

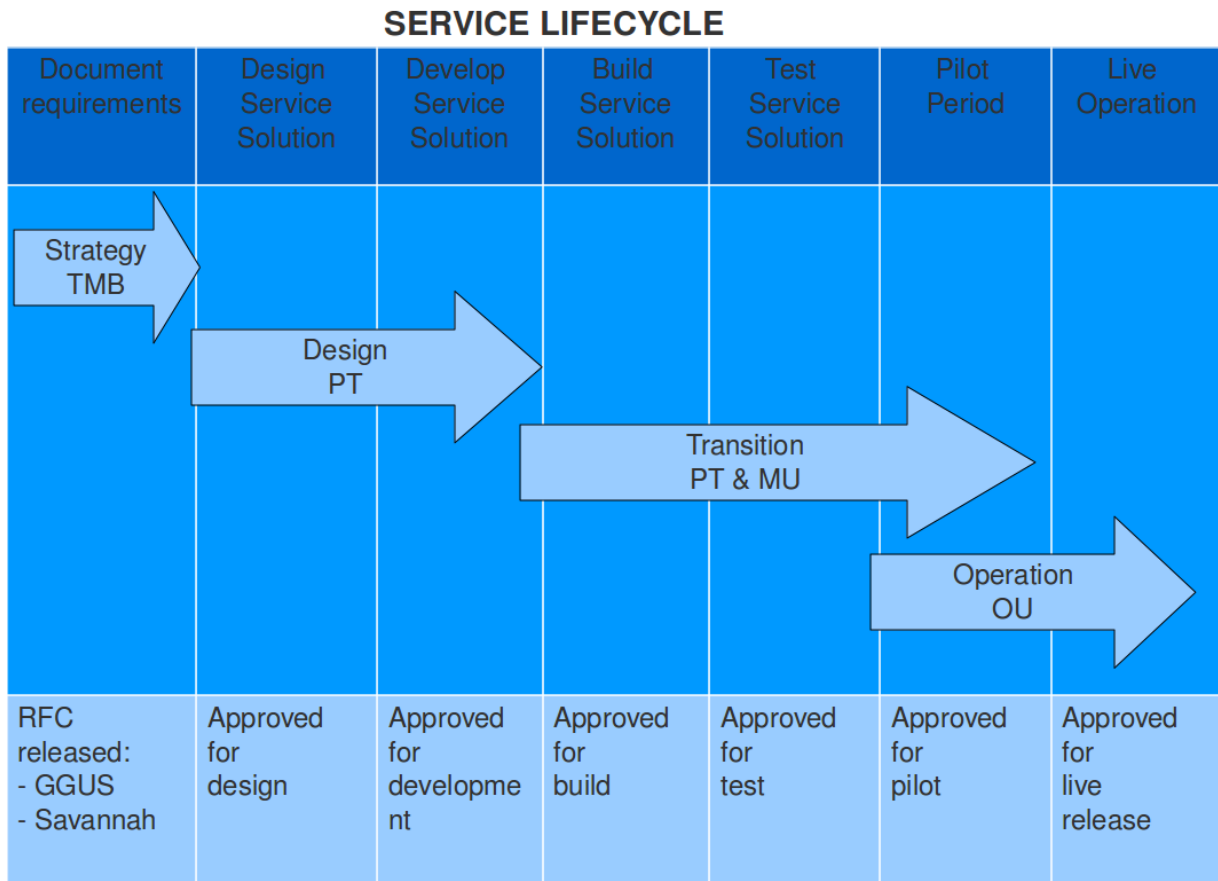
ITIL Concepts for O-E-9 MIDDLEWARE ROLLOUT PROCESS

Document complementary to “Ibergrid O-E-9 notes and issues”

Version 0.1 Javier López Cacheiro 1 March 2010

Introduction

Based on ITIL best practices the overall EGI service lifecycle can be represented as follows:



In this document we will focus in the service transition stage describing the processes and functions involved there.

SERVICE TRANSITION

This stage starts when the approval for building has been given and it is a coordinated effort between the Product Team responsible for the given component and the Middleware Rollout Team.

The purpose is to move a middleware service from design into operation and hand it out to the Operations Team.

Operational Level Agreements (OLA) should be established both between each Product Team and the Middleware Rollout Team and the Operations Teams. **The terms of these agreements should be discussed.**

According to ITIL there are three processes we should consider:

- Service Asset and Configuration Management
- Change Management
- Release and Deployment Management

In the following sections we will explore the main concepts inside each of these processes. We should understand them and define how to use them in the EGI middleware rollout framework.

