

towards a theory of software design

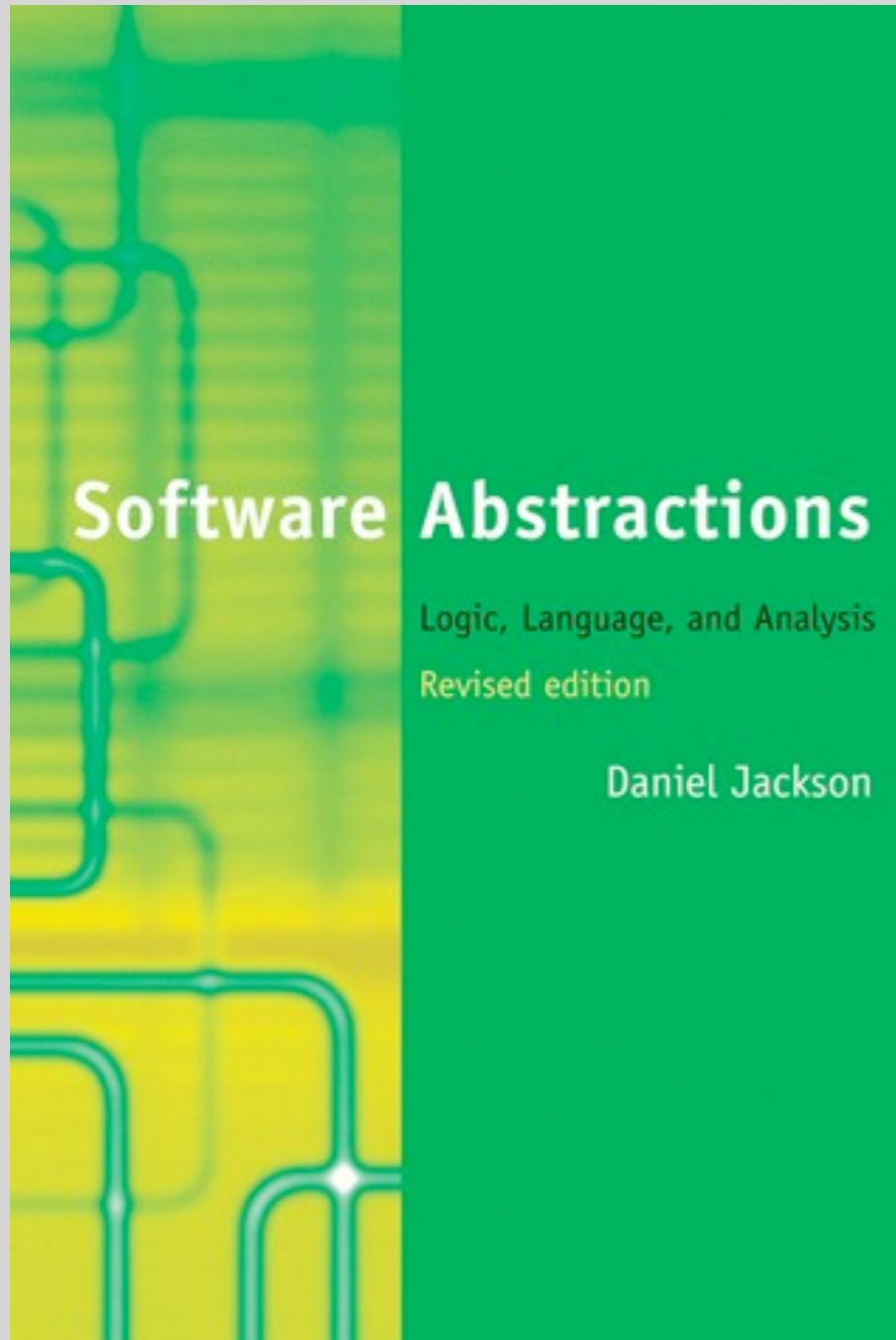
Daniel Jackson · MIT CSAIL

UIUC · November 7, 2016

9, 2012 6:58 AM	1.6 MB	Portab...(PDF)
2012 6:58 AM	2.7 MB	Application
2012 4:47 PM	4.9 MB	MP3 audio
2012 3:40 PM	4.4 MB	JPEG image
, 2011 3:47 PM	20.3 MB	TIFF image
1998 8:47 PM		

my
mid-life
crisis

rethinking my assumptions



a traditional view
quality = code correctness

my career so far
lightweight formal models
design \gg code
exploration \gg certification

but now I'm wondering
do details matter?
are these the wrong details?
are we looking under the lamppost?

dropbox
woes

a sad dropbox tale

Quora

🔍 Search



Dropbox: [Edit](#)

Someone accidentally deleted thousands of files in my company Dropbox: how can I quickly undelete them? [Edit](#)

[Add Question Details](#)

[Comment](#) · [Share](#) · [Report](#) · [Options](#)

Friends don't let friends delete shared Dropbox items



Christopher Breen
@BodyofBreen

Sep 9, 2013 5:00 AM



Reader Paul Cramblett has a problem with others who just don't know how to share. He writes:

I maintain a Dropbox folder that I use to share files with a select group of friends. I've tried to explain how Dropbox works to these people but someone invariably drags all the files out of the folder, which means they're no longer available to the rest of us. Is there some way to prevent files from being removed by someone who doesn't understand the difference between "copy" and "move"?

alyssa & ben plan a party

 Get free space!

Dropbox



Search

Share 'party' with others



Allow members to invite others

alyssa

[Import contacts](#)



Alyssa P. Hacker
dnj+alice@csail.mit.edu

Share folder

Cancel

alyssa spoils everything

Deleted 1 item. [Undo](#)

Dropbox

Delete folder?

Are you sure you want to delete **party** from your Dropbox?

Delete

survey of MIT dropbox users

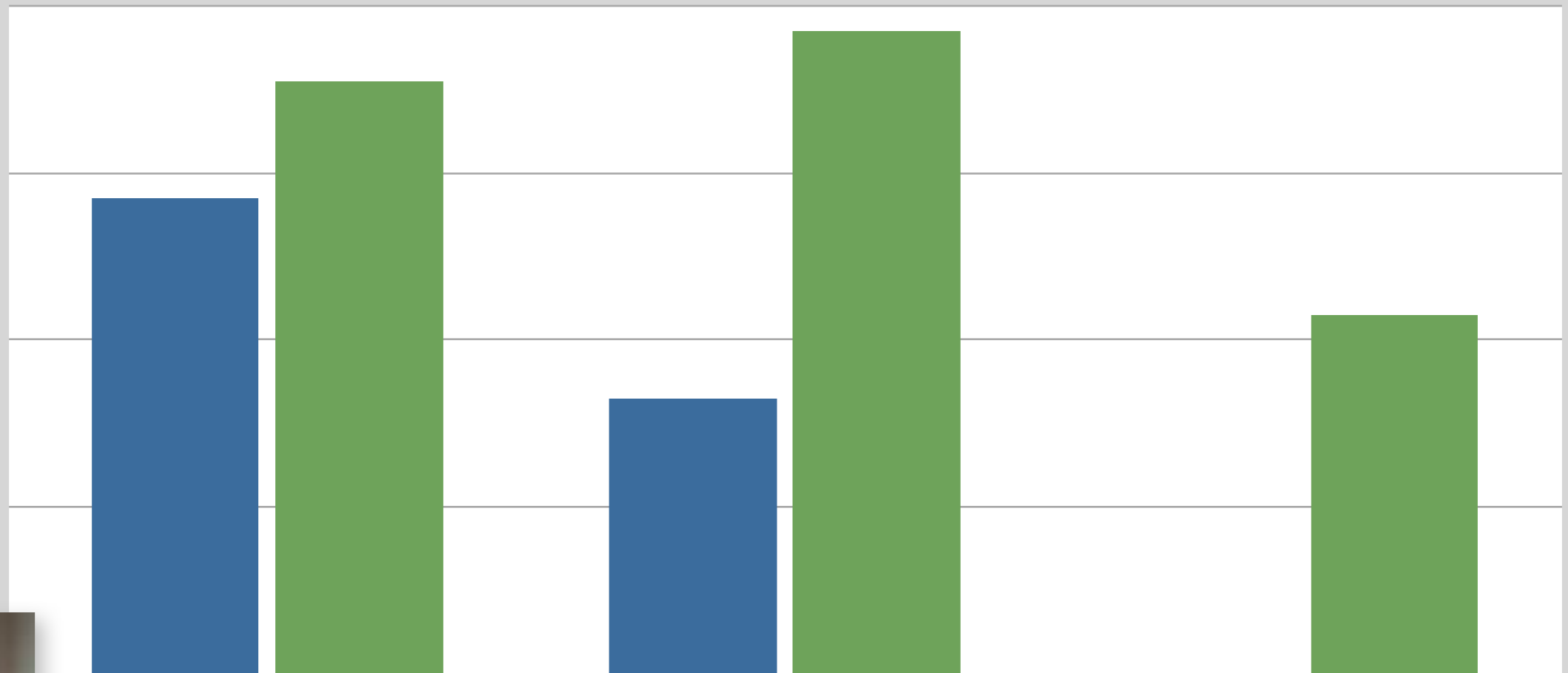
correctly predicting behavior

80.0%

60.0%

40.0%

20.0%



good knowledge

average knowledge

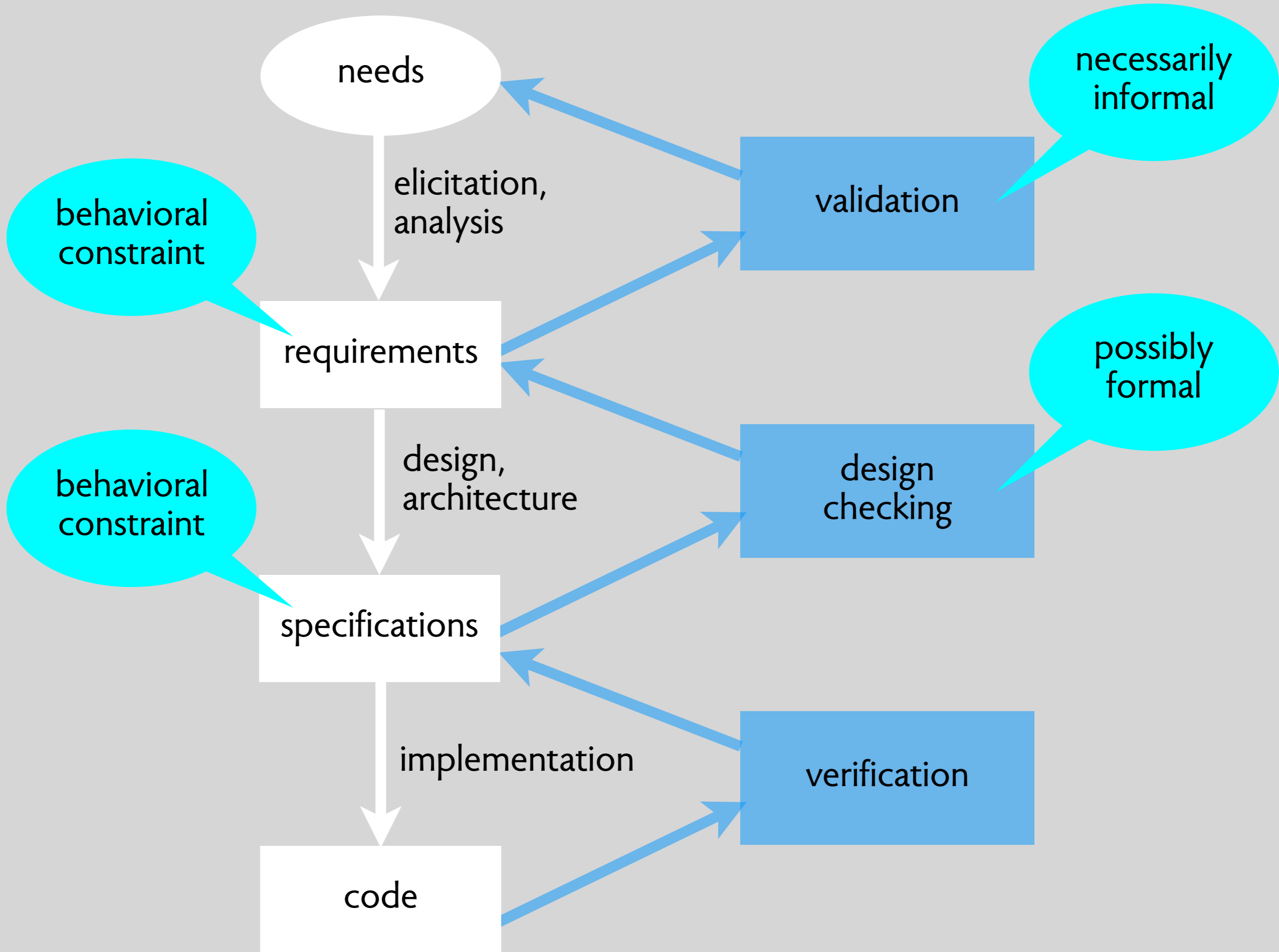
poor knowledge

- delete shared folder results in leaving
- delete shared subfolder removes it



Kelly Zhang

the
software
problem



example: museum ticketing



needs

forgot staff, press passes, family tickets

requirements

"everyone inside holds a ticket"
"every ticket has been paid for"

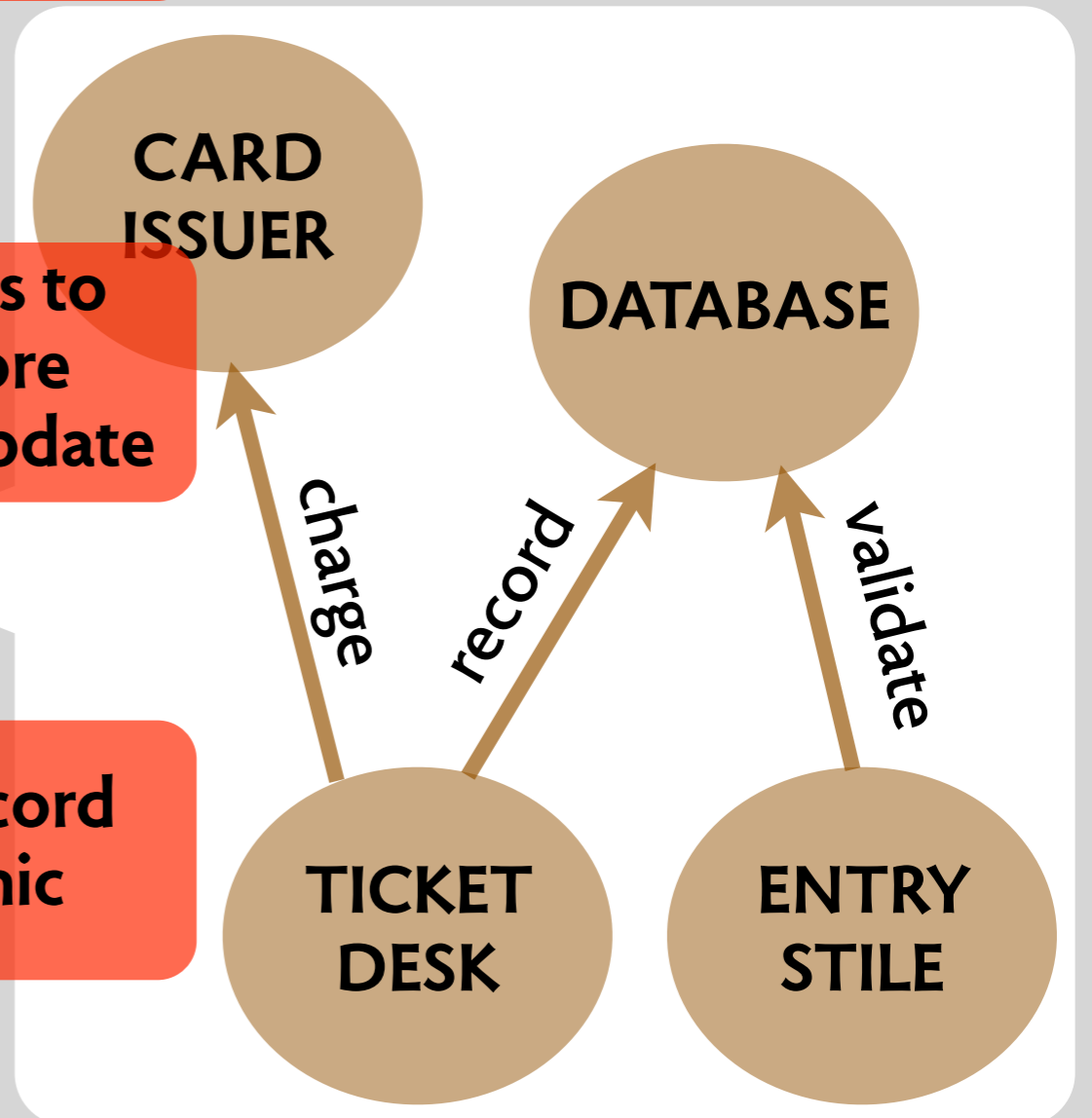
specifications

buy: charge, record, issue ticket
enter: if ticket is valid

code

charge/record not atomic

visitor gets to stile before database update



needs

must get these right

forgot staff, press passes, family tickets

but these matter more

this is a **design!**

requirements

"everyone inside holds a ticket"
"every ticket has been paid for"

