Specification and requirements document for the LEAP web-based application.

***Context***

The purpose of this document is to describe the requirements and specification of the LEAP web application. LEAP stands for Lean Enterprise Architecture Platform and is to be used to create models and maintain information in the field of Enterprise architecture.

The main idea is to provide a web environment where analysts/architects/managers can enter and maintain information on the structure or architecture of their organization. **For more information on the context of the application, please refer to the video of the kickoff session. Please check the video, because it will provide you with the right info.** Briefly stated, is it important to have a clear overview of the ‘structure’ of your organization as, for example, a manager. However, currently companies need to draw these structures themselves in Powerpoint, Visio and so on. The problem with that is that the model changes and this means a lot of work to update the Powerpoint drawing etc. everytime. Therefore, we need an application where you can enter information and the application draws the model. In an ideal situation, you can export these models to a powerpoint format or PDF.

The specific components that are needed in the application are explained below. The idea is that an organization can create a sort of repository where you can store information on these components and link them to each other to understand and show relations between the components. The second part of the application is then generating models, based on the information on the different components that we have defined/stored.

The relevant ‘components’ for a user are the following. Please refer to the use cases for more information on the requirements.

* **Environments** (a user can create 1 or more environments. This is mainly the ‘company’ for which the user will create the other objects).
* **Capabilities**. A capability is a ‘building block of an organization’. All capabilities in an organization show all the things that organization needs to be able to do, but on a high level (read: low level of detail). We call this the ‘what’ view because it shows us what an organization needs to do. We use this to provide a clear overview of all things the company needs to think of. They relate sometimes to concepts such as ‘departments’, but they help us to look at the organization from a functional point of view. Below you can find an example of the level 1 capabilities of a professional sports club, such as a football club.

Text

Description automatically generated

Each capability is linked to 1 and only 1 environment. A capability can exist on level 1, 2 or 3 (always on 1 level). This means that a capability can have 0 to 1 parent and 0 to n children. (In other words: a capability can only belong to 1 parent capability). Example of that same football club:

Diagram, text

Description automatically generated

In this example, the capability *4.0 Product delivery and material management* is on level 1. *Supplier management, procurement, inbound logistics & stock* and others are on level 2. They all have *4.0 Product delivery and material management* as a parent capability. Procurement, on level 2, has also 3 child capabilities: *manage tenders, manage purchase orders and manage framework agreements*. You can see that these 3 capabilities have only one parent, being *procurement*. You can ignore the silver color for now.

*[General remark: the icon (tetris blocks) on the top right are not needed in LEAP/the application you will build. They are there, because these examples were drawn with a program called Signavio, which uses the Archimate language specification).*

* **Strategy**. We want to allow to define strategies in LEAP. Each strategy is linked to 1 and only 1 environment.
* **Strategy items**. A strategy can be further split up in strategy items and these items then can be linked to capabilities. More on this below in the use cases. Each strategy is linked to 1 strategy and to 0 or more capabilities.
* **ITapplication**. We want to be able to list all applications that a company uses and we want to know what capabilities use an application. We do this, so we can estimate the importance of an application and analyse how well the application support business in executing its processes. Furthermore, there will be certain properties that we can store on the application level and on the relationship between an application and a capability.

Each IT application can be linked to 0 to many capabilities and a capability can also be linked to 0 or many IT applications (read: is supported by 0 to many IT applications).

* **Resource**. There will be 2 types of resources (human and other). Each resource can be linked to 0 to many capabilities and a capability can be linked to 0 to many resources. Again, some properties will be stored on the resource level and on the relation with the capability.
* **Business processes**. Each business process can be linked to 0 to many capabilities and a capability can be linked to 0 to many business processes.
* **Program and project**. Programs can be defined, as well as projects. The idea here is that a program can exist of multiple projects (so project is a lower level object). A project can then be linked with capabilities (in other words, we want to define the following: what capabilities is a project impacting?). The program will be a property of the project: in other words, you can assign a project to a program (if you want to).

Another important part of the application is the report and export part. The main purpose is to create an application which allows to easily generate reports and, if possible, exports like slideshows (e.g. powerpoint). This is important, because this should be a business-oriented application and this type of users often use dashboards, powerpoints and spreadsheets to communicate.

For examples, we refer again to the video of kickoff session, where you can see for example how a specific parameter is displayed on the capability map (in that example, we showed the strategic importance of a capability in relation to a specific *strategy item*).

**Environment**

|  |  |  |
| --- | --- | --- |
| **Property name** | **Property type** | **Example value** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Strategy**

|  |  |  |
| --- | --- | --- |
| **Property name** | **Property type** | **Example value** |
| Name |  |  |
| Timeframe from | Date |  |
| Timeframe to | Date |  |
|  |  |  |

**StrategyItems**

|  |  |  |
| --- | --- | --- |
| **Property name** | **Property type** | **Example value** |
| Strategy ID |  |  |
| Name |  |  |
| Description |  |  |
|  |  |  |

**Capability**

|  |  |  |
| --- | --- | --- |
| **Property name** | **Property type** | **Example value** |
| Name | Text |  |
| Parent | ID of other capability |  |
| Level | Calculated field | 1, 2 or 3 |
| Pace of change | List value | Standard, differentiation or innovative |
| TOM | List value | Coordination, unification, diversification, replication |
| Resources quality | Number | 1-5 (1 = poor, 3 = fair, 5 = good) |
| Information quality | Calculated field | \*Explanation below. |
| Application fit | Calculated field | \*\*Explanation below. |
|  |  |  |

\*For each CapITApp, linked to this capability, take the average score of the 3 information quality properties and multiply this with the importance factor. Example: if there are 2 applications linked to this capability, the calculation would be for example: AVERAGE (3,5,1)\*60% + AVERAGE(1,1,4)\*40% = 3\*0,6+2\*0,4 = 1,8+0,8 = 2,6.  
The max should always be 5, the values above are illustrative.

