

Enterprise Learning:

How to gather requirements to deliver effective learning

by Paul McKey

Last month I explained the virtues of requirements gathering in the processes of designing, developing and delivering first class learning and development programs. This article will now endeavour to explain the art and science of how this is done.

I say endeavour since it is difficult to understand why requirements are so critical to the success of any learning and development program until you have lived through a couple of disasters. Hindsight, the great teacher, allows us to focus on the things that, given the opportunity, we would probably do differently next time.

The first thing hindsight teaches us is that we should have asked more questions at the start of the project. It also tells us we should have asked more poignant questions and we possibly should have asked more appropriate people. Vague beginnings lead to vague outcomes. While it is a wonderful goal, clarity is rarely found by one person telling and the others listening. Clarity of purpose for the project is only reached by all participants asking questions until everyone involved reaches a shared

understanding of what is needed. The gathering of requirements aids this process.

Yet this process shouldn't exclude imagination. In fact imagining exactly how you want the outcome will be a major first step in being able to describe and hence, achieve it.

The gathering of requirements goes through a number of phases from the abstract to the concrete. Begin with the questioning stage where people describe their wants and needs with diagrams and narrative. These are the requirements. Then comes the define and design phase where we shape and mould these requirements in the form of scenarios with prototypes. Finally we reach the fine tuning phase where the product, learning program or software application is readied for use.

Unfortunately at each step the twin evils of time and money constrain our plans. Therefore we need to understand as much about constraints (real or perceived) as we do about our needs. At first it would appear that budget and time constraints can

be left till after we have gathered other functional information, but this would be a mistake as they will have such a strong influence on the outcome. Other constraints such as environment, technology, skills and capabilities will also play an important role so these should be understood early in the program.

So requirements gathering, developing business cases and defining learning outcomes all go hand in hand for any learning and development program.

Where to start?

The art of gathering requirements starts with the abstract and moves toward the concrete. What is it we are building here? What purpose would/could it serve? For the sake of this article we will assume that the business case or the Why? questions have been answered and are clear to all. (Elsewhere I have described the role of Synergistic Design in ensuring that your business, purpose, technology and people are aligned behind your goals and strategies.)

A well written requirements document is an important first step in defining the way you do business.

Requirements then are simply a logical way in which to describe what your online learning project, new software application or existing process will deliver in an objective and, hopefully, unambiguous manner. They are the 'language' of any business case. Hence as I emphasised in part one of this article, your requirements should be gathered be-

fore committing to a technology solution to ensure you get what you need and not just what someone else sells.

Who should do the gathering?

The methods for gathering requirements differ depending on the system being developed. For an existing process or procedure the best method is still interview and review with key stakeholders. There are no shortcuts to working closely with people in this process. Yet interview and listening skills are not widely held among the development community. So for the success of your program seek out a generalist to lead it, someone who will take all perspectives into account and who can illicit quality information from respondents while balancing needs with constraints.

If good requirements need a good listener and questioner then the best place to start is with a facilitated workshop where all the key stakeholders are gathered to address the 'big picture' elements of the program. If it is a learning program that is being designed then the group should be able to discuss the business case, learning environment, the course or curriculum, the instructional design, teaching and learning methods, content delivery and so on.

If the program is new the group needs also to be capable of lateral thinking. Perhaps your company writes paper-based courseware and now wants to try online learning. Yet just publishing text on the web won't make you very competitive. A well written requirements document is an important first step in defining the

way you do business. The more detailed the better. So you may need to brainstorm new business and marketing models which will define the required learning programs. You could also utilise new techniques, such as interaction design using personas, to model an effective and

highly useable system. These will help you break out of existing habits and hopefully deliver engaging and invigorating learning programs.

