

7.7 The unclear relationship between change, release and project management

***D**o you have a clear picture of the interfaces and relationships between change, release and project management? Even after analyzing what ITIL and PRINCE2 - market-leader frameworks in the process domain - suggest? In this article, Christian Cantù and Maxime Sottini draw a picture of the potential relationships between these processes, and give practical guidance on how to implement them.*

INTRODUCTION AND OBJECTIVES

Service and project management are recognized to be among the core disciplines in order to effectively and efficiently manage IT organizations and achieve IT goals. Project management is a mature discipline, evolved outside the IT domain, but widely adopted in IT too. The modern development of project management starts in the fifties, although Gantt charts were developed and used before this. From its origin, different approaches and frameworks have evolved and emerged. Among them, some have been more successful, for example PMBOK® (Project Management Body of Knowledge) and PRINCE2™ (Projects IN Controlled Environments). The differences between these frameworks are not so relevant for the purposes of this article and references from the latter will be used where necessary to aid comprehension and limit the need for a glossary and explanations.

Service management has developed later in comparison to project management. In the IT domain, ITIL® (the IT Infrastructure Library), has greatly contributed to the development of the discipline from the eighties to present day.

Both project and service management have specific processes as central components of the respective disciplines. Applied to the same IT management domain, these processes are used to support the same items, such as applications, infrastructures, resources and objectives. Although in the specific frameworks there are some attempts to clarify the general relationships between the different disciplines, many practical questions arise when implementing these disciplines and focusing upon the detail. Theory is not able to answer some questions, often related to the joint use of the processes suggested by the two disciplines. But there are also aspects relating to organization and tools that require further understanding.

The main area of improvement in terms of understanding and knowledge lies in the relationship between project and service management. In the IT domain, the main objective of project management is to support a controlled evolution and transformation of applications and infrastructures. Service management supports the effective delivery of applications and infrastructures. At first sight these two aims appear to be synergistic, but on looking closer we can easily find overlapping areas. This is because IT services are not static, rather they evolve continuously and therefore IT management is concerned about the evolution and transformation of services. Before the advent of ITIL V3, the main processes able to guarantee this evolution were release and change management. These two processes have

a strong relationship with project management processes. In some cases they are mutually exclusive, in some others strongly synergistic. Small service evolutions, for example, may be managed without project management techniques and processes, instead using the basic service management processes such as change management. On the contrary, a significant service evolution will probably need to be managed as a project, which may include the management of several releases.

When considering these relationships, which whilst they are clearly evident are certainly not the only such ones, it is clear that some practical guidance on how to define and apply them would be useful. When does a change have to be managed as a project? Does a release have to be managed as a project? When does a project activate changes and releases? Which are the correct responsibilities and activities of project and change management functions? These are only some of the many questions arising when considering project and service management. The aim of this article is to explore them and to supply practical guidance alongside the theoretical approaches, with explicit references to ITIL and PRINCE2.

In order to achieve this aim, the following steps will be performed. A brief introduction of service (ITIL) and project (PRINCE2) management theory will be given. This introduction will support the identification of change and release management as the main processes to focus on, and will give a better understanding of the main issues concerning the relationships between the target processes. These issues will be explored in a dedicated section. Later, a general overview of the relationships between the processes, including a graphical model, will be given. This will be used to identify the specific interfaces between the processes, and these will be explored in further details, by means of the following specific sections: change and release, change and project, and project and release. Finally, a number of conclusions will be drawn.

ITIL THEORY OF CHANGE AND RELEASE MANAGEMENT

Change management is often related to people (a structured approach to manage the impact of change to individuals, teams, organizations), but in our context we mean a specific process of IT service management. This process has been well described in ITIL V2:

“Change management is 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”, where Changes are meant as “The addition, modification or removal of anything that could have an effect on IT Services. The Scope should include all Configuration Items, Processes, Documentation etc.”

In ITIL V3, the process has been placed into the Transition phase of the Service Lifecycle with minor modifications to its contents, though this is not relevant to the aim of this article. Change management is a sequence of activities starting from recording the Request for Change (RFC) and finishing with change review and closure, through the steps of assessment and authorization, planning and implementation, in order to obtain the intended modifications, subject of the initial RFC. The process according to ITIL V3 is described in figure 1.

Change management is normally performed by permanent service management roles and functions of the organization, such as the change manager, the Change Advisory Board (CAB), and service owners; it is supported by service management integrated tools.

The unclear relationship between change, release and project management

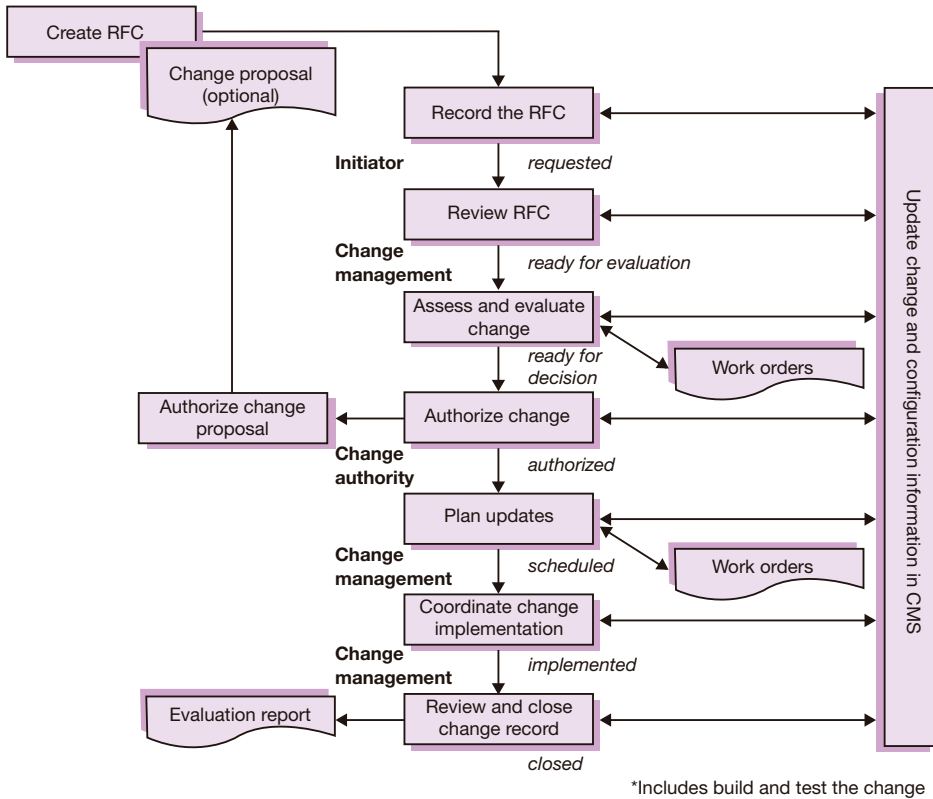


Figure 1 The change management process according to ITIL V3 (Source: OGC)

