ISPE
Los Angeles Chapter

# Change Control and Configuration Management

Jerry Anderson

JAnderson@Watson.com

# Bio (or, "Who is this guy?")

- Director, Watson Corporate Computer Compliance
- 24 years doing/managing IT (13 in GxP-regulated environment)
  - Development of validated applications
  - Network design, operations, & qualification
  - System, database, & data center administration
  - Global Part 11 program management
- IT instructor
- BS Computer Science; MBA; ASQ CSQE
- Soccer referee; Ultimate Frisbee junkie

# Agenda

- **Configuration Management**
- Change Control
- Making It Work for You

"Those who do not remember the past are condemned to repeat their mistakes."

- George Santayana

# What is Configuration Management?

- CM is a discipline for managing and controlling the evolution of a system
  - During development
  - In production
  - During maintenance & use
  - Through retirement
- CM identifies and documents the functional and physical characteristics of a system, and controls changes to those characteristics

# CM: The Bottom-Line

***CM is the process you use to understand, document, and control the ingredients of your system.***

# Some Relevant References

- ISO/IEC 12207, Standard for IT Software Life Cycle Practices
- IEEE 828, Software Configuration Management Plans
- IEEE 1042, Software Configuration Management
- BSI BS 6488, Configuration Management of Computer-Based Systems

# Why Do CM?

- To manage a system well, you have to know how it's built
- In order to know what you've got *after* a change, you have to know what you had *before* the change
- To find & fix a problem, you usually have to know in some detail what your "It works!" configuration was

# Why Do CM?

- Configuration management gives visibility into the state and status of a system
- Such visibility is important during:
  - Development, to end up with the right thing
  - Production, to keep systems & services available
- CM answers the following questions:
  - What constitutes "the system" at any point in time?
  - What changes have been made to get us here?

# What Inspectors Will Look For

## From recent observations and warning letters:

"Written procedures to differentiate between revision or version changes are not employed."

"… lack of verification that software modifications validated on the "test" system are identical to the modifications implemented later in the "live" system."

"There is no documented evidence to indicate that noted problems in the currently distributed software version xxx have been corrected, investigated or tested …"

# Configuration Management Plan

- Specifies how you will identify and manage the elements of your system
- Outlines how you will handle versioning, baselines, changes to baselines
- If you use automated CM tools, your plan talks about HOW you'll use them

# Formal Configuration Management Activities

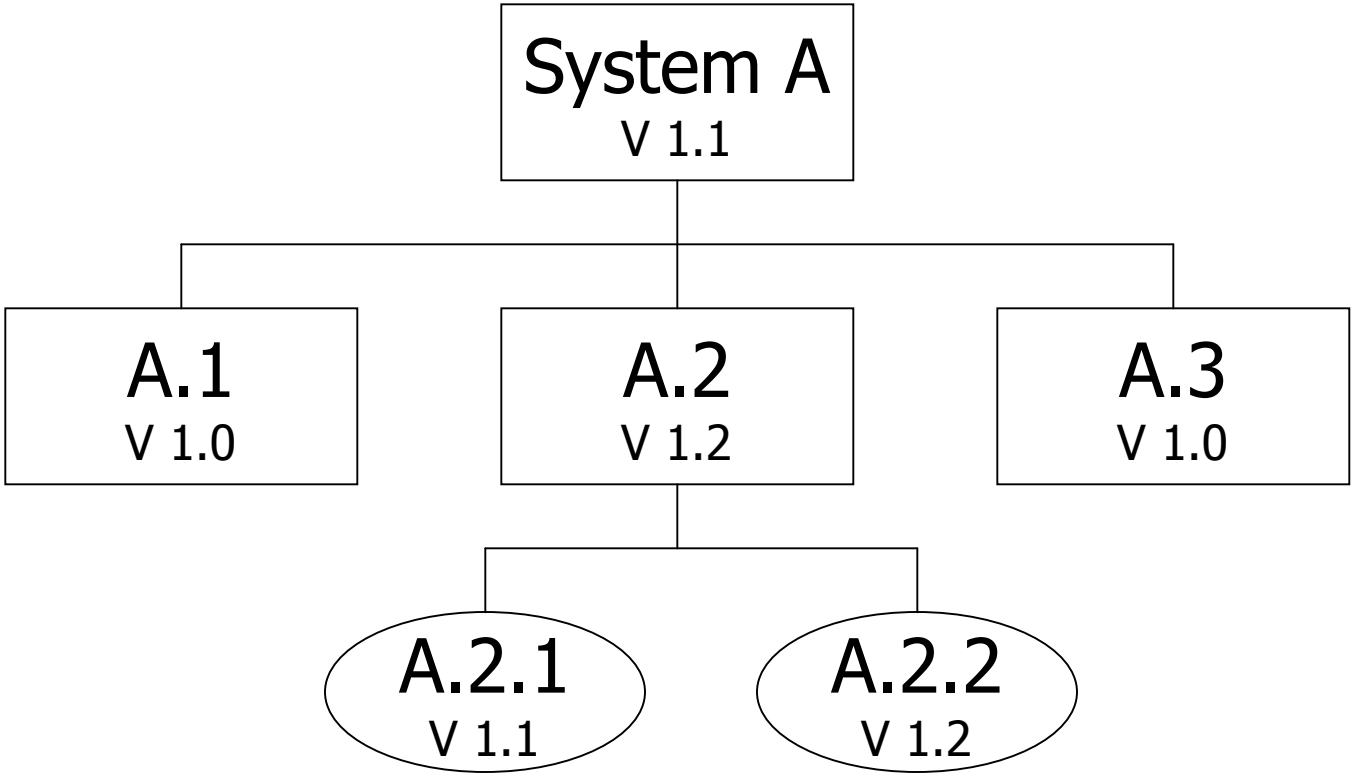
1. Configuration Identification
   - What do we have, and how is it structured?

