

钉钉iOS团队

质量控制和效率提升之道

关于钉钉

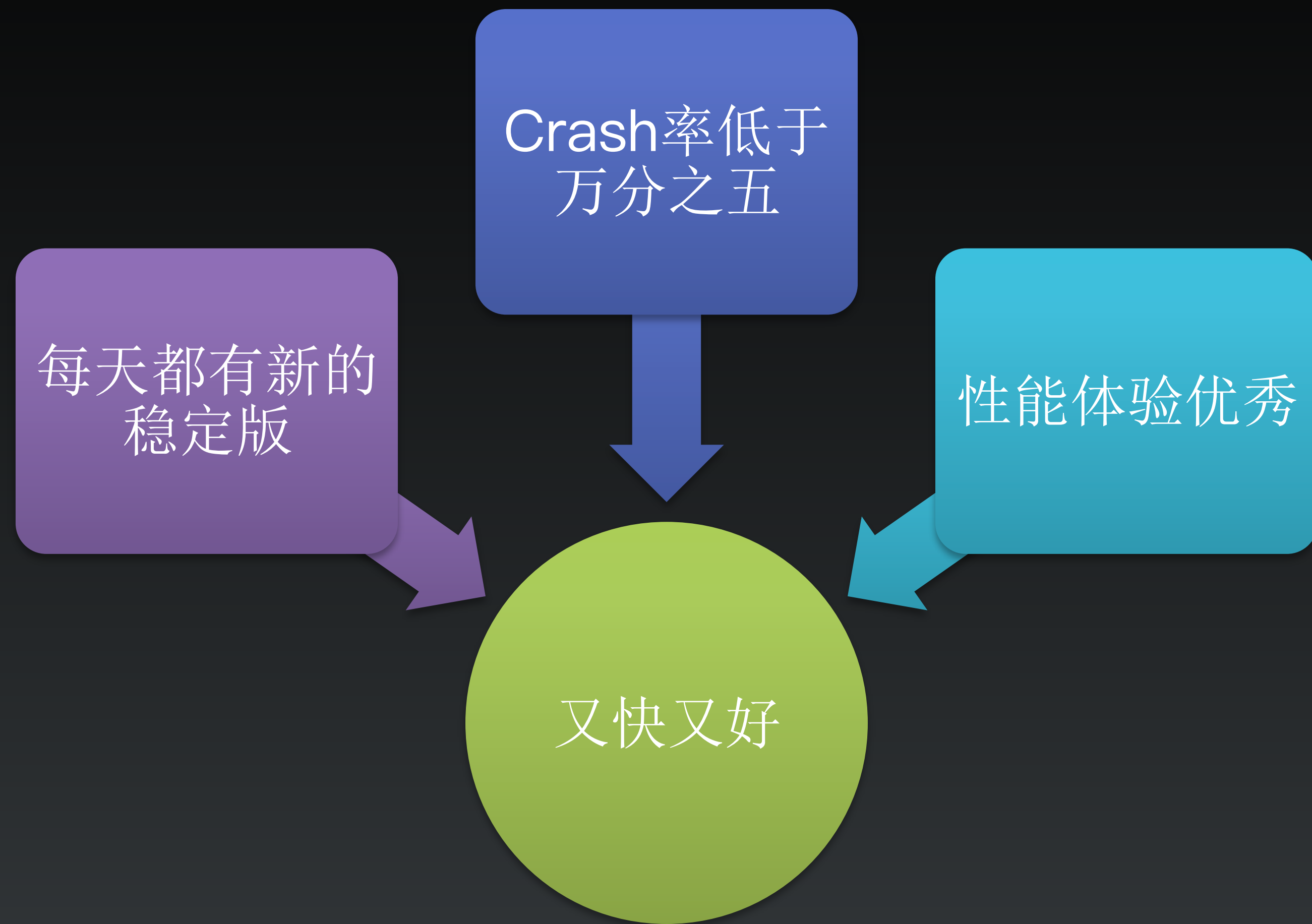
钉钉，阿里巴巴出品，专为中国企业打造的免费沟通协同多端平台，含PC版，Web版和手机版。消息已读未读，DING消息使命必达，让沟通更高效；移动办公签到，审批，邮箱云盘，让工作更简单；澡堂模式，企业通讯录，让信息更安全。使用钉钉，全方位提升企业内部沟通与协同。

