APM / INCOSE UK

Systems Thinking Specific Interest Group

Fusion Point Guidance – Requirements Management

Issue 1.0 March 2017

1 Who is this guidance for

This guide is primarily aimed at Project Managers and Systems Engineering Managers who will be producing their respective plans for requirements definition and management, and the procurement and contract managers developing the contracts.

This guide will also provide useful requirements management information about the benefits of SE and PM to project sponsors in terms of how their project will be executed.

Target Audience

Primarily aimed at:

- *Project Managers*
- *Systems Engineering Managers.*
- *Procurement/Commercial/Contract Managers*

Also provides useful information to:

- *Project Sponsors*
- *Other PM and SE professionals.*

2 What are Requirements

Strong contracts require a robust specification of the work to be undertaken (defining outputs in particular) and the relationships with other areas of the project. Specifications that are fully aligned with (and derive from) user and system requirements must therefore be integrated with the wider project. The test and acceptance criteria will not only provide contractual confirmation for payment, but will also be aligned with project schedules and the vertical verification activity. Passing requirements information through the contract may also provide a wider contextual background that would help suppliers make robust decisions affecting delivery.

Requirements come in a wide range of types, only some of which are functional – many vital requirements are non-functional and form a potential source of issues for contracts. Examples of these are legal and political dimensions, dependencies, workforce issues, and the whole point of people focusing on the product system (i.e. System of Interest) and not thinking widely enough about the delivery system (i.e. Project Enabling Systems).

True requirements

True/real/genuine requirements are those attributes of a solution that will make it fit for purpose. They exist as concepts, and may or may not be recognised and accurately documented.

They extend far beyond the technological focus of the project, and include human factors in the use of the solution, and cover the whole of the solution's life through to decommissioning and disposal.

Identification of real requirements can be extremely challenging, as there is a tendency to break down the solution into pieces, then focus on some of them to the exclusion of others.

Requirements

- *True (real or genuine)*
- *Documented*
  - *Functional*
  - *Non-Functional*
  - *Prioritised*
- *Specifications*

Systems thinking is essential to probing true requirements effectively, through exploring the whole scope of the requirements and ensuring that the requirements for the (potentially complex) interactions between elements of the solution are understood.

There are often legal and regulatory constraints to a solution, and compliance with these constraints are requirements too (e.g. British Standards, nuclear regulations). The standards themselves may be couched as requirements or specifications, commonly a mixture of both.

Documented requirements

Documented requirements are the attempted representations (usually textual and diagrammatic) of the true requirements, and are often referred to simply as "The Requirements". The documented requirements may differ from the true requirements in a number of ways:

- Being incomplete
- Being incorrect
- Being incoherent (internal contradictions)
- Specifying a solution (quite possibly inappropriate), not defining the true requirements

Since documented requirements are normally the foundation of contractual terms, any of the above problems is likely to lead to contractual issues and project variations/change notes, causing delays and cost escalation.

Systems thinking plays a powerful role in correctly documenting the true requirements through identifying the complete scope, flushing out incorrect requirements, eliminating internal inconsistencies and avoiding solution specification rather than requirements definition.

The comprehensive view to requirements covers both:

- PRODUCT Requirements which define the "Performance" of the system of interest, and
- PROCESS requirements which define how the realisation (delivery) of the PRODUCT must take place.

Generally PRODUCT requirements are associated with the system of interest and PROCESS requirements apply to the enabling systems.

Both the PRODUCT and the PROCESS requirements must be met for a successful project.

Functional Requirements

Requirements fall into two categories:

- What the solution does (functional requirements)
- Everything else (non-functional requirements)

Functional requirements are the natural focus of customer attention in projects/programmes, as they define how the customers and solution users will operate once the solution is delivered.

Non-functional Requirements

Non-functional requirements are prone to being under-defined or even overlooked unless the project organisation is highly disciplined and professional, employing systems engineering skills. Key non-functional requirements to be considered include:

- Initial solution performance, and capacity for growth
- Usability, and training requirements for users
- Availability, Reliability and Maintainability (ARM)
- Security (both physical and cyber)
- Adaptability/scalability/longevity

Non-functional requirements are the domain of technical specialists, and **systems thinking is vital to their integration into an effective and efficient solution.**

Specifications

Specifications define the solution and its components. Whereas requirements drive the design process, the specifications result **from** it to describe the solution. The terms "specifications" and "requirements" tend to be used interchangeably, but this confuses the situation.