In ITIL V3, some relationships between project and change management are identified. Probably, the most important is defined in Service Design, where in the principles it is stated that the introduction of, or the major modifications to, IT services have to be managed by means of projects. Figure 2 depicts how the project approach is needed to manage the stages within the lifecycle for the deployment of a new or modified service (change). Figure 2 also highlights the role, among all the Service Lifecycle processes, taken on by build, test, release, deployment management and change management.

In ITIL V3, specifically in the change management process description, some examples of change request are listed, together with the procedures to be used to manage them. We can derive the following:

- Projects should submit changes affecting services to the service change management procedure.
- Projects should use standard changes and service requests for pre-authorized typical operational requests (e.g. user access request).
- Projects have specific change management procedures for changes not impacting the service or design baseline.

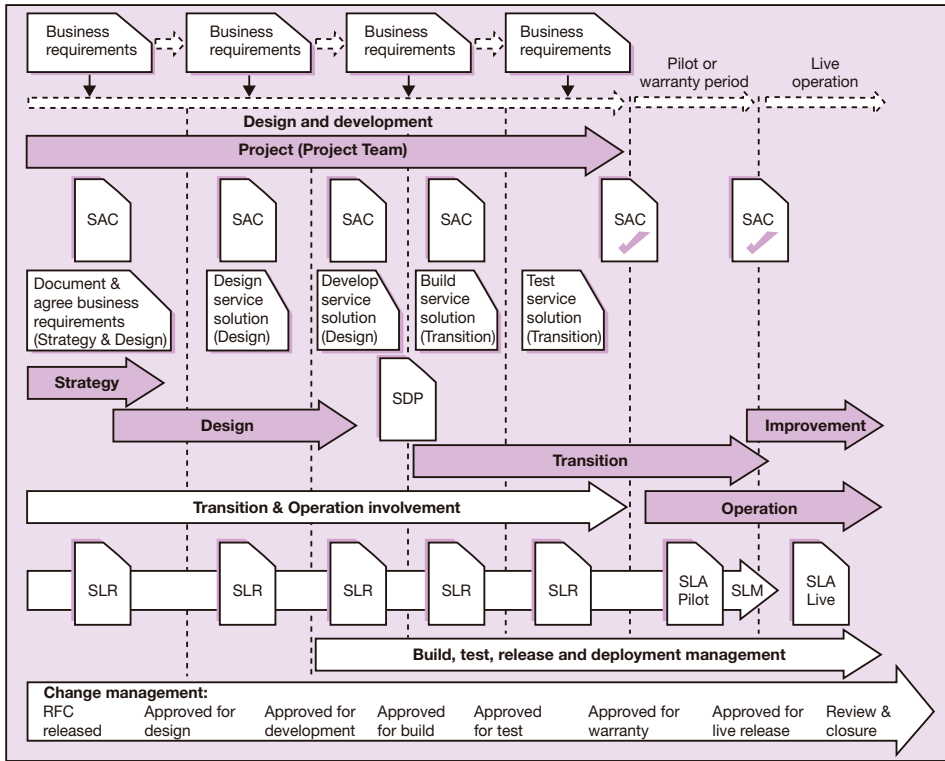


Figure 2 Relationships between project management and service lifecycle phases (Source: OGC)

Release management has greatly evolved with the new version of ITIL. The definition of the process in version 2 is the following:

“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 works closely with Configuration Management and Change Management.”

In the ITIL V3 definition, release management has become *“part of the Release and Deployment Management Process”*. For the purpose of this article, we will consider the definition contained in ITIL V2 to be more appropriate, thus viewing release management as being made up of planning, building, testing and deploying activities. Within this scope we will consider the relationships with the other processes, change and project management.

One of the clearest relationships between release and change management is depicted in figure 2. Release management is the process that actually takes changes into the live environment. From the state “Approved for build” onwards, the change is executed through release management. In other words, each release may include and implement one or more changes.

Figure 3 depicts another existing interface between release and change management, a relation according to which an RFC is created in order to activate an instance of the release management process; this, in turn, deploys a specific release and, therefore, brings specific changes into a live environment. This interface also highlights the role of change management as a means to execute some steps of the release management process.

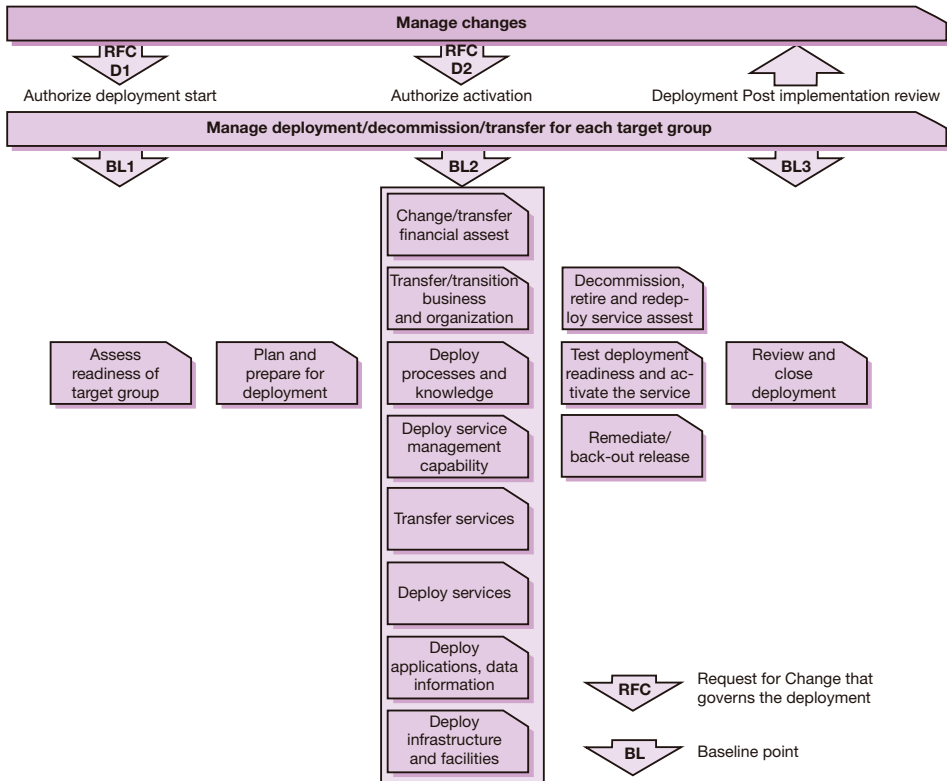


Figure 3 Some relationships between change and release management (Source: OGC)