关于钉钉iOS团队

- 初创5人，当前10人，未来12~14人
- 支持全平台：iPhone、iPad、Apple Watch







开发效率

1. 视觉字典：颜色、字体



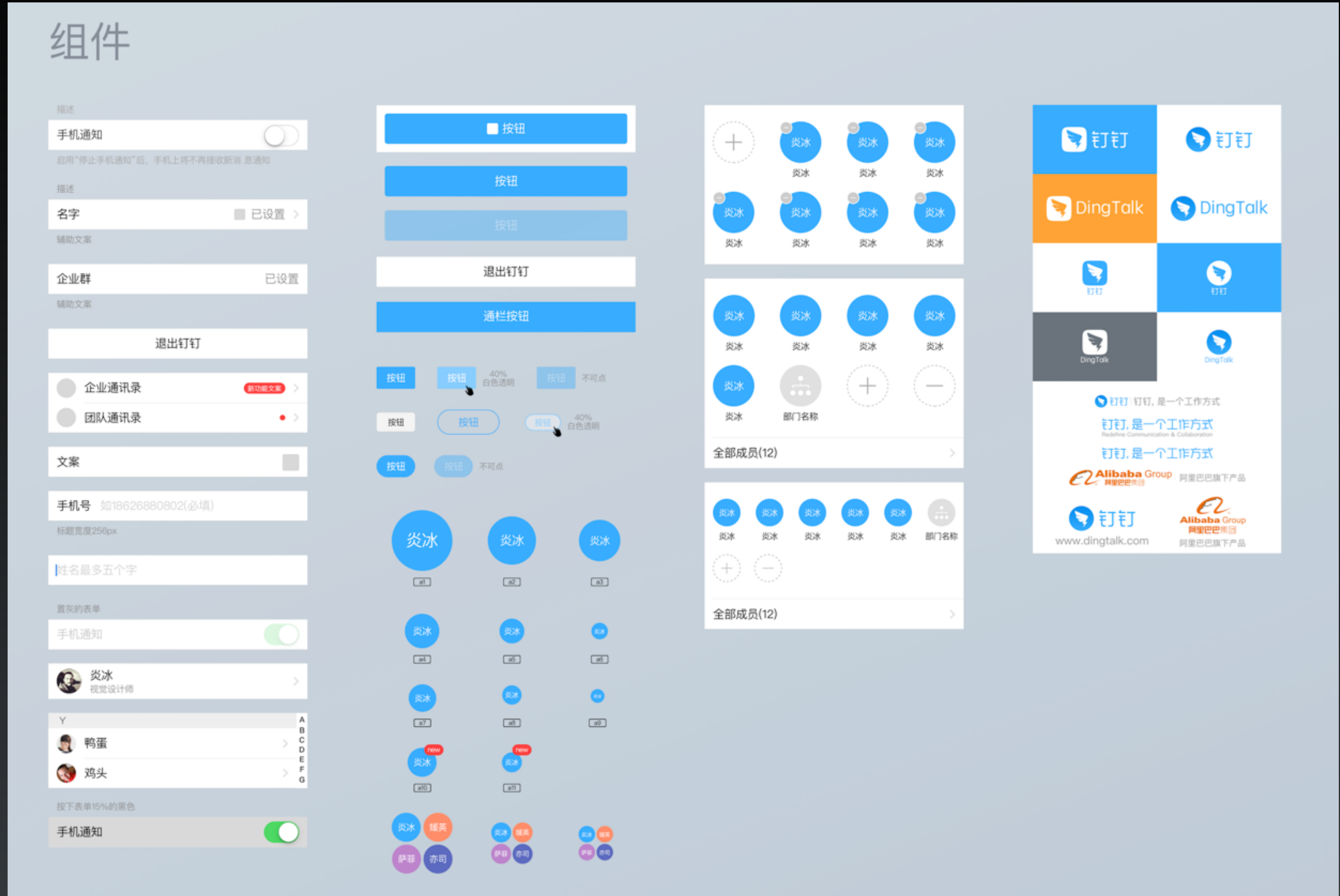
```
78 + (UIFont *)DT_S20_normal;  
79 + (UIFont *)DT_S22_normal;  
80 + (UIFont *)DT_S24_normal;  
81 + (UIFont *)DT_S26_normal;  
82 + (UIFont *)DT_S28_normal;  
83 + (UIFont *)DT_S32_normal;  
84 + (UIFont *)DT_S40_normal;  
85 + (UIFont *)DT_S48_normal;  
86 + (UIFont *)DT_S52_normal;  
87  
88 + (UIFont *)DT_S06_bold;  
89 + (UIFont *)DT_S08_bold;  
90 + (UIFont *)DT_S09_bold;  
91 + (UIFont *)DT_S10_bold;  
92 + (UIFont *)DT_S11_bold;  
93 + (UIFont *)DT_S12_bold;  
94 + (UIFont *)DT_S13_bold;  
95 + (UIFont *)DT_S14_bold;  
96 + (UIFont *)DT_S15_bold;  
97 + (UIFont *)DT_S16_bold;  
98 + (UIFont *)DT_S17_bold;  
99 + (UIFont *)DT_S18_bold;  
100 + (UIFont *)DT_S19_bold;  
101 + (UIFont *)DT_S20_bold;  
102 + (UIFont *)DT_S22_bold;  
103 + (UIFont *)DT_S24_bold;  
104 + (UIFont *)DT_S26_bold;  
105 + (UIFont *)DT_S28_bold;  
106 + (UIFont *)DT_S32_bold;  
107 + (UIFont *)DT_S40_bold;  
108 + (UIFont *)DT_S48_bold;  
109 + (UIFont *)DT_S52_bold;  
110  
111 //调用带Scale方法返回的字体将根据全局字体大小参数动态返回大小  
112 + (UIFont *)DT_S06_normal_scale;  
113 + (UIFont *)DT_S08_normal_scale;  
114 + (UIFont *)DT_S09_normal_scale;  
115 + (UIFont *)DT_S10_normal_scale;  
116 + (UIFont *)DT_S11_normal_scale;  
117 + (UIFont *)DT_S12_normal_scale;  
