APM / INCOSE UK

Systems Thinking Specific Interest Group

Fusion Point Guidance – Interdependency Management

Issue 1.0 April 2017

Target Audience:

Primarily aimed at:

- *Systems Engineers.*
- *Programme Managers.*
- *Project Managers.*
- *Business Change Managers.*
- *Project Support Staff.*

Who is the Guidance for?

This guide is primarily aimed at System Engineers (**SE**), Programme Managers, Project Managers (**PM**), Business Change Managers and Project Support Staff who need to develop and deliver a robust plan based on a firm understanding of their delivery interdependencies.

What is Interdependency Management?

Definitions

Interdependency management describes the identification and management of dependencies outside the direct control of programme/project manager, e.g. between projects within a programme. The dependencies (inputs/outputs) will typically take the form of information shared (e.g. a decision, data) or a physical item (e.g. land, hardware). They can also be a business outcome or a benefit. These dependencies can exist in many situations.

Within the INCOSE Systems Engineering Body of Knowledge (SEBoK), the term Interdependency management is not used. Instead the terms **Systems of Interest** (SoI) and **Systems Context** are used. SEBoK defines SoI as “an interacting combination of elements to accomplish a defined objective”. The idea of a System Context is used to define a SoI and to identify the important relationships between its internal system elements, the wider system elements it works directly with, and the environmental conditions which influence it in some way (e.g. ethical, political and social).

The APM Body of Knowledge (APM BoK) definition of interdependencies states this is “An aspect of programme and portfolio management. The management of dependencies between projects, and projects and business-as-usual activities.” The APM BoK definition of dependencies expands on this, “Something on which successful delivery of the project critically depends, which may often be outside the sphere of influence of the PM, for example another project”. This is expanded on in the Managing Successful Programmes guidance in which there are three types of dependencies[1]:

- Internal dependencies that are inside a programme – These are dependencies that can be managed and contribute to the programme outcomes and benefits
- Intra dependencies on other programmes or projects – These are dependencies that are external to the programme but within the organisation’s programme and project environment
- External dependencies outside the programme environment – These dependencies exist beyond the organisation’s programme environment, and potentially into other organisations. Examples might include partner organisations or strategic decisions.

[1] The labels internal, intra and external reflect a perspective scoped as an organisation’s family of programmes (often called a ‘portfolio’). Different labels would be appropriate if the boundary of the System of Interest is different.

This document represents the thoughts and conclusions of the Systems Thinking SIG and not necessarily the views of the APM or INCOSE UK. It is intended to assist Project, Programme and Portfolio Management and Systems Engineering practitioners wishing to explore concepts and ideas around Systems Thinking in P3M and to stimulate discussion on the subject. Feedback on the contents of this paper should be sent to the Systems Thinking SIG (SystemsThinkingSIG@apm.org.uk). It therefore does not constitute any formal position (or liability arising) on the part of the International Council for Systems Engineering (INCOSE), INCOSE UK Ltd. or the Association for Project Management (APM), nor should any formal endorsement by these bodies be inferred.

PRINCE2© definitions from a project perspective are:

- Internal dependency – This is a dependency between two activities within a project, under the control of the PM
- External dependency – This is a dependency beyond the control of the PM. Examples include delivery of a product from another project and a decision from programme management (e.g. funding).

Identification of the interdependencies can easily be confused with assumptions and constraints. It should be possible to frame an interdependency in terms of the following statement,

"We are dependent on X to deliver Y on date Z in order for <u>us</u> to deliver AA".

> *Interdependencies are best formed where there is certainty of scope, delivery approach and timeline.*

An assumption for Project A could be Project B is delivering X on date Z and we can use it. The assumption should then be tested to see if the statement is valid and increase detail to define the interdependency. PRINCE2© definition of an assumption is "A statement that is taken as being true for the purposes of planning, but which could change later.".

