Apps with love

Bachelor Project

Presentation II – Process & Results

4. February 2020

Alain Stulz / 2020

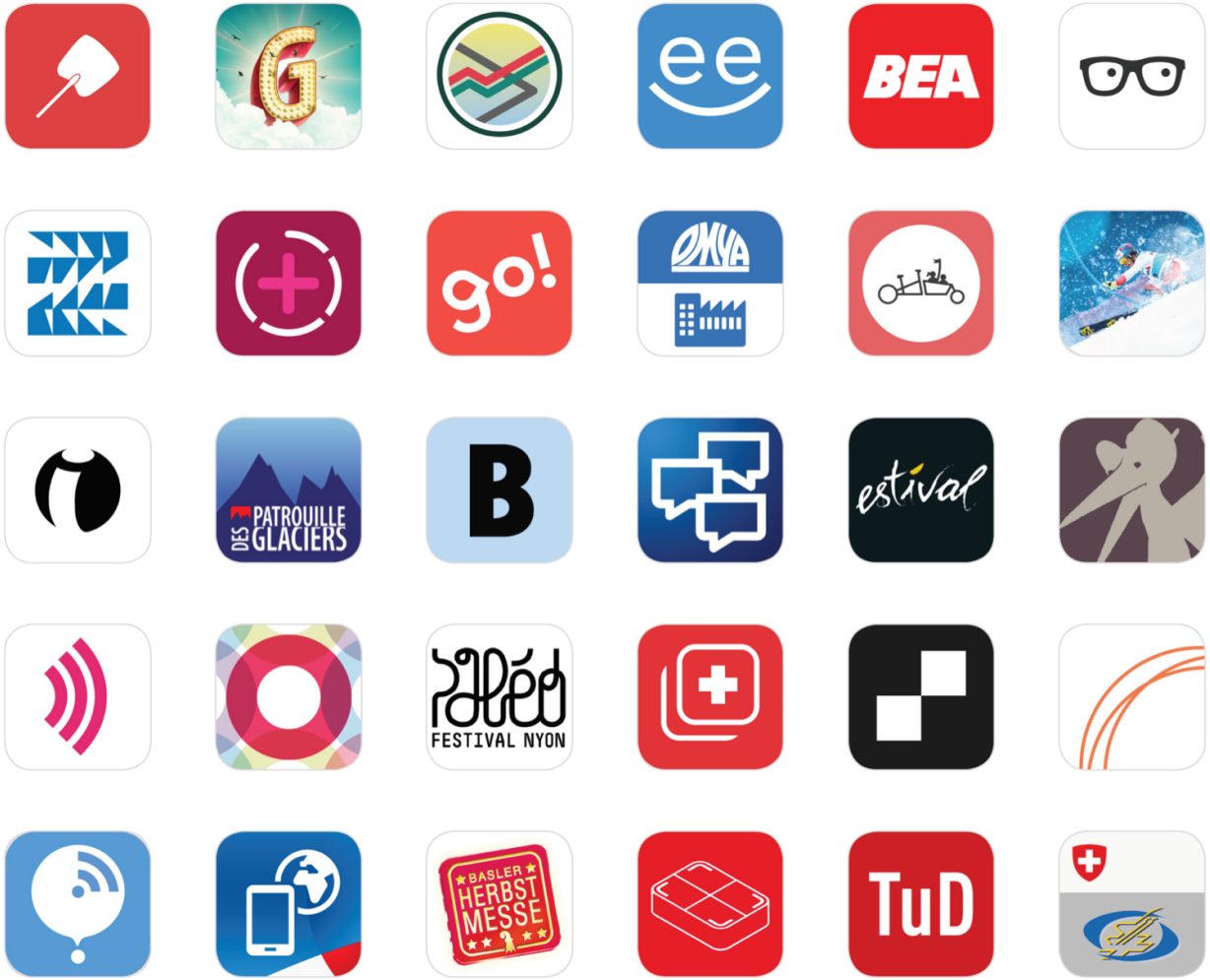
Assessing and Improving the Software Quality of an iOS App Framework