118 + (UIFont *)DT_S13_normal_scale;  
119 + (UIFont *)DT_S14_normal_scale;
```

```
52  
53 + (UIColor *)DT_C1;  
54 + (UIColor *)DT_C2;  
55 + (UIColor *)DT_C3;  
56 + (UIColor *)DT_C4;  
57 + (UIColor *)DT_C5;  
58 + (UIColor *)DT_C6_1;  
59 + (UIColor *)DT_C6_2;  
60 + (UIColor *)DT_C6_3;  
61 + (UIColor *)DT_C6_4; //保留, 慎用  
62 + (UIColor *)DT_C6_5;  
63 + (UIColor *)DT_C6_6;  
64 + (UIColor *)DT_C6_7;  
65 + (UIColor *)DT_C6_9;  
66 + (UIColor *)DT_C7_1;  
67 + (UIColor *)DT_C7_2;  
68 + (UIColor *)DT_C7_3;  
69 + (UIColor *)DT_C7_4;  
70 + (UIColor *)DT_C7_5;  
71 + (UIColor *)DT_C7_6;  
72 + (UIColor *)DT_C7_7;  
73 + (UIColor *)DT_C7_8;  
74 + (UIColor *)DT_C7_9;  
75 + (UIColor *)DT_C8_1;  
76 + (UIColor *)DT_C8_2;  
77 + (UIColor *)DT_C8_3;  
78 + (UIColor *)DT_C8_4;  
79 + (UIColor *)DT_C8_5;  
80 + (UIColor *)DT_C8_6;  
81 + (UIColor *)DT_C8_7;  
82 + (UIColor *)DT_C8_8;  
83 + (UIColor *)DT_C8_9;  
84 + (UIColor *)DT_C9_1;  
85 + (UIColor *)DT_C9_2;  
86 + (UIColor *)DT_C9_3;  
87 + (UIColor *)DT_C9_4;  
88 + (UIColor *)DT_C9_5;  
89 + (UIColor *)DT_C9_6;  
90 + (UIColor *)DT_C9_7;  
91 + (UIColor *)DT_C9_8;  
92 + (UIColor *)DT_C9_9;  
93 + (UIColor *)DT_C9_10;  
94 + (UIColor *)DT_C10_1;  
95 + (UIColor *)DT_C10_2;  
96 + (UIColor *)DT_C10_3;  
97 + (UIColor *)DT_C10_4;  
98 + (UIColor *)DT_C10_5;  
99 + (UIColor *)DT_C10_6;  
100 + (UIColor *)DT_C10_7;
```