In synthesis, the main interfaces identified between change and release management are as follows:

- Releases may embed one or more changes which are deployed and finally implemented through those releases.
- Some activities of the release management process, such as installation of builds, may be activated and performed by means of changes.

Finally, and regarding the interfaces between project and release management, in ITIL V3 an explicit relationship is identified in planning activities. Project plans should include release activities as well as the resources needed to perform them. Among these activities, as an example, an important role is played by testing. More generally speaking, the release management approach, e.g. the V-Model referred to by ITIL V3, shall influence project plans definition.

PRINCE2 THEORY OF PROJECT MANAGEMENT

Founded in 1975 and originally known as PROMPT, this project management framework was adopted by the Central Computer and Telecommunications Agency (CCTA, then Office of Government Commerce, the same founder of the ITIL framework) as a standard to manage information systems projects. It was developed and refined over years of use till the current version, PRINCE2™ project management method.

PRINCE2 is a repeatable method, built on experience, available to organizations willing to adopt a structured “project” approach to deal with the creation and delivery of new business products, or to implement any change to their environments. It is therefore not only applicable in the IT domain.

A key principle of the method is its “product-based” approach, i.e. an approach focused on the products (deliverables) produced by the project along its lifecycle, and not the activities performed to produce those products. The core structure of the method is made up of:

- processes, that provide a controlled start, progress and closure of any project
- components, i.e. the different aspects of a project that are taken into consideration (e.g. business case, organization, controls, plans, ...)
- techniques, supporting the implementation of processes

Figure 4 depicts the process model as detailed in the PRINCE2 framework, a model that needs to be tailored to the specific requirements of the project/organization.

The model is made up of eight processes:

- “Directing a Project (DP)” is the process aimed at the senior management team (project board), responsible for the whole project; the key decision-makers are, therefore, specifically involved in this step, that includes approval of the project, monitoring of its progress and closure.
- “Starting Up (SU)” and “Initiating a Project (IP)” are two initial steps aimed at assessing all the information and knowledge of the project, such as the project aim, project management team, customer’s expectations, and establishing a baseline against which measure the progress. At the end of these steps, a final confirmation for project activation and a plan for the whole project are obtained.
- “Controlling a Stage (CS)” is one of the core process steps owned by the project manager, who is involved in monitoring and controlling activities, in order to manage a single stage from approval to completion.
- “Managing Product Delivery (MP)” provides a control mechanism to ensure the production and delivery of project’ deliverables (products).
- “Managing Stage Boundaries (SB)” allows the project board to be aware of the status of project plan against its business case and the project to go ahead, from one stage to the other.
- “Closing a Project (CP)” is the step according to which the project manager asks the project board for authorization for project closure.
- “Planning (PL)” is the repeatable process step used whenever a plan is required or needs to be updated.

PRINCE2 also includes components and techniques which add guidelines and hints to manage projects successfully. Components are:

- Business case
- Organization
- Planning (this is also a process)

The unclear relationship between change, release and project management

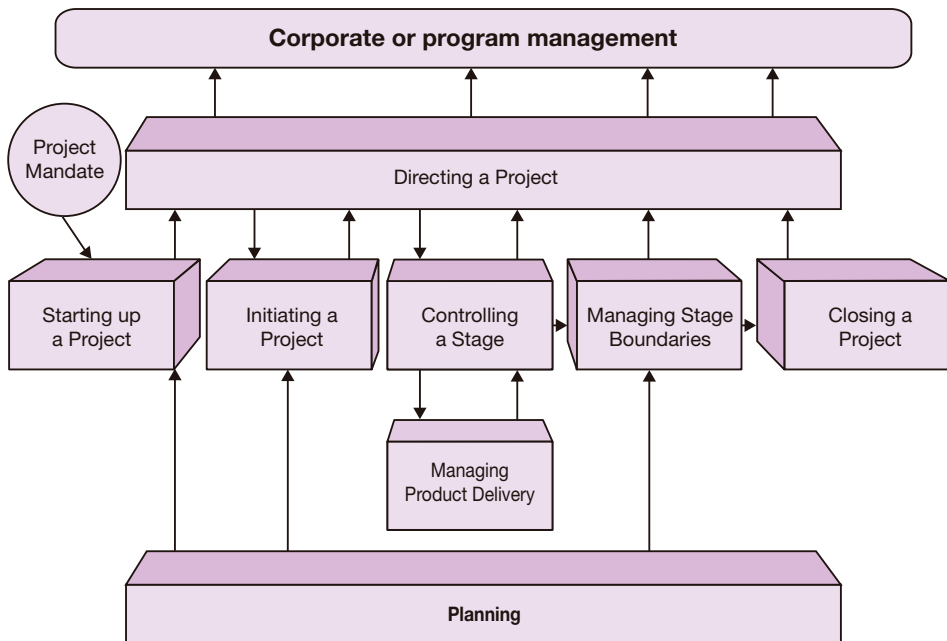


Figure 4 Processes in PRINCE2 (Source: OGC)

- Controls
- Management of risk
- Quality in a project environment
- Configuration management
- Change control

The final component seems to be apparently related to service change management, but similarities end as soon as you begin to get into detail. Change control deals with changes to the scope and specifications of a project and establishes a formal authorisation process for them. A request for change is a request to alter the current specification of a product.

