

IT Applications for Healthcare: Leverage Processes for High Quality

By Ravishankar N

An integrated process framework derived from industry models can help address compliance, cost effectiveness and quality challenges

Healthcare systems make effective use of IT applications to deliver affordable, quality healthcare to the human community. Worldwide, the total IT spend of healthcare organizations is predicted to be around US\$ 94 billion by 2011 [1]. IT applications significantly impact the outcome of a variety of healthcare systems, ranging from a hospital management system to core pharmaceutical research and drug development systems. IT divisions of healthcare organizations face challenges related to compliance, cost effectiveness and quality. Thus, besides delivering the required functionality at affordable cost, the IT applications have to deliver on three key parameters:

- **Performance:** Transaction volume handling ability, response time, etc.
- **Security and Privacy:** Prevention of loss and unauthorized access to critical data.
- **Compliance:** Ability to enforce the controls required by statutory regulations.

Though these parameters could be considered within the broader context of *quality*, each parameter represents a key dimension in the success of a healthcare IT application that satisfies the core functional requirements. Increasingly, with stringent regulations and compliance expectations, the requirements related to the above key parameters are being considered as seriously as the basic functional requirements of a given healthcare system.

CURRENT CHALLENGES

Today IT divisions of healthcare organizations face challenges in delivering cost effective IT solutions doing full justice to the three key parameters listed above. Many statutory regulations impact the IT solutions. While some of these regulations indirectly relate to IT systems, some directly ask for certain elements to be part of IT systems. Some examples are:

- SOX 404 that specifies IT controls to

- minimize financial risks
- FDA 21 CFR Part 11 – electronic records, electronic signatures
- FDA 21 CFR Part 820 – Quality System Regulation

Healthcare IT organizations' applications have to build the necessary controls in their organization and the necessary checks and control features in their products, to satisfy the applicable sections of each of these stipulations.

Security vulnerabilities such as hacking and theft of private data pose a threat to the success of the IT applications and diminish confidence levels of users as well as service providers. These vulnerabilities exist in both paper and electronic records. According to the Los Angeles Times, roughly 150 people for e.g., doctors, nurses, technicians, billing clerks, etc., have access to parts of a patient's records during a hospitalization and about 600,000 payers, providers and other entities that handle providers' billing data have some access to the records too [2]. To thrive in such a scenario, IT organizations have to constantly attempt to build more and more checks and controls to ensure that their products are robust enough to prevent loss, as well as spot and report unauthorized access to private data.

Large integrated databases, data mining, extraction, analysis and accurate reporting of clinical data are some examples of today's requirement in the healthcare field. In 2005, the National Health Service (NHS) in the United Kingdom began an Electronic Health Records system. The goal of the NHS is to have 60,000,000 patients with a centralized electronic health record by 2010. The plan involves a gradual roll-out, providing general practitioners in England, access to the National Program for IT (NPfIT) [3]. That is just an example of the size and processing speed

requirements of a Healthcare IT system. To cater to such a large volume of patient data and to retrieve, process and report data at acceptable speeds to doctors and researchers, the IT applications must inherently provide built-in design elements and software features supported by fast and reliable hardware components.

PROCESSES AND MODELS THAT CAN HELP

The success of a Healthcare IT application is largely influenced by the processes used to conceive, design and develop it. Given the 'mission-critical' nature of majority of healthcare systems it is of utmost importance that the processes used to create them are geared towards assuring near zero-defect quality. A set of structured processes leveraging industry best practices and process models can help an IT organization build the IT applications, to cater to the demanding requirements for success. Healthcare IT organizations have acknowledged the need for standards and processes. According to a Gartner survey, 'Implementing Quality Standards' is among the top 5 IT priorities for healthcare organizations [4].

