



“Fluffy Bunny”

The story of refactoring Digital Trends



In the beginning...



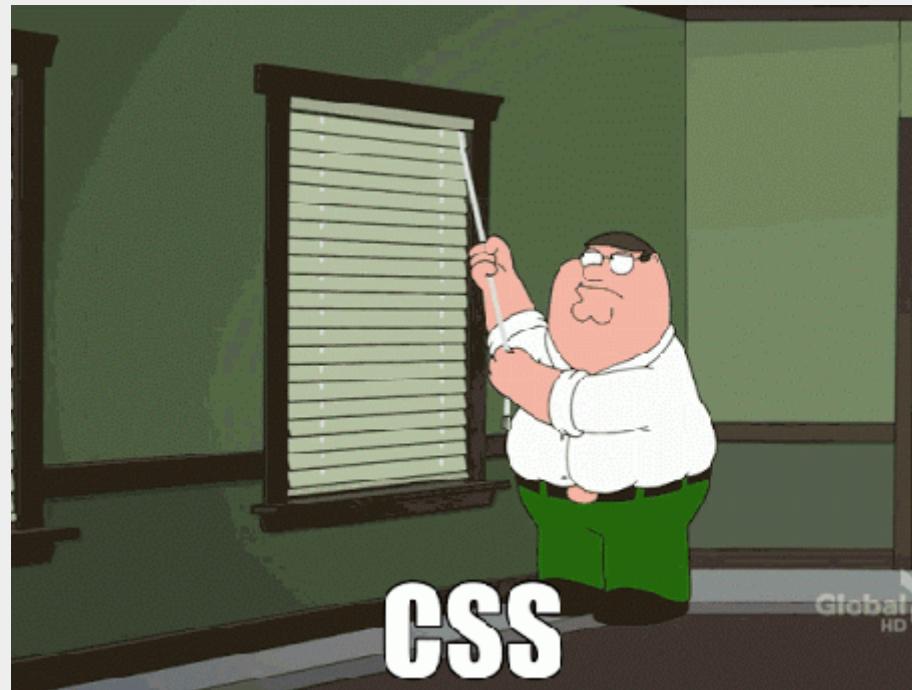
There was the legacy codebase



m.digitaltrends.com: *The bastard no one talks about*



There was the legacy codebase



CSS Bleed and a Rigid DOM



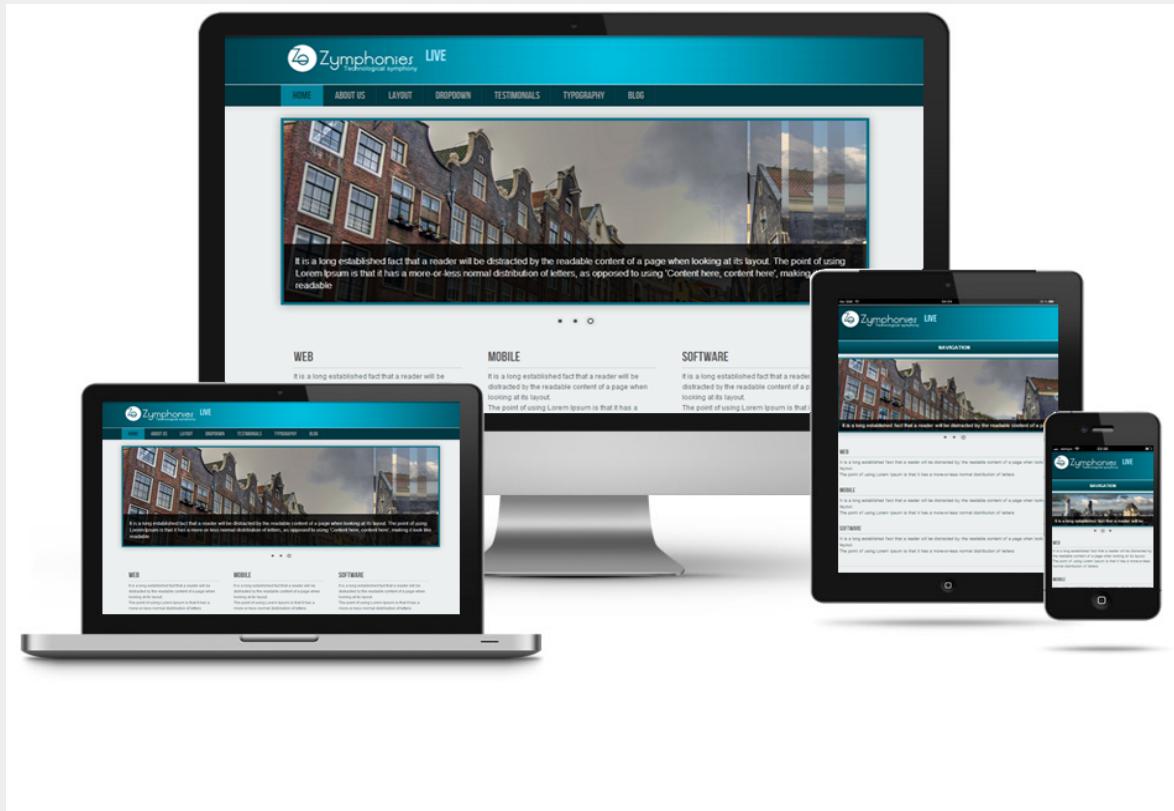
There was the legacy codebase



A review system written for a different time



There was a developer need



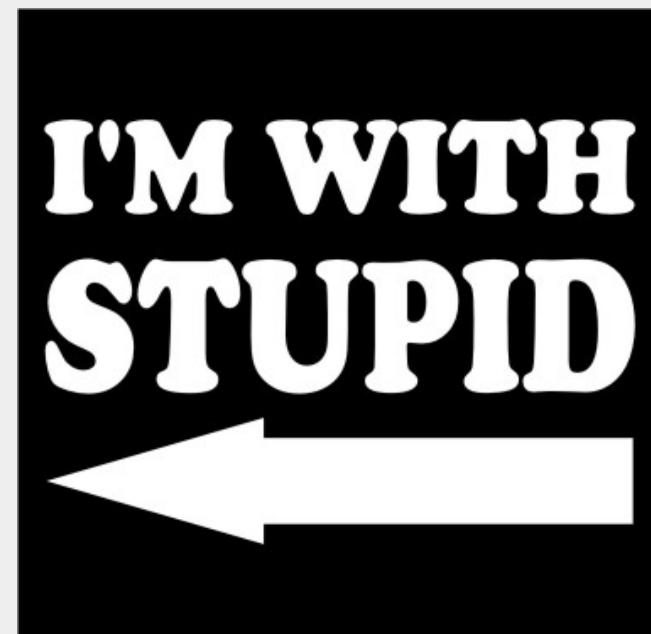
Write once, display everywhere



There was a developer need



Plugins