Given the focus of this article, which is examining the relationship between change and release management processes, there are several concepts of interest in PRINCE2:

- Product orientation - Products are deliverables of the project and they are derived as outputs of service management changes and/or releases.
- Stages - These are phases of the project, each requiring formal approval to commence and to be closed. They may correspond to the phases of a release (for example deployment).
- Project board - This is the decision authority of the project, approving initial objectives, scope, products, etc., and any deviation to it, including changes as previously mentioned and intended.
- Work packages - These are set of instructions issued to a team manager in order to complete an area of work for the project. They contribute to the implementation of products and, therefore, they may also contribute to the implementation of changes and/or releases.

If we look at work packages in particular, they are created during “CS1 Authorising a Work Package” by the project manager with input from the team manager and team members. Once created, the work package is accepted and agreed in “MP1 Accepting a Work Package”, where dates and other details are negotiated. A Work Package is later implemented through “MP2 Executing a Work Package” and finally delivered through “MP3 Delivering a Work Package” process.

Even if PRINCE2 theory doesn't highlight relationships between project management and change or release management explicitly, the main areas where these processes will be identified, are the CS (Controlling a Stage) and MP (Managing a Product) processes.

ISSUES REGARDING RELATIONSHIPS BETWEEN CHANGE, PROJECT AND RELEASE MANAGEMENT

Unless IT service management deals with day-to-day activities, it appears to be a separate and independent domain, unrelated to project management. But when service management is about important modifications to services, the borders are no longer sharply defined. In the previous sections, we have explored the target processes and the relationships identified in the theoretical approaches between them. In this section, we illustrate why further analysis and understanding is needed.

As discussed, some relationships have already been drawn from a theoretical point of view, but are they exhaustive? Some important relationships have not been identified yet. For example, do changes have to be managed as projects? Experience suggests that, in certain cases, when changes are significant in terms of their size and the resources required, then it seems to be reasonable, if not actually recommended. In fact, the change management process may be not suitable and sufficient to guarantee the control of risks and the achievement of all intended results. But when does a change have to be managed as a project? Is there a specific size of the change, according to which project management processes have to be activated? Is a PRINCE2 approach suitable to manage it, or is it too complex?

A similar question may arise for release management. If a release is not derived from an existing project, should it be still managed as a project? This situation is not depicted in theory, but in real life there may be complex circumstances that could lead to such an instance. Again, it is not obvious as to the conditions in which project management should be activated and which approach is best suited.

Other relationships have been identified, but not deeply analyzed. For example, the relationship between release and change management. Release management requires the installation of a specific release in different environments (system test, acceptance test, production, etc...). Should this task be managed through changes, should these changes be standard ones? After all, these are modifications of target environments. The theory does not give a clear answer.

Another relationship, not completely explored and whose impact hasn't been well addressed, is the one related to the organizational issues existing within change, release and project management. These processes may lead to different, and in some cases even contrasting, objectives: project management focuses upon deadlines, whilst with change and release management the emphasis is on assuring minimal disruption of existing IT services. Should

change and release management be supported by permanent functions or should the tasks and responsibilities be performed by transient ones, included in the organizational structure of the project? How should project management processes (in particular those responsible for delivering products, e.g. CS and MP) be combined with change and release management processes? These choices may reinforce or reduce the capacity to achieve specific types of objectives such as timing vs. quality.

Finally tools. Service (including change and release) and project management are mature markets for tools. But generally, these tools do not cover both disciplines. Some questions arise: what are the main tasks to be accomplished by each tool? Which are the main interfaces to be implemented to bridge them?

These are only some of the questions that this article will try to address in the following sections, each of which will focus upon two processes, illustrating and analyzing bi-directionally the interfaces between them. But first of all we will provide a general overview of all relationships.

OVERVIEW OF THE RELATIONSHIPS

We have identified two main scenarios to be considered in order to explore the relationships between change, release and project management processes. The first one is depicted in figure 5 and it is referred to as the day-to-day operations scenario. Service management processes take care of routine. Changes are required by problem or incident management as well as other processes, such as service level or demand management, and they can be related to small evolutions/enhancements/ fixings of the services. In this scenario, the interfaces to explore are:

- **Changes to releases (1)** - Changes are deployed by means of a release.
- **Release to changes (2)** - Specific changes are issued in the release management process in order to install and activate a release.
- **Release to project (3)** - A release (for example a major release) needs additional

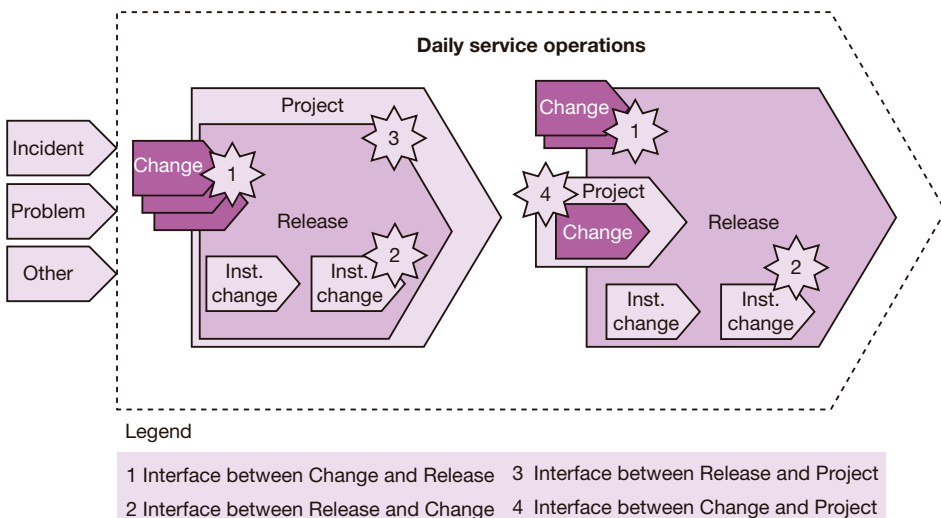


Figure 5 Relationships in the daily operations scenario

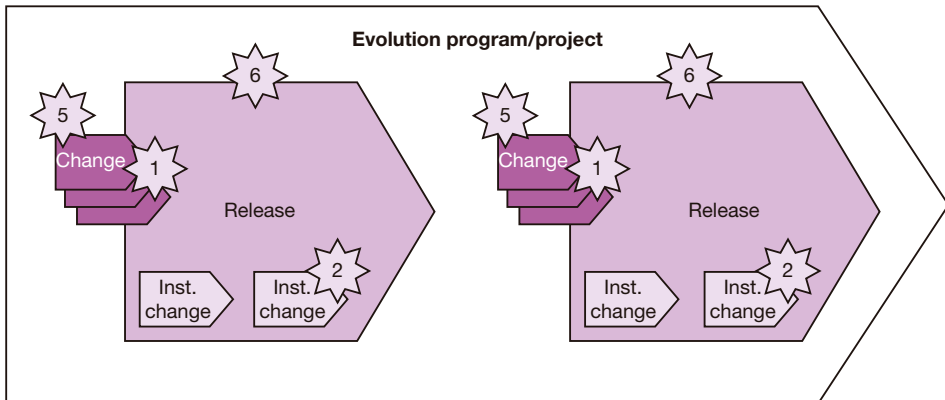
coordination and care in order to assure proper implementation by means of a dedicated project.

