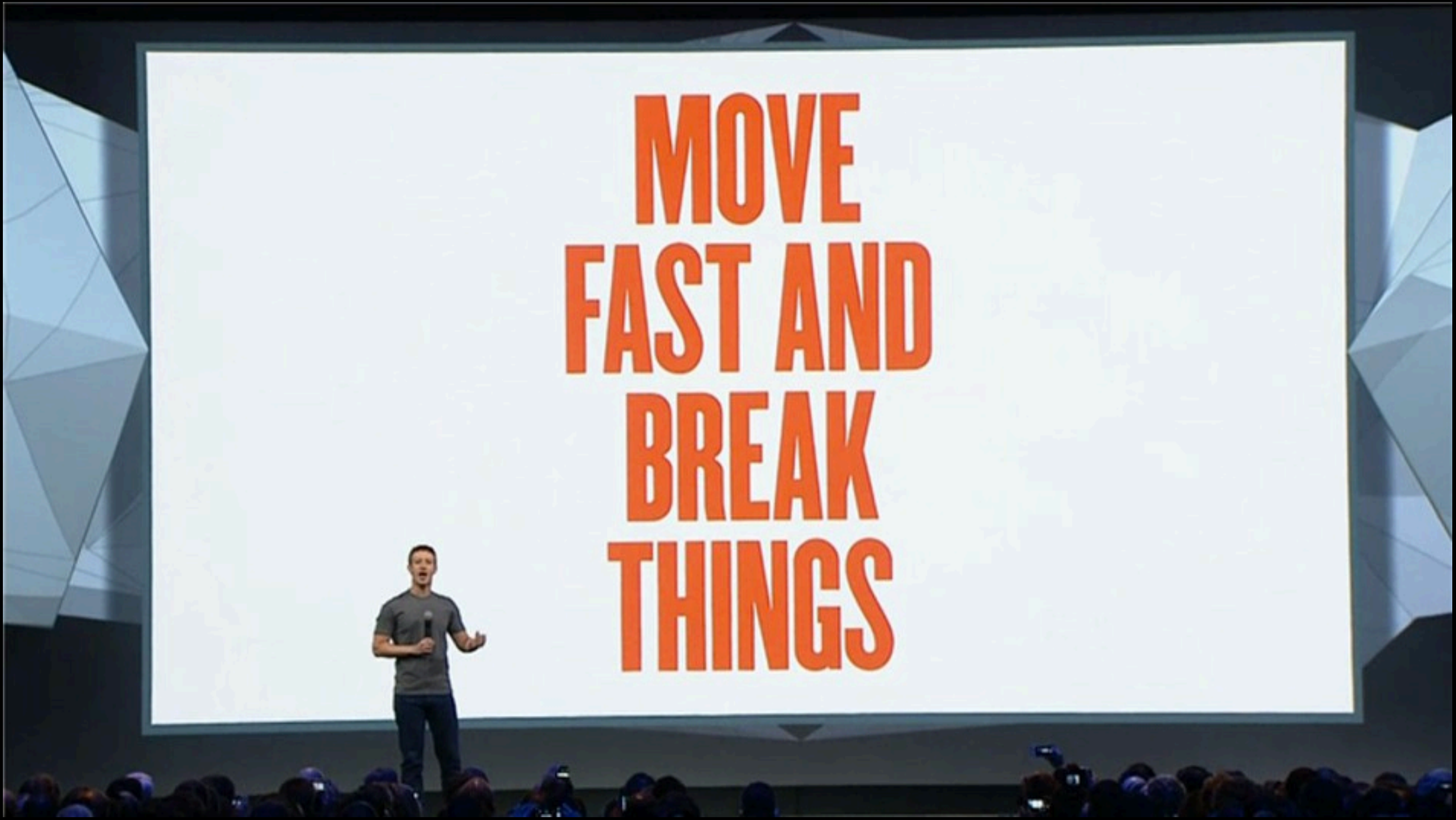
iOS development efficiency at Facebook


MOVE
FAST AND
BREAK
THINGS

Brief history of iOS at Facebook

2011-2016

Best practices for scaling

revision control, branching strategies, development cycle

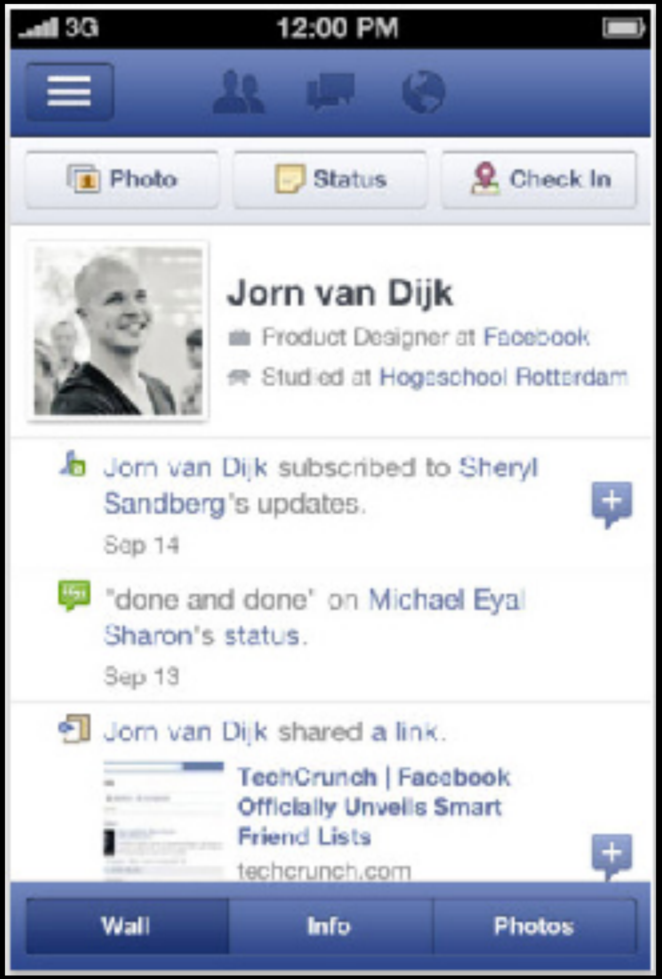
iOS open source tools/frameworks

brief overview

Applying Facebook development efficiency

at Bellabeat

2011
Web company

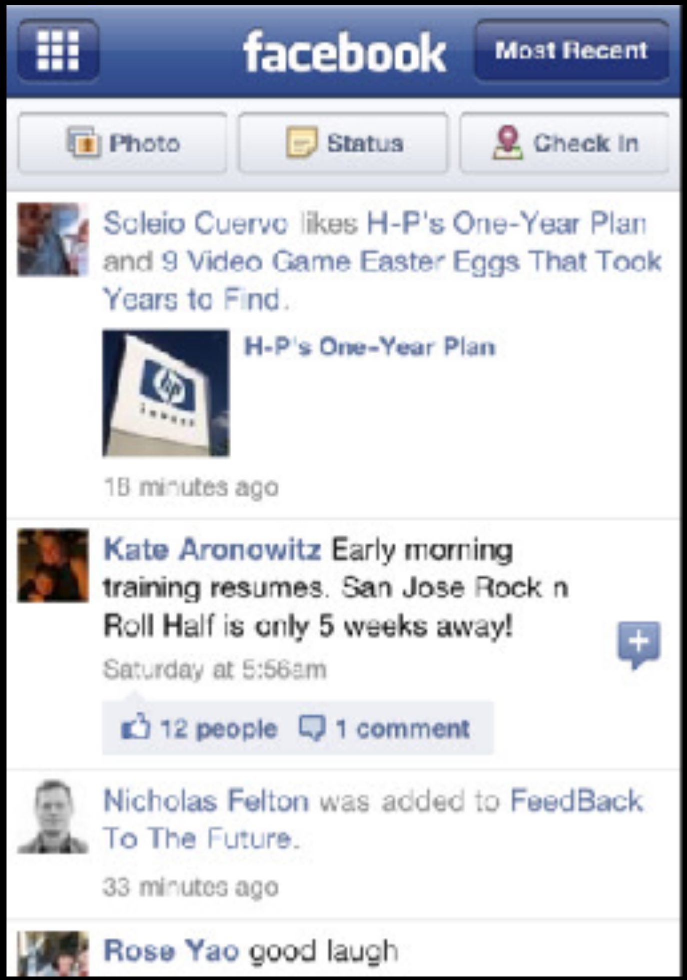


Jorn van Dijk
Product Designer at Facebook
Studied at Hogeschool Rotterdam

Jorn van Dijk subscribed to Sheryl Sandberg's updates.
Sep 14

"done and done" on Michael Eyal Sharon's status.
Sep 13

Jorn van Dijk shared a link.