2. Configuration Control
   - Another way of saying 'change control'
3. Configuration Status Accounting
   - Documents on configuration, versions, change history
4. Configuration Auditing
   - Review of documents and systems

# 1. Configuration Identification

- Identify and uniquely name the components ("configuration items" or "CI") of your system
- Give version numbers to each individual CI
- Establish 'configuration baselines' (e.g. freezes) by identifying the structure of how CIs interconnect to form your system
- Making revisions to your CIs or structure causes revisions to the configuration baseline
- This creates the basis for change traceability

# Applied Configuration Identification




A simple baselined system

# A Note on System Baselines

- When you baseline a system, you are 'freezing' it
  - You're creating a snapshot of your system
  - You know the versions and functionality of all CIs
  - You know how all CIs interconnect
  - You know the version and functionality of the system

- Baselines are good to establish just before:
  - Integration testing
  - Acceptance / qualification testing
  - Release

# 2. Configuration Control

- A process for managing changes to an established system baseline
  - Assign change roles and responsibilities
  - Process change requests
  - Specify required change activities
  - Track change history and approvals
  - Establish new system baselines
- "Configuration Control" is another way of saying "Change Control" which we'll talk about in more detail later

# 3. Configuration Status Accounting

- Simply, a paper trail (or electronic trail) of documentation showing the history of configuration items and changes to them
- Should include dates, component identifications, versions, baseline descriptions, change details, information about the people involved
- May be the result of report queries if an automated CM system is used

# 4. Configuration Auditing

- Also known as "Configuration Evaluation"
- Just what it sounds like: an audit process that evaluates the effectiveness of the configuration management system
- Reviews the trail generated by Status Accounting
- Verifies that the system's current configuration matches what the CM system claims

# Agenda

- Configuration Management
- **Change Control**
- Making It Work for You

"Chaos is perhaps at the bottom of everything."

