User Experiences & Expectations

Managing The User Experience As An Admin

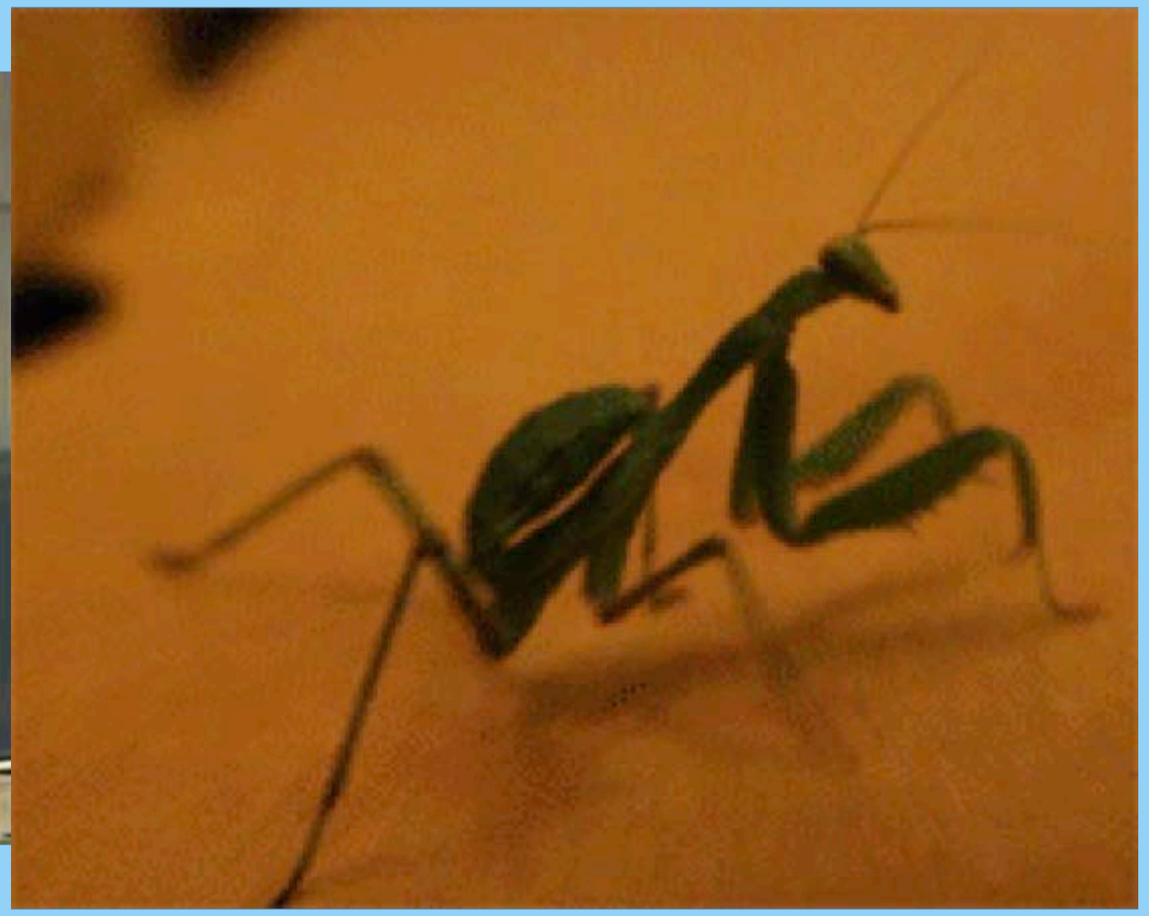
Nick McSpadden

Schools of the Sacred Heart

What do users want to do with their devices?

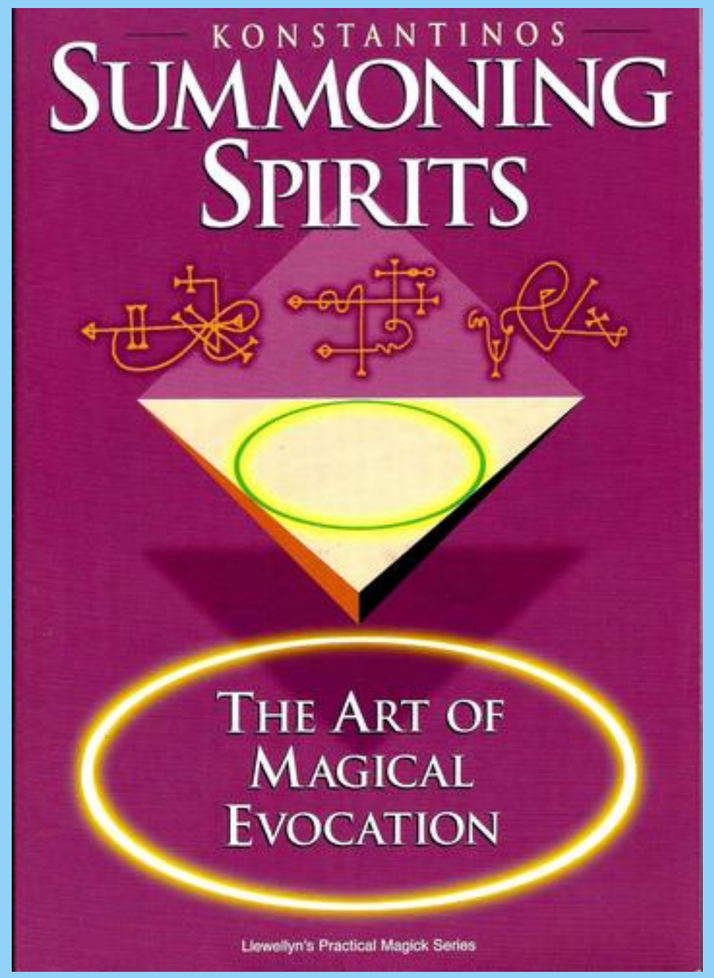
What users think they're doing:

