Suggested/Recommended Audit Points in the Software Lifecycle (From thought to sunset)

Mr. Rick Brunner, CISSP
Assistant Vice President,
Security Strategy and Architecture
GM Financial
Disclaimer

The views, thoughts, claims or opinions in this presentation are solely those of the presenter.

Nothing in this presentation represents the views, thoughts, claims or opinions of GM Financial, Collin College, United States Air Force, the Air Force Reserves, the Department of Defense or the Intelligence Community.
Overview

• Define and gain an understanding into software and software lifecycle terms

• Describe various software development models, with emphasis on identifying similar phases within the various models

• Identify suggested/recommended Audit Points within the identified similar phases

• Provide some thoughts on Software as a Service

• Provide references for additional support or for conducting individual research
Internal Auditing

• Is an independent, objective assurance and consulting activity designed to add value and improve an organization's operations. It helps an organization accomplish its objectives by bringing a systematic, disciplined approach to evaluate and improve the effectiveness of risk management, control, and governance processes.

Information Technology Audit, or Information Systems Audit

• Is an examination of the management controls within an Information technology (IT) infrastructure. The evaluation of obtained evidence determines if the information systems are safeguarding assets, maintaining data integrity, and operating effectively to achieve the organization's goals or objectives. These reviews may be performed in conjunction with a financial statement audit, internal audit, or other form of attestation engagement.

Source: http://en.wikipedia.org/wiki/Information_technology_audit
Software

- Computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system.

Software Life Cycle

• The period of time that begins when a software product is conceived and ends when the software is no longer available for use. The software life cycle typically includes a concept phase, requirements phase, design phase, implementation phase, test phase, installation and checkout phase, operation and maintenance phase, and, sometimes, retirement phase. Note: These phases may overlap or be performed iteratively.

Software Life Cycle—Another View

Source: Jayesh Buwa, “Software Development Life Cycle (SDLC),” GlobalTech Solutions, Nashik
Software Development Cycle

• The period of time that begins with the decision to develop a software product and ends when the software is delivered. This cycle typically includes a requirements phase, design phase, implementation phase, test phase, and sometimes, installation and checkout phase.

Software Development Life Cycle

• Is a structure imposed on the development of a software product. Similar terms include software life cycle and software process. It is often considered a subset of systems development life cycle. There are several models for such processes, each describing approaches to a variety of tasks or activities that take place during the process. Some people consider a life-cycle model a more general term and a software development process a more specific term. For example, there are many specific software development processes that 'fit' the spiral life-cycle model. ISO/IEC 12207 is an international standard for software life-cycle processes. It aims to be the standard that defines all the tasks required for developing and maintaining software

Source: http://en.wikipedia.org/wiki/Software_development_process
“You’ve got to be very careful if you don’t know where you’re going, because you might not get there.”

Yogi Berra
SDLC Model

• A framework that describes the activities performed at each stage of a software development project.

• There are various software development approaches defined and designed that are used/employed during the software development process, these approaches are also referred as "Software Development Process Models."

Source: Jayesh Buwa, “Software Development Life Cycle (SDLC),” GlobalTech Solutions, Nashik
Source: Jayesh Buwa, “Software Development Life Cycle (SDLC),” GlobalTech Solutions, Nashik
Six Phases of Software Development Life Cycle for Software Development

- Requirements Gathering and Analysis
- Design
- Development
- Testing
- Implementation
- Maintenance

Source: Angelin, “Software Development Lifecycle (SDLC)”
SDLC Phases

Source: Angelin, “Software Development Lifecycle (SDLC)”
PMO Project Phases

- Initiate
- Plan
- Execute
- Close

Source: http://weill.cornell.edu/its/consultation/planning/pmo/pmo-process.html
SUGGESTED/RECOMMENDED AUDIT POINTS
# How can audit add value?

<table>
<thead>
<tr>
<th><strong>1. Navigate the integration risk landscape</strong></th>
<th><strong>2. Understand stakeholder perspectives and provide deeper insights</strong></th>
<th><strong>3. Cut through the clutter</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Questions</strong></td>
<td><strong>Questions</strong></td>
<td><strong>Questions</strong></td>
</tr>
<tr>
<td>How well aligned is internal audit’s plan with the critical risks facing the organization?</td>
<td>Does internal audit provide a point of view to help the business improve its responses to risk?</td>
<td>How effectively does internal audit communicate with stakeholders?</td>
</tr>
</tbody>
</table>

Source: Coke and PwC, “SDLC- Key Areas to Audit in IT Projects,” ISACA Geek Week 2013 8/21/2013
How can audit add value? Controls are often overlooked

Source: Coke and PwC, “SDLC- Key Areas to Audit in IT Projects,” ISACA Geek Week 2013 8/21/2013
Managing risk over the program lifecycle