For large projects it may be best to break up into groups to address certain requirement types. A full list of requirement types is listed below:

<p>PROJECT DRIVERS</p> <ol style="list-style-type: none">1. The Purpose of the Project2. Client, Customer and other Stakeholders3. Users of the Product <p>PROJECT CONSTRAINTS</p> <ol style="list-style-type: none">4. Mandated Constraints5. Naming Conventions and Definitions6. Relevant Facts and Assumptions <p>FUNCTIONAL REQUIREMENTS</p> <ol style="list-style-type: none">7. The Scope of the Work8. The Scope of the Product9. Functional and Data Requirements	<p>NON-FUNCTIONAL REQUIREMENTS</p> <ol style="list-style-type: none">10. Look and Feel Requirements11. Usability and Humanity Requirements12. Performance Requirements13. Operational Requirements14. Maintainability and Support Requirements15. Security Requirements16. Cultural and Political Requirements17. Legal Requirements <p>PROJECT ISSUES</p> <ol style="list-style-type: none">18. Open Issues19. Off-the-Shelf Solutions20. New Problems21. Tasks22. Cutover23. Risks24. Costs25. User Documentation and Training26. Waiting Room27. Ideas for Solutions <p>www.volere.co.uk</p>
---	---

A powerful tool used at this stage to give the client interviews some purpose is mind mapping¹. Starting with a central theme mind maps allow people to freely associate their needs and desires in a simple and visual manner. Further questioning allows connections and gaps to be identified and the maps can later be evaluated for overlap, redundancy and priority of ideas. Participants should not try

to provide solutions or the how-to at this stage.

The objective of all this brainstorming is twofold. For non-functional requirements it is designed to get clarity on the subjective and contentious areas such as *10. Look and Feel* or *11. Usability*. The other major purpose is to identify business events. Business events will guide the major

¹ <http://www.jcu.edu.au/studying/services/studyskills/mindmap/>

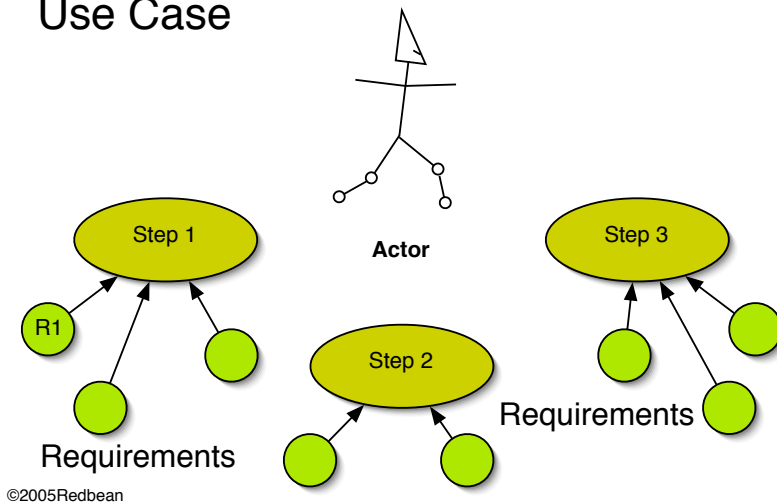
functional aspects of the new program.

Use Case Design

Business events may be simple tasks, such as register a student, or complex sequences, such as develop an end of

grade, no less. We may ask that the amount of tea in a pot is one tea-spoon for each person and one for the pot. We may suggest that the tea draws for three minutes and finally we could even describe the tea service process for optimum enjoyment. These are the functional requirements.

Use Case



year report, yet they all start with a description of a discrete piece of work to be done. This leads to the somewhat cryptic concept of a Use Case. Use Cases describe the pieces of work. That is if you were to describe someone *using* your system or process what are the *steps* they would take to get their *work* done (accurately, timely, cheaply and so on). For instance, in describing the making of a cup of tea (the work) we would follow steps such as boiling the water, drawing the tea and pouring the tea.

The requirements then go on to add increased detail to more accurately describe the criteria for success. For instance we may stipulate the tea water must reach 100 degrees centi-

Another important part of the Use Case is the actor doing the steps. While it will mostly be a person it could also be another process or software application since most Use Cases will be in a sequence of business events with inputs and outputs outside your program design.

Another way for educators to use requirements is to consider many instructional design elements as learning events. Hence the functional design of the teaching and learning model can be embedded in the system. This is particularly important in online learning where the pedagogical method can be easily derailed by poor or limiting application design. Good requirements can restore the balance so that the technology is supporting the education, not vice versa.