TechCrunch | Facebook Officially Unveils Smart Friend Lists
techcrunch.com

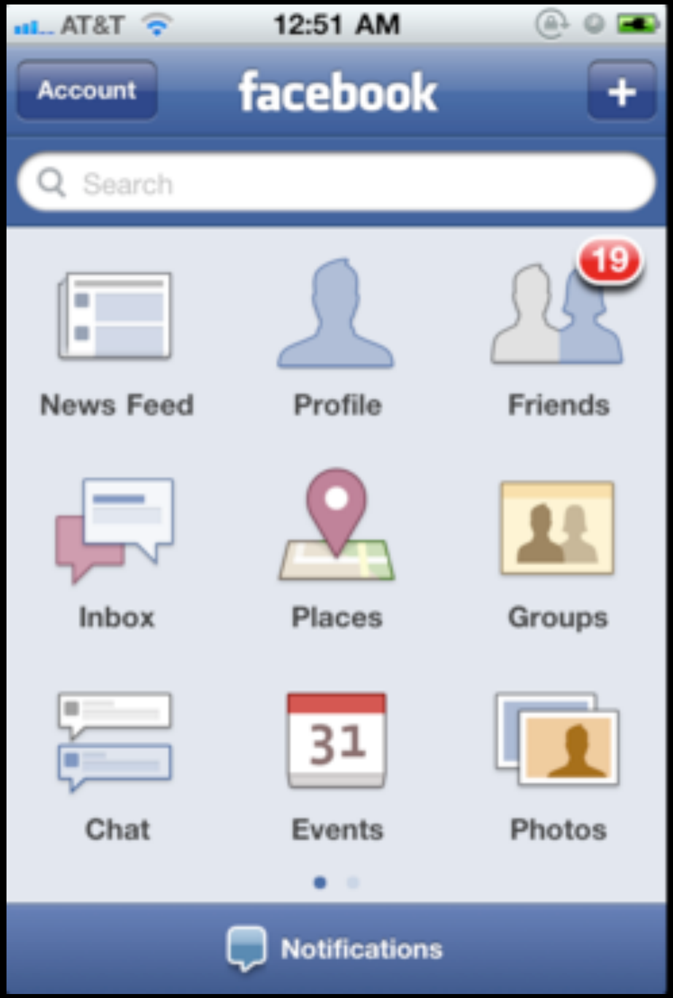


Soleio Cuervo likes H-P's One-Year Plan and 9 Video Game Easter Eggs That Took Years to Find.
H-P's One-Year Plan
18 minutes ago

Kate Aronowitz Early morning training resumes. San Jose Rock n Roll Half is only 5 weeks away!
Saturday at 5:56am
12 people 1 comment

Nicholas Felton was added to FeedBack To The Future.
33 minutes ago

Rose Yao good laugh



2012 - Rebuilding Facebook for iOS

~~Three20~~

~~Scaling up with html5~~

Rebuilding for speed

System of modules - shared code e.g. for Messenger

I DON'T ALWAYS BETA TEST
"BUT WHEN I DO, I DO IT IN PRODUCTION"
memegenerator.net