Recap

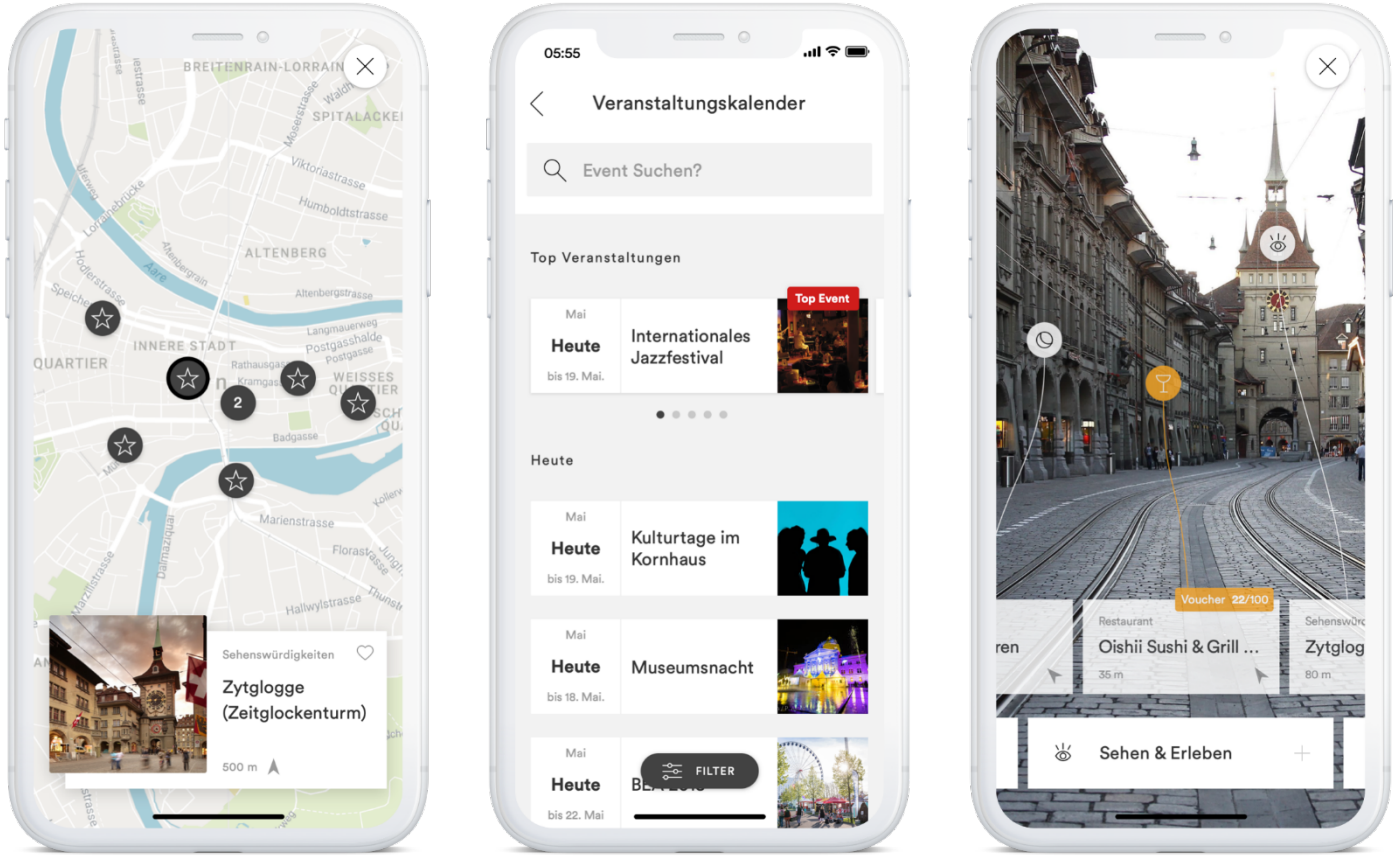
“Festival Buddy”

How To Create Many Apps Efficiently?