What admins think users are doing:



What admins think admins are doing:

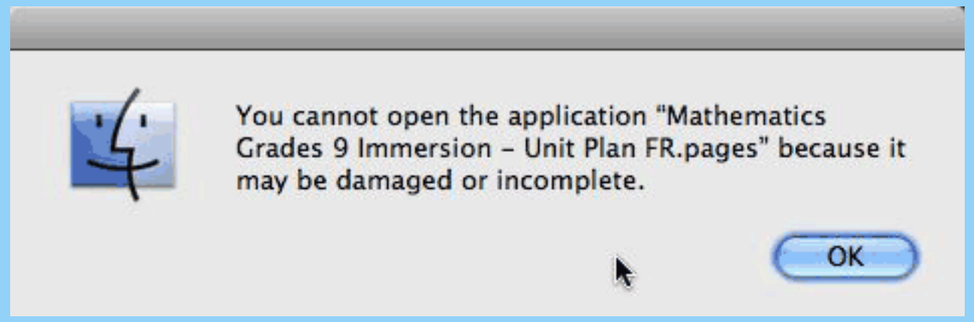
What users think admins are doing:



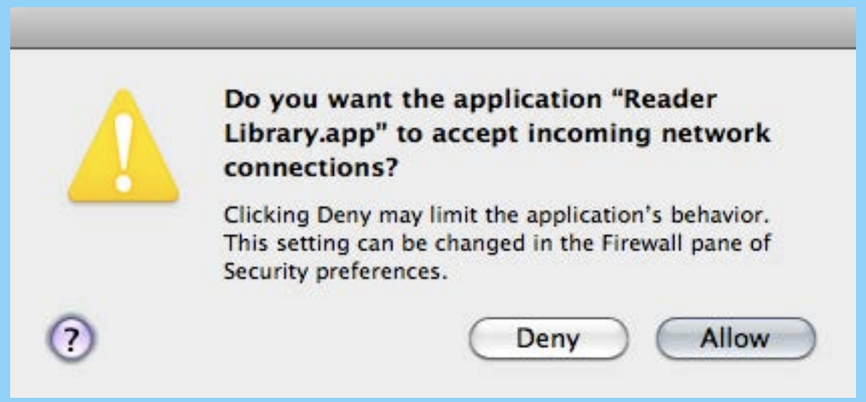
Users want things to be easy and straightforward.

How To Annoy Users:

- Present mysterious or ambiguous messages

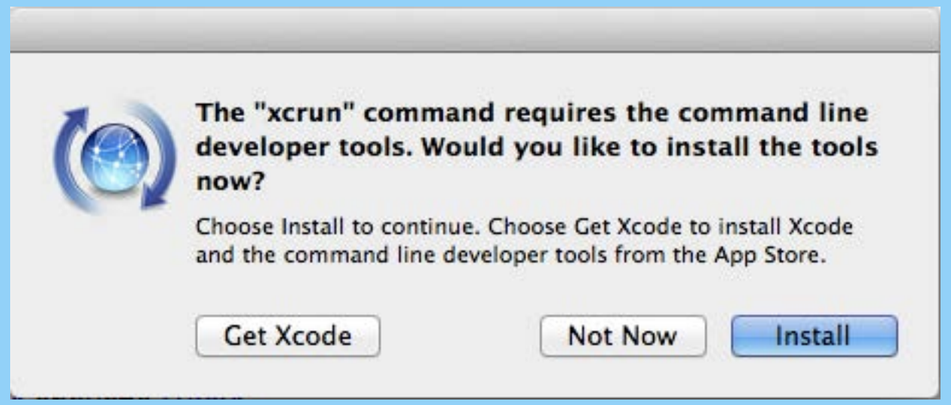


- Demand or prompt for information they don't have

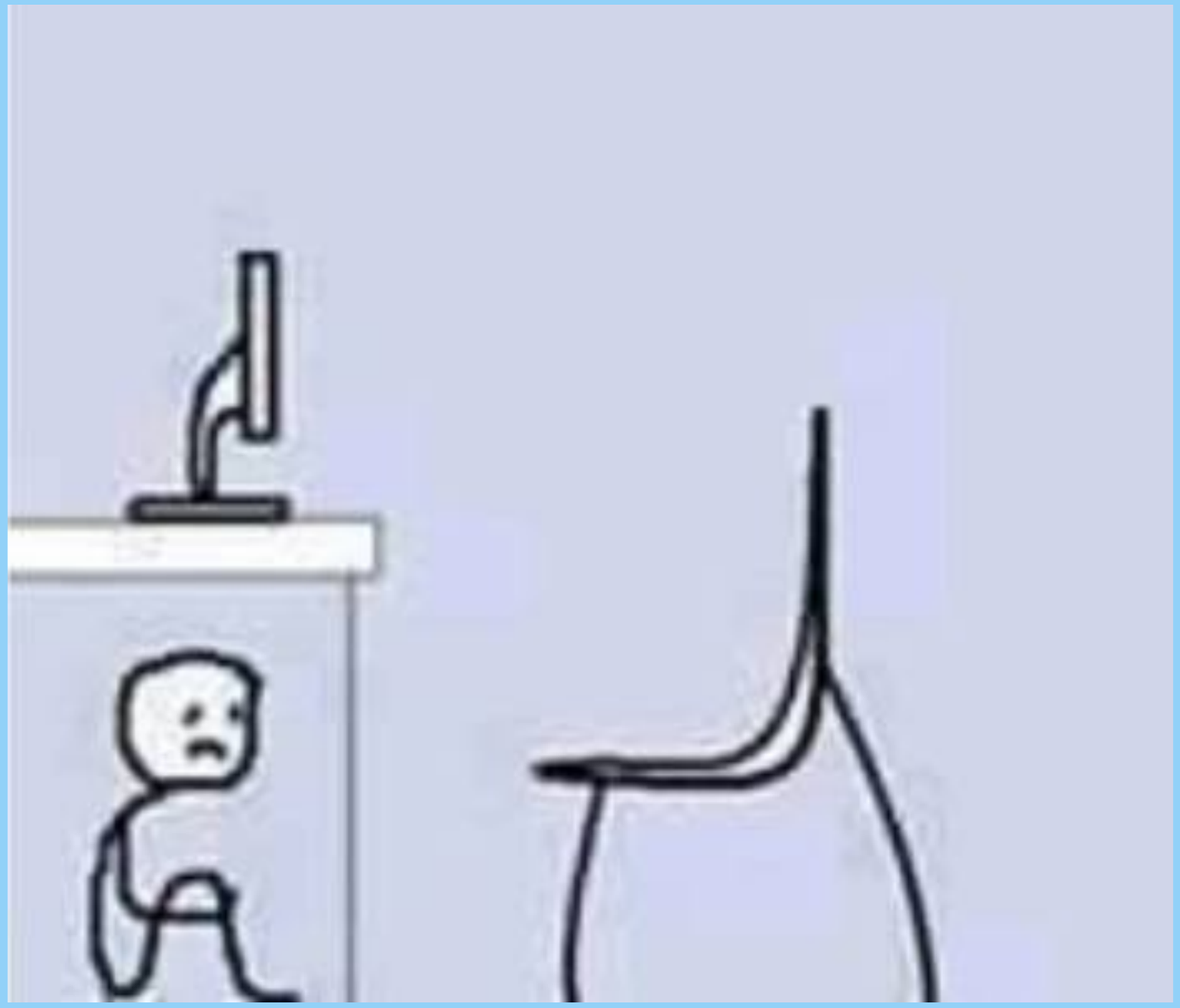


How To Annoy Users:

- Make users guess an answer they don't understand



- Deny access because of policy, without explanation



The more times a user has to ask for help, they more they'll resent asking for help.

And it's always your fault, and your department's fault.

