Release Management Policy

Brunel University London

An ISO/IEC 27001:2013: *Aligned Document - Implementing Cyber and Information Security Best Practice*

Internal Use Only

Mick Jenkins
Chief Information Security Officer

Document History

Version	Author	Comments	Date
V 0.1	Andrew Clarke	Initial Draft	16/11/2016
V 0.2	Andrew Clarke	Define scope more clearly to differentiate release and patch management	17/02/2017
V 0.3	Andrew Clarke	Add that RM should be run as a project in most cases (7 - Overview)	27/02/2017
V 0.4	Andrew Clarke	CISA amendment (TY) to provide for exceptions, flexibility and risk assessment. Approved	05/04/2017
V 1.0	Andrew Clarke	Approved Information Subcommittee	27/04/2017
V 1.1	Andrew Clarke	Approved ISC	20/06/2017
V 1.1	Andrew Clarke	Annual review	18/09/2018
V 1.1	Andrew Clarke	Annual review	09/09/2019
V 1.1	Andrew Clarke	Annual review	07/09/2020

Document Approval

Owner: Michael Jenkins	Chief Information Security Officer
Signature: MGJ	Date: 26 Jan 2018
Approver: Pekka Kahkipuro	Chief Information Officer
Signature: PK	Date: 26 Jan 2018
Distribution:	

This document requires the approval from BUL as defined in the ISMS Compliance document.

Contents

1. Purpose of Document

This document defines the Release Management Policy that will need to be applied relating to the Brunel University London process of planning, building, testing and deploying hardware and software and the version control and storage of software. Its purpose is to ensure that a consistent method of deployment is followed. It reduces the likelihood of incidents as a result of rollouts and ensures that only tested and accepted versions of hardware and software are installed at any time.

Please refer to Brunel University London ISMS Document BUL-GLOS-000 - SyOPs Glossary of Terms for the glossary of terms, acronyms and their definitions for the suite of Brunel University London ISMS documentations.

2. Responsibilities

Table 1 – responsibilities

Title / Role	Description
Systems Manager	Is responsible for monitoring vendors' releases of upgrades and new releases on the operational systems. Is responsible for installing IS infrastructure software updates and for testing software items updates and new implementations.
Head of Infrastructure and Operations	Is responsible for overall Release Management on all systems managed by IS.
Software tester	Is responsible for testing development software items updates and new implementations
Head of Development and Application Services	Is responsible for monitoring vendors' releases of upgrades and new releases on server application software. Is responsible for installing server application software updates and testing software items updates and new implementations
Software Application Owners	Are responsible for tracking upgrades available for their assets
Cyber & Information Security Manager	Is responsible for vulnerability risk assessment

3. ISO 27001 - Conformance

This section indicates the University Conformance to ISO27001:2013.

University ISMS Control Number	SOA – Number A12 – Operations Security

ISO 27001:2013 Conformance Control	Information Classification Objective A.12.5 Control of operational software

4. Reference

UCISA: ITIL – A Guide to Release and Deployment Management
HMG Security Policy Framework

5. Background

Release Management is proactive technical support focused on the planning and preparation of new service deliverables. This provides the assurances for:

- The opportunity to plan expenditure and resource requirements in advance;
- Define and agree release and deployment plans with customers/stakeholders;
- A structured approach to rolling out all new software or hardware, which is efficient and effective;
- Ensure that each release package consists of a set of related assets and service components that are compatible with each other;
- Ensure that integrity of a release package and its constituent components is maintained throughout the transition activities and recorded accurately in the configuration management system;
- Changes to software are 'bundled' together for one release, which minimises the impact of changes on users;
- Testing before rollout, which minimises incidents affecting users and requires less reactive support;
- An opportunity for users to accept or reject functionality of software before it is fully implemented;
- Ensure that all release and deployment packages can be tracked, installed, tested, verified, and/or uninstalled or backed out, if appropriate;
- Ensure that change is managed during the release and deployment activities;
- Record and manage deviations, risks, issues related to the new or changed service, and take necessary corrective action;
- Ensure that there is knowledge transfer to enable the customers and users to optimise their use of the service to support their business activities;
- Ensure that skills and knowledge are transferred to operations and support staff to enable them to effectively and efficiently deliver, support and maintain the service, according to required warranties and service levels;
- Version control and central storage of software, ensuring that correct versions are installed at all times, which minimises incidents and the need for reinstallation.