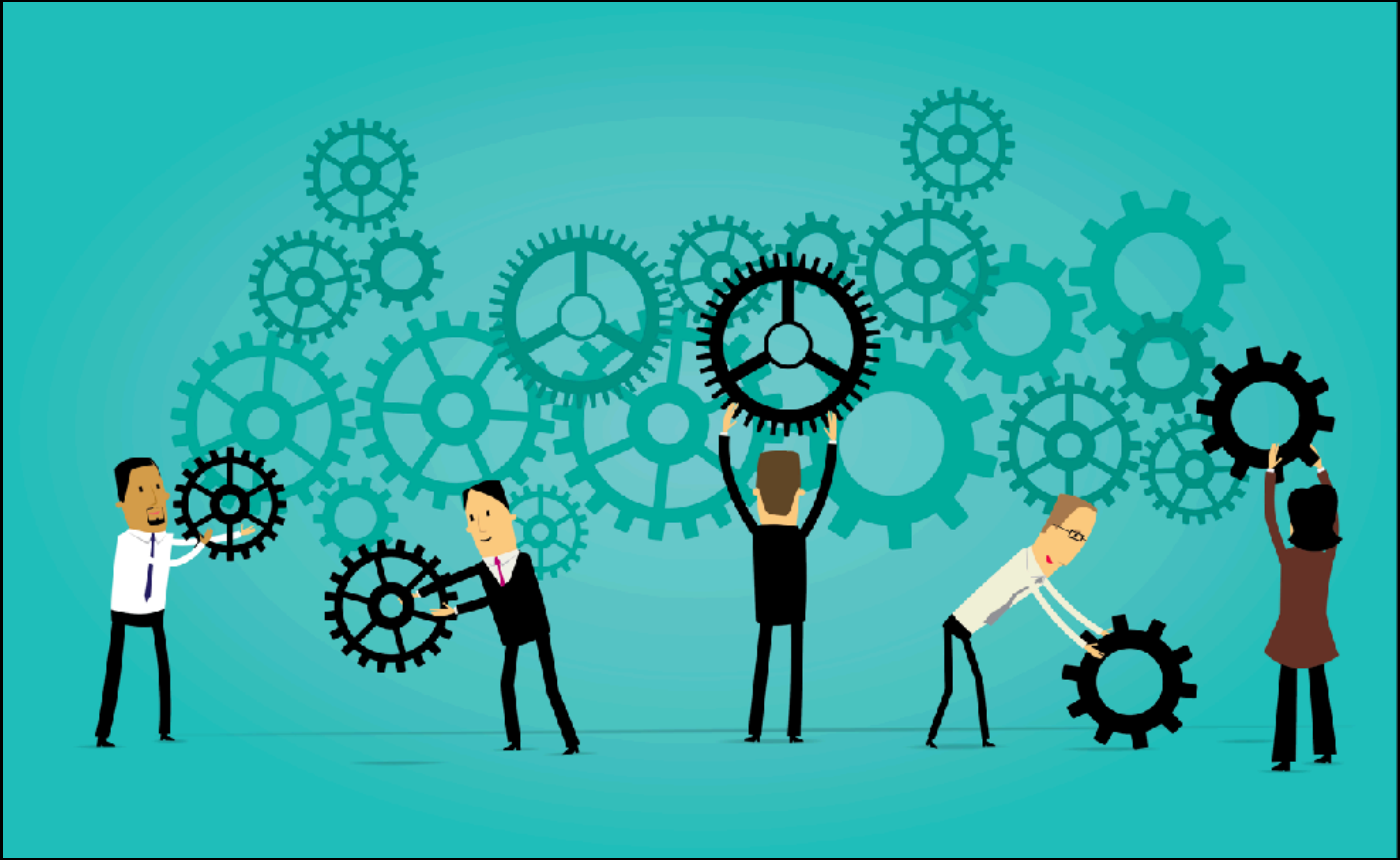
React native

Declarative

Component based

Learn once, write anywhere

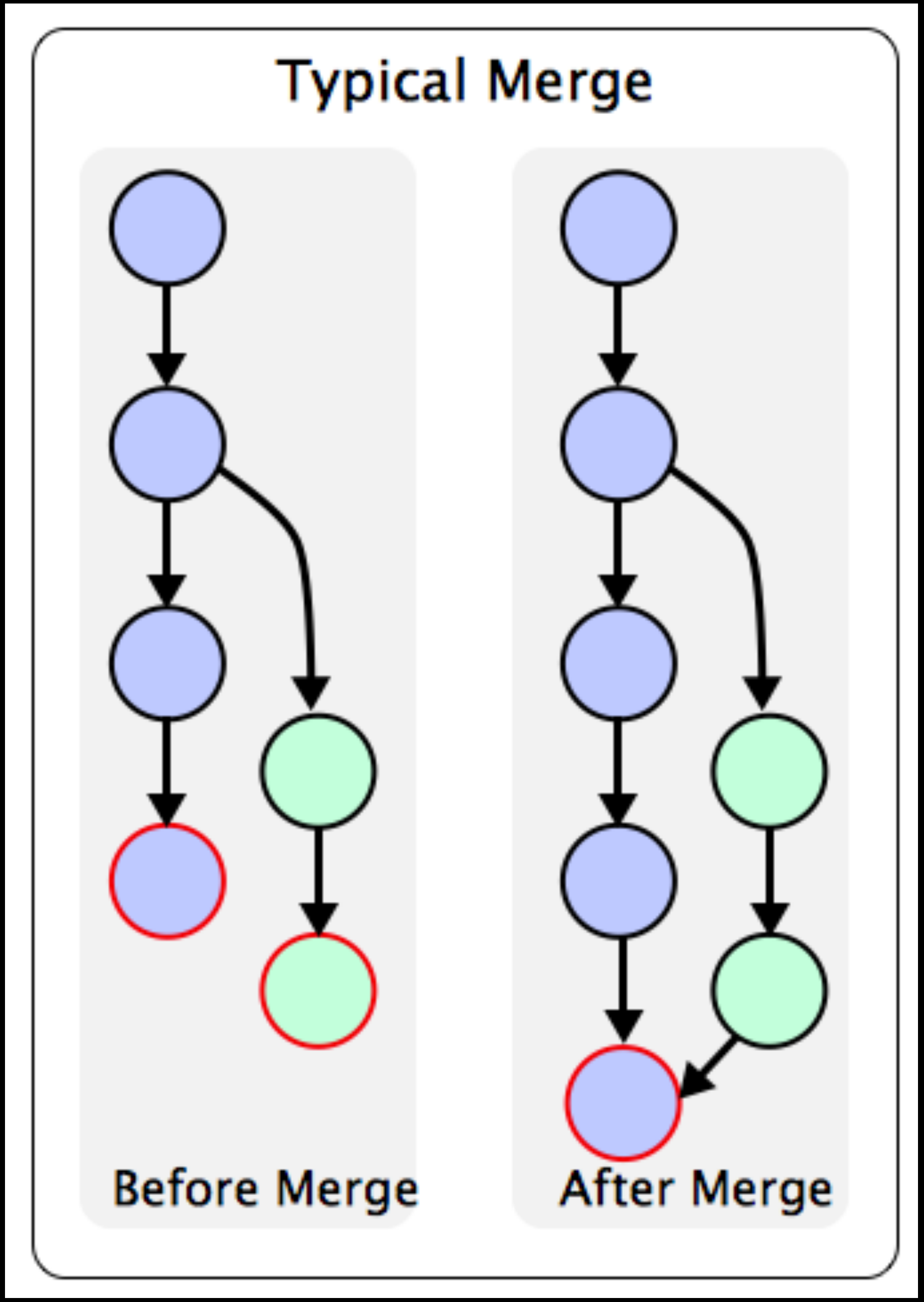
F8
SHOUT!

