

## Agenda

- § What are the challenges?
- § How does Enterprise Change Management address the challenges?
- § Implementation Case Study

ca.com

Computer Associates™

## What are the challenges?

### An Example:

We have to make a change to the Employee Surname field that exists in three of our software applications.

### The Typical Problem:

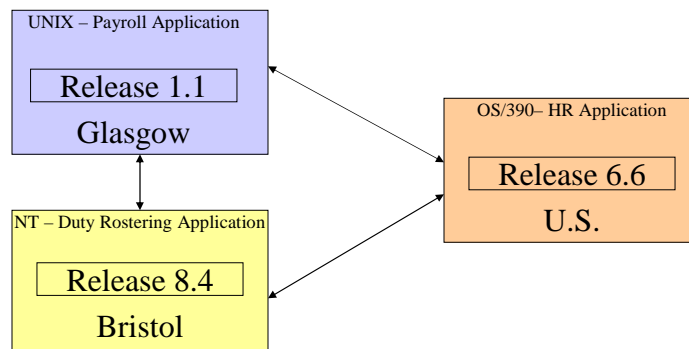
#### Applications:

- On 3 separate platforms,
- In 3 separate locations,
- Maintained by different development teams.



ca.com

## The Typical Scenario



ca.com

## Real Life is Even More Complicated!

Each Application has 3 releases being worked upon:

- Bug-fix release
- Enhancement release
- Euro conversion release

Also, each of the three projects has 5 states:

- Change request
- Development
- Test
- QA
- Production.

Try managing all of these!

ca.com

## Change Management Issues

### § Development and maintenance

- > Where is the current source?
- > Who will be affected by the change?
- > What was changed?
- > How do internal changes fit with vendor changes?
- > "Sorry, I think I overwrote your last version! "



ca.com

## Change Management Issues (continued)

### § QA and test

- > What should be tested?
- > How were executables created?
- > What caused the problem?
- > How are elements moved into production?
- > How many are moved?
- > “Why is the tested feature missing in the final release?”



ca.com

## Change Management Issues (continued)

### § Production and operations

- > Are all the correct components going to migrate together?
- > Has everything been tested successfully?
- > What caused the problem?
- > How do we recover?
- > How do we allow an emergency change?
- > “Why is the old bug in production again?”



ca.com

## Change Management Issues (continued)

### § Processes

- > “I don’t know what our process is. It’s different for every project. I am not sure on how to improve it next time!”

### § Audit Assurance

- > Is source traceable to its executable?
  - § who, when, why, where and what?
- > How much change is occurring?
- > “Who approved this release?”