\*\*For each CapITApp, linked to this capability, take the average score of the 4 business fit properties and multiply this with the importance factor. Example: if there are 2 applications linked to this capability, the calculation would be for example: AVERAGE (2,4,4,2)\*70% + AVERAGE(1,3,3,1)\*30% = 3\*0,7+2\*0,3 = 2,1+0,6 = 2,7.  
The max should always be 5, the values above are illustrative.

**CapStratItems** 0)

|  |  |  |
| --- | --- | --- |
| **Property name** | **Property type** | **Example value** |
| Capability ID | ID of Capability |  |
| Strategy Item ID | ID of strat item |  |
| Strategic emphasis | List value | Low, Medium, High |
|  |  |  |

**Information**

|  |  |  |
| --- | --- | --- |
| **Property Name** | **Property type** | **Example value** |
| Name |  |  |
| Description |  |  |
|  |  |  |
|  |  |  |

**CapInfo** (Relationship object between Information and Capability. The reason here is that we want to document what information objects are supporting what capabilities. Real life meaning: in a capability, you use information to support this. We want to be able to capture this.)

|  |  |  |
| --- | --- | --- |
| **Property name** | **Property type** | **Example value** |
| Capability ID |  |  |
| Information ID |  |  |
| Criticality |  | Low, medium, high |

**Resource**

|  |  |  |
| --- | --- | --- |
| **Property name** | **Property type** | **Example value** |
| Name |  |  |
| Description |  |  |
| FTE yearly value |  |  |
|  |  |  |

**CapResource** (Relationship object between Resource and Capability, because a Resource can be linked to many capabilities. The reason here is that we want to document what Resource are supporting what capabilities. Real life meaning: in a capability, you use one or more resources to support this. We want to be able to capture this.)

|  |  |  |
| --- | --- | --- |
| **Property name** | **Property type** | **Example value** |
| Capability ID |  |  |
| resource ID |  |  |
| Number of resources | Number | 5 |
|  |  |  |

**BusinessProcess**

|  |  |  |
| --- | --- | --- |
| **Property name** | **Property type** | **Example value** |
| Name |  |  |
| Description |  |  |
|  |  |  |
|  |  |  |

**CapBusProcess** (relationship object between BusinessProcess and Capability, because a capability can be linked to many business process objects and a business process object can be linked to many capabilities)

|  |  |  |
| --- | --- | --- |
| **Property name** | **Property type** | **Example value** |
| Capability ID |  |  |
| BusinessProcess ID |  |  |
|  |  |  |

**ITApplication**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property name** | **Property type** | **Quality type** | **Example value** |
| Application name |  | N/A |  |
| Technology |  | N/A |  |
| Version |  | N/A |  |
| Acquisition date | Date | N/A |  |
| End of life | Date | N/A | Undefined is allowed |
| Current scalability | Value | Technical quality | 1,2,3,4 or 5 (1 = poor, 3 = fair, 5 = good) |
| Expected scalability | Value | Technical quality | 1,2,3,4 or 5 (1 = poor, 3 = fair, 5 = good) |
| Current performance | Value | Technical quality | 1,2,3,4 or 5 (1 = poor, 3 = fair, 5 = good) |
| Expected performance | Value | Technical quality | 1,2,3,4 or 5 (1 = poor, 3 = fair, 5 = good) |
| Current security level | Value | Technical quality | 1,2,3,4 or 5 (1 = poor, 3 = fair, 5 = good) |
| Expected security level | Value | Technical quality | 1,2,3,4 or 5 (1 = poor, 3 = fair, 5 = good) 1,2,3,4 or 5 (1 = poor, 3 = fair, 5 = good) |
| Current stability | Value | Technical quality | 1,2,3,4 or 5 (1 = poor, 3 = fair, 5 = good) |
| Expected stability | Value | Technical quality | 1,2,3,4 or 5 (1 = poor, 3 = fair, 5 = good) |
| Cost currency | Text | N/A | Free field |
| Current value for money | Value | Cost | 1,2,3,4 or 5 (1 = poor value, 3 = fair, 5 = good value) |
| Current total cost per year | Number | Cost |  |
| Tolerated total cost per year | Number | Cost |  |
| TIME value | List value | N/A | Tolerate, invest, eliminate, migrate |

**CapItApp** (Relationship object between ITApplication and Capability, because an IT application can be linked to many capabilities and a capability can be linked to many it applications. The reason here is that we want to document what IT applications are supporting what capabilities. Real life meaning: in a capability, you use one or more it applications to support this. We want to be able to capture this and the quality of that relation)

|  |  |  |  |
| --- | --- | --- | --- |
| **Property name** | **Property type** | **Quality type** | **Example value** |
| Importance | % | N/A | (Remark: sum of all weight values of all CapItApp objects, linked to a specific capability should be 100%) |
| Efficiency support |  | Business fit | 1,2,3,4 or 5 (1 = poor, 3 = fair, 5 = good) |
| Functional coverage |  | Business fit | 1,2,3,4 or 5 |
| Correctness |  | Business fit | 1,2,3,4 or 5 |
| Future potential |  | Business fit | 1,2,3,4 or 5 |
| Completeness |  | Information quality | 1,2,3,4 or 5 |
| Correctness |  | Information quality | 1,2,3,4 or 5 |
| Availability |  | Information quality | 1,2,3,4 or 5 |