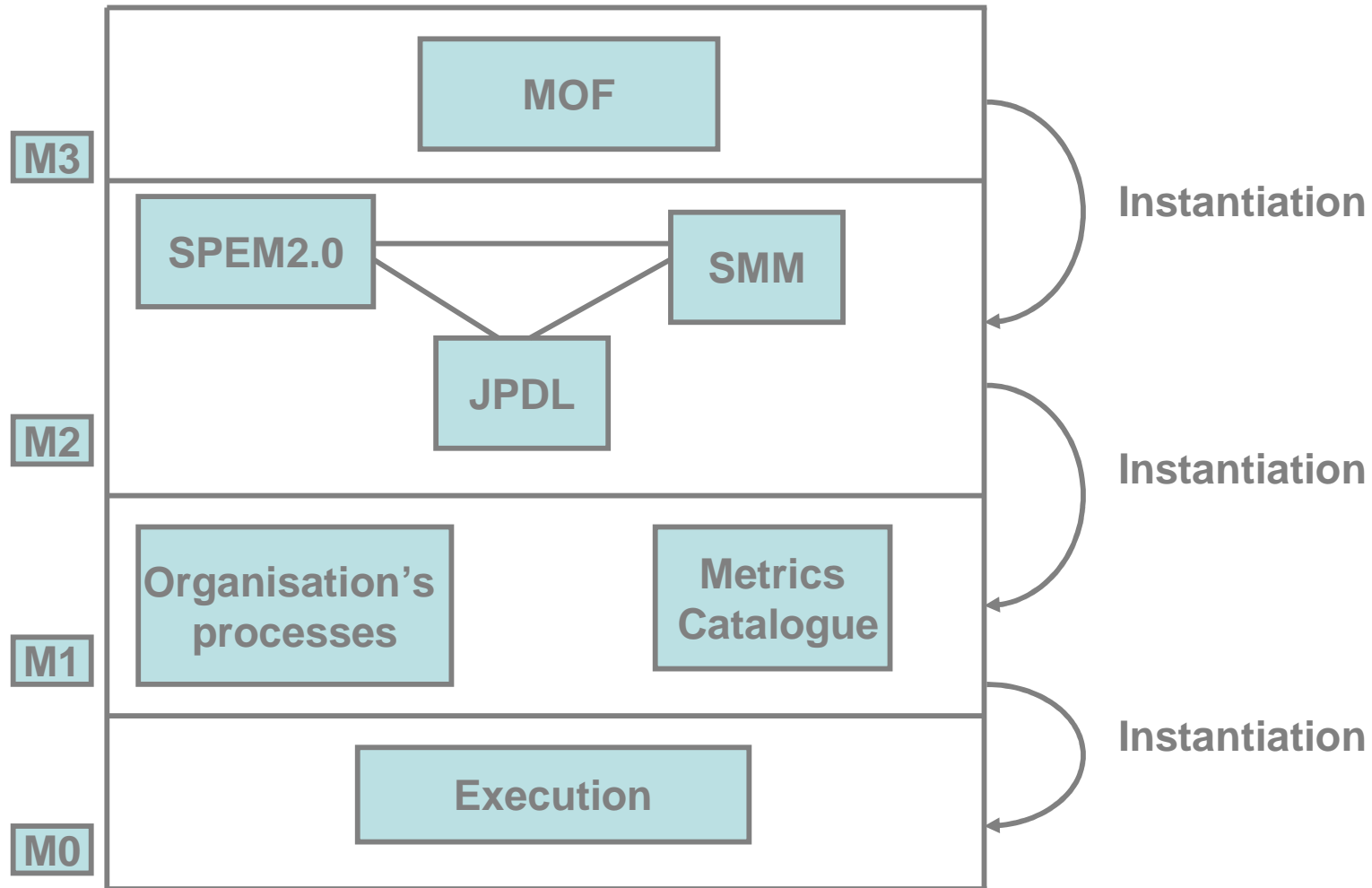
Current situation

- Most of **business process metrics** are used to calculate the **complexity, coupling, cohesion, size and modularity** of business processes based on the elements used such as in (Mendling, 2009), (Vanderfeesten, 2007) and (Cardoso, 2007). Therefore business process metrics have been studied in for years and several metrics models has been applied to analyze their complexity.