6. Scope

This document defines the University approach for Release Management for vendor released upgraded Operating System releases and Service Packs and for Application Major releases.

In the short-term, Release Management should be applied to the installation of single instances of hardware or software. This exercise can be used to begin the generation of standard builds and a centralised store of software and introduces the concept of a standard process for implementing all equipment.

In the long-term, Release Management should be applied as a strategy for introducing all new software or hardware in a planned, controlled and structured manner. This should therefore reduce the need for ad-hoc requirements as far as possible and allow technical support time to be focused on other activities. It also results in economies of scale as the planning and preparation activities do not increase in proportion to the number of items – these tasks must be performed whether the exercise is to install one computer or ten.

The scope of release and deployment management includes the processes, systems and functions to package, build, test and deploy a release into operation.

- All the University's information systems are within the scope of this procedure;
- Server Operating Systems;
- Desktop PCs, Workstations and Laptops;
- Handheld Devices;
- Server Applications both local and IS managed;
- The Definitive Software Library (**DSL**) must hold a copy of all the software installed in the IT environment. This includes not just operating systems and applications, but also device drivers and any associated documentation.

7. Overview

The Release Management process works by providing a consistent framework for defining and creating new services, and ensuring that the correct versions of tested and approved software are implemented on a day-to-day basis (that is, after initial rollout).
It interfaces with the Change Management process to enable implementation and to the Configuration Management process to maintain configuration records.
As Release Management usually entails both potential impact in regard to the changes to either systems or services and for the requirement of resources owing to the scale of the Release, Release Management should be run as a project in most cases.

8. Release Procedure

The release procedure covers release numbering, frequency and the level in the IT infrastructure that will be controlled by definable releases. The University should decide the most appropriate approach, depending on the size and nature of the

systems, the number and frequency of releases required, and any special needs of the users – for example, if a phased rollout is required over an extended period of time. All releases should have a unique identifier that can be used by configuration management.

The term release is used to describe a collection of authorised changes to an IT service. A release is defined by the RFCs that it implements. The release will typically consist of a number of problem fixes and enhancements to the service. A release consists of the new or changed software required and any new or changed hardware needed to implement the approved changes.
Releases are often divided into:

- **Major software releases** and hardware upgrades, normally containing large areas of new functionality, some of which may make intervening fixes to problems redundant.
  A major upgrade or release usually supersedes all preceding minor upgrades, releases and emergency fixes. A major release can be defined by the vendor usually as a renaming convention (e.g. Windows XP to Windows Vista) or as an integer version release (e.g. Windows 8 to Windows 10). A major software OS or application release is usually attributed to changes made that invoke incompatible API (Application Programming Interface) changes.
  A Major Release requires compliance with Project standards.
- **Minor software releases** and hardware upgrades, normally containing small enhancements and fixes, some of which may have already been issued as emergency fixes.
  A minor upgrade or release usually supersedes all preceding emergency fixes. A minor release by a vendor is usually identified in the naming convention as prefixed by the exiting naming convention followed by a decimal point (e.g. Windows 8 to Windows 8.1 or Advanced Aircraft Analysis 3.6 to Advanced Aircraft Analysis 3.7). This is covered under the BUL-POL-12.6 - Patch Management Policy and is out of scope for Release Management.
- **Emergency** software and hardware fixes, normally containing the corrections to a small number of known problems. This is covered under the BUL-POL-12.6 - Patch Management Policy and is out of scope for Release Management.

