TickIT*plus* Implementation Note

Title	Implementing the Lifecycle Model Management process ORG.10		
Date	February 2014	**Reference**	TIN011-1402
Originator	Dave Wynn	**Version**	v1r1
Key Terms	Implementation, Lifecycle Model Management, ORG.10		

The Base Process Library contains many processes associated with the development approach such as Requirements Analysis, Development Implementation, and Integration Management to name a few, but it also includes a process called Lifecycle Model Management. This process isn't actually part of the development approach, but has been included to manage the introduction, deployment and improvement of organisational development lifecycles.

At first glance it may look as if the Base Process Library contains processes that suggest the Waterfall model should be adopted, but this is not the case – the irregular order that they have been defined in the BPL helps support this, i.e. the first development process defined is actually Integration Management. The Base Process Library identifies processes that would typically be expected in all commonly defined models such as the V-model, the Spiral model, the RAD model and even Agile, but it doesn't infer or suggest any time sequence to these processes.

Development Approach Processes	Development Support Processes
Stakeholder Requirements Definition	Verification
Requirements Analysis	Validation
Architectural Design	Change and Configuration Management
Development Implementation	Problem and Incident Management
Integration Management	
Transition and Release Management	
Maintenance Management	

In simple terms, a formal development lifecycle pulls these processes together into a defined sequence and establishes the controls necessary to progress through the lifecycle in a managed approach. For example, in the classic Waterfall model, the sequence of these processes is quite straight forward and starts with establishing the stakeholder requirements, analysing those requirements, understanding the architecture, designing and developing the product, integrating and testing the product and then releasing it for use. The control points tend to be end of phase reviews such as the requirements review, design review, test readiness review, acceptance testing etc.

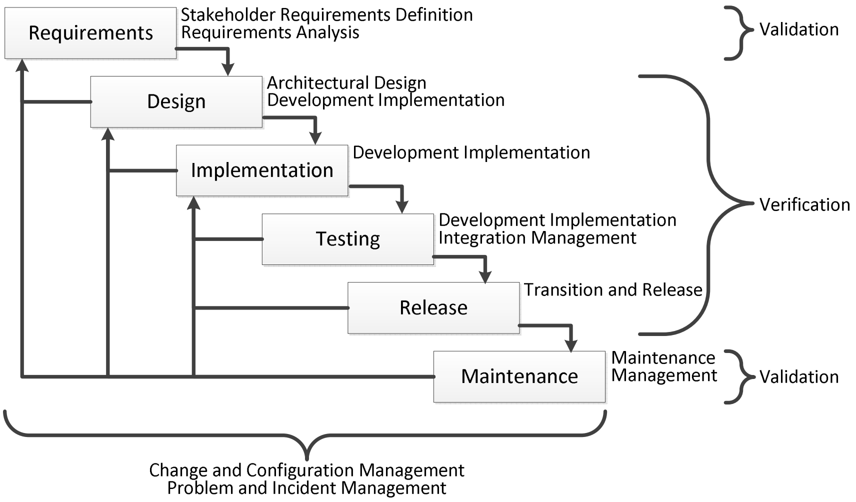

Fig.1 Simplified Waterfall lifecycle model

For Agile, where Sprints are used, these processes often occur in no particularly formal sequence within the Sprints although, clearly, there is some sequence inferred purely because it would not be possible to develop some code until at least some form of requirement has been identified. Often, when product releases are being developed, a series of Sprints are identified. It is typically the early Sprints that 'groom' the requirements, the middle Sprints that address the main design, development and unit testing, and the latter sprints that perform the integration, system testing and acceptance test. For Agile the main control mechanism tends to be the end of Sprint or Exit reviews.

