**Term list: Informatics Session 9, System implementation and management, v. 1.0.1**

Acceptance testing

Testing of a software system as a whole with a focus on meeting business and regulatory requirements. When acceptance testing is successfully completed the leadership of the software users will sign off on successful installation and fitness for operation.

Agile development style

An IT management style in which a project’s components are divided into relatively independent units and completed in sequential “sprints” that involve technical staff and users working together on teams. Ideally, each sprint yields a functional component that can be used with feedback available by the next sprint. The scheduling and goals of the sprints can evolve based on immediate needs as the project develops.

Application management

Operation and maintenance of specific software applications. Requires expertise in the application and is distinct from managing the computers that run the application (system administration) or the local computer network (network administration). LIS and related application management is typically carried out by a dedicated group either in the lab or in central IT but dedicated primarily to lab support.

Change management (also called “change control”)

The use of best practices in the modification of software, data, and computing resources. Changes may include updating configuration data, applying software fixes or enhancements, or adding new software or hardware components. The important elements of change management are authorization/approval, implementation, testing, documentation, acceptance/approval, and evaluation/monitoring.

Collision rate

In network administration and performance monitoring, collision rate is the frequency at which computers try to communicate simultaneously. A high collision rate indicates that a network is over capacity and needs to be reconfigured or upgraded.

Failover

The switch to a hot backup when a primary system fails. Failovers may be automatic or manual. Failovers for system components such as disk arrays can be largely automatic. In healthcare, automatic failovers of complete systems usually have some manual components such as re-configuration of interfaces to communicate with the backup system.

Functional requirements

The necessary features or capabilities of a potential new system (what the system should be able to do). See non-functional requirements.

Hot backup

A replicate system that can be brought into operation quickly if a primary system fails. The backup may be a storage system (ie, a backup disk array) or a complete replacement system including the application software. The switch to the backup system is called a “failover.” Hot backups differ from data backups in that they are replicates of parts or all of an operational system and therefore they back up the hardware and application software in addition to the data. Hot backups do not require that the system be rebuilt from the stored data as with typical data backups.

Integration testing

Integration testing refers to testing of a software module in the setting of additional modules with which it interacts, and it evaluates the ability of the new software to operate correctly with other related software modules (eg, correct connections and communicated data types). Integration testing is part of the testing that occurs in new software installation or replacement of a software module with a significant upgrade.

Maintenance tables and dictionaries

LIS database entries that support the operation of the system such as test and order definitions, personnel information, laboratory locations, instrument and printer information, interface information, and operational rules. These entries must be kept up to date as part of the routine management of an LIS.

Network administration

Configuration and monitoring of network devices and network communications, distinct from system administration or application management. Typically carried out by a group within the central IT organization.

Non-functional requirements

The way a potential new system should work to be successful in a local environment. This may involve meeting particular constraints of a local environment or supporting particular kinds of workflows or user needs (how a system operates as opposed to what it does). See functional requirements.

Open source licenses

A software licensing strategy that provides the user with both operational software and the source code used to build the software, and it may or may not be free (free open source software is often called FOSS). Open source software typically allows users to edit the source code (ie, change the software) for their own use. There are several styles of open source licenses with different restrictions on redistribution of the software and user modifications. Permissive licenses (BSD, MIT) place few restrictions on redistribution and even allow incorporation of software into larger commercial software packages. More restrictive open source licenses (copyleft, GPL) may prohibit redistribution or require that any derived software be redistributed under the same open source license.

Performance monitoring

Ongoing evaluation of applications, systems, networks, and interfaces, with review and signoff. Designed to maintain high performance and catch developing problems early. The CAP checklist requires that the performance and acccuracy of LIS and related systems be evaluated every two years.