开发效率

2. 基础控件：

- DTAvatarView
- DTButton
- DTTextField
- DTLabel
- DTTableView
- DTTableViewCell
- ...



开发效率

3. 制订编码规范，定义大量的Best Practice, Patterns

性能、多线程问题

- DB操作一定要在异步线程；
- 文件I/O一定要在异步线程；
- 开异步线程，请统一使用DTQueueManager，不允许使用dispatch_get_global_queue，除非自己模块需要单独队列；
- Service或者其他公用接口，方法名必须明确回调的block是否在主线程执行；
- 一定不能在异步线程操作UI；
 - 可以考虑一下如何统一监测机制，在debug阶段就能发现问题；
- 使用 performSelector:withObject:afterDelay:和 cancelPreviousPerformRequestsWithTarget 组合的时候要小心
 - afterDelay会增加receiver的引用计数，cancel则会对应减一
 - 如果在receiver的引用计数只剩下1（仅为delay）时，调用cancel之后会立即销毁receiver，后续再调用receiver的方法就会crash
 - 解决方法：

```
__weak typeof(self) weakSelf = self;
[NSObject cancelPreviousPerformRequestsWithTarget:self];
if (!weakSelf)
{
    //NSLog(@"self被销毁");
    return;
}

[self doOther];
```

- NotificationCenter在iOS 8及更老系统上的bug

当post notification早于另一个线程里dealloc的removeObserver，这时selector执行到一半时可能会因为self销毁而触发crash。修复方案是在selec

开发效率

4. 通用组件独立Pod

5. 绝对不要重复造轮子，大量使用二方、三方库（注意开放协议，以MIT、BSD、Apache为主），不满足需求的基于其进行扩展。懂得借力对于小团队提升效率来说非常重要。

6. 技术架构设计在APP创建初期的时候就要考虑好，减少后期适配工作量。比如你的产品是否需要支持多平台，是否有可能换肤（夜间模式），是否需要全局字体缩放，是否需要支持国际化，如果没有什么特殊的原因，最好答案是『是是是是』

解决问题效率

1. 调试辅助工具，增加定位问题速度，App Store版本注意屏蔽。
2. 内存泄露检测工具：实时提醒，有效避免循环引用导致的内存泄露问题。
3. 卡顿监测工具：当出现设定时间的卡顿之后，能打印当前堆栈到本地日志。
4. 诊断工具：本地日志上传、查看钉钉系统权限设置、iOS9企业证书信任设置。
5. 所有测试版本截图自带版本号和uid，降低沟通成本。
6. 报警机制，提前发现问题，服务用户。
7. 共创用户反馈群值班机制，为了即快速接响应，又保证其他同学专注。



