Why SDLC Controls are important for a project

Jason D. Lannen  CISA, CISM
August 21, 2013
11:15 AM
Jason D. Lannen  CISA, CISM

• Founder and Managing Director at TurnKey IT Solutions LLC (established in 2009)
• More than nine years of IT audit and regulatory compliance experience
  – Managing and executing external and internal audits
  – Executing testing for CPA firms (SOC 1 / SOC 2)
  – Advising companies in security and compliance
  – COBIT 5 Peer Reviewer
  – Managing pre-implementation SDLC controls
Opening Quote

• “At his best, man is the noblest of all animals; separated from law and justice he is the worst.”

~ Aristotle
Why are SDLC Controls Important?
Humans need laws and rules\textsuperscript{4}

- We long for consistency and routine
- We need structure for things that are not structured
- We long for change
Organization’s have to properly manage their risk

- Reputational
- Financial
- Operational
- Personnel
- Information Security
- IT
- Legal & Regulatory Compliance
Why are SDLC controls important?

✓ Establish a framework for building, implementing and enhancing systems that all personnel have to follow
Why are SDLC controls important?

✓ Create accountability for IT and business management by requiring documentation and signoffs
Why are SDLC controls important

- Regulatory Compliance
  - SOX
  - SOC 1 / SOC 2
  - Gov’t Regulation
Why Do Projects Fail?
Why Do Projects Fail?

- How's your project coming along?
- It's a steaming pile of failure.
- It's like fifteen drunken monkeys with a jigsaw puzzle.
- How's your project coming along?
- Fine.

© Scott Adams, Inc./Dist. by UFS, Inc.
Why Do Projects Fail?³

- **Requirements**: Unclear, lack of agreement, lack of priority, contradictory, ambiguous, imprecise.
- **Resources**: Lack of resources, resource conflicts, turnover of key resources, poor planning.
- **Schedules**: Too tight, unrealistic, overly optimistic.
- **Planning**: Based on insufficient data, missing items, insufficient details, poor estimates.
- **Risks**: Unidentified or assumed, not managed.
- **POOR COMMUNICATION!**
Why Do Projects Fail?  

- The most common obstacles that interfere with recovering failed projects are:
  - 😞 Getting stakeholders to accept the changes required
  - 😞 Poor communication and stakeholder engagement
  - 😞 Conflicting priorities and politics
  - 😞 Finding enough qualified resources needed to complete the projects.
  - 😞 Lack of a process or methodology to help bring the project back on track
What are Key’s to Project Success?
Keys to Project Success

✔ Top management support
✔ A sound methodology
✔ Solid project leadership
  ✔ IT Project Management Office
  ✔ Tactical IT Management
  ✔ Business Management
Key SDLC Phases
Key SDLC Phases

Initiation

Inputs
- Identify Problem and End Solution
- SDLC Framework Used
- Budget

Outputs
- Project Milestones & Dates
- Initial Project Plan
- Establish Project Charter

Sign off to move to definition

Definition

Inputs
- Business Req’s
- Functional Req’s
- Technical Req’s

Outputs
- Gap Analysis
- Business / Functional / Technical Req
- Documentation
- Traceability Matrix

Sign off to design and build

Design & Build

Inputs
- Customize, Code and Configure System
- Develop user interface
- Unit Testing

Outputs
- Develop Test Cases & Test Scripts

Sign off to test

Test

Inputs
- Unit Testing
- System Integration Testing
- Regression Testing
- User Acceptance Testing

Outputs
- Sign off to begin deployment activities

Sign off to test

Implement

Inputs
- Deployment Plan
- Countdown activities
- Final Signoffs by Business Management

Outputs
- Go Live!
- Post Implementation Support

Change Management

Security & Documentation Repositories

Issue Tracking

Data / Interface / Reports Validation

Configuration Mgmt and Controls Development
Change Management

• Establish a formal change management process when business needs change, functionality / processing errors take place, security requirements are added / changed, infrastructure changes, etc
Change Management

• Should encompass the following documentation attributes:
  – CR #
  – Description of the Change
  – Impact Analysis
  – Testing
  – Signoff by applicable parties