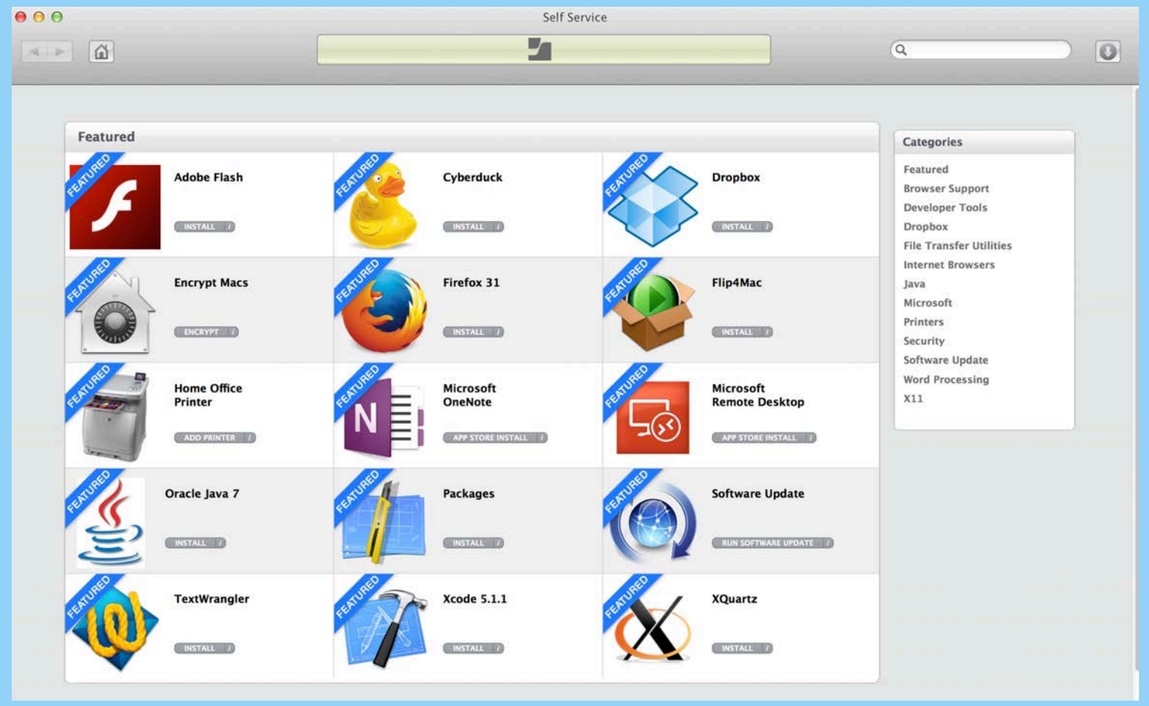
It's our job to make it all easy

- User adoption relies on it
- We need to make it easy on ourselves, too
  - We're all users of our own systems
- Policy is as much for us as it is for our users
- Security is a process

(the most overused image in presentations since clipart)

Give users the ability to do stuff on their own

- Let users pick and choose what productivity software they want



Give users the ability to do stuff on their own

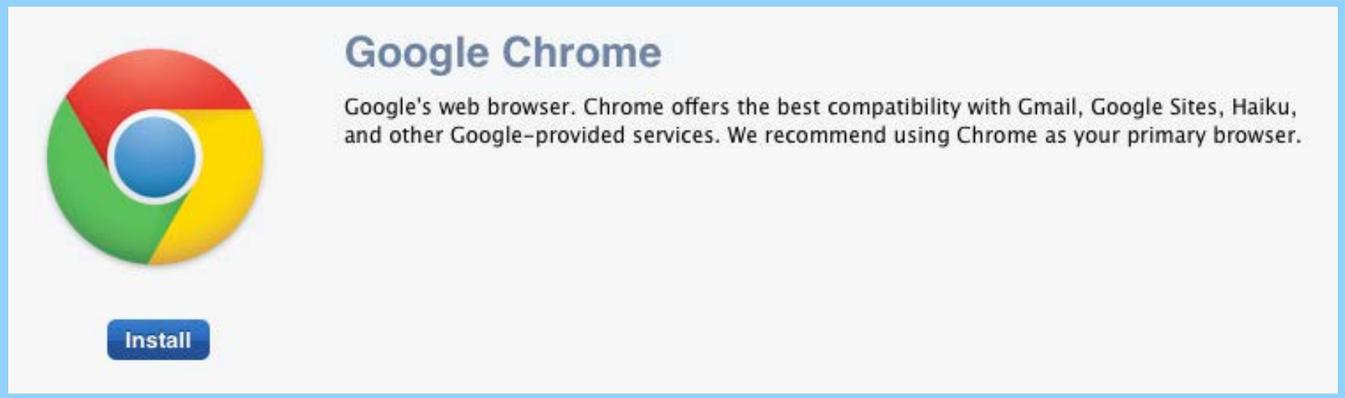
- Let them pick and choose available printers

```
#!/bin/sh
printername="business_manager".sacredsf.org
location="Business Manager"
gui_display_name=$location" 4650"
address=$printername
driver_ppd="/Library/Printers/PPDs/Contents/Resources/hp color LaserJet 4650.gz"
option_1="HPOption_Duplexer=False"
currentVersion="3.0"

if [ -e /private/etc/cups/deployment/receipts/$printername.plist ]; then
    storedVersion=`/usr/libexec/PlistBuddy -c "Print :version" /private/etc/cups/deployment/receipts/$printername.plist`
    echo "Stored version: $storedVersion"
else
    storedVersion="0"
fi

versionComparison=`echo "$storedVersion < $currentVersion" | bc -l`

/usr/bin/lpstat -p $printername
if [ $? -eq 0 ]; then
    if [ $versionComparison == 0 ]; then
        exit 1
    fi
    /usr/sbin/lpadmin -x $printername
fi

/usr/sbin/lpadmin -p "$printername" -L "$location" -D "$gui_display_name" \
        -v lpd://"${address}" -P "$driver_ppd" -o "$option_1" -o printer-is-shared=false \
        -o printer-error-policy=abort-job -E

/usr/sbin/cupsenable $(lpstat -p | grep -w "printer" | awk '{print$2}')

mkdir -p /private/etc/cups/deployment/receipts
/usr/libexec/PlistBuddy -c "Add :version string" /private/etc/cups/deployment/receipts/$printername.plist
/usr/libexec/PlistBuddy -c "Set :version $currentVersion" /private/etc/cups/deployment/receipts/$printername.plist

chown -R root:_lp /private/etc/cups/deployment
chmod -R 700 /private/etc/cups/deployment

exit 0
```

Managing Printers with Munki

Give users the ability to do stuff on their own

- Offer multiple browers, with warnings/notices about compatibility with internal sites



Prompt users to make simple decisions if you need their input.