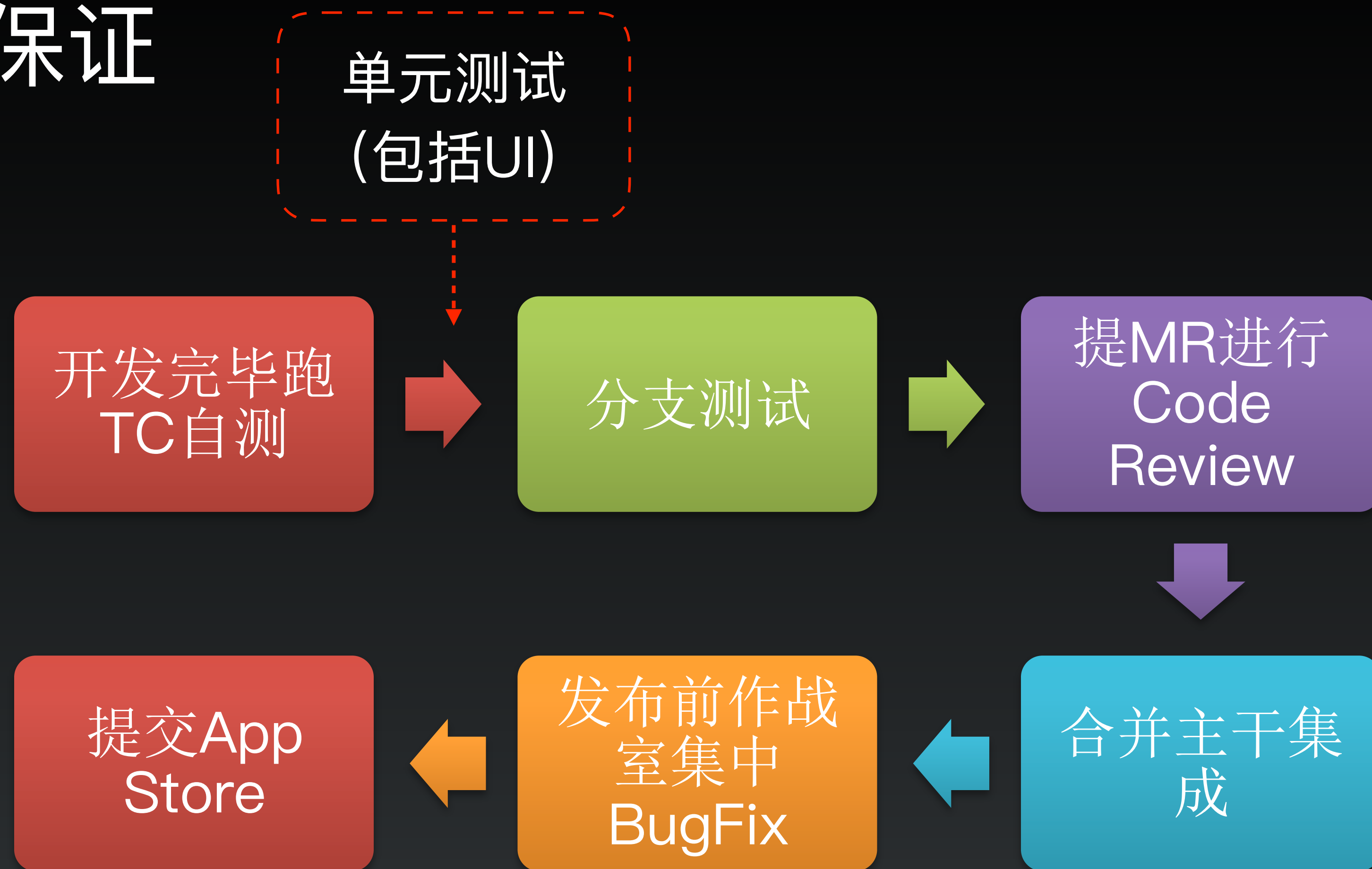
质量-工具保证

1. 内存泄露检测 & Best Practice。

2. 代码静态扫描 (fauxpas, 商业产品, 好像1000多一年, 市面上也有其他的类似工具)。



质量-流程保证



Code Review Show Cases

BadCase

- 不使用现有的公共接口，图中的trim操作，可以直接调用NSCommonModule里面的实现，直接在需要使用的地方实现，没

```
45 45 {
46 - if (!word)
47 + if ([[word stringByTrimmingCharactersInSet:
47 47 {
```

- 拒绝随意颜色值，严格执行UI字典规范才是正确的

```
91 91 _nameLabel.font = [UIFont DT_S14_nc
92 - _nameLabel.textColor = [UIColor dt
92 + _nameLabel.textColor = [UIColor DT
93 93 _nameLabel.numberOfLines = 1;
94 94
95 95 UITapGestureRecognizer *tapGesture
... @@ -416,7 +416,7 @@ static CGFloat Ding
416 416 {
417 417 // 我收到的
418 418 _nameLabel.userId = _dingModel.
419 - _nameLabel.textColor = [UIColor
419 + _nameLabel.textColor = [UIColor
420 420
```

- review说明不够详细，而且git commit日志也不够详细

Merged

fix #5756360, #5756347, #57

GoodCase

数组操作已经开始使用我们创建的安全的操作方式了。

```
533 570 self.mobileContactSection.results = mobileContactList;
534 571 self.phoneSearchSection.results = phoneSearchList;
535 572 NSMutableArray *array = [NSMutableArray array];
536 - [array addObject:self.usualContactSection];
537 - [array addObject:self.friendSection];
573 + [array dt_safeAddObject:self.usualContactSection];
574 + [array dt_safeAddObject:self.friendSection];
538 575 [array addObjectsFromArray:enterpriseArray];
539 - [array addObject:self.mobileContactSection];
540 - [array addObject:self.phoneSearchSection];