Theme

Scope/Dependency Clean-up



There was a developer need



Modularity

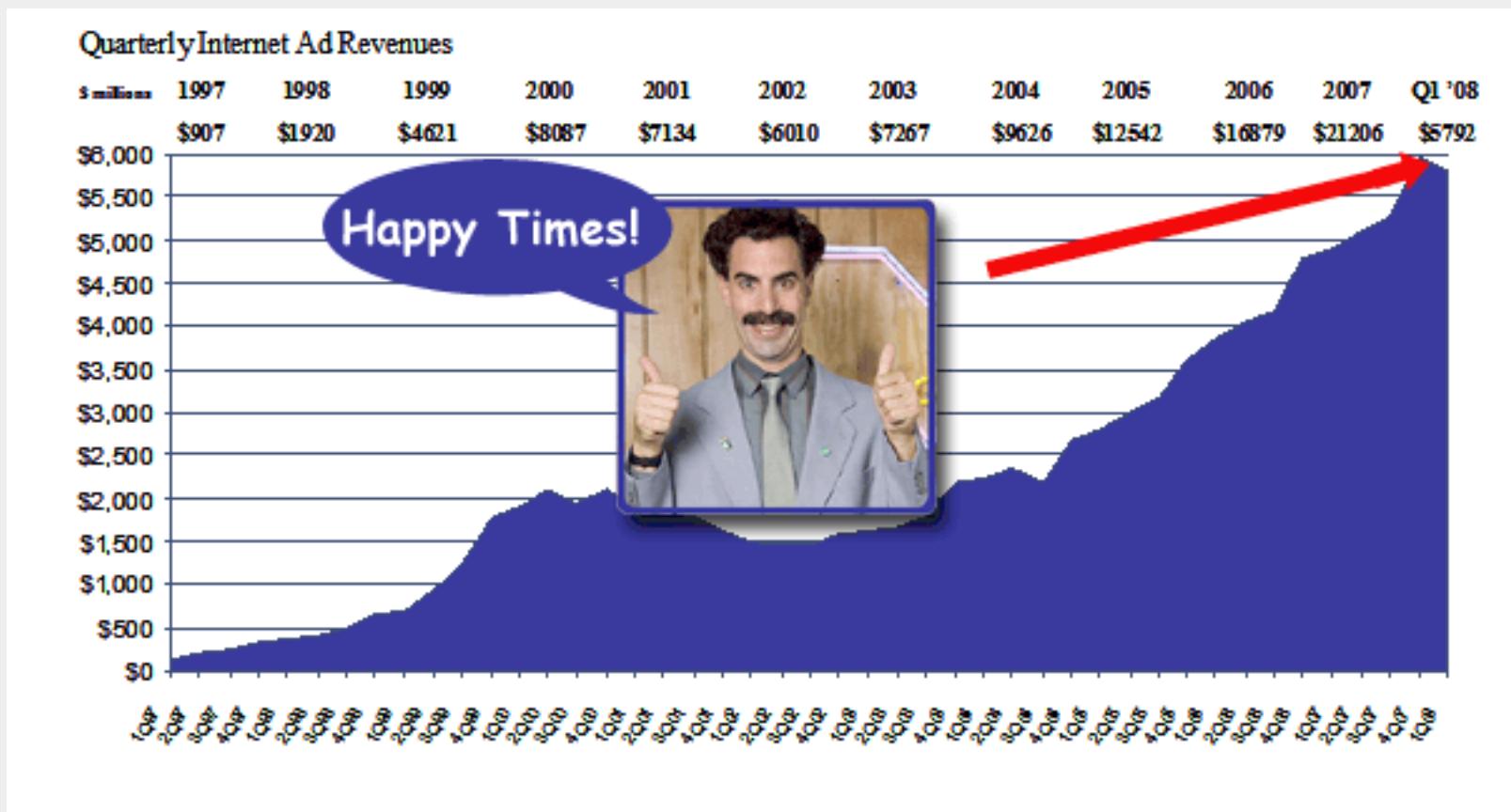
There was a developer need



Code ownership



There was a business need





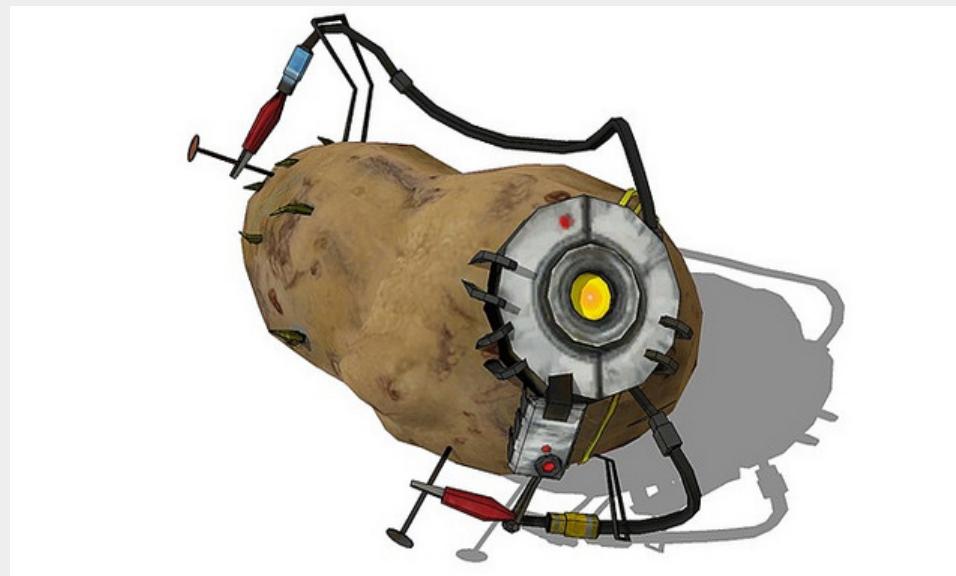
There was a business need



Time on site (Project Falcor)



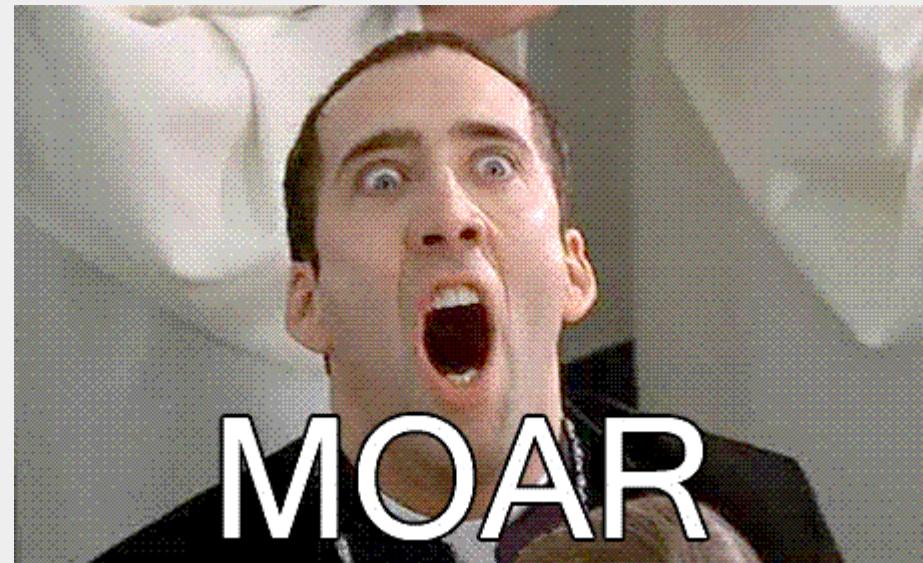
There was a business need



Accessibility



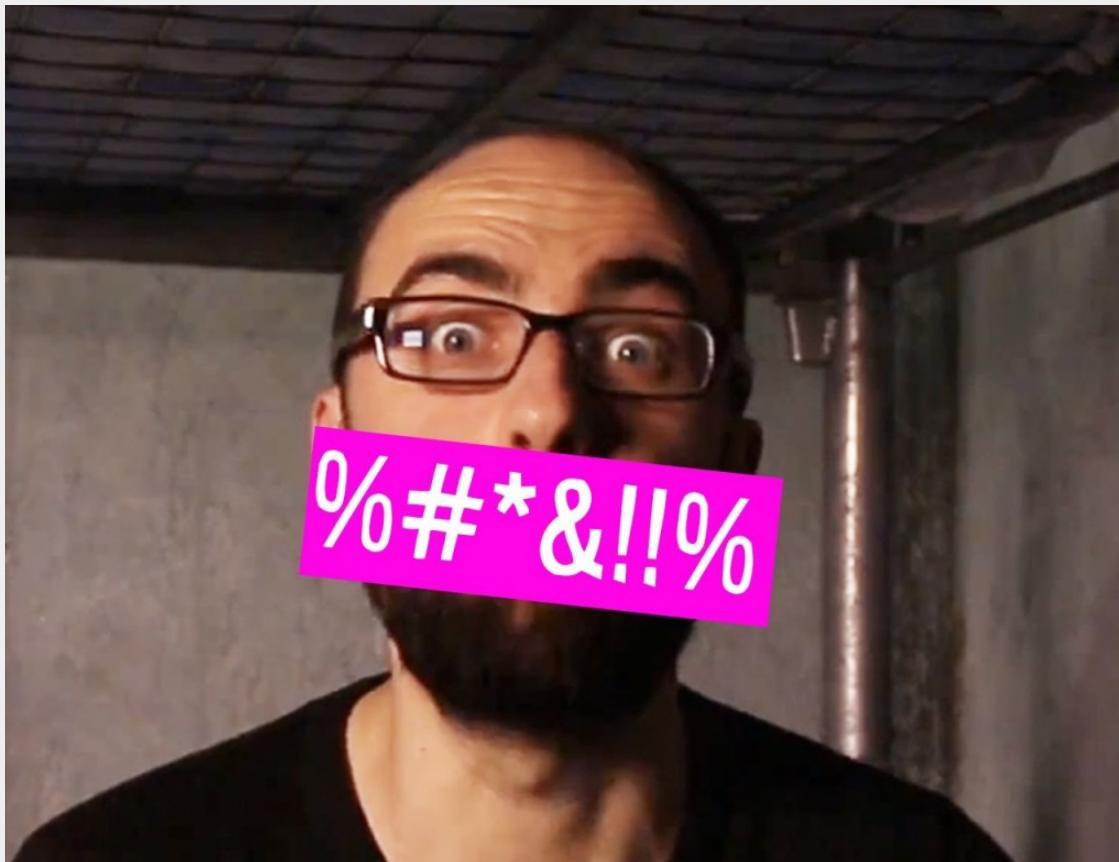
There was a business need



More new features, quicker



There was a hesitant management



“Responsive”



