

Lotus knows.

Smarter software for a Smarter Planet.

BP111

Heel! Sit! Code! Obedience/Compliance Training for Your Developers

Eileen Fitzgerald | VP, GSX

Jamie Magee | Director, MartinScott Consulting



CREATED WITH LOTUS® SYMPHONY™



lotusknows.com

Who & Why?

- Developers - Make life easier
- Admins - Makes life easier
- Developers & Admins - Reduces body count & promotes world peace
- Management - Compliance, reduces costs

Introduction

- Eileen Fitzgerald
 - VP For Product Development and Customer Services at GSX
 - Leader of team of Developers and Admins delivering Domino as a service via ITIL
 - Global Notes Architect and Ops manager
 - ITIL Problem Manager, & Release Manager
- Jamie Magee
 - Partner at MartinScott Consulting
 - Domino development team leader and project manager

Agenda

- What is Release Management
- Approach to presentation
- Steps associated with Release Management
- Reality
- Walk through step by step
- Wrap up – questions

What is Release Management?

- The protection of the live environment and its services through the use of formal procedures and checks
- Relevant to both bespoke (custom) and shrink-wrapped software
- Applicable to both software changes and hardware changes

What is Release Management?

- Incident Management & Problem Management are important feeds into Release Management
- You need a Release Policy
 - Frequency - how often?
 - Calendar - which dates?
 - Back out plans
 - How do we undo it?
 - What is our plan B?
 - Version naming and numbering conventions
 - Roles and responsibilities