576 + [array dt_safeAddObject:self.mobileContactSection];
577 + [array dt_safeAddObject:self.phoneSearchSection];
```

BadCase

再次强调，使用block的时候一定要小心循环引用，一定使用我们最佳实践的方式来对self进行操作。

```
122 123 }
123 124
124 125 self.isLogging = YES;
126 + __weak typeof(self) weakSelf = self;
125 127 [_engine loginWithEmailAddress:emailAddress
126 128 password:password
127 129 successBlock:^(id value) {
130 + __strong typeof(weakSelf) strongSelf = weakSelf;
128 131 NSDictionary *dic = (NSDictionary *)value;
129 - _emailAccount = [[DTEmailAccountModel alloc] init];
132 + strongSelf.emailAccount = [[DTEmailAccountModel alloc] i
```

钉钉+GitLab 随时随地Code Review

The image shows a mobile interface with two overlapping windows. The background window is a DingTalk chat conversation. The foreground window is a GitLab code review page.

DingTalk Chat (Left Window):

- Header: 钉钉(98) iOS代码Review
- Sender: iOS小助手
- Message 1:
 - 【Author】 [Avatar]
 - 【Repo】 iOS-DingTalk
 - 【Branch】 develop
 - 【Comment1】 Merge branch 'develop_...' into 'develop'
 - 技术驱动 - Jsapi插件精简代码, 解决大量重复代码 - 线上
 - See merge request !3308 【Time1】 2016-05-19T20:49:36+08:00
 - 【URL1】 <http://gitlab.alibaba-inc.com/dingtalk/dingtalk/commit/>
- Message 2:
 - 【Author】 [Avatar]
 - 【Repo】 iOS-DingTalk
 - 【Branch】 develop_... fixbug
 - 【Comment1】 精简代码, 解决大量重复代码
 - 【Time1】 2016-05-18T11:41:34+08:00
 - 【URL1】 <http://gitlab.alibaba-inc.com/dingtalk/dingtalk/commit/>