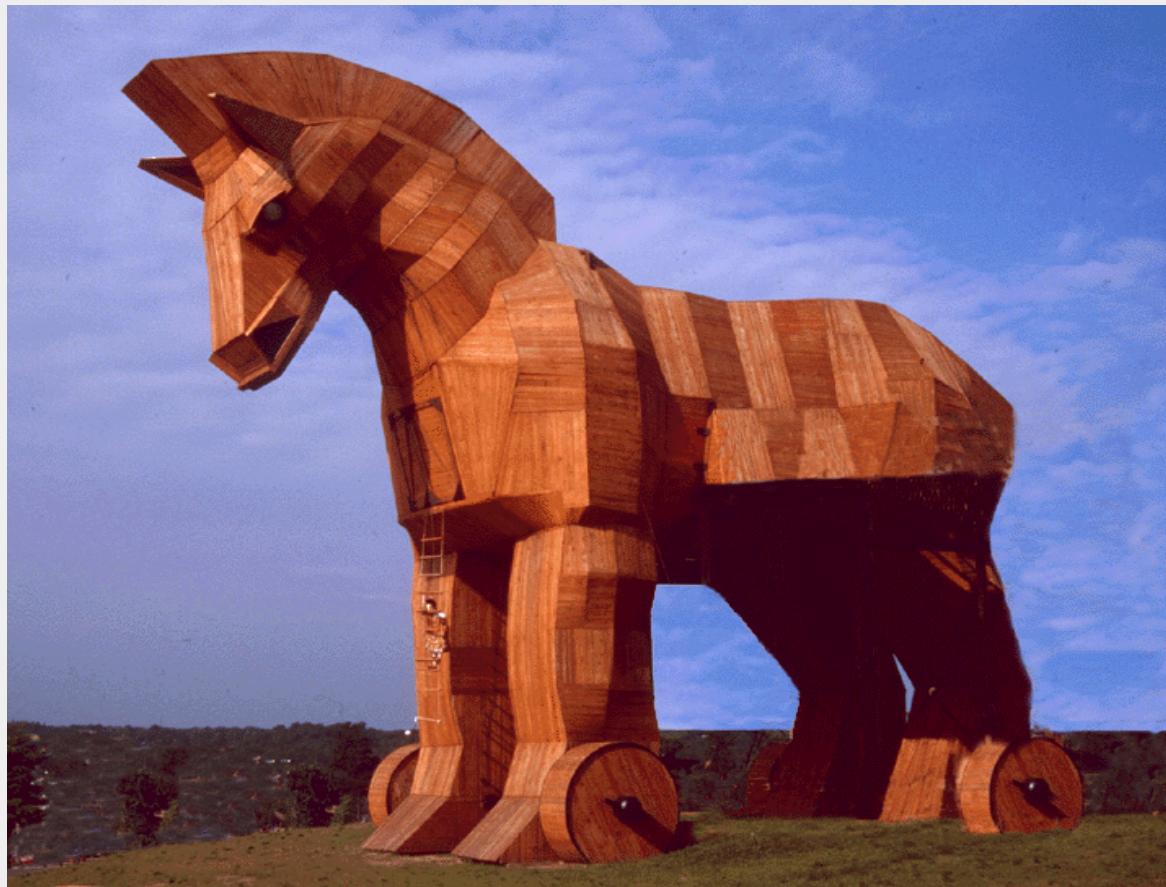
There was a hesitant management



Responsive? That is the sound of inevitability



There was a hesitant management



Adaptive: Responsive in disguise



There was a hesitant management



Project Fluffy Bunny (AKA: Unified Theme)



There and back again



Two projects for the price of two



The Kraken: Because.. why not.