- **Change to project (4)** - A change (for example a major change) needs additional coordination and care in order to assure proper implementation by means of a dedicated project.

In addition, another scenario has been identified where relationships between processes may be activated. This is the scenario depicted in figure 6 where programs or projects will develop, and will implement business or IT requisites with impacts such as new or modified services. In this case, the program/project will interact with the service management processes, including change and release management. The interfaces to explore are therefore:

- **Changes to releases (1)** - Changes are deployed by means of a release.
- **Release to changes (2)** - Specific changes are issued in the release management process in order to install and activate a release.
- **Project to changes (5)** - Projects issue changes in order to implement specific requisites.
- **Project to releases (6)** - Projects cluster changes into releases, using release management techniques and principles to deploy and make IT deliverables operational.

Relationships 1 and 2 are present in both scenarios and it should also be noted that scenarios may coexist. Effectively, it is usual to have scenario one active for daily changes and the management of minor releases, and scenario two for major evolutions, managed as projects.



Legend

5 Interface between Project and Change	1 Interface between Change and Release
6 Interface between Project and Release	2 Interface between Release and Change

Figure 6 Relationships in the context of a program/project scenario

In the following sections we will analyze the 6 interfaces in more details.

EXPLORING INTERFACES: CHANGE AND RELEASE MANAGEMENT

In the overview of relationships, we have identified two interfaces between change and release management. The first one, interface 1 in figure 5 and figure 6, is described in ITIL:

changes are included in releases and deployed accordingly. Let's analyze in more detail the lifecycle of a change, and its relationship with release (see also figure 2). Changes are approved according to change management approval steps. The following steps, "design and development of the change", are performed through change and/or project management activities (this normally include some level of testing, typically regarding each single change). Some types of changes, for example configurations of software or substitution of hardware, may be completed through change management process steps. Others, typically those affecting applications, are better tested and implemented together, bundling them in a release and using the release management process principles and activities to complete the changes. The steps to be performed are, therefore, indicated by release management. For example, the service V-Model can be used to define the levels of testing: service release package test, service operational readiness test, service acceptance test. The bundled changes are synchronized with release statuses so, for example, when the release passes the operational readiness test, the changes included are commonly "Approved for Warranty". After deployment, see interface 2 described in the following paragraph, the "Review and Close Deployment" step of release management also completes the "Review and Closure" step for the included changes.

If a release is a means to deploy changes, then some steps of release management may profit from the change management process to be performed. This is interface 2, in figure 5 and 6. A release needs to be installed in several environments during the release management process. For each installation an RFC can be used for the target environment. A standard change is well suited to manage this type of request where the approval phase of the change can be linked to specific service acceptance criteria depending on the target environment/test phase to be executed and the type of release (minor, major or emergency). Also, the approval authority may differ, according to these drivers. The execution of these changes practically implements some of the activities of the release management process, namely those related to the Deployment (applications, data, information and infrastructure), as depicted in figure 3. Also remediation actions, such as the back-out of a release, can be performed inside an installation change (for example if an application is not running properly after installation) or by means of explicitly dedicated change (for example the installation of a previous version if test results are not satisfactory).

Table 1 depicts some possible correspondences between a specific type of release to be installed in a specific target environment and the related type of change to be utilized. Major releases will contain major evolutions/transformations but also bug fixes. Minor releases will include smaller evolutions and bug fixes too. Evolutions and fixes will be managed by means of changes included in the release for implementation and deployment. Installation changes are dependent on impact and urgency of the release but also on the target environment. For recurrent installations (such as weekly minor releases, with bug fixes and small evolutions) specific standard change procedures may be defined.

Change and release management may be implemented with different tools. The change management tool is normally part of the service management solution, while some release management functionalities may be supported by the software lifecycle management solution, including deployment (software distribution features). In the case of separate tools, it is important to implement the following interfaces in order to facilitate seamless and efficient processes:

- visibility of changes, as well as of their attributes and details, to be included in the release with the possibility to link them to the release record

- automatic update of the status of changes included in a release, based on the status of the latter
- automatic creation of the installation changes linking all the release information and its included changes

Release	Target environment	Installation change
MAJOR	Acceptance test	SIGNIFICANT
	Production	MAJOR
MINOR (weekly scheduled)	Acceptance test	STANDARD CHANGE
	Production	STANDARD CHANGE
MINOR (not scheduled)	Acceptance test	MINOR
	Production	SIGNIFICANT
EMERGENCY	Production	URGENT

Table 1 Sample of correspondences between releases and installation changes

EXPLORING INTERFACES: CHANGE AND PROJECT MANAGEMENT

The first interface we are going to explore is number 4 in figure 5, the interface between change and project. Some changes may be relevant in term of associated risks, costs and/or work effort. In such a case, simple change management process activities (see figure 1) may be not sufficient to guarantee the expected results. Since a change is a transformation, project management processes and techniques may be used to achieve this. There are two practical ways, which may coexist, to define when a change should be managed as a project: use of thresholds and human judgement. One of the most common thresholds to be used is the effort or equivalent cost (estimating all resources, included company's employees by means of standard or actual daily or hourly costs). An example of how to proceed using this type of threshold is depicted in figure 7.

The start of the diagram in figure 7 is normally the domain of the change management process, but having more than a single entry point for requests is probably unavoidable in medium to large organizations. The project management office, change management and demand management are usually the functions that will process the majority of evaluations and, therefore, they should be aligned as far as the drivers and thresholds are concerned, interacting and communicating as required in order to agree on doubtful cases.