GitLab Code Review (Right Window):

- Header: 5) · Commits · Tide-Alibaba / PhotoKit-iOS · GitLab
- Search: Search in this project
- Commit ID: 3f6a73
- Buttons: DOWNLOAD AS, BROWSE FILES
- Comparison: deletions
- View Modes: INLINE, SIDE-BY-SIDE
- File Path: VIEW FILE @B47FA86
- Code Snippets:

```
0FD658B /* DTPHPickerBottomView.m in Sources */ = {isa = PBXBuildFile;
B /* DTPHPickerBottomView.m */; };
0FD658B /* PHAsset+DTPhotoKit.m in Sources */ = {isa = PBXBuildFile;
B /* PHAsset+DTPhotoKit.m */; };
0FD658B /* PHAssetCollection+DTPhotoKit.m in Sources */ = {isa =
CB39D7A00FD658B /* PHAssetCollection+DTPhotoKit.m */; };
068D71 /* DTTransitionLayout.m in Sources */ = {isa = PBXBuildFile;
1 /* DTTransitionLayout.m */; };
068D71 /* UICollectionView+DTTransitioning.m in Sources */ = {isa =
CEC93C60068D71 /* UICollectionView+DTTransitioning.m */; };
04F2FDC /* DTPhotoBrowserTestViewController.m in Sources */ = {isa =
AC5666A004F2FDC /* DTPhotoBrowserTestViewController.m */; };
04F2FDC /* DTPhotoPickerTestViewController.m in Sources */ = {isa =
AC67264004F2FDC /* DTPhotoPickerTestViewController.m */; };
07A9E07 /* libPods.a in Frameworks */ = {isa = PBXBuildFile; fileRef =
```


质量-灰度机制

1. 我们创建了用户社区，维护好长期使用灰度版本的用户，凡是愿意加入这个社区的，就自动加入到灰度名单中。
2. 共创用户要区分内部用户和外部用户，受限于苹果的限制，对于外部用户，我们曾尝试用TestFlight的方式进行内测，但是太慢，往往TestFlight都还没有审核通过，我们正式版本就提交了。所以外部用户，我们更多是邀请进行一些预埋功能的先行验证和共创。
3. 阿里的人数比较多，使用iPhone的占比也非常大，所以适合进行灰度测试。唯一比较麻烦的是，内部用户要求特别高，所以对我们的服务提出了极高的要求。

质量-灰度机制

4. 目前我们可以做到每天一个灰度版本，用户反馈的严重问题，在第二天的灰度版本中一定会解决掉，这样用户积极性相对会更好，形成良性循环。灰度版本也不适合发布频次太高，按照经验值，除了发现了严重bug的，一般3到5天发一个用户接受度相对好一点。
5. 目前灰度还是采取的邀请制，所以每天使用灰度版本的人数还不多，只有几百人，我们正计划针对全员发起长期灰度测试计划。
6. 灰度测试到目前为止最大的收获：能够发现一些特殊场景下的bug，能够提前发现一些crash，有效降低线上crash率，能够发现一些重度用户才会遇到的性能问题。

团队成员都是牛人，团队的质量不可能差，效率也不会低

人员成长

1. Best Practice




2. Code review show cases


3. 每周**强制**分享，一个人业务分享，一个人技术分享（比如最近在强制swift学习分享）。

4. Scrum成员适时轮换，了解更多业务。

5. 团队同学都会owner至少一项纯技术挑战任务，和业务无关。半年为周期，自己通过工作之外时间成。

谢谢大家!

 iOS技术交流
 彦昊  3人



该钉钉群二维码将在2016-05-25失效