Assess
- Is the ‘case for change’ robust with clear scope, business outcomes and ownership?
- Project governance and mgmt review
- Planning and mobilization
- Business case review
- High level target operating model
- Organization change strategy
- Deployment strategy

Design
- Will the organization & technical design deliver the benefits?
- Business process design
- Data and reporting design
- Test and data conversion strategies
- Security & controls
- People and Org Design
- Dedicated vendor management

Construct
- Is the solution being built as designed and robustly tested?
- Solution testing and remediation
- Training plans and execution
- Data conversion
- Security and control configuration
- Business continuity planning
- Benefits management plan
- Support model design

Implement
- Is the business ready to go with detailed go live and support plans in place?
- Test and training results
- Go-live process
- Data conversion process
- Transition to business as usual (BAU) planning
- Stakeholder engagement
- Go-live readiness assessment

Operate & Review
- Are the benefits being delivered and what could be improved?
- 30-90 day support
- Business adoption
- Benefits realization
- Compliance and controls certification

Is the Change Management approach appropriate and delivering success?

Is the organization engaging key stakeholders (including existing vendors/partners) throughout the change?

Is the program being effectively governed against guiding principles and managed across all workstreams?

Is delivery of business benefits a key focus throughout the lifecycle?

Source: Coke and PwC, “SDLC- Key Areas to Audit in IT Projects,” ISACA Geek Week 2013 8/21/2013
Key events for a successful Project Audit

• Stakeholder buy-in & tone at the top, understanding and acceptance of engagement
• Staffing, proper technical skills, qualifications and capabilities allowing the team to quickly establish credibility
• Understanding project needs and expectations, as well as the level of comfort desired
• Scoping appropriately, leveraging a risk-based approach and delivering upon the agreed scope
• Up-front communication regarding scope of review, extent of review, timing of review and level of details to be provided in reporting

Source: Coke and PwC, “SDLC- Key Areas to Audit in IT Projects,” ISACA Geek Week 2013 8/21/2013
Key events for a successful Project Audit

• Execution and completion of work within defined budget and schedule
• Change agility, being able to change with the project needs (adjust timeline, etc.) but avoiding scope creep
• Communication to all parties
• Relevance, providing actionable, useful and timely deliverables (reporting) – consider requirements of the audience (i.e. Audit Committee, Sponsor, Project Manager, etc.)
• Monitoring project progress between checkpoint reviews to minimize ramp-up time required at each checkpoint

Source: Coke and PwC, “SDLC- Key Areas to Audit in IT Projects,” ISACA Geek Week 2013 8/21/2013
Software Configuration Management

Operation Aurora

Software Task Scheduling

• **Rule of Thumb**
  – 1/3 planning
  – 1/6 coding
  – 1/4 component test and early system test
  – 1/4 system test, all components in hand

• **Conventional scheduling**
  – The fraction devoted to planning is larger than normal. Even so, it is barely enough to produce a detailed and solid specification, and not enough to include research or exploration of totally new techniques
  – Half of the schedule devoted to debugging completed code is much larger than normal
  – The part that is easy to estimate (i.e., coding) is given only one-sixth of the schedule.

Software Costs

• 60 percent of cost is spent on maintenance
  – 60 percent of software maintenance costs is spent on enhancements
  – 40 percent on actual maintenance
    • 17 percent on corrections
    • 18 percent on adaptive maintenance, making it work in its environment
    • 5 percent on other, preventive maintenance
• The 60/60 rule: 60 percent of software’s dollar is spent on maintenance, and 60 percent of that maintenance is enhancement. Enhancing old software is, therefore, a big deal.

Quality – the degree to which the software satisfies stated and implied requirements

- Absence of system crashes
- Correspondence between the software and the users’ expectations
- Performance to specified requirements

Quality must be controlled because it lowers production speed, increases maintenance costs and can adversely affect business
Quality v Development Time

- Higher quality (in the form of lower defect rates) and reduced development time go hand in hand.

Quality Assurance Plan

• The plan for quality assurance activities should be in writing
• Decide if a separate group should perform the quality assurance activities
• Some elements that should be considered by the plan are: defect tracking, unit testing, source-code tracking, technical reviews, integration testing and system testing.

Source: condor.depaul.edu/jpetlick/extra/394/Session2.ppt
Quality Assurance Plan

• Defect tracking – keeps track of each defect found, its source, when it was detected, when it was resolved, how it was resolved, etc.
• Unit testing – each individual module is tested
• Source code tracing – step through source code line by line
• Technical reviews – completed work is reviewed by peers
• Integration testing – exercise new code in combination with code that already has been integrated
• System testing – execution of the software for the purpose of finding defects

Source: condor.depaul.edu/jpetlick/extra/394/Session2.ppt
Are you sure you built it right, are you sure you built the right thing

Source: Jayesh Buwa, “Software Development Life Cycle (SDLC),” GlobalTech Solutions, Nashik
Verification and Validation

- Verification – The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.