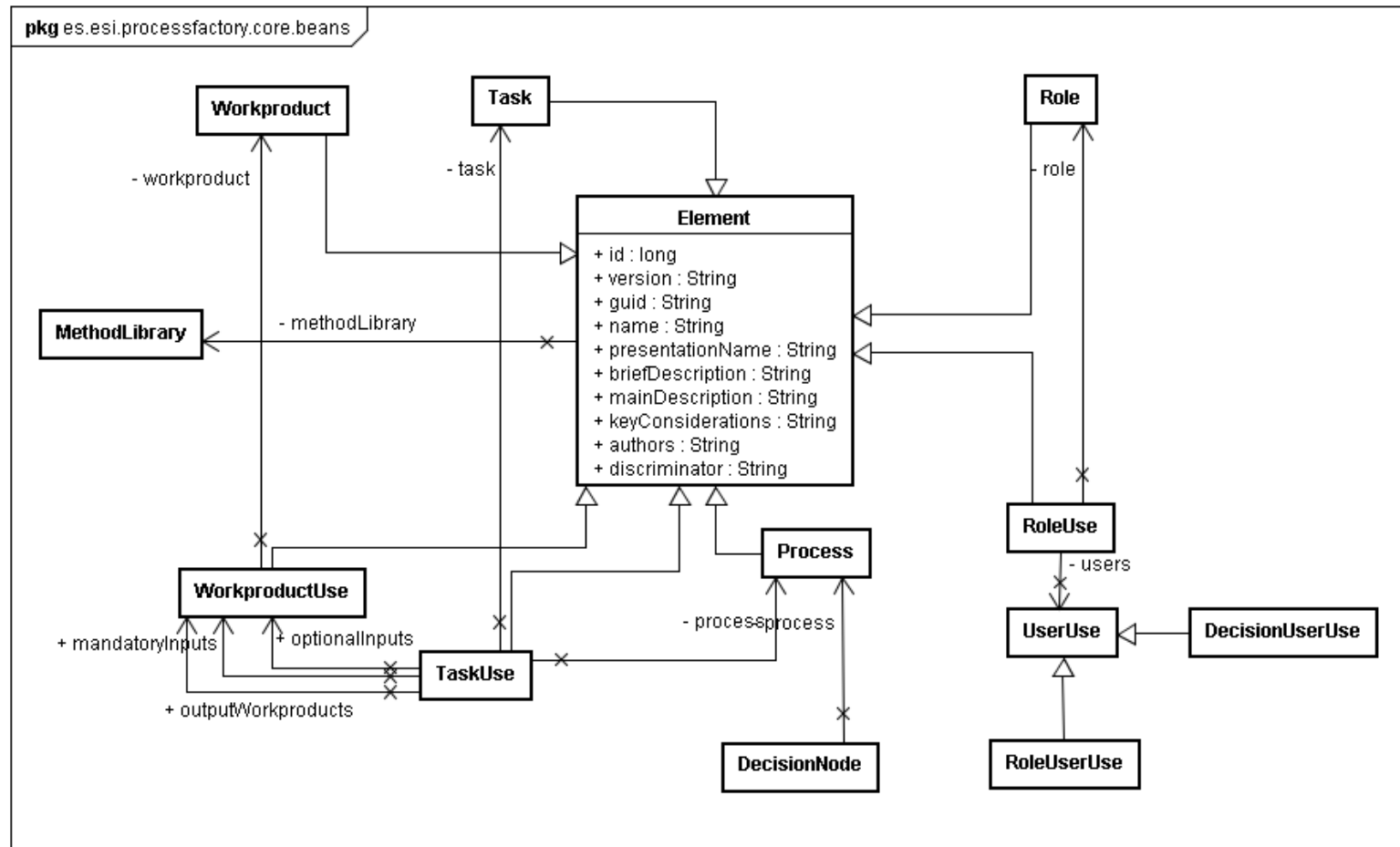
Current situation

- In the context of **metamodelling** a huge effort has been invested to develop **metamodels** for a wide range of domains. In terms of metamodelling our approach is compliant to the **four architectural metalayers** promoted by the Object Management Group (OMG).