- George Santayana

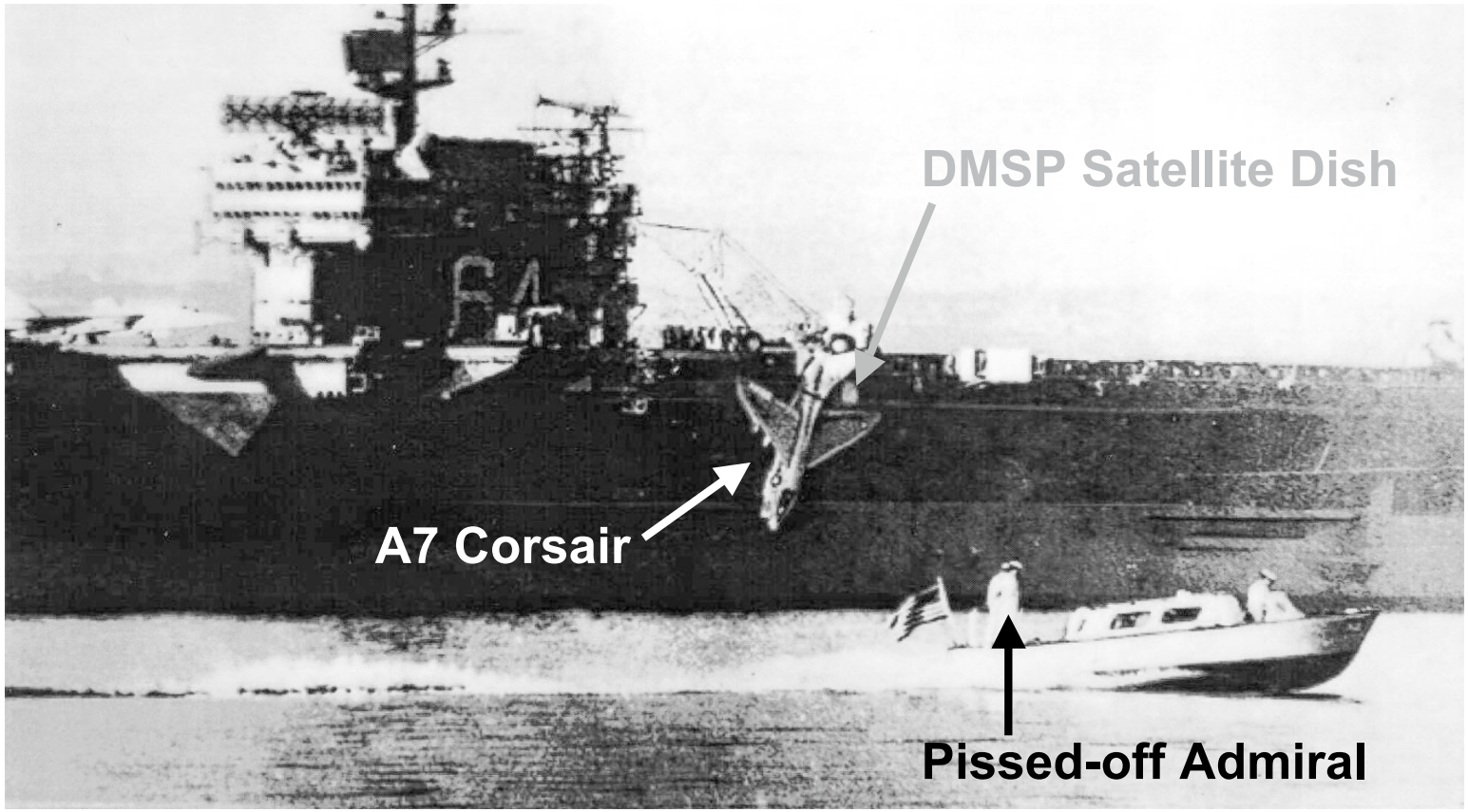
# Change Control

## CV-64 USS Constellation

# Change Control

## *True Story:*

**Defense Meteorological Satellite Program system is removed from ship, but antenna is left on due to CM error. Taxiing jet goes over the side... but catches on the left-over dish and is eventually recovered.**



# What is Change Control?

- As previously mentioned, this is also known as the "Configuration Control" activity of CM
- Change control is the process used to request, review, plan, approve, and implement changes to a system
- When it's properly implemented, change control assures that unplanned changes don't happen, and that planned changes are well-managed

# Change Control: The Bottom-Line

***Change control is the process you use to keep your system in a known good state…***

***… and the documentation that PROVES you're doing it.***

# Business Drivers

- Change control is expensive, but it makes good business sense:
  - Fewer service outages
  - Ability to reliably build "add-on" services
  - Ability to predict the result of changes
  - Ease of knowledge transfer
  - Better reputation for the IT group
- The FDA doesn't worry about some of these factors, but <u>you</u> should

# Compliance Drivers

- Remember that when they look at your system, the first thing inspectors will ask for is your validation documentation
- If you validated or qualified the system a year ago, then your doc set is a picture of the nice, validated state of your system… a year ago
- Inspectors know this, and will want to look at the documentation for changes that have happened since then

# What Inspectors Will Look For

- Change control:

  " … Failure to establish and maintain a change control system for changing documentation, procedures, specifications, or modifying the software source code."