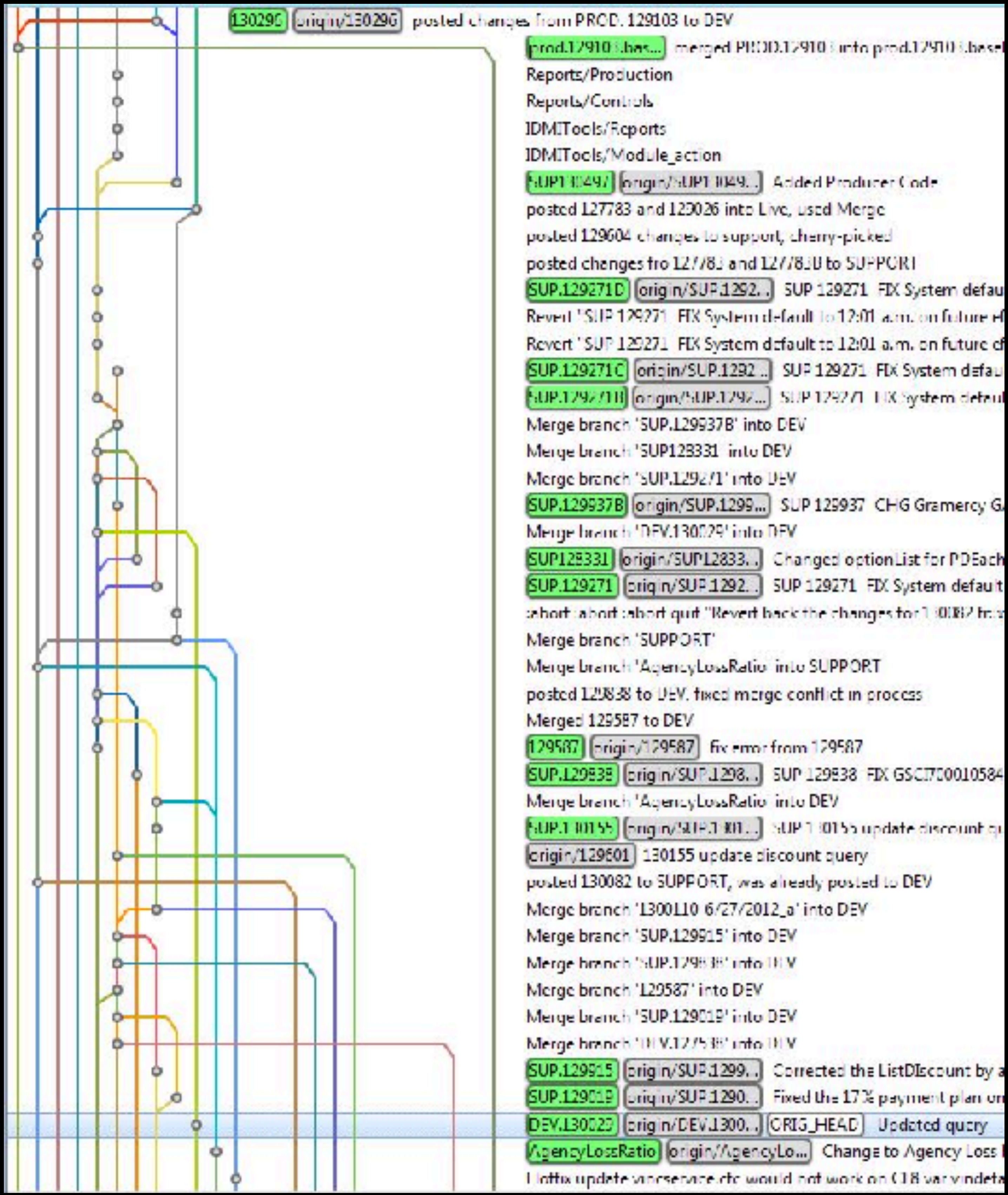


Recommendations on branching

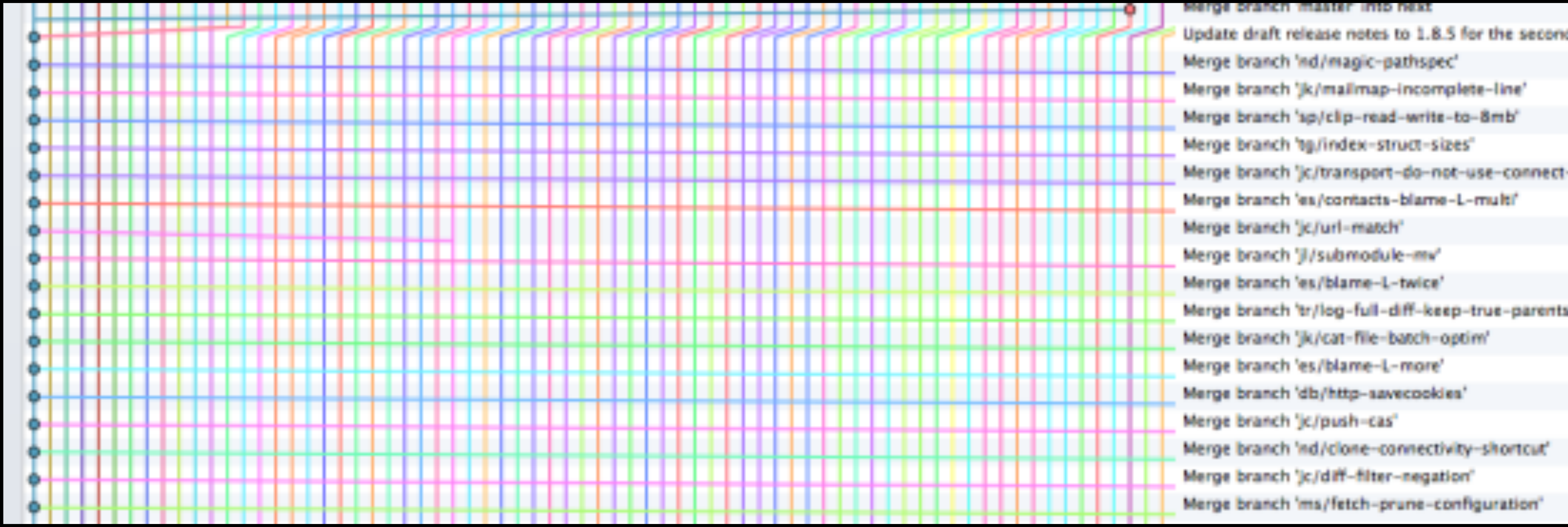
- never put feature branches in the remote/origin/trunk
- control access to new features with runtime configuration, not branching