Proprietary licenses

Proprietary licenses give permission for installation and use of software without providing the source code and without allowing user modification. These licenses are most often sold with commerical software but may also be used with free software. License pricing is often based on the number of user accounts at a site (“seat” pricing), but is sometimes based on the number of simultaneous connections allowed (active users), the amount of data transferred within a time interval, or the total volume of data contained. Previously users paid a fee for installation and then paid separate fees for optional upgrades, but the recognition of the universal necessity of upgrades (for security, regulatory compliance, and interoperability) and ongoing support has moved most vendors to an annual subscription model. This approach also has the advantage for customers of lower initial cost and consistency of cost prediction.

Regression testing

Repeated software testing using established data and benchmarks, to verify that software performance remains acceptable. Regression testing is an integral component of a performance monitoring strategy. It differs from unit testing and integration testing which are designed to verify the performance of new or substantially upgraded software

Request for information (RFI)

A document sent from a potential customer to a list of vendors that defines the customer’s environment and goals, and asks for a general description of the vendors’ organizational characteristics and how their software would fit into the environment and meet the goals.

Request for proposal (RFP)

A more detailed document than an RFI that requests additional information from a few top-ranking vendors. It extends the RFI by asking for proposed licensing terms and cost, an overview of the installation process and requirements, and a proposed strategy for support and cost.

Rotating backups

A best practice for data backup that includes multiple backups, often on separate tapes, that are rotated by date and stored in different locations. A common rotation pattern is “grandfather-father-son” where the “son” backups are daily and 4-5 are rotated, the “father” backups are weekly, and 4-5 are rotated, and the grandfather backups are monthly and 12 are rotated.

SaaS

Software as a service. Refers to systems that are managed by the vendor, usually using a “cloud” design. LIS may be implemented as SaaS, but this does not change requirements for pathologist review and signoff of system testing and performance data, and for documentation and signoff of LIS procedures. The approved procedures should be accurate and easily available to the lab. The SaaS vendor should work closely with the laboratory to meet these requirements. Those considering SaaS systems should also understand whether the vendor will use the data for additional purposes, the ease/cost of access to data for reporting and analytics, and how the data would be recovered and any associated costs if the SaaS agreement were terminated.

Software lifecycle stages

1) Requirements specification, 2) system selection, 3) system installation and testing, 4) system operation and maintenance, and 5) system termination and decommissioning.

Spiral development style (also called “iterative” style)

An IT project management style in which a project is carried out in repeated plan-design-develop-test stages. This style allows adjustment of later stages based on evolving needs and the results of earlier stage testing.

Stakeholder groups

Groups with different organizational perspectives (eg, clinical services, administrative, financial, regulatory) who will be users or otherwise affected by a system and should have a say in how it works.

System administration (Sysadmin)

Management of computers and operating systems. This function is generally managed by central IT and is distinct from application management, provided by application specialist teams that may be managed by stakeholder groups or central IT, and network administration, also usually managed by central IT.

System testing

Testing a software system as a whole in a way that touches many modules and confirms their correct individual and collective operation.

Unit testing

Testing a software module in isolation to make sure that internal functions and calculations are performing correctly.

User training

In system installation, teaching users the operation of a new system to perform the tasks of their roles. This should be carried out as late in the installation process as is practical, and no more than a few weeks before go-live. If user training is carried out too early the knowledge will fade. In contrast, application management training is generally carried out early in installation so that the managers can particpate in the installation process and later user training. User training persists after go-live to support new users, changing user roles, and refresher training.

Waterfall development style

In software development, a project style in which requirements, design, development, testing, deployment, and maintenance are carried out as separate sequential steps. This style has been problematic, especially in large projects, because it can be difficult to specify all requirements up front and because the time required for design and development can cause the requirements to go out of date. Development styles intended to address these problems include iterative, which builds software progressively in cycles with user review and design corrections embedded in each cycle, and agile development in which users are included as part of development teams and software is built in short, intense “sprints” with user feedback throughout.



© 2024 James H. Harrison, Jr. This glossary is made available under a [Creative Commons Attribution 4.0 International License](http://creativecommons.org/licenses/by/4.0/). The contents may be used without restriction, modified, and redistributed if they are attributed and any changes are indicated.