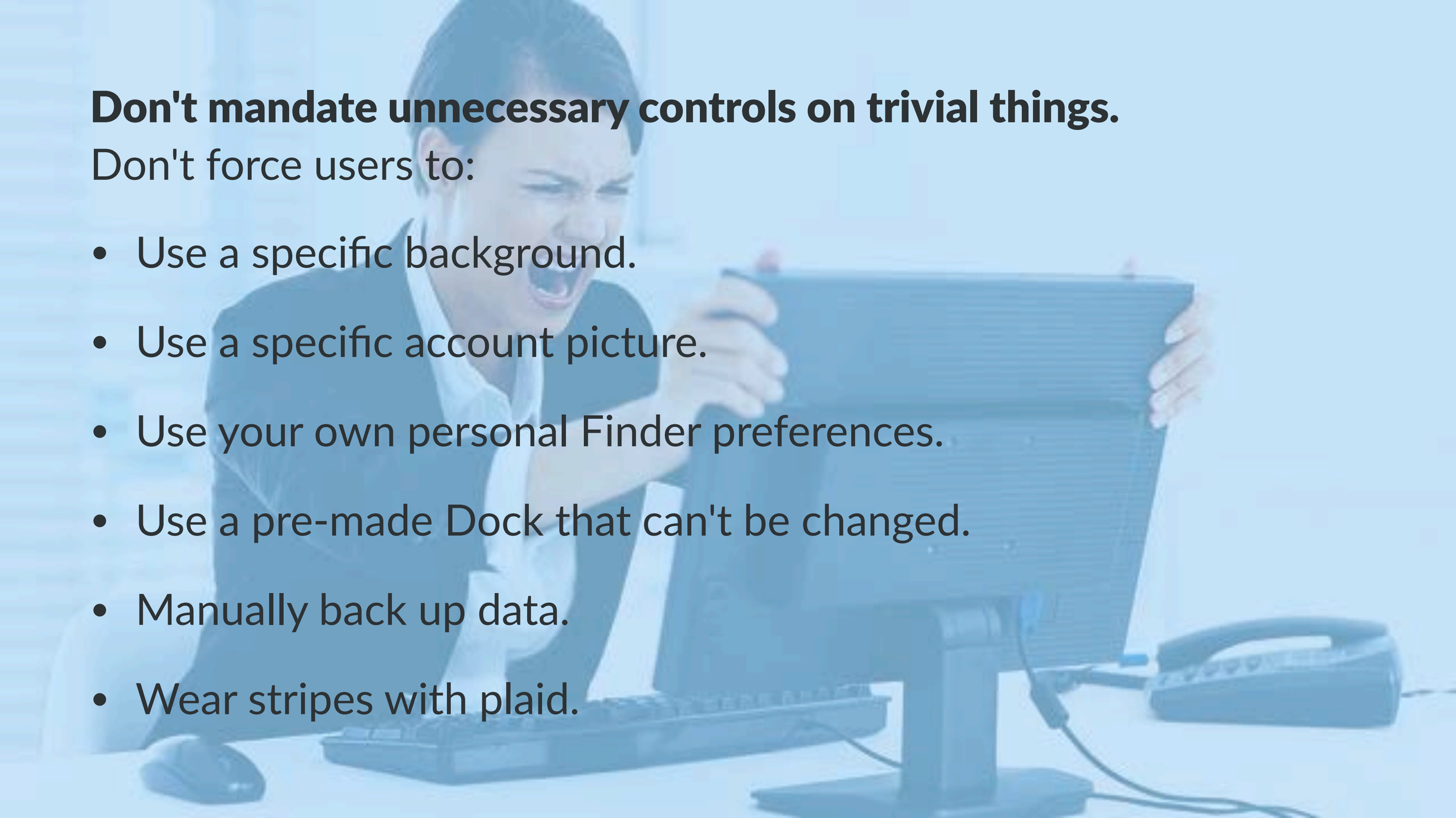
Examples:

- Mail profile requires username, full name, etc.
- Custom setup assistant that loads up network shares, or connects to internal services.

Tools to prompt users:

CocoaDialog

BigHonkingText + AppleScript



Don't mandate unnecessary controls on trivial things.