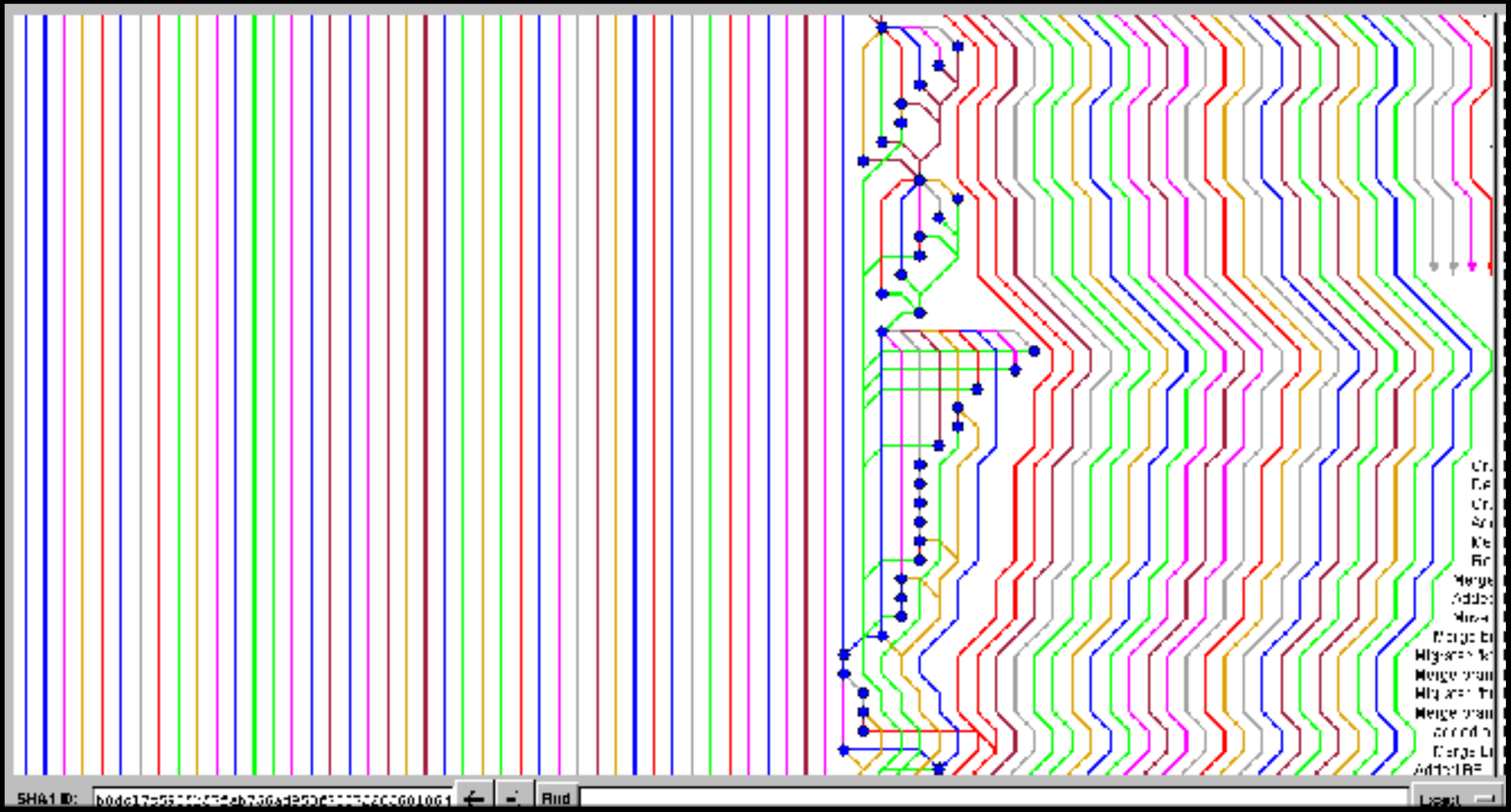
Choose a strategy where one idea is one commit in the authoritative master/remote version of the repository