Requirement importance/prioritisation

Not all requirements are essential and requirements definition should also capture the importance of each requirement. The design process needs to trade off meeting requirements against cost and time – this requires confidence by the design team in being able to drop low-importance requirements to achieve project success.

A good understanding of the complete system and its interactions supports this.

Typically, requirements are classed as one of:

1 Mandatory/essential – must be satisfied
2 Highly Desirable – can be sacrificed under pressure
3 Nice-to-have – not essential at all, but would be beneficial

3 Why is Requirements Management important

Effective requirements management is essential to projects and programmes delivering outcomes that are fit for purpose i.e. achieve quality.

A project that delivers an outcome that is not fit for purpose, even if it is on time and budget, has failed, as making it fit for purpose subsequently will result in delays and/or overspend. A solution that is over-specified will usually cost more, and take longer.

At least 12% of project investment is wasted globally (PMI "Pulse of the Profession" 2016) and this percentage is often much higher for IT projects, in many cases due to poor requirements management.

Requirements Management provides the framework for:

- *Capturing, documenting and validating the requirements;*
- *Managing changes to requirements during the project;*
- *Verifying requirements have been met*

Contracts must support these

4 Where are the issues in Requirements Management

4.1 Requirements capture/definition

4.1.1 Critical Success Criteria - what will make the project a success/failure?

True requirements are often revealed by asking probing questions about the critical success criteria of the project, and what success looks like. **This system level thinking causes customers to take a step back from "specifying" mode into a holistic reflective mode.**

These high-level criteria give a good insight to what is really important.

4.1.2 Planning for acceptance from the start

Planning for acceptance of the project is often left until late in the life-cycle, and draws in many stakeholders who have often not earlier been consulted effectively. This results in a flurry of last-minute changes as their insights and consequent changes are taken on-board, with consequent delay and cost escalation.

SEPM helps:

- *Agree the critical success criteria, focussing the project team on a successful outcome*
- *Plan for successful acceptance right from the start*
- *Capture comprehensive, validated requirements*
- *Select the best PM methodology*

Contracts must reflect these goals

Systems Engineering disciplines encourage planning for acceptance from the start, encouraging the stakeholders to consider scenarios that may well have been overlooked idocumented requirements.

4.1.3 The role of the user in ensuring usability

An unusable solution is worthless. It is easy for designers and engineers to create an unusable solution that meets the functional requirements that is never-the-less not fit for purpose. A web-form for a contact centre that lengthened order call duration by 200% was naturally never used.

Usability is vital to solution success e.g. the Millennium Bridge in London. Only when an apparently-perfect engineering and aesthetic solution met the users was a serious design defect revealed. When first opened, bridge users found that it swayed slightly, and found themselves unconsciously walking in step with the sway, causing the bridge to resonate and the sway becoming uncomfortable. This phenomenon has been well-known for centuries, but forgotten by the designers – the dynamic interaction of users with the products hadn't been thought through.

Soft system tools and techniques involve users in the capture and validation of requirements, as well in the acceptance testing of the solution, minimising late changes from their insights. This is a significant factor in the popularity of Agile methodologies.

4.1.4 Requirements validation – project lifecycle choice

- **Waterfall**

  The principal weakness of pure waterfall lifecycles/methodologies is that the documented requirements are only validated at the end of the project, when the completed solution undergoes acceptance testing. This almost guarantees they will be late and over-budget as flaws in the documented requirements are revealed.

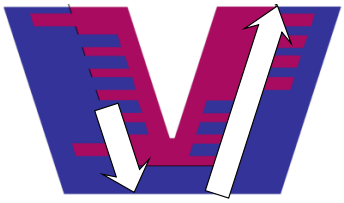
- **Agile**

  Agile methods address this issue through the close-knit integration of the customers into a team in which requirements are validated continuously. The risk is technical blind-alleys, and consequent rework.

- **Multi-phase**

  Hybridising these 2 approaches avoids the primary risks of either, and is consistent with the core principle of most project lifecycles i.e. multiple phases/stages with gates separating them. This approach allows Agile flexibility in the Proof of Concept and Technical Feasibility phases, with the tight control and efficiency of Waterfall for full development.