Managing the code



The function spreadsheet



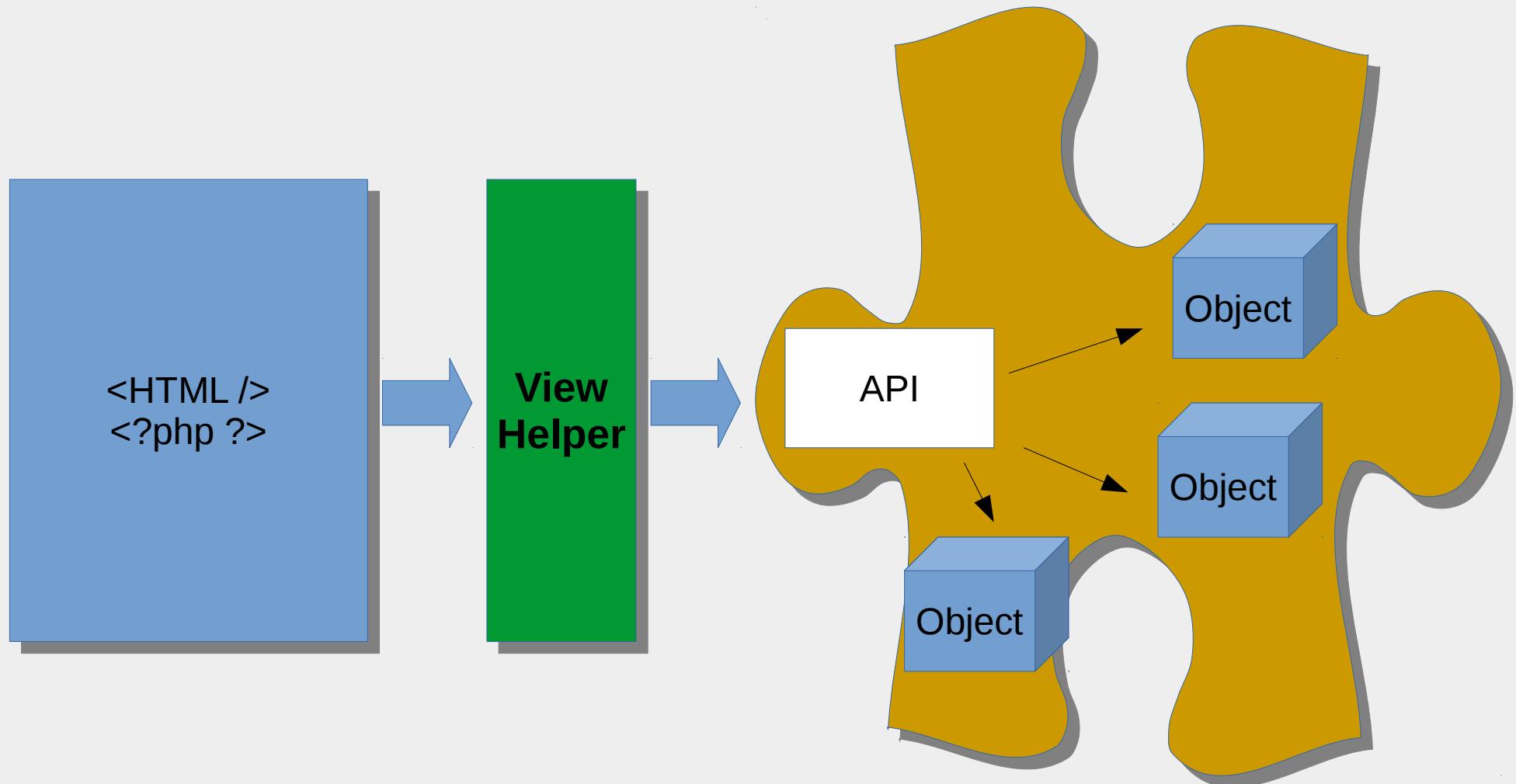
Managing the code



New methodology, new objects



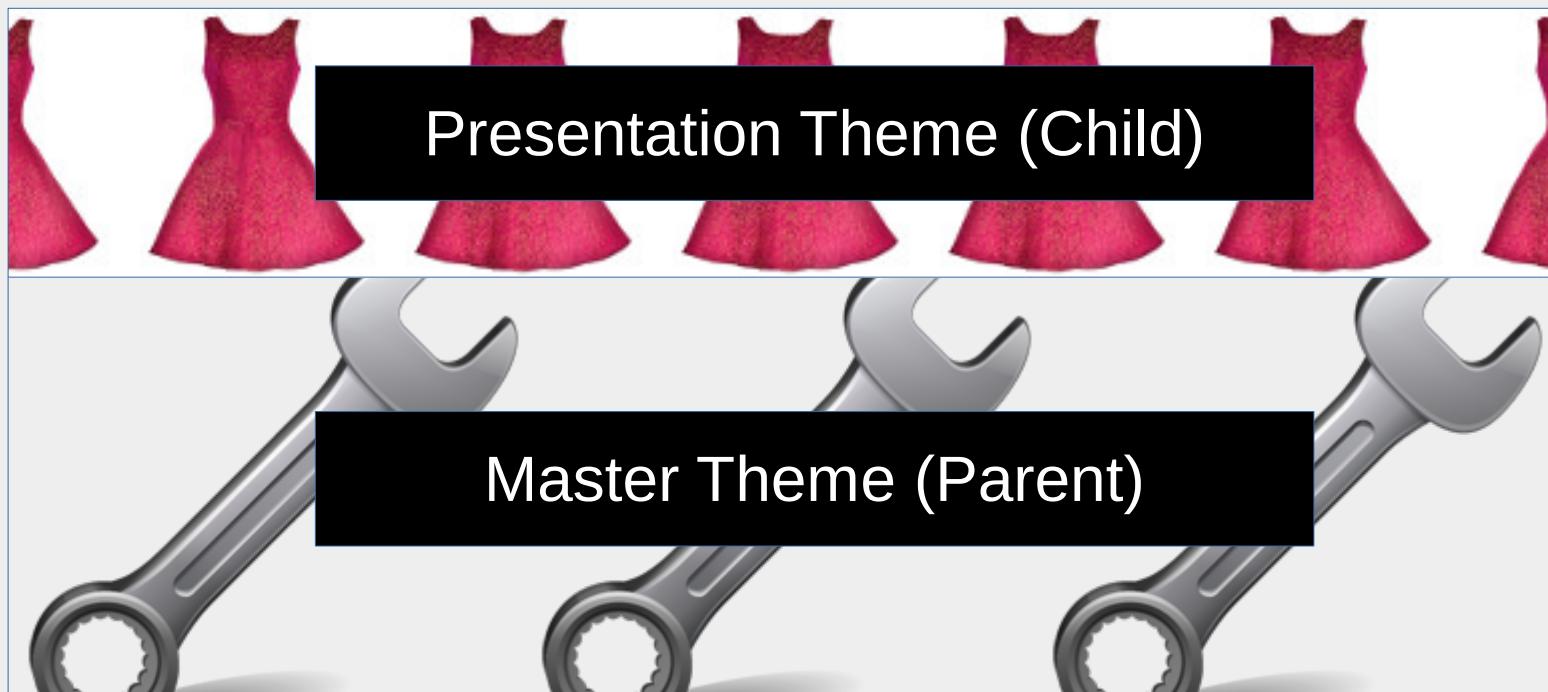
Managing the code



Module APIs



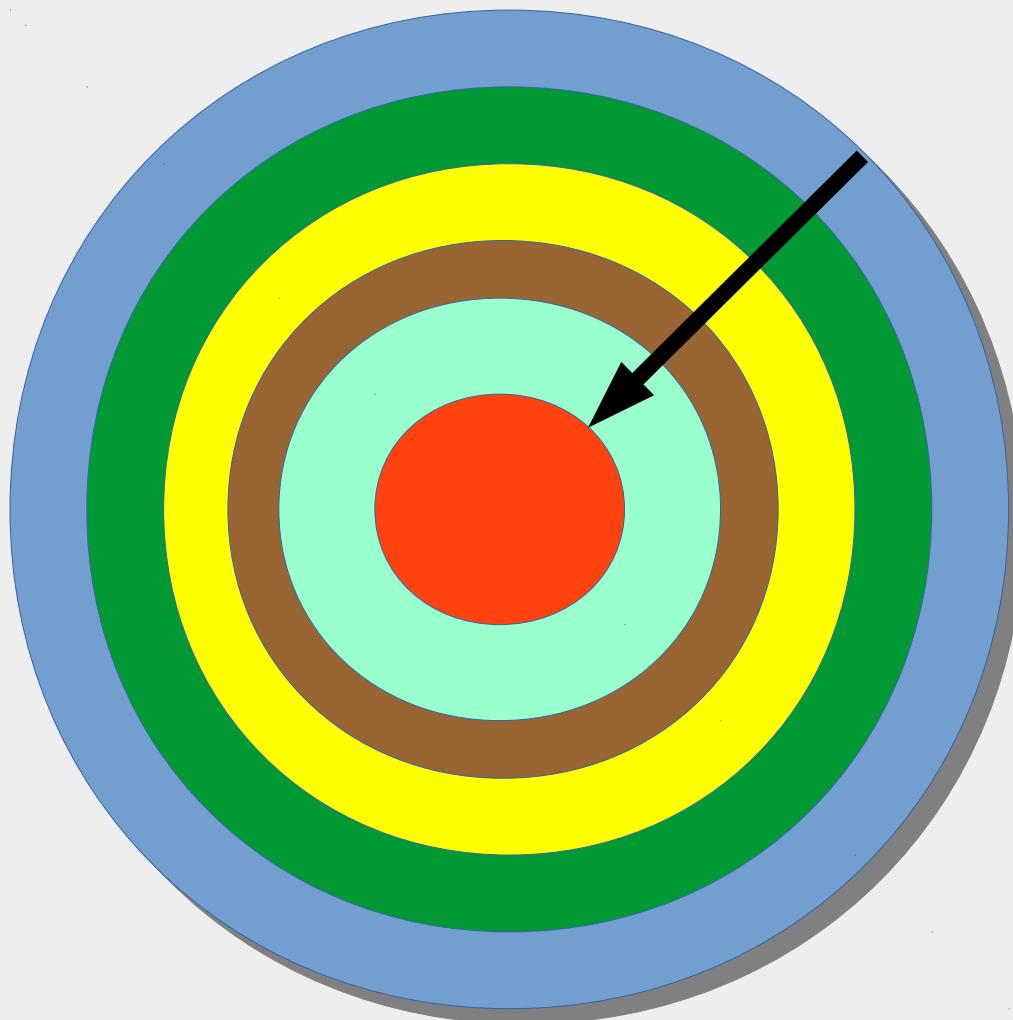
Managing the code



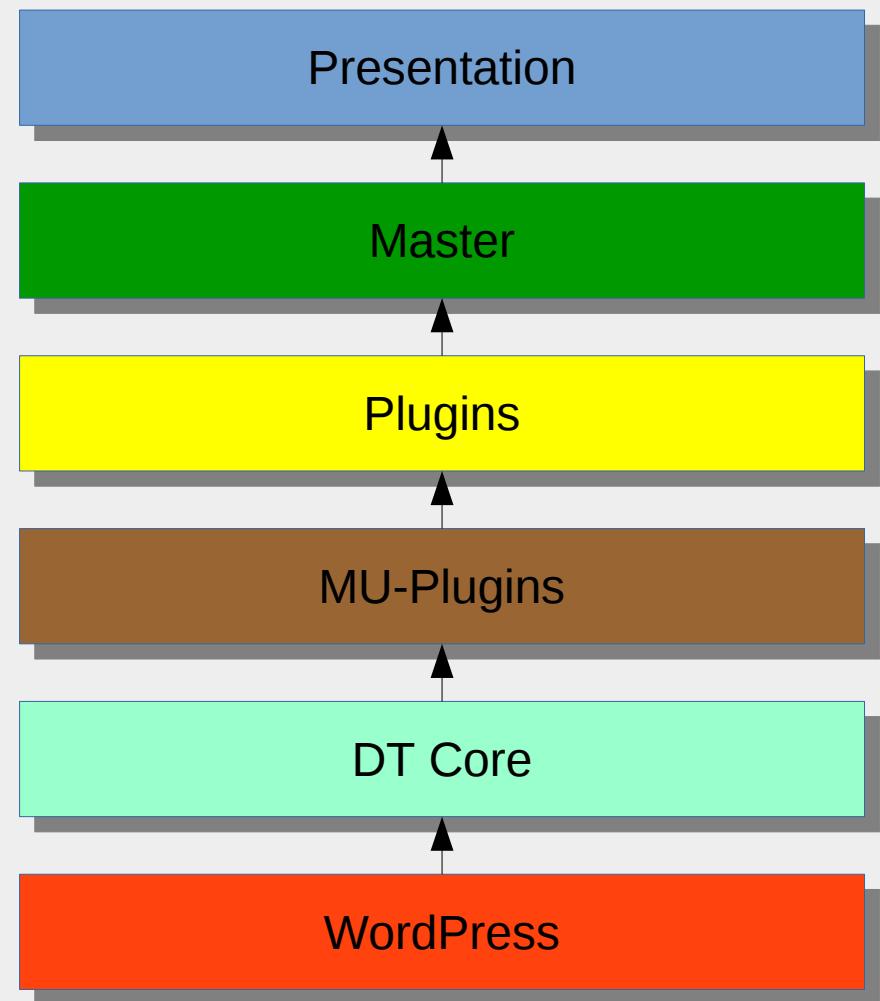
Splitting the theme into function and form