The processes for engineering the software for healthcare applications could be structured in different stages:

- Requirements Elicitation
- Design
- Coding (Development)
- Validation
- Common to these core software engineering processes will be the steps for managing changes to requirements and the associated configuration/version management procedures
- Along with the overarching project management procedures, to support and verify the above activities, two other

Stage	Aspects to be Considered
Requirements Elicitation	<ul style="list-style-type: none"> • Unambiguous, implementable requirements • Documentation and traceability of requirements • Review and sign-off of requirements from the healthcare service providers • Performance, data retrieval, processing and display/reporting
Design	<ul style="list-style-type: none"> • Information availability and confidentiality • Interoperability with multiple external hardware/systems • Interfaces to legacy systems • XML/EDI data exchangeability
Development	<ul style="list-style-type: none"> • Customizable rule engine to deal with geo-specific stipulations • Code modularization and flexibility to take care of frequently changing healthcare regulations • Data encryption/decryption • Security features
Validation	<ul style="list-style-type: none"> • Review of code and design components • Manual and automated testing • Regression testing to unearth unintended changes to code
Configuration Management	<ul style="list-style-type: none"> • Version control • Naming convention • Traceability and change management • Release procedures, documentation • Integrated patient record and case history database to feed government and other research organizations on birth, death and statistics of diseases prevalent in different geo locations
Audits	<ul style="list-style-type: none"> • Security audits • Quality audits
Knowledge Management	<ul style="list-style-type: none"> • Taxonomy and classification of case references

Table 1: Stages to Address Specific Needs of a Healthcare IT system **Source:** Infosys Research

process sets would also be required:

1. Audit/ Assessment Processes
2. Knowledge Management Processes

The above stages could be used to address specific needs of a healthcare IT system. Some examples are given in Table 1.

LIFECYCLE STAGE OF A TYPICAL SOFTWARE PROJECT

A typical software application development project goes through the following software engineering lifecycle stages:

1. Requirements elicitation and analysis
2. Design
3. Development (Coding)
4. Validation (Testing)

To plan and execute these stages the project needs an overarching project management focus. Along with the activities specific to each of the software engineering stages, the success of the project is strongly influenced by support activities such as configuration management, quality assurance, reviews and audits and knowledge management.

Figure 1 overleaf shows how the compliance stipulations, lifecycle stages and the process models are inter-related.

With the compliance stipulations such as HIPAA, FDA 21 CFR and SOX on one side and the key dimensions of success at the core, the execution of the activities in the above lifecycle activities and the quality of the corresponding deliverables will determine the success of the project and the overall quality of the final product.

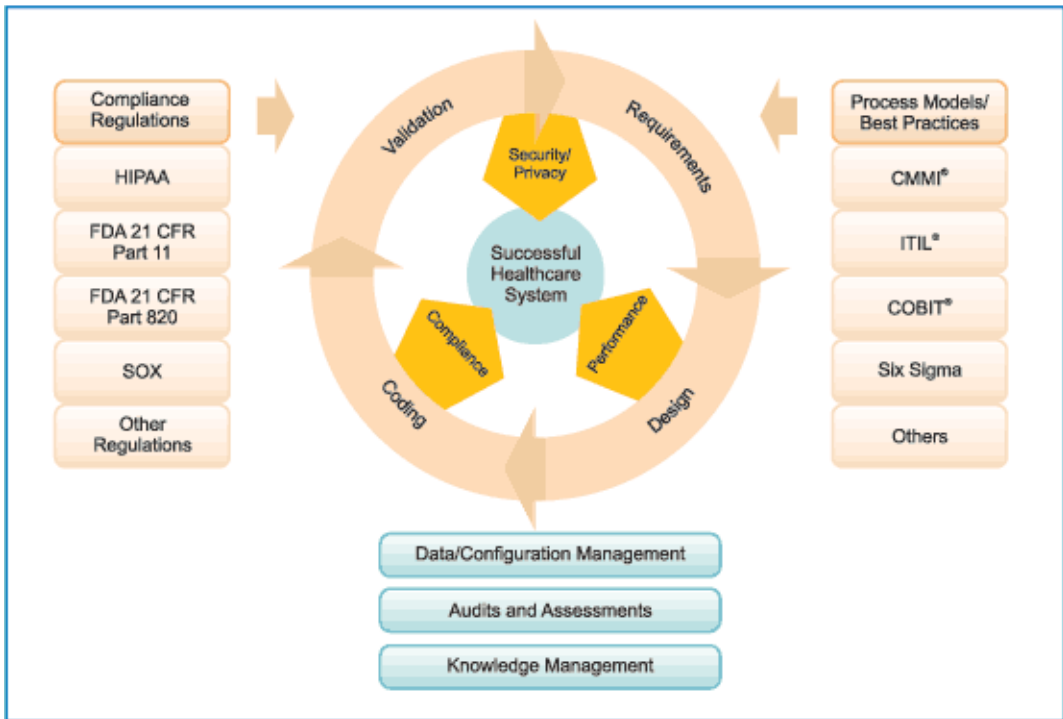


Figure 1: Compliance Regulations, Software Engineering Stages and Supporting Industry Models *Source: Infosys Research*

This is where an IT organization can leverage process models effectively in order to deliver quality healthcare IT solutions. As can be seen in the above diagram, the models provide

the necessary support by offering best practices, methodologies, controls and audit guidelines, thereby positively influencing the software engineering lifecycle stages.