4.2 Requirements change management – critical provision in contract



As any project takes finite time, there will be changes to the requirements during its lifetime, and these must be managed effectively and efficiently. **The contract must make comprehensive provision for this from the start.**

4.2.1 Issues creating changes to requirements

The principal external factors driving requirement changes are:

- Changing commercial and organisational environment
- Changing environment (PESTLE)

Key factors internal to the project are:

- Design trade-offs between time, cost and capability
- Discrepancies between the documented and true requirements revealed as more stakeholders become involved

SEPM helps:

- *Identify the full scope of changes needing management*
- *Impact-assess changes thoroughly*
- *Ensures the post-change solution still meets the requirements*

Contracts must implement fair, fast and project-success-driven change management around requirements

4.2.2 Impact assessment

Whenever a change to the documented requirements is proposed, the first and most important step is a rigorous assessment of its impact on the project. **Systems engineering disciplines ensure that the impact assessment is exhaustive, and there are no unforeseen implications of the chan**ge. In its absence, changes are often approved that have unpleasant consequences.

The reverse is also true – a design change may result in requirements no longer being met, so it is important that such design changes are assessed in terms of requirement satisfaction. Again, **systems engineering disciplines ensure that the impact assessment is exhaustive, and there are no unforeseen implications of the cha**nge.

4.3 Requirements verification and project acceptance

SEPM helps:

- *Design for acceptance*
- *Efficient yet effective acceptance of complex solutions*
- *The building-in of effective quality assurance*

Contracts must implement effective, fair and project-success-driven quality management

Project success demands verification that the requirements have been met, and formal acceptance of the solution. In an ideal situation, contractual payments reflect that.

Complex solutions require complex testing, but the combinatorial explosion of all cases needed to exhaustively test a complex solution makes it prohibitively expensive and time-consuming.

This is best addressed by Quality Assurance – design for prevention of defects rather than their detection afterwards – coupled with carefully-designed testing to allow acceptance. **Systems thinking is vital from the start of the project to design a solution that can be tested and accepted in a realistic length of time without compromising safety and security.**

4.4 Risks in requirements management

SEPM helps:

- *Comprehensive understanding of risks affecting requirements*
  - *Scope*
  - *Interconnectedness*
  - *Emergence*
- *Selection of effective solutions*

Contracts must implement effective, fair and project-success-driven requirements risk management

There are several common risks in requirements management that have the potential to seriously disrupt or even cancel projects:

4.4.1 Requirements validation too late

Major flaws in the documented requirements are uncovered late in the project, requiring a major investment to correct or the abandonment of the project altogether e.g. NHS National Programme for IT [Ref 5.].

4.4.2 Point solution – impacts and opportunities outside not considered (or misunderstood)

The requirements are for the solution to a specific problem, which changes or even disappears during the life of the project. By not looking far enough forward, or outside the scope of the problem, the solution becomes obsolete e.g. Tornado ADV interceptor designed for Cold War defence of the UK only. [Ref 1.]

4.4.3 Reaction – locking in the past

The requirements are often intended to lock in current ways of doing things to avoid change, missing the opportunity to move forward, and the economies of off-the-shelf best-practice solutions (especially in IT).

4.4.4 Contractual terms creates conflict of interest over requirements management

Most fixed price contracts allow for changes to be priced separately, so there is a common practice of suppliers under-pricing to win the original contract then recovering profits through changes to requirements or specifications.

4.4.5 Picking the wrong horse – future proofing

It is possible to select a basis for the solution that doesn't have either the core capability or growth potential to meet the customer's needs e.g. the Nimrod AEW3 which was under development from 1977 until cancelled in 1986, replaced by the US Sentry aircraft of twice its weight, with much more powerful computers. [Ref 2.]

This is particularly a risk with Agile approaches.

4.4.6 Missing the too-obvious requirements

Some requirements are so obvious that they are overlooked in the documented requirements. Lockheed Martin were so focused on the functional requirements of the F22 stealth fighter that they neglected to address reliability and operating costs. The programme was terminated prematurely because both of these were unacceptable. [Ref 3.]

4.4.7 Customer "solutioneering" – the customer specifies the solution

In an ideal world, the designer is free to translate requirements into specifications, but sometimes the customer demands certain components/suppliers be used. This may still meet functional requirements, but may compromise the non-functional requirements e.g. Type 45 destroyer turbine engines, which are unreliable in hot climates and fail, tripping out all power for propulsion and weapons systems. [Ref 4.]

4.4.8 Under- and over-definition of requirements

The under-definition, and unfettered changes, of the requirements for the Scottish Parliament Building led to "boiling the ocean" resulting in massive delay and over-spend. In contrast, over-specification of brackets "for safety" in one factory resulted in brackets that were too heavy to be supported by the factory wall structure, much less carry the pipework intended.