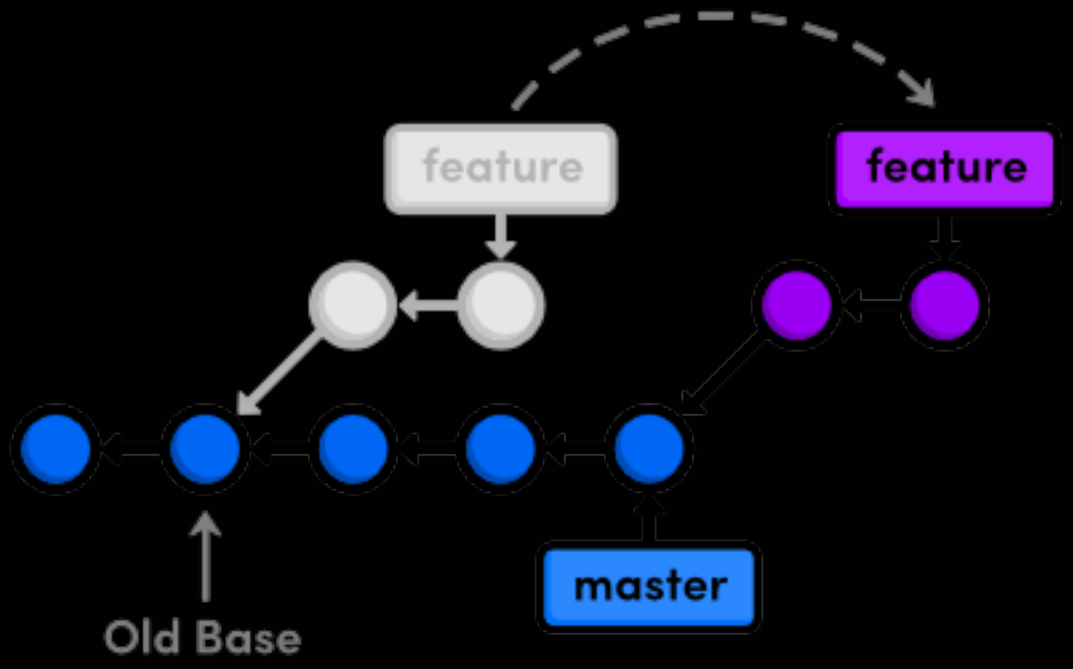


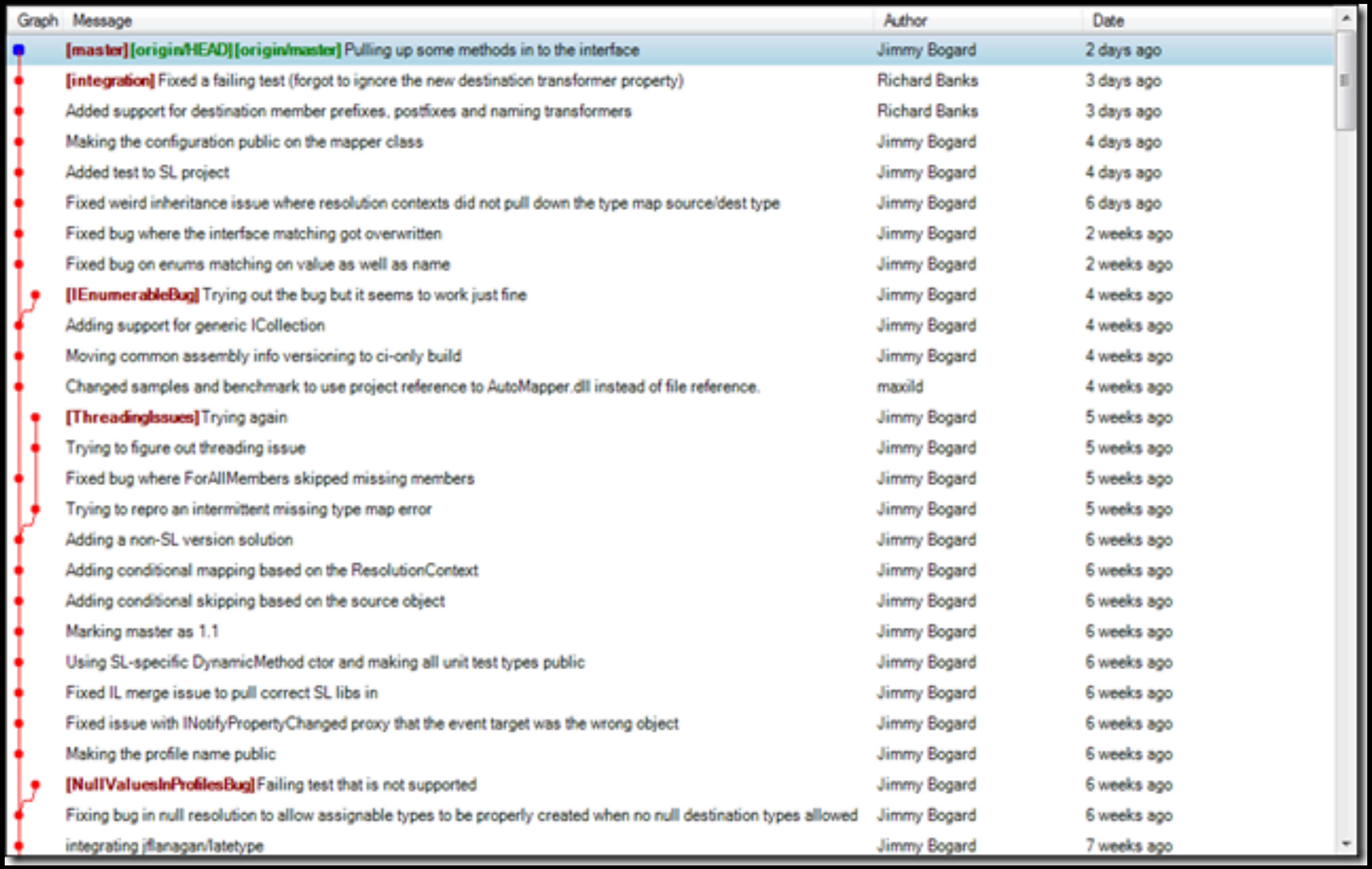


```
130296 origin/130296 posted changes from PROD. 129103 to DEV
prod.129103.bas... merged PROD.129103 into prod.129103.base
Reports/Production
Reports/Controls
IDMITools/Reports
IDMITools/Module_action
SUP130497 origin/SUP13049... Added Producer Code
posted 127783 and 123026 into Live, used Merge
posted 129604 changes to support, cherry-picked
posted changes fro 127783 and 127783D to SUPPORT
SUP.129271D origin/SUP.1292... SUP 129271 FIX System defau
Revert 'SUP 129271 FIX System default to 12:01 a.m. on future ef
Revert 'SUP 129271 FIX System default to 12:01 a.m. on future ef
SUP.129271C origin/SUP.1292... SUP 129271 FIX System defau
SUP.129271B origin/SUP.1292... SUP 129271 FIX System defaul
Merge branch 'SUP.129937B' into DEV
Merge branch 'SUP128331' into DEV
Merge branch 'SUP.129271' into DEV
SUP.129937B origin/SUP.1299... SUP 129937 CHG Gramercy G
Merge branch 'DEV.130029' into DEV
SUP128331 origin/SUP12833... Changed optionList for PDEach
SUP.129271 origin/SUP.1292... SUP 129271 FIX System default
:abort :abort :abort quit "Revert back the changes for 130082 fro
Merge branch 'SUPPORT'
Merge branch 'AgencyLossRatio' into SUPPORT
posted 129838 to DEV. fixed merge conflict in process
Merged 129587 to DEV
129587 origin/129587 fix error from 129587
SUP.129838 origin/SUP.1298... SUP 129838 FIX GSCI70001058
Merge branch 'AgencyLossRatio' into DEV
SUP.130155 origin/SUP.1301... SUP 130155 update discount qu
origin/129601 130155 update discount query
posted 130082 to SUPPORT, was already posted to DEV
Merge branch '130010 6/27/2012_a' into DEV
Merge branch 'SUP.129915' into DEV
Merge branch 'SUP.129838' into DEV
Merge branch '129587' into DEV
Merge branch 'SUP.129019' into DEV
Merge branch 'DEV.127538' into DEV
SUP.129915 origin/SUP.1299... Corrected the ListDiscount by a
SUP.129019 origin/SUP.1290... Fixed the 17% payment plan on
DEV.130029 origin/DEV.1300... ORIG_HEAD Updated query
AgencyLossRatio origin/AgencyLo... Change to Agency Loss
Hotfix update vincservice.cfc would not work on CF8 var vindeta
```


