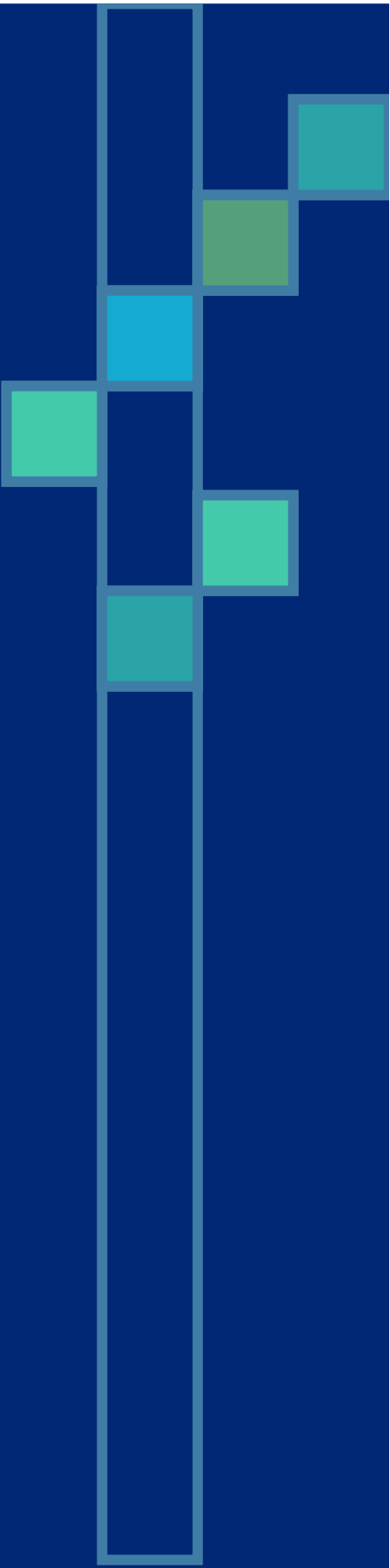



Lessons Learned from OpenDoc

Dave Curbow
Sun Microsystems
formerly Apple OpenDoc UI Design Lead






Agenda

- Fitting In
 - Keeping it Simple
 - Examples How We Failed *to keep it simple*
 - Don't Be Afraid of New Ideas
 - Size of Objects
- 




Fitting In

- Never able to replace existing apps.
 - GOOD Coexistence is critical.
 - At Least -- copy and paste between new and old.
 - Maybe -- retrofit some old apps to work better with new.
- 




Keeping it Simple -- Rule 1

- If there is already a “standard” UI mechanism to do something (e.g. print, create a *new copy, save*) don’t try to improve it.
 - UNLESS users think the old way is a bug and you’re fixing the bug.
- 



Keeping it Simple -- Rule 2

- If two UI mechanisms are similar, try to just use one mechanism. OR, make the mechanisms dramatically different.
 - Otherwise, users will get confused.
 - If a new feature is similar to a “standard”, use the standard (Rule 1)
- 



Keeping it Simple -- Rule 3

- If you don't **absolutely** have to have a feature, leave it out.
 - It's one less thing for the user to learn, and one less for you to build.
- 




Keeping it Simple -- Rule 4

- If you can't explain to a user how a feature works in 30 seconds, it's probably too complicated.
- 



Keeping it Simple -- Rule 5

- If you can't explain it to a developer in fewer than 5 pages, it's probably too complicated.
 - Linking in OpenDoc required 20+ page article to explain how to make it work. It was too complicated!
- 



Examples How We Failed

- 1. Changing File menu to Document (violation of Rule 1)
- 2. Removing the Quit command (violation of Rule 1)

Quit no longer made sense because we automatically quit apps, but users used Quit to mean “close all my windows” -- we didn't understand the user's meaning before we made this change.



Examples How We Failed


- 3. Drafts and Linking weren't critical features (violation of Rule 3)

We thought they solved real problems, but they slowed down implementation and were difficult to explain.





Don't Be Afraid of New Ideas

- Despite what I've said, if you have a good idea for making the UI better, go for it!
 - Be sure the advantage is big enough to justify breaking the Rules.
 - Don't break the Rules too many times.
- 



Size of Objects

- We found that the right size is what the user thinks is a single KIND of content.
e.g. a drawing, folder, timeline, experiment,...
- Too small means switching context too often.
- Too larger means complex menus, controls...
- Would I ever save this as a separate document?"



User Testing Will Surprise You

- No matter how expert you are, your users will (mis)understand how the system is supposed to work and they'll break it in ways you never imagined.
 - Do user tests with real users, not people who have helped design the system.
- 