  "Your firm fails to classify or evaluate changes, modifications… according to significance."

  "Failure to validate each significant change in the computer software…"

(Source: recent observations and warning letters)

# GAMP and Change Control

- CM & CC guidance from various sections of GAMP 4:
  - System configuration management
  - Change control during system development
  - Operational change control
- Also references ISO 9000:2000:
  - Identification and traceability
  - Control of design and development changes
  - Control of documentation

# ISO 9001-3 and Computer Change Control

- CM & CC guidance from various sections of 9001-3:
  - Configuration management used to identify software, hardware, or documentation items in all phases of development
  - Configuration management is used to control documents and data
  - Place implementation tools under configuration management control
  - Procedures must be in place to handle change requests
  - Changes shall be identified, reviewed, and approved by authorized personnel
  - A configuration management system may be used to handle this area

# Change Control: Minimum Required Components

- A quality system
- A written change control procedure
- A change request form
- One or more change review teams
- A computer validation group or function
- A quality assurance group or function

# Quality System

- Specifies all aspects of QA and QC in your organization
- Comprehensiveness is required because a change control process alone doesn't cover all the bases
  - Validation
  - Vendor management
  - Training requirements
  - Employee responsibilities
  - System operations
  - Incident/problem management
  - Software development standards
  - Software quality assurance
  - Security
  - Business continuity

# Change Control Procedure

- One component of the quality system is the change control procedure and form
  - An SOP for the overall change control process
  - Possibly a "sub" SOP covering computer changes
- Everyone who has the capability to make changes to the validated target (production) environment must be trained on the change control procedure
- If an automated system is used, it must be validated

# Change Review Team: Business Owner

- Decide whether changes are worth doing
- Evaluate business justification
- Prioritize requests within the business function
- Work with others to prioritize requests across functions

# Change Review Team: Technical Representative

- Assure that change procedures are followed
- Classify changes by type, risk
- Determine regulatory requirements and qualification requirements
- Give the go or no-go to make a change
- Determine whether an implemented change should get final approval or be rolled back

# Change Review Team: CV

- Reviews each change to determine the impact to the validated system
- Specifies how new functionality is to be tested
- Specifies how old functionality is to be re-tested
- Specifies required documentation updates