SW Engineering Areas	Degree of relationship			
	CMMI®	COBIT®	ITIL®	Six Sigma
Requirements elicitation	**	*	*	*
Design	**	**	**	**
Coding	**	**		*
Validation	**	**	*	*
Configuration management	**	*	**	*
Audits	**	**	*	*
Knowledge Management	*	*		*
** Strong Relationship * Moderate Relationship Blank-Weak/no relationship				

Table 2: Industry Models Relate to Specific Areas of Software Engineering Lifecycle

Source: Infosys Research

	Performance	Compliance	Security
CMMI®	<ul style="list-style-type: none"> Requirements Definition Technical Solution Product Integration Verification, Validation Decision Analysis & Resolution 	<ul style="list-style-type: none"> Configuration Management Process and Product QA Requirements Definition Technical Solution Product Integration Verification, Validation 	<ul style="list-style-type: none"> Configuration Management Requirements Definition Technical Solution Product Integration Verification, Validation
COBIT®	<ul style="list-style-type: none"> Performance and Capacity Plan Requirements Define & Manage Service Levels Manage Operations Acquire and implement technology infrastructure Install and accredit solutions and changes 	<ul style="list-style-type: none"> One of the 7 core information criteria for COBIT More specifically, Monitor and Evaluate Domain - ME3 Ensure Compliance with external requirements – laws, regulations and contractual requirements 	<ul style="list-style-type: none"> 3 of the 7 core information criteria Information Security Confidentiality, Integrity and Availability DS5 – Ensure Systems Security AI 2&3 – Acquire and Maintain Technology and Application Infrastructure
ITIL®	<ul style="list-style-type: none"> Capacity Management Availability Management Service Level Management Incident Management Problem Management Change Management 	<ul style="list-style-type: none"> Change Management Configuration Management Service Level Management Release Management 	<ul style="list-style-type: none"> Configuration Management Change Management Service Level Agreements Availability Management IT Service Continuity Management
Six Sigma	<ul style="list-style-type: none"> House Of Quality Process Capability Analysis Benchmarking Hypothesis Test Design of Experiments Failure Mode & Effects Analysis 	<ul style="list-style-type: none"> Gage Repeatability and Reproducibility SPC/Control Plan 	<ul style="list-style-type: none"> Failure Mode and Effects Analysis

Table 3: Success Dimensions addressed by different Models *Source: Infosys Research*

Though the industry models have their own core objectives, they relate to specific areas of the software engineering lifecycle as shown in Table 2.

MODEL COMPONENTS AND SUCCESS DIMENSIONS

With a plethora of models, controls and best practices available, it is challenging for an IT organization to figure out which ones would best suit them. Having identified the models, it

is important to map the elements of these models to the success dimensions of performance, compliance and security. Such a mapping leads to a superset of elements from each model addressing success criteria in one way or the other. Table 3 depicts this superset of model elements.

As can be seen in Table 3, there are overlaps in the way the models correspond to these dimensions. For example, the success dimension of Security is addressed by Configuration

Management element of CMMI® as well as ITIL®, though with differences in coverage and implementation focus. Similarly, managing service level agreements is addressed by Define & Manage service levels and service level management components of COBIT® and ITIL® respectively. Overlaps such as these pose challenges to a healthcare IT organization in deciding on the right components from each model, to address the generic and specific requirements of IT applications. Thus, to avoid duplication of effort in implementing the overlapping practices and to keep off from the trap of process proliferation, it is imperative that the IT organization identifies the most relevant ones from the above set of process elements.

The following section proposes a framework that would help in making the right choices and integrating the applicable elements in alignment with the objectives of the IT organization.