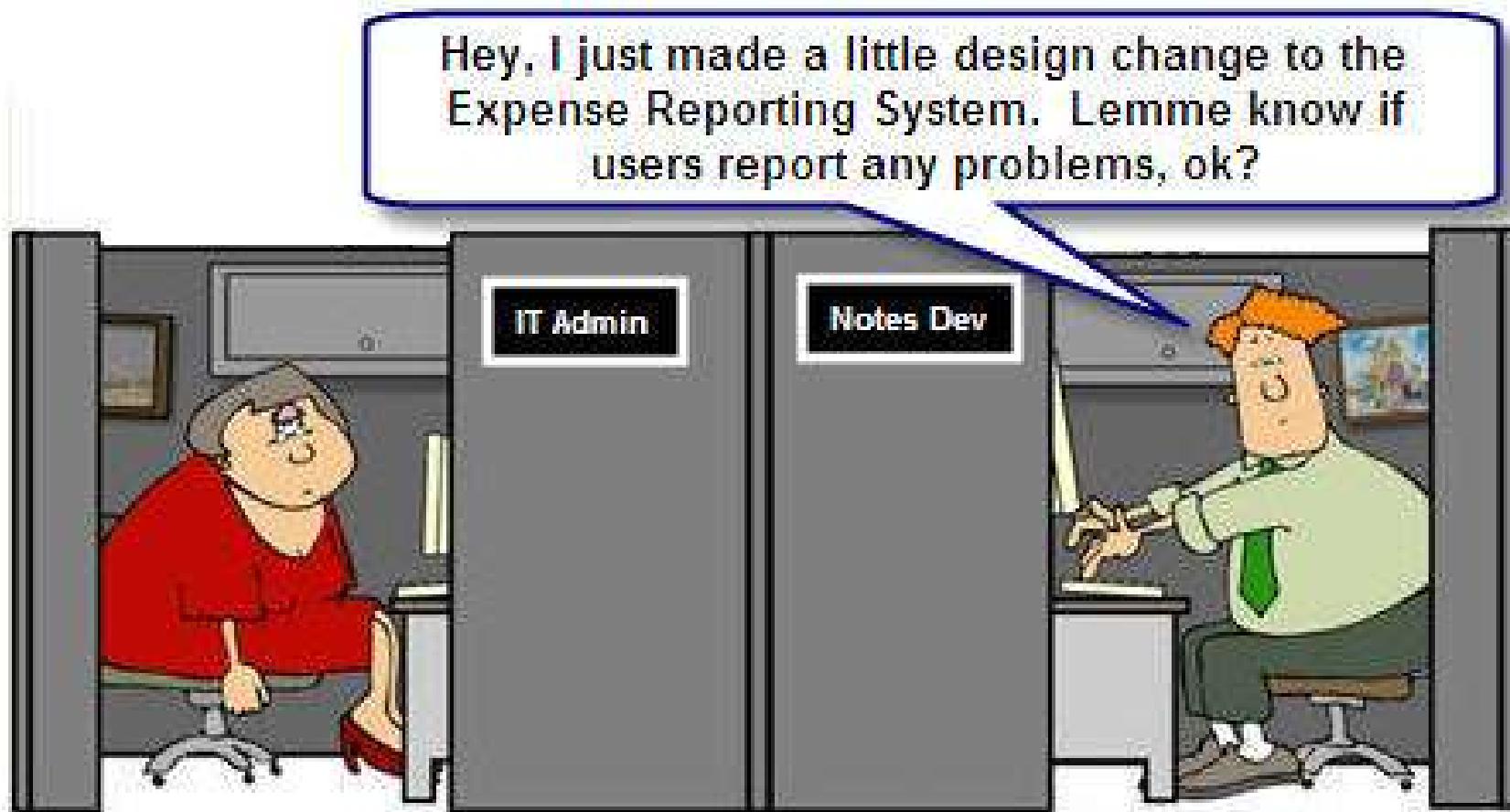
Our Approach

- Release Manager (a.k.a. RM) is a gatekeeper to production
- Admins should refuse to promote to production unless they are satisfied it will not negatively impact production
- Release strategy common to Admins & Devs, but driven from the Admin perspective

We shall discuss Lotus Domino examples of...

- Problem Management review
- Functional changes including admin requests
 - Configuration and version changes
 - Hardware changes
- Development strategy
- Testing
 - FIT
 - Stress
 - Security
 - UAT
- Deployment strategy
- Monitoring
- Closing report (aka Wrap Up report)

What often happens...



Standard Approach to...

- Communication - Have a communication plan
- Training - Have a training plan
- Standard Costing method
- Tooling - Standards
- Rollout Plan
- Back Out plans
- Sign off and acceptance criteria

Requirements

- Functional Changes
- Prioritize changes (to include in a release)
 - ♦ Impact to Business
 - ♦ Time and effort vs benefit
 - ♦ Direct costs (licenses and equipment)
 - ♦ Hard deadlines (business driven)
 - ♦ Resource availability

Requirements

- Problem Management
- Prioritization framework
 - Volume of incidents
 - Severity of incidents
 - Cost to the business
 - Time to resolve

Functional Changes

- Review functional changes (with Admins)
 - Security considerations
 - Performance considerations
 - Architectural changes
 - Domino Version impact
 - Support and licensing considerations

Example – Project, five months in development, refused to be put into production as upgrade project was cancelled. Key development components required upgrade.