Managing the code

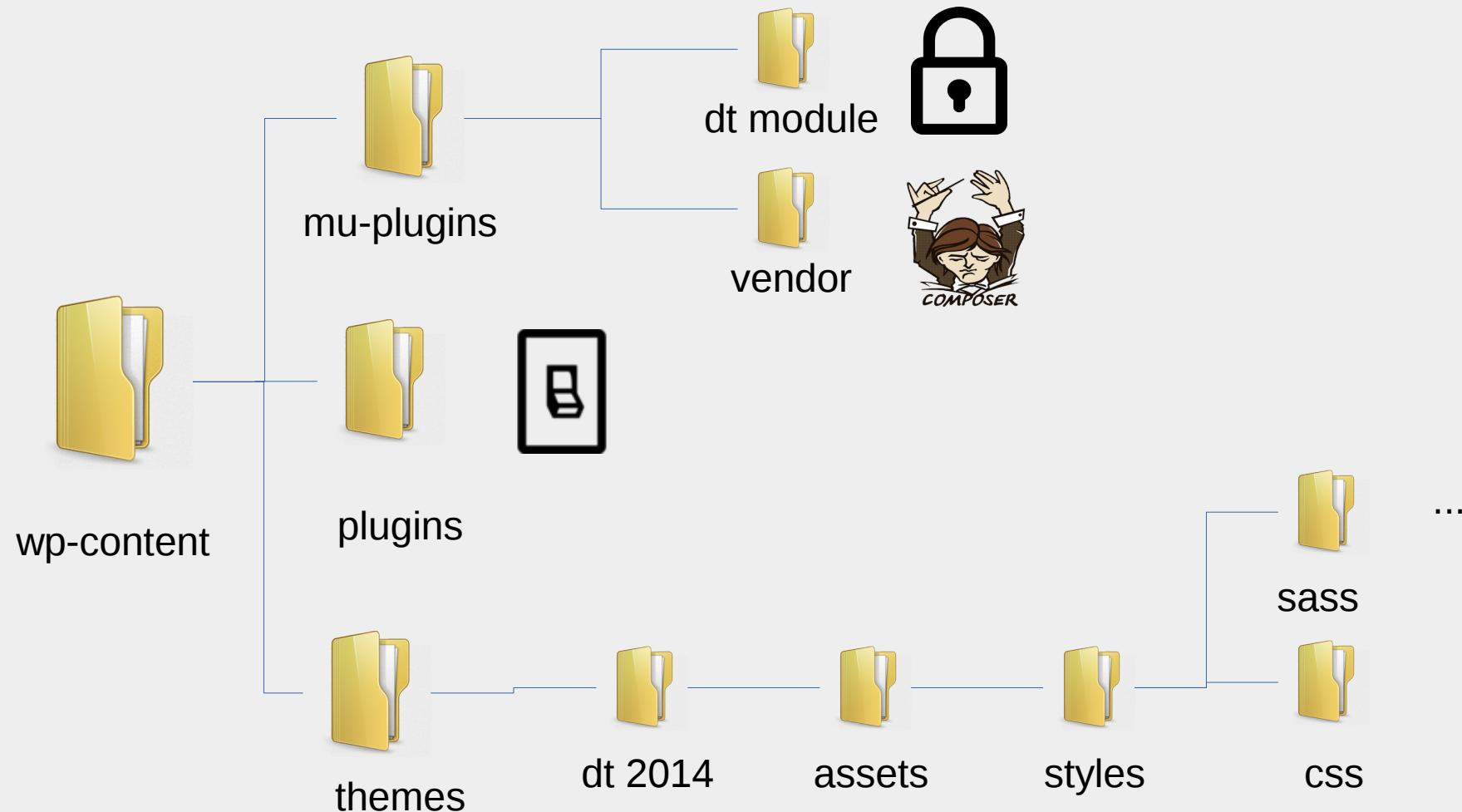


Layers of Scope





Managing the code



Code organization



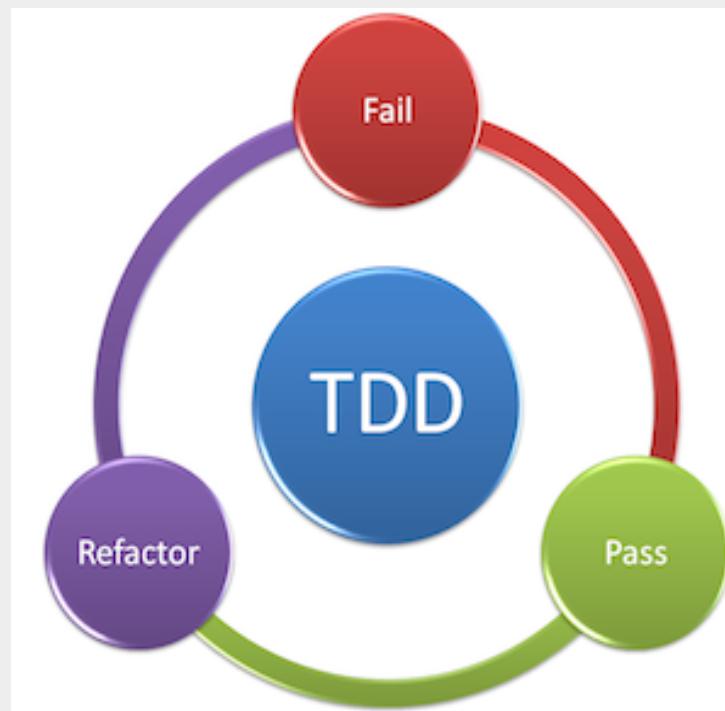
Managing the code



Javascript event handling



Managing the code



Test Driven Development



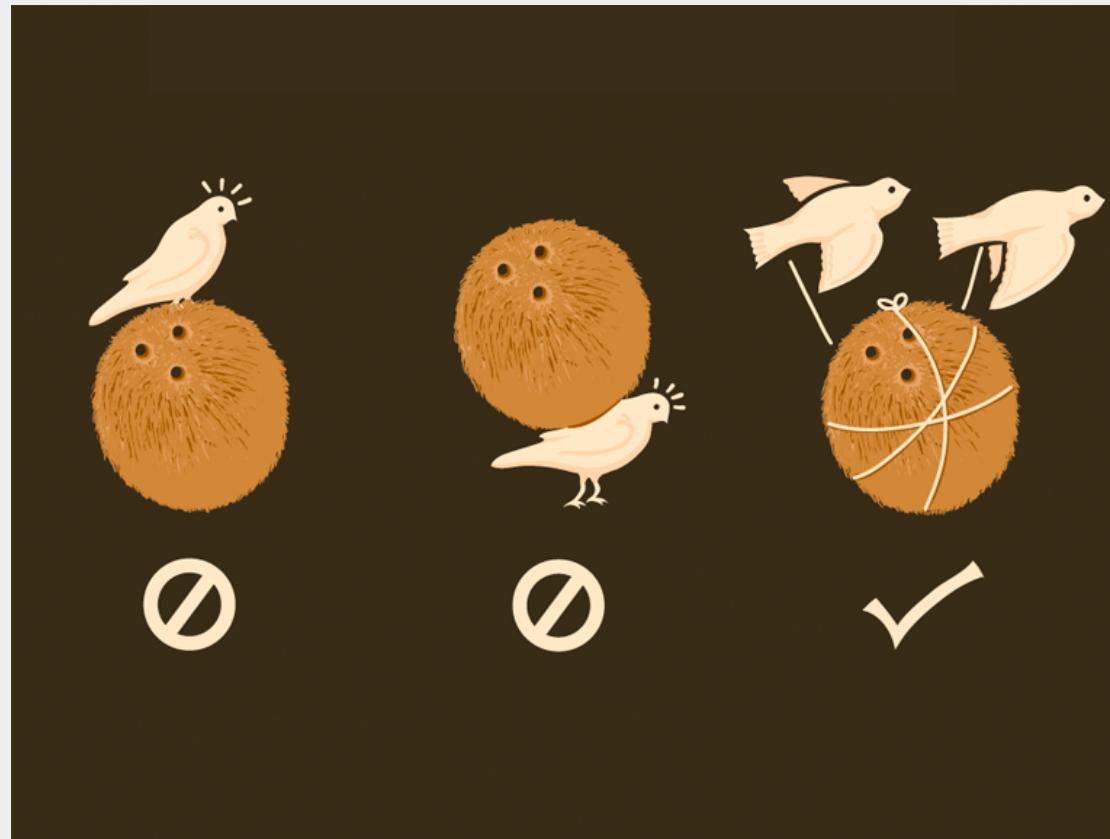
Managing the data



Product Specifications. A lot of them.