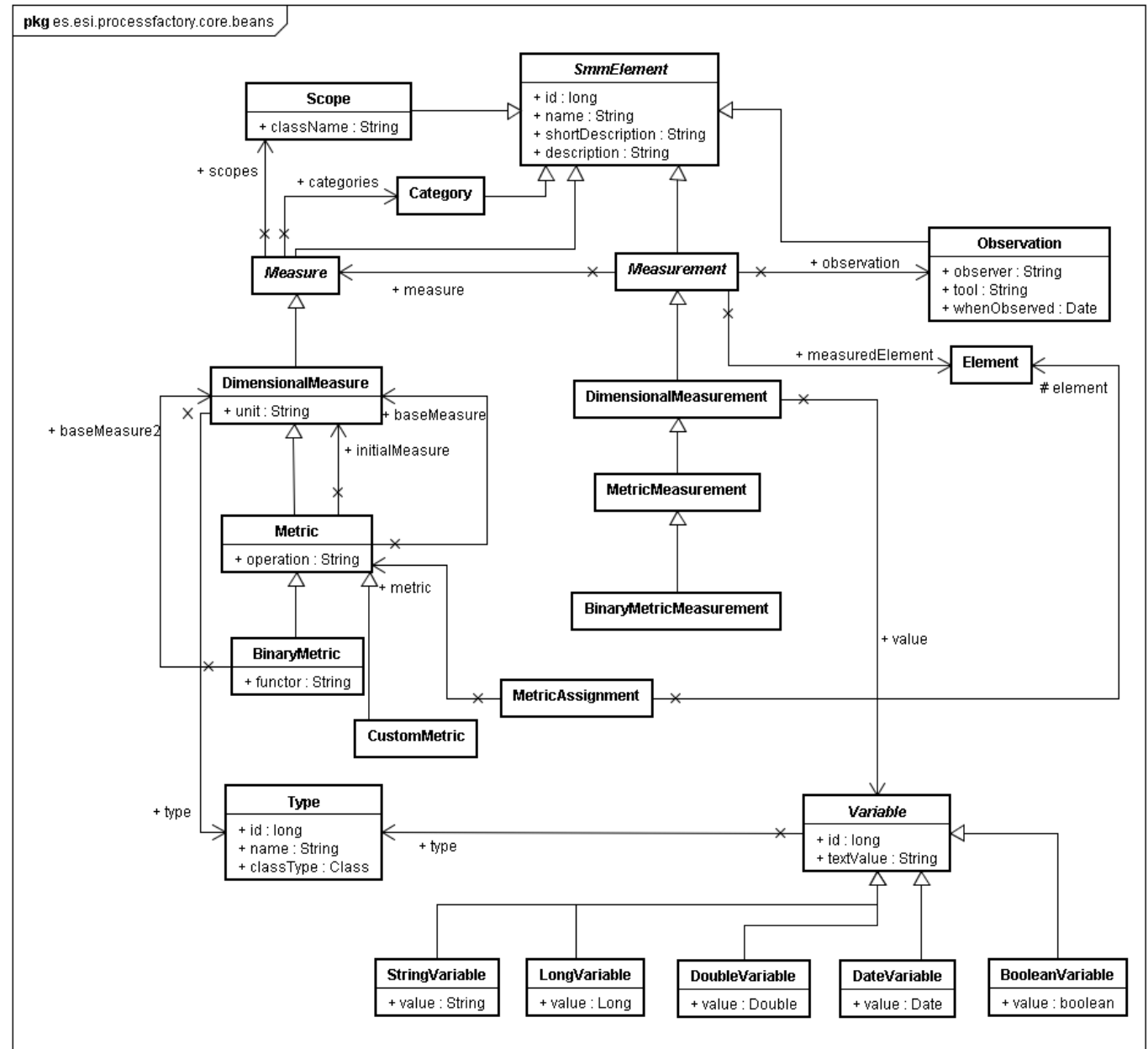
Process and Metrics metamodels



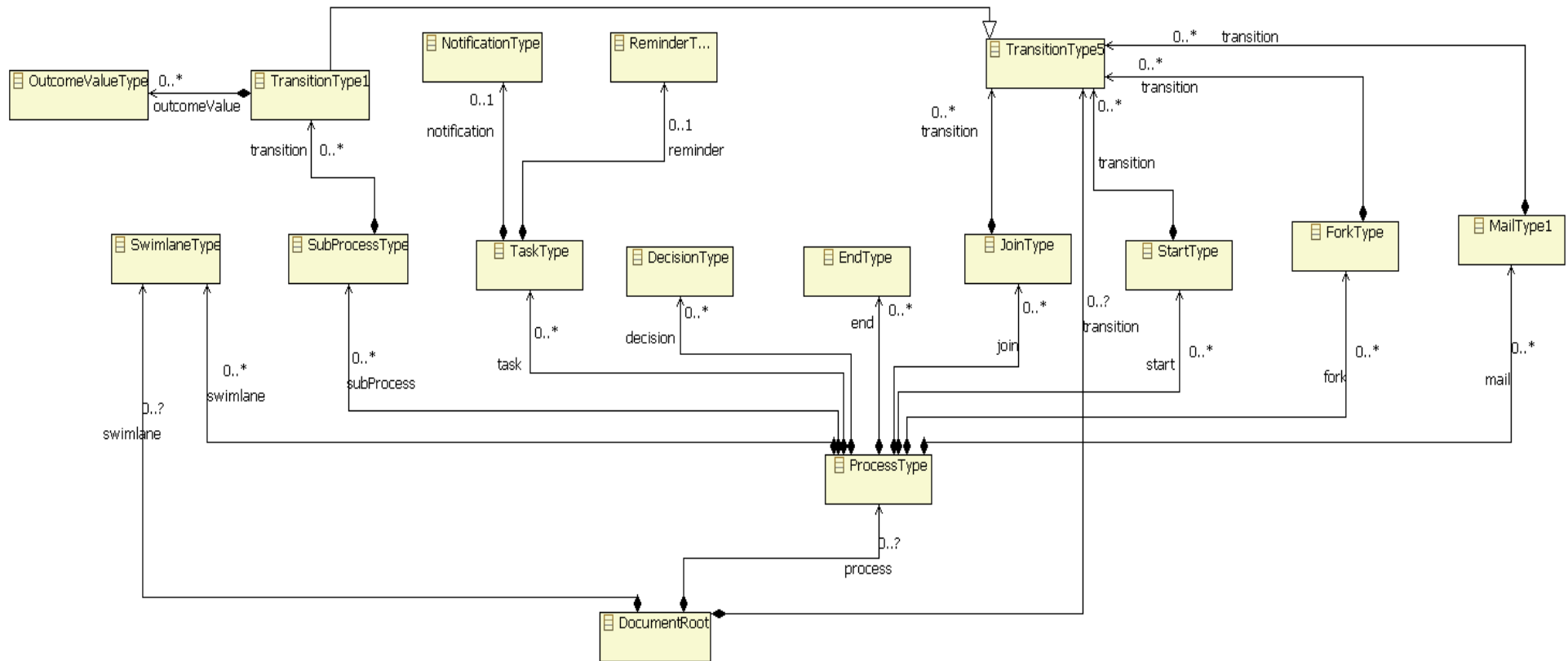
Software Process Engineering Metamodel 2.0