The sample threshold used in figure 7 to define how to manage the RFC is forty man days, which corresponds to many actual situations we have encountered. These forty days can refer to either the internal or external resources needed to implement the initial request, with reference to the organization where the process takes place. This is not the only choice to be made as far as the process is concerned. Of course, a complete PRINCE2 structured process is too heavy for a 60 man days requirement and it is wise to consider other different project management approaches, with a level of structure adequate for the complexity and weight of the objectives. In figure 7, two possible models have been identified: the standard project model corresponds to the full PRINCE2 model (see figure 4) and is applicable to large and complex tasks; the light project model is a simplified process (see an example in figure 8, a simplified project management model based on PRINCE2 for single stage projects¹).

¹ It is not the aim of this article to fully explore the simplified model. Selected PRINCE2 processes shall be customized in order to provide a simple Project management framework, suitable for single stage small to mid-sized Projects.

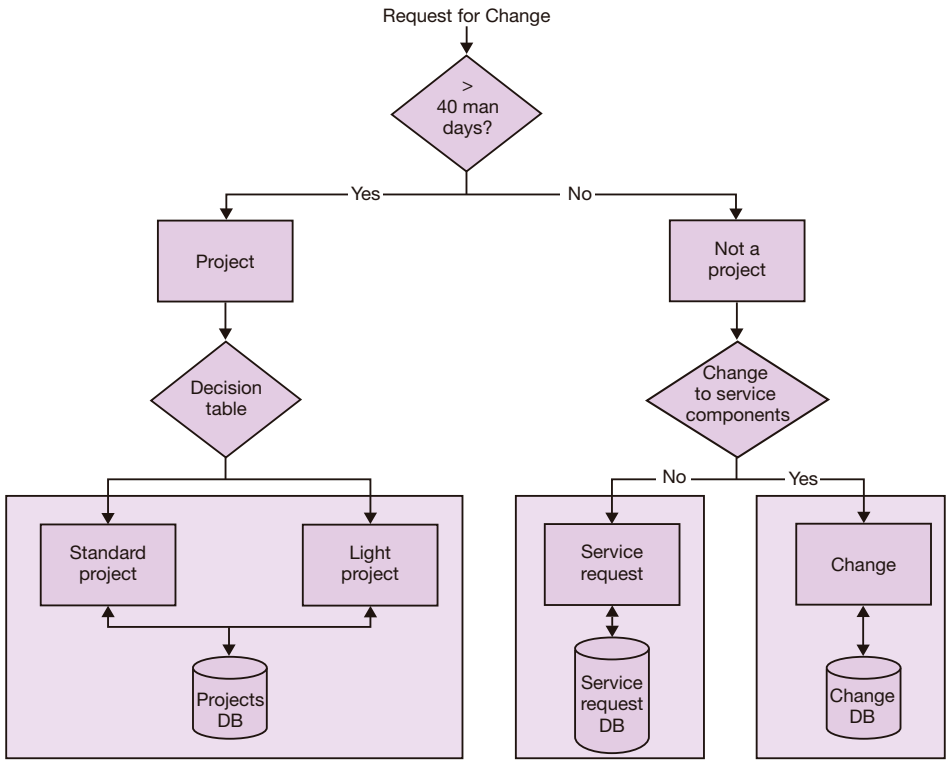


Figure 7 Translating requests into projects, changes or service requests

Making the choice between the “standard” or “light” model can be supported by a decision table. An example of such a table is shown in table 2, where the decision to opt for a model could be taken when two or more drivers are supporting it.

Driver	Light project	Standard project
Effort	40 < man days < 200	> 200 man days
Total cost	< 300k euro	> 300k euro
Strategic Business Plan initiative	No	Yes
Risk	Low to moderate (damage < 300k euro)	High (damage > 300k euro)
Others...		

Table 2 Drivers to choose a model for project management

When it is decided to manage a RFC as a project, the RFC should be closed accordingly and a project opened, filling the initial project mandate with RFC information. This is graphically illustrated in interface number 4 in figure 5.

The other interface we have to analyze in this section is the one from project to change management (number 5 in figure 6). This interface is probably the more complex and is

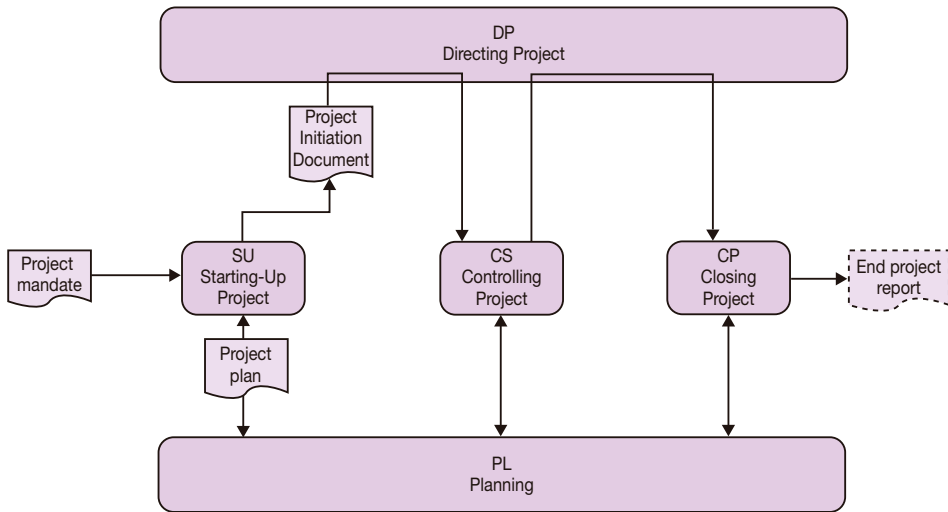


Figure 8 A simplified project management model based on PRINCE2 (Source: OGC)

activated when a project needs to modify the design of any existing service component, or when a new service is designed. When there is such a need, it is usual that certain change management activities are performed collectively in the project tasks. With reference to figure 1, it is common to see assessment, evaluation, planning, authorization, implementation and review tasks performed through project activities. When this happens and activities are executed by an ad hoc project organization not involving the change management functions, then there is a very high risk that there will be conflict between the project and change management objectives in a later phase of the project. This may happen because project management needs to pay attention to the project objectives, especially the timing and costs constraints, while change management, as part of service management, focuses upon the continued functioning of services in line with their requirements. A situation can arise, therefore, where project management are pushing to release a new component, whilst change management are resisting because of either late communication, incapacity to perform the necessary tests or inadequacy of the components to be released. This situation would be detected at the point when project management try to release deliverables to the production environment, as this task is normally under the control of change management.