forgery: people enter with fake tickets

specifications

buy: charge, record, issue ticket
enter: if ticket is valid

visitor gets to stifle before database update

scalping: no tickets available, prices rocket

code

charge/record not atomic

database lost: all issued tickets now invalid

the
software
problem,
revisited

needs

purposes

concept
purposes

concepts

code

Balance budget
Grow attendance
Customer satisfaction

get visitors to
pay

limit
crowds

Ticket

Button

Queue

needs

purposes

concept purposes

concepts

code

not a **design!**

this is a **design!**

these are **misfits** of the ticket concept

Balance budget
Grow attendance
Customer satisfaction

get visitors to pay

limit crowds

Ticket

Button

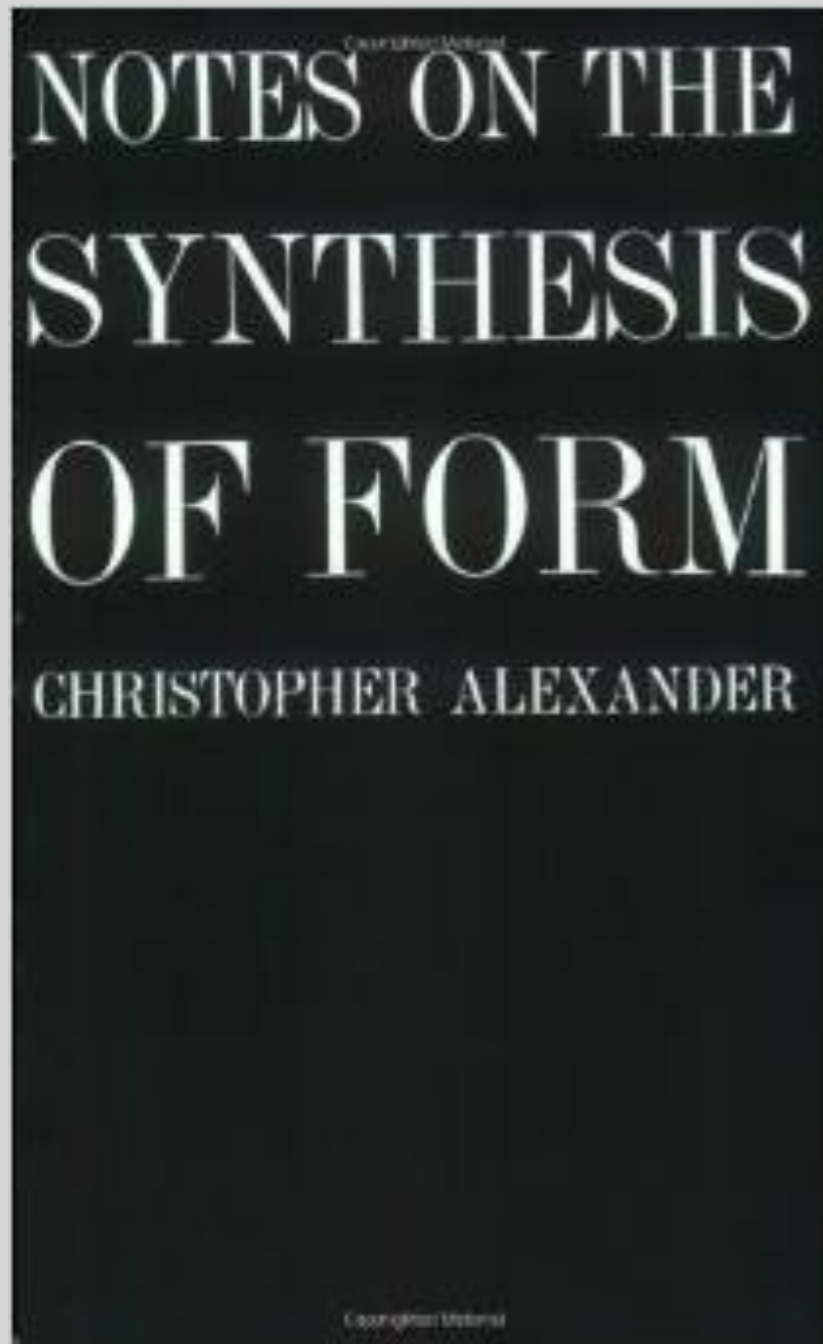
Queue

forgery: people enter with fake tickets

scalping: no tickets available, prices rocket

database lost: all issued tickets now invalid

aside: misfits



Such a list of requirements is potentially endless... But if we think of the requirements from a negative point of view, as potential misfits, there is a simple way of picking a finite set. This is because it is through misfit that the problem originally brings itself to our attention. We take just those relations between form and context which obtrude most strongly, which demand attention most clearly, which seem most likely to go wrong. We cannot do better than this.

so how to prevent misfits?

avoid premature design

eg, conventional requirements

analyze concept design

articulate purpose & mechanism

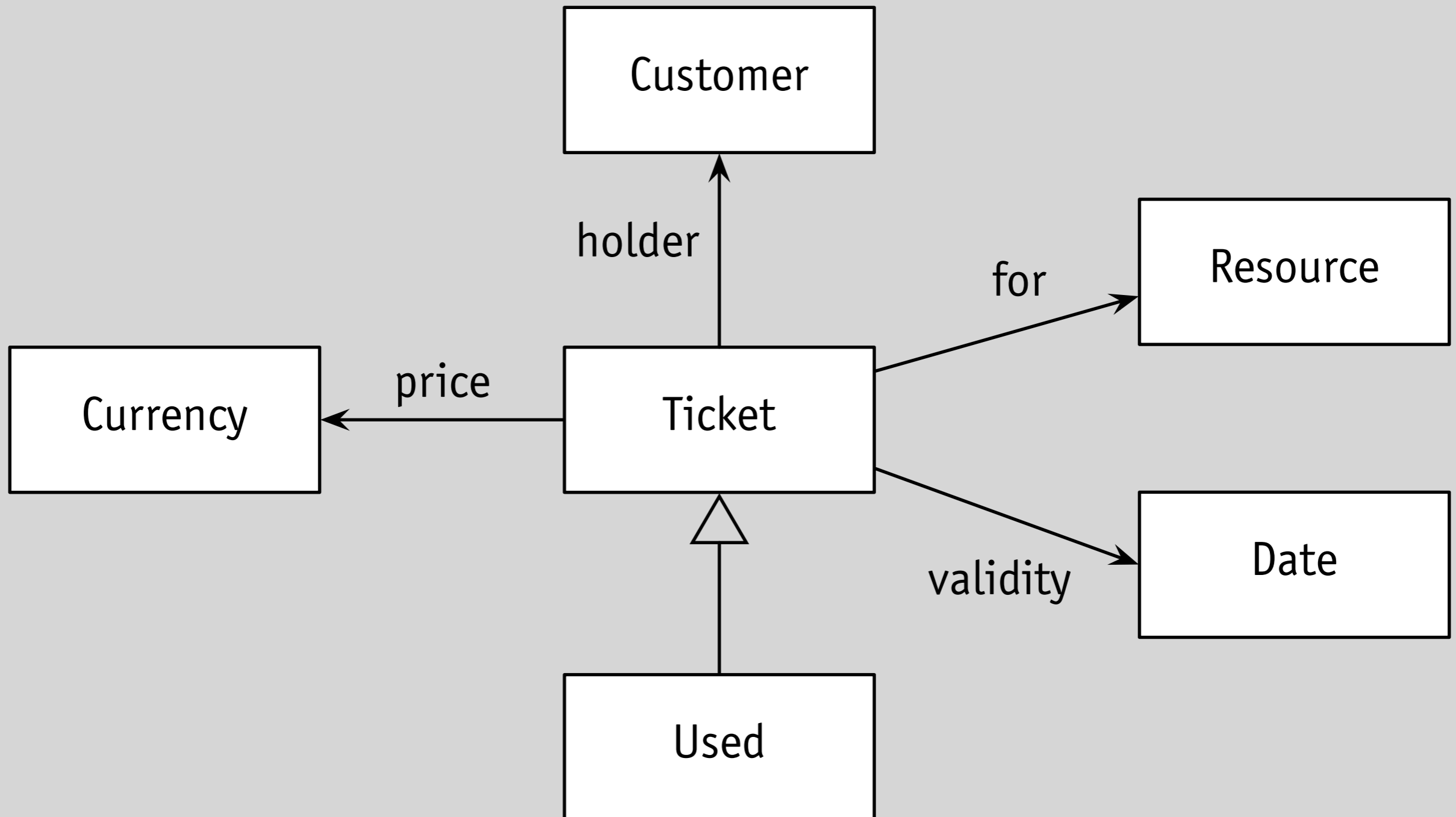
learn from the past

most concepts used before!

concept
parts

the data model

what the concept is *about*



the purpose

what the concept is *for*

**motivating
purpose**

get visitors to pay

separate use of resource from
granting of use

**refined
generic
purpose**

the actions

what the concept *does*

**a joint
action**

issue: exchange money for ticket
validate: check that ticket is valid
use: customer uses resource

**a user
action**

**a system
action**

the operational principle

how the concept *works*

an archetypal
scenario

explains how
to use the
concept

shows how
purpose is
fulfilled

“user requests that system
issue ticket; user presents ticket
for **validation**; if successful,
user can then **use** resource”

**generic
OP**

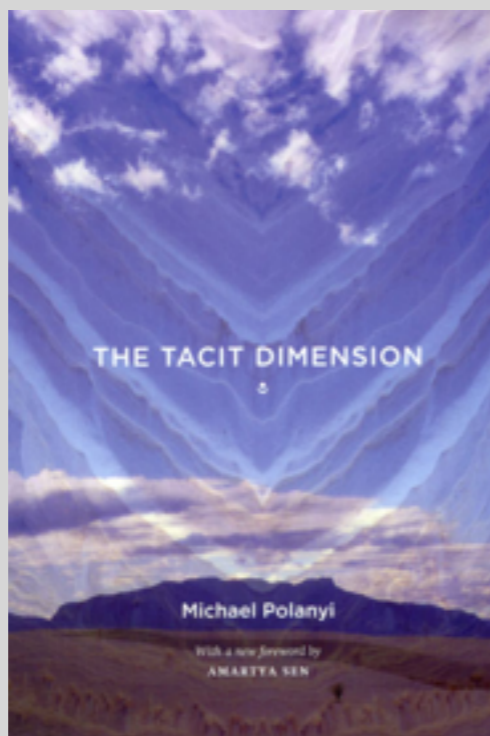
“you insert some money and
the machine gives you a ticket;
then you insert the ticket into
the turnstile, and it lets you in”

**specific
OP**

aside: science vs. engineering



Michael Polanyi



Engineering includes the operational principles of machines... Physics and chemistry, on the other hand, include no knowledge of the operational principles of machines. Hence a complete physical and chemical topography of an object would not tell us whether it is a machine, and if so, how it works, and for what purpose.

ticket concept

<i>name</i>	Ticket
<i>purpose</i>	separate use of resource from granting of use
<i>sample uses</i>	event sales; software licensing; road pricing; flights
<i>mechanism</i>	<i>data, actions, operational principle</i>
<i>variants</i>	transferable, one-time/repeating, open/expiring
<i>misfits</i>	forgery, scalping, transfers
<i>related to</i>	Reservation, Coupon, Credit

dropbox
revisited

folder

purpose

organize files with
localized namespaces

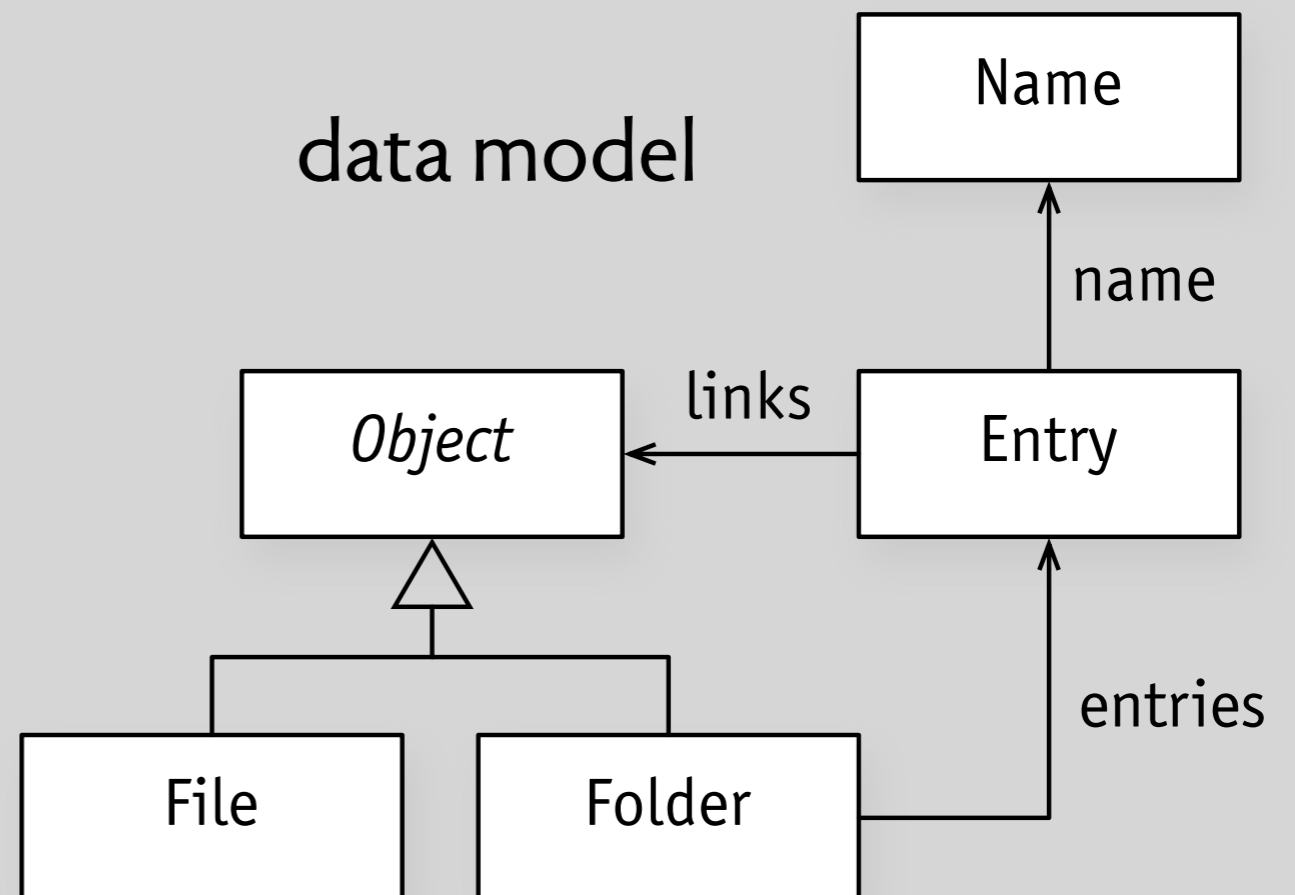
misfits

delete is not destroy
new name, new file

operational principle

if you create a folder
with a name N in a
folder that has a
pathname P, and put
objects inside it, you
can then access them
at the name P/N

data model



synchronized folder

purpose

keep copies of files
consistent across
machines

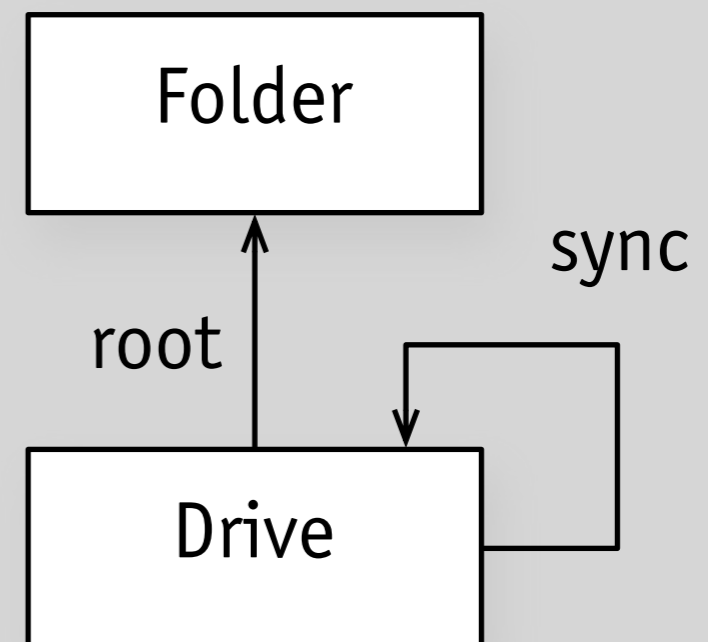
operational principle

if you say two drives
should be synced,
then changes in one
are copied to the
other

