ITIL Version 3.0 (V.3) Service Transition Guidelines

By Braun Tacon

Executive Summary: This document is seven pages. Page one is informational/background only. What follows over the next six pages are the three principles of **Service Transition** which are fundamental to the delivery of Services that brings **Business Value.** By using this document as a checklist to ensure the identification of key requirements and to assess if those requirements are being met, we can help our Service Providers transition Services to Nike that delivers **Business Value**. Additionally by using a common set of criteria and measurement we can baseline where we currently stand and measure progress in a consistent and relevant way from both a tower and program point of view, with a goal of managing to our commitments and deliverables. Finally, since Change is the focus of **Service Transition**, Change is the focus of this document. Pages four and five are entirely devoted to the topic of **Change** and **Change Management**."

ITIL V. 3 is a framework for Service Management that is built around five separate lifecycle phases: Service Strategy, Service Design, Service Transition, Service Operation, and Continual Service Improvement. **In ITIL V. 3 a Service is defined as, "…a means of delivering value to customers by facilitating the outcomes those customers want to achieve without the ownership of specific costs and risks."**

Previously, we have been engaged in the **Service Strategy** phase. Service Strategy activities are:

- Define the market
- Develop the offerings
- Develop Strategic Assets
- Prepare for execution

Next came the **Service Design** phase. Please review this document for a fuller understanding of **Service Design.**

Once **Service Design** is complete, the next phase is **Service Transition**. The core activities of **Service Transition** are:

- Plan and manage the capacity and resources required to package, build, test, and deploy a release (service) into production
- Provide a consistent and rigorous framework for evaluating the Service capability and risk profile
- Establish and maintain the integrity of all identified **Service Assets** and configurations
- Provide good-quality knowledge and information
- Provide efficient and repeatable build and installation mechanisms
- **Core focus**: Ensure that the Service can be managed, operated, and supported according to the requirements and constraints specified within **Service Design**

The three major aspects of **Service Transition** are: **Change Management** (CM), **Service Asset and Configuration Management** (SACM), and **Release and Deployment Management** (RDM). Each of these principles is essential to effective **Service Transition** and each has a specific set of requirements and measures. Since CM is the most complex of these three topics, we will spend a bit more time discussing those concepts. This is not to say the other aspects are less important but since good CM will be a key dependency to our success during transition, it is also good practice to focus there. Before discussing the three aspects of **Service Transition** here are some general concepts and definitions that you should be familiar with:

- **Change** – A Change to an existing Service or the introduction of a new Service including: addition, modification, removal, documentation, etc. From a CM perspective, all **Service Change** should be authorized and planned
- **Change Advisory Board** (CAB) – A body that exists to support the authorization of Changes and to assist CM in the assessment and prioritization of the Changes
- **Change History** – Information about all changes made to a Configuration Item during its life. Change History consists of all those Change Records that apply to the CI.
- **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.
- **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). See Standard Change, Change Advisory Board.
- **Change Record** – A Record containing the details of a Change. Each Change Record documents the Lifecycle of a single Change. A Change Record is created for every Request for Change that is received, even those that are subsequently rejected. Change Records should reference the Configuration Items that are affected by the Change. Change Records are stored in the Configuration Management System.
- **Change Request** – Synonym for Request for Change.
- **Change Schedule** – A Document that lists all approved Changes and their planned implementation dates. A Change Schedule is sometimes called a Forward Schedule of Change, even though it also contains information about Changes that have already been implemented.
- **Change Types** – Changes are categorized into:
    - **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. See Change Model.
    - **Normal Change** – One raised by a request from the initiator (the individual or organizational group that requires the Change)
    - **Emergency Change** – A Change that must be introduced as soon as possible. For example to resolve a Major Incident or implement a Security patch. The Change Management Process will normally have a specific Procedure for handling Emergency Changes. See Emergency Change Advisory Board (ECAB).
- **Change Window** – A regular, agreed time when Changes or Releases may be implemented with minimal impact on Services. Change Windows are usually documented in SLAs.
- **Configuration Item (CI)** – Any Component that needs to be managed in order to deliver an IT Service. Information about each CI is recorded in a Configuration Record within the Configuration Management System and is maintained throughout its Lifecycle by Configuration Management. CIs are under the control of Change Management. CIs typically include IT Services, hardware, software, buildings, people, and formal documentation such as Process documentation and SLAs.