- Validation – The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements.

Another Look at the Definitions

• Validation – was the right system built?

• Verification – was the system built right?

• Quality Assurance – was the system built the right way?
Requirements

• One of the two most common causes of runaway projects is unstable requirements

• Requirements errors are the most expensive to fix when found during production but the cheapest to fix early in development

• Missing requirements are the hardest requirement errors to correct
  – The most persistent software errors—those that escape the testing process and persist into the production version of the software—are errors of omitted logic. Missing requirements result in omitted logic

• When moving from requirements to design, there is an explosion of “derived requirements” (the requirements for a particular design solution) caused by the complexity of the solution process. The list of these design requirements is often 50 times longer than the list of original requirements

Cost of Correcting Defects by Lifecycle Phase

As a rule of thumb, every hour an organization spends on defect prevention reduces repair time for a system in production by three to ten hours. In the worst case, reworking a software requirements problem once the software is in operation typically costs 50 to 200 times what it would take to rework the same problem during the requirements phase.

Configuration Management - First Part

“A management process for establishing and maintaining consistency of a product’s performance, functional and physical attributes with its requirements, design and operational information throughout its life”

ANSI/EIA 649 1998
National Consensus for Configuration Management
A process intended to ensure that the system performs as intended, and is documented to a level of detail sufficient to meet needs for operation, maintenance, repair and replacement
In other words....

The primary goals of configuration management are:

1) Establish System and project/product integrity
2) Maintain this integrity throughout the lifecycle
Configuration Management Principle

- Information Assurance supports Configuration Management.
- A system or capability that does not have sound Configuration Management process and procedures in place and positively acted upon has little to no Information Assurance.
- The Information Assurance posture of a system or capability cannot be maintained or determined without sound Configuration Management process and procedures in place and positively acted upon.

Source: Anonymous
## Managing risk over the program lifecycle

<table>
<thead>
<tr>
<th>Step</th>
<th>Assess</th>
<th>Design</th>
<th>Construct</th>
<th>Implement</th>
<th>Operate &amp; Review</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Is the 'case for change' robust with clear scope, business outcomes and ownership?</strong></td>
<td><strong>Will the organization &amp; technical design deliver the benefits?</strong></td>
<td><strong>Is the solution being built as designed and robustly tested?</strong></td>
<td><strong>Is the business ready to go with detailed go live and support plans in place?</strong></td>
<td><strong>Are the benefits being delivered and what could be improved?</strong></td>
<td></td>
</tr>
<tr>
<td>Delivering Change</td>
<td>- Project governance and mgmt review</td>
<td>- Business process design</td>
<td>- Solution testing and remediation</td>
<td>- Test and training results</td>
<td>- 30-90 day support</td>
</tr>
<tr>
<td></td>
<td>- Planning and mobilization</td>
<td>- Data and reporting design</td>
<td>- Training plans and execution</td>
<td>- Data conversion process</td>
<td>- Business adoption</td>
</tr>
<tr>
<td></td>
<td>- Business case review</td>
<td>- Test and data conversion strategies</td>
<td>- Data conversion</td>
<td>- Security and control configuration</td>
<td>- Benefits realization</td>
</tr>
<tr>
<td></td>
<td>- High level target operating model</td>
<td>- Security &amp; controls</td>
<td>- Business continuity planning</td>
<td>- Business as usual (BAU) planning</td>
<td>- Compliance and controls certification</td>
</tr>
<tr>
<td></td>
<td>- Organization change strategy</td>
<td>- People and Org Design</td>
<td>- Benefits management plan</td>
<td>- Stakeholder engagement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Deployment strategy</td>
<td>- Dedicated vendor management</td>
<td>- Support model design</td>
<td>- Go-live readiness assessment</td>
<td></td>
</tr>
<tr>
<td>Driving Change</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Is the Change Management approach appropriate and delivering success?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Is the organization engaging key stakeholders (including existing vendors/partners) throughout the change?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Is the program being effectively governed against guiding principles and managed across all workstreams?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Is delivery of business benefits a key focus throughout the lifecycle?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source: Coke and PwC, “SDLC- Key Areas to Audit in IT Projects,” ISACA Geek Week 2013 8/21/2013
Software as a Service – Some Thoughts

- Transparency
- Privacy
- Compliance
- Transborder Information Flow
- Certification
- Other key items
  - Events – The service provider should regularly document and communicate changes and other factors that have affected SaaS system availability
  - Logs – A service provider should provide comprehensive information about an enterprise’s SaaS application and runtime environment
  - Monitoring – Any such surveillance should not be intrusive and must be limited to what the cloud provider reasonably needs in order to run its facility

Cloud Computing Security and Privacy Challenges

- Privileged user
- Regulatory compliance
- Data location
- Data segregation
- Recovery
- Investigative support
- Long-term viability

References

• Books