THE PROCESS FRAMEWORK

A framework, enabling achievement of quality and the success dimensions, should provide a structure and discipline for software engineering using the industry best practices. At the same time it should provide the necessary flexibility to include new practices and tailor the existing processes and procedures to cater to the updates in compliance regulations from time to time. Further, with rapid technological advances, the framework should be scalable to exploit these advancements and cater to the changing needs of the user community in terms of performance and data security.

The integrated framework depicts a collection of specific components chosen from the superset of applicable elements from the industry models [Fig. 2]. It accelerates the 'Discover' and 'Define' stages in determining the best-fit practices leveraging on these models.

As seen in the integrated framework (Fig.2), the chosen elements from the models revolve around the software engineering lifecycle activities. Let us consider each lifecycle stage and discuss how the model components address the activities of that stage.

Requirements Elicitation and Analysis: This is an upstream stage in the software engineering lifecycle and entails interaction with the appropriate stakeholders to gather their business requirements, analysis and prioritization of the requirements and creation of specifications that would be fed to the next stage of the lifecycle. This is a critical stage of the lifecycle and the cost of fixing a requirements related defect in production could be 110 times more than the cost of correcting it during requirements definition [5]. In the framework shown (Fig. 2) the specific components from the different models are proposed to capture functional and non-functional (performance, security, compliance, etc.) requirements in a structured manner and to trace the requirements and changes to the requirements, to the subsequent stages of the development lifecycle.

Design: With requirements gathered and documented, the next stage in the lifecycle is Design. In this stage, foundation is laid to give shape to the requirements. This foundation could include a high level architecture, functional design, data models, technical design and detailed program specifications. A weak design could produce error-prone modules that could result in up to 50% of the defects [6]. Thus to make this foundation strong, flexible, scalable and unambiguous, the proposed framework recommends components from the industry models. These components would enable the development team in the following:

Model	Requirements	Design	Development	Validation
CMMI®	Level – 2 Requirements Management	Level -2 Configuration Management	Level - 2 Configuration Management	Level - 2 Configuration Management
	Level – 3 Requirements Definition	Process and Product QA Level – 3 Technical Solution Product Integration	Process and Product QA Level – 3 Technical Solution Product Integration Verification Validation	Process & Product QA Level – 3 Product Integration Verification Validation
COBIT®	AI3 – Acquire and implement technology infrastructure AI7 – Install and accredit solutions and changes DS5 – Ensure Systems Security AI 2&3 – Acquire and Maintain Technology and Application Infrastructure ME3 – Ensure Compliance with external requirements – laws, regulations and contractual requirements			
ITIL®	Change Management Service Level Agreements	Performance Management	Change Management Release Management	
Six Sigma	House Of Quality	Process Capability Analysis Benchmarking Hypothesis Test Design of Experiments Failure Mode and Effects Analysis	Process Capability Analysis Failure Mode and Effects Analysis Hypothesis Test	Failure Mode & Effects Analysis Gage Repeatability & Reproducibility SPC/Control Plan
	Management Commitment	Project Management	IT Org Level Audits/Assessments	People Enablement

Figure 2: Integrated Framework

Source: Infosys Research

- Define an enterprise architecture and lay down the design principles at the IT organization level
- Consider alternative architectures / designs and evaluate them against defined hypotheses and select one that would best suit the requirements
- Consider risk factors and possibilities of failure and provide for building alternatives, self diagnostics and failure handling features
- Provide for high performance, security and flexibility in the final product.

Development (Coding): This is the stage that actually gives shape to the requirements and builds the product. Adherence to the development standards, constantly tracing the software components to the design elements and the requirements and unit testing are key to the success of this stage. In the proposed framework, elements from industry models are chosen in order to ensure that the built software components meet the requirements, adhere to the quality standards, are traceable to the design components and test cases, and are managed

through sound configuration management procedures.

The elements proposed in the framework enable the IT organization to measure productivity of the development team, using size measures such as *lines of code* or *function points* and continuously improve team capability and hence the final quality of their deliverables.

Validation (Testing): Placed towards the end of the lifecycle before moving the application to production, this stage owns the responsibility to ensure that nothing defective escapes into a production scenario. The components suggested in the framework address different types of testing, including Integration Testing, System Testing and Regression Testing when the software undergoes changes. Before releasing the software to the field, it has to go through Acceptance Testing to give confidence to the user that the delivered software indeed performs what is expected of it. This type of testing is addressed by the validation process area of CMMI®.

