




Reconfiguring the Box:

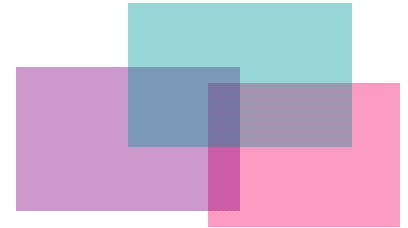


13 Key Practices for Successful
Change Management
Leesa Hicks, Tektronix

The Box: Entrenched CM Processes

- Development teams “standardized” on Base ClearCase years ago
 - Only option at the time
 - Each team with custom processes, automation
 - Typically requires 3-6 months
 - Difficult for SW Tools team to support

Base ClearCase



Customization Layer

- Branch management processes
- Automated config spec creation
- Labeling processes
- Documentation
- Training
- Track changes for a task ?!?

*Team/Project
Specific*

Base ClearCase: Branches, labels, triggers, attributes, hyperlinks

An Alternative: Unified Change Mgmt

- UCM provides automation OOB
- Adds flexibility for organizing work
 - Components, projects
- Task/activity tracking with change sets
 - Integrates with ClearQuest
- Standardized training
- Supported by SW Tools, vendor

Unified Change Management (UCM)

Customization Layer: additional policy management, automation

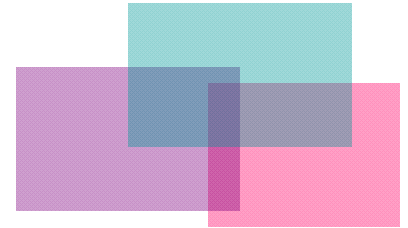
UCM Automation Layer

- Projects
- Components
- Streams (branches + meta data)
- Baselines (labels + meta data)
- Activities: tracks changes for tasks
- Remote dev support
- Built-in policy management
- ClearQuest Integration

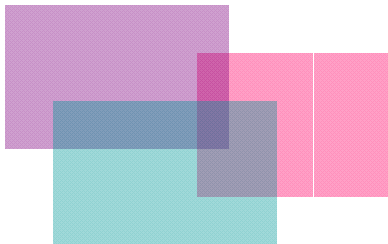
*Easily
Customized*

Base ClearCase: Branches, labels, triggers, attributes, hyperlinks

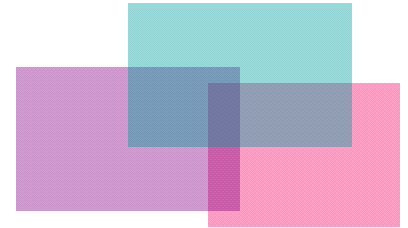
So What?



- UCM looks really useful, but why change?
- Journey of three Teams: **Red**, **Green**, **Purple**
 - What motivated them to change
 - How we went about it
 - What we accomplished



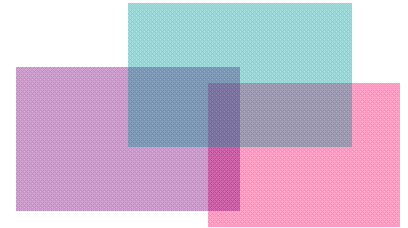
Why Change?



- Teams: **Red, Green, Purple**
 - **Some common needs, some unique**
- Performance issues with merging, labeling
 - 20+ minutes to apply a label
 - 20+ minutes to merge between branches
- Difficult to identify changes made for tasks
- Partially committed changes avail to team



Why Change (cont)?



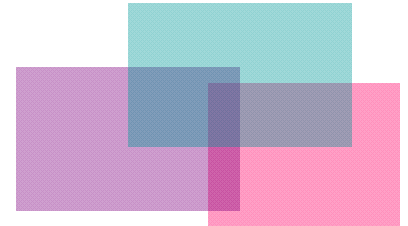
- Co-located team → distributed team
 - Using automation designed for co-located team
 - Remote team using different processes
- Difficulties integrating remote team's work
- Must work on multiple releases at a time
- Limited internal support
- Need to reduce CM support burden



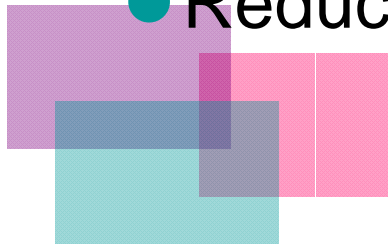
Adoption Process and Summary

	Red	Purple	Green
Relative complexity	High	Medium	Low
Education, planning, documentation	8 days	5 days	2.5 days
Source repository re-structuring	14 days	1 day	.5 days
Configure UCM	1 day	.5 days	.5 days
Deliver Training	2.5 days	2.5 days	2 days
Post-deployment support	7 days	5 days	5 days
Totals	32.5 days	14 days	10.5 days

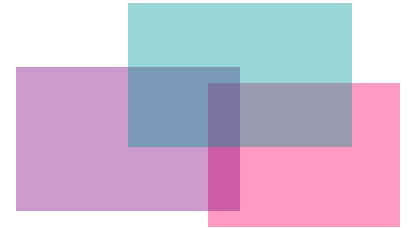
Results



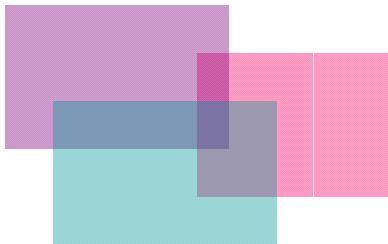
- Developing multiple releases concurrently
- Changes tracked automatically
 - Activities, change sets
- Merge performance much faster
 - Reduced 20+ minutes to seconds
- Labeling performance improved
 - Reduced build time by 20+ minutes