misfits

propagated deletion
propagated temps

data model



shared folder

purpose

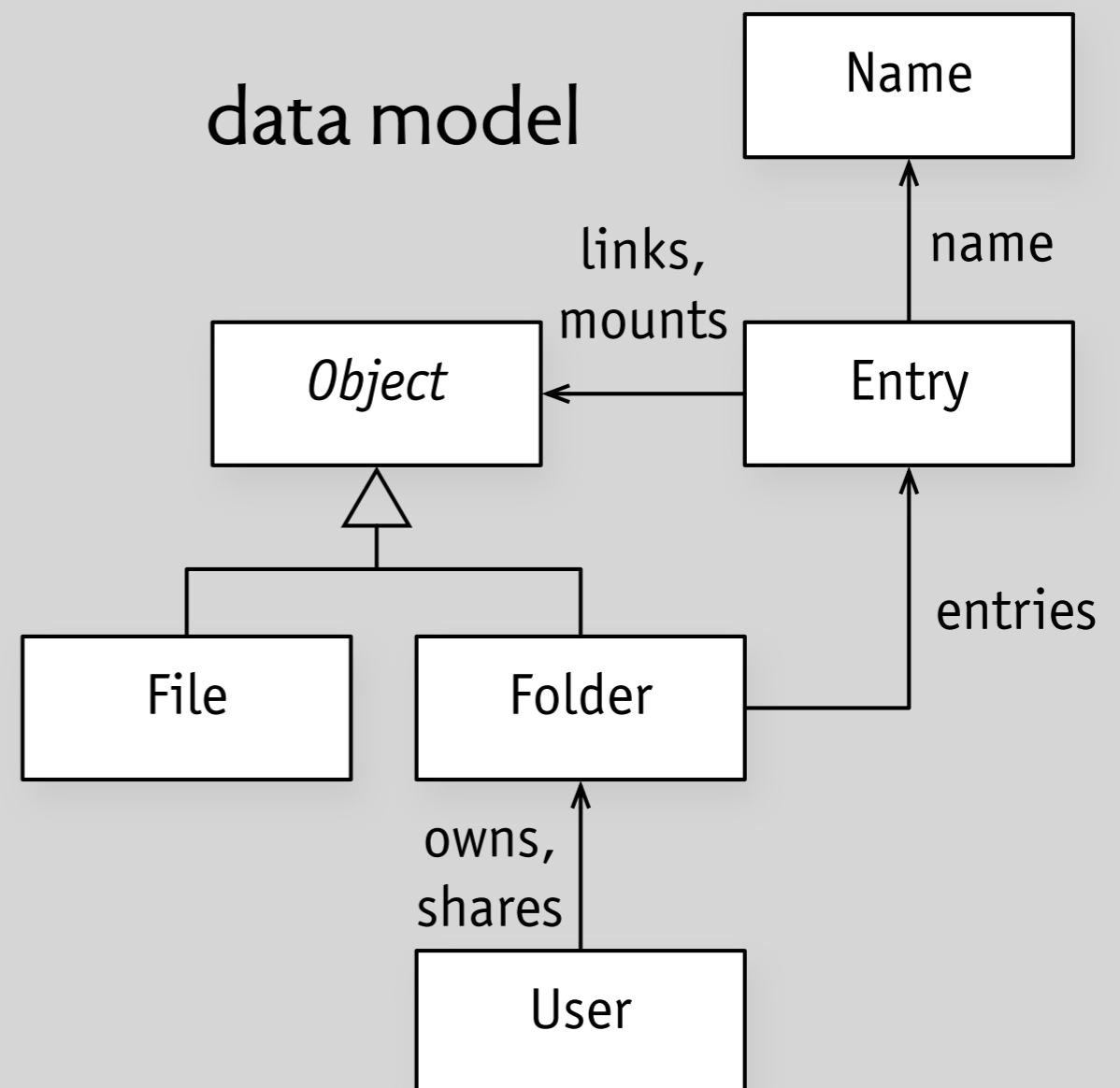
allow users to share files and folders

operational principle

if a user who owns a folder shares it with another user, that user can mount the folder in a folder of theirs and then read and write it

misfits

can't share subfolder



how to
evaluate
a concept

the fundamental principle

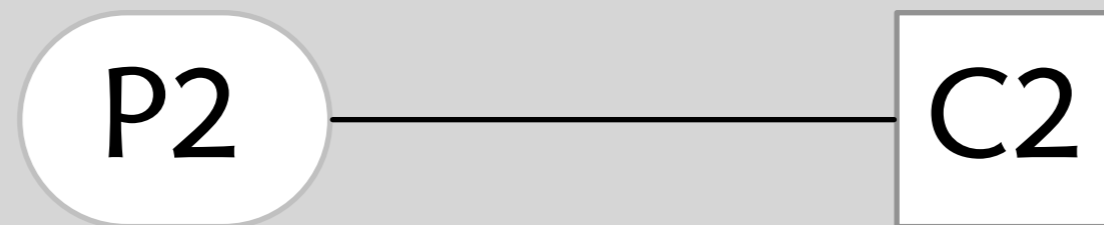
in a well-designed system
each concept is motivated by one purpose



the ideal mapping

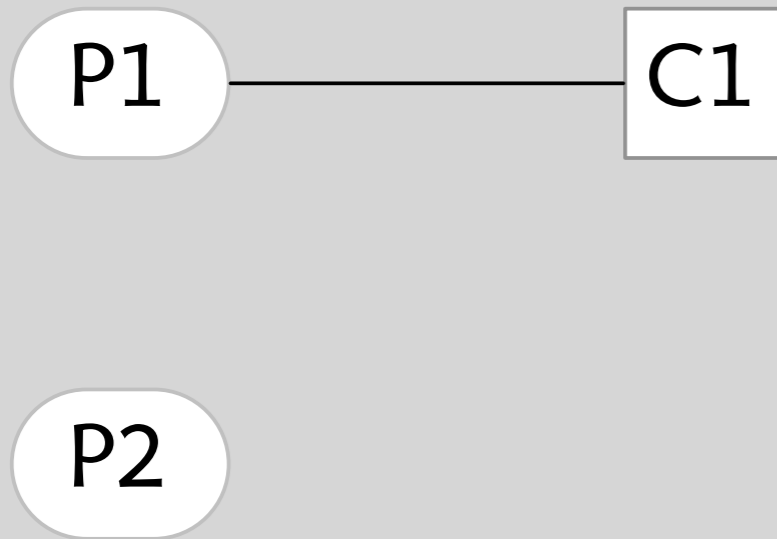
purposes

concepts

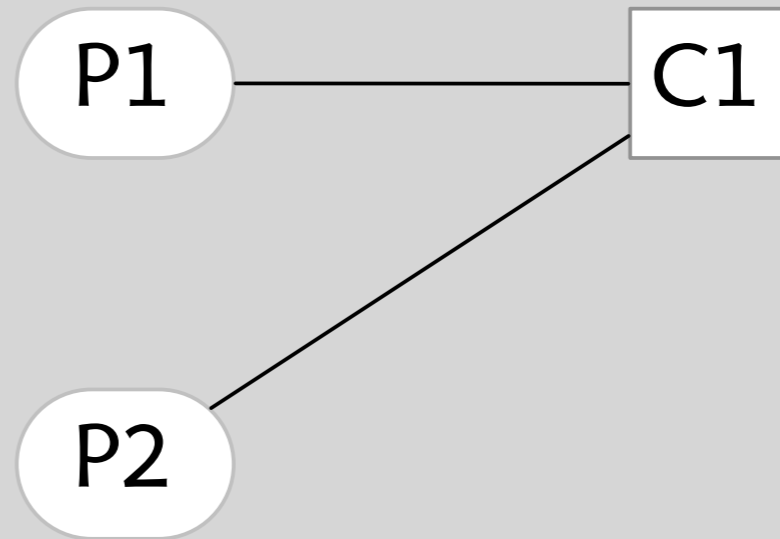


4 bad smells

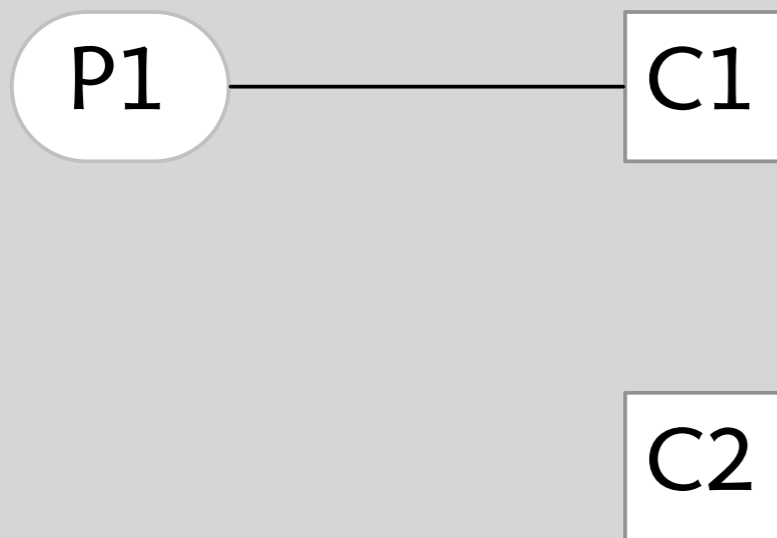
unfulfilled purpose



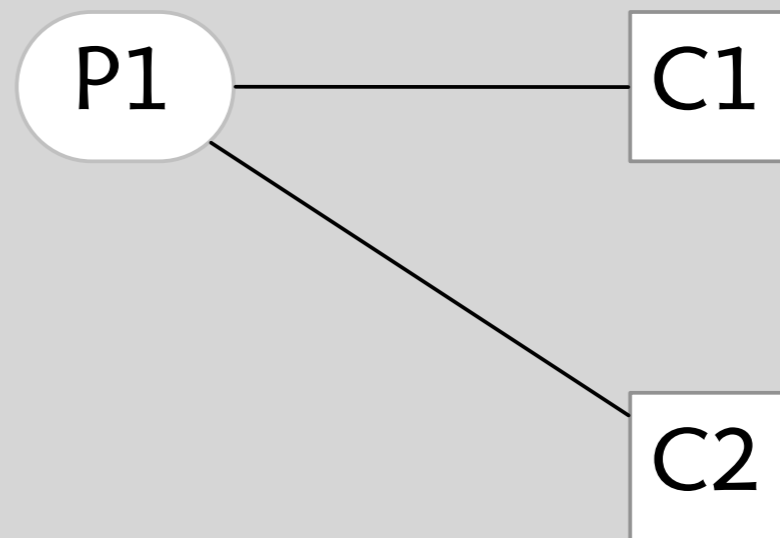
overloaded concept



unmotivated concept



redundant concepts



happy
concepts

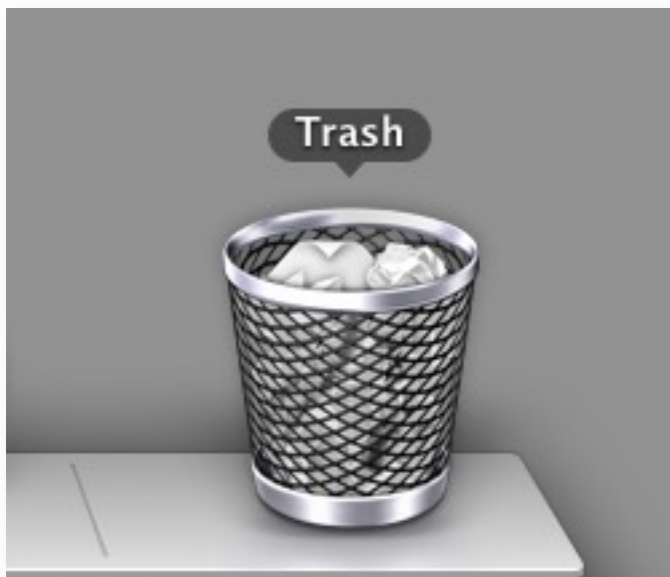
trash

purpose: allow undo of deletions

operational principle: if you delete a file, it moves to a special folder; you can restore from there, but emptying it removes contents for good (and makes space on disk)

misfit: if you delete a file on an external drive, you cannot reclaim the space until you empty the trash, but then you'll lose the ability to restore files deleted from the main drive

misfit: if you delete an old file and change your mind, you may not be able to find it again in the trash (if there are many deleted files and you forgot the file's name)