As can be inferred from the above sequence of events, the software passes through different stages – from the developers desk through different testing areas to the user acceptance stage and finally to the field. To ensure the integrity of the software, versions and security, a robust configuration management process and a well-structured release process should be in place. These two aspects are addressed by configuration management process area of CMMI® and release management process of ITIL®.

With the overarching project management process, the software engineering activities are surrounded by knowledge management and audit/assessment activities that the IT organization has to identify and define at the

organization level. All the models mentioned in the framework provide recommendations for these support activities as well. For example, the project planning and project monitoring and control process areas of CMMI® Level 2 provide inputs on the best practices to plan and manage a project. Similarly the ‘Ensure Compliance with External Requirements’ (ME3) process of Monitor and Evaluate domain of COBIT® provides the necessary guidelines to ensure compliance with statutory and regulatory requirements.

IMPLEMENTATION APPROACH

The framework discussed in the previous section indicates the applicable elements of the industry models, to efficiently execute the different software development lifecycle stages in order to achieve quality, performance, security and compliance requirements. A healthcare IT organization has to carefully tailor the above framework in its specific business context. Choosing and implementing controls and best practices from the above framework requires a structured approach, taking into consideration the organization’s business objectives, expectation of the leadership and stakeholder needs. The following four-stage approach provides an implementation methodology [Fig. 3].

Discover: During the first stage in this methodology, an as-is study of the healthcare IT organization is conducted to understand the business context, challenges, future vision, objectives and targets. Based on the business an objective, management vision, applicable regulatory requirements and a to-be state is defined in consultation with the key stakeholders. Then the existing software development processes and practices are also

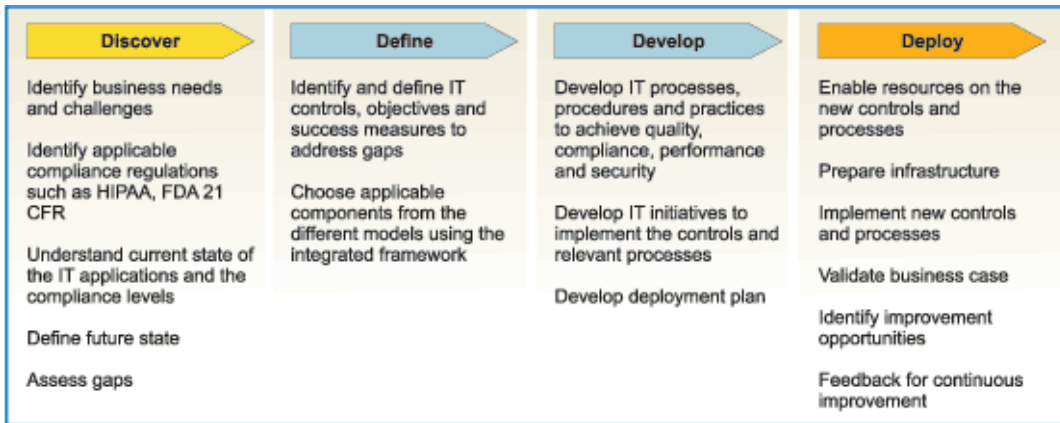


Figure 3: Implementation Methodology

Source: Infosys Research

studied in detail to identify gaps against the to-be state.

Define: In this stage, using the gaps identified in the previous stage, the IT objectives and targets are refined. The applicable IT controls are identified and defined in alignment with the objectives. Parameters for measuring success are also determined at this stage. Based on these, the appropriate components of the framework are selected to meet success criteria and achieve business objectives.

Develop: Detailed processes for the IT organizations are defined using the selected components of the framework. Along with processes the corresponding implementation aids such as templates, checklists and guidelines are also developed at this stage. Involvement of practitioners such as project managers, architects, designers and developers is key during the development of the processes and aids. Towards the end of this stage a plan is created to train and deploy the processes in the IT organization. If a need for piloting the processes in a small group is identified, suitable pilot plans are also created at this stage.