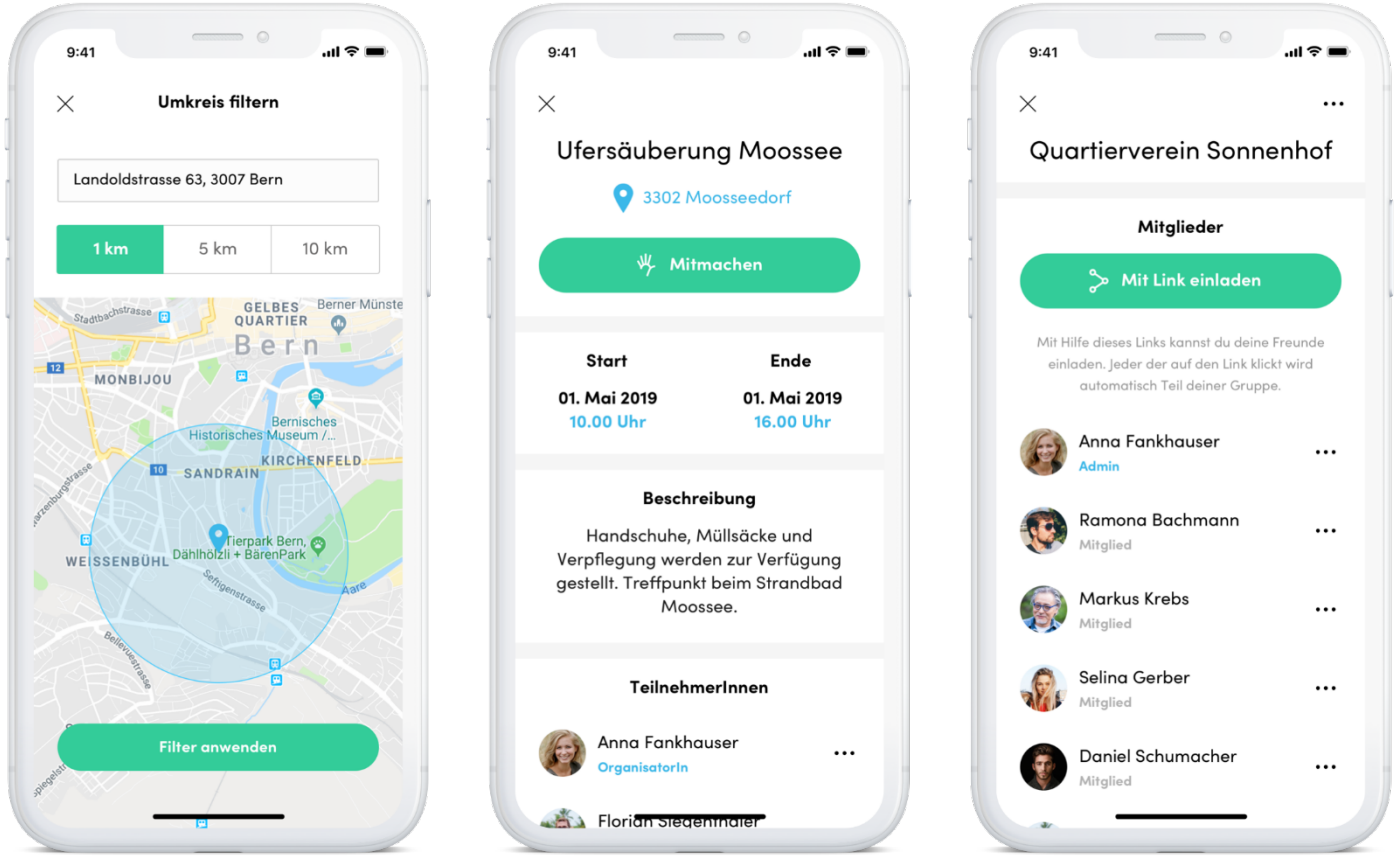
- Custom Framework



Bern Welcome



Five up

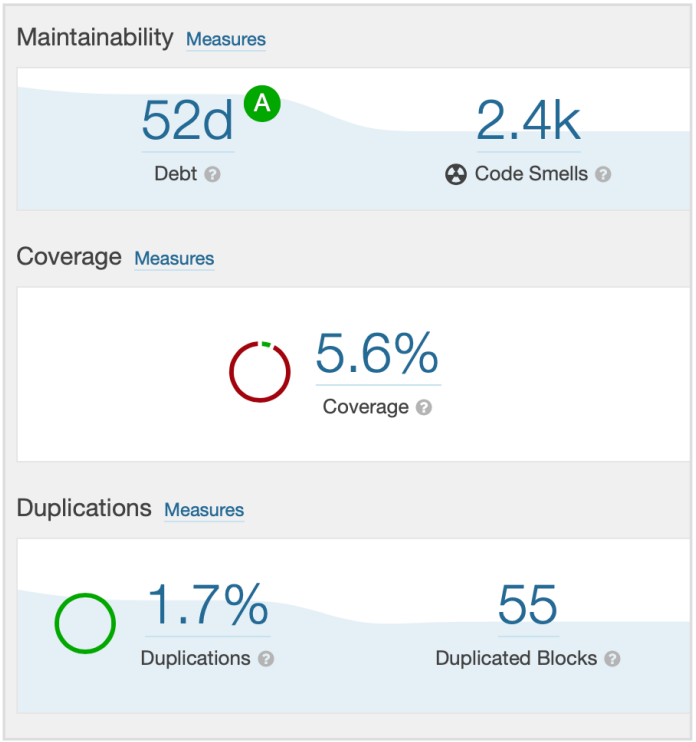

9:41
Umkreis filtern
Landoldstrasse 63, 3007 Bern
1 km
5 km
10 km
GELBES QUARTIER
Bern
MONBIJOU
Bernisches Historisches Museum /...
KIRCHENFELD
SANDRAIN
WEISSENBÜHL
Tierpark Bern, Dählhölzli + BärenPark
Filter anwenden
9:41
Ufersäuberung Moossee
3302 Moosseedorf
Mitmachen
Start
01. Mai 2019
10.00 Uhr
Ende
01. Mai 2019
16.00 Uhr
Beschreibung
Handschuhe, Müllsäcke und Verpflegung werden zur Verfügung gestellt. Treffpunkt beim Strandbad Moossee.
TeilnehmerInnen
Anna Fankhauser
OrganisatorIn
9:41
Quartierverein Sonnenhof
Mitglieder
Mit Link einladen