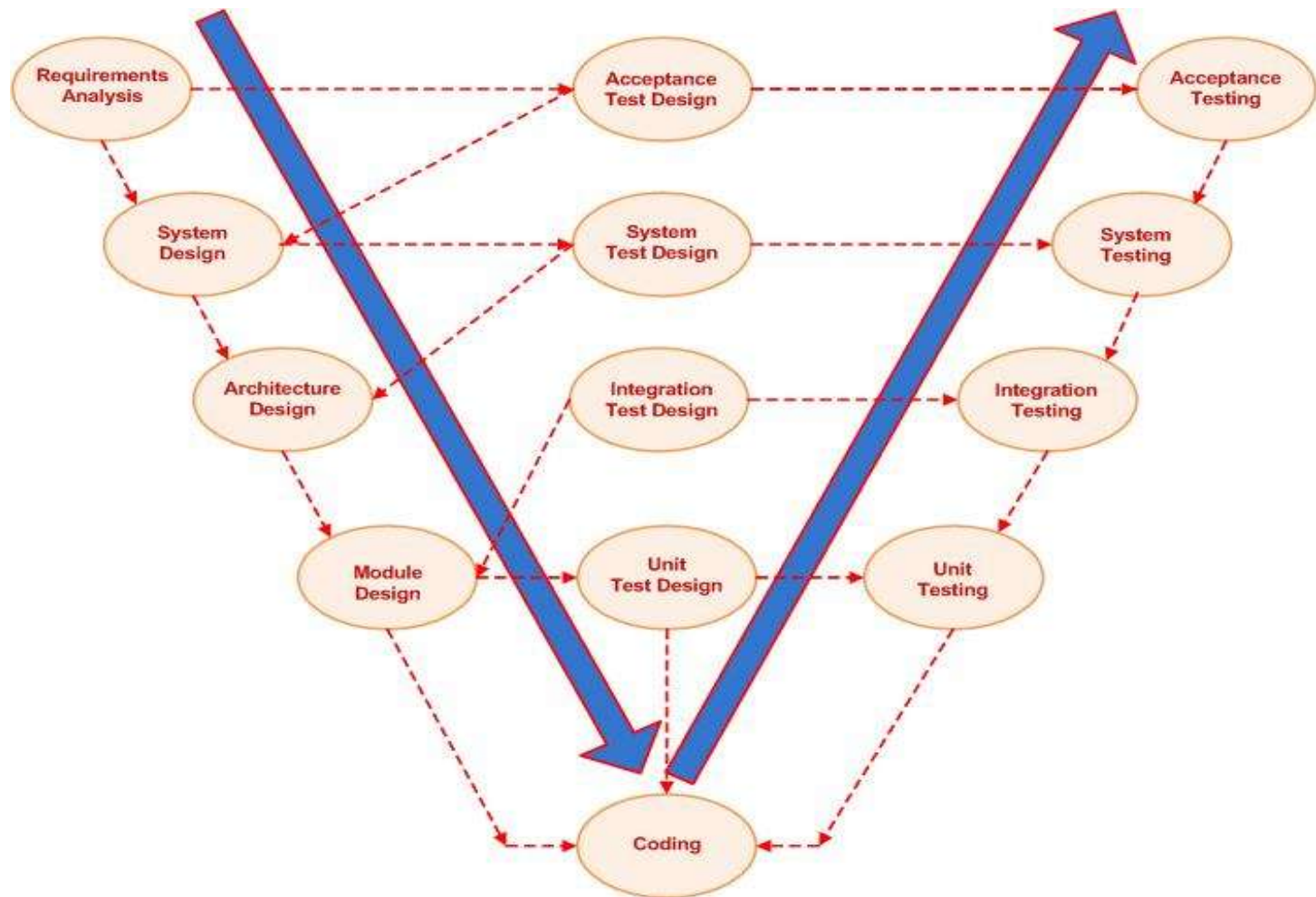
Functional Changes

- Developer should be thinking about...
 - All the items on the Admin checklist (previous slide)
...and...
 - Alert the Admin of any changes to...
 - Scheduled agents: timing, security
 - Replication (traffic, risk of conflicts)
 - ACL and Readers/Authors fields
 - Integration with other systems
 - Web enablement
 - Admin may have suggestions on functional changes to improve operations

Development

- Have a development strategy
- Update documentation as you go along
- Have gate reviews
- Have (at least) separate development, UAT and production environment
- Devs do not have access to UAT, Admins manage (practice promoting new builds to production)

Development Methodology – one example



Development

- Choose a suitable methodology for the business, follow it, and prove you followed it
- Update developer documentation as you go (details ahead...)
- For iterative methodologies (Agile, etc)...
 - Start complex / high-risk items in early builds
- Separate environments are good for everyone
 - Developers can keep coding during testing!

Testing

- Final Integration & Test (Fit for purpose)
- Stress – Get admins to perform this
- Security
- UAT
- Peer Reviews

Testing

- Unit test your code, but separate testers are essential
- Use a Test Plan!
 - Complete functionality
 - Clear statement of input vs expected output
 - Repeatable - reuse the plan each release
 - Developers don't write the Test Plan, but add to it
- Many types of testing:
 - negative - try to break it!
 - boundary cases - end-point of valid data values
 - load - some Domino apps get slower w/ more docs/users
- Testing importance = risk * impact
- Plan for multiple rounds of re-testing

Handover

- Release notes
 - Version
 - Bug fixes
 - Functional changes
- Deployment instructions
 - What files
 - Where they go
 - Pre deployment checklist
 - Disable agents, server processes, check disk space (NOT change control)
 - Post deployment checklist
 - Agents
 - Functional check
 - Production stress testing

Handover

- Release notes (above)
 - Back out plan
- Deployment instructions (above)
- List data, configuration data items
 - Keyword data, Profile Document data
- Old data conversion scripts/agents

Monitoring

- Post deployment monitoring
 - Agreed on time - week, day, etc.
 - Agents
 - Diskspace
 - Database response
 - Access and usage

Monitoring

- Post deployment monitoring
 - ♦ Agent error log
 - ♦ DDM.nsf
 - ♦ Agent profiler
 - ♦ Validate the business process is running end-to-end