Merge branch 'master' into next
Update draft release notes to 1.8.5 for the second
Merge branch 'nd/magic-pathspec'
Merge branch 'jk/mailmap-incomplete-line'
Merge branch 'sp/clip-read-write-to-8mb'
Merge branch 'tg/index-struct-sizes'
Merge branch 'jc/transport-do-not-use-connect-
Merge branch 'es/contacts-blame-L-multi'
Merge branch 'jc/url-match'
Merge branch 'jl/submodule-mv'
Merge branch 'es/blame-L-twice'
Merge branch 'tr/log-full-diff-keep-true-parents
Merge branch 'jk/cat-file-batch-optim'
Merge branch 'es/blame-L-more'
Merge branch 'db/http-savecookies'
Merge branch 'jc/push-cas'
Merge branch 'nd/clone-connectivity-shortcut'
Merge branch 'jc/diff-filter-negation'
Merge branch 'ms/fetch-prune-configuration'







Graph	Message	Author	Date
	[master] [origin/HEAD] [origin/master] Pulling up some methods in to the interface	Jimmy Bogard	2 days ago
	[integration] Fixed a failing test (forgot to ignore the new destination transformer property)	Richard Banks	3 days ago
	Added support for destination member prefixes, postfixes and naming transformers	Richard Banks	3 days ago
	Making the configuration public on the mapper class	Jimmy Bogard	4 days ago
	Added test to SL project	Jimmy Bogard	4 days ago
	Fixed weird inheritance issue where resolution contexts did not pull down the type map source/dest type	Jimmy Bogard	6 days ago
	Fixed bug where the interface matching got overwritten	Jimmy Bogard	2 weeks ago
	Fixed bug on enums matching on value as well as name	Jimmy Bogard	2 weeks ago
	[IEnumerableBug] Trying out the bug but it seems to work just fine	Jimmy Bogard	4 weeks ago
	Adding support for generic ICollection	Jimmy Bogard	4 weeks ago
	Moving common assembly info versioning to ci-only build	Jimmy Bogard	4 weeks ago
	Changed samples and benchmark to use project reference to AutoMapper.dll instead of file reference.	maxild	4 weeks ago
	[ThreadingIssues] Trying again	Jimmy Bogard	5 weeks ago
	Trying to figure out threading issue	Jimmy Bogard	5 weeks ago
	Fixed bug where ForAllMembers skipped missing members	Jimmy Bogard	5 weeks ago
	Trying to repro an intermittent missing type map error	Jimmy Bogard	5 weeks ago
	Adding a non-SL version solution	Jimmy Bogard	6 weeks ago
	Adding conditional mapping based on the ResolutionContext	Jimmy Bogard	6 weeks ago
	Adding conditional skipping based on the source object	Jimmy Bogard	6 weeks ago
	Marking master as 1.1	Jimmy Bogard	6 weeks ago
	Using SL-specific DynamicMethod ctor and making all unit test types public	Jimmy Bogard	6 weeks ago
	Fixed IL merge issue to pull correct SL libs in	Jimmy Bogard	6 weeks ago
	Fixed issue with INotifyPropertyChanged proxy that the event target was the wrong object	Jimmy Bogard	6 weeks ago
	Making the profile name public	Jimmy Bogard	6 weeks ago
	[NullValuesInProfilesBug] Failing test that is not supported	Jimmy Bogard	6 weeks ago
	Fixing bug in null resolution to allow assignable types to be properly created when no null destination types allowed	Jimmy Bogard	6 weeks ago
	integrating jflanagan/latetype	Jimmy Bogard	7 weeks ago

Feature branches

Cons

- you have to merge
- this strategy generally aggregates risk into a single high-risk merge event
  of development
- when you have multiple feature branches, it's impossible to test interactio
  the features until they are merged
- you generally can't A/B test code in feature branches

Pros

- replacing old feature
- the chance that this code will impact production before the merge is nearl

Abandoning feature branches

Advantages

- you don’t have to merge
- risk is generally spread out more evenly into a large number of very small ri created as each commit lands
- you can test interactions between features in development easily
- you can A/B test and do controlled rollouts easily

Tradeoffs

- if a new feature replaces an older feature, both have to exist in the same co a while
- you need an effective way to control access to features so they don't launch they're ready

Controlling access to features

Gatekeeper

```
if is_feature_launched(“like_button”) {
   showLikeButton()
}
```

Gatekeeper

- allowing features to have states like "3%" instead of just "on" or "off" allows you to roll out features gradually and watch for trouble
- if you perform A/B testing, integrating A/B tests with feature rollouts is probably a natural fit.
- building a control panel where you hit "Save" and all production servers immediately reflect the change allows you to quickly turn things off if there are problems

Recommendations on Revision Control

When projects scale, strategies which enforce one idea is one commit are better

- when one idea is many commits, everything you do is more complicated because you need to figure out which commits represent an idea
- release engineering is greatly simplified
- automated testing is greatly simplified
- understanding changes is greatly simplified
- there is no clear value in having checkpoint commits

PHABRICATOR

review code

host git/svn/mercurial

build with continuous integration

review designs

discuss in internal chat channels

Writing reviewable code

- the smallest a commit can be is a single cohesive idea
- there should be a one-to-one mapping between ideas and commit
- turn large commits into small commits by dividing large problems into small problems
- write sensible commit messages