Software Metrics Metamodel



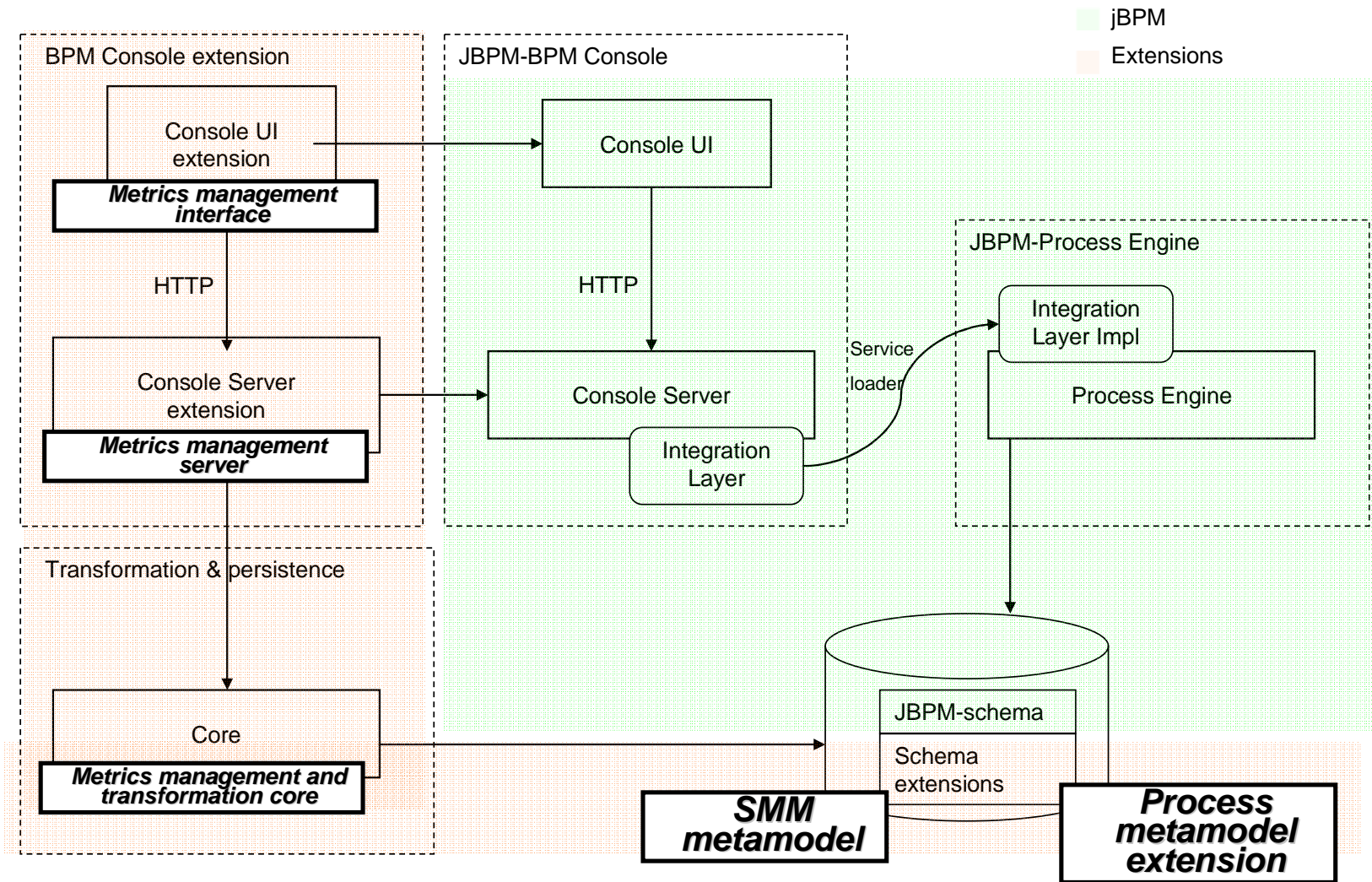
JPDL metamodel



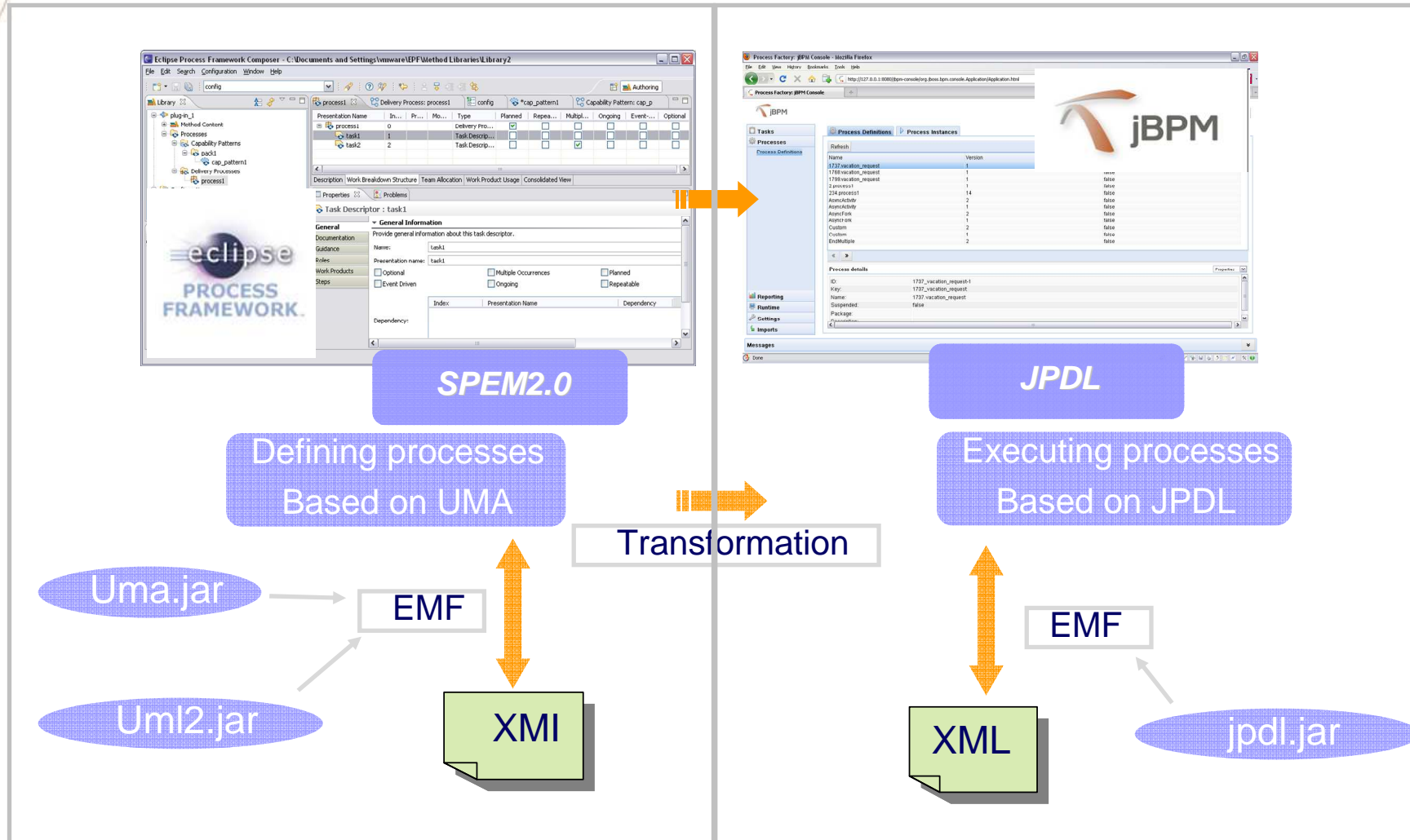
SPEM2.0 to JPDL transformation

| SPEM 2.0 | | | | JPDL |
|-------------------------|--|----------------|-----------------|---|
| Process Behavior: UML | Process with Methods = Process Structure | Method Content | Managed Content | |
| Initial node | --- | --- | --- | Start activity |
| Activity Parameter Node | Task descriptor | Task | --- | Task activity (+ task form) |
| Activity Final Node | --- | --- | --- | End activity |
| Decision node | --- | --- | --- | Decision activity |
| Merge node | --- | --- | --- | Join activity (multiplicity 1) |
| Fork node | --- | --- | --- | Fork activity |
| Join node | --- | --- | --- | Join activity |
| Control Flow | --- | --- | --- | Transition element (in every activity node) |
| --- | Role descriptor | Role | --- | User |
| --- | Workproduct descriptor | Workproduct | --- | Process/Task variable: read-write |
| --- | --- | --- | Guidance | Process/Task variable: only read |