- **Configuration Management** – The Process responsible for maintaining information about Configuration Items required to deliver an IT Service, including their Relationships. This information is managed throughout the Lifecycle of the CI. Configuration Management is part of an overall Service Asset and Configuration Management Process.
- **Configuration Management Database** (CMDB) – The CMDB is a database used to store Configuration Records throughout their Lifecycle. The CMS maintains one or more CMDB (Federated CMS), but it is the CMDB that is used to store the specific attributes of the CIs and their relationships with other CIs. Whenever possible, automation should be used to update and manage the CMDB in order to reduce both cost and errors
- **Configuration Management System** (CMS) – The goal of a CMS is to provide reliable, quick, and easy access to accurate configuration information. Think of CMS as a process, tool, or a combination of both that will allow stakeholders to assess the impact of proposed Changes, to track Changes, and to ensure Changes are delivered to the appropriate party or into the correct environment.
- **CI Type** – A Category that is used to Classify CIs. The CI Type identifies the required Attributes and Relationships for a Configuration Record. Common CI Types include: hardware, Document, User etc.
- **Deployment** – 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.
- **Emergency Change Advisory Board** (ECAB) – A sub-set of the Change Advisory Board who make decisions about high impact Emergency Changes. Membership of the ECAB may be decided at the time a meeting is called, and depends on the nature of the Emergency Change.
- **Release** – A collection of hardware, software, documentation, Processes or other Components required to implement one or more approved Changes to IT Services. The contents of each Release are managed, Tested, and Deployed as a single entity.
- **Release and Deployment Management** – The Process responsible for both Release Management and Deployment.
- **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.
- **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 Window** – Synonym for Change Window.
- **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.
- **Seven Rs of Change Management** – Answer the seven questions to understand the impact of Changes
    - Who **Raised** the Change?
    - What is the **Reason** for the Change?
    - What is the **Return** required from the Change?
    - What are the **Risks** involved the Change
    - What **Resources** are required to deliver the Change?
    - Who is **Responsible** for the build, test, and implementation of the Change?
    - What is the **Relationship** between this Change and other Changes?
- **Utility** – “Fit for Purpose”, meets requirements and expectations
- **Warranty** – Will perform as agreed to, and mitigating and compensating controls exist (SLA)

Change Management Objectives: The goal of the Change Management process is to ensure that Changes are recorded and then: **Evaluated**, **Authorized**, **Prioritized**, **Planned**, **Tested**, **Implemented**, and finally **Documented**.

Change Management Concepts: CM processes should be designed and planned in conjunction with, not separate from, the **Release and Deployment** and **Service Asset Configuration Management** processes. Using this approach helps to evaluate the impact of the Change on the current and planned services and releases. Different types of Changes may require different types of Change requests (RFC). Some examples of various types of Changes are:

- RFC to Service Portfolios could include a new portfolio line item or an alteration to the portfolio's scope, business case or baseline
- RFC to Service or Service Definition could be a Change in existing or planned Service attributes, a Project Change that impacts Service Design, or something as simple as a Service improvement
- Some Changes do not require an RFC; these are known as **Standard Changes**. Two examples would be a request for User Access, or typical operational activities such as rebooting hardware on failure (as long as there is no impact on other Services) and planned maintenance

Change Management Roles: There are two basic roles which support the CM process.

- **Change Manager** – Receives, logs, and allocates priorities. Tables all RFCs for the CAB meeting. Decides which people will come to which meetings. Convenes the CAB or Emergency CAB (ECAB) meetings for RFC assessment. Chairs all CAB and ECAB meetings. Authorizes acceptable Changes.
- **Change Advisory Board** (CAB) – Is an advisory body, requiring appropriate terms of reference such as meeting regulations and scope of influence. The CAB also ensures that formal authorization is obtained for each Change from the appropriate Change authority such as, but not limited to a Change Manager, a CAB or ECAB, an IT Management Board or a Business Executive Board

Change Management Scope: The scope of the CM process covers Changes to Service Assets and CIs across the entire Service Management Lifecycle from Service Strategy to Continuous Service Improvement.

Change Management Process Activities: Any CM process should include the following activities at a minimum.

- Planning and controlling Changes
- Change and Release scheduling
- Communications
- Change decision making and Change authorization
- Ensuring there are remediation or back out plans
- Measurement and control
- Management reporting
- Understanding the impact of Change
- Continual improvement

Additionally, your processes should ensure that you have a mechanism to filter out Changes that are totally impractical, are repeats of earlier RFCs that have been previously accepted or rejected, and incomplete submissions. All Changes should be assigned a Category and Priority, and finally Changes should be authorized by the appropriate authority. While a Standard Change can be done with local authorization, a high cost/risk Change probably requires decisions from executives such as the Board of Directors.

Authorized Changes: All authorized Changes (**and all Changes <u>SHOULD</u> be authorized**) should be passed to the appropriate technical groups. **CM is responsible** for ensuring that Changes are implemented as scheduled. This Role should be viewed as a coordination function, because the implementation will be the responsibility of the **Release and Deployment Management** Role.

Change Review: Is done on the completion of the Change. For major Changes there will be a larger customer and stakeholder input and audience, and should include any Incidents or missed SLA or Regulatory requirements. A **Post-Implementation Review** (PIR) is done to ensure that the Change met objectives, initiator and customer requirements are met to at least the "meets expectations" level, and to confirm that there has been no unexpected consequences. **All lessons learned are fed back into future Change Processes**.