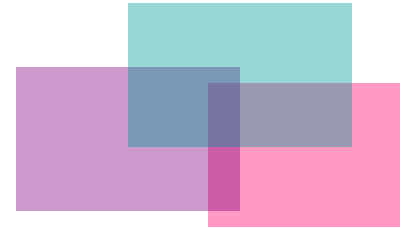
Results (cont)



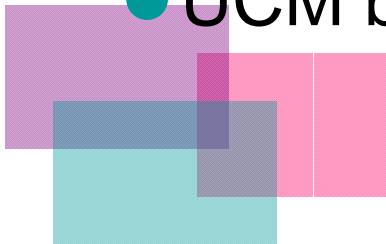
- Eliminated partially committed changes
- Reduced script maintenance
- SW Tools + vendor supported
- Have more options for sharing work
- Entire teams uses same processes



Results (cont)



- Can easily determine what is in a release, what changed between baselines
- Additional control of integration, test, and release cycles
- Training mostly standardized
- UCM skills transferrable to other teams
- UCM basics are constant





Top 13 Ways

To Go Out In a **Blaze of Glory**:
A Humorous Look at Key Practices



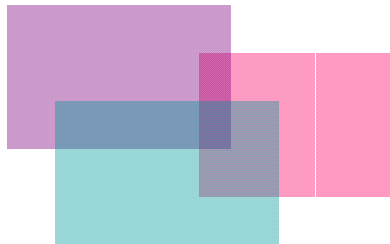
Top 13 Ways to a Blaze of Glory

- 13:** Forget the early adopters, pick the most resistant team first!
- 12:** Listen and understand?!? I know what they need . . .
- 11:** Go for the gusto – skip the prototype!
- 10:** Pilot?!? Who needs a pilot?!?



Top 13 Ways to a Blaze of Glory

- 9: Make the team work within **your** schedule, not theirs
8. The stakeholders don't need time to absorb all this new stuff!
7. Rush the team into making decisions



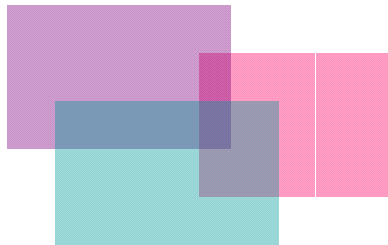
Top 13 Ways to a Blaze of Glory

6. Convince the team you have a silver bullet to fix ALL their problems
5. Get attached to the outcome, regardless of the team's perspective
4. Planning is grossly overrated!
3. Training takes too much time!!!

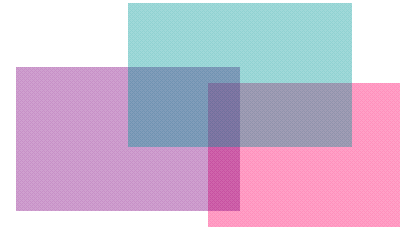


Top 13 Ways to a Blaze of Glory

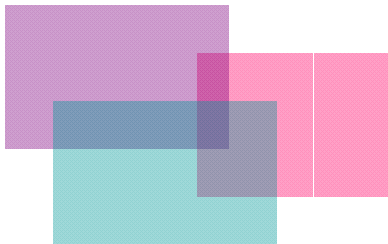
2. The day after the adoption, you're done!
1. So what if the first adoption didn't go so well . . . this is good, the next team will try harder!!!



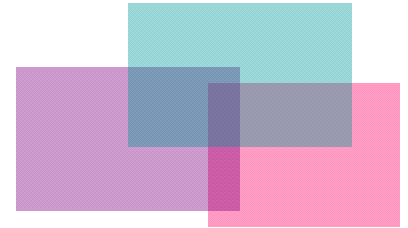
But Seriously . . .



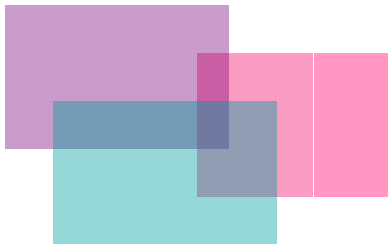
- Red team was early adopter
 - Still challenging to convince to adopt UCM
 - Successful adoption has led to others
- Have now completed four successful UCM adoptions using 13 key practices
 - Completed one more since writing this paper
 - Three more in process

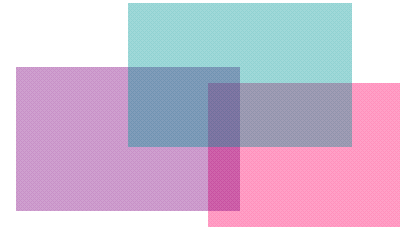


But Seriously . . .



- Reduced CM overhead for everyone
 - Reduces complexity for developers, integrators
 - Teams spend more time on development, less time on configuration management
 - Fewer support requests to SW Tools
- Solved these team's CM issues
 - By re-configuring the box!





Questions?

