



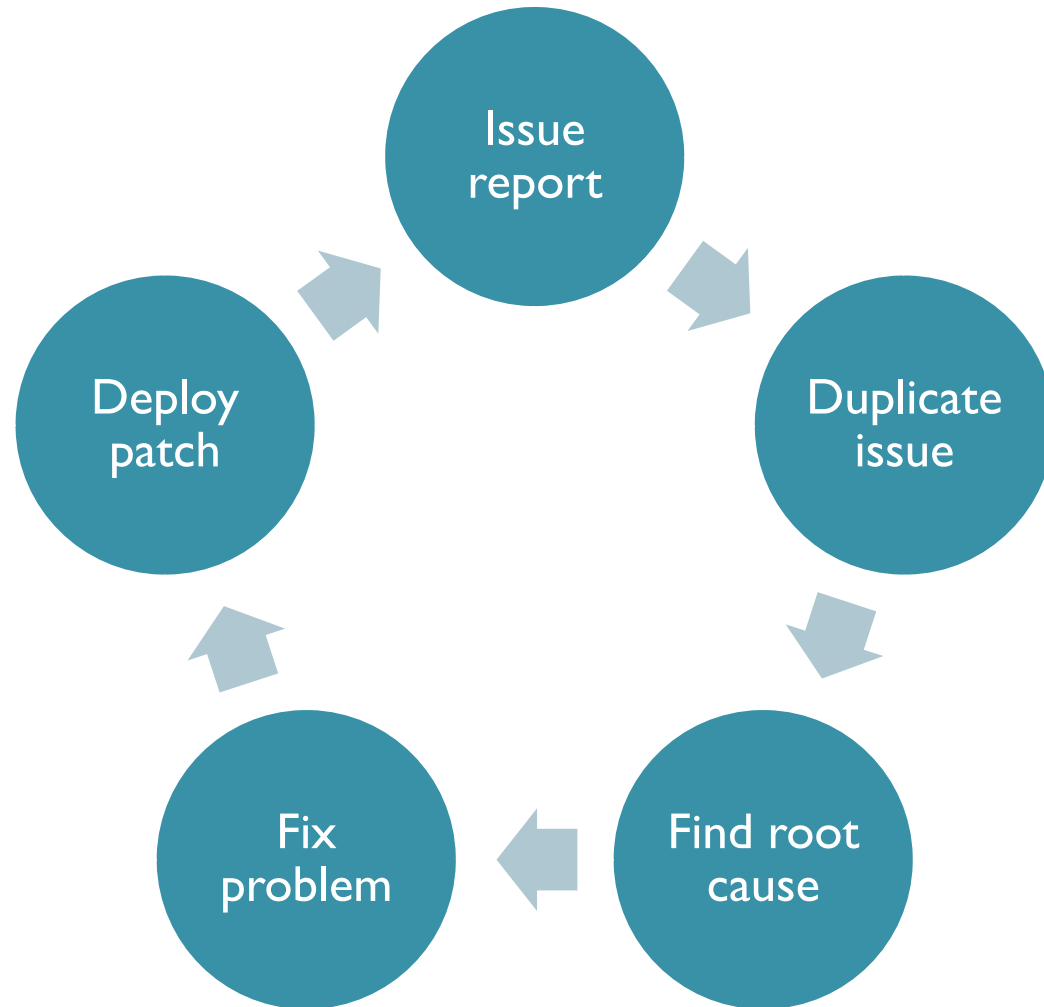
Fire Retardant for the Software Development Process

Michael G. Solomon

Agenda

- The firefighting cycle
- Why firefighting is bad
- Stopping the cycle - change
- The SCM Approach
 - Method and Tools

The Firefighting Cycle



Why Firefighting is Bad

- Fighting fires is reactive
 - The type of fire controls the response
- Preventing fires is proactive
 - Allows you to think
 - Easier to think when there is no pressure
- Fire resistant systems save effort
 - Less time spent fighting fires
 - More time designing functionality

Today's IT Demands

- Shrinking budgets (time and money)
- Shorter development cycles
- Auditing requirements
 - Governance and compliance
 - SOX
 - HIPAA
 - PCI

