

## TL Note – Requirements

Elicitation and management of requirements is important to projects, regardless of what process (e.g., Waterfall, Agile, etc) is being followed. Following are some important starting points for understanding requirements and requirement related activities appropriate for TechLauncher.

### What is a Requirement

The first thing to note is that usually a client brief will include only some high-level needs or features. This can be a starting point for identifying the requirements to be satisfied, but they are not in themselves the requirements. A requirement is something that needs to be satisfied through the project. Implementation, testing and acceptance are carried out against requirements. Hence requirements must include sufficient information for this to occur. Often requirements for functionality (functional requirements) are easy to identify. But don't forget to consider non-functional requirements, especially when deciding on product/system architectures and technologies. This includes things like performance, usability, data security and privacy requirements.

### Requirements Activities

Requirements activities always form part of your overall development process, but there is variation on how the requirements related activities map into the chosen process. For example, for a Waterfall process requirements are all finalised prior to implementation. For agile, it is important to understand scope and features early, but requirements activities are done in a "just in time" iterative way. The activities include:

Elicitation	A common mistake is to focus only on the obvious requirements. Effort and activities involved in elicitation will be very project specific, but should be: <ul style="list-style-type: none"><li>- Known requirements – where there is a clear source of requirements</li><li>- Known, unknown requirements – know what you need to find out.</li><li>- Unknown, known - have information but not yet known why useful.</li><li>- Unknown, unknown – research, investigation and discovery may be needed to surface the requirement.</li></ul>
Analysis/ Evolution	It is unlikely that you will be able to immediately define an accurate requirement at the right level of granularity for implementation. Decomposition, grooming, and the use of various requirements analysis techniques will be needed to ensure accuracy, and cohesion.
Documenting	Requirements can be documented in the classic form of "The system shall...", through to user stories in a User Story Map. How you document them is not so important as ensuring that they can be well understood for verification, implementation, validation, and form a baseline for change.
Verification	Testing activities start with verification of requirements. If the requirement is not correctly defined and/or understood, then implementation will not be of value, no matter how well it has been done.
Managing Change	It is wrong to expect that requirements will be stable for the duration of the project, but ad hoc changes can quickly derail a project. Managing scope, scope creep, impacts of change, and client expectations are all important in a project context.
Implementation	All requirements should eventually trace to some sort of deliverable, and all deliverables should trace back to a verified requirement. This traceability should be constantly used to monitor progress, scope creep, and completion.
Testing	Tests should be defined against requirements. Failed tests can then be used to check for correct implementation, or in some cases identify requirements errors or misunderstandings.