Process Factory architecture



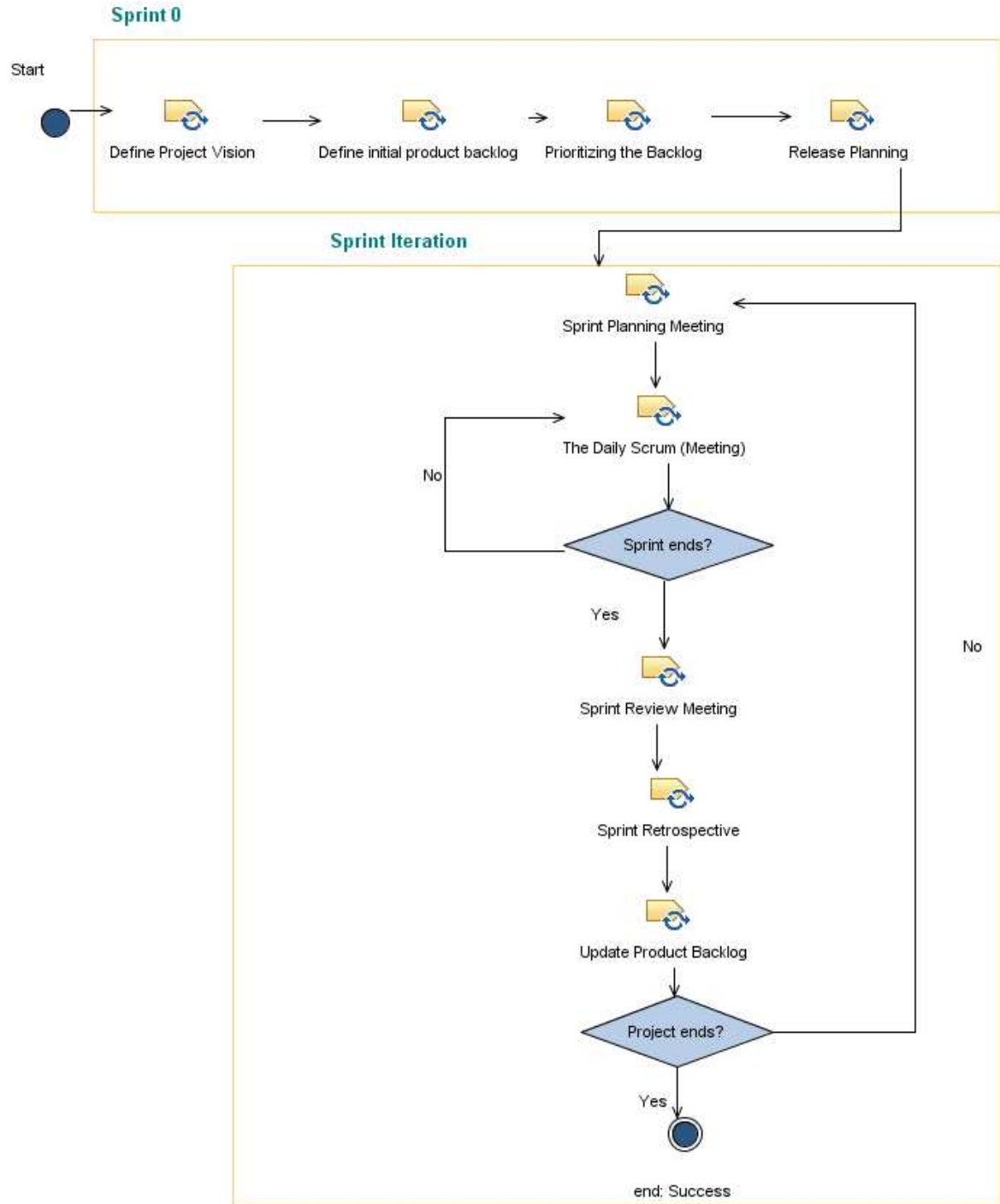
Process Factory approach



Case Study: a SCRUM process

- **Scrum** is an agile process or framework for managing agile projects
- there are **three main roles** involved in a scrum project: Product owner, Sprint Master and Sprint team
- The **work** to be done on a Scrum project is listed in the **Product Backlog**, which is a **list of functional and non-functional requirements** including **estimation of efforts** in order to turn each into deliverable product increments