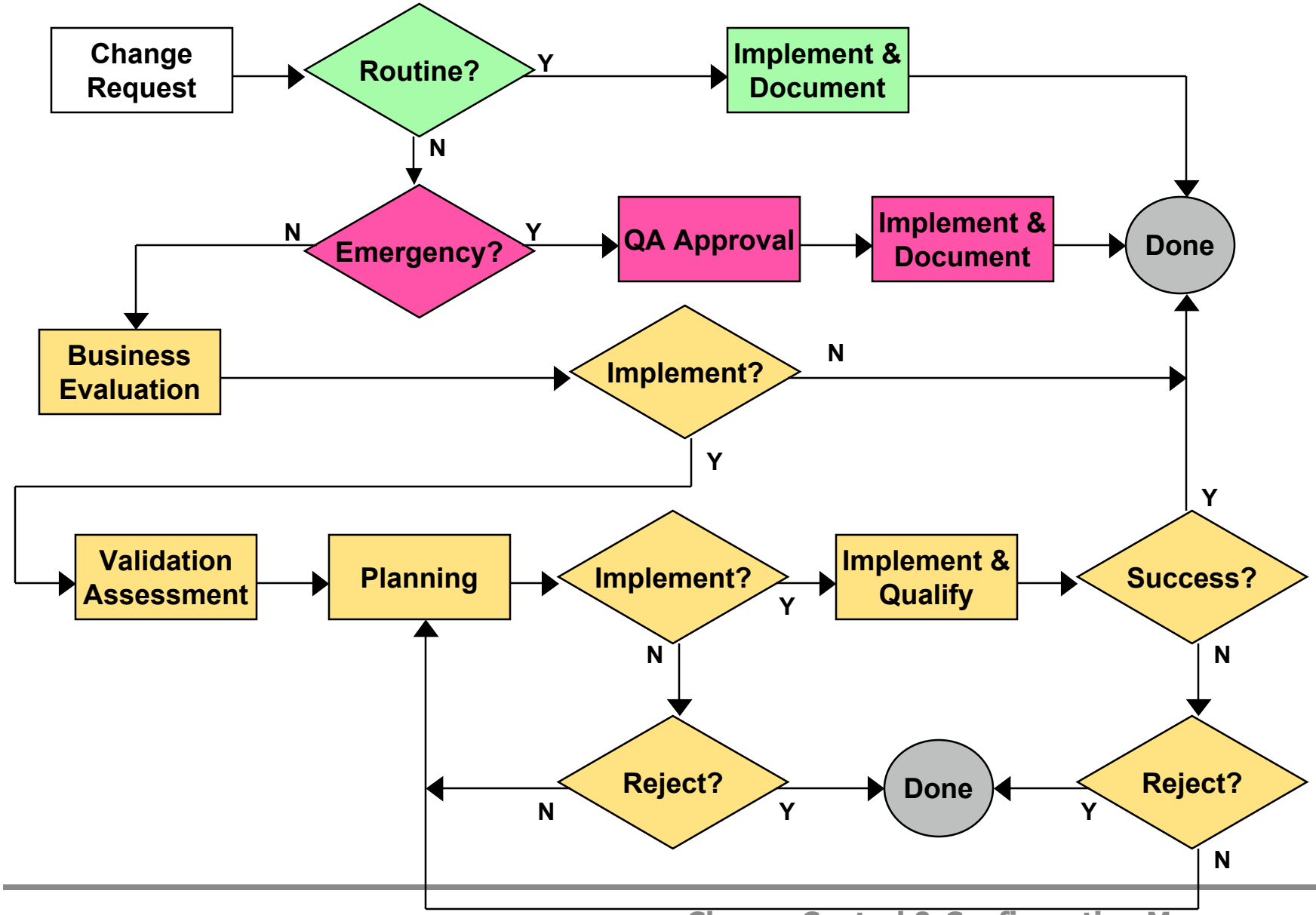
*Each change to a validated system typically requires a mini revalidation / requalification effort*

# Change Review Team: QA

- Assures compliance with regulatory requirements and internal procedures (change control, validation, documentation practices, etc.)

- QA  indicates that everybody did their jobs correctly, and that the change package forms a proper regulatory document

# Change Control Process Flow



# The Change Control Procedure

1. Change request is submitted
2. Change is evaluated
3. Validation assessment
4. Implementation planning
5. Approval to implement
6. Implementation and qualification
7. Approval to close change
8. Special treatment of “emergency changes”
9. Special treatment of “routine changes”

# 1. Change Request

- Anyone can submit a change request
- Information
  - Unique change request number
  - Change requester/owner
  - Description
  - Reason for change
  - Impact and scope of change (systems, processes, locations)
  - Does the change require notification to or approval by regulatory agencies?

# 2. Business Evaluation of Change

- Key stakeholders, system owners, and cognizant managers do this, helped by the change owner and subject matter experts
  - Is the change justified and feasible?
  - Verify impact and scope of change
  - Verify regulatory requirements
- "High-level" check to see if the change is worth devoting resources to
- If yes, communicate the intent to make the change to key stakeholders
- Prioritize this request against others in the queue

# 3. Validation Assessment

- Validation and Quality groups look at the regulated systems, data, processes in the scope of the change
  - What functionality is being added, modified, retired?
  - What is the relative criticality of this functionality?
  - What other areas of functionality *might* be impacted?
- Output of the assessment:
  - SQE and SQA activities
  - Required testing (new and regression)
  - Required documentation updates

# 4. Implementation Planning

- Using input from the validation group, the change owner and technical folks plan the change
  - Project plan and risk management
  - Procurements, technology, and tools
  - Assignment of technical and compliance activities
  - Documentation and training
  - How to smoothly promote to production
  - How to roll back the change if necessary
- Actual development work in the DEV/TEST environments

# 5. Approval to Implement

- Change board and key stakeholders review the implementation plan
- Change board gathers input and makes a decision
  - Approved for implementation
  - Rejected
  - Go away now; come back with a better plan
- If approved for implementation, the change can be made in the target production environment when all required activities are complete