concept: trash

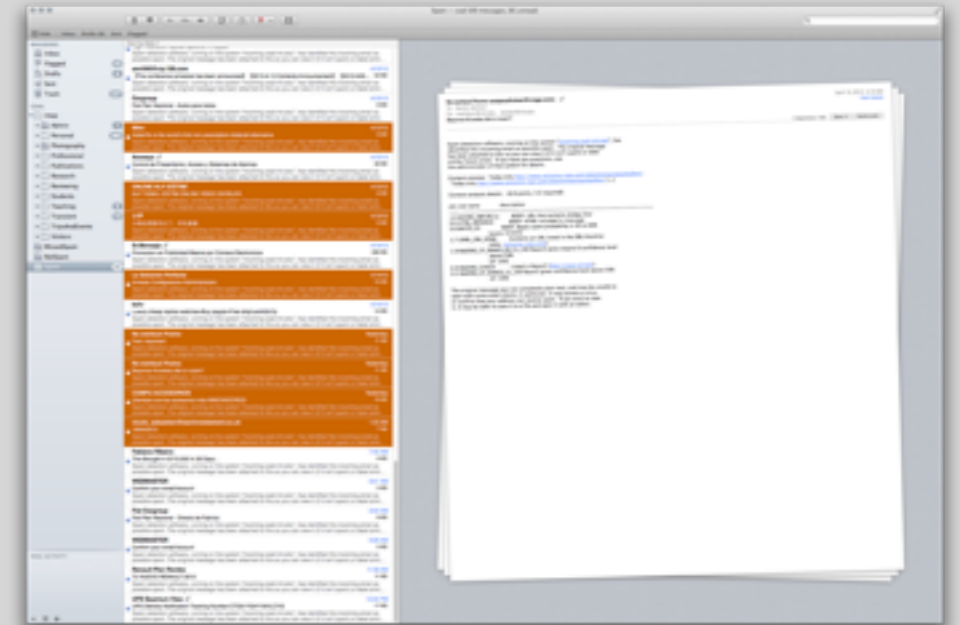
selection



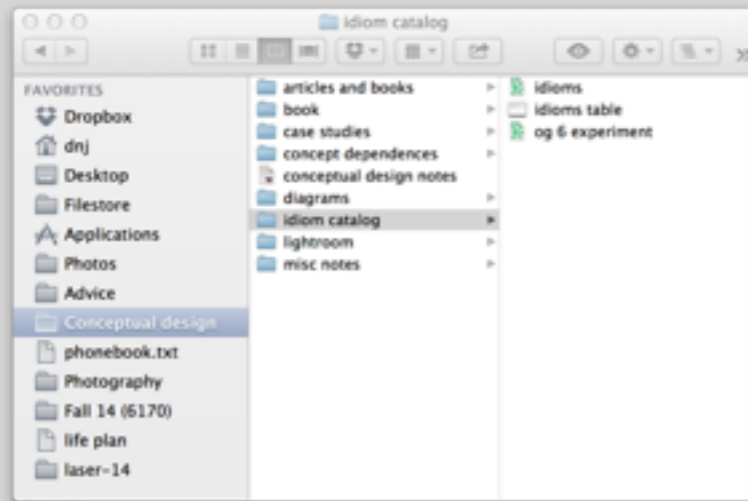
slides in Keynote



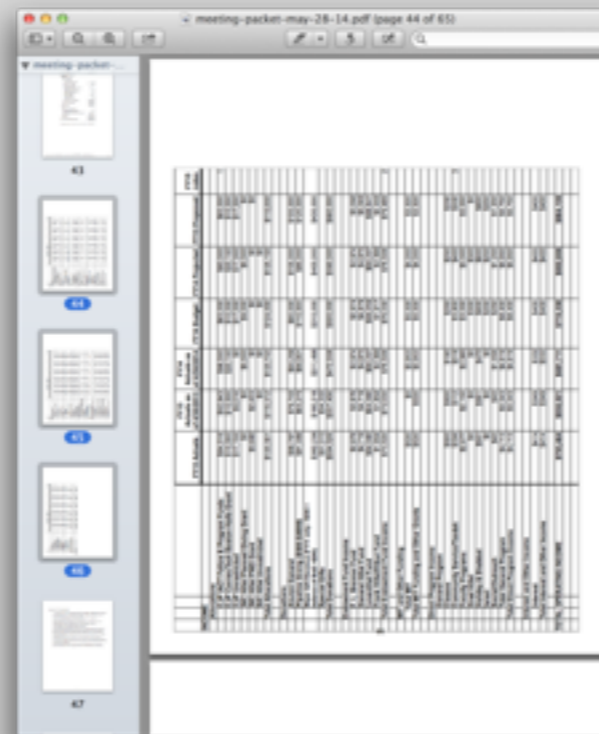
photos in Adobe Lightroom



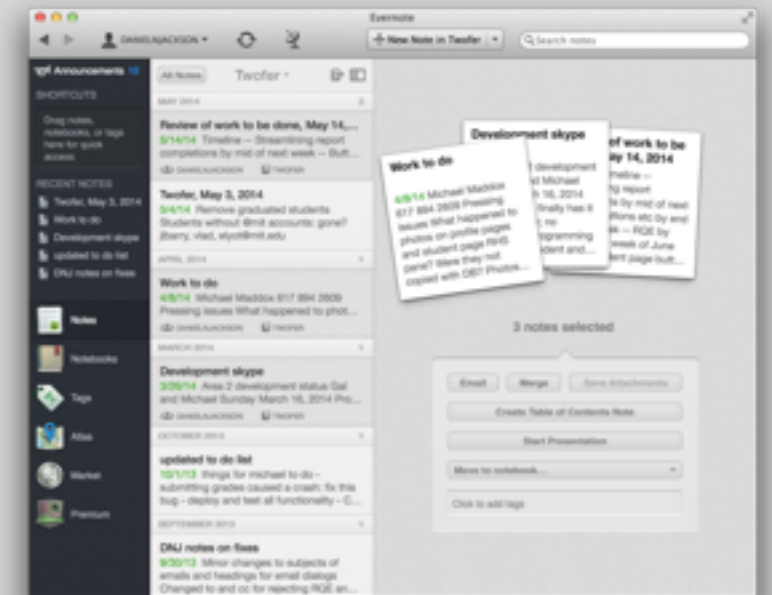
messages in Apple Mail



objects in OS X Finder

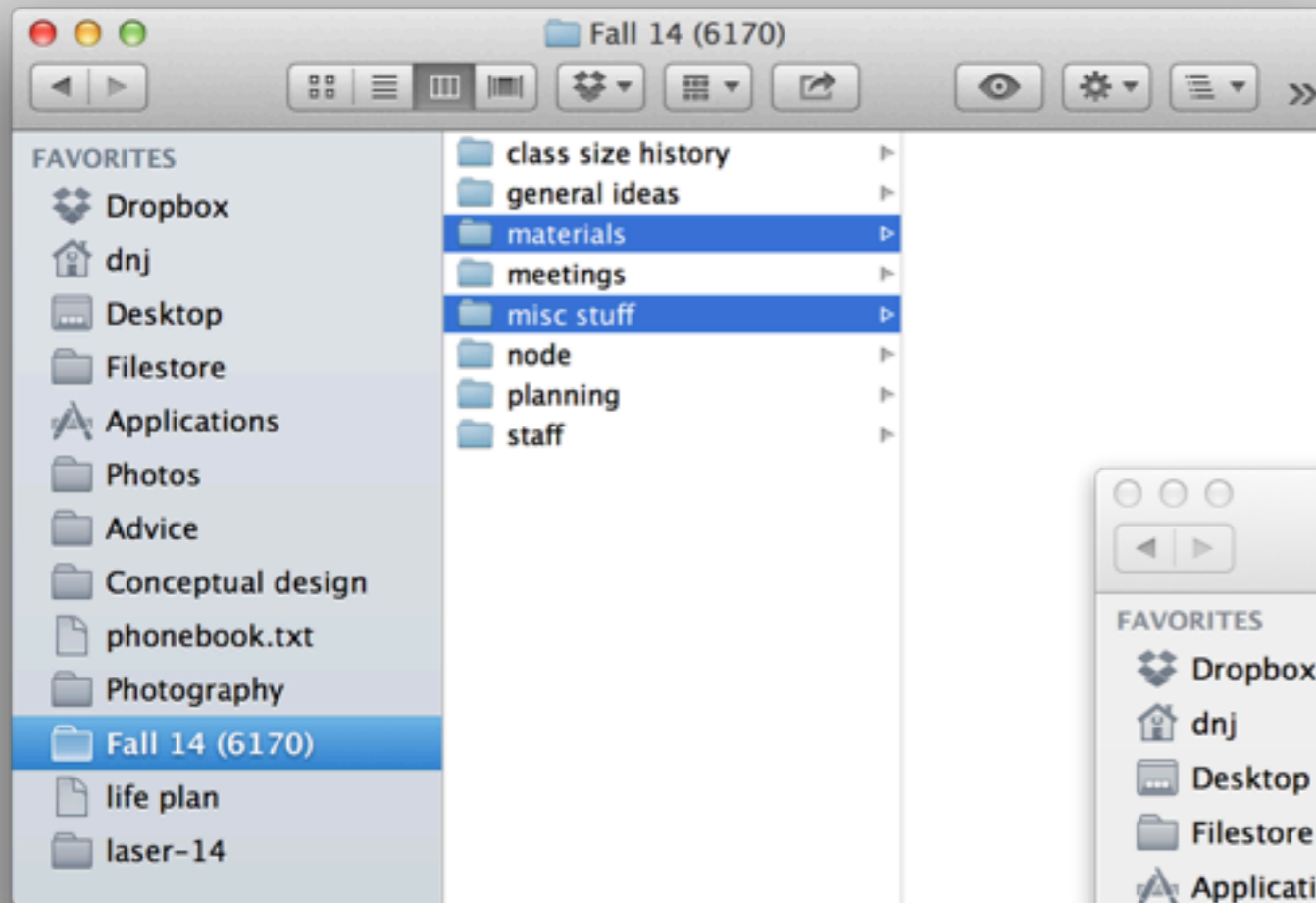


thumbnails in Preview

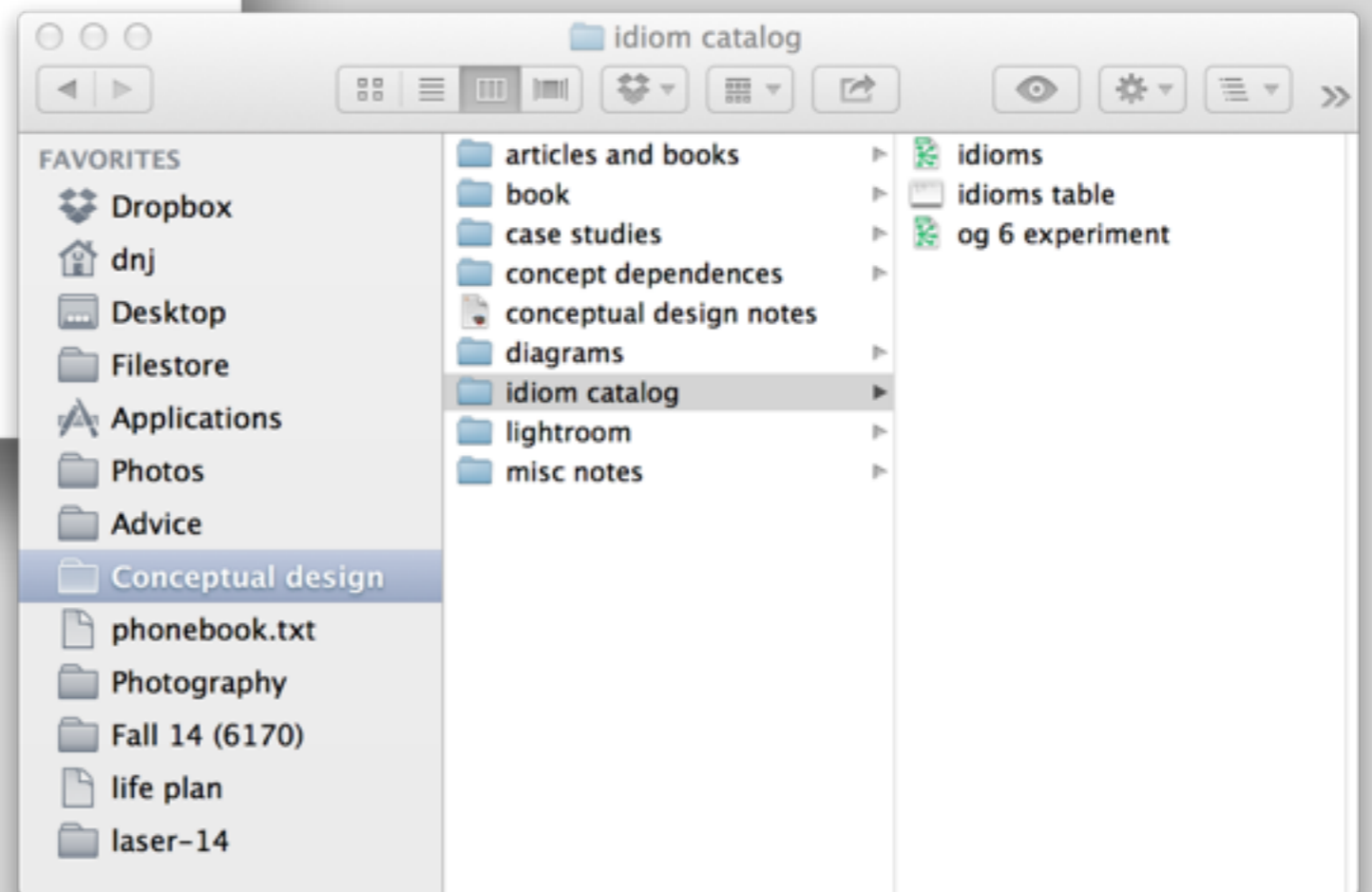


notes in Evernote

subtlety selection scope



subset of selection in scope



subset of selection out of scope

subtlety active element

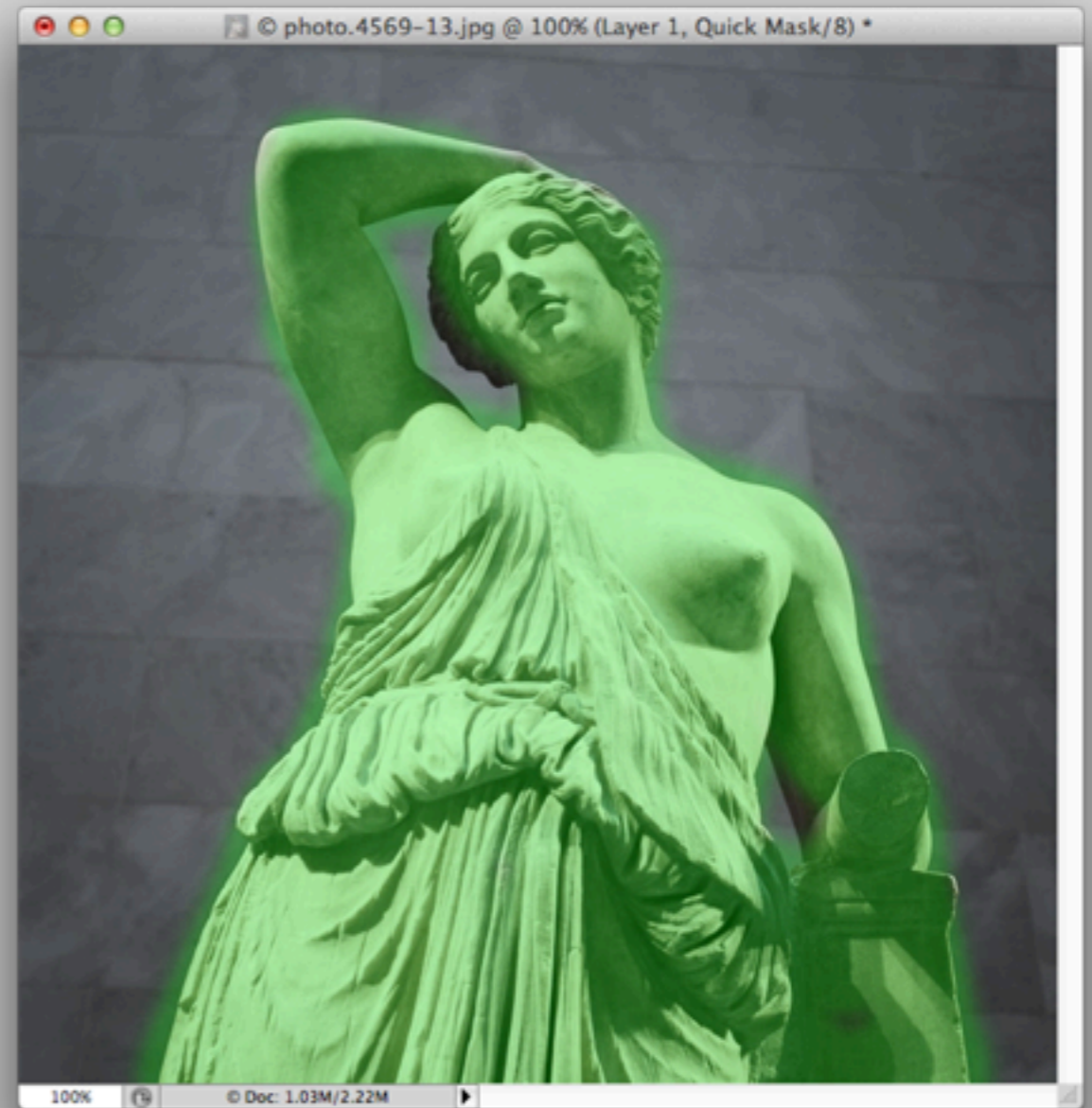


Adobe Lightroom: brightest thumbnail is the “active photo”

subtlety continuous selection

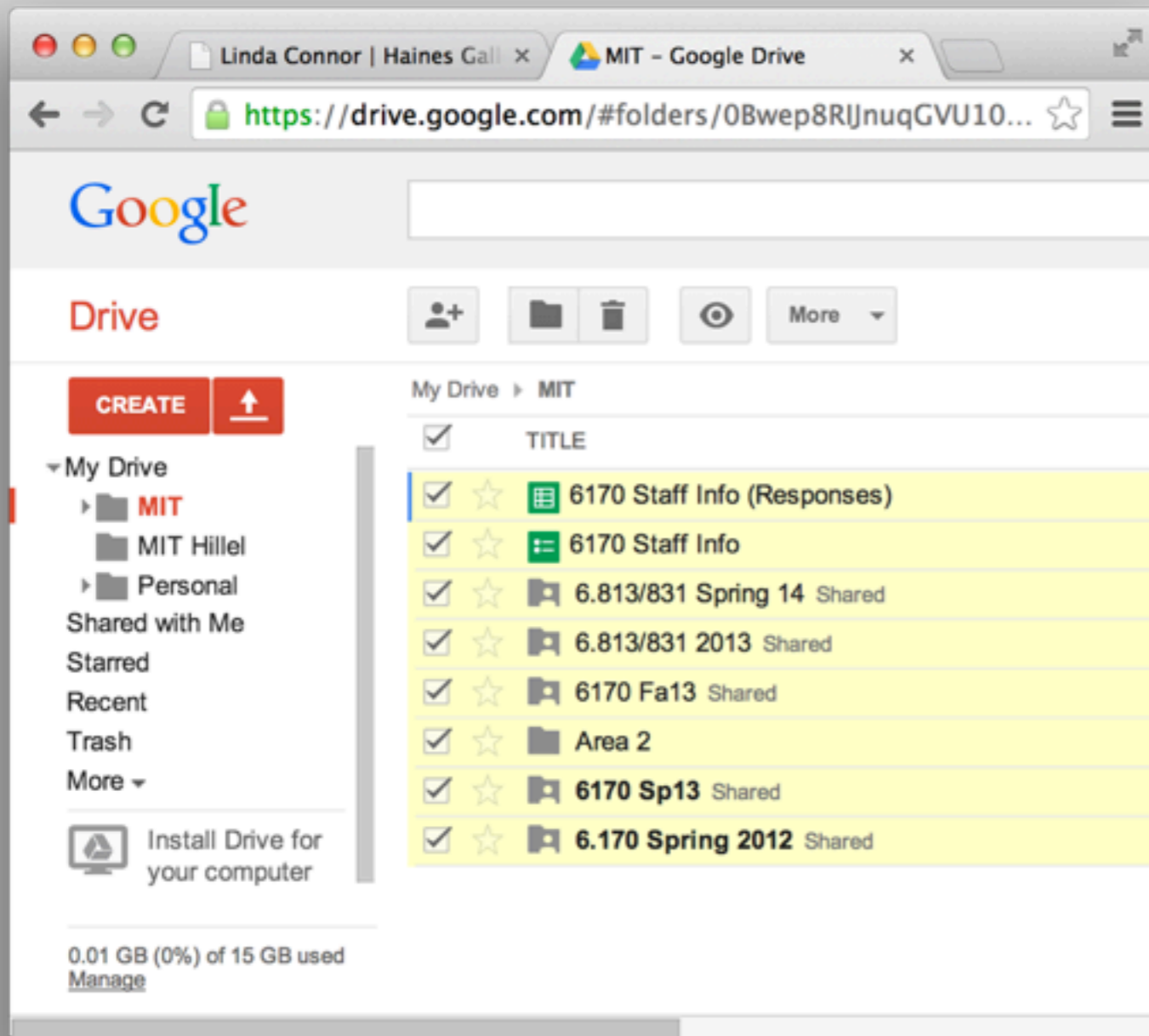


Photoshop: outline shown with
"marching ants"