Deploy: This is the actual process implementation stage where the defined processes are implemented. As a precursor to implementation, the management’s mandate and support are communicated to the project teams, the necessary infrastructure is set up and people are empowered to implement the processes. After implementation at the organization level the benefits are validated against the success measures defined at the Define stage.

As with any change initiative, feedback is actively sought and continuous improvements are carried out recursively using the above stages.

APPLYING THE FRAMEWORK - AN ILLUSTRATION

The following scenarios illustrate how the framework can be applied to typical scenarios in the healthcare software area. These scenarios focus on one of the success parameters - compliance - and consider the compliance requirements of SOX, FDA 21 CFR Part 11 and FDA 21 CFR Part 820.

Tables 4, 5 and 6 illustrate how components of the framework can be used to address each of these compliance requirements.

Scenario 1: In this scenario a set of controls from SOX are considered, though not in any specific sequence.

Requirement	Model Used	Implementation Guidance
A control should exist to ensure that problems and incidents are appropriately resolved.	ITIL [®] : Service Support.	Incident Management: Define and deploy the processes to resolve an incident and reduce its impact. Problem Management: Define and deploy a problem management process to reduce the number and severity of incidents and problems on the business, and report it in documentation to be available for the first-line and second line of the help desk. The proactive process identifies and resolves problems before incidents occur.
A control should exist to ensure that relevant technical and end-user system documentation is updated after significant modifications/ upgrades are made to systems.	CMMI [®] Maturity Level 2 Process Area Requirements Management. CMMI [®] Maturity Level 3 Process Area Technical Solution.	Maintain requirements traceability from a requirement to its derived requirements and allocation to functions, interfaces, objects, people, processes, and work products. Develop and maintain Product Support documentation. Review the requirements, design, product, and test results to ensure that issues affecting the installation, operation, and maintenance documentation are identified and resolved.
High-risk system settings and logical security settings are maintained to ensure adequate system and logical security.	3 of the 7 core information criteria for COBIT [®] address Information Security Confidentiality, Integrity and Availability.	DS5 Ensure Systems Security AI 2&3 Acquire and Maintain Technology and Application Infrastructure.

Table 4: Applying the Framework to SOX

Source: Infosys Research

Scenario 2: In this scenario a set of controls from FDA 21 CFR Part 11 are considered, though not in any specific sequence.

Requirement	Model Used	Implementation Guidance
A clear description of the product functionality or service supplied should be available.	CMMI [®] Maturity Level 3 Process Area Requirements Development.	Establish and maintain product and product component requirements, that are based on the customer requirements.
Tests should be traceable to requirement or design.	CMMI [®] Maturity Level 2 Process Area Requirements Management.	Maintain bidirectional traceability among the requirements and work products.
The architectural layout or description should be clear and available for review.	ITIL [®] ICT Infrastructure Management.	ICT Design and Planning provides a framework and approach for the Strategic and Technical Design and Planning of ICT infrastructures. It includes the necessary combination of business (and overall IS) strategy, with technical design and architecture.
An internal audit program should be in place.	COBIT [®] One of the 7 core information criteria for COBIT [®] .	Monitor and Evaluate Domain - ME3 Ensure Compliance with external requirements laws, regulations and contractual requirements.
An appropriate Quality Manual or SOP should be in place.	COBIT [®] Planning and Organization Control objective.	On a more specific level the Planning and Organization Control objective PO8: Manage Quality defines that Quality needs to be one of the Control Objectives right from planning stage. Within this CO PO8.1: Quality Management System mandates the need for a QMS in place. Related control objectives are PO4: Define IT Processes, Organization and relationships and PO4.7 Responsibility for IT Quality Assurance and AI2.8 Software Quality Assurance.
Communication of personal information with third parties is conducted via a secure medium that ensures the confidentiality and integrity of the information.	3 of the 7 core information criteria for COBIT [®] address Information Security Confidentiality, Integrity and Availability.	DS5 Ensure Systems Security AI 2&3 Acquire and Maintain Technology and Application Infrastructure.

Table 5: Applying the Framework to FDA 21 CFR Part 11

Source: Infosys Research

Scenario 3: In this scenario a set of controls from FDA 21 CFR Part 820 are considered. Though these controls are closely associated with the design, manufacturing and testing of healthcare devices, the software aspects of the design and development process are considered in identifying how the model components can be leveraged seamlessly.