9. Plan and Deployment

Release and deployment plans
Plans for release and deployment will be linked into the overall service transition plan. The approach is to ensure an acceptable set of guidelines is in place for the release into live/operation.
Release and deployment plans should be authorised as part of the change management process.
The plan should define the:

- Scope and content of the release;
- Risk assessment and risk profile for the release;
- Customers/users affected by the release;

- CAB members that approved the change request for the release and/or deployment;
- Team who will be responsible for the release;
- Delivery and deployment strategy;
- Resources for the release and deployment.

Logistics and delivery planning

Once the overall deployment approach is understood, develop the logistics and delivery plans. These plans deal with aspects such as:

- How and when release units and service components will be delivered;
- What the typical lead times are; what happens if there is a delay;
- How to track progress of the delivery and obtain confirmation of delivery;
- Metrics for monitoring and determining success of the release deployment.

Build and test of releases

Key aspects that need to be managed during the activities to build and test a service are:

- Usage of the build and test environments;
- Recording the complete record of the build so that it can be rebuilt if required;
- Maintaining evidence of testing, e.g. test results and test report;
- Checking that security requirements are met;
- Verification activities, e.g. prerequisites are met before a build or test begins.

Release and build documentation

Procedures, templates and guidance should be used to enable the release team to build an integrated release package efficiently and effectively.

Procedures and documents will be required for purchasing, distributing, installing, moving and controlling assets and components that are relevant to acquiring, building and testing a release.

Release packaging

Build management procedures, methodologies, tools and checklists should be applied to ensure that the release package is built in a standard and controlled way in line with the solution design defined in the service design package. As a release package progresses towards production it may need to be rebuilt.

The key activities to build a release package are:

- Assemble and integrate the release components in a controlled manner;
- Create the build and release documentation including: build, installation and test plans, procedures and scripts;
- Monitor and check the quality of the release and how to recognise and react to problems;
- The automated or manual processes and procedures required to distribute, deploy and install the release into the target environment (or remove it as necessary);
- Procedures to back out release units or remediate a change should a release fail;
- Procedures for tracking and managing software licences;
- Install and verify the release package;
- Baseline the contents of the release package;
- Send a notification to relevant parties that the release package is available for installation and use.

If testing of a release package is successful, the release and the contents of the release package are placed under the control of configuration management, baselined and verified against the release design and release package definition.
From this point all changes to the release package are managed through change management.
If at any step the testing of a release package does not complete successfully, reassessment and rescheduling of the release is managed through change management.

Release Exception
Exceptions to the release management policy require formal documented approval from the Head of Infrastructure and Operations and the Cyber & Information Security Manager. Any servers or workstations that do not comply with policy must have an approved exception on file.
An applicable release that cannot be implemented by the implementation deadline is an exception and requires a Security Exception Letter.
The table below describes how this form is completed and approved.

When a patch exception may occur.	Who completes the form and obtains approvals.
During the initial assessment.	Functional Support.
During testing or implementation.	Functional Support or Software Application Owner.
At any time for business reasons.	Software Application Owner.

Note: Functional Support is defined as the group responsible for identifying and assessing releases and performing Functionality Testing.

10. Review and close a deployment

When reviewing a deployment the following activities should be included:

- Capture experiences and feedback on customer, user and service provider satisfaction with the deployment, e.g. through feedback surveys;
- Review quality criteria that were not met;
- Check that any actions, necessary fixes and changes are complete;
- Review performance targets and achievements, including resource use and capacity such as user accesses, transactions and data volumes;
- Make sure there are no capability, resource, capacity or performance issues at the end of the deployment;
- Check that any problems, known errors and workarounds are documented and accepted by the customers/ business and/or suppliers;
- Incident and problems caused by deployment;
- Deployment is completed with a handover of the support for the deployment group or target environment to service operations;
- A post implementation review of a deployment is conducted through change management.

11. Monitoring

The Release Management process should be monitored regularly as soon as possible after introducing it. The initial period will focus on developing and documenting builds and benchmarks. Release management activities should be report monthly.