Don't force users to:

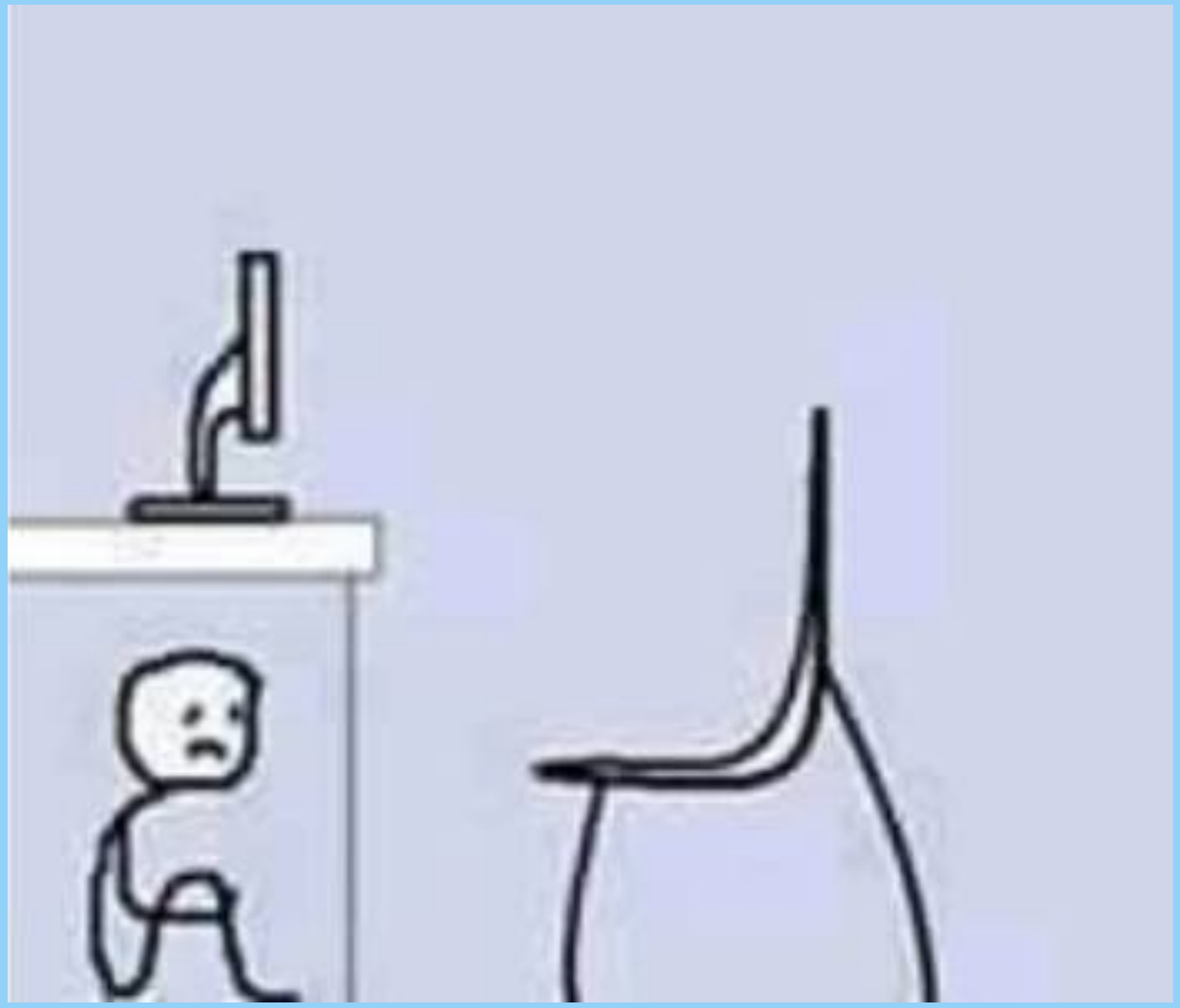
- Use a specific background.
- Use a specific account picture.
- Use your own personal Finder preferences.
- Use a pre-made Dock that can't be changed.
- Manually back up data.
- Wear stripes with plaid.

"Why can't I set my background to something other than this?"

"It's IT policy."

"That's a stupid policy. Who can I talk to about changing it?"

"You'll have to go talk to my boss."



Giving users more control isn't the only way to improve the experience.

We also need to remove unnecessary obstacles, and avoid asking the user to make decisions.

- Manage Apple software updates:
  `softwareupdate --schedule off`
- Manage third-party software updates. If possible, do them in the background, silently, and unobtrusively.
- If you can't do them silently in the background in a reasonable way, schedule updates for a consistent time slot and communicate this to users.
- Make heavy use of profiles to manage settings for the OS, applications, and settings.

What admins see when Software Update appears on a user's screen:

What users see when Software Update appears:

Remove first run dialogues from Apple apps:

- Keynote:

▼ PayloadContent	Dictionary	(1 item)
▼ com.apple.iWork.Keynote	Dictionary	(1 item)
▼ Set-Once	Array	(1 item)
▼ Item 0	Dictionary	(1 item)
▼ mcx_preference_settings	Dictionary	(2 items)
TMAFirstLaunchVersion	Number	65541
TMAFirstLaunchCoachingTipShownKey	Number	1

- Pages and Numbers:

▼ PayloadContent	Dictionary	(1 item)
▼ com.apple.iWork.Pages	Dictionary	(1 item)
▼ Set-Once	Array	(1 item)
▼ Item 0	Dictionary	(1 item)
▼ mcx_preference_settings	Dictionary	(1 item)
TMAFirstLaunchVersion	Number	65541

Remove first run dialogues from Apple apps:

- iMovie:

▼ PayloadContent	Dictionary	(1 item)
▼ com.apple.iMovieApp	Dictionary	(1 item)
▼ Set-Once	Array	(1 item)
▼ Item 0	Dictionary	(1 item)
▼ mcx_preference_settings	Dictionary	(1 item)
showWelcomeScreenAtStartup	String	0

- iPhoto:

▼ PayloadContent	Dictionary	(1 item)
▼ com.apple.iPhoto	Dictionary	(1 item)
▼ Set-Once	Array	(1 item)
▼ Item 0	Dictionary	(1 item)
▼ mcx_preference_settings	Dictionary	(3 items)
FirstLaunch	String	0
GeoCodeLookupPreference	Number	2
TMAFirstLaunchVersion	Number	1

Remove first run dialogues from Apple apps:

- GarageBand:

▼ PayloadContent	Dictionary	(1 item)
▼ com.apple.garageband10	Dictionary	(1 item)
▼ Set-Once	Array	(1 item)
▼ Item 0	Dictionary	(1 item)
▼ mcx_preference_settings	Dictionary	(1 item)
welcomeScreenShown	String	1

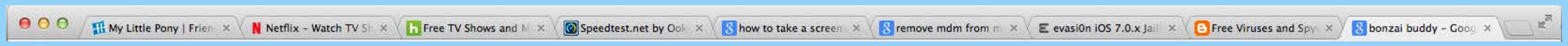
Remove the new account setup assistant:

▼ PayloadContent	Dictionary	(1 item)
▼ com.apple.SetupAssistant	Dictionary	(1 item)
▼ Set-Once	Array	(1 item)
▼ Item 0	Dictionary	(1 item)
▼ mcx_preference_settings	Dictionary	(4 items)
DidSeeCloudSetup	Boolean	☑
LastSeenCloudProductVersion	String	10.10
RunNonInteractive	Boolean	☑
LastSeenBuddyBuildVersion	String	14A389

- Remove menu extras you don't expect to use:

▼ PayloadContent	Array	(1 item)
▼ Item 0	Dictionary	(6 items)
▼ PayloadContent	Dictionary	(1 item)
▼ com.apple.mcxMenuExtras	Dictionary	(2 items)
▼ Forced	Array	(1 item)
▼ Item 0	Dictionary	(1 item)
▼ mcx_preference_settings	Dictionary	(4 items)
Bluetooth.menu	Boolean	☐
Displays.menu	Boolean	☐
TimeMachine.menu	Boolean	☐
delaySeconds	Number	1

Depending on their job, users tend to spend a great deal of time in a web browser window.



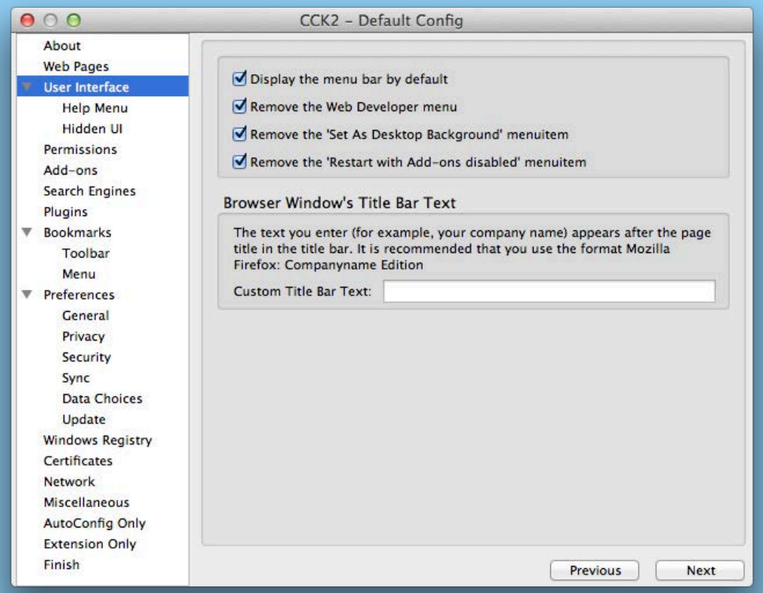
Removing barriers to opening up a web browser and getting access to content will go a long way towards improving the user experience.

Common browser improvements:

- Remove all first run dialogs (Firefox & Chrome, I'm looking at you)
- Turn off automatic updates, manage them yourself
- Test updates / versions for compatibility with critical intranet sites
- Don't manage bookmarks. Instead, provide a single site users can go to find content they need.
- If you want users to gravitate towards a specific browser, consider making it the default.

Firefox:

Use Mike Kaply's Client Customization Kit 2 to remove all the annoying Firefoxisms.