Change Management Key Metrics: The following are some useful metrics (KPIs). This is not an all-encompassing list. Use whatever metrics meet your design and requirement needs, but these metrics will be useful as a guide in developing other metrics.

- The number of Changes implemented to services that met the customer's agreed requirements. This can be a relationship between quality/cost/time against percentage of Changes requested.
- Reduction in the number of disruptions, defects, or rework required caused by poor CM practices or incomplete or inaccurate specifications
- Percentage and reduction of unauthorized Changes
- Percentage and reduction of backlog RFCs
- Percentage and reduction of emergency or otherwise unplanned Changes
- Percentage and reduction of Changes where the back out plan was used
- Percentage and reduction of failed Changes
- Average time to implement based upon Urgency/Priority/Change type.
- Percentage and reduction of Changes which caused incidents
- Percentage of Changes which are "fit for purpose" the first time. This is a measurement of accuracy

Change Management Considerations: Some topics to consider ensuring that our **Change Management** processes deliver the expected **Business Value**

- Are we creating a culture of CM across the org, including zero tolerance for unauthorized Changes?
- Do we align Service CM processes with business, project, and stakeholder CM processes?
- Are we prioritizing Change, and establishing accountability throughout the CM Lifecycle?
- Do we have a Single Point of Contact (SPOC); this could be an individual, role, function or group whose Role is to prevent uncoordinated Change from occurring across multiple functions?
- Do we prevent people who should not be making Changes from accessing the production environment in such a way that they could make Change?
- Are our CM processes integrated with other **Service Management** processes in a way that establishes traceability of Change? Also, do we detect unauthorized Change, and identify Change-related incidents?
- Do we have established Change windows, and are those windows enforced so that Change that occurs outside of those windows is being considered as Emergency Change with a higher level of justification and review?
- Do we conduct performance and risk evaluations on any Change that have the potential to disrupt a Service?

- Do we have a feedback loop for **Continuous Service Improvement** and are we measuring our performance?

Service Asset and Configuration Management Objectives: To define and control the components of Services and Infrastructure and to maintain accurate configuration information on the historical, planned and current state.

Service Asset and Configuration Management Concepts: To deliver a model of the Services, Assets, and Infrastructure by recording the relationships between CIs. In an Enterprise the size of Nike, this is best accomplished with a meta-system commonly known as the CMS. While the concept could be discussed more from a very deep and technical perspective, it will suffice for the purpose of this document that you are familiar with the following two concepts and definitions:

- **Configuration Baseline**: A configuration of a Service, Product or Infrastructure that has been reviewed and agreed upon
- **Snapshot**: A snapshot of the current state of a CI, Environment, or Baseline to demonstrate Stability or Change

Service Asset and Configuration Management Roles:

- **Service Asset Manager** – Delivers to the overall objectives agreed upon with the **IT Service Manager**, evaluates existing **Asset Management,** and manages the scope of the **Asset Management** process
- **Configuration Manager** – Delivers to the overall objectives agree upon with the **IT Service Manager**, oversees existing CMS, and manages the scope of the **Configuration Management** process
- **Configuration Analyst** – Proposes scope, trains **Asset and Configuration Management** specialists, and supports the creation of **Asset and Configuration Management** plans
- **CMS/Tools Administrator** – Evaluates tools, monitors performance and capacity, and additional duties as required

Release and Deployment Management Objectives: To ensure that there are clear and comprehensive release and deployment plans that bring alignment with customer and business Change plans. To build a release package that can be built, installed, tested, and deployed efficiently and on schedule. That all new or changed Services and Systems deliver to agreed upon SLA, Utility, and Warranty. Releases will bring minimal unpredicted impact on Production, Operations, and Support Services, and that all customers are satisfied with all aspects of the **Service Transition** phase, including for example User Documentation and Training.

Release and Deployment Management Concepts: A **Release Unit** is the portion of a Service or an IT Infrastructure released together. This unit may vary; one size does not fit all. When considering a Release, some options are

- Big bang vs. phased
- Push vs. pull
- Automation vs. manual

Additionally, **Release and Deployment** manages the

- Release structure
- Release exit and entry criteria
- Control

- Roles and Responsibilities

Release and Deployment Roles: The following defines the Roles in **Release and Deployment**

- **Release and Deployment Manager** – Is responsible for the planning, design, build, configuration, and testing of all Service, Software, and Hardware used to create the Release Package (**Release Unit**)
- **Release packaging and Build Manager** – Is responsible for defining the final release configuration including, knowledge, information, hardware, software, infrastructure, etc
- **Deployment Staff** – Deliver the final physical implementation, coordinate Release documentation and communications including training, and plan the deployment in agreement with Change and Knowledge Management