Mit Hilfe dieses Links kannst du deine Freunde einladen. Jeder der auf den Link klickt wird automatisch Teil deiner Gruppe.
Anna Fankhauser
Admin
Ramona Bachmann
Mitglied
Markus Krebs
Mitglied
Selina Gerber
Mitglied
Daniel Schumacher
Mitglied

Analyze

RQ1: How can we assess the quality of our system?

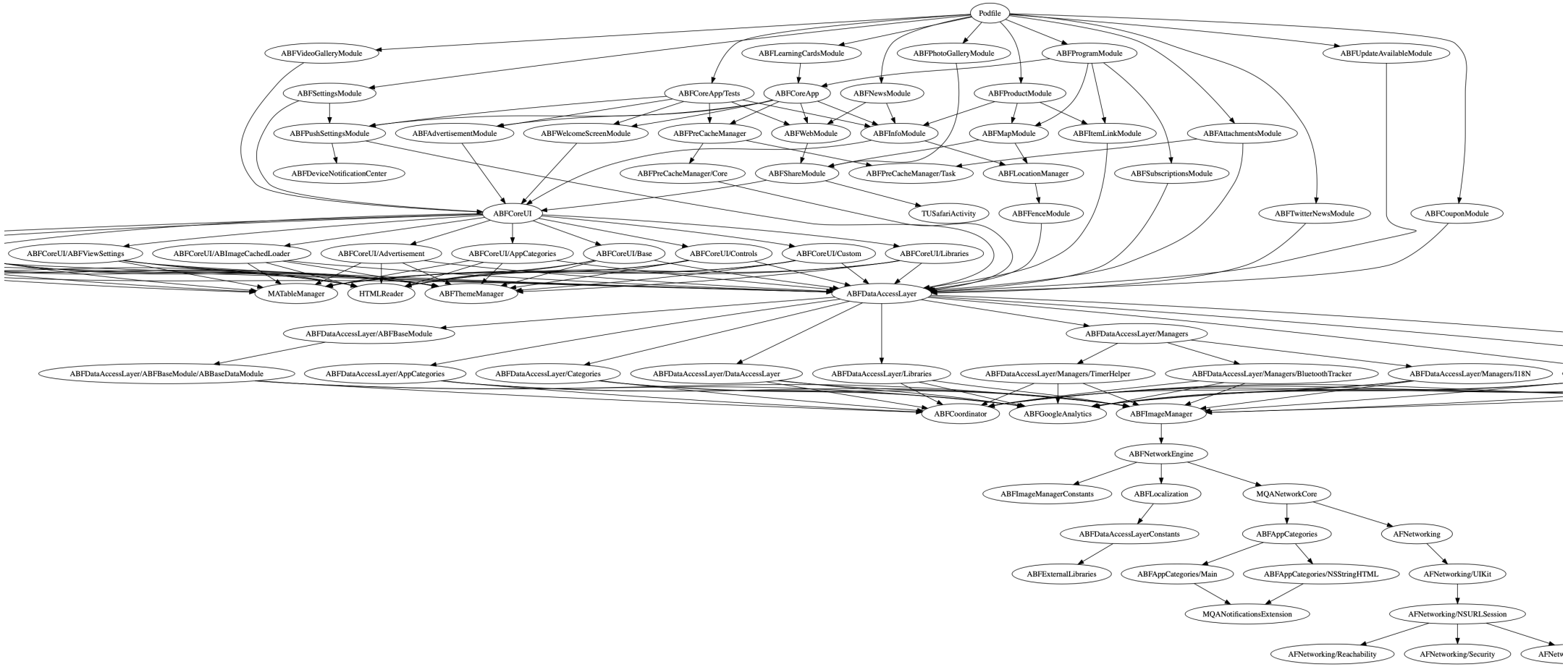
Non-Functional Requirements

Code Level



sonarqube

Module Level



Organization Level

Project	Priority
academy-ios	0
adelboden-ultimatefan-ios	0
arud-consumption-diary-ios	1
BernWelcome_iOS	8
EDA_DFAE_iOS	8
gnomenweg-ios	0
gymme-ios	0
hep-project-journal-ios	1
herbstmesse-ios	0
hornussen-ios	1
inlingua-ios	4
lukb-ios	1
marketing-expert-ios	1
marketing-lexicon-ios	1
mindpills-ios	0
mplay-massimo-fall2019-ios	8