Requirement	Model Used	Implementation guidance
Sub Part C Sec 820.30 Design Control: Procedures for Design and Development Planning, Design Input, Output, Review, Verification and Validation should be established.	CMMI [®] Level 3 Process Area Technical Solution.	Evaluate and select design alternatives that satisfy the requirements. Develop detailed designs for the selected alternative including the information needed to manufacture, code, or otherwise implement the design as a product or product component. Implement the designs as a product or product component. Additionally, the Generic Practices support planning and review of the design activities.
Sub Part D Sec 820.40 Document Controls : Procedures to control all documents should be established and maintained.	CMMI [®] Level 2 Process Area Configuration Management.	The specific practices listed below address this requirement. <ul style="list-style-type: none"> • Identify Configuration Items • Establish a Configuration Management System • Create or Release Baselines • Track Change Requests • Control Configuration Items • Establish Configuration Management Records • Perform Configuration Audits
Sub Part N Sec 820.200 Servicing: Procedures for servicing and verification should be established and maintained.	ITIL [®] Process areas: Service Level Management, Incident Management and Problem Management.	Service Level Management to set up and monitor service levels around solution / application performance availability, downtime etc. Incident Management to manage incidents related to system performance Problem Management to manage problems resulting from system performance


Table 6: Applying the Framework to FDA 21 CFR Part 820 *Source: Infosys Research*

CONCLUSION

With the growing challenges of cost competitiveness and meeting regulatory requirements, a healthcare IT organization has to instill confidence in the management and the business user community, that the software applications are not vulnerable to security threats and data leakage but are rather compliant to statutory requirements and would indeed perform at expected quality levels. To ensure these, the IT organization has to embark on a structured way to define, design and develop the software applications. While

there are many industry models providing best practices to deliver quality software applications, adopting a single model would not address all the specific requirements of quality, performance, compliance and security. Therefore it is imperative that healthcare IT organizations leverage on the best of these process models and methodologies and use an integrated framework such as the one suggested in this paper for process improvement, leading to software applications that are secure, compliant and of superior quality.

REFERENCES

1. John-David Lovelock, Yuko Adachi, Derry N. Finkeldey and Vittorio D'Orazio, *Forecast: Healthcare IT Spending, Worldwide, 2006-2011*, Gartner Research, August 2007. Available on www.gartner.com
2. *At Risk of Exposure*, Los Angeles Times, June 2006. Available at <http://www.latimes.com/features/health/medicine/la-he-privacy26jun26,1,3180537.column?ctrack=1&cset=true>
3. <http://www.connectingforhealth.nhs.uk/delivery/>
4. John-David Lovelock, *Dataquest Insight: Healthcare Industry Primer, 2006*, Gartner Research, April 2007. Available on www.gartner.com
5. Robert B Grady, *An Economic Release Decision Model: Insights into Software Project Management*, Proceedings of the Applications of Software Measurement Conference, Orange Park, Software Quality Engineering, pp 227-239, 1999
6. Capers Jones, *Measuring and Estimating Software Quality*, Software Productivity Research, 2007. 

Author Profile

RAVISHANKAR N

Ravishankar N is a Principal Consultant in Enterprise Quality Solutions Practice of Infosys and possesses 15 years of experience in various quality models and frameworks. He can be reached at RavishankarN@infosys.com

For information on obtaining additional copies, reprinting or translating articles, and all other correspondence, please contact:

Telephone : 91-80-41173871

Email: SetlabsBriefings@infosys.com

© SETLabs 2008, Infosys Technologies Limited.

Infosys acknowledges the proprietary rights of the trademarks and product names of the other companies mentioned in this issue of SETLabs Briefings. The information provided in this document is intended for the sole use of the recipient and for educational purposes only. Infosys makes no express or implied warranties relating to the information contained in this document or to any derived results obtained by the recipient from the use of the information in the document. Infosys further does not guarantee the sequence, timeliness, accuracy or completeness of the information and will not be liable in any way to the recipient for any delays, inaccuracies, errors in, or omissions of, any of the information or in the transmission thereof, or for any damages arising there from. Opinions and forecasts constitute our judgment at the time of release and are subject to change without notice. This document does not contain information provided to us in confidence by our clients.

Infosys[®]

POWERED BY INTELLECT
DRIVEN BY VALUES