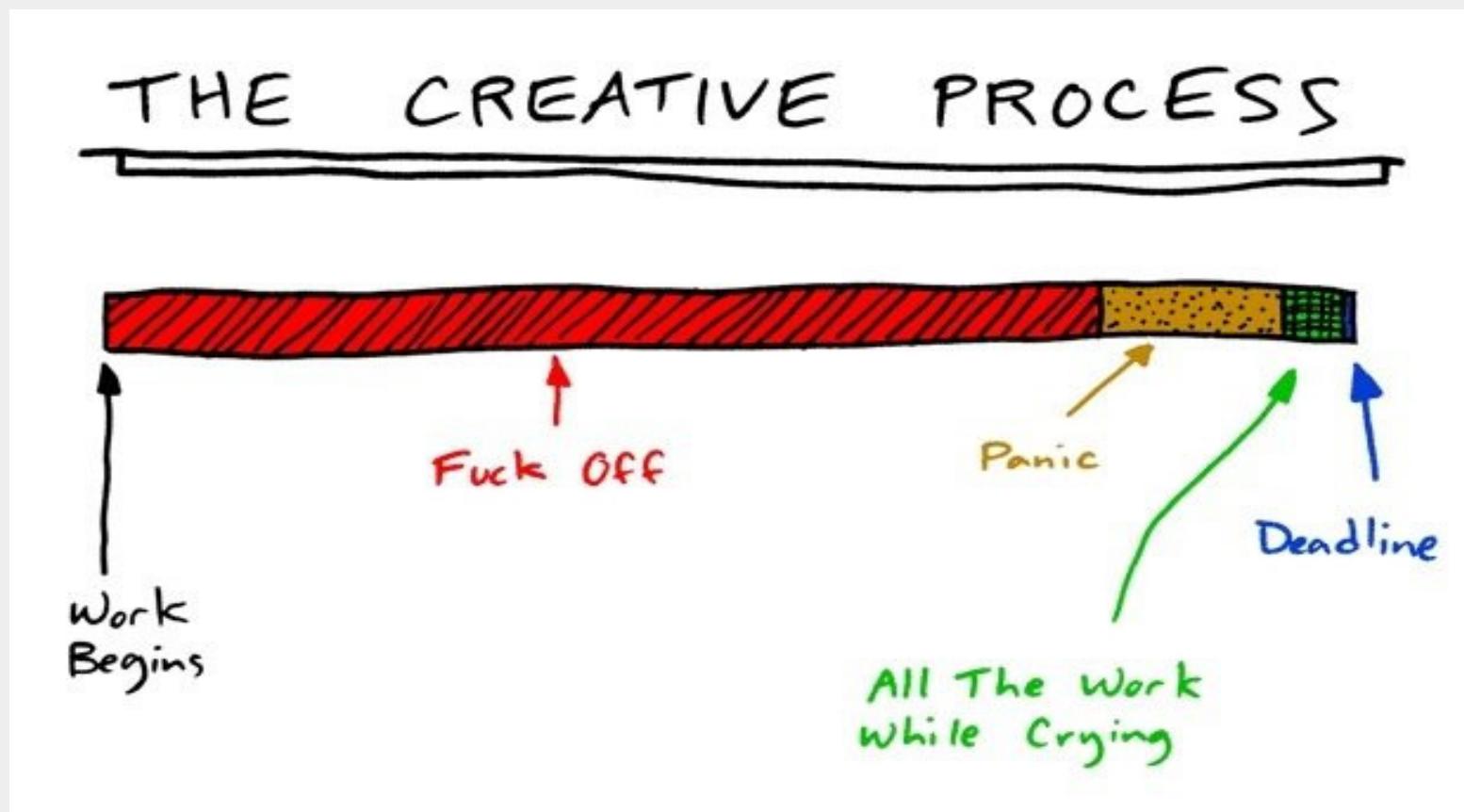
Managing the data



Data migration



Managing the people



Managing the time line

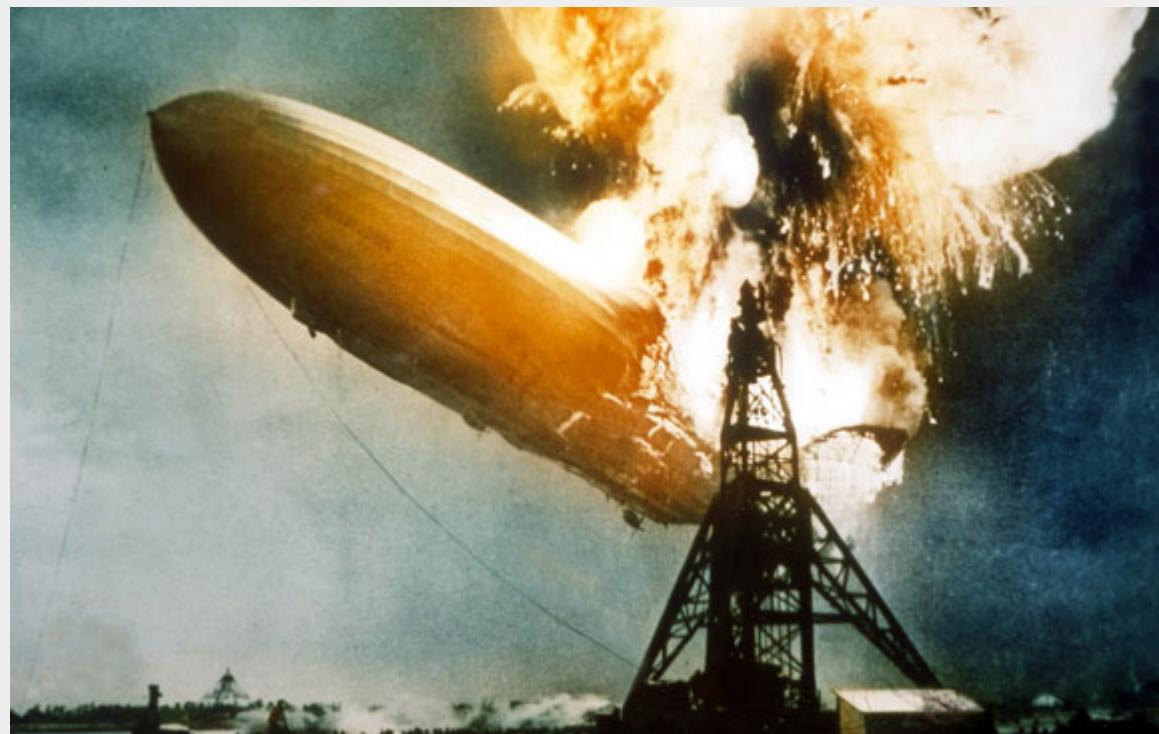
Managing the people



Managing expectations



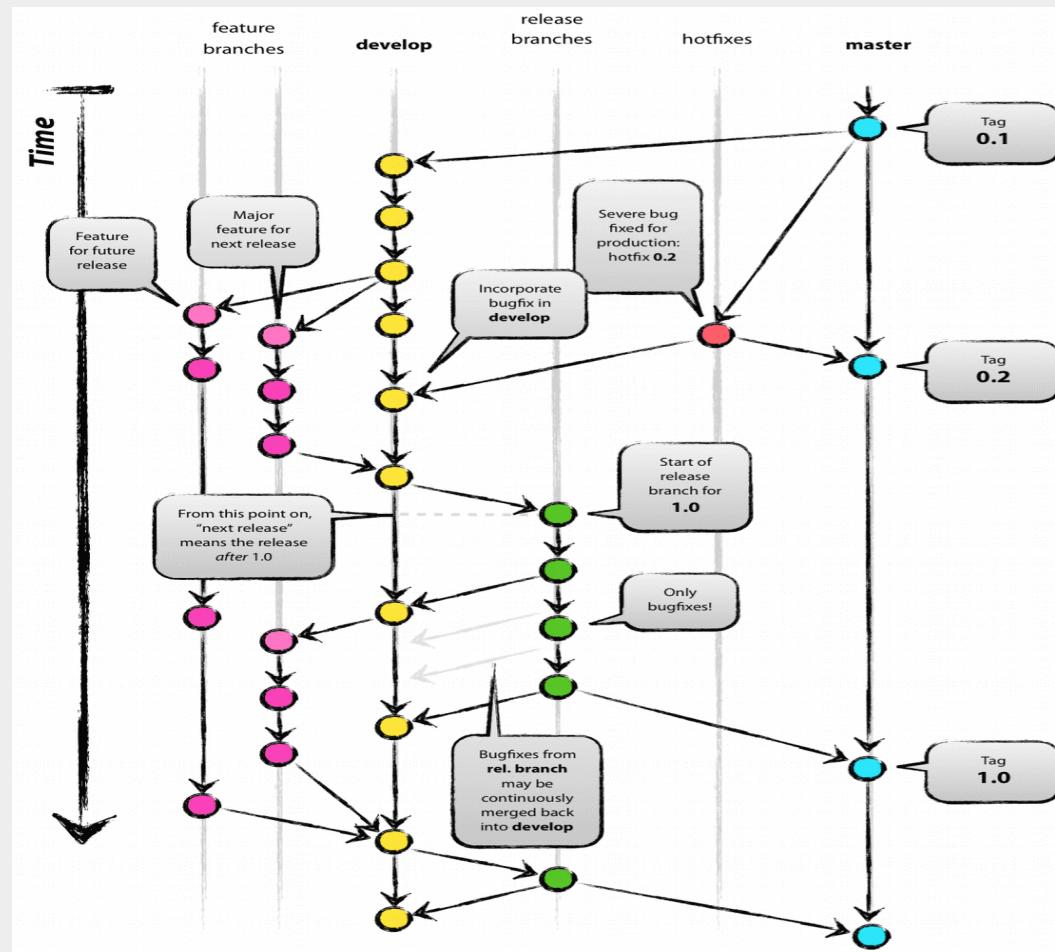
Managing the people



The technology variable



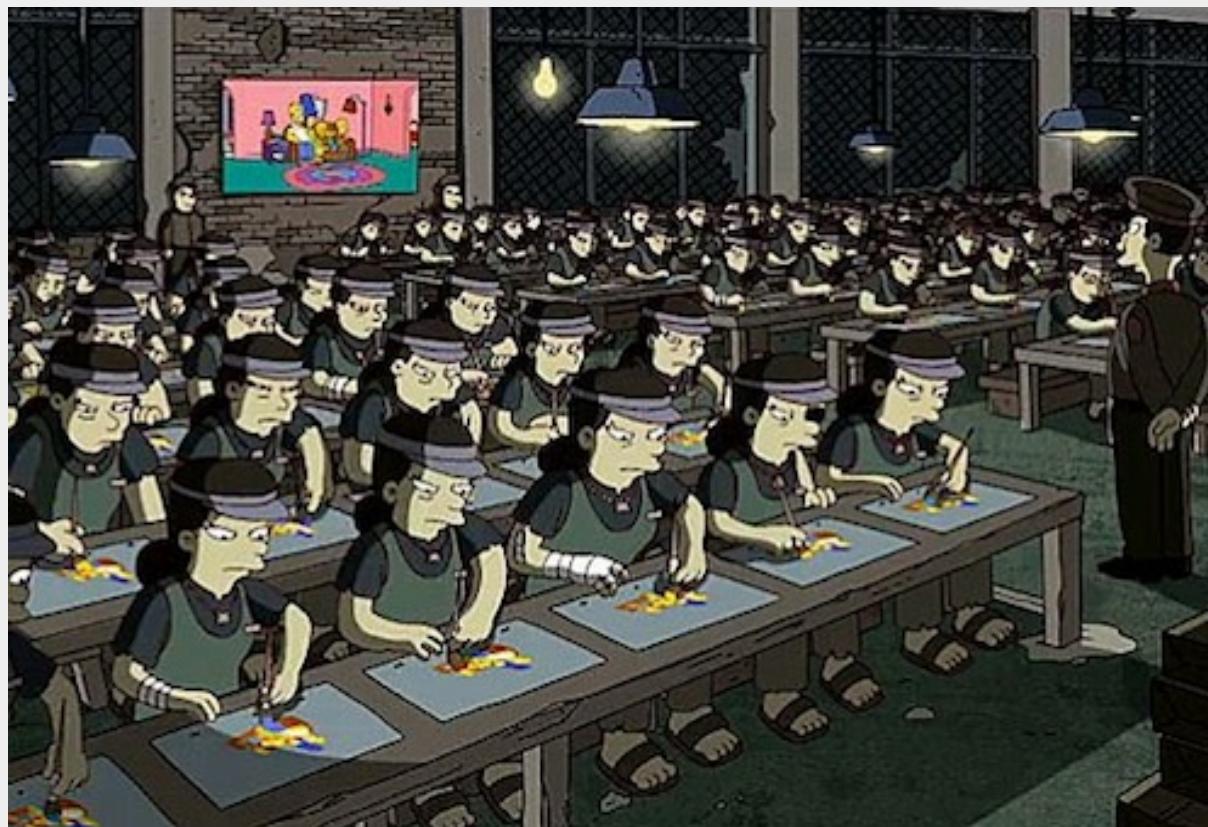
Managing the workflow



Git collaboration



Managing the workflow



Creative outsourcing



Managing the quality



QA Team? Lol



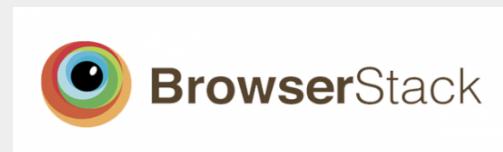
Managing the quality



A process for feedback



Managing the quality



Staging servers that mirror production



Managing the deployment



Setting a date



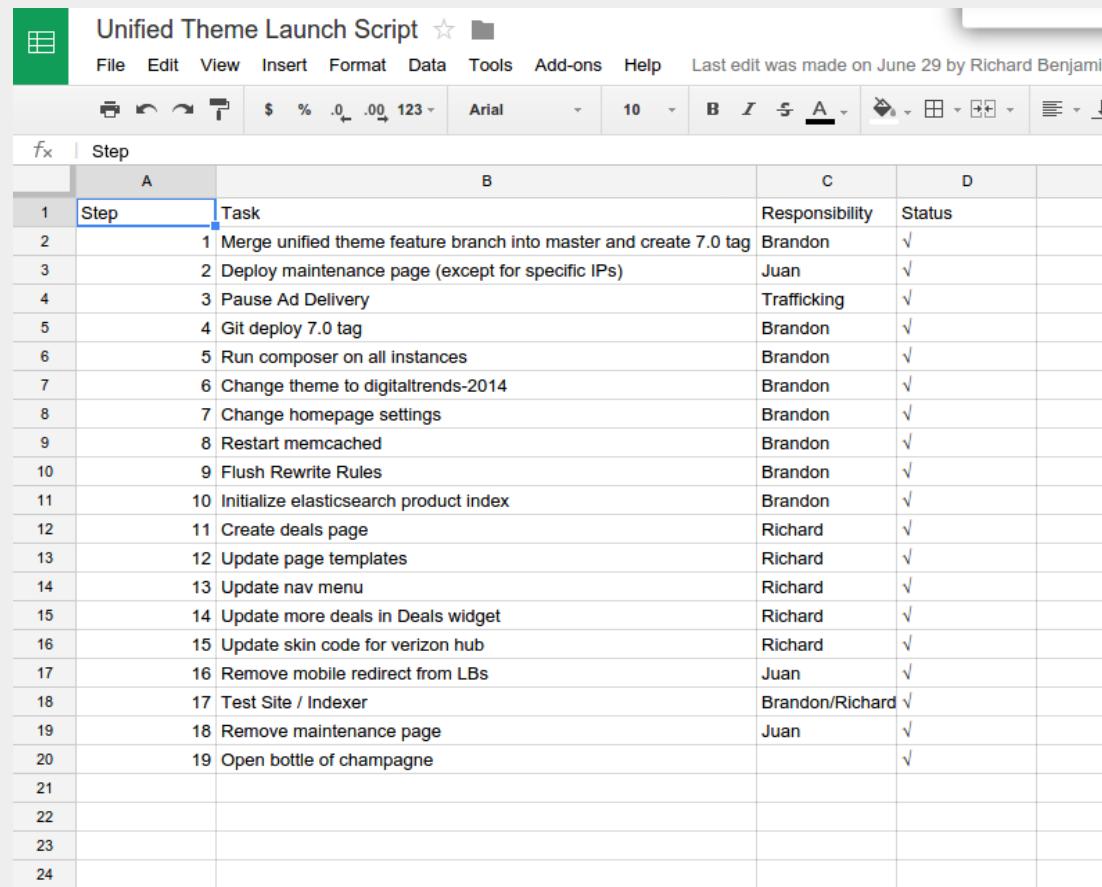
Managing the deployment



Coordinating across depts (ie:Herding the cats)



Managing the deployment