A constraint would be a directive that Project A <u>must</u> use product X delivered by Project B on date Z. The exact nature of product and timing of the interdependency can be a negotiation but there is a directive to use product X. The PRINCE2© definition of a constraint is "The restrictions or limitations that the project is bound by."

Progressive Definition of Interdependencies

In many situations (e.g. in startup or in complex projects), interdependency agreements will need to be formed and baselined even when there is ambiguity in the scope, delivery approach or project timeline. These dependencies should remain under constant review and acknowledged as a source of risk with an associated management plan (e.g. mitigation or contingency).

In extreme circumstances, the programme or project may be re-scoped to remove (difficult) interdependencies to reduce risk, e.g. bring the delivery of the interdependency within the control of the project. This is particularly important for Long Lead Items (e.g. Aircraft Landing Gear) where manufacture of those items may need to start before other (interfacing) designs are complete.

Interdependency Identification

The INCOSE SEBoK notes "a system context can be used to define a SoI and to capture and agree on important relationships between it, such as the systems it works directly with and the systems which influence it in some way. When this approach is used to focus on part of a larger system, a balance of reductionism and holism is applied. This balance sits at the heart of a systems approach. A system context provides the tool for applying this balance, and is thus an essential part of any systems approach and hence, of systems engineering as well. Thus, as a first order output, the SE won't identify interdependencies but rather the systems and sub-systems that are required for the whole to function (and the actors). These can usefully be expressed in Product Breakdown diagrams which can be applied at system of systems, system and sub-system levels of abstraction. The diagrams should reflect both specialist (output) products and management products. This ensures the systems perspective takes into account the delivery system(s) (e.g. project).

There are also many techniques to help identify the system/sub-systems. Many of these encourage the exploration of different viewpoints, or sets of viewpoints. The techniques will vary depending on the sector that you work within to structure thinking. Examples include:

- POTI – Process, Organisation, Technology, Information
- PESTLE – Political, Economic, Social, Technical, Legal, Environmental
- CATWOE – Customers, Actors, Transformation process, World view (Weltanschauung), Owner(s), Environmental constraints

An SE will progress on to explore **relationships between systems.** These are not interdependencies and are dealt with by defining interfaces and flows between systems elements. A flow may indicate a dependency so is a rich source to aid identification, noting interdependency is between delivery systems. SEs and PMs must work together on this to ensure there is full and comprehensive coverage of both identified and potential future dependencies.

> *Start by identifying systems, sub-systems and system elements and their relationships. Consider different viewpoints.*

This is where a diagram showing **System of Interest (SoI)** and **Wider System of Interest (WSoI)** can be useful, when applied to the project or programme itself. The project is assumed to be responsible for the integration of everything on the page (this may take the form of doing it or checking that it has been done). In Figure 1, the business objective of providing euros at cash machines, in addition to pounds, is used as an example. The first step in creating the diagram is to identify all the actors and systems that are involved and identifying relationships/flows that do, or will need to, exist. At this point, the scope boundary is frequently drawn to represent what the project can directly control. As an aid to identification of interdependencies, associated change activities can be added. This may lead to the scope boundary being tweaked.