• Key considerations:
  ✓ Stored in a secured central repository
  ✓ Traceability to change documents & issue logs
Security

• Logical access should be appropriately controlled for:
  – System administrative functions, configurations and environments
  – Data used for testing
  – SDLC project documentation repositories

• Physical access should be restricted to
  – Systems used for SDLC development and testing
  – SDLC project documentation
Security

• **Key Considerations**
  
  ✓ Logical and physical access to all SDLC related information and data should be restricted to appropriate personnel on a need to know basis.
Document Repositories

• Where SDLC project documentation is stored
  – Project Plan, Bus / Functional / Tech Req’s
  – Test Plans / Test Scripts
  – Signoffs

• Should be tightly controlled through physical and logical access measures, especially confidential information
Document Repositories

• Document repositories should be backed up on a regular schedule

Key Considerations:

✓ Document retention should follow Legal and Project requirements

✓ A process for non-compliance should be established and be enforced.
Issue Tracking

• Create a central repository of issues and document:
  – Issue ID
  – Description of the Issue
  – Who identified the issue
  – Status
  – Remediation Plan & Date
  – Remediation Results & Date

• Key Considerations
  ✓ Establish and have an effective issue monitoring team for trending and impact analysis
  ✓ Issues should tie to change requests / test scripts / affected requirements and validation documentation
I DON"T ALWAYS TEST MY CODE

BTU WHEN I DO, IT IS IN PRODUCTION!
Configuration Management

• Establish a pristine environment which other environments can be refreshed from
  – Application
  – Data

• If different environments with different configurations are used for certain scenarios, this should be clearly documented and be approved by relevant IT and affected business parties, but used sparingly
Configuration Management

• Key Considerations:

✓ Don’t just ask if there was a separation of environments for SDLC, understand what was different between the environments and how management obtained comfort over the activities performed within them.

✓ There should be strong security controls in place for who knows user accounts and passwords in dev / test environments and repercussions for those who break security and change management policy.
I'd like to thank all of the people who helped design the technology test parameters.

Thanks to your input, the test had nothing in common with how things work in the real world.

So I wasted two weeks of my life on a test that is not only meaningless... but also dangerously misleading.

This slide shows the gap between the test results and reality.

We'll use the test results anyway because it's the only data we have.

Fine. I hope you all choke to death on your lunches.

Why's he so cranky? Something about data.
Data, Interfaces and Reports

• Data Conversion / Data Validation
  – Data Conversion strategy
  – Conversion team completeness and accuracy check
  – Error handling and resolution

• Data Validation
  – Process of validation
    • Sampling / Visual Review
    • Mass Data Validation
  – Error handling and resolution
Data Interfaces and Reports

• Interfaces & Reports
  – Was real or fake data used
  – Were the interfaces tested for functionality or transactability
  – Was validation over completeness and accuracy performed
Controls Development

- Understand the business processes (old / new)
- Map controls to business and system requirements and testing performed
- Work with management to determine the key financial / operational controls. Importance should be stressed on the regulatory compliance aspects and impact to business operations
- Risk assess controls
Top 10 Ways to Guarantee the Failure of a Project

10. Don’t use a specific methodology because coding is all that is really important.
9. Create the project plan by working backwards from a drop-dead system completion date.
8. Don’t bother with a data model. Just build whatever tables you need.
7. Use a Technical Lead that has never built a similar system. Hiring such talent is too expensive.
6. Hire forty developers to make the coding go faster.
Top 10 Ways to Guarantee the Failure of a Project

1. Buy a commercial, off-the-shelf package and customize it a lot.

2. Change the system to support critical new requirements discovered during final development.

3. Skip the testing phase because the project is way behind schedule.

4. Three months before the system goes live, assign one junior developer to handle the data migration.

5. Build the system in Java, even though most of the development team still thinks that Java is coffee and you have no intention of ever deploying to the Web.
Questions
Contact Information

• Jason D. Lannen  CISA, CISM
  – Managing Director

Phone: 770-680-2650
Email: Jason.Lannen@turnkeyit.net
Twitter: @turnkey_it
Website:  www.turnkeyit.net
References