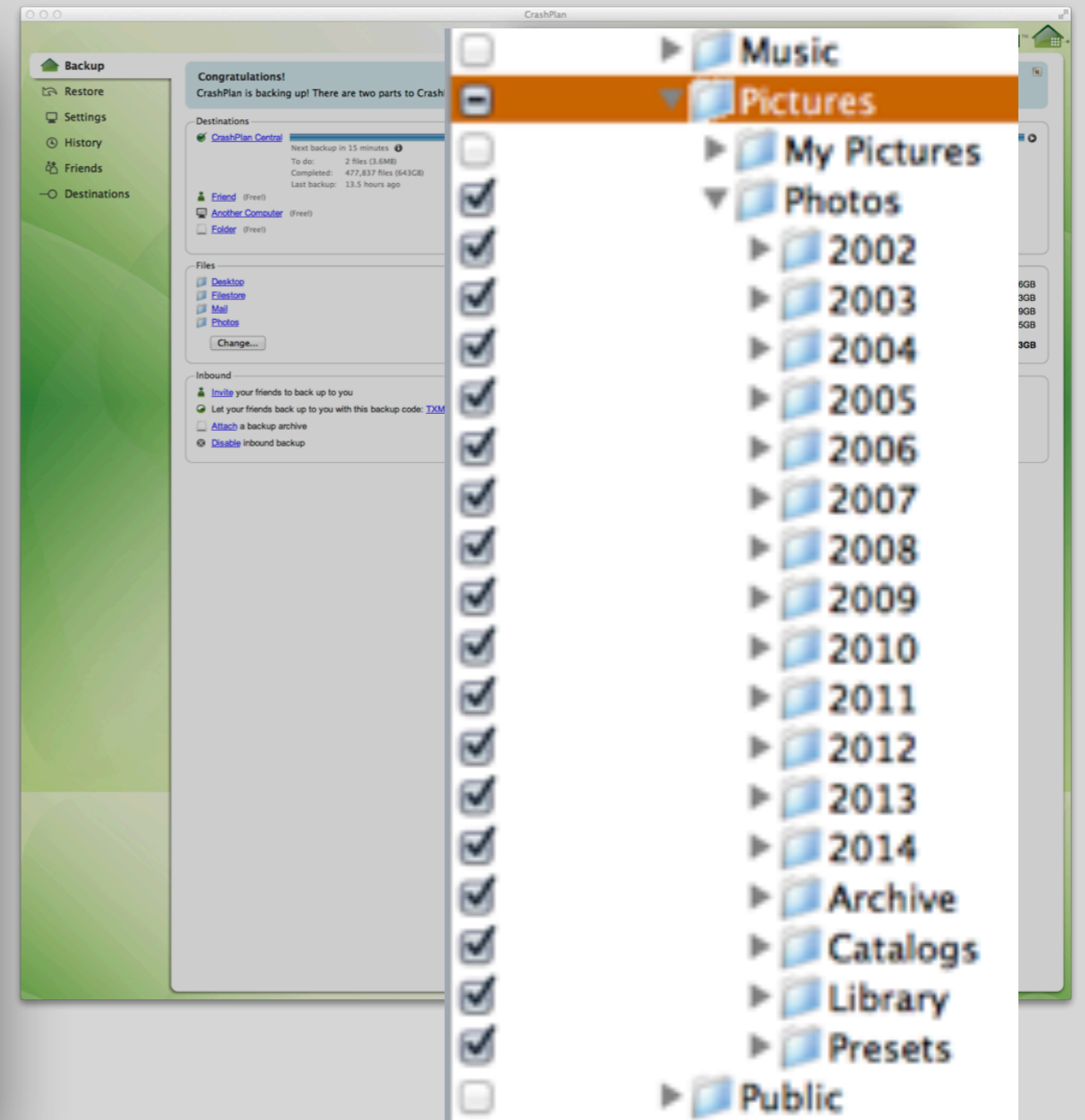


Photoshop: selection shown in
Quick Mask mode

subtlety folder selection



Google Drive: selecting folder = selecting children



CrashPlan: selecting folder = selecting all future children

selection: concept parts

<i>name</i>	Selection
<i>purpose</i>	apply operation in aggregate to many elements
<i>sample uses</i>	text formatting in word processors (eg Word); styling in CSS; color themes in Powerpoint.
<i>mechanism</i>	<i>data, actions, operational principle</i>
<i>variants</i>	scope, active element, continuous, hierarchy
<i>misfits</i>	accidental deletion?
<i>related to</i>	Group, Folder

sad
concepts

email categories

gmail's categories

Primary			Social 23 new		Promotions 100+ new		+
<input type="checkbox"/>	<input type="star"/>	<input type="arrow"/>	Google	New sign-in from Chrome on Mac - New sign-in from Ct		12:30 pm	
<input type="checkbox"/>	<input type="star"/>	<input type="arrow"/>	Keith Muhammad at DeMont.	DeMontrond Auto Group - 14101 North Freeway Housto		12:19 pm	
<input type="checkbox"/>	<input type="star"/>	<input type="arrow"/>	AT&T High Speed Internet.	AT&T High Speed Internet Service Activation - Your A1		10:37 am	
<input type="checkbox"/>	<input type="star"/>	<input type="arrow"/>	Keith Muhammad at DeMont.	DeMontrond Auto Group - 14101 North Freeway Housto		Aug 26	
<input type="checkbox"/>	<input type="star"/>	<input type="arrow"/>	betterbatonrougejobs.com	Job Update -- 2015-08-26 - Looking For An Advantage W		Aug 26	

category tab settings

Categories:

[Learn more](#)

- Primary
- Social
- Promotions
- Updates
- Forums

Starred messages

- Include starred in Primary



Choose which message categories to show as inbox tabs. Other messages will appear in the Primary tab.

some reactions

Google gets it terribly, terribly wrong with Gmail tabs and makes me angry



By [Mark Wilson](#)

Published 2 years ago



71 Comments



Like many Gmail users, I greeted the [news of the introduction of tabs to the interface](#) with a degree of anticipation -- now it was just a matter of waiting for the feature to roll out so I could try it for myself. Earlier today I was randomly signed out of my Gmail account, and after signing back in and checking the settings menus, I could see that tabs were now available to me. Excitement was short-lived, however; it quickly became apparent that this new feature is a disaster.



how google explains labels (!)

Gmail Help 

GMAIL 

FORUM

Using labels

Labels help you organize your messages into categories – work, family, to do, read later, jokes, recipes, any category you want. Labels do all the work that folders do, but with an added bonus: you can add more than one to a message.

what you can't do

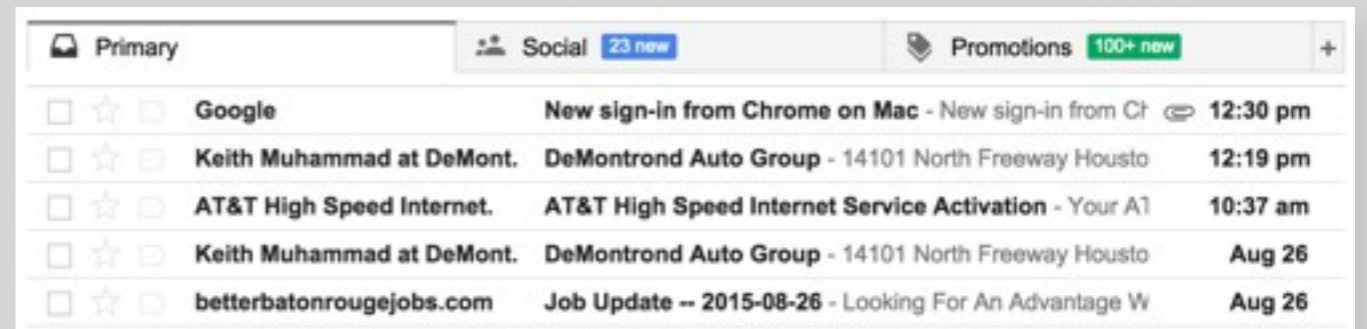
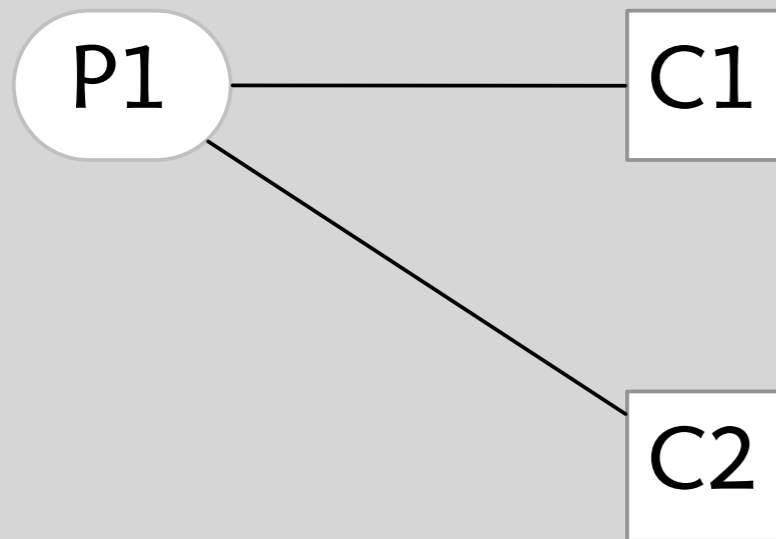
associate tabs with labels

feature available only for categories

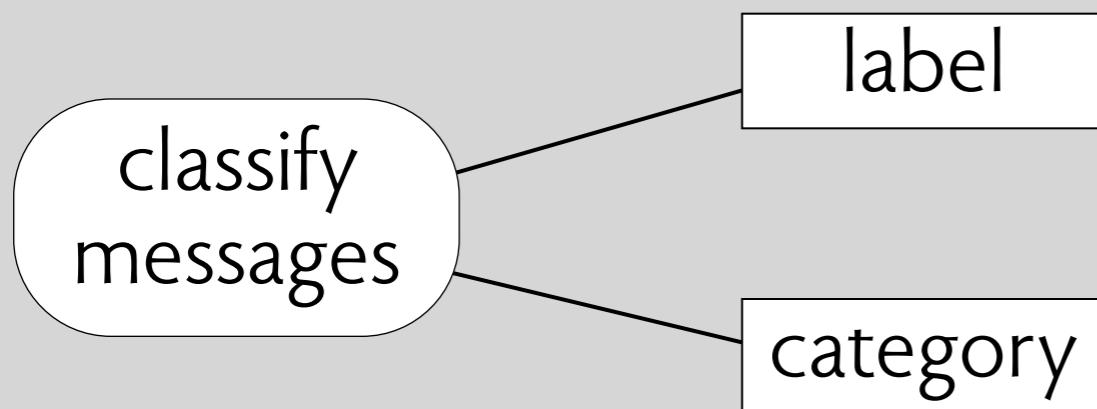
use tabs outside inbox

tabs disappear when you filter on a label

redundant concepts



category in Gmail
a redundant concept



camera settings

my camera fuji x100s



image quality setting



aspect ratio



image size setting



non-standard ratio + raw?



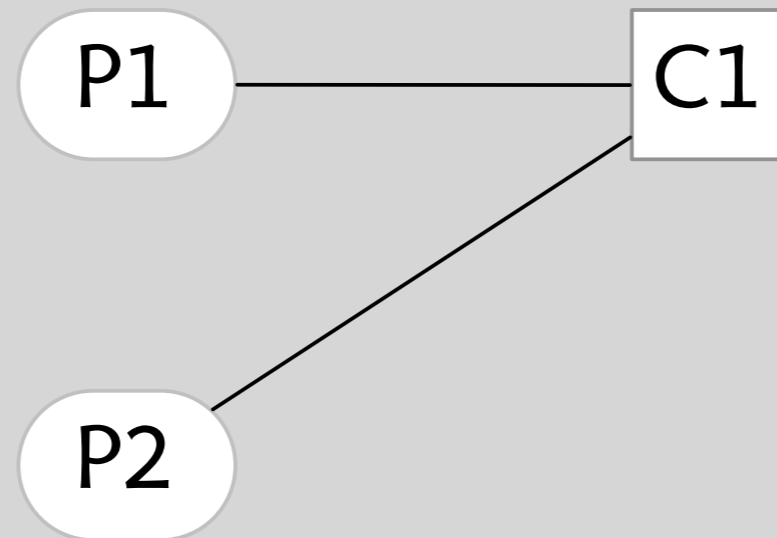
what you can't do

non-standard aspect ratio + raw
even though raw images get nice nondestructive crop!



overloaded concepts

No one can serve two masters. Either you will hate the one and love the other, or you will be devoted to the one and despise the other. [Matthew 6:24]



3 forms of overloading:

piggybacking new purpose hacked onto old concept

false convergence two purposes looked the same

emergent purpose users found second purpose for concept

piggybacking fuji camera

new purpose hacked onto old concept

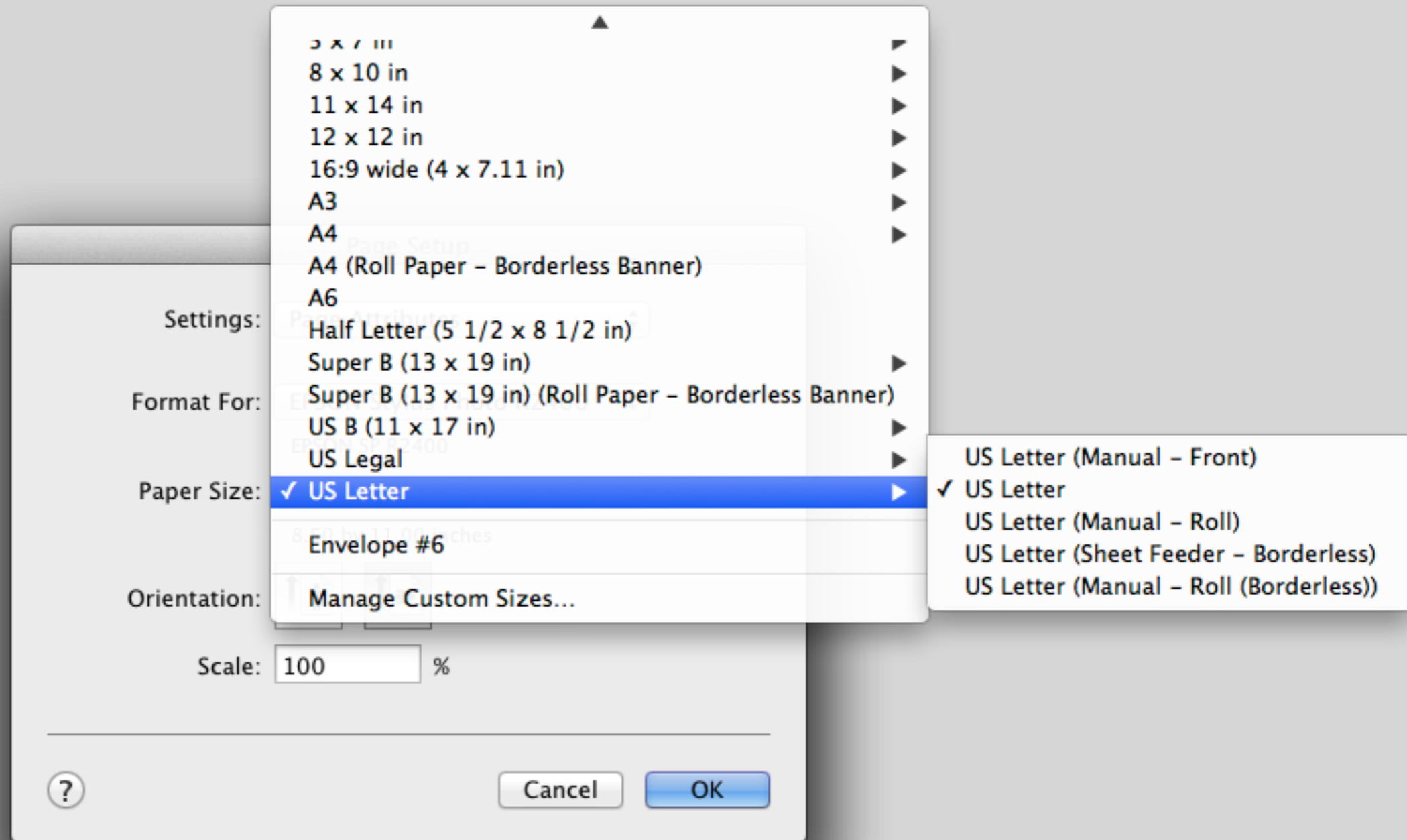


L	3:2	664
L	16:9	681
L	1:1	702
M	3:2	707
M	16:9	719
M	1:1	734
S	3:2	746

image size

aspect ratio piggybacked
on JPEG dimensions

piggybacking epson driver



result: can't create custom size for front loading
also, page size presets in Lightroom hold feed setting

false convergence

two purposes looked the same

Performance Review Form 2016

Performance Summary

Evaluate and discuss the employee's performance. Base your evaluation on the position requirements, [MIT Core Competencies](#), achievement of the goals established during the past year, and your assessment of the employee's accomplishments.