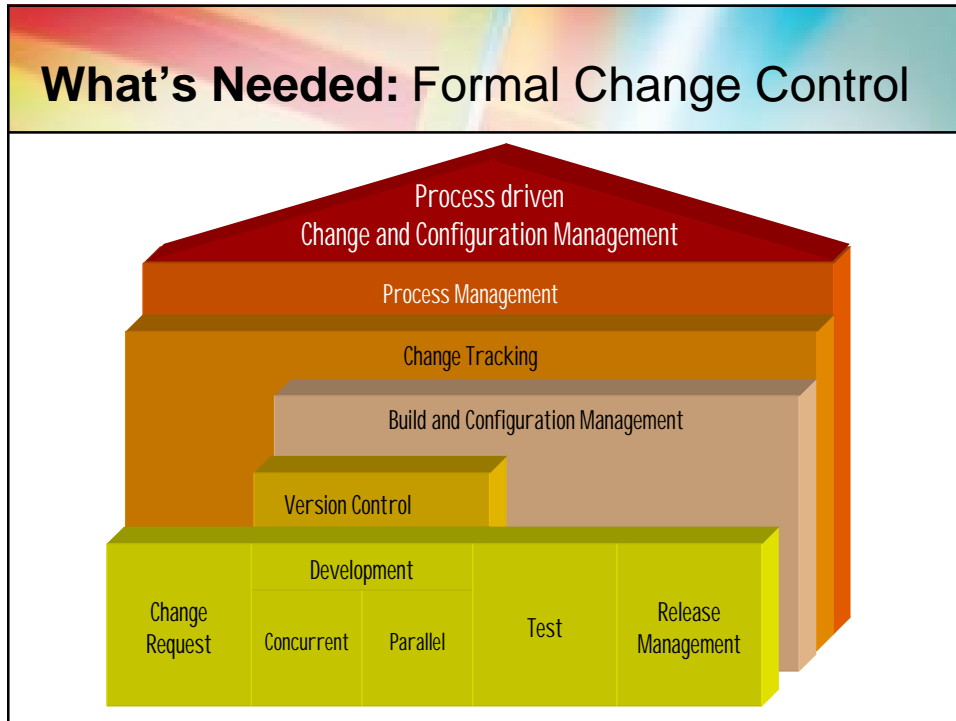


ca.com

## Cost Of Application Failure?



ca.com



### What's Needed: Formal Change Control


§ CCM's purpose is to ensure the integrity of software releases by ensuring that they contain:

- 100% of the correct changes,
- with 100% certainty that they are correct,
- and with the ability to prove it at any time.

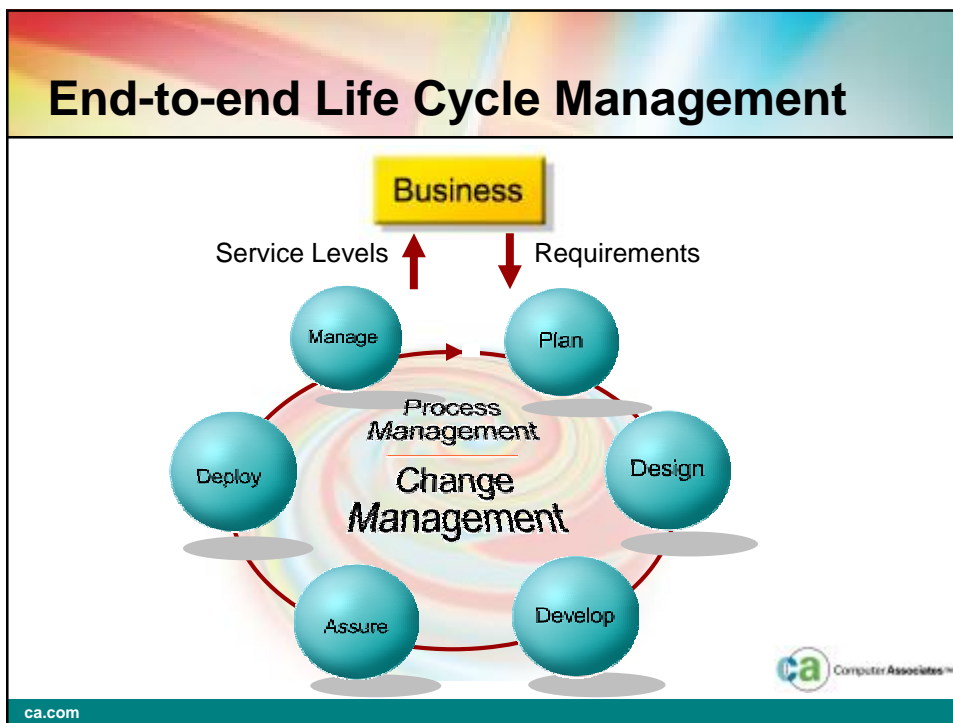
ca.com

## Enterprise Change Management Needs

Development	<ul style="list-style-type: none"> <li>§ Support for individual views</li> <li>§ Team development support</li> <li>§ Packaging capabilities</li> <li>§ Configuration management</li> </ul>
<hr/>	
Administration	<ul style="list-style-type: none"> <li>§ Promotion through the life cycle (develop, test, prep, production)</li> <li>§ Defect / change tracking</li> <li>§ Multi-platform support</li> </ul>
<hr/>	
Management	<ul style="list-style-type: none"> <li>§ Process automation / coordination</li> <li>§ Project automation</li> <li>§ Traceability / auditability</li> </ul>



ca.com



## Process- driven Application Development

Manages the development lifecycle to ensure a **standardised, repeatable** process for application development.