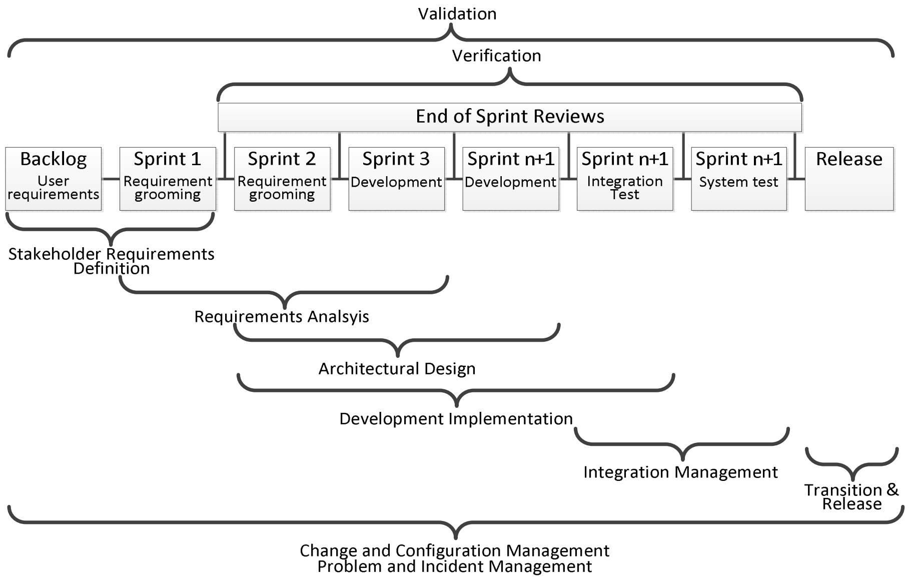

Fig.2 Simplified and generalised Agile lifecycle model

Even for a less common lifecycle such as the Spiral model, the same approach applies. The early phases tend to be based on prototyping ideas in order to gain a better understanding of the actual requirements and then the more traditional Waterfall model is adopted to take these requirements through into a full product. The lifecycle control would typically be through something similar to the Agile approach during the early phases, i.e. exit reviews, and then more like the Waterfall model for the latter phases, i.e. end of phase reviews.

There are other common lifecycle models and probably hundreds more variations on the classic models being used by organisations. This is where the Lifecycle Model Management process comes into play. It is designed to manage the creation, pilot, deployment and improvement of the lifecycles models needed by the organisation.

There are 5 Base Practices defined within the process and these are well described in the TickITplus Base Process Library Guidance document and so not repeated here. However, one question is repeatedly asked which is worthy of some discussion. Many organisations say that they have already got well established development lifecycles and therefore "how can the necessary evidence be provided to demonstrate the implementation of the Lifecycle Model Management process?" All in scope processes must be demonstrated under the TickITplus scheme rules (see rule CSR630 in the Core Scheme Requirements) and retrospectively putting in place the evidence needed from the previous implementation of a lifecycle model wouldn't be useful or beneficial. However, the solution is accommodated in the actual process, where both new lifecycles and changes to existing lifecycles are covered.

Given that TickITplus is aimed at supporting business improvements, it would be likely that some aspect of the existing development lifecycles would have been, or would need to be, improved at some point; handling these improvement changes would therefore be managed through the Lifecycle Model Management process. This would allow all appropriate evidence to be available to demonstrate that the process fundamentally works for the organisation.

Of the Base Practices, a number would be fairly easy to show existing evidence of implementation. There should be a policy and procedures covering BP.1 a need for the lifecycle that already exists, from the business plan if nothing else, to cover BP.2 and the actual defined lifecycle(s) for BP.3. This would leave BP.4 and BP.5. However, all processes should be reviewed on a period or event driven basis and this would include the processes that implement the lifecycle, for example through an end of project lesson learnt review, retrospectives or even through internal audits. Assuming that something is done with the results of the review, this would provide evidence to support BP.5. If this is not being done then there would be a good case for an assessor to raise a finding.

This then leaves BP.4, pilot the lifecycle model. This might be difficult to show for the entire lifecycle after many years of using an existing one, although arguably there would be plenty of evidence to show that the lifecycle has been piloted simple through the years of use. However, the process does talk about "new" or "revised" lifecycles, so this would suggest that changes or improvements to an existing lifecycle should also be piloted following feedback from BP.5. Remember, the BPL processes do not say how the requirements should be implemented, but just what should be done. Therefore piloting could range from a formal pilot of the change on a new 'risk-free' project to the changes being monitored on existing projects, maybe through audits, or again at the end of the first project to use the updated lifecycle, i.e. through BP.5 again.

It could still be argued that there may well be a situation where no lifecycle change is necessary within a given implementation timeframe, but this situation is no different from other standards and schemes which require tangible evidence of implementation. For example, in implementing ISO 9001, the organisation cannot simply say that they haven't had any customer complaints, haven't got around to doing internal audits, haven't seen the need to do preventative action, or, indeed, not yet held a management review because the schedule hasn't required one.