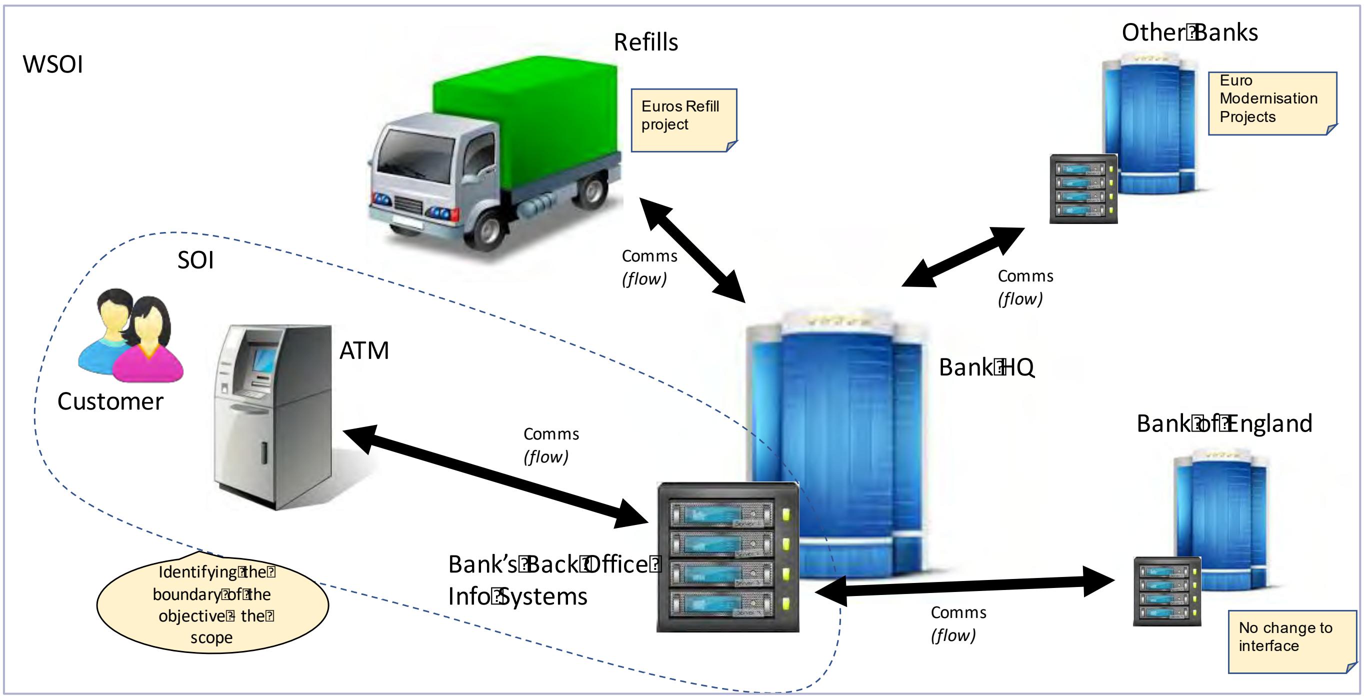

Figure 1: Illustration of SOI/WSOI & Identification of Interdependencies

An interdependency can be needed to realise benefit, rather than part of the end-to-end capability. For instance, shops accepting euros would be part be of the realisation of the benefit for the customer, not part of the end-to-end capability to supply euros.

N-Squared matrices[2], can be used to identify interfaces between systems and, if systems/sub-systems/components are tagged by who is delivering them, then likely dependencies can also be identified. The matrices are also used to group and then define system boundaries.

Stepping from the business problem to proposed solution is often called a "**twist**". Solution "**trading**" will impact the SoI and WSoI (i.e. scope boundary) and impact on interdependencies. Further refinement of delivery scope can

[2] N-squared matrix is a method to group and sort functional or physical interfaces between system elements. It is used to systematically identify, define, tabulate, design, and analyze functional and physical interfaces.

occur when agreeing the delivery approach causing further change. Evidence of change will be visible from the impacts on the work and product based plans.

In a programme environment, a dependency network is usually drawn out to show how projects depend on each other. These support the aim of a programme to realise expected benefits, through the achievement of desired outcomes.

Interdependency Management

Having identified an interdependency, it can be managed by using the following steps[3]:

- **Evaluate** and **document** the interdependencies the project will have (e.g. type (inward/outward), source, specification, timeline)
- **Align** and **agree scope** of the interdependency to ensure clarity and complementary specifications with the source of the interdependency (e.g. if the source is producing a square peg then the project needs to deliver a square hole)
- **Align** and **agree timelines** through specifying when required and aligning with source of the interdependency to confirm achievability (including resource)
- **Signoff dependency agreement** to include commitments, responsibility and accountability. This can eventually include signoff of the completion of the interdependency
- **Monitor** and **control** throughout life of the interdependency.