A Scrum Process



Instantiation process

- **Process Factory provides a user interface where processes' stakeholders can import and deploy defined processes using EPF**
- **We have to set some process features: user-roles, decision makers...**

Instantiation process

- **User assignment to roles**
- **User assignment to decision node's responsible**
- **Metric assignment to process or task**
 - **Productivity metrics**
 - **Sprint Cycle time, Sprint Completion, Velocity, Source Lines of Code (SLOC)**
 - **Quality Metrics**
 - **Technical debt points, Running automated tests, Post sprint defect arrival, Post release defect arrival.**
 - **Predictability Metrics**
 - **Cost per story point.**
 - **Hours per story point. It measures the average number of hours estimated per story point**

Instantiation process

- **User assignment to roles**
- **User assignment to decision node's responsible**
- **Metric assignment to process or task**
 - **Productivity metrics**
 - **Sprint Cycle time.** It measures the turnover of stories throughout the sprint as measured by when a story was started vs. when it was completed.
 - **Sprint Completion.** It measures work completed over a sprint (done, not tested, not coded, waste).
 - **Velocity.** It measures number of points completed per sprint.
 - **Source Lines of Code (SLOC).** It measures the number of lines of code to determine productivity. A higher number does not necessarily translate to higher productivity - only more code.
 - **Quality Metrics**
 - **Technical debt points.** It measures the volume and throughput of technical debt to determine the quality evolution of a product.
 - **Running automated tests.** It measures unit and functional automated tests that are passing each sprint.
 - **Post sprint defect arrival.** It measures the number of defects that are found after the sprint they were initially developed.
 - **Post release defect arrival.** It measures the number of defects found after release to customers.
 - **Predictability Metrics**
 - **Cost per story point.** It measures euros per story point by knowing the number of people of team, the sprint length and the story point velocity to determine the average cost per story point.
 - **Hours per story point.** It measures the average number of hours estimated per story point

User assignment

Process Factory - Mozilla Firefox

File Edit View History Delicious Bookmarks Tools Help

Process Factory

Process Factory

Import Process :: 3. Task Configuration

Scrum process

| Task | Description | Duration |
|----------------------------------|---|----------|
| 1 Assemble Team | Once all the roles on the team are identified and filled the team should be brought together for a team kick off meeting, which should | |
| 2 Define initial product backlog | As a team enters the first Sprint it should have enough Product Backlog items listed and prioritised to enable it to begin useful work. | |
| 3 Define Project Vision | It is important that the whole team understands the essence of what the project or product is trying to achieve. This is where the Pr | |
| 4 Prioritizing the Backlog | The items in the Product Backlog are prioritized by the Product Owner prior to release and sprint planning. | |
| 5 Release Planning | At the beginning of a project the team will create a high-level release plan | |
| 6 Sprint Planning Meeting | The Sprint Planning Meeting is where the Scrum Team and Product owner determine which features and tasks will be attempted in th | |
| 7 Sprint Retrospective | Following the Sprint Review, the team gets together for the Sprint Retrospective. | |
| 8 Sprint Review Meeting | Each sprint concludes with a Sprint Review where the team demonstrates a potentially shippable product increment. | |
| 9 The Daily Scrum (Meeting) | The Daily Scrum is a quick meeting comprised of all members of the Scrum Team and the Scrum Master. | |
| 10 Update Product Backlog | Periodically the Scrum Team will estimate the size of each item on the Product Backlog. | |

Task Details | **User Configuration** | Schedule | WorkProducts

| Role | Description | Primary | User |
|-----------------|---|-------------------------------------|-------------|
| 1 ScrumMaster | The ScrumMaster is responsible for making sure a Scrum team lives by the values and practices of Scrum. The Scrum | <input checked="" type="checkbox"/> | mary |
| 2 Scrum Team | The Scrum Team builds the product that the customer is going to consume: the software or website, for example. The | <input checked="" type="checkbox"/> | mike, peter |
| 3 Product Owner | The Product Owner represents the interests of all stakeholders, defines the features of the product and prioritizes the | <input checked="" type="checkbox"/> | alex |

Metrics

- We can manage **metrics assigned to processes and to process tasks**. In fact we can:
 - **initialize metrics** with estimation values,
 - **monitor metrics** whose value is automatically calculated by the application as the process instance is executing, and
 - **insert metrics' values** for those metrics defined by the user

Metrics

The screenshot shows a web browser window titled "Process Factory - Mozilla Firefox". The browser's address bar shows "Process Factory". The page content is titled "Process Factory" and "Import Process :: 5. Metric Configuration".