Computer Associates™

ca.com

## Process- driven Application Development

States (Life Cycle)	Request	Develop	Test	System Build	Production
Processes					
Roles & Activities	Create CR Approve CR Promote	C/O Browse C/O Update C/I Promote List Version UDP (build)	C/O Browse Promote List Version Approve	C/O Browse Promote List Version Approve	C/O Browse Approve
Packages	CRS, SRC, DOC				
Forms	Change Tickets Problem Reporting Defect Tracking Service Requests and Mgmt Reports				

Computer Associates™

ca.com

## Change Tracking

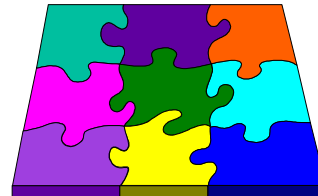
- § Defect tracking
- § Problem tracking
- § Captures all information about change activity
  - > who, when, why, where and what
- § Catches and resolves software problems efficiently



ca.com

## Configuration Management

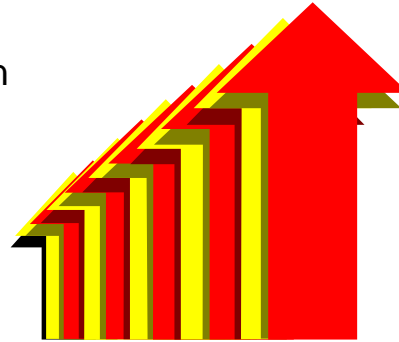
- § Tracks all components that make up an application
- § Records dependencies between components
- § Enables auditability
- § Supports impact analysis - what if...



ca.com

## Version Control

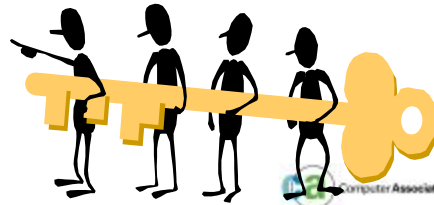
- § Maintains copies of each new version of individual components and configurations
- § Provides a way to recreate a prior version of an application