In order to avoid this situation, there are some mandatory decisions to be taken and some alternatives to be chosen. The mandatory decision involves the change management function and early roles in the project. The alternatives derive from the relationships between the processes (project and change management).

The first alternative is to have an overall project plan where changes to IT services are identified and related activities are planned and updated according to the change management process. All information about these activities, such as timing and costs, should flow between the change and project domains. In this scenario, the change management process and function play an established role for the specific tasks which are part of the

changes, and the project activities are simply “embedding” and coordinating them. We can, therefore, say that project and change coexist: Project management determines changes which are executed by change management and taken into account by project management. The following is an example: the PRINCE2 “MP1 Accepting a Work Package” may be performed through the “Assess and Evaluate Change” and “Authorize Change” activities of change management process when a work package is related to one or more changes.

A second alternative is to manage the typical change management activities as normal project activities, using change management mainly for administrative aims if necessary (for example for homogeneous recording and tracking of changes). This means, for example, that changes may be collectively assessed by a project activity where the change management function is involved and where results are passed to change management in order to update change record information and status (if change management decides to formally open changes). The authorization may be managed as a project activity so that when changes come to the change authority for approval, this is only an information and administrative task, given that the records have been pre-approved by the project management. Implementation tasks (Work Orders generated in change management) are also managed as project tasks. Choosing this second scenario means that project management is given a more relevant role. Change management roles are involved in project management activities, but change management is not executed as an independent process, but rather it is updated based on project activities and results. An example is the following: different work packages (corresponding to changes) are requested by a “CS1 Authorising a Work Package” activity of the project. The change management function takes part to the “MP1 Accepting a Work Package” activity of the project where Work Packages are approved. Accordingly, as a result of these activities, the change management function will create and update the status of changes, if necessary, but without newly executing any assessment activity.

Given these two scenarios, the preferable one is the first which clearly recognizes the specific roles and scope of both processes. The problem arises when change management is not effective and/or efficient and is, therefore, unable to respond to the necessities and constraints of project management, or when the priorities of projects are not addressed adequately by change management. In such cases, the second alternative should be carefully evaluated and could emerge as more suitable.

From an organizational point of view, the responsibilities should be clear between the functions and roles that are managing project management and those managing change management, and communication channels should be well established. ITIL version 2 was not clear about organizational matters (only the main process roles, such as change manager, were identified, but no information was given on how to organize functions), but ITIL V3 has added some details as depicted in figure 9. In some IT organizations the Program and project management depicted in figure 9 is a staff function of the CIO, working for the Application development department as well as for Operations. The scenarios we have illustrated have detailed the “RFCs from projects, formal handover of CIs” relationship as depicted in gray in figure 9.

From a tool point of view, as we have previously remarked, the change management functionalities are normally part of the service management suite. Here it is not common to find features for cost management such as timesheet or expenses management, or sophisticated planning functions typical of project management tools. Having a full view of activities and the workload generated by change and project management is key information

for good decision making and effective planning. Therefore, if a single solution does not cover all all functions it will be important to implement some interfaces between the project and change management tools:

- alignment of common master data (such as resources, projects)
- alignment of forecast data regarding changes (activities with timing, resources, workload, status, details)
- alignment of actual efforts and costs

EXPLORING INTERFACES: RELEASE AND PROJECT MANAGEMENT

The first interface we are going to explore is the one between release and project management, depicted as number 3 in figure 5. This interface is, in some aspects, similar to the one between change and project management (number 4 in figure 5). A release, which is a collection of changes to be deployed, may be required by project management, or it can be generated by service management. The first case is out of scope for this specific interface, given that the release is already part of a project (we will explore this interface later in this section). Service management deals with releases periodically (for example for regular updates or bug fixing), packing changes into them, or for emergency releases. This latter case will not generate, by definition, any project. Depending on the type of changes packed into a release, the need to manage a specific release as a project may arise to further improve the likelihood of achieving the expected results. But in a well established service management organisation with properly set transition management functions (especially, with reference to figure 9, service knowledge management, service release and deployment, service test management) and having mature release management processes in place, this will probably not happen as these processes should be sufficient to guarantee a well managed transition of the release itself.

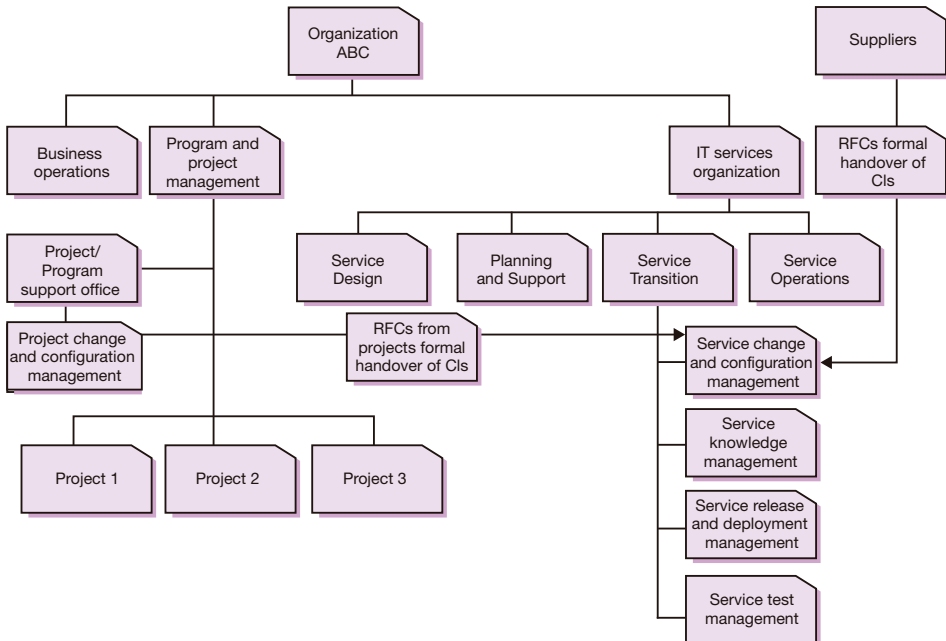


Figure 9 ITIL V3 example of Service Transition organization (Source: OGC)

In organizations where service management is not mature or adopted, at least as far as release management is concerned, then when a release is requested to be installed, service operations should evaluate if the usual practice and normal activities will be sufficient to deal safely with it. Again, we can rely on human experience and judgement to make such a decision, or we can define some thresholds. A useful one, if change management is in place, is the number of changes included in the release, by type (for example major, significant, minor). An example of the use of this type of driver as a threshold is given in table 3.

