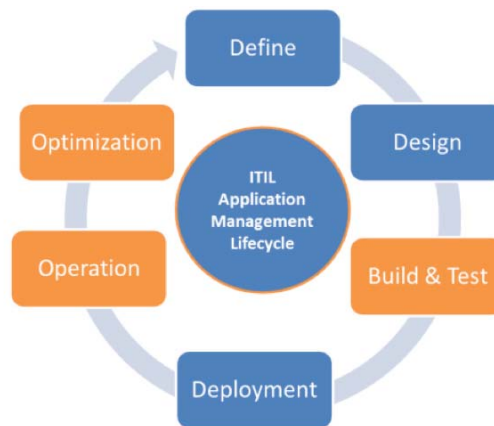


ITIL Application Management Lifecycle: A SolarWinds Sysadmin Guide

ITIL Application Management Lifecycle: A SolarWinds Sysadmin Guide

ITIL Application Management Lifecycle is comprised of five main functions – from defining application requirements to design of the application, build, test and deployment to production, ensuring effective production operation and lastly, optimization of the application. This guide will help you understand the capabilities SolarWinds provides to help sysadmins and application support personnel in implementation of key ITIL processes. The figure below highlights in SolarWinds orange the primary phases of the lifecycle where SolarWinds has strong tooling.



Requirements:

This phase is where application requirements are defined by the line of business or application owner – whether you are building internally, outsourcing development or buying an app off the shelf. The sysadmin can provide guidance into this process for performance requirements. If the application is new, performance requirements can be taken from existing in house similar applications or competitive applications. For example, if you are developing a new e-commerce application, you may want to test the responsiveness of competitive e-commerce applications to establish a baseline.

Design:

This phase takes the requirements and translates them to development stories and tasks to include defining the software architecture and operational (software and hardware) requirements for running the software. This phase is done by the product management & development team or by the team/person that selects, evaluates and procures software off the shelf.

Build & Test:

Build and test is an iterative phase. If software is developed, this phase will occur in release one, and again as the application is updated and fixes are implemented. If software is procured off the shelf, patches need to be researched, tested and applied. For this phase, SolarWinds provides the following capabilities:

- Web application & website testing: Test website availability and web application responsiveness from multiple geographic locations to include monitoring from the Amazon EC2 cloud for businesses that do not have physical geographic presence.
- Composite application resource monitoring: Monitor the performance of applications and dependent infrastructure such as database SQL query responsiveness.
- Patching applications: Deploy, manage and report on patches for Microsoft & 3rd party applications (Oracle, Mozilla, Adobe, etc.) and utilize before and after actions to deploy complicated patches like Java.

Deployment:

This includes provisioning the application to include supporting components such as the OS, the hardware, database and so forth. Testing is important in this phase as well, to ensure the application works as expected in the production environment. It is important to test all aspects of the composite application (i.e. the application and supporting infrastructure elements) to ensure the application works as planned in the production environment.

Operation:

This phase is the primary job of the sysadmin, IT Operations Manager or the application owner, depending on the organization. This activity requires understanding the operational model of the application – what are all the components/configuration items of the application where something can go wrong. Composite applications include such configuration items as: hardware, operating systems, virtual infrastructures, databases, web servers, application servers, authentication services, and the application code itself – whether it's Java, .NET or an off the shelf application like Microsoft Outlook. To effectively monitor this application, you need to consider several aspects:

- Are the resources available or unavailable? This is the most basic type of monitoring and helps you identify if a problem is related to a failure of some sort – did the server or service stop working completely?
- How is the performance trending? Performance monitoring provides useful statistics such as SQL query responsiveness, utilization statistics, heap usage, garbage collection statistics to help sysadmins proactively identify performance problems with the application BEFORE they turn into application outages. It is recommended to collect and warehouse performance statistics to help establish a baseline for normal behavior to set thresholds and to also have data for performance analysis. Performance data is also used to understand capacity limitations at both the resource and application layer – is overall server cpu capacity reaching its limit or can heap size be adjusted downward and still retain adequate application responsiveness?
- How does the end user perceive performance? For web applications, this is often your first line of defense to understanding performance degradations with your application. SolarWinds provides the ability to test web application transactions from different geographic locations to provide advanced warning to performance issues. Since this capability breaks down response times for each step of the transaction, sysadmins can better troubleshoot where the performance problem is coming from - is it in the authentication step, or the checkout step, etc.

Optimization:

Are service levels performing as expected? What needs to be done to improve service levels? Can we use less capacity and keep service levels in tact? Do we need to retire this CI because it is causing more pain than benefit? These are the questions that need to be asked, and answered in the optimization phase. This activity is accomplished by analyzing help desk and problem records as well as analyzing performance data. SolarWinds provides performance monitoring as well as out of the box and customizable reports to show application availability and performance history.