Modules from Framework	Imports	Usages	Decision	Focus
ABFCoreUI	8	8	Scope	Quality
ABFDataAccessLayer	8	8	Scope	Quality
ABFCoreApp	8	8	Scope	Quality
ABFInfoModule	8	6	Scope	Quality
ABFNetworkEngine	8	6	Scope	Quality
ABFCoordinator	7	5	Scope	Quality
ABFDataAccessLayerConstants	8	4	Ignore	
ABFLocationManager	8	3	Ignore	
ABFAttachmentsModule	3	3	Ignore	
ABFPreCacheManager	8	3	Ignore	
MATableManager	9	3	Scope	Dependencies
ABFAppCategories	8	2	Scope	Dependencies
ABFImageManager	8	1	Scope	Dependencies
ABFNewsModule	1	1	Ignore	
ABFPushSettingsModule	8	2	Scope	Dependencies
ABFSettingsModule	2	2	Ignore	
ABFShareModule	8	2	Scope	Dependencies
ABFWebModule	8	2	Scope	Dependencies
ABFAuthenticationModule	1	1	Ignore	
ABFLearningCardsModule	1	1	Ignore	
ABFLocalization	8	1	Scope	Dependencies
~~ABFMapModule~~	~~0~~	~~0~~	~~Remove~~	
ABFPhotoGalleryModule	1	1	Ignore	
ABFPhotoPicker	1	1	Ignore	
~~ABFProductModule~~	~~0~~	~~0~~	~~Remove~~	
~~ABFProgramModule~~	~~0~~	~~0~~	~~Remove~~	

Developer Interviews

+ Easier to start projects
+ Consistency and speed

- Hard to understand and change
- Effects of changes not visible

Lessons Learned

RQ1: How can we assess the quality of our system?

- Focus on Developer’s Perspective
  - NFRs: Maintainability, Evolvability, …
- Use Different Data Sources
- Specific Tools for iOS Projects

Improve

RQ2: How can we improve the system’s quality?

Improve

1. Process
2. Maintenance
3. Refactoring

Improve

1. Process
2. Maintenance
3. Refactoring

- Developer Conventions
- Continuous Integration
  - Automate Client Projects
  - Build Feedback

Improve

1. Process
2. Maintenance
3. Refactoring

- Remove Old Code
- Rearrange Methods

Improve

1. Process
2. Maintenance
3. Refactoring

- Cover and Modify
- Techniques

Lessons Learned

RQ2: How can we improve the system’s quality?

- Set up Systems and Conventions
- Perform Basic Maintenance
- Cover and Modify
- Use Delegates for Tests

Rewrite

RQ3: What would a better software design look like?

Rewrite

1. Architecture
2. Benefits

Rewrite

1. Architecture
2. Benefits

- Clean Swift Architecture
  - Split into Scenes
  - Protocols
- Modularization

Rewrite

1. Architecture
2. Benefits

- Only Relevant Features
- Less Clutter in Framework
  - Define UI in Client Projects
- More Flexibility

Lessons Learned

RQ3: What would a better software design look like?

- Small and Generalized
- Newer Tools
- Well-Defined Architecture

Conclusion

Conclusion

- Recommendation: Rewrite
  - Changed Requirements, New Opportunities
- Refactor vs. Rewrite
  - Only if you need to
  - Refactor if you need to keep system
  - Rewrite if system holds you back

Summary

Analyze

- Focus on NFRs, Different Sources, Specific Tools

Improve

- Systems and Conventions, Reduce Size, Cover and Modify

Rewrite

- Small and Flexible, Define Architecture, Allow Customization

Testability

- Business Logic, not UI
- Move Logic out of Centralized Classes
  - Single Responsibility
  - DB and Network Calls
- Reduce Dependencies
  - Precompiler Flags
  - Leverage Delegates for Tests

Strategies

- Precompiler Flags
  - #define and #ifdef
  - Escape out of a method early
- Mock Delegates
  - Offer a well-defined interface
  - Control and get insight into class
- Optional Protocol Methods
  - Use @optional Keyword
  - Implement methods only for testing (e.g. shouldCallAPI)