Build a Prototype

A prototype is something tangible everyone can sit around and discuss whether it is a new mousetrap design, a software application or just a process flowchart printed out so everyone can scribble on it. Have a chat about it, write the must-haves and must-not-haves as requirements and

then verify the design. You may need to do this several times. It is always better to improve the design in development than to alter the finished product. Alan Cooper (www.cooper.com) the interaction designer also suggests to always throw your prototype away after its job is done. This is mostly good advice since the small extra effort and expense will invariably result in a better outcome.

Prototypes form a critical communication tool for all parties to discuss. This is true for both functional and non-functional requirements.

Capturing Requirements

Finally you will come to the task of capturing your requirements into a

system where they are easily modified and maintained if need be. A number of commercial applications are available to help you do this. At workshops gather fast flowing ideas on flip charts. In interviews use mind maps. Capture Use Cases with flow charts then highlight each step and its requirements for success. Gather key stakeholders to thrash out the non-functional requirements. Use a facilitator if need be and a decision-making process such as multi-voting to prioritise. Your objective is a crisp, clear set of requirements you can hand to a developer or vendor and say "this is what we want , please cost it". Below is a simple table view of some major requirements for an online learning system.

Requirement Type	Description	Rationale
4. Mandated constraint	All systems and applications will aim to achieve a high level of future-proofing through the use of standards, maximum customisability and flexibility in design, and the minimisation of proprietary components.	The working life of the core systems should be within the five+ year range before a major overhaul is required.
8. Product Scope	The learning environment and systems should support the building of Communities of Practice within the system.	If not required immediately systems should at least have some capability for ad hoc collaboration in the future.
9. Functional	The required learning environment and systems shall provide the ability to match a competency profile to a role description.	Supports individual learning pathways.
9. Functional	The learning environment and systems should support persona profiles in each mode sharing a common record with customer, student and learner type data.	One person, one record is a goal of the system. This allows integration of an individual's data.

Requirement Type	Description	Rationale
13. Operations	All applications in the learning environment should be capable of being hosted by commercial hosting providers without loss of functionality	Hosting flexibility is important to ensure the best access for all users.

Some people prefer to use paper cards for gathering data, as illustrated below and described in the Volere system, yet eventually they should be digitised for easy communication. The level of detail for each requirement will depend upon the critical nature of the program or application and the relevant importance of the requirement being described.

Requirement #: **75** Requirement Type: **9** Event/use case #: **7, 9**

Description: **The product shall record all the roads that have been treated**

Rationale: **To be able to schedule untreated roads and highlight potential danger**

Source: **Arnold Snow - Chief Engineer**

Fit Criterion: **The recorded treated roads shall agree with the drivers' road treatment logs and shall be up to date within 30 minutes of the completion of the road's treatment**

Customer Satisfaction: **3** Customer Dissatisfaction: **5**

Dependencies: **All requirements using road and scheduling data** Conflicts: **105**

Supporting Materials: **Work context diagram, terms definitions in section 5**

History: **Created February 29, 2000**



Copyright © Atlantic Systems Guild

Summary

Learning and development program design is difficult at the best of times. Yet there is a well articulated method for improving the success rate of new implementations and reducing associated risk. We never build a house without a detailed plan so why do we regularly design learning programs with little more than a bunch of vague learning outcomes and a list of system functions? Particularly when enterprise learning projects can cost as much as a house! Requirements gathering is a simple method for developing a blueprint for any new project, not just software applications. Yet invariably all projects will also use technology in some form so providing clarity and an easily understandable blueprint about its use and scope is crucial. It is an easily learned and readily understood process which will help ensure the success of your programs and provide returns for your time and effort many times over.

Bibliography

1. De Bono, Edward, 1970, "Lateral Thinking", International Center for Creative Thinking, ISBN: 0140137793
2. Robertson, Suzanne and Robertson, James, 1999, "Mastering the Requirements Process", ACM Press, ISBN: 0201360462
3. Tiwana, Amrit, 2000, "The Knowledge Management Toolkit", Prentice Hall, ISBN: 0130128538

Paul McKey is managing director of Redbean Learning Solutions, and writes and speaks extensively on the subject of online learning. He incorporates a background in business, technology and education to bring a unique, creative, and commercial perspective to the implementation of learning in organisations.

Redbean Learning Solutions is an international learning and development organisation specialising in the design, development and implementation of online learning programs. Redbean can provide better business outcomes for your organisation through an independent and objective approach to helping improve your people, purpose and practices.