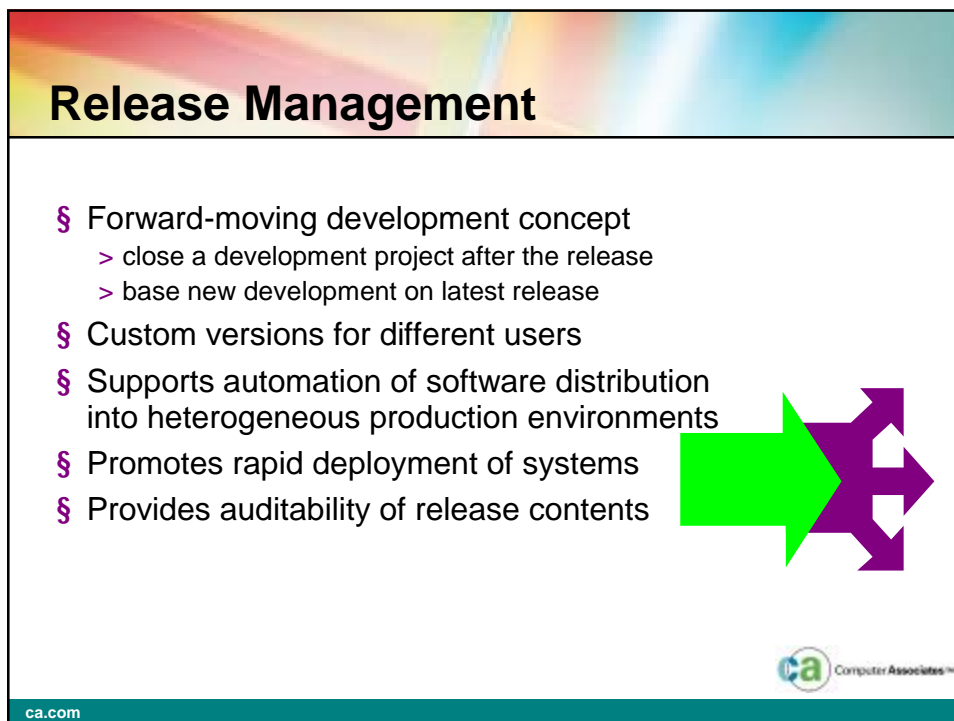
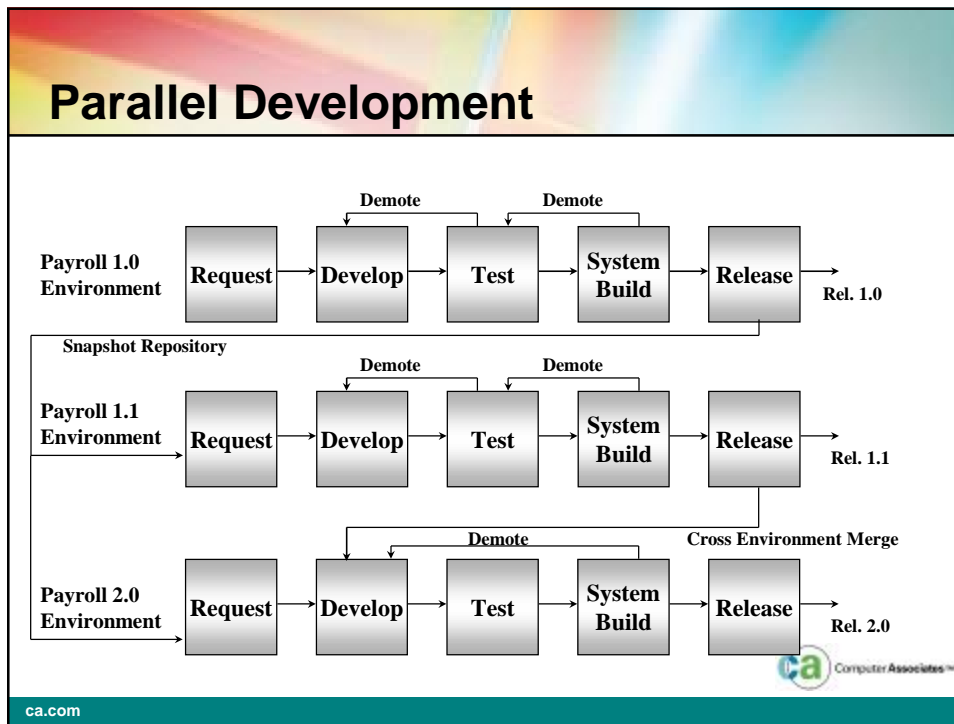
ca.com

## Concurrent Development

- § Supports multiple developers working on a single component in the same project
  - > Joe updates lines 1, 7 and 95
  - > Sharad changes lines 1, 43 and 102
- § Automates conflict detection and merging of changes
- § Eliminates regression


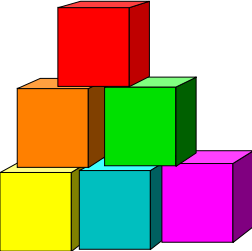


ca.com



## Build Management


- § Creates an executable program from source code
- § Integrates multiple executables into an application
- § Allows specification of rules to determine how to build an application
  - > specific types
  - > target environments
- § Ensures consistency and reliability of applications



ca.com

## Benefits of Process-Driven CCM

- § Ensures only approved changes are allowed
  - > Minimise risk of production failure
  - > Improve Quality
- § Define and control what is released to production for distribution
- § Meet Audit directive/requirements (FSA, ISO9002, CMM, ITIL etc)
- § Provides vehicle for repeatable processes - minimise risk
- § Efficient coordination between work groups



ca.com