Driver	Threshold
Major changes	> 0
Significant changes	> 3
Minor changes	> 10

OR → project

Table 3 Example of use of number and types of changes as a threshold to manage releases as projects

If change management is not in place, the estimated effort (as with the interface between change and project) to complete all release management activities (planning, testing, deploying, and review) may be adopted and a threshold of forty man days may be reasonable.

Once it is agreed that a release has to be managed as a project, another decision may be necessary, i.e. the choice of project model. Again, it is possible to use a decision matrix similar to the one already described in table 2. Nevertheless, the probability that a full PRINCE2 approach is adopted is, in practice, very low, because such a high impact release would have probably been managed earlier as a project (hopefully, not when it is the time to deploy it!). So a light project approach, as described earlier (see figure 8) is probably preferable.

The next and final interface to explore is the one between project and release management, number 6 in figure 6. Projects, when dealing with IT applications and software, normally organize deployment in releases which implement some or the full set of requisites. If service management is in place (see interface 5 in figure 6), these requisites have already been mapped into changes and interface 6 practically corresponds to interface 1 (between change and release management). Project management knows from change management when the changes will be ready for testing and deployment, and requires release management to plan for these activities (this task is normally performed together by the project and release management organizations). All defined and supplied schedules will contribute to the project plan update.

Another possible scenario is similar to the one already described in alternative 2 for the project to change management interface. In this case, project management takes the lead in making the planning decisions, and change and release management are aligned to these project management decisions. The involvement of change and release management functions as early as possible is an important driver for the success of the project, because it ensures that all aspects of service transition will be considered and the necessary resources from change and release management functions will be involved as and when required.

In this scenario, not only the change activities but also the release activities are managed as part of the project. Planning, testing deployment, etc., are planned by the project

management organization and are executed according to project plans, which are defined in conjunction with change and release management staff. These staff will also ensure that the release management principles, such as the V-Model for testing, are applied to the project and will guarantee that other releases managed outside project boundaries will take place in compatible slots with the necessary resources.

As suggested for the interface between project and release management, the preferable scenario is the first one where project management interfaces to a mature service management organization, working closely with it to define the timing of activities according to its processes and letting it taking care of activities.

Finally, as already mentioned, project and release management tools are often distinct, the latter being embedded in software lifecycle management solutions. In practice it is often possible to manage the interfaces between the supported processes manually. Further improvements may arise through the implementation of an interface to automatically update project plans with release information (releases and their related activities may be loaded as project activities so that any change to the release schedule and allocated resources is immediately reflected in the project plan).

CONCLUSIONS

In this article we have clearly identified and explored the relationships between change, project and release management, assuming ITIL and PRINCE2 as reference frameworks to determine the scope and details of the processes. To this aim, a simple graphical model has been created.

The relationships between the service management processes (change and release) are traced in ITIL, especially in the latest version 3 of the framework, but not fully developed to be practically implemented. The article contains suggestions and practical examples useful to this aim.

Relationships between projects and the other processes are suggested as being necessary in the frameworks, but are not detailed even at a very rough level. They are more complex than those relationships between change and release management, primarily because they involve powerful but independent functions of the organization. Different implementation scenarios for the relationships are possible, depending on the maturity and influence of the corresponding functions, with project management being traditionally better established in application development units, whilst service management functions tend to be located in more of an operations context. The way the processes are interfaced can also be viewed as an indication of how the balance between stability versus responsiveness is established in the IT organization.

We have learned that changes should be managed as projects if their size and associated risks become relevant; less common is the need to manage releases as projects, as they can normally be addressed by release management processes or are already included in existing projects. Projects generate changes to implement specific project requirements and may use release management to test and deploy them.

The scenario where project management leverages service management processes and organizational units to develop requirements into changes, group them into releases and

have them implemented, tested and deployed is the preferable one for organizations where service management is mature; this optimizes the use of resources and the overall scheduling of activities, whilst minimizing the risks of adverse impact. But it is not the only possibility. Project management can also embed and substitute the typical activities of change and release management, applying the principles of the two disciplines and keeping them aligned with feed-back information.

Whatever solution is adopted, based on organizational structure and maturity, the secret of the success of the interfaces between project and change and project and release lies in good communication and involvement between the project management organizational units and the service management functions. An effective communication channel should be established as soon as possible between project management and project activities, and the timing should be compatible with change and release management principles and procedures.

In the practical implementation of interfaces, some issues must be addressed or they may threaten success:

- It is a pre-requisite to have mature service and project management processes and functions.
- The correct balance must be achieved between project management and service management responsibilities and spheres of influence.
- The interfaces must be designed in detail and based on real context, maturity and established processes and organization.
- All resources must be required to act according to the newly designed scenarios.
- Project and service management tools should be integrated to improve process efficiency and effectiveness.

Christian Cantù (Italy), economical engineer, is co-founder and partner at iCONS – Innovative Consulting, a specialist in IT service management and project management, and a member of ITSMF Italy.

Maxime Sottini (Italy), electronic engineer, is founder and Chief Executive Officer at iCONS – Innovative Consulting, a specialist in IT service management and project management, board member of itSMF Italy, and itSMF International IPESC and IQaC delegate for itSMF Italy and itSMF Italy responsible for Certifications and Qualifications.

REFERENCES

- Office of Government Commerce (2007). *ITIL: Service Transition*. London: The Stationary Office.
- Office of Government Commerce (2007). *ITIL: Service Support*. London: The Stationary Office.
- Office of Government Commerce (2007). *ITIL: Service Design*. London: The Stationary Office.
- Office of Government Commerce (2005). *ITIL: Managing Successful Projects with PRINCE2*. London: The Stationary Office.