# 6. Implementation and Qualification

- If not yet fully developed, technical work continues here
- Development/integration testing is completed
- IQ is executed/re-executed as required
- Change is promoted to production
- All or part of OQ/PQ re-executed as required
- Summary report of success is written (or, unsuccessful change is rolled-back)

# 7. Approval to Close Change

- The change board meets again to decide:
  - If the change worked, should we keep it?
  - If the change failed, should we reject it or go back to the drawing board and try again?
- If the change is to be kept:
  - Approvals from change board, QA, validation
  - Approval, if required, from regulatory agencies

# 8. Emergency Changes

- There must be an "out" clause that allows you to make changes required to fix things that are
  - Broken
  - Just about to break
- The emergency process:
  - Implement and test the fix
  - Document the change
  - Get approval to close the change

# Agenda

- Configuration Management
- Change Control
- **Making It Work for You**

"Insanity: Doing the same thing over and over again and expecting different results."

- Albert Einstein

# Controlling Change Volume

- More change = more risk to the system's controlled state
- Limit change by forcing the business/users to choose what changes are implemented
  - Educate users on the compliance reasons for limiting change
  - Implement chargeback for the cost of changes, or identify the resources in IT, QA, CV that are available to support changes to establish the 'change budget'
  - Change requests should include a business/compliance justification, with criteria that prevent "Because I want it!" changes
  - On cross-functional (e.g. SAP) or multi-site (e.g. LIMS) systems, set up a team of business leaders that can prioritize (and reject) changes across functions

# Pre-Defined Change Risk

- Systems have a risk designation based on the risk of the process being automated
- Many systems also have different levels of risk associated with different system function points. Examples:
  - Changing 'standalone' code vs. code that's shared across modules
  - ERP: GxP vs. non-GxP modules and transactions
- Different types of changes have different risk/impact
- Can be difficult, complex, and time consuming to do up-front, but will pay dividends in the future when changes force you to requalify the system

# Pre-Defined System Testing

- If you pre-define risk by module/transaction, you can pre-define the testing that must occur when the module/transaction is modified
  - Pre-define the requalification deliverables that will be required based on the impact of various change types
  - Partition the system OQ into mini-OQs that can be run separately for regression testing at the right level of granularity to requalify after a change
  - Note: adding new functionality always requires new test challenges
- If feasible for your organization, automated testing tools (e.g. Mercury) can dramatically speed up qualification testing, especially when you have to execute the full OQ

# Routine Changes

- A pre-defined list of changes that can be implemented without 'formal' change control
- Pre-approved by QA; need no hands-on QA or CV involvement
- Typically low-risk/low-impact activities
- Examples:
  - Reboot server or restart application/process
  - "Like-for-like" hardware changes (e.g. disk in RAID 5 array)
  - Change start time of batch job
  - Some types of performance tuning

# Routine Changes

- Caveats
  - Be very clear and specific in your wording to prevent misinterpretation (e.g. swapping servers is not a 'like-for-like' hardware change)
  - Some routine changes must still be tested in a non-production environment
  - Some (maybe all) require an SOP to be in place to guide the people implementing the change
  - All require some type of documentation (e.g. work request, system log)
  - History of routine changes must be accessible
  - Requires periodic auditing

# A Note About Change Control Documentation

- Each change is a mini development and qualification effort
- When you're done with a change, the approved change request plus associated documentation form a regulatory document
- System change packages should be filed along with the original system validation/qualification documents
- This combination shows how you got and kept your system in compliance

# Get Your IT Group Thinking "GxP!"

- Odds are good that the areas we've discussed are managed by IT people who've had little exposure to CFRs or computer validation
- Now, they will at minimum be responsible for maintaining the qualified state of your network
- How do you build a compliance mindset in those groups?

# Get Your IT Group Thinking "GxP!"

- First: Enlist IT management support
- Second: Educate on the issues
  - The requirement for compliant systems and networks
  - The risks if compliance is not achieved and maintained
  - What they can do to help
- Third: Training
  - Basic GxPs; good documentation procedure
  - Required SOPs: validation, change control, operations, security, etc.
- Fourth: Audits!
  - Periodically audit IT groups to assure that new employees have been trained, that everyone is keeping up on SOP reading, etc.

# Questions?