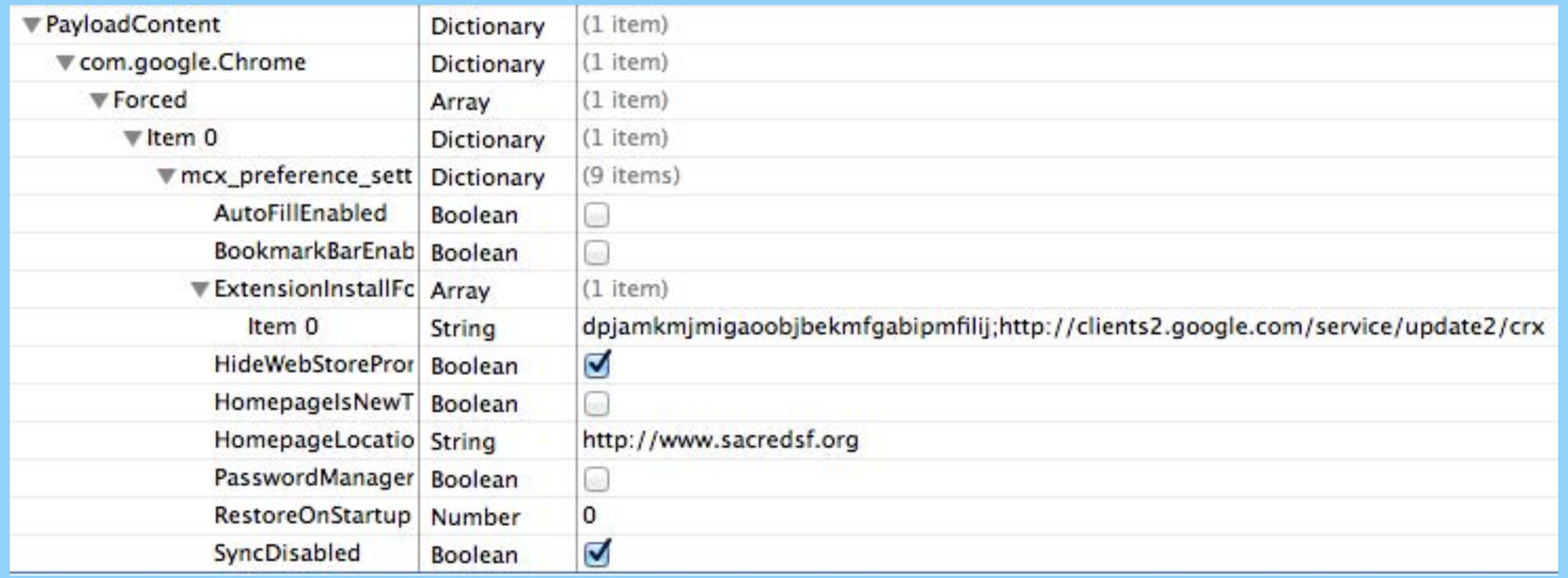
Safari:

Simple to manage with a profile.
Avoid over-managing, as users rely on this to sync with iOS devices.

▼ PayloadContent	Dictionary	(1 item)
▼ com.apple.Safari	Dictionary	(1 item)
▼ Forced	Array	(1 item)
▼ Item 0	Dictionary	(1 item)
▼ mcx_preference_settings	Dictionary	(11 items)
ShowFavoritesBar	Boolean	☐
AutoFillCreditCardData	Boolean	☐
HomePage	String	http://www.sacredsf.org/
NewWindowBehavior	Number	0
NewTabBehavior	Number	1
AutoFillPasswords	Boolean	☐
AutoFillMiscellaneousForms	Boolean	☐
AutoFillFromAddressBook	Boolean	☐
DownloadsPath	String	~/Downloads
DownloadsClearingPolicy	Number	1
AlwaysShowTabBar	Boolean	☑

Chrome:

There's a theme here:



▼ PayloadContent	Dictionary	(1 item)
▼ com.google.Chrome	Dictionary	(1 item)
▼ Forced	Array	(1 item)
▼ Item 0	Dictionary	(1 item)
▼ mcx_preference_sett	Dictionary	(9 items)
AutoFillEnabled	Boolean	☐
BookmarkBarEnab	Boolean	☐
▼ ExtensionInstallFc	Array	(1 item)
Item 0	String	dpjamkmjmigaoobjbekmfgabipmfilij;http://clients2.google.com/service/update2/crx
HideWebStorePror	Boolean	☑
HomepageIsNewT	Boolean	☐
HomepageLocatio	String	http://www.sacredsf.org
PasswordManager	Boolean	☐
RestoreOnStartup	Number	0
SyncDisabled	Boolean	☑

Sadly, there's more work involved in removing the first run dialog.

How do I figure out what settings to manage for other software?

- Use a VM with snapshots to be able to easily test the same procedure over and over.
- Use a tool like fseventer, opensnoop, or fs_usage to figure out what files are created / modified when the program launches the first time.
- If it's a plist, use `defaults read` to see what keys exist. Use the `defaults` command to try making modifications to see what happens to the software.

How do I deploy these settings?

- If it's a plist, use Tim Sutton's Make-Profile-Pkg to create a profile that can be easily deployed in Apple pkg format.
- If it's not a plist, collect the preference files and create your own deployable package out of them.
- Alternatively, use Profile Manager on OS X Server to upload a plist file to create a profile.

To summarize:

We admins have a lot of power over the user experience.

Use it wisely!