Typically, the project manager is responsible for ensuring that interdependencies are

- identified, documented and agreed then
- progress monitored and changes controlled.

In most circumstances, the Systems Engineer will do most of the work to identify and document the interdependencies and then intervene where there are changes or technical ramifications of non-adherence.

The following diagram (Figure 2) illustrates an interdependency management process that a programme might use to understand the impacts of any "threat", such as an interdependency being late or off-spec, and links in to the programme's risk management process.

[3] A good example of collaboration is the 'RECDEL' approach, used on the Cassini mission to Saturn, where the approach to managing the project was essentially (just) Interdependency Management. However, there is a limit to how far the approach can be scaled up.

https://trs.jpl.nasa.gov/bitstream/handle/2014/17977/99-1430.pdf?sequence=1.

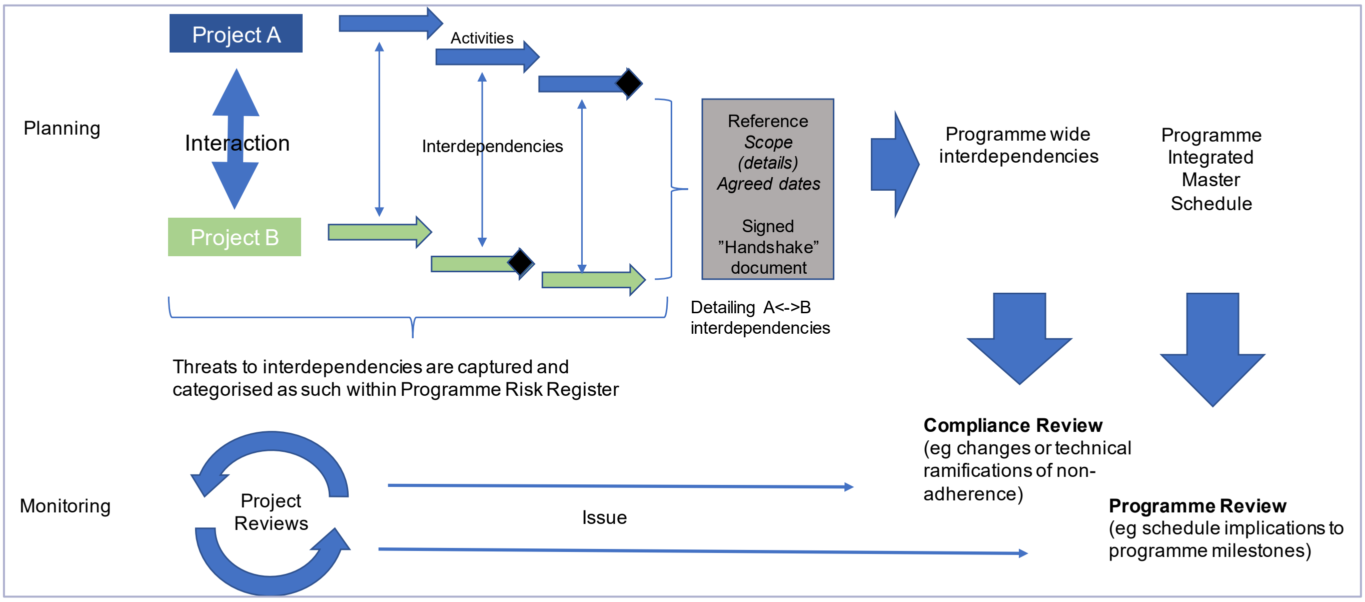
Planning
Project A
Interaction
Project B
Activities
Interdependencies
Reference
Scope
(details)
Agreed dates
Signed
"Handshake"
document
Programme wide interdependencies
Programme Integrated Master Schedule
Detailing A<->B interdependencies
Threats to interdependencies are captured and categorised as such within Programme Risk Register
Compliance Review
(eg changes or technical ramifications of non-adherence)
Monitoring
Project Reviews
Issue
Programme Review
(eg schedule implications to programme milestones)

Figure 2: Example of an Interdependency Management process

Why is Interdependency Management important?