Goal Setting

State and discuss the expectations and goals for the upcoming review period. Give examples of how these goals can be met (e.g., training). How will you support the employee to accomplish these goals? [Reference here](#)

evaluation & goal setting
incompatible purposes

false convergence

two purposes looked the same



filter incoming posts
control access to my posts
distinct purposes

2011: Facebook added
subscribe/follow

emergent purpose

users find second purpose for concept

To: Daniel Jackson <dnj@mit.edu>

Re: Catch me if you can in real life!

initial purpose: summarize content

To: csail-related@lists.csail.mit.edu

Re: [csail-related] turn off the lights?

emergent purpose: show sender
if you bcc a list, subject reveals to-address

thanks to Shriram Krishnamurthi

To: Daniel Jackson <dnj@mit.edu>

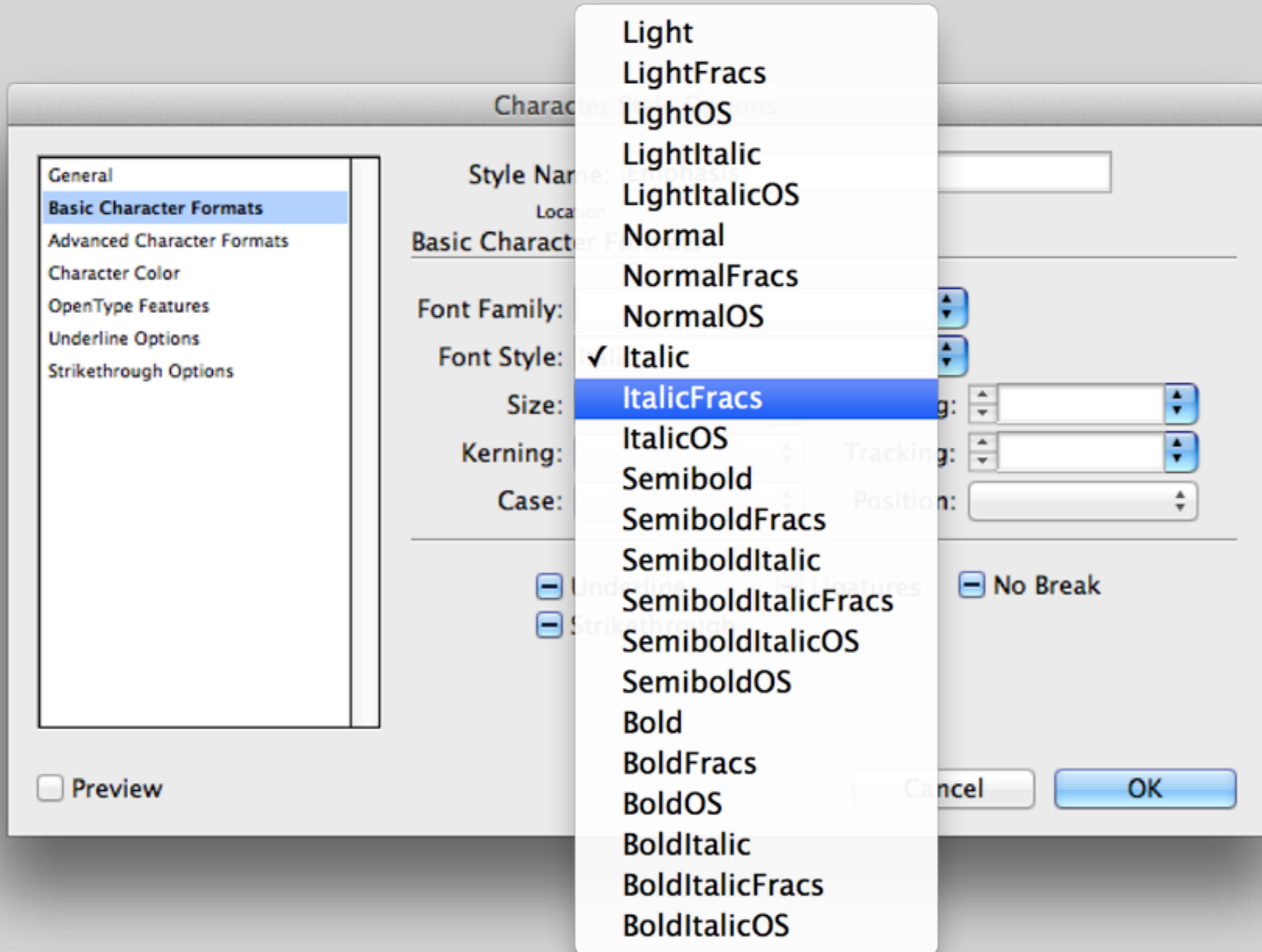
your trip reservation

emergent purpose: group by conversation
can't label reservations from Expedia by trip

thanks to Eunsuk Kang

fonts & styles

what's a font?



what you can't do

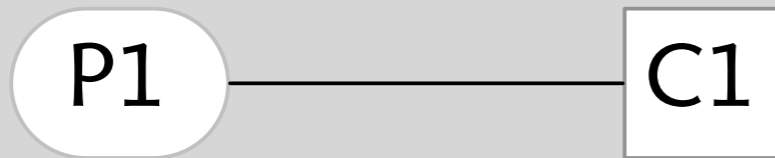
define a style that italicizes

Arno Regular to Arno Italic

Futura Book to Futura Book Oblique

Magma Light to Magma Light Italic

unfulfilled purposes



allow
typeface
independent
styling

subfamily

- Light
- LightFracs
- LightOS
- LightItalic
- LightItalicOS
- Normal
- NormalFracs
- NormalOS
- ✓ Italic
- ItalicFracs**
- ItalicOS
- Semibold
- SemiboldFracs
- SemiboldItalic
- SemiboldItalicFracs
- SemiboldItalicOS
- SemiboldOS
- Bold
- BoldFracs
- BoldOS
- BoldItalic
- BoldItalicFracs
- BoldItalicOS

Adobe Indesign:
an unfulfilled purpose

introducing a concept

New character style:

Name:

▼ Include these character attributes:

<input checked="" type="checkbox"/> Font: Magma Light	<input checked="" type="checkbox"/> Capitalization: Standard
<input checked="" type="checkbox"/> Size: 12.0 pt	<input checked="" type="checkbox"/> Superscript:
<input checked="" type="checkbox"/> Character Spacing: 0%	<input checked="" type="checkbox"/> Baseline Shift: 0.0 pt
<input checked="" type="checkbox"/> Bold: Off	<input checked="" type="checkbox"/> Underline: None
<input checked="" type="checkbox"/> Italic: Off	<input checked="" type="checkbox"/> Color: <input type="color" value="black"/>
<input checked="" type="checkbox"/> Color: <input type="color" value="black"/>	<input checked="" type="checkbox"/> Strikethrough: None
<input checked="" type="checkbox"/> Shadow: Off	<input checked="" type="checkbox"/> Color: <input type="color" value="black"/>
<input checked="" type="checkbox"/> Fill: <input type="color" value="white"/>	
<input checked="" type="checkbox"/> Ligatures: Default	<input checked="" type="checkbox"/> Advanced Font Features <small>See Typography in the Font Panel</small>
<input checked="" type="checkbox"/> Language: English	

Apply this new style on creation

Keynote '09: has subfamilies

Text

Body*

Font

Magma

Lightitalic 11 pt

B **/** U


Character Styles

Keynote 6: gone again!

rethinking Git

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.

 **Learn Git in your browser for free with Try Git.**



About

The advantages of Git compared to other source control systems.



Documentation

Command reference pages, Pro Git book content, videos and other material.



Downloads

GUI clients and binary releases for all major platforms.



Community

Get involved! Bug reporting, mailing list, chat, development and more.



Pro Git by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

 **Mac GUIs**

 **Tarballs**

 **Windows Build**

 **Source Code**

NAME

git-rebase - Forward-port local commits to the updated upstream head

SYNOPSIS

```
git rebase [-i | --interactive] [options] [--exec <cmd>] [--onto <newbase>]
           [<upstream> [<branch>]]
git rebase [-i | --interactive] [options] [--exec <cmd>] [--onto <newbase>]
           --root [<branch>]
git rebase --continue | --skip | --abort | --edit-todo
```

NAME

git-rebase - Forward-port local commits to the updated upstream head

SYNOPSIS

```
git rebase [-i | --interactive] [options] [--exec <cmd>] [--onto <newbase>]
           [<upstream> [<branch>]]
git rebase [-i | --interactive] [options] [--exec <cmd>] [--onto <newbase>]
           --root [<branch>]
git rebase --continue | --skip | --abort | --edit-todo
```

one by one, in order. Note that any commits in HEAD which introduce the same textual changes as a commit in HEAD..`<upstream>` are omitted (i.e., a patch already accepted upstream with a different commit message or timestamp will be skipped).

It is possible that a merge failure will prevent this process from being completely automatic. You will have to resolve any such merge failure and run `git rebase --continue`. Another option is to bypass the commit that caused the merge failure with `git rebase --skip`. To check out the original `<branch>` and remove the `.git/rebase-apply` working files, use the command `git rebase --abort` instead.

Assume the following history exists and the current branch is "topic":

NAME

`git-wave-stash` — wave all staged stashes next to various cherry-picked non-applied applied trees

SYNOPSIS

```
git-wave-stash --predict-whistle-tree --dodge-pack
```

NAME

`git-wave-stash` — wave all staged stashes next to various cherry-picked non-applied applied trees

SYNOPSIS

```
git-wave-stash --predict-whistle-tree --dodge-pack
```

OPTIONS

- predict-whistle-tree**
the subtree should not be flashed by a requested pack
- dodge-pack**
fast-import the histories of a few files that are parsed

SEE ALSO

[git-gouge-head\(1\)](#), [git-strip-history\(1\)](#), [git-recommend-pack\(1\)](#), [git-tilt-branch\(1\)](#)

NAME

`git-distinguish-tree` — distinguish a few non-cleaned remote trees inside various rev-listed upstreams

SYNOPSIS

```
git-distinguish-tree [ --distinguish-grope-history | --relieve-ref | --delineate-log ]
```

NAME

`git-distinguish-tree` — distinguish a few non-cleaned remote trees inside various rev-listed upstreams

SYNOPSIS

```
git-distinguish-tree [ --distinguish-grope-history | --relieve-ref | --delineate-log ]
```

--distinguish-grope-history
import the bases of a few files that are archived

--relieve-ref
use ref to checkout `origins/stages/` to an exported ref

--delineate-log
save the histories of a few stages that are failed

SEE ALSO

[git-engineer-submodule\(1\)](#), [git-lecture-archive\(1\)](#)

NAME

`git-control-stash` — control some non-bundled staged stashes over any shown submodules

SYNOPSIS

```
git-control-stash [ --steer-stash | --scout-area | --collide-index-origin ]
```

DESCRIPTION

git-control-stash(1) Manual Page

NAME

`git-control-stash` — control some non-bundled staged stashes over any shown submodules

SYNOPSIS

```
git-control-stash [ --steer-stash | --scout-area | --collide-index-origin ]
```

--steer-stash

without this argument, `git-scan-commit --grab-branch` cherry-picks indices that fck the specified archives

--scout-area

the tag can not be stacked by a merged tree

--collide-index-origin

the change should not be blocked by a cloned stash

SEE ALSO

[git-page-path\(1\)](#), [git-pocket-stash\(1\)](#), [git-race-head\(1\)](#)

GIT

|<

< PREV

RANDOM

NEXT >

>|

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.



|<

< PREV

RANDOM

NEXT >

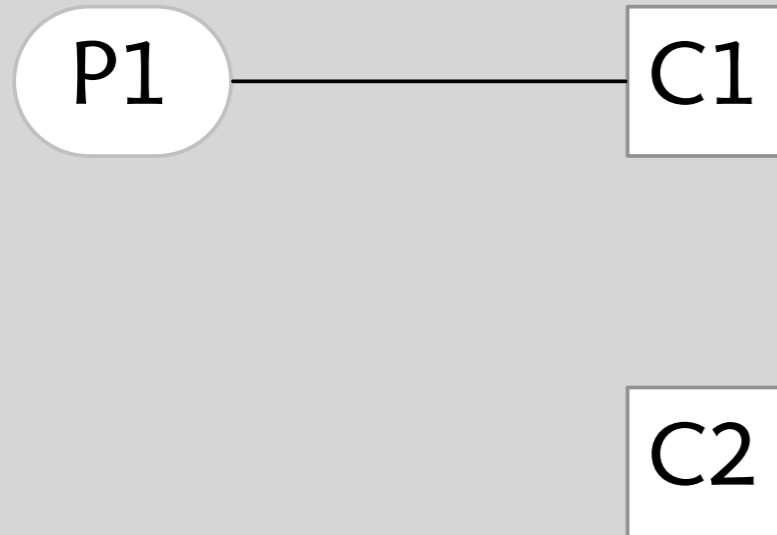
>|

PERMANENT LINK TO THIS COMIC: [HTTP://XKCD.COM/1597/](http://xkcd.com/1597/)

IMAGE URL (FOR HOTLINKING/EMBEDDING): [HTTP://IMGS.XKCD.COM/COMICS/GIT.PNG](http://imgs.xkcd.com/comics/git.png)

conceptual problems in Git

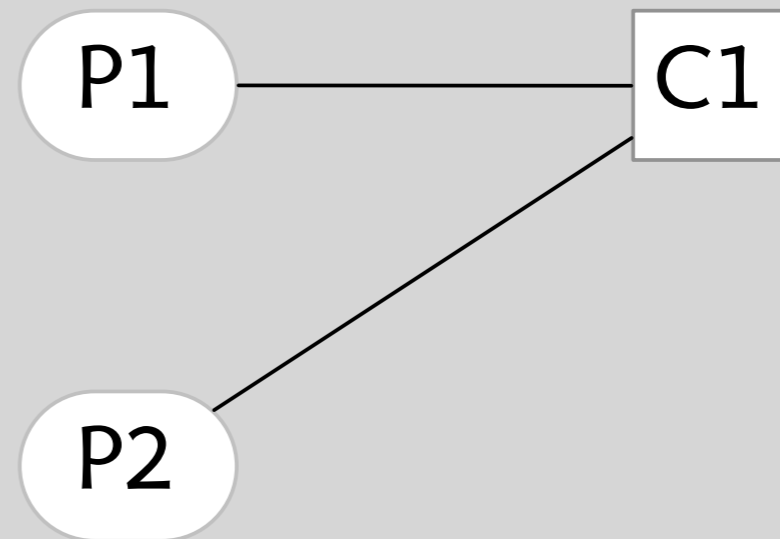
unmotivated concept



example: stash

pseudo purpose:
overcome misfits in
branching concept

overloaded concept



example: commit

P1. group logically
related changes
P2. save files to protect
against loss

Misfit	Question	Upvotes	Views	
Saving Changes	Q1	Using Git and Dropbox together effectively?	927	215523
	Q2	Backup a Local Git Repository	122	78674
	Q3	Fully backup a git repo?	54	37502
	Q4	Is it possible to push a git stash to a remote repository?	105	30820
	Q5	Git fatal: Reference has invalid format: refs/heads/master	90	25717
	Q6	Is "git push -mirror" sufficient for backing up my repository?	34	18415
	Q7	How to back up private branches in git	33	10580
Switching Branches	Q8	The following untracked working tree files would be overwritten by checkout	365	378331
	Q9	git: Switch branch and ignore any changes without committing	148	129120
	Q10	Why git keeps showing my changes when I switch branches (modified, added, deleted files) no matter if I run git add or not?	47	10524
Detached Head	Q11	Git: How can I reconcile detached HEAD with master/origin?	784	397694
	Q12	Fix a Git detached head?	490	397985
	Q13	Checkout GIT tag	125	98328
	Q14	git push says everything up-to-date even though I have local changes	113	79203
	Q15	Why did my Git repo enter a detached HEAD state?	202	78856
	Q16	Why did git set us on (no branch)?	65	41866
	Q17	gitx How do I get my 'Detached HEAD' commits back into master	136	42794
File Rename	Q18	Handling file renames in git	315	242864
	Q19	Is it possible to move/rename files in git and maintain their history?	367	153701
	Q20	Why might git log not show history for a moved file, and what can I do about it?	34	17099
	Q21	How to REALLY show logs of renamed files with git?	60	12923
File Tracking	Q22	Why does git commit not save my changes?	177	142189
	Q23	Git commit all files using single command	165	141815
Untracking File	Q24	Ignore files that have already been committed to a Git repository	1588	387112
	Q25	Stop tracking and ignore changes to a file in Git	975	353136
	Q26	Making git "forget" about a file that was tracked but is now in .gitignore	1458	286435
	Q27	git ignore files only locally	562	120700
	Q28	Untrack files from git	218	140663
	Q29	Git: How to remove file from index without deleting files from any repository	110	61498
	Q30	Ignore modified (but not committed) files in git?	135	38293
	Q31	Ignoring an already checked-in directory's contents?	169	49692
	Q32	Apply git .gitignore rules to an existing repository [duplicate]	40	28286
	Q33	undo git update-index -assume-unchanged <file>	165	37262
	Q34	using gitignore to ignore (but not delete) files	55	23381
	Q35	How do you make Git ignore files without using .gitignore?	58	23709
	Q36	Can I get a list of files marked -assume-unchanged?	191	20184
	Q37	Keep file in a Git repo, but don't track changes	74	15572
	Q38	Committing Machine Specific Configuration Files	58	5934
Empty Directory	Q39	How can I add an empty directory to a Git repository?	2383	432218
	Q40	What are the differences between .gitignore and .gitkeep?	841	121484
	Q41	How to .gitignore all files/folder in a folder, but not the folder itself? [duplicate]	227	80119