5 How can a Fusion between SE & PM help

In summary, the fusion of these skills and disciplines can help in requirements management by ensuring that:

- The true requirements across the full scope (and only the full scope) of the project are revealed and documented
- Documented requirements are internally consistent and comprehensive
- The solution design optimises conformance to requirements with cost and schedule
- A sound contractual basis for requirements management is set up as part of the agreed and contracted scope of work for the project.
- Changes to requirements and design are thoroughly impact-assessed by having an integrated (Cost, Schedule, Scope of Work; Commercial and Configuration) project Change Management process.
- Non-functional requirements are not overlooked in the design of the solution
- Acceptance testing is designed and planned-for, early in the life of the project

Acceptance testing (Technical) milestones are synchronised with Project (Phase) milestones i.e. Project Phase (gates) milestones are synchronised with Configuration Baselines.

6 When can a Fusion between SE & PM help

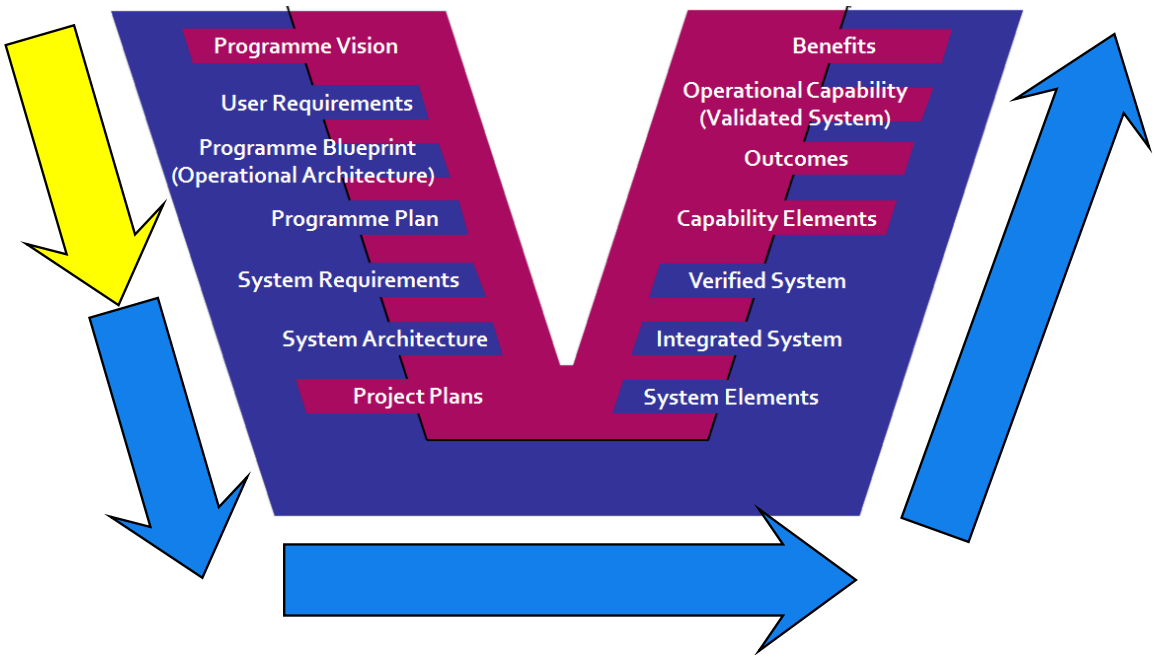
Requirements management is essential throughout the entire project lifecycle, so fusion between the disciplines is beneficial at all stages, but particularly:

At the start:

- In requirements definition/capture
- In confirming all (true) requirements are clarified/agreed with the User/Client and documented as part of the Contract (Baseline).
- In planning for acceptance

Throughout:

- In requirements change management
- In accepting synchronised Project and Technical milestones/baselines.



7 References

1. https://en.wikipedia.org/wiki/Panavia_Tornado_ADV
2. https://en.wikipedia.org/wiki/British_Aerospace_Nimrod_AEW3#Aircraft
3. https://en.wikipedia.org/wiki/Lockheed_Martin_F-22_Raptor
4. https://en.wikipedia.org/wiki/Type_45_destroyer
5. https://en.wikipedia.org/wiki/NHS_Connecting_for_Health