The screenshot shows a Google Sheets document with the title "Unified Theme Launch Script". The spreadsheet has a single sheet named "Step" and contains five columns: A, B, C, D, and E. Column A is labeled "Step" and contains numbers 1 through 24. Column B is labeled "Task" and lists 19 specific tasks. Column C is labeled "Responsibility" and lists names: Brandon, Juan, Trafficking, Brandon, Richard, Richard, Richard, Juan, Brandon/Richard, Juan, and Juan. Column D is labeled "Status" and contains checkmarks (✓) for most tasks. Column E is empty. The document also includes a standard Google Sheets toolbar at the top.

Step	Task	Responsibility	Status	
1	Step			
2	1 Merge unified theme feature branch into master and create 7.0 tag	Brandon	✓	
3	2 Deploy maintenance page (except for specific IPs)	Juan	✓	
4	3 Pause Ad Delivery	Trafficking	✓	
5	4 Git deploy 7.0 tag	Brandon	✓	
6	5 Run composer on all instances	Brandon	✓	
7	6 Change theme to digitaltrends-2014	Brandon	✓	
8	7 Change homepage settings	Brandon	✓	
9	8 Restart memcached	Brandon	✓	
10	9 Flush Rewrite Rules	Brandon	✓	
11	10 Initialize elasticsearch product index	Brandon	✓	
12	11 Create deals page	Richard	✓	
13	12 Update page templates	Richard	✓	
14	13 Update nav menu	Richard	✓	
15	14 Update more deals in Deals widget	Richard	✓	
16	15 Update skin code for verizon hub	Richard	✓	
17	16 Remove mobile redirect from LBs	Juan	✓	
18	17 Test Site / Indexer	Brandon/Richard	✓	
19	18 Remove maintenance page	Juan	✓	
20	19 Open bottle of champagne		✓	
21				
22				
23				
24				

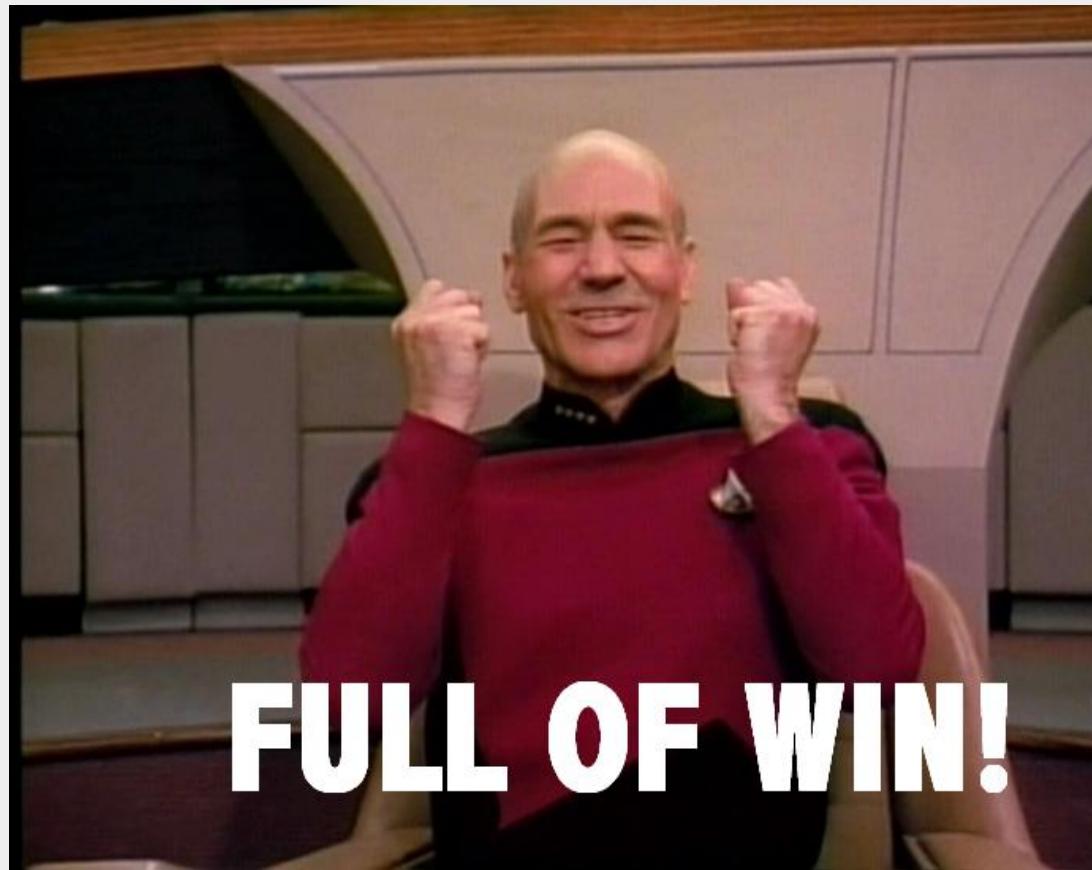
Writing a launch script (not the bash kind)



Win, Lose, and Draw...



WIN



What worked



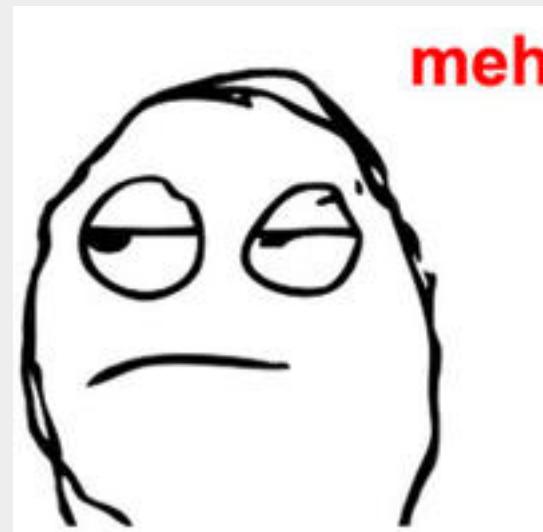
LOSE



What fell short.



DRAW



Well, it didn't get worse at least



Next year (2015)



FIN.



<http://www.digitaltrends.com>



<http://www.twitter.com/digitaltrends>



<http://www.facebook.com/digitaltrendsftw>