Announcements

R1 Next Week

Today

Review exam
Change Accountability Lecture
Project
Required External Reading

http://searchsecurity.techtarget.com/magazineContent/5-Steps-for-Developing-Strong-Change-Management-Program-Best-Practices

Don’t focus on security
Change Accountability

SWEN-343
How do you control change in your project? What difficulties does/will this have?
Where to Changes Originate?

Planned software development
Unexpected problems
   Bugs
   Security
Enhancements
Why do you need a Change Management Plan?

Inform the necessary parties (a Strong FYI)
   Knowledge sharing
   When/if systems will be down
Provide:
   Backup plan
   Trail of production changes
Catch mistakes
   More eyes = more likely to catch issues
Ensure proper protocol
   Reduce chaos
Alternatives to CMP

Free for all: People do what they want
   → Chaos

“Democracy”: Everyone votes.
   → Takes forever to get anything done

Representative Group
   A small group of educated individuals make decisions.
   Like an intelligent version of congress
Change Control Board (CCB) or CAB

A formal group of representatives who approve or reject project changes
Provide guidelines for:
- Preparing change requests
- Evaluate change requests
- Manage implementation of approved changes
Group should be stakeholders
“Typical” Process Plan

Alteration Submitter → CCB Review

More Information Needed

Change Approved

• Meet criteria
• Risk Analysis
• Testable
• Proper Rollback
• Affect Areas

Conduct Changes

Verify updates

Notification

CCB Rejected
What are some challenges this can lead to?
What are some challenges this can lead to?

“I want it now”
Emergency fixes - Emergency CCB (Account for this)
  Bug
  Customer Demand
  Security vulnerability
  Hardware issue
People don’t follow process
  What are negative ramifications?
  How do you enforce this?
People approve since it is “easier”
  Rubber stamp approvals
“The Google Way”

A pushed submission MUST
  Have attached tests
    That run & pass
  Be signed off by:
    Directory owner
    One other engineer
Stages

1. Request & Approval
2. Planning & Testing
3. Scheduling & Communication
4. Implementation
5. Documentation & Follow up
Request For Change (RFC) & Approval

Many items to review…..
  Is it needed?
  Does reward outweigh the risk?
  Is it complete?
  Should the changes be made?
  Does it leave proper audit trail

RFC can be rejected due to poor planning/documentation
  But will often push back to requestor
Planning & Testing

Significant **scrutiny** should occur here....
Requestor should provide:
  - Implementation & Backout plans
  - Easy rollback is of paramount importance
Testing should not be an afterthought
  - Verify that change took place
  - Verify that change properly took place
Much of plan is **driven by risk**
  - How much verification
  - Who should be on hand
Scheduling & Communication

When to perform changes?
   Off hours? - Will you have necessary people?

Turn into a well documented process
   Think checklists

Change management **tools** may be useful
   “eb CM”
   IBM Rational ClearCase
Implementation

Follow plan
Have proper people on call or in attendance
**Don’t be afraid** to roll-back/not conduct update
Documentation & Follow-up

Create a sound **audit** trail

Approvals
Comments
Plans

Implementation & Backout

.. Good for:

Compliance
Project history
Rollback
What Should the Document Look Like?
Example Document

http://www.se.rit.edu/~swen-343/resources/SampleCCB.docx

What changes should be made?
Activity

As a class, come up with a change process
What document will you use?
What “flow” will you use?
Who will your approvers be?
   From what role & group-groups
Will you use change management software?
What will happen to those who don’t follow the process?
Possible Process

Each group come up with their own plan & needs: 15 min
Class discussion & refinement: 15 min
Presentation & documentation: 10 min
Resources

http://www.jamasoftware.com/blog/the-change-control-board/
http://searchsecurity.techtarget.com/magazine/Content/5-Steps-for-Developing-Strong-Change-Management-Program-Best-Practices