Fires tend to feed on one another

- Fixing one problem
 - Often causes new problems
- Effects of “fixes” may
 - Extend beyond immediate impact
 - Create new problems
 - Change interactions

You Can Prevent Fires

- Just like real fires, nearly all IT fires are preventable
- Fire prevention:
 - Behavior
 - Environment
 - Preparation
- In IT, that means
 - Planning
 - Communication
 - Maintenance

Change is necessary to prevent fires

- Paradigm change
 - from fire fighting ...
 - to fire prevention
- People don't fear change
 - they fear changing



Leaving the Comfort Zone

“I’m not afraid of dying -

I just don’t want to be there when it happens” - Woody Allen

Leaving the Comfort Zone

- Familiar is comfortable
 - Requires less work
 - Involves less risk
- Changing involves risking what is comfortable
 - Comfort is king

Why the Negative Perception?

- Past frustrations
 - “I don’t want to ever do that again”
- Lingering unresolved issues
 - “I was left holding the bag”
- Isolation
 - “I was left to do it all myself”

The Issues With Changing

- Additional responsibilities
- Unfair work distribution
- Unclear benefits
- Limited satisfaction
- Weak leadership

How to Fix the Change Process

- **Motivation**
 - Why are we doing this?
- **Communication**
 - What are we supposed to be doing now?
- **Completion**
 - How do we know we're done?

Example Communication Fail

- Extremely long password
- Password policy
 - Passwords must contain at least eight characters and one capital
- Resulting password
 - **MickeyMinniePlutoHueyLouieDeweyDonaldGoofySacramento**

Leading Through Change

- Leadership
 - More than getting people to follow orders
- Main component of culture
- Good leadership inspires action
 - “Follow me”
 - Not “Do as I say”

Leadership

- **Competence**
 - Know what you're doing
- **Vision**
 - Know where you're going
- **Connection**
 - Communicate the vision and status
- **Direction**
 - Know how you're going to fulfill the vision - there is no substitute for thinking

Challenging Chores

- Deploying new releases
- Supporting previous releases
- Managing customizations

What's Needed?

- Higher productivity
- Improved accuracy
- Monitor and manage changes

Software Configuration Management

"Software Configuration Management is the heartbeat of any serious software development effort. One would not start mass producing a car without first building a factory and defining the production line. SCM is THE factory for software development. It is the discipline, based on the principles of the manufacturing industry that brings repeatable, high-quality production to your software applications"

Software Configuration Management

- Software Configuration Management (SCM) is the discipline of managing the entire life cycle of a software project.
- Attitudes, Processes and Tools

Significant Numbers

- 62% Projects failed to meet schedules
- 49% Projects had budget overruns
- 47% Projects higher-than-expected maintenance costs

SCM Activities

- Configuration Identification
- Configuration Control
- Configuration Auditing
- Configuration Status Accounting

Configuration Identification

- Uniquely identify the component items of the application
- Define the hierarchy of the items (physical and functional)
- Identify and demarcate application configurations

Configuration Control

- Establish configuration security
- Manage application changes
- Define workflow (promotion of changes)

Configuration Auditing

- Ensure existence of required components
- Utilizes Configuration Identification
- Review configuration changes

Configuration Status Accounting

- Identify changes made to a configuration
- Identify work in process

More Numbers

- 41% Projects did not deliver ROI
- 25% Cancelled before completion
- 80% IT budgets spent on self-inflicted problems

SCM Benefits & Justifications

- Lower development costs
- Increased project visibility
- Improved communication
- Product audit capability
- Quick response to changing requirements
- Formal quality assurance cycles

SCM Tools

- Subversion

- Very popular open source version control
- Many features, but not “true” SCM
- Works in many environments
- <http://subversion.apache.org/>

- Perforce

- Commercial SCM product
- Many client plug-ins
- <http://www.perforce.com/>

SCM Tools

- Rational ClearCase
 - Commercial SCM product
 - Works with several IDEs and OSs
 - <http://www-01.ibm.com/software/awdtools/clearcase/>
- AccuRev
 - Commercial SCM with built-in best practices
 - <http://www.accurev.com/>
- Roundtable TSMS
 - Complete SCM for Progress OpenEdge
 - <http://www.tugboatsoftware.com/roundtable/>

Q&A