Interdependencies represent potential uncertainty and lack of direct control, and are therefore a source of risks to the success of the project or programme. Therefore, the PM (and SE) responsible for delivering the project or programme must monitor and where able, ensure delivery of the interdependency.

What are the issues?

Once interdependency is understood, it can be documented, agreed and managed. Where necessary, an associated risk can be identified, assessed and managed. The difficulties are:

- Identifying the interdependency
- Agreeing ownership
- Depending on position in lifecycle, it can often be difficult to specify exactly what it is and need by date (see note of the progressive definition of interdependencies).
- Without a stable complete definition, formalising the interdependency agreement ("locking") with need by dates and assigned resources will be problematic. Progressive baselines can be used but typically are not supported by firm resourcing commitments or delivery dates.

> *SE "focus is placed on controlling the interfaces between system elements and external systems" in order to help address ambiguity, fuzziness and complexity. This focus aligns with the need to address the above issues.*

How can Fusion between SE & PM Help?

The identification, development and management of interdependencies is at its most effective when systems engineering and project management skills and tools are used together and a collaborative working relationship is formed between the SE and PM. It's only once the interdependencies (and by implication, scope and delivery approach) are jointly understood can the PM and SE "divide and conquer" through focus on their core functions (e.g. resource management, design). The following step-by-step guide illustrates this:

- **Identification of System Elements** – Together build a common understanding of systems and sub-systems through exploration of viewpoints to identify system elements (*This will also inform stakeholder management planning*). Hierarchical relationships between systems, sub-systems and system elements can usefully be described through product breakdown structures. This step should consider both delivery systems (e.g. business units) and product systems (i.e. outputs).

- **Identification of Relationships –** Exploration of Scope of Interest / Wider Scope of Interest through rich pictures can be used to identify relationships and boundaries (used to inform the project scope). The detail of the relationships can be explored through N-Squared matrices and causal loop (typically used to explore impact of changing one system on the other systems in the SoI).
- **Identification of Dependencies and their Impacts –** The **scope boundary** and **delivery approach** will affect what is an interdependency, the nature, when is it needed and how will it be delivered.
- **Trading –** Through use of Work (including build logic) and Product Based Planning, informed solution trading can occur to minimise dependencies and strengthen management of (i.e. move dependency from outside to inside the project scope).
- **Agree dependencies –** The SE can now focus on the specification of the dependencies and the PM can focus on the timing and ensure dependencies are resourced to ensure the project delivers to the agreed time, cost and quality. The SE and PM must work closely together as the specification can impact on timing and vice versa.
- **Monitor and control –** The PM will manage the delivery of dependencies against the plan and the SE will focus on ensuring dependencies are sufficiently well defined to minimise integration issues and unwanted emergent properties.

Signs of success:

- *Collaborative partnership between SE and PM*
- *Collaboration between projects and business areas*
- *Common understanding of systems, relationships and dependencies in the problem and solution space*
- *Interdependencies based on agreed stable scope and delivery approach*
- *Agreed, robust and resourced interdependency agreements*
- *Focus on interdependency detail to avoid unwanted integration issues and emergent properties*
- *Associated risks are identified, assessed and managed*
- *Excellent working relationships with suppliers and customers of the interdependencies.*

Acknowledgments and References

Managing Successful Programmes, 2011 Ed, London:TSO

Managing Successful Projects with PRINCE2©, 2009 Ed, London:TSO

APM Body of Knowledge, 5th Edition

PRINCE2© and MSP© are Registered Trade Marks of AXELOS Limited

Systems Engineering Body of Knowledge (SEBoK) v1.7

Systems Engineering Handbook, Fourth Edition – INCOSE-TP-2003-002-04 2015 John Wiley & Sons, Inc