gitless: a reworking of git

Gitless: a version control system

Fork me on GitHub

About Gitless

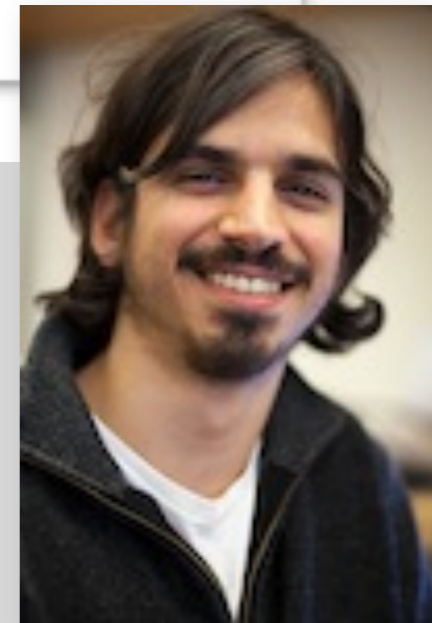
Gitless is an experimental version control system built on top of Git. Many people complain that Git is hard to use. We think the problem lies deeper than the user interface, in the concepts underlying Git. Gitless is an experiment to see what happens if you put a simple veneer on an app that changes the underlying concepts. Because Gitless is implemented on top of Git (could be considered what Git pros call a "porcelain" of Git), you can always fall back on Git. And of course your coworkers you share a repo with need never know that you're not a Git aficionado.

Check out the [documentation](#) to get started. If you are a novice user that never used any version control system the documentation should be enough to get you started. If you are a Git pro looking to see what's different from your beloved Git you'll be able to spot the differences by glancing through the [Gitless vs. Git](#) section.

Download

- [Mac OS X Binary \(.tar.gz\)](#)
- [Linux Binary \(.tar.gz\)](#)
- [Source Code \(.tar.gz\)](#)

For installation instructions [see the readme file](#). After installation, you should be able to execute the `gl` command. The current Gitless version is 0.7 which was released on 4/2015 ([release notes](#)).



Santiago
Perez De Rosso

example: branch



concept: branch

purpose: support independent line of development

operational principle: when you switch branches, your working directory is synchronized with the new branch, and you can make and commit changes which will be invisible on other branches; when you're done, you can merge the branch into the master branch...

misfit: can't switch branches with uncommitted changes; can stash, but only if no conflicts...

fix: give branch its own working directory; when you switch branches, the working directory changes too, and the working directory associated with the previous branch is preserved

user study

experiment design

within-subjects
two hour long sessions
six tasks per session
observation + surveys

tasks

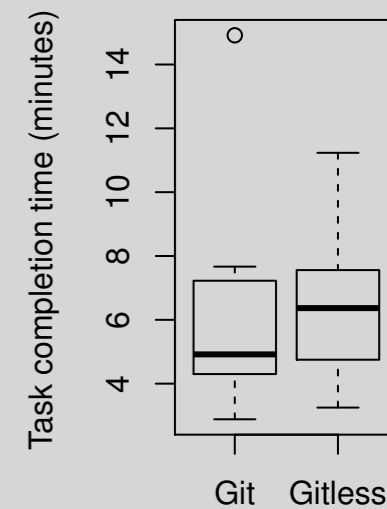
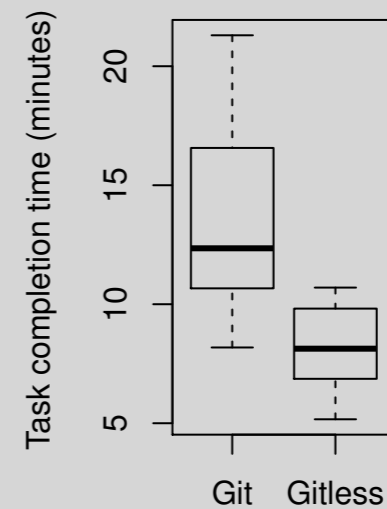
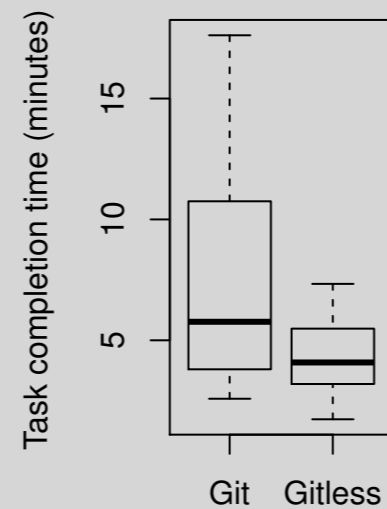
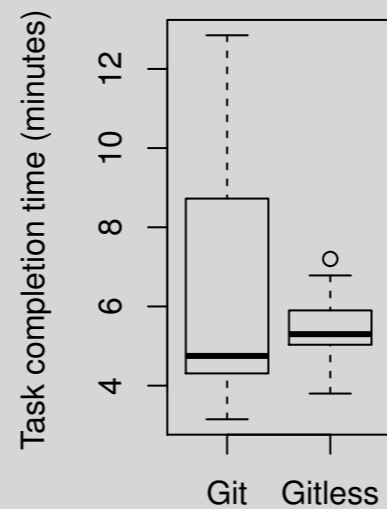
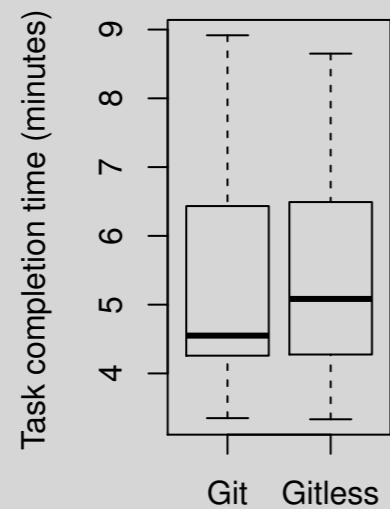
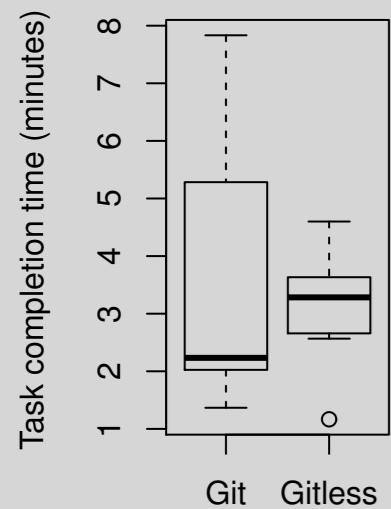
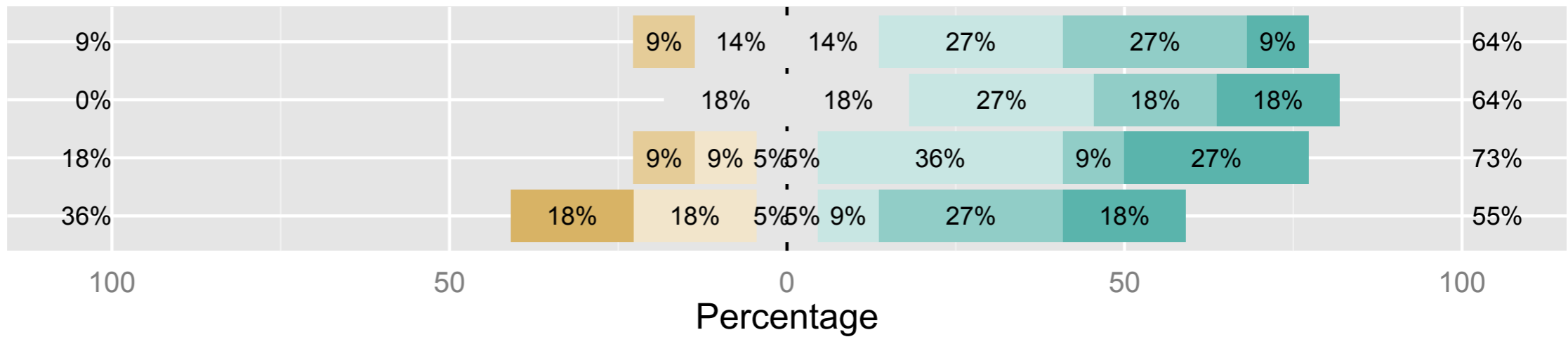
commit staged modified file
create and switch to branch
switch with changes that conflict
switch leaving changes behind
switch in the middle of merge
undo commit

subjects

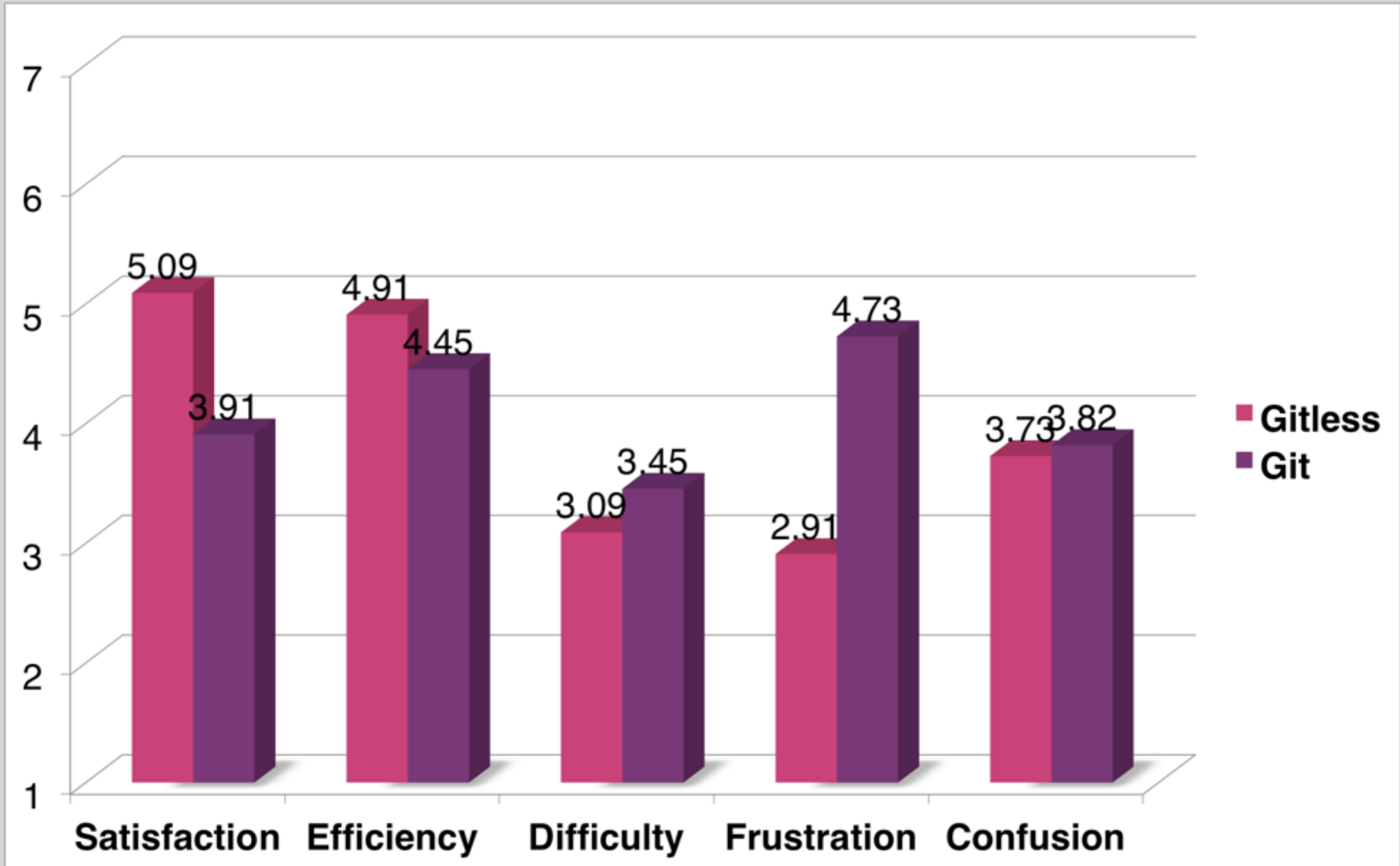
11 = 3 industry + 3 research + 5 student
Git: 4 novices, 3 regular, 4 experts
Gitless: none used before

results of a user study

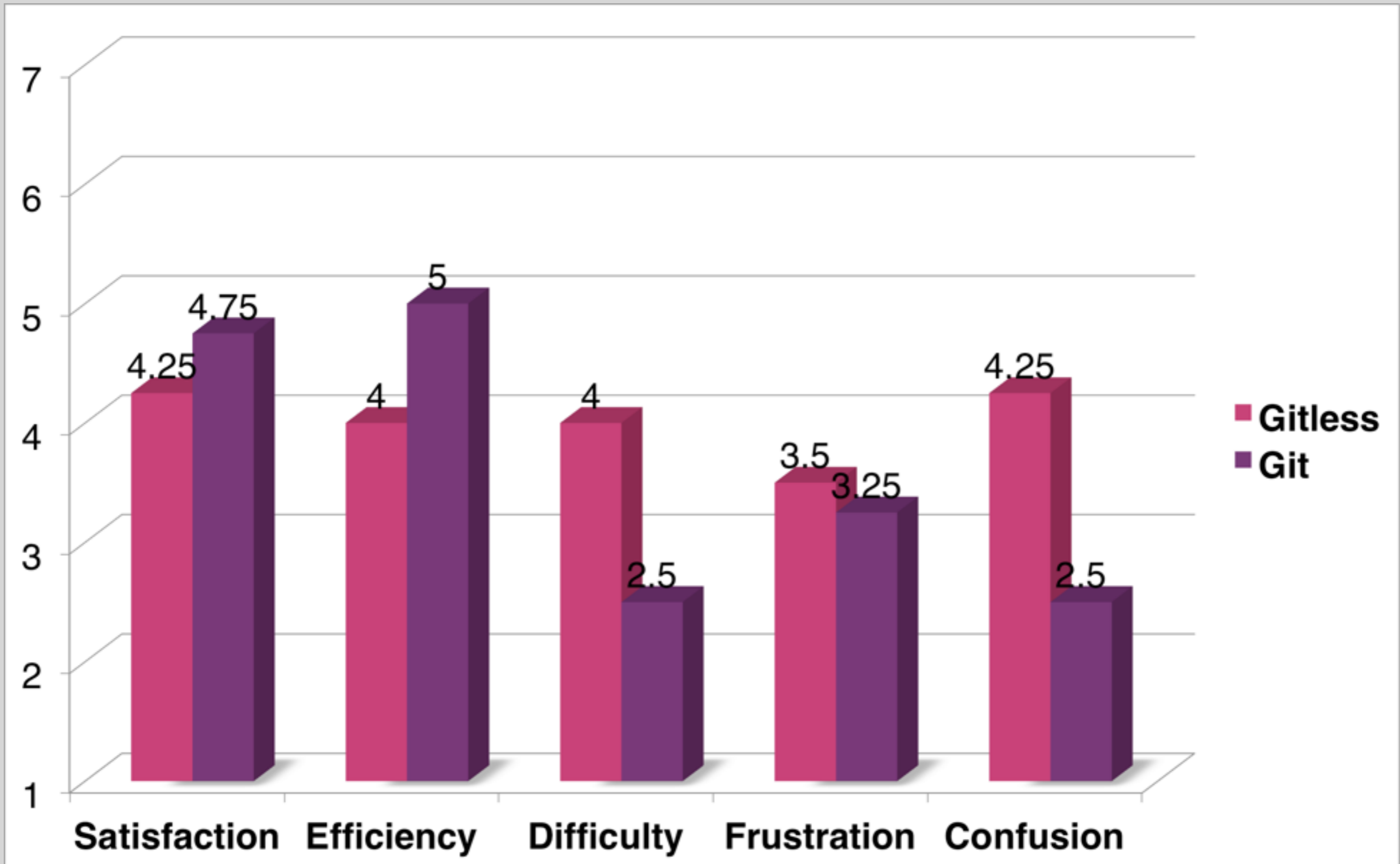
I enjoyed using Gitless
 I found Gitless to be easier to learn than Git
 I found Gitless to be easier to use than Git
 I would continue using Gitless if I could