Title

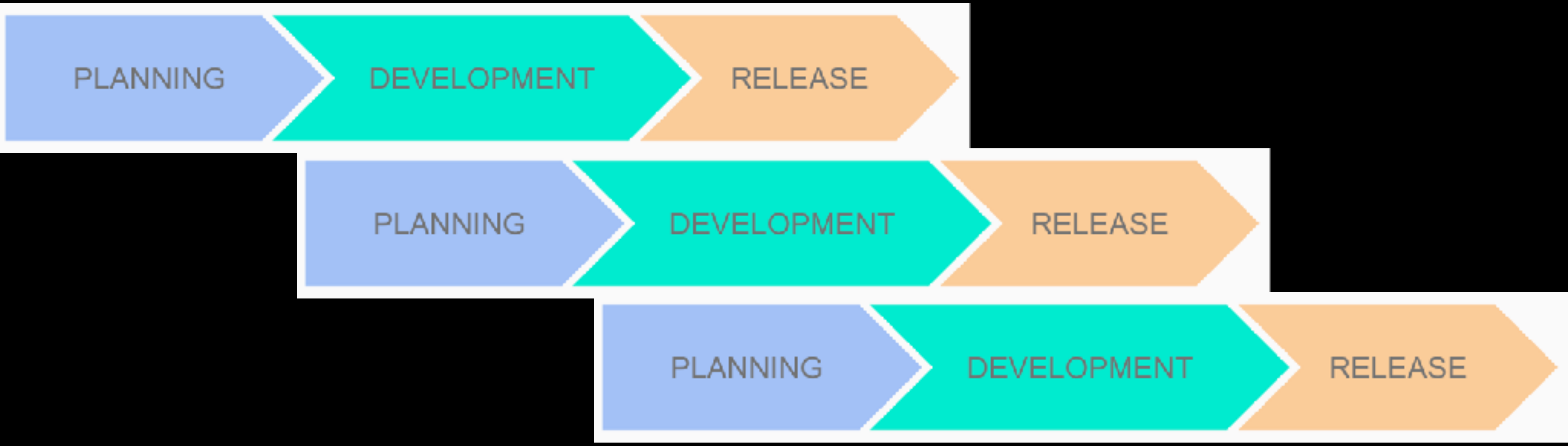
Summary:
Brief explanation what you have done in this commit

Test plan:

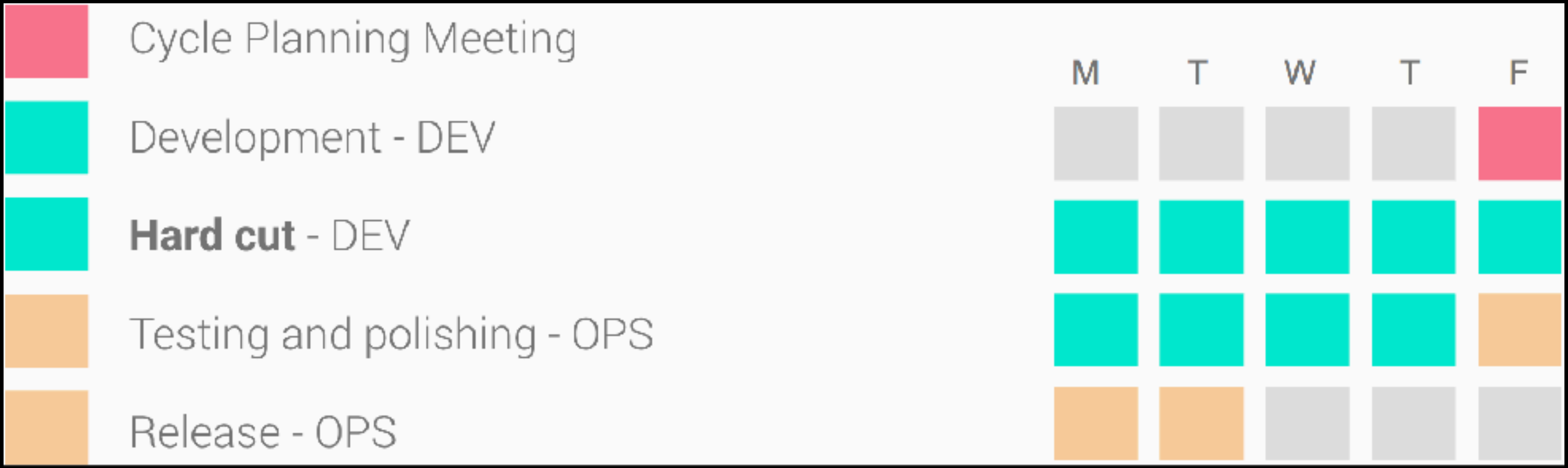
- exhaustive test plan
- - writing down edge cases
- ‘it works’ or ‘it compiles’ is not a good test plan
- error handling, service impact, performance, unit tests, concurrent change robustness, revert plan, security

Development cycle

“A week of coding can save you an hour of thinking.”



Two week cycle



🟥	Cycle Planning Meeting
🟩	Development - DEV
🟩	**Hard cut** - DEV
🟧	Testing and polishing - OPS
🟧	Release - OPS

M	T	W	T	F
⬜	⬜	⬜	⬜	🟥
🟩	🟩	🟩	🟩	🟩
🟩	🟩	🟩	🟩	🟧
🟧	🟧	⬜	⬜	⬜

Development cycle

- Do not postpone releases to ship features.
- Ship a subset of the feature to meet the release deadline.
- During planning phase - split features into smaller batches.
- When you’re blocked, resolve the problem, ask for help.
- Report progress regularly. And setbacks.

Testing

- Engineer
- Dogfooding
- gatekeeper
- Quick experiments
- Monitoring metrics

Buck

Speed up your builds

It encourages the creation of small, reusable modules

Add reproducibility to your builds

Better understanding your dependencies

Component Kit

A React-Inspired View Framework for iOS

one-way data flow from immutable models to immutable components

No need to do any calculations for view layout

Snapshot testing

https://github.com/facebook/ios-snapshot-test-case

It’s straightforward to test logic code, but less obvious how you should test views

FBSnapshotTestCase

https://code.facebook.com/projects/ios/

Shimmer

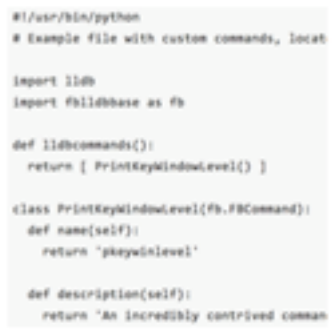
Tweaks

tweaks

pop

pop

Shimmer for iOS



```
#!/usr/bin/python
# Example file with custom commands, locat

import lldb
import fblldbbase as fb

def lldbcommands():
  return [ PrintKeyWindowLevel() ]

class PrintKeyWindowLevel(fb.FBCommand):
  def name(self):
    return 'pkeywinlevel'

  def description(self):
    return 'An incredibly contrived comman
```

chisel

Swift at Bellabeat

transitioning from obj c to swift

far less crashes

far less code

rxSwift

single repo - shared code in modules

Takeaways

switch from pushing feature branches to remote - to single master/remote branch

create a mechanism that allows you to switch off features

create small reusable modules

keep up with the open source community

most of the problems happen only when your repo hits a specific velocity

Thank you