Service Asset and Configuration Management

Some important definitions:

Configuration Management Database (CMDB): A database used to store Configuration Records throughout their Lifecycle. The Configuration Management System maintains one or more CMDBs, and each CMDB stores Attributes of CIs, and Relationships with other CIs.

Configuration Management System (CMS) : A set of tools and databases that are used to manage an IT Service Provider's Configuration data. The CMS also includes information about Incidents, Problems, Known Errors, Changes and Releases; and may contain data about employees, Suppliers, locations, Business Units, Customers and Users. The CMS includes tools for collecting, storing, managing, updating, and presenting data about all Configuration Items and their Relationships. The CMS is maintained by Configuration Management and is used by all IT Service Management Processes.

Definitive Media Library (DML) : One or more locations in which the definitive and approved versions of all software Configuration Items are securely stored. The DML may also contain associated CIs such as licenses and documentation. The DML is a single logical storage area even if there are multiple locations. All software in the DML is under the control of Change and Release Management and is recorded in the Configuration Management System. Only software from the DML is acceptable for use in a Release.

We should identify the tools that will provide the CMDB and DML functionality: ETHICS, GOCDB, LDAP, etc. It is very important to establish the relationships between all the different middleware components (dependencies with other components, product team responsible, etc.) We must identify a tool to maintain this information (wiki?), we must provide a template and each product team should fill it.

Change Management

Some definitions from ITIL:

Change Management: The Process responsible for controlling the Lifecycle of all Changes. The primary objective of Change Management is to enable beneficial Changes to be made, with minimum disruption to IT Services.

Request for Change (RFC): A formal proposal for a Change to be made. An RFC includes details of the proposed Change, and may be recorded on paper or electronically. The term RFC is often

misused to mean a Change Record, or the Change itself.

Change Model: A repeatable way of dealing with a particular Category of Change. A Change Model defines specific pre-defined steps that will be followed for a Change of this Category. Change Models may be very simple, with no requirement for approval (e.g. Password Reset) or may be very complex with many steps that require approval (e.g. major software Release).

Change Advisory Board (CAB): A group of people that advises the Change Manager in the Assessment, prioritisation and scheduling of Changes. This board is usually made up of representatives from all areas within the IT Service Provider, the Business, and Third Parties such as Suppliers.

Standard Change: A pre-approved Change that is low Risk, relatively common and follows a Procedure or Work Instruction. For example password reset or provision of standard equipment to a new employee. RFCs are not required to implement a Standard Change, and they are logged and tracked using a different mechanism, such as a Service Request.

Every new release should be the result of a previous change request. Once the Request for Change is made the Change Advisory Board should decide about it. A formal procedure should be established of how RFC are submitted (Savannah, GGUS?)

We must define the different Categories of Change we will be working with and the Change Models that will be applied to each category. A proposal for these categories is already available in Mario David's document (bug or security vulnerability, new functionality with backwards compatibility, new functionality not necessarily with backwards compatibility, new service or node type).

We must also identify the Standard Changes that will be pre-approved.

Release and Deployment Management

Some definitions from the process:

Release Management: The Process responsible for Planning, scheduling and controlling the movement of Releases to Test and Live Environments. The primary Objective of Release Management is to ensure that the integrity of the Live Environment is protected and that the correct Components are released. Release Management is part of the Release and Deployment Management Process.

Deployment/Rollout Management: The Activity responsible for movement of new or changed hardware, software, documentation, Process, etc to the Live Environment. Deployment is part of the Release and Deployment Management Process.

Release Unit: Components of an IT Service that are normally Released together. A Release Unit typically includes sufficient Components to perform a useful Function. For example one Release Unit could be a Desktop PC, including Hardware, Software, Licenses, Documentation etc. A different Release Unit may be the complete Payroll Application, including IT Operations Procedures and User training.

Release Identification: A naming convention used to uniquely identify a Release. The Release Identification typically includes a reference to the Configuration Item and a version number. For example Microsoft Office 2003 SR2.

We have to define our **release unit**: it could be one package and its corresponding documentation, one complete service including all required packages and documentation, one node type, etc. If it is necessary we can define several release units at different levels.

We have to define what we will use as **release identification**.

Finally we should establish and document our **release policy**.