On the left side, there is a navigation menu with the following items:

- Tasks
- Processes
- Reporting
- Runtime
- Settings
- Imports
 - Imported List
 - New Process
 - Configure Metrics

The main content area is titled "Scrum process" and "Add Custom Metric". It contains the following form fields:

- Scope: A dropdown menu with options: Process, Task, Workproduct, Role/User. "Task" is selected.
- Name: A text input field containing "Number of stories done".
- Unit: A text input field containing "story".
- Short description: A text area containing "It measures the number of stories completed over a sprint."
- Description: An empty text area.
- Type: A dropdown menu with "Long" selected.
- Initial estimation required

Processes and Metrics monitoring

Process Factory - Mozilla Firefox

File Edit View History Delicious Bookmarks Tools Help

Process Factory

Process Factory

Tasks

Processes

Process Definitions

Reporting

Runtime

Process Definitions | **Process Instances**

Refresh Start Terminate Delete

| Instance ID | Status |
|---------------------------|--------|
| Scrum_process_2115.410021 | RU |
| Scrum_process_2115.450001 | RU |

Instance details

| | |
|--------------|--------------------------------|
| Process: | Scrum process.2115 |
| Instance ID: | Scrum_process_2115.450001 |
| Key: | |
| State: | RUNNING |
| Start Date: | 2010-03-08 16:57:35 |
| Activity: | Define initial product backlog |

Process Instance Activity

Instance: Scrum_process_2115.450001

```

graph TD
    Start((Start)) --> D1[Define Project Vision]
    D1 --> D2[Define initial product backlog]
    D2 --> D3[Prioritizing the Backlog]
    D3 --> D4[Assemble Team]
    D4 --> D5[Release Plan]
    
    subgraph Sprint_0 [Sprint 0]
        D1
        D2
        D3
        D4
        D5
    end
    
    D5 --> SPM[Sprint Planning Meeting]
    
    subgraph Sprint_Iteration [Sprint Iteration]
        SPM --> TDSM[The Daily Scrum Meeting]
        TDSM --> SE{Sprint ends?}
        SE -- No --> SPM
        SE -- Yes --> End[ ]
    end
    
```


Processes and Metrics monitoring

Process Factory - Mozilla Firefox

File Edit View History Delicious Bookmarks Tools Help

Process Factory

Process Factory

Process Instance Metrics

Process Elements

- Process
 - Scrum process
- Tasks
 - Prioritizing the Backlog (1)
 - Sprint Review Meeting (2)
 - Define Project Vision (1)
 - The Daily Scrum (Meeting) (20)
 - Update Product Backlog (2)
 - Sprint Planning Meeting (2)
 - Define initial product backlog (1)
 - Release Planning (1)
 - Assemble Team (1)
 - Sprint Retrospective (2)

Sprint Review Meeting

Brief Description
 Each sprint concludes with a Sprint Review where the team demonstrates a potentially shippable product increment.

Metric List

| Metric | Metric Short Description | Type | Unit | Initial Estimation | Value |
|--------------------------------------|---|---------------|------------|--------------------|----------------------|
| Instance ID: 480030 (8 Items) | | | | | |
| Duration | It measures duration of an activity | Predetermined | hour | 4 | 0.046666666666666666 |
| Number of stories done | It measures the number of stories done over a sprint. | User defined | story | 5 | 4 |
| Number of stories Not Coded | It measures the number of not coded stories over a sprint. | User defined | story | | |
| Number of stories Not tested | It measures the number of not tested stories over a sprint. | User defined | story | | 1 |
| Post sprint defect arrival | It measures the number of defects that are found after a sprint. | User defined | defect | | |
| Running automated tests | It measures unit and functional automated tests. | User defined | test | | 15 |
| Source Lines of Code | It measures the number of lines of code to determine the size of a project. | User defined | sloc | | |
| Technical debt points | It measures the volume and throughput of technical debt. | User defined | debt point | | 1 |
| Instance ID: 480159 (8 Items) | | | | | |
| Duration | It measures duration of an activity | Predetermined | hour | 4 | 0.01 |

Conclusions & further work

- **Process Factory architecture is built up from open source tools such as EPF and jBPM.** We have extended both environments in order to allow interoperability between them, to define and monitor metrics during process execution. At the end this architecture provides a way to help a **software quality management** because it provides:
- **Process definition and monitoring:** most of quality systems require a definition of their organizational' set of processes. It provides not only definition capabilities but also monitoring capabilities using an open process engine such as jBPM.

Conclusions & further work

- **Process metrics definition and monitoring:** we have developed some extensions to provide metric management utilities.
- **Metrics definition and monitoring is managed from the Internet and it is planned to be provided as a software as a service.**
- **The case study is used as a validation of our approach. In fact we are applying this approach to agile developments with a limited scope.**



Xabier Larrucea, Ph.D, PMP
R&D Project Leader
xabier.larrucea@esi.es

Parque Tecnológico, # 204
E-48170 Zamudio
Bizkaia (Spain)
Tel.: +34 94 420 95 19
Fax: +34 94 420 94 20
www.esi.es