Release Documentation

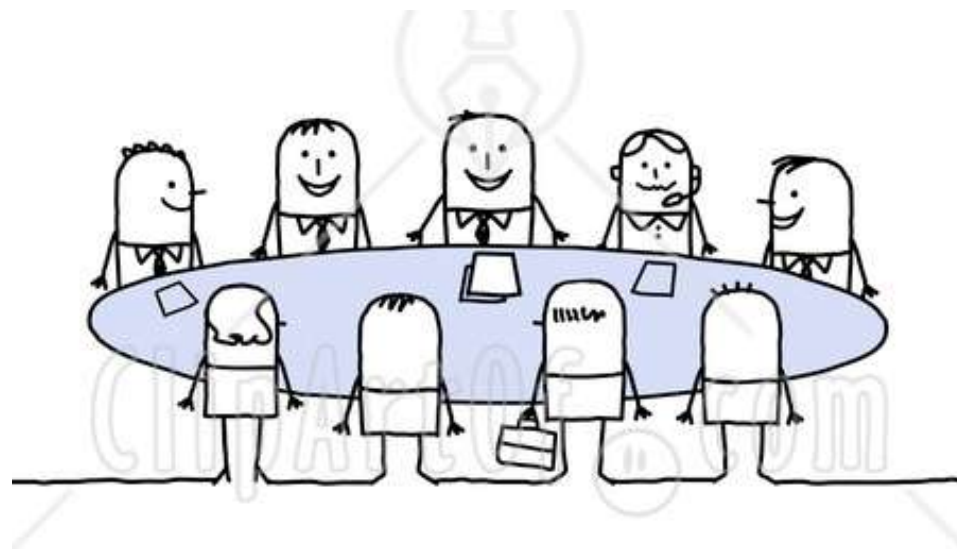
- Functional Specs
- Test Plan
 - Test results
- Release Notes
- Deployment instructions
- Release Closing (Wrap up) report

Release Documentation

- Design documentation
 - System architecture
 - Unusual / complex implementations
 - Dependencies (within Domino and with other systems)
 - ACL settings, roles
 - Scheduled agents

Release Wrap up meeting

- What to discuss
 - Incidents caused by or related to the release
 - Development approaches that worked or didn't
- Who to include
 - Customer ?
 - Developer
 - Admin
 - Customer Service
 - Release Manager



Quality Benefits

- Standard processes, consistency of approach
 - Reduces confusion that creates mistakes
 - Consistent quality comparison points
 - Learn from mistakes and do not repeat (closing report)
- Minimize/eliminate issues affecting business
 - User confusion -> support calls
 - Feature failure -> business interruption
 - Loss of data -> loss of business value
 - exposure of secure data -> liability

Cost Benefits

- Reduce negative impact to the business by failed releases, minimize downtime
- Reduce roll back costs
- Maximize business benefit re change control
- Reduce overhead
- Reduce conflict and misunderstandings
- Consistent approach (Auditors love you)
- Compliance

Summary

- Buzz terms we have not used
 - ♦ Configuration Management Database
 - ♦ Definitive Software Library
 - ♦ Definitive Hardware Store
- Release strategy is dependent on the organization & application
- Saves you time
- Reduces conflict
- Long run saves you money and reduces risk!!

Q&A

Eileen Fitzgerald | VP, GSX

Jamie Magee | Director, MartinScott Consulting

Legal Disclaimer

© IBM Corporation 2009. All Rights Reserved.

The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM's software.

References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

If the text contains performance statistics or references to benchmarks, insert the following language; otherwise delete:

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

If the text includes any customer examples, please confirm we have prior written approval from such customer and insert the following language; otherwise delete:

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Please review text for proper trademark attribution of IBM products. At first use, each product name must be the full name and include appropriate trademark symbols (e.g., IBM Lotus® Sametime® Unyle™). Subsequent references can drop "IBM" but should include the proper branding (e.g., Lotus Sametime Gateway, or WebSphere Application Server). Please refer to <http://www.ibm.com/legal/copytrade.shtml> for guidance on which trademarks require the ® or ™ symbol. Do not use abbreviations for IBM product names in your presentation. All product names must be used as adjectives rather than nouns. Please list all of the trademarks that you use in your presentation as follows; delete any not included in your presentation.

IBM, the IBM logo, Lotus, Lotus Notes, Notes, Domino, Quickr, Sametime, WebSphere, UC2, PartnerWorld and Lotusphere are trademarks of International Business Machines Corporation in the United States, other countries, or both. Unyle is a trademark of WebDialogs, Inc., in the United States, other countries, or both.

If you reference Adobe® in the text, please mark the first use and include the following; otherwise delete:

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

If you reference Java™ in the text, please mark the first use and include the following; otherwise delete:

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

If you reference Microsoft® and/or Windows® in the text, please mark the first use and include the following, as applicable; otherwise delete:

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

If you reference Intel® and/or any of the following Intel products in the text, please mark the first use and include those that you use as follows; otherwise delete:

Intel, Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

If you reference UNIX® in the text, please mark the first use and include the following; otherwise delete:

UNIX is a registered trademark of The Open Group in the United States and other countries.

If you reference Linux® in your presentation, please mark the first use and include the following; otherwise delete:

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product or service names may be trademarks or service marks of others.

If the text graphics include screenshots, no actual IBM employee names may be used (even your own); if your screenshots include fictitious company names (e.g., Renovations, Zeta Bank, Acme) please update and insert the following; otherwise delete:

All references to [insert fictitious company name] refer to a fictitious company and are used for illustration purposes only.