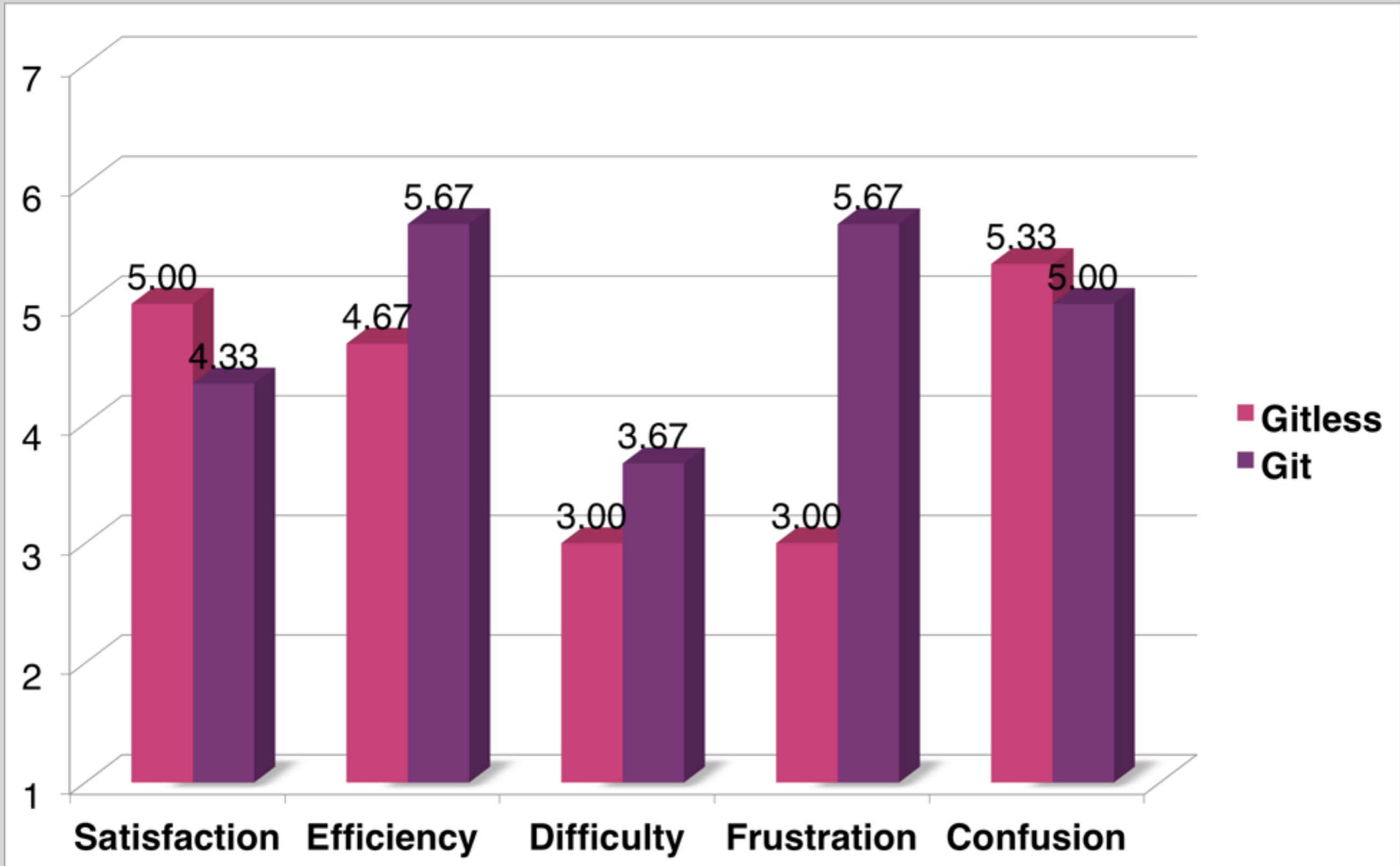
all subjects



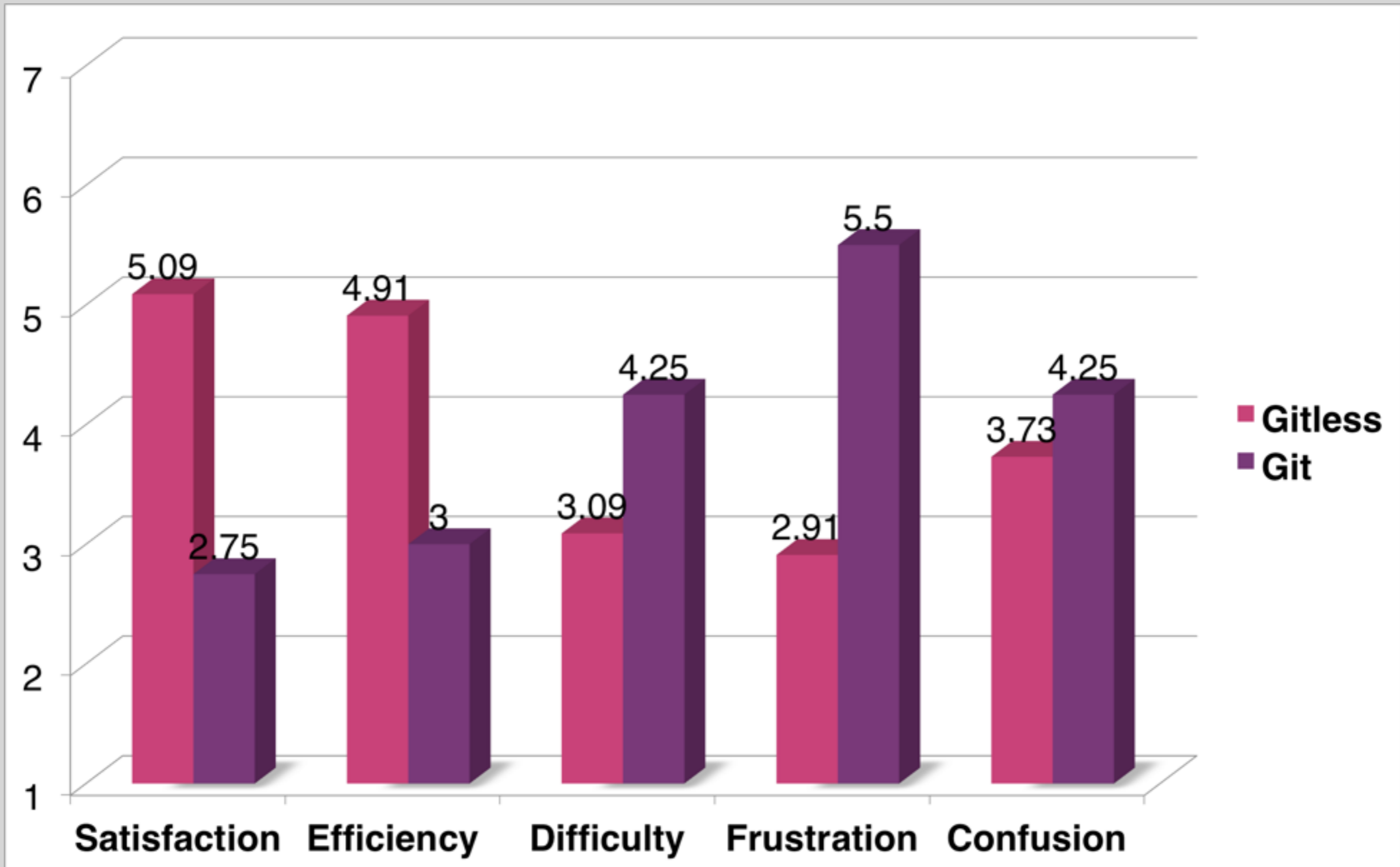
git experts



git regulars



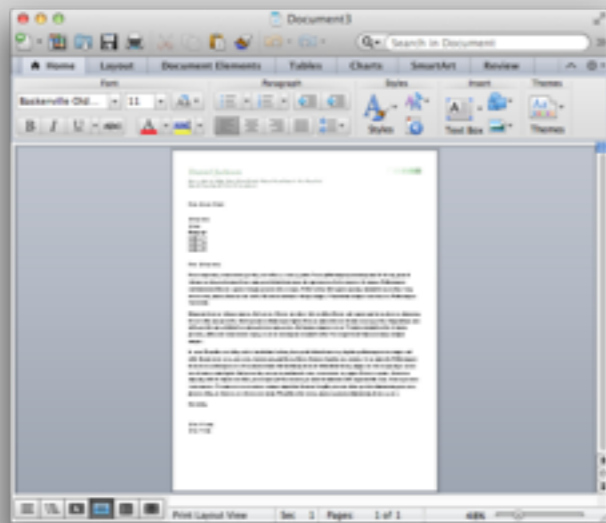
git novices



conclusion

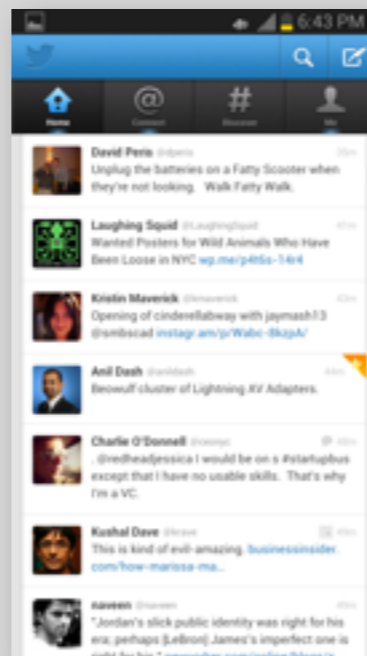
concepts: the key inventions

Microsoft Word



Paragraph
Format
Style

Twitter



Tweet
Hashtag
Following

Photoshop

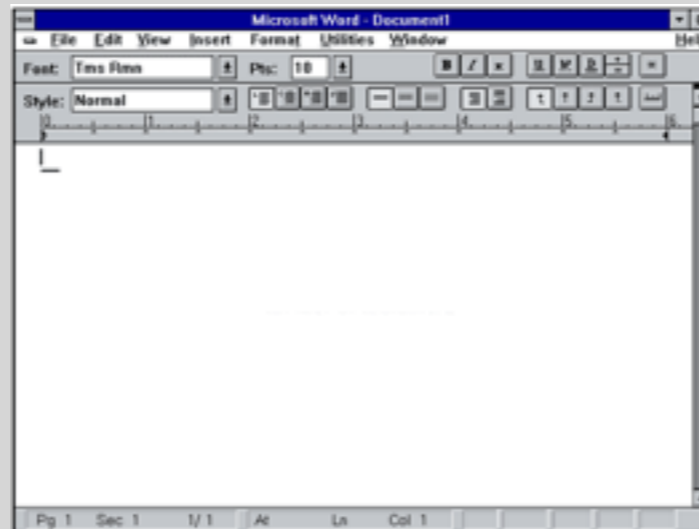


PixelMap
Layer/Mask
Adjustment

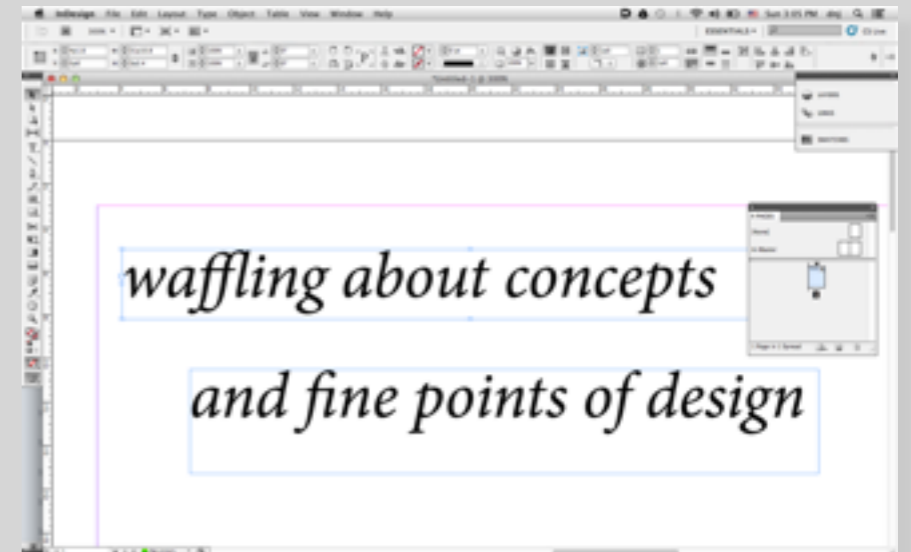
concepts: define app classes



text editor
line
buffer



word processor
paragraph
format
style



desktop publishing app
stylesheet
text flow
page template

jamonh

Oct 22, 2013 7:19 PM

Just upgraded to the new Pages and can't find a way to link text boxes anymore like

<http://www.macobserver.com/tmo/article/pages-linking-text-boxes>

Am I missing something, or is it really not possible anymore?

concept catalog (so far)

<i>instantiate</i>	<i>organize</i>	<i>relate</i>	<i>resource</i>	<i>save</i>	<i>communicate</i>	<i>personalize</i>
stylesheet	selection	friend	access token	history	message	account
master	folder	clique	notification	buffer	posting	karma
stencil	group	invitation	reservation	cursor		OOBA
style buffer	label		REST	sync		rating
	layer		cart	export		status
	stack		subscription			
	alias		purchase order			
	preset		RMA			
	cursor		coupon			
	filter		catalog			
	property		ticket			
	metadata					

a common view of software design

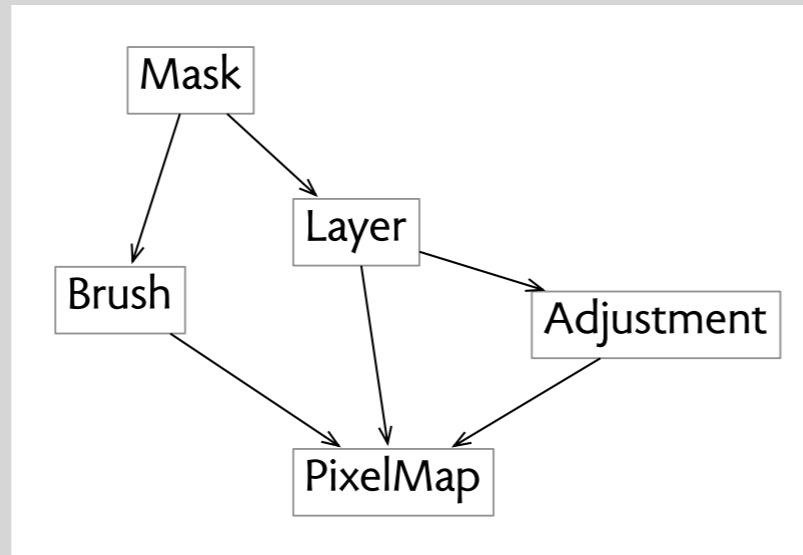


UI design
soft & human
about presentation

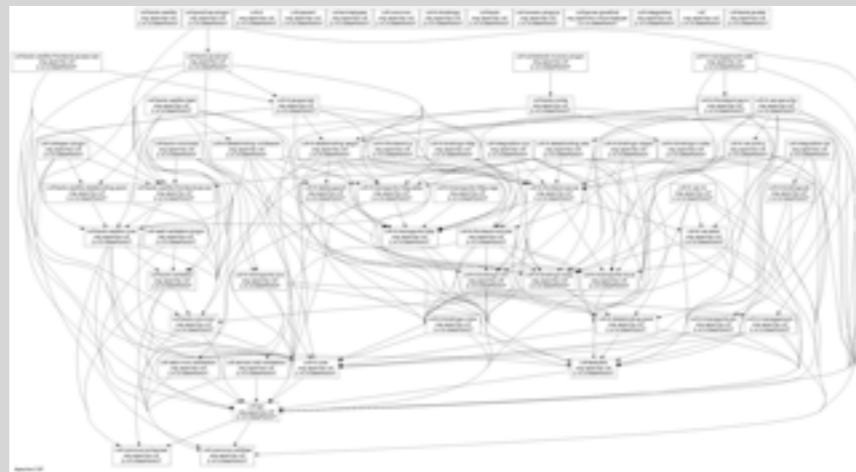


programming
hard & technical
about content

a better view of software design



conceptual design:
essential concepts
& behavior



representation design:
organization & performance